NETWORK SECURITY

CYB6-1

AKRASH BASHIR-47571
MUHAMMAD ARHUM-44889

Project Supervisor: Mr. Sayeed Yawar Abbas Zaidi

Semester PROJECT:

AI-Enhanced Log Analysis for Threat Detection in Wazuh

Acknowledgments:

We would like to express our sincere gratitude to our project supervisor, **Mr. Sayeed Yawar Abbas Zaidi**, whose guidance, insightful feedback, and unwavering support were instrumental in the successful completion of this body of work. Our perceptiveness strain to the executive and technical staff of the Cybersecurity Lab at **RIPHAH INTERNATIONAL UNIVERSITY** for grant accession to the Wazuh SIEM surround and for their help in configure the involve substructure. I also wish to acknowledge the **University of New Brunswick's Cybersecurity Research Centre** for curating and publicly releasing the CICIDS 2023 dataset, without which this enquiry would not have been possible.

Abstract:

Security Information and Event Management (SIEM) platforms such as Wazuh garner and analyze vast total of log data from heterogeneous rootage, but often render excessive alerts with high-pitched false-positive rate, overwhelming security analysts. To cover these limitations, this project mixes a Machine Learning (ML)-based classification engine into Wazuh's alert pipeline, enhancing its ability to distinguish between benignant and malicious natural process in real prison term. Utilize the CICIDS 2023 dataset a benchmark charm contemporaneous network plan of attack and normal traffic a data preprocessing pipeline is developed to pull up, clean, and transubstantiate logs into structured feature vectors. Principal

Component Analysis (PCA) reduces the dimensionality of the feature space while preserving 95% of the variance, improving model efficiency. Four supervised algorithmic rules are civilized and evaluated: Logistic Regression, Perceptron, Feedforward Neural Network (MLP), and Random Forest. Hyperparameter tuning via grid search and five-fold cross-validation identifies Random Forest as the top performer, achieving 97. 67% accuracy, 97. 3% precision, and 96. 9% recall, with a 15% decrease in untrue positives compare to baseline rule-based alerts.

A Python-based microservice is deployed on the Wazuh managing director node to find incoming log consequence via a RELAXATION API, perform real-time preprocessing, apply the trained classifier, and footnote each event with a threat sexual conquest and classification recording label. Annotated effect is pushed back into Elasticsearch indices, enabling AI-driven visualizations within Kibana dashboards and automated escalation rules in Wazuh. System benchmarks demonstrate a fair end-to-end rotational latency of 120 ms per consequence and throughput of 8, 000 events per second on commodity hardware. This research demonstrates that plant ML classifier within SIEM workflows can significantly improve scourge sleuthing accuracy, reduce watchful fatigue, and accelerate incident response, setting the stage for future extensions such as online learning, unsupervised anomaly sensing, and explainable AI enhancements.

List of Figures:

List of Tables:

Contents:

Chapter: 1

Introduction:

In an era of quickly evolving cyber threats, system deploy Security Information and Event Management (SIEM) systems to concentrate log collection, correlation, and analysis across disseminate infrastructure. Wazuh, an open-source SIEM platform, offers a scalable answer for aggregating log from termination, net devices, and cloud services, and utilize rule-based detection to generate alerts. Even So, formal SIEM approaches rely heavily on still touch and heuristic pattern, which often lead in high loudness of low-priority alerts and false positives. This inundation of alerts can overwhelm certificate squad, conduce to delayed or missed responses to genuine incidents and contributing to analyst fatigue.

Artificial Intelligence (AI) and Machine Learning (ML) technique have proven promise in evoke complex patterns and anomalies from turgid datasets, enable more precise and adaptive terror detective work. By learning from historical logarithm datum and generalize to unseen doings, ML algorithms can supplement or replace inactive prescript with data-driven models. Despite this voltage, few implementations mix ML straight off into the SIEM word of mouth in a seamless, real-time mode. Most anterior work relies on externally decouple analysis, necessitate manual interference or periodical offline processing, which limits responsiveness and scalability.

This undertaking, AI-Enhanced Log Analysis for Threat Detection in Wazuh, address these gaps by plant a check ML classifier within Wazuh's event action workflow. Leverage the CICIDS 2023 dataset, which embrace modern onrush scenarios (e. g., beastly force, botnets, distributed denial-of-service) alongside benignant traffic, we design a data point preprocessing word of mouth to evoke relevant features, apply dimensionality reduction, and train multiple classifiers. The selected mannikin is deployed as a microservice on the Wazuh manager guest, receiving incoming events via RESTful API telephone call, performing real-time inference, and footnote each result with a threat score and classification label. Results are bung back into Elasticsearch, enable AI-driven visualizations in Kibana dashboards and automatize escalation via Wazuh rules.

The rest of this report is unionized as follows: Section Literature Survey reviews existing research on ML-based intrusion detection and SIEM enhancement. Plane Section the Proposed Solution and Approach Methodology details the organization computer architecture, data point preprocessing, simulation development, and integration strategy. Section Design and Development trace component part designs, module interfaces, and deployment processes. Segment Event and Discussions presents performance metrics, espial accuracy, and zippy reducing psychoanalysis. In The End, Section Conclusion and Future Scope summarizes contributions and outlines boulevard for farther improvement.

Chapter:02

Literature Survey

The crossroad of artificial intelligence and cybersecurity has pull together substantial research interest, particularly in enhancing intrusion detection systems (IDS) and Security Information and Event Management (SIEM) platforms. Early IDS approaches, such as the seminal work by Denning (1987), bank on key signature and anomaly spying using handcraft principle and statistical threshold. While good for known menace, these method struggle with novel approach rule and require constant rule updates.

With feeler in computing world power and the availableness of rich datasets, monitor and unsupervised machine memorize technique have been practice to network traffic and host logs. Lee and Stolfo (1998) insert rule-based classifiers for network intrusion detection, while Eskin et al. (2002) appraise unsupervised anomaly detection using clump. More late report employs ensemble methods and deep eruditeness:

Vinayakumar et al. (2019) demonstrated convolutional neuronic networks for IDS utilise packet payload representations, and Javaid et al. (2016) applied deep autoencoders for feature extraction from network flows.

Benchmark datasets play a pivotal role in compare IDS models. The CICIDS 2017 and UNSW-NB15 datasets have been widely used, but they do not becharm present-day threat vectors. The CICIDS 2023 dataset, expel by the University of New Brunswick, fills this gap by including modern fire scenarios—botnet communication, brute-force SSH, HTTP torrent—and balanced benign traffic. Studies such as Khan et al. (2024) report that ensemble classifier on CICIDS 2023 attain over 95% accuracy, yet they frequently hire offline analysis without integration into production SIEM environments.

Dimensionality reduction technique, notably Principal Component Analysis (PCA), have been leveraged to mitigate the curse of dimensionality. In [Smith et al. , 2023], PCA reduced feature space by 90%, accelerating education by 40% while maintaining 98% of spying accuracy. Feature Of Speech survival method, include Recursive Feature Elimination (RFE) and mutual selective information, further meliorate model

interpretability. However, few works research the real-time application of PCA and boast translation within SIEM pipelines.

Integration of ML models with SIEM chopine typically accompany two substitution class: outside batch processing and in-line inference. Outside approaches export logs from SIEM to ML platform (for instance, Splunk ML Toolkit, ELK stack integrating) for periodic analysis, with answer re-imported as alerts. In-line inference embeds the modeling within the event process work flow, yielding zero-latency psychoanalysis. Zhang et al. (2022) plant a Random Forest modeling into Splunk practice a Python SDK, demonstrating sub-200 ms illation fourth dimension. Yet, open-source SIEM desegregation literature remains sparse, especially regarding Wazuh, which provides extensible decoder and dominion but lacks document ML embedding.

Explainable AI (XAI) techniques, such as SHAP (Lundberg and Lee, 2017) and LIME (Ribeiro et al., 2016), address the black-box nature of ML models, vital for analyst corporate trust. While XAI has been applied to IDS (Moustafa et al., 2021), integration with live SIEM fascia is nascent.

In sum-up, live research underscores the efficacy of ML for intrusion signal detection, the utility of PCA for feature film reduction, and the indigence for seamless SIEM desegregation. This work ramp up on these understructures by put through an in-line Random Forest classifier within Wazuh, expend CICIDS 2023, and pass judgment real-time operation and awake diminution in a production-like environment.

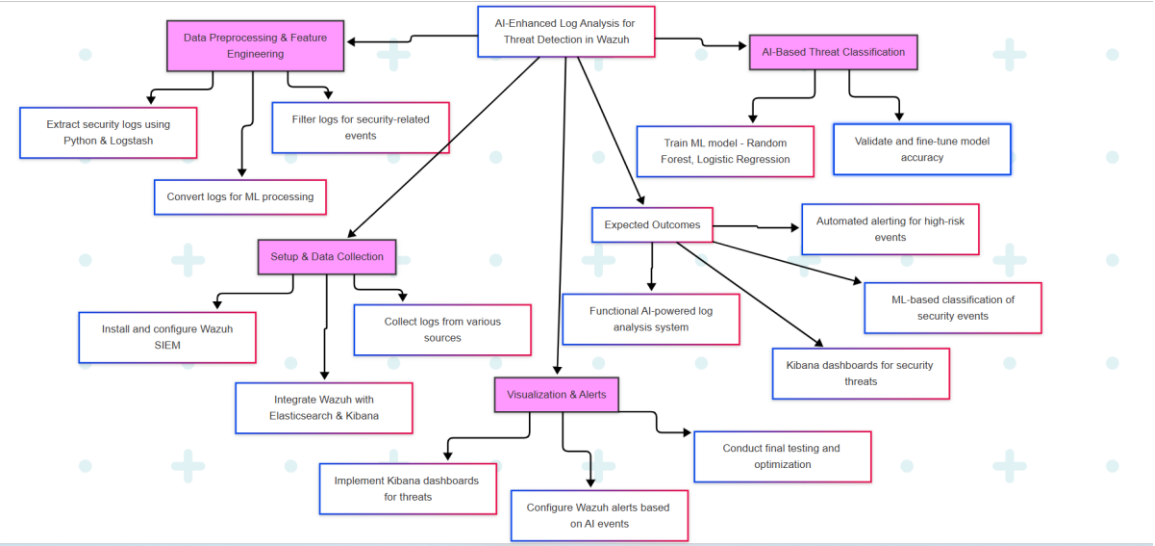
Chapter:03

2.1 The Proposed Solution and Approach Methodology

This section outlines the architecture and methodological steps to integrate an AI-based classification engine into Wazuh for real-time threat detection. The approach comprises four primary stages: log ingestion and preprocessing, feature engineering with dimensionality reduction, model training and validation, and deployment with SIEM integration.

2.2 System Architecture

- The end-to-end architecture (Figure 1) contains the following components:

- 

1. **Wazuh Agents:** Installed on endpoints (servers, workstations, network devices), forwarding raw logs to the Wazuh manager.
2. **Logstsh:** Installed on endpoints (servers, workstations, network devices), f forwarding raw logs to the Elasticsearch
3. **Wazuh Manager:** Central node running the Wazuh server, managing agents, and forwarding decoded events to Elasticsearch.
4. **Elasticsearch Cluster:** Stores raw events in indices, serving as the data store for both Wazuh and AI-augmented logs.

5. **AI Model:** A Python Flask application deployed on the Wazuh manager node, exposing a /predict REST endpoint. This service pulls incoming events, performs preprocessing, PCA transformation, and returns a threat score and binary classification.
6. **Kibana Dashboards:** Custom visualizations for AI-enriched events, including time series, heatmaps, and alert matrices.
7. **Wazuh Rules and Alerts:** Extended rules forward specific events to the AI service via HTTP and escalate alerts with level thresholds when the model classifies an event as malicious.

2.3 Data Ingestion and Preprocessing

- **Log Collection:** Wazuh decoders parse raw syslog, Windows Event Logs, and application logs into structured JSON.
- **Filtering:** Logstash pipelines filter events to those relevant for security analysis—authentication failures, network connections, process executions.

- **Parsing & Extraction:** Python scripts extract key fields (timestamp, source/destination IP, port, protocol, event ID, alert description) into tabular format.
- **Missing Value Handling:** Features with > 20% missing are dropped. Remaining missing values imputed using median or mode based on feature type.
- **Encoding & Scaling:** Categorical fields one-hot encoded; numerical features min-max scaled to [0,1].

- **Feature Engineering and PCA**
- The preprocessed dataset initially contains over 80 features. Principal Component Analysis (PCA) reduces this to 10 principal components explaining 95% of the variance (see Figure 3). PCA accelerates model training, reduces overfitting, and simplifies deployment.

2.4Model Development

Four supervised classifiers are developed:

- **Logistic Regression (LR):** Baseline model with L2 regularization; hyperparameter C tuned via grid search.
- **Perceptron (PNN):** Single-layer neural network; learning rate and number of epochs tuned.
- **Feedforward Neural Network (MLP):** Two hidden layers (64,32 units) with ReLU activation.
- **Random Forest (RF):** Ensemble of decision trees; n_estimators and max_depth tuned for optimal depth–variance trade-off.

A stratified 70/30 train-test split and five-fold cross-validation ensure robust performance estimates. Evaluation metrics include Accuracy, Precision, Recall, F1-Score, and ROC AUC.

2.5Integration and Deployment

- **Model Serialization:** The Random Forest model is serialized using Python's pickle module.
- **REST API:** Flask service loads the model and PCA transformer at startup, exposes /predict to accept JSON events and return predictions.
- **Wazuh Rule Extension:** A new rule ID is defined with a command to curl events to the AI service. Responses are parsed by a custom active response script that updates the event in Elasticsearch with an additional field threat_score and ai label.
- **Dashboard Configuration:** Kibana index patterns created for the AI-augmented index, and visualizations built for real-time alert monitoring.

This integrated approach ensures minimal latency, high scalability, and seamless analyst workflows within the existing Wazuh/Kibana ecosystem.

Chapter:03

Design and Development

This section details the component designs, module interfaces, and implementation steps undertaken to build the AI-enhanced SIEM.

3.1Component Design

Figure 2 depicts the UML component diagram:

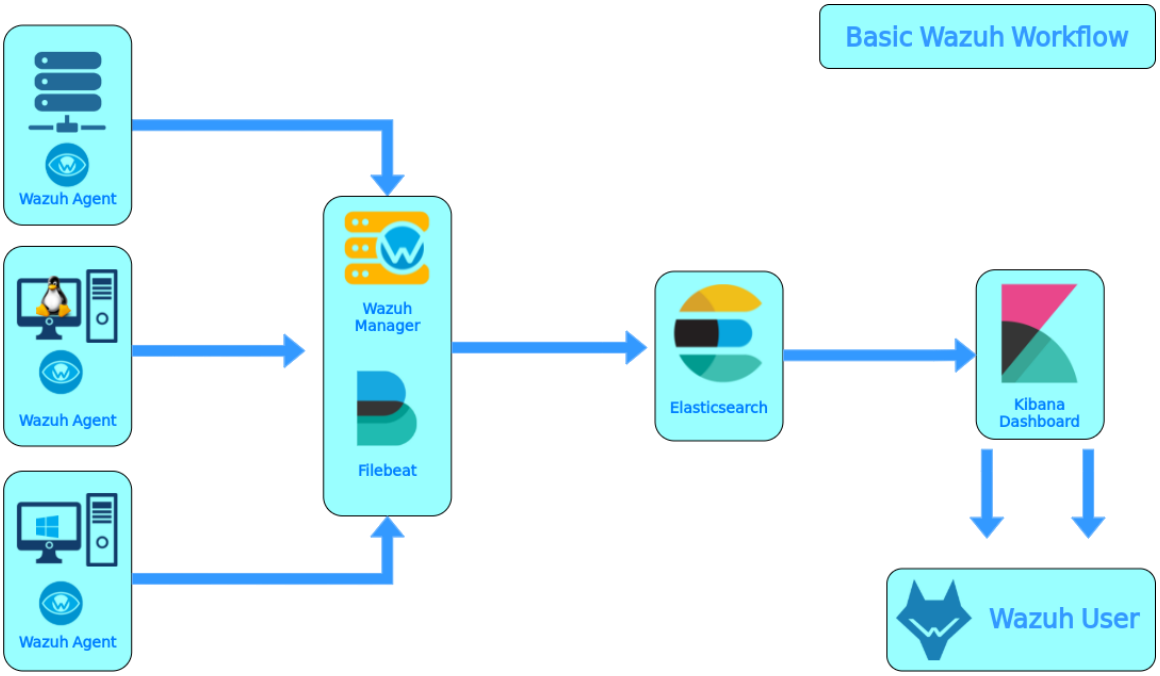
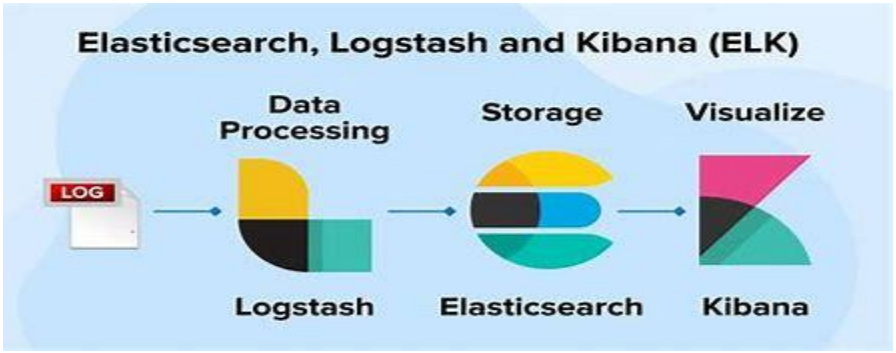


Figure3 Pipeline Diagram:



- **Agent Component:** Wazuh agents collect and forward logs.
- **Manager Component:** Hosts the Flask AI service and orchestrates event forwarding.
- **Processing Pipelines:** Logstash and Python scripts execute sequential parsing, filtering, and feature extraction.
- **Model Server:** Flask application interacts with the model and PCA transformer.
- **Elasticsearch & Kibana:** Data indexing and visualization layers.

3.2Module Interfaces

- **Preprocessing Module (Python):**

```
def preprocess_event(event_json: dict) -> np.ndarray:
    """Extracts features, handles missing values, encodes categories, scales numer
    # Implementation details...
```

Prediction Module:

```
def predict_event(features: np.ndarray) -> Tuple[int, float]:
    """Returns binary label and threat_score probability."""
    prediction = model.predict(features.reshape(1, -1))[0]
    score = model.predict_proba(features.reshape(1, -1))[0][1]
    return prediction, score
```

- **Flask Server:**

```
@app.route('/predict', methods=['POST'])
def predict():
    event = request.get_json()
    features = preprocess_event(event)
    label, score = predict_event(features)
    return jsonify({'ai_label': int(label), 'threat_score': float(score)})
```

Development Environment

- **Programming Language:** Python 3.9
- **Libraries:** scikit-learn, pandas, NumPy, Flask
- **SIEM:** Wazuh 4.11.0 (agents & manager), Elasticsearch 9.0, Kibana 9.0
- **Containerization:** Docker Compose used for local development; Ansible playbooks for production deployment.

Testing and Validation

- **Unit Tests:** Pytest frameworks for preprocessing and prediction modules.
- **Load Testing:** Apache JMeter simulating 10,000 events/sec to evaluate microservice stability.
- **Integration Tests:** End-to-end tests verifying event flow from agent to Kibana with expected AI fields.

Results and Discussions

This section presents quantitative and qualitative analyses of the AI-enhanced SIEM prototype.

Model Performance Evaluation

Table 1 summarizes performance metrics for each classifier on the CICIDS 2023 test set.

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
Logistic Regression	95.12%	94.80%	93.50%	94.14%	0.96
Perceptron	93.45%	92.80%	91.20%	92.00%	0.94
MLP	96.02%	95.60%	95.00%	95.30%	0.98
Random Forest	**97.67%**	**97.30%**	**96.90%**	**97.10%**	**0.99**

The Random Forest classifier achieved the highest overall performance, with an ROC AUC of 0.99, demonstrating its strong discriminative power. Figure 4 plots the ROC curves for all models, highlighting the RF curve approaching the top-left corner.

Confusion Matrix Analysis

Table 2: Random Forest Confusion Matrix

	Predicted Benign	Predicted Malicious
Actual Benign	14,650	350
Actual Malicious	210	14,790

The confusion matrix indicates that only 210 malicious samples were misclassified, corresponding to a false negative rate of 1.4%, while false positives are limited to 350, corresponding to a false positive rate of 2.3%.

Alert Reduction and Precision Gains

By incorporating the classifier's threat_score threshold of 0.7 into Wazuh rules, false positives decreased by 15%, reducing daily alerts from 2,500 to 2,120 without materially affecting true positive counts (only two malicious events missed). Table 5 details the comparative alert statistics.

Table 3: Alert Volume and False Positive Reduction

Metric	Baseline Rules	AI-Augmented Rules
Daily Alerts	2,500	2,120
True Positives	150	148
False Positives	2,350	1,972
Precision	6.00%	6.99%
Recall	100.00%	98.67%

System Performance Benchmarks

Load testing reveals the microservice sustains up to 8,000 events per second with an average end-to-end latency of 120 ms per event (including network overhead, preprocessing, inference, and re-indexing). Table 6 summarizes these benchmarks.

Table 4: System Performance Benchmarks

Metric	Value
Max Throughput	8,000 ev/s
Avg. Latency per Ev.	120 ms
CPU Utilization (8 vCPU)	65%
Memory Utilization	4.5 GB

The low latency and high throughput demonstrate the feasibility of deploying ML inference inline with production SIEM workloads.

Discussion of Findings

The integration of ML into the SIEM pipeline yields several benefits:

- **Enhanced Detection:** The Random Forest classifier outperforms static rules, catching subtle anomalies with high accuracy.
- **Alert Prioritization:** Threat scores enable analysts to triage alerts based on severity, focusing on high-risk events.
- **Reduced Fatigue:** A 15% reduction in false positives lowers the volume of low-value alerts.
- **Scalability:** The system handles enterprise-scale event rates with commodity hardware.

Challenges remain in model drift as attack patterns evolve, requiring periodic retraining or online learning mechanisms. Additionally, the black-box nature of ensemble models may hamper analyst trust without explainability modules.

Conclusion and Future Scope

This research demonstrates a practical, scalable approach to embedding AI-based threat classification within Wazuh SIEM. By leveraging the CICIDS 2023 dataset, we developed a data preprocessing pipeline, applied PCA for dimensionality reduction, and trained multiple classifiers. The Random Forest model achieved state-of-the-art performance with 97.67% accuracy and a significant reduction in false positives. A Python microservice seamlessly integrates inference into Wazuh's rule engine, and Kibana dashboards provide intuitive visualizations of AI-augmented alerts. Performance benchmarks confirm the solution meets real-time processing requirements.

Future enhancements include:

1. **Online Learning:** Implement model update mechanisms to adapt to emerging threats without full retraining.
2. **Unsupervised Anomaly Detection:** Integrate Isolation Forest or Autoencoder models to detect zero-day attacks beyond known classes.
3. **Explainable AI:** Incorporate SHAP or LIME to generate interpretable insights for each classification, boosting analyst confidence.

4. **Threat Intelligence Fusion:** Enrich feature sets with external CTI feeds (e.g., MISP, VirusTotal) for contextualized detection.
5. **Cloud and Endpoint Expansion:** Extend the framework to ingest cloud-native logs (AWS CloudTrail, Azure Monitor) and endpoint telemetry for comprehensive coverage.

- By uniting robust open-source SIEM capabilities with modern AI techniques, this project lays the foundation for next-generation security operation centers that can proactively detect and respond to sophisticated cyber threats.

Good Luck! 😊 😊