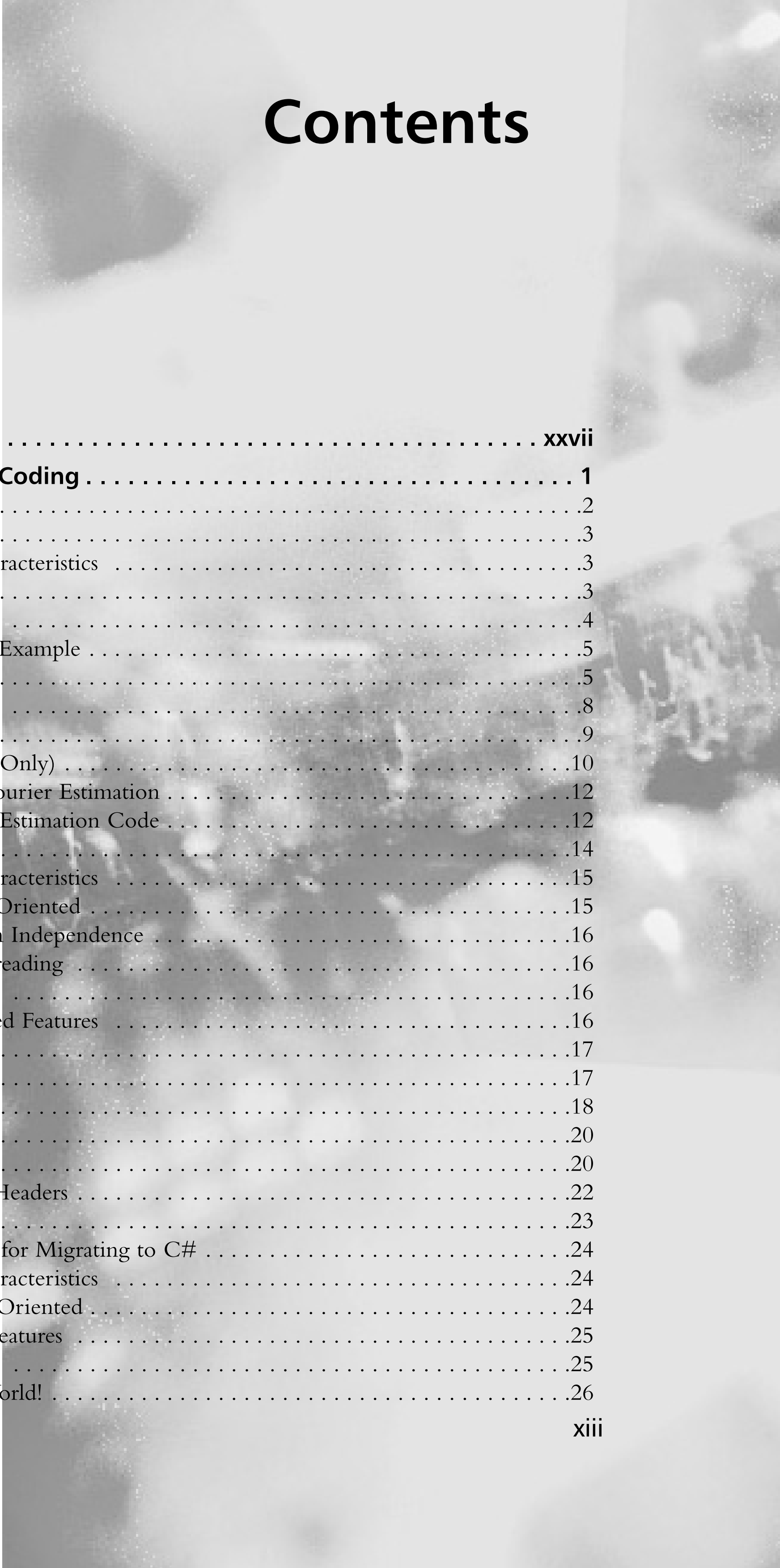

Contents

Foreword xxvii

Chapter 1 Security Coding 1

Introduction 2

C/C++ 3

Language Characteristics 3

C 3

Security 4

Hello, World! Example 5

Data Types 5

Flow Control 8

Functions 9

Classes (C++ Only) 10

Case Study: Fourier Estimation 12

Fourier Estimation Code 12

Java 14

Language Characteristics 15

Object Oriented 15

Platform Independence 16

Multithreading 16

Security 16

Advanced Features 16

Hello, World! 17

Data Types 17

Flow Control 18

Methods 20

Classes 20

GET HTTP Headers 22

C# 23

Business Case for Migrating to C# 24

Language Characteristics 24

Object-Oriented 24

Other Features 25

Security 25

C#'s Hello, World! 26

Data Types .26
Flow Control .27
Methods .29
Classes .30
C# Threading .31
Case Study: Command Line IP Address Parsing .32
Perl .40
Data Types .41
Operators .42
A Sample Perl Script .45
Analysis .46
Special Variables .46
Pattern Matching and Substitution .47
Regular Expression Modifiers .48
Canonical Perl Tools .49
I Am a Perl Coder! .49
Analysis .50
A Log Modification Utility .50
Execution .53
Analysis .53
Python .55
InlineEgg .56
Analysis .57
Analysis .58
Summary .60
Solutions Fast Track .61
Links to Sites .62
Frequently Asked Questions .63

Chapter 2 NASL Scripting 65

Introduction .66
History .66
Goals of NASL .66
Simplicity and Convenience .67
Modularity and Efficiency .67
Safety .67
NASL's Limitations .67
NASL Script Syntax .68
Comments .68
Variables .68
Operators .70
Control Structures .74
Writing NASL Scripts .77
Writing Personal-use Tools in NASL .78
Networking Functions .78
HTTP Functions .78
Packet Manipulation Functions .78

String Manipulation Functions .79
Cryptographic Functions .79
The NASL Command Line Interpreter .79
Programming in the Nessus Framework .80
Descriptive Functions .80
Case Study:The Canonical NASL Script .82
Porting to and from NASL .86
Logic Analysis .86
Identify Logic .86
Pseudo Code .87
Porting to NASL .88
Porting to NASL from C/C++ .89
Porting from NASL .94
Summary .95
Solutions FastTrack .95
Links to Sites .97
Frequently Asked Questions .97
Chapter 3 BSD Sockets . 99
Introduction .100
Introduction to BSD Sockets Programming .100
TCP Clients and Servers .101
Compilation .102
Example Execution .102
Analysis .102
Compilation .105
Analysis .105
Analysis .106
UDP Clients and Server .107
Compilation .109
Example Execution .109
Analysis .109
Compilation .111
Example Execution .111
Analysis .111
Compilation .113
Example Execution .113
Analysis .113
Compilation .115
Example Execution .115
Analysis .115
Socket Options .116
Analysis .118
Network Scanning with UDP Sockets .118
Compilation .125
Example Execution .125

Analysis .125
Network Scanning with TCP Sockets .127
Compilation .136
Example Execution .136
Analysis .136
Threading and Parallelism .139
Summary .141
Solutions Fast Track .141
Links to Sites .143
Frequently Asked Questions .143
Chapter 4 Windows Sockets (Winsock) 145
Introduction .146
Winsock Overview .146
Winsock 2.0 .148
Linking through Visual Studio 6.0 .148
Linking through Source Code .148
Analysis .150
Case Study: Using WinSock to Grab a Web Page .153
Analysis .154
Writing Client Applications .154
Analysis .156
Writing Server Applications .158
Analysis .160
Writing Exploit and Vulnerability Checking Programs .161
Analysis .167
Analysis .168
Summary .169
Solutions Fast Track .170
Frequently Asked Questions .170
Case Study: Using WinSock to Execute a Web Attack .172
Analysis .173
Case Study: Using Winsock to Execute a Remote Buffer Overflow .174
Analysis .176
Chapter 5 Java Sockets. 177
Introduction .178
An Overview of TCP/IP .178
TCP Clients .179
Compilation .181
Example Execution .181
Analysis .182
IP Addresses and Hostname Resolution .183

Example Execution .184
Analysis .184
Example Execution .185
Analysis .185
Text-Based Input/Output: The LineNumberReader Class .186
Compilation .188
Example Execution .188
Analysis .189
TCP Servers .189
Compilation .192
Example Execution .192
Analysis .192
Using a Web Browser to Connect to *TCPServer1* .193
Handling Multiple Connections .194
Compilation .200
Example Execution .200
Analysis .201
WormCatcher .204
Compilation .207
Example Execution .207
Analysis .208
UDP Clients and Servers .209
Compilation .213
Example Execution .214
Analysis .214
Summary .217
Solutions Fast Track .217
Frequently Asked Questions .218
Chapter 6 Writing Portable Code 221
Introduction .222
UNIX and Microsoft Windows Porting Guide .222
Pre-compiler Directives .222
Using ifdefs .223
Determining the Operating System .225
Example Execution .226
Analysis .226
Byte Ordering .226
Example Execution .227
Analysis .228
Process Creation and Termination .229
exec .229
Example Execution .229
Analysis .230
Example Execution .230
Analysis .230
Example Execution .233

Analysis .233
fork .234
exit .234
Multithreading .234
Thread Creation .234
Example Execution .235
Analysis .235
Example Execution .237
Analysis .237
Thread Coordination .237
Example Execution .239
Analysis .239
Example Execution .241
Analysis .241
Signals .242
Analysis .243
Analysis .244
File Handling .244
Analysis .245
Analysis .246
Directory Handling .247
Analysis .248
Analysis .249
Analysis .250
Libraries .250
Dynamic Loading of Libraries .252
Analysis .254
Analysis .255
Daemon/Win32 Service Programming .256
Example Execution .257
Analysis .258
Analysis .261
Memory Management .262
Analysis .263
Command-line Argument Processing .263
Analysis .264
Analysis .266
Example Execution .267
Analysis .268
Integer Data Types .267
Analysis .267
Summary .269
Solutions Fast Track .269
Frequently Asked Questions .269

Chapter 7 Portable Network Programming 273
Introduction 274
BSD Sockets and Winsock 274
Winsock Requirements 274
Analysis 276
Portable Components 276
Return Values 276
Analysis 277
Analysis 277
Analysis 278
Extended Error Information 278
Analysis 280
The API 280
Winsock 2.0 Extensions 280
read(), write() 280
socket() 280
Analysis 282
connect() 282
Analysis 285
bind() 285
Analysis 287
listen() 287
Analysis 290
accept() 290
Analysis 293
select() 293
Analysis 297
send(), sendto() 298
Analysis 301
recv(), recvfrom() 301
Analysis 304
Close(), Closesocket() 305
Analysis 306
setsockopt() 307
Analysis 309
Ioctl(), Ioctlsocket() 309
Analysis 311
Raw Sockets 312
API Overview 312
Header Files 312
IP(v4) Header File: 313
ICMP Header File: 315
UDP Header File: 315
TCP Header File (tcp.h): 316
Local IP Address Determination 317

User Supplied 317
Listing Interfaces 318
Example Execution 321
Analysis 322
Pcap and WinPcap 323
Example Execution 327
Analysis 328
Summary 329
Solutions Fast Track 329
Frequently Asked Questions 330

Chapter 8 Writing Shellcode I 333

Introduction 334
Overview of Shellcode 334
The Tools 335
The Assembly Programming Language .335
Windows vs UNIX Assembly 339
The Addressing Problem 339
Using the call and jmp Trick 339
Pushing the Arguments 340
The NULL Byte Problem 341
Implementing System Calls 342
System Call Numbers 342
System Call Arguments 343
System Call Return Values 344
Remote Shellcode 345
Port Binding Shellcode 345
Socket Descriptor Reuse Shellcode 346
Local Shellcode 348
execve Shellcode 348
setuid Shellcode 349
chroot Shellcode 350
Windows Shellcode 354
Summary 359
Solutions Fast Track 360
Links to Sites 362
Mailing Lists 362
Frequently Asked Questions 363

Chapter 9 Writing Shellcode II 365

Introduction 366
Shellcode Examples 366
The Write System Call 368
Analysis 369
Analysis 371
execve Shellcode 372
Analysis 373

Analysis 373
Analysis 375
Analysis 376
Analysis 378
Analysis 379
Execution 380
Port Binding Shellcode 380
Analysis 381
The socket System Call 383
Analysis 383
The bind System Call 383
The listen System Call 384
Analysis 384
The accept System Call 385
Analysis 385
The dup2 System Calls 385
Analysis 385
The execve System Call 386
Analysis 386
Analysis 389
Reverse Connection Shellcode 391
Analysis 393
Socket Reusing Shellcode 394
Analysis 395
Analysis 395
Reusing File Descriptors 396
Analysis 396
Analysis 398
Analysis 399
Analysis 399
Analysis 400
Analysis 401
Analysis 402
Encoding Shellcode 402
Analysis 403
Analysis 405
Execution Analysis 407
Reusing Program Variables 407
Open-Source Programs 408
Analysis 409
Closed-Source Programs 409
Execution Analysis 410
Analysis 411
OS-Spanning Shellcode 411
Analysis 412
Understanding Existing Shellcode 412

Analysis 414
Summary 416
Solutions Fast Track 416
Links to Sites 418
Mailing Lists 418
Frequently Asked Questions 419

Chapter 10 Writing Exploits I 421
Introduction 422
Targeting Vulnerabilities 422
Remote and Local Exploits 423
Analysis 424
Format String Attacks 424
Format Strings 424
Analysis 425
Analysis 425
Fixing Format String Bugs 426
Case Study: xlockmore User-Supplied Format String Vulnerability
CVE-2000-0763 427
Vulnerability Details 427
Exploitation Details 427
Analysis 429
TCP/IP Vulnerabilities 429
Race Conditions 430
File Race Conditions 430
Signal Race Conditions 431
Case Study: man Input Validation Error .432
Vulnerability Details 432
Summary 435
Solutions Fast Track 435
Links to Sites 436
Frequently Asked Questions 437

Chapter 11 Writing Exploits II 439
Introduction 440
Coding Sockets and Binding for Exploits 440
Client-Side Socket Programming 441
Analysis 441
Analysis 442
Server-Side Socket Programming 442
Analysis 444
Stack Overflow Exploits 444
Memory Organization 444
Stack Overflows 446
Finding Exploitable Stack Overflows in Open-Source Software 449
Case Study: X11R6 4.2 XLOCALEDIR Overflow 450
The Vulnerability 450

The Exploit .452
Conclusion .454
Finding Exploitable Stack Overflows in Closed-Source Software .454
Heap Corruption Exploits .455
Doug Lea Malloc .456
Analysis .458
Case Study: OpenSSL SSLv2 Malformed Client Key Remote Buffer Overflow Vulnerability CAN-2002-0656 .459
The Vulnerability .460
Exploitation .460
The Complication .461
Improving the Exploit .462
Conclusion .463
Exploit Code for OpenSSL SSLv2 Malformed Client Key Remote Buffer Overflow .463
System V Malloc .468
Analysis .470
Analysis .471
Integer Bug Exploits .472
Integer Wrapping .472
Analysis .473
Analysis .474
Bypassing Size Checks .475
Analysis .475
Analysis .476
Other Integer Bugs .476
Case Study: OpenSSH Challenge Response Integer Overflow Vulnerability CVE-2002-0639 .477
Vulnerability Details .477
Exploitation Details .478
Case Study: UW POP2 Buffer Overflow Vulnerability CVE-1999-0920 .480
Vulnerability Details .480
Summary .488
Solutions Fast Track .488
Links to Sites .489
Frequently Asked Questions .490
Chapter 12 Writing Exploits III . 491
Introduction .492
Using the Metasploit Framework .492
Exploit Development with Metasploit .498
Determining the Attack Vector .499
Finding the Offset .499
Selecting a Control Vector .504
Finding a Return Address .509
Using the Return Address .513

Determining Bad Characters 514
Determining Space Limitations 515
Nop Sleds 517
Choosing a Payload and Encoder 518
Integrating Exploits into the Framework 527
Understanding the Framework 527
Analyzing an Existing Exploit Module 528
Overwriting Methods 533
Summary 534
Solutions Fast Track 534
Links to Sites 535
Frequently Asked Questions 536

Chapter 13 Writing Security Components 539
Introduction 540
COM 540
COM Objects 540
COM Interfaces 541
IUnknown 541
Calling Convention 541
The COM Runtime 541
COM Object Implementation 542
COM Registration 543
HKEY_CLASSES_ROOT\CLSID 544
HKEY_CLASSES_ROOT\CLSID\
{xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx} 544
InprocServer32 544
LocalServer32 544
COM IN-PROCESS Server Implementation 544
DllGetClassObject 545
DllCanUnloadNow 545
DllRegisterServer 545
DllUnregisterServer 545
ATL 546
C++ Templates 546
ATL Client Technologies 547
Smart Pointers 547
Datatype Support 548
BSTR 548
VARIANT 548
ATL Server Technologies 550
Class Composition 550
Interface Definition Language 553
Class Registration 556
COM IN-PROCESS Server Implementation 559
The _AtlModule Global Variable 559

DLL Exports .560
Module Entry Point .561
COM OUT-OF-PROCESS Server Implementation .561
Module Entry Point .562
ATL Attributes .563
Module Attribute .564
Interface Attributes .565
Component Attribute .566
Adding COM Extensions to the RPCDump Tool .567
COM EXE Server Implementation .568
Analysis .570
Control Flow .571
Analysis .572
Application Integration Routines .573
Analysis .574
Tool Interface Definition .575
IRpcEnum .576
IEndpointCollection .577
IEndpoint .578
Component Classes .578
Analysis .579
Analysis .580
Analysis .583
Application Integration: COMSupport.h .584
Analysis .585
Application Integration: RPCDump.C .585
Analysis .585
Analysis .586
Analysis .586
Analysis .586
Analysis .587
Analysis .587
Summary .587
Solutions Fast Track .588
Links to Sites .588
Frequently Asked Questions .589
Chapter 14 Creating a Web Security Tool 593
Introduction .594
Design .594
Attack Signature Format .594
Signatures .595
In-Depth Analysis .595
Sockets and Execution .596
Analysis .603
Parsing .605

Analysis .608
Analysis .614
Header Files .616
Compilation .619
Execution .619
The Usage Screen .620
Tool Output .620
Summary .621
Solutions Fast Track .621
Links to Sites .622
Frequently Asked Questions .622

Appendix A Glossary 625

Appendix B Security Tool Compendium 633
Source Code Auditing .633
Shellcode Tools .634
Debuggers .634
Compilers .634
Hardware Simulators .635
Security Libraries .636
Vulnerability Analysis .636
Network Traffic Analysis .637
Packet Generation .638
Scanners .638

Appendix C Exploit Archives 639
Online Exploit Archives .640

Appendix D Syscall Reference 641
exit(int) .642
open(file, flags, mode) .642
close(filedescriptor) .642
read(filedescriptor, pointer to buffer, amount of bytes) .642
write(filedescriptor, pointer to buffer, amount of bytes) .642
execve(file, file + arguments, environment data) .642
socketcall(callnumber, arguments) .642
socket(domain, type, protocol) .643
bind(file descriptor, sockaddr struct, size of arg 2) .643
listen (file descriptor, number of connections allowed in queue) .643
accept (file descriptor, sockaddr struct, size of arg 2) .643

Appendix E Data Conversion Reference 645

Index 653