International journal of imaging science and engineering

ISSN: 1934--9955 www.ijise.net

Vol-19 Issue-02 Sep 2024

Post-Compromise Behavior Analysis in Enterprise Networks Using Graph Anomaly Detection

Lohith Vanama

Integration Specialist, Dienst Uitvoering Onderwijs, North Holland, Netherlands

Abstract

After initial compromise, adversaries often rely on stealthy lateral movement and privilege escalation to complete their objectives. Detecting such activity requires correlating diverse events across users, systems, and time. This paper introduces a graph-based anomaly detection approach for identifying post-compromise behaviors in enterprise networks. Using authentication logs, Active Directory data, VPN access logs, and process trees, we build dynamic user-resource interaction graphs. These are processed using GraphSAGE and subgraph matching algorithms to identify deviations from normal communication paths and resource usage patterns. The system is evaluated on real-world datasets from two enterprise SOCs and enriched with simulated attack paths involving techniques from the MITRE ATT&CK framework. Our model achieves a precision of 89% and recall of 84% in identifying suspicious privilege escalation chains and unusual remote desktop access paths. Time-aware embeddings improve performance by modeling access sequences across day/night cycles and weekdays. Visualizations help SOC analysts trace anomalies back to initial access points and monitor for recurring patterns. We discuss operational challenges such as log normalization, identity correlation across domains, and graph storage efficiency. The study demonstrates that graph-based analysis is highly effective for identifying subtle, stage-2 attack behaviors and offers a scalable enhancement to EDR and SIEM platforms when integrated into a modern SOC workflow.

1. Introduction

Post-compromise activity represents one of the most dangerous stages of a cyberattack. After an adversary gains initial access—whether through phishing, malware, or stolen credentials—they often pivot laterally across an enterprise network, escalate privileges, and access high-value assets. Detecting these activities is inherently difficult because they blend in with legitimate user behavior and are rarely flagged by signature-based systems. Traditional endpoint detection and response (EDR) tools and security information and event management (SIEM) systems focus heavily on per-event or rule-based alerting, which struggles to identify multistep, low-and-slow attacks.

To address this gap, we present a graph-based anomaly detection approach that models user, system, and resource interactions over time. By building dynamic, heterogeneous graphs from log sources such as Active Directory authentication records, VPN connection histories, process

International journal of imaging science and engineering
ISSN: 1934--9955 www.ijise.net
Vol-19 Issue-02 Sep 2024

execution trees, and file access patterns, we can capture relational behaviors rather than isolated events. These graphs are analyzed using a combination of GraphSAGE—a graph neural network model for inductive representation learning—and subgraph matching algorithms tuned to detect suspicious access paths.

This research aims to identify subtle behavioral anomalies indicative of post-compromise actions. Our results demonstrate that graph-based modeling provides significant improvements in detecting lateral movement and privilege escalation chains, with high precision and operational interpretability. This paper details the system design, experimental setup, evaluation metrics, and integration roadmap for modern security operations centers (SOCs).

2. Related Work

Traditional post-compromise detection methods rely heavily on heuristic rules or behavioral baselines computed over single entities (e.g., user login frequency, process spawning rates). While these methods offer low overhead, they lack the relational context needed to detect sophisticated attack paths. Signature-based systems like Snort and Suricata cannot detect novel privilege escalations unless the attack matches a known pattern.

Recent advances in graph-based security analysis show promise in uncovering attack patterns across entities and time. Graph approaches have been used to model provenance in data flow, map lateral movement paths, and visualize identity sprawl in Active Directory. Techniques like graph traversal and spectral clustering have been applied in detecting anomalies in large-scale enterprise telemetry.

Graph neural networks (GNNs), particularly GraphSAGE and GCNs, have been increasingly used in cybersecurity for tasks such as malware classification and access prediction. GraphSAGE's inductive learning ability makes it suitable for evolving enterprise networks where new nodes and edges are constantly added.

MITRE ATT&CK provides a taxonomy for common adversarial behaviors, which many modern threat detection systems use to enrich alert context. However, mapping low-level events to ATT&CK tactics often requires correlation across multiple log sources and entity relationships—an area where graph methods excel.

Our work builds on these foundations by combining graph-based representation learning, ATT&CK mapping, and anomaly scoring to detect post-compromise behavior in real-time enterprise environments.

3. Methodology

3.1 Data Sources and Preprocessing

International journal of imaging science and engineering
ISSN: 1934--9955 www.ijise.net
Vol-19 Issue-02 Sep 2024

We collect and normalize logs from:

- **Active Directory (AD)**: Kerberos authentication events, logon types, group membership changes.
- **VPN logs**: Remote access sessions with geolocation and device fingerprinting.
- **Endpoint logs**: Process execution trees, file creation, and registry access.
- **Network logs**: Internal RDP/SMB connections and data transfers.

Log sources are normalized using a unified schema and enriched with context (e.g., user role, device type, time zone). Identity correlation is applied to map multiple login identities (e.g., UPN vs. SID) to a canonical entity.

3.2 Graph Construction

We construct time-aware, directed graphs where:

- **Nodes** represent users, systems, sessions, and resources.
- **Edges** represent authenticated access, process relationships, file usage, or network connections.
- Edge weights are time-decayed to prioritize recent activity.
- Temporal windows (6-hour sliding) are used to construct dynamic graphs over time.

3.3 Learning and Detection

We employ **GraphSAGE** to compute node embeddings based on neighborhood aggregation. Embeddings are updated at each time window, enabling tracking of behavioral drift.

Anomaly detection is performed in two stages:

1. **Subgraph Matching**: Comparing observed subgraphs to known benign patterns.
2. **Embedding Outlier Detection**: Applying Isolation Forest on embedding space to flag anomalous users or access paths.

Graph embeddings are stored in a vector database for efficient similarity lookup and historical analysis.

4. Experimental Setup and Evaluation Criteria

4.1 Environment and Dataset

We evaluate our system using:

- A **real-world dataset** from two enterprise SOCs over six months (anonymized, 5.2 TB).

International journal of imaging science and engineering
ISSN: 1934--9955 www.ijise.net
Vol-19 Issue-02 Sep 2024

- **Simulated attack paths** generated in a lab using Red Team tools (Cobalt Strike, Empire) following MITRE ATT&CK techniques (T1078, T1021, T1086, etc.).
- **Baseline comparison** against log anomaly detection (z-score), EDR event correlation, and static graph clustering.

4.2 Evaluation Metrics

We assess performance using:

- **Precision and Recall**: For correctly identifying malicious post-compromise behaviors.
- **Graph anomaly score**: Normalized ranking of node/path outliers.
- **Detection delay**: Time between behavior occurrence and detection.
- **Analyst usability**: Time to root cause analysis based on graph visualizations.

Our system achieves:

- **89% precision**, **84% recall**, and an average detection delay of **5.2 minutes**.
- 18 previously undetected incidents confirmed through SOC validation.

5. Results

The graph-based detection system yielded strong results across both real and simulated datasets. When evaluated against a labeled ground truth comprising confirmed attack sequences and benign sessions, the model achieved an **overall precision of 89%** and a **recall of 84%** for detecting post-compromise behavior. Among detected behaviors were **unusual remote desktop access**, **unexpected inter-domain authentications**, and **privilege escalation chains** not flagged by conventional EDR systems.

Time-aware node embeddings improved detection of **time-sensitive behaviors**, particularly access patterns deviating from organizational norms during off-hours. For example, a lateral movement campaign using PowerShell remoting over WinRM was successfully identified by detecting deviations in host interaction sequences and domain controller access at 3:17 AM.

Compared to static anomaly detection based on statistical baselines, the graph-based approach **reduced false positives by 43%**, leading to more actionable alerts. Detection latency averaged **5.2 minutes** post-behavior onset, enabling SOC analysts to intervene before further propagation.

Analysts validated 18 incidents that had previously gone undetected, including use of stale administrator credentials and resource access from unmanaged devices. Visual correlation tools enabled clear tracing of these incidents from entry point to objective, enhancing situational awareness.

International journal of imaging science and engineering
ISSN: 1934--9955 www.ijise.net
Vol-19 Issue-02 Sep 2024

6. Discussion

The observed results confirm that **graph anomaly detection is an effective means of surfacing multi-stage post-compromise activity**. Unlike flat event-based detection, the graph-based approach inherently models relationships and context—providing a significant advantage in detecting **low-noise, multi-hop attacker behavior**.

One strength of this approach is its ability to combine **diverse telemetry sources** and track attacker movement across siloed systems. The dynamic construction of time-bounded graphs ensures freshness and reduces memory overhead. The use of GraphSAGE for inductive representation learning allows the system to generalize to **previously unseen users, hosts, and subgraphs**, supporting deployment in evolving environments.

Challenges were also encountered:

- **Log inconsistencies** across systems (e.g., missing logoff events or duplicate user sessions) impacted graph completeness.
- **Graph storage and query latency** became significant as node and edge counts exceeded 10 million in larger enterprises.
- **Identity stitching** errors occasionally misclassified legitimate behavior as anomalous when a user used multiple credentials or devices.

Despite these limitations, the system provides high contextual fidelity and integrates well with existing SOC platforms for threat hunting and triage.

7. Limitations

While promising, the proposed system has limitations:

- **High computational cost** of subgraph matching and embedding training on large graphs requires GPU acceleration for real-time use.
- **Graph drift** from rapidly changing environments (e.g., mergers, remote workforce shifts) can reduce model stability.
- **No ground truth for zero-day behaviors** means some anomalies could go unverified or misinterpreted.

Moreover, **partial logging coverage**, especially from unmanaged endpoints and mobile devices, can lead to blind spots in graph construction. Accuracy of detection is also sensitive to the **granularity of logging**, particularly with regard to file system activity and memory-level process relationships.

Lastly, while GraphSAGE generalizes well, **explainability remains a challenge**, requiring visualizations and attention mechanisms to help analysts interpret why a node or edge was flagged.

International journal of imaging science and engineering
ISSN: 1934--9955 www.ijise.net
Vol-19 Issue-02 Sep 2024

8. Conclusion

This study demonstrates that **graph-based modeling of user-resource interactions provides a powerful approach to detecting post-compromise behaviors** in enterprise networks. By leveraging GNNs and time-aware subgraph analytics, the system uncovers suspicious paths, anomalous access sequences, and privilege escalations that traditional systems often miss.

With **89% precision and reduced false positives**, this solution enhances both detection capability and analyst productivity. Its modularity allows it to complement existing SIEMs and EDRs while adding a novel behavioral detection layer grounded in structural and temporal relationships.

The findings encourage broader adoption of graph analytics in security operations and emphasize the need for **multimodal correlation**, bridging identity, access, and network data into a unified model of behavior.

9. Future Work

Future enhancements will focus on:

- Integrating **attention-based GNNs** (e.g., GAT) for better explainability.
- Developing **graph compression techniques** for efficient long-term storage and querying.
- Correlating graph anomalies with **threat intelligence feeds** to boost contextual prioritization.
- Incorporating **streaming graph updates** to reduce latency for high-velocity environments.
- Extending to **cross-cloud deployments**, enabling consistent detection across hybrid infrastructures.

Research is also underway to develop **analyst-in-the-loop feedback systems**, allowing SOC personnel to label nodes or paths as benign or malicious, improving model precision over time through semi-supervised refinement.

10. References (APA Style)

1. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60, 19–31.
2. Bhattacharya, S., & Chen, Z. (2022). A survey on graph-based anomaly detection in cyber security. *ACM Computing Surveys*, 55(3), 1–37.

International journal of imaging science and engineering
ISSN: 1934--9955 www.ijise.net
Vol-19 Issue-02 Sep 2024

3. Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD*, 855–864.
4. Hamilton, W., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 1025–1035.
5. Hutchinson, B., & Singh, P. (2023). Graph neural networks for cybersecurity: Applications and challenges. *IEEE Transactions on Network and Service Management*, 20(1), 3–15.
6. Neubauer, C., Brocke, J. V., & Ziebarth, T. (2021). Visual analytics for security incident response using graph-based models. *Computers & Security*, 105, 102242.
7. Pastrana, S., & Thomas, D. R. (2020). Understanding the dynamics of cyberattacks through graph analysis. *IEEE Access*, 8, 146437–146450.
8. Rahman, M. S., & Muttukrishnan, R. (2023). Real-time detection of advanced persistent threats using dynamic graph embeddings. *Journal of Information Security and Applications*, 72, 103454.
9. Yuan, X., Lu, Y., & Li, X. (2017). A deep learning-based approach for intrusion detection. *IEEE International Conference on Big Data*, 468–475.
10. Bellamkonda, S. Network Segmentation and Micro-Segmentation: Reducing Attack Surfaces in Modern Enterprise Security.