Automatisierte Erzeugung von Korrelationsregeln - Unterstützung der Arbeit von Sicherheitsanalysten in einem SOC

Arne Thiele
Department Informatik
Hochschule für angewandte Wissenschaften Hamburg
Hamburg, Germany
arne.thiele@haw-hamburg.de

***Zusammenfassung*—In dieser Ausarbeitung werden Funktionsweisen von Security Information and Event Management Systemen erläutert und in den Zusammenhang mit Security Operation Centers eingeordnet. Es wird auf die Arbeitsweise mit solchen Systemen eingegangen und die Wichtigkeit einzelner Bearbeitungsaspekte hervorgehoben. Ein essentieller Bestandteil dieser Systeme ist die Event-, bzw. Alarm-Korrelation. Im Zuge dessen werden verschiedene Ansätze und Algorithmen erläutert. Es wird eine in bisherigen Arbeiten gebildete Verarbeitungskette von Alarmen dargestellt. Auf Basis der erläuterten Vorteile von Korrelationsalgorithmen wird dann eine Erweiterungsmöglichkeit dieser Verarbeitungskette vorgestellt. Diese dient abschließend als Ausrichtung für zukünftige Projektarbeiten.**

***Index Terms*—Intrusion Detection, IT-Security, SIEM, Security Information and Event Management Systems, Security Operation Center, SOC, Alarm Correlation**

I. Einführung

Die zunehmende Vernetzung informationstechnischer Systeme (IT-Systeme) und die Verlagerung wirtschaftlicher und privater Werte und Prozesse auf solche Systeme hat in den letzten Jahren stark zugenommen [1]. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) veröffentlicht jährlich eine Bestandsaufnahme der aktuellen Bedrohungen und Angriffe. In der aktuellen Veröffentlichung, die das Jahr 2019 beschreibt, wird hervorgehoben dass der Schwerpunkt der bekannten Cyber-Attacken im Bereich der Cyber-Kriminalität liegt und damit meistens finanziell motiviert ist. Als Beispiel wird ein norwegischer Aluminiumlieferant genannt, welcher Opfer eines Ransomware-Angriffs wurde. Im Falle eines solchen Angriffs werden vermeintlich wichtige Daten vom Angreifer verschlüsselt und im Anschluss eine Lösegeldforderung an die Opfer gestellt. Im genannten Beispiel wurde so ein Großteil des automatisierten Betriebs des Aluminiumlieferanten stark beeinträchtigt, wodurch die Produktion weitestgehend auf manuellen Betrieb umgestellt werden musste. [2]
Weiterhin wird das Risiko beschrieben, dass Endgeräte Teil eines Bot-Netzes werden („Verbünde von Rechnern oder Systemen, die von einem fernsteuerbaren Schadprogramm [Bot] befallen sind“ [2]). Dieser Sachverhalt wird im weiteren Verlauf dieser Ausarbeitung in Sektion II erneut aufgegriffen.
Die Anzahl an erfolgreichen Angriffen zeigt, dass ausschließlich präventive Maßnahmen nicht mehr ausreichen um einen umfassenden Schutz zu gewährleisten. Zusätzlich werden verbesserte Erkennungssysteme benötigt, um darauf aufsetzend reaktive Maßnahmen zu begünstigen [1]. Solche Erkennungssysteme (Intrusion Detection Systems [IDS]) versprechen die frühzeitige Erkennung von Sicherheitsverletzungen und Angriffen. Und ermöglichen somit ein schnelles Reaktionsvermögen um potentielle Schäden frühzeitig einzudämmen.
Diese Ausarbeitung ist im Weiteren wie folgt gegliedert: Sektion II beschäftigt sich mit relevanten Grundlagen auf dem Gebiet Security Information and Event Management und leitet in die Positionierung von Alarm-Korrelation in der Verarbeitung von IDS-Alarmen ein. Anschließend wird in Abschnitt III die in früheren Arbeiten entworfene und eingeführte Verarbeitungskette erläutert. Die Sektion IV stellt drei unterschiedliche Klassen von Korrelationsalgorithmen vor. In Abschnitt V wird eine Erweiterungsmöglichkeit der in Abschnitt III aufgezeigten Verarbeitungskette vorgestellt. Sektion VI fasst diese Ausarbeitung zusammen und abschließend wird in Abschnitt VII ein

Ausblick auf mögliche Experimente und zukünftige Arbeiten gegeben.

II. Grundlagen

In den folgenden Abschnitten werden die relevanten Grundlagen auf dem Gebiet Security Information and Event Management erläutert. Dabei geht es um die praxisnahe Darstellung, wie mit von der Intrusion Detection Infrastruktur geschalteten Alarmen umgegangen wird. Um dies anschaulich darzustellen, wird schrittweise das Abstraktionslevel der Betrachtung angehoben.

A. Intrusion Detection Systems

Intrusion Detection Systems (IDS) gibt es in einer großen Vielfalt. Dies lässt sich durch die individuellen Anforderungen an die Systeme begründen, welche sehr stark vom Kontext des eingesetzten Systems abhängen. Dabei spielen Faktoren wie bspw. die Domäne, die bestehende Infrastruktur und die damit verbundenen zu schützenden Komponenten oder auch die finanzielle Situation des Unternehmens oder der Organisation, welche eine IDS-Infrastruktur einführen möchte, eine Rolle [3]. IDS können anhand ihrer Platzierung im Netzwerk in Host-based Intrusion Detection Systems (HIDS) und Network-based Intrusion Detection Systems (NIDS) unterteilt werden (siehe Abschnitte II-B und II-C). Zusätzlich gibt es noch Hybride Ansätze, welche NIDS und HIDS vereinen, sowie multi-agenten Systeme. Unabhängig von dieser Unterteilung arbeiten diese Systeme im wesentlich anhand zweier Vorgehensweisen:

1) Anomalieerkennung (engl. Anomaly detection) [1], [4]–[6]
2) Fehlgebrauchserkennung (engl. Misuse detection) [1], [4]–[6]

Bei der Anomalieerkennung wird das Verhalten des zu schützenden Rechners oder Netzwerks beobachtet und überwacht. Dabei wird ein Modell erstellt, welches das „normale Verhalten“ beschreibt. Abweichungen davon werden entsprechend gemeldet [5]. Die Fehlgebrauchserkennung geht genau anders herum vor. Dabei werden anhand von Regeln bestimmte Aktionen definiert, welche als Sicherheitsverstoß gelten sollen. Der definierte Regelsatz kategorisiert in einer bestimmten Reihenfolge auftretende Ereignisse als sicherheitsrelevant. Bei einem detektierten Verstoß wird dieser gemeldet [5], [7].
Die Unterschiede der beiden Erkennungsmethodiken liegen darin, dass die Anomalieerkennung versucht Abweichungen von bekanntem, normalen Verhalten zu ermitteln. Die Fehlgebrauchserkennung hingegen versucht bekanntes Fehlverhalten anhand der zugrundeliegenden Regeln zu detektieren [6]. Letztere lässt sich anhand der zugrunde liegenden Sicherheitspolitik intuitiver umsetzen, hat allerdings den Nachteil einer „closed world assumption“, sodass unbekannte Angriffe demnach nicht erkannt werden, da sie noch nicht als Regel modelliert wurden. Andererseits kann es dazu kommen, dass die Modelle der Anomalieerkennung bereits schädliches Verhalten als „normal“ klassifizieren, sofern zur Erstellungszeit bereits schädliches Verhalten oder kompromittierte Komponenten in der zu überwachenden Infrastruktur vorhanden sind.
Im Folgenden werden HIDS und NIDS jeweils genauer betrachtet und einige Produktbeispiele gegeben.

B. Host-based Intrusion Detection Systems

Host-based Intrusion Detection Systems überwachen einzelne Systemkomponenten oder Rechner. Hierbei werden meist verschiedene Quellen genutzt, um eine Menge von Ereignissen zu analysieren. Quellen können hierbei bspw. die System-Logs oder die Logs der Host-Firewall des überwachten Systems sein. Aus diesen Logs werden Events extrahiert, welche mittels der gewählten Strategie (Anomalie- oder Fehlgebrauchserkennung, siehe Abschnitt II-A) analysiert werden. Die daraus entstehenden Meldungen (auch *Alarme*) werden je nach Sicherheitsinfrastruktur entweder direkt an einen Sicherheitsanalysten oder an ein Security Information and Event Management System (SIEM, siehe Abschnitt II-D) weitergeleitet [5], [7].
HIDS verfügen über sehr Host-spezifische Informationen, auf die andere Sicherheitsmechanismen (wie bspw. NIDS, siehe Abschnitt II-C) keinen Zugriff haben. Allerdings sind HIDS auch sehr auf diese Informationen spezialisiert, weshalb sie auch keine Informationen über Aktivitäten außerhalb ihres betrachteten Systems haben. Gängige Software-Lösungen sind bspw. OSSEC [8], welches auch gleichzeitig in die Kategorie der Multiagent IDS fällt, der Security Event Manager von solarwinds, welcher bei genauerer Betrachtung bereits Funktionalitäten eines SIEMs bereitstellt [9], oder Snort, welches zusätzlich ein Intrusion Prevention System darstellt [10].

C. Network-based Intrusion Detection Systems

Network-based Intrusion Detection Systems (NIDS) überwachen im Gegensatz zu HIDS Netzwerke und keine Systeme [4]. Hierzu wird der Netzwerkverkehr betrachtet und analysiert. Von besonderem Interesse sind hierbei die Attribute des (Sender-IP, Empfänger-IP, Paketgröße, Zeit und Inhalt)-Tupels [11]. Auch in diesem Fall kann

dann anhand der gewählten Vorgehensweise (Anomalie- oder Fehlgebrauchserkennung, siehe Abschnitt II-A) entschieden werden, ob sicherheitsrelevante Informationen vorliegen und dementsprechend ein Alarm geschaltet werden muss [7].
Produkte dieser Kategorie sind bspw. das „Artificial Immune System“ von Darktrace [12] und Open-Source Projekte wie Zeek (ehemals Bro IDS) [13] oder Suricata [14].

D. Security Information and Event Management Systems

Security Information and Event Management Systems (folgend SIEM-Systeme genannt) bilden einen zentralisierten Sammelpunkt für Daten- und Event-Quellen, welche sicherheitsrelevante Informationen zur Verfügung stellen. Datenquellen können etwa Logging-Systeme, HIDS, NIDS, Firewalls, etc. sein. Die gesammelten Daten werden gespeichert, verwaltet und zu Events korreliert. Anhand dieser Events können Alarme für die zuständigen Sicherheitsanalysten geschaltet werden [7], [15]. Der Entstehung von SIEM-Systemen liegt der Wachstum der Diversität von Datenquellen zugrunde. Diese Datenquellen beinhalten jeweils spezifische Nutzerschnittstellen und produzieren divers geartete Alarme. Zusätzlich zu dieser Zusammenfassung und Abstraktion der Daten der unterschiedlichen Quellen, bieten SIEM-Systeme ein deutlich größeres Kontext-Wissen als die einzelnen Sensoren für sich betrachtet (wie bspw. HIDS). Dieses Wissen schafft zusammen mit geeigneten Visualisierungen einen verbesserten Überblick über Geschehnisse im betrachteten Kontext der Infrastruktur, sowie die Möglichkeit durch Event-Korrelation weitere Einsichten zu erlangen und Fehlalarme zu reduzieren. Aufgrund der genannten Eigenschaften können SIEM-Systeme auch für post-hoc Forensik genutzt werden und so ggf. auch langsame, bzw. stille Angriffe detektieren [7], [15]. Solche Angriffe schaffen es die IDS-Infrastruktur zu durchdringen, ohne dass ein Alarm geschaltet wird. Ein Beispiel dafür sind „Evasion-Attacks“, welche das Wissen über die Arbeitsweise der verwendeten IDS ausnutzen um nicht entdeckt zu werden [16].
SIEM-Systeme haben eine hohe Leistungsanforderung an die verwendete Hardware (oder an die der Installation zugrunde liegenden virtuellen Maschinen). Dies ist dadurch zu begründen, dass diese Systeme in einem sicherheitskritischen Bereich tätig sind, in welchem die Detektionszeit von Angriffen ausschlaggebend sein kann, um mögliche Schäden einzudämmen. Um dies gewährleisten zu können, muss die Event-Korrelation (siehe auch Abschnitt IV) in einer (je nach Einsatzszenario des Systems) angemessenen Zeit durchgeführt werden [7], [15].

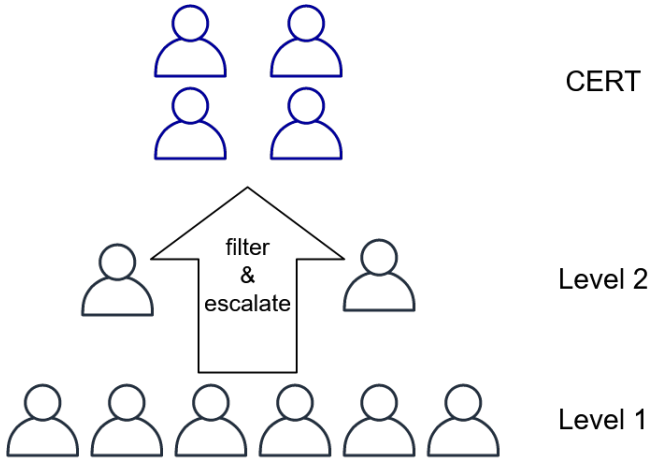

Abbildung 1. Hierarchische Anordnung von Analysten in einem SOC

E. Computer Security Incident Response Teams

Computer Security Incident Response Teams (CSIRTs) erfassen, beurteilen und reagieren auf detektierte Sicherheitsvorfälle oder Berichte dieser [15], [17]. CSIRTs sind Experten-Teams aus erfahrenen Analysten mit einer festen Zugehörigkeit zu einer Organisation, Regierung, Bildungseinrichtung, Region oder zu einem Land [15], [17]. Um gewonnene Erfahrungen auszutauschen und so mehr Nutzen aus dem Wissen einzelner CSIRTs ziehen zu können, bestehen Foren und Kommunikationsnetze für etablierte CSIRTs [17]. Ein zusätzlicher Vorteil dieser Kommunikationsnetze ist, dass im Falle eines regionalen Vorfalls eine schnelle Wissensverbreitung gewährleistet ist, sodass andere Betroffene gewarnt werden können [7], [18].
CSIRTs sind nicht ausschließlich auf die genannten Aufgaben beschränkt. Neben der Überwachung der ihnen zugewiesenen Infrastruktur können diese Teams auch Workshops und Schulungen für das Personal anbieten. Bspw. wird so versucht die Mitarbeiter für Anzeichen eines Angriffs zu sensibilisieren (engl. *awareness*) und ein möglichst flächendeckendes Verständnis von IT-Sicherheit zu vermitteln [7], [17], [18].

F. Security Operations Center

Damit CSIRTs ihrer Hauptaufgabe nachkommen können und sich mit möglichst wenigen Fehlalarmen beschäftigen müssen, arbeiten diese Teams häufig nicht

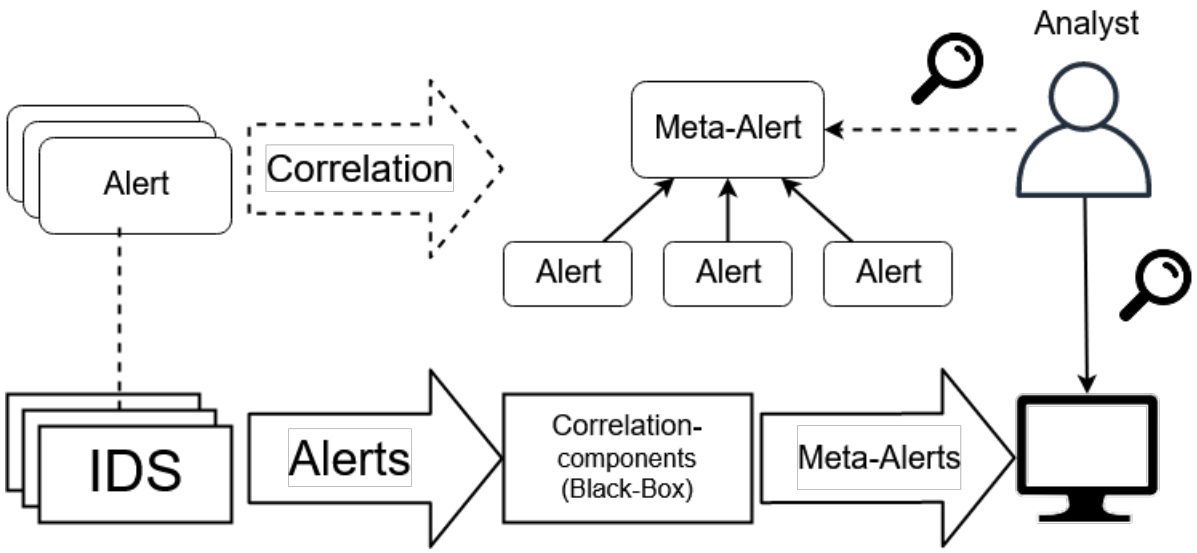

Abbildung 2. Bildung von Meta-Alarmen im Zuge der Korrelation innerhalb eines SIEM-Systems unter Berücksichtigung der Arbeitsweise von SOC-Analysten

direkt mit einem SIEM-System. Deshalb wird bspw. in größeren Unternehmen eine Verarbeitungsstruktur um das eigentliche SIEM-System gebildet. Dieser sog. Security Operations Center (SOC) ist eine meist zentralisierte Verarbeitungsstruktur von eingehenden Alarmen. Der Aufbau eines solchen SOCs ist hierarchisch gegliedert. Dabei werden Sicherheitsanalysten in verschiedene Stufen eingeteilt. Diese Unterteilung erfolgt anhand der Erfahrung des jeweiligen Sicherheitsanalysten. Je mehr Erfahrung oder Spezialisierung ein Sicherheitsanalyst vorweisen kann, desto höher ist die Stufe auf welcher er agiert. Dabei nimmt die Anzahl an Analysten auf höheren Stufen ab und endet bei einem CSIRT [15]. Die im Folgenden erläuterte hierarchische Anordnung kann der Abbilding 1 entnommen werden. Die Abbildung zeigt zwei Hierarchiestufen innerhalb des SOCs und die Integration eines CERT[1] auf der dritten Hierarchiestufe. Der Abbildung 2 kann die Arbeitsweise von SOC-Analysten entnommen werden. Dabei stellt der obere horizontale Verlauf die Erstellung von Meta-Alarmen (hier *Meta-Alerts*) dar. Der untere horizontale Verlauf zeigt eine vereinfachte „Black-Box-Betrachtung“ des Korrelationsprozesses.

Sicherheitsanalysten der Stufe 1: Die Hauptaufgabe von Analysten dieser Stufe ist es, eine Vorsortierung der eingehenden Alarme vorzunehmen. Dabei soll ein möglichst hoher Durchsatz erzielt werden, bei welchem eindeutige Fehlalarme aussortiert werden. Kann der Analyst nicht entscheiden ob es sich um einen Fehlalarm handelt oder sogar sicher sagen, dass es sich um einen potentiellen Sicherheitsvorfall handelt, so wird dieser Alarm eskaliert und von einem Analysten höherer Stufe bearbeitet. Es kann auch dazu beigetragen werden Regeln eines SIEM-Systems zu justieren, sodass diese weniger Fehlalarme erzeugt. So kann die auslösende Regel an einen SOC-Engineer gemeldet werden, welcher die Schwellenwerte entsprechend anpasst [7], [15].

Sicherheitsanalysten der Stufe 2: Auf dieser Stufe werden Alarme bearbeitet, die aus der Stufe 1 zur weiteren Analyse übergeben wurden. Analysten dieser Stufe können anhand weiterer Quellen (bspw. Threat Activity Alerts, Bedrohungsanalysen und damit zusammenhängende Alarme von öffentlichen Organisationen oder interne System-Logs) ein größeres Kontextwissen aufbauen um die zu bearbeitenden Alarme genauer zu untersuchen und zu bewerten. Auf diese Weise erkannte Sicherheitsvorfälle werden mit allen damit zusammenhängenden Daten an die Stufe 3 weitergeleitet [7], [15].

Spurensicherung und Security Engineers (Sicherheitsanalysten der Stufe 3 und höher): Auf dieser Ebene werden Analysen über den Umfang und die Auswirkungen des detektierten Sicherheitsvorfalls durchgeführt und weitere Maßnahmen eingeleitet, um einem ggf. noch aktiven Angriff entgegenzuwirken. Hierbei wird das zuständige CSIRT in die Arbeit eingebunden [7], [15], [18].

[1]CERT ist ein geschützter Begriff für bestimmte CSIRTs und wird in diesem Fall synonym verwendet.

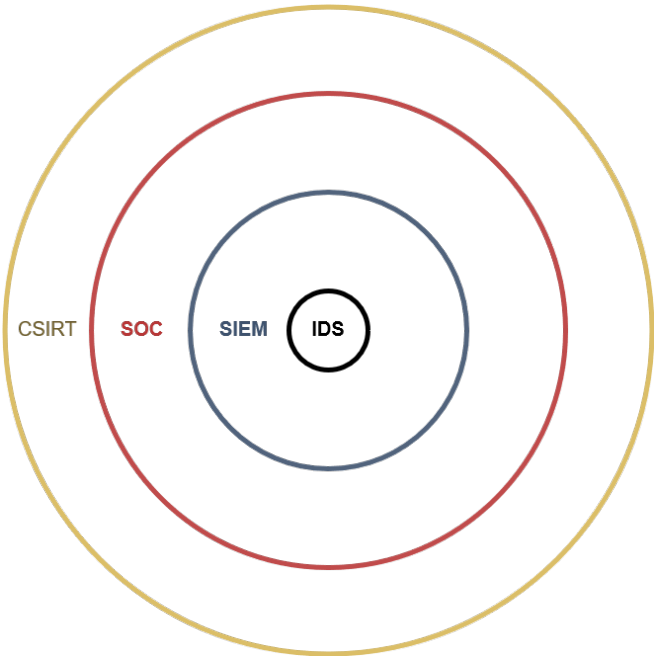

Abbildung 3. Übersicht der um die IDS-Infrastruktur gruppierten Parteien im Umgang mit IDS-Alarmen

SOCs sind darauf ausgelegt, einen möglichst hohen Durchsatz in der Bewertung von geschalteten Alarmen zu erlangen. Das liegt unter anderem daran, dass sehr viele Alarme (je nach Größe des SOCs und der überwachten Infrastruktur) geschaltet werden. Selbst wenn nur wenige Fehlalarme (false-positives) auftreten, hat ein SOC oftmals eine hohe Personalauslastung. Des Weiteren sind SOCs eher auf die Erkennung von Sicherheitsvorfällen, als auf die tiefgreifende Analyse solcher ausgelegt. Diese wird im Falle eines Sicherheitsvorfalls an das zuständige CSIRT übergeben oder im Falle eines größeren SOCs durch die oben genannten Stufen der Sicherheitsanalysten abgedeckt [7], [15], [18].
Eine Übersicht über die Anordnung der genannten Komponenten und Parteien kann der Abbildung 3 entnommen werden. Dabei ist zu beachten, dass CSIRTs ggf. mit mehreren SOCs zusammenarbeiten.

III. Bisher untersuchte Verarbeitungskette

Die bisherige Umsetzung im Zuge vorangegangener Ausarbeitungen und Projekte basiert auf der Korrelationskette nach Valeur et al. [19]. Diese beschreibt, wie eine vollständige Verarbeitungskette aussehen kann und wie den später (in Abschnitt IV) aufgegriffenen Anforderungen, bzw. aktuellen Problemstellungen begegnet werden kann. Die einzelnen Schritte der Kette sind in Abbildung 4 dargestellt. Die dort gezeigten Korrelationsschritte adressieren verschiedene Problemstellungen. Bswp. wird die Heterogenität der geschalteten Alarme durch einen Normalisierungsschritt (*Normalization*) in ein einheitliches Format gebracht, um weitere Verarbeitungen zu ermöglichen. Weiterhin wird der Herausforderung von redundanten Sensoren begegnet, welche auf Basis der gleichen (vermeintlich) schädlichen Aktion Alarme generieren. Dies wird von dem Fusionsschritt (*Alert Fusion*) abgefangen. Nachdem dies geschehen ist, wird versucht zu verifizieren, ob der Alarm überwachte Systeme betrifft und ob diese die für einen solchen Angriff notwendigen Vorbedingungen erfüllen (*Alert Verification*). Die weiteren Schritte dienen einer detaillierten Zusammenfassung des Angriffshergangs (*Thread Reconstruction* bis *Focus Recognition*) und der Erkennung von mehrstufigen Angriffen (*Multi-Step Correlation*). Abschließend wird versucht zu analysieren welche Systeme der überwachten Infrastruktur betroffen sind (*Impact Analysis*) und dem so entstandenen Meta-Alarm eine Priorität zuzuweisen (*Prioritization*) [19].
In Abbildung 5 ist die bisherige Umsetzung einer Verarbeitungskette von Alarmen skizziert. Dabei ist zu beachten, dass die Verarbeitungskette nach [19] nicht vollständig implementiert wird. Die dort abgebildete Verarbeitungskette basiert auf [18] und ist eine Abänderung und Erweiterung der Architektur des Open-Source Projekts „SIEM-Monster“ [20] unter Verwendung einer Untermenge der dort verwendeten Produktkomponenten. Die Abbildung zeigt eine beispielhafte Verarbeitung der Alarme zweier IDS-Systeme. Dabei wurden Regeln im Logstash-Indexer hinterlegt, die die geschalteten Alarmtypen normalisieren und in Elastic-Search einspielen. Index A und B stehen dabei für die Speicher der jeweils nicht normalisierten Alarme der IDS-Systeme. Dies war notwendig um die Alarme während der Einführung des Systems unabhängig von der Normalisierung und der späteren Fusionskomponente untersuchen zu können. Index C dient als Speicher für die normalisierten Alarme beider Systeme und als Ansatzpunkt der Fusionskomponente, welche die Verarbeiteten Alarme in den Index D verlagert. Auf dem Index D wiederum setzt ein Regelsatz zur Filterung von Alarmen an. Dieser Regelsatz D dient unter Anderem zum Herausfiltern von Alarmen unterhalb eines Prioritätswerts, welcher von den IDS-Systemen definiert wurde (Threshold). Zusätzlich können anhand dieser Regeln grundlegende Kontextinformationen wie bspw. der Wochentag hinzugezogen werden und so die Priorität des Alarms angepasst werden. Ein Beispiel hierfür wäre eine Verbindung auf eines der überwachten Systeme mittels SSH an einem Feiertag. Anhand des

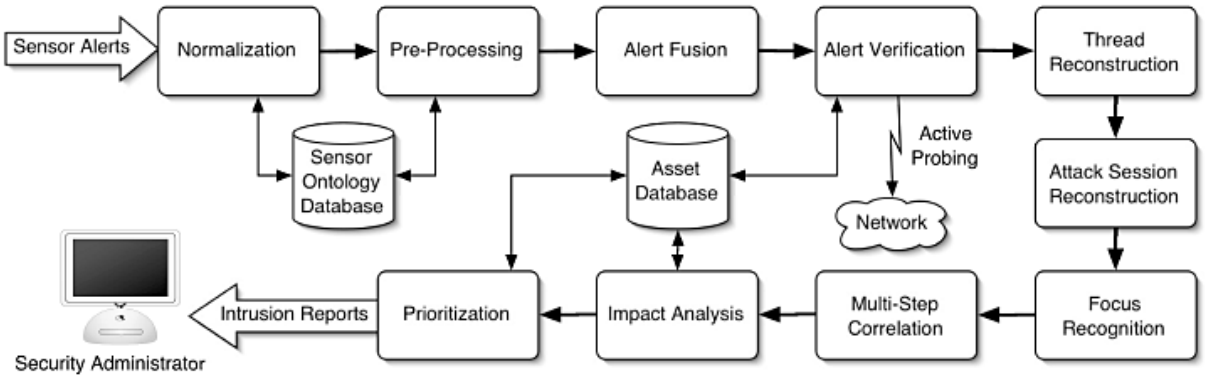

Abbildung 4. Korrelationsschritte nach Valeur et al. [19]

Regelsatzes können Notifikations-Tools wie bspw. „411-Alerting“ eingebunden werden, welche das zuständige Personal bei einer Überschreitung des Thresholds o.Ä. verständigen. Dieses Vorgehen bietet sich an, wenn verhältnismäßig wenig Personal zur Verfügung steht, bspw. in kleinen Unternehmen, welche kein SOC (siehe II-F) zur Verfügung haben.

IV. Alarm Korrelation

Der Einsatz von Intrusion Detection Systemen ist heutzutage weit verbreitet. Allerdings existieren verschiedene Herausforderungen, welche anhand von Alarm Korrelation vermindert oder behoben werden können. Zu diesen Herausforderungen zählen nach [21]:

- **Hohe Anzahl von Alarmen**: Die Anzahl von Alarmen welche von IDS-Infrastrukturen erzeugt werden ist oftmals zu hoch, um vom zuständigen Personal gänzlich überprüft zu werden. Dies kann unterschiedliche Gründe haben. Bspw. können die zugrundeliegenden Mechanismen nicht präzise genug sein. In selteneren Fällen sind die detektierten auffälligen oder unzulässigen Handlungen korrekt erkannt worden, dienen im Vorgehen des Angreifers aber lediglich dazu, die eigentlichen Angriffsziele zu verschleiern und das Personal zu binden.
- **Heterogene Alarmstrukturen**: Die Tatsache, dass unterschiedliche Sensoren auch unterschiedlich geartete Alarme generieren, macht eine Normalisierung der Alarme notwendig, um diese automatisiert verarbeiten zu können (siehe auch III).
- **Falsche Alarme oder nicht erkannte Sicherheitsvorfälle**: Falsche Alarme (engl. *false positives*) sind eine der Ursachen der hohen Anzahl produzierter Alarme. Diese im Nachhinein zu korrigieren ist einer der möglichen Schritte, um die Anzahl der Alarme zu reduzieren. Schwieriger ist dies allerdings bei nicht erkannten Sicherheitsvorfällen, bzw. bei nicht gemeldeten, sicherheitskritischen Aktivitäten (engl. *false negatives*). Diese können im Nachhinein nur anhand von erweitertem Kontextwissen (bspw. dem Wissen über das Angriffsmuster) gemeldet werden. Dies ist allerdings nur möglich, wenn Alarme und Rohdaten für eine gewisse Zeit gespeichert werden.
- **Keine Verkettungsmöglichkeit von Alarmen mit historisierten Alarmen**: Viele Angriffe basieren auf mehreren Angriffsphasen. Diese können sich über lange Zeiträume erstrecken und somit die zeitliche Korrelation umgehen. Zusätzlich sind die Phasen eines Angriffs nicht eindeutig im Sinne des verfolgten Angriffsmusters. So können mehrere Angriffsszenarien die gleiche Phase durchlaufen. Ein großer Vorteil wäre es, wenn genug Wissen über einzelne Schritte und Angriffsmuster bekannt wäre, sodass das zuständige Personal anhand von bereits durchlaufenen Phasen vorhersagen könnte, was der nächste Schritt des Angreifers wäre.
- **Keine Priorisierung von Alarmen oder Informationen über die Verlässlichkeit der Erkennung**: Um Alarmen eine Priorität bzw. Gewichtung zuweisen zu können ist es notwendig, dass Informationen über die Verlässlichkeit der betrachteten Alarme, sowie die Wichtigkeit des betroffenen Systems zur Verfügung stehen. Anhand dieser Informationen kann dann automatisiert eine Priorisierung der geschalteten Alarme stattfinden. Dies würde die Vorgehensweise von Analysten deutlich unterstützen. Allerdings liefern viele IDS keine Information über die Verlässlichkeit der geschalteten Alarme.

In den folgenden Abschnitten werden verschiedene Kor-

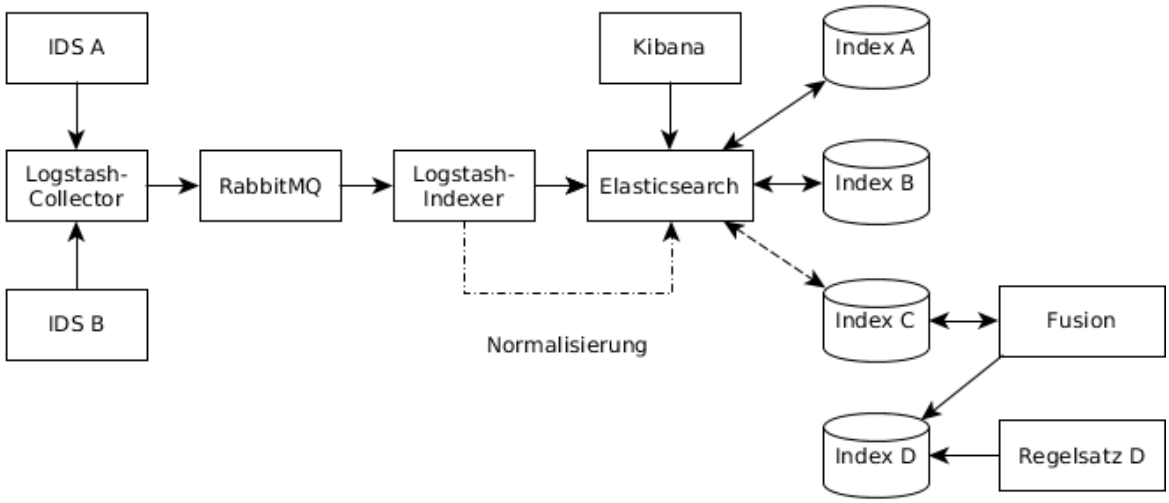

Abbildung 5. Bisher betrachtete beispielhafte Umsetzung der Verarbeitungskette

relationsalgorithmen aufgegriffen und erläutert. Diese lassen sich grob in drei Klassen von Algorithmen unterteilen.

A. Similarity-based algorithms

Algorithmen dieser Kategorie vergleichen Alarme und Meta-Alarme auf definierte oder gelernte Gleichheitsmerkmale. Dabei werden Alarme in einem bestimmten Zeitfenster betrachtet. Diese Algorithmen gruppieren demnach eingehende, ausreichend ähnliche Alarme zu neuen Meta-Alarmen [21].

1) Simple Rules: Der Kerngedanke dieses Ansatzes ist das Verknüpfen verschiedener Alarm-Attribute anhand von abgeschlossenen Regeln. Da diese Regeln nicht zusammenhängend betrachtet werden, können Regelsätze sehr gut parallelisiert werden. Die einzige Voraussetzung ist die Bereitstellung von allen Alarmen (auch bereits zu Meta-Alarmen korrelierten Alarmen) innerhalb eines bestimmten Zeitfensters.
In [22] wird der Begriff „Alert Fusion“ in diesem Zusammenhang verwendet. Die Autoren nehmen als Vorbedingung dieses Schritts eine Art der Normalisierung als gegeben an. Um Diese Art der Korrelation anwenden zu können, müssen die Alarme ein gewisses Maß an Attributgleichheit aufweisen (engl. „Feature Overlapping“). Damit ist gemeint, dass die Alarme eine gleichartige Struktur aufweisen, sodass zumindest einige Attribute in beiden verglichenen Alarmen auftreten. Beispiele für solche Attribute können die involvierten Ports oder die Ausgangs- und Ziel-IP-Adresse sein. Für jedes dieser Attribute wird im Ansatz von [22] eine Regel definiert, welche die Gleichheit der Werte auf einen numerischen Wert zwischen 0 und 1 abbildet. Da dies für jedes Attribut durchgeführt wird, kann anschließend ein Maß der Gleichheit der verglichenen Alarme festgestellt werden. Diese können dann ggf. zu einem Meta-Alarm zusammengefasst werden.

2) Hierarchical Rules: Dieser Ansatz fügt dem der Simple Rules weitere Hierarchieebenen hinzu, um Generalisierungen vorzunehmen [21]. Im Beispiel von [23] werden für bestimmte Attribut-Typen, die in einem Alarm vorhanden sein können verschiedene Generalisierungshierarchien eingeführt. Ein Beispiel dafür ist in Abbildung 6 dargestellt. In der dargestellten Hierarchie werden Wochentage in ihren Kontext eingeordnet. Dabei gibt es einerseits die Generalisierung dahingehend, ob es sich um einen Werktag handelt und andererseits die Einordnung in den aktuellen Monat (Anfang oder Ende). Der in [23] geschilderte Anwendungsfall beschäftigt sich mit der Erkennung von Ursachen für Fehlalarme. Dies geschieht unter der Beobachtung, dass vielen Fehlalarmen die gleiche Ursache zugrunde liegt und die Beseitigung dieser Ursache dazu führt, dass bis zu 87% der Fehlalarme entfernt werden können. Der Ansatz ermöglicht für die Zeit in der die Ursachenbehebung noch nicht durchgeführt wurde, eine Generalisierung (bzw. Zusammenfassung) von Alarmen mit der vermeintlich gleichen Ursache. Des Weiteren kann die Behebung einer Ursache sehr teuer oder unmöglich sein (bspw. wenn kein Zugriff auf das betroffene System möglich ist).

3) Machine Learning: Diese Untergruppe von Algorithmen generiert Vergleichsmerkmale, bzw. erkennt die

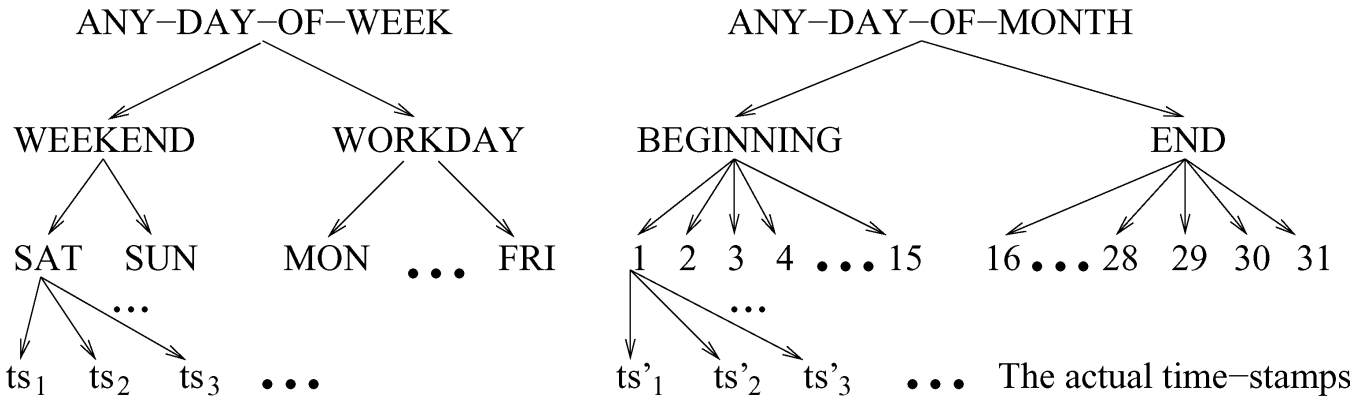

Abbildung 6. Generalisierungshierarchie von Zeit-Attributen nach [23]

Vergleichsmerkmale automatisiert. Diese Kategorie kann in überwachtes (engl. *supervised*) und unüberwachtes (engl. *unsupervised*) Lernen unterteilt werden. Ein Beispiel für überwachtes Lernen sind sog. *decision trees*. Wie alle Algorithmen aus der Kategorie des überwachten Lernens benötigen diese einen Datensatz anhand dessen sie Muster und Zusammenhänge erlernen können. Decision Trees können sowohl für single-step, als auch für multi-step clustering eingesetzt werden. Durch die starke Abhängigkeit von geeigneten Datensätzen kann nicht gewährleistet werden, dass die Algorithmen in ihrer Anwendung korrekt arbeiten falls der zugrunde liegende Datensatz fehlerhaft ist.
Weitere Bestreben auf diesem Gebiet ist der Einsatz neuronaler Netze, sowie ein Algorithmus, welcher lernt ob Alarme richtig oder falsch geschaltet wurden. Dieser Algorithmus bietet die Möglichkeit während seines Einsatzes weiter optimiert zu werden. Allerdings müssen die jeweiligen Alarme durch einen Menschen kategorisiert werden [24].

B. Knowledge-based algorithms

Algorithmen dieser Kategorie setzen auf einer etwas höheren Abstraktionsebene an als die zuvor betrachteten similiarity-based Algorithmen. Dabei wird das Wissen über bestehende Angriffsmuster genutzt um ein Netz von möglichen Angriffsmustern aufzubauen. Somit wird jeder Alarm in ein solches Netz von Meta-Alarmen eingeordnet. Die Grundvoraussetzung ist, dass ein bestimmter Satz von Aktionen (repräsentiert durch aufgezeichnete Alarme) als Vorbereitung oder Bedingungen (engl. *prerequisites*) für einen erfolgreichen Angriff definiert sind. Diese können dann direkt mit möglichen Konsequenzen verknüpft werden, welche ebenfalls in einer Wissensbasis definiert sind. Dieses Vorgehen wird im Folgenden genauer erläutert [21].

1) Prerequisites/Consequences: Die in dieser Unterkategorie befindlichen Algorithmen arbeiten anhand einer Wissensbasis. Diese beschreibt welche Vorbedingungen erfüllt sein müssen, damit ein bestimmter Angriff erfolgreich sein kann (*prerequisites*) und welche Konsequenzen daraus folgen (*consequences*). Diese Algorithmen können für die Erkennung von Angriffen mit verschiedenen Angriffsphasen genutzt werden, unter der Annahme, dass für jeden Alarm Vorbedingungen und Konsequenzen bekannt sind, sowie Wissen über die aktuelle Netzwerkkonfiguration und -struktur existiert. Wenn diese beiden Bedingungen gegeben sind, liefern diese Algorithmen sehr akkurate Ergebnisse und lassen sich in Echtzeit anpassen, da sie ausschließlich anhand der gegebenen Wissensbasis arbeiten. Zusätzlich liefern die Ergebnisse der Algorithmen detaillierten Aufschluss darüber, was genau ein einzelner Alarm im Kontext bedeutet und welche Konsequenzen sich daraus ergeben [21]. In [25] wird die Existenz einer Schwachstelle als Vorbedingung genannt und das Bekanntwerden dieser Schwachstelle als Konsequenz aufgeführt. Diese Zusammenhänge von Vorbedingungen und Konsequenzen können zu einem Graphen zusammengeführt werden, sodass der Angriffsverlauf schnell einsehbar wird. Des Weiteren wird in [25] die Möglichkeit beschrieben, von der zugrunde liegenden IDS-Infrastruktur nicht aufgezeichnete Angriffsschritte zu erkennen.

2) Scenario: Algorithmen dieser Kategorie werden dazu verwendet um mehrschrittige Angriffe (engl. *multi-step attacks*) zu erkennen. Dazu können unter Anderem die in IV-B1 genannten Algorithmen mit definierten Vorbedingungen und Zielen eines Angriffs verwendet werden. In [26] wird die „Correlated Attack Modeling

Language“ (CAML) verwendet, um eine Szenarioerkennung zu ermöglichen. Dabei werden CAML-Module geschrieben, welche drei Bestandteile aufweisen. Die erste Sektion eines solchen Moduls ist die eigentliche Aktivität, die am Ende dazu verwendet wird um das Modul auszulösen. Die zweite Sektion ist ein Satz von Vorbedingungen, die erfüllt sein müssen, damit der Angriff erfolgreich ist (analog zu IV-B1). Die dritte Sektion zeigt die Nachbedingungen, welche im Fall von [26] das Zusammenstellen eines neuen Events ist, welches die Informationen der aufgetretenen Aktivität und der erfüllten Vorbedingungen zusammenfasst und somit eine gewisse Semantik zuweist. Das so erzeugte Event kann dazu führen, dass andere CAML-Module ausgelöst werden und gewährleistet so eine Verkettung und automatisierte Erkennung von Szenarien.

C. Statistical-based algorithms

Statistische Ansätze gehen davon aus, dass relevante Angriffe ähnliche statistische Eigenschaften haben und die Erkennung dieser zu einer akkuraten Erkennung und Einstufung führt. Algorithmen dieser Kategorie speichern Zusammenhänge wie Eintrittsfrequenzen von Angriffen während der Initialisierungsphase, indem Datensätze statistisch untersucht werden. Anschließend werden Angriffsschritte extrahiert. Diese Angriffsschritte werden von einem Sicherheitsanalysten bestätigt oder abgelehnt. Nachdem dieser Schritt abgeschlossen ist, wird das generierte Modell verwendet um Angriffsschritte zu identifizieren. Im Folgenden werden drei Ansätze genauer erläutert [21].

1) Statistical Traffic Estimation: Algorithmen dieser Kategorie arbeiten anhand von Mustern im Auftreten von Alarmen. Anschließend werden wie bei der Anomaliedetektion (siehe II-A) Abweichungen von diesen Mustern erkannt [21].
Bspw. befasst sich der in [27] genannte Ansatz mit Auftrittsfrequenzen von Alarmen. Dabei werden Alarme anhand gewisser Kriterien (ähnlich wie in IV-A erläutert) zu sog. *alert-flows* zusammengefasst. Diese flows werden überwacht und auf Änderungen in ihrer Auftrittsfrequenz überprüft. Auf diese Weise werden Alarme mit niedriger Priorität zusammengefasst und herausgefiltert, sodass nur die Änderung im Verhalten dazu führt, dass Sicherheitsanalysten alarmiert werden. Andere Ansätze erweitern dieses Vorgehen noch um weitere Korrelationsregeln in Abhängigkeit der betrachteten flows [21].

2) Causal Relation Estimation: Die Algorithmen dieser Kategorie lernen kausale Zusammenhänge zwischen dem Auftreten verschiedener Alarme [21]. In [28] wird bspw. anhand dieses Vorgehens versucht den Angriffsplan des Angreifers vorherzusehen. Dies geschieht anhand eines Kausalnetzes bestehend aus Bäumen, welche bestimmte Angriffsszenarien darstellen. Dabei stellt die Wurzel eines Baumes das Ziel des Angriffs dar und Knoten, die keine Blätter sind, stellen Unterziele dar. Das in [28] beschriebene Vorgehen setzt dabei auf bereits teilweise korrelierten Alarmen auf, um auf einem höheren Abstraktionslevel zu arbeiten.

3) Reliability Degree Combination: Wenn Alarme von IDS-Systemen geschaltet werden, beinhalten diese oftmals keine Information darüber, wie zuverlässig (engl. *reliable*) die Information des Alarms ist. Dies kann man auch als Wahrscheinlichkeit eines Fehlalarms oder richtig geschalteten Alarms interpretieren [21].
[29] behandelt das Thema der „Alert-Fusion“. Dabei geht es darum Alarme die von der gleichen Aktion ausgelöst wurden zusammenzufassen (siehe auch III). In diesem Fall soll dann anhand der verschiedenen Alarmquellen evaluiert werden, wie hoch die Wahrscheinlichkeit eines richtig geschalteten Alarms ist und somit eine erste Einschätzung gegeben werden, wie hoch der so entstandene Meta-Alarm zu priorisieren ist [29].

V. Erweiterung der Verarbeitungskette

Diese Sektion erläutert mögliche Erweiterungen der in III erläuterten Verarbeitungskette und Unterstützungen für die Analysten eines SOC. Die folgenden Abschnitte beschäftigen sich mit drei denkbaren Komponenten, welche zu einer Erweiterung des Korrelationsmechanismus beitragen und somit die Arbeit der Analysten vereinfachen. Dabei wird davon ausgegangen, dass es einen Regelsatz gibt, auf dessen Basis Alarm-Korrelation vorgenommen wird.

A. Regelerzeugungskomponente

Die Regelerzeugungskomponente ist dafür zuständig neue Regeln zu erzeugen, welche sich aus bereits verarbeiteten Alarmen herleiten lassen. Auf diese Weise sollen, sofern losgelöst von den beiden noch folgenden Komponenten, Regelverbesserungen unterstützt werden. So können zum Beispiel Fehlalarme reduziert werden, welche dazu führen dass Sicherheitsanalysten stark ausgelastet werden (siehe II-F). Allerdings soll diese Komponente nicht autonom und ohne Prüfung neue Regeln in den bisher verwendeten Regelsatz integrieren. Hierzu ist entweder menschliche Überprüfung notwendig, oder ein Bewertungsmechanismus, welcher den Regelsatz vor und nach Einspielen neuer Regeln bewertet und den

präziseren auswählt. Allerdings würde dies ein hohes Maß an Kontextinformationen verlangen.

B. Simulationskomponente

Die Simulationskomponente spielt verschiedene Regelsätze durch und versucht alle Informationen zusammenzutragen, die ein Sicherheitsanalyst benötigt, um entscheiden zu können welcher der getesteten Regelsätze der geeignetste ist. Problematisch ist hierbei, dass für eine fundierte Bewertung Wissen über die Verteilung von Fehlalarmen zu korrekt geschalteten Alarmen benötigt wird. Dieses Wissen ist allerdings nur bei extra dafür ausgelegten Datensätzen bekannt und im operativen Betrieb eines solchen Systems nicht gegeben. Daher müssen andere Metriken herangezogen werden, um die Qualität der jeweiligen Regelsätze messen zu können.
Denkbar wäre hierbei den Grad der Korrelation anhand eines bekannten Datensatzes zu messen. Dabei werden die Gesamtzahlen der Alarme verglichen und aufbereitet. Für diesen Zweck werden Meta-Alarme, welche eine gewisse Anzahl an Basis-Alarmen zusammenfassen, als ein Alarm gewertet, sodass aus dieser Messung der Grad der Korrelation abgeleitet werden kann. Die Korrektheit muss allerdings immer noch durch einen Sicherheitsanalysten überprüft werden.
Ein großer Vorteil dieser Komponente wäre die Möglichkeit einer Abschätzung der Auswirkungen einzelner Regeln auf den operativen Betrieb. Dies könnte so realisiert werden, dass der Regelsatz nicht auf einen bekannten Datensatz angewendet wird, sondern gespeicherte Real-Daten verwendet werden. Diese wären unter Berücksichtigung der in Abbildung 5 dargestellten Architektur in einem auf den Index D folgenden Speicher einsehbar, da Index D ggf. auf Alarme eines definierten Zeitfensters eingeschränkt ist.

C. Vorschlagskomponente (UI)

Die Vorschlagskomponente stellt die Schnittstelle zum Analysten dar. Diese ist für die Anzeige und Inbetriebnahme der anderen beiden Komponenten zuständig und mit einem User-Interface ausgestattet. Somit führt sie die Simulations- und Regelerzeugungskomponente zusammen. Dem Analysten soll so ermöglicht werden die von der Regelerzeugungskomponente erstellten Regeln einzusehen und für einen angegebenen Zeitraum zu simulieren.
Die Bereitstellung der Funktionen der anderen beiden Komponenten, sowie die übersichtliche Darstellung der Ergebnisse, soll dazu beitragen den Optimierungsprozess der Alarm-Korrelation zu unterstützen.

VI. Zusammenfassung

In dieser Ausarbeitung wurden die Grundlagen der Intrusion Detection erläutert und in den Kontext von Information Security and Event Management eingeordnet. Anschließend wurden die Verarbeitungsschritte aufgezeigt, welche innerhalb eines Security Opterations Center durchlaufen werden und der Bezug zu Computer Security Incident Response Teams hergestellt.
Anschließend wurde auf die in vorherigen Arbeiten untersuchte Verarbeitungskette eingegangen und die einzelnen Arbeitsschritte dieser erläutert. Diese Verarbeitungskette beinhaltet nur grundlegende Funktionen von Alarm Korrelation. Daran anknüpfend wurden verschiedene Klassen von Algorithmen der Alarm Korrelation aufgeführt und erläutert.
Schlussendlich wurden drei Komponenten zur Erweiterung der Verarbeitungskette um weitere Bestandteile der Alarm-Korrelation beschrieben, welche in zukünftigen Arbeiten genauer betrachtet werden (siehe VII). Diese Komponenten dienen zur Teilautomatisierung der Optimierung von Korrelationsregeln. Zusätzlich sollen die Erweiterungen dazu beitragen Analysten eines SOC auf übersichtliche Art und Weise zu unterstützen und so durch die Optimierung der Regeln zu einer Entlastung der Analysten führen.

VII. Ausblick

Die in Sektion III erläuterte, bisher untersuchte Verarbeitungskette kann um die in Sektion V genannten Komponenten erweitert werden. Denkbar wären mehrere Untersuchungen nach Implementierung verschiedener Algorithmen anhand eines IDS-Datensatzes, wie bspw. dem „2000 DAPRA“-Datensatz, welcher auch in [25] verwendet wird. Weitere Beispiele sind die Datensätze der Cyber Grand Challenge von 2016 [30]. Diese Datensätze beinhalten Alarme, zu welchen bekannt ist welche Angriffe verfolgt wurden. So könnten verschiedene Algorithmen auf Performanz und Erkennungsrate, sowie die Reduzierung von Fehlalarmen und Erweiterbarkeit der Erkennung untersucht werden.
Interessant wäre des Weiteren der Austausch der Regelerzeugungskomponente durch andere, nicht direkt regelbasiert arbeitende Algorithmen. Der bisherige Aufbau der Komponenten würde ein online-Verfahren begünstigen, welches anhand der Entscheidungen der Analysten dazulernt. Die weitere Gestaltung der Komponenten wird in den zukünftigen Projektarbeiten des Grund- und Hauptprojekts detaillierter betrachtet und die Implementierung verschiedener Ansätze gegeneinander abgewogen.

Denkbar wären auch Untersuchungen der Skalierbarkeit der Umsetzung, sowie Architektur eines SIEM-Systems auf Basis von verteilten Services unter Nutzung der durch die Hochschule bereitgestellten Cloud-Computing Ressourcen.

LITERATUR

[1] M. Meier, *Intrusion Detection effektiv!: Modellierung und Analyse von Angriffsmustern*. Springer-Verlag, 2007.

[2] Bundesamt für Sicherheit in der Informationstechnik, "Die lage der it-sicherheit in deutschland 2019," October 2019. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2019.pdf?__blob=publicationFile&v=7

[3] L. N. Tidjon, M. Frappier, and A. Mammar, "Intrusion detection systems: A cross-domain overview," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3639–3681, Fourthquarter 2019.

[4] O. Al-Jarrah and A. Arafat, "Network intrusion detection system using attack behavior classification," in *2014 5th International Conference on Information and Communication Systems (ICICS)*, April 2014, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6841978/

[5] Y. Lin, Y. Zhang, and Y. j. Ou, "The design and implementation of host-based intrusion detection system," in *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, April 2010, pp. 595–598. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5453694/

[6] A. Sundaram, "An introduction to intrusion detection," *Crossroads*, vol. 2, no. 4, pp. 3–7, Apr. 1996. [Online]. Available: http://doi.acm.org/10.1145/332159.332161

[7] A. Thiele, "Security information and event management systems," 2019. [Online]. Available: https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2018-gsem/Thiele_A/bericht.pdf

[8] O. P. Team, "Ossec website," 2020. [Online]. Available: https://www.ossec.net/

[9] solarwinds, "Solarwinds security event manager website," 2020. [Online]. Available: https://www.solarwinds.com/security-event-manager

[10] The Snort Team, "Snort website," 2020. [Online]. Available: https://www.snort.org/

[11] R. Heady, G. Luger, A. Maccabe, and M. Sevilla, "The architecture of a network level intrusion detection system," University of New Mexico, Albuquerque, Tech. Rep., 08 1990.

[12] Darktrace, "Darktrace website," 2020. [Online]. Available: https://www.darktrace.com/en/

[13] Zeek Project Team, "The zeek network security manager website," 2020. [Online]. Available: https://www.zeek.org/

[14] Suricata Project Team, "Suricata website," 2020. [Online]. Available: https://suricata-ids.org/

[15] S. Bhatt, P. K. Manadhata, and L. Zomlot, "The operational role of security information and event management systems," *IEEE Security Privacy*, vol. 12, no. 5, pp. 35–41, Sept 2014. [Online]. Available: https://ieeexplore.ieee.org/document/6924640/

[16] D. Wagner and P. Soto, "Mimicry attacks on host-based intrusion detection systems," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ser. CCS '02. New York, NY, USA: ACM, 2002, pp. 255–264. [Online]. Available: http://doi.acm.org/10.1145/586110.586145

[17] G. Killcrece, K.-P. Kossakowski, R. M. Ruefle, and M. T. Zajicek, "State of the practice of computer security incident response teams (csirts)," pp. 11, 20, 2003. [Online]. Available: http://repository.cmu.edu/sei/544/

[18] A. Thiele, "Bachelorthesis - Konfiguration und Evaluation eines Open Source Security Information and Event Management Systems," 2018.

[19] F. Valeur, G. Vigna, C. Kruegel, and R. A. Kemmerer, "Comprehensive approach to intrusion detection alert correlation," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 3, pp. 146–169, July 2004. [Online]. Available: https://ieeexplore.ieee.org/document/1366134/

[20] C. Rock and J. Bycroft, "Siemonster version 3 high level design," May 2018. [Online]. Available: https://dyzz9obi78pm5.cloudfront.net/app/image/id/5af953a3ad121c9c30841d43/n/siemonster-v3-high-level-design-v15.pdf

[21] S. A. Mirheidari, S. Arshad, and R. Jalili, "Alert correlation algorithms: A survey and taxonomy," in *Cyberspace Safety and Security*. Springer, 2013, pp. 183–197.

[22] A. Valdes and K. Skinner, "Probabilistic alert correlation," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2001, pp. 54–68.

[23] K. Julisch, "Clustering intrusion detection alarms to support root cause analysis," *ACM Trans. Inf. Syst. Secur.*, vol. 6, no. 4, p. 443–471, Nov. 2003. [Online]. Available: https://doi.org/10.1145/950191.950192

[24] T. Pietraszek and A. Tanner, "Data mining and machine learning—towards reducing false positives in intrusion detection," *Information security technical report*, vol. 10, no. 3, pp. 169–183, 2005.

[25] P. Ning, D. Xu, C. G. Healey, and R. S. Amant, "Building attack scenarios through integration of complementary alert correlation method." in *NDSS*, vol. 4, 2004, pp. 97–111.

[26] S. Cheung, U. Lindqvist, and M. W. Fong, "Modeling multistep cyber attacks for scenario recognition," in *Proceedings DARPA Information Survivability Conference and Exposition*, vol. 1, April 2003, pp. 284–292 vol.1.

[27] J. Viinikka, H. Debar, L. Mé, A. Lehikoinen, and M. Tarvainen, "Processing intrusion detection alert aggregates with time series modeling," *Information Fusion*, vol. 10, no. 4, pp. 312 – 324, 2009, special Issue on Information Fusion in Computer Security. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253509000189

[28] X. Qin and W. Lee, "Attack plan recognition and prediction using causal networks," in *20th Annual Computer Security Applications Conference*, Dec 2004, pp. 370–379.

[29] G. Gu, A. A. Cárdenas, and W. Lee, "Principled reasoning and practical applications of alert fusion in intrusion detection systems," in *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 136–147. [Online]. Available: https://doi.org/10.1145/1368310.1368332

[30] L. Laboratory, "Cyber grand challenge - datasets," 2016. [Online]. Available: https://www.ll.mit.edu/r-d/datasets/cyber-grand-challenge-datasets