IT-Defense 2023 – Visionen und düstere Prognosen

IT-Defense 2023: Visionen und düstere Prognosen

Mikko Hyppönen von WithSecure legte in seiner Keynote mit dem Titel „Scorched Earth“ am zweiten Konferenztag bemerkenswerte Ansichten zur Zukunft der KI dar.

Von Jörg Riether

Auf der IT-Defense 2023 reflektierte zunächst der finnische Sicherheitsexperte und Autor Mikko Hyppönen über die Vergangenheit, in der 1997 IBMs Deep Blue den seinerzeit amtierenden Schachweltmeister Garry Kasparov schlug. In seinen Augen ist die KI-Vision seitdem und bis heute gleichermaßen großartig wie angsteinflößend. Die Geschwindigkeit entwickele sich rasant und allein in den letzten sechs Monaten sei hier mehr passiert als in den letzten 30 Jahren zusammen, so Hyppönen.

Er führte als Beispiele den Text-zu-Bild-Generator Stable Diffusion sowie die Textgeneratoren und Dialogsysteme ChatGPT und Bard an. Seine Prognose mutet unerhört an: Schon in naher Zukunft würden Computer bessere Kunst als Menschen erschaffen. Computer würden die besseren Poeten, die besseren Musiker, die besseren Programmierer und die besseren Maler sein, so Hyppönen. Das, was Computer dann produzieren, würde die Menschen tiefer und intensiver berühren als das, was ein Mensch jemals zustande bringen könnte. Er würde diese Vorstellung hassen. Es ändere aber nichts an der Realität und genau so werde es kommen, dies sei für ihn klar. Mehr noch:

Wenn die Entwicklung so weitergehe wie aktuell, könnten Computer in 100 Jahren menschliche Intelligenz stimulieren.

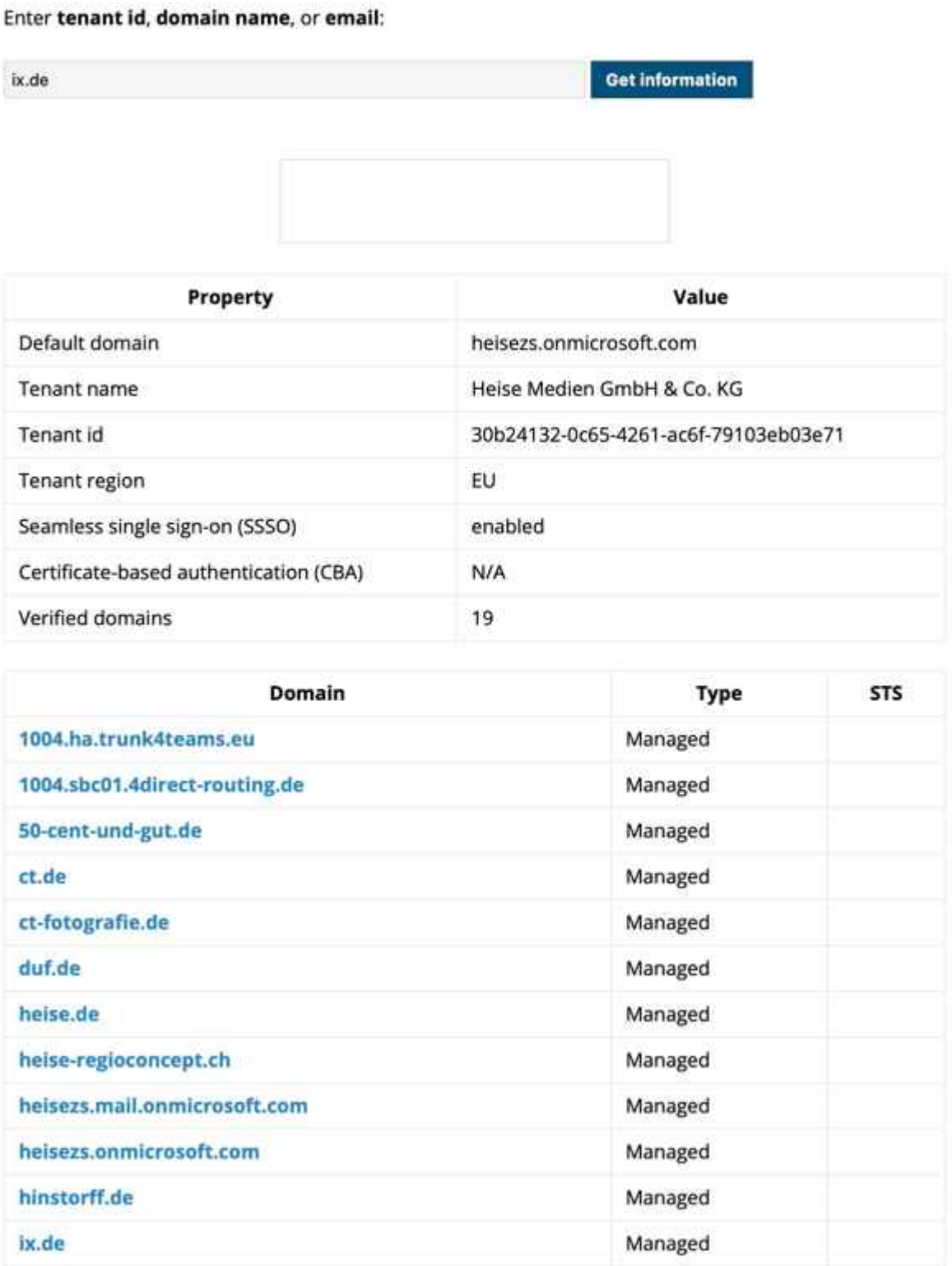
Hyppönens radikale Zukunftsvision: Computer werden in fast allem besser als Menschen sein (Abb. 1).

Gesprächige Tenants

Azure-AD- und Microsoft-365-Experte Nestori Syynimaa sprach

über das Vertrauen in M365-Umgebungen. In diesem Zusammenhang stellte er seine eigenen Tools vor. Diese beinhalten neben diversen PowerShell-Abfragewerkzeugen auch ein Web-GUI (siehe ix.de/zyfa), das über öffentlich zugängliche Quellen zahlreiche Tenant-Informationen einsammeln kann. Dass diese durchaus detailliert sein können, zeigt ein Versuch des Autors mit der Domain ix.de (Abbildung 2).

Es offenbart sich auf einen schnellen Klick, dass ix.de zur M365-Standarddomain heisezs.onmicrosoft.com gehört, die in der EU-Region beheimatet ist, außerdem gibt es im Tenant 19 verifizierte Domänen. Darunter gibt es auch Einträge wie „1004.ha.trunk4teams.eu“, „50-cent-und-gut.de“ sowie „heisezs.mail.onmicrosoft.com“. Die Seamless-Single-Sign-on-Technik (SSSO) ist aktiviert und zertifikatbasierte Authentifizierung (Certificate-based Authentication, CBA) ist nicht vorhanden, dies kann man mit einer gültigen Mailadresse optional prüfen.



Nestori Syynimaas Werkzeuge haben es in sich und können sowohl zur Informationssammlung in M365-Umgebungen, wie hier bei ix.de, als auch aktiv-offensiv benutzt werden (Abb. 2).
Dieses Beispiel ist noch relativ harmlos. Spannend ist, dass all diese Informationen „per Design“ öffentlich verfügbar sind. Es lohnt sich, mit dem Werkzeug selbst ein wenig mit der eigenen Unternehmensdomain oder anderen bekannten Domains zu spielen. Man könnte das Tool auch dazu missbrauchen, valide

Mailadressen herauszufinden. So gab das Web-GUI im Test bei einer vermutlich gültigen Microsoft-Mailadresse aktivierte CBA an, während es bei einer erfundenen Mailadresse aus vielen zufälligen Zeichen „nicht vorhanden“ ausgab. Es lassen sich also möglicherweise mehr Informationen ableiten, als dem einen oder anderen Unternehmen lieb sein könnte.

Auch die PowerShell-Werkzeuge sind mächtig. Neben der passiven Informationsbeschaffung gibt es hier sogar ein Phishingmodul, mit dem man live einen Angriff durchführen kann, der einen bei Erfolg direkt ins Outlook Web Access des Opfers führt. Syynimaa macht sich hier die Azure Device Code Authentication zunutze (siehe ix.de/zyfa).

Aus dem Tesla-Nähkästchen

Der IT-Sicherheitsforscher Martin Herfurt stellte in seinem Vortrag Details zum Projekt TEMPA vor, das Werkzeuge und Details zum proprietären VCSEC-Protokoll bereitstellt. Gewonnen hat er diese Informationen durch Analyse der dekompilierten offiziellen Tesla-Android-Anwendung – und konnte so Einblicke in die Angreifbarkeit des Protokolls und damit des Fahrzeugs erhalten (mehr Details siehe ix.de/zyfa).

Herfurt berichtete, dass er die Relay-Verwundbarkeiten, über die mit zwei Raspberry Pis und Telefonen ein Tesla entwendet werden konnte (siehe ix.de/zyfa), an den Hersteller gemeldet hatte. Tesla ließ verlauten, dass man dies nicht ändern werde und die Kunden doch bitte PIN2Drive einsetzen sollen, also eine zusätzliche PIN-Abfrage, bevor man das Fahrzeug starten kann. Dazu rät auch Herfurt, denn leider seien die Relay-Angriffe auf Teslas immer noch sehr einfach möglich. (ur@ix.de)

IT-Defense 2023: Visionen und düstere Prognosen

Mikko Hyppönen von WithSecure legte in seiner Keynote mit dem Titel „Scorched Earth“ am zweiten Konferenztag bemerkenswerte Ansichten zur Zukunft der KI dar.

Cybervorfälle-Top-Gefahren für Unternehmen

Im neuen Allianz-Risk-Barometer 2023 (ix.de/zh1v) belegen wie im vergangenen Jahr Cybervorfälle und Betriebsunterbrechungen die ersten beiden Plätze der **Top-Gefahren für Unternehmen.**

ix.de/zh1v

- **Vollständiges Programm der Kongressmesse SecIT**
- **Allianz-Risk-Barometer 2023**
- **Wissenschaftscomics von CASA**
- **BSI-Grundschutzkompendium**
- **WithSecure-Forschungsbericht „Creatively malicious prompt engineering“**
- **heise-Artikel zu ChatGPT-Versuchen von WithSecure**
- **Pressemitteilung Check Point zum Umgehen der Zugangsbeschränkungen von ChatGPT**
- **Pressemitteilung von Sophos zur Jagd nach Cyberkriminellen mit ChatGPT**

ChatGPT und Co.-KI als Gamechanger für Gut und Böse?

ChatGPT und Co.: KI als Gamechanger für Gut und Böse?

Seit Veröffentlichung des smarten Chatbots von OpenAI überschlagen sich die Meldungen, was man damit alles machen kann – aber auch die Warnungen, gerade in Sachen IT-Sicherheit.

Sicherheitsforscher von Check Point hatten kürzlich den ersten durch ChatGPT erstellten Angriffscode im Darkweb entdeckt (siehe auch iX 2/2023, Seite 20). Nun gibt es weitere Erkenntnisse: In Untergrundforen tauschten sich russische Kriminelle darüber aus, wie sie die von OpenAI für Russland

vorgesehenen Beschränkungen – Kontrolle von IP-Adressen, Zahlkarten und Telefonnummern – umgehen können. Die Forscher stießen auf Nachfragen, wie man mit gestohlenen Zahlkarten an einen leistungsfähigeren OpenAI-Account gelangt, außerdem auf zahlreiche halblegale russische Online-SMS-Dienste, die Anleitungen zum Registrieren bei ChatGPT geben.

Auch Sicherheitsexperten von WithSecure loteten die kriminelle KI-Kompetenz aus und ließen das Sprachverarbeitungsmodell GPT-3 in verschiedenen Bereichen wie Fake News, Social Media und Phishingmails agieren. Voraussetzung für die in der Regel gut lesbaren und glaubwürdigen Texte ist laut Ergebnisbericht (siehe ix.de/zh1v) ein präzises Briefing der KI. Wie bei echten Kriminellen wird eine maßgeschneiderte Spear-Phishing-Mail umso glaubwürdiger, je mehr Details über den Mitarbeiter, das Unternehmen und den Kontext bekannt sind.

Einen Gamechanger nicht nur für die dunkle Seite sieht das Sicherheitsunternehmen Sophos: Auch für die Jagd nach Cyberkriminellen entfalten die neuen KI-Tools Potenzial. So ließ sich GPT-3 per „Few-Shot Learning“ mit nur wenigen kommentierten Beispielen für Erkennung trainieren und wies nicht die Schwächen herkömmlicher maschineller Lernmodelle der Überanpassung und dadurch fehlender Verallgemeinerungen auf (Details siehe ix.de/zh1v). Einsetzen lässt sich das Werkzeug laut Sophos vor allem in der Spamerkennung und beim Reverse Engineering von Befehlszeilen, bei dem es eine Befehlszeile in eine verständliche Beschreibung übersetzen kann. In den Augen der Sophos-Forscher ist GPT-3 „ein Meilenstein für die Cybersicherheit“. (ur@ix.de)

Das BSI hat den Entwurf des Mindeststandards zur Protokollierung und Detektion von Cyberangriffen Version 1.0a.4 als Community Draft veröffentlicht

Das BSI hat den Entwurf des Mindeststandards zur **Protokollierung und Detektion von Cyberangriffen** Version 1.0a.4 als Community Draft veröffentlicht (siehe ix.de/zwq6).

ix.de/zwq6

- **Sicherheitslücke in MatrixSSL**
- **Informationen der Telekom-Forscher**
- **Report von Atlas VPN**
- **Blogartikel von Checkpoint, wie Kriminelle GPT missbrauchen können**
- **Blogartikel von Checkpoit zu ersten Fällen des Missbrauchs von ChatGPT**
- **BSI Community Draft**
- **LKA-Warnung vor gefälschten Domain-Rechnungen**
- **Google OSV Scanner**

---

gefälschten Domain-Rechnungen von D.D.N. Hosting.

Zum Jahreswechsel kommt es laut LKA Niedersachsen verstärkt zu **gefälschten Domain-Rechnungen** von D.D.N. Hosting. Ein Beispielschreiben findet sich auf der LKA-Seite.

(siehe ix.de/zwq6)

ix.de/zwq6

- **Sicherheitslücke in MatrixSSL**
- **Informationen der Telekom-Forscher**
- **Report von Atlas VPN**
- **Blogartikel von Checkpoint, wie Kriminelle GPT missbrauchen können**
- **Blogartikel von Checkpoit zu ersten Fällen des Missbrauchs von ChatGPT**
- **BSI Community Draft**
- **LKA-Warnung vor gefälschten Domain-Rechnungen**
- **Google OSV Scanner**

Hacking-Tools-Werkzeuge für Experten-2021

Gute Tools, böse Tools

Hacking-Werkzeug für Fortgeschrittene

Mit den Hacking-Tools von Penetrationtestern finden Sie Sicherheitslücken in Ihren Websites, Netzwerken und Anwendungen, bevor es andere tun.

Von Ronald Eikenberg und Alexander Königstein

Hollywood weiß Hacker-Aktivitäten in Szene zu setzen: Vor unzähligen Monitoren mit monochromatischen Benutzeroberflächen sitzen Gestalten im Kapuzenpulli und brechen durch die Firewalls. In der Realität geht es weitaus nüchterner zu, denn die eigentliche Action spielt sich hinter den Kulissen ab. Das ist aber nicht weniger faszinierend, denn Hacking-Tools leisten erstaunliche Dinge, wenn man sie richtig einsetzt. Das setzt etwas Wissen und Erfahrung voraus, doch beides baut sich ganz von selbst auf, wenn Sie erst mal Feuer gefangen haben. In diesem Artikel stellen wir eine Auswahl interessanter Profi-Werkzeuge vor, die sowohl auf der dunklen als auch auf der hellen Seite der Macht genutzt werden. Stöbern Sie auch im Artikel „Hack Dich selbst“ auf Seite 18, der nützliche Problemlöser für den Alltag präsentiert.

Mit den im Folgenden vorgestellten Profi-Tools spüren Sie Sicherheitslücken in Ihren Websites, Netzwerken, Apps, IoT-Geräten und vielem mehr auf. Anschließend können Sie gezielt Schutzmaßnahmen ergreifen und die Schlupflöcher stopfen, bevor es zu spät ist. Die meisten Hacking-Tools laufen am besten oder ausschließlich unter dem Betriebssystem Linux. Eine gute Grundlage für die ersten Schritte ist **Kali Linux**, das von Haus aus bestens auf die Bedürfnisse von Hackern zugeschnitten ist. Auf Seite 30 erfahren Sie, wie Sie sich einen Kali-USB-Stick

mit persistenter Datenpartition für Ihre Experimente erstellen. Download-Links und weiterführende Informationen zu allen vorgestellten Tools finden Sie online unter ct.de/ygg5. Aber genug der Vorrede – jetzt geht es in die Vollen!

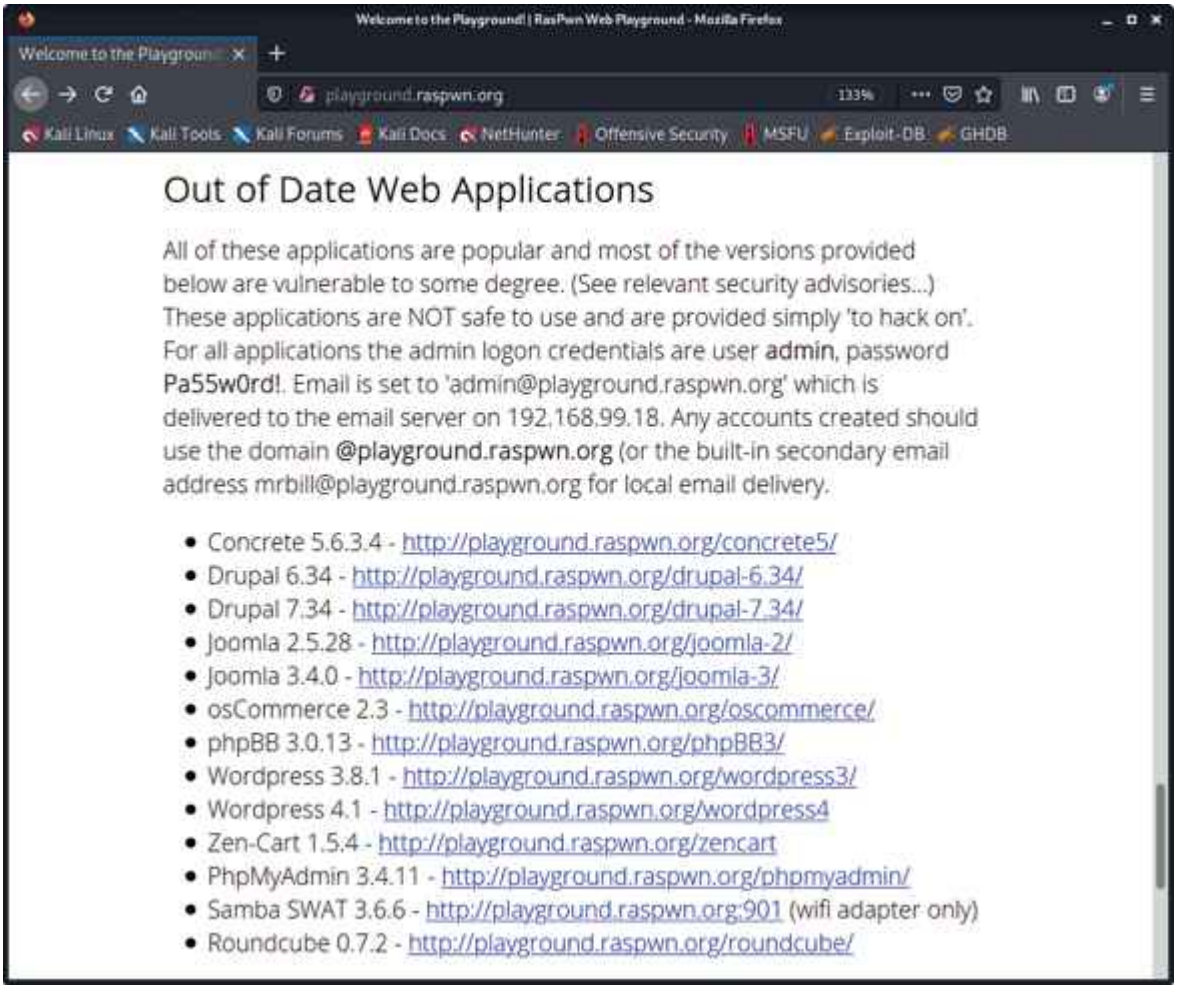
Angreifen erlaubt

Die hier genannten Hacking-Tools sind nicht illegal, aber natürlich dürfen Sie damit nicht gegen geltende Gesetze verstoßen (siehe Seite 170). Damit Sie gar nicht erst in Versuchung kommen, die Tools unerlaubt an fremden Servern zu testen, sollten Sie sich eine geeignete Übungsumgebung schaffen – zum Beispiel ein Testnetz, in dem sich ausschließlich Systeme befinden, die Sie attackieren möchten und dürfen.

Ein geeignetes Angriffsziel ist **RasPwn**, das ein ganzes Netzwerk voller verwundbarer Server simuliert, an denen Sie sich austoben können. Sie übertragen es einfach auf eine MicroSD-Karte, die Sie anschließend in einen Raspi-Kleincomputer stecken (mindestens Raspi 2B). Nach dem Booten meldet sich ein WLAN namens „RasPwn OS“, zu dem Sie mit dem Passwort „In53cur3!“ eine Verbindung herstellen. Aus dem Netz öffnen Sie http://playground.raspwn.org mit einem Browser Ihrer Wahl, wo Sie mit allen wichtigen Informationen über das virtuelle Netzwerk und die angreifbaren Server versorgt werden. Ein Netzwerkkabel darf nicht mit dem Raspi verbunden sein, andernfalls hat das hochgradig verwundbare Image unter Umständen Zugriff auf Ihr Hauptnetzwerk und das Internet – was Sie tunlichst vermeiden sollten.

Zu den möglichen Angriffszielen zählen verwundbare WordPress-Installationen, eine steinalte Version des Webshop-Systems osCommerce, das Datenbank-Tool phpMyAdmin, ein Mailserver, Samba und so weiter. Auch das Debian-Linux, auf dem RasPwn basiert, hat schon fast sieben Jahre auf dem Buckel und ist so löchrig wie ein Schweizer Käse. Obendrauf gibt es zahlreiche Web-Applikationen wie OWASP Bricks und Damn Vulnerable Web

Application (DVWA), die nur mit dem Ziel entwickelt wurden, möglichst verwundbar zu sein, um typische Sicherheitslücken am lebenden Objekt zu demonstrieren. Viele dieser Projekte sind online dokumentiert, wodurch sie sich hervorragend zum Lernen eignen (siehe ct.de/ygg5).



Das Raspi-Image RasPwn enthält etliche verwundbare Web-Apps – und das mit voller Absicht.

Netzwerk auskundschaften

Hat sich ein Angreifer Zugriff auf ein fremdes Netzwerk verschafft, etwa durch eine frei zugängliche Netzwerkbuchse im Aufenthaltsraum, eine per E-Mail eingeschleuste Malware oder ein schwaches WLAN-Passwort, dann wird er sich erst mal einen Überblick über die Geräte im Netz verschaffen, um mögliche Angriffsziele auszumachen. Hierbei ist der mächtige Netzwerkscanner **Nmap** (Network Mapper) die erste Wahl. Er spürt nicht nur die Rechner, Drucker, NAS, Server, Router und vieles

mehr auf, sondern auch die darauf laufenden Dienste. Durch Skripte lässt sich der Scanner beliebig erweitern, etwa um die entdeckten Clients gleich noch auf Sicherheitslücken abzuklopfen. Das alles ist nützlich, um verwundbare Geräte im eigenen Netz aufzuspüren und sie anschließend entweder abzusichern oder aus dem Verkehr zu ziehen.

Nmap läuft auf Linux, macOS und Windows, bei Kali Linux ist er inklusive. Wenn Sie auf einer Shell nmap ohne Parameter eintippen, zeigt das Tool die wichtigsten Betriebsmodi an. Um einfach und schnell die offenen Ports eines bestimmten Hosts herauszufinden, hängen Sie einfach dessen IP-Adresse an den Befehl an, etwa nmap 192.168.178.1. Das müssen Sie zwar nicht als root ausführen, es lohnt sich aber: So finden Sie mehr über die Clients heraus, im konkreten Fall die MAC-Adressen. IPv6-Adressen scannen Sie mit dem Parameter -6.

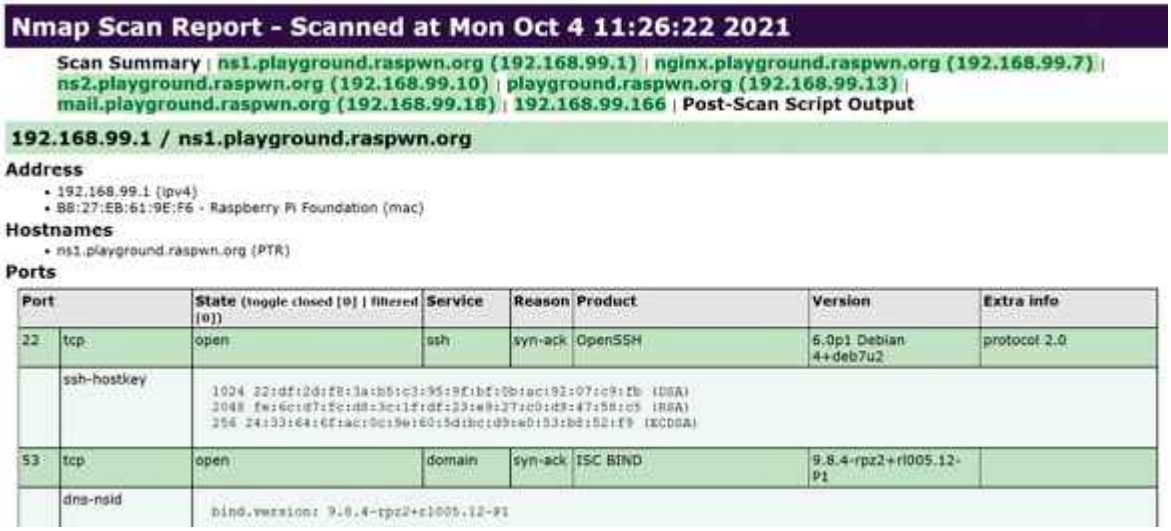
Sie können den Scan auf einen IP-Bereich ausweiten, den Sie zum Beispiel mit 192.168.178.1-50 definieren (alle IP-Adressen, die mit 192.168.178 anfangen und mit .1 bis .50 enden). Oder Sie scannen gleich das gesamte /24-Subnetz (alle bis .255): nmap 192.168.178.0/24. Ist der Scan abgeschlossen, präsentiert Ihnen Nmap die Ergebnisse auf der Shell, vorher lässt das Tool nicht von sich hören. Wer ungeduldig ist, kann mit --stats-every 10s festlegen, dass Nmap regelmäßig ein Statusupdate ausgibt.

Wirklich komfortabel lesbar ist der Bericht auf der Shell nicht. Sie können jedoch leicht einen formatierten HTML-Report erstellen, indem Sie zunächst Nmap mit -oX ergebnis.xml anweisen, einen XML-Export der Ergebnisse zu schreiben. Anschließend bauen Sie daraus mit dem unter Kali vorinstallierten Tool xsltproc eine HTML-Datei, die Sie mit jedem Browser öffnen können: xsltproc ergebnis.xml -o ergebnis.html

Mit dem einfachen Scan kratzen Sie erst an der Oberfläche der Möglichkeiten. Mehr können Sie Nmap über verschiedene Scan-

Optionen entlocken (siehe ct.de/ygg5). Sehr umfangreich ist der Modus -A, der unter anderem die Betriebssystem- und Versionserkennung (Fingerprinting) scharf schaltet. Diesen Modus sollten Sie mit sudo starten, damit Ihnen nichts entgeht. Aber aufgepasst: Nmap greift in diesem Fall aktiv auf die entdeckten Server zu, um Informationen einzuholen. Das kann zu unerwarteten Effekten führen, unser Epson-Drucker etwa spuckt bei jedem Scan eine spärlich bedruckte Seite aus. Sie sollten Ihre ersten Schritte daher besser im oben erwähnten Testnetz machen.

Eine Übersicht über die mitgelieferten Skripte finden Sie in der Dokumentation von Nmap (siehe ct.de/ygg5). Praktisch ist etwa das vulners-Skript, das zu den ermittelten Serverversionen bekannte Schwachstellen aus einer Online-Datenbank heraussucht. Eigene Skripte können Sie in der Programmiersprache Lua entwickeln.



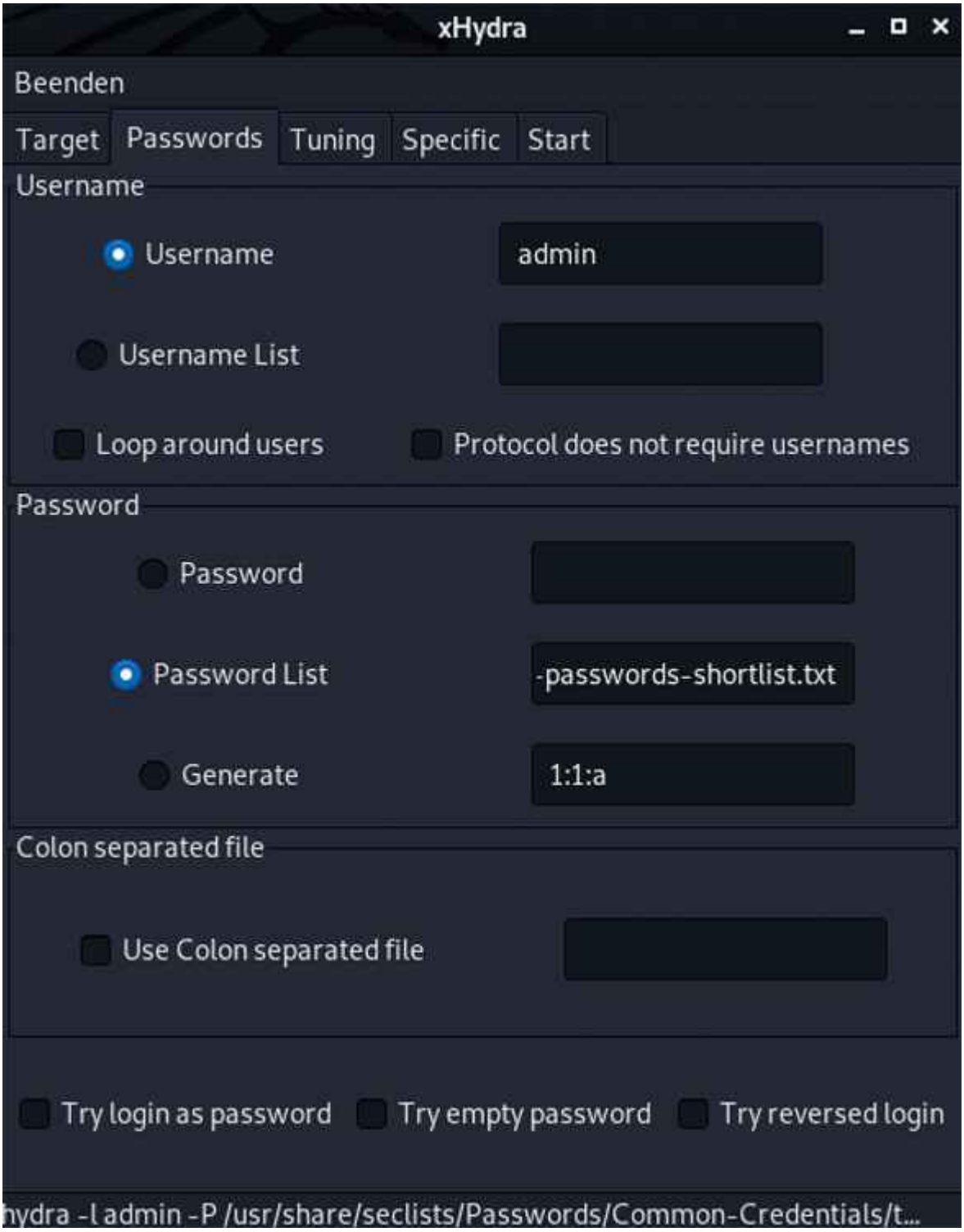
Was ist los im Netz? Der Netzwerkscanner Nmap liefert einen HTML-Bericht über alle Geräte und Dienste.

Zugriff auf Server

Vernetzte Geräte wie WLAN-Kameras oder Smart-Home-Komponenten sind oft für eine Überraschung gut: Auf manchen Exemplaren

laufen unerwartete Dienste, die im Worst Case sogar mit einem Standardpasswort für Gott und die Welt aus dem Internet erreichbar sind. Die entdecken Sie zum Beispiel mit einem Nmap-Scan (siehe „Netzwerk auskundschaften“). Doch dann stehen Sie erst mal vor verschlossener Tür, denn das Zugriffspasswort ist häufig ebenso wenig dokumentiert wie der Dienst selbst. Solche Dienste sind ein unkalkulierbares Sicherheitsrisiko.

Fehlt Ihnen das Passwort, können Sie versuchen, es zu erraten – oder Sie überlassen dem Login-Cracker **Hydra** die ganze Arbeit. Er unterstützt viele gängige Protokolle wie FTP, HTTP(S), SMB, SSH, Telnet und VNC, wodurch er universell einsetzbar ist. Sie können Hydra wahlweise auf der Shell benutzen oder mit xHydra eine grafische Oberfläche starten, um ein paar Parameter einzustellen und die Passwortsuche zu starten. Wichtig sind das Ziel, der Port und das richtige Protokoll im ersten Tab. Danach folgt die Konfiguration des Nutzernamens und einer Passwortliste. Falls Sie gerade keine zur Hand haben, können Sie unter Kali das Paket seclists installieren, das diverse Listen unter /usr/share/seclists/Passwords ablegt. Im letzten Tab ist der Output des Tools zu sehen, also im besten Fall das gesuchte Passwort.



Sesam, öffne Dich: Hydra probiert, sich mit beliebig langen Passwortlisten bei einem Server einzuloggen.

IPv4-Traffic umleiten

ARP-Spoofing (auch ARP-Poisoning genannt) ist ein alter, aber nach wie vor effektiver Trick, um IPv4-Netzwerkverkehr umzulenken. Ein Angreifer im gleichen Netzwerk kann so den Datenverkehr anderer Teilnehmer ohne deren Zutun mitlesen und

manipulieren, etwa um sensible Daten abzugreifen oder Schadcode zu verbreiten. Das Ziel des Angriffs sind die ARP-Tabellen der Netzwerkclients. Darin ist vermerkt, unter welchen MAC-Adressen die IPs im lokalen Netz erreichbar sind. Durch gefälschte Nachrichten im Address Resolution Protocol (ARP) kann ein Angreifer die Tabellen verändern und Traffic umleiten, mitlesen und manipulieren. Eine solche Umleitung ist aber auch praktisch, um den Netzwerkverkehr einzelner Clients zu untersuchen, zum Beispiel, um herauszufinden, mit welchen Servern ein Smart-Home-Gerät spricht und ob die übertragenen Daten verschlüsselt sind.

Mit dem Sniffing-Tool **Ettercap** ist ARP-Spoofing sehr einfach, weil es alle nötigen Schritte vereint. Kali-Nutzer starten es über den Launcher („Sniffing & Spoofing/ettercap-graphical“). Wählen Sie zunächst das gewünschte Netzwerk-Interface. Anschließend müssen Sie noch die beiden IPs einstellen, zwischen denen Sie lauschen möchten, zum Beispiel Router-IP und die IP des Clients, für den Sie sich interessieren. Klicken Sie hierzu auf den Menüknopf (drei Punkte), „Targets“ und „Current Targets“. Über die Add-Buttons tragen Sie die IPs als Target 1 und 2 ein. Alternativ können Sie auch erst mal im lokalen Netz nach Clients scannen. Klicken Sie dafür im Menü unter „Hosts“ auf „Scan for hosts“. Kurz darauf können Sie die Netzwerkteilnehmer unter „Hosts/Host list“ einsehen und per Rechtsklick als Target hinzufügen.

Verkehrsumleitung: Ettercap nutzt ARP-Spoofing, um den Datenverkehr anderer Rechner über sich umzuleiten.
Jetzt müssen Sie das ARP-Spoofing nur noch auslösen: Klicken Sie oben rechts auf den Knopf, der an eine Weltkugel erinnert („MITM menu“) und auf „Arp poisoning…“. Über das Menü und „View/Connections“ können Sie live beobachten, wie die Daten durch Ihr System fließen. Sie erfahren dort unter anderem IP-Adresse, Hostname und Land der Gegenstelle, den genutzten Port und den Datenumfang. Ein Doppelklick auf eine Verbindung zeigt die übertragenen Daten an. Interessant sind zum Beispiel unverschlüsselte HTTP-Verbindungen auf Port 80, weil Sie deren Inhalt ohne weitere Hilfsmittel als Klartext lesen können.

Ettercap bringt einige interessante Plug-ins mit, die Sie im Menü unter „Plugins/Manage plugins“ durchstöbern und per Doppelklick aktivieren können. Darunter findet sich auch ein Gegengift für ARP-Spoofing: Der „arp_cop“ soll ARP-Manipulationen anzeigen. Wenn Ihnen die Analysefunktionen von Ettercap nicht ausreichen, können Sie Werkzeuge wie Wireshark nutzen, denn der angezapfte Traffic ist auf dem anfangs eingestellten Netzwerk-Interface sichtbar. Mit den Linux-Werkzeugen iptables oder nftables können Sie den Datenverkehr

zudem beliebig umleiten, zum Beispiel an einen lokalen Server.

WLAN auf dem Prüfstand

Funknetzwerke müssen viel aushalten, denn jeder in Reichweite kann sie attackieren. Wenn Sie sich nicht darauf verlassen möchten, dass Ihr WLAN schon sicher genug sein wird, können Sie mit Hacking-Tools die Probe aufs Exempel machen. Kali hat mehrere davon an Bord, die unterschiedliche Angriffsszenarien durchspielen. Zur Nutzung benötigen Sie ein WLAN-Interface, das sich in den „Monitor Mode“ schalten lässt und zudem gut von Linux unterstützt wird. Solche gibt es als USB-WLAN-Adapter schon für weniger als 20 Euro, zum Beispiel von CSL Computer (Modell 27395) oder Alfa Network. Ob Ihr Interface den nötigen Modus unterstützt, erfahren Sie über eine Google-Suche nach dem Chipsatz, etwa „Ralink RT5572 monitor mode“.

Um die gängigsten Angriffsarten zu simulieren, können Sie zu **wifite2** greifen, das diverse WLAN-Hacking-Werkzeuge für Sie ansteuert, um Sicherheitsprobleme aufzuspüren. Sie starten es wie folgt:

```
sudo wifite --random-mac --kill
```

Die Option --random-mac sorgt dafür, dass die genutzte Geräteadresse des WLAN-Adapters zufällig ausgewürfelt wird und --kill beendet störende Prozesse, die dem Tool in die Quere kommen könnten. Wifite fragt Sie zunächst, welches WLAN-Interface genutzt werden soll und macht sich anschließend sofort an die Arbeit. Kurz darauf listet es alle Netze in Reichweite auf.

Mit wifite2 finden Sie heraus, wie sicher Ihr WLAN wirklich ist. Im ersten Schritt zeigt es alle Netze in Reichweite samt Verschlüsselung und WPS-Status an.
Sobald Sie Ihr WLAN gefunden haben, beenden Sie den Scan mit Strg+C und geben die Indexzahl des Netzes ein. Wifite testet anschließend die wichtigsten Angriffsmöglichkeiten der Reihe nach durch, allen voran WPS-Attacken (Pixie Dust und Brute Force auf die PIN), die bei anfälligen Routern am schnellsten zum Ziel führen. Danach nimmt sich das Tool WPA(2) zur Brust und schließlich das steinalte WEP-Verfahren. Gegen WPA3 kommt es derzeit nicht an.

Der WPA(2)-Angriff läuf relativ simpel ab: Zunächst zwingt wifite die Clients per Deauthentication-Paket, die Verbindung zum Router zu trennen. Bei der anschließenden Neuanmeldung zeichnet es den Handshake auf und setzt anschließend den Passwort-Cracker hashcat darauf an. Der probiert eine Reihe von Passwörtern aus einer langen Liste durch, bis er fündig wird. Die wichtigsten Schutzmaßnahmen in aller Kürze: Nutzen Sie lange WPA-Passwörter (mindestens 16 Zeichen, besser mehr), aktivieren Sie möglichst WPA2/3 (Mixed Mode) und die geschützte Anmeldung von WLAN-Geräten (Protected Management Frames, PMF).

Datenlecks im Webserver finden

Webserver sind prinzipbedingt meist für jeden erreichbar – und damit zwangsläufig auch für Angreifer, die nach Sicherheitslücken, Datenlecks und schwachen Passwörtern suchen. Das geschieht längst nicht mehr mühsam von Hand, sondern automatisiert. So können die bösen Buben tausende Websites innerhalb kurzer Zeit auf Schwachstellen abklopfen und müssen bei der Wahl ihres Angriffsziels nicht wählerisch sein.

Wenn Sie eine Website betreiben, müssen Sie also fest mit ungebetenem Besuch rechnen. Und wenn es eine Sicherheitslücke gibt, wird diese früher oder später auch ausgenutzt. Sie können den Angreifern jedoch die Petersilie verhageln, indem Sie sich deren Tools zu eigen machen, um etwaige Schwachstellen selbst frühzeitig zu finden. Auch diese Tools dürfen Sie nur gegen eigene Server und niemals unbefugt gegen fremde Systeme einsetzen, sonst drohen juristische Konsequenzen (siehe Seite 170). Beachten Sie, dass die Werkzeuge sehr viele Anfragen und damit potenziell auch eine hohe Last erzeugen, was die Erreichbarkeit des Servers beeinträchtigen kann.

Ein einfaches, aber effektives Werkzeug zur Suche nach Datenlecks ist **DIRB**. Es probiert eine lange Liste mit gängigen Verzeichnisnamen wie /admin, /backups oder /internal durch, um Ordner zu finden, die nicht für die Öffentlichkeit bestimmt, aber trotzdem für jeden zugänglich sind. Ferner kann das Hacking-Programm Verzeichnisnamen per Brute Force erraten. Gibt es einen Treffer, versucht DIRB auch noch mögliche Unterordner zu entdecken. Die Bedienung ist einfach:

```
dirb https://ihre-website.example
```

Unzureichend geschützte Verzeichnisse sind häufig die Ursache für Datenlecks, etwa wenn darin Backups der MySQL-Datenbank oder Konfigurationsdateien mit Zugangsdaten gespeichert sind.

Diese Blindgänger sollten Sie rechtzeitig entschärfen, zum Beispiel durch einen Zugriffsschutz auf dem Verzeichnis, sofern die Daten überhaupt auf dem öffentlichen Server liegen müssen.

WordPress-Lücken aufspüren

Das Content-Management-System WordPress ist sehr verbreitet (siehe S. 60 ff.) und bei Angreifern entsprechend hoch im Kurs. Häufig wird es in veralteten – und somit verwundbaren – Versionen betrieben oder mit anfälligen Plug-ins und Themes. Auch Konfigurationsfehler begünstigen eine Fremdübernahme. Solche Schlupflöcher aufzudecken ist inzwischen ein Kinderspiel – zum Beispiel mit dem WordPress Security Scanner **WPScan**. Der kann Ihnen gute Dienste beim Absichern Ihrer Website leisten.

Auf der GitHub-Seite des Ruby-Tools erfahren Sie, wie Sie es unter Linux, macOS und als Docker-Container an den Start bringen (siehe ct.de/ygg5). Kali-Nutzer können sich das sparen, das Programm ist vorinstalliert. Um Ihre WordPress-Installation zu scannen, füttern Sie WPScan einfach mit der URL: wpscan --url https://ihre-website.example/wordpress

Bevor die Analyse beginnt, lädt der Security Scanner eine Datenbank mit aktuellen Infos aus dem Netz. Das geschieht normalerweise automatisch, wenn Sie es jedoch auf die verwundbaren WordPress-Installationen von RasPwn loslassen möchten (siehe „Angreifen erlaubt“), haben Sie keine Internetverbindung, solange Sie mit dem Raspi-Testnetz verbunden sind. In diesem Fall sollten Sie sich zunächst mit Ihrem normalen Netz verbinden und das Update mit wpscan --update manuell starten. Trennen Sie die Verbindung danach, ehe Sie schließlich den Scan aus dem RasPwn-Netz anwerfen.

Nach und nach gibt WPScan interessante Informationen über die WordPress-Installation aus, darunter die WordPress-Version samt Erscheinungsdatum und eine Einschätzung, ob diese Ausgabe

nach aktuellem Stand der Dinge sicher ist. Weiterhin identifiziert das Tool die Versionen von Webserver und PHP sowie Themes, Plug-ins und diverse Konfigurationsfehler. Prinzipiell kann man selbst im Netz recherchieren, welche Sicherheitslücken in den identifizierten Versionen klaffen. Aber auch das kann Ihnen WPScan abnehmen. Diese Informationen fragt das Tool über ein Web-API vom Server der Entwickler ab – dafür ist eine kostenfreie Registrierung nötig (siehe ct.de/ygg5).

Datenbank-Lecks verhindern

Die Kronjuwelen einer Website sind häufig Kunden- oder gar Nutzerdaten. Diese können Onlinegauner im Darknet leicht zu Geld machen. In der Regel bewahren Webanwendungen solche Daten in einer Datenbank auf, die natürlich gut geschützt sein sollte. Die Betonung liegt auf sollte, denn allzu oft gelingt es Cyberkriminellen, Kundendaten im großen Stil aus Datenbanken abzugreifen.

Eine häufige Ursache sind sogenannte SQL-Injection-Lücken: Dabei spricht der Angreifer nicht direkt mit dem Datenbankserver, sondern versucht stattdessen, die Webanwendung dazu zu bringen, eingeschleuste SQL-Befehle auf der Datenbank auszuführen. Das Resultat ist häufig, dass die Datenbank über die Web-Anwendung massenweise sensible Datensätze ausspuckt.

Sie ahnen es vielleicht schon: Auch für solche Lücken gibt es ein Hacking-Tool, in diesem Fall **SQLmap**. Es unterstützt zahlreiche Datenbanken, unter anderem Oracle, MySQL, MariaDB, MS SQL Server, PostgreSQL und SQLite. Je nach Datenbanktyp und Berechtigungen kann es auch Dateien auf den Webserver schreiben. Hacker können so versuchen, eine Web-Shell hochzuladen, um den Server dauerhaft fernzusteuern.

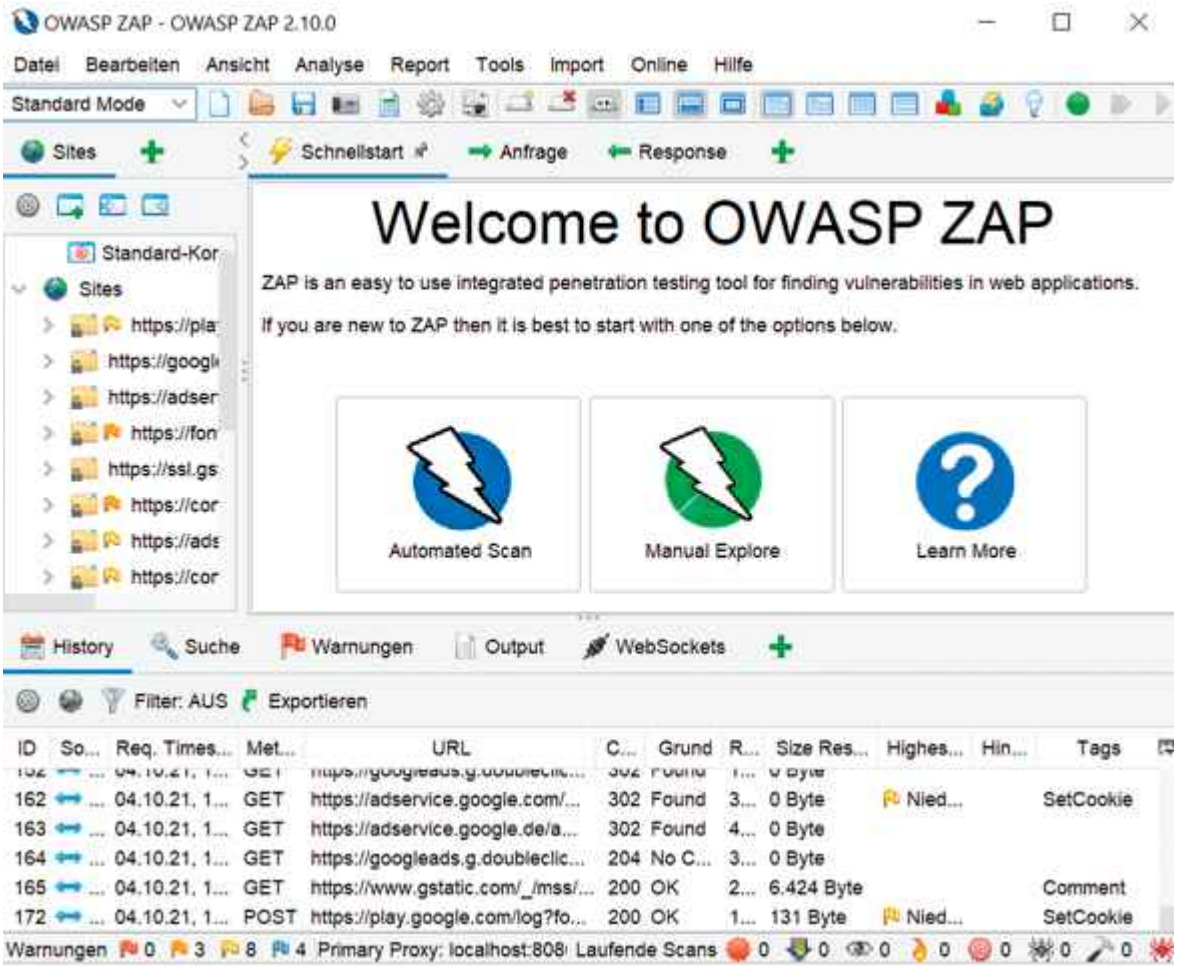
Für einen ersten Funktionstest können Sie die absichtlich anfällige „Wacko Picko“-Website von RasPwn mit SQLmap scannen.

Das Login-Formular der Website sendet beim Abschicken zwei POST-Parameter, nämlich „username“ und „password“, die der Schwachstellenscanner in diesem Beispiel in die Mangel nehmen soll. Um zu überprüfen, ob die Website bei der Auswertung dieser Parameter patzt, können Sie das Tool mit --data anweisen, genau das herauszufinden:

```
sqlmap -u "http://wackopicko.playground.raspwn.org/users/login.php" --data="username=1&password=1" --banner
```

Die Option „banner“ findet die Datenbankversion und das Betriebssystem des Servers heraus, wenn die Website verwundbar ist. Falls Sie den Datenbankinhalt gleich auslesen möchten, ersetzen Sie --banner einfach durch --dump.

Solche SQL-Injections vermeiden Sie, indem Sie Eingaben von außen konsequent überprüfen, bevor sie verarbeitet oder gar in Datenbankbefehle integriert werden. Weiterhin ist der Einsatz sogenannter „Prepared Statements“ sinnvoll, bei denen Sie zunächst den Aufbau des SQL-Befehls festlegen, ehe Sie darin einen Platzhalter mit den von außen angelieferten Werten füllen. Am besten basteln Sie die Datenbankbefehle nicht selbst zusammen, sondern setzen auf eine hinreichend getestete ORM-Bibliothek (Object-Relational Mapping), die bereits gegen alle Eventualitäten abgesichert ist.



Der Zed Attack Proxy (ZAP) macht verschlüsselten Datenverkehr im Klartext sichtbar und spürt Schwachstellen in Web-Anwendungen und APIs auf.

Browser- und App-Traffic

Der **OWAP Zed Attack Proxy (ZAP)** ist ein universelles Werkzeug zur Analyse und Manipulation von Web-Traffic (HTTP/HTTPS). Sie können sich damit zum Beispiel zwischen Browser und Internet klemmen oder den Datenverkehr Ihres Smartphones durch den Proxy schleusen, um herauszufinden, welche Daten wohin übertragen werden. Eine Stärke des ZAP ist, dass es die identifizierten Gegenstellen, also Webanwendungen, API-Endpunkte und so weiter gleich noch auf Sicherheitsprobleme abklopfen kann. ZAP ist eine Java-Anwendung und läuft unter Windows, Linux und macOS.

Nach dem ersten Start klicken Sie am besten auf den Browser-Knopf in der Symbolleiste, um einen perfekt vorkonfigurierten

Webbrowser zu starten. Dessen Datenverkehr wird automatisch durch den Proxy geschleust. Öffnen Sie damit eine Website, um den Traffic in ZAP zu inspizieren. Wenn Sie den Browser auf diese Weise starten, schleust ZAP in die geöffneten Websites eine eigene Oberfläche namens ZAP HUD ein, über die Sie zahlreiche Funktionen direkt aus dem Browser steuern können. Klicken Sie auf den Knopf „Take the HUD Tutorial“, um eine Einführung zu erhalten und einige der nützlichen Funktionen kennenzulernen.

Gemischtwaren

Dieser Artikel liefert Ihnen nur eine kleine Auswahl an Hacking-Tools. Das Angebot ist riesig und täglich kommen neue dazu. Einige davon sind sehr komplex oder nur für bestimmte Zielgruppen interessant. Dazu zählt das modular aufgebaute Pentesting-Framework **Metasploit**, das professionelle Penetrationtester nutzen, um einen kompletten Angriff zu simulieren: vom Aufspüren der Ziele über das Ausnutzen von Sicherheitslücken bis hin zum Ausleiten der Datenbeute. Falls Sie sich eingehender mit Hacking beschäftigen möchten, sollten Sie einen Blick darauf werfen. In eine ähnliche Kerbe schlägt **PowerShell Empire**, das Pentestern weitreichenden Rechnerzugriff verschafft, ohne verdächtigen Binärcode auf dem System zu hinterlassen – die Angriffsmodule bestehen aus Skripten für die Windows PowerShell.

Wer eine Windows-Domäne administriert, sollte Tools wie **mimikatz** kennen, das Anmeldeinformationen aus dem Arbeitsspeicher der Windows-Clients ausliest. Angreifer gelangen damit schlimmstenfalls an die Zugangsdaten eines Domänen-Administrators und können das gesamte Netzwerk übernehmen. Den Domänencontroller spüren die Eindringlinge vorher mit **AdFind** von joeware auf. Auch **PsExec** aus Microsofts SysInternals-Kollektion birgt ein gewisses Missbrauchspotenzial: Es wird genutzt, um Befehle auf anderen Rechnern im Netzwerk auszuführen. Angreifer nutzen es mit

zuvor erbeutete Anmeldeinformationen.

Das von der NSA entwickelte Reverse-Engineering-Toolkit **Ghidra** ist interessant, wenn Sie ausführbaren Code (wie EXE- und DLL-Dateien) bis ins letzte Bit auseinandernehmen und verstehen möchten. Es dekompiliert Binärdateien und kann sie auch wieder zusammenbauen, ähnlich wie der kommerzielle Disassembler IDA Pro. In c't 14/2020 haben wir Ghidra ausführlicher getestet (siehe ct.de/ygg5).

Fazit

Die vorgestellten Hacking-Tools decken einen weiten Bereich ab. Manche Techniken sind erschreckend simpel, andere fordern viel Einarbeitung und Erfahrung. Sich damit zu beschäftigen lohnt sich aber: Sie lernen so, wie ein Angreifer zu denken und Ihre eigenen Sicherheitsprobleme und -lücken aufzuspüren. Das ist hilfreich – ganz gleich, ob Sie nur eine private WordPress-Site betreiben oder gar für die Sicherheit Ihrer Kunden verantwortlich sind. *(rei@ct.de)*

Hacking-Tools & weitere Infos: ct.de/ygg5

Sicherheitsforscher Sönke Huster über Lücken im WLAN-Stack des Linux-Kernels

„Es reicht, wenn du dein WLAN anhast“

Über die Luft gehackt werden, nur weil das WLAN eingeschaltet ist? Im August hat Sönke Huster Sicherheitslücken im WLAN-Stack des Linux-Kernels gefunden, die einen solchen Angriff theoretisch ermöglicht hätten. Seine Entdeckung zeigt, wie wichtig es ist, Software ausführlich zu testen.
Über die Luft gehackt werden, nur weil das WLAN eingeschaltet ist? Im August hat Sönke Huster Sicherheitslücken im WLAN-Stack des Linux-Kernels gefunden, die einen solchen Angriff theoretisch ermöglicht hätten. Seine Entdeckung zeigt, wie wichtig es ist, Software ausführlich zu testen.

Von Kathrin Stoll

Sönke Huster ist wissenschaftlicher Mitarbeiter am Secure Mobile Networking Lab (SEEMOO) der TU Darmstadt. Im August 2022 hat er fünf Sicherheitslücken im WLAN-Stack des Linux-

Kernels entdeckt. Mittlerweile gibt es Patches. Wir haben mit ihm über den Fund, seine Methodik und den Disclosure-Prozess gesprochen.

Der Sicherheitsforscher Sönke Huster hat fünf Sicherheitslücken im Linux-Kernel gefunden. Wie er das gemacht hat, verrät er im Gespräch mit c’t. *Josephine Franz*

c’t: Wie kommt man darauf, im Linux-Kernel nach Sicherheitslücken zu suchen?

Sönke Huster: Ich habe dieses Jahr meine Masterthesis über Bluetooth-Fuzzing unter Linux geschrieben. Die Idee kam von meiner Masterarbeitsbetreuerin Dr. Jiska Classen. Im Bluetooth-Stack habe ich dann auch ein paar kleine Sicherheitslücken gefunden. Dann wurde ich wissenschaftlicher Mitarbeiter am Secure Mobile Networking Lab von Prof. Matthias

Hollick und es lag nahe, es auf WLAN auszuweiten. Aus Angreifersicht sind WLAN und Bluetooth super interessant und auch irgendwie ähnlich. Wenn ich dich hacken will, ist es ja viel cooler, ich kann das durch die Luft aus dem Raum nebenan machen, ohne dass ich dafür erst physisch auf deinen Rechner zugreifen können muss, um zum Beispiel einen USB-Stick einzustecken. Beide Protokolle sind dafür prädestiniert.

c't: Du hast gleich fünf Lücken im Linux-Kernel gefunden. Wie bist du dabei vorgegangen?

Huster: Die Methode, die ich verwende, heißt Fuzzing. Sie wurde in den Achtzigerjahren von Barton Miller [Professor der Informatik in Madison, Wisconsin, Anm. d. Red.] entdeckt, der sich über eine Telefonleitung auf Holzmasten remote auf seinem Arbeitsrechner einloggte. Bei Gewitter wurde die Übertragung des Signals gestört und seine Eingaben kamen verzerrt an. Das führte dazu, dass Programme abstürzten oder sich anders verhielten als erwartet. So kam man dahinter, dass man zufällige Eingaben nutzen kann, um Bugs und Sicherheitslücken zu finden und das Fuzzing – auch Fuzz-Testing – war erfunden. Heute verwendet man dazu sogenannte Fuzzer. Das sind im Grunde Programme, die die Eingabeschnittstellen von Programmen, Betriebssystemen oder Netzwerken mit zufälligen Daten fluten.

Mit komplett zufälligen Eingaben arbeitet man heute aber nicht mehr. Man kann das Verfahren verfeinern und Eingaben benutzen, die nah an denen sind, die das Target – in diesem Fall eben Linux in meiner VM – erwartet. Um WLAN zu untersuchen, lasse ich den Fuzzer WLAN-Pakete mit kleinen Anomalien an das Linux-System in meiner virtuellen Maschine schicken, die er fortlaufend verändert. Dabei beobachtet und dokumentiert der Fuzzer, welcher Code im Kernel zur Verarbeitung der mutierten WLAN-Pakete getriggert wird. Man könnte auch sagen: welchen Weg ein Paket bei der Verarbeitung nimmt. Immer, wenn bei der Verarbeitung eines Pakets Code abgedeckt wurde, der vorher noch nicht ausgeführt wurde, nimmt der Fuzzer dieses Paket in sein Eingabeset auf und nutzt es als Ausgangspunkt für neue

Mutationen. Diese veränderten Pakete schickt er dann wieder an den Kernel. Das Ganze passiert ein paar Tausend Mal pro Sekunde. Das Ziel ist es, möglichst viel Code „zu covern“, also durch die mutierten Eingaben Teile des Kernel-Codes abzudecken, die der Fuzzer noch nicht kennt. Coverage-Guided Mutational Fuzzing lautet der Fachbegriff für diese Art von Fuzz-Testing.

c’t: Wenn das Target abstürzt, hat man einen Treffer gelandet?

Huster: Genau. Ein Absturz oder anderes unerwartetes Verhalten, zum Beispiel, wenn es sich aufhängt, sind eigentlich immer ein Hinweis auf einen Bug oder eine Schwachstelle. Die Eingaben, die so etwas bewirken, speichert der Fuzzer separat ab, sodass ich den Crash reproduzieren kann. Bei einer der fünf Lücken, die ich gefunden habe, war es zum Beispiel so, dass ein kaputtes Paket – oder eine Reihe von Paketen – eine sogenannte Linked List korrumpierte und quasi das letzte Paket in der Liste wieder auf das erste gezeigt hat. Bei der Verarbeitung wusste das Betriebssystem nie, wann die Liste zu Ende ist und hat sich schließlich aufgehängt, weil es aus dieser Schleife nicht rauskam.

c’t: Das klingt nach einem ärgerlichen Bug, aber nicht nach einem, den ein Angreifer für eine Remote Code Execution nutzen könnte.

Huster: Nein. Es wäre schwierig, eine Möglichkeit zu finden, das auszunutzen. Die Endlosschleife führt dazu, dass das Betriebssystem sich aufhängt und das wars. Aber eine andere der Lücken ermöglicht es einem Angreifer, den Speicher zu überschreiben, sodass er theoretisch Code aus der Ferne ausführen könnte. Der Kernel reserviert Speicher für die Ausführung von Programmen und Prozessen. Wenn jetzt beispielsweise 128 Byte an einer Stelle im Speicher für einen bestimmten Vorgang vorgesehen sind, dann darf man da eigentlich auch nicht mehr als diese 128 Byte reinschreiben. Bestimmte Eingaben des Fuzzers haben Fehler in der

Paketverarbeitung aufgedeckt, die dazu führen, dass man mehr als die vorgesehene Länge in einen für einen Vorgang reservierten Teil des Speichers schreiben kann – ein sogenannter Buffer Overflow.

c't: Das wäre bereits ausreichend, damit ein Angreifer einen Rechner aus der Ferne übernehmen könnte?

Huster: Theoretisch. Es war möglich, als Angreifer 256 Byte kontrolliert in den Speicherbereich zu schreiben, der auf den zugewiesenen folgte. Für eine RCE müsste man zusätzlich herausfinden, wo im Speicher die kaputten WLAN-Pakete, die diesen Fehler im Kernel-Code triggern, überhaupt verarbeitet werden. Das ist aber gar nicht so einfach, weil es Mechanismen gibt, die dafür sorgen, dass der Kernel immer an unterschiedlichen Stellen im Speicher ausgeführt wird. Kernel Address Space Layout Randomization nennt sich das. Aber es wäre denkbar, dass sich noch weitere Sicherheitslücken finden, die einem das verraten.

c't: Ist das eine Hypothese oder hast du das auch erfolgreich prüfen können?

Huster: Nein. Das übersteigt meine Fähigkeiten. Es ist schon eher eine Hypothese. Aber eine, die sehr wahrscheinlich zutrifft. Es gibt verschiedene Arten von Sicherheitslücken und eine Lücke von diesem Typ bietet sich – in diesem konkreten Fall eben in Kombination mit weiteren – theoretisch dafür an.

Aus Angreifersicht das Spannende an den Sicherheitslücken ist, dass man überhaupt keine Nutzerinteraktion braucht. Du musst dich nicht aus Versehen mit einem Hotspot verbinden, den der Hacker kontrolliert, damit er dir böse WLAN-Pakete schicken kann. Es reicht, wenn du dein WLAN anhast und dein Gerät nach Netzwerken in der Umgebung sucht. Im Hintergrund passiert das relativ häufig zur Standortbestimmung. Es ist nicht wie bei einem Phishing-Versuch, bei dem der Angreifer das Opfer erst dazu bringen muss, auf einen Button zu klicken und Login-Daten

einzugeben. Genau das macht solche Lücken potenziell so kritisch. Linux-Nutzer gibt es nicht so viele, aber drei der Lücken betreffen Android, und Android-Nutzer gibt es eine ganze Menge. Am Smartphone haben die meisten Nutzer ihr WLAN in der Regel an.

c’t: Ist der Fuzzer eine Eigenentwicklung des Secure Mobile Networking Labs?

Huster: Ja. Wir nutzen Komponenten aus LibAFL. Das ist eine Bibliothek, die ein sehr gutes Grundgerüst mitbringt, aber die Architektur unseres Fuzzers unterscheidet sich stark von der bestehender Fuzzer.

c’t: Kannst du sicher sein, dass es außer den fünf Lücken nicht noch weitere gibt?

Huster: Ich denke, man kann auf jeden Fall sagen, dass WLAN unter Linux durch unsere Arbeit ein bisschen sicherer geworden ist. Wir waren an Stellen im Kernel, wo meines Wissens nach noch nicht so viel gefuzzt wurde. Momentan gucken wir uns noch weitere Teile an und bisher haben wir nichts weiter gefunden. Aber hundertprozentige Sicherheit, dass es nicht noch mehr Bugs und Sicherheitslücken gibt, wird man nie haben. Es kann immer unvorhergesehene Eingaben geben, die einen Bug oder eine Sicherheitslücke offenlegen. Ein Angreifer kann sie genauso gut finden wie wir. Genau deshalb ist Fuzz-Testing so wichtig.

c’t: Seit Oktober gibt es Patches. Wie und an wen hast du die Sicherheitslücken gemeldet?

Huster: Es gibt gefühlt 1000 Anlaufstellen für Linux-Sicherheitssachen, zum Beispiel eine Mailing-Liste aller Hersteller irgendwelcher Linux-Distributionen. Dort hätte ich das melden können. Parallel hätte ich dann noch die Kernel-Leute informieren müssen. Ich hab mich entschieden, den Prozess an einen Hersteller abzugeben und habe mich an SUSE gewandt. Die SUSE-Leute haben Johannes Berg von Intel ins Boot geholt. Er ist der Maintainer des WLAN-Stacks unter Linux. Für

mich war es superspannend, mit ihm in so einem engen Austausch zu stehen, während er die Patches für die beiden Sicherheitslücken, die ich initial an SUSE gemeldet hatte, geschrieben hat.

Er hat mir die Patches dann geschickt und ich habe meinen Fuzzer darauf angesetzt. So sind wir auf die drei weiteren Sicherheitslücken – und insgesamt noch ein paar weitere kleinere Bugs – gestoßen. Das Ganze hat ein paar Wochen gedauert. Als alle Patches fertig waren, hat SUSE alle anderen Hersteller im Geheimen informiert und man hat einen Zeitpunkt festgelegt, zu dem man die Öffentlichkeit über die Lücken informiert. Die Hersteller hatten bis dahin über eine Woche Zeit, entsprechende Updates rauszubringen. Überrascht hat mich, dass manche Hersteller ihre Updates erst mehrere Tage nach der Bekanntgabe der Lücken verteilt haben.

c’t: C gilt als relativ unsichere Programmiersprache. Künftig soll es möglich sein, Kernel-Komponenten stattdessen in Rust zu schreiben. Hätte das deine Sicherheitslücken verhindert?

Huster: Sehr wahrscheinlich wären diese Lücken nicht aufgetreten, hätte man die Module in Rust geschrieben. Gerade die Geschichte, dass man Speicher überschreiben kann. Der Rust-Compiler hätte verhindert, dass die Kernel-Entwickler diesen Fehler überhaupt einbauen. Aber es gibt natürlich auch Fehler, die durch keine Programmiersprache der Welt verhindert werden.

c’t: Gibt es etwas, was du Admins und Anwendern raten würdest?

Huster: Sicherheitsupdates immer schnell einzuspielen. Wie gesagt, bis alle größeren Distributionen die Updates verteilt haben, hat es nach Veröffentlichung noch ein paar Tage gedauert. Gerade bei Android dauert es oft länger. Es kann einfach sein, dass die betreffende Sicherheitslücke schon eine Weile öffentlich ist, bis man als Nutzer ein Sicherheitsupdate bekommt. Deshalb sollte man Updates möglichst sofort

installieren. Auch wenn es nervt. Aber dann holt man sich in der Zwischenzeit halt mal einen Kaffee. (*kst@ct.de*)

Weitere Infos: ct.de/yvwk

Fake-Shops erkennen und Schäden vermeiden

Niemals ausgeliefert

Beim digitalen Einkauf betrügerischen Läden auf den Leim zu gehen, ist ärgerlich und teuer. Mit ein paar Grundregeln und hilfreichen Tools vermeidet man das. Wir stellen sie vor und

erklären, was im Schadensfall noch möglich ist.
Beim digitalen Einkauf betrügerischen Läden auf den Leim zu gehen, ist ärgerlich und teuer. Mit ein paar Grundregeln und hilfreichen Tools vermeidet man das. Wir stellen sie vor und erklären, was im Schadensfall noch möglich ist.

Von Nick Akinci

Über vier Millionen Deutsche sind schon einmal auf einen Fake-Shop hereingefallen. Das schätzt das von der Bundesregierung geförderte Marktbeobachtungsinstitut „Marktwächter digitale Welt“. Besonders häufig bieten solche Shops nach Angaben des Instituts Sportartikel, Elektronik sowie Haushaltsartikel, Bekleidung und Fahrräder, aber auch Brillen und Schmuck.

Wir zeigen, wie Sie Ihnen unbekannte Shops anhand verlässlicher Kriterien und mit hilfreichen Tools auf Seriosität prüfen, wie Sie Zahlungen absichern und was Sie tun können, falls Sie doch auf einen Fake-Shop hereingefallen sind.

Was ist ein Fake-Shop?

Fake-Shops sind Online-Shops, mit denen Kriminelle gutgläubigen Kunden ihr Geld abnehmen wollen, ohne ihnen die versprochene Ware zu liefern. In der einfachsten Variante erhalten Kunden, die darauf hereinfallen, überhaupt keine Ware. Etwas perfidere Betrüger versenden leere Kartons. Im Nachhinein behaupten sie, dass die Ware auf dem Versandweg abhandengekommen sein müsse. Mitunter verschicken sie auch Ware, die in keiner Weise der Produktbeschreibung entspricht.

Viele Fake-Shops sind nur für einen relativ kurzen Zeitraum online, da sie fast immer auffliegen und der Hoster sie im besten Fall vom Netz nimmt. In diesem Zeitfenster versuchen die Betrüger, möglichst viel Geld zu ergaunern. Sitzt der Hoster im Ausland, können sich solche Shops auch über Jahre halten.

Prüfender Blick

Fake-Shops sind häufig nicht auf den ersten Blick als solche zu erkennen. In Zeiten von Baukastensystemen wie Shopify & Co. klicken Betrüger professionell aussehende Online-Shops in wenigen Stunden zusammen. Es gibt jedoch eine Reihe von Indizien, die für einen Fake-Shop sprechen.

Um Kunden anzulocken, bieten die Täter die Ware in Fake-Shops oft deutlich günstiger an als in anderen Online-Shops. Insbesondere beliebte und häufig gehandelte Markenware preisen sie unter dem Marktwert an, gern als Sonderangebot getarnt. Schnäppchenjäger können sich auf Preisvergleichsseiten einen Eindruck verschaffen, ob die Preisgestaltung realistisch ist.

Als Nächstes schaut man in das Impressum. Fake-Shops haben oft keines, obwohl dies in Deutschland gesetzliche Pflicht ist – die Betrüger wollen ihre Identität verschleiern. Aber Achtung: Manche Fake-Shops enthalten ein echt aussehendes Impressum, welches jedoch schlicht falsche, unvollständige oder von anderen Websites kopierte Angaben enthält. Ob die Firma an der angegebenen Adresse sitzt, kontrolliert man am besten mit Google Maps. Den Unternehmensnamen und die zugehörige Handelsregisternummer prüft man auf handelsregister.de [1].

Abgesehen vom Impressum fehlen in vielen Fake-Shops auch Telefonnummern oder E-Mail-Adressen, um Kontakt aufzunehmen. Ebenfalls kein gutes Zeichen ist es, wenn sich Kontaktmöglichkeiten beschränken auf ausschließlich Handy- oder kostenpflichtige Nummern, Postfachadressen oder lediglich ein Kontaktformular. Misstrauen ist geboten, wenn AGB und Datenschutzerklärung sowie Widerrufsbelehrungen und Versandbedingungen fehlen.

Gütesiegel sind ein Hinweis auf vertrauenswürdige Shops, doch in Fake-Shops trifft man immer wieder einfach hineinkopierte oder frei erfundene Varianten an. Letztere ähneln teils bekannten Gütesiegeln – wie etwa dem von Trusted Shops.

Verfügt der Online-Shop über ein Gütesiegel, kann man auf der Homepage der Organisation prüfen, ob es sich um ein tatsächlich anerkanntes Gütesiegel handelt und ob der Online-Shop es rechtmäßig erworben hat. Durch einen Klick auf das Siegelsymbol muss man auf die Seite der dahinterstehenden Organisation gelangen. Verbreitet und vertrauenswürdig ist außer Trusted Shops auch das EHI Retail Institute („Geprüfter Online-Shop“). Als zuverlässig gilt außerdem das in Kopenhagen ansässige Bewertungsportal Trustpilot (alle unter ct.de/yu3d).

Kein Impressum, kein Handelsregistereintrag, keine Umsatzsteuer-ID, Shop ganz neu, Google Maps kennt den Shop an der angegebenen Adresse nicht und als Kontaktmöglichkeit nur eine Mobiltelefonnummer: Hier heißt es Finger weg!

Zahlmethoden

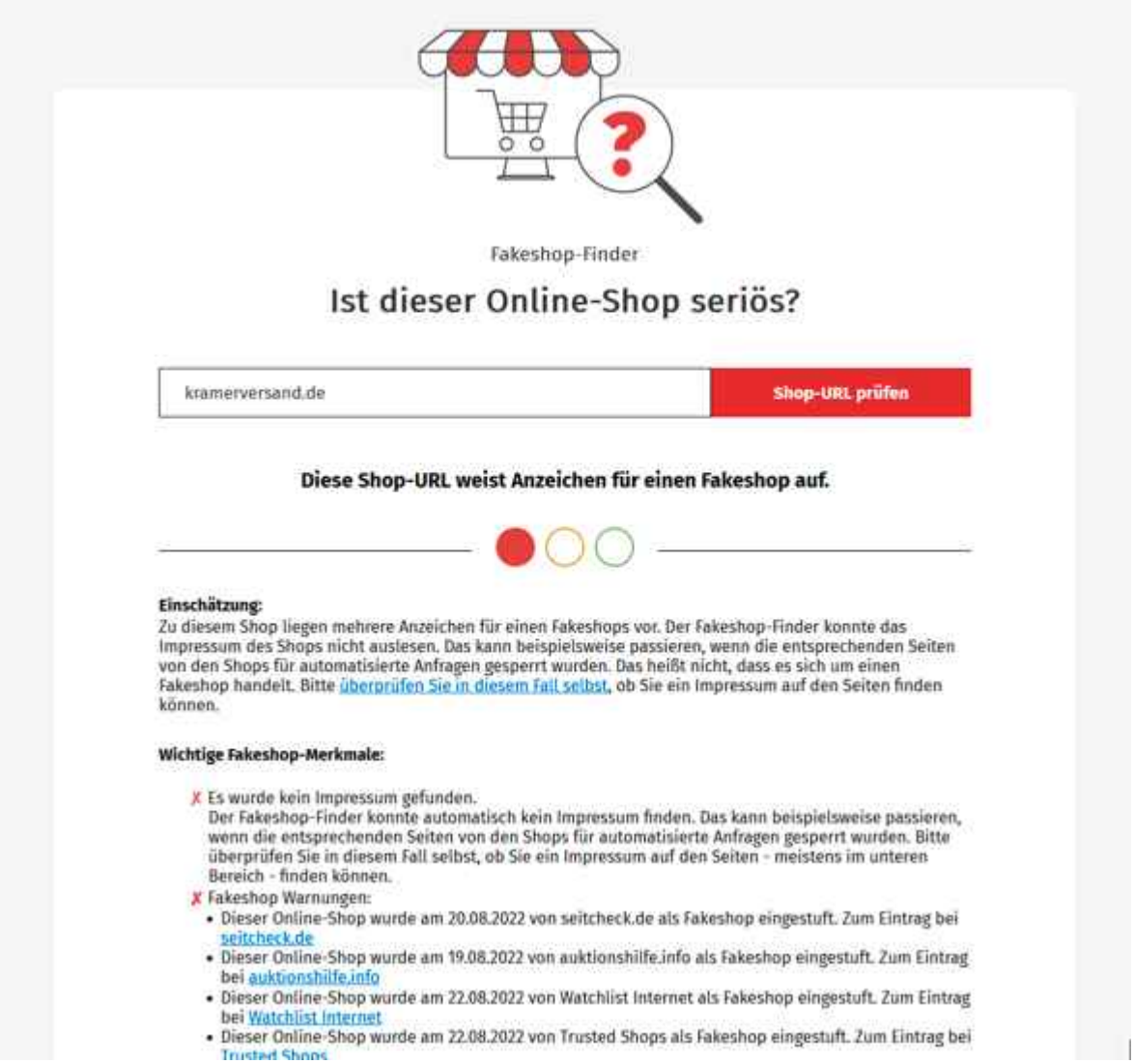
Als Zahlart bieten viele Fake-Shops ausschließlich Vorkasse per Banküberweisung an, da man solche Zahlungen in der Regel nicht rückgängig machen kann. Mitunter wollen betrügerische Händler Kunden auch gerne zu PayPal-Zahlungen in der Variante „Freunde und Familie“ verleiten. Die beinhalten aber im Unterschied zur Option „Waren und Dienstleistungen“ keinen Käuferschutz. Manchmal bietet der Fake-Shop auch zum Schein weitere Zahlarten an, um Vertrauen zu schaffen. Die

funktionieren dann aber aus vorgeschobenen Gründen nicht. Daraufhin bitten die Täter um Vorkasse oder die unsichere PayPal-Variante.

Auch bei vermeintlich sicheren Bezahlmethoden gibt es Haken. Der PayPal-Käuferschutz ist zum Beispiel an Bedingungen wie Paketversand mit elektronischer Sendungsverfolgung geknüpft [3]. Ähnlich halten es Amazon oder Klarna. Manche Betreiber von Fake-Shops schicken die Pakete daher an Adressen von Strohleuten, um Kunden über die Sendungsverfolgung erst in Sicherheit zu wiegen und anschließend Käuferschutzverfahren zu erschweren. Mehr zu Vor- und Nachteilen von Zahlarten haben wir unter [2] zusammengetragen.

Blacklists und Prüftools

Bleibt man unsicher, helfen Tools von Verbraucherschützern und anderen Organisationen. Zunächst lohnt sich ein Blick auf Blacklists. Hierbei handelt es sich um Listen von Online-Shops, die bereits als Fake-Shops eingestuft oder die mehrfach als solche gemeldet worden sind. Solche Listen finden sich zum Beispiel auf der Website der Verbraucherzentrale Hamburg, der Präsenz des Siegel-Anbieters Trusted Shops oder auf der Watchlist Internet. Der Fake-Shop-Kalender der Verbraucherzentrale Bundesverband macht zusätzlich auf zeitweise besonders häufig betroffene Branchen aufmerksam (alle Seiten unter ct.de/yu3d). Darüber hinaus kann sich der Besuch der Preisvergleichsseiten Geizhals und Idealo lohnen (Hinweis: Geizhals gehört wie c’t zu Heise Medien). Sie listen nur geprüfte Online-Shops sowie Händler auf Marktplätzen mit starkem Käuferschutz. Mehr zu den Eigenheiten von Marktplätzen wie Amazon und eBay finden Sie unter [3].



Mit dem Fakeshop-Finder der Verbraucherzentralen überprüft man Shop-Websites. Bei einer roten Ampel handelt es sich nahezu sicher um einen Fake-Shop.

Hilfreich bei der Recherche ist außerdem der Fakeshop-Finder der Verbraucherzentralen. Dort gibt man die URL des zu prüfenden Online-Shops in eine Eingabemaske ein. Anschließend ordnet das Tool ihn nach einem Ampelsystem einer Kategorie zu. Zeigt die Ampel Rot, so ist der betreffende Shop bereits als Fake-Shop aufgefallen. Bei gelber Ampelfarbe hat die automatische Prüfung allgemeine Indizien für betrügerische Absichten, aber auch Indizien für seriöses Gebaren gefunden und listet sie samt Erklärung auf. Entdeckt die Prüfroutine beispielsweise kein Impressum, kann das auch heißen, dass der Betreiber des Shops es lediglich für automatisierte Abfragen gesperrt hat. Das muss man dann selbst nachsehen. Die Einstufung „Grün“ bedeutet, dass der Shop den

Verbraucherzentralen „bisher nicht negativ aufgefallen“ ist; man soll aber trotzdem auf eine sichere Zahlungsmethode und die Rücksendekonditionen achten.

Schäden begrenzen,Shops melden

Ist das Kind bereits in den Brunnen gefallen, kann man versuchen, das im Fake-Shop ausgegebene Geld zurückzubekommen. Im besten Fall hat man eine sichere Zahlungsmethode verwendet und veranlasst über seine Bank oder den Zahlungsdienstleister eine Rückerstattung. Bei einer Banküberweisung wird es hingegen schwierig. Meldet man sich sofort oder zumindest am selben Tag bei seiner Bank, kann diese die Überweisung manchmal noch stoppen.

In jedem Fall sollte man Strafanzeige bei der Polizei oder Staatsanwaltschaft erstatten. Dies geht heutzutage unkompliziert über die „Onlinewache“ (ct.de/yu3d). Zusätzlich kann man einen Rechtsanwalt damit beauftragen, den Rückzahlungsanspruch auf zivilrechtlicher Ebene durchzusetzen. Der Anwalt beantragt Einsicht in die Ermittlungsakte der Strafverfolgungsbehörden und findet im besten Fall die Identität des Betrügers heraus.

Wer einen Fake-Shop erkannt hat oder darauf hereingefallen ist, kann dazu beitragen, dass der Shop aus dem Internet verschwindet. Hat man als Betroffener Strafanzeige erstattet, kümmern sich meist Polizei und Staatsanwaltschaft darum, dass der Hoster den Shop abschaltet. Ansonsten meldet man den Fake-Shop dem Hoster oder Shopsystemanbieter sowie den Verbraucherzentralen, zum Beispiel über das Onlineformular der Verbraucherzentrale Hamburg (ct.de/yu3d). *(mon@ct.de)*

1. Literatur
2. Jo Bager, Gefährliche Offenheit, Online-Handelsregister lädt zum Datenmissbrauch ein, c’t 24/2022, S. 134
3. Markus Montz, Geld her!, Onlinekauf-Checkliste

Bezahlmethoden, c’t 8/2022, S. 26

4. Georg Schnurer, Händler-Roulette, Onlinekauf-Checkliste Shop-Auswahl, c’t 8/2022, S. 24

Nützliche Websites: ct.de/yu3d

SQL Injection in Java mit JPA und Hibernate verhindern

entwickler.de – entwickler.de Deine Wissensplattform

[…]Weiterlesen…
Wirft man einen Blick auf die Top-10-Schwachstellen der OWASP [1], sind SQL Injections immer noch in einer prominenten Position zu finden. In diesem Artikel diskutieren wir verschiedene Möglichkeiten, wie SQL Injections effizient vermieden werden können.

Wenn Anwendungen auf Datenbanken zugreifen, bestehen immer wieder hohe Sicherheitsrisiken für die Applikation. Hat ein Angreifer die Möglichkeit, die Datenbankschicht einer Anwendung zu kapern, kann er zwischen mehreren Optionen wählen. Die Daten der gespeicherten Benutzer zu stehlen, um sie mit Spam zu überfluten, ist dabei nicht das schlimmste mögliche Szenario. Noch problematischer wäre es, wenn gespeicherte Zahlungsinformationen missbraucht würden. Eine weitere Variante eines SQL-Injection-Cyberangriffs ist der illegale Zugriff auf eingeschränkte kostenpflichtige Inhalte

und/oder Dienste. Wie wir sehen, gibt es viele Gründe, sich um die Sicherheit von (Web-)Anwendungen zu kümmern.

Um eine gut funktionierende Prävention gegen SQL Injections etablieren zu können, müssen wir zunächst verstehen, wie ein solcher Angriff funktioniert und auf welche Punkte wir achten müssen. Kurz gesagt verhält es sich so: Jede Benutzerinteraktion, die die Eingabe ungefiltert in einer SQL-Abfrage verarbeitet, ist ein mögliches Angriffsziel. Die Dateneingabe kann so manipuliert werden, dass die übermittelte SQL-Abfrage eine andere Logik enthält als das Original. Der folgende Code gibt eine gute Vorstellung davon, was möglich ist:

```
SELECT Username, Password, Role FROM User
  WHERE Username = 'John Doe' AND Password = 'S3cr3t';
SELECT Username, Password, Role FROM Users
  WHERE Username = 'John Doe'; --' AND Password='S3cr3t';
```

Die erste Anweisung zeigt die ursprüngliche Abfrage. Wird die Eingabe für die Variablen Benutzername und Passwort nicht gefiltert, entsteht das klassische Angriffsszenario. Die zweite Abfrage fügt für die Variable Benutzername einen String mit dem Benutzernamen *John Doe* ein und erweitert ihn um die Zeichen *; –*. Diese Anweisung umgeht die *UND*-Verzweigung und gibt in diesem Fall Zugriff auf das Log-in. Mit der Zeichensequenz *,;* schließen Sie die *WHERE*-Anweisung und mit *–* werden alle folgenden Zeichen auskommentiert. Theoretisch ist es möglich, zwischen diesen beiden Zeichenfolgen jeden gültigen SQL-Code auszuführen. Es lässt sich leicht ahnen, welcher Schabernack an dieser Stelle möglich ist.

Mein Plan ist natürlich nicht, zu verbreiten, welche SQL-Befehle die schlimmsten Folgen für das Opfer haben könnten. Bei diesem einfachen Beispiel gehe ich davon aus, dass die Botschaft klar angekommen ist. Wir müssen jede UI-Eingabevariable in unserer Anwendung vor Benutzermanipulation schützen. Auch dann, wenn sie nicht direkt für Datenbankabfragen verwendet werden. Um diese Variablen zu

erkennen, ist es immer eine gute Idee, alle vorhandenen Eingabeformulare zu validieren. Doch moderne Anwendungen haben meist mehr als nur ein paar Eingabeformulare. Aus diesem Grund sage ich auch sehr eindringlich: Behalten Sie Ihre REST-Endpunkte im Auge. Oft sind deren Parameter auch mit SQL-Abfragen verbunden.

Security Afternoon

Durch einen stetigen Strom an Releases, neuen Features und spannenden Projekten rückt Security in der IT-Welt gerne einmal in den Hintergrund. Beim Security Afternoon rücken wir mit Michael Kaufmann und Inko Lorch einen ganzen Nachmittag lang die IT-Sicherheit in den Fokus und zeigen, warum es so wichtig ist, Anwendungssicherheit nicht als lästige Fleißaufgabe zu verstehen.

Deshalb sollte die Eingabevalidierung generell Teil des Sicherheitskonzepts sein. Annotationen aus der Spezifikation Bean Validation [2] sind für diesen Zweck sehr mächtig. Beispielsweise sorgt *@NotNull* als Annotation für das Datenfeld im Domänenobjekt dafür, dass das Objekt nur persistiert werden kann, wenn die Variable nicht leer ist. Um die Bean Validation Annotations in Ihrem Java-Projekt zu verwenden, müssen Sie

lediglich eine kleine Bibliothek einbinden:

```
<dependency>
  <groupId>org.hibernate.validator</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>${version}</version>
</dependency>
```

Eventuell ist es notwendig, komplexere Datenstrukturen zu validieren. Mit regulären Ausdrücken haben Sie ein weiteres mächtiges Werkzeug an der Hand. Aber seien Sie vorsichtig: Es ist nicht so einfach, korrekt funktionierende RegEx zu schreiben. Schauen wir uns dazu ein kurzes Beispiel an (Listing 1).

Listing 1: Validierung durch reguläre Ausdrücke in Java

```
public static final String RGB_COLOR = "#[0-9a-fA-F]{3,3}([0-9a-fA-F]{3,3})?";

public boolean validate(String content, String regEx) {
  boolean test;
  if (content.matches(regEx)) {
    test = true;
  } else {
    test = false;
  }
  return test;
}

validate('#000', RGB_COLOR);
```

Die RegEx zur Erkennung des korrekten RGB-Farbschemas ist recht einfach. Gültige Eingaben sind *#ffF* oder *#000000*. Der Bereich umfasst die Zeichen *0-9* und zusätzlich noch Buchstaben *A-F*. Groß-/Kleinschreibung wird in unserem Beispiel nicht beachtet. Wenn Sie Ihre eigene RegEx entwickeln, müssen Sie bestehende Grenzen immer sehr gut im Auge behalten. Ein gutes Beispiel, um obere beziehungsweise untere Schranken zu verstehen, ist das 24-Stunden-Zeitformat. Typische Fehler sind ungültige Eingaben wie *23:60* oder *24:00*. Ein Blick auf die

Anzeige der Digitaluhr zeigt für ein 24-Stunden-Format als untere Schranke *00:00* und als obere Schranke *23:59*, alles andere ist ungültig.

Die Methode *validate* vergleicht die Eingabezeichenfolge mit der RegEx. Wenn das Muster mit der Eingabe übereinstimmt, gibt die Methode *TRUE* zurück. Wenn Sie weitere Ideen zu Validatoren in Java erhalten möchten, können Sie auch in meinem GitHub-Repository [3] nachsehen.

Zusammengefasst ist unsere erste Idee, um Benutzereingaben vor Missbrauch zu schützen, alle problematischen Zeichenfolgen herauszufiltern wie SQL-Kommentare und so weiter. Und solch eine Sperrliste ist auch nicht schlecht. Zumindest für den Anfang. Eine Blacklist weist aber einige Einschränkungen auf. Zunächst erhöht sich die Komplexität der Anwendung, da das Blockieren einzelner Zeichen wie *–;* und *,* manchmal unerwünschte Nebenwirkungen verursachen kann. Auch eine anwendungsweite Standardbegrenzung der Zeichen könnte Probleme bereiten. Stellen Sie sich vor, es gibt einen Textbereich für ein Blogsystem oder Ähnliches.

Das bedeutet, dass wir ein weiteres leistungsstarkes Konzept benötigen, um die Eingabe so zu filtern, dass unsere SQL-Abfrage nicht manipuliert werden kann. Um dieses Ziel zu erreichen, bietet der SQL-Standard eine sehr gute Lösung. SQL-Parameter sind Variablen innerhalb einer SQL-Abfrage, die als Inhalt und nicht als Anweisung interpretiert werden. Das ermöglicht es, große Texte entgegenzunehmen, ohne einige gefährliche Zeichen blockieren zu müssen. Schauen wir uns an, wie das mit einer PostgerSQL-Datenbank [4] funktioniert:

```
DECLARE user String;
SELECT * FROM login WHERE name = user;
```

Für den Fall, dass Sie den OR-Mapper Hibernate [5] verwenden, gibt es mit dem Java Persistence API (JPA) einen eleganteren Weg (Listing 2).

Listing 2: Hibernate-JPA-SQL-Parameter verwenden

```
String myUserInput;

@PersistenceContext
public EntityManager mainEntityManagerFactory;

CriteriaBuilder builder =
  mainEntityManagerFactory.getCriteriaBuilder();

CriteriaQuery<DomainObject> query =
  builder.createQuery(DomainObject.class);

// create Criteria
Root<ConfigurationDO> root =
  query.from(DomainObject.class);

//Criteria SQL Parameters
ParameterExpression<String> paramKey =
  builder.parameter(String.class);

query.where(builder.equal(root.get("name"), paramKey);

// wire queries together with parameters
TypedQuery<ConfigurationDO> result =
  mainEntityManagerFactory.createQuery(query);

result.setParameter(paramKey, myUserInput);
DomainObject entry = result.getSingleResult();
```

Listing 2 zeigt ein vollständiges Beispiel für Hibernate mit JPA und dem Criteria API. In der ersten Zeile wird die Variable für die Benutzereingabe deklariert. Die Kommentare in der Auflistung erklären sehr deutlich, wie es funktioniert. Wie Sie sehen können, ist das keine Raketenwissenschaft. Die Lösung hat neben der Verbesserung der Sicherheit von Webanwendungen einige weitere nette Vorteile. So wird kein einfaches SQL verwendet. Dadurch wird sichergestellt, dass jedes Datenbankverwaltungssystem, das von Hibernate unterstützt wird, durch diesen Code gesichert werden kann.

Die Nutzung sieht vielleicht etwas komplizierter aus als eine einfache Abfrage, aber der gewonnene Nutzen für Ihre Anwendung ist enorm. Andererseits gibt es natürlich einige zusätzliche Codezeilen. Aber die sind nicht so schwer zu verstehen, wie dieser Artikel gezeigt hat.

Marco Schulz studierte an der HS Merseburg Diplominformatik und twittert regelmäßig als @ElmarDott über alle möglichen technischen Themen. Seine Schwerpunkte sind hauptsächlich Build- und Konfigurationsmanagement, Softwarearchitekturen und Release-Management. Seit knapp 20 Jahren realisiert er in internationalen Projekten für namhafte Unternehmen umfangreiche Webapplikationen. Er ist freier Consultant/Trainer. Sein Wissen teilt er mit anderen Technikbegeisterten auf Konferenzen, wenn er nicht gerade wieder einmal an einem neuen Fachbeitrag schreibt.

Links & Literatur

[1] https://owasp.org

[2] https://beanvalidation.org

[3] https://github.com/ElmarDott/TP-CORE/blob/master/src/main/java/org/europa/together/utils/Validator.java

[4] https://www.postgresql.org/docs/9.1/plpgsql-declarations.html

[5] https://hibernate.org

[6] https://elmar-dott.com/courses/de/web-application-securiy

[7] Originalartikel: https://elmar-dott.com/articles/preventing-sql-injections-in-java/

TLS mit Wireshark entschlüsseln

TLS mit Wireshark entschlüsseln

Was es beim kriminellen Man in the Middle zu verhindern gilt, gehört bei legal agierenden Systemadmins zum notwendigen Handwerkszeug: der Zugriff auf verschlüsselte Datenströme zwecks Fehlersuche.
Was es beim kriminellen Man in the Middle zu verhindern gilt, gehört bei legal agierenden Systemadmins zum notwendigen Handwerkszeug: der Zugriff auf verschlüsselte Datenströme zwecks Fehlersuche.

Von Benjamin Pfister

Der Anteil des verschlüsselten Datenverkehrs nimmt ständig zu. Fast alle Webdienste nutzen Transport Layer Security (TLS) und aktuelle Browser warnen bei unverschlüsselten HTTP-

Verbindungen ausdrücklich vor dem damit verbundenen Risiko. Das ist aus Sicht der Sicherheit und des Datenschutzes sehr zu begrüßen – doch die Verschlüsselung verhindert auch eine legale Analyse des Datenstroms, etwa seitens berechtigter Admins. Es gibt jedoch Möglichkeiten der Fehlersuche trotz TLS-Verschlüsselung, zum Beispiel mit dem im Folgenden beschriebenen Paketanalysewerkzeug Wireshark.

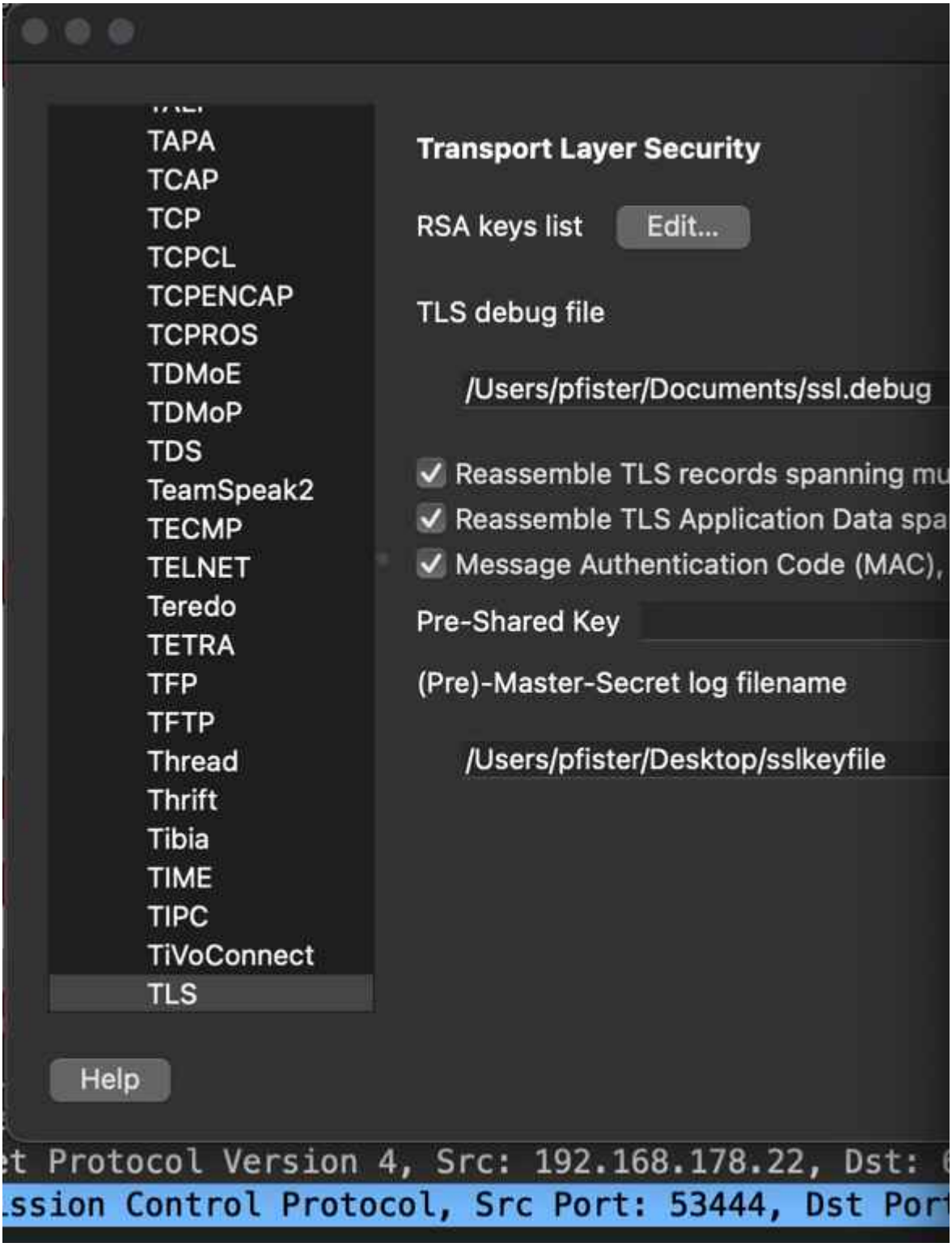
Wireshark bringt einen eigenen Dissector (wörtlich übersetzt Sezierer) für TLS mit. Er ermöglicht neben der Aufteilung und Darstellung der Protokolle auch die Entschlüsselung der Nutzdaten. Dazu bedarf es der passenden Schlüssel. Je nach eingesetzter Cipher Suite kommen unterschiedliche Entschlüsselungsmethoden zum Einsatz: auf Basis eines Session- (Pre-Master Secret) oder eines privaten RSA-Schlüssels.

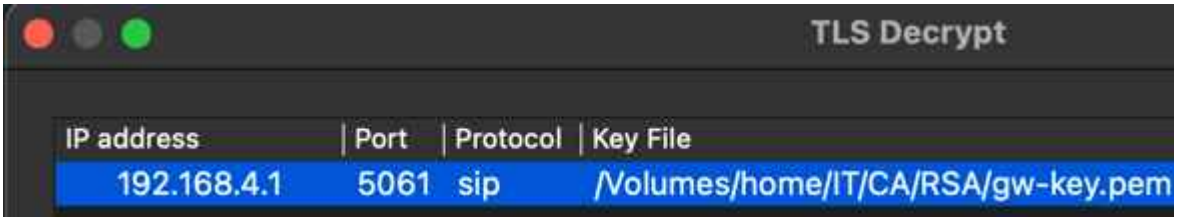
Welche der beiden zur Anwendung kommt, hängt von der Cipher Suite ab: mit Perfect Forward Secrecy (PFS) oder ohne. Falls für die Übertragung keine PFS Cipher Suites vorgesehen sind, kann die Entschlüsselung auf Basis des privaten Schlüssels des Serverauthentifizierungszertifkats stattfinden. In diesem Fall kann Wireshark jedoch auch die Methode Pre-Master Secret nutzen. Dies ist beispielsweise dann interessant, wenn man – wie bei öffentlichen Webdiensten – nicht im Besitz der privaten Schlüssel ist.

Bei Nutzung von Perfect Forward Secrecy (PFS) lässt sich der Datenstrom selbst bei Kenntnis des privaten Schlüssels nicht nachträglich entschlüsseln. Daher empfiehlt das BSI zum Schutz personenbezogener oder anderer sensibler Daten diese Cipher Suites. Darunter fallen die Cipher Suites mit Diffie-Hellman Ephemeral (DHE) und Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). Um diese Varianten zu entschlüsseln, muss man die Methode Pre-Master Secret einsetzen.

Der Besitz des privaten Schlüssels nützt also nur dann etwas, wenn keine (EC)DHE Cipher Suites zum Einsatz kommen. Zudem funktioniert diese erste Variante nicht mit TLS 1.3. Einen

weiteren Fallstrick birgt der TLS Session Resume, bei dessen Anwendung das Entschlüsseln fehlschlägt. Es bedarf der Aufzeichnung eines ClientKeyExchange im TLS Handshake. Zum Entschlüsseln der Daten benötigt man das Serverauthentifizierungszertifikat – genauer dessen privaten Schlüssel. In den TLS-Protokolleinstellungen von Wireshark und dem Menüpunkt „RSA keys list“ referenziert man die Datei mit dem privaten Schlüssel und verknüpft ihn mit der IP-Adresse, dem Port und dem Protokoll des Servers. Abbildung 1 zeigt eine solche Hinterlegung für die IP-Adresse 192.168.4.1 mit dem Port 5061 und dem Protokoll SIP. Der Private Key liegt im Beispiel unter /Volumes/Home/IT/CA/RSA/gw-key.pem. Daran erkennt man, dass nicht nur HTTPS als Applikationsprotokoll zur Verfügung steht. Die Referenzen liegen im Beispiel von macOS unter /Users/<username>/.config/wireshark/ssl_keys.





Hinterlegung des Private Key aus dem Serverauthentifizierungszertifikat (Abb. 1).
Nach der korrekten Hinterlegung beginnt der Dissector mit der Entschlüsselung. Bei eventuellen Fehlern lohnt ein Blick in

die TLS-Debug-Datei, die beispielsweise fehlerhafte Private-Key-Zuweisungen oder Probleme beim Laden der Private Keys aufzeigt. Deren Zielverzeichnis und Namen kann man selbst wählen (siehe Abbildung 1).

Auf der Kommandozeile kann man das in Wireshark enthaltene CLI-Tool tshark nutzen. Für die RSA-Methode lautet der Befehl

```
tshark -o "ssl.keys_list: 192.168.4.1,5061,sip,/Volumes/Home/IT/CA/RSA/gw-key.pem" -r siptls.pcapng -Y sip
```

Über die Option

```
-o "ssl.keys_list: 192.168.4.1,5061,sip,/Volumes/Home/IT/CA/RSA/gw-key.pem"
```

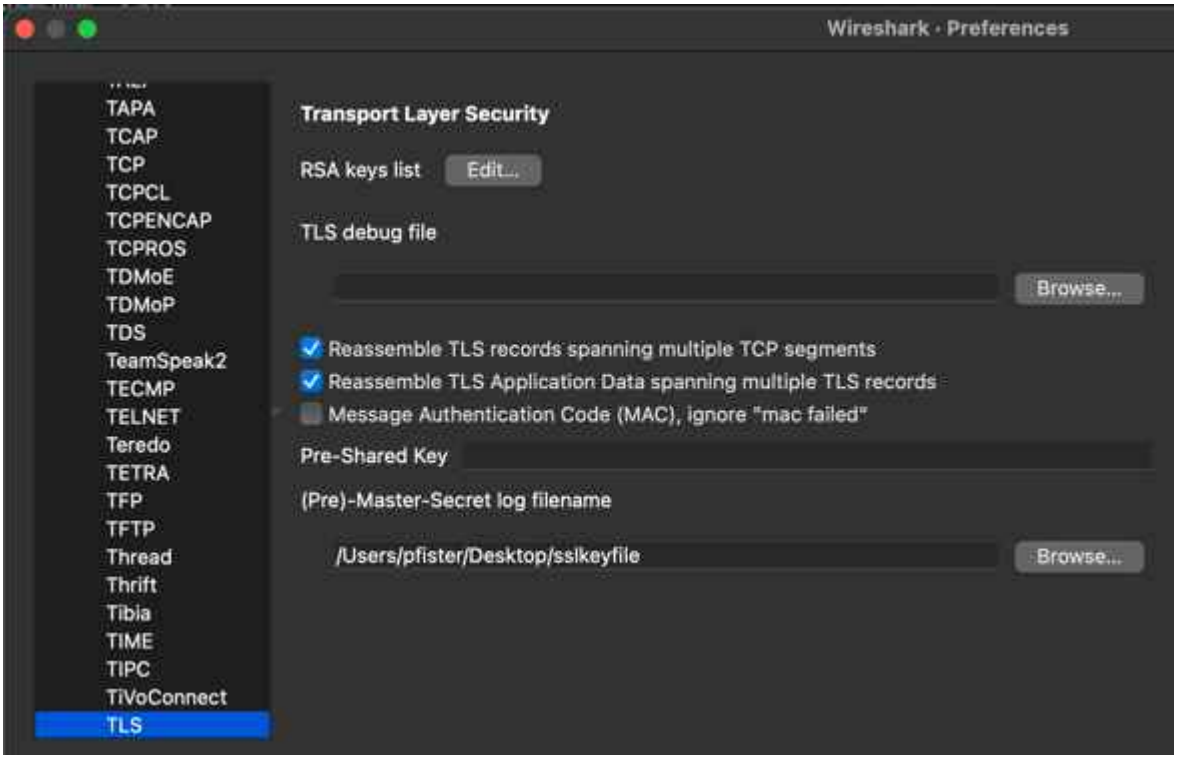
verknüpft man die in Abbildung 1 dargestellten Einstellungen – ähnlich wie mit der GUI-Variante. Das Argument -r siptls.pcapng liest dabei lediglich die PCAPNG-Datei. Das Argument -Y sip setzt einen Display-Filter auf das VoIP-Signalisierungsprotokoll SIP, sodass keine Pakete anderer Protokolle die Ausgabe fluten.

Die zweite Variante – keine Kenntnis des privaten Schlüssels und der Einsatz von (EC)DHE – setzt eine Keylog-Datei voraus, also eine Textdatei, die von unterschiedlichen Kryptobibliotheken bereitgestellt wird, beispielsweise OpenSSL oder NSS. Darauf aufbauende Applikationen wie Chrome, Firefox oder Curl generieren diese Datei, wenn die Umgebungsvariable SSLKEYLOGFILE gesetzt ist. Unter macOS kann man diese beispielsweise wie folgt anlegen: export SSLKEYLOGFILE="/Users/<username>/Desktop/sslkeyfile".

Die Bibliotheken schreiben den Pre-Master Key dann in die in der Umgebungsvariablen referenzierte Datei. Der Client generiert diesen in der Client Exchange Phase des TLS Handshake. Der Export kann auf Client- oder Serverseite stattfinden. Ein Mitlesen auf dem Transportweg ist somit nicht möglich. Wireshark kann den Pre-Master Key aus dem Handshake

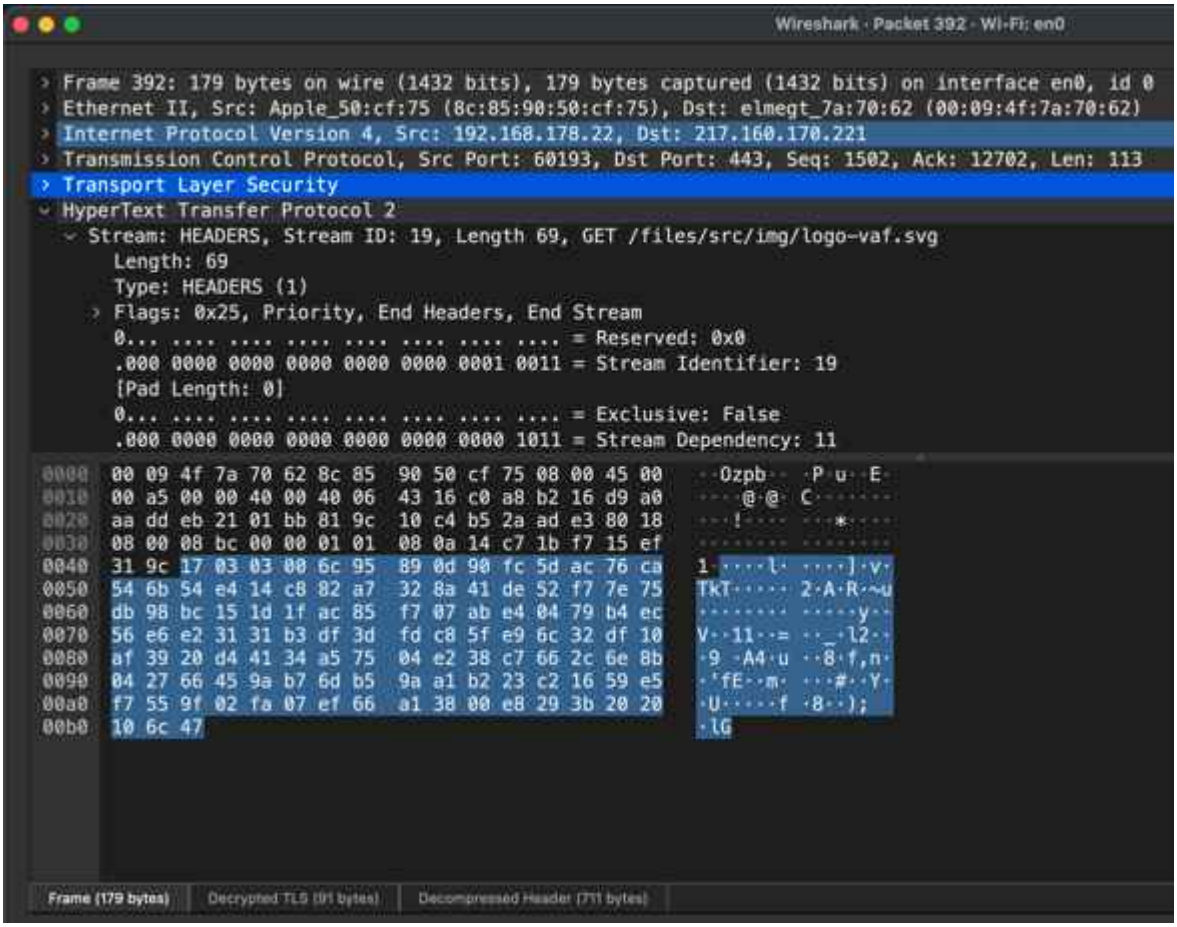
dafür nutzen, den Master Key abzuleiten und damit den Datenverkehr zu entschlüsseln. Im Anschluss an die Konfiguration der Umgebungsvariablen startet man den Mitschnitt in Wireshark und öffnet dann über die Konsole beispielsweise Firefox mittels open /Applications/Firefox.app unter macOS. Nachdem die erste TLS-verschlüsselte Seite aufgerufen wurde, zeigt sich, ob die Schlüsseldatei korrekt gespeichert wurde. In der ersten Zeile der Datei erkennt man auch, dass sie Bibliothek NSS für den Schreibvorgang zuständig war (siehe Abbildung 2).

Nach dem Laden der ersten TLS-verschlüsselten Daten zeigt sich am SSLKEYLOG, ob die Schlüsseldatei korrekt gespeichert wurde (Abb. 2).
Damit Wireshark die Datei mit den passenden Schlüsseln zum Entschlüsseln heranzieht, ist sie als „(Pre)-Master-Secret log filename“ unter „Preferences/Protocols/TLS“ zu referenzieren (siehe Abbildung 3).



Referenzierung der PMK-Datei in den TLS-Protokolleinstellungen in Wireshark – in diesem Fall /Users/pfister/Desktop/sslkeyfile (Abb. 3).

Sobald der TLS Dissector in Wireshark den Traffic entschlüsselt hat, wird der HTTP2-GET-Request im Klartext lesbar (siehe Abbildung 4). Dass eine Entschlüsselung stattgefunden hat, zeigen die Angabe „HyperText Transport Protocol 2“ unterhalb der Zeile „Transport Layer Security“ und der Hinweis „Decrypted TLS“ im unteren Bereich.



Entschlüsselter HTTP2-GET-Request (Abb. 4).

Wer die Kommandozeile bevorzugt, kann mit tshark arbeiten – es folgt ein Beispiel einer Aufzeichnung und Entschlüsselung per tshark. Zunächst wird wieder die Umgebungsvariable angelegt, gefolgt vom Öffnen des Browsers Mozilla Firefox. Anschließend startet tshark für 60 Sekunden (-a duration:60) ohne direkte Ausgabe (-Q) und schreibt die aufgezeichneten Daten in eine PCAPNG-Datei (-w /Users/pfister/Desktop/tls_decrypt.pcapng). In der letzten Zeile liest tshark die PCAPNG-Datei (-r) mit dem Argument für die Referenz zur Keylog-Datei ein (-o tls.keylog_file:$SSLKEYLOGFILE) und filtert die Ausgabe über einen Displayfilter auf HTTP (-Y http):

```
export SSLKEYLOGFILE="/Users/<username>/Desktop/sslkeyfile"
open /Applications/Firefox.app
tshark -Q -a duration:60 -w /Users/pfister/Desktop/tls_decrypt.pcapng &
tshark -r /Users/pfister/Desktop/tls_decrypt.pcapng -o tls.keylog_file:$SSLKEYLOGFILE -Y http
```

Fazit

Mit der Session-Key-Methode lassen sich selbst aktuelle Protokolle wie TLS 1.3 entschlüsseln. Dafür bedarf es jedoch einer Applikation, die den Sessionschlüssel in eine Logdatei schreibt. Falls dies nicht der Fall ist und Server und Client keine (EC)DHE Cipher Suite nutzen, kann der Analyst als Fallback die RSA-Methode anwenden. Grundsätzlich kann die Möglichkeit einer Entschlüsselung ein Troubleshooting jedenfalls immens erleichtern. Wireshark bietet dafür einen recht einfach zu nutzenden Ansatz. (un@ix.de)

1. Quellen
2. Weiterführende Informationen finden sich unter ix.de/ztmc.