Interpretable Next-Generation SIEM

ZAYAN TAHIR
01-134221-091
UMAR KHAN
01-134221-083

Supervisor: Dr. Arshad Farhad

Bachelor of Science in Computer Science

Department of Computer Science
Bahria University, Islamabad

December 2025

Certificate

We accept the work contained in the report titled "Interpretable Next-Generation Security Information and Event Management (SIEM)", written by Zayan Tahir and Umar Khan as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor: Dr. Arshad Farhad , Assistant Professor

_____

Internal Examiner:

_____

External Examiner:

_____

Project Coordinator: Dr. Asim Ali , Assistant Professor

_____

Head of the Department: Dr. Khurrram Ehsan , Senior Associate Professor

_____

Abstract

Traditional Security Information and Event Management (SIEM) systems are characterized by the overload of alerts, high levels of false-positives, low susceptibility to detecting advanced persistent threats (APTs), and slow incident response. Such limitations lower the efficiency of real-time security monitoring in the present-day infrastructures. This paper introduces a next-generation SIEM, which can be interpreted and is automated to detect threats, respond to them, and improve the overall security of the system using artificial intelligence. The suggested platform will generalize Wazuh-based log collection with machine-learning models, LightGBM and the Bi-LSTM model. The techniques used to achieve explainable AI include LIME and SHAP, which can be used to produce interpretable threat scores, allowing an analyst to know the reasoning behind any given alert. Threat intelligence feeds are included, and well-known attack signatures are identified rapidly, whereas the unknown threats are discussed with the help of the AI models. There is a scalable log indexing and real-time visualization supported by a modular backend architecture based on Elasticsearch. The experimental findings show better detection rates, less workload on the analysts and greater responsiveness to APTs. The system proves the significant shortcomings of the traditional SIEM platforms and delivers high detection as well as interpretability in the AI-based decisions. This Interpretable Next-generation SIEM is to revolutionize the security operations by solving the most important issues of the traditional version of the system: absence of interpretability, workload, and APTs. It provides not only precise detection but also creates trust in AI-driven decisions, thus allowing security teams to decide confidently in dynamically changing threat environments.

Acknowledgment

It is with the grace of Allah Almighty that we have been able to capitalize on the opportunity to create this project. His guidance gave us the confidence we needed to complete this service successfully.

We are thankful to our supervisor, Dr. Arshad Farhad, for his unconditional support and encouragement throughout this project. He believed in our capabilities which has been instrumental in pushing us to deliver our best work. Working under the guidance of a true mentor has been an honor, and we are grateful for the valuable mentorship that he provides.

Last but not least, we express our heartfelt appreciation to our parents for their prayers and unwavering support. Their blessings and encouragement have been a source of strength, propelling us forward in this project and beyond.

Zayan Tahir
Umer Khan
Bahria University Islamabad, Pakistan.

December, 2025

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AI	Artificial Intelligence
APT	Advanced Persistent Threat
CCTV	Closed-Circuit Television
GDPR	General Data Protection Regulation
GUI	Graphical User Interface
HIPAA	Health Insurance Portability and Accountability Act
IDS	Intrusion Detection System
IOC	Indicator of Compromise
IPS	Intrusion Prevention System
Kafka	Apache Kafka (Distributed Event Streaming Platform)
LIME	Local Interpretable Model-agnostic Explanations
LSTM	Long Short-Term Memory
ML	Machine Learning
MISP	Malware Information Sharing Platform
PyTorch	Python-based Deep Learning Framework
SHAP	SHapley Additive exPlanations
SIEM	Security Information and Event Management
SOAR	Security Orchestration, Automation, and Response
TCP/IP	Transmission Control Protocol / Internet Protocol
UI	User Interface
URL	Uniform Resource Locator
VM	Virtual Machine
Wazuh	Open Source Security Platform

Chapter 1

Introduction

Cybersecurity threats are increasing in scale and affect, making it difficult for organizations to defend against emerging attacks and cyber-criminals. Traditional Security Information and Event Management (SIEM) systems which tend to perform basic log analysis and rule-based detection, often fail in identifying unknown threats and adapting to evolving strategies for cyber attacks. This creates a critical gap in active threat mitigations, particularly in large-scale, distributed environment.

The Interpretable Next generation SIEM system presented in this project aims to address these limitations by integrating real-time monitoring with intelligent AI and ML models. The system is built on Wazuh framework, the system collects logs from various endpoints and evaluates the logs collected using the models also performs explainability by LIME or SHAP. In some cases, when the threat activity does not match one of the activities that have already been recognized and mitigated in the past, system allows AI to overtake and flag the activity with the befitting severity thus allowing for the automated mitigation action.

The analysts are provided with a vast data to analyze in the custom dashboard to have a check on the alerts and interpretabilities of the methods such as LIME or SHAP. This will help to reduce the alert fatigue and enhance confidence in the automated mitigation actions. This project provides a balance between rule-based detection and machine learning and explainability to offer a scalable and adaptive defense approach that proves to improve the visibility and trust in cybersecurity operations. This is the method that makes it interpretable and proactive to provide the protection and security to the organizations.

1.1 Problem Description

Conventional Security Information and Event Management (SIEM) tend to have natural limitations that hinder their effectiveness in the modern cybersecurity environments. The system of today generates unrelenting amount of alerts that leads to the phenomenon

known as alert fatigue. This does not only therefore exhaust human resources but also creates the possibility of overlooking the real threats.

The other limitation is that they are based on existing threat detection rules. Although it works with familiar threats/threat patterns , this method is not suitable in situations where it is necessary to deal with advanced persistent threats (APTs) that are not restricted by the established patterns. Resultantly, conventional SIEMs become both false positive and false negative in nature hence decreasing the overall level of accuracy and reliability of the system.

Moreover, the responses of these legacy systems tend to be slow or manual and not automated. This type of latency stalls the real time mitigation of threat activities which is particularly harmful in IT infrastructures at a large. These limitations together bring out the need to have a smart, adaptable, and automated security system that can respond to the advanced threat activities of the newer generation.

1.2 Objectives

To develop an open source AI powered Security Information and Event Management system is the main goal of this project. The platform offers threat detection features, minimized false positives and automated response mechanisms via machine learning, anomaly detection and real time processing. The particular, specific goals are as follows:

1. **To develop an open-source AI-driven SIEM framework:** Develop a modular SIEM architecture based on open-source technology that can integrate machine learning models and advanced analytics in order to identify unknown and advanced threats such as zero-day attacks and Advanced Persistent Threats (APTs) [1].

2. **To implement interpretable AI models for anomaly detection:** Develop and learn machine learning models that can be used to analyze security-related data such as network traffic, system log, and user behavior to identify anomaly which could pose as a zero-day vulnerability or an evasive threat. The focus will be made on transparency with the help of interpretable and explainable AI techniques [2].

3. **To enable real-time threat detection and automated response:** Introduce real-time processing and automation capabilities, which will be able to immediately respond to detected threats by taking pre-defined mitigation measures, including isolating compromised machines or malicious IP addresses, eliminating the necessity of human intervention [3][4].

4. **To reduce false positives and alert fatigue:** Establish filtering and prioritization mechanisms in the system to reduce the rate of false positives so that only high

confidence alerts will reach the security analysts and eventually improve the level of productivity and efficiency in responding to the threats.

1.3 Project Scope

This Interpretable Next-Generation SIEM project aims to improve cybersecurity capabilities by combining real-time threat intelligence correlation, automated threat response systems, and AI-based anomaly detection. The goals of this project are to improve monitoring precision, reduce alert fatigue, and offer scalable security solutions appropriate for dynamic IT environments.

Included in Scope

1.3.1 AI-Driven Anomaly Detection:

To create and add machine learning models that recognize complex threats like Advanced Persistent Threats (APTs). For log analysis and behavior-based threat identification, the system uses AI and Machine Learning models such as light-GBM and bi-LSTM models.

1.3.2 Automated Threat Response:

To automate mitigation action by integrating Security Orchestration, Automation, and Response (SOAR) functionalities. Therefore carrying out actions like IP blocking, incident action blocking, system isolation in response to detected threats and logging the detected intruder out of the system if somehow infiltrated, these are the actions that are needed.

1.3.3 Threat Intelligence Integration:

The system has been integrated with threat intelligence platforms like OpenCTI/MISP for better correlation between external threat data and internal logs and alerts.

1.3.4 Scalability and Performance Optimization:

To optimize resources and allow large data logs ingestion and processing to ensure that the system runs efficiently in large scale IT environments.

1.3.5 Compliance and Reporting:

To ensure that the system is able to help in generation of detailed reports to help with audit preparation and maintain transparency within the organization.

Excluded from Scope

1.3.6 Physical Security Measures:

Physical measures like biometric authentication systems or CCTV surveillance will not be addressed in this project.

1.3.7 End-to-End Enterprise Security Solutions:

The platform is designed to complement, not replace, existing security solutions like endpoint protection, firewall management, and anti-malware software.

1.3.8 Industry-Specific Customization:

The scope is focused on general-purpose cybersecurity applications and does not cover specialized features for specific industries like healthcare, finance, or critical infrastructure.

Assumptions

1. The system itself is expected to be deployed in distributed IT environments that contain cloud-based and on premise infrastructure.
2. Organizations implementing the system are presupposed to have basic security measures such as firewalls and endpoint protection systems in place.
3. This project has thus far been utilizing the Bridged Adapter Network Configuration which is in use and has the fixed stability of the Ethernet cabled connection of both server and agents.
4. The assumption is that there will be access to real-time logs and external threat intelligence capable of being used in the system.
5. The AI models incorporated among the system will require continuous learning and training to suit changing threat environments.
6. Connection between the server and the endpoints both to the internet should be within the same LAN.

1.4 Solution Application Areas

This Interpretable Next-Generation SIEM project is developed to manage the important cybersecurity problems or issues in organizations. It is best suitable when used in different fields where detecting possible threats and responding to them in a minute is essential with its AI capabilities. The key areas where such solution can be deployed are:

1.4.1 Enterprise IT Environments:

Large organizations that have large networks and IT infrastructures can benefit from real-time monitoring and the automated responses to threats. The system helps in safeguarding corporate data, intellectual property, and internal communications.

1.4.2 Cloud Infrastructure Monitoring:

The system may be implemented to track cloud-native services and applications. The growing trend towards cloud services has caused virtual networks. It also supports hybrid environments and provides a steady visibility and threat detection across on-premise and cloud environments.

1.4.3 Managed Security Service Providers (MSSPs):

The platform can help MSSPs to provide smart security surveillance to multiple clients. This system has the multi agent capability that facilitates the delivery of services, management and alerts of individual clients.

1.4.4 Government and Public Sector Networks:

This system can also be used by the public sector agencies as a national level cybersecurity, compliance auditing and incident responses. Analytics based on AI is highly appropriate to track vital services and prevent internet espionage and cyber attacks.

1.4.5 Critical Infrastructure Protection:

The system can be implemented in the monitoring and security of critical infrastructure like transportation networks, water treatment and energy grids. It enhances the dependency of key public services by identifying and responding to aberrations in operational technologies (OT).

1.4.6 Healthcare IT Security:

This system can be used to protect data in hospitals and healthcare providers so that the electronic health records (EHRs) and medical devices meet policies such as HIPAA and reduce the dangers of ransomwares and data leaks.

1.4.7 Financial Services and FinTech:

Cybersecurity is needed by financial institutions to secure customer information and transactional records. The resolution offers fraud identification, insider threat detection and regulatory compliance support of the sectors of strict data security requirements.

1.4.8 Education and Research Institutions:

This system can be used to protect sensitive information in universities and research organizations, security data, student information, and also identify unauthorized access while providing services and endorsement of adherence to cybersecurity policies on education.

Chapter 2

Literature Review

This section investigates the under-research, market trends, and technological ad-research. It is required to design an explainable next-gen SIEM. The goal is to realize the capabilities and weakness of prevailing security monitoring solutions. The evolving threat environment, and the role of artificial intelligence in the contemporary. cybersecurity frameworks. In order to create the AI driven SIEM platform, there is a need to examine the available industry practices and indicate the gaps that reduce the effective threat detection and response. By this review, we have been able to place our project within the domain of intelligent security monitoring and matching our goals with the requirements of the industry.

2.1 Introduction

The traditional SIEM systems have turned out to be comparatively inept at guaranteeing reacquired defenses with the increase of complexity and size of cyber threats. SIEM environments are generally rule based and that is, their protection coverage is determined by rules to identify threats, which in effect limits their capability to recognize high level attack patterns like zero-day, weaknesses or Advanced Persistent Threats (APTs). This type of system also tends to produce a large number of low-confidence alerts that lead to alert exhaustion, reduced speed of response, and increased risk of managing operations.

Additionally, the growing role of artificial intelligence and machine learning offers tremendous opportunity for SIEM systems by providing real-time anomaly detection and intelligent event correlation that also feeds automation to combat incidents. However, a lack of interpretability is still a barrier for many to use even AI-driven systems to run critical cybersecurity operations. Security teams often require more transparency to feel confident in automated decisions, making interpretability an equally important construct in audience acceptance of a model.

This literature review seeks out existing SIEM platforms, AI or machine learning integrated cybersecurity systems, and then evaluates explainable AI (XAI) methodologies as applied to threat detection. It establishes rational support for a proposed solution by demonstrating the need for automation, accuracy, and interpretability all on a single security framework.

2.2 Comparative Analysis

Table 2.1 presents a comparative analysis of various traditional and modern SIEM systems, evaluating key features relevant to AI integration, real-time response, and model interpretability.

Table 2.1: Comparison of Existing SIEM Platforms and the Proposed System

SIEM Platform	Open Source	AI Integration	Real-Time Detection	Automated Response	XAI Support	Custom Rules	Threat Intelligence
Splunk Enterprise Security	No	✓	✓	✓		✓	✓
IBM QRadar	No	✓	✓	✓		✓	✓
Elastic SIEM	Yes	Limited	✓			✓	✓
OSSIM (AlienVault)	Yes		✓			✓	✓
Sagan	Yes		Limited			✓	
Proposed SIEM System	Yes	✓	✓	✓	✓	✓	✓

2.3 Summary

The examination of contemporary SIEM systems identifies serious limitations in their adaption to modern cybersecurity threats and impacts of adoption. Most platforms focus heavily on rule-based detection and require significant manual setup. Some existing commercial solutions feature an AI component that is transparent (e.g., an anomaly-based approach) yet still fall short with explainable AI (XAI) for analysts ensuring confidence in their decisions.

Further, open-source SIEM platforms provide flexibility and cost-effectiveness while often not having advanced AI-driven features for detecting threats and resource automation in real-time. This disconnect between adaptability and AI-driven capacity, explainability and real-time and response to cyber threats indicates a need for a system to be generating the new direction for SIEM systems.

As such, our Next-Generation Interpretable SIEM directly addresses this gap by ensuring machine learning (ML) classification-based models will be used for anomalous detection, as well as SOAR for real-time threat mitigation, and embedded explainable AI (XAI) techniques to provide transparency to our decisions. This ensures analysts improve more accurately in cyber threat detection and hypothesis generation, thus increasing trust and usability of a SIEM system analyst, as well as provide a holistic solution to securing IT infrastructures in your organisation.

Chapter 3

Requirement Specifications

3.1 Existing System

The current security monitoring landscape is largely founded on traditional Security Information and Event Management (SIEM) systems, which are built on the detection features of a fixed and rule-driven system. Even though these systems have been extensively implemented in organizations, they are unable to keep pace with the growing expansiveness, intricacy, and intricacy of current cyber threats. The usefulness of traditional SIEM solutions is diminishing as the enterprise infrastructures grow more dynamic and the attack techniques are increasingly sophisticated.

3.1.1 Limitations of Traditional SIEMs

Traditional SIEM systems primarily rely on set rules and known attack signatures to identify security incidents. This would have a critical impact on their capability to detect the zero-day attacks, including the use of the Log4j vulnerability, where no previous signatures exist. The other major problem is the issue of alerts fatigue because the Security Operations Center (SOC) teams usually receive approximately over 10,000 alerts daily, and nearly 70 percent of them are false positives which are a burden to the analysts causing a slowness in operations. Incident investigation and response is also highly manual making the mean time to respond (MTTR) span over four hours since very little is automated in the triage and remediation procedures. In addition, the conventional SIEM systems do not have the proper connectivity to the third party threat intelligence frameworks like MISP, and therefore cannot correlate internal security incidents with global threat data and enhances threat visibility.

3.2 Proposed System

The proposed system introduces an interpretable AI-based Security Information and Event Management (SIEM) solution designed to overcome the limitations of traditional rule-based platforms.

3.2.1 Architecture Overview

This diagram illustrates the system's components. (Frontend, Backend, Data layer and The ML Pipeline.)

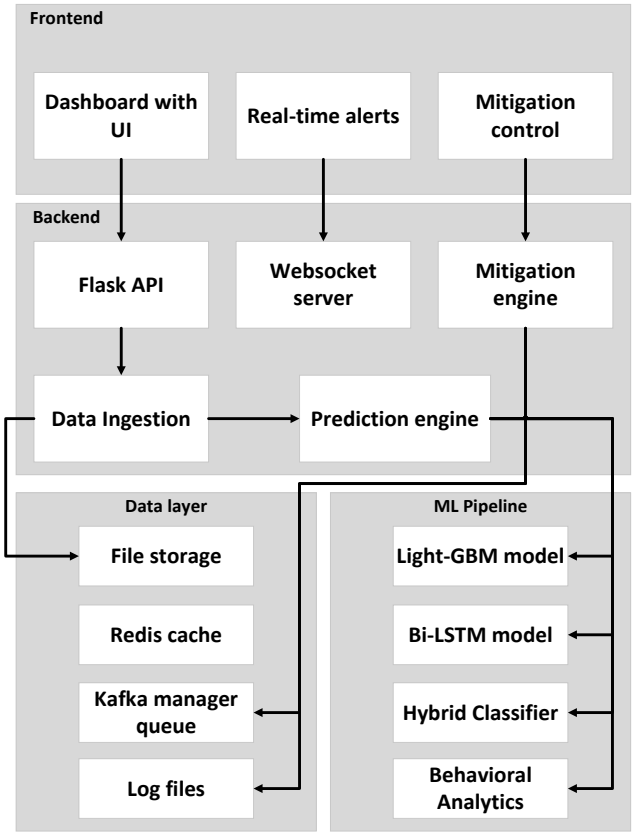

Figure 3.1: System Components Overview

3.2.2 Key Innovations

1. **AI-Powered Detection**: light-GBM and bi-LSTM models for anomaly detection.
2. **Automated SOAR**: Response actions triggered within 30s.
3. **Interpretability**: LIME/SHAP values to explain AI/ML model decisions to analysts[1].
4. **Scalability**: Kafka-based log ingestion at 10,000+ EPS (events per second).

3.3 System Requirements

3.3.1 Functional Requirements

Table 3.1: Functional Requirements

S. No.	**Functional Requirement**	**Type**	**Status**
1	Collect logs from Wazuh agents	Core	Completed
2	Parse and store logs via Kafka & Elasticsearch	Core	Completed
3	Match logs against known attack signatures (MISP/OpenCTI)	Core	Completed
4	Apply ML anomaly detection (bi-LSTM/lightGBM)	Core	Completed
5	Generate interpretable reports with AI rationale	Core	Completed
6	Perform automated mitigation via SOAR	Advanced	Completed
7	Display incidents on Kibana/Custom dashboards	Intermediate	Completed

3.3.2 Non-Functional Requirements

Table 3.2: Non-Functional Requirements

S. No.	**Non-Functional Requirement**	**Category**
1	Real-time processing of high-volume logs	Performance
2	Secure transport with TLS	Security
3	Support for multi-cloud and hybrid environments	Scalability
4	Alerting and response time < 5 seconds	Availability
5	Explainability of AI models	Usability

3.4 Use Cases

Figure 3.2: Use Case Diagram

The Use Case diagram illustrates all the use cases of all the Actors. (Data Source, Security Analyst and the Server.)

Table 3.3: Use Case: Point to server IP

<table>
<tr><td>USE CASE NAME</td><td colspan="2">Point to server IP</td></tr>
<tr><td>USE CASE ID</td><td colspan="2">SIEM-01</td></tr>
<tr><td>PRIORITY</td><td colspan="2">High</td></tr>
<tr><td>PRIMARY ACTOR</td><td colspan="2">Data source</td></tr>
<tr><td>DESCRIPTION</td><td colspan="2">Allows the data source to send logs to the system.</td></tr>
<tr><td>BASIC FLOW</td><td colspan="2">1. User enters IP in var/ossec/etc/ossec.conf file in wazuh agent.
2. System collects and stores logs.
3. Upon success, logs start appearing on dashboard for further evaluation.</td></tr>
<tr><td>ALTERNATE FLOW</td><td colspan="2">1. Pointing the agent to invalid IP shows no logs on the server.</td></tr>
<tr><td>PRE-CONDITION</td><td colspan="2">User must be connected to the internet from the same LAN.</td></tr>
<tr><td rowspan="3">COURSE EVENT</td><td>ACTOR ACTIONS</td><td>SYSTEM RESPONSE</td></tr>
<tr><td>Enter IP of server.</td><td>Validate IP.</td></tr>
<tr><td>Allow the logs to be shown on the wazuh dashboard.</td><td>System starts receiving wazuh agent logs.</td></tr>
<tr><td>POST-CONDITION</td><td colspan="2">Wazuh agent is granted connection to the SIEM system and start getting displayed to the main dashboard.</td></tr>
</table>

Table 3.4: Use-Case: Inject Real Time data Logs

<table>
<tr><td>USE CASE NAME</td><td colspan="2">Inject Real Time data Logs</td></tr>
<tr><td>USE CASE ID</td><td colspan="2">SIEM-02</td></tr>
<tr><td>PRIORITY</td><td colspan="2">High</td></tr>
<tr><td>PRIMARY ACTOR</td><td colspan="2">Data Source</td></tr>
<tr><td>DESCRIPTION</td><td colspan="2">This use case describes how Wazuh agents (data source) sends its activity logs to the server. The agent's logs are indexed in the wazuh indexer and then displayed on the wazuh dashboard.</td></tr>
<tr><td>BASIC FLOW</td><td colspan="2">1. Wazuh Agent sends logs from endpoint systems.
2. Agent formats and packages the logs.
3. Logs are sent to the Wazuh Manager.
4. Wazuh Manager forwards logs to Kafka.
5. Logs are ingected into Elasticsearch for analysis.</td></tr>
<tr><td>ALTERNATE FLOW</td><td colspan="2">1. Network failure: queues logs locally.
2. Format error: Logs sent in any format other than json gets tricky to parse.
3. Manager unavailable: Logs cached until connection restores.</td></tr>
<tr><td>PRE-CONDITION</td><td colspan="2">Wazuh Agent is active and connected to the wazuh server; Manager and pipeline (Kafka/Elasticsearch) are operational.</td></tr>
<tr><td rowspan="4">COURSE EVENT</td><td>ACTOR ACTIONS</td><td>SYSTEM RESPONSE</td></tr>
<tr><td>Package logs.</td><td>Prepare for transmission.</td></tr>
<tr><td>Send to Wazuh Manager.</td><td>Display acknowledgement of sent logs on dashboard.</td></tr>
<tr><td>Forward to Kafka & Elasticsearch.</td><td>Logs are queued, processed and stored.</td></tr>
<tr><td>POST-CONDITION</td><td colspan="2">Logs are successfully received, cached, and made available for analysis and visualization.</td></tr>
</table>

Table 3.5: Use-Case: Connector settings and configuration

<table>
<tr><td>USE CASE NAME</td><td colspan="2">Connector settings and configuration</td></tr>
<tr><td>USE CASE ID</td><td colspan="2">SIEM-03</td></tr>
<tr><td>PRIORITY</td><td colspan="2">High</td></tr>
<tr><td>PRIMARY ACTOR</td><td colspan="2">Security Analyst</td></tr>
<tr><td>DESCRIPTION</td><td colspan="2">This use case allows security analyst to setup the connection of wazuh server and check whether the agent is active and connected and sending the logs to correct destination.</td></tr>
<tr><td>BASIC FLOW</td><td colspan="2">1. Wazuh manager allows the analyst to see which agents are connceted and active.
2. Analyst filters, searches, and explores alerts by host, severity, or timestamp.</td></tr>
<tr><td>ALTERNATE FLOW</td><td colspan="2">1. Dashboard load failure: fallback to textual log output.
2. No alerts generated: display idle or heartbeat status.
3. Elasticsearch delay: events displayed with a slight lag.</td></tr>
<tr><td>PRE-CONDITION</td><td colspan="2">System must be connected to a stable internet connection.</td></tr>
<tr><td rowspan="4">COURSE EVENT</td><td>ACTOR ACTIONS</td><td>SYSTEM RESPONSE</td></tr>
<tr><td>Open wazuh manager’s agent control list.</td><td>Display active agents.</td></tr>
<tr><td>Apply filters to view logs.</td><td>Dashboard refreshes and visualizes relevant data.</td></tr>
<tr><td>Click on an alert.</td><td>Show detailed metadata and threat context.</td></tr>
<tr><td>POST-CONDITION</td><td colspan="2">Analyst successfully lists the active agents and views, filters, and investigates live security events and alerts in real time.</td></tr>
</table>

Table 3.6: Use-Case: Perform manual playbook executions

<table>
<tr><td>USE CASE NAME</td><td colspan="2">Perform manual playbook executions</td></tr>
<tr><td>USE CASE ID</td><td colspan="2">SIEM-04</td></tr>
<tr><td>PRIORITY</td><td colspan="2">Medium</td></tr>
<tr><td>PRIMARY ACTOR</td><td colspan="2">Security Analyst</td></tr>
<tr><td>DESCRIPTION</td><td colspan="2">This use case allows the analyst to perform the playbook executions manually if analyst wants to apply mitigation.</td></tr>
<tr><td>BASIC FLOW</td><td colspan="2">1. Analyst accesses the custom dashboard.
2. Redirects to the SOAR mitigation page.
3. Select any mitigation and provides the IP of the agent.
4. Mitigation is applied on the provided IP of the agent.</td></tr>
<tr><td>ALTERNATE FLOW</td><td colspan="2">1. Provided IP of the agent is unavailable.
2. Agent to be mitigated is disconnected from the wazuh environment.
3. Incorrect IP for any agent.</td></tr>
<tr><td>PRE-CONDITION</td><td colspan="2">Logs and alerts are already stored in Elasticsearch; the agent is connected and mitigation is not already applied and internet connection fro both agent and server is stable.</td></tr>
<tr><td rowspan="3">COURSE EVENT</td><td>ACTOR ACTIONS</td><td>SYSTEM RESPONSE</td></tr>
<tr><td>Open SOAR interface.</td><td>Load available mitigation actions.</td></tr>
<tr><td>Select a mitigation and fill the json data of the agent to be mitigated.</td><td>Mitigation is applied on the agent.</td></tr>
<tr><td>POST-CONDITION</td><td colspan="2">Manual SOAR mitigation is applied to the agent succesfully.</td></tr>
</table>

Table 3.7: Use-Case: Evaluate Injeced Logs

USE CASE NAME	Evaluate Injected Logs	
USE CASE ID	SIEM-05	
PRIORITY	Medium	
PRIMARY ACTOR	Server	
DESCRIPTION	This use case enables the server to evaluate the logs sent to it by using the light-GBM and bi-LSTM models. This use case extent to the LABEL LOGS SEVERITY that is done by the AI/ML models and the results of those two models is averaged Hybrid model. which further extends to PERFORM EXPLAINABILITY which uses the LIME/SHAP models that deduce which vector from among the 10D vectors were responsible for the label each log gets.	
BASIC FLOW	1. Server receives the logs and sends them to the models for their prediction scores. 2. Prediction scores of bi-LSTM and light-GBM model are averaged. 3. Based on the final prediction scores the AI labels the log's severity.	
ALTERNATE FLOW	1. Dashboard fails to load: display error message. 2. AI/ML models labels the log later than it should.	
PRE-CONDITION	The system must have event data indexed in Elasticsearch; the server must have a steady injestion of wazuh agent logs.	
COURSE EVENT	**ACTOR ACTIONS**	**SYSTEM RESPONSE**
	Server sends the logs to AI/ML models.	Prediction scores generated.
	Hybrid model averages the two prediction scores.	Displays the label on the dashboard in the alerts tab.
POST-CONDITION	The wazuh agent log is labeled with its befitting severity and clicking on the explain button populates the graph with the prediction scores and the contributing factors for the concerned log's label.	

Table 3.8: Use-Case: Perform automated mitigation

<table>
<tr><td>USE CASE NAME</td><td colspan="2">Perform automated mitigation</td></tr>
<tr><td>USE CASE ID</td><td colspan="2">SIEM-06</td></tr>
<tr><td>PRIORITY</td><td colspan="2">High</td></tr>
<tr><td>PRIMARY ACTOR</td><td colspan="2">Server</td></tr>
<tr><td>DESCRIPTION</td><td colspan="2">Server automatically mitigates mostly the high or critical labelled wazuh logs.</td></tr>
<tr><td>BASIC FLOW</td><td colspan="2">1. Server gets alerted that AI/ML models labelled an alert as high or critical.
2. It performs SOAR mitigation action/s based on the type of activity of the agent.
3. Dashboards displays the playbook mitigation that is either in execution or was completed successfully.</td></tr>
<tr><td>ALTERNATE FLOW</td><td colspan="2">1. No data for the agent is found.
2. Agent disconnected before the execution of mitigation.
3. Slow mitigation response: show results with a loading indicator until completed.</td></tr>
<tr><td>PRE-CONDITION</td><td colspan="2">Wazuh agent logs are already labelled with befitting severity by the AI/ML models and internet connection is stable.</td></tr>
<tr><td rowspan="3">COURSE EVENT</td><td>ACTOR ACTIONS</td><td>SYSTEM RESPONSE</td></tr>
<tr><td>A log is labelled as High or Critical.</td><td>Selects the mitigation action/s to be performed.</td></tr>
<tr><td>Gets the relevant information of the agent (VM name or IP).</td><td>Applies the selected mitigation action/s on that agent.</td></tr>
<tr><td>POST-CONDITION</td><td colspan="2">Server successfully performs the automated mitigation of the high or critical severity alert just as it appears on the dashboard.</td></tr>
</table>

Chapter 4

Design

Systems design consists of specifying the architecture, components, modules, interfaces, and data for a system to satisfy stated requirements. This chapter provides an extensive design of the Interpretable Next-Generation SIEM system, including the entire design and its structure, constraints, methodology, and specifications at the component level.

4.1 System Architecture

This section summarizes the high-level functional and physical structure of the Next-Generation SIEM system. It is designed to accommodate the scale and complexity of modern IT environments with open-source tools and artificial intelligence for robust threat detection and automated response capabilities. This system uses a comprehensive, fault-tolerant architecture that includes log collection, real-time streaming, machine learning-based analysis, threat intelligence, mitigation and explaination of the detected alerts severity. The goal is to migrate the System from rule based detection to AI powered anomaly detection, capturing zero-day attacks and Advanced Persistent Threats (APTs) that bypass current operational methods.

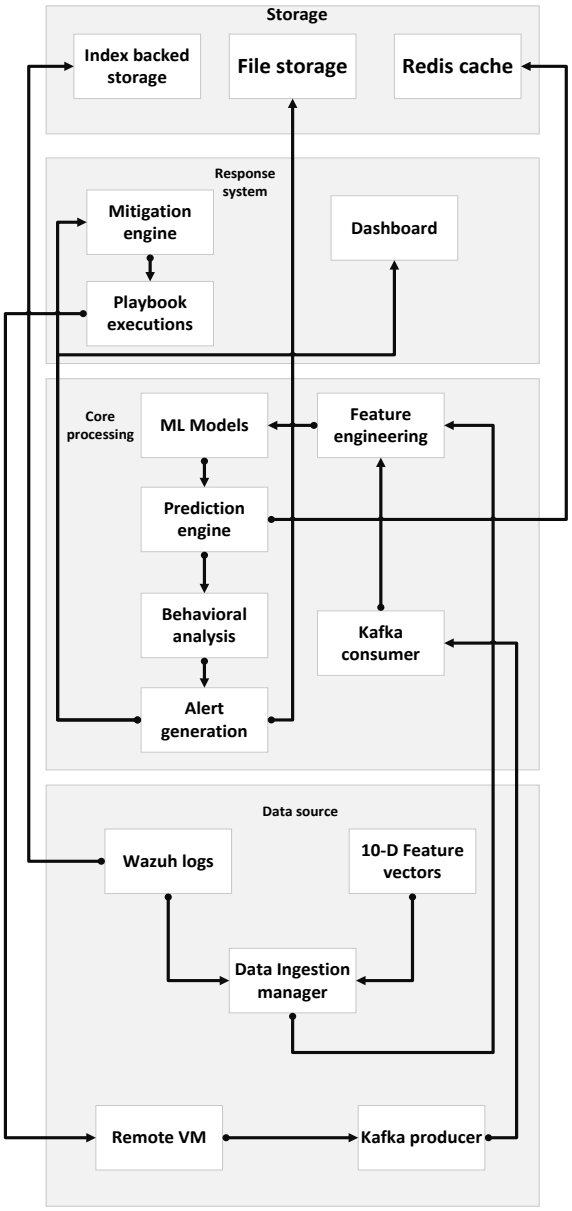

Figure 4.1: System Architecture

This diagram explains that the 10D Feature vectors are fed to the ingestion manager and the wazuh logs are stored and also fed to the same ingestion manager after which core processing is carried out on those logs and then the mitigation.

4.2 Workflow Diagram

The diagram below shows the system's workflow that is essential to understand how the system carries out the work that it does.

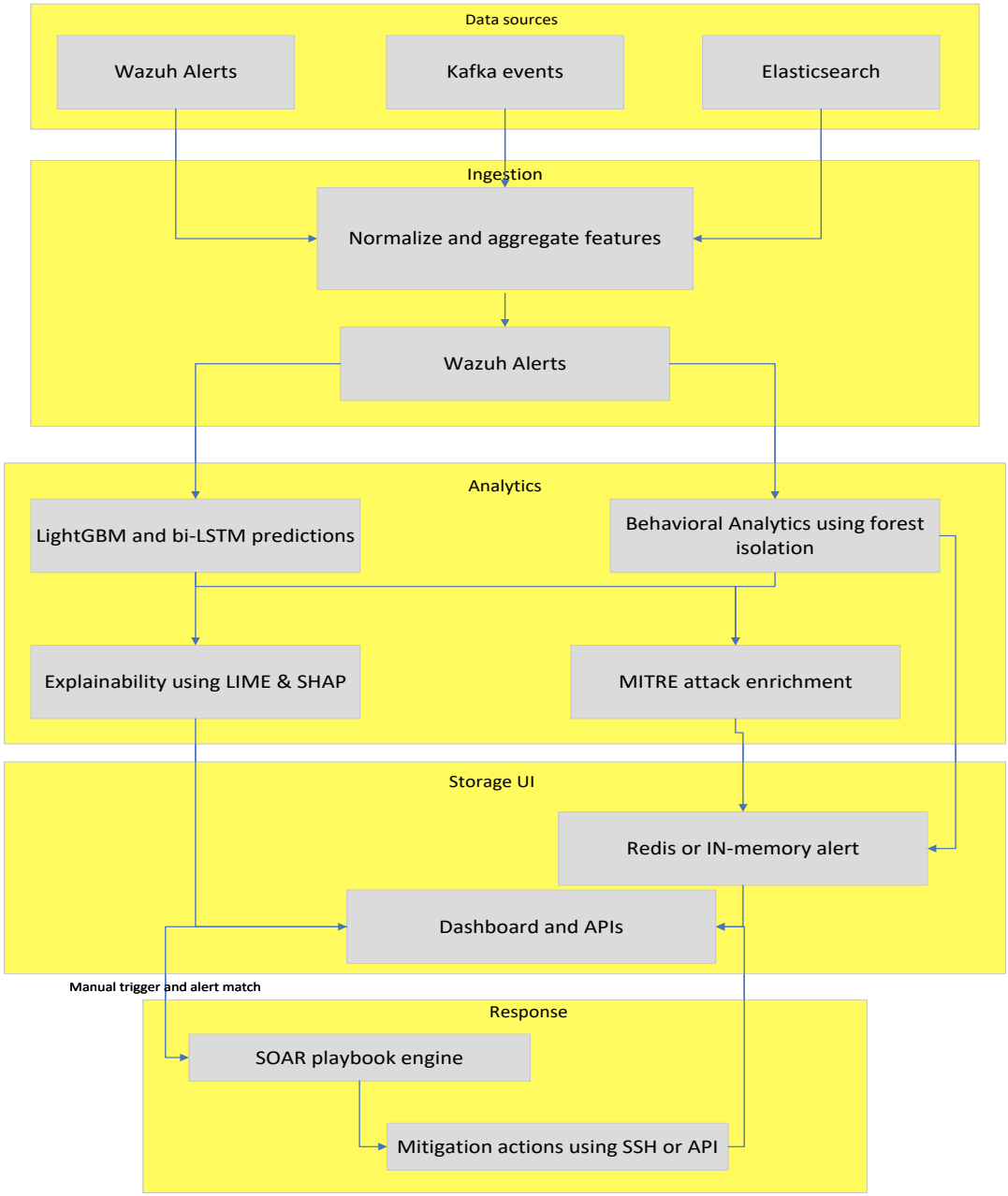

Figure 4.2: Workflow Diagram

4.3 Design Constraints

This section looks over the non functional requirements and limitations that influence the technical design of the Interpretable Next Generation SIEM solution.

4.3.1 Technical and Performance Constraints

1. **Real-Time Processing:** The system must support high performance to handle and process large volumes of data in real time to successfully reduce threats. This needs the use of fast responding technologies such as Apache Kafka.
2. **False Positive Minimization:** Minimize False Positives; An important constraint will be to reduce false positives and alert fatigue, hence security analysts only receive high severity alerts on high-confidence threats.
3. **Open-Source Mandate:** Open Source Requirement; Design will need to use only open-source tools (e.g., Wazuh, Kafka, TensorFlow) to ensure the solution is accessable to all.
4. **Scalability:** The architecture must be scalable to process large datasets and handle real-time log efficiently without performance gaps.

4.3.2 Operational and Environmental Constraints

1. **Integration with Existing Infrastructure:** The design should facilitate the integration of the organization's system into its existing security structure without significant changes. It is assumed that organizations already have some fundamental security controls like firewalls and endpoint protection in place.
2. **Fault-Tolerance:** Building a fault tolerant SIEM solution that still maintains high performance levels despite huge amounts of data being generated is important for continuous protection.

4.3.3 Assumptions

The design of the system depends upon the following assumptions:

1. The SIEM is supposed to be used in a **distributed IT environment**. It should include both on-premise and cloud elements.
2. Already, the basic security measures in organizations that practice this should be in place e.g. firewalls and endpoint protection.
3. The project presupposes the access to the appropriate threat intelligence feeds (OpenCTI, MISP) and effective monitoring real time log sources.

4. The AI-based detectors will need continuous learning and implementation. to make up with new-fashioned type of attacks.

4.4 Design Methodology

To develop the Interpretable Next Generation, the Agile Development Model was used. The reason why this model was selected was because of its flexibility. This will enable it to iteratively enhance and keep on improving the AI powered parts.

4.4.1 Key Phases of Agile Development

1. **Requirement Analysis (Phase 1):** The definition of system and security threats identification. capabilities.
2. **Design (Phase 1):** The designing of system, security threats and identification capabilities.
3. **Implementation (Phase 2-3):** The different modules should be developed and integrated, such as installation of Wazuh, Kafka, and Elasticsearch and development of AI anomaly detection models.
4. **Testing (Phase 4):** Evaluating the system's effectiveness against simulated threats and performing performance testing.
5. **Deployment & Monitoring (Phase 5):** Deploying in a live environment and refining the system based on operational performance feedback and documentation.

4.5 High Level Design

The system is conceptualized as a distributed, data-pipeline architecture consisting of three main functional tiers: Data Collection, Real-Time Processing, Analysis, Response and Visualization.

4.5.1 Functional Breakdown

1. **Log Collection:** Handled by Wazuh Agents on various endpoints.
2. **Data Ingestion:** The data ingestion is carried out in file based integration as well as Kafka stream to transfer logs.
3. **Threat Detection:** The core function, performed by Bi-LSTM model and Light-GBM model and a Hybrid model that comprises of combination of the prior models for anomaly detection [5][6].

4. **Threat Intelligence Enrichment:** External services like OpenCTI and MISP provide context to threats.
5. **Automated Mitigation:** AI interprets and performs mitigation on the activities that it deems critical [7].
6. **Monitoring/Reporting:** Achieved through dashboards provided by Kibana and a custom made dashboard create using flask.
7. **Explainability:** LIME and SHAP used for bi-LSTM and light-GBM models respectively interpret which inputs are the contributing factors to the prediction of severity of logs [4][8].

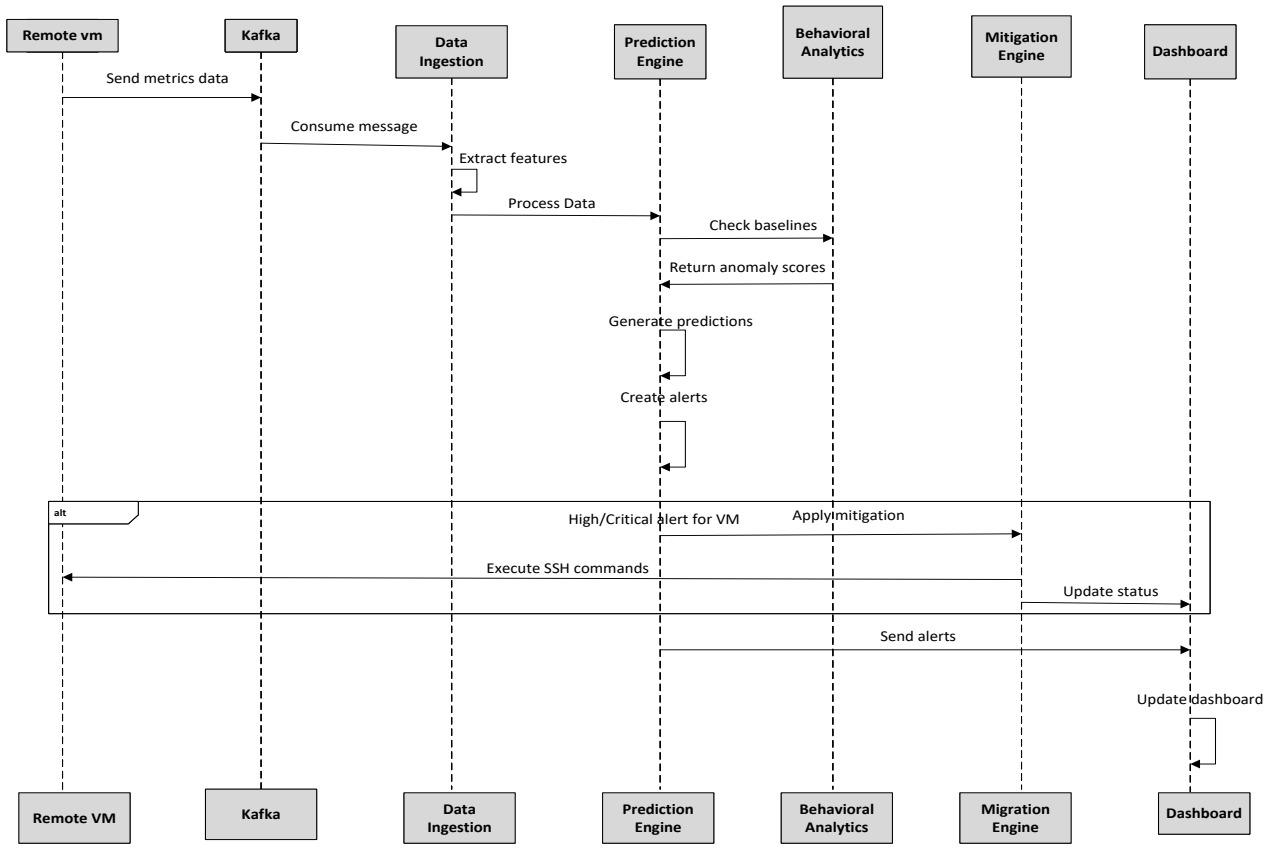

Figure 4.3: Sequence Diagram Real-time Data Processing Flow

This Sequence Diagram illustrates the sending of data from the VM through kafka to prediction engine to then be analyzed for mitigation.

4.6 Low Level Design

The low-level design specifies the individual components, algorithms, and libraries used.

4.6.1 AI/ML Detection Algorithms and Frameworks

1. **Real-Time Processing:** Apache Kafka Streaming processes log data for real-time insights.
2. **Machine Learning (ML):** Scikit-learn, TensorFlow, and PyTorch are the libraries and frameworks for building AI models.
3. **Deep Learning Models:** Bi-LSTMs (Bidirectional Long Short-Term Memory Networks) and Light-GBM are used for advanced threat detection and analysis of log data [2].
4. **Explainable AI (XAI):** Components are integrated to provide interpretability to the AI's detection decisions, which is central to the project's goal using behavioral analytics module [2][4].

4.6.2 Log Collection and Management

1. **Log Collection:** Wazuh collects logs from endpoints, operating systems, and applications.
2. **Log Transfer:** Apache Kafka manage the real-time data streams.

4.6.3 Response and Investigation Tools

1. **Orchestration:** AI automates mitigation actions using SOAR execution playbooks namely APT detection response, insider threat response, network intrusion response.
2. **Incident Response:** SOAR mitigations are automatically applied to endpoints [9].

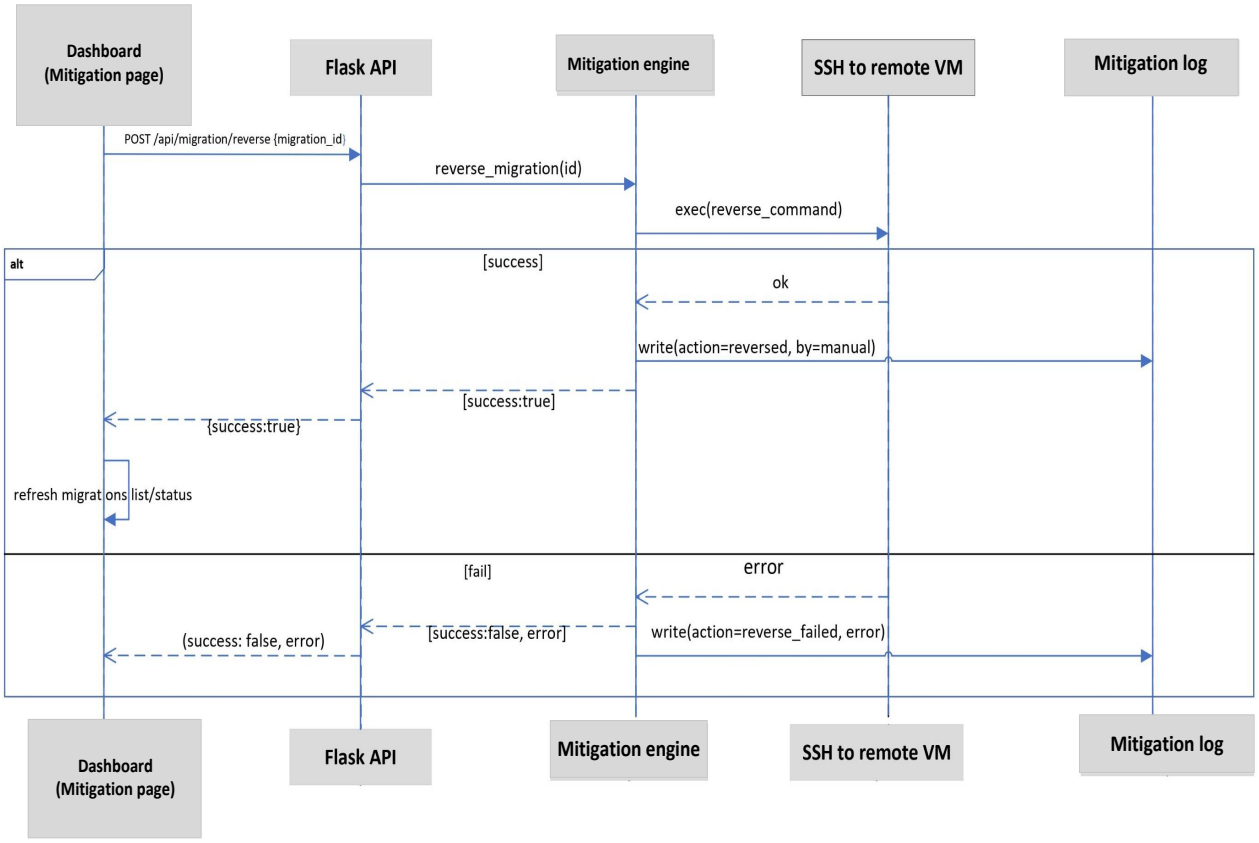

Figure 4.4: SOAR Playbook Execution Reversal

The diagram illustrates how the SOAR Playbook is reversed from the dashboard through the backend and mitigation logs are maintained for it afterwards.

4.7 Database Design

The system relies on a distributed data storage approach for log analysis and indexing.

1. **Primary Log Storage:** The **Wazuh Indexer** stores logs for analysis. This NoSQL, document-store approach is optimized for indexing and searching vast, time-series log data.
2. **Streaming Buffer: Apache Kafka** manages real-time data streams, acting as a robust buffer for log transfer.
3. **Redis:** Redis is used as real-time alert store/queue to persist and share alerts between producers and the dashboard.

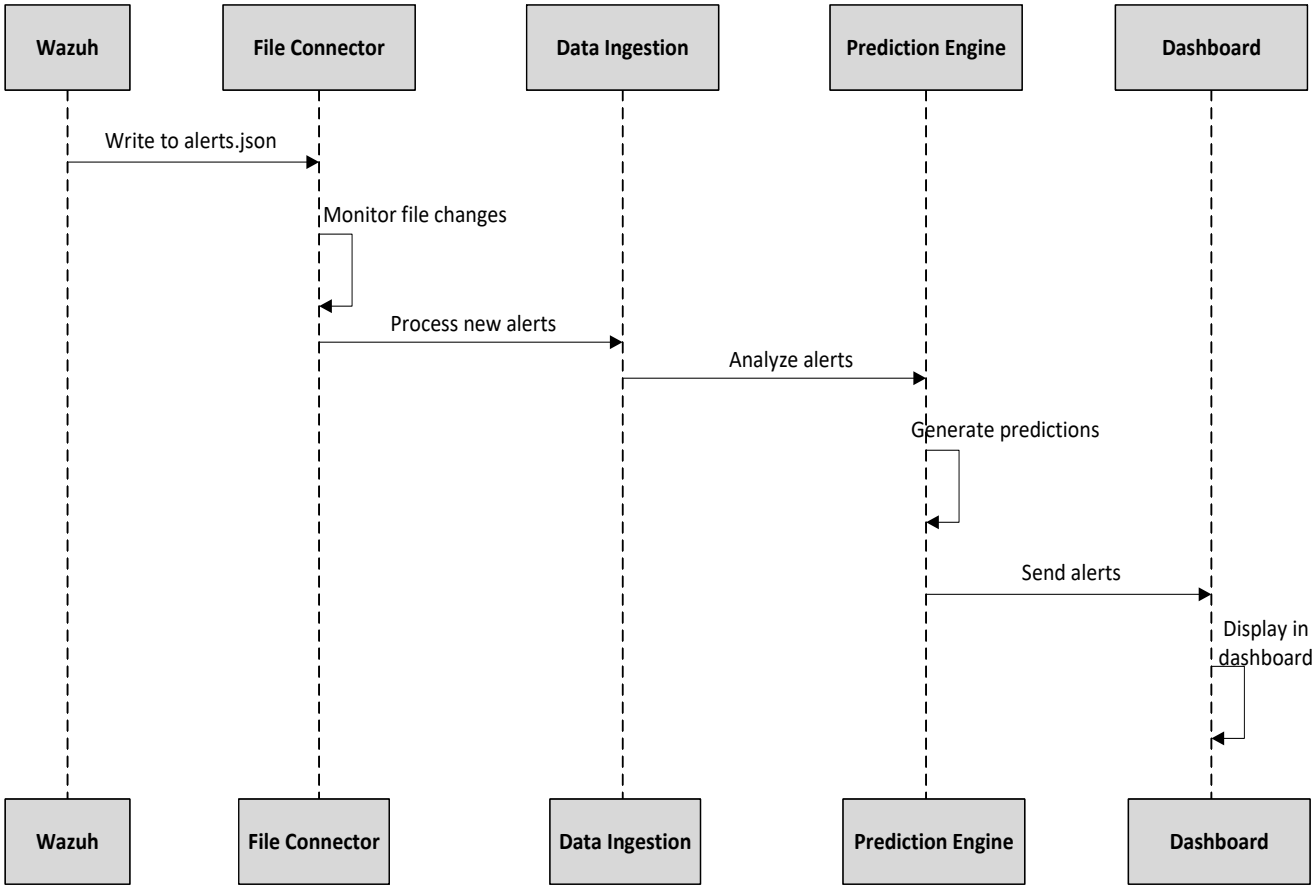

Figure 4.5: Wazuh Integration Flow

This above diagram illustrates how new alerts are labelled for their severity and then displayed on the dashboard.

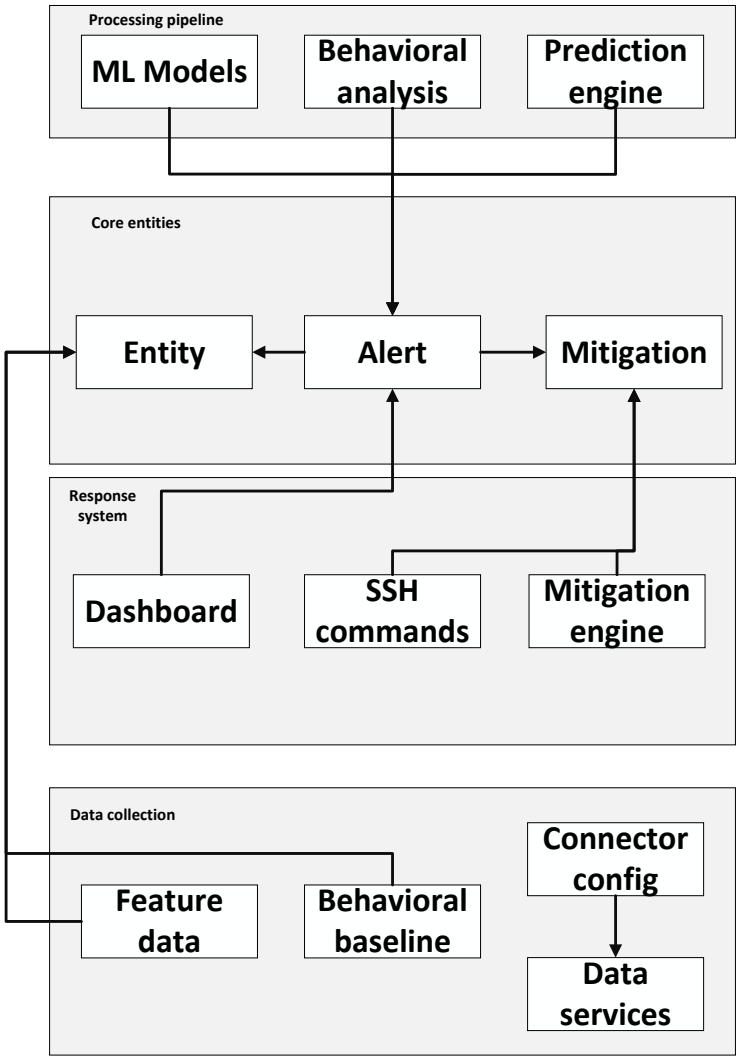

Figure 4.6: Data Model Relationship

This diagram illustrates the connection between Processing pipeline, Core entities, Response System and Data Collection.

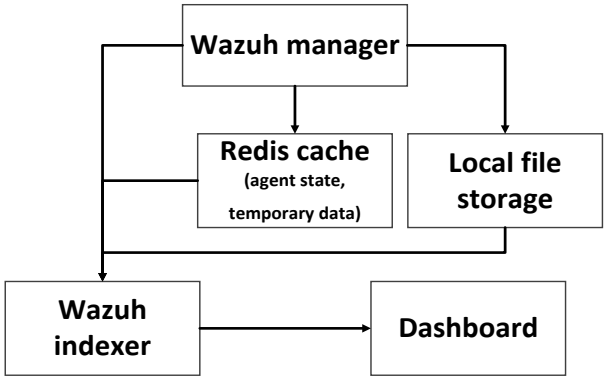

Figure 4.7: Data Storage Mechanism

This figues illustrates the innovative data storage mechanism highlighting the no use of relational database system.

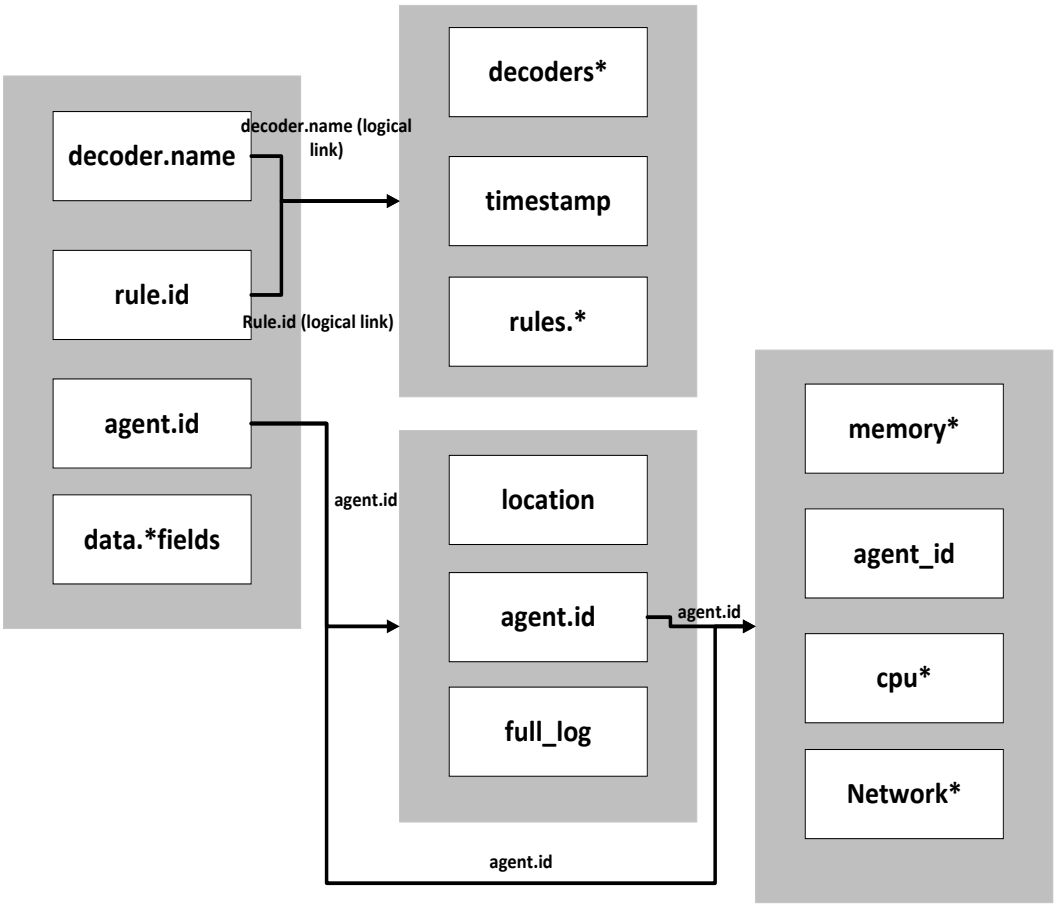

Figure 4.8: ER diagram

This diagram shows the wazuh's configuration of their elasticsearch entitiy relationships

4.8 GUI Design

The system's Graphical User Interface (GUI) design focuses on providing real-time security visibility and tools for in-depth investigation.

1. **Real-Time Dashboards: Custom-made dashboard** and **Kibana** provide visualization and dashboards for real-time security monitoring [5].

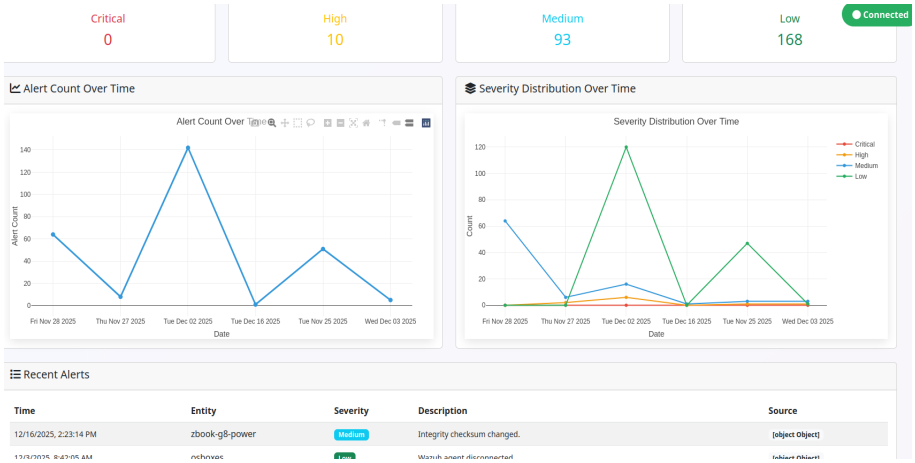

Figure 4.9: Alert Timeline Analysis

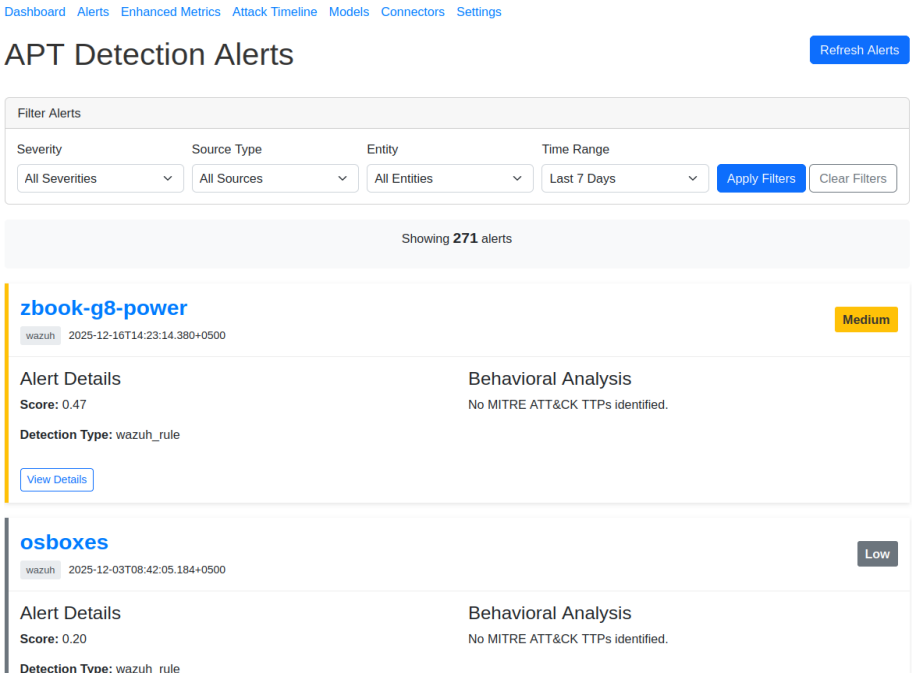

Figure 4.10: Alert Details Dashboard

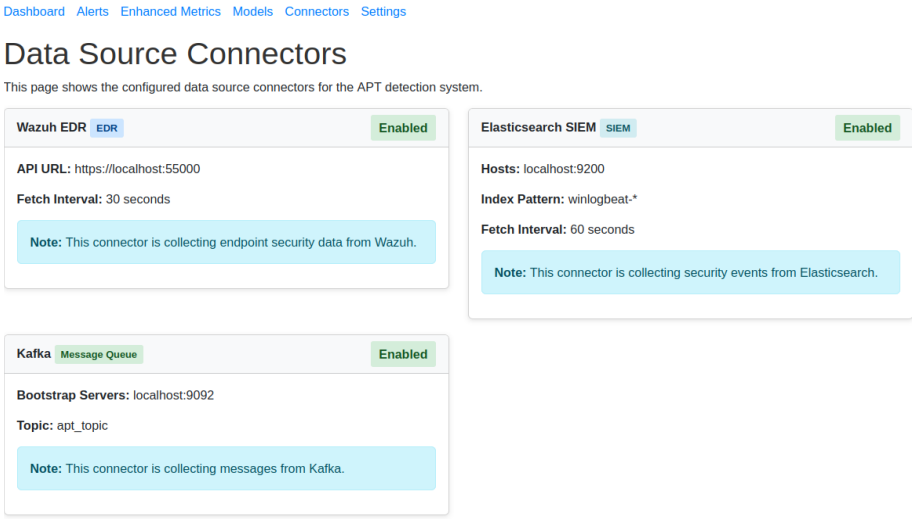

Figure 4.11: Connectors Details

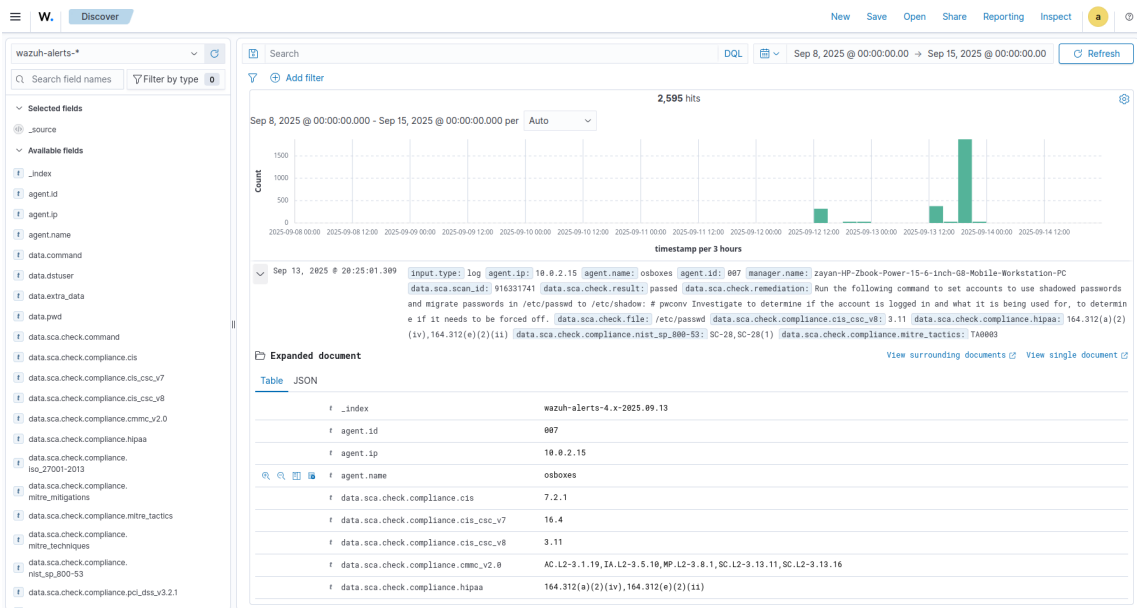

Figure 4.12: Wazuh Kibana Dashboard

2. **SOAR execution:** It offers automated playbook executions along with optional manual execution of those playbooks [9].

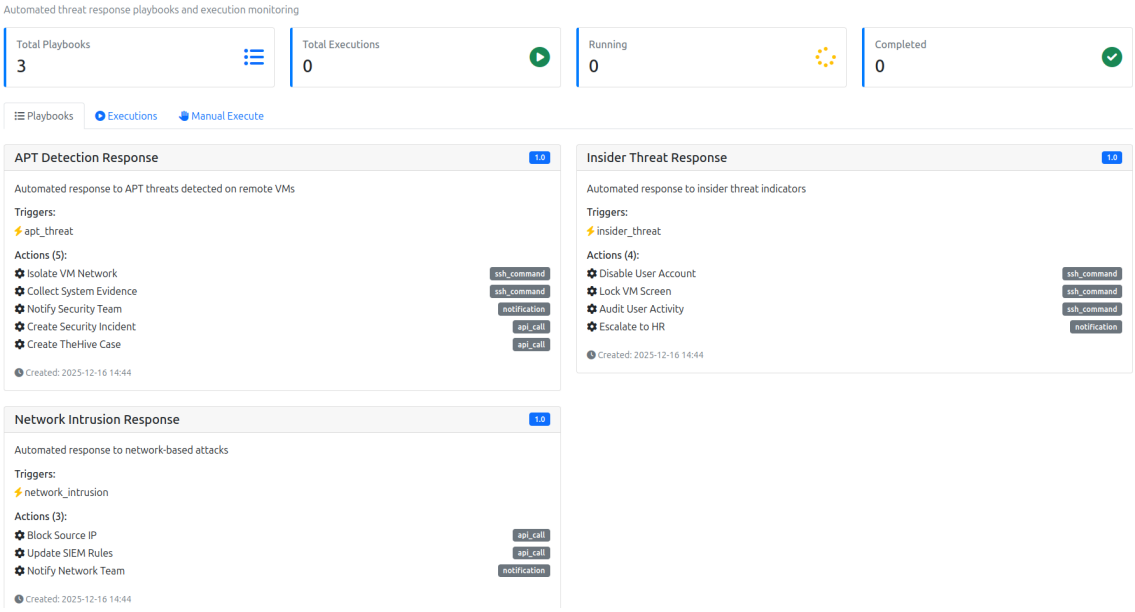

Figure 4.13: Soar Executions Dashboard

3. **Interpretability View:** The interface must support the presentation of interpretability (XAI) results to explain the AI's anomaly detection decisions based on the contributing factors among the inputs given to AI models [2][8].

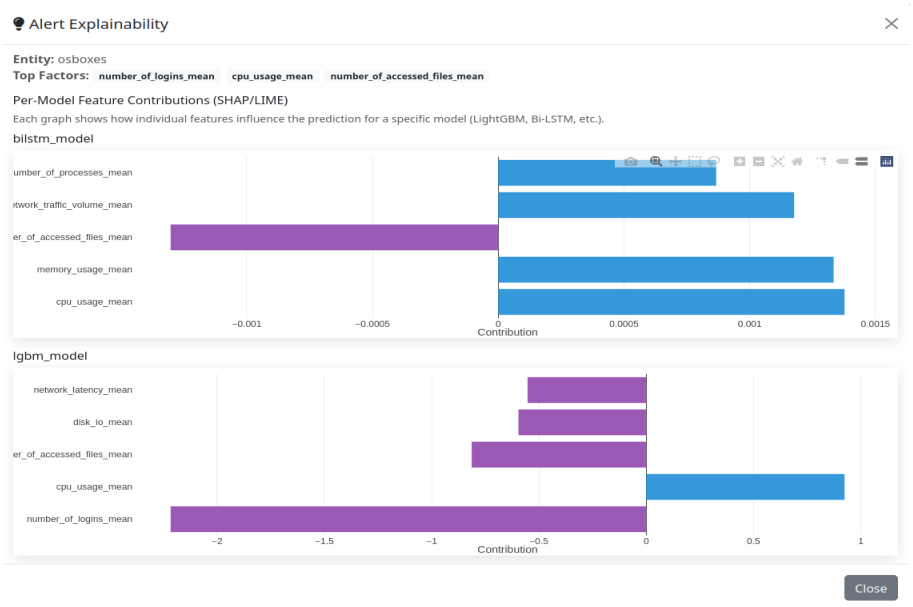

Figure 4.14: Interpretability View

4.9 External Interfaces

The SIEM system must interact with several external systems and data sources to function effectively.

1. **Endpoint Devices:** The system interfaces with endpoint devices (Linux) via **Wazuh Agents** for log collection.
2. **External Threat Intelligence Feeds:** It utilizes external threat intelligence sources like **OpenCTI** and **MISP** to enrich security events and enhance security decisions.
3. **Development and Deployment Environment:** Interfaces with external tools like **Python** (for ML and backend services), **Flask** (for backend services).

Chapter 5

System Implementation

Implementation is the step of moving an idea from concept to reality. System construction is an embodiment of a technical specification or algorithm as a program, software component, or other computer system via programming and deployment. This chapter provides extensive technical implementation of the project, including how components were configured, leveraging intrinsic and extrinsic security and the development environment used to realize the Interpretable Next-Generation SIEM solution into a working solution.

5.1 System Architecture

This section describes the design and implementation of the system architecture, covering internal components, operational logic, communication mechanisms, and the technologies used throughout the pipeline.

5.1.1 System Internal Components and Functionality

The implementation consists of a tightly controlled pipeline built on multiple open-source components, designed to enable real-time threat detection with automated response capability when required.

5.1.1.1 Log Collection Layer (Wazuh Agents and Manager)

1. **Wazuh Agents** are deployed on Linux endpoints to collect system logs, application logs, and security events.
2. The **Wazuh Manager** receives, normalizes, and processes the incoming events. A forwarder (e.g., Filebeat or a custom output module) then ships the processed logs into the real-time streaming layer for downstream analytics.

5.1.1.2 Data Streaming Layer (Apache Kafka)

1. **Apache Kafka** functions as a distributed, fault-tolerant streaming platform that buffers, partitions, and transports high-volume event data between the collection layer and the processing layer. This decouples ingestion from analytics and ensures horizontal scalability.

5.1.1.3 Real-Time Processing and Analysis (Apache Spark Streaming and ML)

1. **Apache Spark Streaming** consumes data from Kafka using micro-batch or continuous processing mode. It performs real-time parsing, enrichment, and feature extraction on the streaming security logs.

2. The **Machine Learning subsystem** provides the intelligence layer. Python-based frameworks are used to build and train the AI models. A combination of bi-directional LSTM networks (for sequential behavior modeling) and LightGBM (for classification) enables accurate detection of Advanced Persistent Threat (APT) patterns [7].

5.1.1.4 Threat Response Logic (SOAR and Wazuh Active Response)

1. **Wazuh Active Response** executes controlled mitigation scripts directly on the affected endpoints, such as blocking malicious IPs or terminating suspicious processes.

2. **SOAR playbooks** orchestrate broader automated responses. Depending on system integrations, these playbooks can restrict network access, isolate hosts through firewall APIs, or disable user accounts to contain active threats.

5.1.2 Tools, Technologies Used, and Development Environment

The project uses a comprehensive open-source technology stack.

5.1.2.1 Programming Languages

Python is used extensively for machine learning model development (Scikit-learn, TensorFlow, PyTorch) and backend service implementation, typically through the **Flask** framework.

5.1.2.2 Big Data and Streaming

Apache Kafka provides real-time, high-throughput event streaming and is the backbone of the system's data movement layer.

5.1.2.3 AI and ML Frameworks

TensorFlow and **PyTorch** are used to build, train, and optimize deep learning models, including bi-LSTM architectures for advanced anomaly and sequence-pattern detection.

5.1.2.4 Visualization

Kibana and a **custom-built dashboard** are used to present real-time security analytics, visualizations, and incident insights [10].

5.1.3 Database Security (Wazuh Indexer)

Security measures for the log storage component, the **Wazuh Indexer**, ensure that the integrity and confidentiality of stored security events meet compliance and reporting requirements.

5.1.3.1 Safe Data Storage

Data integrity is ensured by enforcing write-once, read-many (WORM)-style behavior for raw logs wherever permissible. Retention policies are aligned with regulatory requirements such as HIPAA.

5.1.3.2 Remote Access

Direct access to the Indexer API or underlying file system is restricted to local services and secured maintenance jump-hosts.

5.1.3.3 Authentication and Authorization

Access to Indexer APIs is protected by strict authentication controls. Role-based authorization ensures that users only access log categories permitted by their assigned privileges.

5.1.3.4 Auditing and Logging

The Indexer performs continuous self-auditing. All access events, modifications, and search queries are logged and monitored to ensure that any attempt at tampering or misuse is immediately detectable.

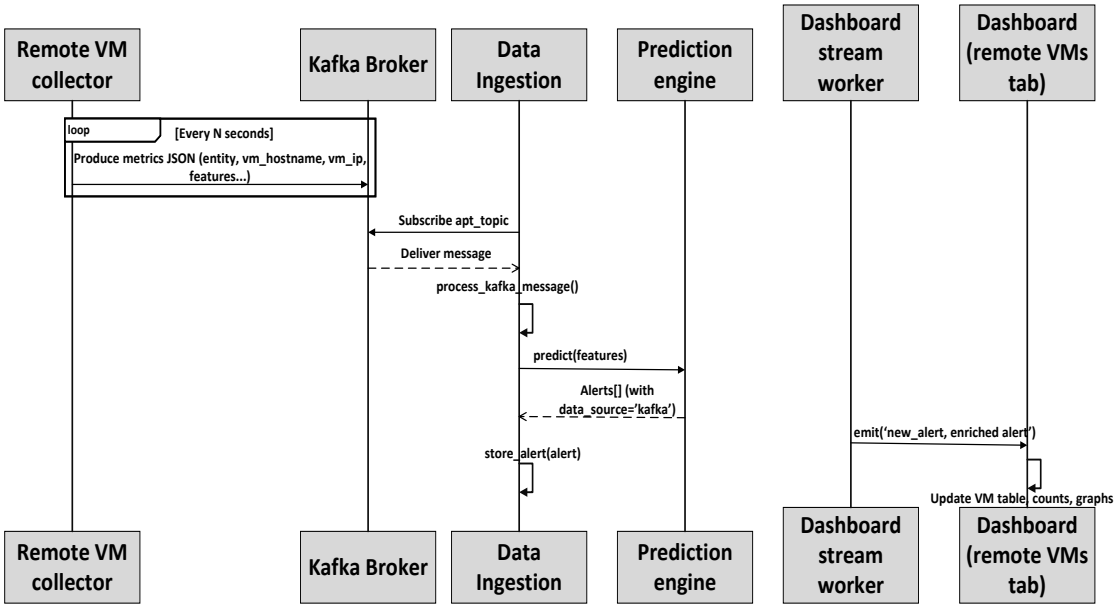

Figure 5.1: Remote VM Flow

This diagram illustrates the workflow of the remote VM and how it sends the data about itself to the prediction engine and the logs are labelled and displayed on the dashboard and updated after an interval of time.

Chapter 6

System Testing and Evaluation

Testing was taken care of especially to ensure both operational performance, reliability, and usability of the proposed Next Generation SIEM solution. System testing verifies that the entire solution is integrated and works well, including all modules, from log ingestion to AI anomaly detection to automated response and meets identified requirements. This section information of the assessment strategies employed and evaluation criteria.

6.1 Evaluation Criteria

The system was fully tested to guarantee that it achieves the security, usability and goals. scalability. The most significant areas of evaluation are summarized in Table 6.1.

Table 6.1: System Evaluation Criteria

Criterion	Description
Expressibility	Ability to represent complex threat scenarios and detection logic using interpretable AI models.
Functionality	Verification of core modules such as log ingestion, anomaly detection, and automated response.
Ease-of-Use	Assessment of dashboard clarity and user interface for security analysts.
Scalability	Performance under increasing data volumes and concurrent threat events.
Accuracy	Precision and recall of AI models in detecting true threats while minimizing false positives.
Resilience	Fault tolerance and recovery from unexpected failures or attacks.

6.2 Types of System Testing

Various types of tests were done to make sure that it was thoroughly validated. These are summarized in Table 6.2.

Table 6.2: Types of System Testing

Testing Type	**Purpose**
GUI Testing	Verified responsiveness and clarity of Kibana and custom dashboards.
Usability Testing	Evaluated interface intuitiveness and navigation for security analysts.
Performance Testing	Measured throughput, latency, and resource usage under load.
Compatibility Testing	Ensured operation across Linux, and on-prem environments.
Exception Handling	Validated system behavior under malformed inputs and failures.
Load Testing	Assessed system stability under high log volume and concurrent events.
Security Testing	Simulated attacks to test detection and response capabilities.

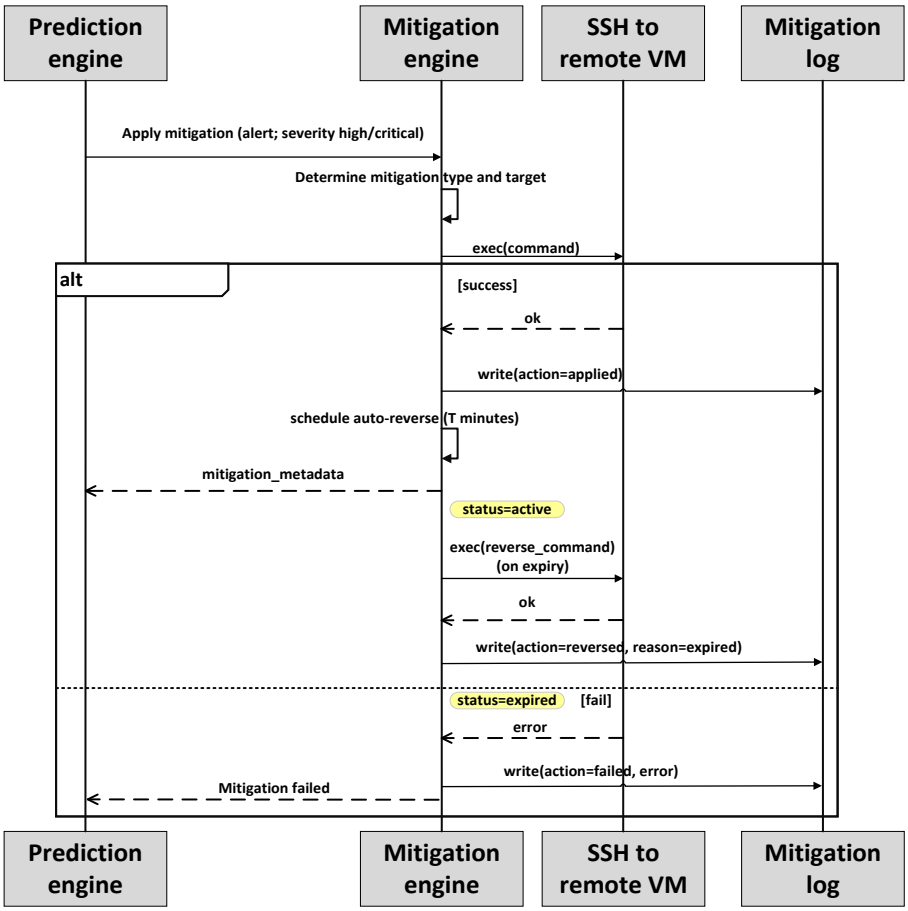

Figure 6.1: Automated Mitigation with Reversal

This diagram illustrates how after the prediction engine labels the log, the mitigation is automatically performed and mitigation logs are then maintained.

6.3 Test Cases

To validate the system's behavior under various conditions, a series of test cases were designed and executed. Each test case includes the objective, input conditions, expected outcome, and actual result. The following test cases were executed to validate the core functionalities and non-functional aspects of the Next-Generation SIEM system. Each test case is presented in its own table for clarity and modular documentation.

Table 6.3: TC01 – Log Ingestion from Endpoint

Test Case ID	TC01
Title	Log Ingestion from Endpoint
Input	Linux logs via Wazuh agent
Expected Output	Logs successfully stored and indexed in Wazuh Manager
Result	Pass

Table 6.4: TC02 – Real-Time Anomaly Detection

Test Case ID	TC02
Title	Real-Time Anomaly Detection
Input	Suspicious new account creation
Expected Output	Alert triggered in the custom dashboard
Result	Pass

Table 6.5: TC03 – Automated Response Execution

Test Case ID	TC03
Title	Automated Response Execution
Input	Detection of malicious IP address
Expected Output	IP blocked via SOAR
Result	Pass

Table 6.6: TC04 – Insider Threat Response Execution

Test Case ID	TC04
Title	Insider Threat Response Execution
Input	Detection of malicious activity in the agent VM
Expected Output	Timely log the user out of the device.
Result	Pass

Table 6.7: TC05 – Network Intrusion Response Execution

Test Case ID	TC05
Title	Network Intrusion Response Execution
Input	Detection of malicious network activity in the agent VM
Expected Output	Timely halt the internet access (incoming and outgoing).
Result	Pass

Table 6.8: TC06 – Log Forwarding Pipeline

Test Case ID	TC06
Title	Log Forwarding Pipeline
Input	Logs sent to AI/ML models
Expected Output	AI/ML models process logs in real time
Result	Pass

Table 6.9: TC07 – Dashboard Rendering

Test Case ID	TC07
Title	Dashboard Rendering
Input	Multiple alerts and VM's CPU usages and memory usage data.
Expected Output	Display the data asynchronously.
Result	Pass

1. **Severity prediction by AI:** AI and ML models successfully predicted the logs from wazuh agent as shown in the digram below

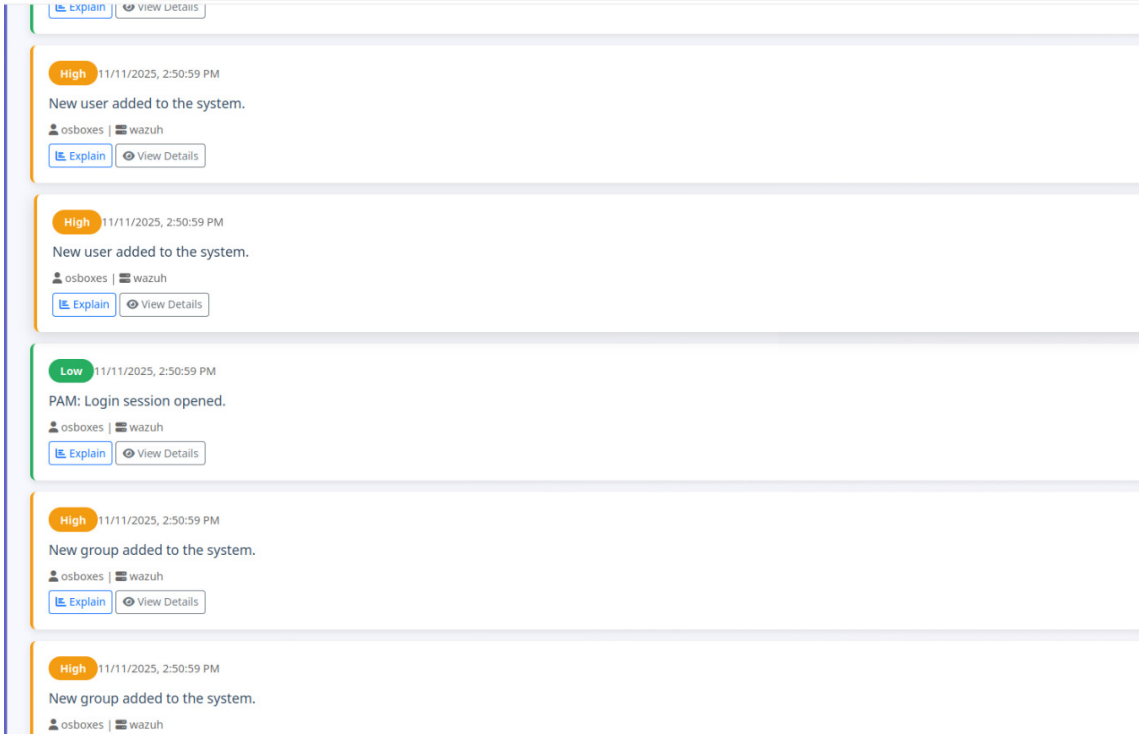

Figure 6.2: Diagram illustrating the labelling of logs from Wazuh agent by AI/ML models.

2. **AI/ML models scores and graph plotted on values of 10D feature vector on one wazuh agent log :** AI and ML models scores and the graph illustrating the 10 values being plotted that shows the contributing factors from among the 10 features.

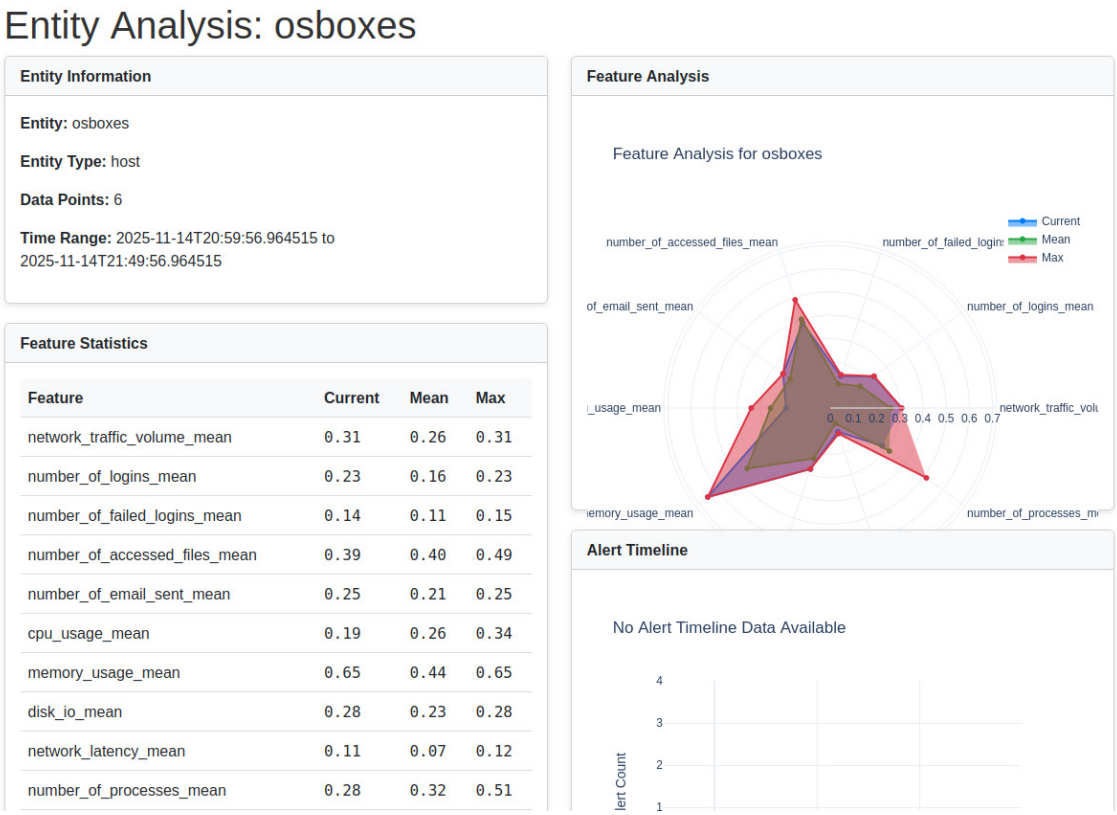

Figure 6.3: Diagram showing the contributing factors for labelling.

3. **Explainability/Interpretability done on the prediction of AI/ML models by LIME/SHAP:**LIME/SHAP graphically represent the contributing factor from among the 10D feature vectors as shown in the feature analysis and the diagram below

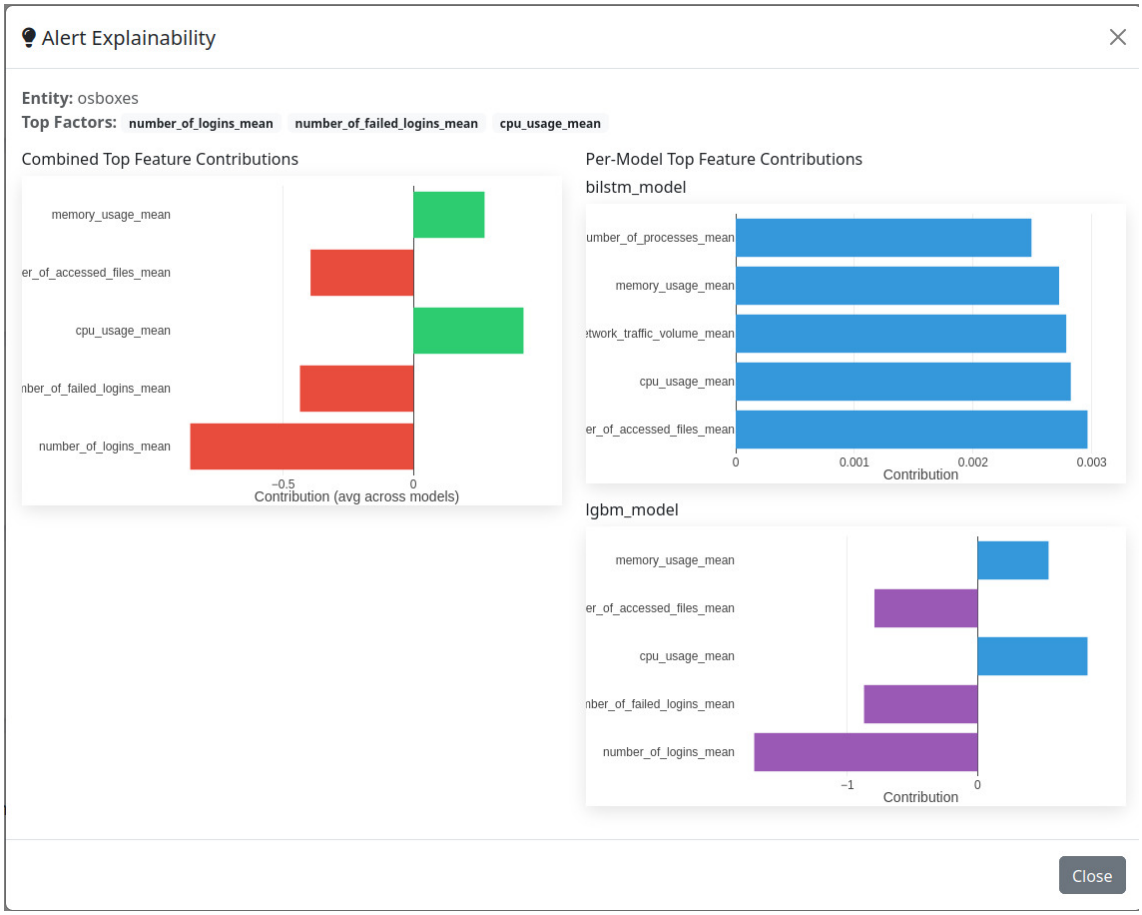

Figure 6.4: Diagram illustrating the explainability/Interpretability by LIME or SHAP.

4. **Automated SOAR playbook Execution:**SOAR applies automatic inside threat response mitigation action when it detects the a suspicious activity, (addition of new user and file accessing for passwd changing attempt as seen the figure 6.2.)

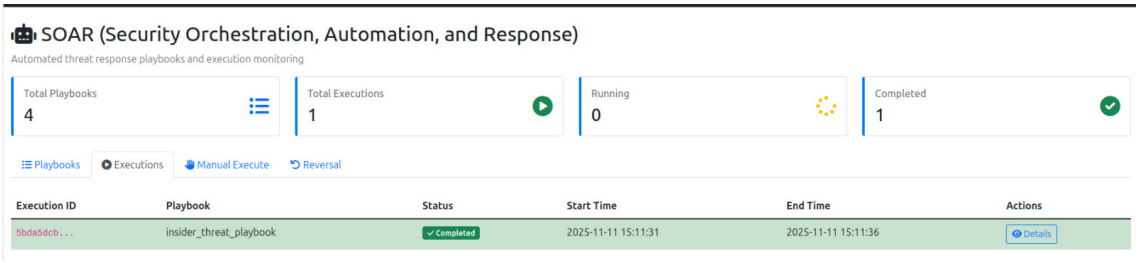

Figure 6.5: Diagram illustrating the list of running SOAR playbook executions.

5. **Log Severity-Time graph:** Displays the severity of alerts over the period of time interval.

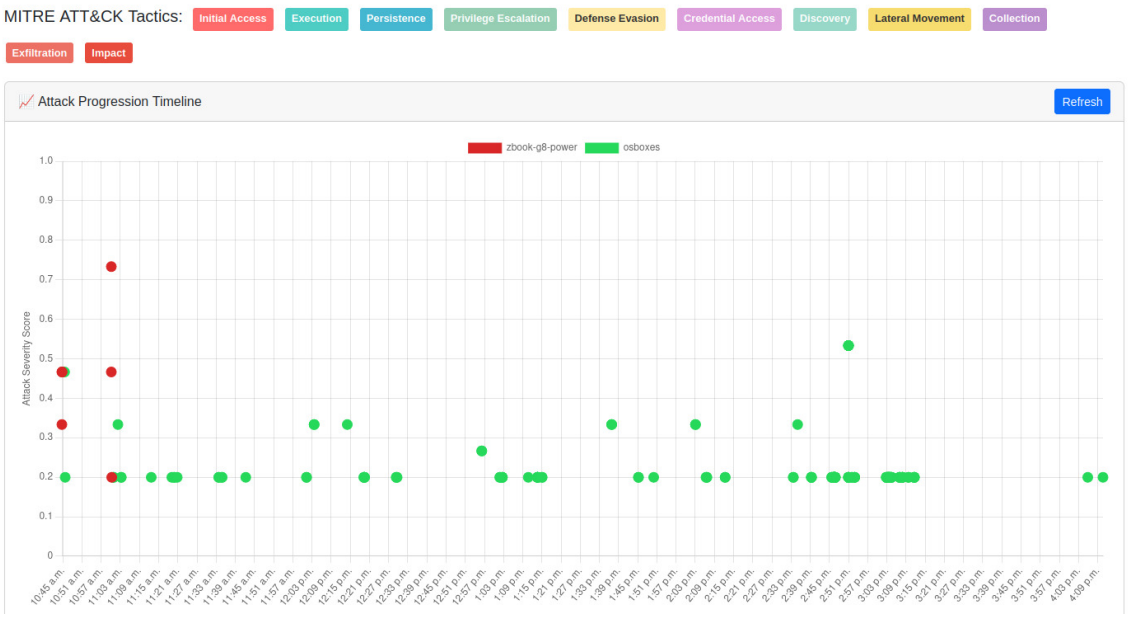

Figure 6.6: Diagram illustrating the graph between severity of real-time wazuh logs and time.

6.4 Evaluation Summary

The system successfully passed all major testing phases. Key outcomes include:

1. **Detection Accuracy:** AI models achieved high precision and recall in identifying threats.
2. **Response Time:** Automated mitigation actions were executed effectively.
3. **Scalability:** The system handled high log volumes without performance degradation.
4. **Usability:** Dashboards were well defined and user-friendly.
5. **Resilience:** The system recovered gracefully after simulated failures.

The Next-Generation SIEM solution was prepared to work with systems with the help of system testing. implementation in business settings where it may bring an improvement of their cybersecurity.

Chapter 7

Conclusions

The feasibility was guaranteed by the development of the Interpretable Next-Generation SIEM system and performance of combining some elements such as AI driven anomaly detection, automated threat response and threat intelligence into one security framework. Evaluation at different stages created significant understandings about the problems, disadvantages and potentials of the contemporary cybersecurity processes. We were able to demonstrate the design and testing of an interpretable next-generation Security Information and Event Management (SIEM) architecture, which addresses the inherent issues of the existing SIEM systems such as alert overload, high false-positive rates, inability to identify advanced persistent threats (APTs), and slow incident response. Due to Wazuh-based holistic log collection and the use of such sophisticated machine learning methods as LightGBM and Bi-LSTM, the suggested system proves to be much more efficient in terms of both recognizing threats and performance. Explainable artificial intelligence methods, namely LIME and SHAP, allow creating interpretable threat scores, which would help security analysts understand the rationale behind every alert and, therefore, build trust in AI-based security decisions.

In addition, the system is effective in integrating external sources of threat intelligence, as well as AI-based detection, to quickly detect known attacker signatures and, at the same time, identify never-before-seen or novel threats based on learned behavioral patterns. Modular backend architecture which is elasticity-enhanced will guarantee scalable log indexing, real-time visualization and effective manipulation of vast amount of security data. Empirical testing proves that the suggested solution enhances a better detection rate, lessens the workload on the analysts, and is more responsive to APTs than the traditional SIEM solutions. In general, this interpretable next-generation SIEM does not only solve the severe flaws of the traditional systems, that is, the absence of interpretability, the overwhelming workload, and the ineffective APT detection but also provides a transparent, automated, and dependable security monitoring tool. With accurate threat detection and explainable AI, the system enables security teams to make informed, confident decisions

that work in a threat environment that has grown extremely fast and thereby enhancing the effectiveness of the contemporary security operations.

7.1 Key Outcomes

The major objectives of the project were met:

1. **AI-Driven Detection:** ML models i.e. bi-LSTMs and light-GBM, could detect threat patterns and APTs with high precision.
2. **Automated Response:** SOAR allowed real-time mitigation actions. This reduces manual intervention and improves response time.
3. **Threat Intelligence Enrichment:** This refers to external threat intelligence platforms (OpenCTI/ MISP) was informative. This improved the accuracy of notifications.
4. **Scalability and Performance:** The system was able to cope with big volumes of logs using Kafka. It therefore is appropriate to use in enterprise settings.
5. **Visualization and Usability:** Flask-based dashboard and Kibana-based dashboard presented specific and practical information to security analysts.

7.2 Lessons Learned

The project identified some of the key factors in the development of efficient SIEM architecture:

1. The modern threats cannot be dealt with using rule-based detection alone. AI models must be fine-tuned, constantly trained on finer results.
2. Intelligent suppression, and correlation mechanisms can be used in order to reduce alert fatigue.
3. In order to prevent bottlenecks, real time processing requires thorough optimization of resources.

7.3 Future Work

Although the present system addresses its design objectives, there are some areas in which the project fails. can be extended or refined:

1. **Adaptive Learning:** It can be achieved through applying online learning models thus threat detection can be improved. That is because these models are able to adapt to new data over time.

2. **Multi-Tenant Support:** This is the way of making the system work with multiple organizations could extend the applicability.
3. **Compliance Automation:** Compliance expanding automation to produce audit and prepared compliance reports would simplify the regulatory processes.

7.4 Refinements for Future Iterations

In case the project were to be repeated or extended, the following improvements might be refined:

1. Perform previous benchmarking of the AI models to choose the best architectures.
2. Add additional log sources (e.g. cloud-native telemetry, IoT devices, real time) and tested APT's that had been successful in their sabotage of security) during testing.
3. Allot more time to real analyst testing of usability to perfect dashboard.
4. Consider hybrid deployment models of distributed IT with the use of cloud computing environments.

To sum up, the given project creates a reliable base to the creation of smart, scaled-up and explainable SIEM systems. It solves some of the major constraints of the traditional solutions and creates new possibilities in future automation of cybersecurity.

Appendix A

User Manual

The supplementary information is presented in this appendix to help users to operate the Inter- pretable Next-Generation SIEM system. It describes the general usage, interface objects, and operational workflow. It will make sure that users interact well with the system.

A.1 System Overview

SIEM system combines the log gathering, real time threat recognition, automatic responses, and threat intelligence enhancement. It is developed to work in distributed environments.It uses open source tools such as Wazuh, Kafka and SOAR components.

A.2 User Roles

1. **Security Analyst:** Monitors dashboards, investigates alerts, and manages cases.
2. **System Administrator:** Manages deployment, configuration, and system health.
3. **Threat Intelligence Officer:** Integrates external feeds and enriches alerts.

A.3 Interface Components

1. **Wazuh Manager:** Central log processing and rule engine.
2. **Kibana** Visualize system metrics, alerts, and log trends.
3. **Custom Dashboard** Custom built dashboard using Python for manual threat mitigation.
4. **LIME or SHAP** dashboards used for visualization and explainability.

A.4 Basic Workflow

1. **Log Collection:** Wazuh agents collect logs from endpoints and forward them to the manager.
2. **Streaming & Processing:** Kafka Streaming handle real-time log ingestion and analysis.
3. **Threat Detection:** AI models detect anomalies and trigger alerts.
4. **Automated Response:** SOAR tools execute mitigation actions (e.g., IP blocking, internet cut-off, logging user out of their device etc).
5. **Investigation:** The events can be investigated through the graphs and explainability of LIME and SHAP.

A.5 System Requirements

1. **Hardware:** Minimum 8-core CPU, 16GB RAM, SSD storage.
2. **Software:** Python 3.8+(Flask), Wazuh, Kafka, Kibana, elasticsearch.
3. **Network:** Secure access to endpoints, threat intelligence feeds, and internal services.

A.6 Maintenance Guidelines

1. Regularly update AI models and threat intelligence feeds.
2. Monitor system performance using Prometheus and Grafana.
3. Backup configuration files and case data weekly.
4. Review alert precision and retrain models as needed.

A.7 Support and Documentation

Detailed technical documentation is available within the deployment repository. For troubleshooting, refer to the system logs, Wazuh documentation, and SOAR integration guides.

Appendix B

B.1 Tools and Technologies:

1. **Frontend:** HTML-5, CSS3, Javascript, Plotly.js, Chart.js Bootstrap 5.3.0, UI frameworks
2. **Backend** Python 3.9+, Flask 2.0+, web framework RESTful JSON APIs, Flask-SocketIO , WebSocket support, Flask-CORS 3.0.10+
3. **Libraries and packages** numpy, pandas, scikit-learn, scipy, lightgbm, tensorflow, keras, shap, lime, flask, flask-cors, flask-socketio, werkzeug, kafka-python, redis, matplotlib, plotly, seaborn, visualization, imbalanced-learn,joblib, pyyaml, requests, python-dateutil, tqdm,aiohttp, paramiko, elasticsearch, cryptography

References

[1] Md Liakat Ali, Kutub Thakur, Helen Barker, and Michael Chan. The rise of artificial intelligence: Industry insights and applications in security information and event management (siem). In *2024 IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 0477–0482. IEEE, 2024.

[2] Mustafa S. Aljumaily, Hayder K. Abd, and Elaf J. Majeed. Enhancing user and entity behavior analytics in siem systems using ai-powered anomaly detection: A data-driven simulation approach. Unpublished manuscript, 2025.

[3] Vijay B. Gadicha, Ajay B. Gadicha, Mohammad Zuhair, Zeeshan I. Khan, and Mayur S. Burange. Revolutionizing security information and event management (siem) systems: Harnessing deep learning for advanced threat detection. In *Project Management Information Systems: Empowering Decision Making and Execution*, pages 233–262. IGI Global Scientific Publishing, 2025.

[4] Diogo Gaspar, Paulo Silva, and Catarina Silva. Explainable ai for intrusion detection systems: Lime and shap applicability on multi-layer perceptron. *IEEE Access*, 12:30164–30175, 2024.

[5] Dattaraj Rao and Shraddha Mane. Zero-shot learning approach to adaptive cybersecurity using explainable ai. *arXiv preprint arXiv:2106.14647*, 2021.

[6] S. Muthuraj, M. Sethumadhavan, P. P. Amritha, and R. Santhya. Detection and prevention of attacks on active directory using siem. In *Proceedings of the International Conference on Information and Communication Technology for Intelligent Systems*, pages 533–541. Springer, 2020.

[7] Charlie Luca. Real-time monitoring & anomaly detection: Identifying adversarial interactions. Unpublished manuscript, 2024.

[8] Hassan Mokalled, Rosario Catelli, Valentina Casola, Daniele Debertol, Ermete Meda, and Rodolfo Zunino. The applicability of a siem solution: Requirements and evaluation. In *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 132–137. IEEE, 2019.

[9] Abdul Samad Mohammed, Vincent Kanka, and Aarthi Anbalagan. Automating security incident mitigation using ai/ml-driven soar architectures. *Advances in Deep Learning Techniques*, 2(2):22–65, 2022.

[10] Saurabh Chavan and George Pappas. From detection to decision: Transforming cybersecurity with deep learning and visual analytics. *AI*, 6(9):214, 2025.