SIEMA: Bringing Advanced Analytics to Legacy Security Information and Event Management

Pejman Najafi(✉), Feng Cheng, and Christoph Meinel

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
{pejman.najafi,feng.cheng,christoph.meinel}@hpi.de

Abstract. Within today's organizations, a Security Information and Event Management (SIEM) system is the centralized repository expected to aggregate all security-relevant data. While the primary purpose of SIEM solutions has been regulatory compliance, more and more organizations recognize the value of these systems for threat detection due to their holistic view of the entire enterprise. Today's mature Security Operation Centers dedicate several teams to threat hunting, pattern/correlation rule creation, and alert monitoring. However, traditional SIEM systems lack the capability for advanced analytics as they were designed for different purposes using technologies that are now more than a decade old. In this paper, we discuss the requirements for a next-generation SIEM system that emphasizes analytical capabilities to allow advanced data science and engineering. Next, we propose a reference architecture that can be used to design such systems. We describe our experience in implementing a next-gen SIEM with advanced analytical capabilities, both in academia and industry. Lastly, we illustrate the importance of advanced analytics within today's SIEM with a simple yet complex use case of beaconing detection.

Keywords: Reference architecture · Next-gen SIEM · Advanced analytic · Big data · Cybersecurity

1 Introduction

Nowadays, an indispensable tool in any organization's arsenal is a Security Information and Event Management (SIEM) system, used as a centralized repository of all aggregated security-related data. The primary source of these data is log data, produced by IT systems across the enterprise's landscape, such as security devices, network infrastructure, host and endpoint systems, applications, and cloud services. Other sources of data include network telemetry, information about inventories, users, assets, and vulnerabilities.

Although originally, the primary purpose of SIEM solutions was to meet regulatory and compliance requirements (e.g., PCI DSS, HIPAA, and SOX), their capability to assist in heterogeneous data correlation, threat hunting, and

J. Garcia-Alfaro et al. (Eds.): SecureComm 2021, LNICST 398, pp. 25–43, 2021.
https://doi.org/10.1007/978-3-030-90019-9_2

monitoring [16] would soon come to play, having more and more companies realizing their ability to detect early, targeted attacks and advanced persistent threats [11]. This would become increasingly important as the years passed since cyber attacks had advanced in sophistication and complexity, rendering the perimeter defenses used by companies at the time insufficient. Therefore, SIEM solutions would come to fill that gap by providing visibility when the traditional defenses (e.g., Antivirus, Firewalls, Intrusion Detection Systems, etc.) are insufficient.

The assumption here is that if a threat has managed to bypass traditional defense perimeters, we expect to see traces of its activities somewhere in the events and logs captured within the SIEM system. Thus, the analysis of such data can be the last safety net to catch potential threats that might slip through.

In the last few years, there has been an increasing interest in analyzing such data for malware detection, both in academia and industry. The industry has taken a more heuristic-based approach [4]. For instance, defining specific patterns and rules such as *alert if Microsoft Word spawns Command-line*. In comparison, the research community has been evaluating statistical, machine learning [18,31] as well as data [28] and graph mining-based approaches [8,13,14].

While these approaches have proven to be effective and successful in academia, their adaptation is yet to be done by industry. The majority of today's SIEM solutions are unable to perform the necessary data analysis. Their architectures are based on proprietary technologies designed in the early 2000s. Since then, we have experienced significant technological advancement, especially in the realm of big data analytics and deep learning. Technologies such as Spark and Hadoop can support heavy data computation and processing and have found application in many industries. Many of today's organizations utilize big data pipelines and analytics to enable data-driven decision making, e.g., LinkedIn [26], Facebook [27]. Therefore, the translation of such work in the cybersecurity industry could be proven invaluable in meeting the ever-changing threat landscape.

In this paper, we outline the requirements needed to build a next-generation SIEM system that would enable advanced analytics whilst supporting traditional SIEM capabilities (Sect. 4.1). Subsequently, we describe the reference architecture allowing individuals to build such systems (Sect. 4.2). Next, we present our experimental setups in two scenarios: an in-house research workbench and a real-world enterprise setup (Sect. 5). Lastly, we discuss a case study highlighting the need for next-gen SIEM with advance analytical capabilities (Sect. 6).

The main contributions of this paper are summarized below:

- Describing the main requirements for a next-gen SIEM with advanced analytical capabilities, hence, Security Information/Event Management and Analytics (SIEMA).
- Outlining the Reference Architecture (RA) for SIEMA with the description of the main components.
- Providing our learnings when implementing and deploying a SIEMA system in a real-world setting.
- Presenting beaconing detection as a case study to highlight the need and value of a SIEM system with analytical capabilities.

2 Background

2.1 Intrusion Detection System

An Intrusion Detection System (IDS) is a monitoring system that is typically installed on a single system attempting to detect suspicious activities and generates alerts that will be consumed by a Security Operator Center (SOC) analysts for further investigation. IDS systems are usually categorized into Host-Based IDS (HIDS) and Network-Based IDS (NIDS) [19]. While HIDs are typically installed on endpoints (hosts) to monitoring operating system resources, NIDs are typically deployed on intermediary network nodes to monitor network traffic.

2.2 Security Information and Event Management

A SIEM system integrates two formerly heterogeneous systems, a Security Information Management (SIM) system and a Security Event Management (SEM) system. SEM systems were originally designed as a tool to provide real-time monitoring for security events and alerts oriented to identify and manage threats. In comparison, SIM systems were designed as a log management tool for record-keeping and reporting of security-related events supporting compliance, forensic investigation, and analysis of security threats [1]. SIEM systems were raised as the result of integrating SIM and SEM to simplify the IT landscape. Since then, these systems have evolved to support a wide variety of needs.

One of the limitations of IDS systems is their limited ability to have a holistic view of the IT landscape to support better decision making, i.e., event correlation. For instance, while a failed login event is nothing to concern with, multiple failed logins to a different host by a single user is concerning. This can only be recognized while correlating events from various endpoints. That is why over time, SIEM systems have evolved to also act as an IDS system supporting threat detection as the last perimeter of defense.

Gartner research group [24] characterized the main requirements for Security Information and Event Management systems as follows:

- **Information and Event Management:** The main requirement for SIEM systems remains as the collection and storage of events and logs from heterogeneous devices in the organization, allowing SOC analysts to monitor the landscape, providing visualization, reporting, and alerting mechanisms. These trends can be created based on real-time and/or historical data to identify patterns that can aid in gaining insight into high-risk behavior. The report can also be used to measure the status against compliance regulations and standards such as PCI DSS, GDPR, HIPAA, and SOX.
- **Threat Hunting and Investigation:** SIEM systems are expected to be the centralized repository holding all security-relevant information and event. These systems are used by SOC analysts to freely explore and analyze data, hunting for threats, or investigating known security incidents. Thus the search features and functionality is fundamental in a SIEM tool. This requires the platform to run efficient ad-hoc queries against massive amounts of data.

- **Rule-based Pattern Matching for Signature-Based Threat Detection:** Today's SIEM systems are highly utilized for threat detection using a rule-based correlation engine [16]. These rules (patterns and signatures) can be as simple as: "alert if there are more than k authentication failures" or as sophisticated as multi-step patterns with dynamic conditions.
- **Automated Correlation and Enrichment:** One of the other most essential features of SIEM systems is the capability to correlate events from disparate sources allowing analysts to see the bigger picture. While the SOC analyst may correlate and join data as part of their hunting/investigation, the ability of the system to automatically enrich certain data points with others may enhance the SOC hunting and investigation activities. For example, the real-time correlation of proxy logs, asset/user information, and DHCP logs would significantly assist during incident handling and response.
- **Cyber Threat Intelligence (CTI) and Open-Source Intelligence (OSINT):** Due to the value of OSINT and CTI, the majority of today's SIEM systems have evolved to support the ingestion of Cyber Threat Intelligence, such as lists of known malicious file hashes, IPs, domains, or other Indicators of Compromise (IOCs), as well as Open-Source Intelligence such as vulnerability data, common weaknesses, domain registrars, etc. The ingested CTI and OSINT enable the development of additional use cases, such as correlation rules based on the IOCs or risk assessment via vulnerability analysis.

3 Related Work

There are numerous commercial SIEM tools. Gartner [16] provides a good overview of the major providers in the SIEM market. The majority of the leading solutions are still based on legacy technologies and architectures. Nevertheless, there are those new players that attempt to bring big data architectures and analytics into Security Information and Event Management.

Gartner [16] identifies Exabeam[1] and Securonix[2] as the top contender for complex security monitoring use cases with advanced threat detection capabilities. The community also identifies these SIEM systems as one of the first truly scalable next-gen SIEM systems, as they both utilize big data technologies such as Hadoop, Apache Spark, Apache Kafka, and trends such as Data Lake, Lambda architecture, stream and batch processing, advanced analytics, and User and Entity Behavior Analytics (UEBA).

There also several open-source attempts to bring SIEM and big data architectures and analytics closer together. Excellent examples of such projects are: Apache Metron [6], Wazuh [32], Apache Spot [7], and Apache Eagle [5].

SANS Institute provides an overview and a guide for the critical features of next-gen SIEM [11]. Menges et al. [20] discuss the shortcoming of traditional SIEM systems such as advanced forensic analysis and propose an extended architecture addressing those shortcomings. Wheelus et al. [33] propose and evaluate

[1] Exabeam, https://www.exabeam.com/.
[2] Securonix, https://www.securonix.com/.

a big data architecture for real-time network traffic processing. The authors discuss several case studies highlighting the potential values in big data analytics in the context of cybersecurity.

In this paper, we provide a reference architecture that abstracts the majority of the above works' capabilities. Ullah et al. [29] provides a comprehensive systematic literature review of frequently reported quality attributes and architectural tactics for big data cybersecurity analytic systems. Each of these tactics can be abstracted by the proposed RA in this paper.

To the best of our knowledge, there are no previous efforts in addressing the limitation of today's SIEM systems, particularly in terms of advanced analytical capabilities.

4 SIEMA

In this section, we first provide an overview of the next-gen SIEM system's main business and architectural requirements. Having covered the main requirements, we provide our reference architecture while highlighting the main components of the blueprint for anyone who wishes to design such systems.

4.1 Requirements

Before proceeding to the reference architecture, it is crucial to outline the added requirements for the next-gen SIEM system. We categorized the main requirements into two groups: Business requirements (BR) and Architectural Requirements (AR).

BR1. Data Science (Advanced Analytics): One of the main limitations of traditional SIEM systems is their reliance on signature/heuristic-based threat detection, limiting the detection only to previously known threats. Finding truly unknowns requires the utilization of state-of-the-art data science (statistics, machine learning, and data mining algorithms). In this regard, the platform should acknowledge state-of-the-art data science tools and techniques [3].

Data science can also be used to eliminate static rules which pose high false-positive rates. For example, instead of looking at 10 authentication failures within 1 min (i.e., attempt to find brute force attacks), one could learn the threshold per user and endpoint, thus reducing false positives.

BR2. Data Engineering (Complex Data Processing): As different use cases may require different shapes of data, the platform should be able to handle data engineering pipelines. This can include data aggregation, correlation, enrichment, normalization, parsing, validation, tagging, duplication, and transformation. In this regard, a next-gen SIEM should allow data scientists, data engineers, and SOC analysts to seamlessly develop, combine, manage, and maintain different data processing pipelines.

AR1. Distributed, Scalable, and Fault-Tolerant: A next-gen SIEM is expected to handle big data with 3Vs: large *volume*, high rate of generation (*velocity*), and heterogeneity of the types of structured and unstructured data (*variety*). Hence, to cope with the volume, velocity, and variety of data produced by today's enterprises, the platform should be scalable and elastic. To achieve this, the best practices in distributed systems (e.g., distributed storage and processing) should be followed and adhered across all architectural levels of the platform.

In a distributed setting, availability and resilience to failure become a challenging yet crucial aspect of the system. In this regard, the platform is also expected to be fault-tolerant and available during a failure/outage, such as network outages or hardware failures.

AR2. Extensible: Today's technology landscape is evolving faster than ever. The most relevant technologies or solutions of today may be irrelevant in a few years. A next-gen SIEM should be able to undergo numerous modifications and extensions to stay relevant in an ever-changing technological world, e.g., able to adopt a new distributed processing framework.

AR3. Open: Today's open-source community is very active and often ahead of its commercial competitors. Therefore, the platform needs to respect open-source solutions and technologies by allowing the adoption of open-source. This will ensure the system's relevance with state-of-the-art technologies.

Furthermore, one of the main criticisms of today's legacy SIEMs is their locked-in data model. A next-gen SIEM should have an open data model respecting the users and data portability.

AR4. Integration: The platform should have standard methods to interface and integrate with other external tools or systems via APIs. This allows other tools to better appreciate the values provided by the next-gen SIEM.

AR5. Data Lake for All Storage Requirements: Storage is a core aspect of a next-gen SIEM. Different use cases require different storage systems, from a relational database to a distributed file store. Particularly, to enable advanced analytics, new data lake architectures are needed.

AR6. Modular Data Ingestion: A next-gen SIEM is expected to ingest a variety of data. These data can be from external sources, such as vulnerability data, indicators of compromise, related OSINT. It can also be from internal sources, such as event logs from network and security systems (e.g., Intrusion

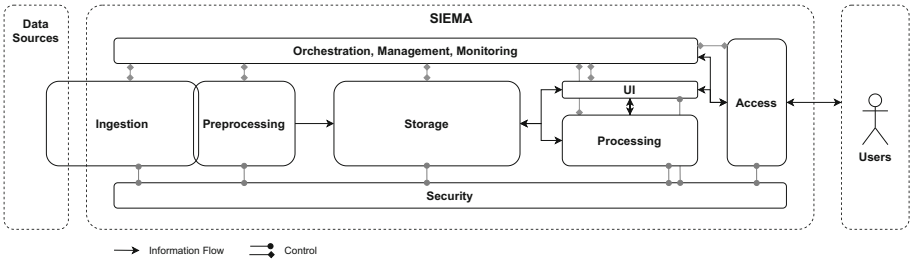

Fig. 1. Module decomposition of the reference architecture

detection systems, endpoint security, firewalls, VPN, proxy, DNS), applications, endpoints, assets, network topologies, security configuration, and policies. Thus, a next-gen SIEM is expected to ingest data from both external and internal sources. The ingestion should mainly expect authenticated incoming data and the possibility of crawling or collecting, e.g., to crawl related OSINT.

AR7. Security and Privacy: A next-gen SIEM is also expected to guarantee security (security by design). This includes but is not limited to: privilege separation, access control, encryption at rest and transit, privacy, anonymization/pseudonymization, least privilege principle, access, and audit logging.

4.2 Reference Architecture

We design our RA based on decade-long experience and knowledge revolving around the best practices in designing big data architectures and pipelines, e.g., LinkedIn [26], Facebook [27], and other reference architectures [17]. Figure 1 shows the high-level reference architecture for the proposed SIEM. Figure 2 illustrates the system workflow consisting of five main stages.

Fig. 2. SIEMA proposed workflow

Data Ingestion and Collection: This layer is expected to consist of a collection of extendable ingestors and collectors, each designed to ingest a particular data source. Data sources can be *internal* data, such as event logs, telemetries, inventories, or *external* data, such as OSINT, and vulnerability data.

These modules can either accept authenticated data being forwarded (push model) or pull particular data points. Data being forwarded can either come directly from the data sources or being forwarded by a remote ingestor node. The pull mechanism is expected to be limited due to the lack of a dedicated agent.

In Sect. 7, we discuss the extension of SIEMA to allow better collection and response, hence introducing Security Orchestration, Automation and Response (SOAR) like capabilities.

Pre-processing: The next stage in the system is pre-processing, i.e., data validation, cleansing, optimization (e.g., de-duplication), parsing, and basic transformation (e.g., standardizing the timestamps), and basic enrichment/tagging (e.g., tagging the events according to their type and source).

All pre-processing steps are expected to be simple, efficient, and scalable. Further enrichment and correlation are expected to be carried out by the analytic/processing modules.

After pre-processing, the data shall be dumped on messaging queues or directly to file storage where the primary data processing pipelines could take the lead for more advanced data processing (e.g., normalization), storage (e.g., ETL, indexing, etc.) or analytics.

Storage: Similar to the data lake architectural pattern, storage is a key and fundamental component for RA. The main objective of the data storage layer is to provide reliable and efficient access to persisted data. Part of this is to offer multiple representations for single data records to accommodate different use cases, e.g., OLAP style data analytics, OLTP queries, or string searches.

Note that the storage system of such next-gen SIEM is expected to be highly scalable and agile. This is integral, as organizational and business needs change over time, calling for adjustments in technologies and environment setups.

One can categorize storage needs based on latency, throughput, access patterns, and data type. Some examples of storage needs in the context of a next-gen SIEM are: NoSQL transactional database for results. Index store and search engine to allow string search on event logs for efficient ad-hoc threat hunting and investigation. A Distributed file/object-store to satisfy data lake requirements for advanced and distributed analytics and machine learning models. A queuing mechanism to enable the reliable transmission of data across different processing layers. For instance, a messaging queue that enables an enrichment module to enhance the result of an anomaly detection algorithm, i.e., outliers are pushed to a queue, where the enrichment module is subscribing to, enriching the outliers with related contexts. An in-memory caching mechanism to allow multiple executors or workers within a processing module to exchange data efficiently. A graph database to store the relationships between various internal and external entities.

Processing: The Processing layer is responsible for efficient, scalable, distributed, and reliable processing. At a high level, it can serve two main purposes: *data engineering* and *data science*. The data engineering sub-layer is responsible for data processing and transformation, e.g., event correlation and enrichment, normalization, ETL pipelines, storage optimization (compression, partitioning,

bucketing), pattern matching, etc. On the other hand, the data science-based modules are concerned with knowledge extraction from the data, e.g., machine learning, data mining, statistical analysis, graph analytics, etc. It is worth noting that analytical processing can be interactive, batch, and stream.

Access: The access layer is the interaction point of the system with external actors. These actors can be SOC analysts, data scientists, data engineers, administrators, managers, or external APIs. Each actor is expected to require interaction with a specific part of the platform for a particular reason. The access layer is responsible for managing these interactions while ensuring security and load balancing. For example, a SOC analyst requiring an interactive interaction with the platform's search capabilities to investigate threats, defining rules for the correlation engine and dashboard to view the alert, and visualizing the trends.

UI: The user interface layer is responsible for abstracting actors' interactions with storage or processing modules. For instance, a UI that enables SOC analysts to run their query against the storage system, or the data science notebooks designed to allow the data scientist to interactively analyze data loaded from the storage layer in the processing layer.

Orchestration, Management, and Monitoring: This layer has three main responsibilities: orchestration, management (administration), and monitoring.

The orchestration module is responsible for providing configuration, management, and coordination between the various platform layers and their modules. This includes job submission, collectors configurations, data engineering pipelines, etc. In addition, this module is also expected to provide monitoring capabilities for every module within each layer, e.g., monitoring the analytical jobs and their status.

The administration/management module is responsible for the configuration, provisioning, and control of the underlying infrastructure and the platform itself, e.g., managing the underlying storage system.

Lastly, the platform auditing module supports the health monitoring of the system and its underlying heterogeneous systems. For instance, it is expected that the storage layer will consist of multiple systems, e.g., distributed files system, NoSQL database, a messaging queue. In this regard, this module should allow administrators to monitor the health and performance of these systems.

Security: Given the nature of our next-gen SIEM system, there are concerns around the security and privacy of such big data platforms. In this regard, this layer is responsible for the security of the platform and its underlying data. This includes enforcement of access rules, restricting access based on classification or need-to-know, and securing data at rest or in transit.

5 Experimental Setup and Deployments

We have endeavored two implementations of the proposed RA, in-house academic research workbench and real-world experimental setup in an international enterprise's infrastructure.

5.1 In-House Research Workbench

Our first attempt to develop and deploy a SIEMA system according to the proposed RA was done on a cluster consisting of two Dell PowerEdge (R730, R820) and five Fujitsu Primergy RX600 with a total of 1,864 GB RAM, 24 CPUs (200 total cores), and 4 TB storage interconnected via 10 Gb optical fiber. In addition, an external Network Attached Storage (NAS) connected to the cluster via 3× 10 Gb optical fiber. Table 1 presents the leading underlying technologies used for this setup.

Table 1. In-house SIEMA as first research POC.

Technology	Reference	Usage
Kubernetes	Orchestration/Management	Backbone system and orchestrator
Ansible	Orchestration/Management	Platform operation and administeration
Zookeeper	Orchestration	Configuration maintenance and synchronization
Apache Spark	Processing	Distributed processing and analytics engine
Presto	Processing	Distributed processing (SQL query engine)
Apache Livy	Management	Multi-tenancy and job management
Apache Kafka	Storage	Distributed messaging and queueing
Hadoop HDFS	Storage	Distributed file system and object store
Elasticsearch	Storage	Search engine, index store
Prometheus	Storage	Time series database for metrics regarding the platform health
HBase	Storage	Distributed NOSQL data store on top of HDFS
Apache Nifi	Processing, UI	Orchestration and data preprocessing
Kibana	UI	UI for interaction with the search engine
Grafana	UI	UI for platform health monitoring
Zeppelin	UI	Notebook for ad-hoc data science
CMAK	UI, Orchestration	Cluster manager for Apache Kafka
Hue	UI	Hadoop interface

The platform designed was utilized during multiple successful research to apply different data mining and machine learning approaches to the problem of malicious Domain/IP detection using proxy and DNS logs, resulting in multiple publications [21,22].

5.2 Real-World Enterprise Setup

We also had the opportunity to explore a SIEMA system in a real-world setting with a large international company with quite a mature cyber defense. This company had a cloud-based legacy SIEM continuously utilized for threat hunting, monitoring, and rule-based threat detection. We attempted to build around it with analytical capabilities to explore the potential values. We utilized available cloud-based services compliant with the companies policies, such as Azure Data Factory, Azure Data Lake Storage, and Databricks, to enable advanced data engineering and science.

The first significant value of this platform was the ability to run basic aggregation, correlation, and statistical queries over larger time frames (over 100 terabytes of data) which would not be possible with traditional SIEM systems as they were designed for only interactive investigation and searches.

The next value was the ability to create successful advanced use cases using the underlying data (EDR, proxy, DNS). Examples of such use cases are beaconing detection, malicious processes detection, suspicious DNS and proxy requests, windows logon anomalies, and user behavior analytics over weeks of data. The result of these use cases led to the rise of multiple incidents missed by traditional rule-based detections.

Lastly, the ability to train a model to rate and prioritize traditional SIEM alerts according to past experiences. Most of today's organizations receive 17,000 alerts per week where more than 51% of the alerts are false positives, and only 4% of the alerts get adequately investigated [2]. Therefore, prioritization of alerts can help the SOC analyst to focus their efforts better. This was the last use case, designed to read past investigated alerts, their artifacts, and the associated responses (thus labels) to train a model that attempts to prioritize the new alerts according to their potential to be true positives. This prioritization was achieved by adding a confidence score to the alert passed to the traditional SIEM dashboard. The initial impression and qualitative evaluation of this use case seemed promising.

6 Case Study: Beaconing Detection

To better understand the need for advanced analytical capabilities within today's SIEM, we decided to prepare a simplified experiment performed on our real-world setup. Particularly a heavy yet straightforward use case of beaconing detection.

One of the characteristics of sophisticated cyber threats, such as Advanced Persistent Threats (APTs), is periodic attempts to reach out to the command and control (C&C) infrastructure controlled by the adversary to receive further instructions. Such heartbeat and callback behavior is known as beaconing.

Malware beaconing is typically characterized by two main configurations: sleep time and jitter (variations from central value). The beaconing frequency can vary from slow and stealth to fast and aggressive (from a few seconds to hours

or even days sleep time). Nevertheless, generally, it is expected that the adversaries maintain regular beacons for better visibility and control of the infected machines [15].

While at first glance, beaconing detection seems simple, it is quite challenging:

- **Temporal Analysis in Big Data**: In order to detect beaconing, one has to analyze the traffic behavior of all source and destination pairs over an extended period of time. This makes beaconing detection a big data problem.
- **Intentional Randomness and Jitter**: One of the other challenges with beaconing detection is the adversarial strategies to hid the beaconing behavior. One of the common ways the adversaries attempt to prevent detection is by varying the sleep time to make it appear as normal traffic. Other methods can include omitting certain beacons or injecting additional random beacons.
- **False Positives (Benign Applications)**: While we discussed the maliciousness of beaconing behavior, there are several scenarios in which beaconing is an integral part of the communication and does not indicate maliciousness. For instance, Network Time Protocol (NTP), automated software patching, mailing clients, updates, or keep alive traffic in long-lived sessions may also appear as beacons.
- **External Factors**: There can be unanticipated external factors that can introduce errors while looking at the periodicity, such as the host (endpoint) going offline or network interruptions.

6.1 Detection Approach

Beaconing detection has been studied widely in the literature [12,15,25,30], and while there are many ways to develop a beaconing detection approach, here we present one of the simplest ones using statistical methods.

(i) *Data Preparation*: We start by pre-processing the network connection events keeping only the source (host unique identifier), the destination (e.g., IP address and port), and the timestamp fields. One could also validate to ensure the destinations are valid and timestamps are in Unix Timestamp format.

(ii) *Delta Time Calculations*: Next, we group connections by source and destination and sort them by their timestamp. This will allow us to calculate the time deltas between connections of each source and destination pair. For instance, if host *H1* connects to destination *D1* at the time *t1* and *t2*, the time delta between these two connections is $t2 - t1$.

(iii) *Clean Time Deltas*: To ensure the detection quality, we need to filter out bad entries, e.g., border time delta (nulls) - indicating no previous connections, or time deltas equal to zero - indicating network issues resulting in multiple connections in a very short time.
As mentioned, one of the challenges with beaconing detection can also be external factors such as network interruptions or the host going offline (host

shutting down). We tackle this challenge by filtering out outliers in time deltas for each source and destination pair using Interquartile Range [34]. This ensures that when the host has gone offline, the time delta showing the big gap is treated as an outlier, hence eliminated from further calculations.

(iv) *Periodicity via Average and Standard Deviations*: While there are more sophisticated ways to detect periodicity [9,10,23], here we take the simplest approach. We calculate the average and the standard deviation of time deltas for each source and destination pair to estimate the periodicity of connections.

(vi) *Destination's Reputation*: To tackle the false positives (i.e., benign applications), we also calculate the prevalence of each destination, i.e., how many hosts (sources) have connected to this destination during the analysis period. This can be achieved by simple grouping and counting.

(vii) *Scoring*: At this point, for each source and destination, we have the average and standard deviation of time deltas, number of connections (beacons), and the prevalence of the destination. One can now define certain heuristics to score the beaconing behavior to prioritize the alerts. For this case study, we simplify our scoring to three main functions:

- Low Coefficient of Variation: The coefficient of variation is defined as the ratio of the standard deviation to the mean.
  While a low standard deviation of time deltas means perfect periodicity (almost all time deltas are the same), it does not consider how big the average is. That is why the relative standard deviation (coefficient of variation) can help by looking at the ratio.
  We utilize an exponential function (Eq. 1) to transform the coefficient of variation into a score. The reasoning here is that all low ratios (an indication of better periodicity and beaconing behavior) should be scored closer to 1, and as the ratio gets bigger, it should have a decaying effect (logarithmic) in the scores, getting closer to 0.

$$S_{cv} = e^{-\sqrt{\frac{\sigma}{\mu}}} \tag{1}$$

  where μ and σ is the average and standard deviation of time deltas respectively, and S_{cv} is the score derived from the coefficient of variation.
- Low Destination Reputation: Destinations with high reputations (largely accessed by the majority of the endpoint) typically indicate benignness. That is why we would prioritize those beaconing alerts whose destination is rarely observed. More specifically:

$$S_{rep} = e^{-\frac{p(dest)}{k}} \tag{2}$$

  where $p(dest)$ indicates the prevalence of the destination (i.e., how many hosts have been observed connecting to this destination). k is a numerical constant internally determined based on domain knowledge as the threshold to smooth the curve (i.e., after k the scores should smoothen as we don't care anymore). For example, if k is set to 100 that means as

$p(dest)$ gets closer and pass the threshold of 100 hosts, the score should be low, and there is no significant difference between 300 to 600 to 1000, as we only care for low numbers (e.g., 1 or 10 hosts).

– Beaconing Consistency: Sometimes, just the destination prevalence is insufficient to filter out benign applications, particularly updates. In this regard, one could take yet another attempt to eliminate those beaconing-like behaviors. A malicious beaconing could be characterized by its consistency, whereas some benign behavior such as an update will only appear for a certain time. Thus, one could analyze the consistency of the beaconing behavior by looking at the ratio of the number of the beacons and their average delta time to the analysis range.

$$S_{con} = \frac{b \cdot \mu}{t_e - t_s} \tag{3}$$

where b is the number of beacons, μ is the average, $t_e - t_s$ is the analysis range (e.g., 86400 s).

Note that while here we discussed only three simple scoring functions on top of the information available with our case study, one could design multiple other heuristics to reduce the false-positive rate. Lastly, we aggregate the scores via an aggregator function such as weighted average to derive a single score.

(vii) *Alerting*: Having a final score for each source and destination pair, we could sort the alerts descending and take the first k items (where k is set by the rate the analyst can handle). One could also do further analysis for the distributions of the scores to dynamically set k.

6.2 Experiment Setup

We carried out our experiment for this case study within the premise of a large international organization. Particularly, we implemented the described methodology for beaconing detection as a use case within the enterprise SIEM system as well as our analytical platform (discussed in Sect. 5.2). Therefore, the implementations were identical in terms of their logic.

While we cannot discuss the details of the traditional SIEM system used by the enterprise due to NDA, we can confirm that the SIEM system is among the top SIEM leaders identified by the Gartner Research Group [16]. Furthermore, the setup is among one of the largest enterprise SIEM setups, designed to handle the ingestion of more than 10 TB per day.

Our analytical platform for this experiment was configured on Databricks with 5 "Standard_D32s_v3" workers. Thus, having a big data platform backed up by Apache Spark with a total of: 640-GB Memory, 160 vCPU Cores, 1280 GB temp SSD storage.

We ran our main experiment on one day of network connections collected from an EDR tool (172.5 million events) which spanned to approximately 102 GB.

6.3 Results

Running the described beaconing detection on the traditional SIEM took 45 min to go over 89 million events before reaching the disk usage limit (set by the enterprise) and return 246 events. This is because traditional SIEMs are not designed for large-scale analysis (i.e., distributed processing). They tend to aggregate the events of interest into a single server where the calculations take place. In contrast, our analytical environment supported by Databricks and Apache Spark was able to go through all 172.5 million events and finishing the use case within only 26 s.

Although there are many variables in place that make this comparison unfair (e.g., the clusters sizes not being the same, the implementations of a simple calculation such as mean, etc.), one can observe the enormous gap between the capabilities.

One of the other expectations of such analytical platforms is their ability to scale out. Figure 3 shows the runtime of beaconing detection as we add more workers to the run the use case.

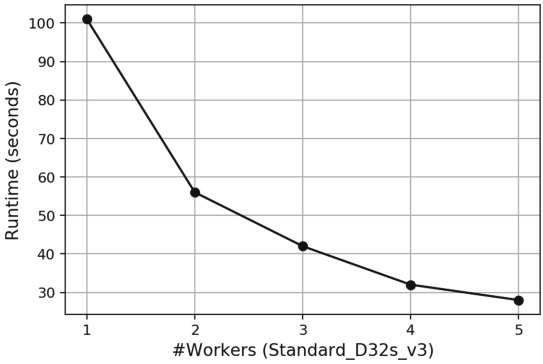

Fig. 3. Beaconing detection use case run-time on databricks based on the number of workers.

Lastly, while we could not run the use case for 5 weekdays on the traditional SIEM, we could run it in the analytical environment. In this regard, the described setup (with 5 "Standard_D32s_v3" workers) could analyze approximately 960 million events (over 550 GB) in 126 s.

6.4 Discussion and Lessons Learnt

With a simple statistical-based use case, we highlighted that traditional SIEM systems are not designed for advanced analytics. One could imagine how more sophisticated analytics, e.g., machine learning, data mining, and graph analytics, will further challenge traditional SIEMs.

As highlighted by most related work, the ability to run complex data analytics is one of the most critical capabilities required for the next-gen SIEM systems to give us a fighting chance against previously unknown threats.

Nevertheless, we cannot underestimate the need for legacy SIEMs, particularly when investigating incidents and alerts. While the analytical platform will take a long time to search, as it requires touching almost all files, the legacy SIEMs are designed for optimized searching, allowing a SOC analyst to run ad-hoc queries investigating incidents and correlating data on-demand. For instance, in our example, while the legacy SIEM took 23 s to search the context of one of the alerts, the analytical platform took more than 1 min. Note that the searches are on one day of data; as the time frame gets bigger, the analytical platform will take even longer (for random searches). Although one could argue that there are ways to speed up the search, e.g., partitioning, bucketing, indexing, and adding meta-data, it still will not be comparable to traditional SIEMs (i.e., index stores) that are designed for optimized searches.

This highlights the need for next-gen SIEM systems that integrate the capabilities of both big data platforms and legacy SIEMs to provide ultimate value to today's SOCs.

7 Future Work

Today's organizations require more than just a SIEM - they require a fully managed system that can interact and respond. That is why many vendors try to close the gap between SIEM systems and Security Orchestration, Automation, and Response (SOAR) systems.

SOAR systems' capabilities mainly include automation that usually occurs through playbooks, runbooks, and incident response capabilities such as triage, containment, and remediation. However, to provide this automation and response capability, there is a need for a dedicated agent running on each endpoint.

In this work, we explicitly focused on SIEM and analytical capabilities, thus putting data collection and incident response out of the scope. However, in our future work, we would like to extend the reference architecture with an assumption of dedicated agents managed by the platform. This agent can enable new capabilities, such as better event collection, event pre-processing on the edge, asset discovery, vulnerability assessment and scanning, policy and configuration checking, file integrity, service availability monitoring, software inventory, and incident response.

Lastly, we would like to better benchmark and evaluate our implemented SIEMA system's capabilities compared to traditional SIEMs.

8 Conclusion

In this paper, we discussed the limitations of the current SIEM systems, in particular, their ability to perform advanced analytics and utilize state-of-the-art data

mining, machine learning, and graph mining approaches. With that in mind, we described the requirements for a next-generation SIEM system with advanced analytical capabilities, hence SIEMA (Security Information/Event Management and Analytics). Next, provided the reference architecture for SIEMA, considering best practices and patterns in big data architectures and pipelines. Next, we described our implementation of such SIEMA under two settings. One, as a research workbench with all open-sources technologies. Second, a version of the proposed architecture in a real-world setting next to an international organization's traditional SIEM system. We also highlighted the value added by the analytical capabilities of such a system to not only help in pushing the research in data mining for threat detection but also developing successful advanced use cases in an industrial setting leading to the detection of genuine threats and incidents. Lastly, we presented beaconing detection as a case study highlighting the limitations of the traditional SIEM systems in comparison to those with analytical capabilities.

We believe the cybersecurity domain, particularly in the industry, is falling behind when it comes to the advancement in data-driven decision-making and data science. One of the main contributing factors to this is the long-delayed evolution of the security systems holding the data (SIEMs), thus limiting the capabilities of SOC analysts to explore statistical and machine learning approaches for threat detection. We hope this paper motivates the design and implementation of next-gen SIEM systems giving more power to SOC analysts, closing the gap between traditional threat hunters and today's data scientists/engineers.

References

1. A Practical Guide to Next-Generation SIEM, Tech. Rep. SENSAGE
2. How many alerts is too many to handle? https://www2.fireeye.com/StopTheNoise-IDC-Numbers-Game-Special-Report.html
3. Improve Threat Detection with Big Data Analytics and AI, Tech. Rep. Databricks
4. Anthony, R.: Detecting security incidents using windows workstation event logs. SANS Institute, InfoSec Reading Room Paper (2013)
5. Apache Software Foundation: Apache eagle. https://eagle.apache.org/
6. Apache metron. https://metron.apache.org/
7. Apache spot. https://spot.apache.org/
8. Chau, D.H.P., Nachenberg, C., Wilhelm, J., Wright, A., Faloutsos, C.: Polonium: tera-scale graph mining and inference for malware detection. In: Proceedings of the 2011 SIAM International Conference on Data Mining, pp. 131–142. SIAM (2011)
9. Elfeky, M.G., Aref, W.G., Elmagarmid, A.K.: Periodicity detection in time series databases. IEEE Trans. Knowl. Data Eng. **17**(7), 875–887 (2005)
10. Elfeky, M.G., Aref, W.G., Elmagarmid, A.K.: Warp: time warping for periodicity detection. In: Fifth IEEE International Conference on Data Mining (ICDM 2005), p. 8. IEEE (2005)
11. Filkins, B.: An Evaluator's Guide to Nextgen SIEM. SANS Institute, Information Security Reading Room (2013)
12. Gardiner, J., Cova, M., Nagaraja, S.: Command & control: understanding, denying and detecting-a review of malware c2 techniques, detection and defences. arXiv preprint arXiv:1408.1136 (2014)

13. Hassan, W.U., Bates, A., Marino, D.: Tactical provenance analysis for endpoint detection and response systems. In: IEEE Symposium on Security and Privacy (SP), vol. 2020, pp. 1172–1189. IEEE (2020)
14. Hassan, W.U., et al.: NODOZE: combatting threat alert fatigue with automated provenance triage. In: Network and Distributed Systems Security Symposium (2019)
15. Hu, X., et al.: Baywatch: robust beaconing detection to identify infected hosts in large-scale enterprise networks. In: 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 479–490. IEEE (2016)
16. Kavanagh, K., Bussa, T., Sadowski, G.: Magic Quadrant for Security Information and Event Management. Gartner Group Research Note (2020)
17. Klein, J., Buglak, R., Blockow, D., Wuttke, T., Cooper, B.: A reference architecture for big data systems in the national security domain. In: IEEE/ACM 2nd International Workshop on Big Data Software Engineering (BIGDSE), vol. 2016, pp. 51–57. IEEE (2016)
18. Kolosnjaji, B., Zarras, A., Webster, G., Eckert, C.: Deep learning for classification of malware system call sequences. In: Kang, B.H., Bai, Q. (eds.) AI 2016. LNCS (LNAI), vol. 9992, pp. 137–149. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50127-7_11
19. Liao, H.-J., Lin, C.-H.R., Lin, Y.-C., Tung, K.-Y.: Intrusion detection system: a comprehensive review. J. Netw. Comput. Appl. **36**(1), 16–24 (2013)
20. Menges, F., et al.: Introducing DINGfest: an architecture for next generation SIEM systems (2018)
21. Najafi, P., Mühle, A., Pünter, W., Cheng, F., Meinel, C.: MalRank: a measure of maliciousness in SIEM-based knowledge graphs. In: Proceedings of the 35th Annual Computer Security Applications Conference, pp. 417–429 (2019)
22. Najafi, P., Sapegin, A., Cheng, F., Meinel, C.: Guilt-by-association: detecting malicious entities via graph mining. In: Lin, X., Ghorbani, A., Ren, K., Zhu, S., Zhang, A. (eds.) SecureComm 2017. LNICST, vol. 238, pp. 88–107. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78813-5_5
23. Rasheed, F., Alhajj, R.: STNR: a suffix tree based noise resilient algorithm for periodicity detection in time series databases. Appl. Intell. **32**(3), 267–278 (2010)
24. Sadowski, G., Bussa, T., Kavanagh, K.: Critical Capabilities for Security Information and Event Management. Gartner Group Research Note (2020)
25. Shalaginov, A., Franke, K., Huang, X.: Malware beaconing detection by mining large-scale DNS logs for targeted attack identification. In: 18th International Conference on Computational Intelligence in Security Information Systems. WASET (2016)
26. Sumbaly, R., Kreps, J., Shah, S.: The big data ecosystem at linkedin. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, pp. 1125–1134 (2013)
27. Thusoo, A., et al.: Data warehousing and analytics infrastructure at Facebook. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, pp. 1013–1020 (2010)
28. Ucci, D., Aniello, L., Baldoni, R.: Survey of machine learning techniques for malware analysis. Comput. Secur. **81**, 123–147 (2019)
29. Ullah, F., Babar, M.A.: Architectural tactics for big data cybersecurity analytics systems: a review. J. Syst. Softw. **151**, 81–118 (2019)
30. Van Splunder, J.: Periodicity detection in network traffic. Technical Report, Mathematisch Instituut Universiteit Leiden (2015)

31. Wang, Q., et al.: You are what you do: hunting stealthy malware via data provenance analysis. In: Symposium on Network and Distributed System Security (NDSS) (2020)
32. Wazuh: The open source security platform. https://wazuh.com/
33. Wheelus, C., Bou-Harb, E., Zhu, X.: Towards a big data architecture for facilitating cyber threat intelligence. In: 2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–5. IEEE (2016)
34. Yang, J., Rahardja, S., Fränti, P.: Outlier detection: how to threshold outlier scores? In: Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing, pp. 1–6 (2019)