BRAIN-MEDIA.DE

PROFESSIONELLES INTERNET- UND OPEN-SOURCE-KNOW-HOW

Holger Reibold

Nmap kompakt

Gratis! Zwei E-Books zum Security Scanning zum Download

Security.Edition

Praxiseinstieg in die Netzwerkerkennung und das Security Scanning

Holger Reibold

Nmap kompakt

© 2015 Brain-Media.de

Herausgeber: Dr. Holger Reibold

Umschlaggestaltung: Brain-Media.de

Satz: Brain-Media.de

Coverbild: U5 / photocase.de

Korrektur: Theresa Tting

Druck: COD

ISBN: 978-3-95444-237-9

Inhaltsverzeichnis

VORWORT **7**

1 NMAP – DER EINSTIEG **9**

1.1 Nmap in Betrieb nehmen **11**

1.2 Erste Schritte mit Nmap **14**

2 NMAP KENNENLERNEN **25**

2.1 Ziele für Nmap **25**

2.2 Host erkennen **29**
2.2.1 List-Scan 30
2.2.2 Ping-Scan 31
2.2.3 TCP-ACK-Ping 32
2.2.4 UDP-Ping 33
2.2.5 ICMP-Ping-Arten 34
2.2.6 IP-Protokoll-Ping 34
2.2.7 ARP-Ping 35
2.2.8 Traceroute 35
2.2.9 DNS-Auflösung 35

2.3 Port-Scanning in der Praxis **36**

2.4 Scan-Tutorial **39**

2.5 Port-Scan-Techniken **44**
2.5.1 TCP-SYN-Scan 45
2.5.2 TCP-Connect-Scan 46
2.5.3 UDP-Scan 46
2.5.4 TCP-NULL-, FIN- und Xmas-Scans 47
2.5.5 TCP-ACK-Scan 48

2.5.6 TCP-Window-Scan 48
2.5.7 TCP-Maimon-Scan 49
2.5.8 Benutzerdefinierter TCP-Scan 49
2.5.9 Idle-Scan 49
2.5.10 IP-Protokoll-Scan 50
2.5.11 FTP-Bounce-Scan 51

2.6 Port-Auswahl 52

3 ERMITTLERFUNKTIONEN 55

3.1 Services ermitteln 55

3.2 Betriebssystem ermitteln 59

4 AUSFÜHRUNG OPTIMIEREN 61

4.1 Bessere Performance 61

4.2 Firewall und IDS umgehen 65

4.3 Berichtausgabe 68

5 NMAP IN DER PRAXIS 73

5.1 Webserver scannen 74
5.1.1 HTTP-Methoden 74
5.1.2 Offener Web-Proxy 75
5.1.3 Interessante Dateien und Verzeichnis aufdecken 76
5.1.4 Brute-Force-Attacke 78
5.1.5 Benutzer-Accounts auslesen 79
5.1.6 Zugangsdaten testen 80
5.1.7 Brute-Force-Attacke gegen WordPress 81
5.1.8 Brute-Force-Attacke gegen Joomla! 82
5.1.9 Web Application Firewall erkennen 83
5.1.10 Schwachstellen aufdecken 83

5.2 Test von Datenbanken ... 87
5.2.1 MySQL-Datenbanken abrufen ... 87
5.2.2 MySQL-Benutzer auslesen ... 88
5.2.3 MySQL-Variablen auslesen ... 88
5.2.4 Root-Account finden ... 89
5.2.5 Brute-Force-Attacke gegen MySQL ... 90
5.2.6 Unsichere MySQL-Konfigurationen ... 90

5.3 Mailserver im Visier ... 92
5.3.1 E-Mail-Accounts aufdecken ... 92
5.3.2 Offene Relays aufspüren ... 94
5.3.3 SMTP-Passwort knacken ... 94
5.3.4 SMTP-User auslesen ... 95
5.3.5 POP3-Server attackieren ... 95
5.3.6 IMAP-Server attackieren ... 96

6 MIT ZENMAP ARBEITEN ... 97

6.1 Scannen und auswerten ... 98

6.2 Netzwerktopologien ... 106

6.3 Der Profileditor ... 111

6.4 Erweiterte Zenmap-Funktionen ... 113

7 EIGENE TEST-SKRIPTS ... 115

7.1 Basics ... 115

7.2 Skript-Struktur ... 117

7.3 Skript-Kategorien ... 119

7.4 Gruß an die Welt! ... 121

7.5 Feinschliff ... 124

ANHANG A – MORE INFO 127

ANHANG B – EIGENE TESTUMGEBUNG 129

INDEX 131

WEITERE BRAIN-MEDIA.DE-BÜCHER 137

Weitere Titel in Vorbereitung 140

Plus+ 140

Vorwort

IT- und Systemadministratoren müssen heute immer komplexer werdende Infrastrukturen permanent auf Schwachstellen und Sicherheitslücken überprüfen. Das Aufdecken von Schwachstellen, das Testen der Anfälligkeit und das Schließen sind heute essentielle administrative Aufgaben.

Fast täglich kann man in den Medien von erfolgreichen Hacker-Attacken hören. Prominentes Opfer war im Sommer 2015 das Netzwerk des Bundestages, das – vermeintlich aus Russland – gehackt worden sein soll. Das BSI, das für die Wartung und die Sicherheit dieses Netzwerks zuständig ist, blamierte sich in diesem Zusammenhang, weil man weder in der Lage war, das Netzwerk ausreichend zu schützen, noch zeitnah eine sichere Umgebung herzustellen.

Solch prominente Geschehnisse sind nur die Spitze eines Eisbergs. Tag für Tag werden Millionen Hacker-Attacken gefahren. Manchmal sind es nur Skript-Kiddies, die ihre erworbenen Hacker-Fähigkeiten testen, doch die überwiegende Anzahl der Attacken dürfte einen kriminellen Hintergrund haben. Oftmals geht es um Wirtschaftsspionage.

Wenn auch Sie für die Sicherheit eines Netzwerks zuständig sind, müssen Sie dieses kontinuierlich auf Sicherheitslücken und sonstige Schwachstellen hin überprüfen. Fachleute sprechen von Penetrationstests. Sie dienen dazu, Netzwerkkomponenten auf bekannte Schwächen hin zu überprüfen.

Ihr Ziel muss es sein, potenziellen Hackern zuvorzukommen. Das Zauberwort lautet dabei: Waffengleichheit. Nur dann, wenn Sie wissen, wie Hacker vorgehen und welche Tools sie dabei einsetzen, sind sie in der Lage, ihnen mit gleichen Mitteln zu begegnen. Dabei sind Sie potenziellen Angreifern klar im Vorteil, denn Sie kennen die kritischen Infrastrukturkomponenten, die Netzwerk-Topologie, potenziellen Angriffspunkte, die ausgeführten Services und Server etc.

Um Ihre eigene Infrastruktur so sicher wie möglich zu machen, müssen Sie immer und immer wieder folgende Schritte ausführen:

1. Identifizierung von Schwachstellen und deren Risiko.
2. Praktische Ausnutzung und Testen der Schwachstellen in einer gesicherten Umgebung.
3. Tests in einer realen Umgebung.
4. Schließen von gefundenen Schwachstellen.

Wenn Sie bei Punkt 4 angelangt sind, fängt alles wieder von vorne an – ein permanenter Kreislauf. Wenn Sie diese Schritte verinnerlichen und kontinuierlich die Sicherheit kritischer Systeme im Blick haben, wird Ihre Umgebung mit jeder Maßnahme sicherer. Das wiederum spart Ihnen langfristig viel Zeit und Ärger, denn Sie geben Hackern kaum eine Chance, ihr Unwesen zu treiben.

Sie können das Ganze auch sportlich betrachten und als Spiel sehen. Jeder hat dabei seine Mittel: Mitspieler, technische Geräte und Techniken. Am Ende ist nur wichtig, dass Sie als Sieger vom Platz gehen.

Neben dem theoretischen Wissen um die Relevanz des Penetration Testing benötigten Sie natürlich auch ein geeignetes Werkzeug. Mit Nmap steht Ihnen ein Klassiker zur Verfügung, der in jeden Admin-Werkzeugkasten gehört. Nmap (Network Mapper) ist von Haus aus ein Werkzeug für die Ermittlung von Netzwerkkomponenten und Diensten sowie der Auditierung.

Das Programm unterstützt Administratoren bei der Inventarisierung, dem Verwalten von Services und dem Monitoring von Services und Hosts. Nmap verwendet dabei IP-Pakete in einer neuen Art und Weise, um die Verfügbarkeit und Erreichbarkeit zu prüfen. Dabei kann der Netzwerk-Mapper verschiedenste Informationen von den gefundenen Hosts ermitteln.

Doch damit nicht genug: Nmap kann auf allen relevanten Betriebssystemen ausgeführt werden, insbesondere unter Mac OS X, Linux und Windows. Die klassische Ausführung erfolgt dabei auf der Konsole. Alternativ steht Anwendern mit Zenmap eine komfortable GUI zur Verfügung. Mit Ncat steht Ihnen ein weiterer Helfer zur Verfügung, mit dem Sie Daten transferieren, umleiten und debuggen können. Wenn Sie die Scan-Ergebnisse vergleichen wollen, greifen Sie zu Ndiff. Und dann steht Ihnen mit Nping ein weiteres Hilfsprogramm für das Generieren von Paketen und der Antwortanalyse zur Verfügung.

In diesem Einstieg lernen Sie die wichtigsten Funktionen von Nmap kennen. Bleibt mir nur noch, Ihnen viel Spaß und Erfolg beim Einstieg in die Welt der Netzwerkermittlung und dem Security Scanning zu wünschen!

Herzlichst,

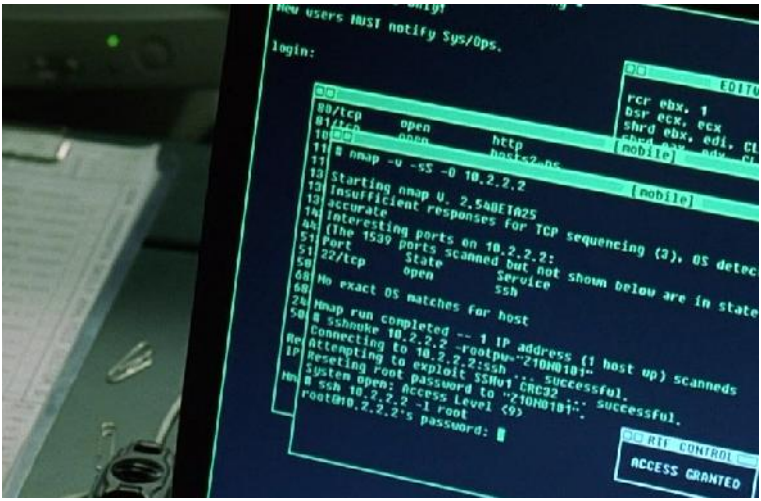
Holger Reibold

(Oktober 2015)

1 Nmap – der Einstieg

Ohne Computer, ohne Netzwerke, ohne Internet und ohne mehr oder minder aufwändige Server-Anwendungen ist unser Alltag kaum mehr vorstellbar. Die Computer- und Netzwerktechnik ist allgegenwärtig, sei es am Arbeitsrechner, im Handy, industriellen Produktionsmaschinen oder dem Auto. Und dieser Trend wird sich vorsetzen und weiter verstärken, bis jedes scheinbar noch so unwichtige Teil unseres Lebens in irgendeiner Weise vernetzt ist.

Die Computertechnik kommt längst nicht mehr nur in technologisch geprägten Unternehmen zum Einsatz, sondern kommt selbst in kleinen Schreinereien oder in jedem Müllauto der städtischen Müllabfuhr zur Müllerfassung zum Einsatz. Auch in unseres Privatleben wird die Technik immer weiter vordringen: Abgesehen von der allgemeinen Internet- und Smartphone-Nutzung denke man nur an das Stichwort Smart Home, das vernetzte Zuhause.

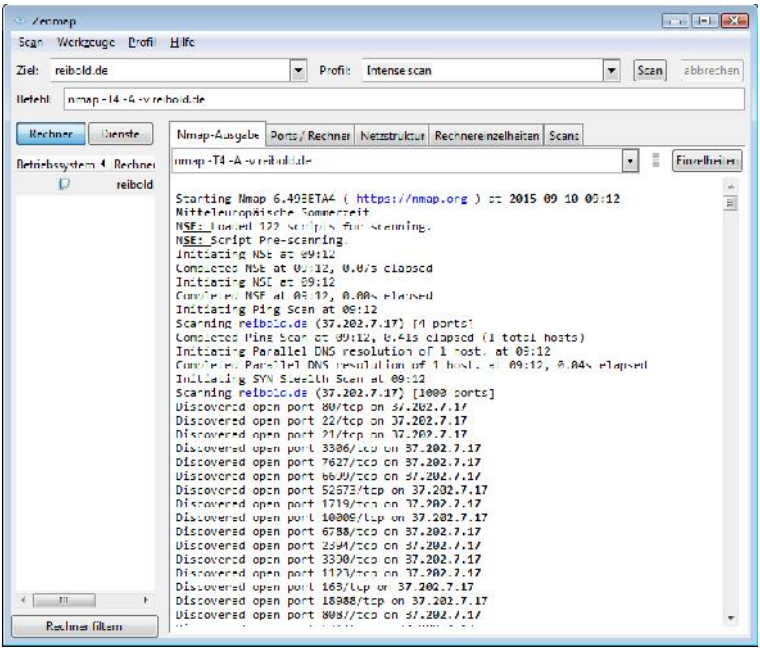


Starker Auftritt: Nmap in *Matrix Reload*.

Die Computertechnik bringt uns im Alltag viele Vorteile, aber je intensiver wir sie nutzen und um so mehr wir uns von ihr abhängig machen, umso angreifbarer sind

wir. Der Preis für die Produktivitätsgewinn und die Abhängigkeit ist hoch: Fallen kritische Systeme aus oder werden Daten zerstört oder entwendet, belaufen sich die Schäden oftmals in Dimensionen, die auch ein solides Unternehmen schnell ins Wanken bringen können.

Für jeden Betreiber von kritischen Computer- und Netzwerkkomponenten ist es daher unverzichtbar, die Sicherheit der Systeme zu prüfen, Schwachstellen zu identifizieren, um diese dann im nächsten Schritt zu schließen. Auf der Suche nach Schwachstellen und möglichen Angriffspunkten sind die Ports der zweite wichtige Ansatzpunkt. Um mehr über die Ports, deren Verwendung und deren Status zu erfahren, benötigen Sie einen Portscanner. Der Klassiker unter diesen Werkzeugen ist Nmap. Aber Nmap kann noch weit mehr.



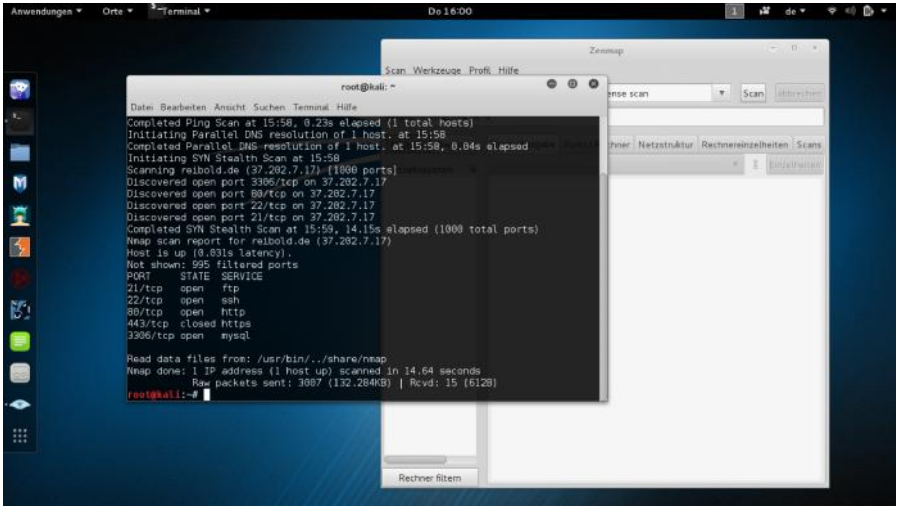
Der Einsatz der Nmap-GUI Zenmap unter Windows.

Anfang September 1997 veröffentlichte Gordon “Fyodor” Lyon die erste Version des Security Scanners Nmap. Damals bestand das Programm gerade einmal aus drei Dateien mit einer gesamten Code-Länge von 2.000 Zeilen. Die erste Version unterstützt lediglich Linux-Betriebssysteme.

Angesicht des holprigen Starts war nicht abzusehen, dass Nmap einmal der populärste Network Security Scanner werden würden (auch dank einer starken Community). Im Laufe der Jahre hat Nmap funktional stark zugelegt und es kamen verschiedene erweiterte Funktionen wie die Remote-Betriebssystemerkennung und die Nmap Scripting Engine hinzu.

Nmap kam auch in einigen weltweit erfolgreichen Filmen zum Einsatz, beispielsweise in *Matrix Reloaded*, das *Bourne Ultimatum* und in *Die Hard Teil 4*. Auch in dem deutschen Cyber-Thriller *Who Am I - Kein System ist sicher* hat Nmap einen Gastauftritt.

Nmap ist so konzipiert, dass das Programm schnell große Netzwerke scannen kann, aber auch einzelne Hosts genau unter die Lupe nimmt. Dabei ist Nmap nichts für schwache Nerven: Das Programm unterstützt mehr als 100 Kommandozeilenoptionen. Das dürfte selbst für routinierte Netzwerk-Gurus mehr als genug sein.



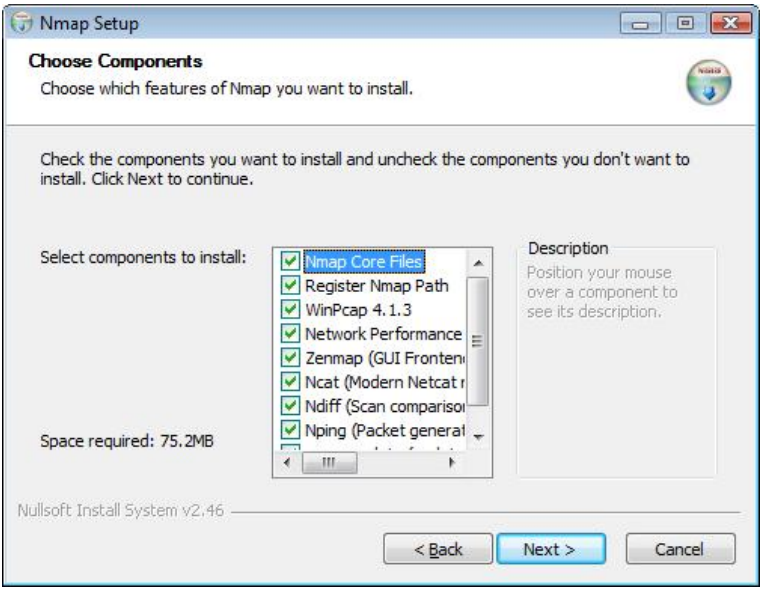
Ein Muss für Penetration Tester: Kali Linux.

1.1 Nmap in Betrieb nehmen

Bevor Sie in den Genuss der vielen Möglichkeiten gelangen, die Ihnen Nmap bietet, müssen Sie das Programm in Betrieb nehmen. Sicherheitsexperten und Penetration Tester greifen meist zu Kali Linux, eine spezielle Linux-Distribution, die Hunderte Tools für das Testen von Infrastrukturkomponenten enthält.

In Kali Linux sind (natürlich) auch Nmap und die GUI Zenmap vorinstalliert. Die Konsolenvariante starten Sie einfach im Terminal. Auch Zenmap greifen Sie über das *Anwendungen*-Menü *Informationsbeschaffung* > *Zenmap* zu. Wir kommen in Kapitel 6 detailliert auf Zenmap zu sprechen.

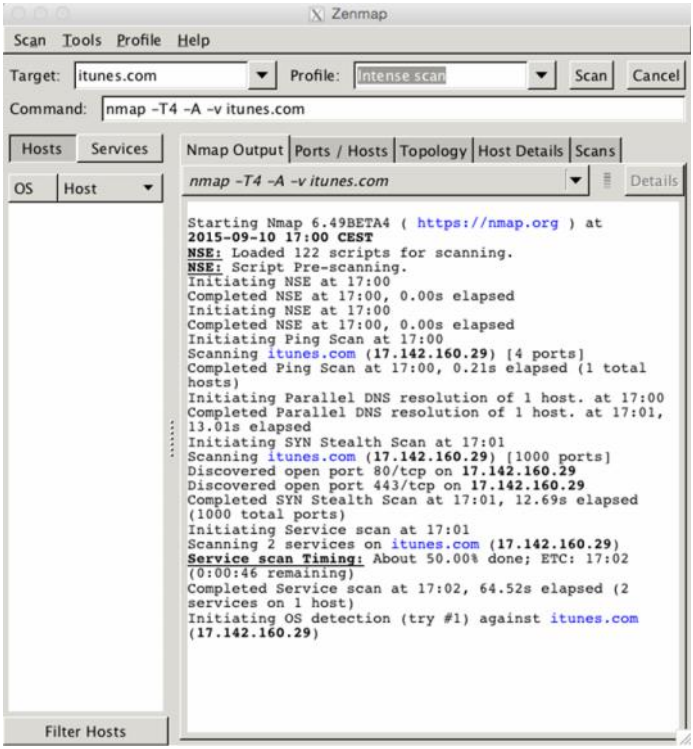
Sie können Nmap natürlich auch auf jedem anderen Linux-Betriebssystem installieren, sofern es dort nicht vorinstalliert ist. Der einfachste Weg: Sie greifen zum Paketmanager des jeweiligen Betriebssystems. Alles andere ist einfach. Den Umweg über das manuelle Installieren können Sie sich in der Regel schenken.



Die Installation von Nmap unter Windows.

Die Inbetriebnahme von Nmap unter Windows ist kinderleicht. Laden Sie sich einfach von der Projekt-Site (*https://nmap.org/download.html*) das aktuelle Installationsprogramm herunter. Diesem Buch liegt Version 6.49 zugrunde. Das zugehörige Installationspaket trägt die Bezeichnung *nmap-6.49-setup.exe*. Starten Sie das Installationsprogramm und folgen Sie den Anweisungen am Bildschirm. In der Regel können Sie einfach immer mit *Next* einen Schritt nach dem anderen abarbeiten. Das Interessante an der Windows-Version: Neben der Konsolenvariante und den verschiedenen oben genannten Tools kann das Installationsprogramm in der Windows-Registry verschiedene Einstellungen bearbeiten, um die Scan-

Performance zu optimieren. Außerdem ist ein Update-Mechanismus verfügbar, der im Bedarfsfall architekturunabhängige Dateien aktualisieren kann.



Zenmap unter Mac OS X.

Unter Penetration Testern ist Mac OS X aufgrund seiner hohen Stabilität eine beliebte Plattform. Auch hier ist die Inbetriebnahme einfach. Einzige Voraussetzung: Die Zenmap-Komponenten setzt eine bestehende X11-Installation voraus. Anschließend können Sie das DMG-Paket von der Projekt-Site herunterladen und ausführen. Da das Paket von dem Betriebssystem als unsicher klassifiziert wird, müssen Sie zunächst die *Crtl*-Taste drücken, dann die Installationsdatei markieren und mit dem Befehl *Öffnen* die Installation starten. Folgen Sie anschließend den Anweisungen am Bildschirm. Nmap ist anschließend über das Terminal verfügbar, die Nmap-GUI Zenmap finden Sie unter *Programme*.

1.2 Erste Schritte mit Nmap

Nmap stammt wie bereits erwähnt ursprünglich aus dem Linux-Umfeld. Daher wird es Sie nicht weiter verwundern, dass der Scanner üblicherweise auf der Konsole verwendet wird. Die Ausführung des Netzwerk-Analysewerkzeugs und Sicherheits-/Port-Scanners erfolgt nach diesem Schema:

```
nmap [ <Scan-Typ>...] [ <Optionen> ] { <Ziel-Spezifikation> }
```

Nmap gibt dabei eine Liste der gescannten Hosts inklusive verschiedener Zusatzinformationen aus. Welche Zusatzinformationen dies sind, ist von den verwendeten Optionen und natürlich von den Zielen abhängig. Die Scan-Ergebnisse werden in Tabellenform ausgegeben. Dort werden konkret der Port und das Protokoll sowie dem Dienstnamen und der Zustand aufgeführt. Mögliche Zustände sind:

- offen
- gefiltert
- geschlossen
- ungefiltert

Die Statusangaben werden durch den jeweiligen englischsprachigen Begriff angezeigt, also *open*, *filtered*, *closed* und *unfiltered*.

Was bedeuten diese Statusinformationen nun konkret? Der Status *offen* bedeutet, dass auf dem Port des Zielrechners eine Anwendung auf eingehende Verbindungen/Pakete lauscht. Dabei handelt es sich oftmals um webbasierte Anwendungen, die über Port 80 und 443 angesprochen werden.

Der Status *gefiltert* zeigt an, dass eine Firewall, ein Filter oder ein anderer Dienst den Port blockiert. In einem solchen Fall kann Nmap nicht in Erfahrung bringen, ob der Port geschlossen oder doch offen ist.

Auf geschlossenen Ports wird keine Anwendung ausgeführt, die auf eingehende Requests wartet. Dann gibt es noch den Status *ungefiltert*. Kann Nmap nicht feststellen, ob es sich um einen offenen oder geschlossenen Port handelt, wird er als *ungefiltert* klassifiziert.

Gelegentlich begegnen Sie auch den Zustandskombinationen *offen|gefiltert* und *geschlossen|gefiltert*. Diese Statusmeldungen zeigen an, dass Nmap nicht feststellen kann, in welchem der beiden Zustände sich ein Port befindet.

Ob mit der Port-Tabelle auch weitere Informationen wie die Software-Version ausgegeben werden, ist von der verwendeten Scan-Konfiguration abhängig. Auch das verwendete Betriebssystem, der Gerätetyp und die MAC-Adresse können mit Nmap abgerufen werden.

Besonders bequem ist die Scan-Konfiguration mit Hilfe von Zenmap. Wenn Sie einen intensiven Scan (intense scan) ausführen, verwendet Nmap folgende Konfiguration:

```
nmap -T4 -A -v server.de
```

In diesem ersten Beispiel kommen zwei wichtige Optionen zum Einsatz: *-T4* sorgt für eine schnellere Ausführung, *-A* für die Betriebssystem- und Versionserkennung, Script-Scanning und Traceroute.

Wenn Sie Nmap ohne irgendeine Argument ausführen, gibt das Programm die Hilfe aus, der Sie wichtigste Informationen entnehmen können. In diesem einführenden Kapitel werfen wird einen kurzen Blick auf die Argumente, die Sie kennen sollten. Im weiteren Verlauf dieses Einstiegs werden wir dann einen genaueren Blick auf die verschiedenen Optionen werfen.

Eine zwingende Angabe für eine Scan ist die Ziel-Spezifikation. Sie können dabei einen Host-Namen, eine IP-Adresse, ein Netzwerk oder ein Netzwerksegment angeben. Oben haben Sie bereits ein Beispiel kennengelernt. Alternative Argumente sind beispielsweise folgende:

```
192.168.0.1
```

```
10.0.0-255.1-254
```

```
server.de/24
```

```
scanme.nmap.org
```

Wenn Sie neu sind in der Welt der Netzwerk- und Sicherheitsscanner, sollten Sie auf keinen Fall Ihre aktuellen Produktivitätssysteme unter Beschuss nehmen. Denn noch wissen Sie ja garnicht, welchen Schaden Sie womöglich mit der Ausführung einer bestimmten Konfiguration anrichten können. Das Nmap-Entwicklerteam stellt Ihnen daher einen eigenen Server zur Verfügung, den Sie gefahrenlos unter die Lupe nehmen können:

```
scanme.nmap.org
```

Wenn Sie ein wenig Erfahrung mit Nmap gesammelt haben, können Sie auch eine lokale Testumgebung aufsetzen und damit beispielsweise eigene kritischen Infrastrukturkomponenten abbilden und diese dann mit Nmap unter die Lupe nehmen (siehe Anhang B).

Anstelle einer IP-Adresse oder eines ganzen Netzwerksegments können Sie auch eine Liste mit Hostnamen und Netzwerken verwenden. Die hierfür zuständige Option:

```
-iL <dateiname>
```

Um eine zufällige Auswahl von Zielen zu scannen, verwenden Sie folgendes Argument:

```
-iR <hosts>
```

Sie können außerdem Hosts und Netzwerke explizit vom Scannen ausnehmen. Dabei können Sie die Hosts einzeln oder in Form einer Liste angeben. Auf diesem Weg können Sie gezielt bestimmte Systeme aus einem Scan-Vorgang ausnehmen:

```
--exclude <host1[,host2][,host3],...>
```

```
--excludefile <ausschlussdatei>
```

Ein weiteres Highlight von Nmap sind die umfangreichen Möglichkeiten zur Host-Ermittlung. Um die Liste der Ziele auszugeben, verwenden Sie das Argument *-sL*.

Um den Ping-Port-Scan zu deaktivieren, verwenden Sie die Option *-sn*. Sie können die Host-Ermittlung deaktivieren. Dann geht Nmap davon aus, dass alle Hosts online sind. Das hierfür relevante Argument lautet *-Pn*. Sie können auch die DNS-Auflösung aktivieren bzw. deaktivieren. Das hierfür zuständige Argument sieht wie folgt aus:

```
-n/-R
```

Wollen Sie bestimmte DNS-Server für die Namensauflösung verwenden, spezifizieren Sie diese wie folgt:

```
--dns-servers <serv1[,serv2],...>
```

Um den Weg der Datenpakete zu verfolgen, verwenden Sie die Option *-traceroute*. Allerdings ist dabei zu beachten, dass nicht immer exakt der zurückgelegte Weg nachgebildet wird.

Auch für die Durchführung von Scans stehen Ihnen umfangreiche Steuer- und Konfigurationsmöglichkeiten zur Verfügung. Mit den folgenden Argumenten führen Sie TCP SYN-/Connect()-/ACK-/Window- und Maimon-Scans durch:

```
-sS/sT/sA/sW/sM
```

Eine UDP-Scan starten Sie mit folgender Option:

```
-sU
```

Mit Nmap können Sie auch TCP-Null-, FIN- und Xmas-Scans durchführen. Was das alles genau ist, erfahren Sie später.

```
-sN/sF/sX
```

Sie können auch TCP-Scan-Flags anpassen:

```
--scanflags <flags>
```

Wenn Sie mit Nmap einen SCTP INIT- oder COOKIE-ECHO-Scan durchführen wollen, verwenden Sie folgende Schalter:

```
-sY/sZ
```

Einen IP-Protokoll-Scan initiieren Sie wie folgt:

```
-sO
```

Um einen FTP-Bounce-Scan durchzuführen, verwenden Sie das folgende Argument:

```
-b <FTP Relay Host>
```

Nicht minder beeindruckend ist die Vielfalt an Möglichkeiten, die Ihnen Nmap für das Port-Spezifikation und die Scan-Reihenfolge zur Verfügung stellt. Um ledig einen bestimmten Port-Bereich zu scannen, geben Sie diesen wie folgt an:

```
-p <port-bereich>
```

Hier einige Beispiele:

```
-p24; -p1-65555; -p U:52,111,137,T:21-25,80,139,8080,S:9
```

Auch der Ausschluss von Ports ist möglich. Die entsprechende Spezifikation sieht wie folgt aus:

```
--exclude-ports <port-bereich>
```

Wenn Sie einen ersten Schnelldurchlauf starten wollen, verwenden Sie den Schnellmodus. Dabei werden weniger Ports als beim Standard-Scan durchgeführt:

```
-F
```

Um die angegebenen Ports anstelle einer zufälligen Reihenfolge der Reihe nach zu testen, verwenden Sie das folgende Argument:

```
-r
```

Sie können einen Scan-Vorgang auch einfach auf die am häufigsten verwendeten Ports beschränken:

```
--top-ports <Zahl>
```

Nmap bietet Ihnen verschiedenste Optionen für das Erkennen von Service und Versionen. Um die auf offenen Ports laufende Dienste und deren Versionen zu bestimmen, verwenden Sie folgendes Argument:

```
-sV
```

Wenn Sie sich für die Details eines Scan-Vorgangs interessieren, verwenden Sie folgendes Argument, das insbesondere für das Debugging interessant ist:

```
--version-trace
```

Oftmals interessiert man sich auch das Betriebssystem auf dem Zielsystem identifizieren. Die Betriebssystemerkennung aktivieren Sie mit folgendem Argument:

```
-O
```

Sie können die Betriebssystemerkennung auch auf vielversprechende Ziele beschränken:

```
--osscan-limit
```

Um das Betriebssystem aggressiver zu bestimmen, verwenden Sie folgendes Argument:

```
--osscan-guess
```

Sie können die Ausführung aus verschiedene Zeit- und Performance-Parameter definieren. Dabei wird der Wert Zeit in Millisekunden (ms), Sekunden (s), Minuten (m) oder Stunden (h) angegeben.

Was Sie bislang noch nicht wissen können: Nmap besitzt sogenannte Timing-Templates. Deren Ausführung bestimmen Sie mit folgendem Argument, wobei eine höherer Wert für eine schnellere Ausführung steht:

```
-T<0-5>
```

Sie können die minimale und maximale Größe von Hostgruppen bestimmen, die parallel geprüft werden:

```
--min-hostgroup/max-hostgroup <größe>
```

Den Timeout-Wert für die Host-Prüfung bestimmen Sie mit folgendem Argument:

```
--host-timeout <zeitspanne>
```

Die Zeitspanne zwischen zwei Tests bestimmen Sie wie folgt:

```
--scan-delay/--max-scan-delay <zeit>
```

Firewalls und Intrusion Detection-Systeme dienen dazu, unerwünschten Traffic in einem Netzwerk zu verhindern bzw. mögliche Attacken und deren Vorbereitungen zu erkennen. Dazu gehören auch Port-Scans, wie sie von Nmap durchgeführt werden. Auch hierfür stehen Ihnen verschiedene Konfigurationsparameter und Einstellungen zur Verfügumg. Sie können beispielsweise Pakete fragmentieren und den MTU-Wert bestimmen:

```
-f; --mtu <wert>
```

Um einen Scan mit einem Köder zu verbergen, verwenden Sie folgendes Argument:

```
-D <koeder1,koeder2...>
```

Nmap erlaubt auch das Spoofen, also Fälschen, der eigenen IP-Adresse:

```
-S <IP-Adresse>
```

Um eine spezifische Schnittstelle zu verwenden, bestimmen Sie diese mit folgendem Argument:

```
-e <interface>
```

Entsprechend können Sie auch den Port vorgeben:

```
-g/--source-port <port-nummer>
```

Auch wenn Nmap eigentlich kein Exploit-Werkzeug ist, um etwaige Schwachstellen zu testen, können Sie dennoch einen benutzerdefinierten Payload an die Pakete anhängen:

```
--data <hex string>
```

Sie können auch ASCII-Zeichenfolgen an die Pakete hängen:

```
--data-string <string>
```

Auch das Fälschen einer MAC-Adresse ist einfach:

```
--spoof-mac <mac-addresse/präfix/hersteller>
```

Sie können auch das Ausgabeformat bestimmen, in dem die Scan-Ergebnisse ausgegeben werden. Um die Ausgabe im Textformat oder im XML-Format zu erhalten, verwenden Sie folgende Argumente (die Ausgabe wird in die entsprechende Datei geschrieben:

```
-oN/-oX <datei>
```

Um das Scan-Ergebnis in allen drei wichtigen Formaten zu schreiben, verwenden Sie folgendes Argument:

```
-oA <ausgabedatei>
```

Um die Geschwätzigkeit der Ausgabe zu erhöhen, verwenden Sie folgende Option (wobei Sie diese mit *-vv* etc. weiter erhöhen können):

```
-v
```

Den Debug-Level können Sie mit folgendem Schalter erhöhen (auch hier können Sie diesem mit *-dd* etc. weiter erhöhen):

-d

Um die Anzeige lediglich auf offene Ports zu beschränken, verwenden Sie dieses Argument:

```
--open
```

Sie können auch alle gesendeten und empfangenen Pakete anzeigen:

```
--packet-trace
```

Um einen unterbrochenen Scan-Vorgang fortzusetzen, verwenden Sie folgende Option:

```
--resume <dateiname>
```

Nmap kann die XML-Aufgabe auch mit Angabe eines XSL-Stylesheets in ein HTML-Dokument schreiben:

```
--stylesheet <pfad/URL>
```

Zum Abschluss dieses einleitenden Kapitels schauen wir uns zwei einige weitere Argumente an. Wenn Sie das IPv6-Scanning aktiveren wollen, verwenden Sie dieses Argument:

```
-6
```

Um die Betriebsystem- und Versionserkennung, das Skript-Scanning und Traceoute auf einen Schlag zu verwenden, führen Sie Nmap mit diesem Argument aus:

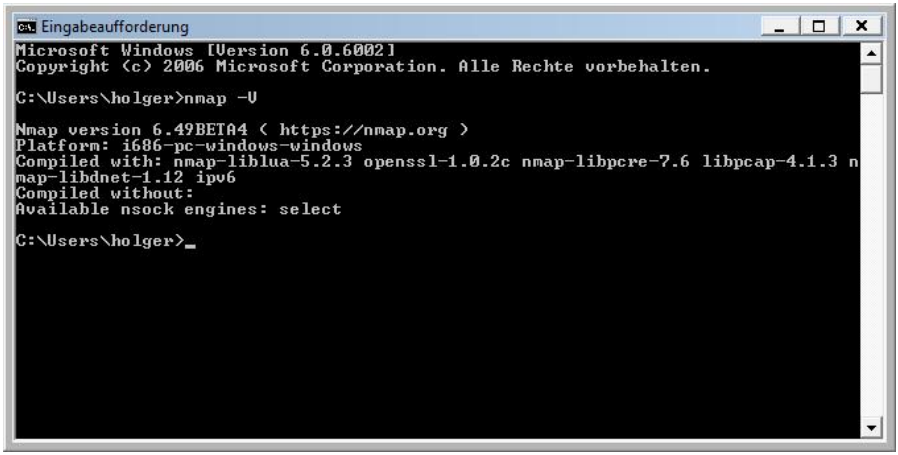
```
-A
```

Sie können auch ein speziellem Verzeichnis für die Nmap-Daten angeben:

```
--datadir <verzeichnisname>
```

Um anzunehmen, dass der Benutzer die vollständigen Rechte besitzt, geben Sie Nmap das folgende Argument mit auf den Weg:

```
--privileged
```



Die Ausgabe der verwendeten Programmversion mit dem Argument *-V*.

Ein letztes Argument sollten Sie für den Einstieg noch kennen, die Versionsausgabe:

```
-V
```

Damit haben Sie einen ersten Überblick über das, was Sie mit Nmap anstellen können. Sie haben auch einen ersten Eindruck erhalten, wie flexibel Sie bei der Verwendung der Optionen sind, die Sie ja miteinander kombinieren können. Im weiteren Verlauf dieses Buches werden wir diese Möglichkeiten weiter vertiefen.

2 Nmap kennenlernen

Im ersten Kapitel haben Sie einen ersten Überblick über die vielfältigen Möglichkeiten erhalten, die Nmap zu bieten hat. In diesem zweiten Kapitel steigen wir weiter in die Verwendung des Scanners ein. Dabei interessieren uns insbesondere die Zielangabe und die Durchführung von Port-Scans.

2.1 Ziele für Nmap

Essentiell für den erfolgreichen Einsatz von Nmap ist die korrekte Verwendung von Argumenten, Optionen, Parametern und Werten. Manchmal muss man bestimmte Angaben mit anderen kombinieren, dann sind wieder andere Eigenheiten des Programms zu berücksichtigen.

Wenn Sie Nmap eine Eingabe ohne Angaben von Argumenten übergeben, dann interpretiert das Programm diese Information als die Bezeichnung eines Zielhosts. Im aller einfachsten Fall scannt Nmap dann die angegebene IP-Adresse oder den namentlich genannten Host.

In der Regel will man aber nicht nur ein System, sondern mehrere Hosts scannen, beispielsweise ein gesamtes Subnetz. In diesem Fall verwenden Sie am einfachsten die Adressenangabe im CIDR-Stil. Dazu erweiteren Sie die IPv4-Adresse oder den Hostnamen um den Zusatz */<numbits>* an.

Bei einer solchen Konfiguration scannt Nmap all die IP-Adressen, bei denen die ersten <*numbits*> mit denen der gegebenen IP oder des gegebenen Hostnamens übereinstimmen.

Hier ein Beispiel für die Ausführung diese Vorgehensweise:

```
nmap -T4 -A -v 192.168.2.1/24
```

In diesem Fall werden alle Hosts des Ziel-Netzwerks gescannt:

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-13
09:29 Mitteleuropäische Sommerzeit
NSE: Loaded 122 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 09:29
Completed NSE at 09:29, 0.00s elapsed
```

```
Initiating NSE at 09:29
Completed NSE at 09:29, 0.00s elapsed
Initiating ARP Ping Scan at 09:29
Scanning 255 hosts [1 port/host]
Completed ARP Ping Scan at 09:29, 1.91s elapsed (255 total
hosts)
Initiating Parallel DNS resolution of 255 hosts. at 09:29
Completed Parallel DNS resolution of 255 hosts. at 09:29,
0.03s elapsed
Nmap scan report for 192.168.2.0 [host down]
Nmap scan report for 192.168.2.2 [host down]
Nmap scan report for 192.168.2.3 [host down]
Nmap scan report for 192.168.2.4 [host down]
Nmap scan report for 192.168.2.5 [host down]
Nmap scan report for 192.168.2.6 [host down]
Nmap scan report for 192.168.2.7 [host down]
Nmap scan report for 192.168.2.8 [host down]
Nmap scan report for 192.168.2.9 [host down]
Nmap scan report for 192.168.2.10 [host down]
```

…

…

Wenn Sie, wie in diesem Beispiel, die Angabe 192.168.2.0/24 verwenden, werden alle 256 Hosts zwischen 192.168.2.0 und 192.168.10.255 der Reihe nach gescannt. Wie Sie anhand obiger Ausgabe entnehmen können, werden die Hosts der Reihe nach gepüft.

Wenn Sie den von den Entwicklern bereitgestellten Host *scanme.nmap.org*, der die IP-Adresse 45.33.32.156 besitzt, scannen und für den Scan-Vorgang *scanme.nmap.org/16* verwenden, werden alle IP-Adressen zwischen 45.33.0.0 und 45.33.255.255, also ingesamt 65.536 geprüft.

Für die Angabe eines bestimmten Netzsegments eignet sich die CIDR-Notation hervorragend. Aber spätestens dann, wenn Sie bestimmte Bereiche auslassen wollen, erweist sich diese Vorgangensweise als unpraktisch. Oftmals will man IP-Adressen, die mit 0 oder mit 255 enden, auslassen, weil sie als Unternetzwerk und Broadcast-Adressen verwendet werden.

In solchen Fällen können Sie die Verwendung einer sogenannten Oktett-Bereichsadressierung verwenden. Dabei geben Sie für den Bereich eine kommaseparierte Liste von Zahlen oder Bereichen für jedes Oktett an.

Ein Beispiel verdeutlicht die Vorgehensweise. Wenn Sie die Angabe 192.168.0-255.1-254 verwenden, werden alle Adressen im Bereich ausgelassen, die mit 0 oder 255 enden.

Die Konfiguration 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. Dabei werden beide Bereichsgrenzen weggelassen, die Standardwerte sind 0 für die linke und 255 für die rechte Grenze.

Anstellen von IPv4- können Sie natürlich auch IPv6-Adressen verwenden. Allerdings können die nur durch ihre vollständige IPv6-Adresse oder ihren Hostnamen angegeben werden. Leider können Sie CIDR und Oktettbereiche bei IPv6 nicht verwenden.

Nmap akzeptiert auch mehrere Host-Angaben, die einen unterschiedlichen Typ aufweisen. Sie können also beispielsweise folgende Angaben kombinieren:

```
reibold.de 192.168.2.0/8 10.0.0,1,3-7
```

Noch deutlich flexibler sind Sie, wenn Sie die Argumente nutzen, die Ihnen Nmap für die Zielauswahl zur Verfügung stellt. Sie können beispielsweise eine Liste mit Hostnamen, IP-Adressen und Adressbereichen verwenden. Das entsprechende Argument hierfür lautet:

```
-iL <dateinname>
```

In der Praxis kommt es häufig vor, dass man nicht nur einzelne Hosts oder Bereiche scannen möchte, sondern oftmals will man Duzende, Hunderte oder sogar Tausende Hosts prüfen. Die auf der Konsole anzugeben, ist nahezu unmöglich.

In diesem Fall verwenden Sie eine Liste, in der die Adressen hinterlegt sind. Wenn Sie die Systeme eines lokalen Netzwerks prüfen wollen, so können Sie die angeschlossenen Systeme schnell bestimmen, indem Sie die Liste der aktuellen Adresszuweisungen aus dem DHCP-Server exportieren.

In der Liste können Sie wieder alle Formate verwenden, die Sie auch auf der Konsole verwenden können:

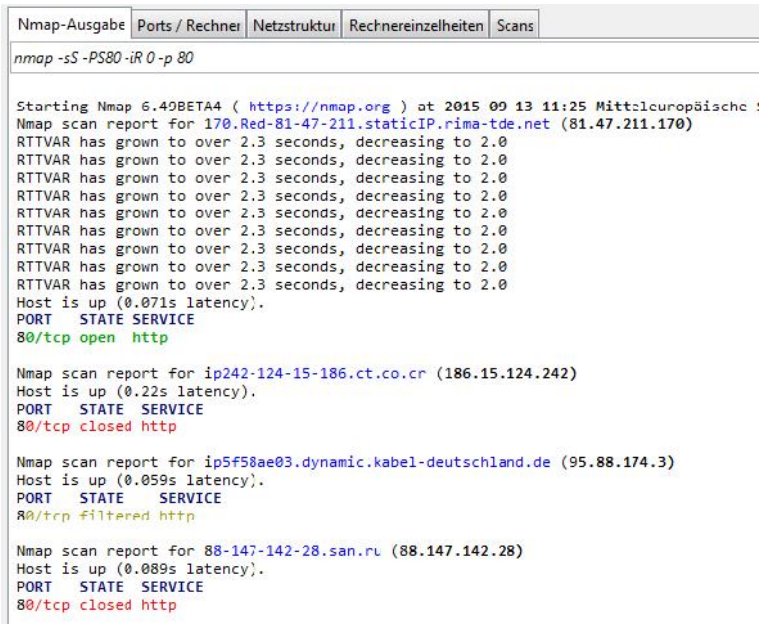
- IP-Adresse
- Hostname
- CIDR
- IPv6-Adresse
- Oktettbereich

Bleibt noch die Frage, wie diese Liste aussehen kann. Auch das ist einfach: Die Listeneinträge müssen durch ein oder mehrere Leerzeichen, Tabulatoren oder Zeilenumbrüche getrennt sein.

Nmap unterstützt auch einen Zufallsmechanismus, bei dem ohne genauere Vorgangen Hosts abgefragt werden. Das hierfür zuständige Argument:

```
-iR <Anzahl_Hosts>
```

Dabei lässt der Scanner automatisch bestimmte unerwünschte IPs aus, beispielsweise private, Multicast- oder unbesetzten Adressbereiche. Sie können dabei auch einen endlosen Scan durchführen, indem Sie das Argument 0 verwenden.



Das zufällige Ermitteln von Webservern.

Sie können mit Nmap mit dem folgenden Befehl beispielsweise per Zufallsmechanismus Webserver in Ihrer Nähe ermitteln:

```
nmap -sS -PS80 -iR 0 -p 80
```

Dazu sollten Sie allerdings Zeit mitbringen, denn der Scan-Vorgang dauert sehr lange.

Oben haben Sie eine Möglichkeit kennengelernt, wie man bestimmte Adressbereiche von einem Scan-Vorgang ausnehmen kann. Doch sonderlich flexibel ist dieses Ausschlussverfahren nicht. Deutlich einfacher ist es, wenn Sie stattdessen eine Liste der auszuschließenden Ziele anlegen:

```
--exclude <host1>,<host2>,..
```

Die Vorgehensweise ist wieder einfach: Sie legen eine kommaseparierte Liste der Einträge an, die Sie von dem Scan aufnehmen wollen. Diese Hosts werden auch dann von dem Scan-Vorgang ausgenommen, wenn Sie in den angegebenen Netzwerkbereich fallen. Die Ausschlussliste kann wieder Hostnamen, CIDR-Netzblöcke, Oktettbereiche etc. aufweisen.

Noch flexibler sind sie, wenn Sie die auszuschließende Datei in einer Ausschlussliste aufführen. Die entsprechende Konfiguration lautet dann wie folgt:

```
--excludefile <ausschlussdatei>
```

Bei der Verwendung einer Ausschlussdatei gibt es allerdings zum Ausschluss einzelner Host mit der Option *--exclude* einen Unterschied: Sie können die Ziele mit Zeilenumbrüchen, Leerzeichen oder Tabulatoren voneinandertrennen.

2.2 Host erkennen

Anhand obigen Beispiels, bei dem mit dem Befehl *nmap -T4 -A -v 192.168.2.1/24*, ein ganzes Subnetz gescannt wurde, zeigt sich eine Herausforderung für Nmap: Die Zielliste muss bei gelegentlich gigantisch großen IP-Bereichen auf eine Liste aktiver oder interessanter Hosts reduziert werden. In der Regel macht es wenig Sinn, bei allen IP-Adressen alle Ports zu scannen. Viel wichtiger wäre es, wichtige von weniger Systemen zu unterscheiden. Doch welches sind die relevanten und welches sind die irrelevanten Systeme?

Die Eigenschaften, die ein Netzwerksystem zu einem wichtigen machen, sind in erster Linie von dem Einsatzbereich und der Aufgabe abhängig. Manchmal ist es ein bestimmter Dienst, dann eine bestimmte Applikation, und wieder ein anderes Mal ist nur wichtig, ob der Druckerserver verfügbar ist.

Bei internen Checks genügt oftmals schon ein ICMP-Ping, um die Erreichbarkeit zu prüfen, ein anderes Mal will man Systeme einem ausgiebigen Penetration Test unterziehen.

So bleibt es zunächst jedem Administrator bzw. Sicherheitsbeauftragten überlassen, welches die wirklich wichtigen Dinge sind. Die Anforderungen an die Host-Erkennungen sind so verschieden wie die Netzwerke. Aus diesem Grund bietet Ihnen Nmap verschiedensten Funktionen zur Ermittlung von für Sie relevante Systeme.

2.2.1 List-Scan

Die Funktionen eines klassischen Ping-Scans gehen weit über die einfachen ICMP Echo-Request-Pakete hinaus, die die meisten Anwender mit diesem Befehl verbinden. Nmap bietet Ihnen dennoch die Möglichkeit, einen List-Scan (*-sL*) durchzuführen oder Ping auszuschalten (*-PN*). Sie können außerdem verschiedene Tests kombinieren. All diese Prüfungen dienen dazu herauszufinden, ob ein Host oder ein Dienst aktiv ist oder nicht. Gerade in lokalen Netzwerken, die einen privaten Adressraum verwenden, ist oftmals nur ein kleiner Bruchteil der verfügbaren IP-Adressen aktiv.

Wenn Sie bei der Ausführung von Nmap kein Argument angeben, übermittelt Nmap ein TCP-ACK-Paket an Port 80 und ein ICMP Echo-Request an alle Zielrechner. Einzige Ausnahme: Bei lokalen Zielen wird ein ARP-Scan durchgeführt. Diese Standardscaneinstellungen entsprechend der Optionen *-PA -PE*. Meist ist das mehr als genug für eine erste Host-Erkennung. Wenn Sie allerdings sicherheitrelevante Informationen der Hosts abfragen wollen, benötigen Sie weit mehr Informationen. Sie können dabei insbesondere die Ping-Abfragen miteinander kombinieren. Das erhöht auch die Chancen, strikte Firewall-Konfigurationen zu überwinden.

Nmap führt standardmäßig zunächst eine Host-Erkennung und dann einen Port-Scan auf jedem Host aus, der als aktiv identifiziert wird. Für die Host-Erkennung stehen Ihnen verschiedene Optionen zur Verfügung.

Oben sind wir bereits dem Begriff List-Scan begegnet. Dabei handelt es sich um eine abgespeckte Form der Host-Erkennung. Das Scan-Ergebnis listet lediglich jeden gefundenen Host auf. Nmap führt außerdem noch einen Reverse-DNS-Auflösung der Hosts durch, um deren Namen aufzulösen. Doch alleine diese Informationen genügen oftmals schon, um die Funktion einer Netzwerkkomponente zu erkennen. Eine Hostname *cms.** deutet auf ein Content-Managementsystem hin, eine Host mit der Bezeichnung *fw.** auf eine Firewall. Einen List-Scan führen Sie wie folgt durch:

```
-sL
```

Der List-Scan gilt als eine gute Plausibilitätsprüfung, um sicherzustellen, dass die IP-Adressinformatien der Ziele verwertbar sind. Sollten Sie in der Ausgabe in den Host-Bezeichnungen auf Domainnamen stosen, die Ihnen nichts sagen, sollten Sie diese Systeme einer genaueren Untersuchung unterziehen.

Bei der Verwendung des List-Scans sollten Sie beachten, dass dieser Scan-Typ nicht mit höheren Funktionalitäten wie beispielsweise dem Port-Scanning oder der Betriebssystemerkennung kombiniert werden können.

2.2.2 Ping-Scan

Wenn Sie lediglich eine Host-Erkennung durchführen wollen, bei der alle verfügbaren Hosts ausgegeben werden, die auf den Scan-Vorgang reagieren, führen Sie einen sogenannten Ping-Scan durch:

```
-sP
```

Bei diesem Scan-Typ führt Nmap keine weiteren Tests durch. Der Vorteil dieses Tests: Er führt flott eine schwache Aufklärung des Zielnetzwerks durch, ohne viel Aufmerksamkeit zu erregen.

Sowohl Angreifer als auch Netzwerkadministratoren profitieren von dieser Information. Sie benötigen nicht immer eine detaillierte Liste mit den IP-Adressen und weiteren Informationen. Vielmehr kann man mit dieser Option umkompliziert die Anzahl der verfügbaren Rechner in einem Netzwerk ermitteln. Man bezeichnet diesen Check gelegentlich auch als Ping-Sweep. Der Vorteil: Er ist wesentlich zuverlässiger als das Abfragen der Broadcast-Adressen mit Ping.

Bei einem Ping-Scan mit der Option *-sP* werden ein ICMP Echo-Request und ein TCP-ACK-Paket an Port 80 gesendet. Sie können den Ping-Scan mit allen Erkennungsmethoden (außer *-PN*) kombinieren. Das ist insbesondere sinnvoll und manchmal notwendig, wenn sich zwischen dem Nmap-Host und dem Zielnetzwerk eine Firewalls befindet.

Nmap bietet Ihnen auch die Möglichkeit, die Ping-Erkennung abzuschalten. Dazu verwenden Sie folgende Option:

```
-PN
```

Das Pingen dient Nmap standardmäßig dazu, die aktiven Rechner zu ermitteltn, die dann im nächsten Schritt einer genaueren Analyse unterzogen werden. Bei aktiven Hosts werden beispielsweise Port-Scans, Versions- oder Betriebssystemerkennung

durchgeführt. Wenn Sie die Host-Erkennung mit der Option *-PN* ausschalten, werden die Scan-Funktionen auf allen angegebenen Ziel-IP-Adresssen durchgeführt.

Nmap kann auch ein leeres TCP-Paket mit gesetztem SYN-Flag an die Hosts übermittelt. Der Standardport ist dabei 80, kann aber auch geändert werden. Sie können dabei einzelne Ports und ganze Listen angeben. Wichtig ist, dass Sie keine Leerzeichen zwischen der Option und der Portliste verwenden:

```
-PS<port-liste>
```

Das SYN-Flag zeigt dem Ziel-Host an, dass Sie eine Verbindung mit diesem aufbauen wollen. Standardmäßig wird dabei der Ziel-Port geschlossen und ein RST-(Reset-)Paket an Nmap zurückgeschickt. Ist der Port allerdings geöffnet, so führt der Ziel-Host in zweiten Schritt einen Three-Way-Handshake durch. Dazu antwortet er mit einem SYN/ACK-TCP-Paket.

Nmap bricht daraufhin die Verbindung ab und antwortet mit RST (anstelle eines ACK-Pakets), um den Handshake zu komplettieren. Hier zwei konkrete Beispiele:

```
-PS501
```

```
-PS80,100-120,1100,24000
```

Für Nmap macht es keinen Unterschied, ob ein Host mit RST oder SYN/ACK antwortet. Für den Scanner ist nur wichtig, dass der Host verfügbar ist. Das wird entsprechend vermerkt.

2.2.3 TCP-ACK-Ping

Mit TCP-ACK-Ping stellt Ihnen Nmap einen weiteren Test zur Verfügung, der dem zuvor beschriebenen SYN-Ping sehr ähnlich ist. Der Unterschied besteht darin, dass das TCP-ACK-Flag anstelle von SYN-Flag gesetzt wird. Ein ACK-Paket erweckt auf Seiten des Ziels den Eindruck, er soll Daten auf einer bestehenden TCP-Vebindung bestätigen. Da eine solche Verbindung aber nicht existiert, gibt der Remote-Host eine RST-Paket zurück und verrät somit seine Existenz.

```
-PA <port-liste>
```

Auch dieser Test verwendet standardmäßig den Port 80.

Sie fragen sich nun womöglich, warum es zwei Tests mit sehr ähnlicher Funktionalität gibt. Die Antwort darauf ist recht einfach: Da Ihnen zwei Testverfahren zur Verfügung stehen, erhöhen sich die Chancen für die Umgehung einer Firewall.

Viele Router und Firewalls sind so konfiguriert, dass sie eingehende SYN-Pakete blockieren. Wenn wie bei iptables die Option *--syn* verwendet wird, um einen zustandslosen Ansatz zu implementieren, werden SYN-Ping-Tests (*-PS*) mit hoher Wahrscheinlichkeit gefiltert. In diesem Fall liefert allerdings der ACK-Test das gewünschte Ergebnis.

Hier ein Beispiel für einen ACK-Test:

```
nmap -sP -PA server.de


Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-22
10:57 Mitteleuropäische Sommerzeit

Note: Host seems down. If it is really up, but blocking our
ping probes, try -Pn

Nmap done: 1 IP address (0 hosts up) scanned in 4.45 seconds
```

Professionelle Firewalls sind außerdem in der Lage, mit sogenannten zustandsbehafteten Regeln unerwartete Pakete zu blockieren. Auch iptables beherrscht das mit der Option *--state*. Bei solchen Konstellationen ist ein SYN-Test in der Regel erfolgreicher, da die Firewall unerwartete ACK-Pakete identifiziert und blockiert. Die Lösung ist indes einfach: Senden Sie PS und PA SYN- und ACK-Testpakete.

2.2.4 UDP-Ping

Nmap bietet eine weitere Möglichkeit zur Host-Ermittlung: den UDP-Ping. Dabei wird ein leeres UDP-Paket an den Ziel-Host übermittelt. Wenn Sie keinen Port oder keine Port-Liste angeben, verwendet Nmap die Standardkonfiguration 31338. Um einen UDP-Ping auszuführen, verwenden Sie folgende Option:

```
-PU <port-liste>
```

Wenn Sie nun einen Host mit einem UDP-Test konfrontieren, sollte dieser ein ICMP-Paket zurückgeben, dass der gewünschte Port erreichbar ist. Für Nmap ist der Rechner daher verfügbar. ICMP-Fehler und ausbleibende Antworten interpretiert Nmap als nicht erreichbarer oder nicht verfügbarer Host.

Der entscheidende Vorteil dieses Tests ist der Umstand, dass man damit Firewalls und Filter umgeht, die nur TCP überprüfen. Da sich die meisten auf das TCP-Protokoll konzentrieren, stehen die Chancen also gut, dass Sie von den Ziel-Hosts die gewünschten Informationen abgreifen können.

2.2.5 ICMP-Ping-Arten

Nmap unterstützt neben den oben genannnten Erkennungsarten auch verschiedene sogenannte ICMP-Ping-Arten. Dabei versendet der Scanner ein ICMP Typ-8-Paket (Echo-Request) an die Zieladressen und erwartet eine Typ-0-Antwort (Echo-Reply) vom verfügbaren Host. Die zugehörigen Optionen:

```
-PE; -PP; -PM
```

Leider werden auch diese ICMP-Scans häufig von Hosts und Routern gefiltert. Daher sind die Ergebnisse nur bedingt verwertbar. Das gilt insbesondere beim Scannen von Internet-Hosts. Anders ist das, wenn Sie Ihr internes Netzwerk unter die Lupe nehmen. Dann sollten Sie insbesondere die Option *-PE* verwenden, um die Echo-Requests einzuschalten.

2.2.6 IP-Protokoll-Ping

Nmap bietet eine weitere Möglichkeit der Host-Erkennung: das IP-Protokoll-Ping. Bei diesem Test sendet Nmap IP-Pakete an den Host in deren IP-Header die angegebene Protokollnummer hinterlegt ist. Das Format sieht wie folgt aus:

```
-PO <protokoll-liste>
```

Wenn Sie kein Protokoll spezifizieren, werden mehrere IP-Pakete für ICMP (Protokoll 1), IGMP (Protokoll 2) und IP-in-IP (Protokoll 4) gesendet.

Das Besondere an dieser Art der Host-Erkennung: Sie sucht nach Antworten, die entweder dasselbe Protokoll wie der Test besitzen, oder nach Meldungen, dass das ICMP-Protokoll nicht erreichbar ist. Letztes weißt darauf hin, dass das gegebene Protokoll vom Zielhost nicht unterstützt wird.

2.2.7 ARP-Ping

Nmap wird überwiegend für das Scannen von Ethernet-LANs verwendeet. Lokale Netzwerke verwenden üblicherweise einen privaten Adressbereich, bei dem ein Großteil der IP-Adressen meistens nicht verwendet wird. Sendet Nmap ein rohes IP-Paket (beispielsweise einen ICMP Echo-Request) muss die Hardware-Zieladresse (ARP) bestimmt werden. Nur so ist eine korrekte Adressierung des Ethernet-Frames möglich.

Mit dem ARP-Scan steht Ihnen ein optimierter Test für ARP-Anfragen zur Verfügung. Erhält der Scanner auf einen solchen Request eine Reaktion des Ziel-Hosts, muss Nmap nicht erst auf eine Antwort der IP-basierten Ping-Pakete warten. Der Vorteil: Ein ARP-Scan ist wesentlch schneller und zuverlässiger als IP-basierte Scans.

Die zugehörige Option lautet wie folgt:

```
-PR
```

Beim Scannen von lokalen Netzwerken wird diese Option standardmäßig verwendet.

2.2.8 Traceroute

Traceroute ist ein Klassiker unter den Neztwerktests. Damit können Sie mit den Scan-Ergebnissen nach einem Scan den wahrscheinlichsten Port und das wahrscheinlichste Protokoll identifizieren, die auf dem Ziel ausgführt werden. Diese Option können Sie mit fast allen Scan-Arten außer Connect-Scans (*-sT*) und Idle-Scans (*-sI*) verwenden. Die zugehörige Option:

```
--traceroute
```

2.2.9 DNS-Auflösung

Schließlich stellt Ihnen Nmap noch einige weitere verschiedene Optionen für die DNS-Auflösung zur Verfügung. Die können beispielsweise die Reverse-DNS-Auflösung bei den gefundenen aktiven IP-Adressen unterbinden. Durch das Deaktiveren können Sie die Scan-Zeiten dramatisch reduzieren:

```
-n
```

Um die DNS-Auflösung bei allen Ziel-IP-Adressen durchzuführen, verwenden Sie folgende Option:

```
-R
```

Meist führt man ein Reverse-DNS nur bei antwortenden Hosts durch, die online sind.

Sie können auch die DNS-Auflösung des verwendeten Betriebssystems nutzen. Der Scanner löst standardmäßig alle IP-Adressen auf, indem er Anfragen direkt an die auf Ihrem Host konfigurierten Nameserver schickt und dann auf Antworten wartet. Um die Performance zu erhöhen, werden meist mehrere Dutzend Anfragen parallel ausgeführt.

Allerdings erweist sich auch das häufig als Flaschenhals. Mit der folgenden Option verwenden Sie die Auflösungsmethode Ihres Systems:

```
--system-dns
```

Beachten Sie, dass bei IPv6-Scans immer die Auflösungsmethode des jeweiligen Betriebssystems verwendet wird.

2.3 Port-Scanning in der Praxis

Nmap ist ein Portscanner, der Ihnen verschiedenste Informationen über Ports liefert. Doch was hat es genau mit diesen Ports auf sich? Und wie können potenziellen Angreifer Informationen über diese nutzen? Ein Port kann man sich bildlich wie eine Nebenstellennummer einer Telefonanlage vorstellen, unter der ein bestimmter Dienst oder eine bestimmte Anwendung verfügbar ist.

Prinzipiell kennt man drei Kategorien von Portnummern. Die sogenannten Well known ports (0-1023) werden von der IANA (Internet Assigned Numbers Authority) vergeben. Diese Ports werden von bestimmten System-Prozessen und Anwendungen genutzt, beispielsweise von FTP, HTTP, IMAP, POP3, SMTP und Telnet. Dabei nutzen TCP und UDP (meist) die gleichen Portnummern. IANA hat außerdem die sogenannten Registered Ports (1024-49151) eingeführt.

Hier einige Beispiele für Well known Ports:

- 20 – FTP-Datentransfer vom Server zum Client
- 21 – FTP-Steuerbefehle durch den Client
- 23 – Telnet-Kommunikation
- 25 – SMTP-Mail-Versand
- 53 – DNS-Auflösung von Domainnamen in IP-Adressen
- 80 – HTTP-Webserver
- 110 – POP3-Client-Zugriff auf E-Mail-Server
- 143 – IMAP
- 194 – IRC
- 389 – LDAP
- 443 – HTTPS
- 666 – DOOM-Online-Spiel
- 901 – SWAT
- 989 – FTPS-Daten
- 990 – FTPS-Steuerbefehle

Software-Hersteller verwenden registrierte Ports für Programme, die ein Anwender installiert. Ein bekannter registrierter Port ist beim SIP (Session Initiation Protocol) der Port 5060. MySQL-Datenbanken verwenden meist 3306. Außerdem gibt es Alternativen zum Standard-Port, beispielsweise 8008 und 8080 für den Standard-Webserver-Port 80.

Sie können die aktuelle Belegung eines Computers einfach einsehen. Auf einem Linux-Rechner finden Sie die Portliste unter */etc/services*, bei einem 32Bit-Windows-System unter *%WINDIR%\system32drivers\etc\services*.

Es gibt eine weitere Klasse: die Dynamic oder Private Ports. Diese Ports sind frei verwendbar und keinem Service zugeordnet. Die vollständige Liste der Port-Nummern finden Sie bei der IANA unter folgender URL:

```
http://www.iana.org/assignments/service-names-port-
numbers/service-names-port-numbers.xml
```

Um ein System sicher zu machen, sollte man alle jene Ports schließen, die man nicht benötigt. Doch dazu müssen Sie zunächst wissen, welche Ports offen und welche geschlossen sind. Genau dabei unterstützt Sie Nmap. Offene Ports sind immer eine Art Eingang zu einem System. Und wenn die dahinter befindlichen Dienste Sicherheitslücken aufweisen, so sind die Dienste recht einfach angreifbar. Nmap verfügt über die Datei *nmap-services*, in der die bekannten Port-Belegungen hinterlegt sind.

Beim Port-Scanning wird das Zielsystem aus der Ferne getestet, um herauszufinden, welche Ports verfügbar sind. Dabei wird insbesondere das Testergebnis mit einem Status verknüpft. Wenn Sie den einfachsten Nmap-Befehl *nmap <zielhost>* ausführen, scannt das Programm die 1000 häufigsten verwendeten TCP-Ports auf dem Host. Dabei wird jedem Host einer der folgenden Zustände zugewiesen:

- offen
- geschlossen
- gefiltert
- ungefiltert
- offen|gefiltert
- geschlossen|gefiltert

Beachten Sie, dass es sich bei diesen Zuständen nicht um Port-Eigenschaften handelt, sondern vielmehr beschreiben Sie, wie Nmap einen Port betrachtet. Wenn Sie im lokalen Netzwerk einen Scan ausführen, so kann ein Port offen sein, während er sich für die Internet-Kommunikation gefiltert darstellt. Welches sind nun die spezifischen Eigenschaften und Besonderheiten der verschiedenen Port-Zustände?

Wird ein Port als *offen* klassifiziert, so wird auf dem betreffenden System ein Programm oder Service ausgeführt, das bzw. der bereit ist, TCP-Verbindungen oder UDP-Pakete auf diesem Port anzunehmen. Beim Port-Scanning geht es meist darum, die Ports zu identifizieren, die einen offenen Status aufweisen, denn sie können von Hackern komprimiert werden. Aber ein Port-Scanning ist nicht immer nur interessant, um mögliche Angriffspunkte zu identifizieren, sondern auch, um Dienste zu finden, die in einem Netzwerk angeboten und genutzt werden können.

Nmap bewertet einen Port als geschlossen, wenn er zwar erreichbar ist (was so viel bedeutet, dass er Nmap-Testpakete empfängt und darauf antwortet), aber, dass dort kein Service verfügbar ist, das den Port abhört. Das Scan-Ergebnis kann nützlich sein, um festzustellen, ob ein Host online ist oder welche IP-Adresse er besitzt. Auch für die Betriebssystemerkennung ist dieser Status relevant.

Und, dass ein Port bei einem ersten Scan-Vorgang geschlossen ist, bedeutet ja nicht zwangsläufig, dass er immer diesen Zustand aufweist. Außerdem werden Ports mit diesem Zustand auch immer mal wieder durch eine Firewall geschützt und erhalten dann den Zustand gefiltert.

Ein weiterer wichtiger Zustand ist *gefiltert*. Der kommt dann zur Verwendung, wenn Nmap nicht feststellen kann, ob ein Port offen ist, weil dieser durch eine Paketfilterung blockiert wird. Filter bzw. Blockaden werden meist durch Firewalls oder Router-Regeln realisiert.

Gelegentlich begegnen Sie auch dem Zustand *ungefiltert*. Der wird dann einem Port zugewiesen, wenn Nmap nicht herausfinden kann, ob er offen oder geschlossen ist. Diesem Zustand begegnen Sie nur bei einem ACK-Scan. Den können Sie verwenden, um Firewall-Regelwerke zu bestimmen. Wenn Sie einem ungefilterten begegnen können Sie oftmals mit Scan-Methoden wie einem Window-, SYN- oder FIN-Scan herausfinden, ob der Port offen ist.

Zwei weitere Port-Zustände kennt Nmap noch. Der Zustand *offen|gefiltert* zeigt an, dass Nmap nicht herausfinden konnte, ob ein Port offen oder gefiltert ist. Dieses Ergebnis erhalten Sie beispielsweise dann, wenn offene Ports keine Antwort geben. Das Fehlen einer Antwort wird von Nmap oftmals so interpretiert, dass ein Paketfilter das Testpaket verworfen hat oder, dass keine Antwort provoziert werden konnte. Nmap ist sich schlicht nicht sicher, ob der Port offen ist oder gefiltert wird. Insbesondere UDP-, IP-Protokoll-, FIN-, NULL- und Xmas-Scans liefern dieses Ergebnis. Auf diese Scans kommen wird später noch zu sprechen.

Dem letzten Scan-Status begegnen Sie nur bei der Durchführung von IP-ID-Idle-Scans: *geschlossen|gefiltert*. Der wird dann ausgegeben, wenn Nmap nicht feststellen kann, ob ein Port geschlossen ist oder gefiltert wird.

2.4 Scan-Tutorial

Nachdem wir in den vergangenen Abschnitten wichtige Techniken und Scan-Methoden kennengelernt haben, schauen wir uns anhand eines typischen Beispiels an, wie man in der Praxis beim Port-Scannen mit Nmap vorgehen kann. Zunächst benötigen Sie ein Ziel. Hierfür können Sie einen lokalen Rechner oder eben den vom Nmap-Team bereitgestellten Server *scanme.nmap.org* verwenden.

Was passiert nun bei einem simplen Scan-Vorgang konkret? Wie Sie bereits wissen, sieht ein einfacher Scan wie folgt aus: *nmap <ziel-host>*. Wenn Sie diesen Befehl ausführen und dabei einen Hostnamen angeben, konvertiert Nmap diesen mit Hilfe des DNS zunächst in eine IPv4-Adresse. Liegt die IP-Adresse bereits im IPv4-Format vor, entfällt die Konvertierung.

Als Nächstes führt Nmap eine Ping-Abfrage aus. Dabei werden standardmäßig ein ICMP Echo Request und ein TCP ACK-Paket an Port 80 gesendet. So prüft Nmap, ob der Host einreichbar ist. Läuft der Host nicht, gibt Nmap eine entsprechende Meldung aus und beendet sich.

Dann konvertiert Nmap die Ziel-IP-Adresse wieder zurück in den Hostnamen. Dabei wird eine DNS-Reverse-Abfrage ausgeführt. Diese Abfrage können Sie auch mit der Option *-n* umgehen.

Es folgt ein TCP-Port-Scan nach den 1000 populärsten Ports, die in der Datei *nmap-services* aufgeführt sind. Anschließend für Nmap den Scan-Vorgang aus und gibt das Scan-Ergebnis auf der Konsole aus. Hier eine Beispielausgabe des Scans eines lokalen Routers:

```
# nmap 192.168.2.1


Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-23
17:22 Mitteleuropäische Sommerzeit

Nmap scan report for Speedport.ip (192.168.2.1)

Host is up (0.000095s latency).

Not shown: 998 filtered ports

PORT   STATE SERVICE

53/tcp open  domain

80/tcp open  http

MAC Address: A4:99:47:1B:FF:8B (Huawei Technologies Co.)


Nmap done: 1 IP address (1 host up) scanned in 6.24 seconds
```

Wenn Sie den von dem Nmap-Team bereitgestellten Server einem Standard-Scan unterziehen wollen, präsentiert Ihnen Nmap in etwa folgende Ausgabe:

```
# nmap scanme.nmap.org


Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-23
17:23 Mitteleuropäische Sommerzeit

Nmap scan report for scanme.nmap.org (45.33.32.156)

Host is up (0.19s latency).
```

```
Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 8.09 seconds
```

Schauen wir uns die zweite Ausgabe näher an. Der ersten Zeile können Sie entnehmen, dass Nmap gestartet wurde. Der ersten Zeile können Sie auch den Ausführungszeitpunkt und den Link zur Projekt-Website entnehmen.

In der zweiten Zeile werden der Hostname des Ziels und die IPv4-Adresse angezeigt. Nmap verrät Ihnen als Nächstes, ob der Host verfügbar ist oder nicht. Kann Nmap auch die MAC-Adresse einlesen, wird diese ebenfalls ausgeben.

Nmap versucht sich standardmäßig an der Ermittlung der vermeintlich wichtigsten Ports. In diesem Fall identifiziert Nmap 996 Ports als geschlossen. Die eigentlich interessanten Informationen können Sie der nachfolgenden tabellarischen Übersicht entnehmen: Die offenen Ports, deren Status und die Dienste, die auf den offenen Ports verfügbar sind. Abschließend verrät Ihnen Nmap noch, wie viele IP-Adressen das Programm gescannt und wie lange dieser Vorgang gedauert hat.

Es versteht sich von selbst, dass Sie auf diesem Weg meist viele Informationen über die Ziel-Systeme erhalten – in der Regel viel zu viele Informationen. Also muss man einen Weg einschlagen, der Sie möglichst zielführend und punktgenau an Ihr Ziel bringt.

Wir verfeinern dazu den einfachen Scan mit vier zusätzlichen Optionen. Dazu verwenden wir die Option *-p0-*, um alle möglichen TCP-Ports zu prüfen. Damit Nmap möglichst viele Informationen ausgibt, aktivieren Sie den geschwätzigen Modus mit der Option *-v*. Um weitere Informationen über das Zielsystem wie das Betriebssystem, die Services und deren Version zu erfahren, verwenden Sie zusätzlich den Schalter *-A*. Wie Sie obiger Ausgabe entnehmen können, hat ein simpler Scan-Vorgang bereits mehrere Sekunden gedauert. Um den Scan-Vorgang zu beschleunigen, verwenden wir zusätzlich die Option *-T4*. Die führt zu einem deutlichen Performance-Gewinn. Und so lautet der vollständige Befehl:

```
nmap –p0- -v –A –T4 scanme.nmap.org
```

Wenn Sie diesen nun ausführen, produziert Nmap folgende Ausgabe:

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-23
20:03 Mitteleuropäische Sommerzeit

Initiating Ping Scan at 20:03

Scanning scanme.nmap.org (45.33.32.156) [4 ports]

Completed Ping Scan at 20:03, 0.77s elapsed (1 total hosts)

Initiating Parallel DNS resolution of 1 host. at 20:03

Completed Parallel DNS resolution of 1 host. at 20:03, 0.00s
elapsed

Initiating SYN Stealth Scan at 20:03

Scanning scanme.nmap.org (45.33.32.156) [65536 ports]

Discovered open port 80/tcp on 45.33.32.156

Discovered open port 22/tcp on 45.33.32.156

SYN Stealth Scan Timing: About 4.42% done; ETC: 20:15
(0:11:11 remaining)

SYN Stealth Scan Timing: About 9.46% done; ETC: 20:14
(0:09:44 remaining)

SYN Stealth Scan Timing: About 21.30% done; ETC: 20:15
(0:09:07 remaining)

SYN Stealth Scan Timing: About 28.24% done; ETC: 20:15
(0:08:26 remaining)

SYN Stealth Scan Timing: About 37.01% done; ETC: 20:16
(0:07:46 remaining)

SYN Stealth Scan Timing: About 42.32% done; ETC: 20:16
(0:07:07 remaining)

SYN Stealth Scan Timing: About 48.78% done; ETC: 20:16
(0:06:28 remaining)

SYN Stealth Scan Timing: About 55.01% done; ETC: 20:16
(0:05:49 remaining)

SYN Stealth Scan Timing: About 60.46% done; ETC: 20:16
(0:05:09 remaining)

SYN Stealth Scan Timing: About 66.27% done; ETC: 20:17
(0:04:28 remaining)
```

```
SYN Stealth Scan Timing: About 71.62% done; ETC: 20:16
(0:03:40 remaining)

Discovered open port 9929/tcp on 45.33.32.156

Discovered open port 31337/tcp on 45.33.32.156

SYN Stealth Scan Timing: About 76.95% done; ETC: 20:16
(0:03:00 remaining)

SYN Stealth Scan Timing: About 82.02% done; ETC: 20:16
(0:02:20 remaining)

SYN Stealth Scan Timing: About 87.07% done; ETC: 20:16
(0:01:40 remaining)

SYN Stealth Scan Timing: About 92.27% done; ETC: 20:16
(0:01:00 remaining)

Completed SYN Stealth Scan at 20:17, 794.48s elapsed (65536
total ports)

Nmap scan report for scanme.nmap.org (45.33.32.156)

Host is up (0.19s latency).

Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f

Not shown: 65532 closed ports

PORT      STATE SERVICE

22/tcp    open  ssh

80/tcp    open  http

9929/tcp  open  nping-echo

31337/tcp open  Elite


Read data files from: C:\Program Files\Nmap

Nmap done: 1 IP address (1 host up) scanned in 801.79 seconds

           Raw packets sent: 66369 (2.920MB) | Rcvd: 77523
(3.101MB)
```

Wie Sie anhand der voranstehenden Ausgabe schnell erkennen können, fällt diese deutlich umfangreicher aus. Dennoch ist auch sie recht einfach zu verstehen. Die wichtigsten Informationen können Sie wieder der tabellarischen Übersicht der offenen Ports entnehmen.

2.5 Port-Scan-Techniken

Nmap bietet Ihnen derart viele Optionen, Schalter und Steuermöglichkeiten, dass es nicht immer einfach ist, das optimale Werkzeug zu wählen. Da geht es Ihnen ähnlich, wie einem Handwerker, der mit einem vollen Werkzeugkasten auf der Baustelle anrückt. Je besser er ausgebildet ist, je routinierter und erfahrener er ist, umso einfacher fällt die Wahl – meistens intuitiv. Ähnlich ist das beim Penetration Testung und beim Port Scanning. Nmap unterstützt verschiedenste Scan-Techniken, die Sie natürlich auch miteinander kombinieren können.

Bevor Sie allerdings diese Scan-Techniken einsetzen können, sind einige Besonderheiten zu beachten. Die meisten Techniken können nur von privilegierten Benutzern verwendet werden. Der Grund: Sie versenden und empfangen rohe IP-Datenpakete. Und genau dafür sind auf Unix-basierten Systemen Root-Rechte erforderlich.

Nicht ganz so strikt sind Windows-Systeme. Aber auch hier empfiehlt sich die Verwendung eines administrativen Accounts. Sie sollten wo immer es möglich ist, Nmap mit einem Admin-Account ausführen, da die privilegierten Optionen den Scanner wesentlich mächtiger und flexibler machen.

Ein weiteres Problem erschwert das Port-Scanning: Viele Hosts reagieren auf Tests nicht RFC-konform und geben nicht die gewünschten bzw. erwarteten Meldungen zurück. Gerade FIN-, NULL- und Xmas-Scans sind hierfür besonders anfällig. Das erschwert die Arbeit eines Port-Scanners zusätzlich.

Nachfolgend lernen Sie die Port-Scan-Techniken kennen, die Nmap zu bieten hat. Ganz besonders wichtig ist dabei, dass Sie diese Techniken nicht miteinander kombinieren können. Eine Ausnahme gibt es dennoch: Den UDP-Scan können Sie mit allen anderen TCP-Scan-Methoden kombinieren. Scan-Techniken beginnen immer mit *-s* und werden dann um einen weiteren Buchstaben wie zum Beispiel ein *S*, ein *T* oder ein *A* ergänzt.

Nmap führt standardmäßig einen SYN-Scan (*-sS*) aus. Besitzt der Benutzer allerdings nicht die notwendigen Rechte für den Versand von rohen Paketen, switcht er zu einem sogenannten Connect-Scan (*-sT*). Schauen wir uns konkret an, was diese Scan-Techniken leisten können.

2.5.1 TCP-SYN-Scan

Nmap verwendet standardmäßig einen SYN-Scan, da dieser ausgesprochen schnell die Ziel-Hosts unter die Lupe nimmt. Laut Angaben des Entwicklerteams können dabei Tausende Ports pro Sekunde analysiert werden, sofern das Scannen nicht von einer intrusiven Firewall erschwert wird.

Diese Scan-Variante ist zudem recht unauffällig, weil die TCP-Verbindungen niemals geschlossen werden. Ein weiterer Vorteil dieses Scans: Er liefert eindeutige und belastbare Ergebnisse über den Port-Zustand. Sie wissen also nach einem Check sehr genau, ob ein Port offen, geschlossen und gefiltert ist. Der SYN-Scan verlangt folgende Option:

```
-sS
```

Hier ein Beispiel für die Ausführung:

```
nmap -sS scanme.nmap.org
```

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-23
21:11 Mitteleuropäische Sommerzeit
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.24s latency).
Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 4.26 seconds
```

Der SYN-Scan wird gelegentlich auch als halboffenes Scannen bezeichnet, da hierbei keine vollständige TCP-Verbindung hergestellt wird. Nmap sendet vielmehr ein SYN-Paket an das Ziel, so, als ob eine echte Verbindung hergestellt wer-

den soll, und wartet auf die Reaktion des Gegenübers. Dabei zeigt die Reaktion SYN/ACK an, dass ein Port lauscht, er also offen ist, und ein RST (Reset) zeigt an, dass niemand lauscht. Kann Nmap kein eindeutiges Ergebnis bei mehreren Tests ausmachen, wird der Port als gefiltert markiert.

2.5.2 TCP-Connect-Scan

Kann Nmap keinen SYN-Scan durchführen, so greift der Scanner standardmäßig zum TCP-Connect-Scan. Das ist beispielsweise dann der Fall, wenn der Anwender nicht über die notwendigen Rechte zum Versand von rohen Datenpaketen verfügt. Auch beim Scannen eines IPv6-Netzwerks verwendet man diese Scan-Technik:

```
-sT
```

Anstelle von rohen Paketen verwendet dieser Scan-Typ den Systemaufruf *connect*. Dabei handelt es sich um den identischen Systemaufruf auf höherer Ebene, den Webbrowser, P2P-Clients und viele andere netzwerkfähige Anwendungen zum Aufbau einer Verbindung verwenden.

Dennoch gilt: Wo immer die Verwendung eines SYN-Scans möglich ist, sollten Sie diesen verwenden. Auch deshalb, weil die meisten Intrusion Detection-Systeme beide Scan-Typen registrieren.

2.5.3 UDP-Scan

Die meisten Internet-Dienste nutzen das TCP-Protokoll, aber auch UDP-Dienste sind weit verbreitet, allen voran DNS, SNMP und DHCP (mit den Ports 53, 161/162 und 67/68). UDP-Scans sind im Allgemeinen deutlich langsamer und schwieriger durchzuführen. Daher werden Sie beim Penetration Testing häufig einfach weggelassen. Doch das ist ein grober Fehler, denn löchrige UDP-Dienste sind keine Seltenheit und können von Hackern gezielt angegriffen werden.

Doch auch hier ist Ihnen Nmap eine Hilfe, denn das Programm stellt Ihnen eine Inventarfunktion für UDP-Ports zur Verfügung:

```
-sU
```

Sie können den UDP- auch mit einem TCP-Scan (siehe oben, -sS) kombinieren. So schlagen Sie zwei Fliegen mit einer Klappe.

Bei diesem Test sendet Nmap einen leeren UDP-Header ohne Daten an alle Ziel-Ports. Senden die Ziele einen ICMP Port-unreachable-Fehler (Typ 3, Code 3) zurück, handelt es sich um einen geschlossenen Port. Alle anderen ICMP Unreachable-Fehler (Typ 3, Codes 1, 2, 9, 10 oder 13) werden als *gefiltert* eingestuft.

Antwortet ein Dienst mit einem UDP-Paket, deutet das auf einen offenen Port hin. Erhält Nmap bei weiteren Prüfungen keine Antwort, wird der Port als *offen|gefiltert* klassifiziert. Konkret bedeutet das, dass der Port offen sein könnte, aber die Kommunikation durch einen Paketfilter unterbunden wird. Mit der Versionserkennung (*-sV*) können Sie geschlossene von offenen Ports unterscheiden.

Ein echtes Problem beim UDP-Scanning ist die Performance. Da offene und gefilterte Ports nur selten reagieren, kommt es häufig zu Zeitüberschreitungen. Noch problematischer sind geschlossene Ports.

Nmap versucht, dieses Problem durch eine Verlangsamung der Tests zu lösen. Sie können die UDP-Scans beschleunigen, indem Sie mehr Hosts parallel scannen. Sie können außerdem die Option *--host-timeout* verwenden, um langsame Hosts von dem Check-Vorgang auszunehmen.

2.5.4 TCP-NULL-, FIN- und Xmas-Scans

Nmap stellt Ihnen mit den TCP-NULL-, FIN- und -Xmas-Scans drei weitere interessante Scan-Techniken zur Verfügung, die ein Schlupfloch im TCP RFC ausnutzen. Die Ausführung:

```
-sN; -sF; -sX
```

Der Hauptvorteil dieser Scan-Techniken: Sie können sich an zustandslosen Firewalls und paketfilternden Routern vorbeischleichen. Außerdem gelten diese Scan-Varianten als deutlich unauffälliger als beispielsweise ein SYN-Scan. Dennoch können Sie auch an IDS geraten, die diese Tests erkennen.

Von Nachteil ist, dass diese Scans keine Unterscheidung zwischen offenen und gefilterten Ports machen. Sie geben lediglich den Zustand *offen|gefiltert* aus.

2.5.5 TCP-ACK-Scan

Für sehr spezifische Aufgaben stellt Ihnen Nmap weitere Scan-Techniken zur Verfügung. Der TCP-ACK-Scan bestimmt nie offene (oder auch nur *offen|gefilterte*) Ports. Man verwendet ihn dazu, Firewall-Regeln zu bestimmen. Mit diesem Scan-Typ finden Sie außerdem heraus, ob die Regeln zustandsbehaftet sind und welche Ports gefiltert werden. Die zugehörige Option sieht wie folgt aus:

```
-sA
```

Hier ein Beispiel für die Verwendung:

```
nmap -sA scanme.nmap.org
```

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-24
15:22 Mitteleuropäische Sommerzeit
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.19s latency).
Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f
All 1000 scanned ports on scanme.nmap.org (45.33.32.156) are
unfiltered

Nmap done: 1 IP address (1 host up) scanned in 5.00 seconds
```

Nmap setzt bei der Ausführung eines ACK-Scans lediglich das ACK-Flag. Der Scanner identifiziert beim Scannen von ungefilterten Systemen offene und geschlossene Ports. Diese werden dann als *ungefiltert* gekennzeichnet. Ports, von denen keine Antwort zurückkommt oder die bestimmte ICMP-Fehlermeldungen zurückgeben (Type 3, Code 1, 2, 3, 9, 10 oder 13), werden als *gefiltert* klassifiziert.

2.5.6 TCP-Window-Scan

Der sogenannte Window-Scan dient der Unterscheidung zwischen offenen und geschlossenen Ports. Dazu wird die TCP-Fenstergröße der zurückgegebenen RST-Pakete analysiert. Systeme mit offenen Ports besitzen meist eine positive Fenstergröße, geschlossene eine Fenstergröße von null. Die Ausführung: *-sW*.

2.5.7 TCP-Maimon-Scan

Eine weitere Scan-Technik trägt den Namen seines Erfinders Uriel Maimon: der Maimon-Scan. Dieses Verfahren ist inzwischen fast 20 Jahre alt. Es ist den oben erwähnten NULL-, FIN- und Xmas-Scans sehr ähnlich. Einziger Unterschied: Es versendet ein FIN/ACK-Testpaket. Die Verwendung:

```
-sM
```

2.5.8 Benutzerdefinierter TCP-Scan

Das Schöne an Nmap ist, dass das Programm für jeden Anwendertyp die passende Werkzeuge, spricht Optionen, parat hält. Das Programm bietet Ihnen mit der Option *--scanflags* sogar die Möglichkeit, eigene Scans anzulegen. Die Verwendung:

```
--scanflags
```

Sie können numerische Flag-Werte wie z. B. 9 verwenden. Alternativ verwenden Sie beliebige Kombinationen von URG, ACK, PSH, RST, SYN und FIN.

2.5.9 Idle-Scan

Eine weitere besondere Testvariante trägt die Bezeichnung Idle-Scan. Statt Pakete von der tatsächlichen IP-Adresse an das Ziel zu senden, wird eine Art Parallel-Host (Zombie-Host) angelegt, der als Urheber des Tests versucht, Informationen über offene Ports auf dem Ziel-Host zu ergattern.

Bei Hackern ist diese Scan-Methode beliebt, weil Intrusion Detection-Systeme den Zombie-Host als Urheber identifizieren. Die Verwendung dieses Scan-Typ erfolgt mit dieser Option:

```
-sI <zombie-host>[:<port>]
```

Gerade bei Hackern ist dieser Scan-Typ sehr beliebt, weil er extrem unauffällig ist. Er erlaubt es zudem, IP-basierte Vetrauensbeziehungen zwischen Rechnern festzustellen.

2.5.10 IP-Protokoll-Scan

Wenn Sie herausfinden wollen, welche IP-Protokolle (TCP, ICMP, IGMP etc.) von Zielsystemen unterstützt werden, verwenden Sie einen IP-Protokoll-Scan. Dabei handelt es sich nicht im eigentlichen Sinne um einen Port-Scan, da das Scannen über Nummern von IP-Protokollen erfolgt. Diese Methode verlangt folgende Option: *-sO*. Hier ein Beispiel für die Ausgabe:

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-24
17:47 Mitteleuropäische Sommerzeit

Nmap scan report for scanme.nmap.org (45.33.32.156)

Host is up (0.20s latency).

Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f

Not shown: 240 closed protocols

PROTOCOL STATE         SERVICE

1        open          icmp

2        open|filtered igmp

4        open|filtered ip

6        open          tcp

17       open          udp

41       open|filtered ipv6

47       open|filtered gre

50       open|filtered esp

51       open|filtered ah

103      open|filtered pim

105      open|filtered scps

108      open|filtered ipcomp

132      open          sctp

136      open|filtered udplite

165      open|filtered unknown

191      open|filtered unknown


Nmap done: 1 IP address (1 host up) scanned in 293.95 seconds
```

2.5.11 FTP-Bounce-Scan

Wenn Sie sich für FTP-spezifische Informationen interessieren, bietet der FTP-Bounce-Scan interessante Möglichkeiten. Dieser Typ nutzt die Möglichkeit sogenannter Proxy-FTP-Verbindungen. Sie können dabei ein Drittsystem für einen Scan-Vorgang missbrauchen. Der FTP-Server führt dabei in Ihrem Auftrag einen Port-Scan auf den Zielsystemen durch. Dieser Scan-Typ eignet sich insbesondere für das Umgehen von Firewalls. Die Verwendung eines FTP-Servers erfolgt mit der Option *-b*:

```
-b <FTP-Relay-Host>
```

Der Bounce-Scan nutzt eine Schwachstelle des FTP-Servers. Heute dürften diese allerdings bei den meisten Servern gepatcht sein.

2.6 Port-Auswahl

Neben der Verwendung verschiedener Scan-Techniken bietet Ihnen Nmap auch die Möglichkeit, die Auswahl der Ports und die Scan-Reihenfolge zu bestimmen. Nmap kann die Ports sequentiell oder per Zufallsmechanismus abfragen. Ich hatte es bereits erwähnt: Der Port-Scanner prüft standardmäßig für jedes System die 1000 meistbenutzten Ports.

Wenn Sie die Standardeinstellung durch eine eigene Auswahl überschreiben wollen, müssen Sie den oder die Port-Bereiche wie folgt definieren:

```
-p <port-bereich>
```

Sie können Ports einzeln oder Port-Bereiche durch Bindestriche spezifizieren, also beispielsweise 1-1024. Sie können auch das gesamte Port-Spektrum scannen: Die Option *-p-* prüft alle Ports von 1 bis 65535. Eine Besonderheit ist bei IP-Protokoll-Scans (*-sO*) zu beachten: Hier gibt diese Option die Protokollnummern an, die gescannt werden sollen.

Sie können Scan-Vorgänge auch explizit auf TCP- oder UDP-Ports beschränken. Dazu stellen Sie einfach den Port-Nummern ein *T:* bzw. *U:* voran. Die Einschränkung greift so lange, bis Sie einen anderen Bezeichner verwenden. Ein Beispiel erläutert die Verwendung:

```
-p U:54,120,137,T:21-25,80,5360,8080
```

Diese Konfiguration prüft die UDP-Ports 53, 111 und 137 sowie die TCP-Ports 21 bis 25, 80, 5360 und 8080. Um sowohl UDP als auch TCP zu scannen, müssen Sie die Option *-sU* und mindestens einen TCP-Scan-Typ (*-sS*, *-sF* oder *-sT*) angeben.

Sie können bei der Port-Angabe auch Platzhalter verwenden. Hierfür stehen Ihnen die beiden Zeichen * und ? zur Verfügung. Sie können also beispielsweise in der Scan-Konfiguration alle Dienste scannen, die mit *http* beginnen:

```
-p http*
```

Wenn Sie unsicher sind, platzieren Sie das Argument einfach in Anführungsstrichen.

Nmap bietet Ihnen außerdem die Möglichkeit, Ports bis zu einem bestimmten Wert anzugeben. Wenn Sie alle Ports kleiner, gleich 1024 prüfen wollen, verwenden Sie folgendes Argument:

```
-p [-1024]
```

Wenn Sie schnell einen ersten Überblick über eine Umgebung erhalten wollen, können Sie einen flotten, aber beschränkten Port-Scan durchführen:

```
-F
```

Bei dieser Option werden lediglich 100 statt der sonst üblichen 1000 Ports untersucht.

Die Reihenfolge der Port-Abfragen erfolgt nach einem Zufallsmechanismus, auch um Intrusion Detecting-Systemen das Erkennen von Port-Scan-Vorgängen zu erschweren. Sie können das auch deaktivieren:

```
-r
```

In diesem Fall werden die Ports der Reihe nach abgefragt.

3 Ermittlerfunktionen

Wenn Sie Nmap auf einen Ziel-Host oder ein ganzes Netzwerk loslassen, so erfahren Sie in der Regel, welche Ports offen sind und welche Dienste dort ausgeführt werden. Doch oftmals ist damit der eigene Wissensdurst noch längst nicht gestillt und Sie wollen wissen, welches Betriebsystem auf einem bestimmten Rechner ausgeführt wird und welche Version dort zum Einsatz kommt. Gerade für Hacker sind diese Informationen sehr hilfreich, da man aufgrund der Kenntnis des verwenderten Betriebssystems Folgeschritte planen kann.

In diesem Kapitel schauen wir uns an, welche Möglichkeiten Nmap für die Dienst-, Versions- und Betriebssystemerkennung bietet. Wie wir gleich sehen werden, sind diese Möglichkeiten sehr umfangreich – gerade auch für die Remote-Erkennung.

3.1 Services ermitteln

Wenn Sie einen Ziel-Host mit Nmap unter die Lupe nehmen und dieser die offenen Ports 25/tcp, 80/tcp und 53/udp meldet, so wissen Sie direkt, dass es sich dabei mit hoher Wahrscheinlichkeit um einen Mailserver (SMTP), einen Webserver (HTTP) und einen Nameserver (DNS) handelt.

Diese Standard-Ports kennt heute jeder Administrator. Doch das bedeutet natürlich noch lange nicht, dass ein Mailserver zwingend auf dem Standard-Port ausgeführt wird. Nicht selten verwenden Administratoren auch völlig untypische Ports – und erhöhen somit mit minmalem Aufwand die Sicherheit der Umgebung.

Zur Beurteilung der Angreifbarkeit oder auch einfach nur zur Erstellung eines Netzwerkinventars müssen Sie aber exakt wissen, welche Dienste sich hinter welchem Port verbergen.

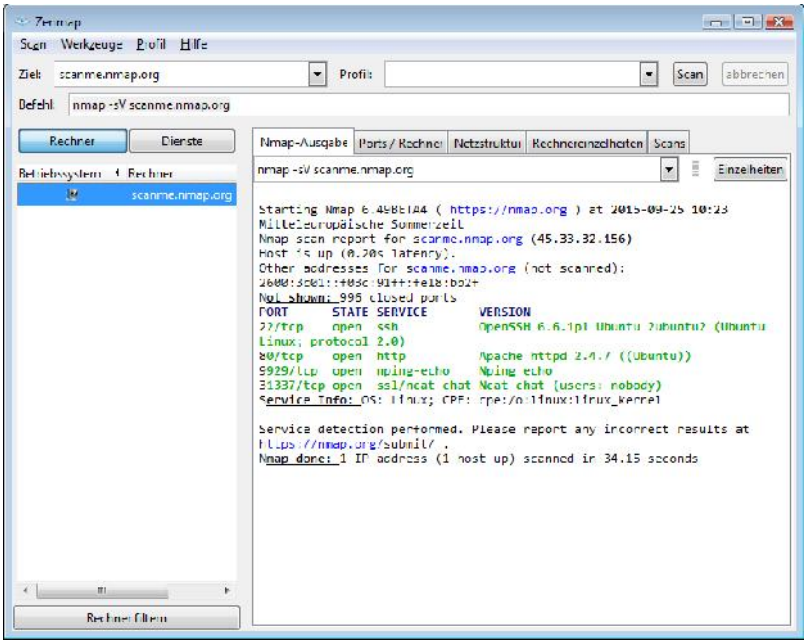
Sie sollten, wann immer möglich, auch die exakte Version ermitteln, dann nur so können Sie die tatsächliche Verwundbarkeit bestimmter Infrastrukturkomponenten ermitteln – Stichwort Exploit. Doch wie gehen Sie dazu vor? Sie ahnen es schon: Nmap stellt Ihnen eine leistungsfähige Versionserkennung zur Verfügung, mit der Sie exakt diese Information herausfinden.

Hat Nmap die TCP- und UDP-Ports der Hosts mit den oben beschriebenen Methoden analysiert, folgt im nächsten Schritt die Versionserkennung, die die Ports abfragt. Diese Erkennung versucht in Erfahrung zu bringen, was tatsächlich auf den Ports läuft.

Dabei greift Nmap auf die Datenbank *nmap-service-probes* zurück, die Testpakete für verschiedenste Abfragen enthält. In dieser Phase versucht Nmap beispielsweise Dienstprotokolle wie FTP, SSH, Telnet und HTTP sowie den Anwendungsnamen (Apache httpd, telnetd etc.), die Versionsnummer, den Host-Namen, den Gerätetyp (z. B. Drucker, Router), die Betriebssystemfamilie (z. B. Windows, Linux) und verschiedene weitere Details zu bestimmen.

Es versteht sich angesichts der Vielzahl von Diensten, dass nicht alle Services diese Informationen besitzen. Haben Sie Nmap mit OpenSSL-Unterstützung kompiliert, verbindet sich der Scanner mit SSL-Servern, um den hinter dieser Verschlüsselungsebene ausgeführten Dienst zu erkennen. Die Versionserkennung versucht auch, UDP-Ports eine Antwort zu entlocken.

Erhält Nmap von einem Dienst eine Antwort, zu der es in der Datenbank keine Übereinstimmung gibt, wird ein spezieller Fingerprint und eine URL ausgegeben. Diese Information können Sie an die Entwickler senden, damit sie dann gesichtet werden kann. Diese aktive Unterstützung der Anwender hat in der Vergangenheit dazu geführt, dass Nmap heute weit über 3.500 Musterübereinstimmungen für über 350 Protokolle wie SMTP, FTP, HTTP etc. kennt.



Eine Versionserkennung mit Zenmap.

Die Versionserkennung aktivieren Sie mit folgender Option:

```
-sV
```

Voranstehende Abbildung zeigt die Verwendung der Versionserkennung am Beispiel des Nmap-Servers *scanme.nmap.org*. Wie Sie obiger Ausgabe entnehmen können, können die Versionen von vier Diensten recht genau ermittelt werden. Sie können alternativ auch die Option *-A* verwenden, die unter anderem auch die Versionserkennung aktiviert.

Bei der Versionserkennung schließt Nmap den TCP-Port 9100 aus, weil verschiedenene Druckermodelle schlicht alles drucken, was man an diesen Port übermittelt. Sie können dieses Verhalten nutzen, indem Sie die Exclude-Anweisung in *nmap-service-probes* bearbeiten. Alternativ können Sie auch die Option *--allports* verwenden. Sie sorgt dafür, dass alle Ports gescannt werden, und zwar unabhängig von einer Exclude-Anweisung:

```
--allports
```

Der Versionserkennungs-Scan mit dem Argument *–sV* senden Testpakete an den Ziel-Host. Diesen Paketen ist eine Intensitätstufe mit den Werten von 0 bis 9 zugewiesen. Der Standardwert lautet 7. Je höher der Wert ist, um so eher gelingt die Diensterkennung.

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 sollen. Je höher die Zahl, desto wahrscheinlicher wird der Dienst richtig identifiziert. Sie können diese Intensität auch gezielt steuern. Hierfür verwenden Sie folgende Option:

```
--version-intensity <wert>
```

Die Intensität geht allerdings zu Lasten der Performance. Hier müssen Sie im Einzelfall die ideale Konfiguration ermitteln. Dazu ist der Standardwert 7 natürlich eine gute Ausgangsbasis. Anstelle der Wertes 2 (*--version-intensity 2*) können Sie auch einen Alias verwenden:

```
--version-light
```

Einen entsprechenden Alias gibt es auch für die volle Intensität (*--version-intensity 9*). Die entsprechende Option lautet wie folgt:

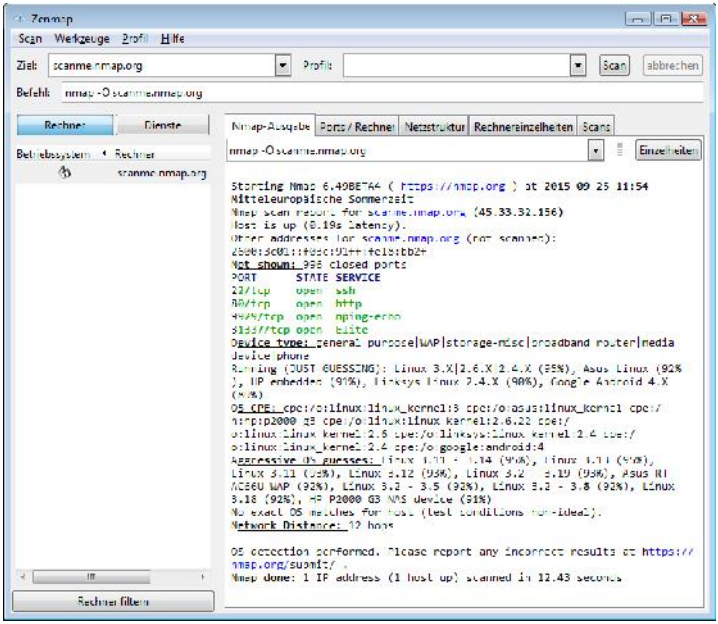
```
--version-all
```

Wenn Sie genau wissen wollen, welche Aktionen die Versionserkennung gerade ausführt, aktivieren Sie die Ausgabe der Debugging-Information. Dabei handelt es sich um ein Subset von *--packet-trace*:

```
--version-trace
```

Bei der Versionserkennung kann auch ein sogenannter RPC-Scan sinnvoll sein. Diese Option überschwemmt alle offenen TCP -/UDP-Ports mit NULL-Befehlen. Das Ziel ist die Ermittlung von RPC-Ports und gegebenenfalls die Erkennung des dort ausgeführten Programms samt Version. Und so führen Sie einen RPC-Scan aus:

```
-sR
```



Die Betriebsystemerkennung mit Nmap.

3.2 Betriebssystem ermitteln

Einer der beliebtesten Anwendungsbereiche von Nmap ist die Erkennung des Betriebssystems auf Remote-Hosts mit Hilfe von TCP/IP-Stack-Fingerprinting. Dabei übermittelt Nmap verschiedene TCP- und UDP-Pakete an den entfernten Host und nimmt dann quasi jedes Bit in der Antwort genauer unter die Lupe.

Nachdem Duzende Tests durchgeführt sind, vergleicht Nmap die Ergebnisse mit den Daten in der Datenbank *nmap-os-db*. Dort sind mehrere Tausend bekannte Betriebssystem-Fingerprints festgehalten. Findet Nmap eine Übereinstimmung, gibt der Scanner die Betriebssystemdetails aus.

Ein Fingerprint ist durch verschiedene Informationen gekennzeichnet. Dort sind üblicherweise eine kurze Beschreibung des Betriebssystems und eine Klassifikation festgehalten, der man folgende Informationen entnehmen kann:

- Herstellername
- Betriebssystem
- Generation
- Gerätetyp

Gelingt es Nmap nicht, das Betriebssystem auf dem Host zu identifizieren, können Sie über eine eingeblendete URL im Ausgabefeld den Fingerabdruck zur weiteren Analye an die Entwickler übermitteln.

Um die Betriebssystem-Erkennung zu aktivieren, verwenden Sie folgende Option:

```
-O
```

Sie können auch alternativ die Option *-A* verwenden, um die Betriebssystem-Erkennung mit weiteren Tests zu aktivieren.

Die Betriebssystemerkennung führt intensive Checks mit den Zielsystemen aus. Daraus ergibt sich eine entsprechend lange Testdauer. Sie können die Betriebssystem-Erkennung allerdings auf vielversprechende Ziele beschränken. Dazu verwenden Sie folgende Option:

```
--osscan-limit
```

Prinpiziell verläuft die Betriebssystem-Erkennung deutlich effizienter, wenn Nmap mindestens einen offenen und einen geschlossenen TCP-Port identifiziert hat.

Wenn Sie die limitierte Scan-Variante verwenden, lässt Nmap all die Hosts aus, die dieses Kriterium nicht erfüllen. Das spart viel Zeit, insbesondere bei umfangreichen Scan-Konfigurationen mit *-PN*.

Kann Nmap keinen eindeutigen Kandidaten identifizieren, gibt das Programm eine Liste von möglichen Übereinstimmungen aus. Sie können aber auch die Ergebnisse der Betriebssystem-Erkennung raten:

```
--osscan-guess; --fuzzy
```

Nmap ist in der Regel ein hartnäckiges Werkzeug, das immer dann, wenn es keine Übereinstimmung mit dem Datenbankeinträgen finden kann, noch weitere Male die Betriebssystem-Erkennung versucht. Nmap führt standardmäßig fünf Versuche durch. Sie können die Ausführung allerdings beschleunigen, indem Sie den Wert beispielsweise auf 2 senken. Die zugehörige Option lautet:

```
--max-os-tries <1-5>
```

Nachdem Sie die wichtigsten Techniken für das Erkennen von Diensten und Betriebssystemen kennen, schauen wir uns als Nächstes an, wie Sie die Ausführung optimieren.

4 Ausführung optimieren

In den vorangegangenen Kapiteln haben Sie die Grundlagen für die Ausführung von Nmap und seine wichtigsten Scan-Techniken und Optionen kennengelernt. In diesem Kapitel schauen wir uns nun verschiedene Möglichkeiten an, wie Sie die Ausführung optimieren können. Dabei insbesondere die Möglichkeiten, die helfen, die Ausführug von Nmap zu beschleunigen und wie Sie Firewall- und Intrusion Detection-Systeme umgehen können. Außerdem interessieren uns, die Ausgabemöglichkeiten von Nmap, die Ihnen beispielsweise das Erzeugen von HTML-basierten Berichten ermöglichen.

4.1 Bessere Performance

Gordon Lyon, der Nmap-Etnwicker, und sein Team haben von Anfang an viel Wert auf eine gute Performance gelegt. Der Standardscan mit dem Kommando *nmap <hostname>* eines Ziels in einem lokalen Netzwerk dauert in der Regel nur eine Fünftelsekunde. Doch in der Praxis sind es zusätzliche Scan-Optionen wie ein UDP-Scan oder gerade auch die Versionserkennnung, die zu langen Scan-Zeiten führen. Auch die Existenz von Firewalls beeinflussen die Scan-Zeiten erheblich.

Um dieses Problem zu lösen, stellt Ihnen Nmap verschiedene Konfigurationsmöglichkeiten für die Performance-Optimierung zur Verfügung. Sie können insbesondere verschiedene Timing-Parameter bei der Ausführung verwenden.

Einige dieser Optionen verlangen die Angabe eines Zeit-Parameters. Der wird standardmäßig in Millisekunden angegeben. Wenn sich diese Notation allerdings als unpraktisch erweißt, können Sie auch den Zusatz *s*, *m* oder *h* für Sekunden, Minuten oder Stunden anhängen. Konkret sind beispielsweise die Time-out-Werte, die Sie mit *--host-timeout* anlegen, von *900000ms*, *900s* und *15m* identisch.

Um das Scannen zu beschleunigen, bedient sich Nmap eines einfachen Tricks: Das Programm kann Port- oder Versions-Scans von mehreren Hosts parallel ausführen. Dazu wird der Ziel-IP-Adressraum in Gruppen aufgeteilt und dann jeweils eine Gruppe nach der anderen gescannt. Dabei gilt die Verwendung von größeren Gruppen prinzipiell als effizienter.

Doch diese Vorgehensweise hat auch einen Nachteil: Die Host-Ergebnisse werden erst dann ausgegeben, sobald die vollständige Host-Gruppe gescannt wurde. Wenn Sie also beispielsweise eine Gruppengröße von 40 verwenden, werden die Ergebnisse erst dann ausgegeben, wenn alle 40 Ziele untersucht wurden.

Dieses Problem löst Nmap auf seine ganz eigene Weise. Der Port-Scanner beginnt mit einer relativ kleinen Gruppengröße von etwa fünf Hosts. So kann Nmap zügig die ersten Ergebnisse ausgeben. Im Folgenden erhöht der Scan die Gruppengröße dann kontinuierlich bis auf den Wert 1024. Wie groß die Gruppen tatsächlich werden, hängt insbesondere von den verwendeten Optionen ab. Nmap erzeugt größere Gruppen bei UDP-Scans und bei TCP-Scans kleinere.

Unabhängig von dem Automatismus können Sie die Größe der parallel gescannten Gruppen konfigurieren:

```
--min-hostgroup <anzahl_hosts>
```

```
--max-hostgroup <anzahl_hosts>
```

Dabei gibt das Argument *--max-hostgroup* die maximale Gruppengröße an. Dieser Wert wird nicht überschritten. Mit dem Argument *--min-hostgroup* konfigurieren Sie entsprechend die minimale Größe. Sollte die Zahl der Hosts unterhalb dem angegebenen Minimum liegen, verwendet Nmap automatisch einen kleineren Wert.

Nun stellt sich natürlich die Frage, welches denn praktikable Werte sind? Wenn Sie ein Class C-Netzwerk scannen, so wird häufig der Wert 256 verwendet. Sind viele Ports abzufragen, bringt ein höherer Wert in der Regel keine Beschleunigung. Wenn Sie allerdings nur vergleichweise wenige Ports prüfen, können Sie auch Werte von 2048 und höher verwenden.

Eine weitere Performance-relevante Einstellung ist die Konfiguration des Timeout-Werts. Mit diesem Wert bestimmen Sie, wie lange Nmap auf eine Antwort zu einem Testpaket wartet, bevor das Programm dieses verwirft oder ein neues Paket sendet. Im Normalfall müssen Sie sich nicht um die Konfiguration des Wertes kümmern, denn Nmap berechnet diesen Wert auf Grundlage der Antwortzeiten bei früheren Testpaketen. Bei extrem hohen Latenzzeiten kann dieser Wert schnell bis auf mehrere Sekunden wachsen.

Sie können die Timeout-Werte von Testpaketen aber auch manuell bestimmen:

```
--min-rtt-timeout <zeit>
```

```
--max-rtt-timeout <zeit>
```

```
--initial-rtt-timeout <zeit>
```

Auch bei dieser Konfiguration stellt sich die Frage nach empfehlenswerten Einstellungen. In der Regel sollte der Wert bei 100 und maximal 1000 Millisekunden liegen.

Einen bisweilen deutlichen Performance-Gewinn kann auch die Anpassung der Sendeversuche von Testpaketen bringen. Erhält der Scanner auf ein Paket keine Antwort, so versucht er es erneut. Identifiziert Nmap Netzwerkprobleme, so wird dieser Versuch mehr oder minder häufig unternommen. Mit der folgenden Option konfigurieren Sie die Anzahl der Wiederholungen:

```
--max-retries <anzahl_wiederholungen>
```

Insbesondere dann, wenn Sie eine maximale Performance wünschen, können Sie den Wert verringern. Der Wert 3 gilt als praktikabel bei langsamen Hosts und kann zu einer erheblichen Beschleunigung führen.

Wenn Sie überwiegend langsame Ziel-Hosts prüfen wollen, so kann es auch sinnvoll sein, dem Scan deutlich mehr Zeit zu geben. Langsam kann in diesem Zusammenhang vieles bedeuten, beispielsweise eine schlechte Netzwerkverbindung, unzuverlässige Netzwerkkomponenten oder auch langsame Software. All diese Kriterien können eine Prüfung verlangsamen. Auch restriktive Firewalls sind eine mögliche Ursache.

Um Nmap auch in solchen Umgebungen ausreichend Zeit zu geben, können Sie den Timeout-Wert entsprechend hochsetzen. Wenn Sie den Wert beispielsweise auf 15 Minuten setzen, bricht Nmap die Prüfung nach Ablauf dieser Zeitspanne ab:

```
--host-timeout <zeit>
```

Nmap kann auch die Verzögerung zwischen zwei Testpaketen bestimmen. Die zugehörige Konfiguration sieht wie folgt aus:

```
--scan-delay <zeit>
--max-scan-delay <zeit>
```

Wenn Sie diese Option verwenden, wartet Nmap die angegebene Zeitspanne mit dem Versand weiterer Pakete an einen Host. Nmap versucht auch hier standardmäßig, die Scan-Verzögerung automatisch an die Gegebenheiten anzupassen. Prinzipiell kann ein niedriger Wert für *--max-scan-delay* Nmap beschleunigen. Allerdings kann ein niedriger Wert auch zu unnötig vielen Wiederholungen führen. Durch den optimalen Einsatz der Option *--scan-delay* gelingt es außerdem, Intrusion Detection- und Prevention-Systeme zu umgehen.

Nmap verwendet ein dynamisches Timing, um die ideale Scan-Rate zu ermitteln. Doch in der Praxis zeigt sich immer mal wieder, dass dieser Automatismus nicht die optimalen Einstellungen bietet und Sie stattdessen eine eigene Konfiguration

verwenden sollten. Sie können beispielsweise dafür sorgen, dass Nmap nicht zu schnell scannt. Für die Steuerung der Scan-Rate stehen Ihnen die beiden folgenden Optionen zur Verfügung:

```
--min-rate <werte>
--max-rate <werte>
```

Mit der Mindestrate bestimmen Sie, wie schnell Nmap Testpakete mindestens verschickt – meist schneller. Wenn Sie beispielsweise das Argument *--min-rate 200* verwenden, verschickt Nmap mindestens 200 Pakete pro Sekunden. Bei günstigen Testbedingungen auch deutlich mehr.

Entsprechend können Sie mit der Maximalrate den Wert bestimmen, der nicht überschritten werden soll. Mit *--max-rate 400* würden beispielsweise maximal 400 Pakete pro Sekunde verschickt. Sie können auch beide Argumente kombinieren und so einen Bereich für die Scan-Rate vorgeben.

In den vorangegangenen Abschnitten haben Sie verschiedene Möglichkeiten zur zeitlichen Steuerung der Scan-Tests kennengelernt. Die sind flexibel und ideal für alle jene Anwender, die bereits über gewisse Erfahrungen mit Nmap verfügen. Doch gerade Einsteiger tun sich bei der Wahl und Konfiguration der optimalen Einstellungen schwer.

Auch für dieses Problem haben die Nmap-Entwickler die passende Lösung: Mit Timing-Templates können Sie auf vordefinierte Zeitsteuerungen zurückgreifen, die gängige Anforderungen abdecken. Für die Verwendung der Templates ist die Angabe der Option *-T* und die Nummer erforderlich. Die Templates werden von 0 bis 5 durchnummeriert:

- paranoid (0)
- sneaky (1)
- polite (2)
- normal (3)
- aggressive (4)
- insane (5)

Die beiden Templates *paranoid* und *sneaky* verwendet man üblicherweise zur Umgebung von Intrusion Detection-Systemen. Das Polite-Template verlangsamt den Scan-Vorgang, um weniger Bandbreite und Ressourcen auf dem Zielrechner zu beanspruchen.

Das Template *normal* nimmt keinerlei Änderungen der zeitlichen Ausführung vor. Anders der Aggressive-Modus. Er beschleunigt den Scan-Vorgang, indem er davon ausgeht, dass Sie sich in einem schnellen und zuverlässigen Netzwerk befinden. Wenn Sie bereit sind, zuguten von mehr Geschwindigkeit auf Genauigkeit zu verzichten, aktivieren Sie den Modus *insane*.

Die Auswahl des Timing-Templates erfolgt nach diesem Schema:

```
-T paranoid|sneaky|polite|normal|aggressive|insane
```

Sie können die Timing-Templates dennoch mit Zeitwerten kombinieren. Bei der Konfiguration *-T4* steigt die dynamische Scan-Verzögerung bei TCP-Ports über 10 Millisekunden an, bei *-T5* über 5 Millisekunden hinweg. Wenn Sie immer nur einen Port nach dem anderen scannen wollen, aktivieren Sie mit *-T0* die Serialisierung der Scans.

4.2 Firewall und IDS umgehen

Netzwerke schützt man üblicherweise durch eine Firewall. Dieser Schutz wird häufig durch ein Intrusion Detection-System ergänzt, um potenzielle Hacker-Angriffe zu erkennen. Für das Port- und Security Scanning sind dies Hindernisse, die eine Zielanalyse (erheblich) erschweren. Dennoch bietet Nmap auch hierfür Möglichkeiten, wie man solche Systeme erkennen und umgehen kann. Der Hintergedanke ist dabei das Penetration Testing aus Perspektive eines Hackers.

Um es vorwegzunehmen: Es gibt kein allgemeines Rezept für die Erkennung und Umgehung von Schutzmechanismen. Aber Nmap bietet verschiedene Optionen und je mehr Erfahrung Sie im Umgang mit den Möglichkeiten des Scanners sammeln, umso zielgerichteter können Sie diese einsetzen.

Eine einfache, aber sehr effektive Methode, es Paketfiltern, Intrusion Detection-Systemen und anderen Schutzmechanismen zu erschweren, Scans zu erkennen und deren Muster zu verstehen, stellt das Fragmentieren von IP-Paketen dar. Hierzu verwenden Sie folgende Option:

```
-f
```

Dabei wird der TCP-Header in mehrere Pakete aufgeteilt und genau das macht es Schutzmechanismen schwer, Scan-Vorgänge als solche zu identifizieren. Alternativ können Sie auch eine eigene Offset-Größe mit der Option --*mtu* angeben. Allerdings sollten Sie diese beiden Optionen nicht miteinander kombinieren.

Beachten Sie außerdem, dass die Fragmentierung von Nmap nur für rohe Pakete unterstützt wird, die Sie mit TCP- und UDP-Port-Scans und der Betriebssystemerkennung verwenden können. Die Versionserkennung und die Nmap Scripting Engine unterstützen keine Fragmentierung.

Eine andere Möglichkeit ist der sogenannte Decoy-Scan. Bei dem verstecken Sie den Scan mit Ködern. Dieser Scan erweckt bei den entfernten Hosts den Anschein, dass der oder die Hosts, die Sie als Köder angeben, das Zielnetzwerk ebenfalls scannen. Ein IDS kann in etwa 5 bis 10 Port-Scans von eindeutigen IP-Adressen registrieren, weiß aber nicht, welche IP sie gescannt hat und welche Köder waren.

Die zugehörige Option sieht wie folgt aus:

```
-D <decoy1>[,<decoy2>][,ICH][,...]
```

Die Köder trennen Sie mit Kommata voneinander. Ihre eigene IP-Adresse (*ICH*) geben Sie am besten an sechster Stelle oder später an. Dann stehen Ihre Chancen gut, dass Sie nicht als Scannender erkannt werden. Ganz wichtig ist natürlich, dass die Host, die Sie als Köder angeben, eingeschaltet sind. Sind sie es nicht, registrieren Sie das moderne IDS.

Nmap unterstützt auch das Spoofing, also das Fälschen der eigenen IP-Adresse. Dazu verwenden Sie die Option *-S*. Einziges Problem: Sie erhalten keine verwertbaren Informationen zurück, weil der Ziel-Host die Anworten an die gefälschte Adresse übermittelt.

Sie können in Nmap auch explizit die Schnittstelle angeben, die Sie für das Senden und Empfangen verwenden wollen.

```
-e <interface>
```

Auch die Quell-Portnummer können Sie mit Nmap vortäuschen:

```
--source-port <port-nummer>
-g <port-nummer>
```

Nmap verschickt überlicherwiese minimale Datenpakete, die nur einen Header enthalten. Aus diesem Grund sind die verschickten TCP-Pakete im Allgemeinen nur 40 Bytes groß, die ICMP Echo-Requests sogar nur 28 Bytes. Dieses Muster kennen natürlich auch IDS bzw. kann von Pakefiltern ermittelt werden.

Nmap bietet Ihnen hier die Möglichkeit, Zufallsdaten an gesendete Pakete anzufügen:

```
--data-length <wert>
```

Allerdings funktioniert das nicht bei der Betriebssystemerkennung (*-O*), wohl aber bei den meisten Ping- und Port-Scan-Paketen.

Eine weitere Möglichkeit, Intrusion Detection-Systemen zumindest das Erkennen von Scan-Vorgängen zu erschweren, ist die zufällige Scan-Reihenfolge:

```
--randomize-hosts
```

Für Netzwerk-Überwachungssysteme wird es schwierger, diese Vorgänge zu erkennen, insbesondere dann, wenn Sie die Tests mit langsamen Timing-Optionen kombinieren.

Nmap kann auch eine MAC-Adresse vortäuschen:

```
--spoof-mac <MAC-Adresse>
```

Schließlich können Sie Pakete mit falschen TCP/UDP-Prüfsummen verschicken. Auch das hilft, Firewalls und IDS das Leben schwer zu machen:

```
--badsum
```

Prinzipiell können die genannten Möglichkeiten helfen, ein Blockieren bzw. ein Erkanntwerden zu erschweren.

4.3 Berichtausgabe

Was nutzt ein noch so tolles Test- und Analysewerkzeug, wenn es die gewonnenen Erkenntnisse nicht in einer praktikablen Form ausgeben kann? Natürlich herzlich wenig. Aus diesem Grund unterstützt Nmap verschiedene Ausgabeformate und einen interaktiven Ausgabemodus. Nmap unterstützt auch die XML-Ausgabe, um die Berichtinformationen in Drittanwendungen weiter zu verarbeiten. Zusätzlich bietet Nmap Optionen zur Steuerung der Ausführlichkeit dieser Ausgabe, mit denen Sie beispielsweise die Debugging-Meldungen steuern können.

Der Port-Scanner generiert seine Ausgabe in fünf verschiedenen Formaten. Das Standardformat ist die interaktive Ausgabe. Sie wird an die Standardausgabe gesendet – also an das Terminal bzw. die Konsole, auf der Sie Nmap ausführen. Die „normale“ Ausgabe gibt weniger Laufzeitinformation und Warnungen aus. Sie geht davon aus, dass eine Analyse erst nach dem Abschluss des Scans erfolgt. Das wichtigste Ausgabeformat ist XML, weil Sie es einfach nach HTML konvertieren oder in Drittprogrammen wie Datenbanken etc. verwenden können. Dann kennt Nmap noch zwei einfache grepbare Ausgaben.

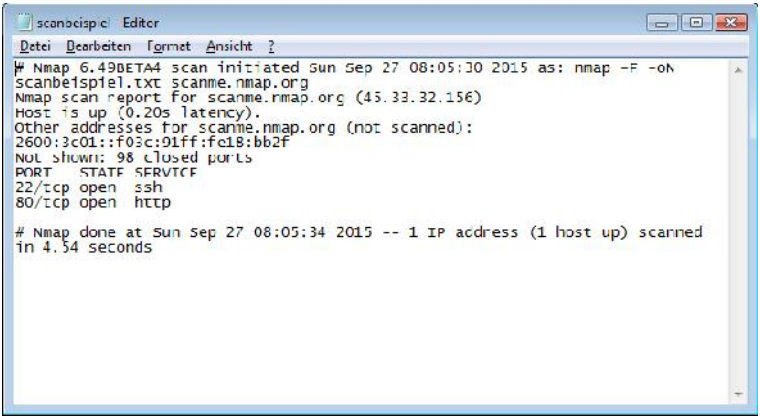
Besonders einfach können Sie ein Scan-Ergebnis in eine Textdatei schreiben. Dazu verwenden Sie die Option *-oN <dateiname>*. Hier ein Beispiel, das den Scan des Nmap-Demoservers in die Datei *scanbeispiel.txt* schreibt:

```
nmap -F -oN scanbeispiel.txt scanme.nmap.org


Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-27
08:05 Mitteleuropäische Sommerzeit
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.20s latency).
Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 98 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http


Nmap done: 1 IP address (1 host up) scanned in 4.54 seconds
```

Die Ausgabedatei können Sie dann je nach verwendetem Betriebssystem öffnen. Unter Linux verwenden Sie dazu am einfachsten den Befehl *cat scanbeispiel.txt* und öffnen die Textdatei mit dem Standardeditor.



Die Ausgabe im Textformat.

Die Ausgabeoption *-oN* kann mit weiteren Ausgabeformaten kombiniert werden. Sie können die Ausgabe beispielsweise in eine Text- und eine XML-Datei gleichzeitig schreiben:

```
nmap -A -oN normale-ausgabe.txt -oX xml-ausgabe.xml scanme.nmap.org
```

Um das Scan-Ergebnis in eine XML-Datei zu schreiben, verwenden Sie die Option *-oX <dateiname>*. Der vollständige Befehl sieht dann wie folgt aus:

```
nmap -A -O -oX scanergebnis.xml scanme.nmap.org
```

Die dabei erzeugte XML-Datei sieht dann wie folgt aus (hier die gekürzte Form):

```
<?xml version="1.0" encoding="UTF-8"?>
<?DOCTYPE nmaprun?>
<!xml-stylesheet href="file:///C:/Program Files/Nmap/nmap.xsl" type="text/xsl">
```

```
<!-- Nmap 6.49BETA4 scan initiated Sun Sep 27 10:03:43 2015
as: nmap -A -O -oX scanergebnis.xml scanme.nmap.org -->

<nmaprun scanner="nmap" args="nmap -A -O -oX scanergebnis.xml
scanme.nmap.org" start="1443341023" startstr="Sun Sep 27
10:03:43 2015" version="6.49BETA4" xmloutputversion="1.04">

<scaninfo type="syn" protocol="tcp" numservices="1000" servi-
ces="1,3-4,6-7,9,13,17,19-26,30,32-33,37,42-43,49,53,70,79

…

64623,64680,65000,65129,65389"/>

<verbose level="0"/>

<debugging level="0"/>

<host starttime="1443341029" endtime="1443341076"><status
state="up" reason="echo-reply" reason_ttl="53"/>

<address addr="45.33.32.156" addrtype="ipv4"/>

<hostnames>

<hostname name="scanme.nmap.org" type="user"/>

<hostname name="scanme.nmap.org" type="PTR"/>

</hostnames>

<ports><extraports state="closed" count="996">

<extrareasons reason="resets" count="996"/>

<elem key="type">ssh-dss</elem>

…

<trace port="3389" proto="tcp">

<hop ttl="2" ipaddr="217.0.119.81" rtt="18.00"/>

<hop ttl="3" ipaddr="217.0.65.222" rtt="20.00"/>

<hop ttl="4" ipaddr="217.239.48.182" rtt="25.00"/>

…

<runstats><finished time="1443341076" timestr="Sun Sep 27
10:04:36 2015" elapsed="54.23" summary="Nmap done at Sun Sep
27 10:04:36 2015; 1 IP address (1 host up) scanned in 54.23
seconds" exit="success"/><hosts up="1" down="0" total="1"/>

</runstats>

</nmaprun>
```

Sie können die Ergebnisse auch in eine SQlite- oder MySQL-Datenbank schreiben. Dazu installieren Sie auf einem Linux-System das Modul PBNJ. Das können Sie unter Debian-basierten Umgebungen einfach installieren:

```
apt-get install pbnj
```

Dann führen Sie den Befehl *scanpbnj* aus und geben die Nmap-Argumente mit der Option *-a* an:

```
scanpbnj -a "-p-" scanme.nmap.org
```

Scanpbnj speichert die Ergebnis in der Datenbank, die in der Konfigurationsdatei *config.yaml* hinterlegt ist. Die Daten werden in die Datei *data.dbl* geschrieben. Um die Daten in eine MySQL-Datenbank zu schreiben, müssen Sie lediglich den Treiber und die Datenbankinformation in der Konfigurationsdatei anpassen.

Sie können die Ergebnisse auch nach HTML konvertieren. Der Vorteil: Die Ergebnisse können beispielsweise lokal veröffentlicht und dann auch anderen Mitarbeitern zentral zur Verfügung stellt werden.

Dazu bedarf es allerdings eines Hilfsmittels: Sie benötigen einen XSLT-Prozessor, der die XML-basierten Daten mit Hilfe eines Stylesheets umwandelt. Unter Linux steht Ihnen hierfür xsltproc zur Verfügung. Wenn Sie mit Windows arbeiten, sollten Sie sich Saxon (*http://saxon.sourceforge.net*) herunterladen und installieren.

Um unter Linux mit Hilfe von xsltproc die XML-Datei nach HTML zu transformieren, führen Sie folgenden Befehl aus:

```
$xsltproc  scanergebnis.xml -o scanergebnis.html
```

Die Gestaltung der Ausgabe wird durch die Datei *nmap.xsl* bestimmt. Die können Sie gegebenenfalls anpassen.

5 Nmap in der Praxis

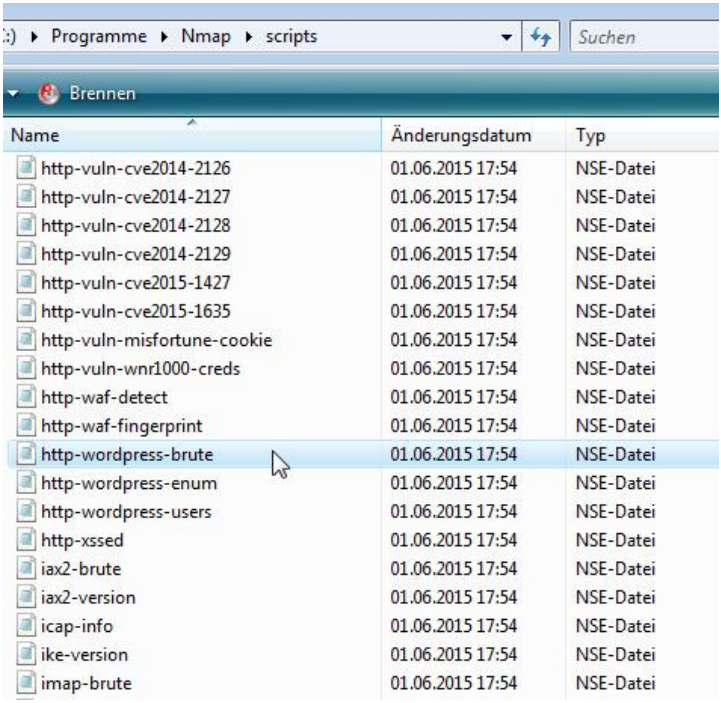
In den vorangegangenen Kapiteln haben Sie die wichtigsten Grundlagen von Nmap kennengelernt. Sie wissen insbesondere, wie Sie Remote-Hosts auf Ports untersuchen können, wie Sie das Scannen optimieren und die Ergebnisse interpretieren können.

Aber Nmap kann noch wesentlich mehr. Sie können mit dem Programm Webserver auditieren und beispielsweise die unterstützten HTTP-Methoden herausfinden und den Status des HTTP-Proxy Servers auslesen. Sie können ermitteln, welche interessanten Dateien und Verzeichnisse sich auf dem Webserver befinden und den Authentifizierungsmechanismus einer Brute-Force-Attacke aussetzen. Sie können mit Nmap Angriffe gegen Joomla!- und WordPress-Installation fahren, Web Application Firewalls erkennen und Verwundbarkeiten in Web-Applikationen aufdecken.

Nmap bietet außerdem umfangreiche datenbankspezifische Funktionen. Sie können beispielsweise MySQL-Datenbankserver auf Datenbanken, Benutzer und Variablen abfragen. Sie können mit Nmap eine Brute Force-Attacke ausführen und Zugang zu einem System erlangen. Das funktioniert auch mit Oracle- und eingeschränkt auch mit MongoDB- und CouchDB-Datenbanken.

Nicht minder interessant sind die Testmöglichkeiten von E-Mail-Servern. Sie können beispielsweise von SMTP-Servern Benutzer und offene Relays identifizieren. Sie können Passwort-Attacken ausführen und die Fähigkeiten von IMAP- und POP3-Server herausfinden.

Möglich werden diese Test durch NSE-Skripts, von denen fast 500 in der Nmap-Installation enthalten sind. Die NSE-Skripts finden Sie im Unterordner *scripts* der Nmap-Installation. Wenn Sie sich für weitere Skripts und aktuelle Entwicklung interessieren, finden Sie unter *https://nmap.org/nsedoc/* die entsprechenden Informationen und Downloads.



Im Verzeichnis *scripts* liegen die NSE-Skripts.

5.1 Webserver scannen

Webserver bilden in vielen Unternehmen das Rückgrad der IT-Infrastruktur. Auf Ihnen werden Content-Management-, CRM/ERP-, Groupware- oder E-Commerce-Systeme ausgeführt. Webserver unterstützen verschiedene HTTP-Methoden – abhängig von ihrer Konfiguration und der Implementierung.

5.1.1 HTTP-Methoden

Welche das im Einzelfall sind, können Sie mit dem NSE-Skript *http-methods* herausfinden. Es erlaubt nicht nur das Auslesen, sondern auch das Testen von womöglich sicherheitsrelevanten Schwachstellen. Und so finden Sie heraus, welche HTTP-Methoden der Nmap-Test-Server unterstützt:

```
nmap -p80,443 --script http-methods scanme.nmap.org
```

Die Ausgabe von Nmap:

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2015-09-27
14:02 Mitteleuropäische Sommerzeit

Nmap scan report for scanme.nmap.org (45.33.32.156)

Host is up (0.19s latency).

Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f

PORT    STATE  SERVICE

80/tcp  open   http

|_http-methods: GET HEAD POST OPTIONS

443/tcp closed https


Nmap done: 1 IP address (1 host up) scanned in 5.49 seconds
```

Das Argument prüft, ob ein Webserver auf Port 80 und 443 gefunden wird. Dieses Skript verwendet die HTTP-Methode *OPTIONS*, um alle unterstützten Methoden des Ziel-Hosts zu identifizieren.

Für das Penetration Testing sind inbesondere die Methoden *TRACE*, *CONNECT*, PUT und *DELETE* interessant, da sie allesamt ein Sicherheitsrisiko darstellen. Die *TRACE*-Methode ist für Cross Site Tracing-Attacken anfällig und könnte einem potenziellen Angreifer den Zugriff auf *httpOnly*-Cookies ermöglichen. Die *CONNECT*-Methode erlaubt unter Umständen die Verwendung des Servers als Web Proxy-Server. Die beiden Methoden *PUT* und *DELETE* erlauben das Verändern der Ordnerinhalte auf dem Webserver.

5.1.2 Offener Web-Proxy

Proxy-Server dienen beim Internet-Zugriff dazu, als „Handlungsvemittler“ für die Clients die Verbindungen nach draußen aufzubauen. Der entscheidende Vorteil: Nur der Proxy-Server ist für potentielle Angreifer als Angriffspunkt sichtbar. Für Angreifer sind offene Proxy-Server ein interessantes Ziel, weil sie über diese Angriffe fahren können. Nmap stellt Ihnen auch für das Prüfen von offenen HTTP-Proxy-Servern ein entsprechendes Skript zur Verfügung. Das führen Sie wie folgt aus:

```
nmap --script http-open-proxy -p8080 <ziel>
```

Dem relevanten Teil der Ausgabe können Sie dann die HTTP-Methode entnehmen, die erfolgreich getestet wurde. In diesem Fall der (womöglich) offene Proxy-Port:

```
PORT     STATE SERVICE
8080/tcp open  http-proxy
|  proxy-open-http: Potentially OPEN proxy.
|_ Methods successfully tested: GET HEAD CONNECT
```

Gegebenenfalls müssen Sie alternative Ports prüfen.

5.1.3 Interessante Dateien und Verzeichnis aufdecken

Webserver haben die Angewohnheit, schnell eine komplexe Struktur zu bekommen, wenn man sie intensiv nutzt. Mit jeder weiteren Anwendung, die Sie auf einem Webserver installieren und ausführen, wird diese Struktur komplexer. Oftmals findet man dort verwaiste Datenbanksicherungen, vergessene Konfigurationsdateien, ungesicherte Admin- und Konfigurationsverzeichnisse etc. Auch die sollten Sie aufspüren und gegebenenfalls entfernen. Nmap stellt Ihnen ein Skript zur Verfügung, mit dem Sie die Verzeichnisse analysieren und auf entsprechende Dateien hin untersuchen können. Dazu führen Sie folgenden Befehl aus:

```
nmap --script http-enum -p80 scanme.nmap.org
```

Eine entsprechende Ausgabe sieht exemplarisch wie folgt aus:

```
PORT   STATE SERVICE
80/tcp open  http
| http-enum:
|   /blog/: Blog
|   /test.php: Test page
|   /robots.txt: Robots file
|   /css/app.generic.css: PHP application
|_  /img/aoo.icon.png: PHP application
```

In der Praxis begegnen Sie natürlich den unterschiedlichsten Ausgaben. Welche das konkret sind, ist von der jeweiligen Umgebung abhängig. Hier ein weiteres Beispiel für das Erkennen und das Auslesen von interessanten Verzeichnissen und Dateien:

```
PORT   STATE SERVICE
80/tcp open  http
| http-enum:
|   /administrator/: Possible admin folder
|   /administrator/index.php: Possible admin folder
|   /home.html: Possible admin folder
|   /test/: Test page
|   /logs/: Logs
|_  /robots.txt: Robots file
```

Die Analyse einer WordPress-Installation liefert beispielsweise folgendes Ergebnis:

```
80/tcp open  http
| http-enum:
|   /wp-login.php: Possible admin folder
|   /robots.txt: Robots file
|   /wp-login.php: Wordpress login page.
|_  /readme.html: WordPress version 3.3.1
```

Das kann bei einer anderen WordPress-Installation schon wieder ganz anders aussehen.

5.1.4 Brute-Force-Attacke

Web-Applikationen, Router, Webcam und vieles mehr sind meist mit einem HTTP-Authentifizierungemechanismus geschützt. Angreifer und Penetration Tester können versuchen, diesen Schutz mit einer Brute-Force-Attacke auszuhebeln. Auch das ist mit Nmap möglich:

```
nmap -p80 --script http-brute –script-args http-
brute.path=/admin/ <ziel>
```

Gelingt dem NSE-Skript das Knacken, gibt Nmap in etwa folgende Daten aus:

```
PORT     STATE SERVICE REASON
80/tcp   open  http    syn-ack
| http-brute:
|   Accounts
|     admin:geheim => Valid credentials
|   Statistics
|_    Perfomed 2034 guesses in 29 seconds, average tps: 70
```

Brute-Force steht für „rohe Gewalt“ und bedeutet nichts anderes, als dass solange eine Liste von Passwörtern getestet wird, bis der Passwörtschutz geknackt ist. Dabei greift das Skript *http-brute* auf zwei Dateien zurück, in denen potenzielle Benutzernamen und Passwörter hinterlegt sind: *usernames.lst* und *passwords.lst*. Die Dateien liegen im Verzeichnis */nselib/data/* Ihrer Nmap-Installation. Die könenn natürlich manuell bearbeitet und erweitert werden.

Das NSE-Skript greift bei der Ausführung auf diese NSE-Bibliotheken *unpwdb* und *brute* zurück. Diese Bibliotheken unterstützen verschiedene Argumente. Unter anderem können Sie auch eigene Benutzer- und Passwortlisten verwenden. Die geben Sie mit den beiden Argumenten *userdb* und *passdb* an:

```
$ nmap -p80 --script http-brute --script-args
userdb=/var/username.txt,passdb=/var/passwort.txt <ziel>
```

Sowie die Attacke erfolgreich verlaufen ist, sollten Sie diese mit dem Argument *brute.firstOnly* beenden:

```
$ nmap -p80 --script http-brute --script-args brute.firstOnly
<ziel>
```

Nmap stellt Ihnen für die Ausführung von Brute-Force-Attacken auch einige Timing-Templates zur Verfügung, mit denen Sie die Timeout-Werte bestimmen können:

```
-T3,T2,T1 - 10 Minuten
-T4 – 5 Minuten
-T5 - 3 Minuten
```

Um ein anderes Zeitlimit zu setzen, verwenden Sie das Argument *unpwd.timelimit*. Diese Tests sind in der Regel sehr zeitintensiv.

5.1.5 Benutzer-Accounts auslesen

Das Apache-Modul *UserDir* ermöglicht den Zugriff auf Benutzerverzeichnisse, die die URI */~benutzername/* besitzen. Mit Hilfe von Nmap können Sie diese Daten auslesen und somit die entsprechenden Benutzer bestimmen. Wenn Sie die Benutzernamen besitzen, könnten Sie als Nächstes mit einer Brute-Force-Attacke versuchen, dessen Passwort zu knacken.

Die Vorgehensweise ist wieder simpel. Sie verwenden das NSE-Skript *http-userdir-enum*:

```
nmap -p80 --script http-userdir-enum <ziel>
```

Das Ergebnis führt die ermittelten Benutzernamen auf. Hier ein typisches Beispiel:

```
PORT   STATE SERVICE
80/tcp open  http
|_http-userdir-enum: Potential Users: root, web, test
```

Dieser Test greift ebenfalls auf die in Nmap hinterlegte Liste der Benutzernamem zurück (*usernames.lst*).

5.1.6 Zugangsdaten testen

Die meisten Web-Applikationen wie der Apache Tomcat Manager, Joomla!, Magento oder WordPress besitzen Standardbenutzer mit den zugehörigen Standardbenutzernamen und -passwörtern. Viele Administratoren vergessen, diese nach der Installation und Konfiguration zu deaktivieren. Das ist fahrlässig, denn sie sind für potenzielle Angreifer ein sehr vielversprechender Angriffspunkt.

Nmap besitzt ein kleines Skript, mit dem Sie die Standard-Accounts von verschiedenen Anwendungen prüfen können:

```
nmap -p80 --script http-default-accounts <ziel>
```

Hier ein Beispiel für eine entsprechende Prüfung:

```
80/tcp open  http    syn-ack
|_http-default-accounts: [Cacti] credentials found -> admin:admin
Path:/cacti/
```

Alternativ können Sie mit dem Skript *http-defaultaccounts.category* auch verschiedene Anwendungskategorien unter die Lupe nehmen:

```
nmap -p80 --script http-default-accounts --script-args http-defaultaccounts.category=web <ziel>
```

Dieses Skript unterstützt vier Kategorien:

- web - Web Application
- router – Router-Schnittstellen
- voip – VOIP-Geräte/-Anwendungen
- security – sicherheitsrelevante Software

5.1.7 Brute-Force-Attacke gegen WordPress

WordPress ist eines der beliebtesten Content-Management- und Blog-Systeme. Es wird insbesondere von vielen Unternehmen genutzt, die über Ihre Produkte und Dienstleistungen bloggen. Nmap integriert auch ein NSE-Skript, mit dem Sie eine Brute-Force-Attacke gegen das System ausführen und sich womöglich Zugang zu dem System verschaffen können.

Und so führen Sie das Skript aus:

```
$ nmap -p80 --script http-wordpress-brute <ziel>
```

Kann das Skript mit den Benutzter- und Passwortlisten den Schutz knacken, so sieht die Ausgabe in etwa wie folgt aus:

```
PORT     STATE SERVICE REASON
80/tcp   open  http    syn-ack
| http-wordpress-brute:
|   Accounts
|     Admin:Passwort => Login correct
|   Statistics
|_    Perfomed 3036 guesses in 43 seconds, average tps: 7
```

Tipp

Verschiedene Filtermechanismen blockieren Requests des Nmap HTTP User Agents. Das kann auch bei zuvor beschriebenen Tests passieren. Die Lösung ist indes recht einfach. Sie verwenden das Argument *http.useragent* und gauckeln dem Filter einen anderen Agent vor:

```
nmap -p80 --script http-wordpress-brute --script-args
http.useragent="Mozilla 42" <ziel>
```

Schon lässt der Filtermechanismus Ihren Request passieren und der Angriff kann ausgeführt werden.

5.1.8 Brute-Force-Attacke gegen Joomla!

Neben WordPress dürfte Joomla! das beliebteste Content-Managementsystem sein. Während WordPress eher für Blogger geeignet ist, deckt Joomla! die gesamte Bandbreite an Funktionen ab, die man von professionellen CMS erwartet.

Sie ahnen es schon: Auch Joomla! kann mit einem NSE-Skript geprüft werden. Und zwar können Sie mit dem Skrtipt *http-joomla-brute* das CMS einer Brute-Force-Attacke aussetzen. Und so geht‘s:

```
nmap -p80 --script http-joomla-brute <ziel>
```

Die Ausgabe einer erfolgreichen Attacke sieht dann wie folgt aus:

```
PORT     STATE SERVICE REASON
80/tcp open  http    syn-ack
| http-joomla-brute:
|   Accounts
|     user:passwort => Login correct
|   Statistics
|_    Perfomed 1404 guesses in 200 seconds, average tps: 7
```

5.1.9 Web Application Firewall erkennen

Mehr und mehr Unternehmen sind in der Vergangheit dazu übergegangen, ihre Webserver mit sogenannten Web Application Firewalls, kurz WAF, zu schützen. Für Penetration Tester ist es natürlich gut zu wissen, dass sich ein Traffic-Filter zwischen ihnen und dem Ziel befindet.

Mit Hilfe von Nmap und einem speziellen NSE-Skript *http-waf-detect* ist es einfach, eine solches System zu identifizieren:

```
nmap -p80 --script http-waf-detect <ziel>
```

Erkennt das Skript ein Filtersystem, so sieht der entsprechende Hinweis wie folgt aus:

```
PORT   STATE SERVICE
80/tcp open  http
|_http-waf-detect: IDS/IPS/WAF detected
```

Dieses Skript existiert in weiteren Varianten: *http-waf-detect.detectBodyChange* kann Änderungen in dem Response-Body feststellen, *http-waf-detect.aggro* kann Payloads verwenden.

5.1.10 Schwachstellen aufdecken

Web-Applikationen sind für die verschiedensten Angriffe anfällig. Während die Brute-Force-Attacken eher mit der Brechstange an die Sache herangehen, nutzen andere Techniken gezielt Schwachstellen und Konfigurationsmängel in einer Umgebung aus.

Oben hatte ich erwähnt, dass Nmap Cross Site Tracing-Schwachstellen (XST) attackieren kann. Diese werden durch die Existenz von Cross Site Scripting-Schwachstellen verursacht. Ausnutzen lässt sich eine XST-Schwäche mit der TRACE-Methode.

Und so gehen Sie konkret vor:

```
nmap -p80 --script http-methods,http-trace --script-args
http-methods.retest <ziel>
```

Ist auf Seiten des Webservers die TRACE-Methode aktiviert, produziert Nmap in etwa folgende Ausgabe:

```
PORT    STATE SERVICE
80/tcp  open  http
|_http-trace: TRACE is enabled
| http-methods: GET HEAD POST OPTIONS TRACE
| Potentially risky methods: TRACE
| See http://nmap.org/nsedoc/scripts/http-methods.html
| GET / -> HTTP/1.1 200 OK
|
| HEAD / -> HTTP/1.1 200 OK
|
| POST / -> HTTP/1.1 200 OK
|
| OPTIONS / -> HTTP/1.1 200 OK
|
|_TRACE / -> HTTP/1.1 200 OK
```

Ist das nicht der Fall, wird TRACE nicht unter *http-methods* gelistet:

```
PORT   STATE SERVICE
80/tcp open  http
| http-methods: GET HEAD POST OPTIONS
| GET / -> HTTP/1.1 200 OK
|
| HEAD / -> HTTP/1.1 200 OK
|
| POST / -> HTTP/1.1 200 OK
|
|_OPTIONS / -> HTTP/1.1 200 OK
```

Nmap ist auch in der Lage, Cross-Site-Scripting-Schwachstellen in Web-Applikationen aufzudecken:

```
nmap -p80 --script http-unsafe-output-escaping  <ziel>
```

Die Ausgabe führt dann die Dateien auf, die das NSE-Script als unsicher bewertet:

```
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack
| http-unsafe-output-escaping:
|_  Characters [> " '] reflected in parameter id at
http://ziel/1.php?id=1
```

Viele Web-Applikationen kranken an SQL Injection-Schwachstellen. Dabei werden Benutzereingaben nicht wie vorgesehen verarbeitet und erlauben dem Angreifer die Kompromitierung des gesamten Systems. Nmap verfügt mit dem NSE-Skript *http-sql-injection* über ein einfaches, aber wirkungsvolles Werkzeug für die Durchführung von SQL-Injektion-Attacken:

```
nmap -p80 --script http-sql-injection <ziel>
```

Alle womöglich anfälligen Dateien führt Nmap in der Ausgabe auf:

```
PORT   STATE SERVICE
80/tcp open  http    syn-ack
| http-sql-injection:
|   Possible sqli for queries:
|_    http://xxx/index.php?param=13'%20OR%20sqlspider
```

Nmap führt bei der Berichtausgabe einen Abgleich mit den Daten das Fuzzdb-Projekts (*https://github.com/fuzzdb-project/fuzzdb/*) durch. So ist sichergestellt, dass nur relevante Schwachstellen aufgeführt werden.

Schließlich können Sie mit Hilfe von Nmap prüfen, ob Ihre Webserver gegen Slowloris Denial-of-Service-Attacken anfällig ist. Das zugehörige NSE-Skript führen Sie wie folgt aus:

```
nmap -p80 --script http-slowloris --max-parallelism 300
<ziel>
```

Hier ein Beispiel für eine Ausgabe, die auf eine entsprechende DoS-Schwachstelle hinweist:

```
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack
| http-slowloris:
|   Vulnerable:
|   the DoS attack took +5m35s
|   with 300 concurrent connections
|_  and 900 sent queries
```

Es lohnt sich, immer mal auch einen Blick in das NSE-Verzeichnis zu werfen, denn dort finden Sie immer auch Hinweise zu neuen Skript-Varianten und weiteren Ausführungs- und Steuermöglichkeiten.

5.2 Test von Datenbanken

Web-Applikationen speichern unterschiedlichste Informationen in Datenbanken. Aus technischer Sicht betrachtet sind Lösungen wie Joomla!, Magento, WordPress & Co. nichts anderes als Datenbankmanager, die das komfortable Speisen der Ablagen und die Ausgabe der Datenbankdaten ermöglichen.

Die eigentlich interessanten Daten von Content-Managementsystemen, Online-Shops etc. liegen in diesen Datenbanken. Damit ist auch klar, warum sie insbesondere das Ziel von Hacker-Attacken sind. Die meisten Web-Applikationen sind für das Zusammenspiel mit MySQL konzipiert, aber auch NoSQL-Datenbanken werden immer beliebter. Auch MS SQL Server können mit Nmap einer Analyses unterzogen werden. In diesem Abschnitt schauen wir uns an, welche Möglichkeiten Nmap für die Analyse von MySQL-Datenbankservern bietet.

5.2.1 MySQL-Datenbanken abrufen

Auf MySQL-Servern laufen in der Regel nicht nur eine, sondern mehrere Datenbanken. Mit Hilfe von Nmap können Sie herausfinden, welche Ablagen auf einem MySQL-Server bestehen.

Auch hierfür stellt Ihnen Nmap mit *mysql-databases* ein NSE-Skript zur Verfügung. Hier ein Beispiel für seine Ausführung:

```
nmap -p3306 --script mysql-databases --script-args mysqlu-
ser=<benutzername>,mysqlpass=<passwort> <ziel>
```

Eine entsprechende Ausgabe sieht dann wie folgt aus:

```
3306/tcp open  mysql
| mysql-databases:
|   information_schema
|   temp
|   shop
|   ids
|_  crm
```

Die Interpretation des Scan-Ergebnisses ist einfach: Die Ausgabe führt die gefundenen Datenbanken auf.

5.2.2 MySQL-Benutzer auslesen

Wenn Sie eine neue Datenbank in MySQL anlegen, dann wird meist auch ein neuer Datenbankbenutzer angelegt. Wenn Sie außerdem Web-Applikationen wie Magento & Co. einsetzen, benötigen Sie weitere User. Auch die können Sie mit Nmap einfach auslesen:

```
nmap -p3306 --script mysql-users --script-args mysqlu-
ser=<benutzer>,mysqlpass=<passwort> <ziel>
```

Die Liste der Benutzernamen können Sie dem Abschnitt *mysql-users* entnehmen:

```
3306/tcp open  mysql
| mysql-users:
|   root
|   crm
|   web
|_  admin
```

Was, wenn Sie nun keinen Benutzernamen und kein Passwort besitzen, also keine Eingaben für *mysqluser* und *mysqlpass* vornehmen können? In diesem Fall versucht das Skript diese mit zwei weiteren Skripts *mysql-brute* und *mysql-empty-password* herauszufinden. Dazu gleich mehr.

5.2.3 MySQL-Variablen auslesen

MySQL besitzt verschiedenste Umgebungsvariablen, die für unterschiedliche Aufgaben relevant sind. Auch die können mit Hilfe von Nmap abgerufen werden:

```
nmap -p3306 --script mysql-variables --script-args mysqlu-
ser=<benutzername>,mysqlpass=<passwort> <ziel>
```

Das Auslesen der Umgebungsvariablen bringt beispielsweise ein Ergebnis wie das folgende:

```
3306/tcp open  mysql
| mysql-variables:
|   auto_increment_increment: 1
```

```
|   auto_increment_offset: 1
|   automatic_sp_privileges: ON
|   back_log: 50
|   basedir: /usr/
|   binlog_cache_size: 32768
|   bulk_insert_buffer_size: 8388608
|   character_set_client: latin1
|   character_set_connection: latin1
|   character_set_database: latin1
|   .
|   version_comment: (Debian)
|   version_compile_machine: powerpc
|   version_compile_os: debian-linux-gnu
|_  wait_timeout: 28800
```

Sollten Sie kein brauchbares Ergebnis erhalten, liegt das womöglich an einem Paketfilter. Auch der Port kann falsch gewählt sein. In diesem Fall geben Sie mit dem Argument *-p* den korrekten Port an.

5.2.4 Root-Account finden

Neue MySQL-Datenbanken besitzen oftmals eine Root-Account ohne Passwort. Das ist für Penetration Tester ein gefundenes Fressen. Um herauszufinden, ob auch Ihr Datenbankserver einen Root-Account mit leerem Passwort besitzt, verwenden Sie das Skript *mysql-empty-password*:

```
nmap -p3306 --script mysql-empty-password <ziel>
```

Eine entsprechende Ausgabe bestätigt dies:

```
3306/tcp open  mysql
| mysql-empty-password:
|_  root account has empty password
```

5.2.5 Brute-Force-Attacke gegen MySQL

Webserver geben gelegentlich Datenbankfehler zurück, denen man den MySQL-Benutzernamen der Web-Applikation entnehmen kann. Potenzielle Angreifer können diese Informationen für eine Brute-Force-Attacke nutzen. Um eine Brute-Force-Attacke gegen einen MySQL-Server auszuführen, verwenden Sie folgenden Befehl:

```
nmap -p3306 --script mysql-brute <ziel>
```

Kann das Skript gültige Zugangsdaten ermitteln, so werden diese in der Skript-Ausgabe aufgeführt:

```
3306/tcp open  mysql
| mysql-brute:
|   root:<leer> => Valid credentials
|_  test:test => Valid credentials
```

Ein solcher Test muss nicht zwangsläufig erfolgreich sein, sondern kann auch ein negatives Ergebnis liefern:

```
3306/tcp open  mysql
| mysql-brute:
|   Accounts: No valid accounts found
|_  Statistics: Performed 50009 guesses in 102 seconds, ave-
rage tps: 477
```

5.2.6 Unsichere MySQL-Konfigurationen

Eine letzte MySQL-spezifische Testfunktion von Nmap möchte ich Ihnen noch vorstellen. Zu Ihrer Nmap-Installation gehört ein kleines Skript, dass unsichere Konfigurationen in einer MySQL-Installation aufdecken kann. Nmap macht sich dabei die Benchmarks des Center for Internet Security, kurz CIS (*https://www.cisecurity.org*) zunutze. Das Skript *mysql-audit* führen Sie wie folgt aus:

```
nmap -p3306 --script mysql-audit --script-args 'mysql-
audit.username="<benutzer>",mysql-audit.password="<pass-
wort>",mysql-audit.filename=/usr/local/share/nmap/nse-
lib/data/mysql-cis.audit' <ziel>
```

Zu jeder Prüfung wird einer der drei folgenden Status ausgegeben: *PASS*, *FAIL* oder *REVIEW*. Hier ein Beispiel für eine entsprechende Ausgabe:

```
PORT     STATE SERVICE
3306/tcp open  mysql
| mysql-audit:
|   CIS MySQL Benchmarks v1.0.2
|       3.1: Skip symbolic links => PASS
|       3.2: Logs not on system partition => PASS
|       3.2: Logs not on database partition => PASS
|       4.1: Supported version of MySQL => REVIEW
|         Version: 5.x
|       4.4: Remove test database => PASS
|       4.5: Change admin account name => FAIL
|       4.7: Verify Secure Password Hashes => PASS
|       4.9: Wildcards in user hostname => PASS
|       4.10: No blank passwords => PASS
|       4.11: Anonymous account => PASS
|       5.1: Access to mysql database => REVIEW
|         Verify the following users that have access to the
MySQL  database
|           user              host
|           root              localhost
|       5.2: Do not grant FILE privileges to non Admin users
=> PASS
|       5.3: Do not grant PROCESS privileges to non Admin
users =>  PASS
|       5.4: Do not grant SUPER privileges to non Admin users
=> PASS
|       5.5: Do not grant SHUTDOWN privileges to non Admin
users =>  PASS
|       5.6: Do not grant CREATE USER privileges to non Admin
users  => PASS
```

```
|       5.7: Do not grant RELOAD privileges to non Admin
users =>  PASS
|       5.8: Do not grant GRANT privileges to non Admin users
=> PASS
|       6.2: Disable Load data local => FAIL
|       6.3: Disable old password hashing => PASS
|       6.4: Safe show database => FAIL
|       6.5: Secure auth => FAIL
|       6.6: Grant tables => FAIL
|       6.7: Skip merge => FAIL
|       6.8: Skip networking => FAIL
|       6.9: Safe user create => FAIL
|       6.10: Skip symbolic links => FAIL
|
|_      The audit was performed using the db-account: root
```

5.3 Mailserver im Visier

Einen Mailserver findet man in nahezu jeder IT-Infrastruktur. Aufgrund der Bedeutung der elektronischen Post sind E-Mail-Server natürlich ein sehr attraktives Ziel. In der Vergangenheit haben die E-Mail-Protokolle SMTP, POP3 und IMAP4 und deren Implementierungen immer wieder erhebliche Schwachstellen offenbart. Die sollten Sie beim Penetration Testing ebenfalls unter die Lupe nehmen.

5.3.1 E-Mail-Accounts aufdecken

Das Aufdecken von gültigen E-Mail-Accounts ist beim Penetration Testing eine zentrale Aufgabe. Für Hacker sind natürlich die meist hoch sensiblen Daten in den Postfächern interessant. Um die Sicherheit von E-Mail-Diensten zu prüfen, müssen Sie zunächst die E-Mail-Accounts auslesen.

Dazu bedienen Sie sich eines NSE-Skripts, das nicht zur Standardinstallation von Nmap gehört. Das finden Sie unter folgender URL:

```
http-google-search.nse from http://seclists.org/nmap-
dev/2011/q3/att-401/http-google-email.nse
```

Kopieren Sie die Datei in das Skript-Verzeichnis und führen Sie als Nächtes ein Update der NSE-Skriptdatenbank aus:

```
nmap --script-updatedb
```

Nmap gibt folgende Ausgabe aus:

```
SE: Updating rule database.
NSE: Script Database updated successfully.
```

Damit ist das Skript einsatzbereit. Das Skript verwendet die Google-Suche und –Gruppen, um nach gültigen Accounts zu fahnden. Sie führen das Skript wie folgt aus:

```
nmap -p80 --script http-google-email <ziel>
```

Eine typische Ausgabe sieht wie folgt aus und führt die gefundenen E-Mail-Accounts des Zielsystems auf:

```
$ nmap -p80 --script http-google-email mail.beispiel.de
PORT   STATE SERVICE
80/tcp open  http
| http-google-email:
| info@beispiel.de
|_admin@beispiel.de
```

Um die Ergebnisse auf einen Hostnamen zu beschränken, verwenden Sie das Argument *http-google-email.domain*.

5.3.2 Offene Relays aufspüren

Offene Relays sind für Hacker und Spammer ein willkommenes Ziel, weil sie ohne eine Authentifizierung von Dritten genutzt werden können. Zum Penetration Testing gehört daher auch das Prüfen einer Mailserver-Umgebung auf offene Relays.

Die Verwendung des entsprechenden NSE-Skripts ist wieder einfach:

```
nmap -sV --script smtp-open-relay -v <ziel>
```

Eine Ausgabe könnte wie folgt aussehen und auf einen offenen Relay hinweisen:

```
Host script results:
| smtp-open-relay: Server is an open relay (1/16 tests)
|_MAIL FROM:<info@server.de> -> RCPT
TO:<relaytest@beispiel.de>
```

5.3.3 SMTP-Passwort knacken

Nmap erlaubt Ihnen mit einem einfachen Skript das Knacken des Passwortschutzes eines SMTP-Servers. Dazu führt das Programm eine Wörterbuchattacke aus:

```
nmap -p25 --script smtp-brute <ziel>
```

Das Ergebnis einer erfolgreichen Attacke sieht dann beispielsweise wie folgt aus:

```
PORT    STATE SERVICE REASON
25/tcp  open  stmp    syn-ack
| smtp-brute:
|   Accounts
|     user1:passwort1 - Valid credentials
|     user2:passwort2 - Valid credentials
|     user3:passwort3 - Valid credentials
|     user4:passwort4 - Valid credentials
|   Statistics
|_    Performed 5432 guesses in 90 seconds, average tps: 60
```

5.3.4 SMTP-User auslesen

Da E-Mail-Accounts gerne auch in Web-Applikationen verwendet werden, sind sie ebenfalls für Hacker sehr interessant. Mit Kenntnis eines Benutzernamens kann man sich dann an weitere Attacken machen, und zwar nicht nur auf E-Mail-Servern, sondern eben auch auf Web-Anwendungen. In Nmap ist ein NSE-Skript integriert, mit dem Sie die Benutzer von SMTP-Servern auslesen können:

```
nmap -p25 –script smtp-enum-users <ziel>
```

Die Ausgabe sieht dann beispielsweise wie folgt aus:

```
Host script results:
| smtp-enum-users:
|_  RCPT, Admin
```

5.3.5 POP3-Server attackieren

POP3-Mailserver bestimmen nach wie vor das Bild. Und da auf Ihnen meist wichtige E-Mails lagern, sind sie für jeden Hacker ein interessantes Ziel. Auch das können Sie mit Nmap versuchen:

```
nmap -p110 --script pop3-brute <ziel>
```

Eine entsprechende Ausgabe sieht dann wie folgt aus:

```
PORT    STATE SERVICE
110/tcp open  pop3
| pop3-brute: postmaster : geheim
```

Wenn Sie mehr über dem POP3-Server und die aktivierten Funktionen herausfinden wollen, gelingt das mit folgendem Befehl:

```
nmap -p110 --script pop3-capabilities <ziel>
```

Der Ausgabe können Sie je nach Testverlauf und Umgebung beispielsweise folgende Fähigkeiten entnehmen: *USER CAPA UIDL TOP OK(K) RESP-CODES PIPELINING STLS SASL(PLAIN LOGIN)*

5.3.6 IMAP-Server attackieren

Was bei einem POP3-Server funktioniert, können Sie auch an einem IMAP-Server testen: das Erlangen eines Zugangs mit Hilfe einer Brute-Force-Attacke. Der entsprechende Befehl lautet wie folgt:

```
nmap -p143 --script imap-brute <ziel>
```

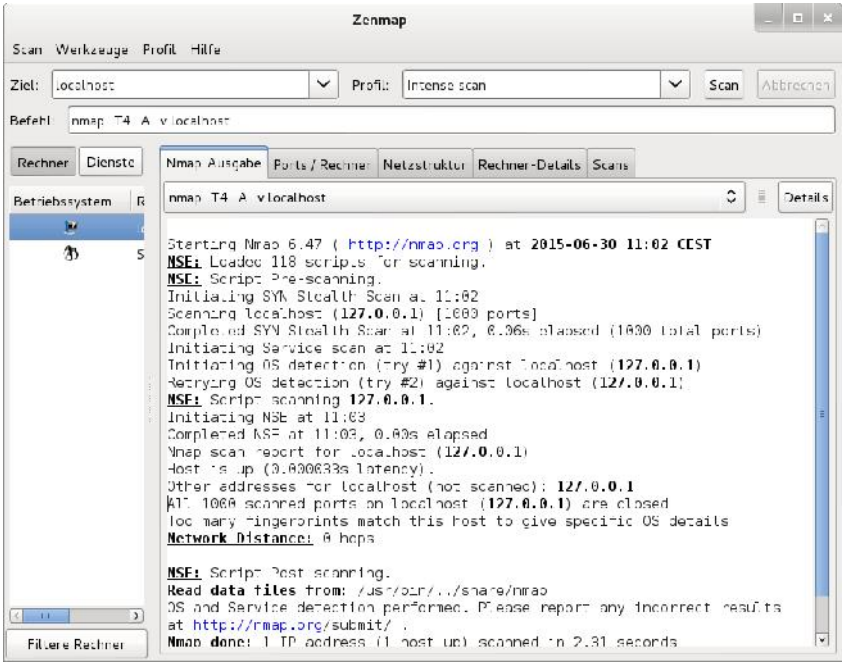
Die Ausgabe des NSE-Skripts verhilft Ihnen im Idealfall zu gültigen Zugangsdaten:

```
PORT    STATE SERVICE REASON
143/tcp open  imap    syn-ack
| imap-brute:
|   Accounts
|     postmaster:geheim - Valid credentials
|     admin:geheimer - Valid credentials
|   Statistics
|_    Performed 1245 guesses in 420 seconds, average tps: 3
```

In diesem Kapitel haben Sie verschiedene Möglichkeiten kennengelernt, wie Sie einen Mailserver auf Schwachstellen prüfen können. Im Skript-Verzeichnis der Nmap-Website finden Sie weitere Informationen und Testmöglichkeiten.

6 Mit Zenmap arbeiten

Bei Zenmap handelt es sich um die offizielle GUI für den Nmap Security Scanner. Die GUI ist wie Nmap Open Source und erlaubt die komfortable Steuerung von Nmap. Über die GUI können Sie die Nmap-spezifischen Befehle ausführen und die Ergebnisse prüfen. Die Ergebnisse der Sicherheitschecks können gesichert und miteinander verglichen werden. Zenmap speichert die Ergebnisse vorheriger Scans in einer durchsuchbaren Datenbank.



Ein erster Scan mit Zenmap. Dem Ergebnis können Sie entnehmen, dass beim Scannen der lokalen Kali Linux-Installation alle Ports geschlossen sind.

Administratoren schwören bekanntlich auf die Verwendung der Konsole, aber für das schnelle und effektive Ausführen von Konsolenprogrammen wie Nmap sind

GUIs wie Zenmap eine große Hilfe. Gerade auch Einsteiger profitieren von Ihnen, weil die Kommandoeingabe weniger fehleranfällig ist. Zwar deckt Zenmap nicht die gesamte Funktionalität von Nmap ab, aber für einen schnellen und zuverlässigen Portscan-Vorgang gibt es kaum eine Alternative.

Der Aufruf von Zenmap kann auf der Konsole mit dem Befehl *zenmap* oder aber mit dem Menübefehl *Anwendungen > Kali Linux > Schwachstellenanalyse > Verschiedene Scanner > Zenmap* erfolgen.

Zenmap kann nicht nur die Ergebnisse der Portscans ausgeben, sondern bietet Ihnen Zusammenfassungen zu einem Host oder allen Host, auf denen ein bestimmter Service ausgeführt wird. Zenmap kann mit den gesammelten Informationen sogar eine Karte der Netzwerktopologie anlegen. Sie können sogar die Ergebnisse von mehreren Scans kombinieren und diese zusammen darstellen.

Eine weitere Besonderheit hatte ich schon angedeutet: Zenmap kann zwei Scans miteinander vergleichen und die Unterschiede herausarbeiten. So können Sie beispielsweise zwei Tests mit unterschiedlichen Scan-Optionen fahren und die Ergebnisse miteinander vergleichen. Auf diesem Weg können Sie einfach neue Hosts und/oder Services im Netzwerk identifizieren. Auf diesem Weg können Sie auch Dienste ermitteln, die nicht mehr verfügbar sind, weil sie ausgefallen sind oder heruntergefahren wurden.

Zenmap speichert die Ergebnisse eines Scan-Vorgangs solange, bis Sie sich dazu entscheiden, diese zu löschen oder anderweitig zu verarbeiten. Und: Sie können einen Scan so oft Sie wünschen wiederholen.

Nmap unterstützt Hundert Optionen und Parameter, die Sie für Ihre Scans verwenden können. Doch gerade für Einsteiger ist die Vielzahl irritierend. Die Steuerung auf der Konsole ist außerdem fehleranfällig. Zenmap nimmt Ihnen hierbei viel Arbeit ab, da Sie verschiedene Einstellungen über Kontrollkästchen, Eingabefelder und Auswahlmenü steuern können.

6.1 Scannen und auswerten

Nach dem Starten der Zenmap ist es einfach, einen ersten Scan durchzuführen. Zenmap präsentiert Ihnen seine eingedeutschte Benutzerschnittstelle. Das vereinfacht die Handhabung zusätzlich. Im Hauptfenster geben Sie in das Eingabefeld *Ziel* den Hostnamen oder die IP-Adresse des Zielsystems an. Für erste Gehversuche verwenden Sie den vom Nmap-Team bereitgestellten Server *scanme.nmap.org*. Im Auswahlmenü *Profil* bestimmen Sie das Scan-Profil, das die Intensität und die Scan-Variante bestimmt. Standardmäßig ist mit *Intense Scan* ein umfangreicher Scan-Vorgang vorgesehen.

Um den eigentlichen Scan zu starten, klicken Sie auf die Schaltfläche *Scan*. In der Nmap-Ausgabe können Sie anhand der Ausgabe erkennen, welche Informationen der Port-Scanner einlesen konnte.

In das *Ziel*-Feld können Sie nicht nur einzelne Ziele, sondern auch mehrere oder sogar ganze Subnetze eingeben. Wenn Sie mehrere Hosts scannen, geben Sie deren IP-Adresse oder Hostnamen getrennt durch ein Leerzeichen an. Sie können auch andere Angaben verwenden. Hier zwei Beispiele:

```
192.168.2.0/24
```

```
10.0.0-5.*
```

Alle durchgeführten Scans sind über das Auswahlmenü rechts der Zieleingabe verfügbar.

Zenmap stellt Ihnen über das Auswahlmenü *Profil* verschiedene vordefinierte Scan-Profile zur Verfügung. Zenmap kennt zehn vordefinierte Profile für unterschiedliche Aufgaben. Das Programm erlaubt Ihnen über das Menü *Profil* das Anlegen neuer und das Bearbeiten bestehender Profile. Wir kommen später darauf zu sprechen.

Eine weitere Besonderheit von Zenmap nennt sich Scan Aggregation. Sie können mit dieser Funktion die Testergebnisse mehrerer Scan-Vorgänge kombinieren. Aus einer Sammlung an Scans können Sie ein regelrechtes Netzwerkinventar erstellen.

Anhand eines Beispiels wird deutlicher, wie das in der Praxis funktioniert und welche Vorteile das bringt. Führen Sie zunächst den Quick Scan an folgendem Host durch:

```
scanme.nmap.org.
```

Dann führen Sie das gleiche Scan-Profil gegen *localhost* aus. Die durchgeführten Scan-Konfigurationen sind über das Auswahlmenü der Registerkarte *Nmap-Ausgabe* verfügbar. Rechner, deren Betriebssystem Nmap identifizieren kann, erweitert Zenmap in der Rechnerliste, um ein entsprechendes OS-Icon. Handelt es sich bei dem Ziel um einen Linux-Rechner, wird dieser mit einem Pinguin gekennzeichnet.

Wenn Sie nun mehr über *scanme.nmap.org* erfahren wollen, so führen Sie anstelle der schnellen Scan (Quick scan) einen intensiven (Intense scan) durch. Wenn Sie nun eine Nmap-Instanz ausführen, so stehen Ihnen dabei die Ergebnisse zweier Scan-Vorgänge an zwei unterschiedlichen Hosts nicht zur Verfügung.

Zenmap sammelt die Ergebnisse und stellt sie Ihnen über die Registerkarte *Scans* zur Verfügung. Sie können übrigens einfach mit der Tastenkombination *Strg + N* weitere Zenmap-Fenster öffnen und mit diesen unterschiedliche Scans und Ziele prüfen. Zenmap erzeugt während und nach dem Scan-Vorgang seine Ausgabe, die auf der Registerkarte *Nmap-Ausgabe* zu finden ist.

Eine typische Scan-Ausgabe.

Bis auf die Syntax-Hervorhebung, die Inhalte farbig absetzt oder fett formatiert, bietet die Nmap-Ausgabe gegenüber der Konsolenausgabe keinen nennenswerten Mehrwert. Aber die anderen Registerkarten bieten Ihnen eine Fülle von interessanten Informationen, die Sie bei der Interpretation der Ergebnisse unterstützen. Neben der Ausgabe stehen Ihnen vier weitere Register zur Verfügung:

- Ports/Rechner
- Netzstruktur
- Rechner-Details
- Scans

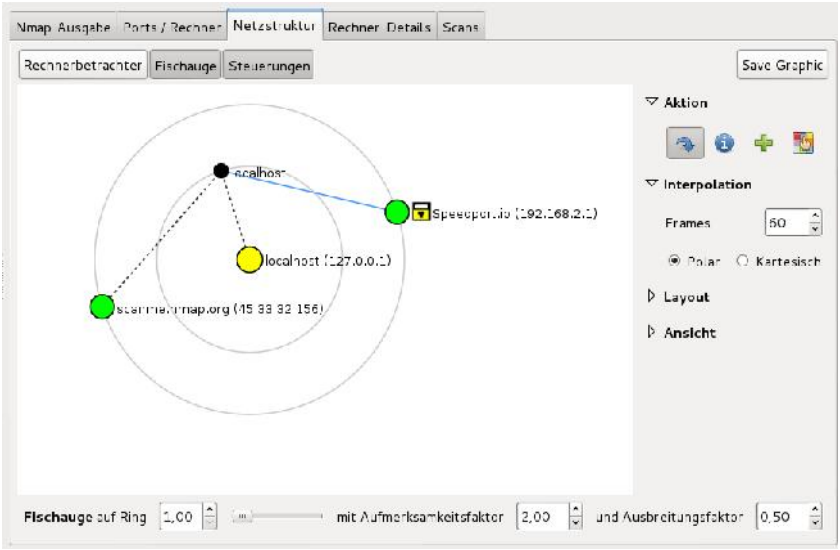
Die Syntaxhervorhebung kann übrigens in der Zenmap-Konfigurationsdatei *zenmap.conf* angepasst werden.

Auf der Registerkarte *Nmap-Ausgabe* finden Sie rechts neben der Auswahl der Scan-Vorgänge die Schaltfläche *Details*. Dem zugehörigen Dialog können Sie verschiedene Informationen über den Scan-Vorgang entnehmen. Neben dem konkreten Befehl und der Nmap-Version können Sie dem Dialog allgemeine Daten über den Zeitpunkt des Scans und den Scan-Typ entnehmen.

Die Details des Scan-Vorgangs.

Welche Informationen auf der Registerkarte *Ports/Rechner* angezeigt werden, ist davon abhängig, ob Sie links die Optionen *Rechner* oder *Dienste* aktiviert haben. Wenn Sie die Rechner ausgewählt haben, werden rechts die verfügbaren Ports auf diesem Host angezeigt. In den tabellarischen Übersichten können Sie übrigens wunderbar die Einträge mit einem Klick auf die Kopfzeile ändern.

Haben Sie *Dienste* aktiviert, werden unterhalb die verwendeten Dienste aufgeführt. Mit einem Klick auf einen Dienst-Eintrag können Sie dann rechts die Rechner einblenden, die einen bestimmten Dienst verwenden.



Die Darstellung der Netzwerkstruktur.

Ein echtes Highlight von Zenmap ist die visuelle Darstellung der Netzwerkstruktur. Dabei werden die Hosts als konzentrische Kreise dargestellt, wobei jeder Kreis einen Netzwerk-Hop, also einen Übergang zwischen zwei Netzwerken darstellt. Mit einem Klick auf einen Rechnereintrag zentrieren Sie diesen.

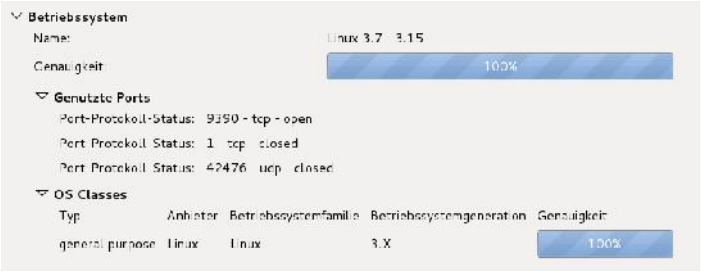
Die Visualisierung der Netzwerkstruktur profitiert dabei von der Verwendung der Nmap-Option *--traceroute*. Sie können damit auch verfolgen, wie der Traffic im Netzwerk läuft.

Auf der Registerkarte *Details* können Sie die nach Kategorien sortierten Informationen über ein System abrufen. Die Kategorien:

- **Rechnerstatus**: Hier finden Sie den aktuellen Status (up/down), die Anzahl der gescannten und der offenen Ports. Zu jedem Rechner wird ein Icon eingeblendet, das die Verwundbarkeit einschätzt. Diese Einschätzung basiert auf der Anzahl der offenen Ports. Nachstehender Abbildung fasst Sie die Symbole und die Anzahl der offenen Ports zusammen:

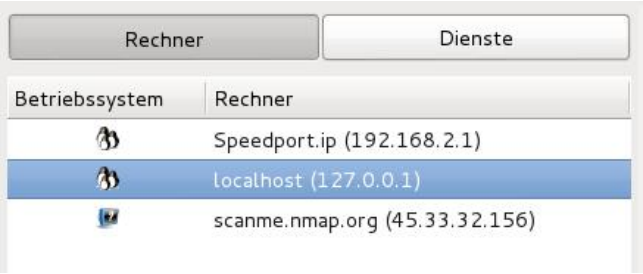
Symbol	Anzahl der offenen Ports
	0 bis 2
	3 bis 4
	5 bis 6
	7 bis 8
	9 und mehr

- **Adressen**: Hier werden die IPv4-, IPv6 und die MAC-Adressen des Rechners aufgeführt.
- **Rechnernamen**: Diesem Bereich können Sie womöglich erkannte Rechnernamen entnehmen.
- **Betriebssystem**: Kann Nmap auch das Betriebssystem identifizieren, finden Sie hier auch die Kategorie *Betriebssystem*, der Sie das verwendete System sowie weitere Informationen entnehmen können.
- **Kommentare**: Den Abschluss bilden die Kommentare.



Die Betriebssystemdetails eines gescannten Rechners.

Die letzte Registerkarte trägt die Bezeichnung *Scans*. Hier finden Sie alle durchgeführten Scans. Sie können bereits gespeicherte Scan-Konfigurationen importieren und nicht mehr benötigte löschen. Aktuell ausgeführte Scans können Sie auch abbrechen.



Die Rechner- und Diensteliste.

Links neben der Nmap-Ausgabe finden Sie die Rechner- und Diensteliste. Mit einem Klick auf *Rechner* wird die Liste der gescannten Rechner eingeblendet. In voranstehender Abbildung sind es gerade einmal drei Stück, doch im Admin-Alltag werden es schnell Dutzende oder gar Hunderte. Die Darstellung auf der rechten Registerkarte *Ports/Rechner* ist mit der Markierung der Dienste bzw. Rechner verknüpft.

Zu Rechnern werden das Betriebssystem und der Rechnernamen sowie die IP-Adresse angezeigt. Über die Kopfzeile können Sie auch die Sortierung ändern. Abhängig vom jeweiligen Betriebssystem wird auch hier wieder ein Icon angezeigt:

Symbol	Betriebssystem
	FreeBSD
	Irix
	Linux
	Mac OS X

Symbol	Betriebssystem
	OpenBSD
	Red Hat Linux
	Solaris or OpenSolaris
	Ubuntu Linux
	Windows
	Anderes
	Keine OS-Erkennung durchgeführt

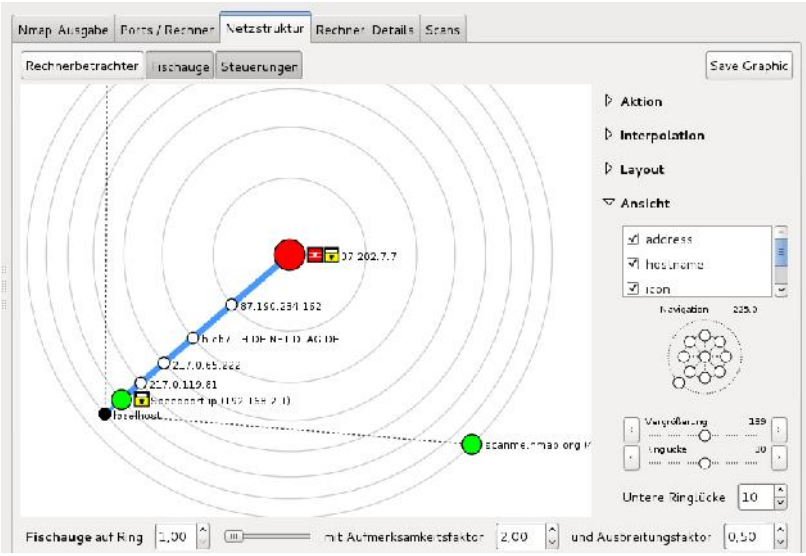
Auf der Registerkarte *Dienste* finden Sie die Dienste, deren Ausführung Nmap auf den Zielsystemen identifiziert hat. Wie wir oben gesehen haben, werden auf der Registerkarte *Scans* die durchgeführten Scans aufgeführt. Diese besitzt nach einem ersten Durchgang den Zusatz *Ungespeichert*, der in der Spalte *Status* gelistet wird.

Das können Sie ändern, indem Sie über das Menü *Scan* > *Speichere Scan* Ihre Untersuchungsergebnisse für eine spätere Verwendung sichern. Die Daten werden in einem Nmap-spezifischen XML-Format gespeichert. Nach dem Speichern verschwindet der Zusatz *Ungespeichert* in der Scan-Übersicht.

Solange Sie Ihre Scans nicht im Nmap-Format sichern, verbleiben diese in der Zenmap-Datenbank *zenmap.db*. Dort sind Sie dann auch mit anderen Werkzeugen wie Datenbank-Viewern einsehbar. Die Einträge verbleiben standardmäßig 60 Tage in der Datenbank bis sie dann gelöscht werden. Das können Sie allerdings auch durch Eingriffe in die Zenmap-Konfigurationsdatei *zenmap.conf* ändern.

6.2 Netzwerktopologien

Die Registerkarte *Netzstruktur* stellt Ihnen eine interaktive, animierte Visualisierung der Verbindungen zwischen den Hosts zur Verfügung. Alle Hosts werden als Punkt auf dem Kreis dargestellt. Sie können mit gedrückter Maustaste die Position der Darstellung verschieben. Rechts finden Sie verschiedene Funktionen für die Anpassung der Darstellung. Unter *Ansicht* finden Sie einen Schieberegler, mit dem Sie die Visualisierung vergrößern und verkleinern können.



Die Ansicht der Netzwerkstruktur.

Wenn Sie sich für einen speziellen Host interessieren, klicken Sie auf diesen, damit er ins Zentrum der Darstellung gerückt wird. Die Grafik passt sich automatisch an. Bei einem erneuten Scan werden neue Hosts und neue Dienste automatisch der Darstellung hinzugefügt.

Die Nmap-Option *--traceroute* sorgt dafür, dass die Netzwerkpfade dargestellt werden können. Die Netzwerkstruktur wird beim Zugriff auf die Visualisierung mit localhost als Mittelpunkt generiert. Im Hintergrund sorgt eine angepasste Version des Programms RadialNet von João Paulo S. Medeiros für die grafische Aufbereitung.

In einer Strukturdarstellung kommen viele Symbole und Farbkonventionen zum Einsatz. Die sollten Sie kennen, damit Sie die Darstellungen interpretieren können.

Ein regulärer Host wird in der Visualisierung als kleiner Kreis darstellt. Die Farben und Kreisgröße werden durch die Anzahl der offenen Ports eines Hosts bestimmt. Je mehr offene Ports ein Host aufweist, umso größer ist seine Darstellung. Ein weißer Kreis repräsentiert einen Host, bei dem kein Portscan erfolgte. Hosts mit weniger als drei offenen Ports werden grün markiert. Bei Hosts mit 3 bis 6 offenen Ports kommt gelb zum Einsatz. Alle jene Hosts, bei denen mehr als sechs offene Ports gefunden werden, kennzeichnet Zenmap rot.

Agiert ein Host als Router, Switch oder WLAN-Access Point, so wird er als Quadrat dargestellt. Distanzen werden als graue konzentrische Kreise visualisiert. Außerdem symbolisiert jeder Ring einen Netzwerk-Hop.

Die Verbindungen zwischen den Hosts werden als bunte Linien dargestellt. Einfache Traceroute-Verbindungen werden als blaue Linien gekennzeichnet, alternative Pfade orange. Auch die Dicke der Verbindung gibt Ihnen Aufschluss über die Verbindung: Die Dicke ist proportional zur Round-Trip-Zeit (RTT). Hosts mit einem höheren RTT-Wert besitzen eine dickere Linie. Hosts ohne Traceroute-Informationen werden mit einer schwarz gepunkten Linie miteinander verbunden. Eine gestrichelte blaue Linie zeigt einen Hop ohne RTT-Wert an.

Hosts mit besonderen Aufgaben werden um ein zusätzliches Symbol erweitert, die deren Funktion deutlich machen. Das gilt insbesondere für folgende Netzwerkkomponenten:

-  – Router
-  – Switch
-  – WLAN Access Points
-  – Firewall
-  – Hosts mit gefilterten Ports

Rechts neben der Visualisierung finden Sie die Funktionsleiste, die Ihnen die Durchführung verschiedener Aktionen und Darstellungsanpassungen erlaubt. Die Funktionen sind in vier Funktionsgruppen zusammengefasst.

Die erste Gruppe trägt die Bezeichnung *Aktion* und stellt Ihnen vier Funktionen zur Verfügung. Diese Funktionen werden beim Klicken auf einen Host-Eintrag ausgeführt. Standardmäßig ist die linke Funktion aktiviert, die die Bezeichnung *Change* focus trägt. Sie sorgt dafür, dass beim Klick auf einen Host-Eintrag die Darstellung auf diesen neu ausgerichtet wird.

Die Steuerfunktionen für die Netzstrukturansicht.

Wenn Sie alternativ das *Info*-Symbol aktivieren, werden die Host-Detailinformationen eingeblendet, die Ihnen eine Fülle an weiteren Host-spezifischen Daten bieten. Dabei handelt es sich um sogenannte Rechnerbetrachter. Ein Klick auf das grün hinterlegte Pluszeichen sorgt für die Gruppierung von Kind-Elementen. Eine Gruppierung wird durch einen blauen Doppelkreis gekennzeichnet. Mit der letzten Funktion bestimmen Sie die Farbfüllung. Mit den Interpolationseinstellungen bestimmen Sie, wie schnell die Animation bei Änderungen der Grafik vorgenommen wird.

Zenmap weist den Knoten ein automatisches Layout zu. Sie haben mit dem Bereich die Wahl zwischen zwei Optionen: *Gewichtet* und *Symmetrisch*. Die symmetrische Darstellung weist jedem Knoten den gleich großen Bereich zu. Die gewichtete Variante weist Hosts mit mehr Kind-Elementen mehr Platz auf dem Darstellungsbereich zu. Bei dieser Variante erkennen Sie schneller, ob sich hinter einem Knoten noch weitere Netzwerksegmente befinden.

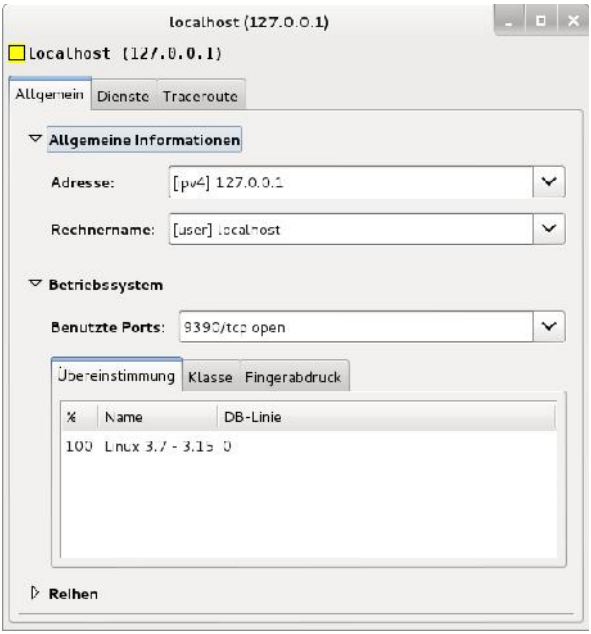
Der Bereich *Ansicht* stellt Ihnen umfangreiche Funktionen für die Anpassung der Ansicht zur Verfügung. Über das Listenfeld können Sie zunächst bestimmen, welche Informationen auf der Karte darstellt werden. Sie können folgende Informationen ein- und ausblenden:

- Adresse
- Hostname

- Symbol
- Latenz
- Ring
- Region

Als Nächstes folgt eine rosettenähnliche Form, mit deren Hilfe Sie die Darstellung verschieben und rotieren können. Ein Klick auf einen Miniaturkreis verschiebt die Darstellung. Mit gedrückter Maustaste auf den äußeren Ring, können Sie die gesamte Darstellung rotieren. Es folgen zwei Schieberegler, mit denen Sie die Vergrößerung und die Ringlücke anpassen können.

Sie können diese Funktionen übrigens einfach ausblenden, indem Sie oberhalb der Netzstruktur auf die Schaltfläche *Steuerungen* klicken. Rechts finden Sie eine weitere praktische Funktion. Mit einem Klick auf *Save Graphic* können Sie die Darstellung als Grafikdatei sichern und dann beispielsweise in Berichten weiter verarbeiten oder an Dritte weitergeben.



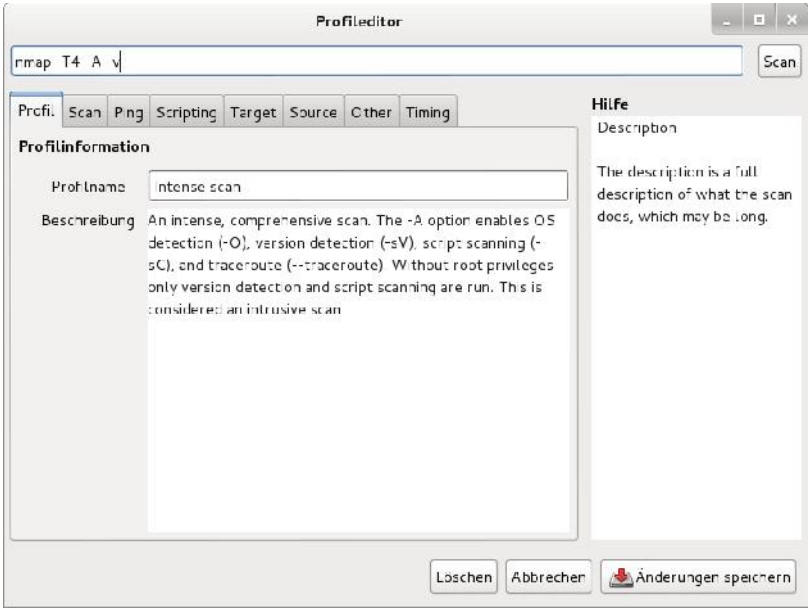
Der Rechnerbetrachter bietet zu jedem Host umfangreiche Detailinfos.

Oberhalb der Grafik finden Sie die Schaltfläche *Rechnerbetrachter*. Dahinter verbirgt sich ein weiterer Dialog, der Ihnen zu allen gescannten Hosts mehr oder minder viele Detailinformationen liefert. Welche Informationen das im Detail sind, ist abhängig davon, was Nmap über ein gescanntes System in Erfahrung bringen kann.

Die Informationen sind auf drei Registerkarten verteilt:

- **Allgemein**: Führt die Adresse und den Rechnernamen sowie betriebssystemspezifische Informationen auf.
- **Dienste**: Auf dieser Registerkarte werden die Dienste und offene Ports aufgeführt.
- **Traceroute**: Führt die Ergebnisse des Traceroute-Befehls auf.

Diese Informationen kombiniert mit den Ergebnissen der Sicherheitscans liefern Ihnen in der Regel bereits ausreichend Informationen über potenzielle Angriffspunkte.



Mit dem Profileditor können Sie Scan-Profile einsehen und bearbeiten.

6.3 Der Profileditor

Zu Beginn dieses Kapitels haben Sie die Scan-Profile kennengelernt. Dabei handelt es sich um vordefinierte Scan-Konfigurationen, die bereits typische Standardaufgaben abdecken. Zenmap stellt Ihren über das Menü *Profil* den sogenannten Profileditor zur Verfügung, mit dem Sie die Profile einsehen und bearbeiten sowie neue anlegen können.

Mit dem Menübefehl *Profil > Neues Profil oder Befehl* legen Sie eine neue Profilkonfiguration an. Alternativ verwenden Sie die Tastenkombination *Strg + P*. Der Profileditor erlaubt auch das Bearbeiten eines ausgewählten Profils. Am einfachsten verwenden Sie hierfür die Tastenkombination *Strg + E*.

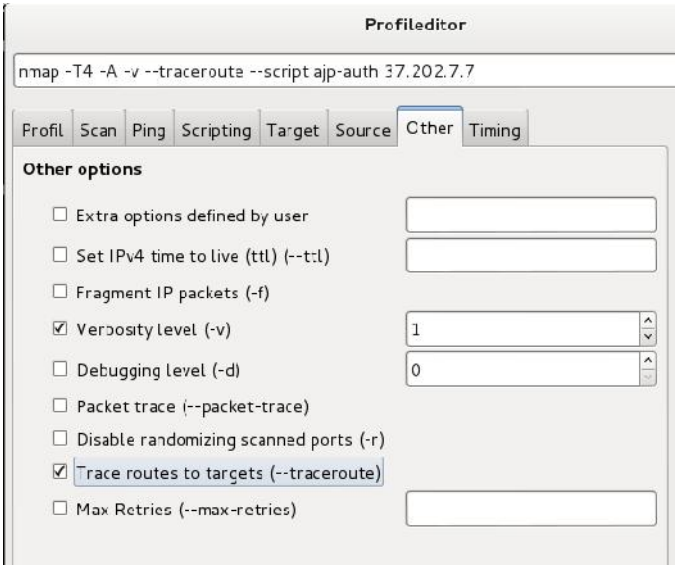
Der Profileditor zeigt im Kopfbereich den editierten Nmap-Befehl an bzw. erlaubt Ihnen das Anlegen eines eigenen Scan-Befehls. Auf der Registerkarte *Profil* weisen Sie einer Scan-Konfiguration eine aussagekräftige Bezeichnung zu.

Sehr umfangreich fallen die Anpassungsmöglichkeiten auf der Registerkarte *Scan* aus. Hier können Sie optional einen oder mehrere Hosts anlegen sowie verschiedene Scan-Optionen wie die Erkennung des Betriebssystems nutzen.

Auch für die Ping-Verwendung stehen Ihnen umfangreiche Anpassungs- und Konfigurationen zur Verfügung, beispielsweise, ob das Pingen vor dem eigentlichen Port-Scan erfolgen soll. Zenmap kann bei der Profilbearbeitung auch auf vordefinierte Skript-Parameter und bereits angelegte Skripts zurückgreifen. Dazu verwenden Sie die Funktionen der Registerkarte *Scripting*.

Mit Hilfe der Registerkarte *Target* können Sie gezielt einzelne oder auch mehrere Hosts vom Port-Scannen ausschließen. Sie können dabei einzelne Hosts in das Eingabefeld eingeben oder alternativ auch Host-Listen verwenden.

Auf der Registerkarte *Source* finden Sie nicht minder interessante Funktionen. Hier können Sie beispielsweise die IP-Adresse des Scanners manipulieren. Wenn Sie sich auch für den Verlauf der Daten zwischen Quelle und Ziel interessieren, können Sie auf der Registerkarte *Other* die Traceroute-Option aktivieren.



Die erweiterten Scan-Optionen.

Wenn Sie intensiven Gebrauch von Nmap machen und viele Hosts prüfen wollen, sind die Einstellungen der Registerkarte *Timing* für Sie interessant. Hier können Sie beispielsweise die maximale Dauer für das Scannen eines Hosts anlegen. Für die Ausführungsgeschwindigkeit sind außerdem die Anpassung der Delay-Zeiten und die parallele Ausführung von Tests relevant.

Das Schöne an den Einstellungen des Profileditors: Hinter jeder Option werden die zugehörigen Nmp-Parameter aufgeführt. Beim Aktivieren einer Option werden diese Parameter automatisch in der Kopfzeile eingefügt. So lernen Sie ganz nebenbei auch noch die zugrundeliegenden Schalter kennen, die der Portscanner bietet.

Wenn Sie bestimme Profile nicht mehr benötigen, öffnen Sie einfach den Bearbeitungsdialog und betätigen die *Löschen*-Schaltfläche.

6.4 Erweiterte Zenmap-Funktionen

Zenmap macht die Steuerung von Nmap dank der übersichtlich gestalteten GUI fast zum Kinderspiel. Neben den Grundfunktionen hat Zenmap einige weitere interessante Funktionen zu bieten, die man bei Nmap in dieser Form nicht findet. Diese Funktionen sind insbesondere über das Menü *Werkzeuge* verfügbar.

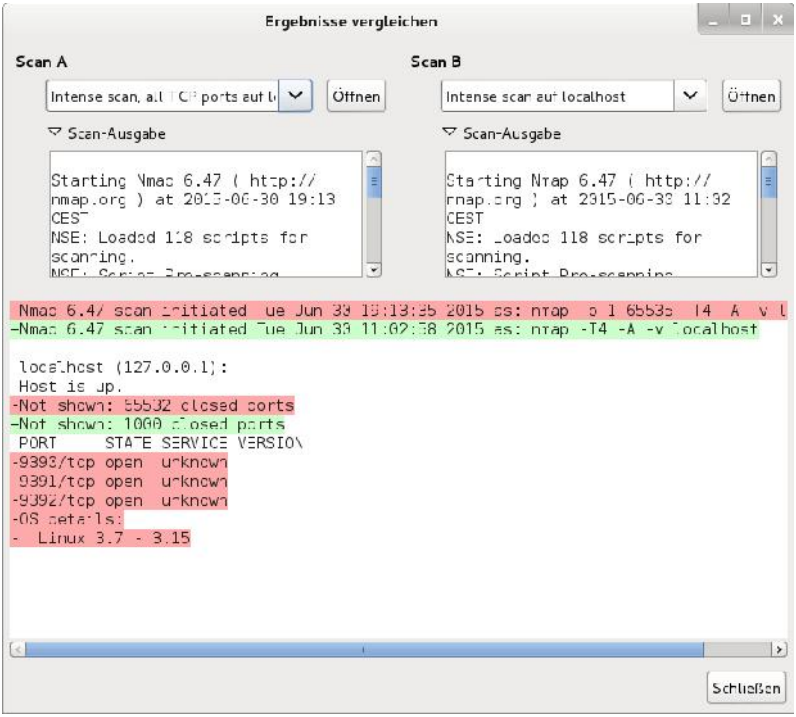
Sie können beispielsweise mit dem Menübefehl *Werkzeuge > Filtere Rechner* (*Strg + L*) einen sogenannten Host-Filter verwenden. Dann werden nur die Hosts, die diesem Filter entsprechen, angezeigt. Der Filter wird unterhalb der Nmap-Ausgabe eingeblendet.

Eine mögliche Eingabe ist beispielsweise *apache*, um die Ausgabe auf rein Apache-spezifische Informationen zu beschränken. Sie können auch Domain-Namen verwenden. Weitere Beispiele für die Filterung:

- **service:ftp** – Beschränkt die Ansicht auf FTP-Dienste.
- **os:linux ssh** – Zeigt nur Linux-Hosts an, auf denen SSH ausgeführt wird.
- **open:445** – Beschränkt die Ansicht auf alle Host mit offenem Port 445.

Da die Ergebnisse der Scan-Vorgänge sehr umfangreich werden können, ist die Suche nach bestimmten Informationen nicht immer einfach. Auch hierfür stellt Ihnen Zenmap die passende Lösung zur Seite: Mit dem Menübefehl *Werkzeuge > Scan-Ergebnisse* suchen (*Strg + F*) steht Ihnen eine Volltextsuche zur Verfügung. Mit Hilfe von logischen Ausdrücken können Sie die Suche auf bestimmte Inhalte beschränken oder Daten explizit ausschließen.

Ein weiteres Highlight ist der Vergleich der Scan-Ergebnisse. So können Sie beispielsweise prüfen, wie sich die Ergebnisse zweier intensiver Checks verschiedener Hosts unterscheiden. Sie können auch die Ergebnisse eines einfachen und eines intensiven Scan am gleichen Host miteinander vergleichen. Die Vergleichsfunktion rufen Sie mit dem Menübefehl *Werkzeuge > Ergebnisse vergleichen (Strg + D)* auf.



Der Vergleich zweier Scan-Ergebnisse mit Zenmap.

Die Verwendung ist ansonsten einfach: Im Bereich *Scan A* suchen Sie den ersten Scan-, im Bereich *Scan B* den zweiten Scan-Vorgang. Sie können zu beiden Scan-Vorgängen deren Ausgabe einblenden.

Die Vergleichsfunktion färben den einen Scan rötlich, den anderen grünlich ein. Insbesondere bei gleichen Zielen können Sie nun sehr schön die unterschiedlichen Ergebnisse direkt vergleichen.

Sie können die Farbzuweisung und auch andere Einstellungen in der Zenmap-Konfigurationsdatei *zenmap.conf* ändern.

7 Eigene Test-Skripts

Nmap (*http://nmap.org*) gehört schon seit Jahren zum Werkzeugkasten eines jeden Systemadministrators. Bei der Durchführung von Penetrationstests führt an dem Klassiker kaum ein Weg vorbei. Dabei stellt der Spezialist bereits standardmäßig eine beeindruckende Palette an Testskripts zur Verfügung. Doch Nmap bietet noch mehr als die simple Ausführung von vordefinierten Skripts. Mit der Nmap Scripting Engine, kurz NSE, steht Ihnen das vielleicht mächtigste und flexibelste Merkmal von Nmap zur Verfügung: Sie können mit der Engine eigene Skripts ausführen und damit verschiedenste Aufgaben beim Scannen und Analysieren automatisieren.

Die Skripts basieren auf der Programmiersprache Lua (*http://www.lua.o*rg), einer schnell zu erlernenden Sprache, die sich perfekt für das Entwickeln von Testskripts geeignet. Nmap führt diese Skripts parallel mit hoher Geschwindigkeit und Effizienz aus. Die meisten Nmap-Anwender verwenden das Tool für einfache Port-Scans und zur OS-Erkennung, aber NSE bringt das Scannen mit Nmap auf einen neuen Level. So können mit der Scan-Engine beispielsweise SQL Injektion-Verwundbarkeiten und mögliche Brute Force-Angriffspunkte identifiziert werden.

7.1 Basics

Nmap besitzt weit über 400 Standardskripts. Die Kernfunktionen von Nmap sind das Netzwerk-, Versions- und Verwundbarkeitserkennung. Mit NSE gehen Sie dann den nächsten Schritt und können diese Schwachstellen konkret ausnutzen und sich beispielsweise Zugriff auf ein System verschaffen. Damit Anwender den Überblick besser behalten, haben die Entwickler die Skripts in verschiedene Kategorien unterteilt. Anhand der Kategorienbezeichnung kann man den Aufgabenbereich erkennen: auth, broadcast, brute, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version und vuln.

Das Nmap-Entwickler-Team wollte frühzeitig eine vielseitig einsetzbare Umgebung anlegen, die Anwendern mehr Funktionen an die Hand gibt, als nur die einfache Ausführung von Testskripts. Die Kernfunktion von Nmap ist und bleibt das Scannen von Netzwerkkomponenten. Das führt NSE fort und bietet vielfältige Look-up-Funktionen auf Grundlage der Zieldomain. Mit Nmap können Sie offene Ports und verfügbare Freigaben (NFS, SMB und RPC) identifizieren.

Das Nmap-Versionserkennungssystem kann Tausende unterschiedliche Services identifizieren. Dabei greift der Scanner auf Tests und ein Übereinstimmungssystem

zurück, das auch die Verwendung von regulären Ausdrücken erlaubt. Nmap kann SNMP-Dienste zuverlässig erkennen und Brute-Force-Schwachstellen ausmachen. Beide Möglichkeiten sind einfach mit der NSE und entsprechenden Testskripts nutzbar.

Zwar ist Nmap kein Sicherheitsscanner à la Nessus oder OpenVAS, aber Dank der Scripting-Engine können Sie auch einfache Verwundbarkeitstests durchführen. Aktuell sind bereits verschiedene solcher Skripts in Nmap enthalten, aber die Entwickler planen, den Bestand auszubauen.

Nmap und die NSE taugen auch zur Backdoor-Erkennung. Hacker und Würmer hinterlassen nach Attacken meist einen Hintereingang, um sich bei erneuten Zugriffen einfacher Zugriff auf Systeme zu verschaffen. Die Nmap-Skripts können diese erkennen – nicht alle, aber doch viele. Die NSE kann auch zur Identifikation von komplexen Würmern und Backdoors verwendet werden.

NSE kann als allgemeine Skripting-Umgebung natürlich auch zur Ausnutzung der erkannten Schwachstellen verwendet werden. Insbesondere für Penetration-Tester ist die Möglichkeit interessant, Exploit-Skripts anzuwenden. Die Entwickler planen allerdings nicht, Nmap in ein vollständiges Exploitation-Framework wie Metasploit zu verwandeln. Aber es ist gut zu wissen, dass man Nmap mit der NSE auch hierfür verwenden könnte. Um in die Entwicklung eigener Nmap-Skripts einzusteigen, müssen Sie die Engine zunächst aktivieren. Bei einer bestehenden Nmap-Installation ist das einfach: Sie verwenden die Option *-sC*. Die Ergebnisse der Skriptausführung werden in Nmap integriert.

NSE-Skripts besitzen die Dateierweiterung *NSE* und sind bei einer Standard-Nmap-Installation im Verzeichnis */usr/share/nmap/scripts/* zu finden. Um ein spezifisches NSE-Skript auszuführen, verwenden Sie folgende Syntax:

```
nmap --script <script_name.nse> <zielhost>
```

Wenn Sie ein Skript unter Angabe des Pfads anwenden wollen, so verwenden Sie hierfür folgenden Befehl:

```
nmap --script </pfad/zum/skript/skript_name.nse> <zielhost>
```

Wenn Sie sich an die Erstellung eigener Skripts machen, sollten Sie diese nicht direkt auf die Menschheit loslassen, sondern testen, ob auch tatsächlich die gewünschten Aktionen dabei ausgeführt werden. Die Nmap-Entwickler stellen hierfür explizit einen Testserver unter der Domain *scanme.nmap.org* zur Verfügung. Verwenden Sie für die ersten Evaluierungen folgenden Befehl:

```
nmap --script </pfad/benutzerdefiniertes_nse_script.nse>
scanme.nmap.org
```

Bei den ersten Gehversuchen ist es zudem sinnvoll, das Debugging zu aktivieren. Hierzu verwenden Sie die Option *-d*, ergänzt um den Wert 1 bis 9. Ein Beispiel:

```
nmap -sV --script exploit -d3 <zielhost>
```

Je höher der Debug-Wert gesetzt ist, umso geschwätziger ist die Ausgabe.

7.2 Skript-Struktur

NSE-Skripts besitzen eine klar definierte Struktur. Die wird am deutlichsten, wenn man einen Blick auf ein bereits existierendes Skript wirft. In den Skripts sind nicht nur die durchzuführenden Tests und Aktionen definiert, sondern auch die Ausgabe. Nmap und NSE kennen vier verschiedene Skript-Typen, die durch eine unterschiedliche Ausführungsregel gekennzeichnet sind:

- prerule
- postrule
- portrule
- hostrule

Die *prerule*-Skripts werden vor den eigentlichen Scan-Vorgängen ausgeführt. Sie können beispielsweise zum Sammeln von Service-Informationen verwendet werden. Einige dieser Skript-Typen bestimmen damit auch neue Ziele, die dann von Folge-Skripts analysiert werden.

Das Skript *targets-sniffer.nse* prüft beispielsweise mit dieser Regel, ob Nmap im privilegierten Modus ausgeführt wurde und ob der Scanner die Netzwerkschnittstelle korrekt bestimmen kann:

```
prerule = function()
  return nmap.is_privileged() and
    (stdnse.get_script_args("targets-sniffer.iface") or
nmap.get_interface())
```

Die *postrule*-Skripts werden ausgeführt, nachdem Nmap alle Ziele gescannt hat. Sie sind beispielsweise sinnvoll für die Formatierung und Präsentation des Nmap-Outputs. Eines der bekanntesten *postrule*-Skripts ist *ssh-hostkey*. Es stellt eine Verbindung zu einem SSH-Server her, liest den öffentlichen Schlüssel aus und gibt ihn aus. Diese Regel ist wie folgt definiert:

```
postrule = function() return (nmap.registry.sshhostkey ~=
nil) end
```

Die sogenannten Service-Skripts enthalten die *portrule*-Funktion, die herauszufinden versucht, welche Services auf dem Ziel-Host mit welchen Ports ausgeführt wird. Nmap verfügt beispielsweise über 15 Skripts für die Prüfung von HTTP-Services auf Webservern. Wird auf einem Host ein Webserver mit verschiedenen Ports ausgeführt, so wird das Skript mehrfach ausgeführt – und zwar einmal pro Port.

Der vierte Skript-Typ im Bunde nennt sich Host-Skript. Dieser Typ kommt in der Regel dann zum Einsatz, wenn Nmap bereits Discovery-, Port-, Versions- oder Betriebssystem-Scans ausgeführt hat. Hierfür wird die *hostrule*-Funktion verwendet. Ein Beispiel ist *whois*, das nach der Zielauswahl mehr über den Eigentümer eines Zielsystems in Erfahrung bringt, oder *path-mtu*, das die maximale IP-Paketgröße bestimmt, die ein Ziel annimmt.

Die Scan-Vorgänge, die Nmap ausführt, sind durch verschiedene Phasen gekennzeichnet. Die sollten Sie kennen, damit Sie genau wissen, in welcher Reihenfolge der Scanner welche Aktionen üblicherweise ausführt und welche Schritte gegebenenfalls übersprungen werden können. Phase 1 wird als Pre-Scanning bezeichnet. Sie wird nur dann ausgeführt, wenn Sie die Optionen *-sC* oder *--script* verwenden. Dabei wird versucht, über NSE-Skripts zusätzliche Informationen über das Ziel zu erhalten.

In der zweiten Phase löst Nmap die Hostnamen der Ziele auf, damit mit der IP-Adresse weiter gearbeitet werden kann. In der dritten Phase findet Nmap dann heraus, ob das bzw. die Ziele erreichbar sind oder nicht. Dazu werden verschiedene Discovery-Techniken verwendet. Sie können diese Phase auch mit der Option *-Pn* überspringen.

Es folgt Phase vier: die Reverse DNS-Auflösung. Hier führt Nmap einen Reverse-Lookup durch, um den Hostnamen jedes Ziels zu erhalten. Sie können diese Phase mit der Option *-R* erzwingen und mit *-n* überspringen. Es folgt in Phase fünf das Port-Scanning – die klassische Aufgabe des Portscanners. In dieser Phase bestimmt Nmap den Status der zu analysierenden Ports. Auch diese Phase können Sie überspringen, indem Sie die Option *-sn* verwenden.

In der nächsten Phase erfolgt eine erweiterte Versionserkennung für die gefundenen offenen Ports. Diese Phase wird nur dann ausgeführt, wenn Sie bei der Nmap-Ausführung das Argument *-sV* angeben. Es folgt die siebte Phase, in der Nmap sich an die Ermittlung des Betriebssystems macht, das auf dem bzw. den Ziel-Hosts ausgeführt wird. Diese Prüfung erfolgt nur dann, wenn Sie die Option *-O* verwenden.

Drei weitere Phasen können bei Nmap-Scans noch durchlaufen werden. Die nächste ist die Traceroute-Phase, in der der Portscanner die Route zu den Hosts ermittelt. Diese Phase verlangt die Verwendung der Option *--traceroute*. In der vorletzten Phase kommen endlich die NSE-Skripts zum Einsatz. In dieser Phase wird auch der Output auf Basis der in dem Skript hinterlegten Ausgabekonfiguration generiert. Nmap formatiert die gesammelten Informationen und gibt Sie aus. Die letzte Phase wird auch als Post Scanning bezeichnet. Hier werden die in NSE-Skripts definierten Post-Scans durchgeführt. Sind keine weiteren abschließenden Prüfungen definiert, wird auch diese Phase weggelassen.

7.3 Skript-Kategorien

NSE-Skripts werden immer einer Kategorie zugeordnet. Das vereinfacht insbesondere Dritten die Beurteilung, ob ein Skript für ein bestimmtes Szenario geeignet oder weniger geeignet ist. Die Kategorien müssen Sie kennen, damit Sie Ihre Skripts über die Header-Konfiguration korrekt zuordnen können. Nachstehende Tabelle fasst die verschiedenen Kategorien zusammen:

Kategorie	Kurzbeschreibung
auth	Diese Skripts hantieren mit Authentifizierungsdaten und versuchen, die Authentifizierung auf dem Zielsystem zu umgehen. Beispiele hierfür sind *x11-access*, *ftp-anon* und *oracle-enum-users*. Ausgenommen sind Skripts für Brute Force-Attacken.
broadcast	Skripts dieser Kategorie führen Broadcasts aus, um bisher noch nicht identifizierte Hosts im lokalen Netzwerk zu identifizieren. Mit dem Argument *newtargets* werden neu erkannte Host automatisch der Scan-Warteschlange hinzugefügt.
brute	Skripts dieser Kategorien führen Brute Force-Attacken durch, um an die Zugangsdaten des Zielsystems zu gelangen. Nmap verfügt über Dutzende solche Skripts für verschiedene Protokolle, beispielswese *http-brute*, *oracle-brute* und *snmp-brute*.

Kategorie	Kurzbeschreibung
default	Hierbei handelt es sich um Standard-Skripts, die zum Einsatz kommen, wenn sie Nmap mit den Optionen *-sC* oder *-A* ausführen.
discovery	Diese Skripts versuchen aktiv, mehr über das Netzwerk zu erfahren, indem Sie beispielsweise öffentliche Register, SNMP-fähige Geräte, Verzeichnisdienste etc. abfragen. Beispiele hierfür sind *html-title*, das den Titel einer Webseite erhält, oder *smb-enum-shares*, das Windows-Freigaben einliest.
dos	Skripts, die dieser Kategorie angehören, können Denial of Service verursachen. Ursächlich dafür ist meist das Testen von Verwundbarkeiten. Dabei ist der Absturz der Dienste in der Regel eher ein unerwünschter Nebeneffekt als das Hauptziel.
exploit	Diese Skripts nutzen Verwundbarkeiten gezielt aus.
external	Die Skripts dieser Kategorie können Daten an externe Datenbanken und Netzwerkdienste senden. So kann Nmap beispielsweise einen Whois-Server nach weiteren Informationen über ein Zielsystem abfragen.
fuzzer	Diese Skripts senden unerwartete oder zufällige Fehler in jedem Paket, das an das Zielsystem übermittelt wird. Diese Technik kann dazu dienen, unerwartete oder unbekannte Schwachstellen in den Zielen zu identifizieren. Ein Beispiel hierfür ist *dns-fuzz*, das einen DNS-Server mit unbekannten Requests konfrontiert, bis dieser entweder abstützt oder die Verbindung unterbricht.
intrusive	Diese Skripts können nicht der Kategorie *safe* zugeordnet werden, weil hier das Risiko recht hoch ist, dass das Zielsystem abstürzt. Ursachen können die Verwendung von erheblichen Ressourcen oder die Ausnutzung bestehender Schwachstellen sein. Wenn Sie Ihre eigenen Skripts nicht einer spezifischen Kategorie zuordnen können, sollten Sie als *safe* oder *intrusive* klassifizieren.
malware	Mit Hilfe dieser Skripts können Sie herausfinden, ob das Zielsystem bereits von Malware oder Backdoors befallen ist. Beispiele hierfür sind *smtp-strangeport*, das SMTP-Server auf Schädlinge überprüft, und *auth-spoof*, das Identd-Spoofing-Daemone identifiziert.

Kategorie	Kurzbeschreibung
safe	Diese Skripts gelten als sicher und verfolgen nicht die Intension, die Ziele in Mitleidenschaft zu ziehen. Sie verbrauchen allerdings beträchtliche Netzwerkbankbreite oder andere Ressourcen. Beispiele hierfür sind *ssh-hostkey*, das den SSH-Schlüssel abfragt oder *html-title*, das den Titel einer Webseite einliest.
version	Diese Skripttypen sind eine Erweiterung der Versionserkennung, können aber nicht explizit ausgewählt werden. Sie werden nur dann ausgeführt, wenn Sie die Versionserkennung mit *-sV* aktiviert haben. Beispiele sind *skypev2-version*, *pptp-version* und *iax2-version.*
vuln	Mit dem letzten Skripttyp prüfen Sie spezifische bekannte Verwundbarkeiten. Eine Berichtausgabe erfolgt allerdings nur dann, wenn auch Schwachstellen identifiziert werden konnten.

Die Steuerung von Nmap kann auf der Konsolenebene erfolgen, aber weitaus einfacher ist das Scannen mit Hilfe der GUI Zenmap, die auch Bestandteil von Kali Linux ist. Mit dieser grafischen Benutzerschnittstelle ist es deutlich einfacher, mehr über Ihr Gegenüber zu erfahren. Bevor wir uns mit dieser Schnittstelle befassen, benötigen Sie noch ein wenig Hintergrundwissen über Ports und mögliche Schwachstellen.

Um zu erkennen, wie viele Ports auf Ihrem System verfügbar sind, verwenden Sie einfach unter Windows auf der Konsolenebene den Befehl *netstat-ano*, unter Linux den Befehl *netstat -nlo*. Sie werden nicht schlecht staunen, angesichts der langen Ausgabenliste. Jeder unsichere Port ist ein potenzieller Angriffspunkt, der von Angreifern genutzt werden kann. Da Hacker ebenfalls Port-Scanner wie Nmap einsetzen, um unbekannte Netzwerke auf erste erfolgversprechende Angriffspunkte hin zu prüfen, sollten Sie auch mit diesem Tool vertraut sein.

7.4 Gruß an die Welt!

Was konkret können wir nun mit NSE-Skripts anfangen und wie werden diese Aufgaben konkret implementiert? Der erste Teil dieser Frage ist noch recht einfach zu beantworten, der zweite Teil schon schwieriger. Sie können mit Ihren Nmap-Skripts Informationen über Zielsysteme sammeln, erweiterte Daten und versteckte Daten ausfindig machen, sich mit Brute-Force-Attacken Zugang zu Systemen verschaffen und die Ziele auf mögliche Verwundbarkeiten überprüfen.

Für die Entwicklung von NSE-Skripts sind Grundkenntnisse in Lua erforderlich. Außerdem greift Nmap auf die NSE-API und die leistungsfähige NSE-Bibliothek zurück. Da aber Lua einfach zu lesen und verstehen ist, schauen wir uns das Grundgerüst eines einfaches NSE-Skripts an, das die Bezeichnung *http-vuln-check.nse* erhält und für alle offenen Ports die Meldung *Hallo Welt!* ausgibt. Das Skript soll eine Web-Applikation auf Schwachstellen überprüfen:

```
-- Header --
-- Regel --
portrule = function(host, port)
    return port.protocol == "tcp"
            and port.number == 80
            and port.state == "open"
end


-- Aktion --
action = function(host, port)
    return "Hallo Welt!"
end
```

Alle Zeilen, die mit zwei Bindestrichen (--) beginnen, sind Kommentare.

Voranstehendes Skript umfasst drei Abschnitte. Im Header werden verschiedene Meta-Informationen zur Skript-Funktion, der Zielsetzung, dem Autor und zur Kategorie hinterlegt. Im Regelabschnitt definieren Sie die notwendigen Bedingungen für die Skriptausführung. Dieser Abschnitt muss zumindest eine der oben beschriebenen Funktionen enthalten: *portrule*, *hostrule*, *prerule* oder *postrule*.

Die meisten Skripts verwenden zumindest die *portrule*-Funktion – wie wir ebenfalls in diesem Beispiel sehen. In obigem Beispiel macht sich die portrule-Funktion die Nmap-API für die Prüfung der TCP-Ports 80 zunutze.

Der Abschnitt *Aktion* definiert die Skript-Logik. Die ist in diesem Fall sehr simpel: Findet das Skript einen offenen TCP-Port 80, gibt es die Meldung *Hallo Welt!* aus. Um das Skript an einem lokalen Webserver zu testen, führen Sie folgenden Befehl aus:

```
# nmap --script /pfad/http-vuln-check lokaler_zielhost -p
22,80,443
```

Die Ausgabe sollte in etwa wie folgt aussehen:

```
Starting Nmap 6.47 ( http://nmap.org ) at 2015-09-12:00:00
CET
Nmap scan report for zielhost (192.168.1.100)
Host is up (0.023s latency).
rDNS record for 192.168.1.100: localhost
PORT    STATE    SERVICE
22/tcp  filtered ssh
80/tcp  open     http
|_http-vuln-check: Hallo Welt!
443/tcp open     https

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

Wenn Sie Zenmap, die Nmap-GUI, für das Scannen verwenden, wird das Scan-Ergebnis auf der Registerkarte *Nmap-Ausgabe* dargestellt. Führen Sie Nmap auf der Konsole aus, erfolgt eine typische Konsolenausgabe.

Was NSE so leistungsfähig und vielfältig einsetzbar macht, ist die Verwendung von Bibliotheken, die Sie einfach in Ihre Skripts einbinden können. Eine der am häufigsten verwendeten Libraries ist *shortport.http*, die kurze *portrules*-Funktionen ausführt. Die geänderte Skript-Konfiguration sieht dann wie folgt aus:

```
-- Header --
local shortport = require "shortport"

-- Regel --
portrule = shortport.http

-- Aktion --
action = function(host, port)
    return "Hallo Welt!"
end
```

Der direkte Vergleich dieser beiden Skript-Konfigurationen zeigt, dass Sie mit der *shortrule*-Bibliothek den NSE-Code deutlich einfacher gestalten. Da die Bibliothek *shortport.http* alle typischen HTTP-Ports (80, 443, 631, 7080, 8080, 8088, 5800, 3872, 8180, 8000) und auch die verschiedenen Services (http, https, ipp, http-proxy etc.) prüft, erhalten Sie weit mehr Informationen über das Zielsystem.

7.5 Feinschliff

Oft ist das Ziel eines Skripts, mehr über die Services und Schwachstellen des Ziels in Erfahrung zu bringen. Hierfür müssen Sie das Basisskript ein wenig erweitern. Um beispielsweise die Verwundbarkeit einer Web-Applikation zu prüfen, versucht man, eine Webseite (etwa die Homepage) herunterzuladen, und begutachtet die Rückgabe des Webservers:

```
-- Header --
local shortport = require "shortport"
local http = require "http"


-- Regel --
portrule = shortport.http


-- Aktion --
action = function(host, port)

    local uri = "/index.html"
    local response = http.get(host, port, uri)
    return response.status

end
```

Voranstehendes Beispiel bedient sich dabei der Bibliothek *http* (*http://nmap.org/nsedoc/lib/http.html*). Man könnte das Skript nun weiter verfeinern, und nur eine Ausgabe generieren, wenn der HTTP-Code 200 zurückgegeben wird.

Der nächste Schritt ist meist das konkrete Identifizieren von Schwachstellen. Auch das ist in der Regel einfacher als vermutet, denn meist genügt es, die Service-Version zu identifizieren. In unserem Beispiel können Sie hierfür zur Bibliothek *string* greifen und folgende Ergänzung in dem Skript-Header vornehmen:

```
local string = require "string"
```

Damit Ihr Skript auch von Dritten eingesetzt werden kann, sollten Sie zum Abschluss noch am Header feilen. Dort können verschiedene Informationen enthalten sein. In Kombination mit dem @-Symbol können Sie Details zur Verwendung (*@usage*) und Ausgabe (*@output*) im Header hinterlegen. Hier ein Beispiel für einen simplen Header:

```
-- Header --
description = [[Beispiel, dass die ITA-Leser in die Entwick-
lung eigener Nmap-Skripts einführt]]
---
-- @usage
-- nmap --script http-vuln-check <ziel_host>
-- @output
-- PORT   STATE SERVICE
-- 80/tcp open  http
-- |_http-vuln-check: Vulnerable

author = "holger reibold"
license = "wie nmap"
categories = {"default", "safe"}

local shortport = require "shortport"
…
```

An diesem Punkt angelangt, stellt sich die Frage, wie man weiter in die Entwicklung von NSE-Skripts einsteigt. Der beste Weg hierfür ist sicherlich das Studium der NSE-Bibliothek, denn dort finden Sie jede Menge Input, um auch andere Dienste als HTTP zu analysieren. Nmap hat weitere interessante Möglichkeiten zu bieten, beispielsweise das parallele Scannen von verschiedenen Hosts.

Anhang A – More Info

Nmap ist ein ausgesprochen leistungsfähiges und flexibles Programm. In diesem Einstieg haben Sie einen Eindruck davon erhalten, was Sie alles mit dem Programm machen können. Je besser Sie Nmap kennenlernen, umso mehr werden Sie über das Programm und seine Möglichkeiten erfahren wollen. Ihre erste Anlaufstelle für weitere Informationen rund um Nmap ist die Nmap-Homepage (*https://nmap.org*).

Parallel dazu sollten Sie regelmäßig die wichtigsten Schwachstellendatenbanken auf für Sie relevante Meldungen prüfen. Nachfolgend finden Sie eine Liste der interessantesten Informationsquellen:

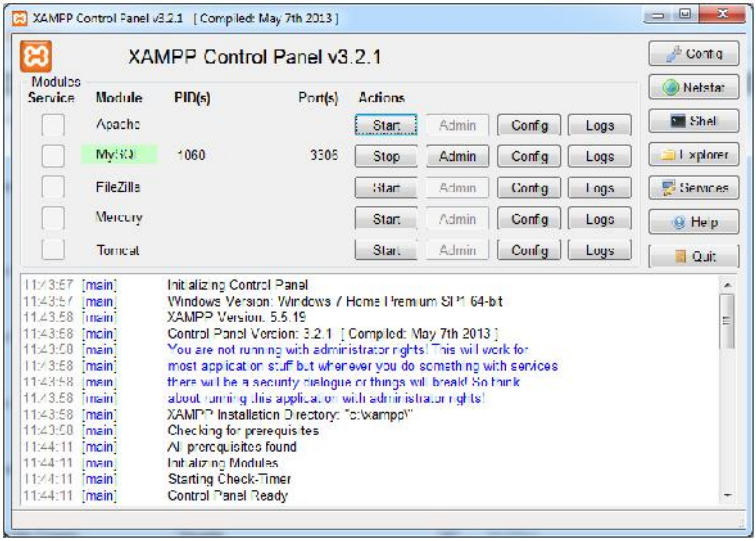
- **CERT Vulnerability Notes Database** – https://www.kb.cert.org/vuls/
- **Datenbank für IT-Angriffsanalysen des Hasso-Plattner-Instituts** – https://www.hpi-vdb.de/vulndb/
- **Exploit Database** – https://www.exploit-db.com/
- **Google Hacking Database** (GHDB) – https://www.exploit-db.com/google-hacking-database/
- **National Vulnerability Database** – https://web.nvd.nist.gov/view/vuln/search/
- **SecurityFocus** - http://www.securityfocus.com/vulnerabilities/

Die Liste erhebt keinen Anspruch auf Vollständigkeit, sondern ist als Ausgangspunkt für Ihre weiteren Recherchen gedacht.

Anhang B – Eigene Testumgebung

In diesem Handbuch haben Sie die wichtigsten Aktionen mit Nmap kennengelernt. Sie dürfen nun einen Fehler nicht begehen: Die hier beschriebenen Aktionen an Ihren Produktivitätsystemen zu testen. Sie fragen zu Recht: Warum denn nicht?

Die Antwort ist einfach: Bevor Sie die eine oder andere Aktion an einem laufenden System vornehmen, sollten Sie diese immer zunächst auf einem Testsystem prüfen. Nur so können Sie sicherstellen, dass die aktuellen Produktivitätsysteme keinen Schaden nehmen oder in irgendeiner Form beeinträchtigt werden.



Die ideale Testumgebung für alle Apache-MySQL-PHP-basierten Anwendungen: XAMPP.

Das bedeutet für Sie, dass Sie am besten eine typische Umgegebung lokal nachbauen oder spiegeln sollten. Mit Hilfe von XAMPP (*http://www.apachefriends.org/de/xampp.html*) ist das recht einfach – insbesondere als Basis für webbasierte Umgebungen.

XAMPP verknüpft all jene Komponenten, die für die Nutzung von solchen Applikationen erforderlich sind. Sie müssen sich dabei nicht durch aufwendige Installationen und Konfigurationsdialoge kämpfen, sondern können unmittelbar nach der Inbetriebnahme loslegen. Über einen speziellen Installer können Sie sogar Shopping-System, Blogs, ERP- und CRM-Systeme mit minimalem Aufwand installieren.

Sie können auch einen bereits angelegten Magento Shop einfach spiegeln und dann einer ausgiebigen Sichherheitsanalyse unterziehen. Dazu gehen Sie wie folgt vor:

1. Packen Sie alle Dateien (auch die versteckten Dateien) auf dem bisherigen Server in ein Archiv.
2. Dann greifen Sie zu phpMyAdmin und exportieren alle Tabellen der bisherigen Datenbank in eine SQL-Datei.
3. Auf Seiten des XAMPP-Systems importieren Sie die SQL-Datei.
4. Dann suchen Sie in der Tabelle *core_config_data* nach den Einträgen für *web/secure/baseurl* und *web/unsecure_baseurl* und passen die Domain an.
5. Entpacken Sie dann das in Schritt 1 angelegte Archiv auf dem neuen Server.
6. Als Nächstes leeren Sie die im Verzeichnis */var* befindlichen Ordner (nicht löschen!).
7. Passen Sie dann die Datei *.htaccess* im Hauptverzeichnis an.
8. Ein letzter Schritt ist erforderlich: Passen Sie im Ordner */app/etc/* die Dateien *config.xml* und *local.xml* an. Tragen Sie hier die neuen Datenbankzugangsdaten ein. Fertig.

Tipp – Magento-, WordPress und XAMPP-Buch zum Download

Zum Anlegen einer Testumgebung stellen wir Ihnen drei, zum Teil, umfangreiche Handbücher zum Download bereit:

- Magento komapkt
- XAMPP 1.8 kompakt
- WordPress kompakt

Außerdem finden Sie im FreeBooks-Bereich der Verlags-Website zwei weitere E-Books zu den Security Scannern Nessus und OpenVAS zum kostelosen Download: *http://www.brain-media.de/freebooks.html*.

Index

A

ACK-Test 33
Adressbereich 28
Aggressive-Modus 65
Angriffspunkt 7
ARP-Ping 35
Auditierung 8

B

Benutzer-Account auslesen 79
Benutzerdefinierter TCP-Scan 49
Berichtausgabe 68
Betriebssystem 41
Betriebssystem ermitteln 59
Betriebssystemdetails 103
Betriebssystemerkennung 11, 19
Blog 81
Broadcast 26
Brute-Force 115
Brute-Force-Attacke 78, 81, 82, 90

C

CERT Vulnerability Notes Database .. 127
CIDR 25
Computertechnik 9
Content-Management 81
Content-Managementsystem 30
Cross Site Scripting 83
Cross Site Tracing 75

D

Dateien aufdecken 76
Datenbank für IT-Angriffsanalysen ... 127
Datenbank testen 87
Datenbanksicherung 76
Debugging 68
Debug-Level 21
Decoy-Scan 66
Delay 112
DHCP-Server 27
Diensteliste 104
Discovery 118
DMG 13
DNS 55
DNS-Auflösung 16, 35
DNS-Reserve-Abfrage 40
DNS-Server 16
DoS 86
Druckermodell 57
Dynamic Ports 37

E

E-Mail-Account aufdecken 92
Erste Schritte 14
Exploit 20
Exploit Database 127

F

Fälschen 20
Filter 39, 113
FIN 39
Fingerprint 56
FIN-Scan 47
Firewall 20, 39, 51, 65
Fragment 65
FTP-Bounce-Scan 17, 51
Fuzzdb 85
Fyodor 10

G

Gefiltert 38
Generation 59
Gerätetyp 15, 59
Geschlossen 38
Geschlossen|gefiltert 38
Google Hacking Database 127
Gruppengröße 62
GUI 8

H

Hacker 7
Header-Konfiguration 119
Herstellername 59
Hop 102
Host 25
Host erkennen 29
Host-Ermittlung 16
Hostgruppe 19
hostrule 117
HTML 68
HTTP 55
HTTP-Methode 75

I

IANA 36
ICMP 34
ICMP Echo-Request 30
ICMP Port-unreachable 47
ICMP-Ping 34
Idle-Scan 49
IDS 65
IGMP 34
IMAP-Server attackieren 96
Infrastruktur 7
Infrastrukturkomponente 7
Installation 12
Intensität 58
Internet 9
Intrusion Detection System 20, 53
Inventarisierung 8
IP-Adresse 25
IP-Protokoll-Ping 34
IP-Protokoll-Scan 17, 50
IPv6-Adresse 27
IPv6-Scanning 22

J

Joomla! 82

K

Kali Linux 12
Köder 66
Konfigurationsverzeichnis 76

L

Latenz 62
List-Scan 30
Lua 115
Lyon, Gordon 10

M

MAC-Adresse 15, 41
Mailserver 55, 92
Matrix Reloaded 11
Monitoring 8
MTU 20
MySQL 37, 71, 87, 90
MySQL-Benutzer auslesen 88
MySQL-Datenbanken abrufen 87
MySQL-Konfiguration 90
MySQL-Variablen auslesen 88

N

Namensauflösung 16
Nameserver 55
National Vulnerability Database 127
Ndiff 8
Network Mapper 8
Netzwerkscanner 15

Netzwerksegment ... 16
Netzwerkstruktur ... 102
Netzwerktechnik ... 9
Netzwerktopologie ... 7, 106
Nmap ... 8, 115
nmap <zielhost> ... 38
Nmap Scripting Engine ... 11
Nmap-Homepage ... 127
Nmap-Skript ... 116
NoSQL ... 87
Nping ... 8
NSE ... 78, 115
NSE-Bibliothek ... 78
NULL ... 39

O

Offen ... 38
Offen|gefiltert ... 38
Offene Relays aufspüren ... 94
Offener Web-Proxy ... 75
OpenSSL ... 56
OpenVAS ... 116

P

Paketmanager ... 12
Payload ... 20
PBNJ ... 71
Penetration Testing ... 8
Performance ... 19, 41, 47, 61, 62
Ping ... 30, 111
Ping-Scan ... 31
Plausibilitätsprüfung ... 31
Polite-Template ... 65
POP3-Server attackieren ... 95
Port ... 10, 14
Port-Auswahl ... 52
Port-Bereich ... 52
Port-Eigenschaft ... 38
Portliste ... 37
portrule ... 117
Port-Scan ... 16
Portscanner ... 10
Port-Scanning ... 36, 38
Port-Scan-Techniken ... 44
Port-Spezifikation ... 18
Port-Tabelle ... 15
postrule ... 117
prerule ... 117
Private Ports ... 37
Profileditor ... 111
Proxy-Server ... 75
Prüfsumme ... 67

Q

Quick Scan ... 99

R

Rechnerbetrachter ... 110
Rechnerliste ... 104
Rechnerstatus ... 102
Registered Port ... 36
Reverse DNS-Auflösung ... 118
Root-Account finden ... 89
Round-Trip-Zeit ... 107
RST ... 32

S

Saxon ... 71
Scan-Engine ... 115
Scan-Ergebnis ... 8, 68
Scan-Ergebnisse vergleichen ... 114
Scan-Konfiguration ... 15
Scan-Option ... 112
Scan-Performance ... 13
Scan-Profil ... 99, 111
Scan-Rate ... 64
Scan-Reihenfolge ... 18, 52, 67
Scan-Variante ... 98
Scan-Vorgang ... 29, 98, 101
Scan-Zeit ... 61
Schutzmechanismus ... 65
Schwachstelle ... 7
Schwachstellen aufdecken ... 83

Security Scanner 10
SecurityFocus 127
Service ermitteln 55
Session Initiation Protocol 37
Sicherheitslücke 7, 38
Sicherheitsscanner 15
SIP 37
Skriptausführung 116
Skript-Typ 117
Slowloris Denial-of-Service-Attacke 86
Smart Home 9
Smartphone 9
SMTP 55
SMTP-Passwort knacken 94
SMTP-User auslesen 95
Spoofing 20, 66, 120
SQL Injection 85
SQlite 71
Standard-Scan 18, 40
Status 14
Strukturdarstellung 107
Subnetz 29
SYN/ACK 32
SYN-Flag 32
Syntax-Hervorhebung 100

T

TCP-ACK-Paket 30
TCP-ACK-Ping 32
TCP-ACK-Scan 48
TCP-Connect-Scan 46
TCP-Header 66
TCP-Maimon-Scan 49
TCP-NULL-Scan 47
TCP-Paket 32
TCP-SYN-Scan 45
TCP-Window-Scan 48
Test-Skript 115
Testumgebung 129
Three-Way-Handshake 32
Timeout 19, 62
Timing-Template 64, 79
TRACE 83
Traceroute 15, 22, 35, 111, 119

U

UDP 38
UDP-Ping 33
UDP-Scan 17, 46
Ungefiltert 38
Update 13

V

Versionsausgabe 22
Versionserkennung 15, 57
Verzeichnis aufdecken 76
Visualisierung 102
Volltextsuche 113

W

WAF 83
Web Application Firewall 83
Webserver 28, 55
Webserver scannen 74
Well known ports 36
Windows-Registry 13
WordPress 77, 81

X

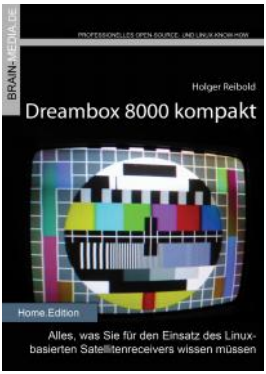
X11 13
XAMPP 129
Xmas 39
Xmas-Scans 47
XML 21, 68, 69
XML-Aufgabe 21
XSL 21
xsltproc 71
XSLT-Prozessor 71

Z

Zeitspanne 19

Zenmap 8, 97, 121
Zenmap-Konfigurationsdatei 105
Zielnetzwerk 31
Zufallsmechanismus 28, 53
Zugangsdaten testen 80
Zusatzinformation 14
Zustand 14, 38
Zustandskombination 14

Weitere Brain-Media.de-Bücher



Dreambox 8000 kompakt

Die Dreambox 8000 stellt ihre Vorgänger allesamt in den Schatten. Was Sie alles mit der Dreambox 8000 anfangen können, verrät Ihnen die Neuauflage unseres Dreambox-Klassikers. Mit einem Vorwort des Dream Multimedia-Geschäftsführers Karasu.

Umfang: 450 Seiten plus CD
ISBN: 978-3-939316-90-9
Preis: 29,80 EUR

X-Plane 10 kompakt

Der Klassiker unter den Flugsimulatoren geht in die zehnte Runde. Viele neue Funktionen und verbessertes Handling warten auf die Anwender. Kein Wunder also, dass die Fangemeinde wächst und wächst. Unser Handbuch beschreibt alles, was Sie für das Fliegen mit X-Plane wissen sollten.

Umfang: 430 Seiten
ISBN: 978-3-939316-96-1
Preis: 24,80 EUR

Audacity 2.0 kompakt

Audacity ist zweifelsohne das beliebteste freie Audioprogramm. Vom anfänglichen Geheimtipp hat sich der Editor zum Standard für die Aufzeichnung und Bearbeitung von Audiodaten gemausert. Das Vorwort steuert der ehemalige Core-Entwickler Markus Meyer bei.

Umfang: 306 Seiten
ISBN: 978-3-95444-027-6
Preis: 24,80 EUR

Evernote kompakt

Bei der alltäglichen Informationsflut wird es immer schwieriger, Wichtiges von Unwichtigem zu trennen, Termine und Kontakte zu verwalten. Mit Evernote können Sie diese Flut bändigen und Ihren Alltag optimieren. "Evernote kompakt" vermittelt das notwendige Know-how für den Einsatz von Evernote auf Ihrem Desktop, Smartphone und online.

Umfang: 320 Seiten
ISBN: 978-3-95444-098-6
Preis: 22,80 EUR

Fire TV kompakt

Mit Fire TV hat Amazon eine tolle kleine Box für das Online-Entertainment auf den Markt gebracht, die für wenig Geld die gesamte Palette der Internet-basierten Unterhaltung abdeckt. In diesem Handbuch erfahren Sie, was Sie alles mit der kleinen Box anstellen können.

Umfang: 182 Seiten
ISBN: 978-3-95444-172-3
Preis: 16,80 EUR

Magento SEO kompakt

Magento ist die Standardumgebung für den Aufbau eines Online-Shops. Doch damit Sie mit Ihren Shop-Angebot auch im Internet wahrgenommen werden, müssen Sie ein wenig die Werbetrommel rühren und den Shop für Google & Co. optimieren. Mit wenigen Handgriffen machen Sie Ihren Online-Shop SEO-fest und maximieren Ihre Verkäufe.

Umfang: 100 Seiten
ISBN: 978-3-95444-098-6
Preis: 14,80 EUR

Wireshark kompakt

Wireshark ist der mit Abstand beliebteste Spezialist für die Netzwerk- und Protokollanalyse. In diesem Handbuch lernen Sie, wie Sie mit dem Tool typische Administratoraufgaben bewältigen. Das Buch beschränkt sich dabei auf die wesentlichen Aktionen, die im Admin-Alltag auf Sie warten, und verzichtet bewusst auf überflüssigen Ballast.

Umfang: 170 Seiten
ISBN: 978-3-95444-176-1
Preis: 16,80 EUR

Scribus 1.5 kompakt

Scribus ist längst ein ebenbürtiger Gegenspieler von InDesign & Co. In unserem Handbuch erfahren Sie alles, was Sie für den erfolgreichen Einstieg wissen müssen.

460 Seiten Praxis-Know-how. Dazu viele Tausend ClipArts und Schriften zum kostenlosen Download.

Umfang: 460 Seiten
ISBN: 978-3-95444-124-2
Preis: 27,80 EUR

Weitere Titel in Vorbereitung

Wir bauen unser Programm kontinuierlich aus. Aktuell befinden sich folgende Titel in Vorbereitung:

- Android Forensik
- Android Security
- Alfresco 5.0 kompakt
- WordPress 4.x kompakt
- Smart Home kompakt
- Das papierlose Büro
- wa3f kompakt
- SmoothWall kompakt

Plus+

Plus+ − unser neues Angebot für Sie ... alle E-Books im Abo. Sie können 1 Jahr alle Brain-Media-Bücher als E-Book herunterladen und diese auf Ihrem PC, Tablet, iPad und Kindle verwenden − und das ohne irgendwelche Einschränkungen. Das Beste: Plus+ schließt auch alle jene Bücher ein, die in diesem Jahr noch erscheinen.

Und das zum Sonderpreis von 29 Euro! Ein unschlagbares Angebot!

Auf unserer Website steht ein detaillierter Überblick aller Titel im PDF-Format zum Download bereit (ca. 6,2 MB), der bereits zu Plus+ gehörende Titel aufführt und die in naher Zukunft hinzukommen.