Hacking & Security

Das umfassende Handbuch

DIE LESEPROBE

Kapitel 1
Einführung

Dieses Kapitel gibt eine erste Einführung in das Themengebiet *Hacking und Sicherheit* und beantwortet einige grundlegende Fragen:

- Was ist Hacking? Gibt es *gute* und *böse* Hacker?
- Was bedeutet Sicherheit?
- Warum ist Software so unsicher?
- Was sind Angriffsvektoren, und welche Angriffsvektoren gibt es?
- Was sind (Zero-Day) Exploits?
- Wozu dient Penetration-Testing?
- Was ändert sich mit KI-Tools?
- Welche Gesetze und Normen gelten für Hacking und Sicherheit?

Wenn Sie sich dieses Buch gekauft haben, interessieren Sie sich offensichtlich für diese Themen und haben vermutlich auch schon Vorwissen. Dennoch raten wir Ihnen, einen genaueren Blick in dieses relativ untechnische Kapitel zu werfen. Es dient auch als Glossar für die weitere Lektüre des Buchs. Selbst IT-Profis, zumeist Spezialisten in einem recht engen Fachgebiet, ist selten die vielfältige Nomenklatur der Security-Welt geläufig. Das Kapitel ist also nicht nur eine Einführung, es will auch eine sprachliche Basis für die weitere Lektüre dieses Buchs schaffen.

1.1 Hacking

Die Wikipedia definiert einen *Hack* als eine Maßnahme, die Sicherheitsmechanismen eines Systems zu brechen oder zu umgehen. Ein Hack ist in diesem Kontext also ein nicht vorgesehener Weg, in ein System einzubrechen, Daten zu verändern, zu manipulieren oder zu zerstören. (Ein *Hack* kann auch eine unschöne, schnell erstellte Lösung eines Problems sein oder die Zweckentfremdung eines Geräts zur Durchführung anderer Aufgaben. Das ist aber nicht Thema dieses Buchs.)

Hacking ist demnach die Suche nach Hacks, ein *Hacker* die Person, die sich damit beschäftigt. In den Medien wird der Begriff *Hacking* meist in einem negativen oder kriminellen Kontext verwendet. Das ist aber nicht korrekt. Hacking an sich ist wert-

neutral. So wie ein Messer gleichermaßen dazu verwendet werden kann, Gemüse zu schneiden oder jemanden umzubringen, kann die Suche nach einem Hack dazu dienen, die Sicherheit des Systems zu verbessern oder aber das System anzugreifen und Schaden anzurichten.

Auch für Hacker gelten Regeln. Zum einen verbieten Gesetze jede unbefugte Datenmanipulation, mitunter auch schon den Versuch des Eindringens in ein Computersystem. Zum anderen hat die Hacking-Community immer wieder eigene Ethikregeln definiert. Freilich gibt es dafür keinen internationalen Standard. Vielmehr hängt es stark vom kulturellen und politischen Kontext ab, was ein Hacker tun darf oder tun soll. Unter diesem Aspekt werden Hacker mitunter in drei Gruppen unterteilt, wobei die Grenzen nicht immer ganz exakt zu ziehen sind:

- Verantwortungsvolle Hacker halten sich sowohl an Gesetze als auch an die Hacker-Ethik. Sie verwenden ihr Wissen, um die Sicherheit von Computersystemen zu verbessern, teilen entdeckte Sicherheitslücken den jeweiligen Herstellern mit etc. Für diese Art des Hackings wird der Begriff *Ethical Hacking* verwendet.
- Kriminelle oder bösartige Hacker nutzen ihr Wissen für kriminelle Aktivitäten und nehmen bewusst in Kauf, dass ihre Aktivitäten Schäden verursacht.
- Dazwischen stehen Hacker, die es mit den Regeln nicht so genau nehmen, aber höhere Ziele verfolgen – also z. B. die Verbesserung der Gesellschaft oder einen verantwortungsvolleren Umgang mit der Technik. Hier gibt es einen großen Graubereich, der eine klare Unterscheidung zwischen gut und böse schwierig bzw. von der eigenen gesellschaftlichen oder politischen Position abhängig macht.

Politisch korrekt?

Die gerade skizzierten Hacker-Typen werden oft als *White Hats*, *Black Hats* und *Grey Hats* bezeichnet. Der Sicherheitsexperte David Kleidermacher hat 2020 eine Diskussion über diese Begriffe angestoßen, weil sie rassistisch interpretiert werden könnten. Viele Mitglieder der Community argumentieren dagegen, dass *white* und *black* in diesem Kontext nichts mit der Hautfarbe zu tun haben, sondern mit dem Dualismus zwischen Tag und Nacht oder mit der Farbe von Hüten in alten Cowboyfilmen: In einigen dieser Filme tragen die Bösewichte schwarze Hüte.

https://www.infosecurity-magazine.com/news/google-vp-withdraws-black-hat

Der abwertende Begriff *Script Kiddies* bezeichnet Personen, die ohne tiefergehendes Wissen mit im Internet leicht zu findenden Scripts bzw. mit KI-Tools Hacking-Angriffe durchführen und mitunter großen Schaden anrichten. Es ist aber umstritten, ob auch Script Kiddies zu Hackern zählen. Der zur besseren Abgrenzung vorgeschlagene Begriff *Cracker* hat sich auf jeden Fall nicht durchgesetzt.

Hacking-Contests, Capture the Flag

Hacking will gelernt werden. Natürlich können Sie dazu Bücher wie dieses lesen und die hier präsentierten Techniken selbst ausprobieren. Wesentlich unterhaltsamer und speziell in IT-Studentenkreisen sehr beliebt sind Hacking-Wettbewerbe. Dabei erhalten die Teilnehmer Zugang zu speziell präparierten Computersystemen, meist in Form von virtuellen Maschinen. Die Zielsetzung besteht häufig darin, in das System einzudringen und darin möglichst schnell versteckte »Schätze« (*Flags*) zu finden. Die Sammelbezeichnung für derartige Wettbewerbe lautet *Capture the Flag* (CTF). Als Teilnehmer sind oft nicht nur Einzelpersonen zugelassen, sondern ganze Teams.

Zu den klassischen CTF-Wettbewerben gibt es auch diverse Varianten, bei denen beispielsweise jedes Team einen Server erhält. Das Ziel besteht nun darin, den eigenen Server gegen die Angriffe der anderen Teams zu schützen und gleichzeitig die Server der anderen Teams anzugreifen und zu »erobern«. Einzelne Teilaufgaben werden dabei durch Punkte belohnt. Das Team, das die meisten Punkte erreicht, ist Sieger.

Im Internet gibt es diverse Seiten, auf denen virtuelle Maschinen ehemaliger Hacking-Wettbewerbe zum Download zur Verfügung stehen (suchen Sie z. B. nach »hacking ctf images«). Damit können Sie selbst versuchen, wie weit Sie kämen. Oftmals gibt es auch mehr oder weniger konkrete Lösungsanleitungen (suchen Sie nach »hacking ctf writeups«).

Virtuelle Maschinen zum Üben

Einsteigerinnen und Einsteiger sind mit den zumeist ganz speziellen Aufgaben bei Hacking-Contests überfordert. Ein besserer Startpunkt sind speziell für den Unterricht konzipierte virtuelle Maschinen oder Docker-Images, in denen veraltete Versionen gängiger Software zum Einsatz kommen. Außerdem sind diese Maschinen mit diversen Sicherheitslücken präpariert. Damit ist ein Erfolgserlebnis nahezu garantiert.

Einige derartige Testsysteme stellen wir Ihnen in Kapitel 4, »Hacking lernen«, vor.

Pen-Test versus Hacking

Ein *Penetrations-Test* (kurz *Pen-Test*) ist ein umfassender Sicherheitstest für ein Computersystem (siehe auch Kapitel 11, »Externe Sicherheitsüberprüfungen«, und Kapitel 12, »Penetration-Testing«). Oft wird damit eine firmenfremde Person oder Organisation beauftragt. Der oder die Pen-Tester versuchen, wie Hacker vorzugehen, also das System anzugreifen und Sicherheitslücken zu finden. Es kommen also dieselben Arbeitstechniken zur Anwendung. Der wesentliche Unterschied zwischen Hackern und Pen-Testern besteht also weniger in der Arbeitsweise als vielmehr darin,

dass der Pen-Tester einen expliziten Auftrag für seine Arbeit hat und im Rahmen seiner Tests keine Daten manipuliert oder zerstört, sondern die gefundenen Mängel meldet, damit diese dann behoben werden.

Pen-Tester haben allerdings einen großen Vorteil im Vergleich zu Hackern: Sie müssen nicht im Verborgenen agieren. Ein Hacker wird seinen Angriff in der Regel nicht mit einem großen Scan starten, weil dessen intensive Tests bei einem gut gesicherten Server alle Alarmglocken läuten lassen. Ein Pen-Tester, der im Einverständnis mit der Firma agiert, kann derartige Werkzeuge dagegen ohne Bauchweh einsetzen.

Bei Pen-Tests oder Security-Übungen ist oft vom *Red Team* und *Blue Team* die Rede: Das *Red Team* besteht aus den Angreifern; zumeist handelt es sich dabei um externe Beraterinnen und Berater. Im *Blue Team* arbeitet die Verteidigung, in der Regel die für die Sicherheit verantwortlichen Mitarbeitenden der Firma oder Organisation. Manchmal gibt es auch ein *Purple Team*, in dem sich einzelne Mitglieder aus beiden Gruppen austauschen, damit spätestens nach Abschluss der Übung beide Seiten einen Wissensgewinn erzielen und mehr über die eingesetzten Werkzeuge und Techniken lernen.

Hacking-Verfahren

Wenn es darum geht, an fremde Daten heranzukommen, sie zu manipulieren oder auf andere Weise Schaden an IT-Systemen zu verursachen, führen viele Wege zum Ziel:

- **Network Hacking:** Gewissermaßen das »klassische« Hacking; es erfolgt über Netzwerkverbindungen. Es nutzt z. B. unsichere Passwörter, eine schlampige Konfiguration oder bekannte Fehler aus, um den Angriff durchzuführen. Das Ziel ist es zumeist, entweder direkt oder durch das Erraten/Abhören eines Passworts oder Passwort-Hashes einen uneingeschränkten Zugriff auf den Rechner zu erhalten (Root-Zugriff).

  Varianten dazu sind fingierte Webseiten zur Passworteingabe (*Phishing* im Sinne von *Password Fishing*) oder das Ausnutzen von Programmierfehlern, um auf Webseiten eigenen Code bzw. SQL-Statements auszuführen (HTML-Injections, SQL-Injections etc., siehe Kapitel 17, »Sicherheit von Webanwendungen«).
- **Passwort-Hacking:** Die Kenntnis des richtigen Passworts bietet den einfachsten Weg in den angegriffenen Rechner. Dementsprechend viele Techniken zielen darauf ab, ein Passwort herauszufinden. Dazu zählen systematisches Ausprobieren (*Cracking*), das Mitprotokollieren aller Tastatureingaben durch Software oder Hardware (*Key-Logging*), das Auslesen und Wiederverwenden von Passwort-Hashes etc. Die meisten dieser Verfahren setzen allerdings bereits Zugriff auf den Rechner voraus, entweder über das Netzwerk oder physisch (z. B. um einen USB-Key-Logger zu applizieren oder die Funktastatur abzuhören).

- **Backdoors:** Den ganzen Hacking-Aufwand kann sich ein Angreifer ersparen, wenn er eine sogenannte *Backdoor* in ein Programm kennt oder gar selbst einbaut. Im einfachsten Fall ist das eine nur dem Hersteller bekannte Kombination aus Login-Name und Passwort, wie dies bei vielen Routern, Mainboards etc. üblich ist. Selten lässt sich verhindern, dass diese Passwörter früher oder später entdeckt und im Internet veröffentlicht werden. Die Backdoor kann aber natürlich auch einen wesentlich raffinierteren Mechanismus verwenden.

  Bei Open-Source-Software sind dauerhafte Backdoors nahezu auszuschließen – sie würden im öffentlich zugänglichen Code auffallen. Es hat allerdings schon Fälle gegeben, bei denen ein Hacker eine modifizierte Version eines Open-Source-Programms zum Download angeboten hat. Eine besonders raffinierte Variante besteht darin, die Backdoor nur in das Kompilat (aber nicht in den Quellcode) einzubauen; das erfordert aber, dass der Hacker vollen Zugriff auf das Projektrepository hat.

  Derartige Manipulationen sind leicht zu bewerkstelligen und fallen oft erst nach einiger Zeit auf. Deswegen ist es empfehlenswert, Software generell nur von offiziellen Websites herunterzuladen und sich die Mühe zu machen, die Prüfsummen zu kontrollieren. (In der Praxis muss man freilich davon ausgehen, dass sich bestenfalls wenige securityaffine Enthusiasten diese Mühe machen.)

  Ganz anders sieht es bei kommerzieller Software aus, die nicht im Quellcode vorliegt: Es geistern unzählige Verschwörungstheorien durch das Netz, dass Hersteller oder staatliche Geheimdienste routinemäßig Backdoors in Betriebssysteme und Kommunikations-Software einbauen. Angesichts der Snowden-Enthüllungen ist das durchaus plausibel. Und da sich mangels Quellcode weder die Existenz noch die Nicht-Existenz einer Backdoor beweisen lässt, wird diese Unsicherheit bleiben.

- **Bugdoors**: Noch schlechter lässt sich die Existenz sogenannter *Bugdoors* beweisen. Das sind Fehler, *Bugs*, die wie ein Sicherheitsproblem aussehen, aber bei einem Blick auf den Code den Anschein erwecken, als seien sie absichtlich eingebaut worden. Das Ziel eines *Bugdoors* besteht also darin, eine Backdoor zu öffnen und diese als Sicherheitsproblem zu tarnen. Wird der »Fehler« dann entdeckt, wird er reumütig behoben.

  Software enthält Fehler, das ist eine unumstößliche Weisheit. Ob ein Fehler absichtlich eingefügt wurde, lässt sich hinterher schwer beweisen, ohne die Intention der Entwicklerinnen und Entwickler zu kennen. Aber bei der Behebung mancher Fehler verbleibt ein schaler Beigeschmack. Ist dieser (oft triviale) Fehler wirklich versehentlich passiert? Und über mehrere Jahre niemanden im Entwicklerteam aufgefallen?

- **Supply Chain Attacks:** Software setzt sich aus unzähligen Komponenten zusammen, also Programmiersprachen, Bibliotheken, Zusatzmodulen usw. Bei der Pro-

grammentwicklung kommen weitere Bausteine hinzu: Entwicklungsumgebungen oder Editoren, deren Erweiterungen, Plug-ins für den Webbrowser, Tools zum Testen/Ausführen/Ausliefern des Codes, zur Versionsverwaltung etc. Gerade im Open-Source-Bereich gibt es viele Module oder Minibibliotheken, die nur eine kleine, scheinbar triviale Aufgabe erledigen. Um deren Wartung kümmern sich winzige Teams, vielfach Einzelpersonen, oft unbezahlt (und unbedankt). Angreifer haben in diesen Komponenten ein attraktives Ziel entdeckt: Durch die Manipulation des Codes können Backdoors oder andere Funktionen eingeschleust werden.

- **Viren, Würmer und andere Schad-Software (Malware):** Schad-Software ist ein Programm, das auf einem Computer oder Gerät unerwünschte Funktionen ausführt. Je nachdem, wie sich derartige Software ausbreitet bzw. tarnt, ist dann von Viren, Würmern, trojanischen Pferden, Backdoors die Rede.

  Die technische Ausführung passt sich im Laufe der Zeit an die gerade gängige IT-Infrastruktur an. Während sich erste Viren noch über Disketten ausbreiteten, waren im letzten Jahrzehnt E-Mails der populärste Weg. Aktuell stehen vernetzte Geräte abseits herkömmlicher Computer stärker im Vordergrund (Smart/Embedded Devices, Internet of Things).

  Schad-Software ist auch auf Smartphones äußerst beliebt (siehe Kapitel 23). Ein Klassiker war die Taschenlampen-App, hinter deren an sich nützlicher Funktion andere Funktionen zum Ausspähen des Nutzers verborgen sind. Heute ist die Tarnung zumeist besser, aber die Idee ist gleich geblieben.

  Auch die Zielsetzung von Schad-Software ändert sich, unterliegt gleichsam Modetrends. Besonders populär waren zuletzt Verschlüsselungsprogramme (*Ransomware*), die zuerst möglichst viele Dateien verschlüsseln. Danach erhält der Nutzer, die Firma oder das Krankenhaus das Angebot, gegen die Bezahlung einer hohen Summe in Bitcoins oder einer anderen Crypto-Währung einen Schlüssel zu erwerben, um die eigenen Dateien wieder freizuschalten. Dieses Geschäftsmodell funktioniert derartig gut, dass Kriminelle auf entsprechenden Seiten mit wenigen Mausklicks ihren eigenen Verschlüsselungs-Trojaner zusammenklicken und kaufen können (*Cybercrime as a Service*).

  Der Umsatz dieses Geschäftszweigs betrug 2023 deutlich mehr als eine Milliarde Dollar. 2024 gab es einen Rückgang, vermutlich aufgrund besserer Vorsichtsmaßnahmen der Firmen.

- **Denial of Service (DoS):** Einen ganz anderen Ansatz haben Denial-of-Service-Attacken. Dabei geht es einzig darum, den Betrieb einer Firma oder den Zugang auf eine missliebige Webseite durch so viele Anfragen zu stören, dass ein regulärer Betrieb nicht mehr möglich ist. Besonders gut funktionieren Denial-of-Service-Attacken, wenn dabei gleichzeitig ein Software-Fehler ausgenutzt werden kann, der die Software des Servers gezielt zum Absturz bringt.

Für DoS-Angriffe werden oft *Botnets* verwendet. Ein Botnet ist ein Netzwerk von Computern oder Geräten, die schon früher mit anderen Verfahren unter die Kontrolle des Hackers gebracht wurden. Ein Botnet kann dazu verwendet werden, koordiniert Hunderttausende von Anfragen pro Sekunde an einen bestimmten Server zu senden, bis dieser vom Ansturm überfordert nicht mehr richtig reagiert. Bei dieser Art von Angriff spricht man von einem *Distributed Denial of Service*, kurz DDoS.

Einzelne Firmen sind in der Regel nicht in der Lage, sich gegen einen gezielten DDoS-Angriff zu wehren. Dazu bedarf es der Hilfe der Firmen, die für die Internet-Infrastruktur verantwortlich sind. Diese können z. B. in den großen Netzwerkknoten mit Filtern oder Firewalls eingreifen.

In der Kombination besonders gefährlich

In der Praxis nutzen viele Angriffe mehrere Exploits aus und wenden unterschiedliche Verfahren parallel an. Raffinierten Hackern gelingt es immer wieder, durch die Kombination aus für sich alleine genommen relativ harmlosen Schwachstellen einen erfolgreichen Angriff durchzuführen.

Hacking-Ziele

Die Anzahl der Hacking-Ziele hat sich in den vergangenen Jahren dramatisch erhöht. Während sich das »klassische« Hacking gegen Computer bzw. Server richtete, gilt es nun auch, Smartphones sowie alle vernetzten Geräte im Auge zu behalten. Dazu zählen Netzwerk-Router, -Switches und -Firewalls, Drucker, TV-Geräte, WLAN- oder Bluetooth-fähige Lautsprecherboxen, automatische Staubsauger, Webkameras, sonstige elektronische Geräte und Gadgets (*Internet of Things* = IoT), Heizungs-, Belüftungs- und Beschattungssysteme (*Home Automation*), Wechselrichter für Photovoltaikanlagen, Komponenten für das Mobilfunknetz, elektronische Türen und Schlösser, Autos, Flugzeuge, medizinische Geräte, Industrieanlagen und vieles andere mehr. Und weil der Großteil dieser Produkte in China produziert wird und zum Teil auch die Software von dort kommt, wird speziell von den USA immer wieder das Potenzial staatlicher Einflussnahme bis hin zum Einbau von Überwachungsfunktionen thematisiert. Wirklich bewiesen wurde diesbezüglich bisher nichts, aber zumindest die theoretische Möglichkeit ist unbestritten.

Ein Thema für sich ist die Cloud: Naturgemäß besteht auch die Cloud aus Computern oder virtuellen Maschinen, die als solche angegriffen werden können. Gleichzeitig ist aber auch das Cloud-System als Ganzes ein Angriffsziel: Aus der Amazon Cloud wurden schon unzählige Geheimdokumente heruntergeladen, weil ein Administrator übersehen hatte, dass die betreffenden Verzeichnisse ohne jeden Schutz öffentlich

zugänglich waren. (Es lässt sich aber darüber streiten, ob das Ausnutzen einer derartigen Nachlässigkeit etwas mit Hacking zu tun hat.)

Angriffsziele auf Hardware-Ebene

In eine ganz andere Richtung gehen Angriffe auf Subkomponenten eines Geräts, also z. B. auf den WLAN-Chip oder die CPU: Beispielsweise stellte sich im Herbst 2017 heraus, dass viele Intel-CPUs, die über einen Zeitraum von zwei Jahren produziert wurden, auf unterster Ebene ein Minibetriebssystem mit Verwaltungsfunktionen besitzen – die sogenannte *Management Engine*. (Genau genommen handelt es sich dabei um ein adaptiertes Minix, also um eine für Schulungszwecke entwickelte winzige Unix-Variante.)

Man mag darüber streiten, wer derartige Funktionen überhaupt braucht – aber katastrophal wird die Sache, wenn sich herausstellt, dass über diese Verwaltungsfunktionen aufgrund grundlegender und zum Teil trivialer Fehler die CPU und über die CPU jede darauf laufende Software angegriffen werden kann. Es ist kein Wunder, wenn manche Kritiker hier sogar eine Backdoor vermuten.

Anfang 2018 wurde die nächste Sicherheitskatastrophe auf CPU-Ebene bekannt: Ein Fehler in mehreren CPU-Architekturen, der bei Intel-Modellen besonders gravierend ausgeprägt ist, ermöglicht Prozessen den Zugriff auf isolierte Speicherbereiche anderer Prozesse. Der Fehler ist derart elementar, dass es gleich eine ganze Reihe von Angriffsvarianten gibt. Die beiden wichtigsten bekamen die Namen *Meltdown* und *Spectre* (siehe Abschnitt 19.10).

Seither wurden diverse Varianten entdeckt, die nahezu alle Hersteller betreffen. Von solchen Fehlern sind Milliarden Geräte betroffen. Zwar besteht in vielen Fällen die Möglichkeit, Updates auf CPU-Ebene (sogenannte *Microcode-Updates*) durchzuführen. Viele Geräte werden die erforderlichen Updates aber nie erhalten. Glücklicherweise lassen sich die meisten CPU-Fehler in der Praxis nur schwer für konkrete Angriffe ausnutzen. Insofern ist eine davon ausgehende Katastrophe – zumindest, soweit diese öffentlich bekannt wäre – bisher ausgeblieben.

Ähnlich problematisch wie CPU-Fehler können Fehler in GPUs oder in Netzwerk-Chips sein: So wurde Anfang 2020 die Sicherheitslücke *Kr00k* bekannt, die WLAN-Chips von Broadcom und Cypress betrifft. Diese Chips sind Schätzungen zufolge in mehr als einer Milliarde Geräte (vor allem in Smartphones) eingebaut! Zwar gibt es Software-Updates, aber auch in diesem Fall muss man davon ausgehen, dass viele Geräte diese Updates nie erhalten haben.

Sie sehen schon: Fehler auf Hardware- oder Firmware-Ebene werden immer häufiger, und ihre Tragweite ist enorm: Zum einen sind solche Fehler betriebssystemunabhängig auszunutzen, zum anderen ist eine Behebung per Update besonders schwierig:

Zwar sind bei den meisten Chips Firmware-Updates möglich, deren Durchführung ist aber bei vielen Betriebssystemen kompliziert, bei anderen gar nicht vorgesehen. Für Mitarbeitende, die für die Sicherheit einer Firma oder Organisation verantwortlich sind, ist das ein Alptraum: Müssen nun alle PCs, Smartphones, Router usw. ausgemustert werden, für die kein Firmware-Update verfügbar ist? Wer übernimmt bzw. rechtfertigt die damit verbundenen Kosten?

Angriffsziele in der Supply Chain

In den vergangenen Jahren gab es immer mehr Angriffe auf die *Supply Chain*, also auf eher kleinere Komponenten, die in vielen Software-Stacks verwendet werden. Die Vorgehensweise variiert dabei stark. Der einfachste Ansatzpunkt besteht darin, ein Repository auf GitHub oder im Modulverwaltungssystem der jeweiligen Programmiersprache zu klonen, zu manipulieren und dann unter einem ähnlichen Namen anzubieten. Mit etwas Pech greift ein Projekt auf das manipulierte Repository anstelle des Originals zurück.

Es geht aber auch viel raffinierter: In einer zwischen 2022 und 2024 stattgefundenen Attacke auf die XZ-Bibliothek zur Komprimierung von Daten erschlichen sich Angreifer das Vertrauen des Projektbetreuers. In der Folge schleusten sie manipulierten Code ein, um so den SSH-Server zu unterwandern, der auf die Bibliothek zurückgreift. Die Vorgehensweise war insofern verblüffend, als sich die Angreifer über einen Zeitraum von über zwei Jahren in das Projekt einarbeiteten und damit einen enorm langen Atem bewiesen. Das eigentliche Sicherheitsproblem wurde erst am Ende dieses Zeitraums geschaffen.

Die Manipulation wurde im März 2024 gerade noch rechtzeitig entdeckt – nur Tage bzw. Wochen, bevor eine neue Version des SSH-Servers über viele gängige Linux-Distributionen hätte ausgeliefert werden sollen und auf diese Weise Millionen von SSH-Servern kompromittiert hätte. Der Angriff hätte zu einer der größten Sicherheitskatastrophen im Linux-Server-Umfeld werden können.

Die rechtzeitige Entdeckung war ein Glücksfall: Andreas Freund, einem bei Microsoft angestellten Entwickler, war ein untypisch hoher CPU-Verbrauch bei einer Testversion des SSH-Servers aufgefallen. Er ging der Sache auf den Grund und stieß letztlich auf den manipulierten Code. Der Vorfall hat aber einmal mehr die Abhängigkeit der Open-Source-Welt von winzigen Teilprojekten im Hintergrund verdeutlicht, eben von der *Supply Chain*.

Während der Angriff auf die XZ-Bibliothek gerade noch rechtzeitig entdeckt wurde, traf dies bei *Log4Shell* leider nicht zu. Der Fall ist aber auch vollkommen anders gelagert. `Log4j` ist eine sehr populäre Open-Source-Bibliothek. Sie wird in unzähligen Java-Programmen verwendet, um Meldungen zu protokollieren. Leider hat sich 2021 herausgestellt, dass viele Programme, die `Log4j` verwenden, damit einen geradezu trivial

einfachen Weg anbieten, fremden Code auszuführen – paradiesisch für jeden Hacker. Der vielfach ausgenutzte Exploit bekam den Namen *Log4Shell* und den maximalen Serverity-Score 10.0.

In diesem Fall lässt sich darüber streiten, ob die Bibliothek überhaupt fehlerhaft war: Eigentlich hat die Bibliothek mit einer besonders eleganten Logging-Syntax seit 2013 exakt wie beschrieben funktioniert. Dass sich der Mechanismus auch missbräuchlich anwenden lässt, ist erst acht Jahre später aufgefallen.

Das Fehlerverhalten bzw. die zu universelle Anwendungsmöglichkeit der Bibliothek wurde nach Bekanntwerden rasch behoben. Trotzdem sind noch heute angreifbare Programme im Einsatz. Jedes Programm, das `Log4j` verwendet, muss neu kompiliert und dann beim Kunden bzw. im jeweiligen Rechner oder Gerät aktualisiert werden. Und genau an dieser Stelle hapert es: Es gibt eine Menge Software, die nicht mehr gewartet wird oder wo die Verteilung von Updates (z. B. in IoT-Devices) sehr aufwendig ist.

Hacking-Tools

Um die Arbeit des Hackings zu erleichtern, wurden unzählige Programme entwickelt. Die Palette reicht von simplen Scripts für einen Netzwerk-Scan bis hin zu umfassenden Analysewerkzeugen, die einen Server oder ein Gerät systematisch auf alle gerade bekannten Sicherheitslücken und -probleme hin absuchen.

Dazu kommen Programme, die ursprünglich zur Analyse von Netzwerk-, WLAN- oder Bluetooth-Problemen oder für ähnliche Aufgaben konzipiert wurden, die sich aber natürlich wunderbar zweckentfremden lassen. Ein Großteil dieser Software ist im Internet kostenlos erhältlich, häufig sogar im Quellcode (Open-Source-Idee).

Daneben gibt es Firmen, die sich auf dieses Segment konzentrieren und Software für ganz spezielle Hacking-Aufgaben verkaufen, mitunter in einem gehobenen Preissegment für elitäre Zielgruppen (Polizei, Geheimdienste, Militär, internationale Sicherheitsunternehmen).

In diesem Buch konzentrieren wir uns auf gängige Tools, die kostenlos erhältlich und in der Praxis entsprechend häufig im Einsatz sind (siehe Kapitel 3, »Hacking-Tools«). Anstatt jedes Hacking-Tool separat zu suchen und herunterzuladen, greifen viele Hacker und Pen-Tester auf vollständige Toolboxen zurück, die eine riesige Sammlung von Werkzeugen in Form eines Werkzeugkastens anbieten. Am bekanntesten ist in diesem Kontext *Kali Linux* (siehe Kapitel 2): Das ist eine Linux-Distribution, die mehrere Tausend unter Linux lauffähige Hacking-Programme bündelt.

Hacking-Hardware

Hacking-Tools sind keinesfalls nur auf Software beschränkt. Für Hacking-Hardware hat sich mittlerweile ein ganzer Markt etabliert. Das Angebot beginnt bei simplen »Gadgets«, die wie ein USB-Stick aussehen, sich aber wie eine Tastatur verhalten und unter Windows flugs eine PowerShell öffnen, dort mit einem Kommando eine Schad-Software herunterladen und ausführen. Wenn es dem Ziel nicht gelingt, diesen Vorgang innerhalb von zwei, drei Sekunden zu stoppen, ist es schon zu spät.

Es gibt aber auch wesentlich intelligentere Geräte, bei denen es sich um unauffällig verpackte Minicomputer handelt. Wenn es dem Hacker gelingt, diese Geräte richtig zu platzieren (das erfordert in der Regel physischen Zugriff auf den Rechner des Ziels), kann er sich damit in die Netzwerk-, USB- oder Bluetooth-Kommunikation einklinken oder andere Aufgaben ausführen. In Kapitel 9, »WLAN, Bluetooth und SDR«, sowie in Kapitel 10, »Angriffsvektor USB-Schnittstelle«, stellen wir Ihnen einige derartige Hacking-Gadgets inklusive des sehr populären *Flipper* vor und zeigen, wie Sie sich dagegen schützen können. Lesenswert ist in diesem Zusammenhang die Zeitschrift c't, die in längeren Artikeln immer wieder aktuelle Tools präsentiert, zuletzt in der Ausgabe 5/2023:

https://www.heise.de/select/ct/2023/5/23005110103672248 43 (Paywall)

Quasi als Einstieg in die Welt der Hacking-Hardware empfehlenswert ist schließlich der Raspberry Pi: Dieser Mini-Computer ist zwar nicht für Hacking-Aufgaben konzipiert, lässt sich aber flugs als WLAN-Access-Point konfigurieren. Damit können Hacker z. B. versuchen, ihre Ziele in ein kostenloses, aber leider unverschlüsseltes WLAN zu locken. In weiterer Folge lassen sich dann alle möglichen Gemeinheiten realisieren, etwa die Manipulation von DNS-Einträgen zur Umleitung des Ziels auf Phishing-Webseiten.

1.2 Sicherheit

Was hat Hacking nun mit Sicherheit zu tun? Auf den ersten Blick scheint es sich ja um einen Gegensatz zu handeln. Das Ziel dieses Buchs ist es, Ihnen bei der Absicherung von Computersystemen zu helfen. Dazu benötigen Sie Hacking-Wissen aus zweierlei Gründen:

- Zum einen müssen Sie wissen, mit welchen Mitteln und Werkzeugen Angriffe durchgeführt werden. Eine vollständige Beschreibung aller Hacking-Tools würde den Rahmen dieses Buchs sprengen – aber wir bemühen uns, Ihnen zumindest einen ersten Überblick zu geben.

- Zum anderen ist es wichtig, zu verstehen, warum Computersysteme und Software angreifbar sind. Deswegen gehen wir in mehreren Kapiteln auf die Grundlagen und Interna von Sicherheitslücken (Exploits) ein.

Der Schlüsselansatz in diesem Buch besteht darin, dass wir Ihnen zuerst zeigen möchten, wie einfach es in vielen Fällen ist, ein System anzugreifen. Das ist der Hacking-Aspekt. Im zweiten Schritt geht es dann darum, Abwehrmaßnahmen zu treffen. Unser Motto laute also: Mehr Sicherheit durch Hacking.

Ein wenig überspitzt könnte man sogar formulieren: »Angriff ist die beste Verteidigung!« Indem Sie also Ihre eigenen Systeme selbst oder durch dazu beauftragte Pen-Tester angreifen, lernen Sie ihre Schwächen kennen und sind in der Lage, geeignete Schutzmaßnahmen zu ergreifen.

Dabei wollen wir keinesfalls falsche Hoffnungen wecken: Hundertprozentige Sicherheit ist mit den gegenwärtigen IT-Technologien nicht möglich. Das bedeutet aber keinesfalls, dass es sich nicht lohnt, die Sicherheit zu verbessern! Viele Cyber-Kriminelle suchen sich einfach die Ziele, die mit dem geringsten Aufwand zu attackieren sind. Schon ein paar einfache Sicherheitsmaßnahmen können dann den entscheidenden Unterschied machen.

Die in diesem Buch präsentierten Maßnahmen werden also nicht ausreichen, um eine professionell durchgeführte Firmenspionage oder gar einen Hacker-Angriff eines Geheimdienstes abzuwehren. Der Schutz gegen staatlich sanktionierte Cyber-Attacken ist klar außerhalb der Reichweite dieses Buchs.

Sicherheit im Kontext dieses Buchs

Wenn in diesem Buch von Sicherheit die Rede ist, dann meinen wir damit ausschließlich die Sicherheit vor Hacking-Angriffen. Naturgemäß geht Sicherheit viel weiter. Wenn Ihnen Ihr Betrieb, Ihre Organisation am Herzen liegt oder wenn Sie sich vor rechtlichen Konsequenzen eines fahrlässigen Umgangs mit fremden Daten schützen möchten, dann müssen Sie ganz andere Faktoren berücksichtigen:

Was passiert, wenn eine Festplatte/SSD unerwartet defekt ist? Wenn ein Bagger versehentlich die Netzwerkanbindung zu Ihrem Büro durchtrennt? Wenn das Firmengebäude oder der Serverstandort durch einen Brand zerstört wird? Wenn der Cloud-Zugang nicht mehr funktioniert? Gibt es dezentrale Backups? Gibt es Sicherheitsrichtlinien? Gibt es für einen Katastrophenfall konkrete Listen mit verantwortlichen Personen, durchzuführenden Aufgaben? Gibt es ganz allgemein Notfallpläne?

Naturgemäß gibt es eine Menge Sicherheitsmaßnahmen, die nicht nur gegen einen Hacking-Angriff schützen, sondern auch bei anderen Notfällen hilfreich sind. Aber in diesem Buch beschränken wir uns auf den Aspekt der IT-Security.

Warum sind IT-Systeme so unsicher?

Je mehr man sich mit Sicherheit beschäftigt, desto mehr kann man zu Frustration oder Resignation tendieren: Jedes Programm, jedes moderne technische Gerät – von der NAS-Festplatte bis zum Auto – scheint voller Fehler und Sicherheitslücken zu stecken. Und fatalerweise täuscht dieser Eindruck nicht: Es gibt diverse Statistiken, wie viele Fehler pro 1.000 Zeilen Code üblich sind. Sinkt die Anzahl der Fehler auf 0,5, also auf *nur* einen Fehler pro 2.000 Zeilen, spricht man bereits von *stabilem Code*. Betriebssysteme wie Windows, Linux, iOS oder Android bestehen aber aus vielen Millionen Zeilen Code!

Warum ist die Fehlerhäufigkeit so hoch? Programmierer(innen) sind Menschen, und die machen nun mal Fehler. Natürlich lässt sich die Anzahl der Fehler durch Sorgfalt, durch Reviews und Testdurchläufe reduzieren, aber es werden immer Fehler bleiben. (Glücklicherweise ist nicht jeder Fehler sicherheitsrelevant. Aber Hacker sind oft sehr kreativ darin, auch vermeintlich harmlose Fehler auszunutzen.)

Es kommt hinzu, dass nicht jede Software unter optimalen Bedingungen entwickelt wird: Das erste Ziel ist in der Regel, eine gewisse Funktion überhaupt einmal zu erreichen (»erster Milestone«). Das dauert oft länger als geplant. Sicherheitstechnische Checks sowie eine Absicherung des Codes werden zuerst nach hinten gereiht – und dann aus Zeitgründen gar nicht mehr durchgeführt.

Man sieht einem Programm oder Produkt leider nicht an, wie sicher es ist. Die Verkaufszahlen hängen vielmehr von der Verpackung, von der Funktionalität, von der Eleganz der Benutzeroberfläche und vom Marketing ab. Wenn dann ein Jahr später ein Sicherheitsproblem bekannt wird, beschäftigt sich die Firma (wenn es sie dann noch gibt) längst mit neuen Produkten. Rein wirtschaftlich gesehen lohnt sich die Behebung der Probleme nicht.

Ein weiterer Punkt besteht darin, dass selbst exzellente Programmierer nicht zwangsläufig Security-Experten sind. Die IT-Welt ist längst zu weitläufig, als dass eine Person in jedem Bereich federführend sein kann. (Nicht von ungefähr wurde dieses Buch von einem ganzen Team von Autoren verfasst.)

Zu guter Letzt passieren fundamentale Fehler selbst unter optimalen Voraussetzungen: Die Entwicklung und Implementierung neuer kryptografischer Verfahren ist derart komplex, dass sogar Teams von international anerkannten Experten Fehler passieren. Solche Fehler schlummern oft jahrelang im Code, bis sie – mitunter durch einen Zufall – entdeckt werden. Dann ist sprichwörtlich Feuer am Dach: Milliarden von Geräten verlassen sich auf Standardbibliotheken für TLS, HTTPS, WPA oder andere übliche Techniken der Verschlüsselung oder Authentifizierung.

Schwarzes Schaf IoT

Besonders gravierend machen sich derartige Probleme aktuell bei Internet-of-Things-Geräten bemerkbar (siehe auch Kapitel 24, »IoT-Sicherheit«): Die Webkamera oder die über WLAN oder Bluetooth steuerbare LED-Lampe wirft so geringe Margen ab, dass viele Firmen auf eine seriöse Absicherung und eine langfristige Wartung gleich ganz verzichten.

Mitunter ist gar keine Update-Möglichkeit vorgesehen. Von außen ist einem IoT-Gerät nicht anzusehen, wie seine Sicherheit technisch implementiert ist – die Verantwortung auf den Käufer abzuwälzen, funktioniert also nicht. Hier hat wohl nur der Gesetzgeber eine Chance, durch klare Regeln inklusive Produkthaftungsklauseln für Folgeschäden für mehr Sicherheit zu sorgen. Immer mehr IoT-Geräte verlassen sich auf herstellerspezifische Cloud-Systeme, senden dorthin Daten und beziehen von dort Updates. Wird ein derartiges Cloud-System nach einer Firmenpleite deaktiviert, funktionieren plötzlich unzählige Geräte nicht mehr oder nur noch eingeschränkt. Sicherheitstechnisch noch problematischer sind unsicher implementierte Cloud-Funktionen.

Eine weitere Unsicherheitsquelle sind Smartphones (siehe Kapitel 23, »Mobile Security«): Manche Hersteller, vor allem im Billig-Segment, stellen für ihre Geräte nur für wenige Jahre Updates zur Verfügung. Dass es bisher (Stand Mitte 2025) noch nicht zu einer regelrechten Sicherheitskatastrophe gekommen ist, grenzt eigentlich an ein Wunder.

Angriffsvektoren

Der sperrige Begriff *Angriffsvektor* bezeichnet Wege, die ein Hacker beschreiten kann, um in das Computersystem Ihrer Firma oder Organisation einzudringen (siehe Abbildung 1.1).

Im Folgenden fassen wir einige gängige Verfahren zusammen und zeigen so die Breite der Möglichkeiten, aber auch die Schwierigkeit bzw. Unmöglichkeit, Rundumsicherheit zu schaffen.

- **Network Hacking:** Der Begriff *Network Hacking* wird oft verwendet, um »klassische« Wege des Hackings zusammenfassen. Mithilfe von Hacking-Werkzeugen, die einen Angriff über das Netzwerk/Internet ermöglichen, versucht ein Angreifer, in die Computer einer Firma/Organisation einzudringen und dort Daten zu entwenden oder zu manipulieren oder anderweitig Schaden anzurichten. Dabei werden Konfigurationsfehler und nicht behobene Schwachstellen ausgenutzt. Das eigentliche Angriffsziel sind in der Regel nicht die Rechner von Mitarbeitenden und andere Clients, sondern Server.

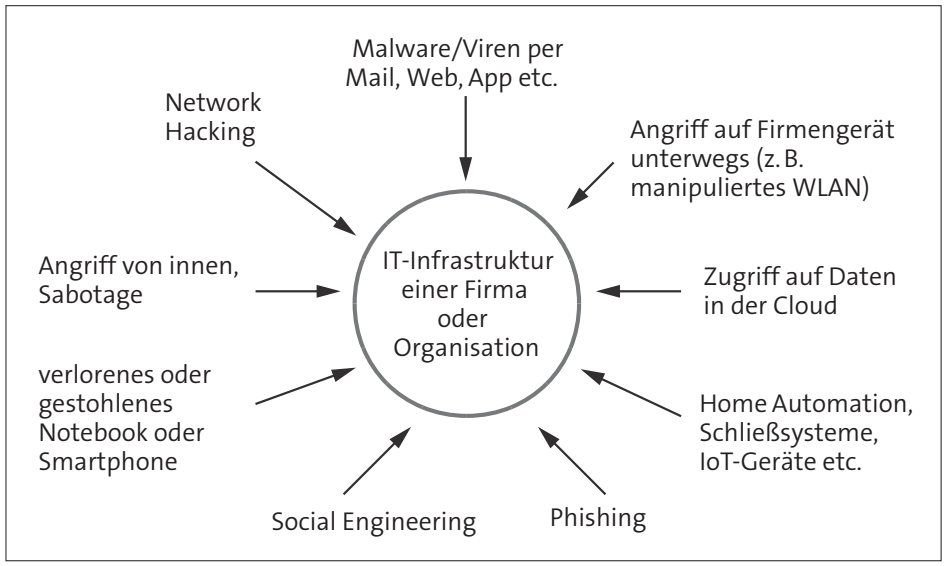

Abbildung 1.1 Populäre Angriffsvektoren

- **Gestohlene/Verlorene Notebooks oder Smartphones:** Wie auch immer ein Smartphone, Tablet oder Notebook in die Hände eines Angreifers gerät – es enthält auf jeden Fall eine Fülle von Daten, die für weitere Angriffe genutzt werden. Dazu zählen nicht nur vertrauliche Dateien, sondern z. B. auch die vom Webbrowser gespeicherten Zugangspasswörter für Cloud- und E-Mail-Accounts.

  Die Frage ist, ob der Angreifer an diese Daten herankommt. Bei modernen Smartphones ist dies ohne Passwort in der Regel unmöglich. Bei Notebooks hängt es davon ab, ob das Dateisystem verschlüsselt war (siehe Kapitel 6, »Offline Hacking«). Ist das nicht der Fall, ist der Zugriff auf sämtliche Daten für den Dieb ein Kinderspiel.

- **Zugriff auf Firmengeräte unterwegs:** Innerhalb einer Firma sind Smartphones und Notebooks normalerweise gut geschützt, sowohl physisch als auch (dank Firewalls etc.) innerhalb des Firmennetzwerks. Ganz anders sieht es aus, wenn Mitarbeitende mit ihren Geräten unterwegs sind, sich in unsicheren WLANs anmelden, Bluetooth-Funktionen nutzen etc.

  Auch ohne direkten physischen Zugriff auf das Gerät können Angreifer versuchen, Schwachstellen in den Funkprotokollen bzw. deren Verschlüsselungstechniken auszunutzen (siehe Kapitel 9, »WLAN, Bluetooth und SDR«). Eine andere Variante besteht darin, das Ziel mit einem kostenlosen WLAN in die Falle zu locken. Indem der Angreifer das WLAN selbst anbietet, oft unter fingiertem Namen in der Art von »Free-Hotel-WiFi«, stehen ihm diverse Manipulationsmöglichkeiten zur Verfügung.

Gegen solche Arten von Angriffen können sich die Mitarbeitenden Ihres Unternehmens zumindest teilweise schützen, indem sie unbekannte und unverschlüsselte WLANs meiden, nie Passwörter oder andere Informationen auf unverschlüsselten Webseiten eingeben und für den Zugang zum Firmennetz immer VPNs (*Virtual Private Networks*) verwenden.

- **Infizierung von Geräten durch Malware, Viren etc.:** Ein weiterer Weg in den Rechner des Ziels führt über E-Mails, infizierte Webseiten oder Malware-Apps (speziell bei Smartphones). Der E-Mail-Titel verweist auf ein wichtiges Firmendokument, das rasch geöffnet werden muss, die Webseite verspricht iPhones oder andere Güter zum halben Preis, die App verwendet nützliche Funktionen, um dahinter Malware zu tarnen. (Ein Klassiker sind »Taschenlampen-Apps« zum Einschalten des LED-Lichts. Je nach Betrachtungsweise kann man auch manche Social-Media-Messenger als Spyware sehen.) In allen Fällen ebnen gutgläubige Anwender dem Angreifer den Weg. Fehlen auf dem Gerät des Anwenders nun auch noch die letzten Updates, gelingt es dem Angreifer über eine Schwachstelle, eigenen Code auszuführen, und das Unglück nimmt seinen Lauf …
- **Manipulierte Hardware:** Schadware kann natürlich auch über einen USB-Stick oder eine SD-Karte ihren Weg in den Computer finden. Der USB-Stick kann scheinbar seriös wirkenden Bewerbungsunterlagen beiliegen oder einfach auf der Straße vor dem Firmengebäude herumliegen und die Neugierde eines Mitarbeiters wecken.

  Außerdem gilt: Nicht alles, was wie ein USB-Stick aussieht, ist auch einer! In einem USB-Stick-Gehäuse kann ein ganzer Mini-Computer stecken, der sich z. B. dem Computer als Tastatur zu erkennen gibt, sofort nach dem Einstecken Tastatureingaben simuliert und innerhalb von zwei, drei Sekunden ein PowerShell-Fenster öffnet, dort ein Schadprogramm herunterlädt und dieses startet. Es gibt auch rein zerstörerische Varianten. Diese Geräte laden sich über den USB-Port mehrere Sekunden lang auf und legen dann eine so hohe Spannung an, dass der Impuls den USB-Port und mit etwas Pech den ganzen Computer zerstört.

  Gegen solche Bedrohungen kann sich eine Firma nur durch gezielte Mitarbeiterschulung wehren: USB-Sticks und andere elektronische Geräte aus fraglicher Herkunft dürfen *nie* an einen Firmenrechner angesteckt werden!
- **Angriffe auf die Cloud:** Wozu soll sich der Angreifer die Mühe machen, in das Netzwerk oder die Geräte einer Organisation einzudringen, wenn die Daten ohnehin in der Cloud zum freien Download liegen? Ganz so schlimm ist es in der Praxis zwar nicht immer, in den vergangenen Jahren gab es aber immer wieder Fälle, bei denen Firmen oder Organisationen (im Herbst 2017 pikanterweise sogar der amerikanische Geheimdienst) Dateien aufgrund fehlender oder falscher Konfiguration unverschlüsselt und ohne Login auf Cloud-Servern liegen ließen.

Selbst wenn dieser Worst Case nicht vorliegt, ist die Cloud ein verlockendes Angriffsziel. Hacker können versuchen, die Cloud als solche anzugreifen. Zielführender ist in der Regel der Versuch, die Zugangsdaten zur Cloud zu entwenden oder den Anwender durch Phishing (siehe den folgenden Punkt) dazu zu bringen, dass er die Zugangsdaten selbst verrät.

Schließlich ist zu vermuten, dass zumindest die US-Geheimdienste weitreichenden Zugriff auf alle Dateien haben, die in den Clouds der großen amerikanischen Internetfirmen liegen. Soweit Sie Ihre Dateien nicht selbst verschlüsselt haben, ist davon auszugehen, dass der US-Geheimdienst sie mühelos auswerten kann.

- **Angriffe auf die Netzwerkinfrastruktur:** Seit 2019 tobt ein Kampf darum, mit welcher Technik die nächste Generation der Mobilfunknetze implementiert wird. Technologisch und preislich führend scheint Huawei zu sein. Den USA ist das ein Dorn im Auge, wobei schwer zu differenzieren ist, ob dahinter nicht (auch) wirtschaftliche Interessen stehen. Die offizielle Argumentationslinie geht dahin, dass China eine Backdoor in die Hard- oder Software einbauen könnte, sodass das 5G-Netz die Basis für Spionageangriffe des chinesischen Geheimdiensts sein könnte. (Und diese Möglichkeit besteht tatsächlich, das ist nicht von der Hand zu weisen.)

  In der Diskussion weitgehend untergegangen ist die Tatsache, dass auch die etablierten 3G- und 4G-Netze voller Sicherheitslücken sind. Es wäre spannend zu wissen, welche Staaten hier Überwachungsmöglichkeiten eingebaut oder (aufgrund von Planungs- oder Implementierungsfehlern) gefunden haben. Ein bisschen drängt sich der Eindruck auf, dass die USA deswegen so große Angst vor Huawei hat, weil sich deren Geräte und die dazugehörende Software komplett ihrer Kontrolle entziehen. Das führt zur fast schon philosophischen Frage, ob sich ein Staat bzw. seine Kommunikationsunternehmen lieber von den USA oder von China ausspionieren lassen wollen. Pest oder Cholera also?

  Klar ist natürlich auch, dass Angriffe auf die Infrastruktur selten von »kleinen Hackern« ausgehen, sondern eher von Staaten oder staatsnahen Organisationen. Insofern ist die Infrastrukturfrage weit außerhalb der Reichweite dieses Buchs.

- **Phishing:** Beim *Phishing* (also *Password Fishing*) versucht ein Angreifer, den Anwender zur Eingabe eines Passworts zu überreden – z. B. in einer Mail, die die Verifizierung eines Online-Kontos verlangt und dabei auf eine falsche Webseite führt. Mehr Details zu Phishing und zum Umgang mit Passwörtern folgen in Abschnitt 1.4, »Authentifizierung und Passwörter«.

- **Social Engineering:** Hacking ist eine sehr technische Disziplin. Da vergisst man leicht die menschliche Komponente. Wozu einen riesigen technischen Aufwand betreiben, wenn mit etwas Recherche und zwei Anrufen das Passwort herausgefunden oder die Anzahlung für einen fingierten Großauftrag erreicht werden

kann? Gegen derartige Angriffe helfen weder Firewalls noch Updates; hier bedarf es klarer Arbeitsrichtlinien und regelmäßiger Schulungen.

- **Physischer Zugang:** Oft unterschätzt wird die simpelste Form des Hackings: der physische Zugang zu Hardware. Dafür gibt es viele Formen, von dem auf der Bahnfahrt vergessenen (oder gestohlenen) Notebook bis hin zur unzufriedenen Mitarbeiterin, die das NAS-Gerät mit allen Backups aus dem offenstehenden Rechnerraum mitnimmt.

  In diesem Zusammenhang spielen auch Hardware-Hacking-Tools eine große Rolle: Wenn ein Besucher in einem unbemerkten Moment einen als USB-Stick getarnten Minirechner an einen Desktop-Computer ansteckt oder an eine freie Ethernet-Buchse anstöpselt, kann das mit etwas Pech wochen- und monatelang unbemerkt bleiben. Derartige Geräte bieten einem Angreifer aber oft weitreichende Überwachungs- und Manipulationsfunktionen (siehe auch den Kasten »Hacking-Hardware« am Ende von Abschnitt 1.1, »Hacking«).

- **Angriff von innen:** Je größer die Firma, desto wahrscheinlicher ist es, dass es unzufriedene Mitarbeitende gibt. Was nutzen die beste Firewall, VPNs etc., wenn ein Mitarbeiter aus Zorn, Frust oder gegen Bezahlung Ihr Computersystem von innen sabotiert? Am größten ist die Gefahr, wenn dieser Mitarbeiter einer Ihrer Administratoren ist – dann hat er oder sie oft nahezu unbegrenzten Zugriff auf alle Computer und Daten, kennt alle Sicherheits- und Backup-Verfahren etc. Wie der Fall Snowden gezeigt hat, war nicht einmal die *National Security Agency* (NSA) auf diese Situation vorbereitet.

  Einen vollständigen Schutz gegen den Angriff von innen gibt es nicht. Aber mit einigen allgemeinen Maßnahmen lässt sich das Risiko zumindest ein wenig mindern. Generell sollten Mitarbeitende nur Zugriff auf die Daten/Rechner/Systeme haben, die sie für ihre Arbeit tatsächlich brauchen. Wenn Mitarbeitende die Firma verlassen, sollten alle Passwörter, Netzwerkzugänge etc. sofort zurückgesetzt werden, Firmen-Notebooks und -Handys usw. schnellstmöglich eingezogen werden. Ein gutes Betriebsklima schadet natürlich auch nicht …

Wer ist Ihr Feind?

»Ich habe keine Feinde«, werden Sie vielleicht denken. Natürlich ist es schön, wenn das im privaten Umfeld zutrifft. Sobald Sie für die Sicherheit einer Firma oder Organisation zuständig sind, müssen Sie aber umdenken.

- **Ungezielte Angriffe von kriminellen Hackern:** Viele Hacking-Angriffe haben kein konkretes Ziel. Vielmehr geht es manchen Angreifern darum, einen möglichst einfachen Weg zum Geldverdienen zu finden. Das trifft z. B. für die meisten Crypto-Trojaner zu, die zuerst die Dateien der Festplatte verschlüsseln und dann ein »Lösegeld« für den Schlüssel zur Wiederherstellung der Daten verlangen. Parallel dazu

ziehen die Angreifer häufig persönliche Daten ab (Datensätze aus einem Krankenhaus, um nur ein Beispiel zu nennen) und drohen mit deren Veröffentlichung, wenn nicht rasch Lösegeld fließt.

Wenn Sie von einem derartigen Angriff betroffen sind, dann nicht, weil sich jemand die Mühe gemacht hat, Sie persönlich bzw. genau Ihre Firma anzugreifen. Vielmehr versucht der Angreifer, so viele Ziele wie möglich zu finden. Wenn von den Betroffenen jeder Hundertste zahlt, ergeben sich bereits schöne Einnahmen.

Ärgerlicherweise funktioniert dieses Geschäftsmodell umso besser, je mehr Firmen oder Organisationen (bzw. ihre Versicherungen!) sich für Bezahlung entscheiden. Auch wenn das für die betroffene Firma unter Zeitdruck der günstigste Ausweg erscheint, vergrößert sich mit jeder Zahlung das Risiko für die Allgemeinheit. Die amerikanische *Federal Trade Commission* (FTC) schätzt den durch *Crypto-Scams* verursachten Schaden auf 680 Millionen US$ allein für das erste Quartal 2022. (Das Schlüsselwort *Crypto* bezieht sich dabei sowohl auf die Verschlüsselung der Dateien als auch auf das übliche Zahlungsmittel, nämlich Bitcoins oder andere Crypto-Währungen.)

- **Script Kiddies:** Es ist kaum zu erwarten, dass ein Script Kiddy es explizit auf Sie oder Ihre Firma abgesehen hat. Am ehesten sind derartige Angriffe auf die IT-Infrastruktur einer Schule zu erwarten, durchgeführt von Jugendlichen, denen die Dimension eines derartigen »Streichs« unklar ist.

  Davon abgesehen wird die Rolle von Script Kiddies in den Medien eher aufgebauscht. Auch den meisten Jugendlichen ist die Tragweite von Hacking-Angriffen bewusst. Es mag vorkommen, dass durch das absichtliche Anwenden oder auch durch das bloßes Ausprobieren eines Scripts größerer Schaden entsteht, doch dürften solche Fälle die Ausnahme sein.

- **Gezielte Spionage/Sabotage:** Deutlich realer ist die Gefahr, dass Ihre Firma deswegen zum Ziel wird, weil ein Angreifer Firmengeheimnisse entwenden oder Ihren Betrieb durch Sabotage zu schädigen versucht. Das gilt naturgemäß speziell für Firmen, die High-Tech-Produkte erzeugen – egal, ob es sich dabei um Messgeräte, Medikamente oder moderne Konsumartikel handelt. Aber auch reine Daten sind wertvoll und somit ein attraktives Ziel – beispielsweise die Ergebnisse einer aufwendigen wissenschaftlichen Untersuchung, das Drehbuch oder der noch nicht ausgestrahlte Film einer TV- oder Kinoserie.

  Dass der Angreifer direkt von einer konkurrierenden Firma beauftragt wird, ist im internationalen Umfeld zwar nicht ausgeschlossen, aber doch recht unwahrscheinlich. Auch ein erfolgreicher Angriff eines an der Sache selbst gar nicht interessierten Hackers macht Sie oder Ihre Firma erpressbar bzw. kann riesigen Schaden anrichten!

Einige Staaten erwecken den Eindruck, organisierte Hacking-Gruppen zwar nicht offiziell zu unterstützen, aber doch zu dulden – zumindest, solange diese sich nicht erwischen lassen.

- **Geheimdienste:** Aufgabe der Geheimdienste ist es, den jeweiligen Staat vor Angriffen zu schützen. Das Argument »Ich habe ohnedies nichts zu verbergen« mag zutreffen, dennoch ist es nicht wünschenswert, wenn Ihre firmeninterne Kommunikation routinemäßig mitgelesen wird, Ihre in der Cloud gespeicherten Dateien automatisch ausgewertet werden.

  Unklar ist zudem, welche Rolle die Geheimdienste bei der Firmenspionage spielen. Sicher ist, dass die amerikanischen Geheimdienste umfassenden Zugriff auf in der Cloud gespeicherte Daten haben. Nicht eindeutig beweisbar ist aber, ob bzw. wie weit Geheimdienste – egal, welcher Nationalität – (zufällig?) gefundene Erkenntnisse auch an Firmen ihres Mutterlandes weitergeben. Mit ein wenig Interpretationsfreiraum kann man ohne weiteres argumentieren, dass der wirtschaftliche Erfolg von Firmen aus der Fahrzeug- und Flugzeugindustrie, dem Maschinenbau oder der Chemie/Pharmazie letztlich dem Staatsinteresse dient.

  Verdachtsmomente in diese Richtung gibt es definitiv. Insofern kann man europäischen Firmen nur dazu raten, auch Geheimdienste als »Feind« zu betrachten und in der Cloud möglichst nur selbst verschlüsselte Daten zu speichern. Das ist freilich schwieriger, als es hier klingt (siehe auch Kapitel 21, »Sicherheit in der Cloud«).

- **Staatlich gelenktes Hacking, Terrorangriffe:** Denken Sie beim Begriff »Cyberwarfare« nicht an reißerische Kinofilme – diese Art der Kriegsführung ist längst Realität, auch wenn die beteiligten Parteien es natürlich nicht zugeben. Beispielsweise wurde der Computerwurm »Stuxnet« gezielt entwickelt, um die Urananreicherung im Iran zu sabotieren. Der immense Aufwand, der in diesen extrem fokussierten Angriff investiert wurde, das tiefe Insider-Know-how, das erforderlich war, schließt »gewöhnliche« Hacker-Gruppen als Urheber aus. Auch im Ukraine-Konflikt gibt es viele Indizien, dass vor oder während des Kriegs durchgeführte Hacking-Angriffe nicht einfach lokalen Hackern zuzuschreiben sind, sondern von staatlichen Akteuren durchgeführt oder zumindest unterstützt werden. Die Wikipedia listet eine Menge weiterer Vorfälle auf, bei denen vermutet wird, dass die Hacker mit staatlicher Unterstützung vorgegangen sind:

  https://en.wikipedia.org/wiki/Cyberwarfare

  Bis zum Erscheinen dieses Buchs nicht eingetreten (oder nicht öffentlich bekannt geworden) sind erfolgreiche Hacking-Angriffe durch terroristische Gruppen. Es ist aber zu befürchten, dass auch das nur eine Frage der Zeit ist. Als besonders gefährdet gelten neben militärischen Einrichtungen vor allem Kraftwerke, Wasserwerke und andere Infrastruktureinrichtungen.

Die NATO betrachtet Cyberwarfare mittlerweile als einen zentralen Aspekt der Verteidigung und koordiniert entsprechende Aktivitäten seit 2008 im *Cooperative Cyber Defence Centre of Excellence* (CCD CoE) in Tallinn (Estland).

Wir haben es bereits erwähnt: Wir wollen absolut nicht den Eindruck erwecken, dass Sie sich mit dem Know-how aus diesem Buch der geballten Macht eines Geheimdienstes entgegenstellen können. Aber viele Hacker, egal mit welchem Hintergrund, gehen wie Einbrecher vor: Sie wählen die Ziele aus, die ihnen einen Angriff am einfachsten machen. Deswegen reichen bereits grundlegende Sicherheitsmaßnahmen aus, um zumindest ungezielte Angriffe abzuwehren.

Intrusion Detection

Manche Arten von Hacking-Angriffen bleiben dem Ziel nicht lange verborgen: Wenn der Rechner neu startet und eine sofortige Bitcoin-Zahlung verlangt, damit die verschlüsselten Dateien nicht auch noch gelöscht werden, ist auch Anwendern ohne Security-Vorwissen klar, dass sie Ziel eines Hacks geworden sind.

Viele Hacker, egal aus welchem Lager, sind aber durchaus nicht an ein paar schnell verdienten Bitcoins interessiert. Vielmehr kann auch die längerfristige Analyse des Ziels das eigentliche Ziel sein, beispielsweise zur Firmen- oder Staatsspionage oder zur Erkundung noch lohnenderer Angriffsmöglichkeiten.

In diesem Zusammenhang kommt der Begriff *Intrusion Detection* ins Spiel (siehe auch Kapitel 18): Er bezeichnet die Erkennung, dass ein Rechner (zumindest teilweise) unter fremder Kontrolle steht. Auf den ersten Blick mag die Erkennung von Schad-Software trivial klingen; tatsächlich ist sie aber ausgesprochen schwierig: Die Schad-Software befindet sich oft nur im RAM – ein Festplatten-Scan durch ein Virenschutzprogramm läuft deswegen ins Leere. Die Software ist oft winzig; ein Prozess versteckt sich hinter harmlosen Namen und verbraucht kaum Ressourcen. Am ehesten ist Schad-Software an besonderen Verhaltensmustern oder an auffälligen Netzwerkpaketen zu erkennen. Die zumeist verschlüsselten Pakete aus dem restlichen Netzwerkverkehr herauszufiltern, hat aber Ähnlichkeiten mit der sprichwörtlichen Suche nach der Nadel im Heuhaufen.

Für die Zeitspanne vom Einbruch bis zur Entdeckung eines Hacks sind je nach Quelle unterschiedliche Begriffe üblich, beispielsweise *Detection Time Span* oder *Breach Detection Gap*. Verschiedenen Statistiken zufolge ist die Zeitspanne auf jeden Fall erschreckend groß, oft viele Monate lang.

Forensik

IT-Forensik bezeichnet die Analyse von Computern, Smartphones oder anderen IT-Geräten. Forensik kommt vor allem aus zweierlei Gründen zum Einsatz:

- Einerseits möchte man nach einem erfolgreichen Hacking-Angriff in der Regel herausfinden, wer bzw. welche Gruppe sich auf welchem Weg Zugang zum Gerät verschafft hat. Diese Ursachenforschung hilft dabei, vergleichbare Fehler in der Zukunft zu vermeiden.
- Andererseits wollen Polizei, Geheimdienste etc. nach einem Betrug, Überfall oder Terroranschlag natürlich wissen, mit wem der Täter zusammengearbeitet hat und welche brisanten Daten sonst auf dem Gerät versteckt sind. Forensische Analysen spielen naturgemäß auch für Gerichtsprozesse eine große Rolle.

Weil die Dateisysteme von modernen Smartphones und Notebooks zumeist verschlüsselt sind (siehe auch Kapitel 6, »Offline Hacking«), berücksichtigen forensische Analysen auch die Spuren der Täter im Internet oder in der Cloud. Einen Einblick in forensische Methoden gibt Kapitel 8, »IT-Forensik«.

Zehn Schritte zu mehr Sicherheit

Die folgende Liste ersetzt nicht die Lektüre der folgenden Kapitel, kann aber als erste Checkliste dienen:

- **Updates:** Halten Sie die Software auf Ihren Geräten auf dem neuesten Stand. Sortieren Sie alle Geräte aus (z. B. Smartphones, NAS-Geräte, Webcams, Netzwerkdrucker, Switches oder Rechner mit nicht mehr gewarteten Windows-Versionen), für die es keine Updates mehr gibt, oder betreiben Sie solche Geräte, wenn es sich gar nicht vermeiden lässt, ausschließlich in getrennten Netzwerken.

  Laut Dr. Wieland Holfelder, Leiter des Google-Entwicklungszentrums in München, sind drei Viertel aller Hacking-Angriffe deswegen erfolgreich, weil schon lange bekannte Sicherheitslücken vorliegen.

- **Faktor Mensch:** Schulen Sie regelmäßig Ihre Mitarbeitenden im Hinblick auf den sicherheitsverantwortlichen Umgang mit Rechnern, Smartphones und anderen Geräten. Weisen Sie auf aktuelle Trends hin, z. B. auf Social-Engineering-, Phishing- oder Malware-Angriffe.

  Wenn man einschlägigen Studien trauen darf, haben zwei Drittel aller erfolgreichen Angriffe eine »menschliche Komponente«, funktionieren also nur, weil ein Mitarbeiter einen Fehler begeht, den USB-Stick vom Parkplatz einsteckt, den falschen E-Mail-Anhang öffnet oder ein Passwort auf der falschen Webseite eintippt.

- **Dateisystem verschlüsseln:** Alle Notebooks, die von Mitarbeitenden außerhalb der Firma eingesetzt werden, sollten über verschlüsselte Dateisysteme verfügen. Unter macOS ist dies längst eine Selbstverständlichkeit. Unter Windows muss Bitlocker explizit aktiviert werden (erfordert Windows Pro; die Sicherheit von

privaten Nutzern ist Microsoft nicht besonders wichtig). Unter Linux ist eine Verschlüsselung ärgerlicherweise nur während der Installation möglich.

- **Smartphones zentral administrieren:** So wie Windows-Clients im Unternehmen normalerweise zentral administriert werden, sollte dies auch für betrieblich genutzte Smartphones gelten. Entsprechende EMM-Werkzeuge (*Enterprise Mobility Management*) stellen wir in Kapitel 23, »Mobile Security«, vor.
- **VPN verwenden:** Der Zugang zu unternehmenskritischen Daten von außen, also z. B. durch Notebooks Ihrer Mitarbeitenden, sollte ausschließlich über verschlüsselte Verbindungen oder über ein *Virtual Private Network* (VPN) möglich sein. Kontrollieren Sie dessen Funktionsweise, oder beauftragen Sie jemanden mit der Einrichtung eines VPNs.
- **Pen-Tests durchführen:** Führen Sie für die IT-Infrastruktur Ihres Unternehmens erste grundlegende Security Tests durch, z. B. Port-Scans für alle Rechner, Exploit-Scan, Kontrolle auf triviale Passwörter etc. Geeignete Werkzeuge stellt z. B. die Distribution Kali Linux kostenlos zur Verfügung.

  Wenn Sie nicht über ausreichend Hacking-Know-how verfügen, beauftragen Sie einen externen Penetration-Test. (Das ist selbst dann eine gute Idee, wenn Sie über eine kompetente Security-Abteilung verfügen. Allzu leicht wird man betriebsblind!)
- **Eigene Software absichern:** Soweit Sie selbst Apps, Webapplikationen oder Geräte mit integrierter Software entwickeln: Beziehen Sie auch diese Produkte in Ihre Sicherheitsüberlegungen und -kontrollen ein.
- **Externe Server, Cloud und Backups absichern:** Schließen Sie in Ihre Überlegungen auch externe Root-Server, die von Ihrer Firma genutzte Cloud-Infrastruktur sowie das Backup-System ein.
- **Maßnahmen außerhalb der IT-Security:** Vergessen Sie nicht Dinge, die außerhalb der Reichweite dieses Buchs liegen, aber dennoch elementar sind: Dazu zählen betriebsinterne organisatorische Maßnahmen (Notfallpläne, Verantwortlichkeiten klären), die physische Sicherheit Ihrer IT-Infrastruktur (Wer hat den Schlüssel zum Serverraum? Ist der Raum wirklich immer verschlossen? etc.) sowie eine juristische Bewertung Ihrer IT-Security bzw. der Konsequenzen, wenn etwas schiefgehen sollte.
- **KISS (Keep it Simple, Stupid!):** Der letzte Punkt sollte vielleicht am Anfang stehen. Wie dem auch sei – bemühen Sie, die Komplexität Ihrer Systeme so niedrig wie möglich zu halten. Das gilt auf allen Ebenen, für kleine wie für große Organisationen, für eigenen Code ebenso wie für Docker-Setups, für jede im Betrieb eingesetzte Software. Systeme, die Sie bzw. die Ihre Administratoren nicht verstehen oder für deren Wartung zu wenig Zeit zur Verfügung steht, werden nie sicher sein.

Der Faktor Zeit

Hacker in Kinofilmen arbeiten immer unfassbar schnell. Natürlich ist das Teil der Dramaturgie, aber dennoch bleibt der Eindruck haften, Hacker seien generell allwissend und wüssten bei jeder Hürde sofort die richtige Lösung.

Das trifft keinesfalls zu! Hacker, egal ob verantwortungsvoll oder kriminell, sind IT-Spezialisten mit einem oft recht eingeschränkten Fokus. Wer regelmäßig Sicherheitslücken in Microsoft-Netzwerken oder im Active Directory sucht, ist nicht zwangsläufig auch Experte für Webserver, die unter Linux laufen.

Hacking erfordert Zeit. Wenn Sie als Administratorin oder Sicherheitsverantwortlicher tätig sind, spielt Ihnen der Faktor Zeit in die Hände. Je besser die grundlegenden Sicherheitsvorkehrungen sind, desto aufwendiger wird der Angriff – und desto größer ist die Chance, dass sich Angreifer einem anderen Ziel zuwenden.

Mehr Resilienz, mehr Unabhängigkeit, weniger Monokultur

Eine Grundregel für Anleger ist die Diversifikation des Portfolios. Diese naheliegende Regel senkt das Risiko, wenn eine einzelne Aktie, ein einzelner Rohstoff (Gold) etc. plötzlich an Wert verliert.

In großen IT-Infrastrukturen wird dagegen meist alles auf eine Karte gesetzt: Alle Rechner inklusive der Server verwenden Windows, befinden sich im gleichen Active Directory, nutzen die gleichen (meist amerikanischen) Cloud-Services. Bei einem erfolgreichen Angriff fällt die Infrastruktur zusammen wie das sprichwörtliche Kartenhaus. Es dauert Wochen, um ein von Schadsoftware infiltriertes Netzwerk wieder in Betrieb zu nehmen.

Natürlich wäre es am besten, wenn Sicherheitsmaßnahmen jeden Angriff verhindern könnten. So ehrlich sollten Sie aber sein: Garantierte Sicherheit gibt es nicht, auch nicht bei perfekten Schutzmaßnahmen. Deswegen sollten Sie in einem zweiten Schritt überlegen, wie Sie den Schaden minimieren, falls doch ein Angriff erfolgreich ist. Und diese Überlegungen führen zurück zur Einleitung dieses Abschnitts.

Nun ist uns klar, dass voneinander getrennte Netzwerke, voneinander losgelöste Active Directories, ein Mix aus Hardware und Software von unterschiedlichen Herstellen, die Nutzung unterschiedlicher Service- und Cloud-Anbieter den Administrationsaufwand und auch die Anzahl der möglichen Bedrohungen vergrößern. (Sie sollten es nicht übertreiben! Naturgemäß gelten die hier formulierten Ratschläge nur für große Betriebe oder Organisationen!)

Gleichzeitig erzielen Sie mit dem Abkehr von der Monokultur eine größere Herstellerunabhängigkeit und entkommen dem Alles-oder-Nichts-Prinzip. Wenn nach einem

Hacking-Angriff nur eine Abteilung und nicht gleich die ganze Firma steht, ist das definitiv ein Vorteil. Wenn sich nicht alle Ihrer Daten in der gleichen Cloud, sind Sie weniger erpressbar und nicht im gleichen Ausmaß von (wirtschafts)politischen Wirren abhängig.

Sicherheit ist nicht sichtbar

Plakativ formuliert: Wenn Apple eine neue iOS-Version vorstellt, können Sie später auf IT-Websites und in Zeitschriften nachlesen, mit welchen neuen Emoticons Sie Textnachrichten jetzt anreichern können. Das sind für Endanwender sichtbare Features. Wenn gleichzeitig ein ganzes Team von Entwicklern mit riesigem Zeit- und Ressourcenaufwand neue Sicherheitsmechanismen implementiert hat, ist dies in aller Regel niemandem auch nur eine Zeile Text wert. Und selbst wenn der Redakteur die Sicherheitsbemühungen anerkennt: Vermutlich ist es nicht möglich, die neuen Sicherheitsmechanismen so in drei Sätzen zu erklären, dass die Leser sie verstehen.

Ganz generell ist die Arbeit für mehr Sicherheit undankbar: Wenn alles gut geht, werden Ihre Mühen als Administrator oder IT-Sicherheitsbeauftragter als selbstverständlich angesehen. Die Chef-Etage wird höchstens gelegentlich die Frage stellen, wozu man Sie braucht, alles ist ja ohnedies bestens, oder? Wo ist der Nutzen? Im Mittelpunkt des Interesses werden Sie erst dann stehen, wenn tatsächlich einmal etwas schiefgegangen ist. Dann haben es alle schon immer gewusst, nur Sie haben das offensichtliche Problem zu spät erkannt.

Sicherheit ist unbequem

Als Sicherheitsverantwortlicher sollten Sie von der Anwenderseite nicht auf Lob hoffen. Von dort wird höchstens Kritik kommen, wenn der Anmeldeprozess in das neue VPN-System umständlicher ist als bisher, wenn der Zwei-Faktor-Login drei Sekunden länger dauert als der herkömmliche Login, wenn die Verschlüsselung der SSD das Notebook ein wenig langsamer macht, wenn Sicherheitsrichtlinien die Installation eigener Apps am Firmenhandy blockieren, wenn sich Benutzer an einen neuen Cloud-Client gewöhnen müssen etc.

Weniger Schlangenöl!

Als »Schlangenöl« wird speziell im Amerikanischen ein unwirksames, oft teures Mittel bezeichnet. Es heilt kaputte Gelenke, behebt Potenzprobleme und nach ein paar Wochen verschwinden alle Falten.

Manche Sicherheitsprodukte haben starke Ähnlichkeiten mit Schlangenöl. Sie versprechen Unmögliches oder realisieren Funktionen, die ohnedies schon existieren. (Viele Virenschutzprogramme für Windows gehen in diese Richtung.)

Warum ist Schlangenöl trotzdem so beliebt? Zum einen, weil sich damit mangelndes IT- und Security-Know-how bequem kaschieren lässt. Zum anderen, weil es als Argument für eine juristische Auseinandersetzung nach einem Schadensfall dienen kann: »Wir haben alles technisch Mögliche gemacht, wir haben sogar xxx €/Jahr für dieses oder jenes Produkt ausgegeben. Uns trifft keine Schuld, wir sind ja selbst das Opfer.« Und mit den richtigen Sachverständigen sind die Chancen nicht schlecht, dass diese Strategie sogar aufgeht.

Trotzdem wollen wir hier ein Plädoyer für »seriöse« Sicherheit halten. Wir haben es schon ausgeführt: Gute Administratoren zu bezahlen ist teuer, die Beauftragung eines Pen-Test ebenfalls; die von Sicherheitsexperten empfohlenen Maßnahmen sind selten populär. Trotzdem gilt: Wenn Ihre Firma IT-Produkte oder Dienstleistungen sicher anbieten will, dann kostet diese Sicherheit Geld und Arbeitszeit. Das ist unvermeidbar.

Wir wollen mit diesem kurzen Abschnitt keinesfalls andeuten, dass jede Sicherheits-Software nutzlos ist! Natürlich gibt es viele seriöse Anbieter. Aber soweit Sie auf zusätzliche Software setzen, um Ihre Firma oder deren Produkte besser abzusichern, machen Sie sich zumindest die Mühe, diese sorgfältig auszuwählen. Hinterfragen Sie die Funktionsweise im Detail.

Das CrowdStrike-Debakel

Im Juli 2024 fielen nahezu gleichzeitig ca. 8,5 Millionen Windows-PCs aus. Schuld war kein Hacker-Angriff, sondern ein Fehler in der weit verbreiteten Sicherheits-Software von CrowdStrike. Ein Software-Update für einen »Sensor« von CrowdStrike brachte Windows zum Absturz. Betroffen waren Banken, Fluglinien, Supermärkte etc.

CrowdStrike verspricht, Rechner vor Angriffen (inklusive Ransomware) zu schützen. Wir kennen CrowdStrike nicht aus eigener Erfahrung und können und wollen hier nicht erörtern, wie gut die Software tatsächlich funktioniert. Klar ist, dass die Kunden und Anleger von CrowdStrike dem Produkt weiter vertrauen: Der Börsenkurs von CrowdStrike brach zwar im Juli 2024 auf die Hälfte ein; aber schon fünf Monate später erreichte die Aktie einen neuen Höchststand.

Die Grenzen dieses Buchs

Im Vergleich zu anderen Hacking- oder Security-Büchern, die sich häufig auf *einen* Aspekt des Hackings und entsprechende Gegenmaßnahmen konzentrieren, verfolgen wir in diesem Buch einen wesentlich breiteren Ansatz: Wir berücksichtigen Windows *und* Linux, neben herkömmlichen Computern und Servern auch Smart-

phones und IoT-Geräte, stellen in zwei Kapiteln die Risken der Auslagerung von Daten in die Cloud dar, gehen in einigen Kapiteln zumindest kurz auf *Safe Coding* ein etc.

Dennoch ist eine umfassende und vollständige Beschreibung von Hacking und Sicherheit in einem Buch von vornherein unmöglich. Selbst zehn Bücher dieses Kalibers würden nicht ausreichen. Die folgende Liste nennt ganz kurz in Stichwörtern einige Themen, die in unserem Buch außen vor bleiben oder nur kurz angerissen werden:

- Sicherheitsmaßnahmen für Clientrechner (Virenschutzprogramme, Sicherheitseinstellungen, VPN etc. inklusive der Berücksichtigung von macOS)
- organisatorische Maßnahmen (Mitarbeiterschulungen, Backup-Strategien und -Systeme, Logging und Monitoring, Notfallpläne, Zertifizierungen) sowie juristische Absicherung
- applikationsspezifische Sicherheit, also Hacking-Verfahren und Sicherheitsmaßnahmen für spezifische Programme oder Programmgruppen. Dazu zählen SAP und andere betriebswirtschaftliche Standard-Software, Oracle und andere Datenbankserver, Joomla!, Typo3 und andere Webanwendungen, Java EE und andere Software-Frameworks zur Implementierung von eigenen Software-Lösungen etc. Diese Aufzählung ließe sich beinahe endlos fortsetzen.
- Sicherheit von Mobilfunknetzen (GSM, UMTS, LTE, 5G etc.)
- IT-Sicherheit in industriellen Anlagen, Geräten und Gebäuden (Home Automation, Industrie 4.0, Sicherheit von Autos, Flugzeugen, im öffentlichen Nahverkehr, in Krankenhäusern usw.)
- physische Sicherheit (Schließ- und Brandschutzanlagen, elektronische Schlüsselsysteme etc.)
- Safe Coding (Auswahl von Programmiersprachen und Testwerkzeugen, sichere Programmiertechniken etc.)
- mathematische und IT-theoretische Grundlagen (z. B. kryptografische Algorithmen, Hash-Verfahren, Zufallszahlen)

1.3 Exploits

Ein *Exploit* dient dazu, einen Fehler (eine Schwachstelle oder *Vulnerability*) in einem Computersystem auszunutzen, um auf diese Weise Zugang zum System oder zu seinen Daten oder auch nur zu einer einzelnen (Software-)Komponente zu erlangen oder um den Betrieb des Systems zu stören. Computersysteme, auf denen ein Angreifer einen bekannten Exploit entdeckt, stehen ihm offen wie Scheunentore. Aus der Sicht des Sicherheitszuständigen eines Betriebs gilt es, bekannte Schwachstellen so

- Social-Engineer Toolkit (SET): Phishing-Attacken und Co. realisieren
- Burp Suite: Schwachstellensuche in Webapplikationen
- Sliver: Command-and-Control Server

Zu vielen dieser Werkzeuge folgen in den weiteren Kapiteln konkrete Anwendungsbeispiele.

Mehr Kommandos

Wir gehen in diesem Buch nicht auf elementare Netzwerkkommandos wie `ping`, `curl`, `ip` oder `route` ein. Hierbei handelt es sich um weit verbreitete Kommandos, die Ihnen mit etwas Linux-Erfahrung bekannt sein sollten. Natürlich können auch solche Kommandos im Hacking-Kontext eingesetzt werden, aber das ist nicht ihre primäre Aufgabe.

Dafür stellen wir im Kontext der nächsten Kapitel eine Menge weiterer Hacking-Tools vor. Um nur ein paar besonders populäre Kommandos zu nennen: `john` und `hashcat` in Kapitel 7, »Passwörter«, `minikatz` in Kapitel 14, »Active Directory« oder `sqlmap` in Kapitel 17, »Sicherheit von Webanwendungen«.

Hilfreich zur Gewinnung eines ersten Überblicks ist das »Kali Cheatsheet«. Es enthält eine übersichtliche Referenz der wichtigsten Hacking-Tools, die mit Kali mitgeliefert werden. Sie können die PDF-Datei von der folgenden Seite herunterladen:

https://www.comparitech.com/net-admin/kali-linux-cheat-sheet

3.1 nmap

Das Kommando `nmap` (»Network Mapper«) versendet IP-Pakete und wertet die eintreffenden Antworten aus, um herauszufinden, welche IP-Adressen eines Netzwerksegments aktiv sind, welche Betriebssysteme auf den entsprechenden Geräten laufen und auf welchen Ports diese Geräte Netzwerkdienste anbieten. Das Kommando schafft damit die Arbeitsgrundlage für viele Formen des Penetration-Testings.

Das Kommando steht in fast allen Linux-Distributionen als Paket zur Verfügung und kann unkompliziert installiert werden. Sie brauchen für die Anwendung von `nmap` also keine Kali-Linux-Installation! Auf der Website *https://nmap.org* finden Sie auch Versionen für Windows und macOS.

Syntax

Die Syntax für den Aufruf von `nmap` sieht so aus:

```
nmap [optionen] adress(bereich)
```

Dabei sind unter anderem die folgenden Optionen zulässig:

- `-A`: »aggressiver« und ausführlicher Scan, entspricht `-sV -O -sC --traceroute`
- `-F`: nur die 100 wichtigsten Ports aus `/usr/share/nmap/nmap-services` berücksichtigen (schnell)
- `-iL datei`: scannt die in der Datei angegebenen IP-Adressen
- `-oN datei` / `-oG datei` / `-oX datei.xml`: schreibt die Ergebnisse in eine normale Textdatei, in eine Textdatei, die mit `grep` gut weiterverarbeitet werden kann, oder in eine XML-Datei. Ohne die Option verwendet `nmap` die Standardausgabe und das normale Textformat.
- `-O`: versucht, das Betriebssystem zu erkennen. Diese Option muss mit einer Scan-Option kombiniert werden, z. B. `-sS`, `-sT` oder `-sF`.
- `-p1-10,22,80`: berücksichtigt nur die angegebenen Ports.
- `-p1-65535` oder `-p-`: berücksichtigt wirklich *alle* Ports. Vorsicht, das macht `nmap` sehr langsam!

  Standardmäßig überprüft `nmap` nur die 1000 gebräuchlichsten Ports. Zur Auswahl dieser Ports wertet `nmap` die Datei `/usr/share/nmap/nmap-services` aus. Diese Textdatei enthält eine Referenz aller gebräuchlichen Ports. Die dritte Spalte enthält einen Wert, wie häufig dieser Port genutzt wird (basierend auf statistischen Daten von vielen in der Vergangenheit untersuchten Rechnern). Wenn Sie eine explizite Auflistung der 1000 Ports wünschen, führen Sie das folgende Kommando aus:

  ```
  nmap --top-ports 1000 -v -oG -
  ```

- `-Pn`: keinen Ping-Test durchführen, sondern alle Hosts als online betrachten und scannen (langsam!)
- `-sL`: listet alle Ports auf und gibt in der Vergangenheit zugeordnete Hostnamen an. Das gelingt besonders schnell, liefert aber veraltete Daten auch von Geräten, die aktuell gar nicht mehr online sind.
- `-sP`: nur Ping (schnell)
- `-sS`: TCP-SYN-Scan, gilt per Default, wenn `nmap` mit Root-Rechten ausgeführt wird
- `-sT`: Connect-Scan, gilt per Default, wenn `nmap` ohne Root-Rechte verwendet wird
- `-sU`: auch UDP berücksichtigen, darf zusammen mit einer anderen `-s`-Option verwendet werden
- `-sV`: Service Version Detection anwenden. Damit versucht `nmap`, bei offenen Ports herauszufinden, welcher Service dort angeboten wird.
- `--script <name>`: führt das angebene Script aus (siehe die folgende Überschrift).

- `-T0` bis `-T5`: wählt ein Timing-Schema. `-T5` ist am schnellsten. `-T3` gilt per Default. `-T0` und `-T1` sind extrem langsam, minimieren aber das Risiko, dass der Scan bemerkt wird.
- `-v`: detailliertere Ausgabe (*verbose*)

Sie müssen sich beim Aufruf für *eine* `-s`-Option entscheiden. Sie bestimmt das Scan-Verfahren. Einzig `-sU` darf mit anderen `-s`-Optionen kombiniert werden. Generell ist die richtige Wahl der Optionen ein Kompromiss zwischen Gründlichkeit und Geschwindigkeit.

Bitte beachten Sie, dass die obige Optionenreferenz nur einen stark vereinfachten Überblick über die Möglichkeiten von `nmap` gibt. Schon deutlich mehr Informationen enthält die rund 20-seitige `man`-Page. Und wenn Ihnen das noch nicht reicht, beschreibt das Buch »Nmap Network Scanning« auf beinahe 500 Seiten alle erdenklichen Grundlagen und Details des Network-Scannings. Rund die Hälfte dieses Buchs können Sie auf der `nmap`-Website sogar kostenlos lesen:

https://nmap.org/book

nmap-Scripting-Engine (NSE)

Neben den Port-Scanning-Funktionen gibt es in `nmap` einen Scripting-Modus, den Sie mit der Option `--script <name>` nutzen. `nmap` greift dabei auf vordefinierte Scripts zurück, die unter Kali Linux im Verzeichnis `/usr/share/nmap/scripts` gespeichert sind. Mitte 2025 standen mehr als 600 `*.nse`-Dateien zur Wahl! Die Scripts erfüllen alle erdenklichen Aufgaben, von denen wir hier nur einige beispielhaft nennen:

- `http-enum`: Suche nach versteckten Dateien auf einem Webserver
- `smb-enum-shares`, `smb-enum-users`, `smb-protocols`, `smb-os-discovery` usw.: Auflistung von Ressourcen auf einem Windows- oder Samba-Server
- `ssh2-enum-algos`: Auflistung unterstützter SSH-Algorithmen
- `ssl-enum-ciphers`: Auflistung unterstützter SSL-Algorithmen eines HTTPS-Servers

Eine Referenz über alle Scripts erhalten Sie mit `nmap --script-help all`.

Die NSE-Scripts benötigen eine Portnummer. Sie können die gewünschten Portnummer direkt übergeben (beispielsweise `nmap -p 443 --script http-enum`). Ohne die Option `-p` führt `nmap` zuerst einen Standard-Port-Scan aus und wendet das Script dann auf die passenden gefundenen Ports an.

Unsere Erfahrungen mit `nmap --script` waren durchwachsen. In der Theorie bietet das Kommando ein praktisches Subset von großen Vulnerability-Scannern wie OpenVAS, aber ohne deren riesigen Overhead. Insofern wäre `nmap --script` ein großartiges Tool, um grundlegende Tests schnell und bequem durchzuführen. Leider zeigt sich in der Praxis, dass viele NSE-Scripts alt und schlecht gewartet sind. Manche liefern

brauchbare Ergebnisse, andere gar keine. Zumeist bleibt unklar, ob ein Script falsch angewendet wurde oder ob die gewünschten Daten wirklich nicht ermittelt werden können.

Aus unserer Sicht ist der praktische Nutzen der NSE-Funktionen somit gering. Oft liefern für die jeweilige Aufgabe optimierte Spezial-Kommandos bessere Ergebnisse als `nmap`. Deswegen empfehlen wir, das Kommando `nmap` primär im Rahmen seiner Kernfunktionalität einzusetzen, also als Port-Scanner.

Beispiele

Das folgende Kommando führt einen schnellen Netzwerk-Scan im lokalen Netzwerk durch (256 IP-Adressen). Dank der Konzentration auf die wichtigsten 100 Ports ist die Sache in gut zwei Sekunden erledigt. Die `nmap`-Ausgaben wurden aus Platzgründen stark gekürzt:

```
nmap -F  -T4 10.0.0.0/24

  Nmap scan report for imac (10.0.0.2)
  Host is up (0.00019s latency).

  PORT    STATE SERVICE
  22/tcp  open  ssh
  88/tcp  open  kerberos-sec
  445/tcp open  microsoft-ds
  548/tcp open  afp
  MAC Address: AC:87:A3:1E:4A:87 (Apple)

  Nmap scan report for raspberrypi (10.0.0.22)
  Host is up (0.00038s latency).
  Not shown: 99 closed ports
  PORT   STATE SERVICE
  22/tcp open  ssh
  MAC Address: B8:27:EB:11:44:2E (Raspberry Pi Foundation)

  ...
  Nmap done: 256 IP addresses (6 hosts up) scanned
             in 2.42 seconds
```

Das zweite Beispiel analysiert den Rechner mit der IP-Adresse 10.0.0.36 wesentlich gründlicher und ermittelt das dort laufende Betriebssystem und nach Möglichkeit die Versionen der Netzwerk-Services. Deswegen dauert der Scan des einen Rechners länger als eine Minute. (Zum Testzeitpunkt lief unter der Adresse 10.0.36 eine virtuelle Maschine mit Metasploitable 2. Das ist ein veraltetes Linux-System, das manchmal für Hacking-Übungen verwendet wird.)

```
nmap -sV -O 10.0.0.36

   Nmap scan report for 10.0.0.36
   Host is up (0.00025s latency).
   Not shown: 977 closed ports
   PORT     STATE SERVICE     VERSION
   21/tcp   open  ftp         vsftpd 2.3.4
   22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (2.0)
   23/tcp   open  telnet      Linux telnetd
   25/tcp   open  smtp        Postfix smtpd
   53/tcp   open  domain      ISC BIND 9.4.2
   80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
   111/tcp  open  rpcbind     2 (RPC #100000)
   139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (WORKGROUP)
   445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (WORKGROUP)
   ...
   8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
   8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1

   MAC Address: 08:00:27:6D:C8:74 (Oracle VirtualBox virtual NIC)
   Device type: general purpose
   Running: Linux 2.6.X

   OS CPE: cpe:/o:linux:linux_kernel:2.6
   OS details: Linux 2.6.9 - 2.6.33
   Network Distance: 1 hop
   Service Info: Hosts:  metasploitable.localdomain, localhost,
     irc.Metasploitable.LAN; OSs: Unix, Linux;
     CPE: cpe:/o:linux:linux_kernel

   Nmap done: 1 IP address (1 host up) scanned in 65.95 seconds
```

Varianten und Alternativen

Auch wenn `nmap` der vermutlich universellste und populärste Netzwerk-Scanner ist, so gibt es doch unzählige Alternativen: Diese Kommandos sind im Hinblick auf bestimmte Netzwerkprotokolle oder -verfahren optimiert, arbeiten besonders schnell oder besonders unauffällig etc. Die folgende Liste gibt ohne Anspruch auf Vollständigkeit einige Beispiele:

- `fping`: schnelles `ping` für viele IP-Adressen
- `ikescan`: Netzwerk-Scanner für Virtual Protected Networks (VPNs) auf IPsec-Basis
- `masscan`: besonders schneller Netzwerk-Scanner, der nur ausgewählte Ports testet
- `netdiscover`: WLAN-Netzwerk-Scanner

- `p0f`: analysiert den Netzwerkverkehr und liefert Informationen (sogenannte *Fingerprints*) zu allen externen Rechnern. Bemerkenswert an `p0f` ist, dass es selbst keinen Netzwerkverkehr initiiert und daher unbemerkt bleibt.
- `smbtree`: listet alle Windows- und Samba-Server im Netzwerk bzw. in einer Workgroup auf

Eine Menge weiterer Programme finden Sie, wenn Sie in Kali Linux durch das Submenü des Punkts Informationsbeschaffung blättern.

Es gibt auch grafische Benutzeroberflächen zu `nmap`. In der Vergangenheit war das Programm Zenmap am beliebtesten. Dann schlitterte das Projekt in die Krise. Der Code basierte auf Python 2; diese Python-Version ist aber komplett veraltet und wird von den meisten gängigen Linux-Distributionen nicht mehr ausgeliefert. Erst Ende 2022 wurde eine Portierung auf Python 3 fertig. Seither genießt Zenmap wieder den Status als quasi-offizielle Oberfläche zu `nmap`. Eine gute Alternative ist `nmapsi4`.

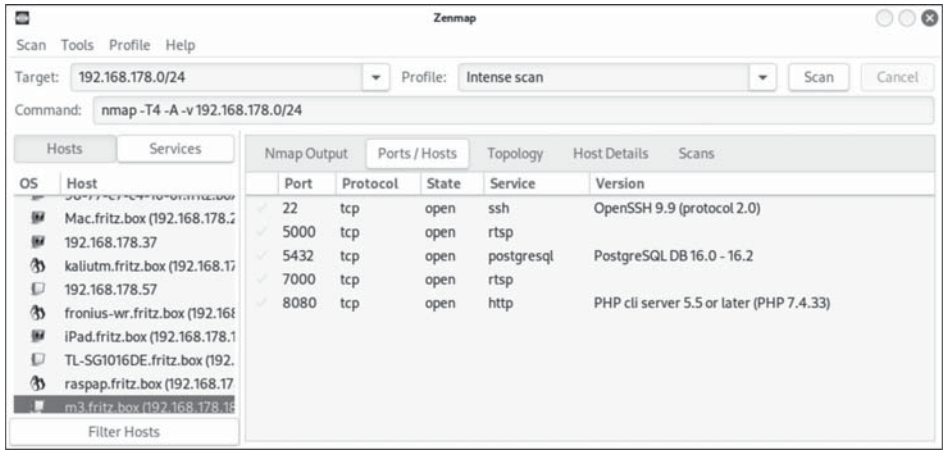

Abbildung 3.1 Die beliebteste grafische Oberfläche zu nmap ist Zenmap.

Darüber hinaus gibt es natürlich weitere Netzwerk-Scanner mit Benutzeroberfläche, sowohl als Open-Source-Software als auch in Form kommerzieller Programme. Bekannte Vertreter sind *Angry IP Scanner*, *Advanced IP Scanner* (nur Windows), *Qualys FreeScan* (Webservice) und *SuperScan* (nur Windows).

Scanner in Metasploit

Wenn Sie in der Metasploit Console arbeiten, sollten Sie statt `nmap` das Kommando `db_nmap` verwenden. Es greift auf `nmap` zurück, speichert die Ergebnisse aber in einer Datenbank und vereinfacht und beschleunigt so die weitere Verwendung der ermittelten Informationen.

Das Metasploit Framework enthält neben `db_nmap` diverse Module mit speziellen Netzwerk- und Service-Scannern. `search scanner` generiert eine Liste mit mehreren Hundert Modulen. Wenn es Ihnen explizit um Port-Scanner geht, versuchen Sie es mit `search portscan`.

3.2 hydra

Das Kommando `hydra` ist ein *Network Login Cracker*. Das Programm versucht also, einen Login durchzuführen und dabei das unbekannte Passwort zu erraten. Wobei »erraten« eigentlich übertrieben ist – das Programm probiert einfach der Reihe nach Passwörter aus einer Textdatei aus, die Sie zur Verfügung stellen müssen. Deswegen spricht man oft auch von einem »Wörterbuchangriff«. Da bis heute manche Benutzer Passwörter wie »123456« oder »passwort« verwenden, ist `hydra` erschreckend oft erfolgreich.

Die Stärke von `hydra` besteht unter anderem darin, dass es die Login-Versuche parallelisiert in mehreren Threads durchführt und mit sehr vielen Netzwerkprotokollen zurechtkommt, z. B. mit FTP, HTTP(S), IMAP, MySQL, Microsoft SQL, POP3, PostgreSQL, SMTP, Telnet und VNC. `hydra` kann auch Logins in Webformularen versuchen (`GET`, `PUT`, `POST`). Welche Dienste `hydra` unterstützt, hängt davon ab, wie `hydra` kompiliert wurde. Um die von Ihrer Version unterstützten Dienste herauszufinden, starten Sie `hydra` einfach ohne Parameter.

Syntax

Sie rufen `hydra` wie folgt auf:

```
hydra optionen hostname/ip-adresse service
```

Die folgende Auflistung erläutert die wichtigsten Optionen. Mehr Details gibt wie üblich die `man`-Seite.

- `-6`: verwendet nach Möglichkeit IPv6
- `-C datei`: verwendet die in der Datei angegebenen Kombinationen aus Login-Name und Passwort. Die Logins und Passwörter müssen zeilenweise in der Form `login:passwort` angegeben sein.
- `-e nsr`: probiert zusätzlich ein leeres Passwort (`n` wie *null*), den Login-Namen als Passwort (`s` wie *same*) und den umgekehrten Login-Namen (`r` wie *reverse*)
- `-f`: beendet Hydra, sobald eine gültige Login-Passwort-Kombination gefunden wurde
- `-l loginname`: verwendet den angegebenen Login-Namen

- `-L datei`: liest die Login-Namen zeilenweise aus der angegebenen Textdatei
- `-m optionen`: übergibt zusätzliche Optionen, die spezifisch für den Netzwerkdienst gelten. Zulässige Optionen können Sie mit `hydra -U dienst` ermitteln, also z. B. mit `hydra -U http-get`.
- `-M datei`: liest die anzugreifenden Hostnamen bzw. IP-Adressen aus der Datei und greift alle Hosts parallel an
- `-o datei`: speichert die erfolgreichen Login-Passwort-Kombinationen in der angegebenen Datei anstatt in der Standardausgabe
- `-p password`: verwendet das angegebene Passwort
- `-P datei`: probiert die Passwörter aus der angegebenen Textdatei der Reihe nach aus
- `-R`: setzt den zuletzt unterbrochenen `hydra`-Aufruf fort, sofern es die Datei `hydra.restore` gibt. Es müssen keine weiteren Optionen angegeben werden, diese sind in `hydra.restore` enthalten.
- `-s portnr`: verwendet den angegebenen Port anstelle des Default-Ports des jeweiligen Dienstes
- `-t n`: führt `n` Tasks (Threads) parallel aus. Die Standardeinstellung lautet 16. Das kann zu hoch sein, weil manche Dienste bei zu vielen parallelen Anfragen (noch dazu von derselben IP-Adresse) den Login blockieren.
- `-x min:max:chars`: generiert Passwörter, die zwischen `min` und `max` Zeichen lang sind und die die angegebenen Zeichen enthalten. Dabei gilt `a` als Kurzschreibweise für Kleinbuchstaben, `A` für Großbuchstaben und `1` für Ziffern. Alle anderen Zeichen, unter anderem die deutschen Buchstaben ä, ö, ü und ß, müssen einzeln angegeben werden.

  Beispiel: Mit `-x '4:6:aA1-_$%'` verwendet `hydra` Passwörter, die vier bis sechs Zeichen lang sind und neben Buchstaben und Ziffern auch die Zeichen `-`, `_`, `$` und `%` enthalten. Mit `-x '4:4:1'` probiert `hydra` alle vierstelligen Zahlen. Das ergibt 10.000 Möglichkeiten.

  Die Option `-x` ist nur in Ausnahmefällen sinnvoll, nämlich wenn Sie (fast) unendlich viel Zeit haben und Ihr Ziel unbegrenzt viele Login-Versuche toleriert.

Als `service` können diverse Netzwerk- oder Serverdienste wie `ssh`, `cisco`, `ftp`, `mysql` etc. verwendet werden (siehe die Dokumentation mit `man hydra`).

Passwortlisten

Vernünftig abgesicherte Server bzw. Netzwerkdienste lassen den Angreifer nicht beliebig lang Logins ausprobieren. Vielmehr wird die IP-Adresse des Angreifers nach einigen erfolglosen Versuchen für einige Zeit gesperrt. Oft wird zudem automatisch

eine E-Mail mit einer Einbruchswarnung an den Administrator versandt. Am ehesten ist eine hydra-Attacke zielführend, wenn Sie mit den erfolgversprechenden Passwörtern beginnen. Dazu benötigen Sie möglichst aktuelle und für das Zielpublikum passende Passwortlisten. Geeignete Listen finden Sie im Internet zuhauf, wenn Sie nach »password list« suchen. Wir beschränken uns hier exemplarisch auf zwei Seiten: Die Wikipedia enthält Listen der Top-25-Passwörter über die letzten Jahre und zeigt, was (erschreckend wenig) sich von Jahr zu Jahr ändert. Die GitHub-Seite enthält wesentlich umfassendere Passwortlisten mit bis zu 10 Millionen Einträgen:

https://en.wikipedia.org/wiki/List_of_the_most_common_passwords
https://github.com/danielmiessler/SecLists/tree/master/Passwords/Common-Credentials

Beachten Sie aber, dass die meisten Listen aus dem englischen Sprachraum stammen. So ist qwerty auf einer US-Tastatur ein naheliegendes Passwort, auf einer deutschen Tastatur aber keineswegs. Sie können natürlich auch mit einem Editor eine englische Top-100-Liste mit populären deutschen Passwörtern anreichern. Zum Hacking sollten Sie Passwortlisten verwenden, in denen die Passwörter nicht alphabetisch geordnet sind, sondern nach ihrer Häufigkeit.

Beispiele

Das folgende Kommando versucht, einen MySQL-Login für root auf einer Installation von Metasploitable 2 im lokalen Netzwerk durchzuführen. Dabei werden die Passwörter aus der Datei top_10000.txt ausprobiert. Die Passwortliste stellt sich allerdings als überflüssig heraus, der Root-Login hat schlicht gleich gar kein Passwort. In diesem Fall zeigt hydra einfach den verwendeten Login-Namen, aber eben kein Passwort an:

```
hydra -l root -e nsr -P top_10000.txt 10.0.0.36 mysql

   [INFO] Reduced number of tasks to 4 (mysql does not
          like many parallel connections)
   [DATA] max 4 tasks per 1 server, overall 64 tasks,
          10003 login tries (l:1/p:10003), ~39 tries per task
   [DATA] attacking service mysql on port 3306
   [3306][mysql] host: 10.0.0.36   login: root
   1 of 1 target successfully completed, 1 valid password found
```

Im zweiten Beispiel versucht hydra, auf einem Linux-Server einen Account mit trivialem oder gar keinem Passwort für einen SSH-Login zu finden. Dazu erzeugen Sie zuerst auf einem Linux-Rechner (idealerweise auf einem, auf dem dieselbe Distribution wie auf dem Zielrechner läuft) eine Liste mit allen Linux-Systemaccounts, indem Sie die Datei /etc/passwd auslesen:

```
cut -d: -f1 /etc/passwd > logins.txt
```

Anschließend soll hydra für alle in logins.txt gespeicherten Accounts einen SSH-Login ausprobieren, wobei als Passwort der Accountname, der umgedrehte Accountname sowie eine leere Zeichenkette verwendet werden. Die IP-Adresse 10.0.0.36 ist wieder die einer virtuellen Maschine mit Metasploitable im lokalen Netzwerk. Die Attacke bleibt allerdings erfolglos:

```
hydra -L logins.txt -e nsr 10.0.0.36 ssh

   [WARNING] Many SSH configurations limit the number of
             parallel tasks, it is recommended to reduce
             the tasks: use -t 4

   [DATA] max 16 tasks per 1 server, overall 64 tasks,
          165 login tries (l:55/p:3), ~0 tries per task
   [DATA] attacking service ssh on port 22
   1 of 1 target completed, 0 valid passwords found
```

Angriffe auf Webformulare und Login-Seiten

hydra wird auch dazu verwendet, GET- oder POST-Requests an einen Webserver zu senden, beispielsweise um ein Login-Formular anzugreifen. Dazu gibt es Dienstnamen wie http-get-form oder https-post-form. Eine zusätzliche Zeichenkette beschreibt dann, welche Daten bzw. Parameter übergeben werden sollen. Diese Zeichenkette besteht normalerweise wie im folgenden Beispiel aus drei Teilen, die durch Doppelpunkte getrennt sind:

```
/test/login.php:name=^USER^&pwd=^PASS^:Login error
```

- Der erste Teil gibt die anzugreifende Adresse an (relativ zum Hostnamen).
- Der zweite Teil gibt an, welche Parameter übergeben werden sollen. hydra ersetzt dann ^USER^ durch den Benutzernamen und ^PASS^ durch das Passwort.
- Der dritte Teil enthält Informationen, wie hydra erkennen kann, ob der Login erfolgreich war oder nicht. Normalerweise wird hier einfach eine Zeichenkette angegeben, die die Webseite im Falle eines fehlerhaften Logins anzeigt.

  Alternativ können Sie die Zeichenkette auch in der Form F=text angeben, wobei F für *failure* steht. Wenn der Login umgekehrt an einem Text zu erkennen ist, der normalerweise nach einem erfolgreichen Login im Browser angezeigt wird, können Sie diesen Text in der Form S=text angeben (S wie *success*).

Ein vollständiges Kommando sieht z. B. so aus:

```
hydra -L emails.txt -P pws.txt -o result.txt eine-firma.de \
  https-form-post \
  "/admin/login.php:email=^USER^&password=^PASS^:Login-Fehler"
```

In der Praxis ist die Durchführung von Brute-Force-Angriffen auf Login-Seiten freilich nicht ganz so einfach, wie es hier aussieht. Der erste Schritt besteht darin, dass Sie mit den Entwicklerwerkzeugen eines Webbrowsers oder mit Web-Analyse-Werkzeugen wie Burp ergründen, wie die Namen der erforderlichen Formularfelder oder Parameter lauten. Idealerweise verfügen Sie als Angreifer über einen gültigen Login (z. B. für einen Demoaccount), damit Sie das Verhalten der Seite sowohl im Fehlerfall als auch bei einem erfolgreichen Login ausprobieren können.

Oft scheitert der Angriff dann aber an Schutzmechanismen der Webseite. Moderne Seiten verlangen die Übergabe weiterer Parameter (Tokens), die dynamisch in das Login-Formular eingebaut werden (oft mit JavaScript) und die nur einmal gültig sind.

Schließlich sind viele Webseiten gegen wiederholte Login-Versuche abgesichert und blockieren nach einer bestimmten Anzahl fehlerhafter Versuche die Kommunikation. Weitere Informationen zum Angriff und zur Absicherung von Webseiten finden Sie in Abschnitt 3.13, »Burp Suite«, sowie in Kapitel 17, »Sicherheit von Webanwendungen«. Soweit sich eine Website nicht selbst gegen Brute-Force-Angriffe schützt, kann dies durch externe Programme wie Fail2ban erfolgen (siehe Abschnitt 15.6).

Alternativen

Wenn Sie im Internet nach Password-Crackern suchen, stoßen Sie unweigerlich auf diverse Alternativen zu `hydra`. Die beiden populärsten sind `ncrack` und `medusa`. Beide sind standardmäßig in Kali Linux installiert. Einen leider nicht mehr aktuellen Vergleich aller drei Tools finden Sie hier:

http://foofus.net/goons/jmk/medusa/medusa-compare.html

- **ncrack:** `ncrack` ist ein Passwort-Cracker aus der `nmap`-Familie. `ncrack` unterstützt zwar viel weniger Protokolle als `hydra`. Das Kommando ist dafür besonders einfach anzuwenden: Wenn Sie es beispielsweise in der Form `ncrack -v 10.0.0.36:22` aufrufen, dann versucht `ncrack` mit gängigen Accountnamen und Passwörtern einen SSH-Login (Port 22) beim Rechner mit der angegebenen IP-Adresse. `ncrack` verwendet dabei beim Kompilieren vorgegebene Account- und Passwortlisten. Sie können aber über Optionen eigene Listen angeben. Im laufenden Betrieb können Sie mit verschiedenen Tasten den Feedback-Level des Programms steuern. Drücken Sie einfach [?], um eine Liste der Tastenbefehle aufzurufen.
- **medusa:** Das Kommando `medusa` bietet einen ähnlichen Funktionsumfang wie `hydra`. Erfreulicherweise lauten auch viele Optionen gleich wie bei `hydra`. Der Vorteil von `medusa` liegt im modularen Design, das eine relativ einfache Ergänzung um weitere Protokolle möglich macht. `medusa -d` listet alle zur Verfügung stehenden Module auf. `medusa -M modulname -q` liefert Detailinformationen zum angegebenen Modul. Sie können das Kommando beispielsweise so anwenden:

```
medusa -M postgres -h 10.0.0.36 -u postgres -P top-10000.txt
```

Damit versucht `medusa` mit allen Passwörtern aus `top-10000.txt` einen Login beim PostgreSQL-Server auf dem Rechner 10.0.0.36 für den Benutzernamen `postgres`. Wie üblich können Sie Details zu den zahlreichen Optionen des Kommandos mit `man medusa` nachlesen.

Auch Metasploit enthält Werkzeuge zum Erraten von Passwörtern. Ein Beispiel ist `auxiliary/scanner/ssh/ssh_login`. Dieses Modul ist allerdings viel weniger flexibel als die zuvor vorgestellten Werkzeuge und kommt ausschließlich mit SSH zurecht.

Passwort-Hashes knacken

In Kapitel 7, »Passwörter«, stellen wir Ihnen eine Reihe weiterer Werkzeuge zum Erraten von Passwörtern vor. Diese Tools setzen in der Regel voraus, dass die Hash-Codes der Passwörter in einer lokalen Datei vorliegen. Man spricht dann vom sogenannten *Offline-Cracking*. Das ermöglicht ein wesentlich schnelleres Ausprobieren als über Netzwerkverbindungen. Kommandos wie `hashcat` greifen dabei auf die Grafikkarte zurück, was den Prozess erheblich beschleunigt.

3.3 sslyze, sslscan und testssl

Das verschlüsselte Protokoll HTTPS ist in den letzten Jahren zur Selbstverständlichkeit geworden. Aber mit der Umstellung des Webservers auf HTTPS ist es nicht getan: Immer mehr Verschlüsselungsverfahren und -bibliotheken gelten als veraltet, bei einigen sind sogar gravierende Sicherheitsmängel festgestellt worden (Heartbleed-Bug).

sslscan und sslyze

Bei der Überprüfung der HTTPS-Konfiguration helfen Kommandos wie `sslscan` oder `sslyze`, die beide standardmäßig unter Kali Linux installiert sind. Wie die folgenden Beispiele zeigen, ist ihre Anwendung unkompliziert. Aus Platzgründen sind die Ausgaben jeweils stark gekürzt wiedergegeben:

```
sslscan pi-buch.info:443

  SSL/TLS Protocols:
    SSLv2     disabled
    SSLv3     disabled
    TLSv1.0   disabled
    TLSv1.1   disabled
    TLSv1.2   enabled
```

```
    TLSv1.3   enabled
  TLS Fallback SCSV:
    Server supports TLS Fallback SCSV
  TLS renegotiation:
    Session renegotiation not supported
  TLS Compression:
    Compression disabled
  Heartbleed:
    TLSv1.3 not vulnerable to heartbleed
    TLSv1.2 not vulnerable to heartbleed
  Supported Server Cipher(s):
    Preferred TLSv1.3  128 bits  TLS_AES_128_GCM_SHA256       ...
    Accepted  TLSv1.3  256 bits  TLS_AES_256_GCM_SHA384       ...
    Accepted  TLSv1.3  256 bits  TLS_CHACHA20_POLY1305_SHA256 ...
    Preferred TLSv1.2  256 bits  ECDHE-RSA-AES256-GCM-SHA384  ...
    Accepted  TLSv1.2  256 bits  DHE-RSA-AES256-GCM-SHA384    ...
    ...
  Server Key Exchange Group(s):
    TLSv1.3  128 bits  secp256r1 (NIST P-256)
    TLSv1.3  192 bits  secp384r1 (NIST P-384)
    ...
    TLSv1.2  224 bits  x448
  SSL Certificate:
    Signature Algorithm: sha256WithRSAEncryption
    RSA Key Strength:    2048
    Altnames: DNS:pi-buch.info, DNS:www.pi-buch.info
    Not valid before: Mar  7 04:37:44 2025 GMT
    Not valid after:  Jun  5 04:37:43 2025 GMT
    ...

sslyze pi-buch.info:443

 ...
 * TLS 1.2 Cipher Suites:
   Attempted to connect using 156 cipher suites.
   The server accepted the following 5 cipher suites:
   TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 256  ...
   The group of cipher suites supported by the server has
   the following properties:
   Forward Secrecy        OK - Supported
   Legacy RC4 Algorithm   OK - Not Supported
 * TLS 1.3 Cipher Suites:
   Attempted to connect using 5 cipher suites.
   The server accepted the following 3 cipher suites:
    TLS_CHACHA20_POLY1305_SHA256   256   ECDH: X25519 (253 bits)
```

```
   TLS_AES_256_GCM_SHA384         256   ECDH: X25519 (253 bits)
   TLS_AES_128_GCM_SHA256         128   ECDH: X25519 (253 bits)
 * Deflate Compression:     OK - Compression disabled
 * OpenSSL CCS Injection:   OK - Not vulnerable to OpenSSL
                                CCS injection
 * Downgrade Attacks:
   TLS_FALLBACK_SCSV:       OK - Supported
 * OpenSSL Heartbleed:      OK - Not vulnerable to Heartbleed
 * ROBOT Attack:            OK - Not vulnerable, RSA cipher
                                suites not supported.

 COMPLIANCE AGAINST MOZILLA TLS CONFIGURATION

 Checking results against Mozilla's "MozillaTlsConfigurationEnum.
 INTERMEDIATE" configuration. See https://ssl-config.mozilla.org/
 for more details.

 pi-buch.info:443: FAILED - Not compliant.
     * tls_curves: TLS curves {'X448', 'secp521r1'} are
                   supported, but should be rejected.
```

Im obigen Listing stört sich `sslyze` daran, dass der Webserver zwei veraltete Verschlüsselungsverfahren unterstützt, die nach aktuellen Richtlinien abgelehnt werden sollten. Für Nachbesserungsarbeiten können Sie unter *https://ssl-config.mozilla.org* aktuelle SSL-Konfigurationsdateien für alle gängigen Webserver (Apache, nginx etc.) erzeugen. Nach einem Neustart des Webservers probieren Sie es noch einmal:

```
sslyze pi-buch.info:443

 ...
 COMPLIANCE AGAINST MOZILLA TLS CONFIGURATION

 pi-buch.info:443: OK - Compliant.
```

testssl

Eine Alternative zu `sslscan` und `sslyze` ist das Shell-Script `testssl`. In Kali Linux können Sie es mit `apt install testssl` installieren und dann unter dem Namen `testssl` ausführen.

Das Script `testssl` überprüft die SSL-Konfiguration auf alle bekannten Schwachstellen. Die Ausgabe des Kommandos erstreckt sich über rund 200 Zeilen. Die Ausgaben sind zudem grün, gelb oder rot gekennzeichnet, je nachdem, ob die Konfiguration in Ordnung ist oder ob Probleme entdeckt wurden. Das folgende Listing kann hier nur stark gekürzt und leider ohne Farben wiedergegeben werden.

```
testssl pi-buch.info

  Testing protocols via sockets except SPDY+HTTP2
     SSLv2      not offered (OK)
     SSLv3      not offered (OK)
     TLS 1      not offered
     TLS 1.1    not offered
     TLS 1.2    offered (OK)
     TLS 1.3    offered (OK): final
     NPN/SPDY   not offered
     ALPN/HTTP2 http/1.1 (offered)
  Testing standard cipher categories
    NULL ciphers (no encryption)                  not offered (OK)
    Anonymous NULL Ciphers (no authentication)    not offered (OK)
    Export ciphers (w/o ADH+NULL)                 not offered (OK)
    LOW: 64 Bit + DES encryption (w/o export)     not offered (OK)
    ...
  Testing robust (perfect) forward secrecy, (P)FS ...
    PFS is offered (OK)       ECDHE-RSA-AES256-GCM-SHA384 ...
    Elliptic curves offered:  prime256v1 secp384r1 secp521r1 ...
    DH group offered:         RFC3526/Oakley Group 14 (2048 bits)
  Testing vulnerabilities
    Heartbleed (CVE-2014-0160)             not vulnerable (OK)
    CCS (CVE-2014-0224)                    not vulnerable (OK)
    Ticketbleed (CVE-2016-9244), exper.    not vulnerable (OK)
    ...
    LUCKY13 (CVE-2013-0169), experimental  not vulnerable (OK)
    RC4 (CVE-2013-2566, CVE-2015-2808)     no RC4 ... (OK)
  Running client simulations via sockets
    Android 4.4.2  TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, ...
    Android 5.0.0  TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, ...
    ...
```

Mit der zusätzlichen Option `--log` oder `--html` speichert `testssl` das Ergebnis in einer einfachen Textdatei bzw. in einer HTML-Datei. Der Dateiname setzt sich aus dem Hostnamen und der Uhrzeit zusammen.

Online-Tests

Während der Aufruf von `sslscan`, `sslyze` oder `testssl` schnell erledigt ist, ist die Interpretation der Ergebnisse schon schwieriger: Welche Verschlüsselungsalgorithmen gelten als unsicher, welche sollten deaktiviert werden, und bei welchen Uralt-Browsern könnte das zu Problemen führen? Diese Fragen beantwortet die Webseite *https://www.ssllabs.com/ssltest* in einem detaillierten Prüfbericht samt konkreten Empfehlungen zur Verbesserung der Konfiguration (siehe Abbildung 3.2):

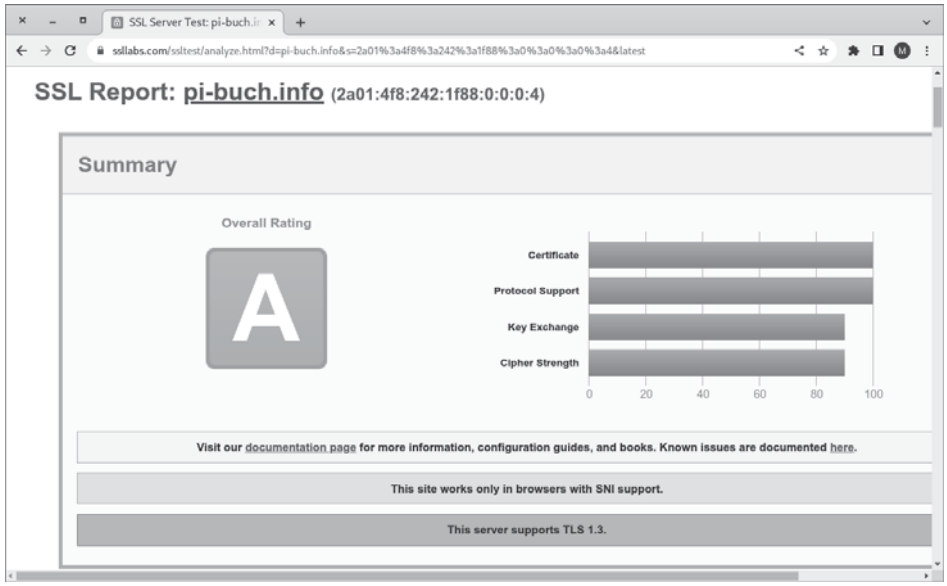

Abbildung 3.2 »ssllabs.com« bietet einen ausgezeichneten Online-Test der SSL-Konfiguration.

3.4 whois, host und dig

Wer ist der Domain-Administrator einer Website? Wie lautet die Mailserver-Adresse der Domäne? Welcher Hostname ist der IP-Adresse 1.2.3.4 zugeordnet? Solche und ähnliche Fragen können mitunter Tools beantworten, die Informationen von *Domain-Name-Servern* (DNS) auswerten. Die so ermittelten Informationen waren bis vor kurzem öffentliche Daten, die sich beim regulären Betrieb einer Website oder anderer Internetdienste nicht verheimlichen ließen. Was ein Angreifer mit derartigen Informationen tun kann, beschreibt sehr anschaulich Kapitel 12, »Penetration-Testing«.

Ein Informationsschatz versiegt

Mittlerweile ist der Datenschutzgedanke auch in viele Nameserver-Organisationen (sogenannte *Registrys*) vorgedrungen. Natürlich ist die Zuordnung zwischen Hostnamen und IP-Adressen weiter öffentlich – sonst würde das Internet nicht mehr funktionieren. Aber Informationen darüber, wer einen Hostnamen registriert hat (samt Adresse, Telefonnummer und E-Mail), wer für die Administration verantwortlich ist etc. lassen sich immer seltener mit einem simplen `whois`-Kommando ermitteln. Wie reich die Datenausbeute ist, hängt stark von der jeweiligen Domänenkennung und der dazugehörenden Registry ab.

whois

Der »Klassiker« unter den DNS-Kommandos ist `whois`:

```
whois derstandard.at

  domain:         derstandard.at
  registrant:     SVM13418489-NICAT
  source:         AT-DOM
  ...
  organization:   Standard Verlagsgesellschaft m.b.H.
  ...
```

Anstatt `whois` selbst auszuführen, können Sie auch eine der zahlreichen Websites in Anspruch nehmen (z. B. *https://whois.com*), die nach der Eingabe des Hostnamens in ein Formular dieselben Informationen liefern.

host

`host` liefert die IP-Adresse zum angegebenen Hostnamen bzw. den Hostnamen zur angegebenen IP-Adresse. Zusätzlich verrät das Kommando weitere Informationen, z. B. den Hostnamen des Mailservers (also den MX-Eintrag).

```
host kofler.info

  kofler.info has address 168.119.33.110
  kofler.info has IPv6 address 2a01:4f8:242:1f88::4
  kofler.info mail is handled by 10 mail.kofler.info.

host 2a01:4f8:242:1f88::4

  4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.8.f.1.2.4.2.0.8.f.4.0.1.0.a.\
  2.ip6.arpa domain name pointer host1.kofler.info.
```

Noch ausführlicher fallen die Informationen aus, wenn Sie die Option `-a` (*all*) übergeben. Um nur die Einträge eines bestimmten Typs abzufragen, geben Sie diesen mit `-t` an. Beispielsweise liefert `host -t txt` alle Texteinträge, die unter anderem zur Veröffentlichung der SPF-, DKIM- und DMARC-Informationen des Mailservers dienen.

dig

Zumeist liefern `whois` und `host` bereits ausreichend viele Informationen; wollen Sie aber noch gründlicher und zielgerichteter suchen, dann müssen Sie sich mit den unzähligen Optionen des `dig`-Kommandos anfreunden.

Wenn Sie an `dig` einfach einen Hostnamen übergeben, liefert das Kommando den A-Record in der nicht gerade lesefreundlichen Notation des DNS-Servers `bind`. Kommentare werden dabei mit Strichpunkten eingeleitet.

```
dig bsi.de

  ; <<>> DiG 9.18.1-1-Debian <<>> bsi.de
  ;; global options: +cmd
  ;; Got answer:
  ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64392
  ;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ...
  ;; OPT PSEUDOSECTION:
  ; EDNS: version: 0, flags:; udp: 65494
  ;; QUESTION SECTION:
  ;bsi.de.                IN   A    80.245.144.218
  ;; Query time: 48 msec
  ;; SERVER: 127.0.0.53#53(127.0.0.53)
  ;; WHEN: Tue Jun 28 08:35:04 CEST 2022
  ;; MSG SIZE  rcvd: 35
```

Wenn Sie nur am MX-Record interessiert sind und gleichzeitig die Informationsfülle reduzieren möchten, rufen Sie `dig` wie folgt auf:

```
dig bsi.de MX +short

  10 mx2.bund.de.
  10 mx1.bund.de.
```

Um anstelle des Default-DNS-Servers Ihres Rechners einen anderen Server zu befragen, geben Sie dessen IP-Adresse mit `@` explizit an:

```
dig @8.8.4.4 TXT kofler.info +short

  "v=spf1 a mx ~all"
```

Mit `dig -f datei` können Sie Anfragen für mehrere Hostnamen zugleich verarbeiten, wobei Sie die Hostnamen vorweg in einer Datei speichern müssen.

dnsrecon

Das Kommando `dnsrecon` hilft dabei, Subdomänen zu finden, deren Name unbekannt ist. Ein Anwendungsbeispiel für `dnsrecon` finden Sie in Abschnitt 12.3, »Scanning von interessanten Zielen«.

3.5 Wireshark

Die Open-Source-Oberfläche *Wireshark* (ehemals *Ethereal*) ist ein Analyseprogramm für Netzwerkprotokolle. Das Programm verfolgt den gesamten Netzwerkverkehr einer Schnittstelle, analysiert ihn, zerlegt ihn in zusammenhängende Teile und zeigt ihn »live« an (siehe Abbildung 3.3).

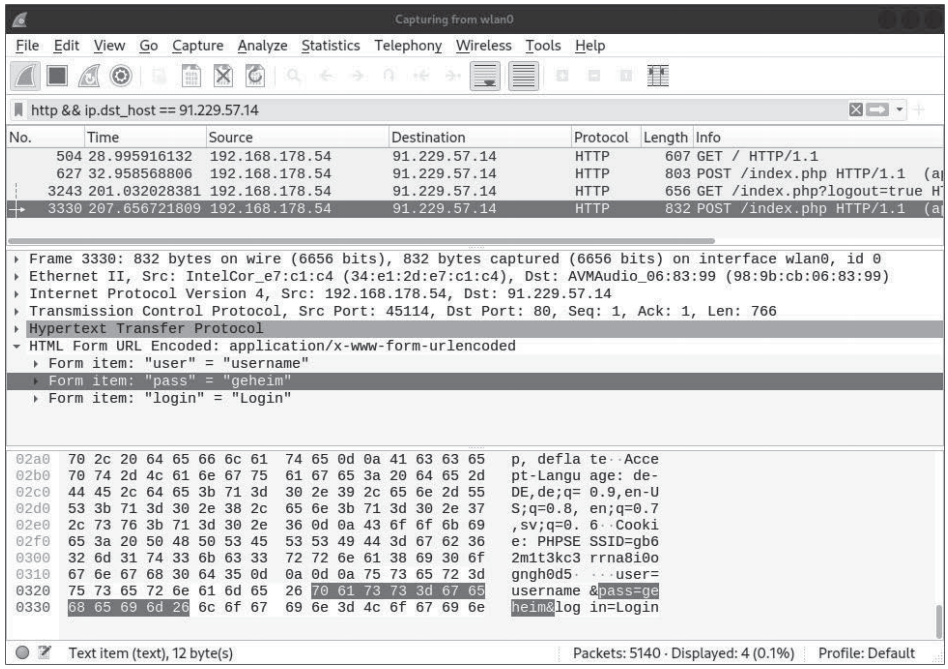

Abbildung 3.3 Wireshark, das hier auf einem als WLAN-Access-Point konfigurierten Raspberry Pi läuft, hat einen HTTP-Post-Request aufgezeichnet. Das Passwort ist im Klartext zu lesen.

Da eine direkte Beobachtung bei den hohen anfallenden Datenmengen zumeist unmöglich ist, bietet das Programm einerseits die Möglichkeit, den Datenfluss sehr gezielt nach bestimmten Paketen zu filtern (z. B. nach allen HTTP-Requests oder nach IP-Adressen), andererseits können Sie die Daten zur späteren Analyse speichern.

Für Wireshark gibt es naturgemäß viele Anwendungsmöglichkeiten: Entwickler ergründen damit die Funktionsweise von Netzwerkprotokollen und suchen nach Fehlern in eigenen Programmen. Netzwerkadministratoren können das Programm einsetzen, um nach verdächtigen Datenpaketen zu suchen, die auf eine Malware auf dem Rechner oder auf eine Backdoor in einem Programm schließen lassen.

Für Angreifer ist Wireshark vor allem dann eine Goldgrube, wenn es gelingt, das Programm auf einem Gateway für den Internetzugang des Ziels auszuführen. Eine denkbare Vorgehensweise besteht darin, dass der Angreifer einen kostenlosen WLAN-Hotspot anbietet und darauf hofft, dass sein Ziel diesen verwendet. Praktischerweise kann Wireshark auch auf dem Raspberry Pi installiert werden, der sich für derartige Anwendungen gut eignet.

Wireshark stößt an seine Grenzen, wenn der Netzwerkverkehr verschlüsselt ist. Das gilt z. B. für HTTPS-, SSH- oder VPN-Verbindungen. Wireshark kann zwar natürlich

auch solche Pakete samt allen Metadaten anzeigen, aufgrund der Verschlüsselung ist der Inhalt solcher Pakete aber nicht im Klartext lesbar.

Speicherbedarf

Wireshark benötigt ausreichend Arbeitsspeicher. Wenn Sie Wireshark in einer virtuellen Maschine mit Kali Linux ausführen, benötigt die virtuelle Maschine zumindest 4 GiB RAM.

Installation

Unter Kali Linux ist Wireshark bereits standardmäßig installiert. Für die meisten anderen Linux-Distributionen steht Wireshark als Paket zur Verfügung, das je nach Distribution mit `apt` oder `dnf` installiert werden kann. Downloads für Windows und macOS finden Sie hier:

https://www.wireshark.org/download.html

Bei der Installation unter Linux erscheint je nach Distribution eine Rückfrage (siehe Abbildung 3.4), ob das Programm nur durch `root` verwendet wird (das war früher der Normalfall) oder ob es auch von gewöhnlichen Benutzern ausgeführt werden darf, wenn diese Mitglied der Gruppe `wireshark` sind. Die zweite Variante hat den Vorteil, dass weniger Code mit Root-Rechten ausgeführt werden muss. Entscheiden Sie sich also für Yes. Bei Bedarf können Sie diese Konfiguration unter Kali Linux, Debian und Ubuntu mit `dpkg-reconfigure wireshark-common` wiederholen.

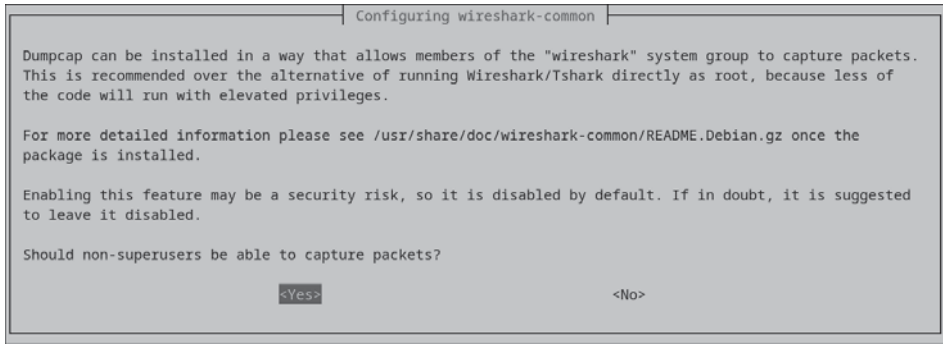

Abbildung 3.4 Rückfrage bei der Installation von Wireshark unter Debian- oder Ubuntu-basierten Distributionen

Als Administrator müssen Sie die betreffenden Benutzer nun zur `wireshark`-Gruppe hinzufügen. Unter Kali Linux ist dies bereits standardmäßig der Fall. (Überzeugen Sie sich mit `groups`, dass die Gruppe `wireshark` aufgelistet wird.) Bei anderen Distributionen fügen Sie den gewünschten Benutzer zu dieser Gruppe hinzu:

```
sudo usermod -a -G wireshark <accountname>
```

Dieser Benutzer muss sich nun aus- und neu einloggen und kann dann Wireshark direkt verwenden. Weitere Details zu diesem Modus können Sie in `/usr/share/doc/wireshark/README.Debian` sowie auf der folgenden Webseite nachlesen:

https://wiki.wireshark.org/CaptureSetup/CapturePrivileges

Grundfunktionen

Beim Start des Programms wählen Sie per Doppelklick die Netzwerkschnittstelle aus, die Sie überwachen möchten. In der Folge erscheint eine rasch wachsende Liste von Paketen im obersten Bereich des Wireshark-Fensters. Wählen Sie eines der Pakete aus, dann zeigt der zweite Bereich des Fensters Metadaten zum Netzwerkpaket (Frame-Größe, Paket- und Protokolltyp, Sender- und Empfängeradresse usw.), der dritte Bereich die eigentlichen Daten in hexadezimaler Form sowie als Text.

Die Datenflut ist anfänglich überwältigend. Nun gilt es, durch Anzeigefilter genau die Daten auszuwählen, die Sie tatsächlich sehen möchten. Dazu geben Sie in der Zeile unterhalb der Buttonleiste den Suchausdruck an. Wenn der Suchausdruck syntaktisch korrekt ist, wird die Eingabezeile grün unterlegt, sonst rot. Der Filterausdruck wird ähnlich wie eine `if`-Bedingung in einer Programmiersprache formuliert (siehe auch Tabelle 3.1).

Beispielsweise bedeutet der folgende Ausdruck, dass Wireshark nur Pakete anzeigen soll, die dem Protokoll HTTP entsprechen und bei denen die Quell- oder die Ziel-Adresse eine bestimmte IP-Adresse ist:

```
http && (ip.dst_host == 1.2.3.4 || ip.src_host == 1.2.3.4)
```

Ausdruck	Bedeutung
`tcp`, `udp`, `http`, `ftp`, `ssh` etc.	Wählt ein Protokoll aus.
`tcp.port == ...`	Wählt einen Port aus.
`ip.dst_host == ...`	Gibt die Zieladresse an.
`ip.src_host == ...`	Gibt die Quelladresse an.
`ipv6.xxx == ...`	IPv6-Attribute
`http.xxx == ...`	HTTP-Attribute
`&&`	UND-Verknüpfung
`||`	ODER-Verknüpfung

Tabelle 3.1 Aufbau von Ausdrücken für den Wireshark-Anzeigefilter

Wireshark bietet schier endlose Möglichkeiten, Bedingungen für jede Art von IP-Paketen, Protokollen und Zuständen zu formulieren. Zur Eingabe können Sie über das Analyze-Menü den Dialog Display Filter Expression öffnen (siehe Abbildung 3.5). In der deutschen Lokalisierung erreichen Sie den Dialog über Analyse • Anzeigefilterausdruck. Er listet die unzähligen Wireshark bekannten Parameter auf, geordnet nach Protokolltyp.

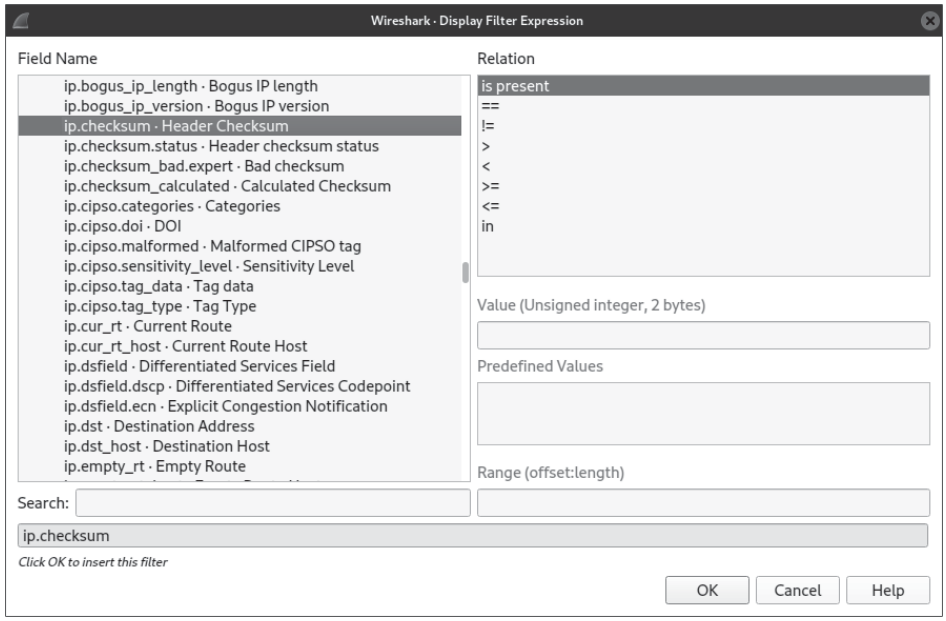

Abbildung 3.5 Ein Dialog hilft bei der Auswahl aus Hunderten von Filterparametern.

Eine weitere Eingabeerleichterung ist direkt in die Benutzeroberfläche von Wireshark integriert: Wenn Sie in der Oberfläche eine IP-Adresse, einen Protokollnamen oder andere Informationen mit der rechten Maustaste anklicken, können Sie per Kontextmenü den Filter entsprechend erweitern.

Farbenspiele

Wireshark färbt die Zeilen der Paketübersicht standardmäßig in unterschiedlichen Farben. Wenn Ihnen die Farben zu unübersichtlich sind, können Sie die Farbdarstellung mit dem Button Draw packets using your coloring rules unkompliziert deaktivieren und ebenso schnell wieder aktivieren.

Welche Farbe für welche Art von Paket gilt, geht aus View • Coloring Rules hervor. In diesem Dialog können Sie die Regeln auch ändern, eigene Regeln hinzufügen und das Regel-Set speichern bzw. laden.

Aktuelle Versionen von Kali Linux verwenden den *Dark Mode*, das heißt, alle Elemente von grafischen Oberflächen werden in dunklen Farben dargestellt. Das entspricht nicht nur dem Zeitgeist, sondern auch allen Hacker-Vorurteilen. Wireshark übernimmt diese Einstellung.

Wenn Sie Wireshark wie in den Abbildungen dieses Buchs im *Light Mode* ausführen möchten, reicht es nicht aus, Kali Linux mit Settings • Appearance in den *Light Mode* umzustellen. Das liegt daran, dass Wireshark, abweichend von anderen Programmen, die Qt-Bibliothek verwendet. Die Farbpalette der Qt-Programme muss extra im Programm `qt6ct` eingestellt werden – dann klappt es.

Arbeitstechniken

Auf einem frequentierten Netzwerkknoten fallen in kurzer Zeit riesige Datenmengen an. Wireshark zeichnet die Daten wie ein Recorder auf und behält sie im Arbeitsspeicher. Das setzt natürlich voraus, dass genug RAM zur Verfügung steht. In der Praxis ist es zweckmäßig, die Aufnahme möglichst rasch mit dem roten Stopp-Button zu beenden. Danach kann die Aufzeichnung in Ruhe analysiert werden.

Mit File • Save können Sie eine abgeschlossene Aufnahme zur späteren Analyse speichern. Dabei sollten Sie das Wireshark-eigene Format in `*.pcapng`-Dateien verwenden und zusätzlich die Option Compress with gzip auswählen. Später können Sie die Datei mit File • Open wieder laden, durchaus auch auf einem anderen, leistungsstärkeren Rechner. Wireshark kann auch Dateien analysieren, die mit anderen Programmen aufgezeichnet wurden, beispielsweise mit `tcpdump` oder *Sniffer* (siehe den folgenden Abschnitt, »Alternativen«).

Um die Datenmengen von vornherein zu reduzieren, bietet Wireshark die Möglichkeit, bereits die Aufnahme zu filtern. Dazu wählen Sie unter Capture • Capture Filter einen Filter aus (siehe Abbildung 3.6). Das reduziert zwar ganz erheblich die Datenmengen, die sich Wireshark merken muss, allerdings ist die Entscheidung im Gegensatz zum vorhin erläuterten Anzeigefilter endgültig: Was nicht aufgezeichnet ist, kann später auch nicht mehr eingeblendet werden, wenn sich herausstellt, dass zur Analyse doch noch weitere Pakete erforderlich wären.

Dokumentation und Hilfe

Wireshark bietet weit mehr Analysemöglichkeiten, als dieser Abschnitt andeutet. Wenn Sie sich intensiver mit dem Programm auseinandersetzen möchten, lohnt ein Blick in zwei oder drei Video-Anleitungen auf YouTube. Darüber hinaus geht an der umfassende Online-Dokumentation kein Weg vorbei:

https://www.wireshark.org/docs

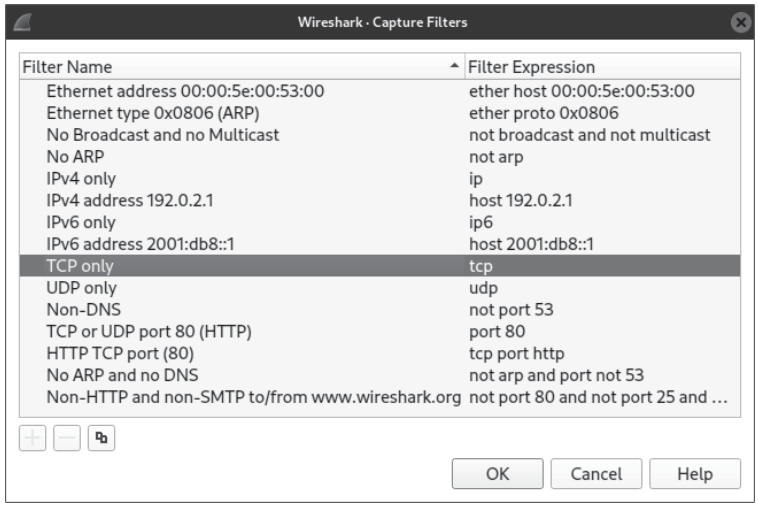

Abbildung 3.6 Die Einstellung eines Filters reduziert die aufgezeichneten Datenmengen, in diesem Fall auf das TCP-Protokoll.

Alternativen

Wireshark ist zwar das bekannteste Programm innerhalb seiner Kategorie, es gibt aber natürlich jede Menge Alternativen. Kostenlos, aber nur im Microsoft-Universum ausführbar ist der *Microsoft Network Monitor*. Das Programm ist bis heute beliebt, obwohl die Entwicklung 2010 mit Version 3.4 beendet wurde.

https://www.microsoft.com/en-us/download/4865

Speziell auf HTTP- und HTTPS-Verkehr optimiert ist Programm *Fiddler*. Fiddler agiert als Proxyserver und kann die analysierten Programme auch manipulieren, z. B. um Man-in-the-Middle-Angriffe durchzuführen. Fiddler war in der Vergangenheit kostenlos, aktuelle Versionen erfordern aber ein kostenpflichtiges Abo, deren billigster Tarif 9 $ pro Monat beträgt (Stand: Herbst 2025).

https://www.telerik.com/fiddler

Für Freunde der Kommandozeile gibt es natürlich auch diverse Tools, die zumindest Teilaufgaben von Wireshark erfüllen:

- Die Kommandos `tcpdump` und `ngrep` (siehe Abschnitt 3.6, »tcpdump«) filtern die über eine Netzwerkschnittstelle fließenden TCP-, UDP- oder ICMP-Pakete und zeichnen die gewünschten Pakete in einer Datei auf. Diese können Sie später mit einem anderen Werkzeug analysieren, z. B. mit Wireshark.
- `ettercap` kann trotz einer viel einfacheren Oberfläche ähnlich wie Wireshark dazu verwendet werden, aus dem Netzwerkverkehr interessante Informationen (Passwörter etc.) herauszufiltern. Die eigentliche Spezialität des Programms ist aber die Durchführung von Man-in-the-Middle-Angriffen (siehe Abschnitt 12.12).

3.6 tcpdump

Das Kommando `tcpdump` liest den Netzwerkverkehr einer Schnittstelle mit, filtert ihn nach Kriterien, zeigt ihn am Bildschirm an oder speichert ihn in einer Datei. Anders als der Name vermuten lässt, kommt das Kommando nicht nur mit TCP-Paketen, sondern auch mit UDP- und ICMP-Paketen zurecht. `tcpdump` greift intern auf die `pcap`-Bibliothek zurück, um die Netzwerkpakete auszulesen und zu filtern.

`tcpdump` ist unter macOS standardmäßig installiert. Die meisten Linux-Distributionen stellen das Kommando im gleichnamigen Paket zur Verfügung. Unter Windows installieren Sie das mit `tcpdump` kompatible Programm *WinDump*.

https://www.tcpdump.org
https://www.winpcap.org/windump

Syntax

`tcpdump` muss mit Root-Rechten ausgeführt werden. Ohne weitere Optionen zeigt es Metadaten zu allen Netzwerkpaketen »live« auf dem Bildschirm an, bis das Kommando mit Strg+C beendet wird. Die Syntax von `tcpdump` sieht so aus:

```
tcpdump [optionen] [filterausdruck]
```

Mit Optionen geben Sie an, was `tcpdump` tun soll:

- `-a`: zeigt Paketinhalte in Textform an (ASCII)
- `-c <n>`: beendet das Programm nach `n` Paketen
- `-i <schnittstelle>`: berücksichtigt nur Pakete, die über die angegebene Schnittstelle fließen. Eine Liste der in Frage kommenden Schnittstellen erzeugen Sie mit `tcpdump -D`.
- `-n`: zeigt IP-Adressen statt Hostnamen an
- `-q`: zeigt weniger Informationen an (*quiet*)
- `-r <datei>`: liest die Pakete aus einer zuvor mit `tcpdump -w` gespeicherten Datei
- `-w <datei>`: speichert die Pakete in binärer Form (*raw*) in die angegebene Datei. Die Datei kann später durch `tcpdump -r` oder von anderen Programmen, z. B. von Wireshark (siehe Abschnitt 3.5), wieder gelesen und ausgewertet werden.
- `-x`: zeigt Paketinhalte in hexadezimaler Form an

Den Optionen kann ein Filterausdruck folgen, der unter anderem aus den folgenden Schlüsselwörtern zusammengesetzt sein kann. Darüber hinaus gibt es aber zahlreiche weitere Filtermöglichkeiten, die in `man pcap-filter` vollständig beschrieben sind.

- `greater <n>`: berücksichtigt nur Pakete, die größer als `n` Byte sind

- `host <ipadr>` oder `host <hostname>`: berücksichtigt nur Pakete, die die angegebene IP-Adresse bzw. den entsprechenden Host als Quelle oder als Ziel verwenden
- `less <n>`: berücksichtigt nur Pakete, die kleiner als `n` Byte sind
- `net <cidr>`: berücksichtigt nur Pakete, deren Quelle oder Ziel dem angegebenen Adressbereich in CIDR-Notation entspricht (z. B. `10.0.0.0/24`)
- `port <n>` oder `portrange <n1-n2>`: berücksichtigt nur Pakete, die die angegebenen Port-Nummern als Quelle oder Ziel verwenden
- `proto ether|fddi|tr|wlan|ip|ip6|arp|rarp|decnet|tcp|udp`: berücksichtigt nur Pakete des angegebenen Protokolls. Dem Ausdruck kann `ip` oder `ip6` vorangestellt werden, wenn nur IPv4- bzw. IPv6-Pakete analysiert werden sollen (z. B. `ip6 proto udp`).

Den Schlüsselwörtern `host`, `net` und `port` kann wahlweise `dst` oder `src` vorangestellt werden, wenn sich die Angabe nur auf das Paketziel bzw. die Paketquelle bezieht.

Mehrere Filterbedingungen verknüpfen Sie mit `and` oder `or`. Komplexe Ausdrücke müssen Sie mit `\(` und `\)` klammern. Alternativ können Sie auch einfache Klammern verwenden, dann müssen Sie aber den gesamten Filterausdruck in Apostrophe stellen (z. B. `'(port 1 or port 2)'`).

Beispiele

Das folgende Kommando gibt Informationen über alle HTTP-Pakete aus, die über die Schnittstelle `wlan0` fließen:

```
tcpdump -i wlan0 port 80
  tcpdump: verbose output suppressed, use -v or -vv for full
    protocol decode
  listening on wlan0, link-type EN10MB (Ethernet)
  10:34:33.681218 IP imac.57402 > bpf.tcpdump.org.http:
    Flags [S], seq 755525464, win 65535,
    options [mss 1460,nop,wscale 5,nop,nop,
             TS val 595975353 ecr 0,sackOK,eol], length 0
  10:34:33.681793 IP imac.57403 > bpf.tcpdump.org.http:
    Flags [S], seq 2954861158, win 65535, ...
```

Das zweite Kommando zeichnet die nächsten 100 HTTP-Pakete in der Datei `dump.pcap` auf, die von der oder zur IP-Adresse 192.139.46.66 fließen. Das Kommando ist hier aus Platzgründen über zwei Zeilen verteilt. Es ist einzeilig und ohne das `\`-Zeichen auszuführen:

```
tcpdump -i wlan0 -n -c 100 -w dump.pcap \
  port 80 and host 192.139.46.66
```

Das dritte Kommando zeigt HTTP- und HTTPS-Pakete an, die an den Host `cert.org` adressiert sind oder von dort kommen:

```
tcpdump -i wlan0 host cert.org and \( port 80 or port 443 \)
```

Gleichwertig wäre das folgende Kommando:

```
tcpdump -i wlan0 'host cert.org and (port 80 or port 443)'
```

ngrep

Eine interessante Alternative zu `tcpdump` ist das Kommando `ngrep`. Es greift wie `tcpdump` auf die `pcap`-Bibliothek zurück, berücksichtigt aber darüber hinaus den Inhalt der Pakete. Das funktioniert naturgemäß nur bei nicht verschlüsselten Protokollen, also z. B. bei HTTP oder FTP. Die meisten Linux-Distributionen stellen `ngrep` im gleichnamigen Paket zur Verfügung. Für das Kommando gilt die folgende Syntax:

```
ngrep [optionen] [grep-suchausdruck] [pcap-filterausdruck]
```

Dabei geben Sie mit dem `grep`-Suchausdruck das Suchmuster an, nach dem Sie in den Paketen suchen. Das Muster ist ein regulärer Ausdruck (siehe `man 7 regex`). Für den `pcap`-Filterausdruck zur Auswahl der Pakete gelten dieselben Regeln, die wir im vorigen Abschnitt schon für `tcpdump` erläutert haben.

Die wichtigsten Optionen sind:

- `-d <schnittstelle>|any`: gibt die Netzwerkschnittstelle an
- `-i`: ignoriert die Groß- und Kleinschreibung im `grep`-Suchausdruck
- `-v`: invertiert die Suche. `ngrep` liefert nur die Pakete, in denen das `grep`-Suchmuster *nicht* erkannt wurde.
- `-w`: interpretiert den `grep`-Suchausdruck als Wort
- `-W byline`: berücksichtigt bei der Ausgabe Zeilenumbrüche, was zu besser lesbaren Ausgaben führt

Im Gegensatz zu `tcpdump` kann `ngrep` die gefundenen Pakete allerdings nicht in einer Wireshark-kompatiblen Form aufzeichnen.

Das folgende Beispiel lauscht auf allen Schnittstellen nach HTTP-Paketen, in denen die Schlüsselwörter `user`, `pass` usw. vorkommen:

```
ngrep -d any -i 'user|pass|pwd|mail|login' port 80

  interface: any
  filter: (ip or ip6) and ( port 80 )
  match: user|pass|pwd|mail|login
  T 10.0.0.87:58480 -> 91.229.57.14:80 [AP]
  POST /index.php HTTP/1.1..Host: ...
    user=name&pass=geheim&login=Login ...
```

3.7 Netcat (nc)

Das Programm *Netcat* (Kommandoname `nc`, Windows-Version `nc.exe`) verarbeitet und transportiert Netzwerkdaten über die Standardeingabe bzw. Standardausgabe. Das Kommando kann z. B. dazu verwendet werden, interaktiv Netzwerkprotokolle wie HTTP oder SMTP auszuprobieren oder Dateien oder Streams zu übertragen.

Bei einigen Linux-Distributionen ist das Kommando `nc` im gleichnamigen Paket enthalten, bei anderen Distributionen müssen Sie `netcat` installieren. Beachten Sie, dass es unterschiedliche Implementierungen von Netcat gibt. So kommt unter Kali Linux, Debian und Ubuntu `netcat-traditional` zum Einsatz, während RHEL eine Variante der `nmap`-Entwickler anbietet (Paket `nmap-ncat`, siehe *https://nmap.org/ncat*). In der Praxis ergeben sich daraus keine großen Unterschiede, allerdings sind möglicherweise einzelne Optionen je nach Version anders (oder gar nicht) implementiert.

Netcat ist kein dediziertes Hacking-Werkzeug, aber wegen seiner universellen Einsatzmöglichkeiten eignet es sich wie viele andere Tools aus diesem Kapitel auch für Hacking-Aufgaben.

Syntax

`nc` zeichnet sich durch eine simple Syntax aus:

```
nc [optionen] [hostname/ip-adresse] [port]
```

Darauf aufbauend gibt es unzählige Optionen, von denen wir hier nur einen Bruchteil beschreiben. Werfen Sie einen Blick in die `man`-Seiten (also `man nc`).

- `-4` oder `-6`: verwendet ausschließlich IPv4 oder IPv6
- `-l`: wartet für den angegebenen Port auf einen Verbindungsaufbau (*listen*)
- `-p <port>`: gibt den lokalen Port (Source-Port) an. Der üblicherweise am Ende des `nc`-Kommandos angegebene Port ist hingegen der Ziel-Port (Destination-Port).
- `-x <proxyadr:port>`: verwendet die angegebene Proxyadresse und den dazugehörenden Port

Beispiele

Im einfachsten Fall verwenden Sie `nc` interaktiv anstelle von `telnet`, um mit einem externen Server im Textmodus zu kommunizieren. Sie können auf diese Weise z. B. ergründen, welche Authentifizierungsverfahren ein Mailserver unterstützt. Im folgenden Listing sind die durchgeführten Eingaben mit `<==` gekennzeichnet.

```
nc kofler.info smtp

  220 host1.kofler.info ESMTP Postfix (Ubuntu)
  ehlo test  <==
```

```
250-host1.kofler.info
250-PIPELINING
250-SIZE 20480000
250-ETRN
250-STARTTLS
250-AUTH PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN

quit <==
221 2.0.0 Bye
```

Um eine Datei über einen beliebigen Port (hier 1234) von Host 1 nach Host 2 zu kopieren, starten Sie zuerst auf Host 2 den Empfänger und initiieren die Übertragung der Datei dann auf Host 1.

```
                                    host2$ nc -l 1234 > datei
host1$ nc host2 1234 < datei
```

Hürde Firewall

Bei Ports, die größer als 1024 sind, sind zur Ausführung von `nc` keinerlei besondere Rechte erforderlich. Das obige Beispiel sowie die folgenden Beispiele funktionieren allerdings nur, wenn es einen freien Port gibt, der nicht durch eine Firewall blockiert ist.

Wenn Sie sich mit einer zweiten Person unkompliziert und ohne die Installation eines Chat-Programms austauschen möchten, müssen Sie und Ihr Gesprächspartner sich lediglich auf einen Port einigen. Das folgende Listing zeigt in zwei Spalten die Kommunikation für Host 1 und 2. Der Chat wird auf einem Rechner mit `nc -l` initiiert. Damit überwacht `nc` den angegebenen Port 1234 und wartet auf einen Verbindungsaufbau.

Auf dem zweiten Rechner wird `nc` ohne Optionen gestartet, um die Verbindung zum ersten Host herzustellen. Eine sichtbare Bestätigung des Verbindungsaufbaus gibt es zwar nicht, aber sobald nun einer der beiden Gesprächspartner Text eingibt (Standardeingabe) und mit ↵ bestätigt, erscheint der Text im Terminal des anderen Gesprächspartners (Standardausgabe).

Im folgenden Listing sind Eingaben mit `<==` gekennzeichnet. Die Kommunikation endet, sobald ein Benutzer `nc` mit Strg+C schließt.

```
host1$ nc -l 1234
                                    host2$ nc host1 1234
wie geht's? <==
                                    wie geht's?
                                    gut <==
gut
<Strg>+<C>
```

Zur besseren Tarnung kann der Chat ebenso gut via UDP durchgeführt werden, also mit `nc -l -u 1234` und mit `nc -u host1 1234`.

Im folgenden Beispiel (Idee: *https://en.wikipedia.org/wiki/Netcat#Examples*) ersetzt `nc` einen minimalen Webserver. `nc` wartet auf Port 8080 auf einen Verbindungsaufbau. Findet dieser statt, sendet das Kommando zuerst ein HTTP-OK, dann die Länge des Dokuments und schließlich das Dokument selbst, also hier den Inhalt von `hello.html`. Das Kommando `nc` endet danach, d. h., die Seite kann nur ein einziges Mal abgerufen werden.

```
{ printf 'HTTP/1.0 200 OK\r\nContent-Length: %d\r\n\r\n' \
    "$(wc -c < hello.html)"; cat hello.html; } | nc -l 8080
```

Mit einer FIFO-Datei (*First In, First Out*) können Sie `nc` zum Webproxy machen. Das funktioniert allerdings nur für Webseiten, die noch HTTP verwenden und nicht sofort auf HTTPS umleiten.

```
mkfifo myfifo
nc -l 1234 < myfifo | nc hostname 80 > myfifo
```

Mit Einschränkungen funktioniert das auch für HTTPS-Verbindungen. Allerdings benötigen Sie nun zwei FIFO-Dateien, wobei die Kommunikation über `openssl` geleitet wird, weil Netcat SSL nicht unterstützt. Bei unseren Versuchen erwies sich die Verwendung von Netcat als SSL-Proxy allerdings nicht praxistauglich, die Verbindung brach rasch ab.

```
mkfifo f1
mkfifo f2
nc -l 1234 -k > f1 < f2 &
while true; do
  openssl s_client -connect kofler.info:443 -quiet < f1 > f2
done
```

Das Gefahrenpotenzial von Netcat zeigt sich im nächsten Beispiel: Hier wird `nc` auf Host 1 so eingerichtet, dass es alle auf Port 1234 empfangenen Eingaben an die Shell `bash` weitergibt. Deren Ausgaben werden wieder zurück an den Sender übertragen. Von einem zweiten Host (siehe die rechte Spalte im folgenden Listing) können nun Shell-Kommandos auf Host 1 ausgeführt werden:

```
host1$ nc -l 1234 -e /bin/bash
```

```
host2$ nc host1 1234
ls
  datei1
  datei2
  datei3
```

Die Option `-e` zur Ausführung eines Kommandos steht allerdings nicht bei allen Netcat-Versionen zur Verfügung. Sie fehlt insbesondere bei der unter Debian und Ubuntu üblichen `netcat-traditional`-Implementierung. Abhilfe: Installieren Sie das Paket `nmap`, und führen Sie das dort enthaltene Kommando `ncat` aus.

socat

`socat` ist eine Variante zum `nc`-Kommando. Die Projektwebsite beschreibt `socat` (*Socket Cat*) als `netcat++`. Das Kommando `socat` unterstützt auch das Protokoll SCTP, kann über Proxyserver arbeiten, auch serielle Schnittstellen bedienen und die Daten für die Übertragung verschlüsseln.

http://www.dest-unreach.org/socat
https://technostuff.blogspot.co.at/2008/10/some-useful-socat-commands.html

3.8 OpenVAS

OpenVAS steht für *Open Vulnerability Assessment System*; es ist ein Programm, das Sicherheitslücken auf Rechnern sucht. Umgangssprachlich nennt man derartige Programme »Security-Scanner« oder etwas exakter auch »Vulnerability-Scanner«.

Das bekannteste derartige Programm war lange Zeit *Nessus*. Auch Nessus war ursprünglich ein Open-Source-Projekt, seit 2005 wird es aber unter einer proprietären Lizenz weiterentwickelt. Es steht seither nur mehr kommerziellen Kunden zur Verfügung.

OpenVAS baut auf der letzten GPL-Version von Nessus auf. Das Programm wird nun von der Firma *Greenbone* weiterentwickelt. Weitere Informationen finden Sie auf der Website des Projekts:

https://openvas.org

Wie so viele andere Security-Tools beginnt auch OpenVAS mit einem Port-Scan für den oder die zu untersuchenden Rechner. Im nächsten Schritt versucht es auf vielfältige Weise zu erkennen, welche Programme in welcher Version den aktiven Ports zuzuordnen sind. Bis zu dieser Stelle agiert OpenVAS im Prinzip wie `nmap`, auch wenn die Programm- und Versionserkennung ausgeklügelter ist als bei `nmap`.

An dieser Stelle beginnen die Unterschiede zwischen OpenVAS und einem gewöhnlichen Port-Scanner: Im nächsten Schritt testet OpenVAS, ob die erkannten (oder auch nicht erkannten) Programme für bekannte Sicherheitslücken anfällig sind. OpenVAS kann dabei auf eine riesige Datenbank zurückgreifen, in der nicht nur die Beschreibung des jeweiligen Problems enthalten ist, sondern auch Module (NVTs, also *Network Vulnerability Tests*) zur Erkennung der Probleme.

Schließlich zeigt OpenVAS die erkannten Sicherheitslücken geordnet nach verschiedenen Kriterien an. Bei vielen Problemen führt ein Klick auf den jeweiligen Eintrag zu einer Anleitung, die bei der Behebung der Sicherheitslücke hilft.

OpenVAS erfordert einiges an Ressourcen bei der Installation und im laufenden Betrieb; das Programm ist aber über eine Weboberfläche selbst für Einsteiger relativ einfach zu bedienen.

Der schwierigste Aspekt ist sicherlich die richtige Einschätzung der angezeigten Sicherheitswarnungen. Die Sicherheitsanforderungen für eine Website zum Online-Banking sind naturgemäß ungleich höher als die für einen Webserver, der für eine private WordPress-Installation dient. Entsprechend pedantisch werden Sie den Ratschlägen von OpenVAS folgen, wenn die Sicherheitsanforderungen sehr hoch sind.

Empfehlung der Redaktion :-)

Aus unserer Sicht zählt OpenVAS zu den wertvollsten Tools zur Absicherung eines Windows- oder Linux-Servers. OpenVAS erkennt nicht nur sehr viele Probleme, sondern ist auch eine große Hilfe bei ihrer Beseitigung. Das Programm fußt auf einem Open-Source-Projekt und steht in der Grundversion kostenlos zur Verfügung.

Allerdings folgt auch Greenbone (also die Firma hinter OpenVAS) dem *Freemium*-Modell: Die Software als solche ist zwar kostenlos, der Zugang auf die Scan-Datenbanken unterliegt hingegen Einschränkungen. In der kostenlosen Version können Sie lediglich nach Sicherheitslücken in *consumer grade software* suchen, wobei Greenbone eine genaue Auflistung schuldig bleibt, was darunter zu verstehen ist. Bei unseren Tests erwies sich OpenVAS auf jeden Fall als große Hilfe bei der Absicherung von typischen Web- und Mail-Servern auf Linux-Basis.

Um auch nach Sicherheitslücken in kommerzieller Software zu suchen (dazu zählen unter anderem Programme von Broadcom/VMware, Cisco, Microsoft oder Oracle), müssen Sie ein Upgrade auf *Greenbone Basic* durchführen. Die jährlichen Kosten betrugen zuletzt rund 2500 €/ Jahr:

https://www.greenbone.net/en/test_now/

Bei allem Lob für OpenVAS sollten Ihnen auch die Grenzen von OpenVAS bewusst sein: Selbst wenn OpenVAS keine Probleme entdeckt, bedeutet das keineswegs, dass der Zielrechner tatsächlich sicher ist! Einerseits gibt es natürlich Sicherheitslücken, die in der OpenVAS-Datenbank nicht oder noch nicht enthalten sind. Andererseits sucht OpenVAS nur nach bekannten Sicherheitslücken, aber beispielsweise nicht nach zu einfachen Passwörtern. OpenVAS kann sehen, wie sich ein Zielrechner nach außen präsentiert, aber es kann natürlich nicht in diesen »hineinsehen«.

Und noch einen Nachteil müssen wir festhalten: OpenVAS ist ein Ressourcenfresser. Sicherheits-Scans erfordern wegen ihrer langen Ausführdauer nicht nur viel Geduld, sondern lassen während der Wartezeit auch die CPU heiß laufen.

Installation

Sie können OpenVAS als fertige virtuelle Maschine (basierend auf Debian Linux) oder in Form von Paketen für Fedora/RHEL bzw. für Ubuntu Linux herunterladen.

Platzbedarf und Voraussetzungen

Beachten Sie, dass der Platzbedarf für OpenVAS mehr als 5 GByte beträgt. Das betrifft sowohl die zu installierenden Pakete als auch diverse Datenbanken von Sicherheitslücken.

OpenVAS benötigt außerdem eine Menge Arbeitsspeicher. Falls Sie OpenVAS in einer virtuellen Maschine installieren, müssen Sie ihr genug Speicher zuordnen (zumindest 6 GByte).

Bei Netzwerk-Scans profitiert OpenVAS schließlich von möglichst vielen CPU-Cores.

Wir haben die Installation von OpenVAS zuletzt im April 2025 unter Kali Linux ausprobiert. Während in den vergangenen Jahren wenig Probleme auftraten, war der Installationsprozess dieses Mal ziemlich mühsam und fehleranfällig. Alle weiteren Kommandos sind mit Root-Rechten auszuführen.

```
apt update
apt full-upgrade
apt autoremove
apt install openvas
apt autoclean
gvm-setup

  ...
  md manage:WARNING:2025-04-21 18h48.19 utc:59841: sql_open:
  PQerrorMessage (conn): connection to server on socket
```

```
"/var/run/postgresql/.s.PGSQL.5432" failed:
FATAL:  database "gvmd" does not exist
...
You can now run gvm-check-setup to make sure everything
is correctly configured
```

Das letzte Kommando endete mit Fehlermeldungen, dass die Datenbank für OpenVAS nicht eingerichtet werden konnte. Das empfohlene Script `gvm-check-setup` bestätigt den Fehler und gibt immerhin einen ersten Hinweis, was schiefgegangen ist. Es gibt offenbar ein Problem mit der Sortierordnung (*collation*) des PostgreSQL-Datenbank-Servers.

```
gvm-check-setup

  Step 1: Checking OpenVAS (Scanner)...
  Step 2: Checking GVMD Manager ...
  Step 3: Checking Certificates ...
  Step 4: Checking data ...
  Step 5: Checking Postgresql DB and user ...
          OK: Postgresql version and default port are OK.
  WARNING:  database "postgres" has a collation version mismatch
  DETAIL:  The database was created using collation version 2.40,
           but the operating system provides version 2.41.
  HINT:  Rebuild all objects in this database that use the
         default collation and run ALTER DATABASE postgres
         REFRESH COLLATION VERSION, or build PostgreSQL with the
         right library version.
  ERROR: The Postgresql DB does not exist.
  FIX: Run 'sudo runuser -u postgres --
            /usr/share/gvm/create-postgresql-database'
```

Der Troubleshooting-Guide enthält weitere Hintergrundinformationen zu diesem Problem. Offensichtlich enthält Kali eine zu neue Version von PostgreSQL.

https://greenbone.github.io/docs/latest/22.4/kali/troubleshooting.html#fixing-the-gvmd-collation-version-mismatch

Abhilfe schaffen ein manuelles Update der *collation version*, das neuerliche Einrichten der Datenbanken und schließlich eine Wiederholung von `gvm-setup`:

```
sudo -u postgres psql -d postgres -c \
  'ALTER DATABASE postgres REFRESH COLLATION VERSION;'

sudo -u postgres psql -d postgres -c \
  'ALTER DATABASE template1 REFRESH COLLATION VERSION;'
```

```
runuser -u postgres -- \
  /usr/share/gvm/create-postgresql-database

gvm-setup
  ...
  Please note the password for the admin user
  User created with password '0ecb324a-...'
```

`gvm-setup` gibt unzählige Statusmeldungen aus. Wirklich wichtig ist die letzte Zeile. Sie enthält das Passwort für den Benutzer `admin`, das Sie später für den Login in die Weboberfläche benötigen. Speichern Sie es!

Mit `gvm-check-setup` überprüfen Sie nun nochmals, ob die Installation erfolgreich war, und starten OpenVAS erstmalig.

```
gvm-check-setup

  ...
  It seems like your GVM-25.04.0 installation is OK.
```

Bei Bedarf können Sie später mit dem Kommando `gvmd` weitere Benutzer einrichten, ihr Passwort verändern, alle Benutzer auflisten etc. Beachten Sie, dass `gvmd` im User-Account `_gvm` ausgeführt werden muss.

```
sudo -u _gvm gvmd --create-user=user2
sudo -u _gvm gvmd --user=user2 --new-password=geheim
sudo -u _gvm gvmd --get-users
  admin
  user2
```

Solange Sie OpenVAS nur sporadisch zum Pen-Testing einsetzen, ist die Verwendung in einer virtuellen Maschine vollkommen ausreichend. Sie können OpenVAS freilich auch so einrichten, dass es z. B. ein gesamtes Firmennetzwerk regelmäßig auf Sicherheitslücken überprüft, automatisch Berichte über die vorgefundenen Probleme versendet etc. Der Scan eines großen Netzwerks ist mit erheblichem Zeit- und Ressourcenaufwand verbunden. In solchen Anwendungsszenarien empfiehlt es sich, OpenVAS auf einen dedizierten, gut ausgestatteten Server zu installieren.

OpenVAS ohne Installation verwenden

OpenVAS ist als Webdienst realisiert. Deswegen ist es technisch relativ einfach, einen OpenVAS-Zugang auf einer Webseite zu realisieren. Sowohl Greenbone als auch diverse andere Firmen bieten derartige (kostenpflichtige!) Cloud-Dienste an:

https://www.greenbone.net/en/cloud-service

OpenVAS starten und aktualisieren

Unmittelbar nach der Installation werden die OpenVAS-Hintergrunddienste automatisch gestartet. In Zukunft, d. h. nach einem Neustart von Kali Linux, ist das aber nicht mehr der Fall. Sie müssen OpenVAS bei Bedarf explizit starten:

```
gvm-start
```

`gvm-start` ist ein Script, das die Dienste `gsad`, `gvmd`, `ospd-openvas` sowie `notus-scanner` startet. Ein automatischer Start von OpenVAS beim Hochfahren von Kali Linux ließe sich natürlich einrichten, würde den Startprozess von Kali Linux aber unnötig verlängern und den Speicherbedarf des Systems vergrößern.

OpenVAS lädt während der Installation sämtliche zu diesem Zeitpunkt verfügbaren Module (NVTs) zur Erkennung von Sicherheitslücken herunter. Naturgemäß bleibt die Sicherheitswelt aber nicht stehen. Deswegen sollten Sie zumindest einmal wöchentlich `sudo greenbone-feed-sync` ausführen, um die jeweils neuesten Module und CERT-Informationen herunterzuladen. Wenn Sie regelmäßig OpenVAS nutzen, ist es sinnvoll, einen entsprechenden Cron-Job einzurichten.

Anwendung

Die Bedienung von OpenVAS erfolgt über eine Weboberfläche über Port 9392. Sie öffnen also in Kali Linux im Webbrowser die Seite *https://127.0.0.1:9392*. Da OpenVAS ein selbst signiertes Zertifikat verwendet, müssen Sie für dieses im Webbrowser eine Ausnahmeregel definieren und so zum Ausdruck bringen, dass Sie diesem Zertifikat vertrauen. Schließlich loggen Sie sich mit dem Benutzernamen `admin` und dem von `openvas-setup` angezeigten Passwort ein und gelangen so in die Weboberfläche (siehe Abbildung 3.7).

Die Startseite zeigt anfänglich kaum relevante Informationen. Das ändert sich, sobald Sie OpenVAS anwenden. Dann gibt das sogenannte Dashboard einen Überblick über die zuletzt durchgeführten Aktivitäten (Tasks).

Die Anwendung von OpenVAS ist denkbar einfach: Sie öffnen die Seite Scans • Tasks, klicken auf das Symbol Task Wizard und geben die IP-Adresse oder den Hostnamen des Rechners an, den Sie überprüfen möchten. Es sind auch Adressbereiche in der Form 10.0.0.1–10.0.0.99 oder 192.168.27.0/24 erlaubt.

OpenVAS beginnt nun einen Sicherheits-Scan. Der Scan kann, je nachdem, wie viele Dienste auf dem zu testenden Host laufen, durchaus eine Stunde oder länger dauern. Bei größeren IP-Adressbereichen können die Scans zwar zu einem gewissen Maß parallelisiert werden, aber Scans, die über mehrere Stunden laufen, sind nicht ungewöhnlich.

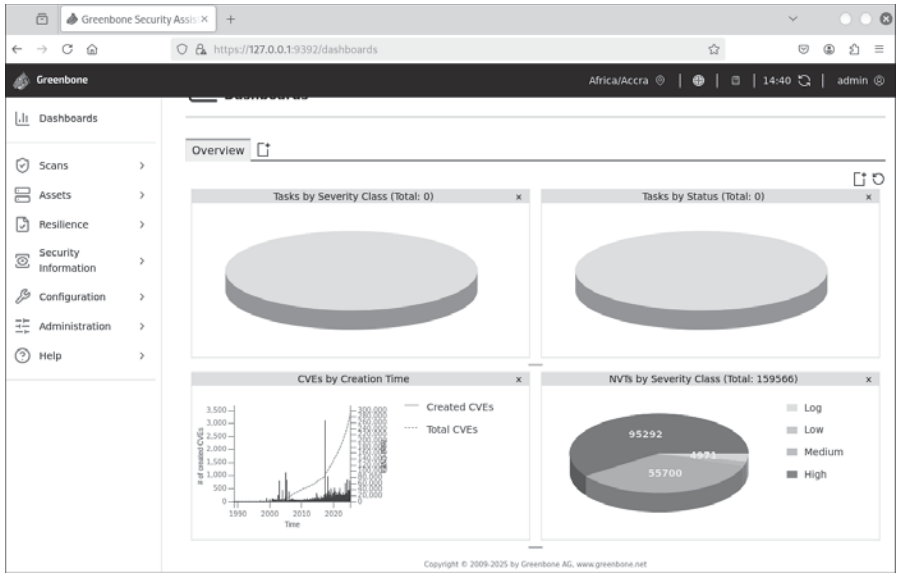

Abbildung 3.7 Die Startseite der OpenVAS-Oberfläche ist beim ersten Start relativ leer.

Die Seite Scans • Reports listet die Berichte aller abgeschlossenen Scans auf. Ein Klick auf die Datums- und Zeitspalte führt zu den Detailergebnissen (siehe Abbildung 3.8). Die Ergebnisseite ist auch bei noch laufenden Scans zugänglich und zeigt dann alle bisher gefundenen Ergebnisse. Ein unscheinbarer Upload-Button bietet die Möglichkeit, den Report in verschiedenen Formaten (PDF, XML) zu exportieren.

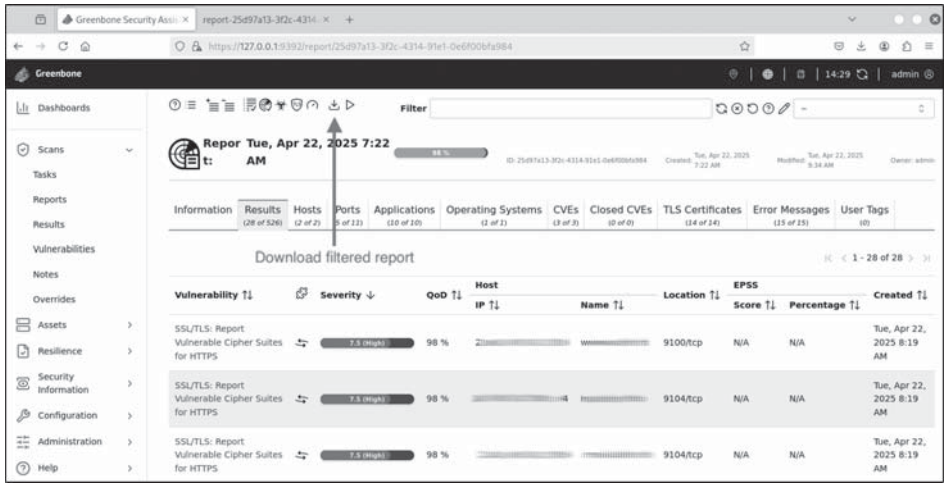

Abbildung 3.8 Ergebnis eines Scans eines Linux-Servers mit Web- und Maildiensten

Das Scan-Ergebnis enthält zumeist diverse Doppelgänger, wenn ein Server unter IPv4 und IPv6 sowie unter mehreren Hostnamen (*example.com*, *www.example.com*, *mail.example.com* etc.) erreichbar ist. Hilfreicher ist diesebezüglich die Ansicht SCANS • VULNERABILITIES, die jedes Problem nur einmal anzeigt (siehe Abbildung 3.9). Allerdings bezieht sich diese Ansicht nicht auf einen Scan, sondern auf alle bisher durchgeführten Scans. (Sie können die Liste aber nach verschiedenen Kriterien filtern.)

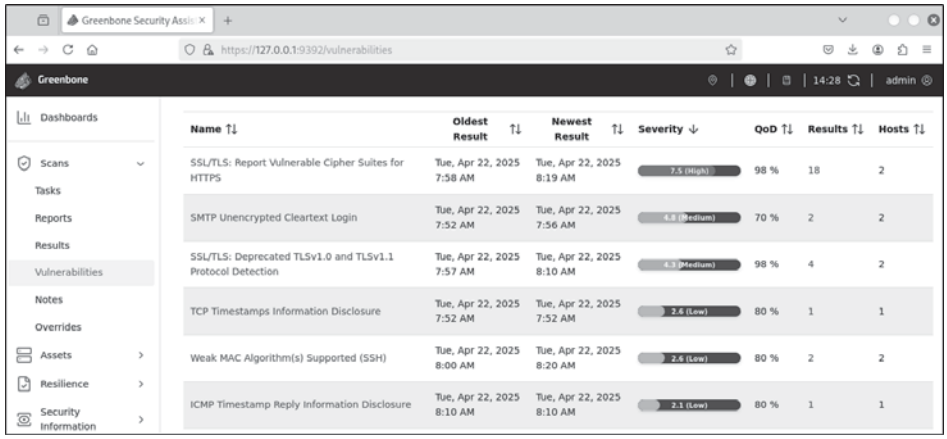

Abbildung 3.9 Auflistung aller gefundenen Sicherheitsprobleme

In der Ergebnisliste können Sie einzelne Punkte anklicken und so eine Detailbeschreibung des Problems aufklappen (siehe Abbildung 3.10). Teilweise finden Sie dort auch konkrete Konfigurationstipps mit Informationen, wie Sie das Problem beheben können. In anderen Fällen müssen Sie selbst recherchieren, wie Sie das Problem lösen können.

Severity und QoD

OpenVAS zeigt zusammen mit jedem festgestellten Problem zwei Parameter an. *Vulnerability* gibt an, wie groß die Gefahr durch die Sicherheitslücke ist. Dabei wird das *Common Vulnerability Scoring System* (CVSS) als Basis verwendet. Der Wertebereich liegt zwischen 0 und 10, wobei Werte ab `7.0` als hohe Gefahr betrachtet werden.

https://en.wikipedia.org/wiki/CVSS

QoD steht für *Quality of Detection* und beschreibt als Prozentwert, mit welcher Zuverlässigkeit das Problem festgestellt wurde. 80 % bedeutet beispielsweise, dass ein Versionstest durchgeführt werden konnte und die festgestellte Programmversion von der Sicherheitslücke betroffen ist. Werte ab 95 % bedeuten, dass die Lücke durch aktive Tests verifiziert werden konnte.

Eine genaue Aufschlüsselung der Werte finden Sie im OpenVAS-Handbuch, das Sie von der folgenden Seite als PDF-Datei herunterladen können:

https://docs.greenbone.net/

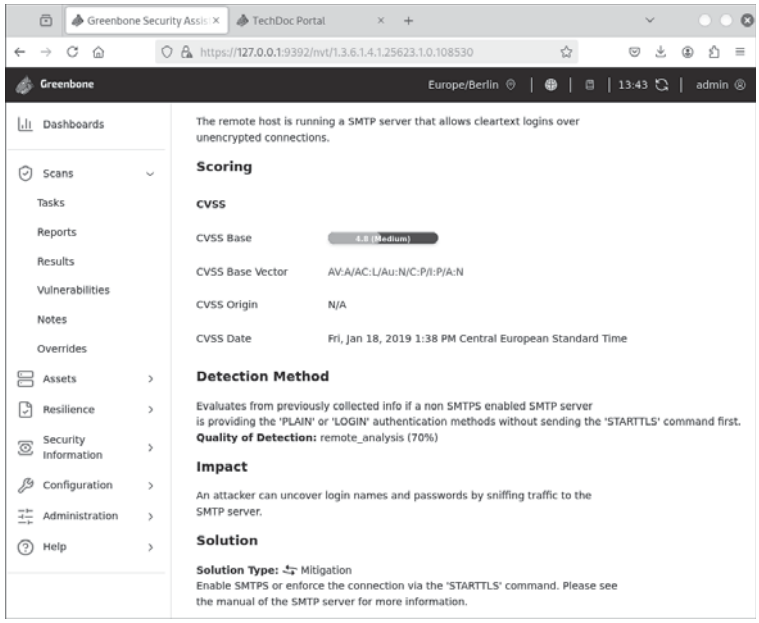

Abbildung 3.10 OpenVAS liefert zu jedem gefundenen Problem eine detaillierte Erklärung und häufig auch konkrete Hinweise zur Behebung.

Selten sind alle Sicherheitswarnungen oder -empfehlungen tatsächlich relevant. OpenVAS bietet deswegen vielfältige Möglichkeiten, die Ergebnisse zu filtern und die einmal definierten Filterregeln dauerhaft zu speichern. Die Benutzeroberfläche hilft bei der Einstellung der Filterregeln allerdings nur bedingt weiter. Für komplexere Regeln wird Ihnen ein Blick in das Handbuch nicht erspart bleiben.

Die Ergebnisse der Vulnerability-Scans werden dauerhaft in einer lokalen Datenbank gespeichert und stehen auch dann noch zur Verfügung, wenn Sie zwischenzeitlich die OpenVAS-Oberfläche verlassen, Kali Linux neu starten etc. Erst wenn es bei einem Scan mehrere Durchläufe gibt, werden ältere Ergebnisse nach und nach gelöscht. Die Weboberfläche bietet Ihnen aber die Möglichkeit, Scans explizit zu löschen.

Alive-Test

Standardmäßig testet OpenVAS nur durch ein simples ICMP-Ping, ob ein Zielrechner online ist. Wenn ein Scan überhaupt keine Resultate liefert, Sie sich aber sicher sind, dass der betreffende Rechner eingeschaltet und im Netz erreichbar ist, ist die

wahrscheinlichste Fehlerursache der zu simple *Alive-Test*. Bei manchen Servern ist aus Sicherheitsgründen ICMP deaktiviert.

Abhilfe schafft in so einem Fall die Veränderung des Alive-Tests in den Target-Eigenschaften des Scans. Um in der OpenVAS-Oberfläche in den richtigen Dialog zu gelangen, wählen Sie einen bereits durchgeführten Scan in Scan • Tasks aus und klicken in dessen Detailansicht auf den Link Target for … scan.

Damit gelangen Sie auf eine Seite, die die Details des Scan-Targets beschreibt. Dort führt der Button Edit Target in einen Dialog, in dem Sie einstellen können, wie der Alive-Test durchgeführt werden soll (siehe Abbildung 3.11). In besonders hartnäckigen Fällen wählen Sie den Eintrag Consider Alive – dann verzichtet OpenVAS auf jegliche Tests und geht einfach davon aus, dass der Zielrechner läuft. Die Default-Einstellung Scan Config Default bzw. ICMP Ping soll sicherstellen, dass ein Vulnerability-Scan über ein ganzes Netzwerk nicht länger dauert als unbedingt notwendig.

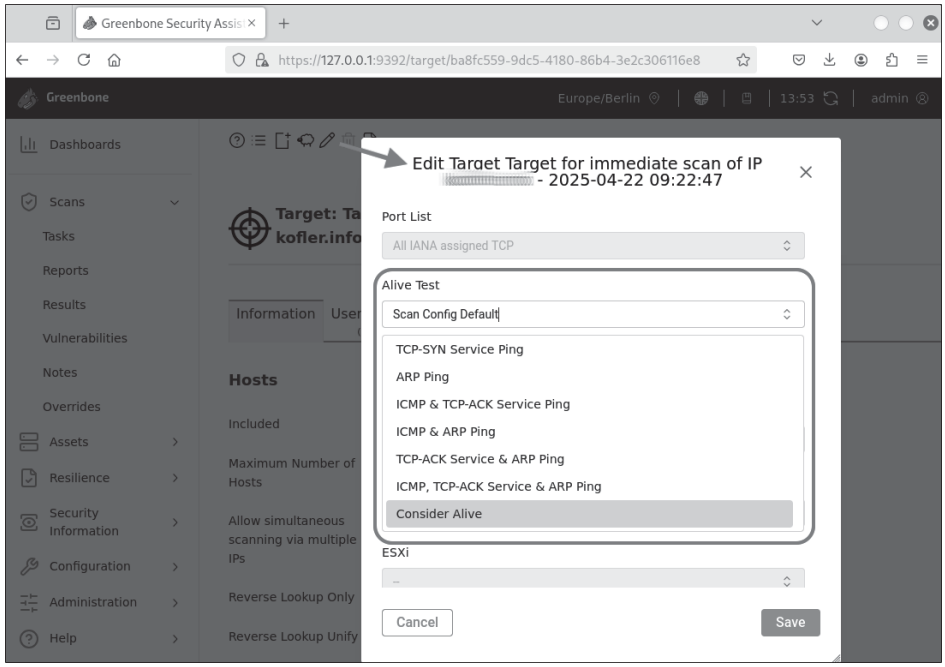

Abbildung 3.11 Der Alive-Test entscheidet darüber, wie OpenVAS feststellt, ob der Zielrechner läuft.

Tasks selbst einrichten

Um einen neuen Scan einzurichten, klicken Sie auf der Seite Scans • Tasks auf den Wizard-Button und haben dann die Wahl zwischen dem einfachen Task Wizard und

dem schon etwas komplexeren Advanced Task Wizard. Noch viel mehr Möglichkeiten offenbart der Dialog New Task (siehe Abbildung 3.12).

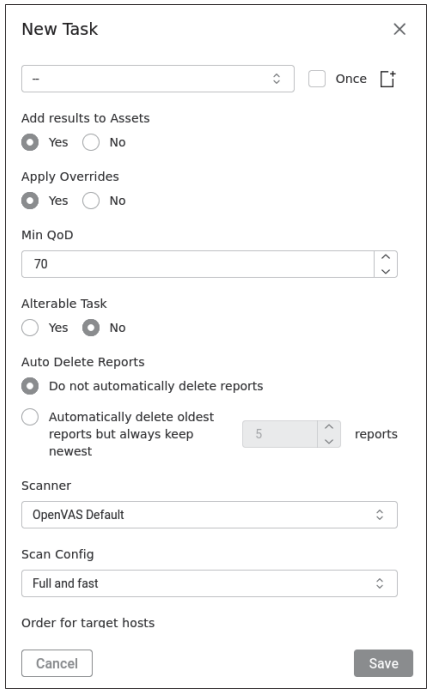

Abbildung 3.12 Wenn Sie Tasks manuell einrichten (nicht mit dem Assistenten), stehen Ihnen noch viel mehr Optionen zur Auswahl.

Bevor Sie den Dialog New Task richtig nutzen können, müssen Sie sich aber zuerst mit dem Configuration-Menü der OpenVAS-Oberfläche auseinandersetzen. Dort können Sie die Bausteine neuer Targets einrichten:

- Configuration • Targets listet alle bisher verwendeten Targets (IP-Bereiche oder Hostnamen auf) und bietet die Möglichkeit, neue Targets zu definieren. Dabei können Sie auch angeben, welche Ports berücksichtigt werden sollen (je mehr Ports, desto länger dauert der Scan) und wie der oben schon erwähnte Alive-Test durchgeführt werden soll.
- Configuration • Port Lists zeigt die vordefinierten Port-Listen an. All IANA assigned TCP umfasst beispielsweise ca. 5.800 TCP-Ports, aber keine UDP-Ports. Wenn keine der Listen Ihren Ansprüchen genügt, können Sie mit etwas Mühe auch eine eigene Liste definieren.
- Configuration • Scan Configs zeigt vordefinierte Scan-Sets an. Ein derartiges Set beschreibt, welche Vulnerability-Tests im Rahmen eines Scans durchgeführt werden. Die Default-Konfiguration von OpenVAS lautet Full and fast. Dabei sind

über 160.000 Tests vorgesehen! Tatsächlich durchgeführt werden aber nur die Tests, die auch sinnvoll sind. Findet OpenVAS auf dem Zielrechner keinen Webserver, dann erspart er sich natürlich alle diesbezüglichen bzw. darauf aufbauenden Tests. Auch hier besteht die Möglichkeit, eigene Testlisten einzurichten.

- In Configuration • Alerts können Sie festlegen, wie OpenVAS reagieren soll, wenn ein bestimmtes Ereignis eintritt (Scan abgeschlossen, Problem der Stufe *n* entdeckt etc.). Im einfachsten Fall sendet OpenVAS dann eine E-Mail. Das setzt aber voraus, dass auf dem OpenVAS-Host ein funktionierender Mailserver eingerichtet ist. Unter Kali Linux, das ja oft in einer virtuellen Maschine läuft, ist das nicht der Fall.
- Mit Configuration • Schedules können Sie einen Zeitplan einrichten, der festlegt, wann und wie häufig (z. B. alle sieben Tage) ein Task ausgeführt werden soll.

Hoher Ressourcenbedarf

OpenVAS startet bei einem Sicherheits-Scan eine Menge Hintergrundprozesse. Damit der Scan flüssig läuft, sollte der Rechner bzw. die virtuelle Maschine zumindest über 6 GByte RAM und 2 CPU-Cores verfügen. In den OpenVAS-Diskussionsforen gibt es Empfehlungen, für Scans großer Netzwerke wesentlich leistungsstärkere Hardware oder entsprechend große Cloud-Instanzen zu verwenden, also mit 32 GByte und mehr Speicher und mit möglichst vielen CPUs bzw. Cores. Generell kann OpenVAS CPU-Cores gut ausnutzen und führt dann diverse Tests parallel aus. Je mehr Cores Sie OpenVAS zur Verfügung stellen, desto kürzer ist die Laufzeit von Sicherheits-Scans.

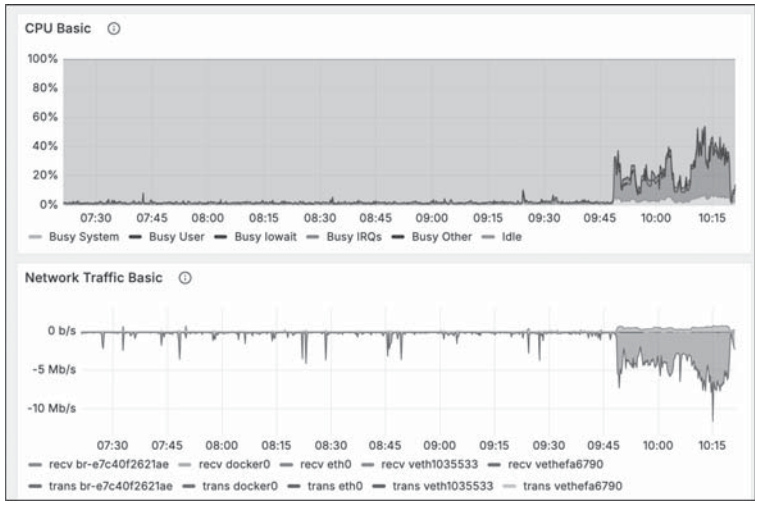

Abbildung 3.13 Der Zielrechner wird mit Prometheus überwacht. Der Start des OpenVAS-Scans ist sowohl bei der CPU-Auslastung als auch im Netzwerkverkehr unübersehbar.

Beachten Sie auch, dass OpenVAS erheblichen Netzwerkverkehr auslöst (siehe Abbildung 3.13). Sofern auf dem Zielrechner Monitoring-Software läuft, werden die vielen Sicherheitstests nicht unbemerkt bleiben. Dessen Administratoren werden vermutlich benachrichtigt, dass unüblicher Netzwerk-Traffic stattfindet, dass die WordPress-Installation angegriffen wird etc.

Alternativen

OpenVAS ist nach eigenen Angaben der weltweit beste Open-Source-Vulnerability-Scanner – aber er ist natürlich keineswegs der einzige. Es gibt eine ganze Reihe kommerzieller und zum Teil sehr teurer Alternativen. Bekannte Vertreter sind das schon erwähnte Programm *Nessus*, der zur Metasploit-Familie gehörende Scanner *Nexpose* oder das Programm *Core Impact*.

Für kommerzielle Vulnerability-Scanner spricht das in der Regel höhere Budget, das für die Weiterentwicklung zur Verfügung steht. Die Anbieter werben mit der besonders schnellen Reaktion auf neue Sicherheitsprobleme, mit diversen Zusatzfunktionen, besserer Bedienung etc. Es gibt im Internet einige Seiten mit Vergleichstests zwischen unterschiedlichen Vulnerability-Scannern, aber naturgemäß sind auch diese Tests mit Vorsicht zu genießen:

https://www.datamation.com/security/openvas-vs-nessus
https://pentest-tools.com/benchmarks/network-vulnerability-scanners

Naheliegend ist, dass Sie mit der parallelen Anwendung mehrerer Werkzeuge die umfassendsten Ergebnisse erhalten werden. Das erforderliche Budget werden aber nur die Sicherheitsabteilungen großer Firmen oder professionelle Pen-Tester aufbringen.

3.9 Metasploit Framework

Metasploit ist ein Open-Source-Projekt, dessen Module bei der Suche nach Sicherheitslücken sowie bei deren Ausnutzung helfen. Metasploit enthält eine riesige Sammlung von Exploit-Modulen. Mit weiteren Modulen kann Analyse- oder Schadcode (eine sogenannte *Payload*, wörtlich »Nutzlast«) auf dem angegriffenen Rechner installiert werden. Das bekannteste Payload-Modul ist das Programm *Meterpreter*.

Metasploit besteht aus zwei Bausteinen:

- **Metasploit Framework:** Die Basis von Metasploit, das sogenannte *Metasploit Framework*, ist eine riesige Tool- und Exploit-Sammlung. Alle Komponenten dieses Frameworks liegen in einem GitHub-Projekt als Open-Source-Code vor. Unter Kali Linux ist das Metasploit Framework standardmäßig installiert.