8th IEEE International Conference and Workshops on Recent Advances and Innovations in Engineering- ICRAIE 2023
(IEEE Record #59459)

2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE) | 979-8-3503-1551-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRAIE59459.2023.10468438

A Log Aggregation Design Criteria for Robust SIEM (Security Information and Event Management) in Enhancing Threat Detection

Mohsen Bin Mohamad Hata
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA
Selangor, Malaysia
mohsen@uitm.edu.my

Mohamad Yusof Bin Darus
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA
Selangor, Malaysia
yusof_darus@uitm.edu.my

Muhammad Zul Akmal Bin Shafiee
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA
Selangor, Malaysia
muhammadzulakmalbinshafiee@gmail.com

Elvisianah Petrus
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA
Selangor, Malaysia
elvisianahptrs@gmail.com

Yasmin Athira Jamian
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA
Selangor, Malaysia
yasminathira1210@gmail.com

***Abstract*—Security Operations Centers (SOCs) play a vital role in protecting organizations from cyber threats. Supported by skilled Security Analysts, they are the first line of defense, monitoring and responding to incidents. The Security Information and Event Management (SIEM) system is a critical tool for managing log data efficiently. This research focuses on optimizing log data aggregation within a SOC's SIEM framework. By exploring various log aggregation techniques, we aim to enhance the performance of data collectors, leading to quicker response times and improved security. This research contributes to a more robust defense against the ever-changing landscape of cyber threats. It empowers organizations to face evolving challenges with confidence and resilience.**

***Keywords*—Security Operation Center (SOC), Security Information and Event Management System (SIEM), Log Aggregation, Log Management, Incident Handling**

I. Introduction

In the fast-evolving digital landscape, organizations confront a rising tide of cyber threats, compelling many to establish Security Operations Centers (SOCs). These centers serve as vital hubs, overseeing security, detecting incidents, and responding promptly. SOCs are indispensable for safeguarding digital assets amidst the surge in cyber threats, playing a pivotal role in proactively identifying and thwarting security incidents. Security Analysts, akin to digital Sherlock Holmes, form the core of SOCs. With keen insights and swift decision-making, they analyze security data, spotting and halting potential threats. An effective SOC relies on a Security Information and Event Management (SIEM) system, the intelligence center that consolidates data from diverse sources. SIEM employs smart tools and rules, enabling organizations to decipher security events and respond swiftly and intelligently.

Ensuring the effectiveness of a SOC hinges on a robust Security Information and Event Management (SIEM) system—a cognitive hub that consolidates data from various sources. SIEM, the brain behind SOC operations, employs intelligent tools and rules to decipher security events. This research delves into the realm of SIEM, focusing on its handling of log data within a SOC. Through an exploration of diverse log aggregation techniques, the goal is to optimize the collector's performance. This optimization translates to faster response times and heightened security for organizations grappling with an ever-shifting threat landscape.

II. LITERATURE REVIEW

Now, let's delve into the core studies and research papers that illuminate log aggregation techniques in Security Information and Event Management (SIEM) systems. These works provide valuable insights into optimizing log data handling for enhanced efficiency within Security Operations Centers (SOCs).

A. *Aggregation of Elastic Stack Instruments for Collecting, Storing and Processing of Security Information and Events*

The paper aims at development of the generic architecture and a research prototype of the system for collecting, storing and processing of data and security events based on big data technology, as the basis for a next generation SIEM system [8].

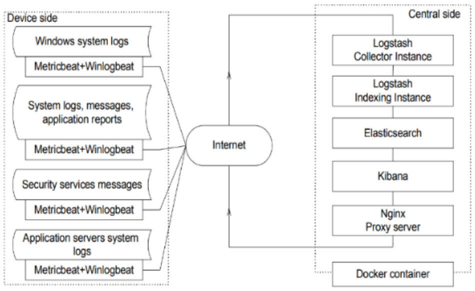


Fig. 1. The architecture of the prototype

979-8-3503-1551-6/23/$31.00 ©2023 IEEE

The implemented prototype functions within a well-defined architecture, collecting XML document data from client devices and transmitting it to an ELK stack hosted on a server—comprising Elasticsearch, Logstash, and Kibana. Winlogbeat and Metricbeat act as collectors, with Logstash playing a pivotal role in collecting and indexing incoming data. A specialized Logstash instance indexes information from these collectors and forwards it to Elasticsearch for storage and processing. Analytical processing primarily occurs through Kibana, enabling users to visualize data through charts and graphs. External access to Kibana is facilitated by an Nginx server acting as a reverse proxy.

Conceptually, the prototype's architecture encompasses components for sending data, pipelining and data delivery, fault tolerance, load balancing, and a subsystem housing the search and analytical core with storage, along with a dedicated visualization component. The prototype offers extensive coverage for analysis, supporting event collection from various sources such as syslog protocol, Windows event logs, telemetry hardware, OS, and services. Additionally, it handles network traffic flow data from netflow/sflow, facilitated by Beats like Filebeat, Winlogbeat, Metricbeat, and Packetbeat.

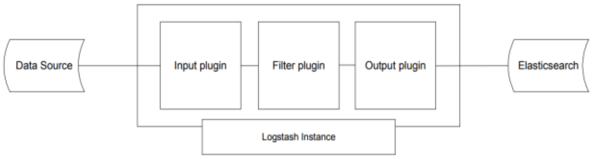


Fig. 2. The scheme of Logstash Function

Logstash assumes a critical role in the data processing pipeline, comprising three key functional blocks: Input, Filter, and Output. The Input plugin discerns the event source, while the Filter plugin executes intermediate processing, extracting vital information and structuring data. The Output plugin dictates the final processing route, dispatching data to Elasticsearch. To ensure scalability and fault tolerance, multiple Logstash instances are deployed, each fulfilling distinct functional roles, thereby enabling efficient load balancing across data sources and the Logstash cluster. Additionally, an Elasticsearch cluster is established to store and index information, with indexes subdivided into types and further segmented into shards, strategically distributed across nodes.

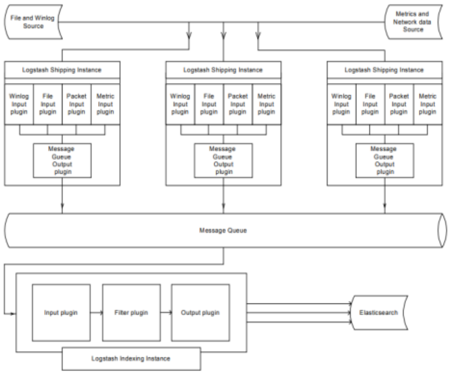


Fig. 2. The architecture of the pipeline for data processing and delivery

Finally, the system elements are deployed in Docker containers, enabling automated deployment and management. This microservices approach facilitates easier upgrades, testing, and compatibility checks between services. It also addresses issues of reliability and backup of infrastructure services, allowing engineers to focus on the logic of the solution rather than infrastructure concerns.

B. *Social Media Monitoring using ELK Stack*

This paper mainly aims to implement a SOC Environment which takes in logs from Twitter and then forwards it to a central server [2].

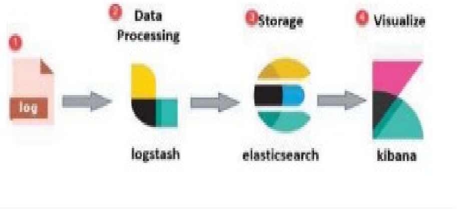


Fig. 3. ELK Architecture

Beats function as agents collecting data, which is sent to Logstash for filtering, parsing, and transformation before being directed to ElasticSearch for storage. This stored data is visualized in Kibana. To access real-time Twitter data, users must request access to the Twitter API, obtaining consumer key, consumer secret key, Access Token, and Access Token Secret, acting as a password for making requests on behalf of the app. The acquisition process, contingent on owning a Twitter account, takes 24 hours. The obtained Twitter logs are shipped to Elasticsearch for subsequent purposes.

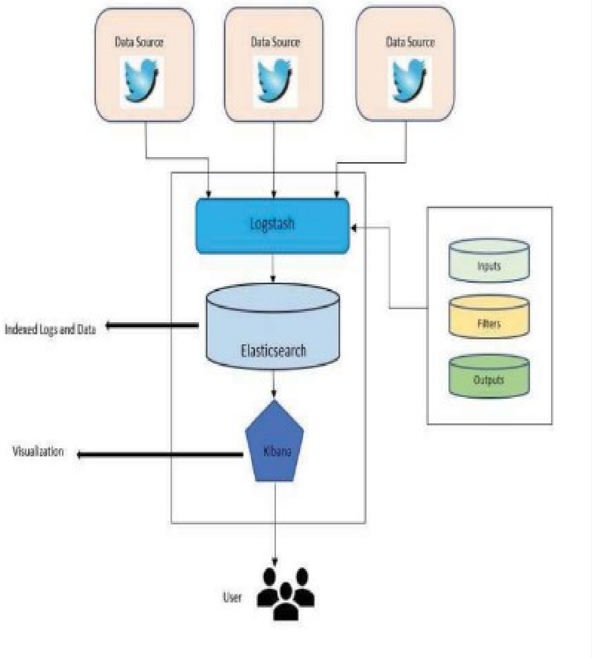


Fig. 4. Flowchart ELK with Twitter API

C. *Integration of Splunk Enterprise SIEM for DDoS Attack Detection in IoT*

The proposed paper presents a method for a basic investigation (triage) of the alarm, IP reputation check of the destination system with OSINT and recommendations for isolation of the internal device. It is crucial to quarantine the IoT system to mitigate the risk of malware spreading across the environment and transforming other devices in remotely controlled bots, suitable for launching DDoS attacks or mining cryptocurrency [5].

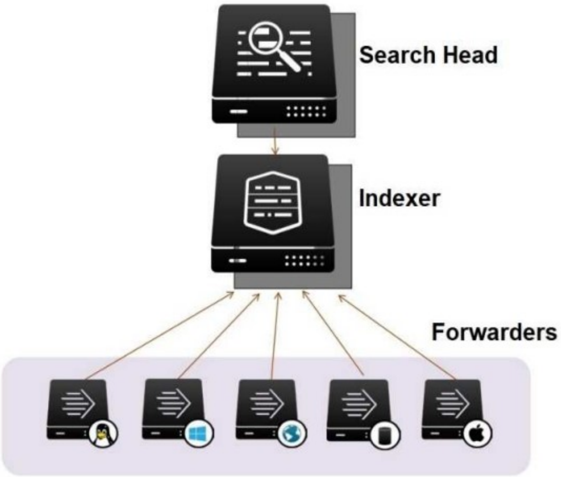


Fig. 5. Splunk Structure

The methodology for implementing the Splunk Enterprise SIEM involves a structured approach utilizing three main components: the Search Head, Indexer, and Forwarder. Each of these elements plays a crucial and distinct role in establishing a comprehensive SIEM solution.

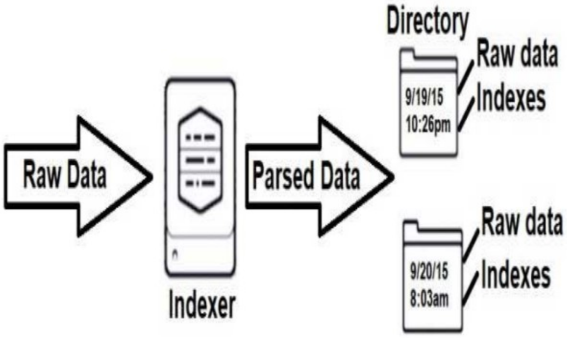
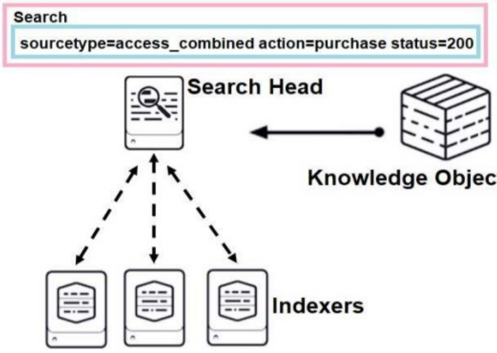


Fig. 6. Splunk Indexer

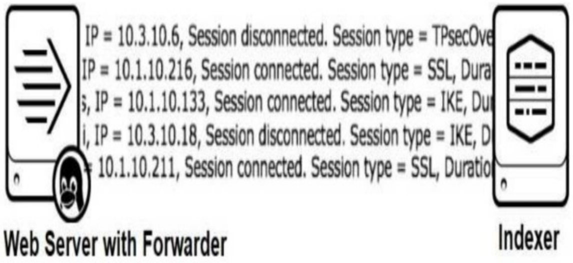
The Indexer is a pivotal component responsible for processing logs and meticulously organizing them into specific indexes. This organizational step is vital as it significantly enhances the efficiency of data analysis and search capabilities. By categorizing logs effectively, the Indexer lays the foundation for in-depth examination of the collected data.



```
sourcetype=access_combined action=purchase status=200
```

Fig. 7. Splunk Search Head

Once the data is appropriately indexed, the Search Head comes into play. This component facilitates querying for various events using the Search Processing Language (SPL). Moreover, it serves as a powerful tool for generating an array of reports, charts, and dashboards. The Search Head is instrumental in translating raw data into meaningful insights, providing a comprehensive view of the security landscape.



```
IP = 10.3.10.6, Session disconnected. Session type = TPsecOve
IP = 10.1.10.216, Session connected. Session type = SSL, Dura
s, IP = 10.1.10.133, Session connected. Session type = IKE, Du
i, IP = 10.3.10.18, Session disconnected. Session type = IKE, D
= 10.1.10.211, Session connected. Session type = SSL, Duratio
```

Fig. 8. Splunk Forwarder

The Forwarder, a crucial component in a Splunk Enterprise SIEM system, is responsible for collecting and forwarding information either to the Indexer or another Forwarder. Notably, it has minimal impact on system performance, consuming minimal resources. There are three types of Splunk Forwarders, each designed for specific functions and resource characteristics. The Splunk Universal Forwarder (SUF) efficiently collects logs from installed machines, forwarding them to the Splunk SIEM or another Forwarder. It has replaced the deprecated Splunk Light Forwarder due to its versatile and resource-efficient nature.

Conversely, the Splunk Heavy Forwarder focuses on forwarding data between Splunk Enterprise instances or to a third-party system, lacking the ability to execute distributed searches. The deprecated Splunk Light Forwarder, used until Splunk Enterprise version 6.0, forwarded data with reduced functionality but consumed fewer resources. Understanding the roles of these components allows the implementation of an effective Splunk Enterprise SIEM system, facilitating the processing, analysis, and visualization of log data to enhance organizational security operations.

D. Apache Spark Based Analytics of Squid Proxy Logs

This paper proposes a Spark based system for Squid proxy log analysis and use it to generate Internet traffic statistics like top domains accessed, top users etc in order to study traffic behaviour and detect threats [6].

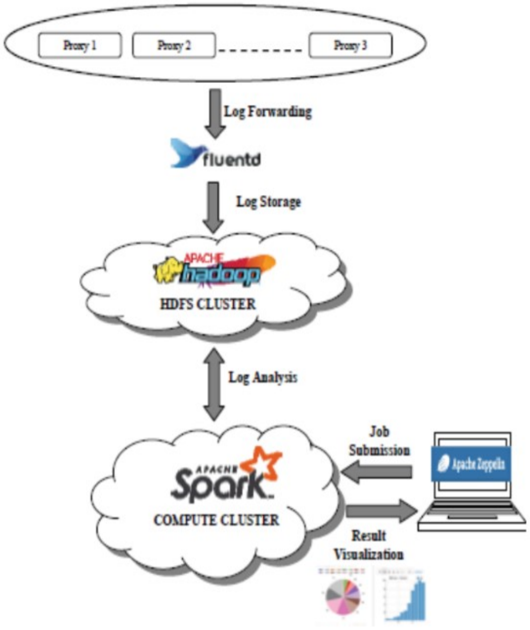
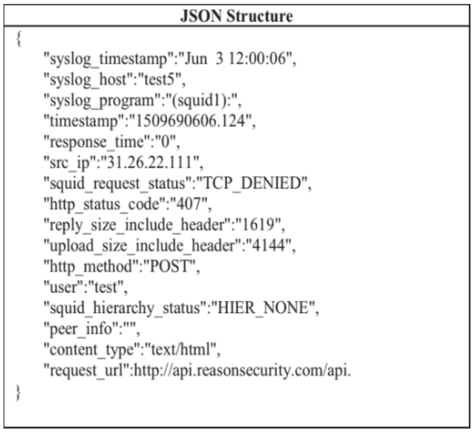


Fig. 9. System Architecture

The methodology employed for the proposed Squid log analysis system using Spark is outlined as follows, illustrated by a schematic diagram in Figure 10. An in-house OpenStack-based cloud infrastructure with five nodes was utilized, including one master node and four slave nodes, each equipped with a 4-core AMD Opteron Processor, 16 GB of memory, and a 40 GB hard disk, all running on Centos 7, 64-bit operating systems.

Open-source software components were exclusively relied upon for log collection, storage, and visualization. This included the Fluentd log collector (version-3.1.1) for log aggregation, Apache Spark computation engine (version-2.2.0) for processing, Apache Hadoop (version-2.7.4) for storage and cluster management, and Apache Zeppelin web notebook (version-0.7.3) for querying and visualization. The process initiated with forwarding logs from multiple proxy servers to the Fluentd log collector, which adeptly transformed them into the structured JSON format. These logs were then seamlessly stored within the Hadoop Distributed File System (HDFS). The analytical phase utilized the Apache Zeppelin web notebook, providing an interactive environment for formulating and executing Spark jobs within the computation cluster. The results were dynamically visualized in various formats, such as graphs and pie charts, creating a cohesive and efficient platform for log data analysis and insightful visualization.



JSON Structure

```
{
    "syslog_timestamp":"Jun  3 12:00:06",
    "syslog_host":"test5",
    "syslog_program":"(squid1):",
    "timestamp":"1509690606.124",
    "response_time":"0",
    "src_ip":"31.26.22.111",
    "squid_request_status":"TCP_DENIED",
    "http_status_code":"407",
    "reply_size_include_header":"1619",
    "upload_size_include_header":"4144",
    "http_method":"POST",
    "user":"test",
    "squid_hierarchy_status":"HIER_NONE",
    "peer_info":"",
    "content_type":"text/html",
    "request_url":http://api.reasonsecurity.com/api.
}
```

Fig. 10. Generated JSON Structure

Furthermore, the JSON structure of the squid logs, as generated by Fluentd, was meticulously defined. This schema served as a foundational reference for subsequent processing and analysis, ensuring consistency and coherence throughout the investigative process. In summary, this methodology, underpinned by a robust technological stack and a carefully orchestrated sequence of operations, laid the groundwork for comprehensive log analysis within the proposed system.

E. Security Analytics and Benchmarking Log Aggregation in the Cloud

The research paper explores the critical topic of securing data in Cloud environments, particularly focusing on the challenges organizations face when migrating to the Cloud. It emphasizes the importance of data protection techniques and security analytics in mitigating risks associated with Cloud deployments. The paper highlights the shared responsibility model between customers and Cloud providers in Platform as a Service (PaaS) Cloud setups, emphasizing the customer's role in ensuring security [13].

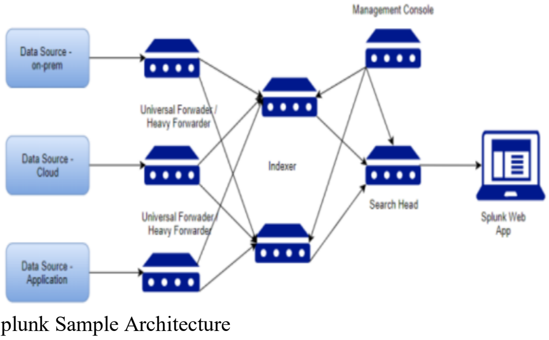

Management Console
Data Source - on-prem
Data Source - Cloud
Data Source - Application
Universal Forwader / Heavy Forwarder
Universal Forwader/ Heavy Forwarder
Indexer
Search Head
Splunk Web App

Fig. 11. Splunk Sample Architecture

Splunk, relying on Python, utilizes Boto3 libraries for accessing logs from AWS buckets. This process involves an adapter file containing IAM credentials, allowing access to the designated AWS account for log retrieval. Subsequently, logs are extracted from the S3 bucket and imported into Splunk. An additional file manages log decryption and the extraction process, as AWS log files are stored in gzip format. Following this, inputs are incorporated into Splunk, facilitating data indexing and display, all transactions occurring in JSON format. To ensure proper categorization, logs are assigned specific source-types based on their type in the 'props.conf' file.

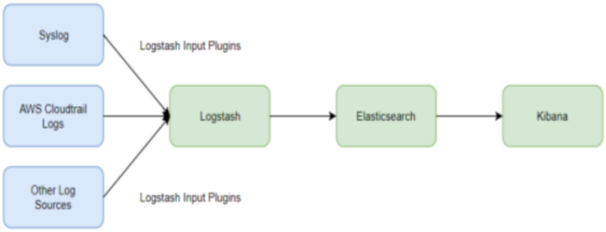


Fig. 12. ELK Stack Sample Architecture

The Elastic Stack comprises three essential components: Elasticsearch, Logstash, and Kibana. Elasticsearch serves as both a search engine and real-time document storage mechanism. Logstash, on the other hand, acts as the data collection engine, responsible for the collection and transportation of logs. Lastly, Kibana provides a graphical user interface that facilitates data visualization.

Amazon Web Services (AWS) offers a diverse array of logs crucial for monitoring the security of the utilized infrastructure. Among these, AWS CloudTrail logs are particularly valuable as they document all API calls made for the account. These logs contain pertinent information such as the user's identity, source IP address, request parameters, and response elements. Config logs in AWS serve to store the AWS resource inventory, notifications regarding configuration changes, and configuration history. AWS Config also provides the capability to automatically create rules. Furthermore, CloudWatch, an AWS monitoring service, accumulates and tracks metrics, in addition to monitoring log files. It also allows users to set alarms should certain services exceed predefined thresholds. By harnessing these logs from various AWS services, a robust security mechanism can be established for the associated cloud accounts. Once these logs are integrated into Splunk, the data can be thoroughly analyzed, providing real-time insights via a centralized view.

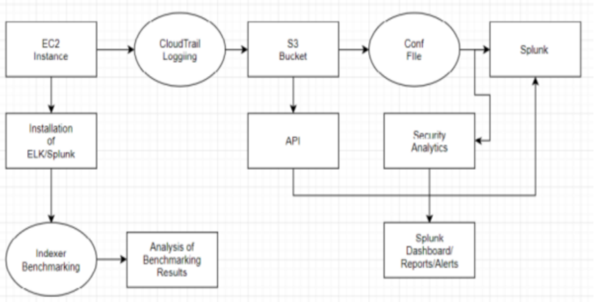


Fig 13. Splunk Sample Authorize VS Revoke activity for security group

Benchmarking exercises were conducted on both on-premise virtual machines and AWS Cloud instances to assess the performance of the Elastic Stack and Splunk. The 'Rally' benchmarking tool was used for the Elastic Stack, generating diverse data types and indexing them into Elasticsearch. Splunk was benchmarked using SplunkIt for data generation and Splunk on Splunk for performance assessment, generating approximately 50GB of Syslog data. A comparative analysis between Splunk and Elastic Stack reveals distinctions in licensing, data shipping, data onboarding, deployment options, query languages, costs, and security features. Splunk is a licensed tool, while Elastic Stack is open-source. Splunk uses forwarders for data shipping, while Logstash and beats are used in Elastic Stack. Data onboarding is more straightforward in Splunk, though data types must be explicitly defined in ELK. Both tools can be deployed on-premise or in the cloud. Splunk employs a proprietary language (Splunk Processing Language), while ELK uses Query DSL with JSON syntax. Costs for both tools include licenses, infrastructure, maintenance, and installation. Splunk integrates security features, while ELK provides Xpack for security at an additional cost. Despite differences, both are popular for log aggregation, with suitability depending on specific requirements.

III. METHODOLOGY

In this section, we propose the methodology employed in establishing our design criteria. These criteria are the result of extensive research and analysis, aimed at enhancing log aggregation in SIEM systems. We detail the specific steps taken to formulate these criteria, ensuring they are well-informed and aligned with the objectives of our study.

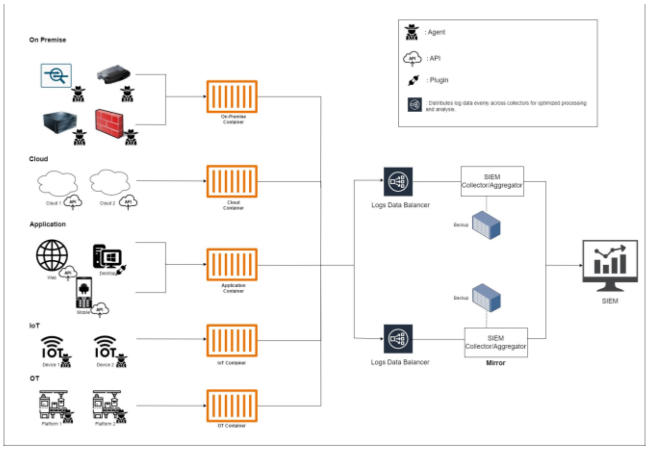


Fig. 14. Robust Log Aggregation Design Criteria

Fig. 15 above shows how the system operates by gathering logs from a variety of sources. This is accomplished through agents installed on multiple devices, as well as interfaces like APIs, plugins, and agents, which facilitate the collection of logs from diverse platforms and environments. To ensure efficient aggregation, the system leverages containerization and a Logs Data Balancer. This combination optimizes real-time detection capabilities. Moreover, the system places a premium on storage reliability by implementing backups and mirrors for the SIEM collector/aggregator, bolstering data security and continuity.

To begin, the log aggregation system should be versatile in data collection, capable of gathering information from a variety of sources. This includes essential inputs like server logs, web application interfaces such as the Twitter API, and external network devices. This diversity ensures that the system comprehensively monitors security. Next, it's crucial to employ a technology stack that handles data efficiently. This stack must adeptly manage data processing, storage, and visualization. Additionally, it should be adaptable to accommodate a wide range of data sources. For seamless event collection, the system should be equipped to gather events from a multitude of sources. This encompasses protocols like syslog, Windows event logs, telemetry hardware, operating systems, and various services.

To enhance automation and scalability, deploying the log aggregation system within Docker containers proves beneficial. This allows for streamlined deployment, management, and scalability, providing a robust foundation for the system. In the face of potential faults, the system should feature mechanisms to ensure reliability. This could involve deploying multiple instances of log aggregation components to balance the load. Such fault tolerance measures are pivotal for uninterrupted operation. Leveraging big data technologies is essential for handling substantial volumes of log data effectively. This capability empowers the system to select and monitor the most productive architecture for optimal performance.

For SIEM functionality, the system must adeptly process data from diverse sources. This includes logs from servers, network security devices, on-premise devices, cloud or external networks and communication or social media applications through APIs whether for desktop, web or mobile. This comprehensive approach guarantees thorough security monitoring. Agents play a pivotal role in data collection and transmission to the central log aggregator component. They enable critical processes like filtering, parsing, and transformation before storage, contributing to the system's efficiency. For systems employing specific log aggregators like Splunk, ELK Stack, or Apache services, it's crucial to adhere to best practices and consider the unique components and features of each tool.

Considering the query language and licensing model of the log aggregator tool is essential. For instance, Splunk employs SPL, while ELK utilizes Query DSL with JSON. These specifics should be factored in during the system design. Lastly, the log aggregation system should be versatile in its deployment options. It should be capable of functioning both on-premise and in cloud environments. This flexibility ensures that the system can be tailored to various deployment scenarios as needed.

IV. CONCLUSION

In conclusion, our research has led to the development of robust design criteria for improving log aggregation in SIEM systems. Through thorough investigation and analysis, we have identified key factors that contribute to the effectiveness of log aggregation, encompassing data sources, compatibility, event collection, containerization, scalability, and integration of big data technologies. Additionally, we've highlighted the importance of SIEM-specific features, agent-based collection, and considerations for specific log aggregator tools. With these criteria in place, we anticipate a significant advancement in the efficiency and reliability of log aggregation processes within SIEM systems. We believe that our findings will serve as a valuable foundation for future advancements in this critical area of cybersecurity.

ACKNOWLEDGEMENT

The authors acknowledge the College of Computing, Informatics and Mathematics for supporting this study.

REFERENCES

[1] P. Sankar, D. E. George and A. S. N. S, "Social media monitoring using ELK Stack," 2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), THIRUVANANTHAPURAM, India, 2022, pp. 231-235, doi: 10.1109/SPICES52834.2022.9774273.

[2] M. Hristov, M. Nenova, G. Iliev and D. Avresky, "Integration of Splunk Enterprise SIEM for DDoS Attack Detection in IoT," 2021 IEEE 20th International Symposium on Network Computing and Applications (NCA), Boston, MA, USA, 2021, pp. 1-5, doi: 10.1109/NCA53618.2021.9685977.

[3] D. D. Mishra, S. Pathan and C. Murthy, "Apache Spark Based Analytics of Squid Proxy Logs," 2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Indore, India, 2018, pp. 1-6, doi: 10.1109/ANTS.2018.8710044.

[4] I. Kotenko, A. Kuleshov and I. Ushakov, "Aggregation of elastic stack instruments for collecting, storing and processing of security information and events," 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), San Francisco, CA, USA, 2017, pp. 1-8, doi: 10.1109/UIC-ATC.2017.8397627.

[5] Pathak, P., Rangasamy, K., &amp; Selvaraj, T. (2018, April 12). Security analytics and benchmarking log aggregation in the cloud. EAI Endorsed Transactions on Cloud Systems. https://eudl.eu/doi/10.4108/eai.11-4-2018.154464.