A Security Information and Event Management Pattern

MANFRED VIELBERTH, University of Regensburg
GÜNTHER PERNUL, University of Regensburg

In order to achieve a high level of cyber security awareness most mid to large sized companies use Security Information and Event Management (SIEM) embedded into a Security Operations Center. These systems enable the centralized collection and analysis of security relevant information generated by a variety of different systems, to detect advanced threats and to improve reaction time in case of an incident. In this paper, we derive a generic SIEM pattern by analyzing already existing tools on the market, among additional information. Thereby, we adhere to a bottom-up process for pattern identification and authoring. This article can serve as a foundation to understand SIEM in general and support developers of existing or new SIEM systems to increase reusability by defining and identifying general software modules inherent in SIEM.

Categories and Subject Descriptors: D.2.11 [**Software Engineering**]: Software Architectures—*Patterns*; K.6.5 [**Management of Computing and Information Systems**] Security and Protection

General Terms: Security patterns

Additional Key Words and Phrases: SIEM, Security Information and Event Management, Security Analytics

ACM Reference Format:

Vielberth, M. and Pernul, G. 2018. A Security Information and Event Management Pattern. 12th Latin American Conference on Pattern Languages of Programs (SugarLoafPLoP 2018), November 2018, 12 pages.

1. INTRODUCTION

The protection of corporate IT infrastructures against cyber attacks is becoming a more and more demanding task. Trends like Industry 4.0 and Internet of Things transform today's IT-landscapes into a complex and mazy structure with a growing amount of attack points. In most mid to large size companies, a Security Operations Center (SOC) is established to gain a holistic and centralized view on IT security and to enable fast reactions in case of an incident. According to the SANS 2017 Security Operations Center Survey [Crowley 2017], more than 80% of those SOCs are supported by a SIEM system in order to increase IT security awareness. Besides, SIEM systems enable the automation of incident detection and subsequent reactions in order to mitigate imminent damage or to preserve forensic evidence. Therefore, these systems collect security relevant data at a central point, to gain a holistic view of the organizations IT security. Additionally, historic and correlated analyses and further measures are enabled.

Although there are many SIEM systems on the market, there is no pattern to the best of our knowledge, which defines the generic structure of a general SIEM setting. To fill this gap and to create a foundation for understanding the basic components of a SIEM system, we propose a pattern for SIEM in this paper. Additionally this pattern can

This research was partly supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project (https://dingfest.ur.de).
Authors' address: Lehrstuhl für Wirtschaftsinformatik I, Universitätsstraße 31, 93053 Regensburg, Germany; M. Vielberth email: manfred.vielberth@ur.de; G. Pernul email: guenther.pernul@ur.de;

help developers to delimit generic modules. This enables the enhancement of reusability and replaceability of selected modules and improves the general structure and compatibility of SIEM software.

The proposed pattern is deduced in conjunction with the DINGfest project. DINGfest is funded by the Federal Ministry of Education and Research in Germany and aims at developing an architecture for next generation SIEM systems. Especially collecting forensic evidence from virtual environments and leveraging data from Identity and Access Management is emphasized. Furthermore, the project is focusing on improving methods for Visual Security Analytics, by involving experts more efficiently.

One purpose of patterns is to show the current state of best practice for a recurrent problem in a specified context. Thus, it should be used in at least two different use cases [Gamma et al. 2011]. In this paper, we follow this approach and in order to identify the current state of best practice in SIEM we analyze established SIEM systems and related information. In the following, the SIEM pattern is deduced and explained oriented on the POSA (Pattern-Oriented Software Architecture) template [Buschmann et al. 2013].

The intended audience for this pattern includes developers and administrators of SIEM systems. Additionally, it can serve as a foundation for people who are new to this topic in order to gain a basic understanding. We assume our readers to have basic knowledge about UML and a fundamental awareness of IT security issues.

In this paper, we give a theoretical foundation by explaining SIEM in general in Section 2. Subsequent, the derived pattern is described in Section 3. In Section 4, we describe the pattern mining / identification process, which we followed. In Section 5 we finally conclude our work and give hints for future directions, to gain a more detailed view on SIEM. Appended an explanation of domain specific terminology can be found.

2. SECURITY INFORMATION AND EVENT MANAGEMENT

The first notion of Security Information and Event Management is attributed to a report of Gartner Inc. [Williams and Nicolett 2005]. According to this work the expression is composed of two terms: Security Information Management (SIM) and Security Event Management (SEM) [Goldstein et al. 2013]. While SIM deals with centralized management, collection, preservation of historic log data and the generation of reports for compliance purposes, SEM covers threat management, real-time monitoring of security incidents and triggering proper reactions in case of an incident. Thereby, the collected data is aggregated to reduce the amount of data and facilitate the usage for appropriately reacting to security events.

Nevertheless, nowadays' SIEM has evolved from a sole combination of those two technologies to a more holistic and integrated security solution and thus combines the advantages of SIM and SEM in one single centralized system. Due to the numerous functionalities a SIEM has to fulfill, an appropriate academic definition is hard to accomplish. However, the Gartner IT Glossary [Gartner Inc. 2018] outlines the expression quite well. According to their definition, SIEM systems collect relevant data and conduct historic analyses on security events from a broad range of different types of event or contextual data sources. Additionally, it supports reporting for compliance purposes and forensic investigation by analyzing the stored historic data from the same sources. The main functionalities of SIEM are its broad scope on event sources as well as its ability to correlate and analyze these events across heterogeneous sources.

The general purpose of collecting and analyzing such a big amount of data in a central place is the identification of anomalies and incidents, which would not be visible considering data of only one single system or device. Additionally, it facilitates the monitoring and management of the organizations' state of IT security.

However, SIEM is not an isolated software, which runs untouched for a long period of time, but it is in most cases embedded into a Security Operations Center (SOC). A SOC as described by Bhatt et al. [Bhatt et al. 2014] is a centralized organizational unit, with the goal to monitor all relevant events related to security of an IT infrastructure. A SOC basically consists of software, processes and a team of security analysts and experts [Radu 2016]. The

utilized software for collecting data and log files from all relevant assets and for further analyses is a SIEM system. After the detection of a potential incident, the SOC staff determines the measures to be taken, like informing an incident response team or the stakeholders of affected systems. In most cases SOCs are organized hierarchically in multiple layers, whereby the lowest layer depends on a large amount of employees. In large organizations they are organized in multiple shifts for being able to react accordingly to security incidents around the clock.

Due to the growing amount of different sources for log data and other security relevant events, and the subsequent growing need for well trained security analysts, the demand for a higher level of automation rises. Thereby, SIEM can help to automatically analyze and react to security incidents in a centralized manner. It further helps analysts by better visualizing the big amount of data to leverage the expert knowledge of the analysts more efficiently.

Further explanations of relevant terminology is given in appendix A.

3. SIEM PATTERN

3.1 Also Known As

SIEM, SIEM system, (Big Data) Security Analytics System, Cyber Threat Intelligence Tool/System

3.2 Intent

Collect all security relevant data in a central point in order to identify threats or incidents. Therefore, provide the opportunity to deal with different data formats and to analyze the data in real-time and historically. In addition, offer the possibility to interact with human experts.

3.3 Context

In today's organizations, a big number of different security systems and devices are part of IT-infrastructures. In most cases they have been grown historically and thus it is hard to keep track of the overall state of corporate IT-security. Additionally, isolated security systems, like firewalls can only detect or prevent very specific attacks. To gain a holistic picture of the state of security and to uncover more advanced cyber attacks collecting all security relevant information in a centralized system is required. Therefore, real time threat analysis is essential for being able to react quickly, whereby historic analyses have to be performed in addition.

3.4 Problem

The big amount of different security relevant devices and software makes it hard to gain a holistic view of an organizations security. Especially, sophisticated attacks affecting multiple systems (commonly referred to as advanced persistent threats) often remain undetected. Moreover, the tremendous amount of generated log data makes the situation even more challenging, because it complicates security related analyses and makes it more difficult to find appropriate reactions in case of an incident. The forces associated with this problem are as follows.

Forces:

- *Detection rate and false positives:* We want to decrease the manual workload of experts and analysts as much as possible by automatically detecting incidents and anomalies. In order to unburden them, the false positive rate has to be very low, as staff has to analyze the wrongly detected incident and decide for further steps. Additionally, the detection rate should be as high as possible, to provide a high level of security. Every incident that was not detected in time might end up causing more work hours or additional damage.
- *Time of discovery and reaction:* We want to be able to detect and react to an attack as fast as possible, because the time a company needs to detect and react to an attack directly impacts the financial damage [Kaspersky Lab 2016]. In order to mitigate the losses, it is important either to react automatically to an incident or to inform the right people in time, who ideally carry out a reaction plan.

- *Usability:* We need a system, that is easy to use by analysts and experts. The effort to connect new devices and configure the detection of new incidents (e.g. create new detection rules) has to be as easy and as little time consuming as possible. Furthermore, the necessary expert knowledge should be as low as possible.
- *Visual preparation of data and integration of expert knowledge:* In the future, it will not be possible to replace experts who manually analyze the data so quickly. Especially the automatic detection of incidents that have never occurred before is challenging. Thus, visual preparation of the data and enrichment with context data is very important for optimally integrating expert knowledge, not only at the beginning and the end of the analytics process, but also in the middle [Ropinski et al. 2017; Gates and Engle 2013].
- *Degree of automation:* Well trained staff in the security domain is quite rare. However, in big organizations SOCs have to be staffed around the clock. We want a system, that can help to reduce the number of people needed by automating certain steps. On the one hand, the analysis should be further automated and on the other hand, reactions should be triggered or carried out automatically wherever possible.
- *Number of connectible devices:* In order to provide a holistic view on corporate security, we want to exploit all relevant data. Therefore, every device pertinent for security has to be connected. This poses two requirements: First, it must be possible to process a large number of log data. Second, nearly every different kind of device or software has to be able to be connected. Thus, we must be capable of handling many different log formats.
- *Analytics:* For being able to detect and identify attacks, a large variety of events and additional data has to be analyzed. Therefore, the possibility to create rules or other detection mechanisms have to be provided. The analysis mechanism should be able to correlate multiple events, which enables the detection of even more incidents or threats. Thereby, capabilities for automatic as well as manual or human-supported analyses must be offered.
- *Reusability and intelligence sharing:* We want to provide a high degree of reusability. Therefore, the pattern should be split into different modules, which are easy to exchange and to integrate by other similar systems. Thus, they should be loosely coupled and interfaces have to be defined. Additionally, interfaces for sharing analyses results with other organizations should be provided. Moreover, reporting for compliance reasons should be possible as it becomes increasingly important especially for critical infrastructure providers [European Parliament 2016; Congress of the United States of America 2014].
- *Costs:* Costs play an important role for every system, intended for application in a company. Therefore, we want to reduce the time needed for implementation and staff necessary for running and monitoring the software. However, it is hard to make a statement about profitability for most cyber security topics [Weishäupl et al. 2018].

3.5 Solution

Implement a SIEM system that collects, enriches, normalizes and stores all relevant log data in a centralized manner. This system automatically detects and reacts to as many incidents as possible, provides interfaces for reporting those and enables the integration of expert knowledge.

Structure:

The resulting SIEM Pattern can be split into eight components, as shown in Fig. 1. The first module is responsible for *log collection* and for providing the collected *log data* to further modules. The input data can be provided by various *log sources*.

The *enrichment* component uses *context data* for providing additional information for bare *log data* in order to improve further processing by providing *enriched log data*. Various *context data sources* supply the needed data.

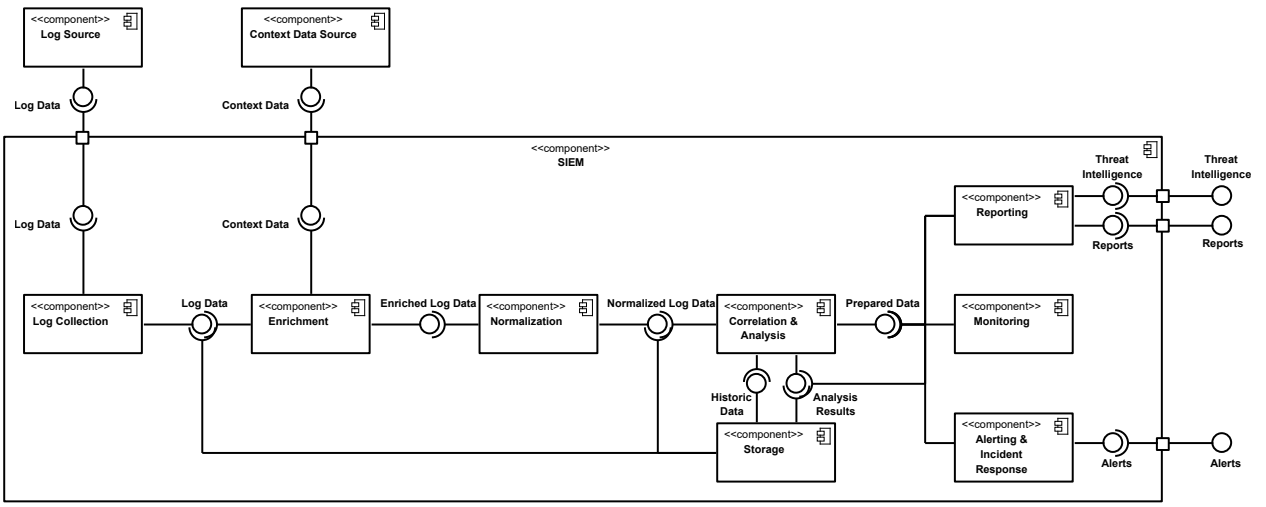

Fig. 1. SIEM Pattern as UML component diagram

For enabling and facilitating further processing, the data is restructured into a standardized format (*normalized log data*) in the *normalization* component.

The main part is the *correlation and analysis* component, as there the incidents are finally recognized based on the incoming *normalized log data*. Additionally, analyses are conducted based on *historic data* provided by the *storage* component. *Analysis results* and raw *log data* are also stored herein. In order to enable further processing by other modules, *prepared data* is provided.

Finally, the findings can either be *reported* or shared with other systems or organizations in form of *threat intelligence*. The *monitoring* component provides an interface for human interaction and the *alerting & incident response* component can propagate *alerts* to responsible stakeholders or react automatically in case of an incident.

3.6 Implementation

In this section we present additional information for implementing the proposed pattern and corresponding components. Additionally, we describe two examples for SIEM systems and show that their pattern fits the one presented in this paper for the most part. Therefore, we first describe a SIEM, which is quite popular in industry and then introduce DINGfest, a SIEM proposed by a research project.

The *log collection* component can either pull the logs from a variety of *log sources* or the log generating source pushes the log data into the *log collection* module. In most cases, the pull based method is implemented by simply accessing the file system of the log generating source by well known standard protocols like FTP (file transfer protocol) or SCP (secure copy). In order to push the logs to the SIEM, the *log source* has to use supported protocols like syslog or a so called agent has to be installed on the source, which is responsible sending logs to the SIEM. Logs can also be integrated as a data stream.

An example for *enrichment* of log data is to resolve the IP address to a geolocation, which can enable the detection of some specific threats. Furthermore, asset discovery tools provide information about systems or sensitive data which requires more protection. The data from vulnerability assessment can help to relate threats to specific data sources. Additionally, enriched data is in most cases much easier to read and interpret by a human analyst in subsequent modules.

The *normalization* component is needed as the various connected log generating sources use many different formats for storing and sending logs in practice. In addition, varying storage technologies are applied. For example, some systems utilize relational databases and others use plain text files or proprietary technologies like syslog. Thus, it is necessary to transform them into a single uniform log format. Additionally, it is much easier for experts to create detection rules for standardized log formats in further steps.

The *correlation & analysis* is conducted in real time on incoming *normalized log data* for recognizing incidents as fast as possible. Additionally, historic analyses are performed for forensic purposes or in order to detect incidents overlooked before, as, for example, the pattern of an attack was not yet known at this time. A multitude of different methods for correlating and analyzing logs exist. However, the most common and very simple approach is detecting incidents on the basis of rules. Such rules can be derived automatically, created by human analysts or retrieved from other systems or organizations. The structure and content of the provided *prepared data* depends on the requirements of consuming components.

Reports can be generated for multiple internal reasons as well as for meeting regulatory compliance obligations. For example, the USA [Congress of the United States of America 2014] and the EU [European Parliament 2016] have laws in place that demand the reporting of information about occurred incidents in certain cases. Second, *threat intelligence* can be exchanged with other SIEM systems. Therefore, threat exchange platforms like the Alien Vault Open Threat Exchange[1] can be applied.

In the *monitoring* component not automatically detected incidents can be recognized by analysts and a decision can be made, whether a recognized threat is a false positive. Additionally, after an incident was detected, experts need additional information for determining further steps in most cases. Furthermore, detection rules of the *correlation and analysis* module can be extended or created manually.

According to the Gartner magic quadrant for security information and event management [Kavanagh and Bussa 2017], IBM QRadar is the most advanced tool. Thus, we compare our proposed pattern to the one used by QRadar. The information therefore can be found in the official documentation [IBM Corporation 2017]. Therein, the QRadar architecture is structured in the layers "data collection", "data processing" and "data searches".

The first layer collects logs ("QRadar Event Collectors") and data streams ("QRadar QFlow Collector") and normalizes them by first parsing and then restructuring the data into a usable format.

The second layer enriches the logs with additional data from sources like the "QRadar Risk Manager", which provides a map of the organizations' network topology and the "QRadar Vulnerability Manager", which identifies security risks in the network. Additionally, the "Custom Rules Engine" analyses the data for incidents and offenses. After the data is analyzed, it is written to the storage, while the "QRadar Incident Forensics" provides historic in-depth investigation by storing collected raw data.

The "data searches" layer provides interfaces for human interaction. Thereby, reports can be created and the user has the possibility to search and analyze the collected data with the help of the "QRadar Console". Additionally, alerts are triggered and presented or forwarded to analysts.

To sum up, IBM QRadar covers all major SIEM modules presented in Fig. 1, although they are named differently with brand specific terms.

The DINGfest project as introduced by [Menges et al. 2018] aims at developing an architecture for next generation SIEM systems. The architecture is divided into the layers "Data Acquisition", "Data Analysis" and "Digital Forensics & Incident Reporting".

The "Data Acquisition" layer provides basic log collection with the help of Logstash[2], an open source tool for managing logs. In addition, virtual machine introspection [Jain et al. 2014] is used in order to monitor a system in a

[1] https://www.alienvault.com/open-threat-exchange

[2] https://www.elastic.co/products/logstash

virtualized environment from the outside without the need for installing an agent on the system. Both collection methods normalize their data into a semi-structured JSON format and push it into a data stream implemented with Apache Kafka[3].

Within the "Data Analysis" layer the collected data gets enriched with information from "Identity Behavior Analytics". Thereby additional details are provided for logs containing user data, such as granted permissions, assigned roles and statistics. Occured incidents can be recognized by the "Event Processing" module by matching pre-defined events stored in a fingerprint database. The "Visual Security Analytics" module provides monitoring and alerting capabilities by presenting occurred incidents and the collected data to the user.

The analysis results and relevant raw data are stored in a so called "IoC (Indicators of Compromise) Vault" in the "Digital Forensics & Incident Reporting" layer. Within this layer incidents can be shared in the STIX[4] format, a representation for cyber threat intelligence.

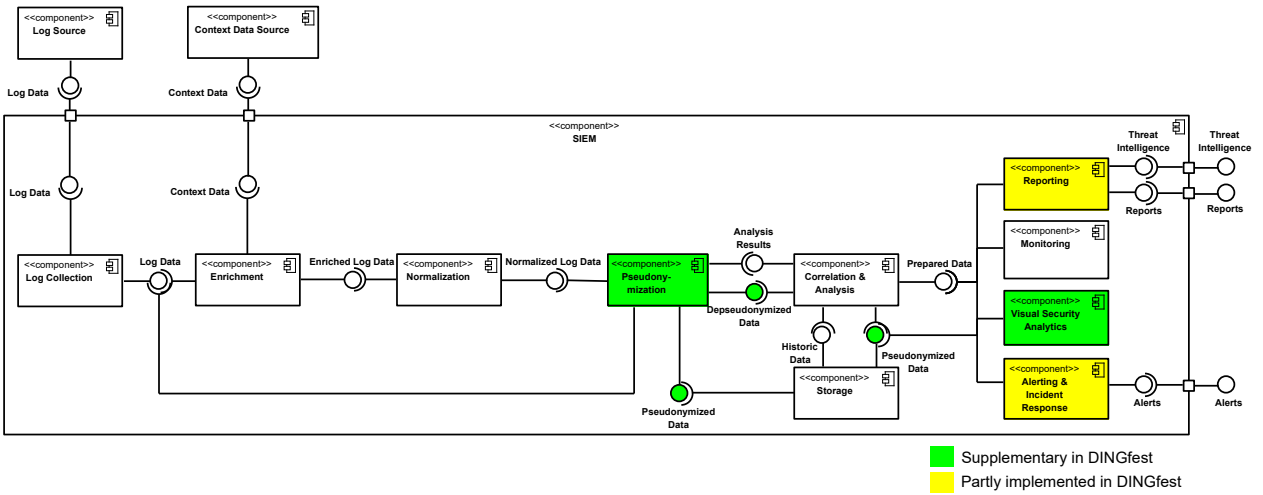

Fig. 2. Pattern of the Dingfest SIEM as UML component diagram

Although DINGfest fits the SIEM pattern shown in Fig. 1, it lacks some capabilities, but at the same time goes beyond certain aspects. The comparison between the DINGfest SIEM and the abstract SIEM pattern is portrayed in Fig. 2 and further elaborated in the following.

First, the ability to generate custom reports for internal purposes is not contained. Second, incident response capabilities are missing, though they could be appended through third party software.

On the contrary, the DINGfest project provides aspects beyond a standard SIEM. In order to preserve forensic evidence it can collect snapshots of a whole virtual machine, which gives an untampered view on the state of a compromised system. Furthermore, experts are involved more closely into the analysis process by applying advanced *visual security analytics* methods. Additionally, *pseudonymization* plays a central role in the DINGfest SIEM for providing a higher level of privacy. Therefore, all data is pseudonymized before it is stored, analyzed or further processed. Only on reasonable grounds of suspicion this data can get depseudonymized.

[3] http://kafka.apache.org/
[4] https://stixproject.github.io/

3.7 Variants

Of course not all current SIEM systems on the market fit into the previously defined pattern with all their parts. Especially the *enrichment* and *normalization* modules are in some cases switched. Thereby, the logs are first normalized and afterwards enriched with relevant context data. However, the input data for the *correlation and analysis* module remains the same.

Furthermore, our research revealed that most SIEM vendors provide the possibility to outsource certain parts in the cloud (e.g. storage) or similar environments. However, the overall structure is mostly not affected thereby.

An additional variant is to extend a SIEM system by active defense capabilities as described by [Docking et al. 2015]. This can improve security by mitigating attacks before they even happen. It also relies on centralized threat intelligence, which can be provided by a SIEM.

3.8 Known Uses

- *IBM QRadar:* IBM QRadar is a modular system and therefore is applicable for medium to large size companies. Furthermore it can be deployed as a standalone system, in a distribute architecture or in the cloud. [IBM Corporation 2018]
  A more detailed comparison of IBM QRadar with our pattern can be found in section 3.6.
- *Splunk:* Splunk Enterprise is the core of their SIEM solution and provides collection and storage of data. It can be extended with different packages, which for example extend it by user behavior analytic capabilities. [Splunk Inc. 2018]
- *LogRhythm:* LogRhythm provides host and network monitoring abilities in addition to core SIEM functionalities. Furthermore, an AI engine aims at automating certain operations. [LogRhythm Inc. 2018]
- *McAfee ESM:* McAfee ESM can be connected with a threat intelligence feed, which provides additional information and signatures. Furthermore, it supports big data technologies like Elasticsearch and Kafka. [McAfee LLC 2018]
- *AlienVault USM & OSSIM:* Compared to the previously mentioned systems, AlienVault OSSIM is distributed as open source software. Thereby, it is the most widely used SIEM system of this kind. AlienVault USM is a paid version of OSSIM hosted in a cloud environment extended by additional features. [AlienVault Inc. 2018]

These examples are just a small selection of SIEM systems on the market fitting our pattern at a large extent. Though, they are in our estimation the best-known ones. A detailed market analysis of SIEM systems can be found in [Kavanagh and Bussa 2017].

3.9 Consequences

The SIEM pattern has the following *advantages*:

- *Detection rate and false positives:* By analyzing the data in a correlated manner, the detection rate should be much better compared to systems, which analyze only data of a single system. Additionally, context data can improve the detection rate even more. In addition, the false positive rate can be lowered by this approach.
- *Time of discovery and reaction:* The proposed solution can improve the time of discovery and reaction significantly, as it is much easier to react to events which are collected centrally instead of reacting to incidents, which could happen in any system located anywhere in the organization. Additionally, our approach enables automatic incident response.
- *Usability:* The usability is improved by providing a central user interface for monitoring different connected systems and by providing an interface for easily connecting various data sources.
- *Visual preparation of data and integration of expert knowledge:* In order to integrate expert knowledge, a monitoring module is provided. Thereby, experts can analyze the visually prepared collected data and

customize the automatic analysis. Furthermore, the interaction with humans is in general enhanced by enriching the log data with context data. As a result, for example hard to interpret information is transformed into more useful intelligence (e.g. IP-addresses get translated into the geolocation of its origin). Moreover, a high level of usability enables experts to work efficiently.

- *Degree of automation:* A high level of automation is maintained by normalizing, correlating and analyzing the data automatically. Additionally, reports can be generated without human interaction and incident response processes can be conducted autonomously. To reduce time and effort, the data is visually prepared and presented bundled at a single place for the whole companies IT infrastructure.
- *Number of connectible devices:* An interface for connecting log sources and context data is provided in order to easily connect any data generating device. Additionally, the data gets normalized into a standard format for being able to further process it independently of its initial representation.
- *Analytics:* The central component of the pattern is the correlation and analysis module, which can detect incidents or threats automatically. Therefore, the incoming data is correlated. Additionally, the monitoring module provides human analysts with the possibility to analyze data manually.
- *Reusability and intelligence sharing:* In order to provide a high level of reusability, interfaces and modules are identified. Furthermore, sharing of threat intelligence and generating reports for compliance reasons is taken into account. Especially the definition of standardized reporting and sharing formats for detected incidents plays an important role and is a very active topic in current research [Menges and Pernul 2018].
- *Costs:* In order to reduce costs, our approach aims at a high level of automation. Additionally, it supports staff by providing a single interfaces for monitoring the whole organizations' security. The modular design can help to reduce costs even further.

In contrary to the advantages the SIEM pattern has the following *liabilities*:

- *High effort for introduction:* Due to the complexity of historically grown organizations' IT infrastructures, integrating a SIEM demands high effort. Additionally, the heterogeneity of demands to the system makes it hard to create a standard solution, usable by multiple companies.
- *Demand for experts:* Although the demand for staff is reduced, there is still a need for experts for introducing and monitoring the system.
- *Lack of information from observations made by humans:* The pattern only uses log and context data generated by machines. However, humans could also provide valuable information for a SIEM. For example if an employee receives a malicious phone call, which could be the start of a large-scale attack, there is no possibility for him to feed it into the SIEM system.

3.10 Related Patterns

- *Security Patterns for Intrusion Detection Systems (IDS):* The security pattern for IDS [Kumar and Fernandez 2012] is quite similar to the SIEM pattern proposed in this paper. However an IDS does not analyze log data, but requests from clients. Additionally, it protects certain points in a network and is thus not a centralized security system. In the context of SIEM, an IDS can serve as log source as it generates security relevant events, when detecting attacks.
- *Centralized systems logging:* The centralized systems logging pattern uses the factory pattern to create loggers for different applications to gather logs in one single place [Bijvank et al. 2013]. However, it does not address any kind of analysis or security related processing.

– *Adapter Pattern:* In order to be able to add modules with incompatible interfaces, after the SIEM system was installed, Adapter patterns [Gamma et al. 2011] are needed. Additionally, this can enable or at least simplify the connection of external data sources or consumers.

4. PATTERN MINING / IDENTIFICATION PROCESS

The process for identifying the SIEM pattern is derived from the process for pattern identification, authoring, and application introduced by [Fehling et al. 2014]. However, we will mainly focus on the identification of a SIEM pattern. In the following, we describe how those processes are applied. The pattern identification process contains five phases, which we followed accordingly:

Domain Definition: In the first step the domain has to be defined in order to gain a common understanding and knowledge about it. The domain of our pattern is SIEM, as described in chapter 2. In addition, we describe the used terminology in the appendix in order to gain a deeper understanding of domain specific terms.
Coverage Consideration: As our domain is a quite broad research area, we narrow down the coverage of our pattern. Specifically, we limit our pattern to one level of abstraction. In addition, only the software part of SIEM is considered. We do not focus on surrounding processes, which for example are performed by staff in a SOC.
Information Format Design: The information format, we applied in our research is UML. Specifically, we decided to use a component diagram, as it perfectly fits the intended level of abstraction and is well known by most researchers in the field of software engineering.
Information Collection: To gain a holistic view of the relevant aspects we considered several sources of information:

– *Software products:* The most important source of information for our research are products that are already well established in the market of SIEM. In this way the outlined pattern represents the state of the art of SIEM systems and is as close to reality as possible.
– *Research papers:* In order to integrate the current view on SIEM in research we also used scientific papers as source of information. However, there are only few papers, which specifically address SIEM as a whole.
– *Whitepapers:* In most cases companies, developing SIEM systems provide whitepapers, elaborating the functionalities of their products in more detail. In some cases even their architecture is displayed.
– *Manuals:* With the help of the corresponding manuals of the SIEM products additional information can be gained. However, manuals are written for the users of the product and thereby only give few information about the functionality of the product in depth.
– *Documentations:* SIEM products usually provide several kinds of documentations. They mostly target developers, who for example want to extend the software with plug-ins. In addition, documentations for implementing SIEM in a company are a valuable source of information to gain insight into the structure of the software.
– *Product websites:* Websites of the SIEM product serve as additional source of information. However, they normally provide a very shallow insight into the software. Furthermore, they mostly serve marketing purposes. Thus, this information should be treated with caution.

Information Review: To reduce the amount of different sources of information, we have only analyzed chosen SIEM systems. For choosing the most advanced systems, the Gartner Magic Quadrant for Security Information and Event Management [Kavanagh and Bussa 2017] was utilized. Thereby, available SIEM systems are classified regarding the two dimensions *ability to execute* and *completeness of vision*. The first dimension essentially describes the usefulness of the system in daily usage inside an organization and the second one the strategic orientation. We analyzed all systems, classified as leaders: "IBM QRadar", "Splunk Enterprise Security", "LogRhythm Enterprise", "HPE ArcSight" and "McAfee Enterprise Security Manager". In addition, the open source SIEM called "OSSIM" is considered, as it gives a deeper insight into its functionality and is frequently utilized in research projects.

5. CONCLUSION AND FUTURE WORK

In this paper we have deduced a generic SIEM pattern by among other things, analyzing the most advanced SIEM systems on the market. This was done by adhering to a pattern identification process published in literature. In order to derive the pattern, forces that SIEM systems depend on were identified. The pattern was visualized as a UML component diagram and its advantages and liabilities were discussed. In order to give deeper insights, the pattern was compared to a commercially used SIEM and a SIEM resulting from a research project. This paper can be applied by developers of SIEM tools for delimiting modules in order to improve the general structure. Furthermore, it can serve as a foundation for understanding the functional principles and assembly of SIEM.

However, the proposed pattern is not enough to depict SIEM in detail. Thus, it is necessary to continue our research by creating further patterns, explaining the corresponding modules which we have identified in more detail and divide them each into modules themselves.

APPENDIX

A. Terminology

In this section the relevant terminology is outlined. Thereby definitions for specific terms are given in order to establish a common understanding. We deduce the definitions valid in the context of SIEM whereby they may not be applicable in other contexts.

Event / Log: In the context of SIEM the nouns event and log are frequently used synonymously. However, a log is the record of an occurred event. An event can simply be defined as something that happens and can come in different sizes and levels of abstraction (e.g. opening a text file compared to processing an order of a customer) [Luckham 2012]. Thus, multiple logs can describe one event. Furthermore, logs can be generated from a multitude of devices or programs and their representation and structure can vary greatly. Additionally, logs can also be emitted as a data stream. Logs most pertinent for SIEM systems originate from security-relevant devices, like firewalls and routers, but are not restricted to those.

Incident: According to [Cichonski et al. 2013] an incident in the context of IT security is a special form of an event. Thereby it is restricted to adverse events which include loss of data confidentiality or disrupt integrity or availability of systems or data. Additionally, the violation of security policies and standard security practices can be an incident.

IT security analyst: Security analysts in the context of SIEM are usually organized in a SOC. An IT security analyst is an expert in the domain of cyber security and thus is trained to identify vulnerabilities and attacks on an organizations' IT infrastructure. Thereby, the expert analyzes different kinds of data and uses several tools in order to detect anomalies. In connection with SIEM, this data usually consists of logs, which are visually prepared and presented to facilitate its analysis.

ACKNOWLEDGMENTS

We thank our shepherd, Eduardo B. Fernandez, for his valuable comments that have significantly helped to improve this paper.

REFERENCES

AlienVault Inc. 2018. OSSIM: The Open Source SIEM. (2018). Retrieved 24.09.2018 from `https://www.alienvault.com/products/ossim`

Sandeep Bhatt, Pratyusa K. Manadhata, and Loai Zomlot. 2014. The Operational Role of Security Information and Event Management Systems. *IEEE Security & Privacy* 12, 5 (2014), 35–41. `DOI:http://dx.doi.org/10.1109/MSP.2014.103`

Roland Bijvank, Wiebe Wiersema, and Christian Köppe. 2013. Software architecture patterns for system administration support. In *Proceedings of the 20th Conference on Pattern Languages of Programs*. 1.

Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. 2013. *Pattern-Oriented Software Architecture, A System of Patterns* (1. aufl. ed.). Wiley, s.l.

Paul Cichonski, Tom Millar, Tim Grance, and Karen Scarfone. 2013. Computer security incident handling guide. *International Journal of Computer Research* 20, 4 (2013), 459.

Congress of the United States of America. 2014. National Cybersecurity and Critical Infrastructure Protection Act of 2014. (2014).

Christopher Crowley. 2017. Future SOC: SANS 2017 Security Operations Center Survey. (2017).

Michael Docking, Anton V. Uzunov, Chris Fiddyment, Richard Brain, Scott Hewett, and Lee Blucher. 2015. UNISON: Towards a Middleware Architecture for Autonomous Cyber Defence. In *2015 24th Australasian Software Engineering Conference*. IEEE, 203–212. DOI:http://dx.doi.org/10.1109/ASWEC.2015.29

European Parliament. 2016. DIRECTIVE (EU) 2016/1148 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL. (2016).

Christoph Fehling, Johanna Barzen, Uwe Breitenbücher, and Frank Leymann. 2014. A process for pattern identification, authoring, and application. In *Proceedings of the 19th European Conference on Pattern Languages of Programs - EuroPLoP '14*, Veli-Pekka Eloranta and Uwe van Heesch (Eds.). ACM Press, New York, USA, 1–9. DOI:http://dx.doi.org/10.1145/2721956.2721976

Erich Gamma, Richard Helm, Ralph E. Johnson, and John Vlissides. 2011. *Design patterns: Elements of reusable object-oriented software* (39. printing ed.). Addison-Wesley, Boston.

Gartner Inc. 2018. Security Information and Event Management (SIEM). (2018). Retrieved 20.06.2018 from https://www.gartner.com/it-glossary/security-information-and-event-management-siem

Carrie Gates and Sophie Engle. 2013. Reflecting on visualization for cyber security. In *2013 IEEE International Conference on Intelligence and Security Informatics*. IEEE, 275–277. DOI:http://dx.doi.org/10.1109/ISI.2013.6578842

Markus Goldstein, Stefan Asanger, Matthias Reif, and Andrew Hutchison. 2013. Enhancing Security Event Management Systems with Unsupervised Anomaly Detection. In *ICPRAM*. 530–538.

IBM Corporation. 2017. IBM Security QRadar: Architecture and Deployment Guide. (2017). Retrieved 24.09.2018 from https://www.ibm.com/support/knowledgecenter/SS42VS_7.3.1/com.ibm.qradar.doc/b_siem_deployment.pdf

IBM Corporation. 2018. IBM QRadar SIEM. (2018). Retrieved 24.09.2018 from https://www.ibm.com/us-en/marketplace/ibm-qradar-siem

Bhushan Jain, Mirza Basim Baig, Dongli Zhang, Donald E. Porter, and Radu Sion. 2014. SoK: Introspections on Trust and the Semantic Gap. In *2014 IEEE Symposium on Security and Privacy*. IEEE, 605–620. DOI:http://dx.doi.org/10.1109/SP.2014.45

Kaspersky Lab. 2016. Measuring financial impact of it security on businesses: IT Security Risks Report 2016. (2016).

Kelly M. Kavanagh and Toby Bussa. 2017. Magic Quadrant for Security Information and Event Management. *Technical Report - Gartner Inc.* (2017).

Ajoy Kumar and Eduardo B. Fernandez. 2012. Security patterns for intrusion detection systems. In *1st LACCEI International Symposium on Software Architecture and Patterns (LACCEI-ISAP-MiniPLoP'2012), Panama City, Panama*.

LogRhythm Inc. 2018. LogRhythm, The Security Intelligence Company. (2018). Retrieved 24.09.2018 from https://logrhythm.com/

David C. Luckham. 2012. *Event processing for business: Organizing the real time strategy enterprise.* Wiley, Hoboken, N.J. http://onlinelibrary.wiley.com/book/10.1002/9781119198697

McAfee LLC. 2018. McAfee Enterprise Security Manager. (2018). Retrieved 24.09.2018 from https://www.mcafee.com/enterprise/en-us/products/enterprise-security-manager.html

Florian Menges, Fabian Böhm, Manfred Vielberth, Alexander Puchta, Benjamin Taubmann, Noëlle Rakotondravony, and Tobias Latzo. 2018. Introducing DINGfest: An architecture for next generation SIEM systems. In *SICHERHEIT 2018*, Hanno Langweg, Michael Meier, Bernhard C. Witt, and Delphine Reinhardt (Eds.). Gesellschaft für Informatik e.V, Bonn, 257–260.

Florian Menges and Günther Pernul. 2018. A comparative analysis of incident reporting formats. *Computers & Security* 73 (2018), 87–101. DOI:http://dx.doi.org/10.1016/j.cose.2017.10.009

Sabina Georgiana Radu. 2016. Comparative Analysis of Security Operations Centre Architectures; Proposals and Architectural Considerations for Frameworks and Operating Models. In *Innovative Security Solutions for Information Technology and Communications*, Ion Bica and Reza Reyhanitabar (Eds.). Springer International Publishing, Cham, 248–260.

Timo Ropinski, Daniel Archambault, Min Chen, Ross Maciejewski, Klaus Mueller, Alexandru Telea, and Martin Wattenberg. 2017. How do Recent Machine Learning Advances Impact the Data Visualization Research Agenda?. In *IEEE VIS Panel*. Phoenix.

Splunk Inc. 2018. Splunk. (2018). Retrieved 24.09.2018 from https://www.splunk.com/

Eva Weishäupl, Emrah Yasasin, and Guido Schryen. 2018. Information security investments: An exploratory multiple case study on decision-making, evaluation and learning. *Computers & Security* (2018). DOI:http://dx.doi.org/10.1016/j.cose.2018.02.001

Amrit T. Williams and Mark Nicolett. 2005. Improve IT Security With Vulnerability Management. *Technical Report - Gartner Inc.* (2005).