

Open-Source-Tools prüfen IT-Sicherheit

Viele Werkzeuge für Security-Tests stehen quelloffen kostenlos zur Verfügung. Allerdings gilt es, die Spreu vom Weizen zu trennen, die Programme richtig einzusetzen und die Resultate korrekt auszuwerten.

VON SIMON WEPFER*

Sicherheitstests sind eine noch junge IT-Disziplin. Normen wie ISO 17799 oder das Grundschutzhandbuch des Bundesamts für Sicherheit in der Informationstechnik (BSI) beschreiben zwar, wie der Schutz der Unternehmens-IT zu erhöhen ist, gehen jedoch kaum auf die praktischen Aspekte von Security-Tests ein. Diese Lücke kann das „Open Source Security Testing Methodology Manual“ (OSSTMM) schließen: Dieses Dokument stellt einen praktischen Leitfaden für Sicherheitstests dar und ist zu den gängigen Normen kompatibel.

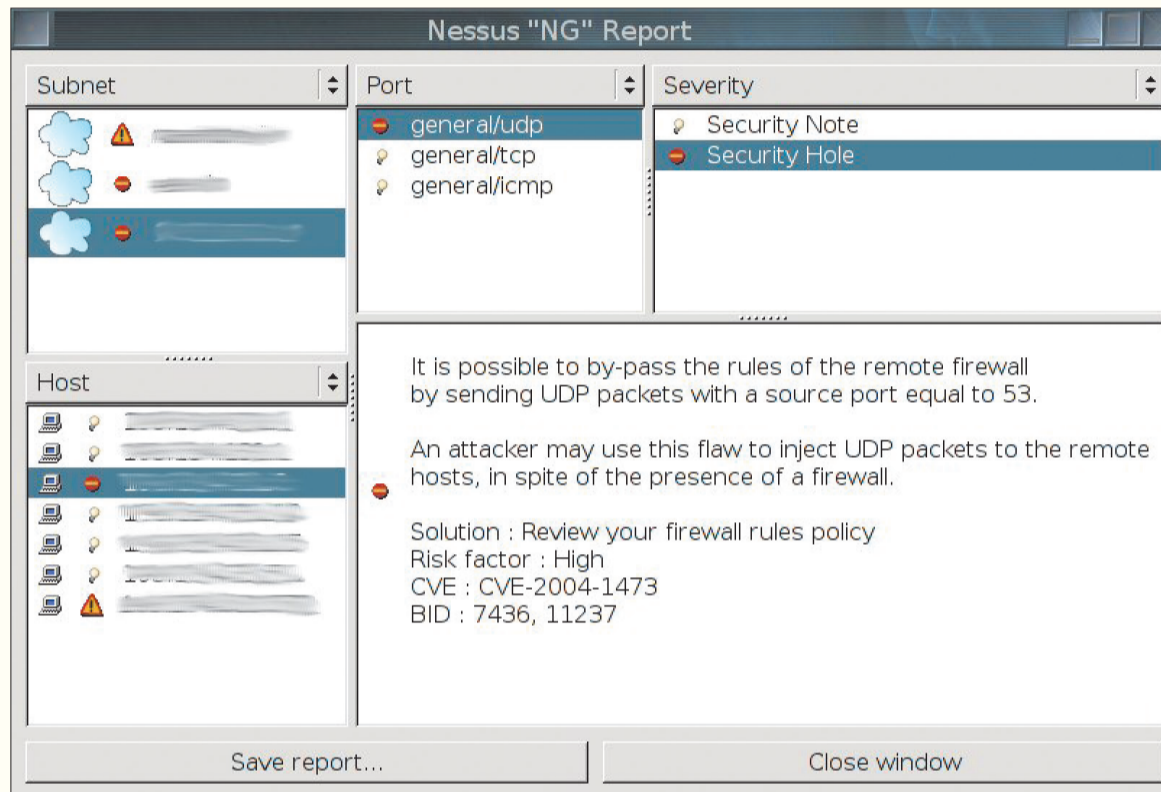
Hier lesen Sie ...

- ◆ was bei Sicherheitstests zu beachten ist;
- ◆ wie Ergebnisse bewertet werden;
- ◆ welche ethischen Grundsätze gelten;
- ◆ über die wichtigsten Open-Source-Tools;
- ◆ wie damit Systeme und Dienste erkannt werden;
- ◆ was TCP-Sequenznummern und Zeitstempel aussagen;
- ◆ die Vor- und Nachteile von automatisierten Tests.

Wie der Name vermuten lässt, ist die Anleitung offen für Verbesserungsvorschläge. Viele Erfahrungen von Security-Testern aus aller Welt sind bereits eingeflossen. Das Handbuch besteht aus der Methodik selbst sowie einigen Vorlagen für die Dokumentation der Testergebnisse und ist unter www.osstmm.org/ verfügbar.

Sicherheitsniveau evaluieren

Das OSSTMM befasst sich auch mit der objektiven Bewertung des Sicherheitsniveaus. Das Management des Unternehmens kann und muss nicht in der Lage sein, eine Liste von gefundenen Sicherheitslücken zu interpretieren. Es interessiert nur, wie sicher die Infrastruktur letztlich ist. Damit die Tester verbindliche Aussagen hierzu machen können, liefert die Vorgehensweise nach OSSTMM einen „Risk Assessment Value“ (RAV): Die Überprüfung der in dem Doku-



Nach Abschluss eines Sicherheits-Scans listet Nessus die ermittelten Schwachstellen in einem Report auf.

ment genannten Punkte ergibt einen Zahlenwert. Je näher dieser bei 100 liegt, desto sicherer ist das Untersuchungsobjekt. Der RAV berechnet sich im Grunde aus drei Eingangsvariablen: dem Aufbau der Infrastruktur (Sichtbarkeit, Anzahl Systeme, Trusts), den zusätzlich getroffenen Sicherheitsmaßnahmen (zum Beispiel Redundanz, Verschlüsselung, Benutzerfreundlichkeit) sowie den festgestellten Sicherheitslücken.

Die Risiken werden in fünf Kategorien eingeteilt: Verwundbarkeit, Schwachstelle, Bedenken, Informationsabfluss und Anomalie. Was darunter jeweils zu verstehen ist, erläutern nachfolgend ein paar Beispiele. Bei einer Verwundbarkeit kann es sich um eine Anfälligkeit für einen Buffer Overflow handeln, die es dem Angreifer ermöglicht, beliebigen Code auf dem System auszuführen. Eine Schwachstelle stellt im Gegensatz dazu keine direkte Bedrohung dar, sondern einen Fehler im Sicherheitsmechanismus. Ein typisches Beispiel findet sich in Web-Applikationen, welche beim erfolgreichen Einloggen verraten, ob der Benutzername oder das Passwort falsch waren. So kann ein Angreifer erfahren, welche Benutzerkonten gültig sind.

Bedenken werden formuliert, wenn die Konfiguration oder Architektur nicht den Best Practices entspricht. Ein Informationsabfluss gibt Informationen über die Firma, den Betreiber oder das interne Netzwerk preis. Ein klassisches Beispiel hierfür ist die interne IP-Adresse des Mail-Servers, welche im E-Mail-Header eingetragen wird. Unter Anomalien schließlich werden sämtliche Phänomene erfasst, die der Tester mit den ihm zur Verfügung stehenden Informationen innerhalb der Projektzeit erklären konnte. Hierzu zählt etwa eine nicht erwartete Antwort eines Routers.

Ethik

Neben dem technischen Vorgehen, dem RAV und den Dokumentationsvorlagen werden im OSSTMM auch ethische Richt-

linien festgehalten. Die Tester verpflichten sich, ausschließlich die Komponenten zu untersuchen, die mit dem Kunden vereinbart wurden, und sich dabei an die vorgeschriebenen Rahmenbedingungen zu halten. Security-Tester, die sich an solche Richtlinien halten, werden gerne als „Ethical Hacker“ bezeichnet, obwohl sich die ursprüngliche Hacker-Ethik damit nicht deckt.

Tools

Für technische Sicherheitsüberprüfungen benötigt der Tester diverse Tools. Erste Wahl in Sachen Betriebssystem ist Linux. Zwar sind einige der zusätzlich notwendigen Tools wie „nmap“ auch für Windows verfügbar, doch wer ernsthaftes Security-Testing betreiben will, kommt an der Open-Source-Plattform nicht vorbei. Da bestimmte Tests (zum Beispiel im LAN oder Denial-of-Service-Tests) vor Ort betrieben werden, ist ein Notebook von Vorteil. In der Praxis hat sich ein Dual-Boot-System mit Windows bewährt.

Bei der Auswahl der Linux-Distribution sollte unter anderem das verwendete Paketierungssystem beachtet werden. Pakete im RPM-Format (Red Hat Package Manager), wie sie zum Beispiel von Suse oder eben Red Hat an-

geboden werden, hinken zeitlich stark hinterher. Um auf dem aktuellen Stand zu bleiben und Probleme mit Programmibliotheken zu verringern, sollte man sich für ein Paketierungssystem entscheiden, welches direkt mit Quellcodes arbeitet und lokal kompiliert. Gentoo, FreeBSD, OpenBSD und Debian sind daher beliebte Testsysteme. Wer nicht gleich seine Festplatte umpartitionieren und nur mal einen Einblick in Linux-Tools gewinnen möchte, der kann zu Alternativen wie zum Beispiel „Knoppix“ greifen, die sich von CD-ROM starten lassen.

Sind diese Vorarbeiten erledigt, benötigt man lediglich einige wenige Tools, um einen Basistest zu starten (siehe Kasten „Das Basissystem: Weniger ist oft mehr“). Die Technik allein reicht jedoch nicht aus: Letztlich hängt die Qualität der Ergebnisse stark vom Wissen und der Erfahrung des Testers ab. Die Resultate sowie der Netzwerkverkehr müssen korrekt interpretiert werden. Praktische Erfahrungen in systemnaher Programmierung und Netzwerken sind hierfür unabdingbar, Erfahrungen in der Systemadministration zudem von Vorteil. Empfehlenswert ist daneben ein Schuss Kreativität, gemischt mit einer Portion Neugier.

Netzwerkanschluss

Ist das Testsystem einsatzbereit, kommt es darauf an, es am richtigen Netzknoten anzuschließen: Als Regel gilt, dass Sicherheitstests ohne filternde Komponenten auskommen sollten. Aus diesem Grund ist das Testsystem – vom Internet her gesehen – vor der Firewall zu positionieren. Ein NAT-Router (Network Address Translation) darf nicht dazwischengeschaltet sein, außerdem sind Software-Firewalls auf dem Testsystem zu deaktivieren. Damit der Tester nicht selbst Ziel eines Angriffs wird, sollte er sein System zunächst härten, indem er alle unbenötigten Dienste deaktiviert und sein Gerät mit denselben Tools überprüft.

Monitoring

Sobald das Notebook korrekt konfiguriert und die Netzwerkverbindung getestet ist, sollte der Netzwerkverkehr mit dem folgenden Befehl im Hintergrund mitgeschnitten werden:

Mehr zum Thema

www.computerwoche.de/go/525557:

Penetrationstests: Einbruch auf Bestellung;

567499: Nessus sucht Schwachstellen;

557538: BOSS: Freie Software für netzweite Sicherheitsprüfungen.

```
# tcpdump -nw file.dmp net
192.168.1.0/24 &
```

Damit auch der Tester die Kommunikation in Echtzeit beobachten kann, muss tcpdump erneut gestartet werden, zum Beispiel:

```
# tcpdump -nwvs0
```

System-Enumeration

Ist als Untersuchungsobjekt ein IP-Adressbereich definiert, sollte der Tester zunächst herausfinden, welche IP-Adressen tatsächlich belegt sind. Dazu eignet sich das Kommando

```
# whois <ip-adresse>
```

Durch DNS-Lookups auf IP-Adressen können bereits aktive Adressen und Domänen identifiziert werden:

```
# host <ip-adresse>
```

Bei großen Adressbereichen ist die Verwendung eines Perl- oder Shell-Scripts sinnvoll.

Durch die Abfrage der DNS Server findet man den Mail-Server:

```
# dig domain.com MX
```

Eventuell kann man durch einen Zonentransfer die gesamte Zone einer Domäne herunterladen:

```
# dig @nameserver -t AXFR
domain.com
```

Falls dies gelingt, hat der Tester bereits einen ersten Informationsabfluss entdeckt.

Hilfreich ist zudem der „Ping“-Befehl – dieser wird wahrschein-

Eine Hand voll Tools reicht aus, um ein System auf Schwachstellen zu testen.

lich an der Firewall scheitern. Der Tester sollte dabei nicht vergessen, auch die Netzwerk- und Broadcast-Adresse anzupingen. Ein automatisierter Ping-Scan kann auch mittels nmap erfolgen:

```
# nmap -sP 192.168.1.0/24
```

OSSTMM verwendet daneben weitere Techniken wie den ACK-Scan zur System-Enumeration. ACK bezieht sich auf das gesetzte Acknowledge-Flag innerhalb eines TCP-Datenpakets. Ein Ack-Scan mit Source-Port 80 auf die Ziel-Ports 3100 bis 3150 lässt sich auf verschiedene Arten starten:

```
# unicornscan -B 80 -mTsA
192.168.1.1:3100-3150
```

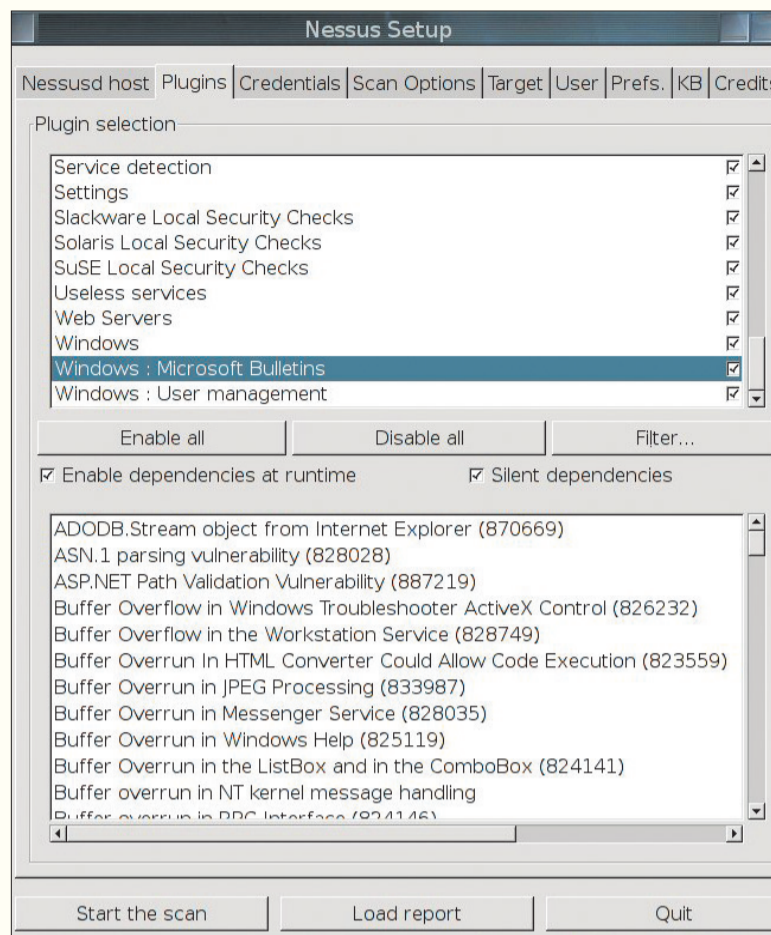
```
# hping2 -s 80 -A --scan 3100-3150
192.168.1.0
```

oder

```
# nmap -sA -n -PO -p 3100-3150
192.168.1.0
```

Portscan

Beim Portscan versucht der Tester herauszufinden, welche Dienste auf einem System erreichbar sind. Aus Netzsicht kann ein Port offen, geschlossen oder gefiltert sein. Ein offener TCP-Port beantwortet eine Verbindungsanfrage (SYN) mit dem Senden eines SYN-ACK-Pakets. Wird dieses Paket vom Client erneut bestätigt, ist der so genann-



In Form von Plug-ins stellt der Vulnerability-Scanner Nessus Tausende von speziellen Tests zur Verfügung.

te Dreibege-Handshake vollendet: Die TCP-Verbindung steht so lange, bis eine Seite sie abbricht. Ein geschlossener Port hingegen antwortet auf die SYN-Anfrage mit einem Reset-Paket (RST). Ein gefilterter (zum Beispiel durch die Firewall geschützter) Port verwirft die Anfrage.

Problematischer sind UDP-Ports (User Datagram Protocol). Da dieses Protokoll verbindungslos ist, werden die Anfragen „auf gut Glück“ abgeschickt. Das Internet Control Message Protocol (ICMP) kann zwar den Status von UDP-Paketen überwachen, jedoch wird davon kaum Gebrauch gemacht. Bei UDP muss also eine Antwort auf Applikationsebene vom Testsystem erzwungen werden, um Aussagen über den Status machen zu können. Folgende Kommandos führen einen komplet-

ten TCP-SYN-Scan mit nmap sowie einen UDP-Scan mit unicornscan aus:

```
# nmap -sS -PO -n -p 0-65535
192.168.1.1
```

```
# unicornscan -mU 192.168.1.1:all
```

Erkennung der Dienste

Viele Dienste laufen auf Standard-Ports. So ist zu erwarten, dass hinter dem Port 22 ein SSH-

Die Qualität der Ergebnisse hängt letztlich stark vom Wissen des Testers ab.

Dienst (Secure Shell) horcht. Eine Liste mit bekannten Ports findet sich beispielsweise unter www.oneconsult.com/downloads/ports.html. Nmap unterstützt mit der Option „-A“ das aggress-

sive Erkennen von Diensten. Spezial-Tools wie „amap“ (<http://www.thc.org>) erkennen Dienste ebenfalls. Mit „netcat“ oder „telnet“ kann sich der Tester manuell mit dem Dienst verbinden, um mehr über ihn zu erfahren. Es folgt ein Verbindungsversuch zu einem SSH-Dienst:

```
# nc 192.168.1.20 22
SSH-2.0-OpenSSH_3.9p1
```

Die Antwort lässt erkennen, welche SSH-Protokolle und -Produkte eingesetzt werden. Beantwortet der Tester nun den Banner korrekt, erhält er weitere Informationen über die aktiven Algorithmen und Hash-Funktionen.

Betriebssystem identifizieren

Einer der Punkte, in denen sich Betriebssysteme voneinander unterscheiden, ist das Kommunikationsverhalten. So lässt sich durch die Auswertung der Reaktion auf bestimmte Anfragen (beispielsweise ein FIN-Paket auf einen geschlossenen TCP-Port) auf das installierte Betriebssystem sowie dessen Version schließen. Diese Technik wird als „OS-Fingerprinting“ bezeichnet. Der Port-Scanner Nmap unterstützt die Funktion mit dem Schalter „-O“, benötigt jedoch mindestens einen offenen sowie einen geschlossenen Port, was bei durch Firewalls geschützten Systemen fehlschlägt. Das Tool „Xprobe“ hingegen begnügt sich mit einem geschlossenen oder einem offenen Port. Für ein OS-Fingerprinting bei offenem TCP-Port 22 genügt folgendes Kommando:

```
# xprobe2 -p tcp:22:open
192.168.1.20
```

TCP-Sequenznummern und Timestamps

Wenn sich TCP-Sequenznummern voraussagen lassen, kann eine bestehende Verbindung gekapert oder unterbrochen werden. Daher ist es wichtig, dass ein System pseudozufällige Zahlenwerte nutzt. Mit der Option „-vv“ versucht Nmap, die Sequenznummern sowie die TCP-Timestamps auszuwerten. Noch besser funktioniert dies

Fazit

Bereits **eine Hand voll Tools** reicht aus, um ein System auf Sicherheitslücken zu testen. Neben den erwähnten Programmen existieren noch **Hundert weitere Speziallösungen**, welche auf bestimmte Dienste ausgerichtet sind. Diese zu finden, zu testen und korrekt einzusetzen gehört ebenfalls zu den **Aufgaben eines Security-Testers**. Dessen Arbeit ist mit den Sicherheitsüberprüfungen allein noch nicht getan. Sämtliche **Resultate** müssen **validiert und dokumentiert** werden. Ein Sicherheitstest sollte schließlich in einem kompletten **Report** enden, dessen Management-Summary auch Leser mit geringem Technikwissen verstehen können.

mit „hping2“. Unterstützt ein System die TCP-Timestamps und stellt den Zähler nicht regelmäßig zurück, lässt sich die Zeit seit dem letzten Neustart auslesen. Dies scheint auf den ersten Blick nicht so tragisch, doch da Patches oder Firmware-Updates oft einen Neustart erzwingen, könnte ein Angreifer mit dieser Information auf den Patch-Level des Systems schließen und so seinen Angriff besser planen:

```
# hping2 -S 192.168.1.20 -p 22 --tcp-timestamp
HPING 192.168.1.20 (eth0
192.168.1.20): S set, 40 headers + 0
data bytes
len=56 ip=92.168.1.20 ttl=52 DF
id=6211 sport=22 flags=SA seq=0
win=33012 rtt=3388.5 ms
TCP timestamp: tcpts=34284190
HZ seems hz=100
System uptime seems: 3 days,
23 hours, 14 minutes, 1 seconds
```

Vulnerability Scanner

Für umfangreiche Schwachstellen-Scans können Anwender auf das Tool „Nessus“ zurückgreifen. Dabei handelt es sich um einen klassischen Vulnerability-Scanner, dessen Server-Komponente unter Linux läuft und die Tests vornimmt. Zur Steuerung des Servers wird eine Client-Komponente verwendet, welche auch für Windows-Systeme zur Verfügung steht. Die einzelnen Tests sind in Form von Plug-ins in einer proprietären Sprache (NASL) realisiert. Fast zehntausend solche Tests stehen derzeit zur Verfügung.

Da Vulnerability-Scanner stark automatisiert sind und den Gesamtkontext nicht bewerten, tendieren derartige Produkte zu Falschmeldungen. Der Tester muss die Ergebnisse also manuell verifizieren. (ave) ◆

*SIMON WEPFER ist Chief Technology Officer (CTO) und Consultant bei der Sicherheitsberatungsfirma Oneconsult in Thalwil, Schweiz.