Security Information and Event Management Systems

Arne Thiele
arne.thiele@haw-hamburg.de
Ausarbeitung im Rahmen des Grundseminars im Master Informatik
Hochschule für angewandte Wissenschaften Hamburg
Department Informatik
20099 Hamburg, Germany

28. Februar 2019

Zusammenfassung

In dieser Ausarbeitung werden die Nutzen von Security Information and Event Management Systems herausgestellt. Zusätzlich wird die Erweiterung dieser Systeme durch Event-Korrelation thematisiert, welche eine Reduzierung von zu behandelnden Alarmen und so zu einer Entlastung von Sicherheitsanalysten führen kann.

1 Motivation

Leistungsfähige und sichere Kommunikationssysteme und die Abhängigkeit von diesen in Bezug auf wirtschaftliche Interessen von Unternehmen erhöht implizit die Abhängigkeit von Sicherheitsmechanismen [3]. Dabei spielt das Schützen vor Angriffen und somit die Wahrung der Wirtschaftlichkeit eine entscheidende Rolle für Unternehmen.
Es wird versucht mittels verschiedener Sicherheitsmechanismen auf diese Bedrohungen zu reagieren. Dabei führt die Vielzahl an unterschiedlichen Sensoren zu unübersichtlichen Zuständen und überlastetem IT-Sicherheitspersonal, welches sich mit einer Vielzahl von Alarmen beschäftigen muss. Diese Alarme haben meist ein zu geringes Abstraktionsniveau und können mittels technischer Unterstützung besser gefiltert, zusammengefasst und aufbereitet werden.
Diese Ausarbeitung beschäftigt sich mit Security Information and Event Management-Systemen (SIEM-Systemen), welche eine bessere Übersichtlichkeit über gemeldete Alarme schaffen sollen. Weiterhin soll ein Überblick über generelle Einsatzzwecke geschaffen werden und Möglichkeiten aufgezeigt werden, wie beschäftigte Sicherheitsanalysten oder Computer Security Incident Response Teams (CSIRTs) entlastet werden können.

2 Intrusion Detection Systems

Intrusion Detection Systems (IDS) dienen der Erkennung und Meldung von verdächtigen Aktivitäten oder Angriffen. Im Zuge dessen werden Rohdaten (Logs) ausgewertet und sog. *Events* extrahiert. Diese Events können nach unterschiedlichem Vorgehen bewertet oder zusammengefasst werden. Angriffe oder schadhafte Aktivitäten können demnach anhand dieser Events beschrieben werden. Allerdings können diese unterschiedliche Granularität aufweisen, da die detektierenden Systeme auf unterschiedlichen Ebenen ansetzen. [10] 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 2.1 und 2.2). Unabhängig von dieser Unterteilung arbeiten diese Systeme im wesentlich anhand zweier Vorgehensweisen:

1. Anomalieerkennung (engl. Anomaly detection) [1][6][8]
2. Fehlgebrauchserkennung (engl. Misuse detection) [1][6][8]

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 [6]. 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 [6].
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 [8].

2.1 Host-based Intrusion Detection Systems

Host-based Intrusion Detection Systems überwachen einzelne Rechner. Hierbei werden meist verschiedene Quellen genutzt, um eine Menge von Ereignissen zu analysieren. Quellen können hierbei bspw. die System-Logs, oder die Firewall-Logs des überwachten Systems sein. Aus diesen Logs werden Events extrahiert, welche mittels der gewählten Strategie (Anomalie- oder Fehlgebrauchserkennung, siehe Abschnitt 2) 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 (siehe Abschnitt 3) weitergeleitet [6].
HIDS verfügen über sehr Host-spezifische Informationen, auf die andere Sicherheitsmechanismen (wie bspw. NIDS, siehe Abschnitt 2.2) keinen Zugriff haben. Allerdings sind HIDS auch sehr auf diese Informationen spezialisiert, weshalb sie auch keine Informationen über Aktivitäten außerhalb ih-

res betrachteten Systems haben.

2.2 Network-based Intrusion Detection Systems

Network-based Intrusion Detection Systems (NIDS) überwachen im Gegensatz zu HIDS Netzwerke und keine Systeme [1]. Hierzu wird der Netzwerkverkehr betrachtet und analysiert. Von besonderem Interesse sind hierbei die Attribute des (Sender-IP, Empfänger-IP, Paketgröße, Zeit und Inhalt)-Tupels [4]. Auch in diesem Fall kann dann anhand der gewählten Vorgehensweise (Anomalie- oder Fehlgebrauchserkennung, siehe Abschnitt 2) entschieden werden, ob sicherheitsrelevante Informationen vorliegen und dementsprechend ein Alarm geschaltet werden muss.
Die grundlegende Annahme von NIDS ist allerdings, dass der überprüfte Netzwerkverkehr nicht verschlüsselt ist, damit detailliertere Informationen gewonnen werden können.

3 Security Information and Event Management Systems

Security Information and Event Management Systems (folgend SIEM-Systeme genannt) bilden einen zentralisierten Sammelpunkt für Daten-/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 [2]. 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. IDS). Dieses Wissen schafft zusammen mit geeigneten Visualisierungen einen verbesserten Überblick über Geschehnisse im betrachteten Kontext der Infrastruktur, sowie die Möglichkeit den in Abschnitt 6 geschilderten Prozess auszuführen. Aufgrund der genannten Eigenschaften können SIEM-Systeme auch für post-hoc Forensik genutzt werden und so ggf. auch langsame, bzw. stille Angriffe detektieren [2]. 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 [11].
Abbildung 1 zeigt den Datenimport eines Open Source SIEM-Systems (SIEMonster). Die auf der linken Seite angeordneten virtuellen Maschinen, sind hierbei die Grundlage des Systems und der betriebenen Services zur Annahme, Verarbeitung sowie zur Korrelation von Alarmen und Logs. Diese werden von den Systemen auf der rechten Seite erzeugt und geliefert [7].
SIEM-Systeme haben eine hohe Leistungsanforderung an die verwendete Hardware (oder an die der Installation zugrunde liegenden virtuellen Maschinen). Dies ist da-

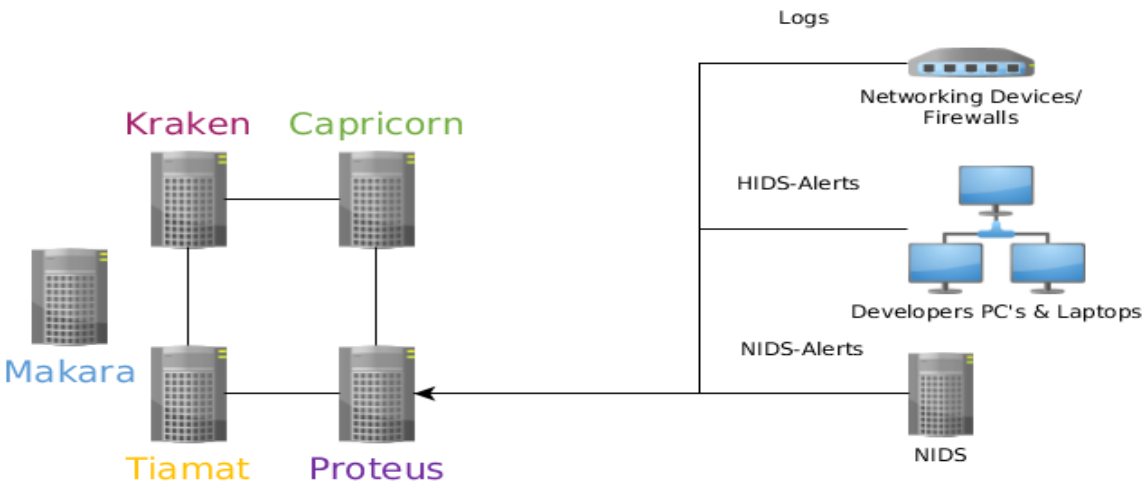

Abbildung 1: Aufbau und Datenfluss eines Open-Source SIEM-Systems (SIEMonster) nach [7]

durch 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-Korrellation (siehe Abschnitt 6) in einer (je nach Einsatzszenario des Systems) angemessenen Zeit durchgeführt werden [2].

4 Computer Security Incident Response Teams

Computer Security Incident Response Teams (CSIRTs) erfassen, beurteilen und reagieren auf detektierte Sicherheitsvorfälle oder Berichte dieser [2][5]. CSIRTs sind Experten-Teams aus erfahrenen Analysten mit einer festen Zugehörigkeit zu einer Organisation, Regierung, Bildungseinrichtung, Region oder zu einem Land [2][5]. Um gewonnene Erfahrungen auszutauschen und so mehr Nutzen aus dem Wissen einzelner CSIRTs ziehen zu können, bestehen Foren und Kommunikationsnetze für etablierte CSIRTs [5]. Ein zusätzlicher Vorteil dieser Kommunikationsnetze ist, dass im Falle eines regionalen Vorfalls eine schnelle Wissensverbreitung gewährleistet ist, sodass andere Betroffene gewarnt werden können [9].

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 [5][9].

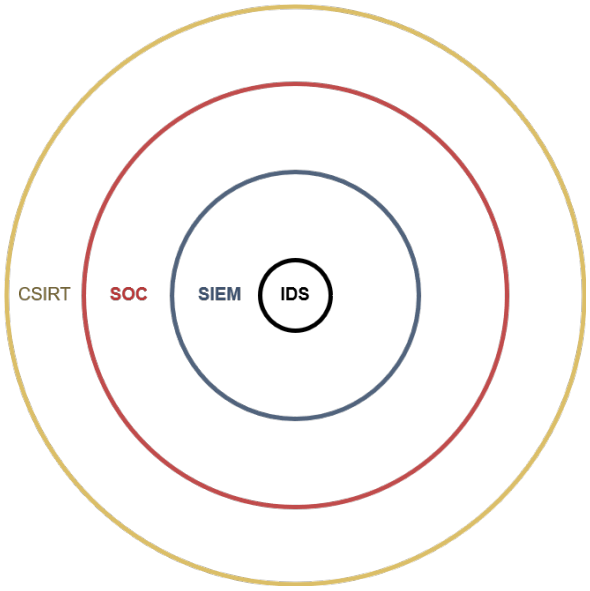

Abbildung 2: Übersicht der genannten Komponenten und Parteien im Umgang mit SIEM-Systemen

5 Security Operations Center

Damit CSIRTs ihrer Hauptaufgabe nachkommen können und sich mit möglichst wenigen false-positives beschäftigen müssen, arbeiten diese Teams häufig nicht 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 [2].

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 Schwellwerte entsprechend anpasst [2].

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 [2].

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 [2][9].
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 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 [2][9].
Eine Übersicht über die Anordnung der genannten Komponenten und Parteien kann der Abbildung 2 entnommen werden.

6 Event Korrelation

Die in einem SIEM-System verarbeiteten Alarme sind oftmals von spezialisierten Sensoren wie HIDS erzeugt worden. Wie in Abschnitt 2.1 erläutert, sind die Informationen solcher Alarme nur auf den jeweiligen Kontext beschränkt. Allerdings können Angriffe an verschiedenen Stellen der Infrastruktur stattfinden und ergeben als einzelne Events im Kontext des jeweiligen Sensors nur Aufschluss über die dort detektierte Aktivität selbst. Ziel der Event-Korrelation ist es, Alarme zusammenzuführen die zum gleichen Angriff gehören. Dadurch wird auch die Anzahl an manuell zu bearbeitenden Alarmen (siehe Abschnitte 4 und 5) reduziert, da einzelne Alarme zu sog. Meta-Alarmen zusammengeführt werden. „Dabei wird eine Baumstruktur gebildet, sodass jeder Meta-Alarm alle unter sich zusammengefassten Alarme als Knoten hält. Auch Meta-Alarme können zu einem weiteren Meta-Alarm zusammengefasst werden. Blätter eines solchen Baumes können allerdings nur Basis-Alarme, also die von IDS oder anderen Sensoren erzeugten Alarme, sein “ [9].
Ein SIEM-System ist nur so leistungsfähig wie die in ihm zusammengeführten Sensoren, sodass weiterhin Fehlalarme auftreten können und so auch verarbeitet bzw. behandelt werden müssen. Zusätzlich können Alarme Informationen über fehlgeschlagene Angriffe enthalten, welche für weitere Untersuchungen nicht relevant sind, da die gegebene Infrastruktur für die gewählte Angriffsstrategie nicht anfällig ist. Auch solche Alarme sollten aussortiert werden, um die zuständigen Analysten in ihrer Aufgabe zu unterstützen [10]. Allerdings sind diese Information ggf. für andere Institutionen oder die in Abschnitt 4 angesprochenen Kommunikationsnetze und Foren interessant und können so mit anderen Teams geteilt werden.
Die im Folgenden erläuterten Korrelationskomponenten können der Abbildung 3 entnommen werden.

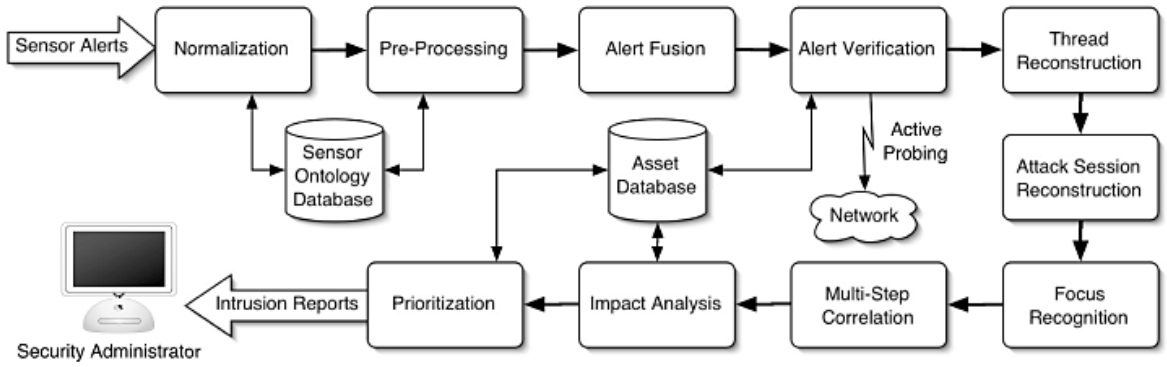

Abbildung 3: Darstellung der Eventkorrelationskomponenten nach Valeur et al [10]

6.1 Normalisierung

Um eine erfolgreiche Verarbeitung zu gewährleisten muss jeder eingehende Alarm in ein einheitliches Format gebracht werden. Die Notwendigkeit dessen wurde bereits in 3 angedeutet, als es um die Entstehung von SIEM-Systemen ging und bleibt auch in diesem Zusammenhang bestehen: Vielfältige Sensoren liefern verschieden kodierte Alarme. Die Eignung eines für diesen Schritt gewählten Formats hängt von der involvierten IDS-Infrastruktur ab. Valeur et al [10] nennen in diesem Zusammenhang das Intrusion Detection Message Exchange Format (IDMEF-Standard) [9][10].

6.2 Vorverarbeitung

Die verwendeten Sensoren der IDS-Infrastruktur liefern neben einer Varianz von unterschiedlich gearteten Formaten auch unterschiedliche Informationen. Deshalb ist es notwendig dass bestimmte Attribute von Alarmen ggf. mit Standardwerten gefüllt werden [10]. Dabei spielen bestimmte Attribute, wie bspw. „[...] start-time, end-time, source, and target“[10] eine wichtige Rolle im Korrelationsprozess und müssen befüllt werden um eine weitere Verarbeitung zu ermöglichen.

6.3 Fusion

Die Fusionskomponente führt Alarme zusammen die von unterschiedlichen IDS-Systemen stammen, sich aber auf den gleichen Angriff beziehen. Dies sorgt gerade dann für eine Verringerung von Alarmen, wenn die IDS-Infrastruktur redundant ausgelegt ist oder Sensoren mit gleichem Zuständigkeitsbereich enthält [10]. Auch hier spielt die Abstraktion von dem in Abschnitt 2.1 genannten lokalen Wissen von Sensoren eine maßgebliche Rolle.

6.4 Verifikation

Um weitere nicht relevante Alarme oder Fehlalarme herauszufiltern kann versucht werden zu validieren ob ein detektierter Angriff erfolgreich war. Dieser Prozess kann sowohl passiv, als auch aktiv sein.
Passive Verifikation arbeitet mit einem Modell von der überwachten Infrastruk-

tur. Dabei wird die erkannte Angriffsstrategie mit dem Verhalten des Modells abgeglichen. Wird festgestellt, dass der Angriff keinen Erfolg haben kann, dann wird der zugehörige Alarm aussortiert. Der Vorteil von passiver Verifikation ist die Geschwindigkeit der Überprüfung, da direkt und ohne weitere Prüfungen mit einem Modell verglichen wird. Der Nachteil ist, dass das Modell sorgfältig gepflegt werden muss und ggf. falsche Ergebnisse liefert wenn neue Services eingeführt werden, oder die Netzwerktopologie verändert wird. Problematisch ist ebenfalls, dass keine zusätzlichen Informationen über das Fehlschlagen des Angriffs gesammelt werden. Wenn ein Angriff auf ein System fehlschlägt kann dies aus verschiedenen Gründen geschehen. Wenn also ein Angriff „zufällig“fehlschlägt, obwohl eine Verwundbarkeit gegenüber der Angriffsstrategie gegeben ist kann dies dazu führen, dass Alarme fälschlicherweise aussortiert werden [9][10]. In diesem Fall spricht man von false-negatives, welche möglichst zu vermeiden sind.

Aktive Verifikation findet anhand direkter Verbindungen zu den Zielsystemen statt. Dabei werden die Zielsysteme auf Anhaltspunkte eines erfolgreichen Angriffs untersucht. Anhaltspunkte können bspw. temporäre Dateien oder Port-Öffnungen in der Host-Firewall sein. Mögliche Teilaspekte von aktiver Verifikation können Port-Scans oder direkte Zugriffe auf das Zielsystem sein. Nachteilig bei diesem Vorgehen ist die erzeugte Netzwerklast, sowie der Konfigurationsaufwand von anderen Sicherheitsmechanismen, welche ggf. von der aktiven Verifikation alarmiert werden (das HIDS des Zielsystems, oder das NIDS, welches den Port-Scan bemerkt). Zusätzlich birgt der Ansatz von direkten Zugriffen auf die Zielsysteme die Problematik der Authentifizierung, bzw. des Schlüsselmanagements. Dies kann eine größere Herausforderung darstellen wenn viele Systeme involviert sind. Daraus folgt, dass diese Komponente selbst ein lukratives Angriffsziel darstellt, da Zugänge auf viele der überwachten Systeme ermöglicht werden [10].
Der Vorteil gegenüber passiver Verifikation ist allerdings, dass mit dem Realzustand der überwachten Infrastruktur gearbeitet wird.

6.5 Bedrohungsrekonstruktion (Threat Reconstruction)

Diese Komponente fasst Alarme zusammen die mutmaßlich vom gleichen Angreifer ausgelöst wurden (bspw. gleiche Ausgangs- und Ziel-IP-Adresse). Dabei wird ein Zeitfenster festgelegt, in welchem solche Alarme zusammengefasst werden sollen. Problematisch ist hierbei, dass Angriffe welche über einen längeren Zeitraum stattfinden (Advanced Persistent Threats) ggf. nicht korrekt erkannt oder zusammengefasst werden [10].

6.6 Angriffsrekonstruktion (Attack Session Reconstruction)

In diesem Schritt wird versucht Alarme einer Angriffs-Session zuzuordnen. Hierzu werden Alarme von verschiedenen Sicherheitsmechanismen miteinander kombiniert und in Relation gesetzt. Die zeitliche Reihenfolge spielt hierbei eine wichtige Rolle, damit ein Angriffsmuster korrekt erkannt

werden kann. Anhand zusätzlicher Informationen über Vorgehensweisen von Angreifern und aktuelle Bedrohungen (bspw. bezogen aus CSIRT-Foren und von anderen Teams, siehe Abschnitt 4) kann ermittelt werden, ob bspw. noch ein weiterer Angriff zu erwarten ist, weil eine Aktivität als Vorbereitung klassifiziert werden konnte [10].

6.7 Fokussierungserkennung (Attack Focus Recognition)

Diese Komponente fasst Alarme zusammen, welche die gleiche Fokussierung aufweisen. Bei einigen Angriffen fokussieren sich viele Systeme auf ein einzelnes Zielsystem. Dabei werden die geschalteten Alarme zu einem „Many2One“ Meta-Alarm zusammengefasst, da sie die gleiche Ziel-IP-Adresse aufweisen.
Ein weiteres Beispiel sind großflächige Netzwerk-Scans, die von einem einzigen System ausgeführt werden. Die so entstehenden Alarme weisen in diesem Fall die gleiche Ausgangs-IP-Adresse auf, sodass sie zu einem „One2Many“ Meta-Alarm zusammengefasst werden können [10].

6.8 Korrelation von mehrstufigen Angriffen

Durch ausreichendes Kontextwissen über Angriffsstrategien können bestimmte Abläufe von Aktivitäten zu einem Meta-Alarm zusammengefasst werden. So können bspw. alle (Meta-)Alarme, die mit einem „Island Hopping“-Angriff zusammenhängen, zu einem Meta-Alarm zusammengefasst werden. Das Vorgehen bei einem solchen Angriff kann wie folgt beschrieben werden:

1. Verwundbarkeits-Scan auf der Suche nach einem Zielsystem
2. Übernahme des ersten Zielsystems
3. Verwundbarkeits-Scan ausgehend vom ersten Zielsystem
4. Angriff weiterer Systeme ausgehend vom ersten übernommenen System, bzw. der anschließend übernommenen Systeme.

Anhand des so entstandenen Meta-Alarms können die betroffenen Systeme ausgelesen werden und bei Bedarf zu einem Graph zusammengeführt werden [10].

6.9 Auswirkungsanalyse

Diese Komponente sammelt Informationen über die Auswirkungen erfolgreicher Angriffe. Dazu werden genaue Informationen über den Zeitpunkt von Ausfällen von Services benötigt, sowie Informationen über Abhängigkeiten von den überwachten Systemen untereinander. Letzteres kann in einer Datenbank gespeichert werden, auf welche immer dann zugegriffen wird, wenn ein Angriff gemeldet wird. So wird anschließend jedes System untersucht, welches in Verbindung zu dem angegriffenen System steht. Die hierbei ermittelten Informationen werden mit in den entstehenden Meta-Alarm integriert. Somit erlangen Sicherheitsanalysten einen Überblick über potentiell betroffene Systeme [10].

6.10 Priorisierung

Um eine sinnvolle Reihenfolge in der Verarbeitung von gemeldeter Alarme zu gewährleisten, werden (Meta-)Alarme priorisiert. Dabei muss beachtet werden, welche Aktivitäten bzw. Systeme für die jeweilige Infrastruktur besonders kritisch sind. So würden Angriffe auf besonders wichtige Systeme dementsprechend höher Priorisiert werden. Zusätzlich können detektierte Alarme aufgrund der zugrundeliegenden Aktivität priorisert werden. So würde bspw. ein Port-Scan weniger kritisch eingestuft werden, als ein Denial of Service Angriff. Dieser Punkt ist allerdings abhängig von der Sicherheitsrichtlinie der jeweiligen Infrastruktur oder Organisation [10].

7 Fazit und Ausblick

SIEM-Systeme sind notwendig um das IT-Sicherheitspersonal zu entlasten, forensische Untersuchungen zu erleichtern und die Sicherheit von Betrieben und Netzwerken zu verstärken. Allerdings ist ein solches System nicht immer sinnvoll und dient eher der Zusammenführung von Sensoren. Dabei ist zu beachten, dass ein SIEM-System immer nur so leistungsfähig sein kann, wie die zugrundeliegende IDS-Infrastruktur. Die Einführung eines solchen Systems ist nur dann sinnvoll, sofern eine solche Infrastruktur existiert und sinnvolle Daten liefert. Die Arbeit des IT-Sicherheitspersonals wird unterstützt und es wird ein besserer Überblick über die im Netzwerk und auf den Rechnern der überwachten Infrastruktur stattfindenden Ereignisse erlangt. Die Einführung ist nicht nur für große Unternehmen relevant, da kleine und mittelständische Unternehmen oftmals wenig IT-Sicherheitspersonal beschäftigen und somit auch hier eine hohe Auslastung besteht (siehe Abschnitt 5).
Im Zuge meiner Bachelorarbeit habe ich ein Open-Source SIEM-System (SIEMonster [7]) bei einem Hamburger Start-Up eingeführt. Bei der Einführung der technischen Komponenten sind mir einige Schwierigkeiten, bzw. Verbesserungsmöglichkeiten des Systems aufgefallen. Dabei habe ich mich insbesondere mit der Erweiterung des Systems um fehlende Korrelationskomponenten, wie der Einführung eines Fusionsprozesses, befasst (siehe Abschnitt 6.3).
Im weiteren Verlauf meines Master-Studiums werde ich mich mit weiteren Open-Source Projekten auseinandersetzen um zusätzliche Realisierungsmöglichkeiten für ein solches SIEM-System zu verstehen. Anschließend wäre eine Projektarbeit denkbar, in welcher ich ein modulares System entwerfe und prototypisch umsetze. Hierbei steht für mich im Vordergrund, dass möglichst viele Funktionalitäten wählbar sind, sodass ein passendes Deployment durchgeführt werden kann, ohne dass unbenutzte Komponenten mit eingespielt werden und so die Übersichtlichkeit einschränken. Ein weiterer Forschungsbereich könnte die Realisierung oder Erweiterung der genannten Projektarbeit um effektive Event-Korrelationskomponenten sein.

Literatur

[1] Al-Jarrah, O. ; Arafat, A.: Network Intrusion Detection System using attack behavior classification. In: *2014 5th International Conference on Information and Communication Systems (ICICS)*, URL https://ieeexplore.ieee.org/abstract/document/6841978/. – Zugriffsdatum: 20.02.2019, April 2014, S. 1–6

[2] Bhatt, S. ; Manadhata, P. K. ; Zomlot, L.: The Operational Role of Security Information and Event Management Systems. In: *IEEE Security Privacy* 12 (2014), Sept, Nr. 5, S. 35–41. – URL https://ieeexplore.ieee.org/document/6924640/. – Zugriffsdatum: 17.02.2019. – ISSN 1540-7993

[3] Bundesamt für Sicherheit in der Informationstechnik: Die Lage der IT-Sicherheit in Deutschland 2017. (2017), August. – URL https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2017.pdf;jsessionid=772AE732C02D5504DCE2FB794E0EC0AA.1_cid369?__blob=publicationFile&v=4. – Zugriffsdatum: 20.02.2019

[4] Heady, Richard ; Luger, George ; Maccabe, Arthur ; Sevilla, Mark: The Architecture of a Network Level Intrusion Detection System / University of New Mexico, Albuquerque. 08 1990. – Forschungsbericht

[5] Killcrece, Georgia ; Kossakowski, Klaus-Peter ; Ruefle, Robin M. ; Zajicek, Mark T.: State of the practice of computer security incident response teams (CSIRTs). (2003), S. 11, 20. – URL http://repository.cmu.edu/sei/544/. – Zugriffsdatum: 20.02.2019

[6] Lin, Y. ; Zhang, Y. ; Ou, Y. j.: The Design and Implementation of Host-Based Intrusion Detection System. In: *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, URL https://ieeexplore.ieee.org/abstract/document/5453694/. – Zugriffsdatum: 20.02.2019, April 2010, S. 595–598

[7] Rock, Chris ; Bycroft, James: *SIEMonster Version 3 High Level Design*. May 2018. – URL https://dyzz9obi78pm5.cloudfront.net/app/image/id/5af953a3ad121c9c30841d43/n/siemonster-v3-high-level-design-v15.pdf. – Zugriffsdatum: 22.02.2019

[8] Sundaram, Aurobindo: An Introduction to Intrusion Detection. In: *Crossroads* 2 (1996), April, Nr. 4, S. 3–7. – URL http://doi.acm.org/10.1145/332159.332161. – ISSN 1528-4972

[9] Thiele, Arne: Bachelorthesis - Konfiguration und Evaluation eines Open Source Security Information and Event Management Systems. (2018)

[10] Valeur, F. ; Vigna, G. ; Kruegel, C. ; Kemmerer, R. A.: Comprehensive approach to intrusion detection

alert correlation. In: *IEEE Transactions on Dependable and Secure Computing* 1 (2004), July, Nr. 3, S. 146–169. – URL https://ieeexplore.ieee.org/document/1366134/. – Zugriffsdatum: 26.02.2019. – ISSN 1545-5971

[11] Wagner, David ; Soto, Paolo: Mimicry Attacks on Host-based Intrusion Detection Systems. In: *Proceedings of the 9th ACM Conference on Computer and Communications Security*. New York, NY, USA : ACM, 2002 (CCS '02), S. 255–264. – URL http://doi.acm.org/10.1145/586110.586145. – Zugriffsdatum: 22.02.2019. – ISBN 1-58113-612-9