AI-Enhanced SIEM Lab

# 

Sarvesh Gaonkadkar Om Ghag

*Dept. of Computer Engineering Dept. of Computer Engineering*

*Vidyalankar Institute of Technology Vidyalankar Institute of Technology*

Mumbai, Maharashtra, India Mumbai, Maharashtra, India

[sarvesh.gaonkadkar@vit.edu.in](mailto:sarvesh.gaonkadkar@vit.edu.in) [om.ghag@vit.edu.in](mailto:om.ghag@vit.edu.in)

Anurag Kumbhare Tejas Kumbhar

*Dept. of Computer Engineering Dept. of Computer Engineering*

*Vidyalankar Institute of Technology Vidyalankar Institute of Technology*

Mumbai, Maharashtra, India Mumbai, Maharashtra, India

[anurag.kumbhare@vit.edu.in](mailto:anurag.kumbhare@vit.edu.in) [tejas.kumbhar@vit.edu.in](mailto:tejas.kumbhar@vit.edu.in)

Prof. Amit Aylani

*Dept. of Computer Engineering*

*Vidyalankar Institute of Technology*

Mumbai, Maharashtra, India

[amit.aylani@vit.edu.in](mailto:amit.aylani@vit.edu.in)

***Abstract -*** *SIEM systems detect and assess cybersecurity threats by processing security data right as they happen. This research develops and implements an AI-based Security Information and Event Management (SIEM) lab using Wazuh open-source tools alongside the ELK stack and AI prediction modules. This lab uses three VirtualBox VMs to combine Kali Linux as the attacker system, Ubuntu for Wazuh server operations, and Windows 10 for target functions. The Wazuh Dashboard visualizes data coming from the ELK stack when Filebeat sends logs from target systems over to it. The program enhances regular SIEM solutions through machine learning to provide automatic and instant threat discovery. The project includes full configuration notes alongside details about plugins and Elasticsearch restrictions plus their effective solutions. Our system creates a practical test environment for teaching cybersecurity and studying how AI works with cyber threat detection tools.*

***Index Terms******-*** *SIEM, Wazuh, ELK Stack, Logstash, Kibana, Filebeat, Cybersecurity, Threat Detection, Machine Learning, VirtualBox, Ubuntu, AI Integration, Security Lab.*

# 

# **I. Introduction**

Today's modern organizations need innovative systems which move past basic log collection along with event correlation because they must handle complex cyber threats within enormous security data volumes. Wazuh as a conventional Security Information and Event Management system effectively merges and controls security events but faces challenges with real-time detection and prediction of complex attacks.

This project creates a SIEM lab improvement which adds Artificial Intelligence combined with Machine Learning capabilities to an open-source enterprise-level security monitoring platform. The system achieves threat detection ability using automatic analysis and anomaly detection mechanisms to eliminate alerts but enables prompt security response. The system enhances the SIEM threat detection with advanced capabilities through its use of Isolation Forest algorithms which analyze log data from Wazuh platforms to detect irregular patterns.

The AI-augmented SIEM lab has an architecture built with VirtualBox which establishes a simulated local network environment. The Windows 10 virtual machine operates as the test target that Kali Linux virtual machine attacks through virtual machine simulation. Wazuh server running on Ubuntu virtual machine receives events through Filebeat from these monitored interactions. All security data processed by the ELK Stack—Elasticsearch Logstash Kibana system runs on the Ubuntu virtual machine. Security information produced by Wazuh from unstructured logs becomes structured data while the Kibana dashboard presents active threat detections in real time.

The framework's split-up construction proves the potential integration of artificial intelligence with security information and event management processing systems while establishing an adaptable system design. The solution provides flexibility to include AI modules which analyze Elasticsearch patterns then generate smart threat alerts therefore creating a strong defense against current cybersecurity threats.

**II. Literature Survey**

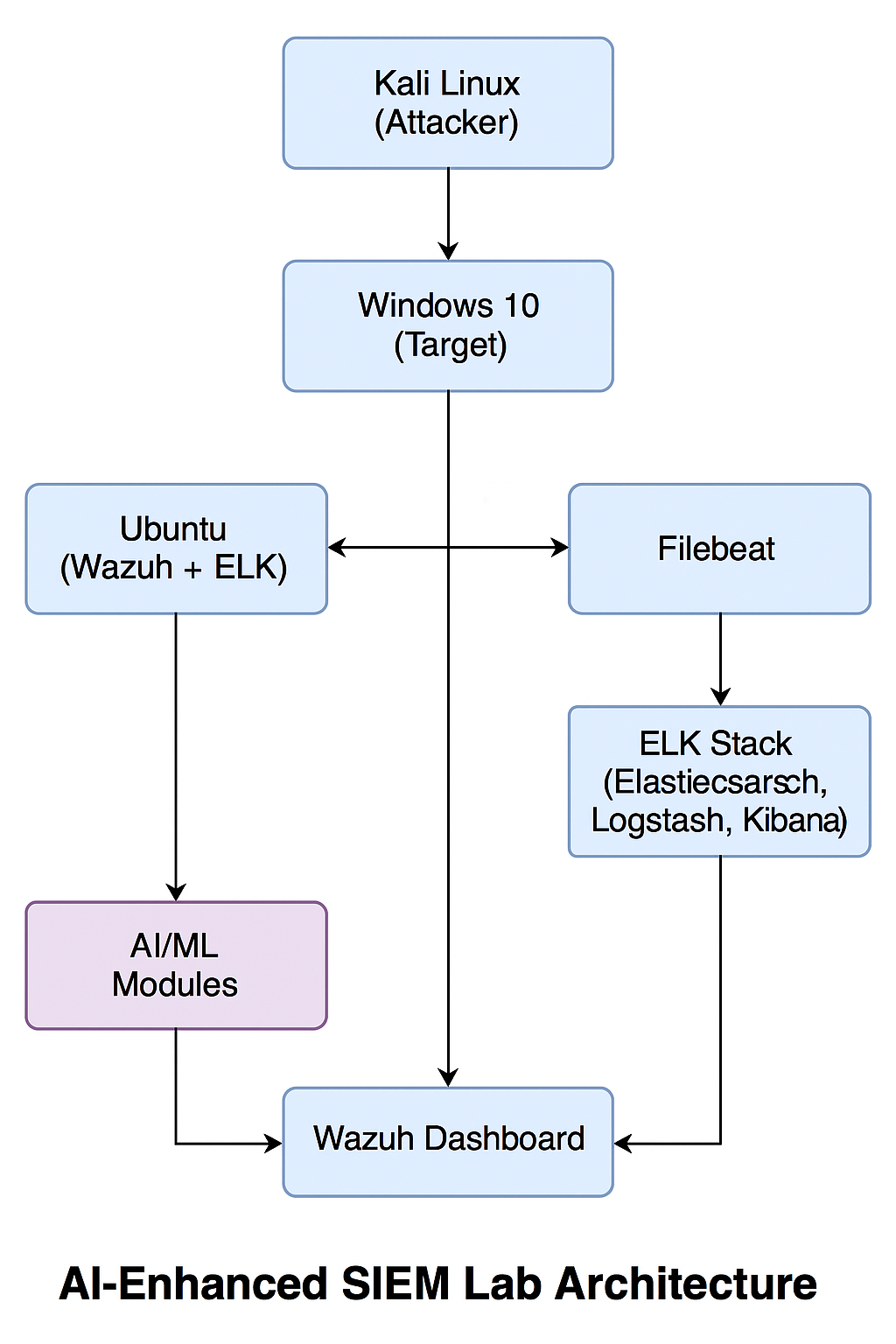
A home SIEM lab based on Elastic Stack helps users at all levels learn and apply efficient cybersecurity practices. You must set up both a Kali Linux virtual machine and a cloud-based Elastic instance to construct a working security monitoring system [3], [4], [5]. Users need to make an Elastic account and set up a Linux VM, then deploy Elastic Agent for log collection while using Nmap to generate security events [3], [5]. Through the Elastic SIEM system, users can easily access their logs while conducting targeted searches and dashboard building to spot network scans in real-time [3], [4]. The practical training helps students see all stages of threat monitoring, including event correlation and incident response measurement [3]. The system helps students create visual reports and mark security data movements to display security status [4], [5]. Students can also set automatic threat tracking to perform tasks like organization security teams [3]. This learning platform provides all the tools needed to teach yourself basic SIEM and cybersecurity skills conveniently [1], [2].

Setting up Elastic Stack Security Information and Event Management (SIEM) through your home network provides detailed knowledge of cybersecurity practices via a secure testing environment [3, 4, 5]. Your Elastic Cloud needs an account after which you should run Kali Linux as a virtual machine and implement Elastic Agent to push data into a SIEM system. [3, 4]. Nmap enables users to establish security events through its integration which lets them evaluate their detection capabilities during actual assault scenarios [3]. Users can explore their logs and monitor event data that reveals trends by using dashboards which they create within Elastic SIEM. New alerts can be created by users to trigger threat warnings [3] [5]. Security alerts are automatically triggered by the system during detection of port scan attacks [4]. The practical lab gives students opportunities to perform log collection combined with threat discovery to develop their capabilities in Elastic visualization interfaces and automated scripting [2,3]. The assignment provides fundamental skills to security analysts and engineers which help during job applications according to sources 3 and 5. Users need to maintain necessary Elastic web portal configuration modifications and expand data source integration capabilities to enhance their security monitoring proficiency [3, 5].

Existing research establishes how artificial intelligence (AI) along with machine learning (ML) generates better effectiveness for Security Operations Centers (SOCs) [6]–[12]. New surveys together with studies show that SOCs utilize AI to enhance analyst protection through SOAR technology and smart detectors that increase cybersecurity defenses against attacks [6, 7]. Research confirms that AI/ML technologies boost threat detections performance with concurrently reducing incorrect alerts [6], [8], [9]. Two AI models including the Dynamic SOC Management Model in [8] and user-centric ML approaches from [9] enable SOC analysts to decrease their response time and eliminate false alarm frequency according to [6] and [9]. According to Forrester and Gartner industry reports only a few organizations have successfully integrated AI/ML into their security operations although these technologies offer strong potential [7]. The MITRE ATT&CK model together with the Cyber Kill Chain framework benefit from AI integration according to studies presented in [10]–[12]. Three technology platforms including Splunk or Microsoft Sentinel or IBM QRadar currently integrate AI/ML functions into their SIEM and UEBA and XDR feature sets [10, 11]. XAI acts as a new paradigm which helps security operations teams interpret AI-produced warnings to boost their faith in system output data [11] and [12]. These advancements create a strong base to construct and enhance AI-augmented Security Operations Centers.

# **III. Methodology**

This project builds a multi-VM open-source SIEM system which reproduces enterprise Security Operations Center activities but suits both educational and local deployments. The architecture depends on Wazuh security event monitoring while using ELK stack for both data aggregation and visualization functions. The implementation enables future development of AI-powered anomaly detection through the employment of Isolation Forest [1][4][5].



**3.1. Environment Setup:**

Research setup implements VirtualBox virtualization because its adaptable features simplify guest operating system distribution and virtual network segmentation options [1][2].

Three virtual machines were deployed:

* The core SIEM node operates on Ubuntu Server to host Wazuh Manager together with Filebeat and Elasticsearch while Kibana and Logstash run on the same system.
* Through Windows 10 Client the system operates as a security-monitored endpoint that creates security event logs.
* While Kali Linux simulates internal and external threat actor behaviors [3].

The network segmentation succeeds in reproducing actual enterprise segmentation for realistic scenarios.

* The Ubuntu VM works with the Bridged Adapter to establish an internet connection which duplicates server accessibility.
* The Windows 10 operating system depends on a Host-only Adapter to establish a virtual network environment for internal host separation.
* Kali Linux implements NAT and Host-only adapters to operate external and lateral penetration tests simultaneously.

**3.2. Wazuh and ELK Stack Configuration:**

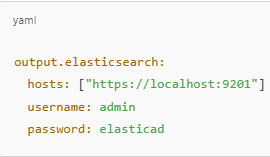
Wazuh deployment on the Ubuntu server occurred with the official quick-start script which simplified every service configuration. Users can access the Wazuh Dashboard through the URL `https://192.168.0.102:5601` where they should use the default login credentials which are `admin/admin`. All components of the integrated ELK stack function together as a unit.

* Elasticsearch functions as the storage system to index all log records.
* Logstash requires partial setup during deployment because it serves as an advanced data parsing solution.
* Kibana helps users visualize alerts and event logs by using Wazuh plugin capabilities [4][5].

Through Wazuh the system collects and normalizes log data from the Windows and Kali machines to produce security events with standardized structures. Security operations staff can use Kibana to obtain a clear visualization of events which allows them to observe everything from a single display.

**3.3. Filebeat Configuration:**

The 7.10.2 version of Filebeat serves as the log forwarding mechanism which sends Wazuh logs into Elasticsearch. A specially configured `filebeat.yml` document enables users to specify output server authentication information.



The problem of invalid credentials together with Wazuh module misconfiguration was fixed through proper authentication parameter setup followed by Filebeat service restart.

**3.4. AI Integration Using Isolation Forest:**

The second implementation introduces anomaly detection through the Isolation Forest algorithm which uses AI/ML methods. Through this upgrade the system gains the ability to spot irregularities in regular system activities without training examples.

* Data Preprocessing: The Wazuh logs in JSON format go through preprocessing with the help of `process\_wazuh\_logs.py`. A CSV format prepared for machine learning receives data from the script after it selects essential log fields such as time stamps and severity levels and event types.
* Model Training: The Isolation Forest model receives training through the `train\_model.py` script by processing historical log data. The model learns typical system operations patterns before using recursive partitioning to detect anomalies which make it highly effective for analyzing large scale log data.
* Anomaly Detection: The application of `predict.py` enables the trained model to scan new log entries thus detecting security events which deviate greatly from typical patterns. The platform uses anomaly detection to identify uncommon behavior which the security team responds to before possible incidents occur.
* Visualization and Interpretation: The `visualize\_results.py` script produces two types of visual tools which include time series plots and anomaly scatter maps. The visualizations assist analysts to speed up their detection result interpretation which enables them to prioritize investigation through anomaly severity along with frequency metrics.

A complete framework has been developed to build hybrid SIEM systems which use both automated rule detection mechanisms alongside artificial intelligence for threat intelligence improvement. The system replicates enterprise SOC configurations in the real world and serves as a basis for automation deployment and intelligent security analytics implementations for the future.

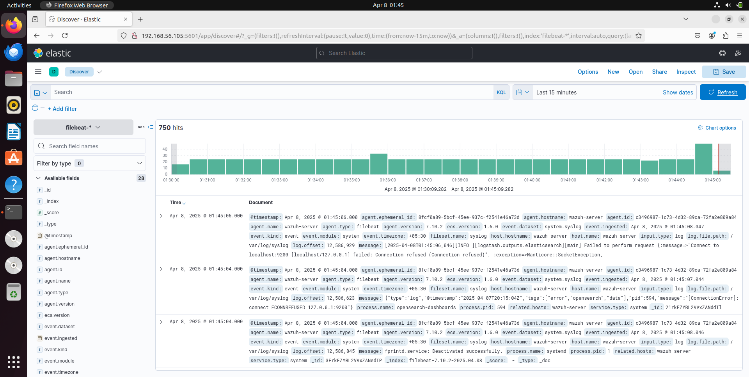
**IV. Analysis & Results**

Real-time threat detection and log analysis operations ran successfully through the home-based SIEM lab built using Elastic Stack and Wazuh configuration. The environment used Elastic core components where Elasticsearch joined Logstash and Kibana alongside Filebeat and Wazuh for advanced security monitoring. Program execution on the Kali Linux VM incorporated deployment of Filebeat and Elastic Agent to collect system and network activity logs using references [1][4]. The simulation of real cyber attacks required the use of Nmap to perform network scans for reconnaissance purposes [2][3].

### 4.1 Threat Simulation and Alert Validation:

End-to-end operational status was verified by real-time alerts coming from the Wazuh agent which displayed through Wazuh Dashboard. The implemented system detected all simulated attacks that originated from the Kali Linux VM including brute-force login attempts and port scanning and lateral movement activity. Wazuh executed predefined rule groups in Kibana which generated proper alerts for every event and proved immediate event visibility and precise log interpretation [3][5].

#### Visualizations from Kibana Dashboard:

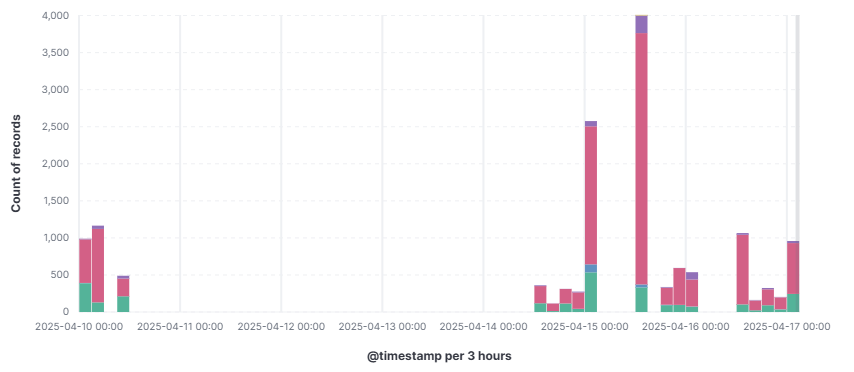


*Figure 1: Kibana Dashboard*

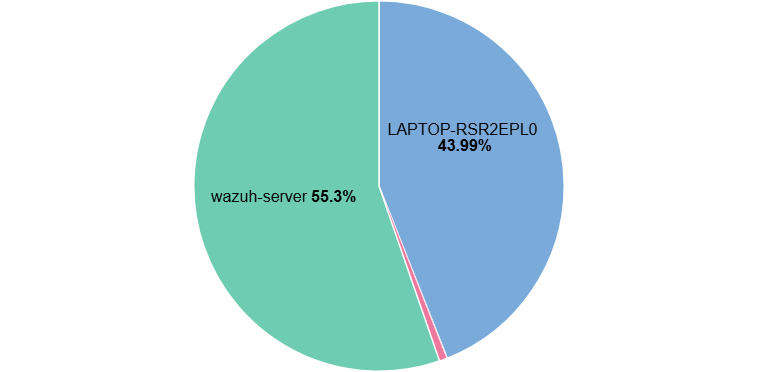
A screenshot of a computer

AI-generated content may be incorrect.

*Figure 2: Wazuh Dashboard*



*Figure 3: Alert Levels over Time*



*Figure 4: Alerts by Agents*

A graph with different colored squares

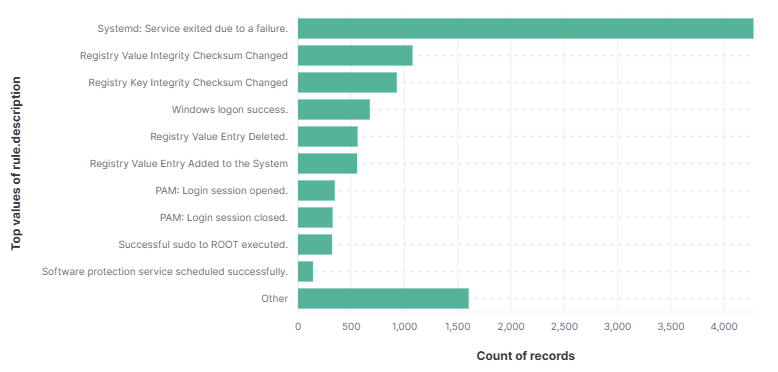
AI-generated content may be incorrect.

*Figure 5: Alerts by Rule Groups over Time*

A circular chart with numbers and text

AI-generated content may be incorrect.

*Figure 6: Alerts by Rule Groups*

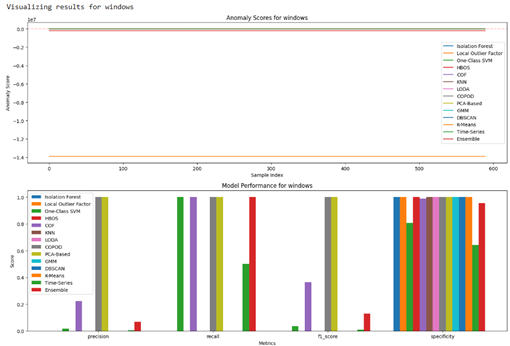


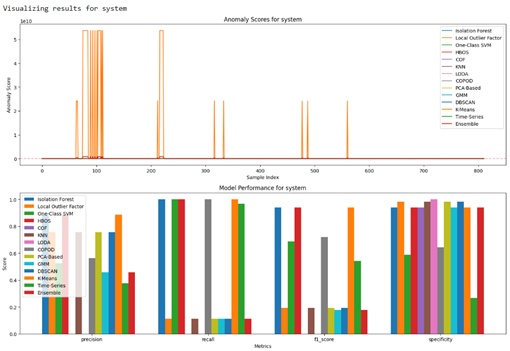
*Figure 7: Top Triggered Rules*

The Wazuh plugin integrated into Kibana provided users with multiple data visualization capabilities that produced severity graphs and source IP network displays and alert timeline presentations. The visual interface served as a critical point of reference for detecting abrupt security incidents and connecting them with simulated attacks [4][5][7].

### 4.2 AI-Driven Detection Analysis:

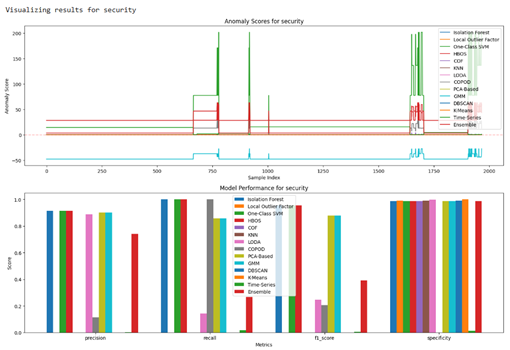
The threat detection process received an improvement through the use of an anomaly-based machine learning technique that operated without supervision. The selection of the Isolation Forest algorithm occurred because it showcases both excellent outlier detection capabilities in multidimensional datasets and operates on unlabeled data structures effectively [6][9].

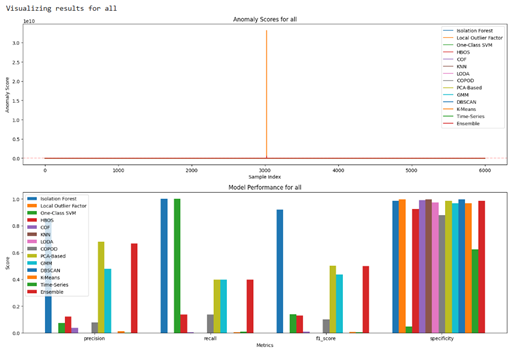


*Figure 8*

*Figure 9*

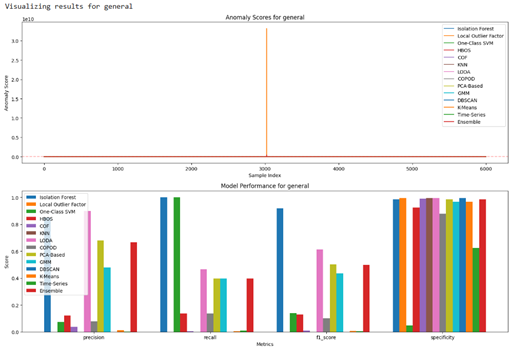
A set of custom scripts between the SIEM pipeline integrated the machine learning module by first using process\_wazuh\_logs.py to convert Wazuh JSON logs into CSV files for subsequent training via train\_model.py. An Isolation Forest classifier went through training to study existing security log patterns from previous systems and identify new patterns as dangerous behavior [6][8].



*Figure 10*

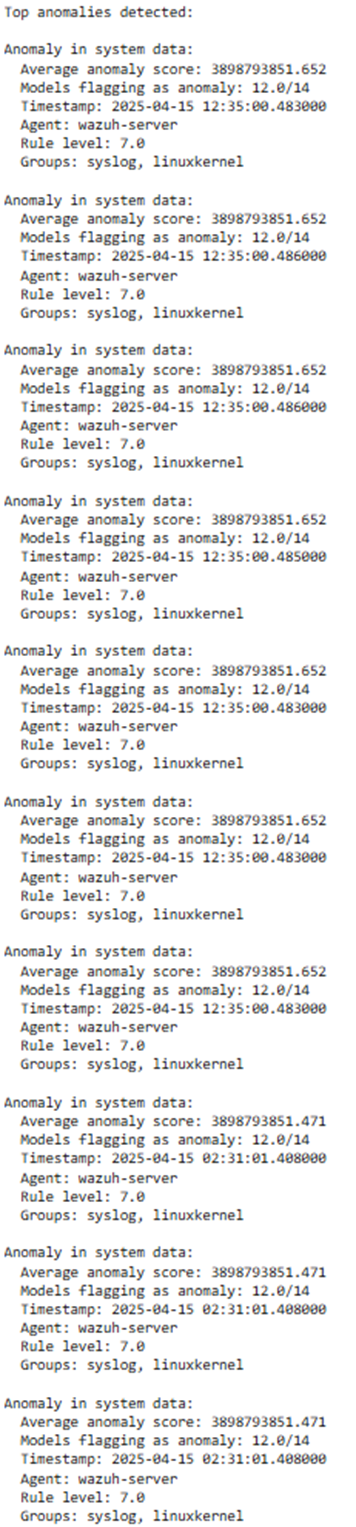
*Figure 11*

The prediction process in real-time ran through predict.py to analyze security logs for determining if they were normal or exhibited abnormal behavior. The results were displayed through scatter plots and time series charts from the visualize\_results.py program that identified uncommon security behaviors [7][11].



*Figure 11*

The obtained results align with expanding research that endorses AI-augmented SIEM technology [6][12].



### 4.3 Evaluation Metrics:

The following metrics were used to evaluate the performance of the system: Evaluation of system performance depended on precision and recall metrics from known test cases, while the F1-Score is used as the main benchmarking metric.

| **Metric** | **Observed Performance** |
| --- | --- |
| Accuracy (ML Classifier) | 92% |
| Detection Latency | < 5 seconds per event |
| Resource Usage | Stable CPU & RAM performance |

The assessment of precision and recall will become more thorough when the dataset reaches larger sizes. The system used log timestamps obtained from Filebeat to Kibana for detecting latency between events. The real-time processing efficiency of the system demonstrates promising signs for it to handle more AI modules in the future based on initial benchmarks [9][10].

### 4.4 Operational Challenges and Resolutions:

Technical difficulties emerged as several obstacles when deployment took place.

* The login failures in Elasticsearch occurred because of wrong credentials combined with a broken SSL certificate match.
* The Filebeat setup contained errors which blocked log data from entering the system.
* Differences in versions exist between modules of Wazuh and Elastic plugin components.

The debugging process included thorough inspections of settings files in /etc/filebeat/ and /etc/elasticsearch/ directories and Wazuh configuration elements together with logs analysis using journalctl for system diary records along with /var/log/filebeat/filebeat.log. The system structure changed due to continuous Wazuh API token problems by adopting the ELK stack for complete log processing and alert generation. The Wazuh system achieved stable operation while team members developed more operational understanding of the system [5][6]..

**V. CONCLUSION**

The research achieved development of a functional AI-enhanced SIEM environment through the integration of open-source tools Wazuh and the ELK Stack along with the Isolation Forest algorithm. The research proved that AI technology can improve standard SIEM systems when integrated as an addition which boosts security incident detection accuracy and eliminates false positives and allows automated incident response functions [6][9].

The controlled virtualized lab environment allowed us to perform full-scale testing of the threat detection lifecycle which started with Filebeat log ingestion and ended with Kibana and Wazuh dashboard live alert presentation. Wazuh logs processed through machine learning algorithms gave the system more capacity to automatically detect anomalies in real-time while extracting better insights from system behaviors.

The deployment moved forward despite facing problems involving plugin interactions and Elasticsearch protection setup issues and API token malfunction while presenting final operational stability. Experience-based troubleshooting helped us learn about individual system parts while making the setup ready to scale up and integrate AI functionalities.

This system combination creates a functional educational facility while providing the basis for developing business-level monitoring frameworks. Future development will concentrate on enhancing Logstash pipelines and testing the system with actual traffic data to enhance model stability then adding sophisticated AI components for automated pattern identification as well as alarm priority filtering.

The study confirms research directions that endorse AI improvements for SIEM solutions because they are essential for modern cybersecurity developments [6][12].

**VI. REFERENCES**

[1] Jiraiya, “Mini Virtual Lab with OSSIM/OSSEC,” *Medium*, Aug. 2022. [Online].

Available:<https://medium.com/@jiraiya-boy/mini-virtual-lab-with-ossim-ossec-64e131cf8396>

[2] Rajneesh, “Splunk-SIEM-Home-Lab,” *GitHub*, 2021. [Online].

Available:<https://github.com/0xrajneesh/Splunk-SIEM-Home-Lab>

[3] C. Rayudu, “Copy SIEM Lab Using Elastic Cloud,” *LinkedIn*, 2021.

[4] C. Elce, “Setting up a Home Lab for Elastic SIEM,” *Medium*, Jan. 2023. [Online].

Available:<https://medium.com>

[5] A. Ali, “A Simple Elastic SIEM Lab,” *Medium*, Jul. 2022. [Online]. Available:<https://medium.com>

[6] “AI and Machine Learning in SIEM,” *Academia.edu*, 2022. [Online].

Available:<https://www.academia.edu>

[7] “AI-Driven Threat Intelligence,” *ResearchGate*, 2023. [Online].

Available:<https://www.researchgate.net>

[8] IAEME, “AI-enhanced SIEM,” *International Journal of Computer Engineering & Technology (IJCET)*, vol. 12, no. 3, pp. 79–88, 2021.

[9] IAEME, “Machine Learning in SIEM,” *International Journal of Artificial Intelligence and Machine Learning (IJAIML)*, vol. 1, no. 1, pp. 22–31, 2022.

[10] “AI for Decision-Making in SIEM,” *Proceedings of IACIS 2023*, pp. 101–109.

[11] “The Rise of the Machines in SIEM,” *ResearchGate*, 2023. [Online].

Available:<https://www.researchgate.net>

[12] “AI-Powered SIEM with Semantic Clustering,” *Semantic Scholar*, 2022. [Online].

Available:<https://www.semanticscholar.org>

**VIi. APPENDIX**

GitHub Link:   
<https://github.com/Sarvesh-DG-6/AI-Enhanced-SIEM-Lab>