UNIVERSITY OF TARTU
Institute of Computer Science
Cybersecurity Curriculum

Mark Borissov

Deploying Open-Source SIEM system for Waldur-based services at the University of Tartu

Master's Thesis (30 ECTS)

Supervisor(s): Risto Vaarandi,
Ilja Livenson.

Tartu 2024

Deploying Open-Source SIEM system for Waldur-based services at the University of Tartu

Abstract:

Security Information and Event Management (SIEM) systems are cybersecurity tools that are used by organizations to monitor and analyze log information from different sources, allowing the detection and response to security threats in a timely manner. Waldur is an open-source platform used to manage hybrid cloud resources with multiple services built on top of it and with a large user base. To address the security requirements of the platform in regard to business event data, this thesis work aims to identify and implement a suitable open-source SIEM solution for Waldur that aligns with the operational requirements provided by the University of Tartu HPC team. An overview of Waldur's architecture, business event logging and SIEM requirements has been conducted. OpenSearch, an open-source data management platform with SIEM functionality, was selected for a proof of concept implementation. A high-level design of the architecture and components of the implementation as well as sample security rules based on Waldur's logs and requirements were developed. Validation using synthetic and real data was performed in the proof of concept implementation, providing insight into OpenSearch's SIEM capabilities, with challenges being encountered for complex detection scenarios.

This study demonstrates the development of a SIEM architecture for Waldur platform's business event logging and validates it with a selected SIEM solution, providing insights into the setup, usage, and potential limitations.

Keywords:

Waldur, SIEM, OpenSearch

CERCS: T120

Avatud lähtekoodiga turvasündmuste halduse tarkvara (SIEM) kasutamine Walduri platvormil põhinevate teenuste jaoks Tartu Ülikoolis

Lühikokkuvõte:

Infoturbe ja sündmuste haldamise (SIEM) süsteemid on küberturbe tööriistad, mida organisatsioonid kasutavad erinevatest allikatest pärineva logiteabe jälgimiseks ja analüüsimiseks. See võimaldab õigeaegselt tuvastada ja reageerida turvaohtudele.

Waldur on avatud lähtekoodiga hübriidpilve ressursside haldamise platvorm, mis on aluseks mitmetele teistele teenustele ning millel on märkimisväärne kasutajabaas. Antud lõputöö eesmärgiks on tuvastada ja rakendada Walduri jaoks sobiva avatud lähtekoodiga SIEM-lahenduse, mis tagab äri-andmete turvalisust ja vastab Tartu Ülikooli HPC meeskonna poolt kehtestatud operatiivnõuetele. Selleks sai läbi viidud Walduri arhitektuuri ja sündmuste logi ning SIEM-nõuete analüüs. Kontseptsiooni tõestamiseks valiti avatud lähtekoodiga OpenSearch andmehaldusplatvorm, millel on SIEM-funktsionaalsus. Vastavalt Walduri nõutele ja logidele loodi arhitektuur, arendati rakendamiseks vajalikud komponendid ning näidisturvareeglid. Valideerimisprotsessi käigus kasutatud tehis- ja reaalandmed võimaldasid hinnata OpenSearch-i SIEM-võimekust ning see tõi esile, et keerukate ohutuvastusstsenaariumite puhul tekkisid valitud lahendusel raskused.

Koostatud uurimistöö eesmärgiks on Walduri platvormi andmete logimise jaoks SIEM-arhitektuuri väljatöötamine ja valideerimine valitud SIEM-lahendusega. Lisaks sellele annab tehtud töö ülevaate valitud tarkvara seadistamisest, kasutamisest ja võimalikest piirangutest.

Võtmesõnad:

Waldur, SIEM, OpenSearch

CERCS: T120

Table of Contents

Introduction

Security Information and Event Management (SIEM) systems are commonly used to provide a management approach for an organization's security infrastructure. Its principle of operation focuses on aggregation of important and relevant data from different sources, detecting any irregularities or deviations and acting on them [1].

In the current digital landscape, service providers face an increasing number of cyber threats that can include phishing, malware, distributed denial of service (DDOS) attacks, ransomware, and others. The use of SIEM offers organizations a centralized platform to gather event and log data, with a set of predefined rules to generate security alerts if such are detected [1]. Organizations implementing SIEM solutions benefit from this real time analysis and event generation as it allows responding to security incidents in a timely manner.

Waldur is an open-source platform, co-developed by University of Tartu for managing hybrid cloud resources [2]. It is used in various services, one example being the ETAIS portal, which is a self-service portal used for managing computational resources of ETAIS consortium members [3]. It is used to run different external services and supports users from various institutions. As it is used by different organizations there is a lot of personal and sensitive data involved in the operation of the portal. To meet security compliance standards and data protection requirements, monitoring and logging user activities and resource usage is essential. While logs are being shown to users and system logs are used for network protection, currently there is a lack of proactive deviation analysis.

The implementation of a SIEM system will improve the safeguarding of sensitive user data and provide increased and effective security measures for the platform. In the event of a security incident, the SIEM system implements response mechanisms and alerting for effective mitigation. Different services or software often have unique characteristics and usage patterns. A SIEM system would need to be tailored to the specific requirements of Waldur-based platforms. An analysis on the specific requirements and needs would be conducted before addressing the implementation. For the purpose of this thesis, we will be taking a look at common SIEM requirements and considerations for IT infrastructure, an overview of Waldur's current standing and services, and the potential requirements. We will aim to produce a proof-of-concept SIEM implementation, to test the validity of the SIEM concept for Waldur and the applicability of integrating such software with the current Waldur infrastructure.

1 Abbreviations

SIEM – Security Information and Event Management.

IT – Information Technology.

API – Application Programming Interface.

UT – University of Tartu.

HPC – High Performance Computing.

PI – Principal Investigator.

ETAIS – Estonian Scientific Computing Infrastructure.

JSON – JavaScript Object Notation.

ELK – Elastic, Logstash, Kibana.

OWASP – Open Worldwide Application Security Project.

2 Background

Waldur is widely used within the UT's digital infrastructure, for services like ETAIS self-service [4] and Puhuri portals [5]. In the current day, cyber threats are a growing concern, as such securing our digital assets is paramount to maintaining the integrity and reliability of essential services. The absence of a dedicated SIEM system within the context of multi organizational use of Waldur, makes it susceptible to security vulnerabilities. This research is driven by the need to address, driven by the need to improve security measures and implement protection against potential threats.

Foremost among these challenges is the absence of an integrated open-source SIEM solution aligned with Waldur's operational constraints. Waldur is used for cross organizational applications that generate many different business process events and logs. User activity is high, users using the applications, allocating resources, creating virtual machines, and more. This necessitates careful evaluation and implementation of a suitable SIEM system to manage, analyze and act on these logs and suspicious activity.

We need to know what to look for and what we currently have. An overview of the logs generated by Waldur is required. We will need to gather knowledge that is available on the topic of SIEM implementation and operation. We will examine the role of SIEM systems in modern security frameworks, the value they offer in terms of security event monitoring and incident response. For proper implementation and integration, we will identify the necessary requirements for SIEM solutions, and the various event correlation and aggregation methods that are required for applications that generate large event log volumes. An understanding of how Waldur is typically utilized will also provide a foundational understanding which may be crucial for designing a robust alerting system. Additionally, the development of a Proof-of-Concept integration will allow us to explore the feasibility and suitability of SIEM for Waldur before going for a full-scale implementation.

2.1 Waldur Overview

Waldur is an open-source cloud infrastructure management platform. Waldur can be used for both managing the already existing cloud resources of organizations and for the setup of new cloud services. Waldur provides a management platform for overseeing and controlling the cloud services. Functionality that is offered involves resource management, access control, cloud monitoring, payment services, storage, analytics, and others.

Waldur also acts as a cloud service brokerage platform, offering a wide range of cloud services for different customers. This allows users to deploy and manage cloud resources with Waldur for various needs.

A brief overview of the offered functionality provided by Waldur [6]:

- Integrated marketplace for publishing cloud services.
- Self-service portal that allows the users to access and manage their cloud environment. Allows for deployment of virtual machines, configuring resources, managing user permissions, billing, monitoring cloud usage and more.
- Reporting system which provides customers with system activities and other events within their cloud environments.
- Accounting that supports third-party billing systems integration. Optimizes cloud service spending and budgeting for organizations.
- Customer support ticketing system with integration options for third-party support helpdesks. Can be configured to offer a flexible support line for all managed resources assisting in resolution of issues or inquiries that may arise.

- Open source, which allows users to customize and adjust the available functionality to meet their needs or work on new features. Options for developer support are also present.

We will take a look at the user groups and roles within Waldur platform. These roles play an important role in access control, as they dictate who can access or modify sensitive data within the platform. Knowing the user roles allows us to better design and implement appropriate security measures and access controls.

The main user types along with their permissions within Waldur are listed in a table within the documentation [7, "Users, Organizations and Projects"]. This is presented in Table 1:

Table 1. User roles within platform.

Role	**Web and API access**	**Create support requests**	**Provide user support**	**See all projects and resources**	**Manage Organizations**	**Access admin area**
Operator Roles						
Staff	Yes	Yes	Yes	Yes	Yes	Yes
End-user roles						
User	Yes	Yes	No	No	No	No
Support agent	Yes	Yes	Yes	Yes	No	No

From the table above, we can see that roles available for end users can access basic functionalities as well as user support and project visibility, with staff members, being the more sensitive role with the highest level of access, can manage organizations, access administrative areas, and perform platform-wide actions.

In addition, specific user roles within different organization types are presented (Table 2):

Table 2. Roles within different organization types in Waldur.

Permissions	**Cloud (roles)**	**Academic (roles)**	**Academic Shared (roles)**
Manage Team	Owner, Project Manager	PI, co-PI	Resource allocator, PI
Manage Projects	Owner	PI	Resource Allocator

Request and Manage Resources	Owner, Project Manager, Administrator	PI, co-PI, Member	Resource Allocator, PI, co-PI
Resource Request Creation Approval	Owner, Project Manager, Administrator	PI, co-PI, Member	Resource Allocator, PI, co-PI
Resource Request Approval	Owner, Service Manager	PI, Service Manager	Resource Allocator, Service Manager
Manage Offerings (provider specific)	Owner, Service Manager	PI, Service Manager	Resource Allocator, Service Manager

Waldur utilizes role-based access control mechanisms [7, "Role-based access control"] to manage the user permissions within the platform. Users should only have access to functionalities or features intended for their role.

2.1.1 Waldur-based Projects in University of Tartu

Several services that are built using Waldur software are in use and operated by the University of Tartu. First one to note is Puhuri [5]. It is a cloud service for service providers such as HPC (High-Performance Computing) centers. These centers make use of powerful systems to address complex high-scale calculations or simulations. Widely used in scientific organizations for education and research. The objective of Puhuri is to offer easy and available access to HPC systems for organizations across Europe. The target audience consists of both resource allocators and service providers. The services offered have been utilized by dozens of organizations across Nordic and Central European regions. Countries involved with the development and use of Puhuri are Estonia, Finland, Denmark, Iceland, Sweden, and Norway.

Estonian Scientific Computing Infrastructure (ETAIS) is an important part of Estonia's research infrastructure. It is providing computational, cloud and storage resources to the country's scientific community. This is achieved by combining all major Estonian HPC providers into a single service. The ETAIS infrastructure includes scientific computation centers, the clusters, and data repositories [8]. A self-service portal has been implemented into ETAIS, allowing easy access to service providers and potential customers. Using this portal, access is granted to organizations and users regardless of their location. This allows for more effective collaboration between institutions and research groups by sharing the research infrastructure without the need for negotiations of the rate quotas, payments, and resource access [8, pg. 10].

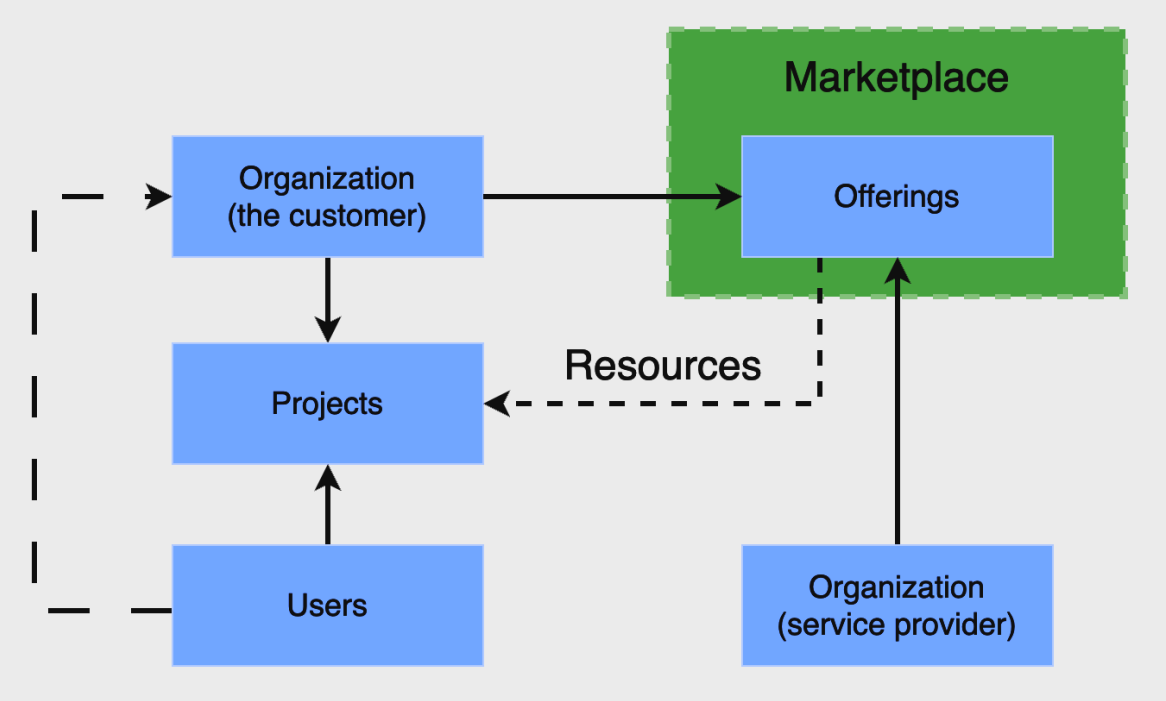

Figure 1. Main components of the Waldur system and their relationships.

The main customers are organizations which can be companies, research groups or educational institutions. The shared responsibility model applies here, with the organizations being responsible for the actions of their users. Projects are entities that are created within organizations to allocate resources for the users to then access and utilize. Providers are organizations that offer specific services such as HPC, virtual machines, cloud, and storage services.

Within Estonia ETAIS's resources are divided among the University of Tartu, Tallinn University of Technology and National Institute of Chemical Physics and Biophysics, with all these institutions offering their hardware resources for the platform.

2.1.2 Overview of Waldur Event Logging and Generated Events

Proper logging of security events plays a crucial role for event correlation as well as SIEM implementation. Events logged by the Waldur platform will be the main basis for the security threat detection and potential response and remediation activities. We will take a look at Waldur's event logging approach, noting the types of events captured, their log format, and the relevance to security monitoring.

To start off, Waldur's documentation provides a comprehensive list of events [7, "Events"]. These events are separated into categories, ranging from authentication and authorization to project and support. These events offer us information about different aspects of system and user activity within the platform. These events provide invaluable insights into the platform's operation and user interactions.

For example, for the Authentication category "auth_logged_in_with_username" event type is listed which is emitted on successful user login into their account with username and password. Another example is the "user_creation_succeeded" event in the User category, which is triggered in the case of a new user account being created within the platform.

The user creation event, for example, is an important aspect of identity management. Tracking user creation events is valuable for access and permissions management in addition to monitoring for potentially unauthorized or malicious account creation activities. For login activities, monitoring these events helps security systems to ensure the security of user accounts by detecting potentially suspicious login activities. For example, a potential rule configuration may be set up to detect successful login attempts inconsistent with past activity based on time zone. As in, successful login attempts that are being recorded during non-activity periods such as at night may be considered as potentially malicious and flagged. This will of course depend on the security assessment and policies in place at an organization. A copy of the list of events from Waldur's documentation has been provided in [9].

The picture below offers a code snippet of the logging mechanism implemented within Waldur's source code (Figure 2).

```
token = self.refresh_token(user)
user.last_login = timezone.now()
user.save(update_fields=["last_login"])

logger.debug("Returning token for successful login of user %s", user)

event_logger.auth.info(
    "User {user_username} with full name {user_full_name} "
    "authenticated successfully with username and password.",
    event_type="auth_logged_in_with_username",
    event_context={"user": user, "request": request},
)
```

Figure 2. Sample code of event logging within Waldur. Logs successful user login activity.

In the code above, we can see several activities take place upon a successful login by the platform's user. We can see that the user object's (a user object is a collection of data that contains information on the user) field (fields store specific data within an object) "last_login" receives an update, with the current time of the login being written into it. After this information is added, the user is then saved to make sure this information persists. This is valuable for security purposes, as it can be used for tracking user activity within the system and provide detailed history in the event of an attack investigation.

Proceeding after this, two log activities are triggered and logged by the application, with "logger.debug" and "event_logger.auth.info". It should be noted that there are at least two different types of logging here. The "logger.debug" is related to operational logging, which is used to capture technical system-level information, internal errors, and application behavior and are stored differently. The logs made by "event_logger" logger however can be considered to be more related to the business side of operations. These provide more cohesive information regarding the various user interactions, user activities, client resource utilization or other business-related events and are accessible to end users, filtered by their corresponding permissions.

For the purpose of our proof-of-concept test implementation of a SIEM, we will be primarily targeting the logs generated by the business side of our application. These logs capture most of the events that are related to the various use activities, data access and resource utilization.

These logs are being directed to and stored within a PostgreSQL database [10]. The logs are stored within the "logging_event" table. An example of a log table entry is as follows:

```
SELECT *
FROM logging_event
WHERE event_type = 'auth_logged_in_with_username'
LIMIT 1;

8 a86794e5-1e41-48f6-8e59-c3e1063a71c9  2024-01-19 01:10:50.294509+00
auth_logged_in_with_username User admin with full name  authenticated successfully
with username and password. {"os": {"name": "Macintosh"}, "bot": false, "flavor":
{"name": "MacOS", "version": "X 10.15.7"}, "browser": {"name": "Chrome", "version":
"120.0.0.0"}, "location": "", "platform": {"name": "Mac OS", "version": "X
10.15.7"}, "user_uuid": "a2a8646bd5964a53a7223c6e227bbee9", "ip_address":
"192.168.65.1", "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36",
"user_is_staff": "True", "user_username": "admin", "user_full_name": "",
"user_is_support": "False", "user_native_name": "", "user_token_lifetime": "3600"}
```

Figure 3. Sample of a successful login event logged to PostgreSQL database.

From the above (Figure 3), we can see that the main information comes in a JSON formatted object (a data format to store and transport data between applications) as well as observe multiple valuable pieces of information that are being collected. A more detailed description of the fields of the events, and their corresponding information is provided in Table 3 below:

Table 3. Sample JSON object event log fields for user login events.

Field	Description	Value
ID	A unique identifier of the log entry.	a86794e5-1e41-48f6-8e59-c3e1063a71c9
Timestamp	Time and date when the event took place.	2024-01-19 01:10:50.294509+00
Event Type	Type of event	auth_logged_in_with_username
Message	Additional information and context for the event.	User admin with full name authenticated successfully with username and password.
OS	Details about the operating system that was used.	{"os": {"name": "Macintosh"}
Bot	Whether the user is a bot.	"bot": false
Flavor	Additional details about the OS version.	“flavor": {"name": "MacOS", "version": "X 10.15.7"}

Browser	Information about the web browser used.	"browser": {"name": "Chrome", "version": "120.0.0.0"}
Location	Geo location of the user.	-
Platform	Details of the platform.	"platform": {"name": "Mac OS", "version": "X 10.15.7"}
User UUID	Unique identifier of the user.	a2a8646bd5964a53a7223c6e227bbee9
IP Address	IP address used by the user.	"ip_address": "192.168.65.1"
User Agent	Information about the user agent.	"user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36"
User Is Staff	Whether or not the user is a staff account.	"user_is_staff": "True"
User Username	Username of the user account.	"user_username": "admin"
User Is Support	Whether or not the user is a support account.	"user_is_support": "False"
User Native Name	Native name of the user (if any).	"user_native_name": ""
User Token Lifetime	How long the user token is valid for.	"user_token_lifetime": "3600"

All of these attributes can be used in comprehensive understanding of the login event and potentially implemented into the SIEM's rulesets for security monitoring and anomaly detection.

2.1.3 Waldur API

Conveniently, Waldur offers a flexible API [11] endpoint that allows for querying for the generated events with a comprehensible documentation being available for the events endpoint. The endpoint, "api/events" returns batches of the most recent events, along with multiple headers such as total event count, as well as pagination links. Additional filtering

options are available using parameters, such as “created_from”, “created_to”, “event_type”. Other endpoints offer specific data or information such as event counts, event groups, accepted event types and other options. This allows us to filter the incoming events by date or event type.

3 Related Work

The design and implementation of a Security Information and Event Management (SIEM) system can be critical for companies who seek to enhance their cybersecurity measures in order to be up to date with today's emerging threats. In this chapter, we will look at the existing literature on the topics of SIEM implementation, cloud security considerations, and event log monitoring approaches. The intent is to provide a strong foundation for the theoretical analysis and a proof-of-concept implementation.

3.1 Methodology

The literature review was conducted through a systematic approach, by gathering and analyzing relevant works connected to the implementation of Security Information and Event Management (SIEM) systems in various scenarios. This was done by conducting a broad search across different academic databases and repositories such as Google Scholar, IEEE Xplore, CORDIS as well as industry-related articles. Predefined keywords and search terms included "Open-Source SIEM", "Security Information and Event Management," "SIEM implementation", "Cloud service security", "Log management and event monitoring", "cloud security", "cloud security challenges", and others. The selection process prioritized articles and papers with details to the security environment and requirements for the described services or systems. To ensure the inclusion of modern-day best practices and solutions the results were filtered to include more recent papers, written within the last decade if possible. The methodology for the literature review was guided by established frameworks for systematic literature reviews, including the formulation of research questions, definition of inclusion and exclusion criteria, and the synthesis of findings to identify key themes and trends. Through this approach, the literature review aimed to provide an overview of the existing knowledge and practices of SIEM implementation, considering insights and recommendations which could be relevant for a SIEM solution for Waldur-based services at the University of Tartu.

3.2 The Role of Security Information and Event Management (SIEM) Solutions

Events are often defined as information-containing messages or alarms regarding specific activities on a network or information system [12, pg. 2]. Derived from events are alerts, which are events that contain information about unusual activity. Both events and alerts play a crucial role in security monitoring systems, where they are collected and analyzed in order to detect and act upon malicious activity.

Security Information and Event Management (SIEM) solutions play an important role in diverse organizational contexts. Specific user personas can be viewed to be using and implementing SIEM solutions for different needs. User personas ranging from network administrators to auditors face various challenges, including real-time threat management, network oversight, and compliance with regulations [1, pg. 48]. Several scenarios described by the authors show that security administrators, auditors and stakeholders recognize the importance of log data generation and analysis in the event of a security incident [1, pg. 48-50]. Suggested solutions involve the implementation of log managers, event managers and other tools with similar functionalities to SIEM solutions. This addresses the unique requirements of organizations. It is also effective in incident reporting and systems monitoring, enhancing the investigation of security incidents significantly.

The examples provided by IBM showcase various scenarios and imply the complex nature of SIEMs catering to various organizational needs [1]. Apart from log aggregation and

monitoring, compliance and adherence to security controls were brought to attention. Particularly in scenarios requiring organizations to demonstrate compliance with external audits and industry regulations. Furthermore, a chapter highlights the significance of a SIEM solution that encompasses aspects such as security information management reporting, event management reporting, monitoring, log management and forensic capabilities. We can see the versatility of SIEMs in addressing security challenges from various organizational needs.

SIEM systems play an important role in organizing the collection and analysis of data from multiple sources. These can be external services, API requests, physical sensors, firewalls, databases, and servers. These systems play a role in decision making and enabling a timely response to, or prevention of malicious activity. Despite their importance, common SIEM solutions face constraints. One such constraint is the limited ability to contextualize events spanning multiple organizations and event collection environments. The work "Security and Reliability Requirements for Advanced Security Event Management" [2] goes over some scenarios, needs and limitations of SIEMs within such conditions.

The study focuses on several scenarios that require more advanced and robust, scalable SIEM solutions. Scenarios have examples from large-scale sporting events, payment systems and water dam control systems. The application of SIEM tools in securing the huge IT framework of the Olympic Games shows the challenges in managing an event of scale, with a multitude of sports, athletes, services, servers, and network devices [2, pg. 172]. Similarly, the scenario involving mobile money transfer services presents a complex landscape. Challenges include the optimization of detection performance, false positives and maintaining scalability and security. The growing adoption of managed services by businesses, either through outsourcing or cloud-based models, signifies an extra layer of complexity. Additionally, dams, classified as critical infrastructures, demand real-time monitoring to ensure their efficiency in water supply, power generation, and other crucial functions. It is noted that current SIEM systems are primarily focused on digital and information systems [2, pg. 174]. Applications combining the monitoring and managing of both critical infrastructure and digital systems with multiple sources can be challenging to develop.

The work has included a suggested list of guidelines for different application scenarios and usage requirements. Notable ones include enhanced correlation across security event layers, multi-level security event modeling and malicious behavior monitoring. The needs for securing evidence integrity and ensuring trustworthiness are also noted. Aspects like real-time processing, scalability, handling diverse data sources, and various rulesets may be relevant for our aim on SIEM implementation for Waldur.

The research paper titled "Why SIEM is Irreplaceable in a Secure IT Environment?" [13] by O. Podzins and A. Romanovs argues that a properly implemented SIEM solution may be the best security solution for a company to invest into. It talks about how any security solutions such as firewalls or IDS (Intrusion detection system) may detect malicious activity in different components of the infrastructure but are often not capable of understanding the "context" of what may be happening [13, pg. 1]. Activities and events can be consolidated from different endpoints with the information used to make security decisions. Contextual analysis may be used to detect anomalous activity from different systems, allowing for potential specific measures like security team notification or for example IP-address blocking. A good list of advantages of using a SIEM system is also provided [13, pg. 4]. These include but are not limited to finding system misconfigurations, identifying bot activities, anomaly detection, data theft detection and post-incident evidence preservation. Several disadvantages are also stated, some notable ones are the difficulty of setup and implementation

of a SIEM, the associated costs of the implementation and the need for continued updates and maintenance. The final conclusions of the paper state that organizations interested in effective security against cyberthreats may find SIEM solutions to be “irreplaceable”.

SIEMs also have an important role to play in critical infrastructure services, such as financial, energy and healthcare sectors. It has been stated [14, pg. 20] that in some cases these sectors may possess unique challenges in terms of cybersecurity. Points are made that many of these systems have been designed long ago with operational stability in mind, with little security considerations. With modern day emerging cyber threats, these sectors, being critical infrastructure, require careful and thorough security measures. Constant real-time monitoring solutions are required. Emphasis is also made on the specific needs and requirements of different sectors.

The same paper also offers arguments on the potential future of SIEM solutions, raising the question of the importance of investing into said systems [14, pg. 12]. Various economic, political, societal, legal, technological, and even environmental factors are presented and taken under consideration. Political factors include a serious increase of cyber security investments from legal and government entities, with a notable one being the EU Commission’s signing agreement with the industry with 1.8 billion euro of investments back in 2016 [15]. The trends in the employment market show an increase of cybersecurity positions and also strengthen the notion that more robust and flexible security solutions will be of greater need in the industry [14, pg. 13]. Technological factors include cloud storage and cloud service integration, which offer larger processing power and allow for more scalable systems. A note on machine learning technologies is made, suggesting that once improved such technologies can be used by SIEMs for better and more efficient analytics and decision making [16] [14, pg. 13]. Environmental factors affected the transformation of the data handling practices conducted by various businesses, with more cloud environments being utilized. The data generation increase will affect the design of SIEM systems, to accommodate these changes. In conclusion, the authors have argued that the conditions for long term SIEM system investment are favorable. It is likely that such systems will see increased integration and adoption by not only major entities but also small and medium sized businesses.

Additionally, an important feature of SIEM software lies in report generation, which may play an important role from an organization’s regulation and compliance point of view. Some SIEM tools allow automatic generation of security reports with all important logged security events, aiding organizations with regulatory requirements and possible audits [17]. Such information may provide valuable insight into the organization’s current security situation, potentially allowing us to identify current strengths and weaknesses.

3.3 SIEM Implementation and Considerations

Here, we will look at potential criteria examples to consider when implementing or selecting a SIEM within the context of a specific organization or needs. The work “SIEM Selection Criteria for an efficient contextual security “[18] gives us an overview of some potential criteria for SIEM selection. Choosing the right SIEM solution among the many available on the market poses a significant challenge. Two main categories were proposed by the authors: functional criteria and technical criteria [18, pg. 3]. Functional criteria assess whether the SIEM tool performs its intended functions such as data correlation, alert management, and reporting. For example, a table is shown, showcasing the different log formats and their specifics [18, pg. 3]. Technical criteria delve into aspects like the vendor's reputation, documentation, integration, and product support. The authors have selected three SIEM tools for testing. The tools were OSSIM, ELK and LogPoint. Integration tests with the SIEMs were conducted. The strengths and weaknesses of the SIEMs were highlighted. This work

is interesting as it proposes potential selection criteria for SIEM solutions for different organizations. Important strengths to look at are mentioned such as automation, incident management, correlation, and multi-source data support. Weaknesses noted are partial risk coverage of traditional SIEMs, difficulties in handling large data, and false alerts that are created due to poor analysis.

An example of the use of a SIEM for a cloud-based project is described in the paper "Toward the SIEM Architecture for Cloud-based Security Services" [19]. Within the context of the described project (a cloud-based security platform), a SIEM system was proposed to handle the logs generated by security services running in virtual machines. The SIEM's architecture was adjusted to handle the significant volume of security event logs generated by these virtualized systems. The SIEM Engine allows for intelligent threat analytics and data output due to their focus on data processing and mining [19, pg. 1]. The SIEM User Layer supports incident response activities from various sources for the platform users. The integrated SIEM was implemented with a focus on time and correlation analytics. The obtained data is extracted, and afterwards correlation processes are run to identify similarities or anomalies among security events, improving the security of their cloud operations.

A Gartner report from 2020 titled "Critical Capabilities for Security Information and Event Management" provides considerations when evaluating SIEM solutions [20]. First and foremost, one must consider the scope and potential use cases of the deployment. Potentially relevant stakeholders and users may be noted. For initial stages the most critical steps and functionality should be focused on, with more complex use cases to be developed upon later [20, pg. 2]. This kind of approach helps organizations better optimize and manage the available resources and deployment costs. Organizations that are heavily integrated with cloud services are suggested to look at having SIEM as a service, for reduced costs and setup and maintenance complexity. Additional factors to consider during evaluation would be the scale and complexity of deployment, data collection, log forwarding and compliance requirements.

Gartner proceeds to provide tables and graphs with scoring for available on the market SIEM products and their vendors. Scores are presented for different monitoring use cases such as basic security, basic searching and reporting, complex security monitoring and others [20, pg. 5]. Notable high-ranking products that can be seen are Exabeam SIEM, Splunk, Securonix and IBM QRadar SIEM. While the scoring graphs are for commercial non-open-source products from 2020, they offer valuable insights into the comparative performance of some of those systems.

The report also talks about the growing IT landscape and use for SIEM systems. With growing use so do increase the requirements of the users. To address these requirements SIEM solutions are constantly improving and developing various functionalities. These include log management, event analytics and reporting [20, pg. 20]. Some examples are the means to handle large amounts of data input and analysis with tools like Elasticsearch. Additionally integrating new analytics methods such as machine learning for threat detection usage. Furthermore, the addition of Security Orchestration, Automation, and Response (SOAR) functionality into SIEM solutions shows the importance of automation and orchestration security response capabilities. These factors can be taken into consideration for organizations exploring the idea of SIEM implementation.

A few core functions of SIEM solutions are listed [20, pg. 21], these include multi source event collection, real time analysis and alerting, visual dashboards, generating workflows. Some of the typical intended uses for SIEM include monitoring user activities, machine, and resource access.

The report also mentions nine critical capabilities of SIEM technologies which may help potential users to select a solution for their specific use case [20, pg. 22]. These include:

- Architecture/Deployment/Scalability – the SIEM solutions should be compatible with different environments and architectures. It is suggested that SIEM tool buyers should at least to an extent evaluate the potential complexities of the deployment for their specific environment. Supporting integration with various other external security tools and platforms should also be considered.
- Cloud Readiness – with increased use of cloud platforms and services, buyers may consider whether they will require a solution which will support cloud integration.
- Operations and Support – SIEMs are complex integrations, which will require dedicated maintenance and support. Rule language, management consoles, dashboards and the user-experience may be considered.
- Data Management Capabilities – SIEM tools should be able to support a variety of different data sources. Be these plaintext log files, formatted logs, network packets or others. To be able to manage and access the right data and integrate with the data sources.
- Analytics – to analyze data in real-time and react upon it SIEM solutions should include different analytics methods. This will assure the most accurate results.
- Incident Management and Response – potential for analytics and collaboration should be considered, especially for more complex cases that require multiple users or analysts. Incident or case management would be a plus, with support for notes, comments, or annotations as well as status change. Everything that will improve response actions and minimize redundancy.
- Content Packaging and Management - SIEM solutions rely on different content, including collectors, analyzers, use cases, configurations, and response capabilities. Effective content management structures, including centralized repositories and integration of external services, make content easier to access, deploy, and modify. Notably for first time deployments and use, as the ease of use and integration may be valued over advanced functionality.
- Forensics and Threat Hunting – it may be needed for SIEM tools to support investigation and digital forensics functionality. Flexible search is a fundamental feature of a system built around data processing and analytics. This can also include user-friendly interface, flexible search patterns and quick loading times.
- User Experience and User Interface – SIEM tools can be complex to use and manage. Tools with intuitive and understandable user interfaces and good user experience for users of different roles and skill sets is a good bonus. In some cases, a single tool may be utilized by dozens or hundreds of analysts or operators. In such cases, it is reasonable to expect for the tool to accommodate the needs of flexible customization, for example for dashboards or alerting. Other features can include task delegation and communication means.

Finally, examples of potential use cases are provided for new SIEM users to look for when going for a particular tool [20, pg. 26]. “Basic Searching and Reporting” focuses on fundamental queries and occasional reporting. It is suitable for smaller organizations seeking basic log analysis capabilities, or those who are only starting out and have not yet fully allocated resources for more complex scenarios. More advanced use cases such as “Compliance and Control Monitoring” are more towards organizations that are required to demonstrate fulfillment of legal obligations and industry standards. “Basic Security Monitoring” supports threat detection and deployment. It is ideal for first-time SIEM buyers and those with minimal security needs. “Complex Security Monitoring” is for organizations with

specific and complex architectures and data challenges. Such organizations need easily scalable solutions for their environments which are able to cope with large data log volumes. Such organizations will likely set up advanced security measures, with data collection, investigation and incident response measures prepared. “Advanced Threat Detection and Response” focuses on early threat discovery and immediate response to targeted attacks. Advanced threat hunting activities are to be part of the SIEM solution in this case. This use-case is particularly relevant for high-risk sectors with advanced adversaries. These sample use cases provide a basic understanding for us for what to consider in our needs for a SIEM tool. For all the above-mentioned use-cases the prior mentioned critical capabilities list still applies. Therefore, the tool must offer good user experience and visual interfaces, flexible and customizable search capabilities, and supported integration with third-party security services to effectively address different security threats and requirements.

A white paper published by SANS Institute [21] delves into the evaluation of SIEM performance and capabilities via benchmarking. The paper discusses how enterprises can gather and measure the performance of their SIEM systems, allowing them to assess their system. For event log normalization, the term EPS (events per second) is used for example. This metric refers to the number of events that the SIEM can process in a single second timeframe. A high ratio allows for larger data volumes to be processed, which may be crucial to avoid slowdowns or bottlenecks. In some extreme cases such as firewalls, the number of connections per second may be tens of thousands, with the potential of all of them generating related logs or events.

Another important metric is Mean Time to Remediate (MTTR), which measures the average time it takes for the SIEM to respond to an anomaly or an incident after initial detection. Lower MTTR numbers means that it takes less time for the SIEM to respond to detected threats. This is valuable in terms of reducing security risks and incident impact, as the SIEM will be able to quickly handle and resolve the incidents [21, pg. 2].

Different factors can go into the requirements for the metrics. Scenarios including peak loads or potential DDoS (Denial of Service) may be considered when setting the bars for the expected metrics. Variables such as Peak Events (PE) per second and Normal Events (NE) per second are provided to help determine the expected normal and peak usages. Requirements for PEs and NEs as well as formulas for calculating EPS (num. of events/ time in seconds) are provided [21, pg. 4].

The question of data storage and analysis is also brought up within the paper. While the EPS rating should be considered as a guideline for SIEM integration planning, one must also consider the data output and storage in the case of an incident. The depth of analysis and post-incident actions and updates may depend on the amount of data that was gathered during an incident. An example provided [21, pg. 8] is that in the case of a 20000 EPS system, an 8-hour long incident will possess over 500 million data records, which may end up taking more than 150 gigabytes of disk space. Factors like available storage, data retention regulations and event processing capabilities should be considered to take full advantage of the SIEMs ability to handle incidents.

The above benchmarking metrics and criteria have been expanded upon in the paper “SIEM Selection Criteria for an efficient contextual security” [18]. The authors begin by examining different log formats that are available and used in software and equipment. Log formats dictate how the data is structured within the log files, they can include specific fields, attributes, event types and other identifiers. The formats vary, some may be standardized and utilized across different systems providing a consistent standard, while others may be exclusive or proprietary to a specific software or platform. Different log formats may introduce

additional difficulties in SIEM's data processing, and as such the authors suggest the logs to be standardized if possible [18, pg. 2]. A small table is provided within the paper, describing several available log formats that may be encountered in different software implementations [18, pg. 13]. The log formats listed include Syslog, Syslog-NG, CLF (Common Log Format, often used for web requests), WELF (Firewalls), Proprietary and IDMEF (Intrusion Detection Message Exchange Format, used for incident reports in some Intrusion Detection Software). Knowing the log formats generated by the systems is important for the SIEM software to properly process the logged data.

Further on, the authors in [18] introduce their suggested SIEM selection criteria. The two major categories that are described are [18, pg. 3] functional selection criteria and technical selection criteria. The Functional criteria mainly include and focus on whether the SIEM tool possesses the required features and capabilities that are desired by the organization. These can be integration with other software, log and event correlation functionality, management of the alerts to be sent via email or other communication tools. These criteria describe how well a tool or solution will align with the intended goals of the company. The technical criteria are the ones that relate to the SIEMs technology, performance, integration, and support. Example given on how poor documentation may induce errors and difficulty in utilizing or troubleshooting the tool. These can include [18, pg. 4]:

- Vendor – the main provider of the SIEM solution. Things to consider when looking at the vendor would be their history of the product, experience in the field, the implemented levels of technical support. The pricing plans and quality-to-price ratio. Scalability and reliance on third party vendors or services may also be looked at.
- Integration level of the solution – these will affect how seamlessly the solution can be integrated into the current organizational workflow. Log compatibility means the tool can ingest and process logs from different sources and devices. Wide range of log formats is preferred. Integration with the organizations already existing tools such as email alerting or chat notifications for event alerting should be looked at as well.
- The ease of deployment – an important consideration especially for organizations with smaller security teams or more limited operational budgets. SIEM is multi-context, with different variables to consider. Pre-defined configuration files, multi system support and good documentation will help organizations to quickly deploy and begin monitoring their systems.
- Product evolution – SIEM solutions are being constantly developed upon, adding new up-to-date features and technologies. Regular and constant changes and updates to improve the functionality and quality of the tool are good signs of the developers supporting and maintaining their product. New cyber threats are constantly emerging, meaning for organizations to remain prepared their tools must evolve and adapt to new security approaches. Version history, update history, perhaps a plan or roadmap are good to be looked at for the SIEM tool.

In order to reinforce and evaluate some of the above points, the authors have decided to apply their SIEM selection criteria to three random SIEM solutions that were available on the market [18, pg. 4]. The three SIEMs that have been added and configured to their IT environment were OSSIM AlienVault, ELK (Elastic Search, Logstash, Kibana) and Logpoint. An immediate issue with AlienVault that was brought to light and mentioned was the lack of up-to-date documentation for integration AlienVault with CheckPoint firewall. The offered descriptions of the integration process did not work. This resulted in a delay of approximately one week, thus highlighting the earlier point made by the authors for the documentation's importance. Another issue occurred with the normalization/parsing for an

OSSIM screen. The explanation offered considered two options, one being the occasional change in log structure by the vendors during an update or that the plugins were designed with specific tools and versions. This highlights the “product evolution” technical criteria from before, suggesting the vendor’s slow response to customer needs. The issue can be resolved manually, albeit time consuming.

A more detailed summary of the solutions was provided by the end of the paper [18, pg. 5]. Some notes are that for example AlienVault at the time had rather poor documentation with difficult integration albeit with frequent updates. ELK has easy deployment and a good source of information from the documentation and an active community. LogPoint having good support and smooth integration with deployment, however being proprietary means a lack of complete knowledge of on-going processes. In conclusion, the presented criteria along with examples of evaluation provide valuable insight on the approach for how organizations can get a summary of the selected SIEM tool.

4 Design of SIEM Implementation for Waldur

In this chapter we will provide a design overview of the SIEM solution for the Waldur platform giving us greater context and understanding of our existing architecture, operational constraints, and requirements as well as provide details for the proof of concept architecture and test environments.

4.1 High Level Design

In this section we will cover the main architecture components and the expected interactions between these components that will be in the final proof of concept implementation.

4.1.1 Architecture components

The primary components that will be involved in our SIEM system are: Waldur as a Log Source, a Log Collector which is responsible for collecting events from Waldur. Data Enrichment component for enriching them with valuable contextual information when necessary. The SIEM system, which processes the collected data and offers different security capabilities, such as detection rule creation, monitoring, alert generation and notification generation as well as a visualization tool to provide a comprehensible user interface and visualization of the events.

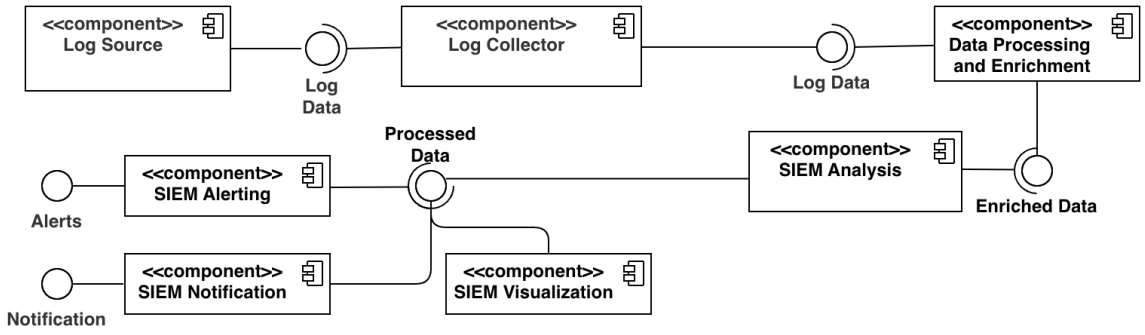

Figure 4. Diagram illustrates the high-level interactions between the components.

More information regarding the components can be found in the following table (Table 4):

Table 4. Description of main components

Component	Description
Log Source	The system or device that generates and stores logs for the SIEM to collect data from.
Log Collector	Responsible for gathering the logs from the source and also forwarding them onwards.
Data Processing and Enrichment	Used for parsing and enriching the collected log data with additional information or context if needed.

SIEM Analysis	The SIEM analytics engine that performs the analysis on the incoming log data for threat identification based on rules that can be defined within the SIEM.
SIEM Alerting	Generates an alert when the SIEM detects an anomaly based on defined SIEM rules.
SIEM Notification	Forwards notifications of the alerts and other security threats detected by the SIEM to the security team, ensuring timely communication of generated alerts.
SIEM Visualization	Provides a graphical user interface to read and interact with the processed data.

4.1.2 Data Processing and Low Flow

Below is a flow chart illustrating the data flow process of the SIEM system.

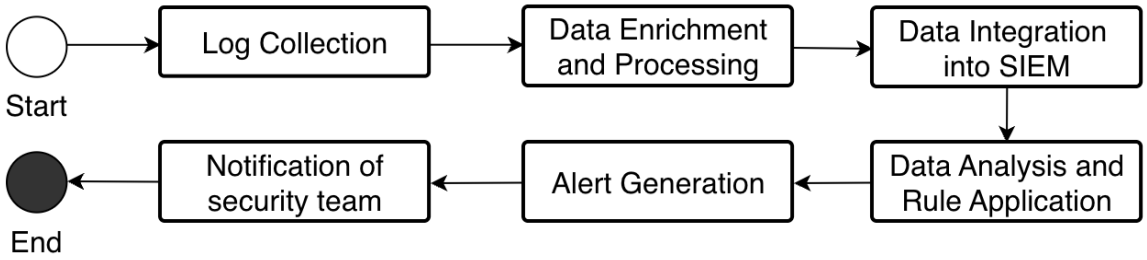

Figure 5. SIEM data flow chart.

The data flow begins with the collection of logs from the log source. Afterwards during the enrichment phase, the data may be modified, with additional fields or information being added for more details or context which may be necessary for our security rules. Next, the enriched data is sent into our SIEM, where it is stored and used for further analysis. The SIEM then processes the data according to the rules, algorithms and functions, detecting irregularities or potential threats, generating alerts in such cases. In the end, notifications are created for the security team to inform them of these findings so they may take necessary actions.

4.1.3 Waldur SIEM Requirements

In section 3.3 we have looked at a number of different papers that explore criteria and considerations for SIEM solutions. A comprehensive paper offering an overview and comparison of several SIEM solutions [22] has been also consulted, being a recent publication and covering several popular choices currently available on the market. The paper included an analysis of the different features. Some of these features were mentioned as being important or essential, and should be present within a SIEM tool [22, pg. 15]. Based on our research and consultations with the Waldur team, we have prepared a list of requirements that would be beneficial for a potential SIEM tool to have, these include:

- **Open-source (licensing)** - it is preferable for a potential SIEM solution to be licensed as open-source. This model offers advantages in terms of costs, being lower or absent compared to more commercial solutions. Greater customizability and control over the SIEM tool, as there will be no licensed or unknown code or dependencies within. The code can be inspected for security purposes or even adjusted if the necessity arises. Independence from vendors or security providers is a bonus, as it reduces the need to switch providers in cases such as pricing plans or licensing changes. Non-viral licensing can prove advantageous, especially in the event Waldur developers decide to integrate SIEM as part of the Waldur product.
- **Local Data Storage** - the security events and logs can potentially contain private and sensitive information. We must consider potential data regulations and data storage compliance that may be required. SIEM solutions that are cloud-based or require the analyzed events and data to be sent off-site must be avoided to maintain control over the information.
- **Graphical User Interface** - intuitive and flexible user interface is a good for the security team members to be able to effectively monitor the on-going and incoming alerts.
- **Scalability** - as the Waldur platform's user base will grow, so will the user activity and the number of generated events. Scalable solutions capable of handling large amounts of data are preferable to accommodate this.
- **Data Management Capabilities** – this is related to the ability of the SIEM to easily integrate with the existing infrastructure. For us it is important that the SIEM solution is capable of supporting different log formats and data sources as well as data filtering and querying functionality for proper search and analysis capabilities.
- **Incident Response** – if the SIEM is capable of incident response functionalities such as automated actions for isolating or blocking affected systems or user accounts. It would come as a benefit, as it will allow us to respond to identified security incidents in an active manner.
- **Community, Support and Learning** - commercial solutions often have dedicated customer support teams that may assist with SIEM deployment, configuration, integration, and any other issues that may arise. Free and open-source solutions rarely if ever possess such dedicated resources, instead relying on the user community for discussion and problem solving. The learning curve is important to consider as well as there is preference to utilize tools that may have familiar features and functionalities to those that are already in use. As such a comprehensible documentation with distinct explanation of setup, functionalities, configuration with provided examples. Community platforms where users can ask questions or require assistance. Use of familiar technologies or solutions would be an advantage.
- **Data Security, Log Storage and Retention** - whether or not there exist mechanisms for ensuring the security of the transferred data into the SIEM as well as the duration of storage of the logs for compliance and investigation purposes.
- **Rule Customization** - many security solutions come with sets of predefined configurations or detection and security rules. This may work well for applications using specific standards. However the ability to adjust the tool to meet Waldur's specific needs may require flexible customization options for the specific events that are being emitted by Waldur.
- **Containerization** - Waldur is designed as a cloud native application, as such the SIEM solution is preferred to be available in a containerized package (e.g., Docker) for ease of deployment, scalability, and management.

It should be noted that the SIEM requirements can change over time. The cybersecurity landscape is changing, new threats emerging and technology changing every year. The platform itself evolves and deploys new features and functionality, thus the SIEM and other security requirements may need adapting.

4.2 Low Level Design

In this chapter we will delve into more details regarding the tool selection, configuration, detection rule design and other specifications necessary for our test implementation. We will provide an overview of available Open-Source SIEM Solutions, evaluating their features and selecting a solution to be implemented. A brief overview of our currently logged event types and whether any gaps within current events exist will be included. We will take a look at the selected tools for the additional components that have been presented in the high level design. Finally, we will write a list of sample rules to be implemented in our SIEM solution for alert generation as well as offer details on our test environment setup, highlighting the components, services and tools used for our proof of concept setup.

4.2.1 Overview and Selection of Available Open-Source SIEM Solution

In this section we will take a look at the currently available open-source SIEM solutions on the market. We will narrow down our choice to a few tools that meet our task of a proof of concept test deployment. It is important to acknowledge that for our proof of concept we may not necessarily select the best tool or one that has all the desired features and covers all of our requirements outlined in the previous chapter. Our target aim is to evaluate the feasibility of a SIEM system for Waldur, validate its potential functionality, and assess if it is suitable for our needs of security monitoring of user activities within the cloud platform. During the proof of concept phase we may prioritize certain criteria or functionalities while accepting that some features may require a different approach or that the selected tool may not be the final choice for production.

Some security tools that are present within different open-source SIEM rankings and recommendations are Wazuh, AlienVault OSSIM, SIEMonster [23] [24] [25]. Additionally, we consider software tools that offer SIEM functionality but are not focused solely on it, rather the security being a part of the product along with other features. Given the potentially large volumes of data processed by Waldur, we took a look at data storage and indexing tools that are capable of handling such data sets and also come with security features that meet the requirements of a SIEM system.

Below we provide short summaries of each of the tools that was considered.

4.2.1.1 Wazuh

Wazuh is a free open-source SIEM and security monitoring platform [26]. Wazuh was built on top of OSSEC [27], which is an open-source host-based intrusion detection system (HIDS). OSSEC is capable of file integrity checking, log analysis, security alert generation, incident response and compliance monitoring. Wazuh preserves many of OSSEC's core functionalities, and in addition expands them further with the addition of better dashboards, integration with other services, and larger rule configuration database.

4.2.1.2 AlienVault OSSIM

AlienVault OSSIM (Open-source Security Information Management) is an open-source SIEM platform developed and maintained by AlienVault. Apart from the free OSSIM version, there is the commercial version called AlienVault USM Anywhere. OSSIM comes

with the basic SIEM functionalities although it lacks threat intelligence modules [22, pg. 9], which however is listed for the commercial USM solution [28].

4.2.1.3 SIEMonster

SIEMonster is a platform for security management, offering analytics, log collection, access control and incident response functionalities [29] [22]. The tool features a custom implementation of Wazuh in its XDR (Extended Detection and Response) processes. XDR describes a set of different security technologies or methodologies combined together to improve security operations and security of systems and infrastructure. In the past it came with a free community edition and a paid professional edition, however in the new V5 version there is no longer separation, with only the commercial version available.

4.2.1.4 OpenSearch

OpenSearch is an open-source data collection, analytics and aggregation platform, which is maintained by Amazon Web Services and is based on Elasticsearch [30]. It offers its users flexible search options with full-text queries, result filtering, and data aggregation all while maintaining performance even with huge data volumes. OpenSearch was made from Elasticsearch, also known as ELK stack, which was built on top of Apache Lucene and is schema-free with an SQL-like query language [31]. OpenSearch comes with its own security module that includes SIEM functionality [32]. Unlike Elasticsearch, OpenSearch is licensed under the open-source Apache license, providing all of its functionality for free. OpenSearch has been noted as a viable SIEM and log management platform by authors in [33, pg. 99], noting relevant features such as anomaly detection, alerting, flexible dashboards engine and data and index management. It is a powerful search and analytics tool, which makes it a potential choice as a SIEM solution for the Waldur platform.

4.2.1.5 Comparison of SIEM solutions

A comprehensive study of multiple SIEM solutions including the four above and their comparison has been just recently released in 2024 [22]. A test environment was set up by the authors to gather different metrics of the efficiency and performance of the above [22, pg. 10]. The test environment included end user machines with different operating systems, a firewall, an intrusion detection system, and a dedicated server running the SIEM software. DDoS attacks were simulated using traffic generators. Malware attack simulations that were targeting end user systems and performing file changes [22, pg. 12]. The paper is valuable as it offers performance evaluations and other metrics of some of the SIEM systems. The performance of the SIEMs were evaluated using EPS (Events per second) metrics, which look at the number of events that a SIEM can process at one time. For the EPS metrics evaluation, the highest EPS count was made for Wazuh reaching 14K events per second, with OSSIM being around 9.5K and SIEMonster 7.6K. On average, Wazuh's performance in terms of EPS was 30%-200% higher compared to other SIEMs, depending on the source of the traffic or systems. Similar evaluations for EPS count were not found for OpenSearch. It should be noted however that the proof of concept was conducted with emphasis on flexibility and adaptability of the SIEM for Waldur's requirements, rather than performance, which may be the subject of future work.

In order to provide a better overview of the potential SIEM solutions to assist us in the choice, the features of the systems are presented in a table according to the defined SIEM requirements in chapter 4.1.3.

Table 5. Comparison of SIEM solutions

Feature	Wazuh	AlienVault	SIEMonster	OpenSearch
Open source (licensing)	GPL 2 license. Requires publicly released code to be licensed under GPL as well.	GPL 2 license. Requires distributed software and publicly released code to be licensed under GPL as well.	Apache license 2.0. It is permissive. Allows modification and distribution with few restrictions. Derivative work not required to be licensed.	Apache license 2.0. It is permissive. Allows modification and distribution with few restrictions. Derivative work not required to be licensed.
Local Data Storage	Supported.	Supported.	Supported.	Supported.
Graphical User Interface	Kibana (open-source data visualization engine).	Dashboards and data visualization.	Kibana (open-source data visualization engine).	OpenSearch Dashboards based on Kibana.
Scalability	Supports horizontal scalability (more power with more machines) in clusters. EPS metrics show high numbers.	Centralized solution, less scalable. Does not support distributed environments. EPS metrics reads 9.5K.	Supports scalability with their price plan. 7.6K EPS metrics	Supports horizontal and vertical scalability. EPS metrics measurements not provided.
Data Management Capabilities	Allows for new data decoders and parses on top of in-built ones.	Allows for new plugins for new data sources.	No support.	Supports different data types and log formats with Logstash.
Incident Response	Included. Has Automated Incident Response functionality.	Included with the paid version of the platform.	Unclear from documentation. Task creation for team members and analyzers with threat analysis available. Some options require subscription.	Not included.

Community, Support and Learning	Good and comprehensible documentation. Active community with regular updates, forums, and discord server.	Strong community with dedicated forums for troubleshooting. Documentation intertwines with commercial USM, which makes it a bit more difficult to discern functionality.	Good comprehensible documentation available with videos, configurations, and explanations.	Dedicated forum, communication channels and comprehensive documentation available. Based on ElasticSearch, which is utilized in other projects and is familiar to the team.
Data Security, Log Storage and Retention	AES-256 for agent communication. Log retention has no limit, depending on available storage.	AES-128 for communication. Secure, albeit key length is less than AES-256. Log retention default is 5 days.	AES-256 for agent communication. Log retention has no limit, depending on available storage.	AES-256 for encryption. Supports flexible log retention policies, no limits and depends on available system storage capacity.
Rule Customization	Custom rule creation available.	Custom rule creation available.	Documentation is scarce, but video materials available on the process.	Offers custom Rule creation as well as a flexible query language.
Containerization	Docker images and docker compose files for different components are provided.	Not clear. Downloads come with ISO files. No docker images or container setup files found.	Docker images and Docker configuration files present.	Docker deployment is documented, containers provided for different components.

Based on the above table and in light of our requirements, two solutions stand out, OpenSearch and the Wazuh SIEM. Both support varying functionality, custom rule integration, and scalability for larger organizations. The two other solutions, while with their advantages, fall somewhat short, as they come with commercial versions that include better and more advanced functionality which is likely the primary focus of the development teams. In the case of SIEMonster it must be noted the possibility of dropping support for the community edition, as is suggested by their announcements for their new V5 version. Both Wazuh and OpenSearch are open-source, albeit with different licensing models, with Wazuh

having the more restrictive GPL 2 license, but possessing advantage in having advanced functionalities such as incident response modules.

Given the scope of the thesis and its limitations, for the purpose of a proof-of-concept test implementation to test the validity for a SIEM integration for Waldur, a single SIEM solution must be chosen. After consulting with the team, OpenSearch was suggested for the test implementation. Its familiarity due to being based on ElasticSearch as well as having a less restrictive license were major advantages. The license issue in particular would have likely resulted in the SIEM solution not being distributed together with Waldur in the event of an open-source licensing such as GPL 2 (although this would not exclude the possibility of use by Waldur's team). Due to Waldur's lack of security event monitoring software, the need for local data storage which is not reliant on data transfer to cloud, open-source requirements and modern capabilities, OpenSearch was selected for its SIEM functionality for the proof of concept implementation.

4.2.2 Visualization

From our SIEM comparison table, it can be noted that Kibana is a widely used visualization and dashboarding tool. In the case of OpenSearch, it is shipped with a visualization tool called OpenSearch Dashboards [34], which is a fork of Kibana. The inclusion of a software tool catered towards OpenSearch and being based on Kabana makes it an appealing choice, in line with the platform. Additionally, much like OpenSearch itself, the OpenSearch Dashboards are licensed under the Apache 2.0 open-source license. This is important, as Kibana is currently licensed by Elastic under the SSPL [35] license, which while being open-source includes several limitations, notable being the requirement of releasing the code of any modified versions of the software as well as in the case of distribution of software as a service models. This is a major factor as was discussed prior, as there is a strong requirement of less restrictive open-source licenses for software deployed or shipped along with Waldur.

4.2.3 Log Collector

Our proof of concept contains components for log collection and data enrichment functionalities. These components are vital, as a log collector performs the gathering of relevant data, ensuring it is acquired in a timely and accurate fashion. Conveniently, Opensearch comes bundled with Logstash [36], which is a data processing tool that supports data ingestion and transportation for different sources. Being bundled with OpenSearch, it provides native integration options with the platform, as well as supporting a wide range of data input options, including API endpoint polling, which is relevant as Waldur comes with a REST API endpoint that can be used to gather the logging data. Given the scope of the work and this being a proof of concept, it was decided to go with the bundled Logstash tool, with the possibility of swapping the log collection component still remaining available in the future.

4.2.4 Data Enrichment

Data enrichment is used to add more contextual and valuable information for our data sets, transforming the incoming logs and allowing for potentially better in-depth analysis of our data and SIEM rule adjustments. Enriching data with valuable information helps the SIEM in identifying threats or unusual patterns within our data, increasing the chances of detecting security incidents. Enrichment can come in different forms, such as adding geo location data, correlating data from other sources, historical user data and others. The end goal with the data enrichment process is to improve the incoming datasets for better accuracy in our SIEM detection.

Logstash, which comes bundled with OpenSearch, offers powerful data enrichment options with its filters and plugins. It allows mutations to be made on the incoming data, changing the values or adding new ones. Built-in filters [36, filters] include data processing and enrichment options such as GeoIP filter, which can be used to add positional information for events that include an IP address. HTTP filters allow web queries mid-processing to gather extra context data from external sources. Translation filter that allows replacement of values using predefined dictionaries, as well as custom code filters that allow writing custom logic for incoming events.

4.2.5 Alerting

The alerting component is responsible for generating alerts based on the results of data processing and analysis routines.. When the results align with defined rules and other criteria, indicating a potential security incident, the alerting component generates alerts that include detailed information about the event, such as its type, time, identifier and severity. The alerting component comes bundled with the SIEM solution, which in case of OpenSearch is its security analytics plugin and its related functionalities and alerting mechanisms.

4.2.6 Notification

Notifications must be created, managed and delivered to the security personnel based on the alerts generated by the alerting system. The personnel responsible for the security of the platform must be informed of on-going incidents via different communication channels such as email, chat or push notifications. OpenSearch comes with its own notifications plugin, which offers multiple communication integration options.

4.2.7 Analysis of Waldur Events

Before describing the SIEM rules to be implemented, it is necessary to review Waldur's existing event types that are being logged for the business processes. The events currently generated by Waldur can be found within the documentation [7] or in [9].

Several best logging practices, recommendations and standards have been consulted to serve as a base for security event logging, ensuring adherence with security practices. Information was taken from OWASP's logging suggestions [37], standards implemented by government institutions [38] and other common practices [39]. Although many of these logging practices and standards are aimed towards application-level logs and not business events, which are our focus, several gaps have been nonetheless revealed.

Waldur offers a wide range of event types separated into different categories such as Access, Authentication, Customers, Invoices, Payments, Permissions, Projects, Resources, SSH, Users, and others.

For instance, it has been found that while account lockout on multiple failed logins is implemented, it does not generate a log with a relevant event type. Account lockouts happen on consequent failed login attempts and may be an indication of brute-force attacks or unauthorized access attempts.

Furthermore, although failed authentication attempts are logged, the user object is not retrieved, thus the log information does not contain details such as user ID or account type. While not a major oversight, this may limit more in-depth detection requirements such as separate monitoring of failed login attempts for administrator accounts.

Finally, in sources such as [37, “Sensitive Data Changes”] and [38, pg. 1] there are suggestions to have logging implemented for sensitive data access and possibly even actions taken by accounts with administrator privileges. This should be taken into consideration whether in the context of business logs such actions should be actively monitored. Given the presence of payment transactions, invoices, and staff accounts such type of logging may be a valid approach and have been proposed.

Additional audits of event types and logs may be conducted in the future for further analysis of potential gaps.

4.2.8 Test Environment and Setup

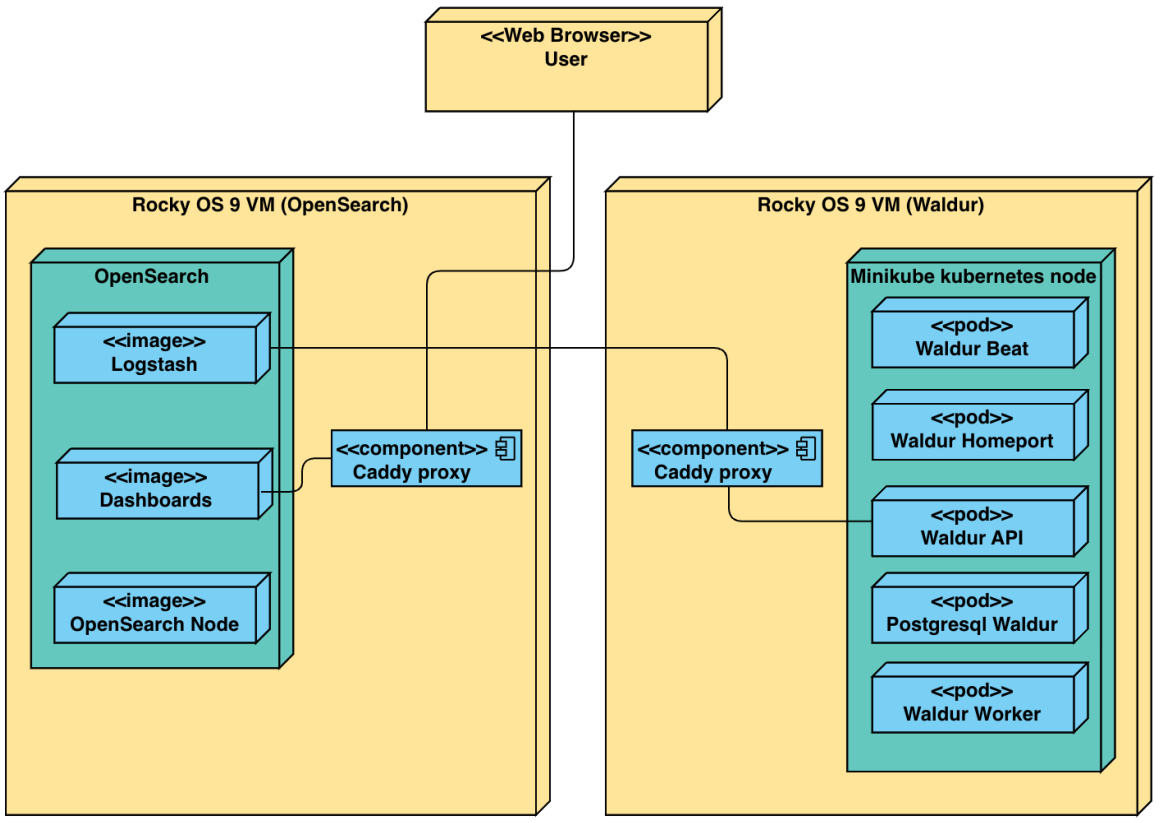

Figure 6. Test environment setup diagram.

The test environment was set up using 2 virtual machines, both running Rocky Linux, version 9.4. Hardware configuration was 16 virtual CPUs, 64GB of RAM and 90GB storage per machine. The first machine served as the host for the Waldur platform. There Waldur is deployed with Helm [40], which is a Kubernetes [41] package manager. Kubernetes is an open-source platform used for deployment and management of containerized applications, with useful features such as resource scaling and load balancing. Per instructions found in Waldur’s documentation [7], Waldur was via Helm on a Minikube cluster [42]. Minikube is a lightweight tool that allows the deployment of a local Kubernetes cluster. It is often used for testing and development purposes as it offers a fast and easy way to set up an environment. The second virtual machine hosts the OpenSearch stack, which is run with Docker [43]. Docker is a set of tools that allow organizations to deploy applications and services in a quick and compact manner. Docker takes advantage of containers, which can be described as a compact package that stores the application within, along with all of the necessary packages, configurations and dependencies that are required for the application to run. They can be easily transferred between different systems and run without the need

for additional system preparation or setup, allowing for companies to manage software and applications more effectively.

To allow secure external access and communication for the machines, a reverse proxy was implemented with Caddy [44]. Caddy simplified HTTPS configuration, which allowed for secure communication with the services as well as web browser access to both platforms.

The OpenSearch platform setup includes Logstash [36], which is used to ingest, transform and send data to its destination (OpenSearch nodes) as well as OpenSearch Dashboards, which provide a user interface that allows the visualization of the OpenSearch data. An important feature of Logstash are modular components called plugins. They are configured within Logstash pipeline to perform various tasks, such as log collection, log data modification and enrichment, and data output. The plugins that have been utilized in proof of concept are:

- **Date** - used to parse the "created" field into a timestamp for the event to be then indexed into OpenSearch.
- **Mutate** - a filter plugin which allows for mutations on incoming fields, used to remove or rename fields during data enrichment.
- **Ruby** - a filter plugin that allows executing custom code, used in data enrichment processes.
- **OpenSearch** - an output plugin which is used to establish a connection with the running OpenSearch node and send data to it.
- **HTTP** - a filter plugin used to gather data for enrichment purposes from the running OpenSearch node.
- **Exec** - allows execution of our custom Python script, which is used to query the Waldur API and gather the event data.

4.2.9 Software Components

The primary software components that are involved in our SIEM system are Waldur, Logstash, OpenSearch, OpenSearch Dashboards, Alerting, Notifications and Caddy.

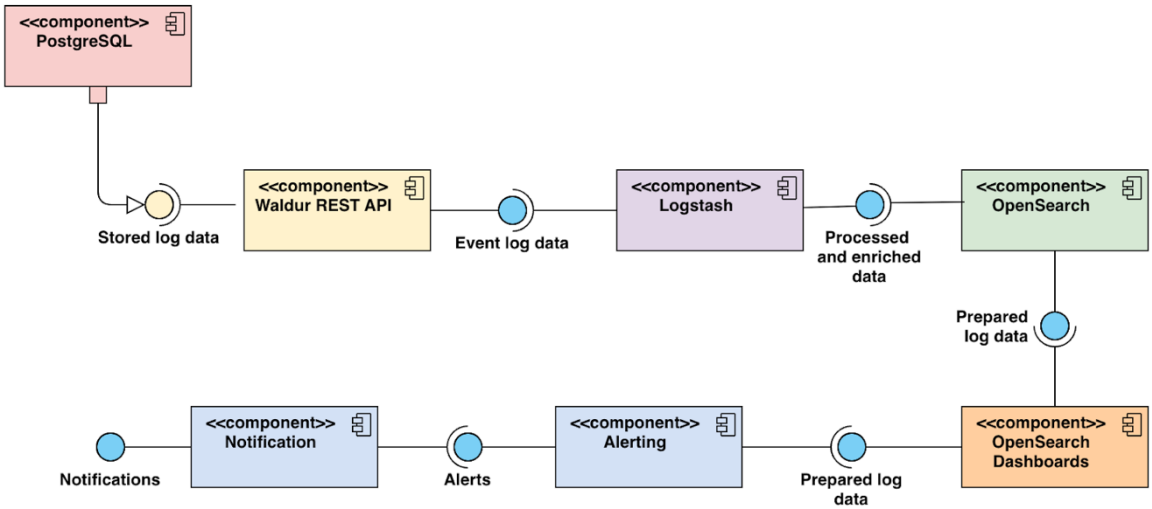

Figure 7. Components involved in the proof of concept implementation.

Additional description of components and their roles:

Table 6. Description of components and their setup notes.

Name	Role	Function	Setup
PostgreSQL	Log Source	Stores events and logs related to various system or user activities.	Runs on a local Kubernetes cluster with Minikube, on a separate virtual machine running Rocky Linux version 9.4. Part of Waldur Platform deployment.
Waldur API	Log Access	Allows access to the stored business event logs. Supports filtering.	Runs on a local Kubernetes cluster with Minikube, on a separate virtual machine running Rocky Linux version 9.4. Part of Waldur Platform deployment.
Logstash	Log Collector, Data Processing and Enrichment	Gathers the events from Waldur Platform, applies necessary filters, transformations, and preprocessing, and then forwards the data to the OpenSearch cluster.	Separate Docker container within the Docker deployment on the system running OpenSearch.
OpenSearch	SIEM and Analysis	Used to index and store the processed events. Offers various additional functionalities, including the SIEM security features.	Two nodes, each in a separate Docker container, running on a virtual machine with Rocky Linux 9.4.
OpenSearch Dashboards	OpenSearch data visualization engine	Provides a flexible and comprehensible user interface to navigate and explore the stored data as well as utilize SIEM functionalities.	Separate Docker container running on the OpenSearch virtual machine.

SIEM Alerting	SIEM Alerting	Part of OpenSearch Security Analytics. Holds the created detection rules and monitors incoming data for anomalies and generates relevant alerts.	Configured with rules and alert triggers within OpenSearch Dashboards.
Notification	SIEM Notifications	Dispatches notifications of alerts triggered by the SIEM via various channels such as email, or other communication tools or platforms.	Configured to work together with the SIEM module via OpenSearch Dashboards.

4.3 SIEM Rules

In this section, we will design 10 sample security rules that are to be integrated with our chosen SIEM solution. 10 rules should be a good starting point for SIEM evaluation. These rules are categorized into three groups: Authentication and Access Control, Resource Management, and User Management. These categories were selected with consideration of OWASP's security logging suggestions. For instance, logs related to authentication and access control as well as user account management (including high-risk activities such as account activation/deactivation and privilege changes) are listed to be logged always if such a possibility exists. Each rule offers a description of its functionality and purpose, the security events it monitors and its severity level.

The Authentication and Access Control category was selected to implement rules that help in detecting potentially suspicious login activities and unauthorized access attempts.

The Resource Management category's rules are used to detect mass resource creation, deletions, and update events. As one of our platform's core activities is resource management, these rules will monitor cases of mass resource manipulation within short time frames within the platform. This allows effective detection of any unusual activity related to resource management.

The final category selected, the User Management category includes rules that monitor user account manipulation. This includes activities such as account activations and deactivations, user deletions, password updates, and role changes. These rules help in detecting important changes to user accounts and permissions, ensuring that such activities are monitored and verified.

A table is with the rules is provided, including rule descriptions and relevant event types.

Notes:
Rule severity is used to describe the importance of the specific event or alert associated with the event. This can be used for priority in cases of multiple on-going alerts being active, with the more critical ones taking precedence over the others.

Event types mentions the expected value of the field "event_type" in the incoming events. All events generated by Waldur are separated into different categories, and an event type from the category is assigned to the specific event, detailing its context.

Table 7. Sample Rules.

Rule No.	Name	Description	Severity	Event Types
Authentication and Access Control				
1	Detect Successful Authentication Attempts in a Short Timespan.	Detects multiple successful login attempts with a specific timeframe. Multiple successful logins in a short time frame could indicate activities such as account sharing or credential leaks. It was decided for the time interval to be 5 minutes.	Medium	auth_logged_in_with_username
2	Detect new IP login.	Detects if the user logged in from a previously not seen IP address. New IP address may indicate that the account is in use by someone else than the original user.	High	auth_logged_in_with_username
3	Token Lifetime Updated to be Indefinite.	Users are given an API token which may be used in requests to the API to provide authorized access. By default, API tokens are set to expire after a set amount of time passes which is aimed to limit the consequences in the event the token somehow becomes compromised. This rule will monitor for the API token to be updated to have an indefinite lifetime.	High	token_lifetime_updated
4	Detect Multiple IPs Used by a User in a Short Time Frame.	Detects cases where a user account performs activities from multiple IP addresses within a short period. Switching between IP addresses may suggest that an attacker is attempting to avoid detection or that the account is being accessed from different	High	Events should have the IP within the context object as well as the user's UUID. Requires custom implementation

		locations. Sample time frame selected was 15 minutes.		
Resource Management				
5	Mass Resource Creation	Detects if a large number of resources are created within a short time period. This can indicate potentially malicious or unintended activity.Sample time frame selected was 15 minutes.	Medium	resource_creation_succeeded
6	Mass Resource Deletion	Detects if a large number of resources are deleted within a short time period. This can indicate potentially malicious or unintended activity. Sample time period selected is 15 minutes.	Medium	Resource_deletion_succeeded
7	Mass Resource Update	Detects if a large number of resources are updated within a short time period. This can indicate potentially malicious or unintended activity. Sample time period selected is 15 minutes.	Medium	Resource_update_succeeded
User Management				
8	User Role Changed	Detects events that contain changes to the user's role within the system. This is important to monitor user permissions and access levels. Unauthorized or unnecessary role changes may be an indicator of security threats such as privilege escalation.	Medium	role_updated, role_revoked, role_granted
9	Detect Admin User Account Activation and Deactivation	Detects activation and deactivation activities of administrator accounts. Due to their elevated permissions, admin accounts are good targets for malicious actors. Monitoring admin account activations and deactivations can help detect unauthorized admin account manipulations.	High	user_activated, user_deactivated

10	Detecting User Deletion	Detects deletion of user accounts. Deletion of user accounts are rare within the Waldur platform, and may be indicative of suspicious activities.	High	user_deletion_succeeded

While the above rules do not cover all of the possible security scenarios, they will serve as a basis for our proof of concept implementation of our selected SIEM, and can be improvised and expanded upon in the future.

4.4 Proof of Concept OpenSearch SIEM Implementation

The documentation contained sample Docker files to build the Docker environment with Docker compose. While OpenSearch is a fork of ElasticSearch, thus sharing many of the core features and functionalities, by default the Logstash setup did not contain the OpenSearch output plugin. The output plugin is the final part of the Logstash pipeline and defines the end destination for the data, in our case our OpenSearch cluster. As such, a custom Dockerfile was made to include the necessary dependencies and plugins for our setup. The Dockerfile was then used by Docker for building the Logstash image. The data that was being sent to OpenSearch could be accessed via the dashboards user interface, which was exposed to the outside using Caddy proxy.

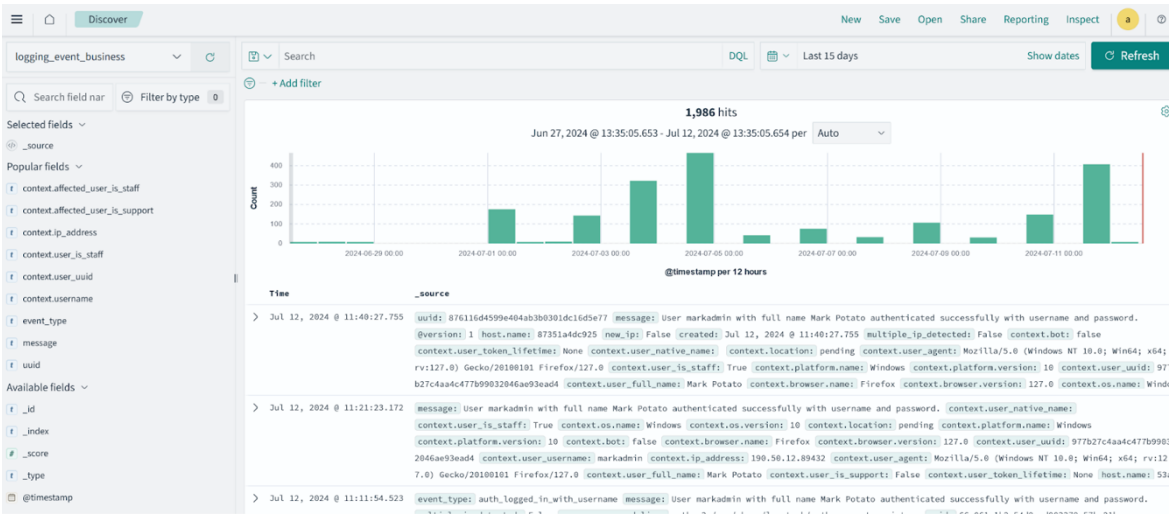

Figure 8. OpenSearch dashboards example with incoming data.

OpenSearch comes with its own set of security features and solutions. The OpenSearch Security Analytics is the in-built SIEM solution provided by OpenSearch [32]. Notable security features of the SIEM functionality include:

- Detectors: they are used to identify threats within the ingested logs. They generate findings, which are used for identifying suspicious activities. These findings are used to highlight events with unusual activities.
- Log types: used to define the various data sources which are used for threat detection. Detection rules are categorized according to log types. Log types are used in the process of detector creation, automatically selecting relevant rules for the specific data source.

- Detection rules: The specific security rules that contain the logic that is applied to incoming events and mark matching events to be of interest. The detection rules are based on Sigma rules [45]. These rules are written in a standardized format and can be applied to different SIEM systems. They focus on detecting matching criteria on incoming events.
- Findings: Generated when a detector matches an event with a defined rule. While not necessarily an indication of a threat, they serve to isolate events of interest.
- Alerts: Detectors can have triggers which may be configured to fire alerts based on specific conditions such as detection rule matching. Notifications can be sent via different communication channels such as email or Slack [46].
- Correlation: Correlation can be used to adjust findings based on different log types and detect specific relationships between seemingly unrelated events. Correlation rules can be set up to match findings or perform queries.

With the above in mind, an attempt was made to implement the sample security rules that we have designed prior. Figure 9 shows an example of "Token Lifetime Updated to be Indefinite" rule:

```
id: qrjNppAB80N40iH9RHzJ
logsource:
  product: waldur_platform
title: Waldur_Token_Lifetime_None
description: Detects if a user's API token was updated to be
permanent.
tags: []
falsepositives: []
level: medium
status: experimental
references: []
author: ''
detection:
  condition: Selection_1
  Selection_1:
    event_type|all:
      - token_lifetime_updated
    context.affected_user_token_lifetime|all:
      - None
```

Figure 9. Rule that Detects events with token_lifetime_updated field, and checks for value "None" (indefinite).

It should be noted that Sigma Rules focus on detecting specific strings, fields, error codes or keywords within individual log entries. This works well for detection based on expected log field information or Indicators of Compromise (IoC). Because of this dependence on specific values within the logs, more complex rules required additional work to be properly implemented. Detecting complex attack patterns and unusual behavior spread out through multiple log files requires additional information and data enrichment for our log data, in order to take advantage of Sigma rules in such cases. An example of a rule requiring

additional data enrichment is the detection of multiple resource creation events for a specific project. This approach requires checking whether or not there were multiple event logs that contained the same value for the project's id within a specific timeframe. We would need to gather this information and append the result to the log that is being processed, so that Sigma rules can be later applied.

As the information is being stored in our OpenSearch cluster, we may query our available data to check how many event logs in a specific timeframe field had the same values for event type and project id, using OpenSearch query search language (DSL). An example of a query that takes in the event type and project uuid of the incoming log, and checks against data within the last 15 minutes for the amount of such log events is shown in Figure 10:

```
"size": 0,
"query": {
  "bool": {
    "must": [
      {
        "range": {
          "@timestamp": {
            "from": "%{[created]}||-15m",
            "to": "%{[created]}||+1m"
          }
        }
      },
      {
        "terms": {
          "event_type": ["%{[event_type]}"]
        }
      },
      {
        "term": {
          "context.project_uuid.keyword": "%{[context][project_uuid]}"
        }
      }
    ]
  }
},
"aggs": {
  "resource_count": {
    "value_count": {
      "field": "context.project_uuid.keyword"
    }
  }
}
```

Figure 10. Query to get the number of resource manipulation events, within a specific timeframe.

The above query would return the number of resources that were updated, created or deleted within 15 minutes of the incoming event's timestamp for a single project, depending on the

type of event. The query is run inside of Logstash's filters plugin on incoming events that match the event type to be one of the resource manipulated ones. The "filters" section within the Logstash pipeline is used to process and transform incoming data. As users perform different activities, including managing projects and resources, we seek events containing information about these activities, specifically events that contain resource management information. Once an incoming event is detected to have the relevant fields, we execute a HTTP filter with our query against our opensearch node and retrieve the response with results. The response will contain the results of the executed query, which are afterwards used in our data enrichment process by creating new fields or generating new events for the SIEM rules. The HTTP filter allows the integration of HTTP web applications and services as well as APIs. Logstash comes with a filter plugin named "Elasticsearch filter plugin" [47], which is optimized for querying Elasticsearch nodes. While the HTTP filter is more generic, Elasticsearch filter plugin provides fields and functionalities for seamless integration with the Elasticsearch cluster. Unfortunately, despite the similarities in core functionality of the platforms, the Elasticsearch filter does not support the OpenSearch platform. The OpenSearch community maintains a number of their own plugins, catered towards OpenSearch, including OpenSearch filter plugin. The attempt to implement the plugin however was unsuccessful, due to missing functionality for SSL/TLS security support. A number of options that were present in the Elasticsearch's counterpart were absent. A few issues on GitHub related to a similar problem have been discussed [48][49]. Due to time constraints and a lack of experience in modifying the plugin's code, it was decided to use the HTTP filter plugin instead.

```
if [context][user_uuid] {

 http {
    url => "https://opensearch-node1:9200/logging_event_business/_search"
    verb => "GET"
    ssl_verification_mode => none
    user => "${OPENSEARCH_USER}"
    password => "${OPENSEARCH_PASSWORD}"
    body => '{"size": 0,"query": {"bool": {"must": [{"exists": {"field": "context.user_uuid","boost": 1}},{"range": {"created": {"from": "now-5
0m/m","to": "now/m","include_lower": true,"include_upper": false,"boost": 1}}},{"term": {"context.user_uuid":  "%{[context][user_uuid]}"}}],"adju
st_pure_negative": true,"boost": 1}},"aggregations": {"multiple_ips": {"cardinality": {"field": "context.ip_address.keyword"}}}} '
    body_format => "json"
    target_body => "multiple_ip_response"
  }
```

Figure 11. Example of HTTP filter for multiple IP checking.

Afterwards, we proceed to check the retrieved results, assessing whether or not multiple IPs have been detected for the user.

```
ruby {
  code => "
    if event.get('[multiple_ip_response][aggregations][multiple_ips][value]').to_i > 1
      event.set('multiple_ip_detected', 'True')
    else
      event.set('multiple_ip_detected', 'False')
    end
    event.remove('multiple_ip_response')
  "
}
```

Figure 12. Check if multiple IPs have been retrieved and create a new field depending on the result.

We add a new field "multiple_ip_detected" to the incoming event with a value of True or False, depending on whether there was more than one IP address for the user. This field will be later used in our Sigma rule to create findings and have an alert to be triggered by it.

```
id: GrjpppAB80N40iH91n9Y
logsource:
  product: waldur_platform
title: Waldur Multiple IP Detected for User
description: Multiple IPs detected for a user within short time frame.
tags: []
falsepositives: []
level: medium
status: experimental
references: []
author: ''
detection:
  condition: Selection_1
  Selection_1:
    multiple_ip_detected|all:
      - 'True'
```

Figure 13. Sigma rule for multiple IPs detection.

Alert trigger name	Detector	Status	Alert severity
Multiple IP for User Detected	Waldur Authentication and Access Control	Active	2 (High)

Figure 14. An example of the alert being fired within our alerts module.

Some rules required more complicated logic written in Ruby programming language in the logstash's filters. An example being the need to create new custom events to act as triggers for specific rules. This was the case with our resource rules, with the associated query posted above. The response returned by the HTTP filter would contain information as to the number of similar events for the same project that have been seen within the timeframe. In such cases using Logstash's Ruby filter, code logic was written to generate new events at these timestamps. These events would contain an "event_type" field, the value of which would be a custom event_type that would act as trigger for our Sigma rules.

```
code => '
  require "securerandom"
  require "time"

  aggregations = event.get("[opensearch_resource_data][aggregations]")

  aggregations.each do |agg_name, aggregation|
    case agg_name
    when "resource_creation_succeeded"
      buckets_path = "[opensearch_resource_data][aggregations][#{agg_name}][resource_count][buckets]"
      message = "More than 10 resources created in a short time span for project"
      event_type = "resource_creation_exceeded"
    when "resource_deletion_succeeded"
      buckets_path = "[opensearch_resource_data][aggregations][#{agg_name}][delete_count][buckets]"
      message = "More than 10 resources deleted in a short time span for project"
      event_type = "resource_deletion_exceeded"
    when "resource_update_succeeded"
      buckets_path = "[opensearch_resource_data][aggregations][#{agg_name}][update_count][buckets]"
      message = "More than 10 resources updated in a short time span for project"
      event_type = "resource_update_exceeded"
    else
      next
    end

    buckets = event.get(buckets_path)

    if buckets && !buckets.empty?
      buckets.each do |bucket|
        if bucket["doc_count"] > 0
          new_event = LogStash::Event.new(
            "project_uuid" => bucket["key"],
            "created" => Time.now.utc.iso8601,
            "event_type" => event_type,
            "message" => message,
            "uuid" => SecureRandom.uuid
          )
          new_event_block.call(new_event)
        end
      end
    end
  end

  event.cancel
'
```

Figure 15. Logstash Ruby filter code to generate new events for resource manipulation cases.

It must be also noted that the HTTP Poller input plugin which we utilized initially to call our events API endpoint to poll our event data did not support pagination functionality. Waldur's API endpoint offers pagination, with maximum page size being 100. This means that only 100 events at a time can be retrieved from the API. In the event that more than a 100 events were generated between our data ingestion, we would need to use pagination to ingest all of the events. The total number of entries that were retrieved were present within the API's response header "X-Result-Count". This header became the basis for a custom python script, which was written to gather all of the available data within a specified time period. This workaround allowed us to implement proper pagination and gather all of the available events from a specified time period.

OpenSarch comes with a Notifications component plugin, which allows users to send alerts and other relevant notifications from OpenSearch directly to different communication channels, such as Email, Amazon Chime or Slack. This plugin integrates with OpenSearch's security features to allow real-time communication of alerts and anomalies. The setup can be done for separate alert triggers that are defined within our detectors. Example provided below:

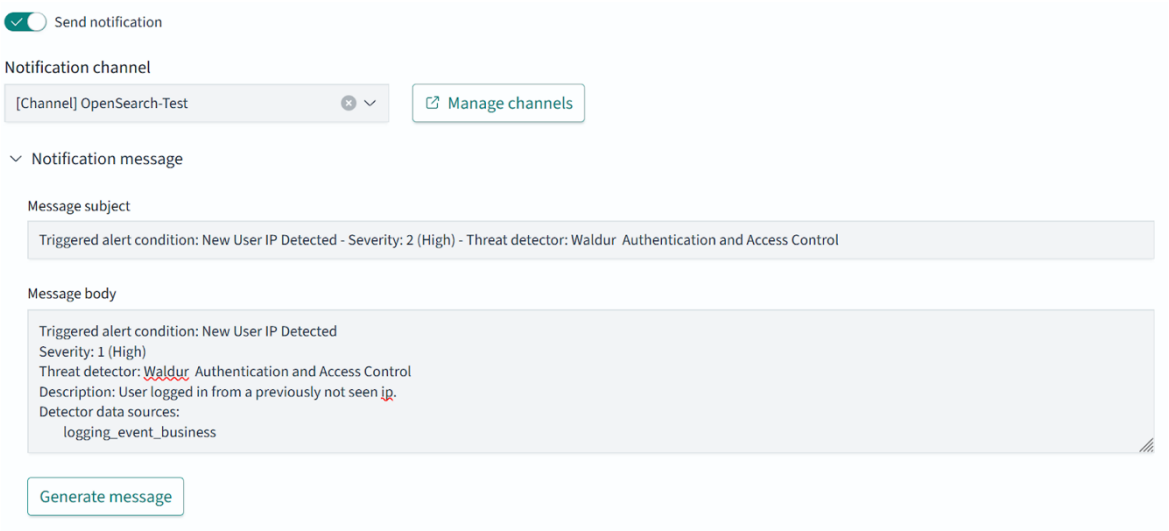

Figure 16. Setup of Slack notification for the "New User IP Detected" rule.

The notification channel was set up with Slack's webhook [50]. This allows for effective security team notification and coordination in case of security incidents.

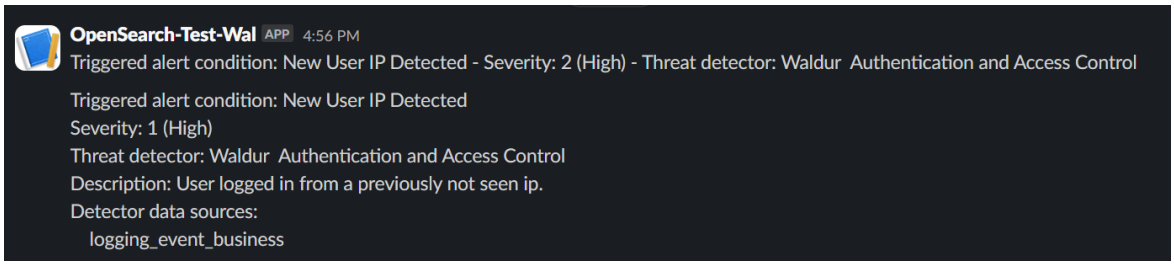

Figure 17. Example of the alert that is seen by the Waldur team in Slack.

4.5 Proof of Concept Validation

In this chapter we will attempt to validate the proof of concept. This approach will involve two phases: testing with synthetically generated data to make sure that all sample rules trigger correctly and then transition to real historical data from a live Waldur instance. The goal is to confirm that the SIEM solution effectively identifies and responds to security events and provides a well integrated security workflow.

4.5.1 Validation With Synthetic Data

For validation of our chosen SIEM system, the first part was using artificially generated synthetic data. This will allow us to test and verify the function of our sample rules in a controlled environment.

A set of synthetic data was prepared. The dataset contains an array of Waldur consistent events, which are adjusted to specifically trigger each of our predefined sample rules. The sample of the dataset is provided with GitHub [Appendix 1]. With this approach we have verified that the assumed threat scenarios would be detected by the SIEM and alerts will be generated.

07/31/24 3:12 pm	Multiple IP for User Detected	Waldur Authentication and Access Control	Active	2 (High)	
07/31/24 3:12 pm	New User IP Detected	Waldur Authentication and Access Control	Active	2 (High)	
07/31/24 3:12 pm	New User IP Detected	Waldur Authentication and Access Control	Active	2 (High)	
07/31/24 3:12 pm	API Token Permanent	Waldur Authentication and Access Control	Active	3 (Medium)	
07/31/24 3:12 pm	Admin Account Activated	Waldur User Management	Active	2 (High)	
07/31/24 3:12 pm	Waldur User Deleted	Waldur User Management	Active	2 (High)	
07/31/24 3:12 pm	User Role Revoked	Waldur User Management	Active	3 (Medium)	
07/31/24 3:12 pm	User Role Updated	Waldur User Management	Active	3 (Medium)	
07/31/24 3:12 pm	User Role Granted	Waldur User Management	Active	3 (Medium)	
07/31/24 3:12 pm	Admin Account Deactivated	Waldur User Management	Active	2 (High)	
07/31/24 3:12 pm	Mass Resource Deletion Detected	Waldur Resource Manipulation Detector	Active	3 (Medium)	

Figure 18. Example of the alerts triggered by inputting the synthetic data.

The detection of all events and execution of alert triggers have been successful and confirmed the rule configurations as well as the SIEM functionality of the OpenSearch platform in identifying and alerting on threats.

4.5.2 Validation With Real Data

In this chapter we provide a summary of our testing of the proof of concept with real live data taken from the running Waldur instance. The synthetic data used in the previous chapter provided us with an overview of our alert triggering within a controlled environment. To ensure the potential of our chosen SIEM and its practical application, we must validate it with events taken from the production environment. The events taken for the validation were from the past 6 months, from February until August 2024. The events have been pre-filtered during the API requests, with only events that contained relevant event types for our sample rules were ingested. The total number of events that have been gathered by Logstash were 22600. This real data contains actual events and provides a test of the SIEM's capabilities. Some statistics for event counts are presented below:

Table 8. Table showcases the number of events by event type in the sample real data.

Event Type	Count
resource_creation_succeeded	7634
resource_deletion_succeeded	13535
resource_update_succeeded	0
role_granteed	636
role_revoked	527

role_updated	6
auth_logged_in_with_username	251
token_lifetime_updated	2
user_activated	2
user_deactivated	4
user_deletion_succeeded	3

Additional insight into event distribution can be gained from the following diagrams:

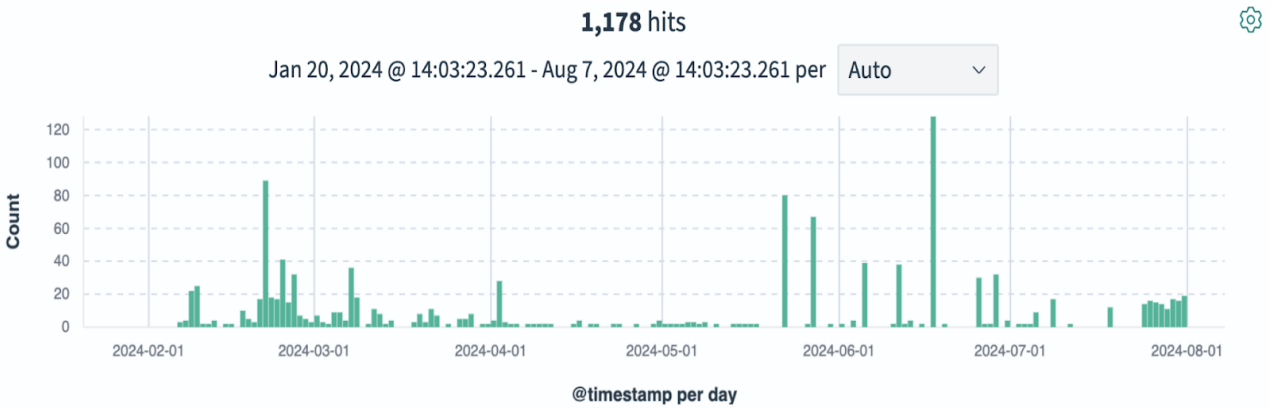

Figure 19. Distribution of events for User Management category (event types: role_revoked, role_granted, role_updated, user_deletion_succeeded, user_activated, user_deactivated).

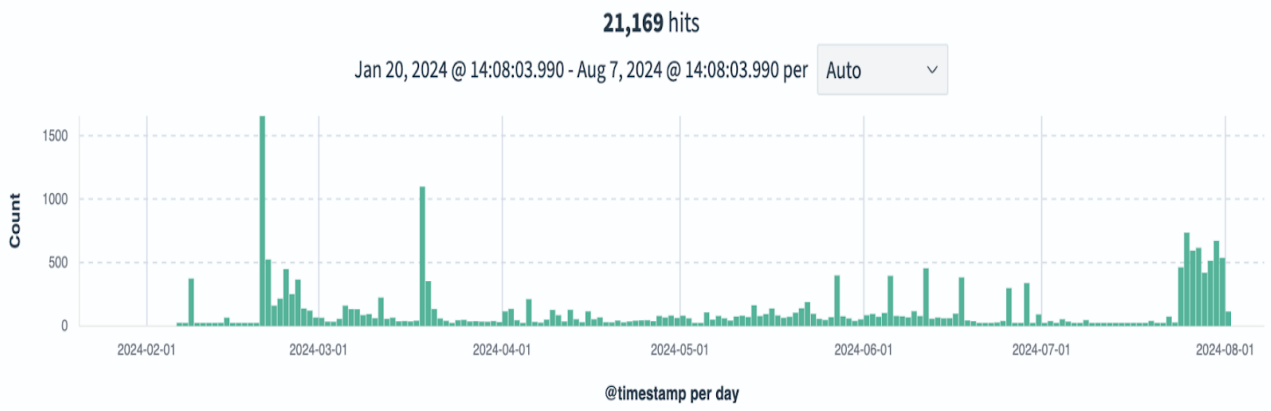

Figure 20. Distribution of events for Resource Management category (event types: resource_creation_succeeded, resource_deletion_succeeded, resource_update_succeeded)

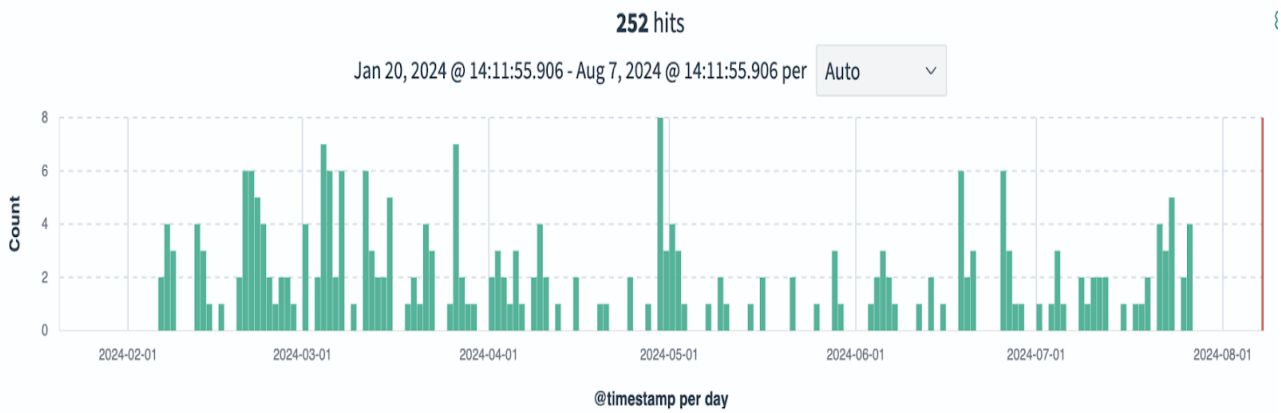

Figure 21. Distribution of events for Authentication and Access Control category (event types: auth_logged_in_with_username, token_lifetime_updated).

We will focus on presenting our findings with a table and graphs of the applied rules and the corresponding number of generated alerts.

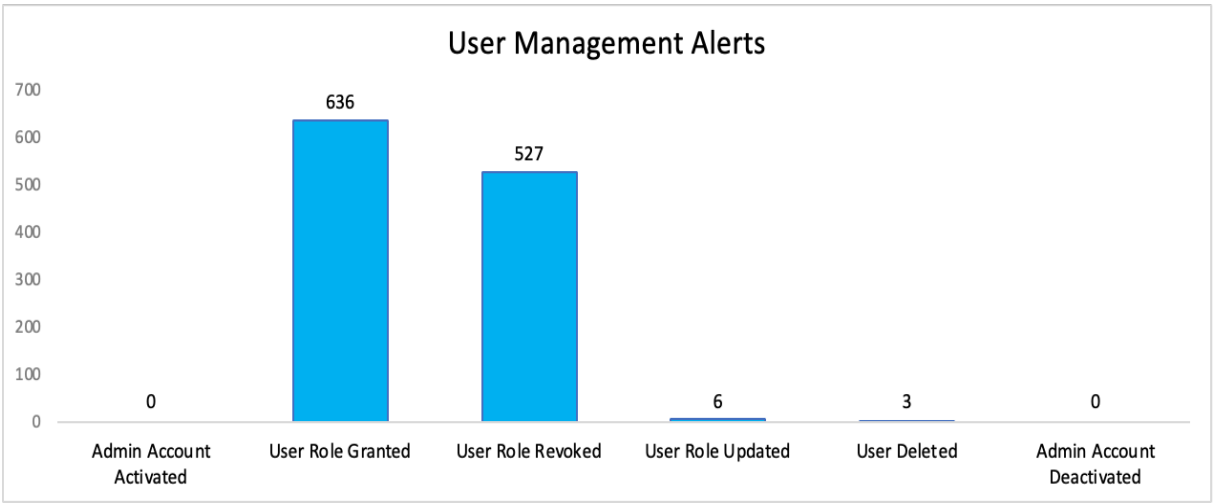

Figure 22.Graph showcasing the number of alerts within the User Management category.

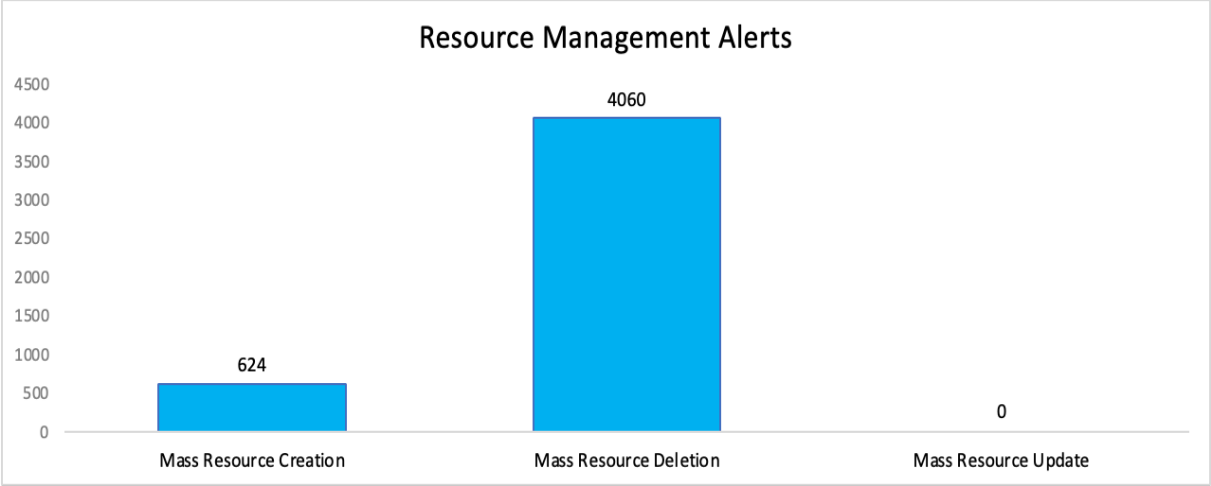

Figure 23. Graph showcasing the number of alerts within the Resource Management category.

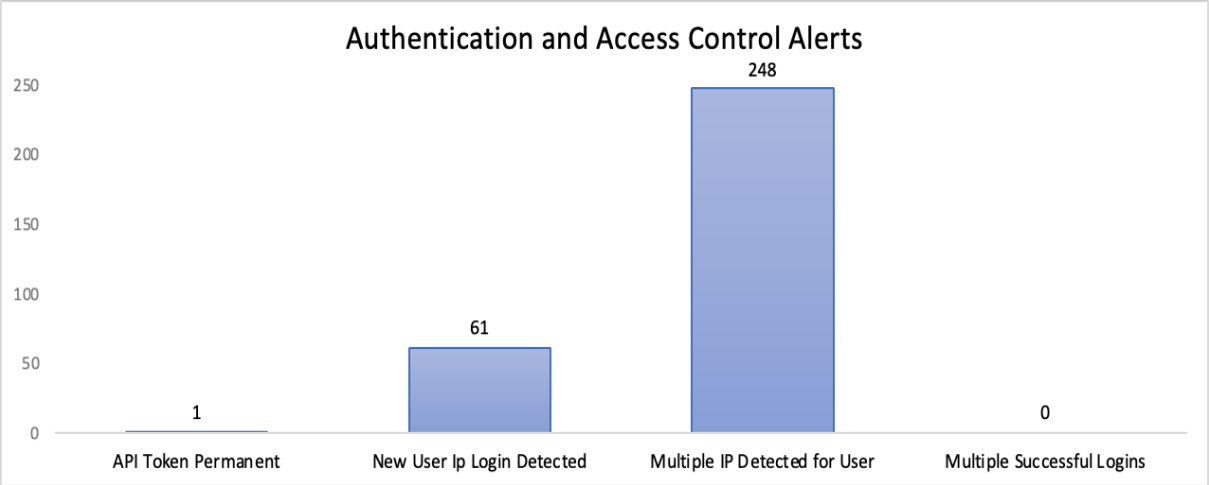

Figure 24. Graph showcasing the number of alerts within the Authentication and Access Control category.

Table 9. Results of real data validation, number of alerts.

Alert Type	Amount
Admin Account Activated	0
Admin Account Deactivated	0
User Role Granted	636
User Role Revoked	527
User Role Updated	6
User Deleted	3
Mass Resource Creation	624
Mass Resource Deletion	4060
Mass Resource Update	0
API Token Set To Permanent	1
New IP Detected	61

Multiple IP Detected For User	248
Multiple Successful Logins	0

As this was a proof of concept implementation, we have performed manual checking of the generated alerts, to confirm whether the rules were triggering in valid cases, and have found them to be overall consistent.

During testing with large datasets, it was discovered that when a large number of events are created within a short timeframe, and the Logstash's HTTP filter plugin is used for their enrichment, additional alerting events are generated for each subsequent event past the initial trigger. For example, this happened with Resource Management rules that detect if more than 10 resources were manipulated within the last 15 minutes within a specific time period. In our data, there were cases where hundreds of "resource_deleteion_succeeded" or "resource_creation_succeeded" events were generated within a minute. As such for each new incoming event past the 10th for these event types was generating an alerting event for our detectors. This resulted in multiple alerts being generated that would point towards the same project and timeframe. This can be considered a valid approach, as the incoming events are valid and highlight an on-going active issue of a large case of resource creation going on. It was discussed that this behavior may be a signal of unusual persistent behavior within the data, highlighting that attention should be diverted towards the on-going situation. Additional adjustments to the queries may be needed in the future, for better balancing of the alert event volumes in such cases, for better clarity and usefullness. In future work, methods aimed at suppression of the duplicate alerts may also be looked at.

Additionally, due to Logstash's HTTP filter plugin's parallel event processing, in cases of large batches of incoming events, there was a delay between an event being processed and becoming indexable for retrieval for a query. This caused some queries to miss retrieving the expected data on time for the results. A workaround was found by introducing a minor delay in injection of specific event types.

To check whether or not an IP is new, we must have a set of IP addresses to check against. Since we had no data available from production before validation, to check against, the detector had no data to use as a base for confirmation whether the IP was indeed not user prior. This happens during first time data indexing and injection, and as such the new IP alerts generated during this time can be considered false flags.

5 Conclusions

5.1 Summary of Findings

We have gone over the various uses and requirements of SIEM systems. We have looked into the Waldur platform, offering information about its operation and functionalities along with the event logging mechanisms within, identifying various event types, their potential relevance for SIEM implementation and security monitoring as well as the process for logging the events. Furthermore, a proof of concept was designed and implemented, which involved a full event log flow from the log source, with data processing and enrichment performed in Logstash with the resulting data then sent to OpenSearch. This filtered information was then ingested by our chosen SIEM software, which used the processed data for analysis and alert generation, for the security team to be then informed of critical events.

Based on the conducted analysis and practical test implementation, it can be concluded that the integration of a SIEM solution into Waldur's infrastructure could be a viable approach to enhance the platform's security posture. From observation, OpenSearch has the potential to meet Waldur's security requirements, given the functionality, scalability, and compatibility of the software.

It should be noted that several difficulties have been encountered with the proof of concept implementation of OpenSearch and its security package.

- **Limited Support of Specific Plugins** - Logstash's native support of some OpenSearch filter plugins is lacking or missing in features. This is of note, as similar plugins made for ElasticSearch (which served as the basis for OpenSearch) are well maintained and offer similar functionality (but are not compatible with OpenSearch). In our case it required us to implement the HTTP filter for our OpenSearch queries for data enrichment, which was a less effective solution than a dedicated OpenSearch filter.
- **Lack of Pagination Support for HTTP Polling in Logstash** - Logstash's HTTP Poller input plugin also lacked native support of pagination for large datasets. This required implementing workarounds, in our case custom Python scripts to be triggered by Logstash which would handle the pagination. This adds another layer of maintenance and complexity, as the script will require updating as will the Logstash's configuration in case of changes to the API. The script may also not perform as optimal as would a dedicated Logstash filter plugin.
- **Sigma Rules and Their Limitations** - Sigma rules come with their strengths and weaknesses as platform-independent detection rules. There have been difficulties in utilizing the provided SIEM rules for detection of patterns of specific sequences. They also lack contextual awareness of the data. Our experience during the proof of concept implementation suggests that the default Sigma rule detection provided by OpenSearch should be combined with other tools or data enrichment solutions for more complicated and nuanced detection cases.
- **Lack of Specific Functionality** - OpenSearch security is lacking in several non-critical but useful SIEM functionalities or features. Notable ones are incident response capabilities, for automated response pipelines, such as user account or IP blocking, access limitations. Integration with external security platforms and advanced threat detection is also limited if not absent.

- **HTTP Filter Data Enrichment Delay** - during the active phase of real data testing an issue was discovered during Logstash's data filtering and enrichment. The custom queries which were meant to detect complex sequences of suspicious activities, were not retrieving the results on time for the conditional checks for alerting event creation. The issue was with Logstash's parallel processing of events in batches, as there was a delay in indexing the incoming events before they can be seen and retrieved by a query. In cases of large quantities of events in a short lifespan this resulted in the querie's results not containing the expected information to add a new field value for the Sigma rules or generate an event used for alerting. Any such behavior is in case of mass resource creation events. In case there were 15 resource creation events incoming in a single batch, created seconds apart, the events would be processed in parallel, resulting in the query returning the wrong value for event counts, as the data was simply not there yet. The delay was between data being injected into OpenSearch and the query firing was roughly a second, but it was sufficient to break these rules. A temporary workaround was the introduction of a short delay for event injection.

To conclude, the proof of concept implementation served as a valuable experience, showcasing the strengths of OpenSearch's SIEM functionality, the potential relevance of such tools for analysis of the business event logs generated by the platform. It also revealed areas that require further development, design and consideration in terms of security needs and implementation. We can state that a SIEM solution is a valid consideration for the security needs of Waldur, with OpenSearch security being a valid candidate for the task.

5.2 Future Work

The potential gaps of the conducted research must be noted. Given the limited scope of the work, certain limitations to the results can apply that may require additional research in the future.

- **Rule Coverage** - The coverage of rules for that were designed may have left gaps, with other complex and demanding scenarios not taken into proper consideration. Proper rule evaluation for alerting may require thorough testing for longer periods of time with large data samples.
- **Historical Data Analysis** - The data volumes differ when historical data comes into account, as the logging mechanisms changed over the course of Waldur's development, thus resulting in old data having less contextual information in those events. This may require additional analysis of existing data and the development of adjustments to the alerting rules and data enrichment processes.
- **Unexplored Functionalities** - OpenSearch comes with multiple functionalities that have not been thoroughly tested such as anomaly detection and multi-source event correlation.
- **Event Gaps Analysis** - A thorough analysis of currently generated events and the gaps in generation may be necessary. This would require looking over all of the currently generated events and available event types, and analyzing the contents of the events for fields types, context and other valuable information. While we have conducted a brief overview of Waldur's events, a thorough gap analysis for all event types may be necessary as well as compliance review for logging standards, for enhanced and accurate detection.

- **Scalability** - Scalability options have not been thoroughly explored as the proof of concept was conducted within a more limited environment. Future work may involve testing the scalability of the SIEM functionality under different work loads and multiple data sources.
- **Comparative Analysis of Different SIEM Solutions** - Due to time constraints, only one SIEM solution has been given a practical test implementation. There are other tools and solutions that may potentially perform better than OpenSearch and its security analytics in different areas. One such solution that fit Waldur's criteria was identified to be Wazuh. A future work could potentially involve a test implementation of Wazuh, and a thorough overview and comparison between the two SIEMs for Waldur's needs could be conducted.
- **Knowledge and Expertise Gaps** - Gaps in skill and technical expertise with SIEM solutions and event correlation software may have affected the effectiveness of the implementation, as more experienced specialists may have defined better rules or configuration. Additional work may involve analysis of the approach of data enrichment with OpenSearch queries, their adjustments and modification.
- **Addressing the HTTP Filter Data Enrichment Delay** - the current workaround of introducing a short delay for an event that uses HTTP filter for data enrichment must be resolved, as it reduces performance and is inefficient for processing many thousands of events, requiring future work to address this issue.

With the above considerations, future work may be conducted using the current results as a foundation of SIEM implementation for Waldur.

References

[1] Buecker, A.; Amado, J.; Druker, D; Lorenz, C.; Muehlenbrock, F. & Tan, R. IT Security Compliance Management Design Guide with IBM Tivoli Security Information and Event Manager. *IBM Redbooks,* 2010.

[2] Reliability Requirements for Advanced Security Event Management. *Computer Network Security*. https://link.springer.com/chapter/10.1007/978-3-642-33704-8_15

[3] A. Gillis. Security Information and Event Management (SIEM). Techradar, 2022. https://www.techtarget.com/searchsecurity/definition/security-information-and-event-management-SIEM

[4] ETAIS. https://etais.ee/

[5] Puhuri. https://puhuri.io/

[6] Waldur. https://waldur.com/

[7] Waldur Documentation. https://docs.waldur.com/

[8] Õun J. "Testing Estonian Scientific Computing Infrastructure Self-Service in Cypress Framework". 2022. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=75146

[9] Waldur Events. https://docs.waldur.com/developer-guide/events/

[10] PostgreSQL database. https://www.postgresql.org/

[11] Waldur API. https://docs.waldur.com/API/reporting-api/

[12] Kotenko I.; Gaifulina D. & Zelichenok I. "Systematic Literature Review of Security Event Correlation Methods". *IEEE Access*, *vol. 10,* 2022, pp. 43387-43420, doi: 10.1109/ACCESS.2022.3168976

[13] Pordzins O.; Romanovs A. Why SIEM is Irreplaceable in a Secure IT Environment? *Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 2019, Vilnius, Lithuania, pp. 1-5. https://ieeexplore.ieee.org/document/8732173

[14] González-Granadillo G.; González-Zarzosa S.; Diaz R. Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures. *Sensors,* 2021. https://doi.org/10.3390/s21144759

[15] EU Commission. Commission signs agreement with industry on cybersecurity and steps up efforts to tackle cyber-threats. Brussels, 2016.

[16] Suarez-Tangil G.; Palomar E.; Ribagorda A. & Sanz, I. Providing SIEM systems with self-adaptation, *Information Fusion, Volume 21,* 2015, pp. 145-158. https://doi.org/10.1016/j.inffus.2013.04.009.

[17] Gillis S. A. Security Information And Event Management. *Techtarget, 2022.* https://www.techtarget.com/searchsecurity/definition/security-information-and-event-management-SIEM

[18] Nabil M.; Soukainat S.; Lakbabi A. & Ghizlane O. SIEM selection criteria for an efficient contextual security. *International Symposium on Networks, Computers and Communications (ISNCC)*, 2017, Morocco, pp. 1-6. doi:10.1109/ISNCC.2017.8072035.

[19] Lee J.; Kim Y. S.; Kim J. H. and Kim I. K. Toward the SIEM architecture for cloud-based security services. *IEEE Conference on Communications and Network Security (CNS),* 2017, pp. 398-399.

[20] Sadowski G.; Kavanagh K. & Bussa T. Critical Capabilities for Security Information and Event Management. *Gartner,* 2020, ID G00381141.

[21] Butler J. M. Benchmarking Security Information Event Management (SIEM). *SANS, 2009.*

[22] Jawad Manzoor J.; Waleed A.; Fareed A. J. & Masood A. "Cybersecurity on a budget: Evaluating security and performance of open-source SIEM solutions for SMEs". *PLOS ONE,* 2024, doi: 10.1371/journal.pone.0301183

[23] Talha A. A. "Unveiling the Top 8 Open Source SIEM Tools of 2024 for Unrivaled Cyber Defense". *Systems Weakness,* 2024. https://systemweakness.com/empower-your-security-unveiling-the-top-8-open-source-siem-tools-of-2024-for-unrivaled-cyber-d124088ede3f

[24] Top Five Free and Open Source SIEM. https://utmstack.com/top-five-free-and-open-source-siem

[25] Top Open Source SIEM Tools. 2023. https://www.atatus.com/blog/top-open-source-siem-tools/

[26] Wazuh. https://documentation.wazuh.com/

[27] OSSEC. https://www.ossec.net/

[28] AlienVault OSSIM. https://cybersecurity.att.com/products/ossim

[29] SIEMonster. https://siemonster.com/

[30] ElasticSearch. https://www.elastic.co/

[31] ElasticSearch. *DB-Engines.* https://db-engines.com/en/system/Elasticsearch

[32] OpenSearch Security Analytics. https://opensearch.org/docs/latest/security-analytics/

[33] Bolla A. & Talentino F. Threat Hunting driven by Cyber Threat Intelligence. 2022. *Politecnico di Torino, Master's Thesis*. https://webthesis.biblio.polito.it/22631/

[34] OpenSearch Dashboards. https://opensearch.org/docs/latest/dashboards/

[35] Elastic SSPL License. https://www.elastic.co/pricing/faq/licensing#sspl

[36] Logstash. https://www.elastic.co/logstash

[37] OWASP Logging CheatSheet. https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html#which-events-to-log

[38] Security Logging and Monitoring Standard. 2023. *Office of the Chief Information Officer, State of Minnesota.* https://mn.gov/mnit/assets/Security%20Logging%20and%20Monitoring%20Standard_tcm38-323794.pdf

[39] Logging Best Practices That Can Improve Your Cybersecurity Game. 2024. *Exabeam*. https://www.exabeam.com/blog/security-operations-center/logging-best-practices-that-can-improve-your-cybersecurity-game

[40] Helm Package Manager. https://helm.sh/

[41] Kubernetes. https://kubernetes.io/

[42] Minikube. https://github.com/kubernetes/minikube.

[43] Docker. https://www.docker.com/

[44] Caddy. https://caddyserver.com/docs/

[45] Sigma Rules. https://github.com/SigmaHQ/sigma

[46] Slack. https://slack.com/

[47] ElasticSearch filter plugin. https://www.elastic.co/guide/en/logstash/current/plugins-filters-elasticsearch.html

[48] OpenSearch filter plugin GitHub issue. https://github.com/jgough/logstash-filter-opensearch/issues/2

[49] Elasticsearch filter plugin GitHub discussion. https://github.com/opensearch-project/opensearch-clients/issues/4

[50] Slack API. https://api.slack.com/messaging/webhooks

Appendix

I. GitHub Configuration Files

https://github.com/bugblasterX/Waldur-OpenSearch-Config

II. License

Non-exclusive licence to reproduce thesis and make thesis public.

I, **Mark Borissov**,

(*author's name*)

1. grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

Deploying Open-Source SIEM system for Waldur-based services at the University of Tartu,

(title of thesis)

supervised by Risto Vaarandi, Ilja Livenson.

(supervisor's name)

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Mark Borissov,

Tartu, **14.08.2024**