

Table of Contents

Preface	xxiii
1. Introduction	xxiii
2. Intended Audience and Organization	xxiv
3. Conventions	xxiv
4. Other Resources	xxvi
5. Request for Comments	xxvi
6. Acknowledgements	xxvii
1. Getting Started with Nmap	1
1.1. Introduction	1
1.2. Nmap Overview and Demonstration	1
Avatar Online	1
Saving the Human Race	8
MadHat in Wonderland	10
1.3. Legal Issues	13
Is Unauthorized Port Scanning a Crime?	13
Can Port Scanning Crash the Target Computer/Networks?	18
Nmap Copyright	19
1.4. The History and Future of Nmap	20
2. Obtaining, Compiling, Installing, and Removing Nmap	25
2.1. Introduction	25
Testing Whether Nmap is Already Installed	25
Command-line and Graphical Interfaces	25
Downloading Nmap	28
Verifying the Integrity of Nmap Downloads	28
Obtaining Nmap from the Subversion (SVN) Repository	30
2.2. Unix Compilation and Installation from Source Code	31
Configure Directives	32
If You Encounter Compilation Problems	33
2.3. Linux Distributions	34
RPM-based Distributions (Red Hat, Mandrake, Suse, Fedora)	34
Updating Red Hat, Fedora, Mandrake, and Yellow Dog Linux with Yum	36
Debian Linux and Derivatives such as Ubuntu	37
Gentoo Linux	38
Other Linux Distributions	38
2.4. Windows	38
Windows Self-installer	39
Command-line Zip Binaries	39
Installing the Nmap zip binaries	40
Compile from Source Code	40
Executing Nmap on Windows	41
2.5. Sun Solaris	42
2.6. Apple Mac OS X	43
2.7. FreeBSD / OpenBSD / NetBSD	43



OpenBSD Binary Packages and Source Ports Instructions	44
FreeBSD Binary Package and Source Ports Instructions	44
Installation of the binary package	44
Installation using the source ports tree	45
NetBSD Binary Package Instructions	45
2.8. Amiga, HP-UX, IRIX, and Other Platforms	45
2.9. Removing Nmap	45
3. Host Discovery (“Ping Scanning”)	47
3.1. Introduction	47
3.2. Specifying Target Hosts and Networks	47
Input From List (-iL)	48
Choose Targets at Random (-iR <i>Numtargets</i>)	48
Excluding Targets (--exclude, --excludefile <i>filename</i>)	48
Practical Examples	49
3.3. DNS Resolution	49
Speeding DNS Resolution by Carefully Selecting DNS Servers	50
3.4. Host Discovery Controls	51
List Scan (-sL)	51
Ping Scan (-sP)	52
Disable Ping (-P0)	53
3.5. Host Discovery Techniques	54
TCP SYN Ping (-PS[portlist])	55
TCP ACK Ping (-PA[portlist])	56
UDP Ping (-PU[portlist])	57
ICMP Ping Types (-PE, -PP, and -PM)	57
ARP Scan (-PR)	58
Default Combination (-PB)	59
3.6. Putting it All Together: Host Discovery Strategies	60
Related Options	60
Choosing and Combining Ping Options	62
TCP probe and port selection	62
UDP port selection	64
ICMP probe selection	64
Designing the ideal combinations of probes	64
3.7. Finding an Organization's IP addresses to Scan	66
3.8. Host Discovery Code Algorithms	66
4. Port Scanning Overview	67
4.1. Introduction to Port Scanning	67
What Exactly is a Port?	67
What is Port Scanning?	70
Why Scan Ports?	71
4.2. A Quick Port Scanning Tutorial	72
4.3. Command-line Flags	76
Selecting Scan Techniques	76
Selecting Ports to Scan	77
Timing-related Options	78



Output Format and Verbosity Options	79
Firewall and IDS Evasion Options	80
Specifying Targets	81
Miscellaneous Options	81
4.4. IPv6 Scanning [-6]	81
4.5. [RECIPE] Scanning a Large Network for a Certain Open TCP Port	82
Problem	82
Solution	82
Discussion	83
See Also	89
5. Port Scanning Techniques and Algorithms	91
5.1. Introduction	91
5.2. TCP SYN (Stealth) Scan (-sS)	93
5.3. TCP Connect Scan (-sT)	96
5.4. UDP Scan (-sU)	98
Disambiguating Open from Filtered UDP Ports	99
Speeding up UDP Scans	102
5.5. TCP FIN, NULL, and Xmas Scans (-sF, -sN, -sX)	103
5.6. Custom Scan Types with --scanflags	107
Custom SYN/FIN Scan	107
PSH Scan	108
5.7. TCP ACK Scan (-sA)	109
5.8. TCP Window Scan (-sW)	110
5.9. TCP Maimon Scan (-sM)	113
5.10. TCP Idle Scan (-sI)	113
Finding a Working Idle Scan Zombie Host	116
Executing an Idle Scan	116
Idle Scan Implementation Algorithms	117
5.11. IP Protocol Scan (-sO)	121
5.12. TCP FTP Bounce Scan (-b)	123
5.13. Scan Code and Algorithms	125
Network Condition Monitoring	125
Host and Port Parallelization	125
Round Trip Time Estimation	126
Congestion Control	126
Port Scan Pings	128
Inferred Neighbor Times	128
Adaptive Retransmission	128
Scan Delay	129
6. Optimizing Nmap Performance	131
6.1. Introduction	131
6.2. Scan Time Reduction Techniques	131
Omit Non-Critical Tests	132
Optimize Timing Parameters	133
Upgrade Nmap	133
Execute Concurrent Nmap Instances	133



Scan From a Favorable Network Location	134
Increase Available Bandwidth and CPU Time	134
6.3. Coping Strategies for Long Scans	135
Use a Multi-stage Approach	135
Estimate and plan for scan time	135
6.4. Low Level Timing Controls	136
Hostgroup Size (--min-hostgroup, --max-hostgroup)	136
Probe Parallelism (--min-parallelism, --max-parallelism)	137
Round Trip Time Timeouts (--min-rtt-timeout, --max-rtt-timeout, --initial-rtt-timeout)	137
Maximum Number of Retransmissions (--max-retries)	138
Host Timeout (--host-timeout)	138
Scan delay (--scan-delay, --max-scan-delay)	138
6.5. Timing Templates (-T)	139
6.6. Scanning 676,352 IP Addresses in 46 Hours	140
6.7. Scan Performance Examples	142
7. Service and Application Version Detection	143
7.1. Introduction	143
7.2. Usage and Examples	145
7.3. Technique Described	147
Cheats and Fallbacks	148
Probe Selection and Rarity	149
--version-intensity	150
--version-light	150
--version-all	150
7.4. Technique Demonstrated	151
7.5. Post-processors	153
Nmap Scripting Engine Integration	153
RPC Grinding	154
SSL Post-processor Notes	155
7.6. nmap-service-probes File Format	156
Exclude Directive	156
Probe Directive	157
match Directive	157
softmatch Directive	159
ports and sslports Directives	160
totalwaitms Directive	160
rarity Directive	160
fallback Directive	161
Putting it all together	161
7.7. Community Contributions	162
Submit Service Fingerprints	162
Submit Database Corrections	162
Submit New Probes	163
7.8. [RECIPE] Find All Servers Running an Insecure or Nonstandard Application Version	164
Problem	164



Solution	164
Discussion	165
7.9. [RECIPE] Hack Version Detection to Suit Custom Needs, such as Open Proxy Detection	166
Problem	166
Solution	167
Discussion	167
8. Remote OS Detection	169
8.1. Introduction	169
Reasons for OS Detection	169
Determining vulnerability of target hosts	169
Tailoring exploits	169
Network inventory and support	170
Detecting unauthorized and dangerous devices	170
Social engineering	170
8.2. Usage and Examples	170
8.3. TCP/IP Fingerprinting Methods Supported by Nmap	175
Probes Sent	178
Sequence generation (SEQ, OPS, WIN, and T1)	178
ICMP echo (IE)	179
TCP explicit congestion notification (ECN)	180
TCP (T2–T7)	180
UDP (U1)	180
Response Tests	181
TCP ISN greatest common denominator (GCD)	181
TCP ISN counter rate (ISR)	181
TCP ISN sequence predictability index (SP)	181
TCP IP ID sequence generation algorithm (TI)	182
ICMP IP ID sequence generation algorithm (II)	182
Shared IP ID sequence boolean (SS)	183
TCP timestamp option algorithm (TS)	183
TCP options (O, 01–06)	184
TCP initial window size (W, W1–W6)	184
Responsiveness (R)	185
IP don't fragment bit (DF)	185
Don't fragment (ICMP) (DFI)	185
IP initial time-to-live (T)	185
IP initial time-to-live guess (TG)	186
Explicit congestion notification (CC)	186
TCP miscellaneous quirks (Q)	186
TCP sequence number (S)	187
ICMP sequence number(SI)	187
TCP acknowledgment number (A)	187
TCP flags (F)	188
TCP RST data checksum (RD)	188
IP type of service (TOS)	188



IP type of service for ICMP responses (TOSI)	189
IP total length (IPL)	189
Unused port unreachable field nonzero (UN)	189
Returned probe IP total length value (RIPL)	189
Returned probe IP ID value (RID)	189
Integrity of returned probe IP checksum value (RIPCK)	190
Integrity of returned probe UDP length and checksum (RUL and RUCK)	190
Integrity of returned UDP data (RUD)	190
ICMP response code (CD)	190
IP data length for ICMP responses (DLI)	190
8.4. Fingerprinting Methods Avoided by Nmap	191
Passive Fingerprinting	191
Exploit Chronology	191
Retransmission Times	192
IP Fragmentation	192
8.5. Understanding an Nmap Fingerprint	192
Decoding the Subject Fingerprint Format	193
Decoding the SCAN line of a subject fingerprint	194
Decoding the Reference Fingerprint Format	195
Free-form OS description (Fingerprint line)	196
Device and OS classification (Class lines)	197
Test expressions	198
8.6. OS Matching Algorithms	199
8.7. Dealing with Misidentified and Unidentified Hosts	200
When Nmap Guesses Wrong	201
When Nmap Fails to Find a Match and Prints a Fingerprint	202
Modifying the nmap-os-db database Yourself	203
8.8. [RECIPE] Detect Rogue Wireless Access Points on an Enterprise Network	203
Problem	203
Solution	204
WAP Characteristics	204
9. Nmap Scripting Engine	207
9.1. Introduction	207
9.2. Usage and Examples	209
Script Categories	209
Arguments to Scripts	210
Command-line Arguments	210
Usage Examples	211
9.3. Script Format	212
id Field	212
description Field	212
author Field	212
license Field	212
runlevel Field	212
Port and Host Rules	212
Action	213



9.4. Script Language	213
Lua Base Language	213
9.5. Lua Extensions	214
Bitwise Logical Operations	214
Perl Compatible Regular Expressions	215
IP Operations	216
Short Portrules	217
Functional Programming Style List Operations	217
String Buffer Operations	218
URL Manipulation Functions	219
Buffered Network I/O Helper Functions	219
Various Utility Functions	220
9.6. Nmap API	220
Information Passed to a Script	220
Target Information Retrieving by a Script	223
Various Utility Functions for Raw Packet Support	224
Network I/O API	224
Connect-style network I/O	224
Raw packet network I/O	227
Exception Handling	229
The Registry	230
9.7. Script Writing Tutorial	230
The Head	230
The Rule	231
The Mechanism	232
9.8. Version Detection using NSE	233
9.9. Example Scripts	235
Finger-Test Script	235
Service Owner Lookup via Identd	237
9.10. Implementation	238
Initialization Phase	239
Matching of Scripts to Targets	240
Running Scripts	241
Adding C Modules to Nselib	241
9.11. NSE Script License and Community Contributions	242
10. Detecting and Subverting Firewalls and Intrusion Detection Systems	243
10.1. Introduction	243
10.2. Why Would Whitehats Ever Do This?	243
10.3. Determining Firewall Rules	244
Standard SYN Scan	244
Sneaky firewalls that return RST	245
ACK Scan	246
IP ID Tricks	248
UDP Version Scanning	250
10.4. Bypassing Firewall Rules	252
Exotic Scan Flags	252



Source Port Manipulation	253
IPv6 Attacks	254
IP ID Idle Scanning	256
Multiple Ping Probes	256
Fragmentation	256
Proxies	257
Source Routing	257
FTP Bounce Scan	257
Take an Alternative Path	258
10.5. Subverting Intrusion Detection Systems	258
Intrusion Detection System Detection	259
Reverse probes	259
Sudden firewall changes and suspicious packets	260
Naming conventions	260
Unexplained TTL jumps	261
Avoiding Intrusion Detection Systems	261
Slow down	261
Scatter probes across networks rather than scanning hosts consecutively.	264
Fragment packets	264
Evade specific rules	265
Avoid easily detected Nmap features	266
Misleading Intrusion Detection Systems	266
Decoys	266
Port scan spoofing	268
MAC address spoofing	268
Idle scan	268
DNS proxying	268
DoS Attacks Against Reactive Systems	269
Exploiting Intrusion Detection Systems	269
Ignoring Intrusion Detection Systems	269
10.6. Detecting Packet Forgery by Firewall and Intrusion Detection Systems	270
Look for TTL Consistency	271
Look for IP ID and Sequence Number Consistency	272
The Bogus TCP Checksum trick	273
Round Trip Times	273
Close Analysis of Packet Headers and Contents	274
Unusual Network Uniformity	274
11. Defenses Against Nmap	275
11.1. Introduction	275
11.2. Proactive Scanning	275
11.3. Blocking and Slowing Nmap with Firewalls	276
11.4. Detecting Nmap Scans	277
11.5. Clever Trickery	278
Hiding Services on Obscure Ports	278
Port knocking	280
Honeypots and Honeynets	281



OS Spoofing	281
Tar Pits	283
Reactive Port Scan Detection	284
Escalating Arms Race	284
12. Nmap Output Formats	285
12.1. Introduction	285
12.2. Command-line Flags	286
Controlling Output Type	286
Controlling Verbosity of Output	287
Enabling Debugging Output	291
Enabling Packet Tracing	292
Resuming Aborted Scans	293
12.3. Interactive Output	293
12.4. Normal Output (-oN)	293
12.5. \$crIpT kIddI3 OuTPut (-oS)	294
12.6. XML Output (-oX)	295
Using XML Output	297
12.7. Manipulating XML Output with Perl	300
12.8. Output to a Database	302
12.9. Creating HTML Reports	303
12.10. Grepable Output (-oG)	303
Grepable Output Fields	304
Host field	304
Ports field	305
Protocols field	306
Ignored State field	306
OS field	307
Seq Index field	307
IP ID Seq field	307
Status field	307
Parsing Grepable Output on the Command Line	308
13. Understanding and Customizing Nmap Data Files	309
13.1. Introduction	309
13.2. Well Known Port List: nmap-services	309
13.3. Version Scanning DB: nmap-service-probes	311
13.4. SunRPC Numbers: nmap-rpc	312
13.5. Old Nmap OS Detection DB: nmap-os-fingerprints	313
13.6. Nmap OS Detection DB: nmap-os-db	314
13.7. MAC Address Vendor Prefixes: nmap-mac-prefixes	314
13.8. IP Protocol Number List: nmap-protocols	315
13.9. Using Customized Data Files	315
14. Nmap Reference Guide	319
14.1. Description	319
14.2. Options Summary	320
14.3. Target Specification	323
14.4. Host Discovery	324



14.5. Port Scanning Basics	330
14.6. Port Scanning Techniques	331
14.7. Port Specification and Scan Order	336
14.8. Service and Version Detection	337
14.9. OS Detection	339
14.10. Nmap Scripting Engine (NSE)	340
14.11. Timing and Performance	342
14.12. Firewall/IDS Evasion and Spoofing	346
14.13. Output	350
14.14. Miscellaneous Options	355
14.15. Runtime Interaction	357
14.16. Examples	357
14.17. Bugs	358
14.18. Author	358
14.19. Legal Notices	359
Nmap Copyright and Licensing	359
Creative Commons License for this Nmap Guide	360
Source Code Availability and Community Contributions	360
No Warranty	360
Inappropriate Usage	361
Third-Party Software	361
US Export Control Classification	361
A. Nmap XML Output DTD	363
A.1. Purpose	363
A.2. The Full DTD	363
Index	371



List of Figures

1.1. Trinity begins her assault	8
1.2. Trinity scans the Matrix	9
1.3. Terminal-view of the hack	10
1.4. Strong opinions on port scanning legality and morality	14
2.1. NmapFE presents a simple graphical interface to Nmap	26
2.2. Zenmap is the newest and most advanced Nmap GUI	27
2.3. Executing Nmap from a Windows command shell	42
4.1. IPv4 header layout	67
4.2. TCP header layout	68
4.3. UDP header layout	68
5.1. ICMPv4 destination unreachable header layout	92
5.2. SYN scan of open port 22	93
5.3. SYN scan of closed port 113	94
5.4. SYN scan of filtered port 139	95
5.5. Connect scan of open port 22 (nmap -sT -p22 scanme.nmap.org)	97
5.6. Idle scan technique (simplified)	114
5.7. Congestion window and threshold	127
5.8. Scan delay during a UDP scan	129
8.1. IPv4 header layout	176
8.2. TCP header layout	177
8.3. ICMP echo request or reply header layout	177
8.4. ICMP destination unreachable header layout	178
8.5. UDP header layout	178
10.1. BlackICE discovers an unusual intruder	260
10.2. An attacker masked by dozens of decoys	267
12.1. Reading XML in a web browser	299



List of Tables

1. Formatting style conventions	xxvi
3.1. Valuable TCP probe ports, in descending order of accessibility	63
5.1. ICMP destination unreachable (type 3) code values	92
5.2. How Nmap interprets responses to a SYN probe	95
5.3. How Nmap interprets responses to a UDP probe	98
5.4. How Nmap interprets responses to a null, FIN, or Xmas scan probe	103
5.5. How Nmap interprets responses to an ACK scan probe	109
5.6. How Nmap interprets responses to a Window scan ACK probe	111
5.7. How Nmap interprets responses to a Maimon scan probe	113
5.8. How Nmap interprets responses to an IP protocol probe	122
6.1. Timing templates and their effects	140
7.1. <code>versioninfo</code> field formats and values	159
8.1. O test values	184
8.2. DFI test values	185
8.3. CC test values	186
8.4. S test values	187
8.5. SI test values	187
8.6. Ack test values	188
8.7. F test values	188
8.8. TOSI test values	189
8.9. CD test values	190
8.10. DLI test values	191
8.11. Reference fingerprint test expression operators	199
9.1. <code>port.version</code> values	222



List of Examples

1. A typical Nmap scan	xxv
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	12
1.5. Nmap-report execution	13
2.1. Checking for Nmap and determining its version number	25
2.2. Verifying the Nmap and Fyodor PGP Key Fingerprints	29
2.3. Verifying PGP Key Fingerprints (Successful)	29
2.4. Detecting a bogus file	29
2.5. A typical Nmap release digest file	30
2.6. Verifying Nmap hashes	30
2.7. Installing Nmap from binary RPMs	35
2.8. Building and installing Nmap from source RPMs	35
2.9. Installing Nmap from a system Yum repository	37
3.1. Enumerating hosts surrounding WWW.Stanford.Edu with list scan	52
3.2. Attempts to ping popular Internet hosts	54
3.3. Retry Host Discovery using port 80 SYN probes	55
3.4. Attempted ACK ping against Microsoft	57
3.5. Raw IP ping scan of an offline target	58
3.6. ARP ping scan of an offline target	59
3.7. Generating 50,000 IP Addresses, then ping scanning with default options	65
3.8. Repeating ping scan with extra probes	65
4.1. Viewing and increasing the ephemeral port range on Linux	69
4.2. Simple scan: <code>nmap scanme.nmap.org</code>	73
4.3. More complex: <code>nmap -p0- -v -A -T4 scanme.nmap.org</code>	75
4.4. A simple IPv6 scan	82
4.5. Discovering Playboy's IP space	84
4.6. Pinging Playboy's Web Server for a Latency Estimate	84
4.7. Digging through Playboy's DNS records	85
4.8. Pinging the MX servers	86
4.9. TCP Pinging the MX servers	86
4.10. Launching the scan	88
4.11. Egrep for open ports	88
5.1. A SYN Scan showing three port states	93
5.2. Using <code>--packet-trace</code> to understand a SYN scan	96
5.3. Connect scan example	98
5.4. UDP scan example	99
5.5. UDP scan example	99
5.6. Improving Felix's UDP scan results with version detection	100
5.7. Improving Scanme's UDP scan results with version detection	100
5.8. Attempting to disambiguate UDP ports with TTL discrepancies	101
5.9. Example FIN and Xmas scans	105



5.10. SYN scan of docsrv.caldera.com	105
5.11. FIN scan of docsrv.caldera.com	106
5.12. A SYN/FIN scan of Google	108
5.13. A custom PSH scan	108
5.14. A Typical ACK Scan	109
5.15. An ACK scan of Docsrv	110
5.16. Window scan of docsrv.caldera.com	112
5.17. A failed Maimon scan	113
5.18. An idle scan against the RIAA	117
5.19. IP protocol scan of a router and a typical Linux 2.4 box	123
5.20. Attempting an FTP bounce scan	124
5.21. Successful FTP bounce scan	124
6.1. Bandwidth usage over local 100Mbps ethernet network	134
6.2. Estimating scan time	135
7.1. Simple usage of version detection	144
7.2. Version detection against www.microsoft.com	145
7.3. Complex version detection	146
7.4. NULL Probe Cheat Example Output	149
7.5. Enumerating RPC services with rpcinfo	154
7.6. Nmap direct RPC scan	155
7.7. Version scanning through SSL	156
8.1. OS Detection with Verbosity (-O -v)	171
8.2. Using version scan to detect the OS	173
8.3. A typical subject fingerprint	193
8.4. A cleaned up subject fingerprint	194
8.5. A typical reference fingerprint	196
8.6. Some typical fingerprint descriptions and corresponding classifications	198
8.7. The MatchPoints structure	200
8.8. Scan results against a consumer WAP	205
9.1. Typical NSE Output	209
9.2. Exception handling example	229
9.3. Using local variables to save data.	240
10.1. Detection of closed and filtered TCP ports	245
10.2. ACK scan against Scanme	246
10.3. Contrasting SYN and ACK scans against Para	248
10.4. UDP scan against firewalled host	251
10.5. UDP version scan against firewalled host	251
10.6. FIN scan against stateless firewall	252
10.7. Bypassing Windows IPsec filter using source port 88	254
10.8. Comparing IPv4 and IPv6 scans	255
10.9. Exploiting a printer with the FTP bounce scan	258
10.10. Host names can be deceiving	261
10.11. Noting TTL gaps with traceroute	261
10.12. Slow scan to bypass the default Snort 2.2.0 Flow-portscan fixed time scan detection method.	263
10.13. Default Snort rules referencing Nmap	265
10.14. Detection of closed and filtered TCP ports	272



10.15. Testing IP ID sequence number consistency	273
11.1. An all-tcp-port version scan	279
11.2. Deceiving Nmap with IP Personality	282
12.1. Scanrand output against a local network	285
12.2. Greping for verbosity conditionals	289
12.3. A comparison of interactive output with and without verbosity enabled.	290
12.4. Some representative debugging lines	291
12.5. Using <code>--packet-trace</code> to detail a ping scan of Scanme	292
12.6. A typical example of normal output	294
12.7. A typical example of <code>\$crIpt KiDDi3 OutPut</code>	295
12.8. An example of Nmap XML output	296
12.9. Nmap XML port elements	297
12.10. <code>Nmap::Parser</code> sample code	301
12.11. <code>Nmap::Scanner</code> sample code	302
12.12. A typical example of grepable output	304
12.13. Grepable output for IP protocol scan	306
12.14. Ping scan grepable output	308
12.15. List scan grepable output	308
12.16. Parsing grepable output on the command line	308
13.1. Excerpt from <code>nmap-services</code>	310
13.2. Excerpt from <code>nmap-service-probes</code>	311
13.3. Excerpt from <code>nmap-rpc</code>	312
13.4. Excerpt from <code>nmap-os-fingerprints</code>	313
13.5. Excerpt from <code>nmap-mac-prefixes</code>	314
13.6. Excerpt from <code>nmap-protocols</code>	315
14.1. A representative Nmap scan	320

