NAME

nmap − Netzwerk−Analysewerkzeug und Sicherheits−/Portscanner

SYNOPSIS

nmap [*Scan Type...*] [*Options*] {*target specification*}

BESCHREIBUNG

Nmap (“Network Mapper”) ist ein Open−Source−Werkzeug für die Netzwerkanalyse und Sicherheitsüberprüfung. Es wurde entworfen, um große Netzwerke schnell zu scannen, auch wenn es bei einzelnen Hosts auch gut funktioniert. Nmap benutzt rohe IP−Pakete auf neuartige Weise, um festzustellen, welche Hosts im Netzwerk verfügbar sind, welche Dienste (Anwendungsname und −version) diese Hosts bieten, welche Betriebssysteme (und Versionen davon) darauf laufen, welche Art von Paketfiltern/−Firewalls benutzt werden sowie Dutzende anderer Eigenschaften. Auch wenn Nmap üblicherweise für Sicherheitsüberprüfungen verwendet wird, wird es von vielen Systemen und Netzwerkadministratoren für Routineaufgaben benutzt, z.B. Netzwerkinventarisierung, Verwaltung von Ablaufplänen für Dienstaktualisierungen und die Überwachung von Betriebszeiten von Hosts oder Diensten.

Die Ausgabe von Nmap ist eine Liste gescannter Ziele mit zusätzlicher Information zu jedem, abhängig von den benutzten Optionen. Die entscheidende Information dabei steht in der “Tabelle der interessanten Ports”.. Diese Tabelle listet die Portnummer und das −protokoll sowie den Dienstnamen und −zustand auf. Der Zustand ist entweder offen, gefiltert, geschlossen oder ungefiltert. Offen. bedeutet, dass auf diesem Port des Zielrechners eine Anwendung auf eingehende Verbindungen/Pakete lauscht. Gefiltert. bedeutet, dass eine Firewall, ein Filter oder ein anderes Netzwerkhindernis den Port blockiert, so dass Nmap nicht wissen kann, ob er offen oder geschlossen ist. Für geschlossene. Ports gibt es keine Anwendung, die auf ihnen lauscht, auch wenn sie jederzeit geöffnet werden könnten. Als ungefiltert. werden Ports dann klassifiziert, wenn sie auf Nmaps Testpakete antworten, Nmap aber nicht feststellen kann, ob sie offen oder gechlossen sind. Nmap gibt die Zustandskombinationen offen|gefiltert. und geschlossen|gefiltert. an, wenn es nicht feststellen kann, welcher der beiden Zustände für einen Port zutrifft. Die Port−Tabelle enthält eventuell auch Details zur Softwareversion, sofern eine Versionserkennung verlangt wurde. Wurde ein IP−Protokoll−Scan verlangt (**−sO**), dann bietet Nmap Angaben über die unterstützten IP−Protokolle statt über lauschende Ports.

Zusätzlich zur Tabelle der interessanten Ports kann Nmap weitere Angaben über Ziele bieten, darunter Reverse−DNS−Namen, Mutmaßungen über das benutzte Betriebssystem, Gerätearten und MAC−Adressen.

Einen typischen Nmap−Scan sehen Sie in Example 1. Die einzigen in diesem Beispiel benutzten Nmap−Argumente sind **−A** für die Betriebssystem− und Versionserkennung, Script−Scanning und Traceroute und **−T4** für eine schnellere Ausführung. Danach kommen die Namen der Zielhosts.

Example 1. Ein repräsentativer Nmap-Scan

```
# nmap −A −T4 scanme.nmap.org

Starting Nmap ( https://nmap.org )
Interesting ports on scanme.nmap.org (64.13.134.52):
Not shown: 994 filtered ports
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH 4.3 (protocol 2.0)
25/tcp  closed smtp
53/tcp  open   domain  ISC BIND 9.3.4
70/tcp  closed gopher
80/tcp  open   http    Apache httpd 2.2.2 ((Fedora))
|_ HTML title: Go ahead and ScanMe!
113/tcp closed auth
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.20−1 (Fedora Core 5)
```

```
TRACEROUTE (using port 80/tcp)
HOP RTT   ADDRESS
[Cut first seven hops for brevity]
8   10.59 so−4−2−0.mpr3.pao1.us.above.net (64.125.28.142)
9   11.00 metro0.sv.svcolo.com (208.185.168.173)
10  9.93  scanme.nmap.org (64.13.134.52)

Nmap done: 1 IP address (1 host up) scanned in 17.00 seconds
```

Die neueste Version von Nmap erhält man unter **https://nmap.org**, und die neueste Version der Manpage ist unter **https://nmap.org/book/man.html** verfügbar.

ÜBERSICHT DER OPTIONEN

Diese Übersicht wird ausgegeben, wenn Nmap ohne Argumente aufgerufen wird; die neueste Version davon ist immer unter **https://nmap.org/data/nmap.usage.txt** verfügbar. Sie hilft dabei, sich die am häufigsten benutzten Optionen zu merken, ist aber kein Ersatz für die detaillierte Dokumentation im Rest dieses Handbuchs. Einige obskure Optionen werden hier nicht einmal erwähnt.

```
Nmap 4.85BETA8 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0−255.1−254
  −iL <inputfilename>: Input from list of hosts/networks
  −iR <num hosts>: Choose random targets
  −−exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  −−excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  −sL: List Scan − simply list targets to scan
  −sP: Ping Scan − go no further than determining if host is online
  −PN: Treat all hosts as online −− skip host discovery
  −PS/PA/PU[portlist]: TCP SYN/ACK or UDP discovery to given ports
  −PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  −PO[protocol list]: IP Protocol Ping
  −n/−R: Never do DNS resolution/Always resolve [default: sometimes]
  −−dns−servers <serv1[,serv2],...>: Specify custom DNS servers
  −−system−dns: Use OS's DNS resolver
  −−traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  −sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  −sU: UDP Scan
  −sN/sF/sX: TCP Null, FIN, and Xmas scans
  −−scanflags <flags>: Customize TCP scan flags
  −sI <zombie host[:probeport]>: Idle scan
  −sO: IP protocol scan
  −b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  −p <port ranges>: Only scan specified ports
    Ex: −p22; −p1−65535; −p U:53,111,137,T:21−25,80,139,8080
  −F: Fast mode − Scan fewer ports than the default scan
  −r: Scan ports consecutively − don't randomize
  −−top−ports <number>: Scan <number> most common ports
  −−port−ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  −sV: Probe open ports to determine service/version info
  −−version−intensity <level>: Set from 0 (light) to 9 (try all probes)
```

−−version−light: Limit to most likely probes (intensity 2)
−−version−all: Try every single probe (intensity 9)
−−version−trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
−sC: equivalent to −−script=default
−−script=<Lua scripts>: <Lua scripts> is a comma separated list of
directories, script−files or script−categories
−−script−args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
−−script−trace: Show all data sent and received
−−script−updatedb: Update the script database.
OS DETECTION:
−O: Enable OS detection
−−osscan−limit: Limit OS detection to promising targets
−−osscan−guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
Options which take <time> are in milliseconds, unless you append 's'
(seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
−T<0−5>: Set timing template (higher is faster)
−−min−hostgroup/max−hostgroup <size>: Parallel host scan group sizes
−−min−parallelism/max−parallelism <time>: Probe parallelization
−−min−rtt−timeout/max−rtt−timeout/initial−rtt−timeout <time>: Specifies
probe round trip time.
−−max−retries <tries>: Caps number of port scan probe retransmissions.
−−host−timeout <time>: Give up on target after this long
−−scan−delay/−−max−scan−delay <time>: Adjust delay between probes
−−min−rate <number>: Send packets no slower than <number> per second
−−max−rate <number>: Send packets no faster than <number> per second
FIREWALL/IDS EVASION AND SPOOFING:
−f; −−mtu <val>: fragment packets (optionally w/given MTU)
−D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
−S <IP_Address>: Spoof source address
−e <iface>: Use specified interface
−g/−−source−port <portnum>: Use given port number
−−data−length <num>: Append random data to sent packets
−−ip−options <options>: Send packets with specified ip options
−−ttl <val>: Set IP time−to−live field
−−spoof−mac <mac address/prefix/vendor name>: Spoof your MAC address
−−badsum: Send packets with a bogus TCP/UDP checksum
OUTPUT:
−oN/−oX/−oS/−oG <file>: Output scan in normal, XML, s|<rIpt kIddi3,
and Grepable format, respectively, to the given filename.
−oA <basename>: Output in the three major formats at once
−v: Increase verbosity level (use twice or more for greater effect)
−d[level]: Set or increase debugging level (Up to 9 is meaningful)
−−reason: Display the reason a port is in a particular state
−−open: Only show open (or possibly open) ports
−−packet−trace: Show all packets sent and received
−−iflist: Print host interfaces and routes (for debugging)
−−log−errors: Log errors/warnings to the normal−format output file
−−append−output: Append to rather than clobber specified output files
−−resume <filename>: Resume an aborted scan
−−stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
−−webxml: Reference stylesheet from Nmap.Org for more portable XML
−−no−stylesheet: Prevent associating of XSL stylesheet w/XML output

MISC:
−6: Enable IPv6 scanning
−A: Enables OS detection and Version detection, Script scanning and Traceroute
−−datadir <dirname>: Specify custom Nmap data file location
−−send−eth/−−send−ip: Send using raw ethernet frames or IP packets
−−privileged: Assume that the user is fully privileged
−−unprivileged: Assume the user lacks raw socket privileges
−V: Print version number
−h: Print this help summary page.
EXAMPLES:
nmap −v −A scanme.nmap.org
nmap −v −sP 192.168.0.0/16 10.0.0.0/8
nmap −v −iR 10000 −PN −p 80
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES

ANGABE VON ZIELEN

Nmap betrachtet alles in der Kommandozeile, was keine Option (oder ein Argument einer Option) ist, als Bezeichnung eines Zielhosts. Der einfachste Fall ist die Beschreibung einer IP−Zieladresse oder eines Zielhostnamens zum Scannen.

Manchmal möchten Sie ein ganzes Netzwerk benachbarter Hosts scannen. Dafür unterstützt Nmap Adressen im CIDR−Stil.. Sie können /*numbits* an eine IPv4−Adresse oder einen Hostnamen anfügen, und Nmap wird alle IP−Adressen scannen, bei denen die ersten *numbits* mit denen der gegebenen IP oder des gegebenen Hostnamens übereinstimmen. Zum Beispiel würde 192.168.10.0/24 die 256 Hosts zwischen 192.168.10.0 (binär: 11000000 10101000 00001010 00000000) und 192.168.10.255 (binär: 11000000 10101000 00001010 11111111, inklusive) scannen. 192.168.10.40/24 würde genau dieselben Ziele scannen. Dadurch, dass der Host scanme.nmap.org. die IP−Adresse 64.13.134.52 hat, würde die Angabe scanme.nmap.org/16 die 65.536 IP−Adressen zwischen 64.13.0.0 und 64.13.255.255 scannen. Der kleinste erlaubte Wert ist /0, der das gesamte Internet scannt. Der größte Wert ist /32 und scannt lediglich den Host mit angegebenem Namen oder IP−Adresse, da alle Adressen−Bits festgelegt sind.

Die CIDR−Notation ist kurz, aber nicht immer flexibel genug. Vielleicht möchten Sie z.B. 192.168.0.0/16 scannen, aber IPs auslassen, die mit .0 oder .255 enden, weil sie als Unternetzwerk und Broadcast−Adressen benutzt werden können. Nmap unterstützt das in Form einer Oktett−Bereichsadressierung. Statt eine normale IP−Adresse anzugeben, können Sie eine mit Kommata getrennte Liste von Zahlen oder Bereichen für jedes Oktett angeben. Zum Beispiel überspringt 192.168.0−255.1−254 alle Adressen im Bereich, die mit .0 oder .255 enden, und 192.168.3−5,7.1 scannt die vier Adressen 192.168.3.1, 192.168.4.1, 192.168.5.1 und 192.168.7.1. Beide Bereichsgrenzen können weggelassen werden, die Standardwerte sind 0 für die linke und 255 für die rechte Grenze. Wenn Sie allein − benutzen, ist das identisch mit 0−255, aber denken Sie daran, im ersten Oktett 0− zu benutzen, damit die Zielangabe nicht wie eine Kommandozeilenoption aussieht. Diese Bereiche müssen nicht auf die endgültigen Oktetts beschränkt sein: die Angabe 0−255.0−255.13.37 führt einen internetweiten Scan über alle IP−Adressen aus, die mit 13.37 enden. Diese Art von breiter Abtastung kann bei Internet−Umfragen und −Forschungen hilfreich sein.

IPv6−Adressen können nur durch ihre vollständige IPv6−Adresse oder ihren Hostnamen angegeben werden. CIDR und Oktettbereiche werden für IPv6 nicht unterstützt, weil sie selten nützlich sind.

Nmap akzeptiert in der Kommandozeile mehrere Host−Angaben, die auch nicht vom selben Typ sein müssen. Der Befehl **nmap scanme.nmap.org 192.168.0.0/8 10.0.0,1,3−7.−** macht also das, was Sie erwarten würden.

Auch wenn Ziele normalerweise in der Kommandozeile angegeben werden, gibt es auch die folgenden Optionen, um die Zielauswahl zu steuern:

−iL *inputfilename* (Eingabe aus einer Liste) .
Eine sehr lange Liste von Hosts in der Kommandozeile anzugeben ist oft sehr umständlich, kommt aber sehr häufig vor. Ihr DHCP−Server z.B. exportiert vielleicht eine Liste von 10.000 aktuellen Adresszuweisungen (engl. leases), die Sie scannen möchten. Oder vielleicht möchten Sie alle

IP−Adressen *außer* denjenigen scannen, um Hosts zu finden, die unautorisierte statische IP−Adressen benutzen. Erzeugen Sie einfach die Liste der zu scannenden Hosts und übergeben Sie deren Dateinamen als Argument zur Option **−iL** an Nmap. Die Einträge dürfen alle Formate haben, die Nmap auf der Kommandozeile akzeptiert (IP−Adresse, Hostname, CIDR, IPv6 oder Oktettbereiche). Alle Einträge müssen durch ein oder mehrere Leerzeichen, Tabulatoren oder Zeilenumbrüche getrennt sein. Wenn Sie einen Bindestrich (−) als Dateinamen angeben, liest Nmap die Hosts von der Standardeingabe statt aus einer normalen Datei.

−iR *num hosts* (zufällige Auswahl von Zielen) .
Für internetweite Umfragen und andere Forschungsaktivitäten möchten Sie Ziele vielleicht zufällig auswählen. Das kann man mit der Option **−iR**, die als Argument die Anzahl der zu erzeugenden IPs annimmt. Nmap lässt automatisch bestimmte unerwünschte IPs aus, wie solche in privaten, Multicast− oder unbesetzten Adressbereichen. Für einen endlosen Scan kann man das Argument 0 angeben. Denken Sie aber daran, dass manche Netzwerkadministratoren sich gegen unautorisierte Scans ihrer Netzwerke sträuben. Lesen Sie the section called “” sorgfältig, bevor Sie −iR benutzen.

Falls Sie mal an einem regnerischen Tag wirklich Langeweile haben, probieren Sie einmal den Befehl **nmap −sS −PS80 −iR 0 −p 80**. aus, um zufällig Webserver zu finden, auf denen Sie herumstöbern können.

−−exclude *host1***[,***host2*[,...]] (Ziele ausklammern) .
Gibt eine mit Kommata getrennte Liste von Zielen an, die vom Scan ausgeschlossen sein sollen, selbst wenn sie in den angegebenen Netzwerkbereich fallen. Die übergebene Liste benutzt die normale Nmap−Syntax und kann folglich Hostnamen, CIDR−Netzblöcke, Oktettbereiche usw. enthalten. Das kann nützlich sein, wenn das zu scannende Netzwerk hochkritische Server und Systeme enthält, die man nicht anfassen darf, weil sie bekanntermaßen ungünstig auf Port−Scans reagieren, oder Unternetze, die von anderen Leuten administriert werden.

−−excludefile *exclude_file* (Liste aus Datei ausklammern) .
Das bietet dieselbe Funktionalität wie die Option **−−exclude**, mit dem Unterschied, dass die ausgeklammerten Ziele in der mit Zeilenumbrüchen, Leerzeichen oder Tabulatoren getrennten Datei *exclude_file* statt auf der Kommandozeile angegeben werden.

HOST-ERKENNUNG

Einer der allerersten Schritte bei jeder Netzwerkerkundung ist die Reduktion einer (manchmal riesigen) Menge von IP−Bereichen auf eine Liste aktiver oder interessanter Hosts. Wenn man für alle einzelnen IP−Adressen alle Ports scannt, so ist das nicht nur langsam, sondern normalerweise auch unnötig. Was einen Host interessant macht, hängt natürlich stark vom Zweck der Untersuchung ab. Netzwerkadministratoren interessieren sich vielleicht nur für Hosts, auf denen ein bestimmter Dienst läuft, während Sicherheitsprüfer sich vielleicht für alle Geräte interessieren, die eine IP−Adresse haben. Ein Administrator benötigt vielleicht nur einen ICMP−Ping, um Hosts in seinem internen Netzwerk zu finden, während ein externer Penetrationstester vielleicht Dutzende ganz verschiedener Tests einsetzen wird, um zu versuchen, die Firewall−Beschränkungen zu umgehen.

Da die Anforderungen bei der Host−Erkennung so verschieden sind, bietet Nmap eine breite Palette von Optionen zur Anpassung der eingesetzten Verfahren. Trotz seines Namens geht ein Ping−Scan weit über die einfachen ICMP Echo−Request−Pakete hinaus, die mit dem allgegenwärtigen Werkzeug ping verbunden sind. Man kann den Ping−Schritt völlig auslassen, indem man einen List−Scan (**−sL**) benutzt, Ping ausschaltet (**−PN**) oder beliebige Kombinationen von Multi Port TCP−SYN/ACK, UDP− und ICMP−Testanfragen auf ein Netzwerk loslässt. Der Zweck dieser Anfragen ist der, Antworten hervorzurufen, die zeigen, dass eine IP−Adresse tatsächlich aktiv ist (d.h. von einem Host oder Gerät im Netzwerk benutzt wird). In vielen Netzwerken ist nur ein kleiner Prozentsatz von IP−Adressen zu einem bestimmten Zeitpunkt aktiv. Das gilt besonders für einen privaten Adressraum wie 10.0.0.0/8. Dieses Netzwerk enthält 16,8 Millionen IPs, aber ich habe auch Firmen gesehen, die es mit weniger als tausend Rechnern benutzen. Mit der Host−Erkennung kann man diese spärlichen Rechnerinseln in einem Meer von IP−Adressen finden.

Falls keine Optionen für die Host−Erkennung angegeben werden, sendet Nmap ein TCP−ACK−Paket an

Port 80 und ein ICMP Echo−Request an alle Zielrechner. Eine Ausnahme ist, dass bei allen Zielen in einem lokalen Ethernet−Netzwerk ein ARP−Scan benutzt wird. Für unprivilegierte Unix−Shell−Benutzer wird mit dem **connect**−Systemaufruf ein SYN−Paket statt eines ACK gesendet.. Diese Standardeinstellungen sind äquivalent zu den Optionen **−PA −PE**. Diese Host−Erkennung ist oft ausreichend, wenn man lokale Netzwerke scannt, aber für Sicherheitsüberprüfungen empfiehlt sich eine umfangreichere Menge von Erkennungstestpaketen.

Die Optionen **−P*** (die Ping−Typen auswählen) lassen sich kombinieren. Sie können Ihre Chancen steigern, bei strengen Firewalls durchzukommen, indem Sie viele Testpaketarten mit verschiedenen TCP−Ports/−Flags und ICMP−Codes senden. Beachten Sie auch, dass die ARP−Erkennung (**−PR**). bei Zielen in einem lokalen Ethernet−Netzwerk standardmäßig erfolgt, selbst dann, wenn Sie andere **−P***−Optionen angeben, weil sie fast immer schneller und effizienter ist.

Standardmäßig führt Nmap eine Host−Erkennung und dann einen Port−Scan auf jedem Host aus, den es als online erkennt. Das gilt auch dann, wenn Sie nicht standardmäßige Host−Erkennungstypen wie UDP−Testpakete (**−PU**) angeben. Lesen Sie über die Option **−sP** nach, um zu lernen, wie man nur eine Host−Erkennung durchführt, oder über **−PN**, um die Host−Erkennung zu überspringen und einen Port−Scan aller Zielhosts durchzuführen. Folgende Optionen steuern die Host−Erkennung:

−sL (List−Scan) .
Ein List−Scan ist eine degenerierte Form der Host−Erkennung, die einfach jeden Host im angegebenen Netzwerk (bzw. den Netzwerken) auflistet, ohne Pakete an die Ziel−Hosts zu senden. Standardmäßig führt Nmap immer noch eine Reverse−DNS−Auflösung der Hosts durch, um deren Namen zu lernen. Es ist oft erstaunlich, wie viel nützliche Informationen einfache Hostnamen verraten. Zum Beispiel ist fw.chi der Name einer Firewall einer Firma in Chicago. Nmap gibt am Ende auch die gesamte Anzahl der IP−Adressen aus. Ein List−Scan ist eine gute Plausibilitätsprüfung, um sicherzustellen, dass Sie saubere IP−Adressen für Ihre Ziele haben. Falls die Hosts Domainnamen enthalten, die Ihnen nichts sagen, lohnt sich eine weitere Untersuchung, um zu verhindern, dass Sie das Netzwerk der falschen Firma scannen.

Da die Idee einfach die ist, eine Liste der Zielhosts auszugeben, lassen sich Optionen für eine höhere Funktionalität wie z.B. Port−Scanning, Betriebssystemerkennung oder Ping−Scanning damit nicht kombinieren. Falls Sie einen Ping−Scan abschalten und trotzdem solch höhere Funktionalität durchführen möchten, lesen Sie bei der Option **−PN** weiter.

−sP (Ping−Scan) .
Diese Option verlangt, dass Nmap nur einen Ping−Scan (Host−Erkennung) durchführt und dann die verfügbaren Hosts ausgibt, die auf den Scan geantwortet haben. Darüber hinaus werden keine weiteren Tests (z.B. Port−Scans oder Betriebssystemerkennung) durchgeführt, außer bei Host−Scripts mit der Nmap Scripting Engine und traceroute−Tests, sofern Sie diese Optionen angegeben haben. Das ist eine Stufe aufdringlicher als ein List−Scan und kann oft für dieselben Zwecke benutzt werden. Sie führt schnell eine schwache Aufklärung des Zielnetzwerks durch, ohne viel Aufmerksamkeit zu erregen. Für Angreifer ist es wertvoller, zu wissen, wie viele Hosts verfügbar sind, als die Liste aller IPs und Hostnamen aus einem List−Scan zu kennen.

Für Systemadministratoren ist diese Option oft ebenfalls wertvoll. Mit ihr kann man sehr leicht die verfügbaren Rechner in einem Netzwerk zählen oder die Server−Verfügbarkeit überwachen. So etwas nennt man oft auch einen Ping−Sweep, und es ist zuverlässiger als ein Pinging auf die Broadcast−Adresse, weil viele Hosts auf Broadcast−Anfragen nicht antworten.

Die Option **−sP** sendet standardmäßig einen ICMP Echo−Request und ein TCP−ACK−Paket an Port 80. Bei Ausführung ohne Sonderrechte wird nur ein SYN−Paket (mit einem **connect**−Aufruf) an Port 80 an das Ziel gesendet. Wenn ein Benutzer mit Sonderrechten versucht, Ziele in einem lokalen Ethernet−Netzwerk zu scannen, werden ARP−Requests verwendet, es sei denn, die Option **−−send−ip** wird angegeben. Die Option **−sP** kann mit allen Erkennungsmethoden (die Optionen **−P***, außer **−PN**) kombiniert werden, um eine höhere Flexibilität zu erhalten. Falls zwischen dem Ausgangs−Host, auf dem Nmap läuft, und dem Zielnetzwerk strenge Firewalls installiert sind, empfehlen sich diese

fortgeschrittenen Methoden. Ansonsten könnten Hosts übersehen werden, wenn die Firewall Testanfragen oder Antworten darauf verwirft

−PN (Ping abschalten) .
Eine weitere Möglichkeit ist die, die Erkennungsphase von Nmap völlig auszulassen. Normalerweise bestimmt Nmap in dieser Phase aktive Rechner, die es anschließend stärker scannt. Standardmäßig führt Nmap heftigere Tests wie Port−Scans, Versions− oder Betriebssystemerkennung bei Hosts durch, die es bereits als aktiv eingestuft hat. Das Ausschalten der Host−Erkennung mit der Option **−PN** bewirkt, dass Nmap versucht, die gewünschten Scan−Funktionen auf *allen* angegebenen Ziel−IP−Adresssen durchzuführen. Wenn also ein Zieladressraum der Größe Klasse B (/16) auf der Kommandozeile angegeben wird, werden alle 65.536 IP−Adressen gescannt. Eine richtige Host−Erkennung wird wie bei einem List−Scan übersprungen, aber statt anzuhalten und die Zielliste auszugeben, fährt Nmap mit der Durchführung der gewünschten Funktionen fort, so als ob jede Ziel−IP aktiv wäre. Bei Rechnern im lokalen Ethernet−Netzwerk wird ein ARP−Scan weiterhin ausgeführt (es sei denn, es wird **−−send−ip** angegeben), da Nmap MAC−Adressen braucht, um Zielhosts weiter zu scannen. Diese Option lautete früher einaml **P0** (mit einer Null), wurde dann aber umbenannt, um Verwirrung mit der Option **PO** von Protokoll−Pings (benutzt den Buchstaben O) zu vermeiden.

−PS *port list* (TCP−SYN−Ping) .
Diese Option sendet ein leeres TCP−Paket mit gesetztem SYN−Flag. Der vorgegebene Zielport ist 80 (lässt sich zum Zeitpunkt des Kompilierens durch Ändern von *DEFAULT_TCP_PROBE_PORT_SPEC*. in nmap.h konfigurieren),. aber man kann einen alternativen Port als Parameter angeben. Die Syntax ist dieselbe wie bei **−p**, mit dem Unterschied, dass Porttypenbezeichner wie T: nicht erlaubt sind. Beispiele hierfür sind **−PS22** und **−PS22−25,80,113,1050,35000**. Beachten Sie, dass es kein Leerzeichen zwischen **−PS** und der Port−Liste geben darf. Falls mehrere Tests angegeben werden, werden sie parallel durchgeführt.

Das SYN−Flag bedeutet für das entfernte System, dass Sie versuchen, eine Verbindung herzustellen. Normalerweise wird der Zielport geschlossen sein, und es wird ein RST−(Reset−)Paket zurückgeschickt. Falls der Port offen ist, führt das Ziel den zweiten Schritt eines TCP−three−way−handshake durch, indem es mit einem SYN/ACK−TCP−Paket antwortet. Der Rechner, auf dem Nmap läuft, bricht dann die entstehende Verbindung ab, indem er mit einem RST antwortet, statt ein ACK−Paket zu senden, mit dem der three−way handshake komplett und eine vollständige Verbindung hergestellt wäre. Das RST−Paket wird als Antwort auf das unerwartete SYN/ACK vom Betriebssystem−Kernel des Rechners gesendet, auf dem Nmap läuft, nicht von Nmap selbst.

Für Nmap ist es egal, ob der Port offen oder geschlossen ist. Aus beiden Antworten, ob RST oder SYN/ACK, schließt Nmap, dass der Host verfügbar ist und antwortet.

Auf Unix−Rechnern kann im Allgemeinen nur der mit Sonderrechten ausgestattete Benutzer root. rohe TCP−Pakete senden und empfangen.. Bei normalen Benutzern kommt automatisch eine Umgehungslösung zum Tragen,. bei der für alle Zielports der **connect**−Systemaufruf verwendet wird. Das bewirkt, dass an den Zielhost ein SYN−Paket gesendet wird, mit der Absicht, eine Verbindung herzustellen. Falls **connect** schnell ein erfolgreiches Ergebnis oder einen ECONNREFUSED−Fehler zurückgibt, muss der darunterliegende TCP−Stack ein SYN/ACK oder RST empfangen haben, und der Host wird als verfügbar vermerkt. Falls der Verbindungsversuch hängenbleibt, bis eine Zeitbeschränkung erreicht ist, wird der Host als inaktiv vermerkt. Diese Behelfslösung wird auch bei IPv6−Verbindungen verwendet, da Nmap den Bau roher IPv6−Pakete noch nicht unterstützt..

−PA *port list* (TCP−ACK−Ping) .
Der TCP−ACK−Ping ist ziemlich ähnlich zum SYN−Ping. Der Unterschied ist der, dass das TCP−ACK−Flag statt dem SYN−Flag gesetzt wird, was Sie sich bestimmt schon gedacht haben. Ein solches ACK−Paket erweckt den Eindruck, es wolle Daten auf einer bestehenden TCP−Vebindung bestätigen, während eine solche Verbindung gar nicht existiert. Entfernte Hosts sollten darauf immer

mit einem RST−Paket antworten, wobei sie ihre Existenz verraten.

Die Option **−PA** benutzt denselben Standard−Port wie der SYN−Test (80) und nimmt ebenfalls eine Liste von Zielports im selben Format an. Falls ein unprivilegierter Benutzer das ausprobiert oder falls ein IPv6−Ziel angegeben wird, wird die bereits erwähnte Behelfslösung mit **connect** eingesetzt. Diese ist nicht perfekt, da **connect** tatsächlich ein SYN−Paket statt eines ACK sendet.

Der Grund für die Existenz sowohl von SYN− als auch ACK−Ping−Tests liegt darin, die Chancen für die Umgehung von Firewalls zu erhöhen. Viele Administratoren konfigurieren Router und andere einfache Firewalls so, dass sie eingehende SYN−Pakete blockieren, außer bei solchen für öffentliche Dienste wie bei der Website oder dem Mailserver der Firma. Das verhindert weitere eingehende Verbindungen zur Organisation, während es den Benutzern freie Verbindungen ins Internet erlaubt. Dieser zustandslose Ansatz benötigt wenige Ressourcen in der Firewall bzw. im Router und wird von Hardware− und Software−Filtern weithin unterstützt. Die Firewall−Software Netfilter/iptables. in Linux bietet die komfortable Option **−−syn**, um diesen zustandslosen Ansatz zu implementieren. Wenn solche Firewall−Regeln vorhanden sind, werden SYN−Ping−Tests (**−PS**), die an geschlossene Zielports gesendet werden, sehr wahrscheinlich blockiert. In solchen Fällen greift der ACK−Test, da er diese Regeln einfach kappt.

Eine weitere häufige Art von Firewalls verwendet zustandsbehaftete Regeln, die unerwartete Pakete verwerfen. Dieses Merkmal konnte man zuerst bei hochwertigen Firewalls finden, es hat sich aber mit der Zeit deutlich verbreitet. In Linux unterstützt das Netfilter/iptables−System das mit der Option −−state, die Pakete nach einem Verbindungszustand kategorisiert. In solchen Fällen hat der SYN−Test eine wesentlich bessere Chance auf Erfolg, da unerwartete ACK−Pakete im Allgemeinen als fehlerhaft erkannt und verworfen werden. Eine Lösung aus diesem Dilemma besteht darin, mit **−PS** und **−PA** SYN− und ACK−Testpakete zu senden.

−PU *port list* (UDP−Ping) .
Eine weitere Möglichkeit bei der Host−Erkennung ist der UDP−Ping, bei dem ein leeres (außer bei Angabe von **−−data−length**) UDP−Paket an die angegebenen Ports gesendet wird. Die Portliste hat dasselbe Format wie bei den weiter oben beschriebenen Optionen **−PS** und **−PA**. Falls keine Ports angegeben werden, ist die Standardeinstellung 31338. Dieser Wert kann zum Zeitpunkt des Kompilierens dürch Änderung von *DEFAULT_UDP_PROBE_PORT_SPEC*. in nmap.h konfiguriert werden.. Es wird absichtlich ein sehr unwahrscheinlicher Port verwendet, weil bei dieser bestimmten Art des Scannens das Senden an offene Ports oft unerwünscht ist.

Trifft der UDP−Test beim Zielrechner auf einen geschlossenen Port, so sollte dieser ein ICMP−Paket zurückschicken, das besagt, dass der Port nicht erreichbar ist. Daraus schließt Nmap, dass der Rechner läuft und verfügbar ist. Viele weitere Arten von ICMP−Fehlern, z.B. bei unerreichbaren Hosts/Netzwerken oder überschrittener TTL (Time To Live), sind Zeichen für einen abgeschalteten oder unerreichbaren Host. Auch eine ausbleibende Antwort wird so interpretiert. Falls ein offener Port erreicht wird, ignorieren die meisten Dienste das leere Paket einfach und geben keine Antwort zurück. Deswegen wird als Standardport 31338 benutzt, bei dem es sehr unwahrscheinlich ist, dass er benutzt wird. Einige Dienste, wie z.B. das Character Generator−Protokoll (chargen), antworten auf ein leeres UDP−Paket und enthüllen damit Nmap gegenüber, dass der Rechner zugänglich ist.

Der Hauptvorteil dieses Scan−Typs liegt darin, dass er Firewalls und Filter umgeht, die nur TCP überprüfen. Ich hatte z.B. einmal ein BEFW11S4, einen Wireless−Breitband−Router von Linksys. Die externe Schnittstelle dieses Geräts filterte standardmäßig alle TCP−Ports, aber UDP−Tests entlockten ihm weiterhin Meldungen über unerreichbare Ports und verrieten damit das Gerät.

−PE; **−PP**; **−PM** (ICMP−Ping−Arten) .
Zusätzlich zu den genannten ungewöhnlichen TCP− und UDP−Host−Erkennungsarten kann Nmap auch Standardpakete senden, wie sie das allgegenwärtige Programm ping sendet. Nmap sendet ein ICMP Typ−8−Paket (Echo−Request) an die Ziel−IP−Adressen und erwartet eine Typ−0−Antwort (Echo−Reply) vom verfügbaren Host.. Zum Leidwesen von Netzwerkerkundern blockieren viele

Hosts und Firewalls heute diese Pakete, statt, wie in **RFC 1122**[1] verlangt, darauf zu antworten. Aus diesem Grund sind ICMP−Scans allein bei unbekannten Zielen über das Internet selten zuverlässig genug. Aber für Systemadministratoren, die ein internes Netzwerk überwachen, kann das ein praktischer und wirksamer Ansatz sein. Benutzen Sie die Option **−PE**, um dieses Verhalten mit Echo−Requests einzuschalten.

Auch wenn ein Echo−Request die Standard−ICMP−Ping−Abfrage ist, hört Nmap hier nicht auf. Der ICMP−Standard (**RFC 792**[2]) spezifiziert auch Anfragepakete für Zeitstempel, Information und Adressmaske mit den jeweiligen Codes 13, 15 und 17. Während diese Anfragen angeblich den Zweck haben, an Informationen wie Address Mask und Timestamp zu gelangen, können sie auch leicht für die Host−Erkennung benutzt werden. Im Moment implementiert Nmap keine Information−Request−Pakete, da sie nicht weit verbreitet sind (RFC 1122 besteht darauf, dass “ein Host diese Nachrichten NICHT implementieren SOLLTE”). Anfragen nach Timestamp und Address Mask können jeweils mit den Optionen **−PP** und **−PM** gesendet werden. Eine Timestamp−Antwort (ICMP−Code 14) oder Address−Mask−Antwort (Code 18) enthüllt, dass der Host greifbar ist. Diese beiden Abfragen können wertvoll sein, wenn Administratoren ausdrücklich Echo−Request−Pakete blockieren, aber vergessen, dass man für den gleichen Zweck auch andere ICMP−Abfragen benutzen kann.

−PO *protocol list* (IP−Protokoll−Ping) .
Die neueste Möglichkeit der Host−Erkennung ist ein IP−Protokoll−Ping, der IP−Pakete sendet, in deren IP−Header die angegebene Protokollnummer gesetzt ist. Die Protokoll−Liste hat dasselbe Format wie Portlisten bei den weiter oben vorgestellten Optionen der TCP− und UDP−Host−Erkennung. Ohne Angabe von Protokollen werden standardmäßig mehrere IP−Pakete für ICMP (Protokoll 1), IGMP (Protokoll 2) und IP−in−IP (Protokoll 4) gesendet. Die Standardprotokolle können zum Zeitpunkt des Kompilierens durch Veränderung von *DEFAULT_PROTO_PROBE_PORT_SPEC*. in nmap.h konfiguriert werden. Beachten Sie, dass für ICMP, IGMP, TCP (Protokoll 6) und UDP (Protokoll 17) die Pakete mit den richtigen Protokoll−Headern gesendet werden, während andere Protokolle ohne weitere Daten über den IP−Header hinaus gesendet werden (es sei denn, die Option **−−data−length** wird angegeben).

Diese Methode der Host−Erkennung sucht nach Antworten, die entweder dasselbe Protokoll wie der Test haben, oder Meldungen, dass das ICMP−Protokoll nicht erreichbar ist, was bedeutet, dass das gegebene Protokoll vom Zielhost nicht unterstützt wird. Beide Antworten bedeuten, dass der Zielhost am Leben ist.

−PR (ARP−Ping) .
Eines der häufigsten Einsatzszenarien für Nmap ist das Scannen eines Ethernet−LANs. In den meisten LANs, besonders jenen, die durch **RFC 1918**[3] erteilte private Adressbereiche verwenden, wird der Großteil der IP−Adressen meistens nicht genutzt. Wenn Nmap versucht, ein rohes IP−Paket wie z.B. ein ICMP Echo−Request zu senden, muss das Betriebssystem die der Ziel−IP entsprechende Hardware−Zieladresse (ARP) bestimmen, damit es den Ethernet−Frame korrekt adressieren kann. Das ist oft langsam und problematisch, da Betriebssysteme nicht in der Erwartung geschrieben wurden, dass sie in kurzer Zeit Millionen von ARP−Anfragen bei nicht erreichbaren Hosts durchführen müssen.

Beim ARP−Scan ist Nmap mit seinen optimierten Algorithmen zuständig für ARP−Anfragen. Und wenn es eine Antwort erhält, muss sich Nmap nicht einmal um die IP−basierten Ping−Pakete kümmern, da es bereits weiß, dass der Host aktiv ist. Das macht den ARP−Scan viel schneller und zuverlässiger als IP−basierte Scans. Deswegen wird er standardmäßig ausgeführt, wenn Ethernet−Hosts gescannt werden, bei denen Nmap bemerkt, dass sie sich in einem lokalen Ethernet−Netzwerk befinden. Selbst wenn verschiedene Ping−Arten (wie z.B. **−PE** oder **−PS**) angegeben werden, benutzt Nmap stattdessen ARP bei allen Zielen, die im selben LAN sind. Wenn Sie einen ARP−Scan auf gar keinen Fall durchführen möchten, geben Sie **−−send−ip** an.

−−traceroute (Traceroutes zum Host) .

Traceroutes werden nach einem Scan mit Hilfe der Information aus den Scan−Ergebnissen durchgeführt, um den wahrscheinlichsten Port und das wahrscheinlichste Protokoll zu bestimmen, die zum Ziel führen. Es funktioniert mit allen Scan−Arten außer Connect−Scans (**−sT**) und Idle−Scans (**−sI**). Alle Traces benutzen Nmaps dynamisches Timing−Modell und werden parallel durchgeführt.

Traceroute funktioniert dadurch, dass es Pakete mit kurzer TTL (Time To Live) sendet und damit versucht, ICMP Time−Exceeded−Nachrichten von Sprungstellen zwischen dem Scanner und dem Zielhost hervorzurufen. Standardimplementationen von Traceroute fangen mit einer TTL von 1 an und inkrementieren die TTL, bis der Zielhost erreicht ist. Nmaps Traceroute fängt mit einer hohen TTL an und verringert sie, bis sie Null erreicht. Durch dieses umgekehrte Vorgehen kann Nmap clevere Caching−Algorithmen benutzen, um Traces über mehrere Hosts zu beschleunigen. Im Durchschnitt sendet Nmap je nach Netzwerkbedingungen 5–10 Pakete weniger pro Host. Wenn ein einziges Unternetz gescannt wird (z.B. 192.168.0.0/24), muss Nmap an die meisten Hosts eventuell nur ein einziges Paket senden.

−n (keine DNS−Auflösung) .
Weist Nmap an, *niemals* eine Reverse−DNS−Auflösung. bei den gefundenen aktiven IP−Adressen durchzuführen. Da DNS selbst mit Nmaps eingebautem parallelen Stub−Resolver langsam sein kann, kann diese Option die Scan−Zeiten dramatisch reduzieren.

−R (DNS−Auflösung für alle Ziele) .
Weist Nmap an, *immer* eine Reverse−DNS−Auflösung bei den Ziel−IP−Adressen durchzuführen. Normalerweise wird Reverse−DNS nur bei anwortenden Hosts (die online sind) durchgeführt.

−−system−dns (verwendet DNS−Auflösung des Systems) .
Standardmäßig löst Nmap IP−Adressen auf, indem es Anfragen direkt an die auf Ihrem Host konfigurierten Nameserver schickt und dann auf Antworten wartet. Um die Performance zu erhöhen, werden viele Anfragen (oftmals Dutzende) parallel ausgeführt. Wenn Sie diese Option angeben, verwenden Sie stattdessen die Auflösungsmethode Ihres Systems (zu jedem Zeitpunkt nur eine IP mit dem Aufruf **getnameinfo** call). Das ist langsam und selten nützlich, es sei denn, Sie finden einen Fehler bei der parallelen Auflösung in Nmap (bitte teilen Sie uns das mit). Bei IPv6−Scans wird immer die Auflösungsmethode des Systems verwendet.

−−dns−servers *server1***[,***server2*[,...]] (Server, die für Reverse−DNS−Anfragen benutzt werden) .
Standardmäßig bestimmt Nmap Ihre DNS−Server (für die rDNS−Auflösung) aus Ihrer Datei resolv.conf (Unix) oder der Registry (Win32). Mit dieser Option können Sie alternative Server dazu angeben. Diese Option bleibt unbeachtet, falls Sie **−−system−dns** oder einen IPv6−Scan benutzen. Oft ist es schneller, mehrere DNS−Server zu benutzen, besonders dann, wenn Sie für Ihren Ziel−IP−Raum maßgebende Server benutzen. Diese Option kann auch die Heimlichkeit erhöhen, da Ihre Anfragen von fast jedem rekursiven DNS−Server im Internet abprallen können.

Diese Option ist auch beim Scannen privater Netzwerke praktisch. Manchmal bieten nur einige wenige Nameserver saubere rDNS−Information, und Sie wissen vielleicht nicht einmal, wo sie sind. Sie können das Netzwerk auf Port 53 scannen (vielleicht mit Versionserkennung), dann Nmap−List−Scans versuchen (**−sL**) und dabei mit der Option **−−dns−servers** immer nur einen Nameserver angeben, bis Sie einen finden, der funktioniert.

GRUNDLAGEN VON PORT-SCANS

Nmap hat über die Jahre an Funktionalität zugelegt, aber angefangen hat es als effizienter Port−Scanner, und das ist weiterhin seine Kernfunktion. Der einfache Befehl **nmap** *target* scannt die am häufigsten verwendeten 1000 TCP−Ports auf dem Host *target* und klassifiziert jeden Port in einen der Zustände offen, geschlossen, gefiltert, ungefiltert, offen|gefiltert oder geschlossen|gefiltert.

Diese Zustände sind keine echten Eigenschaften eines Ports selbst, sondern beschreiben, wie Nmap ihn sieht. Ein Nmap−Scan z.B., bei dem Ausgangs− und Zielnetzwerk identisch sind, könnte Port 135/tcp als offen anzeigen, während ein Scan zur selben Zeit mit denselben Optionen über das Internet diesen Port als gefiltert anzeigen könnte.

Die sechs von Nmap erkannten Port−Zustände

Ein Programm ist bereit, TCP−Verbindungen oder UDP−Pakete auf diesem Port anzunehmen. Beim Port−Scanning ist es oftmals das Ziel, solche Ports zu finden. Sicherheitsbewusste Leute wissen, dass jeder offene Port eine breite Einfahrtstrasse für Angriffe darstellt. Angreifer und Penetrationstester wollen offene Ports ausbeuten (engl. exploit), während Administratoren versuchen, sie zu schließen oder mit Firewalls zu schützen, ohne legitime Benutzer zu behindern. Offene Ports sind auch für Scans von Interesse, bei denen es nicht um Sicherheit geht, weil sie Dienste anzeigen, die im Netzwerk benutzt werden können.

Ein geschlossener Port ist erreichbar (er empfängt und antwortet auf Nmap−Testpakete), aber es gibt kein Programm, das ihn abhört. Er kann von Nutzen sein, um zu zeigen, dass ein Host online ist und eine IP−Adresse benutzt (Host−Erkennung oder Ping−Scanning), sowie als Teil der Betriebssystemerkennung. Weil geschlossene Ports erreichbar sind, sind sie es wert, gescannt zu werden, falls sie später einmal geöffnet werden sollten. Administratoren möchten solche Ports vielleicht mit einer Firewall blockieren, damit sie im Zustand gefiltert erscheinen, der als Nächstes beschrieben wird.

Nmap kann nicht feststellen, ob der Port offen ist, weil eine Paketfilterung verhindert, dass seine Testpakete den Port erreichen. Die Filterung könnte durch dafür vorgesehene Firewall−Geräte, Router−Regeln oder hostbasierte Firewall−Software erfolgen. Weil sie so wenig Information bringen, sind diese Ports für Angreifer frustrierend. Manchmal antworten sie mit ICMP−Fehlermeldungen wie Typ 3, Code 13 (Destination Unreachable: Communication Administratively Prohibited), aber Filter, die Testpakete ohne Antwort einfach verwerfen, kommen wesentlich häufiger vor. Das zwingt Nmap zu mehreren wiederholten Versuchen, um auszuschließen, dass das Testpaket wegen einer Netzwerküberlastung statt durch eine Filterung verworfen wurde. Diese Art der Filterung verlangsamt einen Scan dramatisch.

Der Zustand ungefiltert bedeutet, dass ein Port zugänglich ist, aber Nmap nicht feststellen kann, ob er offen oder geschlossen ist. Nur der ACK−Scan, der benutzt wird, um Firewall−Regelwerke zu bestimmen, klassifiziert Ports in diesen Zustand. Um festzustellen, ob ein ungefilterter Port offen ist, kann es hilfreich sein, ihn mit anderen Scan−Methoden wie Window−Scan, SYN−Scan oder FIN−Scan zu scannen.

Nmap klassifiziert einen Port in diesen Zustand, wenn es nicht feststellen kann, ob der Port offen oder gefiltert ist. Das kommt bei Scan−Methoden vor, in denen offene Ports keine Antwort geben. Das Fehlen einer Antwort könnte auch bedeuten, dass ein Paketfilter das Testpaket verworfen hat oder dass keine Antwort provoziert werden konnte. Deswegen weiß Nmap nicht sicher, ob der Port offen ist oder gefiltert wird. Ports werden von den UDP−, IP−Protokoll−, FIN−, NULL− und Xmas−Scans auf diese Weise klassifiziert.

Dieser Zustand wird benutzt, wenn Nmap nicht feststellen kann, ob ein Port geschlossen ist oder gefiltert wird. Er wird nur vom IP−ID−Idle−Scan benutzt.

PORT-SCANNING-METHODEN

Als Hobby−Automechaniker kann ich mich stundenlang damit herumquälen, meine einfachsten Werkzeuge (Hammer, Klebeband, Schraubenschlüssel etc.) an mein Problem anzupassen. Wenn ich dann kläglich versage und meine alte Blechkiste zu einem echten Mechaniker schleppe, fischt er immer so lange in einer riesigen Werkzeugkiste herum, bis er das perfekte Ding gefunden hat, mit dem sich die Aufgabe fast von allein löst. Bei der Kunst des Port−Scannings ist es ähnlich. Experten kennen Dutzende von Scan−Methoden und wählen für jede Aufgabe die geeignete (oder eine Kombination von mehreren) aus. Auf der anderen Seite versuchen unerfahrene Benutzer und Script−Kiddies,. jedes Problem mit dem standardmäßigen SYN−Scan zu lösen. Da Nmap gratis ist, ist Unwissen das einzige Hindernis auf dem Weg zur Meisterschaft im Port−Scanning. Das ist bestimmt besser als in der Autowelt, wo man eventuell sehr viel Können haben muss, um festzustellen, dass man einen Federbein−Kompressor benötigt, und dann immer noch Tausende dafür bezahlen muss.

Die meisten Scan−Typen stehen nur privilegierten Benutzern zur Verfügung,. und zwar deswegen, weil sie rohe IP−Pakete. senden und empfangen, wofür auf Unix−Systemen root−Rechte benötigt werden. Auf Windows empfiehlt sich ein Administrator−Account, wenngleich auf dieser Plattform Nmap manchmal auch für unprivilegierte Benutzer funktioniert, sofern WinPcap bereits in das Betriebssystem geladen

wurde. Als Nmap 1997 veröffentlicht wurde, war die Voraussetzung von root−Rechten eine ernsthafte Beschränkung, da viele Benutzer nur Zugriff zu Shell−Accounts hatten. Die Welt von heute ist anders. Computer sind billiger, wesentlich mehr Menschen verfügen über einen immer verfügbaren direkten Internet−Zugang, und Desktop−Unix−Systeme (inklusive Linux und Mac OS X) sind weit verbreitet. Eine Windows−Version von Nmap ist nun auch verfügbar, wodurch es nun auf noch mehr Rechnern laufen kann. Aus all diesen Gründen sind Benutzer nur noch selten gezwungen, Nmap von einem beschränkten Shell−Account aus einzusetzen. Das ist erfreulich, denn die privilegierten Optionen machen Nmap wesentlich mächtiger und flexibler.

Auch wenn Nmap versucht, genaue Ergebnisse zu produzieren, sollten Sie nicht vergessen, dass all seine Erkenntnisse auf Paketen basieren, die von den Zielrechnern (oder den Firewalls davor) zurückkommen. Solche Hosts können unzuverlässig sein und eine Antwort senden, die Nmap verwirren oder täuschen soll. Wesentlich häufiger sind Hosts, die nicht RFC−konform sind und auf Testpakete von Nmap nicht so antworten, wie sie sollten. FIN−, NULL− und Xmas−Scans sind für dieses Problem besonders anfällig. Solche Probleme sind spezifisch für bestimmte Scan−Methoden und werden daher in den jeweiligen Abschnitten erörtert.

Dieser Abschnitt dokumentiert die etwa ein Dutzend von Nmap unterstützten Port−Scan−Methoden. Es darf immer nur eine Methode allein benutzt werden, mit der Ausnahme von UDP−Scans (**−sU**), die sich mit allen anderen TCP−Scan−Methoden kombinieren lassen. Hier eine Gedächtnisstütze: Optionen für Port−Scan−Methoden haben die Form **−s***C*, wobei *C* ein bedeutender Buchstabe im Scan−Namen ist, normalerweise der erste. Die eine Ausnahme hiervon ist der als veraltet betrachtete FTP−Bounce−Scan (**−b**). Nmap führt standardmäßig einen SYN−Scan durch, ersetzt diesen aber mit einem Connect−Scan, falls der Benutzer nicht die nötigen Rechte hat, um rohe Pakete (benötigen unter Unix root−Rechte) zu senden, oder falls er IPv6−Ziele angegeben hat. Von den in diesem Abschnitt aufgelisteten Scans dürfen Benutzer ohne Sonderrechte nur den Connect− und FTP−Bounce−Scan ausführen.

−sS (TCP−SYN−Scan) .
Der SYN−Scan ist aus gutem Grund die Standardeinstellung und die beliebteste Scan−Methode. Er kann schnell durchgeführt werden und scannt dabei Tausende von Ports pro Sekunde, wenn das Netzwerk schnell ist und nicht von einer intrusiven Firewall behindert wird. Der SYN−Scan ist relativ unauffällig, da er TCP−Verbindungen niemals abschließt. Außerdem funktioniert er auch bei allen konformen TCP−Stacks und ist unabhängig von spezifischen Eigenarten von Plattformen, wie es bei den FIN−/NULL−/Xmas−, Maimon− und Idle−Scans in Nmap der Fall ist. Er erlaubt auch eine klare, zuverlässige Unterscheidung zwischen den Zuständen offen, geschlossen und gefiltert.

Diese Methode wird oft als halboffenes Scannen bezeichnet, weil keine vollständige TCP−Verbindung hergestellt wird. Sie senden ein SYN−Paket, als ob Sie eine echte Verbindung herstellen würden, und warten dann auf eine Antwort. Ein SYN/ACK zeigt, dass jemand auf dem Port lauscht (dass er offen ist), während ein RST (Reset) anzeigt, dass niemand darauf lauscht. Falls nach mehreren erneuten Übertragungen keine Antwort erhalten wird, wird der Port als gefiltert markiert. Der Port wird auch dann als gefiltert markiert, wenn ein ICMP Unreachable−Fehler (Typ 3, Code 1, 2, 3, 9, 10 oder 13) empfangen wird.

−sT (TCP−Connect−Scan) .
Der TCP−Connect−Scan ist der standardmäßig eingestellte TCP−Scan−Typ, falls der SYN−Scan nicht möglich ist. Das ist dann der Fall, wenn der Benutzer kein Recht hat, rohe Pakete zu senden, oder wenn er IPv6−Netzwerke scannt. Statt rohe Pakete zu schreiben, wie es die meisten anderen Scan−Typen machen, bittet Nmap das darunterliegende Betriebssystem, eine Verbindung mit dem Zielrechner und −port herzustellen, indem es einen Systemaufruf namens **connect** benutzt. Das ist derselbe Systemaufruf auf höherer Ebene, den Webbrowser, P2P−Clients und die meisten anderen netzwerkfähigen Anwendungen benutzen, um eine Verbindung herzustellen. Er ist Teil einer Programmierschnittstelle, die unter dem Namen Berkeley Sockets−API bekannt ist. Statt Antworten in Form roher Pakete von der Leitung zu lesen, benutzt Nmap diese API, um zu jedem Verbindungsversuch eine Statusinformation zu erhalten.

Wenn der SYN−Scan verfügbar ist, ist er normalerweise die bessere Wahl. Nmap hat weniger Einfluss

auf den **connect**−Systemaufruf als auf rohe Pakete, wodurch es weniger effizient wird. Der Systemaufruf beendet Verbindungen zu offenen Ziel−Ports vollständig, statt sie in halboffenen Zustand zurückzusetzen, wie es der SYN−Scan macht. Das dauert nicht nur länger und erfordert mehr Pakete, um an dieselbe Information zu gelangen, sondern es ist sehr viel wahrscheinlicher, dass die Zielrechner die Verbindung protokollieren. Ein anständiges IDS wird beides mitbekommen, aber die meisten Rechner verfügen nicht über ein solches Alarmsystem. Viele Dienste auf Ihrem durchschnittlichen Unix−System fügen eine Notiz ins syslog hinzu und manchmal eine kryptische Fehlermeldung, wenn Nmap eine Verbindung herstellt und sofort wieder schließt, ohne Daten zu senden. Ganz armselige Dienste stürzen auch ab, wenn so etwas passiert, was aber eher selten ist. Ein Administrator, der in seinen Protokollen einen Haufen Verbindungsversuche von einem einzelnen System aus sieht, sollte wissen, dass er Ziel eines Connect−Scans wurde.

−sU (UDP−Scan) .
Obwohl die meisten bekannten Dienste im Internet über das TCP−Protokoll laufen, sind **UDP**[4]−Dienste weitverbreitet. Drei der häufigsten sind DNS, SNMP und DHCP (auf den registrierten Ports 53, 161/162 und 67/68). Weil UDP−Scans im Allgemeinen langsamer und schwieriger als TCP−Scans sind, werden diese Ports von manchen Sicherheitsprüfern einfach ignoriert. Das ist ein Fehler, denn ausbeutbare UDP−Dienste sind recht häufig, und Angreifer ignorieren bestimmt nicht das ganze Protokoll. Zum Glück kann Nmap helfen, diese UDP−Ports zu inventarisieren.

Ein UDP−Scan wird mit der Option **−sU** aktiviert. Er kann mit einem TCP−Scan−Typ wie einem SYN−Scan (**−sS**) kombiniert werden, um beide Protokolle im gleichen Durchlauf zu prüfen.

Beim UDP−Scan wird ein leerer UDP−Header (ohne Daten) an alle Ziel−Ports geschickt. Falls ein ICMP Port−unreachable−Fehler (Typ 3, Code 3) zurückkommt, ist der Port geschlossen. Andere ICMP Unreachable−Fehler (Typ 3, Codes 1, 2, 9, 10 oder 13) markieren den Port als filtered. Gelegentlich wird ein Dienst mit einem UDP−Paket antworten, was beweist, das er offen ist. Falls nach einigen erneuten Übertragungen keine Antwort erhalten wird, wird der Port als offen|gefiltert klassifiziert. Das heißt, der Port könnte offen sein, oder aber es gibt Paketfilter, die die Kommunikation blockieren. Man kann eine Versionserkennung (**−sV**) benutzen, um bei der Unterscheidung der wirklich offenen von den geschlossenen Ports zu helfen.

Eine große Herausforderung beim UDP−Scanning ist Geschwindigkeit. Offene und gefilterte Ports antworten nur selten, wodurch Nmap Zeitbeschränkungen überschreitet und seine Anfragen erneut sendet, für den Fall, dass das Testpaket oder die Antwort verloren ging. Geschlossene Ports sind oftmals ein noch größeres Problem. Sie senden normalerweise eine ICMP Port−unreachable−Fehlermeldung zurück. Aber anders als die RST−Pakete, die von geschlossenen TCP−Ports als Antwort auf einen SYN− oder Connect−Scan geschickt werden, beschränken viele Hosts standardmäßig die Rate. der ICMP Port−unreachable−Nachrichten. Linux und Solaris sind dabei besonders streng. Der Linux−Kernel 2.4.20 z.B. beschränkt Destination−unreachable−Nachrichten auf eine pro Sekunde (in net/ipv4/icmp.c).

Nmap erkennt eine Ratenbeschränkung und verlangsamt seinen Betrieb entsprechend, um zu vermeiden, dass das Netzwerk mit nutzlosen Paketen überflutet wird, die vom Zielrechner verworfen werden. Unglücklicherweise führt eine Beschränkung wie in Linux auf ein Paket pro Sekunde dazu, dass ein Scan von 65.536 Ports über 18 Stunden dauert. Um Ihre UDP−Scans zu beschleunigen, können Sie z.B. mehr Hosts parallel scannen, zuerst nur einen schnellen Scan der beliebten Ports durchführen, von hinter der Firewall scannen und die Option **−−host−timeout** benutzen, um langsame Hosts auszulassen.

−sN; **−sF**; **−sX** (TCP−NULL−, FIN− und −Xmas−Scans) .
Diese drei Scan−Typen (noch mehr sind mit der im nächsten Abschnitt beschriebenen Option **−−scanflags** möglich) beuten ein subtiles Schlupfloch im **TCP RFC**[5] aus, um zwischen offenen und geschlossenen Ports zu unterscheiden. Seite 65 von RFC 793 besagt: “Falls der Zustand des [Ziel−] Ports GESCHLOSSEN ist ... bewirkt ein eingehendes Segment, in dem sich kein RST befindet, dass

ein RST als Antwort gesendet wird.” Die nächste Seite beschreibt dann Pakete, die ohne gesetztes SYN−, RST− oder ACK−Bit an offene Ports geschickt werden, und dort heißt es weiter: “Es ist unwahrscheinlich, dass Sie hierhin kommen, aber wenn doch, dann verwerfen Sie das Segment und springen Sie zurück.”

Beim Scannen von Systemen, die konform zu diesem RFC−Text sind, führt jedes Paket, das kein SYN−, RST− oder ACK−Bit enthält, dazu, dass ein RST zurückgegeben wird, wenn der Port geschlossen ist, bzw. zu gar keiner Antwort, falls der Port offen ist. Solange keines dieser drei Bits gesetzt ist, sind alle Kombinationen der anderen drei (FIN, PSH und URG) okay. Das nutzt Nmap mit drei Scan−Typen aus:

Null−Scan (**−sN**)
Setzt keinerlei Bits (der TCP−Flag−Header ist 0).

FIN−Scan (**−sF**)
Setzt nur das TCP−FIN−Bit.

Xmas−Scan (**−sX**)
Setzt die FIN−, PSH− und URG−Flags und beleuchtet das Paket wie einen Weihnachtsbaum (engl. Xmas).

Diese drei Scan−Typen haben exakt dasselbe Verhalten und unterscheiden sich nur in den TCP−Flags ihrer Testpakete. Wenn ein RST−Paket empfangen wird, wird der Port als geschlossen betrachtet, während keine Antwort bedeutet, dass er offen|gefiltert ist. Der Port wird als gefiltert markiert, falls ein ICMP Unreachable−Fehler (Type 3, Code 1, 2, 3, 9, 10 oder 13) empfangen wird.

Der Schlüsselvorteil dieser Scan−Arten ist, dass sie sich an bestimmten zustandslosen Firewalls und paketfilternden Routern vorbeschleichen können. Ein weiterer Vorteil ist, dass diese Scan−Arten ncoh ein wenig unauffälliger sind als ein SYN−Scan. Aber verlassen Sie sich nicht darauf – die meisten modernen IDS−Produkte können so konfiguriert werden, dass sie diese Scans erkennen. Der große Nachteil ist, dass nicht alle Systeme sich ganz genau an RFC 793 halten. Eine Reihe von Systemen sendet RST−Antworten auf die Testpakete, unabhängig davon, ob der Port offen ist oder nicht. Das bewirkt, dass alle Ports als geschlossen markiert werden. Hauptvertreter der Betriebssysteme, die das machen, sind Microsoft Windows, viele Cisco−Geräte, BSDI und IBM OS/400. Aber auf den meisten Unix−basierten Systemen funktioniert dieser Scan. Ein weiterer Nachteil dieser Scans ist, dass sie keine Unterscheidung zwischen offenen und bestimmten gefilterten Ports machen, sondern lediglich das Ergebnis offen|gefiltert ausgeben.

−sA (TCP−ACK−Scan) .
Dieser Scan unterscheidet sich insofern von den bisher hier vorgestellten, als er nie offene (oder auch nur offene|gefilterte) Ports bestimmt. Er wird dazu benutzt, Firewall−Regeln zu bestimmen, festzustellen, ob sie zustandsbehaftet sind oder nicht, und welche Ports gefiltert werden.

Beim Testpaket eines ACK−Scans wird nur das ACK−Flag gesetzt (es sei denn, Sie benutzen **−−scanflags**). Beim Scannen ungefilterter Systeme werden sowohl offene als auch geschlossene Ports ein RST−Paket zurückgeben. Nmap markiert sie dann als ungefiltert, d.h. sie werden vom ACK−Paket erreicht, aber es kann nicht bestimmt werden, ob sie offen oder geschlossen sind. Ports, die nicht antworten oder bestimmte ICMP−Fehlermeldungen zurückgeben (Type 3, Code 1, 2, 3, 9, 10 oder 13), werden als gefiltert markiert.

−sW (TCP−Window−Scan) .
Der Window−Scan ist genau derselbe wie der ACK−Scan, nur dass er ein Implementationsdetail bestimmter Systeme zur Unterscheidung zwischen offenen und geschlossenen Ports nutzt, statt bei jedem erhaltenen RST immer nur ungefiltert anzugeben. Das geschieht durch Analyse der TCP−Fenstergröße der zurückgegebenen RST−Pakete. Auf manchen Systemen benutzen offene Ports eine positive Fenstergröße (sogar für RST−Pakete), während geschlossene eine Fenstergröße von Null haben. Statt einen Port immer als ungefiltert aufzulisten, wenn von dort ein RST zurückkommt, listet

der Window−Scan den Port als offen oder geschlossen auf, je nachdem, ob die TCP−Fenstergröße in diesem Reset jeweils positiv oder Null ist.

Dieser Scan baut auf einem Implementationsdetail einer Minderheit von Systemen im Internet auf, d.h. Sie können sich nicht immer darauf verlassen. Systeme, die es nicht unterstützen, werden normalerweise alle Ports als geschlossen zurückgeben. Natürlich ist es möglich, dass auf dem Rechner wirklich keine offenen Ports sind. Falls die meisten gescannten Ports geschlossen, aber einige Ports mit geläufigen Nummern (wie 22, 25 und 53) gefiltert sind, dann ist das System sehr wahrscheinlich anfällig. Gelegentlich zeigen Systeme auch genau das gegenteilige Verhalten. Falls Ihr Scan 1000 offene und drei geschlossene oder gefilterte Ports anzeigt, dann könnten jene drei sehr wohl die wirklich wahren offenen Ports sein.

−sM (TCP−Maimon−Scan) .
Der Maimon−Scan wurde nach seinem Erfinder, Uriel Maimon,. benannt. Er hat diese Methode im Phrack−Magazin, Ausgabe #49 (November 1996),. beschrieben. Zwei Ausgaben später war diese Methode in Nmap enthalten. Sie macht genau das Gleiche wie der NULL−, FIN− und Xmas−Scan, außer, dass sie ein FIN/ACK−Testpaket verwendet. Laut **RFC 793**[5] (TCP) sollte als Antwort auf solch ein Paket ein RST−Paket erzeugt werden, egal ob der Port offen oder geschlossen ist. Allerdings hatte Uriel bemerkt, dass viele von BSD abgeleitete Systeme das Paket einfach verwerfen, wenn der Port offen ist.

−−scanflags (Benutzerdefinierter TCP−Scan) .
Wirklich fortgeschrittene Nmap−Benutzer brauchen sich nicht auf die vorgefertigten Scan−Typen zu beschränken. Mit der Option **−−scanflags** können Sie Ihren eigenen Scan entwerfen, für den Sie beliebige TCP−Flags angeben können.. Lassen Sie Ihrer Kreativität freien Lauf und umgehen Sie Intrusion−Detection−Systeme,. deren Hersteller einfach die Nmap−Manpage durchgeblättert und spezifische Regeln dafür angegeben haben!

Das Argument für **−−scanflags** kann ein numerischer Flag−Wert wie z.B. 9 (PSH und FIN) sein, aber symbolische Namen sind einfacher zu benutzen. Erstellen Sie einfach eine beliebige Kombination von URG, ACK, PSH, RST, SYN und FIN. So setzt z.B. **−−scanflags URGACKPSHRSTSYNFIN** alle Flags, auch wenn das beim Scannen nicht besonders hilfreich ist. Die Reihenfolge, in der Sie diese Flags angeben, spielt keine Rolle.

Zusätzlich zu den gewünschten Flags können Sie einen TCP−Scan−Typen (z.B. **−sA** oder **−sF**) angeben. Dieser Basistyp sagt Nmap, wie es die Antworten interpretieren soll. Ein SYN−Scan z.B. betrachtet das Fehlen einer Antwort als einen Hinweis auf einen gefilterten Port, während ein FIN−Scan das als einen Hinweis auf einenoffen|gefilterten Port ansieht. Nmap verhält sich genauso wie beim Scan−Basistyp, nur mit dem Unterschied, dass es die von Ihnen angegebenen TCP−Flags benutzt. Ohne Angabe eines Basistyps wird ein SYN−Scan benutzt.

−sI *zombie host***[:***probeport*] (Idle−Scan) .
Diese fortgechrittene Scan−Methode ermöglicht einen wirklich blinden TCP−Port−Scan des Ziels, d.h. es werden keine Pakete von Ihrer wahren IP−Adresse an das Ziel gesendet. Stattdessen wird mit einem Angriff auf einem parallelen Kanal eine vorhersagbare Erzeugung von Folgen von IP−Fragmentation−IDs auf dem Zombie−Host ausgebeutet, um an Information über offene Ports auf dem Ziel zu gelangen. IDS−Systeme zeigen als Urheber des Scans den Zombie−Rechner an, den Sie angeben (der aktiv sein und einige bestimmte Bedingungen erfüllen muss). Da dieser faszinierende Scan−Typ zu komplex ist, um ihn in diesem Handbuch vollständig zu beschreiben, habe ich einen Artikel mit vollständigen Details dazu geschrieben und unter **https://nmap.org/book/idlescan.html** veröffentlicht.

Dieser Scan−Typ ist nicht nur extrem unauffällig (wegen seiner Blindheit), sondern erlaubt auch, IP−basierte Vetrauensbeziehungen zwischen Rechnern festzustellen. Die Portliste zeigt offene Ports *aus der Sicht des Zombie−Hosts* an. Also können Sie versuchen, ein Ziel mit verschiedenen Zombies zu scannen, von denen Sie denken, dass sie vertrauenswürdig sind. (über Router−/Paketfilterregeln).

Wenn Sie einen bestimmten Port auf dem Zombie auf IP−ID−Änderungen testen möchten, können Sie einen Doppelpunkt gefolgt von einer Portnummer an den Zombie−Host hinzufügen. Sonst benutzt Nmap den Port, den es standardmäßig bei TCP−Pings benutzt (80).

−sO (IP−Protokoll−Scan) .
Der IP−Protokoll−Scan ermöglicht die Bestimmung der IP−Protokolle (TCP, ICMP, IGMP etc.), die von Zielrechnern unterstützt werden. Rein technisch ist das kein Port−Scan, da er über Nummern von IP−Protokollen statt TCP− oder UDP−Ports vorgeht. Dennoch benutzt er die Option **−p** für die Auswahl der zu scannenden Protokollnummern, gibt seine Ergebnisse im normalen Port−Tabellenformat aus und benutzt sogar dieselbe grundlegende Scan−Engine wie die echten Port−Scanning−Methoden. Damit ist er einem Port−Scan ähnlich genug, um an dieser Stelle beschrieben zu werden.

Abgesehen davon, dass er schon als solcher nützlich ist, zeigt der Protokoll−Scan die Macht von Open−Source−Software. Auch wenn die grundlegende Idee recht simpel ist, hatte ich nicht daran gedacht, ihn hinzuzufügen, und bekam auch keine Anfragen nach einer solchen Funktionalität. Dann, im Sommer 2000, hatte Gerhard Rieger. die Idee, schrieb einen exzellenten Patch als Implementation und sendete ihn an die Mailingliste nmap−hackers.. Diesen Patch habe ich in den Nmap−Baum aufgenommen und einen Tag später eine neue Version veröffentlicht. Es gibt nur wenig kommerzielle Software, deren Benutzer so enthusiastisch sind, dass sie eigene Verbesserungen dafür entwerfen und beitragen!

Der Protokoll−Scan funktioniert auf ähnliche Weise wie der UDP−Scan. Statt über das Portnummernfeld eines UDP−Pakets zu iterieren, sendet er IP−Paketheader und iteriert über das acht Bit große IP−Protokollfeld. Die Header sind normalerweise leer, enthalten keine Daten und nicht einmal den richtigen Header für das behauptete Protokoll. Die drei Ausnahmen davon sind TCP, UDP und ICMP. Für diese werden richtige Protokoll−Header verwendet, da manche Systeme sie sonst nicht versenden und weil Nmap bereits über die Funktionen verfügt, um sie zu erzeugen. Statt Nachrichten der Art ICMP Port unreachable sucht der Protokoll−Scan nach ICMP *Protocol* unreachable. Falls Nmap zu irgendeinem Protokoll eine Antwort vom Zielhost erhält, markiert es das Protokoll als offen. Bei einem ICMP Protocol−unreachable−Fehler (Typ 3, Code 2) wird das Protokoll als geschlossen markiert. Bei anderen ICMP Unreachable−Fehlern (Typ 3, Code 1, 3, 9, 10 oder 13) wird das Protokoll als gefiltert markiert (auch wenn sie gleichzeitig beweisen, dass ICMP offen ist). Falls nach einigen erneuten Übertragungen keine Antwort erhalten wird, wird das Protokoll als offen|gefiltert markiert.

−b *FTP relay host* (FTP−Bounce−Scan) .
Eine interessante Eigenschaft des FTP−Protokolls (**RFC 959**[6]) ist dessen Unterstützung sogenannter Proxy−FTP−Verbindungen. Damit kann sich ein Benutzer mit einem FTP−Server verbinden und dann verlangen, dass Dateien an einen Server einer dritten Partei gesendet werden. Solch eine Eigenschaft ist auf vielen Ebenen sturmreif für Missbrauch, weswegen die meisten Server sie nicht mehr unterstützen. Ein solcher Missbrauch, den diese Eigenschaft ermöglicht, ist, den FTP−Server für Port−Scans anderer Hosts zu benutzen. Bitten Sie den FTP−Server einfach, eine Datei nacheinander an alle interessanten Ports eines Zielhosts zu senden. Die Fehlermeldung wird beschreiben, ob der Port offen ist oder nicht. Das ist ein guter Weg, Firewalls zu umgehen, weil FTP−Server von Organisationen oft an Orten platziert sind, von denen aus sie besseren Zugriff auf weitere interne Hosts haben, als es jeder alte Internet−Host hätte. Nmap unterstützt den FTP−Bounce−Scan mit der Option **−b**. Sie erwartet ein Argument der Form *username*:*password@server*:*port*. Dabei ist *Server* der Name oder die IP−Adresse eines anfälligen FTP−Servers. Wie bei einer normalen URL können Sie *username*:*password* auch weglassen, wobei dann eine anonyme Anmeldung erfolgt (username: anonymous password:−wwwuser@). Die Portnummer (samt Doppelpunkt davor) können Sie ebenfalls weglassen, wobei dann auf *server* der Standard−FTP−Port (21) benutzt wird.

Als Nmap 1997 veröffentlicht wurde, war diese Schwachstelle weit verbreitet, wurde seitdem aber größtenteils behoben. Aber da es immer noch anfällige Server gibt, lohnt sich ein Versuch, falls alles andere versagt. Wenn Sie eine Firewall umgehen möchten, scannen Sie das Zielnetzwerk nach einem

offenen Port 21 (oder sogar nach beliebigen FTP−Diensten, falls Sie alle Ports mit Versionserkennung scannen können), und probieren Sie dann für jeden einen Bounce−Scan aus. Nmap wird Ihnen sagen, ob der Host angreifbar ist oder nicht. Versuchen Sie lediglich, Ihre Spuren zu verwischen, dann brauchen Sie sich nicht (und tatsächlich sollten Sie das nicht einmal) auf Hosts im Zielnetzwerk zu beschränken. Bevor Sie anfangen, zufällige Internet−Adressen nach anfälligen FTP−Servern zu scannen, bedenken Sie, dass Sysadmins keinen Gefallen daran finden werden, dass Sie ihre Server auf diese Weise missbrauchen.

PORT-ANGABE UND SCAN-REIHENFOLGE

Zusätzlich zu all den bisher erläuterten Scan−Methoden bietet Nmap Optionen, mit denen man angibt, welche Ports gescannt werden und ob die Scan−Reihenfolge randomisiert oder sequentiell ist. Nmap scannt standardmäßig für jedes Protokoll die 1000 meistbenutzten Ports.

−p *port ranges* (scannt nur angegebene Ports) .

Diese Option gibt an, welche Ports Sie scannen möchten, und überschreibt die Voreinstellung. Einzelne Portnummern sind okay, ebenso wie mit einem Bindestrich getrennte Bereiche (z.B. 1−1023). Anfangs− und/oder Endwerte eines Bereichs können weggelassen werden und werden von Nmap dann mit jeweils 1 bzw. 65535 ersetzt. So können Sie mit **−p−** alle Ports von 1 bis 65535 scannen. Es ist erlaubt, den Port Null. zu scannen, wenn Sie ihn explizit angeben. Bei IP−Protokoll−Scans (**−sO**) gibt diese Option die Protokollnummern an, die Sie scannen möchten (0–255).

Wenn Sie sowohl TCP− als auch UDP−Ports scannen, können Sie ein bestimmtes Protokoll angeben, indem Sie den Portnummern ein T: bzw. U: voranstellen. Dieser Kennzeichner gilt so lange, bis Sie einen anderen angeben. Zum Beispiel werden bei dem Argument **−p U:53,111,137,T:21−25,80,139,8080** die UDP−Ports 53, 111 und 137 sowie die angegebenen TCP−Ports gescannt. Beachten Sie, dass Sie **−sU** und mindestens einen TCP−Scan−Typ (z.B. **−sS**, **−sF** oder **−sT**) angeben müssen, um sowohl UDP als auch TCP zu scannen. Falls kein Protokollkennzeichner angegeben ist, werden die Portnummern zu allen Protokolllisten hinzugefügt. Ports können auch mit dem Namen angegeben werden, der für diesen Port in nmap−services definiert ist. Sie können bei diesen Namen sogar die Joker * und ? verwenden. Um z.B. FTP und alle Ports zu scannen, deren Namen mit “http” anfangen, benutzen Sie **−p ftp,http***. Passen Sie auf eine eventuelle Erweiterung durch die Shell auf und setzen Sie das Argument von **−p** in Anführungszeichen, wenn Sie unsicher sind.

Port−Bereiche können in eckigen Klammern angegeben werden, um Ports innerhalb dieses Bereiches anzugeben, die in nmap−services vorkommen. Zum Beispiel scannt Folgendes alle Ports in nmap−services kleiner oder gleich 1024: **−p [−1024]**. Passen Sie auf eine eventuelle Erweiterung durch die Shell auf und setzen Sie das Argument von **−p** in Anführungszeichen, wenn Sie unsicher sind.

−F (schneller (beschränkter Port−) Scan) .

Gibt an, dass Sie weniger Ports scannen möchten, als standardmäßig vorgesehen. Normalerweise scannt Nmap die 1000 am häufigsten vorkommenden Ports bei jedem gescannten Protokoll. Mit **−F** werden diese auf 100 beschränkt.

Nmap benötigt die Datei nmap−services mit Informationen zur Häufigkeit, um zu wissen, welche Ports am häufigsten benutzt werden. Wenn keine Angaben über die Port−Häufigkeit verfügbar sind, vielleicht weil eine benutzerdefinierte nmap−services−Datei verwendet wird, dann bedeutet **−F**, dass nur Ports gescannt werden, die in der Dienstedatei mit Namen vorkommen (normalerweise scannt Nmap alle benannten Ports plus die Ports 1–1024).

−r (Ports nicht randomisieren) .

Standardmäßig randomisiert Nmap die Reihenfolge der gescannten Ports (bis auf einige allgemein zugängliche Ports, die aus Effizienzgründen vorgezogen werden). Diese Randomisierung ist normalerweise erwünscht, aber Sie können stattdessen auch **−r** für einen sequentiellen Port−Scan angeben.

−−port−ratio <decimal number between 0 and 1>
Scannt alle Ports in der Datei nmap−services mit einem größeren Bruchteil als die Zahl, die als Argument angegeben wird.

−−top−ports <integer of 1 or greater>
Scannt die N Ports mit dem höchsten Bruchteil in der Datei nmap−services.

DIENST- UND VERSIONSERKENNUNG

Lassen Sie Nmap auf einen entfernten Rechner los, und Sie erfahren z.B. dass die Ports 25/tcp, 80/tcp und 53/udp offen sind. Dank der über 2200 bekannten Dienste in seiner Datenbank in nmap−services. würde Nmap noch ausgeben, dass diese Ports wahrscheinlich jeweils zu einem Mailserver (SMTP), Webserver (HTTP) und Nameserver (DNS) gehören. Normalerweise sind diese Angaben genau — die überwiegende Mehrheit an Daemons, die den TCP−Port 25 abhören, sind tatsächlich Mailserver. Allerdings sollten Sie nicht Ihre Sicherheit darauf verwetten! Manche Leute können nicht nur Dienste auf seltsamen Ports betreiben, sondern tun es auch..

Selbst wenn Nmap recht hat und auf dem Server im obigen Beispiel SMTP−, HTTP− und DNS−Server laufen, ist das nicht besonders viel an Information. Bei der Beurteilung der Angreifbarkeit (oder auch nur beim Erstellen einfacher Netzwerkinventare) Ihrer Firmen oder Kunden möchten Sie auch wissen, welche Mail− und DNS−Server und welche Versionen davon laufen. Eine genaue Versionsnummer hilft enorm bei der Bestimmung der Exploits, für die ein Server anfällig ist. Die Versionserkennung hilft Ihnen, an diese Information heranzukommen.

Nachdem TCP− und/oder UDP−Ports mit einer der anderen Scan−Methoden entdeckt wurden, fragt die Versionserkennung diese Ports ab, um mehr darüber zu erfahren, was tatsächlich darauf läuft. Die Datenbank in nmap−service−probes. enthält Testpakete für die Abfrage verschiedenster Dienste und Ausdrücke für den Vergleich und das Parsen der Antworten. Nmap versucht, das Dienstprotokoll zu bestimmen (z.B. FTP, SSH, Telnet, HTTP), aber auch Anwendungsnamen (z.B. ISC BIND, Apache httpd, Solaris telnetd), Versionsnummer, Hostnamen, Gerätetyp (z.B. Drucker, Router), die Betriebssystemfamilie (z.B. Windows, Linux) und manchmal verschiedene Details: etwa ob ein X−Server Verbindungen annimmt, die SSH−Protokollversion oder der KaZaA−Benutzername. Natürlich bieten die meisten Dienste nicht all diese Information. Falls Nmap mit OpenSSL−Unterstützung kompiliert wurde, verbindet es sich mit SSL−Servern, um den hinter dieser Verschlüsselungsebene. lauschenden Dienst zu ermitteln. Wenn RPC−Dienste erkannt werden, wird automatisch Nmaps RPC−Holzhammer. (**−sR**). benutzt, um die RPC−Programm− und Versionsnummern zu bestimmen. Manche UDP−Ports bleiben im Zustand offen|gefiltert, nachdem ein UDP−Port−Scan nicht bestimmen konnte, ob der Port offen oder gefiltert ist. Die Versionserkennung versucht, diesen Ports eine Antwort zu entlocken (genauso wie bei offenen Ports) und den Zustand auf offen zu ändern, wenn das gelingt. Offene|gefilterte TCP−Ports werden genauso behandelt. Beachten Sie, dass die Nmap−Option **−A** unter anderem auch die Versionserkennung einschaltet. A paper documenting the workings, usage, and customization of version detection is available at **https://nmap.org/book/vscan.html**.

Wenn Nmap Antworten von einem Dienst erhält, aber keine Übereinstimmungen dafür in seiner Datenbank finden kann, gibt es einen speziellen Fingerprint und eine URL aus, damit Sie diese Antwort einsenden können, falls Sie genau wissen, was auf diesem Port läuft. Bitte nehmen Sie sich ein paar Minuten Zeit, um sie einzusenden, damit Ihr Fund für alle ein Gewinn sein kann. Dank dieser Beiträge hat Nmap über 3000 Musterübereinstimmungen für über 350 Protokolle wie SMTP, FTP, HTTP usw..

Die Versionserkennung wird mit den folgenden Optionen aktiviert und gesteuert:

−sV (Versionserkennung) .
Aktiviert die Versionserkennung wie oben beschrieben. Alternativ dazu können Sie **−A** benutzen, was unter anderem auch die Versionserkennung aktiviert.

−−allports (keine Ports von der Versionserkennung ausschließen) .
Standardmäßig schließt Nmaps Versionserkennung den TCP−Port 9100 aus, weil manche Drucker einfach alles ausdrucken, was an diesen Port gesendet wird, was zu Dutzenden von Seiten mit HTTP−GET−Requests, binären SSL−Session−Requests usw. führen würde. Dieses Verhalten kann man ändern, indem man die Exclude−Anweisung in nmap−service−probes verändert oder entfernt,

oder Sie geben **−−allports** an, um alle Port zu scannen, unabhängig von einer Exclude−Anweisung.

−−version−intensity *intensity* (Intensität des Versions−Scans setzen) .
Bei einem Versions−Scan (**−sV**) sendet Nmap eine Reihe von Testpaketen, die alle über einen zugeordneten Seltenheitswert zwischen eins und neun verfügen. Die Testpakete mit kleineren Werten sind bei einer großen Zahl verbreiteter Dienste wirkungsvoll, während die mit höheren Werten seltener nützlich sind. Die Intensitätsstufe gibt an, welche Testpakete angewendet werden sollten. Je höher die Zahl, desto wahrscheinlicher wird der Dienst richtig identifiziert. Allerdings brauchen Scans mit hoher Intensität mehr Zeit. Diese Intensität muss zwischen 0 und 9 liegen. Die Standardeinstellung ist 7. Wenn ein Testpaket mit der ports−Anweisung in nmap−service−probes für den Zielport registriert ist, wird dieses Testpaket ausprobiert, unabhängig von der Intensitätsstufe. Das garantiert, dass die DNS−Testpakete bei jedem offenen Port 53 immer benutzt werden, das SSL−Testpaket bei Port 443 usw.

−−version−light (leichten Modus setzen) .
Das ist ein Alias für **−−version−intensity 2** aus Bequemlichkeitsgründen. Dieser leichte Modus macht die Versionserkennung wesentlich schneller, identifiziert die Dienste aber mit geringerer Wahrscheinlichkeit.

−−version−all (benutze alle Testpakete) .
Das ist ein Alias für **−−version−intensity 9**, der garantiert, dass jedes einzelne Testpaket bei jedem Port ausprobiert wird.

−−version−trace (verfolge Aktivität des Versions−Scans) .
Das bewirkt, dass Nmap umfangreiche Debugging−Information darüber ausgibt, was die Versionserkennung gerade macht. Das ist eine Untermenge dessen, was Sie mit **−−packet−trace** erhalten.

−sR (RPC−Scan) .
Diese Methode funktioniert zusammen mit den verschiedenen Port−Scan−Methoden von Nmap. Sie nimmt alle offenen TCP−/UDP−Ports und überflutet sie mit NULL−Befehlen für das SunRPC−Programm, in dem Versuch, festzustellen, ob es RPC−Ports sind, und wenn ja, welches Programm und welche Versionsnummer darauf läuft. Dadurch können Sie quasi dieselbe Information herausfinden wie mit **rpcinfo −p**, selbst wenn der Portmapper des Ziels hinter einer Firewall liegt (oder von TCP−Wrappern geschützt wird). Köder funktionieren im Moment nicht mit dem RPC−Scan.. Das wird automatisch als Teil einer Versionserkennung aktiviert (**−sV**), wenn Sie diese verlangen. Da die Versionserkennung das enthält und wesentlich umfangreicher ist, wird **−sR** selten benötigt.

BETRIEBSSYSTEM-ERKENNUNG

Eines der bekanntesten Merkmale von Nmap ist dessen Erkennung entfernter Betriebssysteme mit TCP/IP−Stack−Fingerprinting. Nmap sendet eine Reihe von TCP− und UDP−Paketen an den entfernten Host und untersucht praktisch jedes Bit in der Antwort. Nach der Durchführung Dutzender von Tests, wie z.B. einer TCP−ISN−Abtastung, Unterstützung und Reihenfolge von TCP−Optionen, IP−ID−Abtastung und Prüfung der initialen Fenstergröße, vergleicht Nmap die Ergebnisse mit seiner Datenbank in nmap−os−db. von über eintausend bekannten Betriebssystem−Fingerprints und gibt die Details zum Betriebssystem aus, wenn es eine Übereinstimmung gefunden hat. Jeder Fingerprint enthält eine formlose Beschreibung des Betriebssystems und eine Klassifikation, aus der der Herstellername (z.B. Sun), das eigentliche Betriebssystem (z.B. Solaris), dessen Generation (z.B. 10) und der Gerätetyp (allgemein, Router, Switch, Spielkonsole usw.) hervorgeht.

Falls Nmap das Betriebssystem eines Rechner nicht erraten kann und die Umstände günstig sind (z.B. wenn mindestens ein offener und ein geschlossener Port gefunden wurde), präsentiert Nmap eine URL, unter der Sie den Fingerprint einsenden können, wenn Sie (ganz sicher) wissen, welches Betriebssystem auf dem Rechner läuft. Dadurch erweitern Sie den Pool der Betriebssysteme, die Nmap kennt, wodurch es für alle Benutzer genauer wird.

Die Betriebssystem−Erkennung verwendet einige weitere Tests, die Informationen benutzen, die während des Vorgangs ohnehin gesammelt werden. Eine davon ist die Klassifikation der Vorhersagbarkeit der

TCP−Sequenznummern. Sie gibt ungefähr an, wie schwer es ist, eine gefälschte TCP−Verbindung zum entfernten Host aufzubauen. Sie ist hilfreich zur Ausbeutung von Vertrauensbeziehungen auf Basis von Quell−IPs (rlogin, Firewall−Filter usw.) oder zum Verbergen des Ursprungs eines Angriffs. Diese Art von Täuschung wird kaum noch praktiziert, aber viele Rechner sind nach wie vor anfällig dafür. Die eigentliche Maßzahl basiert auf statistischen Abtastungen und kann schwanken. Im Allgemeinen ist es besser, die englische Bezeichnung wie z.B. “worthy challenge” oder “trivial joke” zu benutzen. Das wird nur in der normalen Ausgabe im ausführlichen Modus (**−v**) ausgegeben. Wenn dieser Modus zusammen mit **−O** aktiviert ist, wird auch die IP−ID−Sequenzerzeugung berichtet. Die meisten Rechner finden sich in der Klasse “incremental”, d.h. sie inkrementieren das ID−Feld im IP−Header für jedes von ihnen gesendete Paket. Dadurch werden sie anfällig für diverse avancierte Angriffe mittels Täuschung und Sammlung von Informationen.

Eine weitere zusätzliche Information, die die Betriebssystem−Erkennung aktiviert, ist eine Schätzung der Betriebszeit des Zielhosts. Dabei wird die TCP−Timestamp−Option (**RFC 1323**[7]) benutzt, um zu raten, wann ein Rechner das letzte Mal neu gestartet wurde. Wenn der Timestamp−Zähler nicht bei null gestartet wurde oder der Zähler überläuft und wieder von vorn zählt, kann diese Schätzung ungenau werden, weshalb sie nur im ausführlichen Modus ausgegeben wird.

A paper documenting the workings, usage, and customization of OS detection is available at **https://nmap.org/book/osdetect.html**.

Eingeschaltet und gesteuert wird die Betriebssystem−Erkennung mit den folgenden Optionen:

−O (Betriebssystem−Erkennung aktivieren) .
Aktiviert die Betriebssystem−Erkennung wie oben beschrieben. Alternativ dazu können Sie **−A** benutzen, um eine Betriebssystem−Erkennung zusammen mit anderen Dingen einzuschalten.

−−osscan−limit (Betriebssystem−Erkennung auf vielversprechende Ziele beschränken) .
Die Betriebssystem−Erkennung ist wesentlich effektiver, wenn mindestens ein offener und ein geschlossener TCP−Port gefunden werden. Wenn Sie diese Option angeben, versucht Nmap eine Betriebssystem−Erkennung gar nicht erst bei Hosts, die dieses Kriterium nicht erfüllen. Das kann viel Zeit sparen, besonders bei Scans mit **−PN** auf vielen Hosts. Diese Option gilt nur, wenn eine Betriebssystem−Erkennung mit **−O** oder **−A** verlangt wird.

−−osscan−guess; **−−fuzzy** (Ergebnisse der Betriebssystem−Erkennung raten) .
Falls Nmap keine perfekte Übereinstimmung mit einem Betriebssystem finden kann, präsentiert es manchmal mögliche Kandidaten, die dem sehr nahe kommen. Damit Nmap das standardmäßig macht, muss die Übereinstimmung sehr hoch sein. Diese beiden (äquivalenten) Optionen lassen Nmap aggressiver schätzen. Dann gibt Nmap auch unvollkommene Übereinstimmungen aus, zusammen mit einem Vertrauensgrad (in Prozent).

−−max−os−tries (setzt die maximale Anzahl der Versuche für eine Betriebssystem−Erkennung bei einem Ziel) .
Wenn Nmap eine Betriebssystem−Erkennung auf einem Ziel durchführt und dafür keine perfekte Übereinstimmung findet, macht es normalerweise einen weiteren Versuch. Standardmäßig macht Nmap fünf Versuche, wenn die Bedingungen zum Einsenden eines Betriebssystem−Fingerprints günstig sind, und zwei, wenn sie es nicht sind. Die Angabe eines kleineren Wertes für **−−max−os−tries** (z.B. 1) beschleunigt Nmap, auch wenn Sie Versuche auslassen, bei denen das Betriebssystem möglicherweise erkannt werden könnte. Alternativ dazu kann ein hoher Wert gesetzt werden, um bei günstigen Bedinungen noch mehr Versuche zu erlauben. Das macht man aber selten, außer um bessere Fingerprints zu erzeugen, die man einsenden kann, um sie in Nmaps Betriebssystem−Datenbank zu integrieren.

NMAP SCRIPTING ENGINE (NSE)

Die Nmap Scripting Engine (NSE) ist eines der mächtigsten und flexibelsten Merkmale von Nmap. Mit ihr können Benutzer einfache Scripts schreiben und weitergeben (geschrieben in der **Programmiersprache Lua**[8].

Zu den Aufgaben, die wir bei der Konzeption dieses Systems anvisierten, gehören die Netzwerkerkennung, eine ausgefeiltere Versionserkennung sowie eine Verwundbarkeitserkennung. NSE kann aber sogar bei der

Ausbeutung von Schwachstellen benutzt werden.

Um diese verschiedenen Anwendungen widerzuspiegeln und um die Auswahl des richtigen Scripts zu vereinfachen, verfügen alle Scripts über ein Kategorie−Feld. Im Moment sind folgende Kategorien definiert: safe, intrusive, malware, version, discovery, vuln, auth und default. Sie alle werden at **https://nmap.org/book/nse-usage.html#nse-categories**.

Die Nmap Scripting Engine wird detailliert at **https://nmap.org/book/nse.html** beschrieben und wird mit den folgenden Optionen gesteuert:

−sC .
Führt einen Script−Scan mit dem Standardsatz an Scripts durch. Das ist äquivalent zu **−−script=default**. Manche der Scripts in dieser Kategorie werden als aufdringlich betrachtet und sollten nicht ohne Genehmigung auf einem Zielnetzwerk ausgeführt werden.

−−script *script−categories*|*directory*|*filename*|**all**.
Führt einen Script−Scan (wie z.B. **−sC**) durch und benutzt dabei die mit Kommata getrennte Liste von Script−Kategorien, individuellen Scripts oder Script−Verzeichnissen statt des Standardsatzes. Zuerst versucht Nmap, die Argumente als Kategorien zu interpretieren, dann (wenn das nicht gelingt) als Datei− oder Verzeichnisnamen. Ein Script oder Verzeichnis von Scripts kann als absoluter oder relativer Pfad angegeben werden. Absolute Pfade werden unverändert benutzt. Relative Pfade werden an den folgenden Orten gesucht, bis sie gefunden werden:. $NMAPDIR/;. (wird unter Windows nicht durchsucht);. scripts/−Unterverzeichnis ausprobiert.

Falls ein Verzeichnis angegeben und gefunden wird, lädt Nmap alle NSE−Scripts (alle Dateinamen, die mit .nse enden) aus diesem Verzeichnis. Dateinamen ohne die Endung nse werden ignoriert. Nmap sucht keine Unterverzeichnisse rekursiv durch, um Scripts zu finden. Wenn individuelle Dateinamen angegeben werden, dann muss deren Dateierweiterung nicht nse lauten.

Nmap−Scripts werden standardmäßig in einem scripts−Unterverzeichnis des Nmap−Datenverzeichnisses gespeichert (see **https://nmap.org/book/data-files.html**). Aus Effizienzgründen werden Scripts in einer Datenbank indiziert, die in scripts/script.db.. gespeichert ist und für jedes Script auflistet, in welche Kategorie bzw. Kategorien es gehört. Um alle Scripts in der Nmap−Script−Datenbank auszuführen, geben Sie das Argument all an.

Die Scripts werden nicht in einer Sandbox ausgeführt und können Ihr System somit versehentlich oder böswillig beschädigen oder in Ihre Privatsphäre eindringen. Sie sollten Scripts von Dritten nur dann ausführen, wenn Sie deren Autoren vertrauen oder sie selbst eingehend studiert haben.

−−script−args *name1=value1*,*name2=*{*name3=value3*},*name4=value4* .
Hiermit können Sie Argumente für NSE−Scripts angeben. Argumente werden als name=value−Paare angegeben. Die Argumente werden verarbeitet und in einer Lua−Tabelle gespeichert, auf die alle Scripts Zugriff haben. Die Namen werden als Strings (die alphanumerische Werte sein müssen) in argument−table als Schlüssel gespeichert. Die Werte sind ihrerseits entweder Strings oder Tabellen (innerhalb von ‘{’ und ‘}’). Sie könnten z.B. diese mit Kommmata getrennten Argumente angeben: user=bar,pass=foo,whois={whodb=nofollow+ripe}. String−Argumente werden potenziell von mehreren Scripts benutzt, während Untertabellen normalerweise nur von einem Script benutzt werden. In Scripts, die eine Untertabelle annehmen, wird diese Untertabelle normalerweise nach dem Script benannt (in diesem Fall z.B. whois).

−−script−trace .
Diese Option macht das, was **−−packet−trace** macht, aber eine ISO−Ebene höher. Wenn diese Option angegeben wird, wird die gesamte ein− und ausgehende Kommunikation von Scripts ausgegeben. Die angezeigte Information enthält das Kommunikationsprotokoll, Quell− und Zieladressen sowie die übertragenen Daten. Falls mehr als fünf Prozent der übertragenen Daten nicht druckbar sind, werden sie stattdessen als Hexadezimal−Auszug ausgegeben. Auch die Angabe von **−−packet−trace** schaltet diese Mitverfolgung von Scripts ein.

−−script−updatedb .

Diese Option aktualisiert die Script−Datenbank in scripts/script.db, die von Nmap benutzt wird, um die verfügbaren Standard−Scripts und Kategorien zu bestimmen. Man muss die Datenbank nur dann aktualisieren, wenn man NSE−Scripts in das Standardverzeichnis scripts hinzufügt oder von dort entfernt, oder wenn man die Kategorie eines Scripts ändert. Diese Option wird ohne Argumente benutzt: **nmap −−script−updatedb**.

TIMING UND PERFORMANCE

Bei der Entwicklung von Nmap hatte dessen Performance immer eine der höchsten Prioritäten. Ein Standardscan (**nmap** *hostname*) eines Hosts in meinem lokalen Netzwerk dauert eine Fünftelsekunde. In dieser Zeit kann man kaum blinzeln, aber wenn man Hunderte oder Tausende von Rechnern scannt, dann kommt einiges zusammen. Außerdem können bestimmte Scan−Optionen wie UDP−Scanning und eine Versionserkennung die Scan−Zeiten erheblich erhöhen. Das Gleiche gilt für bestimmte Firewall−Konfigurationen und besonders für die Beschränkung der Antwortrate. Auch wenn Nmap parallel arbeitet und viele ausgefeilte Algorithmen benutzt, um diese Scans zu beschleunigen, hat doch der Benutzer die letzte Kontrolle darüber, was Nmap genau macht. Experten erstellen ihre Nmap−Befehle sehr sorgfältig, um in einer beschränkten Zeit genau die gewünschte Information zu bekommen.

Um die Scan−Zeiten zu verbessern, kann man z.B. nicht−kritische Tests weglassen und auf die neueste Version von Nmap aktualisieren (Performance−Verbesserungen werden häufig gemacht). Auch die Optimierung von Timing−Parametern kann einen großen Unterschied ausmachen. Diese Optionen werden im Folgenden beschrieben.

Manche Optionen erwarten einen time−Parameter. Dieser wird standardmäßig in Millisekunden angegeben, aber Sie können ‘s’, ‘m’ oder ‘h’ an den Wert anhängen, um Sekunden, Minuten oder Stunden anzugeben. Das heißt, bei **−−host−timeout** haben die Argumente 900000, 900s und 15m alle denselben Effekt.

−−min−hostgroup *numhosts*; **−−max−hostgroup** *numhosts* (Größe der parallel gescannten Gruppen anpassen) .

Nmap hat die Fähigkeit, Port−Scans oder Versions−Scans von mehreren Hosts parallel durchzuführen. Das macht Nmap, indem es den Ziel−IP−Adressraum in Gruppen aufteilt und dann jeweils eine Gruppe scannt. Im Allgemeinen sind größere Gruppen effizienter. Der Nachteil daran ist, dass die Host−Ergebnisse erst dann ausgegeben werden können, wenn die gesamte Gruppe fertig ist. Wenn Nmap mit einer Gruppengröße von 50 anfängt, würde der Benutzer erst dann Ergebnisse sehen (bis auf die Aktualisierungen im ausführlichen Modus), wenn die ersten 50 Hosts abgearbeitet sind.

Diesen Konflikt löst Nmap standardmäßig mit einem Kompromissansatz. Es fängt mit einer kleinen Gruppengröße von etwa fünf an, damit die ersten Ergebnisse schnell kommen, und erhöht dann die Gruppengröße bis auf etwa 1024. Die genau vorgegebenen Zahlen hängen von den benutzten Optionen ab. Aus Effizienzgründen benutzt Nmap größere Gruppen bei UDP−Scans oder bei TCP−Scans über wenige Ports.

Falls mit **−−max−hostgroup** eine maximale Gruppengröße angegeben wird, wird Nmap diese nie überschreiten. Und wenn Sie mit **−−min−hostgroup** eine minimale Größe angeben, versucht Nmap, die Gruppengröße oberhalb dieses Wertes zu belassen. Falls es auf einer gegebenen Schnittstelle nicht genug Zielhosts gibt, um dieses Minimum zu erfüllen, muss Nmap einen kleineren Wert benutzen. Es können auch beide gesetzt werden, um die Gruppengröße in einem bestimmten Bereich zu belassen, aber das ist selten gewünscht.

Diese Optionen haben während der Host−Entdeckungsphase eines Scans keinen Effekt. Dazu gehören einfache Ping−Scans (**−sP**). Die Host−Entdeckung funktioniert immer in großen Gruppen von Hosts, um die Geschwindigkeit und Genauigkeit zu verbessern.

Der Hauptnutzen dieser Optionen liegt darin, eine hohe minimale Gruppengröße anzugeben, damit der gesamte Scan schneller läuft. Häufig wird 256 gewählt, um ein Netzwerk in Brocken der Größe von Klasse C zu scannen. Bei einem Scan mit vielen Ports bringt eine größere Zahl vermutlich keine Vorteile. Bei Scans über nur wenige Ports können Gruppengrößen von 2048 oder mehr hilfreich sein.

−−min−parallelism *numprobes*; **−−max−parallelism** *numprobes* (Parallelität von Testpaketen anpassen) .
Diese Optionen steuern die Gesamtanzahl an Testpaketen, die für eine Host−Gruppe anstehen dürfen. Sie werden beim Port−Scanning und bei der Host−Entdeckung benutzt. Abhängig von der Netzwerk−Performance berechnet Nmap standardmäßig eine immer wechselnde ideale Parallelität. Falls Pakete verworfen werden, verlangsamt sich Nmap und erlaubt weniger unerledigte Testpakete. Die ideale Anzahl von Testpaketen steigt mit den zunehmenden Möglichkeiten des Netzwerks. Diese Optionen setzen minimale oder maximale Schranken für diese Variable. Standardmäßig kann die ideale Parallelität auf eins sinken, wenn sich das Netzwerk als unzuverlässig herausstellt, und im Idealfall kann sie auf mehrere hundert steigen.

Meistens setzt man **−−min−parallelism** auf eine Zahl größer als eins, um Scans von langsamen Hosts oder Netzwerken zu beschleunigen. Aber das Spielen mit dieser Option kann gefährlich sein, weil die Genaugkeit leiden kann, wenn man einen zu großen Wert setzt. Dabei verringert sich auch Nmaps Möglichkeit, die Parallelität je nach Netzwerkbedingungen dynamisch anzupassen. Ein Wert von zehn mag vernünftig sein, auch wenn ich nur als letzter Ausweg an diesem Wert drehe.

Die Option **−−max−parallelism** wird manchmal auf eins gesetzt, um zu verhindern, dass Nmap mehr als ein Testpaket auf einmal an Hosts sendet. In Kombination mit **−−scan−delay** (wird später behandelt) kann das nützlich sein, auch wenn Letzteres diesen Zweck gut genug allein erfüllt.

−−min−rtt−timeout *time*, **−−max−rtt−timeout** *time*, **−−initial−rtt−timeout** *time* (Timeouts von Testpaketen anpassen) .
Nmap verwaltet einen laufenden Timeout−Wert, der bestimmt, wie lange es auf eine Antwort zu einem Testpaket wartet, bevor es aufgibt oder das Paket erneut sendet. Dieser wird auf der Grundlage der Antwortzeiten bei früheren Testpaketen berechnet.

Falls die Netzwerk−Latenzzeit sich als groß genug und variabel erweist, kann dieser Timeout bis auf mehrere Sekunden wachsen. Er beginnt auch bei einem konservativen (hohen) Wert und kann diesen eine Weile behalten, wenn Nmap Hosts scannt, die nicht antworten.

Werden Werte für **−−max−rtt−timeout** und **−−initial−rtt−timeout** angegeben, die kleiner als deren Standardwerte sind, so kann die Scan−Zeit erheblich verkürzt werden. Das gilt besonders für pinglose (**−PN**) Scans und solche von stark gefilterten Netzwerken. Aber verlangen Sie nicht zu viel. Der Scan kann am Ende länger brauchen, wenn Sie einen so kleinen Wert angeben, dass bei vielen Testpaketen der Timeout erreicht wird und sie erneut gesendet werden, während die Antwort unterwegs ist.

Falls alle Hosts in einem lokalen Netzwerk sind, sind 100 Millisekunden ein vernünftig aggressiver Wert für **−−max−rtt−timeout**. Falls ein Routing ins Spiel kommt, pingen Sie zuerst einen Host im Netzwerk, sei es mit einem ICMP−Ping oder mit einem Programm zur Erstellung benutzerdefinierter Pakete wie **hping2**,. das eine höhere Chance hat, durch eine Firewall zu kommen. Betrachten Sie dann die maximale Umlaufzeit bei circa zehn Paketen. Diese möchten Sie vielleicht für **−−initial−rtt−timeout** verdoppeln und für **−−max−rtt−timeout** verdrei− oder vervierfachen. Im Allgemeinen setze ich die maximale RTT nicht unter 100 ms, egal, welche Ping−Zeiten ich habe. Und ich gehe auch nicht über 1000 ms.

Die Option **−−min−rtt−timeout** wird selten benutzt, könnte aber nützlich sein, wenn ein Netzwerk so unzuverlässig ist, dass selbst Nmaps Standardeinstellung zu aggressiv ist. Da Nmap das Timeout nur dann auf das Minimum reduziert, wenn das Netzwerk zuverlässig scheint, sollte ein Bedarf hierfür eher ungewöhnlich sein und sollte als Fehler auf der nmap−dev−Mailingliste. berichtet werden.

−−max−retries *numtries* (gibt die maximale Anzahl erneuter Sendeversuche bei Port−Scan−Testpaketen an) .
Falls Nmap keine Antwort auf ein Testpaket eines Port−Scans erhält, könnte das heißen, dass der Port gefiltert ist. Oder aber das Testpaket oder die Antwort darauf ging im Netzwerk verloren. Es ist auch möglich, dass der Zielhost eine Ratenbeschränkung aktiviert hat, die die Antwort temporär blockiert hat. Also versucht es Nmap erneut, indem es das ursprüngliche Paket noch einmal sendet. Falls Nmap

eine mangelnde Netzwerkzuverlässigkeit feststellt, führt es eventuell viele weitere Wiederholungen durch, bevor es bei einem Port aufgibt. Das verbessert zwar die Genauigkeit, verlängert aber auch die Scan−Zeiten. Wenn es auf die Performance ankommt, kann man die Scans durch eine Beschränkung der Anzahl dieser Wiederholungen beschleunigen. Sie können sogar **−−max−retries 0** angeben, um sie ganz zu verbieten, auch wenn sich das nur in Situationen wie formlosen Überprüfungen empfiehlt, bei denen einige verpasste Ports oder Hosts nicht so wichtig sind.

Der Standardwert (ohne **−T**−Template) sind bis zu zehn Wiederholungen. Falls das Netzwerk zuverlässig zu sein scheint und die Zielhosts keine Ratenbeschränkung haben, führt Nmap normalerweise nur eine Wiederholung durch. Daher sind die meisten Scans gar nicht betroffen, wenn man **−−max−retries** z.B. auf den kleinen Wert drei verringert. Solche Werte können Scans von langsamen (ratenbeschränkten) Hosts aber erheblich beschleunigen. Wenn Nmap frühzeitig bei Ports aufgibt, verlieren Sie eventuell einiges an Information, aber das kann vorteilhafter sein, als ein Timeout von **−−host−timeout** zu erreichen und alle Informationen über das Ziel zu verlieren.

−−host−timeout *time* (bei langsamen Zielhosts aufgeben) .
Bei manchen Hosts braucht es einfach *lange*, sie zu scannen. Das kann an leistungsschwacher oder unzuverlässiger Netzwerk−Hardware oder −Software, an einer Paketratenbeschränkung oder einer restriktiven Firewall liegen. Die langsamsten paar Prozent der gescannten Hosts können einen Großteil der Scan−Zeit verbrauchen. Dann ist es manchmal das Beste, die Verluste abzuschneiden und diese Hosts erst einmal wegzulassen. Geben Sie **−−host−timeout** mit der gewünschten maximalen Wartezeit an. Sie können z.B. 30m angeben, um sicherzustellen, dass Nmap nicht mehr als eine halbe Stunde verschwendet, indem es auf einen einzelnen Host wartet. Beachten Sie, dass Nmap während dieser halben Stunde auch andere Hosts scannen kann, d.h. es ist keine komplette Zeitverschwendung. Ein Host, der das Timeout erreicht, wird übersprungen. Für diesen Host werden keine Ergebnisse der Port−Tabelle, Betriebssystem− oder Versionserkennung ausgegeben.

−−scan−delay *time*; **−−max−scan−delay** *time* (Verzögerung zwischen Testpaketen anpassen) .
Diese Option bewirkt, dass Nmap mindestens die angegebene Zeit zwischen zwei Testpaketen an einen Host wartet. Das ist besonders bei einer Ratenbeschränkung. sinnvoll. Solaris−Rechner (und viele andere auch) antworten auf UDP−Scan−Testpakete normalerweise nur mit einer ICMP−Nachricht pro Sekunde. Wenn Nmap mehr sendet, ist das Verschwendung. Mit einem **−−scan−delay** von 1s bleibt Nmap bei dieser langsamen Rate. Nmap versucht eine Ratenbeschränkung zu erkennen und die Scan−Verzögerung entsprechend anzupassen, aber es schadet nicht, sie explizit anzugeben, falls Sie schon wissen, welche Rate am besten funktioniert.

Wenn Nmap die Scan−Verzögerung nach oben anpasst, um mit der Ratenbeschränkung klarzukommen, verlangsamt sich der Scan dramatisch. Die Option **−−max−scan−delay** gibt die größte Verzögerung an, die Nmap erlaubt. Ein kleiner Wert für **−−max−scan−delay** kann Nmap beschleunigen, ist aber riskant. Ein zu kleiner Wert kann zu verschwenderischen wiederholten Sendungen führen und möglicherweise zu verpassten Ports, wenn das Ziel eine strenge Ratenbeschränkung implementiert.

Ein weiterer Einsatz von **−−scan−delay** liegt bei der Umgehung schwellwertbasierter Intrusion−Detection− und −Prevention−Systeme (IDS/IPS)..

−−min−rate *number*; **−−max−rate** *number* (direkte Steuerung der Scan−Rate) .
Nmaps dynamisches Timing findet sehr gut die passende Scan−Geschwindigkeit. Aber manchmal kennen Sie vielleicht die passende Scan−Rate für ein Netzwerk, oder vielleicht müssen Sie garantieren, dass ein Scan bis zu einem bestimmten Zeitpunkt fertig wird. Oder Sie müssen Nmap vielleicht davon abhalten, zu schnell zu scannen. Für diese Situationen sind die Optionen **−−min−rate** und **−−max−rate** vorgesehen.

Bei der Option **−−min−rate** versucht Nmap sein Bestes, um Pakete so schnell wie die damit angegebene Rate zu senden oder noch schneller. Das Argument ist eine positive Fießkommazahl, die die Paketrate in Paketen pro Sekunde angibt. Die Angabe **−−min−rate 300** bedeutet z.B., dass Nmap

eine Rate von 300 Paketen pro Sekunde oder höher einzuhalten versucht. Die Angabe einer Minimalrate hält Nmap nicht davon ab, bei günstigen Bedingungen schneller zu werden.

Umgekehrt beschränkt **−−max−rate** die Senderate auf das angegebene Maximum. Bei **−−max−rate 100** wird sie auf 100 Pakete pro Sekunde bei einem schnellen Netzwerk beschränkt. Und bei **−−max−rate 0.1** wird der Scan auf ein Paket pro zehn Sekunden verlangsamt. Mit **−−min−rate** und **−−max−rate** gleichzeitig können Sie die Rate in einem bestimmten Bereich halten.

Diese beiden Optionen sind global und betreffen den gesamten Scan, nicht nur einzelne Hosts. Sie betreffen nur Port−Scans und Host−Entdeckungs−Scans. Andere Merkmale wie die Betriebssystemerkennung implementieren ihr eigenes Timing.

Unter zwei Bedingungen kann die tatsächliche Scan−Rate unter das verlangte Minimum fallen. Die erste ist, wenn das Minimum schneller als die schnellste Rate ist, mit der Nmap senden kann, was hardwareabhängig ist. In diesem Fall sendet Nmap Pakete einfach so schnell wie möglich, aber Sie sollten wissen, dass solch hohe Raten sehr wahrscheinlich einen Verlust an Genauigkeit bedeuten. Die zweite Bedingung ist, wenn Nmap nichts zu senden hat, z.B. am Ende eines Scans, nachdem die letzten Testpakete gesendet wurden und Nmap darauf wartet, dass sie einen Timeout verursachen oder beantwortet werden. Eine sinkende Scan−Rate am Ende eines Scans oder zwischen Gruppen von Hosts ist normal. Die Senderate kann temporär das Maximum übersteigen, um unvorhergesehene Verzögerungen auszugleichen, aber im Durchschnitt bleibt die Rate bei oder unter dem Maximum.

Eine minimale Rate sollte man mit Vorsicht angeben. Scans, die schneller sind, als das Netzwerk erlaubt, können zu einem Verlust von Genauigkeit führen. In manchen Fällen kann die Wahl eines schnelleren Scans dazu führen, dass er *länger* braucht als bei einer kleineren Rate. Das liegt daran, dass Nmaps Algorithmen für die adaptive retransmission die von einer zu hohen Scan−Rate verursachte Netzwerküberlastung erkennen und die Anzahl der Wiederholungen erhöhen, um die Genauigkeit zu verbessern. Das heißt, selbst wenn Pakete mit höherer Rate gesendet werden, werden mehr Pakete insgesamt gesendet. Begrenzen Sie diese Anzahl von Wiederholungen nach oben mit der Option **−−max−retries**, wenn Sie eine obere Grenze für die gesamte Scan−Zeit setzen müssen.

−−defeat−rst−ratelimit .
Viele Hosts haben lange eine Ratenbeschränkung. benutzt, um die Anzahl der von ihnen gesendeten ICMP−Fehlermeldungen (z.B. Port−Unreachable−Fehler) zu reduzieren. Manche Systeme wenden nun ähnliche Ratenbeschränkungen auf die von ihnen erzeugten RST−(Reset−)Pakete an. Das kann Nmap dramatisch verlangsamen, weil es sein Timing an diese Ratenbeschränkungen anpasst. Mit der Option **−−defeat−rst−ratelimit** können Sie Nmap sagen, dass es diese Ratenbeschränkungen ignorieren soll (z.B. bei Port−Scans wie dem SYN−Scan, die nicht−antwortende Ports *nicht* als offen behandeln).

Diese Option kann die Genauigkeit reduzieren, da einige Ports nicht zu antworten scheinen, weil Nmap nicht lange genug auf eine ratenbeschränkte RST−Antwort gewartet hat. Bei einem SYN−Scan führt die ausbleibende Antwort dazu, dass für den Port der Zustand gefiltert und geschlossen markiert wird, den wir sehen, wenn RST−Pakete empfangen werden. Diese Option ist nützlich, wenn Sie sich nur für offene Ports interessieren und eine Unterscheidung zwischen geschlossenen und gefilterten Ports die zusätzliche Zeit nicht wert ist.

−T paranoid|sneaky|polite|normal|aggressive|insane (setzt ein Timing−Template) .
Auch wenn die feinkörnigen Timing−Einstellungen im letzten Abschnitt mächtig und effektiv sind, finden manche Leute sie verwirrend. Außerdem kann die Wahl der passenden Werte manchmal mehr Zeit erfordern als der Scan, den Sie damit optimieren möchten. Deswegen bietet Nmap auch einen einfacheren Ansatz mit sechs Timing−Templates. Diese können Sie mit der Option **−T** und ihrer Nummer (0–5) oder mit ihrem Namen angeben. Diese Template−Namen lauten: **paranoid** (**0**), **sneaky** (**1**), **polite** (**2**), **normal** (**3**), **aggressive** (**4**) und **insane** (**5**). Die ersten beiden sind für die Umgehung von IDS gedacht. Der Polite−Modus verlangsamt den Scan, damit er weniger Bandbreite und Ressourcen auf dem Zielrechner verbraucht. Der Normal−Modus ist der Standardwert, d.h. **−T3**

macht gar nichts. Der Aggressive−Modus beschleunigt Scans, indem er davon ausgeht, dass Sie sich in einem einigermaßen schnellen und zuverlässigen Netzwerk befinden. Und schließlich geht der Insane−Modus. davon aus, dass sie sich in einem außergewöhnlich schnellen Netzwerk befinden bzw. gewillt sind, für mehr Geschwindigkeit auf etwas Genauigkeit zu verzichten.

Mit diesen Templates können Benutzer angeben, wie aggressiv sie vorgehen möchten, und trotzdem Nmap die Wahl der genauen Zeitwerte überlassen. Diese Templates führen außerdem auch kleine Geschwindigkeitsanpassungen aus, für die es zur Zeit keine feinkörnigen Einstellungsoptionen gibt. Zum Beispiel verbietet **−T4**,. dass die dynamische Scan−Verzögerung bei TCP−Ports über 10 ms ansteigt, und **−T5** begrenzt diesen Wert auf 5 ms. Templates könen auch in Kombination mit feinkörnigen Einstellungen benutzt werden, wobei die feinkörnigeren Optionen die entsprechenden allgemeinen Werte aus den Templates überschreiben. Ich empfehle den Einsatz von **−T4** beim Scannen halbwegs moderner und zuverlässiger Netzwerke. Wenn Sie diese Option angeben, auch dann, wenn Sie zusätzliche feinkörnige Einstellungen benutzen, profitieren Sie von den weiteren kleinen Optimierungen, die damit verbunden sind.

Falls Sie eine anständige Breitband− oder Ethernet−Verbindung haben, würde ich empfehlen, immer **−T4** zu benutzen. Manche Leute lieben **−T5**, was für meinen Geschmack aber zu aggressiv ist. Manchmal geben Leute **−T2** an, weil sie denken, dadurch würden Hosts weniger abstürzen, oder weil sie sich selbst im Allgemeinen für höflich halten. Oft realisieren sie einfach nicht, wie langsam **−T polite**. tatsächlich ist. Ein solcher Scan kann zehnmal mehr Zeit benötigen als ein Standard−Scan. Rechnerabstürze und Bandbreitenprobleme kommen mit den standardmäßigen Timing−Optionen (**−T3**) selten vor, weswegen ich das normalerweise vorsichtigen Scannern empfehle. Auf die Versionserkennung zu verzichten ist weit wirksamer bei der Reduktion dieser Probleme als das Herumprobieren mit Zeiteinstellungen.

Zwar sind **−T0**. und **−T1**. vielleicht hilfreich bei der Vermeidung von IDS−Alarmen, benötigen aber außergewöhnlich viel Zeit beim Scannen von Tausenden von Rechnern oder Ports. Für einen derart langen Scan möchten Sie vielleicht doch lieber die genau benötigten Zeitwerte angeben, statt sich auf die vorgefertigten Werte von **−T0** und **−T1** zu verlassen.

Die Haupteffekte von **T0** sind die Serialisierung des Scans, bei der immer nur ein Port gescannt wird, und eine Wartezeit von fünf Minuten zwischen zwei Testpaketen. **T1** und **T2** sind ähnlich, warten aber jeweils nur 15 bzw. 0,4 Sekunden zwischen zwei Testpaketen. **T3** ist die Standardeinstellung in Nmap, die eine Parallelisierung. beinhaltet. **−T4** macht das Äquivalent von **−−max−rtt−timeout 1250 −−initial−rtt−timeout 500 −−max−retries 6** und setzt die maximale TCP−Scan−Verzögerung auf 10 Millisekunden. **T5** macht das Äquivalent von **−−max−rtt−timeout 300 −−min−rtt−timeout 50 −−initial−rtt−timeout 250 −−max−retries 2 −−host−timeout 15m** und setzt die maximale TCP−Scan−Verzögerung auf 5 ms.

FIREWALL-/IDS-UMGEHUNG UND -TäUSCHUNG

Viele Internet−Pioniere hatten die Vision eines globalen, offenen Netzwerks, in dem ein universeller IP−Adressraum virtuelle Verbindungen zwischen zwei beliebigen Knoten erlaubt. Dadurch können Hosts als echte, gleichberechtigte Partner agieren und Information untereinander senden und empfangen. Die Menschen könnten von ihrer Arbeitsstelle auf all ihre Systeme daheim zugreifen, die Einstellungen der Klimaanlage ändern oder die Türen für verfrühte Gäste aufsperren. Diese Vision einer universellen Konnektivität wurde durch eine Verknappung im Adressraum und Sicherheitsbedenken abgewürgt. In den frühen 1990er Jahren begannen Organisationen mit der Aufstellung von Firewalls mit dem ausdrücklichen Zweck einer Reduktion der Konnektivität. Riesige Netzwerke wurden mit Anwendungs−Proxies, NAT (Network Address Translation)−Geräten und Paketfiltern vom ungefilterten Internet abgeriegelt. Der ungehinderte Fluss von Informationen hat einer strengen Regulierung von zugelassenen Kommunikationskanälen und der darüber ausgetauschten Inhalte Platz gemacht.

Netzwerkhindernisse wie Firewalls können die Analyse eines Netzwerks außerordentlich schwer machen. Und leichter wird es nicht werden, da das Verhindern von Ausspähungen oft ein Schlüsselziel beim Einsatz dieser Geräte ist. Trotzdem bietet Nmap viele Eigenschaften, um beim Verständnis dieser komplexen

Netzwerke zu helfen und um zu überprüfen, dass diese Filter arbeiten wie gewünscht. Es bietet sogar Mechanismen zur Umgehung schlechter Abwehrstrategien. Eine der besten Methoden, Ihre Lage in puncto Netzwerksicherheit zu verstehen, ist die, sie anzugreifen. Versetzen Sie sich selbst in die Denkweise eines Angreifers und wenden Sie Verfahren aus diesem Kapitel gegen Ihr Netzwerk an. Starten Sie einen FTP−Bounce−Scan, Idle−Scan, Fragmentierungsangriff, oder versuchen Sie durch einen Ihrer eigenen Proxies zu tunneln.

Zusätzlich zur Beschränkung der Netzwerkaktivität überwachen Firmen ihren Datenverkehr immer mehr mit Intrusion−Detection−Systemen (IDS). Alle wichtigen IDS werden mit Regeln ausgeliefert, die entworfen wurden, um Nmap−Scans zu erkennen, weil Scans manchmal Vorboten von Angriffen sind. Viele dieser Produkte haben sich in Intrusion−*Prevention*−Systeme (IPS). verwandelt, die für böswillig gehaltenen Datenverkehr aktiv blockieren. Dummerweise ist es für Netzwerkadministratoren und IDS−Hersteller eine sehr schwierige Aufgabe, böswillige Absichten durch die Analyse von Paketdaten zuverlässig zu erkennen. Angreifer mit Geduld, Geschick und der Hilfe bestimmter Nmap−Optionen können meist unerkannt an einem IDS vorbeikommen. Währenddessen müssen Administratoren mit riesigen Mengen falscher positiver Ergebnisse kämpfen, bei denen eine nicht böswillige Aktivität fehldiagnostiziert wird und Alarm schlägt oder blockiert wird.

Ab und zu schlagen Leute vor, dass Nmap keine Eigenschaften für die Umgehung von Firewallregeln oder IDS enthalten sollte. Ihr Argument ist, dass diese Eigenschaften genauso wahrscheinlich von Angreifern missbraucht werden wie von Administratoren, die die Sicherheit verbessern. Das Problem bei dieser Logik ist, dass diese Methoden trotzdem von Angreifern benutzt würden, die einfach andere Werkzeuge finden oder die Funktionalität in Nmap einbauen würden. Zugleich wäre es für Administratoren sehr viel schwerer, ihren Job zu machen. Das Aufstellen nur moderner, gepatchter FTP−Server ist eine wesentlich bessere Verteidigung als der Versuch, die Verbreitung von Werkzeugen zu verhindern, die einen FTP−Bounce−Angriff implementieren.

Es gibt keine Wunderlösung (oder Nmap−Option) zur Erkennung und Umgehung von Firewalls und IDS−Systemen. Es braucht Kompetenz und Erfahrung. Eine Anleitung dazu würde den Rahmen dieses Referenz−Handbuches sprengen, das nur die wichtigsten Optionen auflistet und beschreibt, was sie machen.

−f (Pakete fragmentieren); **−−mtu** (benutzt angegebene MTU) .

Die Option **−f** bewirkt, dass der gewünschte Scan (inklusive Ping−Scans) winzig fragmentierte IP−Pakete benutzt. Die Idee dabei ist, den TCP−Header über mehrere Pakete aufzuteilen, um es Paketfiltern, Intrusion−Detection−Systemen und anderen Ärgernissen schwerer zu machen, Ihre Aktivitäten zu durchschauen. Seien Sie dabei vorsichtig! Manche Programme haben Mühe, mit diesen winzigen Paketen umzugehen. Ein Sniffer alter Schule namens Sniffit ist beim Erhalt des ersten Fragments sofort mit einem Segmentation−Fault−Fehler abgestürzt. Wenn Sie diese Option einmal angeben, spaltet Nmap die Pakete in acht Bytes oder weniger nach dem IP−Header auf. Das heißt, ein 20 Byte langer TCP−Header würde in drei Pakete aufgeteilt, zwei mit acht Bytes des TCP−Headers und eines mit den restlichen vier. Natürlich hat jedes Fragment auch einen IP−Header. Geben Sie erneut **−f** an, um 16 Bytes pro Fragment zu benutzen (was die Anzahl der Fragmente verkleinert).. Oder Sie geben eine eigene Offset−Größe mit der Option **−−mtu** (für engl. maximum transmission unit) an. Wenn Sie **−−mtu** angeben, sollten Sie nicht auch **−f** angeben. Das Offset muss ein Vielfaches von acht sein. Zwar kommen fragmentierte Pakete nicht durch Paketfilter und Firewalls durch, die alle IP−Fragmente in eine Warteschlange stellen, wie z.B. die Option *CONFIG_IP_ALWAYS_DEFRAG* im Linux−Kernel, aber einige Netzwerke können sich den damit verbundenen Performance−Einbruch nicht leisten und lassen sie folglich deaktiviert. Andere können sie nicht aktivieren, weil die Fragmente auf verschiedenen Routen in ihre Netzwerke kommen könnten. Manche Quellsysteme defragmentieren hinausgehende Pakete im Kernel. Ein Beispiel dafür ist Linux mit dem Verbindungsmodul iptables.. Führen Sie einen Scan aus, während ein Sniffer wie z.B. Wireshark. läuft, um sicherzustellen, dass die gesendeten Pakete fragmentiert sind. Falls Ihr Host−Betriebssystem Probleme macht, probieren Sie die Option **−−send−eth**. aus, um die IP−Schicht zu umgehen und rohe Ethernet−Rahmen zu schicken.

Eine Fragmentierung wird von Nmap nur für rohe Pakete unterstützt, die man mit TCP− und

UDP−Port−Scans (außer beim Connect−Scan und FTP−Bounce−Scan) und der Betriebssystemerkennung benutzen kann. Merkmale wie die Versionserkennung und die Nmap Scripting Engine unterstützen im Allgemeinen keine Fragmentierung, weil sie sich auf den TCP−Stack Ihres Hosts verlassen, um mit anderen Zielen zu kommunizieren.

−D *decoy1***[,***decoy2***][,ME][,...]** (verdeckt einen Scan mit Ködern) .
Führt einen Decoy−Scan durch, was für den entfernten Host den Anschein erweckt, dass der oder die Hosts, die Sie als Köder angeben, das Zielnetzwerk ebenfalls scannen. Deren IDS kann also 5–10 Port−Scans von eindeutigen IP−Adressen verzeichnen, aber es weiß nicht, welche IP sie gescannt hat und welche unschuldige Köder waren. Das kann man zwar bekämpfen, indem man Router−Pfade mitverfolgt, Antworten verwirft oder weitere aktive Mechanismen anwendet, aber im Allgemeinen ist es eine wirksame Methode zum Verbergen Ihrer IP−Adresse.

Trennen Sie alle Köder mit Kommata voneinander, wobei Sie optional ME. als einen der Köder angeben können, um die Position Ihrer echten IP−Adresse zu bestimmen. Falls Sie ME an sechster Stelle oder später setzen, zeigen einige verbreitete Port−Scan−Detektoren (wie z.B. der hervorragende Scanlogd. von Solar Designer. ME angeben, setzt es Nmap an eine zufällig gewählte Position. Sie können auch RND. benutzen, um eine zufällige, nicht−reservierte IP−Adresse zu erzeugen, oder RND:*number*, um *number* Adressen zu erzeugen.

Beachten Sie, dass die Hosts, die Sie als Köder benutzen, eingeschaltet sein sollten, sonst könnten Sie versehentlich einen SYN−Flood−Angriff auf Ihre Ziele auslösen. Außerdem lässt sich der scannende Host sehr einfach bestimmen, wenn nur einer davon im Netzwerk eingeschaltet ist. Vielleicht möchten Sie IP−Adressen statt −Namen benutzen (damit die Köder−Netzwerke Sie nicht in ihren Nameserver−Protokollen sehen).

Köder werden sowohl im initialen Ping−Scan (mit ICMP, SYN, ACK oder was auch immer) als auch während der eigentlichen Port−Scan−Phase benutzt. Auch bei der Erkennung entfernter Betriebssysteme (**−O**) werden Köder benutzt. Bei der Versionserkennung oder beim TCP−Connect−Scan funktionieren Köder jedoch nicht. Falls eine Scan−Verzögerung stattfindet, wird sie zwischen zwei Stapeln vorgetäuschter Testpakete erzwungen, nicht zwischen einzelnen Testpaketen. Weil Köder stapelweise auf einmal gesendet werden, können sie vorübergehend die Beschränkungen der Überlastungssteuerung verletzen.

Man sollte hierbei noch erwähnen, dass beim Einsatz von zu vielen Ködern Ihr Scan sich verlangsamen und sogar ungenauer werden kann. Manche ISPs filtern außerdem Ihre vorgetäuschten Pakete, aber viele beschränken solche vorgetäuschten IP−Pakete in keinster Weise.

−S *IP_Address* (Quelladresse vortäuschen) .
Unter gewissen Umständen kann Nmap eventuell Ihre Quelladresse nicht bestimmen (wenn dem so ist, dann sagt Ihnen Nmap Bescheid). Benutzen Sie in diesem Fall **−S** mit der IP−Adresse der Schnittstelle, über die Sie die Pakete senden möchten.

Eine weitere mögliche Anwendung dieses Flags ist eine Vortäuschung des Scans, um die Ziele glauben zu machen, dass *jemand anderes* sie scannt. Stellen Sie sich eine Firma vor, die wiederholt von einem Mitbewerber gescannt wird! Bei dieser Art von Anwendung werden im Allgemeinen die Optionen **−e** und **−PN** benötigt. Beachten Sie, dass Sie normalerweise Antwortpakete zurückbekommen (sie werden an die IP adressiert, die Sie vortäuschen), d.h. Nmap kann keinen sinnvollen Bericht produzieren.

−e *interface* (angegebene Schnittstelle benutzen) .
Sagt Nmap, auf welcher Schnittstelle es Pakete senden und empfangen soll. Nmap sollte das automatisch erkennen können, sagt Ihnen aber Bescheid, wenn nicht.

−−source−port *portnumber***; −g** *portnumber* (Quell−Portnummer vortäuschen) .
Eine Fehlkonfiguration, die überraschend häufig vorkommt, ist es, dem Netzwerkverkehr allein auf Basis der Quell−Portnummer zu vertrauen. Wie das zustande kommt, kann man leicht verstehen. Ein

Administrator setzt eine glänzende neue Firewall auf und wird sofort mit Beschwerden von undankbaren Benutzern überflutet, deren Anwendungen nicht mehr laufen. Vor allem DNS könnte einen Aussetzer haben, weil die UDP−DNS−Antworten von externen Servern nicht länger ins Netzwerk hineinkommen. Ein weiteres häufiges Beispiel ist FTP. Bei aktiven FTP−Übertragungen versucht der entfernte Server, eine Verbindung zurück zum Client herzustellen, um die gewünschte Datei zu übertragen.

Für diese Probleme existieren sichere Lösungen, oftmals in Form von Proxies auf Anwendungsebene oder Protokoll−parsenden Firewall−Modulen. Leider gibt es auch einfachere, unsichere Lösungen. Viele Administratoren haben beobachtet, dass DNS−Antworten von Port 53 und aktive FTP−Antworten von Port 20 kommen, und sind in die Falle getappt, eingehenden Datenverkehr nur von diesen Ports zu erlauben. Oft gehen sie davon aus, dass kein Angreifer solche Firewall−Lecks bemerken und ausbeuten würde. In anderen Fällen betrachten das Administratoren als kurzfristige Überbrückungsmaßnahme, bis sie eine sicherere Lösung implementieren können. Und dann vergessen sie diese Sicherheitsaktualisierung.

Aber nicht nur überarbeitete Netzwerkadministratoren tappen in diese Falle. Zahlreiche Produkte wurden mit diesen unsicheren Regeln ausgeliefert. Sogar Microsoft hat sich schuldig gmacht. Die IPsec−Filter, die mit Windows 2000 und Windows XP ausgeliefert wurden, enthalten eine implizite Regel, die jeden TCP− oder UDP−Datenverkehr von Port 88 (Kerberos) erlaubt. Ein weiterer bekannter Fall sind Versionen der Zone Alarm Personal−Firewall bis 2.1.25, die alle empfangenen UDP−Pakete vom Quell−Port 53 (DNS) oder 67 (DHCP) erlauben.

Nmap bietet die Optionen **−g** und **−−source−port** (sind äquivalent), um diese Schwächen auszunutzen. Geben Sie einfach eine Portnummer an, und Nmap wird, wenn möglich, Pakete von diesem Port senden. Damit es richtig funktioniert, muss Nmap für bestimmte Betriebssystemerkennungstests verschiedene Portnummern benutzen, und DNS−Anfragen ignorieren das **−−source−port**−Flag, weil Nmap sich bei ihnen auf System−Bibliotheken verlässt. Die meisten TCP−Scans, inklusive dem SYN−Scan, unterstützen die Option vollständig, ebenso wie der UDP−Scan.

−−data−length *number* (Zufallsdaten an gesendete Pakete anfügen) .
Normalerweise sendet Nmap minimale Pakete, die nur einen Header enthalten. Daher haben seine TCP−Pakete im Allgemeinen nur 40 Bytes und die ICMP Echo−Requests nur 28. Mit dieser Option sagen Sie Nmap, dass es die angegebene Anzahl von zufälligen Bytes an die meisten gesendeten Pakete hinzufügen soll. Pakete für die Betriebssystemerkennung (**−O**) sind davon nicht betroffen,. weil dort aus Genauigkeitsgründen konsistente Pakete verlangt werden, aber die meisten Ping− und Port−Scan−Pakete unterstützen das. Das kann den Scan etwas verlangsamen, aber auch etwas unauffälliger machen.

−−ip−options *S|R [route]|L [route]|T|U ...* **;** **−−ip−options** *hex string* (sendet Pakete mit angegebenen IP−Optionen) .
Laut **IP−Protokoll**[9] können in den Paket−Headern mehrere Optionen enthalten sein. Anders als die allgegenwärtigen TCP−Optionen sieht man IP−Optionen aus Gründen der praktischen Anwendbarkeit und Sicherheit nur selten. Tatsächlich blockieren die meisten Internet−Router die gefährlichsten Optionen wie Source Routing sogar. Dennoch können diese Optionen in manchen Fällen nützlich sein, um die Netzwerk−Route zu Zielrechnern zu bestimmen und zu manipulieren. Sie können z.B. vielleicht die Option Record Route dazu benutzen, einen Pfad zum Ziel sogar dann zu bestimmen, wenn traditionellere, traceroute−artige Ansätze versagen. Oder wenn Ihre Pakete von einer bestimmten Firewall verworfen werden, können Sie mit den Optionen Strict oder Loose Source Routing möglicherweise eine andere Route angeben.

Die meisten Möglichkeiten bei der Angabe von IP−Optionen hat man, wenn man einfach Werte als Argumente für **−−ip−options** angibt. Stellen Sie vor jede Hex−Zahl ein \x und zwei Ziffern. Einzelne Zeichen können Sie wiederholen, indem Sie ihnen ein Sternchen und dann die Anzahl der Wiederholungen nachstellen. So ist z.B. \x01\x07\x04\x00*36\x01 ein Hex−String mit 36

NUL−Bytes.

Nmap bietet auch einen verkürzten Mechanismus für die Angabe von Optionen. Geben Sie einfach die Buchstaben R, T oder U an, um jeweils Record Route,. Record Timestamp. oder beide Optionen gemeinsam anzugeben. Loose oder Strict Source Routing. kann man mit L bzw. S, gefolgt von einem Leerzeichen und einer mit Leerzeichen getrennten Liste von IP−Adressen angeben.

Wenn Sie die Optionen in den gesendeten und empfangenen Paketen sehen möchten, geben Sie **−−packet−trace** an. Mehr Informationen und Beispiele zum Einsatz von IP−Optionen mit Nmap finden Sie unter **http://seclists.org/nmap-dev/2006/q3/0052.html**.

−−ttl *value* (setzt IP−Time−to−live−Feld) .
Setzt bei IPv4 das Time−to−live−Feld in gesendeten Paketen auf den angegebenen Wert.

−−randomize−hosts (randomisiert Reihenfolge der Zielhosts) .
Verlangt von Nmap, dass es alle Gruppen von bis zu 16.384 Hosts durcheinanderwürfelt, bevor es sie scannt. Das kann den Scan für verschiedene Netzwerk−Überwachungssysteme weniger offensichtlich machen, besonders dann, wenn Sie ihn mit einer langsamen Timing−Option kombinieren. Wenn Sie größere Gruppen randomisieren möchten, müssen Sie *PING_GROUP_SZ*. in nmap.h. erhöhen und neu kompilieren. Eine alternative Lösung ist es, die Liste der Ziel−IPs mit einem List−Scan (**−sL −n −oN** *filename*) zu erzeugen, dann z.B. mit einem Perl−Script zu randomisieren, um sie schließlich als Ganzes mit **−iL**. an Nmap zu übergeben.

−−spoof−mac *MAC address, prefix, or vendor name* (MAC−Adresse vortäuschen) .
Verlangt von Nmap, dass es in allen gesendeten rohen Ethernet−Rahmen die angegebene MAC−Adresse. benutzt. Diese Option impliziert **−−send−eth**,. um sicherzustellen, dass Nmap tatsächlich Pakete auf Ethernet−Ebene sendet. Die MAC−Adresse kann in mehreren Formaten angegeben werden. Wenn es einfach die Zahl 0 ist, wählt Nmap eine völlig zufällige MAC−Adresse für diese Sitzung. Falls der angegebene String aus einer geraden Anzahl von Hexadezimalziffern besteht (in dem Paare optional mit Doppelpunkten getrennt sein können), benutzt Nmap diese als MAC. Werden weniger als 12 Hexadezimalziffern angegeben, dann füllt Nmap die restlichen sechs Bytes mit zufälligen Werten. Falls das Argument weder null noch ein Hex−String ist, schaut Nmap in nmap−mac−prefixes nach, um einen Herstellernamen zu finden, der den angegebenen String enthält (unabhängig von der Schreibweise). Wird eine Übereinstimmung gefunden, benutzt Nmap die OUI dieses Herstellers (einen drei Byte langen Präfix). und füllt die verbleibenden drei Bytes mit Zufallswerten. Gültige Beispiele für Argumente von **−−spoof−mac** sind Apple, 0, 01:02:03:04:05:06, deadbeefcafe, 0020F2 und Cisco. Diese Option betrifft nur Scans mit rohen Paketen wie den SYN−Scan oder die Betriebssystemerkennung, keine verbindungsorientierten Merkmale wie die Versionserkennung oder die Nmap Scripting Engine.

−−badsum (sendet Pakete mit falschen TCP/UDP−Prüfsummen) .
Verlangt von Nmap, bei den an Zielhosts gesendeten Paketen ungültige TCP− oder UDP−Prüfsummen zu benutzen. Da so gut wie alle Host−IP−Stacks solche Pakete verwerfen, kommen eventuelle Antworten sehr wahrscheinlich von einer Firewall oder einem IDS, das sich nicht die Mühe macht, die Prüfsumme zu überprüfen. Mehr Details zu dieser Methode finden Sie unter **https://nmap.org/p60-12.html**.

AUSGABE

Alle Sicherheitswerkzeuge sind nur so gut wie die Ausgabe, die sie erzeugen. Komplexe Tests und Algorithmen haben einen geringen Wert, wenn sie nicht auf übersichtliche und verständliche Weise dargestellt werden. Da Nmap auf vielfältige Weise von verschiedenen Leuten und anderer Software benutzt wird, kann kein Format allein es allen recht machen. Daher bietet Nmap mehrere Formate, darunter den interaktiven Modus, den Menschen direkt lesen können, und XML, das von Software leicht geparst werden kann.

Zusätzlich zu verschiedenen Ausgabeformaten bietet Nmap Optionen zur Steuerung der Ausführlichkeit dieser Ausgabe sowie Debugging−Meldungen. Die Ausgaben können an die Standardausgabe oder an benannte Dateien gehen, die Nmap überschreiben bzw. an die es seine Ausgabe anfügen kann. Mit den

Ausgabedateien können außerdem abgebrochene Scans fortgesetzt werden.

Nmap erzeugt seine Ausgabe in fünf verschiedenen Formaten. Das Standardformat heißt interaktive Ausgabe. und wird an die Standardausgabe (stdout). gesendet. Es gibt auch die normale Ausgabe,. die ähnlich zur interaktiven Ausgabe ist, außer dass sie weniger Laufzeitinformation und Warnungen ausgibt, weil man davon ausgeht, dass sie erst nach Abschluss des Scans analysiert wird und nicht, während er noch läuft.

Die XML−Ausgabe. ist eines der wichtigsten Ausgabeformate, da sie einfach nach HTML konvertiert, von Programmen wie Nmap−GUIs geparst oder in Datenbanken importiert werden kann.

Die zwei verbleibenden Ausgabeformate sind die einfache grepbare Ausgabe,. in der die meiste Information über einen Zielhost in einer einzigen Zeile enthalten ist, und sCRiPt KiDDi3 0utPUt. für Benutzer, die sich selbst als |<−r4d sehen.

Die interaktive Ausgabe ist standardmäßig vorgegeben und verfügt über keine eigenen Kommandozeilenoptionen, aber die anderen vier Formate benutzen dieselbe Syntax. Sie erwarten ein Argument, den Namen der Datei, in der die Ergebnisse gespeichert werden sollen. Es können mehrere Formate angegeben werden, aber jedes nur einmal. Vielleicht möchten Sie z.B. eine normale Ausgabe für eine eigene Untersuchung speichern und eine XML−Ausgabe desselben Scans für eine programmbasierte Analyse. Das erreichen Sie mit den Optionen **−oX myscan.xml −oN myscan.nmap**. Auch wenn in diesem Kapitel der Kürze wegen einfache Namen wie myscan.xml benutzt werden, empfehlen sich im Allgemeinen aussagekräftigere Namen. Welche Namen Sie wählen, ist Geschmackssache, aber ich benutze lange Namen, die das Scandatum und ein oder zwei Worte über den Scan enthalten, in einem Verzeichnis, das den Namen der gescannten Firma enthält.

Auch wenn diese Optionen Ergebnisse in Dateien speichern, gibt Nmap weiterhin die interaktive Ausgabe wie üblich auf die Standardausgabe aus. Zum Beispiel speichert der Befehl **nmap −oX myscan.xml target** XML in myscan.xml und füllt die Standardausgabe mit demselben interaktiven Ergebnis, wie es auch ohne Angabe von **−oX** der Fall wäre. Das können Sie ändern, indem Sie ein Minuszeichen als Argument für eines der Formate angeben. Dann schaltet Nmap die interaktive Ausgabe ab und gibt stattdessen Ergebnisse im gewünschten Format auf den Standardausgabestrom aus. Das heißt, der Befehl **nmap −oX − target** schreibt nur die XML−Ausgabe auf die Standardausgabe.. Ernste Fehler werden weiterhin auf den normalen Standardfehlerstrom, stderr,. ausgegeben.

Anders als bei anderen Nmap−Argumenten ist das Leerzeichen zwischen dem Options−Flag für eine Ausgabedatei (z.B. **−oX**) und dem Dateinamen oder Minuszeichen obligatorisch. Falls Sie die Leerzeichen weglassen und Argumente wie z.B. **−oG−** oder **−oXscan.xml** angeben, erzeugt Nmap aus Gründen der Rückwärtskompatibilität Ausgabedateien im *normalen Format*, die jeweils die Namen G− und Xscan.xml haben.

All diese Argumente unterstützen **strftime**−ähnliche. Umwandlungen im Dateinamen. %H, %M, %S, %m, %d, %y und %Y sind alle exakt gleich wie in **strftime**. %T entspricht %H%M%S, %R entspricht %H%M und %D entspricht %m%d%y. Ein %, dem ein anderes Zeichen folgt, ergibt nur genau dieses Zeichen (%% ergibt ein Prozentzeichen). Also erzeugt **−oX 'scan−%T−%D.xml'** eine XML−Datei in der Form scan−144840−121307.xml.

Nmap bietet auch Optionen zur Steuerung der Scan−Ausführlichkeit und Optionen, um an Ausgabedateien anzuhängen, statt sie zu überschreiben. All diese Optionen werden unten beschrieben.

Nmap−Ausgabeformate

−oN *filespec* (normale Ausgabe) .
Verlangt, dass eine normale Ausgabe in der angegebenen Datei gespeichert wird. Wie oben erwähnt, unterscheidet sich das leicht von der interaktiven Ausgabe.

−oX *filespec* (XML−Ausgabe) .
Verlangt, dass eine XML−Ausgabe in der angegebenen Datei gespeichert wird. Nmap fügt eine DTD (Document Type Definition) hinzu, mit der XML−Parser Nmaps XML−Ausgabe validieren können. Diese ist vor allem für die Benutzung durch Programme gedacht, kann aber auch Menschen bei der Interpretation von Nmaps XML−Ausgabe helfen. Die DTD definiert die gültigen Elemente des

Formats und zählt an vielen Stellen die dafür erlaubten Attribute und Werte auf. Die neueste Version ist immer unter **https://nmap.org/data/nmap.dtd** verfügbar.

XML bietet ein stabiles Format, das man mit Software leicht parsen kann. Solche XML−Parser sind für alle wichtigen Programmiersprachen wie C/C++, Perl, Python und Java gratis verfügbar. Manche Leute haben sogar Anbindungen für die meisten dieser Sprachen geschrieben, um speziell die Ausgabe und Ausführung von Nmap zu steuern. Beispiele sind **Nmap::Scanner**[10]. und **Nmap::Parser**[11]. für Perl in CPAN. In fast allen Fällen, in denen eine nicht−triviale Anwendung eine Schnittstelle zu Nmap benutzt, ist XML das bevorzugte Format.

Die XML−Ausgabe verweist auf ein XSL−Stylesheet, mit dem man die Ergebnisse als HTML formatieren kann. Am einfachsten benutzt man das, indem man einfach die XML−Ausgabe in einem Webbrowser wie Firefox oder IE lädt. Standardmäßig funktioniert das nur auf dem Rechner, auf dem Sie Nmap ausgeführt haben (oder auf einem, der ähnlich konfiguriert ist), weil der Pfad zu nmap.xsl darin festkodiert ist. Um portable XML−Dateien zu erzeugen, die auf allen mit dem Web verbundenen Rechnern als HTML angezeigt werden, können Sie die Optionen **−−webxml** oder **−−stylesheet** benutzen.

−oS *filespec* (ScRipT KIdd|3−Ausgabe) .
Die Script−Kiddie−Ausgabe ist ähnlich zur interaktiven Ausgabe, mit dem Unterschied, dass sie nachbearbeitet ist, um die 'l33t HaXXorZ besser anzusprechen! Vorher haben sie wegen dessen konsistent richtiger Schreibweise und Buchstabierung auf Nmap herabgesehen. Humorlose Menschen sollten wissen, dass diese Option sich über Script Kiddies lustig macht, bevor sie mich dafür angreifen, dass ich “ihnen helfe”.

−oG *filespec* (grepbare Ausgabe) .
Dieses Ausgabeformat wird zum Schluss beschrieben, weil es als überholt gilt. Das XML−Ausgabeformat ist wesentlich leistungsstärker und für erfahrene Benutzer fast genauso bequem. XML ist eim Standard, für den Dutzende hervorragender Parser verfügbar sind, während die grepbare Ausgabe nur mein eigener einfacher Hack ist. XML ist erweiterbar und kann neue Nmap−Eigenschaften unterstützen, die ich beim grepbaren Format aus Platzgründen oft weglassen muss.

Dessen ungeachtet ist die grepbare Ausgabe immer noch recht beliebt. Es ist ein einfaches Format, das pro Zeile einen Host auflistet und das mit Unix−Standardwerkzeugen wie grep, awk, cut, sed, diff und auch mit Perl auf triviale Weise durchsucht und geparst werden kann. Selbst ich benutze es für einmalige schnelle Tests in der Kommandozeile. Zum Beispiel kann man alle Hosts, auf denen der SSH−Port offen ist oder auf denen Solaris läuft, auf einfache Weise mit einem grep bestimmen, das die Hosts findet, umgeleitet in einen awk− oder cut−Befehl, der die gewünschten Felder ausgibt.

Die grepbare Ausgabe besteht aus Kommentaren (Zeilen, die mit einem # anfangen). sowie aus Zielzeilen. Eine Zielzeile enthält eine Kombination aus sechs benannten Feldern, durch Tabulatoren getrennt, gefolgt von einem Doppelpunkt. Diese Felder lauten Host, Ports, Protocols, Ignored State, OS, Seq Index, IP ID und Status.

Das wichtigste dieser Felder ist im Allgemeinen Ports, das Details zu einem interessanten Port enthält. Es ist eine mit Kommata getrennte Liste von Port−Einträgen, wobei jeder Eintrag einen interessanten Port darstellt und aus sieben mit Schrägstrichen (/) getrennten Unterfeldern besteht. Diese Unterfelder lauten: Port number, State, Protocol, Owner, Service, SunRPC info und Version info.

Wie bei der XML−Ausgabe kann diese Manpage auch hier nicht das vollständige Format dokumentieren. Eine detailliertere Betrachtung des grepbaren Ausgabeformats in Nmap finden Sie from **https://nmap.org/book/output-formats-grepable-output.html**.

−oA *basename* (Ausgabe in allen Formaten) .
Aus Gründen der Bequemlichkeit können Sie Scan−Ergebnisse mit **−oA** *basename* gleichzeitig in

normalem, in XML− und in grepbarem Format speichern. Sie werden jeweils in *basename*.nmap, *basename*.xml und *basename*.gnmap, gespeichert. Wie in den meisten Programmen können Sie vor den Dateinamen ein Präfix mit einem Verzeichnispfad darin setzen, z.B. ˜/nmaplogs/foocorp/ unter Unix oder c:\hacking\sco unter Windows.

Optionen für Ausführlichkeit und Debugging

−v (größere Ausführlichkeit) .
Erhöht die Ausführlichkeit, d.h. Nmap gibt mehr Informationen über den laufenden Scan aus. Offene Ports werden angezeigt, direkt nachdem sie gefunden werden, und es werden Schätzungen für die Dauer bis zur Fertigstellung angegeben, falls Nmap meint, dass ein Scan mehr als ein paar Minuten benötigt. Noch mehr Information erhalten Sie, wenn Sie diese Option zweimal oder noch öfter angeben.

Die meisten Änderungen betreffen nur die interaktive Ausgabe, manche betreffen auch die normale und die Script−Kiddie−Ausgabe. Die anderen Ausgabearten sind für die Weiterverarbeitung durch Maschinen gedacht, d.h. Nmap kann in diesen Formaten standardmäßig alle Details angeben, ohne einen menschlichen Leser zu ermüden. Allerdings gibt es in den anderen Modi einige Änderungen, bei denen die Ausgabegröße durch Weglassen einiger Details erheblich reduziert werden kann. Zum Beispiel wird eine Kommentarzeile in der grepbaren Ausgabe, die eine Liste aller gescannten Ports enthält, nur im wortreichen Modus ausgegeben, weil sie ziemlich lang werden kann.

−d [level] (erhöhe oder setze Debugging−Stufe) .
Wenn nicht einmal der wortreiche Modus genug Daten für Sie liefert, können Sie beim Debugging noch wesentlich mehr davon bekommen! Wie bei der Ausführlichkeits−Option (**−v**) wird auch das Debugging mit einem Kommandozeilen−Flag eingeschaltet (**−d**), und die Debug−Stufe kann durch eine mehrfache Angabe gesteigert werden.. Alternativ dazu können Sie eine Debug−Stufe auch als Argument an **−d** übergeben. So setzt z.B. **−d9** die Stufe neun. Das ist die höchste verfügbare Stufe, die Tausende von Zeilen produziert, sofern Sie keinen sehr einfachen Scan mit sehr wenigen Ports und Zielen ausführen.

Eine Debugging−Ausgabe ist sinnvoll, wenn Sie einen Fehler in Nmap vermuten oder wenn Sie einfach verwirrt darüber sind, was und warum Nmap etwas genau macht. Da dieses Merkmal überwiegend für Entwickler gedacht ist, sind Debug−Zeilen nicht immer selbsterklärend. Vielleicht bekommen Sie etwas wie: Timeout vals: srtt: −1 rttvar: −1 to: 1000000 delta 14987 ==> srtt: 14987 rttvar: 14987 to: 100000. Wenn Sie eine Zeile nicht verstehen, ist Ihre einzige Zuflucht, sie zu ignorieren, im Quellcode nachzuschauen oder Hilfe auf der Entwicklerliste (nmap−dev).. zu erfragen. Manche Einträge sind selbsterklärend, aber je höher die Debug−Stufe ist, desto obskurer werden die Meldungen.

−−reason (Gründe für Host− und Portzustände) .
Gibt die Gründe an, warum ein Port auf einen bestimmten Zustand gesetzt wurde und warum ein Host als ein− oder ausgeschaltet betrachtet wird. Diese Option zeigt die Paketart an, die einen Port− oder Hostzustand ermittelt hat, z.B. ein RST−Paket von einem geschlossenen Port oder ein Echo Reply von einem eingeschalteten Host. Die Information, die Nmap angeben kann, hängt von der Art des Scans oder Pings ab. Der SYN−Scan und der SYN−Ping (**−sS** und **−PS**) sind sehr detailliert, aber der TCP−Connect−Scan (**−sT**) wird durch die Implementierung des **connect**−Systemaufrufs beschränkt. Dieses Merkmal wird automatisch von der Debug−Option (**−d**). aktiviert, und die Ergebnisse werden auch dann in XML−Protokolldateien gespeichert, wenn diese Option gar nicht angegeben wird.

−−stats−every *time* (periodische Timing−Statistik ausgeben) .
Gibt periodisch eine Timing−Statusmeldung nach einem Intervall der Länge *time* aus. Dabei kann diese Zeitangabe beschrieben werden, wie in the section called “TIMING UND PERFORMANCE” dargestellt, d.h. Sie können z.B. **−−stats−every 10s** benutzen, um alle 10 Sekunden eine Statusaktualisierung zu erhalten. Diese erscheint in der interaktiven Ausgabe (auf dem Bildschirm) und in der XML−Ausgabe.

−−packet−trace (gesendete und empfangene Pakete und Daten mitverfolgen) .

Bewirkt, dass Nmap für jedes gesendete oder empfangene Paket eine Zusammenfassung ausgibt. Das wird bei der Fehlersuche oft gemacht, ist aber auch eine willkommene Methode für Neulinge, um genau zu verstehen, was Nmap unter der Oberfläche macht. Um zu verhindern, dass Tausende von Zeilen ausgegeben werden, möchten Sie vielleicht eine beschränkte Anzahl zu scannender Ports angeben, z.B. mit **−p20−30**. Wenn Sie nur wissen möchten, was im Versionserkennungssubsystem vor sich geht, benutzen Sie stattdessen **−−version−trace**. Wenn Sie nur an einer Script−Mitverfolgung interessiert sind, geben Sie **−−script−trace** an. Mit **−−packet−trace** erhalten Sie all das zusammen.

−−open (zeige nur offene (oder möglicherweise offene) Ports an) .
Manchmal interessieren Sie sich nur für Ports, mit denen Sie tatsächlich eine Verbindung herstellen können (offene Ports), und wollen Ihre Ergebnisse nicht mit anderen Ports überhäufen, die geschlossen, gefiltert und geschlossen|gefiltert sind. Die Ausgabe wird normalerweise nach dem Scan mit Werkzeugen wie grep, awk und Perl angepasst, aber dieses Merkmal wurde auf überwältigend vielfachen Wunsch hinzugefügt. Geben Sie **−−open** an, um nur offene, offene|gefilterte und ungefilterte Ports zu sehen. Diese drei Ports werden ganz wie gewöhnlich behandelt, d.h. dass offen|gefiltert und ungefiltert in Zählungen zusammengefasst werden, wenn es eine sehr große Anzahl davon gibt.

−−iflist (liste Schnittstellen und Routen auf) .
Gibt die Liste der Schnittstellen und Systemrouten aus, die Nmap entdeckt hat. Das ist hilfreich bei der Fehlersuche bei Routing−Problemen oder fehlerhaften Gerätebeschreibungen (z.B. wenn Nmap eine PPP−Verbindung als Ethernet behandelt).

−−log−errors (protokolliere Fehler/Warnungen in eine Datei im normalen Ausgabeformat) .
Von Nmap ausgegebene Warnungen und Fehlermeldungen gehen normalerweise nur auf den Bildschirm (interaktive Ausgabe), was die Ordnung aller Ausgabedateien im normalen Format (üblicherweise mit **−oN** angegeben) nicht stört. Wenn Sie diese Meldungen in den angegebenen normalen Ausgabedateien wirklich sehen möchten, können Sie diese Option benutzen. Diese ist dann hilfreich, wenn Sie die interaktive Ausgabe nicht übersehen oder wenn Sie Fehler beim Debugging speichern möchten. Die Fehlermeldungen und Warnungen werden auch im interaktiven Modus weiterhin erscheinen. Bei den meisten Fehlern bezüglich schlechter Kommandozeilenargumente wird das nicht funktionieren, da Nmap seine Ausgabedateien eventuell noch nicht initialisiert hat. Außerdem benutzen einige Nmap−Fehlermeldungen und −Warnungen ein anderes System, das diese Option noch nicht unterstützt.

Eine Alternative zu **−−log−errors** ist die Umleitung der interaktiven Ausgabe (inklusive des Standardfehlerstroms) in eine Datei. Die meisten Unix−Shells machen einem diesen Ansatz leicht, aber auf Windows kann er schwierig sein.

Weitere Ausgabeoptionen

−−append−output (an Ausgabedateien hinzufügen, statt sie zu überschreiben) .
Wenn Sie einen Dateinamen für ein Ausgabeformat wie z.B. **−oX** oder **−oN** angeben, wird diese Datei standardmäßig überschrieben. Wenn Sie deren Inhalt lieber behalten und die neuen Ergebnisse anhängen möchten, benutzen Sie die Option **−−append−output**. Dann wird bei allen angegebenen Ausgabedateinamen dieses Nmap−Aufrufs an die Dateien angehängt, statt sie zu überschreiben. Mit XML−Scandaten (**−oX**) funktioniert das nicht so gut, da die erzeugte Datei im Allgemeinen nicht mehr sauber geparst wird, es sei denn, Sie reparieren sie von Hand.

−−resume *filename* (abgebrochenen Scan fortsetzen) .
Manche umfangreichen Nmap−Läufe benötigen sehr viel Zeit – in der Größenordnung von Tagen. Solche Scans laufen nicht immer bis zum Ende. Vielleicht gibt es Beschränkungen, die verhindern, dass man Nmap während der normalen Arbeitszeit ausführen kann, das Netzwerk könnte abstürzen, der Rechner, auf dem Nmap läuft, könnte einen geplanten oder ungeplanten Neustart erleben oder Nmap selbst könnte abstürzen. Der Administrator, der Nmap ausführt, könnte es auch aus irgendeinem anderen Grund abbrechen, indem er ctrl−C eingibt. Und den ganzen Scan von vorne neu zu starten, ist eventuell nicht wünschenswert. Wenn ein normales (**−oN**) oder ein grepbares (**−oG**) Protokoll geführt wurde, kann der Benutzer Nmap jedoch bitten, den Scan bei dem Ziel fortzusetzen, an dem es beim

Abbruch gearbeitet hat. Geben Sie einfach die Option **−−resume** an und übergeben Sie die normale/grepbare Ausgabedatei als Argument. Andere Argumente sind nicht erlaubt, da Nmap die Ausgabedatei parst, um dieselben Argumente zu benutzen, die zuvor benutzt wurden. Rufen Sie Nmap einfach als **nmap −−resume** *logfilename* auf. Nmap fügt neue Ergebnisse dann an die Datendateien an, die im vorherigen Lauf angegeben wurden. Diese Fortsetzung funktioniert nicht aus XML−Ausgabedateien, weil es schwierig wäre, die zwei Läufe in einer gültigen XML−Datei zu kombinieren.

−−stylesheet *path or URL* (setze XSL−Stylesheet, um eine XML−Ausgabe zu transformieren) .
Die Nmap−Distribution enthält ein XSL−Stylesheet. namens nmap.xsl. zum Betrachten oder Übersetzen einer XML−Ausgabe nach HTML. Die XML−Ausgabe enthält eine xml−stylesheet−Anweisung, die auf nmap.xml an der Stelle verweist, wo es von Nmap ursprünglich installiert wurde (oder im aktuellen Arbeitsverzeichnis unter Windows). Laden Sie einfach Nmaps XML−Ausgabe in einem modernen Webbrowser, und er sollte nmap.xsl im Dateisystem finden und benutzen, um die Ergebnisse darzustellen. Wenn Sie ein anderes Stylesheet benutzen möchten, geben Sie es als Argument für **−−stylesheet** an. Dabei müssen Sie den vollständigen Pfadnamen oder die URL angeben. Sehr häufig wird **−−stylesheet https://nmap.org/data/nmap.xsl** benutzt. Das sagt einem Browser, dass er die neueste Version des Stylesheets von Nmap.Org laden soll. Die Option **−−webxml** macht dasselbe, verlangt aber weniger Tipparbeit und Merkfähigkeit. Wenn man das XSL von Nmap.Org lädt, wird es einfacher, die Ergebnisse auf einem Rechner anzuschauen, auf dem kein Nmap (und folglich auch kein nmap.xsl) installiert ist. Daher ist die URL oft nützlicher, doch aus Datenschutzgründen wird standardmäßig das nmap.xsl im lokalen Dateisystem benutzt.

−−webxml (lade Stylesheet von Nmap.Org) .
Diese bequeme Option ist nur ein Alias für **−−stylesheet https://nmap.org/data/nmap.xsl**.

−−no−stylesheet (lasse XSL−Stylesheet−Deklaration im XML weg) .
Geben Sie diese Option an, wenn Nmap in seiner XML−Ausgabe auf keinerlei XSL−Stylesheet verweisen soll. Die xml−stylesheet−Anweisung wird dann weggelassen.

VERSCHIEDENE OPTIONEN

Dieser Abschnitt beschreibt einige wichtige (und weniger wichtige) Optionen, für die es keinen anderen richtig passenden Ort gibt.

−6 (schaltet IPv6−Scans ein) .
Seit 2002 unterstützt Nmap bei seinen beliebtesten Features IPv6. Insbesondere das Ping−Scanning (nur für TCP), Connect−Scanning und die Versionserkennung unterstützen IPv6. Die Befehlssyntax ist die übliche, nur dass man auch die Option **−6** angibt. Natürlich müssen Sie die IPv6−Syntax angeben, wenn Sie eine Adresse statt eines Hostnamens angeben. Eine Adresse könnte wie folgt aussehen: 3ffe:7501:4819:2000:210:f3ff:fe03:14d0, d.h. es empfehlen sich Hostnamen. Die Ausgabe sieht genauso aus wie üblich. Nur die IPv6−Adresse in der Zeile der “interessanten Ports” deutet auf IPv6.

Zwar hat IPv6 die Welt nicht gerade im Sturm erobert, aber in einigen (besonders asiatischen) Ländern wird es stark eingesetzt, und von den meisten modernen Betriebssystemen wird es unterstützt. Um Nmap mit IPv6 zu benutzen, müssen sowohl die Quelle als auch das Ziel Ihres Scans für IPv6 konfiguriert sein. Falls Ihnen Ihr ISP (so wie die meisten) keine IPv6−Adressen bereitstellt, gibt es frei verfügbare sogenannte Tunnel−Broker,. die mit Nmap funktionieren. Weitere Tunnel−Broker sind **in Wikipedia aufgelistet**[12]. Ein weiterer freier Ansatz sind 6to4−Tunnels.

−A (aggressive Scan−Optionen) .
Diese Option schaltet zusätzlich erweiterte und aggressive Optionen ein. Ich habe noch nicht entschieden, wofür sie genau steht. Im Moment schaltet sie die Betriebssystemerkennung (**−O**), die Versionserkennung (**−sV**), das Scannen mit Scripts (**−sC**) und traceroute (**−−traceroute**) ein. In der Zukunft kommen vielleicht noch weitere Eigenschaften hinzu. Ziel ist es, einen umfassenden Satz von Scan−Optionen zu aktivieren, ohne dass man sich viele Flags merken muss. Weil aber das scriptbasierte Scannen mit dem Standardsatz als aufdringlich betrachtet wird, sollten Sie **−A** nicht ohne Genehmigung auf Zielnetzwerke loslassen. Diese Option aktiviert nur Eigenschaften, aber keine Optionen für das Timing (z.B. **−T4**) oder die Ausführlichkeit (**−v**), die Sie eventuell auch benutzen

möchten.

−−**datadir** *directoryname* (gibt benutzerdefinierten Ort für Nmap−Datendateien an) .
Nmap erhält einige spezielle Daten zur Laufzeit aus Dateien namens nmap−service−probes, nmap−services, nmap−protocols, nmap−rpc, nmap−mac−prefixes und nmap−os−db. Falls der Ort einer dieser Dateien angegeben wurde (mit den Optionen −−**servicedb** oder −−**versiondb**), wird dieser Ort für diese Datei benutzt. Danach sucht Nmap diese Dateien im Verzeichnis, das mit der Option −−**datadir** angegeben wurde (sofern vorhanden). Dateien, die dort nicht gefunden werden, werden in einem Verzeichnis gesucht, das durch die Umgebungsvariable NMAPDIR angegeben wird. ˜/.nmap. für echte und effektive UIDs (nur bei POSIX−Systemen) oder der Ort des ausführbaren Nmap−Programms (nur unter Win32) und dann ein bei der Kompilierung angegebener Ort wie z.B. /usr/local/share/nmap oder /usr/share/nmap. Als letzte Rettung sucht Nmap im aktuellen Arbeitsverzeichnis.

−−**servicedb** *services file* (gibt benutzerdefinierte Dienstedatei an) .
Verlangt von Nmap, die angegebene Dienstedatei zu benutzen statt der Datendatei nmap−services, die in Nmap enthalten ist. Bei dieser Option wird außerdem auch ein schneller Scan (−**F**) benutzt. Weitere Details zu Nmaps Datendateien finden Sie in der Beschreibung zu −−**datadir**.

−−**versiondb** *service probes file* (gibt benutzerdefinierte Dienstepakete an) .
Verlangt von Nmap, die angegebene Dienstepaketedatei zu benutzen statt der Datendatei nmap−service−probes, die in Nmap enthalten ist. Weitere Details zu Nmaps Datendateien finden Sie in der Beschreibung zu −−**datadir**.

−−**send−eth** (sendet rohe Ethernet−Pakete) .
Verlangt von Nmap, Pakete auf der rohen Ethernet−(Datenlink−)Schicht zu schicken, statt auf der höheren IP−(Netzwerk−)Schicht. Nmap wählt standardmäßig diejenige, die im Allgemeinen die beste für die gegebene Plattform ist. Rohe Sockets (IP−Schicht). sind im Allgemeinen auf Unix−Rechnern am effizientesten, während unter Windows Ethernet−Rahmen benötigt werden, da Microsoft keine rohen Sockets unterstützt. Trotz dieser Option benutzt Nmap rohe IP−Pakete unter Unix, wenn es keine andere Wahl hat (z.B. Verbindungen über etwas anderes als Ethernet).

−−**send−ip** (sendet auf der rohen IP−Schicht) .
Verlangt von Nmap, Pakete über rohe IP−Sockets zu senden, statt über low−level Ethernet−Rahmen. Diese Option ist das Komplement zur weiter oben beschriebenen Option −−**send−eth**.

−−**privileged** (nimmt an, dass der Benutzer alle Sonderrechte genießt) .
Sagt Nmap, dass es davon ausgehen soll, dass es über genügend Rechte verfügt, um über rohe Sockets zu senden, Paket−Sniffing und ähnliche Operationen zu betreiben, die auf Unix−Rechnern normalerweise root−Rechte. benötigen. Standardmäßig terminiert Nmap, wenn solche Operationen verlangt werden, aber **geteuid** nicht null ist. −−**privileged** ist nützlich bei Linux−Kernel−Capabilities und ähnlichen Systemen, die so konfiguriert sein können, dass sie Benutzern ohne Sonderrechte erlauben, rohe Paket−Scans durchzuführen. Vergewissern Sie sich, dass Sie diese Option vor weiteren Optionen angeben, die Sonderrechte benötigen (SYN−Scan, Betriebssystemerkennung usw.). Als äquivalente Alternative zur Option −−**privileged** kann die Umgebungsvariable **NMAP_PRIVILEGED**. gesetzt werden.

−−**unprivileged** (nimmt an, dass der Benutzer keine Sonderrechte für rohe Sockets genießt) .
Diese Option ist das Gegenteil von −−**privileged**. Sie sagt Nmap, dass es den Benutzer so behandeln soll, als genösse er keine Sonderrechte für rohe Sockets und Sniffing. Das ist nützlich beim Testen, Debugging oder falls die Möglichkeiten des rohen Netzwerkzugriffs auf Ihrem Betriebssystem vorübergehend irgendwie defekt sind. Als äquivalente Alternative zur Option −−**unprivileged** kann die Umgebungsvariable **NMAP_UNPRIVILEGED**. gesetzt werden.

−−**release−memory** (gibt Speicher vor Terminierung frei) .
Diese Option ist nur bei der Suche nach Speicherlecks nützlich. Sie bewirkt, dass Nmap den von ihm belegten Speicher direkt vor seiner Terminierung freigibt, damit man echte Speicherlecks einfacher finden kann. Normalerweise macht Nmap das nicht, weil es das Betriebssystem ohnehin macht, wenn es den Prozess terminiert.

−−interactive (startet im interaktiven Modus) .
Startet Nmap im interaktiven Modus, in dem es eine interaktive Nmap−Eingabeaufforderung gibt, bei der man mehrere Scans ausführen kann (entweder synchron oder im Hintergrund). Das ist nützlich für Leute, die von Mehrbenutzersystemen scannen, weil sie ihre Sicherheit meist testen wollen, ohne dass alle anderen im selben System genau mitbekommen, welche Systeme sie scannen. Benutzen Sie **−−interactive**, um diesen Modus zu aktivieren, und geben Sie dann h ein, um eine Hilfe zu erhalten. Diese Option wird selten benutzt, weil echte Shells für die Leute vertrauter sind und ihnen viel mehr Möglichkeiten bieten. Diese Option enthält einen bang(!)−Operator zur Ausführung von Shell−Befehlen, was einer der vielen Gründe dafür ist, Nmap nicht mit setuid root. zu installieren.

−V; **−−version** (gibt Versionsnummer aus) .
Gibt Nmaps Versionsnummer aus und terminiert.

−h; **−−help** (gibt zusammengefasste Hilfeseite aus) .
Gibt eine kurze Hilfeseite mit den am meisten benutzten Optionen aus. Sie kommt auch dann, wenn man Nmap ganz ohne Argumente startet.

LAUFZEIT-INTERAKTION

Während der Ausführung von Nmap wird jeder Tastendruck abgefangen. Das ermöglicht Ihnen, mit dem Programm zu interagieren, ohne es abzubrechen und neu zu starten. Bestimmte Spezialtasten ändern Optionen, während alle anderen Tasten eine Statusmeldung über den Scan ausgeben. Konvention ist, dass der Ausgabeumfang durch *Kleinbuchstaben vergrößert* und durch *Großbuchstaben verkleinert* wird. Sie können auch ‘*?*’ drücken, um eine Hilfe zu erhalten.

v / **V**
Vergrößert/verkleinert die Ausführlichkeit

d / **D**
Vergrößert/verkleinert die Debugging−Stufe

p / **P**
Schaltet Paketverfolgung ein/aus

?
Gibt einen Hilfeschirm zur Laufzeit−Interaktion aus

Alles andere
Gibt eine Statusmeldung wie die folgende aus:

Stats: 0:00:08 elapsed; 111 hosts completed (5 up), 5 undergoing Service Scan

Service scan Timing: About 28.00% done; ETC: 16:18 (0:00:15 remaining)

BEISPIELE

Hier sind einige Anwendungsbeispiele für Nmap, von einfachen und routinemäßigen bis zu etwas komplexeren und esoterischen. Um die Sache etwas konkreter zu machen, werden einige echte IP−Adressen und Domainnamen benutzt. Diese sollten Sie mit Adressen/Namen aus *Ihrem eigenen Netzwerk* ersetzen. Auch wenn ich nicht der Meinung bin, dass Port−Scans anderer Netzwerke illegal sind oder sein sollten, mögen manche Netzwerkadministratoren es nicht, wenn ihre Netzwerke unverlangt gescannt werden, und könnten sich beschweren. Der beste Ansatz ist der, sich zuerst eine Genehmigung zu verschaffen.

Zu Testzwecken haben Sie die Genehmigung, den Host scanme.nmap.org zu scannen. Diese Genehmigung gilt nur für das Scannen mit Nmap und nicht für das Testen von Exploits oder Denial−of−Service−Angriffen. Bitte führen Sie nicht mehr als ein Dutzend Scans pro Tag auf diesem Host durch, um die Bandbreite nicht zu erschöpfen. Falls diese freie Dienstleistung missbraucht wird, wird sie abgeschaltet, und Nmap wird dann Failed to resolve given hostname/IP: scanme.nmap.org ausgeben. Diese Genehmigung gilt auch für die Hosts scanme2.nmap.org, scanme3.nmap.org usw., auch wenn diese Hosts noch nicht existieren.

Diese Option scannt alle reservierten TCP−Ports auf dem Rechner scanme.nmap.org. Die Option **−v**

schaltet den ausführlichen Modus an.

Startet einen Stealth−SYN−Scan auf allen aktiven Rechnern unter den 256 IPs im Netzwerk der Größe “Klasse C”, in dem Scanme sitzt. Es versucht auch herauszufinden, welches Betriebssystem auf jedem aktiven Host läuft. Wegen des SYN−Scans und der Betriebssystemerkennung sind dazu root−Rechte notwendig.

Startet eine Host−Auflistung und einen TCP−Scan in der ersten Hälfte von allen 255 möglichen acht−Bit−Unternetzen im Klasse−B−Adressraum 198.116. Dabei wird getestet, ob die Systeme SSH, DNS, POP3 oder IMAP auf ihren Standardports laufen haben oder irgendetwas auf Port 4564. Falls einer dieser Ports offen ist, wird eine Versionserkennung benutzt, um festzustellen, welche Anwendung darauf läuft.

Verlangt von Nmap, 100.000 Hosts zufällig auszuwählen und sie nach Webservern (Port 80) zu scannen. Eine Host−Auflistung wird mit **−PN** unterbunden, weil es Verschwendung ist, zuerst eine Reihe von Testpaketen zu senden, um festzustellen, ob ein Host aktiv ist, wenn Sie auf jedem Zielhost ohnehin nur einen Port testen.

Das scannt 4096 IPs nach Webservern (ohne sie anzupingen) und speichert die Ausgabe im grepbaren und im XML−Format.

DAS NMAP-BUCH

Auch wenn dieser Reference Guide alle wesentlichen Nmap−Optionen genau beschreibt, kann er nicht vollständig zeigen, wie man diese Features anwendet, um Aufgaben der realen Welt zu lösen. Zu diesem Zweck haben wir das Buch Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Es zeigt, wie man Firewalls und Intrusion Detection−Systeme unterwandert, die Performance von Nmap optimiert, und wie man häufige Netzwerkaufgaben mit der Nmap Scripting Engine automatisiert. Außerdem enthält es Tipps und Anleitungen für häufige Nmap−Aufgaben wie die Netzwerkinventarisierung, Penetrationstests, die Erkennung schurkischer Wireless Access Points und das Verhindern von Wurmausbrüchen im Netzwerk. Dabei zeigt es mit Beispielen und Diagrammen, wie die Kommunikation auf der Leitung aussieht. Mehr als die Hälfte des Buches ist online frei verfügbar. Weitere Informationen finden Sie unter **https://nmap.org/book**.

Die deutsche Übersetzung dieses Buches von Dinu Gherman ist im Mai 2009 unter dem Titel **Nmap: Netzwerke scannen, analysieren und absichern**[14] im **Open Source Press**[15]−Verlag erschienen.

FEHLER

Wie sein Autor ist auch Nmap selbst nicht perfekt. Aber Sie können helfen, es zu verbessern, indem Sie Fehlerberichte schicken oder sogar Patches schreiben. Falls Nmap sich nicht wie erwartet verhält, sollten Sie zuerst auf die neueste Version aktualisieren, die unter **https://nmap.org** verfügbar ist. Wenn das Problem anhält, versuchen Sie herauszufinden, ob es bereits erkannt und bearbeitet wurde. Suchen Sie nach der Fehlermeldung auf unserer Suchseite unter **http://insecure.org/search.html** oder bei Google. Stöbern Sie in den nmap−dev−Archiven unter **http://seclists.org/**.. Lesen Sie auch diese Manpage vollständig. Wenn Sie keine Lösung finden, schicken Sie einen Fehlerbericht per E−Mail an <dev@nmap.org>. Beschreiben Sie darin bitte alles, was Sie über das Problem wissen, inklusive der Nmap−Version und der Betriebssystemversion, unter der Sie Nmap einsetzen. Berichte von Problemen und Fragen zur Anwendung von Nmap werden sehr viel wahrscheinlicher beantwortet, wenn sie an <dev@nmap.org> geschickt werden statt direkt an Fyodor. Wenn Sie sich erst auf der nmap−dev−Liste eintragen, bevor Sie Ihre E−Mail schicken, entgeht Ihre Nachricht auch der Moderation und kommt schneller an. Eintragen können Sie sich unter **https://nmap.org/mailman/listinfo/dev**.

Code−Patches zur Behebung von Fehlern sind noch besser als Fehlerberichte. Eine einfache Anweisung für die Erstellung von Patch−Dateien mit Ihren Änderungen ist unter **https://nmap.org/data/HACKING** verfügbar. Patches können an nmap−dev (empfohlen) oder direkt an Fyodor geschickt werden.

AUTOR

Fyodor <fyodor@nmap.org> (**http://insecure.org**)

Über die Jahre haben hunderte von Menschen wertvolle Beiträge zu Nmap geleistet. Sie sind detailliert in der Datei CHANGELOG. aufgeführt, die mit dem Nmap−Quellcode verbreitet wird und auch unter **https://nmap.org/changelog.html** verfügbar ist.

Sorry, this section has not yet been translated to German. Please see the **English version**[16].

NOTES

1. RFC 1122
   http://www.rfc-editor.org/rfc/rfc1122.txt
2. RFC 792
   http://www.rfc-editor.org/rfc/rfc792.txt
3. RFC 1918
   http://www.rfc-editor.org/rfc/rfc1918.txt
4. UDP
   http://www.rfc-editor.org/rfc/rfc768.txt
5. TCP RFC
   http://www.rfc-editor.org/rfc/rfc793.txt
6. RFC 959
   http://www.rfc-editor.org/rfc/rfc959.txt
7. RFC 1323
   http://www.rfc-editor.org/rfc/rfc1323.txt
8. Programmiersprache Lua
   http://lua.org
9. IP-Protokoll
   http://www.rfc-editor.org/rfc/rfc791.txt
10. Nmap::Scanner
    http://sourceforge.net/projects/nmap-scanner/
11. Nmap::Parser
    http://nmapparser.wordpress.com/
12. in Wikipedia aufgelistet
    http://en.wikipedia.org/wiki/List_of_IPv6_tunnel_brokers
13. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning
    https://nmap.org/book/
14. Nmap: Netzwerke scannen, analysieren und absichern
    https://www.opensourcepress.de/index.php?26&backPID=178&tt_products=270
15. Open Source Press
    http://www.opensourcepress.de
16. English version
    https://nmap.org/book/man-legal.html