Hochschule Worms

Fachbereich Informatik

Studiengang Angewandte Informatik B.Sc.

TBD

Bachelorarbeit xxx

Bruno Macedo da Silva
676839
inf3645@hs-worms.de
Bebelstraße 22 Z10
67549 Worms

Betreuer	Prof. Dr. Zdravko Bozakov
Bearbeitungszeitraum:	Sommersemester 2023
Abgabedatum:	xx. xxx 2023
Sperrvermerk:	Ja/Nein

Inhaltsverzeichnis

Abstract

The aim of this thesis is to develop a reliable, cost-effective solution for monitoring security events by utilizing an Open Source, Security Information and Event Management (SIEM)-like tool. Since many existing SIEM solutions are either proprietary or offer limited free features, we chose to use Grafana and its integrated tools - Promtail, Loki, and Alerting - to create our monitoring system. Grafana is primarily used to generate customizable graphics based on user input, and in our study, we used Secure Shell Protocol (SSH) log files as input. Promtail extracted the files from Endpoints and sent them to Loki, which used defined rules to aggregate and filter the content in order to identify possible cyberattacks against an SSH server. Once the information was extracted, Grafana was used to provide a visual overview of the SSH connections. Additionally, we employed the Alerting tool to send notifications about potential attacks identified by our rules. The ruleset we used to recognize potential attacks and the descriptions of these attacks were based on the Mitre ATT&CK Matrix. We found that the combined use of these tools was reliable, affordable, and useful for detecting static-based attacks. The main challenges in using these tools as a replacement for a SIEM solution are properly defining the ruleset used to read and extract information about cyberattacks from log files and adapting those rules to scenarios where attacks have more dynamic flows.

Keywords: Monitoring Tool, Grafana Loki Cyberattacks,SIEM

Abbildungsverzeichnis

Glossar

Abfragesprache Die *Query Language* funktioniert wie ein Filter für die Suche nach spezifischen Daten in einer Datenbank (at, 2022). 18

Application Programming Interface (API) bezieht sich auf Code und Regeln, die die Kommunikation zwischen verschiedenen Anwendungen ermöglichen. In diesem Fall kann eine Anwendung eine Anfrage an eine andere Anwendung senden, um Daten zu holen oder zu senden (IBM, 2020). 32, 43

Brute-Force Angriffe systematische Versuche, Zugangsdaten oder andere sensible Daten zu erraten, indem verschiedene Buchstaben, Ziffern und Symbole kombiniert werde (Sowmya et al., 2012). 9, 23, 34, 53

Container funktionieren ähnlich wie virtuelle Maschinen (VMs), jedoch sind Container Anwendungen mit den notwendigen Ressourcen, um eingepackte Anwendungen auszuführen. Container werden oft für einzelne verwendet und teilen Ressourcen wie den Kernel des Host-Betriebssystems. Jeder Container ist in einer isolierten Umgebung mit den notwendigen Ressourcen für den Betrieb der ausgewählten Anwendung. Docker ist eine der bekanntesten Plattformen zur Verwaltung von Containern (Douglis and Nieh, 2019). 20, 24, 30, 32

Cortex ist eine Open-Source-Plattform zur Verwaltung und Weiterverarbeitung von Sicherheitsvorfällen. Es fungiert als Analyse-Engine, indem es Informationen sammelt und je nach Fall Antworten oder Aktionen durchführt. Cortex kann eigenständig oder in Kombination mit anderen Tools verwendet werden (Project, 2021). 41

Cyberangriff Angriffe über Cyberspace. Solche Angriffe zielen darauf ab, Unternehmen und ihre Infrastrukturen zu zerstören, zu lähmen, zu kontrollieren oder die Integrität ihrer Daten zu stehlen oder zu manipulieren (NIST, 2020b). 1, 2, 5, 20–22

Confidentiality, Integrity and Availability (CIA) beschreiben die drei wichtigsten Schutzziele der IT-Sicherheit: Vertraulichkeit, Integrität und Verfügbarkeit (Wendzel, 2018). 7

Taktiken, Techniken, Prozeduren (TTP) beschreiben in der Mittre ATT&CK Matrix Verhalten, Methode un Mustern bei Cyberangriffe (Maymi et al., 2017). 1, 21, 24

Cyber Kill Chain (CKC) auch *Cyberattack Lifecycle* genannt, bezieht sich auf ein Sicherheitsmodell für die Identifizierung, Analyse und Unterbrechung von fortgeschrittenen Cyberangriffen. Dieses Modell hat sieben festgelegte Phasen: *Reconnaissance*, *Weaponization*, *Delivery*, *Exploitation*, *Installation*, *Command & Control (C2)* und *Actions on Objectives* (Martin, 2018). 9, 21, 47

Cybersicherheit - Diese Domäne umfasst Kenntnisse und Methoden für den Schutz,

die Prävention und Wiederherstellung von elektronischen Kommunikationsmitteln und deren Inhalten. Dabei konzentriert sie sich auf deren Verfügbarkeit, Integrität, Authentizität, Vertraulichkeit und Nichtabstreitbarkeit. (NIST, 2020b). 22

Endpoint bezieht sich auf Geräte oder Systeme, die mit dem Netzwerk verbunden sind. Diese können z.B. Handys, Servers, Computers, Sensoren sein (Microsoft Security, 2022). iii, 4, 5, 16, 18, 32, 33, 35, 41, 43

Hashwerte sind Zeichenfolgen, die durch Anwenden einer mathematischen Funktion (Hashfunktion) auf einen Text oder eine Datei erzeugt werden. Die Rückführung auf das ursprüngliche Objekt aus dem Hashwert sollte jedoch unmöglich sein (Wendzel, 2018). 23

Hypertext Transfer Protocol (HTTP) ist die Grundlage des Internets. Dieses Protokoll definiert die Regeln für die Übertragung von Texten und Dateien im Internet. Das Protokoll verwendet acht Methoden, um die Kommunikation zwischen Clients und Servern herzustellen: *GET, POST, HEAD, DELETE, CONNECT, OPTIONS, PUT* und *TRACE* (Chai and Ferguson, 2021) and (tutorialspoint, 2009). 32

Falsch Positiv ist eine Warnmeldung einer nicht vorhandenen Verwundbarkeit (NIST, 2020c). 13

grafische Benutzeroberfläche (GUI) - Es handelt sich dabei um eine visuelle Schnittstelle, die es dem Benutzer ermöglicht, mit Anwendungen mittels Symbolen und grafischen Elementen zu interagieren. Im Gegensatz dazu verwendet die textbasierte Benutzeroberfläche (CLI) Befehlszeilen und Texteingabe zur Steuerung von Anwendungen (Fu, 2018). 11, 24, 37

Hydra ist eine Open Source Tool für Brute-Force Angriffe (Kali, 2022a). 25, 27

Health Insurance Portability and Accountability Act (HIPAA) ist ein US-Bundesgesetz über den Schutz von sensitiven personenbezogenen Gesundheitsdaten (U.S. Department of Health & Human Services, 2016). 7

Kali ist eine Open-Source-Linux-Distribution, die speziell auf den Einsatz von Sicherheitstools für Angriffe und Sicherheitstests ausgelegt ist (Kali, 2022b). 24

Künstliche Intelligenz (KI) bezeichnet die Fähigkeit, Maschinen menschenähnliche kognitive Fähigkeiten wie Verständnis, Entscheidungsfindung, Lernen und Problemlösung zu entwickeln (Collins et al., 2021). 46

LogQL ist eine für Grafana Loki entwickelte Abfragesprache. Sie wird verwendet, um Logdateien zu zusammenzustellen (Grafana Labs, 2021c). 34–37, 42, 45

Multi-Faktor-Authentisierung (MFA) bezeichnet ein Authentifizierungsverfahren, bei dem mindestens zwei unabhängige Komponenten zur Identitätsprüfung verwendet

werden, um eine höhere Sicherheit zu gewährleisten. Zum Beispiel kann ein Benutzer aufgefordert werden, sich mit einem Passwort und einem Fingerabdruck oder einem Token und/oder einer Gesichtserkennung zu authentifizieren (Ibrokhimov et al., 2019). 34

Machine Learning (ML) bezieht sich auf die Fähigkeit von Systemen, automatisch und menschenähnlich Probleme zu lösen und spezifische Aufgabe zu erledigen (Janiesch et al., 2021). 8, 14

Mitre ATT&CK Abkürzung für *Adversarial Tactics, Techniques and Common Knowledge*. Es bezieht sich auf eine weltweit zugängliche Wissensbasis mit detaillierter Beschreibung, Klassifizierung und Bekämpfung von verschiedenen Angriffstechniken (MITRE ATT&CK, 2018a). iii, 1, 2, 19–22, 24, 45

Mimir ein in Grafana integriertes Tool, das ähnlich wie Grafana Loki funktioniert. Es ermgölicht skalierbare Dateispeicherung, Bearbeitung und Abfrage mit der Abfragesprache LogQl (Grafana Labs, 2022f). 41

National Institute of Standards and Technology (NIST) ist eine US-Behörde, die für die Regelungen, Vereinheitlichung und Weiterentwicklung von Standards im Bereich Informationstechnologie zuständig ist (NIST, 2020a). 1

Open Source beschreibt Software, die folgende Voraussetzungen erfüllen: freie Verteilung, Kopierung, Modifizierung und Nutzung und keine Diskriminierung gegenüber Personen und/oder Gruppe (Open Source Initiative, 2007). iii, 1, 2, 4, 7, 9, 12, 15, 19, 45, 47

Password Spraying ist ein Angriff gegen Anmeldedaten, indem mögliche Passwörter gegen verschiedenen viele Benutzernamen verwendet werden. Das Ziel dieses Angriffes ist eine Kontosperrung zu vermeiden, indem wenige Versuche pro Nutzer stattfindet (Swathi, 2022). 23, 25, 28, 29

Password Stuffing ist ein Angriff gegen Passwörtern, indem bekannte Anmeldedaten von vorherigen Angriffen verwendet werden. Dieser Angriff basiert sich auf die Idee, dass Nutzer dasselbe Passwort für verschiedenen Systemen verwenden (Ba et al., 2021). 23, 26, 27

Payment Card Industry Data Security Standard (PCDI DSS) sind Sicherheitsstandards, die von Unternehmen, die Kreditkarten akzeptieren, verarbeiten, speichern oder übertragen, eingehalten werden müssen (Centers for Disease Control and Prevention, 2016). 7

Polymorphe Malware sind Schadprogramme, deren Signatur sich ständig ändern, um nicht von Anti-Malware-Systemen erkannt zu werden (Selamat et al., 2016). 46

Network Operations Center (NOC) ist ein zentralisierter Bereich eines Unternehmens, der für die Überwachung und Verwaltung von Netzwerkaktivitäten verantwortlich ist. (Mohammed et al., 2021). 15

Plugin sind optionale Software-Komponenten, die weitere Funktionalitäten zu einer Anwendung hinzufügen (IT-Service.Network, 2020). 15, 17, 47

Prometheus ist ein Open-Source-Tool der Firma SoundCloud. Es dient der Überwachung und Erstellung von Warnmeldungen, die auf der Grundlage von vordefinierten Regeln konfiguriert werden (Prometheus, 2016). 35, 41

Proprietär bezieht sich auf Software, die einer Firma oder Person gehört. Für die Nutzung ist in der Regel der Kauf einer Lizenz erforderlich. In diesem Fall haben Kunden nur begrenzten oder keinen Zugriff auf den Quellcode (Nexcess, 2022). 2, 7, 19

Rockyou ist eine Textdatei mit über 8 Milliarden Passwörtern im Klartext. Diese Datei stammt aus einem Angriff gegen Yahoo im Jahr 2009 und wird seitdem ständig aktualisier (Mikalauskas, 2023). 25, 28

Security Operations Center (SOC) ist ein zentralisierter Bereich eines Unternehmens, der für die Überwachung, Identifizierung, Bewertung und Reaktion auf Sicherheitsvorfälle verantwortlich ist. (Vielberth, 2021). 1, 4

Secure Shell Protocol(SSH) ist ein Netzwerkprotokoll, das eine verschlüsselte Verbindung zwischen Endpunkten bietet. SSH wird meistens für die Fernadministration von Computern verwendet. Dieses Protokoll ermöglicht die Erstellung einer sicheren Verbindung in einer unsicheren Umgebung (Wendzel, 2018). iii, 25

Ubuntu ist eine Linux-Distribution, die oft für Server, Clients und Internet of Things (IoT) verwendet wird (Ubuntu, 2023b). 24

Use Cases sind narrative Beschreibungen der Interaktionen zwischen Systemen und Benutzern. Sie dienen der Anforderungserhebung für ein System (Savic et al., 2012). 2, 9, 19

Virtuelle Maschine (VM) ist eine Kopie der Hardware-Struktur mit einer eigenen Aufteilung von Ressourcen und einem eigenen Betriebssystem. Auf einer physischen Maschine, auch Host genannt, können mehrere solcher VMs ausgeführt werden. Sie emulieren ein echtes und unabhängiges System (Tanenbaum, 2009). 20, 24

Webhook Webhooks funktionieren ähnlich wie APIs, ohne dass die Client-Seite nach Aktualisierungen fragen muss. Bei Webhooks sendet der Server die Aktualisierung, sobald sie verfügbar ist. Die Kommunikation findet in Echtzeit statt (Tas, 2021). 43

Abkürzungsverzeichnis

API Application Programming Interface.

BSI Bundesamt für Sicherheit in der Informationstechnik.

CIA Confidentiality, Integrity and Availability.

TTP Taktiken, Techniken, Prozeduren.

CKC Cyber Kill Chain.

HTTP Hypertext Transfer Protocol.

IDS Intrusion Detection System.

GUI grafische Benutzeroberfläche.

IPS Intrusion Prevention System.

FPO Fachspezifische Prüfungsordnung.

HIPAA Health Insurance Portability and Accountability Act.

KI Künstliche Intelligenz.

MFA Multi-Faktor-Authentisierung.

ML Machine Learning.

NIST National Institute of Standards and Technology.

OTX Open Threat Exchange.

LML Log Monitoring Lackey.

OSSIM Open Source Security Information Management.

PCDI DSS Payment Card Industry Data Security Standard.

NOC Network Operations Center.

owasp Open Web Application Security Project.

RegExp Regular Expression.

SIEM Security Information and Event Management.

SEM Security Event Management.

SIM Security Information Management.

SOC Security Operations Center.

SSH Secure Shell Protocol.

USM Unified Security Management.

VM virtuelle Maschine.

1. Einleitung

Der heutige Netzwerkverkehr ist fast tausendfach größer als vor 20 Jahre (Roser et al., 2015). Das Internet wird heutzutage für fast all unsere Tätigkeiten verwendet: Soziale Netzwerke, Video und Audio-Streaming, Einkauf, behördliche Angelegenheiten und viele andere. So viel Verkehr generiert eine unermessliche Menge von Daten, die alle möglichen Inhalte beinhalten, von unschuldigen Anfragen nach einem eigenen Kontostand bis zur Ausführung von beabsichtigten Anfragen, um Systeme lahmzulegen. Um ersteres vom letzterem zu unterscheiden, verwenden viele Firmen das sogenannte Security Information and Event Management (SIEM) oder Log-Analyse-Tools.

Das National Institute of Standards and Technology (NIST) definiert SIEM als Software für die Sammlung, Anpassung, Analyse, Überwachung und Bedrohungserkennung von Sicherheitsdaten aus verschiedenen Quellen (NIST, 2020d). Die Bewertung dieser Daten spielt eine wesentliche Rolle bei solchen Anwendungen, um zu entscheiden, ob es sich um legitime Anfrage oder um einen Cyberangriffe handelt. Mit den Daten von SIEM kann das Security Operations Center (SOC) Team Maßnahmen ergreifen. Log Analysis und Log Management beziehen sich auf die Sammlung, Bearbeitung, Speicherung und/oder Löschen, Weiterleitung und Überwachung von Loginformationen. In dieser Arbeit benutzen wir den Begriff „Log-Analyse-Tools“, um diese Systeme zu referenzieren.

In diesem Projekt recherchieren und vergleichen wir existierende SIEM und Log-Analyse-Tools. Danach entscheiden wir uns für eine Open Source Lösung, um eine kostengünstige Verbreitung und Implementierung zu ermöglichen. Mit dem ausgewählten Tool wollen wir spezifische Logdateien analysieren und bewerten, damit wir demnächst potenzielle Angriffe erkennen können. Die Regelsätzen für die Angriffserkennung sollen automatisch mithilfe der Taktiken, Techniken, Prozeduren (TTP) von Mitre ATT&CK aufgebaut werden.

Unser Ziel ist es, eine umfangreiche Open Source Lösung zu finden bzw. zu gesltaten, die uns ermöglicht, Cyberangriffe nach vordefinierten Regelsätzen zu detektieren. Proprietäre Lösungen gibt es viele auf dem Markt. Sie sind meistens kostenpflichtig und verlangen spezielle Wartung. Da sich solche Lösungen eher an große Konzerne richten, beschäftigen wir uns mit dem Aufbau und Strukturierung einer eigenen Lösung mithilfe von Open Source Tools.

Diese Arbeit wird in folgende Teile geteilt:

- Definition von SIEMs und Log-Analyse-Tools
- Beschreibung von existierenden Proprietären und Open Source Lösungen
- Entscheidung für die Implementation einer Open Source Lösungen
- Installation und Konfiguration von der ausgewählten Anwendung
- Implementierung von zwei spezifischen Cyberangriffen
- Definition der Use Cases und Implementierung von Regelsätze für die automatische Erkennung von der vorherigen Angriffen anhand der Mitre ATT&CK Matrix
- Empfang, Bearbeitung und Eingabe der spezifischen Logdateien der Hochschule in der ausgewählten Lösung

1.1. Problemstellung

Während der Entwicklung dieser Arbeit wollen wir uns mit folgenden Fragenstellung beschäftigen:

- Wie können wir ein Log-Analyse-Tool konfigurieren, dass es vordefinierte Angriffe nach der Mitre ATT&CK Matrix automatisch erkennen kann?
- Wie können wir allgemeine Uses Cases definieren, sodass wir sie später für verschiedene Angriffsmuster nach Mitre ATT&CK Matrix leicht anpassen können?

Das folgende Diagramm stellt den Aufbau und Entwicklung dieser Arbeit dar, wie oben beschrieben:

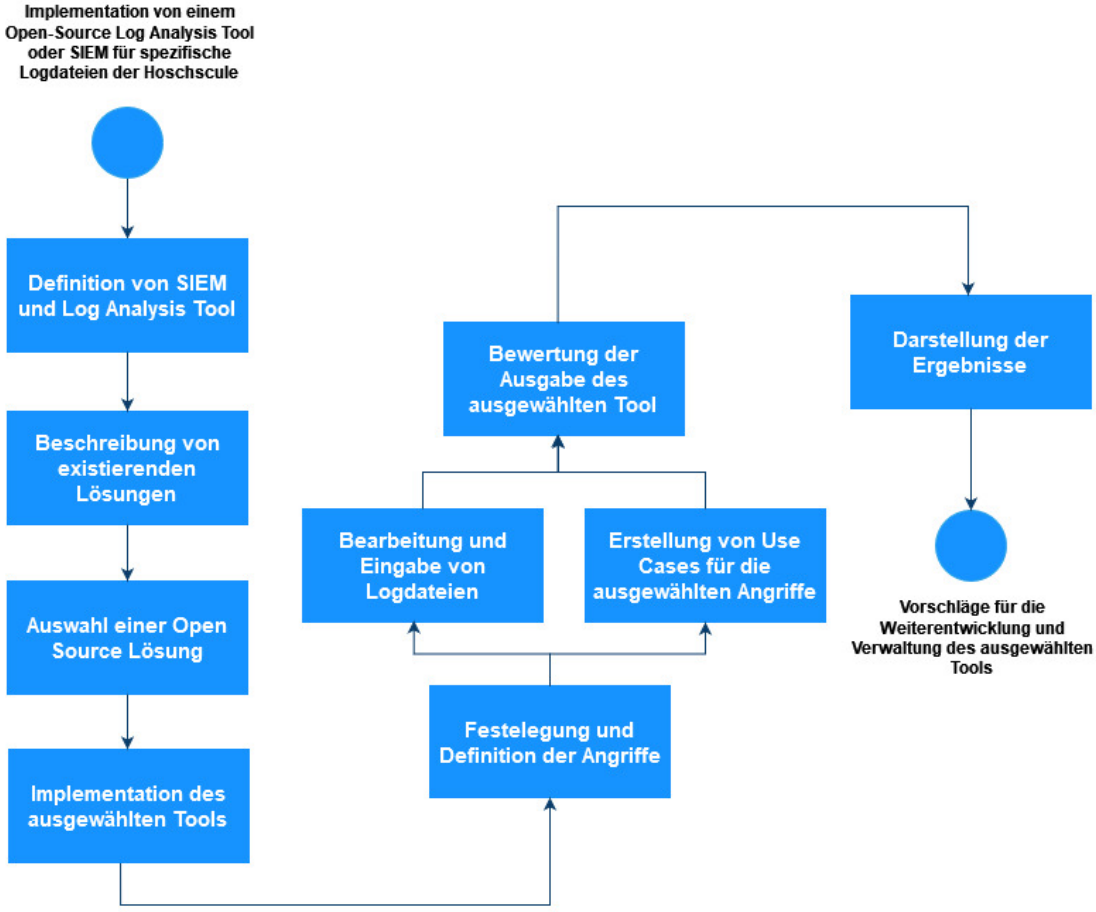


Abbildung 1: Aufbau dieser wissenschaftlichen Recherche
Quelle: Eigene Darstellung

2. Definition von SIEMs und Log-Analyse-Tools

SIEM ist das Ergebnis einer Kombination zwischen dem Security Event Management (SEM) und Security Information Management (SIM) (Dorigo, 2012). Das Erste bezieht sich auf die Identifizierung, Bewertung, Beobachtung und den Bericht von Sicherheitsvorfällen mithilfe von verschiedenen Log Dateien (techopedia, 2015). Das Zweite ist eine Software, die bei der automatischen Sammlung von Loginformationen aus vielen Quellen, wie Firewall und Servern unterstützt (techopedia, 2022). Da die meisten SIEM-Lösungen kostenpflichtig sind, existieren auch viele Open Source Log-Analyse-Tools, die eine ähnliche Aufgabe erledigen, ohne die Kernelementen von SIEM zu besitzen.

Log-Analyse-Tools sind meistens Anwendungen die Logdateien empfangen, speichern, bearbeiten und nach spezifisichen eingegebenen Regeln bewerten. Diese Tools unterstützen Programmierer und Systemadministratoren bei der Überwachung des Zustands eines Systems oder einer Software. Ein solches Tools kann Logdateien von verschiedenen Endpoints und mit verschiedenen Formatierungen bekommen und editieren, so dass es schließlich einen Bericht oder eine Grafik erzeugt (Łukasz Korzeniowski and Goczyła, 2022). Die Nutzung dieser Tools schränkt sich nicht in dem Sicherheitsbereich ein, sondern kann für gesamte Rechenzentren nützlich sein.

In dem Universum des SOC mischen sich verschiedene Begriffe, die manchmal zur Verwirrung führen, weil sie ähnliche Bedeutungen und Verantwortungen haben. Intrusion Detection System (IDS), Intrusion Prevention System (IPS), Security Information and Event Management (SIEM) und Log-Analyse-Tools werden von *nonnative users* und sogar von Spezialisten oft verwechselt, da ihre Aufgaben mehr Gemeinsamkeiten als Unterschiede haben. Um den Umfang dieser Arbeit wegen der zeitlichen Einschränkungen zu verringern, fassen wir kurz die Unterschiede zwischen denen zusammen und legen unsere Grenze auf den SIEMs Lösungen und auf Log-Analyse-Tools fest.

Intrusion Detection System (IDS) sind Software oder Hardware, die Cyberangriffe identifizieren und berichten. Sie haben eine passive Rolle, da sie die Cyberangriffe weder stoppen noch verhindern können. Intrusion Prevention System (IPS)allerdings haben sie eine aktive Haltung gegenüber Cyberangriffe - die sie automatisch behandeln können, indem sie Blocking-Mechanismen einschalten, um den Angriff zu stoppen (Wendzel, 2018). Wie das Intrusion Detection System (IDS), kann das Intrusion Prevention System (IPS) auch Logdateien generieren, die von einer SIEM-Lösung gesammelt werden können. SIEMs können einerseits die Logdateien von diesen und von anderen Endpoints bekommen und diese nach vordefinierten Regeln bewerten, um dem SOC-Team über Sicherheitsvorfälle zu informieren oder automatisch Maßnahmen ergreifen. Wie SIEMs bekommen Log-Analyse-Tools auch Logdateien, um Bericht oder Darstellung zu genieren. Ihre Nutzung ist aber nicht so spezifisch wie die von SIEMs.

Die folgende Abbildung stellt didaktisch eine allgemeine Struktur von SIEM-Lösungen dar:

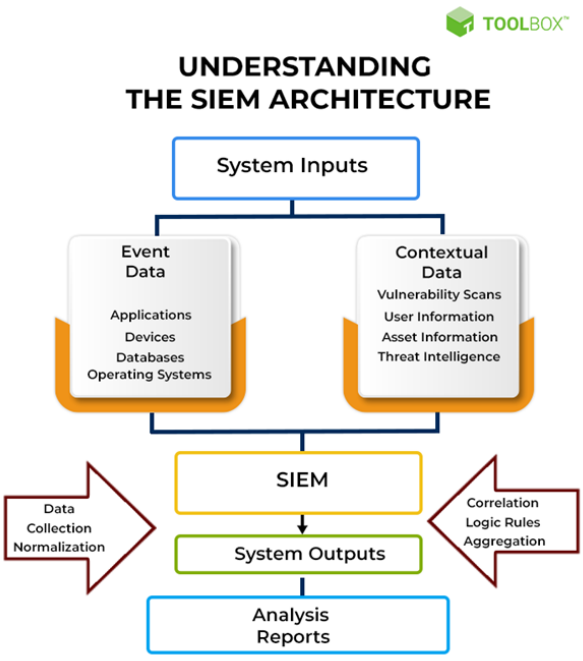


Abbildung 2: Allgemeine Struktur eines SIEM
Quelle: (Mohanan, 2022)

Aus dem Bild können wir feststellen, dass SIEMs für die Zentralisierung von Sicherheitsdaten zuständig ist. Diese werden dann bearbeitet und in einem oder mehreren Berichten dargestellt, damit das SOC-Team schnellere und effektive Entscheidungen treffen können. Der Informationsfluss einer SIEM-Lösung können wieder in der folgenden Abbildung dargestellt werden:

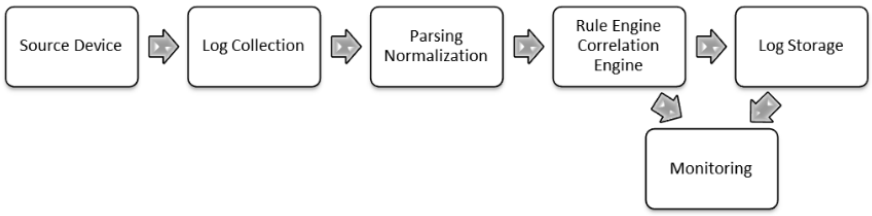


Abbildung 3: Allgemeine Informationsfluss eines SIEM
Quelle: (Granadillo et al., 2021)

Die folgende Abbildung zeigt eine allgemeine Architektur von Log-Analyse-Tools:

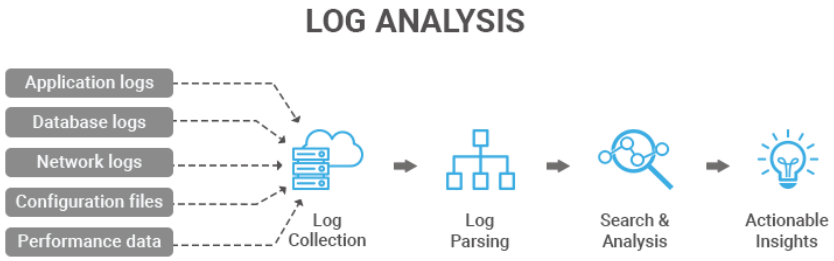


Abbildung 4: Allgemeine Struktur von Log-Analyse-Tools
Quelle: (Tek-Tools, 2020)

Den Informationsfluss eines Log Analyse Tools bildet folgende Grafik ab:

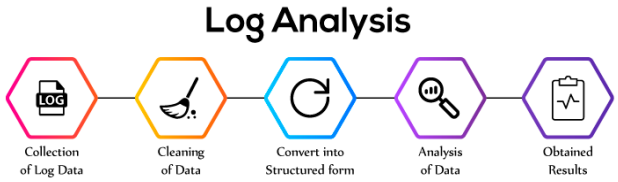


Abbildung 5: Allgemeine Informationsfluss von Log-Analyse-Tools
Quelle: (neptune, 2023)

Aus bisheriger Beschreibung stellen wir fest, dass SIEM viel mehr als eine Sammlung von Logdateien sind. Das Ziel dieser Software ist es die automatische Analyse zu ermöglichen, indem Daten kombiniert und bewertet werden können. In vielen Bereichen, wie Finanzen (Payment Card Industry Data Security Standard (PCDI DSS)), Gesundheitswesen (Health Insurance Portability and Accountability Act (HIPAA)), sind SIEMss eine gesetzliche Verpflichtung (Jog, 2020). In Deutschland verpflichtet das zweite Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme Organisationen mit kritischen Infrastrukturen die Anwendungen solcher Lösungen, um Störungen der Confidentiality, Integrity and Availability (CIA) zu verhindern (BSI, 2021). Log-Analyse-Tools sind seinerseits allgemeine Tools zu der Speicherung, Anpassung, Bewertung und Darstellung von Logdateien, ohne dass sie sich auf die Sicherheitsebenen fokussieren.

2.1. Existierende SIEMs Lösungen und Log-Analyse-Tools

Die existierenden SIEMs und Log-Analyse-Tools können in zwei Kategorien getrennt werden: *Proprietär* und *Open Source*. In folgenden Abschnitten präsentieren wir die proprietäre SIEM Splunk, um einen Maßstab für unsere Auswahl zu definieren, wenn es um Funktionalitäten geht. Wir analysieren folgende SIEMs und Log-Analyse-Tools:

- Prelude
- AlienVault Open Source Security Information Management (OSSIM)
- FortiSIEM
- Elastic Stack
- Grafana

2.1.1. Splunk

Splunk von dem Unternehmen Splunk Technology wurde 2003 in den USA veröffentlicht (Splunk, 2022b). Er gehört weltweit zu der meistverwendeten SIEM-Software und gilt als *State of the art* für andere ähnliche Lösungen (Kazarov et al., 2018). Zu ihren Kunden gehören große Konzerne wie Airbus, Coca-Cola, Intel und die Deutsche Bahn.

Splunk bietet laut seiner Webseite folgende Funktionalitäten an (Splunk, 2015a):

- Skalierbare Datenplattform
- Risk-based Warnmeldung
- Bedrohungserkennung mithilfe von Machine Learning (ML)
- Automatische Aktualisierung von der Bedrohungs- und Schwachstelle-Database
- Unkomplizierte Installation und Anwendung

Die allgemeine Architektur und der Informationsfluss von Splunk unterscheidet sich nicht von der oben dargestellten Struktur 2, Seite 5, und Informationsfluss3, Seite 6. Da es sich hier um eine proprietäre Lösung handelt, lässt sich Splunk mit vielen anderen Funktionalitäten verwalten und erweitern. Die folgende Abbildung zeigt ein zusammenfassendes Diagramm über den Umfang des Informationsflusses von Splunk:

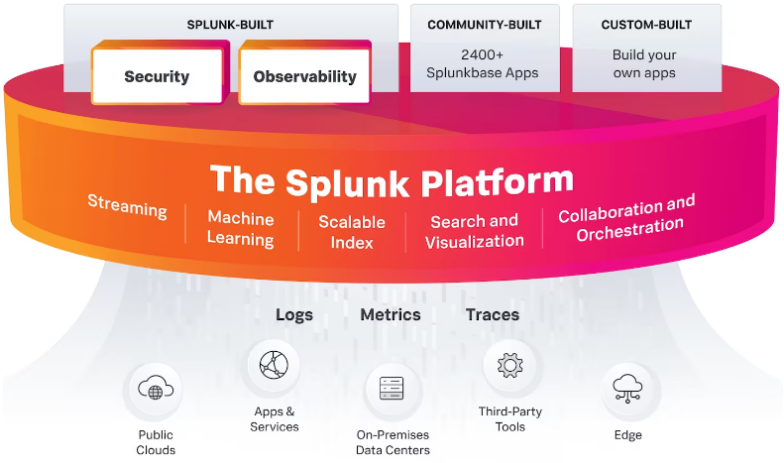


Abbildung 6: Allgemeine Informationsfluss von Splunk
Quelle: (Splunk, 2015b)

In Splunk funktioniert die Bedrohungserkennung mithilfe von Uses Cases. Laut der Dokumentation existieren sie in folgenden Szenarien: Überwachung, Untersuchung und Erkennung. Die Software ist sowohl mit glsmitre Matrix als auch mit Cyber Kill Chain (CKC) für die Gestaltung ihrer Uses Cases integriert (Splunk, 2022a).

In einer spezifischen Arbeit wurden Angriffe auf einem System simuliert und schließlich mit Splunk analysiert, um Gefahren zu identifizieren und diese im Voraus zu sehen (Su et al., 2016). In anderer Arbeit beschrieben die Autoren, wie eine Splunk-Instanz installiert und konfiguriert wurde, um spezifische Brute-Force Angriffe zu erkennen (Selvaganesh et al., 2022).

2.1.2. Prelude

Das im Jahr 2002 in Frankreich von Yoann Vandoorselaere freigegebene Tool Prelude zählt zu einer europäischen Open Source SIEM Lösung. Laut dem Anbieter verfügt Prelude unter anderem folgende Funktionalitäten (Prelude SIEM, 2018):

- Informationszentralisierung
- Datenaggregation und -Zusammenhang mit vordefinierten und von dem Nutzer angepassten Regeln
- Einbruchserkennungsmechanismen
- Datennormalisierung

Die Anwendung besteht aus verschiedenen unabhängigen Modulen. Unter denen highlighten wir Warnmeldung, Archivierung, Analyse und Verwaltung. Das Erste gehört zu der zentralen Aufgabe dieser Lösung - es ist dafür zuständig, Daten zu empfangen, zu normalisieren, Zusammenhänge zu erschließen und Meldungen zu generieren. Das zweite Modul - Archivierung konzentriert sich auf die Speicherung und Verfügbarkeit der Daten. Zu dem Analyse-Modul gehören statistische Aufgaben und Darstellungen in verschiedenen Formaten. Das letzte Modul dient dazu, die Anwendung zu steuern, Nutzer zu erstellen dessen Rechte zu konfigurieren (European Comission, 2015).

Die folgende Abbildung zeigt die Integration verschiedener Module von Prelude und wie sie mit einander kommunizieren, um Analyse, Meldung und Speicherung zu generieren:

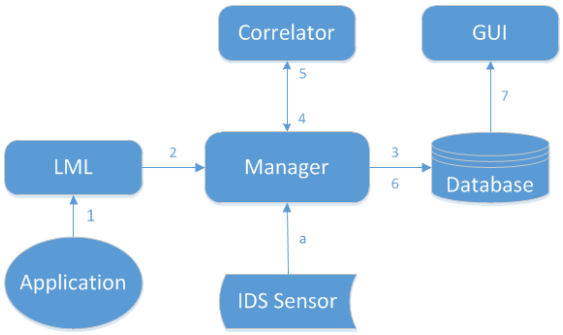


Abbildung 7: Integration zwischen den Modulen von Prelude
Quelle: (Prelude Team, 2007)

Aus der Abbildung und der Dokumentation können wir folgenden Informationsfluss erkennen - die Daten werden von Endanwendung generiert und zum Loganalyzer (Prelude Log Monitoring Lackey (LML)) geschickt, wo sie normalisiert und bewertet werden. Für solche Logs, wo es verdächtige Werte gibt, werden Warnmeldungen generiert. Diese Meldungen werden zum Manager Module weitergeleitet. Der Correlator oben sucht nach einem Zusammenhang zwischen anderen Daten. Das Ergebnis von Correlator wird wieder zum Manager geschickt und danach zu der Datenbank. Schließlich stehen die Berichte in dem User-Interface zur Verfügung (Prelude SIEM, 2020).

Die Architektur der Anwendung ermöglicht sowohl einen zentralisierten als auch einen dezentralisierten Aufbau. In der nächsten Abbildung sehen wir eine einfache Darstellung des Informationsflusses von Prelude:

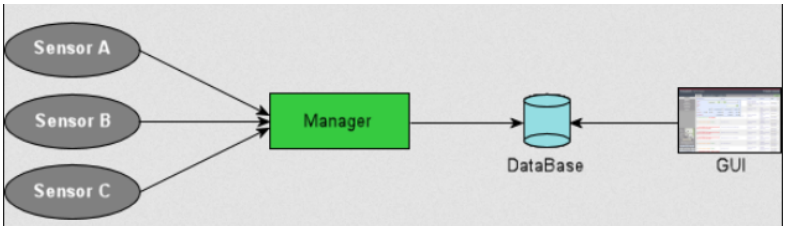


Abbildung 8: Informationsflussin Prelude
Quelle: (Prelude Team, 2007)

In einer dezentralisierten Umgebung werden Daten von verschiedenen und getrennten Quellen generiert und bearbeitet. Schließlich können die Nutzer auf diesen Daten über eine grafische Benutzeroberfläche (GUI) zugreifen.

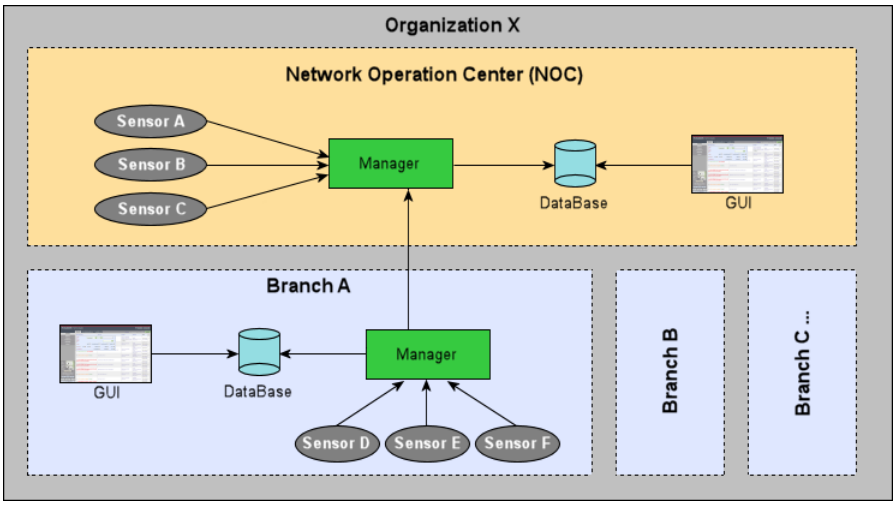


Abbildung 9: Erweiterte Architektur von Prelude mit dezentralisierten Datenquellen Datenverarbeitung
Quelle: (Prelude Team, 2007)

Die wissenschaftliche Literatur über Prelude ist sehr eingeschränkt. Wenige Publikationen fokussieren sich auf die Entwicklung, Implementation und unternehmerische Anwendung dieses Tools. Eine Studie von 2021 versuchte dieses und zwei andere Tools (AlienVault und Cyberoam iView) anhand technischer und nutzerfreundlicher Kriterien zu vergleichen. Von diesn Kriterien highlighten wir folgende(Radoglou-Grammatikis et al., 2021):

- **technische Kriterien**
  - Echtzeige Leistung*Real-time performance*,
  - Umfang und Flexibilität der Meldungen *Range and flexibility of reporting*
  - Zusammenhang von Warnmeldung *Alert correlation*
- **nutzerfreundliche Kriterien**
  - Vollständige Dokumentation *Documentation comprehensiveness*
  - Komplexität der Installation *Complexity of the installation process*

– Komplexität der Einstellung *Complexity of the system configuration*

In den technischen Kriterien lag Prelude auf dem dritten Platz und in den benutzerfreundlichen Kriterien bekam Prelude den ersten.

Auch in den nicht wissenschaftlichen Publikationen existiert eine begrenzte Anzahl von Texten über Preludes. Die existierenden Texte kommentieren ganz zusammenfassend die ausreichende Dokumentation und heben hervor, dass es eher eine in Europa verbreitete Lösung ist.

2.1.3. AlienVault OSSIM

AlienVault OSSIM ist eine im Jahr 2007 entwickelte Open Source SIEM Lösung. Im Jahr 2018 wurden sie von der Firma AT&T Communication gekauft (CBNINSIGHTS, 2020). In der Beschreibung des Anbieters steht, dass er sie auch dabei unterstützt, Daten zu sammeln, zu normalisieren und zu bewerten. Er behauptet auch, dass sein Tool in der Lage ist, Schwachstellen und Angriffe zu erkennen, das Verhältnis zu beobachten und Datenzusammenhänge zu erschließen (AT&T Cybersecurity, 2022).

AlienVault hat eine kostenpflichtige Version, die Alien Vault Unified Security Management (USM) heißt. Auf der Webseite von AT&T steht, dass es keine spezifische Dokumentation für die Open Source Version AlienVault OSSIM gibt, da viele Funktionalitäten von der anderen Version stammen (AT&T Cybersecurity, 2022).

Die folgende Abbildung zeigt das von dem Anbieter freigelegte Architekturdiagramm von der USM Version:

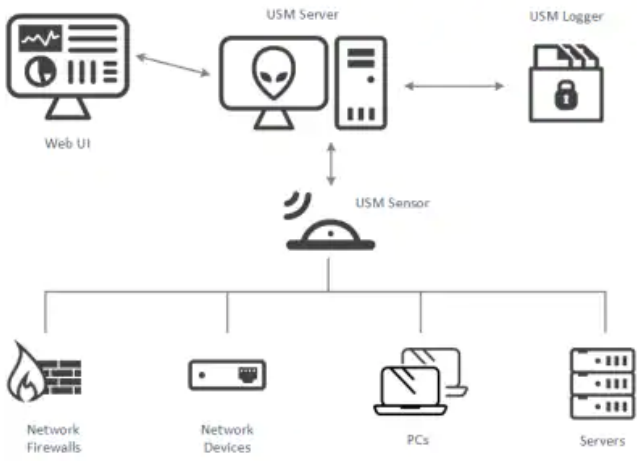


Abbildung 10: Architekturdiagramm von AlienVault USM
Quelle: (AT&T Cybersecurity, 2022)

Laut der Website Comparitech steht AlienVault auf dem 13ten Platz von den bestbewertetsten SIEM-Lösungen. Die Seite beschreibt auch, dass zu dem Tool einen IDS, ein Verhaltensüberwachungssystem und einen Schwachstellen-Scanner integriert sind. Die Anwendung ist auch mit der Plattform Open Threat Exchange(OTX) verbunden - diese ermöglicht eine Teilung von Informationen über die Schwachstelle. Comparitech highlighted, dass die Anwendung wegen ihrer niedrigen Kosten besser für kleine oder mittelständige Unternehmen geeignet ist (comparitech, 2023).

Die Anwendung soll konsistenten Daten Zusammenhang anbieten und soll das Auftauchen von Falsch Positiv vermeiden. AlienVault kommt auch mit vordefinierten Use-Cases, die dabei unterstützen, gewöhnliche Angriffsszenarien zu erkennen. Die Installation, die Einstellung und die Integration mit anderen Tools ist auch benutzerfreundlich (Gómez et al., 2022). Aus einer anderen wissenschaftlichen Quelle fanden wir heraus, dass für viele Quellen eine manuelle Normalisierung der Logdatein notwendig ist (Nabil et al., 2017). Die Anwendung hat aber einen zuverlässigen Berichtsmechanismus.

Während unserer Recherche gab es wenig wissenschaftliche Literatur, die sich um AlienVault OSSIM kümmert. Die meisten Publikationen waren aus kommerziellen Quellen und diese konzentrierten sich auf eine kostenpflichtige SIEM-Lösung von AT&T..

2.1.4. FortiSIEM

FortiSIEM ist eine US-amerikanische SIEM-Lösung von der Firma Fortinet. Fortinet kaufte im Jahr 2016 das Unternehmen AccelOps und dessen SIEM-Lösung und benannte es zum FortSIEM (Fortinet, 2016).

Laut dem Anbieter hat FortiSIEM eine robuste Integration mit anderen Tools und lässt sich leicht und einwandfrei skalieren. Andere Versionen des Tools sind mit Machine Learning (ML) integriert, sodass die Anwendung auch Verhältnisanalysen durchführen kann (Fortinet, 2022). Das Tool bietet auch eine umfangreiche und ausführliche Dokumentation an. Die nächste Abbildung zeigt die skalierbare Architektur des Tools:

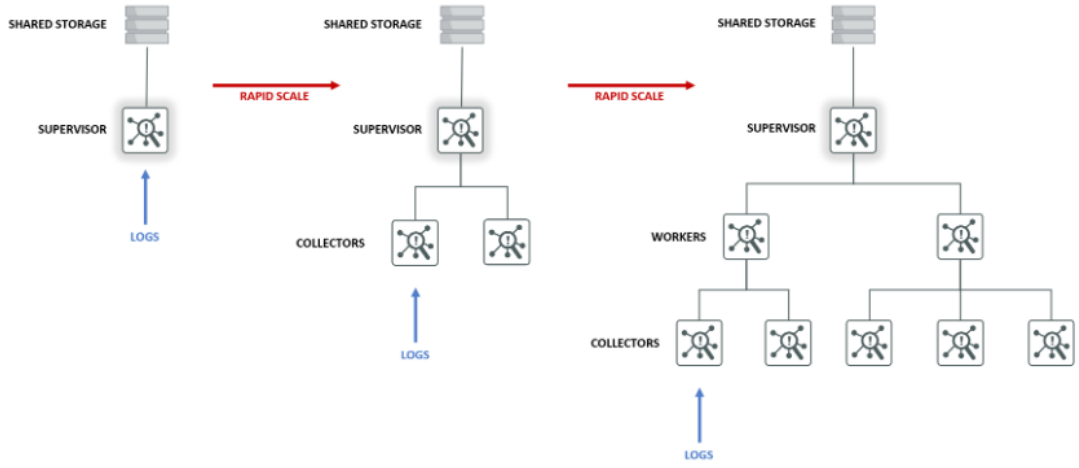


Abbildung 11: Skalierbare Architektur von FortiSIEM
Quelle: (Fortinet, 2020)

Auch zu dieser SIEM-Lösung ist die wissenschaftliche Produktion eingeschränkt. Eine von der gefundenen Publikation betont, dass FortiSIEM eine schnelle Erkennung von Angriffen anbietet und über Network Operations Center (NOC) Funktionalitäten verfügt (Ramírez Tomás, 2018). Wie andere SIEMs Lösungen, hat FortiSIEM folgende Funktionalitäten:

- Datensammlung und Normalisierung
- Daten Zusammenhang
- Generierung von Berichten
- Warnmeldungen
- Datenauswertung

2.1.5. Elastic Stack

Elastic Stack stammt aus der Verbindung von drei Tools: Elasticsearch, Logstash und Kibana. Das Erste ist eine Such-und Analyse-Maschine. Das Zweite ist eine serverseitige Anwendung zur Datenverarbeitung und -Weiterleitung. Schließlich Kibana ist dafür zuständig, visuelle Darstellungen in einem Grafik-Format auszugeben (packt, 2019). Von diesen drei Tools Logstasch ist der einzige Open Source (elastic, 2021). Obwohl die anderen zwei kostenlos verwendet werden können, gehören sie nicht zu der Open Source Kategorie (Open Source Initiative, 2007). Dieses Tool besitzt viele Eigenschaften einer SIEM-Lösung und wird von vielen SOC verwendet, ist aber für viele Experten, kein SIEM für sich, da es über keine Warnmeldungssystem, Daten Zusammenhang und Vorfälleverwaltung verfügt (Miller, 2021). Diese und anderen Funktionalitäten lassen sich aber durch Plugins integrieren.

Das folgende Diagramm stellt die Architektur von Elastic Stack mit ihren integrierten Elementen dar:

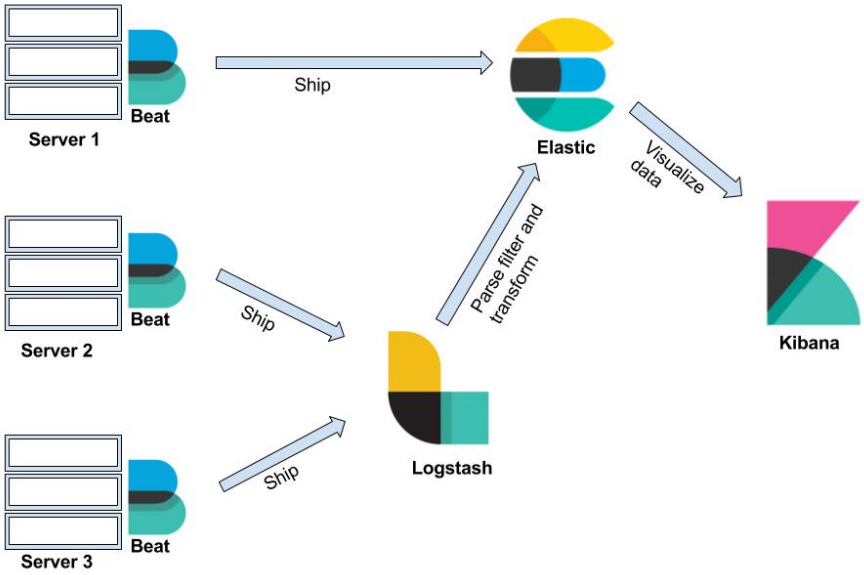


Abbildung 12: Integration zwischen Elasticsearch, Logstash und Kibana
Quelle: (packt, 2019)

Die Beats auf dem Bild sind an der Endpoints installiert und leiten Daten entweder zu Elasticsearch oder zu Logstash weiter, wo sie schließlich bearbeitet werden (Jain, 2018).

Ein Teil der wissenschaftlichen Literatur zeigt die Log Analyse-Funktionalitäten von Elastic Stack und die Unterstützung bei Normalisierung und Indexierung von Daten für eine lesbare Ausgabe (Advani et al., 2020). Die starke Skalierbarkeit wurde auch bei einer Studie erwähnt, wo Elastic Stack für Wi-Fi Logging eingesetzt wurde (Wang et al., 2019).

Die offizielle Dokumentation von Elastic Stack betont, dass die Anwendung folgende Funktionalitäten besitzt (elastic, 2022):

- Datensuche, -Normalisierung, -Analyse und
- Speicherung
- visuelle Ausgabe

Folgendes Diagramm aus der offiziellen Dokumentation zeigt die Aufteilung der Funktionalitäten pro Element von Elastic Stack:

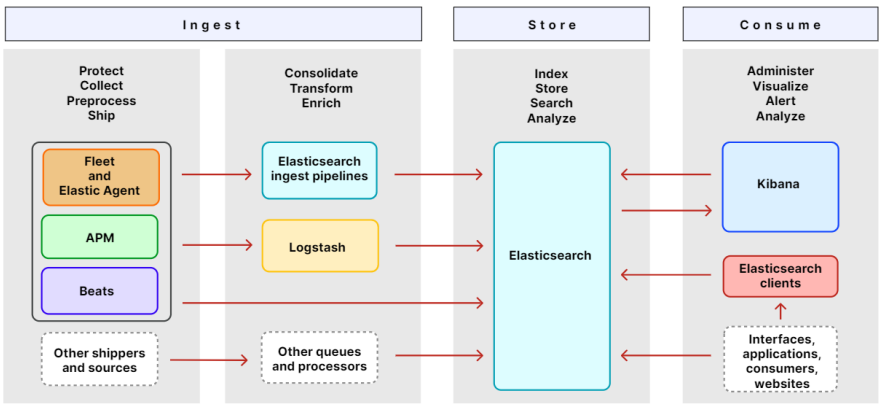


Abbildung 13: Aufteilung der Funktionalitäten zwischen den Komponenten
Quelle: (elastic, 2022)

Die wissenschaftliche Publikation über Elastic Stack ist vielfältiger als bei anderen recherchierten Tools. Es ist aber wichtig zu betonen, dass die Mehrheit von denen sich eher mit dem Logging als mit den SIEM-Eigenschaften der Anwendung beschäftigt.

2.1.6. Grafana

Von allen recherchierten Lösungen ist Grafana die Einzige, die nicht als SIEM dargestellt ist. Grafana wird aber als Plattform für Visualisierung von Daten beschrieben. Mit dem Tool ist es möglich eine Graphik zu erstellen und Meldungen zu definieren. Das Ziel der Anwendung ist, Information in einer einfachen und verständlichen Art und Weise zur Verfügung zu stellen (redhat, 2022).

Im Jahr 2014 wurde Grafana von der Firma Grafana Labs veröffentlicht. Das Tool basiert auf Kibana3,2.1.5. Ursprünglich sollte Grafana ein einfacheres Bearbeitungstool für Grafiken sein und ermöglichen, Datenanfragen unkomplizierter zu machen. Die neuste Version, 9.4.3. wurde im März 2023 veröffentlicht und bietet viele Funktionalitäten an. Es ist auch möglich das Tool mithilfe von Plugins zu erweitern (Ödegaard, 2019)..

In der Webseite betont der Anbieter, dass Grafana die Zentralisierung und Zugang von Daten vereinfachen. Alle Art von Daten lassen sich analysieren und darstellen, von der Leistung von Anwendungen bis Verkaufsdaten und Krankheitsfällen. Die Anwendung soll auch den Zusammenhang von Daten ermöglichen, um wichtige Informationen herauszunehmen (Grafana Labs, 2016).

Grafana ist auch mit dem Logging Tool Loki und Promptail integriert. Promtail ist für Sammlungen der Logdateien und Weiterleitung an Loki zuständig. Promptail wird an jeden Endpoint installiert. In Loki werden diese Logdateien ohne Index für den schnellen Zugriff gespeichert. Diese Daten können dann in Grafana mithilfe der Abfragesprache LogQL aufgerufen werden. Schließlich können Warnmeldungen mit spezfisichen Regeln generiert werden, die in Loki eingeführt werden (Grafana Labs, 2018). Auf dem Folgenden Bild wird die Struktur von Grafana Loki dargestellt:

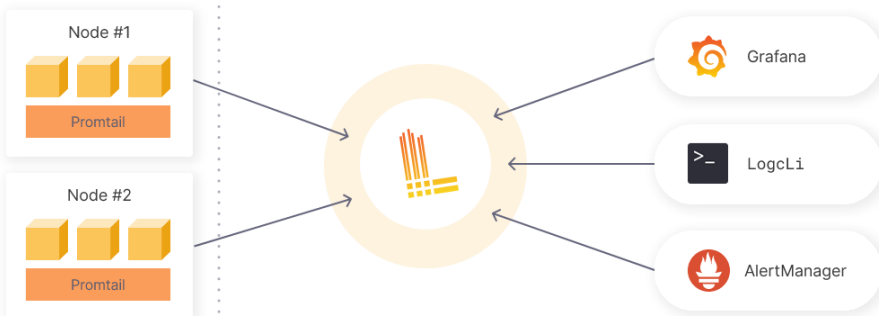


Abbildung 14: Integration von Log-Quellen mit Promptail (links), Loki (mitte) und Grafana (rechts)
Quelle: (Grafana Labs, 2022a)

Das Tool hat auch eine umfangreiche Dokumentation, die ausführlich erklärt, wie es zu installieren, bedienen und mit anderen Tools integrierbar ist.

Die wissenschaftlichen Literatur über Grafana konzentriert sich eher auf die Anwendung des Tools für die grafische Darstellung von Daten als für ihre Nutzung in dem Sicherheitsbereich. Eine Recherche, z.B., wollte das Ergebnis von der Überwachung von Cloud-basierten Systemen, von Netzwerkaktivitäten und von Netzwerkverkehr mithilfe

von Grafana darstellen (Manases and Zinca, 2022). In dieser Hinsicht gibt es wenige wissenschaftliche Arbeit, wo die Implementation und Integration von Grafana mit anderen Tools für den Sicherheitsbereich die Hauptfigur ist.

2.2. Auswahlkriterien

Eine umfangreiche SIEM Software die viele automatische Lösungen für die Erkennung und Bekämpfung von Cyberangriffen würde perfekt für jede Situation passen. Da solche Lösungen meistens (oder alle) Proprietär sind und nur für teure Preise angeboten werden, entschieden wir uns für die Anpassung eines Open SourceTools, das zu unserem Kontext und Einschränkungen gehört.

Demnächst beschäftigen wir uns mit Grafana. Wir beschreiben, wie wir das Tool installieren, konfigurieren und mit verschiedenen Logdateien eingeben. Nachdem die Grundfunktionalitäten eingerichtet sind und einwandfrei funktionieren, generieren wir anhand der Mitre ATT&CK Matrix Uses Cases für die zukünftigen ausgewählten Angriffe. Unser Ziel ist Grafana so einzustellen, dass es in der Lage ist, die Muster dieser Angriffe zu erkennen und darüber zu berichten.

3. Implementierung

In diesem Kapitel geht es um die Implementierung und den Aufbau von Grafana, um Cyberangriff mithilfe der Mitre ATT&CK Matrix zu erkennen. Das Arbeitslabor wird mit Container und virtuellen Maschine (VM) aufgebaut, wie im Diagramm in der folgenden Abbildung dargestellt.

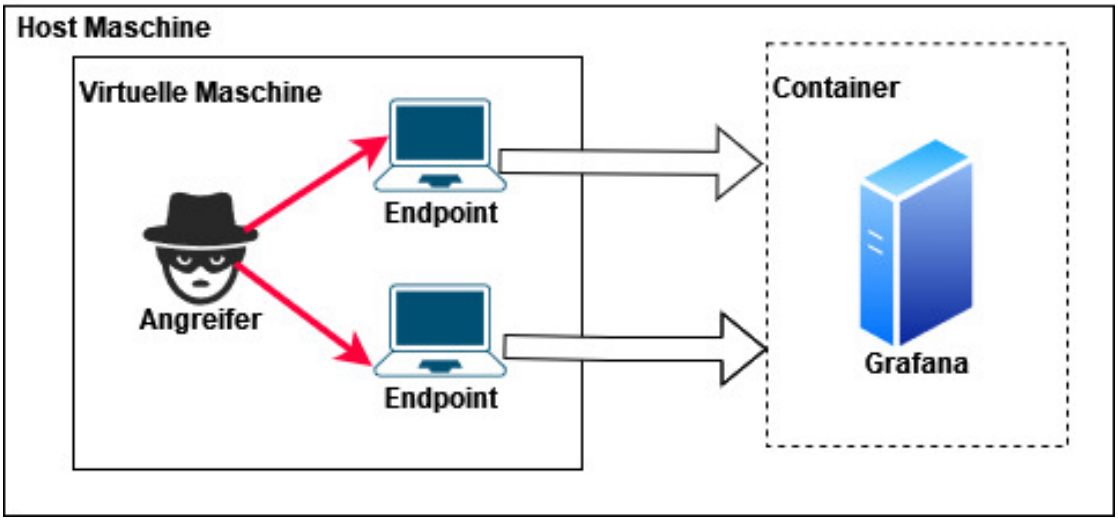


Abbildung 15: Aufbau unseres Arbeitslabors
Quelle: Eigene Quelle

Von unserem Aufbau aus streben wir folgende Ziele an: die Aufnahme und Anpassung von Logdateien für Grafana, die Mustererkennung für ausgewählte Cyberangriffe und schließlich die Erstellung von Warnmeldungen für die Endnutzer, damit sie geeignete Sicherheitsmaßnahmen ergreifen können.

Der gezielte Ablauf ist in dem folgenden Diagramm dargestellt:

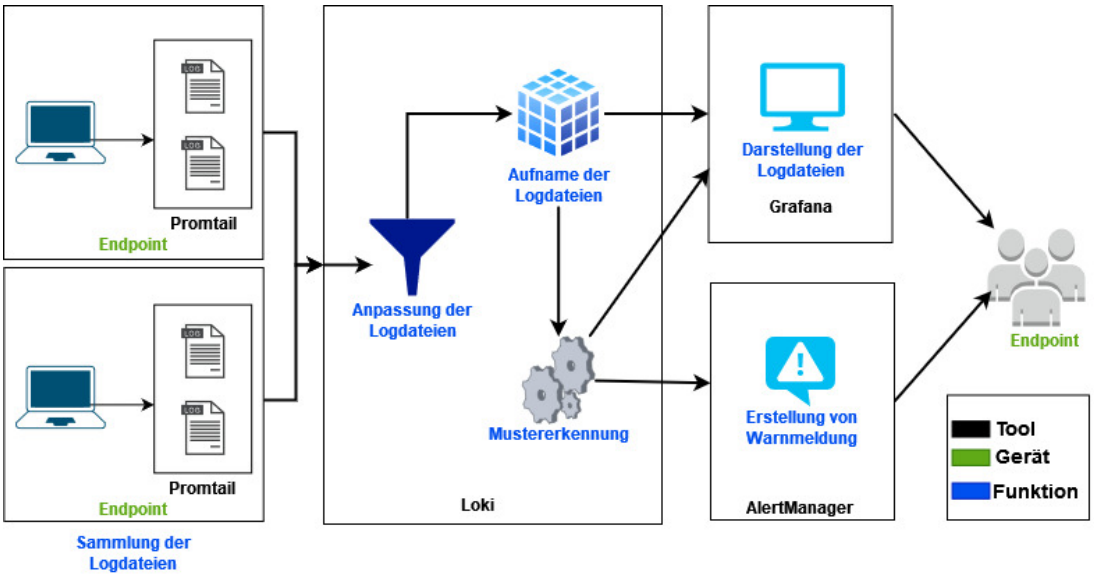


Abbildung 16: Erwarteter Ablauf der Sammlung der Logdateien bis zur Warnmeldung
Quelle: Eigene Quelle und (Grafana Labs, 2018)

3.1. Angriffserkennung anhand der Mitre ATT&CK Matrix

Es gibt verschiedene Methoden und Frameworks zur Vermeidung, Erkennung und Unterbrechung von Cyberangriffe. Zu den Beispielen gehören das Open Web Application Security Project (OWASP), das Cyber Kill Chain (CKC)und die Mitre ATT&CK Matrix, die von SOC-Teams verwendet werden, um die Sicherheit von Systemen und/oder Netzwerken zu gewährleisten. Da sich die Richtlinien und Schwerpunkte dieser Frameworks unterscheiden können und somit unterschiedliche Anforderungen an den Aufbau unserer Struktur stellen könnten, haben wir uns entschieden, die Mitre ATT&CK Matrix zur Erkennung von Cyberangriffe anzuwenden, insbesondere da dieses Framework auch in Splunk integriert ist. Die Mitre ATT&CK Matrix ist auf Taktiken, Techniken, Prozeduren (TTP) basiert. Angriffe, Gegenmaßnahmen und Erkennung werden nach TTP definiert.

Die Mitre ATT&CK Matrix hat folgende Hauptnutzung (MITRE ATT&CK, 2018b):

- Erkennung und Analyse von Angriffstechnik
- strukturierte Datensammlung über Bedrohungen
- Emulieren von Cyberangriffe für die Anwendung an Angriffsübungen
- Systemhärtung und Verbesserung der Verteidigungsmaßnahmen

Die Matrix bietet Unternehmen und SOC-Teams umfassende Möglichkeiten, um ihre wertvollen Ressourcen zu schützen und ihr Fachwissen im Bereich der Cybersicherheit zu erweitern (Hazel, 2021). In dieser Arbeit konzentrieren wir uns auf die Entwicklung und Implementierung einer Methode zur automatischen Erkennung und Analyse von Angriffstechniken in Grafana.

Das Mitre ATT&CK Framework besteht aus 14 Taktiken, zu denen jeweils Techniken gehören, die wiederum in Sub-Techniken unterteilt sind. Jede Sub-Technik wird mit Beispielen, Härtungsmaßnahmen und Erkennungsregeln beschrieben.

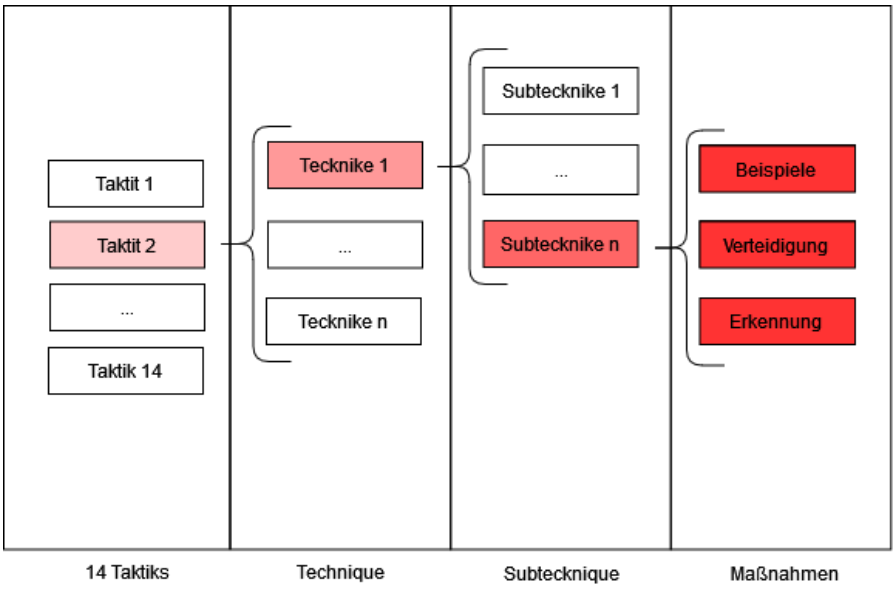


Abbildung 17: Struktur der Mitre ATT&CK Matrix
Quelle: Eigene Quelle und (MITRE ATT&CK, 2018b)

Die 14 Taktiks sind folgende:

- Informationssammlung für zukünftige Angriffe

- Entwicklung von Ressource von Angreifer
- Erster Zugang zum Opfersysteme
- Ausführung von bösartigen Coden
- Beharrlichkeit von System
- Privilegienausweitung
- Vermeidung von Verteidigungssysteme
- **Zugang zu Anmeldedaten**
- Umgebungserkennung
- Seitliche Bewegung zu anderem Systemen innerhalb des Angriffsziels
- interne Informationssammlung
- Steuerung und Kontrolle (C2 - Command and Control im Original)
- Datenextrahierung
- Auswirkung auf die Integrität

3.2. Auswahl des Angriffes

In dieser Arbeit beschäftigen wir uns mit der Taktik „Zugang zu Anmeldedaten“ und deren Technik Brute-Force Angriffe. Diese Technik ist in vier Untertechniken aufteilt:

In dieser Arbeit beschäftigen wir uns mit der Taktik „Zugang zu Anmeldedaten“ und ihrer Technik „Brute-Force Angriffe“. Diese Technik ist in vier Untertechniken unterteilt:

- Erraten von Anmeldedaten
- Entschlüsselung von Hashwerte
- *Password Stuffing*
- *Password Spraying*

Da unser Ziel hier ist, Grafana zu verwenden, um Angriffe zu erkennen, haben wir uns für einen einfachen und reproduzierbaren Angriff entschieden, der wenige Ressourcen erfordert. In diesem Fall kann ein Brute-Force Angriffe mit zwei VMs problemlos durchgeführt werden. Für diesen Angriff verwenden wir die Sub-Technik Ërraten von Anmeldedaten und *Password Stuffing*, da sie ähnliche Erkennungsmethoden aufweisen. Andere Maßnahmen schließen wir hierbei aus.

Die nächste Abbildung zeigt den Umfang unseres Implementationsversuchs mithifel von Mitre ATT&CK:

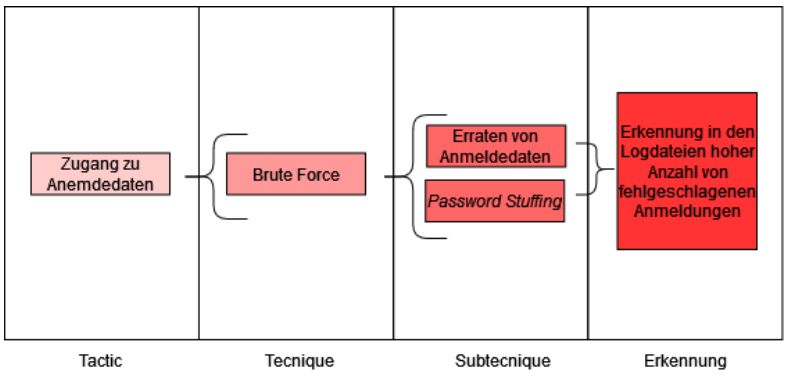


Abbildung 18: Taktiken, Techniken, Prozeduren (TTP) für unseren Angriff
Quelle: Eigene Quelle und (MITRE ATT&CK, 2020)

3.3. Installation und Erstellung von Logdateien

In diesem Abschnitt konzentrieren wir uns auf die folgenden Punkte:

1. Einrichtung von VMs für das Opfersystem und den Angreifer
2. Simulation des Angriffe zur Erzeugung von Logdateien
3. Installation und Konfiguration von Grafana Loki und Promtail mit Container
4. Weiterleitung der Logdateien an Grafana

Die Installation und Verwendung können entweder über die grafische Benutzeroberfläche (GUI) oder über die Kommandozeile durchgeführt werden. In dieser Arbeit verwenden wir die Kommandozeile.

3.3.1. Einrichtung der VMs für Opfersystem und Angreifen

Die beiden VMs sind eine vorgebaute „Kali virtuellen Maschine (VM)“ und „Ubuntu Server 22.04.2“ in ihren standardmäßigen Einstellungen. Beide Maschinen wurden entsprechend ihrer jeweiligen Dokumentation installiert (Kali, 2019) und (Ubuntu, 2023a).

Für das Opfersystem haben wir uns für die Passwörter „qwertz“ und „password“ entschieden. Laut einer Umfrage gehören diese Passwörter zu den zehn am häufigsten verwendeten Passwörtern in Deutschland (silicon.de, 2022).

Für die Durchführung des Password Spraying haben wir folgende Passwortkombinationen erstellt:

Opfersystem 1	Opfersystem 2
admin:123456	bob:hallo
user1:passwort	master:alice
user2:abc123	hans:daniel
user3:qwertyuiop	bruno:super123

3.3.2. Generierung von Logdateien mit der Angrifsssimulation

Für den Angriff verwenden wir folgende Tools:

- Secure Shell Protocol (SSH)
- Hydra

In diesem Szenario sendet Hydra gleichzeitig mehrere Authentifizierungsversuche an das Opfersystem, um eine SSH-Verbindung herzustellen. Das Tool verwendet ein sogenanntes Wörterbuch mit verschiedenen Einträgen, die als Passwörter dienen. Für unseren Test benutzen wir die bekannte Rockyou-Datei.

Die folgende Abbildung zeigt, wie das Password Stuffing abläuft:

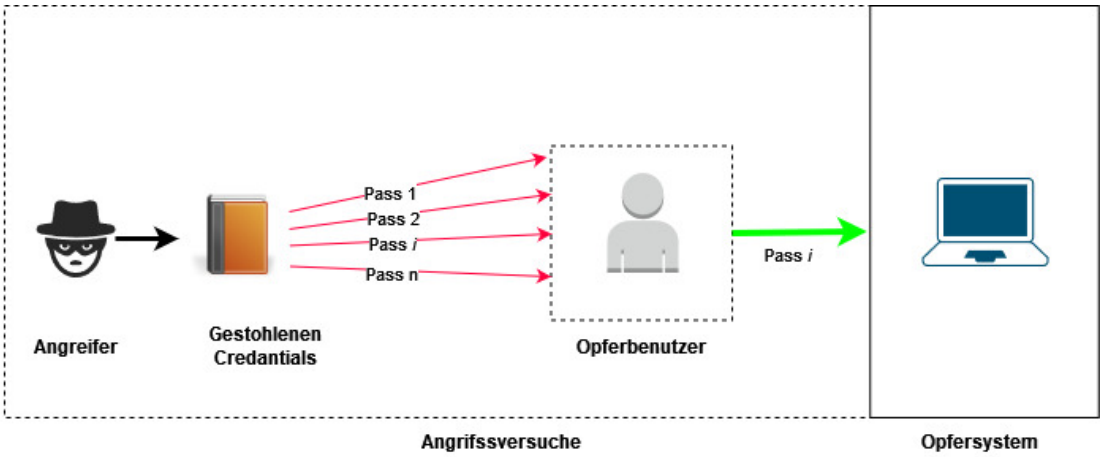


Abbildung 19: Darstellung von *Password Stuffing*
Quelle: Eigene Quelle und (Ba et al., 2021)

Password Stuffing wurde mit folgendem Kommando durchgeführt (Kali, 2022a):

```
hydra -l [Benutzername] -P rockyou.txt [Opfersystem] ssh -V -t 4

# Erklärung
-l: Spezifikation des Benutzernamens, den wir angreifen wollen
-P: Auswahl der Datei mit bekannten Passwörtern
ssh: Auswahl der Anwendung, die wir angreifen wollen
-V: Ausführliche Ausgabe über Versuche, Fehler und Erfolg
-t 4: Anzahl von gleichzeitigen Verbindungen
```

Das folgende Bild zeigt einen Teil der Ausgabe von Hydra während der Ausführung von Password Stuffing gegen das Opfersystem1:

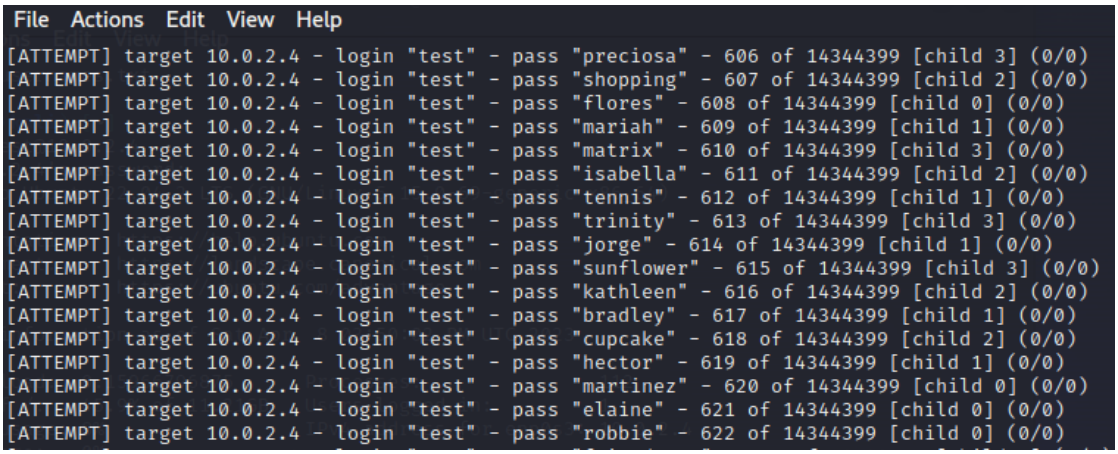


Abbildung 20: Ausführung von *Password Stuffing* gegen Opfersystem1
Quelle: Eigene Quelle und (Ba et al., 2021)

Und gegen Opfersystem2:

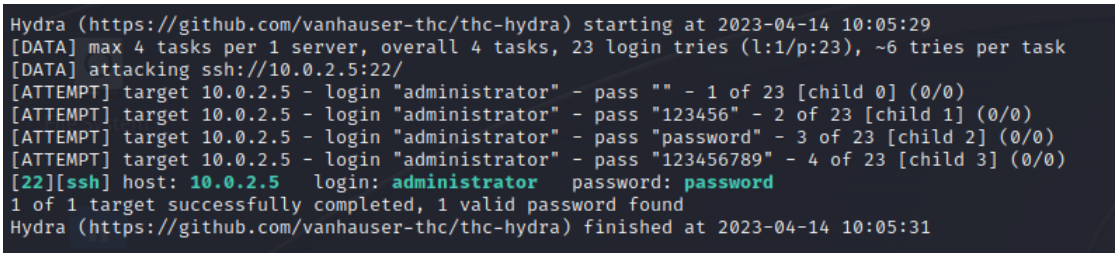


Abbildung 21: Ausführung von *Password Stuffing* gegen Opfersystem2
Quelle: Eigene Quelle und (Ba et al., 2021)

Unser nächster Angriff, Password Spraying, sieht wie folgende aus:

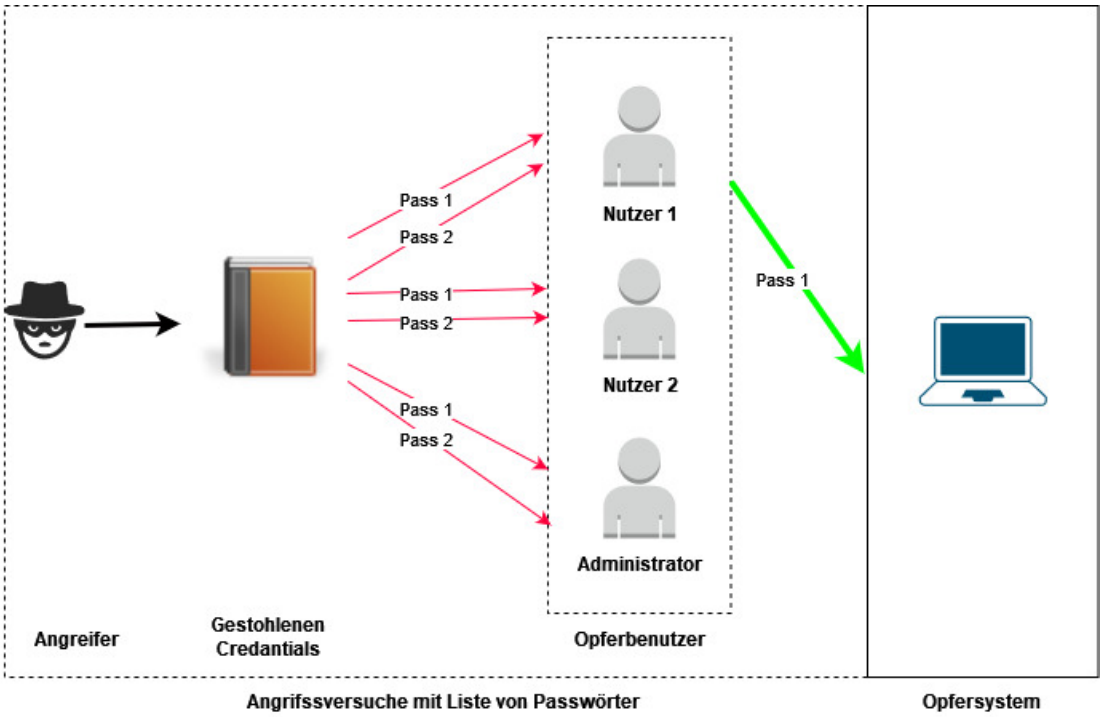


Abbildung 22: Darstellung von *Password Spraying*
Quelle: Eigene Quelle und (Swathi, 2022)

Für diesen Angriff benutzen wir folgendes Kommando:

```
hydra -L username2.txt -P passwoerter.txt [Opfersystem2] ssh -V -t 4

# Erklärung
-L: Auswahl der Datei mit gefunden Benutzernamen
```

In diesem Fall gehen wir davon aus, dass der Angreifer einige oder alle Benutzernamen bereits kennt. Da bei diesem Angriff weniger Anmeldeversuche pro Nutzer durchgeführt werden, verwenden wir eine selbst erstellte Datei mit weniger Passwörtern als die Rocky-ou-Datei. Unsere Datei enthält die am häufigsten verwendeten Passwörter in Deutschland (silicon.de, 2022).

Die folgenden Screenshots zeigen die Ausführung von Password Spraying:

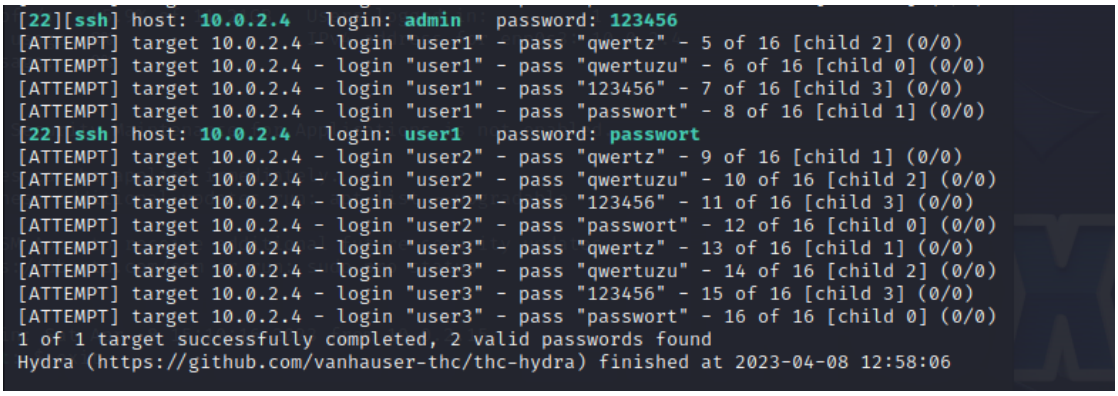


Abbildung 23: Ausführung *Password Spraying* in Kali Linux gegen Opfersystem1
Quelle: Eigene Quelle

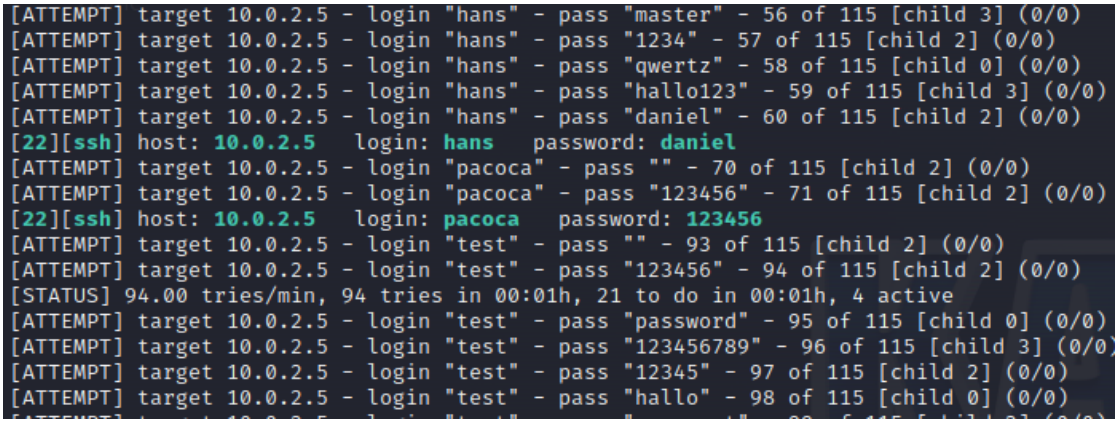


Abbildung 24: Ausführung *Password Spraying* in Kali Linux gegen Opfersystem2
Quelle: Eigene Quelle

3.3.3. Installation und Einrichtung von Grafana Loki und Promtail

Die offizielle Dokumentation von Grafana war nicht immer eindeutig in Bezug auf die Ausführung, daher haben wir auch auf externe Quellen zurückgegriffen, um die Einstellungen an unsere Umgebung anzupassen (Polinowski, 2019). Unter befinden sich die von Grafana zur Verfügung gestellten Konfigurationsdateien und Installationsverfahren (Grafana Labs, 2020a):

```
wget https://raw.githubusercontent.com/grafana/loki/v2.8.0/cmd/
loki/loki-local-config.yaml -O loki-config.yaml
(die Datei wurde angepasst)

wget https://raw.githubusercontent.com/grafana/loki/v2.8.0/
clients/cmd/promtail/promtail-docker-config.yaml
-O promtail-config.yaml (die Datei wurde angepasst)

docker-compose -f docker-compose.yaml up
```

Im Anhang befinden sich die originalen (Siehe Anhang A) und die angepassten Dateien (Siehe Anhang B).

Die obigen Kommandos haben folgende Bedeutungen:

1. Herunterladen der Konfigurationsdatei von Loki
2. Herunterladen der Konfigurationsdatei von Promptail
3. Ausführung von den Containers, indem beide Konfigurationsdateien in eine eingepackt und angepasst wurden und schließlich von der Container-Anwendung gelesen werden

Für spezifische Versionen oder weitere Einstellungen bietet die Dokumentation umfangreiche Möglichkeiten an (Grafana Labs, 2020a).

Für diesen ersten Test wurden die Logdateien des Opfersystems manuell auf den Container übertragen.

Nach der Ausführung des Kommandos ist die Anwendung schon benutzbar, wie in dem folgenden Screenshot:

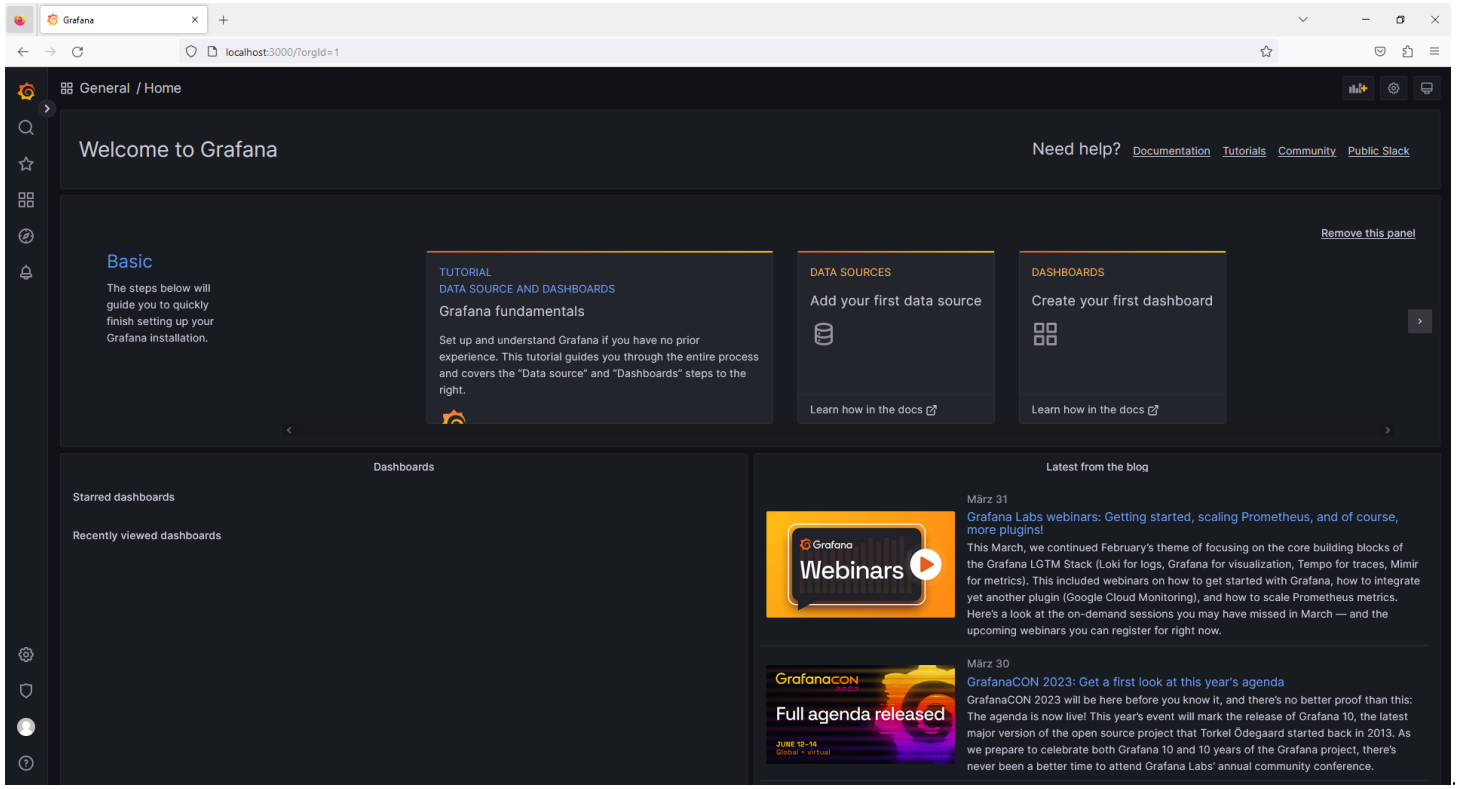


Abbildung 25: Screenshot der Willkommensseite von Grafana Loki
Quelle: Eigene Quelle und (Grafana Labs, 2022a)

3.3.4. Weiterleitung der Logdateien zu Grafana

Grafana Loki bietet mehrere Möglichkeiten, Logdateien zu empfangen. In unserer Arbeit verwenden wir **Promtail**, der in einem Container läuft. Diese Instanz sendet die von uns ausgewählten Logdateien an Grafana und bearbeitet alle Dateien innerhalb eines sogenannten „jobs“. Wenn wir verschiedene Arten von Logdateien hätten, würde jeder Typ einem eigenen „job“ zugewiesen (Grafana Labs, 2021b). Jeder „job“ hat seine eigenen Regeln, um nach den gewünschten Informationen zu suchen.

In einer produktiven Umgebung wäre die Installation von **Grafana Agents** auf jedem Endpoint eine andere Lösung, um Grafana Loki mit Logdateien zu füllen. In diesem Fall würde jeder Endpoint mithilfe von Promtail die Dateien weiterleiten (Grafana Labs, 2022b). Wie bei unserer Lösung müsste der Nutzer für jeden Typ von Logdateien einen spezifischen „job“ konfigurieren.

Der Inhalt von Logdateien lässt sich auch mithilfe der **Application Programming Interface (API)** an Grafana Loki senden. In dieser Situation sendet der Endpoint eine HTTP POST-Anfrage an den Endpunkt von Grafana Loki mit dem Inhalt der Logdateien (Grafana Labs, 2020b):

```
# Endpoint
POST [Addresse_von_Grafana_Loki_Instanz]/loki/api/v1/push

# Inhalt
{
  "streams": [
    {
      "stream": {
        "label": "value"
      },
      "values": [
          [ "Zeit in Unixformat", "<Inhalt der Logdateie>" ],
      ]
    }
  ]
}
```

Grafana Loki bietet auch eine Integration mit dem Open-Source-Tool OpenTelemetry an, um Logdateien zu empfangen (Grafana Labs, 2022c). Im Allgemeinen wird OpenTelemetry verwendet, um Daten zu senden, zu verarbeiten und zu empfangen. Laut dem Anbieter ist OpenTelemetry mit verschiedenen anderen Tools integriert, um die Datenübertragung zu ermöglichen. Das Tool besteht aus *Agents* und *Collectors*. Der Agent wird auf jedem Endpunkt installiert, um Daten zu sammeln und der Collector empfängt die Daten und leitet sie weiter (Grafana Labs, 2022c). Die Integration mit Grafana Loki erfolgt über die Nutzung von APIs. Der Collector läuft in derselben Umgebung wie Grafana Loki, damit er die Logdateien empfangen und verarbeiten kann. Die *Agents* laufen auf jedem Endpunkt und kommunizieren mit dem *Collector*. Die folgende Abbildung soll diesen Vorgang besser darstellen:

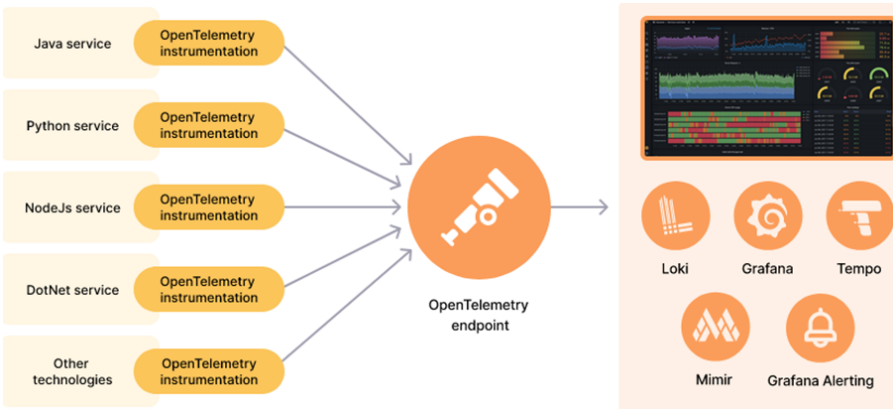


Abbildung 26: Datenfluss zwischen OpenTelemetry und Grafana Loki
Quelle: (Grafana Labs, 2021d)

An der linken Seite haben wir die verschiedenen Endpoints, auf denen jeweils ein *Agent* läuft. In der Mitte haben wir den *Collector*, der die Logdateien schließlich an Grafana Loki und/oder an andere Tools weiterleitet.

3.4. Aufbau der Erkennungsregel für den ausgewählten Angriff

Ein Brute-Force Angriffe lässt sich durch die Anzahl der fehlgeschlagenen Anmeldeversuche erkennen (Selvaganesh et al., 2022). Wir betrachten eine Situation, in der keine Gegenmaßnahmen wie Kontosperre nach n beliebigen Versuchen oder MFA, implementiert sind. Das folgende Aktivitätsdiagramm stellt einen allgemeinen Ablauf eines Anmeldungsverfahrens dar

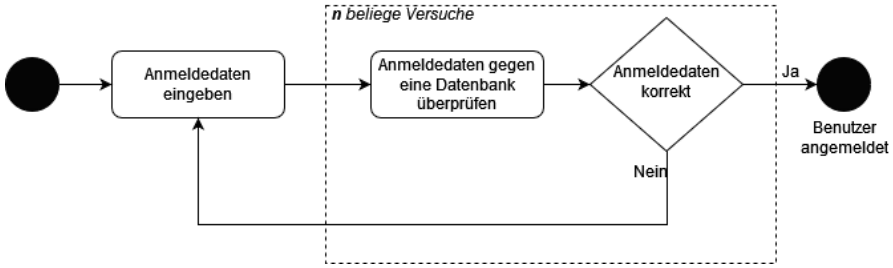


Abbildung 27: Allgemeiner Ablauf eines Anmeldungsverfahrens
Quelle: Eigene Quelle und (Selvaganesh et al., 2022)

Grafana Loki bietet ein Konfigurationsmuster für die Eingabe und Darstellung von SSH Logdateien an. In dieser Konfiguration sind bereits Grafiken und Regelsets enthalten, die eine umfassende Analyse dieser Daten ermöglichen (VoidQuark, 2022). Die extrahierten Logdateien werden mithilfe der folgenden Elemente gelesen und bearbeitet:

Element	Beschreibung
json	Lesbare Dateiformat, deren Daten nach dem Regel *Schlüssel:Wert* gespeichert sind
Muster	Lesen und Extraktion der Information der Logdateien
Regex	Mustererkennung aus der Logdatei
Logfmt	Extraktion von Schlüssel:Wert Paar der Logdateien

Tabelle 1: Elementen eines Regelsätzes in Grafana Loki
Quelle: Eigene Quelle, (VoidQuark, 2022) und (Setter, 2015)

Für jedes Angriffszenario benutzen wir spezifische Regeln, die mit LogQL aufgebaut sind.

Die Filterung findet mithilfe von zweil Labels „Instance“ und „Job“ statt. In Promtail wird jeder Endpoint als „Instance“ bezeichnet. Eine oder mehrere „Instances“ werden einem „Job“ zugewiesen. „Jobs“ beziehen sich auf die Bearbeitung der Logdateien nach dem spezifizieren Regeln, in unserem Fall, Überprüfung von SSH-Logdateien. Diese Struktur stammt aus dem Tool Prometheus. Alle unsere „Instance“ werden in einem „Job“ eingepackt, wo sie nach den gleichen Regeln verarbeitet. Das folgende Diagramm stellt die Beziehung zwischen dieser beiden Labels dar:

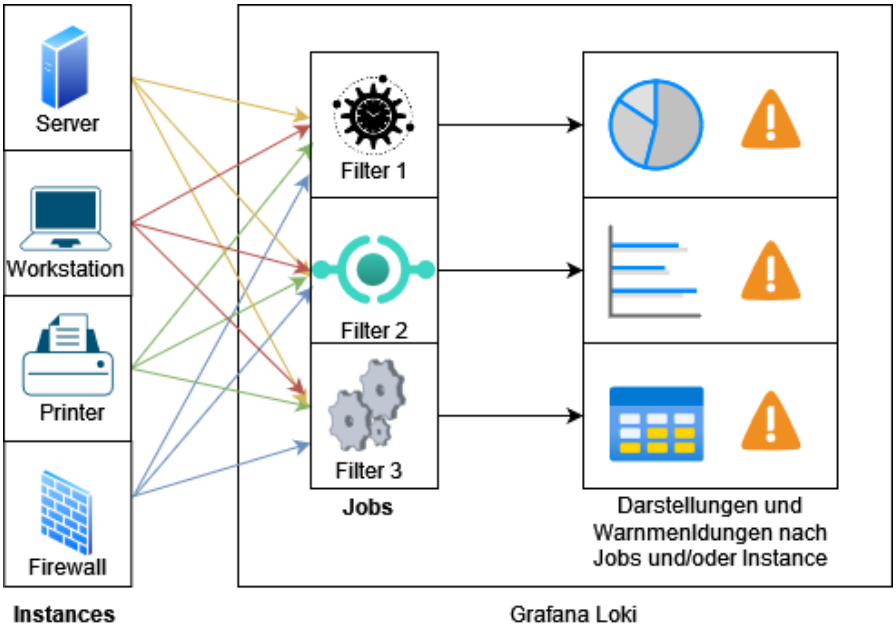


Abbildung 28: Beziehung zwischen „Instance“ und „Job“
Quelle: Eigene Quelle und (Prometheus, 2015)

In dem nächsten Abschnitt beschreiben wir, wie diese Regel in LogQL geschrieben werden.

3.4.1. Regelsätze in LogQL

In diesem Abschnitt fassen wir zusammen, wie eine Abfrage in LogQL für eine Logdatei mit SSH Einträge aussieht. Für ausführliche Informationen über den Aufbau der Abfrage empfehlen wir die offizielle Dokumentation, auf die diese Erklärung basiert ist (Grafana Labs, 2021c). Unsere Logdatei enthält unter anderem folgende Zeile:

```
14 14:05:30 opfersystem2 sshd[1698]: Failed password for administrator
from 10.0.2.15 port 58036 ssh2
```

Um SSH-Einträge zu erkennen und bestimmte Informationen zu extrahieren, die anzeigen, ob es sich um einen fehlgeschlagenen Anmeldeversuch handelt, möchten wir die hervorgehobenen Elemente extrahierenund später hochzählen:

```
14 14:05:30 opfersystem2 sshd[1698]: Failed password for administrator
from 10.0.2.15 port 58036 ssh2
```

Wir teilen die Abfrage unten mit, um ihre Bestandteile besser zu verstehen:

sum by(add) (rate({job="varlogs", instance=~"$instance"}	Hiermit wird die Aufsummierung der Benutzernamen definiert, die wir mit "Patterns" in LogQL definiert haben. "Patterns" ermöglichen die einfache Extrahierung von Informationen aus einer Zeile. Wir holen alle Log-Einträge, die sich auf den Job "varlogs" beziehen. Wir können auch nach spezifischen Endpoint filtern, indem wir das Schlüsselwort „instance" benutzen.
|	„|" funktioniert in LogQL wie eine Pipeline für die Verkettung von mehreren Suchmustern.
|= `sshd[` |= `: Failed`	Regular Expression für die Suche nach Zeilen mit diesen Einträgen.
!~ `invalid user` !~ `test` !~ `10.0.2.15`	Regular Expression für die Suche nach Zeilen **ohne** diese Einträge. Wir können beispielsweise Einträge ausschließen, die sich nicht bösartige Nutzer sind, um falsche Positive zu vermeiden
| pattern `<_> for <Benutzername> from <QuelleAddress> port <_>` [$__range]))	Die Definition der Wörter "Benutzername" " QuelleAddress" und als "Pattern" dienen dazu, einen Benutzernamen und eine Quelle IP-Adresse aus der Logdatei zu extrahieren. Die Platzhalter "<_>" sind unbenannte Elemente, die in diesem Fall auf die Einträge "password" und Portnummber in der Zeile verweisen.

Tabelle 2: Aufbau der Regelsätze in Grafana Loki für SSH Logdateien
Quelle: Eigene Quelle, (VoidQuark, 2022) und (Grafana Labs, 2021c)

Das sollte verbessert werden Eine Erkennungsregel hätte folgende Logik:

```
# Gefundene Werte in den Logdateien
# Av = Anzahl fehlgeschlagener Anmeldungsversuche
# Ia = Intervallzeit zwischen fehlgeschlagenen Anmeldungsversuchen

# Festgelegte Werte für legitime und bösartige Verbindungen
# Ga = Grenze zwischen legitimen und bösartigen Anmeldungsversuchen
# Nt = Intervallzeit zwischen legitimen Anmeldungsversuchen

wenn (Av >= Ga) und (Ia < Nt)
   Warnmeldung(Brutefoce)
sonst
   weiterBeobachten()
```

3.5. Hinzufügen der Regelsätze Grafana Loki

Die Regelsätze in Grafana Loki können sowohl manuell im Menü „Code“ als auch über die GUI im Menü „Builder“ geschrieben werden. Letzteres bietet eine benutzerfreundlichere Umgebung, um die Regeln zu schreiben. Die folgenden Abbildungen zeigen diese beiden Optionen:

```
sum by (username) (count_over_time({job=~"varlogs", job=~".*",
instance=~".*"} |="sshd[" |~": Invalid|: Connection closed by
authenticating user|: Failed .* user" | pattern `<_> user <username> <_>
port` | __error__="" [2m]))
```

Abbildung 29: „Code“ in Grafana Loki für manuelle die Eingabe des LogQL-Codes. Quelle: (VoidQuark, 2022)

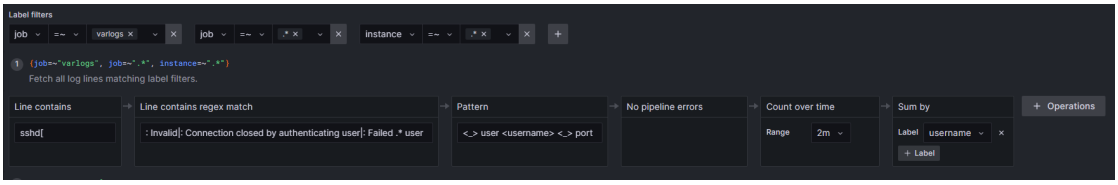


Abbildung 30: „Builder“ in Grafana Loki für nutzerfreundlichere Eingabe des LogQL-Codes. Quelle: (VoidQuark, 2022)

Beide Optionen bieten die Möglichkeit, eine Erklärung zur Abfrage anzuzeigen:

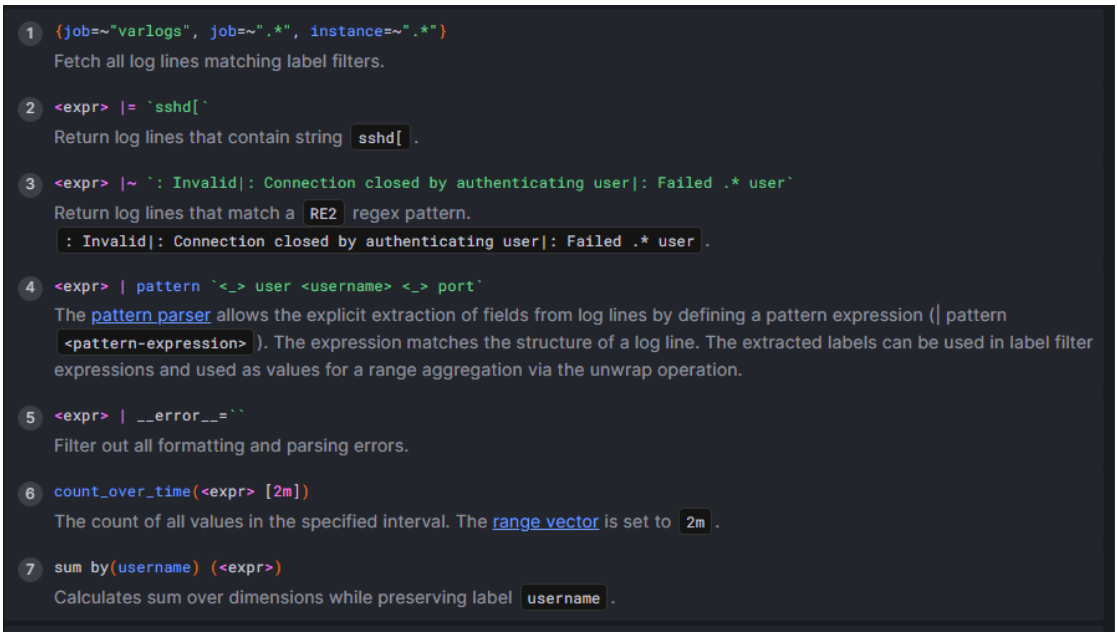
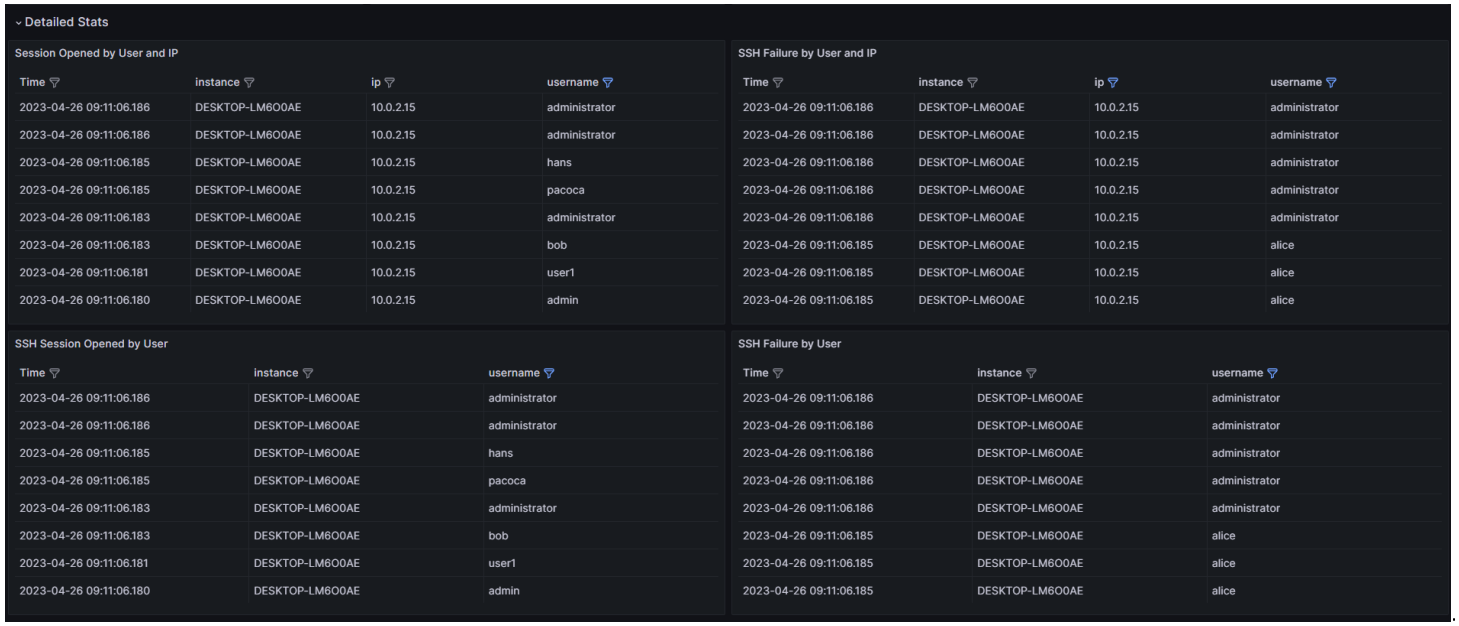


Abbildung 31: Ausführliche Information über die Abfrage
Quelle: (Grafana Labs, 2022d)

Nachdem die SSH-Logdateien gelesen und bearbeiten wurden, bekommen wir von Grafana Loki folgende Zasammenfasung der Ergebnissen:



Detailed Stats

Session Opened by User and IP

Time	instance	ip	username
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	10.0.2.15	hans
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	10.0.2.15	pacoca
2023-04-26 09:11:06.183	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.183	DESKTOP-LM6O0AE	10.0.2.15	bob
2023-04-26 09:11:06.181	DESKTOP-LM6O0AE	10.0.2.15	user1
2023-04-26 09:11:06.180	DESKTOP-LM6O0AE	10.0.2.15	admin

SSH Failure by User and IP

Time	instance	ip	username
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	10.0.2.15	alice
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	10.0.2.15	alice
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	10.0.2.15	alice

SSH Session Opened by User

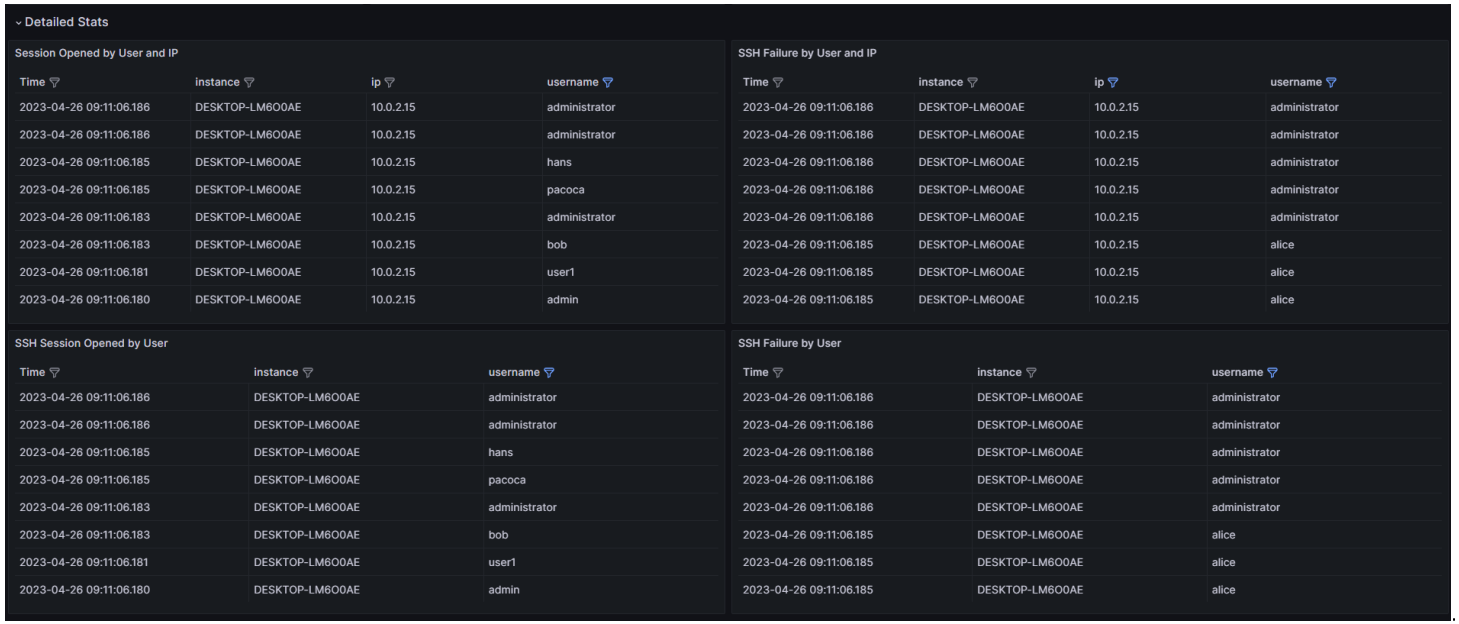
Time	instance	username
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	hans
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	pacoca
2023-04-26 09:11:06.183	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.183	DESKTOP-LM6O0AE	bob
2023-04-26 09:11:06.181	DESKTOP-LM6O0AE	user1
2023-04-26 09:11:06.180	DESKTOP-LM6O0AE	admin

SSH Failure by User

Time	instance	username
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	alice
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	alice
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	alice

Abbildung 32: Bearbeitung der SSH Logdateien von Grafana Loki
Quelle: Eigene Quelle and (VoidQuark, 2022)

Das nächste Bild gibt ausführliche Informationen der Logdateien: **Bild Korrigieren**



Detailed Stats

Session Opened by User and IP

Time	instance	ip	username
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	10.0.2.15	hans
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	10.0.2.15	pacoca
2023-04-26 09:11:06.183	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.183	DESKTOP-LM6O0AE	10.0.2.15	bob
2023-04-26 09:11:06.181	DESKTOP-LM6O0AE	10.0.2.15	user1
2023-04-26 09:11:06.180	DESKTOP-LM6O0AE	10.0.2.15	admin

SSH Failure by User and IP

Time	instance	ip	username
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	10.0.2.15	administrator
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	10.0.2.15	alice
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	10.0.2.15	alice
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	10.0.2.15	alice

SSH Session Opened by User

Time	instance	username
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	hans
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	pacoca
2023-04-26 09:11:06.183	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.183	DESKTOP-LM6O0AE	bob
2023-04-26 09:11:06.181	DESKTOP-LM6O0AE	user1
2023-04-26 09:11:06.180	DESKTOP-LM6O0AE	admin

SSH Failure by User

Time	instance	username
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.186	DESKTOP-LM6O0AE	administrator
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	alice
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	alice
2023-04-26 09:11:06.185	DESKTOP-LM6O0AE	alice

Abbildung 33: Ausführliche Darstellung der SSH Logdateien von Grafana Loki
Quelle: Eigene Quelle and (VoidQuark, 2022)

3.6. Einrichtung des Warnmeldungskomponent

In den vorherigen Teilen dieser Arbeit haben wir uns damit auseinandergesetzt, Grafana so einzustellen, dass wir schließlich eine Lösung ähnlich einer SIEM erhalten. Von unseren ursprünglichen Vorschlägen haben wir bereits Folgendes erreicht:

1. Sammlung der Logdateien aus den Endpoints mit Promtail
2. Anpassung der Logdateien für die nachträgliche visuelle Darstellung mit Loki
3. Nutzung von Regelsätzen in Loki für die Analysierung der SSH Logdateien
4. Graphische Darstellung der Logdateien in Grafana mit den in Loki verwendeten Regelsätzen

Unser letztes Ziel besteht darin, Warnmeldungen für potenzielle Angriffe mithilfe der Ergebnisse von Loki zu generieren. Grafana kann intern und extern mit Tools integriert werden, um Warnmeldungen zu erstellen. Eines dieser externen Tools ist der **Alertmanager**, der bereits integriert ist. Dieses Tool kann Daten von Prometheus, Cortex und Mimir als Datenquelle verwenden (Grafana Labs, 2021a) und kann Daten von beliebigen Endpoints empfangen. Die Regelsätze des Alertmanagers haben folgendes Muster:

```
# Warnmeldungen können in beliebigen Gruppen kategorisiert werden. Diese
können von den Nutzern entsprechend ihrer Anforderungen und Bedürfnisse
definiert werden.
groups:

      # Ab diesem Punkt beginnen wir mit der Definition der Regelsätze
      für die Erkennung von Warnmeldungen. Diese umfassen:
   - name: example
     rules:
    - alert: HighRequestLatency

      # LogQL-Regelsätze für die Erkennung der Warnmeldung, welche die
      in den vorherigen Schritten definierten Abfragen verwenden.
      expr: job:request_latency_seconds:mean5m{job="myjob"} > 0.5
      for: 10m
       labels:
         severity: page
       annotations:
         summary: High request latency
```

Grafana hat auch ein eigenes internes Tool, um Warnmeldungen zu konfigurieren: **Alerting**. In dieser Arbeit versuchen wir unser Warnmeldungs-System mithilfe dieses Tools aufzubauen.

Die Warnmeldungen können direkt in der GUI von Grafana konfiguriert werden. Dazu folgt man den folgenden Schritten (Grafana Labs, 2019):

1. Name der Regel
2. Regelsätze in LogQL
3. Definition von Gruppen für jede Art von Warnmeldung. Gruppen können später verschiedenen Einstellungen zugewiesen werden, wie z.B. Benachrichtigungen und Inhalte.
4. Informationen über die Warnmeldung, wie eine eindeutige ID und eine Beschreibung. Der Nutzer kann diese Felder so definieren, wie es notwendig ist.
5. Benachrichtigung der Zielgruppe, die diesen Fall später bearbeiten wird.
6. Labels zur besseren Organisation der Warnmeldungen.
7. Konfiguration von E-Mail in Grafana für die Weiterleitung der Warnmeldungen.

Für unseren ersten Test möchten wir Warnmeldungen für fehlgeschlagene Anmeldeversuche erstellen. Wir haben die oben genannten Elemente definiert und die folgenden Regelsätze verwendet (VoidQuark, 2022):

```
# (A) Anzahl von fehlgeschlagenen Anmeldeversuche für existierenden
Benutzernamen:
sum by (username) (count_over_time({$label_name=~"$label_value",
job=~"$job", instance=~"$instance"} |="sshd[" |~": Invalid|:
Connection closed by authenticating user|: Failed .* user" |
pattern `<_> user <username> <_> port` | __error__=""
[$__interval]))

# (B) Anzahl von Fehlgeschlagenen Anmeldeversuche für nicht
existierenden Benutzernamen:
sum by (username) (count_over_time({$label_name=~"$label_value",
job=~"$job", instance=~"$instance"} |="sshd[" |=": Failed" !~"invalid
user" | pattern `<_> for <username> from <_> port` | __error__=""
[$__interval]))

# Wenn die Anzahl von (A) oder von (B) größer als fünf ist, dann wird
die Warnmeldung als E-Mail an dem Ziel geschickt.
```

Im Anhang (Siehe Anhang C) befindet sich die Konfigurationsdatei für unsere Warnmeldung. Nachdem alles korrekt konfiguriert wurde, haben wir die folgende E-Mail erhalten:

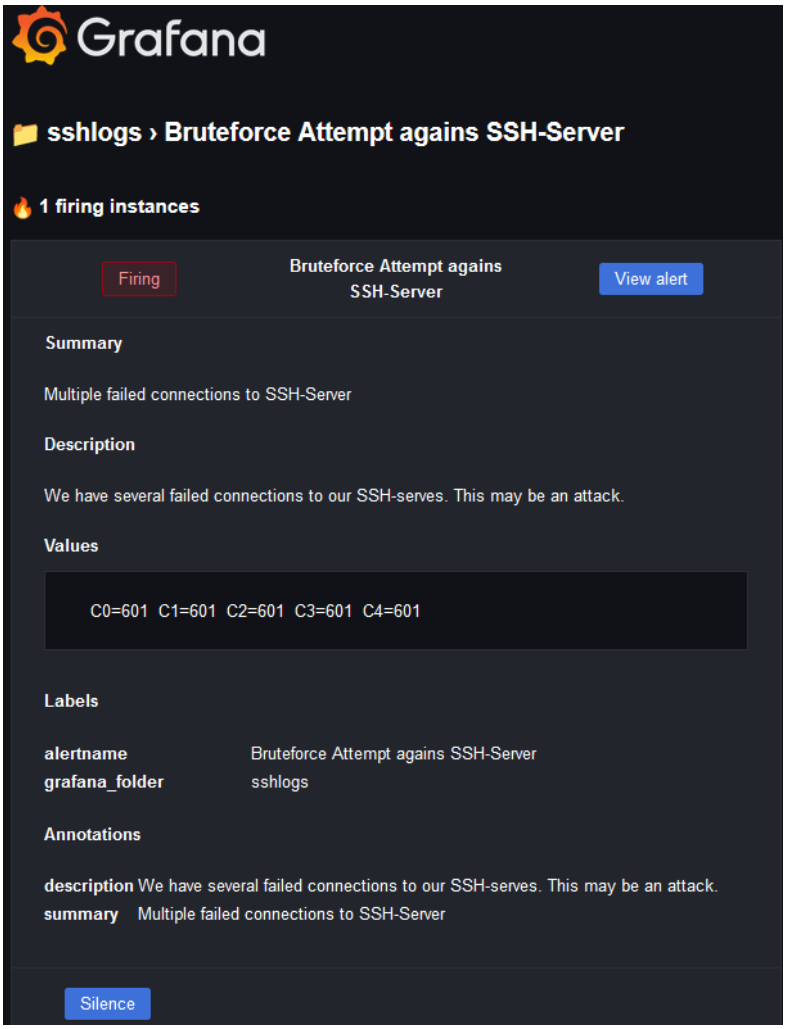


Abbildung 34: E-Mail Warnmeldung von Grafana
Quelle: Eigene Quelle und (Grafana Labs, 2019)

Das Alerting-Tool von Grafana bietet keine direkte Integration zu einem IDS, IPS, SIEM oder einer API an. Die Kommunikation mit solchen Endpoints lässt sich jedoch mithilfe von **Webhook** konfigurieren (Grafana Labs, 2022e).

4. Evaluation der Implementation mit echten Logdateien

5. Fazit

In dieser Arbeit haben wir versucht, eine Open Source-basierte Lösung ähnlich einem SIEM zu verwenden, um Überwachungsmechanismen anhand von Logdateien zu erstellen. In der folgenden Tabelle vergleichen wir die vorhandenen Funktionalitäten eines SIEM mit denen, die wir durch unsere Implementierung erreichen konnten.

Funktionalitäten	Proprietäre Lösung	Verwendete Open Source Tools
Datensammlung und Normalisierung	SIEM	Promtail
Daten Zusammenhang		Grafana Loki
Generierung von Berichten		Grafana
Warnmeldungen		Alerting (integriert in Grafana)
Datenauswertung		Grafana Loki

Tabelle 3: Verwendete Tools für den Aufbau einer SIEM ähnlichen Lösung
Quelle: Eigene Quelle und (Granadillo et al., 2021)

Aus prinzipieller Sicht können wir feststellen, dass die verwendeten Tools eine kosteneffektive Möglichkeit bieten, ein Überwachungssystem in einem Rechenzentrum zu implementieren. Die Methoden zur Erkennung von Angriffen lassen sich klar anhand der Mitre ATT&CK-Matrix oder anderer Frameworks definieren. Nach der Auswahl des Angriffe erstellen wir Regelwerke mit der Abfragesprache LogQL in Loki, um Muster zu identifizieren, die auf den ausgewählten Angriff hindeuten. Diese Regelwerke werden dann verwendet, um Warnmeldungen über den Angriff zu generieren und zu versenden.

Unser Aufbau birgt zwei große Herausforderungen, wobei die erste einfacher zu bewältigen ist als die zweite. Diese sind:

- **Definition der Regelsätzen**

Für eine präzise Implementierung spielt die richtige Entwicklung der Regelsätzen zur Identifizierung potenzieller Angriffe eine wesentliche Rolle. Da Logdateien aus produkti-

ven Umgebungen eine große Menge an Informationen enthalten, müssen diese Regelsätzen so definiert werden, dass sie die eindeutigen Informationen wie IP-Adresse, Portnummer, Zeitfenster und Zeitabstände zwischen Anfragen filtern und nach Angriffsmustern kategorisieren können.

- **statische Regel in einer dynamischen Angriffswelt**

Die von uns definierten Regeln haben statische Elemente wie die „Anzahl von Anfragen", den „Zeitabstand zwischen Requests" und die „Anzahl von fehlgeschlagenen Anmeldeversuchen". Die heutigen Angriffe haben jedoch auch einen dynamischen Aspekt, der sich an die Umgebung anpasst, insbesondere durch die starke Entwicklung von Künstliche Intelligenz (KI). Während KI einerseits für die Automatisierung von Aufgaben oder für effiziente Datenanalyse verwendet wird, könnte sie auch für Cyberkriminalität genutzt werden. KI ist am Ende nur ein Werkzeug, dessen Nutzung von den Absichten ihrer Benutzer abhängt.

Verschiedene Angriffstechniken lassen sich schneller und effizienter mit KI durchführen. Die Nutzung von Polymorphe Malware ist ein Beispiel, wo weder Antivirus-Programme noch Log-Analyse-Tools einen normalen von einem abnormalen Ablauf unterscheiden können. Auch die Verkehrsanalyse kann durch KI gefährdet sein, da Angriffe und normaler Verkehr ähnlich dargestellt werden können. Darüber hinaus kann KI auch gegen Authentifizierungsverfahren eingesetzt werden, um beispielsweise Anmeldedaten schneller zu erraten und/oder vorauszusehen (Fritsch et al., 2022).

Das folgende Diagramm zeigt, wo sich KI bei Cyberangriffen anhand derCKC integrieren lässt:

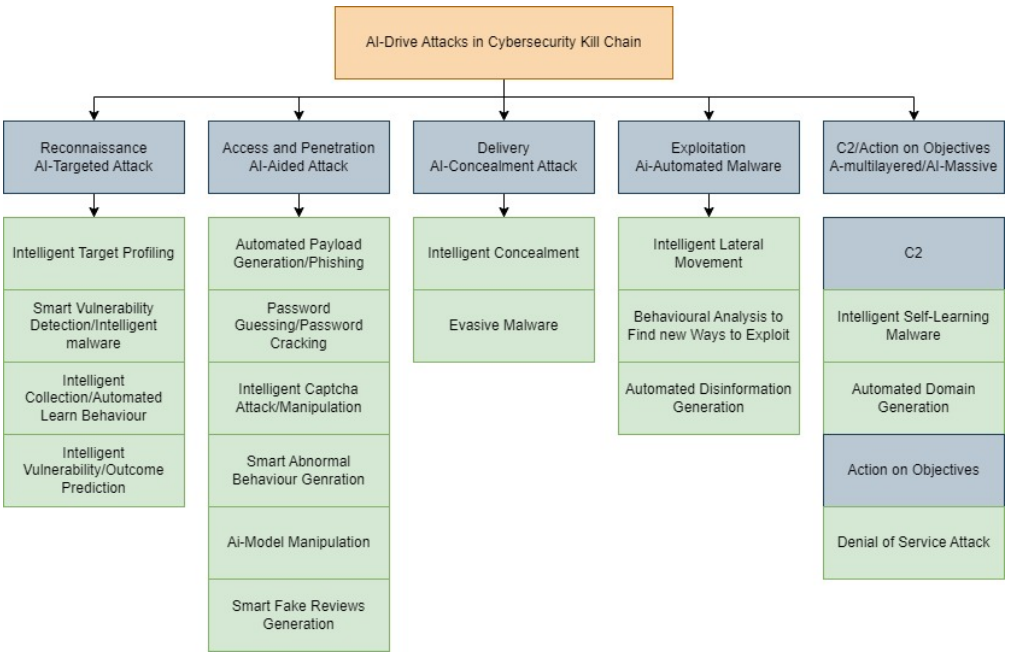


Abbildung 35: KI in der Cyber Kill Chain (CKC)
Quelle: (Guembe et al., 2022)

5.1. Zukünftige Recherche

Um sicherzustellen, dass unsere vorgeschlagenen Lösungen sich an diese neue und dynamische Realität anpassen können, können zukünftige Regelsätze mithilfe von KI erstellt werden. Nachdem die meisten möglichen Angriffsflächen abgedeckt wurden, sollten die Regeln so angepasst werden, dass sie möglichst viele Szenarien abdecken.

Mit der rasanten Entwicklung von KI, insbesondere während der Erstellung dieser Arbeit, können wir auch erwarten, dass sich sowohl Loki als auch Grafana bald mit verschiedenen Open Source Plugin integrieren lassen, die auch KI unterstützen. Diese sollen dazu beitragen, die Loganalyse effizienter und zuverlässiger zu machen. All dies würde dabei helfen, einen sicheren Netzwerkverkehr zu gewährleisten.

Literaturverzeichnis

Advani, S., Mridul, M., Vij, P. S. R., Agarwal, M., and A., L. P. (2020). Iot data analytics pipeline using elastic stack and kafka. *International Journal of Computer Sciences and Engineering*, 8:144–148.
https://www.ijarcce.com/upload/2016/april-16/IJARCCE%2013.pdf. Zugriff am 07.03.2023.

at (2022). Abfragesprache.
https://www.alexanderthamm.com/de/data-science-glossar/abfragesprache/. Zugriff am 08.04.2023.

AT&T Cybersecurity (2022). Alienvault ossim.
https://cybersecurity.att.com/products/ossim. Zugriff am 05.03.2023.

Ba, M. H. N., Bennett, J., Gallagher, M., and Bhunia, S. (2021). A case study of credential stuffing attack: Canva data breach. In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 735–740.
https://doi.org/10.1109/CSCI54926.2021.00187. Zugriff am 26.03.2023.

BSI (2021). Zweites Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme (IT-Sicherheitsgesetz 2.0).
https://www.bsi.bund.de/DE/Das-BSI/Auftrag/Gesetze-und-Verordnungen/IT-SiG/2-0/it_sig-2-0_node.html. Zugriff am 04.03.2023.

CBNINSIGHTS (2020). Alienvault.
https://www.cbinsights.com/company/alienvault. Zugriff am 05.03.2023.

Centers for Disease Control and Prevention (2016). Health Insurance Portability and Accountability Act of 1996 (HIPAA).
https://www.pcicomplianceguide.org/faq/. Zugriff am 04.03.2023.

Chai, W. and Ferguson, K. (2021). What is HTTP?
https://www.techtarget.com/whatis/definition/HTTP-Hypertext-Transfer-Protocolk. Zugriff am 17.04.2023.

Collins, C., Dennehy, D., Conboy, K., and Mikalef, P. (2021). Artificial intelligence in information systems research: A systematic literature review and research agenda. *International Journal of Information Management*, 60:102383.
https://www.sciencedirect.com/science/article/pii/S0268401221000761. Zugriff am 21.02.2023.

comparitech (2023). The Best SIEM Tools for 2023 Vendors & Solutions Ranked.
https://www.comparitech.com/net-admin/siem-tools/. Zugriff am 05.03.2023.

Dorigo, S. (2012). Security Information and Event Management. Master's thesis, Radboud University Nijmegen.
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiNu-XkhsD9AhV4FzQIHdMkBWYQFnoECCYQAQ&url=https%3A%2F%2Fwww.ru.nl%2Fpublish%2Fpages%2F769526%2Fthesissanderdorigo.pdf&usg=AOvVaw3oPn4KBFwgJwexoXZ1Be40. Zugriff am 03.03.2023.

Douglis, F. and Nieh, J. (2019). Microservices and containers. *IEEE Internet Computing*, 23(6):5–6.
https://doi.org/10.1109/MIC.2019.2955784. Zugriff am 236.03.2023.

elastic (2021). *FAQ on 2021 License Change*.
https://www.elastic.co/pricing/faq/licensing. Zugriff am 26.03.2023.

elastic (2022). *Elastic Docs*.
https://www.elastic.co/guide/en/welcome-to-elastic/current/new.html. Zugriff am 5.02.2023.

European Comission (2015). SIEM design and development.
https://cordis.europa.eu/project/id/644425. Zugriff am 05.03.2023.

Fortinet (2016). Fortinet Announces Acquisition of AccelOps .
https://www.fortinet.com/corporate/about-us/newsroom/press-releases/2016/fortinet-announces-acquisition-of-accelops. Zugriff am 06.03.2023.

Fortinet (2020). FortiSIEM Reference Architecture.
https://www.fortinet.com/content/dam/maindam/PUBLIC/02_MARKETING/02_Collateral/DeploymentGuide/dg-fortisiem-reference-architecture.pdf. Zugriff am 06.03.2023.

Fortinet (2022). FortiSIEM Solutions.
https://www.fortinet.com/products/siem/fortisiem. Zugriff am 06.03.2023.

Fritsch, L., Jaber, A., and Yazidi, A. (2022). An overview of artificial intelligence used in malware. In Zouganeli, E., Yazidi, A., Mello, G., and Lind, P., editors, *Nordic Artificial Intelligence Research and Development*, pages 41–51, Cham. Springer International Publishing.
https://doi.org/10.1007/978-3-031-17030-0_4. Zugriff am 25.04.2023.

Fu, F. (2018). Chapter six - design and analysis of complex structures. In *Design and Analysis of Tall and Complex Structures*, pages 177–211. Butterworth-Heinemann.
https://www.sciencedirect.com/science/article/pii/B978008101018100006X. Zugriff am 06.03.2023.

Grafana Labs (2016). Dashboard anything. Observe everything.
https://grafana.com/grafana/. Zugriff am 12.03.2023.

Grafana Labs (2018). Grafana Loki.
https://grafana.com/oss/loki/. Zugriff am 08.04.2023.

Grafana Labs (2019). Alerting.
https://grafana.com/docs/grafana/latest/alerting/. Zugriff am 21.04.2023.

Grafana Labs (2020a). Getting started.
https://grafana.com/docs/loki/latest/getting-started/. Zugriff am 09.04.2023.

Grafana Labs (2020b). Grafana Loki HTTP API.
https://grafana.com/docs/loki/latest/api/. Zugriff am 17.04.2023.

Grafana Labs (2021a). Alertmanager.
https://grafana.com/docs/grafana/latest/alerting/manage-notifications/alertmanager/. Zugriff am 21.04.2023.

Grafana Labs (2021b). Collect logs with Grafana Agent.
https://grafana.com/docs/grafana-cloud/data-configuration/logs/collect-logs-with-agent/. Zugriff am 17.04.2023.

Grafana Labs (2021c). LogQL: Log query language.
https://grafana.com/docs/loki/latest/logql/. Zugriff am 14.04.2023.

Grafana Labs (2021d). What is opentelemetry?
https://grafana.com/oss/opentelemetry/. Zugriff am 17.04.2023.

Grafana Labs (2022a). Dashboard anything. Observe everything.
https://grafana.com/logs/. Zugriff am 12.03.2023.

Grafana Labs (2022b). Grafana Agent.
https://grafana.com/docs/agent/latest/. Zugriff am 17.04.2023.

Grafana Labs (2022c). How to send logs to grafana loki with the opentelemetry collector using fluent forward and filelog receivers.
https://grafana.com/blog/2022/06/23/how-to-send-logs-to-grafana-loki-with-the-opentelemetry-collector-using-fluent-forward-and-filelog-receivers/. Zugriff am 17.04.2023.

Grafana Labs (2022d). Loki query editor.
https://grafana.com/docs/grafana/latest/datasources/loki/query-editor/. Zugriff am 26.04.2023.

Grafana Labs (2022e). Notifications.
https://grafana.com/docs/grafana/latest/alerting/fundamentals/notifications/. Zugriff am 26.04.2023.

Grafana Labs (2022f). What is Grafana Mimir?
https://grafana.com/docs/loki/latest/logql/. Zugriff am 21.04.2023.

Granadillo, G., González-Zarzosa, S., and Diaz, R. (2021). Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures. *Sensors*, 21:4759.
file:///C:/Users/bruno/Downloads/Security_Information_and_Event_Management_SIEM_Ana.pdf. Zugriff am 21.02.2023.

Guembe, B., Azeta, A., Misra, S., Osamor, V. C., Fernandez-Sanz, L., and Pospelova, V. (2022). The emerging threat of ai-driven cyber attacks: A review. *Applied Artificial Intelligence*, 36(1):2037254.
https://www.tandfonline.com/doi/full/10.1080/08839514.2022.2037254. Zugriff am 24.04.2023.

Gómez, E. C. F., Almeida, O. X. B., and Gamboa, L. M. A. (2022). Analysis of centralized computer security systems through the alienvault ossim tool. *Ecuadorian Science Journal*, 6(1):23–31.

https://journals.gdeon.org/index.php/esj/article/view/181. Zugriff am 03.03.2023.

Hazel, T. (2021). How To Use the MITRE ATT&CK Framework. https://www.chaossearch.io/blog/how-to-use-mitre-attck-framework. Zugriff am 26.03.2023.

IBM (2020). What is an api (application programming interface)? https://www.ibm.com/topics/api. Zugriff am 17.04.2023.

Ibrokhimov, S., Hui, K. L., Al-Absi, A. A., hoon jae lee, and Sain, M. (2019). Multi-Factor Authentication in Cyber Physical System: A State of Art Survey. In *2019 21st International Conference on Advanced Communication Technology (ICACT)*, pages 279–284. https://doi.org/10.23919/ICACT.2019.8701960, Zugriff am 26.03.2023.

IT-Service.Network (2020). Was ist ein plug-in? https://www.hhs.gov/hipaa/for-professionals/privacy/index.html. Zugriff am 04.03.2023.

Jain, U. (2018). *Lateral Movement Detection Using ELK Stack*. PhD thesis, University of Houston. https://uh-ir.tdl.org/handle/10657/3109. Zugriff am 07.03.2023.

Janiesch, C., Zschech, P., and Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3):685–695. https://doi.org/10.1007/s12525-021-00475-2. Zugriff am 13.03.2023.

Jog, Y. (2020). Security Information and Event Management (SIEM). https://www.linkedin.com/pulse/security-information-event-management-siem-yatin-jog. Zugriff am 04.03.2023.

Kali (2019). Kali inside virtualbox (guest vm). https://www.kali.org/docs/virtualization/install-virtualbox-guest-vm/. Zugriff am 02.04.2023.

Kali (2022a). Hydra. https://www.kali.org/tools/hydra/. Zugriff am 02.04.2023.

Kali (2022b). What is kali linux & kališ features. https://www.kali.org/docs/introduction/. Zugriff am 02.04.2023.

Kazarov, A., Avolio, G., Chitan, A., and Mineev, M. (2018). Experience with splunk for archiving and visualisation of operational data in atlas tdaq system. *Journal of Physics: Conference Series*, 1085:32052. http://dx.doi.org/10.1088/1742-6596/1085/3/032052. Zugriff am 04.03.2023.

Manases, L. and Zinca, D. (2022). Automation of network traffic monitoring using docker images of snort3, grafana and a custom api. In *2022 21st RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–4. https://doi.org/10.1109/RoEduNet57163.2022.9921063. Zugriff am 13.03.2023.

Martin, L. (2018). The cyber kill chain. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-cha

in.html. Zugriff am 12.03.2023.

Maymi, F., Bixler, R., Jones, R., and Lathrop, S. (2017). Towards a definition of cyberspace tactics, techniques and procedures. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 4674–4679.
http://dx.doi.org/10.1109/BigData.2017.8258514. Zugriff am 09.05.2023.

Microsoft Security (2022). Endpoints defined.
https://www.microsoft.com/en-us/security/business/security-101/what-is-an-endpoint. Zugriff am 12.03.2023.

Mikalauskas, E. (2023). Rockyou2021: largest password compilation of all time leaked online with 8.4 billion entries.
https://cybernews.com/security/rockyou2021-alltime-largest-password-compilation-leaked/. Zugriff am 02.04.2023.

Miller, J. (2021). Is Elastic STACK (ELK) the best SIEM option?
https://www.bitlyft.com/resources/is-elk-the-best-siem-option#:~:text=The%20ELK%20stack%20is%20a,system%20from%20a%20system%20provider. Zugriff am 07.03.2023.

MITRE ATT&CK (2018a). Frequently Asked Questions.
https://attack.mitre.org/resources/faq/. Zugriff am 12.03.2023.

MITRE ATT&CK (2018b). Getting Started.
https://attack.mitre.org/resources/getting-started/. Zugriff am 26.03.2023.

MITRE ATT&CK (2020). Brute Force.
https://attack.mitre.org/techniques/T1110/. Zugriff am 26.03.2023.

Mohammed, S. A., Mohammed, A. R., Côté, D., and Shirmohammadi, S. (2021). A machine-learning-based action recommender for network operation centers. *IEEE Transactions on Network and Service Management*, 18(3):2702–2713.
https://doi.org/10.1109/TNSM.2021.3095463. Zugriff am 20.02.2023.

Mohanan, R. (2022). What Is Security Information and Event Management (SIEM)? Definition, Architecture, Operational Process, and Best Practices.
https://www.spiceworks.com/it-security/vulnerability-management/articles/what-is-siem/. Zugriff am 26.02.2023.

Nabil, M., Soukainat, S., Lakbabi, A., and Ghizlane, O. (2017). SIEM selection criteria for an efficient contextual security. In *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6.
https://doi.org/10.1109/ISNCC.2017.8072035. Zugriff am 26.02.2023.

neptune (2023). A Machine Learning Approach to Log Analytics: How to Analyze Logs? .
https://www.nexcess.net/blog/open-source-vs-proprietary/. Zugriff am 12.03.2023.

Nexcess (2022). Open source vs. proprietary: Which is better?
https://www.nexcess.net/blog/open-source-vs-proprietary/. Zugriff am

26.02.2023.

NIST (2020a). About nist.
https://www.nist.gov/about-nist. Zugriff am 19.02.2023.

NIST (2020b). Cyber attacke.
https://csrc.nist.gov/glossary/term/Cyber_Attack. Zugriff am 19.02.2023.

NIST (2020c). False positive.
https://csrc.nist.gov/glossary/term/false_positive. Zugriff am 05.03.2023.

NIST (2020d). Glossary.
https://csrc.nist.gov/glossary/. Zugriff am 19.02.2023.

Open Source Initiative (2007). The Open Source Definition (Annotated).
https://opensource.org/definition/. Zugriff am 17.02.2023.

packt (2019). What is elk stack?
https://subscription.packtpub.com/book/big-data-and-business-intellige
nce/9781788831031/1/ch01lvl1sec10/what-is-elk-stack. Zugriff am 07.03.2023.

Polinowski, M. (2019). What is elk stack?
https://mpolinowski.github.io/docs/DevOps/Provisioning/2021-04-07--lok
i-prometheus-grafana/2021-04-07/. Zugriff am 09.04.2023.

Prelude SIEM (2018). Prelude SIEM: Smart Security.
https://www.prelude-siem.com/en/prelude-siem-en/. Zugriff am 05.03.2023.

Prelude SIEM (2020). *Prelude Documentation: version 5.2.*
https://www.prelude-siem.org/docs/5.2/en/. Zugriff am 06.03.2023.

Prelude Team (2007). *Manual User.*
https://www.prelude-siem.org/projects/prelude/wiki/. Zugriff am 06.03.2023.

Project, T. (2021). Thehive - a 4-in-1 security incident response platform.
https://thehive-project.org/. Zugriff am 21.04.2023.

Prometheus (2015). Jobs and instances.
https://prometheus.io/docs/concepts/jobs_instances/. Zugriff am 08.05.2023.

Prometheus (2016). Documentation.
https://prometheus.io/docs/introduction/overview/. Zugriff am 14.04.2023.

Radoglou-Grammatikis, P., Sarigiannidis, P., Iturbe, E., Rios, E., Martinez, S., Sarigiannidis, A., Eftathopoulos, G., Spyridis, Y., Sesis, A., Vakakis, N., Tzovaras, D., Kafetzakis, E., Giannoulakis, I., Tzifas, M., Giannakoulias, A., Angelopoulos, M., and Ramos, F. (2021). Spear siem: A security information and event management system for the smart grid. *Computer Networks*, 193:108008.
https://doi.org/10.1016/j.comnet.2021.108008. Zugriff am 03.03.2023.

Ramírez Tomás, I. (2018). *Implementación de un sistema de gestión de eventos de seguridad en una empresa de tamaño medio.* PhD thesis, Universitat Politècnica de València.
https://riunet.upv.es/bitstream/handle/10251/109765/Ram%c3%adrez%20-%20

Implementaci%c3%b3n%20de%20un%20sistema%20de%20gesti%c3%b3n%20de%20eventos%20de%20seguridad%20en%20una%20empresa%20de%20tama%c3%b1....pdf?sequence=1&isAllowed=y. Zugriff am 06.03.2023.

redhat (2022). What is grafana? https://www.redhat.com/en/topics/data-services/what-is-grafana. Zugriff am 13.03.2023.

Roser, M., Ritchie, H., and Ortiz-Ospina, E. (2015). Internet. *Our World in Data*. https://ourworldindata.org/internet. Zugriff am 17.02.2023.

Savic, D., da Silva, A. R., Vlajic, S., Lazarevic, S., Stanojevic, V., Antovic, I., and Milic, M. (2012). Use case specification at different levels of abstraction. In *2012 Eighth International Conference on the Quality of Information and Communications Technology*, pages 187–192. https://doi.org/10.1109/QUATIC.2012.64. Zugriff am 12.03.2023.

Selamat, N. S., Ali, F. H. M., and Othman, N. A. A. (2016). Polymorphic malware detection. In *2016 6th International Conference on IT Convergence and Security (ICITCS)*, pages 1–5. https://doi.org/10.1109/ICITCS.2016.7740362. Zugriff am 24.04.2023.

Selvaganesh, M., Karthi, P., Kumar, V. A. N., and Moorthy, S. R. P. (2022). Efficient brute-force handling methodology using indexed-cluster architecture of splunk. In *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, pages 697–701. https://doi.org/10.1109/ICEARS53579.2022.9752323. Zugriff am 12.03.2023.

Setter, M. (2015). Logfmt: A Log Format That's Easy To Read and Write. https://www.cloudbees.com/blog/logfmt-a-log-format-thats-easy-to-read-and-write. Zugriff am 10.04.2023.

silicon.de (2022). Das beliebteste deutsche Passwort 2022 lautet: 123456. https://www.silicon.de/41703603/das-beliebteste-deutsche-passwort-2022-lautet-123456. Zugriff am 02.04.2023.

Sowmya, G. V., Jamuna, D., and Reddy, M. V. K. (2012). Blocking of Brute Force Attack. *International journal of engineering research and technology*, 1.

Splunk (2015a). Splunk Enterprise Security. https://www.splunk.com/en_us/products/enterprise-security.html. Zugriff am 12.03.2023.

Splunk (2015b). The splunk platform enables end-to-end visibility from edge to cloud. https://www.splunk.com/en_us/products/splunk-enterprise.html. Zugriff am 03.05.2023.

Splunk (2022a). Use Cases. https://docs.splunk.com/Documentation/ES/7.1.0/Usecases/Overview. Zugriff am 12.03.2023.

Splunk (2022b). What Is Security Information and Event Management (SIEM)?

https://www.splunk.com/en_us/data-insider/what-is-siem.html. Zugriff am 12.03.2023.

Su, T.-J., Wang, S.-M., Chen, Y.-F., and Liu, C.-L. (2016). Attack detection of distributed denial of service based on splunk. In *2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE)*, pages 397–400. https://doi.org/10.1109/ICAMSE.2016.7840355. Zugriff am 12.03.2023.

Swathi, K. (2022). Brute Force Attack on Real World Passwords. *International Journal of Research Publication and Reviews*, 3(11):552–558. https://www.ijrpr.com/archive.php?volume=3&issue=11. Zugriff am 26.02.2023.

Tanenbaum, A. S. (2009). *Moderne Betriebssysteme*. Pearson, München.

Tas, Y. C. (2021). What are webhooks? https://yagmurcetintas.com/journal/what-are-webhooks. Zugriff am 26.04.2023.

techopedia (2015). Security Event Management. https://www.techopedia.com/definition/25763/security-event-management. Zugriff am 03.03.2023.

techopedia (2022). Security Information Management (SIM). https://www.techopedia.com/definition/25763/security-event-management. Zugriff am 03.03.2023.

Tek-Tools (2020). Log Analysis – How to Use a Log Analyzer Tool? https://www.tek-tools.com/apm/choosing-log-analyzer-tool. Zugriff am 12.03.2023.

tutorialspoint (2009). HTTP - Methods. https://www.tutorialspoint.com/http/http_methods.htm. Zugriff am 17.04.2023.

Ubuntu (2023a). Get Ubuntu Server. https://ubuntu.com/download/server. Zugriff am 31.03.2023.

Ubuntu (2023b). Ubuntu. https://ubuntu.com/. Zugriff am 31.03.2023.

U.S. Department of Health & Human Services (2016). The HIPAA Privacy Rule. https://www.hhs.gov/hipaa/for-professionals/privacy/index.html. Zugriff am 04.03.2023.

Vielberth, M. (2021). *Encyclopedia of Cryptography, Security and Privacy*, chapter Security Operations Center (SOC), pages 1–3. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-27739-9_1680-1. Zugriff am 04.03.2023.

VoidQuark (2022). Parsing SSH Logs with Grafana Loki. https://voidquark.com/parsing-ssh-logs-with-grafana-loki/. Zugriff am 10.04.2023.

Wang, Y.-T., Yang, C.-T., Kristiani, E., and Chan, Y.-W. (2019). The implementation of wi-fi log analysis system with elk stack. In *Frontier Computing*, pages 246–255, Singapore. Springer Singapore.

https://link.springer.com/chapter/10.1007/978-981-13-3648-5_28. Zugriff am 07.03.2023.

Wendzel, S. (2018). *IT-Sicherheit für TCP/IP- und IoT-Netzwerke*. Springer Vieweg, Wiesbaden.

Ödegaard, T. (2019). The (Mostly) Complete History of Grafana UX. https://grafana.com/blog/2019/09/03/the-mostly-complete-history-of-grafana-ux/. Zugriff am 13.03.2023.

Łukasz Korzeniowski and Goczyła, K. (2022). Landscape of automated log analysis: A systematic literature review and mapping study. *IEEE Access*, 10:21892–21913. https://doi.org/10.1109/ACCESS.2022.3152549. Zugriff am 12.03.2023.

A. Originalle Einstellungsdateien

Unten befindet sich die originalle Konfigurationsdateien (Grafana Labs, 2020a):

- **Grafana Loki** für die Speicherung und Bearbeitung der Logdateien

```
auth_enabled: false

server:
  http_listen_port: 3100
  grpc_listen_port: 9096

common:
  instance_addr: 127.0.0.1
  path_prefix: /tmp/loki
  storage:
    filesystem:
      chunks_directory: /tmp/loki/chunks
      rules_directory: /tmp/loki/rules
  replication_factor: 1
  ring:
    kvstore:
      store: inmemory

query_range:
  results_cache:
    cache:
      embedded_cache:
        enabled: true
        max_size_mb: 100

schema_config:
  configs:
    - from: 2020-10-24
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h

ruler:
  alertmanager_url: http://localhost:9093

# By default, Loki will send anonymous, but uniquely-identifiable
usage and configuration
# analytics to Grafana Labs. These statistics are sent to
# https://stats.grafana.org/

# Statistics help us better understand how Loki is used, and they
# show us performance levels for most users. This helps us
#  prioritize features and documentation.

# For more information on what's sent, look at
# https://github.com/grafana/loki/blob/main/pkg/usagestats/stats.go
# Refer to the buildReport method to see what goes into a report.

# If you would like to disable reporting, uncomment the following
# lines analytics:
#  reporting_enabled: false
```

- **Promtail** für die Sammlung der Logdateien

```
  server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
  - url: http://loki:3100/loki/api/v1/push

scrape_configs:
- job_name: system
  static_configs:
  - targets:
      - localhost
    labels:
      job: varlogs
      __path__: /var/log/*log
```

B. Angepasste Einstellungsdateien von Grafana

Unten befindet sich die angepasste Konfigurationsdateien (Polinowski, 2019):

- **Grafana Loki**

```
auth_enabled: false

server:
  http_listen_port: 3100
  grpc_listen_port: 9096

ingester:
  wal:
    enabled: true
    dir: /tmp/wal
  lifecycler:
    address: 127.0.0.1
    ring:
      kvstore:
        store: inmemory
      replication_factor: 1
    final_sleep: 0s
  # Any chunk not receiving new logs in this time will be flushed
  chunk_idle_period: 1h
  # All chunks will be flushed when they hit this age, default is
  1h max_chunk_age: 1h
  # Loki will attempt to build chunks up to 1.5MB, flushing first
  if chunk_idle_period or max_chunk_age is reached first
  chunk_target_size: 1048576
  # Must be greater than index read cache TTL if using an index
  cache (Default index read cache TTL is 5m)
  chunk_retain_period: 30s
  # Chunk transfers disabled
  max_transfer_retries: 0

schema_config:
  configs:
    - from: 2020-10-24
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h

storage_config:
  boltdb_shipper:
    active_index_directory: /tmp/loki/boltdb-shipper-active
    cache_location: /tmp/loki/boltdb-shipper-cache
    # Can be increased for faster performance over longer query
    # periods, uses more disk space
    cache_ttl: 24h
    shared_store: filesystem
  filesystem:
    directory: /tmp/loki/chunks

compactor:
  working_directory: /tmp/loki/boltdb-shipper-compactor
  shared_store: filesystem

limits_config:
  reject_old_samples: true
```

```
  reject_old_samples_max_age: 168h

chunk_store_config:
  max_look_back_period: 0s

table_manager:
  retention_deletes_enabled: false
  retention_period: 0s

ruler:
  storage:
    type: local
    local:
      directory: /tmp/loki/rules
  rule_path: /loki/rules-temp
  alertmanager_url: http://localhost:9093
  ring:
    kvstore:
      store: inmemory
  enable_api: true
```

- **Promtail**

```
---
server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
  - url: http://loki:3100/loki/api/v1/push
    tenant_id: tenant1

scrape_configs:
- job_name: Opfersystem
  static_configs:
  - targets:
      - loki
    labels:
      instance: OpferSystem
      env: Variable
      job: varlogs
      __path__: /opt/*.log
```

- **Docker Compose Datei**

```
version: "3"

networks:
  loki:

services:
  loki:
    image: grafana/loki:2.3.0
    volumes:
      - <lokales_Verzeichnis>/loki-config.yaml:/etc/loki
        /loki-config.yaml
    ports:
      - "3100:3100"
    command: -config.file=/etc/loki/local-config.yaml
    networks:
      - loki

  promtail:
    image: grafana/promtail:2.3.0
    volumes:
      - <lokales_Verzeichnis>/promtail-config.yaml
      - <lokales_Verzeichnis>/ssh1.log:/opt/ssh1.log
      - <lokales_Verzeichnis>/ssh2.log:/opt/ssh2.log
    command: -config.file=/etc/promtail/promtail-config.yaml
    networks:
      - loki

  grafana:
    image: grafana/grafana:latest
    ports:
      - "3000:3000"
    networks:
      - loki
```

C. Angepasste Einstellungsdateien von Grafana

Unten befindet sich unser Regel für die Generierung von Warnmeldungen in Fälle eines Brute-Force Angriffes gegen SSH Server.

```
apiVersion: 1
groups:
    - orgId: 1
      name: sshTeam
      folder: sshlogs
      interval: 1m
      rules:
        - uid: lHYZTLPVz
          title: Bruteforce Attempt agains SSH-Server
          condition: C
          data:
            - refId: A
              queryType: range
              relativeTimeRange:
                from: 600
                to: 0
              datasourceUid: sx2e5YE4k
              model:
                datasource:
                    type: loki
                    uid: sx2e5YE4k
                editorMode: code
                expr: 'sum by(username) (count_over_time({job=~"varlogs",
                job=~".*", instance=~".*"} |= `sshd[` |~ `: Invalid|:
                Connection closed by authenticating user|: Failed .*
                user` != `test` | pattern `<_> user <username> <_> port`
                | __error__=`` [2400h]))'
                hide: false
                intervalMs: 1000
                maxDataPoints: 43200
                queryType: range
                refId: A
            - refId: B
              queryType: range
              relativeTimeRange:
                from: 600
                to: 0
              datasourceUid: sx2e5YE4k
              model:
                datasource:
                    type: loki
                    uid: sx2e5YE4k
                editorMode: code
                expr: 'sum by(username) (count_over_time({job=~"varlogs",
                job=~".*", instance=~".*"} |= `sshd[` |= `: Failed` !~
                `invalid user` != `test` | pattern `<_> for <username>
                from <_> port` | __error__=`` [2400h]))'
                hide: false
                intervalMs: 1000
                maxDataPoints: 43200
                queryType: range
                refId: B
            - refId: C
              datasourceUid: __expr__
              model:
                conditions:
                    - evaluator:
```

```
                  params:
                      - 5
                      - 0
                  type: gt
                operator:
                  type: and
                query:
                  params:
                      - A
                reducer:
                  params: []
                  type: count
                type: query
              - evaluator:
                  params:
                      - 5
                      - 0
                  type: gt
                operator:
                  type: or
                query:
                  params:
                      - B
                reducer:
                  params: []
                  type: count
                type: query
          datasource:
              name: Expression
              type: __expr__
              uid: __expr__
          expression: ""
          intervalMs: 1000
          maxDataPoints: 43200
          refId: C
          type: classic_conditions
      noDataState: NoData
      execErrState: Error
      for: 5m
      annotations:
        description: We have several failed connections to our
        SSH-serves. This may be an attack.
        summary: Multiple failed connections to SSH-Server
      isPaused: false
```