Host Behavioral Consistency Modeling for Causal and Interpretable Intrusion Detection

Boyuan Xu[1,2], Yun Li[1,2], Yiru Gong[1,2], Linxu Li[1,2], Chen Zhang[1,2], Tian Tian[1,2(✉)], Bo Jiang[1,2], and Zhigang Lu[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
{xuboyuan,tiantian}@iie.ac.cn

Abstract. Advanced persistent threats (APTs) evade host defenses by distributing malicious activity over long time spans and blending it into benign workloads. Recent work leverages provenance graphs to model system-wide causal dependencies for intrusion detection, but practical deployments remain hindered by extreme class imbalance, temporally dispersed evidence across entities, and the need for causal, interpretable alerts. We address these challenges with a host behavioral consistency framework on temporal provenance graphs. The core design is to model normal interaction consistency in context, then convert inconsistencies into causal evidence and aggregate it across dependent entities. This yields compact attack chains that remain interpretable while controlling false positives under extreme class imbalance. Across three DARPA Transparent Computing datasets, our method reduces false positives and improves detection at scale, with a 0.40 MCC gain on CADETS over the best baseline.

Keywords: Behavioral Consistency Modeling · Graph Neural Networks · Host-based intrusion detection

1 Introduction

Advanced Persistent Threats (APTs) are among the most sophisticated cyber threats, characterized by stealthy, prolonged, and targeted attacks designed to evade detection over extended periods [17, 10]. Unlike traditional attacks, APTs blend into the system's normal operations, often spanning months or years to achieve their objectives, making them exceptionally difficult to detect with conventional methods [3, 15]. The persistence and subtlety of APTs pose significant challenges, as these attacks evolve in phases, including initial compromise, lateral movement, and data exfiltration, all while avoiding detection [14, 7].

The detection of APTs can be framed as a modeling problem: identifying deviations from normal system behavior over a long period. While conventional

(✉) Corresponding author.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

approaches like signature-based detection or anomaly detection based on isolated events fall short, APTs require continuous monitoring of system activity with a focus on subtle, long-term behavior patterns [6, 20]. One promising approach is to model system activity as a provenance graph, a causal representation of system interactions, capturing both temporal and structural dependencies between processes, files, and network flows [3, 21, 13].

Recent research has explored statistical analysis, rule-based reasoning, and deep learning on provenance graphs to uncover attack footprints embedded in system execution [20, 15, 3]. Despite encouraging progress, several fundamental challenges remain unresolved for practical and trustworthy intrusion detection on real-world hosts [9, 25, 16]. ① **Extreme class imbalance causes excessive false alarms.** Malicious activities constitute only a tiny fraction of system events, heavily interleaved with diverse benign operations. Many existing approaches treat the problem as a binary classification task, where the class imbalance leads to high false positive rates, resulting in overwhelming false alarms that limit deployability [10, 21, 6]. ② **APT behaviors form long, temporal causal chains across heterogeneous entities.** Real intrusions unfold through multi-stage interactions among processes, files, and network endpoints, where critical evidence is temporally dispersed and linked by causal dependencies. Capturing such threats requires jointly modeling temporal dynamics and structural context, yet many existing approaches still struggle to preserve long-range dependencies in complex execution traces [1, 3, 9]. ③ **Detection results must be causal and interpretable to support investigation.** In practice, high anomaly scores alone are not actionable unless they can be attributed to coherent causal evidence, e.g., compact attack paths or subgraphs that explain *why* an alert is raised. Without such interpretability, analysts face substantial investigation overhead and limited confidence in automated alarms, especially under high false-positive rates [25, 19, 13].

We address these challenges with a host behavioral consistency modeling framework that scores system interactions in context and converts them into causal evidence. We represent execution as a temporal provenance graph and assign an anomaly score to each edge by testing whether the observed syscall matches learned interaction regularities between its endpoints. An attention-based spatio-temporal encoder summarizes each entity's recent behavior, and syscall type prediction serves as a proxy objective for edge consistency. At detection time, edge scores are aggregated and expanded via lightweight causal propagation to reconstruct compact attack chains, yielding actionable and interpretable alerts.

Our main contributions are as follows:

- We define a stricter notion of normality at the interaction level by using syscall type prediction to score edge consistency, producing calibrated anomaly evidence that suppresses false positives under extreme imbalance.
- We develop an attention-based spatio-temporal encoder that selectively fuses temporal context with dependency structure, capturing behavior patterns in long-range, multi-entity execution flows.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

– We propose a lightweight causal propagation method that aggregates edge evidence and reconstructs compact attack chains for interpretable analysis.
– We evaluate on three DARPA TC datasets, showing improved detection with fewer false positives and practical efficiency at scale, including a 0.40 MCC gain over the best baseline on CADETS.

The remainder of the paper is organized as follows: Section 2 presents preliminaries and related work. Section 3 details our proposed methodology. Section 4 reports experimental results, and Section 5 concludes the paper.

2 Preliminaries and Related Work

Syscall and Host Behavior Modeling Syscall is the operating system interface through which user-space programs access core resources such as computation, persistent storage, and communication [15, 7]. During program execution, syscalls are invoked at high frequency, producing a dense and time-ordered event stream that records concrete resource operations and interactions among system entities [20, 19]. This resource-centric view provides a unified abstraction of host behavior that is largely independent of application logic, enabling systematic modeling and analysis of runtime activities [3, 9].

Provenance Graph Representation To model system execution with explicit dependencies, prior work typically organizes audit data into a provenance graph [3, 21]. Nodes represent system entities such as processes, files, and communication endpoints, while directed edges encode dependencies induced by syscalls between entities.

Each edge is timestamped and annotated with semantic attributes (e.g., syscall type), yielding a temporal and heterogeneous graph that captures how interactions evolve over time [10]. This representation supports reasoning about multi-hop causal dependencies across entities [1, 25].

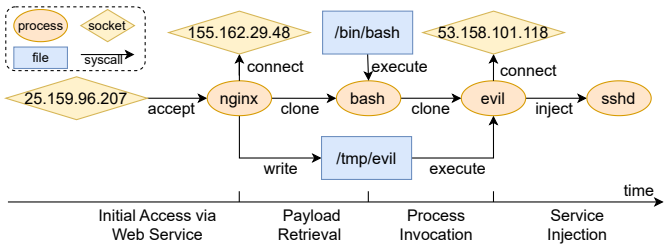

Fig. 1. A provenance graph illustrating a simplified multi-stage attack, where abnormal system behaviors emerge as structured interaction patterns among system entities.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

Figure 1 presents a simplified multi-stage attack and its corresponding provenance graph, illustrating how abnormal system behaviors manifest as structured interaction patterns within the graph.

Related Work Based on provenance graph representations, intrusion detection methods can be broadly grouped into rule-based, statistics-based, and learning-based approaches [15, 20, 9, 26].

Rule-based methods encode prior knowledge as handcrafted rules or templates and match audit records against known attack patterns [14, 5, 12, 6, 7]. While effective for recognizable behaviors, they require continuous maintenance and often generalize poorly to evolving or previously unseen APT tactics [8].

Statistics-based methods characterize entities and interactions using low-order graph statistics or lightweight heuristics, and flag deviations from typical profiles [20, 19, 23, 18]. These techniques are efficient and often interpretable, but their limited context modeling makes it difficult to capture multi-stage intents that unfold across entities and time [21].

Learning-based methods learn normal execution patterns from provenance graphs and detect departures via representation learning [3, 9, 4, 24]. Recent graph-based models explicitly leverage the heterogeneous structure of system entities and incorporate temporal context to better model dependencies among interactions [25, 16, 22, 1, 2]. Despite this progress, existing methods still struggle to control false alarms under extreme imbalance and to provide compact, causal explanations for practical triage, motivating further investigation [10, 13].

3 Methodology

This section introduces our framework for detecting abnormal host behavior using provenance graph analysis. As illustrated in Figure 2, system audit logs are first modeled as a provenance graph, from which semantic features are extracted and processed by a spatio-temporal encoder and a syscall type prediction decoder to derive anomaly scores, followed by propagation-based analysis to reconstruct attack-relevant subgraphs.

3.1 Graph Representation

Provenance Graph Construction We construct a directed temporal provenance graph over three entity types: processes (identified by name `comm` or command-line `cmdline`), files (by filesystem path), and network endpoints (by socket attributes). Syscall events are mapped to timestamped directed edges and serve as input to subsequent feature construction and spatio-temporal representation learning.

Feature Representation The provenance graph constructed in the previous stage is a heterogeneous graph, where nodes represent system entities and edges

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

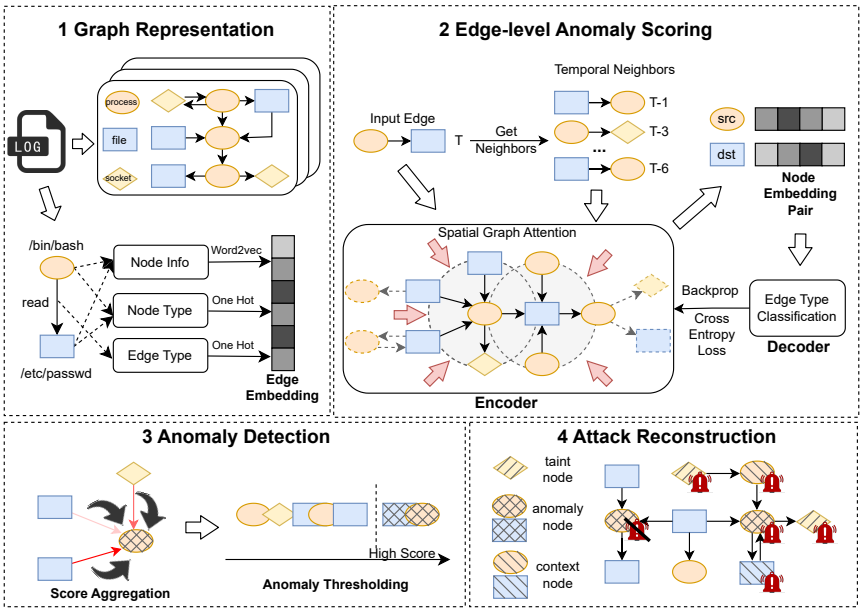

Fig. 2. Overview of the proposed framework. Semantic features are extracted from a provenance graph and processed to generate anomaly scores and reconstruct attack-relevant subgraphs.

correspond to syscall events between them. While temporal ordering and dependency structure are captured by the graph topology, additional semantic information required for behavior analysis resides in entity attributes and syscall metadata and must be explicitly represented.

In our design, semantic information is primarily associated with nodes. Each node is described by its entity type together with auxiliary attributes, which are essential for distinguishing entities of the same type. For example, file entities such as `/var/log/auth.log` and `/var/log/syslog` should be represented as distinct nodes because they often participate in different behaviors, while remaining similar due to shared directory hierarchy and semantics. To achieve this, string-valued attributes are transformed into dense vectors using tokenization and word embeddings with linearly decaying aggregation.

Edge features capture interaction semantics by combining the syscall type with the representations of the source and destination nodes, enabling downstream reasoning about interaction consistency. The constructed node and edge features (summarized in Table 1) serve as heterogeneous inputs for spatio-temporal representation learning, enabling the model to integrate structural, temporal, and semantic information.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

Table 1. Semantic information used for feature representation.

Semantic Category		Information Used	Examples
Node Info	Process	Execution context	`bash -i`
	File	File path	`/home/user/.bashrc`
	Network	Socket tuple	`192.168.1.10:22`
Node Type		Entity category	`process`, `file`, `socket`
Edge Type		Syscall type	`read`, `write`, `execve`

3.2 Edge-level Anomaly Scoring

Provenance graphs encode both structural dependencies among system entities and fine-grained temporal dynamics of system interactions. Effective anomaly detection therefore requires jointly modeling temporal locality and structural dependencies, particularly to capture long-range, multi-entity behaviors characteristic of stealthy attacks. To this end, we adopt a spatio-temporal graph modeling architecture that explicitly combines time-aware neighborhood sampling with attention-based graph convolution.

Spatio-temporal Encoder The above spatio-temporal modeling strategy is implemented using an attention-based graph encoder. For each system event arriving at node v at time t, the encoder aggregates information from the source node together with a temporally localized set of prior interactions to compute a context-aware representation of v.

To construct the temporal neighborhood of node v at time t, recent incoming interactions are sampled. Let $S_t(v)$ denote the set of all incoming edges to v whose timestamps precede t. From this set, a fixed-size subset consisting of the K interactions whose timestamps are closest to t from below is retained:

$$S_K(v,t) = \arg \min_{S \subseteq S_t(v),\, |S|=K} \sum_{(u,v,t_{uv}) \in S} |t_{uv} - t|, \tag{1}$$

where t_{uv} denotes the timestamp associated with edge (u, v). This time-aware neighbor sampling strategy enforces temporal locality, bounds the receptive field of message passing, and enables inductive generalization to previously unseen nodes.

On the sampled temporal neighborhood, we perform attention-based message passing to aggregate structural and semantic information. For each edge $(u, v) \in S_K(v, t)$, an attention coefficient is computed to quantify the importance of the interaction from node u to node v:

$$\alpha_{u,v} = \frac{\exp\left(\frac{(\mathbf{W}_Q \mathbf{x}_v)^\top (\mathbf{W}_K \mathbf{x}_u + \mathbf{W}_E \mathbf{e}_{uv})}{\sqrt{d}}\right)}{\sum_{u' \in S_K(v,t)} \exp\left(\frac{(\mathbf{W}_Q \mathbf{x}_v)^\top (\mathbf{W}_K \mathbf{x}_{u'} + \mathbf{W}_E \mathbf{e}_{u'v})}{\sqrt{d}}\right)}, \tag{2}$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

where $\mathbf{x}_u$ and $\mathbf{x}_v$ denote the input feature vectors of nodes u and v, respectively, $\mathbf{e}_{uv}$ denotes the feature vector of edge (u, v), $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_E$ are learnable projection matrices, and d is the dimensionality of the key vectors.

Using the attention coefficients, the representation of node v is updated by aggregating messages from its temporal neighbors:

$$\mathbf{h}_v = \sum_{(u,v)\in S_K(v,t)} \alpha_{u,v}\big(\mathbf{W}_V \mathbf{x}_u + \mathbf{W}_{E'}\mathbf{e}_{uv}\big), \tag{3}$$

where $\mathbf{W}_V$ and $\mathbf{W}_{E'}$ are learnable projection matrices. The resulting representation $\mathbf{h}_v$ captures recent temporal information and structural interaction semantics, and supports interaction-specific reasoning in downstream prediction tasks.

Syscall Type Prediction Decoder Given the spatio-temporal representations generated by the encoder, the decoder predicts the syscall type of each observed interaction to assess its consistency with learned normal behavior. Specifically, for a syscall event represented by an edge (u, v), let $\mathbf{h}_u$ and $\mathbf{h}_v$ denote the encoder outputs corresponding to the source and destination nodes. The decoder first projects the two endpoint representations and combines them to form an edge-level feature:

$$\mathbf{z}_{uv} = \mathrm{ReLU}(\mathbf{W}_{\mathrm{src}}\mathbf{h}_u \parallel \mathbf{W}_{\mathrm{dst}}\mathbf{h}_v)\,, \tag{4}$$

where $\parallel$ denotes vector concatenation, $\mathbf{W}_{\mathrm{src}}$ and $\mathbf{W}_{\mathrm{dst}}$ are learnable projection matrices, and $\mathrm{ReLU}(\cdot)$ denotes the rectified linear unit activation function.

The combined representation $\mathbf{z}_{uv}$ is then used to predict the syscall type through a linear classifier followed by a softmax function:

$$\hat{\mathbf{y}}_{uv} = \mathrm{softmax}(\mathbf{W}_o \mathbf{z}_{uv})\,, \tag{5}$$

where $\hat{\mathbf{y}}_{uv}$ represents the predicted probability distribution over syscall types and $\mathbf{W}_o$ is a learnable weight matrix. During training, the decoder parameters are optimized by minimizing the cross-entropy loss between $\hat{\mathbf{y}}_{uv}$ and the ground-truth syscall type labels collected from normal execution traces.

Prediction errors are used to derive edge-level anomaly scores, which are further processed by downstream reasoning and propagation modules.

3.3 Anomaly Detection

Based on the syscall type prediction results produced by the decoder, we perform anomaly detection by aggregating edge-level anomaly signals to the node level. Each system interaction is assigned an anomaly score derived from its prediction error, and a node's anomaly score is computed by aggregating the scores of its associated syscall interactions, indicating how strongly it participates in abnormal behavior. This yields a continuous anomaly score per node for detection and downstream reconstruction.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

To flag anomalous nodes, we adopt an automatic validation-based threshold. We reserve a subset of benign execution data as validation and exclude it from training. We then set the threshold to the maximum node anomaly score observed on this validation set. During inference, nodes whose anomaly scores exceed the threshold are flagged as suspicious.

This strategy removes manual threshold tuning and anchors decisions in observed benign behavior, helping reduce false positives while preserving sensitivity to significant deviations.

3.4 Attack Reconstruction

Learning-based anomaly detection can identify deviations from normal execution patterns, but severe imbalance between benign and malicious activities in provenance graphs often results in noisy alerts. In contrast to isolated anomalies, real attacks typically manifest as temporally coherent data or control flows driven by the attacker. Accordingly, we cast risk aggregation as an attack-flow reconstruction problem, which determines whether alerted nodes can be causally connected to attacker-controlled interaction points through time-consistent execution paths.

Taint Definition & Temporal Dependency Propagation. Our framework defines taint sources as entities that may receive external input, which could potentially be controlled by an attacker. In host-based intrusion scenarios, network sockets and user-interactive processes, such as shell processes, are treated as taint sources. These taint sources are not assumed to be inherently malicious but serve as points from which attacker-driven data or control may propagate.

Given a set of alerted nodes, we search both upstream and downstream to determine whether a temporally valid dependency path exists between an alert and any taint source. Nodes encountered along such paths are treated as attack context nodes and jointly form a reconstructed attack flow. Alerts that cannot be connected to taint sources through any such context are considered less likely to be attack-related and are therefore attenuated.

To realize this computation efficiently, we propagate temporal dependency sets over the provenance graph using a dynamic programming strategy. For each node v, we maintain a dependency set

$$S(v) = \{(n, t_n)\}, \tag{6}$$

where (n, t_n) indicates that entity n appears on a temporally valid dependency path reaching v with interaction time t_n. Dependency sets are propagated by scanning provenance edges in temporal order. For an edge (u, v, t), the dependency state of v is updated as

$$S(v) \leftarrow S(v) \cup S(u) \cup \{(u, t)\}. \tag{7}$$

This update is a dynamic programming recurrence: the state of v is incrementally constructed from previously computed states of its neighbors under the temporal ordering constraint.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

Naïvely propagating dependency sets may lead to unbounded growth in dense provenance graphs. To ensure scalability while retaining high-relevance causal context, we impose a time-aware capacity constraint. When $|S(v)|$ exceeds a predefined bound M, only the M dependency elements whose timestamps are closest to the current propagation time t are retained:

$$S(v) \leftarrow \arg \min_{S' \subseteq S(v),\, |S'|=M} \sum_{(n,t_n)\in S'} |t_n - t|. \tag{8}$$

Bidirectional Reconstruction & Alert Aggregation Effective reconstruction requires reasoning about both potential entry points and downstream impact. We therefore apply the same propagation rule under two temporal traversal orders: ascending timestamps to propagate dependencies forward and model attacker-driven impact, and descending timestamps to propagate dependencies backward to trace alerts toward possible attack entry points.

The reconstructed dependency sets provide a principled basis for risk aggregation and denoising. We first enforce a taint-support gate to suppress alerts that cannot be embedded into an attacker-driven execution flow. Let $a(v) = \mathbb{I}[r(v) \geq \theta]$ denote the initial alert indicator derived from anomaly score $r(v)$. We retain an alert if *either* its upstream or downstream dependency set intersects with the taint source set T:

$$\hat{a}(v) = a(v) \cdot \mathbb{I}\Big[\big(S_{\text{up}}(v) \cap T \neq \emptyset\big) \ \vee \ \big(S_{\text{down}}(v) \cap T \neq \emptyset\big)\Big]. \tag{9}$$

For each retained alert, we reconstruct its attack context by aggregating upstream and downstream dependencies. We define the reconstructed context closure as $C(v) = \{v\} \cup S_{\text{up}}(v) \cup S_{\text{down}}(v)$. A node u is included in the reconstructed attack flow if it appears in the context closure of any taint-supported alert:

$$\tilde{a}(u) = \mathbb{I}\Big[\exists v \in V : \ \hat{a}(v) = 1 \ \wedge \ u \in C(v)\Big]. \tag{10}$$

Finally, taint-supported alerts together with their context closures induce an attack-relevant subgraph, which forms a compact attack flow subgraph and provides structured, causal evidence for downstream analysis and investigation.

3.5 Complexity Analysis

Both training and inference scale linearly with the number of audit events N. The dominant costs arise from Word2Vec feature extraction and GNN training: the spatio-temporal encoder incurs $O(Kd^2)$ work per event, yielding $O(NKd^2)$ per training epoch. The attack reconstruction stage requires two linear passes over the provenance graph with per-edge set operations bounded by M, contributing $O(NM)$ at inference time. As K, d, and M are fixed hyperparameters, the end-to-end complexity is $O(N)$.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

4 Experimental Evaluation

4.1 Experimental Settings

Datasets We evaluate our approach on three datasets from the DARPA Transparent Computing (TC) Engagement 3: CADETS, TRACE, and THEIA[11], which capture realistic, multi-stage attack behaviors under different system environments and workloads.

CADETS is collected from FreeBSD-based server environments and primarily reflects backend-oriented activities, such as attacks targeting web and email services. In contrast, TRACE focuses on user-facing scenarios, including browser-based backdoors and phishing-driven compromises, and thus exhibits more interactive and noisy system behaviors. THEIA contains attack patterns similar to TRACE but is collected under a different operational setting, providing complementary system dynamics for evaluating robustness across environments.

We select these datasets as they are publicly available, widely used, and collectively cover diverse system behaviors and attack scenarios at realistic scale. Table 2 summarizes the key statistics of the evaluated datasets, including the number of nodes and edges in the provenance graphs and the diversity of observed system call types. Since all attacks in these datasets originate from external network sources, we conservatively treat all network nodes as taint sources, assuming they represent potential entry points for attacker-controlled interactions.

Table 2. Summary statistics of evaluation datasets.

Dataset	Nodes		Edges					
	#total	#attack	#total	#clone	#execute	#open	#write	#connect
CADETS	2,682,632	64	36,484,667	223,311	210,914	3,843,324	1,602,681	128,594
THEIA	1,258,362	141	44,366,117	238,489	116,621	6,606,629	3,678,040	591,943
TRACE	2,769,336	49	6,352,285	12,605	8,894	678,636	314,794	96,896

Baseline We compare our approach with three representative provenance-based intrusion detection systems that adopt different detection paradigms. Threatrace[21] is a rule-guided analysis system that detects intrusions by propagating suspicious dependencies from seed entities across execution graphs, focusing on causal expansion and attack investigation rather than statistical anomaly scoring. Flash[19] is an efficiency-oriented method that performs node-level detection by reusing learned node embeddings to reduce inference overhead, and reports results on DARPA TC datasets[11] using lightweight classification with dataset-specific thresholds. MAGIC[9] is a graph autoencoder–based approach that identifies anomalies via reconstruction-based outlier scoring on provenance graphs; it is trained exclusively on benign execution data and produces node-level anomaly scores.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

Evaluation Metrics Provenance graphs derived from system audit logs exhibit extreme class imbalance, where malicious entities constitute only a very small fraction of all nodes. Under such conditions, commonly used metrics such as accuracy or F1-score can be misleading, as they may be dominated by the majority benign class.

To address this issue, we adopt the Matthews Correlation Coefficient (MCC) as our primary evaluation metric. MCC takes into account true positives, false positives, true negatives, and false negatives, and provides a balanced measure even when class distributions are highly skewed. As a result, MCC offers a more reliable assessment of detection performance in realistic intrusion detection settings.

For completeness, we also report precision and recall to illustrate the trade-off between false alarms and missed detections, but focus our analysis on MCC when comparing overall detection effectiveness across methods.

Table 3. Main results on three DARPA TC datasets. Our method consistently achieves the best overall detection quality, producing accurate and actionable alerts under extreme class imbalance by controlling false positives while preserving attack coverage.

Dataset	Method	TP	FP	TN	FN	Precision	MCC
CADETS	Threatrace	57	252k	16k	7	0.00	0.00
	Flash	15	2.4k	266k	49	0.00	0.04
	MAGIC	49	80k	188k	15	0.00	0.02
	Ours	34	60	268k	30	0.36	0.44
THEIA	Threatrace	80	670k	29k	61	0.00	-0.03
	Flash	26	22k	677k	115	0.00	0.01
	MAGIC	98	352k	347k	43	0.00	0.01
	Ours	41	101	699k	100	0.29	0.29
TRACE	Threatrace	39	258k	41k	10	0.00	0.00
	Flash	5	10k	289k	44	0.00	0.00
	MAGIC	12	199k	100k	37	0.00	-0.01
	Ours	20	24	299k	29	0.45	0.43

4.2 Detection Effectiveness

Main Results Across the three DARPA TC datasets[11], our approach consistently achieves the strongest overall detection quality, especially under extreme class imbalance (Table 3). While ThreatRace[21], Flash[19], and MAGIC[9] can recover a subset of true attacks, they also generate an overwhelming number of false positives (e.g., tens to hundreds of thousands of FP), which drives precision close to zero and yields near-zero or even negative MCC. In contrast, our method maintains a low false-alarm rate while still detecting a meaningful portion of attack activity, resulting in substantially higher MCC.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

Concretely, on CADETS and TRACE we reach MCC values of 0.44 and 0.43 with precisions of 0.36 and 0.45, respectively, indicating balanced performance rather than trading recall for noise. On THEIA, despite a more challenging background volume, we still obtain the best MCC (0.29) with only 101 false positives, demonstrating that the combination of edge-level anomaly scoring and causality-constrained reconstruction produces alerts that are both accurate and actionable.

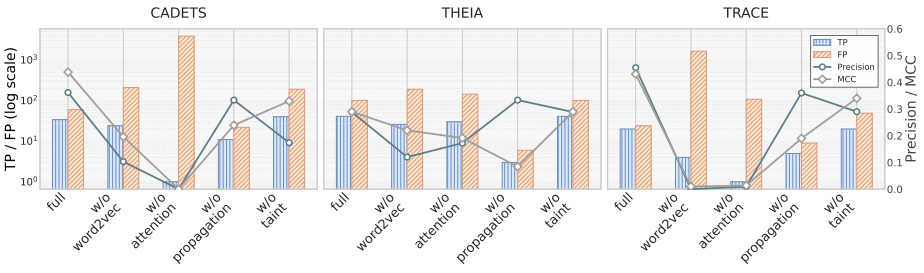

Fig. 3. Ablation results on three datasets. Removing word2vec and attention causes the largest drop in MCC, showing their importance for semantic encoding and context modeling. Propagation and taint filtering contribute to precision and false positive reduction.

Ablation Study We conduct an ablation study to quantify the contribution of major components. For each dataset, we evaluate the full pipeline against four ablated variants: w/o_word2vec uses randomly initialized embeddings instead of pretrained Word2Vec; w/o_attention replaces attention with mean pooling; w/o_propagation omits causal expansion and outputs only the initial seed alerts; and w/o_taint removes taint-based filtering in the propagation stage.

Across datasets, the strongest degradations arise from weakening semantic and contextual modeling, indicating that reliable interaction characterization relies on both informative attribute encodings and selective aggregation. Disabling causal expansion mainly reduces alert completeness by failing to recover dispersed evidence along multi-step attack chains. In contrast, removing taint-based filtering prevents the system from suppressing alerts that lack any temporally valid dependency path to taint sources, substantially increasing false positives and reducing alert actionability. Figure 3 summarizes the results.

Case Study and Visualization To illustrate the interpretability of our approach, we use our method to extract a representative attack subgraph from the CADETS dataset. As shown in Figure 4, several malicious processes (e.g., XIM, test) are flagged with high anomaly scores due to inconsistent *execute* behaviors that deviate sharply from benign execution patterns.

Through propagation, our method reconstructs the full attack chain, which includes exploiting an nginx vulnerability to download and execute a payload,

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

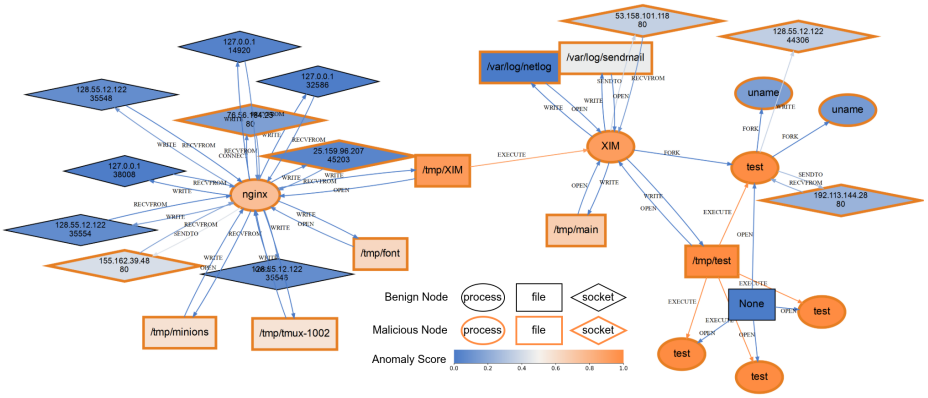

Fig. 4. Visualization of a detected attack subgraph from CADETS. Malicious process executions (e.g., XIM, test) are assigned high anomaly scores and connected via propagation to reveal the full attack path from exploitation to data exfiltration.

followed by information extraction and outbound communication to the attacker's server. This example highlights how behavioral consistency modeling and causal reconstruction together support precise and interpretable detection.

5 Conclusion

We presented a provenance-based intrusion detection framework that models host behavioral consistency at the interaction level. By combining spatio-temporal encoding with syscall type prediction, our method captures fine-grained semantic patterns and assigns calibrated anomaly scores to system interactions. A lightweight propagation mechanism reconstructs causal attack chains, providing interpretable evidence under complex and imbalanced system conditions.

Extensive experiments on DARPA TC datasets demonstrate that our approach achieves higher detection accuracy and significantly lower false positive rates compared to prior methods. Ablation and case studies further confirm the importance of each component in enabling precise and explainable detection. Future directions include generalizing the framework to multi-host environments and integrating human-in-the-loop feedback for adaptive detection.

Acknowledgments. This research is supported by National Key Research and Development Program of China (No.2023YFC2206402), the Strategic Priority Research Program of the Chinese Academy of Sciences (No.XDA0460100), the Innovation Program of the Institute of Information Engineering, CAS (E3Z0101116), the Program of Key Laboratory of Network Assessment Technology, the Chinese Academy of Sciences, Program of Beijing Key Laboratory of Network Security and Protection Technology.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

References

1. Zijun Cheng, Qiujian Lv, Jinyuan Liang, Yan Wang, Debin Sun, Thomas Pasquier, Xueyuan Han, and Jianfeng Ma. Kairos: Practical Intrusion Detection and Investigation Using Whole-System Provenance. In *Proc. IEEE Symposium on Security and Privacy (SP)*, pages 3533–3551, 2024.
2. Akul Goyal, Gang Wang, and Adam Bates. R-caid: Embedding root cause analysis within provenance-based intrusion detection. In *45th IEEE Symposium on Security and Privacy (S&P)*, pages 3515–3532, 2024.
3. Xueyuan Han, Thomas Pasquier, Adam Bates, James Mickens, and Margo Seltzer. Unicorn: Runtime provenance-based detector for advanced persistent threats. In *Proc. Network and Distributed System Security Symposium (NDSS)*, 2020.
4. Xueyuan Han, Xiao Yu, Thomas Pasquier, Ding Li, Junghwan Rhee, James Mickens, Margo Seltzer, and Haifeng Chen. SIGL: Securing software installations through deep graph learning. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2345–2362, 2021.
5. Wajih Ul Hassan, Adam Bates, and Daniel Marino. Tactical Provenance Analysis for Endpoint Detection and Response Systems. In *Proc. IEEE Symposium on Security and Privacy (SP)*, pages 1172–1189, 2020.
6. 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 *Proc. Network and Distributed System Security Symposium (NDSS)*, 2019.
7. Md Nahid Hossain, Sadegh M. Milajerdi, Junao Wang, Birhanu Eshete, Rigel Gjomemo, R. Sekar, Scott D. Stoller, and V. N. Venkatakrishnan. SLEUTH: Real-Time Attack Scenario Reconstruction from COTS Audit Data. In *Proc. 26th USENIX Security Symposium (USENIX Security)*, pages 487–504, 2017.
8. Md Nahid Hossain, Sanaz Sheikhi, and R. Sekar. Combating Dependence Explosion in Forensic Analysis Using Alternative Tag Propagation Semantics. In *Proc. IEEE Symposium on Security and Privacy (SP)*, 2020.
9. Zian Jia, Yun Xiong, Yuhong Nan, Yao Zhang, Jinjing Zhao, and Mi Wen. MAGIC: Detecting Advanced Persistent Threats via Masked Graph Representation Learning. In *Proc. 33rd USENIX Security Symposium (USENIX Security)*, pages 5197–5214, 2024.
10. Baoxiang Jiang, Tristan Bilot, Nour El Madhoun, Khaldoun Al Agha, Anis Zouaoui, Shahrear Iqbal, Xueyuan Han, and Thomas Pasquier. ORTHRUS: Achieving High Quality of Attribution in Provenance-Based Intrusion Detection Systems. In *Proc. 34th USENIX Security Symposium (USENIX Security)*, pages 7173–7192, 2025.
11. Angelos D. Keromytis. Transparent computing engagement 3 data release. https://github.com/darpa-i2o/Transparent-Computing/blob/master/README-E3.md, 2018. DARPA/I2O. Original release: August 30, 2018. Accessed: 2026-02-09.
12. Yonghwi Kwon, Fei Wang, Weihang Wang, Kyu Hyung Lee, Wen-Chuan Lee, Shiqing Ma, Xiangyu Zhang, Dongyan Xu, Somesh Jha, Gabriela Ciocarlie, Ashish Gehani, and Vinod Yegneswaran. MCI: Modeling-based causality inference in audit logging for attack investigation. In *Network and Distributed System Security Symposium (NDSS)*, 2018.
13. Hongmei Li, Tiantian Zhu, Jie Ying, Tieming Chen, Mingqi Lv, Chunlin Xiong, and Lili Shi. MIRDETECTOR: Applying Malicious Intent Representation for Enhanced APT Anomaly Detection. *Computers & Security*, 157:104588, 2025.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13

14. Sadegh M. Milajerdi, Birhanu Eshete, Rigel Gjomemo, and V. N. Venkatakrishnan. POIROT: Aligning Attack Behavior with Kernel Audit Records for Cyber Threat Hunting. In *Proc. ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1795–1812, 2019.
15. Sadegh M. Milajerdi, Rigel Gjomemo, Birhanu Eshete, R. Sekar, and V. N. Venkatakrishnan. HOLMES: Real-Time APT Detection through Correlation of Suspicious Information Flows. In *Proc. IEEE Symposium on Security and Privacy (SP)*, pages 1137–1152, 2019.
16. Wei Qiao, Weiheng Wu, Song Liu, Yebo Feng, Zehui Wang, Junrong Liu, Teng Li, Bo Jiang, Zhigang Lu, and Baoxu Liu. SAURONEYES: Disentangling Voluminous Logs to Unveil Camouflaged Attack Intentions. *IEEE Transactions on Information Forensics and Security*, 20(11):11744–11758, 2025.
17. Ankit Sharma, Brij B. Gupta, Ankit Kumar Singh, and V. K. Saraswat. Advanced persistent threats (APT): Evolution, anatomy, attribution and countermeasures. *Journal of Ambient Intelligence and Humanized Computing*, 14(7):9355–9381, 2023.
18. Xiaoyan Sun, Jun Dai, Anoop Singhal, Peng Liu, and John Yen. Using bayesian networks for probabilistic identification of zero-day attack paths. *IEEE Transactions on Information Forensics and Security*, 13(10):2506–2521, 2018.
19. Mati Ur Rehman, Hadi Ahmadi, and Wajih Ul Hassan. FLASH: A Comprehensive Approach to Intrusion Detection via Provenance Graph Representation Learning. In *Proc. IEEE Symposium on Security and Privacy (SP)*, pages 3552–3570, 2024.
20. Qi Wang, Wajih Ul Hassan, Ding Li, Kangkook Jee, Xinyu Yu, Kexin Zou, Junghwan Rhee, Zhi Chen, Wei Cheng, and Carl A. Gunter. You Are What You Do: Hunting Stealthy Malware via Data Provenance Analysis. In *Proc. Network and Distributed System Security Symposium (NDSS)*, 2020.
21. Su Wang, Zhiliang Wang, Tao Zhou, Hongbin Sun, Xia Yin, Dongqi Han, Han Zhang, Xingang Shi, and Jiahai Yang. THREATRACE: Detecting and Tracing Host-Based Threats in Node Level Through Provenance Graph Learning. *IEEE Transactions on Information Forensics and Security*, 17:3972–3987, 2022.
22. Weiheng Wu, Wei Qiao, Wenhao Yan, Bo Jiang, Yuling Liu, Baoxu Liu, Zhigang Lu, and Junrong Liu. Brewing Vodka: Distilling Pure Knowledge for Lightweight Threat Detection in Audit Logs. In *Proc. ACM Web Conference 2025 (WWW '25)*, 2025.
23. Yulai Xie, Yafeng Wu, Dan Feng, and Darrell D. E. Long. P-gaussian: Provenance-based gaussian distribution for detecting intrusion behavior variants using high efficient and real time memory databases. *IEEE Transactions on Dependable and Secure Computing*, 18(6):2660–2674, 2021.
24. Fan Yang, Jiacen Xu, Chunlin Xiong, Zhou Li, and Kehuan Zhang. PROGRAPHER: An anomaly detection system based on provenance graph embedding. In *32nd USENIX Security Symposium (USENIX Security 2023)*, 2023.
25. Jun Zeng, Xiang Wang, Jiahao Liu, Yinfang Chen, Zhenkai Liang, Tat-Seng Chua, and Zheng Leong Chua. ShadeWatcher: Recommendation-Guided Cyber Threat Analysis Using System Audit Records. In *Proc. IEEE Symposium on Security and Privacy (SP)*, pages 489–506, 2022.
26. Michael Zipperle, Florian Gottwalt, Elizabeth Chang, and Tharam Dillon. Provenance-based intrusion detection systems: A survey. *ACM Computing Surveys*, 55(7):1–36, 2022.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_13