Provenance-based Cloud Application Detection and Response (P-CADR)

PI: Wajih Ul Hassan, Assistant Professor, Department of Computer Science, University of Virginia

Cash funding needed: $75,000 USD
AWS Promotional Credits needed: $25,000 USD

Abstract

Modern cloud-native applications create sophisticated, multi-layered attack surfaces that traditional security tools struggle to defend effectively. Tools such as CNAPPs, EDRs, and cloud logging solutions typically operate in isolation, producing fragmented alerts and failing to detect or explain complex, multi-stage threats, such as MOVEit and Log4Shell. We propose Provenance-based Cloud Application Detection and Response (P-CADR), constructing Cross-Layer Provenance Graphs (CPGs) for unified causal reasoning across application, kernel, and cloud layers. Our approach includes an eBPF-based observability layer for real-time telemetry, advanced event correlation for real-time CPG construction, and an anomaly detection engine using supervised Graph Neural Networks. We will integrate embedding recycling for efficiency and explainability tools for analyst validation. By adopting P-CADR, organizations gain unprecedented clarity into their cloud environments, enabling precise, actionable, and proactive threat management.

Keywords: Threat Detection; Cloud Security; Graph Neural Networks;

Introduction

Modern enterprises run cloud applications that span multiple layers, including application logic, container runtimes, and cloud services, creating complex attack surfaces. Adversaries increasingly chain exploits across these layers, yet existing defenses remain fragmented. Cloud-Native Application Protection Platforms (CNAPPs) [22, 17] focus on configuration and posture management but offer little visibility into live application behavior. Host-based EDRs [6] flag isolated process anomalies without cloud context, and cloud-native tools [1, 2, 18] detect suspicious API activity without linking it to application or host-level causes. This siloed monitoring leaves security teams blind to the full narrative of multi-stage attacks and forces them to manually correlate disconnected alerts, often too late to prevent damage.

Cloud Application Detection and Response (CADR) offers a necessary shift by unifying observability across application, workload, and cloud layers. Existing cloud detection systems centralize logs for analysis but struggle with correlating events across stack boundaries. This limitation stems from a deeper architectural gap: *lack of runtime provenance*. CNAPPs lack execution context, EDRs ignore cloud identities, and cloud-native tools miss process lineage. Without end-to-end causal visibility, advanced threats remain undetected or poorly explained [8, 10].

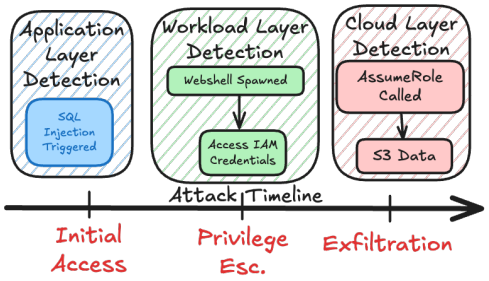

Figure 1: Timeline of the MOVEit attack highlighting the need for cross-layer visibility across application, workload, and cloud layers for complete detection and investigation.

The MOVEit breach [19], depicted in Figure 1, demonstrates the operational urgency for this shift. A SQL injection at the application layer triggered remote shell access in the workload layer, followed by unauthorized cloud data access through inherited IAM credentials. Each stage of the attack occurred in a different operational plane, but existing tools failed to observe or correlate the entire chain in real time. Even when alerts were raised, they lacked causality, context, and triage guidance. Some tools will flag isolated symptoms, such as anomalous file access or role assumption, but none supported cross-layer reasoning. Incidents like Log4Shell [7] and Spring4Shell [11] followed similar trajectories from initial injection to lateral host movement and eventual cloud compromise. Without unified telemetry and causally linked reasoning, security teams are left chasing fragments rather than coherent attack narratives.

To address these gaps, we introduce the novel notion of Provenance-based Cloud Application Detection and Response (P-CADR). P-CADR extends provenance principles to the runtime cloud stack by construct-

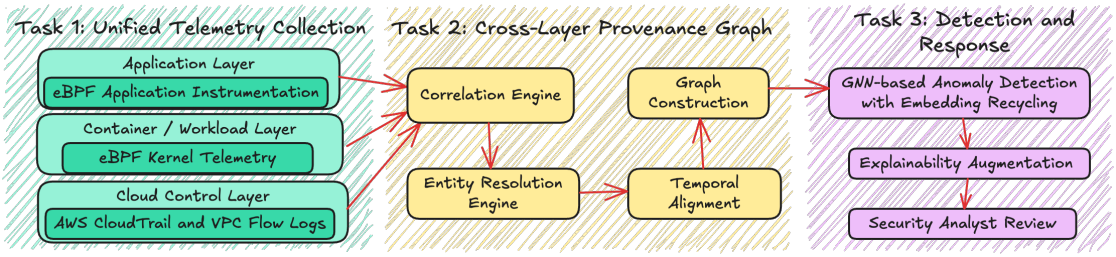

Figure 2: End-to-end workflow showing our three interconnected tasks.

ing *Cross-Layer Provenance Graphs (CPGs)*. These graphs unify semantically rich telemetry from the application, container, and cloud layers, enabling causal reasoning across traditionally disjoint components. This design provides full-stack observability and traceability, allowing security teams to trace attack causality end-to-end. Unlike traditional systems that detect isolated anomalies, P-CADR infers attack narratives by encoding *how*, *where*, and *why* an event occurred in its broader context. It empowers security teams with actionable, explainable incident graphs and enables autonomous response actions mapped to cloud-native APIs. We design algorithms and architectures to leverage CPGs for detecting complex threats and responding in real time, effectively augmenting the transparency of cloud systems and enabling security teams to see the full story of an intrusion, not just isolated fragments.

We propose a three-pronged approach. First, we will develop a cloud-native instrumentation layer using eBPF to collect semantically rich telemetry across all layers (Task 1). Second, we will construct unified Cross-Layer Provenance Graphs (CPGs) that encode causal interactions using entity linkage, temporal correlation, and semantic motifs (Task 2). Third, we will develop a provenance-graph-based detection and response engine capable of recognizing complex attack chains (Task 3).

Methods

Task 1. *(Cross-Layer Runtime Observability Using eBPF)* Modern attacks often traverse application, workload, and cloud control layers, yet existing security agents remain blind to these multi-layered threat paths. They suffer from poor coverage of interpreted runtimes, lack correlation with cloud APIs, and incur high overhead, especially in containerized, ephemeral environments. We will address these limitations by building a unified observability layer that captures and correlates security-relevant events across all layers.

Our architecture will leverage **eBPF as a programmable, low-overhead instrumentation substrate**, enabling dynamic tracing of both user-space and kernel-space activity. As shown in Figure 3, we will develop a workflow that identifies and prioritizes security-critical application functions using NLP over MITRE ATT&CK and documentation sources. We will then attach *uprobes* to the selected functions for tracing high-level application behavior (e.g., query evaluation, template rendering). In parallel, *kprobes* will monitor low-level system calls (e.g., `execve`, `connect`), and cloud API modules will collect telemetry from CloudTrail, STS, VPC Flow Logs, and Kubernetes audit trails. Each event will be enriched with container metadata (e.g., cgroup, pod, namespace), execution context (PID, UID, IAM identity), and correlation tags, then joined in-kernel using extended BPF maps to minimize overhead and preserve temporal causality.

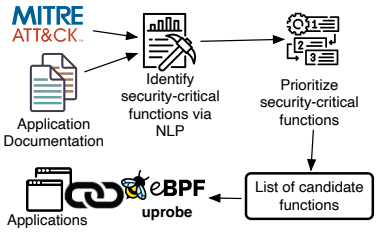

Figure 3: Workflow for identifying and instrumenting security-critical application functions using NLP-guided analysis and eBPF uprobes.

In the MOVEit breach scenario, our system would capture the SQL injection through an application-layer uprobe, trace the shell spawn via `execve`, monitor IAM escalation through STS logs, and record S3 exfiltration, all causally linked in a unified telemetry graph. We will deploy our instrumentation across AWS EKS clusters, simulate multi-stage attacks using Caldera [4] and Atomic Red Team [5], and evaluate trace completeness, data fidelity, and CPU/I/O overhead.

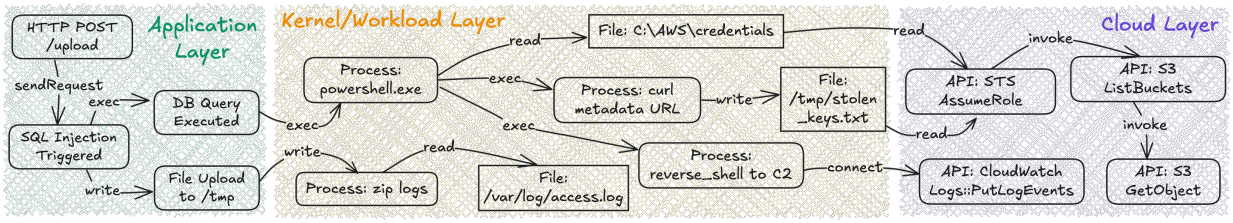

Figure 4: Example Cross-Layer Provenance Graph (CPG) for the MOVEit attack scenario.

Task 2. *(Cross-Layer Provenance Graph Construction (CPG))* While Task 1 will enable telemetry collection across application, kernel, and cloud layers, raw logs alone remain insufficient for attack attribution. The key challenge is correlation: identities diverge across layers (e.g., IAM roles vs containers), timestamps are often misaligned, and ephemeral processes or containers lack persistent identifiers. Existing provenance models cannot unify these fragmented signals, nor can they tie host-side activity to cloud-side consequences. Without precise correlation, defenders are left with disconnected clues and incomplete attack narratives.

We will address this by constructing real-time Cross-Layer Provenance Graphs (CPGs) that fuse heterogeneous telemetry into a single causal structure. To solve the correlation problem, we will use application-layer hooks to intercept AWS SDK calls, logging explicit links between processes and cloud API actions. For unhooked events, we will apply time-and-entity correlation, matching processes that initiate connections to cloud endpoints with subsequent CloudTrail events, augmented by cloud metadata such as instance IDs and request IDs. IAM roles will be represented as nodes, and privilege transitions via AssumeRole will form causal edges connecting roles, API calls, and originating processes. We will resolve identity mismatches using metadata fingerprinting, fuzzy role matching, and container ancestry tracking [9, 12, 16]. Temporal misalignment will be addressed with adaptive windowing [3] and entropy-based alignment [15].

Figure 4 illustrates how the MOVEit attack will be captured end-to-end, linking SQL injection, shell spawn, credential access, IAM role usage, and S3 exfiltration in a unified graph. This graph structure enables precise investigation and forms the basis for Task 3, where we will apply graph-based deep learning techniques to detect anomalous paths and unseen attack patterns. We will evaluate CPGs based on fidelity against expert-labeled traces, ingestion latency under burst load, and memory overhead in continuous deployment.

Task 3. *(Provenance-Guided Detection and Automated Response)* The CPGs constructed in Task 2 will serve as the foundation for this downstream threat detection task. Traditional tools lack the ability to trace fine-grained, multi-step attack chains and often miss stealthy tactics that span across system and cloud layers. They rely on static signatures or coarse anomaly metrics, producing noisy alerts with limited context. By leveraging rich, real-time CPGs, we can detect complex adversarial behaviors with greater precision, explainability, and coverage.

We will build a detection engine that processes streaming subgraphs from CPGs using a hybrid learning pipeline. We will extract candidate subgraphs using sliding windows and model them using supervised GNN classifiers, such as GraphSAGE [13] and GAT [21] to assign semantic labels. PI Hassan's prior system, Flash [20], successfully detected APTs on single-host provenance graphs; here, we will extend that methodology to multi-layer CPGs. To accelerate inference, we will incorporate embedding recycling, a technique introduced in Flash [20], to avoid recomputing node embeddings for stable regions of the graph. To improve analyst trust and enable human-in-the-loop validation, we will adapt GNNExplainer [23] and PGExplainer [14] to generate subgraph-level justifications that highlight the key entities and transitions responsible for triggering an alert. We will evaluate detection performance using traces from MOVEit, XZ Utils, Spring4Shell, and custom APT simulations. Metrics will include detection precision, false positive rate, response latency, and accuracy in triage tasks.

Expected results

We will release instrumentation code and cross-layer telemetry datasets by Month 3, complete the CPG construction engine by Month 6, deliver the detection and explanation modules with code and benchmarks by Month 9, and submit a peer-reviewed publication and present results by Month 12. Technical reports and documentation will be shared throughout.

Funds Needed

We request $60,000 for one year of graduate student support (stipend, tuition, health insurance) and $15,000 for one month of summer salary for PI Hassan. Additionally, we seek $25,000 in AWS Promotional Credits to support ML workloads and data storage. Our experiments on p3.2xlarge and m5.large instances will support ML training and baseline system operations. We also require 3 TB of secure Amazon S3 storage for Task 1 logs. The credits will cover these expenses.

References

[1] Amazon Web Services. *Amazon GuardDuty – Intelligent Threat Detection*. 2025. URL: https://aws.amazon.com/guardduty/.

[2] Amazon Web Services. *Detection and Response on AWS*. 2024. URL: https://aws.amazon.com/products/security/detection-and-response/.

[3] A. Bifet and R. Gavalda. "Learning from time-changing data with adaptive windowing". In: *Proceedings of the 2007 SIAM international conference on data mining*. SIAM. 2007, pp. 443–448.

[4] *CALDERA*. https://www.mitre.org/research/technology-transfer/open-source-software/caldera.

[5] R. Canary. *Atomic Red Team*. https://github.com/redcanaryco/atomic-red-team. Accessed: 2025-04-21. 2025.

[6] *CrowdStrike*. https://www.crowdstrike.com/.

[7] Cybersecurity and Infrastructure Security Agency. *Apache Log4j Vulnerability Guidance*. 2021. URL: https://www.cisa.gov/news-events/news/apache-log4j-vulnerability-guidance.

[8] W. U. Hassan, M. A. Noureddine, P. Datta, and A. Bates. "OmegaLog: High-fidelity attack investigation via transparent multi-layer log analysis". In: *NDSS*. 2020.

[9] Y. Huo, Y. Su, C. Lee, and M. R. Lyu. "Semparser: A semantic parser for log analytics". In: *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE. 2023, pp. 881–893.

[10] M. A. Inam, W. U. Hassan, A. Ahad, A. Bates, R. Tahir, T. Xu, and F. Zaffar. "Forensic Analysis of Configuration-based Attacks". In: *NDSS*. 2022.

[11] Intruder.io. *Spring4Shell [CVE-2022-22965]: All you need to know*. 2022. URL: https://www.intruder.io/blog/spring4shell-cve-2022-22965.

[12] V.-H. Le and H. Zhang. "Log parsing with prompt-based few-shot learning". In: *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE. 2023, pp. 2438–2449.

[13] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann. "E-graphsage: A graph neural network based intrusion detection system for iot". In: *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE. 2022.

[14] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang. "Parameterized explainer for graph neural network". In: *Advances in neural information processing systems* 33 (2020), pp. 19620–19631.

[15] A. Makanju, A. N. Zincir-Heywood, and E. E. Milios. "An evaluation of entropy based approaches to alert detection in high performance cluster logs". In: *2010 Seventh International Conference on the Quantitative Evaluation of Systems*. IEEE. 2010, pp. 69–78.

[16] A. Nawaz and H. Kazemian. "A fuzzy approach to identity resolution". In: *International Conference on Engineering Applications of Neural Networks*. Springer. 2021, pp. 307–318.

[17] Orca Security. *Cloud-Native Application Protection Platform (CNAPP)*. https://orca.security/platform/cnapp-cloud-security-platform/. Accessed: 2025-04-21. 2024.

[18] Palo Alto Networks. *What Is CDR (Cloud Detection and Response)?* 2023. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-cloud-detection-and-response-cdr.

[19] Progress Software Corporation. *MOVEit Transfer Critical Vulnerability (May 2023) (CVE-2023-34362)*. 2023. URL: https://community.progress.com/s/article/MOVEit-Transfer-Critical-Vulnerability-31May2023.

[20] M. U. Rehman, H. Ahmadi, and W. U. Hassan. "FLASH: A Comprehensive Approach to Intrusion Detection via Provenance Graph Representation Learning". In: *IEEE Symposium on Security and Privacy (S&P)*. 2024.

[21] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[22] Wiz. *What is Cloud Detection and Response (CDR)?* 2024. URL: https://www.wiz.io/academy/what-is-cloud-detection-and-response-cdr.

[23] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. "Gnnexplainer: Generating explanations for graph neural networks". In: *Advances in neural information processing systems* 32 (2019).

Wajih Ul Hassan

+1 217-904-5884 • hassan@virginia.edu • www.cs.virginia.edu/~hur7wv/
wajihulhassan • Lab Website: dartlab.org

Research Interests

System Security, Threat Detection, Forensic Investigation, and Data Provenance

Current Professional Appointment

The University of Virginia (UVA) **USA**
Tenure-Track Assistant Professor *August 2022 – Present*
Department of Computer Science and School of Data Science

Education

University of Illinois at Urbana-Champaign (UIUC), USA
Ph.D., Computer Science *2015 – 2021*
Advisor: Dr. Adam Bates.
Thesis: Investigating System Intrusions with Data Provenance Analytics

Lahore University of Management Sciences (LUMS), Pakistan
Bachelor of Science, Computer Science *2011 – 2015*

Selected Awards & Honors

- NSF CAREER Award 2024-2029
- Weaver Faculty Fellowship, UVA 2022-2025

Selected Publications

[1] Mati Ur Rehman, Hadi Ahmadi, and Wajih Ul Hassan. "FLASH: A Comprehensive Approach to Intrusion Detection via Provenance Graph Representation Learning". In: *IEEE Symposium on Security and Privacy (S&P)*. Acceptance Rate: 17.8%. 2024.

[2] Muhammad Adil Inam, Yinfang Chen, Akul Goyal, Jason Liu, Jaron Mink, Noor Michael, Sneha Gaur, Adam Bates, and Wajih Ul Hassan. "SoK: History is a Vast Early Warning System: Auditing the Provenance of System Intrusions". In: *IEEE Symposium on Security and Privacy (S&P)*. Acceptance Rate: 17.1%. 2023.

[3] Wajih Ul Hassan, Adam Bates, and Daniel Marino. "Tactical Provenance Analysis for Endpoint Detection and Response Systems". In: *IEEE Symposium on Security and Privacy (S&P)*. Acceptance Rate: 12.3%. 2020.

[4] Qi Wang, Wajih Ul Hassan, Ding Li, Kangkook Jee, Xiao Yu, Kexuan Zou, Junghwan Rhee, Zhengzhang Chen, Wei Cheng, Carl A Gunter, et al. "You Are What You Do: Hunting Stealthy Malware via Data Provenance Analysis." In: *Symposium on Network and Distributed System Security (NDSS)*. Acceptance Rate: 17.4%. 2020.

[5] Wajih Ul Hassan, Shengjian Guo, Ding Li, Zhengzhang Chen, Kangkook Jee, Zhichun Li, and Adam Bates. "NoDoze: Combatting threat alert fatigue with automated provenance triage". In: *Symposium on Network and Distributed System Security (NDSS)*. Acceptance Rate: 17%. 2019.

[6] Wajih Ul Hassan, Mark Lemay, Nuraini Aguse, Adam Bates, and Thomas Moyer. "Towards scalable cluster auditing through grammatical inference over provenance graphs". In: *Symposium on Network and Distributed System Security (NDSS)*. Acceptance Rate: 21.5%. 2018.

Previously Funded Project Summary

PI Hassan has not previously received cash funding or AWS Promotional Credits directly or indirectly from Amazon.