

# Contents

<b>Preface</b>	<b>xv</b>
<b>About the Author</b>	<b>xix</b>
<b>Acknowledgments</b>	<b>xxi</b>
<b>I Network Security</b>	<b>1</b>
<b>1 Packet Sniffing and Spoofing</b>	<b>5</b>
1.1 How Packets Are Received . . . . .	6
1.1.1 Network Interface Card (NIC) . . . . .	6
1.1.2 BSD Packet Filter (BPF) . . . . .	7
1.2 Packet Sniffing . . . . .	8
1.2.1 Receiving Packets Using Sockets . . . . .	8
1.2.2 Packet Sniffing using Raw Sockets . . . . .	9
1.2.3 Packet Sniffing Using the <code>pcap</code> API . . . . .	11
1.2.4 Processing Captured Packet . . . . .	12
1.3 Packet Spoofing . . . . .	15
1.3.1 Sending Normal Packets Using Socket . . . . .	16
1.3.2 Sending Spoofed Packets Using Raw Sockets . . . . .	17
1.3.3 Constructing ICMP Packets . . . . .	19
1.3.4 Constructing UDP Packets . . . . .	20
1.4 Sniffing and Then Spoofing . . . . .	22
1.5 Sniffing and Spoofing Using Python and Scapy . . . . .	24
1.5.1 Installing Scapy . . . . .	24
1.5.2 A Simple Example . . . . .	24
1.5.3 Packet Sniffing . . . . .	25
1.5.4 Spoofing ICMP Packets . . . . .	26
1.5.5 Spoofing UDP Packets . . . . .	26
1.5.6 Sniffing and Then Spoofing . . . . .	27
1.5.7 Sending and Receiving Packets . . . . .	27
1.6 Spoofing Packets Using a Hybrid Approach . . . . .	28
1.6.1 A Hybrid Approach . . . . .	28
1.6.2 Constructing Packet Template Using Scapy . . . . .	29
1.6.3 Modifying and Sending Packets Using C . . . . .	29
1.7 Endianness . . . . .	31

1.8	Calculating Checksum . . . . .	32
1.9	Summary . . . . .	34
<b>2</b>	<b>Attacks on the TCP Protocol</b>	<b>35</b>
2.1	How the TCP Protocol Works . . . . .	36
2.1.1	TCP Client Program . . . . .	36
2.1.2	TCP Server Program . . . . .	37
2.1.3	Data Transmission: Under the Hood . . . . .	40
2.1.4	TCP Header . . . . .	41
2.2	SYN Flooding Attack . . . . .	42
2.2.1	TCP Three-Way Handshake Protocol . . . . .	42
2.2.2	The SYN Flooding Attack . . . . .	43
2.2.3	Launching the SYN Flooding Attack . . . . .	44
2.2.4	Launching SYN Flooding Attacks Using C Code . . . . .	46
2.2.5	Countermeasure . . . . .	48
2.3	TCP Reset Attack . . . . .	49
2.3.1	Closing TCP Connections . . . . .	49
2.3.2	How the Attack Works . . . . .	50
2.3.3	Launching the TCP Reset Attack: Setup . . . . .	50
2.3.4	TCP Reset Attack on Telnet connections . . . . .	51
2.3.5	TCP Reset Attack on SSH connections . . . . .	52
2.3.6	TCP Reset Attack on Video-Streaming Connections . . . . .	53
2.4	TCP Session Hijacking Attack . . . . .	55
2.4.1	TCP Session and Session Hijacking . . . . .	55
2.4.2	Launching TCP Session Hijacking Attack . . . . .	56
2.4.3	What Happens to the Hijacked TCP Connection . . . . .	59
2.4.4	Causing More Damage . . . . .	59
2.4.5	Creating Reverse Shell . . . . .	60
2.5	Summary . . . . .	62
<b>3</b>	<b>Firewall</b>	<b>63</b>
3.1	Introduction . . . . .	64
3.2	Types of Firewalls . . . . .	65
3.2.1	Packet Filter . . . . .	65
3.2.2	Stateful Firewall . . . . .	66
3.2.3	Application/Proxy Firewall . . . . .	66
3.3	Building a Simple Firewall using Netfilter . . . . .	66
3.3.1	Writing Loadable Kernel Modules . . . . .	67
3.3.2	Compiling Kernel Modules . . . . .	68
3.3.3	Installing Kernel Modules . . . . .	68
3.4	Netfilter . . . . .	69
3.4.1	netfilter Hooks for IPv4 . . . . .	70
3.4.2	Implementing a Simple Packet Filter Firewall . . . . .	70
3.5	The iptables Firewall in Linux . . . . .	73
3.5.1	The structure of the iptables Firewall . . . . .	73
3.5.2	Traversing Chains and Rule Matching . . . . .	74
3.5.3	iptables Extensions . . . . .	75
3.5.4	Building a Simple Firewall . . . . .	76

3.6	Stateful Firewall using Connection Tracking . . . . .	79
3.6.1	Stateful Firewall . . . . .	79
3.6.2	The Connection Tracking Framework in Linux . . . . .	80
3.6.3	Example: Set up a Stateful Firewall . . . . .	80
3.7	Application/Proxy Firewall and Web Proxy . . . . .	81
3.8	Evading Firewalls . . . . .	82
3.8.1	Using SSH Tunneling to Evade Firewalls . . . . .	82
3.8.2	Dynamic Port Forwarding . . . . .	83
3.8.3	Reverse SSH Tunneling . . . . .	85
3.8.4	Using VPN to Evade Firewall . . . . .	85
3.9	Summary . . . . .	85
<b>4</b>	<b>Domain Name System (DNS) and Attacks</b> . . . . .	<b>87</b>
4.1	DNS Hierarchy, Zones, and Servers . . . . .	88
4.1.1	DNS Domain Hierarchy . . . . .	88
4.1.2	DNS Zone . . . . .	89
4.1.3	Authoritative Name Servers . . . . .	90
4.1.4	The Organization of Zones on the Internet . . . . .	90
4.2	DNS Query Process . . . . .	92
4.2.1	Local DNS Files . . . . .	92
4.2.2	Local DNS Server and the Iterative Query Process . . . . .	93
4.3	Set Up DNS Server and Experiment Environment . . . . .	95
4.3.1	Configure the User Machine . . . . .	96
4.3.2	Configure the Local DNS server . . . . .	96
4.3.3	Set Up Zones in the Local DNS Server . . . . .	98
4.4	Constructing DNS Request and Reply Using Scapy . . . . .	100
4.4.1	DNS Header . . . . .	100
4.4.2	DNS Records . . . . .	101
4.4.3	Example 1: Sending a DNS Query . . . . .	102
4.4.4	Example 2: Implement a Simple DNS Server . . . . .	103
4.5	DNS Attacks: Overview . . . . .	105
4.6	Local DNS Cache Poisoning Attack . . . . .	106
4.6.1	Launch DNS Cache Poisoning Attack . . . . .	107
4.6.2	Targeting the Authority Section . . . . .	109
4.7	Remote DNS Cache Poisoning Attack . . . . .	110
4.7.1	The Kaminsky Attack . . . . .	111
4.7.2	Construct the IP and UDP headers of DNS reply . . . . .	113
4.7.3	Construct the DNS Header and Payload . . . . .	114
4.7.4	Result Verification . . . . .	116
4.8	Reply Forgery Attacks from Malicious DNS Servers . . . . .	117
4.8.1	Fake Data in the Additional Section . . . . .	117
4.8.2	Fake Data in the Authority Section . . . . .	119
4.8.3	Fake Data in Both Authority and Additional Sections . . . . .	120
4.8.4	Fake Data in the Answer Section . . . . .	121
4.8.5	Fake Answer in Reverse DNS Lookup . . . . .	121
4.9	DNS Rebinding Attack . . . . .	123
4.9.1	How DNS Rebinding Attack Works . . . . .	123
4.9.2	Attack Environment Setup . . . . .	125

4.9.3	Set Up the User Machine . . . . .	126
4.9.4	Emulating a Vulnerable IoT Device's Web Server . . . . .	126
4.9.5	Set Up the Web Server on Attacker Computer . . . . .	127
4.9.6	Setting Up the Malicious DNS Server . . . . .	129
4.9.7	Launching the Attack . . . . .	130
4.9.8	Defending Against DNS Rebinding Attack . . . . .	132
4.10	Protection Against DNS Spoofing Attacks . . . . .	132
4.10.1	DNSSEC . . . . .	132
4.10.2	TLS/SSL Solution . . . . .	133
4.11	Denial of Service Attacks on DNS Servers . . . . .	134
4.11.1	Attacks on the Root and TLD Servers . . . . .	134
4.11.2	Attacks on Nameservers of a Particular Domain . . . . .	135
4.12	Summary . . . . .	136
<b>5</b>	<b>Virtual Private Network</b>	<b>137</b>
5.1	Introduction . . . . .	138
5.1.1	Virtual Private Network . . . . .	138
5.1.2	How a Virtual Private Network Works . . . . .	140
5.2	An Overview of How TLS/SSL VPN Works . . . . .	141
5.2.1	Establishing A TLS/SSL Tunnel . . . . .	142
5.2.2	Forwarding IP packets . . . . .	142
5.2.3	Releasing IP Packets . . . . .	143
5.3	How TLS/SSL VPN Works: Details . . . . .	144
5.3.1	Virtual Network Interfaces . . . . .	144
5.3.2	Creating a TUN Interface . . . . .	145
5.3.3	Routing Packets to a TUN Interface . . . . .	147
5.3.4	Reading and Writing Operations on the TUN Interface . . . . .	148
5.3.5	Forwarding Packets via the Tunnel . . . . .	149
5.3.6	Packet's Return Trip . . . . .	149
5.4	Building a VPN . . . . .	149
5.4.1	Establish the Tunnel . . . . .	150
5.4.2	Monitoring File Descriptors . . . . .	152
5.4.3	From TUN To Tunnel . . . . .	152
5.4.4	From Tunnel to TUN . . . . .	153
5.4.5	Bring Everything Together . . . . .	153
5.5	Setting Up a VPN . . . . .	154
5.5.1	Network Configuration . . . . .	154
5.5.2	Configure VPN Server . . . . .	156
5.5.3	Configure VPN Client . . . . .	156
5.5.4	Configure Host V . . . . .	156
5.6	Testing VPN . . . . .	157
5.6.1	Ping Test . . . . .	157
5.6.2	Telnet Test . . . . .	158
5.7	Using VPN to Bypass Egress Firewall . . . . .	159
5.7.1	Network Setup . . . . .	159
5.7.2	Setting Up VPN to Bypass Firewall . . . . .	160
5.8	Summary . . . . .	161

<b>6</b>	<b>Reverse Shell</b>	<b>163</b>
6.1	Introduction	164
6.2	File Descriptor and Redirection	164
6.2.1	File Descriptor	164
6.2.2	Standard IO Devices	166
6.2.3	Redirection	167
6.2.4	How To Implement Redirection	168
6.3	Redirecting Input/Output to a TCP Connection	169
6.3.1	Redirecting Output to a TCP Connection	169
6.3.2	Redirecting Input to a TCP Connection	170
6.3.3	Redirecting to TCP Connection From Shell	171
6.4	Reverse Shell	172
6.4.1	Redirecting the Standard Output	172
6.4.2	Redirecting the Standard Input	172
6.4.3	Redirecting the Standard Error	174
6.4.4	Code Injection	174
6.5	Summary	175
<b>7</b>	<b>The Heartbleed Bug and Attack</b>	<b>177</b>
7.1	Background: the Heartbeat Protocol	178
7.2	Launch the Heartbleed Attack	180
7.2.1	Attack Environment and Setup	180
7.2.2	Launch an Attack	181
7.3	Fixing the Heartbleed Bug	183
7.4	Summary	183
<b>II</b>	<b>Cryptography</b>	<b>185</b>
<b>8</b>	<b>Secret-Key Encryption</b>	<b>189</b>
8.1	Introduction	190
8.2	Substitution Cipher	190
8.2.1	Monoalphabetic Substitution Cipher	190
8.2.2	Breaking Monoalphabetic Substitution Cipher	191
8.2.3	Polyalphabetic Substitution Cipher	194
8.2.4	The Enigma Machine	195
8.3	DES and AES Encryption Algorithms	197
8.3.1	DES: Data Encryption Standard	197
8.3.2	AES: Advanced Encryption Standard	198
8.4	Encryption Modes	198
8.4.1	Encryption Modes	199
8.4.2	Electronic Codebook (ECB) Mode	200
8.4.3	Cipher Block Chaining (CBC) Mode	200
8.4.4	Cipher Feedback (CFB) Mode	202
8.4.5	Output Feedback (OFB) Mode	203
8.4.6	Counter (CTR) Mode	204
8.4.7	Modes for Authenticated Encryption	205
8.4.8	Padding	206

8.5	Initialization Vector and Common Mistakes . . . . .	207
8.5.1	Common Mistake: Use the Same IV . . . . .	207
8.5.2	Common Mistake: Use a Predictable IV . . . . .	210
8.6	Programming using Cryptography APIs . . . . .	213
8.7	Authenticated Encryption and the GCM Mode . . . . .	215
8.7.1	The GCM Mode . . . . .	216
8.7.2	Programming using the GCM Mode . . . . .	217
8.8	Summary . . . . .	218
<b>9</b>	<b>One-Way Hash Function</b>	<b>219</b>
9.1	Introduction . . . . .	220
9.2	Concept and Properties . . . . .	220
9.2.1	Cryptographic Properties . . . . .	220
9.2.2	Replay the Number Game . . . . .	221
9.3	Algorithms and Programs . . . . .	221
9.3.1	The MD (Message Digest) Series . . . . .	222
9.3.2	The SHA (Secure Hash Algorithm) Series . . . . .	222
9.3.3	How Hash Algorithm Works . . . . .	223
9.3.4	One-Way Hash Commands . . . . .	223
9.3.5	Computing One-Way Hash in Programs . . . . .	224
9.3.6	Performance of One-Way Hash Functions . . . . .	226
9.4	Applications of One-Way Hash Functions . . . . .	226
9.4.1	Integrity Verification . . . . .	227
9.4.2	Committing a Secret Without Telling It . . . . .	227
9.4.3	Password Verification . . . . .	228
9.4.4	Trusted Timestamping . . . . .	230
9.5	Message Authentication Code (MAC) . . . . .	231
9.5.1	Constructing MAC and Potential Attacks . . . . .	232
9.5.2	Launching the Length Extension Attack . . . . .	233
9.5.3	Case Study: Length Extension Attack on Flickr . . . . .	236
9.5.4	The Keyed-Hash MAC (HMAC) Algorithm . . . . .	236
9.6	Blockchain and Bitcoins . . . . .	237
9.6.1	Hash Chain and Blockchain . . . . .	237
9.6.2	Make Chaining Difficult . . . . .	238
9.6.3	Adding Incentives and Bitcoin . . . . .	240
9.7	Hash Collision Attacks . . . . .	241
9.7.1	Security Impact of Collision Attacks . . . . .	241
9.7.2	Generating Two Different Files with the Same MD5 Hash . . . . .	242
9.7.3	Generating Two Programs with the Same MD5 Hash . . . . .	244
9.7.4	Making the Two Programs Behave Differently . . . . .	247
9.7.5	Hash-Colliding X.509 Certificates . . . . .	249
9.8	Summary . . . . .	250
<b>10</b>	<b>Public Key Cryptography</b>	<b>251</b>
10.1	Introduction . . . . .	252
10.2	Diffie-Hellman Key Exchange . . . . .	252
10.2.1	Diffie-Hellman Key Exchange . . . . .	253
10.2.2	Turn DH Key Exchange into a Public-Key Encryption Algorithm . . . . .	254

10.3	The RSA Algorithm . . . . .	255
10.3.1	Math Background: Modulo Operation . . . . .	256
10.3.2	Math Background: Euler's Theorem . . . . .	256
10.3.3	Math Background: Extended Euclidean Algorithm . . . . .	257
10.3.4	The RSA Algorithm . . . . .	258
10.3.5	Exercise: Small Number . . . . .	259
10.3.6	Exercise: Large Number . . . . .	260
10.3.7	Performance . . . . .	262
10.3.8	Hybrid Encryption . . . . .	263
10.3.9	Other Public-Key Encryption Algorithms . . . . .	263
10.4	Using OpenSSL Tools to Conduct RSA Operations . . . . .	264
10.4.1	Generating RSA keys . . . . .	264
10.4.2	Extracting the public key . . . . .	265
10.4.3	Encryption and Decryption . . . . .	266
10.5	Paddings for RSA . . . . .	266
10.5.1	Attacks Against Textbook RSA . . . . .	267
10.5.2	Paddings: PKCS#1 v1.5 and OAEP . . . . .	267
10.6	Digital Signature . . . . .	268
10.6.1	Digital Signature using RSA . . . . .	269
10.6.2	DSA and Other Digital Signature Algorithms . . . . .	271
10.7	Programming using Public-Key Cryptography APIs . . . . .	271
10.7.1	Key Generation . . . . .	272
10.7.2	Encryption and Decryption . . . . .	272
10.7.3	Digital Signature . . . . .	274
10.8	Applications . . . . .	276
10.8.1	Authentication . . . . .	276
10.8.2	HTTPS and TLS/SSL . . . . .	278
10.8.3	Chip Technology Used in Credit Cards . . . . .	278
10.9	Blockchain and Bitcoins . . . . .	280
10.10	Summary and Further Learning . . . . .	280
<b>11</b>	<b>Public Key Infrastructure</b> . . . . .	<b>283</b>
11.1	Attack on Public Key Cryptography . . . . .	284
11.1.1	Man-in-the-Middle (MITM) Attack . . . . .	284
11.1.2	Defeating MITM Attacks . . . . .	285
11.1.3	Public Key Infrastructure . . . . .	285
11.2	Public Key Certificates . . . . .	286
11.2.1	X.509 Digital Certificate . . . . .	286
11.2.2	Get Certificate from a Real Server . . . . .	287
11.3	Certificate Authority (CA) . . . . .	288
11.3.1	Being a CA . . . . .	289
11.3.2	Getting X.509 Certificate from CA . . . . .	290
11.3.3	Deploying Public Key Certificate in Web Server . . . . .	293
11.3.4	Apache Setup for HTTPS . . . . .	294
11.4	Root and Intermediate Certificate Authorities . . . . .	295
11.4.1	Root CAs and Self-Signed Certificate . . . . .	295
11.4.2	Intermediate CAs and Chain of Trust . . . . .	296
11.4.3	Creating Certificates for Intermediate CA . . . . .	297

11.4.4	Apache Setup . . . . .	298
11.4.5	Trusted CAs in the Real World . . . . .	298
11.5	How PKI Defeats the MITM Attack . . . . .	299
11.5.1	Attacker Forwards the Authentic Certificate . . . . .	299
11.5.2	Attacker Creates a Fake Certificate . . . . .	299
11.5.3	Attackers Send Their Own Certificates . . . . .	300
11.5.4	The Man-In-The-Middle Proxy . . . . .	301
11.6	Attacks on the Public-Key Infrastructure . . . . .	302
11.6.1	Attack on CA's Verification Process . . . . .	303
11.6.2	Attack on CA's Signing Process . . . . .	303
11.6.3	Attacks on the Algorithms . . . . .	304
11.6.4	Attacks on User Confirmation . . . . .	305
11.7	Types of Digital Certificates . . . . .	305
11.7.1	Domain Validated Certificates (DV) . . . . .	306
11.7.2	Organizational Validated Certificates (OV) . . . . .	306
11.7.3	Extended Validated Certificates (EV) . . . . .	307
11.8	Summary . . . . .	307
<b>12</b>	<b>Transport Layer Security</b>	<b>309</b>
12.1	Overview of TLS . . . . .	310
12.2	TLS Handshake . . . . .	311
12.2.1	Overview of the TLS Handshake Protocol . . . . .	311
12.2.2	Certificate Verification . . . . .	313
12.2.3	Key Generation and Exchange . . . . .	313
12.3	TLS Data Transmission . . . . .	315
12.3.1	Sending Data with TLS Record Protocol . . . . .	315
12.3.2	Receiving Data with TLS Record Protocol . . . . .	316
12.4	TLS Programming: A Client Program . . . . .	317
12.4.1	The Overall Picture . . . . .	318
12.4.2	TLS Initialization . . . . .	318
12.4.3	TCP Connection Setup . . . . .	320
12.4.4	TLS Handshake . . . . .	320
12.4.5	Application Data Transmission . . . . .	321
12.4.6	Set Up the Certificate Folder . . . . .	322
12.4.7	The Complete Client Code . . . . .	323
12.5	Verifying Server's Hostname . . . . .	324
12.5.1	Modified Client Code . . . . .	324
12.5.2	An Experiment: Man-In-The-Middle Attack . . . . .	326
12.5.3	Hostname Checking . . . . .	327
12.6	TLS Programming: the Server Side . . . . .	329
12.6.1	TLS Setup . . . . .	329
12.6.2	TCP Setup . . . . .	331
12.6.3	TLS Handshake . . . . .	331
12.6.4	TLS Data Transmission . . . . .	333
12.6.5	Testing . . . . .	333
12.7	Summary . . . . .	334



<b>13 Bitcoin and Blockchain</b>	<b>337</b>
13.1 History	338
13.2 Cryptography Foundation and Bitcoin Address	339
13.2.1 Generating Private and Public Keys	339
13.2.2 Turning Hash Value Into Bitcoin Address	341
13.2.3 Wallet	344
13.3 Transactions	344
13.3.1 The “Safe” Analogy	345
13.3.2 An Example	346
13.3.3 Input	347
13.3.4 Output	348
13.4 Unlocking the Output of a Transaction	349
13.4.1 Some Fun but Non-standard Locks	350
13.4.2 Pay-to-Pubkey-Hash Type (P2PH)	352
13.4.3 Pay-to-Multisig (P2MS)	353
13.4.4 Pay-to-ScriptHash (P2SH)	354
13.4.5 P2SH Example: Multi-Signature	355
13.4.6 Case Study: A Real Transaction	356
13.4.7 Propagation of Transactions	358
13.5 Blockchain and Mining	358
13.5.1 Generating Blocks	358
13.5.2 Rewarding	359
13.5.3 Transaction and Merkle Tree	360
13.5.4 Branching and Reaching Consensus	361
13.5.5 Double Spending and Majority of Hash Power	363
13.5.6 Case Study: Users with Majority of Hash Power	364
13.6 Summary	365
<b>III Web Security</b>	<b>367</b>
<b>14 Cross Site Request Forgery</b>	<b>371</b>
14.1 Cross-Site Requests and Its Problems	372
14.2 Cross-Site Request Forgery Attack	373
14.3 CSRF Attacks on HTTP GET Services	374
14.3.1 HTTP GET and POST Services	374
14.3.2 The Basic Idea of CSRF Attacks	375
14.3.3 Attack on Elgg’s Add-friend Service	375
14.4 CSRF Attacks on HTTP POST Services	377
14.4.1 Constructing a POST Request Using JavaScript	377
14.4.2 Attack on Elgg’s Edit-Profile Service	378
14.5 Countermeasures	380
14.5.1 Using the referer Header	381
14.5.2 Same-Site Cookies	381
14.5.3 Secret Token	381
14.5.4 Case Study: Elgg’s Countermeasures	382
14.6 Summary	382

<b>15 Cross-Site Scripting Attack</b>	<b>385</b>
15.1 The Cross-Site Scripting Attack	386
15.1.1 Non-persistent (Reflected) XSS Attack	387
15.1.2 Persistent XSS Attack	388
15.1.3 What damage can XSS cause?	388
15.2 XSS Attacks in Action	389
15.2.1 Prelude: Injecting JavaScript Code	389
15.2.2 Use XSS Attacks to Befriend with Others	390
15.2.3 Use XSS Attacks to Change Other People's Profiles	393
15.3 Achieving Self-Propagation	395
15.3.1 Creating a Self-Propagating XSS Worm: the DOM Approach	396
15.3.2 Create a Self-Propagating Worm: the Link Approach	398
15.4 Preventing XSS attacks	399
15.4.1 Getting Rid of Code from User Inputs	399
15.4.2 Defeating XSS Attacks using Content Security Policy	400
15.4.3 Experimenting with Content Security Policy	402
15.5 Summary	404
<b>16 SQL Injection Attack</b>	<b>407</b>
16.1 A Brief Tutorial of SQL	408
16.1.1 Log in to MySQL	408
16.1.2 Create a Database	408
16.1.3 CREATE a Table	408
16.1.4 INSERT a Row	409
16.1.5 The SELECT Statement	409
16.1.6 WHERE Clause	410
16.1.7 UPDATE SQL Statement	411
16.1.8 Comments in SQL Statements	411
16.2 Interacting with Database in Web Application	412
16.2.1 Getting Data from User	412
16.2.2 Getting Data From Database	413
16.3 Launching SQL Injection Attacks	415
16.3.1 Attack Using cURL	416
16.3.2 Modify Database	416
16.3.3 Multiple SQL Statements	417
16.4 The Fundamental Cause	418
16.5 Countermeasures	421
16.5.1 Filtering and Encoding Data	421
16.5.2 Prepared Statement	421
16.6 Summary	423