

“Mark Osborne’s articles are
a very useful resume
on this important issue.”
—Buckingham Palace

HOW TO CHEAT AT

Managing Information Security

**Straight Talk from the “Loud-Fat-Blocke” Who Protected
Buckingham Palace and Ran KPMG’s Security Practice**

- Design and Implement Policies and Strategies
- Understand the Design Flaws of E-commerce and DMZ Infrastructure
- Review Penetration Tests and Audit Reports

Mark Osborne

VISIT US AT

www.syngress.com

Syngress is committed to publishing high-quality books for IT Professionals and delivering those books in media and formats that fit the demands of our customers. We are also committed to extending the utility of the book you purchase via additional materials available from our Web site.

SOLUTIONS WEB SITE

To register your book, visit www.syngress.com/solutions. Once registered, you can access our solutions@syngress.com Web pages. There you will find an assortment of value-added features such as free e-booklets related to the topic of this book, URLs of related Web site, FAQs from the book, corrections, and any updates from the author(s).

ULTIMATE CDs

Our Ultimate CD product line offers our readers budget-conscious compilations of some of our best-selling backlist titles in Adobe PDF form. These CDs are the perfect way to extend your reference library on key topics pertaining to your area of expertise, including Cisco Engineering, Microsoft Windows System Administration, CyberCrime Investigation, Open Source Security, and Firewall Configuration, to name a few.

DOWNLOADABLE EBOOKS

For readers who can't wait for hard copy, we offer most of our titles in downloadable Adobe PDF form. These eBooks are often available weeks before hard copies, and are priced affordably.

SYNGRESS OUTLET

Our outlet store at syngress.com features overstocked, out-of-print, or slightly hurt books at significant savings.

SITE LICENSING

Syngress has a well-established program for site licensing our ebooks onto servers in corporations, educational institutions, and large organizations. Contact us at sales@syngress.com for more information.

CUSTOM PUBLISHING

Many organizations welcome the ability to combine parts of multiple Syngress books, as well as their own content, into a single volume for their own internal use. Contact us at sales@syngress.com for more information.

HOW TO CHEAT AT

Managing Information Security

Mark Osborne

Paul M. Summitt Technical Editor

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

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

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

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

Syngress Media®, Syngress®, “Career Advancement Through Skill Enhancement®,” “Ask the Author UPDATE®,” and “Hack Proofing®,” are registered trademarks of Syngress Publishing, Inc. “Syngress: The Definition of a Serious Security Library”™, “Mission Critical™,” and “The Only Way to Stop a Hacker is to Think Like One™” are trademarks of Syngress Publishing, Inc. Brands and product names mentioned in this book are trademarks or service marks of their respective companies.

KEY SERIAL NUMBER

001	HJIRTCV764
002	PO9873D5FG
003	829KM8NJH2
004	JK2387BSPP
005	CVPLQ6WQ23
006	VBP965T5T5
007	HJJJ863WD3E
008	2987GVTWMK
009	629MP5SDJT
010	IMWQ295T6T

PUBLISHED BY

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

How to Cheat at Managing Information Security

Copyright © 2006 by Syngress Publishing, Inc. All rights reserved. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

Printed in Canada.

1 2 3 4 5 6 7 8 9 0

ISBN: 1597491101

Publisher: Andrew Williams
Acquisitions Editor: Gary Byrne
Technical Editor: Paul M. Summitt
Cover Designer: Michael Kavish

Page Layout and Art: Patricia Lupien
Copy Editor: Darlene Bordwell
Indexer: Richard Carlson

Distributed by O’Reilly Media, Inc. in the United States and Canada.

For information on rights, translations, and bulk sales, contact Matt Pedersen, Director of Sales and Rights, at Syngress Publishing; email matt@syngress.com or fax to 781-681-3585.



Acknowledgments

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

Syngress books are now distributed in the United States and Canada by O'Reilly Media, Inc. The enthusiasm and work ethic at O'Reilly are incredible, and we would like to thank everyone there for their time and efforts to bring Syngress books to market: Tim O'Reilly, Laura Baldwin, Mark Brokering, Mike Leonard, Donna Selenko, Bonnie Sheehan, Cindy Davis, Grant Kikkert, Opol Matsutaro, Steve Hazelwood, Mark Wilson, Rick Brown, Tim Hinton, Kyle Hart, Sara Winge, Peter Pardo, Leslie Crandell, Regina Aggio Wilkinson, Pascal Honscher, Preston Paull, Susan Thompson, Bruce Stewart, Laura Schmier, Sue Willing, Mark Jacobsen, Betsy Waliszewski, Kathryn Barrett, John Chodacki, Rob Bullington, Kerry Beck, Karen Montgomery, and Patrick Dirden.

The incredibly hardworking team at Elsevier Science, including Jonathan Bunkell, Ian Seager, Duncan Enright, David Burton, Rosanna Ramacciotti, Robert Fairbrother, Miguel Sanchez, Klaus Beran, Emma Wyatt, Krista Leppiko, Marcel Koppes, Judy Chappell, Radek Janousek, Rosie Moss, David Lockley, Nicola Haden, Bill Kennedy, Martina Morris, Kai Wuerfl-Davidek, Christiane Leipersberger, Yvonne Grueneklee, Nadia Balavoine, and Chris Reinders for making certain that our vision remains worldwide in scope.

David Buckland, Marie Chieng, Lucy Chong, Leslie Lim, Audrey Gan, Pang Ai Hua, Joseph Chan, June Lim, and Siti Zuraidah Ahmad of Pansing Distributors for the enthusiasm with which they receive our books.

David Scott, Tricia Wilden, Marilla Burgess, Annette Scott, Andrew Swaffer, Stephen O'Donoghue, Bec Lowe, Mark Langley, and Anyo Geddes of Woodslane for distributing our books throughout Australia, New Zealand, Papua New Guinea, Fiji, Tonga, Solomon Islands, and the Cook Islands.



Author Acknowledgements

Thanks to Chris, Jules, Alex, and Jim plus wife`n kiddies.



About the Author

Mark Osborne is currently the CISO at Interoute Communications Limited, owner and operator of Europe's largest next-generation network. Previous to this he was the Head of the Security Practice at KPMG, where he established KPMG's Security Engineering team. This was a multimillion-pound business that he built up from scratch. Although this team no longer operates, this was one of the U.K.'s largest, most highly regarded, and most profitable security teams. Mark proudly states that managing these high-performance security experts for a period exceeding six years was one of his greatest achievements.

He holds an MBA and computing degree. He also is certified as a CISSP, CISM, CCSP and CCSE. He is generally acknowledged with publicizing many of the security flaws with WAP. He has also authored many zero-day vulnerabilities and several IDS/security tools. Most certified ethical hacker books/courses have three separate sections on his work. His achievements include:

1988 Designed and programmed a security subsystem that allowed the popular ADABAS database (used by the stock exchange and many banks) to be secured by the leading security products RACE, ACF2, or Top-Secret. It was distributed with the products.

1995 Played a part in two landmark legal cases.

Was KPMG security expert on the windup of a famous bank.

Expert witness on computer security in the cash-for-rides action (an extension of the Dirty Tricks campaign) between two major airlines. Misuse of the computer-held passenger lists was proved and an out-of-court settlement was reached in the U.K.

1997–1998 Worked as security adviser on the U.K.'s first three Internet banks. Many more followed. Subsequently, each presentation starts with the strapline that *I had broken into more banks than Jessie James.*

1998 Highlighted and publicized the security flaws in WAP. Most notable was the WAP-gap. With various papers and presentations appearing on most manufacturers' Web sites and university portals. Oh, how soon they forget.


2002 Arranged with a major manufacturer to do a series of security surveys on mobile commerce. They took 40 pieces and did a really poor job consisting of a minor war-driving exercise with a unknown boutique supplier.

As a response, I ran the first U.K. honeypot survey recording actual wireless intrusive activity at multiple locations, correlated against accepted standards of intrusive behavior. This attracted attention worldwide and was source material for many government-sponsored activities.

2003 Designed the popular WIDZ IDS and the fatajack zero-day vulnerabilities.

During this time I worked as a security manager, security consultant or security tester at or on behalf of Pru/Egg, Commercial Union, TSB, Lloyds TSB, Co-operative Bank/Smile, Halifax, Barclays, Bank of Scotland, RBS, CSFB, Barclaycard, Yorkshire Bank, Astra Zeneca, Czech National Bank, National Bank of Greece, Merrill Lynch, Sakura, Mercedes-Benz, BMW, NatWest, Fuji Bank, Hiscox Insurance, Nestle, HSBC National Audit Office, DKB Bank, Cheshire Building Society, Alliance and Leicester, Deutsche Bank, British Telecom, Cable & Wireless, TeleWest, EuroBel, AxA Insurance, Churchill Insurance, Esure, Std Chartered Bank, Hill Samuel, NaB, EBRD, BIS, Hayes, DX, various government departments, Lombard Tricity Finance, MBNA, Newcastle Building Society, Woolwich Building Society, Cedel, Singer & Friedlander, BskyB, and RailTrack.

Mark isn't a complete nerd. He is married to a wife who tolerates his behavior and two fantastic kids who see him as an irresponsible older brother.



About Interoute Communications Limited

Interoute is Europe's fastest-growing communications technology provider. Its full-service next-generation network serves more than 14,000 customers from retail to aerospace, every major European incumbent as well as the major operators of North America, East and South Asia, governments, universities and research agencies.

www.interoute.com



About the Technical Editor

Paul M. Summitt (MCSE, CCNA, MCP+I, MCP) holds a master's degree in mass communication. Paul has served as a network, an Exchange, and a database administrator, as well as a Web and application developer. Paul has written on virtual reality and Web development and has served as technical editor for several books on Microsoft technologies. Paul lives in Columbia, MO, with his life and writing partner, Mary.



How to Use this Book

This book is based on actual experience over a very unusually wide body (I also have a wide body!) of experience. For a security professional, I have operated at the highest and (probably) the lowest levels within organizations. This will bring a perspective that might be different to many texts, but might help you balance your opinions. When some technician is shouting the odds about a firewall, use the knowledge you have gained from the book to make him justify his argument.

Each chapter is started by one of my “real-life experiences”; I hope that keeps the book light and reinforces some key messages.

Contents

Preface	xxiii
Introduction	xxv
Chapter 1 The Security Organization	1
Anecdote2
Introduction2
Where to Put the Security Team2
Where Should Security Sit?	
Below the IT Director Report3
Pros4
Cons4
Where Should Security Sit? Below the Head of Audit5
Pros5
Cons6
Where Should Security Sit? Below the CEO, CTO, or CFO6
Pros6
Cons6
Your Mission—If You Choose to Accept It7
Role of the Security Function: What’s in a Job?7
Incident Management and Investigations8
Legal and Regulatory Considerations9
Policy, Standards, and Baselines Development10
Business Consultancy10
Architecture and Research11
Assessments and Audits11
Operational Security12
The Hybrid Security Team: Back to Organizational Studies12
Making Friends14

The Board	15
Internal Audit	15
Legal	15
IT	15
Help Desk	16
System Development	16
Tech Support	16
What Makes a Good CISO?	17
Summary	18
Chapter 2 The Information Security Policy	19
Anecdote	20
Introduction	20
Policy, Strategy, and Standards: Business Theory	21
Strategy	22
Tactics and Policy	23
Operations: Standards and Procedures	24
Back to Security	25
The Security Strategy and the Security Planning Process	25
Security Organization	28
Security Tools	29
Security Policy Revisited	30
Policy Statements	32
What Do I Need to Set a Policy On?	33
Template, Toolkit, or Bespoke?	34
So Why Haven't I Just Told You How to Write a Good Information Security Policy?	35
Security Standards Revisited	36
Compliance and Enforcement	37
Information Security Awareness: The Carrot	38
Active Enforcement: The Stick	40
Patch Management	40
Automated Audit Compliance	40
Summary	42
Chapter 3 Jargon, Principles, and Concepts	49
Anecdote	50

Introduction	.50
CIA: Confidentiality, Integrity, and Availability	.51
Confidentiality	.51
Integrity	.52
Availability	.52
Nonrepudiation	.53
When Is CIA Used?	.54
The Vulnerability Cycle	.54
Types of Controls	.56
Protective Control	.57
Detective Control	.57
Recovery Controls	.58
Administrative Control	.58
Segregation of Duties	.58
Job Rotation	.58
Risk Analysis	.58
Types of Risk Analysis	.59
Quantitative Analysis	.59
Qualitative Analysis	.60
How It Really Works: Strengths and Weaknesses	.61
So What Now?	.62
AAA	.63
Authentication	.63
Types of Authentication	.64
Authorization	.64
Accounting	.65
AAA in Real Life	.65
Other Concepts You Need to Know	.66
Least Privilege	.66
Defense in Depth	.66
Failure Stance	.67
Security through Obscurity	.67
Generic Types of Attack	.67
Network Enumeration and Discovery	.67
Message Interception	.68
Message Injection/Address Spoofing	.68
Session Hijacking	.68

Denial of Service 68
 Message Replay 69
 Social Engineering 69
 Brute-Force Attacks on Authenticated Services 69
 Summary 70

Chapter 4 Information Security Laws and Regulations 71

Anecdote 72
 Introduction 73
 U.K. Legislation 73
 Computer Misuse Act 1990 73
 How Does This Law Affect a Security Officer? 75
 The Data Protection Act 1998 75
 How Does This Law Affect a Security Officer? 76
 Other U.K. Acts 77
 The Human Rights Act 1998 77
 The Regulation of Investigatory Powers Act 2000 .. 78
 The Telecommunications (Lawful Business Practice)
 (Interception of Communications) Regulations 2000 79
 The Freedom of Information Act 2000 80
 Audit Investigation and
 Community Enterprise Act 2005 80
 Official Secrets Act 80
 U.S. Legislation 82
 California SB 1386 83
 Sarbanes-Oxley 2002 83
 Section 201 83
 Section 302 84
 Section 404 84
 Gramm-Leach-Bliley Act (GLBA) 84
 Health Insurance Portability
 and Accountability Act (HIPAA) 85
 USA Patriot Act 2001 85
 Summary 86

Chapter 5 Information Security Standards and Audits. 87

Anecdote 88
 Introduction 89

BS 7799 and ISO 17799	89
A Canned History of BS 7799	90
History of BS 7799, Part 2	92
PDCA	93
ISO/IEC 27001:2005: What Now for BS 7799?	98
PAS 56	99
What Is PAS 56?	99
The Stages of the BCM Life Cycle	100
Stage 1: Initiate the BCM Project	100
Stage 2: Understand the Business	100
Stage 3: Define BCM Strategies	100
Stage 4: Produce a BCM Plan	101
Stage 5: Instill a BCM Culture	101
Stage 6: Practice, Maintain, and Audit	101
FIPS 140-2	102
Should I Bother with FIPS 140-2?	102
What Are the Levels?	102
Common Criteria Certification	103
Other CC Jargon	103
The Security Target	103
Protection Profile	103
Evaluation Assurance Level	103
Types of Audit	104
Computer Audit as Part of the Financial Audit	104
Section 39 Banking Audit	105
SAS 70	106
Other Types of Audits	107
Tips for Managing Audits	108
Summary	110
Chapter 6 Interviews, Bosses, and Staff	111
Anecdote	112
Introduction	112
Interviews as the Interviewee	112
Interview 1	113
Interview 2	114
Interview 3	115

Interview 4	116
Preinterview Questionnaires	117
Interviews as the Interviewer	119
Interview 1	119
Interview 2	119
Bosses	120
Runner-up for the Worst Boss in the World	120
Worst Boss in the World	120
Worst Employees	122
Summary	122
Chapter 7 Infrastructure Security	123
Anecdote	124
Introduction	124
Network Perimeter Security	124
The Corporate Firewall	126
Threat Analysis	127
E-mail Protection	128
Browser Content Control and Logging	130
Web and FTP Server	131
Remote Access DMZ	131
Threat Analysis	131
Remote Access Design Options	132
E-commerce	133
Threat Analysis	136
Threat Analysis	139
Just Checking	140
Summary	140
Chapter 8 Firewalls	143
Anecdote	144
Introduction	144
What Is a Firewall, and What Does It Do?	144
Why Do We Need Firewalls?	146
Firewall Structure and Design	147
Firewall Types	147
Screening Routers	148
Application-Level Gateways or Proxies	148

Circuit-Level Gateways	149
The Stateful Inspection Firewall	149
So What Are the Features You Want from a Firewall?	151
Stateful Rule Base	151
NAT/PAT	151
Antispoofing	155
Advanced Logging	155
User-Authenticated Traffic	155
IPSec Termination	156
Ability to Define Your Own Protocols	156
Time-Based Rules	157
Other Types of Firewalls	157
Stealth Firewalls	157
Virtualized Firewalls	158
Commercial Firewalls	158
The Cisco PIX	158
Features	159
Adaptive Security Algorithm	159
Cut-Through Proxy	161
Failover	161
Configuration	163
Check Point FireWall-1	164
How It Works	165
The Gory Details	167
Security Policy: Global Policies	170
SYNDefender	171
Antispoofing	171
Summary	174
Chapter 9 Intrusion Detection Systems: Theory	175
Anecdote	176
Introduction	177
Why Bother with an IDS?	178
Problems with Host-Based IDSes	179
Whether to Use a	
HIDS or Not? That Is the Question	179
And Is It A Bad Thing?	180

NIDS in Your Hair	181
Detection Flaws	182
Dropped Packets	182
Fragment Reassembly	183
Packet Grepping versus	
Protocol Analysis, or Just Not Working Right	184
Lazy Rule Structure	188
Poor Deployment	188
Switches	189
SSL and Encryption	190
Asymmetric Routing	192
Poor Configuration	193
Signature Analysis	193
Anomalous Traffic Detection	195
For the Technically Minded	199
Snort	199
RealSecure	201
Summary	204

Chapter 10 Intrusion Detection Systems: In Practice 205

Anecdote	206
Introduction: Tricks, Tips, and Techniques	206
Deploying a NIDS: Stealth Mode	206
Spanning Ports	207
Tap Technology	209
Failover Monitoring	210
Aggregating Different Flows	211
Asymmetric Routing	212
IDS Deployment Methodology	213
The Methodology	214
Selection	215
Deployment	216
Step 1: Planning Sensor	
Position and Assigning Positional Risk	217
Sensor 2	217
Step 2: Establish Monitoring Policy and Attack Gravity	219
Step 3: Reaction	223

Step 4: Further Action: IPS	223
Firewalls, Master Blocking, and Inline IPSes	223
Host Detectors	224
Application Interface	224
Honeypots	225
Information Management	225
Log Management	225
Console Management	226
Logical Access Controls	226
Incident Response and Crisis Management	227
Identification	229
Documentation	229
Notification	229
Containment	229
Assessment	229
Recovery	230
Eradication	230
Other Valuable Tips	230
Test and Tune	231
Tune	231
Reduce False Positives	231
Reduce False Negatives	232
Test	232
Technical Testing	232
Covert Penetration Testing	233
Summary	234
Chapter 11 Intrusion Prevention and Protection	235
Anecdote	236
Introduction	237
What Is an IPS?	237
Active Response: What Can an IPS Do?	238
A Quick Tour of IPS Implementations	239
Traditional IDSes with Active Response	240
In-Line Protection	241
General In-Line IPSes	242
DDoS	243

Application Firewall243
Deception Technology245
Why Would I Want One?245
Extended Host OS Protection246
Why Would I Want One?246
Example Deployments247
Dealing with DDoS Attacks247
How It Works247
Scrubbing and Cleansing: The Cisco Guard249
An Open Source In-Line IDS/IPS: Hogwash250
Summary254
Chapter 12 Network Penetration Testing	255
Anecdote256
Introduction257
Types of Penetration Testing258
Network Penetration Test258
Application Penetration Test258
Periodic Network Vulnerability Assessment258
Physical Security259
Network Penetration Testing259
An Internet Testing Process259
Test Phases259
Passive Research259
Network Enumeration and OS Fingerprinting262
Host Enumeration262
Vulnerability Scanning265
Scenario Analysis266
Reporting269
Internal Penetration Testing270
Application Penetration Testing270
Application Pen Test	
Versus Application System Testing270
Controls and the Paperwork You Need274
Indemnity and Legal Protection274
Scope and Planning275
Success Criteria275

Escalation	275
DoS	276
Social Engineering	276
What's the Difference between a Pen Test and Hacking?	276
Who Is the Hacker?	276
The Digital Blagger: Hacking for Profit	277
Hacktivists: The Digital Moral Outrage	277
White Hats: The Digital Whistleblowers	278
Script Kiddies	278
The End of the Story	279
Summary	280
Chapter 13 Application Security	
Flaws and Application Testing	281
Anecdote	282
Introduction	282
The Vulnerabilities	283
Configuration Management	284
Unvalidated Input	285
Buffer Overflows	286
Cross-Site Scripting	288
SQL Injection	291
Command Injection	294
Bad Identity Control	295
Forceful Browsing	296
URL Parameter Tampering	297
Insecure Storage	297
Fixing Things	298
Qwik Fix	299
For the More Technically Minded	299
Does It Work?	301
Summary	302
Index	303

Preface

Sometimes I'm asked why I wrote this book, and my answer can be summed up by a very simple story. While I worked for a large audit firm, I was phoned up by an auditor I vaguely knew. "Hi, I have an interview for the position of security manager next week," he said with obvious enthusiasm. "I know it's got a lot to do with passwords and hackers, but can you give me more details?"

He must have thought I hung up by mistake because he phoned back—twice!

This book isn't the most comprehensive security text ever written, but I think it contains many of the things you need to understand to be a good IT security manager. It's exactly the kind of book my auditing chum would never buy.

—*Mark Osborne*
2006

Introduction

Information security is different from many other disciplines both within mainstream information technology and other business areas. Even though there are now many good books on various areas, getting the breadth of knowledge across the many subareas is still difficult, but it is essential to success.

Unlike so many functions of IT, security is an area that requires practitioners to operate across the whole organization. A chief information security officer (CISO) or a security manager is likely to be asked advice on many aspects of security in situations where there is no alternative but to give some sort of counsel. Sometimes your best shot may be the best hope available. So the sensible security officer strives to have a good foundation in most areas; unfortunately, however, many don't and rely not on knowledge (either formal or self-taught) but instead use an authoritative tone, tactical Google searches, or the various mantras about "security policy." Those experts who know everything about everything but whose advice needs to be reversed 50 percent of the time often cost companies hundreds of thousands of pounds in project delays and even fines.

This book can't possibly prepare you for everything you are likely to come across. And in its defense, no other single volume can either, but this book is designed to be a rather good start for that preparation.

This book is designed to cover both the basic concepts of security (i.e., the nontechnical principles and practices) and basic information about the technical details of many of the products—real products, not just theory.

Throughout the book, I have tried to explain "why we do things the way we do." I don't know this because I'm very clever; let's say I know this because I'm slightly older than you and was in on the ground floor while people were still trying to work things out.

The Security Organization

The purpose of this chapter is to:

- Review typical positions of the information security function and the benefits of each
- Define the role of the security function
- Discuss the qualities of a good CISO

Anecdote

To be a chief information security officer (CISO), you must demonstrate certain key qualities to an employer. At the interview for my last position, I sat down, miscalculating the touch-down so the arm of the chair slid neatly into my pants pocket with a ripping sound. My Top-Shelf consultancy suite was now complete with air-conditioning.

I immediately announced, “I’ve ripped my trousers”—so my interviewers would know the exact source of the sound that had so obviously come from my seat. Then I said, “Now you can see that I’m not talking out of the seat of my pants.

Now that’s the voice of experience!

Introduction

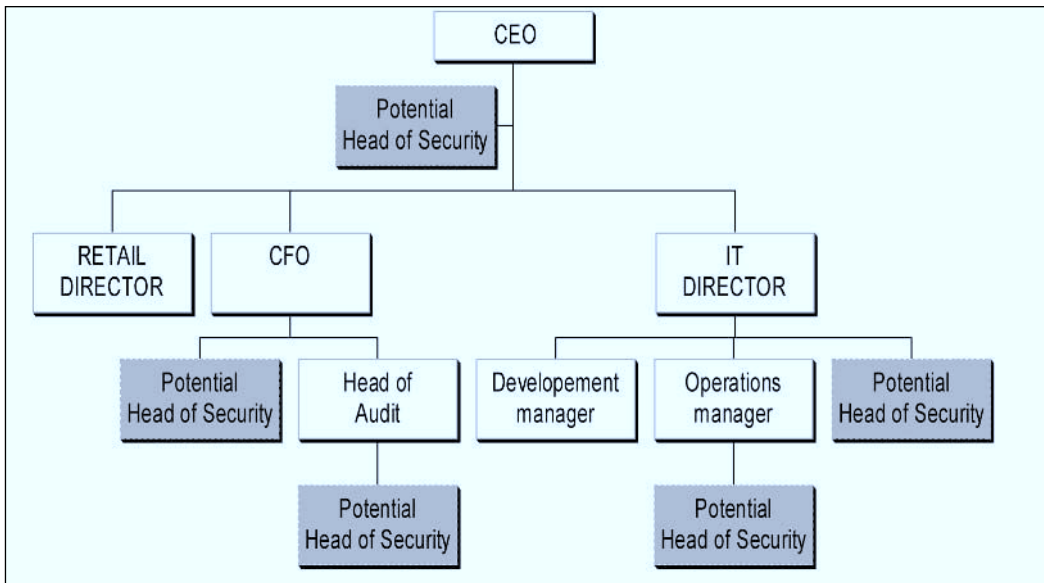
No two organizations are the same; they are always different culturally and in terms of size, industrial sector, and staff. Consequently, there is no right (but probably plenty of wrong) answer to the question, “Where should we position the head of security and the security team(s) in an organization?” Separation of the position of the operational security teams away from the head of security is often a purposeful and commercial decision.

This chapter reviews how organizations, both big and small, set up their security functions. It is based on my observations gained during 10 years experience in security consulting at both a strategic and a technical detailed level to many of the United Kingdom’s leading blue-chip companies.

I have never seen this subject covered in any textbook or manual.

Where to Put the Security Team

Figure 1.1 shows a typical firm with a number of potential positions for the security function. We will analyze the pros and cons of each position to answer the age-old question, where should information security sit?

Figure 1.1 An Information Security Organization's Hierarchy of Personnel

Where Should Security Sit? Below the IT Director Report

The most common position for the CISO and the security function is reporting up through the IT director or the head of computer operations. Certainly the latter organizational structure is common in small firms where there is no regulatory requirement for security. If the company is regulated or even quoted on an exchange, the authorities may encourage a more elevated position. Strangely enough, it is also common in more visionary firms that have had a security team for 20 years—perhaps because the team evolved from a solid team of Resource Access Control Facility (RACF) administrators (RACF is security software for IBM mainframes)!

Visit any organization with this structure and you will, within a very short time, recognize these benefits and failings.

Pros

Advantages of positioning the security team below the IT director include:

- The information security function will not receive much “outsider resistance” when it makes IT decisions, simply because it is part of the computer department. Therefore, it isn’t “external” interference.
- Operational computer security tasks (firewall installs, router access lists, and the like) will tend to be carried out by the team rather than by producing a specification for another team to execute. As a result, the team will become acknowledged local experts.
- Technical security staff can be allowed to specialize and work closely with other technical areas. Therefore, not only will there be skill transfer, but relationships should generally be better.

Cons

Disadvantages of positioning the security team below the IT director report include:

- Security will not have a powerful voice.
- Security will probably be under-funded.
- Security will not be independent; it will always be seen as taking the easiest route for the IT department. Typically, because of the low-ranking positions and the fact that it is embedded in the IT department, the focus will tend to be on *computer* security rather than *information* security. Business risk techniques to assess loss and impact will tend not to play a key role.

Obviously, in some situations this positioning will not be a big disadvantage. One of the largest U.K. banks is organized exactly in this manner. But when you are a direct report to an IT director who is responsible for 5,000 people and you have over 100 security staff reporting to you, you probably won’t feel that your punch lacks power. Similarly, if the organization has nearly all its problems within the IT department and IT is the core business

(such as with an Internet company), placement here could be a significant advantage.

Generally, however, good all-round risk management cannot prosper in this layout. The scope of the role will allow the security function to manage digital and computer security very effectively, but influence over information risk management for nondigital assets may be advisory at best. This fact will have significant drawbacks at times (such as in the security of paper files), but computing is ubiquitous these days, so the influence of the role may still be considerable. As discussed later in the chapter, sound partnering with other departments may reduce this drawback considerably.

Where Should Security Sit? Below the Head of Audit

Another far from ideal place to position a security team is to have it report to the head of the audit function. In my experience, this is where security teams are often dumped when they grow up and move from being a subdepartment of the computing department to having a wider scope.

But if you have any sort of life, you don't want to spend it with auditors, I promise you.

Pros

Advantages of positioning the security team below the head of auditing include:

- The team is independent from the computer department.
- The team will benefit from “whole business” governance mandate of the audit department. If the accounts team members are sharing passwords and you catch them, they will no longer excuse it by saying, “Oh, it’s just IT.”
- Your boss (the head of auditing) will insist that you take a holistic *information security* approach rather than just apply *computer security*.
- The security team will have powerful friends such as regulators or the audit committee.

Cons

Disadvantages of positioning the security team below the head of auditing include:

- Nobody is ever pleased to see an auditor. The team will tend to be perceived as *judgmental and reactive*, not *proactive fixers* or *problem solvers*.
- Auditors are often jacks-of-all-trades, not uncommonly struggling technically to do the jobs they do. The team will never be recognized as subject matter experts.

Where Should Security Sit? Below the CEO, CTO, or CFO

Placing security below the CEO, CTO, or CFO is the best of all the basic positions. This reporting position ensures that other departments will take notice of your findings, yet it is independent from any operational department.

Pros

Advantages of positioning the security team below the CEO/CTO/CFO include:

- The security team is endowed with power.
- It is independent.
- The position is high enough to have a “whole business” remit.
- It shows everyone that your organization is *taking security seriously*.

Cons

Disadvantages of positioning the security team below the CEO/CTO/CFO include:

- The security team will be accused of being in an ivory tower (but so what).

- The security team will find it hard to look into the IT director's business and organization.

Your Mission: If You Choose to Accept It

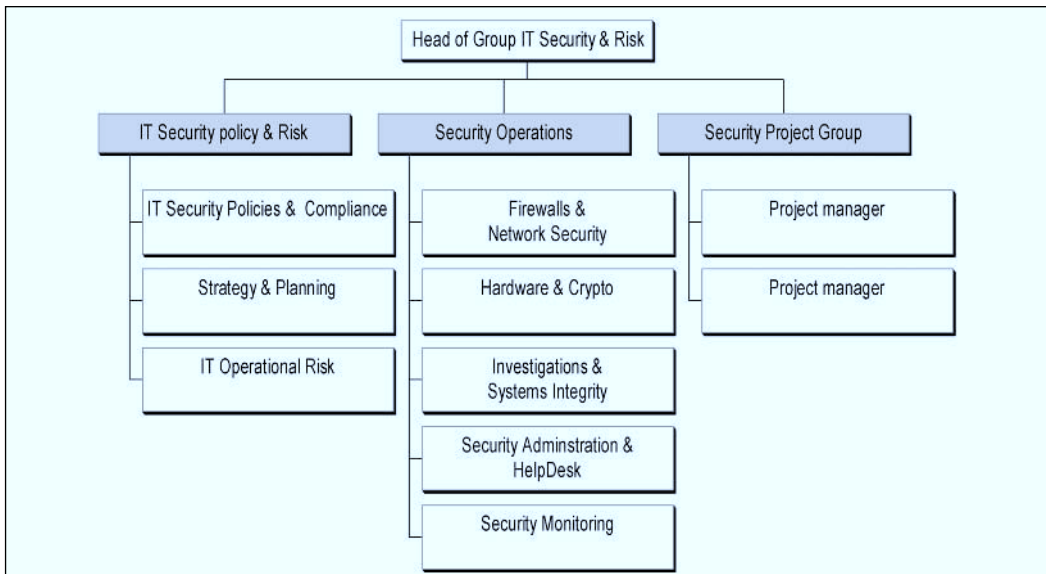
So what does a good security team do? What are the team's objectives? The answers to these questions will change from organization to organization, dependent on the particular information security strategy. The factors that may influence the answers, detailed at length in the next chapter, include legal requirements, regulatory requirements, and supplier and customer information security requirements.

This section describes the common activities of an information security department.

Role of the Security Function: What's in a Job?

Figure 1.2 shows the well-respected security team of a live organization.

Figure 1.2 A Large Information Security Team



This chart provides a good example of the roles or skills required within a security team that are needed to manage information risk. Management of information risk includes the following duties:

- Incident management
- Legal and regulatory requirements
- Architecture and research
- Policy, standards, and baseline development
- Security consultancy
- Assessments and governance
- Operational security

The following sections review each of these functions in turn.

Incident Management and Investigations

Every organization needs to deal with a number of categories of security incident. These can vary considerably in their nature and impact on the organization. Typically, the team will be involved in the full range of computer misuse activities, including:

- Viewing and transmitting pornography
- Fraudulent use of computers
- Information theft

Because of the legal implications relating to security incidents, evidence gathering, preservation, and representation are paramount. Because of the specialist skills required to do these things, often the team relies on external agencies to perform the bulk of these investigations. However, expert knowledge is still required, to ensure that you know when to call your supplier of computer forensic skills and to ensure that evidence is preserved until that point.

The other types of incident are:

- Hacker attacks
- Virus/worm detection and cleanup

The second type of incident can be the most commercially significant. Although preparing a case against a fraudster is a grave and exciting matter, containing a worm might keep your company online. Only a few years have passed since Code Red and SQL Slammer cost enterprises billions of dollars worldwide. Corporate networks collapse on a daily basis because staff don't handle this mundane area correctly.

Because most hacker attacks are relatively automated and trivial and conducted with no particular objective other than to gain access, the skills required here are similar. After all, what is the difference between an intelligent worm and an unintelligent script kiddie? Given the frequency of these sorts of events, managing them is a core skill that's essential for the survival of an organization's information systems.

Legal and Regulatory Considerations

A key role of the security team is legal and regulatory compliance. The security team must help the company and its legal advisors interpret security and data protection legislation and regulations. This task can vary from advising on monitoring of e-mails to the use of data and encryption in satellite offices around the globe (because encryption can be illegal in some countries) through controls documentation and meeting the requirements of Sarbanes-Oxley.

Increasingly, legislation is getting to grips with the concept of digital crime, data protection, and the rights of the individual. The result is that in many jurisdictions there is an increasing legal requirement to protect data or systems. For years, many companies and their directors cut costs on protecting and managing the data their organizations depended on to the extent that they actually put the organizations' viability in peril. Look at surveys from vendors or security organizations alike (www.thebci.org or www.survive.com); you will find an alarming number of companies will not survive a simple fire that destroys their servers.

Since September 11, 2001, and the Enron failure, the United States has led the world in proactive legislation that forces companies to take a responsible line on information security. In some states, for example, companies that suffer hacks that could impact customer data are obliged by law to inform the customers. (One of the following chapters provides some brief details of the legislation that U.K. companies encounter. Although not intended as definitive legal advice, this section is included as an essential primer; most security books are written by American authors and do not contain information on U.K. legislation.)

Additionally, regarding legal statutes, the security officer will also have to advise on the impact of the industry regulators, such as the Financial Services Authority (FSA) in the United Kingdom or the Securities and Exchange Commission (SEC) in the United States. These are particular to the individual industry sector of your organization and are most relevant in the health care, government, and finance sectors. Later in this book there is a whole chapter covering the basic legislation a security officer should be aware of.

Policy, Standards, and Baselines Development

Pick up a book on security and you will no doubt read that the most important document in the world, bar none, is your company's security policy. Forget the Bill of Rights, the three volumes of *TCP/IP Illustrated* by W. Richard Stevens, the data protection act, or the book that documents your faith (if you have one); the security policy is foremost.

I don't hold with this view, and for this reason I am in a minority. But there can be no doubt that a company cannot be uniformly secure, without expressing "what secure is" in general by a good, sound policy, then expanding that policy in the specific, with solid standards and operating system baselines.

Developing these and ensuring that they cover the full range of risks and technologies while remaining up to date is the responsibility of the CISO.

Business Consultancy

It was discovered in the mid 1990s that security wasn't effective in new application systems because the programmers and business managers were trying to

add it as an afterthought, which, in practice, proved very ineffective and very expensive. IBM has produced figures that show that security added into a system costs 100 times more than security designed into a system at the design stage. Obviously, adding it on later is far from ideal; this has become most problematic with the Web systems (and deperimeterization) where internal systems are exposed to noncompany users. The final chapter of this book covers this area in detail.

Consequently, it is critically important to have security input and compliance checks incorporated into the application system development life cycle of any new system. This input comes best from trained security staff and therefore falls into the responsibility of the security function.

You should ensure that your security team spends a significant body of time working with developers of new applications, assessing the type of data (information assets) the system will hold and the requirements for confidentiality, integrity, and availability. Even if it is a bought-in service, these elements should be considered. At a more technical level, your staff must be able to meet specific organizational requirements for encryption, password storage, and system logging. Doing it up front just makes sense.

Architecture and Research

Security architecture is creative envisioning of what the security regime should look like in the future. It can be very “airy-fairy.” Alternatively, it can be very practical, involving buying specific products to solve new problems, which often involves extensive research.

Typically, research involves chasing new fixes, CERTS advisories (computer security incident response teams that provide valuable security information), and *bugtraq* entries. These activities are very operational and therefore typically done by operational groups.

Assessments and Audits

To protect its information, the organization needs to make sure that the security rules are upheld across the whole organization. This is done by regularly performing compliance audits, which often can be performed by the audit team. However, technical complexity or organizational sensitivity frequently means that the information security department will get the job. Ultimately,

the security team must proactively ensure that their security policy and standards are implemented.

Operational Security

When you think of computer security, you tend to think of:

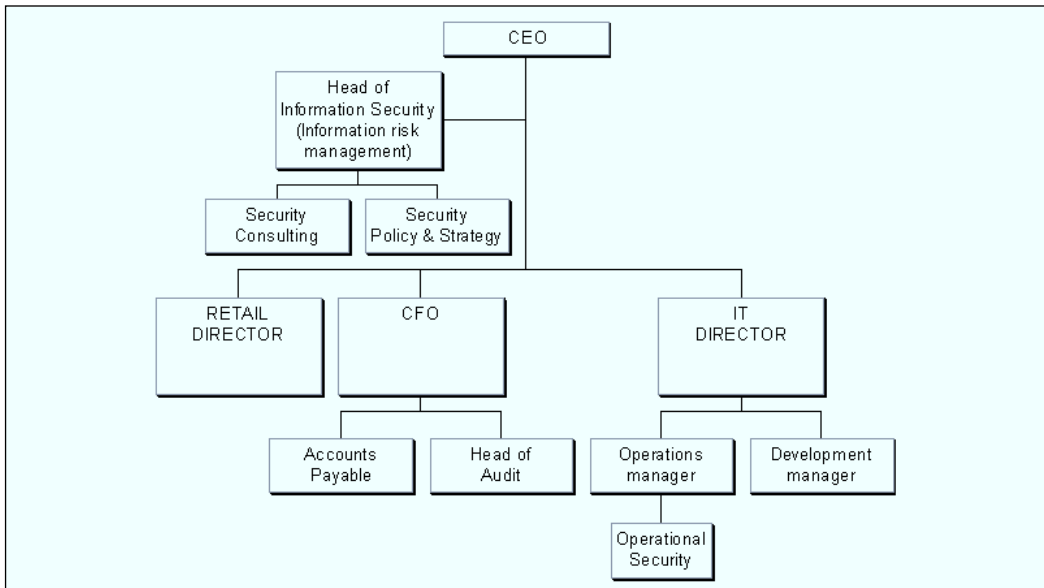
- Adding users
- Changing passwords
- Changing access lists on firewall or servers
- Reviewing security logs and security fixes

These activities can stretch across mainframes, UNIX and Wintel systems, IDSes, and firewalls. They are the essential *bread and butter* of any security framework. It may be menial, but it is essentially important. If it is over-engineered, the processes will be too arduous, causing disruption to business effectiveness and resulting in complaints that security is getting in the way. Alternatively, it could be lax, resulting in vulnerability. It might not fall to you or the security team to do these types of activities, but you must have significant control over their effectiveness.

The Hybrid Security Team: Back to Organizational Studies

Although the head of security needs to have resources at his or her disposal, all the security analysts and administrators within an organization do not have to report to the security head. Figure 1.3 illustrates this fact.

The various roles of a security department are shown in Figure 1.3, which was taken from the organization chart of a major bank. Although the function names might not correspond exactly with the titles used in this chapter, a relationship can be clearly seen. It should be noticed that the technical computer disciplines are quite distinct from the other risk management functions. This gives us the opportunity to locate them in a part of the organization together with other operating systems and network specialists—maybe reporting to the head of computer operations. This means that they will gain expertise; it also has the advantage that their ideas will gain acceptance more readily with other technicians, if for no other reason but to keep the peace.

Figure 1.3 A Hybrid Information Security Organization

The head of security and his or her compliance team would still need to be independent from IT and report to the CEO, so a split security group is formed. This hybrid security organization, often known as an *information risk management team*, is becoming increasingly popular in larger organizations.

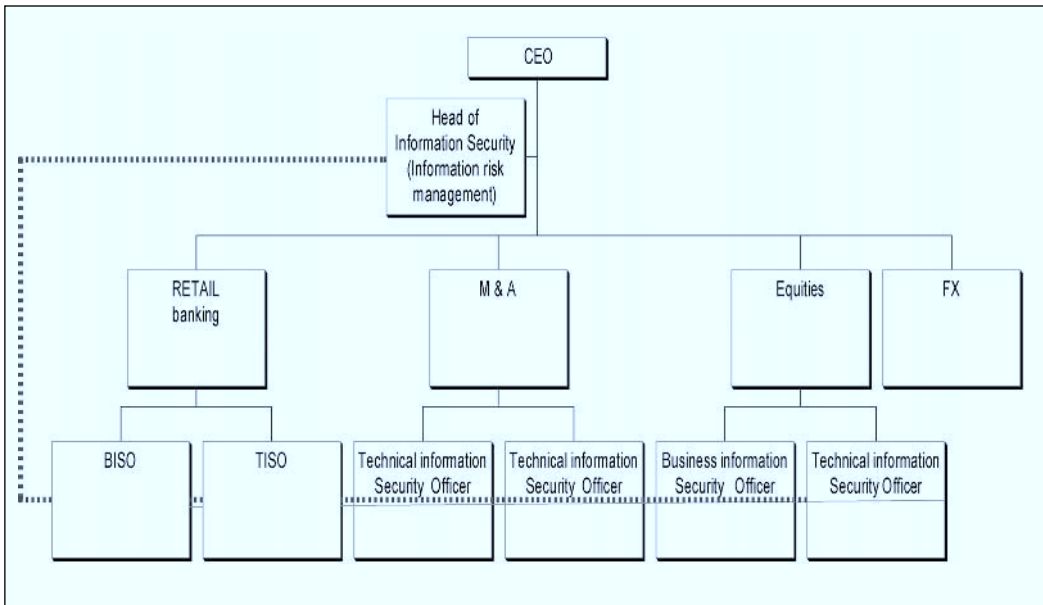
This approach can work very effectively where the organization is centralized. However, concentrated organizations like this one will not perform well in many modern organizations that exhibit the following characteristics:

- **Extreme diversity in terms of location** This structure would be insufficient in an organization that operates IT and information processing on a multinational scale. But it will also fail even if multiple administration centers are geographically dispersed in the same country.
- **Functional and divisional diversification** Often, large firms have many divisions. These firms could also operate in international markets, so could have much in common with the previous category. But small firms can have distinct, separated divisional structures because they have been successful in multiple markets or because they have

grown by acquisition and have wanted to maintain the identity of component firms.

Many organizations overcome this “organizational distance” by implanting a divisional security officer in each division. Others extend this principle and include technical staff. Typically, they operate with dual reporting lines (see Figure 1.4).

Figure 1.4 Positioning Divisional Security Officers in an Organization



Making Friends

It’s true that you catch more flies with sugar than with vinegar (although I’m not at all sure why anyone would want to catch flies!). Likewise, if the sensible CISO studies his or her mission, extracts what is needed to get the job done and finds out who else in the organization wants to achieve that goal too, that CISO can find powerful allies.

The Board

It is essential that the IT security department, specifically the head of information security have exposure to and credibility with the board of directors.

It is likely that if senior managers don't see the information security team adding value, they will make budgets small and limit the team's involvement in projects that really require CISO input. This in no way differs from any other department head, but it is something that often is done badly by security officers.

Internal Audit

The IT security team needs to ensure the security of the organization's systems. If the organization is large enough, it will have a computer audit team and/or a computer audit program. They too have to assess and report on the security of systems, which means you both have compatible goals.

With a bit of cooperation, you can offload some of your assessment requirements onto the IT audit function. In the past, I have actively provided the opportunity for computer audit staff to take a very active role in a few exciting new projects. In exchange, I demanded that they do the majority of the compliance work (such as ensuring that things are ticking over nicely) at a level and depth that could give me confidence, which made for a classic win/win scenario.

Legal

If you have a legal and compliance department, make contact with them. They will be a good and authoritative source of legal advice within the firm. They will need your opinion in some cases, so it can be a win/win situation.

Don't rely on Google for legal advice. If a case comes to court, Internet Explorer will be a weak counsel.

IT

The IT department staff are the worst culprits for bad security practices, but they also can help in specific areas. Build bridges.

Help Desk

Sharing responsibilities with the help desk can really reap benefits. All companies suffer security incidents, and the help desk is the first place that victims are likely to call. Good training with incident response and security will pay dividends.

It is also good to hand off simple security operations tasks to the help desk. Password resets and unlocking quarantined e-mail are time-consuming tasks that can be semiautomated and then handled by the help desk.

System Development

These days, most new applications systems either are Web based or have databases; both are notorious for bad security. This makes interfacing with application system development even more important than it used to be.

As covered earlier in this chapter, instill a culture in which project managers automatically involve the security team when they're designing new systems, and you'll catch the problems early.

Tech Support

Any firewall you want to use, any server that is built, any IDS you evaluate involves interaction with tech support staff. If you know what you're talking about, show them you understand—then take them for a drink. If you are *not* up to the mark technically, still take them for a drink. It might oil the cogs.

What Makes a Good CISO?

I now specialize in working for new companies that have grown well beyond the name “startup” and are truly enterprises or new multinationals in their own right. These companies employ someone like me because they need good governance systems, since they will soon float on a stock market or be bought by a large company. For this reason, I seem to spend a lot of time talking to the press, whose favorite question is, “What makes a good CISO?” Here are my favorite answers, based on years of picking up info by rubbing shoulders with the late and great (and always remember, do as I say, *not* as I do!):

- **Be an ambassador for security** Everybody knows that security doesn’t get the exposure it deserves; getting it is your job even while you set an example. If you don’t fill in the access request forms or don’t bother to wear the building pass, why should anyone else?
- **Be a leader** You will be expected to be leader, not a quiet mouse-like freak in a crumpled suit or a man with a brown shirt and clip-board that looks at fire extinguishers (that’s a car park attendant). Risk being wrong. Take a plunge. Have a strong opinion.
- **Have some original ideas** Security as a profit center is almost as tired a mantra as fear, uncertainty, and doubt (FUD), but you will be expected to be commercial. Try to make or save the company some money every single day.
- **Know your field** There are so many people who know nothing about security and still manage to use gamesmanship to make their way to the top that it’s made people skeptical. Your job is to know information security. Don’t believe the hype about understanding the business; lawyers need to know law, engineers need to know physics and mechanics, and you need to know your subject.
- **Have integrity** As I said, gamesmanship has allowed many CISOs to claw their way to the top. That doesn’t mean they are good at their jobs. You have to let people know you will stand your ground when you think it matters.

Summary

This chapter covered practical information rarely found in other volumes on information security management. Yet the factors we discussed are becoming increasingly significant as the subjects of auditing inspections and regulatory authorities. Details covered included:

- **Reporting lines** The various advantages and disadvantages of the most common positions for the security team within an organization. These included the team being based in the IT department, where the role suffers a bias toward computer security, or based in the auditing department, where the role can lose a proactive role and become a mere subsection of auditing. Although reporting to the CEO or CTO is the most influential and ideal placing to bring a “whole organization” aspect to the role, there is a risk that the CISO could lose touch with operational activity.
- **Team functions** The typical functions that a chief security officer and his team should perform vary from organization to organization. Functions usually include:
 - **Operation security** Adding user accounts
 - **Security consultancy** Working with application developers or potential suppliers to ensure that the overall security regime is maintained
 - **Compliance** Advising the business on the impact of law and regulation
- **The CISO** Lastly, this chapter covered the qualities that make a good CISO.

In the next chapter, we will review ways to define your information security regime and the part the security policy plays in its enforcement.

The Information Security Policy

The purpose of this chapter is to:

- **Review the purpose of the information security policy**
- **Relate the policy to security guidelines and baselines**
- **Define the information security strategy and a methodology to develop one**

Anecdote

Those of you who are kind will enthusiastically remember the early (and might I say ground-breaking) work I did on wireless security. As a result, one day the CEO of a very large insurance company had seen a piece in the Wall Street Journal on my work and demanded that his security manager set up a meeting with me.

Dutifully, off I trotted with one of my staff who always had a copy of Kismet on a Zaurus Linux PDA (war-driving software on a palmtop computer). I explained most of the risks of wireless Ethernet to the corporate guy, emphasizing that they had been overexaggerated in the press, but sensible precautions really were necessary. The company's security manager, a past-retirement-age duffer, seemed to be enraged by my very existence. Cutting me short, he announced, "That can never happen here; our information security policy strictly forbids it." When I asked whether he had conducted any wireless audits, he announced, "I don't need to. I know we don't have wireless; it's prevented by the policy. Now do excuse me for a minute; I have something important to do."

While he was gone we whipped out the war-driving PDA. When he returned, we showed him that within his building there was a wireless access point called Marketing, one with the same name as the company, and two with defaults of tsunami and net-gear. Plus we could connect to two of them and get an IP address. When we showed him, he just repeated, "Our information security policy strictly forbids it." Which was an entirely different reaction to that of his CEO, who phoned me a week later!

And the moral of the story is, an information security policy is an administrative control. If it is not actively enforced, it will provide little protection.

Introduction

These days, any book on servers, firewalls, or protocols will have a whole section, if not several chapters, on why the security policy is the most important component of any firm's information security defense—that the policy should be held in awe, like some sort of paper deity that anyone in the industry should pay homage to.

What a load of old nonsense! Such talk has held back the information security industry for many years and provided shelter for industry rogues who have no specialist knowledge or skill. They are wrong, and I'll show you why:

- Did a policy ever stop a hacker breaking into a network? No, that was a firewall.
- Did a security policy ever assess the security of a network link and then deem it *too dangerous* and so force its disconnection? No, that was a brave security auditor.
- Did a security policy ever stop a worm in its tracks? No, that was antivirus software.

Of course, the hacker didn't stop his exploitation of a hack on a truly vulnerable IIS server when he heard that the organization had a really good security policy. Even some of the best organizations with the best policies didn't stop the MyDoom or Nimda viruses. But maybe, just perhaps, these best policies meant that many corporate computers around the world had up-to-date virus scanners that prevented their infection from these viruses.

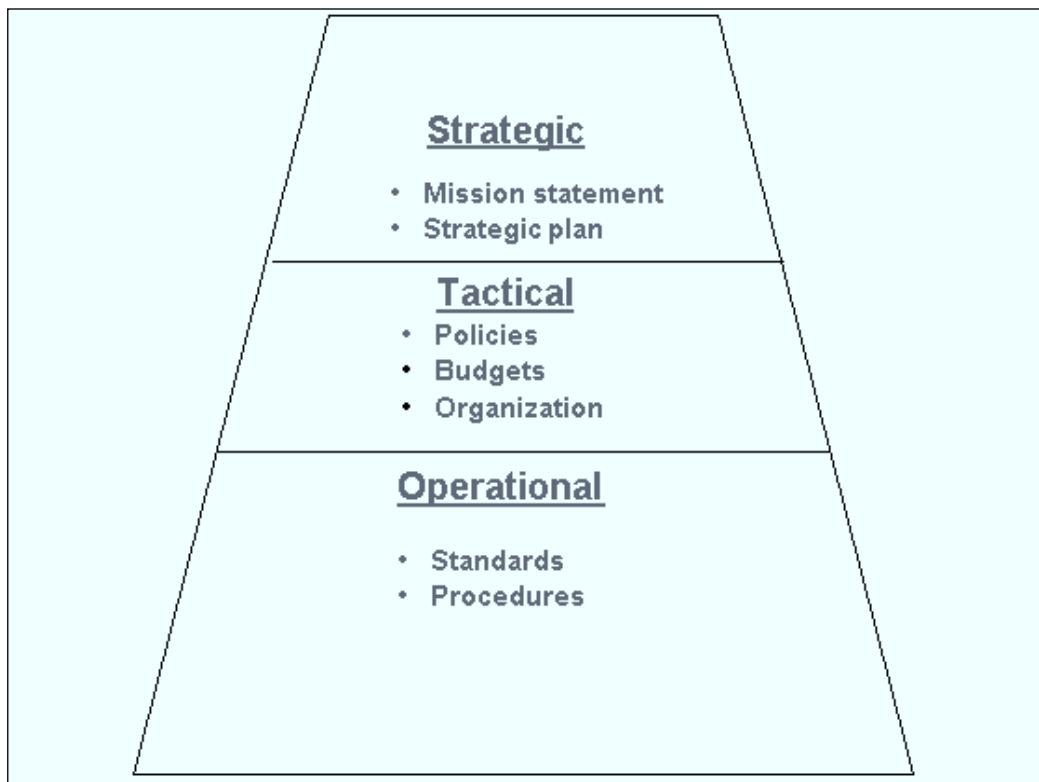
That is the key. Security policies don't provide good security; they *define* security for an organization and mandate its use.

There is no doubt that your information security policy is important, but many new security managers focus on it to the exclusion of other important management areas. In the following sections I would like to forget about security for a bit and review general business theory. The point is to show that the policy is only part of a vital management process that is consistent across human resources, payroll, or sales and produces essential characteristic of your management regime.

Later in the chapter, we will return to security and evaluate the policy-setting process in light of these new insights.

Policy, Strategy, and Standards: Business Theory

We are all aware that there are different levels of management (don't say *asleep* or *awake*) and that these operate on different time scales. Figure 2.1 shows a classic representation of these management levels as a hierarchical structure.

Figure 2.1 The Three Levels of Management

Each level is described in the following sections.

Strategy

Most people have heard the term *strategy*. In business, a strategy is a high-level future objective—simple vision of what we, the firm, will be doing in the future. How far into the future? That depends on the volatility of the business environment. Typically a business will set a strategy to be enacted over a five- to seven-year period. You’ve heard of a seven-year plan, right? This period isn’t set in stone, but three to seven years is representative of most businesses.

An example strategy, sometimes known as a *mission statement*, for a hypothetical software business might be “*To become a top-five in-category accounting package provider for the Windows/Intel platform targeting FTSE 1000 customers (but excluding FTSE 250) based in the United Kingdom and Europe.*”

As mission statements go, this one is fairly elegant. It describes the customer, the suppliers, and the market segments. But you could not drive through any large-scale project (let alone the future of a large corporation) with this bare-bones information. You need a plan—the *strategic plan*.

The strategic-planning process expands and illustrates all key areas that need to be considered to achieve the overall strategic objective:

- Which partners to engage with
- Whether to develop off shore or outsource
- What investment will be needed and at what stages
- What branding will be required
- What the customer base is
- What assets will be involved
- What risks have to be considered

Notice the use of the word *what*; the *how* comes later.

This type of strategic analysis was popularized by such authors as Michael Porter (*Competitive Strategy*, by Michael E Porter; Free Press, 1980) with his Five Forces model.

Tactics and Policy

So, the very senior management and a whole bunch of highly paid consultants decide what we as a firm are going to do as a strategy. They then sit down and develop a blend that combines what we currently do and what we will need to start doing to achieve the first couple of years of our strategy—the first stepping stones, if you will. This process is called developing *tactics*.

But senior management can't be watching over our shoulders all the time to make sure we stay on course. They need to have lunch, attend conferences, and do off-sites. Plus we management types do like to have meetings, don't we! And, based on our example, while we are in an off-site meeting, our sales team might sell a whole range of accounting packages to small businesses in the Americas to run on UNIX—a blow for our strategy, since it takes us in *exactly* the wrong direction.

What we need is a set of rules to guide our junior management and staff while still allowing them discretion to act and use their skills. These rules and the document that contains them is a *policy*—a set of high-level rules designed to direct and govern our business doings. A policy communicates and coordinates the strategy throughout the business departments and locations. It's not educational, nor is it elaborate; it is simply a set of rules that dictate how we behave.

An example policy that would ensure that our sales team toed the strategic line regarding platform and product would be: “*Customers must only be provided equipment on the approved platform list. Exceptions must be subject to the platform approval procedure.*”

This gives the *how*.

Operations: Standards and Procedures

And so we are introduced to standards and procedures. A *standard* usually gives a detailed technical description or a specification that is to be used. A *procedure* is a detailed list of instructions to be followed in a particular circumstance.

In this instance, it is very clear why we need the standard—the equipment list. Let's look at a fairly common example for the software industry: Say that our accounting package has some pretty spectacular graphics and a voice command interface. Our company needs to test or even develop specific software or drivers for the sound cards and graphic cards in the Wintel computers. To do every single possible make of card would be a waste of effort and liable to failure, since there would not be enough hours in a day. So we test the most popular, and publish them in a baseline. That helps the customer and us. (I promised I would not talk about security until the next section, but I have lost count of the times I worked on RFC-compliant versions of IPsec implemented on Firewall Z or X, which didn't talk to each other. Live that experience a couple of times and you will soon understand the purpose of standards.)

The procedure is not quite as ubiquitous. In this case the procedure is a step-by-step list of tests that proposed equipment must pass before becoming standard.

So, to summarize:

- **Strategy and the strategic planning process** provide us future objectives and outline what we need to get. This dictates policy.
- **Policy** is driven by strategy; provides specific technology free direction—a list of do’s and don’ts. Policy dictates operational standards and procedures.
- **Standards and procedures** are precise technical descriptions of our way of doing things and are designed to avoid debate on the best way of doing a job. There is only one right way, the standard.

I hope I have convinced you that in many organizations, the information security policy has been promoted above its level of significance. It is an important part of an important process. Now, let’s return to the particulars of information security.

Back to Security

Now imagine you are an IT manager in our example accounting software firm, and the CEO storms into your office, announcing, “We’ve nearly won a contract with Central Government, but they’ve found out that we don’t have a security policy or a security officer. So *you* are the new security officer. Write me a policy.”

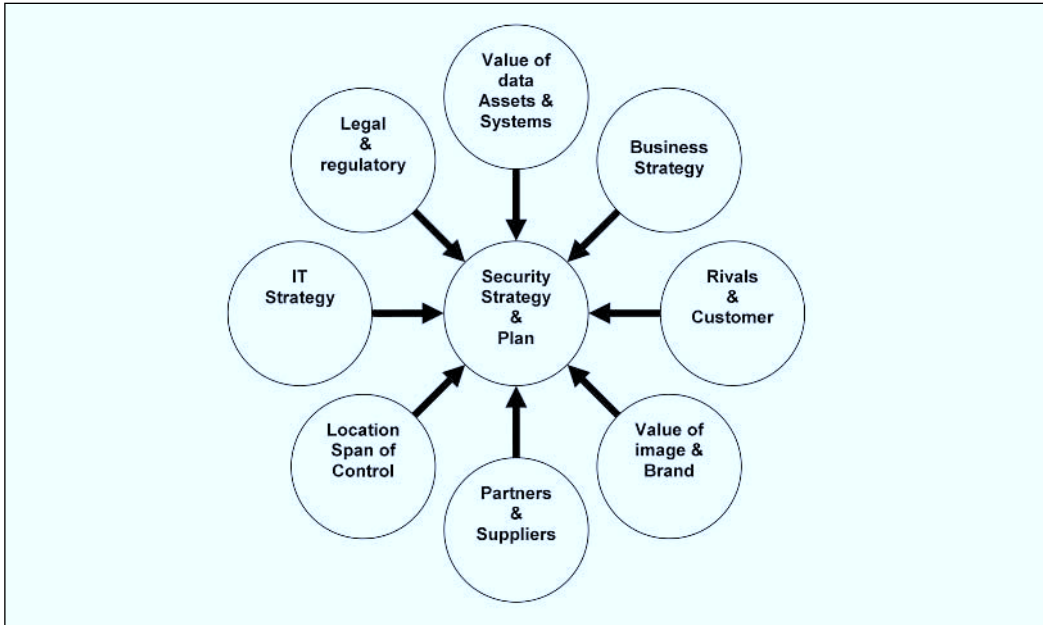
Hopefully, after reading the explanation in the previous section, you know enough to say, “Thanks for the promotion, Boss. Actually, we need a full security strategy. These government auditors don’t mess around, so we need to do it properly.” But what is properly?

The Security Strategy and the Security Planning Process

The development, appreciation, and expression of the security strategy and plan are the single most important factors in setting the security tone for an organization (see Figure 2.2). If you try to achieve a level of security that is too ambitious and inappropriate for the organization, you will not retain support from management. They will soon see that you are pumping money and

head-count where it is not effective. Yet unless you take a high-level, whole organization approach, you will never obtain the backing or the funding to enforce any policy you write. The management will just not get it.

Figure 2.2 The Factors Influencing the Security Strategy and Plan



The strategic planning process for the whole enterprise, as we described it, considered several key factors. The security planning process should consider the same factors—which is not surprising, since it is fundamentally the same process, only focused on information security. These factors are:

- **Legal and regulatory** Health, government, and finance have specific legal and regulatory requirements for security. In the case of the government, these requirements are very specific. In the banking industry, the requirements may be more stringent but open to a level of interpretation. However, most industries have to consider such legalities as privacy, data protection, and human rights legislation. For more details, see the chapter on law later in this volume.
- **Business strategy** If the CEO is determined to brand the organization as the *most secure bank in Britain*, your security arrangements must

reflect this claim, because you can be sure someone is going to put the claim to a test. If you are a small engineering firm in Birmingham, maybe you can be less rigorous.

- **IT strategy** If your IT director is determined to outsource your entire IT organization or replace all systems with SAP, your security strategy better reflect this change. You need to spend less money on security tools to deliver access controls and more on compliance audits to check them.
- **Partners and suppliers** If your organization partners with the military, you will be asked to gain security clearance for your staff and key systems. Suppliers may be forced (by legislation) to insist that you meet certain standards.
- **Value of brand image** A supplier of security software will have to have good security; otherwise, the brand image will suffer. This is similar for banks, insurance, health care, and a number of other key industries.
- **Customers** Obviously, customer expectations count.
- **Rivals**
- **Value of data assets and systems**

With regard to this last stage, methodologies for considering the value of systems and data are very rigorously documented. Many methodologies cover data asset valuation and business impact analysis processes, including the ubiquitous BS 7799 British Standard for Information Security Management, detailed at great length in a later chapter.

Where you will probably find little help is the softer side, comparing where you are now to where you want to go. This might sound wishy-washy, but unless your strategy considers where you are, it is likely to be unrealistic. If it doesn't consider your future vision, it will not help your business develop; the policies it produces will hold you back. To this aim, the addendum to this chapter contains a basic template to help you through the process.

The information security policy is just one of the outputs of this program. Other important outputs might include:

- A list of required security standards and baselines, such as the Windows security standard
- A list of security procedures, such as incident response procedures
- Security budget, such as expenditure on headcount; if headcount is not available, ensure that you budget for professional services
- A list of security tools
- Security organizations and responsibilities
- An initial, probably qualitative, documentation of the key risks and assets of the organization
- Most important, the security plan, which encapsulates much of the preceding outputs, states the security objectives plus the security posture and goals

Security Organization

Typically, the process will reveal an accurate numbers of heads required to achieve the ultimate ends. The positioning of any security group is covered in depth in the preceding chapter. However, any management system must cover reporting and responsibility:

- **Responsibility** To comply with any recognized security certification, it is important to define who is responsible for the various aspects of security—usually a security manager. Other stakeholders who may hold some information security responsibility include platform owners, security directors, operations staff, and general management.
- **Reporting and accountability** To ensure that these stakeholders are adequately represented, many organizations form a security steering committee or forum. This group sits monthly or quarterly, with the security officer reporting progress on the implementation of

the policy and standards in the organization. Incidents and other significant events may be reported.

These layers add *measurability* to the security processes. In addition to formal reports from security audits, the framework will include a *security dashboard* so that the effectiveness of security mechanisms can be determined. The security dashboard is great tool for senior management. Typically, it will include:

- The number of security incidents in a month
- Estimated cost in man hours
- Attacks on the firewall and attacks that passed the firewall
- Attacks prevented by the IPS
- The number of password resets, user adds/deletions, and ACL changes
- Virus outbreaks and time to resolution
- Softer incidents such as computer-abuse incidents
- Some measure that shows that money you spent is worthwhile—and where more expenditure would result in a saving

Security Tools

To support these activities, we need specific security tools. It is crucial at this stage that you make management aware of what you are likely to need. As part of your strategic thinking, you will have considered very earnestly your currently readiness, so it should simply be a matter of documenting your embryonic requirements and providing some time scales to give your board a list of what you need in terms of money and when. Plainly speaking, many security managers, even the best, suffer from underfunding; however, many do so because they simply didn't ask or try to predict very likely future requirements.

The following is a list of 10 likely security expenditures. If you haven't considered these features in software, appliance, or service form, you are simply not doing your job:

- Intrusion detection systems (IDSes)
- Virus protection

- URL filtering
- Firewalls
- Virtual private networks (VPNs)
- Strong authentication
- Baseline monitoring and vulnerability scanning
- Patch management and software fix alerting
- Single sign-on or identity management
- Log monitoring and correlation software

The eventual tool selection should be the subject of a separate evaluation paper/business case for each product, depending on your own organization's processes, but early sight of this technology plan will ensure that the security forum and the board are able to proactively support you when it comes to budget setting. Alternatively, it might give you an opportunity to recut your cloth to suit your means.

Security Policy Revisited

As mentioned, the security policy is driven by the security strategy. The security policy sets rules on how we treat information security. A good security policy is:

- **Directive** It should read like a set of rules, in very plain, forceful English. It is definitely not a discussion document, and we are not asking for compliance; it is simply a condition of working in your company.
- **Abstracted from technology** The policies should be written so that operating systems can be changed or augmented without a need for any changes to policy. The technology is just a tool.
- **Supported** You must ensure that the management signs off on the policy. Worse still, it is likely to be your job to ensure that management is compliant.

- **Implemental** Policies must be capable of being complied with, in a reasonable and constructive way.
- **Owned** Someone must “own” the document, its updating, and its enforcement.

A good policy should contain the components shown in the following sidebar. It is a good idea to stay with this format.

Tools & Traps...

Components of a Good Security Policy

A good security policy includes the following components:

- **Objective/purpose** This states clearly the purpose of the policy by indicating what it is designed to protect the organization from.
- **Scope** This typically says something like “this policy applies to all permanent and contract employees.”
- **Policy statements** See the section in this chapter titled “Policy Statements.”
- **Enforcement (or else)** This typically says something like “a breach of this policy will be considered a disciplinary matter.”
- **Exceptions** This typically says something like “All exceptions to this policy must be reported to the director of security.”
- **Review frequency** When it is reviewed and by whom.
- **Ownership** Who is responsible for it?

It is also a good practice to ensure that the policy contains a version number, a date, and a document reference.

I prefer a one- or two-page policy document, broken down into discrete subject areas. I believe there is more chance of these documents being read if they appear light and airy. Older-style policies tended to be one document covering the whole subject of information security, but your choice is a matter of personal taste.

Policy Statements

The policy is really only as good as the policy statements that it contains. Policy statements must be written in a very clear and formal style.

Good examples of policy statements are:

- All computers must have antivirus protection activated to provide real-time, continuous protection.
- All servers must be configured with the minimum of services to perform their designated functions.
- All access to data will be based on a valid business need and subject to a formal approval process.
- All computer software must always be purchased by the IT department in accordance with the organization's procurement policy.
- A copy of the backup and restoration media must be kept with the off-site backups.
- While using the Internet, no person is allowed to abuse, defame, stalk, harass, or threaten any other person or violate local or international legal rights.

Now, as referred to earlier, you must have established a basic asset register and performed a business impact analysis on those assets (even if it is only notional analysis in your head but based on your discussion with senior management). This should help guide the level of control you mandate in your policy (and other controls). For example, if availability of your core systems is your most pressing threat, this must be reflected in your policy. If all your assets are in the public domain, confidentiality and encryption might not be major policy areas.

To ensure enforcement, policy statements should be related to baseline configuration standards. This aids implementation and permits effective compliance checking. If you don't do this you are ensuring that the company's whole security strategy is in the hands of an anonymous server administrator; more on this later in the chapter.

What Do I Need to Set a Policy On?

I like to travel light. Table 2.1 would make a good initial policy document set.

Table 2.1 A Basic Document Set of Information Security Policies

Policy	Description
Information classification	Describes how information should be classified. Should include a data ownership policy and a data treatment table. Later we'll see how to develop a data classification policy. This is one of the more advanced policies.
Data protection	Covers data protection: How the company will manage personal data and precautions employees should take to avoid infringing on others rights.
Host access controls	Describes the: <ul style="list-style-type: none"> ■ Logon process ■ Login banners ■ Password rules ■ Audit rules ■ Data roles
Internet usage	Describes acceptable "Netiquette."
E-mail usage	Warns users about the dangers of e-mail.
Virus control	Describes the rules for virus protection and tells users what to do if their computers are infected.
Backup and data disposal	The backup policy mandates that systems should be backed up when they are in use and that these backups should be tested and protected according to the needs of the business. The disposal policy will mandate that:

Continued

Table 2.1 continued A Basic Document Set of Information Security Policies

Policy	Description
Remote access	<ul style="list-style-type: none"> ■ Disks should be destroyed before disposal. ■ CDs should be sanded and snapped. ■ Tapes should be degaussed.
Physical protection	How to access the network remotely.
Encryption	Describes physical protection.
Software licensing	Describes confidentiality.
Acceptable use policy (AUP)	<p>Describes use of legal software.</p> <p>This document is a little different from the rest because it should be educational in its nature. It exemplifies acceptable use of company facilities and IT equipment and describes forbidden activities. Banned behavior tends to include:</p> <ul style="list-style-type: none"> ■ Using illegal software ■ Viewing offensive material ■ Hacking or virus distribution or otherwise infringing on an individual's rights <p>The big question here is whether to allow or disallow personal use; the latter is becoming increasingly difficult in some legal jurisdictions.</p> <p>All policy should be linked to the contract of employment, but the AUP should be distributed with the offer letter (perhaps even with a signature required).</p>

Template, Toolkit, or Bespoke?

Speak to any policy writer and he or she will tell you that the worst thing you can do is download a set of policies from the Internet and impose them on your organization. That is absolutely true, but it doesn't mean you can't download a good set of policies and tailor them to your organization's

requirements. This will be a very unpopular view with many security managers, but here, I believe, is some very convincing proof.

When I took over the security consultancy department of a large accounting firm, I inherited dozens of Master of Science (MSC) students. One was working on security policies at a large international industrial chemical firm. Another was working on rationalizing security policies for a European investment bank. Coming from two of the best companies in the world with two of the best CISOs in charge, these security policies must be considered good, yet everybody must concede that the companies were completely different—with different sectors and different regulators and in different part of the country.

As a research project, I got one of the info sec MSC students to normalize the language (to eliminate different styles of writing) in a policy covering host access from both organizations. When we compared these two normalized policies, we found that 73 percent of the statements matched. This strongly suggests that although organizations differ, rules governing good security will remain broadly constant. Who in this day and age couldn't do with someone else doing 70 percent of their work (or this case their policy statements)? You don't have to believe me; browse the Internet, where many organizations publish key security policies. Note the different styles, and particularly note the truism of my contention.

The SANS (SysAdmin, Audit, Network, Security) Institute (www.sans.org), one of the more respected security organizations, carries a wide set of template policies. To use them, you can just do a scan and replace. I recommend a far more tailored approach (in fact, I think many of the SANS policies are not technology neutral enough for me), but it is always good to benefit from another expert's work.

So Why Haven't I Just Told You How to Write a Good Information Security Policy?

The answer is, I have. I have told you *how* to write it, but not *what* to write. I just haven't printed five dozen policy statements in a couple of chapters, prepended arbitrary titles to each dozen, and shouted "Voilà!" You can gain that from practically any volume that covers security; it produces a very bad security policy and indicates a very bad CISO. What I have shown you is that

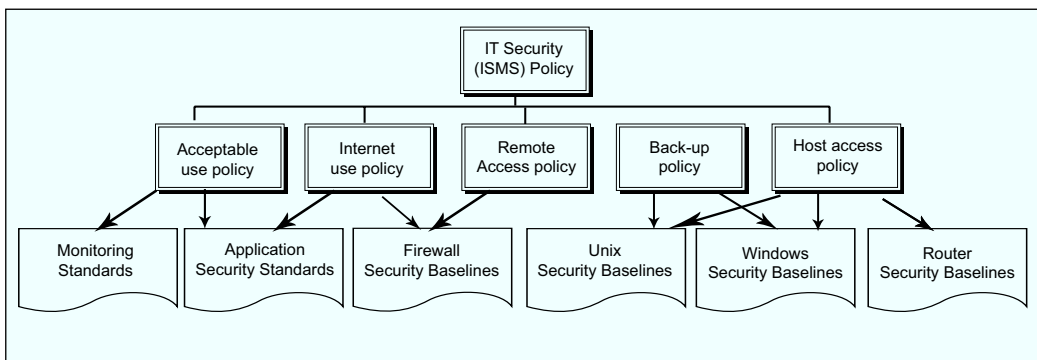
a security policy is the documentation of how you need to protect *your* information assets and systems, both now and in the future. It must take into account your asset register and how you seek to protect those assets (a typical process that is outlined in Chapter 5, on BS 7799), the laws you must embrace (covered in Chapter 4) and the business strategy for the future. However, if you need to read more, you will have to read several lengthy volumes. I commend you to *Writing Information Security Policies*, by Scott Barman, or any work by Charles Cresson Wood.

Security Standards Revisited

In the computing and networking worlds, we deal with an area of immense complexity. In managing areas of this complexity, it is often possible to achieve a policy objective in a number of ways. Therefore, for key platforms, you should implement platform security standards that dictate *exactly how* the policy will be implemented for the specific platform.

Figure 2.3 shows the relationship between policies and standards. Each policy statement is analyzed, and the appropriate Windows, UNIX, or firewall setting is embodied in the baseline, which is then mandated. Good practice settings for technical areas not covered by the policy can be incorporated from manufacturers' recommendations and other sources.

Figure 2.3 Security Policies and Standards



For example, remember the following policy statement from earlier:

All servers should be configured with the minimum of services to perform their designated functions.

This would spawn a UNIX security standard containing the following directives:

- **Inetd.conf** should not contain any of the prohibited protocols: ftp, rlogin, telnet.
- **SSH** will be used to replace these protocols.
- **NFS** is banned, so S66NFS should be deleted or disabled.
- **Sendmail** will only be started on designated mail servers.
- **DNS daemons** will only be started on designated DNS servers.

This process is critical. Unless you put the time and effort into this area, you are delegating the implementation of the policy to an engineer. Flatly, it is his or her job, and even if the engineer does implement similar settings that achieve the same end (perhaps by implementing *IPTABLES* on the platform and blocking the appropriate ports), the standardization is lost, *and you have just lost any chance of automated compliance checking.*

When we design our security functions, we must be businesslike. Businesses learned a long time ago, with Henry Ford's Model T and Taylorism, that standardization and componentization bring huge efficiency gains through automation and management by exception. This leads us nicely on to the topic of enforcement and checking.

Compliance and Enforcement

This is all great stuff, but how do you get people to be interested enough to comply with the security rules? A carrot-and-stick approach is best.

First, you need to make it as easy as possible to follow your company policies. That means, as in our examples, you need to give the technical staff no room for doubt. You need to make them sign up to a standard that is practical. Then for all classes of users, you need to run a security awareness program to make sure everyone understands that they should comply.

Second, there is the stick. This means automated compliance audits and active enforcement. That could mean you being a “mean person,” but rules are rules.

Information Security Awareness: The Carrot

Information security is all about people. If people understand and appreciate the dangers and risks associated with mismanaging information, the exposures become measurably reduced.

Awareness programs differ from organization to organization. A very formal investment bank will require a different technique from an Internet technology company. However, here are some tips that should work in both environments:

- **Best to get them when they are fresh** Most companies have an induction process whereby they give new employees pension details and show them where the toilet is. Try and get information security included in the induction process. My last few organizations offered:
 - **A short (one hour) “first day” induction session by HR** Get a five-slide show together on passwords, viruses, and the like and then coach the HR people on how to deliver it.
 - **A company induction day, conducted with a group of new employees a couple of months after hire** This “getting to know the company” session is good practice. Get a half-hour session there. Talk about the cost of security. Ask employees if they think they should be fired and prosecuted for viewing illegal pornography in the workplace—that focuses the mind.
- **Focus on the IT department** The IT department can be the greatest ally or the worst offender (every server administrator will know more about it than you), but the distinct areas must be treated differently according to their roles:
 - **Help desk and first-line support** These are the guys that get calls from social engineers about viruses and about things not working the way they should, which could be an indication of an upcoming attack. Getting these guys on your side is important; as

long as you can teach them something, they will usually reciprocate. Teach them about incident response and intrusion. Tutor them on the importance of not sharing passwords. This team will know if there is a new contractor in the building who hasn't got a badge or a system account.

- **Technical support** Tech support staff can provide insight into operating system security. Ensure that they buy into the standards for the operating systems. Be prepared to change the standards in deference to their expertise. Help them fight a few battles. Conduct a vulnerability scan of pre-hardened and post-hardened hosts, then present the results to them.
- **Application development** Get security consultancy written into the development process so that every new system development starts with business impact assessment on data CIA. This way, risks and countermeasures can be designed into new applications. Do a demonstration of cross-site scripting and SQL injection (see the chapter on application security) on vulnerable systems to show that security is just domain of tech support.
- **Execs and board members** Getting and keeping sponsorships. A lot of managers believe security is a waste of time, but you need to keep their attention, so try the following:
 - Keep security war stories about competitors flowing to them.
 - Make sure that your security dashboard emphasizes how well you do in areas where you have had budget. Make sure it shows the potential improvement where you haven't.
- **Ongoing** Conduct brown-bag sessions over lunch about latest issues. If nobody turns up, eat your sandwich and tell everyone it was great success.
- **Don't display posters with padlocks** Or a pink elephant called Snorky who tells you not to write your password down. Both look dorky.

Active Enforcement: The Stick

In the previous section on awareness, we affectively ensured that everybody understood our security rules. Now we are going to use active security exposure management to identify and peruse all employees that signed up to the rules and are now ignoring them.

The term *active security exposure management* covers a number of activities:

- Patch management
- Automated audit compliance

Patch Management

It is essential that IT platforms are updated, in a timely manner, with important security patches. It is also essential that systems and processes are in place that allow us to:

- Be alerted to manufacturers' security updates and alerts
- Adequately assess the impact of a given security vulnerability based on querying a configuration management system to establish the number and classification of devices that are affected
- Implementing a tested fix (or workaround) through normal change control mechanisms at an appropriate priority

This system fits very neatly in with FIRST's CVEE assessment methodology (check out www.first.org).

On less critical systems and desktops, I prefer my IT department to use automated means as much as possible. I have a third-party service that independently produces reports of patches their patch management system should have applied in a given month, based on a profile I give them. I review any gaps with platform owners monthly.

Automated Audit Compliance

Automated compliance audit allows us to turn compliance auditing into a monthly or quarterly exercise.

Proactively Monitoring Configurations

You need to proactively monitor configurations to detect noncompliance. Because you have mandated very precise security baselines, you should easily be able to generate baseline templates from your security standard in an audit tool. Commercial tools for key operating systems are available from Symantec, NetIQ, and ISS. Cisco provides the Cisco Security auditor for its firewalls, routers, and switches.

If cash is short, there are plenty of public domain tools that can be used: Microsoft security checker, RATS for auditing routers, and many more (COPS, SAINT, TAMU).

Regular Vulnerability Scans

Use network scanners to scan key devices. Only use scanners that have a memory or that can do a differential scan. These only report on changes and can be automated so that they send you an e-mail for review. For things that can't be checked regularly, run a manual rolling program of audits.

Whether derived from an automated audit or manual audit, the results should be formally recorded. Report all issues to the platform owner and allow him or her to comment. If there is a good reason for the exception, make sure you document the exception formally. Then enter all unresolved compliance failures into an exposure register, which is reviewed at the security forum at every meeting. When an item is fixed, remove it from the register.

Summary

In the first part of this chapter, we put information security to one side and reviewed general business theory. We learned that at each level of management, control is enforced by standards and procedures, policy, and strategy. At the end of the chapter, we learned about monitoring and enforcement, which provide management the ability to monitor the effectiveness of their control. Understanding this fact allows us to apply basic good management techniques to our specific mission of information security management. In turn, this allows to appear businesslike and effective to other parts of the organization; security management often has little to do with computers. (If you would like to know more, refer to the works of Stafford Beer on diagnosing the system.)

Having set the commercial backdrop, we:

- Related this general theory to the generation of a strategic security plan
- Detailed the format of a policy, the type of statements it should contain (based on your analysis of assets and impact), and the type of subjects that should form an initial policy set
- Described how to produce technical standards and baselines
- Dealt with enforcement—how to make sure it is working

NOTE

Value this chapter and the small pearls of wisdom it contains. To the disinterested or unperceptive layperson, it might appear lightweight, but there is a lifetime of experience represented in these few words.

The following addendum to this chapter contains a basic methodology that can help security officers capture the information security strategic requirements and consolidate them into a plan.

Addendum to Chapter 2: A Security Program Framework

This addendum details a methodology I have used many times when running a security program to develop and document a security plan. It is based on “envisioning.” Don’t be totally put off by the “consultancyspeak”; the term only means that you need to “know what you need to do before you start, but are big enough to listen to others.”

The proposed phases are:

- Phase I: Launch

Quick Wins

- Phase II: Review known security issues

Strategic Initiatives

- Phase III: Define security posture
- Phase IV: Current state assessment
- Phase V: Future state visioning
- Phase VI: Define and build

Tables 2.2 through 2.7 outline the first four phases of a security program. The objective of Phase I is to establish scope and initial resources (see Table 2.2).

Table 2.2 Phase I: Launch

Description	Activity	Deliverable
Launch	Draft scope and objectives statements. Identify candidate team. Establish costs and initial schedule.	Project planning documents (strategy, work plan, resources, schedule)

Continued

Table 2.2 continued Phase I: Launch

Description	Activity	Deliverable
Task: Define scope and objectives.	Build the initial team and assign responsibilities. Establish requirements for any particular skills (for example, telecom compliance – health care expertise). Establish management sensitivities to any tools or techniques (e.g., pen testing)	Document scope and objectives

The objective of Phase II is to identify known security problems that require urgent attention (see Table 2.3).

Table 2.3 Phase II: Review Known Security Issues

Description	Activity	Deliverable
Task: Review existing audit and security data to identify any security issues.	Review existing audit and security data to identify any security issues. Flag any red-hot issues.	Recommendations for improvement to the system and network configuration Assign a hit squad to eliminate Capture potential impact and loss expectancy; present to the board to ensure sponsorship

The objective of Phase III is to gain agreement on the business importance of security. This will dictate how we design new security features of our regime and provide context on the assessment of the current regime (see Table 2.4).

Table 2.4 Phase III: Define Security Posture

Description	Activity	Deliverable
Establish posture.	Develop an understanding of factors affecting the security regime.	A security mission statement
Task: Gain an understanding of corporate value delivered by information technologies; corporate strategy affecting information technologies; threats to technologies; relationship and contracts with customers and how security impacts them; repudiation and brand impact of security problems; and legal, commercial, and regulatory factors impacting information security.	Interview business unit leaders. This should include senior management and compliance, legal, accounts, and audit staff, if those functions exist. Interview technology management; gain their perceptions of information risk. Interview other key personnel.	Establish any misalignment between leaders of information risk; produce analysis to show impact of misalignment. Produce a subjective but agreed-on statement of the value of security to the organization—a prediction of its future importance. Sponsors and management will be asked to sign up to this posture/mission statement.

Phase IV generates an appropriate set of measurement criteria and then uses this information to determine the effectiveness of the current security framework (see Table 2.5).

Table 2.5 Phase IV: Current-State Assessment

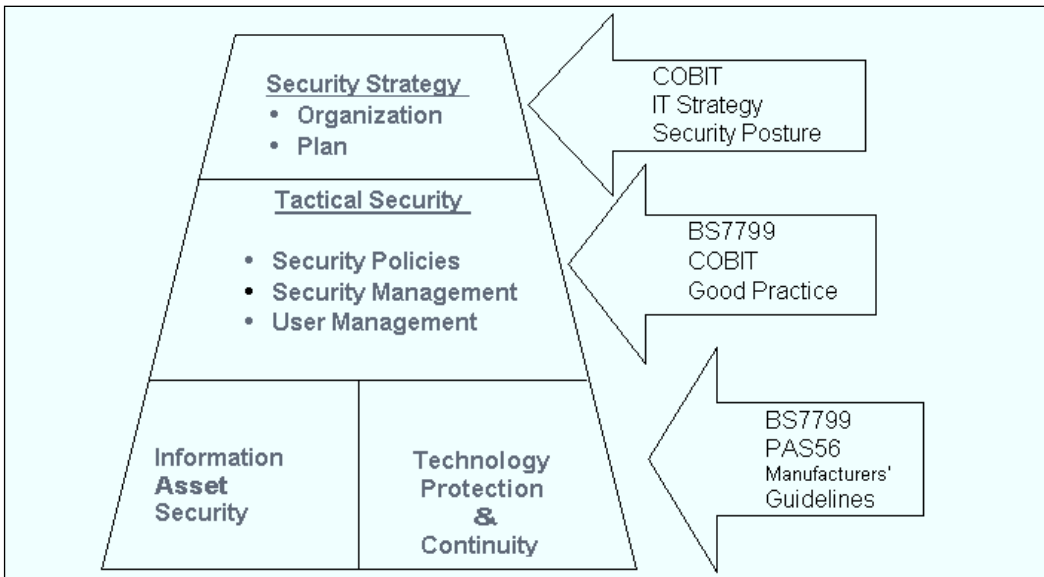
Description	Activity	Deliverable
Establish current state.	Develop current-state baseline understanding of current information security regime.	
Task: Define and agree on evaluation criteria.	Define and agree on evaluation criteria.	Produce evaluation criteria checklist
Task: Perform assessment	Interview key personnel. Review organizational charts for security responsibilities. Review network diagrams. Review security policies. Review operations procedures. Review account management policies. Review selected system configurations.	Current-state baseline of risks resulting from valuable corporate information assets, perceived threats, and gaps in control effectiveness
Task: Analyze results.	Highlight areas that require attention.	Current-state baseline presentation

The assessment of the current security effectiveness will be based on established management and technology standards (see Figure 2.4).

NOTE

Expect that evaluation should produce results well short of what people believe the current state really is.

Figure 2.4 Assessing Current Security Effectiveness



The objective of Phase V is to create visualization to establish the future vision—the objective state (see Table 2.6).

Table 2.6 Phase V: Future State Visioning

Description	Activity	Deliverable
Task: Information security creative visualization.	Imagine the possible alternatives for information security Where does it report? Is it decentralized or centralized? Is it highly matrixed? Is it a production function or a policing function?	Document preferred objective state
Task: How to get from dream to reality	Do a gap analysis between High-level information the current state and the security project plan objective state. Produce a project plan.	
Task: Secure formal management approval.	Present plan for management approval.	Approval from the management

Finally, we reach Phase VI (see Table 2.7).

Table 2.7 Phase VI: Define and Build

Description	Activity	Deliverable
Task: Define and build an information security program.	Define and build your information security program.	An information security program

Now all you have to do is do it!

Jargon, Principles, and Concepts

The purpose of this chapter is to:

- **Define jargon, principles, and concepts of the information security professional**
- **To help you understand what security people are carping about**

Anecdote

The United Kingdom's Government Communications Headquarters (GCHQ) in Cheltenham is one of the most secure places on the planet. Many documents that are more confidential than "top secret" are stored there.

I went there—went through gates, past guards, signed forms, and showed various credentials. As you know, it always rains in the U.K., and like any proper English gent, I carry a rolled umbrella. So I put my broly in the umbrella stand, which was behind the locked doors, razor wire, and cameras, and went to my meeting. When I came back, my umbrella was gone—stolen! I guess all those countermeasures weren't there for the protection of my broly.

The moral: A countermeasure or protection for one asset doesn't necessarily provide good protection for another.

Introduction

I pretty much fell into information security, and as a relatively late starter coming from a relatively senior position, I found the first year very difficult. I had very broad and very deep technical experience, so I always seemed to produce a technical analysis or solution that pleased. I had a Business honors degree, had true management experience, and had done enough work to be a partially qualified accountant, so really, I wasn't found lacking there (although nontechies always accuse techies of not *understanding* commerce). It really was the jargon that lost me; no security textbook seemed to hold the answer. Dictionaries gave definitions that didn't match the way the words were being used. Normal words with abnormal meanings but surrounded by a haze of mystique that allowed no ordinary person to play as a peer—all this put me at a disadvantage.

The aim of this chapter is simple and far too ambitious: to prevent this language barrier from impacting you, to provide you with all you need to know about the jargon of security management in one chapter.

CIA: Confidentiality, Integrity, and Availability

When I first heard someone say that they were going to do a basic CIA risk analysis, I thought it was something to do with the U.S. Secret Service and the Central Intelligence Agency. Of course, it wasn't. What they were trying to describe was some of the primary properties and principles of information that we (as security people) are interested in. These are:

- Confidentiality
- Integrity
- Availability

Confidentiality

The meaning of confidentiality in security is no different from its normal usage of the word. Confidentiality is the requirement that particular information be restricted to the appropriate people. *Confidentiality* is related to the terms *secrecy* and *privacy*, but they are not identical.

Mechanisms that are often used to maintain confidentiality include:

- **Data classification** This is the process of labeling information so that people understand who is allowed to see it and who isn't. In a military organization, you might see "top secret" stamped on a folder; in a commercial organization, "private & confidential," "addressee only," and "company secret" are labels used to maintain the correct level of confidentiality.
- **Encryption** Encryption is technical mechanism that is used to maintain confidentiality. Information is often encrypted to maintain confidentiality; only people with the right key are authorized and able to decrypt it.
- **Equipment disposal** Formatting disks seven times, degaussing tapes, shredding paper, and sanding CD-ROMs are all activities to protect confidentiality when we throw away information storage.

Integrity

Integrity is the principle that requires information to maintain its precision. It stems from the need to ensure that information is only modified, extended, or deleted by the people who are supposed to do so, *when* they are supposed to do so. Integrity is very similar to accuracy, but the terms are not identical. As security professionals, we are only concerned that information *comes out the same as it goes in*; whether it is true (accurate) is the responsibility of the data owner.

Measures to maintain data integrity may include:

- **Checksums** When we transmit data, checksums are often made and transmitted as well, to ensure that data has not been altered in transit. A *checksum* is a number produced by a mathematical function to verify that a given block of data hasn't be changed. If you rerun the function and get a different result, you know that the data has been altered. Hashes and Message Authentication Codes (MACs) are similar devices. Common examples include SHA-256, SHA-1, and MD5. In practice, an architect may implement a standard encryption scheme like IPsec or SSL rather than develop a bespoke application. These, by default, have integrity controls.
- **Access control** By ensuring that only the correct people can update, add, and delete data, we can protect its integrity.

Availability

The availability principle ensures that our data will be available in a timely manner. This principle underpins the whole principle of redundant systems.

Measures to maintain data availability may include:

- **Redundant systems** disk arrays and clustered machines
- **Antivirus software** to stop worms destroying our networks
- **Distributed denial-of-service (DDoS)** prevention systems

With the advent of e-commerce, we have recently added another principle to CIA: the principle of nonrepudiation.

Nonrepudiation

Nonrepudiation effectively defines a principle or state that ensures that an action or transaction cannot be denied: *If the system says you received this, then you jolly well did get it.* This principle has become more necessary as more *paperless* trading has occurred. In the majority of cases, these services replace the use of your signature or that of notary on a paper document. Everyone has been in a situation where someone has sent you an e-mail demanding that some action occur; the typical outcome in such a situation is just to deny receiving it. This is not a stable basis for online trading, so the concept of nonrepudiation was invented.

In his book (*Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption*, 1997, Ford & Baum) about public key encryption (PKI), W. Ford, one of the pioneers of PKI in e-commerce, establishes a number of broad types of nonrepudiation. They are summarized here:

- **Nonrepudiation of receipt** The sender can prove that the message was delivered to the right person. This avoids the situation where you can prove you sent it but cannot prove that the other party genuinely did get it.
- **Nonrepudiation of sender** This is the common case; the sender's message appears to be from, say, Mark Osborne, but can we really be sure when dealing with such a fickle character? We need to be able to prove the message is from the right person. This avoids the situation where you receive instructions or an order and then the other party denies sending it.
- **Nonrepudiation of time** No one denies receiving or sending anything; they just deny getting it at a time that makes it meaningful. Imagine an instruction to a broker demanding that some shares be sold on January 25. What do you do if the broker claims the message arrived on January 27? What does he do?

Typically, the mechanisms to meeting these principles involve digital signatures. Trusted third-party notary services can also be used to resolve receipt and timing issues.

When Is CIA Used?

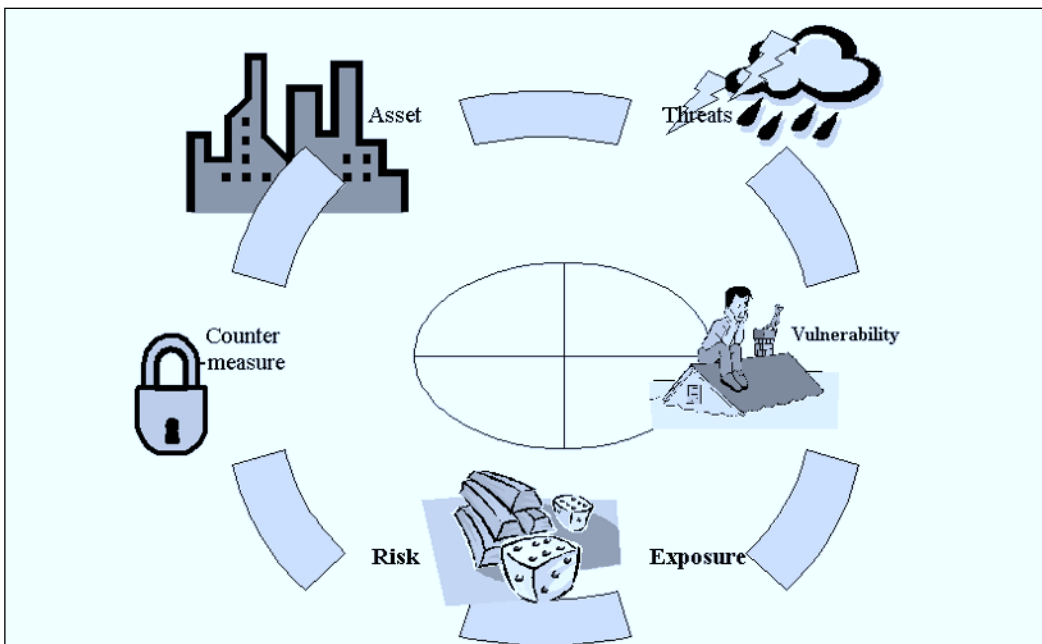
Typically, when you assess a threat, you evaluate the impact on each principle in turn:

- **Confidentiality** How will the use of packet sniffers affect confidentiality of data? Answer: Greatly! There will be a total failure of confidentiality.
- **Integrity** How will the use of packet sniffers affect integrity of data? Answer: Not at all.
- **Availability** How will the use of packet sniffers affect availability of data? Answer: Not at all.

The Vulnerability Cycle

The terminology surrounding risk is very ambiguous. Figure 3.1 is my pictorial take on it; definitions of terms used in the figure follow.

Figure 3.1 Terminology Surrounding Risk



1. **Asset** An asset is typically described as something that is needed and that has worth. In our case, the asset could be the computer system or the data it holds. Some assets are deemed to be *dependent assets*; these are assets such as disks and tapes that have no value themselves but are valuable because of the data they hold or, in the case of buildings, because of the people they contain.
2. **Threats** Threats are potential or theoretical actions that could result in damage that impairs the value of an asset. Damage, destruction, depriving the use of, disappearance, disclosure, and alteration are the kind of things we are talking about here. The perpetrator of such actions is called the *threat agent*.
3. **Vulnerability** Not all assets are susceptible to all threats, but if they are, they are said to have a *vulnerability*. A piece of code that takes advantage of a vulnerability is an *exploit*.
4. **Exposure** Each instance of a vulnerability is an *exposure*. Each exposure has a different profile and will be more or less likely to happen. The exposure will have an impact on the financial value of the asset. This is sometimes just known as *loss*. The *exposure factor* is the probability of any of this happening.
5. **Risk** This is the most ill-used word in the English language. In words, it is the quantification (not the likelihood) of the threat happening, which is meaningless. The best way to do most things in life is to express it as a formula and financial value:

Risk = Loss * Exposure factor

This is a key point: *Risk is a product of a factor of “probability of occurrence” and a factor of “expected loss.”* A significant minority of the more technical security books and papers on exposure, system survivability, and vulnerability express risk as a sum, which is plainly nonsense. If you expect a bomb to kill 30 people and you expect two attacks a year, are you likely to lose 60 (= 30 * 2) or 32 (= 30 + 2) colleagues? Risk calculation is a specialist field that emerges in many industries, but it is important to understand this basic relationship between frequency and loss—you *need* to, if you are ever going to manage it.

In information security, safety systems, and physical security, this method of calculation is predominant. In financial risk calculation, often the calculations are more complicated and work around the assessment of a bond that makes, say, a 10, 20, or 30 percent return.

According to a technical editor in engineering, the formula is:

$$\text{Risk} = (\text{Event Probability} / \text{Event Per Year}) * \text{Loss Per Event}$$

Just for convention purposes, it would be more consistent if you remained constant with your colleagues in the information security industry; doing so would aid communication and exchange of information. However, if you are particularly dogmatic about a specific calculation, as long as you use the formula of your choice for comparative analysis only, you could use the formula you are most comfortable with.

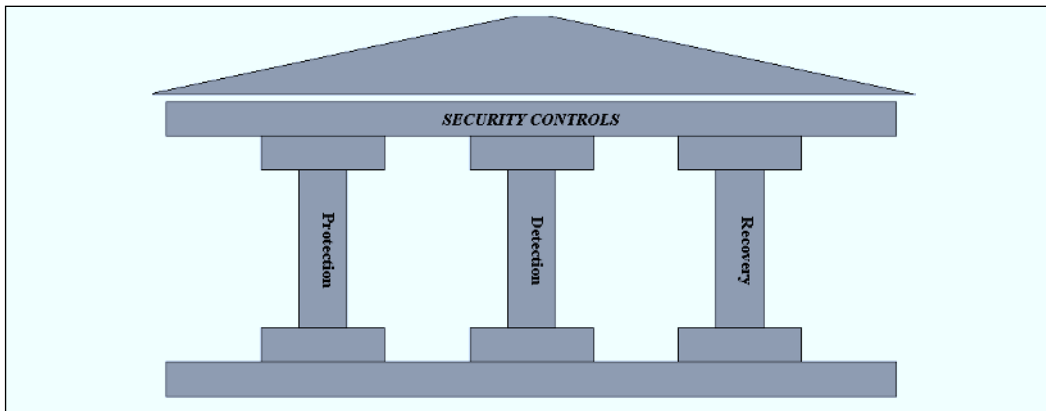
Security professionals use the $\text{Risk} = \text{Loss} * \text{Exposure Factor}$ formula because it delivers a monetary result that can be easily related to real-world spending in mitigation of the cause—the ultimate objective of most things we do.

6. **Countermeasure** A countermeasure is a protecting mechanism such as a firewall or a fire protection mechanism. This is also known as a *control*.

Types of Controls

There are many control categories. Mainly, however, we talk about three main types of control when dealing with IT systems (see Figure 3.2):

- Protective control
- Detective control
- Recovery control

Figure 3.2 Security Control

I like the image in Figure 3.2 because it illustrates the fact that system security is the combination of all three main areas of security control. When you go on to do more advanced measuring of system strengths, you will see that the security of a system is measured by time:

$$\begin{aligned} \text{System strength} &= \text{Time that it is resistant to attack} \\ &+ \text{Time that it takes to react to a breach} \\ &+ \text{Time it takes to recover from a breach} \end{aligned}$$

Not used often when dealing with computers is another category of control that cannot be forgotten: administrative control.

Protective Control

Protective controls are the classic preventative safeguards that we typically think about when we talk about security: firewalls, door locks, fences, and logon screens.

Detective Control

Detective controls alert us to security problems; they detect problems. IDS, file checksum generators, audit logs, fire alarms, and movement sensors are all forms of detective control.

Recovery Controls

Recovery controls are those safeguards that help us recover from the problem. Good examples of recovery controls are backups, alternative sites, and clustered disk storage.

Administrative Control

Administrative controls are controls that are in place because we manage ourselves correctly. They include things like security clearing staff, proper supervisory controls, audits, and references. There are two key administrative controls: segregation of duties and job rotation.

Segregation of Duties

Segregation of duties is a key type of administrative control that is fundamental to the holistic security of any process and therefore needs a separate discussion section. The crux is that you design processes so that no one person is able to easily abuse his or her privilege. In effect, segregation of duties usually means having a *doer* and a *checker*. In payment systems, for example, you usually find that one operator can raise a payment and another is responsible for checking, then releasing, the *transmission*. To defeat the system, you need *collusion* between the checker and the doer.

Job Rotation

Job rotation (sometimes referred to as *job vacations*) is the process of periodically moving roles within an organization. Rarely used outside the military, the idea is that rotating jobs prevents the risk of collusion.

Risk Analysis

The details we have covered to now have one primary aim: to help us quantify and manage the risk to our information. The quantifying approach is known as risk analysis, and these days many of you will be very familiar with some semiformal techniques. These techniques appear in the system development methodologies, project management methodologies, health and safety processes, and insurance evaluations. If I had written this chapter five years

ago, the examples and explanations would have needed to be far more detailed. However, I am going to assume everybody needs a refresher.

Types of Risk Analysis

There are many types of risk analysis. Common security risk analysis methods and tools include:

- CRAMM
- SARAH
- IS1 and IS3
- VISART
- Delphi

Most texts suggest that these methods fall into one of two categories: either quantitative or qualitative. The former is based on *math*, the latter on *expert feel*. This is certainly the approach you have to take if you want to pass your CISSP. However, the realism of the situation is that all good methods use a mix of both techniques, so they tend to vary along a continuum of more qualitative versus more quantitative. I have read some articles that suggest a qualitative approach isn't objective—complete tripe! Qualitative methods have been successful for years, and executives have been analytical since companies began. Make your own mind up; if you want to read more, I have been commended to the International Society for the Scientific Study of Subjectivity (www.qmethod.org).

Quantitative Analysis

In theory, quantitative analysis always has a mathematical basis for your grading. Take, for example, an assessment that tries to establish the risk of your main office (with a view to setting up alternative facilities).

Your methodology would work through a series of threats. Sooner or later it would come to the threat of flooding:

1. As a first step in a quantitative analysis, you would access the environmental agencies' flood data for a percentage. If it is less than 0.01 percent, you probably would not bother to analyze further.

2. DoE will give you a broad number. You might wish to contact your insurance company for a better number. On the last project I worked on, the number was 2 percent chance of a flood in a year. You now have a probability; this is known as the *annual rate of occurrence*, or ARO.
3. You use historic information from your insurer, building contractors, or the London fire brigade regarding how long it will take to clean up and get back in business after a disaster. In the example, we estimated a three-week period.
4. You contact your accounts department for the amount of revenue you would lose (at your busiest period) if you were unable to operate for this period of time. This is called *single-loss expectancy* (SLE). In this example, the SLE was \$1 million.
5. You annualize your loss due to flooding. The *annual loss expectancy* (ALE) is the product of the SLE multiplied by the ARO—in our case, $\$1\text{m} * 1/100 = \$100,000$.

Qualitative Analysis

Qualitative analysis is portrayed as being very emotional. It should be very interview based, and you would seek to talk to all major department heads. You would brief them and they would probably allocate a senior member of their department to work with you. You would then run through, either in a series of workshops or on an individual basis, the probability of each threat and rate it as high, medium, or low. Table 3.1 is a typical table resulting from such an exercise.

Table 3.1 Qualitative Analysis of Threats to a Business

Threat	Likelihood
Loss of business due to flooding	HIGH
Loss of business due to fire	MEDIUM
Loss of business due to bomb	LOW

Hence the term *qualitative*, since the use of terms such as HIGH, MEDIUM, and LOW are completely subjective and determined by factors outside the control of the researcher. You would then ask them to guess at a financial impact of such an event.

How It Really Works: Strengths and Weaknesses

No one conducts these exercises in such a banal manner. A good exercise will draw the best from both types of analysis:

- It is very important to get a ballpark figure from DoE to show the local propensity to flooding. But you as a security expert need to use your expertise to modify this information. For example, your area might not be generally at risk of flooding, but if you built your business in a bomb-proof bunker 20 feet below ground at the foot of a hill, you as an expert could decide that the risk is greater than for other standard buildings in the same locale. Especially if last Thursday, you ruined your best shoes in a puddle the size of Loch Ness getting to your car.
- The length of the outage is specific to your trade. If you use custom-built machines that take a year to build, you can't replace them in three weeks.
- You need to consult both the accounts team and the individual departments to get a *balanced scorecard* value to represent loss. It is unlikely that department managers will know the true accounting revenue of a department at a given period. They will nearly always "big it up." It is also true that the accounts department might not be aware of interdepartmental dependencies. For example, the IT department might not be a revenue center, but how many businesses these days can survive without it? Relying on accounting and reporting revenue alone is a big mistake. It certainly will not account for the following important aspects:
 - **Customer churn** In many businesses, the loss of a few customers' revenue is not as important as keeping the customers.

- **Third-party consequential loss** Will you be liable for unlimited loss?
 - **Loss of reputation** How much do you spend on telling customers that you are sinkable—that money is now wasted?
 - **Legal or regulatory infraction**
- But doing the $ALE = SLE * ARO$ calculation is essential. Risk or impact should always be expressed as a monetary value.

This type of analysis works very well for physical disasters but can be very hard to apply to other areas. This is because it requires general statistics on external threats and their likelihood but then requires you to modify them for your own local conditions. I have heard many complaints about this, particularly relating to analysis of hacks and virus exposures.

I recommend you take a look at FIRST's (www.first.org) Common Vulnerability Scoring System (CVSS). This system takes into account global factors about a threat, such as how a vulnerability compromises an operating system and how that vulnerability affects the classic CIA principles; these are provided by a manufacturer or a CERT. It then allows each individual site to consider the placement of the potential vulnerable system and the importance of that computer to the organization. It's a nice technique that combines technical and local factors.

In practice, to do this kind of job right for an enterprisewide threat will involve external statistical references and facts modified by local subjective threat modifiers. Take a look at FIRST's CVSS. This takes global facts, such as how a vulnerability compromises an operating system, accounts for classic CIA principles, and then allows each site to consider the placement and importance of that computer to the organization. It's both quantitative *and* qualitative.

So What Now?

You now know which risks affect you the most. This is your *risk profile*. Now you have to prioritize the risks based on the potential loss and deal with them in turn (see Table 3.2).

Table 3.2 Prioritizing Business Risks

Threat	Annual Expected Loss (\$)	Priority
Loss of business due to flooding	10,000,000	1
Loss of business due to fire	500,000	2
Loss of business due to DDoS attack	400,000	3

For each threat, you have the following choices:

- **Accept the risk** Make sure that the directors of the company formally document that it is a gamble that *they* are prepared to take. A member of the senior management team waving his hand saying “It’ll never happen” isn’t quite the same.
- **Transfer the risk** Typically, this means insurance, but it can mean outsourcing—for example, outsourcing the plant to a bigger organization that can provide alternative processing facilities as part of the deal.
- **Counter, reduce, or manage the risk** This means fixing the problem. Obviously the fix needs to cost less than the financial impact.

The one thing you can’t let happen is for management to ignore the risk. This process is known as your *risk treatment*.

AAA

When you start looking at network security, you will come across the acronym *AAA*, which stands for *authentication, authorization, and accounting*.

Authentication

Logging on to a computer is a two-stage process, typically. You enter your:

- **Username** This is the *identify* process
- **Password** This is the *authenticate* process. It authenticates or proves your identity as posited in the username stage.

Types of Authentication

Authentication types are typically summarized as:

- Something you know
- Something you have
- Something you are

Type 1: Something You Know

“Something you know” is the good old password. Everybody has an idea of the strengths and weaknesses of passwords. Most systems these days store the password in a nonreversible form (an md5 *checksum* of the password appended to some other data, for example).

Type 2: Something You Have

“Something you have” requires a physical object to authenticate, such as a swipe card, a number-generating token, or an X509 certificate.

Type 3: Something You Are

In the technology world, “something you are” means *biometrics*, which involves fingerprint readers, voice recognition, and iris scanners. If you are involved in the selection of such equipment, please ensure that the equipment checks only “living” body parts—that these parts are warm and have blood flowing. Such equipment should also feature *duress alarms* with accompanying procedures so that if someone is being forced to authenticate, they can make a cry for help. Some of the gory films you see on TV in relation to this technology are not based on theory but on its early implementation. In some parts of the world, implementation of cheap biometrics is simply an encouragement for villains to kidnap, extort, and hurt people. Preservation of employee safety is every security officer’s prime objective.

Authorization

Authorization is the process that establishes whether a given identity or subject can perform a given function against a given object. For example, some

user may be authorized to view data, and others may be authorized to delete data; both must be valid users, but they have different capabilities.

Authorization or access control is typically defined by access control lists (ACLs).

The authorization systems typically fall under several different types:

- **Discretionary access control (DAC)** Here the owner of the system can decide who has access to what and can divulge authority for administration.
- **Mandatory access control (MAC)** Here the key thing is that level of access is predefined. The owner cannot change it. This typically means central control. It often means control by privilege levels or security labels (open secret, top secret, and so on), but that is an implementation detail, not a specification of MAC.
- **Role-based access control (RBACs)** With RBAC, a subject is given privilege based on his or her role. This is very similar to the notion of groups in most operating systems; however, such systems usually implement arbitrary rules such as “a subject can have only one active role at a time.” Typically, these are relaxed after implementation, because the key thing is that someone has defined what type of user should have what type of access.

Accounting

The third *A* stands for *accounting*, which means logging authorization or authentication data. Most systems have the capability to do it but don't enable it.

AAA in Real Life

In a typical IT department, AAA is done by three types of system:

- **TACACS+** A Cisco invention used mainly on Cisco devices. Work in a big population of routers and switches, and you won't be able to live without it.

- **RADIUS** A less proprietary AAA system universally used by remote access solutions, wi-fi APs, and even Layer 2 switches implementing EAP.
- **Kerberos** MIT's academic labor of love, now embedded in Windows.

The thing to remember is that these features do not add to the security of a service. They are just ways of providing central control. For example, a router with Telnet exposed to the Internet is an acknowledged security risk. Adding TACACS to it solves nothing; it is still insecure. If the TACACS system holds a user/password of *Cisco Cisco*, you've made it worse.

Other Concepts You Need to Know

In this section, we address the concepts of least privilege, defense in depth, failure stance, and security through obscurity.

Least Privilege

The principle of least privilege dictates that you should grant only those privileges that are absolutely required. Don't add access rights, because they might come in handy. The major advantage of this strategy is that it limits exposure to attacks, thus minimizing the possible damage inflicted by a successful infiltration.

Defense in Depth

A common problem with all security systems is that we must assume they will fail at some point in time. The *principle of defense in depth* counters this assumption with layers of security that ensure that one breach alone is not sufficient to allow access to critical data. A typical example of defense in depth can be found in firewall architecture when you secure an e-commerce system with, say, a PIX on the outside and Check Point FireWall-1 in front of the application and DB servers. The PIX may have a vulnerability, but that will not allow access to the data because the Check Point firewall will prevent it.

Failure Stance

In the event of a failure, the *failure stance* is the state that a device is left in:

- **Fail open** When failure occurs, traffic passes freely. This option is ideal for fire doors and network taps.
- **Fail closed** When failure occurs, traffic is blocked. This option is ideal for firewalls and access control systems.

Security through Obscurity

The “security through obscurity” strategy is based on the theory that if you keep a low profile, attackers will “pass you by.” A practical application of this policy is the nonpublication of modem numbers, divulging them only on a need-to-know basis. It must be noted that although this is a sensible precaution, it is a poor basis for long-term security and would only be effective if combined with other strategies.

Generic Types of Attack

When you are analyzing a new system or protocol against malevolent intrusion, starting at the very basic primitives of CIA can seem self-defeating and long-winded. After all, most attacks inevitably lead to loss of integrity, availability, and confidentiality. For example, a successful buffer overflow attack that allows a hacker shell access will allow that hacker to impact CIA; the same failed attack may compromise availability and integrity, corrupting memory or stalling the applicable service.

Even if you are a great fan of CIA impact analysis, when it’s applied to specific protocol security analysis many feel it is too abstract and academic. Many prefer to either use common criteria analysis (documented in the next chapter) or to analyze the protocol against generic attack types, as detailed in this section.

Network Enumeration and Discovery

Not really an attack, network enumeration and discovery can be used to assess the extent to which a network will divulge information about itself. Good

examples of bad practices are route protocols that provide routing tables to any peer, just for the asking, and name services and directory services that do the same thing.

Message Interception

Message interception attacks exploit weaknesses in a network's privacy. If you can intercept a message and keep a copy (i.e., packet sniffing), you can obtain valuable data.

Message Injection/Address Spoofing

These attacks exploit weaknesses in the way a network establishes transport connections, allowing the attacker to inject traffic masquerade as a valid IP address and thus gain system access. If I know your network management system is on address 10.0.0.1 and your key system is 10.0.0.100, and if I send a system down message to 10.0.0.1 seemingly from 10.0.0.100 in an attempt to cause panic, I *am spoofing the source address*.

Session Hijacking

Session hijacking is a combination of interception and injection. It allows an attacker to avoid password protections by taking over an existing connection once authentication is complete. For example, if I am sniffing your network, I might be aware that you have a Telnet session between your network management system on address 10.0.0.1 and your key system 10.0.0.100. If I send a series of packets to the NMS on 10.0.0.1 that causes you to drop the connection but at the same time continue to send packets to 10.0.0.100 with a spoofed address of 10.0.0.1, I have hijacked the session.

Denial of Service

Denial-of-service (DoS) attacks are designed to deny legitimate users access to resources. They can involve many attackers, in which case it is said to be a distributed DoS (DDoS) attack.

Message Replay

Message replay attacks cause disruption by replaying genuine traffic that has been recorded previously using sniffer software.

Social Engineering

Social engineering is a term used to describe situations in which an attacker masquerades as a genuine employee and tricks a third party into divulging information (such as a password) that will allow the attacker access to the system. Typical examples include pretending to be an employee, phoning up the help desk, and asking for that employee's password.

Brute-Force Attacks on Authenticated Services

Brute-force attacks use automated methods to repetitively guess authentication credentials. For example, repeated attempts to log in at the Telnet prompt is an online brute-force attack. Offline attacks include using *joe-doe* or *killer-crack* to crack a UNIX shadow file or using the *crypto workbench* to find a secret key.

Summary

In this chapter we have discussed the jargon and some of the techniques associated with security management:

- **The vulnerability life cycle** This cycle describes how *threats, vulnerabilities, and countermeasures* interact with each other to provide a circle of protection.
- **The CIA triad** This is the absolute foundation for everything we do. These properties of information are *confidentiality, integrity, and availability*.
- **Standard control categories** *Protective, detective, and recovery* are categories of control that are deployed to protect CIA.
- **Risk analysis** The chapter covered the two distinct type of risk analysis, with the associated processes as a refresher for the reader.
- **Generic attack types** The basic attack types were reviewed as the basis of more detailed technical analysis.

Information Security Laws and Regulations

The purpose of this chapter is to:

- **Provide a brief overview of U.K. legislation regarding information security**
- **Provide a brief overview of U.S. legislation regarding information security**

Anecdote

A renowned white-hat hacker joined my team to replace the “management fop” who occupied the position of chief penetration tester. The new team member had previously worked for a small consultancy of no consequence and small reputation. I told him to bring along anyone he thought was good, but either the “fop” grassed on us or the conversation was overheard.

A few days later, an e-mail accusation of a hacking attempt, allegedly by my new guy trying to gain access to his previous employer’s system, arrived at HR through a friend of a friend of a friend of the HR manager’s boyfriend. Within seconds, it was possible to dispel the claim because:

- *The target system was owned by an ISP mate of my new employee. He rallied to the cause with an open-ended invite, in writing, proving no malfeasance.*
- *The evidence demonstrated obvious hacking activity but no breach of law.*
- *Best of all, the law enforcement agency my new guy was working for on his first engagement provided a statement that he was in “no position to undertake these activities” at the stated time.*

In anyone’s book, it was a trumped-up charge. But we were dealing with HR—the land minds forgot. My new employee was suspended, pending a hearing, along with a comment from the HR director that evidence from such a trusted source (her boyfriend) would inevitably lead to his dismissal. A week passed; legal told HR to drop it—but still no hearing and no reinstatement.

Eventually I could stand it no longer, so I phoned the HR manager and explained that I had just won a prolific computer security legal case between two large airlines—and I assured her, based on my expertise, that there was no case against my guy. I also informed her that because I deemed her action to be unreasonable, I had suggested that my new employee gain legal representation. “He can’t do that—it’s against our rules,” she said.

I assured her that, with very few exceptions, everyone in the U.K. is allowed to engage a lawyer. But she replied, “Ah-ha, but we are an American company—so we come under American law!” I phoned her boss and suggested that he ask her a few questions about the legal dynamics of the upcoming court case. My guy was reinstated that afternoon but all his references and health insurance benefits kept getting mislaid; I wonder why.

And the morals of the story are:

1. *Never assume corporate management understands its responsibility with regard to the law.*
2. *Knowing something about the law always helps.*
3. *Sometimes just talking to a lawyer will solve the problem.*

Introduction

It is always the best course of action to defer questions of legality to a good lawyer who knows what he or she is talking about.

The trouble is, not all lawyers are good, and even good lawyers might not be familiar with computer security legislation. So you need to know the basics.

There are loads of texts covering U.S. legislation, and in fairness, in the last couple of years U.S. law has been blazing a fine path for the rest of the world to follow in respect to championing care of data.

With California SB 1386 and Sarbanes-Oxley, U.S. legislation has begun to really make corporate executives responsible for poor security—and it's about time. This really means that an exec can no longer rationally say, "I don't care about security."

You'll be able to find loads of information on these laws in various textbooks, but few of these cover U.K. legislation too. So here it is—a whistle-stop tour of both sides of the pond.

U.K. Legislation

Let's start with the two acts that form the mainstay of U.K. IS law: the Computer Misuse Act and the Data Protection Act.

Computer Misuse Act 1990

The Computer Misuse Act 1990 creates three distinct criminal offenses:

- Unauthorized access to computers, including the illicit copying of software held in any computer. This carries a penalty of up to six months' imprisonment or up to a £5000 fine and will be dealt with by a magistrate. This covers hobby hacking and, potentially, penetration testing.

- Unauthorized access with intent to commit or facilitate commission of further offenses (such as fraud or theft), which covers more serious cases of hacking with a criminal intent. This has a penalty of up to five years' imprisonment and an unlimited fine. Because it is a serious offense, it will be a trial by jury (12 jolly good people).
- Unauthorized modification of computer material, which includes the intentional and unauthorized destruction of software or data; the circulation of "infected" materials online ("viruses"); and the unauthorized addition of a password to a data file ("crypto viruses"). This offense also carries a penalty of up to five years' imprisonment and an unlimited fine. It is also a serious offense, so it too will be a trial by jury.

This act has been the chief means of dealing with unauthorized computer access such as hacking. However, the law has been heavily criticized. I remember one of my old bosses giving lectures and stating that "You practically had to be standing over the offender's shoulder while he was doing it to get a conviction."

This comment does hold an element of truth, but the difficulty in gaining convictions is more down to the poor state of monitoring, evidence handling, and awareness in the industry, because there certainly *have* been convictions.

However, there is a need to show that the person committing the unauthorized access was aware that he or she was not authorized to access the service. Therefore, to cover the internal threat, this means you must define *authorized* and *unauthorized* activity in your acceptable use policy (AUP). Otherwise, there is a risk that disgruntled employees viewing confidential data outside their normal job access requirements could claim they were unaware of any misuse. (Review Chapter 2 for a description and content of an AUP.)

Clearly, DDoS and DoS are simply not covered by the act. In May 2002, the Earl of Northesk attempted to pass an amendment relating to these activities, but it failed.

How Does This Law Affect a Security Officer?

The Computer Misuse Act 1990 includes the following requirements:

- Your security policy must contain an AUP and be communicated to all employees.
- Your systems should contain logon banners stating that access is for authorized personnel only and must not contain a “welcome.”
- Penetration tests should be accompanied by appropriate paperwork. See Chapter 12 for a description of penetration tests and the controls needed to use them safely.

The Data Protection Act 1998

The Data Protection Act 1998 came into force on March 1, 2000. Covering the use of personal data (data relating to identifiable living individuals), it implements the European Directive on data protection (95/46/EC) in U.K. law.

The act covers manual and computerized records and is concerned with the processing of “personal data.” It works in two ways:

- Giving individuals (data subjects) certain rights over the way that their data is processed
- Requiring those who decide how and why personal data is processed (data controllers) to be open about their use of those data and to comply with the data protection principles in their information-handling practices

The act establishes a set of eight principles for the fair and secure handling of personal data. Typically, it is a breach of the principles, rather than the act itself, that precipitates action. This will take the form of a complaint to the information commissioner.

A data controller must comply with the eight principles of good practice, which require that personal information is:

- Fairly and lawfully processed
- Processed for limited purposes and not processed in any manner incompatible with those purposes

- Adequate, relevant, and not excessive
- Accurate
- Not kept for longer than is necessary
- Processed in accordance with the data subject's rights
- Kept secure
- Not transferred to countries without adequate protection for the information

These rules can be shortened to the acronym *FARSTAR*:

F—Fair

A—Accurate

R—Relevant

S—Secure

T—Transferable

A—Adequate

R—Rights

A number of Codes of Practice have been derived from this law. The most interesting is Part 3 of the Data Protection Code on Employment Practices, titled “Monitoring at Work.”

How Does This Law Affect a Security Officer?

The Data Protection Act 1998 includes the following requirements:

- You must make sure that all your employees are aware of their responsibilities under the Data Protection Act (DPA) 1998.
- You might have to register with the Data Protection registrar.
- You must ensure that you monitor your use of data so that it complies with the DPA.

- Particularly, you must ensure that personal data has appropriate access controls to ensure that no individuals' rights are infringed. If you are going to monitor communications, you must perform this activity in an informed, responsible, and nonintrusive manner.
- You must also make sure that data is destroyed in a timely manner.
- You must ensure that a data subject's rights are upheld and that any request for information held on a data subject is processed within 40 days. Because most archiving systems for e-mail and disk don't easily afford the location, extraction, modification, and deletion of single records about an employee, disgruntled employees have been making such requests just to make awkward things awkward for the employer; these requests take a phenomenal amount of time to process. Tying up internal resources this way is an excellent method for disgruntled employees to get their revenge on their employers.

Other U.K. Acts

Several other acts in the United Kingdom can impact a CISO in the execution of his duty. Typically these acts affect security monitoring.

The Human Rights Act 1998

Based on the European Convention on Human Rights, the Human Rights Act 1998 came into force in October 2000. Under Article 8 of the Convention, people are afforded the right to privacy. This not only covers privacy while people are in the workplace, it also applies to e-mail communications, Internet use, and telephone calls. Bottom line: If you are going to monitor employees, you must let people know in advance.

How Does This Law Affect a Security Officer?

Your security policy must be communicated to employees and include a warning that systems may be monitored for security purposes. Monitoring would include:

- Pen tests
- IDS

- Mail scanning
- Packet sniffers

The Regulation of Investigatory Powers Act 2000

The Regulation of Investigatory Powers, or RIP, Act 2000 covers interception and monitoring of communications. It provided extensive civil and criminal definitions and was generally unpopular because it was perceived as making life difficult for business and the private individual while allowing “Big Brother” to invade our privacy at a whim. Three major areas of impact are unlawful interception, lawful interception, and key surrender.

Unlawful Interception

RIP made it unlawful to intentionally intercept communications over a public or private telecommunications system without lawful authority.

Lawful Interception

RIP acknowledged such activity as lawful interception if it was reasonably believed that both parties to the communication consented to the interception. Additionally, provisions were included for the monitoring of communications where it was reasonably required for business purposes.

Key Surrender

If encrypted communications are intercepted, the act will force the individual to surrender the “keys” (personal identification numbers, or PINs, which allow users to decipher encoded data), on pain of jail sentences of up to two years. The government says keys will only be required in special circumstances and promises that the security services will destroy the keys as soon as they are finished with them.

How Does This Law Affect a Security Officer?

As before, your security policy must be communicated to employees and include a warning that systems may be monitored for security purposes. Monitoring would include:

- Pen tests
- IDS
- Mail scanning
- Packet sniffers

Similarly, if you are going to monitor live communications—say, Instant Messenger and VoIP—you must ensure that any external recipient is informed so that he or she is aware of this fact and can terminate the conversation if desired.

Additionally, if you are going to implement a public key infrastructure (PKI), you must provide a key recovery method; Big Brother may demand to know your secrets.

The Telecommunications (Lawful Business Practice) (Interception of Communications) Regulations 2000

The Telecommunications (Lawful Business Practice) (Interception of Communications) Regulations 2000 (known as “the Regulations”) provided that an employer retains the right to carry out monitoring despite the fact the employee has not given his or her express consent, if such monitoring is required to carry out the following:

- Recording evidence of business transactions
- Ensuring compliance with regulatory or self-regulatory guidelines
- Maintaining the effective operation of the employer’s systems (for example, preventing viruses)
- Monitoring standards of training and service
- Preventing or detecting criminal activity
- Preventing the unauthorized use of the computer or telephone system

In short, the Regulations underlined some of the precepts of the RIP. Nonetheless, to follow the principle of “prudent man” and sheer common decency, warn people of these rules in your security policy.

The Freedom of Information Act 2000

The Freedom of Information Act 2000 will be brought into force in two parts, with full implementation by January 2005. It gives private individuals the right to access information held by public authorities, including:

- Central government
- Local authorities
- NHS
- Schools
- Police

Audit Investigation and Community Enterprise Act 2005

The Audit Investigation and Community Enterprise Act 2005 reinforces powers already in place from the companies act. This law makes a director responsible for giving accurate information to auditors, liable for prosecution for withholding relevant information of which the auditor is unaware, and signing off audit reports attesting that fact. This responsibility takes the form of a statement in the director's report to the effect that there is no relevant information that has not been disclosed to the auditors.

Should an inspector discover that information has been withheld, the directors will be liable to imprisonment and/or a fine.

The act also contains a whistleblower protection clause that excludes liability for breach of confidence for those who provide information to authorities.

Official Secrets Act

If you work in the government, you should know the Official Secrets Act backward and forward. The original 1911 act was designed to stop all the post-World War I spying:

- (l) any person for any purpose prejudicial to the safety or interests of the State—

c. obtains, collects, records or publishes, or communicates to any other person any secret official codeword, or password, or any sketch, plan, model, article, or note or any other document or information which is calculated to be or might be or is intended to be directly or indirectly useful to an enemy; he shall be guilty of felony.

The Official Secrets Act 1989 places very particular constraints on government employees who fall under the act:

(1) A person who is or has been—
a member of the security or intelligence services; or
a person notified that he is subject to the provisions of this subsection,

is guilty of an offence if without lawful authority he discloses any information, document or other article relating to security or Intelligence which is or has been in his possession by virtue of his position as a member of any of those services or in the course of his work while the notification was in force.

(2) The reference in subsection (1) above to disclosing information relating to security or intelligence includes a reference to making any statement which purports to be a disclosure of such information or is intended to be taken by those to whom it is addressed as being such a disclosure.

(3) A person who is or has been a Crown servant or government contractor is guilty of an offence if without lawful authority he makes a damaging disclosure of any information, document or other article relating to security or intelligence which is or has been in his possession by virtue of his position as such but otherwise than as mentioned in subsection (1) above.

(4) For the purposes of subsection (3) above a disclosure is damaging if—

a. it causes damage to the work of, or of any part of, the security and intelligence services;

Or

b. It is of inflammation or a document or other article which is such that its unauthorized disclosure would be likely to cause such damage or which falls within a class or description of information, documents or articles the unauthorized disclosure of which would be likely to have that effect.

(5) It is a defence for a person charged with an offence under this section to prove that at the time of the alleged offence he did not know and had no reasonable cause to believe, that the information, document or article in question related to security or intelligence or, in the case of an offence under subsection (3), that the disclosure would be damaging within the meaning of that subsection.

Subsections 6-8 explain with notification from (1) above

(9) In this section security or intelligence means the work of, or in support of, the security or intelligence services or any part of them, and references to information relating to security or intelligence include references to information held or transmitted by those services or by persons in support of, or in any part of them.

And you can't say it fairer than that: Don't be a spy or an underhanded, unreporting type. If you work in this sector, keep STUMM (i.e., don't talk about it). In the U.K., we even protect the stupid with clause 5, which keeps gullible people out of jail.

U.S. Legislation

There is no reason that a security officer of a wholly owned U.K. or European firm should need to know U.S. law. However, after being relatively unpunished in the data area for many years, U.S. legislation has begun to set the

standard for information security legislation in a very direct and prescriptive way. This is in stark contrast to European law, which is very much more open to interpretation.

In particular, the following statute is groundbreaking because it embodies in legislation a penalty for the ethereal concept of reputational loss and makes it difficult to sweep security breaches under the carpet.

California SB 1386

I believe California SB 1386 is one of the profound pieces of legislation. Currently, it applies only to data of California residents, but apparently a federal version is in the pipeline. In short, this act makes reputational risk of poor security a reality because it requires public disclosure of any security breach that involves personal information if it is unencrypted or if it is reasonably believed that the information has been acquired by an unauthorized person.

In cases involving over 500,000 people, the organization can warn the potential victims *en masse* through a Web site and by alerting the media.

Sarbanes-Oxley 2002

At the beginning of the new century, a plethora of informal recommendations came down from the Securities and Exchange Commission (SEC) about auditor independence after a number of well-publicized cases of false reporting. With the full extent of the Enron case coming to light, the Sarbanes-Oxley Act was introduced.

As an instrument for accounting reform and investor protection, this legislation was intended to reestablish investor confidence. It also was intended to reduce the stranglehold that the Big Six accounting firms had on professional services in larger corporations. Unfortunately, the law resulted in so much process design work, the Big Six didn't notice any revenue loss.

Key sections of the act include Sections 201, 302, and 404.

Section 201

Relating to auditor independence, it is no longer allowed for your auditor to perform such activities as financial information systems design and implementation; internal audit outsourcing services; and legal services and expert services (including security).

Section 302

The CEOs and CFOs of the accounting company's clients must sign statements verifying the completeness and accuracy of financial reports.

Section 404

CEOs, CFOs, and auditors must report on and attest to the effectiveness of internal controls for financial reporting. This report shall:

- State the responsibility of management for establishing and maintaining an adequate internal control structure and procedures for financial reporting.
- Contain an assessment, as of the end of the most recent fiscal year of the issuer, of the effectiveness of the internal control structure and procedures of the issuer for financial reporting.
- Each registered public accounting firm that prepares or issues the audit report for the issuer shall attest to, and report on, the assessment made by the management of the issuer. An attestation made under this subsection shall be made in accordance with standards for attestation engagements issued or adopted by the Board. Any such attestation shall not be the subject of a separate engagement.

Gramm-Leach-Bliley Act (GLBA)

The objective of the Gramm-Leach-Bliley Act was to ease the transfer of financial information between institutions and banks while making the rights of the individual through security requirements more specific. Key points include:

- Protecting consumers' personal financial information held by financial institutions and their service providers
- The officers and directors of the financial institution shall be subject to, and personally liable for, a civil penalty of not more than \$10,000 for each violation

Although the penalty is small, it is easy to see how it could impact a bank.

Health Insurance Portability and Accountability Act (HIPAA)

The Health Insurance Portability and Accountability Act, universally known as HIPAA, deals with health personal data, which is defined as

1. An individual's past, present, or future physical or mental health or condition
2. An individual's provision of health care, or
3. Past, present, or future payment for provision of health care to an individual

The primary objective of the Security Rule is to protect the confidentiality, integrity, and availability of data when it is managed (i.e., stored, maintained, or transmitted) by a health care provider.

Health care providers must provide notice of privacy policies and procedures to patients, obtain consent and authorization for use of information, and tell how information is generally shared and how patients can access, inspect, copy, and amend their own medical records.

USA Patriot Act 2001

Introduced as a direct result of the events of September 11, 2001, the USA Patriot Act has had a huge impact on how government agencies could obtain information on private individuals. Particularly:

1. Wiretap orders now can be obtained pertaining to a person rather than individual circuits.
2. Internet service providers (ISPs) may volunteer information that they believe is of national importance, without fear of prosecution.
3. Mailbox information can be obtained by subpoena rather than wiretap order.

Summary

The purpose of this chapter is not to teach you how to practice law. It is to show you that careless information security policies can leave an organization open to civil or even criminal prosecution.

By studying U.S. legislation, we gain some insight we might need in conducting e-commerce with U.S. organizations; we also get a very real insight into legislation that will be coming to Europe.

The brief abstracts of U.K. law clearly demonstrate that our actions to protect data and security policies that drive them *must take into account the word of law*. As a CISO, you will be next in line to the CEO to be held accountable should an issue occur. The areas that warrant most attention are:

- Storage of personal data
- Monitoring of transmissions and systems
- Activities to test systems

All U.K. legislation discussed in this chapter is available online from the U.K. Cabinet Office (www.opsi.gov.uk/legislation/about_legislation.htm).

Information Security Standards and Audits

The purpose of this chapter is to:

- Provide a brief overview of information security standards
- Provide a brief overview of various types of security audits

Anecdote

I don't dislike auditors, but as a profession it does seem to attract herds of the wrong kind of people—all cufflinks and unsupported arrogance. I should know; I worked with them for long enough.

At the time I was well on the way to becoming a partner in one of these audit firms, mainly because I kept being engaged in very large security assignments. In Hong Kong, I was doing a job for a world-class bank. Everything was decidedly “barely adequate” in terms of firewalls, but I had yet to look at the routers and switches. In fact, the truth was, the firewalls were so bad that I was looking for good news on the routers to soften the blow. As a courtesy, I, the fledgling partner, was taken to meet the IT auditors (who had recently completed a review of the e-commerce firewalls and routers); the rest of the team had to stay and work. All the way up the stairs to the meeting, the senior partner told me how wonderful this team was, so when I arrived I was not surprised to see them perfectly dressed and well spoken.

Halfway through our meeting, I mentioned I had yet to do the routers and I immediately was reassured by the senior auditor, “I checked the routers personally, no problems there.” He even waxed lyrical about the special processes they had for adding access lists to them each time a new server was added.

With some difficulty, I got the router listing. On the last page I noted a genuine but deadly mistake: VTY 5- 64 had been defined with no access lists and a default password. For nonrouter people, that means everybody on the Internet could just log into them. Tilt! These routers had been reviewed by experts! I dreaded to think what joys the access lists would hold.

I wasn't wondering for long. Line after line of ACCESS-LIST 101 PERMIT IP ANY HOST-ADDRESS. Every time they added a host into the DMZ, they added a new line with a new host address. This resulted in a unique access list designed to ensure that hackers only attacked hosts that really existed, only using the IP protocols, but of course, the list made sure no port was restricted. After all, we would not want the black hats to waste any time scanning for nonexistent servers!

And the routers had passed the audit test. The checklist asked for access lists and they were there—big tick. And it was about that time I started to feel my partnership slipping away from me. I should never have said that senior auditor should save a few pounds by not buying handmade shirts and spend the money on a book about routers; my truly unreasonable character was coming to the fore.

But there are a couple of morals to the story:

- *The bank's security standard only required an "access list." It didn't mention restricting traffic at all.*
- *The auditor was told to audit only to the bank's standard, not to rock the boat, not to think for himself.*

So when viewing an audit report or a certification based on an audit, it is imperative that you understand the basis of the audit. Was the audit a compliance check of a particular standard? Was it against good practice, and who did it? What were his or her objectives?

Introduction

At one time, I truly believed that doing a job well was all that was needed, and in some cases, that's still true. But confidence and independence of thought are rare in this world. Pretty soon, even if you establish yourself as a well-known thinker, someone will ask for an external reference, verification, or comparison. Your word is not enough for them—often due to insecurity, often due to an act of belittlement—but you will be asked, *Who says so?*

This chapter covers the major standards, audits, and certifications in the industry so that if external verification is required, you know what to expect.

In the case of audits, they can have a real bearing on you and the perception of your security team within an organization. Preparing poorly or leaving others unprepared can leave a bad perception in the auditors' minds that will be transmitted to management. All you need is some opinionated server administrator to be interviewed and you can find yourself explaining a bad report to the CEO.

So, in short, this chapter is not going to teach you how to conduct an SAS70 audit and become BS 7799 certified in 20 pages. This chapter *will* explain what to expect if an auditor or consultant suddenly turns up and starts performing such an exercise.

BS 7799 and ISO 17799

BS 7799 is considered by most authorities in Europe to be the major information security standard. It was derived out of commercial need and has

grown to be a small industry in its own right. I have never been an avid supporter of the standard, but I can say that I've watch it grow and personally know many people who shaped it to be the foremost standard it is today.

A Canned History of BS 7799

In 1995, I sat in a conference hall just off Fleet Street in London, listening to Richard Hackworth, the chief information security officer of HSBC, talk about the future of the new *Information Security Good Practice Guide*, how it came about, and his involvement in producing it. At the time, there was great feeling that something significant had happened, but nobody really was sure what.

Hackworth explained that in the early 1990s, some leading companies suffered very badly from the lack of quality of IT systems. The example he gave was Marks & Spencer (M&S), a U.K. department store, which really embraced information technology but that sourced much of its woolly jumpers from small firms in Yorkshire via systems that were fragile, at best—usually with no backup. Failures and outages at these firms were reputed to be hampering smooth operation, and let's face it, M&S is without a doubt one of the U.K.'s most important institutions. Pants, brilliant sandwiches, affordable-quality suites for the junior staff, and more sandwiches (with a nice Chianti, ahhh!)—clearly, disruption of this bounty is unthinkable. An educational process and a benchmark were needed to encourage firms to maintain IT at a required standard.

So in 1991 the U.K. Department of Trade and Industry established a Working Group comprising experienced information security managers from the banks, M&S, and other leading institutions. From this group the Information Security Management Code of Practice was produced.

Published by the British Standards Institution (BSI) in September 1992 as an Industry Code of Practice, the Code provided a framework for an organization to examine and improve the security of its IT systems environment. In 1995 the working committee edited the Code so that it was suitable for publication as BS 7799, the U.K. standard. The effect was not prolific, but it was unsettling. Banks firmly and confidently announced that on a bad day they exceeded every aspect of it. However, the majority of the firms reeled at the concept. Ten little clip-art keys on a pamphlet got passed around many boardrooms; no longer could board members claim that it was all technical mumbo-

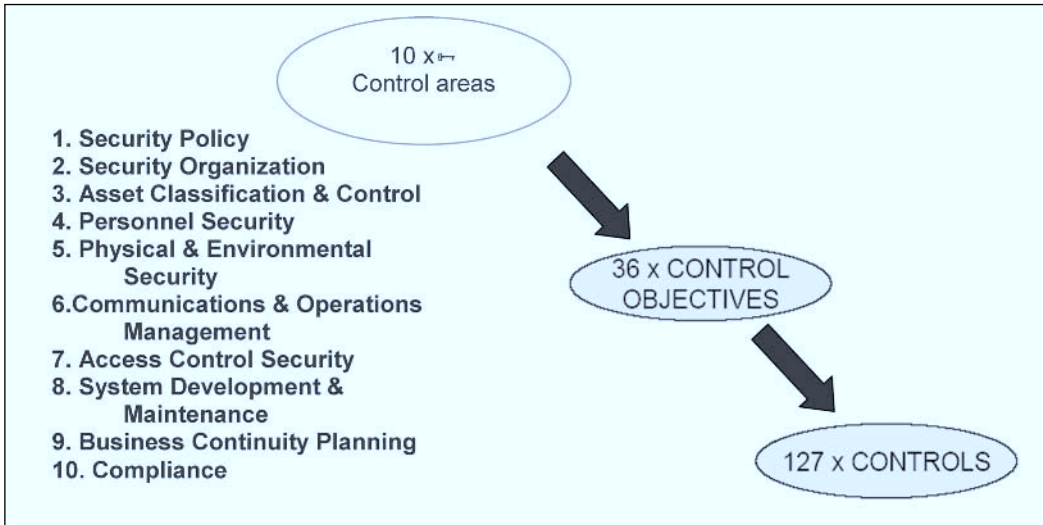
jumbo, because it was backed by the U.K. Department of Trade and Industry (DTI) and the BSI and was presented so even an idiot could understand it. If managers ignored it, it was because they *chose* to ignore it as a risk judgment. Virus-scanning software and tape backup systems sales rose dramatically.

The key controls (as of 1995) are as follows (Information Security Policy Document [§1.1.1]):

- Allocation of information security responsibilities (§2.1.3)
- Information security education and training (§4.1.2)
- Reporting of security incidents (§4.3.1)
- Virus controls (§6.3.1)
- Business continuity planning process (§9.1.1)
- Control of proprietary software copying (§10.1.1)
- Safeguarding of organizational records (§10.1.2)
- Data protection (§10.1.3)
- Compliance with security policy (§10.2.1)

There you have it—a simple list of minimal control areas (Figure 5.1) that *every* organization using computers should have, all implementable, sensible, and testable. Every accountancy and consultancy firm started a BS 7799 compliance audit service.

The industry called for a means of certifying against the Code. A new steering committee was formed, comprising the U.K. Accreditation Service (UKAS) and International Register of Certified Auditors (IRCA). By 1999, with the release of the second edition of the code (BS 7799-1:1999), the heyday of BS 7799 had been reached; new controls had been added to account for new technology and you could be fully certified by Lloyd's as compliant.

Figure 5.1 Ten Key Control Areas and 127 Control Objectives within BS 7799

History of BS 7799, Part 2

BS 7799, Part 2, was the beginning for some; for others, it was the end. It mandated the establishment of an information security management system (ISMS). It turned BS 7799 from a baseline standard to the auditor’s much-loved “risk-based approach.” It specified controls to be implemented according to security, legal, and business requirements—in other words, you decide what security you need based on your risk assessment.

The most recognized standard was republished on September 5, 2002, with BS 7799-2:2002. This included the harmonization with ISO 9001 and, most important, in words of one of the press releases, “used the prescriptive ‘shall’ statement”—a meteoric step forward.

The intention of this standard was to:

- Harmonize with other quality management systems (such as ISO 9001 or BS 15000)
- Facilitate integrated and better auditing of this standard
- Incorporate the concept of Kaizen: continual improvement by the Plan, Do, Check, Act (PDCA) model

- Acknowledge the need for corporate governance and information assurance
- Implement Organization for Economic Cooperation and Development (OECD) guidelines

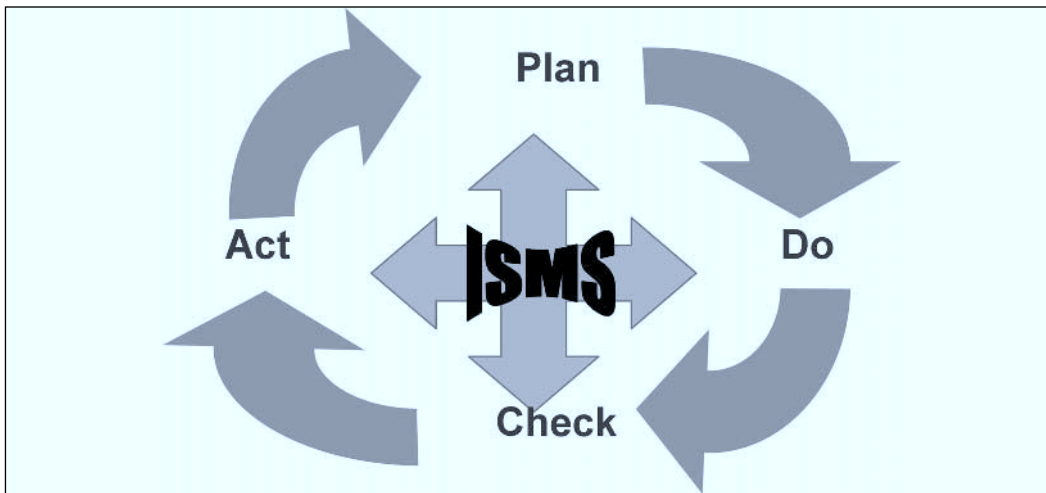
Particularly, the new standard:

- Introduced the PDCA model
- Included a process-based approach around PDCA
- Clarified the relationship among risk assessment, selection of controls, and the Statement of Applicability (SOA)

PDCA

The PDCA phased approach is shown in Figure 5.2.

Figure 5.2 The PDCA Phased Approach to IT Security



The Plan Phase

The plan phase is the most exhaustive stage of the process cycle. However, it is a traditional asset identification, risk analysis, and countermeasure design process that most IS professionals will be aware of. It consists of the following stages:

- **Define the initial scope of the ISMS** This is a business-based decision. Some real scopes are shown in Table 5.3.
- **Define the ISMS policy** This is the security policy plus an ISMS manual in practice. However, you can't really finalize this until you have completed the following analysis.
- **Identify assets** Most companies have different assets. Many will have information assets that are common. Exhaustive lists are available to help you quickly identify the easy ones, thus leaving you time to get to the difficult ones. In the most simple example, you can record this in a simple spreadsheet, as shown in Table 5.1.

Table 5.1 Asset List

Information Asset Name	Description	Owner
HR database	Database containing all employee records and salary details	F Smith HR director
Customer database	Database containing all customer details	J Smith Sales director
Configuration folders	Folders containing all device names and configurations	I.C. Noships IT manager
Insurance files	Insurance files	Doris Frank Admin

- **Identify threats** Various common sources (inferred from BS 7799 example controls, CRAMM, etc.). In practice this tends to be more of a cognitive step. At this time you can also start to build a controls register.
- **Do a risk assessment** In BS 7799, this is known as a *business impact analysis*. Typically, it involves importing your asset register into a spreadsheet and, for each asset, assessing the effect on the business resulting from a loss of CIA. Table 5.2, which is a typical example, shows an arbitrary value between 1 (low) and 5 (high) assigned to each asset based on confidentiality, integrity, and availability.

Table 5.2 Business Impact Assessment

Information Asset Name	Confidentiality		Integrity		Availability (Outage)		
	Internal	External	Internal	External	2 hrs	1 day	1 wk
HR database		5	3	3	3	3	5
Customer database	3	5	3	3	3	3	3

This example shows that we need to focus our controls to protect confidentiality of both the HR and Customer databases. We should also ensure that we have recovery controls that can restore the HR database within a week.

- **Select controls** This is called the *risk treatment plan*. It is a basic list of controls that you use to counter threats. Generally, these controls can be referenced by the control number at the back of the standard. Additional controls can be added to suit your environment.
- **Complete the Statement of Applicability (SOA)** This will include a scope, as shown in Table 5.3, as well as a list of controls that have been deemed not applicable. A common example is system development. If you are a small company that only uses prepackaged software, this step might not apply to you.

Table 5.3 Example Statements of Applicability as Published by the Certifying Body

Company	Statement of Applicability
TelCO	The Information Security Management System in relation to Data Center Services at Tokyo Internet Solution Center. This is in accordance with Statement of Applicability Issue Ver. 2, February 17, 2003.

Continued

Table 5.3 continued Example Statements of Applicability as Published by the Certifying Body

Company	Statement of Applicability
Web designer	The management of information security system in all activities relating to security application services, including Managed Security Service (MMS), consulting, and solutions. This is in accordance with the Statement of Applicability Issue 2.0.
TelCO	The Information Security Management system supporting the provision of customer-selected Web hosting/housing services from U.K. ISCs. This is in accordance with the Statement of Applicability.
MSS provider	Remote Managed Services operated out of the Network Operating Center (NOC) provide the following services to clients: network fault monitoring, network fault management, network performance management, firewall management, remote access authentication management, security management, and documentation management. Key NOC systems and associated hardware are managed and supported by engineers at the registered address. Statement of Applicability SOA_2 dated July 1, 2002.

You now have a risk treatment plan and an ISMS manual to implement. We now enter the Do phase.

The Do Phase

Although implementing this whole ISMS will probably take at least six months for even one of the smallest scope, the standard really is brief in the remaining stages. Remember, this is a quality management system, not a technical security blueprint. This stage includes:

- **Finalize and fine-tune your risk treatment plan** In theory, you should not have to do this, but in practice when you sit down and do a detailed technical design of the controls, reality may have to take over.

■ **Implement this risk treatment plan and associated controls**

Be sure that the technical implementation will support the ISMS. This means, for example, having manuals to cover virus maintenance and logs or records to show you are following the virus maintenance process. A BS 7799 auditor will not check the setting on your firewall, but he will look at an audit trail of changes, if you say you have one. Remember, its effectiveness has to be measurable, which brings us nicely to the next stage.

The Check Phase

The Check phase consists of these tasks:

- Execute monitoring.
- Undertake regular reviews of the efficiency and effectiveness of the ISMS.
- Monitor the acceptable risk.
- Conduct regular audits of the ISMS.

The ISMS is supposed to be a management system. As we learned in the Chapter 1, management systems must consist of both monitoring and control. You will be expected to have monitoring facilities in place so that the efficiency of the ISMS can be checked. This includes saving a log of who entered the computer room and periodically signing it to show that a review process has been conducted.

Audits should be conducted to ensure that these monitoring processes are being carried out.

The Act Phase

From the audits and operational failures that have occurred above:

- Implement any identified improvements.
- Take appropriate corrective actions.
- Communicate results to affected parties.
- Ensure improvements meet their objectives.

To complete the cycle, at six-month periods the ISMS should be changed to take account of any failing identified by the audits. In this way BS 7799 ensures that the management system remains in touch with current practices and the countermeasures remain effective.

The cycle begins again.

ISO/IEC 27001:2005: What Now for BS 7799?

Yet another overhaul, BS 7799-2:2002 (the British Standard for Information Security Management Systems), has been updated and was released on October 15, 2005, as an international standard, ISO/IEC 27001:2005. This means that the standard is recognized internationally.

It is fundamentally the same standard as BS 7799, with the following changes:

- Justifications must be specified for any exclusions in the scope
- A clearly defined risk analysis approach
- Countermeasures selected in the risk treatment must be linked back to the risk assessment (I don't know how you could have been certified without this commonsense step)

To conclude, the ISO 27001/BS 7799 is the only truly internationally recognized standard for security management. It is comprehensive in nature and a true risk-based approach—meaning you have to assess your own environment and take responsibility for the controls you implement. It does not prescribe specific controls to be implemented by rote; it assumes you have the skills to do this. Indeed, it suggests in the guidance notes that if you are struggling in this area, you should employ third-party help who knows security.

NOTE

Bottom line: If you want to work in information security, you should gain a good understanding of this standard.

PAS 56

PAS 56 was published in 2003 by the British Standards Institution and will be the U.K.'s first publicly recognized standard for business continuity management.

What Is PAS 56?

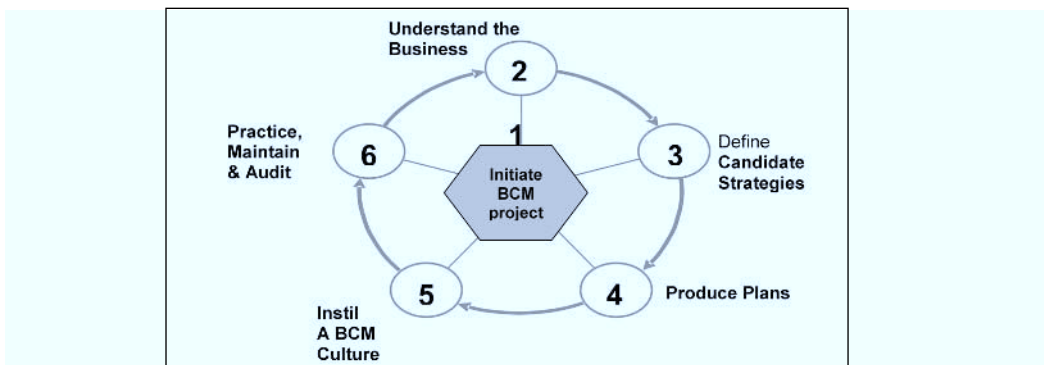
Organizations can use PAS 56 to improve their performance in business continuity management (BCM), whether they're starting afresh or rehashing existing plans. PAS 56 provides an approach for managing a BCM program and evaluation criteria to assess the overall readiness of an organization.

PAS 56 isolates a number of idealistic success factors that influence BCM effectiveness in a target organization. These include:

- BCM should be driven from the top of the organization and sponsored by the board or the CEO.
- A director-level responsibility/accountability should be assigned for the effectiveness of the organization's BCM competence and capability.
- BCM must be supported by cross-business representatives.

PAS 56 is built around the BCM life cycle, a process model intended to underpin all business continuity activity of an organization (see Figure 5.3). At every stage of this cycle, the standard outlines its component tasks, the object of that exercise, and expected outputs.

Figure 5.3 The BCM Life Cycle



The Stages of the BCM Life Cycle

PAS 56 describes six stages of the BCM life cycle.

Stage 1: Initiate the BCM Project

From the very start, all BCM activities should be run as a formal program. This program should have:

- An agreed upon and signed scope to cover all mission-critical activities with the highest level of sign-off.
- Documented and agreed responsibilities
- BCM policy, principles, strategy, and standards
- Annual BCM review
- Dedicated BCM budget

Stage 2: Understand the Business

All security projects begin with an analysis phase, and PAS 56 is no different. This stage begins with a business impact analysis (BIA) and risk assessment. At the end of this process, we will know:

- Mission-critical activities (MCAs)
- Dependencies for the MCAs
- Single points of failure of the MCAs
- Influences that may impact MCAs
- Level of business continuity (LBC)

Stage 3: Define BCM Strategies

We covered this ground in a previous chapter, but a BCM strategy includes:

- **Formally accept the risk** Where the cost of fixing or transferring the risk makes this danger a “fact of life.”

- **Transfer the risk** Typical examples include insurance or outsourcing to partners who are better able to manage the risk.
- **Mitigate the risk** Reduce the likelihood or reduce the impact, or both.

Stage 4: Produce a BCM Plan

The standard, as you would expect, is a little thin on detail here. But for each mission-critical activity, you need to develop a recovery solution. This solution includes

- Continuity for various scenarios, considering the recovery time objectives (RTOs) and recovery point objectives (RPOs) defined in the BIA
- Invocation, definitions, and authorizations
- Procedures and processes to perform the recovery tasks and activities
- Definition of team structures, leaders, and deputies
- Call-out trees

Stage 5: Instill a BCM Culture

This stage involves education, awareness training, and participation!

Stage 6: Practice, Maintain, and Audit

Tests should:

- Demonstrate competence and capability for business continuity and crisis management; where business continuity protects the operation while crisis management protects image and reputation.
- Document the results.
- Provide feedback into the planning process for updating and maintenance purposes.

NOTE

It has been long claimed that PAS 56 will follow the path of BS 7799 by first becoming a BS standard and then an ISO standard. However, as yet no real progress has been made along that route.

FIPS 140-2

Federal Information Processing Standard 140-2 (FIPS 140-2) is a standard that describes requirements for unclassified data encryption. The standard was published by the U.S. National Institute of Standards and Technology (NIST) and has been adopted as a de facto standard.

The standard defines four levels of security, from Level 1 (lowest) to Level 4 (highest). These levels cover the basic design and documentation, module interfaces, authorized roles and services, physical security, software security, operating system security, key management, cryptographic algorithms, electromagnetic interference/electromagnetic compatibility (EMI/EMC), and self-testing.

Should I Bother with FIPS 140-2?

If you are going to bother using a hardware security module (HSM), you might as well get a good one. But all you have to do is look on the side of the tin to see what level your device is cleared to.

What Are the Levels?

The standards provide four different levels of security:

- **Level 1** No physical security mechanisms are required in the module beyond the requirement for production-grade equipment. It's very hard to see why anyone would use an HSM for this lowest level of security.
- **Level 2** Tamper-evident physical access controls with role-based authentication.

- **Level 3** Tamper-resistant physical security. Level 3 provides for identity-based authentication.
- **Level 4** Physical security provides an envelope of protection around the cryptographic module. In practice, this means some level of physical separation.

Common Criteria Certification

Common Criteria (CC) allow organizations, usually product vendors, to demonstrate conformance of a product to its documented specification. The process is arduous and complex.

The manufacturer, known in CC parlance as a *sponsor*, doesn't do this testing. The evaluation is carried out by a Commercial Licensed Evaluation Facility (CLEF), a third-party organization that reports on the conformance to a set of security claims made about the product. This report is submitted to the certifier for approval. If the certifier is satisfied, a certification report is produced and a CC certificate is awarded.

Other CC Jargon

You should also be familiar with the following terms related to CC: security target, protection profile, and evaluation assurance level (EAL).

The Security Target

The security target document is the security spec. It describes the security functionality the product offers. It may reference a protection profile.

Protection Profile

The protection profile is effectively a security use case and should include an EAL.

Evaluation Assurance Level

An EAL defines the rigor that must be applied to the development and presentation of the evaluation. The levels are:

- **EAL0** Failed, no badge today
- **EAL1** Functionally tested
- **EAL2** Structurally tested
- **EAL3** Methodically tested and checked
- **EAL4** Methodically designed, tested, and reviewed
- **EAL5** Semiformally designed and tested
- **EAL6** Semiformally verified design and tested
- **EAL7** Formally verified design and tested

Types of Audit

As a security officer, you will come across many auditors who want to inspect your security. It is important that you understand their role and objectives.

Computer Audit as Part of the Financial Audit

If you work for a public listed company—in other words, a company that has shares on the stock market—the company’s act (or local equivalent) requires you to appoint an independent auditor to make sure that the shareholders’ money isn’t being spent on frills like swimming pools (unless you are swimming pool vendor). These auditors will produce an annual report with a “true and fair” financial statement; if anything goes wrong, blame them.

Originally, as recently as the early 1980s, audits involved counting money and stock items, but in a multinational world this became impossible. So auditors came up with a “system-based approach” whereby they use the finance systems of your company to produce the financial statement. This involves placing *reliance* on a system of internal controls that surround the company’s finance systems. Some companies have good internal controls, and other have bad—and if a company has bad controls, anyone might type any number into the accounts payable system. This would lead to a misrepresentation of the financial health of the company and yet another accounting firm would disappear under the burden of a large lawsuit.

The idea is the better the control, the more reliance the auditor can place on the company system. This means that the auditor has to do less *substantive* audit testing (where the auditor actually tests the validity of the number in the system). To make this value judgment, the auditor will call a *computer auditor*, who will check IT controls. Computer auditors vary, from accountants with no real computer experience who have been on a few courses to experts from the security practice trying to keep their utilization up on their timesheet. Mainly, they are the former.

Computer auditors will check, at the briefest level, details very similar to ISO 17799 (where do you think the standard came from?). This covers, for example, access controls, leaver-joiner processes, segregation of duties, change control, training, backup, and disaster recovery/business continuity planning (DR/BCP). There will be some testing (they will almost certainly look at a sample UNIX password file and W2003 Group policy password controls), but most will be interview based.

At the end, they will produce a list of *management letter points* (MLPs) that you have agreed to regarding factual accuracy. You must make sure you do this and ensure that you have a proper closeout meeting. The MLPs are purely informative; only the gravest control failing works its way into the annual report. However, it is very important that you take this opportunity to raise the priority of the points that you are championing. Although the firm is under no obligation to implement the suggestions, there is a good chance it will so that this process can make budgets available. However, after doing hundreds of audits, I saw many ridiculous MLPs that had not been challenged and caused companies to spend money needlessly. If you find something wrong, make the auditor remove it.

Section 39 Banking Audit

The Section 39 banking audit is driven by the regulations in the Financial Service Authority (FSA) Prudential handbook, but because these are “business regulations, not computer regulations,” you can expect very similar activity during a banking regulatory audit, with a few exceptions. An accounting firm will be appointed by the FSA; typically this will be your normal auditor (the reporting accountant). Your company pays for the audit, which will cover a scope agreed by the FSA in a formal scope letter. The big difference is that

the findings are not really advisory; the auditor presents his or her report at a trilateral meeting (where the FSA, the auditor, and your company are represented), and the actions that are agreed in that meeting are binding. If you don't take those actions, you could lose your banking license.

SAS 70

Statement on Auditing Standards (SAS) No. 70, Service Organizations, is an internationally recognized auditing standard developed by the American Institute of Certified Public Accountants (AICPA).

SAS 70 provides a standard format and approach for:

- The directors of a service organization to disclose their control activities and processes to their customers
- An independent auditor to issue an opinion on a service organization's description of controls

Effectively, it is only a method of reporting, because the SAS 70 does not have any baseline or standard control activities that must be in operation. No evidence of risk assessment, penetration testing, or impact analysis needs to have been undertaken by the auditee. The only extent of prescribed rigor is that all controls declared in the report need to have been in operation for a period of over six months. Correspondingly, any SAS 70 project always begins with a hefty “controls objective selection exercise”—literally, a game of “hunt the risk management.” The results will include a mix of obvious best practices (to have glaring omissions would be silly) and a load of controls objectives derived from those countermeasures that the organization knows it has in place.

There are two types of SAS 70s: an SAS 70 Type I report and the more advanced SAS 70 Type II report. Typically, a company will do Type I and then move onto Type II.

In the Type I report, the service auditor will:

- Give an opinion on whether the controls (as described by the organization) were in operation at the time.
- Give an opinion on ability of controls in operation to meet the stated control objectives.

A Type II report includes the aforementioned steps for Type I plus a statement that the controls were tested and were operating with sufficient effectiveness to provide reasonable assurance.

The SAS 70 report was designed so that service providers need not be continually audited; instead, they would have only one SAS 70 audit. Financial auditors (“end-user auditors”) auditing the financial statements of a company using the services of the provider (the “user organization”) could place reliance on the report.

Service organizations that provide such services could be application service providers, credit card processors, claims processing centers, or other data processing service bureaus. SAS 70 has been widely used to meet the SBO Act Section 404—if an organization outsources processing to a third party, that organization can take reliance from an SAS 70 report. However, there has been severe criticism of organizations relying on blanket control objectives that aren’t sympathetic to those of the user organization.

As quoted directly from the manual, SAS No. 70, Service Organizations, is an internationally recognized auditing standard. This is the U.S. definition of international (in other words, recognized from East to West Coast, including Canada). While I was clawing my way up the slippery pole to become a partner in what is probably U.K.’s most respected accountancy firm, I was involved in nearly all the SAS 70 projects, which numbered four. Because of the time it takes and because it requires an American accounting partner to sign the report, SAS 70 is the province of the Big Four Accountancy Firms and is terrifically expensive.

Other Types of Audits

Some other types of audits are described in Table 5.4. These are less common but are included for completeness.

Table 5.4 Other Types of Audits

Name	Description
FIT 1/94	Testing of a set of actual controls designed to satisfy a set of control objectives. The control objectives and description of actual controls are provided by the company directors; in other words, we assess whether the controls are appropriate for the environment, to assure third parties that the controls in place meet the control objectives.
FRAG 21	Pretty much as above but for pension trustees.
SysTrust	Another international standard issued mainly from the United States by the AICPA. This follows on from the group's blinding success of Webtrust certification, which sold single figures in Europe and little better in the States. SysTrust uses four principles to evaluate whether a system is reliable: <ul style="list-style-type: none"> ■ Availability ■ Security (physical and logical access controls) ■ Integrity ■ Maintainability This time they have prepublished control objectives as the basis of the audit, but I suspect nobody cares.
VISA CISP	VISA Cardholder Information Security Program; provides a standard for protecting sensitive information, including 12 basic security requirements with which VISA payment system users must comply. Includes a penetration test.

Tips for Managing Audits

Many people manage audits badly. Don't. Here are some tips for getting through them successfully:

1. **Never lie** Lying can constitute fraud. Always encourage the auditor to see your side of an issue.
2. **Engage the auditor** Ask for a scope and a work plan so that you can clear a path for the auditor—and know exactly what to expect. Ask if the auditor needs to run special software or requires special user accounts. Make sure you have plenty of time to prepare.

Cleaning up beforehand is expected. Not to do any of these things is *unprofessional*.

3. **Help the auditor** Arrange a schedule for the auditor that's filled with people you want him or her to see and not the converse. Book the meetings for the auditor.
4. **Audit evidence** Never produce audit evidence for the auditor unless you're asked. Ask him or her what things you can prepare in advance. Allow the auditor to list his or her own directories or configs, even if you have to type the commands yourself. Let the auditor see the results come directly from the system.
5. **Keep a copy** Keep a copy of everything you give the auditor.
6. **Audit room** Get the auditor an office—one out of the main area of business.
7. **Clearance** Ask to clear the factual accuracy of all points.
8. **Never obstruct** Claiming that information is privileged and unavailable to a regulatory auditor because it is “intellectual property” is asking for trouble.
9. **Review the draft report** If it's wrong, it's wrong—get it changed.

Nothing can stop an auditor who is hell-bent on giving you a bad report, but this should prevent the common slipups.

Summary

This chapter provided an overview of key standards related to the management of information security:

- ISO 27001/BS 7799 is the primary standard covering information security management.
- PAS 56 covers business continuity and is predicted to be an ISO standard in the future.
- FIPS 140 is a U.S. standard covering the storage of crypto material that is used by banks and military organizations for unclassified keying systems around the world.
- Common Criteria is the emergent standard for assessing security equipment.

Any security manager or CISO should be familiar with all these standards.

Audits present a different challenge. Unless you are trained, much of the information detailed in this chapter will be new to you. Yet it is highly likely that the auditor will be assessing you and your team's effectiveness and reporting this information to your CEO. Doesn't it make sense to understand what the auditor will be looking for and what process he or she will use? For the major audit types, this chapter provides the information you need.

Interviews, Bosses, and Staff

The purpose of this chapter is to:

- Give you a bit of a laugh
- Show you how *not* to behave at interviews

Anecdote

For years, I documented some of my more memorable interviews on www.loud-fat-bloke.co.uk, along with a wealth of thought-provoking technical papers. I was horrified to find that when I told colleagues that I was writing a book, most showed only polite interest on the technical aspects, but they all insisted I publish the interview section. They seem to think my misfortunes are funny—and I hope you do, too.

Introduction

This chapter is designed as a bit of light relief.

When I gave up heading the security practice, partnership, and annual riches beyond my dreams at a large accountancy firm (and ran away with a very large check), it was the middle of a recession. Although I slipped a notch in status, I fell on my feet, but I was no longer in a position to help many of the talented people I had always sponsored previously. Many of these people were now struggling a little in a post-9/11 world. To lend moral support, I started recording some of the interview encounters I had experienced. These became a feature of my Web site and a regular topic of discussion.

As I have worked with a “right bunch of nutters,” I have expanded this material to include bosses and staff.

Interviews as the Interviewee

It is imperative that any young, ambitious person develop a robust interview technique. If you can bite your tongue for a couple of one-hour interviews, you will receive a job offer, and then you can consider at your own leisure whether to take the job or not. If you decide that the guy interviewing you is a moron, transmit this information to him *after* he has admitted to you that you are the best thing since sliced bread by giving you an offer.

Of course, that’s not they way I do it.

Interview 1

I was asked to interview for the position of divisional head of security at a telecom company. The interview took place one Tuesday morning and was performed by a very stern lady.

Interviewer: You seem to think this job has a lot to do with IDS. Why is that?

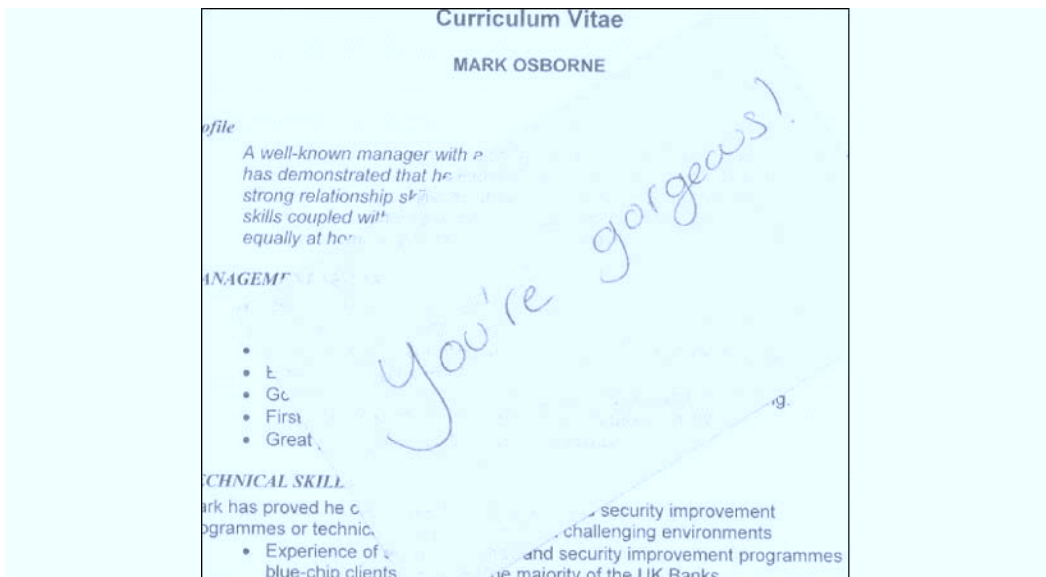
Fat bloke: Well, you have had one of your staff asking me questions about IDS, Snort, and ISS for about half an hour, the job description you handed me at the start has “knowledge of tuning IDS” listed as the primary skill, and the headhunter told me that expertise was essential.

Interviewer: What do you know about . . . (*Dring dring dring dring dring. Fire alarm rings!*)

Fat bloke: Who?

Interviewer: Don't you even know whom you are interviewing with? Surely, you must know the name?

Fat bloke: Yes; I was just deafened by the fire bell. Should I be worried by that loud ringing noise?



At this point the interviewer hands me a yellow Post-It; on it are the words, “You’re gorgeous!”

Fat bloke (I’m proving that the mouth is faster than the brain): “You’re just a chubby-chaser; aren’t you, love? I’ve had trouble with your type before.”

NOTE

The Post-It was a prop for a sexual harassment question, so my comment was particularly unhelpful.

Interviewer: The project team has just put in a new system. On Friday afternoon, the project manager walks in to tell you it’s business critical that it goes into production today, and to make it work, all traffic through the firewall needs to be allowed to all hosts with any-2-any rules. What do you do?

Fat bloke: I find this happens all the time. You have to get a whole-business view, so you do x , y , and z . Then you work the problem. But most important, even if it means building a completely new firewall for this new system, you should *not* endanger the whole of your infrastructure by turning off all your security on the say-so of one project manager.

Interviewer: Wrong. Everybody always gets this wrong. The project manager has told you it’s business-critical, so you should add the any-2-any rule! But don’t feel bad; none of the other candidates have got this right, either, and many of them were just as qualified as you.

Interview 2

I interviewed for a “security strategy” position at a large investment bank in Canary Wharf, London. The job seemed very much like a computer audit job and a little junior for me, but it was still worth a look. I had a telephone interview with the team manager, then a formal interview with the team

manager. Then all in the same afternoon a further set of interviews with the global head of department, global head of technical security, and some muppet with a charisma bypass (the last two over a video conference call, with a camera that seemed to focus on the bald bit on the top of my head; you all know I love myself, but they might as well have asked me to interview in the nude while doing a step-aerobics class).

Interviewer 1 (global head of department): How do you sell security to a chief dealer who doesn't think it matters?

Interviewer 2 (charismaless muppet): How do you sell security to a chief dealer who doesn't think it matters?

Interviewer 3 (global head of technical security, as he walks in to join the charisma-bypass muppet): How do you sell security to a chief dealer who doesn't think it matters?

Fat Bloke: “Well the answer I gave to that question when I was asked by Mr. X on first interview was this: It seemed to go down well so I'll give the same answer just for consistency . . .

** ** * * * * *

Interviewer (global head of technical security): “Which is better, AES or MD5?”

Fat Bloke: Well, they're different things, aren't they? One is a cryptographic algorithm and the other is a crypto-hash. I guess with poetic license, if you read “better” as longer key length, you could claim AES was better because it is common to have key lengths of 256. (*Trying desperately to demonstrate knowledge.*) Nevertheless, the question is similar to, “Which is better, a hammer or a screwdriver?”; you need to know for what purpose before you can answer it.

Interviewer (global head of technical security): “Yeah! But which is better?”

Interview 3

I once interviewed for a head of practice position at a boutique security consultancy. I was at the final stage when I met the CEO. Even though I knew I

had the job in the bag, they had been playing good cop/bad cop for about two hours, and I was mightily annoyed.

The CEO: Have you got any questions?

Fat Bloke: Yes, can you give an outline scope of the job?

The CEO: *Holding up a blank piece of paper* That is the scope of the job; you write it.

Fat Bloke: So anything in security is my baby.

The CEO: It is, if you want it.

Fat Bloke: So how many pen testers will I have to manage?

The CEO: Pen testers! You won't be in charge of pen testers—that's not within the scope of the job!

Interview 4

I was bored off my bean and needed a change. I was approached for the role of security supremo at a reseller well known for managed firewalls; we'll refer to them as Via-Windy. I was not keen, but then I met the VP of products, who was very impressive and ensconced in an impressive office near Oxford. I became more interested and went for an interview with the chief operating officer (in what seemed to be a small closet in the city).

Fat Bloke: The headhunter said you'd like to see me present a business plan. (*Thrusting a copy of a lovingly bound plan into his hand*)

The COO: (*Short man with Short Man Syndrome*) I couldn't care less about your plan. I am sick of hearing about security. I don't think Via-Windy can make money in security; Web hosting is where it's at.

Fat Bloke: Well, these days, customers expect commitment from their managed security partners. Too many firms have left their clients in the lurch. Likewise, investors are cagey; with 135 employees, three offices, and a small revenue of less than £20 million after five years of operation, they have some grounds. *But* last and most important, you have to question the motivations of someone who tells interview

candidates that it's a dead-end job. Yes I agree— I don't think YOU can make money in security.

* * * * *

Short COO: You may have been a director and uncrowned partner at a la-de-da Big Six firm, and you claim to have brought in £X million, *but we all know audit clients just hand you the money!*

Fat Bloke: That is a common misconception, but it was never really the case. Anyway, Sarbanes-Oxley has put an end to that for many international clients.

Short COO: Sarbanes-Oxley? What's that?

Fat Bloke: You did say you were considering being quoted on the NASDAQ??

Second Interviewer: *(A very strange northern woman)* I know not about security; sell me some security.

Fat Bloke: Would like to buy some security? No? *(Aimless chit-chat followed, during which time I got some insight into just how un-P.C. and intolerant some people can be. And you can guess it takes a lot to shock me. But I was, so I left.)*

Preinterview Questionnaires

Interviews are a poor way of selecting employees, but what real choice do we have? One company, the leader in the field for alternative site recovery centers, choose to use a basic questionnaire to help them choose new hires. Unfortunately, they got the day-release student from the HR department to write it, and English obviously was not a strength. The company was recruiting for an IT director who specialized in networks, security, and disaster recovery. It sounded like a great job, but I had a nice challenge where I was, and the questionnaire (see Figure 6.1) was too banal to resist. Even after I submitted this questionnaire, they were very eager to retain me for the position, but arrogant fool that I am, I just could not bring myself to go along.

Figure 6.1 Fat Bloke's Preinterview Questionnaire

Candidate Name:	Mark Osborne
Date completed:	29/09/05
Position applied for	Director IT— Europe
Key questions	
1. Bilingual - What languages and please rate fluency on a one (1) to five (5) scale (1 poor/weak and 5 being very strong)	
<i>French comprehension</i>	<i>- 2</i>
<i>French spoken</i>	<i>- 1</i>
2. What type of IT environments have you worked in? (i.e. structured - lots of standards and policies or unstructured with little or no policies/standards established)	
<i>Banking— Regulated and structured using large mainframe</i>	
<i>Banking— Regulated and structured using distributed PC and Unix</i>	
<i>Banking M&A— I want it yesterday</i>	
<i>Service provider— Varied</i>	
<i>Manufacturing— No regulation, emerging standards</i>	
<i>Start-up— No idea</i>	
~~~~~ <b>several more HR questions</b> ~~~~~	
10. - Have you ever reported to a manager/supervisor who was remote? If so, how fair a way is it? How did you feel about that?	
<i>My current boss is very remote and distant— Sometimes he talks to me while staring out at the river Thames below Canary Wharf.</i>	
<i>How Fair? Well, I think its reasonable— He has a lot on his mind, running a large company an all - And I am not a trainee.</i>	
<i>How do I feel?— I guess I feel hungry.</i>	
12. How do you select team members for an interview process?	
<i>I have a magic formula for this - When I select members of my team to participate in an interview process to recruit a new member of staff, I would routinely select:</i>	
<i>the potential employees manager</i>	
<i>a potential workmate of the candidate</i>	
<i>In this way from the manager, I get feedback on whether the candidate "fits" the culture and is manageable. The workmate gives an assessment of the candidate's ability to do the job.</i>	
<i>Revolutionary.</i>	
13. How do you handle challenges that are escalated to the executive level?	
<i>I consider myself to be "executive level". For example, in my current role, I have regular interaction with the CEO. And although it isn't GE, this is a \$250million p.a. revenue company.</i>	
<i>This role, I am told, is one level below the board and has 50 or more staff. So presumably it is also an executive level position.</i>	
<i>So "How do I handle challenges that are escalated to the executive level?"— Well exactly like I was dealing with a peer!!</i>	
15. How do you feel about international travel?	
<i>I feel International travel is brilliant for travelling to other countries or continents.</i>	
<i>International travel can be a pleasant experience— or not, usually dependent on the travel policy of the firm in question. So you tell me— How do you make it easy for me.</i>	



## Interviews as the Interviewer

But I am not the worst interviewee in the world. Here are some contenders for that title.

### Interview 1

When I was a principal security consultant, I was looking for a number of junior consultants. A bloke from Zergo, the leading consultancy of the day, had a great CV.

**Fat bloke:** So you know Sidewinder. Tell me about it.

**Interviewee:** *Silence*

**Fat bloke:** I also see you have used FireWall-1 and Gauntlet. Tell me what generic type or class of firewall each is. Perhaps highlight the pros and cons?

**Interviewee:** *Silence*

**Fat bloke:** Take your time; nod if you can hear me.

### Interview 2

In my current role, I was demonstrating what a good chap I was by helping recruit IP engineers. The following was possibly the worst start to an interview:

**Fat bloke:** I am standing in for the VP in charge of that section; he is on the board and he would be your boss's boss. I am Mark Osborne, chief information security officer.

**Interviewee:** Ha, ha, ha, security officer—what! Have you come to check me badge? Look, here it is! (*Holding up his visitor's pass*) Oh, but I'm not wearing it, look. Ha, ha, ha! You'll have me escorted out the building, ha, ha, ha!

*After 20 minutes*

**Fat bloke:** Look, I have let this run on much longer than normal because of your poor start, and I wanted to be more than fair. But we've asked you 15 questions and you haven't even come near to getting one right. You've applied for a job as an IP engineer and you've consistently demonstrated you don't know anything about routers and switches. It has been torture. When you write IOS 12.1 on a CV, people think you know IOS 12.1, not that you sat in the same office as someone that knew it!

**Interviewee:** I see myself as a manager.

**Fat bloke:** Then in future I suggest you apply for jobs with “manager” in the title.

## Bosses

Now let's discuss bosses.

### Runner-up for the Worst Boss in the World

This coveted award goes to a good-hearted and well-meaning fellow. Apart from his one defect, he had nothing really wrong with him. The main problem was this guy was visibly dirty and stank. He looked the spitting image of Neil from “The Young Ones”—the original hippie.

This guy was so dirty that the catering manager banned him from the canteen, declaring that he was “a health hazard.”

### Worst Boss in the World

*The winner* of the title worst boss in the world definitely goes to a partner I used to work with; let's call him Melvin Sheriff. The man was definitely a loony. He would get overly excited for no apparent reason and rub his hands together so rigorously that there was real risk of human combustion while he reeled off a whole series of self-rebukes.

Later in life, when we moved to more modern buildings with glass-fronted offices and meeting rooms, he took to telling himself off, or more correctly, telling off his reflection that appeared in the glass wall. This involved a furiously wagging finger, like some crazed orchestra conductor, pointing and

stabbing as the rant reached a crescendo. This was bad enough, but if you happened to be passing by or if you were a visiting client inside the meeting room, it could be more than distressing.

Classic incidents included:

- He turned up to work in one brown Oxford shoe and one black brogue and then went into a client meeting without changing.
- A finance director of an asset management firm said he never wanted to speak to Mel after he messed something up and caused the director serious job jeopardy. However, Mel insisted that he retain the role of client partner despite this. Apparently weeks of meetings ensued where Mel insisted on turning up. The finance director would completely ignore him and insist that everyone else do as well. If Mel spoke, the finance director started repeating “We can’t hear you, we can’t hear you” or began “La-la-la-ing” very loudly.
- We were partnering with RSA Security (very famous for encryption and token authentication) and holding a series of lavish marketing events that they had paid for. Mel walked up to Graham, head of sales for Western Europe, and said, “What do RSA do, then? I’ve never heard of them.”
- We were attempting to partner with Cisco (also very famous). Mel refused because “he didn’t want to be associated with a mail-order catalogue firm” (he was thinking of *Misco*).

Commonly, Mel would lose anything—briefcases, laptops, and phones all seemed to feel a desperate desire to escape his influence. At one stage, the IT department refused to issue him with any mobile phone except those had been marked for end of life. Why? Well, because he would leave them places. This trend was not limited to phones. It also included confidential reports. He would constantly write out yellow Post-Its to himself and attach them to reports. “Review this for me by tomorrow, Melvin,” the notes would demand.

Having an attention span of a newt, Mel would go and talk to someone, only to be spirited away to lunch. Lunch, oh lunch—ever calling, ever important for Mel. In an open plan office at least 20 feet from his desk, he would relieve himself of the burden of the weighty report he’d been carrying by

depositing it on some random and unsuspecting junior’s desk—instead of securing it in his office. Predictably, the junior would return, be unable to find Mr. Sheriff because he was down in the pub (with phone AWOL, of course). The ubiquitous junior, always keen to please, would work all night producing an analysis of investment bank regulations in Brazil, following the directives of the Post-It. I know—it happened to me, but once was enough.

Please don’t feel sorry for him—he wasn’t a nice man.

## Worst Employees

I never really had any truly bad employees. But I’ve had a few nutty or funny ones. Here are a few examples:

- The employee who called in sick one day because he thought he had rabies. Apparently, while in Thailand, he was stroking a stray cat that urinated on him. Apparently in Thailand, rats have rabies, and cats eat rats, ipso facto.
- The employee who called in sick because he had “exploding tooth syndrome”—apparently a real affliction, according to the Internet. However, this turned out to be a bit of fingernail stuck between his teeth.
- Then there was the employee who tried to run a mini-cab company from his mobile phone.
- Dave, who took six different doses of compassionate leave for his grandmother’s funerals. I want to know who kept on digging her up.
- Then there was Danny, the stunningly beautiful and violent security analyst who seemed to assault everyone, but nobody complained.

But they were all fantastic and sparkly.

## Summary

You should laugh as much as possible. You have to if you are going to survive in security. But enough levity. Now let’s dive into the more technical side of security.

# Chapter 7

## Infrastructure Security

The purpose of this chapter is to:

- Help you decide where to put those firewalls
- Help you decide how many of them to deploy
- Describe the other key components of infrastructure security that you need to make it look like you know what you're doing

## Anecdote

*I may be a violent, raging psycho—but I'm sensitive. And I always buy my wife a nice birthday present and something nice for Valentine's Day. My wife used to be an executive officer at the local council but gave it all up to teach kiddies arts and crafts (design and technology, they say—but whatever floats your boat!). When she was finishing her teaching course, we were watching a DIY program on TV where some bozo makes a £40 table out of £700 of wood, and she announces, "Buy me a router for Valentine's Day." I know she has no idea about money, so I didn't remark that the router would cost quite a bit more than the average card and chocolates.*

*At work, I recouped the losses by gaining full comic value from all the lads around. "Look at my wife's Valentine present!" I shouted as all the blokes pawed the JCB (nothing but the best) yellow monster while all the women flicked hair and tsch'ed to no effect. All the blokes wanted to be as Cool 'n Hard as me.*

*At home, I gave this beautiful router to She Who Must Be Obeyed—and got no response. Finally: "Oh," she said. In my best Tarzan voice, I said, "You want router, I get router." She screamed, "No! I read your latest research article—I want 802.11g, not 802.11b! I want 54Mbit connections!" Obviously, the light of my life wanted a new wireless broadband router. (I kept the one I got her—why not!)*

*The lesson here: When buying routers, proxies, and firewalls, you need to understand what they do and where you are going to put them, plus what they can plug into.*

## Introduction

When you are designing a hugely expensive and critical infrastructure, no matter how experienced you are, you will need to validate and cross-check your work. Well, you will if you are a thinking person. This chapter covers this less-than-common subject.

## Network Perimeter Security

The approach and techniques detailed here step through a series of decisions, both financial and risk based, to help you find a solid, cost-effective architecture. I have also provided a series of example designs, rules of thumb, and design paradigms that are good practice, if not best practice. These will help you get a robust design.

Here is the **first rule of thumb**: When designing secure firewalls, try to separate systems used for e-commerce from those used for browsing or other corporate communications. Indeed, as a broad paradigm, where possible you should separate the security arrangements for different types of traffic.

This statement might sound lame, but operations and network management will try to squeeze everything into one domestic home firewall hanging off a small DSL line. Then they will leave you holding the baby.

The e-commerce and corporate browsing infrastructures should be segregated because they have different:

- **Availability requirements** Loss of your browsing capability for half a morning will get you a nasty phone call from the marketing director, who spends all day “connected.” Loss of your online shop for half a morning could cost you dozens of customers and serious revenue.
- **Bandwidth requirements** Do you really want your e-channel being ground to a halt by staff watching video streams of the latest sporting event?
- **Regulation requirements** Data protection, money laundering?
- **Security requirements** Different containment and separation requirements, different confidentiality requirements that should be reflected in access controls and authorization procedures and the like.

But I’m not suggesting every corporate brochureware Web site has its own firewall and management platform. I usually use the following acid tests to help me decide:

- **Does it take transactions?** If it does, it will be a revenue center on someone’s balance sheet. These people will almost certainly demand a level of service greater than can be expected for a simple corporate Web server. They should pay for that greater service.
- **Has it got customers?** Not all customers create transactions, but they might have paid for a service. So the same argument applies.

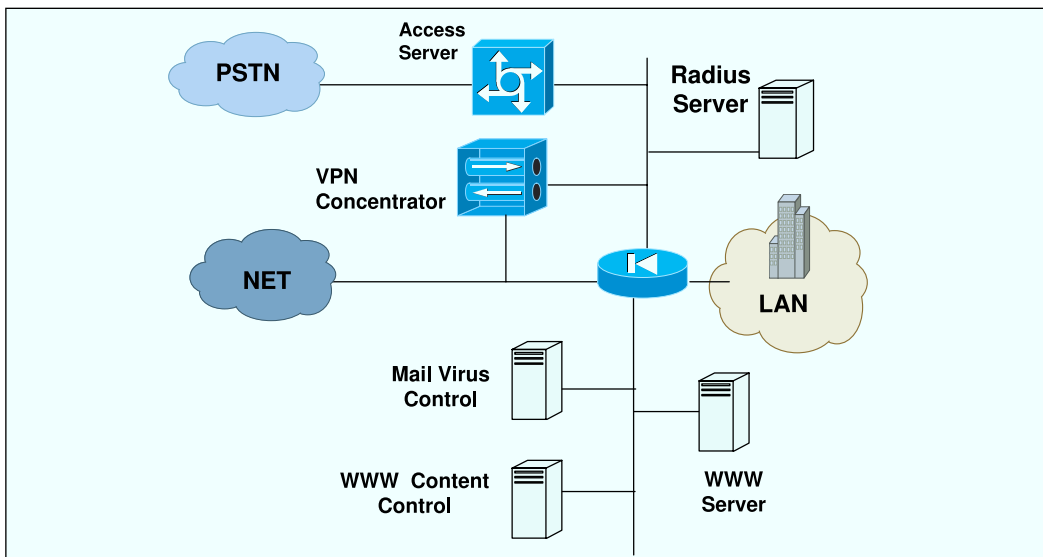
- **Has it got a nontrivial database with personal details on it?**  
This is a little avant garde, but in my opinion, systems that have databases need DBAs, so they are not trivial.

If any of these answers is yes, I tend to look for separation between the e-commerce system and the corporate firewall.

## The Corporate Firewall

The corporate firewall typically handles a number of services that the organization uses to communicate with the outside world. Here we are usually talking about internal Web browsing and the ubiquitous e-mail. However, few organizations these days are without remote access for roaming or home workers. This requires that we include IPSec-based or SSL-based VPN and/or PSTN dial-in (see Figure 7.1).

**Figure 7.1** The Corporate Firewall



If you went to a professional security boutique and asked for a design, you might receive something like this. It is aimed at a small organization so it hasn't used a defense-in-depth firewall configuration, but this layout is more than fit-for-purpose. And your designer has done it so many times before, he



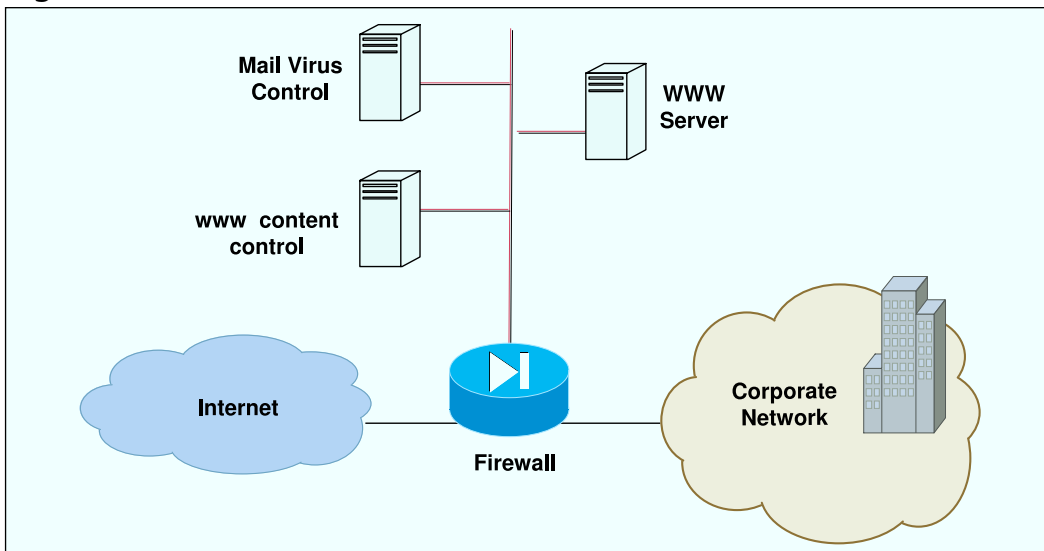
probably did it automatically, without thinking. But let's take a moment to analyze the services and the ways the inherent risks are managed.

There are really two structures here—the hardware (plus software) deployed to provide:

- Remote access into the organization
- Internet surfing and mail

There is nothing wrong in combining them if the throughput is negligible, but let's see how they look when deployed as two separate structures, the way a medium-sized organization should. (Why? Well, the functions require different services and therefore have a different risk profile. Remember the first rule of thumb.) Figure 7.2 diagrams an Internet access firewall.

**Figure 7.2** An Internet Access Firewall



## Threat Analysis

From Table 7.1 we get another design **rule of thumb**: Terminate all inward traffic in a firewall DMZ. This can be achieved by either placing the server in the DMZ or using a protocol-aware proxy.

**Table 7.1** Threat Analysis of a Corporate Firewall

Activity	Threat	Countermeasure
Public network connectivity	Hacking/unauthorized access	Firewall
External access caused by security flaws in operating systems or apps	Hacking	DMZ; all inward services terminate in DMZ
Web Browsing by staff	Looking at porno, time wasting	Content control and reporting
Mail inward	Virus and worms Spam	Mail virus scanning Spam filters
Mail outward	Offensive comment or company secrets	Banned word list on Virus software

## E-mail Protection

Protection against mail viruses is *essential*. About 80 percent of all security incidents arise from the lack of such protection. Although as a function this protection is available on leading firewalls, it historically has not performed as well there. However, this hasn't deterred suppliers, and now the successors to PIX, NetScreen, and Proventia all have it as a feature.

However, I still believe e-mail protection is much more scalable when included as a mail relay in the DMZ. This has the following advantages:

- Your mail that isn't time sensitive can be offloaded to another processor.
- You can stack multiple engines. I recommend CLAM (freeware) and at least one commercial library.
- Extra features like disclaimer and spam control are available from the separate products.

Recent tests (from sources that must remain unnamed; there are many in security) support my theory, showing that certain combined appliances missed over 20 percent of the polymorphic viruses passing through them. Other security appliances only check a hot-list of most frequently encountered

viruses, meaning that old copies of Code-Red that are still circulating can still infect you. All-in-one appliances are good in many situations, but they are often an excuse to combine three poor products into one that is marketable. See the sidebar for more details.

## Tools & Traps...

### Notes on Security Appliances

Many industry analysts and vendors recommend that you get key security features bundled on one single security appliance firewall. These products are available, and sometimes they're appropriate. Here's an analogy.

I have a Swiss Army knife and it's great. It cuts nearly as well as any other knife on the market. It also has a little can and bottle opener. These aren't very good, but in an emergency, they will do the job at the risk of hacking a lump out of my fingers. Lastly, my trusty Swiss Army knife has a little saw that is truly hopeless, but it looks good. It's thrown in for free; it doesn't cut a tree down like my chain saw, nor will it cut mitres like my cross-pull power saw. But it looks good and it's free. Where's the harm as long as you don't need it every day?

So it is with multifunction security appliances:

- Some features are good (for example, the firewall and encryption functions). These features are equally as good as sole-purpose products.
- Some functions, such as URL blocking for Web sites, do an acceptable job *and can be compared to standalone products*.
- Some functions, such as IDS or virus protection, generally *don't do the job to the required standard*. If you buy the MessageLabs service, it uses three separate virus signature libraries. I use MIMESweeper, with a minimum of two. The risk of infection by mail is high, so you really don't want to make false economies here. These functions also detract from the main function of the firewall. Often you find that a 100mb/s firewall turns into a 50mb/s one because of the extra CPU requirements of these other functions.

Continued

You are the security officer and you are responsible for the maintenance of the security of your organization's very expensive IT, so don't let these important functions be trivialized for the saving of a few hundred dollars and one shelf in a rack.

Typical products are:

- Spam Assassin
- MIMESweeper
- Mail Marshall

Alternatively, you can outsource all activity to a third party that will do the spam and virus removal for you. Leaders in the field are MessageLabs, BlackSpider, and Postini.

## Browser Content Control and Logging

Content control products prevent employees browsing unsuitable content, such as porno or hacking sites. Although there are no laws that require these control facilities, corporations increasingly buy them to limit legal exposure. A number of corporations have been sued for constructive dismissal on the grounds that they did not provide a suitable workplace because staff viewing pornography and other offensive material was making the office an “unfit workplace.”

Additionally, it is clear that many employees who work the hardest and have no time to help others always seem to have a browser open on their machine. Targeting them or even viewing packets with a packet sniffer could infringe their human rights, but these automatic devices have generally been recognized by courts in sensible countries as reasonable controls.

Typical products are:

- CensorNet
- Surf Control
- WebWasher
- Web Sense
- Blue Coat

## Web and FTP Server

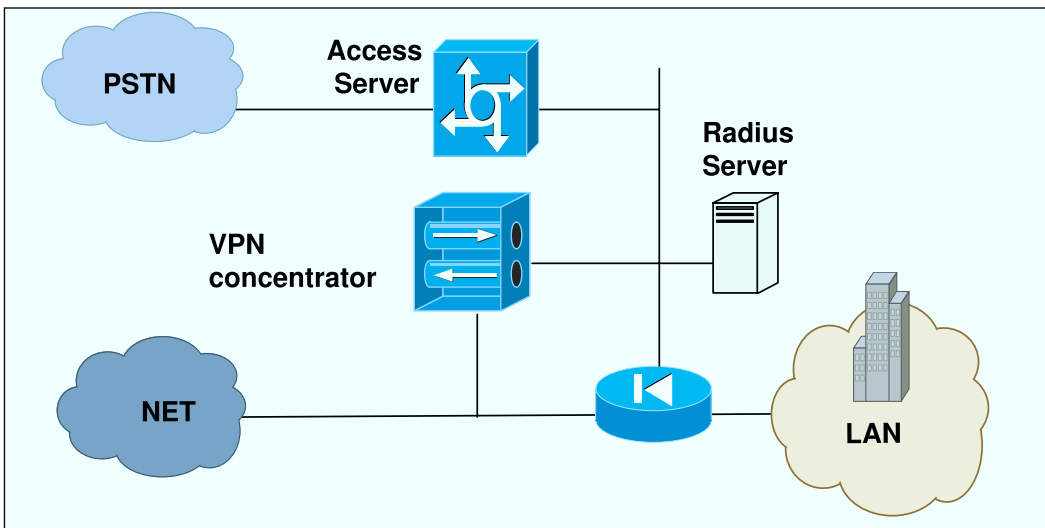
Typically, the DMZ also holds a simple corporate Web server or FTP server, even if the company has an e-commerce site elsewhere. Make sure that these servers are suitably hardened and administered separately.

## Remote Access DMZ

Many organizations have a legacy dial-in remote access server. This server will need access to a radius-based authentication server (see Figure 7.3).

More modern remote access will be enabled by either an SSL or IPSec VPN server, using the ubiquitous Internet. It is good practice to “DMZ” both of these to allow better access control. I’ll come right out and say it: Many VPN servers provide lame firewall service and aren’t certified to EAL4. Rant over.

**Figure 7.3** Remote Access DMZ



## Threat Analysis

The threats and countermeasures look something like the list in Table 7.2.

**Table 7.2** Threat Analysis of a Remote DMZ

Activity	Threat	Countermeasure
Public network connectivity	Hacking/unauthorized access through insecure, nonmandated protocols	Firewall
	Hacking/brute-force attack providing entry via authorized protocols by repeated trial of user ID and password	Strong authentication
	Getting access to confidential information	Encryption
	Man-in-the-middle attack	Encryption plus strong authentication
Mail inward	Viruses and worms	Optional mail virus scanning

## Remote Access Design Options

Personally, these days I usually run encryption over the PSTN. The local operators in some parts of the world aren't very secure and run it through IP over the Internet, so it makes sense. To achieve this goal, a direct connection into the VPN concentrator from the access server is required. Typically, a port on the same virtual LAN (VLAN) or a new VLAN and an extra network interface card (NIC) in the concentrator will do it.

For strong authentication, I am particularly fond of the one-time password generators, so you should consider deploying:

- Cryptocard
- RSA SecuriD
- Vasco token—Digipass

These can be used with the PSTN, SSL VPN, or IPsec VPN. They should speak “radius” as a native tongue so they require no further integration.

As a one-time evangelist of PKI, I should recommend digital certificates, but I'm not, simply because they are too difficult to manage for a typical

medium-sized organization. (Please note that I'm not suggesting that risk/threat is a function of size only; capital expenditure and head count to maintain large infrastructure *often* are. Many investment banks are “medium-sized” but need and use Rolls Royce countermeasures.)

If you are dying to use an integrated firewall appliance, this is your chance. As mentioned before, many of them have virus capability, and *extra* virus scanning of your remote access connection is a useful addition. Many of them can also speak SSL-VPN and IPSec-VPN, so you can end up with all the functionality in one box. That has to be good for the flexible enterprise. But please make sure that virus scanning occurs on the decrypted VPN stream. Use the dummy virus EICAR to check.

Lastly, and as mentioned earlier in this chapter, it would be good to have IDS included. As an old fossil, I would not jump at the chance to integrate IDS into an appliance. An IDS is a detective control, and therefore I like it to have complete separation (in other words, segregation of duties).

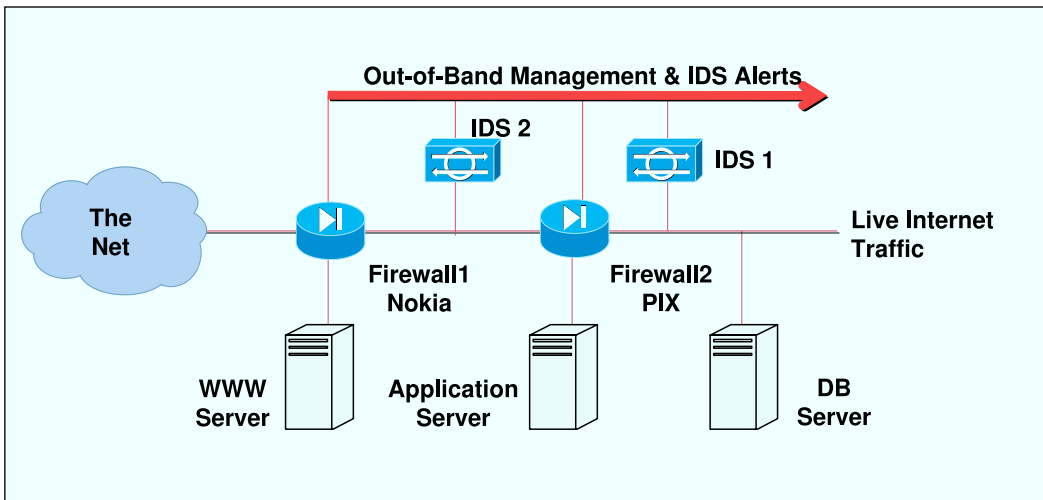
## E-commerce

Designing an e-commerce configuration is not very much more difficult, but it does require an understanding of the application that is going to be hosted and the products that are to be used. I once worked at a hosting organization that applied one standard £50,000 infrastructure onto the typical £2 million application, with no regard to the protocols and the risks. Surprise, surprise—the net result was not just poor firewalling. It extended to a virtual war between support and sales, with perplexed and unhappy customers to boot.

The fact is that application of defense in depth is a lot more complex than it might seem. Issues that are most likely to be encountered are:

- Cost—as always
- The need for resilience
- Better detective controls

A good nonresilient configuration might look like the diagram in Figure 7.4.

**Figure 7.4** A Network Configured for E-commerce

This is a classic nonresilient design. Let's observe the key design characteristics to help us form some rules of thumb for good practice:

- **Dual “defense in-depth” firewall configuration** Two firewalls from different manufacturers. In this design, we place a PIX with its robust stack and limited OS as the external firewall. We would place a more flexible, software-based firewall, such as Check Point FireWall-1 running on a Nokia platform, as the internal firewall. This firewall is feature rich and can provide some excellent enhancements.
- **Use of tiered design** Each application layer has its own security zone, or tier. This extends the concept of defense in depth into the realm of the principle of least privilege. Each class of server can only communicate to servers in the same tier (and therefore with the same risk profile) or to specific servers outside that tier on a specific port allowed by a firewall rule. This means that if an exposure develops, it is contained in the appropriate tier.
- **Use of detective and preventive controls** IDs and firewalls.



- **Out of band management** Access to the firewall, servers, and IDS is via a special separate network. This provides a degree of separation discussed in the previous section.

## Notes from the Underground...

### Out-of-Band Networks

Out-of-band (OOB) networks have to be made especially secure. Many times during my years as a penetration tester, I found that an OOB had not been prepared properly, so it represented a high-speed back door for hackers into the heart of the corporate network. OOB networks should always be terminated on a firewall. This really is the application of our first rule of thumb that we mentioned earlier.

In the earlier example, we have achieved a degree of separation in the most common and economic manner. We run the OOB network to the IDS and firewalls but from then on use the existing infrastructure to reach the application servers. This means that management traffic could be entering into the same server interface as data, and this might be unacceptable.

If more separation is required, each server must have a separate NIC installed. Define each DMZ as a separate Security Zone and give it a separate physical switch to prevent VLAN hopping. In our example:

- Zone 1 (outside) = Firewall1
- Zone 2 (middle) = WWW server, IDS2, and Firewall2
- Zone 3 (inner) = Application DB servers and IDS1

If you are completely paranoid, you can do what I have seen recently and give each device a separate VLAN—doable, but nuts.

## Threat Analysis

The threats and countermeasures look something like the list in Table 7.3.

**Table 7.3** Threat Analysis of a Network Configured for E-commerce

Activity	Threat	Countermeasure
Public network Connectivity	Hacking/unauthorized access causing reputation loss because the Web site is defaced	External firewall
	Hacking/unauthorized access causing regulatory or financial loss because personal data is accessed or damaged	Internal firewall and IDS/IPS
	All admin/privileged access allowed only via out-of-band network	Out-of-band admin network, strong authentication
	DDOS attack	No countermeasure
	Unavailability due to configuration error	No countermeasure
Identity theft	Getting access to confidential information	Encryption (SSL)
	Man-in-the-middle attack	Encryption plus authentication
Unauthorized access from LAN	Insider attack	Internal firewall, IDS/IPS, out-of-band admin network with strong authentication

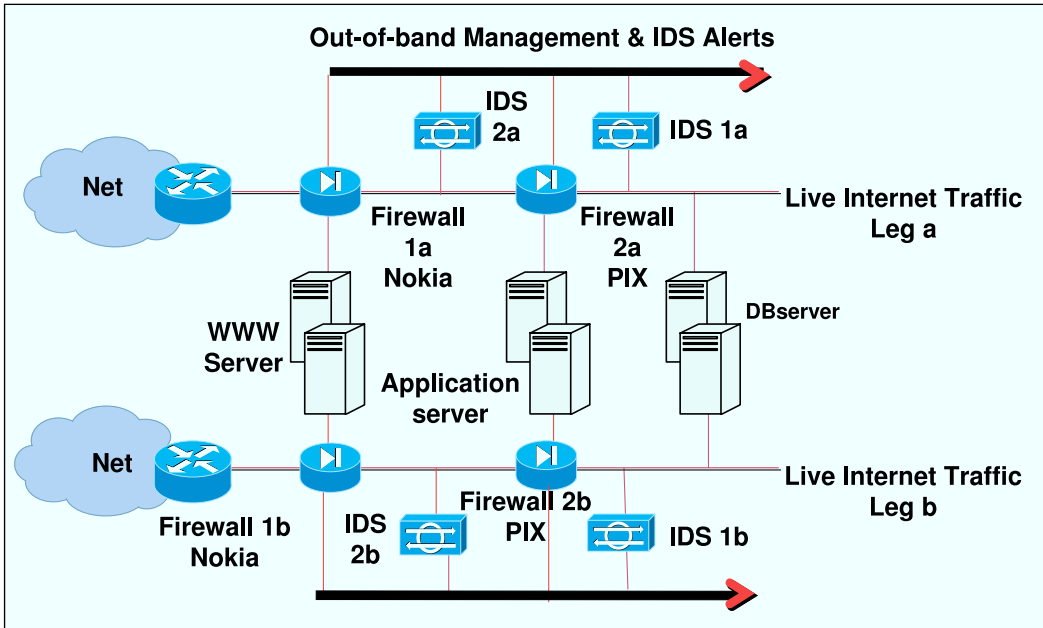
The previous configuration deals well with confidentiality and integrity but not availability. Everything is a single point of failure. A small error in a firewall rule or a disk failure will see the whole e-business offline for a number of hours while lengthy restores are conducted or replacement firewalls found.

In the configuration shown in Figure 7.5, the single points of failure have been removed by the duplication of all key components. In effect, it is two of the simple configuration “bolted” together to form two legs. This gives rise to another rule of thumb.

**NOTE**

Where possible, eradicate single points of failure.

**Figure 7.5** Duplicating Key Components in a Network Configured for E-commerce



Obviously, this should be done based on needs, mean time to failure (MTTF), and service level agreements (SLAs). Redundant configurations can be created very simply, usually with the minimum of effort in active/passive mode. This is where you buy two pieces of kit (firewall, router and switch) but only use the second of each (nominated the secondary) when the first (nominated as the primary) fails.

Management, however, tends to find this solution unacceptable because they pay twice as much but 50 percent of the purchase sits in a corner gathering dust (figuratively speaking, unless it is a very cold spare).

Management, therefore, tends to prefer active/active configurations, because they gain a perceived performance benefit from the having two

pieces of kit in use at once. The presumption is that *two is better than one*. This tends to be a myth, because typically, the performance bottleneck isn't the routers, firewalls, and switches but the application—but *c'est la guerre*.

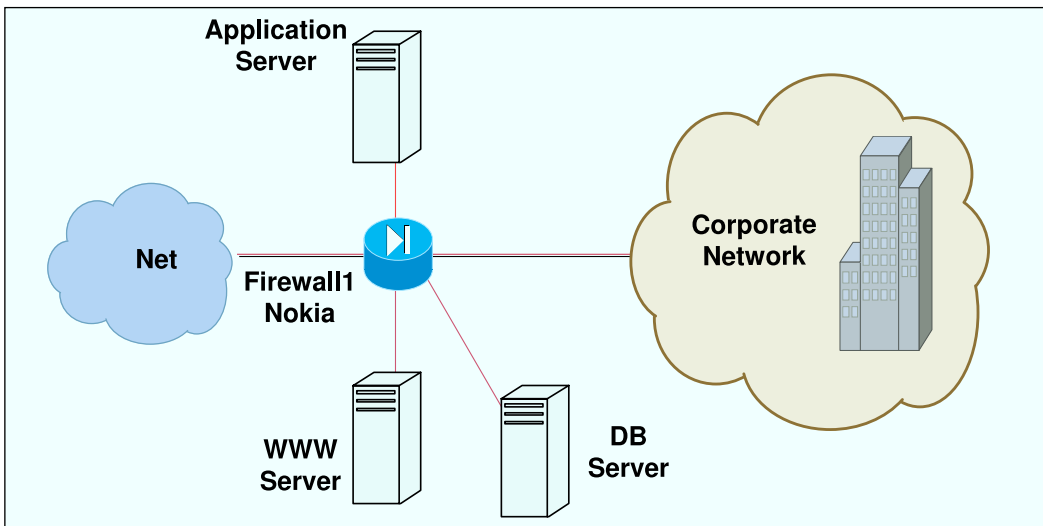
When you're trying to create active/passive structures, beware of:

- **Asymmetric routing** Your firewall and your IDS will probably not like an inward path being different from your outward path.
- **Nonsticky sessions** Your application will almost certainly need to be served continuously from one server; it needs to be sticky. Otherwise cookies and session data go walkies.
- **Memory creep** Data retrieval time increases through loss of locality of reference.
- **Span tree or route convergence problems**
- **Problems with client-side certificates**

And despite it all, the application will probably run slower.

However, if your organization is a little strapped for cash, the single-firewall configuration shown in Figure 7.6 has a degree of the protection that a bigger configuration gives.

**Figure 7.6** A Single-Firewall Configuration



Here we have a single firewall. However, each tier is given a separate interface to form a DMZ—a reasonably low-cost option that provides a high degree of containment between the presentation tier, the application tier, and the database tier. This means that each can have a separate access list that prevents direct access to standing data from the outside.

## Threat Analysis

The threats and countermeasures look something like the list in Table 7.4.

**Table 7.4** Threat Analysis for a Single Firewall Configuration

Activity	Threat	Countermeasure
Public network connectivity	Hacking/unauthorized access causing reputation loss because the Web site is defaced	The only firewall access list on interface 1
	Hacking/unauthorized access causing regulatory or financial loss because personal data is accessed or damaged	The only firewall access list on interface 2; outside addresses are never allowed to touch the databases; access must be from the application
	All admin/privileged access allowed only via out-of-band network	No countermeasure
	DDoS attack	No countermeasure
	Unavailability due to configuration error	No countermeasure
Identity theft	Getting access to confidential information	Encryption (SSL)
	Man-in-the-middle attack	Encryption plus authentication
Unauthorized access from LAN	Insider attack	No countermeasure

# Just Checking

When you think you've finished your design, just ask yourself if you've remembered:

- **NTP** You will need a time server or at least a time source; otherwise your logs will be useless.
- **Syslog server** Oh yeah, *logs*. You'll need a syslog server for the UNIX box's switches and routers.
- **Console** If the data center is remote, you need to get to the console, so you'll need a console server.
- **Authentication** Are you going to centralize user accounts or use strong authentication? This can be very problematic with Windows and Active Directory.
- **Backup** Have you got backup software and a tape drive installed (on your management LAN) to back up the app?

## Summary

The purpose of this chapter was not to show you how to design complex DMZs and firewall solutions for your organization. That's not your job (although if you have absorbed the information as I have hoped, you would do a better job than most). The purpose of this chapter was to show you the rationale behind common designs and the considerations behind the designs. This knowledge will enable you to ask searching questions about the design and the protection it affords to your company—and that *is* your job. How can you do it without this knowledge?

In the chapter, we covered my basic rules for design. These are only guidelines but are listed here again. When you begin asking the questions about your particular setup, these points provide a very useful start. Use:

- Separate firewalls for e-commerce and general corporate use
- Proxies or DMZ-servers to terminate all inward traffic to the corporation

- Dual defense-in-depth firewall configuration, where required
- A tiered firewall design
- Detective and preventative controls
- Out-of-band management
- Duplicate components to eradicate single points of failure

Now on to firewalls.





## Firewalls

**The purpose of this chapter is to:**

- **Illustrate key features of a firewall**
- **Provide a very brief overview of Cisco PIX**
- **Provide a very brief overview of Check Point FireWall-1**

## Anecdote

*I was sitting in Deutsche Bank talking to the main man in information security, the CISO. He said, “If it wasn’t for the Internet and firewalls, I’d still be earning \$45,000 a year working for a minor IT manager.” This dude was on a yearly salary of at least \$150,000.*

*And that is a fact for most of us. Nobody, not even the banks, paid good money for security before IP networks became predominant. When the Internet took off, if you were one of a couple dozen people in the U.K. who could spell firewall, you got treated like a minor rock star.*

*God, how I miss it.*

## Introduction

The previous chapter provided basic rules of thumb and design paradigms for deployment of various key pieces of infrastructure such as firewalls, IDSs, and IPSes. This and the following chapters look at these building blocks in more detail.

## What Is a Firewall, and What Does It Do?

The term “firewall” has been adopted to describe a single piece of software and hardware that protects a network. A firewall is a little black box that keeps the bad guys out.

Historically and more properly, it is more than just one device/function—it is that combination of hardware or servers, software and management activities used to control communications between internal networks and external networks. But I’ve pushed back so much against common belief in this book so that in this chapter when I refer to firewall, I’ll be talking about the more generally accepted meaning—a firewall gateway like PIX, FireWall-1, NetScreen, or Fortigate. When I refer to firewall architectures I’ll be referring to the overall security structure that may comprise:

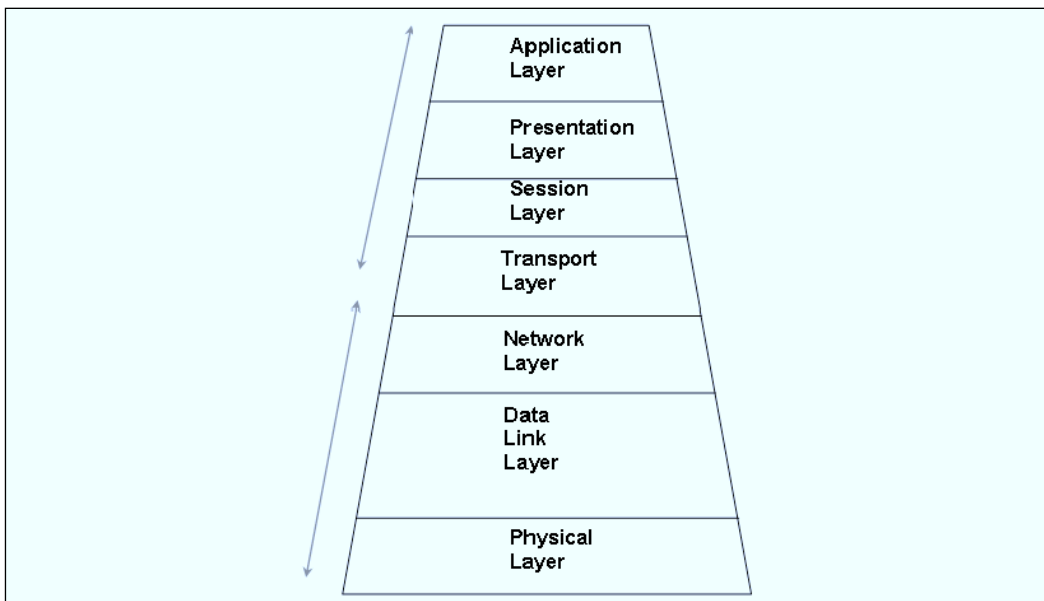
- Proxy servers, Web filters, and IPSes
- Routers and load balances
- Switches and DDoS devices

The term *firewall* comes from the bulkhead often found in a plane or ship between the nasty, sparky, explosive engine and the nicely combustible crew. The firewall keeps the crew or employees safe from danger; hence, the analogy gives rise to the name for network firewalls because they keep us safe, too. Commercial firewalls (and they are not fads; you will need them when you start to deploy security in the wild) exhibit a host of features, but the actual purpose of a firewall is to act as:

- A mechanism for allowing only authorized traffic (incoming and outgoing) to pass through it
- A robust, impenetrable gateway between internal and external networks that prevents any unauthorized traffic to traversal

But how does the firewall do all that? To understand this, you need to understand IP networking. The most widely used protocols on which the Internet is based are collectively known as *Transmission Control Protocol / Internet Protocol* (TCP/IP). These protocols were proposed and designed in the 1970s. TCP/IP is loosely based around a simple five-layer model. However, nobody uses this model to describe networking. They all use the Open System Interconnection (OSI) model, which is shown in Figure 8.1.

**Figure 8.1** The OSI Model



- Firewalls are fundamentally Layer 4 devices that use IP addresses from Layer 3—you know, the things like 10.0.0.1 that you see pop up on your screen from time to time—to police which computers on the inside of your network communicate to which computers on the outside of your network. To provide more granular control, firewalls also use the port numbers from Layer 4 that represent the various services such as www/HTTP, FTP, or mail to restrict access still further.
- Firewalls generally are controlled by a firewall policy, which is not a document but a set of rules that defines these sets of communities that can or cannot talk to each other.
- Whatever make or model you buy, firewall policies can generally be expressed as a simple set of rules that specify the to's and the from's:
  - *ALLOW <my-address any-port> to <outside-address mail port>*
  - *ALLOW <any-inside address port> to <any outside-address Web port>*
  - *DENY <any- address port> to <any address Web port>*

The last line is the logical implementation of the paradigm *Deny what is not specifically allowed* and is thoroughly good practice.

## Why Do We Need Firewalls?

We don't *need* firewalls, and a growing body of policy makers and academics in positions generally shielded from the real world promulgate this idea. Others share this view, particularly those who work in businesses that need to communicate more widely and with a greater level of granularity than can be protected by a firewall. And in theory, they are right. The fact is, if every database manufacturer, operating system programmer, and enterprise resource management (ERM) vendor put as much effort into security as a firewall manufacturer does, there would be absolutely no need for firewalls. Furthermore, if each system administrator worked as hard on security as the typical firewall administrator does on security and devoted as much time to hardening their servers and laptops, no centralized firewalls would be needed. Of course, this is extra effort that simply isn't performed today, so there will

be a large cost to prohibiting traffic on each server and PC in the organization instead of doing it only once on your centralized firewalls—but *surely* no organization would shirk at hiring a dozen extra sys-admins or spending several million on distributed LAN-based IPS and laptop firewall software.

And there you have the real truth. At the moment (for most organizations), all traffic enters the network through several perimeter routers linking your local loop to your network provider (or via a wireless AP). It is, therefore, for the average organization, absolute common sense in terms of cost and logistics to secure the connections here. In the future, this will change, I genuinely believe, but to add a note of caution, this revolution has been predicted to happen any day—for at least the past seven years.

For the present, we use cost-effective, centralized firewalls for three main reasons:

- **Intrusion** We don't want outsiders stealing our CPU MIPS, bandwidth, or corporate secrets. Firewalls provide access lists that prevent unauthorized parties doing some or all of these things.
- **Denial of service** We don't want outsiders stopping us from doing our business with our IT systems by flooding our network or entering garbage.
- **Outbound control** We have a duty to protect our users from using insecure services and to prevent them being destructive to the outside world.

## Firewall Structure and Design

Firewalls have different structures and designs. In this section we discuss various firewall types.

### Firewall Types

Traditionally, there were two types of firewall: the proxy-based firewall, which was considered the most secure for a long time, and the circuit gateway. These days most manufacturers use a mix of techniques, but an overview of their history is valuable.

## Screening Routers

Routers are extremely effective at routing packets across a network. *Screening routers* are configured using rules to filter access using specified protocols or to and from predefined addresses, passing or rejecting an IP packet based on information contained in the packet header. They have the advantage that no changes are required to infrastructure because they operate entirely at the transport and network layers of the TCP/IP model.

However, used as a stand-alone solution, the screening router has several disadvantages:

- There are little or no logging facilities making it difficult to determine whether an attempt has been made to compromise security.
- Packet filtering are stateless—meaning that even the simple act of allowing FTP results in complex filtering rules, which are difficult to implement and maintain and are frankly inferior. But they still have their place in asymmetric routing, for example.
- Routers are not firewall strength. Services are open rather than closed by default. TCP packets are often checked only on the initial packet.

## Application-Level Gateways or Proxies

*Application-level gateways or proxies* are specialized application or server programs that are resident on a gateway that is set up not to route (i.e., IP_FORWARDING=0), so the only way to afford a connection to an external interface is through one of these server programs. The user connects to the gateway proxy; the proxy then connects to the required service on the untrusted network. Application gateways operate at the higher layers of the TCP/IP model and can therefore provide protocol-specific (i.e., *get* but no *put* on FTP) access controls, handling “store and forward” traffic as well as interactive traffic.

The main disadvantage associated with application-level gateways is that they require special-purpose code for each service to be relayed, increasing the complexity and maintenance overheads. Also, in these days of high-speed networks, it is considered quite an overhead to double the number of sockets opened and processed—which is what happens when you open a proxied

connection. This also has an effect on how servers can determine the endpoint of traffic; proxied firewalls limited the use of client authentication of x509 certificates, for example.

The main advantage is that the proxy can enforce specifics of the protocol that other types of firewall never could—enforce RFC-compliant HTTP, for example. However, the sad fact is that such firewalls never really did.

These mechanisms ruled the Internet for a great many years with programs such as TCPD, TIS firewall toolkit, and TIS Gauntlet.

## Circuit-Level Gateways

*Circuit-level gateways* do not interpret application protocols in the same way as application-level gateways. Their primary advantage is that they can provide services for a wide range of protocols. They allow protocols purely on the protocol and to/from port/address tuples. This meant that no protocol-specific security information was checked. This meant that they were useless for protocols such as FTP or RPC that need specific protocol processing. However, no one could live without the flexibility of this type of firewall, so the second-generation circuit-level gateway gave birth to the modern firewall.

What was needed was a blend of the two.

## The Stateful Inspection Firewall

The stateful inspection firewall is the firewall of today. It is ubiquitous, and it was the product of the need for something as secure as a proxy but with the simplicity of a circuit gateway. In essence, the stateful inspection firewall may examine not just the IP and TCP header information but also the contents of the packet up through the application layer, to determine more about the packet than just information about its source and destination.

From that it determines whether the:

- Connection is genuinely using the protocol associated with the target well-known port or is it an attempt just to tunnel to malevolent endpoint.

- Connection is using an advanced protocol, which requires additional ports opened for it to work. The most common example here is FTP, which requires both TCP port 20 and TCP 21 opened to facilitate file transfer.
- Connection contains an option that should be disallowed.

If the firewall code (which is usually implemented as a shim on the stack) verifies that the connection is safe, it will allow the packet to be routed on by the host operating system.

This is great, because allowing FTP (for example) into an organization used to be a nightmare with routers, or in the case of a proxy, it required the user to learn a whole new syntax. With stateful inspection, one line on the firewall allows not only the command channel through but also the subsequent data channel.

All vendors overplayed the extent that they delved past the TCP and UDP headers. Many protocols were checked to see that they corresponded to the format specified in the requests for comment (RFCs); for instance, ICMP was usually checked in most implementations to ensure that incoming packets not only had the correct IP protocol number in the IP header but also contained the correct verb codes (i.e., *echo unreachable time* and the like). If they were incorrect, they were dropped.

However, DNS on most of these firewalls, at one time or another, had issues that were borne out of a less thorough implementation of the protocol in the stateful inspection mechanisms.

For many years, hackers used to scan networks from a source port of 53, the DNS well-known port, because they knew that many firewall manufacturers didn't implement rigorous protocol analysis or state checking. For example, protocol analysis on UDP 53 should assume that the incoming packet is a DNS packet, but if it doesn't look like a DNS packet with the appropriate structure, the firewall should drop it. Similarly, state checking should ascertain if anyone has requested what seems to be a DNS response and if not drop the packet. What seemed to be generally implemented by most vendors was the assertion "It's UDP 53 That's DNS! That's OK!" This isn't providing ideal protection.

These kinds of problems gave rise to the sequel to stateful inspection, *deep packet inspection*—which is basically the same thing only it works better.



## So What Are the Features You Want from a Firewall?

In this section, we'll discuss the features that you'll want a firewall to include.

### Stateful Rule Base

As discussed earlier in this chapter, these days a firewall is expected to have a stateful rule base. One rule (one line of policy) should allow the packet out and back in.

### NAT/PAT

There are two fundamental reasons that network address translation is used on nearly all Internet firewalls these days. These are:

- The number of available unique addresses is finite, and so the number of public addresses allocated to an organization is usually too small to be used for every device on the network.
- Security through obscurity; the idea is, by keeping a low profile, attackers will “pass you by.”

To overcome the first problem, most corporate networks use RFC 1918 addresses for their internal LANs. These are:

- Class A 10.0.0.0 mask 255.0.0.0 (/8)
- Class B 172.16.0.0 –172.31.255 mask 255.240.0.0 (/12)
- Class C 192.168.0.0 mask 255.255.0.0(/16)

With these, a network manager can divide his or her network into many addresses and not worry about the number of public addresses allocated in your Internet address registry. Network address translation will help the manager translate the RFC 1918 private address to real Internet routable addresses.

Security by obscurity is often discredited—and generally it is true that in any field, a security expert in that field will quickly see through the façade. However, such mechanisms are not designed to fool experts; they are designed to distract the amateur, and that is exactly what security by obscurity does. Try doing an NMAP scan on an Internet firewall with NAT and then compare it

to one without. You'll see that the network is far more difficult to enumerate when NAT is used. And if it's harder to see, it's harder to hit.

There are three types of NAT: static, dynamic NAT, and dynamic PAT

### *Static NAT*

A one-to-one translation based on a STATIC table, hence the name. The internal address will always be translated to a specific external address. These will be used for mail servers and their MX record, Web servers, and FTP servers that must always translate to the same internal or DMZed address because they need visibility from the Internet.

### *Dynamic NAT*

Dynamically translate a source address to an address selected from an address pool. This method is used for browsing and desktop traffic. It relies on the notion that not everyone will be browsing the Internet at the same time. Typically, the address used is a factor of 10 or even 100 smaller than the user community. There must be no need for visibility from the Internet.

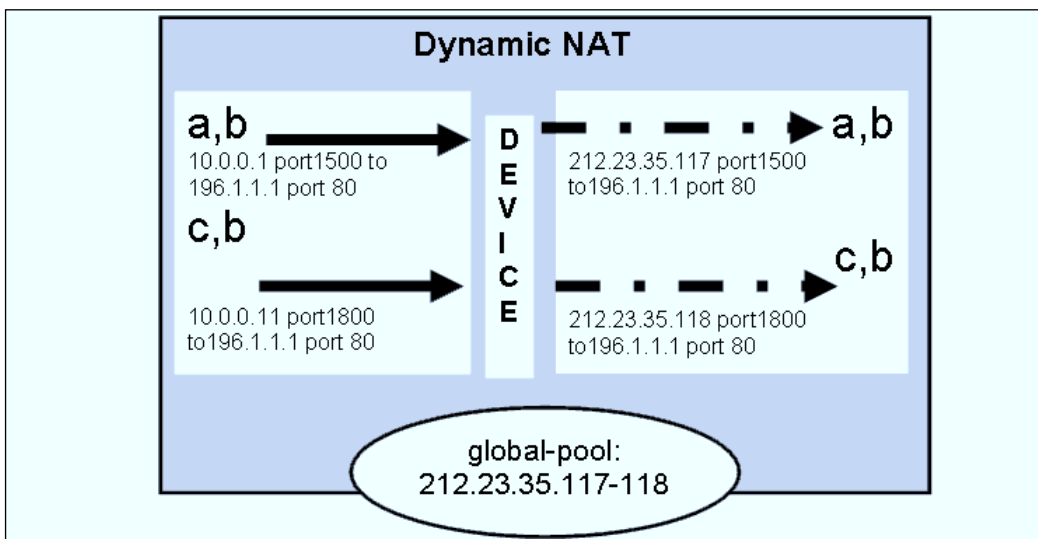
When a user accesses the Internet:

- A check is made in a table to see if the user has an entry; if not, a slot in a translation table is created. This contains a time and the user's internal address.
- The user is then randomly allocated an address from the address pool, which is stored in the address translation table. Every time the user browses the Internet, he or she is given that same external address.
- When the address pool runs low, the old slots in the translation table are deleted, and the external addresses are returned to the free global pool.

As long as the browsing community is not too big and the global pool not too small, this scheme works well.

In Figure 8.2, user *a* on address 10.0.0.1 wants to visit Web site *b* at 192.1.1.1.

Figure 8.2 Dynamic NAT



User *a* sends a packet from his address 10.0.0.1, an internal address that cannot be used on the Internet because it is not unique (from a random port TCP 15000), to port 80 on 192.1.1.1. This passes through the firewall, which recognizes that 10.0.0.1 must be translated.

The firewall searches the global address pool for a free address. It finds 212.23.35.117 free and so sets up a dynamic translation table between 212.23.35.117. This will remain in force until the firewall notes that it hasn't made a translation for some hours.

The firewall modifies the packet so that the request to 192.1.1.1 comes from 212.23.35.117 (in other words, from:212.23.35.117 TCP 15000 -> to:192.1.1.1. port 80).

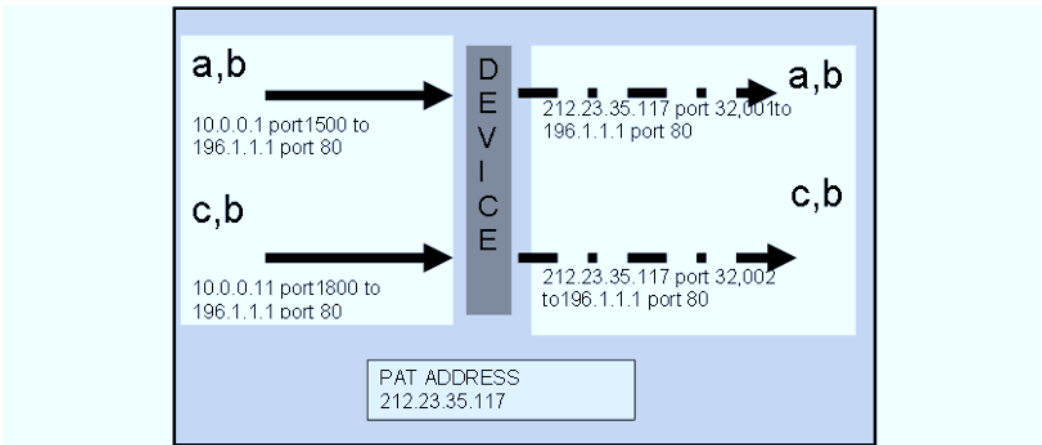
When the Web server responds, it sends data to 212.23.35.117 TCP 15000. The firewall translates this (from 192.1.1.1. port 80-> 10.0.0.1 TCP 15000) and forwards the data on to the requestor.

When 10.0.0.11 makes a request, the firewall repeats the process but uses the next free address (212.23.35.118) because it knows 212.23.35.117 is already reserved.

## Dynamic PAT

Where the browsing community is large and the pool of available public addresses small, Dynamic PAT can help. Dynamic Port Address Translation dynamically translates a source address to a public address (sometimes that of the outbound interface) and changes the source port to a free ephemeral port (those client ports above 10000) on that machine. Effectively, it behaves just like a proxy client. The diagram in Figure 8.3 explains it all.

**Figure 8.3** Dynamic PAT



In the PAT example, user *a* on address 10.0.0.1 wants to visit Web site *b* at 192.1.1.1. She sends a packet from her address 10.0.0.1, which is an internal address that cannot be used on the Internet because it is not unique (from a random port TCP 15000), to port 80 on 192.1.1.1. This passes through the firewall, which recognizes that 10.0.0.1 must be translated.

The firewall searches for a free TCP ephemeral port. It finds port 32,001 free and so sets up a dynamic translation table between 212.23.35.117 (the PAT address) and TCP port 32,001 (see Table 8.1). This will remain enforced until the firewall notes that it hasn't made a translation for some hours.

**Table 8.1** A Dynamic PAT Table

Original Address	Original Port	PAT Address	PAT Port
10.0.0.1	TCP 15000	212.23.35.117	TCP 32,001

The firewall modifies the packet so that the request to 192.1.1.1 comes from 212.23.35.117 TCP 32001 (i.e., from:212.23.35.117 TCP 32001 -> to:192.1.1.1. port 80).

When the Web server responds, it sends data to 212.23.35.117 TCP 32001. The firewall translates this (from 192.1.1.1. port 80-> 10.0.0.1 TCP 15000) and forwards the data on to the requestor.

When 10.0.0.11 makes a request, the firewall repeats the process but uses the next free port (i.e. TCP 32002) because it knows TCP 32001 is already reserved.

## Antispoofing

*IP spoofing* occurs when a machine masquerades as another machine by *borrowing* the IP address of a host that is trusted. This is a real problem in a modern network. If you concede that most firewalls rely on IP addresses being unfakeable and then you discover that IP addresses are easily fakeable, you should feel your faith in firewalls rocked.

Good firewalls overcome this via *antispoofing* rules. Examples for both Check Point FireWall-1 and Cisco PIX are shown later in this chapter. Basically, the firewall is made *interface aware*, and instead of just passing traffic based on its routing table and the security policy, it also labels the incoming packet with the interface that the packet arrived on. It then asks itself whether that packet arrived from the *outside* with a source address which should be on the *inside*. If that is the case, the firewall drops the packet.

## Advanced Logging

A firewall that has no logging or few logging features is a waste of time.

## User-Authenticated Traffic

Although user authentication is not traditionally a function of a firewall, it is a phenomenally useful feature of the two leading firewalls.

Typically, firewalls implement a number of schemes:

- **Per-session authentication** Here protocols such as Telnet, FTP, and HTTP that already have authentication are reauthenticated at the firewall. Typically, this is done to add strong authentication for

external users. For example, when a user attempts to contact the inside Telnet server from the outside, the firewall intercepts the call. The user still receives a login prompt but has to reply with both the firewall user ID and the Telnet user ID, usually in the format *<firewall_userid@Telnet_Userid>*. The firewall strips out the firewall user ID (and the @ sign) and passes the other user ID to the Telnet server. The same process is conducted from the passwords, and if all is okay, a session is established. This process is repeated for each session and for each service on every server.

- **Per-workstation authentication** Where protocols other than Telnet, FTP, and HTTP are used that don't afford an authentication scheme, modern firewalls have workstation authentication. Here the client must contact a dummy address on the firewall using HTTP or Telnet. The client then logs in, and the firewall notes the workstation the user is operating from and allows this address through the firewall for a set period of time. When the user finishes work, he or she is supposed to log in to the firewall and then close the session down. In reality, users never do, so it is good to ensure that the timeout is set to a low value. Otherwise the next user of that workstation will inherit the access!
- **Client tunnel authentication** This usually requires a special piece of software to perform authentication. The most common is IPSec remote access software, which is the subject of the next section.

## IPSec Termination

IPSec has become the de facto encryption tunnel scheme for use on the Internet. Typically, your firewall should provide for fixed site2site tunnels and remote access clients, as described previously.

## Ability to Define Your Own Protocols

New products come out all the time. Often, software developers don't consider security, and that can make it very difficult to secure new products.

Recently, new protocols that accompany new products have become very

advanced, and if you don't have the ability to define custom protocols, it can make life very difficult.

## Time-Based Rules

If you check out modern authentication systems, they usually allow you to restrict login times, giving you the flexibility to say, for example, that this terminal may be used Monday through Friday from 9:00 A.M. 5:00 P.M. Some firewalls allow rules to be constructed like that, too.

## Other Types of Firewalls

Two different types of firewall are becoming common: stealth firewalls and virtual firewalls.

### Stealth Firewalls

Imagine you work for a bank or a government organization that has a policy (or a regulator) that states that every Web application must be protected by at least *two* firewalls between the Internet and the app. Then this bank buys a wonderful third-party HR and training system that runs on the Web, and you find out that this magical company only has one single firewall to protect all its five banking customers. (Sound familiar?) What do you do?

You've got two choices:

- **Install a firewall in front of your existing firewall** Because you've got a public addressable firewall and perimeter router, to be efficient with registered addresses you've probably used a /30 (subnet mask 255.255.255.252) address so you can't fit a conventional firewall in.
- **Install a firewall behind your existing firewall** To fit a device behind your existing firewall means a complete redesign and testing of the rules.

Whatever you chose, you are looking at extensive work! Before 2001, there was a third option—SunScreen EFS. This was Sun's firewall product of the time and it was not unlike Check Point's firewall. The major difference was that it had a stealth mode. In the preceding scenario, stealth mode, more

properly known as *transparent bridging mode*, meant that the firewall device could be easily slotted in between the perimeter router and the existing firewall, with no IP address changes required—fantastic.

The device simply sat on the network, watching packets arrive at its external interface. If the IP header corresponded to its rules or its internal state table, it would copy the packet unaltered to its internal interface, where it would continue its journey. If the headers did correspond to any rule, the device simply didn't copy it and wrote an entry to its log.

Obviously, management required an IP address, but that should be provided out of band—there was its downfall. SunScreen's user interface was terrible, so even though it is still around in one form or other, people stopped using its stealth mode.

Recently, a flood of Linux-based firewalls has reintroduced the feature, which in turn has revived the concept. Both PIX and Fortigate have this feature.

## Virtualized Firewalls

We have VLANs and VPNs—they are *virtual*. So now we need virtual firewalls. If you work for a Web hosting company, you could end up with hundreds of firewalls that cost a fortune in terms of just rack space and wire.

By allowing a firewall to manage dozens of separate and distinct rule sets that link into a dozen different private VLANs using dot1q, a good balance of operational efficiency and medium security can be met.

## Commercial Firewalls

This section is for the more technically minded reader and describes key features of the two most popular firewalls—Cisco PIX and Check Point FireWall-1.

### The Cisco PIX

The Cisco PIX is probably the second most popular firewall in use today and was possibly the first truly popular appliance-based device. It is very simple and, at the version discussed in this book (version 6.3(2); version 7.0 has a



host of new features and is scheduled for early release as of summer 2005), is a lot more basic than FireWall-1 or Cyberguard.

However, it is robust, durable, fast, and trustworthy; what more can you want in a firewall?

## Features

Attend any of the Cisco security courses and you'll see that the PIX has, in pseudo-marketing speak:

- **Cisco PIX firewall OS** An embedded operating system that is real time and secure, not subject to typical operating system vulnerabilities such as poor configuration of RSH or NFS like a Unix platform or maybe Null Session on Windows NT.
- **Adaptive Security Algorithm (ASA)** Technology that provides stateful connection control and security by obscurity. It also forces the organization to identify different security zones and prescribe security policies to each.
- **Cut-through proxy** User-based authentication of inbound and outbound connections, providing improved performance over proxy filters and increased transparency.
- **Stateful failover/hot standby** Fully redundant topology (failover) can be configured extremely easily.

## Adaptive Security Algorithm

The Adaptive Security Algorithm (ASA) is Cisco's name for the subroutines, processes, algorithms, or techniques that perform stateful packet filtering and address translation.

The mechanisms for providing stateful access lists and performing address translation are combined as follows:

1. All data packets that initiate a flow across the firewall are analyzed. Critical information is extracted and placed in a state table—the xlate table. Then an address translation is made as described by the firewall policy. The translated address and details are also stored in the xlate table.

2. On the receipt of return traffic, the incoming packet is checked against the table. For the connection to be established, there must be a match between the two.
3. Once the connection is terminated, the connection information, including the session object, is eventually deleted.

The process is similar for other protocols, but where no state exists, timers are used to “tear down” the connection object.

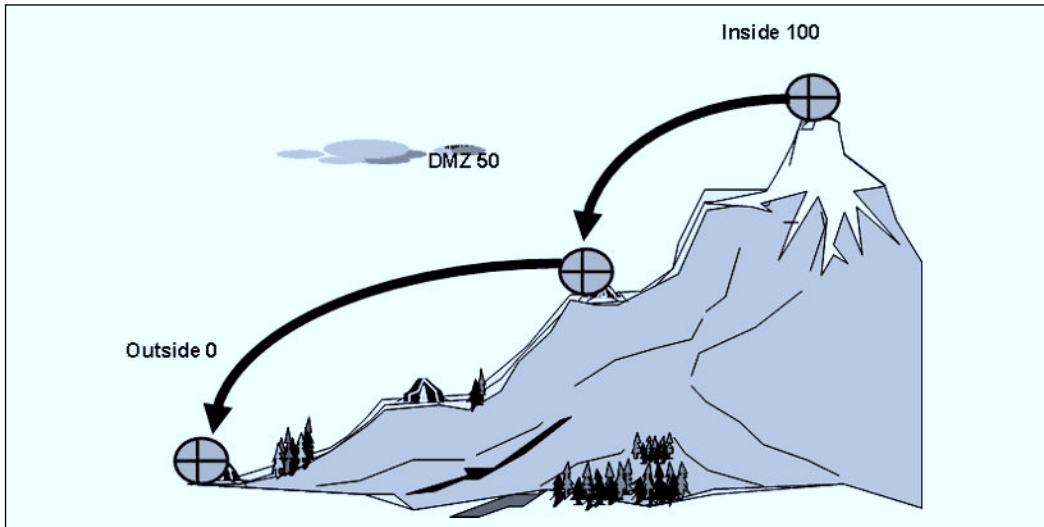
Nice and simple—*not*. There are a few more rules you have to be aware of before you can get data in and out. A few paragraphs earlier I mentioned that the ASA enforces security zones. These zones have a numeric value and are assigned to each interface—with no two interfaces having the same value (possible, but daft). Security levels range from 0 through 100:

- 100 is the most secure; default for inside interface and can't be changed
- 0 is least secure; default for outside interface and can't be changed
- 1 through 99 can be assigned to any other interface, such as a DMZ

Data can pass from lower- to higher-level interfaces but not from higher- to lower-level interfaces without an access list. Effectively:

- **Inside to outside (hi2lo)** Data traveling from more secure to less secure requires only an address translation—which may be static NAT, dynamic NAT, or dynamic PAT.
- **Outside to inside (lo2hi)** Data traveling from a less secure to a more secure interface needs both of the following to pass through: a static translation and an access list.

Figure 8.4, which I use in my training courses, really helps, showing how data can roll down the PIX mountain without any need for access lists. Of course, most people tell me I am a deranged. You make up your own mind—it's all good.

**Figure 8.4** Data on a Network Protected by a PIX Firewall

## Cut-Through Proxy

The cut-through proxy provides a method for user-based authentication. Both inbound and outbound connections can be authenticated. The method is superior to a traditional proxy filter because it uses fewer resources—no sockets are not terminated and reopened; the device never becomes an end-point. Instead, it monitors defined streams for authentication messages. When it spots one, it doesn't forward that packet immediately. Instead, it triggers the authentication mechanism, prompting the user for a user ID and password.

After authentication by a TACACS+ or RADIUS server, per-user connection state information is maintained by the firewall.

For protocols that don't support authentication, a virtual Telnet server exists. Just Telnet to a specific address and validate your user ID and password by signing on—then your PC's address will be authenticated for a specified time period.

## Failover

These days everybody is talking about five 9s availability (99.999 percent) and eliminating single points of failure. However, that is often easier said than done. The PIX for at least six years has implemented a scheme that is easily

achievable, effective, and manageable among a crowd of solutions that nearly always cost more and work less effectively. Okay, it's simple, but it does work (and anyone with any experience in disaster recovery planning (DRP) will tell you that is a recipe for success, not a reason to criticize).

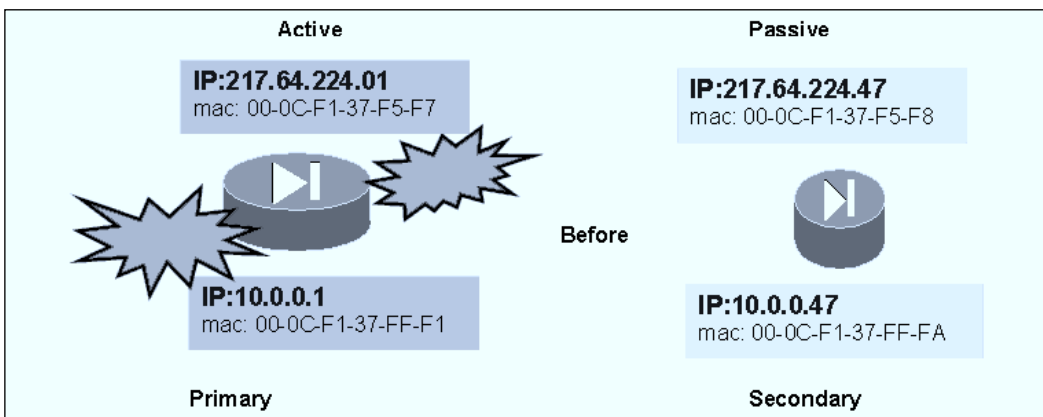
The PIX supports active/passive failover, which means that one is active and doing a job of work while the other is resting, ready to be called into action. Think of it like a football match, with reserves on a bench, or a famous stage actor who has a fresher, less battered understudy. The PIX also can operate in a stateful and nonstateful mode:

- **Nonstateful** In this mode, the secondary (one passive) firewall starts up automatically, but all active sessions are lost.
- **Stateful** In this mode, the secondary (one passive) firewall starts up automatically, but nearly all active sessions are maintained. This mode requires a third interface dedicated to replicating sessions to the alternate firewall.

In either case, the two PIX Firewalls must be identical models and identically configured. Whichever mode is chosen, these devices are connected by a special cable. When disaster happens, the devices swap IP and MAC addresses—then the passive/secondary continues on as the active/secondary (and vice versa).

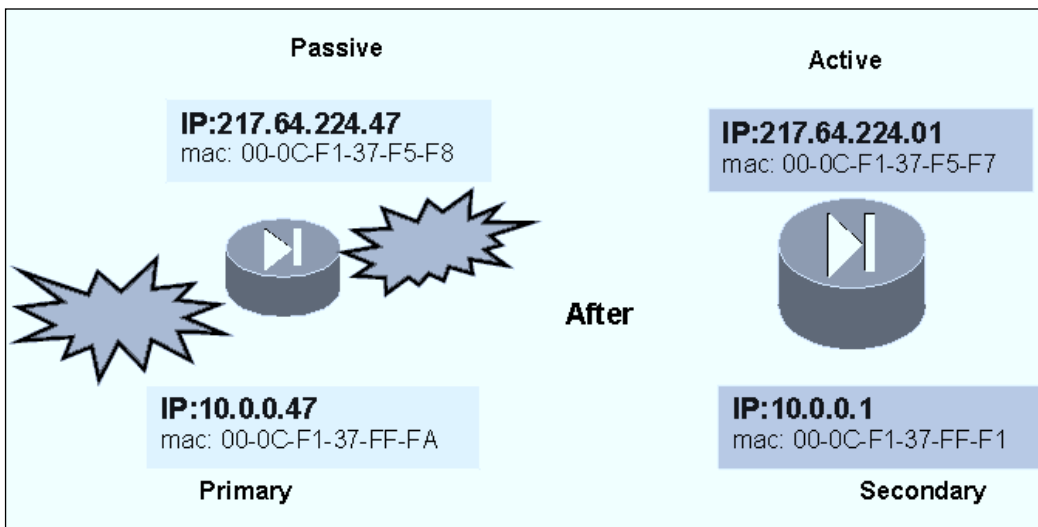
What was the active primary becomes a passive primary while it is repaired. In Figure 8.5, the firewall on the right is passive.

**Figure 8.5** Primary and Secondary Firewalls



When the primary fails, the secondary becomes active and claims all the addresses that used to be associated with the primary (see Figure 8.6). This technique for swapping everything (all interface MAC addresses and IP addresses) makes deployment simple.

**Figure 8.6** When the Primary Firewall Fails



To get a stateful situation (i.e., no loss of ongoing sessions), a special failover cable plus a dedicated Ethernet interface is required. This interface copies the session information from one firewall to the next.

## Configuration

The PIX firewall policy is devised around a simple command text very similar to that for routers. This makes it very popular with any organization (such as ISPs) that has a wealth of experience managing routers.

These commands are typed into a PIX or retrieved from a TFTP server just like a router. The following example is a very basic configuration:

```

pix(config)# interface ethernet0 auto
pix(config)# interface ethernet1 100full
pix(config)# nameif ethernet0 outside security0
pix(config)# nameif ethernet1 inside security100
pix(config)# ip address outside 81.2.94.93 255.255.255.240
pix(config)# ip address inside 10.0.0.2 255.255.255.0

```

```
pix(config)# route outside 0.0.0.0 0.0.0.0 81.2.94.81 1
pix(config)# global (outside) 1 interface
pixl(config)# nat (inside) 1 0 0
```

This configuration allows all outbound traffic. The commands have the following significance:

1. The *interface* command enables properties of an interface, usually speed.
2. The *nameif* command assigns a logical name to an interface. It also assigns a security value 0–100.
3. The *IP address* command adds an IP address to an interface.
4. The *route* command is similar to the Windows command that adds a static route; it performs the same function on a PIX.
5. The *global* command defines a global pool of addresses to use for address translation.
6. The *NAT* command defines a range of source addresses that could use a particular global pool.

There you have it.

## Check Point FireWall-1

Check Point FireWall-1 is probably one of the oldest firewalls around. It is also the most popular and the most feature rich. Ninety-eight out of 100 of the best technology companies in the world use it—the two that don't are Cisco and Microsoft!

FireWall-1 is a software-based firewall that is very simple to use and can be installed on many popular operating system platforms. However, in recent years there has been a shift from installing it on hardened Windows NT platforms to using it on a preloaded appliance—the most notable of which is the Nokia IPSO-driven appliance (IP330s are as common as muck) or the Nortel platform. But many people still install it on a Sunfire or Del D1380, and now that it ships with its own hardened Linux (Secure Platform OS), you will see more people spLATING (yes, that's the word they use) on to the platform of their choice.

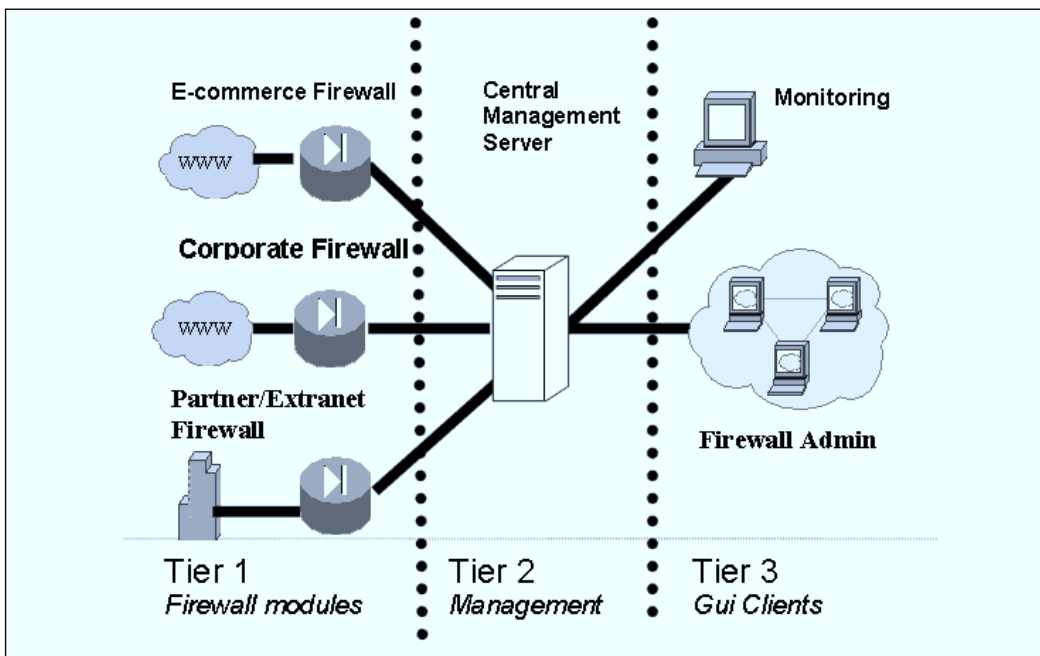
## How It Works

FireWall-1 is based on a three-tier architecture that includes:

- A number of desktop management clients
- A central management server
- A number of enforcement points or firewall modules

These components are shown in Figure 8.7.

**Figure 8.7** Check Point's FireWall-1 Architecture



### *Tier 3: GUI Clients*

Tier 3 contains the management graphical user interface (GUI) and so is the most instantly recognizable tier of all the three tiers. Although the GUI is just a front end for the other two far more functional tiers, we will use it throughout this section to illustrate the whole functionality of Check Point FireWall-1. After all, this tier is the bit you see.

## Tier 1: Firewall Modules

Tier 1 is the business end of FireWall-1—what actually stops the bad traffic. It does it very effectively, too. The operating system is still used to drive all the network functions, but a specific Check Point module is pushed (yes—on the old Unix versions it really did do an `ioctl`-push of a line module) onto the protocol stack between Layers 2 and 3 protocols. This inspects all traffic as it tries to progress up the TCP/IP stack. Should the packet fail the checks imposed by your particular firewall policy, the firewall logs it and dispossess of it.

The firewall module can test all layers from the OSI model, and if the tests indicate that the packet should pass, the firewall simply lets the packet progress up the stack, where the host operating system will forward it.

The speed is maintained because the firewall policy is actually a binary object that is interpreted by the Inspect module. This module is generated from a GUI on the administrator's PC and compiled on the management server, then pushed down to the enforcement point. This makes the business end very efficient. All log messages and alerts are sent to the management server.

There are very few user functions that can be performed from an enforcement point. You can, for example, load a policy manually, unload a policy (very useful if you lock yourself out), or display license details—and not much else. All the frills are handled by the management server.

FireWall-1 is generally a stateful inspection firewall, but it can run proxies or security servers. These need to be set up, for example, if you do virus scanning on SMTP or authentication of HTTPS.

## Tier 2: Management Server

The management server is typically a Windows box that

- **Controls generating, storing, and installing of all policies** A management server can manage well over a dozen enforcement points. Each one of these may have a separate policy. It is the management server that compiles these from the meta language that is used to control the GUI into a series of *inspect routines* and *preambles*. This internal language that pre-dates the GUI is then further converted into a binary and pushed down to the firewall.

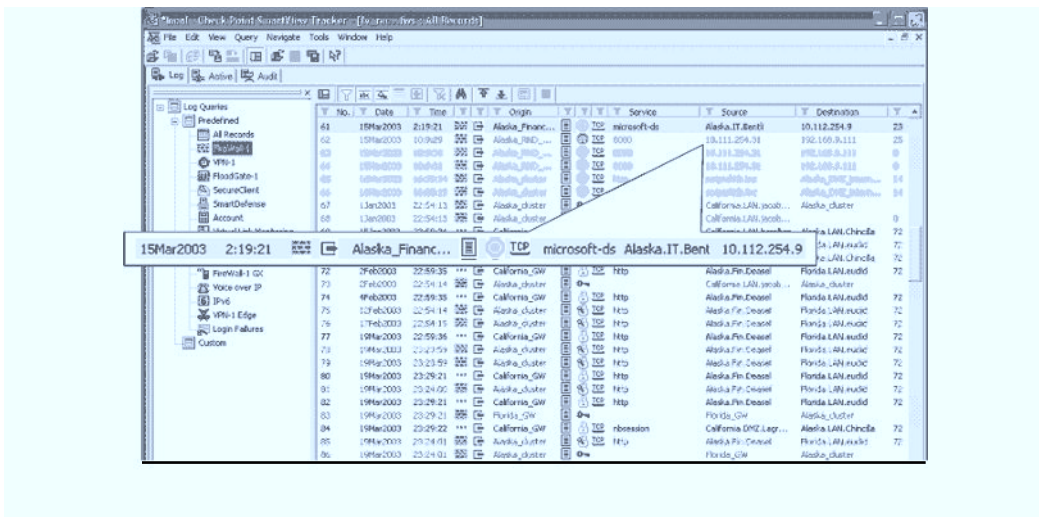


- **Holds all logs and manages alerting** All logs are sent from the firewall to the management server, where they are stored in a database. Should advanced alerting be required, it is executed here.
- **Stores all user definitions** FireWall-1 allows a wide variety of roles to be defined—from superuser to read-only auditor, they are all stored here.
- **Contains a mini-PKI** This ensures that communication between the components is secure.

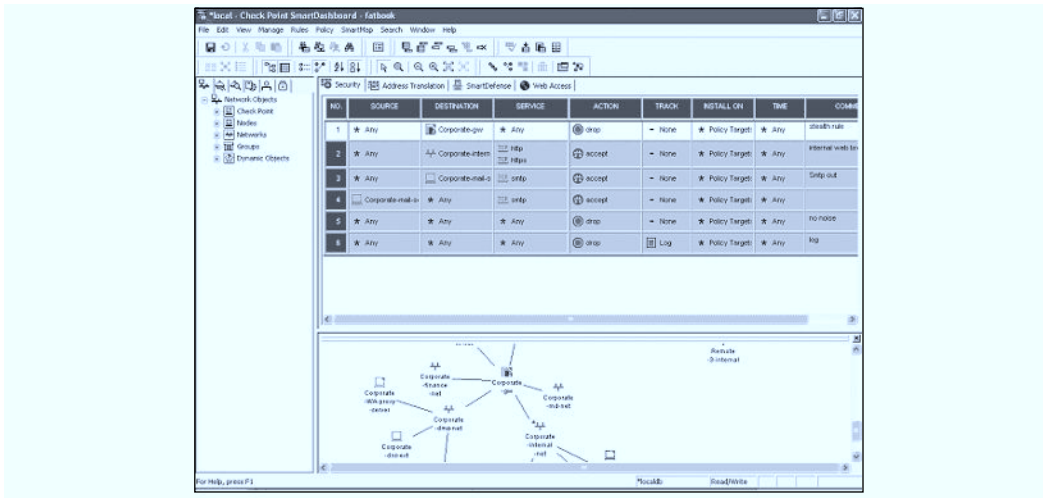
## The Gory Details

The Check Point Smart Centre GUI provides a pretty front end. There is a fairly unhelpful status monitor that shows what firewalls are up. There is also a really useful log-viewing application called SmartView Tracker (see Figure 8.8).

**Figure 8.8** Check Point’s SmartView Tracker



But the really useful part of SmartCenter or the GUI is the firewall policy editor. From the graphical user interface, you can load and display the firewall policy using the typical **File | Open** option. This will display a screen similar to the one shown in Figure 8.9.

**Figure 8.9** Check Point's FireWall Policy Editor

An expanded view of the policy is shown in Figure 8.10.

**Figure 8.10** An Expanded View of Check Point's FireWall Policy Editor

NO	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON	TIME	COMMENT
1	* Any	Corporate-gw	* Any	drop	- None	* Policy Target	* Any	stealth rule
2	* Any	Corporate-intern	TCP http TCP https	accept	- None	* Policy Target	* Any	internal web in
3	* Any	Corporate-mail-s	TCP smtp	accept	- None	* Policy Target	* Any	smtp out
4	Corporate-mail-s	* Any	TCP smtp	accept	- None	* Policy Target	* Any	
5	* Any	* Any	* Any	drop	- None	* Policy Target	* Any	no route
6	* Any	* Any	* Any	drop	Log	* Policy Target	* Any	log

These are the basic access lists and contain the following fields:

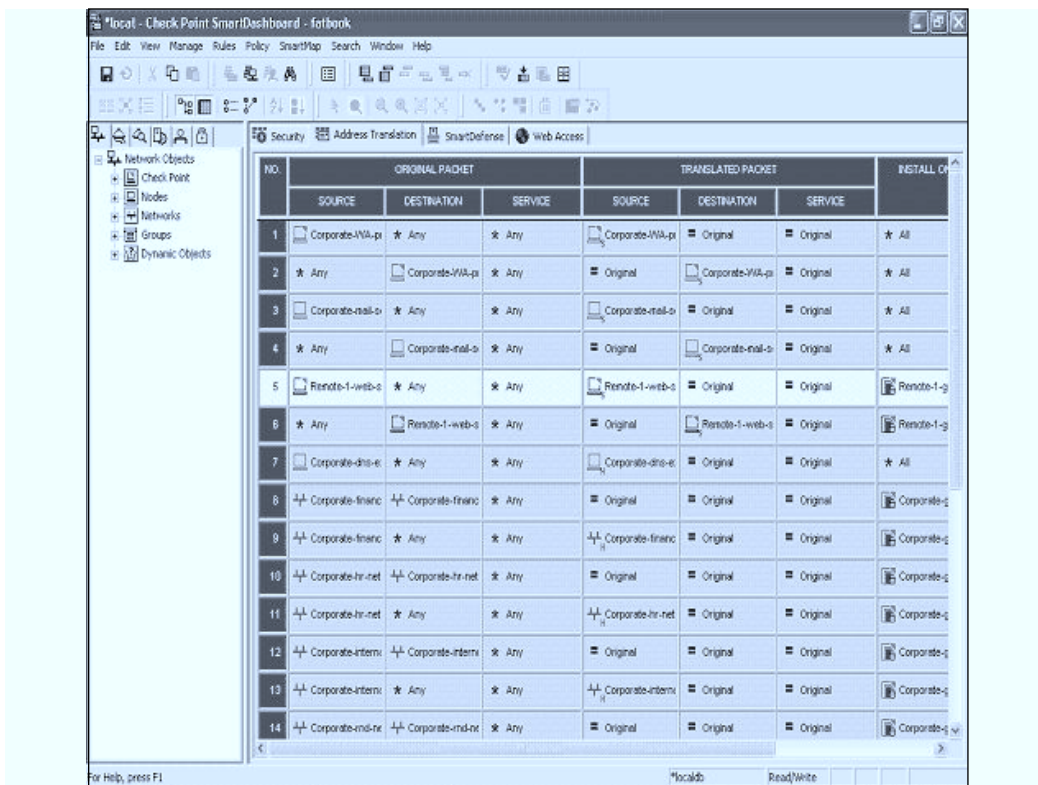
- Source object** All objects in FireWall-1 must be lovingly defined before they can be used in a policy. The source object is, obviously, the transmitting object, which can be a firewall, a network, an address, or an address range plus a group of any of the previous objects. You simply select it from a drop-down menu.
- Destination object** Like the previous object, only this is the destination. Like the source object, it must have been previously entered.
- Service** Here think port or group of ports, because like everything else in FireWall-1, objects can be created from other objects. For

example, there is a DNS group that consists of a combination of TCP/53 and UDP/53.

- **Action** This is typically accept, deny (disallow and inform by ICMP), or drop (disallow silently). It can also include options to encrypt or authenticate.
- **Track** To log or not to log, that is the question.
- **Install On** Specifies on which firewall to install a rule.
- **Time** This allows a time object so that you allow, for example, SCP to copy critical files across the Internet to a backup site on Friday night but disallow that service at any other time.
- **Comment** No comment required.

Address translation is handled in a similar way (see Figure 8.11).

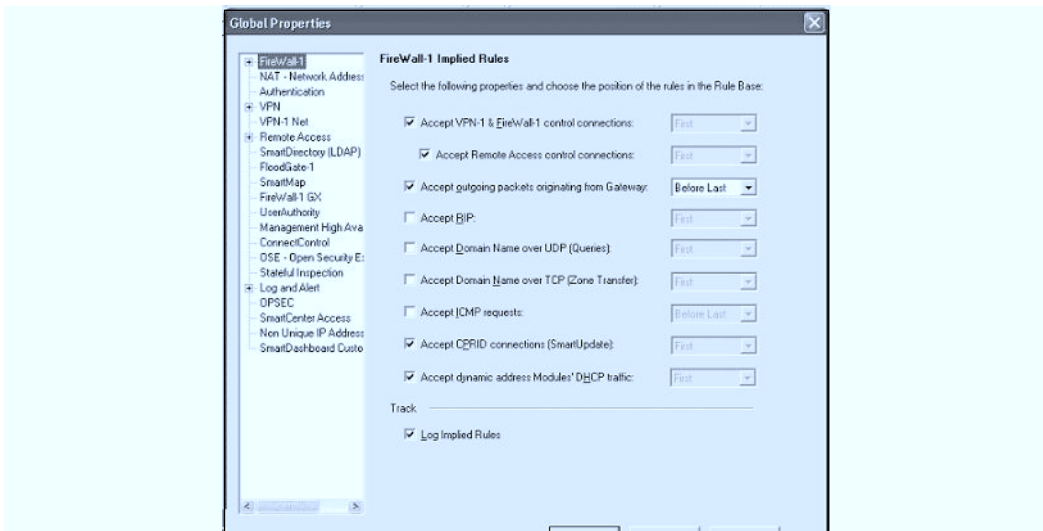
**Figure 8.11** Check Point's Address Translation



## Security Policy: Global Policies

Beware: a longstanding problem with FireWall-1 has been that you didn't just get the rules you type in. There are a number of default rules that affect the behavior of firewalls. These are found under the Properties button (see Figure 8.12).

**Figure 8.12** The Check Point FireWall-1 Global Properties



This screen is the most important of the Properties screens. The following is a brief explanation:

- **Accept FireWall-1 control connections** If this option is ticked, the firewall will accept management connections from a variety of locations. It is tightened up now, but even in recent versions it was possible to detect you were using FireWall-1 because this option was ticked.
- **Accept outgoing** This allows any outward conversations (from the firewall).
- **Accept RIP** Allows the routing protocol RIP through to the firewall.
- **Accept ICMP packets** Allow ping.

- **Accept UDP domain queries** Normal DNS queries.
- **Accept TCP domain download** Transfers from secondary DNS.

A later version allows you to view implicit rules (the rules generated from these options); it is well worth looking at them.

## SYNDefender

Like the PIX firewall, FireWall-1 has protection against basic SYN flood attacks. And as you would expect from FireWall-1, there is a rich set of associated options. Collectively they are known as SYNDefender and can be enabled from a GUI drop-down box (not shown for brevity). Key settings are:

- **SYN relay** Makes the firewall validate every connection before passing it to the original destination. Effectively turns FireWall-1 into a proxy.
- **SYN gateway** The firewall opens a connection to the original destination and spoofs the SYN-ACK response, then waits for the final ACK from the source before allowing a connection to take place.
- **Passive SYN gateway** Allows the first two stages of a TCP handshake to the original destination but monitors traffic until the ACK is received from the source. If none is received, it spoofs a reset.

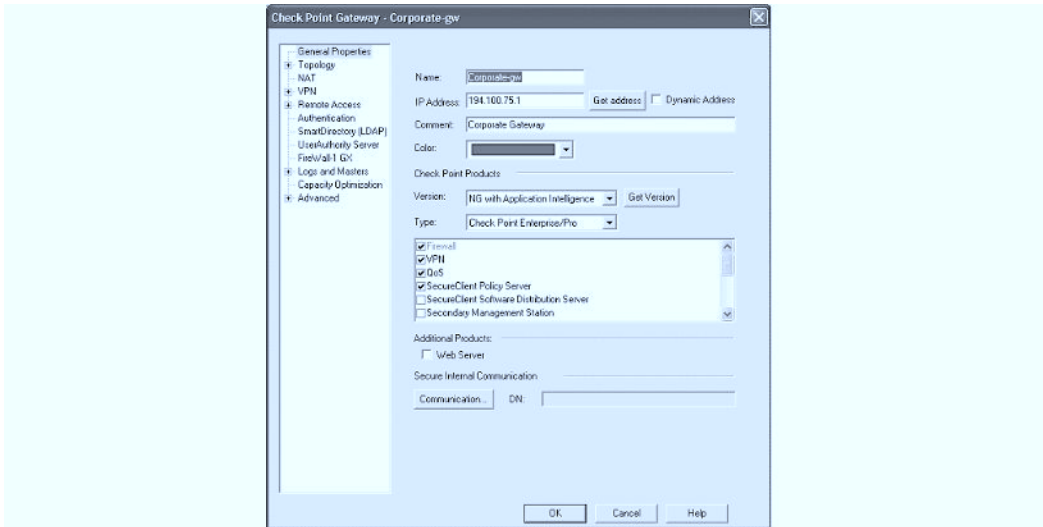
Other session variables include:

- **Timeout** Specifies the time the firewall waits for an acknowledgment before concluding that the connection is a SYN attack.
- **Maximum session** Specifies a maximum number of protected sessions that SYNDefender can handle.

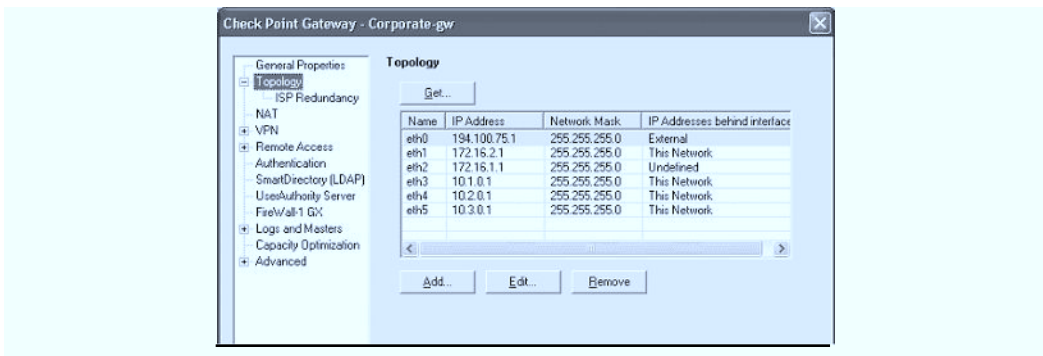
However, given the volume of traffic generated by a modern attack, this should be considered only as a bonus. Modern attacks regularly exceed 1 million PPS, well beyond the capability of most firewall CPUs to cope with.

## Antispoofing

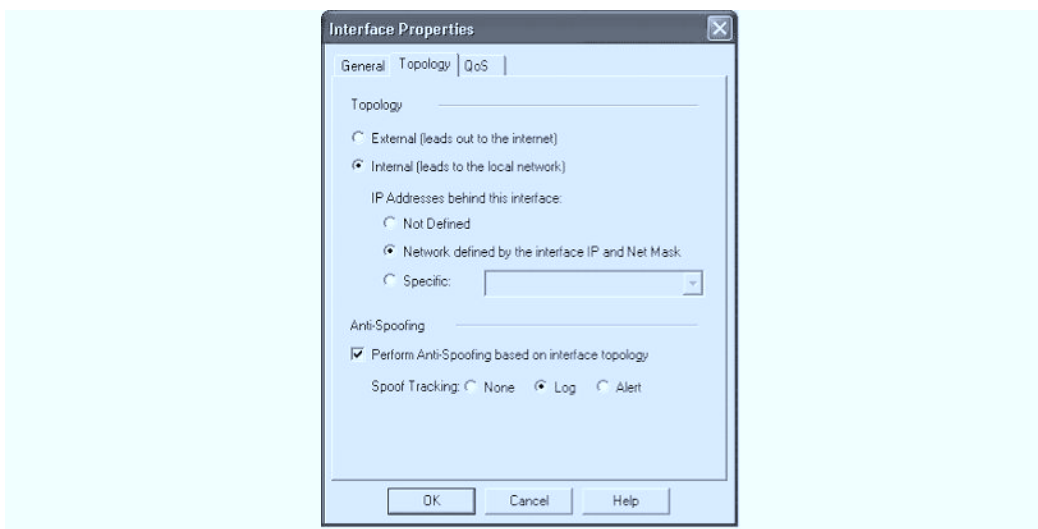
Viewing the properties of your firewall gateway will reveal the various options installed (see Figure 8.13).

**Figure 8.13** Viewing Check Point FireWall-1's Properties

The topology screen shows all the interfaces and the antispoofing rules defined on them (see Figure 8.14). This is absolutely key. FireWall-1, with no or inappropriate spoofing rules, doesn't make for a secure environment.

**Figure 8.14** Check Point FireWall-1's Topology Screen

Each interface can be set up as shown in Figure 8.15.

**Figure 8.15** Check Point FireWall-1's Interface Properties

To protect against spoofing, populate “valid address.” This can be set to **External—any /no checking**.

All internal addresses can have:

- **Not defined** No checking is bad.
- **Specific** You define a list of addresses as an object. FireWall-1 rejects packets with a source address that are contained in the list object. Typically, you create an object that would contain all your link addresses, external addresses, and 1918 addresses for this purpose. In earlier versions, this had to include both the internal and external addresses of the object where NAT was used. Since Version 4.1, only the internal addresses are required. So, if your LAN contained a bunch of 10.0.0.0 addresses and a bunch of 192.168.0.0, I'd create two network objects: one called *internal-10s*, defined as 10.0.0.0/8, and the other called *internal-192-168s*, defined as 192.168.0.0/16. I'd create a group called *internal-lan* and include both *internal-192-168s* and *internal-10s* within it. Then I'd set Specific to *internal-lan* using the drop down box.
- **This net** Rejects packets originating from any network that has an IP address that corresponds to the interface's network address.

## Summary

The Cisco PIX is a nice firewall that is easy and cheap to maintain. Its router-like appearance makes it a favorite of ISPs. Despite its robustness, it has a basic feature set.

Check Point FireWall-1 can just about do anything you'd want a firewall to do. Apart from normal firewall functions, it can block URLs, scan viruses, and even control access to particular shares on a particular Windows 2003 Server. It is easy to use and ideal for the enterprise environment. However, it is expensive, even in the enterprise environment, and phenomenally so in the ISP or provider space.

Understanding the functionality of a good firewall is key for a security manager—if only to maintain credibility with technical staff.



## Intrusion Detection Systems: Theory

The purpose of this chapter is to:

- Define the term *intrusion detection system* (IDS)
- Outline the common problems and techniques associated with IDSes
- Provide an overview of the major IDSes

## Anecdote

*Intrusion detection systems (IDSes) and packet sniffers do not replace your brain; you need to understand what they are telling you.*

*One day the Fat Bloke found himself dragged in to consult on a project for a huge bank buying a bit of another huge bank. The whiff of disaster was everywhere. The Bank of England (BoE) was demanding regular reports, and their top man gave my Big Six consultancy a call. Our best suits were mobilized; my job was to monitor how the team from a consultancy starting with A (commonly known as robots) joined the two networks together, providing encouraging, soothing sounds, if necessary, and to blow the whistle if the project looked like it needed to be cancelled (so that it could be done quietly, without public fuss).*

*In meetings before I arrived, the teams decided that they needed some firewalls, so some shiny new Nokias were bought to go into the racks next to the other shiny new Nokias. Now what to do?*

*In the first meeting I attended, they were struggling to decide what rules to put on the firewalls. “Well, we don’t really know what goes over the network,” the consultant moaned. So I told him to go find out. A couple of days later, he came back crying that nobody knows. In the next meeting they mentioned that they were doing a series of connectivity tests. I told them to log everything on the firewall and that would be a good start. The test happened that weekend, and afterward they discovered that the logs were set to “overwrite” themselves after 100MB of data.*

*I told them about packet sniffers. “I use Snort,” I said. Just for them, I installed it on a large Linux box and left them to play. The next test went off without a hitch. They were still surprised by the volume of the logs on the firewalls and on Snort, but they said they had enough to go away and design some firewall rules. I left them to it, glad I helped.*

*I then got a call from the network manager of the “purchased” bank, who was on notice (with a nice retainer) saying that I’d better come down. Off I trotted.*

*The meeting didn’t go well. With a grin on his face, the exiting incumbent slid a copy of the firewall rules across the table. Excited, clean-shaven faces stared back at me as I looked around; tired-looking pros stared fixedly at interesting invisible things on the wall and on their shoes. Here’s what I read:*

```
allow tcp 32098 10.0.0.1 to tcp 23 192.168.1.1
allow tcp 41011 10.0.121 to tcp 23 192.168.1.1
allow tcp 21994 10.0.102 to tcp 23 192.168.1.1
allow tcp 19467 10.0.0.64 to tcp 23 192.168.1.1
allow tcp 32098 10.0.0.1 to tcp 21 192.168.1.10
```

*This went on for pages. I looked at the old network manager, and he assured me this wasn't a joke. They didn't understand that DHCP would allocate workstation addresses dynamically and that TCP/UDP uses ephemeral client ports. I told them that I had to blow the whistle on them; any progression of the project in its current guise would ruin the bank's reputation to execute. In return, I was severely verbally violated by them; how dare I?*

*Obviously, my comments didn't go down well, but they were persuaded to inform the BoE that there was a delay, and the final dress rehearsal was rebadged as just another test. It went stunningly well for me. Two hours into the test, the bank's second best firewall expert, whom they had flown in from Zurich, could not get the firewall to work. Unfortunately, Number 2 could speak only French and Turkish; nobody else could speak Turkish and obviously their French wasn't much good either, as the expert had been frantically rebooting the wrong box. The scene set nicely, the proper firewalls were turned on, but the effect was similar: Nobody could even log in. The domain controller and DNS were on the wrong side of the firewall! Oh, life is sweet. My delay cost the bank several million pounds—but the failure of the integration would have cost tens or hundreds more.*

*The lesson is simple: IDS/IPS can help any network or security person understand network traffic. The modern devices can even brief you on new attacks and their mitigations that might have passed you by. But:*

- *You need to understand your IT infrastructure.*
- *You need to understand the mechanics of networks and applications.*
- *You need to understand the risks involved.*

## Introduction

What is intruder detection? An IDS is effectively the computer equivalent of a burglar alarm. Its main purpose is to provide a warning when a hacker breaches your security regime and accesses your IT system.

These days you'll find it hard to buy an IDS. You'll find it a lot easier to buy an *intrusion prevention system* (IPS). These are very similar to IDSes (in fact, they are often identical), but they have enhanced capability to respond proactively to attacks—to block *naughty* packets. So if an IDS is a burglar alarm, an IPS is a mantrap or a mousetrap.

## Why Bother with an IDS?

These days, an IDS is represented as legacy technology, especially after Gartner Group's ranting. Frankly, this is a bum rap. An IDS is a technology very similar to the firewall; in fact, many people suggest that IDSes exist because of inadequate monitoring firewalls. Unlike the firewall, the IDS was introduced into the turnkey software world of recent years. The early users of firewalls that used TIS FWTK or TCPD were prepared to take the time to make things work and had the low-level network knowledge to cut C code to make it happen, whereas many IDS users had just enough talent to install a default policy that came with the software and then complain ever so eloquently when it didn't produce great results. What would happen if you installed a default policy on FireWall-1—would you expect a secure network?

IDSes are worth studying for two key reasons:

- IPS (the new technology) is IDS with *muscle*. The problems and challenges of attack detection are *common to both*. This is exemplified by the inline Snort IPS, which is a normal Snort IDS glued onto a firewall.
- There are still many situations (because of ownership, legal, or regulatory requirements) where you want a *monitor-only* product—and surprise, surprise, most IPSes afford this option.

Lastly, most objections to IDS are really objections to monitoring; certainly the thrust of Gartner's reported argument was that products only reported on intrusions, and 99 percent of the time that doesn't add any value. The fact is that most people *do* think monitoring is a waste of time—until there is a fraud or a hack. Then you find that the forensic investigators charge you a fat fee and claim they can't give solid results because there is insufficient evidence. "No logging or monitoring, you see, Mate!"—I've said it myself. If

we applied the same logic to backups, we'd be in trouble. After all, 99 percent of the time, a backup doesn't add any value either.

## Problems with Host-Based IDSes

IDSes come in a number of varieties but can usually be divided into two categories: a *host-based IDS (HIDS)* and a *network-based IDS (NIDS)*. In host-based IDSes, software agents are installed on your key servers; these agents then watch for computer misuse and break-ins. The other major type, an NIDS, is the more popular and easier to deploy because it is positioned at key points of the network, inspecting passing traffic for signs of computer misuse and break-ins.

Although host-based IDSes are experiencing a bit of a comeback, it will be short lived. Let's discuss them first so that we can then deal with stuff that matters.

### Whether to Use a HIDS or Not? That Is the Question

Some problems are historical. Many host-based IDSes did not monitor a server in real time; they actually used a form of technology similar to Tripwire or the UNIX commands *bomverify*, *pdfchck*, or *tcbchck*. These commands compare a stored, known correct checksum with a newly generated checksum of key files to detected changes (a process known as *state monitoring*). Run at regular intervals (say, every half day), this method can provide notification of changes. This technique's limitation results in the reduced warning of hacking activity, which in turn provides less time for you to react.

For example, imagine a situation where you detect a hacker by discovering that a file has been changed in `/etc/rc5.d`. Great, you caught a bad guy when he changed something he shouldn't have. But this file almost certainly wouldn't have been the hacker's first choice of target. What about the dozen attempts to update `/etc/shadow`, `/etc/passwd`, and `/etc/hosts` that he would have tried beforehand? Your checksum-based HIDS failed to detect these attempts because the hacker didn't manage to change a file. Because the point of an IDS is to provide early warning of hacking, this approach is poor because it warns you only after the damage is done and an unauthorized

change has been made. With a typical gap of an hour between runs, the hacker could have time to really cover his tracks.

Being aware of this fact, you might compensate by scheduling the product to run several times an hour. A checksum analysis of the disk every 30 minutes will add significantly to the IO of a machine, probably equivalent to a nightly backup twice every hour. This means you often are asked to monitor a small community of files—but which?

The fact of the matter is, Tripwire, and its successors such as AFICK, Osiris, and AIDE, are great tools, but they are not IDSes.

## And Is It A Bad Thing?

To work properly, a HIDS needs superior event data acquisition using a kernel mod (i.e., LIDS) or by links into the audit subsystem. This way, the IDS is informed when things get changed or accessed or when users log in—and the IDS gets this event information in real time.

And here we hit a couple of major problems:

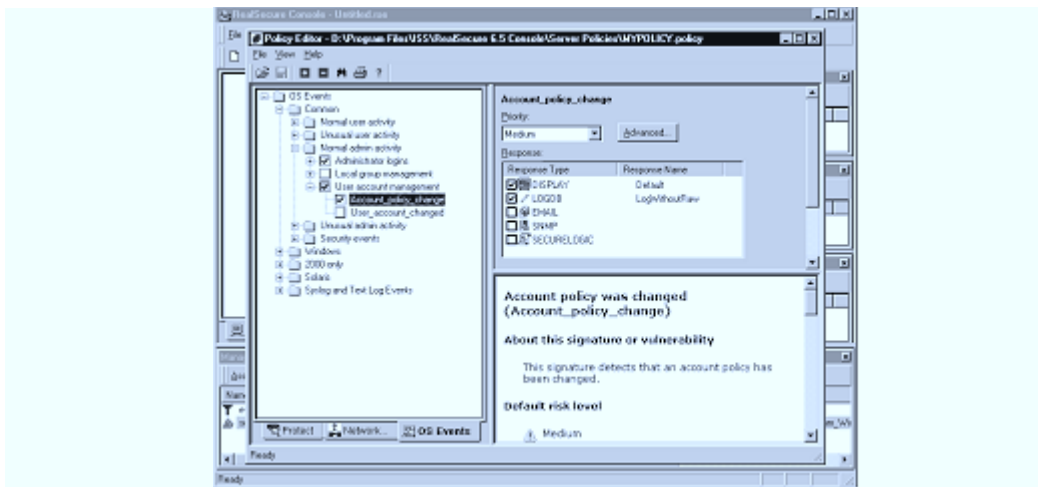
- Typically, enabling the C2-audit subsystem (which, in many UNIX systems, can be achieved by *Audit on* or *bsmconv*) causes about a 15 to 20 percent increase in CPU utilization. What happens if your servers are already running at an average of 92 percent? An extra 20 percent will kill them.
- Some Unix systems don't have the audit feature or, if they do, they place other security restrictions on the OS that will affect (break) other applications.

In at least a couple of cases, these facts have caused banks to swap the HIDS license for NIDS licenses.

But even where the IDS writers have used superior IO-level data acquisition, users complain that a HIDS added little over the operating system audit function, especially considering the cost. This is because there is just not enough granularity in the rules to allow the device to detect malevolent activity—to detect the difference between normal use and misuse. It might not be clear to the software engineers writing these packages, but *administrator logins* are

frequent events on most networks. See Figure 9.1, which shows a RealSecure sensor alerting every time an admin logs in.

**Figure 9.1** A Commercial HIDS



The HIDS needs to flag events showing when an event is abnormal. This may be when an administrator logs in from a strange workstation or out of hours, but certainly not every time the admin performs a normal job function from his or her usual workstation.

Recently, the way around this has been to add an event management console, a collation engine, or the newly termed *security management system*. This can add logic on top of the IDS system logic to get round these problems. The alerts are sent from the IDS/IPS to the collation engine, which does extra processing to decide whether it really should produce an alert—but it does cost more money and will not allow your host IDS to do IPS-type actions.

For all these reasons, I believe the HIDS is history.

## NIDS in Your Hair

Or was that nits? The fact is that both have caused some head scratching. When deploying or choosing a network-based IDS/IPS, you have to be aware of the common faults in the technology and the way these faults can be exploited or avoided. This will help you purchase a device that best suits your needs or allow you to minimize the effect of any less-than-perfect features of the technology.

There are three main areas where an NIDS or an IPS can run into problems:

- Detection flaws—a device misses attacks that it should pick up
- Poor deployment
- Poor configuration

We'll deal with each of these flaws in turn. It should be noted that the vast majority of these problems affect both network-based IPSes *and* network-based IDSes.

## Detection Flaws

Devices can fail to detect attacks for a number of reasons. The common causes are dropped packets, fragment reassembly, packet grepping, and lazy rule structure. Let's look at these in a bit more detail.

### Dropped Packets

The NIDS is particularly susceptible to performance problems since one sensor device has to receive, interpret, and react to network traffic created by many. This is a real problem for an inline IPS that has to perform tasks at wire speed; otherwise network latency goes through the roof and business suffers.

This situation can be easily rectified with faster technology, *if* you know about it. Many operating systems don't even report dropped packets in promiscuous mode (which is why many IDSes are implemented on the BSD-based operating system, which does), but even if your chosen platform does report dropped packets, some IDSes and IPSes ignore these events. And if the device you use doesn't warn you that you are dropping packets, you'll remain blissfully ignorant and not bother to upgrade your infrastructure. It is quite likely that the first indication that something is amiss will come when the IDS misses an attack and you get hacked.

According to reports, hackers have exploited this combination of relatively low traffic capabilities of some devices (compared to full-duplex gigabit switch networking) combined with no throughput alarm. It certainly is a possibility, and there are hacker tools that attempt to flood the network to take advantage of this weakness. However, to help dispel these rumors, there are no



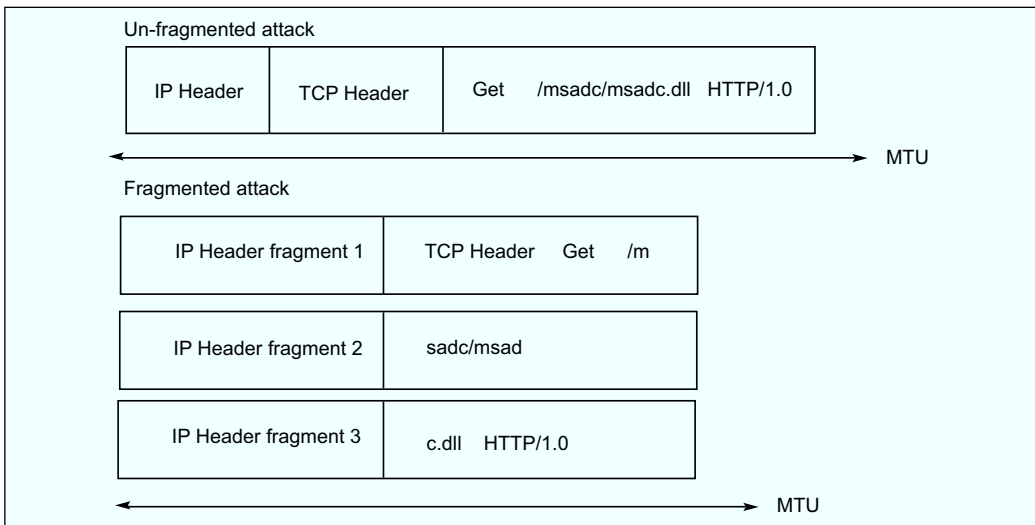
surefire ways that guarantee that attacks will go undetected; after all, which packet is dropped and which is processed is pure luck. If these attacks do work, it's probably because the poor analyst is moribund in log messages and has gone home with a headache. In the case of an IPS or IDS, flooding will simply cause the unnecessary detection features to be turned off, to help keep up with the throughput.

## Fragment Reassembly

Oh, no, it's basic network theory! Fragmentation occurs because some networks can transmit large packets and some can transmit only very small packets. This maximum size is known as the *maximum transmission unit* (MTU). The Network layer (IP) of the router between these two networks is responsible for taking one large packet and splitting (fragmenting) it into lots of small bits that will fit down the wire of the tiny network, as shown in Figure 9.2. These fragments will be reassembled and passed up to the Transport layer on the destination host, but in between, the datagrams are fragmented and may even arrive out of sequence.

This can cause some problems for applications that make assertions about the meaning of the data en route because one packet might not contain all the relevant data. A good stateful inspection firewall overcomes this weakness by rejoining the original fragments in memory and using this copy of the data to base its decisions on. This process is called *virtual packet reassembly*.

A good IDS/IPS must also use virtual packet reassembly; otherwise, it could miss important information. Imagine an IDS/IPS that worked by reading in a simple packet trace, one record at time, looking for indications of hacking. In the unfragmented attack in Figure 9.2, if it searches for the string *msadc*, it will find it with no problems. But in the fragmented example, the string *msadc* doesn't exist in its entirety in any single packet, so the IDS will miss the attack. Dealing with single fragments will not detect all attacks.

**Figure 9.2** An Attack in a Packet Fragment

Enterprising hackers have become aware that some products do not cope with this situation too well and have created proxies that can take a data stream from TCP or UDP attacks and fragment the packets to avoid detection. The most famous of these is Fragrouter, which is reasonably effective.

**TIP**


---

Make sure that packet reassembly is enabled on your IDS/IPS.

---

## Packet Grepping versus Protocol Analysis, or Just Not Working Right

For a network-based IDS/IPS to work effectively, it has to understand and interpret the data sent in exactly the same way to the destination server and service. This is a “warts and all” relationship that must take into account the reality of how the real servers behave with strange input. If malevolent commands can be sent to servers and result in an exploit, the IDS/IPS must recognize that it as an attack, even if the data is not constructed to the correct

protocol specification or breaks the rules. This is not a cricket match; the IDS/IPS is there to keep the bad guys out.

In some cases, it is as simple as recognizing that some servers will treat *get /etc/passwd* as equivalent to *GeT /etc/passwd*. If the endpoint server is smart enough to adjust the case of the *get* command and remove leading spaces, that's exactly the behavior the IDS/IPS should have when interpreting the attack. Many technical books refer to this as *protocol analysis* (interpreting the packet according to the protocol). This is opposed to just searching each packet for a unique attack signature, oblivious to protocol or any other context. This latter technique is known as *packet grepping* (*grep* being Unix-speak for *find*).

For me, if the IDS/IPS doesn't pay appropriate regard to the context of the protocols in transit, it just isn't working right. Table 9.1 lists a number of techniques that highlight the problem.

**Table 9.1** Avoidance Techniques

Avoidance Technique	Description
Port scanners	Many IDSes/IPSes recognize a port scanner after two or three ports per minute. In such a case, if you use IDS avoidance options with a tool like MingSweeper, your scans will not register. If you really want to defeat most IDSes, spoof your source address and alter it for each port. Because most IDSes collate their port scans by source address, this shouldn't even trigger an alert.
Encoding	For a properly working Web server: <pre>GET /cgi-bin/ HTTP/1.0</pre> is equivalent to: <pre>GET /%63%67%69%2d%62%69%6e/ HTTP/1.0</pre> A simple text match could miss this.
////////	These exposures are not restricted to www. FTP servers have peculiarities that can trick the unwary, such as: <pre>GET /etc/ passwd /tmp/attackinfo</pre> which s equivalent to: <pre>GET //////////etc/passwd /tmp/attackinfo</pre>

Continued

**Table 9.1 continued** Avoidance Techniques

Avoidance Technique	Description
<code>././</code>	<p>Everyone knows that <code>./</code> means <i>in this directory</i>, so that:</p> <pre>././cgibin/testcgi</pre> <p>is equivalent to:</p> <pre>/cgibin/testcgi</pre> <p>Simple, but it can fool many an IDS. It is also a nice little resource consumption attack. Try it and notice how your response times drop.</p>
<code>\cgibin\testcgi</code>	<p>Some Web servers don't care what delimiter you use to denote a directory structure. So:</p> <pre>\cgibin\testcgi</pre> <p>is equivalent to</p> <pre>/cgibin/testcgi</pre> <p>This doesn't work on IIS servers, but it works on many others.</p>

## Tools & Traps...

### Avoidance Techniques

The code in this sidebar shows how avoidance techniques have been implemented in real life in common hacker tools (**note: proxy/bounce support has been removed until v2.0**):

```

$./whisker.pl -?
-- whisker / v1.4.0 / rain forest puppy / www.wiretrip.net --

-n+ *nmap output (machine format, v2.06+)
-h+ *scan single host (IP or domain)

-I 1 IDS-evasive mode 1 (URL encoding)
-I 2 IDS-evasive mode 2 (./ directory insertion)
-I 3 IDS-evasive mode 3 (premature URL ending)

```

Continued

```
-I 4 IDS-evasive mode 4 (long URL)
-I 5 IDS-evasive mode 5 (fake parameter)
-I 6 IDS-evasive mode 6 (TAB separation) (not NT/IIS)
-I 7 IDS-evasive mode 7 (case sensitivity)
-I 8 IDS-evasive mode 8 (Windows delimiter)
-I 9 IDS-evasive mode 9 (session splicing) (slow)
-I 0 IDS-evasive mode 0 (NULL method)
```

## NOTE

If you are worried, use IDS Workbench, ADMutate, or Blade Software's IDSinformer to test your software.

## Tools & Traps...

### Decoders and Reassembly Routines

Make sure you enable any decoders and reassembly routines needed in your IDS. An example from Snort is shown in this sidebar. Pay special note to options for HTTP and Unicode processing:

```
$ more snort.conf
# http_decode: normalize HTTP requests
# -----
# http_decode normalizes HTTP requests from remote
#           machines by converting any %XX character
# substitutions to their ASCII equivalent. This is very useful for
#           doing things like defeating hostile
# attackers trying to stealth themselves from IDSes by
#           mixing these substitutions in with the request.
# Specify the port numbers you want it to analyze as arguments.
#           unicode           - normalize unicode
#           iis_alt_unicode   - %u encoding from iis
```

Continued

```
#         double_encode    - alert on possible double encodings
#         iis_flip_slash   - normalize \ as /
#         full_whitespace  - treat \t as whitespace (for apache)
#
preprocessor http_decode: 80 unicode iis_alt_unicode double_encode
iis_flip_slash full_whitespace
```

## Lazy Rule Structure

A few years ago, I decided to find out why I was getting so many false alarms on a well-tuned policy. I was working on the best-selling NIDS positioned on a well-secured network, so all the data was false-positive because no hacking was occurring. But from the data, it was clear that most of the problems occurred from the nonspecific nature of the rules; by *nonspecific*, I mean not related to the network inventory and network typology.

For example, on this site they had IIS and Apache servers, so both sets of rules were enabled. However, this meant that if an IIS exploit was directed toward the Apache server, IDS would raise the IIS alert. Don't forget, IDSes were designed to detect attacks, not necessarily *successful* attacks.

And no, the open-source bible thumpers can't feel smug here either. Even if you bother (most don't) to tailor the config of the fabulously flexible Snort and accurately set the variables `$HTTP-SERVER` `$HTTP-PORTS` to the correct values, the situation I described will be true in a multiple-Web-server environment. The result is that every IIS-related *MSADC* & *Jill* attack that is misdirected to an Apache server will result in a high-priority alert.

The situation is still worse with UDP datagrams. In this case, Snort will fire an alert for an attack on a server that doesn't even exist. This kind of false alarm represents the majority of the alerts most IDSes produce. My I-am-Doh project proved that a simple pre- and post processor could significantly reduce the false-positive alerts by about 75 percent—a clear case for event correlation and targeted IDS/IPS, sometimes known as *virtual patching*.

## Poor Deployment

A poor workman always blames his tools, but even the best chisel will become blunt if you use it as a screwdriver. Likewise, even an excellent IDS

might not react to significant attacks because of their environment and intended use. This section examines brief details of the problems inherent in poor deployment; more technical details are available in a later section.

## Switches

The advantage in using switches is that traffic between any two servers gets a nearly dedicated channel; it doesn't share. This is opposed to a hub, where servers share the channel between each connected server. A NIDS relies on eavesdropping on traffic passing through a shared segment such as a hub. Plug a NIDS into a switch, and it will only see broadcast traffic or its own control traffic.

Obviously, this is a fundamental problem, but fortunately it's not an insurmountable one. If you need to deploy a NIDS, you have the following choices:

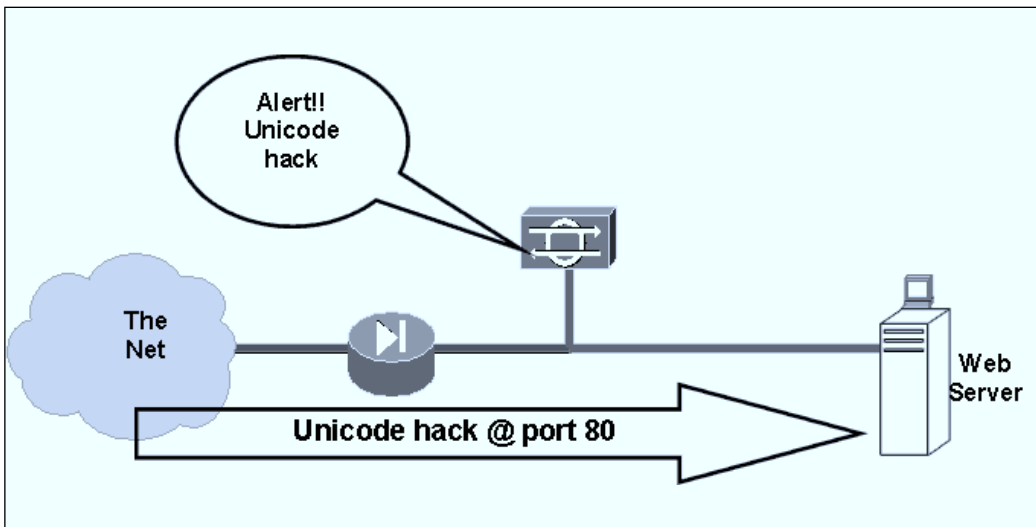
- **Use a spanning port** A *spanning port* is a dedicated port that is linked to one or more normal data ports on the switch. Each time one of these data ports receives a packet; the spanning port receives a copy. This works perfectly satisfactorily on small underutilized networks with powerful switches. On larger networks, the overhead on the switch might cause problems, plus most switches have restrictions on the number of ports that can be bound in this way.
- **Use tap technology** A *tap* is an inline device that provides a method of directly viewing traffic on a full-duplex or half-duplex 10/100 Ethernet segment. Typically, the tap is a hardware device that copies the electronic signal, like a Y splitter for a TV aerial. A tap adds virtually no delay to the network but can rapidly leave you with a huge tangle of wires.
- **Use a NIDS switch card** Some switch manufacturers also make IDS software that integrates with the switch by capturing data from the "trunk" or data bus, providing minimal performance impact. These switch cards are very popular for switched installations but do require some complex configuration. Currently, these devices are available from Cisco and Crossbeam.

## SSL and Encryption

SSL and encryption have always been among our best weapons in the security armor. However, they cause more than a few problems for designers attempting to deploy NIDS. Consider the attacks already described in the previous section. Those that involve contacting ports and the services behind them, which are now categorized by those that like to categorize things as *context attacks*, can usually be identified by the content of the packet header. Firewalls, which are generally far better configured than they were five years ago, block the majority of these. They are also completely unaffected by SSL, which only encrypts data payloads, not packet headers.

*Content attacks*, where the attack is contained within the data payload, are another story. My feeling is that these attacks are now the main focus of any intruder—and they are far more deadly. In a normal case (see Figure 9.3), such as an HTTP-based attack, IDS can detect these.

**Figure 9.3** Typical Attack with SSL

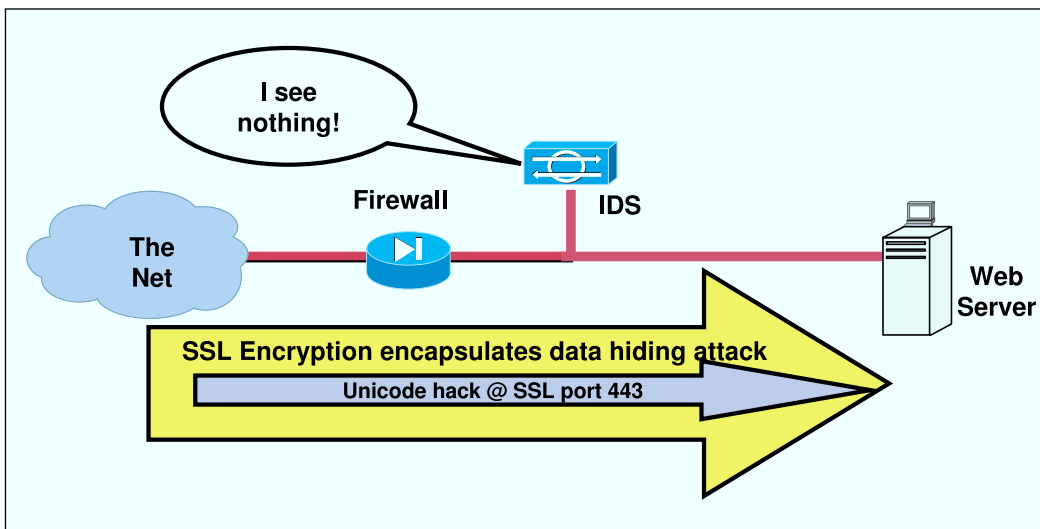


Unfortunately, because SSL encrypts the data payload of a packet, this blinds the NIDS. Consider all e-commerce and banking apps on the Internet;

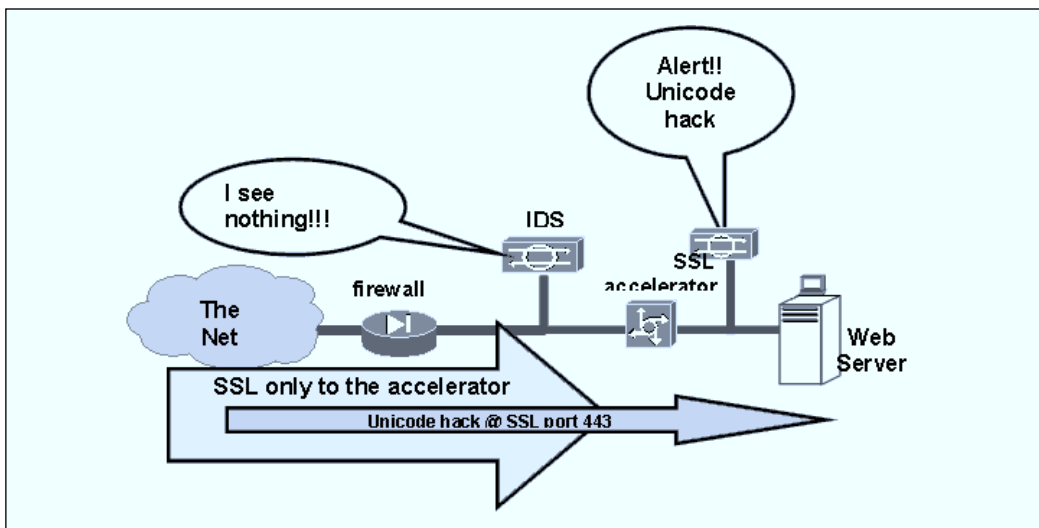


they are all encrypted because they are most critical, but they are just as vulnerable to *JILL* as any other app. Our NIDS can't tell us when such an attack is being launched, because the attack is wrapped in a 128-bit encryption envelope. In short, most of the launched attacks are being missed because of our own security mechanisms. See Figure 9.4.

**Figure 9.4** An Attack Hidden within SSL



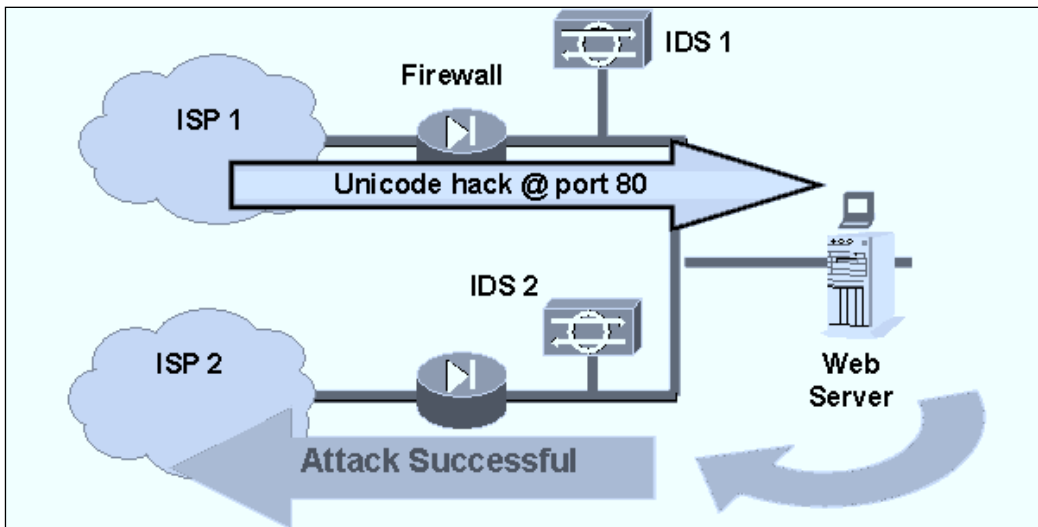
The only solution I can come up with for this problem is to use an SSL accelerator as the encryption endpoint, as shown in Figure 9.5. This solution does have a few disadvantages, but it allows the network for malevolent traffic to be detected. The main disadvantage is that use of client-side certificates can become very difficult, since the Web server never receives the details. Again, this problem affects inline IPS and sniffer-based IDS.

**Figure 9.5** Detecting an Attack with SSL

## Asymmetric Routing

*Asymmetric routing* occurs when a packet can send the reply to a packet via a different route; this often occurs where the network designers want to provide resilience or balance the network load (see Figure 9.6). This method causes problems for some more advanced IDSes that look at the response that servers generate from attacks. In the preceding example, the sensor IDS 1 sees the initial attack, but it doesn't see the appropriate response because the return packet follows a different route past IDS 2, so the sensor does not raise an alarm—it causes a false negative.

This affects any IDS that does analysis on the state of the packet, and particularly any inline IPS. The ISS Proventia appliance overcomes this problem by having multiple port capability and aggregating traffic on those ports. However, if route diversity has been created to provide resilience and load balancing, this solution might not be ideal.

**Figure 9.6** Asymmetric Routing

## Poor Configuration

When I have been called to help recover hacked systems, the sites often had an IDS installed. So why did the IDS not discover and help the site prevent the hack? I think there are three main reasons:

- The SSL problem above is very significant.
- The staff's reaction is often ineffective.
- The IDS detection method is deficient.

There are two main types of detection: signature analysis and anomalous traffic detection.

## Signature Analysis

Most IDSes and IPSes these days use attack signature detection. This technique is very similar to the approach used by virus scanners. Key patterns are identified in the data and header for a common attack. A pattern might be found in a particular flag/ID in headers or even a URI string in the data. The IDS sits on a network sniffing packets and reacts when the data matches one of these patterns. Given that the IDS designer is not incompetent, this can be a remarkably effective way of looking for at least the initial overtures of a hacker.

But here's the crunch: Does the IDS detect successful attacks, where an intruder invades your network? Or does it *just report on attacks that should have already been patched anyway?*

Let's face it. In most cases, you will have to configure your NIDS to look for, for example, the Unicode hack. So you'll configure your IDS to look for a command such as:

```
GET
```

```
/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\inetpub\wwwroot
```

which, if successful, will allow you to execute arbitrary commands on an IIS server.

But the key point here is that you must be aware of this hack to set the IDS looking for it. For this to happen, in most IDSes you'll have to have downloaded a new attack signature and, as a security professional, you would have read about the subject on the CERT and Bugtraq Web sites, right?

But before you did all this work, you would have almost certainly made arrangements to install any available patches to critical servers or introduced other countermeasures such as blocking ports at the firewall. This should make you immune to this hack, so when the hacker tries it and the IDS goes beep, what have you learned? Only that someone launched an attack to which you were immune.

Not convinced yet? Well, think about the first time an attack occurs "in the wild," before anyone has designed a signature for it. How does the signature IDS help here? (What would you say about a burglar alarm that didn't go off when a first-time offender broke in?) This question highlights the need to choose an IDS from a vendor that produces signatures with minimal lead time and designs internal procedures to ensure that signatures are updated either automatically on issue or manually, but in a timely manner.

Experienced practitioners like us know that for large organizations, patches can take some time to get deployed. Standard builds will have to be amended to avoid regression, and there are always those independently maintained servers. And what if the patch doesn't work with all combinations of your software, so you can't patch your servers? A NIDS pays you back in this instance in the following ways:

- It provides advanced notice that a perpetrator is targeting a particular server with an attack. And these days, with 80 percent of attacks being content attacks, where the attack is contained within the data payload, rather than context attacks, this might not be detectable with just firewall logs.
- You also get the opportunity to be proactive in reacting to the attack. If no patch is available, you may choose to reset the connection or shun the source address. This technique can provide adequate protection. Those with NIDS will remember that such techniques really saved their bacon when SYN flooding was a bright, shiny attack.

But however useful an NIDS may be, it is not detecting an unwanted invasion of your network; it is not intrusion detection—it is monitoring your networks for certain attacks. We need something more!

## Anomalous Traffic Detection

This extra something is *anomalous traffic detection*, and it works by triggering an alert when traffic that should never occur is detected. This traffic might not always be evidence of a hacker; it could be a Web server administrator making changes in a nonstandard manner—by, say, not using the staging server. However, this type of detection is not understood and is much maligned.

Many vendors have poo-pooed the concept of anomalous detection by implying that:

- Most networks are too diverse to baseline.
- Training time will be too long.
- It will produce too many false positives.

Together, the last two arguments seem bizarre. Have IDS vendors ever used their own products? Usually, most sites have to spend a large amount of time tuning the IDs and still are left with an unacceptable level of time-consuming false positives. The fact is, defining every single unused protocol would be time consuming, but defining a set of anomalous conditions that would not be normal and would be typical of a hack is a simple task. Such a profile will significantly improve your chances of catching a hacker.

## *Anomalous Detection Profiles*

Read a typical description of a hack. When the hacker finds he has the capability to execute commands on your Web server, the hacker will try to download his favorite tools, often with TFTP. If he can't, he will try to gain access to other servers or the firewalls. This makes failed Telnet attempts from your Web server to your firewalls or external TFTP traffic worthy of investigation, even if they turn out to be false alarms.

Consider vendors' assertions that most perimeter networks are too diverse to baseline and review the following points:

- Typically, when you're designing a DMZ, one or more firewalls are positioned to behave as choke points to restrict network traffic. Also, it is commonplace to control this by firewall rules that exactly define what is allowed into/out of the segment. Traffic passing through these points that doesn't conform with these rules would certainly be some kind of unusual.
- A good designer would also provide for maintenance access from an administrative LAN or designated management server, hopefully via SSH or IPSec. So, maintenance access such as FTP, Telnet, and Rlogin between servers or the firewalls would be a good indication that an external party might be attempting to escalate his or her zone of control.
- If all else fails, a few UNIX commands such as `tcpdump | cut | sort | uniq`, slightly enhanced with the correct command-line augments (forgive me, but I have left them out for the sake of brevity), would provide a nice little summary of what occurs in the DMZ.

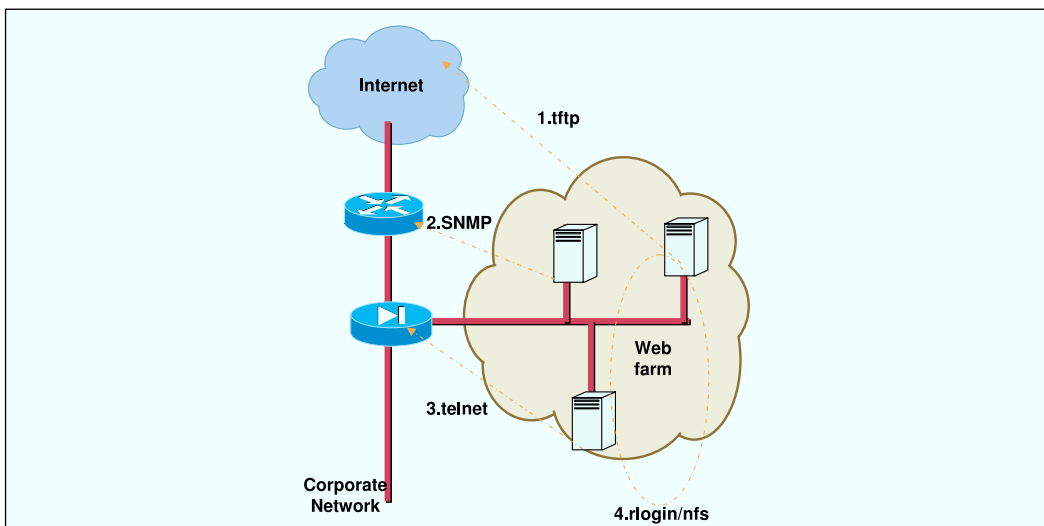
## Tools & Traps...

### Developing an Anomalous Traffic Profile

So when you have you have a good idea of normal traffic, you can develop an anomalous traffic profile. But don't fall into the trap set by the NIDS manufacturers; you are not trying to document every unused port that someone could inadvertently contact. Focus on hacker-favored protocols. In a typical e-commerce configuration, you would know (see Figure 9.7):

- TFTP or FTP traffic would not normally be initiated from a Web server to external random addresses.
- SNMP traffic from application servers to perimeter routers is unusual.
- Telnet access would not normally be initiated from inside the DMZ, particularly not to your firewalls.
- Site-banned protocols such as Rlogin, NFS, and Netbeui should not simply appear in your DMZ.

**Figure 9.7** Interesting Traffic for an Anomalous Traffic Profile



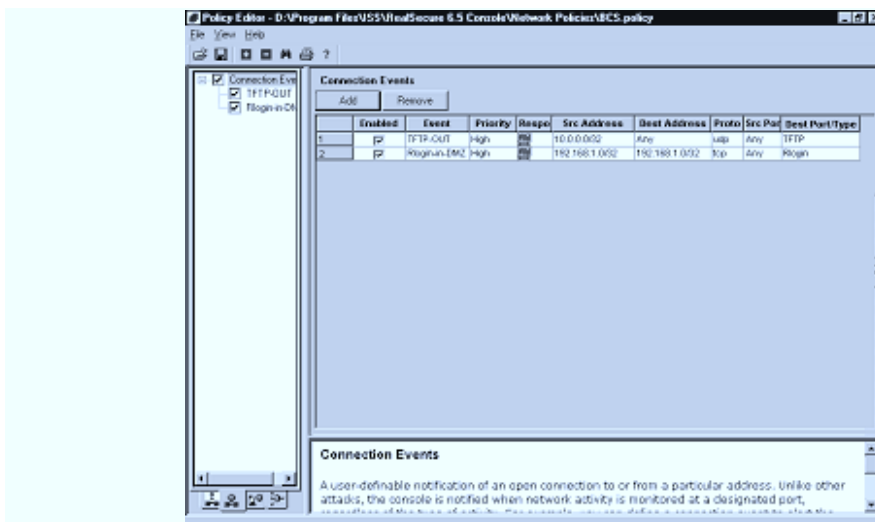
Simply code these assertions into your IDS (using the Connection tab or a combination of connection signatures and filters) and you'll be in a position to pick up all sorts of interesting stuff. Figure 9.8 shows an example set of Snort rules that do exactly that.

**Figure 9.8** An Anomalous Traffic Profile Defined in Snort

```
#var/snort/snort.conf
# use options -d -o
Var DMZ 192.168.1.0/24
Var MY_NET 10.1.1.0/24
# Pass or ignore any exceptions
pass udp $DMZ any -> 192.168.1.16 161
# Telnet from inside DMZ back into the Internal Net
Alert tcp $DMZ any -> $MY_NET 23 (msg: "Suspicious telnet")
# TFTP from DMZ out to the internet or between dmz hosts
Alert UDP $DMZ any -> ! $MY_NET 69 (msg: "Suspicious TFTP")
# Snmp from DMZ out to dmz hosts
Alert UDP $DMZ any -> $DMZ 161 (msg: "Suspicious Snmp")
```

And for those who prefer ISS RealSecure, see Figure 9.9.

**Figure 9.9** An Anomalous Traffic Profile in RealSecure





# For the Technically Minded

For the more technically minded reader, here I provide some details of the two most popular IDS products: Snort and RealSecure.

## Snort

The Snort IDS is a very popular open-source network IDS developed by Brian Caswell and Marty Roesch. I am a great fan of it and really find it hard to do it justice in such a short write-up. However, because Snort has emerged from packet-sniffer roots, it is fair to say that it is a very simple product. Also note that a huge amount of effort has gone into its development, so, even as I write, I know most of the shortcomings will be addressed in later versions.

Snort has a very flexible rule base that is updated regularly, but some of the rules are not brilliant. For example, some do little more than report when a service attempts a connection to a well-known port. Because these are simple text-based rules, it is easy to preprocess them to weed out the poor ones with a bit of shell script or to filter the alert in output. A typical rule is shown here:

```
Alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS
$HTTP_PORTS (msg:"WEB-IIS msadcs.dll access"; flow:to_server,established;
uricontent:"/msadcs.dll"; nocase; reference:cve,CVE-1999-1011;
reference:bugtraq,529; classtype:web-application-activity;
sid:1023; rev:7;)
```

This rule demonstrates that Snort does perform a form of protocol analysis, but it has been introduced in recent versions and it is a little on the immature side. It certainly does not analyze the response of a suspected attack and therefore cannot be accused of performing advanced protocol analysis. (As a testimony to these slightly negative statements, see `attack-response.rules`—statements guaranteed to produce false positives.)

Regular expressions (an absolute must for interpreting SQL injection and application-level attacks) have been available as an option for a while but are now fully supported.

Other nice features of the rule structure include the ability to reference them back to online documentation (through `reference.config`) at CERT or

Bugtraq. You can also set a centralized classification and priority for various alerts (through `classification.config`).

Another feature that holds great promise is Snort's *activate/dynamic* feature. This feature uses one rule, the *activate rule*, to define malevolent traffic. The subsequent *dynamic rule* can be used to log a predefined number of packets from the original host. This means that after an attack, you have a complete session trace—a feature available in only a few commercial IDSes. But with a bit of fiddling, this can even be used to set up a basic DEFCON scheme so that your IDS automatically increases its monitoring levels. Normally, a sensor will run with a low level of monitoring in place until the activate rule triggers a more rigorous set of rules.

Snort historically is not able to do any active IDS responses (this functionality is being introduced), but these are available from add-ons such as Inline Snort and other scripts. This has got to be one of the strengths of the product; someone, somewhere has developed an add-on to do everything. However, in the raw form, it can alert to a log, a database, or syslog; the latter is the most used because it provides a simple means of monitoring the organization.

To deploy Snort, you need to tailor `snort.conf`:

1. Tell Snort your various network addresses and the addresses of your Web servers, mail server, and various other ports. Tell it the path of your rules directory and the network interface to be used as a monitor.
2. Enable various detection options: decoders to detect scans, fragment reassembly, or HTTP processing.
3. Choose the output for alerts.
4. Enable a series of rule includes, just like a C program. Care here will pay dividends.

To start Snort, type the following at your command line:

```
$ /usr/sbin/snort -D -c /usr/local/etc/snort.conf
```

## RealSecure

The RealSecure IPS product is probably the most popular IDS/IPS in the world. It has a fairly typical client/server or two-tiered design. Each sensor reports to either the workgroup manager (deprecated) or these days to the site protector server. These centralized servers perform all management and administration associated with the IDS network.

The management software has to run on a Windows 2000 (or better) server and performs the following functions:

- **Console management** Allows you to display graphs, edit server policies, and view logs.
- **Event collection** Collect sensor events from the various support sensors.
- **Database storage** Data is stored in an MS-SQL database.

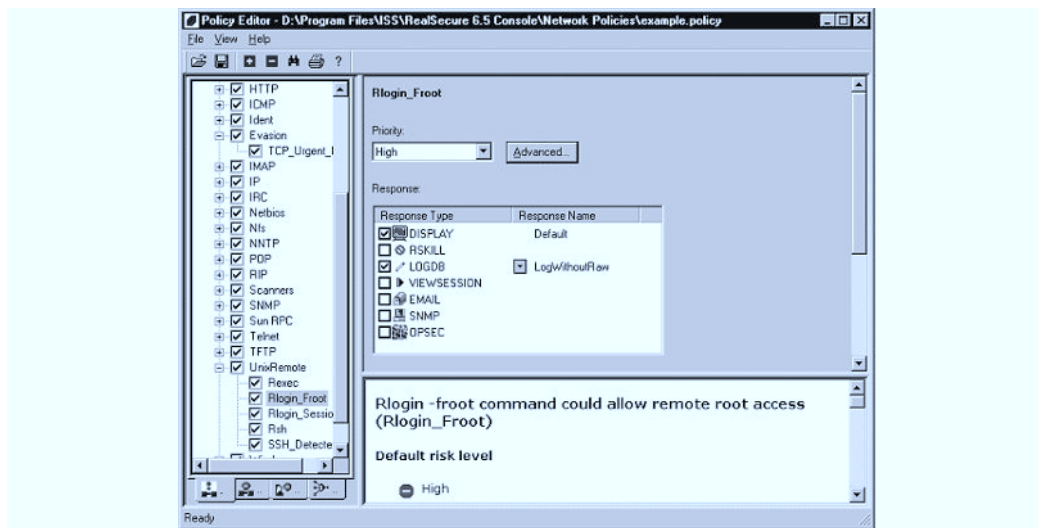
In a basic configuration, these components can run on a modest server. In a large configuration, these functions can be split on to different servers for performance and capacity. You can also build hierarchies of “site protector domains” to monitor remote sites or different client cells in an outsourcing environment.

Currently, RealSecure supports a number of sensors, which include:

- **Network sensor** A good old-fashioned NIDS.
- **Host sensor** A fairly average HIDS, which is becoming depreciated.
- **Server sensor** A combination of the network sensor and the host sensor. The main difference is that the promiscuous mode code has been removed.
- **Desktop sensor** This is a renewed version of the BlackICE desktop security software.
- **Appliance IPS** This is basically a network sensor with any necessary add-ons to make it compatible with Nokia and Provencia appliances. The latter can be inline or tap based.

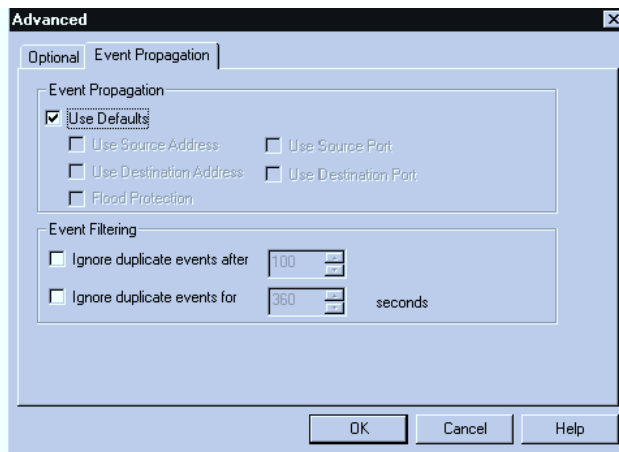
Once logged into the site protector, you can define new sensors to monitor or define new policies that can be pushed down to the individual sensor. This involves selecting one of a series of read-only default sensor policies and then “deriving” your custom policy by selecting additional events or deselecting events (see Figure 9.10). When you select a particular event, you also get the opportunity to select advanced IDS actions (covered in the IPS chapter) or to change the priority (covered in the next chapter).

**Figure 9.10** A Typical Response Screen



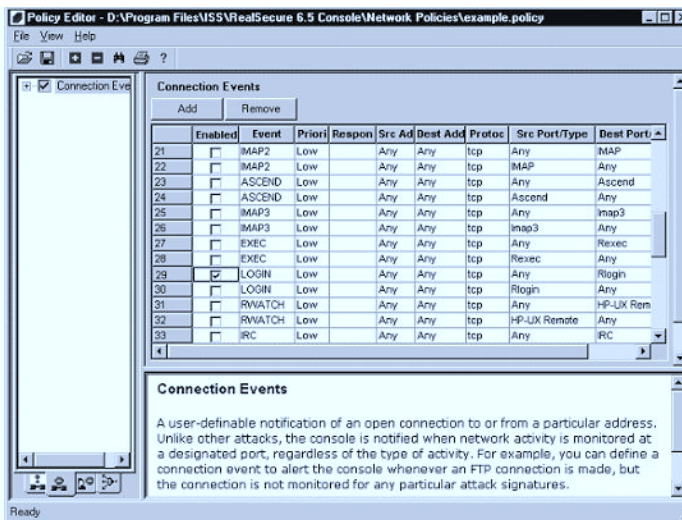
When editing events, you get the opportunity to change the ignore time period and the ignore duplicates count. These are usually very coarsely set; changing these values can reap very good efficiency savings (see Figure 9.11).

Figure 9.11 Advanced Options to Throttle Events

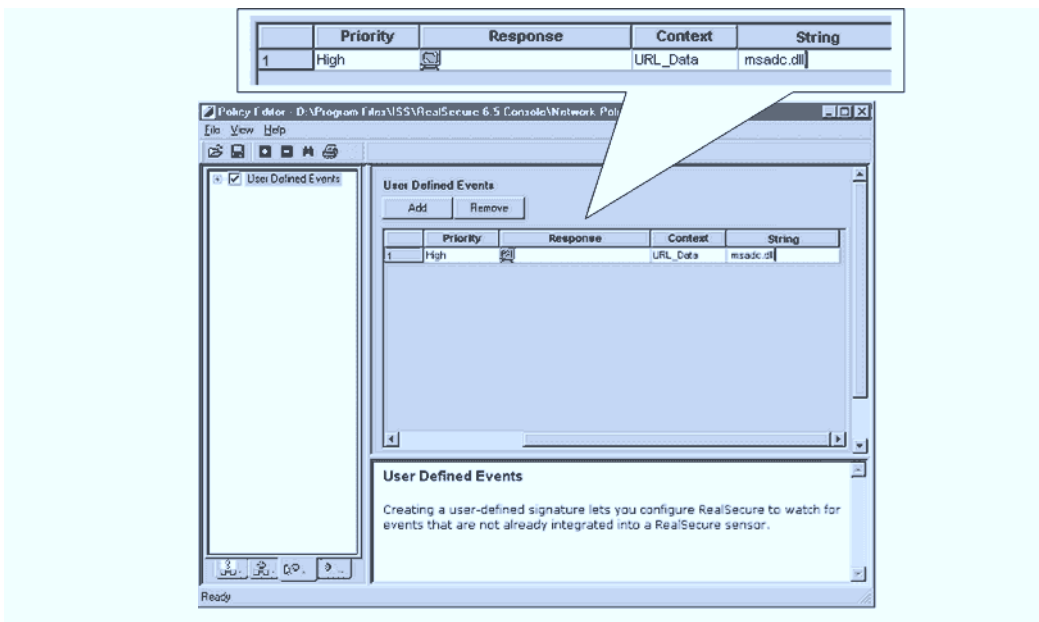


Apart from basic attack signatures, you get the opportunity to define connection events (see Figure 9.12). These are extremely useful for defining anomalous traffic profiles, as detailed previously.

Figure 9.12 Connection Events



When you discover you have a unique vulnerability or a vulnerability that hasn't had a patch released for it, you can design a custom event (see Figure 9.13). In this example, we prevent any user from executing a particular DLL, by killing the session if it is being referenced in any URL.

**Figure 9.13** User-Defined Rules

## Summary

We started this chapter with an overview of IDS types. The body of the chapter focused on the most common type of intrusion detection system, the NIDS.

The text outlined the common problems associated with an NIDS. This included suboptimal deployment, avoidance, and configuration problems.

Lastly, we provided an overview of two major IDSes, Snort and RealSecure.

## Intrusion Detection Systems: In Practice

The purpose of this chapter is to:

- Provide details of tips and techniques
- Provide a methodology for deploying an IDS
- Provide a detailed approach for tuning an IDS
- Provide a strategy for deploying an IDS

## Anecdote

*Intrusion detection systems (IDSes) are a form of monitoring. Another form of monitoring is open-source monitoring, whereby you scan newsgroups and forums for damaging information. Traditionally, a good pen tester or hacker will do this before every job; it's amazing how many administrators post configs to newsgroups, asking "Why doesn't this work?"—just the "in" we're looking for.*

*Years ago, my team was doing a job for a large chemical company. The executives had some branded e-mail accounts on a dial-in basis, but the test was to see if the new corporate gateway was up to snuff.*

*But we hit a problem. The finance director (FD), our sponsor, a 55-year-old, silver-headed fossil, was advertising his services as a "toy-boy. The evidence was clear: many postings from john.smith@acme-chemical.demon.co.uk, all offering male escort services.*

*I escalated it to a senior partner, since this was a problem well above my salary range. We were told to continue. Some time later, we were invited to present interim findings to the board. But it was a most extraordinary meeting.*

*The room collapsed with laughter when we retold the point about escort services. The FD's face, bright red with mock anger, bristled delightedly with each new jibe—nobody had paid him this much attention in years. Lots of "You're looking tired, John, out earning some extra cash?" and "Trying to make the books balance?" made his day.*

*Apparently, he had lent his laptop to his struggling MBA student intern, who was the one responsible for the escort posting. Some helpful IT person had shown the FD where to find the "tick this box to remember password" setting. And that was that!*

## Introduction: Tricks, Tips, and Techniques

Theory is all well and good—but knowing the practical side of things helps, too. This chapter provides tips and implementation hints for putting your IDS knowledge to good use.

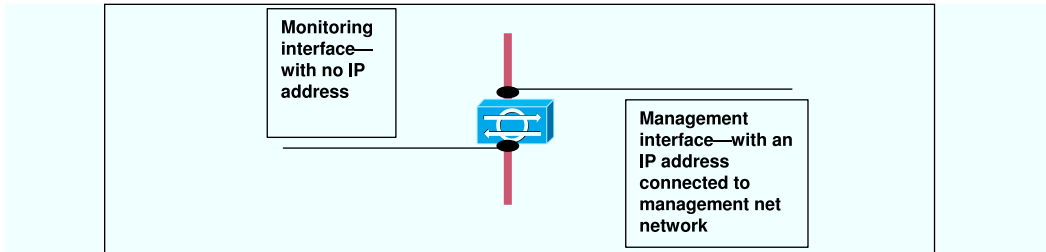
## Deploying a NIDS: Stealth Mode

These days, IDS sensors run any number of different operating systems. It is good practice to build your sensors in stealth mode, which requires two network interfaces. One of these will have an IP address and will be used for all



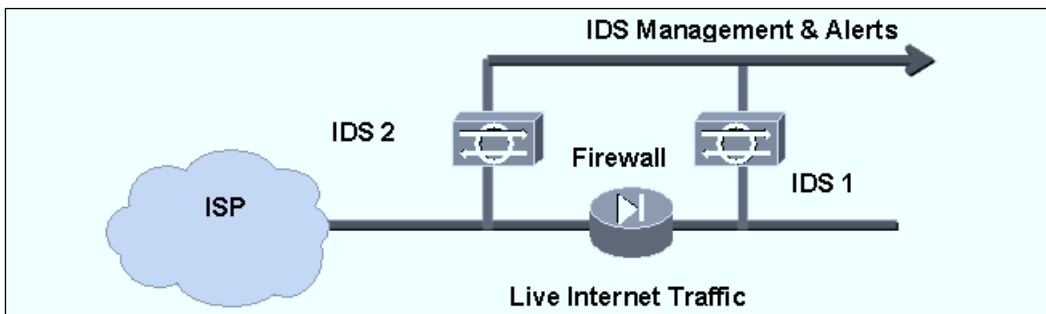
management traffic, maintenance, and alerting. This is the *management interface*. The second interface is the *stealth, monitor, or probe port*. This interface isn't given an IP address, because it doesn't transmit information—it just listens (see Figure 10.1). The port needs to be in promiscuous mode. However, most IDS do this programmatically with the appropriate *ioctl()* call, so no manual intervention is required.

**Figure 10.1** A Typical Stealth-Mode Implementation



A simple but typical configuration is shown in Figure 10.2.

**Figure 10.2** A Typical Deployment



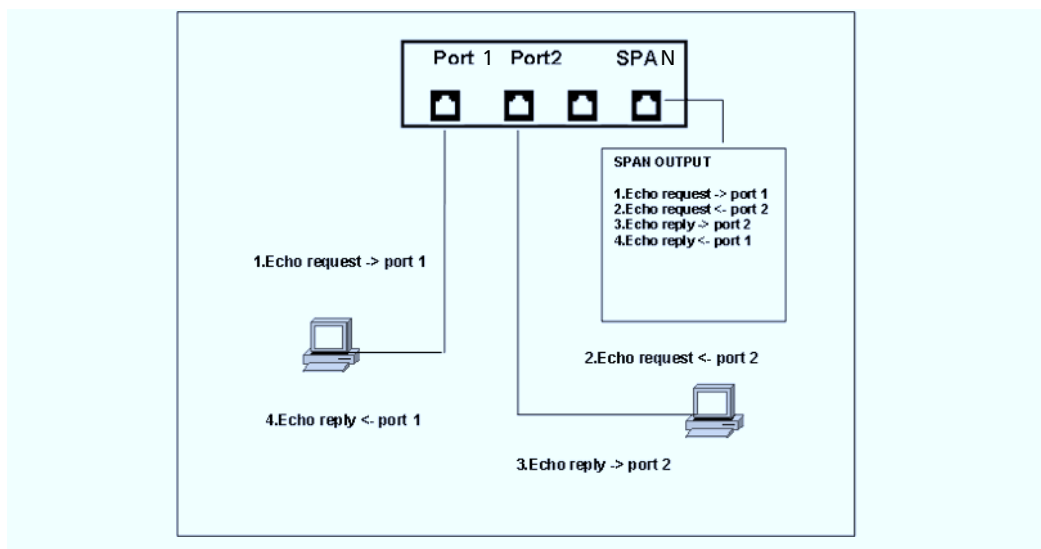
## Spanning Ports

As mentioned in the previous chapter, many larger IDS sites become “sick” of spanning ports. This is due to three factors:

- Dropped packets** A device connected to a span port on a switch does not see all traffic. Corrupted network packets or packets below minimum size (giants and runts) can be dropped by the span process. This could cause you to miss certain attacks.

- Span port capacity** For you to monitor one full-duplex port on a 100Mbps link, a span port would need 200Mbps of capacity. Effectively, a 100Mbps span port will be oversubscribed if the switch runs at 30 percent capacity and monitors four ports, unless a gigabit spanning port is used. Even though gigabit ports are often limited, you have to use one.
- Traffic duplication** By default, a span port monitors traffic as it enters and leaves a port. This means that if Port 1 and Port 2 are both monitored and the host on Port 1 pings the host on Port 2, the span port will receive two echo requests and two echo replies (see Figure 10.3).

**Figure 10.3** Packet Duplication on a Span Port



Traffic duplication can cause problems in packet reassembly for the IDS and accelerate oversubscription of the port (the load on the spanning port is double the actual traffic rate monitored). In Cisco CatOS, this situation can be mitigated by monitoring only inbound traffic on the spanning statements:

```

Console> (enable) set span 1/2 1/4 tx inpkts disable
Console> (enable) set span 1/1 1/4 tx inpkts disable
  
```

Please note the options used to prevent received (*rx*) traffic being mirrored. The final options (*inpkts disable*) prevent the IDS from providing input into the switch or participating in spanning trees. In aggregating traffic, these options are essential or routing loops will occur. IOS-based switches do not afford this facility.

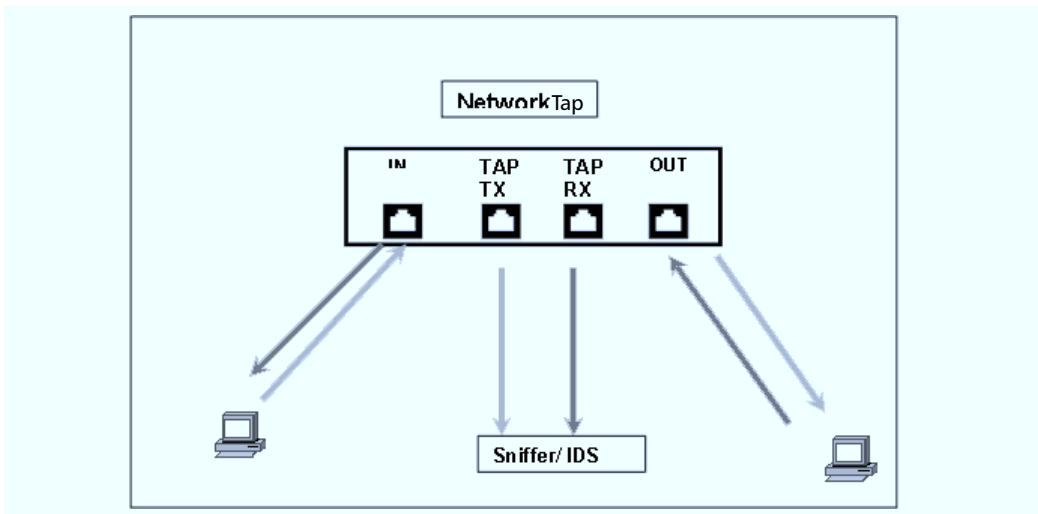
```
Switch(config)# interface fa 0/4
Switch(config-if)# port monitor fastethernet 0/2
Switch(config-if)# port monitor fastethernet 0/1
```

Port spanning is also limited by the following factors:

- **Span limits** On some switches, the number of ports that can be monitored is limited to about half a dozen.
- **Span servicing** Copying the data on to a span buffer takes CPU. This will cause the CPU of the switch to increase. When the switch approaches 100-percent CPU utilization, the span port will drop packets because the span service has a very low internal priority.
- **Switch maintenance** Maintaining a large number of spanning ports for a large sensor population increases the network team's workload.

## Tap Technology

A *tap* is an inline device that provides a method of directly viewing traffic on a full-duplex or half-duplex 10/100/1000 Ethernet segment. Typically, the tap (see Figure 10.4) is a hardware device that copies the electronic signal—like a Y splitter for a TV aerial. It adds virtually no delay to the network but can rapidly leave you with a huge tangle of wires.

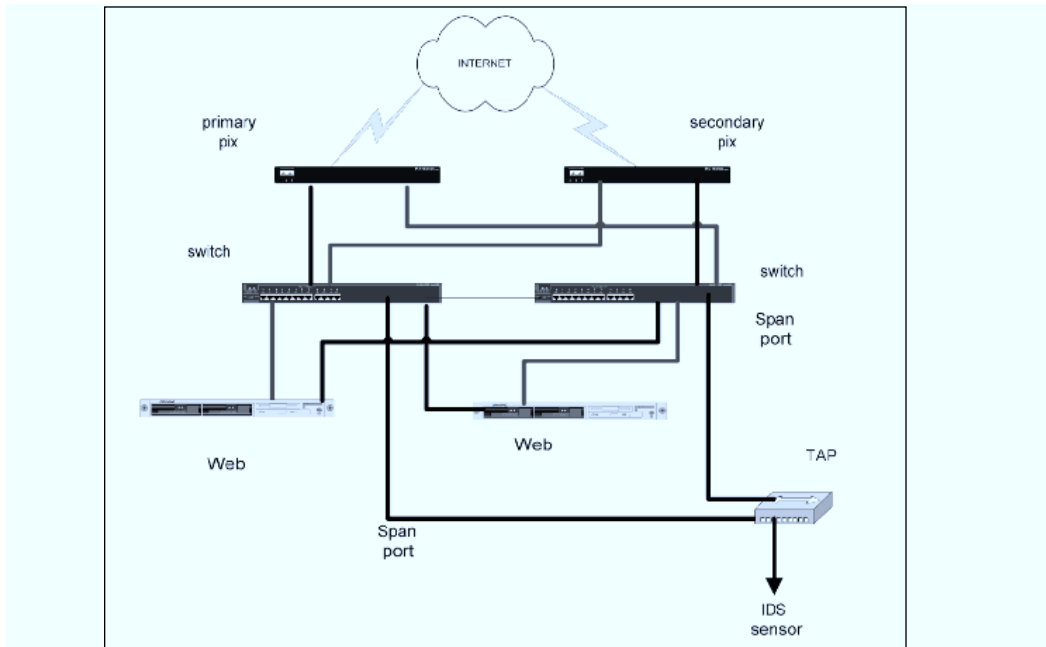
**Figure 10.4** A Basic Network Tap

Many modern taps aggregate RX/TX into one port.

## Failover Monitoring

In an active/passive failover situation, as shown in Figure 10.5, multiple IDS sensors are often used. Even if freeware Snort is the sensor of choice, the hardware costs and effort required is doubled—and yet one of the two machines will remain dormant most of the time. An intelligent network tap can be configured to allow traffic only from the active leg to reach IDS sensor, as shown in Figure 10.5.

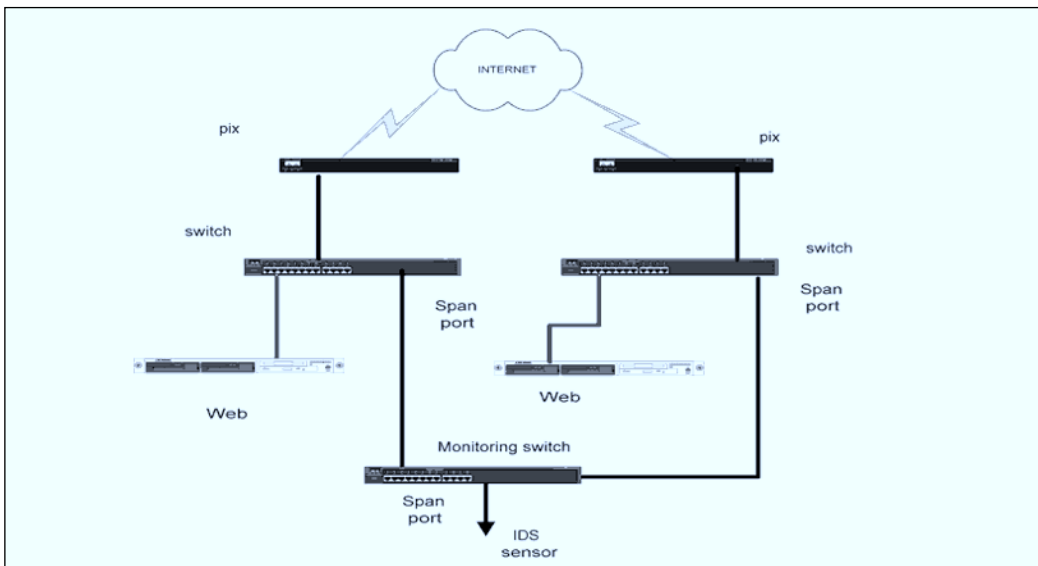
Alternatively, if you have a spare switch, you can aggregate the traffic by aggregating different flows.

**Figure 10.5** Active/Passive Segment Monitoring with a Tap

## Aggregating Different Flows

In Figure 10.6, two low-volume segments need monitoring. Two sensors would be an unnecessary expense. One sensor can be used. To monitor the two segments, you'll need to follow these steps:

1. Set all spanning ports so they are not allowed to act as input ports or participate in spanning trees. This must be done or the configuration will cause routing loops.
2. Do not connect the spanning ports from Web segments to an IDS, but connect the spanning ports from Web segments to another switch.
3. On this monitoring switch, span all ports onto another span port, then connect the IDS sensor to this port.

**Figure 10.6** Monitoring Two Segments with One IDS

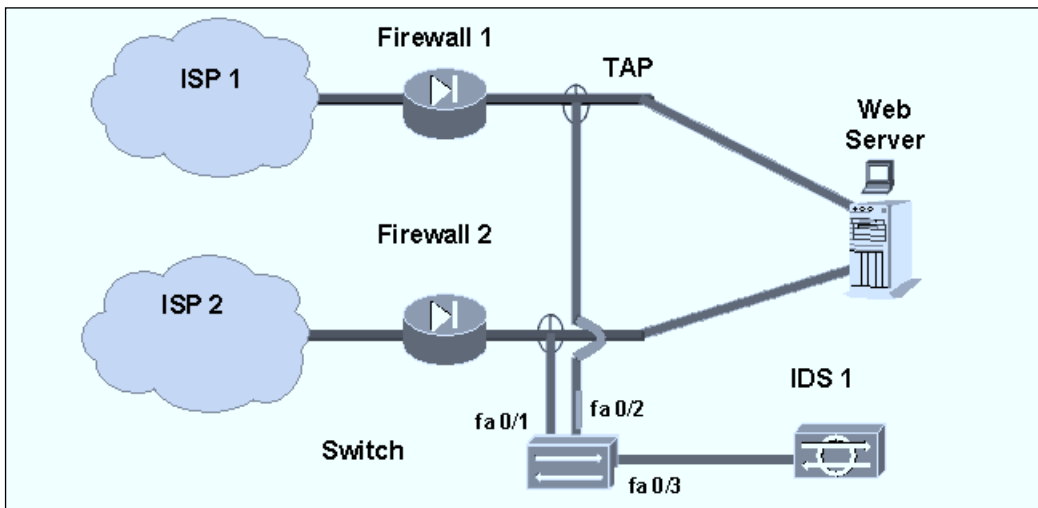
## Asymmetric Routing

Asymmetric routing problems were described in the previous chapter. Let's recap: In resilient environments, dynamic routing protocols are employed to gain route diversity. In Figure 10.7, this could mean that:

- **Client A** from ISP 1 sends a packet to the Web server via Firewall 1 and then returns through Firewall 1 and ISP 1 to the client.
- **Client B** from ISP 1 sends a packet to the Web server via Firewall 1 and it returns through Firewall 2 and ISP 2 to Client B because that is the preferred route.

This latter situation will cause problems for any stateful device (including the firewall; in real life look for filtering routers) and also more advanced IDSes that keep state or inline IPSes.

The solution is to tap the traffic and then combine it with an aggregating switch. (Note that there is no good inline IPS solution if the objective is to build resilience—one more reason that you are more likely to see an IDS than an IPS on a backbone). The following configuration will do the job:

**Figure 10.7** Asymmetric Routing

```
Switch(config)#
Switch(config)# interface fa 0/3
Switch(config-if)# DESC port 3 SPAN port feeds IDS
Switch(config-if)# port monitor fastethernet 0/2
Switch(config-if)# :: fa02 sniffs ISP2 packets
Switch(config-if)# port monitor fastethernet 0/1
Switch(config-if)# :: fa01 sniffs ISP1 packets
Switch(config-if)# exit
```

## IDS Deployment Methodology

Increasingly, IDSes have been used to augment security, and when deployed correctly they can make a significant contribution to the security regime. They enhance security for the following reasons:

- IDSes provide detection to the firewall regime that is mainly prevention based.
- IDSes add the inspection of application data and session data, whereas firewalls concentrate on network protocol exposures.

- IDSeS can aid in the processing of log data, which needs to be inspected. It is quite common to have no security monitoring behind a firewall; this makes it impossible to detect security problems.

IDSeS can be ineffective because:

- They are typically installed by value-added resellers (VARs), which don't really understand them.
- They aren't tailored to fit the environment to detect unauthorized traffic.
- They overloaded by inappropriate signature.
- They are not linked to manual procedures.

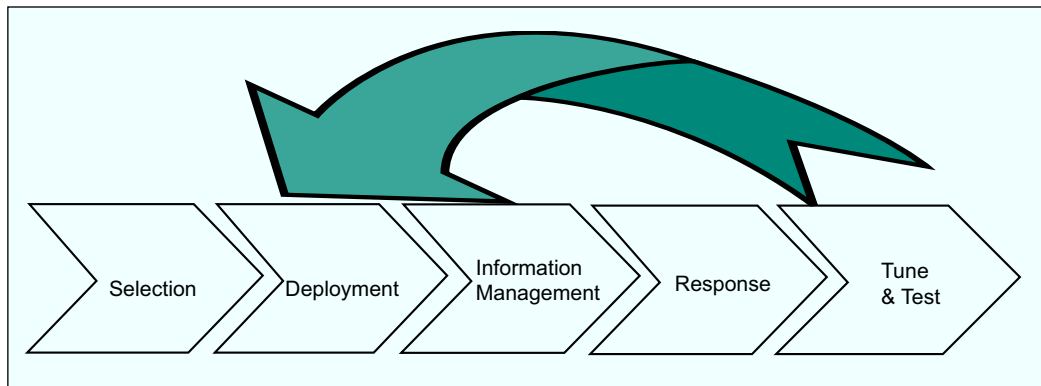
The following methodology was designed to avoid these problems.

## The Methodology

The methodology shown in Figure 10.8 comprises a number of identifiable steps:

1. Selection
2. Deployment
3. Information management
4. Incident response
5. Testing

**Figure 10.8** Diagram of the IDS Methodology





This methodology is flexible and can be used when a product has already been selected or in a greenfield situation. Since 1998, the methodology has been successfully used in a number of e-banking solutions and has been used successfully in an enterprisewide “fixed perimeter” situation.

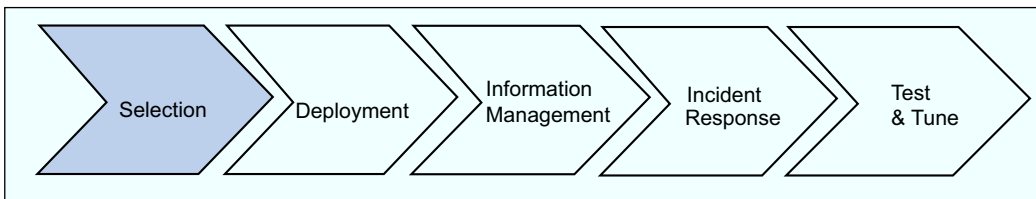
## Selection

The first stage of the IDS methodology is selection (see Figure 10.9). This is not a security product selection methodology; however, it might be worth mentioning some of the key selection criteria that may be used:

- **Speed of signature release**
- **Type of systems architecture in use** Most IDSes/IPSes work with common UNIX or Windows, but if a significant part of the infrastructure is based on mainframe computing, there could be an issue that the IDS does not cover your major assets.
- **Type of network architecture in use** Again, most IDSes work with 100BaseT Ethernet, but today’s business networks include ATM, Gigabit Ethernet, copper, or fiber. If you are a provider, can the IDS manage overlapping address ranges? You may have five clients, each using the same RFC 1918 address range. Will this confuse the IDS?
- **Ease of customization** This is particularly important because a good IDS should be able to interface with a network management system such as Tivoli, OpenView, or Unicenter. It must also be able to receive messages from applications and communicate with unusual devices.
- **Need for customization** Some IDSes arrive in kit form and won’t interact with firewalls or routers.
- **Deployment platform** Is a firmware appliance version available? Is one required due to unattended operation or lack of onsite system skills and the like?

- **Scalability** Two key features have to be considered here: the ability of the console to manage more than, say, 20 data collectors and the ability of the database to store the data.
- **Ease of maintenance** Don't forget: Policy updates come out very frequently (once a week). Centralized management is must.

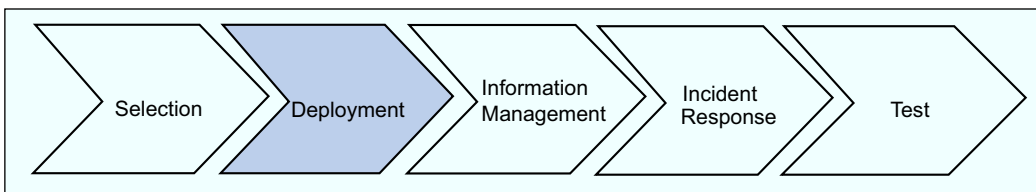
**Figure 10.9** The Selection Phase



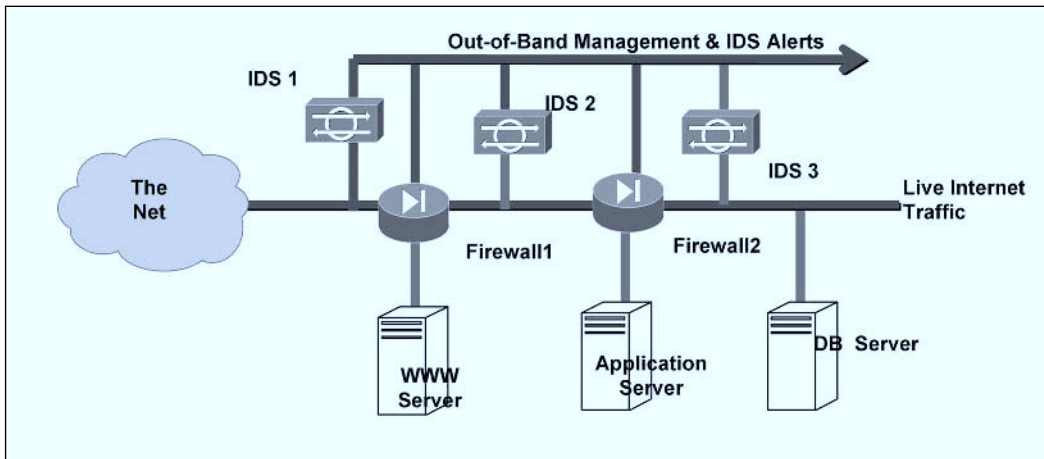
## Deployment

The second stage (Figure 10.10) is deployment.

**Figure 10.10** The Deployment Phase



Sensors are the key elements of the network IDS/IPS system that are capable of identifying patterns of suspicious network traffic and questionable user activities. The sensors' effectiveness depends on their internal design and, even more important, their position within the corporate architecture (see Figure 10.11).

**Figure 10.11** IDS Sensors in a Typical Corporate Network

Generally, sensors can be classified into two categories: network sensors and host sensors. For the purposes of this chapter, we will initially focus on network sensors and then cover the deployment of host sensors.

## Step 1: Planning Sensor Position and Assigning Positional Risk

Network sensors monitor a defined network segment. The positions they are deployed in are determined by two fundamental points:

- **Reason 1** The network segment contains assets that require protection and are at risk from attack.
- **Reason 2** The network segment would give a sensor the ability to predict an attack or defend against an attack.

Figure 10.11 shows three classic positions of the IDS sensors.

### Sensor 2

This is the ideal position for a sensor. The network segment the sensor is on contains servers that require protection (Reason 1). However, the DMZ is traditionally considered an intermediate stepping-stone to the main network. Correspondingly, a sensor could be justly positioned for preemptive reasons (Reason 2). Sensor 3 is justified by Reason 1 entirely. The positioning of

Sensor 1 is justified by Reason 2. It provides no pure security value and will generate lots of noise and false positives, so why put it there?

## Tools & Traps...

### Positioning Sensors

Many people suggest placing sensors in front of the firewall because, they suggest, it is important to know what attacks are being thrown at you but are blocked by the firewall. This is not a security reason; it is *probably* not your job to police the Internet and report on the breadth and variety of attacks. However, this information might be extremely useful to present as cost justification in boardroom battles. Showing the board the number of attacks launched at your firewall can carry a powerful message—when presented in a nice security dashboard.

However, a sensor placed in such a position is very likely to cause dozens of false alarms from attacks that will not affect you. It is vitally important to configure “outside firewall sensors” so that they only log stats. Perhaps consider a honeypot (a dummy computer) if you need to monitor hacker behavior, but be aware that honeypots do attract unwanted attention and could lead people to believe that your organization has weak defenses.

Each sensor has a positional threat rating associated with it, depending on its position within the network. This rating roughly relates to the sensor’s proximity to the assets on a network. Correspondingly, an attack detected at Sensor Position 1 (where there are no assets) represents a potentially lower threat to the organization than if it was detected in the DMZ at Position 2. Similarly, attacks registered at Position 3 are very serious because they have bypassed both the perimeter and the internal firewall. These attacks indicate a real intrusion of the internal network, where there are high-value assets. Table 10.1 shows the three sensors with their associated risk levels.

It should be noted that I first published this methodology for rating sensors in 1999 and have been presenting it since 1998. Today many respected

manufacturers include an *absolutely identical* methodology, but back then I had to customize my own to implement this scheme.

**Table 10.1** Three Sensors with Their Associated Risks

Type of Sensor	Positional Threat Rating
Sensor 1	Low
Sensor 2	Medium
Sensor 3	High

## Step 2: Establish Monitoring Policy and Attack Gravity

Each time one of the sensors identifies an intrusion, an alert is generated and reported to the main IDS/IPS system. What is classified as an intrusion is controlled by a monitor policy. To minimize the detection of false-positive alerts and the overhead of the IDS/IPS, it is good practice to tailor the policy. This usually consists of first tailoring a provided list of attacks monitored by the sensor so that it is relevant to your environment; for example, if you are running a UNIX environment, it is not good use of your CPU to monitor the network for winnuke or null session attacks, which affect only Windows environments. Second, you should modify the detector to detect remarkable traffic. This process is expanded in Table 10.2.

### NOTE

Attack signatures and abnormal traffic alerts are flags for abnormal events on your network (see Table 10.2).

**Table 10.2** Network Alerts

Type of Alert	Examples
Attack signatures	<p>A known attack like nmap port scan, evil-ping, or Unicode hack exploit. Decide which signature to use based on these questions:</p> <ul style="list-style-type: none"> <li>■ Do I run the software or OS referenced in the signature?</li> <li>■ Is the signature referring to software that is older than my site's standard, or very old? (Many signatures still refer to Windows 98.)</li> <li>■ Is the attack covered better by other software (i.e., antispam or antivirus)?</li> </ul>
Abnormal traffic alert	<p>Traffic that is due to the security regime or system environment is suspicious. Typical examples of these may be:</p> <ul style="list-style-type: none"> <li>■ An rlogin, a common UNIX utility, attempting a session in a Windows network environment</li> <li>■ An SSH session from the perimeter firewall to a Web server, in an environment where console only access is prescribed</li> <li>■ An attempted session with two DMZ servers using their external addresses in a NAT'ed environment</li> <li>■ Telnet from inside DMZ back into the internal network</li> <li>■ TFTP from Web server out to the Internet</li> <li>■ Web browsing from the firewall</li> </ul> <p>None of these events is definite proof of a hacked network. However, in many cases they are indicative of an abnormal event that must require further investigation. It may have only been caused by a new firewall administrator unaware of the rules governing Web server access, or it may indicate a failure of the anti-spoofing rules (as in the last case). An investigation of front and rear firewalls will provide detailed information here.</p>

Continued

**Table 10.2 continued** Network Alerts

Type of Alert	Examples
	<p>On RealSecure, for example, this can be captured by a connection event or, in some cases, a user event. In Snort, for example, this is a simple to/from rule. In the Cisco product this is a little harder to achieve.</p> <p>Designing an anomalous traffic profile was covered in the previous chapter.</p>

Once a monitoring policy is established, the severity of each alert must be assigned. To minimize the detection of false positives, different gravity levels are assigned to the reported alarms. Otherwise, a high number of false positives will lead the operators of the IDS to ignore its output, which could lead to an actual intrusion being detected but ignored by the operators.

Various network events with their conceived risk levels are shown in Table 10.3.

**Table 10.3** Network Activities with Their Risk Levels

Network Events	Attack Threat
MSADC or Unicode hack	High
Port scanning	Medium
Outdated attempt of a DOS attack	Medium
Telnet attempt	Low

A key concept of this methodology is that the overall alert severity depends on the detected attack signature combined with the position of the sensor(s) that raised it.

**Alert Severity = Position Threat + Attack Threat**

However, it must be noted that there is a degree of subjectivity involved. Figure 10.12 shows a sample set of network activities, together with their relationship to the sensor position. The greatest security threats come from attacks on the internal network because the majority of protective barriers (firewalls and RAS servers) have already been bypassed at this point; simply

put, they are more likely to succeed from that point. Therefore, whenever the IDS sensors are triggered by abnormal patterns on the internal network, they are classified as having a greater risk level than the other sensors.

**Figure 10.12** Network Activities with Their Associated Gravity Levels, Depending on the Position of Each Sensor

NETWORK EVENT	SENSOR POSITION			
		SENSOR 1	SENSOR 2	SENSOR 3
Port scannings	GRAVITY LEVEL	LOW	HIGH	HIGH
Telnet (attempt from outside)		LOW	MEDIUM	HIGH
Outdated attempt of a D.O.S attack		MEDIUM	HIGH	HIGH

To clarify the gravity levels shown in Table 10.3, we should consider an example. Assume that a port scan is detected at the Internet router (Sensor 1). Although this event does not constitute a high-impact attack, it is a common initial sign that a more serious attack will shortly follow. Therefore, this event should be characterized as low risk and recorded in a log file. However, if a port scan is detected in the internal network (Sensor 3), this implies that either the attacker has been successful in compromising one or more internal computer systems or an internal user is trying to abuse the network. The latter event has clearly more potential for harm than the former, and therefore it is characterized as high risk. In short, an attack detected on an inside network is more worrying than an attack detected outside your firewall.



## Step 3: Reaction

Once the severity level of the reported alert has been established, the IDS can respond via a set of predefined actions. Table 10.4 shows the associated risks, together with suggested alert actions, in a typical environment.

A full table of IPS/IDS active responses is available in the chapter on IPS.

**Table 10.4** Proposed Actions Depending on the Level on Risk

Event Importance	Alert Action
Low	No further action at this stage; possibly record for forensic use. Example: Log the incident in the log files.
Medium	Flag the incident for next-working-day follow-up. Example: E-mail firewall administrator.
High	Immediately alert the operator and/or act against the offensive connection. Example: Page firewall administrator and send a TCP/IP RST to initiating tuple.

When a high-level alert is generated, the IDS has to immediately inform the operator that an attempt at intrusion has been detected. If the attack is considered to be harmful to the network, the IDS can actively act against it.

## Step 4: Further Action: IPS

If you are considering using an HIDS or IPS, you should consider the following issues. IPS is covered in full in the next chapter.

### Firewalls, Master Blocking, and Inline IPSes

Typically, an IDS can respond by resetting the suspicious connection, by locking-out particular addresses in the firewall, or even by shutting down the firewall. A full-blown inline IDS can drop just that single packet, making the repercussions of a false positive slightly less significant. Only this significance is overplayed—if a user cannot connect to an exchange to pick up her mail, she won't care if her login failed because her address is blocked or because the data contained in her authentication request is dropped; the fact is, she still can't log in.

Another example is shutting the firewall whenever a port scan is performed on the router. This is considered to have a more severe effect (effectively, denial of service) than the original activity.

In any case, you must take care to ensure that the adverse consequences of the active response outbalance the impact of the detected attack. However, there is great value in proactive response. Most IPS vendors use active response on only about 30 or 40 percent of their signatures. Be guided by this statistic; it shows the ones they have the most confidence in. Use proactive response for a reason; if you know that your firewall will block that packet, why bother?

### TIP

---

At this stage your policy regarding active protection should be defined and documented. Knowing what you block and why provides a helpful diagnostic tip.

---

## Host Detectors

Host detectors can be deployed using the same methodology. Doing so requires only a further iteration:

- To plan which servers to deploy detectors on
- To establish what the policy should contain

Even if the IDS in use doesn't support host detectors, normal facilities such as *syslogd* and *swatch* can often be deployed to great effect, with the messages being interpreted by the NIDS. Additionally, PortSentry and Tripwire provide strong alternatives.

## Application Interface

With the advent of e-commerce, more external interaction with applications is encouraged. This activity exposes the applications to security threats such as brute-force attacks and code manipulation. Therefore, the need for application interfaces into the IDS is becoming more important—after all, often the IDS

is purchased to protect an Internet banking application. Therefore, it is important that the IDS be able to respond to, for example, multiple requests to the custom application login function for multiple accounts from one address in a set period of time, exactly as commercial IDSes respond to repeated standard HTTP authentication requests in the same situation.

But most applications can send a *syslog* message. Important ones should be captured by the IDS and reacted to appropriately.

## Honeypots

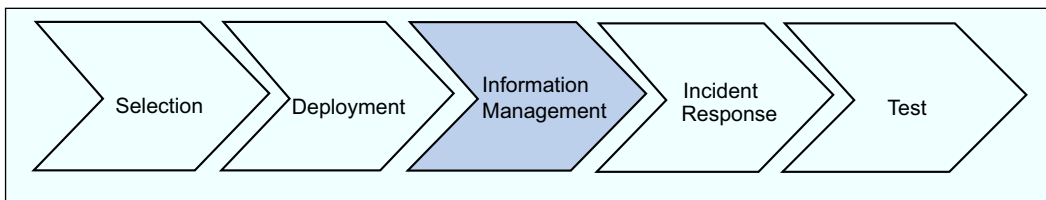
Honeypots are a great tool for research. Production honeypots—those used to detect hacking traffic—are covered in the next chapter on IPSes.

# Information Management

The information management stage is usually very short, but it is often forgotten (see Figure 10.13). It deals with:

- Where the information is delivered
- What form the information takes
- In what time frame the information is delivered
- In what form the information is retained

**Figure 10.13** The Information Management Phase



## Log Management

Once the sensors are configured, the IDS will potentially accumulate a huge amount of information. This information must be processed and archived. It is therefore necessary to have log management procedures. These procedures should define:

- When the log should be archived and cleared down
- How long it should be retained
- Who should have access to the information

## Console Management

Console management falls under two categories:

- Logical access controls
- Alert management and consolidation

### Logical Access Controls

The IDS alert can contain confidential information. Inline IPS superuser access to an IDS console can allow you to shut down the network. If you have HIDS sensors, access to them allows you limited superuser access to the network. However, operations or audit may legitimately want access to the alerts and event logs. At this stage, you need to consider access roles and appropriate access rights.

#### *Alert Management and Consolidation*

For large organizations that have 24x7 operations departments and proper network management software, it is good practice to report all high-severity incidents to an enterprise console such as OpenView, Tivoli, and CA-Unicenter, which will be continuously viewed by operators.

At this stage you might want to consider a purpose-built correlation console or security information management software. These can correlate:

- Previous/subsequent events from the same sensor (aka *attack correlation*)
- Events from other sensors to the same destination (aka *multipoint detection*)
- Events from other sources, sensors, or logs; previous/subsequent events (aka *multisource capture*)

- Events to inventory information
- Events to vulnerability scan information (aka targeted IDS)

These products are often in constant contact with the Internet, downloading proprietary signature, CVE, CIDE, and CISL information. The software may also have an inference engine and contain a degree of artificial intelligence. These features can drastically reduce the number of false positives and increase the detection capability. In 2001, my “I-AM-DOH” project proved that by simply combing Snort, Nessus, and an inventory database (in a very flaky Perl patchwork, I might add), false positives could be reduce 70 percent in a live environment.

These correlation consoles can enhance a weak IDS considerably. They can also compensate for weak staff, proving an element of expert analysis and reasoning, and provide response and countermeasure suggestions.

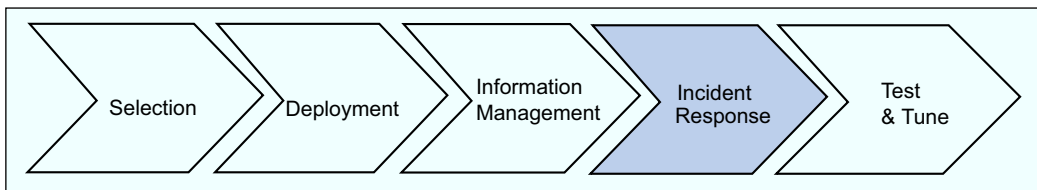
However, they are very expensive and often require a lot of setup. For simple DMZ monitoring by a high-end IDS/IPS such as RealSecure, which has much of the necessary functionality in Site Protector, the return would not be significant.

Products in this category include:

- Cisco CS-MARS
- Symantec Incident Manager
- NetForensics
- ArcSight ESM

## Incident Response and Crisis Management

There is no point in having IDS/IPS software installed if adequate incident response procedures are not present in the organization (see Figure 10.14).

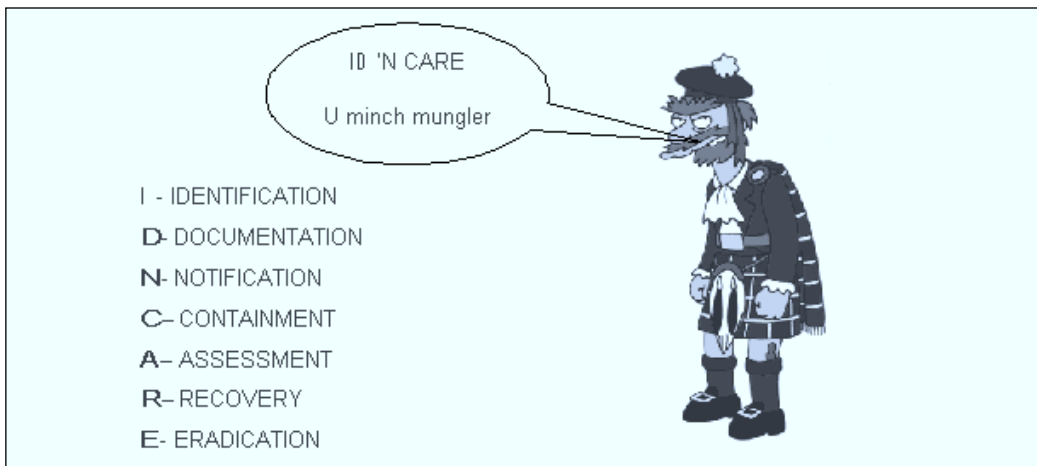
**Figure 10.14** The Incident Response Phase

Key elements of good incident response procedures include:

- Early notification of potential events
- Clear escalation procedures with *defined* time limits for duration of each stage
- Automatic percolation up the stages of escalation, reversed only by formal sign-off
- Providing the people on the ground with power to make the decisions

But most of all, procedures should be *written down*.

However, designing incident response and crisis management processes is a very specialized job. A few details of the classic stages are provided in the next section. These stages are identification, documentation, notification, containment, assessment, recovery, and eradication. Collectively they are known as IDNCARE (see Figure 10.15).

**Figure 10.15** The IDNCARE Stages

## Identification

The first step to successfully managing a security incident is recognizing that you have a security problem. Indicators can be:

- An alert from an IDS
- A missing file or piece of equipment
- A crashed Web server
- An anomaly in the system behavior
- A virus message

To ensure effective processing, staff must be appropriately trained.

## Documentation

In dealing with an incident, it is essential that you have all the information possible easily to hand.

## Notification

Whom do you tell, and when? And when the news is still bad, whom else do you tell?

## Containment

Containment of the incident is necessary to minimize and isolate the damage incurred by your company.

## Assessment

The following are some of the factors to consider during the assessment:

- The size of the event (how many computers are affected by this incident?)
- How sensitive is the information involved?
- Where did the incident occur? What is the point of ingress (e.g. wireless, network, phone dial)?
- What is the potential damage caused by the incident?

- What is the estimated time to recover from the incident?
- What resources are required to manage the situation?
- What laws or regulations could be involved?
- How should the assessment be performed effectively?

## Recovery

We build systems to use them, and generally when we are not using them, we are losing money. We don't want to work sequentially if it means we stay down longer than we have to. The recovery phase is about getting your systems usable again so that normal business operations can be resumed, free from the original vulnerability.

Solutions here may include:

- Using patched DR equipment
- Running essential systems in a closed, restricted environment
- Temporarily blocking firewall ports or Web sites
- Using a standby generator

## Eradication

Here we eliminate the threat from the system and return to normal practice. Typical solutions may be:

- Wholesale patching program
- Transfer from DR equipment to live
- Firing an employee

## Other Valuable Tips

Here are some other tips:

- Know where to go for help (don't scrap all your dial-up accounts).
- Know the systems owners (know their telephone and fax numbers).

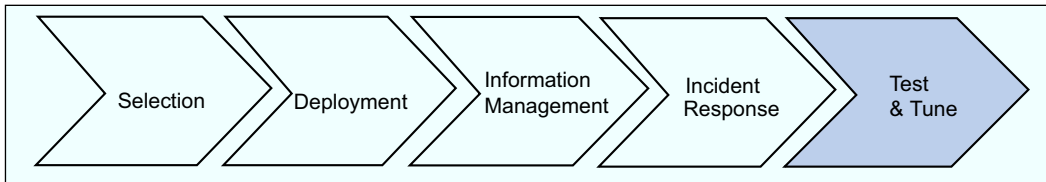


- Do you have a system inventory and a network map?
- *Get a packet sniffer* and a selection of media converters.
- Get authority to do what you need to do.

## Test and Tune

The last stage (Figure 10.16) is test and tune.

**Figure 10.16** The Test and Tune Stage



## Tune

Tuning an IDS involves three basic tasks:

- Reduce false positives.
- Reduce false negatives.
- Reduce data overload.

## Reduce False Positives

Make sure your IDS/IPS engine is up to date and that you upgrade your IDS with a protocol analysis module. This will ensure that the detection engine is as good as it can be. Many of the “last version” detection engines are poor.

### *Disable Noisy Rules/Signatures*

Load the top 10 events in a spreadsheet. For each, consider:

1. Do you need to know what a signature is telling you? If no, turn the signature off.

2. If you need the signature but it is generating false positives because it is firing on the wrong kit (Linux destinations receiving w2k alerts, for example), filter the alerts for those address ranges.
3. If you are still getting false positives, is the volume due to repeats? If so, look at *alert after #* or *ignore after #* values.
4. If the signature is still producing false positives, consider recoding it or dropping its priority.

### *Make IDS Maintenance a Learning Process*

If after research the alert does resolve to a false positive, filter it out or disable the rule. Don't make the next sucker redo the research you've just done.

## Reduce False Negatives

To reduce false negatives, you'll need to:

- Use automated or pre-emptive procedures to load new signatures.
- Use anomalous detection.

### *Review Actions Taken*

For each event that caused you to do some work, ask yourself:

- Could I reduce this effort by using active response?
- Could I reduce this effort by changing another sensor or changing a firewall?
- Do I need to write a rule to catch the event earlier?

## Test

Testing should be conducted at two levels: technical and covert penetration.

### Technical Testing

Manufacturers make a great many claims about their IDSes. You need to establish the capability for yourself. This process should include:

- **Throughput** Take the network to 60-percent utilization using the tool of your choice (*sprayd* or *observer.pro*), then check for dropped packets. Make sure you devise techniques for detecting dropped packets and, if possible, very unusual traffic rates (even if not from the IDS).
- **Avoidance** Use commercial software such as Blade or open-source software such as Adm-mutate, Fragrouter, or ITB, plus the avoidance modes of Whisker or Nmap.

## Covert Penetration Testing

Use a covert penetration testing service to test the configuration of an IDS and the reaction of your staff to the alerts. Usually this covert activity will involve a mixture of off-site and on-site testing. Ensure that you avoid your own quick-brew “Cuban missile crisis” by making sure all the right people know what’s going on in advance. Otherwise, you will end up with the bad publicity you are trying to avoid.

# Summary

In this chapter we covered the more practical details for deploying an IDS, including:

- A number of tips to save money on sensors by using switches and taps
- A methodology for deploying IDS
- A detailed approach for tuning IDS
- A strategy for deploying IDS

In the following chapter we deal with active response to attacks via the IPS.

## Intrusion Prevention and Protection

The purpose of this chapter is to:

- Provide a working definition of an IPS
- Provide a catalog of the active responses and the way attacks can be responded to
- Provide a quick tour of some current IPS implementations
- Provide a description of problems IPSes can solve

## Anecdote

*I was very lucky to work on most of England's Internet banks, and it really was a pleasure. Apart from the general excitement that always surrounded a new e-commerce project, the banks were risk-averse organizations that rarely cut corners on security, allowing me to delve deep into the areas I was working on.*

*An Internet bank was being launched in North and my company won the contract to review and improve the security controls. I had made all the necessary improvements to the firewalls and the routers. The intrusion detection system (IDS) was the last component that needed to be reviewed, but that was not going to take place until the morning of the first day the site went live. The implementation team had been let down by the supplier, so the system administrators were going to install and configure the IDS on the big day. The rationale was that an IDS was only a detective control, so the bank could survive it suboptimally for a day or two. It's not as if it was a really important detective control.*

*When I arrived at the office, all was chaos, with apparently nothing working, no e-mail, no web access—the whole thing had gone to pot. The best thing I could have done was review the configuration somewhere out of the way. However, after 10 minutes I gathered some vital information that could help. On inspection of the IDS policy, I had found every box ticked and therefore enabled. This included commands such as:*

- HTTP get
- HTTP put
- SMTP HELLO

*This was definitely not good. Every time anyone sent an e-mail (SMTP HELLO) or accessed a Web page (HTTP GET), the IDS would trigger an alarm. This required a closer look. Examining the action setting for each of these events revealed the problem. Each event had every conceivable action set, including the Reset option, which sends a TCP reset to the sending address every time the event fires. So every time a user connected to the site and tried to access the bank's Web page, the IDS terminated the session and sent a flood of mail and log messages.*

*It transpired that the poor administrator had never seen an IDS before and had little in-depth protocol experience. He thought he was making it extra-secure, but by simply ticking every box! While I explained the problem to the unfortunate adminis-*

trator, he repeated the immortal phrase, “Doesn’t it only affect naughty packets?” Presumably he thought that if you pay extra, you get “wickedness detection” as well! But there is a serious side:

- When tuning an IPS, know your protocols and understand the attack signatures. This was an easy problem to solve, but if you get one signature wrong, you could hunt for it for months.
- Always run the system in passive mode until you are confident you have got it right and are sure you’ve got the thresholds right.
- Only enable positive block actions, whether you’re shunning, blacklisting, or just dropping one packet, with logging and alerting. This allows you to diagnose any problems.

Note: For years I used this story as an ice breaker in courses and presentations. Many years later I went to work at a startup as head of security. How surprised was I to have this story repeated to me by the junior administrator! The location and the cast of characters had changed (with this junior admin in the starring role), but the product and the “naughty packets” had not!

## Introduction

This chapter was a tricky one to write; after all, how do you write definitively about something that has such a wide difference of opinion over definition? Many people are still convinced that the term *intrusion prevention* is marketing hype for the next-generation IDS. The best place to start our discussion is with what it is and what it does.

## What Is an IPS?

A network IDS has a uniform definition that people agree on; however, there seems to be no agreement on what an *intrusion prevention system* (IPS) is. After much searching on the Internet, I found over 300 products claiming to be IPSes, but I retrieved only one definition:

An intrusion-prevention system is used to actively stop packets of data or disconnect connections that contain unauthorized or harmful data.

Even as a working definition, this is ambiguous, because firewalls and virus scanning software would be covered by this definition, as would Layer 7 or content switches, load balancers, and SSL-accelerators.

## Active Response: What Can an IPS Do?

As a first step, I have documented the type of active response currently used in IPSes. Take a quick look at Table 11.1 and you'll find many more techniques than are generally used.

**Table 11.1** Active Responses Used in an IPS

Response	Description
IP log	Logging all packets from a particular IP address for a period of time or for a set number of packets. Great for forensics, this feature is implemented very well in Cisco secure products. It is also available in both Snorts and RealSecure.
Shun	This is old-style language for blocking a particular IP. Generally, this means blocking source address on a firewall or router for a period of time. This is extremely useful if you know a machine is long-term compromised. However, this can also be achieved on a per-packet basis in an inline IPS; see below.
Drop a packet	Sounds like a day at the races or the bookmakers', but a per-packet-basis drop significantly reduces the impact of false positives. The classic situation with the older "shun all packets from one address" approach involved one bad AOL user sending a bad packet that causes that address to be shunned. Because AOL uses a proxy system, now no AOL user can use your systems and you lose money. A per-packet drop avoids this.
TCP reset	For TCP packets only, the IPS can send a TCP reset. Arrangements are available that do the following: The reset is sent to the source IP address The reset is sent to the destination IP address The reset is sent to both the source and destination IP addresses

Continued



**Table 11.1 continued** Active Responses Used in an IPS

Response	Description
	Typically, the reset is sent to both the source and destination IP addresses. This has the effect of closing the socket on the attacker's system, but also it cleans out any embryonic connection on the attacked system.
Window size	This is a great delaying tactic. By resetting the TCP window to a small size, it is possible to reduce the rate of packet flow, thus reducing the effectiveness of many attacks. La Brae uses this technique.
Rate limiting	Window size is a type of rate limiting. Cisco routers have rate-limiting ACLs built in. A number of open source scripts can generate them.
Divert route	This can be used to divert contaminated traffic to a countermeasure or a quarantine area. This is known as the <i>sinkhole defense</i> . The Riverhead Anti-DDOS IPS uses this technique. Also, some of the "self-protecting" networks use this approach to place infected PCs in quarantine.
Kill process	In a host-based solution, the IPS/IDS can kill a malevolent process. Although I have never seen it, presumably the IDS can use the same code to send a suspend signal, or use the <i>nice</i> command to reduce the priority.
Account lockout	When a user is detected performing unauthorized activity, the HIDS can lock out the misbehaving account. This option is available in RealSecure.
Attack back	Not recommended.

## A Quick Tour of IPS Implementations

So, how do different products perform these different IPS techniques? This section divides the products into a number of different IPS types:

- Traditional IDSeS with Active Response
- In-line IPSeS

- Deceptive technology
- Extended host OS protection.

Let's review each of these categories in turn.

## Traditional IDSes with Active Response

For some time, most commercial IDSes have been able to actively respond to attacks as part of their standard functionality. Although many sites have used this functionality to deter specific attacks, this technology was never generally recommended, even in the heyday of IDSes, because the approach has some pitfalls and faults:

- IDSes have gained a reputation for being inaccurate. In passive mode, this is an irritant because the production of large amounts of erroneous false-positive alerts consumes valuable time in processing, even to the extent that real attacks could be missed. However, if an active response is used, the false positives go from being an irritant to a showstopper, because legitimate traffic will be disrupted. Consequently, most IDS users use active response only as a last resort or in special circumstances, to prevent legitimate traffic being affected.
- In many attack scenarios, by the time the IDS detects the attack, the offending packet has reached the target server and therefore the damage has already been done.
- In a number of cases, this sort of active protection has been turned against the network owner, resulting in large-scale blocking of legitimate addresses. Imagine that a hacker launches an attack and afterward notices that his IP address is blocked at the firewall. It may take him a number of attempts, from a number of different addresses, but eventually he will conclude that it is as a result of an IDS/IPS. If he is particularly malicious, he could respond by launching a large number of these attacks from a range of spoofed IP addresses, moving from one to the next after his attack causes the previous address to be blocked.

The net result is that a large number of addresses could be blocked on the firewall, effectively making your service unusable and causing more damage (in terms of breach of contract, reputation, etc.) than the original hack.

Despite the fact that the majority of IDS vendors now have the words *intrusion prevention* sprinkled throughout their literature, don't be fooled—this is a good old IDS, warts and all.

## In-Line Protection

These products are generally positioned *in line* between the client and server inspecting traffic, only allowing traffic to pass when it considers that traffic harmless—just as a firewall does. There are two subtypes in this category: in-line IDSes and application firewalls.

Generally (and it is a broad but fair generalization), *in-line IDSes* are products that have the breadth and depth of protocol coverage of traditional IDSes. Apart from the fact that they are implemented in line and therefore have the ability to block addresses without the support of a separate firewall, most do not offer many extra features over a typical network IDS. They do not generally suffer from the timing problems and abuse problems described in the previous section, but false positives are still a problem with this technology. This means that they are not as popular as IDSes.

Although there are general IPS products that address a broad range of functions and protocols, this type of product comes into its own for combating DDoS or providing application security.

Care should be used in deploying these technologies. Network IDSes were great because in nearly every circumstance, they could be deployed with minimal operational risk. In contrast, the in-line IPS affects all traffic and therefore can and will be blamed for every operational failure (he says with hard and bitter experience).

Things to consider are:

- **Throughput problems** These devices have to do a lot of CPU-intensive work. Ensure that they have the muscle for the job.

- **Connectivity issues** These devices ostensibly replace a wire, but they are not “just a wire.” They can suffer from
  - **MTU problems** Your switch is passing jumbo packets, but the IPS can’t.
  - **Buffer problems** Your IPS needs to perform reassembly to assess the packet, but the buffer is too small.
  - **Protocol problems** Your switch is passing strange Layer 2 packets (proxy ARP or spanning tree, for example), but your device can’t handle them.
  
- **Failure mode issues** Do I fail open or fail closed?

For these reasons, I recommend deploying these products first in in-line IDS mode and then turning on blocking functionality.

## General In-Line IPSes

General in-line IPSes provide good protection for common attacks that are detectable with a high level of certainty. Table 11.2 shows some of the most popular.

**Table 11.2** In-Line IPS Products

Product Type	Commercial	Open Source
General in-line IPS	ISS Proventia Check Point Interspect	Hogwash

### *Why Would I Want One?*

Fundamentally, any Web site on which transactions are made should be protected by this type of device. Sadly, this is far from the reality.

One major advantage of these products is that they can be deployed without IP addresses in stealth mode. This means that they can be deployed quickly, without the need to do anything but plug the main cable in and connect it between the perimeter router. This can add defense in depth prior to that major audit or cure that cross-site scripting problem that was going to take your programmers three months to fix.

## DDoS

As mentioned in the previous section, IDS detection engines have been heavily criticized for detecting attacks that weren't there—false positives. This means an active response is inappropriate except for the coarsest attack. Often IDSes detect DoS attacks (say, Syn floods) from simple Syn-port scans because they see a few hundred syns in a time period, which exceeds a preset limit, indicating to the IPS that it is attack. With DDoS, because the attacks use such huge volumes, even today's flawed algorithms can guarantee 100-percent accurate detection. If you let a few hundred packets through, who cares? Modern servers and firewalls will cope with that—it's the millions and millions of malevolent packets per second that they can't deal with.

In a recent DDoS attack I worked on, the packet rate rose to many hundreds of millions of packets in an eight-hour period. At this sort of rate, you can afford to let the odd 100 bad packets go by, just to be sure, before triggering the event or action.

Table 11.3 lists types of DDoS protection solutions.

**Table 11.3** DDOS Protection Products

Product Type	Commercial	Open Source
DDoS protection	Riverhead (now Cisco) TopLayer	

### *Why Would I Want One?*

If you run an online business with service-level commitments, the £100,000 needed to deploy these types of techniques might be a lot less than the cost of missing service targets.

## Application Firewall

Application firewalls generally focus on Web application traffic (HTTP, with perhaps some FTP and SMTP); hence the name *application firewall*. This specialization is probably a wise decision, since most hacks (upward of 70 percent, to be covered in the last chapter) now occur at this application level.

With this specialization comes a more in-depth level of processing, often maintaining various states and application-specific information.

The functionality of these various application firewalls differs slightly, but a best-of-breed commercial product will provide the following:

- URL/URI access lists
- Input validation at a field level
- Protection from SQL-injection and operating system command injection
- Forceful browsing protection
- Cookie poisoning protection
- Protection from common configuration flaws (such as publishing and admin functions)

Placing another network device in line positioning can cause architectural and performance problems, especially when you're dealing with load balancers, SSL, and cookies. As a result, some developers have integrated application firewall features into the Web server, perhaps as a plug in, replaceable library, wrappers, or basic front end. Table 11.4 shows various types of firewalls.

**Table 11.4** Types of Firewall Products

Product Type	Commercial	Open Source
Application firewall (network based)	Appshield, by Sanctum and F5 Networks Interdo, by Kavado	Apache Mod_Security
Application firewall (host based)	Secure IIS, by eEye	CodeSeeker

### *Why Would I Want One?*

If you are in the banking industry, you probably will buy two; you have the money and a good reason to believe that application errors will be exploited. Most other people will try to fix the application security errors at the source. This means correcting the code. Chapter 13 covers this topic more fully.

## Deception Technology

This approach to an IPS is derived from a device known as a *honeypot*. A honeypot is a decoy or a trap, a system whose sole purpose is to be attacked and to subsequently record the attack activity to which it is subjected.

Traditionally, these systems have been most successful in research projects, of which the most notable is the HoneyNet Project ([www.honeynet.org](http://www.honeynet.org)). This project provided insight into system survivability and hacker techniques and captured hacker e-dialogue that has provided valuable insight into malevolent hacker activity and motivations.

Using a similar approach, you can detect threats to your network without the maintenance or design headache. A single sensor machine can emulate multiple virtual servers running a variety of operating systems—making a fairly realistic dummy target but requiring minimal configuration because the devices usually occupy unused addresses on your network automatically.

The detection principle is different and simple:

- If you are accessing one of these decoy machines, your reason for being there must be invalid. In short, you are either there by accident or because you have a malevolent intent to enumerate a network.
- If access continues beyond various thresholds, the software discounts accidental access and takes some definitive action, which typically means blocking the address at the firewall.

### Why Would I Want One?

Although these devices provide little defense against Web application attacks, they are extremely effective at identifying and impeding traditional network-level hacking. There is also some evidence to suggest that they are good at preventing self-propagating worms. The Security Focus Web site has a number of good papers on how Honeyd and LaBrae have been used against worms.

Nonetheless, these are primarily research tools that are demanding in terms of time and resources—not for the jobbing security manager. Table 11.5 lists some types of honeypots.

**Table 11.5** Types of Honeypots

Product Type	Open Source	Commercial
Deception technology	Honeyd	ActiveScout, by ForeScout
	LaBrea	Decoy Server, by Symantec

## Extended Host OS Protection

So far, all the technologies we've described have been very network-centric. However, IPS technology exists that provides protection on a host basis.

Generally, these products use one of two approaches. The first approach is to harden the operating system by extending access control capability and privilege mechanisms, typically in a manner similar to a B2-certified operating system. This allows data access lists or system privileges to be defined with a better degree of granularity.

The second approach is to enhance the operating system's storage protection mechanism, particularly regarding stack and heap overflows. A *stack overflow* occurs when the stack (a special area of storage used by programs to hold temporary variables and environment parameters) is overwritten because of an overly long parameter, allowing malevolent code to be executed. In recent years, these buffer overflows have proved extremely costly to many companies because they often are the original vulnerability behind the very damaging self-propagating worms such as MS-Blaster and SQL-Slammer.

### Why Would I Want One?

Many early adopters of stack-protection products who started using the products as a result of the initial spate of worms like SQL-Slammer enjoyed a high degree of immunity to many of the buffer-overflow attacks that were prominent in the latter half of 2003. An organization with a large user base of older Windows operating systems might find stack protection a great interim tool. Hardened operating systems require skilled maintenance and can be useful when deployed in high-risk situations. Table 11.6 lists types of solutions designed to protect extended host OSes.



**Table 11.6** Extended Host Protection Products

Product Type	Commercial	Open Source
Extended host OS protection—stack protection	Prevx1 by Prevx	Stackshield, Stackguard
	Stackdefender by NGSec	
	CSA by Cisco	
Extended host OS protection—security kernel enhancement	Pitbull by Argus Systems	LIDS, AppArmor, Trustix

However, these products can cause things to stop working, so be cautious when deploying them.

## Example Deployments

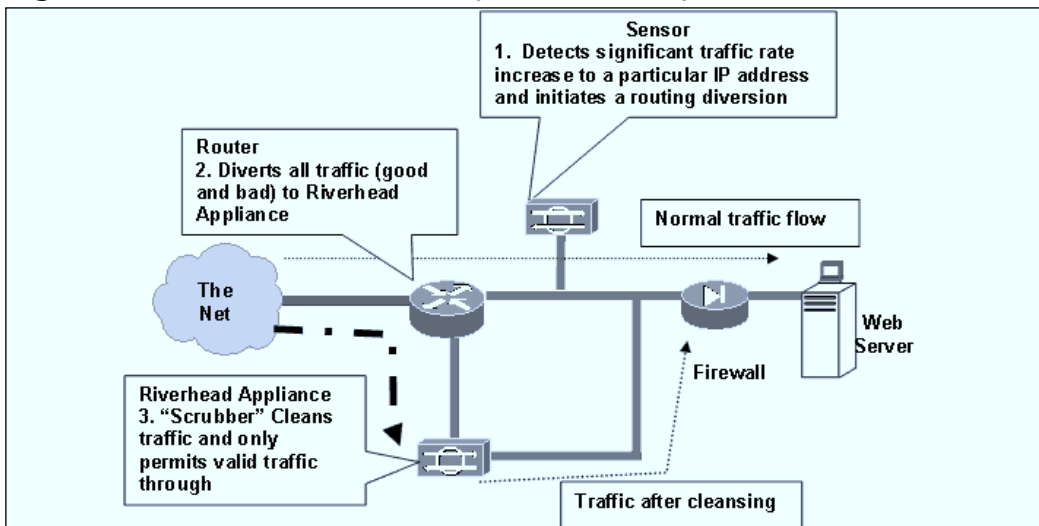
For the more technically minded reader, in this section we cover the workings of two products described in the previous section.

### Dealing with DDoS Attacks

Although the TopLayer DDOS prevention appliance is a great product, I have included details of the Riverhead/Cisco Guard system because it uses a dual approach. Like the TopLayer device, it cleans the attack by passing all affected packets through an in-line IPS device. However, the Riverhead IPS also utilizes a separate sensor that uses a route-diversion technique (see Figure 11.1).

#### How It Works

1. A basic sensor detects that an attack is under way. Typically this is signaled by a bandwidth or connection count threshold being exceeded. Obviously, different techniques can be used to detect the attack. In fact, the sensor only needs to be able to send a command via SSH to the “scrubber” that then triggers the diversion.

**Figure 11.1** The Riverhead/Cisco Systems Guard System

2. When triggered, the scrubber uses routing protocols, typically BGP, to set the IP address of the scrubber appliance as the next IP hop to the target destination. This action uses basic IP routing to divert traffic through the scrubber, which is effectively a specialized in-line IPS.
3. The scrubber initially performs a number of identification processes and then, after the attack type is identified, a number of traffic-cleansing functions (detailed later). It forwards (material) volumes of legitimate traffic and an insignificant amount of malevolent traffic. It does this using Level 2 injection to forward the traffic to an appropriate next hop (it can use “long-diversion” techniques using GRE or MPLS, but that discussion is too advanced for this text).

The reason for this elaborate diversion technique is twofold. It reduces the effect of dropping legitimate traffic. Under normal conditions, legitimate traffic flows at wire speed to the destination, not through any IPS; this means that packets aren't dropped inadvertently by the IPS, but also there are no buffering problems, no buffer overruns, no increase in network latency, no ARP or Layer 2 protocol anomalies. Your application works fine, and if doesn't, you can't blame the IPS.

This is great marketing and a fundamental truism: The reputations of IDSes and IPSEs have been damaged markedly not for the 99 percent of the packets that they pass or drop correctly—they have been vilified for the 1 percent they process incorrectly. But during a DDoS attack, you need protection or *you are out of business*. When it happens, you are inherently grateful for what you can get. The diversion technique capitalizes on this emotive response and ensures that you are reminded that it fundamentally works while it draws your attention away from any flaws that you'd otherwise notice during calm times.

The diversion techniques are also good for network providers; they mean that one scrubber can protect more than one Web farm. As long as the scrubber is fairly close geographically, no victim is going to care that a few milliseconds are added during an attack

## Scrubbing and Cleansing: The Cisco Guard

First-stage attack detection is done by the sensor. In the documentation, you will see that this is the Cisco attack detector. In practice, most sites use Netflow records (a type of accounting record) cut directly from the routers, or switches and Arbor Network's Peakflow. Once the sensor of your choice has detected an attack, the Cisco Guard takes over.

The Cisco Guard has two main functions:

- Attack identification and attacker identification
- Attack mitigation

Once the Guard begins its processing, it works on a number of key pieces of information:

- Destination host and destination subnet
- Source host and source subnet

The Guard uses the destination information to determine whether these addresses fall within a *zone* (the unit of protection on these boxes). Having established the zone, the Guard can retrieve a set of previously learned baseline profiles that categorize the normal traffic. Concentrating on the TCP traffic rather than ICMP and UDP, the anomaly engine will do comparisons on:

- Packets per second
- SYN packets per second
- SYNs/FIN ratios
- Open sessions (a bit like SYN packets per second) per source

If the Guard decides that the zone is really being attacked, it determines which traffic may be spoofed and drops all spoofed sources. It does this by a number of techniques, the most effective being TCP cookies or TCP SYN-cookies. With HTTP, it even spoofs an HTTP redirect to verify the authenticity of the sender. Obviously, if the source address is spoofed, this is a sure way to confirm it. The redirect will never happen!

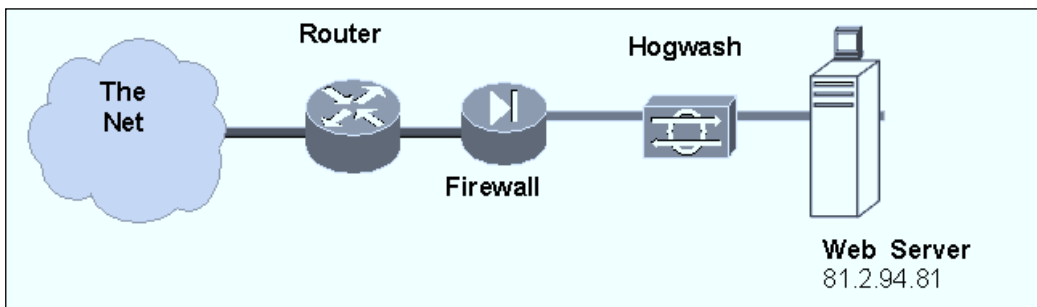
For all other traffic, TCP, UDP, and IP, it drops anomalous sources (primarily based on rate) or imposes rate limits on them.

And you know what? The system is remarkably effective for volume basic flood attack. Firewalls and other types of IPSes will just crumble at the packet rates these things lap up. However, it will do nothing to protect an application attack; for that you need a more conventional IPS, something like Hogwash.

## An Open Source In-Line IDS/IPS: Hogwash

Hogwash is a great inline IDS—by far the best open source IPS I have used. For many years I used it to protect my Web servers and modest lab, and it did a fantastic job (see Figure 11.2).

**Figure 11.2** The Hogwash IDS/IPS Protecting a Web Server on a Network



It can operate in a number of modes—notably both Layer 3 and Layer 2. However, at Layer 3 or 4, if you can live with your device having an IP

address or you need to protect a Web server, you should go for something like Apache Mod-Security.

Hogwash can operate as an in-line bridge with no need for any IP addresses, which is a lovely feature that makes its installation really plug and play.

Here is the actual configuration file.

```
##### hogwash.config #####
# interface definitions
<interface eth0>
Type=linux_raw
Proto=Ethernet
Role=internal
</interface>

<interface eth1>
Type=linux_raw
Proto=Ethernet
Role=external
</interface>

# IPLists are just lists of IP addresses for specific processing
#
<IPList WebServers>
81.2.94.81
</list>

<IPList DNSServers>
0.0.0.0/0
</list>

<IPList FATServers>
81.2.94.80/15

</list>

<IPList AllServers>
WebServers
DNSServers
```

```
FATServers
</list>

#####
<action default>
response=alert console
response=alert file (hogwash.alert)
response=dump packet (packet.log)
response=drop
</action>

#### move all packets from eth0 to eth1 & visa versa
<routing>
SBridge(eth0, eth1)

</routing>

##### end
```

*Ethernet1* is defined as the outside interface and *Ethernet0* the inside. Packets are simply forwarded across the bridge. The configuration file also shows that the default actions were (the configuration is long gone):

1. Tell the console.
2. Log to file.
3. Do a packet dump.
4. Then drop it!

These example rules, when triggered, simply initiate the default action outlined above. But it could do much more. One good feature is the ability to reroute packets. Just for laughs I found that when people attacked me, I routed them off elsewhere (I'll leave that to your imagination).

The other great feature is the “mangle” feature, which allows you to alter packets more extensively:

```
<rule>
ip dst (WebServers)
tcp dst (80)
tcp nocase (cmd.exe)
```

```

message=cmd.exe attempt
action=default
</rule>

<rule>
ip dst(AllServers)
tcp nocase(/etc/passwd)
message=attempt to retrieve /etc/passwd
action=default

</rule>

```

The disadvantage with this pig-snorter is the rules. Simple packet *grep*ing is used. For example, the rule set contains a simple directive:

```
nocase(cmd.exe)
```

This simply searches the packet for the string *cmd.exe*. When it finds the string, Hogwash produces the following alert plus a *tcpdump*-formatted dump entry in the dump file:

```

00000001 12/28/2003 21:05:28 81.2.94.82:1086-
>66.35.250.209:80 attempt to retrieve /etc/passwd
00000002 12/28/2003 21:05:31 81.2.94.82:1086->66.35.250.209:80 attempt to
retrieve /etc/passwd
00000003 12/28/2003 21:05:37 81.2.94.82:1086->66.35.250.209:80 attempt to
retrieve /etc/passwd
00000004 12/28/2003 21:05:49 81.2.94.82:1086->66.35.250.209:80 attempt to
retrieve /etc/passwd
00000005 12/28/2003 21:07:07 81.2.94.82:1097->66.35.250.209:80 cmd.exe
attempt
00000006 12/28/2003 21:07:10 81.2.94.82:1097->66.35.250.209:80 cmd.exe
attempt

```

Like good open source software, Hogwash runs on Linux. Hogwash-ing is started with the following command:

```

$ hogwash -c hogwash.config -r
/usr/local/hw/rules/stock.rules -l
/var/log/hogwash

```

## Summary

In this chapter, we complete the chapters on IDSes and IPSes by covering the kind of response IPSes can provide. These include:

- Sending reset commands or killing processes
- Dropping individual packets
- Blocking an address or address range
- Rate limiting traffic

IPSes are going to become increasingly more common, and eventually, the functionality will be integrated into other security components like firewalls and routers. Currently, however, most specialist devices are clearly differentiated by their effectiveness—the IDS/IPS vendors' products tend to work well, whereas many more utility devices prove less effective. This chapter should help you decide what end of the spectrum suits you best.

In the last section of the chapter we cover two mainstream IPSes, Hogwash and Cisco Guard. The Guard is definitely a device that will become more commonly used among quality network providers because it provides a well-thought-out solution to the problem of keeping e-businesses in business. Why Hogwash? Well, it's an outstanding product that will give the reader with time on his hands an insight into the most exciting part of network security.



## Network Penetration Testing

The purpose of this chapter is to:

- Provide a working definition of penetration testing (pen testing) and show how it differs from a hacker attack
- Outline a methodology and overview of the techniques for doing pen tests
- Describe the paperwork you need to protect yourself

## Anecdote

*In 1996, when pen testing was very new to the commercial world and covert pen testing was unheard of, I was part of a team that was doing a covert test on a major securities business in Luxembourg.*

*It was one of the best jobs of my life. Knee-tremblingly exciting work, three weeks in a five-star-plus hotel, flying back home every Friday afternoon on business class flights with free champagne—a real consultant’s dream.*

*But the client really got their money’s worth. After about two weeks we had hacked our way into everything (apparently one of the systems directly affected the Dow Jones and various exchanges), but it wasn’t difficult. The Solaris boxes all ran unsecured NFS, RLOGIN, with Rhosts files on NFS-mountable home directories. It was really adventurous to do a Mitnick-style hack at a time when most of the full details of his exploits were not public. Even the IBM mainframe had a user named “IBM” with a password of “IBM.”*

*Eventually, we called up the head of security and invited him over to where we were working to show him the bad news. He wasn’t able to come at once, so we busied ourselves while we waited, trying new tools and exploits. Unfortunately, one of them had a sting in its tail; it sent warning messages to a number of consoles.*

*Brian, the head of security, arrived. We were running through things when a riot seemed to start outside. One minute later, a large man in a black suit and someone who identified himself as the IT security manager burst in, demanding to know who we were.*

*We identified ourselves as Big Six consultants working on the PeopleSoft implementation that was going on next door. They weren’t having any of it, and behind the large man, through the open door, I could see a growing group of rather agitated staff.*

*I tried to convince the IT security manager that we were good guys because we were with the head of security. But as far as he was concerned, we were criminals and the security head was colluding with us. Then someone called out in bad French to call the police.*

*Trapped in this second-floor office, I jumped up and locked the office door. If things turned really nasty, at least we outnumbered them. I then whipped out our “letter of authority” from my jacket pocket, which had been signed by the FD and the CEO. I handed the letter over to the man in the black suit, stating that the operation was covert but we had the highest possible authority.*

*The stern recipient of the letter broke out into a broad smile. “They aren’t hackers,” he exclaimed. “Only consultants can run up expenses like these.” A white and pudgy*

*finger, altogether less threatening, pointed to a line item. “I think that bar tab looks like criminal activity, however.” Instead of the letter of authority, I’d handed him my hotel bill. We were staying at the Hotel Le Royal—that’s where prime ministers and other heads of state stay, so it was a bit pricey.*

*But the last laugh was on us. When they called the CEO to confirm the details, he denied all knowledge of the activity, because he wanted to see how the PR department handled the incident. He was paranoid after a series of Nazi gold-laundering allegations against the company. The IT director, incensed because she had not been told about the security test, insisted that we be arrested when the police arrived. We were only saved from a cell because a couple of police officers went around to the FD’s house and demanded that he confirm that it was his signature on the letter of authority.*

*But there is a serious side:*

- *When you are doing any pen test, make sure you have the correct backup paperwork; in many companies, pen testing is technically criminal activity.*
- *Make sure that the scope of your pen test is fully understood and the risks are managed.*
- *Be particularly careful over covert work such as internal hacking or social engineering. These activities can violate individual’s rights and can put the tester in harm’s way. In the real-life situation that I just described, the organization lost a top-quality IT director as a direct result of covert testing.*

## Introduction

What is penetration testing? A *pen test* is a test to ensure that gateways, firewalls, and systems are appropriately designed and configured to protect against unauthorized access or attempts to disrupt services. These objectives can be achieved by audits and critical review; however, penetration testing can augment these activities to reveal flaws that might not have been uncovered by such exercises. Pen tests also provide an evaluation of *time-based security*—in other words, from a hacker’s perspective, how long can the system survive or how long will the attack remain undetected?

Effective penetration testing requires experienced practitioners with current knowledge of attack strategies and the use of specialized testing tools. Such teams are sometimes known as *ethical hackers* or *tiger teams*.

# Types of Penetration Testing

A penetration test evaluates access controls, but there a number of distinct types: network pen tests, application pen tests, periodic network vulnerability assessments, and physical security tests.

## Network Penetration Test

A *network pen test* is a test of your critical network infrastructure. Internet-based penetration testing is the most common of this type and often used when new e-applications are deployed. Much of this chapter covers the testing of your Internet-facing systems from a remote lab located somewhere on the Internet.

However, network penetration testing can also be performed against a wide variety of infrastructures and from a number of different perspectives. Examples include covert tests of your local area network (LAN) or an open test of your wide area network (WAN).

## Application Penetration Test

Network tests are important, but their significance and “sexiness” have declined in the last few years. These days most firewall products provide great protection at the network level; after all, these products are tested by security experts before they are released to the market. Your bespoke applications might not be exposed to this rigor.

*Application penetration testing* is that missing expert test. It is designed to check the security of the applications developed by your system development staff, who usually have no security training, to provide some confidence that, for example, users will not be able to inspect other users’ confidential details.

## Periodic Network Vulnerability Assessment

A *periodic network vulnerability assessment* is not fully intrusive and should be used only to augment a full penetration test. Often, this service is little more than a variety of automated scanners scanning your address ranges on a quarterly or monthly basis and reporting on any changes or new exposures.

## Physical Security

Many organizations rely on the *physical security* of their building to a greater extent than their firewalls, yet they test their firewall but not their physical security.

## Network Penetration Testing

The most common target of a network penetration test is an Internet firewall or gateway. This section details a structured test of such a gateway.

### An Internet Testing Process

Figure 12.1 shows a high-quality structured approach to attacking a network. Most quality methodologies cover the same stages.

### Test Phases

In practice, many testers simply run a variety of automated vulnerability scanners, such as ISS Internet Scanner and Nessus. If they have some pride, they might check the results. Where a more disciplined approach is required—for example, in the U.K. government's CHECK scheme—a test will typically follow a set series of phases. The first two phases are usually known as *discovery*.

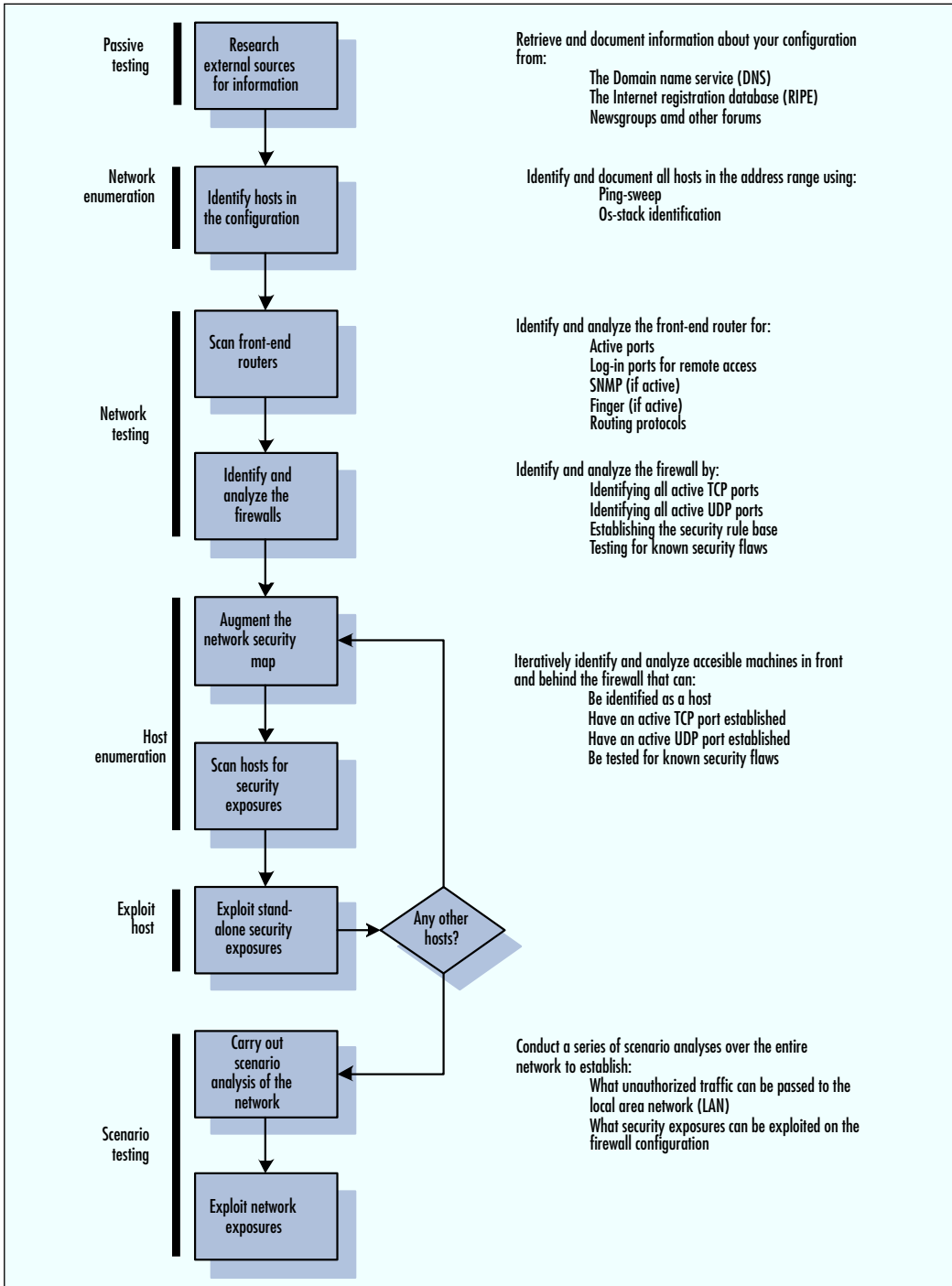
#### Passive Research

Passive research is a technique used to gather as much information as possible about an organization, without doing any kind of invasive testing. Passive research is generally performed at the beginning of an external penetration test. It involves retrieving information about the configuration from public domain sources such as DNS records, name registries, ISP looking-glass servers, Usenet newsgroups, and plain old Google.

##### *DNS Records*

Domain Name Service (DNS) records can provide contact details (and possibly usernames) of key staff members and help identify systems and the network's addressing scheme. DNS also will reveal mail relay systems and sometimes hardware types.

**Figure 12.1** The Penetration Test Process



## *Name Registries*

There are currently five registry organizations within the Internet Number Resource Organization (INR). These are:

- Réseaux IP Européens (RIPE) is the regional registry for Internet numbers in Europe.
- American Registry for Internet Numbers (ARIN) is the regional registry for Internet numbers in North America and Canada.
- AfriNIC is based in Mauritius and serves the African Internet community as the regional registry for Internet number resources.
- APNIC is the regional Internet registry that represents the Asia/Pacific region.
- Latin American and Caribbean Internet Addresses Registry (LACNIC) is the regional registry for Internet numbers in Latin America and the Caribbean.

Most of these groups provide a *whois* service that will reveal network ranges and key staff members.

## *ISP Looking-Glass Servers*

ISP looking-glass servers provide a wealth of information about BGP, routing, traceroutes, and anything that an ISP thinks is useful information to help customers and other ISPs debug problems. When security testers are researching a network, that information is manna from heaven.

## *Usenet Newsgroups*

You can obtain information ranging from the mail headers of Usenet postings to actual system configurations. People are careless with the information they give out.

## *POG (Plain Old Google)*

It's worth doing a Google search or searching the hacker archives for previous hacks. Sometimes people don't know they've been attacked!

## Network Enumeration and OS Fingerprinting

*Network enumeration* is a posh name for network mapping. Network mapping creates a picture of the configuration of the network being tested. A network diagram can be created from which you can infer the logical locations and IP addresses of routers, firewalls, Web servers, and other devices.

There is no mystery here. The information obtained from passive research is combined with data from tools such as *ping*, *visual route*, and *traceroute*. This information provides a reasonably accurate network map. For example, say you were testing [www.loud-fat-bloke.co.uk](http://www.loud-fat-bloke.co.uk):

- You could ping it to confirm the address. Pinging it several times would confirm that it had one Web server. If different addresses are returned, round-robin DNS or some other load-balancing technique may be used.
- *Traceroute* to it will give you the previous three hops. One of these will typically be a firewall and a perimeter router.
- *Traceroute* to HTTP port 80 with *visual-route* (or *lft/mtd*, the shareware UNIX equivalents). This will confirm the results and deal with any ICMP manipulation.

After you perform these easy steps, you will have a pretty good idea of what the network looks like.

You can confirm any guesswork by “fingerprinting” the operating systems. Each operating system responds in a different way to network requests, and this helps you identify them. For example, OSes have different default TTLs on network requests or different ISNs for TCP packets. These traits allow a fairly accurate guess to be made. Tools such as NMAP and POF can perform this function.

## Host Enumeration

In theory, firewalls, routers, and Web servers are types of *hosts*. Typically, they could be all treated the same way, but because I’m odd, I like to make a clear distinction and test devices in three distinct classes:



- Firewall
- Router
- Server

This approach allows for a more tailored test. In nearly all cases, this test will start with a port scan for TCP and UDP services on the target.

### *Port Scanning*

*Port scanning* is a technique to identify the services that are available on a machine. This scanning will reveal the function of a computer (Web server, mail server, and so on) as well as revealing ports that might be open. Typically, a computer will only be attacked on a port that is open, so by identifying open ports you are identifying types of attacks to which the computer might be susceptible. Port scans can be classified in a number of ways, including Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Remote Procedure Call (RPC) port scans.

There are a variety of TCP scans that vary in accuracy:

- **Connect() scan** *Connect* is the network equivalent of the open (file) system call. It doesn't send any data packets, but it does establish a connection. This method is very slow because a failed connection has to timeout. This process can take around 10 minutes per port. *Connect* is the most accurate method, but nobody uses it.
- **SYN (or half open) scan** This initiates the TCP three-way handshake. It sends a SYN packet and then listens. If it receives a SYN-ACK, we know the port is open. This is the scan of choice.
- **RST (or Xmas-tree) scan** Rarely used, this scan is not accurate but is difficult to detect. This is similar to the SYN scan but uses a TCP reset.
- **TCP window and ACK scan** The ACK scan is similar to an Xmas-tree scan, but uses the TCP ACK flag. The window scan is exactly the same as the ACK scan but detects window size changes on open ports, to provide further corroboration of results.

- **FTP-Bounce scan** This one is never used. It employs an open FTP service to initiate a connect to a specific port via the FTP port command. Can be used for scanning behind a firewall.
- **Zombie-bounce scan** This is another bounce scan, sometimes known as the “idle” scan. Put simply, you find an idle server on the Internet and then bounce spoofed packets off it. Because of IP fragmentation, ID number generation on the zombie host is predictable and increments only occur when talking to open ports. A port map can then be produced.
- **UDP scan** A UDP scan is usually useless because it relies on Internet Control Message Protocol (ICMP) port unreachable messages that good practice dictates firewalls block.
- **RPC** Dynamic ports used by RPC should be scanned with a tool such as RPCinfo.

Nmap is the tool of choice for all types of scan. MingSweeper is an alternative but only because it was written by a mate.

After running your scans on each host, you’ll know what is available to exploit. But I tend to do the routers and firewalls first.

### *Routers and Firewalls*

Routers and firewalls deserve special attention for a number of reasons. First, spotting a badly configured firewall or router saves a huge amount of effort—and because of their significance, a good tester will alert you to them as needing more urgent attention. They also give you a feel for how the site itself is configured, allowing you to tailor the rest of the test accordingly. For example, if you detect that a firewall is blocking all ICMP, you can avoid mass UPD port scanning and target specific applications. Alternatively, if the firewall is sending back “Administratively Blocked,” you can use this information to map hosts and firewall rules. Who blocks traffic to nonexistent hosts?

Here are a few extra techniques you may use on routers and firewalls:

- Routers and switches are particularly vulnerable to specific attacks that aren’t covered by most vulnerability scanners. Yersinia is a good

router/switch-specific package, but most of the sexy stuff needs to be done locally. Manipulation of services such as CDP is often valuable.

- Firewalking allows you to identify the access lists or firewall rules that are running on the screening device.
- SNMP hacking on a standard UNIX server often isn't a valuable exercise. Typically it's set read-only. On routers, it can result in pay dirt. Often you can shut the router down or get it to send you its configuration.
- Often, firewalls and routers run IPSec, and IPSec uses IKE to set up key information. Unfortunately, it is hard to restrict this access and keep users mobile. Use of IKE scanning can identify make and type of IPSec endpoint (manufacturer) because this information is sent in negotiations.

## Vulnerability Scanning

Good vulnerability analysis requires automatic tools plus human analysis for verification. Don't listen to those software salesmen who say different.

*Vulnerability scanning* is generally a fully automated method of identifying security weaknesses on a system. This is performed by tools that will test for a multitude of potential weaknesses very quickly, reporting on those that are found. Assuming that the scanning software is up to date, this testing will check for most security problems on any open service. General-purpose scanners that check many aspects of a system, such as ISS Internet Scanner and Network Associates CyberCop, are available. However, vulnerability scanners designed for specific services are also available, such as Whisker, which checks for weaknesses specifically in Web servers.

After running such tools, a good tester will verify that the service is truly vulnerable and able to facilitate intrusion. Be prepared for a lot of hard work here. Forty percent error rates are not unusual. I use two automated tools—and correlate the results.

By the end of this stage, the tester will have a map of hosts and their open services, plus a list of real vulnerabilities on each system. At this time the tester may also realize that some more testing may be required, so there's another iteration through the process for, say, a newly discovered host.

## Scenario Analysis

*Scenario testing* should be performed subsequent to vulnerability testing. Scenario testing involves exploiting identified security weaknesses to perform a system penetration that will produce a measurable result, such as stolen information, stolen usernames and passwords, or system alteration. This level of testing assures that no false positives are reported and makes risk assessment of vulnerabilities much more accurate. Moreover, you often find that two low-priority vulnerabilities could be combined to result in much high overall exposure.

Many tools exist to assist in exploit testing, although the process is often highly manual. Exploit testing tends to be the final stage of penetration testing. It may involve one of the following techniques: brute-force attacks on authenticated services, denial of service, network sniffing, spoofing, or Trojan attacks.

### *Brute-Force Attacks on Authenticated Services*

*Brute-force testing* can be performed once you have identified username and password challenges on a system—for example, at a Telnet login prompt. Brute-force attacks involve trying a huge number of alphanumeric combinations, with the objective of guessing a valid user/password combination.

Brute-force attacks can overload a system and stop it responding to legitimate requests. Additionally, if account lockout is being used, brute-force attacks may close the account to legitimate users.

Look for a more sophisticated attack here. Typically:

- Many systems don't lock accounts and have well-known default accounts—root and enable. A full dictionary attack can be applied here.
- Some systems advertise their users—for example, lists of employees and a predictable naming scheme for usernames (as on eBay). Construct an attack that parses the whole community just once for an obvious password like *123456*. Leave it a day, and try *qwerty* as a password for every account, and so on—that's the way a hacker would do it. After a week, you'll have some accounts.

## *Denial of Service*

Denial-of-service (DoS) testing involves attempting to exploit specific weaknesses on a computer that will cause it to stop responding to legitimate requests. This testing can be performed using automated tools or manually. DoS attacks can take advantage of a number of inherent system vulnerabilities, including these commonly used ones:

- **Resource overload** These attacks intend to overload the resources (such as memory) of a target so that it no longer responds.
- **Flood attacks** These attacks involve sending a large number of network requests, with the intention of overloading the target. This can be performed via:
  - Internet Control Message Protocol (ICMP), known as “smurf” attacks
  - User Datagram Protocol (UDP), known as “fraggle” attacks
- **Half-open SYN attack** This involves partially opening numerous TCP connections on the target so that legitimate connections cannot be started.
- **Echo loop** This involves getting a host to send itself endless information. Such attacks include `land.c` and `arnie.c`.
- **Other resource consumption attacks** can include:
  - Mail bombs that fill `/var` temporary space on mail relays
  - Sysfoggging—filling syslogs on target machines
  - Long URLs—exhausting CPU on Web servers by processing long URLs
- **Out-of-band attacks** These attempt to crash targets by breaking IP header standards
- **Oversized packets (Ping of Death)** The packet header indicates that there is more data in the packet than there actually is.
- **Fragmentation (teardrop attack)** Sends overlapping fragmented packets (pieces of packets) that are underlength.

- **IP source address spoofing (land attack)** Causes a computer to create a TCP connection to itself.
- **Malformed UDP packet header (UDP bomb)** UDP headers indicate an incorrect length.

DoS testing tends to be performed while systems are still in development, shortly before they go live. This testing is generally performed against critical Internet-facing devices such as perimeter routers, firewalls, and hosts in a DMZ such as Web servers and mail servers. DoS testing performed against live production systems must be managed carefully, to ensure as little disruption as possible.

### *Network Sniffing*

*Sniffing* is used to capture data as it travels across a network. Sniffing can capture specific information, such as passwords, or it can capture an entire conversation between specific computers. Sniffing is performed by a computer whose network card is in promiscuous mode so that it captures all data being sent across the network.

Sniffing tends to be used on internal testing, where the promiscuous mode computer is directly attached to the network. A great deal of information can be captured this way. Sniffing can be performed by a number of commercial tools such as Network Associates SnifferPro, TCPDUMP, and Ethereal.

More recently, with the introduction of switch networks, sniffing has been achieved with ARP flooding and ARP spoofing. Tools include DSNIF, Bournemouth, and Ethercap.

### *Spoofing*

*Spoofing* involves sending packets to a target computer that has a faked source address. This address can be at a number of layers on the OSI model, but typically it's the MAC or the IP address that is spoofed. This technique is used in internal and external penetration testing to access computers that have access controls that restrict communications to specific computers. Since most firewall systems rely on access controls determined by IP address, this can result in unauthorized systems usage.

Typically, the forms of this attack are:

- **IP spoofing** Typically, UDP-based attacks on TFTP or SNMP.
- **IP spoofing and session hijacking** Attacking Telnet and FTP with tools such as Sniper that spoof TCP/IP address and use sequence number prediction.
- **ARP spoofing** LAN-based MITM attacks.

### *Trojan Attacks*

*Trojans* are malicious programs that are installed on a client computer without the user's knowledge. They are generally sent into the network as e-mail attachments that users install without realizing they're doing it. Once installed, Trojans can open remote control channels to attackers or capture information. Testing involves attempting to send specially prepared Trojans into a network. This isn't done that often.

### *General Vulnerabilities*

The catch-all phrase *general vulnerabilities* describes the most typical of all vulnerabilities, the headline getters—the buffer overflows, shell escapes, and path traversals that make life so exciting. However, for our purposes, we only need to see if the target systems are really vulnerable to these attacks, or was the vulnerability scanner just producing another false positive?

## Reporting

Remember, this book isn't meant to be an instruction manual on how to perform pen tests. This book, and therefore this chapter, is designed to explain key details to a novice security manager; therefore, the *report* is by far the most significant output from this process.

A good pen test report should not include printouts from scanners but entirely consist of analysis. Each issue raised should provide:

- A clear description of the issue
- A description of potential impact
- A rating, usually fairly arbitrary but worth a stab
- A number of potential mitigations.

An overall executive summary is also worth having. It gives management an idea of how much time should be devoted to fixing the problems.

## Internal Penetration Testing

An internal penetration test will follow much the same stages as a network pen test:

1. Discovery/passive research
2. Network enumeration
3. Host enumeration
4. Exploitation/scenario analysis
5. Privilege escalation

Only the last element, privilege escalation, is new. Typically, it isn't done on external testing, since in that case *a win is a win*. In internal testing, penetration is practically guaranteed, so usually the aim of the test is to gain privileged system access. This is a point well worth emphasizing so that you can frame the test in the correct light to senior management. In internal testing, a system penetration is *expected*; the test is to determine the weak points (how long it takes) and how quickly it is detected. After all, your physical security means that only authorized parties have access to a LAN connection.

## Application Penetration Testing

Even with a fully tested firewall in place, many serious security flaws may exist in an e-commerce configuration. Your Java or ASP applications often contain passwords, comments explaining quick fixes, back doors, and poor authentication mechanisms that can be readily exploited by a skilled user. An *application penetration test* can help here.

### Application Pen Test versus Application System Testing

The question is often asked: What is the difference between an application pen test and application system testing? The simple answer is that there need



be no difference between systems testing of your application's security and an application penetration test. In fact, I would recommend that security testing be integrated into your system development life cycle.

Generally, however, the difference between penetration testing and the testing of security is that during penetration tests, the objective is to get in by any means possible—by changing code, variables, and input validation. Indeed, much of the investigation done during this work is to establish whether the designed regime is sufficient to prevent the bad guys from obtaining access.

During the testing phase of most new applications, the objective is to establish whether the security routines and validation work to specification, not whether they are sufficient to prevent a hacker from getting in. Most development teams do not possess the skills to do this, and if they did, they would not be impartial.

Code walk-throughs are a rare occurrence in modern development, but without them it is unlikely that anything like embedded passwords would ever be discovered. Figure 12.2 diagrams the application penetration testing process.

Several common tests are completed as part of the application penetration testing: code examination, multiple concurrent logons, session stealing, persistence of information tests, timeout tests, brute-force and DoS tests, and form poisoning.

### *Code Examination*

A *code examination test* examines code for client-side script errors or comments that present a risk.

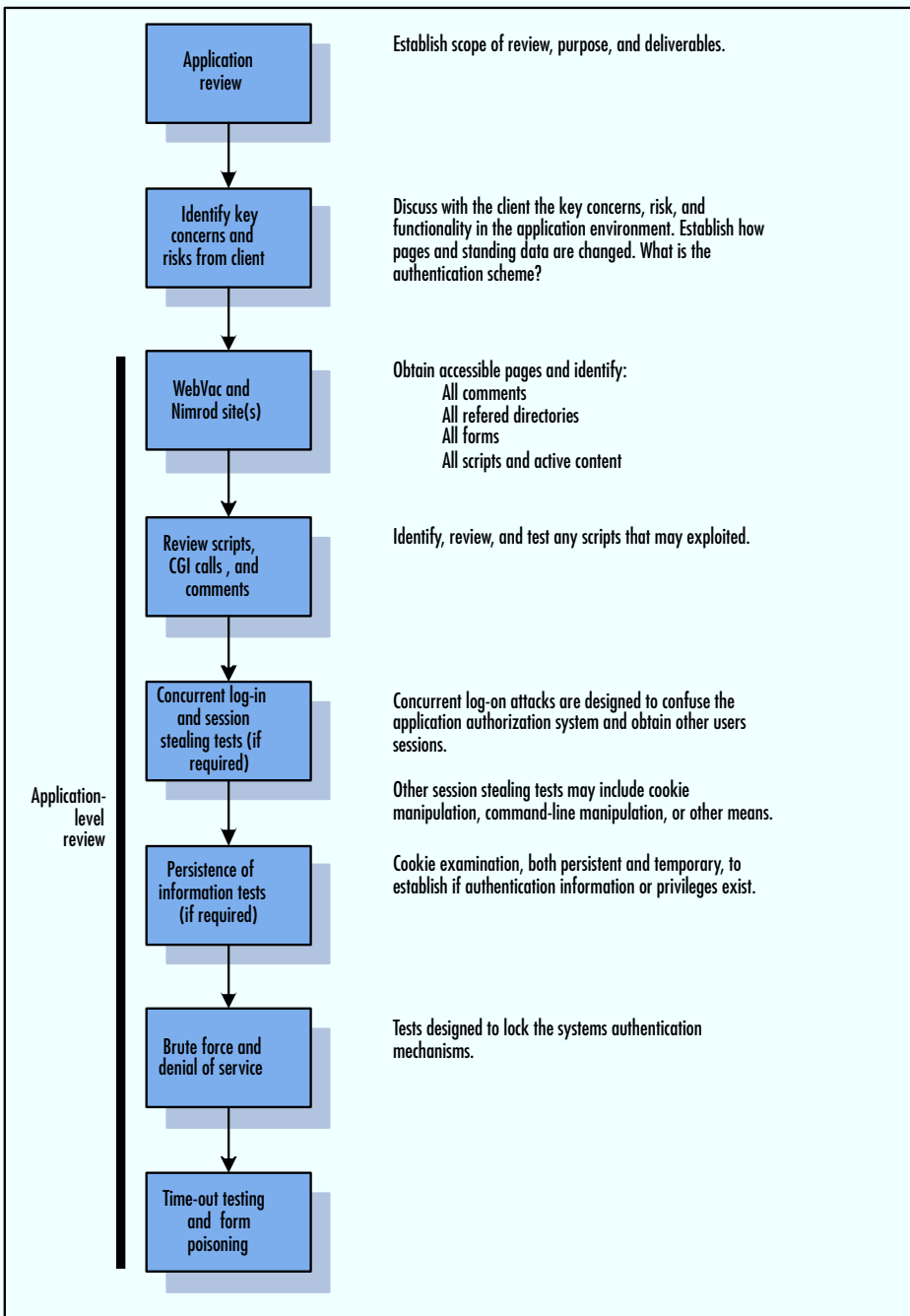
### *Multiple Concurrent Logons*

The *multiple concurrent logons test* was designed to confuse the system and obtain inappropriate sessions from it. This can take the form of either repeated tests from different stations or repeated tests from the same station.

### *Session Stealing*

*Session stealing* is often conducted in conjunction with the persistence tests (described in the next subsection) to ascertain whether it is possible to defeat the authentication mechanism and steal sessions. This can be achieved by cookie manipulation, command-line manipulation, or other means.

**Figure 12.2** The Application Penetration Testing Process



## *Persistence of Information Tests*

Testing for *persistence of information* consists of:

- **Cookie examination** Both persistent and temporary, to establish whether authentication information or privilege persists through a session and that a session is actually disabled after being logged off. This typically is achieved by enabling cookie-monitoring software on standard browsers, to establish whether cookies are used, and then using proxied sessions via tools such as Stunnel.
- **Storage examination** Often swap files contain important information that should have been obfuscated.
- **Cache testing** If caching parameters are set incorrectly, authentication information may be compromised in the “Internet café” scenario.

## *Timeout Testing*

*Timeout tests* are used for testing the length and conditions of the timeout. They validate that users are rejected under all relevant conditions and attempt to steal a logged-off session.

## *Brute-Force and Denial of Service*

At this stage testers will attempt a DoS using lockouts. This would be achieved by authenticating to numerous accounts with passwords likely to cause that account to be locked. Will it lock out every account on the system and if so, how much disruption will be caused to the business?

Tiger teams may also test to see whether accounts are vulnerable to known text scans—for instance, is it possible to try every account on the system for a common password such as 12345? Probably, but what controls are there to warn security administrators? Have multiple screen authentication processes been deployed to prevent this. Are responses from the system non-committal so that it does not highlight which accounts actually have a password of 12345?

## *Form Poisoning*

*Form poisoning* attempts to taint any form passed back to the server, to defeat the security mechanisms or to run a system command. Classically, this includes delimiter tests that allow the early termination of string literals and the insertion of *system()* calls. It may also include variable overflows and negative value/zero value tests.

## Controls and the Paperwork You Need

Pen testing rarely causes a problem for the client or the tester, even if both are very inexperienced. But the tales of uncontrolled tests going wrong are not myths.

Before I got into security, I was the technical support supervisor for a large oil company. One day all the users on the U.S. system with last names beginning with *O* through *W* got locked out of the system. This prevented our users (whose names started with *U*, for UK) using the system. Why? Because the U.S. office was doing a pen test that day—trying a dictionary attack. Unfortunately, RACF commonly locks accounts after five bad attempts.

The correct controls should include indemnity and legal protection as well as scope and planning.

## Indemnity and Legal Protection

Always have a senior member of staff from the company being tested provide you with a warrant or a letter of authorization that clearly states that the testers are doing what they are doing with the full permission of the company's directors. Computer abuse is a criminal offense, and without such a letter, the authorities may intercede, which would result in poor publicity for the tester and the testee.

Additionally, a full contract should be in place. I believe it is fair that this contract provide the tester protection from damage that results from legitimate testing. The argument I have always used is that you can't prosecute a tester for discovering a vulnerability that a hacker could have exploited scot-free, with no legal recourse.

However, if the tester shows incompetence or deviates from scope, he should be fully liable for resulting direct and indirect loss. If he hasn't got insurance to cover such problems, he shouldn't be doing pen testing.

## Scope and Planning

A clear scope is essential. It so easy for a client to instruct a tester to “see what you can find”—but that is plainly unprofessional. Companies undergoing pen testing should provide a clear list of target machines or subnets. If the company wants the testers to “discover” any accessible machine, there should be a clear break between the discovery stage and any further testing. At this time, tester and testee can meet to analyze the systems discovered and provide appropriate written agreements for the systems to be tested, before any ingress occurs.

During the planning stage, certain points should be considered: success criteria, escalation, DoS, and social engineering.

### Success Criteria

Most people trust their testers, but some people need further verification of the pen test's success than just the tester's word. That verification might take the form of calling cards (files left on the system) or simple screen shots.

### Escalation

Some systems are so sensitive that elaborate special arrangements have to be made in case of a system penetration. Usually, it is good form for a tester to call the company at the end of the day with all the high-risk points. With very sensitive systems, however, the end of the day could be too late. For example, I once worked on a test of a drug company that was developing AIDS treatments. During a simple Internet test, we came across a list of patients doing a drug trial. This kind of thing requires immediate action, and you should plan for it.

## DoS

As standard, most testers exclude DoS tests. Testers should make it clear to clients that these tests are excluded. If you want to include such a test, restrict it to certain times.

## Social Engineering

Social engineering tests can include extensive research into the background of systems users to gain an insight into their password-forming habits or trying to trick operators or help desk users into resetting passwords. This is really the worst of all testing practices; it highlights individuals' failings and causes deep-seated resentment among employees. It will lose you staff and can infringe on employee rights under the Human Rights Act.

## What's the Difference between a Pen Test and Hacking?

The absolute key thing to remember is that pen testers work on a budget, but hackers have all the time in the world. A clean pen test report, therefore, does not mean that a persistent hacker will not succeed, because he will test the obscure and unlikely over a period of time.

Other than that, you should also understand the threat agent; what are the hacker's motives and objectives?

## Who Is the Hacker?

Dorothy E. Denning, of DEC, wrote a well-respected text, based on "hacker" interviews, that portrayed hackers as generally harmless gypsies of the Internet, moving from machine to machine to learn and get access to new technology, borrowing a few machine cycles here or a MIP there. When this book was published, I was heavily involved in the e-commerce boom, often associating with hacker groups or trying to determine what they had done to particular systems so I could repair the damage. I have also been approached by people who seem keen for me to use my talents in a less than ethical manner. I do not share this view.

I am happy to believe that these “harmless gypsies” really do exist and that it wasn’t just a bunch of lads pulling Denning’s wire (the most probable explanation, really). Even if they are as harmless as they are portrayed, by committing these acts they still may be unwittingly damaging mission- or life-critical systems, breaking laws, or depriving cash-strapped organizations the CPU cycles they desperately need.

Flatly, Denning’s book does not cover the full extent of what is happening out there. The following is my typology of *hacker types and their motivations* that I originally presented at ComSec.

## The Digital Blagger: Hacking for Profit

Blogger is London slang for a professional armed robber of the city’s fine banking institutions—a gentleman of the sawed-off 12-gauge. As long as there was money “in” computers, there would be people (blaggers) trying to steal that money. If it wasn’t for headline-grabbing cases like the famous Citibank hack of a VAX and Kevin Mitnick’s antics, most people would not believe such incidents could occur. But the fact is that most fraud these days involves some type of computer misuse.

Recently, hacking for financial gain has become more exposed. In the summer of 2004, just about every online bank and online gambling enterprise suffered a DDoS attack. Minutes or hours before the attack, the security officer or a board member would receive an e-mail demanding money. I know these stories are true; I was at some of these sites.

It’s a fact: *Some people hack for money.* There is a real underground economy out there—one in which five credit card numbers buy a compromised computer. How do you think bot-nets (compromised PCs controlled by software such as Barbed-wire) grow?

## Hactivists: The Digital Moral Outrage

Hactivists (hacking + activists) don’t like you, what you do, or what you believe. They believe that what you are doing is wrong, and so they are undertaking a moral crusade to stop you—digitally.

Off the top of my head, some reasons for hacktivist behavior are:

- **Politics** Very common; Labour and Conservative parties, White House, and so on.
- **Religion** Very common, but I won't give examples, for health reasons.
- **Exploitation** Your trading practices appear unfair to someone.

## White Hats: The Digital Whistleblowers

“White hat” hackers are the closest to Denning’s vision of the hacker. Occasionally, you will find someone hacking to, say, expose insecure software implementers, claiming it “makes the Internet a safer place.”

Usually people who want to expose the frailty of software or protocols are industry insiders. These people publish a vulnerability to the CERTS in a fairly standard manner and raise awareness in that way, as I and many others have done for years.

Generally, these people are misguided.

## Script Kiddies

If you get hack-sawed, it is almost certainly done by a script kiddie, whose characteristics are typically:

- Age between 14 and 24
- Operating after school or on a Sunday
- Has no knowledge of the victim

The modus operandi is simple: The script kiddie obtains an exploit, or a series of exploits, and then learns how to use them. He or she is not a master hacker; running these programs stretches the kiddie’s meager ability. The kiddie will scan huge address ranges on the Internet, looking for any random machine that has a particular vulnerability. The kiddie knows nothing about the targets other than that they are vulnerable. Then he or she runs the exploit against the victim’s machine; this probably involves defacing a Web server so that the kiddie can get evidence of the hack and boast to friends. Review the hacked-site archives at Zone-h ([www.zone-h.org](http://www.zone-h.org)) for more details.



There is nothing personal about this attack. The script kiddie doesn't know you or your business and doesn't want to know the effect of his or her attack on your life. Your network was figuratively "in the wrong place at the wrong time." The kiddie just wants to be famous.

The following is an extract from hacker exploits; it reads like an Oscar speech, doesn't it? I think it proves my point:

—!ADM!ROX!YOUR!WORLD!—

# special greets to trambottic, hex_edit, vacuum (technotronic), all

#!adm,!w00w00 &

#rhino9 (that's a lot of people, and they are all very elite and good friends!),

#wiretrip, l0pht, nmrc & all of phrack

# thumbs up to packetstorm, hackernews, phrack, security-focus, ntsecadvice

#I wish I could really name everyone, but I can't.

#Don't feel slighted if you're not on the list... :)

## The End of the Story

Penetration testing is a good tool for assessing network security. It has a far greater "wow" factor than a standard audit. Exploiting this testing can be good strategically for information security budgets, but use it wisely. As a tool, pen testing is more expensive and probably less revealing than a standard audit or security review.

## Summary

In this chapter, we described the purpose of network penetration testing. We covered the various types of test that you can use and the controls that you need to put in place before you engage a professional testing team.

The chapter describes a number of typical testing processes. When you engage a professional team, you should ensure that they use a structured method similar to that described here and don't just "go with the flow."

Lastly, we described the ways that a penetration test is different from a hacker attempt and tried to provide a model for some hackers' motivations.

## Application Security Flaws and Application Testing

The purpose of this chapter is to:

- Show how application security flaws differ from network-level exposures
- Provide a brief description of the common application security flaws

## Anecdote

*When I left consultancy, I was glad to do it. I was even happier when, six months later, a researcher showed me that by entering the classic SQL-injection command string into my former company's Web site, you could gain access to all manner of information. Sweet.*

*But application security is like that! Programmers “think inside the box”—the “what if I don't bother to visit the login page and go straight to the accounts page” scenario seems to defeat so many of them and make the headlines again and again. And it is simply because they aren't trained. I recommend that every security manager run a short course on application hacking; you'll soon see an improvement.*

## Introduction

Many in the IT industry have generally ignored or misunderstood application security. The facts show that it has been with us since the first mainframe applications. In those days, in the late 1970s through to the late 1980s, we used 3270 locally attached terminals driven by operating systems that had a level of security that we would accept as good today. On these centralized and generally hardened beasts, network hacking or brute-forcing passwords, in the way hackers do today to gain access to a UNIX or Windows box, would simply not get a hacker in, so no one tried. In those days we hacked applications by trying to subvert TSO clists, by pressing *Clear* or *sysreq* keys to break out of captive menus systems. When we started to develop Web-based applications, the poor security came along; now, because most applications on the Web cannot rely on users being internal to the company, the threat has increased.

The reality of the situation is that you may have witnessed these kinds of Web application security flaws yourself while browsing the Net. Examples are commonplace. Perhaps you have been in an Internet café or borrowed a friend's computer to enter a Web portal and found that the portal thinks you are someone else. Or you might have used an information terminal in a hotel or business center and, after hitting a few keys, found that you have direct access to a Windows desktop or a UNIX command prompt that allows you, an outsider, to amend or delete the data of that particular organization.

These are classic examples of application security exposures that can occur accidentally or as a result of intentional behavior. Yes, you guessed it—hacking. That is why people are excited. There has been a sudden realization that although a lot of attention has been paid to firewalls to ensure that the ubiquitous hacker can't use network-level attacks, most organizations could be vulnerable to attacks that manipulate Web servers. These people are probably right to be worried; according to analysis of our test assignments, most sites are vulnerable to this type of abuse.

## The Vulnerabilities

This chapter outlines the types of hack that can occur and the risks they can expose you to. One of the most referred to sources on application security is the Open Web Application Security Project (OWASP). OWASP is a not-for-profit foundation with a mission to advance the community's knowledge and awareness of Web application or Web services security issues. To this end, OWASP produces a list of the top 10 most serious vulnerabilities. Here's the current list:

1. Unvalidated input
2. Broken access control
3. Broken authentication control
4. Cross-site scripting
5. Buffer overflow
6. Injection flaws
7. Improper error handling
8. Insecure storage
9. Denial of service
10. Configuration management

For the purposes of this chapter, I group these vulnerabilities into three categories:

- Configuration management
- Unvalidated input
- Bad identity control

## Configuration Management

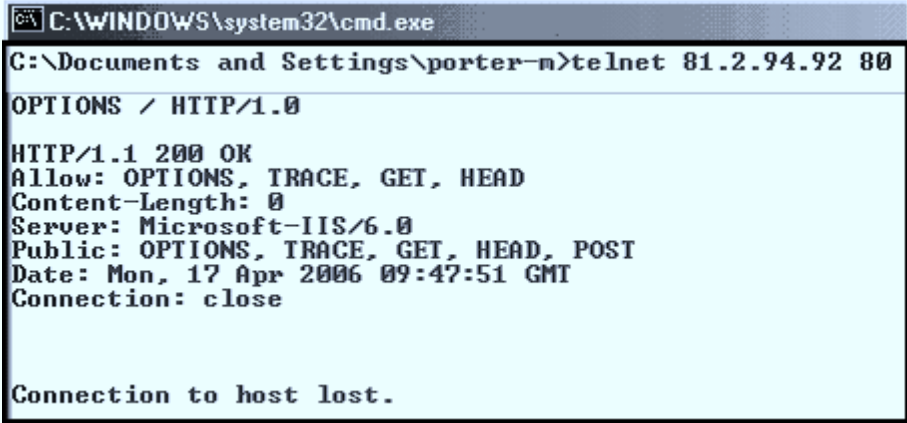
Configuration management is the easiest category to deal with and it is the only category that will *not* involve disciplining the programming team. Nearly every hardware and infrastructure software component, whether a firewall, router, or Web server (the particular case we are interested in), has a browser-based administration interface. Often, on Web servers, these are protected by only a simple password and available through TCP port 80, which often means they can be accessed from the Internet. Web authoring functionality, application servers, and content control software also commonly present problems in this area. Fixing the problems can be quite hard; if you block it at the firewall with a URL rule, you add a resource overhead and level of complexity that really would be better ignored. If you restrict access at the Web server, you'll probably end up just adding another password that cracking software like Brutus will make quick work of.

To complicate matters, this problem frequently falls through an organizational responsibility gap. Most infrastructure departments will happily take responsibility for a Web server, but if the vulnerability derives from content creation and delivery software, the situation may be very different. Development teams would rarely take responsibility for the security of software (other than the code they develop). The likely effect is that the software appears on a production server with standard defaults, which are often not very secure.

Other common examples from this category include:

- **Web server headers** that disclose make, type, and version of the server in use (see Figure 13.1).

Figure 13.1 Web Server Headings



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\porter-n>telnet 81.2.94.92 80
OPTIONS / HTTP/1.0
HTTP/1.1 200 OK
Allow: OPTIONS, TRACE, GET, HEAD
Content-Length: 0
Server: Microsoft-IIS/6.0
Public: OPTIONS, TRACE, GET, HEAD, POST
Date: Mon, 17 Apr 2006 09:47:51 GMT
Connection: close

Connection to host lost.
```

- **Unnecessary HTTP methods** *TRACE* and *OPTIONS* can be used to devastating effect as well.
- **Information disclosure** This refers to those standard error returns that can easily be used to probe the server.
- **Logs and directory structures** Common directory structures take the uncertainty out of hacking.

## Unvalidated Input

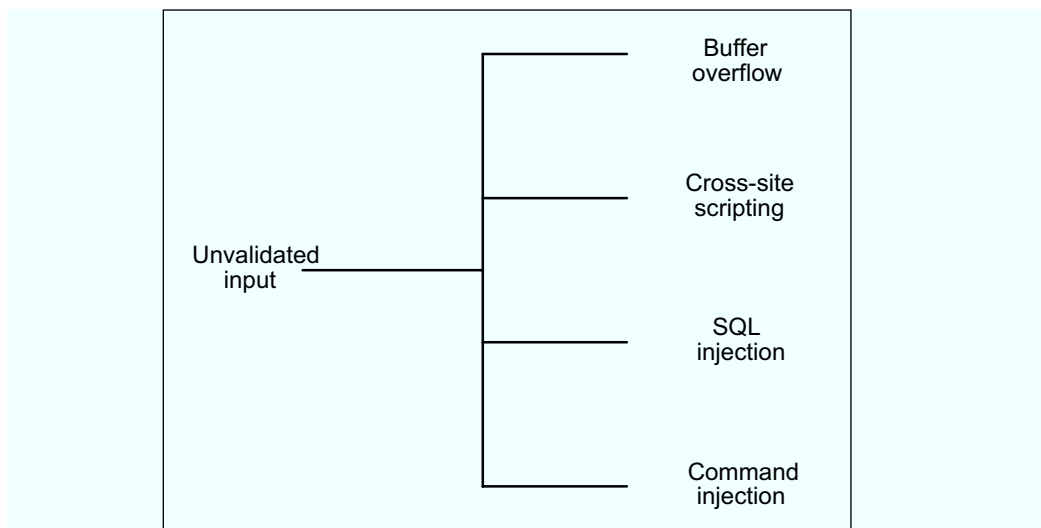
Anyone who has spent any time programming or working with spreadsheets understands the normal result of developing systems that do not validate user input correctly. Typically, the system stores or displays rubbish and then stops. The old adage of system and program design, “garbage in, garbage out” (GIGO), is as true today as it ever was.

Don’t be fooled into thinking that the error message on the machine is the only effect. Sometimes these actions result in a corporate database containing invalid or past dates and other fields generally containing inconsistencies. And if online code has insufficient validation, it is more than likely your batch processing has none at all, in which case you may have created a denial-of-service attack, which will stop that night’s batch run with arithmetic exceptions. Similarly, if you do validation in client-side JavaScript, you have

little reason to feel smug. The first thing any self-respecting hacker would do is whip this client-side code out (that's right, hackers can use Wordpad too) and resubmit the form without any validation. Then your site becomes as vulnerable as a site with no validation at all.

But this is all common sense and much more about the quality of code than “code security.” Certain omissions in validation can result in a number of much more damaging types of attacks (see Figure 13.2): buffer overflows, cross-site scripting, SQL injection, and command injection.

**Figure 13.2** Attacks Resulting from Unvalidated Input



## Buffer Overflows

Buffer overflows occur when you pass a large, overly long parameter to a program that is expecting a much smaller value. If no validation prevents it, this long parameter value will fill up not only the storage reserved for it but also surrounding storage areas. The most common result of this overflow is that the program produces an error message. However, sometimes with effort and persistence, security analysts discover situations where an overly long parameter can overwrite special pieces of storage that control execution sequence. These stack overflows allow you to manipulate various instruction pointers the program uses to remember what to do next. With clever manipulation, you can provide a parameter that will cause execution to jump from the



intended functionality of the normal program to a section of code that loads another program or command and then executes it—a dream for any hacker, being able to run a command of his or her choice.

Buffer overflows used to be discovered by accident or restricted to expert assembler programmers with too much time on their hands. Today comprehensive papers, methodologies, and utilities have been published that make it well within the reach of an average script kiddie. In fact, more than 19 percent of all security vulnerabilities reported to CERT are now buffer overflow based. To show how straightforward discovering these attacks has become, I have outlined a basic procedure here:

1. Select a potentially vulnerable target program. If it is a common program (either open source or readily available), you should install it on a local system. This will make the whole process much easier by allowing greater speed of work and allowing direct memory access.
2. Use a specially designed FUZing program (FUZing is hackers' slang for repetitive techniques, usually involving incremental length increases to parameters that are used to find buffer overflows—programs are freely available from your local hacker's site) to repetitively pass the target program longer and longer parameters until the program suffers an abnormal termination (indication of a buffer overflow). Use a debugging program to attach to the running program or analyze your dump (core) file to identify important areas of memory.
3. Using the addresses from the debugger and a simple bit of math, you can craft a parameter of a particular length containing a section of machine code. This code, known as *ShellCode*, can be downloaded readily from the Internet and, when used in the correct position in the parameter, will invoke a command shell (i.e., `/bin/sh` in UNIX or `cmd.exe` in Windows).

To illustrate the effectiveness of this process, I once asked one of my team to practice this technique. After two weeks, this expert hacker (but novice buffer-overflow discoverer) found a buffer overflow in a major e-mail program and in a popular Web application. And that's exactly what many large

security organizations or even hacker groups have their members do as routine use of their time—hence the volume of security alerts in this area.

## Cross-Site Scripting

Cross-site scripting has been known about for many years—and largely ignored because many security managers and developers do not understand the implications. A cross-site exploit occurs when a script tag is entered as a parameter, often as a URL, and sent to the Web server. If the Web server replays the unaltered script to the browser, the script will be executed.

So you enter a command; it bounces off a Web server and then runs on your machine. If the command does do damage, it will be self-inflicted damage, and that's your own fault. Thus, many security managers rationalize away any dangers of cross-site scripting. Before we inspect and shoot this particular school of thought down in flames, let's look at an example.

The most basic possible cross-site scripting (known as XSS) is:

```
<SCRIPT> alert('I was here') </SCRIPT>
```

The `<SCRIPT>` and `</SCRIPT>` tags simply define the start and end of the script body. The real functionality is:

```
alert('I was here')
```

which displays a message box, “I was here,” on the screen.

This screen image in Figure 13.3 shows the script command being entered on real application that could be vulnerable:

The screen in Figure 13.4 shows a dummy application that is vulnerable to cross-site scripting.

Figure 13.3 Inserting a Script Tag into an Input Form

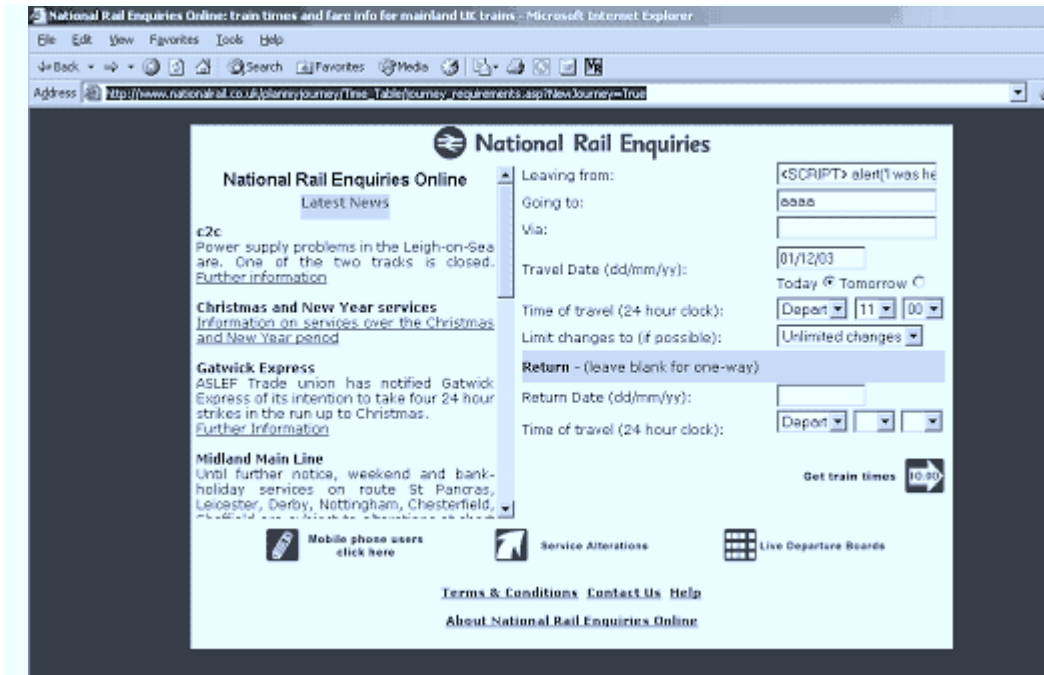


Figure 13.4 Cross-Site Scripting

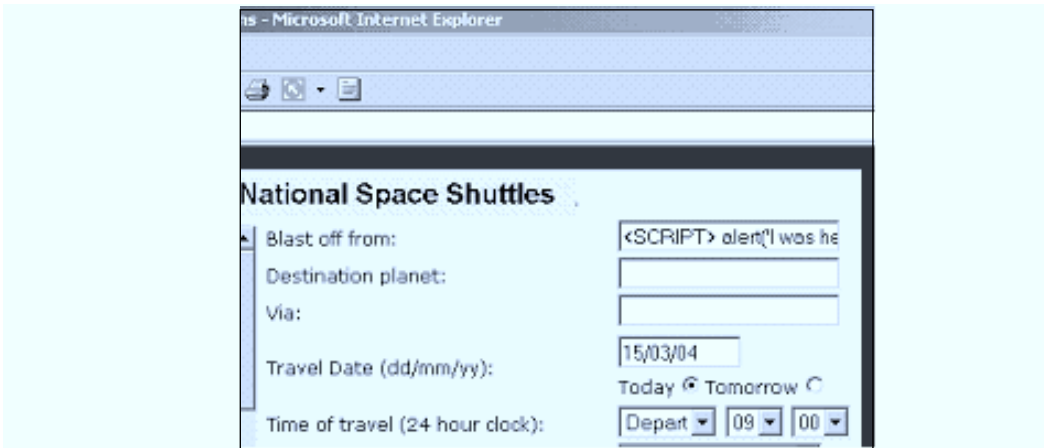
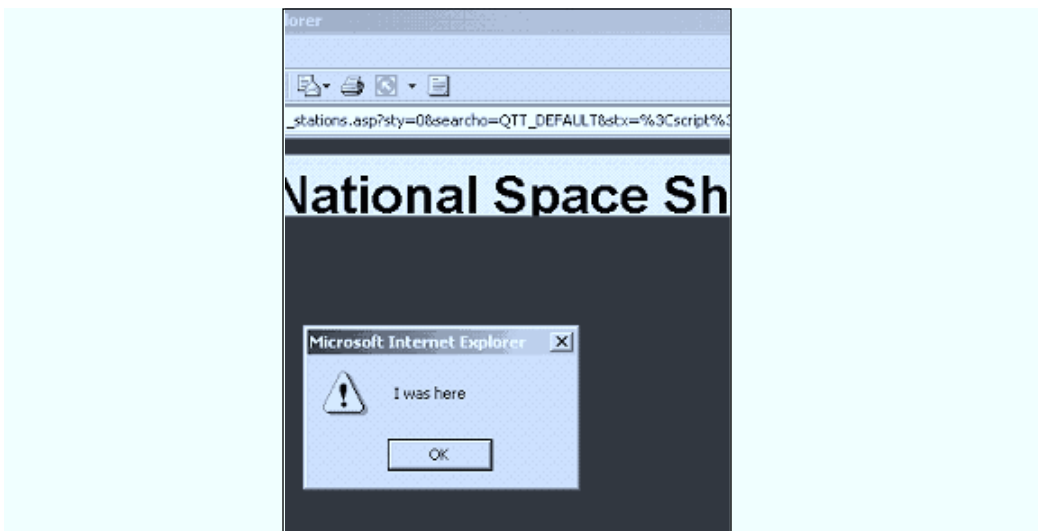


Figure 13.5 shows the results of the *script* command, with the alert box requesting user action.

**Figure 13.5** Results of the Script Command

But these attacks aren't always self-inflicted, so security managers who ignore the exposure run the risk of causing regulatory or even serious legal problems. With a small element of social engineering, the insecurity of the application will be exposed, damaging the brand of any organization that was aware of the exposure. Some of the recent phishing attacks on banks that conned users into revealing their user IDs and passwords used an e-mail as the attack vector. Okay, you could argue that clicking on an e-mail link is plain silly.

Imagine if the attack were initiated from a link on a Web site or search engine (you only need a credit card to get a sponsored link). Would that add more credibility to the attack? If a login panel were displayed from an XSS that looked exactly like the real thing and came from the support desk of the organization, who would question it, especially if when the cautious user clicked the Secure padlock at the bottom of the screen, the certificate information showed clearly that the correct organization was being accessed? Even if you are a highly security conscientious organization and use client-side, certificate-based mutual authentication (PKI), the XSS will still work.

If your application is highly configurable and you manage to plant the attack in the cookie or database that holds the configuration options, the effects can also be devastating.

Until now, the full effect of XSS hasn't been felt, but it is really just around the corner.

## SQL Injection

*SQL injection* is an attack that manipulates parameters that are used directly in SQL statements. These form fields could include any input fields that you see on your screen when you enter your customer number, address, or search phrase at your favorite Web sites. No special equipment or particular expertise is required; all you need is to learn the 10-character string and the variants that form the fundamental attack technique.

These attacks allow you to alter restrictions on the access query so you can read data that you would not normally be allowed to access. It is not difficult to see how this could lead to legal repercussions (such as breach of the Data Protection Act) or damaging publicity *simply by entering a series of normal speech marks and SQL commands at the browser*. In more serious cases, it allows attackers to insert fraudulent records or delete any record and even run commands on the target server.

For most managers who are unaware of this vulnerability, this information is *too* much information; they simply want to check whether they are vulnerable, immediately.

Now let's look at a simple code segment extracted from a vulnerable application. It simply reads employee names and telephone extensions from a database.

The variable *name*, when entered in the form, will be concatenated onto the rest of the query to form a conditional statement, which is designed simply to return one employee's record:

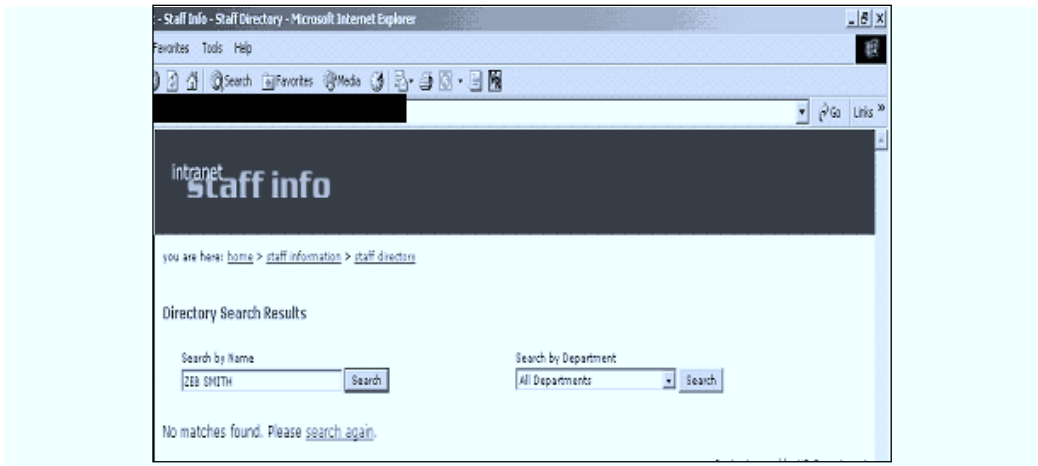
```
SQLquery = "Select Employee_Name Employee_Extno
from Employee_Table
Where Employee_Name = '" +name +'" ;"
```

If we enter *Zeb Smith* in the appropriate form field, *+name* is replaced by Zeb's name. This causes the query to evaluate to:

```
"Select Employee_Name Employee_Extno
from Employee_Table
Where Employee_Name = 'ZEB SMITH' ;
```

Because we are not lucky or exotic enough to employ anyone called Zeb Smith, the query returns no records and we receive no output (see Figure 13.6).

**Figure 13.6** Query on Zeb Smith Produces No Output

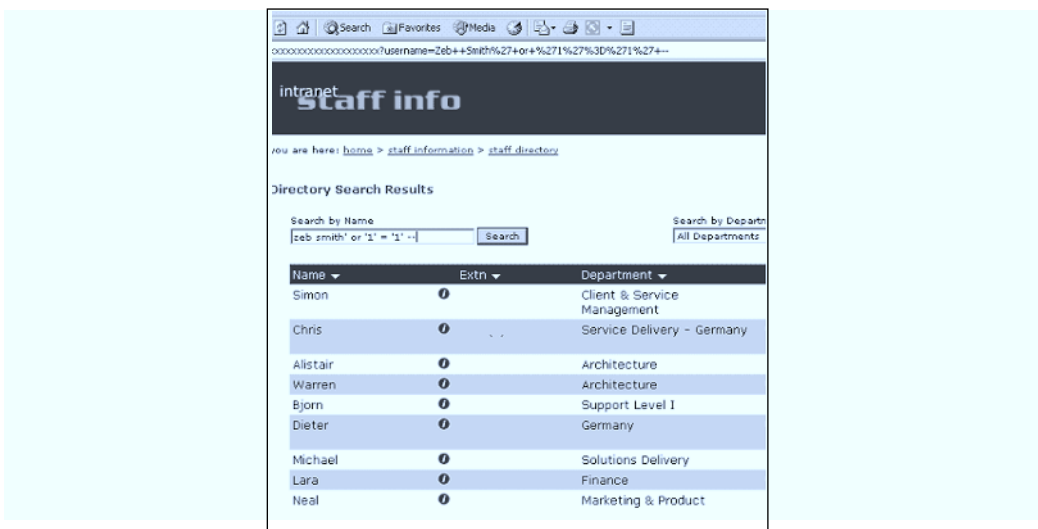


But if we enter:

ZEB SMITH' or '1' = '1' --

we suddenly receive a listing of the whole database (see Figure 13.7).

**Figure 13.7** Modified Zeb Smith Query Returns All Records



This is because we have modified the query by carefully positioning the parameters to:

```
Select Employee_Name Employee_Extno
from Employee_Table
Where Employee_Name = 'ZEB SMITH' or '1' = '1' --
```

We have *injected* a new clause of SQL (or `'1' = '1'`) into an existing SQL statement to change its behavior. The portion `'1' = '1'` is just a condition that is always true. So each record is evaluated as follows:

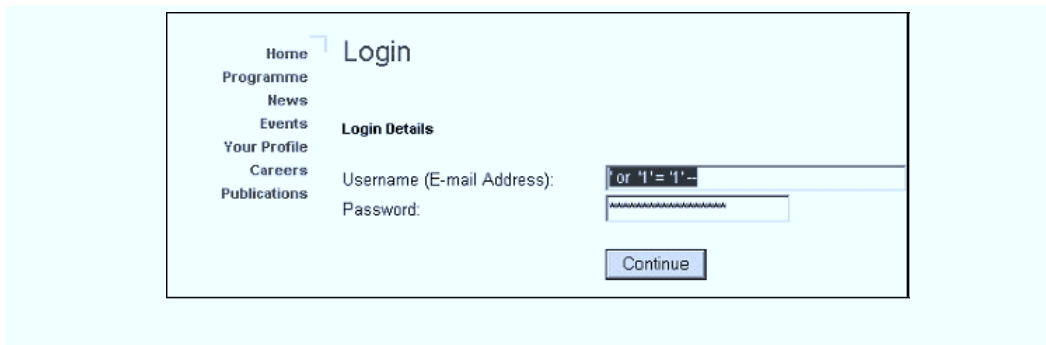
```
Is Name equal to Zeb? - no ; or
Is 1 the same as 1? - Yes
```

Thus, the whole database is returned. The double dash (`--`) is just a comment to nullify an SQL *sort by* or *group by* statement that might follow.

This technique was originally discovered and publicized in cases where the conditional expression related to user and password authentication—a one-time common failing that allowed you to look at other users' account details or even passwords.

This could have occurred in the screen shown in Figure 13.8. When we entered the correct string sequences, it might have been possible to access the details of an individual called Robert Lom. Could this be a breach of SB 1386?

**Figure 13.8** Real Example of a Site That Was Defeated by SQL Injection



However, under the right conditions, things can get a whole lot worse. You could run the:

- *Delete* command
- *Insert* command
- *Xp_cmdshell* command, which will allow you to run the command of your choice on the database server

It is easy to see how you could cause huge amounts of damage by exploiting this vulnerability.

## Command Injection

*Command injection* is very similar to SQL injection, but instead of subverting a channel (in an academic environment, this type of vulnerability is known as a *covert channel*) to the SQL interpretation, a channel is opened to the command shell or a language interpreter.

Three or four years ago I came across this little program while doing a penetration test. It was designed to display a press release. Unfortunately, it allowed us to display nearly every file on the Web server. Although I have heard claims that hacks like this would result in the complete compromise of the server, the firewall only allowed port 80 through, so despite the fact that I could download the password file (by entering the parameter as `../../../../etc/password`), the fact that I could not initiate a Telnet session to take advantage of the situation prevented further progress. In any case, the Web server was configured correctly (running as *nobody*), so listing the encrypted password file (`/etc/shadow`) would have been impossible.

```
#!/usr/local/bin/perl5
print "HTTP/1.0 200 OK\n";
print "Content-Type: text/html\n\n";
print <<"EOF";
<HTML><HEAD> <TITLE>News Releases</TITLE> </HEAD> </HEAD> <BODY> <p>start
bit here
EOF
# system("cat /WEB-DIRECTORY/$ENV{QUERY_STRING}");
open(INFILE, "/WEB-DIRECTORY/$ENV{QUERY_STRING}");
while (<INFILE>)
    { print "$_";
      }
print <<"End_Footer";
```



```
<CENTER> <p>end bit here </CENTER>
```

```
End_Footer
```

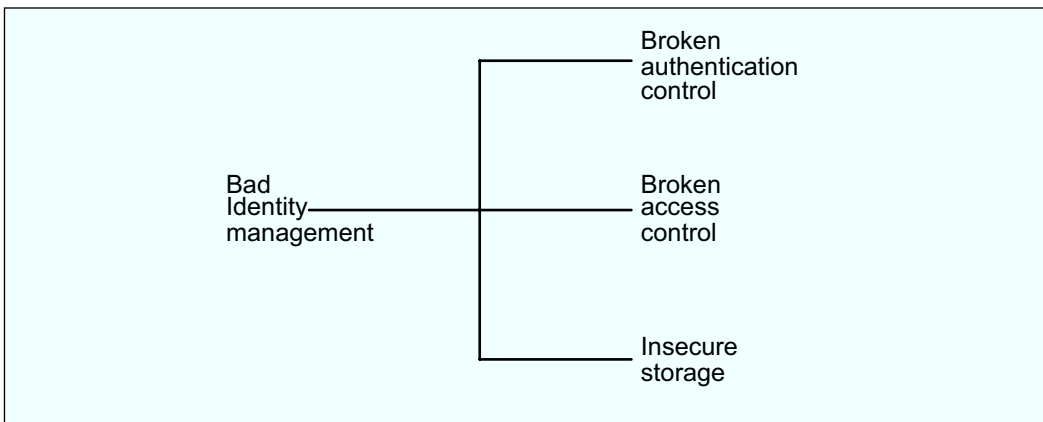
This situation occurs because the parameters are not checked before they are used in commands or statements; in Perl terms, they are called *tainted*. In essence, it is very similar to SQL injection.

Often they are far more dangerous, as demonstrated by the commented and underlined statement in the previous example. This statement had obviously been removed for security purposes. It used the *system* command, which effectively runs the parameter as an operating system command, just as though you had typed it in. If this had still been in place, we could have run any Unix command we passed as a parameter. It is unlikely that the server would have survived.

## Bad Identity Control

Under the category of bad identity control, I have grouped three of the OWASP classes of application security threat (see Figure 13.9): broken authentication control, broken access control, and insecure storage.

**Figure 13.9** Bad Identity Control

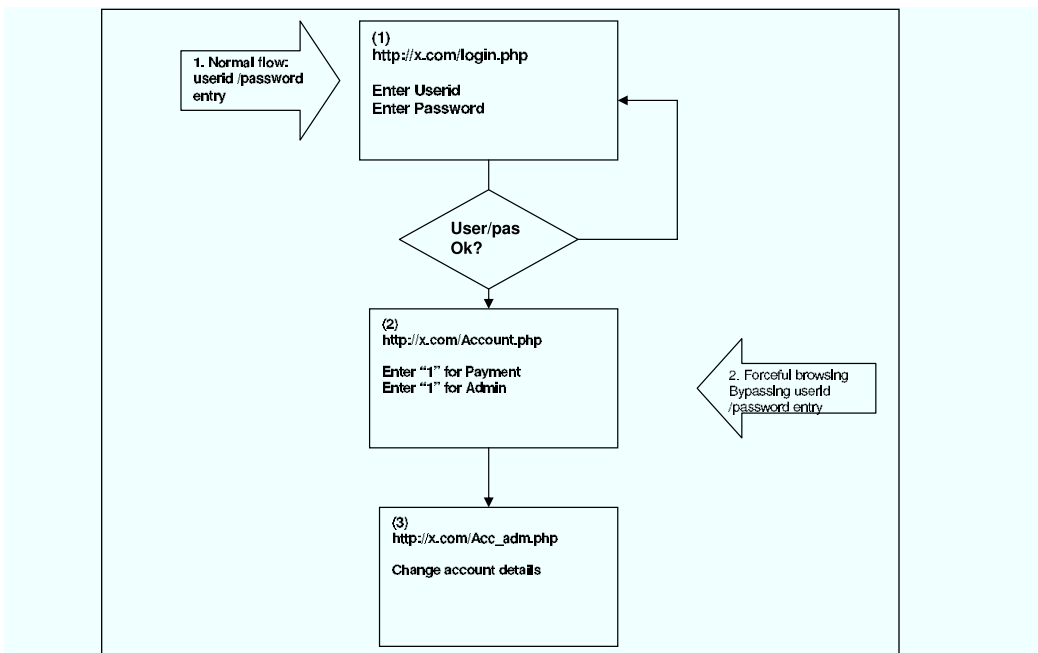


I chose this collection because often when we're dealing with Web applications and security breaches, it is not clear whether the flaw is in the authentication system or the authorization mechanism. Consequently, I will describe a number of common situations in this area.

## Forceful Browsing

In many e-commerce applications, authentication and authorization functions tend to be provided by the application and therefore not actually linked into the Web server that serves the pages (see Figure 13.10).

**Figure 13.10** Flowchart Explaining Forceful Browsing



This often results in a system design that is vulnerable to forceful browsing—a situation where a single start page provides all authentication and authorization processing. Subsequent pages and processes make no attempt to verify security credentials; they infer that these have been validated by the initial entry page. This leads to the situation represented by the flow chart in Figure 13.10—an all too common example of poor security design.

Once authenticated, the user is presented with a series of links that he or she can use (see Path 1 in the diagram). The user selects the links and enjoys using them so much that she bookmarks the individual pages. Next time she wants to use the pages, she goes straight to the bookmark (Route 2). She may receive a variety of errors or other customer details. However, in the worst

case, she simply starts using the application, which she may perceive as a great user-friendly attribute—that is, until her 14-year-old daughter discovers this handy cash-producing feature.

How does this happen? By going directly to pages deep within the application, the user has bypassed the entry page, so she has not identified and authenticated herself. Since the functions or pages in the application are available to everybody (because the authentication functions are not integrated with the OS or the Web server), she operates unhindered.

This may seem far-fetched, but I reviewed an Internet dealing system that did exactly this. They were even using complicated token-based authentication, but with a design like this it provided less than satisfactory protection. This is often most deadly when the application authorization system supports the concept of “power users” (privileged users who can perform administration). In the case of the banking system, forceful browsing allowed us access to the admin screens and allowed the tester to access administrative functions from a normal unprivileged account.

## URL Parameter Tampering

The most well known of application hacks is best described as URL tampering, and surprisingly enough it is still fairly common. Imagine in the previous example that once `http://x.com/login.pl` has processed a valid user, it allocates this user a number, which is used to uniquely identify that user. It passes this number (*uid*) as a parameter to *account.pl*:

```
Get http://x.com/account.php?uid=10
```

But imagine that a malevolent user has decided to be adventurous and experiment with changing the UID field to 11. All of a sudden the adventurous user is presented with a different user’s account. Although this is a simple example, it is common to find slightly more complicated examples (which are just as easy to crack) used in sophisticated systems.

## Insecure Storage

You can view most HTML source code from a browser. Despite this, it is amazing how many of these pages contain passwords, either in JavaScript or in comments in the code. Often you can recover this information from a Google search.

## Fixing Things

Application systems and Web applications have inherently become merged these days, with nearly every IT director demanding a Web front end for even the most sophisticated package so that the problem can no longer be considered isolated. Orthogonal development of Web applications means that they have become so complex that a complex approach is required to provide acceptable application security.

Everybody wants secure and resilient code. But what is secure code? Going back 15 years, when people moved from mainframes with attached terminals to Unix systems connected with TCP/IP, management asked, “What is a secure network?” or “What is a secure Unix server?”

Organizations answered the question by defining policies, procedures, and processes that defined what the organizations considered to be an adequately secured network and server. They produced security standards and baselines for routers and servers, then reinforced their use with awareness campaigns and compliance audits. This engendered best practices in the industry, so now if you ask an administrator to “name five things you must turn off to make a Unix box or a router secure,” most administrators will give a good answer. This wasn’t the case even five years ago.

We need to apply this technique to the coding problem. Education and awareness must be guided by management policy to encourage and reward good practice. Audit and review should be used to ensure compliance. Organizations must have clear security standards covering design and coding.

Standards aren’t enough by themselves, though. Programming teams need to be trained in these techniques to avoid the common pitfalls. Usually, a couple of days will do it; in the scheme of a major programming project, that is nothing.

Don’t believe it’s all about programming, however. The systems architects must be made to understand, too. Many complex systems need advanced security, and the corporate security team should be involved at the design stage to review arrangements made. This frequently does not happen. If you look at forced browsing or insecure storage, they are flaws that occur at the design stage, not during programming.

Organizations should undertake regular baseline scans. The traditional once-a-year penetration test is inappropriate these days, with many organizations having a once-a-week code drop cycle. The application scans should be incremental and regular (either monthly or quarterly), and the results should be related to the organization's application security standards. This approach provides an effective and tested security framework.

## Qwik Fix

After conducting tests on large applications live on the Internet, it common to find teams left with six or 10 weeks of programming effort to repair the code. That is a sizeable cost coupled with a lengthy exposure to abuse.

It would make sense to install an application firewall. However, these firewalls are very expensive compared to, say, a Cisco PIX—probably in the order of five to 10 times the cost when you consider platform, OS, and SSL acceleration. For most companies, this cost is prohibitive, but some banks do use the market leaders:

- Kavado, Interdo Product
- Sanctum, Appshield (now part of F5)

These can be effective, but certainly deployment is often difficult.

For many companies, purchasing a high-function IPS or firewall device is a better option. Traditional firewalls such as FireWall-1 NG contain deep-inspection options such as Web intelligence that provide an effective cover against most common exposures. In addition, they have the facility to add custom rules to cover the rest.

## For the More Technically Minded

As stated, this is really one of those situations where you are better off solving the problem manually with process and education rather than technology.

However, if you are looking for a quick fix, the easiest of all has got to be Apache Mod_Security. This tool has been developed to coexist with Apache Web servers. If you need to protect another system, you simply need to set it up as a reverse proxy in front of the Web server (as you would need to do

with the commercial products). It provides very robust and commercial-class protection.

These days I find installing RPFs on Linux systems a complete pain, so this install always cheers me up because it is source only. Once you've created your library, you need to make sure it is loaded by adding it to your `apache2.conf` file, like so

```
LoadModule security_module modules/mod_security.so
```

Now you have to add some boilerplate configuration to turn it on. It is very good practice to place rules in a separate file under `/etc/http` called `mod_security.conf` and then include them into `apache2.conf`. But this isn't a how-to guide; I want to get to the meat (rules) as soon as possible, so while we are playing, just slam in `apache2.conf` anywhere that works for you:

```
<IfModule mod_security.c>
# Turn it on
SecFilterEngine on
SecFilterDefaultAction "deny,log,status:666"

# Don't just check gets do posts as well
SecFilterScanPOST on
# See previous chapter on IDS evasion - u need to talk the talk
SecFilterCheckURLEncoding on
# UTF-8 encoding as above
SecFilterCheckUnicodeEncoding on
# Accept almost all byte values
SecFilterForceByteRange 1 255
# Defeat worms nobody knows you're a dog on the Internet
SecServerSignature "FATBOY"
SecAuditEngine RelevantOnly
# include or insert your rules
Include /etc/security/mod_sec_rules.conf
</IfModule>
```

By default, `mod_security` does a bit of the functionality of `Lockdown/Urlscan` on IIS Server, but it can do so much more. When you start adding rules into your rules file, the fun really starts.

To protect against the first XSS attack I described, add:

```
SecFilterSelective ARGS "<script"
# line above looks for a stream containing the script directive - if spots #
performs the defined action - block and alert in our case

SecFilterSelective ARGS "alert[[:space:]]*\("
# line 2 looks for the alert command
```

To protect against SQL injection, the following rules could be used:

```
# Line below catches a very simple probe for an sql-injection vulnerability
SecFilterSelective ARGS "or 1=1--'"
# The following line shows the real world directive with complex regex to
# remove leading and embedded spaces
SecFilterSelective ARGS "or.+1[[:space:]]*=[[:space:]]1"
```

Both of these commands still look for “*or 1=1*” in one form or other. The second accounts for white space and typing techniques. In real life, you’d need a much bigger set of rules, but that is available—just visit [www.modsecurity.org](http://www.modsecurity.org) and prepare to be amazed.

## Does It Work?

Only you can answer that question. Obviously, you should endeavor to fix all security problems at the source, but we also have to live in the real world. If `mod_security` or another product gets you out of a scrape, so be it.

A word to the wise; check that it really does work. I have implemented these things a number of times; sometimes they just make it worse. Here is the first acid test. At any name field, whether it be delivery name, customer name, or name and address, try entering *o’donald* or *Stjohn-smythe*. These are valid names, so if you get an error, the solution is broken.

My second acid test is a little harsh. It was defined when I ran security at a company that provided Web services for one of the largest software manufacturers in the world. There were many forums where users had to enter SQL commands like *select * from table*. This drove the application firewall mad. Check that you can exclude such fields from processing, and enjoy!

## Summary

This chapter reviewed the 10 OWASP application flaws. It provided examples of the most common and discussed ways that they could be mitigated.

In my experience, a significant number of Web portals, Web-enabled interfaces, or e-commerce applications will contain these flaws. Sometimes, like the more common XSS vulnerabilities, they will result only in embarrassing failures. The more malevolent command or SQL injection vulnerabilities can render all other security mechanisms useless—allowing the bad guys to run arbitrary commands on your servers. This is a recipe for disaster.

The ultimate goal must be to fix these pesky programs and train the programmers so that they can avoid creating the problem, but if you are after a quick fix, this chapter provided you with a working knowledge of application firewalling.



# Index

## A

AAA (authentication, authorization, accounting), 63–66

acceptable use policy (AUP), 34, 74

access

- browser content control, logging, 130
- remote. *See* remote access

accountability, security

- organization function, 28–29

accounting in AAA (authentication, authorization, accounting), 65–66

ACK scans, 263

Adaptive Security Algorithm (ASA), 159–160

adding users, 12

address spoofing, 68

administrative control described, 58

analysis

- risk, 58–63
- signature, 193–195

anomalous traffic detection and profiling, 195–198

antispoofing firewall rules, 155, 171–173

antivirus software

- See also* viruses

- and data availability, 52

application development, security awareness programs for, 39

application firewalls, 243–244, 299

application-level gateways, 148–149

application security

- application penetration tests, testing, 257, 270–274
- buffer overflows, cross-site scripting, 286–291
- command injection attacks, 294–295
- configuration management, 284–285
- introduction to, 282–284
- solutions for, 298–301
- SQL injection attacks, 291–294

architecture

- firewall, 144
- FireWall-1 (Check Point), 165
- of security team, 11

ARP spoofing, 269

assessments, security, 11–12

assets in vulnerability cycle, 54–56

asymmetrical routing, 138, 192–193, 212–213

attacks

- See also specific attack*
- application vulnerabilities, 284–285
- generic types of, 67–69

- and IDSes, 177–178
  - Audit Investigation and
    - Community Enterprise Act 2005 (U.K.), 80
  - auditing department
    - automated compliance, 40–41
    - and IT security team, 15
    - relationship to security team, 5–6
  - audits
    - HIDS, lack of, 180
    - information security, 104–110
  - authentication
    - in AAA (authentication, authorization, accounting), 63–64
    - and bad identity control, 295
    - strong, 30, 132
    - user, firewall rules, 155–156
  - authorization in AAA (authentication, authorization, accounting), 63–64
  - automated audit compliance, 40–41
  - availability in CIA risk analysis, 52
- B**
- backups, security organization function, 33
  - bad identity control, 295–297
  - banking audit, FIPS 140-2, 104–105
  - Beer, Stafford, 42
  - blaggers, 277
  - board of directors and IT security department, 15
  - bosses, 120–122
  - browsing
    - content control, 130
    - forceful, 296–297
  - brute-force attacks, 69, 266, 273
  - BS 7799 information security standard, 89–98
  - buffer overflows, 286–288
  - business consultancy, security, 10–11
  - business impact analysis processes, 27
  - business strategy and security planning process, 26–27
- C**
- cache testing, 273
  - California SB 1386, 73, 83
  - Caswell, Brian, 199
  - CEO/CTO/CFO, relationship to security team, 6–7
  - certification, Common Criteria (CC), 103–104
  - Check Point FireWall-1, 164–173

- Check Point SmartCentre, 167–168
  - checksums
    - and data integrity, 52
    - and HIDS, 179–180
  - chief information security officer.
    - See* CISO
  - CIA (confidentiality, integrity, availability) risk analysis, 51–54
  - circuit-level gateways, 147, 149
  - Cisco CatOS, 208
  - Cisco Guard, 249–250
  - Cisco PIX firewall, 158–164
  - Cisco Security auditor, 41
  - CISO (chief information security officer)
    - laws and regulations to know, 73–85
    - position in organization, function, 2–6
    - what makes a good, 17
  - command injection attacks, 294–295
  - commercial firewalls, 158–173
  - Common Criteria (CC)
    - certification, 103–104
  - Competitive Strategy (Porter), 23
  - compliance
    - and enforcement, 37–41
    - of policy statements, 32
    - security organization function, 28–29
  - computer auditors, 104–105
  - Computer Misuse Act 1990 (U.K.), 73–75
  - confidentiality in CIA risk analysis, 51
  - configurations
    - management of, 284–285
    - PIX firewall, 163–164
  - content, context attacks, 190
  - cookie examination, 273
  - corporate firewall, and
    - infrastructure security, 126–128
  - covert penetration testing, 233
  - crisis management and incident response, 227–231
  - cross-site scripting, 288–291
  - customer churn, 61–62
  - cut-through proxies, 161
- D**
- data asset valuation, 27
  - data availability principle, 52
  - data classification and confidentiality, 51
  - data disposal function, 33
  - data integrity in CIA risk analysis, 52
  - data protection function, 33
  - Data Protection Act 1998 (U.K.), 75–77, 291
  - deception technology, 245–246

decoders and reassembly routines, 187–188

defense in depth principle, 66, 134

denial-of-service (DoS) attacks

- described, 68
- IPS protection, 243–244, 247–249
- testing, 267–268, 273–274

Denning, Dorothy E., 276

deploying intrusion detection systems (IDSes), 213–227

designing

- e-commerce network, 133–139
- secure firewalls, 125–126

detective controls, 57

discretionary access control (DAC), 65

distributed denial-of-service (DDoS) attacks, prevention systems, 52

divisional security officers, 14

DMZ (demilitarized zone), remote access, 131–132

DNS (Domain Name Service)

- described, 150
- and network penetration testing, 259

documentation

- for incident response, 229
- security policy, 31
- security program framework, 43–48

DoS (denial-of-service) attacks

- described, 68

- IPS protection, 243–244, 247–249
- testing, 267–268, 273–274

dropped packets, 207

duress alarms, 64

dynamic NAT, 152–154

## E

e-commerce, designing

- configuration, 133–139

e-mail

- usage, 33
- virus protection, 128–130

encoding, avoidance techniques, 185

encryption

- and confidentiality, 51
- IPSec termination, 156
- security organization function, 34
- SSL and, 190–192

enforcement and compliance, 37–41

Enron, 83

equipment disposal and confidentiality, 51

exposure, in vulnerability cycle, 54–56

extended host OS protection, 246–247

**F**

failover  
   monitoring, 210–211  
   PIX support, 161–163

failure stance of failed devices, 67

Financial Services Authority  
   (FSA), 10, 105–106

FIPS 140-2 (Federal Information  
   Processing Standard 140-2),  
   102–103

FireWall-1 (Check Point),  
   164–173, 299

firewalls  
   access lists for, 12  
   application firewalls, 243–244,  
   299  
   corporate, 126–128  
   described, importance of,  
   144–147  
   desired features, 151–157  
   and IDSes, 223–224  
   securing, 264–265  
   security organization function,  
   30  
   stealth, virtualized, commercial,  
   157–173  
   structures and types, 147–150

FIRST  
   Common Vulnerability Scoring  
   System (CVSS), 62  
   CVEE assessment methodology,  
   40

FIT 1/94 audit, 108

flooding risk, 61

forceful browsing attacks, 296–297

Ford, W., 53

form poisoning, 274

FRAG 21 audit, 108

fragment reassembly, 183–184

framework of security program,  
   43–48

Freedom of Information Act 2000,  
   the (U.K.), 80

FTP and stateful inspection, 150

FTP-bounce scans, 264

FTP servers, securing, 131

FUZing programs, 287

**G**

Gartner Group and IDSes, 178

gateways, application-level, proxies,  
   148–149

Google search, 261

Government Communications  
   Headquarters (GCHQ), 50

Gramm-Leach-Bliley Act (U.S.),  
   84

**H**

hacking, and penetration testing,  
   276–279

- hacktivists, 277–278
  - Health Insurance Portability and Accountability Act (HIPAA), 85
  - help desk
    - security awareness programs for, 38–39
    - and security function, 16
  - HIDS (host-based IDS), problem with, 179–181
  - Hogwash IDS, 250–253
  - HoneyNet Project, 245
  - honeypots, 225, 245–246
  - host access controls, security
    - organization function, 33
  - host-based IDS (HIDS), 179–181
  - host detectors, 224
  - host enumeration, 262–263
  - Human Rights Act 1989, the (U.K.), 77–78
- I**
- ICMP protocol, 150
  - “idle” scans, 264
  - IDSes. *See* intrusion detection systems
  - in-line protection, IPSes, 241–242
  - incident
    - management function, 7–9
    - response and crisis management, 227–231
  - information classification function, 33
  - information management, IDS deployment, 225–227
  - information risk management team, 13
  - information security
    - standards and audits, 89–110
    - U.K. legislation, 73–82
    - U.S. information security legislation, 82–86
  - information security policy
    - characteristics and concerns of, 30–36
    - introduction to, 20–21
    - program outputs, 28
    - and standards, strategy, 21–27
  - infrastructure security
    - browser content control, logging, 130
    - corporate firewall, 126–128
    - e-mail protection, 128–130
    - network perimeter security, 124–126
  - input, unvalidated, 285–295
  - integrity in CIA risk analysis, 52
  - internal penetration testing, 270
  - Internet usage and “Netiquette,” 33
  - Internet Number Resource Organization (INRO), 261
  - interviews, how to handle, 110–120

- intrusion detection systems (IDSes)
    - deploying NIDS in stealth mode, 206–207
    - deployment methodology, 213–227
    - described, importance of, 177–181
    - NIDS and IPS vulnerabilities, 182–198
    - RealSecure IDS/IPS, 201–204
    - security organization function, 29
    - Snort IDS, 199–200
    - testing, 231–233
  - intrusion prevention systems (IPSeS)
    - active responses used in, 238–239
    - described, 178
    - detection flaws, 182–188
    - in-line protection, 241–242
    - poor configuration, 193–198
    - poor deployment of, 188–193
    - RealSecure IDS/IPS, 201–204
    - traditional with active response, 240–241
  - investigation of security incidents, 7–9
  - IP address spoofing, 155, 268
  - IP addresses and firewalls, 146
  - IPSec termination, 156
  - IPSeS. *See* intrusion prevention systems
  - ISO/IEC 27001:2005 information security standard, 98–102
  - ISP looking-glass servers, 261
  - IT department
    - relationship to security team, 3–5
    - security awareness programs for, 38–39
    - and security function, 15
    - system control types, 56–58
  - IT director, relationship to security team, 3–5
- J**
- job rotation as system control, 58
- K**
- Kerberos, and AAA system, 66
- L**
- legal advice, and IT security team, 15
  - legal considerations
    - penetration testing, 274–276
    - of security planning process, 26
    - of security team, 9–10

## legislation

- U.K. information security, 73–82
- U.S. information security, 82–86
- and security function, 9–10

## licensing, software, 34

## logging

- and browser content control, 130
- desired firewall feature, 155

**M**

## managing

- audits, 108–109
- configurations, 284–285
- interviews, 110–120

## mandatory access control (MAC), 65

## maximum transmission unit (MTU), 183–184

## mean time to failure (MTTF), 137

## measurability and security function, 29

## memory creep, 138

## message interception, 68

## message replay attacks, 69

## monitoring

- failovers, 210–211
- with HIDS, 179–180
- proactive configurations, 41

## multiple concurrent logons test, 271

**N**

## NAT/PAT (network address translation), 151–155

## Netflow records, 249

## network address translation (NAT), 151–155

## network alerts, 219–221

## network-based IDS (NIDS), 179

## network enumeration and discovery, 67–68, 262

## network penetration testing, test phases, 259–270

## network perimeter security, 124–126

## network sniffing, 268

## networks

- out-of-band (OOB), 135
- tap technology, 189, 209–210

## NIDS (network-based IDS)

- deploying in stealth mode, 206–207

## detection flaws, 182–188

## poor configuration, 193–198

## poor deployment of, 188–193

## nonrepudiation principle in CIA risk analysis, 53

**O**

## Official Secrets Act (U.K.), 80–82

## OOB (out-of-band) networks, 135



open source in-line IDS/IPS,  
250–253

Open System Interconnection  
(OSI) networking model,  
145–146

operations, standards and  
procedures, 24–25

organizations, hybrid information  
security, 13–16

OS fingerprinting, 262

OS protection, extended host, 246

OSI (Open System  
Interconnection) networking  
model, 145–146

out-of-band (OOB) networks, 135

## P

packets, dropped, 207

partners and security planning  
process, 27

passwords  
and authentication, 64  
changing, 12  
insecure storage of, 297  
one-time password generators,  
132

patch management, security  
awareness programs for, 40

penetration testing  
application, 270–274  
controls, paperwork, 274–276

described, types of, 257–259

process of, 260

periodic network vulnerability  
assessment, 257–258

persistence of information tests,  
273

physical security, 259

plans, strategic, 23

policies  
information security. *See*  
information security policy  
responsibility for developing  
security, 10

port scanners, scanning, 185, 224,  
263–264

Porter, Michael, 23

positioning sensors, 218–219

prevention systems and intrusion  
prevention systems (IPSeS),  
178

principle of least privilege, 66

profiling anomalous traffic,  
195–198

programs, security awareness,  
38–39

protective controls, 57

protocols  
defining your own, 156–157  
IDS/IPS analysis, 185–186  
Internet, 145–146

proxies (application-level  
gateways), 148–149

proxy-based firewalls, 147

**Q**

qualitative and quantitative risk analyses, 59–60

**R**

RADIUS system, 66, 161

RealSecure IDS/IPS, using, 201–204

recovery controls, 58

Regulation of Investigatory Powers Act 2000, the (U.K.), 78–79

regulatory considerations  
of security planning process, 26  
of security team, 9–10

remote access

design options, 132–133

DMZ (demilitarized zone), 131–133

security organization function, 34

reporting

lines in security organization, 17

penetration testing, 269–270

security organization function, 28–29

research, responsibility of security team, 11

Resource Access Control Facility (RACF), 3–5

risk

analysis, 58–63

and vulnerability cycle, 54–56

Riverhead/Cisco Guard IPS, 247–249

Roesch, Marty, 199

role-based access control (RBACs), 65

routers

screening, 148

securing, 264–265

routing, asymmetrical, 138, 192–193, 212–213

RPC scans, 264

RST scans, 263

rules

firewall, 146

screening routers, 148

Snort's, 200

**S**

SANS (SynAdmin, Audit, Network, Security) Institute's policy template, 34

Sarbanes-Oxley 2002 (U.S.), 9, 73, 83–84

SAS 70 information security audits, 106–107

scans. *See specific scans*

scenario testing, 266–269

screening routers, 148

script kiddies, 278–279

- scripting, cross-site, 288–291
- Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption* (Wood), 53
- Securities and Exchange Commission (SEC), 10
- security
  - application, 282–284
  - IDSes. *See* intrusion detection systems
  - infrastructure. *See* infrastructure security
  - IPSes. *See* intrusion prevention systems
  - jargon, principles, concepts, 49–69
  - program framework, 43–48
  - responsibility for developing policies, standards, 10
  - tools required for, 29–30
- security appliances, 129–130
- security awareness programs, 38–39
- security dashboards, 29
- security management systems, 181
- security officers, laws and regulations to know, 73–85
- security organization
  - management system of, 28–29
  - roles and duties, 7–16
- security policy
  - characteristics of, 30–31
  - laws and regulations affecting, 73–85
- security programs, methodology and framework, 43–48
- selecting IDSes, 215–216
- sensors, positioning, 218–219
- servers, access lists for, 12
- session hijacking, 68
- session stealing, 271–272
- signature analysis, 193–195, 219–220
- SNMP hacking, 265
- Snort IDS, using, 176, 199–200
- social engineering, 69, 276
- software licensing function, 34
- spanning ports, 189, 207–209
- spoofing, penetration testing, 268–269
- SQL injection attacks, 291–294
- SSL and encryption, 190–192
- stack overflow protection, 246
- standards
  - information security, 89–104
  - security policy and strategy, 21–25, 36–37
- state monitoring, 179
- stateful inspection firewalls, 149–150
- stateful rule base, firewall, 151
- Statement on Auditing Standards (SAS) No. 70, 106–107
- statements, policy, 32–33
- static NAT (network address translation), 152

stealth firewalls, 157–158  
 strategy and security policy, 21–27  
 strong authentication, 30, 132  
 SunScreen EFS, 157  
 suppliers and security planning  
   process, 27  
 support, technical, and security  
   function, 16  
 switches in NIDS, 189  
 SYN flood attacks, 171, 267  
 SYN scans, 263  
 system control types, 56–58  
 system development and security  
   function, 16  
 SysTrust audit, 108

## T

TACACS+ system, 65, 161  
 tactics and security policy, 23–24  
 tap technology, 189, 209–210  
 TCP (Transmission Control  
   Protocol) scans, 263–264  
 TCP/IP (Transmission Control  
   Protocol / Internet Protocol),  
   145  
 technical support  
   security awareness programs for,  
   39  
   and security function, 16  
 Telecommunications (Lawful  
   Business Practice)

(Interception of  
 Communications)  
 Regulations 2000, the (U.K.),  
 79

Telnet, 155–156

testing

  covert penetration, 233  
   denial-of-service (DoS), 267–268  
   IDSes, 231–233  
   penetration test types, 257–259  
   vulnerabilities, 265

threat analysis

  corporate firewall, 127–128  
   DMZ (demilitarized zone),  
   131–132  
   network configured for e-  
   commerce, 136

threats in vulnerability cycle,  
 54–56

timeout testing, 273

tools, security, 29–30

TopLayer DDOS prevention  
 appliance, 247

traffic

  aggregating different flows,  
   211–212

  monitoring with IDSes. *See*  
   intrusion detection systems

Transmission Control Protocol /  
 Internet Protocol (TCP/IP),  
 145

transparent bridging mode, 158

Trojan attacks, 269

**U**

- UDP bombs, 268
- UDP datagrams, 188
- UDP scans, 264
- U.K. information security
  - legislation, 73–82
- unvalidated input
  - buffer overflows, 286–288
  - introduction to, 285–286
- updates, patch management, 40
- URL filtering, 30
- URL parameter tampering, 297
- U.S. information security
  - legislation, 82–86
- USA Patriot Act 2001, 85
- Usenet newsgroups, 261
- user-authenticated traffic, 155–156

**V**

- virtual packet assembly, 183–184
- virtual private networks (VPNs),
  - security organization
    - function, 30
- virtualized firewalls, 158
- viruses
  - e-mail protection, 128–130
  - and policy controls, 33
  - protection from, 29
- VISA CISP audit, 108

- VLANs (virtual local area networks), 158
- vulnerabilities and application security, 283–284
- vulnerability cycle, risk and, 54–56
- vulnerability scans, scanning, 30, 41, 265

**W**

- Web servers
  - and application security, 284–285
  - securing, 131
- white hat hackers, 278
- Wood, Charles Cresson, 35
- workstation authentication, 156
- Writing Information Security Policies* (Wood), 35

**X**

- x509 certificates, 149
- XSS cross-site scripting, 288, 290

**Z**

- zombie-bounce scans, 264













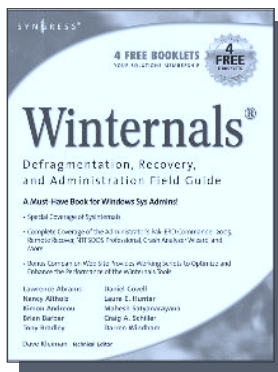






# Syngress: *The Definition of a Serious Security Library*

**Syn-gress** (sin-gres): *noun, sing.* Freedom from risk or danger; safety. See *security*.



AVAILABLE NOW  
order @  
[www.syngress.com](http://www.syngress.com)

## Winternals Defragmentation, Recovery, and Administration Field Guide

Dave Kleiman, Laura E. Hunter, Tony Bradley, Brian Barber,

Nancy Altholz, Lawrence Abrams, Mahesh Satyanarayana, Darren Windham, Craig Schiller

As a system administrator for a Microsoft network, you know doubt spend too much of your life backing up data and restoring data, hunting down and removing malware and spyware, defragmenting disks, and improving the overall performance and reliability of your network. The Winternals® Defragmentation, Recovery, and Administration Field Guide and companion Web site provide you with all the information necessary to take full advantage of Winternals comprehensive and reliable tools suite for system administrators.

ISBN: 1-59749-079-2

Price: \$49.95 US \$64.95 CAN

## Syngress IT Security Project Management Handbook

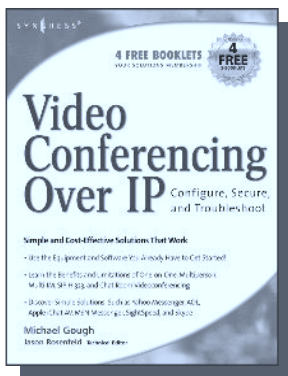
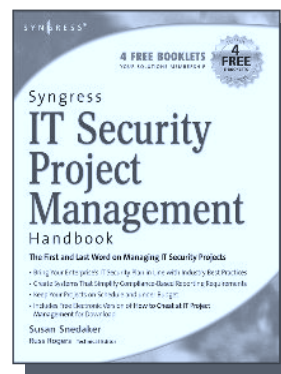
Susan Snelaker

The definitive work for IT professionals responsible for the management of the design, configuration, deployment and maintenance of enterprise-wide security projects. Provides specialized coverage of key project areas including Penetration Testing, Intrusion Detection and Prevention Systems, and Access Control Systems.

ISBN: 1-59749-076-8

Price: \$59.95 US \$77.95 CAN

AVAILABLE NOW  
order @  
[www.syngress.com](http://www.syngress.com)



AVAILABLE NOW  
order @  
[www.syngress.com](http://www.syngress.com)

## Video Conferencing over IP: Configure, Secure, and Troubleshoot

Michael Gough

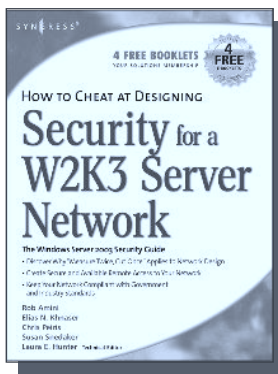
Until recently, the reality of videoconferencing didn't live up to the marketing hype. That's all changed. The network infrastructure and broadband capacity is now in place to deliver clear, real-time video and voice feeds between multiple points of contacts, with market leaders such as Cisco and Microsoft continuing to invest heavily in development. In addition, newcomers Skype and Google are poised to launch services and products targeting this market. *Video Conferencing over IP* is the perfect guide to getting up and running with video teleconferencing for small to medium-sized enterprises.

ISBN: 1-59749-063-6

Price: \$49.95 U.S. \$64.95 CAN

# Syngress: *The Definition of a Serious Security Library*

**Syn-gress** (sin-gres): *noun, sing.* Freedom from risk or danger; safety. See *security*.



AVAILABLE NOW  
order @  
[www.syngress.com](http://www.syngress.com)

## How to Cheat at Designing Security for a Windows Server 2003 Network

Neil Ruston, Chris Peiris

While considering the security needs of your organization, you need to balance the human and the technical in order to create the best security design for your organization. Securing a Windows Server 2003 enterprise network is hardly a small undertaking, but it becomes quite manageable if you approach it in an organized and systematic way. This includes configuring software, services, and protocols to meet an organization's security needs.

ISBN: 1-59749-243-4

Price: \$39.95 US \$55.95 CAN

## How to Cheat at Designing a Windows Server 2003 Active Directory Infrastructure

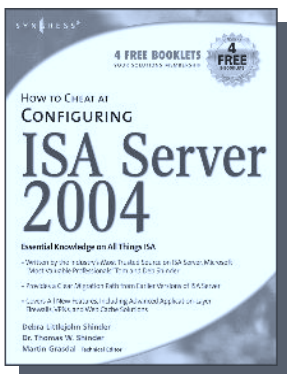
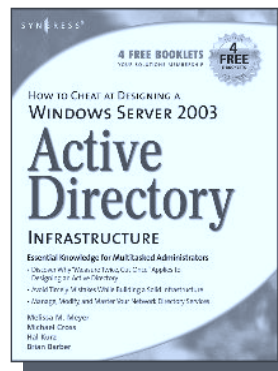
Melissa Craft, Michael Cross, Hal Kurz, Brian Barber

The book will start off by teaching readers to create the conceptual design of their Active Directory infrastructure by gathering and analyzing business and technical requirements. Next, readers will create the logical design for an Active Directory infrastructure. Here the book starts to drill deeper and focus on aspects such as group policy design. Finally, readers will learn to create the physical design for an active directory and network Infrastructure including DNS server placement; DC and GC placements and Flexible Single Master Operations (FSMO) role placement.

ISBN: 1-59749-058-X

Price: \$39.95 US \$55.95 CAN

AVAILABLE NOW  
order @  
[www.syngress.com](http://www.syngress.com)



AVAILABLE NOW  
order @  
[www.syngress.com](http://www.syngress.com)

## How to Cheat at Configuring ISA Server 2004

Dr. Thomas W. Shinder, Debra Littlejohn Shinder

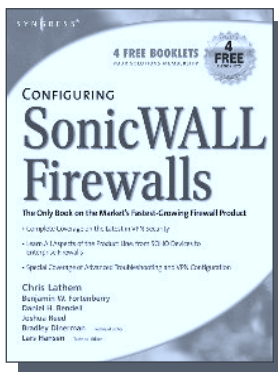
If deploying and managing ISA Server 2004 is just one of a hundred responsibilities you have as a System Administrator, "How to Cheat at Configuring ISA Server 2004" is the perfect book for you. Written by Microsoft MVP Dr. Tom Shinder, this is a concise, accurate, enterprise tested method for the successful deployment of ISA Server.

ISBN: 1-59749-057-1

Price: \$34.95 U.S. \$55.95 CAN

# Syngress: *The Definition of a Serious Security Library*

**Syn-gress** (sin-gres): *noun, sing.* Freedom from risk or danger; safety. See *security*.



AVAILABLE NOW  
order @  
[www.syngress.com](http://www.syngress.com)

## Configuring SonicWALL Firewalls

Chris Latham, Ben Fortenberry, Lars Hansen

Configuring SonicWALL Firewalls is the first book to deliver an in-depth look at the SonicWALL firewall product line. It covers all of the aspects of the SonicWALL product line from the SOHO devices to the Enterprise SonicWALL firewalls. Advanced troubleshooting techniques and the SonicWALL Security Manager are also covered.

ISBN: 1-59749-250-7

Price: \$49.95 US \$69.95 CAN

## Perfect Passwords: Selection, Protection, Authentication

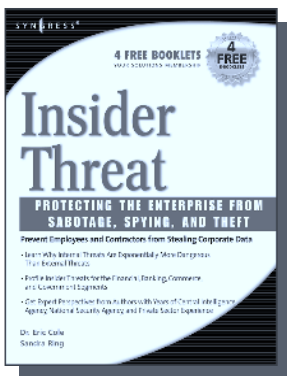
Mark Burnett

User passwords are the keys to the network kingdom, yet most users choose overly simplistic passwords (like password) that anyone could guess, while system administrators demand impossible to remember passwords littered with obscure characters and random numerals. Author Mark Burnett has accumulated and analyzed over 1,000,000 user passwords, and in this highly entertaining and informative book filled with dozens of illustrations reveals his findings and balances the rigid needs of security professionals against the ease of use desired by users.

ISBN: 1-59749-041-5

Price: \$24.95 US \$34.95 CAN

AVAILABLE NOW  
order @  
[www.syngress.com](http://www.syngress.com)



AVAILABLE NOW  
order @  
[www.syngress.com](http://www.syngress.com)

## Insider Threat: Protecting the Enterprise from Sabotage, Spying, and Theft

Dr. Eric Cole and Sandra Ring

As network defense perimeters get stronger and stronger; IT, security, law enforcement, and intelligence professionals are realizing that the greatest threats to their networks are increasingly coming from within their own organizations. These insiders, comprised of current and former employees or contractors, can use their inside knowledge of a target network to carry out acts of sabotage, espionage, and theft of data.

ISBN: 1-59749-048-2

Price: \$34.95 U.S. \$48.95 CAN