Cyberrisiko-Check für KKUs nach DIN

Kleine Unternehmen sind genauso von Cyberangriffen bedroht wie die großen, haben häufig aber weder Mittel noch speziell geschultes Personal, um sich zu schützen. Ein neuer Standard soll ihnen zu einer Einschätzung der Risiken und entsprechenden Schutzmaßnahmen verhelfen.

Mehr lesen …

EU-Vorschriften zu mehr Cybersicherheit

Die Europäische Union bringt im Rahmen ihrer Digitalstrategie zwei wichtige Gesetze auf den Weg: den Cyber Resilience Act (CRA) sowie die sogenannte NIS2-Richtlinie.Da wir immer digitaler werden, muss auch immer mehr Wert auf Onlinesicherheit, oder wie es auf Neudeutsch heißt, Cybersecurity, gelegt werden.

Die EU-Kommission hat am 15. September 2022 einen Entwurf des CRA vorgeschlagen, der noch vom europäischen Parlament und vom Rat angenommen werden muss.

Recht auf Updates durch den CRA

Mit dem CRA werden erhöhte Sicherheitspflichten auf Hersteller, Vertreiber, Importeure und Händler von IT-Produkten zukommen. Dazu zählen insbesondere geplante Meldepflicht für aktiv ausgenutzte Schwachstellen und

Sicherheitsvorfälle. Durch die Vorgaben des CRA sollen sicherere Hardware- und Softwareprodukte gewährleistet werden.

Zu den dabei erfassten Produkten zählt jedes Software- oder Hardwareprodukt sowie seine Datenfernverarbeitungslösungen, einschließlich Software- oder Hardwarekomponenten, die separat in Verkehr gebracht werden. Er ist anwendbar auf „Produkte mit digitalen Elementen“, also auf solche Produkte, die ohne ihre digitalen Elemente nicht sinnvoll genutzt werden können (z. B. Smartphones). Der CRA teilt die Produkte mit digitalen Elementen in drei Kategorien ein:

- Standardkategorie
- kritische Klasse I
- kritische Klasse II

In die Standardkategorie fallen voraussichtlich gut 90 Prozent aller Produkte, wie z. B. Textverarbeitung, Fotobearbeitung oder Festplatten. Zum CRA gehören verschiedene Anhänge. Darin, nämlich in Anhang III, werden die kritischen Produkte mit digitalen Elementen der Klassen I und II aufgeführt. Zur kritischen Klasse I zählen u. a. Software für Identitätsmanagementsysteme, Browser, Passwortmanager, Antivirensoftware, VPN-Lösungen, Netzwerkmanagementsysteme, Werkzeuge zur Verwaltung der Netzwerkkonfiguration, Systeme zur Überwachung des Netzwerkverkehrs, Verwaltung von Netzwerkressourcen, Systeme zur Verwaltung von Sicherheitsinformationen und -ereignissen (SIEM), Update-/Patch-Verwaltung (einschließlich Bootmanager), Systeme zur Verwaltung der Anwenderkonfiguration, Software zur Verwaltung mobiler Geräte, Firewalls, Router/Modems, Anwenderspezifische integrierte Schaltungen (ASIC) oder auch Industrielle Automatisierungs- und Steuerungssysteme (IACS). Zur kritischen Klasse II zählen hingegen insbesondere Betriebssysteme für Server, Desktops und mobile Geräte, Infrastrukturen für öffentliche Schlüssel und Aussteller digitaler Zertifikate,

Hardwaresicherheitsmodule (HSM), sichere Kryptoprozessoren, Smartcards, Smartcard-Lesegeräte und Token oder auch Geräte des industriellen Internets der Dinge.

Folgende Pflichten sollen zukünftig auf die vom Anwendungsbereich des CRA erfassten Unternehmen zukommen:

- Berücksichtigung der Cybersicherheit schon in der Planungs-, Entwurfs- und Entwicklungsphase sowie auch in der Produktions-, Liefer- und Wartungsphase („Security by Design“)
- umfangreiche Dokumentationspflichten in Bezug auf Cybersicherheitsrisiken
- Meldepflicht für aktiv ausgenutzte Schwachstellen und Vorfälle
- Überwachungs- und Beseitigungspflichten von Schwachstellen während der erwarteten Produktlebensdauer (max. fünf Jahre)
- Pflicht zur Lieferung von klaren und verständlichen Gebrauchsanweisungen
- Pflicht zur Bereitstellung von bestimmten Pflichtinformationen (u. a. Name, Anschrift und Kontaktdaten des Herstellers, Typen-, Chargen-, Versions- bzw. Seriennummer, Verwendungszweck, Art der technischen Sicherheitsunterstützung, die der Hersteller anbietet, sowie der Zeitpunkt, bis zu dem sie geleistet wird)
- Pflicht zur Bereitstellung von Sicherheitsupdates für jedenfalls fünf Jahre

Ganz konkret und unabhängig von der jeweiligen Kategorie müssen Produkte mit digitalen Elementen zudem immer einer Risikobewertung unterzogen werden.

Unter die Produkte mit digitalen Elementen im Sinne des CRA fallen jedoch weder „Produkte mit digitalen Inhalten“ noch „digitale Produkte“. Die Erstgenannten zeichnen sich dadurch

aus, dass der digitale Teil des Produkts für dessen Funktionsfähigkeit nicht von zentraler Bedeutung ist (z. B. Kühlschrank mit Bestellfunktion via App). Bei den „digitalen Produkten“ handelt es sich um rein digitale Produkte (z. B. Apps oder Musikdateien). Der CRA hat folglich einen sehr weit gefassten Anwendungsbereich, von dem nur ein paar spezifische Produktkategorien ausgenommen werden, wie beispielsweise Medizinprodukte. Software, die als Dienstleistung angeboten wird, also Software-as-a-Service-bzw. Cloudleistungen, wird ebenfalls gesondert geregelt.

Sichere Infrastrukturen durch NIS2

Speziell auf den Bereich der sogenannten kritischen Infrastruktur (KRITIS) zielt die NIS2-Richtlinie ab. Es geht also um den besseren Schutz von Stromversorgung, Wasserwerken oder Telekommunikationsleitungen. Es soll ein Höchstmaß an Ausfallsicherheit gewährleistet werden, um beispielweise Strom-Blackouts oder Störungen der Trinkwasserversorgung für die Bevölkerung möglichst zu vermeiden oder jedenfalls so schnell und gut wie möglich auf derartige Störungen reagieren zu können.

In Deutschland ist mit Blick auf den KRITIS-Sektor bereits 2015 das IT-Sicherheitsgesetz (IT-SiG) in Kraft getreten. Darin war auch eine Änderung des damaligen § 13 Telemediengesetz (TMG) enthalten, der in einem Absatz 7 die Pflicht für alle Websitebetreiber zur Absicherung ihrer Websites mit sich brachte (z. B. durch Verschlüsselung nach dem Stand der Technik per SSL-/TLS-Zertifikat). Diese Norm findet sich nach der letzten Änderung des TMG nun in § 19 Abs. 4 des Telekommunikations-Telemedien-Datenschutz-Gesetzes (TTDSG). Seit Mai 2021 ist nunmehr das zweite IT-Sicherheitsgesetz (IT-SiG2) in Kraft, welches sowohl den Adressatenkreis als auch den Pflichtenkatalog der KRITIS-Betreiber merklich erweitert hat.

Aber auch auf EU-Ebene tut sich einiges. 2016 ist die Richtlinie (EU) 2016/1148 des Europäischen Parlaments und des Rates vom 6. Juli 2016 über Maßnahmen zur Gewährleistung eines hohen gemeinsamen Sicherheitsniveaus von Netz- und Informationssystemen in der Union (kurz: NIS-Richtlinie) in Kraft getreten. Sie ist der europäische Rahmen für Cybersecurity im KRITIS-Bereich und soll ein hohes Sicherheitsniveau für Netzwerke und Informationssysteme sicherstellen.

Seit dem Jahr 2021 wird die NIS-Richtlinie überarbeitet. Ihr Nachfolger, die sogenannte NIS2-Richtlinie, soll den bestehenden Rechtsrahmen modernisieren, um die Herausforderungen des zunehmenden Grades an Digitalisierung und der stetig wachsenden Bedrohungen für die Cybersicherheit meistern zu können. Nach Inkrafttreten der NIS2-Richtlinie muss diese noch in das jeweilige nationale Recht der EU-Mitgliedsstaaten umgesetzt werden. In NIS2 wird zwischen kritischen und wichtigen Einrichtungen unterschieden:

- *Kritische Einrichtungen:* Energie (Strom, Fernwärme und Fernkälte, Erdöl, Erdgas und Wasserstoff); Verkehr (Luft, Schiene, Wasser und Straße); Bankenwesen; Finanzmarktinfrastrukturen; Gesundheitswesen; Herstellung pharmazeutischer Erzeugnisse (einschließlich Impfstoffe und kritischer Medizinprodukte); Trinkwasserversorgung; Abwasserwirtschaft; digitale Infrastrukturen (Internetknoten, DNS-Anbieter, Anbieter von Clouddienstleistungen, Anbieter von Rechenzentrumsdiensten, Netze zur Bereitstellung von Inhalten, öffentliche elektronische Kommunikationsnetze und elektronische Kommunikationsdienste,…); öffentliche Verwaltung; Weltraum.
- *Wichtige Einrichtungen:* Post- und Kurierdienste; Abfallwirtschaft; Chemikalien; Lebensmittel; Herstellung anderer Medizinprodukte, von Computern, Elektronik und Kraftfahrzeugen sowie Maschinenbau; Anbieter digitaler

Dienste (Onlinemarktplätze, Onlinesuchmaschinen und Plattformen der sozialen Netzwerke).

Sowohl die kritischen als auch die wichtigen Einrichtungen müssen u. a. folgende Cybersecurity-Maßnahmen treffen:

- Erlass und Umsetzung von Richtlinien für Risiken und Informationssicherheit
- Umsetzung von Maßnahmen zur Prävention, Detektion und Bewältigung von Cybersecurity-Vorfällen (Sicherheitspannen)
- Ergreifen von Maßnahmen zum Business Continuity Management (BCM) inkl. Backup- bzw. Krisenmanagement
- Gewährleistung der Sicherheit bei der Beschaffung von IT- und Netzwerksystemen
- Beachtung von Vorgaben für Kryptografie bzw. Verschlüsselung
- Umsetzung angemessener Maßnahmen zur Zugangskontrolle
- Einsatz sicherer Sprach-, Video- und Textkommunikation
- Einsatz gesicherter Notfallkommunikationssysteme

Durch die NIS2-Richtlinie wird der Aspekt der Cybersecurity zukünftig in der gesamten Lieferkette zu berücksichtigen sein. Außerdem sollen die Aufsicht und die Zusammenarbeit zwischen den Behörden und den von NIS2 betroffenen Betreibern innerhalb der EU vertieft werden. Die Sanktionen bei Verstößen gegen die NIS2-Vorgaben sollen durch die einzelnen EU-Mitgliedsstaaten selbst geregelt werden. Allerdings wird von Seiten des EU-Gesetzgebers bestimmt, dass die Sanktionen wirksam, verhältnismäßig und abschreckend sein müssen. Gegen kritische Einrichtungen sollen Geldbußen mit einem Höchstbetrag von mindestens 10 Mio. Euro oder von mindestens 2 Prozent des gesamten weltweit erzielten Vorjahresumsatzes verhängt werden können. Bei Sanktionen gegen wichtige Einrichtungen soll der Höchstbetrag mindestens 7 Mio. Euro oder 1,4 Prozent des Vorjahresumsatzes betragen.

Praxistipp

Neben dem CRA und NIS2 beinhaltet die Strategie der EU noch weitere Gesetze, die den Umgang mit digitalen Daten regeln sollen. Dazu zählen insbesondere der Data Governance Act (DGA) zur Förderung der Weiterverwendung von Daten des öffentlichen Sektors, der Digital Markets Act (DMA) und der Digital Services Act (DSA) zur Regulierung großer Onlineplattformen, der Artificial Intelligence Act (AIA) zur Regulierung von Künstlicher Intelligenz (KI) oder auch der Data Act (DA) zur besseren Weiterverwendung von Unternehmensdaten. Bei diesen Rechtsakten handelt es sich nicht um bloße Zukunftsmusik, denn der DMA ist bereits seit dem 1. November 2022 in Kraft. Der DGA wird ab dem 24. September 2023 anwendbar sein, der DSA bereits ab dem 2. Mai 2023.

Michael Rohrlich hat als Rechtsanwalt und Fachautor seinen Kanzleisitz in Würselen, Nähe Aachen. Seine beruflichen Schwerpunkte liegen auf dem Gebiet des Onlinerechts sowie des gewerblichen Rechtsschutzes. Weitere Infos zu den Themen aus den Rechtsbeiträgen sowie Gesetze und Gerichtsentscheidungen bietet er unter www.rechtssicher.info an.

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.

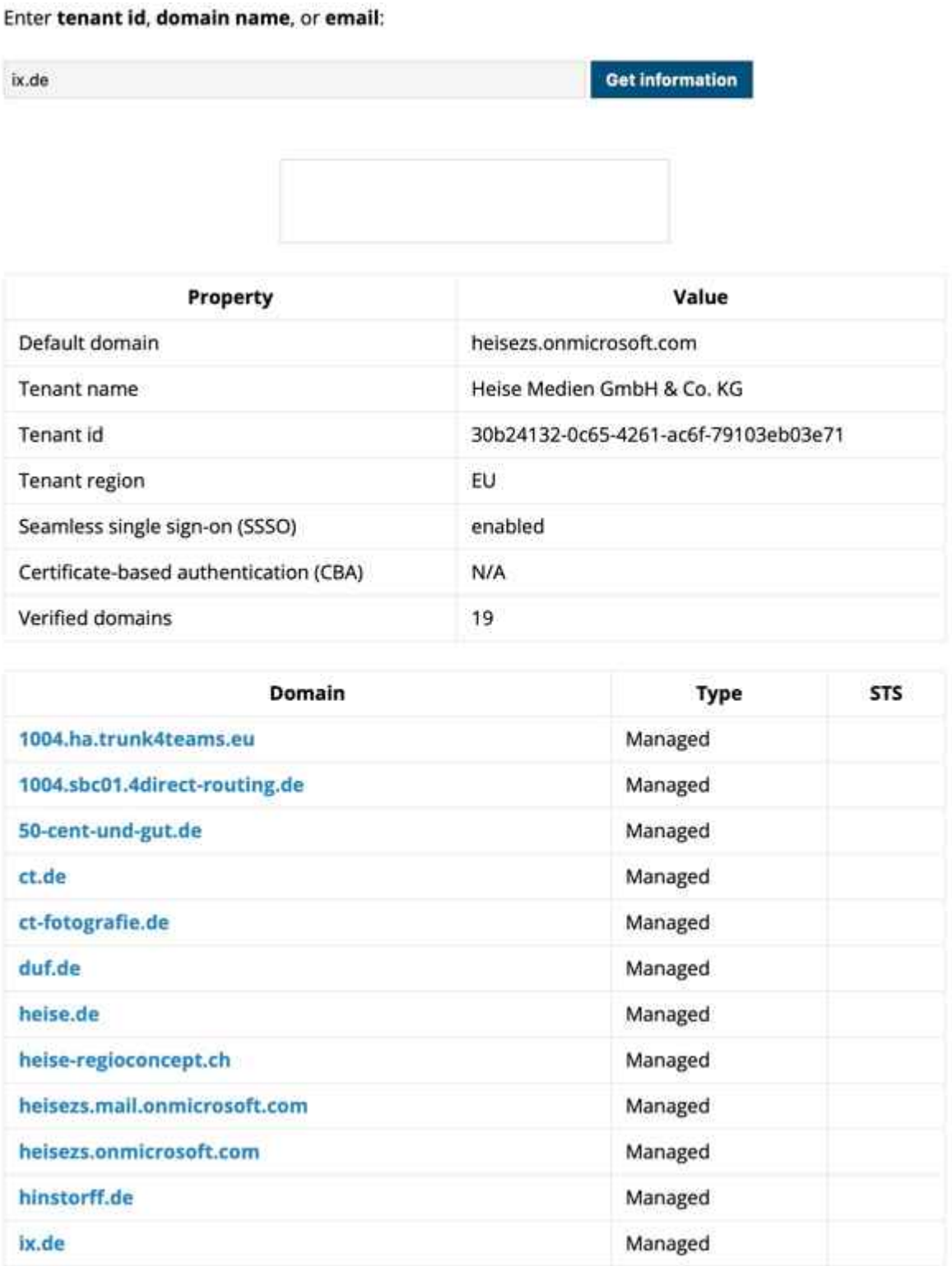
Hyppönens radikale Zukunftsvision: Computer werden in fast allem besser als Menschen sein (Abb. 1).

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.

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

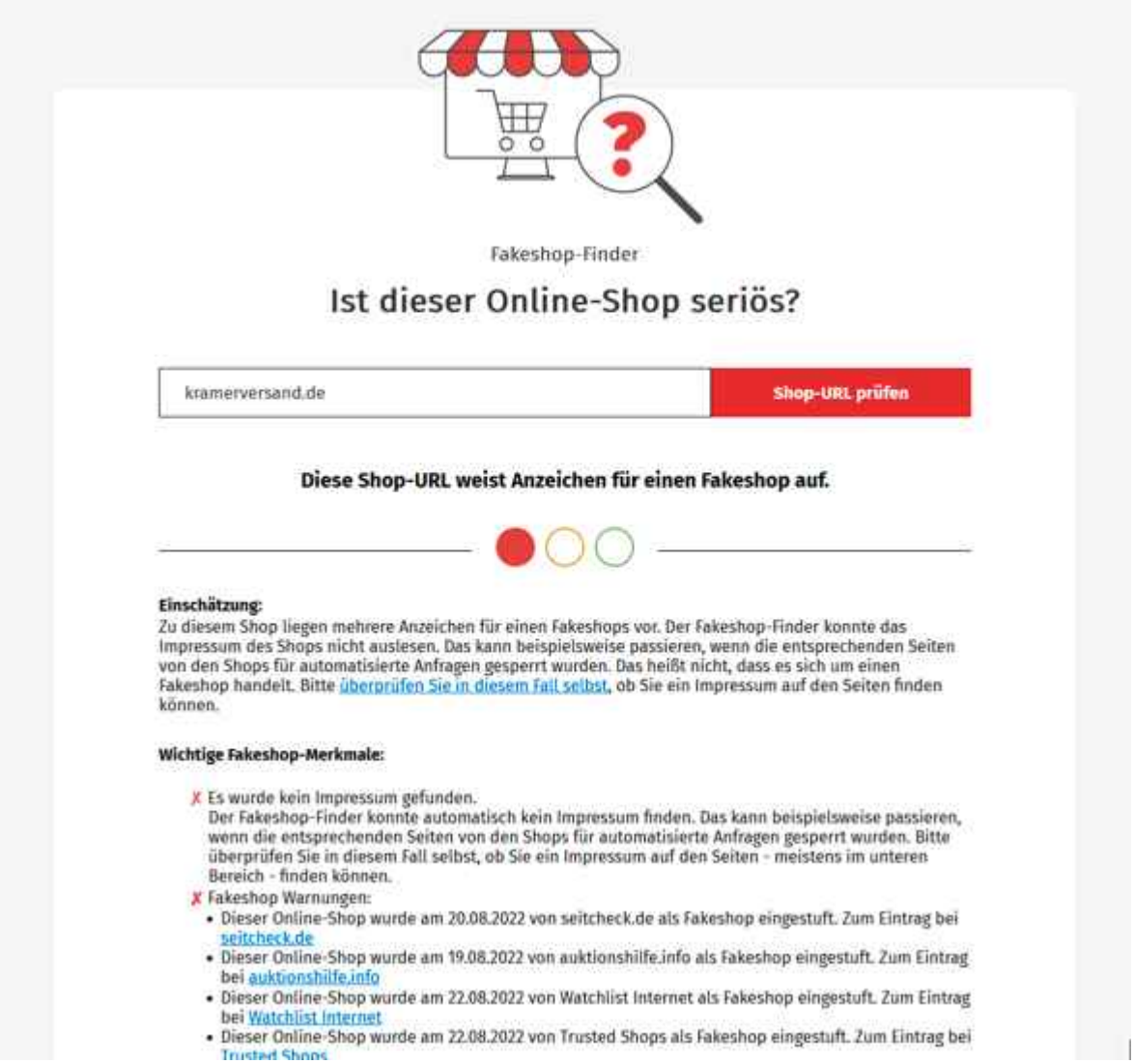
Als Zahlart bieten viele Fake-Shops ausschließlich Vorkasse per Banküberweisung an, da man solche Zahlungen in der Regel nicht rückgängig machen kann. Mitunter wollen betrügerische

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.

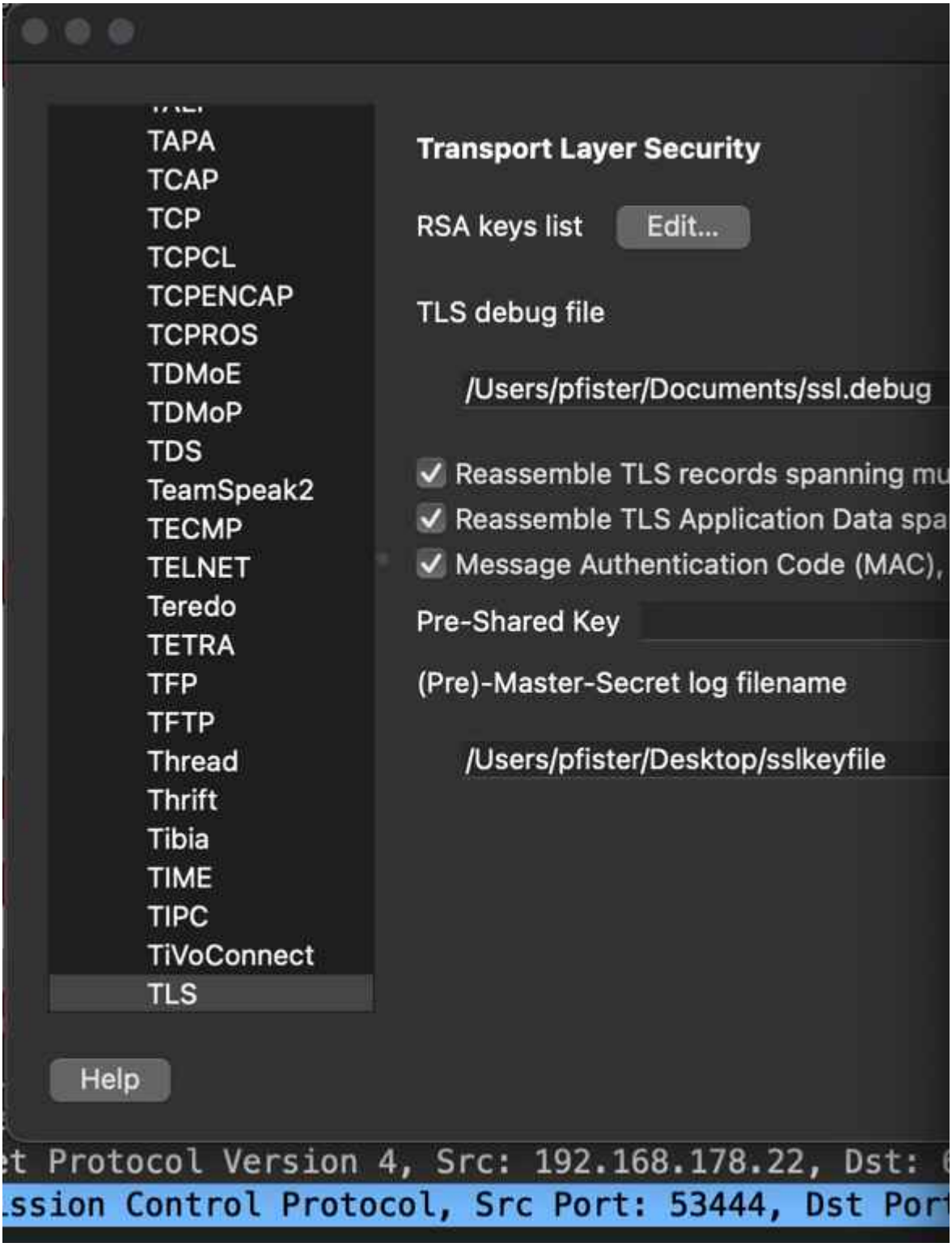
Wireshark bringt einen eigenen Dissector (wörtlich übersetzt Sezierer) für TLS mit. Er ermöglicht neben der Aufteilung und Darstellung der Protokolle auch die Entschlüsselung der Nutzdaten. Dazu bedarf es der passenden Schlüssel. Je nach eingesetzter Cipher Suite kommen unterschiedliche Entschlüsselungsmethoden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

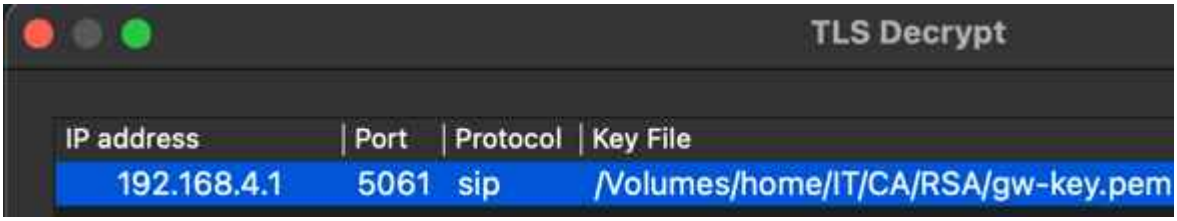
Welche der beiden zur Anwendung kommt, hängt von der Cipher Suite ab: mit Perfect Forward Secrecy (PFS) oder ohne. Falls für die Übertragung keine PFS Cipher Suites vorgesehen sind, kann die Entschlüsselung auf Basis des privaten Schlüssels des Serverauthentifizierungszertifkats stattfinden. In diesem 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"
```

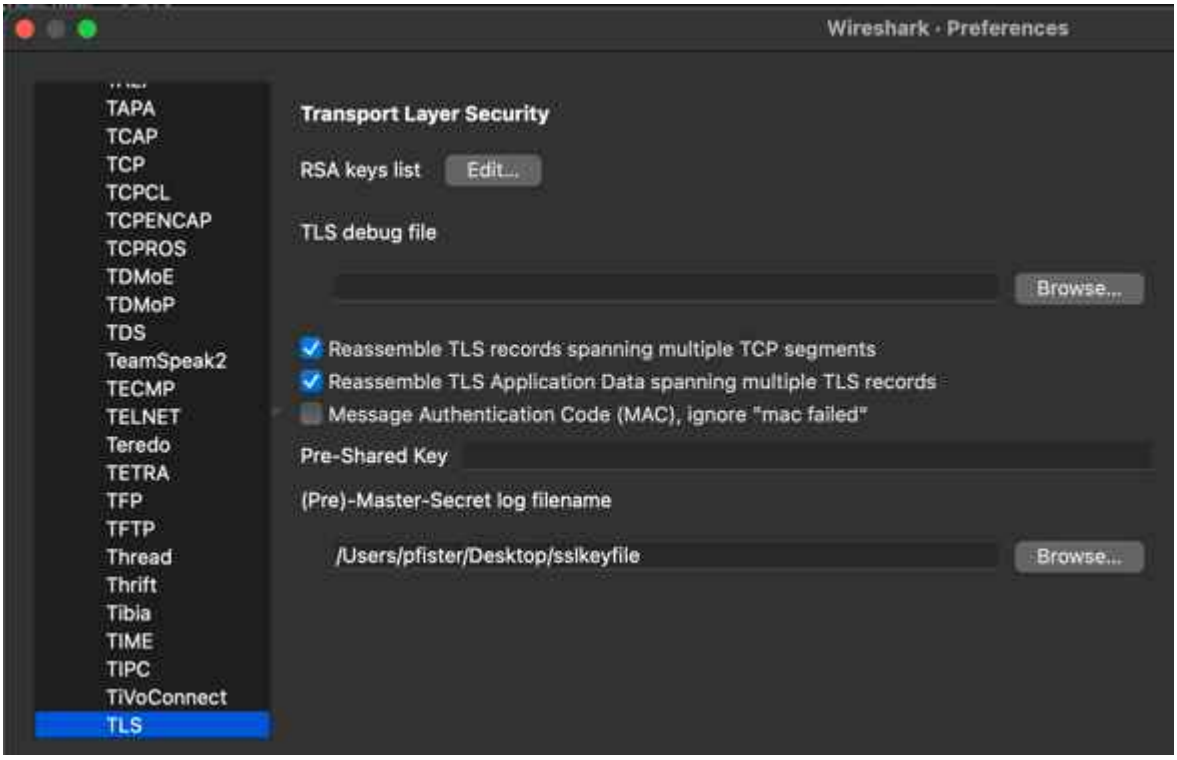
verknüpft man die in Abbildung 1 dargestellten Einstellungen – ähnlich wie mit der GUI-Variante. Das Argument -r siptls.pcapng liest dabei lediglich die PCAPNG-Datei. Das Argument -Y sip setzt einen Display-Filter auf das VoIP-Signalisierungsprotokoll SIP, sodass keine Pakete anderer Protokolle die Ausgabe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

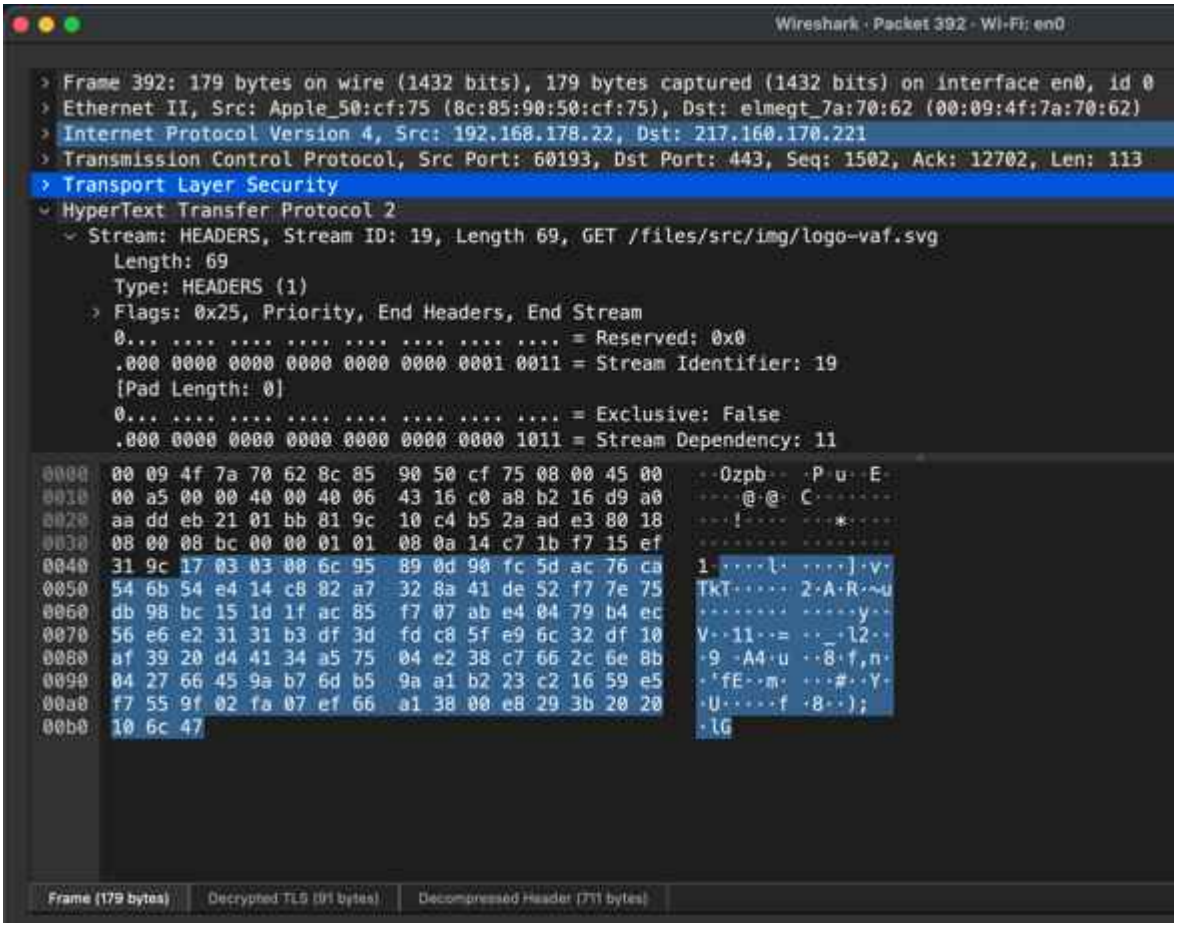
dafür nutzen, den Master Key abzuleiten und damit den Datenverkehr zu entschlüsseln. Im Anschluss an die Konfiguration der Umgebungsvariablen startet man den Mitschnitt in Wireshark und öffnet dann über die Konsole beispielsweise Firefox mittels open /Applications/Firefox.app unter macOS. Nachdem die 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.

Spuren kompromittierter E-Mail-Konten analysieren

Spuren kompromittierter E-Mail-Konten analysieren

Nachdem der erste Teil des Playbooks zu gekaperten E-Mail-Accounts zeigte, wie man die Logs mithilfe von Microsofts zentraler Logfunktion einsammelt, erklärt der zweite Teil, wie man sie auf verdächtige Vorgänge auswertet und welche Rückschlüsse sich daraus ziehen lassen.
Nachdem der erste Teil des Playbooks zu gekaperten E-Mail-Accounts zeigte, wie man die Logs mithilfe von Microsofts zentraler Logfunktion einsammelt, erklärt der zweite Teil, wie man sie auf verdächtige Vorgänge auswertet und welche Rückschlüsse sich daraus ziehen lassen.

- Beim ersten Anzeichen verdächtiger Aktivität rund um E-Mail-Accounts sollte man IT-forensische Untersuchungen anstoßen, um zu verstehen, was genau passiert ist. Ausgangspunkt der Analyse sind die gesammelten Logdaten und Artefakte.
- Aussagekräftig im Hinblick auf Eindringlinge ins Firmennetz sind unter anderem fehlgeschlagene Anmeldevorgänge, eingerichtete Mailweiterleitungen oder

neu vergebene Berechtigungen. Solche Hinweise sollten sorgfältig untersucht werden.
- Die Ursachenforschung und eine Nachbereitung sind das A und O nach der Bewältigung von Sicherheitsvorfällen. Daraus abgeleitete technische Maßnahmen sowie die Sensibilisierung von Mitarbeitenden sollen künftige Angriffe zumindest erschweren.

Die umfassendste Datenquelle zur Analyse von Unregelmäßigkeiten oder Verdachtsmomenten für einen Sicherheitsvorfall bietet Microsofts zentrale Logfunktion Unified Audit Log (UAL). Hier werden Benutzer- und Administratoraktivitäten auch unabhängig vom Einsatz zusätzlicher Produkte wie Microsoft Sentinel oder Microsoft Defender for Identity aufgezeichnet (wie die Logdaten im Detail gesichert werden, beschreibt [1]). Die nachfolgenden Schritte zeigen, wie man bei der Analyse vorgeht und die Logdaten sinnvoll durchsuchen kann.

Schritt 4: Untersuchen der Anmeldeaktivitäten

Jedes Mal, wenn sich ein Benutzer bei seinem Konto anmeldet, wird ein Ereignis im UAL erstellt. Dieses Ereignis enthält wichtige Informationen, etwa die Quell-IP-Adresse, die sich unter anderem für eine geografische Suche verwenden lässt. Die Ergebnisse lassen sich mit den erwarteten geografischen Standorten eines Unternehmens und seiner Nutzer vergleichen. Wenn zum Beispiel ein Unternehmen in Deutschland ansässig ist und keine Niederlassung in Asien hat oder das VPN des Unternehmens nicht zu einer IP-Adresse in Asien auflöst, würde man keine Ereignisse aus Asien erwarten. Daher wären Anmeldungen aus Asien in diesem Fall verdächtig.

Natürlich kann es auch sein, dass ein Mitarbeiter sich im Urlaub in Asien befindet und sein Firmenhandy dabeihat, dennoch erfordern diese Ausreißer Aufmerksamkeit. Verdächtige

Anmeldungen kann man durch die Suche nach bestimmten Schlüsselwörtern im UAL entdecken. Neben der IP-Adresse liefern auch die Uhrzeit sowie Informationen zum verwendeten Gerät (UserAgent: Betriebssystem, Browser et cetera) gute Anhaltspunkte. Ob das verwendete Gerät dem Unternehmen bekannt ist und von der IT verwaltet wird oder nicht, lässt sich ebenfalls den Ereignissen entnehmen. Für die Suche nach verdächtigen Anmeldeereignissen kann man folgende Schlüsselwörter verwenden:

Schlüsselwort	**Bedeutung des Logeintrags**
MailboxLogin	Hinweis auf einen erfolgreichen Log-in-Vorgang
UserLoggedIn	Hinweis auf einen erfolgreichen Log-in-Vorgang
UserLoginFailed	Hinweis auf einen fehlgeschlagenen Log-in-Vorgang
IdsLocked	Hinweis auf einen Brute-Force-Angriff. Der Account wurde gesperrt, da zur viele fehlgeschlagene Anmeldeversuche unternommen wurden.
UserKey=“Not Available“	Hinweis auf einen Brute-Force-Angriff. Die Anmeldung ist fehlgeschlagen, da der Benutzeraccount nicht existiert.

Neben Ereignissen rund um das Log-in können auch Fehlermeldungen zur Multi-Faktor-Authentisierung (MFA) Indikatoren für mögliche schädliche Aktivitäten sein. Ein Angreifer könnte das Passwort eines Anwenders ausgespäht haben, um dann an der MFA-Abfrage zu scheitern. UAL-Einträge mit den folgenden Schlüsselwörtern sollten näher untersucht werden:

Schlüsselwort	Bedeutung des Logeintrags
UserStrongAuthClientAuthNRequired	Der Benutzer wird zur Bestätigung einer MFA-Abfrage aufgefordert.
UserStrongAuthClientAuthNRequiredInterrupt	fehlgeschlagene MFA-Abfrage

Schritt 5: Untersuchen von Weiterleitungsregeln

Nachdem ein Angreifer einen Benutzeraccount kompromittiert hat, erstellt er häufig Weiterleitungsregeln, um eingehende E-Mails an ein externes Postfach zu schicken. Auf diese Weise kann er die Aktivitäten eines Opfers kontinuierlich überwachen, ohne sich aktiv in das Konto einzuloggen. Selbst wenn das Passwort eines kompromittierten Kontos zurückgesetzt wird, kann der Angreifer weiterhin E-Mails mitlesen.

Ebenfalls beliebt ist der Einsatz von Weiterleitungsregeln zum automatisierten Löschen von E-Mails, um Spuren, die auf Unregelmäßigkeiten hinweisen, zu verwischen. Auch können Weiterleitungsregeln dazu dienen, Spuren vor dem Anwender zu verstecken, indem E-Mails automatisch als gelesen markiert und in einen anderen Ordner (zum Beispiel in den Junk- oder den RSS-Ordner) verschoben werden.

Einem Angreifer bieten sich in einer Microsoft-365-Umgebung gleich mehrere Möglichkeiten, E-Mails an ein externes Postfach umzuleiten. Er kann zunächst einmal Inbox-Regeln anlegen, um E-Mails auszuleiten. Verfügt das Konto zudem über administrative Berechtigungen, ist auch eine Ausleitung über die globalen Postfacheinstellungen oder Exchange-Transportregeln möglich.

Aktive Inbox-Regeln lassen sich mit der Exchange-Management-

Shell auffinden, falls sie nicht bereits mittels des im ersten Artikel vorgestellten Tools Hawk extrahiert wurden:

```
Get-InboxRule -Mailbox | ? {$_.forwardto -or $_.forwardasattachmentto -or $_.redirectto}
```

Auch aktive Mailbox-Weiterleitungen kann die Exchange-Management-Shell anzeigen:

```
Get-Mailbox <identity> | Format-List ForwardingSMTPAddress,DeliverToMailboxandForward
```

Der Powershell-Befehl Get-TransportRule liefert eine Übersicht über alle bestehenden Weiterleitungsregeln.

Des Weiteren kann man im UAL potenzielle Angreiferaktivitäten im Zusammenhang mit Weiterleitungsregeln analysieren. Hier lassen sich auch Regeln nachvollziehen, die der Angreifer schon wieder gelöscht hat. Folgende Schlüsselwörter führen zu den relevanten Logeinträgen:

Schlüsselwort	**Bedeutung des Logeintrags**
New-InboxRule	Anlegen einer neuen Weiterleitungsregel (Inbox-Ebene)
New-TransportRule	Anlegen einer neuen Transportregel (Mail Flow Rule)
Set-Mailbox	Änderungen an den Einstellungen einer Mailbox; kann zum Einrichten einer Weiterleitung auf Mailbox-Ebene verwendet werden
Set-InboxRule	Änderung an einer bestehenden Weiterleitungsregel (Inbox-Ebene)
Set-TransportRule	Änderung an einer bestehenden Transportregel (Mail Flow Rule)
DeliverToMailboxAndForward	Hinweis darauf, dass eine E-Mail an eine andere Mailbox weitergeleitet wurde

Schlüsselwort	Bedeutung des Logeintrags
ForwardingSMTPAddress	Hinweis auf eine erfolgte Weiterleitung einer E-Mail
ForwardingAddress	Hinweis auf eine erfolgte Weiterleitung einer E-Mail
SentTo	Hinweis darauf, wohin eine E-Mail gesendet beziehungsweise weitergeleitet wurde
BlindCopyTo	Hinweis darauf, wohin eine E-Mail gesendet beziehungsweise weitergeleitet wurde
ForwardTo	Hinweis darauf, wohin eine E-Mail gesendet beziehungsweise weitergeleitet wurde

Schritt 6: Persistent Access – Hintertüren entdecken

Im nächsten Schritt gilt es zu prüfen, ob der Angreifer Hintertüren eingerichtet hat. Das würde ihm auch im Fall einer Entdeckung noch Zugriff auf die erbeuteten Konten gewähren. Hier gibt es im Wesentlichen drei beliebte Techniken: App-Kennwörter, das Einrichten schädlicher OAuth-Applikationen und die Manipulation von Berechtigungen.

App-Kennwörter dienen eigentlich der Absicherung von Netzwerkprotokollen, die Microsofts „Modern Authentication“ nicht unterstützen. Um die Sicherheit eines Kontos nicht durch die Verwendung des Kennwortes über ein Protokoll, das nicht dem aktuellen Sicherheitsstand entspricht, zu gefährden, bietet Microsoft die Möglichkeit, ein spezifisches Kennwort einzurichten. Es gilt nur für dieses Protokoll.

Wird es kompromittiert, erhält der Angreifer nur Zugriff zu einem einzelnen Protokoll, zum Beispiel IMAP oder POP, nicht aber zum gesamten Nutzerkonto. Doch Angreifer können diese

Funktion auch missbrauchen, damit sie über ein selbst eingerichtetes App-Kennwort auch nach Änderung des Kennworts im Azure AD noch Zugriff auf die Mails eines Nutzers haben und gegebenenfalls auch weiterhin illegitime Mails verschicken können.

Zur Prüfung auf App-Passwörter sollten Administratoren zum einen im Azure AD die für den jeweiligen Benutzeraccount hinterlegten Authentifizierungsmethoden sichten und zum anderen im Kontext des Kontos selbst die Liste der App-Kennwörter abrufen (siehe ix.de/z2y8).

Anwendungen als Hintertür missbrauchen

Auch Enterprise-Applikationen, die sich mittels OAuth authentifizieren, können als Hintertür zu einem kompromittierten Konto genutzt werden. Berechtigt der Angreifer eine von ihm kontrollierte Enterprise-Applikation zum Zugriff auf das übernommene Konto, erlaubt er damit der Applikation, Aktionen im Kontext des Benutzers durchzuführen.

So ist über diese Applikation auch nach Änderung des Kennworts ein Zugriff mit den gewährten Berechtigungen möglich. Um zu prüfen, ob im Rahmen eines Angriffs Enterprise-Applikationen Berechtigungen erhielten – Microsoft spricht in diesem Zusammenhang von „Illicit Consent Attacks“ –, gibt es mehrere Möglichkeiten.

Administratoren können die Berechtigungen über das Azure-Active-Directory-Portal über den Menüpunkt „Nutzer“ und Auswahl des betroffenen Nutzerkontos prüfen. Eine globale Liste zeigt im Azure AD der Unterpunkt Enterprise-Applikationen. Wer lieber mit PowerShell arbeitet, kann das Skript AzureADPSPermissions.ps1 (siehe ix.de/z2y8) verwenden, um sämtliche OAuth-Berechtigungen eines Tenant in eine CSV-Datei zu exportieren und anschließend zu überprüfen.

Das Hinzufügen von Enterprise-Applikationen beziehungsweise

das Erteilen von Berechtigungen für sie im Analysezeitraum wird im UAL erfasst. Das Werkzeug Hawk extrahiert die Artefakte automatisch (Azure_Application_Audit.csv und Consent_Grant.csv).

Eine Variante zum Phishing mittels OAuth-Applikationen ist das sogenannte Device-Code-Phishing, mit dem sich Office-365-Konten übernehmen lassen. Details zu dieser Angriffstechnik sowie Hinweise zur Detektion und Aufklärung finden sich in einem Artikel des Sicherheitsforschers Nestori Syynimaa (siehe ix.de/z2y8).

Schlüsselwort	Bedeutung des Logeintrags
Add 0Auth2PermissionGrant	Einer Enterprise-Applikation wurden Berechtigungen erteilt.
Consent to application	Einer Enterprise-Applikation wurden Berechtigungen durch einen Admin erteilt.
Add app role Assignment grant to use	Ein Benutzer wurde einer Applikation hinzugefügt.

Hat ein Angreifer mehrere Konten eines Unternehmens kompromittiert, kann er sie dazu missbrauchen, Hintertüren einzurichten, indem er den anderen kompromittierten Konten Zugriff auf eine Mailbox gibt. Solange die Verteidiger nicht sämtliche betroffenen Konten identifizieren, behält der Angreifer weiter Zugriff.

Ereignisse im Zusammenhang mit Berechtigungsänderungen lassen sich durch die Suche nach den folgenden Schlüsselwörtern im UAL ausfindig machen:

Schlüsselwort	Bedeutung des Logeintrags
Add-MailboxPermission	Neue Berechtigungen auf ein Postfach wurden vergeben.

Schlüsselwort	Bedeutung des Logeintrags
Add-MailboxFolderPermission	Neue Berechtigungen auf einen Order in einem Postfach wurden vergeben.
Add-RecipientPermission	Hinweis darauf, dass einem Benutzer die „Senden als“-Berechtigung zugewiesen wurde.
Set-MailboxFolderPermission	Bestehende Berechtigungen eines Ordners in einem Postfach wurden geändert.

Hat ein Angreifer sogar ein Konto mit administrativen Berechtigungen gekapert, kann er zudem eigene neue Benutzerkonten anlegen, die dann als Hintertür dienen. Auch das hinterlässt Spuren im UAL.

Schlüsselwort	Bedeutung des Logeintrags
Added user	Ein neuer Benutzer wurde angelegt.

Schritt 7: Datenexfiltration analysieren

Bestätigt es sich, dass jemand Unbefugtes Zugriff auf das Unternehmensnetzwerk hatte, stellt sich in erster Linie die Kernfrage: Worauf hat der Angreifer zugegriffen? Dem zugrunde liegt oft die (späte) Erkenntnis über Art und Umfang der Informationen, die mit einem Benutzerkonto prinzipiell erreichbar wären, verbunden mit dem Wunsch, dieses Worst-Case-Szenario irgendwie einzugrenzen.

Hier zunächst die schlechte Nachricht vorweg: Es ist in der Praxis selten möglich, einen Negativbeweis zu führen, also festzustellen, was die Angreifer nicht mitgenommen haben. Die Aussagekraft der Artefakte ist meist begrenzt, da schlicht nicht alles protokolliert wird. In der Regel muss bei einer gesicherten Kompromittierung eines Kontos unterstellt werden, dass der Angreifer alle erreichbaren Inhalte ausgespäht hat. Das hat erhebliche Konsequenzen beispielsweise für die

datenschutzrechtliche Bewertung eines Vorfalls.

Die gute Nachricht ist, dass auch Microsoft das erkannt hat. Konten, die mit einer E5-Lizenz ausgestattet sind, verfügen über eine „erweiterte Überwachung“. Diese Funktion protokolliert unter anderem Zugriffe auf einzelne E-Mails, was die Chance auf den seltenen Negativbeweis zumindest für die Inhalte des Postfachs deutlich verbessert.

Im UAL finden sich dann Einträge der Art MailItemsAccessed. Diese haben unter anderem ein Attribut MailAccessType, das zwischen Bind und Sync unterscheidet.

Operation	Bedeutung des Logeintrags
MailItemsAccessed	Hinweis auf den erfolgten Zugriff auf Inhalte eines Postfachs

Bind-Einträge werden erzeugt, wenn eine einzelne E-Mail abgerufen wird. Die ID der Nachricht steht dann im Attribut InternetMessageId. Die Protokollierung unterliegt jedoch einer wichtigen Einschränkung: Werden innerhalb von 24 Stunden mehr als 1000 Zugriffe dokumentiert, wird die Protokollierung für Bind-Ereignisse für 24 Stunden ausgesetzt (Throttle).

Zuerst sollte also geprüft werden, ob das UAL Einträge des Typs MailItemsAccessed für die zu untersuchende Mailbox enthält. Anschließend gilt es auszuschließen, dass ein Throttling stattgefunden hat. Dazu schaut man, ob es bei den MailItemsAccessed-Ereignissen welche gibt, die beim Attribut IsThrottled den Wert True vermerkt haben. Im Idealfall gibt es keinen solchen Eintrag.

Welche Sitzung gehört zu wem?

Der nächste Schritt besteht darin, die zum Angreifer gehörenden Sitzungen zu ermitteln. Dafür gleicht man die MailItemsAccessed-Vorgänge im UAL mit den Informationen des Angreifers (verdächtige Log-in-Aktivitäten, IP-Adressen, Zeitstempel, Art des Zugriffs) und den Informationen über den

legitimen Anwender ab. Die Einträge haben mitunter mehrere Session-IDs und IP-Adressen für ein Benutzerkonto. Anhand der in den vorangegangenen Schritten ermittelten Kompromittierungsindikatoren lässt sich feststellen, welche Sitzungen wahrscheinlich legitim oder gültig sind. Einige Sitzungen haben möglicherweise keine Session-ID, weil für die Anmeldung eine alte (Legacy-)Authentifizierung verwendet wurde. Die verdächtigen MailItemsAccessed-Einträge werden dann weiter analysiert.

Sync-Einträge entstehen immer dann, wenn ein E-Mail-Client, beispielsweise Outlook, ein Postfach synchronisiert und dabei Inhalte auf einen lokalen Computer herunterlädt. Hierbei entsteht kein Logeintrag pro Element, sondern pro Ordner des Postfachs. Finden sich im UAL MailItemsAccessed-Einträge mit dem MailAccessType Sync, die dem Angreifer zugeordnet werden, so muss man davon ausgehen, dass alle E-Mails im synchronisierten Ordner kompromittiert wurden.

Zuletzt bleiben die Bind-Vorgänge, die dem Angreifer zugeordnet werden. Diese enthalten eine InternetMessageID. Um damit auf die eigentlichen Nachrichten schließen zu können, ist es notwendig, das Message Trace Log mit den IDs abzugleichen. Leider reicht das Message Trace Log nicht so weit zurück wie die Einträge im UAL, sondern lediglich zehn Tage. Auch lässt sich die InternetMessageID nicht als Suchparameter im Rahmen einer Suche nach Beweismitteln (E-Discovery) verwenden.

Können E-Mails nicht mehr über das Message Trace Log zugeordnet werden, bleibt lediglich der Weg, das Postfach selbst zu exportieren und die E-Mails zu durchsuchen. Die ID ist in den Eigenschaften der E-Mails gespeichert. Der Export des Postfachs lässt sich außerdem über die E-Discovery-Funktion realisieren, die auch bereits gelöschte Elemente berücksichtigt (sofern entsprechende Aufbewahrungsrichtlinien konfiguriert sind und die Elemente noch vorgehalten werden).

Rekonstruieren, was geklaut wurde

Wie beschrieben können E-Mails auch über Weiterleitungsregeln abgegriffen werden. Findet man bei einer Untersuchung solche Regeln, kann sowohl das UAL (siehe Schritt 5) wie auch die Logik der Regeln selbst Aufschluss über die betroffenen Inhalte geben. Neben dem Abgleich der Einträge im UAL mit dem Message Trace Log sollte die Mailbox nach den Parametern der Regel(n) durchsucht werden.

Sofern ein Angreifer Zugang zu einem Konto mit administrativen Berechtigungen und der E-Discovery-Suche hatte, kann er auch auf diesem Weg Inhalte gesucht und exportiert haben. Hinweise darauf lassen sich wieder im UAL finden.

Analog zu den E-Mails sind alle weiteren Inhalte zu berücksichtigen, die mit dem kompromittierten Konto für den Angreifer erreichbar waren. Das beinhaltet sowohl in OneDrive geteilte Dateien wie Teams-Nachrichten und SharePoint-Seiten als auch sämtliche nachgelagerten Applikationen, die Azure AD zur Authentifizierung verwenden. Die Analyse ist allerdings oft sehr individuell und würde den Rahmen dieses Artikels sprengen.

Schritt 8: Remediation

Nachdem die Aktivitäten eines Angreifers nachvollzogen wurden, gilt es, alles rückgängig zu machen, also alle gefundenen Weiterleitungsregeln, Enterprise-Applikationen, App-Kennwörter et cetera zu entfernen und die Kennwörter der betroffenen Konten, falls noch nicht geschehen, zurückzusetzen. Auch sollten alle Analysen und eingeleiteten Maßnahmen dokumentiert und mit den zugehörigen Logdateien aufbewahrt werden.

Zeigte die Untersuchung einen unberechtigten Zugriff auf Postfächer, handelt es sich um einen meldepflichtigen Vorfall gemäß der DSGVO. Dementsprechend ist eine Erklärung an den zuständigen Landesdatenschutzbeauftragten verpflichtend. Dabei

gilt es, die gesetzlichen Fristen zu beachten. Binnen 72 Stunden ab dem Zeitpunkt der Kenntnisnahme muss die Meldung erfolgen. Zu diesem Zeitpunkt ist gegebenenfalls noch nicht das gesamte Ausmaß des Vorfalls bekannt. In diesem Fall sollte die Meldung einfach alle bisher gesicherten Informationen enthalten. Die Meldung sollte durch den benannten Datenschutzbeauftragten des betroffenen Unternehmens erfolgen.

Neben den Datenschutzbehörden müssen gegebenenfalls auch die betroffenen Personen informiert werden. Dies ist dann der Fall, wenn besonders heikle personenbezogene Daten gemäß Art 9 DSGVO – also beispielsweise religiöse oder weltanschauliche Überzeugungen oder Gesundheitsdaten – betroffen sind. In diesem Fall sind die betroffenen Personen direkt zu benachrichtigen. Die Prüfung einer solchen Meldepflicht obliegt dem Datenschutzbeauftragten. Gegebenenfalls sollte bei Verdacht auf einen solchen Fall juristischer Beistand hinzugezogen werden.

Schritt 9: Root Cause Analysis – woran liegts?

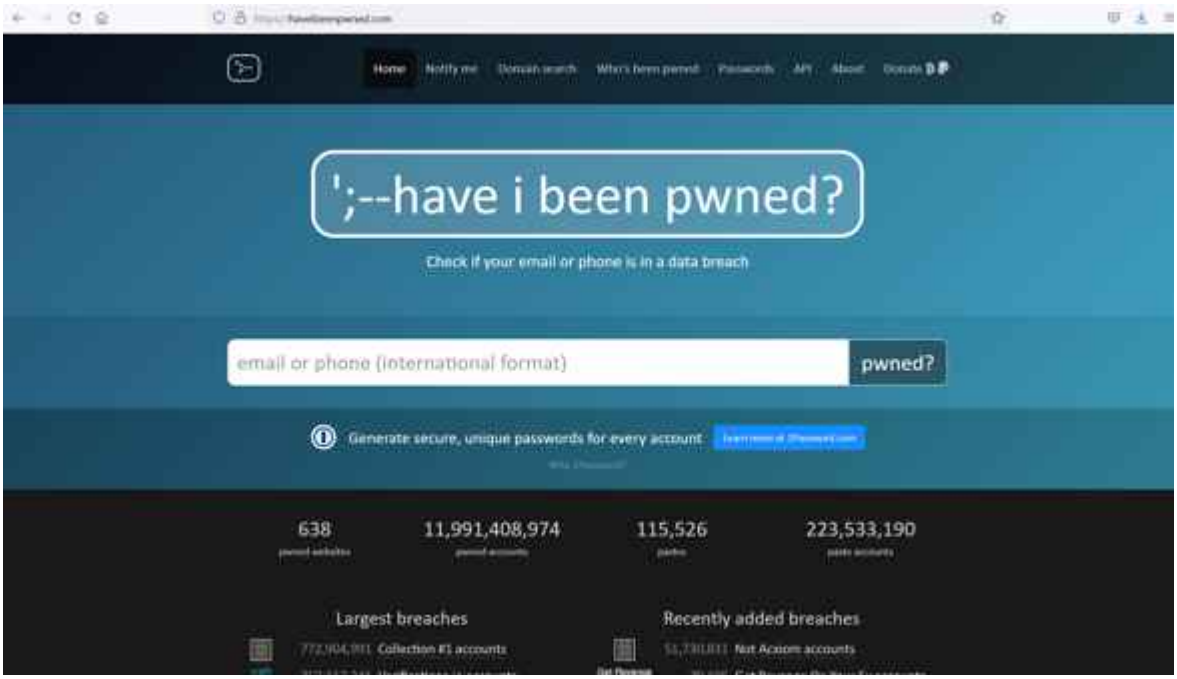
Nachdem aufgeklärt ist, wie ein Angreifer vorgegangen ist und was er genau getan hat, bleibt noch die Frage, wie das passieren konnte. Wie hat er initial Zugang erhalten?

Auch hier ist leider keine pauschale Anleitung möglich, doch die häufigsten Ursachen sind folgende:

- Password Spraying / Brute Force / einfach zu erratende Passwörter: Allen drei Szenarien ist gemeinsam, dass sie in der Regel mit mehrfachem Ausprobieren einhergehen. In den Logs äußert sich dies durch multiple fehlgeschlagene Log-in-Versuche bei einem oder mehreren Konten, ausgehend von derselben IP-Adresse und/oder ähnlichen Parametern wie User-Agent, Protokoll und Zeitpunkt.
- (Spear-)Phishing: Bei einem Phishingangriff erhält das

Opfer eine E-Mail, die einen Link oder einen Anhang enthält, über den die Zugangsdaten abgegriffen werden (funktioniert teilweise auch bei MFA) oder eine Enterprise App via OAuth-Berechtigungsanfrage untergeschoben wird. In dem Fall sind keine gehäuften fehlgeschlagenen Log-in-Versuche zu beobachten. Stattdessen gilt es, die Phishingmail im Postfach oder den aufgerufenen Link ausfindig zu machen.

- Password Re-use / Leaked Credentials: Oft verwenden Anwender ein Passwort für mehrere Dienste und Konten oder recyceln ein privates Passwort für Firmenzwecke. In dem Fall kann es sein, dass das Kennwort bei einem der anderen Dienste ausgespäht wurde und dann für die Anmeldung am Microsoft-365-Account ausprobiert wird. Auch hier ist nicht unbedingt eine gehäufte Anzahl an Fehlversuchen zu beobachten, sofern nicht zusätzlich MFA aktiviert ist. Um der Ursache in dem Fall näherzukommen, empfiehlt es sich, mit dem Benutzer ein offenes Gespräch zu führen oder die Unternehmens-E-Mail des Anwenders bei seriösen Diensten wie haveibeenpwned.com einzugeben (siehe Abbildung).



Ob ein Passwort geleakt wurde, kann man beispielsweise bei Diensten wie „Have I Been Pwned“ herausfinden. Dieser Dienst

des australischen Sicherheitsforschers Troy Hunt hat einen guten Ruf, da er nicht das Passwort selbst, sondern nur den Benutzernamen abfragt.
Nach der erfolgreichen Bewältigung des potenziellen oder realen Sicherheitsvorfalls sollte immer auch geprüft werden, welche Lektionen man daraus lernen kann und welche Maßnahmen zu ergreifen sind, damit ähnliche Vorfälle in Zukunft seltener oder gar nicht mehr vorkommen. Dabei soll es explizit keine Schuldzuweisungen geben, das Stichwort lautet hier vielmehr „Blameless Post Mortem“.

Awareness-Maßnahmen und Schulungen können gängige Betrugsmuster vermitteln und damit die Anfälligkeit der Mitarbeitenden für solche Angriffe verringern. Klar definierte Prozesse zur Veranlassung von Zahlungen helfen außerdem, bestimmte Arten von finanziellem Betrug zu erschweren. Häufig werden aber im Rahmen der Vorfallsbehandlung vor allem technische Gegebenheiten identifiziert, die die Kompromittierung erleichtert oder die Untersuchung des Vorfalls erschwert haben. So ist es hilfreich, die SPF-, DKIM- oder DMARC-Konfiguration (Sender Policy Framework; DomainKeys Identified Mail; Domain-based Message Authentication, Reporting and Conformance) nachzurüsten, falls sie im Vorfeld des Vorfalls noch nicht aktiv war, die Protokollierung lässt sich verbessern, wenn Logs für die Aufklärung des Angriffs fehlten, oder das Installieren von OAuth-Anwendungen kann für Nutzer des Tenants eingeschränkt werden, falls Angreifer solche Anwendungen als Hintertür installiert haben.

Microsoft gibt im Rahmen einer Referenzarchitektur zahlreiche Hinweise für das Absichern von Microsft-365- und Azure-AD-Umgebungen (siehe ix.de/z2y8), die im Nachgang eines Vorfalls (re-)evaluiert werden und bei Bedarf in das Sicherheitskonzept des Unternehmens integriert werden können. Dedizierte Dienste wie Microsoft Defender for Office, Microsoft Defender for Identity oder Microsoft Defender for Cloud Apps können gegen Angriffe schützen oder bei ihrer Entdeckung und Aufbereitung helfen. Allerdings sind sie häufig nur in den teureren

Lizenzen der Microsoft-Produkte enthalten oder müssen sogar separat lizenziert werden. (ur@ix.de)

1. Quellen
2. Jens Lüttgens, Dominik Oepen; E-Mail-Betrug in MS-365-Umgebungen; iX 12/2022, S. 102
3. Vertiefende Microsoft-Artikel, das erwähnte PowerShell-Skript sowie die Microsoft-Referenzarchitektur sind über ix.de/z2y8 zu finden.

Introducing a new phishing technique for compromising Office 365 accounts

The ongoing global phishing campaings againts Microsoft 365 have used various phishing techniques.
Currently attackers are utilising forged login sites and OAuth app consents. In this blog, I'll introduce a new phishing technique based on Azure AD device code authentication flow.
I'll also provide…

Mehr IT-Sicherheit für KMU

Zwei neue Publikationen sollen kleinen und mittleren Unternehmen zu mehr Widerstandsfähigkeit gegen Angriffe verhelfen.

Markt + Trends | IT-Sicherheit

Unternehmen bezahlen Pentester dafür, dass sie versuchen, in ihre Systeme einzubrechen. So dubios das für Außenstehende

klingen mag – Penetrationstests sind seit Jahren ein etabliertes Mittel, die Sicherheit von Systemen und Netzwerken zu überprüfen. Im Folgenden erfahren Sie, wie ein sogenannter Bl…

Kleine und mittlere Unternehmen haben in der Regel nur wenige Ressourcen für IT-Sicherheit. Trotzdem müssen sie den mit der Digitalisierung einhergehenden Bedrohungen begegnen und individuelle Schutzmaßnahmen ergreifen. Für das Entwickeln einer unternehmensweiten Sicherheitsstrategie und das systematische Etablieren von Sicherheit in allen Prozessen ist ein Informationssicherheitssystem (ISMS) hilfreich. Gerade bei KMU ist dieses Werkzeug jedoch noch wenig bekannt oder scheidet aus Mangel an finanziellen und personellen Mitteln aus.

Hier setzt die Broschüre „Kleine und mittlere Unternehmen mit Sicherheit digitalisieren“ des BMWi-geförderten Netzwerks

Mittelstand-Digital an. Mittels einer Umfrage erforschten die Autoren, wie verbreitet oder bekannt ISMS bei kleinen Unternehmen sind, welche Hindernisse oder Motivationen es für eine Einführung oder Zertifizierung gibt und welche Effekte es hat, ein ISMS zu implementieren. Die Broschüre stellt die gängigsten ISMS vor, gibt Handlungsempfehlungen für KMU und nennt weitere kostenlose Angebote – beispielsweise zum Ermitteln des Status quo der IT-Sicherheit, einen Leitfaden zum Implementieren eines ISMS, Erfahrungsberichte von KMU, Checklisten zum IT-Sicherheitsmanagement oder eine interaktive Plattform zur Entwicklung eines eigenen Sicherheitskonzepts.

Einen anderen Ansatz fährt die jüngste Publikation aus dem Bundesamt für Sicherheit in der Informationstechnik (BSI). Dort hat man im Jahr 2020 ein eigenes Referat für kleine und mittlere Unternehmen eingerichtet, die (nach EU-Klassifikation) 99,4 Prozent aller Unternehmen in Deutschland ausmachen. Jenseits von ISO-Normen und IT-Grundschutz-Kompendium vermittelt die Broschüre die wichtigsten Grundlagen der IT-Sicherheit in 14 Fragen. Geschrieben sind sie so, dass auch nicht mit der Technik befasste Geschäftsführer nach der Lektüre wissen, was sie in ihrem Unternehmen umsetzen müssen – oder, wenn das nicht selbst zu stemmen ist, durch einen Dienstleister umsetzen lassen sollten. Beide Broschüren sind über ix.de/zjqs zu finden. (ur@ix.de)