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

As cloud providers face increasingly sophisticated Advanced Persistent Threats (APTs), scalable Cloud Detection and Response (CDR) systems are essential. Centralized threat detection in cloud environments struggles to handle the vast volume of audit logs generated by distributed worker nodes, creating significant data transfer overhead, network congestion, and slower response times. Additionally, manual investigation of CDR alerts by Security Operations Centers (SOCs) is slow and prone to error, highlighting the need for scalable, automated solutions. This proposal introduces a novel framework that addresses these limitations through a two-pronged approach: decentralized threat detection and AI-assisted alert investigation. By processing threat insights locally on each node using federated learning and Graph Neural Networks, the proposed FEDDETECT system reduces data transfer overhead and enhances scalability. To further improve SOC efficiency, the proposal introduces QUERYAI, an AI-driven assistant that automates the generation of investigation queries for analyzing audit logs stored in Security Information and Event Management (SIEM) systems. Leveraging a pre-trained language model, QUERYAI translates SOC prompts into optimized queries, streamlining investigations and reducing the need for manual analysis. This combined approach aims to significantly enhance cloud security by enabling more efficient threat detection and response.

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

Introduction

Cloud Application Detection and Response (CADR) represents a necessary evolution in runtime cloud security. Traditional Cloud Detection and Response (CDR) systems, which aggregate logs at a centralized point for analysis, struggle to detect the full scope of modern cloud-native threats that span application, workload, and cloud control layers. Existing approaches either focus too narrowly on individual layers or fail to provide sufficient context for detection and investigation [**berthoty2025runtime**]. Real-world attacks such as MoveIT, Log4j, and Spring4Shell illustrate how attackers exploit application vulnerabilities, pivot through containers, and ultimately access cloud resources without raising coherent alerts across layers.

The MoveIT breach exemplifies the need for a fundamentally new defense paradigm. The attack initiated with a SQL injection at the application layer, which led to remote shell access in the workload layer, and culminated in unauthorized access to cloud data via inherited IAM credentials. Each phase of the attack occurred in a different plane of the cloud environment, but existing tools—CNAPPs, EDRs, and SIEMs—were unable to holistically observe or correlate the full attack chain in real time. Most importantly, even when alerts were raised, they lacked causality, context, and actionable triage support. As highlighted in Berthoty's analysis [**berthoty2025runtime**], many tools could detect isolated pieces of the attack—such as unusual file access or anomalous role assumption—but lacked the cross-layer reasoning needed to tell the full story.

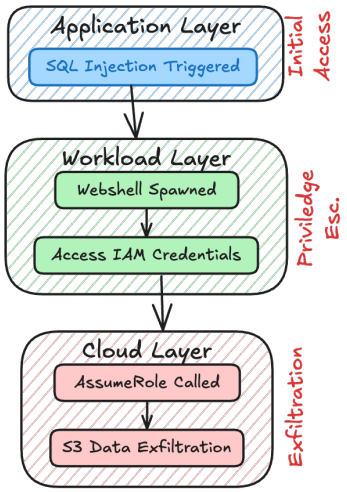

Figure 1: TODO

This fragmentation stems from a deeper architectural flaw in current cloud security tooling: siloed observability and lack of provenance. CNAPPs such as Wiz and Orca excel at static configuration analysis but offer no visibility into dynamic application behavior. Host EDRs operate in isolation from cloud identity and privilege systems, while cloud-native solutions like GuardDuty can observe cloud API calls but are blind to the processes or users that trigger them. As a result, modern attacks that exploit applications, pivot through workloads, and culminate in cloud data access are either detected too late—or not at all.

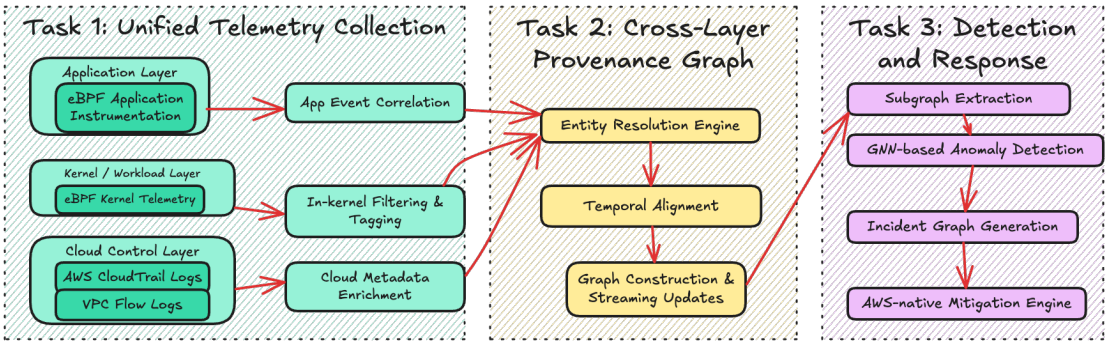

Figure 2: Workflow of our proposed P-CADR system.

To address these limitations, we introduce the novel concept of **Provenance-based Cloud Application Detection and Response (P-CADR)**. P-CADR brings provenance principles to runtime cloud security by constructing **Cross-Layer Provenance Graphs (CPGs)** that unify and correlate telemetry from the application, container, and cloud control planes. These graphs allow security teams to trace fine-grained causal chains—such as connecting a SQL injection to a webshell execution and subsequent exfiltration via S3 APIs. Unlike host-centric provenance systems or siloed anomaly detectors, P-CADR provides real-time, full-stack observability, enabling comprehensive threat attribution and contextual response.

What distinguishes P-CADR is its holistic and layered architecture. It continuously collects structured, causally linked telemetry from security-critical points in the application, kernel/runtime, and cloud control plane. Rather than treat each source as an isolated log stream, P-CADR fuses them into a *Unified Provenance Graph (UPG)*. This model captures every critical event—function calls, syscalls, IAM actions, data movements—and encodes how they influence each other across the cloud stack. The unified provenance enables new classes of detection, revealing stealthy multi-stage attacks that span multiple domains, and provides a solid foundation for automating incident triage and response.

P-CADR's novelty lies not just in linking layers, but in how it does so: using in-kernel log reduction for scalable telemetry, application-level instrumentation for security-aware observability, and advanced subgraph-based anomaly detection enhanced with embedding reuse for low-latency decisions. It is the first system to offer real-time detection and response that is fully provenance-informed, semantically rich, and cloud-native in design. P-CADR detects not only that an attack occurred, but how and why, offering security teams a full narrative rather than fragmented alerts.

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 and orchestrating contextual mitigations through cloud-native automation (Task 3).

Methods

Task 1. *(Cross-Layer Runtime Observability Using eBPF)*

Runtime visibility across application, container, and cloud control planes remains fragmented. While CNAPPs claim broad coverage, they largely function as glorified EDRs and fall short of understanding application-layer logic or cloud-native privilege flows. Tools like Wiz and Orca provide posture visibility but lack meaningful runtime detection. Most CNAPPs reduce detection to alerting on process starts or hash-based signatures, which cannot capture nuanced multi-step attacks like MoveIT.

We will build a distributed observability system that fuses telemetry from three layers, using **eBPF as the central collection mechanism**. eBPF enables efficient, in-kernel instrumentation with negligible

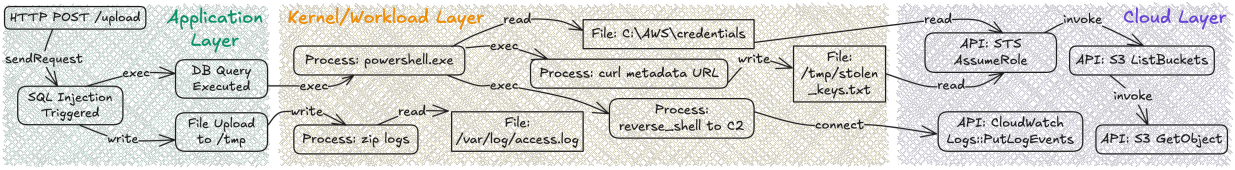

Figure 3: Example Cross-Layer Provenance Graph (CPG) for MoveIT vulnerability.

overhead and extensibility to both kernel and user-space events. Our system will attach *uprobes* to trace application-layer functions (e.g., SQL eval, template rendering, deserialization), *kprobes* to monitor syscall-level activities (e.g., execve, fork, open), and integrate cloud telemetry streams such as AWS CloudTrail, STS AssumeRole logs, and VPC Flow Logs to capture control plane activity.

Each event will be tagged with rich provenance metadata: PID namespace, cgroup lineage, IAM session, execution context, and time. We will use extended BPF maps to cache session identities, avoid redundant queries, and facilitate low-overhead joins across system and cloud events. In-kernel filtering will suppress benign noise and preserve only security-relevant causal links.

In the MoveIT case, our stack will capture the SQL injection using a uprobe on SQL query parsing, trace the shell spawn via execve, track the IAM role usage through STS calls, and link it to the S3 PutObject operation—reconstructing the entire causal chain across all layers. This is currently infeasible with CNAPPs or standalone host sensors.

We will deploy our instrumentation across AWS EKS clusters, simulate multi-stage attacks using Caldera and Atomic Red Team, and evaluate trace completeness, data fidelity, and CPU/I/O overhead. Baseline comparisons will include Falco [**falco**], Tetragon [**tetragon**], and commercial agents from Wiz and CrowdStrike.

Task 2. *(Cross-Layer Provenance Graph Construction (CPG))*

Cloud-native environments introduce significant challenges for provenance graph construction: identity mismatches across layers, asynchronous event timing, and transient execution environments. CloudTrail events may be delayed by minutes, while syscall traces are millisecond-level. Moreover, containers scale dynamically and inherit IAM roles without persistent identifiers.

We will build **Cross-Layer Provenance Graphs (CPGs)** to encode runtime causality across application, container, and cloud layers. Nodes will represent system-level entities such as SQL calls, processes, containers, IAM sessions, and cloud APIs. Edges will denote both direct causality (e.g., spawn, read, assume-role) and inferred links based on timing and shared execution contexts.

To unify identities, we will develop a metadata harmonization engine that fuses low-level runtime metadata (e.g., PID, cgroup, container ID) with cloud context (e.g., STS session name, role ARN, API caller identity). We will resolve temporal drift through a sliding-window alignment mechanism that reorders and merges asynchronous events using local clocks and density metrics. The result will be an incrementally constructed, high-fidelity causal graph updated in real time.

In the MoveIT scenario, the CPG will reveal a coherent path from an SQL call in a webserver, to a spawned shell, to a file read of credentials, to an AWS role assumption, and finally to a PutObject S3 API. These edges will not only reflect causal order but be annotated with semantic motifs to indicate patterns like RCE, privilege escalation, and lateral movement.

We will evaluate the CPG engine by measuring graph accuracy against hand-labeled traces, ingestion latency under burst load, and memory overhead in continuous deployment. We will compare against CamFlow [1], AWS Detective, and enterprise graph solutions, focusing on their inability to construct cross-layer runtime graphs with fine-grained causality.

Task 3. *(Provenance-Guided Detection and Automated Response)*

Even with unified graphs, detecting adversarial activity in the cloud remains difficult. Modern attacks are stealthy, span multiple layers, and often mimic benign behavior. Traditional rule-based systems like Sigma are brittle, and anomaly detection methods trained on single-layer logs generate noisy false positives.

Moreover, most tools stop at alerting and lack the ability to execute meaningful responses grounded in the context of an unfolding attack.

We will build a CPG-powered detection engine that applies graph learning techniques over streaming provenance subgraphs. We will extract candidate subgraphs using sliding windows and score them using contrastive learning and structural anomaly models. To distinguish adversarial motifs from benign behavior, we will use GNN architectures like GraphSAGE and GAT, trained on labeled traces and contrastive pairs to identify malicious execution patterns.

Our system will convert anomalous subgraphs into **incident graphs**, which encapsulate the full causal path from initial exploitation to impact. These incident graphs will be mapped to automated mitigations through a response module integrated with AWS APIs. For example, if a low-privilege pod spawns a shell and accesses an S3 bucket using a suspicious IAM role, the system will revoke the session, isolate the pod, and rotate credentials.

The novelty of our approach lies in transforming causal subgraphs into explainable and actionable response plans. Unlike rule-based detectors, our system can generalize to unseen attack patterns, adapt to evolving environments, and offer contextual justifications that SOC analysts can audit. The integration of GNNExplainer will support interpretable subgraph explanations, enabling human-in-the-loop validation.

We will evaluate detection precision, false positive rates, and response latency using labeled traces from MoveIT, XZ Utils, and Spring4Shell. Human-centered evaluations will assess the clarity, completeness, and utility of our incident graphs in live triage settings.

Funds Needed

We request cash funding to cover personnel costs, including $60,000 for one year's salary for a graduate student (covering 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 manage computational and storage demands using Amazon EC2 and Amazon S3. Our project involves running experiments on the p3.2xlarge instance for ML tasks at $3.06 per hour, 24 hours a day, 5 days a week, totaling $17,625.60 annually for EC2. We also require secure storage of 5 TB on Amazon S3 for storing and sharing system logs produced in Task 1, with operational costs totaling $2,580 annually. This $25,000 in AWS Promotional Credits will also accommodate potential fluctuations and unforeseen costs.

References

[1] T. Pasquier, X. Han, M. Goldstein, T. Moyer, D. Eyers, M. Seltzer, and J. Bacon. “Practical whole-system provenance capture”. In: *Symposium on Cloud Computing*. 2017.

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.