1 WannaCry Ransomware-Angriff (EternalBlue Exploit)

Der WannaCry-Ransomware-Angriff hatte erhebliche Auswirkungen auf Organisationen weltweit. Am 12. Mai 2017 verbreitete sich der WannaCry-Ransomware-Wurm auf über 200.000 Computer in über 150 Ländern. Darunter Systeme der Deutschen Bahn, Krankenhäuser oder große Unternehmen wie FedEx.[1] Bei WannaCry handelt es sich um einen Verschlüsselungs-Wurm, welcher sich über eine Sicherheitslücke im SMBv1-Netzwerkprotokoll in Windows XP und Windows 8 verbreitete. Der genutzte Exploit EternalBlue wurde ursprünglich von der NSA entdeckt und 5 Jahre genutzt, dann von einer geheimen Gruppe namens ShadowBrokers kompromittiert und später veröffentlicht. Die Schwachstelle EternalBlue nutzt einen Fehler in der Implementierung des SMBv1 Protokolls aus. Dadurch ist es entfernten Angreifern möglich beliebigen Code über manipulierte Pakete an einem Zielserver auszuführen, auch bekannt als „Windows SMB Remote Code Execution Vulnerability“. Im ersten Schritt infizierte sich ein System mit dem Wurm per E-Mail-Anhang, danach werden die Daten des Opfers verschlüsselt und nach weiteren potenziellen Opfern im Netzwerk gesucht. Zur Verschlüsselung der Systeme nutzte WannaCry eine Kombination aus dem RSA-2048 und dem AES-Algorithmus. Zur Verschlüsselung der Daten wurde einerseits der RSA-2048 genutzt, wobei für jedes System ein eigenes Schlüsselpaar generiert wurde. Andererseits nutze man den AES-Algorithmus und verschlüsselte den symmetrischen Schlüssel mit dem RSA-2048. Die Kombination der beiden Algorithmen machte WannaCry äußerst effektiv. Nachdem die Systeme verschlüsselt waren, sollten die Opfer 300 US-Dollar in Bitcoin an die Erpresser zahlen, um den Schlüssel zur Entschlüsselung zu erhalten.[2]

Um den Wurm zu stoppen, brachte Windows bereits am gleichen Tag Patches für betroffene Betriebssysteme in den Umlauf. Echte Abhilfe schaffte der Sicherheitsforscher Marcus Hutchins. Er fand eine nicht ungewöhnliche Auffälligkeit in der Ausführung des WannaCry-Wurms, welche von den Entwicklern als Schutz vor Reverse Engineering und Ransomware-Forschern dienen sollte. Der Wurm fragte vor der Ausführung eine nicht existente Website an, um festzustellen, ob er in einer Sandbox-Umgebung ausgeführt wird oder nicht. In Sandbox-Umgebungen laufen Anfragen nach draußen in der Regel über einen virtuellen Proxy, welcher dem System in der Sandbox immer zurückgibt, dass eine Seite erreichbar wäre, obwohl sie es nicht ist. Die Entwickler wussten, dass die Domäne: *iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com* nicht existiert und falls angeblich doch, befinde man sich in einer Sandbox-Umgebung und stellt daher die Ausführung des Programms

[1] [boj17].
[2] [Car20].

ein, um keine Aufmerksamkeit zu erregen. Der Sicherheitsforscher Hutchins registrierte nun allerdings die Domäne und erreichte damit, dass sich die WannaCry Würmer von selbst abschalteten, womit der WannaCry Vorfall gestoppt werden konnte.[3]

Die Folgen dieses Angriffs waren trotzdem fatal, neben dem großen Datenverlust durch die starke Verschlüsselung wurden verschiedene Organisationen und das öffentliche Leben gestört. Darüber hinaus waren die finanziellen Schäden enorm, Schätzungen zufolge beliefen sich die Kosten für Wiederherstellung der Daten und Betriebsunterbrechungen auf mehrere Milliarden US-Dollar. Heute sollte WannaCry keinerlei Auswirkungen mehr auf unsere Systeme haben, die genutzte Schwachstelle EternalBlue wurde bereits in Windows 7 geschlossen und war in Windows 10 noch nie ausnutzbar.[4]

[3] [Clo17].
[4] [boj17].

2 EFAIL Angriff auf E-Mail-Verschlüsselung (OpenPGP & S/MIME)

2018 testeten Forscher der Fachhochschule Münster, des Horst-Görtz-Instituts für IT-Sicherheit der RUB und der belgischen Universität Leuven die Angreifbarkeit der Implementierungen von OpenPGP und S/MIME bei vielen E-Mail-Programmen. Darauffolgend wurden diverse Defizite in der Informationssicherheit aufgedeckt und gefundene Sicherheitslücken in E-Mail-Clients geschlossen. EFAIL-Angriffe, wie sie von den Forschern bezeichnet werden, nutzen Schwachstellen in den OpenPGP und S/MIME-Standards und -Implementierungen, um den Klartext verschlüsselter E-Mails nach der Entschlüsselung durch den Client zu lesen. Diese Angriffe beruhen auf der Möglichkeit des Man-in-the-Middle Angriffs oder kompromittierten E-Mail-Accounts und Mailservern. Der Angreifer muss die Möglichkeit haben die Nachricht zu präparieren oder manipulieren.[5]

Aus technischer Sicht stellen die Forscher zwei verschiedene Angriffstechniken dar, die „Direct Exfiltration“ und „Malleability Gadget Exfiltration Channel“ -Angriffe. Der erste Angriff, nutzt Schwachstellen der Implementierung von Ver- und Entschlüsselung in Mail-Programmen aus. Ein solcher Angriff läuft in der Praxis wie folgt ab: eine E-Mail wird vom Angreifer beispielsweise als Multipart-Mail manipuliert, was bedeutet, er verändert den Inhalt der E-Mail derart, dass diese aus mehreren Body-Teilen besteht. Im ersten Body-Teil wird ein HTML-Image-Tag mit Anführungsstrichen begonnen und nicht beendet. Es folgen zwei weitere Body-Teile. Im zweiten Body-Teil steht der verschlüsselte Text und im dritten Body-Teil wird das HTML-Image-Tag geschlossen. Diese präparierte Mail sendet der Angreifer an den Empfänger, dessen Mail-Client den verschlüsselten zweiten Body-Teil entschlüsselt und alle drei Body-Teile zusammenfasst dargestellt. Der Angreifer musste sich die entschlüsselte Mail nun nur mehr zukommen lassen, dies funktionierte beim Test der Forscher beispielsweise über Manipulationen des User Interfaces bei beispielsweise Mozilla Thunderbird, so dass beim Klick des Empfängers auf die Mail Aktionen im Hintergrund ausgeführt werden und den Klartext an den Angreifer senden. Dieser erste Angriff war nach kurzer Zeit nicht mehr anwendbar, da die Entwickler der betroffenen E-Mail-Programme die Sicherheitslücken schnellstmöglich schlossen und passende Patches herausgaben. Der zweite Angriff von EFAIL baut ebenso auf einen Man-in-the-Middle Angriff auf, allerdings unabhängig vom E-Mail-Client und nutzt hingegen Schwachstellen in der Spezifikation von PGP und S/MIME aus. Dieser Angriff ist für jede E-Mail-Verschlüsselungsart speziell. Im Fall einer S/MIME-Verschlüsselung nutzt man den sogenannten CBC-Gadget-Angriff. Dieser Angriff richtet sich gegen die Cipher Block Chaining Technologie, welche S/MIME nutzt. Die Schwachstelle hierbei ist, dass S/MIME

[5] [Dip18].

verschlüsselte E-Mails in der Regel mit einem String „Content-type: multipart/signed“ beginnen, so dass ein Angreifer mindestens einen vollen Block des Klartextes kennt und diesen als Initialisierungsvektor für den folgenden Angriff nutzen kann. Ziel ist es damit, einen Block zu erstellen, welcher ähnlich zum ersten Angriff einen Image-Tag darstellt und darauffolgend der verschlüsselte Nachrichteninhalt enthält. Dieser Nachrichteninhalt wird vom Mail-Programm entschlüsselt und eine Anfrage des eingefügten Images an den Server im Image-Tag gestellt, in dieser Anfrage ist dann der entschlüsselte Klartext enthalten. Neben den HTML-E-Mails mit Image-Tags können sowohl Stylesheets oder JavaScript als auch das Laden von Zertifikaten und die Anfragen zur Zertifikatsprüfung bei S/MIME für Exfiltration-Angriffe genutzt werden.[6] Dieser zweite Angriff richtet sich gegen die Standards der E-Mail-Verschlüsselungsverfahren, wodurch jeder standardkonforme Mail-Client anfällig ist. Auch das Signieren entsprechender E-Mails ist in dem Fall keine Lösung, da der Angreifer die Signatur aus der Mail entfernen kann. Um diesen Angriffen entgegenzuwirken, müssen entweder die Standards angepasst werden oder Man-in-the-Middle Angriffe durch Sicherung der Nachrichtenauthentizität mit Technologien wie: SPF, DKIM, DMARC oder DANE besser vorgebeugt werden. Als Workaround kann man das Nachladen von Daten ausschalten oder externe Programme zur E-Mail-Entschlüsselung nutzen.

Durch die Offenlegung des Inhalts verschlüsselter E-Mail-Kommunikation ist es Angreifern möglich Informationen zu erhalten, wodurch die größte Folge eines solchen Angriffs der Informationsabfluss an Dritte ist. Neben jenem wird sicherlich die Sicherheit der verschlüsselten Kommunikation in Frage gestellt und dadurch das Vertrauen in die Verschlüsselung verloren, wobei ebenso die Notwendigkeit stets aktueller Software-Patches bestätigt wurde.

[6] [Dam18].

3 DROWN – SSL- und TLS-Schwachstelle

DROWN steht für „**D**ecrypting **R**SA with **O**bsolete and **W**eakend E**n**cryption" und ermöglicht es Angreifern verschlüsselte Verbindung von bis zu 33% der weltweit betriebenen Server zu knacken. Die Angreifer nutzen dazu die Schwächen des längst veralteten SSLv2 Protokolls. Die genutzte Schwachstelle im SSLv2 wurde bereits 1998 gefunden und war bis 2015 in Updates enthalten. Angreifbar waren die Server also nur durch ungenügende Aktualisierung. Ziel des Angriffs ist es über SSL-/TLS-gesicherte Verbindungen übertragene Daten zu entschlüsseln.

Die Attacke beruht grundlegend auf dem Bleichenbacher-Angriff. Dabei handelt es sich um einen Chosen-Ciphertext-Angriff auf PKCS#1 v1.5, dem RSA-Padding-Standard der in SSL und TLS verwendet wird. Dieser Angriff wurde bereits 1998 entwickelt und ermöglicht die Entschlüsselung RSA-Chiffraten. 2016 enthielten allerdings fast alle SSL/TLS-Server bereits passende Gegenmaßnahmen gegen diese Attacke. Die Forscher des DROWN Angriffs entdeckten, dass das SSLv2-Protokoll fatal verwundbar für die Entschlüsselung von RSA-Chiffraten auf Basis des Bleichenbacher-Angriffs ist. Man nutzt SSLv2 zum Brechen von passiv aufgezeichneten RSA-Schlüsselaustauschen zwischen TLS-Servern. In der ersten Entwicklung von DROWN nutzt der Angreifer eine direkte Schwachstelle im SSLv2-Handshake aus, wobei der Server mit einer „ServerVerify" Nachricht direkt nach dem Erhalt des „ClientMasterKey" antwortet. Der Client Master Key ist ein RSA-PKCS#1 v1.5-Chiffrat welches Angreifbar für die Bleichenbacher-Attacke ist. Dabei nutzt man die Tatsache aus, dass RSA-Chiffretexte in PKCS#1 v1.5-konforme Klartexte entschlüsselt werden sollen. Wenn eine derartige Implementierung nun ein RSA-Chiffrat welches in einen ungültigen PKCS#1 v1.5-Klartext resultiert, ist es wahrscheinlich, dass Informationen darüber in Fehlermeldungen durch Schließen der Verbindung preisgegeben werden. Mithilfe der erhaltenen Informationen und einiger Berechnungen kann ein kryptografisches Orakel für den Entschlüsselungsprozess erstellt werden. Um einen solchen Angriff erfolgreich durchzuführen benötigt man ungefähr 1000 aufgezeichnete RSA-Schlüsselaustausche, 40.000 SSLv2-Verbindungen zum Opfer-Server und muss 1.125.899.900.000.000 symmetrische Verschlüsselungsversuche durchführen. Der Angriff funktioniert und wurde von den Entwicklern mithilfe einer AWS EC2 Instanz getestet. Sie konnten ein 2048-Bit-RSA-Chiffrat in 18 Stunden mit einer einzelnen Maschine entschlüsseln.

Es wurde allerdings noch eine spezielle Version des DROWN-Angriffs entwickelt, welche Schwachstellen von OpenSSL ausnutzt und bedeutend effizientere Entschlüsselung ermöglicht. Es erfordert ebenso 1000 aufgezeichnete RSA-Schlüsselaustausche, jedoch nur

weniger als halb so viele Verbindungen zum Opfer-Server und keine aufwändigen Berechnungen. Dadurch ist es möglich diesen Angriff auf handelsüblicher Hardware auf einem Prozessorkern innerhalb einer Minute durchzuführen, dabei begrenzt vor allem die Antwortgeschwindigkeit des Servers die Angriffsdauer. Mit dieser neuen Version ist es möglich Echtzeitangriffe auf laufende TLS-Sitzungen als Man-in-the-Middle durchzuführen. 2016 waren laut Forschungen 26% aller Webserver aufgrund Implementierungsfehlern für derartige Echtzeitangriffe anfällig.[7]

Um dem DROWN-Angriff entgegenzuwirken, muss das SSLv2 Protokoll auf den betroffenen Servern deaktiviert werden oder neuere Versionen von OpenSSL und Microsoft IIS verwendet werden. Außerdem muss sichergestellt sein, dass die privaten Schlüssel des Servers auf keinen anderen Servern oder durch weitere Dienste verwendet werden, welche SSLv2-Verbindungen unterstützen. Große publizierte Folgen durch die DROWN-Angriffe lassen sich nicht finden, allerdings lässt sich annehmen, dass in Unwissenheit der Server-Betreiber potenziell eine Menge an Daten und Konten durch Angreifer kompromittiert werden konnten. Die möglichen Folgen eines solchen Angriffs sind Datenabfluss und damit verbundene Datenschutzverletzungen und hoher finanzieller Aufwand zur Beseitigung des entstandenen Schadens.[8]

[7] [Nim16].
[8] [Ste16].

4 SolarWinds Hack

Der SolarWinds-Hack, der im Dezember 2020 bekannt wurde, war ein massiver Cyberangriff, bei dem das IT-Management-Softwareunternehmen SolarWinds infiltriert wurde. Der Angriff hatte weitreichende Auswirkungen, da die Angreifer es schafften, eine Hintertür in die SolarWinds Orion-Software-Plattform einzuschleusen, die von zahlreichen Regierungsbehörden, Unternehmen und Organisationen weltweit genutzt wird. Die Angreifer infiltrierten das Entwicklungsnetzwerk von SolarWinds und fügten der vertriebenen Software schädlichen Code hinzu, der unbemerkt blieb. Diese manipulierte Version wurde dann als legitimes Softwareupdate an Kunden verteilt. Dadurch bekamen die Angreifer Zugriff auf die Netzwerke und Systeme der Kunden, die die infizierte SolarWinds-Software verwendeten.

Die verwendete Angriffstechnik nennt man „Supply-Chain-Angriff“, dabei ist es das Ziel der Angreifer, möglicherweise weniger gut gesicherte Drittanbieter-Unternehmen zu infiltrieren und dadurch Zugriff auf Netzwerke und Ressourcen größerer Kunden zu erlangen. Bei SolarWinds verschafften sich die Angreifer bereits knapp ein Jahr bevor der Angriff bekannt wurde in den Systemen von SolarWinds, denn in Hackerkreisen war es schon länger bekannt das die IT-Infrastruktur des Unternehmens wohl Schwachstellen hatte. Der Sicherheitsforscher Vinoth Kumar entdeckte einige Zeit vor dem Angriff ein öffentliches GitHub-Repository des Unternehmens, welches FTP-Zugangsdaten für einen Software-Update-Server von SolarWinds im Klartext enthielt. Als Beweis für seinen Fund lud er sogar eine Datei auf den Server hoch. Das benötigte Passwort war lediglich „solarwinds123“. Er warnte das Unternehmen in einer Mail vor dieser Schwachstelle, doch diese fand entweder keine Beachtung oder es war schon zu spät und die Angreifer waren bereits in den Systemen. Mithilfe dieser Schwachstelle wurde die Malware SunSpot im Build-Prozess der SolarWinds Orion-Software eingefügt. SunSpot überwacht die aktiven Prozesse, die an der Erstellung des Orion-Produkts beteiligt sind, und ersetzt Daten in der Quelldatei „SolarWinds.Orion.Core.BusinessLayer.dll“, um den Backdoor-Code von SunBurst in neu gebaute Software-Versionen einzufügen. Der SunBurst-Trojaner hatte die Aufgabe Informationen über infiltrierte Netzwerke zu sammeln und diese per Command-and-Control-Kommunikation an einen steuernden Server zu senden. Durch die damit entstandenen Steuerungsmöglichkeiten wurden im Folgenden weitere Backdoors für den Fernzugriff auf die kompromittierten Systeme geladen. Dies war unter anderem der Dropper Teardrop, welcher spezielle Komponenten aus der Penetration-Testing-Umgebung Cobalt Strike extrahierte und damit gezieltes Ausspähen der Systeme und Kommunikation der Opfer ermöglichte. Die zweite genutzte Backdoor hieß Raindrop, welche ebenso Komponenten aus Cobalt Strike extrahiert, allerdings ermöglicht diese verschärfte Version Hands on Keyboard Angriffe

wodurch es möglich war weitere Infrastruktur der Opfer zu kompromittieren und zu überwachen.[9]

Genau um diese Überwachung der Systeme ihrer Opfer und der Kommunikation ging es den Angreifern auch, sie forderten von keinem Opfer Lösegeld, oder beschädigten die Systeme, es ging jederzeit nur um Informationsdiebstahl. Insgesamt über 18.000 Organisationen, Behörden und Unternehmen hatten die Backdoor-Version der SolarWinds-Software installiert und waren damit potenziell von Datenabfluss betroffen. Darunter FireEye, wo der Angriff das erste Mal bekannt wurde, Microsoft und verschiedene US-Behörden. Als Angreifer vermutet man bis heute eine Russische Hacker Gruppe namens „Cozy Bear“. Die Folgen des Angriffs waren weitreichend, von hohem finanziellem Aufwand, um den durch Datenabfluss entstandenen Schaden zu beseitigen, über gestiegene Aufmerksamkeit auf die Informationssicherheit und Netzwerksicherheit und dem damit verbundenen gesunkenen Vertrauen in SolarWinds und weitere Unternehmen, bis hin zu strengeren gesetzlichen Vorschriften.[10] [11]

[9] [Det21].
[10] [Sah23].
[11] [Sec20].

5 Zerologon-Schwachstelle

Die Zerologon-Schwachstelle, welche im Jahr 2020 bekannt wurde, betraf den Netlogon-Authentifizierungsmechanismus in Windows-Servern und ermöglichte es Angreifern, sich als Domänenadministrator in einem Netzwerk einzuloggen und die vollständige Kontrolle über das Active Directory zu erlangen. Der Angriff nutzte eine Schwäche im kryptografischen Algorithmus des Netlogon-Remote-Protokolls (NRPC) aus, um im Folgenden Anmeldedaten zu manipulieren und sich ohne Kenntnis von Benutzername und Passwort in das System einzuschleusen. Dadurch wurde es einem Angreifer ermöglicht, alle Sicherheitsmaßnahmen zu umgehen und Zugriff auf geschützte Daten und Systeme zu erlangen.[12]

NRPC wird bei der Client- und Server-Domainauthentifizierung verwendet. Dabei findet ein Handshake zwischen Domain-Controller und dem zu authentifizierenden Client statt, wobei zwischen der Verwendung von 2DE und AES als Verschlüsselungsverfahren für die Verbindung unterschieden wird. Microsoft nutzt im NRPC eine spezielle Version des Advanced Encryption Standard, nämlich die AES-CFB8 Verschlüsselung. Die Schwachstelle entsteht, weil ein fixer Initialisierungsvektor, welcher immer 16 Nullbytes hat, verwendet wird. Das spricht gegen die AES-CFB8 Sicherheitsvorschrift, die besagt, dass der Wert zufällig generiert sein muss. Um diese Schwachstelle auszunutzen, muss der Angreifer das Passwort auf 8 Nullen setzen. Eine interessante Eigenschaft, die dadurch entsteht, ist dass in einem von 256 Fällen, in denen der IV bestehend aus 16 Nullen in Kombination mit 8 Bytes folgenden Nullen mittels AES-CFB8 verschlüsselt wird, ein Ergebnis entsteht, das nur aus Nullen besteht. Dieser spezielle Null-Ciphertext ermöglicht also in einem von 256 Fällen eine erfolgreiche Authentifizierung. Dieser Vorgang dauert höchsten 3 Sekunden und bedeutet im Erfolgsfall, dass es dem Angreifer gelungen ist, dem Domänen-Controller eine falsche Identität vorzutäuschen. Dadurch ist es ihm noch nicht möglich sich selbstständig zu authentifizieren oder Änderungen am System vorzunehmen. Um diese Privilegien zu erreichen, wird im Folgenden die RPC-Signierung- und -Versiegelung der Transportverschlüsselung des NRPC deaktiviert. Dafür wird ein Flag in einer Nachricht an den Server gesetzt, wodurch die folgende Kommunikation im Klartext stattfindet. Nun ist es möglich eine NetServerPasswortSet2-Anfrage an den Domänen-Controller zu senden und dabei den Anfrageinhalt so zu verändern, dass das Passwort des angefragten Kontos entfernt oder auf einen leeren Wert gesetzt wird. War dieser Schritt erfolgreich, kann sich der Angreifer nun über den normalen Prozessweg mit

[12] [Bun20].

den entsprechenden Privilegien anmelden und möglicherweise Schaden an den Systemen anrichten oder Informationen abgreifen.[13]

Es war schnell unterschiedlicher Exploit-Code öffentlich verfügbar, welcher im Bezug auf den möglichen Schaden vergleichsweise einfach zu verwenden war. Microsoft veröffentlichte daher im August 2020 einen Patch für alle Windows-Domänen-Controller, welcher diese Sicherheitslücke schließt, und empfahl diesen schnellstmöglich zu installieren. Durch die Zerologon-Schwachstelle wurde das Bewusstsein für eine sichere Authentifizierung und Netzwerksicherheit gestärkt. Publizierte Folgen für betroffene Unternehmen existieren nur wenige, weshalb sich die globalen Folgen nur schwer abschätzen lassen. Informationsabfluss und Manipulationen an Systemen sind allerdings für viele Betroffene nicht ausschließbar.

[13] [Tre23].

Literaturverzeichnis

[boj17] boj/dpa/AFP/Reuters, „"WannaCry"-Attacke - Fakten zum globalen Cyberangriff“, SPIEGEL Netzwelt, 2017. https://www.spiegel.de/netzwelt/web/wannacry-attacke-fakten-zum-globalen-cyber-angriff-a-1147523.html Abruf: 23.06.2023

[Bun20] Bundesamt für Sicherheit in der Informationstechnik, „Kritische Schwachstelle im Windows Netlogon Remote Protocol (ZEROLOGON)“, 2020. https://www.bsi.bund.de/DE/Service-Navi/Presse/Alle-Meldungen-News/Meldungen/Zerologon_200924.html Abruf: 2023.07.06

[Car20] Carly Burdova, „Was ist EternalBlue und warum ist der Exploit MS17-010 immer noch relevant?“, 2020. https://www.avast.com/de-de/c-eternalblue Abruf: 2023.06.28

[Clo17] CloudFlare, Inc., „Was war der WannaCry-Ransomware-Angriff?“, 2017. https://www.cloudflare.com/de-de/learning/security/ransomware/wannacry-ransomware/ Abruf: 2023.06.27

[Dam18] Damian Poddebniak, Christian Dresen, Jens Müller, Fabian Ising, Sebastian Schinzel, Simon Friedberger, Juraj Somorovsky, and Jörg Schwenk, „EFAIL describes vulnerabilities in the end-to-end encryption technologies OpenPGP and S/MIME that leak the plaintext of encrypted emails“, 2018. https://efail.de/ Abruf: 2023.06.26

[Det21] Detlef Weidenhammer, „Angriff auf die Supply Chain – SolarWinds“, 2021. https://www.all-about-security.de/angriff-auf-die-supply-chain-solarwinds/ Abruf: 2023.07.06

[Dip18] Dipl.-Inform. Carsten Eilers, „EFAIL: Angriff auf verschlüsselte Mails“, entwickler.de, 2018

[Nim16] Nimrod Aviram, Sebastian Schinzel, Juraj Somorovsky, „DROWN: Breaking TLS using SSLv2“, 2016. https://drownattack.com/drown-attack-paper.pdf Abruf: 2023.07.06

[Sah23] Saheed Oladimeji, Sean Michael Kerner, „SolarWinds hack explained: Everything you need to know“, 2023. https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know Abruf: 2023.07.06

[Sec20] Security Lab Hornetsecurity, „SolarWinds SUNBURST backdoor assessment“, 2020. https://www.hornetsecurity.com/en/threat-research/solarwinds-sunburst-backdoor-assessment/

[Ste16] Stephan Augsten, „DROWN attackiert TLS über SSL-v2-Lücke“, 2016. https://www.security-insider.de/drown-attackiert-tls-ueber-ssl-v2-luecke-a-523741/ Abruf: 2023.07.06

[Tre23] Trend Micro Incorporated, „Was ist Zerologon?“, 2023. https://www.trendmicro.com/de_de/what-is/zerologon.html Abruf: 2023.07.06