

Internet Security

v

Contents

Preface	xxi
About the Author	xxv
Acknowledgments	xxvii
I Network Security	1
1 Network Security Basics	5
1.1 The Organization of the Network-Security Module	6
1.2 IP Address and Network Interface	6
1.3 The Life-Cycle of Packet and Protocol Layers	9
1.3.1 Packet Journey at High Level	9
1.3.2 Sending Packets	9
1.3.3 Packet Construction Inside Kernel	10
1.3.4 Receiving Packets	12
1.3.5 Forwarding Packets: Routing	14
1.3.6 Packet-Sending Tools	15
1.4 Packet Sniffing	16
1.4.1 Sniffing with Wireshark	16
1.4.2 Sniffing with <code>tcpdump</code>	16
1.4.3 Sniffing with Scapy	16
1.4.4 Displaying Packets in Scapy Programs	17
1.5 Packet Spoofing	18
1.5.1 Spoofing ICMP Packets	19
1.5.2 Spoofing UDP Packets	20
1.5.3 Sniffing and Then Spoofing	20
1.6 More About Scapy	21
1.6.1 Scapy's Classes for TCP/IP	21
1.6.2 Getting Layers	21
1.6.3 Other Uses of Scapy	22
1.7 Containers and Networks	23
1.7.1 Docker Compose	23
1.7.2 Setting Up Networks	24
1.7.3 Setting Up Containers	25
1.7.4 Sniffing Inside Containers	26

1.8	Summary	27
2	The MAC Layer and Attacks	29
2.1	Introduction	30
2.2	Network Interface Card (NIC)	30
2.2.1	MAC Address	31
2.2.2	Virtual Network Interface	32
2.3	Ethernet Frame	34
2.4	ARP	35
2.4.1	ARP Message Format	36
2.4.2	ARP Cache	37
2.5	ARP Cache Poisoning Attack	38
2.5.1	ARP Cache Poisoning Attack	38
2.5.2	Discussions	40
2.6	Man-In-The-Middle Attack Using ARP Cache Poisoning	41
2.6.1	Launching the ARP Cache Poisoning Attack	41
2.6.2	IP Forwarding	42
2.6.3	Modifying Telnet Data	43
2.6.4	Discussion	45
2.7	Summary	45
3	The Internet Protocol (IP) and Attacks	47
3.1	Introduction	48
3.2	IP Header	48
3.2.1	Time-To-Live (TTL) and Traceroute	50
3.3	IP Fragmentation and Attacks	51
3.3.1	How IP Fragmentation Works	51
3.3.2	Think Like Attackers	53
3.4	Routing	54
3.4.1	Routing on Hosts	54
3.4.2	Routers	55
3.4.3	Reverse Path Filtering (Spoofing Prevention)	56
3.5	ICMP and Attacks	58
3.5.1	ICMP Echo Request/Reply and Ping	59
3.5.2	The Smurf Attack	59
3.5.3	ICMP Redirect Message	60
3.5.4	ICMP Redirect Attack	62
3.5.5	MITM Attack Using ICMP Redirect	64
3.5.6	Other ICMP Attacks	65
3.6	NAT: Network Address Translation	65
3.7	Summary	67
4	Packet Sniffing and Spoofing	69
4.1	Introduction	70
4.2	How Packets Are Received	70
4.2.1	Network Interface Card (NIC)	70
4.2.2	BSD Packet Filter (BPF)	71
4.3	Packet Sniffing	72

4.3.1	Receiving Packets Using Sockets	73
4.3.2	Packet Sniffing using Raw Sockets	74
4.3.3	Packet Sniffing Using the <code>pcap</code> API	76
4.3.4	Processing Captured Packet	78
4.3.5	Packet Sniffing Using Scapy	80
4.3.6	Special Notes Regarding Containers	80
4.4	Packet Spoofing	81
4.4.1	Sending Normal Packets Using Socket	81
4.4.2	Sending Spoofed Packets Using Raw Sockets	82
4.4.3	Constructing ICMP Packets	84
4.4.4	Constructing UDP Packets	86
4.5	Sniffing and Then Spoofing	88
4.6	Spoofing Packets Using a Hybrid Approach	89
4.6.1	A Hybrid Approach	90
4.6.2	Constructing Packet Template Using Scapy	90
4.6.3	Modifying and Sending Packets Using C	91
4.7	Endianness	92
4.8	Calculating Checksum	94
4.9	Summary	95
5	Transport Layer, UDP Protocols and Attacks	97
5.1	The Transport Layer	98
5.1.1	UDP-Based Applications	98
5.1.2	Port Numbers	99
5.2	The UDP Protocol	100
5.2.1	UDP Header	100
5.2.2	UDP Client Program	100
5.2.3	UDP Server Program	101
5.3	Attacks Using UDP	102
5.3.1	Fraggle Attack: Turning One Grenade Into Many	102
5.3.2	UDP Ping Pong: Creating Regenerable Grenade	103
5.3.3	UDP Amplification Attack: Turning Grenade into Missile	104
5.4	Summary	104
6	Attacks on the TCP Protocol	105
6.1	Introduction	106
6.2	How the TCP Protocol Works	106
6.2.1	TCP Client Program in Python	106
6.2.2	TCP Client Program in C	107
6.2.3	TCP Server Program in Python	108
6.2.4	TCP Server Program in C	108
6.2.5	Data Transmission: Under the Hood	111
6.2.6	TCP Header	112
6.3	SYN Flooding Attack	113
6.3.1	TCP Three-Way Handshake Protocol	114
6.3.2	The SYN Flooding Attack	115
6.3.3	Launching the SYN Flooding Attack	116
6.3.4	Issues in SYN Flooding Attacks	118

6.3.5	Launching SYN Flooding Attacks Using C Code	120
6.3.6	Countermeasure	122
6.4	TCP Reset Attack	123
6.4.1	Closing TCP Connections	123
6.4.2	How the Attack Works	124
6.4.3	Launching the TCP Reset Attack: Setup	124
6.4.4	TCP Reset Attack on Telnet connections	124
6.4.5	TCP Reset Attack on SSH connections	126
6.4.6	TCP Reset Attack on Video-Streaming Connections	126
6.5	TCP Session Hijacking Attack	128
6.5.1	TCP Session and Session Hijacking	128
6.5.2	Launching TCP Session Hijacking Attack	129
6.5.3	What Happens to the Hijacked TCP Connection	132
6.5.4	Causing More Damage	133
6.5.5	Creating Reverse Shell	134
6.6	The Mitnick Attack	135
6.6.1	How the Mitnick Attack Works	135
6.6.2	Experiment Setup for the Mitnick Attack	137
6.6.3	Silencing the Trusted Server	137
6.6.4	Spoofing SYN Packet	138
6.6.5	Spoofing SYN+ACK	138
6.6.6	Launching the Attack	140
6.7	Summary	140
7	Firewall	143
7.1	Introduction	144
7.2	Types of Firewalls	145
7.2.1	Packet Filter	145
7.2.2	Stateful Firewall	146
7.2.3	Application/Proxy Firewall	147
7.3	Implementing a Simple Firewall Using Netfilter	147
7.3.1	Writing Loadable Kernel Modules	147
7.3.2	Netfilter and Hooks	150
7.3.3	Hooking Functions to <code>netfilter</code>	152
7.3.4	Experimenting with the Hook Functions	153
7.3.5	Implementing A Simple Firewall	154
7.3.6	Other Applications of <code>netfilter</code>	156
7.4	Configuring Linux Firewall Using <code>iptables</code>	156
7.4.1	The Structure of the <code>iptables</code> Firewall	157
7.4.2	Traversing Chains and Rule Matching	158
7.4.3	Setting Firewall Rules Using <code>iptables</code>	159
7.4.4	<code>iptables</code> Match Extensions	160
7.4.5	<code>iptables</code> Target Extensions	161
7.5	Connection Tracking and Stateful Firewall	162
7.5.1	Connection Tracking	163
7.5.2	Using Connection Tracking in Firewall	164
7.6	Application/Proxy Firewall and Web Proxy	166
7.7	Summary	167

8	Virtual Private Network	169
8.1	Introduction	170
8.1.1	Virtual Private Network	170
8.1.2	How a Virtual Private Network Works	172
8.2	An Overview of How TLS/SSL VPN Works	173
8.2.1	Establishing A TLS/SSL Tunnel	174
8.2.2	Forwarding IP packets	174
8.2.3	Releasing IP Packets	176
8.2.4	Experiment Environment Setup	176
8.3	Creating and Using the TUN Interface	177
8.3.1	Virtual Network Interfaces	177
8.3.2	Creating a TUN Interface	178
8.3.3	Reading from the TUN Interface	180
8.3.4	Writing to the TUN Interface	181
8.4	Implementing the IP Tunnel	182
8.4.1	Feeding Packets to the Tunnel	182
8.4.2	Pulling Packets Across the Tunnel	183
8.4.3	Releasing the Packets Inside the Private Network	184
8.5	Pulling the Return Packets Back to Client	186
8.5.1	Step 1. Routing Packets Towards the Tunnel	186
8.5.2	Step 2. Routing Packets to the TUN Interface	187
8.5.3	Step 3. Sending Packets Across the Tunnel	187
8.5.4	Step 4. Releasing the Packets on the Client Side	189
8.6	Testing VPN	190
8.6.1	Ping Test	190
8.6.2	Telnet Test	190
8.7	Connecting Two Private Networks Using TUN	191
8.8	Bridging Two Private Networks Using TAP	192
8.8.1	Creating a TAP Interface	192
8.8.2	Connecting the TAP Interface to Network via Bridging	194
8.8.3	More About Linux Bridge	195
8.8.4	Bridging Two Networks: Setup	196
8.8.5	Bridging Two Networks via TAP	197
8.9	Summary	198
9	Tunneling and Firewall Evasion	201
9.1	Introduction	202
9.1.1	The General Ideas of Tunneling	202
9.1.2	Network Setup	203
9.2	VPN: IP Tunneling	204
9.2.1	Creating VPN Using SSH	204
9.2.2	Using VPN to Bypass Ingress Filtering	205
9.2.3	Using VPN to Bypass Egress Firewall	207
9.2.4	Bypassing Geo-Restriction	209
9.3	Port Forwarding: SSH Tunneling	209
9.3.1	Port Forwarding: Evading Ingress Firewall	210
9.3.2	Reverse SSH Tunneling	212
9.4	Dynamic Port Forwarding and SOCKS Proxy	213

9.4.1	Dynamic Port Forwarding	213
9.4.2	Testing the Proxy Using Browser	214
9.4.3	The SOCKS Protocol	215
9.4.4	Using SOCKS in Python	216
9.4.5	Difference Between SOCKS5 and VPN	217
9.5	Other Tunneling Methods	217
9.6	Summary	217
10	DNS and DNS Attacks	219
10.1	DNS Hierarchy, Zones, and Servers	220
10.1.1	DNS Domain Hierarchy	220
10.1.2	DNS Zone	221
10.1.3	Authoritative Name Servers	222
10.1.4	The Organization of Zones on the Internet	222
10.2	DNS Query Process	224
10.2.1	Local DNS Files	224
10.2.2	Local DNS Server and the Iterative Query Process	225
10.3	Set Up DNS Server and Experiment Environment	227
10.3.1	Configure the User Machine	228
10.3.2	Configure the Local DNS server	228
10.3.3	Configure the Attacker's Nameserver	230
10.3.4	Add Forward Zones to Local DNS Server	232
10.4	Constructing DNS Request and Reply Using Scapy	233
10.4.1	DNS Header	233
10.4.2	DNS Records	234
10.4.3	Example 1: Sending a DNS Query	235
10.4.4	Example 2: Implement a Simple DNS Server	236
10.5	DNS Attacks: Overview	237
10.6	Local DNS Cache Poisoning Attack	239
10.6.1	Launch DNS Cache Poisoning Attack	240
10.6.2	Targeting the Authority Section	242
10.7	Remote DNS Cache Poisoning Attack	243
10.7.1	The Kaminsky Attack	244
10.7.2	Construct the IP and UDP headers of DNS reply	246
10.7.3	Construct the DNS Header and Payload	247
10.7.4	Launch the Attack	250
10.8	Reply Forgery Attacks from Malicious DNS Servers	251
10.8.1	Fake Data in the Additional Section	252
10.8.2	Fake Data in the Authority Section	253
10.8.3	Fake Data in Both Authority and Additional Sections	254
10.8.4	Fake Data in the Answer Section	255
10.8.5	Fake Answer in Reverse DNS Lookup	256
10.9	DNS Rebinding Attack	257
10.9.1	How DNS Rebinding Attack Works	258
10.9.2	Attack Environment Setup	259
10.9.3	Set Up the User Machine	261
10.9.4	Emulating a Vulnerable IoT Device's Web Server	262
10.9.5	Understanding the Same-Origin Policy Protection	262

10.9.6	Defeating the Same Origin Policy	264
10.9.7	Launching the Attack	265
10.9.8	Defending Against DNS Rebinding Attack	266
10.10	Denial of Service Attacks on DNS Servers	266
10.10.1	Attacks on the Root and TLD Servers	267
10.10.2	Attacks on Nameservers of a Particular Domain	267
10.11	Summary	268
11	DNSSEC: Securing DNS	271
11.1	How DNSSEC Works	272
11.1.1	A High-Level Picture	272
11.1.2	Public Keys	273
11.1.3	Generating KSK and ZSK Keys for Zone	274
11.1.4	Signing the Zone File	274
11.1.5	The DS record for the Key Signing Key	274
11.2	Hands-on Experience on DNSSEC	275
11.2.1	Setting Up Nameserver for <code>example.edu</code>	276
11.2.2	Setting Up Nameserver for <code>edu</code>	278
11.2.3	Setting Up Nameserver for <code>root</code>	278
11.2.4	Setting Up the Local DNS Server	279
11.2.5	Enabling DNSSEC on Local DNS Server	281
11.3	TLS/SSL Solution	281
11.4	Summary	282
12	BGP and Attacks	283
12.1	Introduction	284
12.2	Physical Infrastructure	286
12.2.1	Autonomous Systems	286
12.2.2	Internet Exchange and Peering	286
12.2.3	Laying Cable	287
12.2.4	Case Studies	287
12.3	The BGP Protocol: Overview	289
12.3.1	Routing Protocols	290
12.3.2	The BGP Protocol: A High-Level Explanation	290
12.4	The SEED Internet Emulator	292
12.4.1	The SEED Internet Emulator	292
12.4.2	BIRD: The Routing Software Used in the Emulator	293
12.4.3	Pipe Between Tables	295
12.4.4	BGP Routing Table vs. Kernel Routing Table	295
12.4.5	The Mandatory <code>device</code> Protocol	296
12.5	BGP: IP Prefixes Owned By AS	296
12.5.1	Route Generation Using the <code>direct</code> Protocol	296
12.5.2	Routes Generated From the <code>static</code> Protocol	298
12.5.3	ASN and Its IP Prefixes	298
12.6	BGP Peering	299
12.6.1	Establishing Peering Relationship	299
12.6.2	Import and Export Filters	301
12.6.3	Peering via Route Server	301

12.7	BGP UPDATE Message	302
12.7.1	Route Withdrawal	303
12.7.2	Route Advertisements	304
12.7.3	TTL and BGP TTL Security Hack	306
12.8	Path Selection	307
12.8.1	The Best Path Selection Algorithm	308
12.8.2	Local Preference Value	309
12.8.3	AS Path Prepending	310
12.9	BGP Large Communities	312
12.9.1	Communities Defined in the Emulator	312
12.9.2	The Local Community	313
12.9.3	The Provider and Customer Communities	314
12.9.4	The Peer Community	314
12.10	BGP for Transit Autonomous System	315
12.10.1	Internal BGP (IBGP)	316
12.10.2	Experimenting with IBGP in AS-3	318
12.10.3	Interior Gateway Protocol (IGP)	319
12.10.4	Experimenting with IGP in AS-3	320
12.11	IP Anycast: a BGP Application	321
12.11.1	An Example of IP Anycast	322
12.11.2	How IP Anycast Works	322
12.11.3	Applications of IP Anycast	324
12.12	BGP Hijacking Attack	324
12.12.1	Routing Rule: Longest Match	324
12.12.2	IP Prefix Hijacking	325
12.12.3	Fighting Back	327
12.12.4	Filtering Out Spoofed Advertisement	328
12.12.5	Defending Against IP Prefix Hijacking	329
12.13	Summary	329
13	The Heartbleed Bug and Attack	331
13.1	Background: the Heartbeat Protocol	332
13.2	Launch the Heartbleed Attack	334
13.2.1	Attack Environment and Setup	334
13.2.2	Launch an Attack	335
13.3	Fixing the Heartbleed Bug	337
13.4	Summary	337
14	Reverse Shell	339
14.1	Introduction	340
14.2	File Descriptor and Redirection	340
14.2.1	File Descriptor	340
14.2.2	Standard IO Devices	342
14.2.3	Redirection	343
14.2.4	Understanding the Syntax of Redirection	344
14.2.5	How To Implement Redirection	345
14.3	Redirecting Input/Output to a TCP Connection	347
14.3.1	Redirecting Output to a TCP Connection	347

14.3.2	Redirecting Input to a TCP Connection	348
14.3.3	Redirecting to TCP Connection From Shell	349
14.4	Reverse Shell	350
14.4.1	Redirecting the Standard Output	350
14.4.2	Redirecting the Standard Input	350
14.4.3	Redirecting the Standard Error	352
14.4.4	Code Injection	352
14.5	Summary	353
II	Cryptography	355
15	Secret-Key Encryption	359
15.1	Introduction	360
15.2	Substitution Cipher	360
15.2.1	Monoalphabetic Substitution Cipher	360
15.2.2	Breaking Monoalphabetic Substitution Cipher	361
15.2.3	Polyalphabetic Substitution Cipher	365
15.2.4	The Enigma Machine	366
15.3	DES and AES Encryption Algorithms	367
15.3.1	DES: Data Encryption Standard	367
15.3.2	AES: Advanced Encryption Standard	368
15.4	Encryption Modes	369
15.4.1	Encryption Modes	370
15.4.2	Electronic Codebook (ECB) Mode	370
15.4.3	Cipher Block Chaining (CBC) Mode	370
15.4.4	Cipher Feedback (CFB) Mode	372
15.4.5	Output Feedback (OFB) Mode	374
15.4.6	Counter (CTR) Mode	374
15.4.7	Modes for Authenticated Encryption	375
15.4.8	Padding	376
15.5	Initialization Vector and Attacks	378
15.5.1	Mistake: Using the Same IV	378
15.5.2	Mistake: Using a Predictable IV	380
15.6	The Padding Oracle Attack	383
15.6.1	The Experiment Setup	384
15.6.2	How the Attack Works: the Main Idea	384
15.6.3	Finding the Value of $D_2 [15]$	385
15.6.4	Finding the Value of $D_2 [14]$	387
15.6.5	Finding the Value of $D_2 [13]$	387
15.7	Programming using Cryptography APIs	388
15.8	Authenticated Encryption and the GCM Mode	390
15.8.1	The GCM Mode	391
15.8.2	Programming using the GCM Mode	392
15.9	Summary	394

16 One-Way Hash Function	395
16.1 Introduction	396
16.2 Concept and Properties	396
16.2.1 Cryptographic Properties	396
16.2.2 Replay the Number Game	397
16.3 Algorithms and Programs	397
16.3.1 The MD (Message Digest) Series	398
16.3.2 The SHA (Secure Hash Algorithm) Series	398
16.3.3 How Hash Algorithm Works	399
16.3.4 One-Way Hash Commands	399
16.3.5 Computing One-Way Hash in Programs	400
16.3.6 Performance of One-Way Hash Functions	402
16.4 Applications of One-Way Hash Functions	402
16.4.1 Integrity Verification	402
16.4.2 Committing a Secret Without Telling It	403
16.4.3 Password Verification	404
16.4.4 Trusted Timestamping	407
16.5 Message Authentication Code (MAC)	408
16.5.1 Constructing MAC and Potential Attacks	409
16.5.2 Launching the Length Extension Attack	410
16.5.3 Case Study: Length Extension Attack on Flickr	412
16.5.4 The Keyed-Hash MAC (HMAC) Algorithm	413
16.6 Blockchain and Bitcoins	414
16.6.1 Hash Chain and Blockchain	414
16.6.2 Make Chaining Difficult	415
16.6.3 Adding Incentives and Bitcoin	417
16.7 Hash Collision Attacks	417
16.7.1 Security Impact of Collision Attacks	418
16.7.2 Generating Two Different Files with the Same MD5 Hash	419
16.7.3 Generating Two Programs with the Same MD5 Hash	421
16.7.4 Making the Two Programs Behave Differently	423
16.7.5 Hash-Colliding X.509 Certificates	426
16.8 Summary	426
17 Public Key Cryptography	429
17.1 Introduction	430
17.2 Diffie-Hellman Key Exchange	430
17.2.1 Diffie-Hellman Key Exchange	431
17.2.2 Turn DH Key Exchange into a Public-Key Encryption Algorithm	432
17.3 The RSA Algorithm	433
17.3.1 Math Background: Modulo Operation	434
17.3.2 Math Background: Euler's Theorem	434
17.3.3 Math Background: Extended Euclidean Algorithm	435
17.3.4 The RSA Algorithm	436
17.3.5 Exercise: Small Number	437
17.3.6 Exercise: Large Number	438
17.3.7 Performance	440
17.3.8 Hybrid Encryption	441

17.3.9	Other Public-Key Encryption Algorithms	442
17.4	Using OpenSSL Tools to Conduct RSA Operations	442
17.4.1	Generating RSA keys	442
17.4.2	Extracting the public key	444
17.4.3	Encryption and Decryption	444
17.5	Padding for RSA	444
17.5.1	Attacks Against Textbook RSA	445
17.5.2	Padding: PKCS#1 v1.5 and OAEP	445
17.6	Digital Signature	447
17.6.1	Digital Signature using RSA	447
17.6.2	DSA and Other Digital Signature Algorithms	449
17.7	Programming Using Public-Key Cryptography APIs	449
17.7.1	Key Generation	450
17.7.2	Encryption and Decryption	451
17.7.3	Digital Signature	452
17.8	Applications	454
17.8.1	Authentication	454
17.8.2	HTTPS and TLS/SSL	456
17.8.3	Chip Technology Used in Credit Cards	457
17.9	Blockchain and Bitcoins	459
17.10	Summary and Further Learning	459
18	Public Key Infrastructure	461
18.1	Attack on Public Key Cryptography	462
18.1.1	Man-in-the-Middle (MITM) Attack	462
18.1.2	Defeating MITM Attacks	463
18.1.3	Public Key Infrastructure	463
18.2	Public Key Certificates	464
18.2.1	X.509 Digital Certificate	464
18.2.2	Get Certificate from a Real Server	465
18.3	Certificate Authority (CA)	466
18.3.1	The Core Functionalities of CA	466
18.3.2	Becoming a CA and Setup	467
18.3.3	Generating Keys and Certificates	468
18.4	Getting Certificate from CA	468
18.4.1	Generating Public/Private Keys	468
18.4.2	Generating Certificate Signing Request	469
18.4.3	Adding Alternative Names	470
18.4.4	Asking CA to Sign the Request	471
18.5	Using Public Key Certificate to Secure Web Servers	471
18.5.1	OpenSSL's Built-in Server	472
18.5.2	Apache Setup for HTTPS	473
18.6	Root and Intermediate Certificate Authorities	474
18.6.1	Root CAs and Self-Signed Certificate	474
18.6.2	Intermediate CAs and Chain of Trust	475
18.6.3	Creating Certificates for Intermediate CA	476
18.6.4	Apache Setup	477
18.6.5	Trusted CAs in the Real World	478

18.7	How PKI Defeats the MITM Attack	478
18.7.1	Attacker Forwards the Authentic Certificate	478
18.7.2	Attacker Creates a Fake Certificate	479
18.7.3	Attackers Send Their Own Certificates	479
18.7.4	The Man-In-The-Middle Proxy	480
18.8	Attacks on the Public-Key Infrastructure	481
18.8.1	Attack on CA's Verification Process	482
18.8.2	Attack on CA's Signing Process	483
18.8.3	Attacks on the Algorithms	483
18.8.4	Attacks on User Confirmation	484
18.9	Types of Digital Certificates	485
18.9.1	Domain Validated Certificates (DV)	485
18.9.2	Organizational Validated Certificates (OV)	486
18.9.3	Extended Validated Certificates (EV)	486
18.10	Summary	487
19	Transport Layer Security	489
19.1	Overview of TLS	490
19.2	TLS Handshake (Version 1.2)	491
19.2.1	Overview of the TLS Handshake Protocol	491
19.2.2	Certificate Verification	493
19.2.3	Key Generation and Exchange	494
19.3	TLS Handshake (Version 1.3)	495
19.4	TLS Data Transmission	496
19.4.1	Sending Data with TLS Record Protocol	497
19.4.2	Receiving Data with TLS Record Protocol	498
19.5	TLS Client Program in Python	498
19.5.1	TLS Setup and Handshake	499
19.5.2	The Trusted Certificates	500
19.5.3	Application Data Transmission	502
19.6	Verifying Server's Hostname	502
19.6.1	An Experiment: Man-In-The-Middle Attack	502
19.6.2	Hostname Checking	504
19.7	TLS Server Program in Python	505
19.8	TLS Proxy	507
19.8.1	The Idea of TLS Proxy	508
19.8.2	Certificate Setup	509
19.8.3	The code for a Simple HTTPS Proxy	509
19.9	Summary	511
20	Bitcoin and Blockchain	513
20.1	History	514
20.2	Cryptography Foundation and Bitcoin Address	515
20.2.1	Generating Private and Public Keys	515
20.2.2	Turning Hash Value Into Bitcoin Address	517
20.2.3	Wallet	520
20.3	Transactions	520
20.3.1	The "Safe" Analogy	520

20.3.2	An Example	522
20.3.3	Input	523
20.3.4	Output	524
20.4	Unlocking the Output of a Transaction	525
20.4.1	Some Fun but Non-standard Locks	526
20.4.2	Pay-to-Pubkey-Hash Type (P2PH)	528
20.4.3	Pay-to-Multisig (P2MS)	529
20.4.4	Pay-to-ScriptHash (P2SH)	530
20.4.5	P2SH Example: Multi-Signature	531
20.4.6	Case Study: A Real Transaction	532
20.4.7	Propagation of Transactions	533
20.5	Blockchain and Mining	534
20.5.1	Generating Blocks	534
20.5.2	Rewarding	535
20.5.3	Transaction and Merkle Tree	536
20.5.4	Branching and Reaching Consensus	537
20.5.5	Double Spending and Majority of Hash Power	539
20.5.6	Case Study: Users with Majority of Hash Power	539
20.6	Summary	541

III Web Security 543

21	Web Security Basics 547
21.1	The Web Architecture 548
21.2	Web Browser 548
21.2.1	HTML and Document Object Model (DOM) 548
21.2.2	CSS: Cascading Style Sheets 549
21.2.3	Dynamic Content 549
21.2.4	JavaScript 550
21.3	Web Server: HTTP Server and Web Applications 550
21.3.1	Case Study: Apache Server 551
21.3.2	How HTTP Server Interacts with Web Applications 552
21.4	Browser-Server Communication: The HTTP Protocol 553
21.4.1	Types of HTTP Requests: GET and POST 554
21.4.2	HTTPS 555
21.5	Cookies and Sessions 555
21.5.1	The Stateless Nature 555
21.5.2	Cookies 556
21.5.3	Tracking Using Cookies 556
21.5.4	Sessions and Session Cookies 558
21.6	Sandboxing JavaScript 558
21.6.1	Access Page Data and Document Object Model (DOM) 560
21.6.2	Access Browser Data 561
21.6.3	Access File Systems 561
21.7	Ajax Request and Security 562
21.7.1	Ajax Example 563
21.7.2	Same Origin Policy on Ajax 563

21.7.3	Cross-Domain Ajax Request	564
21.7.4	Case Study: Bypassing Same Origin Policies	564
21.7.5	WebSocket	565
21.8	Summary	566
22	Cross Site Request Forgery	567
22.1	Cross-Site Requests and Its Problems	568
22.2	Cross-Site Request Forgery Attack	569
22.3	CSRF Attacks on HTTP GET Services	570
22.3.1	HTTP GET and POST Services	570
22.3.2	The Basic Idea of CSRF Attacks	571
22.3.3	Attack on Elgg's Add-friend Service	571
22.4	CSRF Attacks on HTTP POST Services	573
22.4.1	Constructing a POST Request Using JavaScript	573
22.4.2	Attack on Elgg's Edit-Profile Service	574
22.5	Countermeasures	576
22.5.1	Using the <code>referer</code> Header	577
22.5.2	Same-Site Cookies	577
22.5.3	Secret Token	579
22.5.4	Case Study: Elgg's Countermeasures	580
22.6	Summary	581
23	Cross-Site Scripting Attack	583
23.1	The Cross-Site Scripting Attack	584
23.1.1	Non-persistent (Reflected) XSS Attack	585
23.1.2	Persistent XSS Attack	586
23.1.3	What damage can XSS cause?	586
23.2	XSS Attacks in Action	587
23.2.1	Prelude: Injecting JavaScript Code	587
23.2.2	Use XSS Attacks to Befriend with Others	588
23.2.3	Use XSS Attacks to Change Other People's Profiles	591
23.3	Achieving Self-Propagation	593
23.3.1	Creating a Self-Propagating XSS Worm: the DOM Approach	594
23.3.2	Create a Self-Propagating Worm: the Link Approach	596
23.4	Preventing XSS attacks	596
23.4.1	Getting Rid of Code from User Inputs	597
23.4.2	Defeating XSS Attacks using Content Security Policy	597
23.4.3	Experimenting with Content Security Policy	600
23.5	JavaScript Code Injection Attacks in General	602
23.5.1	Attack From Third-Party Websites	602
23.5.2	Attacks on Web-Based Mobile Apps	604
23.6	Summary	606
24	SQL Injection Attack	609
24.1	A Brief Tutorial of SQL	610
24.1.1	Log into MySQL	610
24.1.2	Create a Database	610
24.1.3	CREATE a Table	611

24.1.4	INSERT a Row	611
24.1.5	The SELECT Statement	612
24.1.6	WHERE Clause	612
24.1.7	UPDATE SQL Statement	613
24.1.8	Comments in SQL Statements	613
24.2	Interacting with Database in Web Application	614
24.2.1	Getting Data from User	615
24.2.2	Getting Data From Database	616
24.3	Launching SQL Injection Attacks	617
24.3.1	Attack Using cURL	618
24.3.2	Modify Database	619
24.3.3	Multiple SQL Statements	620
24.4	The Fundamental Cause	621
24.5	Countermeasures	623
24.5.1	Filtering and Encoding Data	623
24.5.2	Prepared Statement	624
24.5.3	Defeating SQL Injection Using Prepared Statements	626
24.6	Summary	627
25	Clickjacking Attacks	629
25.1	Prelude	630
25.2	Introduction and Background	630
25.2.1	Overlapping Iframe	630
25.2.2	Opacity	631
25.3	Clickjacking Attacks Using Transparent Iframe	631
25.3.1	Likejacking	631
25.3.2	Hijacking Other Actions	633
25.3.3	Sequence of Clicks	633
25.4	Clickjacking Using Non-Transparent Iframe	634
25.4.1	Likejacking Using Small-Size Iframe	634
25.4.2	Stealing Login Credentials	635
25.5	Countermeasures	636
25.5.1	Framekiller and Framebuster	637
25.5.2	X-Frame-Options	637
25.5.3	Content-Security Policy	639
25.6	Security on Iframes	641
25.6.1	Same Origin Policy	641
25.6.2	Sandboxing Iframes	643
25.7	Summary	644
26	Shellshock Attack	645
26.1	Background: Shell Functions	646
26.2	The Shellshock Vulnerability	648
26.2.1	Vulnerable Version of bash	648
26.2.2	The Shellshock Bug	648
26.2.3	Mistake in the Bash Source Code	649
26.2.4	How Was the Vulnerability Fixed	650
26.2.5	Exploiting the Shellshock vulnerability	651

26.3	Shellshock Attack on Set-UID Programs	651
26.4	Shellshock Attack on CGI Programs	653
26.4.1	Experiment Environment Setup	653
26.4.2	How Web Server Invokes CGI Programs	654
26.4.3	How Attacker Sends Data to Bash	655
26.4.4	Launching the Shellshock Attack	656
26.4.5	Creating Reverse Shell	657
26.5	Remote Attack on PHP	659
26.6	Summary	660