Cyber Anomaly Detection using ML with Wazuh/ELK (CADM)

By

Hussain Ashiq
Namra Gul
Namra Haleema
Uzair Bin Abdul Hameed

Supervised by:
Dr Muhammad Sohail
Co-Supervised by:
Engr. Ali Ahmed Dar

Submitted to the faculty of Department of Computer Software Engineering,
Military College of Signals, National University of Sciences and Technology, Islamabad,
in partial fulfillment for the requirements of B.E Degree in Software Engineering.

June 2024

In the name of ALLAH, the Most benevolent, the Most Courteous

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report Final Year Project titled

Cyber Anomaly Detection using ML with Wazuh (CADM)

is carried out by

Hussain Ashiq

Namra Gul

Namra Haleema

Uzair Bin Abdul Hameed

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Software Engineering in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Approved by

Supervisor
Dr. Muhammad Sohail
Department of CSE, MCS

Date: _____

DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

ACKNOWLEDGEMENTS

First, we are thankful to Allah, the sole guidance I all domains, who gave us the strength and ability to complete our thesis. Without his divine help, we would have never been able to complete it.

We would like to express our sincere appreciation and thanks to our main supervisor,

Dr. Muhammad Sohail and our co-supervisor, Engr. Ali Ahmed, who greatly supported us in completing our work.

Apart from that, we would like to thank all our lecturers for their teachings that contributed to our self-development, knowledge, and attitude towards accomplishing the objectives of this project.

Finally, we would like to share our gratitude to our parents, siblings, and families for supporting our studies.

Plagiarism Certificate (Turnitin Report)

This thesis has ___ similarity index. Turnitin report endorsed by Supervisor is attached.

_____

Hussain Ashiq

NUST Serial No: 00000339400

_____

Namra Gul

NUST Serial No: 00000335290

_____

Namra Haleema

NUST Serial no: 00000359495

_____

Uzair Bin Abdul Hameed

NUST Serial no: 00000331604

_____

Signature of Supervisor

ABSTRACT

This project addresses the pressing need for real-time detection and response to cybersecurity anomalies: cyber-attacks and abnormal behaviours. The current challenge is to effectively identify and mitigate threats in the complex cybersecurity systems. Traditional methods often lack the capability to provide timely insights into anomalies, such as file creation, suspicious logins, and network activities, leaving organizations vulnerable to cyber threats. The gravity of this problem is immense because cyber-attacks are becoming more frequent and sophisticated. A fast detection and response system is of prime importance to reduce the damage associated with security breaches.

The Wazuh system works in near real time, gathering, analysing, and visualizing data. From this platform, the fundamental innovation lies in applying Machine Learning techniques that detect in real-time anomalies in the logs. Artificial Intelligence algorithms pick out deviations from normal patterns of behaviour, alerting users immediately in case of anomalous events and giving threat assessments with actionable recommendations. The visualization component is user-friendly, facilitated through dashboards of Wazuh and OpenSearch, to enable a user with any amount of expertise to navigate and understand the information displayed easily.

The CADM project will be complete with an integrated solution to this urgent problem, providing organizations with a quantum improvement in capability to protect their digital assets against the sophisticated landscape of cyber threats.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1. Problem Statement

The log monitoring and anomaly detection problem in industries is one of volume and complexity in the logs generated in real-time sources, coming from web servers, databases, network devices, and applications. These logs would carry critical information on system activities, user interactions, and network traffic. However, the manual analysis of such a huge amount of data to detect real anomalies from noise and false positives is highly resource-consuming and prone to human error. The industries require automated systems that sift through this data efficiently, identify genuine security threats or operational anomalies, and minimize the burden on human analysts to a minimum while maximizing detection accuracy.

1.2. Proposed Solution

Proposed Solution: CADM is a Cyber Anomaly Detection solution with ML, which will integrate the ML algorithms into Wazuh. Wazuh is an open-source SIEM suite for identifying anomalies among logs from different endpoints, which are injected in to be observed; these are network logs, web application logs, and authentication logs.

CADM solution aims to:

- Improve the accuracy of anomaly detection using ML algorithms.
- Reduce false positives and negatives through advanced analysis.
- Enable organizations to respond to genuine threats swiftly and effectively.
- Enable security analysts to create custom visualizations like bars, pie charts and so on, for quickly checking abnormalities and anomalies.

- Enable security analysts to create custom dashboards for specific use cases like ‘System Health Dashboard’ that will consist of various visualizations for different aspects.

1.3. Working Principle

CADM works on the following principles.

1. **Data Collection:** CADM collects log data from various sources, including network devices, servers, applications, and authentication systems.
2. **Data Ingestion:** The collected log data is ingested into OpenSearch, a part of CADM, which is a scalable and flexible search and analytics engine.
3. **Data Processing:** CADM then processes the ingested log data, transforming it into a suitable format for analysis.
4. **Anomaly Detection:** CADM has an anomaly detection plugin which applies machine learning (ML) algorithms to the processed log data to identify patterns and anomalies.
5. **Random Cut Forest Algorithm**: CADM uses a machine learning algorithm, Random Cut Forest (RCF), to analyze your data in real-time and set an ***anomaly grade*** & a ***confidence level*** to define how usual the events are based on the current dataset.
6. **Detector Creation:** CADM has an easy-to-use UI for creating anomaly detector for specific use cases. The detector can detect anomalies in near-real-time and also on historical basis i.e. when did anomalies occur in a collection of logs over a period of time.

7. **Near Real-time Anomaly Detection:** The trained models analyze incoming log data in real-time, detecting anomalies and deviations from the established baseline. The detections are in near real-time because of the latency in transferring logs from endpoint to CADM.
8. **Rule-Based Detections:** CADM also enables to apply rules on logs and detect various attacks and anomalies. Rules are mostly based on patterns and regular expressions.

1.4. Objectives

This Final Year Project has two types of objectives, general and academic.

1.4.1. 1.5.1. General Objectives:

- To develop a Cyber Anomaly Detection System (CADM) that utilized logs and machine learning (ML) algorithms to detect anomalies and give ease to Security Operations Center (SOC) analysts.
- Improve anomaly detection processes, reduce SOC analyst workload, and optimize resource utilization.
- To enhance overall security posture by enabling quicker detection of potential threats while minimizing manual effort and resource expenditure in monitoring and analyzing logs.

1.4.2. 1.5.2. Academic Objectives: -

- Development of CADM (Cyber Anomaly Detection using Ml) which will meet our FYP requirements.
- To implement the knowledge gained learned in this bachelor's degree.

- To lay the foundation of anomaly detection for the upcoming batches.

1.5. Scope

This final year project, CADM is being undertaken by us, a syndicate of four, to develop a system based on opensource technologies, that will analyze log data from endpoints and detect cyber anomalies i.e. various kinds of cyber-attacks and deviation from normal behavior, using machine learning algorithms and rule-based detections.

CADM will be detect attacks and anomalies in near-real time and historically, in a limited number of predefined use cases including, but not limited to the following:

- Failed logins anomaly
- Linux resource utilization anomaly
- Web Attacks like SQL injection, Path Traversal etc.

This project will not detect all kinds of anomalies, but anomalies in certain specific use cases. Nevertheless, the product will enable users to implement anomaly detectors for other use cases, requiring extra consideration.

1.6. Deliverables

The project will deliver CADM a real-time dashboard for log monitoring and anomaly detection using the Wazuh agent and machine learning (ML). CADM will provide intuitive visualizations of log data categorized by source, type, and severity, enabling quick identification of potential threats. It will highlight patterns and anomalies detected by ML algorithms. Additionally, CADM will display performance metrics of the Wazuh agent and monitored systems, aiding in anomaly detection. The focus on ML-driven anomaly detection outcomes will assist SOC analysts in

prioritizing and responding to security incidents effectively. By enhancing log analysis efficiency and enabling proactive threat detection, CADM aims to strengthen cybersecurity defenses within the organization.

1.7. Relevant Sustainable Development Goals

The goals established by the United Nations are as follows as we relate to our fyp:

1. **SDG-8: Decent Work and Economic Growth:** Strengthening cybersecurity with AI algorithms fosters growth in the sector, ensuring job security and stability for cybersecurity professionals.
2. **SDG-9: Industry, Innovation, and Infrastructure:** CADM contributes to SDG-9 by innovation.

1.8. Structure of Thesis

The structure of this thesis is described in the following table.

Chapter	Description
Abstract	A concise summary of the research objectives, methodology, key findings, and conclusions.
Chapter 1: Introduction	Provides an overview of the research topic, problem statement, proposed solution, working principles, objectives, scope, deliverables, relevant sustainable development goals, and the structure of the thesis.

Chapter 2: Literature Review	Reviews existing literature and background information on the industry, existing solutions and their drawbacks, and a detailed feature analysis of CADM.
Chapter 3: Experimental Lab Setup and Human Interfaces	Describes the virtual lab setup for CADM, user interfaces including login/authentication, main interface, anomaly detection module, and creation of customized dashboards.
Chapter 4: System Features	Details the key features of the system such as anomaly detection, customizable dashboards for user interaction, and the process of adding nodes/endpoints.
Chapter 5: Nonfunctional Requirements	Outlines the nonfunctional requirements including performance, safety, security, software quality attributes, business rules, and user roles and functions.
Chapter 6: System Diagrams	Provides system context diagrams, identifying external entities and data flows.
Chapter 7: Data Diagram	Illustrates data flow diagrams, detailed data descriptions, major data or system entities, and a data dictionary.
Chapter 8: Component Design	Includes component diagrams, detailed use case diagrams, activity diagrams, and sequence diagrams, illustrating the system's structural and behavioral aspects.
Chapter 9: Proofs of Concepts	Demonstrates practical applications through attack emulations, including failed login anomalies, Linux resource utilization anomalies, and HTTP response code anomalies.

CHAPTER 2: LITERATURE REVIEW

A new product is launched by modifying and enhancing the features of previously launched similar products. Literature review is an important step for development of an idea to a new product. Likewise, for the development of a product, a detailed study regarding all similar projects is compulsory. Our research is divided into the following points.

- Industrial Background
- Existing solutions and their drawbacks
- Research Papers

2.1. Industrial background

In this cyber era, one of the major challenges faced globally is the prevalence of cyber threats, which can lead to significant disruptions and losses due to our heavy reliance on technology. Traditional measures like firewalls and basic antivirus software are no longer sufficient against sophisticated cybercrime activities such as hacking and identity theft.

Our software, CADM, has the main aim of providing deep and nearly real-time log analysis from diverse endpoints in a bid to enhance cybersecurity defenses by minimizing False Positives and False Negatives: False Positives arise when actual normal traffic is misclassified as malicious, which is tremendously wasteful in resources and creates unnecessary alerts; at the same time, False Negatives are missed genuine threats, not recognized, thereby letting the malicious activities go through, as they remain undetected.

Manually identifying cyber anomalies is time-consuming and prone to human error, possibly missing key threats. CADM applies machine learning algorithms that automatically find real anomalies in the heap of log data with great accuracy and promptness of threat identification. By leveraging ML, CADM improves cybersecurity effectiveness through better resource utilization and response efforts to meet emerging cyber threats.

2.2. Existing Solutions and their drawbacks

Solution	Pros	Cons
Graylog	- Open-source platform for log analysis and management - Scalable, secure, and easy-to-use - Comprehensive features for log analysis, visualization, and reporting	- Requires setup and configuration - May require additional resources for maintenance and support - Primarily focused on monitoring IT infrastructure, log analysis is a secondary feature
Nagios	- Log analysis and management platform with real-time visibility into system activity, performance, and security	- Configuration can be complex - Requires expertise to utilize effectively - Setup and configuration may be involved
Logalyze	- Flexible and scalable data collection and logging platform - Collects and routes log data from various sources to multiple destinations	- May require training to fully utilize - Requires knowledge of configuration and plugin system for optimal use

Fluentd	- Analyzes logs and detects genuine anomalies, reducing false positives and negatives	- Setting up and maintaining plugins and integrations can be complex - Requires significant resources for implementation and maintenance
CADM	- Reduces SOC analyst time spent on anomaly detection tasks - Customizable dashboard for intuitive visualization of log data - Utilizes machine learning models for accurate anomaly detection - Reduces SOC analyst time spent on anomaly detection tasks	- Complex plugin and integration setup for detecting different types of anomalies - Requires significant resources for implementation and maintenance

Table 1: Existing Solutions and CADM.

2.3. CADM and its Feature Analysis.

2.3.1. Anomaly Detection:

The most important feature is focused on applying Machine Learning techniques for detecting anomalies within the logs.

2.3.2. Real-time Alerts:

The system is designed to alert the user whenever there is an indication of abnormal activity.

2.3.3. User-Friendly Dashboard:

The final deliverable of the project is an interactive visual dashboard seamlessly integrated into the Wazuh. This dashboard provides clear visualizations and an intuitive interface for users to monitor and comprehend anomalies such as anomalous file creations, suspicious login patterns, and network activity anomalies.

2.3.4. Integration with Wazuh:

The software operates using the Wazuh, leveraging Elasticsearch for data storage and retrieval, Logstash for log data processing, and Kibana for data visualization. This integration enhances the overall capability of the system to collect, analyze, and visualize data in real-time.

2.3.5. Log processing:

a. The logs being generated by any node in the system require some processing to make them able to be stored, displayed, monitored, and analyzed. The logs undergo several processes which mainly include but are not limited to:
b. Parsing
c. Formatting
d. Extraction
e. Sequencing

2.3.6. Visualization

All the logs and other useful information being fetched from the whole network will be displayed on the User Interface for the system. This part is also very important because it keeps the analyst aware of the activities and events inside the system.

2.3.7. Generation of Alerts:

This functionality performs a vital role which consists of two steps:

- **Policy defining**: Policy would be defined for the alert generation.
- **Mapping of rules**: All activities are being mapped with the defined rules in the policies.

Whenever there is an activity that is against the defined policy an alert is generated so that the admin can take steps to avoid any damage that may happen to the system.

CHAPTER 3: EXPERIMENTAL LAB SETUP AND HUMAN INTERFACES

This section outlines the whole virtual lab setup for the CADM and its human interfaces for various functions.

3.1. Virtual Lab Setup for CADM in Virtual Box

All the components of this FYP are setup inside Virtual Box, as shown in the following screenshot.

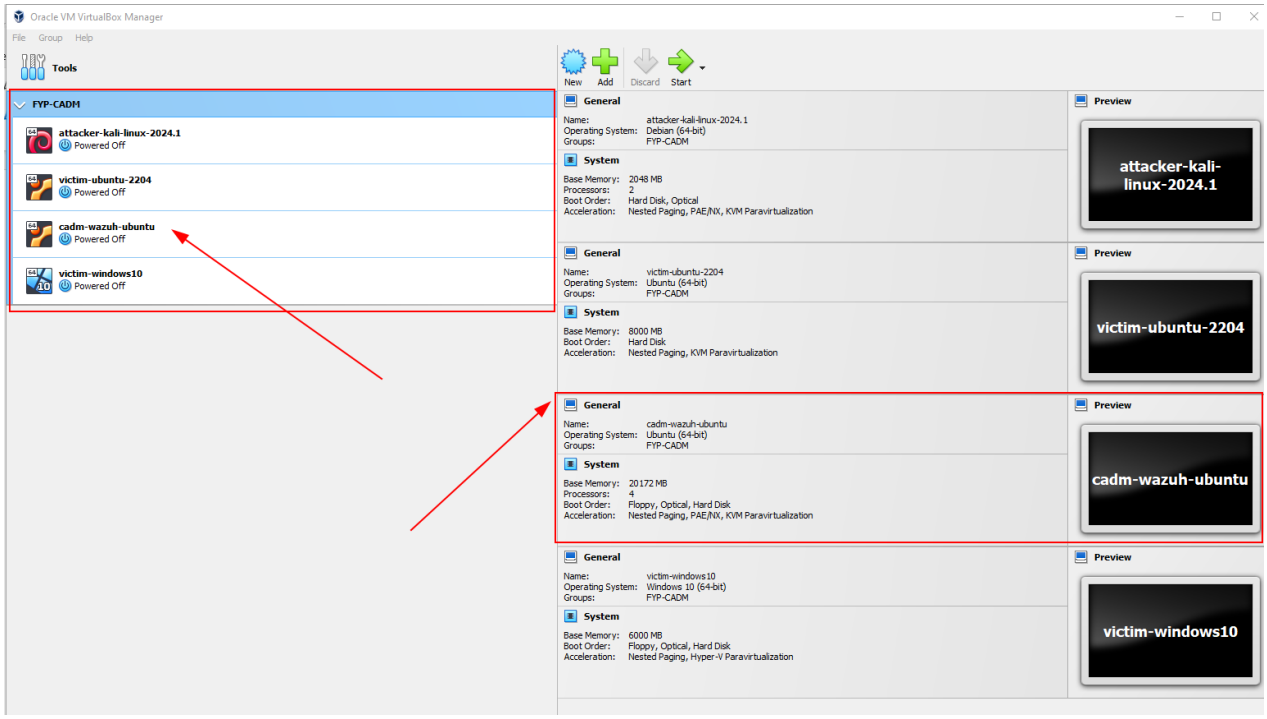


Figure 1 : Experimental Setup in Virtual Box for CADM's testing.

The four machines are further described in the following table.

Table 2 : Description of endpoint machines in lab setup of CADM

Name	Description

attacker-kali-linux	A Kali Linux machines that will act as the attacker machine. It will be used to perform simulated attacks using various tools.
victim-ubuntu-2204	Ubuntu (Linux) based machine that will act as the victim machine receiving attacks from the attacker machine. It has been made intentionally vulnerable by installing vulnerable apps like DVWA and other misconfigurations. A log forwarding CADM agent has been installed on this endpoint that will send logs generated from various simulated attacks, to CADM.
victim-windows10	A Windows 10 machine acting as another victim. This machine will act like victim-ubuntu-2204 machine, but for different use case.
cadm-wazuh-ubuntu	Based on Ubuntu 22.04, this is the core machine of the whole lab setup. CADM is installed on this machine. The agents installed on the victim machines forwards logs to this machine, and here they are analyzed for indications of attacks and cyber anomalies.

The host machine on which virtual lab is set up inside Virtual Box, is a Windows 10 machine with the following specifications.

Table 3 : Host Machine Specification on which whole CADM is deployed.

Attribute	Value
OS Name	Microsoft Windows 10 Pro
Version	10.0.19045 Build 19045
System Manufacturer	Dell Inc.
System Model	Precision 5530
System Type	x64-based PC

Processor	Intel(R) Core (TM) i7-8850H CPU @ 2.60GHz, 2592 MHz, 6 Core(s), 12 Logical Processor(s)
RAM	64 GB
Storage	1 TB SSD

This FYP has been practically tested in this system.

3.2. Login/Authentication Page

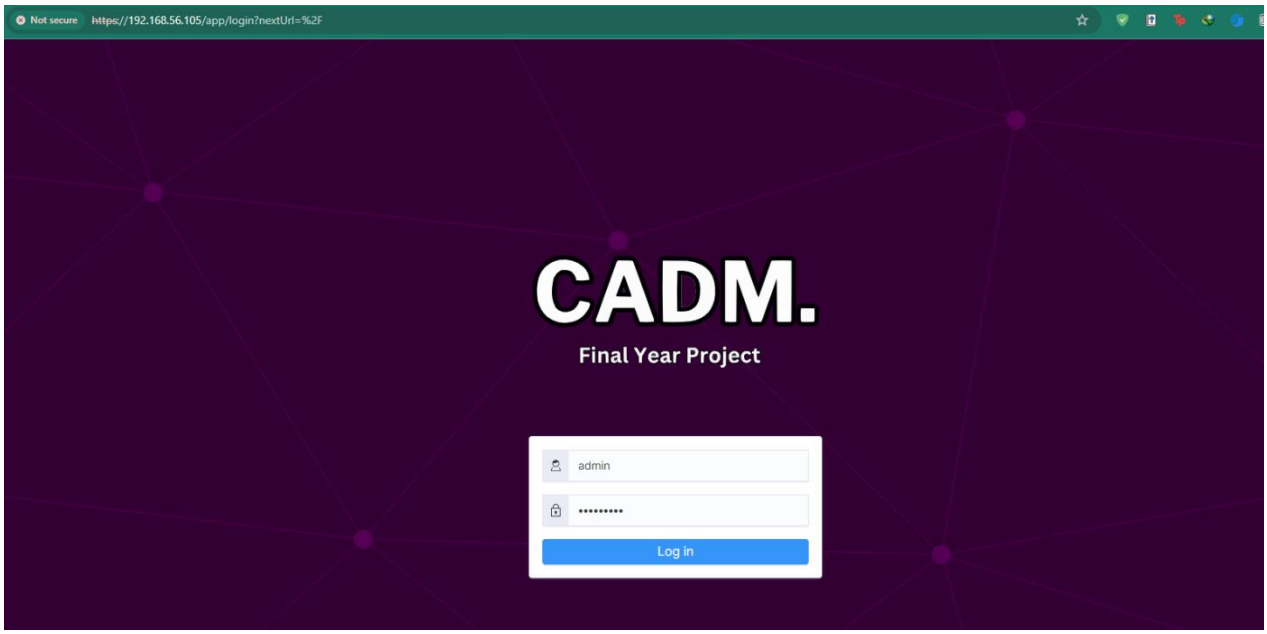


Figure 2 : Login/Authentication Page of CADM

3.3. Home Page of CADM

After login, the home page is displayed. This page shows various modules of the system. Information about the agents is also shown here. As shown in the following screenshot, currently two agents are actively connected, and logs are being forwarded from them to CADM.

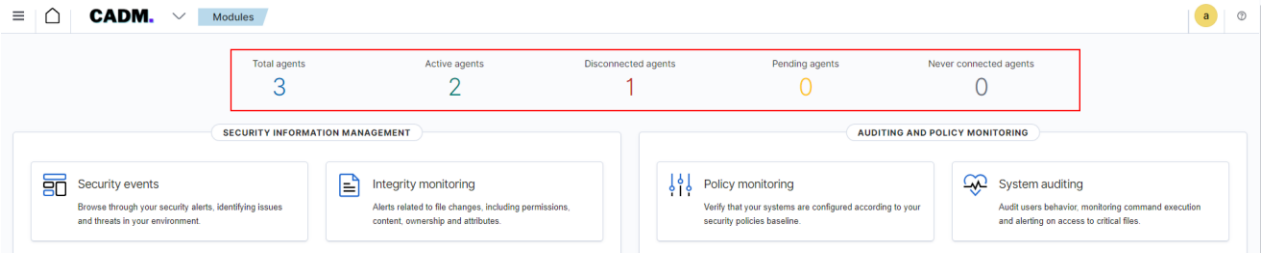


Figure 3 : Home Page of CADM UI.

3.4. Opening Anomaly Detection Module

The anomaly detection module can be opened as shown in the following figure. From the navigation panel on the left side, in the OpenSearch Plugins section, click on Anomaly Detection.

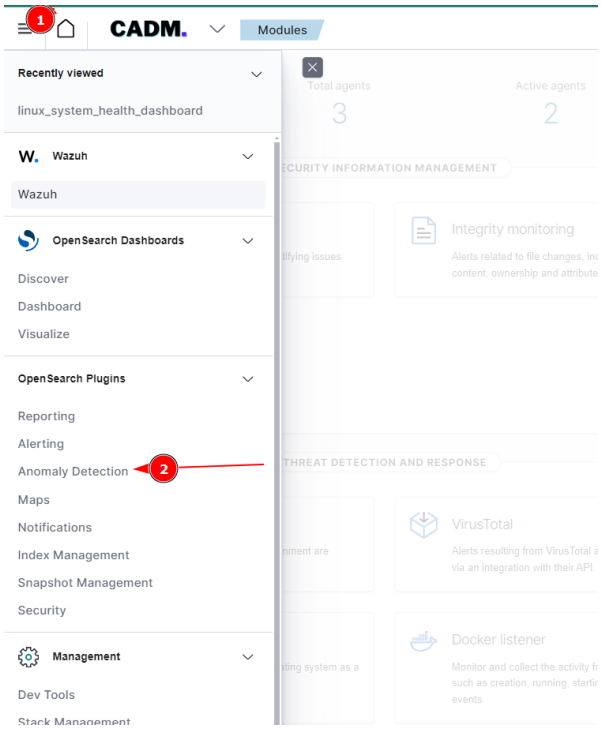


Figure 4 : Steps to open Anomaly Detection Module in CADM

In the following Figure, the main dashboard can be seen. On the left side, we can see already deployed detectors or create a new detector.

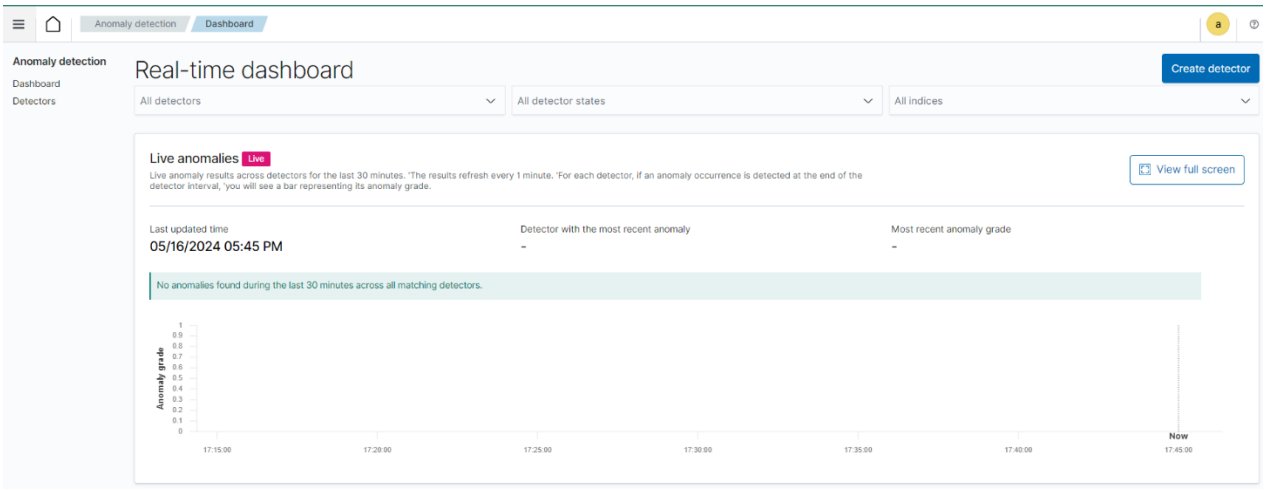


Figure 5 : Dashboard of Anomaly Detector Module in CADM

3.5. Creating a new Anomaly Detector in CADM

CADM gives an easy way to create anomaly detectors for specific use cases. Before creating a detector, you need to already have setup log forwarding according to the use case you are deploying. The following figures show the 4 steps to create a new detector.

1. Define the detector.
2. Configure the detector.
3. Preview your detector.
4. View results.

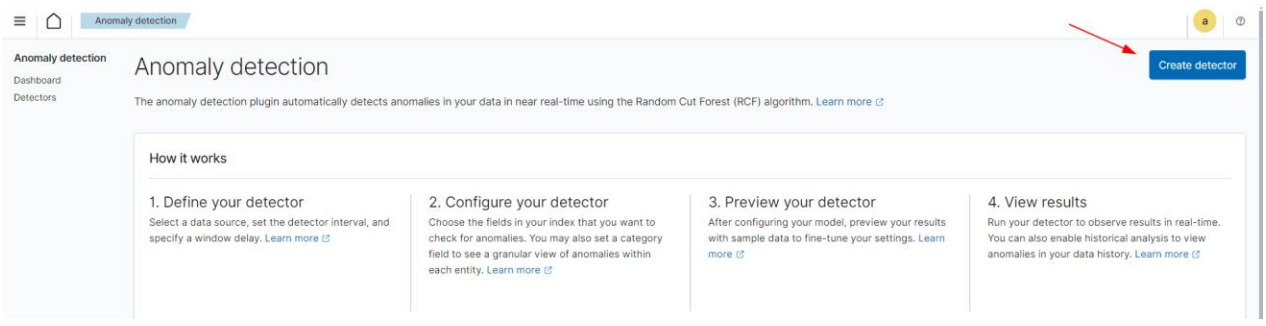


Figure 6 : Creating a new detector for a specific use case.

3.6. UI for detecting other anomalies (cyber-attacks) in CADM.

Other types of anomalies, for example SQLi injection indications, brute force attacks, cobalt strike beacon detection and so on, can also be detected in CADM. The following figure shows a detection of SSH login anomaly – a possible brute force attack.

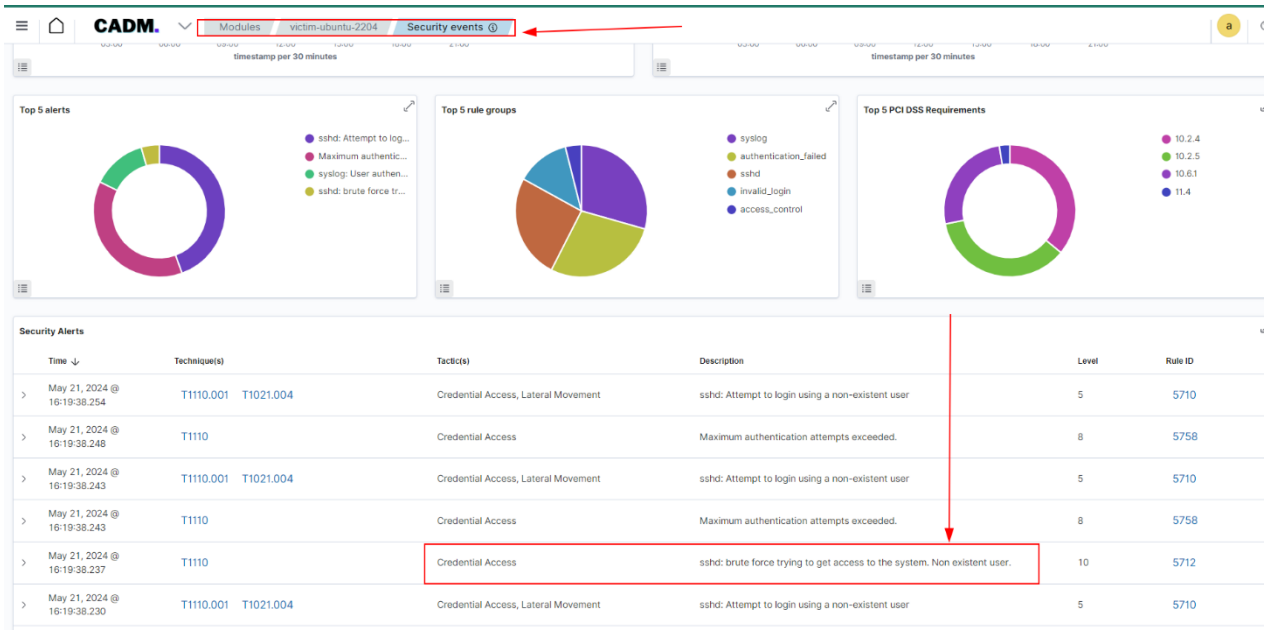


Figure 7 : Detection of attacks/anomalies in streaming logs

This figure shows how the interface of detecting various other anomalies/attacks can be seen in the alerts.

3.7. Customized Dashboards

In CADM, appealing and easy to understand visualization has been made which are then assembled in a dashboard, as shown in the following figure. This figure shows the anomalies in Linux resource usage. Indicated by peaks in the graphs.

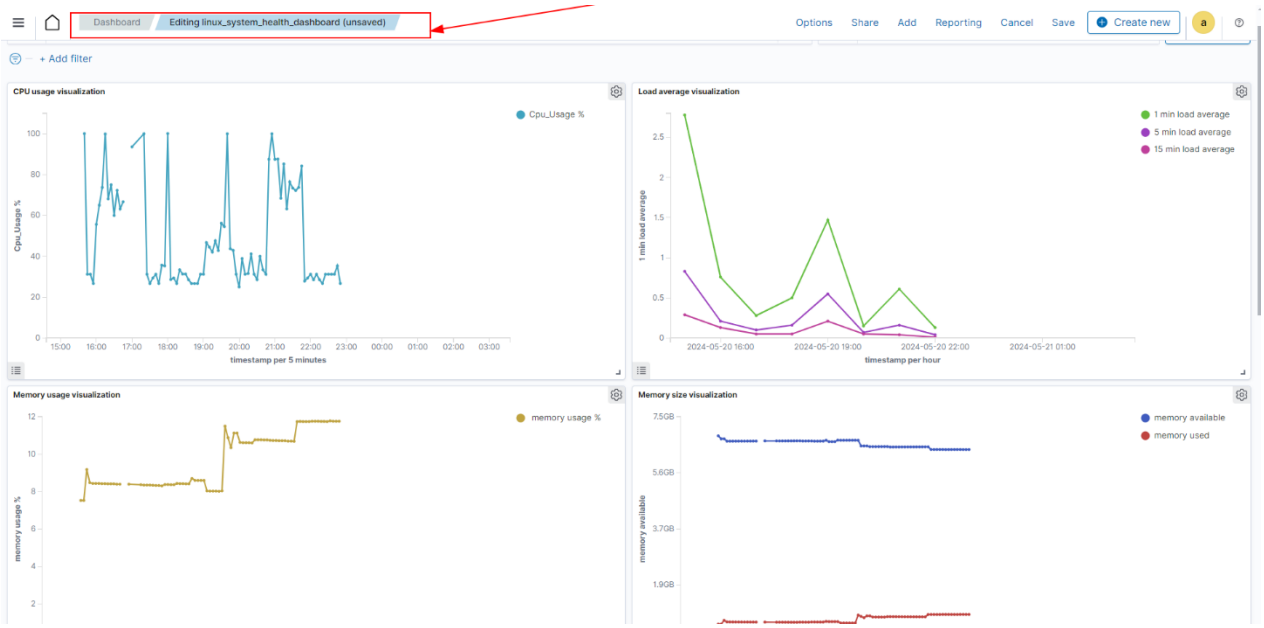
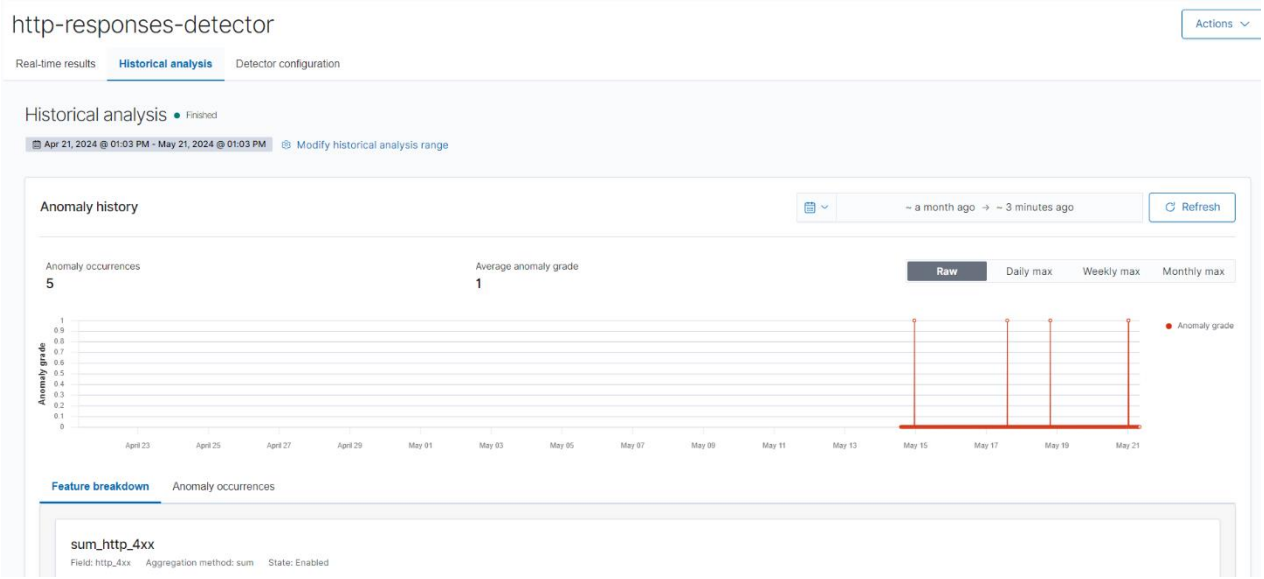


Figure 8 : Customized dashboard for some use cases.

3.8. Http Response Anomaly – An Example

For each use case, a dashboard like following is made, showing anomalies in a timeline wise fashion. For example the following figure shows large number of anomalous http 5xx responses at times where the peaks are high on the timeline.



CHAPTER 4: SYSTEM FEATURES

4.1. Anomaly Detection

Description and Priority	This feature focuses on the real-time detection of anomalies within log data, prioritized as **High** due to its critical role in enhancing cybersecurity measures.
Stimulus/Response Sequences	• *Stimulus:* New log data is received in real-time. • *Response:* The system applies machine learning algorithms to analyze the log data for anomalies. If anomalies are detected, real-time alerts are generated, and threat levels are assessed.
Functional Requirements	**REQ-1: Real-time Log Analysis** • *Description:* The system must continuously analyze incoming log data in real-time. • *Response:* Swift identification of anomalies for further assessment. **REQ-2: Machine Learning Integration** • *Description:* Integrate machine learning models for advanced anomaly detection. • *Response:* Enhance the system's ability to discern patterns and deviations in log data.

	REQ-3: Real-time Alerting • *Description:* Generate real-time alerts when anomalies are detected. • *Response:* Immediate notification to users for prompt threat response

4.1.1. Customizable Dashboards for User Interaction

Description and Priority	This feature enables users to customize dashboards for visualizing anomalies, prioritized as Medium to provide flexibility for varying user preferences.
Stimulus/Response Sequences	• *Stimulus:* User interacts with the system through the web browser interface. • *Response:* The system allows users to customize dashboards, choosing specific visualizations for anomaly monitoring.
Functional Requirements	**REQ-4: Dashboard Customization** • *Description:* Provide users with the ability to customize dashboards. • *Response:* Enable users to tailor visualizations based on individual cybersecurity monitoring needs. **REQ-5: Web Browser Interface** • *Description:* Ensure accessibility through web browsers.

	• *Response:* Facilitate user interaction through a platform-independent web-based interface.

4.1.2. Adding a Node/Endpoint

Description and Priority	• This high-priority feature enables the addition of a node by installing the agent on the node, allowing it to connect to the central server for monitoring. Logs will be transferred and monitored by the admin.
Stimulus/Response Sequences	• *Stimulus:* The node is connected into the network. • *Response:* Node-server communication/exchange is initiated.
Functional Requirements	**REQ-6: Add a new Node/Agent to monitor.** • *Description*: Admins must be able to add a new node to the CADM by installing an agent on the end point.

CHAPTER 5: NONFUNCTIONAL REQUIREMENTS

5.1. Performance Requirements

- **Real-time Log Analysis:** The system must analyze incoming log data in real-time, with a response time of less than one second to ensure timely detection of anomalies.
- **Scalability:** The system should be capable of handling log data from a minimum of 1000 endpoints simultaneously, with a projected growth of 20% annually.

5.2. Safety Requirements

- **Data integrity:** The system needs to preserve log data integrity in the process of collection, processing, and storage against possible data corruption.
- **User Authentication:** Preventing unauthorized entry, the system should authenticate users prior to any access. Multi-factor authentication should be used for admin accounts.

5.3. Security Requirements

- **Encryption:** All endpoint nodes, the central server, and external interfaces should engage in communications encrypted to industry-accepted protocol standards, to safeguard against data breaches.
- **Access Control:** Proper role-based access control must be implemented to limit user access to data according to the user's assigned role, thereby ensuring that sensitive information is accessible only to authorized personnel.

5.4. Software Quality Attributes

- **Reliability:** The system must have at least 99% uptime, ensuring continuous availability for real-time monitoring.
- **Maintainability:** Codebase should be well-documented, and updates or modifications should be easily implementable without disrupting system functionality.
- **Usability:** The user interface should be intuitive, requiring minimal training for users to effectively navigate and customize dashboard

5.5. Business Rules

5.5.1. User Roles and Functions:

- *Security Analysts / Incident Responders:* Have full access to real-time log analysis and anomaly detection.
- *IT Administrators:* Responsible for system administration and configuration.
- *Executives/Management:* Receive summarized reports and high-level insights.

5.5.2. Node Addition:

- Only administrators have the authority to add new nodes for monitoring.
- Nodes must undergo a signup process for network integration.

CHAPTER 6: SYSTEM DIAGRAMS

6.1. Context Diagram

Consider the context diagram of CADM in Figure 01. The objective of this context diagram is to give the overview of our proposed system, CADM. It gives the high-level view of the system and its relationships with the outside world.

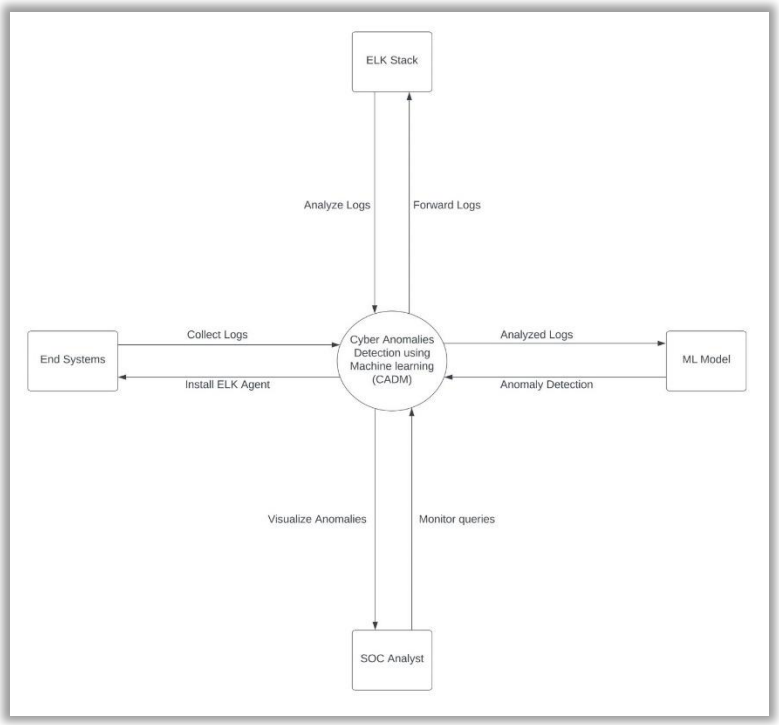


Figure 9: Context Diagram of CADM

Central to our system is the Centralized Anomaly Detection Module.

6.1.1. External entities comprise:

Entity	Description
CADM	Centralized Anomaly Detection Module, the core product of the system.
End Systems	Devices designated for protection, responsible for generating logs.
ML Model	Applies advanced anomaly detection techniques to determine the anomaly status of log entries.
SOC Analyst	Human component using the system's user interface to analyse and respond to potential security threats.

Table 4: External Entities of CADM

6.1.2. Data Flows:

Flow	Description
End Systems to CADM	Initiates data flow by transmitting 'LOGS' to CADM. Receives a confirmation message about ELK agent installation. The ELK/Wazuh agent establishes communication with CADM.
SOC Analyst to CADM	Engages with CADM by formulating queries and configuring visualization dashboards on the Kibana UI.
CADM to ML Model	Facilitates the flow of logs from CADM to the ML Model for analysis.
ML Model to CADM	Provides assessments indicating whether logs indicate anomalies.

Table 5 : Data flows of CADM

This context diagram describes how the primary entities and their interactions within the system are performed. It shows how the information flows seamlessly from an end system to CADM, the interactive nature of the SOC Analyst, and the involvement of the ML Model in detecting anomalies, challenges and optimize the system's performance over time.

CHAPTER 7: DATA DIAGRAM

7.1. Data Flow Diagram

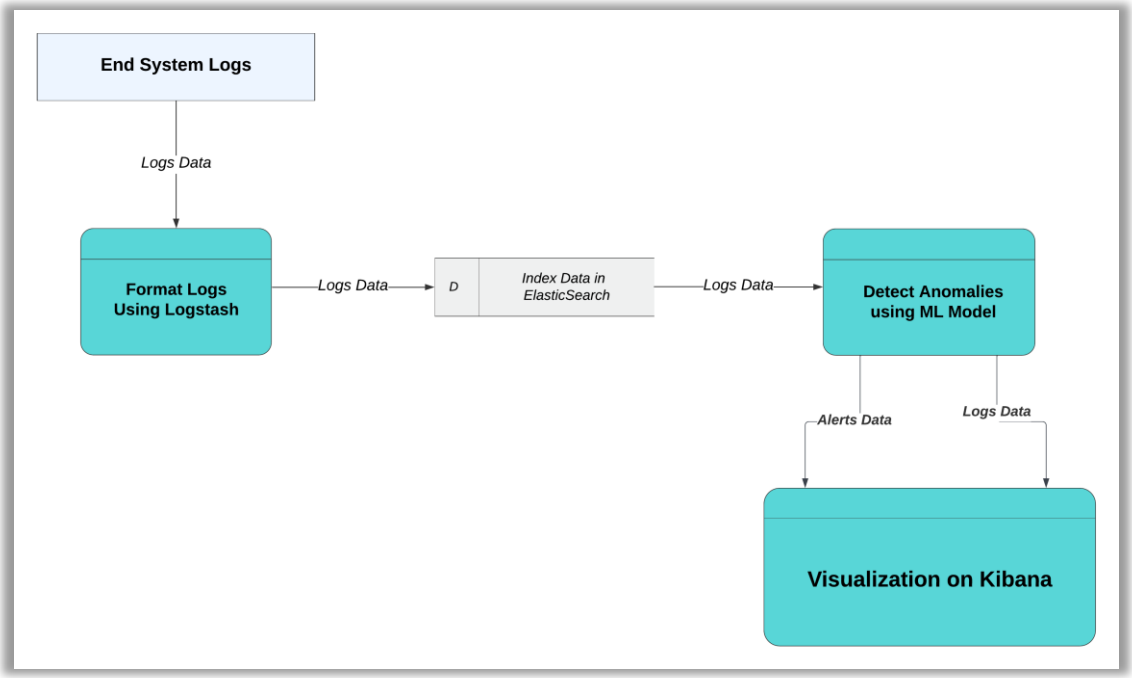


Figure 10 : Data Flow Diagram for CADM

7.2. Data Description

The information domain of the system involves log data, primarily in plain text files or CSV format. Logstash is employed for log parsing, using JSON formatting. These logs, after processing, are stored in Elasticsearch, a NoSQL database, forming the basis of the ELK Stack. Additionally, the log data is fed to Machine Learning (ML) models via JSON objects through API calls.

7.2.1. Major Data or System Entities:

7.2.2. Log Data:

- **Type**: Plain text files, CSV.
- **Storage**: Elasticsearch database.
- **Processing**: Handled by Logstash for parsing and formatting.
- **Organization**: Log entries are organized in Elasticsearch indices.

7.2.3. ML Model Input:

- **Type**: JSON objects.
- **Processing**: ML models are fed with log data through API calls.
- **Organization**: Input parameters structured as JSON objects.

7.3. Data Dictionary

7.3.1. Log Data:

Type: Structured data

Description: Information collected from various endpoints, including timestamp, source IP, destination IP, log type, etc.

7.3.2. User:

Type: String

Description: Individuals using the system, differentiated by roles such as Security Analyst, Security Administrator, Network Engineer, and Compliance Officer.

7.3.3. Node/Endpoint:

Type: String

Description: Represents a device or endpoint in the network that generates log data.

7.3.4. Machine Learning Model:

Type: Python Class

Description: Object encapsulating the machine learning algorithms used for anomaly detection.

CHAPTER 8: COMPONENT DESIGN

This section comprises the details of how the components of CADM work together, and how they are connected to each other, making the whole of CADM. The component diagram, class diagram, detailed use case diagram and all possible activity and sequences diagrams are presented and explained briefly.

8.1. Component Diagram

The following diagram shows the component of CADM.

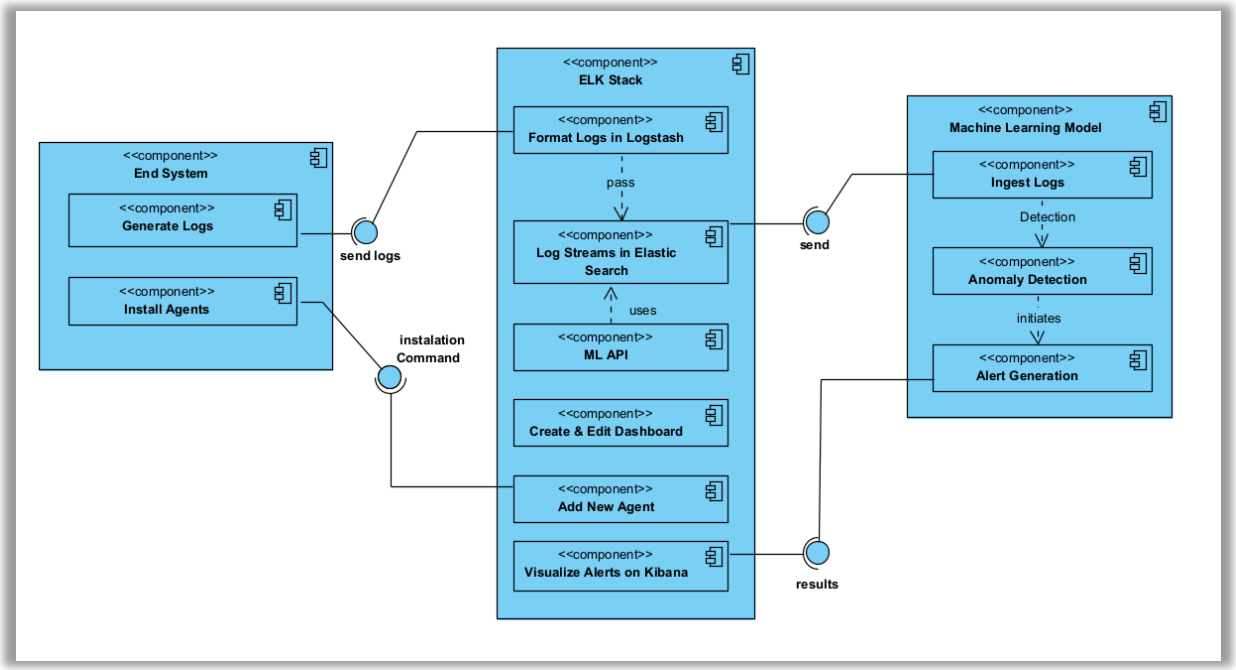


Figure 11 : Component Diagram of CADM

This diagram shows all the components CADM. The three fundamental components are End System, ELK Stack/Wazuh and ML Model.

8.2. Detailed Use Case Diagram

Consider the following Figure 3, which gives detailed uses cases in CADM. The actors are End Systems, SOC Analysts, and ML Engineers.

- ***The End Systems actor*** are responsible for generating logs that will be sent via log agents to CADM. The end systems are also responsible for ingestion of logs in Logstash.
- ***ML Engineer actor*** is responsible for ingesting logs in Elasticsearch database, that will be used to train or refine the ML model that detect anomalies.
- ***The SOC Analys actor*** can interact with CADM by using the trained ML models and adjusting them for steamily coming logs from end systems and visualizing the anomalies and alerts on the customized dashboards.

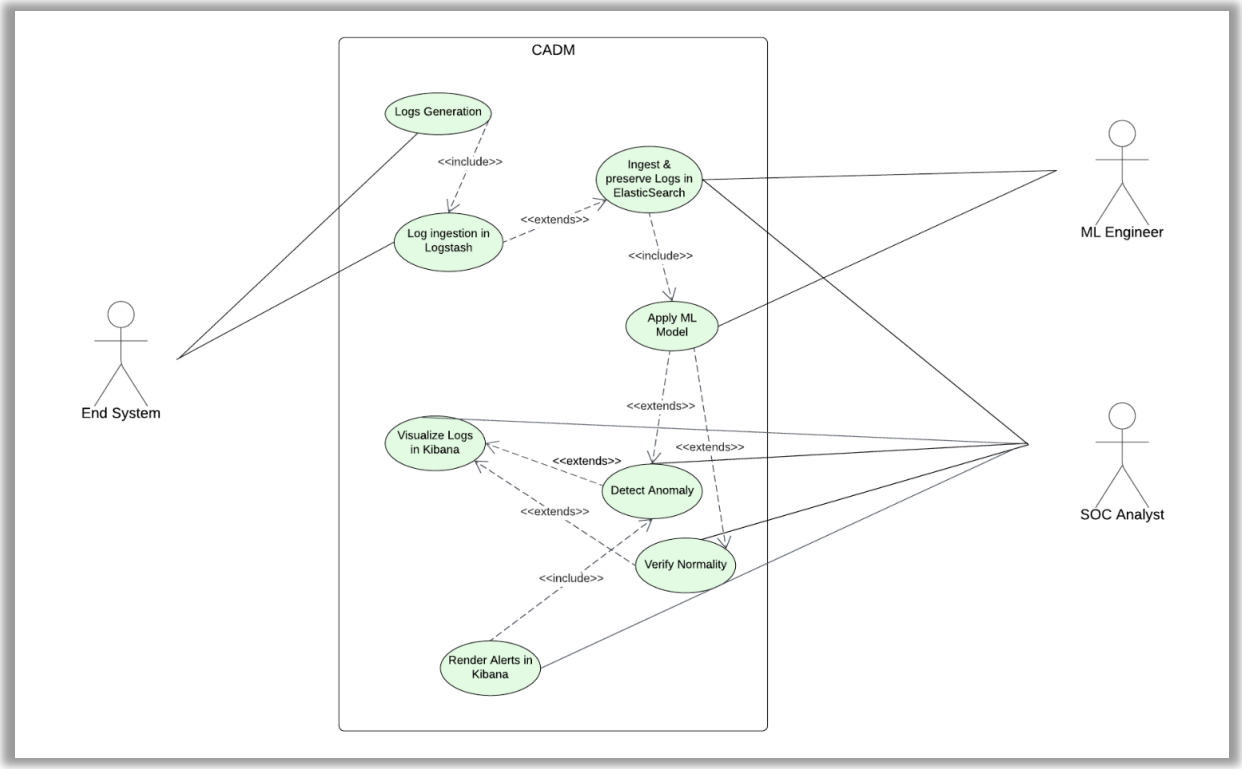


Figure 12 : Detailed Use Case Diagram of CADM

8.3. Activity Diagrams

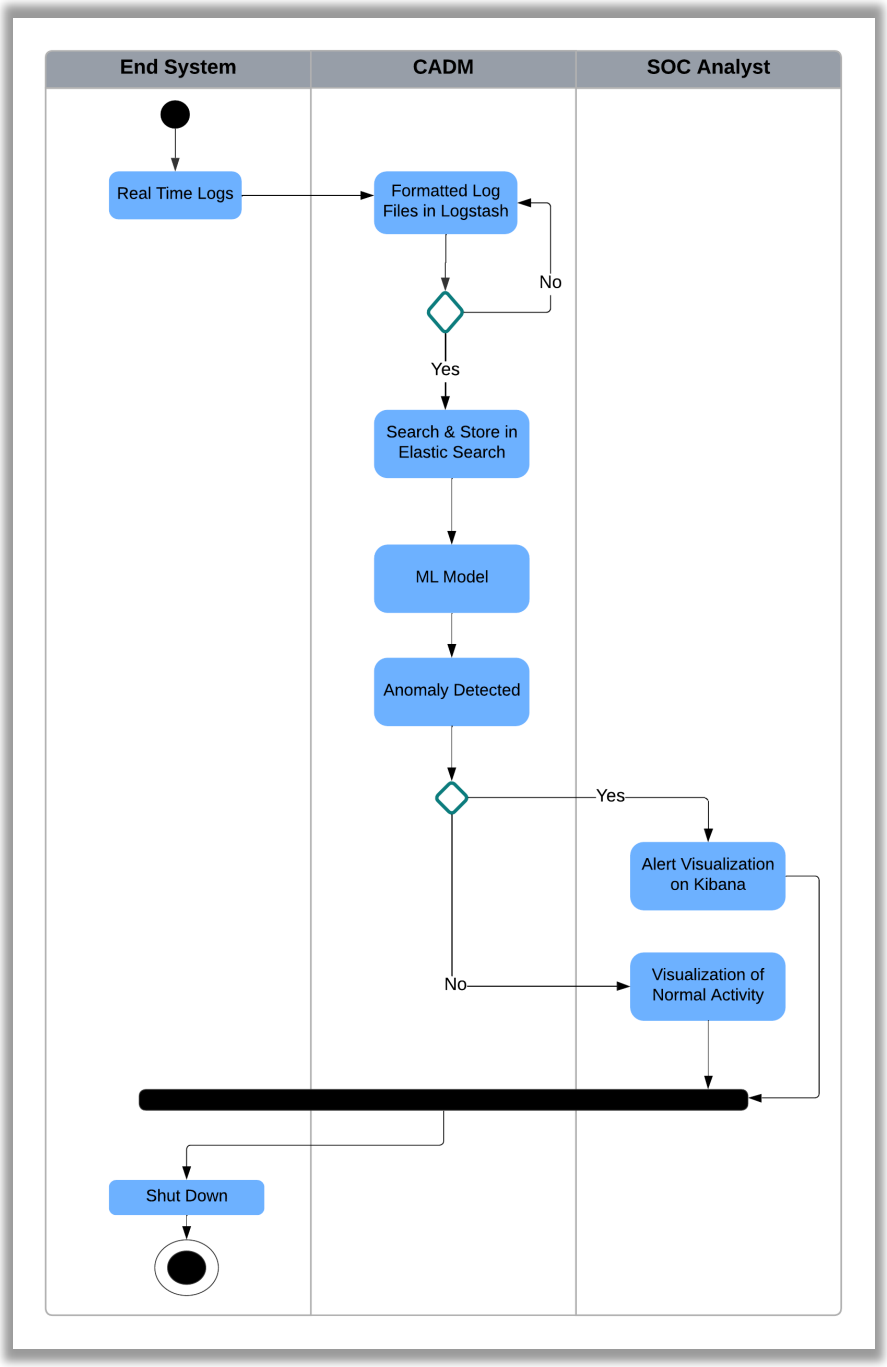


Figure 13 : Overall Activity Diagram of CADM

8.3.1. Overall Activity Diagram of CADM

Consider the overall activity diagram of CADM in Figure 7. It shows the overall working of the system.

Realtime logs are sent from End System to CADM. In CADM the logs are reformatted according to the need of ML model. These logs are the stored in Elasticsearch database. From there, they are forwarded to ML Model that has already been trained for detecting anomalies. If ML Model detects anomalies, it is visually alerted on the dashboard/panels/graphs, that are already created by user (SOC Analyst) according to specific type of anomaly. If no anomaly is detected, normal streams of logs are displayed in near-real-time in the user interface.

8.3.2. Detect Anomaly Activity Diagram

Consider the following activity diagram for detecting anomalies.

In the below diagram, the end systems forward the logs in real time to ML model. The ML model receives the logs in parsed form via Logstash. Then the model determines whether a log entry is anomalous or normal. If it is anomalous then an alert is displayed on the premade visualization in Kibana.

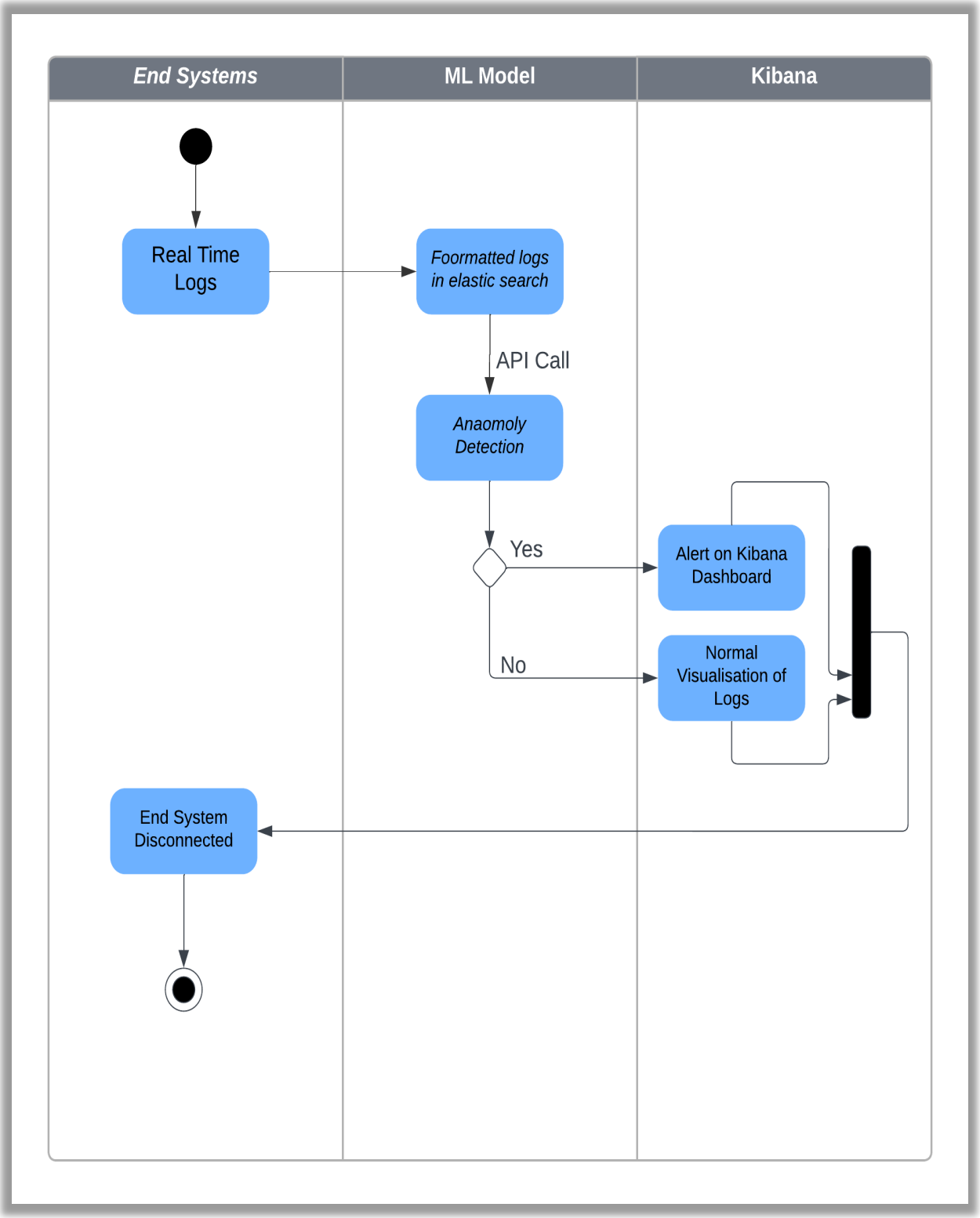


Figure 14 : Activity Diagram for Detecting Anomaly

8.3.3. Add Agent Activity Diagram

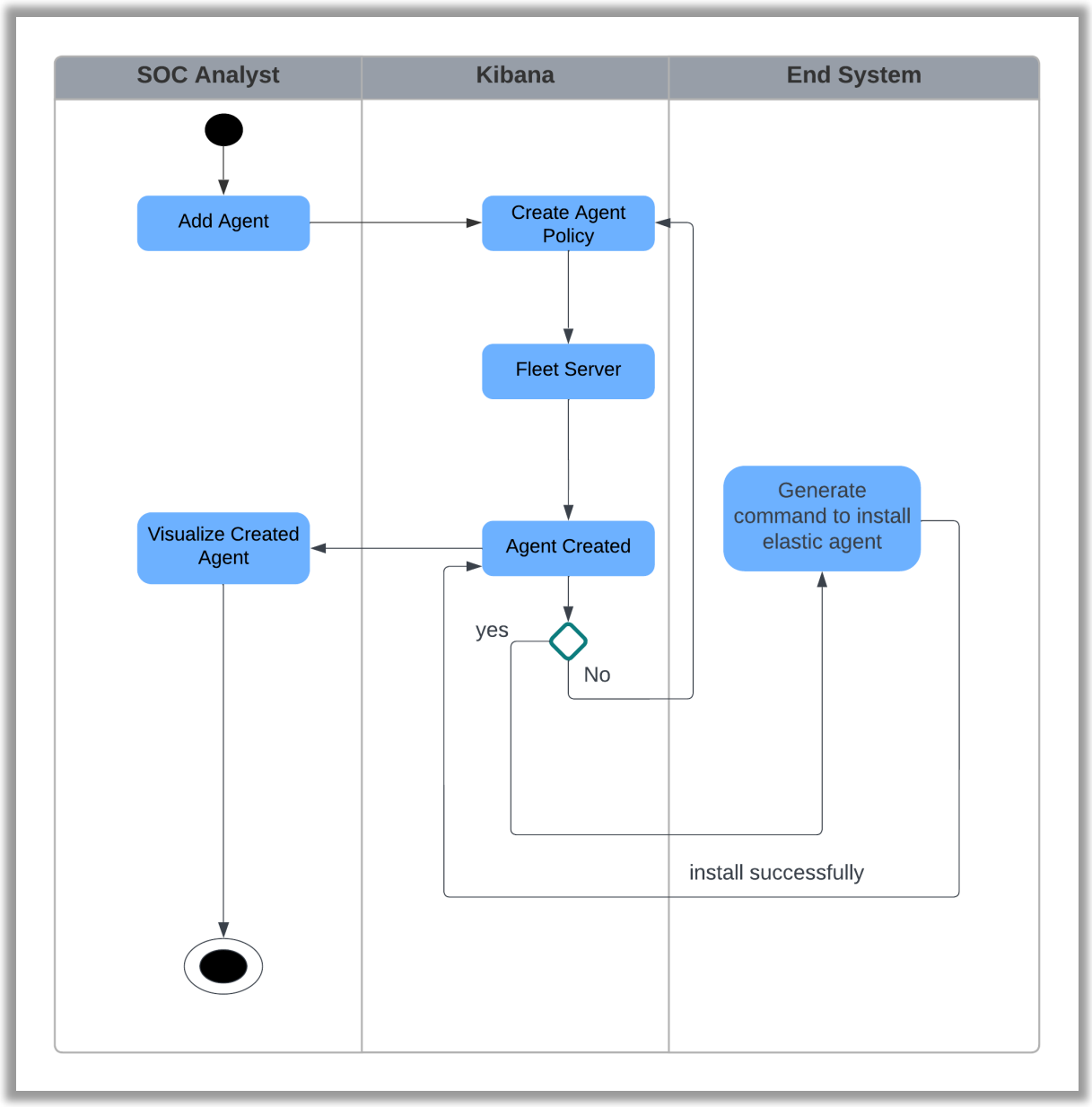


Figure 15 : Sequence Diagram for Adding Agent in CADM for logs forwarding.

8.3.4. Create Dashboard Activity Diagram

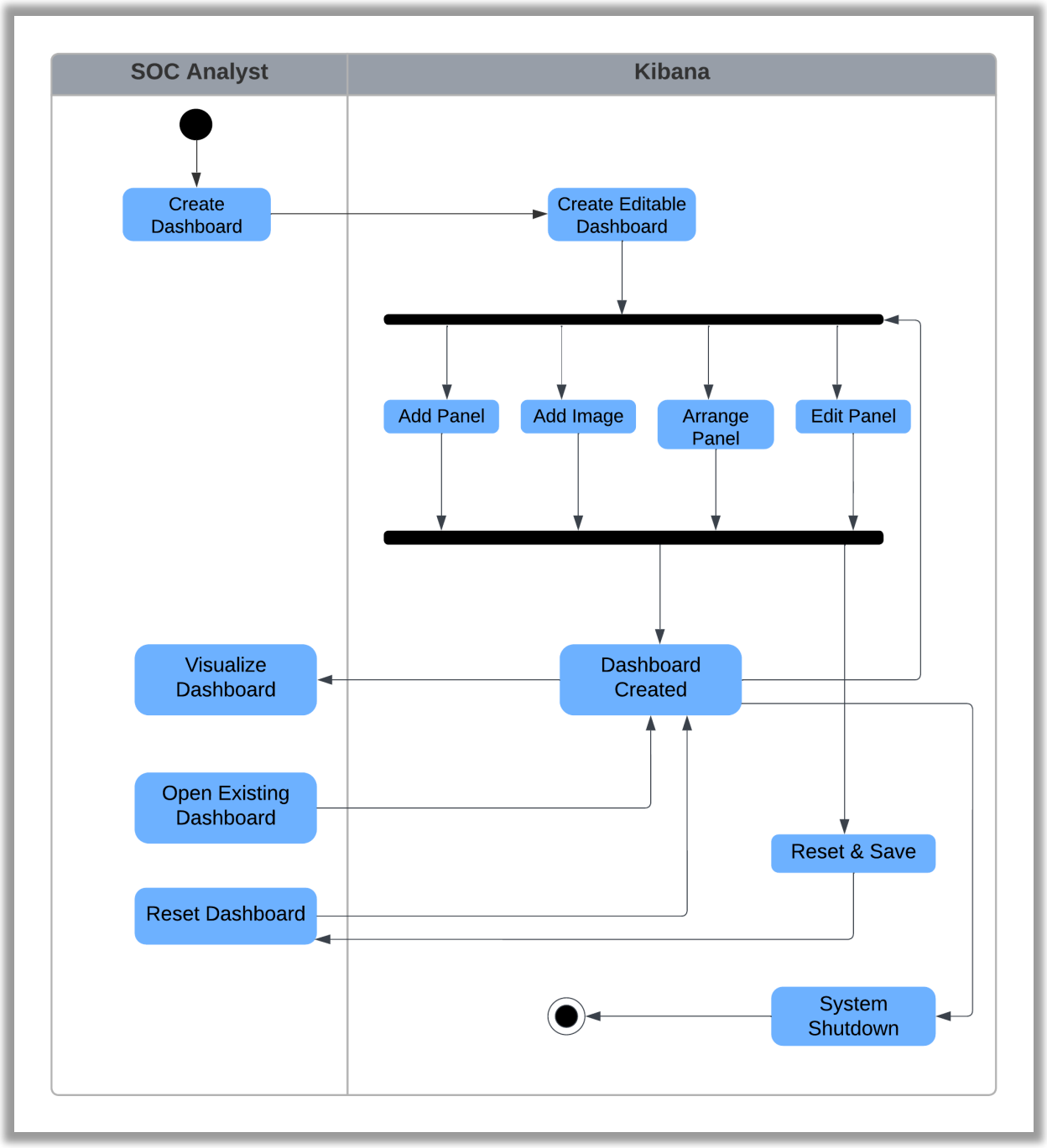


Figure 16 : Activity Diagram for Creating Dashboard in CADM

8.4. Sequence Diagrams

8.4.1. Detect Anomaly Sequence Diagram

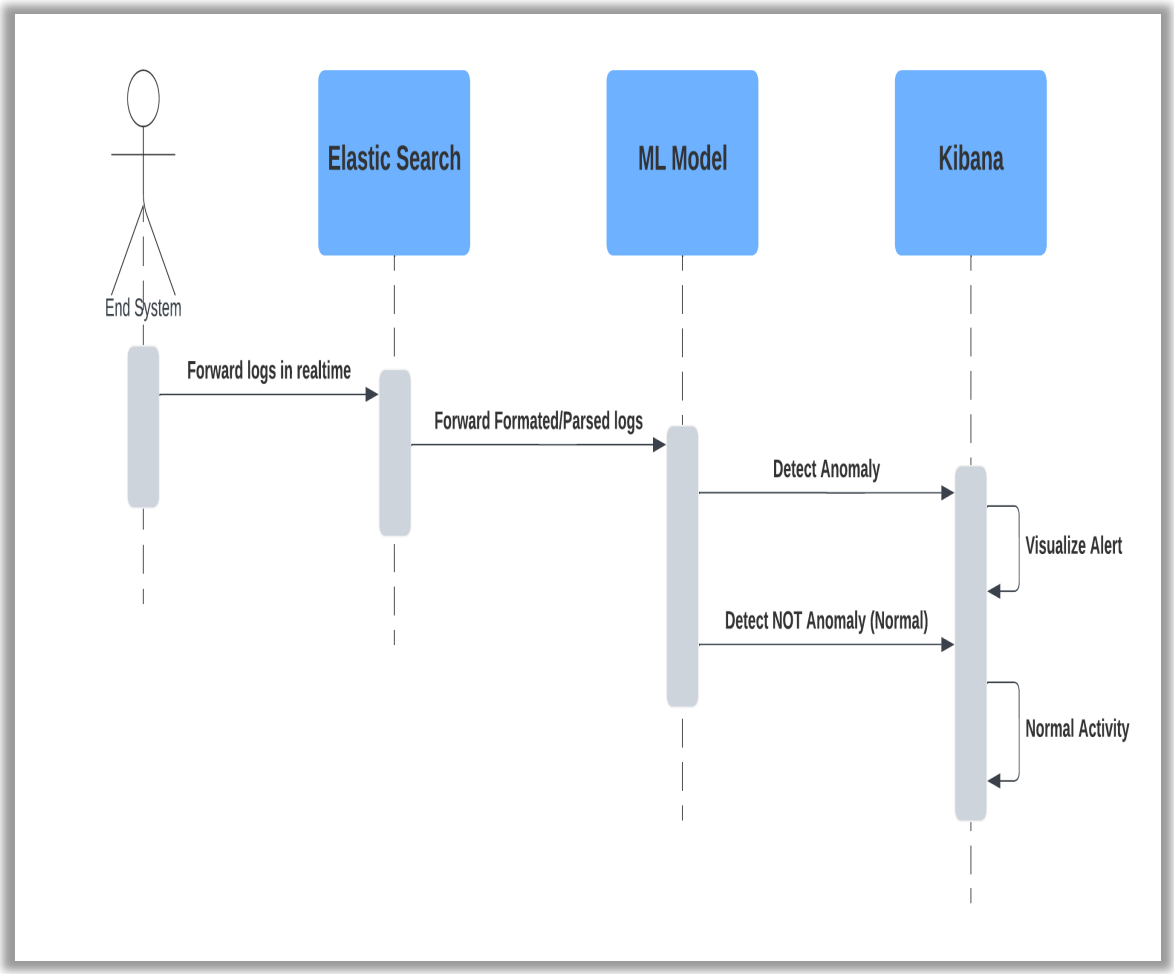


Figure 17 : Sequence Diagram for Detecting Anomaly in CADM

8.4.2. Add Agent Sequence Diagram

Consider the following sequence diagram for adding an agent for log forwarding. A policy has to be created for the agent, using the Fleet Server which is a centralized manager for all the agents in CADM. Then a command is issued to install agent on the end system which is run in PowerShell or a terminal. Then a message is shown on Kibana, showing that the agent has been successfully installed.

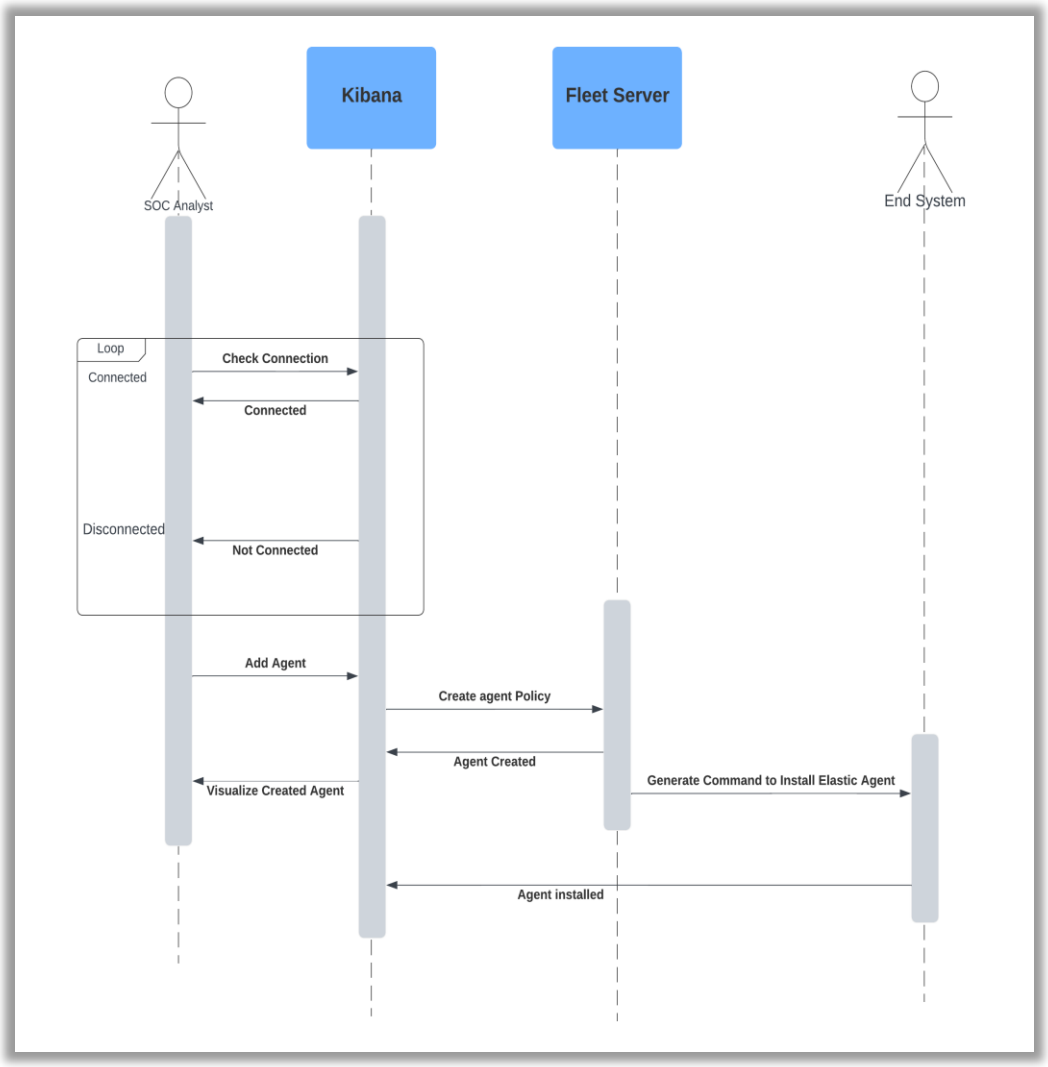


Figure 18 : Sequence Diagram for Add Agent

8.4.3. Sequence diagram for Creating Dashboard in CADM

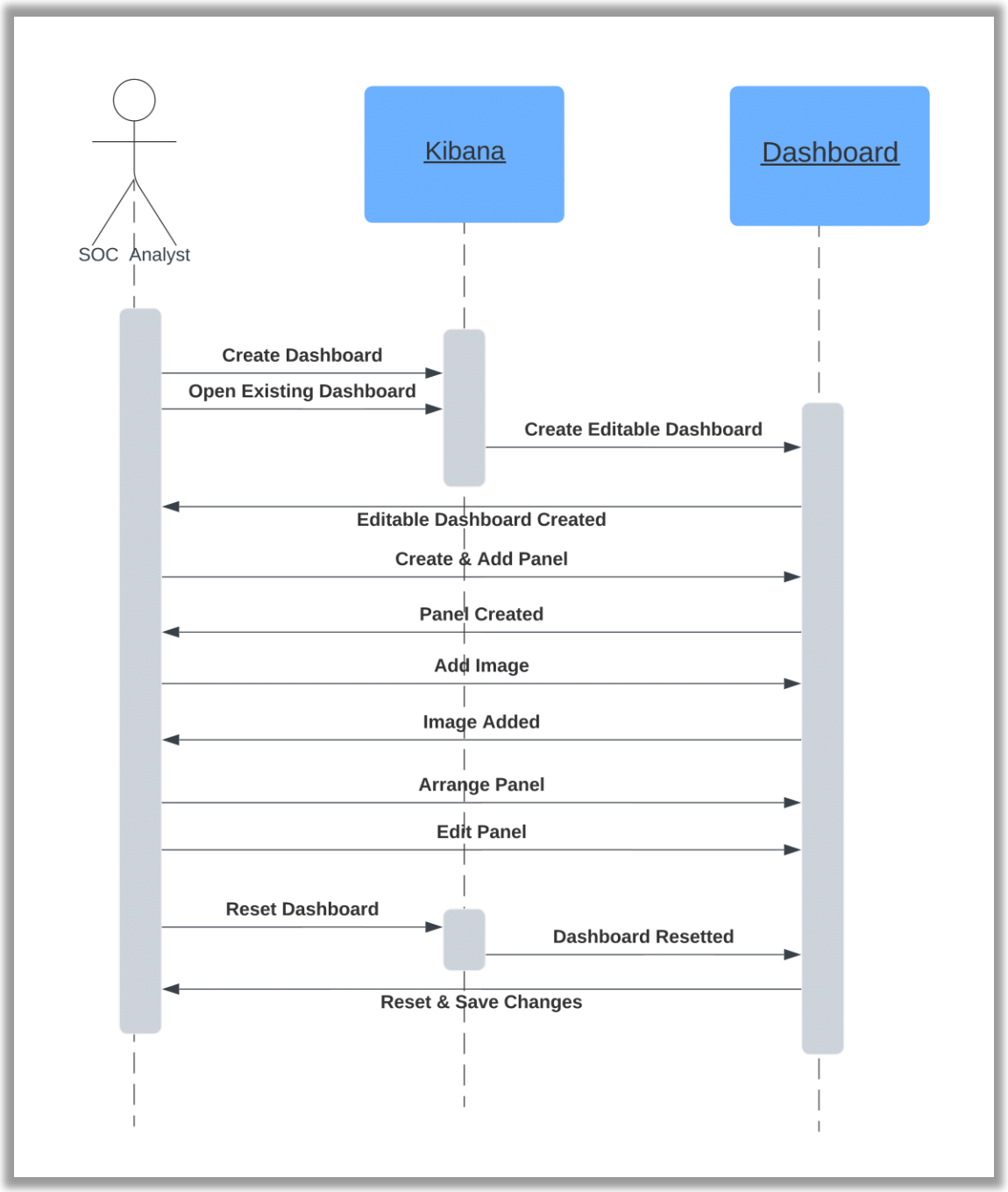


Figure 19 : Sequence Diagram for Creating Dashboard in CADM

CHAPTER 9: PROOFS OF CONCEPTS -

PRACTIAL DEMONSTRATION THROUGH ATTACK

EMULATIONS

This section gave a practical demo of the testing of various use cases implemented with CADM. The various attacks and anomalous situations are emulated and then observed and detected with CADM. In each of these use cases, there is some victim machine where a log forwarding agent is installed, and an attacker machine that attacks the victim machine. The CADM is deployed on a third machine, the monitor machine, which detects the attacks and anomalies in the logs coming from victim machine.

The deployed use cases are as outlined below, then explained in coming subsections.

9.1. Failed Login Anomaly

In this use case, the attacker (bottom terminal) performs an SSH brute force attack on the victim, using Hydra tool. The attack is detected in the CADM as shown in the following screenshot.

Table 6 : Failed Login Anomaly Machines and Command

Attacker	Attacker-kali – 192.168.56.102
Victim	Victim-ubuntu-2204 – ssh://192.168.56.101
Command to emulate attack from attacker side	hydra -v -L user.txt -p pass.txt 192.168.56.101 ssh -t 4

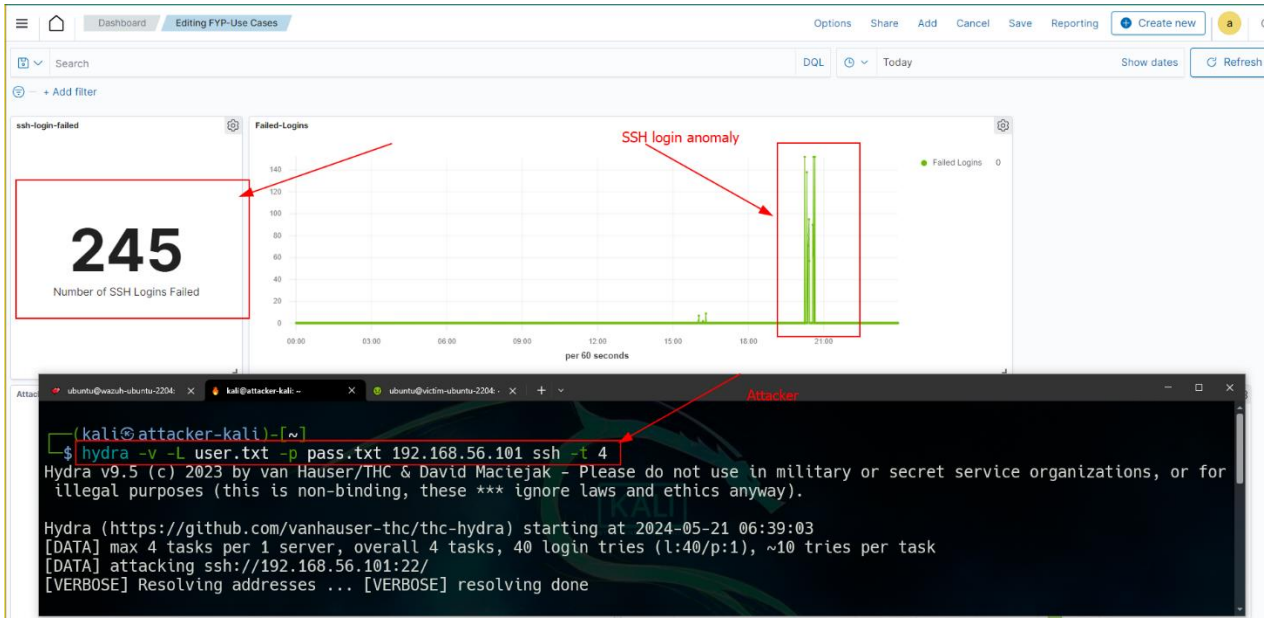


Figure 20 : Detected Failed Login Anomaly

In another dashboard, it can be visualized as follows:

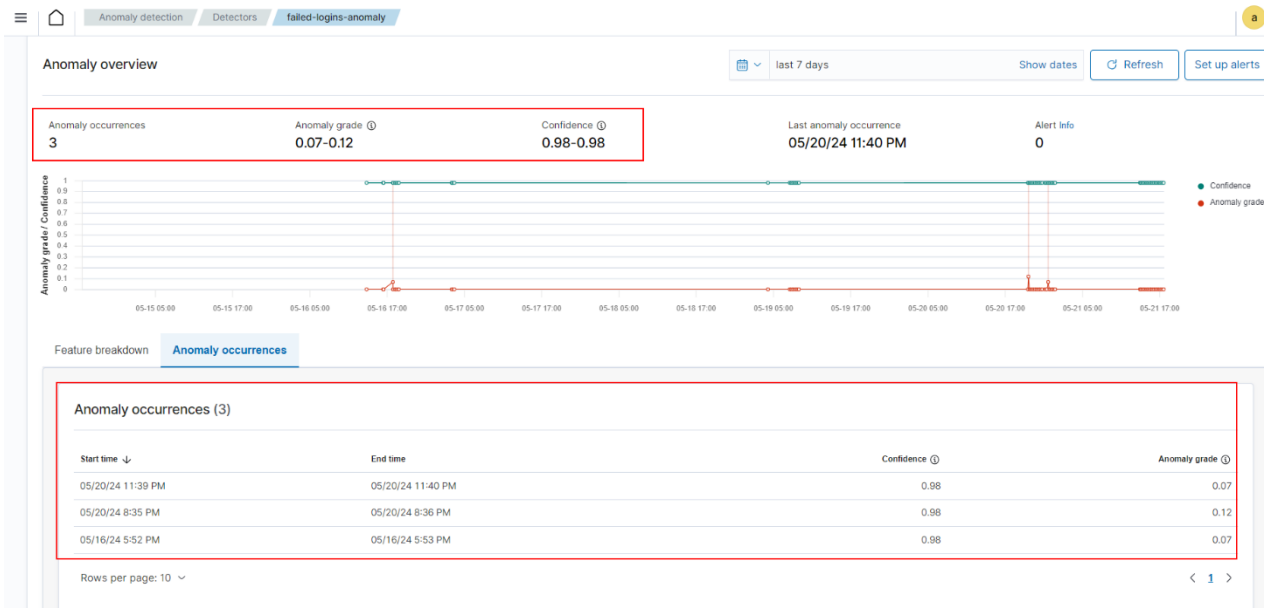


Figure 21: Failed Login Anomaly Dashboard

9.2. Linux resource utilization anomaly

In this use we detect anomalies in CPU and memory usage on a Linux (Ubuntu) endpoint. Ubuntu endpoint has to be already configured to to forward resource usage information to the Wazuh server.

Attacker	Victim itself. Stress testing is done to emulate anomalous resources usage in the machine.
Victim	Victim-ubuntu-2204 – 192.168.56.101
Command to emulate attack from attacker side	stress -c 4 -m 4 --vm-bytes 512M -t 160

Figure 22 : Linux Resource Utilization Anomaly Details

The stress utility must be pre-installed for this attack emulation. The results are shown as follows:

Figure 23: Detected Anomaly in Linux Resources Utilization

The overview includes the total number of anomaly occurrences, the overall Anomaly grade, confidence, date and time of the last anomaly identified by the anomaly detector.

9.3. HTTP Response Code Anomaly

In this use case, the ubuntu victim is attacked by the kali attacker. In the following figure, on the right side is the attackers console performing web attacks using Nikto. On the right left side is the CADM's graph for anomalies generate due to the attack. The graph is a timeline, showing peaks of

http error responses because the attacker is scanning the web app on Victim, and looking for hidden directories and vulnerabilities.

Table 7 : HTTP Response Code Anomaly Details

Attacker	Attacker-kali – 192.168.56.102
Victim	Victim-ubuntu-2204 – 192.168.56.101/DVWA
Command to emulate attack from attacker side	nikto -h http://192.168.56.101/DVWA

The attack and detection results are shown in the following screenshot.

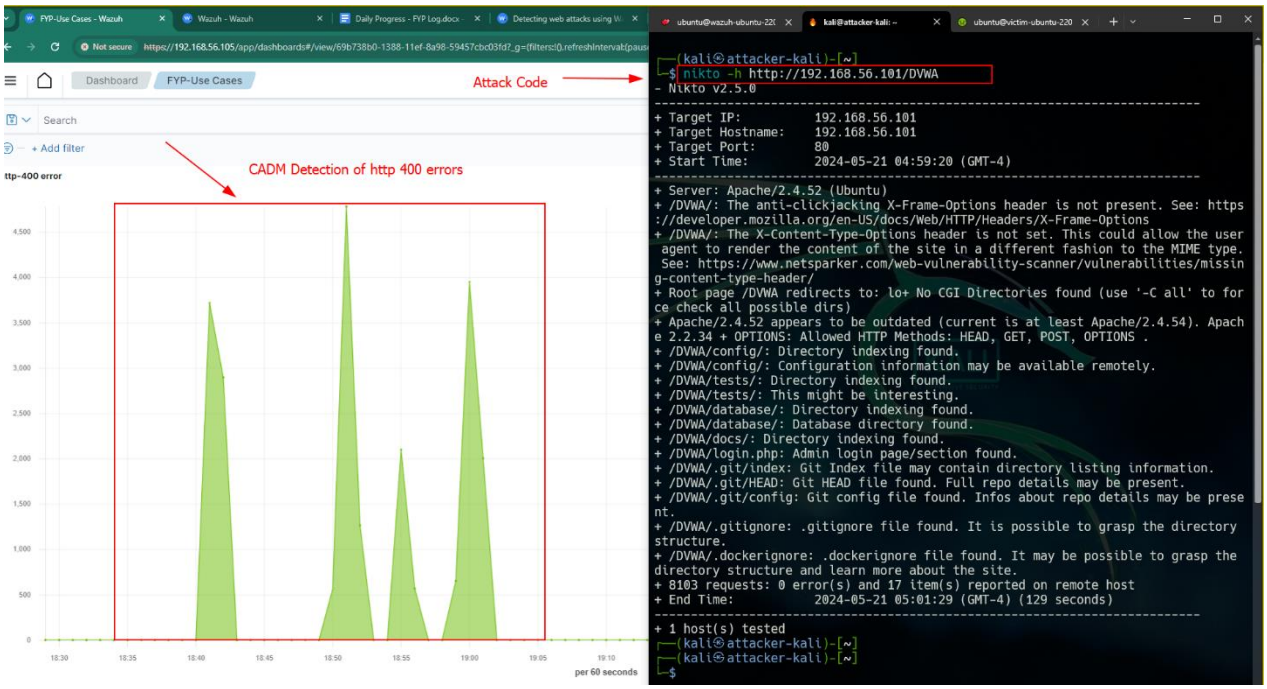


Figure 24: Detection of HTTP Response Code Anomaly - Attack Emulation with Nikto

BIBLIOGRAPHY

Wazuh, & Abdullah Al Noman. (2023, October 12). Enhancing IT security with anomaly detection in Wazuh | Wazuh. Retrieved May 23, 2024, from Wazuh website: https://wazuh.com/blog/enhancing-it-security-with-anomaly-detection/#failed-logins-anomaly

Wazuh, & Bassey, C. (2022, June 15). Detecting Cobalt Strike beacons using Wazuh | Wazuh. Retrieved May 23, 2024, from Wazuh website: https://wazuh.com/blog/detecting-cobalt-strike-beacons-using-wazuh//

Wazuh, & Ifeanyi Onyia Odike. (2022, November 11). Responding to network attacks with Suricata and Wazuh XDR | Wazuh. Retrieved May 23, 2024, from Wazuh website: https://wazuh.com/blog/responding-to-network-attacks-with-suricata-and-wazuh-xdr/

Wazuh, & Abdullah Al Noman. (2023, March 2). Monitoring USB dsrives in Windows using Wazuh. Retrieved May 23, 2024, from Wazuh website: https://wazuh.com/blog/monitoring-usb-drives-in-windows-using-wazuh/

Wazuh, & Iseoluwa Titiloye Oyeniyi. (2023, May 4). Monitoring Linux resource usage with Wazuh.: https://wazuh.com/blog/monitoring-linux-resource-usage-with-wazuh/

Wazuh. (2024). Monitoring execution of malicious commands - Proof of Concept guide., from Wazuh.com website: https://documentation.wazuh.com/current/proof-of-concept-guide/audit-commands-run-by-user.html

Wazuh, & Obahor, V. (2022, October 28). Detecting web attacks using Wazuh and teler | Wazuh. from Wazuh website: https://wazuh.com/blog/detecting-web-attacks-using-wazuh-and-teler/

LinkedIn. (2024)., from Linkedin.com website: https://www.linkedin.com/pulse/detecting-abnormal-network-traffic-using-suricata-wazuh-gupta-1ffpf/

cadm

ORIGINALITY REPORT

12%	7%	2%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	4%
2	wazuh.com Internet Source	1%
3	Submitted to Leeds Beckett University Student Paper	1%
4	Submitted to Higher College of Technology Student Paper	1%
5	community.safe.com Internet Source	1%
6	Submitted to Manchester Metropolitan University Student Paper	<1%
7	fastercapital.com Internet Source	<1%
8	Submitted to University of East London Student Paper	<1%
9	www.coursehero.com	

	Internet Source	<1%
10	Submitted to Institute of Technology Carlow Student Paper	<1%
11	Submitted to University of Westminster Student Paper	<1%
12	Submitted to 79920 Student Paper	<1%
13	Submitted to The University of Wolverhampton Student Paper	<1%
14	Submitted to University of Wollongong Student Paper	<1%
15	research.library.mun.ca Internet Source	<1%
16	Pinto, João Tiago Barbosa. "Tools to Support Practical Teaching of Software Engineering", Universidade do Porto (Portugal), 2024 Publication	<1%
17	pdffox.com Internet Source	<1%
18	ijsrcseit.com Internet Source	<1%
19	ir.iba.edu.pk Internet Source	<1%

20	www.geeksforgeeks.org Internet Source	<1%
21	Harlan Carvey. "Web Server Compromise", Elsevier BV, 2018 Publication	<1%
22	Submitted to Central Queensland University Student Paper	<1%
23	kipdf.com Internet Source	<1%
24	pdfcoffee.com Internet Source	<1%
25	www.be.unsw.edu.au Internet Source	<1%
26	www.mdpi.com Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

How much of this submission has been generated by AI?

0%

of qualifying text in this submission has been determined to be generated by AI.

Caution: Percentage may not indicate academic misconduct. Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

What does the percentage mean?
The percentage shown in the AI writing detection indicator and in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was generated by AI.

Our testing has found that there is a higher incidence of false positives when the percentage is less than 20. In order to reduce the likelihood of misinterpretation, the AI indicator will display an asterisk for percentages less than 20 to call attention to the fact that the score is less reliable.

However, the final decision on whether any misconduct has occurred rests with the reviewer/instructor. They should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in greater detail according to their school's policies.

How does Turnitin's indicator address false positives?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be AI-generated will be highlighted blue on the submission text.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

What does 'qualifying text' mean?
Sometimes false positives (incorrectly flagging human-written text as AI-generated), can include lists without a lot of structural variation, text that literally repeats itself, or text that has been paraphrased without developing new ideas. If our indicator shows a higher amount of AI writing in such text, we advise you to take that into consideration when looking at the percentage indicated.

In a longer document with a mix of authentic writing and AI generated text, it can be difficult to exactly determine where the AI writing begins and original writing ends, but our model should give you a reliable guide to start conversations with the submitting student.