
CONTENTS

Preface	xix
Acknowledgments	xxi
Introduction	xxiii
Part I Introduction to Ethical Disclosure	I
Chapter 1 Ethics of Ethical Hacking	3
How Does This Stuff Relate to an Ethical Hacking Book?	10
The Controversy of Hacking Books and Classes	11
The Dual Nature of Tools	12
Recognizing Trouble When It Happens	13
Emulating the Attack	14
Security Does Not Like Complexity	15
Chapter 2 Ethical Hacking and the Legal System	17
Addressing Individual Laws	19
18 USC Section 1029: The Access Device Statute	19
18 USC Section 1030 of The Computer Fraud and Abuse Act	23
State Law Alternatives	30
18 USC Sections 2510, et. Seq. and 2701	32
Digital Millennium Copyright Act (DMCA)	36
Cyber Security Enhancement Act of 2002	39
Chapter 3 Proper and Ethical Disclosure	41
You Were Vulnerable for How Long?	45
Different Teams and Points of View	47
How Did We Get Here?	49
CERT's Current Process	50
Full Disclosure Policy (RainForest Puppy Policy)	52
Organization for Internet Safety (OIS)	54
Discovery	55
Notification	55
Validation	57
Resolution	60
Release	62
Conflicts Will Still Exist	62

Case Studies	62
Pros and Cons of Proper Disclosure Processes	63
iDefense	67
Zero Day Initiative	68
Vendors Paying More Attention	69
So What Should We Do from Here on Out?	70
Part II Penetration Testing and Tools	73
Chapter 4 Using Metasploit	75
Metasploit: The Big Picture	75
Getting Metasploit	75
Using the Metasploit Console to Launch Exploits	76
Exploiting Client-Side Vulnerabilities with Metasploit	83
Using the Meterpreter	87
Using Metasploit as a Man-in-the-Middle Password Stealer	91
Weakness in the NTLM Protocol	92
Configuring Metasploit as a Malicious SMB Server	92
Brute-Force Password Retrieval with the LM Hashes + Challenge	94
Building Your Own Rainbow Tables	96
Downloading Rainbow Tables	97
Purchasing Rainbow Tables	97
Cracking Hashes with Rainbow Tables	97
Using Metasploit to Auto-Attack	98
Inside Metasploit Modules	98
Chapter 5 Using the BackTrack LiveCD Linux Distribution	101
BackTrack: The Big Picture	101
Creating the BackTrack CD	102
Bootting BackTrack	103
Exploring the BackTrack X-Windows Environment	104
Writing BackTrack to Your USB Memory Stick	105
Saving Your BackTrack Configurations	105
Creating a Directory-Based or File-Based Module with dir2lzm	106
Creating a Module from a SLAX Prebuilt Module with mo2lzm	106
Creating a Module from an Entire Session of Changes Using dir2lzm	108
Automating the Change Preservation from One Session to the Next	109

Creating a New Base Module with All the Desired Directory Contents	110
Cheat Codes and Selectively Loading Modules	112
Metasploit db_autopwn	114
Tools	118
Part III Exploits 101	119
Chapter 6 Programming Survival Skills	121
C Programming Language	121
Basic C Language Constructs	122
Sample Program	126
Compiling with gcc	127
Computer Memory	128
Random Access Memory (RAM)	128
Endian	128
Segmentation of Memory	129
Programs in Memory	129
Buffers	130
Strings in Memory	130
Pointers	130
Putting the Pieces of Memory Together	131
Intel Processors	132
Registers	132
Assembly Language Basics	133
Machine vs. Assembly vs. C	133
AT&T vs. NASM	133
Addressing Modes	135
Assembly File Structure	136
Assembling	137
Debugging with gdb	137
gdb Basics	137
Disassembly with gdb	139
Python Survival Skills	139
Getting Python	140
Hello World in Python	140
Python Objects	140
Strings	141
Numbers	142
Lists	143
Dictionaries	144
Files with Python	144
Sockets with Python	146

Chapter 7 Basic Linux Exploits	147
Stack Operations	148
Function Calling Procedure	148
Buffer Overflows	149
Overflow of meet.c	150
Ramifications of Buffer Overflows	153
Local Buffer Overflow Exploits	154
Components of the Exploit	155
Exploiting Stack Overflows by Command Line	157
Exploiting Stack Overflows with Generic Exploit Code	158
Exploiting Small Buffers	160
Exploit Development Process	162
Real-World Example	163
Determine the Offset(s)	163
Determine the Attack Vector	166
Build the Exploit Sandwich	167
Test the Exploit	168
Chapter 8 Advanced Linux Exploits	169
Format String Exploits	169
The Problem	170
Reading from Arbitrary Memory	173
Writing to Arbitrary Memory	175
Taking .dtors to root	177
Heap Overflow Exploits	180
Example Heap Overflow	181
Implications	182
Memory Protection Schemes	182
Compiler Improvements	183
Kernel Patches and Scripts	183
Return to libc Exploits	185
Bottom Line	192
Chapter 9 Shellcode Strategies	195
User Space Shellcode	196
System Calls	196
Basic Shellcode	197
Port Binding Shellcode	197
Reverse Shellcode	199
Find Socket Shellcode	200
Command Execution Code	201
File Transfer Code	202
Multistage Shellcode	202
System Call Proxy Shellcode	202
Process Injection Shellcode	203

Other Shellcode Considerations	204
Shellcode Encoding	204
Self-Corrupting Shellcode	205
Disassembling Shellcode	206
Kernel Space Shellcode	208
Kernel Space Considerations	208
Chapter 10 Writing Linux Shellcode	211
Basic Linux Shellcode	211
System Calls	212
Exit System Call	214
setreuid System Call	216
Shell-Spawning Shellcode with execve	217
Implementing Port-Binding Shellcode	220
Linux Socket Programming	220
Assembly Program to Establish a Socket	223
Test the Shellcode	226
Implementing Reverse Connecting Shellcode	228
Reverse Connecting C Program	228
Reverse Connecting Assembly Program	230
Encoding Shellcode	232
Simple XOR Encoding	232
Structure of Encoded Shellcode	232
JMP/CALL XOR Decoder Example	233
FNSTENV XOR Example	234
Putting It All Together	236
Automating Shellcode Generation with Metasploit	238
Generating Shellcode with Metasploit	238
Encoding Shellcode with Metasploit	240
Chapter 11 Basic Windows Exploits	243
Compiling and Debugging Windows Programs	243
Compiling on Windows	243
Debugging on Windows with Windows Console Debuggers	245
Debugging on Windows with OllyDbg	254
Windows Exploits	258
Building a Basic Windows Exploit	258
Real-World Windows Exploit Example	266
Part IV Vulnerability Analysis	275
Chapter 12 Passive Analysis	277
Ethical Reverse Engineering	277
Why Reverse Engineering?	278
Reverse Engineering Considerations	279

Source Code Analysis	279
Source Code Auditing Tools	280
The Utility of Source Code Auditing Tools	282
Manual Source Code Auditing	283
Binary Analysis	289
Manual Auditing of Binary Code	289
Automated Binary Analysis Tools	304
Chapter 13 Advanced Static Analysis with IDA Pro	309
Static Analysis Challenges	309
Stripped Binaries	310
Statically Linked Programs and FLAIR	312
Data Structure Analysis	318
Quirks of Compiled C++ Code	323
Extending IDA	325
Scripting with IDC	326
IDA Pro Plug-In Modules and the IDA SDK	329
IDA Pro Loaders and Processor Modules	332
Chapter 14 Advanced Reverse Engineering	335
Why Try to Break Software?	336
The Software Development Process	336
Instrumentation Tools	337
Debuggers	338
Code Coverage Tools	340
Profiling Tools	341
Flow Analysis Tools	342
Memory Monitoring Tools	343
Fuzzing	348
Instrumented Fuzzing Tools and Techniques	349
A Simple URL Fuzzer	349
Fuzzing Unknown Protocols	352
SPIKE	353
SPIKE Proxy	357
Sharefuzz	357
Chapter 15 Client-Side Browser Exploits	359
Why Client-Side Vulnerabilities Are Interesting	359
Client-Side Vulnerabilities Bypass Firewall Protections	359
Client-Side Applications Are Often Running	
with Administrative Privileges	360
Client-Side Vulnerabilities Can Easily Target Specific People	
or Organizations	360

Internet Explorer Security Concepts	361
ActiveX Controls	361
Internet Explorer Security Zones	362
History of Client-Side Exploits and Latest Trends	363
Client-Side Vulnerabilities Rise to Prominence	363
Notable Vulnerabilities in the History of Client-Side Attacks	364
Finding New Browser-Based Vulnerabilities	369
MangleMe	370
AxEnum	372
AxFuzz	377
AxMan	378
Heap Spray to Exploit	383
InternetExploiter	384
Protecting Yourself from Client-Side Exploits	385
Keep Up-to-Date on Security Patches	385
Stay Informed	385
Run Internet-Facing Applications with Reduced Privileges	385
Chapter 16 Exploiting Windows Access Control Model for	
Local Elevation of Privilege	387
Why Access Control Is Interesting to a Hacker	387
Most People Don't Understand Access Control	387
Vulnerabilities You Find Are Easy to Exploit	388
You'll Find Tons of Security Vulnerabilities	388
How Windows Access Control Works	388
Security Identifier (SID)	389
Access Token	390
Security Descriptor (SD)	394
The Access Check	397
Tools for Analyzing Access Control Configurations	400
Dumping the Process Token	401
Dumping the Security Descriptor	403
Special SIDs, Special Access, and "Access Denied"	406
Special SIDs	406
Special Access	408
Investigating "Access Denied"	409
Analyzing Access Control for Elevation of Privilege	417
Attack Patterns for Each Interesting Object Type	418
Attacking Services	418
Attacking Weak DACLs in the Windows Registry	424
Attacking Weak Directory DACLs	428
Attacking Weak File DACLs	433

What Other Object Types Are out There?	437
Enumerating Shared Memory Sections	437
Enumerating Processes	439
Enumerating Other Named Kernel Objects (Semaphores, Mutexes, Events, Devices)	439
Chapter 17 Intelligent Fuzzing with Sulley	441
Protocol Analysis	441
Sulley Fuzzing Framework	443
Installing Sulley	443
Powerful Fuzzer	443
Blocks	446
Sessions	449
Monitoring the Process for Faults	450
Monitoring the Network Traffic	451
Controlling VMware	452
Putting It All Together	452
Postmortem Analysis of Crashes	454
Analysis of Network Traffic	456
Way Ahead	456
Chapter 18 From Vulnerability to Exploit	459
Exploitability	460
Debugging for Exploitation	460
Understanding the Problem	466
Preconditions and Postconditions	466
Repeatability	467
Payload Construction Considerations	475
Payload Protocol Elements	476
Buffer Orientation Problems	476
Self-Destructive Shellcode	477
Documenting the Problem	478
Background Information	478
Circumstances	478
Research Results	479
Chapter 19 Closing the Holes: Mitigation	481
Mitigation Alternatives	481
Port Knocking	482
Migration	482
Patching	484
Source Code Patching Considerations	484
Binary Patching Considerations	486
Binary Mutation	490
Third-Party Patching Initiatives	495

Part V	Malware Analysis	497
Chapter 20	Collecting Malware and Initial Analysis	499
	Malware	499
	Types of Malware	499
	Malware Defensive Techniques	500
	Latest Trends in Honeynet Technology	501
	Honeypots	501
	Honeynets	501
	Why Honeypots Are Used	502
	Limitations	502
	Low-Interaction Honeypots	503
	High-Interaction Honeypots	503
	Types of Honeynets	504
	Thwarting VMware Detection Technologies	506
	Catching Malware: Setting the Trap	508
	VMware Host Setup	508
	VMware Guest Setup	508
	Using Nepenthes to Catch a Fly	508
	Initial Analysis of Malware	510
	Static Analysis	510
	Live Analysis	512
	Norman Sandbox Technology	518
	What Have We Discovered?	520
Chapter 21	Hacking Malware	521
	Trends in Malware	521
	Embedded Components	522
	Use of Encryption	522
	User Space Hiding Techniques	522
	Use of Rootkit Technology	523
	Persistence Measures	523
	Peeling Back the Onion—De-obfuscation	524
	Packer Basics	524
	Unpacking Binaries	525
	Reverse Engineering Malware	533
	Malware Setup Phase	533
	Malware Operation Phase	534
	Automated Malware Analysis	535
	Index	537