MASARYK
UNIVERSITY

FACULTY OF INFORMATICS

# Security Log Management System for Small Bussiness

Bachelor's Thesis

KEVIN SEMAN

Brno, Fall 2024

MASARYK UNIVERSITY

FACULTY OF INFORMATICS

# Security Log Management System for Small Bussiness

Bachelor's Thesis

KEVIN SEMAN

Advisor: Mgr. Pavel Novák

Department of Computer Systems and Communications

Brno, Fall 2024


FACULTAS ARTIS INFORMATICÆ
UNIVERSITAS MASARYKIANA

# Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source. In the proccess of writing my thesis, I have used Grammarly to check for errors in the text and enhance my writing.

Kevin Seman

**Advisor:** Mgr. Pavel Novák

# Acknowledgements

I would like to thank my advisor, Mgr. Pavel Novák and my consultant RNDr. Jaroslav Janáček , Ph.D. for their support and guidance throughout the creation of this thesis.

I would also like to thank IstroSec for the provided opportunity and valuable collaboration.

Special thanks go to my family, and my girlfriend for their constant encouragement and support.

# Abstract

This thesis aims to improve the cybersecurity resilience of small companies by providing a tool containing standard functionalities at minimal cost. In the first part, various related concepts, such as log management, regulatory requirements, and detecting malicious behavior, are introduced. In the second part, open-source components are compared. Components are divided into three categories based on functionality. In the end, a PoC solution is created using the chosen components and enhanced by automatizing parts, which can speed up the deployment.

# Keywords

Log management, SIEM, OpenSearch, Fluentd, Sigma

# Contents

# List of Tables

# List of Figures

# Introduction

Based on Mandiant's research, 54% of organizations discover their compromise through external sources such as law enforcement agencies, cybersecurity companies, or even attackers (1), which implies companies' lack of monitoring and detection capabilities. Small to medium-sized enterprises often lack dedicated security teams and cannot achieve adequate infrastructure visibility to detect cyber attacks due to the high cost of commercial products and missing knowledge and experience in deploying and maintaining solutions. In addition, regulated entities are mandated to implement log management and monitoring solutions due to the expansion of cybersecurity-related directives.

This thesis aims to research and compare available open-source components to design and implement a system that enhances visibility in the company's infrastructure. Developed in cooperation with IstroSec, the solution focuses on providing a practical and user-friendly platform to help small enterprises meet legal requirements, aid IstroSec's incident response team in managing and analyzing security events, and provide actionable security information to integrate into IstroSec's developed products.

The selection of components used to create the end solution will be based on the support of required functionalities, compatibility with other components, ease of use, and IstroSec's familiarity with the querying and correlation language used in the components.

In the first part of the thesis, key concepts of log management and SIEM are introduced, as well as regulatory and compliance standards, which mandate the deployment and operation of these components. In the second part of the thesis, a comparison of open-source components is conducted, focusing on three core categories: collection, indexing, and visualization. Next, a Proof-of-Concept is created based on selected components and tested against common cybersecurity attacks. Logs gathered from tested attacks are analyzed to develop detection rules. Parts of the components are automated using deployment scripts to prepare correct ingestion of various logs, set retention policies, set test notification channels, detection rules, and create visualizations.

**IstroSec** is a cybersecurity company based in Bratislava, Slovakia and primarily focuses on digital forensics and incident response. The company operates its research and development department to develop innovative solutions integrating insights from solving complex incidents and the expertise gained from conducting complex red team operations (2). The thesis cooperation is part of a project nicknamed "Akasa".

# 1 Key concepts of Log Management in Cybersecurity

## 1.1 Log Management

### 1.1.1 Logs and their content

The simplest and most generalized description of a log would be the record of something happening somewhere at a time. Although different types of logs may contain various types of information, a log message should include information about the event that occurred (including but not limited to user authentication and application exception), the source of the event (such as hostname and IP), and the time that the event occurred (timestamp). Security logging is the process of recording events that may help detect security issues such as unauthorized access, malicious events, or misuse of resources. Operational logging is an extensive category that provides information such as failures and actionable conditions for operators. Compliance logging focuses on regulations, mandates, and industry standards such as PCI DSS and ISO/IEC 2700X. Application logging is usually available in the development environment and provides valuable information for developers regarding the development and troubleshooting of the system (3, Ch. 2, pp. 29-32).

Logs progress through a life cycle that begins with their generation by systems and applications, after which the log is collected by a collector, which can forward the log or process it. Received logs from multiple sources are parsed, filtered, and normalized to extract valuable information. The information these logs provide is stored and made available for analysts during incidents and investigations. The number of logs can be overwhelming sometimes; visualizations should help analysts get a quick overview of what is happening without the need to go through each log.

Events extracted from logs can be enriched with external information or correlated with other events to provide better context for the analyst. Alerting rules can be implemented to notify responders in an emergency for quick response and remediation. In the end, old

logs that no longer provide any value should be discarded or archived based on retention policies for future investigations.

Inconsistent log sources may record various attributes and log data in varying structures and representations. Another inconsistency arises from variations and inaccuracies in the timestamps of logs collected from multiple hosts, for example, due to a lack of time synchronization. Text files, database entries, syslog, SNMP, XML, and binary files are some of the many different log formats that log sources use. Logs are designed to be read by humans or machines and use standard (e.g., JSON, CSV) or proprietary formats. These inconsistencies hinder the effective linking of events across multiple sources (4, Sec. 2.3).

### 1.1.2 Log collection

The core of effective log management is the systematic collection of log data from various sources. Log collection includes components that generate log messages and components that collect these messages. Log data collection may be initiated by the creator component (push-based collection) or the collector component (pull-based collection).

An example of a pull-based log collection is collector-initiated Windows Event Forwarding, where the collector connects to each endpoint to gather the required data. In the case of WEF, the collector subscribes to gathered events on each predefined client and periodically requests these events from the endpoint. An example of a push-based log collector is a source-initiated Windows Event Forwarding or Syslog, where endpoints push log messages to the collector without waiting for a request. In the case of WEF, first, the endpoint contacts the predefined collectors for available subscriptions. The subscription details are stored on the endpoint, which periodically sends a heartbeat to the collector and requested events, if available, as visualized in the figure (1.1) (5).


**Figure 1.1:** Simplified visualization of pull-based (source initiated) and push-based (collector initiated) log collection in Windows Event Forwarding

### 1.1.3 Log filtering, normalization and aggregation

Not all log messages provide real value based on our goals. Log filtering is the process of suppressing logs that are unlikely to contain relevant information to focus on more critical data. Filtering is usually conducted without altering the generation of original log files and may be present in other stages of log management. Log aggregation is the process of grouping similar events into a single entry. Aggregation helps to reduce the volume of information by preserving its overall value. Log normalization focuses on transforming gathered data into predefined data representation across multiple sources. An example of log normalization may be the conversion of timestamps, which can be recorded in multiple formats. Log normalization helps analysts by providing a consistent data representation (4, Sec. 3.2).

### 1.1.4 Log correlation

Correlation is the process of linking together various events based on matching or similar data to get a more complete overview. Events can be linked using correlation rules. For example, the creation of a new user and the immediate addition to the Domain Admins group on the Windows Active Directory may indicate a potential malicious activity of an attacker establishing persistence. Or the presence of multiple failed login attempts followed by a successful login, which may indicate brute force attacks by an adversary as visualized in (1.2).

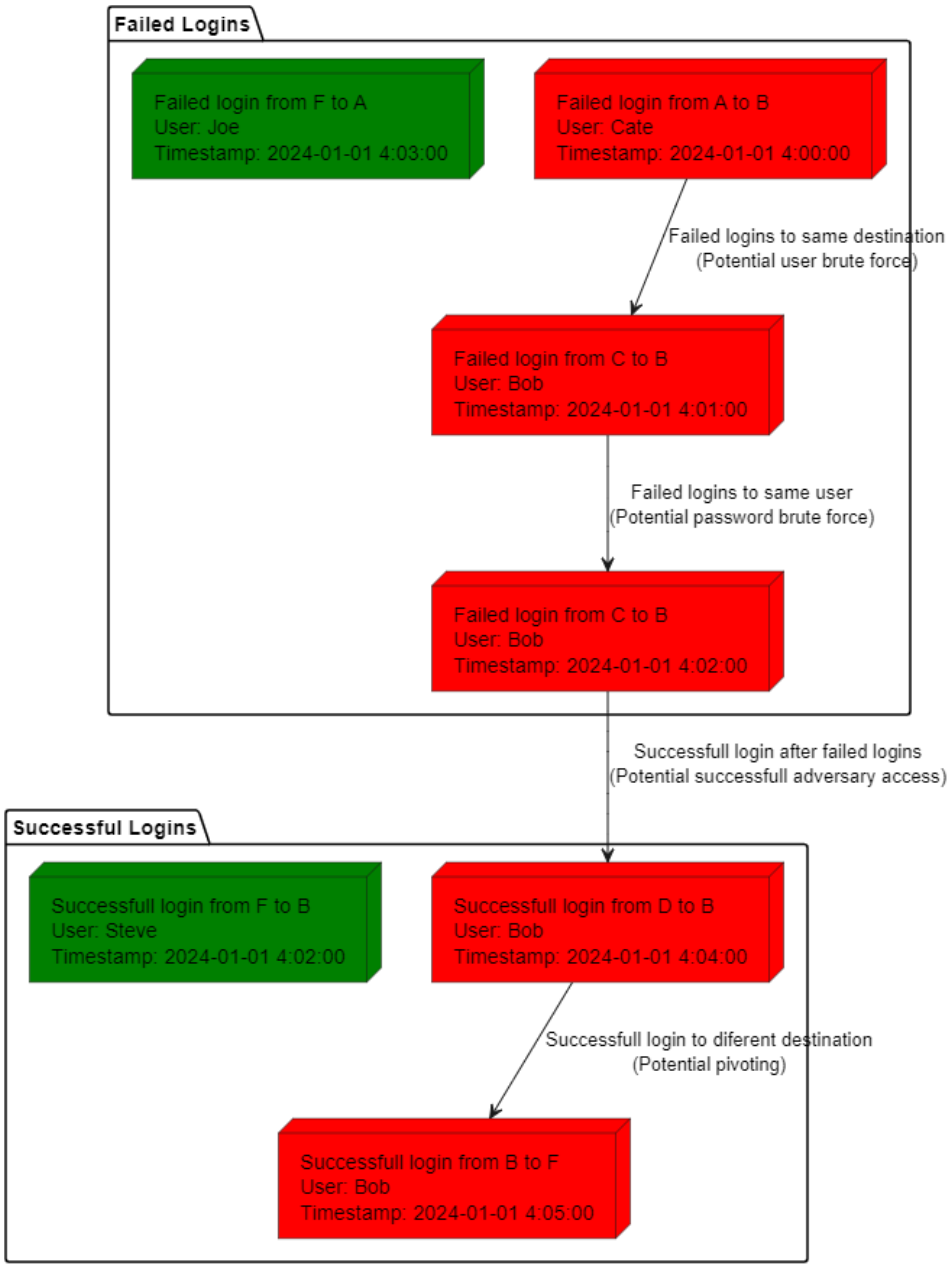


**Figure 1.2:** Simplified visualization of correlating logins to detect potential malicious activity

Micro-level correlation (also referred to as atomic correlation) is focused on linking together events to identify patterns or anomalies within a localized scope. For example, the correlation of multiple failed logins within a short period of time may indicate suspicious behavior. Macro-level correlation (also referred to as fusion correlation) is focused on enhancing events based on additional sources. For example, weblogs can be combined with information from vulnerability assessment tools that may indicate the presence of a vulnerability or correlate threat intelligence information with network traffic to search for malicious IPs or domains (3, Ch. 9, pp. 155-159).

## 1.2 Regulatory and Compliance Standards

In the coming months, a growing number of entities will become regulated, primarily driven by the expansion of sectors under the new NIS2 directive. In addition to NIS2, other regulations such as GDPR and DORA impose requirements that imply the need for log management and monitoring from regulated entities. These regulations raise considerable challenges for entities that need the required knowledge to implement these solutions or lack the resources to maintain these solutions. While not legally binding, beyond the legal frameworks, standards such as PCI DSS and ISO/IEC 27001 provide guidelines and best practices for organizations.

### 1.2.1 Local Cybersecurity Legislation

**Act No. 69/2018 Coll. on cybersecurity and on Amendments and Supplements to certain Acts** The Act came into effect on April 1, 2018, in the Slovak Republic and transposed the European directive on network and information security (NIS Directive). Regulated subjects include operators of essential services, which consist of critical infrastructure, the information systems of public administration, or services in the defined sectors and subsectors that rely on networks and information systems. Although it does not directly require providers of essential services to implement log management, it includes "Cybersecurity incidents detection" and "Cybersecurity incident evidence" as required security measures (6).

**Act No 95/2019 Coll. on information technology in public administration and relevant implementing rules** Implemented in the Slovak Republic based on the conditions under Directive (EU) 2016/2102. The act establishes the organization and management of information systems, the rights and obligations of governing and managing authorities, and the minimal requirements for information systems in public administration and their management. Section 21 of the act requires the administrator of information systems in public administration to "ensure continuous monitoring of the public administration information system" (7).

**Act No 181/2014 Coll. on Cyber Security and Change of Related Acts** The Act came into effect on July 23, 2014, in the Czech Republic and transposes the NIS Directive. The Act requires regulated subjects to introduce and implement security measures, which, among others, include technical measures such as "A tool for recording the activities of the information or communication system, its users and administrators," "A tool for detecting cyber security events," and an "A tool for collecting and evaluating cyber security events," (8).

### 1.2.2 European Union

**DIRECTIVE (EU) 2022/2555 (NIS 2 Directive)** came into effect on January 16, 2023. Among many improvements, NIS 2 expands upon NIS in the scope of regulated sectors and entities and introduces important entities among essential entities with different levels of requirements. NIS 2 also provides a detailed list of minimal security measures opposed to NIS, such as basic cyber hygiene practices, incident handling, and supply chain security (9).

**Digital Operational Resilience Act (DORA)** will come into effect on January 17, 2025. The regulation requires member states to enforce by Article 16 that regulated subjects shall "continuously monitor the security and functioning of all ICT systems" and by Article 17 to "establish procedures to identify, track, log, categorize and classify ICT-related incidents." (10)

### 1.2.3 Industry-Specific standards

**Payment Card Information Data Security Standard (PCI DSS)** PCI Security Standards apply to all entities that store, process, or transmit cardholder data. Requirement 10 of PCI DSS, which describes tracking and monitoring access to data and resources, defines the implementation of audit trails for each event on each system component. Audit trail events should, at minimum, include "user identification, type of event, date and time, success or failure indication, origination of event, and identity or name of affected data, system component or resource." Beyond Requirement 10, other requirements imply the need to implement log management. The standard also defines an audit data retention policy of a minimum of one year, with audit data from the last three months being immediately available for analysis (11).

**ISO/IEC 27001/27002** A voluntary standard for information security management systems that defines requirements designed to protect sensitive information. The standard defines Logging in Annex A as "Logs that record activities, exceptions, faults, and other relevant events shall be produced, stored, protected and analyzed." and Monitoring activities "Networks, systems, and applications shall be monitored for anomalous behavior and appropriate actions taken to evaluate potential information security incidents." (12) ISO 27002 defines that event logs should include user IDs, system activities, dates, times, device identity and network address and protocols. Also events which should be logged include successful and failed system and resource access attempts, changes to system configuration, use of privileges, access and deletion of important files, manipulation with security systems, identity manipulation (13, Sec. 8.15).

## 1.3 Essential security information and event management functions

### 1.3.1 Alerting

Analysis of collected events can be overwhelming. Companies usually do not have the required personnel to monitor their logs 24/7. SIEMs should provide the option to detect potentially malicious behavior and

notify key operators automatically. Alerting can be triggered by rules in the SIEMs rule engine. Rules are predefined descriptions of related events in a given time that may indicate adversary activity. Rules are usually written to match adversary activity based on standard frameworks such as MITRE ATT&CK (14) or Cyber Kill Chain (15).

Sigma rule format provides a human-readable format for defining detection and alerting rules, which can be easily converted into queries for various SIEMs. Sigma rules are a collection of community-driven rules created to detect threat actors. Rules are written in YAML format, and each rule contains the description of malicious activity, the source of logs on which the detection should be monitored, and the detection logic that will trigger the detection (16).

**Listing 1.1:** Example sigma rule for detecting Active Directory information gathering using Adalanche

```
title: Adalanche execution
id: <<exampleID>>
status: experimental
description: Detects command line parameters
   used by Adalanche
references:
    - https://github.com/lkarlslund/Adalanche
author: <<exampleAuthor>>
date: 2022-12-20
modified: 2024-12-20
tags:
    - attack.t1087.001
    - attack.t1087.002
    - attack.t1482
    - attack.t1069.001
    - attack.t1069.002
logsource:
    category: process_creation
    product: windows
detection:
    selection_img:
        - Image|contains:
```

```
            - '\adalanche-collector-windows.
              exe'
    selection_cli:
        CommandLine|contains:
            - ' collect activedirectory '
    condition: 1 of selection_*
falsepositives:
    - Other programs that use these command line
      options
level: high
```

### 1.3.2 Reporting

SIEMs should provide the option to generate detailed reports summarizing security incidents and user activities. These capabilities play a critical role in meeting regulatory requirements by providing evidence of log analysis and monitoring. Compliance reports give a detailed overview of user activities aligning with regulatory requirements. Operational reports provide insights into system performance. Incident reports provide a structured overview of an incident, such as related malicious activities and alerts. (17).

### 1.3.3 Retention

Storing a large amount of data can be costly; however, keeping more logs can provide a more precise overview when investigating incidents and malicious activities. Log rotation manages log file lifecycles by renaming, compressing, or archiving files that reach a predefined size or age, ensuring continuous logging without overwhelming storage (18).

Log retention is storing logs for a specified period, ensuring the information provided by these logs is available in case of investigations. Different log types may have various log retention policies; for example, security logs may provide more useful information in case of emergency than operational logs. Regulations or local authorities may define a minimum retention period for event data. In the case of PCI DSS, the required time for log retention is at least one year; on the

other hand, GDPR requires storing logs of operations with personal information for at least two years. (19).

## 1.4 Identifying malicious activity

### 1.4.1 Common frameworks

**MITRE ATT&CK Framework** A model for categorizing adversary behavior into tactics may include multiple techniques. The framework includes tactics such as reconnaissance, initial access, execution, persistence, privilege escalation, lateral movement, command and control, and exfiltration (14).

**Cyber Kill Chain (Lockheed Martin)** A framework for identification and prevention of adversary behavior. The framework includes seven steps of an attack: Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control, and Actions on Objectives (15)

**Unified Kill Chain** Designed to combine and enhance MITRE ATT&CK Framework and the Cyber Kill Chain. The framework provides 18 attack stages divided into three phases: "Initial Foothold," "Network Propagation," and "Action on Objectives" (20).

### 1.4.2 Common tactics and stages

**Reconnaissance** In this phase, the adversary's primary goal is to gather information passively or actively about the target. Information may include the external threat interface of the target, such as publicly available services and used technologies or information about employees, organizational structure, and data leaks from infostealers and data markets. An adversary may use simple Google Dorking, social media, or active scanning of external IP addresses of the target (21).

**Weaponization/Resource development** Based on the information gathered in the previous stage, the adversary prepares tools and exploits infrastructure to execute the attack. Activities may also include

compromising and stealing non-target assets, which may aid the adversary in compromising the target (14).

**Delivery** The adversary uses various techniques to transmit the created objects to the target, including sending payloads to an externally available component or phishing attacks targeted at employees. Waterholing attacks include compromising a legitimate website, which is often visited by the target, and preparing the malicious payload on this site (20, Sec. 2.5.2).

**Social Engineering** A method that encompasses techniques designed to manipulate individuals into performing actions helpful for the adversary. These actions may provide helpful information to the adversary or allow access to restricted environments. One form of using social engineering is phishing emails, where the adversary usually tries to misuse emotions of fear or curiosity. The adversary may flag the email as time-sensitive (the target needs to do something in the next 24 hours, or his account is locked, or money is lost) and provide a malicious fake website to steal user information (22, pp. 35-39).

**Exploitation** Use prepared malicious objects and payloads delivered to the target environment. The exploitation occurs when the adversary misuses a bug in the software system. A system vulnerability may allow the adversary to bypass authentication, upload malicious payloads, or execute code remotely(23, pp. 178-180).

**Persistence** Granting presence for the adversary in the target environment. Usually, adversaries try to create multiple persistence mechanisms to retain access even in case of identification by the incident response team. Standard persistence techniques include creating accounts, scheduled tasks, and remote access applications (14).

**Defense Evasion** The technique encompasses tactics to bypass security measures deployed on the target environment. Activities may include turning off security measures, obfuscating malicious data, or using Living of the land binaries, which contain tools administrators use in everyday operations (24).

**Command and Control** The adversary establishes a communication channel from a compromised system in the target environment to a system controlled by an adversary outside the target environment. Adversaries may use common domains and services to hide command and control activity, such as GitHub, Twitter, Trello, Telegram, and Slack (25).

**Privilege Escalation** Privilege escalation includes exploiting already achieved environmental capabilities for higher capabilities and privileges. Activities may include exploiting relation misconfigurations or searching for cleartext data in the environment. Adversaries may also use vulnerabilities in applications or services to gain higher privileges (26).

**Lateral Movement** An active adversary in the infra may hop between different hosts and systems to gather information and achieve better visibility. Adversaries may use valid accounts to interact with remote systems or use techniques such as "pass the hash" and "pass the ticket" without needing the knowledge of the cleartext password of the target (14).

**Exfiltration** Exfiltration is transporting targeted data, such as financial or personal information, to systems controlled by adversaries. Adversaries may use common applications or domains to stealthily exfiltrate data from the data environment, such as mega.nz, Google Drive, Dropbox, or GitHub (25).

# 2 Research and comparison of open source components

This chapter compares open-source components suitable for building a SIEM for small businesses. The components are divided into three categories based on functionality: Collect, Store, and Visualize

## 2.1 Architecture

### 2.1.1 Collector

The "collector" component will be responsible for collecting, aggregating, normalizing, and filtering logs from diverse sources, including Windows Event Log, Syslog, and NetFlow. As seen in figure 2.1, Windows Event logs' initial collection and forwarding will be conducted using Windows Event Forwarding. The collection of Windows event forwarding will be set up using Active Directory GPOs. For Syslog and NetFlow, listeners will be created to collect these logs. Candidates for this component are Fluentd, Fluent-bit, Rsyslog, Logstash OSS, Beats OSS, Promtail, Grafana Agent, Syslog-ng OSE, and NXLog Community edition.

### 2.1.2 Indexer

The "Indexer" component will be responsible for ingesting collected data from the collector component. The component should be suitable for easy querying of everyday information and also available for potential threat hunting in case of emergencies. Candidates for this component are ElasticSearch, OpenSearch, Loki, and SigNoz.

### 2.1.3 Dashboard

The "Visualize" component will be responsible for managing and visualizing ingested data by the indexer component. This component should also provide functionalities such as reporting, creating correlation rules, alerting, and providing a configurable data retention process. Candidates for this component are Kibana, OpenSearch Dash-

boards, Grafana, and Graylog Open. Each tool will be evaluated for ease of use, flexibility in creating visualizations, and support for required functionalities. Dashboards should provide key operators with a quick overview of the ongoing activities in the infrastructure.

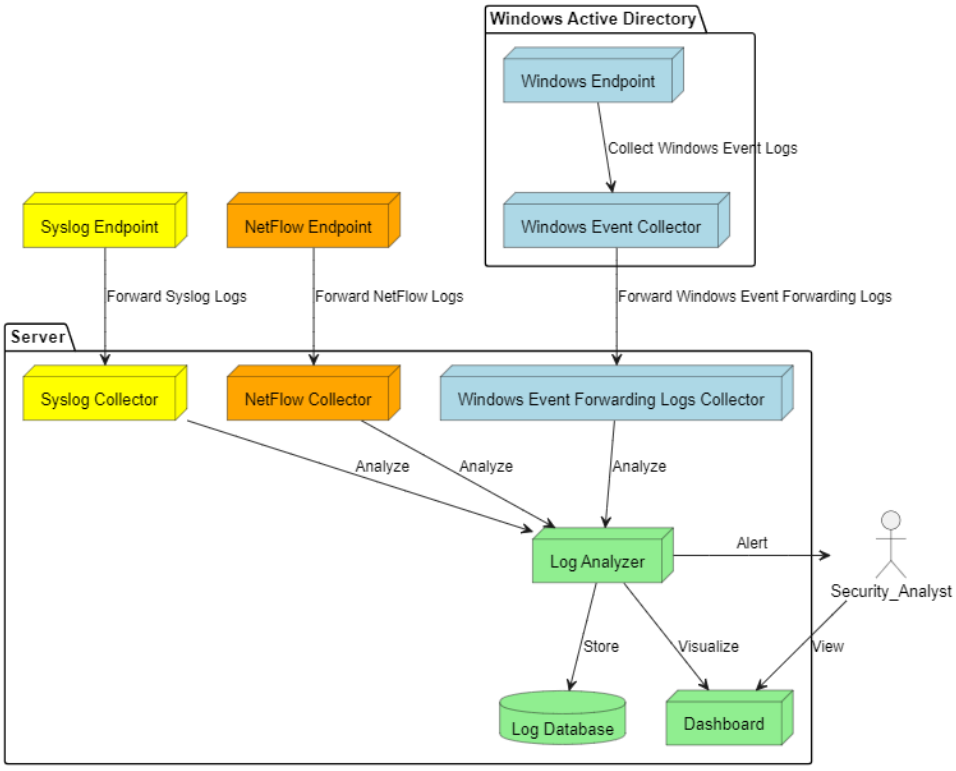


**Figure 2.1:** Akasa deployment diagram

## 2.2 Collected logs

### 2.2.1 Windows Event Log

Event logging is a process by which the Windows operating system records various events and stores them in an event log file. The event log is a collection of recorded events that can be viewed using the Event Viewer or accessed through the programming interface. Five types of Windows event logging include Information, Warning, and Error event types, which describe problems or successful operations

across applications, systems, and services. Success Audit and Failure Audit focus on security-related events such as login or access attempts. Multiple software sources create event logs, as seen in the left panel of 2.2. The security log is reserved for system use only, while the system log is reserved for device drivers. Application logs and custom logs should contain event logs related to different applications. (27).

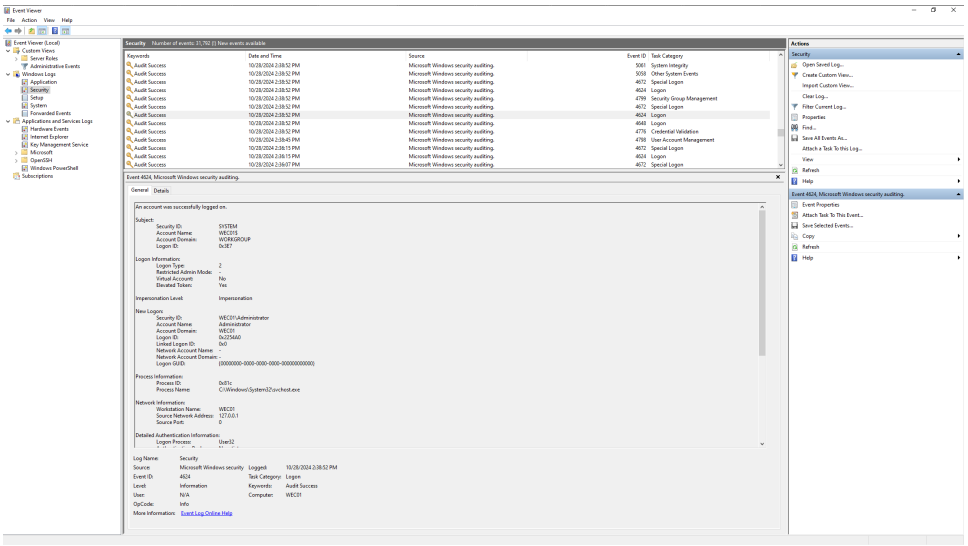

**Figure 2.2:** Windows event log from Security source viewed in Event Viewer

Windows Event Logs from endpoints are forwarded to a Windows Event Collector (WEC) using Windows Event Forwarding (WEF). Subscriptions define collections of events that the WEC should collect. Microsoft's documentation provides two commonly recommended subscriptions: Baseline (which should be applied to all hosts) and Targeted (which should be applied to selected hosts). Collection initiated by a WEC server is called pulling. In a pull scenario, the subscription configuration is present on the WEC server, along with a list of WEF client names from which the Windows Event Logs should be collected. Collection initiated by the WEF clients is called pushing. In a push scenario, the subscription configuration is distributed using manual settings or Group Policy Objects (GPOs), including the address of the WEC server (5).

### 2.2.2 Syslog

Syslog messages can be transported using UDP and TCP. Each Syslog message includes a facility value in the range of 0 to 23 that determines the message's source. For example, facility zero stands for kernel messages, and facility eleven stands for FTP daemon. Each message also has a severity value from zero (emergency) to seven (debug). Hostname identifies the machine that generated the syslog message. The identifier may be described using FDQN, static IP, hostname, dynamic IP, or NILVALUE. The message part of Syslog is not standardized, which provides a considerable challenge for parsing and gathering valuable information (28)

### 2.2.3 Netflow

NetFlow is a protocol developed by Cisco that collects network traffic. The protocol provides detailed traffic data, including IP addresses, ports, and protocols, and is widely used for security and network capacity monitoring. There are multiple versions of NetFlow, with the latest major version introduced by Cisco being version 9, which also supports custom fields. IPFIX is based on NetFlow version 9, and it is focused on being a universal standard for IP flow information gathering. (29)

## 2.3 Collect

This section focuses on open-source components that are suitable for the collection and forwarding of the specified logs. Each description is based on the component documentation. The most essential attributes of each compared component are

- Compatibility (input modules) with Syslog, Windows Event Logs, and NetFlows
- Compatibility (output modules) with components from the "Store" category
- License, compatible with possible commercial closed-source use in the future

### 2.3.1 Fluentd

Created by Treasure Data, Inc. in 2011, Fluentd is an open-source C and Ruby data collector. Fluentd acts as a Unified Logging Layer, integrating various aspects of log data management, which includes collecting, filtering, buffering, and delivering logs from a wide range of data inputs and outputs. Using the data input plugins, it supports sources, such as Netflow (fluent-plugin-netflow) (30), Syslog (in_syslog), and Windows Event Logs (obsolete in_windows_eventlog and in_windows_eventlog2) (31). It can also forward logs to ElastiSearch (out_elasticsearch), OpenSearch (out_opensearch), Grafana Loki (fluent-plugin-grafana-loki) (32), and SigNoz (33) among many other data output plugins. The vanilla instance requires 30-40MB of memory and Ruby on the host machine. Fluentd also has multiple prepackaged versions, such as fluent-package (maintained by Treasure Data, Inc.) and calyptra-fluent (maintained by Calyptia, Inc.) (34).

### 2.3.2 Fluent-bit

Fluent-bit is a lightweight version of Fluentd, created by Eduardo Silva in 2014. As Fluentd, it is also licensed under the Apache License 2.0. Although based on Fluentd architecture and design, they have some differences. Fluent-bit is entirely written in C and requires approximately 1MB of memory. It also provides a higher performance and zero dependencies with the trade-off of having fever-available plugins compared to Fluentd. The data pipeline of Fluent-bit consists of Input, Parser, Filter, Buffer, Router, and Output. Among the many data inputs, it supports Syslog (Syslog) and Windows Event Logs (winlog and winevtlog) but does not support NetFlows. Output plugins support forwarding, for example, to ElasticSearch (es), OpenSearch (open search), Grafana Loki (Loki) (35).

### 2.3.3 Rsyslog

Rsyslog is a GPLv3 system for log processing, which was forked from the sysklogd standard package in 2004. Forwarding is based on rulesets. For each rule, an evaluation is based on a filter followed by the corresponding action until the message is discarded (discard action or after processing the last rule). Modules manage input and output

sources. Input modules include Syslog and Network Traffic Capture and can support Windows Event Logs via Syslog using the Rsyslog Windows Agent. Output modules include support for Elasticsearch (omelasticsearch), which can also work with OpenSearch although not stated (36).

### 2.3.4 Logstash OSS

A limited Logstash version that only includes features licensed under the Apache 2.0 license (37). Logstash supports over 200 plugins. Input plugins support sources such as file (file), beats (input-elastic_agent), and syslog (syslog). Logstash has a deprecated NetFlow module (38) and relies on Filebeats NetFlow module which is not availlable in the OSS version. Logstash requires winlogbeat for Windows event log collection. Logstash provides collection of Syslog messages using the Syslog input plugin. Output plugins support destinations such as elasticsearch (logstash-output-elasticsearch), opensearch (logstash-output-opensearch) (39), Grafana Loki (logstash-output-loki) (40) (41)

### 2.3.5 Beats OSS

Beats OSS features a family of multiple data shippers, which only includes features licensed under the Apache 2.0 license. Each Beat supports a different data type (42). Filebeat OSS collects logs and other data (Syslog, NetFlow) [1]. Metricbeat OSS collects metric data. Packetbeat OSS collect network data [2]. Hearthbeat OSS is used for uptime monitoring [3]. Winlogbeat OSS collects Windows event logs [4]. Auditbeat OSS collects audit data [5].

### 2.3.6 Promtail

Promtail was created to forward logs to Grafana Loki instances. Sources that are called targets are restricted to local log files and systemd jour-

1. https://www.elastic.co/downloads/beats/filebeat-oss
2. https://www.elastic.co/downloads/beats/packetbeat-oss
3. https://www.elastic.co/downloads/beats/hearthbeat-oss
4. https://www.elastic.co/downloads/beats/winlogbeat-oss
5. https://www.elastic.co/downloads/beats/auditbeat-oss

nal. After setting the targets, Promtail will continuously read the logs from them (tailing) (43).

### 2.3.7 Grafana Agent

An OpenTelemetry collector created under the Apache License 2.0 that supports data collection, transformation, and forwarding data to the Prometheus, OpenTelemetry, and Grafana open-source ecosystem. (44).

### 2.3.8 syslog-ng OSE

Syslog-ng Open Source Edition (syslog-ng OSE) is a versatile and scalable logging application tailored for centralized logging solutions, licensed under the combination of the GPL and LGPL licenses. Source drivers support syslog (syslog()) and files (file()). Destination drivers support elasticsearch (elasticsearch-http()) and graylog (graylog2()). Capabilities include filtering, routing, collating, and enriching log messages. (45).

### 2.3.9 NXLog Community Edition

NXLog Community Edition is released under the NXLog Public License. Input modules support among other Syslog (xm_syslog), file (im_file), Windows Event Logs (im_msvistalog), and NetFlow (xm_netflow parser). Output modules include support for ElasticSearch (om_elastisearch), which can be configured to work with OpenSearch though this functionality is not explicitly documented, Graylog (xm_gelf), (46).

### 2.3.10 Summary

Based on table (2.1), the most versatile components for collecting Syslog, Windows Event Log, and NetFlow are Fluentd, Beats OSS, and NXLog CE. Although Fluentbit and Logstash OSS do not directly support some of the collected logs, they are part of their relative family (Fluent and Beats), which provides the necessary inputs.

Based on table (2.2), the most versatile components for forwarding to the "Store" component are Fluentd, Fluenbit, and Logstash OSS. Although Beats OSS does not directly support some of the mentioned

components, it is part of the Logstash family that provides the necessary output.

**Table 2.1:** Collect component input compatibility matrix

| Component | Syslog | Windows Event Log | NetFlow |
|---|---|---|---|
| Fluentd | ✓ | ✓ | ✓ |
| Fluentbit | ✓ | ✓ | x |
| Rsyslog | ✓ | ✓ | x |
| Logstash OSS | x | ✓ | x |
| Beats OSS | ✓ | ✓ | x |
| Promtail | ✓ | ✓ | x |
| Grafana Agent | ✓ | ✓ | x |
| syslog-ng OSE | ✓ | x | x |
| NXLog CE | ✓ | ✓ | ✓ |

**Table 2.2:** Collect component output compatibility matrix

| Component | ElasticSearch | OpenSearch | Loki OSS | SigNoz |
|---|---|---|---|---|
| Fluentd | ✓ | ✓ | ✓ | ✓ |
| Fluentbit | ✓ | ✓ | ✓ | ✓ |
| Rsyslog | ✓ | - | x | x |
| Logstash OSS | ✓ | ✓ | ✓ | ✓ |
| Beats OSS | ✓ | x | x | x |
| Promtail | x | x | ✓ | x |
| Grafana Agent | x | ✓ | x | x |
| syslog-ng OSE | ✓ | x | x | x |
| NXLog CE | ✓ | ✓ | x | x |

## 2.4 Store

This section focuses on open-source components that could store the gathered data and provide search functionality. The essential attributes of each compared solution are compatibility with the components from the visualize category, license compatibility with possible commercial closed source use, and query language support used to retrieve data efficiently.

### 2.4.1 Apache Soir

Apache Soir is an Apache Lucene-based search platform created under Apache License 2.0. Lucene provides full-text search capabilities. Apache Soir supports Domain Specific Language (DSL) and SQL. (47).

### 2.4.2 OpenSearch

The community-driven Amazon project started by forking ElasticSearch. OpenSearch is an Apache Lucene-based search and analytics engine created under the Apache License 2.0, with features such as searching by field, searching multiple indexes, sorting results by field, and aggregating results. Domain-specific language (DSL) is used to query data in OpenSearch (48).

### 2.4.3 ElasticSearch

Elastic released ElasticSearch under the Apache 2.0 license until version 7.10.2; newer versions are dually licensed under Elastic and Server Side Public License (SSPL). Elastic created ElasticSearch as part of the Elastic Stack. It is also compatible with Grafana and OpenSearch Dashboards for visualization. ElasticSearch uses Domain Specific Language (DSL) to query data. It also supports Event Query Language (EQL), Elastisearch Query Language (ES|QL), Lucene query and SQL (49).

### 2.4.4 Grafana Loki OSS

Loki is an AGPL log aggregation system inspired by Prometheus. The typical Loki stack includes Grafana for querying/displaying data and Promtail, or Grafana Agent, which gathers and forwards logs to Loki. Loki's query language (LogQL) is used to query data. The ruler is a component responsible for constantly evaluating queries, after which an action is performed (50).

### 2.4.5 SigNoz

SigNoz is an observability tool released under Apache License 2.0 that uses ClickHouse to store logs. Capabilities include filtering and querying logs. The query language used by SigNoz is a simplified version of SQL (Old Logs Explorer) and JSON filters (New Logs Explorer) (51).

### 2.4.6 Results

Table 2.3 shows the compatibility of analyzed components with components to visualize data. As can be seen, OpenSearch and ElasticSearch are the most versatile visualization compatibilities. Except for Apache Soir, each of the compared solutions has a primary visualization component that belongs to the same stack. Signoz provides visualization in a single component.

Table 2.4 shows the analyzed component's supported query languages. Although ElasticSearch provides support for most query languages, Apache Soir, OpenSearch, and Signoz provide the most license freedom with their Apache 2.0 license.

**Table 2.3:** Store components visualization matrix

| Component | OS Dashboards | Grafana OSS | Kibana | Graylog Open |
|---|---|---|---|---|
| Apache Soir | x | ✓ | x | x |
| OpenSearch | ✓ | ✓ | x | ✓ |
| ElasticSearch | x | ✓ | ✓ | ✓ |
| Loki OSS | x | ✓ | x | x |

**Table 2.4:** Store components query matrix

| Component | DSL | SQL | LogQL | PPL | EQL | ESQL |
|---|---|---|---|---|---|---|
| Apache Soir OSS | ✓ | ✓ | X | X | X | X |
| OpenSearch OSS | ✓ | ✓ | X | ✓ | X | X |
| ElasticSearch | ✓ | ✓ | X | X | ✓ | ✓ |
| Grafana Loki OSS | X | X | ✓ | X | X | X |
| Signoz | X | ✓ | X | X | X | X |

## 2.5 Visualize

This section focuses on open-source components that clearly summarize the analyzed events. One of the most important attributes is the license of the compared components, which should provide the freedom of possible commercial closed-source use. The component should also provide an appealing user interface for querying data. Dashboards are essential for providing a clear overview of threats. Dashboard reporting is needed to comply with regulatory standards. A versatile alerting functionality should notify key employees of ongoing threats.

### 2.5.1 OpenSearch Dashboards

OpenSearch Dashboards is a Kibana fork and the primary visualization tool for OpenSearch data. Data can be viewed using dashboards, which include various visualizations such as bars, heatmaps, maps, timelines, data tables, and pie charts. Reports can be created from dashboards using the Reporting panel. Data can be searched using the Dashboard Query Language (DQL), Lucene Query string query language in the Discover and Dashboard search bar, and SQL, Piper Processing Language PPL in the Query Workbench. Malicious activity can be detected by triggering alerts based on rules. Security Analytics is a security information and event management (SIEM) solution for OpenSearch that includes features such as creating detectors based on detection rules and creating alerts with various notification channels. Detection rules are used to identify potential threats based on ingested log data. Users can be notified of malicious activity, for example, through Team, Slack, email, or custom webhooks. (48).

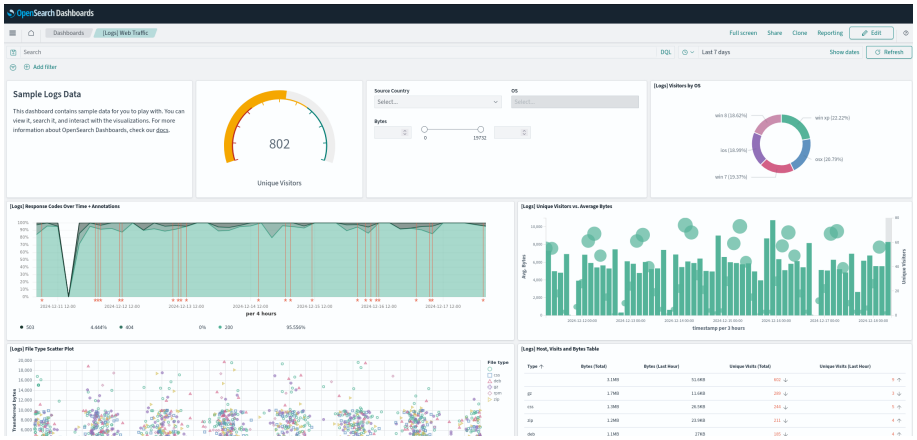


**Figure 2.3:** Opensearch Dashboards predefined Dashboard panel with sample data

### 2.5.2 Grafana OSS

Grafana is a Kibana fork and the primary visualization tool for the Grafana Stack (Loki). Grafana was released under the Apache 2.0 license until version 8.0; newer versions are licensed under the AGPL. Data can be viewed by creating dashboards consisting of multiple panels, the basic visualization blocks used in Grafana. Dashboard reporting is not available in the community edition. Grafana Alerting is a module that can detect malicious activity in logs. Triggered alerts can be declared incidents with the Grafana Incident module. Key employees can be notified of alerts and incidents using the Grafana Alertmanager (52).


**Figure 2.4:** Grafana dashboards using Test Data log source

### 2.5.3 Kibana

Kibana is the primary Visualization tool for the Elastic Stack released under the Apache 2.0 license until version 7.10.2; newer versions are dually licensed under Elastic License and Server Side Public License SSPL. Data can be viewed using dashboards similar to OpenSearch Dashboards. Reports can be created from dashboards or visualizations in paid tiers. Configurable retention is also not available in the free tier. (53).

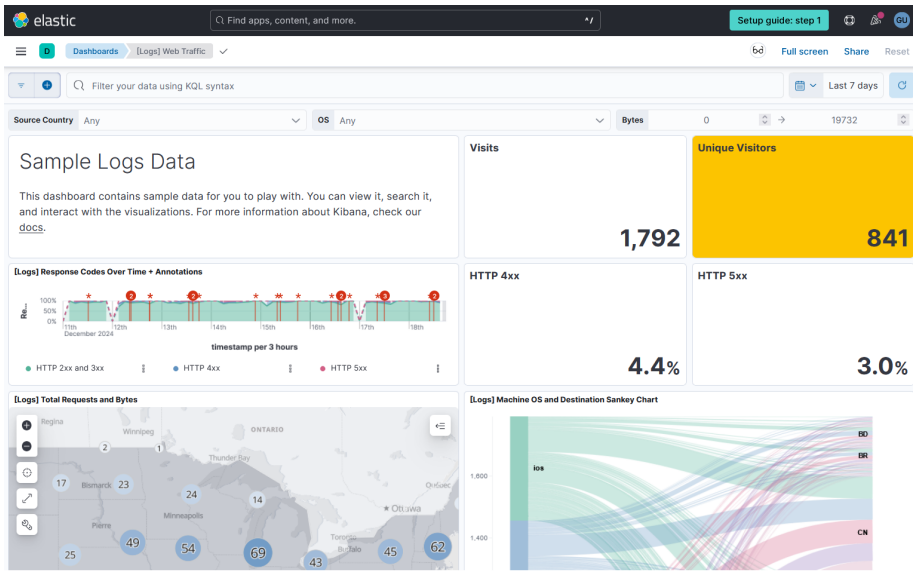


**Figure 2.5:** Kibana predefined Dashboard panel with sample data in a demo environment (54)

### 2.5.4 Graylog Open

Graylog Open is a centralized Log Management System (LMS) that offers basic functionality under the SSPL license. Data can be viewed using dashboards that include multiple widgets. Dashboard reporting is not available in the open tier. Alerting notification is restricted to email and HTTP posts, and the colleration engine is missing from the open tier. In the background, Graylod uses MongoDB to store metadata. Graylog requires a data node such as ElasticSearch or OpenSearch to store logs. (55).

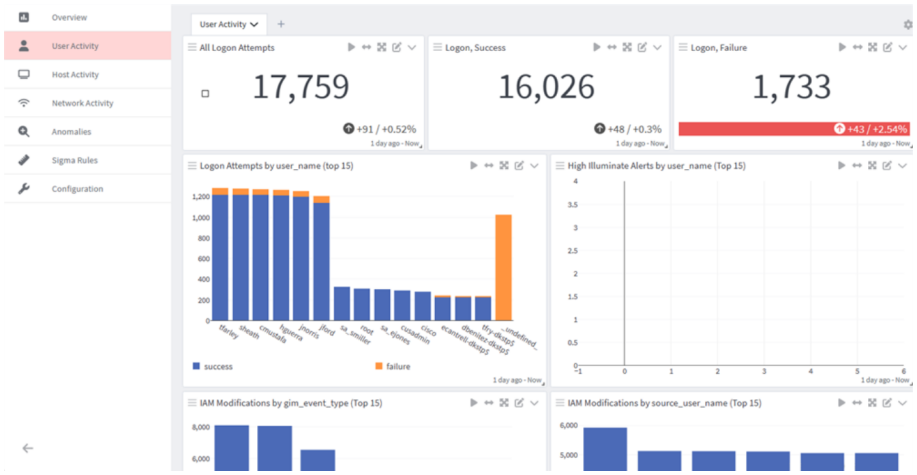


**Figure 2.6:** Example dashboard from demo Graylog Security module (56)

### 2.5.5 Results

Table 2.5 shows the functionalities provided by the analyzed components. As can be seen, Graylog Open provides the least functionalities. OpenSearch Dashboards provide all the required functionalities without additional cost and provide the most license freedom with the Apache 2.0 license.

Grafana (Figure 2.4) provides the most appealing overview, followed by Kibana (Figure 2.5) and OS Dashboards (Figure 2.3), each of them based on Kibana; Graylog Open (Figure 2.6) provides the least vissualy appealing overview.

**Table 2.5:** Visualize components attribute matrix

| Attribute | Kibana | OS Dashboards | Grafana | Graylog Open |
|---|---|---|---|---|
| License | Elastic/SSPL | Apache 2.0 | AGPL | SSPL |
| Alerting | ✓ | ✓ | ✓ | Restricted |
| Reporting | paid | ✓ | paid | paid |
| Retention | paid | ✓ | ✓ | paid |

# 3 Implementation and testing

In this chapter, a POC implementation will be deployed based on selected components. The selected components from the previous chapters are Fluentbit/Fluentd and Winlogbeat OSS and Logstash OSS for the collection of logs. For the storing/indexing of logs OpenSearch was selected. For the visualize component OpenSearch dashboards will be used.

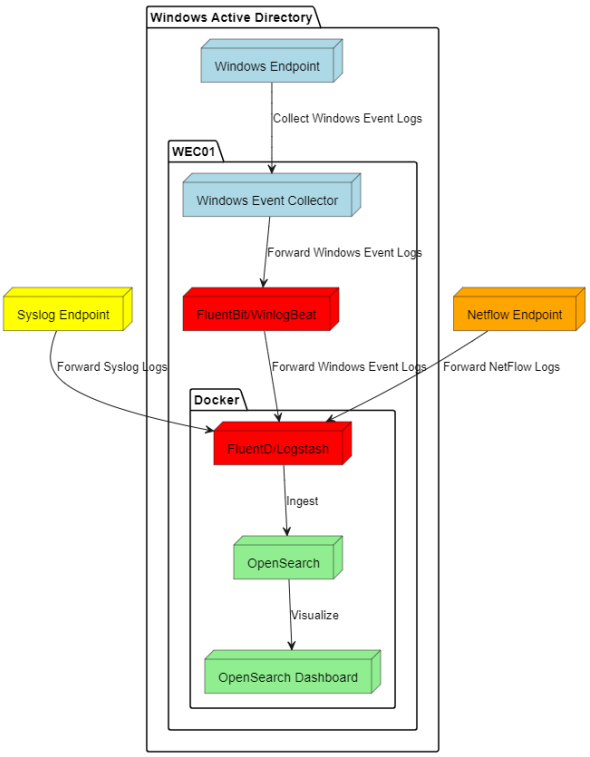


**Figure 3.1:** Implementation diagram

## 3.1 Collect

Fluentbit, Fluentd, Beats OSS, and Logstash OSS were chosen based on their input and output modules. The test environment will consist of three data sources: Windows Event Log, Syslog, and NetFlow, which will be ingested using the tested component or its lightweight version.

**Windows Event Logs** To test ingestion, handling, and forwarding of Windows Event Log events, a simple Active Directory with a Domain Controller and a Windows Event Collector server will be created in a virtual environment using VMWare. The Windows Event Log collection will be set up using Microsoft's recommendations for a Baseline and a Targeted subscription (5). The tested attributes of each component will include reading different channels and parsing Windows event logs.

**fluentd** Fluentd uses the windows_eventlog2 module to read Windows Event Logs. No issues were identified during the testing of reading Windows Event Logs from different channels. A useful feature provided by windows_eventlog2 is the parsing of the message field of logs, which is enabled by the parse_description option. This feature extracts new fields from the message field, for example, in case of an account log-of fields like subject.security_id, subject.account_name, subject.account_domain and subject.logon_id would be created (31). An incorrectly parsed event was found from the forwarded events channel during the testing. Example of DNS Client Events with an Event ID of 8020, which fluentd parsed seemingly incorrectly.

```
"___________adapter_name_:_
{96111592-c6c7-4e55-a698-9b30c47c6f37}\r\n
___________host_name_:_win01\r\n
___________primary_domain_suffix_:_test.local\r\n
___________dns_server_list_:\r\n
____________":"192.168.196.2\r\n
Sent update to server : 192.168.196.134:53\r\n
IP Address(es) :\r\n
192.168.196.134"
```

**fluentbit** Fluent-bit can read Windows Event Logs using winlog and winevtlog. During the testing of the winlog module, an unusual behavior was discovered. After configuring fluent-bit to read the "ForwardedEvents" channel and output the results to stdout, the number of Windows Event Logs was lower than expected. After further analysis, it was discovered that fluent-bit falsely flags the read Windows Event Logs as from the "ForwardedEvents" channel, although they belong

to the "Application" channel. For example, event generated at 2024-11-02 14:16:03 with EventId of 16389 and event generated at 2024-11-02 14:16:53 with EventId of 16384 are flagged as from the ForwardedEvents channel 3.2. However, they originated from the Application channel 3.3. As the issue seems only to occur when reading from the "ForwardedEvents" channel, this could be potentially bypassed by deploying fluent-bit on all Windows endpoints and forwarding the events to the central collector. Module winevtlog does not contain the mentioned unusual behavior and works as expected. Unfortunately, winevtlog does not provide an option to parse the message field for Windows Event Logs. The parsing could be achieved by using complex regexes or parsing using Lua scripts (57).


**Figure 3.2:** Falsely identified Windows Event Log channel by fluent-bit

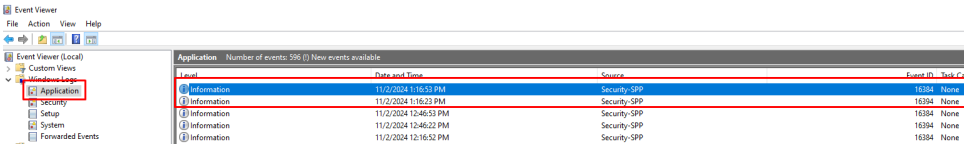


**Figure 3.3:** Correct Windows Event Log channel of events

**beats-oss** Winlogbeat from the beats-oss family is responsible for collecting the Windows Event Logs. During the testing of reading data from different Windows Event Log channels, no difficulties were discovered. Winlogbeat also provides a built-in parser for extracting data from the message field of Windows Event Logs (58).

**Syslog** For the testing of ingestion, handling and forwarding of Syslog a simple test environment is created with a rsyslog forwarded.

Rsyslog was chosen as endpoint collector and forwarder mainly because of the high availability among Linux distributions. Neither of tested collectors handle the parsing of sysmon syslog data correctly. A simple ruby filter plugin for fluentd was created to parse the information from the sysmon message in XML format.

**NetFlow** Ingestion, handling and forwarding of NetFlow data will be tested using SolarWinds NetFlow Generator [1]. Fluentbit and Beats-OSS do not provide an input module for Netflows. Logstash provides a deprecated netflow input. Based on these difficulties and the previous selection of Fluentd, Fluentd is used in the implementation to minimize collection solutions.

## 3.2 Store and Visualize

### OpenSearch

After the successful collection of logs, they need to be stored and indexed to be viewed and analyzed. OpenSearch will be deployed with a test configuration using docker compose (59). To prepare these prerequisites needed to ingest, alert, and visualize the data a python script was created which communicates with the OpenSearch or the OpenSearch Dashboards API.

### 3.2.1 Index management

In OpenSearch, each index contains the definition of every field of a document within the index. OpenSearch can dynamically create field definitions when it identifies incoming data with a never-before-seen index. Issues arise when the data can contain multiple data types in one field. This issue can be prevented by explicitly stating the field definitions of the index, after which OpenSearch will parse the data into a predefined type. If an index ingests too much data, it may become slow. To address this issue, each collector has a set index naming convention for where to send longs. For example, the Windows Event

1. `https://documentation.solarwinds.com/en/success_center/ets/content/toolsetphnetflowgenerator.htm`

collector endpoint will create a new index in the pattern of akasa_wef-%year.%month.%day each day. Because logs from each day will be stored in new indexes, we need to define an index template. After defining an index template with the field definitions and the pattern name, each newly created index will follow the field definitions of the template. In OpenSearch, there are multiple ways to view various indexes at once. Index patterns group indexes by a common prefix. Indexes akasa_wef akasa_netfow and akasa_syslog would all be matched by the index pattern akasa. After defining an index pattern, a timestamp field is needed to be selected. Index patterns can be, for example, in the Discovery module of OpenSearch to view log data from each index matched by the pattern. Another way of referencing multiple indexes is using aliases. Each alias can have a write and multiple read indexes. By referencing the alias of an index, the modules behave as if the action was set for each index in the alias. Aliases can also be predefined by index templates (60).

**Log retention** Log retention in OpenSearch can be achieved using policies. Each policy can contain multiple states, defining actions taken when the index enters the state and transition conditions when to move to the next state. States defined by the Python initializer are with their actions, and transitions can be seen in fig.3.4. Index state management is used to apply policies to newly generated indexes. After applying the policy, the index becomes a managed index (61). Table 3.1 describes the estimated sizes of collected log data for the retention period based on quick measurements. Measurements were conducted using the Netflow generator mentioned earlier in the case of Netflows and in the case of Syslog and Windows Event logs. Attack simulations based on Cymulate Endpoint Security were used with custom-made scenarios to generate events. The low estimate of syslog data size may be due to generated events being short messages that are easily compressible or measurement errors

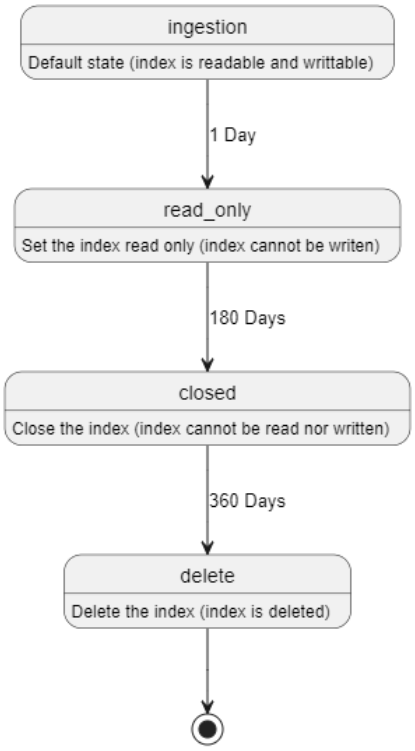


**Figure 3.4:** Index policy defined by python initializer

**Table 3.1:** Storage size estimate for different logs in OpenSearch

| Time | NetFlow (100/s) | Syslog (1/server) | WEF (1/endpoint) |
|---|---|---|---|
| Events 1H | 318k | 11k | 10k |
| 1 H | 28.8mb | 456.2kb | 27.2mb |
| 24 H | 691.2mb | 10,9mb | 652.8mb |
| 30 D | 20.7gb | 328.3mb | 19.6gb |
| 180 D | 124.2gb | 2gb | 117.5gb |
| 360 D | 248.4gb | 4gb | 235gb |

### 3.2.2 Dashboards

Dashboards in OpenSearch are based on visualizations. Each visualization needs to belong to an index. But dashboards can contain visualizations of multiple indexes. The creation of visualizations and dashboards is not supported using the API. To achieve automatization of visualizations and dashboards, we need to create the visualizations manually and upload them as OpenSearch saved objects which contain information about the indexes and so on (62).

**NetFlow Dashboard** Each created dashboard includes a control panel at the top of the dashboard. For example on the NetFlow dashboard, analyst can choose source and destination addresses, ports and used protocol.

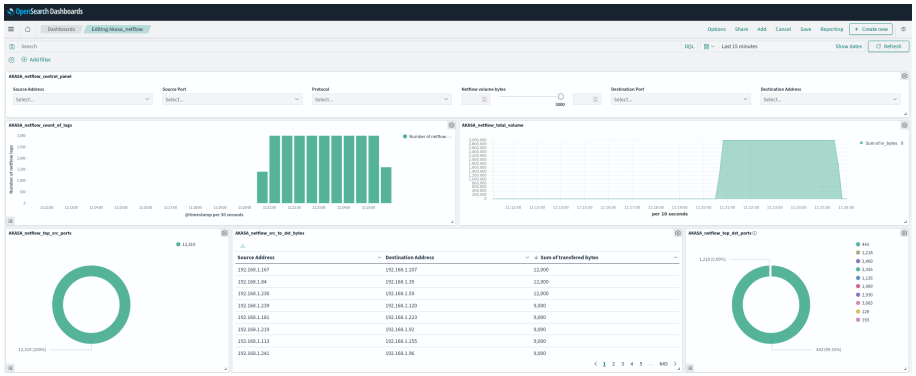

**Figure 3.5:** Dashboard created to provide a basic overview of NetFlows in the infrastructure

**Windows Event Dashboard** Contains a control panel to filter events by endpoint, event id, user, image and dns query. The dashboard contains a quick overview of overall and unusual dns requests. It also contains an overview of executed images and failed logins.

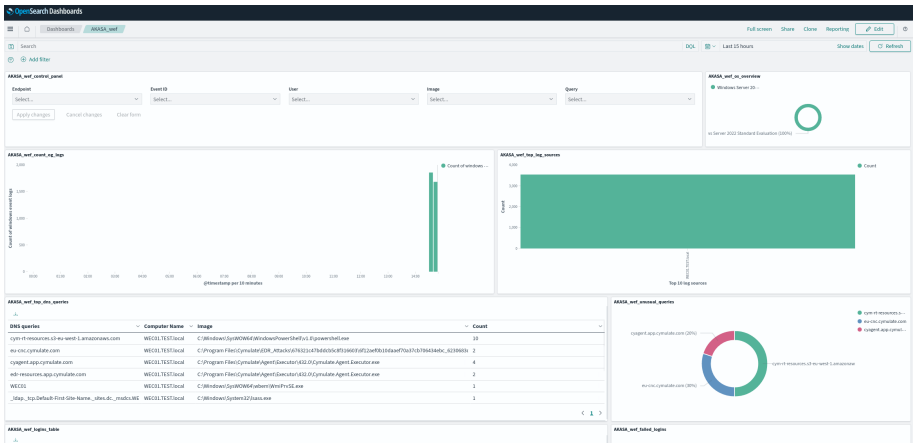

**Figure 3.6:** Dashboard created to provide a basic overview of Windows Events in the infrastructure

**Syslog Dashboard** The syslog dashboard contains executed images, syslog messages, and a control panel to filter hosts, ident and image.

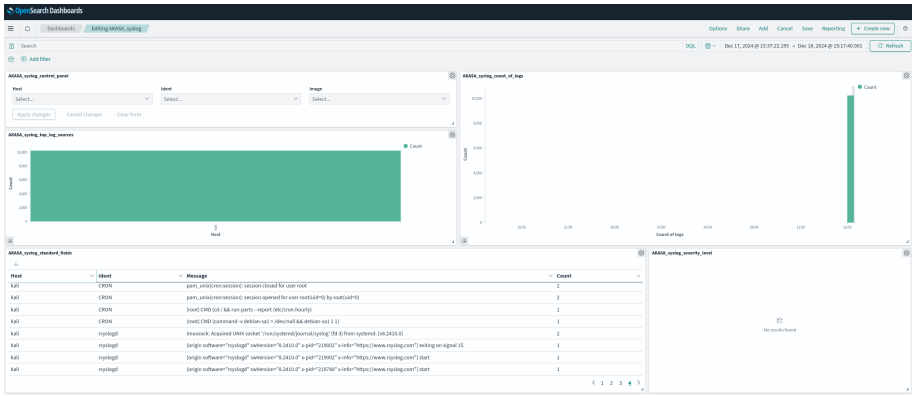

**Figure 3.7:** Dashboard created to provide a basic overview of Syslog events in the infrastructure

### 3.2.3 Security Analytics

**Detection and alerting** OpenSearch includes the Security Analytics plugin, which detects malicious behavior based on sigma rules. First, a log type needs to be created. After that, detection rules can be defined based on sigma rules. Each field of an index needs to be mapped to the field of a detection rule, either by exact match or by using field aliases. The main issue with Security Analytics is the lack of clear documentation.[2] on how to ingest data and how to map fields to be easily detectable by sigma rules (63). During the creation of detectors using the Security Analytics module, multiple issues arose, among others. The detection mechanism does not always trigger, even for most basic Sigma rules, such as checking one field in the index. Also, at the moment, OpenSearch contains event correlations in some way, but the only available colleration is colleration based on security findings and not events. This significantly reduces the detection logic applicable.

2. https://forum.opensearch.org/t/using-security-analytics-is-hard-mostly-because-of-a-la
16271/40

**Notifications** OpenSearch supports multiple notifications such as Slack, Chime, Microsoft Teams, custom webhooks, email, and Amazon SNS. For testing purposes, the docker environment contains a Mailhog [3] server that collects emails sent to it based on alerts. The alerting language is mustache[4] and supports some information injections to the message. However, the missing feature that is most wanted is to attach a direct link to the finding in the alerting message.

3. https://github.com/mailhog/MailHog
4. https://mustache.github.io/

# 4 Conclusion

The main aim of this thesis was to compare different open-source solutions for small businesses building log management systems.

The rise of cyber attacks creates a hostile environment for these entities, which makes them unable to defend themselves due to a lack of resources or personnel. Usually, these companies cannot afford a dedicated person to maintain the required solutions and must rely on third-party companies. The idea behind this project is to provide a starting point for these companies to help them meet requirements for various laws and mandates and elevate their cyber resilience for minimal cost.

Due to time restrictions, the end solution of the project was only partially achieved at the time of writing this thesis and will be continuously developed and evaluated.

The main part I identify as necessary for future work is mapping the required information to collect data. This would reduce the storage requirements for the deployment and provide a clear overview of where to find the required information. Beyond that, a main point of improvement would be thorough testing of the collected data for various cyber attacks to create detection rules and patterns. Last but not least, deployment in a real environment to get feedback from analysts and threat hunters and improve the project.

# Bibliography


1. *M-Trends 2024 Special Report* [online]. Mandiant [visited on 2024-11-10]. Available from: https://services.google.com/fh/files/misc/m-trends-2024.pdf.
2. *About Us* [online]. IstroSec [visited on 2024-11-10]. Available from: https://istrosec.com/#about.
3. ANTON A. CHUVAKIN, Kevin J. Schmidt; PHILLIPS, Christopher. *Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management*. Waltham (Mass): Syngress, 2012. ISBN 978-1-59749-635-3.
4. KENT, Karen; SOUPPAYA, Murugiah. *Special Publication 800-92: Guide to Computer Security Log Management: Recommendations of the National Institute of Standards and Technology*. Gaithersburg, MD: National Institute of Standards and Technology, 2006.
5. *Use Windows Event Forwarding to help with intrusion detection* [online]. Microsoft [visited on 2024-09-22]. Available from: https://learn.microsoft.com/en-us/windows/security/operating-system-security/device-management/use-windows-event-forwarding-to-assist-in-intrusion-detection.
6. *Zákon č 69/2018 Z. z., Zákon o kybernetickej bezpečnosti a o zmene a doplnení niektorých zákonov - § 20: Bezpečnostné opatrenia* [online]. National Council of the Slovak Republic [visited on 2024-11-02]. Available from: https://static.slov-lex.sk/pdf/SK/ZZ/2018/69/ZZ_2018_69_20220630.pdf.
7. *Zákon č 95/2019 Z. z., Zákon o informačných technológiách vo verejnej správe a o zmene a doplnení niektorých zákonov* [online]. National Council of the Slovak Republic [visited on 2024-11-15]. Available from: https://static.slov-lex.sk/pdf/SK/ZZ/2019/95/ZZ_2019_95_20200701.pdf.
8. CZECH REPUBLIC, Parliament of the. *Zákon č. 181/2014 Sb., Zákon o kybernetické bezpečnosti a o změně souvisejících zákonů (zákon o kybernetické bezpečnosti) - Bezpečnostní opatření: § 5* [online]. [visited on 2024-11-02]. Available from: https://www.zakonyprolidi.cz/cs/2014-181/zneni-20220806.

9. *DIRECTIVE (EU) 2022/2555 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive)* [online]. European Union [visited on 2024-11-02]. Available from: `https://eur-lex.europa.eu/eli/dir/2022/2555/oj`.

10. *REGULATION (EU) 2022/2554 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 14 December 2022 on digital operational resilience for the financial sector and amending Regulations (EC) No 1060/2009, (EU) No 648/2012, (EU) No 600/2014, (EU) No 909/2014 and (EU) 2016/1011* [online]. European Union [visited on 2024-11-02]. Available from: `https://eur-lex.europa.eu/eli/reg/2022/2554/oj`.

11. *PCI DSS Quick Reference Guide: Understanding the Payment Card Industry Data Security Standard version 3.2.1*. PCI Security Standards Council.

12. *Information security, cybersecurity and privacy protection: Information security management systems — Requirements: Annex A*. Geneva, Switzerland: International Organization for Standardization, 2022.

13. *Information security, cybersecurity and privacy protection: Information security controls*. Geneva, Switzerland: Internation Organization for Standardization, 2022.

14. *ATT@CK* [online]. The MITRE Corporation [visited on 2024-11-02]. Available from: `https://attack.mitre.org/`.

15. *The Cyber Kill Chain* [online]. Lockheed Martin [visited on 2024-11-02]. Available from: `https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html`.

16. *About Sigma* [online]. SigmaHQ [visited on 2024-11-10]. Available from: `https://sigmahq.io/docs/guide/about.html#the-format`.

17. *SIEM Reporting: Definition and How to Manage It* [online]. SentinelOne [visited on 2024-11-10]. Available from: `https://www.sentinelone.com/cybersecurity-101/data-and-ai/siem-reporting/`.

18. *What Is Log Rotation?* [online]. Arfan Sharif [visited on 2024-11-10]. Available from: https://www.crowdstrike.com/en-us/cybersecurity-101/next-gen-siem/log-rotation/.

19. *Log Retention 101 - What is it and Best Practices* [online]. Eromosele Akhigbe [visited on 2024-11-10]. Available from: https://signoz.io/guides/log-retention/.

20. POLS, Paul. *MODELING FANCY BEAR CYBER ATTACKS*. 2017. MA thesis. Cyber Security Academy.

21. ALLEYNE, Nik. *Learning by practicing: Hack Detect*. n3Security Inc., 2018. ISBN 978-1-7753830-0-0.

22. HADNAGY, Christopher; EKMAN, Paul. *Unmasking the Social Enginee: The Human Element of Security*. John Wiley Sons, Inc., 2014. ISBN 978-1-118-60857-9.

23. DIOGENES, Yuri; OZKAY, Erdal. *Cybersecurity - Attack and Defense Strategies*. Packt Publishing Ltd., 2018. ISBN 978-1-83882-779-3.

24. *Living Off The Land Binaries, Scripts and Libraries* [online]. LOLBAS-project [visited on 2024-11-10]. Available from: https://lolbas-project.github.io/#.

25. *Living Off Trusted Sites (LOTS) Project* [online]. mrd0x [visited on 2024-11-10]. Available from: https://lots-project.com/.

26. *Lateral Movement* [online]. HackTricks [visited on 2024-11-10]. Available from: https://book.hacktricks.xyz/windows-hardening/lateral-movement.

27. *Event Logging (Event Logging)* [online]. Microsoft [visited on 2024-09-22]. Available from: https://learn.microsoft.com/en-us/windows/win32/eventlog/event-logging.

28. *The Syslog Protocol* [online]. R. Gerhards [visited on 2024-11-10]. Available from: https://www.rfc-editor.org/rfc/rfc5424#section-1.

29. *The Syslog Protocol* [online]. B. Claise, Ed. [visited on 2024-11-10]. Available from: https://www.ietf.org/rfc/rfc3954.txt.

30. *Netflow plugin for Fluentd* [online]. repeadetly [visited on 2024-04-20]. Available from: https://github.com/repeatedly/fluent-plugin-netflow.

31. *fluent-plugin-windows-eventlog* [online]. Fluent [visited on 2024-11-02]. Available from: https://github.com/fluent/fluent-plugin-windows-eventlog.

32. *Fluentd client* [online]. Grafana Labs [visited on 2024-04-20]. Available from: https://grafana.com/docs/loki/latest/send-data/fluentd/.

33. *FluentD to SigNoz* [online]. SigNoz [visited on 2024-04-20]. Available from: https://signoz.io/docs/userguide/fluentd_to_signoz/.

34. *fluentd Documentation* [online]. Treasure Data [visited on 2024-03-01]. Available from: https://docs.fluentd.org/.

35. *Fluent Bit v3.0 Documentation* [online]. Treasure Data [visited on 2024-03-01]. Available from: https://docs.fluentbit.io/manual.

36. *Rsyslog Documentation* [online]. Rainer Gerhards (Großrinderfeld), et al. [visited on 2024-03-01]. Available from: https://www.rsyslog.com/doc/index.html.

37. *Download Logstash - OSS only* [online]. Elastic [visited on 2024-03-01]. Available from: https://www.elastic.co/downloads/logstash-oss.

38. *Logstash Netflow Module* [online]. Elastic [visited on 2024-03-01]. Available from: https://www.elastic.co/guide/en/logstash/current/netflow-module.html.

39. *Download Logstash - OSS only* [online]. Logstash Output OpenSearch [visited on 2024-03-01]. Available from: https://github.com/opensearch-project/logstash-output-opensearch.

40. *Logstash plugin* [online]. Grafana Labs [visited on 2024-03-01]. Available from: https://grafana.com/docs/loki/latest/send-data/logstash/.

41. *Logstash Introduction* [online]. Elastic [visited on 2024-03-01]. Available from: https://www.elastic.co/guide/en/logstash/current/introduction.html.

42. *Beats: Lightweight data shippers* [online]. Elastic [visited on 2024-03-01]. Available from: https://www.elastic.co/beats.

43. *Promtail agent* [online]. GrafanaLabs [visited on 2024-03-01]. Available from: https://grafana.com/docs/loki/latest/send-data/promtail/configuration/.

44. *Grafana Agent* [online]. GrafanaLabs [visited on 2024-03-01]. Available from: https://grafana.com/docs/agent/latest/?pg=oss-agent&plcmt=hero-btn-2.

45. *syslog-ng Open Source Edition 3.26 - Administration Guide* [online]. One Identity [visited on 2024-03-01]. Available from: https://www.syslog-ng.com/technical-documents/doc/syslog-ng-open-source-edition/3.26/administration-guide.

46. *NXLog Community Edition* [online]. NXLog [visited on 2024-03-01]. Available from: https://nxlog.co/products/nxlog-community-edition.

47. *Apache Solr Reference Guide* [online]. Apache Software Foundation [visited on 2024-04-01]. Available from: https://solr.apache.org/guide/solr/latest/index.html.

48. *OpenSearch and OpenSearch Dashboards* [online]. OpenSearch [visited on 2024-03-01]. Available from: https://opensearch.org/docs/latest/about/.

49. *Elasticsearch Guide* [online]. Elastic [visited on 2024-03-01]. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html.

50. *Grafana Loki documentation* [online]. GrafanaLabs [visited on 2024-03-01]. Available from: https://grafana.com/docs/loki/latest/?pg=oss-loki&plcmt=quick-links.

51. *Introduction* [online]. SigNoz, Inc. [visited on 2024-04-01]. Available from: https://signoz.io/docs/.

52. *Grafana documentation* [online]. Grafana Labs [visited on 2024-04-01]. Available from: https://grafana.com/docs/grafana/latest/.

53. *Kibana—your window into Elastic* [online]. Elastic [visited on 2024-04-01]. Available from: https://www.elastic.co/guide/en/kibana/current/introduction.html.

54. *Observability Overview*. Elastic. Available also from: https://www.elastic.co/demo-gallery/observability-overview.

55. *Graylog Documentation* [online]. Graylog, Inc. [visited on 2024-04-01]. Available from: https://go2docs.graylog.org/5-2/home.htm.

56. *Graylog Security*. Graylog Inc. Available also from: https://go2docs.graylog.org/current/what_more_can_graylog_do_for_me/graylog_security.html?.

57. *Lua* [online]. Treasure Data [visited on 2024-11-02]. Available from: https://docs.fluentbit.io/manual/dev-3.0/pipeline/filters/lua.

58. *Winlogbeat fields* [online]. Elastic [visited on 2024-11-02]. Available from: https://www.elastic.co/guide/en/beats/winlogbeat/current/exported-fields-winlog.html.

59. *Docker*. OpenSearch. Available also from: https://opensearch.org/docs/latest/install-and-configure/install-opensearch/docker/.

60. *Index Management*. OpenSearch. Available also from: https://opensearch.org/docs/latest/dashboards/im-dashboards/index/.

61. *Index State Management*. OpenSearch. Available also from: https://opensearch.org/docs/latest/im-plugin/ism/index/.

62. *Creating dashboards*. OpenSearch. Available also from: https://opensearch.org/docs/latest/dashboards/dashboard/index/.

63. *About Security Analytics*. OpenSearch. Available also from: https://opensearch.org/docs/latest/security-analytics/.

# A An appendix

**akasa.zip** contains the docker compose enviroment with OpenSearch, OpenSearch dahboards and fluentd. The project is intended to be deployed only on testing environments. The feeder folder contains the data relate to the initialization of objects. Folders such as rules/, dashboards/, index_templates/ and ism_templates cotain definitions of objects created using the python initializer script.