Auditing mit Linux

Entdecken Sie die Schwachstellen Ihres Systems

Marc Ruef

Computersicherheit war lange Zeit ein Thema, bei dem reagiert wurde. Die Entwickler von Anwendungen und die Administratoren von Systemen zeigten sodann Reaktionen auf Aktionen. Auf Attacken wurde mit dem Einspielen der neuesten Patches, dem Anpassen von Firewall-Regeln oder dem Implementieren eines Intrusion Detection-Systems reagiert. Dies alles ist lobenswert, schön und gut. Doch die Angreifer waren den Entwicklern, Administratoren und Benutzern stets einen Schritt voraus. Durch Vulnerability Assessments sollen frühzeitig die Schwachstellen eines Systems erkannt und Massnahmen zur Stärkung der Sicherheit dessen eingeleitet werden. So wird das Zeitfenster minimiert, in dem ein Angreifer erfolgreich seiner Tätigkeit nachgehen kann.

Das Durchführen von Security Audits erfordert einerseits das Verständnis für die organisatorischen und technischen Hintergründe der Umgebung, andererseits eine klare Vorgehensweise. Ein durchdachter Ablaufplan ist das Rückgrat, mit dessen Hilfe das Auditing effizient und professionell durchgeführt werden kann.

Die Hauptpunkte umfassen die Vorbereitungen des Audits, das Durchführen der Scans, das Dokumentieren der Gegebenheiten (Reporting) und das Umsetzen der Gegenmassnahmen. Je nach Umfang des Auditings kann für die einzelnen Punkte mehr Zeit erforderlich sein. So ist es aber auch möglich, dass einzelne Stufen ganz wegfallen. Zum Beispiel, wenn im sehr kleinen Rahmen ein einzelnes System untersucht werden soll. Die Vorbereitungen und das Reporting können in einem solchen Fall, im Gegensatz zu einem professionellen Security Audit eines grossen Konzerns, vernachlässigt werden.

Vorbereitungen
Die Vorbereitungn sollen in erster Linie Probleme während des Durchführens des Security Audits verhindern. Nichts ist ärgerlicher, weder während der technischen Phase des Scannings administrative und organisatorische Gegebenheiten abzuklären, anzupassen oder zusätzliche Abläufe einzubinden. Dies kann den ganzen Fluss des Vorhabens durcheinander bringen und in Form einer Kettenreaktion das Projekt unter Umständen gefährden.

So gilt es zu Beginn genau zu definieren, welche Systeme wie überprüft werden sollen. Hierbei werden die Assets festgelegt. Man bestimmt, welche Geräte und Daten welchen Wert beigemessen bekommen. So macht es während der technischen Durchführung des Audits einen grossen Unterschied, ob man sich gerade einem wichtigen Datenbanksystem annimmt, oder ob man eine der eher unkritischen Workstations untersucht.

Die Arbeit der Auditoren kann dadurch erleichtert werden, indem ihnen alle benötigten Informationen zugekommen lassen werden. Oberstes Ziel eines jeden Security Audits ist die Wahrung und Stärkung der Sicherheit einer Umgebung. Dieses Ziel kann schnellstmöglich erreicht werden, indem alle Mängel und Informationen, die auf diese Mängel hinweisen könnten, zusammengetragen werden. Das Verheimlichen von Unschönheiten und Patzern kann das Erreichen des obersten Zieles erschweren, ja gar unter Umständen verhindern. Um es im Sinne Friedrich Nietzsches zu sagen: Die Wahrheit macht stark.

Desweiteren müssen alle Abklärungen bezüglich der Kompetenzbereiche getroffen werden. Wer ist für welches System zuständig. Und welche Prozedur muss durchlaufen werden, falls ein Problem - zum Beispiel der versehentliche Abschuss eines Hosts - auftaucht. Sowohl für Auditoren als auch für die Auditierten gibt es nichts unangenehmeres, weder unerwünscht negative Einflüsse auf den Betriebs durch das Assessment.

Durchführung
In einer ersten Phase des technischen Teils des Security Audits werden versucht die aktiven Systeme zu erkennen. Dies wird in der Literatur als Mapping bezeichnet.

Die klassische Methode hierzu, die auch viele Netzwerkadministratoren kennen werden, ist durch das ICMP-Mapping gegeben. Dabei verschickt der Auditor ein ICMP echo request-Paket an das potentiell

aktive System. Der Kernel dessen müsste auf diese Anfrage automatisiert mit einer ICMP echo reply-Rückantwort entgegnen, um sein Vorhandensein und seine Erreichbarkeit kundzutun. Dieser Zugriff kann mit dem ursprünglich von Mike Muss entwickelten Netzwerkdiagnoseutility "Ping" durchgeführt werden, das Teil eines jeden modernen, netzwerkfähigen Betriebssystems ist. Diese Mapping-Technik wird deshalb auch gerne Ping-Mapping genannt.

Die ersten Versionen des ping-Kommandos waren sehr simpel. Wird ein Ping für ein Zielsystem durchgeführt, wies das Utility dieses entweder als "alive" (dt. lebend, ansprechbar) oder "not alive" (dt. nicht lebend, nicht ansprechbar) aus. Die jüngeren Ping-Varianten liefern dagegen zusätzliche Diagnoseinformationen, wie zum Beispiel die verstrichene Zeitdauer für den Empfang der erfolgreichen Rückantwort (round trip time, rtt) oder eine Statistik zu den verlorenen und empfangenen Paketen.

Praktisch sämtliche kommandozeilenorientierten Ping-Versionen lassen sich in ihren Grundzügen gleich bedienen. Nach dem Aufruf des Kommandos folgt der Hostname oder die IP-Adresse des Zielsystems.

```
mruef@debian~$ ping www.scip.ch
PING www.scip.ch (192.168.0.2): 56 data bytes
64 bytes from 192.168.0.2: icmp_seq=0 ttl=64 time=0.0 ms
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.0 ms
64 bytes from 192.168.0.20: icmp_seq=2 ttl=64 time=0.0 ms

--- 192.168.0.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Die Ping-Version unter Linux verschickt bei diesem Aufruf unendlich viele ICMP echo request-Pakete, bis der Zugriff durch das Drücken von Control+C abgebrochen wird. Das Verhalten des Ping-Kommandos lässt sich natürlich durch das Miteinbeziehen von verschiedenen Parametern beeinflussen. Die man-Page und readme-Dateien der jeweiligen Ping-Implementierungen sollten alle nötigen Informationen bereithalten. Diese hier aufzuzählen oder nur zu skizzieren würde den Umfang des Artikels bei weitem sprengen.

Etwas unpopulärere Methoden des Mappings sind durch das TCP- bzw. UDP- und das ARP-Mapping gegeben. TCP- und UDP-Mapping wird von nmap und indirekt von jeder anderen TCP- und UDP-Applikation unterstützt. Der Mapper schickt dabei ein "provozierendes" Paket an das Zielsystem. Wenn dieses Reagiert, egal in welcher Form, kann man davon ausgehen, dass es aktiv am Netzwerk ist. TCP-Mapping kann bei nmap zusammen durch das Miteinbeziehen des Parameters -sP aktiviert werden. Einmal mehr liefert die man-Page des Programms zusätzliche Hintergrundinformationen zu dieser Funktion.

```
mruef@debian~$ nmap -sT -p 80 www.scip.ch

Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-07-03 11:15 CEST
Interesting ports on www.scip.ch (192.168.0.2):
Port       State       Service
80/tcp     open        http

Nmap run completed -- 1 IP address (1 host up) scanned in 0.311 seconds
```

Das ARP-Mapping ist sehr unpopulär. Dies liegt hauptsächlich daran, dass es nicht über verschiedene Netzwerk-Segmente - also auch nicht über das Internet - funktioniert. Ähnlich wie beim ICMP-Mapping schickt der Mapper ein provozierendes ARP-Pakets an die IP- bzw. MAC-Adresse des Zielsystems. Ist dieses Vorhanden, konnte die Anfrage entgegennehmen und verarbeiten, schickt es eine entsprechende Meldung zurück. ARP-Mapping kann sehr einfach, ähnlich dem Ping-Kommando, mittels arping durchgeführt werden.

```
debian:~# arping -c 3 192.168.0.2
ARPING 192.168.0.2
60 bytes from 00:04:5a:71:b4:8d (192.168.0.2): index=0 time=109.911 usec
60 bytes from 00:04:5a:71:b4:8d (192.168.0.2): index=1 time=141.978 usec
60 bytes from 00:04:5a:71:b4:8d (192.168.0.2): index=2 time=144.005 usec

--- 192.168.0.2 statistics ---
3 packets transmitted, 3 packets received,   0% unanswered
```

Offene Ports identifizieren

Unter dem Wort Skript-Kiddie versteht man jemanden, der sich irgendwelchen vorgefertigten Tools,

Skripten oder Exploits bedient, um ein System zu penetrieren, ohne das Verständnis für die Funktionsweise der eingesetzten Software aufzubringen. Unweigerlich assoziiert man zu einem Skript-Kiddie Portscanning. Bei dieser Technik werden automatisiert einer, mehrere oder sämtliche Ports eines Systems auf ihren Status überprüft. Das Ziel dabei ist in den meisten Fällen, Ports zu entdecken, die im LISTENING-Status (dt. abhörend) sind. Der Volksmund bezeichnet sie auch als offen, den durch sie ist der Server in der Lage, Anfragen entgegenzunehmen und darauf zu reagieren. Dies stellt stets eine ernstzunehmende Angriffsfläche dar.

Um den Status eines Ports eines Hosts zu identifizieren, reicht im Grunde eine Netzwerk-Applikation (z.B. Telnet) und ein Protokoll-Analyzer (z.B. TCPdump). Durch das Durchführen eines Zugriffs mittels der Netzwerk-Applikation und durch das Mitschneiden der Reaktion kann man mit solidem TCP/IP-Grundwissen den Status eines Ports ausmachen.

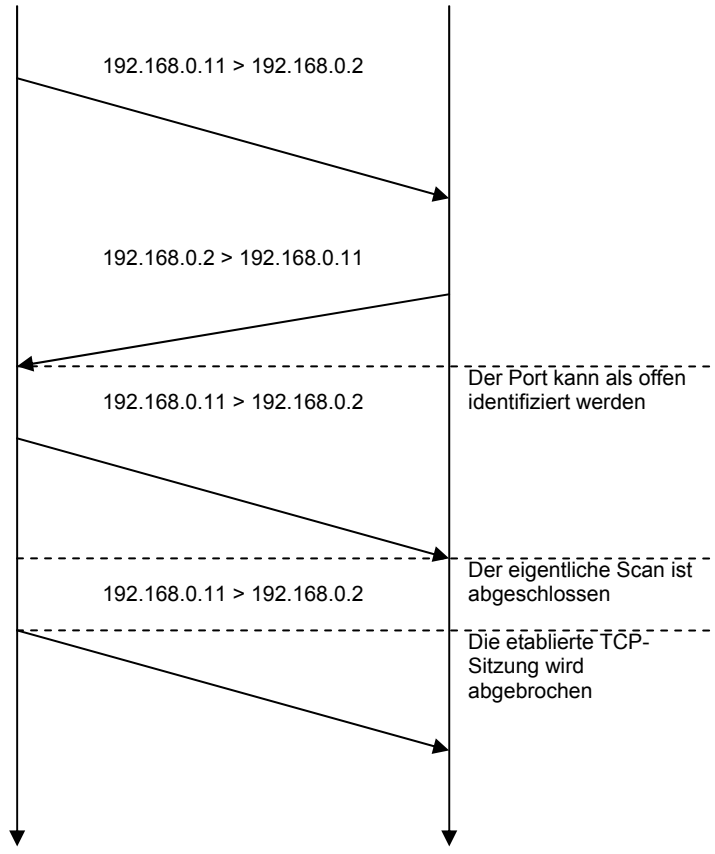
Viel einfacher ist jedoch das heranziehen eines sogenannten Portscanners. Diese Software prüft automatisiert Portbereiche auf ihren Status. Der mitunter populärste Vertreter dieser Software-Gattung ist das von Fyodor entwickelte nmap (Network Mapper). nmap gibt es schon seit vielen Jahren und ist aus der Welt der Netzwerksicherheit nicht mehr wegzudenken. Das zeilenorientierte Programm erfreut sich aufgrund seiner Portabilität, Geschwindigkeit, Kompaktheit und den gegebenen Möglichkeiten grösster Beliebtheit. Selbst im Film "Matrix Reloaded" (Warner Bros., 2003) wird in einer Hacking-Szene ein Durchlauf von nmap gezeigt.

```
maru@debian~$ nmap -sT -p 21,25,80 www.scip.ch

Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-07-03 11:19 CEST
Interesting ports on www.scip.ch (192.168.0.2):
Port       State       Service
21/tcp     open        ftp
25/tcp     open        smtp
80/tcp     open        http

Nmap run completed -- 1 IP address (1 host up) scanned in 0.321 seconds
```

Die zeilenorientierte Version von nmap - es gibt jedoch auch eine grafische Version für Windows oder verschiedene GUI-Addons - kann so wie auch das simple Ping-Programm durch das Übergeben des Hostnamens oder der IP-Adresse des Zielsystems starten. Ohne Angabe der Scan-Variante wird ein herkömmlicher full-connect TCP-Portscan durchgeführt. Wie der Name schon vermuten lässt, werden dabei die TCP-Ports auf ihren Status überprüft. Unter der full-connect Methode versteht man einen Zugriff, der sämtliche drei Stati des Drei-Wege-Handschlags (engl. three way handshake) von TCP durchläuft. Der Client, in unserem Fall der Scanner, verschickt ein TCP-Segment mit gesetzter SYN-Flagge. Der Server, in unserem Fall das gescannte System, antwortet bei geschlossenem Port mit einem TCP-Segment mit gesetzter TCP-Flagge. Damit



gibt es dem Client zu verstehen, dass der Port nicht angesprochen werden kann, da er sich im CLOSED-Status (dt. geschlossen) befindet. Der Scanner weist den TCP-Port sodann als geschlossen aus. Befindet sich der Zielport im LISTENING-Status, schickt der Server ein TCP-Segment mit gesetzten SYN/ACK-Flaggen zurück. Dadurch gibt er seine Empfangsbereitschaft auf diesem Port bekannt und bestätigt den Empfang der SYN-Anfrage. Der Scanner muss den Empfang der Rückantwort des Servers mit einem TCP-Segment mit gesetzter ACK-Flagge bestätigen, um den dritten Teil des Drei-Wege-Handschlags zu beenden.
Man spricht sodann von einer etablierten (engl. established) Verbindung. Es wäre nun eine Datenübertragung im Rahmen von TCP möglich. Dies ist jedoch bei einem reinen Portscan nicht der Fall. Nun muss nämlich der Portscanner die Verbindung wieder abbauen, denn die Backlog-Queues der involvierten Systeme müssen Platz für weitere Verbindungen - vorwiegend im Rahmen des Portscans - haben. Jenachdem, wie der Scanner konzipiert wurde, beendet er die Verbindung anders. Nmap nutzt die abrupte Methode, wie sie in der deutschen Übersetzung des Buches "Intrusion Detection-Systeme" von Stephen Northcutt und Judy Novak genannt wird. Diese brachiale Methode, so wird sie im Buch "Hacking Intern" betitelt, wird mit einem RST-Paket des Scanners eingeleitet. Sobald dieses abgesetzt wurde, kann der Client keine Daten mehr im Rahmen der TCP-Sitzung empfangen; und er wird voraussichtlich auch keine mehr verschicken.

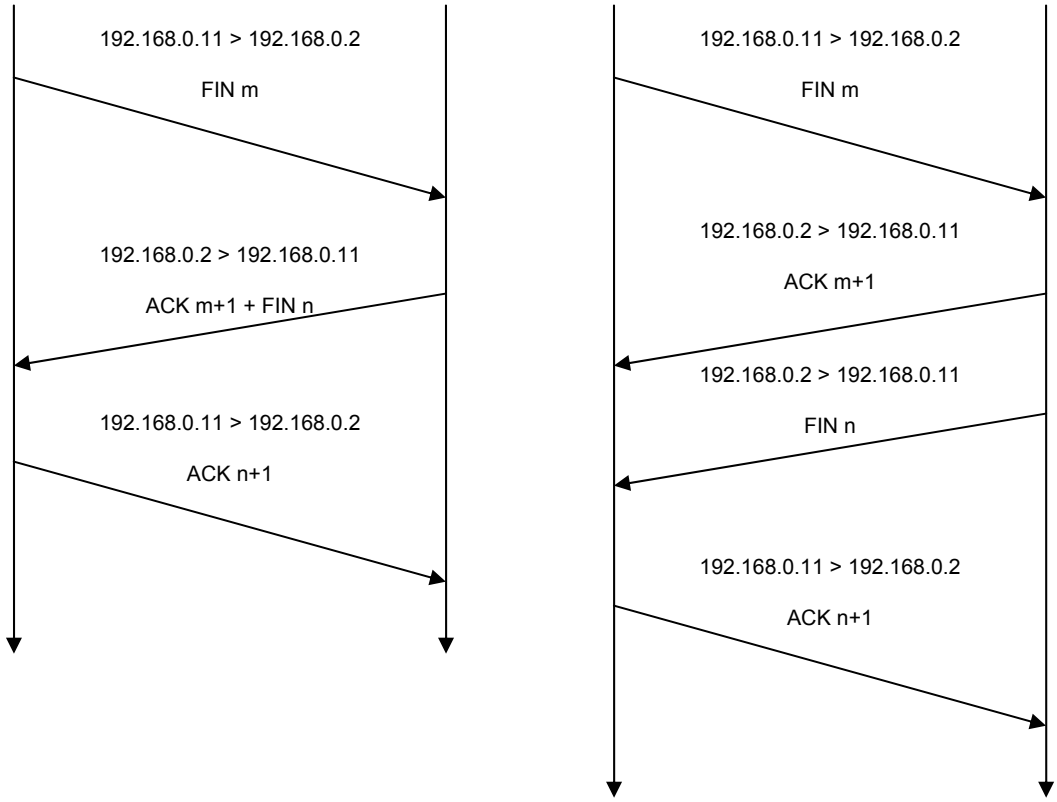


Abb. 2: Die RST- und die FIN-Methode zum Beenden einer TCP-Sitzung

Einige Portscanner - vorwiegend unter Windows oder solche, die mit Skripten realisiert wurden (siehe das folgende Beispiel-Skript) - beenden die Verbindung mit der sogenannt höflichen Methode. Diese benötigt im Schnitt 300 % mehr Pakete weder die abrupte Methode - Sie muss also unweigerlich als höchst ineffizient betrachtet werden. Vor allem, wenn sehr viele Hosts und TCP-Ports überprüft werden sollen, können sich hinter dieser Prozentzahl eine beachtliche Anzahl Segmente verbergen. Die höfliche Methode funktioniert so, dass der Client ein TCP-Segment mit gesetzter FIN-Flagge gesetzt. Dadurch signalisiert er dem Server, das er die etablierte TCP-Sitzung beenden mögliche. Dieses active close wird im Normalfall durch den Server bestätigt, indem er ein TCP-Segment mit gesetzter ACK-Flagge zurückschickt. TCP-Verbindungen gelten als bidirektional. Das bedeutet, dass die Verbindung in die Richtung vom Server zum Client auch noch beendet werden darf. Es schickt also nun auch noch der Server ein TCP-Segment mit gesetzter FIN-Flagge an den Client. Dieser muss wiederum die höfliche Bitte seines Gegenübers mit

einem TCP-Segment mit gesetzter ACK-Flagge bestätigen.

```
#!/bin/sh

[ -z ${3} ] && {
    echo usage: netcatscanner.sh [host] [low port] [high port]
    exit 1
}

i=${2}

while [ ${i} -le ${3} ]
do
    nc -z ${1} ${i} >& /dev/null
    [ $? -eq 0 ] && echo port ${i} open
    let i=i+1
done
```

Nmap, und viele andere Portscanner, bieten eine Fülle von zusätzlichen Scanning- und Auswertungsmöglichkeiten. Schon alleine diese in diesem Artikel kurz zu skizzieren würde den Rahmen des gesamten Magazins sprengen. Die sowohl in Englisch als auch in Deutsch erhältliche man-Page von nmap listet die wichtigsten Parameter und ihre Auswirkungen auf. In den Büchern "Hacking Intern" und "Das Anti-Hacker-Buch" finden sich zusätzlich ausgiebige Erklärungen zu Portscanning und nmap.

Erkennen der Server-Anwendung
Konnten Ports als offen und ansprechbar identifiziert werden, gilt es den angebotenen Dienst zu identifizieren. Die IANA gibt periodisch eine Liste heraus, die als Empfehlung für die Vergabe der Portnummern angesehen wird. So sollte sich nach dieser Portliste auf dem Port 23 der TELNET-Dienst finden. Port 25 ist für SMTP (Simple Mail Transfer Protocol, RFC 821) und Port 80 für HTTP (Hypertext Transfer Protocol) vorgesehen. Wir können also anhand der Ausgabe des Portscanners vermuten, um was für einen Dienst es sich handelt.

Diese Liste ist jedoch nur eine Empfehlung. Genauso wie bei den RFCs haben sich weder die Administratoren noch die Entwickler oder Anwender an diese Vorgaben zu halten. So ist es nicht selten gesehen, dass ebenso auf dem TCP-Port 81 ein HTTP-Dienst angeboten wird. Zum Beispiel, um mit demgleichen Hostnamen auf einem anderen Port ein anderes Webangebot bereitzustellen. Oder oft finden sich auch auf diesen Ports Administrations-Schnittstellen für irgendwelche Dienste. Ähnliches ist auf den Ports 82, 800 und 888 zu beobachten.

Die Gruppe THC (The Hackers Choice) brachte eine Anwendung namens amap (Application Mapper) - in Anlehnung an das populäre Scanning- und Auswertungstool nmap von Fyodor - heraus. Diese baut eine Verbindung zu einem oder mehreren Zielports auf, schickt irgendwelche Anfragen und versucht anhand der Rückgaben das eingesetzte Protokoll zu erkennen. Dies ist sehr gut für automatisierte Prozesse, bei denen man mehrere Hosts und Ports überprüfen sollte.

```
maru@debian~$ amap -s T  www.scip.ch 80
Total amount of tasks to perform: 11
Amap v0.95 started at Thu Jul  3 11:25:05 2003, stand back and keep the children away.
Protocol on IP 192.168.0.2 port 80 tcp matches HTTP
Unidentified ports: None.
Amap v0.95 ended at Thu Jul  3 11:25:10 2003
```

Möchte man diese Überprüfung manuell durchführen, kann man sich einer Terminal-Emulation, wie zum Beispiel TELNET oder NetCat, bedienen. Sodann baut man eine Verbindung zu einem ansprechbaren Port auf und setzt irgendwelche Anfragen ab. Kennt man sich mit den verschiedenen Protokollen der Anwendungsschicht genug gut aus, kann man anhand des Verhaltens und der Rückgabe des Servers das angebotene Protokoll erkennen. Man macht schlussendlich genau das gleiche, was auch amap macht. Wie man mit den jeweiligen Servern zu sprechen hat, liest man am besten in den jeweiligen RFCs nach.

Identifizieren der Server-Implementierung
Die meisten interaktiven Netzwerkapplikationen der Anwendungsschicht begrüssen den Benutzer zu einer etablierten Sitzung mit einer kurzen Statusmeldung. In dieser wird meistens auch der Name und die Versionsnummer des Daemons sowie die Plattform, der Name und die Version des Betriebssystems mitgeschickt. Diese Information ist sehr wichtig, denn anhand dieser Daten können weitere Zugriffe koordiniert und spezifische Schwachstellen gesucht und überprüft werden.

Diese Auswertungs-Zugriffe werden sowohl in der deutschen als auch in der englischen Literatur als Banner-Grabbing bezeichnet. Das Abgreifen des Banners ist wiederum sehr einfach mit einer Terminal-Emulation durchzuführen. Bei den meisten interaktiven Diensten reicht das etablieren einer Sitzung. Dies kann zum Beispiel durch die Eingabe von "telnet www.scip.ch 21" für den TCP-Port 21 (normalerweise FTP) auf dem Host mt dem Namen www.scip.ch durchgeführt werden. Sobald die Verbindung hergestellt werden konnte, begrüsst uns der FTP-Server mit seinem Willkommens-Banner. Um die FTP-Sitzung erfolgreich weiterzuführen, wäre die Authentifizierung mittels Benutzername (USER) und Passwort (PASS) notwendig. Wir sehen jedoch davon ab und beenden die Verbindung mit der Eingabe des Befehls "QUIT". Die Verbindung wird dann auf mit der höflichen FIN-Methode beendet.

```
mruef@debian~$ telnet www.scip.ch 21
Trying 192.168.0.2...
Connected to www.scip.ch.
Escape character is '^]'.
220 ProFTPD 1.1.3rc2 Server (Debian) [www.scip.ch]
QUIT
221 Goodbye.
Connection closed by foreign host.
```

Einige Server-Anwendungen wollen jedoch provoziert werden. Manche geben erst nach einer kurzen Aufforderung die gewünschten Informationen heraus. Bestes Beispiel hierfür HTTP. Nach etablierter Verbindung bleibt der HTTP-Daemon normalerweise Stumm und wartet auf die HTTP-Anfrage des Clients. Sehr beliebt ist in diesem Zusammenhang das Abetzen einer einer HEAD-Anfrage im Rahmen der HTTP-Sitzung. Dabei wird nach etablierter Sitzung mit dem HTTP-Port die Anfrage "HEAD / HTTP/1.0" übergeben. Der Server sollte sodann die Kopfdaten ohne den Inhalt zurückliefern. Vorteil dieses Zugriffs ist, dass nicht unnötig irgendwelche HTTP-Nutzdaten übertragen werden müssen. Der Zugriff erfolgt also schnell, effizient und unkompliziert. Nachteil ist jedoch, dass viele Server oder Administratoren diesen Zugriff, der wirklich nur für Auswertungen genutzt wird, nicht zu. So erhält man eine knappe Fehlermeldung, dass dieses Kommando nicht unterstützt sei. Sodann muss man auf die klassische GET-Anfrage mit all ihren Nachteilen zurückgreifen.

Erkennen des Betriebssystems

Der nächste Schritt ist das Erkennen des Betriebssystems. Dies ist in erster Linie nur dann erforderlich, wenn diese Information dem Auditoren bis dato nicht zugekommen lassen wurde. Der Nutzen dieses Tests ist, dass das weitere Vorgehen optimiert werden kann. So werden Linux-Hosts beispielsweise auf ganz andere Schwachstellen überprüft, weder die Windows-Betriebssystemreihe.

Um das auf einem Host eingesetzte Betriebssystem zu erkennen, können verschiedene Techniken eingesetzt werden. Da wir schon mit der Hilfe eines Portscans die Stati der verschiedenen Ports identifiziert haben, können wir eventuell Rückschlüsse auf das eingesetzte Betriebssystem ziehen. Die offenen Ports 135, 137, 138 und 139 sind typisch für ein Windows-Betriebssystem. Entdecken wir zusätzlich den offenen Port 445, verbirgt sich dahinter sehr wahrscheinlich ein Windows 2000 oder XP. Finden wir die Ports 23, 25, ... offen, verbirgt sich dahinter mit grösster Wahrscheinlichkeit eine Symantec Raptor-Firewall.

Eine etwas zuverlässigere, jedoch zugleich aufwendigere Methode ist im sogenannten OS-Fingerprinting gegeben. Bei dieser Methode werden die Eigenschaften der jeweiligen TCP/IP-Implementierungen zur Determinierung des eingesetzten Betriebssystems herangezogen. Aus diesem Grund wird diese Methode auch TCP/IP- oder Stack-Fingerprinting genannt.

Man unterscheidet zwischen aktivem und passivem OS-Fingerprinting. Bei der aktiven Variante werden durch bestimmte Reize die gewünschten Reaktionen provoziert. Sie benötigt ein Mehr an Aufwand und Ressourcen. Ihr Vorteil ist jedoch, dass nicht auf durch andere Systeme generierten Verkehr mit dem Zielhost gewartet werden muss. Auss diesem Grund wollen wir uns an dieser Stelle nur mit den Tools für aktives Stack-Fingerpringin beschäftigen.

Die Herkunft des OS-Fingerprintings ist nicht ganz klar. Eines der ersten Tools, dass die verschiedenen Charakteristika einer TCP/IP-Kommunikation zur Identifizierung war das für Linux entwickelte Queso. Nmap liess sich jedoch nicht lange bitten und wartete bald auch mit einer solchen Funktion auf. Diese kann durch das Miteinbeziehen des Parameters -O (das O steht für OS-Fingerprinting) aktiviert werden; wahlweise mit einem Portscan oder als eigenständigen Auswertungs-Zugriff.

```
debian:~# nmap -sT -p 80,81 -O www.scip.ch

Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-07-03 11:29 CEST
Interesting ports on www.scip.ch (192.168.0.2):
(The 1 port scanned but not shown below is in state: closed)
Port       State       Service
80/tcp     open        http
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
Uptime 247.936 days (since Mon Oct 28 12:01:54 2002)

Nmap run completed -- 1 IP address (1 host up) scanned in 5.933 seconds
```

Nmap führt beim OS-Fingerprinting mehrere verschiedene Zugriffe durch, deren Reaktion der Gegenstelle das eingesetzte Betriebssystem verraten soll. Dabei wird eine sehr geringe Zahl an Pakete generiert - Nur wenige Intrusion Detection-Systeme sind in der Lage diesen Zugriff als solches zu identifizieren.

Sicherheitslücken identifizieren

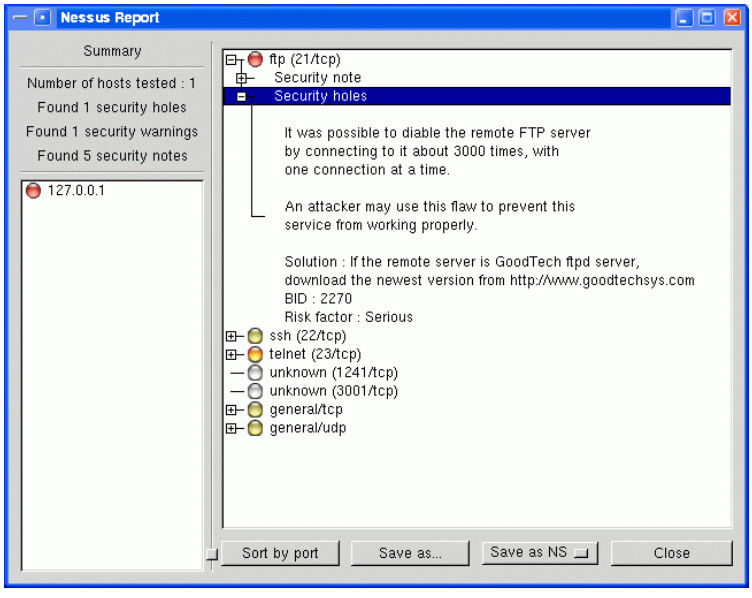
Man muss zwischen dem Finden und dem Identifizieren von Sicherheitslücken unterscheiden. Die verschiedenen Techniken für das Finden neuer Schwachstellen und die entsprechende Vorgehensweise hier zusammenzufassen, würde den Umfang bei weitem sprengen. Wir beschränken uns daher auf das Suchen altbekannter Schwachstellen. Dies ist wirtschaftlicher, da es weniger Zeit und Aufwand in erfordert.

Konnte die auf einem Host eingesetzte Software (Anwendungen und Betriebssystem) identifiziert werden, können in verschiedenen Verwundbarkeits-Datenbanken die Schwachstellen dazu gesucht werden. Die bekannteste und umfassendste davon wird auf SecurityFocus.com bereitgestellt. Eine deutschsprachige Datenbank mit Sicherheitslücken wird von der Firma scip AG unter http://www.scip.ch/cgi-bin/smss/showadvf.pl publiziert. Durch verschiedene Such-Methoden lässt sich anhand des Herstellers, des Produktnamens und der Versionsnummer die Anzahl der Schwachstellen ermitteln. Jenachdem werden Informationen geliefert, wie Sicherheitslücke ausgenutzt werden kann und welche Gegenmassnahmen es gibt.

Automatisiertes Scanning

Besonders im grossen Umfeld macht es wenig Sinn, das Scanning manuell mit den zuvor beschriebenen Zugriffen durchzuführen. Gilt es ein ganzes Netzwerk systematisch nach Schwachstellen abzusuchen, kommen sogenannte Security Scanner zum zug. Diese Software vereinigt die bekannten Methoden zur Auswertung von Systemen, um anhand der gesammelten Informationen die potentiellen oder existenten Schwachstellen auszuweisen.

Es gibt eine Vielzahl verschiedener Vulnerability Scanner. Zum Beispiel ist mit der Freeware LANguard sehr einfach ein kleines Netzwerk abgescannt. Kommerzielle Lösungen wie ISS Internet Scanner oder Symantec NetRecon kommen da schon ein bisschen professioneller daher. Durch die vorgefertigten Reports lässt sich sehr schnell und unkompliziert das Problem Schwarz auf Weiss nachlesen.



Die mitunter populärste Security Scanner Lösung wurde für Linux entwickelt und nennt sich Nessus. Das open-source Projekt basiert auf dem Client/Server-Prinzip, bei dem auf einem Host der Nessus-Daemon installiert wird. Mit diesem verbindet sich der Nessus-Client, um ihn anzuweisen, welche Scans in welcher Form durchgeführt werden sollen. Der Vorteil dieses Ansatzes ist, dass verschiedene Clients unabhängig voneinander platziert den gleichen Server nutzen können.

Ein wichtiges Merkmal eines guten Vulnerability Scanners ist die Anzahl und Aktualität der durchzuführenden Checks. Da es sich bei Nessus um ein

sehr populäres open-source Projekt handelt, werden praktisch täglich neue Plugins nachgereicht, die sich sehr einfach installieren lassen. Ein anderer wichtiger Punkt bei Security Scannern ist die Qualität und der Umfang der am Schluss generierten Reports. Da in diesen die gefundenen Schwachstellen und die empfohlenen Gegenmassnahmen festgehalten werden, ist es unabdingbar, dass diese aktuell und leicht nachvollziehbar sind. Auch hier hat Nessus in den letzten Jahren gewaltige Sprünge nach Vorne gemacht.

Reporting
Ein wichtiger Schritt beim Security Auditing, der gerne unterschätzt oder gänzlich vergessen wird, ist das Reporting. In dieser Phase, nach dem Abschluss der jeweiligen Scans, werden die Resultate zusammengetragen und dokumentiert. Dieses Papier ist sodann die Grundlage für das Einleiten und Umsetzen der entsprechenden Gegenmassnahmen, um die Sicherheit der Umgebung zu wahren und zu stärken.

Wie der Audit Report aufgebaut ist und in welchem Umfang er daherkommt, ist den individuellen Wünschen anzupassen. Ein fünfstufiger Aufbau hat sich jedoch bewährt. In diesem wird an erster Stelle der Auftraggeber, der Auftragnehmer und die desweiteren involvierten Parteien aufgelistet. Dadurch kann auch später noch ausgemacht werden, wer für welche Punkte zuständig war. Im zweiten Teil wird in groben Zügen eine nicht-technische Zusammenfassung des Zustands durch das Management Summary vorgetragen. Dadurch kann man sich schnell und ohne tiefschürfende Fachkenntnisse über die Lage informieren. Vor allem die Entscheidungsträger heissen diesen Teil willkommen. Desweiteren sollte die Vorgehensweise bei der Überprüfung dokumentiert werden. Dadurch kann bei einem zweiten Audit aus vergangenen Fehlern gelernt oder sich auf ältere Daten gestützt werden. Ein Grossteil des Reports macht das Auflisten sämtlicher konzeptioneller und technischen Schwachstellen aus. Dabei sollten Lösungsvorschläge unterbreitet werden, die auf die jeweilige Umgebung angewendet werden können. Die Ausgaben der Scans und die durch den Computer generierten Reports runden das Dokument ab. Anhand derer können Verifikationen durchgeführt oder zusätzliche Informationen gefunden werden.

Gegenmassnahmen umsetzen
Wir haben gesehen, dass das höchste Ziel eines Security Audits das Wahren und Stärken der Sicherheit eines Systems ist. So ist es unabdingbar, die durch das Assessment aufgedeckten Schwachstellen zu beheben. Diese Gegenmassnahmen können meistens entweder auf technischer oder auf konzeptioneller bzw. organisatorischer Ebene angesetzt werden. Es bleibt den Entscheidungsträgern überlassen, wie ein Problem gelöst werden soll. Dies kann je nach Problem und Umgebung anders ausfallen.

Fazit
Das erfolgreiche Umsetzen von Security Audits ist in den meisten Fällen eine komplizierte und nervenaufreibende Sache. Um Problemen aus dem Weg zu gehen, sollten Vorabklärungen getätigt und der gesamte teschnische Teil vorbereitet werden. Sind alle Hindernisse aus dem Weg geschafft worden, gilt es auf technischer Ebene die Schwachstellen eines Systems herauszufinden. Die Vorgehensweise bleibt dabei dem Auditoren überlassen und richtet sich in erster Linie nach den Wünschen des Auftraggebers. Jenachdem kann das Assessment intensiver oder breitflächiger ausfallen. Schwachstellen lassen sich sehr gut durch manuelle Zugriffe (Auswertung und Attacke) aufspüren. Dies ist jedoch in den meisten Fällen nicht wirtschaftlich genug, so dass ein Security Scanner herangezogen werden sollte. Durch diesen kann eine Umgebung automatisiert nach etwaigen Schwachstellen abgesucht werden.

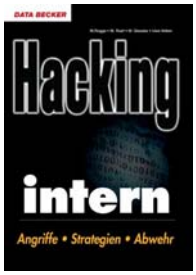
Konnte der technische Teil erfolgreich durchgeführt und gar einige konzeptionelle oder technische Schwachstellen entdeckt werden, gilt es diese in einem Report zu dokumentieren. Dieses Papier wird die Grundlage für das weitere Vorgehen, das Beheben der Schwachstellen oder Durchführen weiterer Tests, sein. Wichtig ist, dass das Beheben von Sicherheitslücken ein unabdingbarer Bestandteil eines Security Audits ist. Dieser alleine ist nämlich wertlos, da dadurch lediglich der Stand der Sicherheit eines Systems dokumentiert werden kann – Die Sicherheit ansich ist damit jedoch noch lange nicht gewährleistet.

Dazu gehört auch, dass ein System einem stetigen Wandel unterworfen ist. Neue Betriebssysteme oder Software können ganz neue Probleme schaffen. Es ist also wichtig den Security Audit in regelmässigen Abständen – zum Beispiel alle paar Monate – zu wiederholen, um den Stand der Dinge zu erfassen und Trends festzustellen. Sicherheit ist ein Prozess und kein Zustand.

Links
scip AG - Durchführung von Security Audits
http://www.scip.ch

Computec – Computer, Technik und Security
http://www.computec.ch

Kryptocrew – Portal zu Computersicherheit
http://www.kryptocrew.de



Über den Autoren

Marc Ruef arbeitet als Security Consultant bei der Firma scip AG (www. scip.ch) in Zürich. Neben dem Umsetzen von Vulnerability Assessments ist er dort unter anderem auch für das Durchführen von Schulungen zum Thema Computersicherheit zuständig. Im September letzten Jahrs ist sein Buch mit dem Titel „Hacking Intern“ im Data Becker Verlag erschienen.