Ostbayerische Technische Hochschule Regensburg
Fakultät Elektro- und Informationstechnik

Projektarbeit 1

Analyse der Mechanismen von Advanced Persistent Threats sowie Recherche und Zusammenfassung aktueller Forschungsansätze zur Angriffserkennung

Vorgelegt von: Andreas Zinkl

Studiengang: Applied Research in Engineering Sciences
Betreuer: Prof. Dr. Christoph Skornia
Abgabedatum: 18.09.2018

Zusammenfassung

Diese Projektarbeit wird zu Beginn des Forschungsprojektes DARCNET durchgeführt. DARCNET beschäftigt sich mit der Detektion und Analyse von Cyberangriffen auf Rechnernetze mittels Event-Logs unter der Anwendung von Techniken des maschinellen Lernens. Speziell die Erkennung von Advanced Persistent Threats (APT) liegt im Fokus des Forschungsprojektes. Das Ziel ist es, APT erkennen zu können, welche kaum vom regulären Nutzerverhalten abweichen. Hierzu wird in dieser Projektarbeit zunächst der aktuelle Stand der Forschung im Gebiet der Angriffserkennung zusammengefasst. Dabei wird auf zwei konkrete und bereits existierende Konzepte für eine modulare Software zur Angriffserkennung eingegangen. Des Weiteren werden im Rahmen dieser Arbeit die wesentlichen Mechanismen von APTs untersucht. Diese Mechanismen sind die laterale Ausbreitung, die Fernsteuerung (engl. „Command and Control“ (C2)) sowie die Exfiltration von Daten. Die Resultate dieser Recherche dienen als Basis zur Entwicklung von schwachen Klassifikatoren für die Erkennung von APTs mit Hilfe von künstlicher Intelligenz.

Inhaltsverzeichnis

1 Aktueller Stand der Forschung

In diesem Kapitel werden zwei aktuelle Forschungsansätze zur Erkennung von Angriffen auf Rechnernetze beschrieben. Diese verzeichnen in ihren Konzepten bereits erste Erfolge einer Angriffserkennung anhand der Analyse von Log-Daten und des Netzwerkverkehrs. Dabei werden in den Konzepten maschinelle Lern- sowie statistische Detektionsmethoden angewandt.

1.1 Das Konzept einer verteilten Framework-Architektur zur APT-Detektion

Das Konzept „DFA-AD: A distributed framework architecture for the detection of advanced persistent threats“ beschreibt eine modulare Herangehensweise in der Entwicklung eines Frameworks für eine in Echtzeit durchführbare APT-Detektion [Sha+17]. Im Folgenden werden nun das Verfahren sowie die erreichten Ergebnisse beschrieben und ein Fazit hinsichtlich der Verwendung innerhalb des Forschungsprojektes DARCNET erläutert.

1.1.1 Das Verfahren

Das Framework DFA-AD basiert auf der Echtzeitanalyse des Netzwerkverkehrs und dient dem Ziel einer schnellen Erkennung von APTs [Sha+17]. Dabei wird ein Ensemble von Klassifikatoren zur Erkennung eines Angriffes verwendet, indem die Resultate der einzelnen Klassifikatoren bei der Analyse des Netzwerkverkehrs betrachtet werden. Im Fokus des Konzeptes steht dabei ein modularer Aufbau des Frameworks und die Integration des Frameworks innerhalb eines „Trusted Platform Modules“ (TPM) (siehe Abbildung 1).

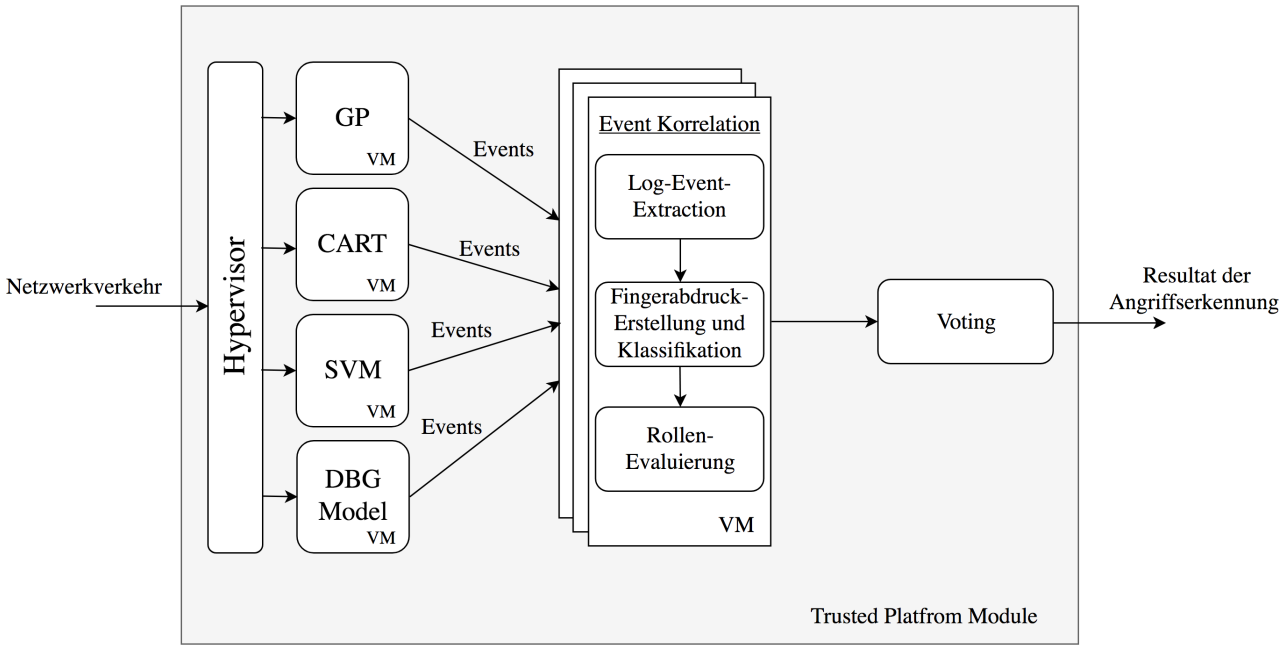

Abbildung 1: *Schema des Framework-Konzepts DFA-AD nach* SHARMA ET AL. *[Sha+17]*

Das in Abbildung 1 dargestellte Framework beschreibt die drei wesentlichen Phasen des Verfahrens. Dieses besteht aus einer zu Beginn durchgeführten Analyse und Klassifizierung des Netzwerkverkehrs, der anschließenden Korrelation der Ergebnisse aus der Klassifizierung und einem abschließenden Voting zur Entscheidung über das Melden eines Angriffes.

Phase 1 - Datenanalyse und Klassifizierung [Sha+17, S. 601 ff]: Innerhalb der ersten Phase analysieren und klassifizieren die einzelnen Klassifikatoren den überwachten Datenverkehr. Jeder Klassifikator befindet sich dabei in einer separaten virtuellen Maschine (VM) und agiert getrennt von den anderen Klassifikatoren. Dabei ist das Ziel eine Detektion von verwendeten Angriffstechniken und außergewöhnlichem Datenverkehr, sowie eine Ermittlung von durchgeführten Events des Angriffes. Die ermittelten Events entsprechen den Resultaten der Phase 1 und den Eingangsdaten zu Phase 2. Ein Event entspricht dabei einem auffälligen Paket im Datenverkehr im Bezug auf einen speziellen Service. Die verwendeten Klassifikatoren innerhalb des Frameworks DFA-AD sind dabei:

- Support Vector Machine (SVM)
- Klassifizierungs- und Regressionsbäume
  (engl. „Classification and Regression Trees“ (CART))
- Genetische Programmierung (GP)
- Dynamisches Bayesian-Spiel Modell
  (engl. „Dynamic Bayesian Game Model“ (DBG-Model))

Die Klassifikatoren werden mit Hilfe von überwachtem Lernen (engl. „supervised learning“) trainiert. Dies wird durch die Aufteilung der Trainingsdaten in zwei Klassen ermöglicht. Diese unterteilen den Datenverkehr zum einen in die Klasse des unauffälligen Datenverkehrs und zum anderen in die Klasse des auffälligen APT-Datenverkehrs. Die Zuordnung zur jeweiligen Klasse erfolgt zunächst anhand der Bewertung eines von Nascimento definierten Anomalie-Scores [Gus10, S. 2]. Wird anhand des Anomalie-Scores ein möglicher APT-Datenverkehr erkannt, so bewertet dies zunächst ein Experte und kennzeichnet den Datensatz entsprechend. Dies soll einen eindeutigen Datensatz mit APT-Datenverkehr ermöglichen. Abschließend werden nun die zwei Klassen für das überwachte Training der Klassifikatoren verwendet.

Phase 2 - Korrelation der Ergebnisse [Sha+17, S. 603 ff]: In der zweiten Phase, der Korrelation der Ergebnisse, werden die Resultate der Klassifikatoren analysiert. Diese entsprechen dabei den entdeckten Techniken und Events eines möglichen Angriffes. Anhand dieser Resultate werden nun alle n Events in einer $n : n$ Verknüpfung korreliert und mögliche Verbindungen zwischen Events errechnet. Die ermittelten Korrelationen werden dann in die letzte Phase des Votings übergeben.

Phase 3 - Voting [Sha+17, S. 605 ff]: Die abschließende Phase des Verfahrens ist das Voting zur Entscheidungsfindung. Dies bedeutet, dass die aus Phase 2 ermittelten Korrelationen aller Klassifizierer zusammengefasst werden, um anhand der Kombination der Klassifikatoren einen Angriff zu erkennen. Dabei beinhaltet das Voting den Schwellwert (engl. „Threshold“) der die Grenze zur Entscheidung ob ein Angriff vorliegt definiert. Wird vom Voting ein Angriff detektiert, so meldet das Framework den Angriff und die ermittelten Ergebnisse an den Administrator.

1.1.2 Ergebnisse und Fazit

Die Evaluation der Erkennungsraten von APTs zeigen sehr gute Erfolge von einer gesamten Trennschärfe mit bis zu 98.5%. Gegenüber steht eine geringe False-Positive-Rate (FPR) von maximal 0.03% für fälschlicherweise als Angriff klassifizierte Informationen.

Im Hinblick auf die weitere Arbeit im Rahmen des Forschungsprojektes DARCNET, dient das Konzept des DFA-AD als gute Grundlage für den Bereich der Analyse von Log-Informationen. Die Kombination der eingesetzten Klassifikatoren ist aufgrund der sehr guten Ergebnisse auch für DARCNET interessant und wird in der Entwicklung von schwachen Klassifikatoren ebenso in Betracht gezogen. Durch die zusätzlichen Log-Informationen ist es in DARCNET möglich, anhand des DFA-AD-Konzeptes eine sehr gute Trennschärfe für die APT-Erkennung zu erreichen. Jedoch werden im Konzept keine genaueren Informationen zu dem verwendeten Threshold genannt, was eine Integration des Verfahrens in DARCNET zunächst erschwert. Eine detaillierte Auflistung der verwendeten Merkmale im Bereich des maschinellen Lernens ist ebenso nicht vorhanden.

1.2 Das Konzept eines Frameworks für eine schrittweise APT-Detektion

Mit der Arbeit „Towards a Framework to Detect Multi-Stage Advanced Persistent Threats Attacks“ beschreiben BHATT, YANO und GUSTAVSSON ein Konzept zur schrittweisen Erkennung von APTs [BYG14]. Das Framework besteht aus mehreren Komponenten von unterschiedlichen Sicherheitssystemen. Dabei fließen im Rahmen des Konzeptes von BHATT, YANO und GUSTAVSSON nicht nur die Verhaltensmuster von APTs in die Detektion mit ein, sondern auch die Erkenntnisse aus den Phasen eines möglichen Einbruchs in das System.

1.2.1 Das Verfahren

Das Konzept des Frameworks besteht zunächst aus drei Komponenten. Diese Komponenten sind ein sogenanntes „Multi-Stage Attack Model“, eine „Layered Security“ Architektur sowie das „Security Event Collection and Analysis“ System [BYG14, S. 2]. Ziel dieser Komponenten ist es, dem Angreifer den Einbruch in das System zu erschweren. Dies soll ermöglichen, den Angreifer zu einer intensiveren Interaktion mit dem System zu zwingen. Die intensivere Interaktion führt innerhalb des Systems zu einer größeren Menge an Log-Informationen für die Analyse und Erkennung von Angriffen. Außerdem ist mit Hilfe dieser Komponenten eine Rekonstruktion des Angriffes möglich.

Multi-Stage Attack Model: Eine für das „Multi-Stage Attack Model“ wichtige Anforderung ist zunächst die Struktur des Systems. Hierbei ist es notwendig, eine Vielzahl an Sicherheitsschichten zu verwenden. Dies bedeutet zunächst, dass ein Prozess bzw. eine Anwendung im System nur über eine innere Sicherheitsschicht erreicht werden kann. Diese innere Sicherheitsschicht ist wiederum nur über eine äußere Schicht erreichbar. Somit entstehen für den Angreifer eine Vielzahl an Schichten, welche er durchdringen muss, um in das System zu gelangen.

Die Analyse im „Multi-Stage Attack Model“ basiert auf der von Hutchins, Cloppert und Amin definierten Intrusion-Kill-Chain (IKC) [HCA11]. Jede Schicht des Modells enthält dabei Sensoren, die anhand definierter Richtlinien einzelne Phasen einer IKC erkennen sollen. Dabei wird für jede Schicht ein Verteidigungsplan erstellt. Dieser Plan repräsentiert die IKC und die darin definierten Phasen.

Layered Security Architecture: Die „Layered Security Architecture“ definiert die drei wesentlichen Anforderungen an das vielschichtige System. Dabei ist das Ziel dieser Architektur, dem Angreifer das Eindringen zu erschweren und ihn in einen größeren Ressourcen- sowie Zeitaufwand zu verstricken. Dies erhöht im System die Wahrscheinlichkeit, dass der Angreifer gegen eine, auf der IKC basierende, Richtlinie in einer der Sicherheitsschichten verstößt.

Eine wesentliche Anforderung für eine sichere innere Schicht ist, dass der Zugang zu Anwendungen und Prozessen ausschließlich über eine äußere Schicht erfolgt. Dies kann rekursiv erweitert werden, um die Anzahl der Schichten zu erhöhen. Außerdem ist es wichtig, dass die einzelnen Schichten keine gleichen Schwachstellen aufweisen. Somit muss der Angreifer in jeder einzelnen Schicht erneut die IKC durchlaufen und zunächst Informationen über die Schicht sammeln, bevor er diese durchdringen kann. Somit können vom Angreifer erlangte Informationen nicht weiter verwendet werden. Diese zweite Anforderung führt zu dem bereits genannten Mehrwert in der Analyse, durch das Erzwingen einer Vielzahl von Log- und Event-Daten während eines Angriffes.

Security Event Collection and Analysis System: Die für eine Erkennung eines Angriffes wesentliche Komponente stellt das „Security Event Collection and Analysis“ System dar. Dieses ist wiederum in fünf Module aufgeteilt:

- Logging Module
- Log-Management Module
- Intelligence Module
- Malware Analysis Module
- Control Module

In jeder Schicht des Systems sind hierfür Sensoren integriert. Diese dienen, wie auch der Agent im Rahmen des Forschungsprojektes Darcnet, zur Datensammlung von Event- und Log-

Daten. Die Sammlung und Speicherung der Daten wird dabei mittels zweier Log-Module umgesetzt. Zudem werden in diesen Modulen die Log-Informationen anhand der von Kruegel, Valeur und Vigna vorgeschlagenen Alert-Normalisierung aufbereitet [KVV05, S. 36 ff]. Diese Normalisierung enthält dabei alle notwendigen Informationen zur Identifizierung des Angriffes. Dies sind Informationen über die mögliche Art des Angriffes, den Sensor welcher den Angriff überwacht hat, die Zeit und das Datum der Information, die Quell-IP-Adresse, die Ziel-IP-Adresse (inkl. Port und Bezeichnung der Anwendung) sowie Informationen zu Nutzdaten-Attributen [KVV05; BYG14].

Das „Intelligence Module“ wird zur Analyse der Daten verwendet. Hier werden maschinelle Lernalgorithmen angewandt und die gesammelten Event- und Log-Daten korreliert. Wird nun vom System ein möglicher Angriff anhand der gesammelten Log-Daten erkannt, so wird die aktuelle Phase des Angriffes der definierten IKC zugeordnet. Anhand dieser Information ist dem System nun bekannt, welche Phasen bereits durchlaufen worden sind. Diese Phasen können nun durch die gesammelten Informationen rekonstruiert und der Angriff vollständig und sicher identifiziert werden. Die Rekonstruktion ist in diesem Framework jedoch bislang nur teilweise automatisch möglich. Hierbei geben Bhatt, Yano und Gustavsson den Hinweis auf Methoden zur automatischen Rekonstruktion mit Hilfe eines Hidden Makrov Modells [BYG14].

1.2.2 Ergebnisse und Fazit

Eine Analyse des Frameworks anhand eines realen Szenarios zeigt bereits sehr gute Ergebnisse. Das von Bhatt, Yano und Gustavsson durchgeführte Szenario eines Angriffes auf eine Universität, anhand eines „Zero-Day Exploits“ im installierten PDF-Reader, ermöglicht die Detektion und Rekonstruktion eines Angriffes innerhalb von weniger als acht Minuten. Dabei werden im genannten Szenario vom System 69.969.233 Log-Datensätze ausgewertet [BYG14, S. 5 f].

Das von Bhatt, Yano und Gustavsson umgesetzte Konzept zur schrittweisen Erkennung bietet viele grundlegende Informationen zur APT-Detektion. Dabei können die sehr guten Ergebnisse, durch die zusätzlichen Informationen aus der Phase des Systemeinbruchs, ein sehr guter Ansatz für die Erkennung von schwer detektierbaren APT sein. Dieses Konzept dient somit als sehr gute Ergänzung zum Forschungsprojekt Darcnet.

2 Mechanismen von Advanced Persistent Threats

Der Hauptfokus dieser Projektarbeit liegt in der Analyse der drei grundlegenden Mechanismen von Advanced Persistent Threats (APTs). Dies sind die Fernsteuerung von Schadsoftware, die laterale Ausbreitung sowie die Datenexfiltration. Dabei werden die untersuchten Mechanismen zunächst erläutert und anschließend Anwendungsbeispiele beschrieben.

2.1 Möglichkeiten zur Fernsteuerung von APTs

Die Fernsteuerung von Schadsoftware wird in der Literatur sowie auch in dieser Arbeit als „Command and Control“ (C2) bezeichnet. Das Rüstungs- und Forschungsunternehmen QinetiQ erläutert in ihrem Bericht zu C2 generelle Funktionsweisen [Qin17]. In diesem Kapitel wird dabei vor allem auf unauffällige Methoden des C2 eingegangen, welche sich kaum vom Nutzerverhalten des Opfers unterscheiden. Da die Techniken aus dem Bereich der Fernsteuerung und Datenexfiltration (Kapitel 2.3) gleichermaßen verwendet werden können, werden in diesem Kapitel nicht alle möglichen und unauffälligen Techniken erläutert.

2.1.1 Definition der Methodiken Push und Pull

In QinetiQs Bericht über die Funktionsweise des C2 wird die Übermittlung von Kommandos zur Fernsteuerung zunächst in zwei Arten unterteilt [Qin17, S. 7]. Dies ist zum einen die „Push“ und zum anderen die „Pull“ Methodik.

Bei der Push-Methode handelt es sich um eine direkte Kommunikation zwischen Angreifer und Schadsoftware. Diese kann dabei ebenso über einen dritten Knotenpunkt, einer sogenannten Steuereinheit verlaufen, um den Angreifer zu anonymisieren. Eine Rückmeldung zur Ausführung der Kommandos erfolgt von der Schadsoftware dabei ebenso auf direktem Weg zurück zum Angreifer.

Die Pull-Methode hingegen entspricht einer passiven Steuerung der Schadsoftware über eine zentrale Steuereinheit. Diese kann zum Beispiel ein dritter Webservice sein, auf dem die Kommandos für die Schadsoftware zur Verfügung gestellt werden. Die Schadsoftware versucht dabei durch den Aufruf der Steuereinheit (zum Beispiel einer Webseite oder eines Servers) neue Kommandos abzufragen. Das Antworten, sowie auch das Steuern, erfolgt dabei über den Zwischenspeicher in der zentralen Steuereinheit, in welcher der Angreifer die Antworten der Schadsoftware abschließend auch abfragen kann.

2.1.2 HTTP und HTTPS

Eine der am häufigsten verwendeten Werkzeuge zur Fernsteuerung ist das „Hypertext Transfer Protocol“ (HTTP), sowie das „Hypertext Transfer Protocol Secure“ (HTTPS) [Qin17]. Dies liegt darin begründet, dass ein Zugang zum Internet meist verfügbar ist und damit die Ports 80 (HTTP) und 443 (HTTPS) auch verwendet werden können. Dabei kann eine unauffällige Steuerung der Schadsoftware ausschließlich über die Pull-Methode durchgeführt werden, die außerdem dem regulären Netzwerkverkehr ähnelt.

Die für den Bereich C2 angewandten Methodiken ähneln den Methoden aus dem Bereich der Datenexfiltration (siehe Kapitel 2.3). Warmer sowie Wang, Zhen, Niu et al. beschreiben in ihren Arbeiten, dass das HTTP das am häufigsten genutzte Protokoll zur Kontrolle von Schadsoftware ist [War11; Wan+16]. Der Grund diesbezüglich liegt, wie bereits beschrieben, in der unauffälligen Kommunikation durch den ähnlichen Datenverkehr des Opfers beim Besuchen von Internetseiten. Dabei beschreiben Wang, Zhen, Niu et al. vor allem die Nutzung bekannter Webseiten wie zum Beispiel Amazon und CNN als zentrale Kommunikationsschnittstelle [Wan+16, S. 2 f]. Die Betreiber bieten auf ihren Webseiten die Möglichkeit zur Nutzung von Kommentarfunktionen. Diese ermöglichen es Angreifern innerhalb eines unauffälligen Kommentars Kommandos für die Schadsoftware auf gezielt gewählten Seiten zu platzieren.

Einfache Tools wie *curl*, *wget* sowie der Browser selbst können dabei als Kommunikationsmittel dienen. Durch spezielle Frameworks (*selenium*[1] für Chrome und *geckodriver*[2] für Firefox) ist es möglich, den Browser selbst innerhalb der Schadsoftware unauffällig anzusteuern und zu verwenden.

2.1.3 DNS

Zur Steuerung der Schadsoftware wird neben HTTP ebenso das „Domain Name System“ (DNS) Protokoll verwendet. Die Verwendung von DNS im Bereich des C2 haben hierbei Xu, Butler, Saha und Yao untersucht [Xu+13]. Bei der Verwendung von DNS zur Kommunikation mit der Schadsoftware, werden im Allgemeinen Daten und Kommandos über die DNS-Records versandt. Dies ermöglicht ein Überwinden der Firewall durch die reguläre Verwendung eines Kommunikationsprotokolls. Ein weiterer Vorteil für einen Angreifer stellt dabei die dezentrale Verfügbarkeit des Servers dar. Somit besteht im Gegensatz zu HTTP eine hohe Fehlerresistenz im Bereich des C2.

Xu, Butler, Saha und Yao unterscheiden jedoch zunächst zwei Methoden zur Anwendung von DNS als C2-Werkzeug [Xu+13, S. 144 f]. Zum einen wird dabei das DNS-Tunneling und zum anderen das DNS-Codeword genannt. Das DNS-Tunneling stellt dabei eine bidirektionale Kommunikation zwischen Server und Client her. Die Methodik des DNS-Codeword hingegen beschreibt eine unidirektionale Verbindung vom Server zum Client. Dies eignet sich somit auch besonders zum Versenden von Kommandos an die Schadsoftware. Im Allgemeinen wird in diesem Verfahren eine reguläre Namensauflösung eines Domainnamens auf einem DNS-Server (engl. „name lookup“) durchgeführt. Dieser „name lookup“ wird dabei vom DNS-Server des Angreifers in verwertbare Daten umgewandelt. Die übermittelten Daten sind hierfür im Domainnamen im URL-konformen Zeichensatz Base32 oder Base64 kodiert (siehe Abbildung 2).

Der wesentliche Unterschied bezüglich der Erkennbarkeit der zwei genannten Methoden ist die Frequenz der Kommunikation. Beim DNS-Tunneling müssen Nachrichten in häufigen, periodi-

[1] Ein auf GitHub (*https://github.com/SeleniumHQ/selenium/wiki/ChromeDriver*) als Open Source frei verfügbares Framework für den Google Chrome Browser.
[2] Ein auf GitHub (*https://github.com/mozilla/geckodriver*) frei verfügbares Framework für den Browser Mozilla Firefox.

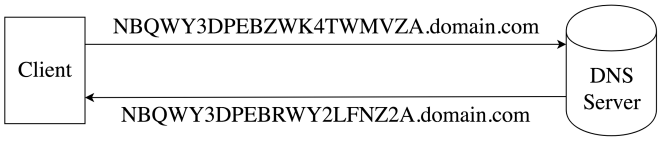

Abbildung 2: *Schematische Darstellung einer Kommunikation über das DNS-Protokoll nach* Xu, Butler, Saha *und* Yao *[Xu+13, S. 145]. Der Client sendet dem DNS-Server eine Nachricht „hallo server“, welche in base32 formatiert wurde. Der Server antwortet durch das Auflösen der Anfrage mit Hilfe des CNAME mit der Nachricht „hallo client“, welche ebenso in base32 kodiert ist.*

schen Zyklen vom Client an den Server gesendet werden. Dies ermöglicht eine relativ einfache Detektion, da diese periodischen Zyklen zwischen zwei Kommunikationsstellen mit der Analyse des Netzwerkverkehrs entdeckt werden können. Das DNS-Codewort hingegen bietet eine für den Angreifer unauffällige Möglichkeit, Kommandos an die Schadsoftware bzw. den Client zu senden. Eine detaillierte Anwendung und Auswertung der Möglichkeiten zur Nutzung des DNS-Protokolls ist in der Arbeit von Xu, Butler, Saha und Yao zu finden [Xu+13].

2.1.4 Echo-Protokoll

Das Echo-Protokoll dient als ein Diagnosewerkzeug für den Netzwerkverkehr. Dabei verwendet Echo das „Internet Control Message Protocol“ (ICMP), welches keine speziellen Ports benötigt. Das wohl bekannteste Werkzeug zur Netzwerkdiagnose, welches das Echo-Protokoll nutzt, ist die Software *ping*. Durch die Nutzung von ICMP-Echo-Requests in der Ransomware Loki, wurde ICMP und damit auch das Echo-Protokoll für die Verwendung zur Datenexfiltration und zur Fernsteuerung von Schadsoftware sehr bekannt [Maz18].

Die Besonderheit des Echo-Protokolls für die Verwendung zur Steuerung der Schadsoftware liegt im Anwendungsbereich von ICMP begründet. Durch die Verwendung von ICMP in der Netzwerkkommunikation zur Status- und Fehlerübermittlung wird ICMP häufig von Firewalls nicht blockiert. Firewalls selbst analysieren dabei, sofern nicht explizit konfiguriert, nur den ICMP-Header und darin die Informationen über Typ, Sender und Empfänger. Die im Datenfeld optional hinzugefügten Informationen werden dabei nicht überprüft. Mazerik beschreibt dabei die Möglichkeit zur Nutzung eines ICMP-Tunnels zur Herstellung einer verschlüsselten Verbindung zwischen Opfer und Angreifer [Maz18]. Dabei kann mit Hilfe von speziellen Tools ein direkter Konsolenzugriff auf den Computer des Opfers ermöglicht werden [Zel15]. Zudem bietet das ICMP die Möglichkeit einer Erschließung der Netzwerkinfrastruktur, beispielsweise durch ICMP-Traceroute, sowie einer Analyse des Ziel-Betriebssystems [Maz18]. Die Nutzung des Echo-Protokolls ermöglicht dem Angreifer somit eine unauffällige Kommunikation, bei der ein verursachter Netzwerkverkehr des Angreifers dem des Opfers gleicht.

2.2 Laterale Ausbreitung von APTs

Eine Studie aus dem Jahr 2014 mit Bezug auf APTs beschreibt für die laterale Ausbreitung im Wesentlichen drei Schritte [CDH14]. Dies sind eine interne Reconnaissance zur Informationsgewinnung, das Infizieren weiterer Systeme inklusive dem Aneignen neuer Rechte, sowie das Identifizieren und Sammeln von wertvollen digitalen Informationen. In diesem Kapitel wird nun auf diese drei Schritte eingegangen. Dabei werden wesentliche Methoden der lateralen Ausbreitung anhand möglicher Praxisbeispiele genannt.

2.2.1 Vorbereitung der lateralen Ausbreitung

Für eine erfolgreiche und unauffällige Ausbreitung der Schadsoftware ist für den Angreifer zunächst die Kenntnis über die Infrastruktur sowie das System des Opfers von großer Wichtigkeit. Aus diesem Grund wird im Rahmen der lateralen Ausbreitung zunächst mit einer internen Reconnaissance zur Informationsgewinnung begonnen.

Das für die Informationsgewinnung am häufigsten verwendete Tool ist dabei *nmap* [Smo16]. *nmap* bietet für den Angreifer eine große Bandbreite an Angriffsmöglichkeiten zur Ermittlung von aktiven Services, dem installierten Betriebssystem und offener Ports für die Kommunikation. Hierbei bietet *nmap* Funktionalitäten die eine Minimierung des zu erzeugenden Netzwerkverkehrs ermöglichen. Ein weiteres Hauptziel der Informationsgewinnung besteht darin, Authentifizierungsdaten, wie z.B. Nutzernamen und Passwörter, sowie private Schlüssel zu stehlen. Mit diesen Informationen kann zum einen der Datenverkehr verschleiert und zum anderen auch das Social Engineering[3] angewandt werden, um weitere Systeme unauffällig und effizient infizieren zu können.

Einblick in nmap: *nmap* bietet eine hohe Bandbreite an Funktionalitäten. Eine gezielte Suche nach speziellen Ports über den sogenannten „Stealth Scan“ (Option *-sS*) erzeugt im Netzwerk einen minimalen Datenverkehr und ermöglicht dem Angreifer eine unauffällige Analyse potentieller Schwachstellen [Smo16; Nma]. Dabei werden im „Stealth Scan“ vom Angreifer keine SYN-Pakete nach dem Erhalt der ACK-Pakete zugesandt, was den Netzwerkverkehr minimiert, jedoch auch in der Diagnose des Netzwerkverkehrs auf die Anwendung des Tools hinweisen kann [Nma]. Zudem kann ein automatischer Port-Scan durch *nmap* mit Hilfe der Option *-sn* deaktiviert werden. Somit können auch nur speziell ausgewählte Ports über die Option *-p* festgelegt und analysiert werden. Dies hilft dem Angreifer zusätzlich den Netzwerkverkehr zu verringern. Neben den genannten Funktionen sind in *nmap* eine Vielzahl weiterer vorgefertigter Skripte zur Netzwerkanalyse und Durchführung von Angriffen vorhanden, welche auch im Online-Benutzerhandbuch (*https://nmap.org/nsedoc/scripts/*) eingesehen werden können. Eine Analyse des Netzwerkverkehrs auf ein typisches Verhalten von spezifischen *nmap* Funktionen (z.B. das untypische Weglassen der SYN-Pakete im Netzwerkverkehr) kann nun dabei helfen, erste Erkenntnisse über einen aktuell laufenden Angriff zu ermitteln.

[3] Social Engineering bezeichnet dabei das Ausnutzen menschlicher Schwachstellen durch gezielte psychische Manipulation des Opfers. [Eck]

2.2.2 Powershell, Windows Sysinternals, WinRM und WMI

Im Rahmen der lateralen Ausbreitung werden die Tools Powershell, DOS-Kommandozeile, Secure Shell sowie Bash als Hauptwerkzeug verwendet [Smo16]. Dabei nennt Smokescreen.io im Windows-Bereich vor allem Powershell als ein sehr gut geeignetes Tool für eine unauffällige und effiziente Ausbreitung.

Powershell: Da Powershell bereits eine große Bandbreite an Werkzeugen mit sich bringt und im System unauffälliger ist als die bereits bekannte Malware, wird von Angreifern immer häufiger auf die Powershell zurückgegriffen [Smo16, S. 5]. Bereits bekannte Angriffe ermöglichen es mit Hilfe von Powershell In-Memory-Authentifizierungsdaten zu ermitteln [Smo16; GiM17, S. 5]. Powershell wird dabei nicht nur explizit alleine eingesetzt, sondern auch mit weiteren Werkzeugen, wie dem Windows Remote Management (WinRM) und dem Windows Management Instrumentation (WMI) kombiniert.

WinRM und WMI: Smokescreen.io und Lambert beschreiben die Nutzung von Powershell in Verbindung mit dem, ab Windows Vista integrierten, Windows Remote Management (WinRM) sowie dem Windows Management Instrumentation (WMI) [Smo16; Lam17]. Das WinRM ist ein Protokoll zur Fernwartung von Computern mit einem Windows Betriebssystem mit Hilfe des WMI Tools. Das WMI kann hierbei zur automatisierten Administration von Windows-PCs genutzt werden. Mit Hilfe des WMI kann dabei auf nahezu alle Einstellungen des Windows-PC zugegriffen werden. Zudem ermöglicht das WMI das Starten von Remote Prozessen. Diese Prozesse, welche über das WMI gestartet wurden, verursachen keine Speicherung von Daten und dienen somit sehr gut zur lateralen Ausbreitung und der Generierung von Backdoors [Smo16; Gra15]. Gräber stellt dabei in seiner Arbeit Möglichkeiten zur Ausnutzung des WMI anhand einer Vielzahl an Angriffsszenarien dar und erläutert ebenso Methoden zum Schutz vor selbigen [Gra15].

Windows Sysinternals: Weitere Möglichkeiten für eine unauffällige laterale Ausbreitung der Schadsoftware bieten die von Sysinternals entwickelten PsTools. Dabei nennt auch Smokescreen.io das Werkzeug *PsExec* als ein mögliches Mittel zur Fernsteuerung, aber auch zur Infizierung weiterer Systeme [Smo16]. *PsExec* erlaubt es, ohne eine Installation auf dem Zielgerät Prozesse auszuführen [Rus04]. Eine besondere Eigenschaft dieses Tools ist es, die Ein- und Ausgabe der Konsole an eine externe Adresse umzuleiten. Ein aktivierter „File- & Print-Sharing“ Service ist die einzige Voraussetzung für die Nutzung der PsTools im Bereich der lateralen Ausbreitung auf Netzwerkebene.

Die PsTools, und somit auch das Tool *PsExec*, bieten einem Angreifer interessante Funktionen zur Steuerung und Nutzung der Schadsoftware über die Konsole. Die Nutzung führt hierbei zu keinerlei Meldungen an den Nutzer und aufgrund der häufigen Verwendung der Sysinternals für die Administration eines Rechnernetzwerks, bleibt der Datenverkehr zudem unauffällig.

Außerdem können die PsTools besonders einfach in Skripte eingebunden und verwendet werden [Smo16]. Das folgende Beispiel-Kommando führt den Befehl *ipconfig* auf dem Zielrechner des Opfers über eine Remote-Verbindung aus [Rus04]:

```
1 psexec \\<REMOTE-HOST> ipconfig
```

Lambert beschreibt in diesem Zusammenhang die Anwendung von *PsExec* unter Ausnutzung des „Admin-Shares“ (siehe Kapitel 2.2.3). Zusammenfassend bieten die PsTools somit sehr gute Voraussetzungen für eine automatisierte Ausbreitung im System. Um jedoch eine Verwendung der Tools durch einen Angreifer zu vermeiden bzw. zu erkennen, empfehlen Ussath, Jaeger, Cheng et al. ein detailliertes Log-Management der ausgeführten Prozesse für eine eindeutige und schnelle Analyse [Uss+16, S. 6].

2.2.3 File Sharing Service

Einer der bekanntesten Werkzeuge zum Verteilen von Dateien ist das „File Transfer Protocol“ (FTP). Die laterale Ausbreitung über einen FTP kann durch das Unterbinden der Nutzung eines FTP Services oder einer detaillierten Überwachung des Datenverkehrs vermieden werden. Der FTP Service wird in dieser Arbeit aufgrund der einfachen und offensichtlichen Detektierbarkeit nicht weiter behandelt.

Neben dem FTP Service gibt es jedoch weitere, vor allem unauffälligere Möglichkeiten, Daten innerhalb des Netzwerks zur Verfügung zu stellen oder zu verteilen. Smokescreen.io beschreibt in diesem Zusammenhang die Ausnutzung der „Admin-Shares“ *ADMIN$* und *C$* unter Windows [Smo16]. In Kombination mit den bereits besprochenen PsTools, WinRM und WMI aus Kapitel 2.2.2, können somit ohne Verwendung eines FTP Services Daten innerhalb des Netzwerks ausgetauscht und verbreitet werden. Zudem verfügt ein Angreifer durch die „Admin-Shares“ über einen vollständigen Ordnerzugriff auf das Verzeichnis *%SYSTEMROOT%* und besitzt zudem Schreib- sowie Leserechte auf dem gesamten Laufwerk. Smokescreen.io weist darauf hin, dass auf diese Ausnutzung bei fast allen Angriffen zurückgegriffen wird [Smo16].

Ein Beispiel für die Nutzung der „Admin-Shares“ beschreibt Lambert in seinem Beitrag zur Verwendung der Windows Sysinternals [Lam17]. In seiner Arbeit verschafft sich Lambert, mit Hilfe der freigegebenen „File- & Print-Shares“ und dem „Server Message Block“ (SMB) Protokoll, Zugriff auf die Powershell-Konsole des Opfers. Die Ein- sowie Ausgaben der Powershell werden dabei direkt an den Angreifer weitergeleitet. Bereits mit wenigen Änderungen in der Registry konnte Lambert das von Windows integrierte „User Account Control“ (UAC) deaktivieren. Das UAC kontrolliert die seit Windows Vista bekannte Meldung zur Eingabe eines Passwortes bzw. der Bestätigung des Nutzers zum Erlangen und Ausführen von Anwendungen als Administrator. Durch die Deaktivierung des UAC besitzt der Angreifer auf dem Zielgerät umgehend einen vollständigen Administrator-Zugriff.

2.2.4 Distributed Component Object Model (DCOM)

Neben den üblichen Methoden zur Verwendung von Schwachstellen und speziellen Tools für die laterale Ausbreitung, wird von Nelson eine neue Methode unter der Verwendung des Windows-Standardprotokolls „Distributed Component Object Model“ (DCOM) beschrieben [Nel17b; Nel17a]. Das DCOM wurde von Microsoft als Erweiterung des „Component Object Model“ (COM) entwickelt, um die Verwendung des COM-Objektes über das Netzwerk zu ermöglichen [Mic17; Mic18]. Das COM selbst soll der Kommunikation zwischen Prozessen dienen und eine Wiederverwendung gleicher Programmteile ermöglichen. Dabei handelt es sich bei der von Nelson untersuchten Methode eher um eine konzeptionelle Schwachstelle des DCOM und nicht etwa um einen Fehler im Werkzeug selbst. Dies bedeutet, dass vom Angreifer keinerlei zusätzliche Eingriffe in die Funktionalität des Protokolls zur Ausbreitung im Netzwerk notwendig sind.

Tsukerman beschreibt in seiner Zusammenfassung unterschiedliche Möglichkeiten zur Nutzung der von Nelson entdeckten Möglichkeit der Ausbreitung mit Hilfe des DCOM [Tsu18; Nel17b; Nel17a]. Dabei nennt Tsukerman unter anderem neue Methoden die auf Microsoft Office Produkte, beispielsweise Microsoft Outlook, zurückgreifen und damit eine unauffällige Ausbreitung ermöglichen [Tsu18].

Für die Nutzung des DCOM oder COM zur lateralen Ausbreitung werden einzig folgende wichtige Informationen aus der Konfiguration eines DCOM- bzw. COM-Objektes benötigt [Tsu18]:

- **CSLID:** CSLID steht für den „Class Identifier“ und entspricht einer eindeutigen ID für ein COM Objekt.
- **ProgID:** ProgID steht für den „Programmatic Identifier“ und kann als Alternative zum CSLID verwendet werden. Im Gegensatz zum CSLID ist jedoch nicht garantiert, dass der ProgID im System eindeutig ist.
- **AppID:** AppID steht für den „Application Identifier“ und entspricht der Konfiguration einer, zu einer Anwendung gehörenden, Gruppe von COM-Objekten. Der AppID beschreibt dabei die Rollen- und Rechteverteilung für die COM-Objekte.

Mit Hilfe dieser Informationen ist es dem Angreifer nun möglich, auf weiteren Zielrechnern Prozesse über das Netzwerk zu starten und die Schadsoftware so zu verbreiten. Dabei erhalten die Prozesse unter der Angabe des CSLID und des AppID, die auf dem jeweilgen Zielsystem hinterlegten Rechte des ausnutzenden DCOM- bzw. COM-Objektes. Eine detaillierte Anwendung des DCOM für die Ausbreitung wird in den Arbeiten von Nelson und Tsukerman beschrieben [Nel17b; Nel17a; Tsu18].

2.3 Methoden zur Datenexfiltration

Eines der Hauptziele eines jeden Angriffes ist der Diebstahl von Informationen. Dies können Informationen über das Opfer selbst oder aber auch Informationen zur Infrastruktur des Opfers sein. Dafür kann der Angreifer unterschiedliche Kommunikationskanäle (engl. „Channels“) nutzen. Eine Analyse der möglichen Channels durch Rashid et al. beschreibt eine Unterteilung (siehe Abbildung 3) in offensichtliche Kanäle (engl. „Overt Channels“) sowie verdeckte Kanäle (engl. „Covert Channels“) [Ras+13, S. 15 ff].

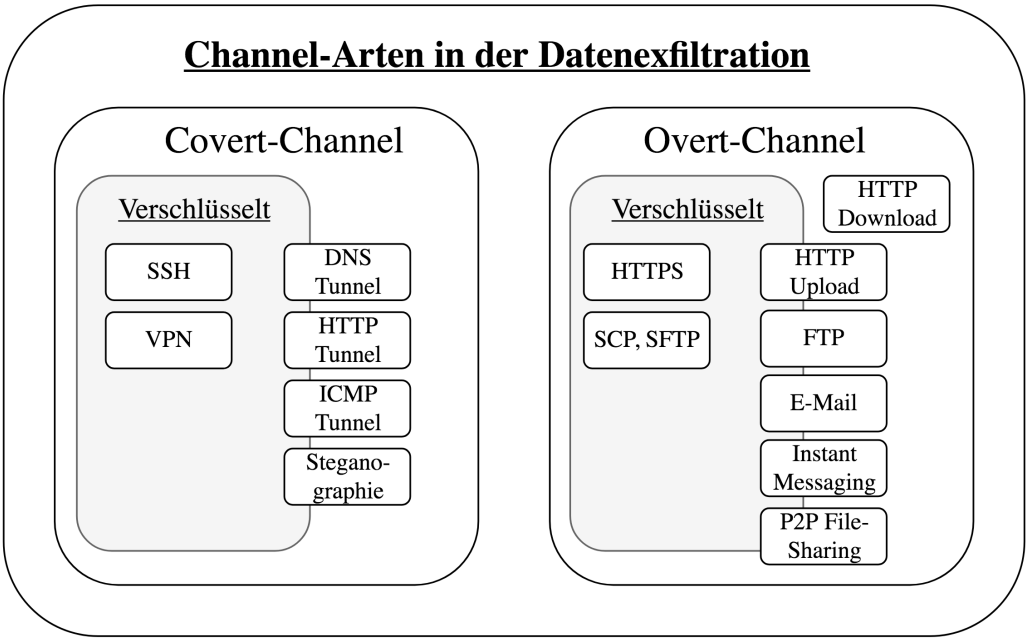

Abbildung 3: *Kategorisierung von Extraktionsmethoden zu „Overt“ oder „Covert Channels“ nach dem Labor des* Security Lancaster *[Ras+13]*

Overt und Covert Channels: Overt Channels bezeichnen Kommunikationswege, die auch von einem im System berechtigten Benutzer, zum gewollten Austausch von Daten verwendet werden können. Dies können zum einen HTTP-Requests beim Surfen des Nutzers, aber zum anderen auch versendete E-Mails sein. Im Gegensatz zu den Overt Channels soll der Datenaustausch bei Covert Channels über unübliche und verdeckte Wege, durch eine zusätzliche Verschlüsselung der zu exfiltrierenden Daten zur Verschleierung des Datenverkehrs erfolgen. Dabei werden beispielsweise Kommunikationswege über einen Secure-Shell (SSH) Tunnel verwendet, welcher nicht dem offensichtlichen Nutzerverhalten entspricht. Hierzu werden neben den verschlüsselten Verbindungen ebenso die Daten selbst verschlüsselt. Dies erfolgt zum einen durch eine Komprimierung der Daten und zum anderen durch die Anwendung von Steganografie-Methodiken. Damit werden ausgehende Verbindungen und Daten für eine detaillierte Analyse der Inhalte einer Kommunikation unkenntlich gemacht.

In-band und out-of-band: Im Bereich der Datenexfiltration wird durch das Security Lancaster der Universität von Lancaster zudem eine Unterscheidung bei der Vorgehensweise hinsichtlich der Channel-Nutzung durchgeführt [Ras+13, S. 7]. Dabei werden Angriffe, die für die Fernsteuerung und die Datenexfiltration unterschiedliche Endpunkte auf Seiten des Angreifers nutzen, mit der Bezeichnung „out-of-band“ beschrieben. Methoden zur Datenexfiltration, welche für die Steuerung und Exfiltration den selben C2-Server als Endpunkt verwenden, werden als „in-band“ Exfiltration bezeichnet. Die Methode der „out-of-band“Exfiltration bedeutet hierbei für den Angreifer einen größeren Aufwand, durch eine zusätzliche Zwischenstelle in der Kommunikation. Jedoch bietet diese laut dem Security Lancaster für den Angreifer die Möglichkeit, die C2-Channels zu verschleiern und zu schützen, falls der Channel zur Datenexfiltration entdeckt würde.

Ein Angriff und eine zugehörige Datenextraktion muss nicht immer über illegale Software bzw. Schadsoftware erfolgen. Trendmicro [Tre13] beschreibt in diesem Zusammenhang die Möglichkeit der Nutzung von installierten Email-Programmen wie Microsoft Outlook. Dies wurde bereits anhand des erfolgreich durchgeführten Angriffes auf die deutsche Bundesregierung gezeigt [Süd15]. Somit ist stets eine klare Trennung in der Detektion anhand der ausgeführten Software zu beachten.

Wie bereits zu Beginn beschrieben, ist die Datenexfiltration eines der Hauptziele eines APT. Die Sammlung von Informationen über das Opfer gilt dabei als Hauptaugenmerk. Dabei werden nach einer Studie von McAfee [McA15] vor allem PDF-Dateien, Microsoft Office Dokumente und XML-Dateien, sowie Bilder und Videos gestohlen. Die nach McAfee [McA15] häufigsten Ziele eines Datendiebstahls sind persönliche Informationen zur Identifikation und Gesundheit des Opfers. Anschließend folgen bereits Informationen zu geistigem Eigentum und Kreditkarteninformationen.

2.3.1 HTTP und HTTPS

Webprotokolle und damit im Detail das „Hypertext Transfer Protocol“ (HTTP) sowie das „Hypertext Transfer Protocol Secure“ (HTTPS), sind nach einer Studie von McAfee [McA15, S. 6] die am häufigsten verwendeten Werkzeuge zur Exfiltration von Daten. Dabei gibt es, wie in Abbildung 3 ersichtlich, Möglichkeiten Overt und Covert Channels zu verwenden.

Verwendung von HTTP-Requests: Unter der Verwendung von HTTP-POST-Requests gibt es unterschiedliche Möglichkeiten im Bereich der Datenexfiltration. Dabei können die Daten innerhalb des HTTP-Headers aber auch im HTTP-Body übermittelt werden. Ebenso können Dateien über einen HTTP-Upload zum Zielserver transferiert werden.

Zur Exfiltration der Daten werden diese zunächst aufgeteilt und in das URL-konforme Format Base64 kodiert [AA17]. Anschließend werden die Daten meist verschlüsselt und mit Hilfe eines

HTTP-POST an den Angreifer gesendet. Hierfür können auch unauffällige Felder im HTTP-Header für den Transport verwendet werden. Ein Beispiel für solch ein unauffälliges Feld im Header ist das Feld „Cookie“. Felder wie der „User-Agent“, „Content-MD5“ und „Warning“ würden sich für die Exfiltration ebenso anbieten. Diese haben jedoch einen bekannten Aufbau, welcher anhand einer Analyse des HTTP-Requests überprüft und dadurch ein Angriff erkannt werden kann. In einem HTTP-POST-Request kann jedoch auch der HTTP-Body als Transportmittel verwendet werden. Mit Hilfe einer Verschlüsselung der Daten kann ein Angreifer den Inhalt der Nachricht verschleiern, was eine Analyse des HTTP-Bodys erschwert. Durch das HTTPS-Protokoll wird zudem der gesamte Verkehr verschlüsselt, was ebenso eine generelle Analyse der Inhalte eines HTTP-POST-Requests erschwert.

Für den Vorgang des Versendens der Daten kann vom Angreifer eine „in-band“ und „out-of-band“ Kommunikation angewandt werden. In-band Methoden beschreiben dabei eine direkte Kommunikation zwischen Opfer und Angreifer. Dabei werden die Informationen direkt zum C2-Server des Angreifers über einen HTTP-POST-Request gesendet. Bei der Anwendung der out-of-band Methode hingegen werden die Informationen an einen unabhängigen Server gesendet und dort zwischengespeichert. Der Angreifer kann nun die zwischengespeicherten Daten, ohne eine direkte Verbindung zum Opfer aufzubauen, vom Server abgreifen [Ant11; Ras+13]. Ein Beispiel zur Durchführung einer Exfiltration mit Hilfe eines HTTP-POST-Requests wird hierbei durch ein Mitglied des InfoSec Institut mit dem Pseudonym *c0d3inj3cT* beschrieben [Ins13]. Auch Azeria [AA17] erläutert ein Beispiel für eine Datenexfiltration mit Hilfe von HTTP-Requests.

Die perfekte Exfiltration nach Klein und Kotler [KK16]: Klein und Kotler beschreiben in ihrer Arbeit eine perfekte Exfiltration von Daten unter der ausschließlichen Verwendung von HTTP und HTTPS [KK16]. Dabei ist es möglich sensible Daten aus einem Unternehmensnetzwerk zu exfiltrieren, ohne Aufmerksamkeit zu erzeugen. Die Arbeit von Klein und Kotler beschäftigt sich hierbei vor allem mit geringen Datenmengen, wie z.B. Authentifizierungsdaten, privaten kryptographischen Schlüsseln oder weiteren sensiblen Informationen.

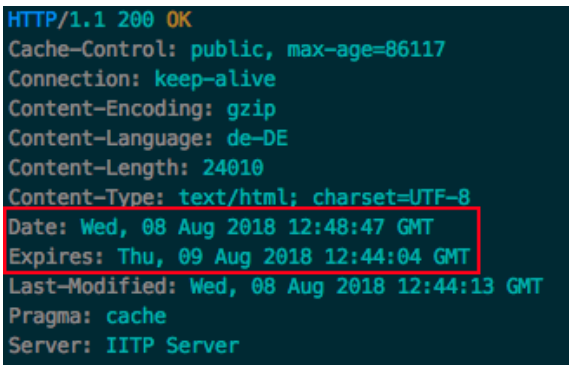

Abbildung 4: *Ausschnitt eines HTTP-Request-Headers aus einer Antwort eines HTTP-POST-Requests*

In der Arbeit von Klein und Kotler wird auf Protokollebene eine Statusänderung des HTTP-

Headers als Werkzeug zur Datenexfiltration verwendet [KK16, S. 8 ff]. Jede Webseite speichert in regelmäßigen Abständen beim Seitenaufruf den aktuellen Status der Seite im Cache des Servers ab. Nach Klein und Kotler sollte hierzu meist eine bekannte eCommerce-Webseite für die Exfiltration gewählt werden. Die Information über die Lebensdauer des Caches und den Zeitpunkt der Sicherung (dem letzten Aufruf) kann dabei meist aus der Antwort eines HTTP-Request-Headers ausgelesen werden (siehe Abbildung 4).

Eine wichtige Anforderung für die Exfiltration ist der Zeitpunkt des Aufrufs der Webseite durch das Opfer und den Angreifer. Dieser muss bereits im Voraus für beide Parteien vordefiniert sein. So kann eine Überprüfung des Caches anhand der folgenden drei Parameter durchgeführt werden:

- Aktuelle Zeit
- Reguläre Lebensdauer eines jeden Cache-Inhalts
- Ablaufzeit des Cache-Inhalts

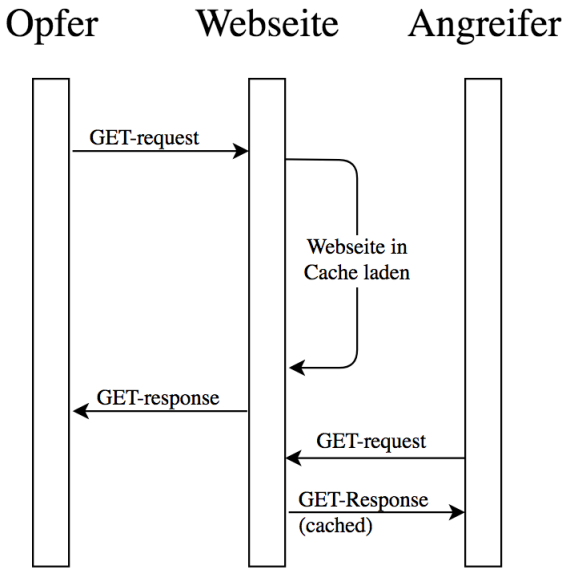

Abbildung 5: *Abbildung des Ablaufes zur Extraktion einer binären „1“ nach* Klein *und* Kotler *[KK16]*

Diese Informationen können nun verwendet werden, um anhand der Aufrufe einer Seite eine binäre Nachricht zu erstellen. Wie in Abbildung 5 dargestellt, muss der Angreifer nur analysieren ob vom Opfer der Aufruf der Webseite zum vereinbarten Zeitpunkt erfolgt ist oder nicht. Der Aufruf wird dabei anhand der Differenz der übrigen Lebensdauer des aktuellen Cache-Inhaltes (*Differenz = Ablaufzeit Cache − Aktuelle Zeit*) ermittelt. Entspricht diese Differenz dem Wert der regulären Lebensdauer eines Cache-Inhaltes, so wurde die Webseite vom Opfer nicht aufgerufen und der Angreifer erhält eine 0 übermittelt. Ist der Wert der errechneten Differenz geringer als die eigentliche Lebensdauer, so wurde die Webseite zuvor vom Opfer aufgerufen und der Angreifer erhält somit eine 1.

Die Verwendung dieser „perfekten“ Exfiltrationsmethode ist jedoch nur unter verschiedenen Anforderungen möglich [KK16, S. 3 f]. Zum einen soll hierzu eine bekannte Seite verwendet werden um kein außergewöhnliches Nutzerverhalten zu erzeugen. Zum anderen sollte die Webseite die Adressinformationen (z.B. IP-Adresse) vom letzten Zugriff nicht zwischenspeichern. Dies würde ansonsten eine erkennbare Verbindung zwischen Angreifer und Opfer bedeuten. Außerdem ist eine geringe Lebensdauer des Cache-Inhalts notwendig, da bei einer zu langen Lebensdauer zum einen die Wahrscheinlichkeit des Rauschens ansteigt und zum anderen eine Exfiltration gar unmöglich werden kann, da der Cache kaum geleert wird. Abschließend ist es zur Vermeidung von Rauschen wichtig, dass die verwendete Seite (z.B. ein Amazon-Artikel) sehr unregelmäßig bis kaum verwendet wird. Die Seitennutzung ist dabei abhängig vom Zeitraum während dem die Seite zur Informationsextraktion verwendet wird. Werden alle Anforderungen beachtet, so ist nach Klein und Kotler eine perfekte Exfiltrationsmethode möglich, die keine nachzuvollziehende Verbindung zwischen einer Kommunikation von Angreifer und Opfer über eine Drittanbieter-Webseite ermöglicht. Klein und Kotler beschreiben außerdem ein Verfahren, um ein potentielles Rauschen durch Aufrufe der Webseite durch dritte Personen zu filtern [KK16, S. 9 f].

2.3.2 E-Mail

Neben den Webprotokollen HTTP und HTTPS ist die E-Mail ein ebenso häufig verwendetes und beliebtes Protokoll zur Datenexfiltration. Dies belegt zum einen die Studie von McAfee [McA15] sowie zum anderen ein bekannter Angriff auf die deutsche Bundesregierung [Süd15]. Dabei wird nicht nur das Protokoll, sondern auch zugehörige Software wie Microsoft Outlook als potentielle Ressource für die Datenexfiltration verwendet. Vor allem die Vorteile der Overt Channels sind hierbei zu betrachten. E-Mails können oft nur schwer vom eigentlichen Nutzerverhalten unterschieden werden und sind in der Regel auch auf allen Geräten verfügbar.

Das Versenden von E-Mails kann dabei über unterschiedliche Möglichkeiten erfolgen. Hierzu ist zunächst die einfachste Möglichkeit des Versendens zu betrachten. Dies entspricht im Betriebssystem Windows der Nutzung des in der Powershell integrierten Tools *Send-MailMessage* [Som13]. Bei Linux und MacOS entspricht dies dem Tool *mail*, dem Äquivalent von *Send-MailMessage* [Fre06]. Mit dem Kommando *Send-MailMessage* kann über folgenden Befehl eine E-Mail über die Powershell versendet werden:

```
1 Send-MailMessage [-To] <string[]> [-Subject] <string> -From <string> [[-
    Body] <string>] [[-SmtpServer] <string>] [-Attachments <string[]>] [-
    Bcc <string[]>] [-BodyAsHtml] [-Cc <string[]>] [-Credential <
    PSCredential>] [-DeliveryNotificationOption {None | OnSuccess |
    OnFailure | Delay | Never}] [-Encoding <Encoding>] [-Priority {Normal
    | Low | High}] [-UseSsl] <CommonParameters>]
```

Auflistung 1: Template des Send-MailMessage Kommandos

Die Nutzung von *Send-MailMessage* ermöglicht es jedem Benutzer eine E-Mail über die Powershell zu versenden. Dies bedeutet eine einfache Exfiltrationsmethode, die der Benutzer des

infizierten Computers während seiner aktiven Nutzung nicht bemerkt. Der Netzwerkverkehr unterscheidet sich dabei nicht von der Nutzung herkömmlicher E-Mail-Programme, da der Ablauf und die Funktionsweise von *Send-MailMessage*, der eines herkömmlichen E-Mail-Programmes entspricht. Zu beachten ist jedoch, dass Anmeldedaten bei der Ausführung des Befehls benötigt werden. Zudem ist bei einer Analyse ersichtlich, dass es sich ggf. um eine unbekannte Sender-Adresse innerhalb des Netzwerks handelt, was auf eine mögliche Infizierung des Systems schließen lässt.

Neben der Nutzung von integrierten Tools, unterstützt die Nutzung installierter E-Mail Programme die Verschleierung des E-Mail-Verkehrs. Dabei ist in der Netzwerkanalyse nicht ersichtlich, dass es sich um eine mögliche Datenexfiltration handelt. Dies wird auch von Halani in seinem Beitrag auf niiconsulting.com beschrieben. Dabei wird der Vorgang einer Datenexfiltration mit Hilfe von Outlook erläutert [Hal16]. Mittels des folgenden Powershell-Skriptes kann auf Windows, sofern ein MS-Exchange Server konfiguriert ist, eine E-Mail über das installierte Outlook versendet werden. Ist Outlook ausgeführt, so sind ebenso keine Authentifizierungsdaten erforderlich.

```
1 [code lang="powershell"]
2 Add-Type -Assembly 'Microsoft.Office.Interop.Outlook' -PassThru;
3 $Outlook = New-Object -ComObject Outlook.Application;
4 $Mail = $Outlook.CreateItem(0);
5 $Mail.Recipients.Add('evil@attacker.com');
6 $Mail.Subject='Data Extracted';
7 $Mail.Body = $encodedCommand;
8 $Mail.Send();
9 [/code]
```

Auflistung 2: Versenden einer E-Mail mit Microsoft Outlook nach Halani [Hal16]

Bei der Verwendung von Outlook werden jedoch Fingerabdrücke hinterlassen, welche auch für den Nutzer des infizierten Computers sehr einfach ersichtlich sind. Solch ein Fingerabdruck ist die Kopie der versendeten E-Mail im *SENT*-Ordner. Diese kann jedoch mit Hilfe des folgenden zweiten Skriptes aus Outlook gelöscht werden. Dies ermöglicht das Verschleiern der versendeten Nachricht, ohne dass ein Nutzer dies bemerkt.

```
1 [code lang="powershell"]
2 $objOutlook = New-Object -ComObject Outlook.Application;
3 $objNamespace = $objOutlook.GetNamespace('MAPI');
4 $objFolder = $objNamespace.GetDefaultFolder(5);
5 $colItems = $objFolder.Items;
6 $colItems.Remove($colItems.Count);
7 [/code]
```

Auflistung 3: Löschen der E-Mail aus dem *SENT*-Ordner aus Microsoft Outlook nach Halani [Hal16]

2.3.3 Instant Messaging

Das Security Lancaster beschreibt, dass in der Nutzung von Instant Messaging (IM) Anwendungen, hauptsächlich die implementierte Funktion zur Dateiübertragung von Angreifern ausgenutzt wird [Ras+13, S. 16]. Dabei stellen installierte, jedoch aber nicht autorisierte Anwendungen zur Kommunikation ein hohes Risiko für Unternehmensnetzwerke dar. Dies kann mittels einer Einschränkung der Rechte, zur Installation von Software auf den Dienstgeräten, verhindert werden.

Jedoch auch vom Administrator und Sicherheitsexperten geprüfte IM-Anwendungen können für die Datenexfiltration genutzt werden. Ein Beispiel für solch eine Software ist Skype. Hartman beschreibt in seiner Arbeit die Möglichkeiten zur Nutzung von Skype als Werkzeug zur Datenexfiltration [Har14]. Skype nutzt zwar eine End-to-End-Verschlüsselung[4] für die Kommunikation, jedoch muss für eine Kommunikation nach außen jede öffentliche IP-Adresse verwendbar sein [Har14, S. 2]. Dies führt zu einem hohen Sicherheitsrisiko, durch eine notwendige Anpassung der Netzwerksicherheitsrichtlinien zur Erfüllung der genannten Anforderung zur Erreichbarkeit von außen. Ein weiteres Sicherheitsrisiko ist die dynamische Nutzung der Ports für die Verbindung nach außen. Skype nutzt vor allem die Ports 80 (HTTP) und 443 (HTTPS) für die Kommunikation zu anderen Clients und Servern. Somit ähnelt diese Kommunikation zunächst einer regulären Kommunikation des Opfers mit dem Internet. Sind diese Ports jedoch nicht verfügbar, können andere offene Ports für die Kommunikation verwendet werden. Somit ist es mittels einer bestmöglichen Konfiguration der Firewall kaum möglich, Skype zu blockieren.

Im Hinblick auf die Datenmenge ist Skype für Angreifer ideal geeignet. Der Skype-Netzwerkverkehr kann von wenigen 30 Kbps durch ein Telefonat bis hin zu 1 Mbps für Videoanrufe beanspruchen. Nutzt ein Angreifer nun diese Eigenschaften und täuscht für die Datenexfiltration einen gültigen Skype-Netzwerkverkehr vor, so kann er ohne Hindernis große Datenmengen exfiltrieren [Har14, S. 2 f]. Die notwendigen Eigenschaften hierzu werden in der Arbeit von Hartman genauer beschrieben [Har14, S. 3-7].

Zur Verwendung von Skype für eine unauffällige Datenexfiltration können verschiedene Bibliotheken verwendet werden [Sky; Kanc; Kanb; Kana]. Diese bieten die Möglichkeit einer Kommunikation über die Powershell bei Windows und der Bash sowie der Secure-Shell bei Linux-Systemen. Dabei ist diese Kommunikation zum einen für das Opfer nicht offensichtlich erkennbar und zum anderen wird hinsichtlich einer Netzwerkanalyse ein vermeintlich regulärer Datenverkehr erzeugt.

2.3.4 Steganographie

In der Anwendung von Steganographie in der Datenexfiltration ist eine Vielzahl an Methoden möglich. Darunter zu nennen sind die Bild-, Voice-over-IP (VoIP) und Netzwerk-Steganographie [Ras+13, S. 16 f]. Nach der Definition des Security Lancaster, bedeutet Steganographie

[4] Die Bezeichnung „End-to-End“ bedeutet hierbei, dass die transportierten Informationen beim Sender verschlüsselt und erst wieder beim Empfänger entschlüsselt werden.

im Allgemeinen ein sogenanntes „verstecktes Schreiben“ (engl. „hidden writting“). Die zu exfiltrierenden Daten werden, vor dem Versenden an den Angreifer, in unauffälligen Dateien oder Datenpaketen versteckt. In einer Analyse der gesendeten Daten sind die zu exfiltrierenden Dateninhalte dabei nicht eindeutig erkennbar.

Das Verstecken von Informationen in VoIP-Anrufen für eine unauffällige Datenexfiltration, ist vor allem aufgrund der weiten Verbreitung und Nutzung des VoIP interessant. Diese wirkt sich zudem kaum auffallend auf den Datenverkehr aus. Dabei reicht es bereits, wenn der Angreifer je VoIP-Paket ein „Least Significant Bit“ (LSB) anhängt und versendet. Ein einzelnes Bit je VoIP-Paket ermöglicht bereits eine sehr hohe Datenexfiltrationsrate von bis zu 50 bit/s [Ras+13, S. 16 f]. Zudem führt das Anhängen eines LSB zu keinerlei auffälligem Rauschen bei der Nutzung von VoIP.

Ein Anwendungsbeispiel der Steganographie mit Hilfe von VoIP wird mittels des Verfahrens von Schmidt, Keller, Caviglione und Mazurczyk erläutert [Sch+17]. Hierbei wird versucht durch VoIP mit aktivierter Sprachpausenerkennung (engl. „Voice Activity Detection“ (VAD)) Daten zu exfiltrieren. Dabei werden innerhalb der Sprachpausen Daten über einen Covert Channel übermittelt. Hierzu ist es notwendig ein VAD-VoIP in ein non-VAD-VoIP umzuwandeln. Innerhalb der Sprachpausen werden nun künstliche „Real-Time Transport Protocol“ (RTP) Pakete erzeugt und gesendet. Um Rauschen beim Empfänger zu verhindern und die Daten somit unauffällig aus dem Netzwerk zu exfiltrieren, werden zwischen den RTP Paketen, welche die zu exfiltrierenden Daten enthalten, Pausen eingefügt (siehe Abbildung 6).

Abbildung 6: *Schematischer Aufbau eines transformierten VoIP-Streams zur Datenexfiltration nach* Schmidt, Keller, Caviglione *und* Mazurczyk *[Sch+17, S. 2 f]*

Die in Abbildung 6 schwarz gekennzeichneten Felder entsprechen den Paketen mit den zu exfiltrierenden Daten. Die weißen Felder innerhalb einer Sprachpause dagegen den Paketen einer Pause zur Verschleierung der exfiltrierenden Datenpakete. Im Unterschied zur Anwendung einer VAD besitzt der VoIP-Stream nun innerhalb der Ruhephase Pakete. Das von Schmidt, Keller, Caviglione und Mazurczyk beschriebene Verfahren ermöglicht je nach Implementierung und Umsetzung eine Payload-Größe von 29 bis 156 Bytes und einer Transferrate von 0,11 bis 7,09 kbit/s [Sch+17, S. 4 f].

2.3.5 Peripherie

Eine Studie von McAfee [McA15] zeigt ebenso einen Überblick über die am häufigsten verwendeten Methoden der Datenexfiltration mittels Hardware. Dabei sind Peripherie-Geräte wie USB-Sticks noch immer eine hohes Risiko für eine Infizierung des Systems und einen folgenden

Datendiebstahl. Laptops und PC's stehen aktuell jedoch an oberster Stelle. Dies ist vorallem mit der nun weit verbreiteten Umsetzung der „Bring-Your-Own-Device“ (BYOD) Philosophie zu begründen und wird durch McAfee und das Security Lancaster als einer der größten Schwachpunkte eines Unternehmensnetzwerks bestätigt [Ras+13; McA15]. Detaillierte Verfahren zu Peripherie werden im Rahmen dieser Arbeit nicht weiter betrachtet, da eine Analyse und Überwachung auf privaten Geräten nicht immer möglich ist und die Fingerabdrücke eines Angriffes in den bereits aufgeführten Methoden und Mechanismen ermittelt werden können.

3 Fazit und Ausblick

Die in dieser Arbeit untersuchten Mechanismen von APTs beschreiben bereits eine Vielzahl an Möglichkeiten für einen Angriff. Dies ist jedoch nur ein kleiner Teil möglicher Methoden unter der Ausführung eines APT. Die Analyse der verwendeten Tools zeigt auf, dass eine Einordnung bzw. spezielle Analyse von Software nur bedingt auf einen einzelnen Mechanismus hinweist. Die Anwendungsbereiche überschneiden sich in vielen Phasen innerhalb der Intrusion-Kill-Chain sowie aller drei Mechanismen. Die untersuchten Mechanismen weisen jedoch auch auf erste Möglichkeiten der Detektion von APTs hin. Ebenso die vorgestellten Konzepte aus dem aktuellen Stand der Forschung bestätigen dies. Bislang gibt es jedoch auch für vereinzelte Angriffe, wie der perfekten Exfiltration [KK16], keine Möglichkeit diese zu erkennen. Somit lässt sich daraus folgern, dass ein APT nur durch die Kombination der untersuchten Mechanismen erfolgreich detektiert werden kann.

Die Zusammenfassung der in dieser Arbeit genannten möglichen Szenarien soll nun als Grundlage für weitere Arbeiten im Rahmen des Forschungsprojektes DARCNET dienen. Dabei wird in der nächsten Projektarbeit eine erste Analyse von schwer erkennbaren Angriffen durchgeführt. Hierfür werden die in dieser Arbeit genannten Mechanismen genauer analysiert. Das Ziel ist es nun, mit Hilfe von maschinellen Lernverfahren und der bekannten Mechanismen eine Detektion eines Angriffes zu ermöglichen.

Anhang

A Erklärung zur Projektarbeit

1. Mir ist bekannt, dass dieses Exemplar der Projektarbeit als Prüfungsleistung in das Eigentum der Ostbayerischen Technischen Hochschule Regensburg übergeht.

2. Ich erkläre hiermit, dass ich diese Projektarbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Regensburg, den 18.09.2018

Unterschrift

B Abbildungsverzeichnis

C Literaturverzeichnis

[Rus04] Mark Russinovich. *PsExec*. 2004. URL: https://www.itprotoday.com/management-mobility/psexec (besucht am 25. 08. 2018).

[KVV05] Christopher Kruegel, Fredrik Valeur und Giovanni Vigna. *Intrusion Detection and Correlation - Challenges and Solutions*. Bd. 14. Advances in Information Security. Springer, 2005, S. 1–118. ISBN: 978-0-387-23398-7. URL: http://dx.doi.org/10.1007/b101493.

[Fre06] FreeBSD. *FreeBSD General Commands Manual: mail*. 2006. URL: https://www.freebsd.org/cgi/man.cgi?query=mail%7B%5C&%7Dapropos=0%7B%5C&%7Dsektion=0%7B%5C&%7Dmanpath=FreeBSD+11.2-RELEASE+and+Ports%7B%5C&%7Darch=default%7B%5C&%7Dformat=html (besucht am 20. 08. 2018).

[Gus10] Gustavo Miguel Barroso Assis do Nascimento. "Anomaly Detection of Web-Based Attacks". Diss. Univsersität Lissabon, 2010. ISBN: 1581137389.

[Ant11] Ryan C Van Antwerp. "Exfiltration Techniques: an Examination and Emulation". In: (2011), S. 1–63. URL: https://repo.zenk-security.com/Techniques%20d.attaques%20.%20Failles/Exfiltration%20Techniques%20-%20An%20examination%20And%20Emulation.pdf.

[HCA11] Eric M. Hutchins, Michael J. Cloppert und Rohan M. Amin. "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains". In: *6th Annual International Conference on Information Warfare and Security* July 2005 (2011), S. 1–14. URL: http://papers.rohanamin.com/wp-content/uploads/papers.rohanamin.com/2011/08/iciw2011.pdf%7B%5C%%7D5Cnhttp://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf.

[War11] Martin Warmer. "Detection of web based command & control channels". In: November (2011). URL: http://essay.utwente.nl/61232/.

[Ins13] Institut InfoSec. *Data Exfiltration Techniques*. 2013. URL: https://resources.infosecinstitute.com/data-exfiltration-techniques-2/ (besucht am 06. 08. 2018).

[Ras+13] Awais Rashid u. a. *Detecting and Preventing Data Exfiltration*. Techn. Ber. Security Lancester Laboratory, 2013.

[Som13] Wolfgang Sommergut. *Send-MailMessage: E-Mails versenden mit PowerShell*. 2013. URL: https://www.windowspro.de/script/send-mailmessage-e-mails-versenden-powershell (besucht am 20. 08. 2018).

[Tre13] Trend Micro. "How Do Threat Actors Move Deeper Into Your Network?" In: (2013). URL: http://about-threats.trendmicro.com/cloud-content/us/ent-primers/pdf/tlp%7B%5C_%7Dlateral%7B%5C_%7Dmovement.pdf.

[Xu+13] Kui Xu u. a. “DNS for massive-scale command and control”. In: *IEEE Transactions on Dependable and Secure Computing* 10.3 (2013), S. 143–153. ISSN: 15455971. DOI: 10.1109/TDSC.2013.10.

[BYG14] Parth Bhatt, Edgar Toshiro Yano und Per Gustavsson. “Towards a framework to detect multi-stage advanced persistent threats attacks”. In: *Proceedings - IEEE 8th International Symposium on Service Oriented System Engineering, SOSE 2014* (2014), S. 390–395. ISSN: 9781479936168. DOI: 10.1109/SOSE.2014.53.

[CDH14] Ping Chen, Lieven Desmet und Christophe Huygens. “A Study on Advanced Persistent Threats”. In: *Communications and Multimedia Security* 8735 (2014), S. 63–72. DOI: 10.1007/978-3-662-44885-4_5. URL: http://link.springer.com/chapter/10.1007/978-3-662-44885-4%7B%5C_%7D5.

[Har14] Kenneth G. Hartman. *Skype and Data Exfiltration*. Techn. Ber. SANS Institute, 2014.

[Gra15] Matt Graeber. “Abusing Windows Management Instrumentation (WMI) to Build a Persistent , Asyncronous , and Fileless Backdoor”. In: *Black Hat* (2015).

[McA15] McAfee. “Grand Theft Data”. In: (2015), S. 4. ISSN: 00157228. URL: https://www.mcafee.com/us/resources/reports/rp-data-exfiltration.pdf.

[Süd15] Süddeutsche Zeitung. *Hackerangriff auf den Bundestag - Gesamtes IT-Netz des Bundestages muss ausgetauscht werden*. 2015. URL: http://www.sueddeutsche.de/politik/hackerangriff-auf-den-bundestag-gesamtes-it-netz-des-bundestages-muss-ausgetauscht-werden-1.2519934%7B%5C#%7Dredirected%20FromLandingpage (besucht am 05. 02. 2018).

[Zel15] Lenny Zeltser. *Using ICMP Reverse Shell to Remotely Control a Host*. 2015. URL: https://zeltser.com/reverse-icmp-shell/ (besucht am 01. 09. 2018).

[Hal16] Wasim Halani. *Exfiltration Using Powershell and Outlook*. 2016. URL: http://niiconsulting.com/checkmate/2016/03/exfiltration-using-powershell-outlook/ (besucht am 20. 08. 2018).

[KK16] Amit Klein und Itzik Kotler. *In Plain Sight : The Perfect Infiltration*. Techn. Ber. Safebreach Labs, 2016.

[Smo16] Smokescreen Technologies Pvt. Ltd. “The Top 20 Lateral Movement Tactics”. In: (2016), S. 10. URL: https://www.smokescreen.io/wp-content/uploads/2016/08/Top-20-Lateral-Movement-Tactics.pdf.

[Uss+16] Martin Ussath u. a. “Advanced persistent threats: Behind the scenes”. In: *2016 50th Annual Conference on Information Systems and Sciences, CISS 2016* (2016), S. 181–186. DOI: 10.1109/CISS.2016.7460498.

[Wan+16] Xu Wang u. a. “Detection of command and control in advanced persistent threat based on independent access”. In: *2016 IEEE International Conference on Communications, ICC 2016* (2016). DOI: 10.1109/ICC.2016.7511197.

[AA17] Azeria und Azeria-Labs. *Data Exfiltration*. 2017. URL: https://azeria-labs.com/data-exfiltration (besucht am 06. 08. 2018).

[GiM17] GiMini. *PowerMemory: Exploit the credentials present in files and memory*. 2017. URL: https://github.com/giMini/PowerMemory (besucht am 21. 08. 2018).

[Lam17] Tony Lambert. *Lateral Movement Using WinRM and WMI*. 2017. URL: https://www.redcanary.com/blog/lateral-movement-winrm-wmi/ (besucht am 21. 08. 2018).

[Mic17] Microsoft. *Distributed Component Object Model (DCOM) Remote Protocol*. 2017. URL: https://msdn.microsoft.com/en-us/library/cc226801.aspx (besucht am 27. 08. 2018).

[Nel17a] Matt Nelson. *Lateral Movement using the MMC20.Application COM Object*. 2017. URL: https://enigma0x3.net/2017/01/05/lateral-movement-using-the-mmc20-application-com-object/ (besucht am 27. 08. 2018).

[Nel17b] Matt Nelson. *Lateral Movement via DCOM: Round 2*. 2017. URL: https://enigma0x3.net/2017/01/23/lateral-movement-via-dcom-round-2/ (besucht am 27. 08. 2018).

[Qin17] QinetiQ Ltd. "Command and Control: Understanding, denying, detecting". In: *QinetiQ Proprietary* (2017), S. 36.

[Sch+17] Sabine S. Schmidt u. a. "A New Data-Hiding Approach for IP Telephony Applications with Silence Suppression". In: *Proceedings of the 12th International Conference on Availability, Reliability and Security - ARES '17* (2017), S. 1–6. DOI: 10.1145/3098954.3106066. URL: http://dl.acm.org/citation.cfm?doid=3098954.3106066.

[Sha+17] Pradip Kumar Sharma u. a. "DFA-AD: a distributed framework architecture for the detection of advanced persistent threats". In: *Cluster Computing* 20.1 (2017), S. 597–609. ISSN: 15737543. DOI: 10.1007/s10586-016-0716-0.

[Maz18] Ryan Mazerik. *ICMP Attacks*. 2018. URL: https://resources.infosecinstitute.com/icmp-attacks/ (besucht am 01. 09. 2018).

[Mic18] Microsoft. *The Component Object Model*. 2018. URL: https://docs.microsoft.com/de-de/windows/desktop/com/the-component-object-model (besucht am 27. 08. 2018).

[Tsu18] Philip Tsukerman. *New lateral movement techniques abuse DCOM technology*. 2018. URL: https://www.cybereason.com/blog/dcom-lateral-movement-techniques (besucht am 27. 08. 2018).

[Eck] Ines Maria Eckermann. "Was ist eigentlich Social Engineering?" In: (). URL: https://www.gdata.de/ratgeber/was-ist-eigentlich-social-engineering.

[Kana] Gopi Kannan. *How to send a skype message using Powershell?* URL: https://gopekannan.wordpress.com/2017/07/23/how-to-send-a-skype-message-using-powershell/ (besucht am 20. 08. 2018).

[Kanb] Gopi Kannan. *PowerShell: Send instant message to another user in LDAP - Skype for Business.* URL: https://github.com/gopekanna/test%7B%5C_%7DSkype%7B%5C_%7DPeertoPeer%7B%5C_%7DInstant (besucht am 20. 08. 2018).

[Kanc] Gopi Kannan. *Voice Call, Video Call, Desktop Sharing and Mute/Unmute in Skype for Business using Lync API and Powershell.* URL: https://github.com/gopekanna/test%7B%5C_%7DSkype%7B%5C_%7DAudioVideo%7B%5C_%7DSharing (besucht am 20. 08. 2018).

[Nma] Nmap.org. *NMAP - Port-Scanning-Methoden.* URL: https://nmap.org/man/de/man-port-scanning-techniques.html (besucht am 21. 08. 2018).

[Sky] Skype4Py GitHub-Community. *Skype4Py: Platform independent Python wrapper for the Skype Desktop API.* URL: https://github.com/Skype4Py/Skype4Py (besucht am 20. 08. 2018).

D Abkürzungsverzeichnis

AppID	Application Identifier
APT	Advanced Persistent Threat
BYOD	Bring your own device
C2	Command and Control
CART	Classification and Regression Trees
COM	Component Object Model
CSLID	Class Identifier
DBG-Model	Dynamic Bayesian Game Model
DCOM	Distributed Component Object Model
DNS	Domain Name System
DFA-AD	Distributed Framework Architecture for APT Detection
FPR	False-Positive-Rate
FTP	File Transfer Protocol
GP	Genetische Programmierung
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IKC	Intrusion-Kill-Chain
IM	Instant Messaging
LSB	Least Significant Bit
ProgID	Programmatic Identifier
RTP	Real-Time Transport Protocol
SMB	Server Message Block
SSH	Secure-Shell
SVM	Support Vector Machine
TPM	Trusted Platfrom Module
UAC	User Account Control
URL	Uniform Resource Locator

VAD	Voice Activity Detection
VM	Virtuelle Maschine
VoIP	Voice over IP
WinRM	Windows Remote Management
WMI	Windows Management Instrumentation