SelfLinux-0.13.1

nmap - der Netzwerksicherheits-Scanner

Autor: Gabriel Welsche (*gabriel.welsche@web.de*)
Autor: Marc Ruef
Formatierung: Florian Frank (*florian@pingos.org*)

Das kleine Port Scanning Tool namens `nmap` wird hier mit dem Ziel vorgestellt, die schier erschreckend scheinenden Funktionen in vollem Umfang nutzen zu können. Die verschiedenen Techniken werden ebenfalls knapp erläutert, sodass dieser Artikel nicht nur ausschließlich für Nutzer von `nmap` interessant sein dürfte.

`nmap` bietet im Gegensatz zu Konkurrenzprodukten erweiterte TCP- und UDP - Funktionalitäten, die mit einer Portion Wissen das Leben eines Netzwerk-Administrators extrem erleichtern können.

Inhaltsverzeichnis

1 Einführung

nmap - der **Network Mapper** - wurde entwickelt, um Systemadministratoren und anderen kuriosen Individuen die Möglichkeit zu geben, Netzwerke zu analysieren, festzustellen welche Hosts aktiv sind und welche Dienste durch sie bereitgestellt werden. nmap ist ein so genannter Portscanner (scannen = abtasten), d. h. er prüft IP - Netzwerkkomponenten auf geöffnete bzw. geschlossene TCP/UDP - Ports. Dabei werden eine Vielzahl verschiedener Abtastverfahren unterstützt, zum Beispiel

* UDP- und TCP Connects
* TCP SYN-Scans
* FTP-Proxy (Bounce-Attacks)
* Reverse-Ident
* ICMP-Scans (Ping-Schleife)
* FIN - Scans
* ACK Schleifen (Antwortpakete)
* X-MAS
* SYN-Schleifen
* Null - Scan

Ebenso bietet nmap eine Vielzahl von zusätzlichen Möglichkeiten, wie

* Betriebssystemerkennung mit TCP/IP Fingerabdruck,
* Stealth - Scans,
* dynamische Verzögerung und Übertragungsberechnungen,
* Parallel - Scans,
* Abfragen von inaktiven Hosts über parallele Pings,
* Portfilter - Identifikation,
* RPC - Scans,
* Scans auf fragmentierte Pakete sowie
* Ziel- und Port- Scans.

Das Resultat eines `nmap` - Durchlaufs ist normalerweise eine Liste sämtlicher interessanter Ports der **gescannten** Geräte. Sofern eine Zuweisung stattfinden kann, benennt nmap die bekannten Ports direkt mit ihrem Service-Namen, Portnummer, Status (offen, gefiltert, ungefiltert) und Protokoll. Der Status **offen** (engl. open) bedeutet, dass das Zielsystem in der Lage ist, auf diesem Port Verbindungen anzunehmen. **Gefiltert** (engl. filtered) weist darauf hin, dass ein dediziertes Firewall - System, TCP/IP-Filter oder Netzwerk-Element die Arbeit von nmap behindert und somit keine verlässlichen Rückschlüsse gezogen werden können. **Ungefiltert** (engl unfiltered) heißt, dass nmap den Port kennt, jedoch beim Zugriff keinerlei Filtermechanismen ausgemacht werden konnten. Der ungefilterte Status wird in den meisten aller Fälle auftreten. Deshalb wird ein solcher nur immer dann explizit ausgewiesen, wenn die meisten der gescannten Ports gefiltert sind.

(Leider) benötigen viele Techniken (z. B. die kernelnahen raw sockets) root-Privilegien. Aus diesem Grund ist es oftmals erforderlich, `nmap` als root auszuführen.

1.1 Anwendung

Wie bei den meisten zeilenorientierten Programmen in einer Linux-Umgebung wird mit dem Parameter `-h` bzw. dem parameterlosen Aufruf von `nmap` die Hilfefunktion auf dem Bildschirm ausgegeben. Die Syntax für das kleine Wunderding ist wie folgt festgelegt (ins Deutsche übersetzt):

```
user@linux / $ nmap [Scan-Typ(en)] [Optionen] <Host oder Netz #1 ...
```

[#N]> Scan-Typen

Folgend nun die original Ausgabe von nmap ohne Parameter-Eingabe:

```
user@linux / $ nmap


nmap V. 2.3BETA6 usage: nmap [Scan Type(s)] [Options] <host or net #1
...[#N]>;
Scan types
  -sT tcp connect() port scan
  -sS tcp SYN stealth port scan (must be root)
  -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (only works against UNIX).
  -sP ping "scan". Find which hosts on specified network(s) are up but
don't
port scan them
  -s UDP port scan, must be r00t
  -sR RPC scan (use in addition to other TCP and/or UDP scan type(s)
  -b <ftp_relay_host> ftp "bounce attack" port scan
Options (none are required, most can be combined):
  -f use tiny fragmented packets for SYN, FIN, Xmas, or NULL scan.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
  -PT Use only "TCP Ping" to see what hosts are up (for normal & ping
scans).
  -PI Use only ICMP ping to determines hosts that are up (default is
ICMP&TCP)
  -PS Use TCP SYN sweep rather than the default ACK sweep used in "TCP
ping"
   -O Use TCP/IP fingerprinting to guess what OS the remote host is
running
  -p <range> ports: ex: '-p 23' will only try port 23 of the host(s)
             '-p 20-30,63000-' scans 20-30 and 63000-65535.
             default: 1-1024 + /etc/services
  -Ddecoy_host1,decoy2,decoy3[,...] Launch scans from decoy host(s) along
             with the real one.
  -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane>
      Use specified timing policy.
  -F fast scan. Only scans ports in /etc/services, a la strobe(1).
  -I Get identd (rfc 1413) info on listening TCP processes.
  -n Don't DNS resolve anything unless we have to (makes ping scans
faster)
  -R Try to resolve all hosts, even down ones (can take a lot of time)
  -o <logfile> Output scan logs to <logfile> in human readable.
  -m <logfile> Output scan logs to <logfile> in machine parseable format.
  -i <inputfile> Grab IP numbers or hostnames from file. Use '-' for
stdin
  -g <portnumber> Sets the source port used for scans. 20 and 53
      are good choices.
  -S <your_IP> If you want to specify the source address of SYN or FYN
scan.
  -v Verbose. Its use is recommended. Use twice for greater effect.
  -h help, print this junk. Also see http://www.insecure.org/nmap/
  -V Print version number and exit.
  -e <devicename>. Send packets on interface <devicename>
(eth0,ppp0,etc.).
Hostnames specified as internet hostname or IP address. Optional '/mask'
specifies subnet. For example: cert.org/24 or 192.88.209.5/24 or
```

```
192.88.209.0-255 or '128.88.209.*' all scan CERT's Class C.
SEE THE MAN PAGE FOR MORE THOROUGH EXPLANATIONS AND EXAMPLES.
```

2 Mögliche Scan - Techniken (nmap Optionen)

Je nach dem welche Optionen verwendet wurden, ist nmap in der Lage, Auskunft über die folgenden Charakteristiken des Zielsystems zu geben:

* Genutztes Betriebssystem
* TCP- Sequenznummern
* Benutzername, unter der die an die Ports gebundene Software abläuft
* DNS-Name
* ob es sich um ein Smurf - System handelt und viele mehr.

Die Kombination verschiedener Optionen ist immer dann möglich, wenn ein Zusammenspiel auch Sinn macht. Einige Parameter können nur in Verbindung mit spezifischen Scan-Methoden genutzt werden.

2.1 Ping - Suchlauf

Mit dem Parameter -sP kann nmap die aktiven Hosts in einem Netzwerk zu ermitteln. Dabei wird traditionell ein ICMP-ECHO-(Typ 8)-Paket an ein Ziel geschickt, wobei bei dessen Erreichbarkeit mit einem ICMP-ECHO-REPLY-(Typ 0)-Paket als Antwort signalisiert wird.

Viele gewissenhafte Firewall- und Systemadministratoren filtern unnötigen ICMP - Verkehr. nmap greift in diesem Falle auf eine andere Technik zurück - es wird ein TCP - Datagramm mit gesetztem ACK - Flag an einen potentiell offenen Port des Zielsystems geschickt. Der Empfang eines RST-Paketes bedeutet, dass das Zielsystem vorhanden und ansprechbar ist.

Führt auch diese Technik nicht zum Erfolg, so versucht nmap eine weitere Methode, die auf ein SYN - Datagramm zurückgreift, das auf ein RST oder SYN/ACK wartet.

Wichtig ist zu wissen, dass der Ping - Zugriff standardmäßig erfolgt. Wird dies nicht gewünscht, so ist mit dem Parameter -P0 dieses Verhalten zu deaktivieren.

Für Mass-Pings sind andere Tools nutzbar, deren Herangehensweise oftmals zu besseren Resultaten führt (Zum Beispiel fping, gping und hping).

```
user@linux / $ nmap -sP 192.168.0.0/24

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Host gateway.matrix.net (192.168.0.1) appears to be up.
Host prometheus.matrix.net (192.168.0.2) appears to be up.
Host rieekan.matrix.net (192.168.0.3) appears to be up.
Host margrit.matrix.net (192.168.0.4) appears to be up.
Host roadrunner.matrix.net (192.168.0.5) appears to be up.
Nmap run completed -- 256 IP addresses (5 hosts up) scanned in 1 second
```

2.2 TCP - Port Scan mit Verbindungsaufbau

Der Parameter -sT führt einen normalen TCP - Port Scan durch, bei dem eine Verbindung zum Ziel - Port aufgebaut wird. Der Client beginnt mit der Übertragung eines SYN-Paketes an den Server. Jener quittiert den Empfang mit einem SYN-/ACK-Paket und wartet dann auf den Erhalt einer weiteren Bestätigung durch ein

ACK-Paket von Seiten des Clients (3-Way-Handshake).

Diese Scanning - Technik ist sehr leicht zu entdecken und wird mit größter Wahrscheinlichkeit in den Protokoll-Dateien des Zielsystems auftauchen. Dies ist die einzige Scan - Technik für unprivilegierte Anwender.

BeOS 4.5

```
user@linux / $ nmap -sT 192.168.0.2

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on prometheus.matrix.net (192.168.0.2):
Port State Protocol Service
23 open tcp telnet

Nmap run completed -- 1 IP address (1 host up) scanned in 2 seconds
```

Microsoft Windows 98

```
user@linux / $ nmap -sT 192.168.0.3

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on rieekan.matrix.net (192.168.0.3):
Port State Protocol Service
139 open tcp netbios-ssn

Nmap run completed -- 1 IP address (1 host up) scanned in 8 seconds
```

SuSE Linux 6.3

```
user@linux / $ nmap -sT 192.168.0.1

 Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
 Interesting ports on gateway.matrix.net (192.168.0.1):
 Port State Protocol Service
 21 open tcp ftp
 23 open tcp telnet
 139 open tcp netbios-ssn
 8080 open tcp http-proxy

 Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
```

2.3 Stealth - Port Scan

Mit Stealth - Scans (Tarnkappen) kann die Netzwerkanalyse verdeckt erfolgen. Viele Firewall- und IDS - Systeme sind gegen solche Scans machtlos. Wie funktioniert das? Das 3-Way Handshake Protokoll wird missachtet oder anders gesagt: ein Stealth - Scan führt keinen kompletten Verbindungsaufbau durch. Die Interpretation der Antworten der Opferrechner führt zu den gewünschten Erkenntnissen. Weil keine vollständige

TCP - Verbindung zustande kommt, spricht man auch vom **halboffenen Scan**.

2.3.1 TCP - SYN - Stealth Portscan

Der Parameter `-sS` erlaubt es, einen halboffenen Scan durchzuführen. Es wird einfach ein SYN-Paket zum Ziel-Port übertragen. Antwortet der Ziel-Port mit SYN/ACK, kann davon ausgegangen werden, dass der Port den Listening - Status besitzt. Wird RST/ACK zurückgegeben, kann mit ziemlicher Wahrscheinlichkeit ein inaktiver Port zugeordnet werden. (Das Client-System überträgt nach dem Empfang des Paketes des Hosts ein RST/ACK-Paket, sodass keine vollständige Verbindung hergestellt wird.)

Diese Scanning - Technik funktioniert normalerweise nur bei UNIX-basierenden TCP - Stacks. Es zeigt sich wieder einmal die inkorrekte Implementierung der Windows - Stacks.

BeOS 4.5

```
root@linux / # nmap -sS 192.168.0.2

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on prometheus.matrix.net (192.168.0.2):
Port State Protocol Service  23 open tcp telnet

Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds
```

Microsoft Windows 98

```
root@linux / # nmap -sS 192.168.0.3

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on rieekan.matrix.net (192.168.0.3):
Port State Protocol Service
41 open tcp graphics
134 open tcp ingres-net
139 open tcp netbios-ssn
168 open tcp rsvd
174 open tcp mailq
[...]
2430 open tcp venus
2604 open tcp ospfd
4500 open tcp sae-urn
6110 open tcp softcm
7006 open tcp afs3-errors

Nmap run completed -- 1 IP address (1 host up) scanned in 20 seconds
```

SuSE Linux 6.3

```
root@linux / # nmap -sS 192.168.0.1

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
```

```
Interesting ports on gateway.matrix.net (192.168.0.1):
Port State Protocol Service
21 open tcp ftp
23 open tcp telnet
139 open tcp netbios-ssn
8080 open tcp http-proxy

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
```

2.3.2 Stealth FIN-, Xmas-Tree- oder Null-Scan-Modis

Die zugrundeliegende Idee besteht darin, dass laut RFC 793, S.64 geschlossene Ports auf derartige Zugriffe mit einem RST - Datagramm antworten während ansprechbare Ports die Anfragen ignorieren. Der FIN-Scan nutzt ein TCP - Datagramm mit gesetztem FIN - Flag, während der Xmas - Tree - Scan die TCP - Flags FIN, URG und PSH aktiviert. Der Null - Scan schaltet alle optionalen Flags ab.

Einige Firewall - Systeme (z. B. Paket-Filter) sind in der Lage, verdächtige SYN-Aktivitäten zu erkennen. Ebenso können Programme wie Synlogger oder Courtney diese SYN - Portscans als solche ausweisen.

2.3.2.1 Stealth - FIN Portscan

Der Parameter `-sF` veranlasst nmap, einen Stealth - FIN Scan durchzuführen. Dabei wird einfach ein FIN-Paket zum Ziel-Port übertragen. Nach der Empfehlung von RFC 793 müsste der Host danach RST für alle geschlossenen Ports zurückgeben. Beispiele:

bei BeOS 4.5

```
root@linux / # nmap -sF 192.168.0.2

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on prometheus.matrix.net (192.168.0.2):
Port State Protocol Service
23 open tcp telnet

Nmap run completed -- 1 IP address (1 host up) scanned in 4 seconds
```

bei Microsoft Windows 98

```
root@linux / # nmap -sF 192.168.0.3

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
No ports open for host rieekan.matrix.net (192.168.0.3)
Nmap run completed -- 1 IP address (1 host up) scanned in 21 seconds
```

bei SuSE Linux 6.3

```
root@linux / # nmap -sF 192.168.0.1
```

```
Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on gateway.matrix.net (192.168.0.1):
Port State Protocol Service
21 open tcp ftp
23 open tcp telnet
139 open tcp netbios-ssn
8080 open tcp http-proxy

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
```

2.3.2.2 TCP - Xmas - Tree Portscan

Der Parameter `-sX` ermöglicht das Senden eines FIN-, URG- und PUSH - Paketes zum gewünschten Ziel-Port. Nach den Empfehlungen von RFC 792 müsste der Host RST für alle geschlossenen Ports zurückgeben.

Es sind root - Rechte für das Durchführen eines solchen Scans von Nöten. Meistens funktioniert dieser Scan nur bei Systemen mit UNIX-Protokollstapeln. Beispiele

bei BeOS 4.5

```
root@linux / # nmap -sX 192.168.0.2

Interesting ports on prometheus.matrix.net (192.168.0.2):
Port State Protocol Service
23 open tcp telnet

Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds
```

bei Microsoft Windows 98

```
root@linux / # nmap -sX 192.168.0.3

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
No ports open for host rieekan.matrix.net (192.168.0.3)
Nmap run completed -- 1 IP address (1 host up) scanned in 19 seconds
```

bei SuSE Linux 6.3

```
root@linux / # nmap -sX 192.168.0.1

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on gateway.matrix.net (192.168.0.1):
Port State Protocol Service
21 open tcp ftp
23 open tcp telnet
139 open tcp netbios-ssn
```

```
8080 open tcp http-proxy

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

2.3.2.3 TCP - Null Portscan

Mit dem Parameter `-sN` führt `nmap` einen sogenannten TCP-Null-Scan durch, bei dem alle Markierungen (Flags) ausgeschaltet werden. Nach den Empfehlungen aus RFC 793 müsste das Ziel-System RST für alle geschlossenen Ports retournieren.

Für den TCP-Null-Scan sind wieder Root-Rechte notwendig. Dieser Scan funktioniert meist nur bei Systemen mit UNIX-IP-Stacks. Beispiele:

bei BeOS 4.5

```
root@linux / # nmap -sX 192.168.0.2

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on prometheus.matrix.net (192.168.0.2):
Port State Protocol Service
23 open tcp telnet

Nmap run completed -- 1 IP address (1 host up) scanned in 4 seconds
```

bei Microsoft Windows 98

```
root@linux / # nmap -sX 192.168.0.3

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
No ports open for host rieekan.matrix.net (192.168.0.3)
Nmap run completed -- 1 IP address (1 host up) scanned in 19 seconds
```

bei SuSE Linux 6.3

```
root@linux / # nmap -sN 192.168.0.1

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on gateway.matrix.net (192.168.0.1):
Port State Protocol Service
21 open tcp ftp
23 open tcp telnet
139 open tcp netbios-ssn
8080 open tcp http-proxy

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

2.4 TCP/ACK-Scan

Mit der Option `-sA` wird diese erweiterte Scan - Technik angewendet, um ein Paketfilter- Regelwerk zu identifizieren. Zusätzlich kann diese Methode eine Stateful Inspection (iptables) aufzeigen.

Dieser Scan - Typ schickt ein ACK - Paket an den spezifizierten Zielport. Kommt ein RST zurück, wird der besagte Port als **unfiltered** (dt. ungefiltert) eingestuft. Wird keine Rückantwort empfangen (oder kommt ein ICMP unreachable zurück), so weist nmap den Port als **filtered** (dt. gefiltert) aus. Wichtig ist, dass `nmap` normalerweise keine **unfiltered**-Meldung ausgibt. So sind keine Ports in der Ausgabe ein Indiz dafür, dass alle Zugriffe durchgekommen sind (und ein RST verursacht haben). Dieser Scan wird die Ports nie in einem **open** (dt. offenen) Status zeigen.

2.5 Window - Scan

Diese erweiterte Scan - Technik, die mit Parameter `-sW` ausgewählt wird, ist dem ACK - Scan sehr ähnlich, außer, dass hiermit manchmal auch offene, ungefilterte und gefilterte Ports durch eine Anomalie in der durch die Betriebssysteme gewählten TCP - Window - Size entdeckt werden können.

2.6 UDP - Portscan

Mit dem Parameter `sU` führt `nmap` einen UDP - Portscan durch, bei dem lediglich ein UDP-Paket (siehe RFC 768) zum Ziel-Port übertragen wird. Wenn der Ziel-Port mit der Nachricht **ICMP Port unreachable** den Erhalt quittiert, ist der Port inaktiv und somit geschlossen.

Entgegen der weitläufigen Meinung, UDP-Scanning sei sinnlos, soll in diesem Zusammenhang auf die Lücke in Solaris' rpcbind hingewiesen werden. rpcbind kann am undokumentierten UDP - Port 32770 gefunden werden. Es ist also vollkommen irrelevant, ob Port 111 durch eine Firewall blockiert wird oder nicht. Als zweites Beispiel für die Sicherheitsrelevanz von UDP sei die populäre, von cDc entwickelte Backdoor namens Back - Orifice genannt, durch die Windows-Maschinen über einen frei wählbaren UDP - Port gesteuert werden können. Und drittens sollten die vielen potentiell verwundbaren UDP - basierten Dienste nicht vergessen werden: SNMP, TFTP, NFS, etc.

Ein UDP - Scan kann sich extrem in die Länge ziehen, wenn im großen Umfang Paketfilter eingesetzt werden. `nmap` ist in der Lage, Limitierungen hinsichtlich der maximalen Anzahl ausgehender ICMP-Fehlernachrichten (RFC 1812 Abs. 4.3.2.8) zu erkennen und mit einer dynamischen Geschwindigkeitsreduzierung zu reagieren. Dies verhindert das Verstopfen des Netzwerks mit unnötigen Paketen, die sowieso vom Zielsystem ignoriert werden würden.

Einmal mehr typisch, ignoriert Microsoft die Empfehlungen des RFCs, weshalb eine Einschränkung ausgehender ICMP - Fehlermeldungen gänzlich bei der TCP/IP - Implementierung auf Windows 9x und NT fehlt. Das Scannen sämtlicher UDP-Ports auf einer Windows - Maschine ist somit kein größeres Problem. Beispiele:

bei BeOS 4.5

```
root@linux / # nmap -sU 192.168.0.2

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on prometheus.matrix.net (192.168.0.2):
Port State Protocol Service
1 open udp tcpmux
```

```
2 open udp compressnet
3 open udp compressnet
4 open udp unknown
5 open udp rje
[...]
10080 open udp amanda
17007 open udp isode-dua
18000 open udp biimenu
31337 open udp BackOrifice
47557 open udp dbbrowse

 Nmap run completed -- 1 IP address (1 host up) scanned in 23 seconds
```

bei Microsoft Windows 98

```
root@linux / # nmap -sU 192.168.0.3

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on rieekan.matrix.net (192.168.0.3):
Port State Protocol Service
137 open udp netbios-ns
138 open udp netbios-dgm
1025 open udp blackjack

Nmap run completed -- 1 IP address (1 host up) scanned in 15 seconds
```

2.7 IP protocol-Scans

Diese Methode kommt dann zum Tragen, wenn herausgefunden werden soll, welche IP - Protokolle das Zielsystem unterstützt. Bei dieser Technik wird für jedes IP - Protokoll ein RAW - IP - Paket mit fehlendem Protokoll - Header (ohne Daten) an das Zielsystem geschickt wird. Wird darauf mit einer **ICMP protocol unreachable** - Fehlermeldung reagiert, so ist davon auszugehen, dass das Zielsystem das Protokoll nicht beherscht. Nun kann `nmap` eine Liste der unterstützten Protokolle erstellen.

Einige Betriebssysteme (z.B. AIX, HP-UX und Digital UNIX) und Firewall-Lösungen verzichten gänzlich auf das Versenden der **ICMP protocol unreachable** - Fehlermeldungen. Als Resultat eines solchen Verhaltens behauptet `nmap`, dass sämtliche Protokolle **offen** sind.

2.8 Verbergen der Identität - Idle-Scan

Diese erweiterte Scan - Technik, die mit Parameter `-sI` gewählt wird, ermöglicht ein **verstecktes** Scannen der TCP-Ports eines Zielssystems (dies bedeutet, dass keinerlei Pakete mit der richtigen IP - Absenderadresse verschickt werden).

Neben der absoluten Gewissheit, nicht direkt erkannt zu werden, können durch einen derartigen Scan IP - basierte Vertrauensbeziehungen aufgedeckt werden. Das Port - Listing zeigt die offenen Ports aus der Sicht eines beliebigen Zombie-Systems, welches durch den Parameter (-sI <Zombie-Host[:Zielport]>) explizit angegeben werden kann.

2.9 RPC - Scan

Diese Methode arbeitet in Kombination mit den meisten möglichen Scan - Typen von `nmap` zusammen. Jeder als offen identifizierte TCP- bzw. UDP - Port wird mit einer Vielzahl von SunRPC - Nullkommandos überflutet, um eine Identifizierung des am RPC - Port lauschenden Dienstes vorzunehmen.

Kann ein solcher Dienst ermittelt werden, wird dessen Programmname und die Version ausgelesen. Beim Einsatz eines Portmappers sei hier das Verwenden von `rpcinfo -p` erwähnt.

2.10 FTP - "Bounce - Attack" Portscan

Diese Scanning - Technik spielt heutzutage fast keine Rolle mehr, denn es basiert auf einem **Feature** des FTP-Protokolls, welches bei den meisten Servern mittlerweile deaktiviert wurde. Es wird mit Parameter `-b <ftp_relay_host>` aktiviert und ermöglicht das erfolgreiche und graziöse Vertuschen der Scan - Herkunft.

Die Technik wurde von Hobbit bei Bugtraq 1995 veröffentlicht: Es wird versucht, Verbindungen zu einem FTP-Server durch den Missbrauch des Ports für FTP - Proxy - Verbindungen zu vertuschen. Wie Hobbit relativ detailliert im o. g. Bericht beschreibt, können solche Angriffe für die Übermittlung von fast nicht zurückverfolgbaren Mails und News, für Attacken auf beliebige Systeme durch das Füllen von Festplatten oder das Durchbrechen von Firewalls genutzt werden.

Zugleich sind hohe Anforderungen für das Gelingen dieser Attacke nötig: Der FTP-Server muss ein beschreibbares Verzeichnis haben sowie falsche Port-Informationen mit dem PORT - Befehl von `nmap` erlauben. Es sei erwähnt, dass diesen Anforderungen an das Zielsystem ein solcher Scan sehr zeitintensiv sein kann.

3 Parameter für zusätzliche Optionen

Keine der folgenden Optionen ist zwingend erforderlich. Einige von ihnen sind jedoch äußerst sinnvoll.

3.1 Bestimmte Ports / Portbereiche auswählen

Der Parameter `-p` legt fest, welche Ports gescannt werden sollen. Wird bei der Wahl der Zielports auf diese Option verzichtet, werden sämtliche **well-known Ports** zwischen 1 und 1024 sowie alle in der services - Datei von `nmap` gelisteten Dienste gescannt. Für einen IP - Protokoll - Scan (`-sO`) legt der Parameter `-p` die zu scannende Protokoll-Nummer (0-255) fest.

Beispiele: `-p 23` wird lediglich einen Zugriff auf den Port 23 (Telnet) der Zielsysteme durchführen. `-p 20-30,139,60000-` scannt die Ports zwischen 20 und 30, Port 139 und alle Ports größer als 60000.

Bei gleichzeitigem TCP- und UDP - Portscan wird das jeweilige Protokoll durch ein vorangestelltes **T:** oder **U:** angewählt. Für die übergebenen Ports gilt so lange das spezifizierte Übertragungsprotokoll, bis ein anderes angegeben wird (Beispiel **-p U:53,111,137,T:21-25,80** = UDP-Ports 53, 111 und 137 sowie die TCP - Ports 21 bis 25 und 80). Bei einem gleichzeitigen TCP- und UDP - Scan muss mindestens eine TCP - Scan - Variante (`-sS`, `-sF` oder `-sT`) angegeben werden.

3.2 Wiederaufnahme eines abgebrochenen Scan

Die Option `--resume <Protokoll-Dateiname>` reaktiviert einen Netzwerk-Scan, der durch das Drücken von Control-C unterbrochen wurde.

3.3 Performance-Optimierung

3.3.1 Schneller Scannen

Mit der Funktion `F` wird ähnlich wie beim Strobe (einfacher und schneller TCP-Portscanner) ein ziemlich schneller Scan durchgeführt, da nur die Ports angesprochen werden, die in /etc/services angeführt wurden. Normalerweise reicht ein Scan auf diese Standard-Ports aus, da damit die gängigsten Dienste in Erfahrung gebracht werden können.

3.3.2 DNS-Auflösung deaktivieren

Um einen Ping-Scan zu beschleunigen, kann mit der Option `-n` die DNS-Auflösung ausgeschaltet werden.

3.3.3 Begrenzung der Sockets (Vermeidung eines Zusammenbruchs des Zielsystems)

Mit `-M <Maximale Sockets>` wird die maximale Anzahl der Sockets bei einem parallel durchgeführten TCP connect()-Scan festgelegt. Dies ist zum Beispiel in Situationen nützlich, wenn der Scanvorgang künstlich verlangsamt werden soll, damit das Zielsystem nicht unter der Last der Zugriffe zusammenbricht. Eine andere Herangehensweise ist die Verwendung von -sS.

3.4 Scannen ohne vorheriges An-ping-en des Hosts

Der Parameter `-P0` verhindert das Pingen eines Hosts, bevor dieser gescannt wird. So können ganze Netzwerke

gescannt werden, die aufgrund einer restriktiv konfigurierten Firewall keine ICMP echo requests/responses) zulassen. Microsoft.com ist ein Beispiel für ein solch gut geschütztes Netzwerk, in dem nur die Verwendung dieser Option zum gewünschten Resultat führt.

3.5 Klein fragmentierte Pakete für "Stealth-Scans"

Durch die zusätzlich Option `-f` werden sehr kleine fragmentierte Pakete für SYN-, FIN-, Xmas- oder NULL-Scans verwendet. Das bedeutet, dass die TCP - Header auf mehrere Pakete verteilt werden. Ziel ist es, den Scan vor Firewall- und IDS - Systemen zu verstecken. Gerade bei älteren Geräten werden die Pakete oftmals nicht zuerst defragmentiert. Heutzutage reihen Paketfilter in den meisten Fällen die einzelnen Pakete in eine Warteschlange ein, bevor sie mit der Auswertung / Filterung beginnen. Damit rückt das Ziel dieser Option in weite Ferne.

Bei verschiedenen Testläufen ist wurde festgestellt, dass ein SYN - Scan mit einer großen Anzahl klein fragmentierter Pakete in einer kleinen Anzahl Versuche `nmap` verfälschte Daten ausgab. Der Test lief von einem SuSE Linux 6.3-Client mit 10 MBit/Sek. (Koaxial-Kabel) gegen ein SuSE Linux 6.3-Gateway. `Nmap` behauptete in einigen Fällen, dass zwischen dem zweitletzten wirklich offenen Port 139 (netbios-ssn) und dem allerletzten Port 8080 (http-proxy) ein gefilterter TCP - Port offen sei.

3.6 Timing-Einstellungen

Mit `Nmap` können Scans weitestgehend unentdeckt bleiben. In einigen Fällen kann mit zusätzlichen Optionen das Timing noch feiner abgestimmt werden.

Mit den `-T` Optionen Paranoid, Sneaky, Polite, Normal, Aggressive und Insane lässt sich das Timing der Scans auf das Zielsystem einstellen:

* Der Paranoid - Modus scannt sehr langsam, in der Hoffnung, nicht von Intrusion Detection-Systemen entdeckt zu werden. Er serialisiert alle Scand und wartet im Normalfall einfach fünf Minuten bis zur Sendung des Folgepaketes.
* Der Sneaky - Modus (dt. schleichend) ist mit dem Paranoid-Modus vergleichbar, allerdings sendet dieser die Pakete im Abstand von 15 Sekunden.
* Der Polite - Modus (dt. höflich) verringert die Netzlast und verkleinert die Gefahr des Zusammenbrechens der Zielmaschine. Er serialisiert ebenfalls die zu sendenden Pakte und wartet dazwischen mindestens 0.4 Sekunden.
* Der Normal-Modus scannt so schnell wie möglich, ohne das Netz dabei zu überlasten.
* Im Aggressive-Modus (dt. aggressiv) wird eine Wartezeit von 5 Minuten zwischen den einzelnen Hosts hinzugefügt, jedoch wird nie länger als 1.25 Sekunden auf Antworten gewartet.
* Der Insane - Modus (dt. geisteskrank) ist lediglich in sehr schnellen Netzwerken verwendbar, oder überall dort zu empfehlen, wo einzelne Resultatsinformationen verloren gehen dürfen. Zwischen den einzelnen Systemen werden 75 Sekunden und zwischen den Zugriffen 300 ms gewartet.

Weiterhin sind verschiedene sehr feingranulare Einstellungen mit verschiedenen Parametern wie z. B. `--host_timeout`, `--max_rtt_timeout`, `--initial_rtt_timeout`, `--max_parallelism`, `--scan_delay`, etc. möglich.

3.7 Fingerprints zur Ermittlung des Betriebssystems

Mit `-O` wird ein so genannter Fingerabdruck des gescannten Systems angefertigt. Es wird eine Anzahl spezifischer Tests mit dem Ziel ausgeführt, das typische Verhalten der jeweiligen TCP/IP - Implementierungen zu erkennen. Die erhaltenen Informationen stellen quasi einen **Fingerabdruck** des Systems dar, der mit bekannten Betriebssystem-Fingerabdrücken verglichen wird. Diese sind in der nmap-os-fingerprints Datei zu

finden.

Es kommt nahezu nie vor, dass ein falsches Betriebssystem prognostiziert wird, solange die Protokollstapel beim Host nicht manipuliert wurden. Es kann höchstens vorkommen, dass `nmap` dem Fingerabdruck des Computers kein Betriebssystem zuordnen kann. Sollte es einem Endanwender von `nmap` möglich sein das Betriebssystem zu identifizieren, kann er per CGI-Script auf http://www.insecure.org/cgi-bin/nmap-submit.cgi aktiv zur Perfektionierung von `nmap` beitragen.

Die Option `-O` aktiviert weiterhin einige zusätzliche Tests wie z. B. das Messen der **Uptime** (wann wurde das Zielsystem das letzte Mal neu gestartet) oder die Klassifizierung der Berechenbarkeit der TCP-Sequenznummer (wie schwer es ist, eine bestehende Verbindung zu entführen).

Mit Eingriffen in die Handhabung des Betriebssystems bei Verbindungs-Anforderungen und direkt beim Protokollstapel könnte ein anderes Betriebssystem vorgetäuscht werden. Diese Aktion ist jedoch mit dem bitteren Beigeschmack eines möglichen Performance- und Stabilitäts-Verlusts des Betriebssystems verbunden. Beispiele:

bei Microsoft Windows 98

```
root@linux / # nmap 192.168.0.3 -O

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on rieekan.matrix.net (192.168.0.3):
Port State Protocol Service
139 open tcp netbios-ssn

TCP Sequence Prediction: Class=trivial time dependency
Difficulty=0 (Trivial joke)
Remote operating system guess: Windows NT4 / Win95 / Win98

Nmap run completed -- 1 IP address (1 host up) scanned in 7 seconds
```

bei SuSE Linux 6.3

```
root@linux / # nmap 192.168.0.3 -O

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on gateway.matrix.net (192.168.0.1):
Port State Protocol Service
21 open tcp ftp
23 open tcp telnet
139 open tcp netbios-ssn
8080 open tcp http-proxy

TCP Sequence Prediction: Class=random positive increments
Difficulty=2150475 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.2.12

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

bei unbekanntem Betriebssystem

```
root@linux / # nmap 192.168.0.6 -O

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on hidden.matrix.net (192.168.0.6):
Port State Protocol Service
7 open tcp echo
9 open tcp discard
13 open tcp daytime
17 open tcp qotd
19 open tcp chargen
[...]
465 open tcp smtps
1030 open tcp iad1
1433 open tcp ms-sql-s
6667 open tcp irc
6668 open tcp irc

TCP Sequence Prediction: Class=trivial time dependency
Difficulty=11 (Easy)
No OS matches for host (If you know what OS is running on it,
        see http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP/IP fingerprint:
TSeq(Class=RI%gcd=1%SI=F4F9)
TSeq(Class=TD%gcd=1%SI=A)
TSeq(Class=TD%gcd=1%SI=B)
T1(Resp=Y%DF=Y%W=4470%ACK=S++%Flags=AS%Ops=M)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=4470%ACK=S++%Flags=AS%Ops=M)
T4(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T7(Resp=N)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=E%UCK=F%ULEN=134%DAT=E)

Nmap run completed -- 1 IP address (1 host up) scanned in 19 seconds
```

3.8 Besitzer eines Prozesses ermitteln

Mit dem Parameter `-I` wird die identd - Information laut RFC 1413 über die laufenden Prozesse auf dem Ziel-System eingeholt. Beispiele:

bei Microsoft Windows 98

```
root@linux / # nmap 192.168.0.3 -I

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on rieekan.matrix.net (192.168.0.3):
Port State Protocol Service Owner
139 open tcp netbios-ssn

Nmap run completed -- 1 IP address (1 host up) scanned in 6 seconds
```

bei SuSE Linux 6.3

```
root@linux / # nmap 192.168.0.1 -I

Starting nmap V. 2.3BETA6 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
Interesting ports on gateway.matrix.net (192.168.0.1):
Port State Protocol Service Owner
21 open tcp ftp mruef
23 open tcp telnet
139 open tcp netbios-ssn rieekan
8080 open tcp http-proxy lanman

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

3.9 zusätzliche TCP- und ICMP- Optionen

3.9.1 TCP - Pings

Diese Funktion ist immer dann anzuwenden, wenn die Erreichbarkeit von Systemen oder Netzwerken identifiziert werden soll und eine Überprüfung mittels ICMP vom Zielsystem nicht zugelassen wird.

Der Parameter `-PT <Port>` verwendet einen TCP-Ping-Scan um zu verifizieren, welche Hosts im Netzwerk erreichbar sind. Diese Option kann auch ohne die Definierung eines Ports verwendet werden (standardmäßig wird HTTP-Port TCP/80 verwendet).

Ansprechbare Systeme sollten auf das gesendete TCP-Datagramm (ACK-Flag gesetzt) mit einem RST antworten.

3.9.2 TCP-SYN-Suchlauf bei TCP-Ping-Scan

Hierbei kann mit der Option `-PS` ein SYN-Suchlauf anstelle des standardmäßigen ACK-Suchlaufs bei TCP - Ping - Scans verwendet werden. Das Setzen des Zielports erfolgt auf die selbe Art und Weise wie bei den zuvor erläuterten TCP - Pings.

3.9.3 TCP reverse ident-Scanning

Mit Option `-I` wird das TCP reverse ident - Scanning aktiviert. Wie Dave Goldsmith in einem Bugtraq-Posting aus dem Jahre 1996 publiziert hat, ermöglicht das ident - Protokoll (RFC 1413) das Identifizieren des Besitzers eines TCP - Dienstes.

3.9.4 ICMP - Ping

Die Option `-PI` nutzt einen klassischen Ping (ICMP echo request), um die Erreichbarkeit von Systemen und Broadcast - Adressen von Subnetzen zu identifizieren. Letztere sind extern erreichbare IP - Adressen, die eine Umwandlung zu einem internen Broadcast des Subnetzes durchführen. Sie stellen die Voraussetzung für eine Reihe von Denial of Service-Attacken (Smurf ist die bekannteste Variante) dar und sollten deshalb verhindert werden.

3.9.5 ICMP timestamp - Anfrage

Die Option -PP benutzt eine ICMP timestamp - Anfrage (Typ 13, Code 0), um ansprechbare Hosts zu finden.

3.9.6 Parallele Ausführung von TCP- und ICMP - Ping

Der Parameter `-PB <Port>` ist der standardmäßig gewählte Ping - Typus. Es werden beide o. g. Techniken (`-PT` und `-PI`) parallel durchgeführt. Auf diese Weise können Firewall - Elemente ausgetrickst werden, die nur eine der beiden Protokolle herausfiltern. Der Zielport wird analog der zuvor erklärten Optionen angegeben.

3.9.7 ICMP address mask request

Es wird ein ICMP address mask request (Typ 17, Code 0) verwendet.

3.10 Ausgabeoptionen

3.10.1 Komplette DNS-Auflösung erzwingen

Die Option `-R` versucht, während eines Scan-Vorgangs alle Host-Namen aufzulösen, und zwar nicht nur die Aktiven. Durch die explizite Anweisung auch inaktive Rechner-Namen in Erfahrung zu bringen, kann diese Funktion unter Umständen einen Durchlauf extrem in die Länge ziehen.

3.10.2 Verbose Option

Im Verbose-Modus, der mit `-v` aktiviert wird und sehr zu empfehlen ist, werden Informationen in ausführlicher Form ausgegeben. Die doppelte Verwendung `-vv` erhöht den Umfang der Ausgabe nochmals. Ebenso kann `-d` einige Male aktiviert werden, falls Sie wirklich vor einem vollem Bildschirm verrückt werden wollen!

3.10.3 Ausgabe in Protokoll-Datei

Mit Parameter `-oN <Protokoll-Dateiname>` werden die Resultate des Scans in einem normalen, für Menschen lesbaren Format abgespeichert. Der Paraemter `-oX` hingeggen protokolliert die Resultate in einer XML-Datei. Sollen die Informationen mit `grep` gefiltert werden, bietet sich der Parameter `-oG` an. Sind alle drei Formate von Belang, so kann man mit Option `-oA <Basisdateiname>` alle Formate bekommen.

Weiterhin existiert die Möglichkeit, mit Parameter `-oS` ein für s|<ripT kiDd|3 lesbares Format auszuwählen.

3.10.4 Protokolldatei erweitern

Mit `--append_output` werden die Scan - Resultate an die spezifizierte Protokoll-Datei angehängt, anstatt diese zu überschreiben.

3.11 Verschleierungsparameter

Ganz offensichtlich kann ein Intrusion Detection System (IDS) nmap - Scans oftmals erkennen. Neben der Deaktivierung der ICMP - Pings sind weitere Manipulationen möglich, die in den folgenden Abschnitten beschrieben werden.

3.11.1 Verbergen der Identität / IP - Source manipulieren

Unter bestimmten Umständen ist `nmap` nicht in der Lage, Ihre Quell - IP - Adresse zu identifizieren (`nmap` wird Ihnen dies mitteilen). In einer solchen Situation kann mit der Hilfe der Option `-S` die IP-Adresse (der gewünschten Schnittstelle) festgelegt werden.

Eine andere Möglichkeit dieser Option ist, die Quelle des Scans zu spoofen, so dass das Zielsystem glaubt, dass jemand anderes den Scan durchführt.

Diese wirklich exzellent ausgedachte und umgesetzte Funktion kann unter Umständen auch für Denial of Service-Attacken genutzt werden (angegebener Host nicht ansprechbar).

3.11.2 Lockvogel

Mit der Option `-D <Decoy1 [,Decoy2][,ME],...>` wird ein so genannter Decoy-Scan (dt. Lockvolgel) veranlasst. Für einen unabhängigen Betrachter sieht es so aus, als würde eine Reihe zusätzlicher Hosts die Zielumgebung scannen. Durch die Option `-D host.fake1.com,ME,test.fake2.com` verfälscht `nmap` die Scan-Pakete mit den Absendeadressen von host.fake1.com und test.fake2.com.

3.11.3 sonstige Verschleierungsversuche

Die Option `--data_length <Anzahl>` legt die Länge der zu versendenden Pakete fest. Normalerweise verschickt `nmap` möglichst kleine Pakete, die lediglich aus dem Header bestehen. So weisen TCP - Datagramme im Normalfall eine Länge von 40 und ICMP echo request-Anfragen 28 Bytes auf. Diese Option weist `nmap` an, die verschickten Pakete um Null-Bytes zu verlängern. Pakete zur Erkennung des Betriebssystens (`-O`) sind im Gegensatz zu Ping-Zugriffen und Portscan-Paketen nicht betroffen. Natürlich verlangsamt sich dadurch das Scannen - aber ebenso erhöht es die Unauffälligkeit des Scans.

Spielt Zeit eine untergeordnete Rolle, können durch Verwendung der Option `-q` ein unauffälliges Verhalten eingestellt werden. (Siehe auch Bugtraq, FTP, ICMP, IP, Linux, Networking, nmap, Ping, Port, RFC 792, RFC 793, RFC 1413, Scanning, Security, Strobe, TCP, UDP, Unix, Windows)

3.12 Sonstige Optionen

3.12.1 Ursprungsort der Ports festlegen

Mit `-g <Portadresse>` wird der Ursprungsport eines Scans definiert. Die Ports 20 und 53 sind erfahrungsgemäss eine gute Wahl.

3.12.2 Auswahl des Device

Mit `-e <Gerätename>` wird bestimmt, an welcher Schnittstelle (Device) die Datenpakete auf die Reise geschickt werden sollen. Je nach dem muss für den Gerätename eth0, eth1 oder ppp0 definiert werden.

3.12.3 IPv6-Unterstützung

IPv6 wir mit der Option `-6` erreicht. Momentan werden nur TCP connect()- und Ping - Scans von `nmap` unterstützt. Sollen UDP- oder andere Scan - Typen genutzt werden, lohnt sich ein Blick auf http://nmap6.sourceforge.net/.

4 ZIEL-SPEZIFIKATION

Alle Angaben, die nmap nicht als Option oder als Argument einer Option interpretiert, wird als Ziel - Spezifikation angesehen. Die einfachste Form ist das Auflisten von einzelnen Hostnamen oder IP-Adressen in der Kommandozeile. Falls Sie ein Subnetz scannen wollen, geben Sie dieses in gebräuchlicher Syntax an (z. B. 192.168.1.0/24 für ein lokales Klasse C-Netzwerk)

nmap verwendet zudem eine sehr mächtige und komfortable Notation zur Spezifikation von IP-Adressbereichen. So kann das Klasse B-Netzwerk 192.168.*.* mit der Angabe von **192.168.*.***, **192.168.0-255.0-255**, **192.168.1-50,51-255.1,2,3,4,5-255** gescannt werden. Falls Sie das Asteriks - Zeichen (dt. Stern, ***) benutzen wollen, denken Sie daran, dass einige Shells einen vorangestellten \ oder ein Auskommentieren mittels Gänsefüßchen verlangen.

Eine weitere Möglichkeit erschließt sich durch die entgegensetze Herangehensweise. Anstatt ein ganzes Klasse B-Netzwerk zu scannen, kann mit der Angabe von ***.*.5.6-7** jede IP - Adresse gescannt werden, die auf .5.6 oder .5.7 endet.

Nutzen einer Eingabedatei.
Mit -i <Eingabedatei> wird aus einer dafür vorgesehenen Datei eine Liste mit IP-Adressen oder Host-Namen eingelesen. Der Parameter -iL <Eingabe-Dateiname> weist nmap an, die Ziel-Spezifizierung ZUERST von der angegebenen Datei einzulesen und erst danach von der Kommandozeile.

5 Beispiele

```
root@linux / # nmap -v ziel.beispiel.com
```

Diese Option scannt alle reservierten TCP-Ports am Ziel-system mit dem Namen ziel.beispiel.com. Das -v aktiviert den Verbose - Modus.

```
root@linux / # nmap -sS -O ziel.beispiel.com/24
```

Hier wird ein stealth SYN-Scan gegen jede der potentiell vorhandenen 255 Maschinen des Klasse C-Netzwerks von ziel.beispiel.com gestartet. Ebenso wird versucht, das Betriebssystem der aktiven Systeme zu ermitteln. Dieser Vorgang erfordert root-Privilegien.

```
root@linux / # nmap -v --randomize_hosts -p 80 '*.*.2.3-5'
```

Manchmal ist es nicht erforderlich, einen IP-Adressbereich zu scannen. So kann es durchaus sein, dass in einer Situation das Absuchen spezieller Geräte nötig wird. Dieses Kommando findet sämtliche Webserver, die eine IP - Adresse aufweisen, die auf .2.3 .2.4 oder .2.5 endet. Es könnten noch mehr interessante Systeme gefunden werden, wenn bei 127 gestartet wird (IMHO). In diesem Fall können die durch die Sterne gegebenen Platzhalter durch **127-222** ersetzt werden.

```
root@linux / # host -l firma.com | cut '-d ' -f 4 | ./nmap -v -iL -
```

Führt einen DNS-Zonetransfer durch, um sämtliche Hosts von firma.com zu finden. Die Ausgabe der IP - Adressen wird sodann für die weitere Verarbeitung zu nmap umgeleitet.

Mit den folgenden beiden Befehlen sollen diejenigen Maschinen identifiziert werden, bei denen named als root läuft:

```
root@linux / # nohup nmap -r -iR -I -sT -p53 > /tmp/named-scan.out & tail
-f /tmp/named-scan.out
```

Die Option `-r` scannt die Ports der Zielmaschine in einer zufälligen Reihenfolge, `-iR` wählt zufällige IP's als Ziel der Scans aus, `-I` aktiviert den reverse ident scan, der nur mit einem TCP-connect()-Scan (`-sT`) funktioniert. `-p53` definiert schließlich Port 53 als Scanziel.

In einem weiteren Beispiel soll target.host gescannt werden, allerdings so, dass wir dabei unentdeckt bleiben. Dazu werden einige Lockvögel (Decoy Hosts) benutzt:

```
root@linux / # nohup nmap -r -P0 -sS -Ddecoy1,decoy2,decoy3,decoy4,decoy5
target.host
```

Die Hosts decoy1 bis decoy5 sollten erreichbar sein. Die Option -P0 deaktivert das an-ping-en von target.host vor dem Scan. -sS aktiviert den SYN-Scan und -Ddecoy1,decoy2,decoy3,decoy4,decoy5 benutzt die Hosts decoy1 bis decoy5 um ein wenig Verwirrung zu stiften.

6 Letzter Kommentar

Ein empfehlenswertes Print - Medium zu Scanning und nmap ist **Das Anti-Hacker-Buch** (ISBN 3-8266-4072-1), erschienen im mitp-Verlag. Dank für freundliche Hilfe und kooperative Zusammenarbeit mit mir an diesem Dokument ernten in erster Linie Snakebyte, Anthraxx und DukeCS von Kryptocrew.