St. Pölten University of Applied Sciences

/informatik & security

/fh/// st.pölten

Vom Log Event zum Alert

Erstellung und Evaluierung eines Prozesses für die Konfiguration von Alert Rules im SIEM am Beispiel LogPoint und Active Directory

Diplomarbeit

zur Erlangung des akademischen Grades

Diplom-Ingenieur/in

eingereicht von

Königshofer Bettina, BSc.

51825026

im Rahmen des

Studienganges Information Security an der Fachhochschule St. Pölten

Betreuung

Betreuer/in: Dipl.-Ing. Daniel Haslinger, BSc

Mitwirkung: -

St. Pölten, January 20, 2023

(Unterschrift Verfasser/in) (Unterschrift Betreuer/in)

Fachhochschule St. Pölten GmbH, Campus-Platz 1, A-3100 St. Pölten,T: +43 (2742) 313 228, F: +43 (2742) 313 228-339, E: office@fhstp.ac.at, I: www.fhstp.ac.at

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich sonst keiner unerlaubten Hilfe bedient habe.
- ich das Thema dieser Arbeit bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.
- diese Arbeit mit der vom Begutachter/von der Begutachterin beurteilten Arbeit übereinstimmt.

Die Studierende/Absolventin räumt der FH St. Pölten das Recht ein, diese Arbeit für Lehre- und Forschungstätigkeiten zu verwenden und damit zu werben (z. B. bei der Projektevernissage, in Publikationen, auf der Homepage), wobei die Absolventin als Urheberin zu nennen ist. Jegliche kommerzielle Verwertung/Nutzung bedarf einer weiteren Vereinbarung zwischen der Studierenden/Absolventin und der FH St. Pölten.

Ort, Datum *Unterschrift*

Kurzfassung

Das Logvolumen an sicherheitsrelevanten Logs steigt mit jedem Gerät, das in einem Unternehmen in Betrieb genommen wird. Mithilfe dieser Logs ist es möglich anormale Aktivitäten zu identifizieren und so möglicherweise Angriffe zu verhindern, bevor größerer Schaden angerichtet werden kann. Doch, um das zu erreichen muss diese enorme Menge an Logs gesammelt und ausgewertet werden. Aus diesem Grund wird in vielen Unternehmen bereits ein Security and Event Management (SIEM) Tool eingesetzt.

Die Implementierung eines SIEM-Tools alleine reicht allerdings nicht aus, um das Potential dieser Logs voll auszuschöpfen. Es genügt nicht, dass die Logs nur gesammelt werden, sie müssen zudem auch geparsed, normalisiert, angereichert und korreliert werden, um sie zum Schutz vor Angriffen verwenden zu können. Anschließend müssen für diese Logs Alert Regeln erstellt werden, damit definiert werden kann, was anormale Aktivitäten darstellen.

Da die Alert Regeln das Kernstück eines SIEM-Tools sind und es ohne diese Regeln wenig Mehrwert in der Erkennung von potentiell boshaftem Verhalten bietet, wurde ein Prozess erarbeitet in dem eine Basis an Alert Regeln strukturiert und gezielt erarbeitet werden kann. Dieser Prozess sieht vor, dass zuerst Use Cases definiert werden anhand denen bestimmt wird, welche Logs bzw. Logquellen benötigt werden, um diese erfüllen zu können. Anschließend müssen die Logquellen entsprechend konfiguriert werden, um Logs an das SIEM zu senden. Danach müssen die Alert Regeln definiert werden. Dazu können unter anderem auch Alert Regel Vorlagen verwendet werden, die das SIEM-Tool zur Verfügung stellt. Einer der wichtigsten Schritte ist die anschließende Evaluierung dieser Alert Regeln. Eine Regel sollte nur aktiviert werden, wenn sie entsprechenden Mehrwert für den Use Case bietet, sie nicht redundant ist und eine vernünftige Anzahl an Alerts im Normalzustand generiert.

In dieser Arbeit wurde dieser Prozess am Beispiel Active Directory und LogPoint SIEM getestet. Dabei wurden die Use Cases Valid Accounts, Account Manipulation, Brute Force und Exploitation for Privilege Escalation nach MITRE ATT&CK Framework gewählt. Nach Konfiguration und Anbindung der Windows Security Event Logs, wurden die built-in Alert Regeln evaluiert. Dabei wurde festgestellt, dass nur etwas über 50 Prozent der Regeln unverändert aktiviert werden können, der andere Teil benötigt unterschiedliche Anpassungen um entsprechend zu funktionieren. Zudem wurden 20 Prozent der Regeln nach der Evaluierung deaktiviert, da sie redundant waren oder den Use Case nicht entsprechend unterstützt haben.

Abstract

The log volume of security-related logs in a compay increases with every device that is deployed. With these logs, it is possible to identify abnormal activities and thus possibly prevent attacks before major damage can be done. But in order to achieve this, this huge amount of logs needs to be collected and analysed. For this reason, many companies already use a Security and Event Management (SIEM) tool. However, implementing a SIEM tool alone is not enough to fully utilise the potential of these logs. It is not sufficient that the logs are only collected, they must also be parsed, normalised, enriched and correlated in order to use them to protect against attacks. Alert rules must then be created for these logs to define what constitutes abnormal activity. Since the alert rules are the core of a SIEM tool and the SIEM without these rules offers little benefit in the detection of potentially malicious behaviour, a process was developed in which a alert rule set can be developed in a structured and targeted manner. This process proposes that use cases are defined first, which are used to determine which logs or log sources are needed to fulfil these use cases. Then the log sources must be configured accordingly in order to send logs to the SIEM. Afterwards, the alert rules must be defined. Alert rule templates provided by the SIEM tool can be used for this purpose, among others. One of the most important steps is the subsequent evaluation of these alert rules. A rule should only be activated if it offers appropriate added value for the use case, it is not redundant and it generates a reasonable number of alerts under normal conditions.

In this thesis this process was tested using Active Directory and LogPoint SIEM as examples. The use cases Valid Accounts, Account Manipulation, Brute Force and Exploitation for Privilege Escalation according to MITRE ATT&CK Framework were selected. After configuration and connection of the Windows Security Event Logs, the built-in alert rules were evaluated. It was found that only a little over 50 percent of the rules can be activated unchanged, the other part needs various adjustments to function accordingly. In addition, 20 percent of the rules were deactivated after the evaluation because they were redundant or did not support the use cases appropriately.

Contents

1 Einleitung

Jedes System in einem Unternehmen, von Firewalls über Endgeräte bis hin zu Industrial Control Systems, generiert Logs. Jeder einzelne dieser generierten Logeinträge enthält Informationen zu einem bestimmten Ereignis, wie beispielsweise einem Login, einem Verbindungsaufbau oder einer Konfigurationsänderung [1]. Abhängig von der Art und Anzahl der Systeme kommen dabei täglich mehrere Terabyte an Logdaten zusammen.

Werden diese Logs entsprechend zentral gesammelt und verwertet, können diese für Security Analyst/innen extrem wertvoll sein, um Angriffe zu verhindern oder diese frühzeitig zu erkennen. Damit das riesige Volumen an Logs allerdings verarbeitet werden kann, wird in vielen Unternehmen immer öfter ein Security Information und Event Management (SIEM) System eingesetzt. Alleine in den letzten zwei Jahren ist das Marktvolumen von 3,58 Milliarden US-Dollar im Jahr 2020 [2] auf 5,2 Milliarden US-Dollar im Jahr 2022 gestiegen und weiteres Wachstum wird erwartet [3].

Ein SIEM-Tool bietet die Möglichkeit die Daten zu aggregieren und zu speichern sowie Funktionen zur Analyse, Korrelation und Visualisierung. Um diese Funktionen im vollen Umfang und zielführend nutzen zu können, reicht die einfache Implementierung des SIEM nicht aus. Es hat aus Security-Sicht nur wenig Mehrwert, beliebige Logdaten von Systemen an das SIEM zu schicken und sie dort zu speichern. Stattdessen muss eine methodische Vorgehensweise entwickelt werden, wie entschieden wird, welche Logquellen anzubinden sind, welche Ereignisse gespeichert werden sollen und welche Alert Regel es braucht, um verdächtige Aktivitäten in den ausgewählten Logereignissen zu erkennen. Wie dieser Prozess aussehen und in weiterer Folge angewendet werden kann, zeigt diese Arbeit. Dazu werden zuerst die notwendigen Schritte des Prozesses beschrieben und anschließend am Beispiel Active Directory angewendet.

1.1 Struktur der Arbeit

Die Arbeit ist in mehrere Kapitel unterteilt. Kapitel 1 stellt das Thema sowie die Motivation vor, während im Kapitel 2 die wichtigsten Grundlagen zu SIEM-Systemen, dem MITRE ATT&CK Framework[1] sowie Active Directory und Kerberos beschrieben werden. Kapitel 3 beschäftigt sich mit bereits publizierten wissenschaftlichen Arbeiten, welche sich mit verwandten Themenstellungen mit dem Fokus SIEM beschäftigen. Kapitel 4 beschreibt, wie die der Prozess erarbeitet und anschließend getestet wird. In Kapitel 5 wird das vorgeschlagene Vorgehen beschrieben, um von einem Use Case zu einem zielführenden Satz an Alert Regeln zu kommen. Anschließend wird dieser vorgeschlagene Prozess im Kapitel 6 am Beispiel Benutzerkontensicherheit im Active Directory durchgeführt. Abschließend werden im Kapitel 7 die Erkenntnisse zusammengefasst.

1.2 Limitationen

Diese Arbeit konzentriert sich vorrangig auf die Erstellung eines Prozesses zur Definition einer Alert Regel Basis und des anschließenden Tests dieser Vorgehensweise. Bei den Funktionalitätstests der Regeln, wird ausschließlich dahingehend getestet, ob eine Regel laut ihrer Definition funktioniert. Es werden dabei zum Teil vereinfachte Tests ausgeführt bzw. Parameter, wie die Zeit oder Grenzwerte verändert. Es wird nicht getestet, ob die Regeln bei tatsächlichen Angriffen alarmieren. Zudem finden die Tests der Effektivität in einer Produktivumgebung statt. Das bedeutet, dass die daraus resultierenden Ergebnisse für die jeweilige Umgebung gültig sind, aber nicht generalisiert werden dürfen.

[1] https://attack.mitre.org/

2 Grundlagen

Im folgenden Abschnitt werden die grundlegenden Themen, die diese Arbeit behandelt erläutert.

2.1 SIEM

Security Information and Event Management (SIEM) Systeme sind ein mittlerweile weit verbreitetes Werkzeug (englisch: Tool) um Cyber-Attacken zu erkennen und um darauf zu reagieren [4]. Im Normalfall sammelt ein SIEM die Events aus der Infrastruktur, um sie anschließend zu aggregieren, zu speichern und zu analysieren [5]. Das SIEM gilt als zentrale Plattform der modernen Security Operations Center (SOC), da sie Events von vielen verschiedenen Sensoren, wie Intrusion Detection Systeme (IDS), Firewalls und Anti-Viren-Systeme u.v.m. sammeln und korrelieren, um daraus Anzeichen für Bedrohungen zu erkennen, Dashboards zu generieren oder Berichte zu erstellen [6], [7].

Der Begriff SIEM wurde im Jahr 2005 von Gartner eingeführt. Es löst somit zwei Systeme ab, das Security Information Management (SIM) System und das Security Event Management (SEM) System. Ein SIM speichert Informationen über einen langen Zeitraum und führt Analysen sowie Berichterstattung durch, während ein SEM Ereignisse in Echtzeit sammelt. Die Kombination beider Systeme erlaubt es nun Ereignisse in beinahe Echtzeit zu sammeln, zu analysieren und dem/der Analyst/in ein überschaubares Bild zu präsentieren, um davon in weiterer Folge Aktionen und Maßnahmen ableiten zu können [8].

2.1.1 Aufbau eines SIEMs

SIEM Lösungen entwickeln sich immer mehr hin zu umfassenden Plattformen, welche aus drei Kernelementen bestehen. Das erste Element beschäftigt sich mit der Datensammlung sowie -management, während es bei dem zweiten Element um die Analyse dieser Daten geht. Das dritte Element wird als Operations bezeichnet und behandelt jene Aspekte, die zur Nutzung des SIEMs benötigt werden. Abbildung 2.1 zeigt eine Übersicht über die Kernelemente eines SIEM sowie deren Komponenten [9].

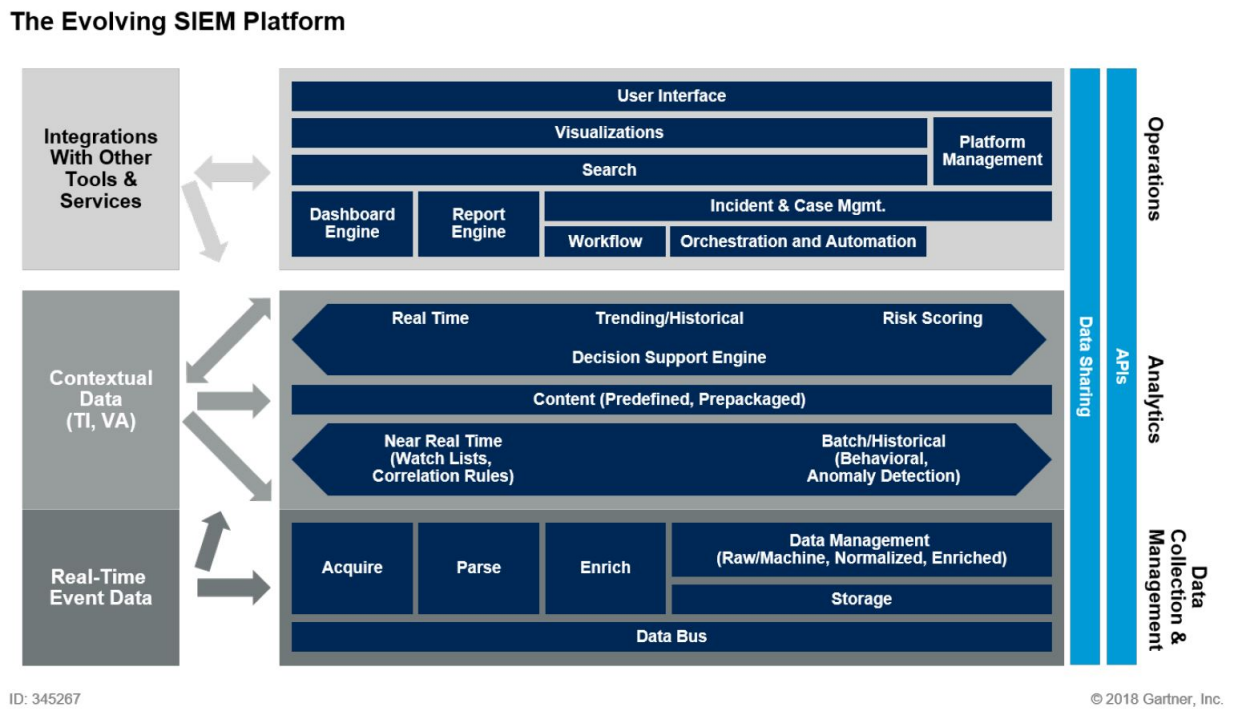

Abbildung 2.1: SIEM-Kernelemente und ihre Komponenten [9]

2.1.2 Datensammlung und -management

Schon die ersten SIEM Lösungen hatten die Möglichkeit Daten zu sammeln und zu verwalten. Diese Funktionen wurden zu einem Zeitpunkt entwickelt, als der Fokus des SIEMs darauf lag, eine begrenzte Menge an Daten in Echtzeit zu überwachen zum Beispiel von kritischen Sicherheitstechnologien, wie Firewalls und Intrusion Prevention Systeme (IPS). Diese Systeme waren allerdings nicht dafür ausgelegt große Mengen an Daten zu speichern und meist wurden nur wenige bestimmte Datenquellen unterstützt [9].

Über die letzten Jahre haben sich die SIEM Lösungen immer weiter entwickelt und wurden zu einem Standard-Tool in der IT-Sicherheit. Mittlerweile werden eine Vielzahl von Datenquellen zur Anbindung an das SIEM unterstützt. Das führt dazu, dass eine riesen Menge an Daten anfällt, die gespeichert werden müssen. Aus diesem Grund haben SIEM Hersteller Big Data Technologien für ihre Zwecke adaptiert und in ihre Tools integriert. Das bietet weitere Funktionen und Möglichkeiten, wie zum Beispiel die Möglichkeit der leistbaren, horizontal skalierbaren Datenspeicherung und die Funktion, Machine-Learning-basierte Analysen auf den gespeicherten Daten durchzuführen [9].

Dass Bedrohungserkennung mittlerweile zu einem der Haupt-Treiber für SIEM Lösungen gehört, wird nach Wegen gesucht, die Identifikation, Analyse, Triage, Priorisierung sowie Response-Möglichkeiten weiter zu verbessern, um eine immer weiter steigende Anzahl an Incidents behandeln zu können. Somit wurde auch

das sogenannte "Enrichment", die Anreicherung der Daten um weitere Informationen, zu einem wichtigen Bestandteil eines SIEMs. Dabei werden die Daten nicht mehr nur geparsed, normalisiert und gespeichert, sondern auch um kontextuelle Informationen bereichert. Je nach SIEM-Lösung werden diese Daten im bereicherten Format gespeichert oder die Daten werden weiterhin im Roh-Format gespeichert und das Enrichment findet bei der Abfrage der Logs statt [9].

Analyse

Die Entwicklung der Analyse-Fähigkeiten von SIEM-Lösungen wird davon getrieben, möglichst gute Erkennungsraten (wenig False Positives, mehr True Positives) zu erreichen, während gleichzeitig die Kosten für den Betrieb und die Feinabstimmung niedrig gehalten werden sollen. Hersteller arbeiten dazu meist mit komplexen Korrelationsregeln oder mit einfachen statistischen Analysen. Aktuell unterstützt jede SIEM-Lösung regel-basierte Analysen in Form von Korrelationsregeln sowie Whitelists, in den unterschiedlichsten Ausbauformen und Implementierungen [9].
In den letzten Jahren wurden einige SIEM-Lösungen durch UEBA-Funktionen (User and Entitiy Behavior Analytics) erweitert. Dabei werden komplexe statistische Analysen sowie Machine-Learning verwendet, um beispielsweise Abweichungen am Benutzerverhalten feststellen zu können [9].

Betrieb (Operations)

Während die Erkennung einer Bedrohung essentiell für die Sicherheit eines Unternehmens ist, so ist es nicht das Endziel sondern der Anfang eines komplexen und oftmals manuellen Prozesses, welcher die Triage von Warnungen, Validierung, Untersuchung und Reaktion bzw. Beseitigung der Bedrohung beinhaltet. Der Schlüssel zur Effizienz liegt somit darin, diese Schritte möglichst reibungslos durchführen zu können. Gartner beschreibt das so, dass sich das Ziel von der Mean-Time-to-Detect (MTTD) immer mehr zu Mean-Time-to-Remediation (MTTR) verschiebt [9]. Das bedeutet, dass sich der Fokus etwas verschiebt. Es ist geht nicht mehr nur darum, dass die Bedrohung so schnell wie möglich erkannt wird, sondern darum, dass diese so schnell wie möglich beseitigt und das System wiederhergestellt wird.
Aus diesem Grund haben Unternehmen immer mehr Wert darauf gesetzt, dass die SIEM-Lösungen einen effizienten Ende-zu-Ende-Prozess unterstützen. Daraus hat sich entwickelt, dass die meisten SIEM-Tools bereits Incident- oder Case-Management Funktionen besitzen sowie die Möglichkeit weitere Informationen in das System zu bekommen, zum Beispiel durch den Zugriff auf andere Tools via API. Manche Systeme unterstützen bereits SOAR-Funktionen (Security Orchestration, Automation and Response). Dabei können weitere Prozessschritte, von der Triage, über Teile der Untersuchung bis hin zum setzen von bestimmten

Maßnahmen, automatisiert werden [9].

2.1.3 Funktionsweise des SIEMs

Ein einfaches SIEM besteht im Normalfall aus mehreren einzelnen Bausteinen, die unabhängig voneinander arbeiten können, aber nur durch ihre Zusammenarbeit kann das SIEM vollumfänglich funktionieren [4]. Abbildung 2.2 zeigt diese Basiskomponenten.

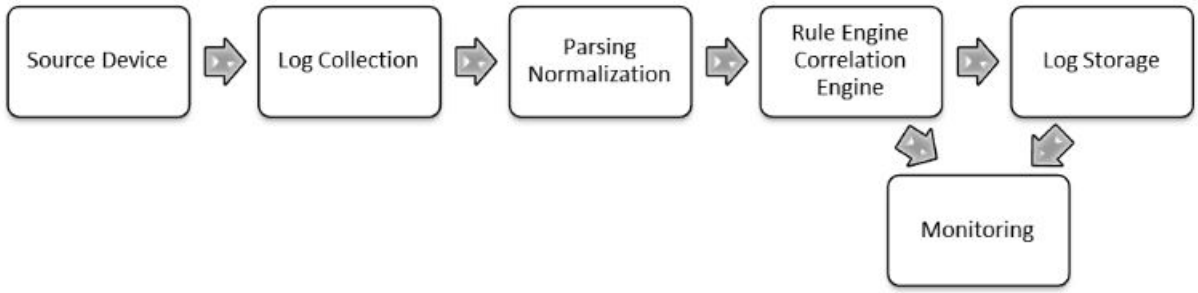

Abbildung 2.2: SIEM-Basiskomponenten [4]

Abbildung 2.2 zeigt den Weg den ein Log von der Log Quelle, in der Abbildung "Source Device" genannt, bis zur Speicherung (Log Storage) geht. Zuerst muss das Ereignis entweder proaktiv an das SIEM geschickt oder vom SIEM geholt bzw. abgefragt werden. Anschließend muss der Logeintrag geparsed und normalisiert werden, um sicherzustellen, dass jedes Log die gleiche Form hat. Je nach SIEM wird an dieser Stelle zusätzlich auch das Enrichment durchgeführt. Abschließend wird das Log gespeichert, dabei wird aber von der Rule bzw. Correlation Engine immer wieder darauf zugegriffen, um auf vordefinierte Anomalien zu überprüfen und in das Monitoring einfließen zu lassen [4].
Um diesen Vorgang durchzuführen, braucht es in den meisten Fällen folgende Elemente [10]:

- Agents: Sammeln Daten von verschiedenen Quellen ein
- Collector-Server: Sammeln die Daten ein, die sie von den Agents erhalten
- Datenbank-Server: Speichert die Informationen
- Korrelations-Server: Führt die Informationsanalyse durch

2.1.4 Herausforderungen in SIEM-Lösungen

Die aktuellen SIEM-Lösungen bieten viele leistungsstarke Features in Bezug auf Korrelation, Speicher, Visualisierung und Performance, als auch die Möglichkeit einzelne Schritte zu automatisieren oder Gegenmaßnahmen zu setzen. Allerdings gibt es aktuell noch einige Herausforderungen und Einschränkungen, die

in Betracht gezogen werden müssen [11], [12].

Datenformate

Da die Daten aus vielen unterschiedlichen Logquellen kommen, haben diese die verschiedensten Formate. Um eine einheitliche Verarbeitung zu gewährleisten, müssen die Original-Logs (englisch: Raw-Logs) in ein homogenes Format transformiert werden. Wie dieses Format aussieht, unterscheidet sich von SIEM zu SIEM. Für diese Umwandlung, die auch Normalisierung genannt wird, werden Adapter benötigt. Diese Adapter sind meistens in den SIEM-Lösungen integriert und jeweils für eine bestimmte Logquelle konzipiert. Herausfordernd wird es, wenn es sich um unstrukturierten Text oder ein generisches Logformat wie Syslog handelt. Im ersten Fall kann oft nur das Raw-Log gespeichert werden, im zweiten Fall werden oftmals nur die Standardfelder erkannt [13].

Korrelationsregeln

Viele SIEM-Lösungen bieten ein vorgegebenes Set an Korrelationsregeln, welche dabei helfen sollen Bedrohungen frühzeitig zu erkennen. Diese sind allerdings meist sehr generisch, wodurch es an dem/der Analysten/in liegt, entsprechende Regeln auszuwählen, anzupassen und zu aktivieren. Dazu muss allerdings auch bekannt sein, wonach gesucht werden soll. Da sich Angriffe kontinuierlich weiterentwickeln und die Vorgehensweisen nicht immer bekannt sind, kann nur nach, zum Zeitpunkt der Aktivierung, bekannten Angriffsmustern gesucht werden [13]. Ein weiteres Problem ist, dass das SIEM in der Lage sein muss, eine riesige Menge an Informationen von unterschiedlichsten Quellen zu verarbeiten. Aus diesem Grund sind aktuelle Systeme meist so ausgelegt, dass zuerst Kollektoren und Normalisierungsfunktionen programmiert werden, um Daten einheitlich darzustellen und danach Korrelations-Regeln entwickelt werden, um die Informationen zusammenzufassen und somit die Datenmenge zu reduzieren. Dabei entsteht das Problem, dass die Syntax über die Semantik gestellt wird und somit die Korrelations-Regeln oft unzureichend sind [14], [15]. Da diese oft auf einfache boolsche Verkettungen von Ereignissen prüfen, die auf einen bestimmten Angriffsvektor abzielen. Nur sehr wenige SIEM-Lösungen verfügen über eine integrierte fortschrittliche Korrelations-Engine, die in der Lage ist, die Abweichung und die historische Korrelation durchzuführen, die zum Beispiel für die Überprüfung nach der Erkennung von Zero-Day-Angriffen nützlich ist [4].

Unvollständige Daten

Auch, wenn ein SIEM mit einer riesigen Menge an Daten arbeitet, hat keines der Systeme alle Daten, die es bräuchte um alle Sicherheitsvorfälle zu erkennen und zu verarbeiten. Ein Grund dafür ist, dass es nicht

wirtschaftlich ist, alle benötigten Daten zu speichern und zu verarbeiten. Nicht jedes SIEM bzw. nicht alle Daten können durchgehend korreliert werden, das führt dazu, dass es nicht immer möglich ist, eine komplette Sicht über die gesamte Umgebung zu haben [4].

Einfache Speichermöglichkeiten

Viele SIEM Systeme speichern Daten standardmäßig nur über einen begrenzten Zeitraum, wie z.B. sechs Monate, da Speicherplatz oft nicht günstig ist [4]. Das hat zum Nachteil, dass wichtige Informationen vor allem in Bezug auf Advanced Persistent Threats (ATP) verloren gehen können. Flexibler könnte dies über Cloud-Speicher geregelt werden, allerdings wird darauf oft verzichtet aufgrund der Vertraulichkeit der Daten und stattdessen lieber zu On-Premise-Servern gegriffen wird [16].

Abhängigkeit vom Menschen

Grundidee einer SIEM-Lösung ist es, eine übersichtliche Darstellung von Bedrohungen zu liefern, um ihre Bedeutung besser bewerten und Maßnahmen entsprechend priorisieren zu können. Trotzdem ist es in vielen Fällen immer noch unvermeidbar, dass Personen die Analyse von Vorfällen übernehmen, Entscheidungen treffen und passende Gegenmaßnahmen setzen. Diese Vorgehensweise verlangsamt den Prozess und führt zu höheren Kosten. Zudem wird gut ausgebildetes Personal benötigt und das Vorgehen ist anfälliger für Fehler. Das sind auch die Gründe weshalb immer mehr in Richtung Automatisierung der Prozesse für das Setzen von Gegenmaßnahmen geforscht wird [4].

Eingeschränkte Berichts- und Visualisierungsmöglichkeiten

Einige der verbreiteten SIEM-Systeme haben zwar die Möglichkeit zur Erstellung von Berichten und Visualisierungen in Form von Dashboards, allerdings sind diese oftmals sehr generisch und erfüllen nicht die individuellen Anforderungen eines jeden Unternehmens [17]. Es besteht zwar meist die Möglichkeit, diesen gerecht zu werden, dies ist aber nur mit entsprechendem Zeitaufwand, Ressourcen und Know-How möglich.

2.1.5 Die richtige SIEM-Lösung

In Anbetracht der Herausforderungen ist es besonders wichtig für ein Unternehmen, sich genau zu überlegen, welche SIEM-Lösung gewählt werden soll. Dabei ist darauf zu achten, was am besten zur Unternehmensstruktur und der Bedrohungslandschaft passt. Außerdem muss es flexibel genug sein, um sich möglichen Änderungen anzupassen. Es ist daher zu empfehlen im Vorhinein Use Cases festzulegen, um

anhand dessen Voraussetzungen abzuleiten und somit die verschiedenen Lösungen hinsichtlich der Eignung für das eigene Unternehmen bewerten und vergleichen zu können.

2.1.6 Risiken bei der Anschaffung eines SIEMs

Bei der erstmaligen Anschaffung einer SIEM-Lösungen oder bei Austausch der vorhandenen gibt es einige Risiken, welche beachtet werden sollten [9]:

- Prozess zur Toolauswahl, Implementierung, Betrieb und Feinabstimmung sowie Überwachung ist nicht ausreichend definiert. SIEM-Lösungen sind mächtige Werkzeuge, müssen allerdings richtig eingesetzt werden, um ihr wahres Potenzial auszuschöpfen. Dazu gehört auch, dass es ordnungsgemäß in die Prozesse und die Infrastruktur des Unternehmens integriert wird.
- Übermäßiges Vertrauen in die eingebauten Funktionen des Tools und die Einstellung, dass qualifizierte Mitarbeiter nicht mehr gebraucht werden. Auch wenn die Möglichkeiten zur Erkennung von Bedrohungen hervorragend ist und die Funktionen zur Automatisierung der Reaktionen ebenfalls überzeugen, braucht es immer noch ein Team an guten und engagierten Leuten. Das SIEM als Tool kann die Leute nicht ersetzen, aber es kann die Fähigkeiten des Teams unterstützen und erweitern.
- Abhängigkeit vom SIEM Hersteller, dass dieser kontinuierlich die Analysefähigkeiten verbessert und Anwendungsfälle, IOCs (Indicators of Compromise) und TTP (Tactics, Techniques and Procedures) hinzufügt, damit die sich schnell veränderte Bedrohungslandschaft akkurat abgedeckt wird. Es ist gut das Know-How im Team durch die Forschung des Herstellers zu ergänzen und kann auch Zeit und Ressourcen sparen. Allerdings sollte das Team deswegen trotzdem ausreichend besetzt sein und man nicht von der Forschung des Herstellers abhängig werden.

Ebenso gibt es diverse Risiken, die jene Unternehmen betreffen, die entscheiden keine SIEM-Lösung zu verwenden [9]:

- Security Teams müssen zwangsweise mit veralteten Tools arbeiten.
- Statt mit einer zentralen Sicht, muss mit diversen Konsolen gearbeitet werden, die jeweils nur eine Teilsicht zeigen.
- Die Möglichkeit fortgeschrittene Bedrohungen zu erkennen, wird in einer Bedrohungslandschaft, die immer komplexer wird, umso schwieriger.
- Angreifer könnten die Möglichkeit haben in die Organisation einzudringen und dort oft für lange Zeit unerkannt zu bleiben.
- Es entsteht eine technologische Lücke, die intern immer schwerer zu schließen sein wird, so dass Unternehmen gezwungen sind, alle Aspekte der Erkennung von und Reaktion auf Bedrohungen an

Dienstleistungsanbieter auszulagern.

2.2 MITRE ATT&CK Framework und Cyber Kill-Chain

Das MITRE ATT&CK Framework ist eine öffentlich zugängliche Wissensdatenbank, worin beobachtete Taktiken und Techniken von Angreifer/innen gesammelt werden. Das Framework ist in drei Technologie-Domänen unterteilt, in denen Angreifer/innen operieren. Diese sind die Domänen Enterprise, Mobile und Industrial Control Systems (ICS). Basierend auf dem Framework werden Threat Models und Methoden entwickelt, die sowohl im privaten Sektor als auch im staatlichen Sektor und der Forschung verwendet werden [18], [19].
Das Framework teilt den möglichen Ablauf eines Angriffs in folgende 14 Taktiken ein.

2.2.1 Reconnaissance

Die erste Taktik ist die Phase der Aufklärung (englisch: Reconnaissance). Dabei geht es darum, Informationen zu sammeln, die für den weiteren Verlauf des Angriffs hilfreich sein können. Die Angreifer/innen verwenden dabei Techniken wodurch sie aktiv oder passiv an Informationen zu ihrem Opfer kommen, dazu gehören beispielsweise Informationen zum Personal oder zur Infrastruktur. Dieses Wissen wird dann weiterverwendet, um die nächsten Schritte des Angriffs zu planen oder auch um die Ziele nach einer erfolgreichen Kompromittierung zu priorisieren [20].
Als Beispiele zu Vorgehensweisen in dieser Phase gehören aktives Scannen der Infrastruktur des Opfers, Phishing mit dem Ziel weitere Informationen einzuholen sowie das Einholen aller öffentlich zugänglichen Informationen (z.B. Webseite).

2.2.2 Resource Development

In der Phase der Ressourcenbeschaffung (englisch: Resource Development) versuchen die Angreifer/innen jene Ressourcen zu akquirieren, die sie benötigen, um den Angriff durchzuführen. Dazu gehört, dass die Angreifer/innen Ressourcen einkaufen oder stehlen bzw. kompromittieren, um sie in weiterer Folge für das eigentliche Ziel nutzen zu können. Bei diesen Ressourcen handelt es sich meist um Infrastruktur, Benutzerkonten und bestimmte Fähigkeiten, die benötigt werden. Nach Beschaffung dieser Ressourcen werden diese in den anderen Phasen der Kill-Chain eingesetzt. Beispielsweise können gekaufte Domänen für Command and Control oder auch für Emailkonten in Phishingkampagnen im Zuge der Phase Initial Access benutzt werden [21].

2.2.3 Initial Access

Mit dieser Taktik versucht der/die Angreifer/in erstmals in das Netzwerk des Opfers einzudringen. Diese Phase besteht aus einigen Techniken, die es dem/der Angreifer/in ermöglichen in das Netzwerk einzudringen, dazu gehören beispielsweise Spear-Phishing sowie die Ausnutzung von Schwachstellen auf öffentlich zugänglichen Web Servern. Der erlangte Zugang kann entweder langfristigen Zugriff ermöglichen, wie durch die Nutzung gültiger Accounts, oder aufgrund von Passwortwechsel auf eine kurze Zeit beschränkt sein [22].

2.2.4 Execution

In der Phase der Ausführung (englisch: Execution) versucht der/die Angreifer/in Schadcode auszuführen. Die Ausführung besteht aus Techniken, die dazu führen, dass von Angreifer/innen kontrollierter Code auf einem lokalen oder entfernten System zum Laufen gebracht wird. Die Vorgehensweisen um schadhaften Code auszuführen werden oft gemeinsam mit anderen Taktiken verwendet, um höhere Ziele zu erreichen, wie beispielsweise die Auskundschaftung eines Netzwerks oder das Stehlen von Daten. Ein Beispiel für eine der Techniken ist die Nutzung eines Remote Access Tools, um ein PowerShell-Skript auszuführen, welches Remote System Discovery durchführt [23].

2.2.5 Persistence

Mit der Taktik der Persistenz (englisch: Persistence) versuchen die Angreifer/innen ihren Zugang zum Netzwerk dauerhaft zu behalten. Somit besteht diese Taktik aus Techniken, die dazu beitragen, dass die Angreifer/innen auch nach Neustart eines Geräts oder der Änderungen von Passwörtern, weiter Zugriff haben. Zu diesen Taktiken zählen jeder Zugriff, Aktion oder Änderung der Konfiguration, die dieses Ziel unterstützten, wie beispielsweise das Ersetzen von legitimen Code oder das Hinzufügen von Startup-Code [24].

2.2.6 Privilege Escalation

Mit der Taktik der Rechteerhöhung (englisch: Privilege Escalation) versucht der/die Angreifer/in höhere Berechtigungen auf Systemen oder im Netzwerk zu erlangen. Dies ist notwendig, da Angreifer/innen zwar mit unprivilegiertem Zugriff ein Netzwerk erkunden können, aber meist höhere Rechte benötigen, um ihr Ziel zu erreichen. Häufige Vorgehensweisen sind dabei die Ausnutzung von Schwachstellen, Fehlkonfigurationen oder Sicherheitslücken. Ziel ist es mindestens eine der folgenden erhöhten Rechte zu erlangen:

- SYSTEM/root Rechte

- Lokale Administratorrechte
- Benutzerkonten mit admin-ähnlichen Berechtigungen
- Benutzerkonten mit Zugriff zu bestimmten Systemen oder zur Ausführung bestimmter Funktionen

Die Techniken die zur Erlangung höherer Rechte angewendet werden, überlappen sich oft mit den Techniken der Persistenz, da Betriebssystemfunktionen, die es einem/r Angreifer/in ermöglichen, im Netzwerk zu verbleiben, in einem erhöhten Kontext ausgeführt werden können [25].

2.2.7 Defense Evasion

Die Taktik der Defense Evasion benutzen Angreifer/innen, um nicht erkannt zu werden. Zu den Techniken gehören beispielsweise die Deinstallation oder Deaktivierung von Sicherheitssoftware oder die Verschleierung bzw. Verschlüsselung von Daten und Skripts. Angreifer/innen verwenden dazu oftmals vertrauenswürdige Prozesse, um Malware damit zu verstecken. Ebenso führt MITRE ATT&CK unter diesem Punkt Querverweise zu anderen Taktiken an, wenn deren Techniken zudem zur Verschleierung beitragen [26].

2.2.8 Credential Access

Mit dieser Taktik will sich der/die Angreifer/in Zugriff zu Anmeldeinformationen (englisch: Credential Access) verschaffen, indem Namen von Benutzerkonten oder Passwörter gestohlen werden. Techniken, die dazu verwendet werden sind beispielsweise die Aufzeichnung von Tastatureingaben (englisch: Keylogging) oder das Dumpen von Anmeldeinformationen. Durch die Nutzung von legitimen Anmeldeinformationen können Angreifer/innen Zugriff zu Systemen erlangen und es erschwert zusätzlich die Erkennung. Zudem kann es den Angreifer/innen die Möglichkeit geben, weitere Accounts zu erstellen, die zur Erreichung ihres Ziels hilfreich sein können [27].

2.2.9 Discovery

Mit der Taktik der Erkundung (englisch: Discovery) will der/die Angreifer/in herausfinden, wie die Systeme und das Netzwerk des Opfers aufgebaut sind. Das hilft den Angreifer/innen dabei die Umgebung zu beobachten, um anschließend zu entscheiden, wie weiter gehandelt werden soll. Die Angreifer/innen bringen so auch in Erfahrung was durch sie kontrolliert werden kann und was sich rund um ihren Eintrittspunkt befindet und wie sie das zur Erreichung ihres Ziels nutzen können. Zur Sammlung dieser Informationen werden oft native Tools des Betriebssystems verwendet [28].

2.2.10 Lateral Movement

Die Taktik des Lateral Movement wird von Angreifer/innen angewandt, um sich durch das Netzwerk zu bewegen, um in weitere Systeme einzudringen und zu kontrollieren. Um das Ziel zu erreichen, müssen die Angreifer/innen häufig mehrere Systeme und Konten durchlaufen. Angreifer/innen installieren dafür ihre eigenen Fernzugriffs-Tools, um Lateral Movement zu betreiben, oder nutzen legitime Anmeldeinformationen mit nativen Netzwerk- und Betriebssystem-Tools, die die Erkennung erschweren könnten [29].

2.2.11 Collection

Bei der Sammlung (englisch: Collection) sammeln die Angreifer/innen Daten, die für ihr Ziel von Interesse sind. Das Sammeln von Informationen umfasst die Techniken, die der Gegner zum Sammeln von Informationen verwenden kann, sowie die Quellen, aus denen Informationen gesammelt werden, die für die Verfolgung der Ziele der Angreifer/innen relevant sind. Meist ist das nächste Ziel das Stehlen dieser gesammelten Daten. Häufige Quellen, die Ziel dieser Taktik werden, sind verschiedene Treiber, Browser, Audio, Video und Email. Gesammelt werden diese Daten oftmals durch die Aufnahme von Screenshots und der Protokollierung der Tastatureingaben [30].

2.2.12 Command and Control

Mit der Taktik Command and Control versucht ein/e Angreifer/in mit kompromittierten Systemen zu kommunizieren um diese zu kontrollieren. Dabei versucht der/die Angreifer/in im Normalfall normalen, erwarteten Netzwerkverkehr zu generieren, um nicht erkannt zu werden. Zum Aufbau einer Command and Control Verbindung gibt es die unterschiedlichsten Techniken, welche sehr stark von der Netzwerkarchitektur und den Verteidigungsmaßnahmen des Opfers abhängen [31].

2.2.13 Exfiltration

Mit der Taktik der Exfiltrierung versucht der/die Angreifer/in die zuvor gesammelten Informationen zu stehlen. Sobald Daten gesammelt wurden, werden diese oftmals komprimiert und verschlüsselt, um zu vermeiden erkannt zu werden. Techniken, um Daten aus dem Zielnetzwerk hinaus zu befördern beinhalten oft die Nutzung des aufgebauten Command and Control Kanals oder die Nutzung eines alternativen Kanals [32].

2.2.14 Impact

Mit dieser Taktik wollen die Angreifer/innen das Zielsystem und deren Daten manipulieren, stören oder zerstören. Techniken, die zu dieser Taktik gehören beschäftigen sich mit der Störung der Verfügbarkeit oder der Kompromittierung der Integrität durch die Manipulation von Geschäfts- und Betriebsprozessen. Auch die Zerstörung oder Manipulation von Daten kann eine der angewandten Techniken sein. Die Angreifer/innen könnten auch versuchen Prozesse so zu verändern, dass diese normal aussehen, aber so verändert sind, dass sie dem Angreifer helfen, sein Ziel zu erreichen [33].

2.3 Active Directory

Das Active Directory (AD) ist ein, von Microsoft entwickeltes, Netzwerkbetriebssystem, das heute in vielen Unternehmen verwendet wird. Ursprünglich entwickelt für Windows 2000, hat es sich über die Jahre kontinuierlich weiterentwickelt. Ein Active Directory wird verwendet um unternehmensweite Informationen über eine zentrale Stelle verwalten und global verteilen zu können. Wurden Informationen zu Benutzerobjekten, Computern, Applikationen, Drucker oder Diensten (englisch: Services) einmal zum AD hinzugefügt, können sie im gesamten Netzwerk zur Verfügung gestellt werden [34].
Jegliche gespeicherten Informationen werden im Active Directory als Objekte abgelegt, welche einer hierarchischen Ordnung unterliegen. Da jedes dieser Objekte eindeutig identifizierbar sein muss, wird ihnen zum Zeitpunkt der Erstellung ein "Globally Unique Identifier" (GUID) zugeordnet, welcher aus einer 128-bit Zahlenfolge besteht. Da diese GUID für den/die Benutzer/in nicht einfach zu merken ist, wird deshalb meist der Distinguished Name (DN) verwendet, um Objekte referenzieren zu können. Der Distinguished Name setzt sich aus dem hierarchischen Pfad zusammen, unter dem das Objekt gespeichert ist, z.B.: dc=mycompany,dc=at. Um auf ein einzigartiges Objekt zu referenzieren, muss der "Relative Distinguished Name" (RDN) verwendet werden, z.B.: cn=Administrator,dc=mycompany,dc=at [34].

2.3.1 Vorteile des Active Directory

Wie schon zuvor beschrieben wird durch die Implementierung eines ADs eine Möglichkeit zur zentralen Administration des Unternehmensnetzwerks geschaffen. Binduf *et al.* beschreiben in ihrem Paper "Active Directory and Related Aspects of Security" [35] einige der großen Vorteile für Unternehmen, die durch den Einsatz eines ADs gegeben sind. So vereinfacht ein AD die Verwaltung von Netzwerkressourcen und Sicherheitsrichtlinien und bietet so die Möglichkeit den wachsenden Anforderungen im Unternehmen gerecht zu

werden. Zudem kann damit zur Sicherheit im Netzwerk durch die Verwendung eines sicheren Authentifizierungsmechanismus mit der Möglichkeit zu Single-Sign-On (SSO) beigetragen werden [34], [35].

2.3.2 Domänen

Domänen sind der zentrale Bestandteil in jedem Active Directory. Der Aufbau einer Domäne folgt der hierarchischen Struktur nach dem X.500-Standard. Die dazugehörigen Domänendienste ermöglichen die Kommunikation zwischen den Objekten, das Speichern der Daten sowie die Such- und Login-Funktionen. Einige weitere Dienste sorgen dafür, dass die Operationen im AD und in der Domäne reibungslos funktionieren können. So erhält die Domäne durch den Domain Name Service (DNS) einen eindeutigen Namen und sorgt für die Namensauflösung des definierten Bereichs. Weiteres können der Zertifikatsdienst, Federation Services sowie Richtlinien (englisch: Policies) zur Sicherheit der Umgebung beitragen. Durch den Zertifikatsdienst wird eine sichere Ausstellung, Verwaltung und Verteilung von Zertifikaten gewährleistet und die Federation Services ermöglichen Single-Sing-On für die Benutzer/innen. Durch Richtlinien können Funktionen und Rechte für Benutzerobjekte und Computer zentral geregelt werden, wodurch das Least-Privilege-Prinzip verfolgt werden kann. Durch Lightweight-Directory Services wird die Möglichkeit geschaffen, dass sich Benutzer/innen in Anwendungen, welche das Open(LDAP)-Protokoll unterstützen, ebenfalls mit den Credentials ihrer AD-User/innen anmelden können [34], [36].

2.3.3 Domain Controller

Domain Controller (DC), im Deutschen häufig auch Domänen Controller geschrieben, sind essentielle Komponenten im Active Directory. Sie sind dafür verantwortlich alle Funktionen und Protokolle im AD bereitzustellen und gelten deswegen auch als besonders schützenswert. Wird ein DC in einer unsicheren Umgebung aufgebaut, bestehen die Risiken, dass ein DC physisch kompromittiert und Passworthashes für die Domäne gestohlen werden sowie das Risiko, dass die Datenbank (ntds.dit) offline modifiziert und anschließend wieder zurück gespeichert wird. Müssen trotzdem DCs in einer Umgebung aufgesetzt werden, der nicht vertraut wird, gibt es die Möglichkeit eines Read-Only Domain Controllers (RODC). Ein RODC speichert keine Passwörter, sondern muss, bei einer Authentifizierungsanfrage einen DC kontaktieren, um das Passwort zu validieren. Mithilfe der "Password Replication Policy" kann kontrolliert werden, welche Passwörter lokal am RODC gespeichert werden dürfen und welche nicht [34].

2.4 Kerberos

Kerberos ist ein Netzwerk-Authentifizierungsdienst, welcher seit Jänner 1989 verfügbar ist. Durch die Verwendung dieses Protokolls als bevorzugten Authentifizierungsmechanismus im Active Directory, hat sich Kerberos über viele Plattformen hinweg verbreitet. Aktuell wird Kerberos 5 verwendet, welches eine Weiterentwicklung des 1989 veröffentlichten Kerberos 4 ist. Kerberos 5 wurde durch einige Features und Sicherheitsverbesserungen erweitert und ist in der RFC4120 [37] dokumentiert [38].
Kerberos wird oft als sicherer, vertrauenswürdiger Dienst zur gegenseitigen Authentifizierung von Drittanbietern beschrieben, bei dem eine einmalige Anmeldung ausreicht (Single-Sign-On). Als sicher gilt das Protokoll, da niemals Passwörter im Klartext über das Netzwerk transportiert werden. Stattdessen werden sogenannte Tickets verwendet. Das sind kryptografische Nachrichten, welche nur für eine begrenzte Zeit gültig sind und die Identität des/der Benutzers/in an einen Server übermitteln, ohne dabei Passwörter zu senden oder zu speichern. Zudem reicht eine einmalige Anmeldung aus, da nach der erstmaligen erfolgreichen Authentifizierung diese Information an alle anderen Ressourcen, auf die zugegriffen wird, weitergegeben wird. Des weiteren muss sich nicht nur der/die Benutzer/in authentifizieren sondern auch der Server oder Dienst auf den zugegriffen werden soll. Auf diese Weise soll sichergestellt werden, dass die Vertraulichkeit von sensible Informationen bewahrt wird. Kerberos wird als vertrauenswürdig bezeichnet, da es mit einem zentralen Authentifizierungsserver arbeitet, dem alle Systeme und Dienste sowie Drittanbieter grundsätzlich vertrauen [38].

2.4.1 Funktionsweise

Die drei As, Authentifizierung, Autorisierung und Auditing sind entscheidende Bestandteile der Netzwerksicherheit und somit auch für Kerberos [38].

Authentifizierung

Die Authentifizierung ist der Vorgang, bei dem die Identität eines/r bestimmten Benutzers/in verifiziert wird. Dazu wird der/die Benutzer/in nach Informationen gefragt, die in zumindest eine der drei Kategorien, auch Faktoren genannt, fallen. Der erste Faktor ist etwas, das der/die User/in weiß. Dabei handelt es sich um die am weitesten verbreitetste Option für die Authentifizierung. Das könnte zum Beispiel ein geheimes Passwort sein, welches beim erstmaligen Zugriff erstellt wird [38]. Der zweite Faktor kann etwas sein, dass man besitzt, zum Beispiel eine Schlüsselkarte oder eine zusätzliche Authentifizierungs-App am Handy, auf der man die Anmeldung bestätigen muss. Der dritte Faktor besteht aus biometrischen Eigenschaften des men-

schlichen Körpers. Dabei kann die Anmeldung beispielsweise über den Fingerabdruck oder durch einen Gesichtsscan bestätigt werden [38], [39].

Aktuelle Best-Practice ist die Multi-Faktor Authentifizierung, dabei müssen zumindest zwei der oben genannten Faktoren zum Einsatz kommen und erfolgreich sein um eine Authentifizierung zu ermöglichen. Beispielsweise sieht der Authentifizierungsvorgang so aus, dass der/die Benutzer/in das Passwort eingeben muss, über eine Authentifizierungs-App am Handy die Anmeldung bestätigen muss und für diese Bestätigung oftmals noch einen biometrischen Faktor oder einen Pin benötigt [39].

Meldet sich ein/e Benutzer/in nun lokal oder remote an einem Client über Kerberos an, dann wird das als interaktives Logon bezeichnet. Dabei erfolgt die Authentifizierung meist über eine der oben beschriebenen Methoden. Meldet sich ein/e Benutzer/in oder ein Computer aus dem Netzwerk an, spricht man von einem Netzwerk Logon. Dabei gibt es folgende Mechanismen zur Authentifizierung [40]:

- Kerberos Version 5 Protokoll
- Public Key Zertifikate
- Secure Sockets Layer/Transport Layer Security (SSL/TLS)
- Digest-Zugriffsauthentifizierung
- NT LAN Manager (NTLM)

Autorisierung

Bei der Autorisierung geht es darum, ob ein/e Benutzer/in basierend auf seiner Identität, eine bestimmte Ressource im Netzwerk verwenden darf oder nicht. Die Autorisierung erfolgt meist durch Access Control Lists (ACL), wodurch einem/r bestimmten Benutzer/in, bestimmte Rechte zugeordnet werden. Dafür ausschlaggebend sind unter anderem auch Informationen über die Gruppenzugehörigkeit des/der Benutzers/in sowie Benutzerrichtlinien. Ausschlaggebend für eine korrekte Autorisierung ist dabei ein zuverlässiger Authentifizierungsmechanismus [38].

Auditing

Bei dem letzten Punkt, Auditing, werden die Ergebnisse aus der Authentifizierung und der Autorisierung in einem Audit Log festgehalten. Im Audit Log sind diese Aktivitäten somit aufgezeichnet und können überprüft werden. Im Gegensatz zur Authentifizierung und Autorisierung handelt es sich beim Auditing um keine präventive, sondern eine reaktive Maßnahme. Auditing verhindert nicht, dass ein Angreifer in das

System eindringt, aber es sorgt dafür, dass nachvollzogen werden kann, wann, wo und wie es geschehen ist [38].

2.4.2 Tickets

Findet die Authentifizierung über Kerberos statt, so basiert diese auf Tickets. Ein Kerberos Ticket ist eine verschlüsselte Datenstruktur, die vom Key Distribution Center (KDC) ausgestellt wird. Das KDC enthält eine Datenbank mit allen Objekten mitsamt ihren Passwörtern für die jeweilige Umgebung [38]. Zudem besteht das KDC aus zwei Subservices, dem Authentifizierungsservice (AS) und dem Ticket-Granting-Service (TGS) [36].

Der Authentifizierungsservice stellt verschlüsselte Ticket-Granting-Tickets (TGT) aus, wenn sich der/die Benutzer/in einloggen möchte. Da das Ticket mit dem Passwort des/der Benutzers/in verschlüsselt ist, kann es nur vom Benutzer oder von der Benutzerin entschlüsselt werden. Das TGT kann dann verwendet werden, um verschiedene Service Tickets anzufordern. Somit ist das TGT der entscheidende Bestandteil des Single-Sign-Ons. Diese Service-Tickets werden vom Ticket-Granting-Service ausgestellt. Um dieses Ticket zu erhalten, müssen sowohl das TGT als auch das Service, das benutzt werden soll, übermittelt werden. Der TGS validiert das TGT und stellt das Service-Ticket aus, sofern dieses gültig ist [38].

Jedes ausgestellte Ticket besteht aus den folgenden Feldern [38]:

- Benutzerprinzipalname (englisch: User Principal Name, UPN) [41] des anfragenden Nutzers
- Dienstprinzipalname (englisch: Service Principal Name, SPN) [41] des angefragten Services
- Start und Ende der Gültigkeit des Tickets
- Liste von IP-Adressen, von wo aus das Ticket genutzt werden darf
- Gemeinsamer geheimer Schlüssel (Session Key) zur Verschlüsselung von Benutzer/innen- und Applikationskommunikation

Manche Felder werden vom KDC ausgefüllt, wie z.B. die Gültigkeitsdauer und der Session Key. Dieser wird neu generiert, sobald ein Ticket ausgestellt wird. Die übrigen Felder werden vom anfragenden Client ausgefüllt, wenn ein Ticket angefragt wird [38].
In Abbildung 2.3 wird abgebildet, wie die Kerberos Authentifizierung mithilfe der Tickets abläuft [38], [42].
Im Detail erfolgen dabei folgende Schritte im Zuge einer Authentifizierung mit Kerberos:

1. Das Benutzerkennwort wird vom Client gehasht. Dieser Hash wird als geheimer Schlüssel verwendet, um die Kommunikation zwischen dem Client und dem KDC zu verschlüsseln.

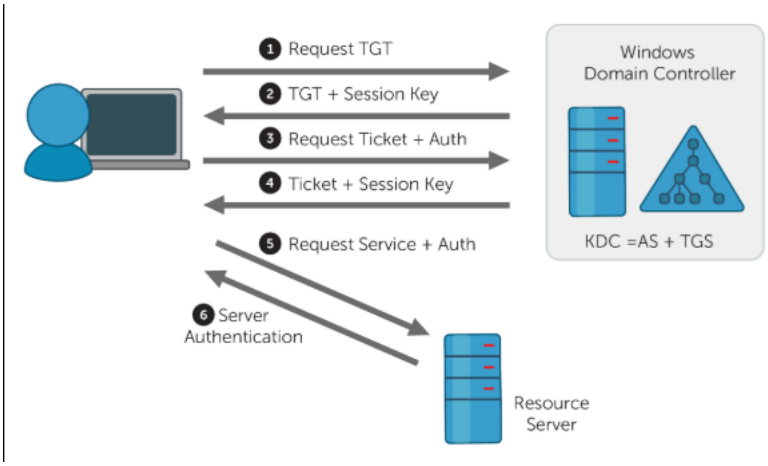

Abbildung 2.3: Ablauf der Kerberos Authentifizierung [42]

Der Client sendet daraufhin einen, mit diesem geheimen Schlüssel, verschlüsselten Zeitstempel an den Authentifizierungsservice, welcher auf diese Weise verifizieren kann, dass der/die Benutzer/in das richtige Kennwort eingegeben hat. Dazu entschlüsselt der AS den Zeitstempel mit dem Hash des Benutzerkennworts aus der AD Datenbank. Wenn der empfangene Zeitstempel dem entschlüsselten entspricht, wurde das Kennwort korrekt eingegeben [43].

Dieses vom Client gesendete Paket wird Authentication Service Request (AS-REQ) genannt und ist notwendig für die Kerberos-Vorauthentifizierung (englisch: Kerberos Pre-Authentication). Nur wenn dieser Schritt erfolgreich durchgeführt werden konnte, kann der Vorgang fortgesetzt werden [34].

2. Daraufhin antwortet der AS mit dem Schlüssel, der für die weitere Kommunikation mit dem KDC benutzt wird (Session Key). Dieser Schlüssel ist wieder verschlüsselt mit dem Hash des/der Benutzers/in. Außerdem wird das TGT mitgeschickt. Dieses ist verschlüsselt mit dem Kennwort des krbtgt-Accounts, welches nur der TGS kennt [43].

   Dieses Paket wird als Authentication Service Response (AS-REP) bezeichnet [34].
3. Das erhaltene TGT kann im nächsten Schritt dazu verwendet werden, einen bestimmten Service anzufragen. Dazu wird das TGT im sogenannten Ticket Granting Service Request (TGS-REQ) an den TGS geschickt [34], [43].
4. Der TGS entschlüsselt das TGT sowie die Anfrage, verifiziert sie und sendet die Antwort in der Ticket Granting Service Response (TGS-REP) zurück [34]. Diese besteht zum einen aus dem Service-Ticket, welches mit dem geheimen Schlüssel des Services verschlüsselt ist und für den Ziel-Server bestimmt ist. Darin befinden sich die Gruppeninformationen des/der Benutzers/in, sowie ein Sitzungsschlüssel für die Kommunikation zum Client und ein Zeitstempel. Außerdem schickt der TGS den Sitzungs-

schlüssel für den/die Benutzer/in retour, der für die Kommunikation zwischen Client und Server bestimmt ist. Dieser ist verschlüsselt mit dem Schlüssel aus Schritt 3 [43].

5. Nun kann der Client eine Anfrage an den Server schicken, indem er das Service-Ticket im Application Request (AP-REQ) übermittelt [34], [43].
6. Der Server akzeptiert das Ticket, wenn er das Ticket mit dem Sitzungsschlüssel erfolgreich entschlüsseln kann. Ist das der Fall bedeutet das, dass die Serviceanfrage vom KDC autorisiert wurde [43]. Um das zu bestätigen sendet der Server eine in manchen Fällen eine Application Response (AP-REP). Dies ist allerdings optional und nicht zwingend erforderlich [34].

2.4.3 Sicherheitsprobleme im Kerberos Protokoll

Eines der Hauptprobleme betreffend der Sicherheit des Kerberos-Protokolls ist die Zustandslosigkeit des Protokolls. Das bedeutet, dass der KDC sowie die Subservices AS und TGS keine Informationen über vorhergehende Sitzungen speichern. Jegliche Information, die der TGS benötigt, um eine Anfrage zu bearbeiten, wird im TGT mitgegeben [44]. Da das TGT mit dem Passwort des krbtgt-Accounts verschlüsselt ist, können es im Normalfall nur der AS, der das TGT ausstellt und der TGS, der das Service-Ticket nach Erhalt des TGTs ausstellt, entschlüsseln. Das führt dazu, dass das Passwort des krbtgt-Accounts das wichtigste Passwort in einem AD ist und, dass jeglicher Information im TGT vertraut wird, sofern es mit dem krbtgt Passwort verschlüsselt ist. Außerdem wird das Passwort für gewöhnlich selten geändert und ohne Ablaufdatum konfiguriert. Das heißt, wenn ein Angreifer dieses Passwort erlangt, dass das System oft für einen längeren Zeitraum angreifbar ist, bis das Passwort geändert wird [36].

2.5 Windows Event Log

Alle Aktivitäten, die auf Windows Geräten auftreten, wie beispielsweise Authentifizierungsanfragen oder das Ausführen von Prozessen wird als Event Logs lokal am Computer aufgezeichnet. Die Event Logs sind ein standardmäßig in Windows integriertes Loggingsystem und können in verschiedene Kategorien unterteilt werden, wo jedes Event eine eigene einzigartige ID besitzt. In der AD Umgebung werden Logs, die die Authentifizierung betreffen, einheitlich in den Domain Controller Event Logs gesammelt, andere Logs, wie die Ausführung von Prozessen, befinden sich nur lokal auf dem jeweiligen Computer [45]. Eine genauere Beschreibung der Windows Event Logs ist in Unterkapitel 6.3.2 zu finden.

2.6 Potentielle Sicherheitsrisiken

Das Logging verschiedenster Ereignisse alleine hilft dabei nicht, die Sicherheitsrisiken, die in der AD Umgebung vieler Unternehmen vorhanden sind, zu minimieren. Binduf *et al.* [35] beschreibt in einem Paper die häufigsten Sicherheitsprobleme im Active Directory. Dazu gehört, dass Service Accounts oft überpriviligiert sind. Es muss genau darauf geachtet werden, welche Rechte ein Service Account braucht und welche nicht. Hat der Service Account administrative Rechte und werden dessen Credentials gestohlen, kann die Domäne kompromittiert werden. Gleichzeitig sind in vielen Unternehmen zu viele Domänen Administratoren/innen vorhanden. Ein/e Domänen Administrator/in hat alle administrativen Rechte betreffend Workstations, Server, Domain Controller und Gruppenrichtlinien, wodurch diese Rolle mehr Rechte hat, als ein Account für sich haben sollte. Im Idealfall sollte die Domänen Admin Gruppe nicht genutzt werden und jedem/jeder Benutzer/in die Rechte zugewiesen werden, die sie tatsächlich benötigen. Außerdem wird aus Bequemlichkeit oftmals das gleiche Passwort für lokale Administratorkonten vergeben. Das hat allerdings zur Folge, dass sobald ein/e Angreifer/in dieses Passwort bekommt, diese/r Zugriff auf alle Clients in der Domäne hat [35].

2.7 Angriffe auf das Active Directory

Durch die weitläufige Nutzung des Active Directory in vielen Unternehmen und Organisationen, hat es sich zu einem begehrten Ziel für Angreifer/innen entwickelt. Zudem stellt es die zentrale Stelle für die Zugriffsverwaltung bereit, wodurch es für Angreifer/innen zusätzlich interessant wird [36].
Viele Angriffe beginnen damit, dass ein einzelner Computer infiziert wird und sich der/die Angreifer/in danach höhere Rechte verschafft, wie z.B. einen Domänen Administrator. Diese werden besonders gerne dafür verwendet, da sie die Rechte haben, auf beinahe alle Ressourcen in der Organisation zuzugreifen. Die Angreifer/innen probieren in weiterer Folge sich auf mehrere Varianten im Netzwerk festzusetzen, um dort möglichst lange zu verweilen [36].

2.7.1 Vorgehensweise bei einem Angriff

Das in Kapitel 2.7 beschriebene Vorgehen kann in Phasen eingeteilt werden. Diese Phasen werden auch als Cyber Kill-Chain bezeichnet und schaffen die Grundlage dafür bestimmte Vorgehensweisen eines Angriff in einen zeitlichen Rahmen einzuordnen (siehe Kapitel 2.2).
Angriffe auf das Active Directory erfolgen für gewöhnlich in bestimmten Stufen bzw. folgen sie einem bestimmten Weg. Im Folgenden wird die Vorgehensweise dieser Angriffe dargestellt, dazu wird vorausgesetzt,

dass der Angreifer bereits in das Netzwerk eingedrungen ist und sich dort festgesetzt hat [46]. Sobald ein/e Angreifer/in im System fest verankert ist und somit Zugriff zu einem Computer oder einem/r User/in im AD hat, wird begonnen die Objekte der Domäne zu durchforsten (englisch: Enumeration). Dadurch kann sich der/die Angreifer/in alle Informationen herausziehen, die er benötigt um sich in der Domäne weiter zu bewegen und seine Rechte zu erhöhen [36]. Diese Fortbewegung in der Domäne ist im Englischen auch besser bekannt als Lateral Movement.
Danach beginnt die nächste Stufe, in der die Angreifer/innen nach lokalen bzw. Admin-Rechten suchen. Sobald sie einen Weg gefunden haben diese zu erhalten, wird nach Möglichkeiten gesucht, sich in der Umgebung so lang wie möglich festzusetzen um in weiterer Folge deren Ziele erfüllen zu können. Die Ziele eines/r Angreifers/in sind zum Beispiel das Exfiltrieren von Daten, Denial-of-Service (DoS) oder ein anderes Ziel. Die beschriebenen Stufen werden in Abbildung 2.4 dargestellt [36].

Das Ziel eines jeden/r Angreifers/in ist in den meisten Fällen entweder Zugriff auf Daten zu bekommen, diese zu exfiltrieren und im Netzwerk zu verweilen. Um diese Ziele zu erreichen gehört auch das Erhöhen der Rechte in vielen Fällen zu einem der Ziele [47].

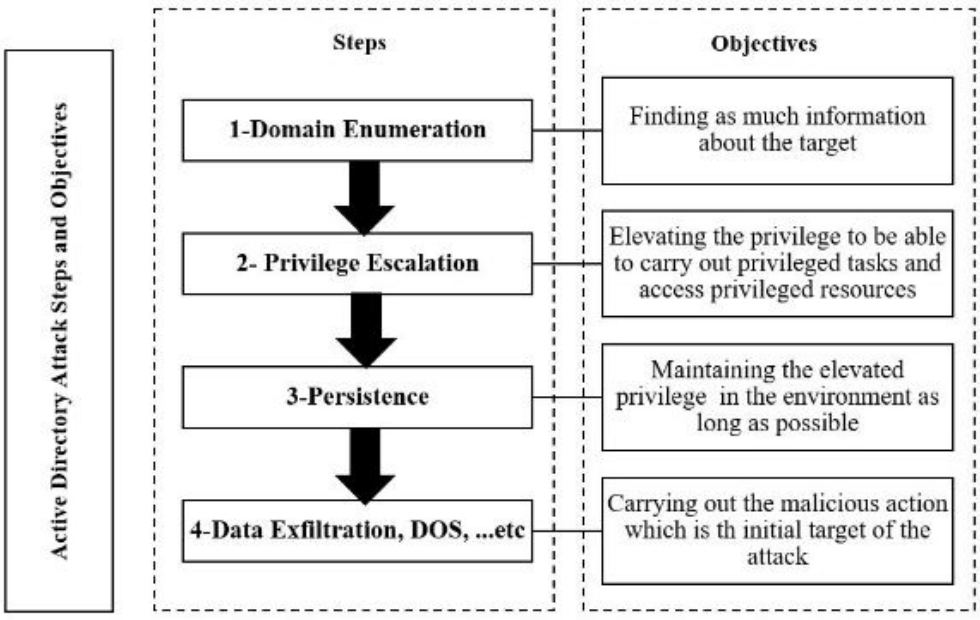

Abbildung 2.4: Phasen eines AD Angriffs [36]

Domain Enumeration

Das Ziel der Angreifer/innen, die sich bereits im System festgesetzt haben, ist im Normalfall deren Rechte zu erhöhen. Dazu haben sie es vor allem auf privilegierte Benutzerkonten wie Domänen Administratoren oder Organisations Administratoren (englisch: enterprise admins) abgesehen. Um diese Accounts zu finden bzw. ausnutzen zu können, führt der/die Angreifer/in Domain Enumeration durch. Dabei wird versucht Schwachstellen zu finden, die ausgenutzt werden können. Einer der Nachteile des ADs ist es, dass diese Enumeration im Normalfall ohne erhöhte Rechte und von jedem/r User/in durchgeführt werden kann [48]. Dazu können auch eine Vielzahl an Tools verwendet werden, zu diesen gehören [36]:

- Net.exe: Net.exe ist ein in Windows eingebautes Tool, das dazu verwendet werden kann, um verschiedenen Tätigkeiten an Gruppen, Benutzerobjekten, Accounts oder Richtlinien auszuführen. Mit diesem Tool kann der/die Angreifer/in alle Attribute eines Benutzerobjekts oder einer Gruppe einsehen und nach vulnerablen Gruppen, wie z.B. den Domänen-Administratoren, suchen und die jeweiligen Mitglieder einsehen [49].
- Active Directory Modul: Im Active Directory ist es Administrator/innen möglich Abfragen und Änderungen über die PowerShell (PS) durchzuführen. Dazu gibt es eine Sammlung an PowerShell-Befehlen (englisch: cmdlets) mit denen eine große Menge an Informationen über die Domäne abgerufen werden kann. Um dieses Modul zu verwenden, müssen im Normalfall die Remote Server Administrator Tools (RSAT) installiert sein [50]. Allerdings gibt es auch Möglichkeiten für den/die Angreifer/in diese Tools auf einem Domänen-Computer zu installieren.
- Powerview: Powerview[1] ist ein PowerShell Tool, welches zum Powersploit Framework gehört. Es kann an Stelle des net-Befehls verwendet werden und bietet die gleiche Funktionalität, wie das AD Modul. Zusätzlich hat es allerdings ein paar Funktionen, die der/die Angreifer/in nutzen kann um zu identifizieren wo sich bestimmte Nutzer/innen anmelden [51], [52].
- Bloodhound: Bloodhound[2] ist ein GUI-basiertes Tool, welches dazu genutzt wird um AD Domänen zu visualisieren und komplexe Angriffspfade identifizieren zu können. Mit Bloodhound ist es möglich ein tiefes Verständnis von privilegierten Verbindungen in einem AD zu erhalten [53], [54].

Unabhängig davon, welches Tool zur Domain Enumeration verwendet wird, der erzeugte Output ist meist sehr umfassend und dementsprechend groß ist auch der Umfang an Informationen die daraus abgeleitet werden können, abhängig davon wie viel Zeit dem/der Angreifer/in dazu zur Verfügung steht. Wichtige Informationen für Angreifer/innen betreffen die aktuelle Domäne und dazugehörenden Forests, privilegierte

[1] https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1
[2] https://github.com/BloodHoundAD/BloodHound

Gruppen, wie Domänen Admins und Enterprise Admins, bestimmte interessante Einträge in Access Control Lists (ACL), die Struktur der Domäne, angewandte Gruppenrichtlinien, sensible Dateien sowie Clients mit aktiven Sitzungen von bestimmten ausgewählten Benutzer/innen [55].

Erhöhung der Rechte

In der zweiten Stufe eines erfolgreichen AD Angriffs wird versucht, die Rechte eines lokalen Administratorkontos zu erlangen um in weiterer Folge zum Domänen Administrator aufsteigen zu können. Im folgenden werden einige Techniken und Angriffsvektoren beschrieben, die bei erfolgreicher Durchführung dazu führen, die gesamte Domäne kontrollieren zu können [36].

Pass the Ticket (PTT) Bei einem PTT-Angriff verkörpert der/die Angreifer/in eine/n Benutzer/in, indem er dessen/deren Kerberos Token stiehlt und diesen benutzt um sich zu authentifizieren. Die Token können oftmals mithilfe von Tools, wie Mimikatz[3], aus dem Cache des kompromittierten Systems ausgelesen werden. Der Token ist standardmäßig 600 Minuten gültig. Das entspricht dem Zeitrahmen indem der/die Angreifer/in den Token nutzen kann, um beim KDC ein TGT anzufordern. Mit diesem kann der/die Angreifer/in dann weitere Tickets anfordern, zu denen der/die jeweilige Benutzer/in von dem der Token stammt berechtigt ist [56].
Die beste Option um diese Angriffe zu entdecken ist auf die Wiederverwendung von Authentifizierungstoken zu achten, im speziellen, wenn diese durch andere Systeme oder IP-Adressen verwendet werden. Das kann durch das Auditieren von Kerberos-Datenverkehr am Domain Controller oder auf den Endpunkten durch Endpoint Detection and Response (EDR) Systeme erfolgen [56].

"Pass the Hash"-Angriff Der "Pass the Hash"-Angriff (PTH) hat sich über die Jahre kontinuierlich weiterentwickelt. Microsoft implementiert mit jeder neuen Windows Version Sicherheitsmaßnahmen, um diese Angriffe zu verhindern, allerdings werden gleichzeitig neue Methoden entwickelt diesen Angriff erfolgreich auszuführen. Der Angriff basiert darauf einen Hash von einem kompromittierten Client zu extrahieren und mit diesem Tickets zu erstellen, um Zugriff auf sensible Ressourcen oder Computer in der Domäne zu erlangen [36].
Windows Hostsysteme speichern Anmeldeinformationen in der Form von NT Hashes. Bei PTH-Angriffen nutzen Angreifer/innen die Tatsache, dass NTLM-Hashes für die Authentifizierung verwendet werden können, ohne, dass das Benutzerkennwort benötigt wird. Der Angriff besteht aus zwei Stufen. Zuerst muss

[3] https://github.com/gentilkiwi/mimikatz

der Hash von bestimmten Benutzer/innen je nach Verfügbarkeit extrahiert werden. Danach können damit Tickets bzw. Token mithilfe von Tools, wie Mimikatz, erstellt werden [57]. Um möglichst hohe Rechte zu erlangen, probieren Angreifer/innen die Hashes von möglichst privilegierten Konten zu erlangen. Das häufigste Vorgehen für die Stufe 1 ist es, einen Dump der Anmeldeinformationen aus dem Speicher des LSASS (Local Security Authority Subsystem Service) Prozess zu erstellen. LSASS speichert alle Informationen inklusive dem NT-Hash von allen Konten, die aktiv genutzt werden [58]. Im Normalfall werden die Hashes gespeichert, um das Single-Sign-On für die Benutzer/innen zu ermöglichen. Das heißt aber auch, dass wenn Angreifer/innen den Hash eines Domänen-Administrators extrahieren können, diese dann auf alle Ressourcen zugreifen können, zu denen der Domänen Admin Zugriff hat, ohne dazu das Passwort zu kennen [36]. Weitere Methoden um an NTLM Hashes zu kommen wären das Mitschneiden und Extrahieren aus dem Datenverkehr sowie die Extrahierung aus der SAM Account Datenbank [56].
Die sicherste Maßnahme, um gegen PTH-Angriffe vorzugehen, wäre den Speicher des LSASS Prozesses zu schützen und Zugriff zu verhindern. Diese Maßnahme wurde mit dem Windows Defender Credential Guard (WDCG) implementiert, indem der Zugriff auf den LSASS Prozess für bösartige Applikationen untersagt wird [56]. Auf diese Weise können die Hashes nicht mehr aus dem LSASS Speicher gelesen werden, allerdings können andere Methoden genutzt werden, um an die Hashes zu kommen [59].
Da der Angriff auf den Domänen bzw. System Administratorenrechte basiert, sollten diese Benutzerkonten besonders geschützt werden. Dazu können beispielsweise Privileged Access Management (PAM) Systeme benutzt werden. Zudem sollte die Anzahl der benutzten Konten mit Administratorrechten eingeschränkt werden. Eine weitere Schutzmaßnahme kann implementiert werden, indem NTLMv2 Authentifizierungen erzwungen werden und LanMan bzw. NTLM untersagt wird [56].

Overpass the Hash (OPTH) Bei diesem Angriff wird der PTH-Angriff auf die nächste Stufe gehoben, indem der gestohlene Passwort-Hash dazu verwendet wird, einen Kerberos Benutzer-Authentifizierungs-Token zu erstellen, um Zugriff zu dessen/deren Benutzerkonto zu erlangen.
Um gegen diesen Angriff vorzugehen, können die Sicherheitsmaßnahmen aus den PTH und PTT-Angriffen kombiniert werden [56].

Kerberoasting Angriff Kerberoasting wird ein Angriff genannt, der darauf basiert, dass manche Dienste unter normalen Benutzerkonten laufen. Wenn ein Service unter einem normalen Benutzerkonto läuft, ist das Service Ticket für diesen Service mit dem NTLM-Hash des Benutzerkennworts verschlüsselt. Greift ein/e Angreifer/in dieses Service Ticket ab, kann das Klartext-Passwort des/der Benutzers/in mittels

Brute-Forcing erlangt werden [36].

Über das SPN Attribut eines/r Benutzers/in kann erkannt werden, ob ein Dienst unter diesem Konto läuft. Der Ablauf eines Kerberoasting Angriffs erfolgt in vier Stufen [36]:

1. Die Angreifer/innen scannen die AD Umgebung nach Benutzer/innen, die ein SPN Attribut definiert haben.
2. Es wird ein Service Ticket mit dem im vorigen Schritt identifizierten SPN beim AD angefragt.
3. Das erhaltene Ticket wird exportiert.
4. Über Brute-Forcing wird das Klartext-Passwort des/der Benutzers/in enthüllt.

Neben dieser Vorgehensweise gibt es noch zwei weitere Varianten des Kerberoasting-Angriffs. Bei der ersten fängt der/die Angreifer/in die AS-REP Nachricht von Kerberos ab. Diese ist ebenfalls mit dem Hash des Benutzerkennworts verschlüsselt. Allerdings erfordert diese Variante die Einstellung "Don't require Kerberos pre-authentication" in der Benutzerkontensteuerung (englisch: User Account Control (UAC)). Die zweite Variante ist nur möglich, wenn der/die Angreifer/in bereits höhere Rechte hat. In diesem Fall kann er/sie den SPN-Wert eines/r beliebigen Benutzers/in auf einen beliebigen Service setzen und anschließend ein Service-Ticket anfragen, um auf diese Weise in der oben beschriebenen Art das Klartext-Passwort dieses/r Benutzers/in zu erhalten [36].

Um diesen Angriff vorzubeugen ist es wichtige eine komplexe Passwortrichtlinie für Service-Accounts implementiert zu haben. Außerdem sollte Kerberosverschlüsselung statt RC4 verwendet und die Rechte für Servicekonten eingeschränkt werden [60]. Zudem kann die Event-ID 4769 (Anfrage eines TGS) überwacht werden. Das Passwort des Service Accounts sollte mindestens 25 Zeichen haben, um es schwer knacken zu können. Außerdem sollten (Group) Managed Service Accounts verwendet werden [47].

Mithilfe von Machine-Learning kann dieser Angriff auch erkannt werden, indem eine Abweichung von Service Ticket Anfragen erkannt wird. Zur Erkennung des Angriffs können auch Honey-Token-Accounts unter die legitimen Servicekonten gemischt werden. Honey-Token-Accounts werden nicht genutzt und haben keine Rechte. Sie dienen lediglich dazu eine Warnung zu senden, wenn sie genutzt werden, um einen möglichen Angriff festzustellen [56].

Ausnutzung von Kerberos-Delegation Die Kerberos-Delegation ist eine Funktion im Active Directory, die es Benutzer/innen oder Computern ermöglicht einen Account zu übernehmen. Um die Funktion nutzen zu können, muss die Einstellung im Benutzer- oder Computerkonto aktiviert werden. Es gibt dabei zwei Arten der Delegation: eingeschränkt (englisch: constrained) und uneingeschränkt (englisch: unconstrained) [61].

Unconstrained Delegation

Mit der uneingeschränkten Delegation darf ein Computer jeden Service verkörpern. Wurde diese Einstellung aktiviert, hat der/die Angreifer/in die Möglichkeit einen privilegierte/n Benutzer/in dazu zu bringen sich auf dem jeweiligen Computer einzuloggen, damit dessen TGT gespeichert wird. Im nächsten Schritt kann der/die Angreifer/in das TGT nutzen, um auf jeden beliebigen Service in der Domäne zuzugreifen [36]. Neben diesem einfachen Angriff, gibt es noch mehrere Optionen, wie diese Funktion missbraucht werden kann.

Ein/e Angreifer/in kann Powerview oder das AD Modul verwenden, um herauszufinden, welche Benutzer/innen zur Delegation berechtigt sind und welche Computer Delegation aktiviert haben. Um den Missbrauch von Delegationen zu verhindern, sollten ausschließlich eingeschränkte Kerberos-Delegation verwendet werden. Außerdem können privilegierte Benutzer/innen einer Gruppe hinzugefügt werden, die es nicht erlaubt, dass deren TGTs zur Delegation verwendet werden [36].

Constrained Delegation

Im Gegensatz zur uneingeschränkten Delegation werden bei der eingeschränkten Delegation bestimmte Services definiert, die verwendet werden dürfen. Trotzdem kann auch die eingeschränkte Delegation von Angreifer/innen missbraucht werden. Ein/e Angreifer/in kann zum Beispiel einen Account übernehmen, der die eingeschränkte Delegation aktiviert hat und in weiterer Folge dessen Hash extrahieren. Der/die Angreifer/in kann somit ein TGT und danach ein TGS anfordern, womit er/sie auf jeden, in der Delegation spezifizierten, Service zugreifen kann. Da der SPN nicht validiert wird, ist es dem/der Angreifer/in möglich für jeden Service, der unter dem übernommenen Account läuft, ein TGS anzufordern. Das bedeutet, wenn zum Beispiel der CIFS Services in der Liste der erlaubten Delegationen eines übernommen Accounts ist, dann kann der/die Angreifer/in auf jeden Dienst zugreifen, der unter dem selben Account läuft, wie dieser Service. Das führt dazu, dass über diesen Angriffsvektor eine gesamte Domäne übernommen werden könnte [36]. Verhindert werden kann dies nur, indem für privilegierte Konten die Einstellung getroffen wird, dass diese nicht delegiert werden können.

Domänenpersistenz

Nachdem Angreifer/innen ihre Rechte erhöht haben, ist es ihr Ziel so lange wie möglich mit diesen Rechten in der Domäne verweilen zu können. Diese Phase wird im Englischen auch (Domain)-Persistence genannt. Es gibt viele verschiedene Techniken und Möglichkeiten, wie Angreifer/innen dies erreichen können. Im

folgenden werden die häufigsten Vorgehen beschrieben [36].

Golden Ticket Der Golden Ticket-Angriff wird möglich, da der KDC die Gültigkeit eines TGTs durch das Entschlüsseln mit dem krbtgt-Hash durchführt und bei erfolgreicher Entschlüsselung die Service Tickets für die angefragten Dienste ausstellt. Der Inhalt des TGT wird dabei nicht verifiziert, solange dieses nicht älter als 20 Minuten ist. Somit kann ein Angreifer der bereits Domänen Administratorrechte besitzt, den krbtgt-Hash extrahieren und mithilfe dessen TGTs fälschen. Er kann dabei TGTs für jedes Benutzerobjekt und Rechte über einen langen Zeitraum hinweg ausstellen. Dem Angriff zu Gute kommt, dass das krbtgt-Passwort selten geändert wird und zudem zweimal geändert werden muss, um diesen Angriff zu stoppen [62].
Einen Golden Ticket Angriff zu erkennen ist sehr schwierig, da ein valides TGT verwendet wird. Allerdings gibt es ein paar Sicherheitsmaßnahmen die getroffen werden können. Zum einen sollte die Anzahl an Benutzerkonten mit Domänen Adminrechten und andere Benutzerkonten mit hohen Rechten auf ein Minimum reduziert werden. Wie auch schon bei den Pass-the-Ticket-Angriffen, sollte darauf geachtet werden, ob ein Ticket wiederverwendet wird, das heißt von einem anderen System oder von einer anderen IP-Adresse. Die Nutzung von dynamischen Software-Whitelisting am DC kann zusätzlich verhindern, dass unautorisierte Tools verwendet werden [56].

Silver Ticket Im Gegensatz zu Golden Tickets, bei denen es sich um TGTs handelt, dreht es sich bei Silver Tickets, um Service Tickets, die dazu genutzt werden Zugriff zu allen Diensten zu erhalten. Dazu muss der/die Angreifer/in Zugriff auf den NTLM-Hash eines Service Accounts bekommen, womit dann mittels z.B. Mimikatz ein Service Ticket für diesen Service ausgestellt werden kann. Somit kann jeder beliebige Benutzer/in verkörpert werden [36]. Auch, wenn der Silver Ticket-Angriff eine eingeschränktere Anwendung hat, so ist er trotzdem gefährlich, da er wesentlich leichter auszuführen und schwieriger zu entdecken ist [47].
Um diesen Angriff bestmöglich zu verhindern, muss darauf geachtet werden, dass möglich komplexe Passwörter für Service Accounts vergeben werden. Zudem können durch die Nutzung von Privileged Attribute Certificates (PAC) die TGS Token vom krbtgt signiert werden. Das verhindert die Möglichkeit den Angriff rein durch den Zugriff und das Dumping der Credentials am lokalen System durchzuführen [56].
Sowohl bei Golden als auch bei Silver Tickets kann es vorkommen, dass sie aufgrund eines veränderten Account Domain Feldes auffallen. Im Account Domain Feld sollte die Domäne drinstehen, allerdings kann es vorkommen, dass es bei Golden oder Silver Tickets leer ist, der Domänen FQDN enthalten ist oder der

Wert ein ganz anderer ist. Event-IDs auf die dabei geachtet werden kann sind 4624 (logon), 4672 (admin logon) und 4634 (logoff) [47].

Skeleton Key Bei dem Skeleton Key-Angriff handelt es sich um einen Angriff auf den LSASS-Prozess des Domain Controllers, welcher für die gesamte Authentifizierung im AD zuständig ist. Dabei verändert der/die Angreifer/in den LSASS-Prozess so, dass ein Master-Passwort erstellt wird, welches für jedes Benutzerkonto funktioniert. Um den Angriff durchzuführen benötigt der/die Angreifer/in einen bereits kompromittierten Domain Controller mit Domänen Administratorrechten. Mit zwei einfachen Befehlen in Mimikatz kann das Master-Passwort erstellt und sich somit in jedes Benutzerkonto eingeloggt werden. Damit das reibungslos in der gesamten Domäne funktioniert, muss der Skeleton Key auf allen DCs installiert werden. Zudem dürfen die DCs nicht heruntergefahren werden, da durch einen Neustart der LSASS-Prozess-Speicher neu geladen und der Key somit nicht länger gültig ist [36].
Der Angriff ist zwar nur auf Server vor WindowsServer 2012 anwendbar, für diese stellt er allerdings ein großes Risiko dar. Um dieses Risiko möglichst zu minimieren, sollte darauf geachtet werden, nur eine begrenzte Anzahl an Domänen Administratoren zu haben und es diesen nicht ermöglichen, sich auf schwächer geschützten Maschinen einzuloggen [36], [56].
Außerdem kann der Angriff erkannt werden, indem nach schwachen Passwörtern oder Passwörtern, die nicht der Passwortrichtlinie einer bestimmten Organisationseinheit (englisch: Organizational Unit (OU)) entsprechen, gescannt wird. Auch die Detektierung von "Encryption Downgrade" während eines Logon-Prozesses kann dabei helfen diesen Angriff zu erkennen [56].

DCShadow Hat der/die Angreifer/in Zugriff auf einen DC der Domäne erhalten, kann er den DCShadow-Angriff ausführen. Der/die Angreifer/in registriert einen falschen bzw. kompromittierten Domain Controller in der AD Domäne und kann somit verschiedene Änderungen, wie z.B. falsche Anmeldeinformationen oder ACLs an die anderen DCs weiter verteilen, indem die Domänenreplikation genutzt wird [56].
Auch dieser Angriff ist schwer zu erkennen, da er von Standardfunktionen wie der Domänen Replikation Gebrauch macht. Allerdings können Schritte unternommen werden, um die Wahrscheinlichkeit eines solchen Angriffs zu reduzieren. Zum einen sollten API-Calls, vor allem für Replikations-Funktionen, von nicht DCs, überwacht werden, da es sich hierbei um bösartige Systeme handeln könnte, die vorgeben ein DC zu sein. Des weiteren kann auf vorgegebene IOCs geachtet werden, um Replikationsanfragen in der Domäne zu überwachen. Im Zuge einer weiteren Maßnahme kann darauf geachtet werden, dass der Kerberos SPN

nicht von Computern verwendet wird, die außerhalb der DC Organisationseinheit liegen [56].

DCSync Um einen DCSync-Angriff durchzuführen, braucht der/die Angreifer/in bereits Zugriff zu Anmeldeinformationen von Domänen Administrator/innen oder andere erhöhte Rechte. Der/die Angreifer/in kann dann vortäuschen ein DC zu sein, indem die gestohlenen Domänen Admin Credentials genutzt werden und der primäre DC über den Standard AD Directory Replication Service dazu aufgefordert wird, die Benutzer/innen-Anmeldeinformationen an den/die Angreifer/in zurück zu replizieren [56]. So wie auch beim DCShadow-Angriff, ist dieser Angriff sehr schwer zu detektieren. Der beste Weg um den Angriff zu erkennen ist es, die Wiederverwendung von Authentifizierungs-Tokens von anderen Systemen oder IP-Adressen zu erkennen. Aber auch Standardmaßnahmen, wie risikobasiertes Schwachstellenmanagement und Netzwerk-Monitoring kann dabei unterstützen diese Angriffe zu entdecken und zu stoppen [56].

DSRM Persistenz DSRM steht für den "Directory Services Restore Mode"-Account, welcher auf jedem DC vorhanden und mit lokalen Administratorrechten ausgestattet ist. Das Passwort dieses Accounts wird typischerweise beim Aufsetzen der Domain Controller gesetzt und nur selten geändert. Die Hauptfunktion dieses Benutzerkontos ist die Bereitstellung einer Hintertür zur Datenbank für den/die Administrator/in, welcher für Reparaturen oder Wiederherstellungen verwendet werden kann. Standardmäßig verfügt dieses Konto nicht über Zugriff zur Domäne oder ihren Diensten [63]. Allerdings kann der/die Angreifer/in den Account so konfigurieren, dass er/sie lokale Administratorrechte für den DC erlangt und somit die gesamte Domäne kompromittiert.
Dazu holt sich der/die Angreifer/in über Mimikatz die SAM Datenbank des DC und führt einen Pass-the-Hash Angriff aus, indem der NTLM-Hash genutzt wird, der benutzt wurde um ein gültiges TGT zu bekommen. Auf dieses Weise bekommt der/die Angreifer/in Zugriff auf alle Dienste in der Domäne.
Um diesen Angriff zu verhindern, kann der Registry Key "DsrmAdminLogonBehavior" auf den Wert 1 gestellt werden, um das Einloggen dieses Accounts zu verhindern [36]. Zudem kann auf Events geachtet werden, bei denen das Passwort dieses Accounts geändert wird.

Missbrauch von ACLs Jegliche Zugriffsrechte auf Objekte im AD Umfeld werden durch Access Control Lists (ACLs) und den darin enthaltenen Einträgen, den Access Control Entries (ACEs), geregelt. Ein ACE definiert die Zugriffs- bzw. Audit-Berechtigungen, die ein/e Benutzer/in oder eine bestimmte Gruppe besitzen [64]. Modifiziert ein/e Angreifer/in ACLs von bestimmten Objekten, dann kann darauf Zugriff erlangt werden. Die folgenden Abschnitte beschreiben gängige Methoden, um durch die Modifikation von ACLs, länger in der Domäne zu verweilen und privilegierte Rechte zu bekommen [36].

Missbrauch von AdminSDHolder ACL

AdminSDHolder ist ein Objekt im AD, das als Sicherheitsvorlage für die Erstellung von AD Objekten, welche Teil von privilegierten Gruppen sind, dienen soll. Werden die ACLs dieser Gruppen überschrieben, stellt dies ein großes Risiko dar und kann das gesamte AD Umfeld gefährden. Aus diesem Grund werden die ACLs dieser privilegierten Gruppen alle 60 Minuten von der AdminSDHolder ACL überschrieben. Die Standard ACL für das AdminSDHolder Objekt ist wie folgt [36]:

- Authentifizierte Benutzer: Lesen
- SYSTEM: Vollzugriff
- Administratoren: Ändern
- Domänen Administratoren: Ändern
- Enterprise Administratoren: Ändern

Der Angriff wird ausgeführt, indem zur ACL des AdminSDHolders eine generische Berechtigung für einen bestimmte/n Benutzer/in hinzugefügt wird. Danach muss 60 Minuten gewartet werden, bis die Änderungen an die Gruppen propagiert werden. Sobald die geänderte ACL angewendet wurde, ist der/die definierte Benutzer/in Teil der Domänen Administratoren Gruppe und kann deren Mitglieder bearbeiten [65].
Der Angriff kann erkannt werden, indem die AdminSDHolder ACL sowie die Gruppenmitgliedschaften der geschützten Gruppen überwacht werden.

ACL Missbrauch über Fernzugriff

Um Zugriff zur AD-Umgebung zu erlangen stehen auch einige Möglichkeiten des Fernzugriffs zur Verfügung. Jede dieser Möglichkeiten wird durch ACLs geregelt, die von Angreifer/innen missbraucht werden können. Die häufigsten Methoden für den Fernzugriff sind Windows Management Instrumentation (WMI) und Fernzugriff über die PowerShell. Diese Methoden bieten dem/der Angreifer/in die Möglichkeit Befehle abzusetzen und somit Geräte im AD-Umfeld zu kontrollieren. Im folgenden werden die zwei Möglichkeiten des Missbrauchs der ACL für den Fernzugriff näher erläutert [36].

Bei WMI handelt es sich um eine von Microsoft Implementierung des webbasierten Enterprise Managements (WBEM). In WMI sind jegliche Daten in Klassen unterteilt, welche wiederum in Namensräume aufgeteilt sind. Zum Beispiel gibt es Klassen für Logische Treiber (Win32_LogicalDisk) und für Prozesse (Win32_Process). Um Daten über WMI zu finden, kann ein Administrator durch den Hierarchiebaum navigieren.

Da sich verschiedene Klassen in verschiedenen Namensräumen befinden ist es möglich, dass verschiedenen Zugriffsberechtigungen für verschiedenen Klassen vergeben werden. Um sich nun über WMI aus der Ferne zu verbinden, müssen die richtigen DCOM-Einstellungen (Distributed Component Object Model) und WMI-Namensraum-Sicherheitseinstellungen für die Verbindung aktiviert sein [66]. Hat ein/e Angreifer/in allerdings die Rechte eines Domänen Administrators, können die ACLs der DCOM Endpunkte und WMI Namensräume dementsprechend verändert werden, um eine Verbindung über WMI zuzulassen.
Die Änderungen an den WMI ACLs werden nicht überwacht und können somit nicht in den Logdateien nachvollzogen werden, allerdings werden bei einem Zugriff über WMI die Windows-Events mit der ID 4624 und 4634 geloggt [66].

PowerShell Fernzugriff ist ein Windows-Feature, welche interaktive Sitzungen aus der Ferne erlaubt und womit Befehle bzw. Skripts ausgeführt werden können. Standardmäßig benötigt es Administratorrechte um den Fernzugriff über PS zu ermöglichen [67]. Allerdings ist es auch hier möglich, dass ein/e Angreifer/in mit Domänen-Administratorrechten die ACL am Client modifizieren kann, um einem/r bestimmten Benutzer/in Lese-, Schreib- und Ausführrechte zu geben. Sobald der/die Benutzer/in diese Rechte hat, können Befehle ausgeführt werden ohne weitere privilegierte Rechte zu benötigen. So wie auch beim Fernzugriff über WMI, werden die Änderungen der ACL nicht geloggt, lediglich die Windows-Events 4624 und 4634 werden beim Einloggen generiert [36].

3 Stand der Forschung

S. *et al.* haben mit ihrem Paper "Detection and Prevention of Attacks on Active Directory using SIEM" [40] die grundlegende Idee für diese Arbeit geschaffen. In ihrer Arbeit wurde kurz zusammengefasst, was es benötigt, um bestimmte Angriffe im Active Directory über das SIEM zu erkennen. Das dabei verwendete SIEM System war Splunk und der Versuchsaufbau unterscheidet sich etwas von jenem in dieser Arbeit. Der Fokus lag darin die Angriffe Kerberoasting, Brute Force Angriffe, das Dumping von Anmeldeinformationen sowie den Diebstahl deren über die Security Event Logs zu erkennen und daraus Erkennungsregeln im SIEM zu erstellen. Als Schlussfolgerung wurde angemerkt, dass zusätzlich zu den Security Event Logs auch PowerShell Logs sowie Sysmon Events die Visibilität verbessert und eine bessere Erkennung erlaubt hätten.

In dem Paper "Detecting Abuse of Domain Administrator Privilege Using Windows Event Log" [45] beschreiben Fujimoto *et al.* wie der Missbrauch eines Domänen Administrator mithilfe des Windows Event Logs erkannt werden kann. Da das Auseinanderhalten von legitimen und bösartigen Zugriffen durch den Domänen Administrator oft schwierig ist, dies aber ein essentieller Teil der AD Sicherheit ist, werden in dieser Arbeit mehrere bereits existierende Methoden zur Erkennung evaluiert und eine neue Erkennungsmethode erarbeitet. Die vorgeschlagene Erkennungsmethode basiert auf den Event-IDs 4672 (Besondere Berechtigungen, die der neuen Anmeldung zugewiesen sind [68].), 4688 (Ein neuer Prozess wurde erstellt [69].), 4768 (Es wurde ein Kerberos-Authentifizierungsticket (TGT) angefordert [70].), 4769 (Ein Kerberos-Serviceticket wurde angefordert [71].) und 5140 (Auf ein Netzwerkfreigabeobjekt wurde zugegriffen [72].). Zur Evaluierung der vorgeschlagenen Erkennungsmethode wurden die Kennzahlen Recall, Genauigkeit und Präzision verwendet und mit den Kennzahlen jener Methoden verglichen, bei denen jeweils nur eine der beschriebenen Event IDs eingesetzt wurde. Dabei schneidet die vorgeschlagene Methode vor allem bei der Hit-Rate (Recall) mit 93.98 Prozent um ein Vielfaches besser ab.

Sornalakshmi erforschen in ihrem Paper "Detection of DoS attack and zero day threat with SIEM" [73],

wie DoS-Angriffe und Zero-Day-Angriffe mittels SIEM erkannt werden. Um DoS-Angriffe zu erkennen, greifen sie auf Weblogs zurück, während für die Erkennung von Zero-Days auf mehrere Aspekte geachtet wird. Dazu zählen Änderungen von Werten in der Registry, unbekannte Verbindungen zu externen Netzwerken, automatische Erzeugung von Hintergrund-Applikationen sowie die Privilegien-Eskalation von Benutzern. In der Schlussfolgerung wird geschrieben, dass durch die Implementierung der vorgeschlagenen Logiken im SIEM DoS-Angriffe und Zero-Day-Angriffe frühestmöglich erkannt werden können.

Myllylä *et al.* wollen mit ihrem Paper "Reducing the Time to Detect Cyber Attacks - Combining Attack Simulation With Detection Logic" [55] eine Methode vorstellen, um bei der Erkennung von Data Breaches wichtige Zeit zu sparen. Dazu setzen sie auf das Framework zur Erkennungslogik, das dabei helfen soll gezielt und strukturiert Erkennungslogiken für bestimmte Angriffe oder Schwachstellen zu erstellen um diese im SIEM zu implementieren. Vorgestellt wird das Framework anhand der Schwachstelle "Zerologon" mit der CVE-2020-1472. Dabei wird zuerst in einer Demo-Umgebung der Angriff simuliert und jeder Schritt dokumentiert, um dann im Anschluss dazu die Logik zur frühzeitigen Erkennung dieses Angriffs erstellen zu können. Die Autoren haben festgestellt, dass sich das Framework als effizient erweist, allerdings zu Erstellung der Logiken umfangreiches Hintergrundwissen zu den den jeweiligen Angriffen vorhanden sein muss.

In dem Paper "Security Information and Event Management (SIEM)" [4] evaluieren González-Granadillo *et al.* verschiedene weit verbreitete SIEM-Lösungen anhand unterschiedlicher Faktoren und Features und stellen fest, welche Verbesserungspotenziale hier noch gegeben sind. Es wird aufgezeigt, dass sowohl die Visualisierungs- und Reaktionsfunktionen als auch die Möglichkeiten zur Datenspeicherung bei vielen der Lösungen noch Potential zur Verbesserung haben. Außerdem sollte daran gearbeitet werden, dass Sensoren redundant eingesetzt werden können und die Möglichkeiten zur Datenspeicherung in der Cloud evaluiert werden.

Sekharan *et al.* vergleichen in ihrem Paper "Profiling SIEM tools and correlation engines for security analytics" [74] die vier bekanntesten SIEM-Lösungen hinsichtlich ihrer Features und Fähigkeiten. Unter Fähigkeiten fallen die Hauptmerkmale von SIEM-Systemen in diesem Fall sind das Echtzeit-Monitoring, Threat Intelligence, Verhaltensanalyse, Daten- und Benutzer/innenüberwachung, Applikations-Monitoring, Log Management und Reporting sowie die allgemeinen Analysefunktionen. Die Autor/innen habe festgestellt, dass die vier SIEM-Systeme IBM QRadar, HP ArcSight, Splunk und LogRythm ähnlich gute Fea-

tures und Fähigkeiten haben und ein Unternehmen entsprechend dessen Anforderungen individuell entscheiden muss, welches der Systeme die geeignetste Lösung ist.

Das Paper "AD2: Anomaly detection on active directory log data for insider threat monitoring" [75] stellt ein Framework zur Verhaltensanalyse von Benutzer/innen vor, um Insider Threat zu erkennen. Das Framework basiert dabei auf der Analyse von Active Directory Domain Service Logs eines jeden internen Accounts. Mit diesen Logs wird über das Analyse Framework, welches das Markov Model als Vorbild nimmt, eine Baseline für jeden Account erstellt. Danach wurden mehrere Tests durchgeführt, die eine Genauigkeit von 66 Prozent, für das vorgeschlagene Framework, geliefert haben. Aus diesem Grund wurde der Schluss gezogen, dass die AD Logs für sich nicht ausreichen und mit anderen Logs kombiniert werden müssen.

Mokalled *et al.* befassen sich in ihrem Paper "The Applicability of a SIEM Solution" [8] damit, wie ein Unternehmen das für deren Anforderungen am besten geeignetste SIEM auswählen kann. Dabei werden die Anforderungen in die Kategorien Plattform, Betrieb, Integrationen, erweiterte Features sowie Lizenzierung und Support eingeteilt. Jede dieser Kategorien sind zudem zwischen zwei und sechs Unterkategorien zugeordnet, die individuell gewichtet werden können. Die Bewertung erfolgt somit anhand der Werte, die in den Unterkategorien vergeben werden unter Beachtung der Gewichtung.

4 Herangehensweise

In diesem Kapitel wird beschrieben, was die Ziele bzw. die Forschungsfragen dieser Arbeit sind und wie vorgegangen wird, diese zu erreichen.

4.1 Vorgehensweise

In dieser Arbeit wird zum einen eine Literaturrecherche durchgeführt, um die verschiedenen Phasen des Prozesses zur Erarbeitung einer Alert Regel Basis im SIEM und deren Bestandteile herauszuarbeiten und anschließend zu definieren. Danach wird der definierte Prozess anhand eines Beispiels getestet. Konkret wird der Prozess angewandt, um ein Rule Set für die Active Directory Umgebung im LogPoint SIEM-Tool zu definieren. Dabei wird der Fokus auf Accountsicherheit im AD gelegt sowie die Vollständigkeit und Funktionalität der vordefinierten Regeln im LogPoint SIEM. Die genaue Vorgehensweise sowie der Versuchsaufbau der Testumgebung wird in Kapitel 6.2 näher beschrieben.

4.1.1 Forschungsfrage

Die Arbeit verfolgt das Ziel einen möglichst generischen Prozess zu definieren, wie vorgegangen werden kann, um eine Umgebung mit Hilfe eines SIEM-Tools bestmöglich zu schützen. Der Prozess soll bei der Auswahl und Anbindung geeigneter Quellen starten und als Ergebnis ein Alert Rule Set liefern, welches die Taktiken und Techniken des MITRE ATT&CK Frameworks bestmöglich abdeckt und dessen Instandhaltung gleichzeitig mit geringem Aufwand schaffbar ist. Das heißt zu einer Menge an Loginformationen, soll ein Rahmen geschaffen werden, diese gezielt zu verarbeiten, um einen proaktiven Ansatz zum Schutz der Unternehmensumgebung zu kreieren.
Das zweite Ziel der Arbeit ist es zu bewerten, wie einfach oder schwierig es ist, diesen Prozess mit dem LogPoint SIEM-Tool umzusetzen. Dabei soll von der Anbindung der benötigten Logquellen, über den Einsatz der eingebauten Alert Rules bis hin zu dessen Adaptierungsmöglichkeiten bzw. deren Notwendigkeit zur Adaptierung bewertet werden.

5 Prozess zur Schaffung einer Alert Regel Basis

Folgender Abschnitt beschreibt den erarbeiteten Prozess zur Schaffung und Instandhaltung eines Basis-Sets an Alert Regeln, um Unregelmäßigkeiten frühestmöglich zu erkennen und Gegenmaßnahmen setzen zu können. In Abbildung 5.1 wird der folgende Prozess grafisch dargestellt.

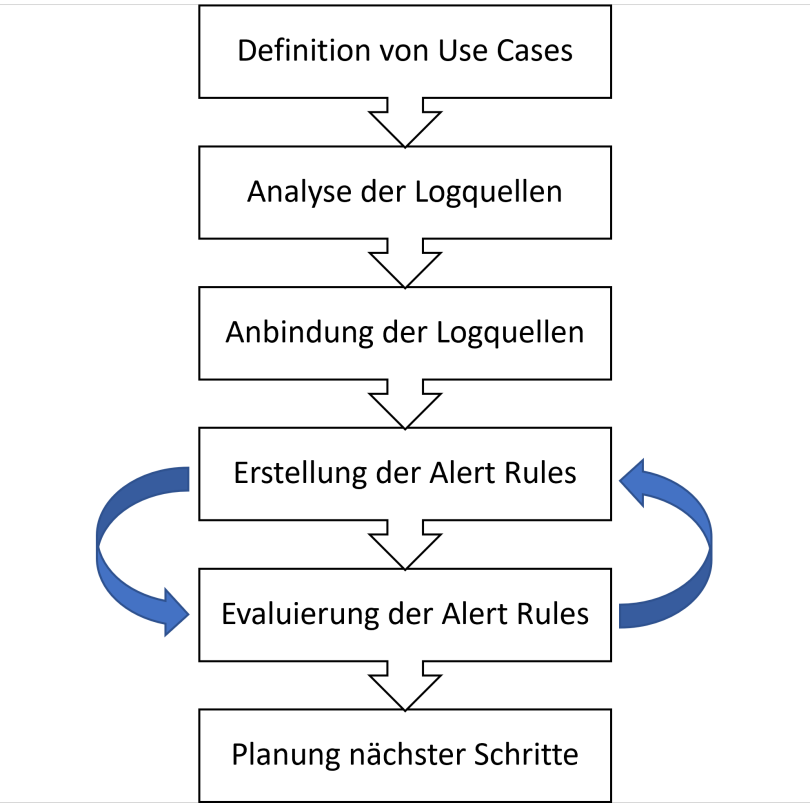

Abbildung 5.1: Prozessablauf

5.1 Definition von Use Cases

Da auch die beste SIEM-Lösung nicht out-of-the-box funktioniert, müssen im ersten Schritt Use Cases definiert werden. In einem Use Case wird ein Szenario definiert, welches auffälliges oder ungewöhnliches Verhalten darstellt und weiter untersucht werden sollte, wenn es auftritt. Anhand dieser Use Cases können alle weiteren Entscheidungen getroffen werden, wie die Arten von Logs die gesammelt werden sollen, die Alert Rules die erstellt werden sollen sowie die Dashboard die gebaut werden sollen. Je nach Anforderungen eines Unternehmens oder dem zu erreichenden Ziel können sich die Use Cases von Unternehmen zu Unternehmen stark unterscheiden [1].
Bei der Erstellung von Use Cases können die Frameworks CIS Critical Security Controls[1] (CIS Controls) und MITRE ATT&CK hilfreich sein. Das Framework von MITRE wurde bereits im Kapitel 2.2 beschrieben. Bei den CIS Controls handelt es sich um ein Framework, welches aktuelle Standards und Empfehlungen hinsichtlich Cyber Security in Unternehmen beinhaltet. Anhand der darin beschriebenen Kontrollen lassen sich ebenfalls Use Case ableiten [76].

5.2 Analyse der Logquellen

Sind Use Cases definiert, so muss anhand dieser definiert werden, welche Logs benötigt werden, um die beschriebenen Szenarien zu erkennen. In diesem Schritt ist es besonders wichtig sich auf jene Logs zu beschränken, die tatsächlich benötigt werden, um nicht zu viel Speicherplatz mit unwesentlichen Logs zu verbrauchen. Um das zu erreichen, muss betrachtet werden, wie die Logs aufgebaut sind und anhand der Use Cases Filter-Kriterien definiert werden. Ein Filter-Kriterium im Active Directory Umfeld wäre beispielsweise die Event-ID.
Dazu kommt, dass die Logs verschiedener Logquellen unterschiedlich aufgebaut sind, das heißt, die Logs müssen normalisiert werden. Falls es noch kein einheitliches Format für die Speicherung von Logs gibt, dann muss das ebenfalls in diesem Schritt festgelegt werden. Dabei ist vor allem auf die einheitliche Namensgebung zu achten. Zum Beispiel soll die Quell-IP-Adresse, in allen Logs unabhängig deren Herkunft als `source_ip` gespeichert werden. Nur so kann eine einheitliche Verarbeitung der Logs und Verwendung für Alert Rules, Dashboards und Suchfunktionen gewährleistet werden.
Der letzte Punkt der an dieser Stelle bedacht werden muss, ist ob die Informationen, die in den Logs enthalten sind für die Use Cases ausreichend sind oder um Informationen bereichert werden sollten. Die Anreicherung der Logs mit Informationen bietet unterschiedliche Möglichkeiten. Zum einen können Tags oder

[1] https://www.cisecurity.org/controls

Labels hinzugefügt werden, um eine schnellere Suche zu ermöglichen und den Analyst/innen auf einen Blick wichtige Informationen zukommen zu lassen, zum anderen können auch Informationen wie DNS Lookups oder Objektinformationen aus dem AD hinzugefügt werden. Eine weitere Möglichkeit ist die Anbindung externer Threat Intelligence Feeds, um beispielsweise einem Hash-Wert den Virus-Total-Score hinzuzufügen oder ein Tag hinzuzufügen, wenn eine Domäne offensichtlich missbraucht wird. Unter Umständen kann die Anreicherung von Logs ressourcenintensiv sein, aus diesem Grund gilt auch hier, dass nicht wahllos alle Informationen hinzugefügt werden sollen, sondern dies genau geplant und den Anforderungen angepasst werden sollte.

5.3 Anbindung der Logquellen

Wenn nun feststeht, welche Logquellen und welche Logs eingebunden werden sollen, muss dies an den Logquellen bzw. im SIEM entsprechend konfiguriert werden. Dabei sind die im Schritt zuvor festgelegten Punkte umzusetzen. Das bedeutet, es muss konfiguriert werden, welche Logs aufgenommen werden (Filterung), wie diese vereinheitlicht werden (Normalisierung) und mit welchen Informationen diese angereichert (Enrichment) werden sollen.
Die Umsetzung des Filterings kann entweder an der Logquelle selbst zum Beispiel in Form einer Log Policy umgesetzt werden oder in der Konfiguration der SIEM-Lösung, zum Zeitpunkt wenn die Logs entgegengenommen werden.
Beim Schritt der Normalisierung hängt der Arbeitsaufwand stark von der gewählten SIEM-Lösung ab. In manchen Lösungen, wie beispielsweise Logstash als Teil der Elastic Stack Lösung, muss eine Konfigurationsdatei geschrieben werden, in der angegeben wird, welcher Teil des Logs in welches Feld geschrieben wird. In anderen, meist kommerziellen, Lösungen ist die Normalisierung oftmals als Plugin bereits eingebaut, kann aber modifiziert oder angepasst werden. Speziell, wenn Logs nicht dem Standard entsprechen, von eigens programmierten Applikationen stammen oder im Syslog-Nachrichten-Feld noch weitere Daten geparsed werden müssen, ist eine Anpassung häufig notwendig.
Im letzten Schritt kann das Enrichment konfiguriert werden. Ähnlich wie bei der Normalisierung, kann das in der SIEM-Lösung integriert sein oder ist über die Konfigurationsdatei realisierbar.

5.4 Erstellung der Alert Regeln

Wurde die Konfiguration abgeschlossen und fließen die Logs erfolgreich und im gewünschten Format ins SIEM, kann mit der Erstellung der Alert Rules begonnen werden. Sind built-in Alert Rules verfügbar, so

sollten diese evaluiert werden, bevor eigene Regeln geschrieben werden.
Bei der Evaluierung bestehender Alert Regeln sollte darauf geachtet werden, wie viel Anpassung diese benötigen und ob sie die definierten Use Cases vollständig abdecken können. Es wird selten der Fall sein, dass Regeln einfach aktiviert werden können. In den meisten Fällen müssen Felder wie IP-Adressen oder -Netze, Namenskonventionen und Ausnahmen definiert werden.
Sind bereits Regeln aktiviert, die in den Wirkungsbereich der Use Cases fallen, so sollten auch diese betrachtet werden, um mögliche Redundanzen oder Lücken in der Abdeckung zu vermeiden.
Bei der Definition neuer Regeln ist darauf zu achten, dass diese weder zu generisch noch zu spezifisch gehalten werden. Zum einen sollen eine große Menge an False Positives (FP) vermieden werden, zum anderen soll es auch nicht zu False Negatives (FN) kommen, wo die Regel nicht anschlägt.
In allen drei Fällen ist es wichtig, dass die Alert Regeln getestet werden, bevor sie aktiv geschalten werden.

5.4.1 UEBA

UEBA steht für User and Entity Behavior Analytics und zielt darauf ab Anomalien zu erkennen. Dafür lernt es zu Beginn über einen gewissen Zeitraum, wie die Baseline aussieht, um anschließend Abweichungen von dieser Baseline festzustellen und bei Bedarf zu alarmieren.
Manche SIEM-Lösungen haben eine UEBA-Lösung integriert bzw. können sie um ein UEBA Modul erweitert werden. Ob der Einsatz von UEBA für bestimmte Quellen und Use Cases unterstützt wird, hängt vom jeweiligen Tool ab und sollte ebenfalls evaluiert werden.

5.4.2 Abdeckungsgrad feststellen

Auch, wenn die Alert Regeln je Use Case erstellt, aktiviert und evaluiert werden sollten, so ist es doch Ziel am Schluss einen möglichst guten Abdeckungsgrad für ein bestimmtes Szenario bzw. Ziel, wie beispielsweise Active Directory Sicherheit zu erreichen. Um diesen Abdeckungsgrad visuell darstellen zu können, kann auf den ATT&CK Navigator von MITRE zurückgegriffen werden [77]. Hier ist es zum einen möglich zu markieren, in welchen Bereichen des MITRE ATT&CK Frameworks bereits Alert Rules im Einsatz sind, um einen Überblick zu schaffen, zum anderen kann es in Kombination mit DeTT&CT Editor verwendet werden, um das Potenzial für weitere Alert Regeln zu erkennen, die mit den Logs, die bereits vorhanden sind, umgesetzt werden können. Dazu wird im DeTT&CT Editor angegeben, welche Logquellen in welchem Ausmaß in das SIEM einfließen und die daraus resultierende YAML-Datei, kann nach einer Umwandlung ins JSON-Format im ATT&CK Navigator visualisiert werden [78]. Daraus kann nicht nur die Notwendigkeit weiterer Alert Regeln abgeleitet werden, sondern auch ein guter Überblick darüber geschaf-

fen werden für welche Taktiken und Techniken die Systeme verwundbar sind.

5.5 Evaluierung der Alert Regeln

Damit sichergestellt werden kann, dass die Alert Rules zum einen tatsächlich funktionieren und diese zum anderen auch effizient sind, müssen diese nach der Erstellung und Aktivierung evaluiert werden. Die Evaluierung sollte erstmals sofort nach initialer Aktivierung durchgeführt werden und danach regelmäßig erfolgen. Dabei können die Zeitabstände anfangs etwas kürzer sein, beispielsweise 30 Tage und nach erfolgreicher Anpassung verlängert werden auf beispielsweise ein Jahr. Es kann aber auch notwendig sein, eine Evaluierung bei Bedarf durchzuführen, wenn ein Alert viel zu viele False Positives liefert. Außerdem sollte eine Evaluierung auch stattfinden, wenn ein Event oder Incident stattgefunden hat und kein Alert ausgelöst wurde, obwohl dieses Event oder dieser Incident durch eine Alert Rule abgedeckt hätte sein sollen. Ziel der Evaluierung ist es, festzustellen, wie gut die Alert Regeln auf die Umgebung zugeschnitten sind und welche Anpassungen getätigt werden müssen, um weniger False Positives zu erzeugen und gleichzeitig nicht zu spezifisch sind und Events übersehen werden.
Bei der Evaluierung sollten zwei Komponenten geprüft werden, die Funktionalität und die Effektivität.

5.5.1 Funktionalität

Bei der Funktionalität geht es darum zu testen, ob die Alert Regel tatsächlich anschlägt oder ob Felder oder Werte geändert bzw. angepasst werden müssen, damit die Regel wie erwartet funktioniert und alarmiert.
Diese Tests können je nach Use Case und Alert Regel unterschiedlich komplex und aufwendig ausfallen und sollten auf jeden Fall in einer Testumgebung ausgeführt werden. Die Ausführung in Produktivsystemen wird in vielen Fällen zudem gar nicht möglich sein, da hoch privilegierte Rechte benötigt werden, die ein/e Analyst/in im Normalfall nicht haben sollte.

5.5.2 Effektivität

Bei der Effektivität soll es darum gehen zu bewerten, ob die Regel sinnvoll und zielführend ist. Dabei ist jede Regel einzeln und in Anbetracht des jeweiligen Use Cases zu analysieren.
Es geht dabei darum folgende Punkte zu bewerten:

- Zielerfüllung: Es soll festgestellt werden, ob die Regel zur Erfüllung der Ziele des Use Cases bzw. mehrerer Use Cases beiträgt.

- Redundanz: Eine große Menge an Alerts kann dazu führen, dass Alerts übersehen werden bzw. nicht mehr ernst genommen werden (Alert Fatigue). Deshalb ist es wichtig, dass Alert Regeln gezielte Fälle abdecken und nicht übermäßig viele Meldungen produzieren. Decken sich zwei Fälle gänzlich ab, sind diese zusammenzulegen. Decken sich Fälle zum Teil ab, dann ist zu evaluieren, ob es sinnvoll ist die Alert Regeln zusammenzulegen.
- False Positives: Es muss überprüft werden, wie viele Alerts eine Regel in einem bestimmten Zeitraum, beispielsweise an einem Tag, in einer Woche oder in einem Monat, produziert. Dabei gilt es auch festzustellen, warum dies so ist, um Alert Fatigue zu vermeiden. Es kann sich dabei um normale Vorgänge im Unternehmen handeln, um eine Ungenauigkeit der Regel oder um eine tatsächliche Kompromittierung. Wurde der Grund festgestellt, gilt es Maßnahmen dagegen zu definieren. In den ersten beiden Fällen kann die Regel angepasst oder sogar deaktiviert werden, wenn der überprüfte Fall nicht zu den Bedürfnissen des Unternehmens passt. Wurde festgestellt und verifiziert, dass eine Kompromittierung vorliegt, hat die Regel funktioniert und es sollte der Incident Management Prozess eingeleitet werden.

Wichtig ist, dass sobald eine Regel geändert wird, die ganze Evaluierungsphase neu durchlaufen werden muss. Speziell das Testen der Funktionalität sollte sofort durchgeführt werden, um keine Blindspots zu erzeugen. Aber auch die Effektivität muss erneut beurteilt werden und sollte hier auch zu einem besseren Ergebnis führen.

5.5.3 Kennzahlen zur Bewertung

Zur Bewertung der Funktionalität, aber auch Effektivität einer Alert Regel gibt es einige Kennzahlen, die herangezogen werden können. Dabei ist allerdings zu beachten, dass die Kennzahl einer Regel oft nur schwer mit der Kennzahl einer gänzlich anderen Regel vergleichbar sein wird. Wozu die Kennzahlen allerdings hilfreich sein können ist eine Regel vor und nach einer Veränderung zu vergleichen. Die klassischen verwendeten Kennzahlen dabei sind folgende [79]:

- True Positives (TP): Anzahl der anormalen Aktivitäten, welche richtig als anormal erkannt wurden und Alerts ausgelöst haben.
- False Positives (FP): Anzahl der normalen Aktivitäten, welche fälschlicherweise als anormal klassifiziert wurden und Alerts ausgelöst haben.
- True Negatives (TN): Anzahl der normalen Aktivitäten, welche richtig als normale Aktivitäten klassifiziert wurden und somit kein Alerts ausgelöst haben.
- False Negatives (FN): Anzahl der anormalen Aktivitäten, welche fälschlicherweise als normale Ak-

tivitäten klassifiziert wurden und keine Alerts ausgelöst haben.

Diese vier Werte können in einer Wahrheitsmatrix (englisch: Confusion Matrix), wie in Abbildung 5.2 dargestellt werden.[79]

		Tatsächliche Aktivität	
		Anormale Aktivität	**Normale Aktivität**
Einstufung der Aktivität aufgrund des Alerts	**Anormale Aktivität**	True Positive	False Positive
	Normale Aktivität	False Negative	True Negative

Abbildung 5.2: Wahrheitsmatrix [79]

Hierzu ist anzumerken, dass nicht alle vier Kennzahlen gleichermaßen geeignet sind, um die Effektivität einer Alert Regel zu bestimmen. Während sich True Positives und False Positives gut eignen, um eine Einschätzung der Regel zu bekommen, ist es schwer True Negatives und False Negatives zu bestimmen. In einem SIEM kann jedes einzelne Log Event, welches keinen Alert erzeugt hat, als True Negative angesehen werden, solange es nicht durch ein anderes Tool, einen Incident oder einer/m Analysten/in als False Negative klassifiziert wurde. Einzig in der Testumgebung bei Funktionalitätstests können False Negatives meist gut identifiziert werden, in diesem Fall stellen FNs Logs dar, die einen Alert hätten auslösen sollen, die Regel allerdings nicht wie erwartet funktioniert hat.

Aus diesen vier Kennzahlen, können weitere Kennzahlen berechnet werden. Allerdings eignet sich nicht jede der möglichen Kennzahlen für die Evaluierung der Alert Regeln. Da True Negatives im SIEM nicht bestimmbar sind, fallen alle jene Kennzahlen weg. Auch jene mit False Negatives können nur eingeschränkt verwendet werden. Eine Kennzahl welche es ermöglicht die Alert Regel in der Produktivumgebung zu bewerten, ist die Präzision. Die Präzision drückt aus, wie viele Ereignisse, aller Alerts tatsächlich verdächtig oder boshaft waren [79]. Je näher der Wert bei 1 ist, desto genauer ist die Regel. Ist der Wert näher bei 0, wirft sie ziemlich viele False Positives und sollte überarbeitet werden.

$$Präzision = \frac{TP}{TP + FP}$$

Ist das Alert Rule Set für einen Use Case bereits mindestens durch die erste Evaluierungsphase gelaufen, kann auch der ganze Use Case auf Funktionalität getestet werden, indem ein bestimmtes Szenario getestet wird. Dies macht Sinn, da eine Alert Regel nur auf eine Aktivität achtet, ein Angriff aber eine Verkettung vieler Aktivitäten ist. Hierzu sollte vorher festgelegt werden, von welchen Regeln erwartet wird, dass diese anschlagen, sowie durch welche Events das geschehen sollte.

In so einem Test-Szenario ist es somit auch möglich, False Negatives zu identifizieren. Das ermöglicht zusätzlich den Einsatz der Detection Rate. Diese wird oft auch True Positive Rate, Sensitivität oder Recall genannt. Sie gibt den Anteil der richtig erkannten Ereignisse an allen geplanten verdächtigen Ereignissen an [79].

$$Detection\ Rate = \frac{TP}{TP + FN}$$

Je näher der Wert an 1 ist, desto besser haben die Alert Regeln in dem Szenario funktioniert. Wobei hier auch darauf geachtet werden sollte, wie die Präzision ausfällt, da bei zu vielen False Positives, die True Positives untergehen könnten und somit wieder die Qualität der Regeln gemindert wird.

5.6 Planung nächster Schritte

Wurde ein vollständiges Alert Rule Set aufgebaut und die erste Evaluierungsphase der Regeln abgeschlossen, müssen weitere Punkte in Betracht gezogen werden, die Schnittpunkte zu darauf folgenden Prozessen beinhalten.

5.6.1 Benachrichtigungen

Schlagen Alert Rules an, dann müssen die Analyst/innen schnellstmöglich benachrichtigt werden. Hier ist zu überlegen, wie diese Benachrichtigung erfolgen soll und welche Daten bereits in der Nachricht enthalten sein sollen. Die Möglichkeiten gehen von der Benachrichtigung per E-Mail, über SMS oder das Auslösen von Anrufen. Der Vorteil von E-Mail-Nachrichten ist jener, dass erste Daten schon in der Mail dargestellt werden können und so unter Umständen eine erste Priorisierung durchgeführt werden kann, ohne das SIEM-Tool zu öffnen. Der Nachteil ist, dass E-Mail-Nachrichten schnell in der Menge an Mails untergehen können und ihnen nicht mit der notwendigen Dringlichkeit nachgegangen wird. Anrufe und vor allem SMS sind wesentlich seltener und bewirken so, dass Alerts schnellstmöglich angesehen und bearbeitet werden. Eine Erstdarstellung der Alert Daten in der Nachricht ist dagegen nicht möglich. Neben diesen Möglichkeit gibt es noch einige mehr, welche für das eigene SOC-Team am besten passt, hängt stark von den verfügbaren Ressourcen sowie dessen Organisation ab und muss individuell entschieden werden.

5.6.2 Bearbeitung der Alerts

Die Bearbeitung der Alerts muss nachvollziehbar dokumentiert werden. Daher muss eine Vorgehensweise definiert werden, welche dies einheitlich ermöglicht. Mit welchen Mitteln oder Tools dies umgesetzt wird,

hängt von den Möglichkeiten des SIEM-Tools und der allgemeinen Tool-Landschaft in einem Unternehmen ab. Manche SIEM-Tools haben built-in Incident Management Funktionen, allerdings sind diese je nach Tool sehr unterschiedlich ausgeprägt. Auch die Schnittstellen zu Ticketing Tools können sich von SIEM zu SIEM unterscheiden und bieten daher unterschiedliche Möglichkeiten.

5.6.3 Automatisierung

Es sollte sich unbedingt angesehen werden, inwiefern eine automatisierte Behandlung mancher Alerts möglich oder sinnvoll ist. Die Automatisierung des Alert Handlings kann nicht nur Ressourcen sparen sondern ermöglicht auch eine sofortige Behandlung des Alerts zum Zeitpunkt des Auftretens. Meistens wird die Automatisierung durch ein SOAR-Tool realisiert. Hier können Workflows mit bestimmten Auslösern definiert werden, die einen Alert komplett abarbeiten oder für den/die Analyst/in aufbereiten können. Manche SIEM-Tools haben diese Funktionalität integriert bzw. kann diese Funktion hinzugekauft werden.

6 Erstellung eines Alert Rule Sets für Active Directory anhand des definierten Prozesses

In diesem Kapitel wird gezeigt, wie der in Kapitel 5 beschriebene Prozess angewendet werden kann. Dazu werden Use Cases aus dem Active Directory definiert und mit dem SIEM des Herstellers LogPoint umgesetzt.

6.1 Eingesetzte SIEM-Lösung - LogPoint

LogPoint ist ein SIEM-Hersteller mit dem Hauptsitz in Kopenhagen, Dänemark. Der Hersteller hat Kunden weltweit, ist allerdings sehr fokussiert auf den europäischen Markt. Die SIEM Lösung bietet ein UEBA, ein SOAR sowie den LogPoint Director an, wobei UEBA als extra Produkt gekauft werden muss. Als Erweiterungen werden auch LogPoint für SAP und Applied Analytics angeboten [2], [80].
Die Lizenzierung basiert auf einem Abo-Modell, wobei die Anzahl an überwachten Entitäten bezahlt wird. Die UEBA-Komponente wird extra lizenziert basierend auf der Anzahl der Mitarbeiter und den weiteren angebundenen Entitäten [2]. Auch die SOAR-Lösung ist für mehrere Benutzer/innen zusätzlich zu lizenzieren, da im Standard SIEM lediglich die Lizenz für eine/n Benutzer/in enthalten ist. Somit ist es nur einem/r SIEM-Nutzer/in möglich, Workflows zu implementieren.
Die SIEM-Lösung ist verfügbar als On-Premise-Lösung oder in der Cloud gehostet. Das UEBA ist allerdings nur als Software-as-a-Service-Lösung (SaaS) verfügbar [2], [80].
In Bezug auf den Magic Quadrant von Gartner ist das SIEM der Firma LogPoint ein Niche Player. Niche Players werden definiert als Lösungen, welche gut auf sehr bestimmte Anwendungsfälle oder Funktionsanforderungen eines bestimmten Marktes passen. Dabei fokussieren sie sich auf ein bestimmtes Kundensegment oder aber stellen limitierte SIEM-Funktionen zur Verfügung. In dem Bericht weist Gartner auf folgende Stärken hin [2]:

- Marketing und Produkte passend auf bestimmte Anwendungsfälle: LogPoint vermarktet produktspezifische Funktionen wie SAP-Sicherheitsüberwachung und Evaluation Assurance Level (EAL) 3+

Zertifizierung an entsprechende Organisationen (z.B. solche, die SAP ERP verwenden) und Branchen wie Behörden und Produktionsunternehmen.

- Unterstützung für Dienstleister und komplexe Organisation: LogPoint verfügt über integrierte Mandantenfähigkeit. Darüber hinaus unterstützt das LogPoint Director-Add-on die zentrale Verwaltung von Multi-Instanz-Implementierungen, was für Dienstleister und Unternehmen interessant ist, die eine SIEM-Lösung suchen, die ein Parent-Child-Implementierungsmodell unterstützt (z. B. Unternehmen mit einem Hauptsitz, der verschiedene Geschäftszweige unterstützt).
- Integrierte Funktionen für Datenschutz und Datensicherheit: Funktionen wie Datenmaskierung und -verschleierung tragen dazu bei, die Anforderungen an die Privatsphäre und den Datenschutz im Zusammenhang mit der Datenschutzgrundverordnung (DSGVO) und dem California Consumer Privacy Act (CCPA) zu erfüllen.

Neben den Stärken weist Gartner auch auf Einschränkungen der SIEM-Lösung hin [2]:

- Fußabdruck außerhalb Europas: Europa ist sowohl der Heimatmarkt von LogPoint als auch sein größter Markt. Beim Direktvertrieb in anderen Regionen liegt LogPoint hinter vielen Wettbewerbern zurück. LogPoint gibt jedoch an, dass es dieses Problem durch Investitionen in seine Vertriebsaktivitäten und deren Reifung sowie durch die Expansion in andere Regionen zur Ergänzung der Aktivitäten seiner Vertriebspartner in Angriff nimmt.
- Eingeschränkte Formfaktoren: Die SIEM-Lösung ist nur als eigene Anwendung verfügbar (physisch oder virtuell), es gibt kein Angebot für SaaS. Die UEBA-Erweiterung wird allerdings nur als SaaS angeboten. Käufer, die eine gehostete Option suchen, müssen die Lösung von LogPoint in ihrer eigenen öffentlichen oder privaten Cloud-Umgebung installieren oder Partner nutzen, die eine gehostete Option anbieten können. Potenzielle UEBA-Käufer sollten sich vergewissern, wie die Anforderungen an den Datenschutz und den Wohnsitz erfüllt werden können.

6.1.1 Aufbau und Funktionsweise von LogPoint

Das LogPoint SIEM besteht im Wesentlichen aus drei Hauptkomponenten. Diese können entweder auf verschiedene physische oder virtuelle Server aufgeteilt oder als eine Appliance geliefert werden. Für kleinere Netzwerke bietet die Lösung in einer Appliance den Vorteil der Einfachheit des Designs und der Wartung, für größere Netzwerke eignet sich die Verteilung auf mehrere Server, um diese in verschiedenen Netzwerkzonen aufteilen zu können [81].
Im folgenden werden die drei Komponenten des LogPoint SIEM beschrieben:

LogPoint Collectors

Die Kollektoren sind dafür zuständig die Logdaten aufzunehmen, zu normalisieren und anschließend anzureichern. LogPoint klassifiziert und kategorisiert die Daten dabei einheitlich und mithilfe der mitgelieferten Plugins werden die Logdaten in standardisierte Key-Value-Paare strukturiert. Im Gegensatz zu anderen SIEM-Lösungen findet diese Normalisierung bei der Aufnahme der Logdaten und nicht erst bei der Suche statt. Auf diese Weise kann der Such- und Korrelationsprozess beschleunigt werden.
Durch die Bereitstellung der Plugins kann der Prozess der Datenaufnahme erheblich beschleunigt werden, da keine manuelle Konfiguration notwendig ist. LogPoint verspricht dabei, ein Plugin im Zuge des Supportprozesses zu erstellen, sollte keines verfügbar sein.
Die Kollektoren bieten zudem die Möglichkeit der Datenanreicherung. So können Logdaten mit Metadaten oder Daten aus Threat-Intelligence-Feeds schon bei der Aufnahme angereichert werden und in Folge dessen die Analysen schneller und genauer erfolgen [81].

LogPoint Backend

Das Backend stellt den Datenspeicher basierend auf einer NOSQL-Datenbank dar. Die Daten können dabei in mehreren Projektarchiven, auch Repos genannt, abgespeichert werden. Diese können vom Unternehmen entsprechend deren Anforderungen erstellt werden. So ist es auch möglich mithilfe der Repos Daten auf unterschiedliche Speicherebenen zu migrieren, um auch das längere Aufheben von Daten (Archivierung) zu ermöglichen. Basierend auf den Repos kann auch der Zugang zu den Daten kontrolliert werden. LogPoint Benutzer/innen kann so Zugriff auf bestimmte Daten, wie beispielsweise Firewall-Logs gegeben werden ohne dabei Zugriff auf die Logs der Endgeräte zu haben [81].

LogPoint Searchhead

Der Searchhead ist die Analyse-Komponente des LogPoint SIEM. Dort kann der/die Analyst/in benutzerdefinierte Inhalte entwickeln indem die Werte aus den rohen oder normalisierten Daten genutzt werden. Mithilfe der eingebauten Log-Analyse-Engine können diese Inhalte verarbeitet und kritische Ereignisse in den angebundenen Systemen automatisch erkannt und gemeldet werden.
LogPoint bietet dabei über 400 vorkonfigurierte Anwendungsfälle in Form von Search Template, Alert Regeln und Dashboards. Die sollen dem/der Analystin die Suche nach anormalen oder verdächtigem Verhalten erleichtern [81].

Neben den Infrastruktur-Komponenten, sind auch Policies ein essentieller Bestandteil des SIEM. Diese müssen für jede Logquelle konfiguriert werden, um die richtige Aufnahme und Verarbeitung der Logs sicherzustellen. Folgende Policies müssen bei der Anbindung einer neuen Logquelle konfiguriert werden:

- **Routing Policy:** Mit der Routing Policy kann konfiguriert werden, welche Logs in welchen Repos gespeichert werden sollen. Dabei kann die Aufteilung auf die Repos mittels Key-Value Paaren oder anhand des Key-Present Kriteriums definiert werden [82].
- **Normalization Policy:** In der Normalization Policy wird angegeben, welche Normalization Packages verwendet werden sollen, um die Logs in Key-Value Paare zu strukturieren. Dabei kann zwischen Compiled Normalizer und den regex-basierenden Normalization Packages gewählt werden. Werden beide ausgewählt, wird zuerst immer der Compiled Normalizer verwendet. Zudem werden die Normalization Packages immer in der Reihenfolge ausgeführt, in der sie in der Policy angegeben werden [83].
- **Enrichment Policy:** Mit der Enrichment Policy kann festgelegt werden, welche Logs mit welchen Daten angereichert werden sollen. Dabei müssen zum einen die Enrichment Kriterien festgelegt werden und zum anderen die Enrichment Regeln. Die Kriterien definieren, welche Logs mit Daten angereichert werden sollen. Nur, wenn die Kriterien erfüllt sind, werden die Enrichment Regeln angewendet. In der Regel wird definiert, welches Feld aus dem Log mit welchem Feld in der Enrichment Quelle übereinstimmen soll, um die entsprechenden Daten anzureichern [84].
- **Processing Policy:** In der Processing Policy werden die zuvor beschriebenen Policies in einer zusammengefasst, um sie anschließend der entsprechenden Logquelle zuzuordnen [85].

6.2 Versuchsaufbau

Um die aufgestellten Alert Rule Sets entsprechend testen zu können wird dies in zwei Umgebungen durchgeführt, in einer Testumgebung und in einer Produktivumgebung.

6.2.1 Testumgebung

Diese Umgebung ist eine reine Testumgebung mit Vollzugriff auf den Domain Controller, diese soll dazu dienen die Funktionalität zu überprüfen. Das heißt es wird geprüft, ob die Regeln tatsächlich anschlagen, um zu verhindern, dass Regeln gebaut werden, die ihr Ziel nicht erfüllen.
Die Testumgebung besteht aus dem SIEM-Server, einem Domain Controller (Server 2019) sowie einem Windows 10 Client. Der SIEM-Server umfasst alle Komponenten des LogPoint-SIEMs, es wurden keine

zusätzlichen Backend-, Collector-, oder Searchhead-Server implementiert. In Abbildung 6.1 wird die Testumgebung grafisch dargestellt.

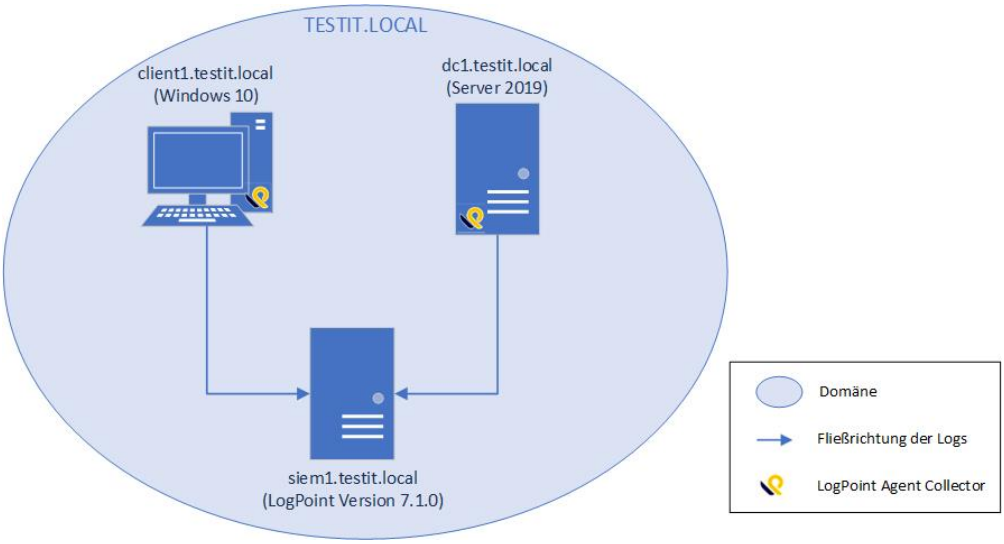

Abbildung 6.1: LogPoint SIEM Testumgebung

6.2.2 Produktivumgebung

Diese Umgebung ist eine Produktivumgebung in einem Unternehmen mit ungefähr 20.000 Active Directory Benutzer/innen exklusive Administratorkonten und Service Accounts. In dieser Umgebung wird vor allem die False Positive Rate einiger Rules überprüft.
In der Produktivumgebung ist bereits ein LogPoint SIEM implementiert. Hier handelt es sich allerdings um mehrere Server mit verteilten Rollen. Um die Alert Regeln zu testen, wurde hier ein/e separate/r Benutzer/in angelegt, der/die nur Zugriff auf die Active Directory Logs hat und noch keine Alert Regeln erstellt wurden. Es gilt zu beachten, dass die Ergebnisse für die Regeln hinsichtlich ihrer Effektivität nur für jene Umgebung gültig sind, in der die Tests durchgeführt wurden.

6.3 Umsetzung laut definiertem Prozess

Die Ausgangslage umfasst das zuvor beschriebene bereits implementierte SIEM-Tool des Herstellers LogPoint. Die Anforderung seitens des Managements besteht darin, dass das Active Directory bestmöglich vor Angriffen geschützt werden soll.

6.3.1 Definition von Use Cases

Da der Schutz des Active Directorys nicht präzise genug ist, werden im folgenden Teil Use Cases definiert, welche in erster Linie umgesetzt werden sollen. Dabei ist wichtig anzumerken, dass diese Use Cases lediglich den Anfang darstellen und nach der Etablierung dieser an weiteren Use Cases gearbeitet werden sollte.
Da die Übernahme oder der Missbrauch von Accounts ein hohes Risiko im Active Directory darstellen, beschäftigen sich die folgenden Use Cases nach MITRE ATT&CK Framework vor allem mit dem Schutz dieser.

Valid Accounts

Angreifer/innen benutzen bereits bestehende Konten in einigen Taktiken des MITRE ATT&CK Frameworks. Dazu gehören Initial Access, Persistence, Privilege Escalation und Defense Evasion. Kompromittierte Zugangsdaten können benutzt werden um Zugangskontrollen zu überwinden und auf diverse Ressourcen in einem Netzwerk zuzugreifen. Möglicherweise können diese Credentials sogar dazu dienen um langfristig Fernzugriff auf Systeme und externe Services, wie VPN, Outlook Web Access oder Remote Desktops zu erlangen. Abhängig von den kompromittierten Konten kann es den Angreifer/innen auch gelingen deren Rechte zu erhöhen, um Zugang zu bestimmten Systemen oder geschützten Bereichen des Netzwerk zu bekommen. Diese Zugangsdaten werden meist nicht zur Ausführung von Malware oder Tools verwendet, um die Erkennung der Aktivitäten zu erschweren. Neben den Standard Benutzerkonten und Administratorkonten sollten auch inaktive Konten überwacht werden. In manchen Fällen missbrauchen Angreifer/innen inaktive Accounts um nicht erkannt zu werden, da auch der/die ursprüngliche Benutzer/in nicht mehr anwesend ist, um das anormale Verhalten zu identifizieren [86].

Account Manipulation

Bei der Manipulation von Accounts geht es darum, den Zugriff zu bereits kompromittierten Accounts langfristig zu behalten. Dazu ändern Angreifer/innen beispielsweise das Passwort des Accounts oder die Gruppenzugehörigkeiten. Oftmals wird das Passwort dabei nicht nur einmal geändert sondern immer wieder, um die Sicherheitsrichtlinien zu umgehen und die kompromittierten Zugangsdaten möglichst lang nutzen zu können [87]. Voraussetzung für die Manipulation oder Erstellung von Benutzerkonten ist, dass die Angreifer/innen bereits ausreichende Berechtigungen auf einem System oder in der Domäne haben. Außerdem kann diese Technik dazu führen, dass die Rechte erhöht werden können zum Beispiel, wenn Änderungen

Zugriff zu zusätzlichen Rechten, Rollen oder Benutzerkonten gewähren [87].

Exploitation for Privilege Escalation

Neben den oberen beiden Techniken kann auch die Ausnutzung von Software-Schwachstellen den Angreifer/innen dazu dienen höhere Rechte zu erlangen. Software-Schwachstelle bezeichnet dabei einen Programmfehler in einem Programm, einem Service oder in der Betriebssystemsoftware bzw. im Kernel. Dieser Programmfehler wird dazu benutzt, um eigenen Schadcode auszuführen. Wenn Angreifer/innen erstmals Zugriff zu einem System bekommen, arbeiten sie oftmals über einen wenig privilegierten Prozess, wodurch sie keinen Zugriff zu bestimmten Ressourcen am System haben. Deshalb suchen sie nach den oben beschriebenen Schwachstellen in Software, die häufig mit höheren Rechten läuft. Auf diese Weise kann es den Angreifer/innen gelingen, von Benutzerrechten zu SYSTEM oder Root Berechtigungen aufzusteigen, abhängig davon worin sich die ausgenutzte Schwachstelle befindet. Dieses Vorgehen kann den Angreifer/innen auch ermöglichen von einer virtuellen Umgebung, wie einer virtuellen Maschine (VM) oder einem Container, auf den darunterliegenden Host zu kommen. Eine andere Möglichkeit dieser Technik ist es, wenn die Angreifer/innen ihre eigenen Treiber mit einer Schwachstelle mitbringen. Dies wird of als "Bring your own vulnerable driver" (BYOVD) bezeichnet. Dabei wird der Treiber meist schon in der Initial Access Phase vom Opfer heruntergeladen oder auf einem bereits kompromittierten System über die Techniken Ingress Tool Transfer oder Lateral Tool Transfer heruntergeladen [88].

Brute Force

Brute Force wird von Angreifer/innen dann genutzt, wenn Passwörter für Account unbekannt sind oder wenn sie in den Besitz von Passworthashes gelangt sind. Das systematische Erraten der Passwörter kann dabei online oder offline erfolgen. Online erfolgt es dann, wenn die Passwörter mithilfe der Interaktion mit einem Service erfolgen, welcher das Passwort überprüft. Offline kann die Aktion nur gegen zuvor erlangte Passworthashes erfolgen. Brute Force kann in unterschiedlichen Phasen eines Angriffs angewandt werden. So können Angreifer/innen beispielsweise versuchen, den Zugriff auf gültige Konten in einer Opferumgebung durch Brute-Force-Verfahren zu erzwingen, indem sie das Wissen nutzen, das sie durch andere Verhaltensweisen nach der Kompromittierung gesammelt haben, wie z. B. OS Credential Dumping, Account Discovery oder Password Policy Discovery. Angreifer/innen können auch Brute-Forcing-Aktivitäten mit Verhaltensweisen wie externen Remote-Diensten als Teil des Erstzugriffs kombinieren [89].

6.3.2 Analyse der Logquellen

Die Logquelle, die benötigt wird, um die beschriebenen Use Cases umsetzen zu können sind die Windows Event Logs der Domain Controller und, um ein ganzheitliches Monitoring zu gewährleisten, auch jene der Endgeräte. Eine weitere Möglichkeit wären Sysmon Logs. Sysmon muss allerdings extra installiert werden, aus diesem Grund werden diese hier nicht betrachtet.

Windows Event Logs

Bei den Windows Event Logs handelt es sich um die Aufzeichnung von Events, die auf dem jeweiligen System stattfinden und dabei Benachrichtigungen und Alarme erzeugen. Microsoft definiert ein Event dabei als jedes signifikante Ereignis im System oder in einem Programm, das eine Benachrichtigung der Benutzer oder ein Eintrag in das Protokoll erfordert [90]. Seit Windows Vista und Windows Server 2008 werden Windows Event Logs im EVTX-Dateiformat unter "C:\Windows\System32\winevt\Logs" gespeichert. Dabei handelt es sich um ein proprietäres Format, das durch das Microsoft Management Console (MMC) Snap-In "eventvwr.msc" gelesen werden kann. Seit der Umstellung von EVT auf das EVTX Format, besteht zudem einen Unterstützung für das Extensible Markup Language (XML) Format. Dies ist besonders wichtig, wenn es um den Aspekt der Log Collection geht [91].

Das Event Log enthält die Logs der folgenden Kategorien sowie benutzerdefinierte Logs [92]:

- Application: Das Application Log enthält die Ereignisse, die von Applikationen geloggt werden. Zum Beispiel würde hier ein Dateifehler einer Datenbankapplikation geloggt werden. Dabei entscheidet der Entwickler der Applikation, welche Ereignisse protokolliert werden sollen.
- Security: Das Security Log zeichnet Ereignisse, wie Login und Logoff auf, aber auch Events, die Ressourcen betreffen, so wie das Erstellen, Öffnen oder Löschen von Dateien und anderen Objekten. Die Auditfunktion im Security Log muss durch eine/n Administrator/in aktiviert werden.
- System: Das System Log protokolliert Ereignisse von System-Komponenten, wie das Fehlschlagen beim Laden eines Treibers oder anderer Komponenten während des Hochfahrens.
- CustomLog: Im CustomLog werden Ereignisse von Applikationen gespeichert, die ein CustomLog erstellen. Dabei kann durch die Applikation kontrolliert werden, wie groß das Logs sein soll und wie die Berechtigungen dafür aussehen.

In jeder dieser Kategorien können fünf verschiedene Typen an Ereignissen geloggt werden. Jeder der Typen hat definierte Daten, die in jedem Log enthalten sind sowie optionale event-spezifische Daten. Jedem Event

kann nur einer der folgenden Typen zugeordnet werden [93]:

- Error: Der Event Typ Error beschreibt Ereignisse, die ein signifikantes Problem, wie den Verlust von Daten oder der Funktionalität, darstellen.
- Warning: Ereignisse des Event Typs Warning sind nicht unbedingt signifikant, aber könnten zu Problemen in der Zukunft führen, beispielsweise bei niedrigem Speicherplatz. Wenn sich eine Anwendung von einem Ereignis ohne Funktions- oder Datenverlust erholen kann, kann sie das Ereignis im Allgemeinen als Event Typ Warning einstufen.
- Information: Information beschreibt das erfolgreiche Ausführen von Applikationen, Treibern oder Services. Zum Beispiel kann beim erfolgreichen Laden eines Netzwerktreibers ein Information Event geloggt werden. Allerdings ist es für eine Desktopapplikation unüblich, dass deren Start als Event geloggt wird.
- Success Audit: Ein Ereignis, das einen auditierten Sicherheitszugriffsversuch aufzeichnet, der erfolgreich war. Zum Beispiel wird der erfolgreiche Versuch eines Benutzers, sich am System anzumelden, als Success Audit-Ereignis protokolliert.
- Failure Audit: Als Failure Audit wird ein Ereignis geloggt, wenn ein auditierter Sicherheitszugriffsversuch fehlschlägt. Zum Beispiel, wenn ein Benutzer probiert auf einen Netzwerkpfad zuzugreifen und dieser Versuch fehlschlägt.

Jedes Windows Event Log Ereignis muss zumindest die nachfolgend beschriebenen Felder enthalten. Erstens muss enthalten sein, wann das Ereignis erfolgt ist (Datum und Uhrzeit), sowie der Name des eingeloggten Benutzers zu diesem Zeitpunkt. Außerdem muss der Computername enthalten sein, sowie die Event-ID, der Typ des Ereignisses und zuletzt die Quelle. Als Quelle wird jenes Programm oder jene Komponente verstanden, die das Event ausgelöst hat [90].

Um nicht zu viel Overhead zu generieren, sollte zum einen aus betrieblicher Sicht festgelegt werden, was auf den jeweiligen Geräten geloggt werden soll und zum anderen auch aus Sicht des Security-Analysten/in bestimmt werden, welche Events davon in das SIEM importiert werden sollen. Im folgenden werden jene Event-IDs hervorgehoben, die es jedenfalls benötigt, um die zuvor beschriebenen Use Cases abzudecken. Das bedeutet nicht, dass diese für andere Use Cases ausreichen sind oder eine ganzheitliche Sicht abbilden. Für ein vollumfängliches AD Security Monitoring wird es einen größeren Umfang an zu loggende Ereignisse benötigen.

Event-IDs	Beschreibung	Event Source	Link für weiterführende Informationen
4624	Ein Konto wurde erfolgreich angemeldet.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4624
4625	Ein Konto konnte sich nicht anmelden.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4625
4634	Ein Konto wurde abgemeldet.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4634
4647	Benutzer hat die Abmeldung initiiert.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4647
4648	Es wurde versucht, sich mit expliziten Anmeldeinformationen anzumelden.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4648
4672	Einer neuen Anmeldung wurden besondere Rechte zugewiesen.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4672
4673	Ein privilegierter Dienst wurde aufgerufen.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4673
4688	Ein neuer Prozess wurde erstellt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4688
4704	Eine Benutzerberechtigung wurde zugewiesen.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4704
4717	Einem Konto wurde der Zugriff auf die Systemsicherheit gewährt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4717
4718	Der Zugriff auf die Systemsicherheit wurde von einem Konto entfernt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4718
4720	Ein Benutzerkonto wurde erstellt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4720
4723	Es wurde versucht, das Kennwort eines Kontos zu ändern.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4723
4724	Es wurde versucht, das Kennwort eines Kontos zurückzusetzen.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4724

4727	Eine sicherheitsfähige globale Gruppe wurde erstellt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/audit-security-group-management
4728	Ein Mitglied wurde einer sicherheitsfähigen globalen Gruppe hinzugefügt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/audit-security-group-management
4729	Ein Mitglied wurde aus einer sicherheitsfähigen globalen Gruppe entfernt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/audit-security-group-management
4730	Eine sicherheitsaktivierte globale Gruppe wurde gelöscht.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/audit-security-group-management
4731	Eine sicherheitsaktivierte lokale Gruppe wurde erstellt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4731
4732	Ein Mitglied wurde einer sicherheitsfähigen lokalen Gruppe hinzugefügt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4732
4733	Ein Mitglied wurde aus einer sicherheitsfähigen lokalen Gruppe entfernt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4733
4734	Eine sicherheitsaktivierte lokale Gruppe wurde gelöscht.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4734
4738	Ein Benutzerkonto wurde geändert.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4738
4740	Ein Benutzerkonto wurde gesperrt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4740
4741	Ein Computerkonto wurde erstellt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4741
4742	Ein Computerkonto wurde geändert.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4742
4754	Es wurde eine sicherheitsfähige universelle Gruppe erstellt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/audit-security-group-management

4756	Ein Mitglied wurde einer sicherheitsfähigen universellen Gruppe hinzugefügt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/audit-security-group-management
4757	Ein Mitglied wurde aus einer sicherheitsfähigen universellen Gruppe entfernt.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/audit-security-group-management
4758	Eine sicherheitsfähige universelle Gruppe wurde gelöscht.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/audit-security-group-management
4768	Ein Kerberos-Authentifizierungsticket (TGT) wurde angefordert.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4768
4769	Ein Kerberos-Dienstticket wurde angefordert.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4769
4771	Fehler bei der Kerberos-Vorauthentifizierung.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4771
4781	Der Name eines Kontos wurde geändert.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4781
4794	Es wurde versucht, das Administratorkennwort für den Verzeichnisdienstwiederherstellungsmodus festzulegen.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4794
5136	Ein Verzeichnisdienstobjekt wurde geändert.	Microsoft-Windows-Security-Auditing	https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-5136

Tabelle 6.1: Benötigte Event-IDs

Nachdem klar ist, welche Event-IDs für die Use Cases benötigt werden, muss noch festgelegt werden welche Enrichment-Quellen benötigt werden. Bei Active Directory ist es sinnvoll dieses als Enrichment Quelle anzubinden, um bestimmte Logs mit bestimmten Feldern anzureichern. Zum Beispiel kann es hilfreich sein, einen Benutzernamen um den DisplayName zu erweitern, damit sofort klar ist, um welchen User es sich handelt. Auch Attribute, wie LastLogon und der UserAccountControl-Wert (UAC) können hilfreiche Informationen bieten. Des weiteren bietet auch der Distinguished Name oft ergänzende Informationen beispielsweise in welcher Abteilung sich ein Benutzer befindet. Dies setzt allerdings voraus, dass sich die

Benutzer in OUs gemäß ihrer Abteilung befinden. Neben den Active Directory Informationen ist es zudem sehr hilfreich die DNS-Namen der Source-IPs aufzulösen. Auf diese Weise kann das Gerät für den/die Analyst/in auf einen Blick erkannt und verarbeitet werden.

6.3.3 Anbindung der Logquellen

Nachdem nun klar ist, welche Logs angebunden werden sollen, müssen sowohl die Logquellen als auch das SIEM entsprechend konfiguriert werden. Zuerst müssen die Auditing Policies am Domain Controller und auf den Endgeräten konfiguriert werden. Dabei wird zwischen Basic Security Audit Policies und Advanced Security Audit Policies unterschieden. Die Basic Security Audit Policy ermöglicht es bestimmte Kategorien des Security Auditing zu aktivieren. Bei diesen Kategorien handelt es sich um folgende [94]:

- Anmeldeereignisse
- Anmeldeversuche
- Kontenverwaltung
- Objektzugriffsversuche
- Prozessnachverfolgung
- Rechteverwendung
- Richtlinienänderungen
- Systemereignisse
- Verzeichnisdienstzugriff

Dabei ist zu beachten, dass die Basic Security Audit Policies auf Domain Controllers standardmäßig aktiviert sind, auf Workstations sind diese allerdings undefiniert [94].
Um mehr Kontrolle darüber zu haben, welche Events geloggt werden sollen bzw. dies granularer zu bestimmen, können die Advanced Security Audit Policies verwendet werden. In diesen Policies können die oben beschriebenen Kategorien auf die eigenen Bedürfnisse abgestimmt und konfiguriert werden. Dazu kommt die Kategorie "Global Object Access Auditing" [95]. Während bei der Basic Security Audit Policy für eine Kategorie nur konfiguriert werden kann, ob "Success" und/oder "Failure" geloggt werden soll, kann bei der Advanced Security Audit Policy diese Einstellung für jedes Ereignis getroffen werden. Wird beispielsweise in der Basic Security Audit Policy die Kategorie Account Login Events so konfiguriert, dass alle erfolgreichen und fehlgeschlagenen Ereignisse geloggt werden, dann entspricht das dem selben Ergebnis, wie wenn in der Advanced Security Audit Policy in eben dieser Kategorie alle Ereignisse aktiviert sind [96].

Beide der Policies können zentral über Gruppenrichtlinien verwaltet werden. Allerdings ist zu beachten, dass die Policies nicht vermischt werden sollten. Es sollte nur eine der beiden genutzt werden, da es sonst zu unerwartetem Verhalten kommen kann [96].
Geht es nun um die Umsetzung des Loggings der im vorigen Schritt definierten Events, so kann dies am genauesten mit einer Advanced Security Audit Policy umgesetzt werden. In dieser müssten die in Tabelle 6.2 gelisteten Kategorien und Unterkategorien konfiguriert werden.

Kategorie	**Unterkategorie**	**Überwachungsereignis**	**Logquelle**
Anmelden/Abmelden	Anmelden überwachen	Erfolg, Fehler	DC, Client
Anmelden/Abmelden	Abmelden überwachen	Erfolg	DC, Client
Anmelden/Abmelden	Spezielle Anmeldung überwachen	Erfolg	DC, Client
Anmelden/Abmelden	Kontosperrung überwachen	Fehler	DC, Client
Berechtigungen	Sensible Verwendung von Rechten überwachen	Erfolg, Fehler	DC, Client
Detaillierte Überwachung	Prozesserstellung überwachen	Erfolg	DC, Client
Richtlinienänderung	Autorisierungsrichtlinienänderung überwachen	Erfolg	DC, Client
Richtlinienänderung	Authentifizierungsrichtlinienänderung überwachen	Erfolg	DC, Client
Kontenverwaltung	Benutzerkontenverwaltung überwachen	Erfolg, Fehler	DC, Client
Kontenverwaltung	Sicherheitsgruppenverwaltung überwachen	Erfolg	DC, Client
Kontenverwaltung	Computerkontoverwaltung überwachen	Erfolg	DC
Kontoanmeldung	Kerberos-Authentifizierungsdienst überwachen	Erfolg, Fehler	DC
Kontoanmeldung	Ticketvorgänge des Kerberos-Diensts überwachen	Erfolg, Fehler	DC
DS-Zugriff	Verzeichnisdienständerungen überwachen	Erfolg	DC

Tabelle 6.2: Advanced Security Audit Policy

Sind die Logging Policies auf den Endgeräten definiert, müssen die Logquellen an das SIEM angebunden werden. Bei dem LogPoint SIEM lässt sich das am besten durch die Installation des LogPoint Agent Collectors realisieren. Dabei muss beachtet werden, dass die Ports 7994 für Klartext-Kommunikation oder 7995 für verschlüsselte Kommunikation auf den Logquellen als Management Ports benötigt werden und deshalb auf eingehende Verbindungen hören müssen. Dementsprechend muss auch die Firewall konfiguriert werden.
Sobald die Agents installiert wurden, muss im SIEM das Plugin über die LogPoint Agent Collector .pak-Datei hochgeladen und installiert werden. Außerdem muss beachtet werden, dass auch am LogPoint Server je nach Konfiguration die Ports 7996 bis 7999 geöffnet werden müssen. Eine Definition wofür welche Ports benötigt werden, findet sich in der öffentlich zugänglichen LogPoint Dokumentation (siehe [97]).

Nach der erfolgreichen Installation der Agents, müssen die Policies im SIEM definiert werden. Zuerst gehört die Normalization Policy erstellt und darin festgelegt, dass der mit dem Agent mitgelieferte "LPA_Windows" Normalizer verwendet werden soll. Auf diese Weise wird sichergestellt, dass die Logs richtig normalisiert und dargestellt werden können. Hier ist auch anzumerken, dass die Logs zuvor bereits vom Agent in das JSON Format umgewandelt werden und somit im Raw-Log nicht das zuvor erwähnte XML-Format zu sehen ist. Aus diesem Grund macht es auch Sinn den "JSONCompiledNormalizer" ebenfalls als Fallback in die Policy zu schreiben. Wird ein zweiter Normalizer unter dem vorherigen in die Policy geschrieben, so wird dieser nur verwendet, sollte der erste nicht angewandt werden können.
Im nächsten Schritt kann eine Routing Policy definiert werden. Je nach Konzept können Logs hier nach Endpoints entsprechend in unterschiedliche Repos gespeichert werden oder man wählt eine andere Art der Aufteilung. In diesem Use Case machen zwei Repositories Sinn, eines für die Client und ein zweites für die Server Logs. So kann als Kriterium für die Routing Policy der Gerätenamen bzw. die Namenskonvention angewendet werden.
Die dritte Policy, die im Zuge der Konfiguration der Logquellen erstellt werden muss, ist die Enrichment Policy. Hier kann anhand gewünschter Kriterien, wie beispielsweise Event-ID oder Log-Typ gewählt werden, welche Quelle für das Enrichment herangezogen werden soll und über welches Feld die Einträge miteinander kombiniert werden sollen. Bevor diese Policy fertiggestellt werden kann, müssen die Quellen für das Enrichment konfiguriert werden. Für die definierten Use Cases braucht es zwei Enrichment Quellen, einmal das Active Directory und einmal DNS Lookup. Für das Active Directory Enrichment wird eine LDAP-Verbindung konfiguriert und definiert welche Attribute bezogen werden sollen. Zudem wird angegeben, wie oft die Objektinformationen aus dem AD bezogen werden sollen, beispielsweise alle 6 Stunden. Hilfreiche Attribute sind in folgenden Feldern enthalten:

- dn (Dinstinguished Name)
- userPrincipalName
- displayName
- sAMAccountName
- mail
- pwdLastSet
- lockoutTime
- userAccountControl
- lastLogon

Um das DNS Lookup zu konfigurieren, gibt es das vorgefertigte Template "IPtoHost" in dem lediglich

der Name des Feldes eingetragen werden muss, das die IP enthält und den Namen jenes Feldes, das den aufgelösten Hostname enthält. Dabei ist es hilfreich gleich drei IPtoHost Quellen festzulegen, um sowohl Source-Adresse, Destination-Adresse als auch Geräte-Adresse umwandeln zu können. Anzumerken ist hier auch, dass Enrichment-Quellen nicht nur während der Logverarbeitung im Zuge der Policy angewendet werden können sondern auch im Zuge einer Suchabfrage verwendet werden können. Aus diesem Grund wurde für diese Use Cases keine Enrichment-Quelle für das GeoIP-Lookup konfiguriert. Das GeoIP-Lookup wird für diese Use Cases nur in wenigen Fällen benötigt und würde somit während der Logverarbeitung zu viele Ressourcen verbrauchen, wo es nicht benötigt wird.
Wurden alle drei Policies konfiguriert, kann die letzte Policy, die Processing Policy definiert werden. Dabei muss lediglich angegeben werden, dass die drei zuvor kreierten Policies verwendet werden sollen. Diese Processing Policy kann, dann auf die jeweilige Logquelle angewandt werden. Dazu müssen die jeweiligen Geräte in der Geräteliste erstellt werden und dann unter dem verwendeten Collector die Processing Policy ausgewählt werden. Da das für eine große Menge an Geräten nicht praktikabel ist, gibt es zudem die Möglichkeit viele Geräte auf einmal hinzuzufügen und dabei gleich die Policy anzugeben bzw. ist es auch möglich eine CSV-Liste mit Geräten zu importieren. Eine alternative Möglichkeit ist es die Logs vor dem SIEM an einem Log Collector zu sammeln und gesammelt weiterzugeben. Dabei ist zu beachten, dass dieser redundant aufgebaut sein sollte bzw. ein Buffering verfügbar sein sollte, für den Fall, dass die Verbindung zum SIEM unterbrochen wird. Außerdem muss beachtet werden, dass das Feld für den Gerätenamen, dann jenen des Log Collectors angibt und es somit wichtig ist, auf den Hostnamen zu achten.
Wenn der Collector das erste Mal ausgewählt wird, fordert das System dazu auf ein Template für die Konfiguration der installierten Agents zu erstellen. In diesem Template können nun die Zertifikate hochgeladen werden, um eine verschlüsselte Kommunikation bzw. Transport der Logs zu ermöglichen. Außerdem wird hier festgelegt aus welcher Kategorie und aus welchen Levels die Logs bezogen werden sollen. Außerdem kann hier nochmals ein Filtering durchgeführt werden, welche Event-IDs inkludiert bzw. exkludiert werden sollen. Werden keine angegeben, werden alle Security Events verarbeitet, die von Windows geloggt werden. Zum Schluss gilt es noch zu beachten, dass bei den Geräten die richtige Zeitzone ausgewählt wird, um hier keine Fehler bei der Umrechnung der Zeitzonen zu generieren.

6.3.4 Erstellung der Alert Rules

Im folgenden Schritt wird ein Alert Rule Set erstellt, welches die beschriebenen Use Cases, soweit mit Windows Logs möglich, abdecken soll. Als Grundlage dazu wurden von LogPoint vordefinierte Alert Regeln herangezogen, an die vorhandenen Gegebenheiten angepasst, ergänzt sowie zusätzliche Regeln hinzuge-

fügt. In Tabelle 6.3 wird eine Übersicht der erstellten Regeln sowie die Use Cases die sie behandeln sollen gezeigt.

Nr.	Alert Regel Name	Beschreibung	Use Cases
1	Windows Authentication on Windows DC	Dieser Alert löst aus sobald sich jemand versucht am Domain Controller einzuloggen. Dabei spielt es keine Rolle, ob der Versuch erfolgreich war oder fehlgeschlagen ist.	Account Manipulation, Valid Accounts
2	Windows Failed Login Attempt Using Service Account	Der Alert wird ausgelöst, wenn sich ein/e Benutzer/in nicht mit einem Service Account anmelden konnte. Fehlgeschlagene Loginversuche mit dem Logon-Type 5 bedeuten für gewöhnlich, dass das Passwort eines Service Accounts geändert wurde ohne den Service zu aktualisieren, es besteht allerdings auch die Möglichkeit, dass ein/e Angreifer/in probiert hat diesen auszunutzen.	Valid Accounts
3	Windows Failed Login Attempt using Locked Out Account	Der Alert wird ausgelöst, wenn ein/e Benutzer/in versucht sich mit einem ausgesperrten Account einzuloggen.	Valid Accounts
4	Windows Failed Login Attempt using an Expired Account	Der Alert wird ausgelöst, wenn ein Benutzer versucht sich mit einem nicht mehr aktiven Account einzuloggen.	Valid Accounts
5	Windows Failed Login Followed by Lockout Event	Der Alert wird ausgelöst, wenn ein Loginversuch fehlschlägt und daraufhin der Account gesperrt wird. Das ist ein Anzeichen davon, dass möglicherweise versucht wird ein Passwort zu erraten.	Brute Force, Exploitation for Privilege Escalation, Valid Accounts
6	Windows Failed User Login Attempt	Der Alert wird ausgelöst, wenn der Loginversuch eines/r Benutzer/in fehlschlägt.	Brute Force, Exploitation for Privilege Escalation, Valid Accounts
7	Windows Kerberos Service Ticket Request	Der Alert wird ausgelöst, wenn ein Kerberos Service Ticket angefragt wird.	Valid Accounts
8	Windows Login Attempt on Disabled Account	Der Alert wird ausgelöst, wenn ein Loginversuch mit einem deaktiviertem Account durchgeführt wird.	Valid Accounts
9	Default Account Created but Password Not Changed	Der Alert wird ausgelöst, wenn ein neuer Account mit Default-Passwort erstellt wird und das Passwort dafür in den nächsten 24 Stunden nicht geändert wurde.	Account Manipulation, Valid Accounts
10	Default Brute Force Attack Attempt - Multiple Unique Sources	Der Alert wird ausgelöst, wenn für eine/n Benutzer/in mehrere Loginversuche von unterschiedlichen Quellen innerhalb von fünf Minuten erkannt werden.	Brute Force, Valid Accounts
11	Default Brute Force Attack Attempt - Multiple Unique Users	Der Alert wird ausgelöst, wenn von einer Quelle Loginversuche für mehr als fünf Benutzer/innen in fünf Minuten erfolgen.	Brute Force, Valid Accounts
12	Default Brute Force Attack Successful	Der Alert wird ausgelöst, wenn innerhalb von fünf Minuten mind. fünf fehlgeschlagene Loginversuche erfolgen und anschließend ein erfolgreicher Login verzeichnet wird.	Brute Force, Valid Accounts

13	Default Excessive Authentication Failures	Der Alert wird ausgelöst, wenn innerhalb von 10 Minuten mehr als 100 fehlgeschlagene Loginversuche erfolgt sind.	Brute Force, Valid Accounts
14	Default Guest Account Added to Administrative Group	Der Alert wird ausgelöst, wenn ein Gast-Account zu einer Sicherheitsgruppe hinzugefügt wird.	Account Manipulation, Valid Accounts
15	Default Successful Login Using a Default Account	Der Alert wird ausgelöst, wenn ein erfolgreicher Loginversuch von einem Default Account erfolgt ist. Dabei muss beachtet werden, dass eine Liste an Default Accounts von eingesetzter Software im Unternehmen erstellt werden muss.	Valid Accounts
16	Default Successful Login outside Normal Hour	Der Alert wird ausgelöst, wenn ein erfolgreicher Loginversuch außerhalb der normalen Arbeitszeit erfolgt ist.	Valid Accounts
17	Default Unusual Number of Failed Vendor User Login	Der Alert wird ausgelöst, wenn mehr als 10 fehlgeschlagene Loginversuche für Default Accounts verzeichnet werden. Dabei muss beachtet werden, dass eine Liste an Default Accounts von eingesetzter Software im Unternehmen erstellt werden muss.	Valid Accounts
18	Multiple Failed Login Followed by Successful Login Followed by Logoff	Der Alert wird ausgelöst, wenn in einem Zeitraum von 5 Minuten auf mehrere fehlgeschlagene Anmeldeversuche ein erfolgreicher Anmeldeversuch erkannt wird und anschließend daran ein Logoff verzeichnet wird.	Brute Force, Valid Accounts
19	Multiple Failed User Login Followed by Successful Login	Der Alert wird ausgelöst, wenn auf mehrere fehlgeschlagene Loginversuche eines/r Benutzer/in ein erfolgreicher Loginversuch folgt.	Brute Force, Valid Accounts
20	Possible Account Misuse-Abnormal Login	Der Alert wird ausgelöst, wenn das Anmelden eines Administrators außerhalb der Arbeitszeit erkannt wird.	Valid Accounts
21	Admin User Remote Logon Detected	Der Alert wird ausgelöst, wenn ein Remote-Login eines Administrators verzeichnet wird.	Valid Accounts
22	Enabled User Right in AD to Control User Objects	Der Alert wird ausgelöst, wenn die Berechtigung "SeEnableDelegationPrivilege" einem/r User/in zugewiesen wird. Damit wäre es dem/der betroffenen Benutzer/in möglich andere User Objekte zu kontrollieren.	Valid Accounts
23	User Login from unauthorized IP address	Der Alert wird ausgelöst, wenn ein Loginversuch aus einem unautorisiertem Land durchgeführt wird. Dazu müssen die autorisierten Länder in einer Liste abgespeichert werden.	Valid Accounts
24	Inactive User Accounts	Der Alert wird ausgelöst, wenn sich ein Benutzer länger als 30 Tage nicht eingeloggt hat.	Valid Accounts
25	Possible Account Misuse-Privilege Escalation	Der Alert wird ausgelöst, wenn an nicht privilegierte Benutzer/innen besondere Rechte vergeben werden (Event-ID 4672) oder sich diese versuchen mit expliziten Anmeldeinformationen anzumelden (Event-ID 4648).	Account Manipulation
26	Windows Account Creation followed by Group Add	Der Alert wird ausgelöst, wenn ein Benutzerkonto erstellt und zu einer Gruppe hinzugefügt wird.	Account Manipulation, Exploitation for Privilege Escalation

27	Windows Logon Rights Changes	Der Alert wird ausgelöst, wenn die Anmelderechte geändert wurden und beispielsweise einem/r Benutzer/in die Rechte zu "Auf diesen Computer vom Netzwerk aus zugreifen" oder "Anmelden als Dienst" gegeben wurden.	Account Manipulation
28	Windows Multiple Account Password changes by User	Der Alert wird ausgelöst, wenn ein/e Benutzer/in die Passwörter mehrere Benutzer/innen ändert.	Account Manipulation
29	Windows Multiple Password Changed by User	Der Alert wird ausgelöst, wenn ein Benutzer das eigene Passwort öfter als einmal in 5 Minuten ändert.	Account Manipulation, Exploitation for Privilege Escalation
30	Windows User Password Never Expires	Der Alert wird ausgelöst, wenn einem Benutzerkonto die Eigenschaft gegeben wurde, dass das Passwort nie abläuft (Event-ID 4731).	Account Manipulation
31	Windows User Account Change to End with Dollar Sign	Der Alert wird ausgelöst, wenn ein Benutzername geändert wurde und nun mit einem Dollar-Zeichen endet.	Account Manipulation
32	Windows User Added to Administrator Group	Der Alert wird ausgelöst, wenn ein/e Benutzer/in einer Administratorgruppe hinzugefügt wird.	Account Manipulation
33	Windows User Added to Domain Enterprise Admin	Der Alert wird ausgelöst, wenn ein/e Benutzer/in der Gruppe "Domain Admins" oder "Enterprise Admins" hinzugefügt wurde.	Account Manipulation
34	Windows User Removed from Administrator Group	Der Alert wird ausgelöst, wenn ein/e Benutzer/in aus einer Administratorengruppe entfernt wurde.	Account Manipulation
35	Windows User Removed from Domain Enterprise Admin	Der Alert wird ausgelöst, wenn ein/e Benutzer/in aus der Gruppe "Domain Admins" oder "Enterprise Admins" entfernt wurde.	Account Manipulation
36	Default Account privilege elevation followed by restoration of previous account state	Der Alert wird ausgelöst, wenn einem/r Benutzer/in Rechte zugewiesen wurden oder er/sie einer Gruppe zugewiesen wurde und diese kurz darauf wieder entzogen bzw. aus der Gruppe entfernt wurde.	Account Manipulation, Exploitation for Privilege Escalation
37	Default Configuration Change on Security Device	Der Alert wird ausgelöst, wenn Sicherheitskonfigurationen von Standard-Benutzer/innen ausgeführt werden.	Account Manipulation
38	Account Manipulated for Persistence Detected	Der Alert wird ausgelöst, wenn die Nutzung von "net"-Befehlen erkannt wird.	Account Manipulation
39	Password Change on DSRM Account Detected	Der Alert wird ausgelöst, wenn das Passwort für das Administratorkonto des Verzeichnisdienstwiederherstellungsmodus (DSRM-Account) geändert wurde.	Account Manipulation
40	Persistence Using Account Creation Detected	Der Alert wird ausgelöst, wenn die Erstellung eines Kontos mit Persistenz erkannt wird.	Account Manipulation
41	Persistence Using Account Manipulation Detected	Der Alert wird ausgelöst, wenn erkannt wird, dass per "net"-Befehl Benutzer/innen zu lokalen Gruppen hinzugefügt werden.	Account Manipulation
42	Windows User Account Lockout	Der Alert wird ausgelöst, wenn ein Benutzerkonto gesperrt wird.	Brute Force

43	Windows Successful Brute Force Attack from Same User	Der Alert wird ausgelöst, wenn 10 oder mehr fehlgeschlagene Anmeldeversuche gefolgt von einem erfolgreichen Anmeldeversuch des/der gleichen Benutzer/in erkannt wurden.	Brute Force
44	Windows Successful Brute Force Attack from Same Source	Der Alert wird ausgelöst, wenn 10 oder mehr fehlgeschlagene Anmeldeversuche gefolgt von einem erfolgreichen Anmeldeversuch vom selben Ursprung erkannt wurden.	Brute Force
45	Windows Multiple Unique Lockouts	Der Alert wird ausgelöst, wenn ein Benutzerkonto in einer Stunde öfter als einmal gesperrt wird.	Brute Force
46	Windows Multiple Failed Attempts against a Single Account	Der Alert wird ausgelöst, wenn mehr als ein fehlgeschlagener Loginversuch eines/r Benutzer/in erkannt wurde.	Brute Force
47	Windows Failed Interactive User Logins Detected	Der Alert wird ausgelöst, wenn mehr als 10 fehlgeschlagene interaktive Loginversuche an einem System erkannt werden.	Brute Force
48	Zerologon CVE-2020-1472 Exploitation Detected	Der Alert wird ausgelöst, wenn die mögliche Ausnutzung der Zerologon Schwachstelle auf einem Domain Controller erkannt wurde. Dabei wird überprüft, ob der Computer Account durch den Benutzer "ANONYMOUS LOGON" verändert wird. Da im AD standardmäßig jeder Computer alle 30 Tage das Passwort wechselt, kann es hier zu False Positives kommen.	Exploitation of Privilege Escalation
49	Active Directory Schema Change Detected	Der Alert wird ausgelöst, wenn ein Verzeichnisdienstobjekt verändert, erstellt, verschoben, gelöscht oder wiederhergestellt wird.	Exploitation of Privilege Escalation

Tabelle 6.3: Alert Rule Set

Um nun den Abdeckungsgrad nach dem MITRE ATT&CK-Framework feststellen zu können, bietet LogPoint eine eigene Ansichtsoption, die die verfügbaren Alert Rules für die jeweiligen Taktiken zeigt und auch darstellt, wie viele der jeweiligen Regeln zu einer Taktik gerade aktiv sind. In Abbildung 6.2 ist ein Ausschnitt der aktuellen sogenannten "Coverage View" zu sehen.

Wie in Abbildung 6.2 zu sehen ist, sind auch Techniken zum Teil abgedeckt, die nicht Teil der Use Cases waren. Das liegt daran, dass einige Alert Rules zur Erkennung mehrerer Techniken angewandt werden können. Beispielsweise überschneiden sich auch zwischen den gewählten Alert Rules teilweise die Techniken, wie in Tabelle 6.3 in der Spalte "Use Cases" zu sehen ist.

Da in Abbildung 6.2 nicht die gesamte Ansicht gezeigt werden kann, wird in folgender Tabelle nochmals zusammengefasst, welche Taktiken und Techniken mit welchem Abdeckungsgrad durch das Alert Rule

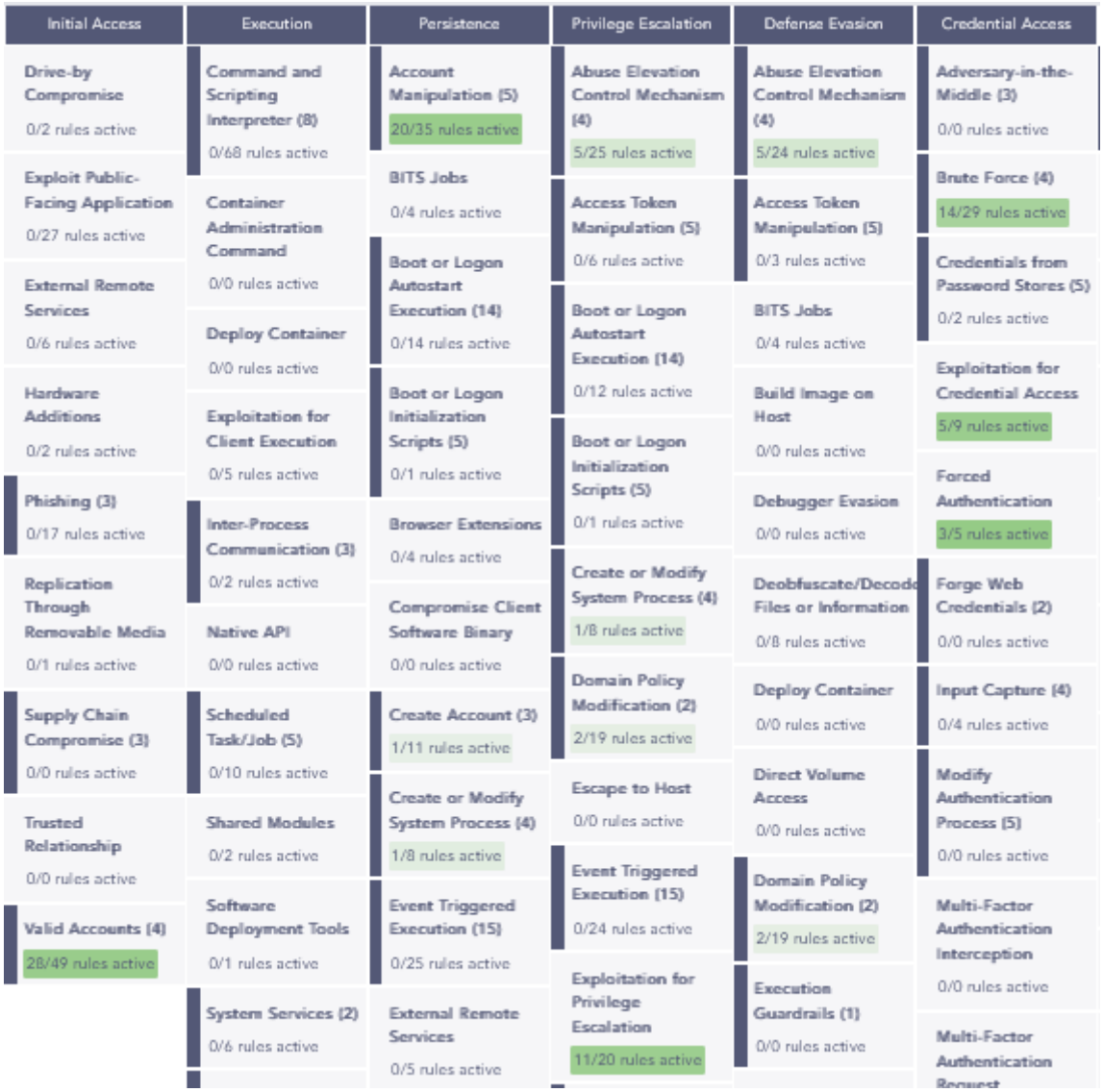

Abbildung 6.2: CoverageView mit aktuellem Alert Rule Set

Set aktuell abgedeckt sind. Der Abdeckungsgrad zeigt die Anzahl der aktuell aktivierten Alert Rules im Vergleich zu den für die Technik verfügbaren Alert Rules an. Unter die verfügbaren Alert Rules sind alle aktuell im System vorhandene Vorlagen und benötigen somit auch andere Quellen, wie Firewalls, IDS, Proxy oder ähnliches.

Taktik	**Technik**	**Abdeckungsgrad**
Initial Access	Valid Accounts	49 % (24/49)
Persistence	Account Manipulation	57 % (20/35)
Persistence	Create Account	9 % (1/11)
Persistence	Create or Modify System Process	13 % (1/8)

Persistence	Valid Accounts	49 % (24/49)
Privilege Escalation	Abuse Elevation Control Mechanism	21 % (5/24)
Privilege Escalation	Create or Modify System Process	13 % (1/8)
Privilege Escalation	Domain Policy Modification	11 % (2/19)
Privilege Escalation	Exploitation for Privilege Escalation	35 % (7/20)
Privilege Escalation	Valid Accounts	49 % (24/49)
Defense Evasion	Abuse Elevation Control Mechanism	21 % (5/24)
Defense Evasion	Domain Policy Modification	11 % (2/19)
Defense Evasion	Exploitation for Defense Evasion	9 % (2/23)
Defense Evasion	Valid Accounts	49 % (24/49)
Credential Access	Brute Force	48 % (14/29)
Credential Access	Exploitation for Credential Access	56 % (5/9)
Credential Access	Forced Authentication	60 % (3/5)
Credential Access	OS Credential Dumping	2 % (1/53)
Impact	Account Access Removal	13 % (1/8)

Tabelle 6.4: Coverage des aktuellen Alert Rule Sets

Wichtig ist hier anzumerken, dass dies eine Momentaufnahme darstellt. Sobald der nächste Schritt durchlaufen und festgestellt wird, dass bestimmte Alert Rules deaktiviert oder hinzugefügt werden müssen, kann sich auch der Abdeckungsgrad ändern. Außerdem hängt diese Darstellung stark vom verwendeten SIEM-Tool ab und wie die Alert-Rules den einzelnen Taktiken und Techniken zugeordnet werden. Dementsprechend gilt es auch die Bestimmung des Abdeckungsgrads an die Gegebenheiten anzupassen und die Vorgehensweise gegebenenfalls zu adaptieren.

6.3.5 Evaluierung der Alert Rules

Nach der Erstellung des Alert Rule Sets erfolgt die Evaluierung der erstellten Rules. Diese ist in zwei Schritten durchzuführen, zum einen muss die Funktionalität getestet werden, damit sichergestellt ist, dass im Falle einer Kompromittierung die Regel auch tatsächlich zieht. Zum anderen muss darauf geachtet werden, wie viele Ergebnisse eine Alert Rule liefert und wie viele Ereignisse davon wirklich verdächtig sind. Liefert eine Alert Rule zu viele False Positives, muss sie abgeändert werden bzw. kann es vorkommen, dass diese auch gänzlich deaktiviert wird.

Der erste Schritt, das Testen der Funktionalität, wird bei dem gegebenen Alert Rule Set in der beschriebenen Test Umgebung durchgeführt. Hier können beliebige Events generiert und somit die Alerts ausgelöst werden. Der zweite Schritt, die Effektivität der Alert Rules zu bestimmen, muss während des laufenden Betriebs immer wieder durchgeführt werden. Aus diesem Grund werden in dieser Arbeit die Beobachtungen

nach einer Woche in einer Produktivumgebung beschrieben, sowie eine Abschätzung des erwarteten Verhaltens abgegeben. Allerdings gilt hier zu beachten, dass die Einschätzung der Effektivität je nach Use Case und Regel variieren kann. Nur, weil eine Regel mehr False Positives erzeugt als eine andere heißt das nicht, dass diese weniger effektiv ist. Es kommt dabei immer darauf an, was das Ziel der Regel ist und ob das zu überprüfende Verhalten im Normalbetrieb vorkommen kann bzw. darf oder nicht. Somit muss dies auch jedes Unternehmen für sich beurteilen, welche Regeln für sinnvoll und effektiv gehalten werden und welche mehr Zusatzaufwand produzieren als sie wert sind.
Im folgenden Abschnitt werden die Alert Regeln getestet und auf ihre Effektivität hin überprüft.

Regel 1: Windows Authentication on Windows DC

Regel laut Vorlage:

```
label=Login (label=Fail OR label=Successful OR label=Attempt)
 device_name IN WINDOWS_DC
| rename target_user as user, target_domain as domain, description as
reason
```

Regel nach Anpassungen:

```
label=Login logon_type=2 -domain="Window Manager" (label=Fail OR
label=Successful) -target_user=*$ -user=*$ device_name IN WINDOWS_DC
| rename target_user as user, target_domain as domain, description as
reason
```

- **Funktionalität**

  Anpassungen: Um die richtige Funktionsweise dieser Regel sicherzustellen, muss die Aktualität der Liste `Windows_DC` regelmäßig überprüft werden. Es muss zudem ein Fehler in der Suchabfrage ausgebessert werden, um alle Ergebnisse zu erhalten. Dabei muss die Syntax für die Suche nach Label korrigiert werden und neben der Ausnahme für Service User im Feld `target_user`, diese Ausnahme auch für das Feld `user` ergänzt werden.

  Test: Die Funktionalität kann anschließend überprüft werden, indem sich ein/e Benutzer/in direkt am DC sowohl erfolgreich anmeldet als auch einen fehlgeschlagenen Anmeldeversuch produziert.

- **Effektivität**

  Zielerfüllung: Da die Alert Regel darauf abzielt jedes Login bzw. jeden Loginversuch am DC festzuhalten, unterstützt die Regel die Ziele der Use Cases Account Manipulation und Valid Accounts.

Durch das Alerting ist es möglich Logins durch unberechtigte Benutzer/innen oder zu ungewöhnlichen Zeiten sofort zu erkennen.

Redundanz: Es gibt aktuell keine weitere Regel, die einen ähnlichen Bereich abdeckt.

False Positives: Auch, wenn die Regel in der Testumgebung sinnvoll erscheint, erzeugt sie in der Produktivumgebung außerordentlich viel False Positives. Als Grund wurde identifiziert, dass alle Benutzer/innen, die sich in der Domäne einloggen, ein Login-Event mit dem Logon-Typ 3 (Netzwerk) am Domain Controller erzeugen, da automatisch Netzlaufwerke eingebunden werden. Aus diesem Grund kann die Regel auf den Logon-Typ 2 (Interaktiv) eingeschränkt werden, um diese Events auszuschließen. Außerdem erzeugen built-in Benutzer, wie "DWM-2" False Positives, weswegen `domain="Window Manager"` ausgenommen wird. Ist das bereinigt, bleiben noch die Logins mit expliziten Anmeldeinformationen, die übermäßig viele False Positives generieren. Um die Alert Regel weiterhin aktiv zu halten, wird entschieden den Teil `label=Attempt` herauszunehmen. Auch die Empfehlungen der Microsoft hinsichtlich der Event-ID 4648 sprechen nicht dagegen, da hier empfohlen wird Unregelmäßigkeiten, Namenskonventionen, bestimmte hoch privilegierte Konten und inaktive Accounts zu überwachen, aber nicht jedes Event [98].

Nach diesen Anpassungen sollten im Normalfall lediglich die Administratoren diese Alert Regel auslösen. Wenn diese ebenfalls ausgenommen werden sollen, kann `-user IN ADMINS` hinzugefügt werde. Dabei ist allerdings zu beachten, dass dann ungewöhnliche Logins von Administratoren aufgrund der Zeit, oder während einer Abwesenheit eines/r Mitarbeiters/in ebenfalls kein Alert erstellt wird.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 2: Windows Failed Login Attempt Using Service Account

Regel laut Vorlage:

```
norm_id=WinServer label=User label=Login label=Fail logon_type=5
-user=*$ -user IN EXCLUDED_USERS
```

- **Funktionalität**

  Test: Um die Funktionalität dieser Regel zu testen, wurde ein Service Account erstellt und anschließend ein Service (PSExecSVC) geändert, damit der eben erstellte Service Account benutzt wird. Danach wurde das Passwort des Service Accounts geändert und der Service neu gestartet, dies hat den fehlgeschlagenen Anmeldeversuch mit Logon-Typ 5 ausgelöst.

- **Effektivität**

  Zielerfüllung: Da diese Regel erfolgreich unübliches Verhalten in Form von fehlerhaften Service Logins erkennt, wird das Ziel des Use Cases Valid Accounts unterstützt.

  Redundanz: Es gibt weitere Regeln, die Failed Logins abdecken. Es gibt allerdings keine für fehlgeschlagene Service Logins.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

  Mögliche False Positives könnten entstehen, wenn das Passwort des Service Accounts geändert wird ohne den Service zu aktualisieren.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 3: Windows Failed Login Attempt using Locked Out Account

Regel laut Vorlage:

```
norm_id=WinServer label=User label=Login label=Fail
sub_status_code="0xC0000234" -target_user=*$ -user=*$ -user IN
EXCLUDED_USERS
| rename target_user as user, target_domain as domain
```

Regel nach Anpassungen:

```
norm_id=WinServer label=User label=Login label=Fail
status_code="0xC0000234" -target_user=*$ -user=*$ -user IN
EXCLUDED_USERS
| rename target_user as user, target_domain as domain
```

- **Funktionalität**

  Anpassungen: Beim Testen wurde festgestellt, dass `sub_status_code` durch `status_code` ersetzt werden muss.

  Test: Um die Funktionalität zu testen, wird zuvor die "Account Lockout Policy" im "Group Policy Management" konfiguriert. Danach wird sich so oft mit einem/r neu angelegten Benutzer/in eingeloggt, dass diese Policy ausgelöst und der/die Benutzer/in gesperrt wird. Danach muss noch mindestens ein Loginversuch erfolgen.

- **Effektivität**

  Zielerfüllung: Diese Regel prüft erfolgreich, ob jemand versucht sich mit einem bereits ausgesperrten Account einzuloggen. Dadurch können potentiell kompromittierte Accounts erkannt werden. Somit unterstützt die Regel den Use Case der Valid Accounts.

  Redundanz: Es gibt weitere Regeln, die fehlgeschlagene Logins abdecken, sowie eine weitere die die Sperrung eines Accounts abdeckt (Regel 44) und eine die davor prüft, ob ein nicht erfolgreicher Anmeldeversuch vor der Sperrung durchgeführt wurde (Regel 5). Zudem gibt es noch Regel 47, die prüft ob in einem bestimmten Zeitraum mehrere Lockouts erfolgt sind. Trotz ähnlicher Regeln, decken diese nicht den Versuch sich mit einem gesperrten Account anzumelden ab.

  False Positives: Die Alert Regel hat in einer Woche Laufzeit eine geringe Anzahl an Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen. Da hier False Positives von Benutzer/innen ausgelöst werden, die ihr Passwort vergessen oder kürzlich geändert haben. Allerdings sprechen wiederholte Anmeldeversuche bei einem gesperrten Account eher für einen Versuch diesen Account zu missbrauchen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 4: Windows Failed Login Attempt using an Expired Account

Regel laut Vorlage:

```
norm_id=WinServer* label=User label=Login label=Fail
sub_status_code="0xC0000193" -target_user=*$ -user=*$ -user IN
EXCLUDED_USERS
| rename target_user as user, target_domain as domain
```

- **Funktionalität**

  Test: Um diese Regel zu testen, wurde ein neuer Account angelegt und dessen Ablaufdatum auf den Tag darauf festgelegt. Danach wurde versucht sich mit dem/der erstellten Benutzer/in anzumelden.

- **Effektivität**

  Zielerfüllung: Diese Regel prüft erfolgreich, ob jemand versucht sich mit einem abgelaufenen Account einzuloggen. Dadurch können potentiell kompromittierte Accounts erkannt werden. Somit unterstützt die Regel den Use Case der Valid Accounts.

  Redundanz: Es gibt weitere Regeln, die fehlgeschlagene Logins abdecken. Allerdings gibt es keine für fehlgeschlagene Loginversuch durch abgelaufene Accounts.

False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 5: Windows Failed Login Followed by Lockout Event

Regel laut Vorlage:

```
[norm_id=WinServer label=User label=Login label=Fail -user IN
EXCLUDED_USERS] as s1
followed by
[norm_id=WinServer label=User label=Account label=Lock user=*] as s2
on s1.user=s2.user
| rename s1.user as User, s1.source_address as SourceAddress,
s2.workstation as ComputerName, s2.caller_domain as Domain,
s1.log_ts as LastFailedLogin_ts, s2.log_ts as LockedOut_ts
```

- **Funktionalität**

  Test: Um die Funktionalität zu testen, wird zuvor die "Account Lockout Policy" im "Group Policy Management" konfiguriert. Danach wird sich so oft mit einem/r neu angelegten Benutzer/in eingeloggt, dass diese Policy ausgelöst und der/die Benutzer/in gesperrt wird.

- **Effektivität**

  Zielerfüllung: Diese Regel soll in drei der vier gewählten Use Cases zum Einsatz kommen. Durch die Erkennung, ob durch zu viele Logins der Accounts gesperrt wurde, lässt sich zum einen erkennen, dass möglicherweise versucht wird das Passwort zu erraten (Brute Force), um in weiterer Folge zu höheren Rechten zu kommen (Exploitation for Privilege Escalation) bzw. stellt dies unübliches Verhalten in Bezug auf Benutzerkonten dar (Valid Accounts).

  Redundanz: Es gibt weitere Regeln, die fehlgeschlagene Logins abdecken, sowie eine weitere die die Sperrung eines Accounts abdeckt (Regel 42) und eine die prüft, ob ein nicht erfolgreicher Anmeldeversuch mit einem gesperrten Account getätigt wurde (Regel 3). Zudem gibt es noch Regel 45, die prüft ob in einem bestimmten Zeitraum mehrere Lockouts erfolgt sind. Im Gegensatz zur Regel 42, ist in diesem Alert durch das vorangehende fehlgeschlagene Login sofort ersichtlich, woher das Lockout ausgelöst wurde und erleichtert somit die Untersuchung.

  False Positives: Die Alert Regel hat in einer Woche Laufzeit eine geringe Anzahl an Alerts produziert,

somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen. Da hier False Positives von Benutzer/innen ausgelöst werden, die ihr Passwort vergessen oder kürzlich geändert haben.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 6: Windows Failed User Login Attempt

Regel laut Vorlage:

```
norm_id=WinServer* label=User label=Login label=Fail -target_user=*$
-user IN EXCLUDED_USERS
| rename target_user as user, target_domain as domain
```

- **Funktionalität**

  Test: Bei der Eingabe eines falschen Passworts wird dieser Alert ausgelöst.

- **Effektivität**

  Zielerfüllung: Diese Regel soll in drei von vier gewählten Use Cases zum Einsatz kommen. Allerdings ist ein falscher Login etwas zu vage, um einen sinnvollen Alert für die Use Cases Brute Force, Exploitation for Privilege Escalation und Valid Accounts darzustellen. Aus einem einzigen fehlgeschlagenen Anmeldeversuch kann kein unübliches oder verdächtiges Verhalten abgeleitet werden.

  Redundanz: Diese Regel ist redundant, da es einige Regeln gibt, die genauere Szenarien abbilden.

  False Positives: Diese Regel hat in einer Woche eine sehr hohe Anzahl an False Positives erzeugt. Grund dafür ist, dass jeder fehlgeschlagene Anmeldeversuch einen Alert generiert und User/innen sich dabei öfter vertippen.

- **Fazit**

  Diese Regel erfüllt keine der drei Kriterien und wird daher deaktiviert.

Regel 7: Windows Kerberos Service Ticket Request

Regel laut Vorlage:

```
norm_id=WinServer* label=Kerberos label=Service label=Request
result_code=* OR status_code=* -user IN EXCLUDED_USERS
| rename result_code as status_code
```

Regel nach Anpassungen:

```
norm_id=WinServer* label=Kerberos label=Service event_id=4769
result_code=* OR status_code=* -user IN EXCLUDED_USERS
| rename result_code as status_code
```

- **Funktionalität**

  Anpassungen: Bei dieser Regel wurde festgestellt, dass die hinterlegte Regel nicht die Service-Ticket Requests prüft sondern die Requests für Ticket-Granting-Tickets. Hier gibt es nun drei Möglichkeiten. Entweder die Regel wird belassen, wie sie ist und umbenannt, oder die Regel wird korrigiert, indem `label=Request` herausgenommen und `event_id=4769` hinzugefügt wird. Die letzte Möglichkeit wäre die Regel zu deaktivieren, da es wenig Sinn macht jedes Service-Ticket oder TGT-Request für jede/n User/in zu alerten (siehe Effektivität).

  Test: Die Regel wird ausgelöst, sobald der/die Benutzer/in eine Ressource in der Domäne anfragt und sich dafür authentifizieren muss.

- **Effektivität**

  Zielerfüllung: Diese Regel erfüllt grundsätzlich das Ziel des Use Cases Valid Accounts, da es theoretisch möglich wäre, auf diese Weise verdächtige Aktivitäten festzustellen. In der Praxis ist das allerdings nicht praktikabel und sollte zwar auf jeden Fall geloggt, allerdings nicht alarmiert werden.

  Redundanz: Es gibt aktuell keine weitere Regel, die einen ähnlichen Bereich abdeckt.

  False Positives: Diese Regel hat in einer Woche eine sehr hohe Anzahl an False Positives erzeugt. Grund dafür ist, dass jede Anfrage eines Service-Tickets alarmiert wird. Dies ist für sich nichts ungewöhnliches. Alternativ könnten bestimmte hoch-privilegierte Benutzer/innen überwacht werden.

- **Fazit**

  Diese Regel erfüllt keine zwei der drei Kriterien und wird daher deaktiviert.

Regel 8: Windows Login Attempt on Disabled Account

Regel laut Vorlage:

```
norm_id=WinServer* label=User label=Login label=Fail
sub_status_code="0xC0000072" -target_user=*$ -user=*$ -user IN
EXCLUDED_USERS
| rename target_user as user, target_domain as domain
```

- **Funktionalität**

Test: Um diese Regel zu testen, wurde ein neuer Account angelegt und bei der Erstellung ein Haken bei "Account is disabled" gesetzt. Danach wurde versucht sich mit dem/der erstellten Benutzer/in anzumelden.

- **Effektivität**

  Zielerfüllung: Diese Regel prüft erfolgreich, ob jemand versucht sich mit einem deaktivierten Account einzuloggen. Dadurch können potentiell kompromittierte Accounts erkannt werden. Somit unterstützt die Regel den Use Case der Valid Accounts.

  Redundanz: Es gibt weitere Regeln, die fehlgeschlagene Logins abdecken, allerdings gibt es keine weitere Regel, die auf Anmeldeversuche durch deaktivierte Benutzerkonten prüft.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 9: Default Account Created but Password Not Changed

Regel laut Vorlage:

```
[label=User label=Create label=Account] as s1
left join
[label=User label=Password (label=Change OR label=Reset)] as s2
on s1.target_user=s2.user
| search -s2.user=*
| rename s1.target_user as User, s1.log_ts as UserCreated_ts
| process current_time(a) as time_ts
| chart max((time_ts - UserCreated_ts)/60/60) as Duration by User,
UserCreated_ts
| search Duration > 24
```

- **Funktionalität**

  Test: Um nicht 24 Stunden warten zu müssen, damit die Funktionalität getestet werden kann, wurde diese vorübergehend auf 1 Stunde heruntergesetzt. Da durch die zuvor getesteten Regeln Accounts erstellt wurden, deren Passwort nicht geändert wurde. Schienen diese auf und hätten die Regel ausgelöst bzw. werden die Regel nach 24 Stunden auslösen.

- **Effektivität**

  Zielerfüllung: Diese Regel prüft, ob Benutzerkonten erstellt werden, deren Passwörter danach nicht geändert wurden. Das kann ein Anzeichen dafür sein, dass diese nicht ordnungsgemäß den Richtlinien erstellt wurden und unterstützt somit die Ziele des Use Cases Valid Accounts.

  Redundanz: Es gibt aktuell keine weitere Regel, die einen ähnlichen Bereich abdeckt.

  False Positives: Die Regel erzeugt wenige False Positives. Der Grund dafür ist, dass Service Accounts erstellt wurden, die kein Passwort hinterlegt haben bzw. das Attribut "Password not required - Enabled" gesetzt ist. Sofern diese Fälle nicht den unternehmensinternen Richtlinien widersprechen, kann dies mit `-user_account_control="*Password Not Required - Enabled*"` als Ausnahme hinzugefügt werden.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 10: Default Brute Force Attack Attempt - Multiple Unique Sources

Regel laut Vorlage:

```
label=User label=Login label=Fail
| rename target_user as user
| chart distinct_count(source_address) as DC by user
| search DC > 5
```

Regel nach Anpassungen:

```
label=User label=Login label=Fail -user=*$
| rename target_user as user
| chart distinct_count(device_ip) as DC by user
| search DC > 5
```

- **Funktionalität**

  Anpassungen: Beim Testen der Regel ist aufgefallen, dass in den Endpoint Logs mit der Event-ID 4625 als Source-Adresse die Loopback Adresse 127.0.0.1 eingetragen wird und somit diese Regeln nicht wie gedacht funktionieren kann.

  Ein mögliche Lösung ist, dass die `source_address` durch die `device_ip` ersetzt wird. Auf diese Weise wird die IP von der Maschine verwendet, von der das Log kommt. Dabei muss allerdings beachtet werden, dass die Logs direkt vom Endpoint kommen und sie nicht zuvor von einem Log

Collector gesammelt und dann ans SIEM gesendet werden.

Tests: Um verschiedene Logs von verschiedenen IP-Adressen zu bekommen, müssen weitere Endpunkte erstellt und der Agent installiert werden, um diese Endpunkte an das SIEM anzuschließen. Anschließend muss ein fehlerhafter Anmeldeversuch auf allen Endpunkten mit dem/der selben Benutzer/in generiert werden. Zu Testzwecken, kann die Anzahl auch heruntergesetzt werden, um den Aufwand zu verringern.

- **Effektivität**

  Zielerfüllung: Da die Regel prüft, ob von mehreren Quellen versucht wird sich einzuloggen und dies im Normalfall nicht üblich ist, ist das ein Anzeichen, dass probiert wird das Passwort eines Accounts zu knacken. Somit unterstützt die Regel das Ziel der Use Cases Brute Force und Valid Accounts.

  Redundanz: Es gibt weitere Regeln, die sich mit Brute Force befassen. Allerdings gibt es keine weitere Regel, die prüft, ob die Loginversuche von unterschiedlichen Quellen kommen.

  False Positives: Die Regel hat zu Beginn eine hohe Anzahl an False Positives geliefert. Der Grund für 99 Prozent der Alerts waren Computerkonten. Da diese Regel den Fokus auf Benutzerkonten haben soll, wird mit `-user=*$` eine Ausnahme hinzugefügt. Danach konnte nur mehr eine geringe Anzahl an False Positives festgestellt werden.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der zweiten angepassten Version aktiviert.

Regel 11: Default Brute Force Attack Attempt - Multiple Unique Users

Regel laut Vorlage:

```
label=User label=Login label=Fail source_address=* -target_user=*$
| rename target_user as user
| chart distinct_count(user) as DC by source_address
| search DC > 5
```

Regel nach Anpassungen:

```
label=User label=Login label=Fail device_ip=* -target_user=*$
-user=*$
| rename target_user as user
| chart distinct_count(user) as DC by device_ip
| search DC > 5
```

- **Funktionalität**

  Anpassungen: Hier gilt das gleichen wie bei Regel 10. Somit muss `source_address` durch `device_ip` ausgetauscht werden. Andernfalls fasst die Regel alle Logins unter der Adresse 127.0.0.1 zusammen und führt so zu False Positives.

  Zudem muss `-user=*$` noch als Ausnahme hinzugefügt werden.

  Test: Um diese Regel in der zweiten Version zu testen, müssen mit mindestens fünf verschiedenen Benutzern/innen fehlerhafte Anmeldeversuche von der selben Workstation generiert werden. Wenn noch keine fünf verschiedenen Testnutzer/innen erstellt wurden, muss dies vorher erledigt werden.

- **Effektivität**

  Zielerfüllung: Da die Regel prüft, ob von einer Quellen versucht wird sich mit mehreren Benutzer/innen einzuloggen und dies im Normalfall nicht üblich ist, ist das ein Anzeichen, dass probiert wird einen Account zu kompromittieren. Somit unterstützt die Regel das Ziel der Use Cases Brute Force und Valid Accounts.

  Redundanz: Es gibt weitere Regeln, die sich mit Brute Force befassen. Allerdings gibt es keine weitere Regel, die prüft, ob sich mehrere Benutzer/innen erfolglos versuchen von einer Quelle anzumelden.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit nur eine geringe Anzahl an False Positives produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der zweiten angepassten Version aktiviert.

Regel 12: Default Brute Force Attack Successful

Regel laut Vorlage:

```
[label=User label=Login label=Fail -target_user=*$
| rename target_user as user
| chart count() as cnt by user
| search cnt >5] as s1
followed by
[label=User label=Login label=Successful
| rename target_user as user] as s2
on s1.user=s2.user
```

```
| rename s2.user as User
```

Regel nach Anpassungen:

```
[label=User label=Login label=Fail -target_user=*$ -user=*$
| rename target_user as user
| chart count() as cnt by user
| search cnt >5] as s1
followed by
[label=User label=Login label=Successful
| rename target_user as user] as s2
on s1.user=s2.user
| rename s2.user as User
```

- **Funktionalität**

  Anpassungen: Um Computer Accounts ganzheitlich auszunehmen, muss `-user=*$` hinzugefügt werden.

  Test: Um diese Regel zu testen, muss darauf geachtet werden, wie aktuell die "Account Lockout Policy" konfiguriert ist. Sperrt diese nach fünf Versuchen den/die Benutzer/in, so kann die Regel nur getestet werden, wenn die Policy geändert oder der Count in der Regel herabgesetzt wird. Wurde die Policy geändert, können mit einem/r Benutzer/in sechs fehlgeschlagene Loginversuche erzeugt werden, bevor sich erfolgreich eingeloggt wird.

- **Effektivität**

  Zielerfüllung: Da die Regel prüft, ob ein/e Benutzer/in nach wiederholten fehlerhaften Anmeldungen, sich erfolgreich anmelden kann, prüft diese auf unübliches Benutzerverhalten, das darauf hindeuten kann, dass ein Passwort erfolgreich erraten wurde. Somit unterstützt die Regel das Ziel der Use Cases Brute Force und Valid Accounts.

  Redundanz: Es gibt weitere Regeln, die sich mit Brute Force befassen. Speziell Regel 19 und Regel 43 testen das gleiche Szenario, nur unterscheiden sie sich durch die Anzahl fehlerhaften Loginversuche, auf die überprüft wird.

  False Positives: Die Alert Rule produziert eine sehr hohe Anzahl an False Positives. Der Grund scheint in der Syntax der Regel zu liegen, da Events öfter als einmal kombiniert werden.

- **Fazit**

  Da die Regel nur eines der drei Kriterien erfüllt, wird sie deaktiviert. Allerdings soll diese Regel

durch Regel 19 abgedeckt werden.

Regel 13: Default Excessive Authentication Failures

Regel laut Vorlage:

```
label=Fail label=Authentication -user=*$
| chart count() as cnt by user
| search cnt>100
```

- **Funktionalität**

  Test: In diesem Fall spielt die "Account Lockout Policy" keine Rolle, da nur auf fehlgeschlagene Authentifizierungen geachtet wird, diese werden auch für bereits ausgesperrte Benutzer/innen generiert. Dies kann getestet werden, indem fehlgeschlagene Logins erzeugt werden. Zu Testzwecken kann die Anzahl in der Query auch heruntergesetzt werden.

- **Effektivität**

  Zielerfüllung: Da die Regel prüft, ob für eine/n Benutzer/in eine extrem hohe Anzahl an wiederholten fehlerhaften Anmeldungen stattfindet, kann unübliches Benutzerverhalten erkannt werden und deutet auf einen möglichen Brute Force Angriff hin. Somit unterstützt die Regel das Ziel der Use Cases Brute Force und Valid Accounts.

  Redundanz: Es gibt weitere Regeln, die sich mit Brute Force und fehlerhaften Anmeldeversuchen befassen. Allerdings prüft keine der Regeln auf eine exzessive Anzahl an Anmeldeversuchen. Ein weiterer Unterschied ist, dass diese Regel auf Event-ID 4771 und nicht auf Event-ID 4625 prüft.

  False Positives: Die Regel erzeugt sehr viele Alerts, wobei diese nicht als False Positives bezeichnet werden können. Nach Analyse der erzeugten Alerts wurde festgestellt, dass die Alerts meist mehrmals in kurzer Zeit für die gleichen Benutzer/innen erzeugt wurden. Das kann daran liegen, dass möglicherweise mit gespeicherten, nicht mehr aktuellen Anmeldeinformationen noch probiert wird ein TGT zu lösen. Aus diesem Grund wird hier Alert Throtteling aktiviert, um einen Alert für die selben Benutzer/innen erst wieder nach sechs Stunden erneut zu erzeugen. Auf diese Weise kann die Anzahl der Alerts vermindert und der Analyseaufwand reduziert werden.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 14: Default Guest Account Added to Administrative Group

Alert laut Vorlage:

```
label=Security label=Group label=Management label=Add
(member_sid="S-1-5-21-*-501" OR target_id="S1-5-21-*-501")
| rename target_user as member, group as group_name
```

Alert nach Anpassungen:

```
label=Security label=Group label=Management label=Add group="*-Admins"
(member_sid="S-1-5-21-*-501" OR target_id="S1-5-21-*-501")
| rename target_user as member, group as group_name
```

- **Funktionalität**

  Anpassungen: Die Regel entspricht standardmäßig nicht dem Alert Regel Namen, da sie lediglich überprüft, ob der Gast Account einer Gruppe zugeordnet wird. Um zu überprüfen, ob er einer administrativen Gruppe zugeordnet wird, ist Voraussetzung, dass es eine Namenskonvention gibt, die dafür verwendet werden kann. In diesem Beispiel wird von einer Namenskonvention ausgegangen, die wie folgt lautet "*-Admins".

  Test: Um die Regel zu überprüfen, muss lediglich eine neue Gruppe laut Namenskonvention angelegt und der Gast User dieser hinzugefügt werden.

- **Effektivität**

  Zielerfüllung: Da ein Gast Account in den meisten Fällen nicht verwendet werden sollte und noch weniger Mitglied einer administrativen Gruppe sein sollte, prüft diese Regel unübliches Benutzerverhalten sowie die Manipulation von Accounts, um höhere Rechte zu erreichen. Die Regel unterstützt somit die Ziele der Use Cases Valid Accounts und Account Manipulation.

  Redundanz: Es gibt weitere Regeln, die das Hinzufügen zu administrativen Gruppen prüfen. Regel 34 überprüft generell das Hinzufügen eines/r Benutzer/in zu einer Administrator-Gruppe. Da diese Regel einen spezifischeren Fall, der besonders selten bis nie im Normalfall eintreten sollte überwacht und somit eine wesentlich höhere Priorität hat, kann diese Regel weiterhin aktiviert bleiben.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 15: Default Successful Login Using a Default Account

Regel laut Vorlage:

```
label=User label=Login label=Successful (target_user=* OR user=*)
(target_user IN DEFAULT_USERS OR user IN DEFAULT_USERS)
| rename target_user as user
```

- **Funktionalität**

  Test: Um diese Regel zu testen, muss sichergestellt sein, dass die Liste `DEFAULT_USERS` den eigenen Anforderungen entsprechend adaptiert wurde. Zu Testzwecken kann entweder ein neues Benutzerobjekt erstellt und der Liste hinzugefügt werden oder es wird ein/e bestehende/r Test-Benutzer/in der Liste hinzugefügt und verwendet.

- **Effektivität**

  Zielerfüllung: Da keine Standardbenutzer/innen verwendet werden sollen, stellt jegliches Login eines/r solchen Benutzer/in unübliches Verhalten dar, das weiter untersucht werden sollte. Aus diesem Grund unterstützt die Regel die Ziele des Use Case Valid Accounts.

  Redundanz: Es gibt keine weitere Regel, die diese Aktivitäten überwacht.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

  Mögliche False Positives könnten hier entstehen, wenn Benutzer/innen die Standardbenutzer/innen beispielsweise aus Bequemlichkeit nutzen. Hier spielt die Komponente der Security Awareness im Unternehmen eine große Rolle.

  Des weiteren spielt hier eine kritische Rolle, dass möglichst viele der im Unternehmen vorhandenen Standardbenutzer/innen definiert werden können. Denn nur jene Konten, die in die Liste eingetragen wurden, können hier überwacht werden.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 16: Default Successful Login outside Normal Hour

Regel laut Vorlage:

```
label=Login label=Successful target_user=*((day_of_week(log_ts)=2 OR
day_of_week(log_ts)=3 OR day_of_week(log_ts)=4 OR day_of_week(log_ts)=5
OR day_of_week(log_ts)=6) (hour(log_ts)>0 hour(log_ts)<9) OR
hour(log_ts)>17) OR (day_of_week(log_ts) IN [1,7])
| rename target_user as user
```

Regel nach Anpassungen:

```
label=Login label=Successful target_user=* user=* -target_user=*$
-user=*$ workstation=* -logon_type=5 ((day_of_week(log_ts)=2 OR
day_of_week(log_ts)=3 OR day_of_week(log_ts)=4 OR day_of_week(log_ts)=5
OR day_of_week(log_ts)=6) (hour(log_ts)>0 hour(log_ts)<6) OR
hour(log_ts)>20) OR (day_of_week(log_ts) IN [1,7])
| rename target_user as user
```

- **Funktionalität**

  Anpassungen: Diese Regel muss bearbeitet werden, da die vorgegebene Regel keine Ergebnisse ausgibt. Um sie zu korrigieren, muss `user=*` hinzugefügt werden, da dies der für diese Logs richtige Feldname ist. Daraufhin kommen zu viele Ergebnisse somit muss dafür gesorgt werden, dass Computer bzw. Service Accounts ausgenommen werden (`(-target_user=*$ -user=*$ workstation=* -logon_type=5)`).

  Wichtig ist auch, dass die Zeiten den Arbeitszeiten des jeweiligen Unternehmens angepasst werden. Beispielsweise könnte hier der zulässige Gleitzeitrahmen, in diesem Fall wäre das 6 bis 20 Uhr, herangezogen werden.

  Test: Getestet werden kann, diese Regel indem sich erfolgreich außerhalb der definierten Zeiten eingeloggt wird.

- **Effektivität**

  Zielerfüllung: Die Regel zielt darauf ab ungewöhnliches Benutzerverhalten festzustellen, indem Logins zu Zeiten festgestellt werden, an denen die User/innen nicht arbeiten. Aus diesem Grund erfüllt die Regel die Ziele des Use Case Valid Accounts.

  Redundanz: Regel 20 deckt den selben Fall ab, allerdings eingeschränkt auf Administrator-User.

  False Positives: Diese Regel erzeugt in der Produktivumgebung eine sehr große Anzahl an False Positives. Der Grund dafür ist jener, dass in einem Teil des Unternehmens Schichtarbeit stattfindet und somit auch außerhalb der definierten Zeiten gearbeitet wird.

  Eine mögliche Lösung wäre die Aktivierung der Regel für einen Teilbereich des Unternehmens, wo keine Schichtarbeit stattfindet oder für eine höhere privilegierte Benutzergruppe, die besonders überwacht werden sollen, wie das in Regel 20 der Fall ist.

- **Fazit**

  Da die Regel nicht alle Kriterien voll erfüllt, wird sie deaktiviert. Allerdings sollen die erwähnten höher privilegierten Benutzer durch die Aktivierung der Regel 20 überwacht werden.

Regel 17: Default Unusual Number of Failed Vendor User Login

Regel laut Vorlage:

```
label=User label=Login label=Fail (target_user IN DEFAULT_USERS OR
user IN DEFAULT_USERS)
| rename target_user as user
| chart count() as Event by user, source_address
| search Event>10
```

Regel nach Anpassungen:

```
label=User label=Login label=Fail (target_user IN DEFAULT_USERS OR
user IN DEFAULT_USERS)
| rename target_user as user
| chart count() as Event by user, device_ip
| search Event>10
```

- **Funktionalität**

  Anpassungen: Bei dieser Regel besteht das gleiche Problem mit der Source Address wie zuvor bei Regel 10. Da diese in den meisten Fällen die Loopback-Adresse sein wird, kann diese Regel so nicht sinnvoll angewendet werden. Aus diesem Grund wird `source_address` durch `device_ip` ersetzt.

  So wie auch bei Regel 15 ist hier essentiell, dass eine Liste mit Default-User/innen erstellt wird.

  Test: Um die Regel zu testen, wird eine/r der Benutzer/innen auf der Liste dazu verwendet, sich mindestens 11 Mal falsch anzumelden. Es spielt dabei keine Rolle, wenn der/die Benutzer/in gesperrt wird.

- **Effektivität**

  Zielerfüllung: Da keine Standardbenutzer/innen verwendet werden sollen, stellt jeglicher Loginversuch eines/r solchen Benutzer/in unübliches Verhalten dar, das weiter untersucht werden sollte. Aus diesem Grund unterstützt die Regel die Ziele des Use Case Valid Accounts.

  Redundanz: Es gibt keine weitere Regel, die diese Aktivitäten überwacht.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

  So wie auch bei Regel 15 spielt hier eine kritische Rolle, dass möglichst viele der im Unternehmen

vorhandenen Standardbenutzer/innen definiert werden können. Denn nur jene Konten, die in die Liste eingetragen werden, können hier überwacht werden.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 18: Multiple Failed Login Followed by Successful Login Followed by Logoff

Regel laut Vorlage:

```
[incident_name="Multiple Failed User Login Followed by Successful
Login" incident_user=*] as FirstAlert
followed by
[norm_id=WinServer* label=User label=Logoff user=* -user IN
EXCLUDED_USERS] as Logoff
on FirstAlert.incident_user=Logoff.user
| rename Logoff.user as User, FirstAlert.incident_address as
SourceAddress
```

- **Funktionalität**

  Anpassungen: Um diese Regel verwenden zu können, muss zuerst sichergestellt werden, dass der `incident_name` genau dem Namen der Alert Regel 19 entspricht.

  Test: Damit die Regel getestet werden kann, muss zuerst die Alert Regel 19 "Multiple Failed User Login Followed by Successful Login" ausgelöst werden, indem mindestens vier fehlgeschlagene Anmeldeversuche erzeugt werden und anschließend ein erfolgreiches Login. Danach muss sich der/die Benutzer/in wieder abmelden.

- **Effektivität**

  Zielerfüllung: Das Verhalten, das diese Regel prüft könnte auf das Vorhaben das Passwort eines Accounts zu erraten hindeuten. Es kann somit unübliches Benutzerverhalten darstellen und unterstützt so die Ziele der Use Cases Brute Force und Valid Accounts.

  Redundanz: Es gibt ähnliche Regeln, die die Kombination von fehlgeschlagenen Anmeldeversuchen gefolgt von erfolgreichen Logins prüfen. Allerdings gibt es keine weitere Regel die auch prüft, ob ein sofortiges Logoff stattfindet.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

  Auch, wenn Regel 19 unbewusst von einem/r Benutzer/in ausgelöst wird, der/die sich mehrmals ver-

tippt, so ist es in weiterer Folge sehr unüblich, dass sich diese/r innerhalb der nächsten fünf Minuten sofort wieder ausloggt.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 19: Multiple Failed User Login Followed by Successful Login

Regel laut Vorlage:

```
[4 norm_id=WinServer* label=User label=Login label=Fail -user=*$
having same user] as Fail
followed by
[norm_id=WinServer* label=User label=Login label=Successful -user=*$
user=*] as Login
on Fail.user=Login.user
| rename user as User, Login.source_address as SourceAddress
```

Regel nach Anpassungen:

```
[4 norm_id=WinServer* label=User label=Login label=Fail -user=*$
having same user] as Fail
followed by
[norm_id=WinServer* label=User label=Login label=Successful -user=*$
user=*] as Login
on Fail.user=Login.user
| rename user as User, Login.source_address as SourceAddress
| chart count() by User, SourceAddress
```

- **Funktionalität**

  Anpassungen: Bei den Tests wurde festgestellt, dass die Felder `User` und `SourceAddress` nicht wie für die Regel 18 "Multiple Failed Login Followed by Successful Login Followed by Logoff" erforderlich, extrahierbar waren. Aus diesem Grund wurde der letzte Teil hinzugefügt.

  Test: Um diese Regel zu testen, muss darauf geachtet werden, wie aktuell die Account Lockout Policy konfiguriert ist. Anschließend wurde die Regel durch viermalige fehlerhafte Anmeldeversuche, gefolgt von einem erfolgreichen getestet.

- **Effektivität**

  Zielerfüllung: Das Verhalten, das diese Regel prüft könnte auf das Vorhaben das Passwort eines Accounts zu erraten hindeuten. Es kann somit unübliches Benutzerverhalten darstellen und unterstützt so die Ziele der Use Cases Brute Force und Valid Accounts.

  Redundanz: Diese Regel entspricht bis, auf die gesetzte Threshold, in ihrem Ergebnis, dem der Regel 12 und Regel 43.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit wenige False Positives produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.
  False Positives können hier auch durch fehlerhafte Passworteingaben generiert werden. Allerdings wird diese Regel als Trigger für Regel 18 gebraucht.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 20: Possible Account Misuse-Abnormal Login

Regel laut Vorlage:

```
label=User label=Login label=Successful user IN ADMINS
((day_of_week(log_ts) IN ["Monday", "Tuesday", "Wednesday", "Thursday",
"Friday"]) AND (hour(log_ts)<9 OR hour(log_ts)>17))
OR (day_of_week(log_ts) IN ["Saturday", "Sunday"])) OR (label=User
label=Login label=Successful sub_status_code="0xC000006F") user=*
(workstation=* OR source_address=*)
```

Regel nach Anpassungen:

```
label=User label=Login label=Successful -user=*$ user IN ADMINS
((day_of_week(log_ts) IN ["Monday", "Tuesday", "Wednesday", "Thursday",
"Friday"]) AND (hour(log_ts)<6 OR hour(log_ts)>20))
OR (day_of_week(log_ts) IN ["Saturday", "Sunday"])) OR (label=User
label=Login label=Fail sub_status_code="0xC000006F") user=*
(workstation=* OR source_address=*)
```

- **Funktionalität**

  Anpassungen: Es muss sichergestellt werden, dass die Namenskonventionen für Administratoruser/innen in der Liste `ADMINS` eingetragen wurden.

Außerdem muss im zweiten Teil der Regel `label=Successful` durch `label=Fail` ausgetauscht werden, da es den Sub-Status-Code der den Fehlergrund angibt, bei erfolgreichen Logins nicht gibt.

Test: Der erste Teil der Regel kann getestet werden, indem sich außerhalb der angegebenen Stunden eingeloggt wird bzw. zu Testzwecken die Zeiten geändert werden. Der zweite Teil kann getestet werden indem einem/r Admin-User/in zulässige Login-Zeiten eingetragen werden und außerhalb dieser Zeiten ein Loginversuch getätigt wird.

- **Effektivität**

  Zielerfüllung: Die Regel zielt darauf ab ungewöhnliches Benutzerverhalten festzustellen, indem Logins zu Zeiten festgestellt werden, an denen die User/innen im Normalfall nicht arbeiten. Aus diesem Grund erfüllt die Regel die Ziele des Use Case Valid Accounts.

  Redundanz: Es sind ähnliche Regeln, wie Regel 16. Allerdings wurde Regel 16 deaktiviert und die privilegierten Benutzer/innen sollen stattdessen mit dieser Regel überwacht werden.

  False Positives: Nach einer Woche Laufzeit wurde eine sehr hohe Anzahl an False Positives festgestellt. Dabei wurde identifiziert, dass die Ausnahme `-domain="NT AUTHORITY"` hinzugefügt werden muss. Zudem wurde festgestellt, dass die Zeiten nicht richtig gesetzt waren und Computer Accounts ausgenommen werden müssen.

  Auch hier kann Alert Throtteling eingesetzt werden, um die Anzahl der Alerts zu verringern.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 21: Admin User Remote Logon Detected

Regel laut Vorlage:

```
norm_id=WinServer event_id=4624 logon_type="10"
(authentication_package="Negotiate" OR package="Negotiate") user="Admin*" -user
| rename package as authentication_package
```

Regel nach Anpassungen:

```
norm_id=WinServer event_id=4624 logon_type="10"
(authentication_package="Negotiate" OR package="Negotiate") user IN
ADMINS -user IN EXCLUDED_USERS
| rename package as authentication_package
```

- **Funktionalität**

  Anpassungen: Da in der ursprünglichen Regel `user="Admin*"` konfiguriert wurde, wurde dies ausgebessert, um allen in der Liste definierten Adminkonten bzw. Namenskonventionen gerecht zu werden.

  Test: Um die Regel zu testen wurde am Client Remote Desktop Protokoll aktiviert und die entsprechenden Firewall regeln gesetzt. Anschließend konnte ein Login über RDP durchgeführt werden.

- **Effektivität**

  Zielerfüllung: Der Remotezugriff eines Administratorkontos kann von Angreifer/innen dazu missbraucht werden, sich im Netzwerk fortzubewegen oder die Privilegien weiter zu erhöhen. Somit unterstützt die Regel das Ziel des Use Cases Valid Accounts.

  Redundanz: Während es weitere Regeln gibt, die Logins überwachen, ist dies die einzige die Remote Logins überwacht.

  False Positives: Die Regel hat nach einer Woche eine sehr geringe Anzahl an Alerts produziert. Allerdings war zu erkennen, dass 12 Alerts innerhalb von 30 Minuten für den selben Benutzer/innen generiert wurden. Somit ist auch hier Alert Throtteling zu empfehlen. Der Zeitraum sollte hier allerdings nicht zu groß gewählt werden und wird deshalb auf eine Stunde gesetzt.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 22: Enabled User Right in AD to Control User Objects

Regel laut Vorlage:

```
norm_id=WinServer event_id=4704 message="SeEnableDelegationPrivilege"
-user IN EXCLUDED_USERS
```

Regel nach Anpassungen:

```
norm_id=WinServer event_id=4704 privilege="SeEnableDelegationPrivilege"
-user IN EXCLUDED_USERS
```

- **Funktionalität**

  Anpassungen: Beim Testen der Regel ist aufgefallen, dass die vergebenen Rechte nicht im Feld `message` stehen sondern im Feld `privilege`.

  Test:: Getestet werden kann die Regel indem im "Local Group Policy Editor" unter den "Computer

Configurations" bei "User Rights Assignment" das Recht "Enable computer and user accounts to be trusted for delegation" ein/e Benutzer/in hinzugefügt wird.

- **Effektivität**

  Zielerfüllung: Die Zuweisung dieses Recht sollte immer überprüft werden, da sie dem Benutzer erlaubt andere Objekte zu kontrollieren und das somit von Angreifern genutzt werden könnte. Aus diesem Grund unterstützt die Regel das Ziel des Use Case Valid Accounts.

  Redundanz: Es gibt keine weitere Regel, die diese Aktivität überwacht.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine False Positives produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 23: User Login from unauthorized IP address

Regel laut Vorlage:

```
label=User label=Login source_address=*
| process geoip(source_address) as country
| search -country IN KNOWN_COUNTRY
```

Regel nach Anpassungen:

```
label=User label=Login -user=*$ source_address=*
| process geoip(source_address) as country
| search -country IN KNOWN_COUNTRY
```

- **Funktionalität**

  Anpassungen: Hier muss sichergestellt werden, dass in der Liste `KNOWN_COUNTRY` "internal" eingetragen wird, sowie alle Länder, aus denen ein Login nicht untypisch wäre. Zudem sollten Computer Accounts ausgenommen werden.

  Test: Getestet wurde die Regel, indem "internal" aus der Liste herausgenommen und danach ein Login durchgeführt wurde. Danach kann "internal" wieder zur Liste hinzugefügt werden. Danach sollte kein Alert für eine Anmeldung über eine interne IP-Adresse ausgelöst werden.

  Die Regel funktioniert in der Form nur für IPv4, da die GeoIP-Auflösung für IPv6 Adressen bis dato nicht funktioniert.

- **Effektivität**

  Zielerfüllung: Ein Login aus einem nicht autorisierten Land, stellt ungewöhnliches Benutzerverhalten dar und unterstützt somit den Use Case Valid Accounts.

  Redundanz: Es gibt keine weitere Regel, die diese Aktivität überwacht.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine False Positives produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 24: Inactive User Accounts

Regel laut Vorlage:

```
table AD_USER -lastLogonTimestamp=0 lastLogonTimestamp=*
| process current_time(a) as time
| chart max((time-(lasLogon/10000000 - 11644473600))/60/60/24) as number_of_days by
| search number_of_days>29
```

- **Funktionalität**

  Anpassungen: Es gilt zu beachten, dass die Regel nur funktioniert, wenn die LDAP Enrichment Source konfiguriert wurde. Zudem muss der Name der Tabelle gegebenenfalls angepasst werden.
  Auch zu beachten ist, dass zwischen den Attributen "lastLogon" und "lastLogonTimestamp" unterschieden werden muss, vor allem, wenn mehr als ein DC im Einsatz ist. Denn "lastLogon" gibt das Datum an zu dem genau dieser DC den/die Benutzer/in zuletzt authentifiziert hat, aber nicht wann der/die Benutzer/in zuletzt in der Domäne gesehen wurde, da dieser Wert nicht repliziert wird.
  "lastLogonTimestamp" wird repliziert und zeigt den Timestamp des Tages an den der/die Benutzer/in zuletzt von einem der DCs authentifiziert wurde.

  Test: Um die Regel zu testen, wurde ein Logon mit einem Benutzer durchgeführt und dieser dann drei Tage nicht benutzt. In der Regel wurde die Anzahl der Tage zuvor auf 2 heruntergesetzt, damit nicht 30 Tage gewartet werden muss.

- **Effektivität**

  Zielerfüllung: Da inaktive Benutzer/innen von Angreifer/innen missbraucht werden können, zählt es zur Best Practice diese zu deaktivieren. Durch diese Regel können somit inaktive Benutzer/innen gefunden und Maßnahmen gesetzt werden. Somit unterstützt die Regel das Ziel des Use Case Valid Accounts.

Redundanz: Es gibt keine weitere Regel, die auf inaktive Benutzer/innen prüft.

False Positives: Nach einer Woche Laufzeit hat die Regel eine geringe Anzahl an Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 25: Possible Account Misuse-Privilege Escalation

Regel laut Vorlage:

```
((label=Privilege label=Assign) OR (label=Login label=Explicit
label=Credential) user=* -user IN ADMINS)
OR (label=User label=Add label=Group user=* group="*-Admin")
```

Regel nach Anpassungen:

```
((label=Privilege label=Assign) OR (label=Login label=Explicit
label=Credential -"process" IN ["*winlogon.exe", "*lsass.exe",
"*svchost.exe", "*task*.exe", "*mmc.exe", "*services.exe",
"*consent.exe"]) domain=TESTIT user="n*" -user=*$ -user IN ADMINS)
OR (label=User label=Add label=Group user=* group="*Admins*")
```

- **Funktionalität**

  Anpassungen: Da False Positives möglichst reduziert werden sollen, wird zusätzlich die Domäne eingeschränkt sowie Computer Accounts ausgenommen.

  Test: Da in der Regel drei Fälle abgedeckt werden müssen alle drei hinsichtlich ihrer Funktionalität überprüft werden.

  Der erste Teil der Regel kann geprüft werden, indem einem/r normalen User/in unter den "User Right Assignments" eine vertrauliche Berechtigung gegeben wird z.B. "Back up files and directories". Das ist wichtig, da bei nicht vertraulichen Berechtigungen kein Event generiert wird.

  Der zweite Teil, das Login mit expliziten Anmeldeinformationen, kann beispielsweise getestet werden, indem einem Service ein anderer Account hinterlegt wird. So würde das hier mit dem Prozess "services.exe" aufscheinen.

  Der letzte Teil kann überprüft werden, indem ein/e Benutzer/in einer Admin-Gruppe hinzugefügt wird. Dabei muss beachtet werden, dass die Namenskonvention für Admin-Gruppen entsprechend angepasst werden müssen.

- **Effektivität**

  Zielerfüllung: Kommt eine der geprüften Aktivitäten im normalen Benutzerkontext vor, so kann das ein Anzeichen dafür sein, dass ein/e Angreifer/in versucht so an höhere Rechte zu gelangen. Somit unterstützt diese Regel das Ziel des Use Case Account Manipulation.

  Redundanz: Es gibt Regel, die Teile dieser Regel abdecken, wie beispielsweise Regel 32. Allerdings wird nicht die gesamte Regel durch andere abgedeckt.

  False Positives: Nach einer Woche Laufzeit hat diese Regel eine sehr hohe Anzahl an Alerts generiert. Vor allem die Event-ID 4648, der Anmeldeversuch mit expliziten Anmeldeinformationen, war der Grund dafür. Darum wurden, die häufigsten Prozesse, für die dieses Event generiert wird ausgenommen und durch `user="n*"` Service Accounts ausgenommen. Dabei wird auf sich auf eine interne Namenskonvention für Standardbenutzer/innen bezogen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 26: Windows Account Creation followed by Group Add

Regel laut Vorlage:

```
[norm_id=WinServer* label=User label=Account label=Management
label=Create -target_user=*$ -caller_user=*$ -user IN EXCLUDED_USERS
| rename target_user as account, caller_user as user] as s1 followed by
[norm_id=WinServer* label=Security label=Group label=Member label=Add
| rename target_user as member, group as group_name] as s2
on s1.account=s2.member
| rename s1.account as Member, s1.user as CreatedUser,
s1.log_ts as UserCreatedTime_ts, s2.log_ts as UserAddedTime_ts,
s2.user as AddedUser, s2.group_name as Group
```

Regel nach Anpassungen:

```
[norm_id=WinServer* label=User label=Account label=Management
label=Create -target_user=*$ -caller_user=*$ -user IN EXCLUDED_USERS_RULE26
| rename target_user as account, caller_user as user] as s1 followed by
[norm_id=WinServer* label=Security label=Group label=Member label=Add
| rename target_user as member, group as group_name] as s2
```

```
on s1.display_name=s2.member
| rename s1.account as Member, s1.user as CreatedUser,
s1.log_ts as UserCreatedTime_ts, s2.log_ts as UserAddedTime_ts,
s2.user as AddedUser, s2.group_name as Group
```

- **Funktionalität**

  Anpassungen: Beim Testen der Regel ist aufgefallen, dass sie auf diese Weise nicht funktioniert. Im Log für das Hinzufügen zu einer Gruppe wird der "Display Name" des/der User/in verwendet und nicht der "User Logon Name". Aus diesem Grund muss `account` durch `display_name` ersetzt werden.

  Test: Getestet wurde die Regel durch das Erstellen eines Accounts, welcher gleich darauf als Mitglied einer Gruppe hinzugefügt wurde.

- **Effektivität**

  Zielerfüllung: Das Erstellen eines/r Benutzer/in und das anschließende sofortige Hinzufügen zu einer Gruppe kann durch eine/n Angreifer/in durchgeführt werden, der versucht an höhere Rechte zu kommen. Somit unterstützt diese Regel die Ziele der Use Cases Account Manipulation und Exploitation for Privilege Escalation.

  Redundanz: Es gibt ähnliche Regeln die das Hinzufügen zu einer Administrator-Gruppe überprüfen. Allerdings gibt es keine andere Regel, die dies in Abhängigkeit der Kontoerstellung setzt.

  False Positives: Die Regel hat nach einer Woche Laufzeit eine mittlere bis hohe Anzahl an False Positives generiert. Das liegt daran, dass das Erstellen eines/r Benutzer/in und das anschließende Hinzufügen zu einer Gruppe Teil des Workflows ist. Aus diesem Grund wird der/die Benutzer/in unter dem dieser Workflow durchgeführt wird durch das Hinzufügen in die Liste `EXCLUDED_USERS_RULE26` ausgenommen. Auf diese Weise kann die Anzahl der Alerts erheblich reduziert werden.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 27: Windows Logon Rights Changes

Regel laut Vorlage:

```
norm_id=WinServer* (event_id=4717 OR event_id=4718)(rights=*
OR privilege=*) -user="ANONYMOUS LOGON" -user IN EXCLUDED_USERS
| rename rights as privilege, caller_user as user
```

- **Funktionalität**

  Test: Diese Regel wird überprüft, indem einem Account so wie schon zuvor bestimmte Logon Rights zugeordnet werden. Welche darunter fallen kann unter "https://learn.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4717" nachgesehen werden. Anschließend wird das Recht wieder entzogen, um auch diese Funktion zu testen.

- **Effektivität**

  Zielerfüllung: Das Hinzufügen von bestimmten Logon Rechten zu Benutzer/innen kann dem/der Angreifer/in dazu verhelfen sich im Netzwerk fortzubewegen oder an höhere Rechte zu kommen. Somit erfüllt die Regel das Ziel des Use Case Account Manipulation.

  Redundanz: Es gibt keine weitere Regel, die diese Aktivitäten überwacht.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 28: Windows Multiple Account Password changes by User

Regel laut Vorlage:

```
norm_id=WinServer* label=User label=Password (label=Change OR
label=Reset) -target_user=*$ -caller_user=*$ -user IN EXCLUDED_USERS_RULE28
| rename user as caller_user
| chart distinct_count(target_user) as Account by caller_user
| search Account > 1
```

- **Funktionalität**

  Test: Die Regel kann getestet werden, indem durch den/die Administrator/in für mindestens zwei Benutzer/innen die Passwörter zurückgesetzt werden.

- **Effektivität**

  Zielerfüllung: Es ist äußerst unüblich, dass ein/e Standardbenutzer/in kurz hintereinander an zwei verschiedenen Accounts das Passwort ändert. Somit deutet das auf eine mögliche Manipulation dieser Accounts hin. Die Regel trägt zum Ziel des Use Case Account Manipulation bei.

  Redundanz: Es gibt ähnliche Regeln, die häufige Passwortwechsel überwachen. Allerdings auditiert keine dieser Regeln, wie viele Passwortänderungen von einem/r Benutzer/in an anderen Accounts veranlasst werden.

False Positives: Nach einer Woche Laufzeit hat diese Regel eine geringe Anzahl an Alerts produziert. Diese können noch weiter verringert werden, da, wie schon in der Regel zuvor, die Benutzer/innen, die für den Workflow vorhanden sind, ausgenommen werden können. Des weiteren fällt auf, dass die übrigen Benutzer/innen zum Großteil Administrator-Benutzer/innen sind, diese können mittels `-user IN ADMINS` ausgenommen werden oder weiterhin überwacht werden, um hier eine mögliche ungewöhnlich hohe Anzahl an Passwortänderungen bei Accounts festzustellen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 29: Windows Multiple Password Changed by User

Regel laut Vorlage:

```
norm_id=WinServer* label=User label=Password label=Change
-target_user=*$ -caller_user=*$ -user IN EXCLUDED_USERS
| rename caller_user as user
| process compare (target_user, user) as match
| search match=True
| chart count() as Event by target_user
| search Events>1
```

- **Funktionalität**

  Test: Diese Regel wird getestet indem hintereinander mindestens zwei Mal das Passwort des/der Benutzer/in selbst geändert wird. Dabei ist zu beachten, dass die aktuelle Regel auch fehlerhafte Versuche das Passwort zu verändern mitloggt. Das falsche Eingeben eines Passworts bei der Änderung kann schnell zu False Positives führen. Aus diesem Grund könnte man die Regel durch `-event_type="*FAILURE"` erweitern.

  Dies wurde allerdings nicht umgesetzt, da ein Mindestalter des Passworts in vielen Fällen ein häufiges Ändern sowieso nicht möglich macht, aber ein ständiger Versuch möglicherweise verdächtig sein kann.

- **Effektivität**

  Zielerfüllung: Da ein/e Benutzer/in im Normalfall das Passwort nur einmal wechselt, stellt das mehrmalige Wechseln ein verdächtiges Verhalten dar. Diese Regel unterstützt die Ziele der Use Cases Account Manipulation und Exploitation for Privilege Escalation.

  Redundanz: Es gibt ähnliche Regeln. Allerdings überprüft keine dieser Regel, ob ein/e Benutzer/in

das eigene Passwort innerhalb kurzer Zeit öfter ändern möchte.

False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 30: Windows User Password Never Expires

Regel laut Vorlage:

```
norm_id=WinServer* label=Change label=Management label=Account
label=User user_account_control="*Don't Expire Password - Enabled"
-target_user=*$ -user IN EXCLUDED_USERS
| rename caller_user as user, caller_domain as domain
```

- **Funktionalität**

  Test: Diese Regel wurde getestet, indem sowohl bei der Erstellung eines/r neuen Benutzer/in der Haken bei "Password never expires" gesetzt wurde als auch bei einem/r bereits bestehenden Benutzer/in.

- **Effektivität**

  Zielerfüllung: Da Accounts, deren Passwörter nie geändert werden müssen eine Schwachstelle darstellen, ist es wichtig, diese aufzudecken, um Maßnahmen ergreifen zu können. Zudem kann ein/e Angreifer/in der bereits im System ist, auf diese Weise dafür sorgen, dass er weiterhin Zugriff hat. Somit unterstützt die Regel das Ziel des Use Case Account Manipulation.

  Redundanz: Es gibt keine weitere Regel, die diese Aktivität überprüft.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit eine geringe Anzahl an Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 31: Windows User Account Change to End with Dollar Sign

Regel laut Vorlage:

```
norm_id=WinServer* label=User label=Account label=Change label=Name
```

```
new_user=*$ -user IN EXCLUDED_USERS
| rename caller_user as user, caller_domain as domain
```

Regel nach Anpassungen:

```
norm_id=WinServer* label=User label=Account label=Change label=Name
-target_user=*$ new_user=*$ -user IN EXCLUDED_USERS
| rename caller_user as user, caller_domain as domain
```

- **Funktionalität**

  Anpassungen: Die Regel laut Vorlage prüft lediglich, ob der neue Name mit einem Dollar-Zeichen endet. Dabei fallen auch jene Events hinein, wo der Name zwar geändert wurde, aber bereits davor ein Dollar-Zeichen am Ende hatte. Aus diesem Grund müssen mit `-target_user=*$` jene Accounts ausgenommen werden, die von Beginn an so benannt wurden.

  Test: Damit diese Regel anschlägt, muss der "User Logon Name (Pre-2000)" geändert werden. Nur dann sind die entsprechenden Werte vorhanden, um die Änderung identifizieren zu können.

- **Effektivität**

  Zielerfüllung: Ein/e Angreifer/in kann einen Account auf diese Weise manipulieren, um die Wahrscheinlichkeit erkannt zu werden, zu senken. Somit unterstützt diese Regel das Ziel des Use Case Account Manipulation.

  Redundanz: Es gibt keine weitere Regel, die diese Aktivität prüft.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 32: Windows User Added to Administrator Group

Regel laut Vorlage:

```
norm_id=WinServer (group="*-Admins" OR group=Administrators) label=Add
label=Member label=Management label=Group label=Security
-user IN EXCLUDED_USERS
```

- **Funktionalität**

  Test: Überprüft kann sie werden, indem ein/e Benutzer/in einer Administratoren-Gruppe zugeordnet wird.

- **Effektivität**

  Zielerfüllung: Das Hinzufügen eines/r Benutzer/in zu einer Administratoren-Gruppe kann dem/der Angreifer/in dazu verhelfen, höhere Rechte zu bekommen und im System zu verbleiben. Somit unterstützt diese Regel das Ziel des Use Case Account Manipulation.

  Redundanz: Diese Regel wird bis auf die Überprüfung der Gruppe Administrators schon mit der Regel 25 "Possible Account Misuse-Privilege Escalation" abgedeckt.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit nur eine geringe Anzahl an Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Da die Regel nur zwei der drei Kriterien erfüllt, wird sie deaktiviert. Allerdings soll diese Regel durch Regel 25 abgedeckt werden.

Regel 33: Windows User Added to Domain Enterprise Admin

Regel laut Vorlage:

```
norm_id=WinServer* action="added" (group_name="Enterprise Admins"
OR group_name="Domain Admins" OR group="Enterprise Admins" OR
group="Domain Admins") (member=* or target_user=*) -user IN EXCLUDED_USERS
| rename target_user as member, group_name as group
```

- **Funktionalität**

  Test: Diese Regel wird überprüft, indem sowohl ein/e Benutzer/in den Enterprise Admins als auch den Domain Admins hinzugefügt wird.

- **Effektivität**

  Zielerfüllung: Das Hinzufügen von Benutzer/innen zur Domain Admin or Enterprise Admin Gruppe sollte nur sehr selten vorkommen und nur von berechtigen Benutzer/innen durchgeführt werden. Diese Aktivität sollte unbedingt überwacht werden, da sie von Angreifer/innen dazu genutzt werden kann hoch privilegierte Rechte zu bekommen. Somit unterstützt diese Regel den Use Case Account Manipulation.

  Redundanz: Auch, wenn das Hinzufügen von Benutzer/innen zu Administrator-Gruppen bereits durch Regel 25 überwacht wird, ist es essentiell einen eigenen hochprioren Alert für diese Events zu generieren.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann

für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 34: Windows User Removed from Administrator Group

Regel laut Vorlage:

```
norm_id=WinServer (group="*Admins*" OR group=Administrators)
label=Remove label=Member label=Management label=Group label=Security
-user IN EXCLUDED_USERS
```

- **Funktionalität**

  Test: Diese Regel wird überprüft, indem der/die zuvor hinzugefügte Benutzer/in wieder entfernt wird.

- **Effektivität**

  Zielerfüllung: Das Entfernen von Benutzer/innen aus Administrator-Gruppen kann von Angreifer/innen dazu genutzt werden den Administrator/innen ihre Rechte zu entziehen. Somit unterstützt diese Regel das Ziel des Use Case Account Manipulation.

  Redundanz: Es gibt keine weitere Regel, die dieses Verhalten abdeckt.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit nur eine geringe Anzahl an Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 35: Windows User Removed from Domain Enterprise Admin

Regel laut Vorlage:

```
norm_id=WinServer* action="removed" (group_name="Enterprise Admins"
OR group_name="Domain Admins" OR group="Enterprise Admins" OR
group="Domain Admins") (member=* OR target_user) -user IN EXCLUDED_USERS
| rename target_user as member, group_name as group
```

- **Funktionalität**

  Diese Regel wird überprüft, indem sowohl der/die hinzugefügte Benutzer/in der Enterprise Admins als auch der Domain Admins entfernt wird.

- **Effektivität**

  Zielerfüllung: Das Entfernen von Benutzer/innen aus Administrator-Gruppen kann von Angreifer/innen dazu genutzt werden den Administrator/innen ihre Rechte zu entziehen. Somit unterstützt diese Regel das Ziel des Use Case Account Manipulation.

  Redundanz: Auch, wenn das Entfernen von Benutzer/innen aus Administrator-Gruppen bereits durch Regel 34 überwacht wird, ist es essentiell einen eigenen hochprioren Alert für diese Events zu generieren.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.
- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 36: Default Account privilege elevation followed by restoration of previous account state

Regel laut Vorlage:

```
[label=User label=Group label=Management label=Add
| rename target_user as account] as s1
followed by
[label=User label=Group (label=Remove or label=Delete) -target_user=*$
| rename target_user as account] as s2
on s1.account=s2.account
| rename s1.log_ts as ElevationTime_ts, s2.log_ts as RestorationTime_ts,
s1.user as UserElevation, s2.user as UserRestoration,
s1.account as Account, s1.message as PrivilegeElevation,
s2.message as PrivilegeRestoration
```

Regel nach Anpassungen:

```
[label=User label=Group label=Management label=Add
-user IN EXCLUDED_USERS_RULE36
| rename target_user as account] as s1
followed by
[label=User label=Group (label=Remove or label=Delete) -target_user=*$
```

```
| rename target_user as account] as s2
on s1.account=s2.account AND s1.group=s2.group
| rename s1.log_ts as ElevationTime_ts, s2.log_ts as RestorationTime_ts,
s1.user as UserElevation, s2.user as UserRestoration,
s1.account as Account, s1.message as PrivilegeElevation,
s2.message as PrivilegeRestoration
```

- **Funktionalität**

  Test: Geprüft wird die Regel indem ein/e Benutzer/in einer Gruppe hinzugefügt und wieder entfernt oder die Gruppe gelöscht wird.

- **Effektivität**

  Zielerfüllung: Bei dieser Regel geht es darum, zu erkennen, wenn ein/e Angreifer/in kurzfristig deren Rechte erhöht, um sie dann wieder zurückzubauen, damit die Veränderung nicht bemerkt wird. Somit unterstützt die Regel die Ziele der Use Cases Account Manipulation und Exploitation for Privilege Escalation.

  Redundanz: Es sind ähnliche Regel vorhanden. Allerdings achtet keine andere der Regeln darauf, ob ein/e Benutzer/in nach dem Hinzufügen zu einer Gruppe schnell wieder entfernt wird.

  False Positives: Nach einer Woche Laufzeit hat die Regel eine sehr große Anzahl an Alerts produziert. Der Grund dafür waren hauptsächlich Gruppenänderungen. Das heißt ein/e Benutzer/in wurde aus einer Gruppe entfernt und einer anderen hinzugefügt. Um das auszunehmen, muss als join-Parameter zusätzlich `AND s1.group=s2.group` hinzugefügt werden. Anschließend konnte die Anzahl der Alerts reduziert werden. Um weitere False Positives zu vermeiden kann der/die Workflow-Benutzer/in wieder ausgenommen werden. Nach den Änderungen war nur mehr eine geringe Anzahl an Alerts zu verzeichnen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 37: Default Configuration Change on Security Device

Regel laut Vorlage:

```
-user IN ADMINS (label=Policy OR label=Configuration) AND (label=Change OR
label=Remove OR label=Reset OR label=Modify OR label=Add
OR label=Delete OR label=Set) device_name IN SECURITY_DEVICES
```

Regel nach Anpassungen:

```
-user=*$ -domain="NT AUTHORITY" -user IN ADMINS (label=Policy OR label=Configuratio
label=Remove OR label=Reset OR label=Modify OR label=Add
OR label=Delete OR label=Set) device_name IN SECURITY_DEVICES
```

- **Funktionalität**

  Anpassungen: Damit diese Regel funktioniert, muss zuerst darauf geachtet werden, dass in der Liste `SECURITY_DEVICES` jene Gerätenamen eingefügt wurden, an denen Richtlinienänderungen vorgenommen können bzw. die überwacht werden sollen.

  Um False Positives zu vermeiden wurden Computer Accounts sowie der Domäne "NT Authority" ausgenommen.

  Test: Damit diese Regel getestet werden kann, müssen einem/r normalen Benutzer/in zuvor höhere Rechte gegeben werden. Zu Testzwecken kann das durch das Hinzufügen in die Administratoren-Gruppe erreicht werden. Dann können wie schon zuvor beispielsweise "User Right Assignment Policies" verändert werden.

- **Effektivität**

  Zielerfüllung: Die Regel soll darauf achten, dass keine Richtlinienänderungen von unberechtigten Benutzer/innen durchgeführt werden. Da eine so potentielle Manipulation von Accounts durchgeführt werden könnte, trägt die Regel zum Ziel dieses Use Case bei.

  Redundanz: Es gibt keine weitere Regel, die auf diese Aktivitäten überprüft.

  False Positives: Nach einer Woche Laufzeit hat die Regel eine niedrige bis mittlere Anzahl an Alerts generiert. Allerdings muss hier beachtet werden, dass bei einem Wartungsfenster mit einem höheren Aufkommen an Alerts gerechnet werden muss.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 38: Account Manipulated for Persistence Detected

Regel laut Vorlage:

```
((norm_id=WinServer label="Process" label=Create) OR
(norm_id=WindowsSysmon image="*net.exe*)) command="*net*localgroup*/add"
-user in EXCLUDED_USERS
```

Regel nach Anpassungen:

```
((norm_id=WinServer label="Process" label=Create) command="*net*localgroup*/add"
```

```
-user in EXCLUDED_USERS
```

- **Funktionalität**

  Anpassungen: Da in den aktuellen Use Cases keine Sysmon-Logs verwendet werden und diese auch nicht vorhanden sind, wird der Teil der Regel entfernt, um Ressourcen zu sparen.

  Beim Testen ist aufgefallen, dass hier nochmal eine Veränderung der Audit Policies sowohl auf dem DC als auch auf dem Client vorgenommen werden muss. Über die Group Policy gehört ebenfalls die Einstellung "Include command line in process creation events" aktiviert. Diese ist unter "Computer Configuration/Administrative Templates/System/Audit Process Creation" zu finden. Die Änderung ist auch für einige der folgenden Regeln wichtig!

  Test: Wurde das Auditing aktiviert kann die Regel mit folgenden zwei Commands getestet werden:

  ```
  <net localgroup Test /add>
  <net localgroup Test account5 /add>
  ```

- **Effektivität**

  Zielerfüllung: Die Regel überprüft, das Erstellen einer lokalen Gruppe oder das Hinzufügen zu einer lokalen Gruppe. Da es sich dabei um Account Manipulation handeln könnte, unterstützt diese Regel den erwähnten Use Case.

  Redundanz: Diese Regel überprüft auf die gleichen Aktivitäten, wie Regel 41. Außerdem überprüft Regel 40 ebenfalls Aktivitäten, wo diese Events hineinfallen würden.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Da die Regel nur zwei der drei Kriterien erfüllt, wird sie deaktiviert. Allerdings soll diese Regel durch Regel 40 abgedeckt werden.

Regel 39: Password Change on DSRM Account Detected

Regel laut Vorlage:

```
norm_id=WinServer event_id=4794 -user IN EXCLUDED_USERS
```

- **Funktionalität**

  Test: Um die Regel zu testen, muss das Passwort für den DSRM Account zurückgesetzt werden. Das kann mit folgenden Commands erreicht werden:

  ```
  ntdsutil
  ```

```
set dsrm password
reset password on server null
```

- **Effektivität**

  Zielerfüllung: Da das Passwort des DSRM Accounts nur sehr selten zurückgesetzt wird, sollten diese Aktivitäten überwacht werden. Es kann sich dabei auch um die Technik der Account Manipulation handeln, deren Ziel diese Regel unterstützt.

  Redundanz: Es gibt keine weitere Regel, die diese Aktivität überwacht.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 40: Persistence Using Account Creation Detected

Regel laut Vorlage:

```
norm_id=WinServer label="Process" label=Create commandline="*net*/add*/y"
-user IN EXCLUDED_USERS
| process eval("attack_class='Persistence'")
| process eval("technique='Create Account'")
| rename commandline as command
```

Regel nach Anpassungen:

```
norm_id=WinServer label="Process" label=Create commandline="*net*/add*"
-user=*$ -user=SYSTEM -user IN EXCLUDED_USERS
| process eval("attack_class='Persistence'")
| process eval("technique='Create Account'")
| rename commandline as command
```

- **Funktionalität**

  Anpassungen: Der Parameter `/y` in der Commandline wurde durch eine Wildcard ersetzt, da dieser nicht zwingend verwendet werden muss. Zudem wurden Computer Accounts sowie der SYSTEM-Benutzer ausgenommen.

  Test: Getestet werden kann diese Regel mit jedem `net` Befehl, wo `/add` verwendet wird, wie `net user Testuser Testpassword123! /add`.

- **Effektivität**

  Zielerfüllung: Die Regel überprüft, das Erstellen von Benutzer/innen oder Gruppen bzw. das Hinzufügen zu einer Gruppe. Da es sich dabei um Account Manipulation handeln könnte, unterstützt diese Regel den erwähnten Use Case.

  Redundanz: Diese Regel deckt einen etwas größeren Bereich ab, als Regeln 38 und 41.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit eine geringe Anzahl an False Positives produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 41: Persistence Using Account Manipulation Detected

Regel laut Vorlage:

```
norm_id=WinServer label="Process" label=Create
commandline="*net*localgroup*/add" -user IN EXCLUDED_USERS
| process eval("attack_class='Persistence'")
| process eval("technique='Create Account'")
| rename commandline as command
```

- **Funktionalität**

  Test: Die Regel kann mit folgenden zwei Commands getestet werden:

  `<net localgroup Test /add>`

  `<net localgroup Test account5 /add>`

- **Effektivität**

  Zielerfüllung: Die Regel überprüft das Erstellen einer lokalen Gruppe oder das Hinzufügen zu einer lokalen Gruppe. Da es sich dabei um Account Manipulation handeln könnte, unterstützt diese Regel den erwähnten Use Case.

  Redundanz: Diese Regel überprüft auf die gleichen Aktivitäten, wie Regel 38. Außerdem überprüft Regel 40 ebenfalls Aktivitäten, wo diese Events hineinfallen würden.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Da die Regel nur zwei der drei Kriterien erfüllt, wird sie deaktiviert. Allerdings soll diese Regel durch

Regel 40 abgedeckt werden.

Regel 42: Windows User Account Lockout

Regel laut Vorlage:

```
norm_id=WinServer* label=User label=Account label=Management
label=Lock -target_user=*$ -user=*$ -user IN EXCLUDED_USERS
| rename caller_computer as workstation
```

- **Funktionalität**

  Test: Um die Funktionalität zu testen, wird zuvor die "Account Lockout Policy" im "Group Policy Management" konfiguriert. Danach wird sich so oft mit einem/r neu angelegten Benutzer/in eingeloggt, dass diese Policy ausgelöst und der/die Benutzer/in gesperrt wird.

- **Effektivität**

  Zielerfüllung: Die Regel prüft das einmalige Sperren eines Benutzerkontos. Obwohl dies ein Zeichen für einen Brute Force Angriff sein kann, ist das einmalige möglicherweise zu generisch und as Ziel dieses Use Cases wird durch andere Regeln besser unterstützt.

  Redundanz: Regel 5 und Regel 45 überprüfen ebenfalls auf Accounts, die gesperrt werden. Allerdings prüft keine der beiden Regeln, das einfache Sperren eines Accounts. Regel 5 prüft es in Abhängigkeit zur zuvor erfolgten Anmeldeversuchen und Regel 45 prüft, ob es in einem bestimmten Zeitraum öfter als einmal vorkommt.

  False Positives: Nach einer Woche Laufzeit hat diese Regel eine hohe Anzahl an Alerts produziert. Allerdings war zu erkennen, dass es sich oft um die selben Benutzer/innen handelt, somit wäre Alert Throtteling ein möglicher Lösungsweg.

- **Fazit**

  Da die Regel nicht alle Kriterien voll erfüllt, wird sie deaktiviert. Allerdings soll diese Regel durch Regel 5 und Regel 45 abgedeckt werden.

Regel 43: Windows Successful Brute Force Attack from Same User

Regel laut Vorlage:

```
[10 norm_id=WinServer* label=User label=Login label=Fail -user
IN EXCLUDED_USERS
| rename target_user as user having same user] as s1
```

```
followed by
[norm_id=WinServer* label=User label=Login label=Successful
| rename target_user as user ] as s2 within 5 minute
on s1.user=s2.user
| rename s2.user as User
```

- **Funktionalität**

  Test: Hier muss beim Testen wieder auf die aktuell eingestellte "Lockout Policy" geachtet werden oder die Anzahl der fehlerhaften Versuche wird zum Testen in der Regel heruntergesetzt.

  Nachdem sich 10 Mal versucht wurde mit dem falschen Passwort anzumelden, wurde beim 11. Mal das richtige eingegeben.

- **Effektivität**

  Zielerfüllung: Die Regel überprüft das für einen Brute Force Angriff typische Verhalten. Somit unterstützt diese Regel den erwähnten Use Case.

  Redundanz: Die Regel entspricht bis auf die Threshold den Regeln 12 und 19.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit wenige False Positives produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Da die Regel nur zwei der drei Kriterien erfüllt, wird sie deaktiviert. Allerdings soll diese Regel durch Regel 19 abgedeckt werden.

Regel 44: Windows Successful Brute Force Attack from Same Source

Regel laut Vorlage:

```
[10 norm_id=WinServer* label=User label=Login label=Fail
-user IN EXCLUDED_USERS
| rename target_user as user having same source_address] as s1
followed by
[norm_id=WinServer* label=User label=Login label=Successful
| rename target_user as user ] as s2 within 10 minute
on s1.source_address=s2.source_address
| rename s2.user as User
```

Regel nach Anpassungen:

```
[10 norm_id=WinServer* label=User label=Login label=Fail -user IN
EXCLUDED_USERS
| rename target_user as user having same device_ip] as s1
followed by
[norm_id=WinServer* label=User label=Login label=Successful
| rename target_user as user ] as s2 within 10 minute
on s1.device_ip=s2.device_ip
| rename s2.device_ip as DeviceIp
```

- **Funktionalität**

  Anpassungen: Da hier, wie schon bei Regel 10, wieder das Problem mit den Source-Adressen besteht, muss diese wieder durch das Feld `device_ip` ersetzt werden.

  Test: Um diese Regel zu testen, werden mindestens 10 fehlgeschlagene Anmeldeversuche produziert und sich anschließend erfolgreich angemeldet. Dabei können unterschiedliche Benutzer/innen gewählt werden, die Versuche müssen allerdings am selben Gerät ausgeführt werden.

- **Effektivität**

  Zielerfüllung: Diese Regel überprüft, ob ein bestimmtes Gerät möglicherweise dazu verwendet wird, um Kennwörter zu erraten. Somit unterstützt diese Regel den bereits erwähnten Use Case.

  Redundanz: Es gibt einige ähnliche Regeln, allerdings prüft keine der Regeln auf ein bestimmtes Gerät, dass möglicherweise für den Angriff genutzt wird.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit keine Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 45: Windows Multiple Unique Lockouts

Regel laut Vorlage:

```
label=Lock label=Account -target_user=*$ -user=*$
| rename target_user as user
| chart distinct_count(log_ts) as cnt by user
| search cnt>1
```

- **Funktionalität**

  Um diese Regel zu testen, muss ein/e Benutzer/in durch zu viele Anmeldeversuche gesperrt werden.

Anschließend muss entweder gewartet werden oder der Account wird manuell entsperrt. Bei der Regel muss darauf geachtet werden, welcher Zeitraum gesetzt wird, auch abhängig davon welcher Zeitraum für das Sperren des Accounts gewählt wurde. Standardmäßig sind 60 Minuten für diese Regel eingestellt.

- **Effektivität**

  Zielerfüllung: Sperrt sich ein/e Benutzer/in in einer Stunde öfter als einmal aus, kann es sich um potentiell schadhaftes Verhalten handeln, das durch einen Brute Force Angriff ausgelöst werden kann. Aus diesem Grund unterstützt diese Regel den erwähnten Use Case.

  Redundanz: Es gibt keine weitere Regel, die überprüft, ob ein Account mehrmals hintereinander gesperrt wird.

  False Positives: Die Regel hat nach einer Woche Laufzeit eine mittlere bis hohe Anzahl an Alerts produziert. Das liegt daran, dass teilweise zwei Lockouts die Stunde angezeigt werden, weil es von zwei DCs gelockt wurde, es sich aber um das gleiche Lockout-Event handelt. Das kann vorgebeugt werden indem statt `count()`, `distinct_count(log_ts)` verwendet wird.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der ursprünglichen Version aktiviert.

Regel 46: Windows Multiple Failed Attempts against a Single Account

Regel laut Vorlage:

```
norm_id=WinServer* label=User label=Login label=Fail -target_user=*$
-user=*$ -user IN EXCLUDED_USERS
| rename target_user as user
| chart count() as FailedTimes by user order by count() desc
| search FailedTimes>1
```

- **Funktionalität**

  Test: Um diese Regel zu testen, muss sich zweimal falsch eingeloggt werden.

- **Effektivität**

  Zielerfüllung: Auch, wenn mehrmalige fehlerhafte Loginversuche ein Indikator für Brute Force Angriffe sind, deckt dieser Alert diesen Use Case nicht sinnvoll ab, da er zu vage ist.

  Redundanz: Diese Events werden ebenso von Regel 6 abgedeckt.

  False Positives: Der Alert hat nach einer Woche Laufzeit eine sehr hohe Anzahl an Alerts generiert. False Positives sind hier zu erwarten, da bei zweimaliger falscher Eingabe des Passworts der Alert

ausgelöst wird.

- **Fazit**

  Die Regel erfüllt keine der drei Kriterien und wird daher deaktiviert.

Regel 47: Windows Failed Interactive User Logins Detected

Regel laut Vorlage:

```
norm_id=WinServer* label=User label=Login label=Fail logon_type=2
-target_user=*$ -user=*$ -user IN EXCLUDED_USERS
| rename target_user as user, target_domain as domain
| chart count() as Event by user, source_address, domain
| search Event>10
```

Regel nach Anpassungen:

```
norm_id=WinServer* label=User label=Login label=Fail logon_type=2
-target_user=*$ -user=*$ -user IN EXCLUDED_USERS
| rename target_user as user, target_domain as domain
| chart count() as Event by user, source_address, host, domain
| search Event>10
```

- **Funktionalität**

  Anpassungen: Das Feld `source_address` wird durch das Feld `host` ergänzt, da so der Rechnername zusätzlich zur IP-Adresse angezeigt wird.

  Test: Diese Regel wird getestet indem sich mit einem Benutzerkonto öfter als 10 Mal falsch eingeloggt wird.

- **Effektivität**

  Zielerfüllung: Diese Regel überprüft, ob ein/e Benutzer/in sich mehrmals versucht mit falschen Anmeldeinformationen anzumelden. Durch die Einschränkung auf interaktive Logins kombiniert mit dem höhren Grenzwert, kann dies auf einen potentiellen Brute Force Angriff hindeuten. Somit erfüllt die Regel die Ziele dieses Use Cases.

  Redundanz: Diese Regel deckt einen ähnlichen Bereich ab, wie Regel 6 "Windows Failed User Login Attempt" und Regel 46 "Windows Multiple Failed Attempts against a Single Account". Der einzige Unterschied ist die Beschränkung auf Interactive Logins und die Threshhold. Zudem wurden die Regel 6 und 46 deaktiviert.

False Positives: Die Alert Regel hat in der ersten Woche Laufzeit nur eine geringe Anzahl an Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 48: Zerologon CVE-2020-1472 Exploitation Detected

Regel laut Vorlage:

```
norm_id=WinServer label=Computer label=Account label=Change computer=*
user="ANONYMOUS LOGON" user_id="S-1-5-7" password_last_set_ts=*
computer IN WINDOWS_DC
```

- **Funktionalität**

  Anpassungen: Es gilt zu beachten, dass die `WINDOWS_DC` Liste so modifiziert wird, dass der DC-Name endend mit einem Dollar-Zeichen darin steht.

  Test: Danach kann die Regel getestet werden, indem ein neues Computerkonto hinzugefügt wird, dessen Name in der Liste steht. Dieses Event löst die Regel bereits aus, da ein Account Passwort dafür erstellt wird. Zudem kann "Reset Account" gewählt werden, um die Regel auszulösen.

- **Effektivität**

  Zielerfüllung: Mit dieser Regel wird überprüft, ob die Zerologon Schwachstelle möglicherweise ausgenutzt wurde, indem mit dem Benutzer "Anonymous Logon" ein Computer Account erstellt wurde. Somit unterstützt die Regel das Ziel des Use Case Exploitation for Privilege Escalation.

  Redundanz: Es gibt keine weitere Regel, die diese Events abdeckt.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit nur eine geringe Anzahl an Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Regel 49: Active Directory Schema Change Detected

Regel laut Vorlage:

```
norm_id=WinServer* label=Directory label=Service label=Object
```

```
(label=Change OR label=Create OR label=Move OR label=Delete OR
label=Undelete) -user IN EXCLUDED_USERS
```

- **Funktionalität**

  Test: Dieser Alert wird ausgelöst indem zum Beispiel ein Computer Account erstellt wird oder Werte dafür geändert werden.

- **Effektivität**

  Zielerfüllung: Die Regel überprüft, ob unberechtigte Änderungen am AD Schema vorgenommen werden, die den Angreifer/innen zu höheren Rechten verhelfen könnten. Somit unterstützt die Regel das Ziel des Use Case Exploitation for Privilege Escalation.

  Redundanz: Es gibt keine weitere Regel, die diese Aktivitäten überprüft.

  False Positives: Die Alert Regel hat in der ersten Woche Laufzeit nur eine geringe Anzahl an Alerts produziert, somit kann für die Ersteinschätzung davon ausgegangen werden, dass die Einstellungen passen.

- **Fazit**

  Diese Regel erfüllt alle Kriterien und bleibt bis auf weiteres in der angepassten Version aktiviert.

Die hier abgebildeten Schritte inkludieren lediglich den ersten Schritt der Evaluierung. Zudem kann hier auch festgestellt werden, dass noch Regeln fehlen, welche ebenfalls hinzugefügt werden sollen. In diesem Fall muss im Prozess für diese Regeln ein Schritt zurück gegangen werden und die gleichen Schritte durchgeführt werden, wie mit diesen Regeln.
Sollen weitere Regeln hinzugefügt werden, dann muss im Prozess wieder zu dem Schritt "Erstellung der Alert Regeln" gegangen werden. Hier können diese Regeln dann erstellt werden und durchlaufen anschließend den weiteren Prozess, damit diese genauso evaluiert werden. Durch die Literaturrecherche hat sich ergeben, dass möglicherweise folgende Alert Regeln erstellt und auf ihre Sinnhaftigkeit zum Einsatz im SIEM evaluiert werden sollten:

- Passwortänderung des krbtgt-Accounts
- Wiederverwendung von TGTs durch andere Systeme (z.B. andere IP-Adresse)
- Änderungen eines Benutzerkontos bezüglich der Einstellung "Don't requrie Kerberos Pre-Authentication" mittel UAC
- TGS-Request von bestimmten Service Accounts
- HoneyToken-Account Aktivitäten
- Werte des AccountDomain-Feldes entsprechen nicht der Domäne (andere Werte, der Fully-Qualified-

Domain-Name (FQDN) oder leere Felder sind Anzeichen eines potentiellen Angriffs)

6.3.6 Planung nächster Schritte

Der letzte Schritt bildet ab, wie nun mit den generierten Alerts umgegangen wird. In diesem Schritt muss auch definiert werden, wie die Anbindung zu einem möglicherweise vorhandenen Security Incident Prozess zu erfolgen hat und vor allem was den Auslöser dafür darstellt.

Benachrichtigungen

Hinsichtlich dem Versenden von Benachrichtigungen für Alerts bietet LogPoint die folgenden Möglichkeiten:

- E-Mail
- SMS
- HTTP
- SSH
- SNMP
- Syslog

Dazu kommt, dass LogPoint mit der SOAR Funktionalität noch weitere Schnittstellen implementiert hat, wodurch es beispielsweise möglich ist, direkt ein Ticket in ServiceNow zu öffnen.

Bearbeitung der Alerts

Egal, ob ein Ticket geöffnet oder eine Mail versendet wurde, jeder Alert muss analysiert und bearbeitet sowie dokumentiert werden. Wie dies zu erfolgen hat, deckt im besten Fall der bereits vorhandene Security Incident Prozess ab.
LogPoint bietet integriert eine Incident-Seite, wo unabhängig vom gewählten Benachrichtigungstyp alle Alerts angezeigt werden. Von dort kann sich der/die Analyst/in direkt die Logdaten zum Alert ansehen, Kommentare dazu verfassen und den Alert anschließend abschließen. Die Analyse von Alerts kann dazu führen, dass die Erkenntnis gewonnen wird, die Alert Regel weiter abändern oder spezifizieren zu müssen. Der Schritt der Evaluierung kann nie als vollständig abgeschlossen angesehen werden.

Automatisierung

Einer jener Schritte, die aktuell am besten automatisiert oder zumindest teil-automatisiert werden können, ist der Schritt der Alert-Bearbeitung. Zum einen kann von Anfang an dafür gesorgt werden, dass im Alert

die wichtigsten Informationen stehen, um schnell analysieren zu können, zum anderen kann das System Teile dieser Analyse übernehmen.
Mit der SOAR-Komponente, welche bei LogPoint ohne Zusatzlizenz für einen/r Benutzer/in inkludiert ist, können Playbooks gebaut werden, die Alert Regel mit Informationen anreichern, Maßnahmen setzen oder sogar ganze Alerts vollständig bearbeiten können.

7 Conclusio

Nachdem der vorgeschlagene Prozess am Beispiel Active Directory und LogPoint SIEM durchgearbeitet wurde, konnte festgestellt werden, dass die Vorgehensweise zielführend ist. Wichtig dabei ist, dass es ein sehr agiler Prozess ist und vom Feedback der jeweiligen Schritte, vor allem der Evaluierung, abhängig ist. Es ist somit kein Prozess, der nach einem Durchlauf beendet ist, stattdessen müssen die Schritte "Erstellung der Alert Regeln" und "Evaluierung der Alert Regeln" regelmäßig durchlaufen werden, da sich auch die Taktiken und Techniken der Angreifer/innen laufend verändern und das Alert Rule Set aktuell gehalten werden muss. Um dabei den Überblick zu behalten ist es ratsam eine entsprechende Dokumentation zu führen und darin festzuhalten, wie Regeln getestet wurden, wie die Ergebnisse ausgesehen haben und was für Anpassungen getätigt wurden. Auch, wenn entschieden wird eine Regel abzudrehen, sollte unbedingt dokumentiert werden, warum dies der Fall war.

Hinsichtlich des LogPoint SIEM Tools kann festgestellt werden, dass es eine gute CoverageView mithilfe des MITRE ATT&CK Framework bietet und somit eine gute Übersicht für Analyst/innen zur Verfügung stellt. Auch das Aktivieren der Alert Regeln ist einfach und in wenigen Schritten durchzuführen. Von den 49 getesteten Regeln, konnten 25 Regeln verwendet werden, ohne, dass Änderungen oder Anpassungen durchgeführt werden mussten. Bei den restlichen 49 Prozent der Regeln mussten Änderungen vorgenommen werden, weil entweder Fehler in der Vorlage waren, die Vorlage zu viele False Positives erzeugt hat oder Felder anders benannt wurden. Zudem wurden von 49 Regeln nach der Evaluierung nur 39 Regeln tatsächlich aktiviert. Die Hauptgründe dafür waren, dass die Regeln entweder redundant waren und durch andere Regeln abgedeckt werden konnten oder, dass sie zu viele Alerts generiert haben. Wenn die Regeln zu viele Alerts generiert haben, war meist der Grund, dass Aktivitäten überwacht wurden, die an sich nicht als potentiell gefährlich zu sehen waren.

Auch, wenn das LogPoint SIEM einige sinnvolle Alert Vorlagen liefert, so sollte sich ein/e Analyst/in nicht auf deren Vollständigkeit verlassen. Es wurden für die betrachteten Use Cases noch mögliche Alert Regeln identifiziert, die ebenfalls einen Mehrwert bringen könnten, allerdings zuvor die Evaluierung durchlaufen müssen.

Zusammengefasst kann gesagt werden, dass sich der Prozess sehr gut eignet, um eine erste Basis an Alert Regeln zu schaffen, sich aber nicht vollständig auf built-in Regeln im SIEM verlassen sollte.

7.1 Weiterführende Arbeiten

In weiterführenden Arbeiten können beispielsweise weitere Use Cases betrachtet werden, mit dem Ziel ein Alert Rule Set für den Schutz der gesamten Active Directory Umgebung zu schaffen. Dabei sind nicht nur die Windows Security Event Logs als Logquellen miteinzubeziehen sondern weitere Quellen wie Windows Sysmon oder auch Firewall Log oder IPS Logs. Ein weiterer interessanter Aspekt ist hier auch die Cloud, vor allem Azure AD. In vielen Unternehmen werden mittlerweile die Daten des On-Premise ADs in die Cloud synchronisiert und auch eine reine Cloud-Lösung ist möglich. Aus diesem Grund können auch die Azure AD Logs einen großen Mehrwert zum Schutz des ADs bieten.
Ein weiterer Aspekt der in zukünftigen Arbeiten betrachtet werden kann, ist jener der Automatisierung. Hier wäre es interessant festzustellen, wie viele der generierten Alerts zu welchem Anteil durch ein SOAR automatisiert behandelt werden könnten. Während hier manche Alerts möglicherweise vollständig automatisiert abgearbeitet werden können, braucht es bei anderen Alerts dann doch das Know-How eines/r Analyst/in. Aber auch die Anreicherung von Informationen mit Hilfe des SOAR zur besseren und schnelleren Ersteinschätzung stellt einen Aspekt dar, der weiter analysiert werden könnte.

Abbildungsverzeichnis

Tabellenverzeichnis

Glossar

ACE	Access Control Entry
ACL	Access Control List
AD	Active Directory
AP-REP	Application Response
AP-REQ	Application Request
API	Application Programming Interface
APT	Advanced Persistent Threat
AS	Authentication Service
AS-REP	Authentication Service Response
AS-REQ	Authentication Service Request
BYOVD	Bring your own vulnerable Driver
CCPA	California Consumer Privacy Act
CN	Canonical Name
CSV	Comma-Separated Values
CVE	Common Vulnerabilities and Exposures
DC	Domain Controller
DCOM	Distributed Component Object Model
DN	Distinguished Name
DOS	Denial of Service
DSGVO	Datenschutzgrundverordnung

EAL	Evaluation Assurance Level
EDR	Endpoint Detection and Response
EVT	Windows Event Log
EVTX	Windows XML Event Log
FN	False Negative
FP	False Positive
FQDN	Fully Qualified Domain Name
GUID	Globally Unique Identifier
ICS	Industrial Control System
IOC	Indicator of Compromise
IPS	Intrusion Prevention System
JSON	JavaScript Object Notation
KDC	Key Distribution Center
LSASS	Local Security Authority Server Service
MMC	Microsoft Management Console
MTTD	Mean Time to Detect
MTTD	Mean Time to Respond
NTLM	NT LAN Manager
OPTH	Overpass the Hash
OU	Organizational Unit

PAC	Privileged Attribute Certificate
PS	PowerShell
PTH	Pass the Hash
PTT	Pass the Ticket
RDN	Relative Distinguished Name
RDP	Remote Desktop Protocol
RODC	Read-Only Domain Controller
SaaS	Software-as-a-Service
SEM	Security Event Management
SIEM	Security Information and Event Management
SIM	Security Information Management
SOAR	Security Orchestration, Automation and Response
SPN	Service Principal Name
SSO	Single-Sign-On
TGS	Ticket Granting Service
TGS-REP	Ticket Granting Service Response
TGS-REQ	Ticket Granting Service Request
TN	True Negative
TP	True Positive
TTP	Tactics, Techniques and Procedures
UAC	User Account Control
UEBA	User and Entity Behavior Analytics
UPN	User Principal Name
VM	Virtuelle Maschine

WBEM	Web Based Enterprise Management
WDCG	Windows Defender Credential Guard
WMI	Windows Management Instrumentation
XML	Extensible Markup Language
YAML	YAML Ain’t Markup Language

Literaturverzeichnis

[1] Igor Anastasov and Danco Davcev, "SIEM implementation for global and distributed environments," en, in *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, Hammamet, Tunisia: IEEE, Jan. 2014, pp. 1–6. DOI: 10.1109/WCCAIS.2014.6916651. [Online]. Available: http://ieeexplore.ieee.org/document/6916651/ (visited on 12/20/2022).

[2] Kelly Kavanagh, Toby Bussa, and John Collins, "Magic Quadrant for Security Information and Event Management," en, Jun. 2021. [Online]. Available: https://www.gartner.com/en/documents/4003080 (visited on 10/30/2022).

[3] *Security Information and Event Management (SIEM) Market 2023-28*, en. [Online]. Available: https://www.imarcgroup.com/security-information-event-management-market (visited on 01/11/2023).

[4] Gustavo González-Granadillo, Susana González-Zarzosa, and Rodrigo Diaz, "Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures," en, *Sensors*, vol. 21, no. 14, Jul. 2021, ISSN: 1424-8220. DOI: 10.3390/s21144759. [Online]. Available: https://www.mdpi.com/1424-8220/21/14/4759 (visited on 09/09/2022).

[5] Allen Harper, Stephen VanDyke, Chris Blask, Shon Harris, and David Miller, *Security Information and Event Management (SIEM) Implementation*. New York; US: McGraw-Hill Osborne Media, 2010, ISBN: 0071701095. DOI: 10.1036/9780071701082. eprint: https://mhebooklibrary.com/doi/pdf/10.1036/9780071701082. [Online]. Available: https://mhebooklibrary.com/doi/book/10.1036/9780071701082.

[6] Gustavo Gonzalez Granadillo, Mohammed El-Barbori, and Herve Debar, "New Types of Alert Correlation for Security Information and Event Management Systems," en, in *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Larnaca, Cyprus: IEEE, Nov. 2016,

pp. 1–7, ISBN: 978-1-5090-2914-3. DOI: 10.1109/NTMS.2016.7792462. [Online]. Available: http://ieeexplore.ieee.org/document/7792462/ (visited on 09/10/2022).

[7] Blake D. Bryant and Hossein Saiedian, "Improving SIEM alert metadata aggregation with a novel kill-chain based classification model," en, *Computers & Security*, vol. 94, Jul. 2020, ISSN: 01674048. DOI: 10.1016/j.cose.2020.101817. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S016740482030095X (visited on 09/10/2022).

[8] Hassan Mokalled, Rosario Catelli, Valentina Casola, Daniele Debertol, Ermete Meda, and Rodolfo Zunino, "The Applicability of a SIEM Solution: Requirements and Evaluation," en, in *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, Napoli, Italy: IEEE, Jun. 2019, pp. 132–137, ISBN: 978-1-72810-676-2. DOI: 10.1109/WETICE.2019.00036. [Online]. Available: https://ieeexplore.ieee.org/document/8795405/ (visited on 09/09/2022).

[9] Gorka Sadowski, Kelly Kavanagh, and Toby Bussa, "Technology Insight for the Modern SIEM," en, Oct. 2018. [Online]. Available: https://www.gartner.com/en/documents/3892089 (visited on 10/30/2022).

[10] Sievierinov Oleksandr and Ovcharenko Margaret, "Analysis of correlation rules in Security information and event management systems," en, *COMPUTER AND INFORMATION SYSTEMS AND TECHNOLOGIES*, pp. 24–25, Apr. 2020.

[11] Yulia De Bari, "Automation for Cybersecurity," en, *Infosys*, 2019. [Online]. Available: https://www.infosys.com/%20about/knowledge-institute/insights/documents/future-tomorrow.pdf (visited on 09/10/2022).

[12] Aaron Pritz, *Security analytics for dummies*, en. Hoboken; NJ: John Wiley & Sons, Inc., 2018, ISBN: 978-1-119-54513-2. [Online]. Available: https://www.slideshare.net/MarusyaMaruzhenko/security-analytics-for-dummies-securonix-special-edition?from_action=save (visited on 09/10/2022).

[13] Marcello Cinque, Domenico Cotroneo, and Antonio Pecchia, "Challenges and Directions in Security Information and Event Management (SIEM)," en, in *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Memphis, TN: IEEE, Oct. 2018, pp. 95–99, ISBN: 978-1-5386-9443-5. DOI: 10.1109/ISSREW.2018.00-24. [Online]. Available: https://ieeexplore.ieee.org/document/8539170/ (visited on 09/09/2022).

[14] Gustavo Gonzalez Granadillo, Yosra Ben Mustapha, Nabil Hachem, and Herve Debar, "An ontology-driven approach to model SIEM information and operations using the SWRL formalism," en, *International Journal of Electronic Security and Digital Forensics*, vol. 4, no. 2/3, 2012, ISSN: 1751-911X, 1751-9128. DOI: 10.1504/IJESDF.2012.048412. [Online]. Available: http://www.inderscience.com/link.php?id=48412 (visited on 09/10/2022).

[15] Augusto Barros, *SIEM Correlation is Overrated*, en, Mar. 2017. [Online]. Available: https://blogs.gartner.com/augusto-barros/2017/03/31/siem-correlation-is-overrated/ (visited on 09/10/2022).

[16] Dan Dobre, Joao Girao, and Ghassan Karame, "Towards Trustworthy Cloud Storage," en, *NEC Technical Journal*, vol. 8, no. 1, Sep. 2013.

[17] Barbara Filkins and Chris Crowley, "An Evaluator's Guide to NextGen SIEM," en, *SANS Analyst Program*, Dec. 2018. [Online]. Available: https://gallery.logrhythm.com/%20independent-white-papers/sans-an-evaluators-guide-to-next-gen-siem-independent-white-paper-2018.pdf.

[18] *MITRE ATT&CK®*, en. [Online]. Available: https://attack.mitre.org/ (visited on 12/17/2022).

[19] "Best Practices for MITRE ATT&CK® Mapping," en, *CISA, FBI*, pp. 1–20, Jun. 2021. [Online]. Available: https://www.cisa.gov/uscert/sites/default/files/publications/Best%20Practices%20for%20MITRE%20ATTCK%20Mapping.pdf (visited on 12/17/2022).

[20] *Reconnaissance, Tactic TA0043 - Enterprise | MITRE ATT&CK®*, Oct. 2020. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0043/ (visited on 12/17/2022).

[21] *Resource Development, Tactic TA0042 - Enterprise | MITRE ATT&CK®*, Sep. 2020. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0042/ (visited on 12/17/2022).

[22] *Initial Access, Tactic TA0001 - Enterprise | MITRE ATT&CK®*, Jul. 2019. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0001/ (visited on 12/17/2022).

[23] *Execution, Tactic TA0002 - Enterprise | MITRE ATT&CK®*, Jul. 2019. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0002/ (visited on 12/17/2022).

[24] *Persistence, Tactic TA0003 - Enterprise | MITRE ATT&CK®*, Jul. 2019. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0003/ (visited on 12/17/2022).

[25] *Privilege Escalation, Tactic TA0004 - Enterprise | MITRE ATT&CK®*, Jan. 2021. [Online]. Available: https://attack.mitre.org/tactics/TA0004/ (visited on 12/17/2022).

[26] *Defense Evasion, Tactic TA0005 - Enterprise | MITRE ATT&CK®*, Jul. 2019. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0005/ (visited on 12/17/2022).

[27] *Credential Access, Tactic TA0006 - Enterprise | MITRE ATT&CK®*, Jul. 2019. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0006/ (visited on 12/17/2022).

[28] *Discovery, Tactic TA0007 - Enterprise | MITRE ATT&CK®*, Jul. 2019. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0007/ (visited on 12/17/2022).

[29] *Lateral Movement, Tactic TA0008 - Enterprise | MITRE ATT&CK®*, Jul. 2019. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0008/ (visited on 12/17/2022).

[30] *Collection, Tactic TA0009 - Enterprise | MITRE ATT&CK®*, Jul. 2019. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0009/ (visited on 12/17/2022).

[31] *Command and Control, Tactic TA0011 - Enterprise | MITRE ATT&CK®*, Jul. 2019. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0011/ (visited on 12/17/2022).

[32] *Exfiltration, Tactic TA0010 - Enterprise | MITRE ATT&CK®*, Jul. 2019. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0010/ (visited on 12/17/2022).

[33] *Impact, Tactic TA0040 - Enterprise | MITRE ATT&CK®*, Jul. 2019. [Online]. Available: https://attack.mitre.org/versions/v12/tactics/TA0040/ (visited on 12/17/2022).

[34] Brian Desmond, Joe Richards, Robbie Allen, and Alistair G. Lowe-Norris, *Active Directory*, en, Fifth edition, Brian Desmond and Joe Richards, Eds. Beijing; Sebastopol: O'Reilly Media, 2013, ISBN: 978-1-4493-2002-7.

[35] Afnan Binduf, Hanan Othman Alamoudi, Hanan Balahmar, Shatha Alshamrani, Haifa Al-Omar, and Naya Nagy, "Active Directory and Related Aspects of Security," en, in *2018 21st Saudi Computer Society National Computer Conference (NCC)*, Riyadh: IEEE, Apr. 2018, pp. 4474–4479, ISBN: 978-1-5386-4110-1. DOI: 10.1109/NCG.2018.8593188. [Online]. Available: https://ieeexplore.ieee.org/document/8593188/ (visited on 09/09/2022).

[36] Basem Ibrahim Mokhtar, Anca D. Jurcut, Mahmoud Said ElSayed, and Marianne A. Azer, "Active Directory Attacks—Steps, Types, and Signatures," en, *Electronics*, vol. 11, no. 16, Aug. 2022, ISSN: 2079-9292. DOI: 10.3390/electronics11162629. [Online]. Available: https://www.mdpi.com/2079-9292/11/16/2629 (visited on 09/09/2022).

[37] Clifford Neuman, Sam Hartman, Kenneth Raeburn, and Taylor Yu, "The Kerberos Network Authentication Service (V5)," Internet Engineering Task Force, Request for Comments RFC 4120, Jul. 2005, Num Pages: 138. DOI: 10.17487/RFC4120. [Online]. Available: https://datatracker.ietf.org/doc/rfc4120 (visited on 12/03/2022).

[38] Jason Garman, *Kerberos: The Definitive Guide*, en, First edition, Mike Loukides, Ed. Sebastopol: O'Reilly Media, 2003, ISBN: 978-0-596-00403-3.

[39] Aleksandr Ometov, Sergey Bezzateev, Niko Mäkitalo, Sergey Andreev, Tommi Mikkonen, and Yevgeni Koucheryavy, "Multi-Factor Authentication: A Survey," en, *Cryptography 2018*, vol. 2, no. 1, Jan. 2018, ISSN: 2410-387X. DOI: 10.3390/cryptography2010001. [Online]. Available: http://www.mdpi.com/2410-387X/2/1/1 (visited on 12/03/2022).

[40] "Detection and Prevention of Attacks on Active Directory using SIEM," en, in *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2020*, ser. Smart Innovation, Systems and Technologies, Muthuraj S., Sethumadhavan M., Amritha P. P., and Santhya R., Eds., 2nd ed., vol. 196, Singapore: Springer Singapore, 2021, pp. 533–541, ISBN: 9789811570612 9789811570629. DOI: 10.1007/978-981-15-7062-9. [Online]. Available: https://link.springer.com/10.1007/978-981-15-7062-9 (visited on 09/09/2022).

[41] Justin Turner, *SPN- und UPN-Eindeutigkeit*, de-de, May 2022. [Online]. Available: https://docs.microsoft.com/de-de/windows-server/identity/ad-ds/manage/component-updates/spn-and-upn-uniqueness (visited on 09/10/2022).

[42] John D. Cyber, *Let's talk about Kerberos.*. en, Jun. 2021. [Online]. Available: https://johndcyber.com/lets-talk-about-kerberos-6376d0bd7d91 (visited on 09/10/2022).

[43] Jennifer G Steiner, Clifford Neuman, and Jeffrey I Schiller, "Kerberos: An Authentication Service for Open Network Systems," en, Jan. 1988.

[44] Alva Duckwall and Benjamin Delpy, *Abusing Microsoft Kerberos*, en, Las Vegas, Aug. 2014. (visited on 09/14/2022).

[45] Mariko Fujimoto, Wataru Matsuda, and Takuho Mitsunaga, "Detecting Abuse of Domain Administrator Privilege Using Windows Event Log," en, in *2018 IEEE Conference on Application, Information and Network Security (AINS)*, Langkawi, Malaysia: IEEE, Nov. 2018, pp. 15–20, ISBN: 978-1-5386-6925-9. DOI: 10.1109/AINS.2018.8631459. [Online]. Available: https://ieeexplore.ieee.org/document/8631459/ (visited on 09/09/2022).

[46] Carlos Diaz Motero, Juan Ramon Bermejo Higuera, Javier Bermejo Higuera, Juan Antonio Sicilia Montalvo, and Nadia Gamez Gomez, "On Attacking Kerberos Authentication Protocol in Windows Active Directory Services: A Practical Survey," en, *IEEE Access*, vol. 9, pp. 109 289–109 319, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3101446. [Online]. Available: https://ieeexplore.ieee.org/document/9501961/ (visited on 09/09/2022).

[47] Sean Metcalf, *Red vs. Blue: Modern Active Directory Attacks, Detection & Protection*, en, Las Vegas, Aug. 2015. [Online]. Available: https://www.blackhat.com/%20docs/us-15/materials/us-15-Metcalf-Red-Vs-Blue-Modern-Active-Directory-Attacks-Detection-And-Protection.pdf (visited on 09/20/2022).

[48] Yuri Diogenes and Erdal Ozkaya, *Cybersecurity, attack and defense strategies: infrastructure security with Red Team and Blue Team tactics*, en. Birmingham: Packt Publishing Ltd., 2018, ISBN: 978-1-78847-529-7.

[49] Nirmal Sharma, *Net Commands for Operating Systems - Windows Server*, en-us, Sep. 2021. [Online]. Available: https://learn.microsoft.com/en-us/troubleshoot/windows-server/networking/net-commands-on-operating-systems (visited on 09/17/2022).

[50] *ActiveDirectory Module*, en-us, May 2022. [Online]. Available: https://learn.microsoft.com/en-us/powershell/module/activedirectory/ (visited on 09/17/2022).

[51] Tjada Nelson and Houssain Kettani, "Open Source PowerShell-Written Post Exploitation Frameworks Used by Cyber Espionage Groups," en, in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, San Jose, CA, USA: IEEE, Mar. 2020, pp. 451–456, ISBN: 978-1-72817-283-5. DOI: 10.1109/ICICT50521.2020.00078. [Online]. Available: https://ieeexplore.ieee.org/document/9092030/ (visited on 09/17/2022).

[52] Will Schroeder, *PowerShellMafia/PowerSploit*, original-date: 2012-05-26T16:08:48Z, Jul. 2018. [Online]. Available: https://github.com/PowerShellMafia/PowerSploit/blob/d943001a7defb5e0d1657085a77a0e78609be58f/Recon/PowerView.ps1 (visited on 09/17/2022).

[53] Michiel Lemmens, *BloodHound – Sniffing Out the Path Through Windows Domains | SANS Institute*, en, Jun. 2021. [Online]. Available: https://www.sans.org/blog/bloodhound-sniffing-out-path-through-windows-domains/ (visited on 09/17/2022).

[54] Andy Robbins, Rohan Vazarkar, and Will Schroeder, *Getting Started with BloodHound*, original-date: 2016-04-17T18:36:14Z, Sep. 2022. [Online]. Available: https://github.com/BloodHoundAD/BloodHound (visited on 09/17/2022).

[55] Juuso Myllylä and Andrei Costin, "Reducing the Time to Detect Cyber Attacks - Combining Attack Simulation With Detection Logic," en, in *Proceedings of the 29th Conference of FRUCT Association*, Oulo, Finland, Oct. 2021, pp. 465–474. (visited on 09/17/2022).

[56] Raj Badhwar, "Advanced Active Directory Attacks and Prevention," en, in *The CISO's Next Frontier: AI, Post-Quantum Cryptography and Advanced Security Paradigms*, Cham: Springer International Publishing, 2021, pp. 131–144, ISBN: 978-3-030-75353-5 978-3-030-75354-2. DOI: 10.1007/978-3-030-75354-2_13. [Online]. Available: https://link.springer.com/10.1007/978-3-030-75354-2 (visited on 09/17/2022).

[57] Benjamin Delpy, *GitHub - gentilkiwi/mimikatz: A little tool to play with Windows security*, Sep. 2022. [Online]. Available: https://github.com/gentilkiwi/mimikatz (visited on 09/17/2022).

[58] Dimo Dimov and Yulian Tzonev, "Pass-the-Hash: One of the Most Prevalent Yet Underrated Attacks for Credentials Theft and Reuse," en, in *Proceedings of the 18th International Conference on Computer Systems and Technologies*, Ruse Bulgaria: ACM, Jun. 2017, pp. 149–154, ISBN: 978-1-4503-5234-5. DOI: 10.1145/3134302.3134338. [Online]. Available: https://dl.acm.org/doi/10.1145/3134302.3134338 (visited on 09/17/2022).

[59] Patrick Ah-Fat, Michael Huth, Joshua Neil, Rob Mead, and Tim Burrell, "Effective Detection of Credential Thefts from Windows Memory: Learning Access Behaviours to Local Security Authority Subsystem Service," en, in *In Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, San Sebastion, Spain: USENIX Association, Oct. 2020, pp. 181–194, ISBN: 978-1-939133-18-2.

[60] Lukáš Kotlaba, Simona Buchovecká, and Róbert Lórencz, "Active Directory Kerberoasting Attack: Detection using Machine Learning Techniques," en, in *Proceedings of the 7th International Conference on Information Systems Security and Privacy*, Online: SCITEPRESS - Science and Technology Publications, Feb. 2021, pp. 376–383, ISBN: 978-989-758-491-6. DOI: 10.5220/0010202803760383.

[Online]. Available: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010202803760383 (visited on 09/17/2022).

[61] Jeff Warren, *Unconstrained Delegation Permissions*, en-US, Jun. 2017. [Online]. Available: https://stealthbits.com/blog/unconstrained-delegation-permissions/ (visited on 09/17/2022).

[62] Miguel Soria-Machado, Didzis Abolins, Ciprian Boldea, and Krzysztof Socha, "Kerberos Golden Ticket Protection," en, *CERT-EU Security Whitepaper 2014-007*, Jul. 2016.

[63] Toni Boger, *Directory Services Restore Mode (DSRM)*, en, Oct. 2012. [Online]. Available: https://www.techtarget.com/searchwindowsserver/definition/Directory-Services-Restore-Mode-DSRM (visited on 09/20/2022).

[64] *How Access Control Works in Active Directory Domain Services*, en-us, Aug. 2020. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/ad/how-access-control-works-in-active-directory-domain-services (visited on 09/20/2022).

[65] Sean Metcalf, *Sneaky Active Directory Persistence #15: Leverage AdminSDHolder & SDProp to (Re)Gain Domain Admin Rights*, en-US, Sep. 2015. [Online]. Available: https://adsecurity.org/?p=1906 (visited on 09/20/2022).

[66] Nikhil SamratAshok Mittal, *RACE - Minimal Rights and ACE for Active Directory Dominance*, en, Aug. 2019. [Online]. Available: http://www.labofapenetrationtester.com/2019/08/race.html (visited on 09/21/2022).

[67] *Running Remote Commands - PowerShell*, en-us, Oct. 2021. [Online]. Available: https://learn.microsoft.com/en-us/powershell/scripting/learn/remoting/running-remote-commands (visited on 09/21/2022).

[68] Vinay Pamnani, *4672(S) Besondere Berechtigungen, die der neuen Anmeldung zugewiesen sind.* de-de, Oct. 2022. [Online]. Available: https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4672 (visited on 10/25/2022).

[69] Vinay Pamnani, *4688(S) Ein neuer Prozess wurde erstellt.* de-de, Oct. 2022. [Online]. Available: https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4688 (visited on 10/25/2022).

[70] Vinay Pamnani, *4768(S, F) Es wurde ein Kerberos-Authentifizierungsticket (TGT) angefordert.* de-de, Oct. 2022. [Online]. Available: https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4768 (visited on 10/25/2022).

[71] Vinay Pamnani, *4769(S, F) Ein Kerberos-Serviceticket wurde angefordert.* de-de, Oct. 2022. [Online]. Available: https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-4769 (visited on 10/25/2022).

[72] Vinay Pamnani, *5140(S, F) Auf ein Netzwerkfreigabeobjekt wurde zugegriffen.* de-de, Oct. 2022. [Online]. Available: https://learn.microsoft.com/de-de/windows/security/threat-protection/auditing/event-5140 (visited on 10/25/2022).

[73] K. Sornalakshmi, "Detection of DoS attack and zero day threat with SIEM," en, in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai: IEEE, Jun. 2017, pp. 1–7, ISBN: 978-1-5386-2745-7. DOI: 10.1109/ICCONS.2017.8250515. [Online]. Available: http://ieeexplore.ieee.org/document/8250515/ (visited on 09/09/2022).

[74] S. Sandeep Sekharan and Kamalanathan Kandasamy, "Profiling SIEM tools and correlation engines for security analytics," en, in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai: IEEE, Mar. 2017, pp. 717–721, ISBN: 978-1-5090-4442-9. DOI: 10.1109/WiSPNET.2017.8299855. [Online]. Available: http://ieeexplore.ieee.org/document/8299855/ (visited on 10/25/2022).

[75] Chih-Hung Hsieh, Chia-Min Lai, Ching-Hao Mao, Tien-Cheu Kao, and Kuo-Chen Lee, "Ad2: Anomaly detection on active directory log data for insider threat monitoring," en, in *2015 International Carnahan Conference on Security Technology (ICCST)*, Taipei, Taiwan: IEEE, Sep. 2015, pp. 287–292, ISBN: 978-1-4799-8690-3 978-1-4799-8691-0. DOI: 10.1109/CCST.2015.7389698. [Online]. Available: http://ieeexplore.ieee.org/document/7389698/ (visited on 09/09/2022).

[76] *CIS Critical Security Controls v8*, en, May 2021. [Online]. Available: https://www.cisecurity.org/controls/cis-controls-list (visited on 12/20/2022).

[77] *ATT&CK® Navigator*. [Online]. Available: https://mitre-attack.github.io/attack-navigator/ (visited on 12/20/2022).

[78] *DeTT&CT Editor*. [Online]. Available: https://rabobank-cdc.github.io/dettect-editor/#/home (visited on 12/20/2022).

[79] Sadeq AlHamouz and Ahmad Abu-Shareha, "Hybrid Classification Approach Using Self-Organizing Map and Back Propagation Artificial Neural Networks for Intrusion Detection," in *2017 10th International Conference on Developments in eSystems Engineering (DeSE)*, IEEE, 2017, pp. 83–87. DOI: 10.1109/DeSE.2017.28.

[80] Toby Bussa, Kelly Kavanagh, and John Collins, *Critical Capabilities for Security Information and Event Management*, en, Apr. 2022. [Online]. Available: https://www.gartner.com/en/documents/4003177 (visited on 11/06/2022).

[81] Morten Dalgaard, *Das SIEM-System von Logpoint - Funktionsweise*, de, 2022. [Online]. Available: https://www.logpoint.com/de/produkt/funktionsweise/ (visited on 01/18/2023).

[82] *Routing Policies — Data Integration latest documentation*, 2023. [Online]. Available: https://docs.logpoint.com/docs/data-integration-guide/en/latest/Configuration/Routing%20Policies.html (visited on 01/18/2023).

[83] *Normalization Policies — Data Integration latest documentation*, 2023. [Online]. Available: https://docs.logpoint.com/docs/data-integration-guide/en/latest/Configuration/Normalization%20Policies.html (visited on 01/18/2023).

[84] *Enrichment Policies — Data Integration latest documentation*, 2023. [Online]. Available: https://docs.logpoint.com/docs/data-integration-guide/en/latest/Configuration/Enrichment%20Policies.html (visited on 01/18/2023).

[85] *Processing Policies — Data Integration latest documentation*, 2023. [Online]. Available: https://docs.logpoint.com/docs/data-integration-guide/en/latest/Configuration/Processing%20Policies.html (visited on 01/18/2023).

[86] *Valid Accounts, Technique T1078 - Enterprise | MITRE ATT&CK®*, Oct. 2022. [Online]. Available: https://attack.mitre.org/versions/v12/techniques/T1078/ (visited on 12/28/2022).

[87] *Account Manipulation, Technique T1098 - Enterprise | MITRE ATT&CK®*, Oct. 2022. [Online]. Available: https://attack.mitre.org/versions/v12/techniques/T1098/ (visited on 12/30/2022).

[88] *Exploitation for Privilege Escalation, Technique T1068 - Enterprise | MITRE ATT&CK®*, Jun. 2022. [Online]. Available: https://attack.mitre.org/versions/v12/techniques/T1068/ (visited on 12/30/2022).

[89] *Brute Force, Technique T1110 - Enterprise | MITRE ATT&CK®*, May 2022. [Online]. Available: https://attack.mitre.org/versions/v12/techniques/T1110/ (visited on 12/30/2022).

[90] María del Carmen Prudente Tixteco, Lidia Prudente Tixteco, Gabriel Sánchez Pérez, and Linda Karina Toscano Medina, "Intrusion Detection Using Indicators of Compromise Based on Best Practices and Windows Event Logs," en, in *ICIMP 2016 : The Eleventh International Conference on Internet Monitoring and Protection*, Valencia, Spanien: IARIA, May 2016, pp. 29–37, ISBN: 978-1-61208-475-6. [Online]. Available: https://community.securid.com/yfcdo34327/attachments/yfcdo34327/netwitness-blog/1015/3/icimp_2016_2_20_30032.pdf (visited on 12/30/2022).

[91] *Windows Event Log :: NXLog Documentation*, en. [Online]. Available: https://docs.nxlog.co/userguide/integrate/windows-eventlog.html (visited on 01/02/2023).

[92] Karl Bridge, *Eventlog Key - Win32 apps*, en-us, Aug. 2021. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/eventlog/eventlog-key (visited on 12/30/2022).

[93] Karl Bridge, *Event Types - Win32 apps*, en-us, Jul. 2021. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/eventlog/event-types (visited on 12/30/2022).

[94] Vinay Pamnani, *Basic security audit policies (Windows 10)*, en-us, Sep. 2022. [Online]. Available: https://learn.microsoft.com/en-us/windows/security/threat-protection/auditing/basic-security-audit-policies (visited on 12/31/2022).

[95] Vinay Pamnani, *Advanced security audit policies (Windows 10)*, en-us, Sep. 2022. [Online]. Available: https://learn.microsoft.com/en-us/windows/security/threat-protection/auditing/advanced-security-auditing (visited on 12/31/2022).

[96] Vinay Pamnani, *Advanced security auditing FAQ*, en-us, FAQ. [Online]. Available: https://learn.microsoft.com/en-us/windows/security/threat-protection/auditing/advanced-security-auditing-faq (visited on 12/31/2022).

[97] Logpoint, *Installation and Upgrade — LogPoint Agent latest documentation*. [Online]. Available: https://docs.logpoint.com/docs/logpoint-agent/en/latest/Installing%20LogPoint%20Agent%20Collector.html (visited on 01/02/2023).

[98] Vinay Pamnani, *4648(S) A logon was attempted using explicit credentials. (Windows 10)*, en-us, Dec. 2022. [Online]. Available: https://learn.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4648 (visited on 01/07/2023).