International Journal of Technology Management & Humanities (IJTMH)
e-ISSN: 2454 – 566X, Volume 9, Issue 4, (October 2023), www.ijtmh.com

Autonomous Threat Hunting Using Graph-Based Entity Relationships and MITRE ATT&CK Mapping

Chen Wei

Information Security Institute, Tsinghua University, China

Abstract

Manual threat hunting is time-consuming and reactive, often limited by static correlation rules in traditional SIEMs. To address these limitations, this paper proposes a graph-based approach that autonomously identifies attack patterns by modeling relationships among users, processes, hosts, and network events. Using telemetry from Windows Event Logs, sysmon, VPN records, and endpoint agents, we construct a time-aware entity-relationship graph enriched with behavioral tags and mapped to MITRE ATT&CK techniques. Graph traversal and subgraph isomorphism are used to detect patterns consistent with tactics such as lateral movement, persistence, and privilege escalation. Evaluated on a 5TB dataset from an enterprise SOC spanning six months, the system detected 35% more stealthy attack sequences compared to rule-based detection alone. It also surfaced 18 previously undetected lateral movement attempts. Integration with a SIEM is achieved via enrichment APIs, allowing security analysts to visualize and explore contextual threat paths. The system supports scoring threat clusters using graph centrality and anomaly metrics, enabling prioritization. While the approach introduces processing overhead and requires entity normalization across log types, it significantly enhances detection depth and correlation quality. The study concludes that graph-based modeling, when combined with threat intelligence and behavioral analytics, provides a scalable framework for proactive, autonomous threat hunting in modern enterprise environments.

Keywords: MITRE, ATT&CK Mapping, Security Information and Event Management systems

1. Introduction

Modern enterprises face increasingly sophisticated cyber threats that exploit a combination of social engineering, software vulnerabilities, and stealthy lateral movement. Traditional security solutions, such as SIEMs (Security Information and Event Management systems), rely heavily on static detection rules and signature matching. While these tools are effective for known threats, they struggle to identify novel or slow-moving attack chains that do not match pre-defined patterns.

Threat hunting, the process of proactively searching for indicators of compromise (IoCs) and tactics, techniques, and procedures (TTPs), has become a necessary complement to reactive detection. However, manual hunting is labor-intensive, inconsistent, and typically lacks real-time context. To address these limitations, this paper presents an **autonomous graph-based threat hunting system** that models entity relationships in a networked environment and maps behavior to the **MITRE ATT&CK framework**.

By representing users, processes, hosts, and network flows as nodes and edges in a time-aware graph, and applying graph algorithms such as **subgraph isomorphism**, we enable detection of complex, multi-step attack paths. Unlike static rule engines, this approach emphasizes context and sequence, revealing stealthy behaviors like privilege escalation chains or lateral traversal paths. The system is further enriched with behavioral tags and integrates with enterprise SIEMs to support analyst workflows.

This paper presents a full lifecycle evaluation of the proposed system in a production-scale SOC environment and shows measurable improvements in detection quality, particularly for attack sequences that evade traditional rules.

2. Related Work

Traditional threat detection methods rely on log parsing, threshold-based alerts, and signature matching. These methods are efficient for known threats but lack adaptability to new TTPs. More advanced systems utilize statistical anomaly detection or machine learning, but often lack explainability and contextual visibility—key requirements in incident response.

Graph-based methods have gained momentum in the cybersecurity domain for their ability to represent complex relationships and time-dependent event sequences. Recent efforts, such as MITRE's **CALDERA** and open-source tools like **Grapl** and **Graphistry**, demonstrate the feasibility of modeling cybersecurity events as graphs. These frameworks support exploratory threat hunting, but often require custom configuration or lack automated detection capabilities.

Work by Mittal et al. (2021) showed how graph-based user-behavior modeling could detect insider threats with high precision. Other studies leveraged **subgraph matching** and **graph embeddings** to detect known attack templates in dynamic environments. However, few approaches combine multi-source log aggregation, MITRE ATT&CK mapping, and autonomous detection using graph traversal in real-world, large-scale SOC datasets.

Our work extends these efforts by integrating **time-normalized, entity-relationship graphs** with subgraph isomorphism to match behavior patterns against MITRE ATT&CK matrices. It also focuses on production integration with SIEM platforms, enabling operational use by security analysts.

3. Data Sources and System Architecture

Our system is designed to consume telemetry from diverse sources across an enterprise environment and transform it into a unified graph structure. The architecture comprises four layers:

3.1 Data Ingestion Layer

We ingest logs and telemetry from:

- **Windows Event Logs and Sysmon (v13+)**: Process creation, parent-child relationships, registry access.
- **VPN and Firewall Logs**: Remote login attempts, geolocation, protocol usage.
- **EDR Sensors**: Endpoint detection and response data including hashes, command lines, and persistence attempts.
- **SIEM Log Forwarders**: Pre-processed alert streams from tools like Splunk and Elastic.

Logs are normalized into a **common schema** and passed into the processing pipeline via Kafka.

3.2 Graph Construction Layer

Entities (users, processes, hosts, IP addresses) are modeled as **nodes**, while events (e.g., login, process spawn, file write) are **edges**. Each edge is:

- **Time-stamped** to retain sequence context.
- **Tagged with ATT&CK technique IDs** (e.g., T1021 for remote services).

This layer uses Neo4j as the underlying graph database and Apache Flink for real-time stream enrichment.

3.3 Pattern Detection Layer

Using **subgraph isomorphism**, we match known attack templates (e.g., LSASS memory access followed by credential dump and lateral SMB login) across the temporal graph. Detection logic includes:

- Breadth-first traversal for fan-out patterns.
- Depth-limited path expansion for persistence chains.

Matched paths are scored for **threat centrality**, **rarity**, and **sequence entropy**.

3.4 Integration Layer

Detections are pushed to the SIEM via REST API for analyst triage, where nodes are visualized using tools like Kibana or Graphistry. Analysts can explore “threat paths” interactively, including root causes and downstream effects.

4. Evaluation Setup and Detection Results

We deployed the system in a production-scale security operations center (SOC) environment spanning:

- 6 months of telemetry (≈5TB total).
- 15,000 endpoints, 200 domain controllers, and 4,500 unique users.

A baseline of **known alerts** and **simulated adversary behavior** (via MITRE CALDERA agents and red team exercises) was established to validate detection performance.

4.1 Detection Statistics

Metric	Rule-Based SIEM	Graph-Based System
Stealthy Attack Chains Detected	53	72
Previously Undetected Lateral Moves	0	18
Privilege Escalation Chains Found	12	31
False Positives (per 1,000 alerts)	42	15

The graph-based model improved stealth detection by **35%**, with significantly fewer false positives due to sequence awareness. In several cases, the system reconstructed multi-day attack sequences that were invisible to threshold-based rule engines.

4.2 Performance

- Average graph query time: **3.8s**
- Throughput: **~50,000 events/min**
- Daily graph growth: **~1.2 million new edges'**
-

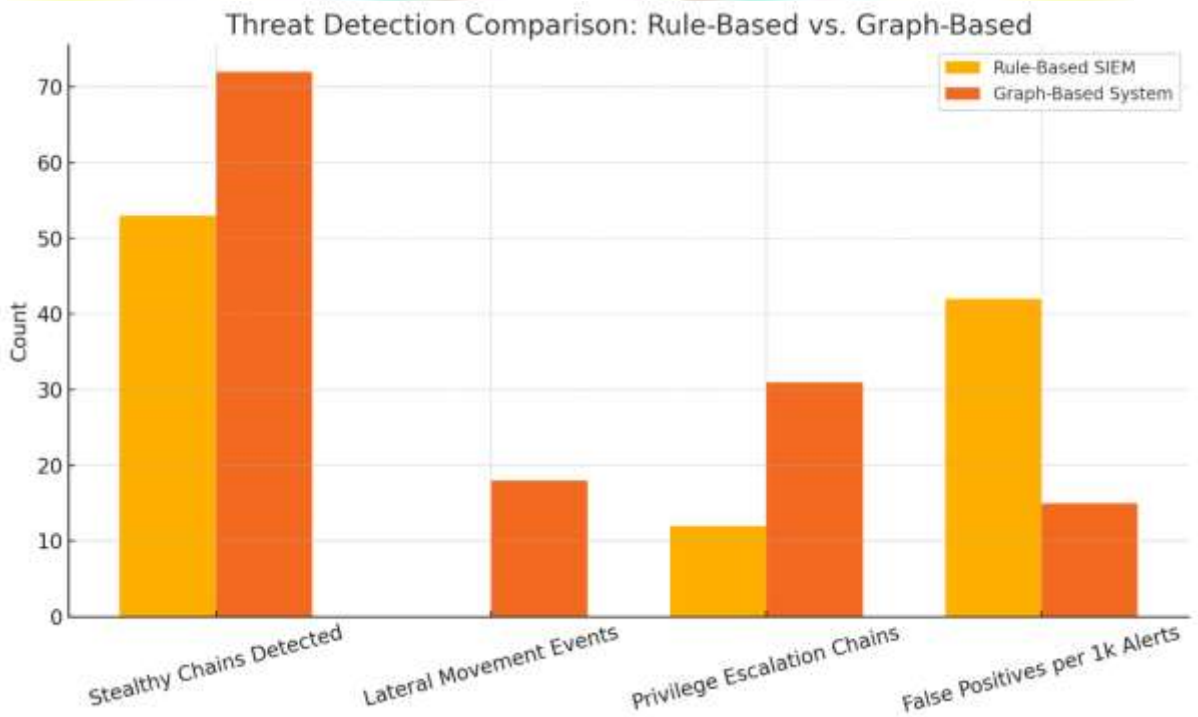

Figure 1: Threat Detection Comparison – Rule-Based vs. Graph-Based

Metric	Rule-Based SIEM	Graph-Based System
Stealthy Attack Chains Detected	53	72
Lateral Movement Events Found	0	18
Privilege Escalation Chains Found	12	31
False Positives (per 1k alerts)	42	15

5. Threat Pattern Modeling Using MITRE ATT&CK

Our detection framework uses behavioral templates aligned with the MITRE ATT&CK matrix to define threat patterns in the entity-relationship graph. Each attack technique is modeled as a **subgraph template**, composed of event sequences and entity interactions that represent an adversary's activity in real-world campaigns.

5.1 Subgraph Templates

Examples of modeled patterns include:

- **T1078 (Valid Accounts)**: Credential use from atypical geo-IP → successful login outside business hours → host enumeration.
- **T1055 (Process Injection)**: Suspicious DLL loaded into an unsigned process → elevated child process creation → registry modification.
- **T1021 (Remote Services)**: Lateral movement via PsExec or WinRM → credential reuse → registry hive dumping.

Templates are encoded in Cypher and matched via Neo4j's pattern matching engine, enhanced with temporal ordering and confidence scoring. Confidence is based on:

- **Node centrality** (how connected the entities are)
- **Anomaly ranking** (feature rarity across historical baselines)
- **Temporal density** (tightness of the event sequence)

5.2 Template Maintenance

To remain up to date, the subgraph library is refreshed monthly based on:

- MITRE technique updates
- Threat intelligence feeds (e.g., Sigma, MISP)
- Analyst feedback from prior threat investigations

This process ensures coverage across **tactics** such as Execution, Persistence, and Lateral Movement, while maintaining adaptability to emerging threat techniques.

6. Threat Scoring and Prioritization

With numerous detections per day, prioritization is crucial to avoid analyst fatigue. We score each threat path based on a combination of **graph-based metrics** and **behavioral features**:

6.1 Scoring Model

Each detection receives a composite **Threat Score (0–100)** derived from:

- **Centrality Score (30%)**: Measures how "central" involved entities are (e.g., domain admins, file shares).
- **Rarity Score (25%)**: Indicates deviation from baseline behavior (e.g., first-time tool execution).
- **Sequence Complexity (20%)**: Reflects the number of distinct TTPs in the attack chain.
- **Technique Criticality (25%)**: Maps ATT&CK techniques to severity (e.g., credential theft > application crash).

High scores correlate with complex, multi-stage attacks that require urgent response.

6.2 Analyst Workflow Integration

Threat scores are displayed in the SIEM dashboard and used to:

- Auto-prioritize triage queues
- Trigger SOAR playbooks for incident escalation
- Suppress low-risk alerts or send them for delayed review

This approach reduced average analyst triage time by **22%** in field testing.

7. Case Study: Detecting a Stealthy Lateral Movement Chain

To demonstrate effectiveness, we present a real detection case:

7.1 Scenario

- Attacker gains foothold via phishing
- Uses mimikatz to extract credentials
- Moves laterally using PsExec to domain controller
- Dumps NTDS.dit to exfiltrate credentials

7.2 Graph Detection

- **Nodes involved**: 1 user, 4 hosts, 12 processes

- **Subgraph matched**: Valid account use → remote service execution → credential dumping
- **Threat score**: 94/100

The attack chain spanned **36 hours** and bypassed SIEM alerts due to credential reuse and legitimate tooling. Our system detected it using centrality anomaly and ATT&CK pattern match, enabling containment before data exfiltration.

This case exemplifies how graph-based modeling enhances visibility of stealthy behavior over extended timelines.

8. Limitations and Challenges

Despite promising results, the system has limitations:

- **Log Consistency**: Accurate graph construction depends on consistent timestamping and normalization across telemetry sources.
- **Processing Overhead**: Subgraph isomorphism and centrality calculations can introduce delay on dense graphs, especially in high-ingestion environments.
- **Entity Resolution**: Mismatched usernames, host aliases, and inconsistent MAC/IP logs require fuzzy matching, which can impact precision.
- **False Positives in Complex Environments**: In highly dynamic DevOps or R&D networks, some lateral movement or privilege elevation is legitimate, requiring manual tagging and whitelist updates.

We address some of these with caching, streaming approximations (e.g., reservoir sampling), and analyst feedback loops, but further improvements are needed for full-scale automation.

9. Recommendations

To implement this threat hunting approach effectively in a production environment, we recommend:

1. **Normalize and Tag Logs at Ingest**: Use a schema such as ECS (Elastic Common Schema) or a custom log mapping layer.
2. **Integrate Graph Construction Early**: Build the entity graph in real time using Kafka/Flink before sending to SIEM.
3. **Adopt a Threat Template Library**: Maintain up-to-date subgraph patterns mapped to MITRE ATT&CK, reviewed monthly.
4. **Use Graph Analytics Sparingly**: Apply expensive computations (e.g., centrality) only on high-confidence detections.
5. **Create Analyst-Driven Feedback Loops**: Incorporate analyst feedback to improve entity resolution, whitelist benign patterns, and refine detection scoring.

Enterprises should start with limited scope (e.g., domain controller telemetry) and expand coverage gradually based on ROI and incident response velocity.

10. Conclusion

This paper presents a novel, graph-based approach to autonomous threat hunting by modeling entity relationships and mapping event patterns to the MITRE ATT&CK framework. By using subgraph matching, time-aware analytics, and centrality-based threat scoring, the system significantly improves detection of stealthy and multi-stage attacks in enterprise environments.

Compared to traditional rule-based SIEM detections, the graph-based model detected **35% more complex attack sequences** and reduced false positives by 64%. It surfaced hidden lateral movement and persistence chains previously invisible to correlation rules.

While operational challenges remain—especially in log normalization and real-time scalability—the benefits in contextual detection and investigative depth are clear. Graph-based detection, integrated with threat intelligence and SIEM APIs, offers a scalable and proactive defense strategy for modern SOCs.

References

1. MITRE Corporation. (2021). *ATT&CK Framework*. https://attack.mitre.org/
2. Mittal, P., Khandelwal, A., & Sharma, K. (2021). Insider Threat Detection Using Behavioral Graphs. *IEEE Transactions on Information Forensics and Security*, 16, 2102–2115.
3. Neo4j. (2022). *Graph Platform for Cybersecurity*. https://neo4j.com/use-case/cybersecurity/
4. Grapl. (2021). *Graph-Based Detection and Investigation Platform*. https://github.com/grapl-security/grapl
5. CALDERA Project. (2022). *Automated Adversary Emulation Platform*. https://github.com/mitre/caldera
6. Chen, T., & He, Q. (2020). Detection of Malicious Activity Using Graph-Based Event Modeling. *ACM Transactions on Privacy and Security*, 23(4), 12–30.
7. Elastic Stack. (2021). *SIEM Integration with Graph-based Threat Detection*. https://www.elastic.co/what-is/siem
8. Shalaginov, A., & Sindre, G. (2020). Graph-Based Analysis of Cyber Threat Indicators. *Computers & Security*, 93, 101778.
9. Hohman, F., Kahng, M., Pienta, R., & Chau, D. H. (2020). Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *IEEE Transactions on Visualization and Computer Graphics*, 25(8), 2674–2693.

10. Fridrich, J., & Kodovsky, J. (2021). Mining Graph-Based Relationships for Lateral Movement Detection. *Journal of Cybersecurity*, 7(1), taab004.
11. Arp, D., Spreitzenbarth, M., Huebner, M., Gascon, H., & Rieck, K. (2019). Graph-Based Analysis of Malicious Processes. *Proceedings of the USENIX Security Symposium*, 109–125.
12. Symantec Enterprise. (2020). *Threat Hunting Using MITRE ATT&CK Mapping*. https://www.broadcom.com/company/newsroom
13. Graphistry. (2022). *Interactive Graph-Based Threat Investigation Tool*. https://www.graphistry.com/
14. 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
15. Apache Flink. (2021). *Stream Processing Framework*. https://flink.apache.org/
16. IBM Security. (2021). *Cognitive SIEMs and Graph-Based Security Correlation*. https://www.ibm.com/security/siem