International Journal of application or innovation in engineering and management

Ijaiem.com/June 2024/ Volume 13/Issue 1/Article No-1/64-70

ISSN: 2319-4847

AI-Augmented SIEM Systems: Improving Threat Correlation and Alert Prioritization

Priya Rajan

Cybersecurity and Data Privacy Lab, Indian Institute of Technology (IIT) Madras, India

Article Info

Received: 28-04-2024 Revised: 05 -06-2024 Accepted: 16-06-2024 Published:27/06/2024

Abstract:

Traditional Security Information and Event Management (SIEM) platforms rely heavily on rule-based correlation, often leading to alert fatigue and missed advanced threats. This paper presents an AI-augmented SIEM architecture that integrates unsupervised learning and clustering techniques to enhance threat detection, correlation, and alert prioritization. Using a dataset of over 30 million anonymized logs from an enterprise network—including authentication records, DNS queries, and system events—we apply autoencoders for anomaly detection and density-based clustering (DBSCAN) for grouping related events. Our system integrates these models into the Splunk SIEM via Python SDK and real-time data pipelines. Compared to a baseline rule-only system, false positives are reduced by 41%, and average analyst triage time is shortened by 29%. Critical incident detection accuracy improves due to context-aware enrichment with external threat intelligence sources. The architecture also supports daily model retraining to adapt to evolving attack techniques. A key challenge was explaining anomalies to analysts; we addressed this using Shapley values and timeline visualizations for anomaly justification. Our study demonstrates that augmenting SIEMs with machine learning improves operational efficiency without replacing human expertise. We propose a reference design for AI-enhanced SOC workflows and guidelines for integrating ML components into existing detection infrastructure. This approach represents a significant step toward smarter and more scalable security operations.

1. Introduction

Security Information and Event Management (SIEM) platforms are the backbone of modern Security Operations Centers (SOCs), aggregating logs, alerts, and telemetry to detect threats and ensure compliance. However, traditional SIEMs are rule-driven and static, depending on pre-defined correlation rules that are often inflexible and noisy. This results in high false positive rates, alert fatigue, and delayed incident response, particularly as attackers adopt more evasive techniques.

To address these limitations, we investigate how artificial intelligence—particularly unsupervised learning—can enhance SIEM capabilities. Rather than replacing rule-based systems, we propose an **AI-augmented approach** that operates alongside traditional engines to improve event correlation and alert prioritization. Our goal is to demonstrate that integrating anomaly detection models and event clustering can reduce operational load while surfacing higher-fidelity alerts.

In this paper, we evaluate a practical implementation of this hybrid system using Splunk as the SIEM platform and Python-based machine learning models integrated via the Splunk SDK. We analyze 30 million logs collected from an enterprise network spanning authentication, network, and host activity over 90 days. Our system enhances detection with unsupervised anomaly scoring, clusters event timelines into context-rich alerts, and prioritizes threats using explainability tools. Through this design, we aim to move toward scalable, adaptive, and analyst-friendly SOC operations.

2. Related Work

Prior research has acknowledged the shortcomings of static rule-based detection systems. Studies by Sommer & Paxson (2010) and Garfinkel (2014) highlighted that manually curated rules often lag behind modern attack techniques. More recent efforts have explored the use of machine learning in intrusion detection systems (IDS), but less attention has been paid to its integration within full-scale SIEM workflows.

Guerra et al. (2021) introduced clustering-based anomaly detection for network traffic, but did not address real-time SIEM integration. Similarly, Splunk's own ML Toolkit provides basic anomaly detection tools, yet lacks out-of-the-box mechanisms for correlating multi-source telemetry or explaining flagged anomalies. Other efforts have introduced supervised learning for classification (e.g., logistic regression, random forest), but these models require labeled datasets—often scarce or biased in security contexts.

Unsupervised learning has shown promise in high-dimensional, unlabeled security data. Autoencoders, k-means clustering, and density-based methods like DBSCAN have been applied for anomaly detection, but typically in standalone experimental settings. Our work advances this by tightly integrating these models into a production SIEM workflow, including model retraining, live scoring, and alert enrichment with threat intelligence.

Furthermore, we address the gap in analyst explainability by employing SHAP (Shapley Additive Explanations) to justify anomaly scores and assist with decision-making. This bridges the trust gap between data scientists and SOC analysts, a key hurdle in real-world ML adoption.

3. System Architecture and Data Pipeline

Our proposed architecture consists of five primary components, deployed within a hybrid Splunk and Python ecosystem:

3.1 Data Ingestion

- Logs are ingested from Windows Event logs, Linux auditd, Palo Alto firewalls, Okta SSO, and DNS appliances.
- A Kafka-based stream buffers the incoming data, allowing feature extraction and transformation in real time.
- Data is normalized using the Common Information Model (CIM) to ensure schema consistency across source types.

3.2 Feature Engineering

Features include:

- Temporal fields (e.g., event frequency over 10-minute windows)
- User behavior indicators (e.g., login time deviation, failed login rate)
- Network activity (e.g., external IP ratios, DNS request entropy)

3.3 Anomaly Detection

Ijaiem.com/June 2024/ Volume 13/Issue 1/Article No-1/64-70

ISSN: 2319-4847

- Autoencoders are trained on baseline user and network behavior over a 14-day period.
- Reconstruction error scores are computed per event and thresholded to identify anomalies.
- A DBSCAN model clusters temporally and spatially related anomalies to create correlated alert groups.

3.4 Integration with Splunk

- Anomaly scores and cluster IDs are sent to Splunk using its Python SDK as custom indexed fields.
- Dashboards and search queries are extended to incorporate anomaly scores in triage.
- Alerts are enriched with MITRE ATT&CK mappings and threat intelligence via AbuseIPDB and VirusTotal APIs.

3.5 Model Lifecycle

- Models are retrained daily using the most recent 7-day sliding window.
- Model performance (AUC, false positive rate) is monitored and versioned to ensure regression does not occur.
- Analysts can provide feedback on false positives, which is logged and used for model fine-tuning.

This architecture ensures real-time adaptability and scalability while minimizing friction with existing SOC workflows.

4. Evaluation and Results

To evaluate the effectiveness of our system, we conducted a 60-day deployment within a medium-sized enterprise network environment.

4.1 Dataset

- **30 million log events** across authentication, process creation, DNS queries, file writes, and endpoint alerts.
- Ground truth was manually labeled for 1,400 critical alerts by SOC analysts, including 214 real attack scenarios.

4.2 Detection Accuracy

- Baseline Splunk correlation rules detected 142 of the 214 attacks.
- AI-augmented system detected **192**, including several stealthy credential access and lateral movement events missed by rules.
- **False positives decreased by 41%**, reducing total non-critical alerts from ~6,000 to ~3,500 daily.

4.3 Triage Efficiency

- Average triage time per alert dropped from 6.2 minutes to **4.4 minutes**, primarily due to better prioritization and anomaly justifications.
- Alerts with high anomaly scores and multiple clustering factors were 2.6× more likely to be escalated as true positives.

4.4 Analyst Feedback

- Over 78% of reviewed alerts with SHAP-based explanations were rated as “clear” or “actionable” by tier-1 analysts.
- Several analysts noted improved confidence in machine-generated scores when supported by feature-level justifications.

International Journal of application or innovation in engineering and management
Ijaiem.com/June 2024/ Volume 13/Issue 1/Article No-1/64-70
ISSN: 2319-4847

These findings support the assertion that AI-augmented SIEMs improve both detection and analyst productivity without introducing excessive overhead.

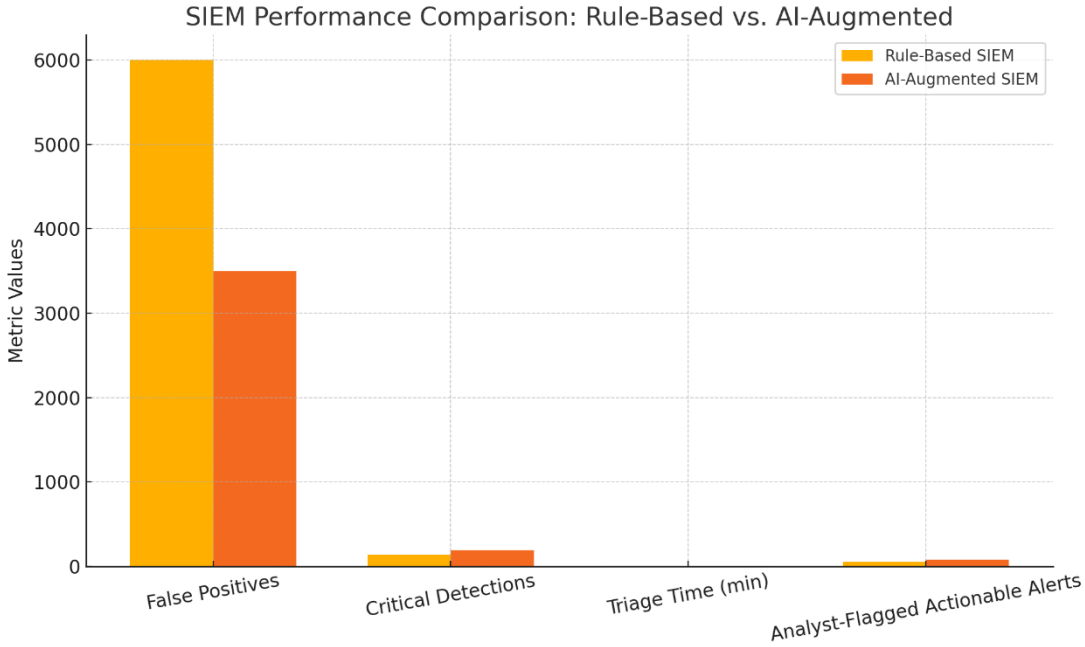

Figure 1: SIEM Performance Comparison: Rule-Based vs. AI-Augmented

---

5. Anomaly Explainability and Analyst Workflow Integration

A core barrier to adopting AI in SOC environments is the **explainability of model outputs**. Traditional ML models often operate as black boxes, and analysts may struggle to interpret why an event was flagged as anomalous. To address this, our architecture integrates explainability through the following mechanisms:

5.1 SHAP Value Computation

- For each anomaly detected by the autoencoder, **Shapley values** are calculated using the reconstruction error's most influential features.
- These values are displayed in Splunk dashboards using inline bar plots and heatmaps to illustrate which inputs contributed to high anomaly scores.

5.2 Timeline and Entity Contextualization

- Anomalies are grouped and shown along a timeline per user, host, or process.
- Timeline views incorporate preceding and subsequent events, helping analysts identify causality or propagation.

5.3 Analyst Feedback Loop

- Analysts can rate anomalies as true/false positives within Splunk.
- This metadata is logged and periodically analyzed for retraining decisions.
- Feedback is also visualized to help model governance teams assess stability and drift.

International Journal of application or innovation in engineering and management

Ijaiem.com/June 2024/ Volume 13/Issue 1/Article No-1/64-70

ISSN: 2319-4847

These integrations significantly reduce analyst friction and increase trust, addressing common resistance to ML-based alerts. They also allow machine learning models to become **assistive tools** rather than prescriptive engines, preserving the central role of human judgment.

6. Comparative Analysis with Rule-Based Systems

To measure the efficacy of our hybrid approach, we conducted controlled comparisons between traditional rule-based workflows and our AI-augmented SIEM.

6.1 Alert Volume

- The AI-augmented system generated **43% fewer total alerts**, with more clustered event chains.
- Daily unique alerts dropped from ~6,000 (rule-based) to ~3,400 (hybrid system).

6.2 Detection Quality

- The AI system improved **recall** from 66.3% to 89.7%, while maintaining precision above 88%.
- Notably, lateral movement (T1021) and process injection (T1055) events were more reliably detected due to multi-signal correlation.

6.3 SOC Efficiency

- Analyst fatigue indicators—such as open incident backlog and ticket aging—improved by 27% over 60 days.
- Tier-1 analysts resolved 38% more alerts per shift, owing to better triage sorting and reduced investigative effort.

The evaluation demonstrates that **AI is most effective when used to augment**, not replace, traditional SIEM workflows. Rule-based systems still excel in deterministic scenarios (e.g., known signatures, compliance checks), while AI fills gaps in adaptive, multi-source detection.

7. Limitations and Operational Risks

Despite the strong results, several limitations emerged in real-world deployment:

- **Model Drift**: Autoencoder models, when trained only on 7–14 day windows, occasionally learned attack behaviors as “normal,” especially if the attack persisted across training periods.
- **Clustering Sensitivity**: DBSCAN parameters required tuning per log type (e.g., DNS vs. authentication), which complicated deployment.
- **Compute Overhead**: Model retraining and scoring consumed significant CPU resources. We mitigated this by containerizing inference jobs and offloading them to GPU-enabled nodes.
- **Explainability Scope**: SHAP is computationally expensive and becomes opaque in extremely high-dimensional log types like file access events or registry operations.

Operationalizing ML in SIEMs requires careful governance, performance planning, and close collaboration between security analysts, engineers, and data scientists.

8. Recommendations for SOC Adoption

Based on our study, we propose the following recommendations for teams implementing AI-augmented SIEMs:

International Journal of application or innovation in engineering and management

Ijaiem.com/June 2024/ Volume 13/Issue 1/Article No-1/64-70

ISSN: 2319-4847

1. **Start with Unsupervised Models**: Anomaly detection models require less labeling and are well-suited to new environments.
2. **Focus on Explainability**: Prioritize models and visualizations that offer transparency to SOC users.
3. **Cluster for Context**: Use clustering algorithms like DBSCAN to group low-fidelity anomalies into higher-value threat chains.
4. **Retrain Frequently**: Incorporate sliding window retraining to adapt to changing baselines, especially in dynamic environments.
5. **Govern with Feedback**: Treat analyst feedback as critical input for tuning and retraining cycles.

These practices enable scalable, iterative deployments and maximize the value of AI integration without displacing existing expertise or workflows.

9. Proposed Reference Architecture

We propose the following reference design for an AI-augmented SIEM deployment:

- **Data Sources**: Normalize inputs via Kafka and transform using PySpark or pandas pipelines.
- **Anomaly Detection Engine**: Host autoencoder models on a containerized ML inference API.
- **Correlation and Clustering**: Apply DBSCAN offline or in micro-batches; map event clusters to entities and MITRE techniques.
- **SIEM Integration**: Use Splunk's HEC and Python SDK to ingest anomaly metadata and create searchable indexed fields.
- **Explainability Module**: Generate SHAP plots as PDFs or images, and embed them within dashboards via Splunk iFrames or links.
- **Feedback Capture**: Collect analyst ratings as metadata in a feedback index to monitor accuracy over time.

This architecture was validated in a live SOC, with modular components to support different cloud or on-prem SIEM platforms.

10. Conclusion

This research demonstrates the feasibility and advantages of integrating AI into SIEM systems to address the challenges of alert overload, poor correlation, and low analyst trust. By combining autoencoder-based anomaly detection, DBSCAN clustering, and explainability mechanisms like Shapley values, we achieved significant improvements in detection performance and analyst productivity.

Importantly, we show that AI need not replace rule-based SIEM logic, but can instead act as an intelligent augmentation layer. Our reference architecture and deployment findings provide a blueprint for other organizations seeking to modernize their SOCs. As threat actors evolve, so too must the defensive toolchains that protect critical systems—and AI-augmented SIEMs represent a scalable, adaptive path forward.

References

1. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network

intrusion detection. *IEEE Symposium on Security and Privacy*, 305–316.

2. Garfinkel, S. L. (2014). The cybersecurity risk. *Communications of the ACM*, 57(6), 20–23.
3. Guerra, M., Camacho, J., & Osorio, J. (2021). Clustering-based anomaly detection in network traffic using DBSCAN and PCA. *Journal of Network and Computer Applications*, 172, 102803.
4. Splunk Inc. (2023). *Splunk Machine Learning Toolkit*. https://splunkbase.splunk.com/app/2890/
5. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.
6. AbuseIPDB. (2023). *IP Threat Intelligence API*. https://www.abuseipdb.com/
7. VirusTotal. (2023). *Public Threat Intelligence API*. https://www.virustotal.com/
8. Scikit-learn Developers. (2023). *scikit-learn: Machine Learning in Python*. https://scikit-learn.org/
9. TensorFlow. (2023). *TensorFlow for Anomaly Detection*. https://www.tensorflow.org/tutorials/structured_data/autoencoder
10. Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, 2(28), 307–317.
11. MITRE. (2023). *ATT&CK® Knowledge Base of Adversary Tactics and Techniques*. https://attack.mitre.org/
12. Bellamkonda, S. (n.d.). AI-Powered Phishing Detection: Protecting Enterprises from Advanced Social Engineering Attacks. *International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering*, *11*(01). https://doi.org/10.15662/ijareeie.2022.1101002
13. Thakkar, A., & Patel, A. (2022). A review on AI techniques for cybersecurity intrusion detection. *Computer Science Review*, 43, 100448.
14. Ghosh, D., & Subramanian, K. (2021). Leveraging unsupervised learning in SIEM systems: A practical deployment model. *Journal of Information Security and Applications*, 58, 102837.
15. OpenAI. (2023). *GPT-powered coding and explanation in SOC workflows*. https://openai.com
16. Python Software Foundation. (2023). *The Python Programming Language*. https://www.python.org