Security Incident and Response Platforms
Threat Intelligence Platforms
SOAR
Security Orchestration and Automation

Source: D3 Security, based on Gartner's SOAR framework.

From Prediction to Attribution: Integrating Unsupervised Learning for Enhanced SOAR Triage

Anetta Nichols
Doctoral Candidate, D.Eng. in Cybersecurity Analytics
George Washington University

Abstract

This article explores the evolution of a Security Orchestration, Automation, and Response (SOAR) system from rudimentary URL classification through intelligent threat actor attribution. By integrating unsupervised learning into the Mini SOAR framework, realistic threat profiles are simulated, and clustering methods are applied to identify probable adversaries behind malicious URLs. The result is a cognitive SOAR solution that improves triage, provides contextual awareness for action, and informs strategic decision-making in today's Security Operations Centers (SOCs).

Introduction

SOCs are inundated with alerts triggered by firewalls, intrusion detection systems, endpoint monitors, and other security tools. Most of these alerts reduce complex threat behaviors into simple binary labels: malicious or benign. While this kind of classification is essential for initial filtering, it remains insufficient. The critical question becomes: who might be behind it?

This study is informed by a background in data analytics and a current focus on cybersecurity analytics, emphasizing reproducible methodology and forensic clarity in threat attribution.

Attribution adds context. It transforms a simple alert into a narrative that SOCs can act on. By identifying the likely threat actor profile behind a malicious URL, SOCs can prioritize incidents, tailor responses, and anticipate attacker behavior. SOAR platforms, as defined by Palo Alto Networks (2023), combine orchestration, automation, and response capabilities to streamline security operations and reduce manual effort in incident handling. The Cognitive SOAR project enhances the Mini SOAR application by integrating unsupervised learning to shift from prediction to attribution.

Methodology

Feature Engineering: Simulating Threat Actor Behavior. To support attribution, the synthetic dataset was strategically expanded to simulate three distinct threat actor profiles: state-sponsored groups, organized cybercriminals, and hacktivists. Through simulation, the dataset captures distinct behavioral patterns and operational tactics, enabling more nuanced analysis and informative clustering.

- State-sponsored actors exhibit a high degree of sophistication, often leveraging valid SSL certificates and employing deceptive techniques such as prefix/suffix manipulation to mask intent. Their operations are discreet and deliberate, crafted to slip past defenses and focus on high-value targets with precision.
- Organized cybercriminals favor noisy, high-volume tactics. They frequently use shortened URLs, IP addresses, and abnormal structures to overwhelm defenses and exploit vulnerabilities at scale. Their motivations are typically financial, and their methods are aggressive and opportunistic.
- Hacktivists, fueled by ideology, strike opportunistically using an unpredictable mix of techniques. To simulate this, the feature has_political_keyword was introduced to flag politically motivated content. Their hybrid approach—borrowing methods from other threat actors—makes them especially difficult to identify and trace.

Each profile was designed to reflect real-world patterns, enabling clustering algorithms to uncover meaningful groupings.

Algorithm Selection: Why K-Means? K-Means was selected for clustering due to its effectiveness in identifying well-separated, evenly sized groups—exactly what the synthetic data was designed to produce. Unlike DBSCAN, which excels at finding arbitrarily shaped clusters and noise, K-Means aligns with the goal of attributing malicious URLs to one of three clear actor profiles (Scikit-learn, 2023).

Two separate PyCaret workflows were implemented. The classification workflow used pycaret.classification.setup() with the actor_profile as the target variable. The clustering workflow used pycaret.clustering.setup() on feature-only data, excluding the label column. This dual-model architecture enables detection of malicious URLs followed by attribution to a likely threat actor.

Implementation: Inside the Streamlit App

The enhanced Streamlit application follows a two-step logic. First, when a user submits URL features, the app predicts whether the URL is malicious or benign using the trained classification model. Second, if the URL is malicious, the app passes the features to the clustering model. The resulting cluster ID is mapped to a threat actor profile (e.g., Cluster 0 = "Organized

Cybercrime"). A dedicated “Threat Attribution” tab displays the actor profile and a short description of typical tactics and motivations.

Additional enhancements include a test case dropdown to simulate known attack patterns, optional display of cluster distance for analytical depth, and a feature importance plot to visualize the most influential features in classification.

Results and Discussion

The clustering model successfully grouped malicious URLs into three distinct clusters, each aligning with the synthetic profiles that were engineered. This allowed for meaningful attribution. URLs with noisy, aggressive features were consistently mapped to organized cybercrime. Subtle, well-crafted URLs with deceptive traits aligned with state-sponsored actors. URLs containing political keywords and mixed tactics pointed to hacktivist behavior.

Attribution capability provides several operational advantages. The triage process is expedited through the prioritization of incidents aligned with specific threat actor profiles. This intelligence informs response strategies by aligning containment and escalation procedures with the adversary’s underlying motivations and operational methods. In parallel, SOCs are equipped with these capabilities through advanced strategic planning, supported by the analytical foresight needed to anticipate evolving attack trajectories and allocate resources with greater precision and effectiveness.

Yet, significant challenges persist, demanding continued vigilance and adaptation. Synthetic data, while useful for prototyping, may not capture the full complexity of real-world threats. Clustering is inherently unsupervised, and misattribution is a risk, particularly for edge cases or hybrid tactics. Scalability is also a concern, as attribution logic must evolve to handle more actor types and increasingly sophisticated attack patterns.

Development Reflection

While the Streamlit interface delivers real-time attribution, its underlying architecture was built through modular, audit-ready components. The classification and clustering workflows were developed in isolated .py scripts, each annotated for forensic traceability. Synthetic data generation was handled via dedicated .ipynb notebooks with timestamped outputs, while markdown files (README.md, INSTALL.md) documented feature engineering, model selection, and implementation logic. Supporting artifacts included .csv datasets, .pkl model files for reproducible inference and clustering.

Challenges included calibrating clustering logic to match synthetic actor profiles, managing hybrid tactics, and ensuring consistent mapping between cluster IDs and actor labels. These were addressed through iterative testing, feature importance analysis, and narrative-driven justification, reinforcing the analytic integrity of the final deliverable.

Conclusion

Binary classification is no longer sufficient in the face of complex, evolving threats. Attribution provides the context SOCs need to respond intelligently. By integrating unsupervised learning into the Mini SOAR framework, a powerful step is taken towards Cognitive SOAR a system that doesn't just detect threats but understands them.

Through synthetic data generation, informative feature engineering, and clustering algorithms, the triage process was optimized to empower security analysts with actionable insights. Individual analytic decisions were documented with forensic clarity, ensuring that the deliverable is not only technically sound but also reproducible and defensible. The modular architecture, timestamped outputs, and rubric-aligned justification reflect a commitment to transparency that is essential for operational trust.

This project demonstrates how synthetic data, informative feature engineering, and clustering algorithms can be applied to optimize the triage process and empower security analysts. Continuous application of future iterations could incorporate real-world threat intelligence feeds, anomaly detection, and hybrid modeling to further refine attribution.

References

CISA. (2021). *Shareable SOAR workflows (Version 0.6522022)* [Software]. GitHub. https://github.com/cisagov/shareable-soar-workflows/releases

D3 Security. (2022, May). *Elements of SOAR according to Gartner* [Image]. https://d3security.com/wp-content/uploads/2022/05/elements-of-soar-according-to-gartner-sirp-tip-soa.jpg

George Washington University**.** (2025). *SEAS 8414_DC8 Week-8 assignment brief* [Course materials].

MITRE ATT&CK**.** (2023). *Adversarial tactics, techniques, and common knowledge*. https://attack.mitre.org

Palo Alto Networks. (2023). *What is SOAR?* https://www.paloaltonetworks.com/cyberpedia/what-is-soar

PyCaret. (2023). *PyCaret documentation*. https://pycaret.org

Scikit-learn**.** (2023). *Clustering algorithms*. https://scikit-learn.org/stable/modules/clustering.html