

Table of Contents

Preface	xxi
1. Introduction	xxi
2. Intended Audience and Organization	xxi
3. Conventions	xxii
4. Other Resources	xxiii
5. Request for Comments	xxiv
6. Acknowledgements	xxiv
6.1. Technology Used to Create This Book	xxv
7. TCP/IP Reference	xxvi
1. Getting Started with Nmap	1
1.1. Introduction	1
1.2. Nmap Overview and Demonstration	1
1.2.1. Avatar Online	1
1.2.2. Saving the Human Race	8
1.2.3. MadHat in Wonderland	9
1.3. The Phases of an Nmap Scan	12
1.4. Legal Issues	13
1.4.1. Is Unauthorized Port Scanning a Crime?	14
1.4.2. Can Port Scanning Crash the Target Computer/Networks?	19
1.4.3. Nmap Copyright	20
1.5. The History and Future of Nmap	20
2. Obtaining, Compiling, Installing, and Removing Nmap	25
2.1. Introduction	25
2.1.1. Testing Whether Nmap is Already Installed	25
2.1.2. Command-line and Graphical Interfaces	25
2.1.3. Downloading Nmap	26
2.1.4. Verifying the Integrity of Nmap Downloads	26
2.1.5. Obtaining Nmap from the Subversion (SVN) Repository	28
2.2. Unix Compilation and Installation from Source Code	29
2.2.1. Configure Directives	30
2.2.2. If You Encounter Compilation Problems	32
2.3. Linux Distributions	33
2.3.1. RPM-based Distributions (Red Hat, Mandrake, SUSE, Fedora)	33
2.3.2. Updating Red Hat, Fedora, Mandrake, and Yellow Dog Linux with Yum	34
2.3.3. Debian Linux and Derivatives such as Ubuntu	35
2.3.4. Other Linux Distributions	35
2.4. Windows	36
2.4.1. Windows 2000 Dependencies	37
2.4.2. Windows Self-installer	37
2.4.3. Command-line Zip Binaries	37
Installing the Nmap zip binaries	37
2.4.4. Compile from Source Code	38
2.4.5. Executing Nmap on Windows	39
2.5. Sun Solaris	40
2.6. Apple Mac OS X	41
2.6.1. Executable Installer	41



2.6.2. Compile from Source Code	41
Compile Nmap from source code	41
Compile Zenmap from source code	42
2.6.3. Third-party Packages	42
2.6.4. Executing Nmap on Mac OS X	42
2.7. FreeBSD / OpenBSD / NetBSD	43
2.7.1. OpenBSD Binary Packages and Source Ports Instructions	43
2.7.2. FreeBSD Binary Package and Source Ports Instructions	44
Installation of the binary package	44
Installation using the source ports tree	44
2.7.3. NetBSD Binary Package Instructions	44
2.8. Amiga, HP-UX, IRIX, and Other Platforms	44
2.9. Removing Nmap	45
3. Host Discovery (Ping Scanning)	47
3.1. Introduction	47
3.2. Specifying Target Hosts and Networks	47
3.2.1. Input From List (-iL)	48
3.2.2. Choose Targets at Random (-iR <numtargets>)	48
3.2.3. Excluding Targets (--exclude, --excludefile <filename>)	48
3.2.4. Practical Examples	49
3.3. Finding an Organization's IP Addresses	49
3.3.1. DNS Tricks	50
3.3.2. Whois Queries Against IP Registries	54
3.3.3. Internet Routing Information	55
3.4. DNS Resolution	56
3.5. Host Discovery Controls	57
3.5.1. List Scan (-sL)	57
3.5.2. Ping Scan (-sP)	58
3.5.3. Disable Ping (-PN)	59
3.6. Host Discovery Techniques	60
3.6.1. TCP SYN Ping (-PS<port list>)	61
3.6.2. TCP ACK Ping (-PA<port list>)	62
3.6.3. UDP Ping (-PU<port list>)	63
3.6.4. ICMP Ping Types (-PE, -PP, and -PM)	64
3.6.5. IP Protocol Ping (-PO<protocol list>)	64
3.6.6. ARP Scan (-PR)	64
3.6.7. Default Combination	66
3.7. Putting It All Together: Host Discovery Strategies	66
3.7.1. Related Options	66
3.7.2. Choosing and Combining Ping Options	68
TCP probe and port selection	68
UDP port selection	70
ICMP probe selection	70
Designing the ideal combinations of probes	70
3.8. Host Discovery Code Algorithms	72
4. Port Scanning Overview	73
4.1. Introduction to Port Scanning	73
4.1.1. What Exactly is a Port?	73
4.1.2. What Are the Most Popular Ports?	75



4.1.3. What is Port Scanning?	77
4.1.4. Why Scan Ports?	78
4.2. A Quick Port Scanning Tutorial	79
4.3. Command-line Flags	82
4.3.1. Selecting Scan Techniques	82
4.3.2. Selecting Ports to Scan	83
4.3.3. Timing-related Options	85
4.3.4. Output Format and Verbosity Options	85
4.3.5. Firewall and IDS Evasion Options	87
4.3.6. Specifying Targets	87
4.3.7. Miscellaneous Options	87
4.4. IPv6 Scanning (-6)	88
4.5. SOLUTION: Scan a Large Network for a Certain Open TCP Port	88
4.5.1. Problem	88
4.5.2. Solution	89
4.5.3. Discussion	89
4.5.4. See Also	94
5. Port Scanning Techniques and Algorithms	95
5.1. Introduction	95
5.2. TCP SYN (Stealth) Scan (-sS)	96
5.3. TCP Connect Scan (-sT)	100
5.4. UDP Scan (-sU)	101
5.4.1. Disambiguating Open from Filtered UDP Ports	102
5.4.2. Speeding Up UDP Scans	105
5.5. TCP FIN, NULL, and Xmas Scans (-sF, -sN, -sX)	107
5.6. Custom Scan Types with --scanflags	111
5.6.1. Custom SYN/FIN Scan	111
5.6.2. PSH Scan	112
5.7. TCP ACK Scan (-sA)	113
5.8. TCP Window Scan (-sW)	115
5.9. TCP Maimon Scan (-sM)	116
5.10. TCP Idle Scan (-sI)	117
5.10.1. Idle Scan Step by Step	118
5.10.2. Finding a Working Idle Scan Zombie Host	120
5.10.3. Executing an Idle Scan	121
5.10.4. Idle Scan Implementation Algorithms	122
5.11. IP Protocol Scan (-sO)	125
5.12. TCP FTP Bounce Scan (-b)	127
5.13. Scan Code and Algorithms	128
5.13.1. Network Condition Monitoring	129
5.13.2. Host and Port Parallelization	129
5.13.3. Round Trip Time Estimation	130
5.13.4. Congestion Control	130
5.13.5. Timing probes	132
5.13.6. Inferred Neighbor Times	132
5.13.7. Adaptive Retransmission	132
5.13.8. Scan Delay	132
6. Optimizing Nmap Performance	135
6.1. Introduction	135



6.2. Scan Time Reduction Techniques	135
6.2.1. Omit Non-critical Tests	136
6.2.2. Optimize Timing Parameters	137
6.2.3. Separate and Optimize UDP Scans	137
6.2.4. Upgrade Nmap	137
6.2.5. Execute Concurrent Nmap Instances	138
6.2.6. Scan From a Favorable Network Location	138
6.2.7. Increase Available Bandwidth and CPU Time	138
6.3. Coping Strategies for Long Scans	139
6.3.1. Use a Multi-stage Approach	139
6.3.2. Estimate and Plan for Scan Time	140
6.4. Port Selection Data and Strategies	140
6.5. Low-Level Timing Controls	141
6.6. Timing Templates (-T)	142
6.7. Scanning 676,352 IP Addresses in 46 Hours	143
7. Service and Application Version Detection	145
7.1. Introduction	145
7.2. Usage and Examples	147
7.3. Technique Described	149
7.3.1. Cheats and Fallbacks	151
7.3.2. Probe Selection and Rarity	152
7.4. Technique Demonstrated	152
7.5. Post-processors	155
7.5.1. Nmap Scripting Engine Integration	155
7.5.2. RPC Grinding	156
7.5.3. SSL Post-processor Notes	157
7.6. nmap-service-probes File Format	158
7.6.1. Exclude Directive	158
7.6.2. Probe Directive	159
7.6.3. match Directive	159
7.6.4. softmatch Directive	161
7.6.5. ports and sslports Directives	162
7.6.6. totalwaitms Directive	162
7.6.7. rarity Directive	162
7.6.8. fallback Directive	163
7.6.9. Putting It All Together	163
7.7. Community Contributions	164
7.7.1. Submit Service Fingerprints	164
7.7.2. Submit Database Corrections	164
7.7.3. Submit New Probes	165
7.8. SOLUTION: Find All Servers Running an Insecure or Nonstandard Application Version	166
7.8.1. Problem	166
7.8.2. Solution	166
7.8.3. Discussion	167
7.9. SOLUTION: Hack Version Detection to Suit Custom Needs, such as Open Proxy Detection	168
7.9.1. Problem	168
7.9.2. Solution	169



7.9.3. Discussion	169
8. Remote OS Detection	171
8.1. Introduction	171
8.1.1. Reasons for OS Detection	171
Determining vulnerability of target hosts	171
Tailoring exploits	171
Network inventory and support	172
Detecting unauthorized and dangerous devices	172
Social engineering	172
8.2. Usage and Examples	172
8.3. TCP/IP Fingerprinting Methods Supported by Nmap	176
8.3.1. Probes Sent	177
Sequence generation (SEQ, OPS, WIN, and T1)	177
ICMP echo (IE)	178
TCP explicit congestion notification (ECN)	179
TCP (T2–T7)	179
UDP (U1)	179
8.3.2. Response Tests	180
TCP ISN greatest common divisor (GCD)	180
TCP ISN counter rate (ISR)	180
TCP ISN sequence predictability index (SP)	180
TCP IP ID sequence generation algorithm (TI)	181
ICMP IP ID sequence generation algorithm (II)	181
Shared IP ID sequence Boolean (SS)	182
TCP timestamp option algorithm (TS)	182
TCP options (O, 01–06)	183
TCP initial window size (W, W1–W6)	183
Responsiveness (R)	184
IP don't fragment bit (DF)	184
Don't fragment (ICMP) (DFI)	184
IP initial time-to-live (T)	184
IP initial time-to-live guess (TG)	185
Explicit congestion notification (CC)	185
TCP miscellaneous quirks (Q)	185
TCP sequence number (S)	186
ICMP sequence number(SI)	186
TCP acknowledgment number (A)	186
TCP flags (F)	187
TCP RST data checksum (RD)	187
IP type of service (TOS)	187
IP type of service for ICMP responses (TOSI)	187
IP total length (IPL)	188
Unused port unreachable field nonzero (UN)	188
Returned probe IP total length value (RIPL)	188
Returned probe IP ID value (RID)	188
Integrity of returned probe IP checksum value (RIPCK)	188
Integrity of returned probe UDP length and checksum (RUL and RUCK)	188
Integrity of returned UDP data (RUD)	188
ICMP response code (CD)	189



IP data length for ICMP responses (DLI)	189
8.4. Fingerprinting Methods Avoided by Nmap	189
8.4.1. Passive Fingerprinting	189
8.4.2. Exploit Chronology	190
8.4.3. Retransmission Times	190
8.4.4. IP Fragmentation	191
8.4.5. Open Port Patterns	191
8.5. Understanding an Nmap Fingerprint	191
8.5.1. Decoding the Subject Fingerprint Format	192
Decoding the SCAN line of a subject fingerprint	193
8.5.2. Decoding the Reference Fingerprint Format	194
Free-form OS description (Fingerprint line)	195
Device and OS classification (Class lines)	196
Test expressions	197
8.6. OS Matching Algorithms	198
8.7. Dealing with Misidentified and Unidentified Hosts	199
8.7.1. When Nmap Guesses Wrong	200
8.7.2. When Nmap Fails to Find a Match and Prints a Fingerprint	201
8.7.3. Modifying the nmap-os-db Database Yourself	202
8.8. SOLUTION: Detect Rogue Wireless Access Points on an Enterprise Network	202
8.8.1. Problem	202
8.8.2. Solution	202
8.8.3. WAP Characteristics	203
9. Nmap Scripting Engine	205
9.1. Introduction	205
9.2. Usage and Examples	206
9.2.1. Script Categories	207
9.2.2. Command-line Arguments	209
9.2.3. Arguments to Scripts	210
9.2.4. Usage Examples	210
9.3. Script Format	211
9.3.1. description Field	211
9.3.2. categories Field	211
9.3.3. author Field	211
9.3.4. license Field	211
9.3.5. runlevel Field	211
9.3.6. Port and Host Rules	212
9.3.7. Action	212
9.4. Script Language	212
9.4.1. Lua Base Language	212
9.5. NSE Scripts	213
9.6. NSE Libraries	236
9.6.1. List of All Libraries	236
9.6.2. Adding C Modules to Nselib	237
9.7. Nmap API	239
9.7.1. Information Passed to a Script	239
9.7.2. Network I/O API	241
Connect-style network I/O	241
Raw packet network I/O	242



9.7.3. Thread Mutexes	243
9.7.4. Exception Handling	244
9.7.5. The Registry	245
9.8. Script Writing Tutorial	245
9.8.1. The Head	245
9.8.2. The Rule	246
9.8.3. The Mechanism	247
9.9. Writing Script Documentation (NSEDoc)	248
9.9.1. NSE Documentation Tags	250
9.10. Version Detection Using NSE	251
9.11. Example Script: finger.nse	253
9.12. Implementation Details	254
9.12.1. Initialization Phase	254
9.12.2. Matching Scripts with Targets	255
9.12.3. Script Execution	255
10. Detecting and Subverting Firewalls and Intrusion Detection Systems	257
10.1. Introduction	257
10.2. Why Would Ethical Professionals (White-hats) Ever Do This?	257
10.3. Determining Firewall Rules	258
10.3.1. Standard SYN Scan	258
Sneaky firewalls that return RST	259
10.3.2. ACK Scan	260
10.3.3. IP ID Tricks	262
10.3.4. UDP Version Scanning	264
10.4. Bypassing Firewall Rules	265
10.4.1. Exotic Scan Flags	265
10.4.2. Source Port Manipulation	266
10.4.3. IPv6 Attacks	267
10.4.4. IP ID Idle Scanning	269
10.4.5. Multiple Ping Probes	269
10.4.6. Fragmentation	269
10.4.7. Proxies	270
10.4.8. MAC Address Spoofing	270
10.4.9. Source Routing	271
10.4.10. FTP Bounce Scan	272
10.4.11. Take an Alternative Path	272
10.4.12. A Practical Real-life Example of Firewall Subversion	272
10.5. Subverting Intrusion Detection Systems	276
10.5.1. Intrusion Detection System Detection	276
Reverse probes	276
Sudden firewall changes and suspicious packets	277
Naming conventions	277
Unexplained TTL jumps	278
10.5.2. Avoiding Intrusion Detection Systems	279
Slow down	280
Scatter probes across networks rather than scanning hosts consecutively	282
Fragment packets	282
Evade specific rules	283
Avoid easily detected Nmap features	284



10.5.3. Misleading Intrusion Detection Systems	284
Decoys	284
Port scan spoofing	286
Idle scan	286
DNS proxying	286
10.5.4. DoS Attacks Against Reactive Systems	287
10.5.5. Exploiting Intrusion Detection Systems	288
10.5.6. Ignoring Intrusion Detection Systems	288
10.6. Detecting Packet Forgery by Firewall and Intrusion Detection Systems	289
10.6.1. Look for TTL Consistency	289
10.6.2. Look for IP ID and Sequence Number Consistency	290
10.6.3. The Bogus TCP Checksum Trick	291
10.6.4. Round Trip Times	292
10.6.5. Close Analysis of Packet Headers and Contents	293
10.6.6. Unusual Network Uniformity	293
11. Defenses Against Nmap	295
11.1. Introduction	295
11.2. Scan Proactively, Then Close or Block Ports and Fix Vulnerabilities	295
11.3. Block and Slow Nmap with Firewalls	296
11.4. Detect Nmap Scans	297
11.5. Clever Trickery	298
11.5.1. Hiding Services on Obscure Ports	299
11.5.2. Port Knocking	300
11.5.3. Honeypots and Honeynets	301
11.5.4. OS Spoofing	302
11.5.5. Tar Pits	303
11.5.6. Reactive Port Scan Detection	304
11.5.7. Escalating Arms Race	304
12. Zenmap GUI Users' Guide	307
12.1. Introduction	307
12.1.1. The Purpose of a Graphical Frontend for Nmap	307
12.2. Scanning	308
12.2.1. Profiles	309
12.2.2. Scan Aggregation	309
12.3. Interpreting Scan Results	311
12.3.1. Scan Results Tabs	311
The Nmap Output tab	312
The Ports / Hosts tab	312
The Topology tab	313
The Host Details tab	314
The Scans tab	315
12.3.2. Sorting by Host	315
12.3.3. Sorting by Service	316
12.4. Saving and Loading Scan Results	316
12.4.1. The Recent Scans Database	317
12.5. Surfing the Network Topology	317
12.5.1. An Overview of the Topology Tab	318
12.5.2. Legend	318
12.5.3. Controls	319



Action controls	319
Interpolation controls	320
Layout controls	320
View controls	321
Fisheye controls	321
12.5.4. Keyboard Shortcuts	322
12.5.5. The Hosts Viewer	322
12.6. The Nmap Command Constructor Wizard	322
12.7. The Profile Editor	323
12.7.1. Creating a New Profile	324
12.7.2. Editing a Profile	324
12.7.3. Deriving a New Profile from an Old One	325
12.8. Searching Saved Results	325
12.9. Comparing Results	328
12.9.1. Graphical Comparison	329
12.9.2. Text Comparison	329
12.10. Files Used by Zenmap	330
12.10.1. The nmap Executable	330
12.10.2. System Configuration Files	331
12.10.3. Per-user Configuration Files	332
12.10.4. Output Files	332
12.11. Description of zenmap.conf	333
12.11.1. Sections of zenmap.conf	333
12.12. Command-line Options	335
12.12.1. Synopsis	335
12.12.2. Options Summary	335
12.12.3. Error Output	336
12.13. History	336
13. Nmap Output Formats	337
13.1. Introduction	337
13.2. Command-line Flags	338
13.2.1. Controlling Output Type	338
13.2.2. Controlling Verbosity of Output	339
13.2.3. Enabling Debugging Output	343
13.2.4. Handling Error and Warning Messages	344
13.2.5. Enabling Packet Tracing	345
13.2.6. Resuming Aborted Scans	346
13.3. Interactive Output	346
13.4. Normal Output (-oN)	346
13.5. \$rIpT kIddI3 OuTPut (-oS)	347
13.6. XML Output (-oX)	348
13.6.1. Using XML Output	350
13.7. Manipulating XML Output with Perl	352
13.8. Output to a Database	354
13.9. Creating HTML Reports	355
13.9.1. Saving a Permanent HTML Report	355
13.10. Grepable Output (-oG)	356
13.10.1. Grepable Output Fields	357
Host field	357



Ports field	357
Protocols field	359
Ignored State field	359
OS field	359
Seq Index field	360
IP ID Seq field	360
Status field	360
13.10.2. Parsing Grepable Output on the Command Line	361
14. Understanding and Customizing Nmap Data Files	363
14.1. Introduction	363
14.2. Well Known Port List: nmap-services	363
14.3. Version Scanning DB: nmap-service-probes	365
14.4. SunRPC Numbers: nmap-rpc	366
14.5. Nmap OS Detection DB: nmap-os-db	366
14.6. MAC Address Vendor Prefixes: nmap-mac-prefixes	368
14.7. IP Protocol Number List: nmap-protocols	369
14.8. Files Related to Scripting	369
14.9. Using Customized Data Files	370
15. Nmap Reference Guide	373
15.1. Description	373
15.2. Options Summary	374
15.3. Target Specification	376
15.4. Host Discovery	378
15.5. Port Scanning Basics	383
15.6. Port Scanning Techniques	384
15.7. Port Specification and Scan Order	389
15.8. Service and Version Detection	390
15.9. OS Detection	392
15.10. Nmap Scripting Engine (NSE)	393
15.11. Timing and Performance	394
15.12. Firewall/IDS Evasion and Spoofing	399
15.13. Output	403
15.14. Miscellaneous Options	408
15.15. Runtime Interaction	410
15.16. Examples	410
15.17. Bugs	411
15.18. Author	411
15.19. Legal Notices	412
15.19.1. Nmap Copyright and Licensing	412
15.19.2. Creative Commons License for this Nmap Guide	413
15.19.3. Source Code Availability and Community Contributions	413
15.19.4. No Warranty	413
15.19.5. Inappropriate Usage	414
15.19.6. Third-Party Software	414
15.19.7. United States Export Control Classification	414
A. Nmap XML Output DTD	415
A.1. Purpose	415
A.2. The Full DTD	415
Index	423



List of Figures

1. IPv4 header	xxvii
2. TCP header	xxviii
3. UDP header	xxviii
4. ICMP header	xxix
1.1. Trinity begins her assault	8
1.2. Trinity scans the Matrix	9
1.3. Strong opinions on port scanning legality and morality	14
2.1. Executing Nmap from a Windows command shell	40
3.1. A business card explains everything	50
3.2. Netcraft finds 36 Target web servers	54
5.1. ICMPv4 destination unreachable header layout	96
5.2. SYN scan of open port 22	97
5.3. SYN scan of closed port 113	98
5.4. SYN scan of filtered port 139	98
5.5. Connect scan of open port 22 (nmap -sT -p22 scanme.nmap.org)	100
5.6. Idle scan of an open port	119
5.7. Idle scan of a closed port	119
5.8. Idle scan of a filtered port	119
5.9. Congestion window and threshold	131
5.10. Scan rate as affected by scan delay	133
8.1. ICMP echo request or reply header layout	177
8.2. ICMP destination unreachable header layout	177
10.1. BlackICE discovers an unusual intruder	277
10.2. An attacker masked by dozens of decoys	285
12.1. Typical Zenmap screen shot	307
12.2. Zenmap's main window	308
12.3. Target and profile selection	309
12.4. Host selection	315
12.5. OS icons	316
12.6. Service selection	316
12.7. Grouping a host's children	319
12.8. Highlighting regions of the topology	320
12.9. Choosing a profile	323
12.10. The profile editor	324
12.11. The search dialog	325
12.12. Keyword search	326
12.13. Expressions search	326
12.14. Comparison tool	328
12.15. Graphical comparison	329
12.16. Text mode comparison	330
13.1. XML output in a web browser	351



List of Tables

1. Formatting style conventions	xxiii
3.1. First pass at listing target.com IPs	51
3.2. Most valuable TCP probe ports, in descending order of accessibility.	68
5.1. ICMP destination unreachable (type 3) code values	96
5.2. How Nmap interprets responses to a SYN probe	98
5.3. How Nmap interprets responses to a UDP probe	101
5.4. How Nmap interprets responses to a NULL, FIN, or Xmas scan probe	107
5.5. How Nmap interprets responses to an ACK scan probe	113
5.6. How Nmap interprets responses to a Window scan ACK probe	115
5.7. How Nmap interprets responses to a Maimon scan probe	117
5.8. How Nmap interprets responses to an IP protocol probe	126
6.1. Required --top-ports values for reaching various effectiveness levels	141
6.2. Low-level timing controls by function	142
6.3. Timing templates and their effects	142
7.1. versioninfo field formats and values	160
8.1. O test values	183
8.2. DFI test values	184
8.3. CC test values	185
8.4. S test values	186
8.5. SI test values	186
8.6. A test values	186
8.7. F test values	187
8.8. TOSI test values	187
8.9. CD test values	189
8.10. DLI test values	189
8.11. Reference fingerprint test expression operators	198
9.1. port.version values	240
12.1. Text diff character codes	330



List of Examples

1. A typical Nmap scan	xxiii
1.1. Nmap list scan against Avatar Online IP addresses	3
1.2. Nmap results against an AO firewall	5
1.3. Another interesting AO machine	7
1.4. nmap-diff typical output	11
1.5. nmap-report execution	12
2.1. Checking for Nmap and determining its version number	25
2.2. Verifying the Nmap and Fyodor PGP Key Fingerprints	27
2.3. Verifying PGP key fingerprints (Successful)	27
2.4. Detecting a bogus file	27
2.5. A typical Nmap release digest file	28
2.6. Verifying Nmap hashes	28
2.7. Successful configuration screen	30
2.8. Installing Nmap from binary RPMs	33
2.9. Building and installing Nmap from source RPMs	34
2.10. Installing Nmap from a system Yum repository	35
3.1. Using the host command to query common DNS record types	51
3.2. Zone transfer failure and success	52
3.3. Nmap reverse-DNS and traceroute scan against www.target.com	53
3.4. Using whois to find owner of www.target.com IP address	53
3.5. Using whois to find netblock containing 161.225.130.163	55
3.6. Enumerating hosts surrounding www.stanford.edu with list scan	58
3.7. Discovering hosts surrounding www.lwn.net with a ping scan	59
3.8. Attempts to ping popular Internet hosts	61
3.9. Retry host discovery using port 80 SYN probes	62
3.10. Attempted ACK ping against Microsoft	63
3.11. Raw IP ping scan of an offline target	65
3.12. ARP ping scan of an offline target	65
3.13. Generating 50,000 IP addresses, then ping scanning with default options	71
3.14. Repeating ping scan with extra probes	71
4.1. Viewing and increasing the ephemeral port range on Linux	74
4.2. Simple scan: nmap scanme.nmap.org	80
4.3. More complex: nmap -p0- -v -A -T4 scanme.nmap.org	81
4.4. A simple IPv6 scan	88
4.5. Discovering Playboy's IP space	90
4.6. Pinging Playboy's web server for a latency estimate	90
4.7. Digging through Playboy's DNS records	91
4.8. Pinging the MX servers	92
4.9. TCP pinging the MX servers	92
4.10. Launching the scan	93
4.11. Egrep for open ports	94
5.1. A SYN scan showing three port states	97
5.2. Using --packet-trace to understand a SYN scan	99
5.3. Connect scan example	101
5.4. UDP scan example	102
5.5. UDP scan example	102



5.6. Improving Felix's UDP scan results with version detection	103
5.7. Improving Scanme's UDP scan results with version detection	104
5.8. Attempting to disambiguate UDP ports with TTL discrepancies	105
5.9. Optimizing UDP Scan Time	107
5.10. Example FIN and Xmas scans	109
5.11. SYN scan of Docsrv	109
5.12. FIN scan of Docsrv	110
5.13. A SYN/FIN scan of Google	112
5.14. A custom PSH scan	112
5.15. A typical ACK Scan	114
5.16. An ACK scan of Docsrv	115
5.17. Window scan of docsrv.caldera.com	116
5.18. A failed Maimon scan	117
5.19. An idle scan against the RIAA	121
5.20. IP protocol scan of a router and a typical Linux 2.4 box	127
5.21. Attempting an FTP bounce scan	128
5.22. Successful FTP bounce scan	128
6.1. Bandwidth usage over local 100 Mbps ethernet network	139
6.2. Estimating scan time	140
7.1. Simple usage of version detection	146
7.2. Version detection against www.microsoft.com	147
7.3. Complex version detection	148
7.4. NULL probe cheat example output	151
7.5. Enumerating RPC services with rpcinfo	156
7.6. Nmap direct RPC scan	157
7.7. Version scanning through SSL	158
8.1. OS detection with verbosity (-O -v)	173
8.2. Using version scan to detect the OS	175
8.3. A typical subject fingerprint	192
8.4. A cleaned-up subject fingerprint	193
8.5. A typical reference fingerprint	195
8.6. Some typical fingerprint descriptions and corresponding classifications	197
8.7. The MatchPoints structure	199
8.8. Scan results against a consumer WAP	203
9.1. Typical NSE output	206
9.2. Connect-style I/O	242
9.3. Mutex manipulation	244
9.4. Exception handling example	244
9.5. An NSEDoc comment for a function	249
9.6. An NSEDoc comment for a module	249
9.7. An NSEDoc comment for a script	250
9.8. A typical version detection script (Skype version 2 detection)	252
10.1. Detection of closed and filtered TCP ports	259
10.2. ACK scan against Scanme	260
10.3. Contrasting SYN and ACK scans against Para	261
10.4. UDP scan against firewalled host	264
10.5. UDP version scan against firewalled host	265
10.6. FIN scan against stateless firewall	266
10.7. Bypassing Windows IPsec filter using source port 88	267



10.8. Comparing IPv4 and IPv6 scans	268
10.9. Exploiting a printer with the FTP bounce scan	272
10.10. Some interesting hosts and networks at Megacorp	273
10.11. Ping scan against the target network	273
10.12. Packet trace against a single IP	273
10.13. Testing an idle scan	274
10.14. Testing source routing	275
10.15. Success at last	275
10.16. Host names can be deceiving	278
10.17. Noting TTL gaps with traceroute	279
10.18. Using the IP record route option	279
10.19. Slow scan to bypass the default Snort 2.2.0 Flow-portscan fixed time scan detection method	281
10.20. Default Snort rules referencing Nmap	283
10.21. Using DNS Proxies (Recursive DNS) for a Stealth List Scan of SecurityFocus	287
10.22. Detection of closed and filtered TCP ports	290
10.23. Testing IP ID sequence number consistency	291
10.24. Finding a firewall with bad TCP checksums	291
11.1. An all-TCP-port version scan	299
11.2. Deceiving Nmap with IP Personality	303
13.1. Scanrand output against a local network	337
13.2. Grepping for verbosity conditionals	341
13.3. Interactive output without verbosity enabled	342
13.4. Interactive output with verbosity enabled	343
13.5. Some representative debugging lines	344
13.6. Using --packet-trace to detail a ping scan of Scanme	345
13.7. A typical example of normal output	347
13.8. A typical example of \$rIpT KiDDi3 OutPut	348
13.9. An example of Nmap XML output	349
13.10. Nmap XML port elements	350
13.11. Nmap::Parser sample code	353
13.12. Nmap::Scanner sample code	354
13.13. A typical example of grepable output	357
13.14. Grepable output for IP protocol scan	359
13.15. Ping scan grepable output	361
13.16. List scan grepable output	361
13.17. Parsing grepable output on the command line	361
14.1. Excerpt from nmap-services	364
14.2. Excerpt from nmap-service-probes	365
14.3. Excerpt from nmap-rpc	366
14.4. Excerpt from nmap-os-db	367
14.5. Excerpt from nmap-mac-prefixes	368
14.6. Excerpt from nmap-protocols	369
15.1. A representative Nmap scan	374

