CHAOS: Robust Spatio-Temporal Fusion for Generalizable APT Provenance Tracing

Teng Li[1,2,6], Wei Qiao[1,2(✉)], Yebo Feng[3], Jiahua Xu[4,5], Paolo Tasca[4,5], Weiguo Lin[1], Zexu Dang[1], Zhuo Ma[1], and Jianfeng Ma[1]

[1] School of Cyber Engineering, Xidian University, Xi'an 710126, China
qiaoweixidian@gmail.com
[2] State Key Laboratory of Integrated Services Networks (ISN), Xi'an, China
[3] Nanyang Technological University, Singapore, Singapore
[4] University College London, London, UK
[5] Exponential Science, London, UK
[6] Songshan Laboratory, Zhengzhou 450018, China

Abstract. As Advanced Persistent Threat (APT) attacks continue to evolve, critical information infrastructure faces increasingly severe security risks. Log-based provenance detection systems have become an effective defense mechanism. However, despite addressing many issues, existing solutions still exhibit several notable shortcomings: 1) they require attack information and expert knowledge, 2) they struggle to manage large-scale provenance graphs composed of massive data, 3) they insufficiently capture long-distance contextual dependencies within provenance graphs, and 4) they cannot adapt to benign changes in system behavior. To address these challenges, we propose CHAOS, an intrusion detection system (IDS) that balances behavioral structures and temporal dependencies. Specifically, CHAOS employs a soft segmentation strategy for large-scale provenance graphs, effectively extracting sub-events while preserving event structures. It then utilizes Graph Neural Networks (GNNs) to learn system behavior structures and combines Long Short-Term Memory (LSTM) networks to capture temporal dependencies between entities. Finally, it implements anomaly detection using a lightweight classifier and further suppresses false positives through the Elastic Weight Consolidation (EWC) mechanism. Evaluated across multiple real-world datasets, CHAOS demonstrates exceptional detection accuracy, significantly reduces false alarms, and exhibits a degree of resistance against adversarial attacks.

Keywords: Advanced persistent threat · Provenance detection · Host log · Intrusion detection system

1 Introduction

An APT (Advanced Persistent Threat) attack is a covert, long-term cyber operation conducted by sophisticated adversaries to infiltrate and compromise specific targets. Unlike traditional attacks, APT attacks often leverage zero-day exploits and they exhibit extended attack chains and long cycles, challenging detection systems. Existing methods like ProGrapher [34] use graph embeddings but fail to capture long-range dependencies, while StreamSpot [21] relies on clustering-based detection but suffers from false alarms due to coarse log segmentation. Therefore, Intrusion Detection Systems (IDS) must reduce dependence on prior knowledge, maintain low false positive rates, and adapt to evolving environments.

Existing methods [4,6,13,17,29,32,34,37] exhibit significant limitations. **Rule-based approaches** (e.g., RapSheet [9], Poirot [22]) heavily depend on predefined threat signatures and expert knowledge, making them ineffective against novel attacks. **Learning-based approaches** (e.g., ATLAS [3], Threatrace [29]) require labeled attack data for training, which restricts their generalizability. Solutions based on Graph Neural Network (GNN) learning often use shallow networks (e.g., GraphSAGE [7]) that miss long-term dependencies, while deeper GNNs face overfitting and high costs. Furthermore, existing provenance tracing schemes often employ coarse-grained log segmentation strategies. It compromises the structural integrity of recorded behaviors, resulting in an increased rate of false positives.

Existing provenance-based IDS face four critical challenges: (1) Attack agnosticism – Rule-based systems (e.g., RapSheet) require predefined threat signatures while learning-based methods (e.g., Threatrace) depend on labeled attack data, restricting generalization to novel APT attacks; (2) Scalability barriers – Coarse-grained log segmentation (e.g., time-window batching) disrupts the structural integrity of provenance graphs, inducing false positives; (3) Long-term dependency gap – Shallow GNNs fail to capture distant spatio-temporal dependencies, while deeper architectures suffer from overfitting and high overhead; (4) Concept drift vulnerability – Models (e.g., NoDoze [10]) trained on historical benign behaviors trigger false alarms when system behaviors undergo legitimate evolution.

To bridge these gaps, we propose Chaos, a robust spatio-temporal fusion framework for generalizable APT provenance tracing. Unlike existing methods, Chaos achieves high-precision detection by comprehensively modeling multi-scale behavioral dependencies—including structural topology, long-term temporal evolution, and cross-entity contextual relationships—within provenance graphs through joint GNN-LSTM learning. By integrating fuzzy clustering for structure-preserving log segmentation and Elastic Weight Consolidation (EWC) [20] for adaptive model updating, Chaos autonomously adapts to new benign behaviors and evolving attack strategies without relying on pre-defined threat signatures or labeled attack data, enhancing its operational resilience.

Chaos integrates four synergistic modules to enable attack-agnostic APT tracing and generate actionable defensive insights: (1) The Provenance Refinement Module processes raw logs into behavioral graphs and applies fuzzy clustering to extract coherent sub-events while preserving dependencies. (2) The Spatio-Temporal Fusion Module jointly learns behavioral semantics through dual-channel encoding: GNNs extract topological relationships by reinforcing intra-cluster node interactions, while Long Short-Term Memory (LSTM) networks model chronological dependencies across entity sequences. (3) The Anomaly Inference Module leverages a LightGBM [15] classifier trained on benign behavior embeddings to detect subtle deviations. (4) The Adaptive Consolidation Module dynamically mitigates concept drift via Elastic Weight Consolidation (EWC), strategically balancing historical knowledge retention with new behavior adaptation during operational updates.

We evaluated Chaos on the widely-adopted DARPA TC dataset [1] and StreamSpot dataset [2] for APT detection. Experimental results show that, compared to state-of-the-art (SOTA) fine-grained detection systems, Chaos achieves higher accuracy and a significantly reduced false positive rate. Moreover, it successfully adapts to system behavior drift, proving effective in handling complex real-world environments.

In summary, this paper makes the following contributions.

- We propose Chaos, a novel provenance tracing framework that integrates spatio-temporal dependency modeling and adaptive learning. It eliminates reliance on attack signatures or labeled malicious data, enabling robust detection of unknown APT threats in dynamic environments.
- We design a fuzzy clustering strategy to decompose massive audit logs into semantically coherent *sub-events*. This preserves behavioral dependencies disrupted by traditional coarse-grained segmentation (e.g., time-window batching), significantly reducing false positives .
- We develop a joint *spatio-temporal graph representation* module that captures topological structures and long-term event sequences, incorporating *EWC* to dynamically adapt to system behavior evolution.
- We evaluate Chaos on real-world datasets. Results demonstrate >20% false positive reduction, superior accuracy over SOTA methods, and resilience to system behavior drift.

2 Motivation

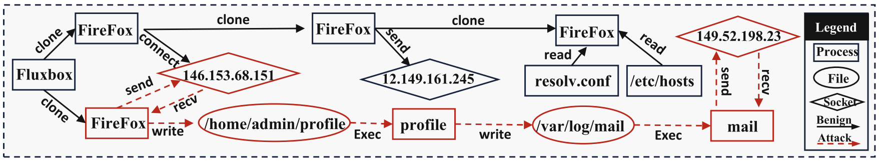


Fig. 1. An example of an attack provenance graph from the DARPA TC dataset [1]. The attack exploited a backdoor vulnerability via filefox. Chaos effectively identifies all critical attack components, as indicated by the red nodes in the figure. (Color figure online)

In this section, we introduce real attack scenarios from the DARPA TC dataset [1] to highlight the limitations of existing provenance-based IDS systems. Additionally, we demonstrate the effectiveness and practicality of Chaos.

2.1 Attack Scenario

Figure 1 illustrates a Firefox backdoor attack. The red subgraph represents a simplified version of the original attack. The attack begins when a victim computer running the vulnerable Firefox 54.0.1 unknowingly accesses a malicious advertisement server located at 146.153.68.151. This server exploits a backdoor in Firefox, injecting the binary executable 'Drakon' into the process memory. 'Drakon' subsequently spawns a new process with root privileges (/home/admin/clean), which connects to an attacker's server at 161.116.88.72, thus granting the attacker full access to the victim's computer.

2.2 Limitations of Existing Provenance IDSes

Provenance-based anomaly detection identifies potential malicious behaviors from provenance graphs, leveraging representations of entities and the flow of information. It can be divided into two categories: Rule-based PIDS and Learning-based PIDS. (1) Rule-based PIDSes: These systems capture network threats by combining audit logs with expert knowledge. RapSheet [9] utilizes predefined TTPs (Tactics, Techniques, and Procedures) to detect kill chains. Poirot [22] extracts attack graphs from threat intelligence and performs graph alignment detection on logs. CAPTAIN [28] adjusts its rules based on normal data using general label propagation rules to minimize false positives. (2) Learning-based PIDSes: These systems build models by learning from historical knowledge. Prographer [34] uses Graph2Vec embeddings to identify graph-level anomalies. Systems like StreamSpot [2] and Unicorn [8] identify anomalies through clustering of graph structures. Threatrace [29] and FLASH [26] use GNNs for node-level anomaly detection. ATLAS [3] performs supervised training by concatenating benign and malignant nodes into sequences. ShadeWatcher [37] utilizes recommendation algorithms to describe preferences in interactive behaviors. Although these solutions have achieved commendable results, they still have limitations in the face of highly complex attack environments.

Attack Agnosticity. Rule-based detection systems can maintain low false-positive rates, but formulating and constructing effective security policies can be challenging. Some deep learning-based APT detection methods use benign data as well as attack-containing datasets to train detection models. Although it can achieve perfect classification on test sets, real-world APT attack patterns are high variability and continuous evolution. Consequently, these approaches that rely on prior knowledge struggle to defend against unknown attacks.

Massive Logs. When facing massive logs, several provenance detection schemes directly split the provenance graph using time windows or batch cuts, which disrupts the internal graph structure and behavioral dependencies. When key connections are severed during log batch segmentation, it results in unlearned graph structures, leading to additional false positives. This method not only compromises the integrity of the detection process but also affects the accuracy and reliability of the system in identifying genuine threats.

Long-Term Dependency. Existing node-level provenance IDSes ignore the long-term dependencies of event nodes from a temporal perspective, focusing solely on graph structural information. However, shallow GNNs are incapable of capturing long-distance dependencies, and deeper GNNs suffer from high overhead and overfitting. Although FLASH considers temporal sequencing over time, it still lacks deep learning of its dependencies.

Concept Drift. Recent provenance schemes that model and train based on benign logs, by learning user benign behaviors as a baseline, detect anomalies whenever there are deviations from these established models. However, as the business environment changes, users may exhibit new benign behaviors, leading to numerous false positives if the detection is solely based on deviations. Although existing approaches have considered concept drift to some extent, they have not adequately balanced the forgetting of historical knowledge with the learning of new knowledge.

3 Threat Model

Like prior PIDS research [8,26,29,37], our work addresses scenarios involving attackers who exploit software vulnerabilities to gain system control, deploy communication backdoors, and establish persistence. However, we exclude hardware trojans and side-channel attacks that evade detection through standard system audits. Furthermore, we rely on the integrity of audit log data ensured by existing security tracing systems and antitampering log technologies. Consequently, the provenance graphs constructed by Chaos are trustworthy and serve as an effective basis for detecting and analyzing cybersecurity threats.

4 Methodology

As illustrated in Fig. 2, Chaos is a provenance model that integrates spatial structure and long-range temporal dependencies, enabling efficient and accurate anomaly detection in massive audit logs with flexible scenario adaptation. Its workflow comprises four core components: (1) Provenance Graph Construction, (2) Spatio-Temporal Graph Representation Module, (3) Anomaly Detector, and (4) Model Adaptor.

4.1 Graph Construction

Chaos first constructs provenance graphs from raw audit logs (Windows and Linux logs). To enable efficient anomaly detection in massive logs while balancing performance overhead and preserving log structural integrity, the model employs soft segmentation to extract behavioral sub-events, providing suitable inputs for the subsequent spatio-temporal graph representation.

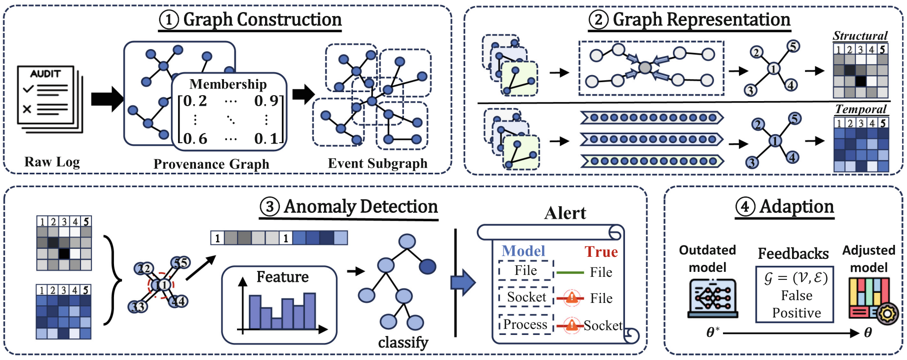


Fig. 2. Schematic of the CHAOS framework. Core components comprise: i) Provenance graph construction via log soft segmentation; ii) Spatio-temporal graph representation module integrating GNN and LSTM; iii) Anomaly detector based on benign behavior learning; iv) Model adaptor for handling concept drift. The model integrates spatial and temporal structure for efficient and accurate anomaly detection with scenario adaptation.

Log Parser. CHAOS constructs a system-wide provenance graph from data collected via Windows Event Logs or Linux Audit Logs. Consistent with existing provenance graph-based research [8,26], the model parses host-logged behaviors (such as process execution, file operations, and network connections) into nodes and relationships. CHAOS defines the provenance graph as G(subject, relation, object), where the subject is a process/thread, the object can be an entity (process/thread, file, or socket), and the relation represents the system call describing the host behavior.

SubEvent Extraction. Provenance logs are typically massive in volume. Existing approaches [13,26,29] often simply partition the data into K batches for processing. However, this coarse-grained segmentation can disrupt the graph structural information, hindering the perception of complete event context during node information aggregation. To overcome this limitation, we innovatively introduce fuzzy clustering into event extraction for provenance graphs, aiming to enhance extraction accuracy while maximizing preservation of structural information within the data. Fuzzy clustering establishes more flexible relationships between data points, enabling the association of similar events across different batches, thereby preventing information loss. Specifically: the raw provenance logs are first preprocessed to construct a high-dimensional feature space through feature extraction and standardization; then, a fuzzy clustering algorithm (e.g., Fuzzy C-Means) is applied to aggregate similar events. Each cluster centroid represents a potential event structure, and the membership degree of each log entry across different clusters reflects its relevance to each event. During event extraction, consideration extends beyond cluster centroids to also encompass the node's position and relational context within the entire graph structure.

4.2 Spatio-Temporal Graph Representation

Currently, anomaly node identification based on provenance graphs primarily relies on GNNs for graph representation learning. This approach encodes information from the neighborhood structure of nodes, capturing structural information of system behavior through node connectivity. However, shallow graph structural information fails to capture long-term dependencies, while deeper GNNs tend to suffer from severe overfitting and substantial computational overhead.

To address this, CHAOS proposes a spatio-temporal information fusion scheme integrating GNNs and LSTMs, it leverages GNNs to model spatial structural features and LSTMs to capture temporal dynamic characteristics. Building upon the sub-events extracted in the previous step, this scheme models them as sub-graphs and sub-sequences, respectively.

Spatial Feature Representation. To fully leverage the node membership degrees obtained from fuzzy clustering and enhance the spatial feature learning of sub-events, we employ a membership-based GNN. Specifically, the membership degrees are incorporated into the message passing mechanism of the GNN. This enables the model to consider nodes' affiliation degrees to clusters during neighbor information aggregation, thereby effectively reinforcing interactions among intra-cluster nodes while mitigating interference from inter-cluster nodes.

After sub-event extraction, we obtain the membership matrix $\mathbf{U} \in \mathbb{R}^{|V| \times c}$, where u_{vk} denotes the membership degree of node v to cluster k, satisfying $\sum_{k=1}^{c} u_{vk} = 1, \forall v \in V$. In traditional GNNs, node feature are updated through neighborhood feature aggregation. To integrate membership information, the message passing mechanism is modified as follows:

The feature update for node v at layer l is given by:

$$\mathbf{h}_v^{(l+1)} = \sigma \left(\sum_{u \in \mathcal{N}(v)} w_{uv} \cdot \mathbf{W}^{(l)} \mathbf{h}_u^{(l)} + \mathbf{b}^{(l)} \right), \tag{1}$$

where $\mathbf{h}_v$ is the feature representation, $\mathcal{N}(v)$ is the neighbor set of node v, $\mathbf{W}$ is a trainable weight matrix, $\mathbf{b}$ is a bias vector, σ denotes the activation function, and w_{uv} is the membership-based edge weight defined as:

$$w_{uv} = \sum_{k=1}^{c} f(u_{uk}, u_{vk}) \quad \text{with} \quad f(u_{uk}, u_{vk}) = \frac{u_{uk} + u_{vk}}{2}. \tag{2}$$

Leveraging the soft assignment property of fuzzy clustering, CHAOS retains nodes' multi-cluster membership relationships. By weighting edges with membership degrees, it enhances interactions among intra-cluster nodes, significantly improving substructure capture capability. Concurrently, it attenuates message passing between inter-cluster nodes, effectively suppressing noise interference.

Temporal Feature Representation. Given an input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ and edge set $\mathcal{E}$, each node v_i possesses a feature vector $\mathbf{x}_i \in \mathbb{R}^{d_{\text{in}}}$. To adapt to LSTM's sequential processing, nodes are chronologically ordered by timesteps $t_1 < t_2 < \cdots < t_N$ (where t_i denotes v_i's timestep identifier), forming the feature matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times d_{\text{in}}}$. This matrix is fed into the LSTM and processed over N timesteps to generate the hidden state sequence $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N]^\top \in \mathbb{R}^{N \times d_{\text{hidden}}}$, where $\mathbf{h}_i$ encodes temporal dependency features at t_i.

Spatio-Temporal Feature Fusion. To integrate the spatial structural features extracted by the GNN and the temporal dynamic characteristics captured by the LSTM network, a feature concatenation approach can be employed.

Assume that for any node v_i in the graph, the spatial feature vector obtained from the GNN is $\mathbf{h}_i^S \in \mathbb{R}^{d_S}$, where d_S is the dimension of the spatial features. Correspondingly, the hidden state from the LSTM at the relevant timestep (i.e., the temporal feature vector) is $\mathbf{h}_i^T \in \mathbb{R}^{d_T}$, where d_T is the dimension of the temporal features.

These two feature vectors are then concatenated to produce the final fused feature vector, $\mathbf{h}_i^F$:

$$\mathbf{h}_i^F = \mathbf{h}_i^S \oplus \mathbf{h}_i^T \tag{3}$$

where $\oplus$ denotes the vector concatenation operation. The resulting fused feature vector $\mathbf{h}_i^F$ has a dimension of $d_S + d_T$. This vector provides a comprehensive representation by combining the node's structural information with its temporal evolution characteristics, offering a richer basis for the downstream anomaly detection task.

4.3 Anomaly Detection

To effectively address the challenges of large-scale data volumes and high-dimensional features, we employ the lightweight gradient boosting framework, **LightGBM**, to perform APT detection after generating feature embeddings from our spatio-temporal representation module.

Inspired by prior work such as `Threatrace` [29] and `FLASH` [26], our model is trained exclusively on benign data. The objective is to learn the characteristic patterns of system entities under normal operating conditions. The underlying assumption is that the graph-structural and temporal features of malicious entities will significantly deviate from these learned benign patterns. LightGBM utilizes a histogram-based algorithm, which reduces the time complexity of finding optimal split points to $O(\text{N})$, making it highly efficient and largely independent of the feature count.

The final prediction from LightGBM is an aggregation of the outputs from multiple decision trees, formulated as:

$$f(x) = \sum_{k=1}^{K} f_k(x) \tag{4}$$

where K is the total number of trees and $f_k(x)$ is the output of the k-th tree. We use the sigmoid function to transform the raw prediction of each tree, $T_k(x)$, into a probability value:

$$f_k(x) = \text{sigmoid}(\alpha_k T_k(x)) \tag{5}$$

where α_k is a scaling factor for the tree's output. The classifier is trained by minimizing an objective function composed of a loss term and a regularization term:

$$\text{Obj} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{6}$$

During the detection phase, the fused spatial and temporal features generated by the spatio-temporal module are fed into the trained classifier to achieve efficient anomaly detection.

4.4 Model Adaption

As user behaviors evolve, IDS may generate false positives for benign behaviors unobserved during training. While existing systems (e.g., ThreaTrace, Flash) typically exclude this scenario by definition, and solutions like *ShadeWatcher* attempt adaptive mechanisms, most fail to effectively address the critical challenge: **selectively forgetting historical information when incorporating new knowledge** to prevent model bloat or obsolescence. To bridge this gap, CHAOS innovatively integrates the **EWC mechanism**.

After training on the old task, compute the Fisher Information Matrix:

$$F_i = \mathbb{E}_{x \sim D}\left[\left(\frac{\partial \log p(y|x;\theta)}{\partial \theta_i}\right)^2\right] \tag{7}$$

where $p(y|x;\theta)$ denotes the predicted output given input x under parameters θ, y is the ground-truth label, and D is the old task's dataset.

The loss function for new task training extends to:

$$\mathcal{L}(\theta) = \mathcal{L}_0(\theta) + \frac{\lambda}{2}\sum_i F_i(\theta_i - \theta_i^*)^2 \tag{8}$$

Here θ_i^* represents parameters learned from the old task, and λ is a regularization parameter controlling EWC term strength. By tuning λ, CHAOS dynamically balances knowledge retention and acquisition.

5 Evaluation

In this section, experiments are performed to validate CHAOS's advantage and answer the following key research questions

- **RQ1:** How effective is Chaos in detecting APTs compared to current state-of-the-art approaches?
- **RQ2:** What is the system overhead associated with using Chaos?
- **RQ3:** How effective are the components of our Chaos in achieving their intended functions?
- **RQ4:** How do different hyperparameters influence the detection capabilities of Chaos?
- **RQ5:** How adaptable is Chaos to system environment changes?

5.1 Experimental Setups

Dataset. We conducted our experimental evaluation on widely-used APT detection datasets, namely DARPA TC and Streamspot.

The DARPA Engagement 3 (E3) dataset, from the DARPA Transparent Computing program, contains audit logs from an enterprise network (Linux, Windows, BSD) during a two-week engagement. The Red Team launched APT attacks exploiting vulnerabilities, while the Blue Team defended. Notably, the Red Team generated benign background activity (e.g., web browsing) during attacks, providing normal behavior data. We evaluate on the Trace, THEIA, and CADETS sub-datasets.

The StreamSpot dataset collected and published by the StreamSpot team [34] using the SystemTap auditing system [41]. This dataset simulates system calls under 600 distinct scenarios (including YouTube, GMail, VGame, Drive-by-download attack, Download, and CNN), totaling 600 audit logs. Among these, 5 scenarios simulate benign user behavior, while the remaining scenarios simulate drive-by download attacks. The StreamSpot dataset does not provide labels for log entries or system entities. Hence, similar to prior work [10,15,17], we perform batch log-level anomaly detection on this dataset.

Implementation. We implement Chaos using PyTorch. The framework utilizes Python 3.9, employing the Gensim library for Word2Vec model training and the PyTorch library for implementing SAGE convolutions and LSTM modules. All experiments were conducted on an Ubuntu 18.04 LTS server equipped with an Intel 13th Gen Core i7-1360P CPU (12 cores, 16 threads at 2.20 GHz base frequency) and 32 GB RAM. To ensure fair comparison, we adopt the node-level intrusion detection evaluation protocol consistent with SOTA baselines including FLASH and ThreaTrace.

5.2 RQ1: Overall Detection Efficacy Comparsion

To comprehensively benchmark against SOTA solutions, we evaluate Chaos under both event-level and batch-processed log scenarios. For batch-processed logs, we adopted the classical StreamSpot dataset, comparing against its baseline methods and the graph-granularity solution Unicorn. As evidenced in Table 1, Chaos maintains superior detection performance at graph granularity. This is

Table 1. Chaos vs. SOTA: Detection Performance in Batch Logs

Method	Accuracy	Precision	Recall	F-Score
StreamSpot	0.93	0.73	0.91	0.81
Unicorn	**0.99**	0.95	0.97	0.96
ThreaTrace	**0.99**	0.98	**0.99**	**0.99**
FLASH	**0.99**	**1.0**	0.96	0.98
Chaos	**0.99**	0.99	**0.99**	**0.99**

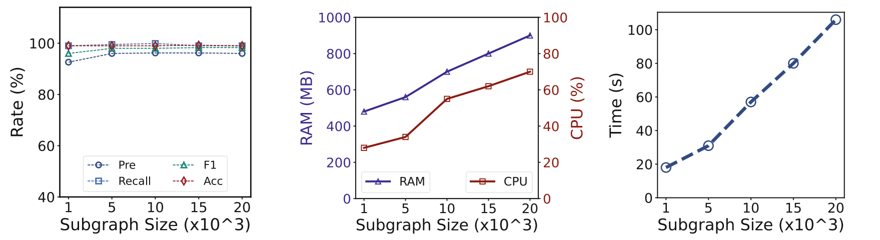
Table 2. Detection Performance in Event Logs

Dataset	Method	Prec.	Rec.	F-Score	TP / FP / FN / TN
	ThreaTrace	0.90	0.99	0.95	12848 / 1361 / 4 / 705,605
	FLASH	0.94	0.99	0.96	12851 / 818 / 1 / 706,148
Cadets (E3)	Chaos	0.96	0.99	0.98	12850 / 502 / 2 / 706,464
	ThreaTrace	0.72	0.99	0.83	67382 / 26774 / 1 / 2,389,233
	FLASH	0.95	0.99	0.97	67382 / 3477 / 1 / 2,412,530
Trace (E3)	Chaos	0.96	1.00	0.98	67382 / 2308 / 1 / 2,4123,699
	ThreaTrace	0.87	0.99	0.93	25,297 / 3765 / 65 / 3,501,561
	FLASH	0.92	0.99	0.95	25,318 / 2282 / 44 / 3,503,044
Theia (E3)	Chaos	0.93	0.99	0.96	25,330 / 1880 / 32 / 3,505,046

attributed to StreamSpot and Unicorn's reliance on simplistic frequency similarity and graph statistics, whereas Chaos employs deep learning to comprehensively characterize system behaviors. On the DARPA event-level dataset, Chaos was compared against node-level SOTA solutions ThreaTrace and FLASH. The results are shown in Table 2, Chaos achieves a significant reduction in false positives. We posit that: (1) ThreaTrace lacks semantic embedding capabilities, and (2) although FLASH incorporates Word2Vec for node semantics, its shallow GNN architecture fails to capture long-range dependencies. In contrast, Chaos effectively models deep long-range system behaviors through spatio-temporal correlation mechanisms, enabling more efficient anomaly detection.

5.3 RQ2: System Overhead

The performance overhead of spatiotemporal graphs primarily depends on the size of subgraph partitioning. Subgraph size directly impacts the sequence length and the time and memory overhead required for node propagation within the graph. We conducted performance overhead analysis using quantitative sizes of



(a) Detection rates with changes in subgraph size. (b) RAM and CPU overheads with changes in subgraph size. (c) Reference time with changes in subgraph size.

Fig. 3. Detection efficacy and efficiency of CHAOS across various subgraph sizes.

[1000, 5000, 10000, 15000, 20000]. As shown in Fig. 3a, the detection overhead of the spatiotemporal graph is proportional to the subgraph size. Furthermore, subgraph size directly influences the detection field of view, thereby affecting model performance. Figure 3 reveals that during detection, model performance improves as the subgraph size increases; however, performance stabilizes when the subgraph can fully encompass the event. Consequently, the CHAOS model achieves optimal detection performance under a reasonable system performance overhead.

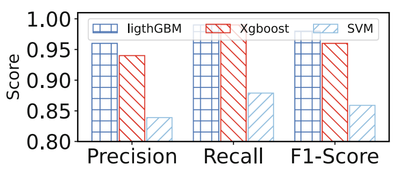
5.4 RQ3: Ablation Study

Table 3. Effects of Temporal and Spatial Information on the Detection Performance of CHAOS

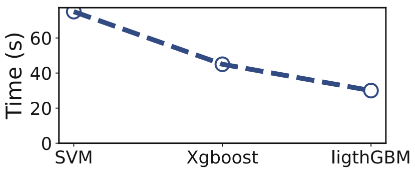
Method	Accuracy	Precision	Recall	F-Score	FPR
CHAOS	0.99	0.96	0.99	0.98	0.07%
w/o GNN	0.99	0.87	0.99	0.93	0.5%
w/o LSTM	0.99	0.94	0.99	0.96	0.11%

We systematically evaluated the performance of CHAOS by removing components, thereby demonstrating their impact on the system's effectiveness (Table 3).

Spatiotemporal Graph Ablation. To verify the influence of spatial topology and long-term temporal dependencies on modeling system behavior, we separately removed the GNN module and the LSTM module from the spatiotemporal graph representation learning. As shown in Table 5, the removal of either module resulted in a significant drop in detection capability. This proves that concurrently capturing long-term behavioral relationships and topological behavioral relationships is crucial for provenance detection: learning from long sequences effectively mitigates the impact of noise from local neighbors, while the graph topology clearly reflects the correlations of local behaviors.



(a) Impact of Different Classifiers on the Detection Performance of Models.

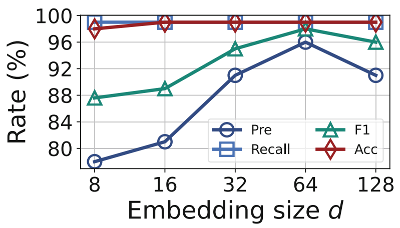


(b) Time overhead comparison.

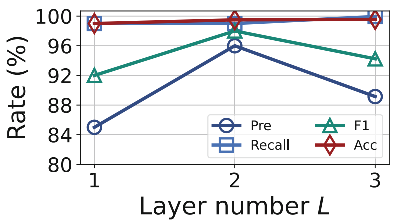
Fig. 4. Impact of Different Classifiers on Models.

Classifier Comparison. To achieve lightweight detection, we compared CHAOS against other classifiers such as SVM and XGBoost on the Cadets dataset, the results are shown in Fig. 4. Ultimately, we selected LightGBM for its lower computational overhead, achieving excellent detection performance while maintaining lower time complexity.

5.5 RQ4: Hyperparameter Analysis



(a) Impact of Different Embedding Dimensions on Model Detection Performance.



(b) Impact of Different GNN Layer on Model Detection Performance.

Fig. 5. Impact of Varying Hyperparameters on Model Performance.

In previous sections, we have evaluated the CHAOS model using fixed optimal hyperparameters. In this section, we conduct a tuning analysis on key hyperparameters that significantly impact model performance. We present the experimental analysis of hyperparameter tuning performed on the Cadets dataset.

Node Embedding Dimension. We investigate the impact of the initial node embedding dimensionality on the model's detection performance. In Fig. 5(a) experimental results demonstrate that embeddings with lower dimensionality are sufficient to characterize the initial node features effectively. Moreover, excessively large embedding dimensions may lead to sparse representations of the initial features and introduce unnecessary computational overhead.

Model Layer. Traditional GNN algorithms typically rely on deep architectures to capture global information and long-range dependencies. In contrast, the Chaos utilizes a temporal LSTM module to effectively capture long-range information, enabling it to achieve optimal performance with only a shallow GNN architecture. In Fig. 5(b), experiments show that increasing the number of GNN layers leads to performance degradation, indicating overfitting.

5.6 RQ5: Model Adaptability

Table 4. Effect of Model Adaptation on False Positive Reduction

	#Feedback	#TP	#TN	#FP	#FN	FPR
origin	0	12,850	337,943	302	2	0.08%
adjusted	200	12,850	338,245	80	2	0.02%

Chaos supports dynamic updates to its recommendation model by incorporating feedback from security analysts on false positives (FPs). We evaluate Chaos's adaptability in reducing the false positive rate (FPR) associated with normal system activities in the dataset. For the model adaptation scenario, we assume that security analysts can report entities that were incorrectly flagged as alerts (i.e., FPs). Chaos is then updated via resilient retraining. Comparing the FPR with and without model adaptation in Table 4, we observe a significant decrease after retraining. We argue that, in real-world settings, it is reasonable to expect security analysts to label a small number of FPs. Crucially, Chaos can effectively forget outdated historical patterns and relearn new system activity patterns as the environment evolves.

5.7 RQ6:Case Study

As illustrated in Fig. 6, we demonstrate Chaos's capabilities in threat detection and attack provenance using an attack trace from the Darpa-Cadets dataset. Chaos effectively captures long-range attack paths through its spatio-temporal graph representation while preserving attack graph integrity during subgraph partitioning. This attack exploited a backdoor vulnerability: an attacker (source IP: 81.49.200.166) connected to a vulnerable Nginx server and gained shell access. Leveraging this shell, the attacker downloaded a malicious payload (located

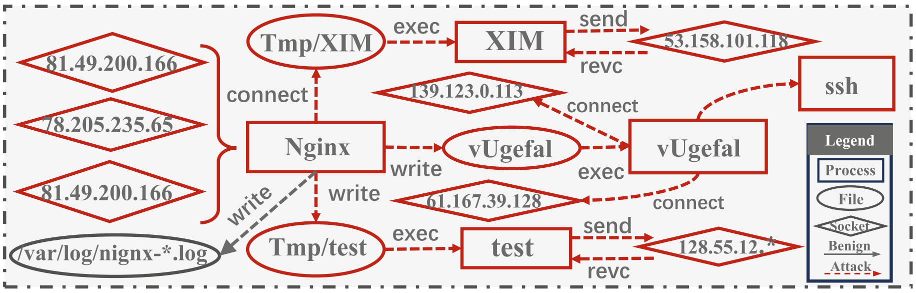


Fig. 6. An example of an attack provenance graph from the DARPA TC dataset [1].

at /tmp/vUgefal) onto the server and executed it with root privileges. The resulting privilege-escalated process, vUgefal, targeted lateral movement to IP 154.145.113.18 and 61.167.39.128. Besides, vUgefal attempted to inject a malicious payload into the sshd process, attempted to create malicious processes such as XIM and test, and conducted port scanning activities.

6 Related Work

6.1 APT Detection

Current mainstream solutions for APT detection primarily rely on provenance-based intrusion detection, categorized into rule-based and learning-based methods. Rule-based methods [9,12,22,23] leverage prior knowledge of known attacks to build heuristic rules, while learning-based methods [3,8,14,19,25,26,29,37] use deep learning to model benign or malicious behavior, often incorporating data mining techniques like clustering and classification [24]. Graph-based approaches, such as Graph Convolutional Networks (GCN) [16], encode structural dependencies in provenance graphs, with Graph Attention Networks (GAT) [27] introducing attention mechanisms for finer-grained relationships. Heterogeneous GNNs [30] show potential in malware detection by modeling cross-entity interactions but face challenges with dynamic provenance graphs in APT scenarios.

6.2 Spatio-Temporal

Numerous studies have focused on designing deep neural networks for spatio-temporal data prediction. To capture temporal dependencies, DLSTM [36] and GWN [33] have developed predictive frameworks based on Recurrent Neural Networks (RNNs) and Temporal Convolutional Networks (TCNs), respectively. GMAN [38] and STGNN [31] have employed temporal self-attention networks to facilitate long-range temporal learning. STGCN [35] and DCRNN [18] exploited GNNs for message passing between regions. Traditional deep learning methods such as ResNet [11] handle the long-distance dependencies of image sequences through residual joins.

7 Conclusion

In this paper, we introduce Chaos, a novel provenance tracing framework designed to address the unique challenges posed by APT attacks. By integrating a spatio-temporal fusion approach with an adaptive learning mechanism, Chaos overcomes the limitations of existing IDS. Our framework incorporates a Provenance Refinement Module that utilizes fuzzy clustering for structure-preserving log segmentation, ensuring the integrity of behavioral dependencies. The Spatio-Temporal Fusion Module jointly learns intricate behavioral semantics through dual-channel encoding, leveraging both GNNs for topological relationships and LSTMs for temporal dependencies. The Anomaly Inference Module, trained exclusively on benign patterns, effectively identifies subtle deviations, while the Adaptive Consolidation Module, powered by EWC, mitigates concept drift by intelligently balancing historical knowledge with adaptation to new benign behaviors. Through rigorous evaluation on the DARPA TC dataset, Chaos demonstrated superior accuracy and a significant reduction in false positive rates compared to state-of-the-art methods. These results highlight Chaos's robust generalizability and its ability to effectively detect unknown APT threats and adapt to evolving system environments, effectively achieving APT detection.

Acknowledgement. This research is funded by the National Key Research and Development Program of China (2022YFB3103505), Natural Science Basic Research Program of Shaanxi (No. 2025JC-JCQN-073), National Natural Science Foundation of China under Grant (No. 62272370), Young Elite Scientists Sponsorship Program by CAST (2022QNRC001), the China 111 Project (No. B16037), Qinchuangyuan Scientist + Engineer Team Program of Shaanxi (No. 2024QCY-KXJ-149), Songshan Laboratory (No. 241110210200), Open Foundation of Key Laboratory of Cyberspace Security, Ministry of Education of China (No. KLCS20240405) and the Fundamental Research Funds for the Central Universities (QTZX23071), the National Research Foundation, Singapore, and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-GC-2023-008), the National Research Foundation, Singapore, and the Cyber Security Agency under its National Cybersecurity R&D Programme (NCRP25-P04-TAICeN), the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme, and Ripple under its University Blockchain Research Initiative (UBRI) [5].

References

1. Darpa transparent computing program engagement 3 data release (2020). https://github.com/darpa-i2o/Transparent-Computing
2. The streamspot dataset (2016). https://github.com/sbustreamspot/sbustreamspot-data
3. Alsaheel, A., et al.: {ATLAS}: a sequence-based learning approach for attack investigation. In: 30th USENIX Security Symposium (USENIX Security 2021), pp. 3005–3022 (2021)

4. Cheng, Z., et al.: Kairos:: practical intrusion detection and investigation using whole-system provenance. arXiv preprint arXiv:2308.05034 (2023)
5. Feng, Y., Xu, J., Weymouth, L.: University blockchain research initiative (UBRI): boosting blockchain education and research. IEEE Potent. **41**(6), 19–25 (2022)
6. Goyal, A., Han, X., Wang, G., Bates, A.: Sometimes, you aren't what you do: mimicry attacks against provenance graph host intrusion detection systems. In: 30th Network and Distributed System Security Symposium (2023)
7. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in Neural Information Processing Systems, vol. 30 (2017)
8. Han, X., Pasquier, T., Bates, A., Mickens, J., Seltzer, M.: Unicorn: runtime provenance-based detector for advanced persistent threats. arXiv preprint arXiv:2001.01525 (2020)
9. Hassan, W.U., Bates, A., Marino, D.: Tactical provenance analysis for endpoint detection and response systems. In: 2020 IEEE Symposium on Security and Privacy (SP), pp. 1172–1189. IEEE (2020)
10. Hassan, W.U., et al.: NoDoze: combatting threat alert fatigue with automated provenance triage. In: Network and Distributed Systems Security Symposium (2019)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
12. Hossain, M.N., et al.: SLEUTH: real-time attack scenario reconstruction from cots audit. In: 26th USENIX Security Symposium (USENIX Security 17), pp. 487–504 (2017)
13. Jia, Z., Xiong, Y., Nan, Y., Zhang, Y., Zhao, J., Wen, M.: MAGIC: detecting advanced persistent threats via masked graph representation learning. arXiv preprint arXiv:2310.09831 (2023)
14. Kapoor, M., Melton, J., Ridenhour, M., Krishnan, S., Moyer, T.: PROV-GEM: automated provenance analysis framework using graph embeddings. In: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1720–1727. IEEE (2021)
15. Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
16. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
17. Li, T., Liu, X., Qiao, W., Zhu, X., Shen, Y., Ma, J.: T-trace: constructing the APTs provenance graphs through multiple syslogs correlation. IEEE Trans. Dependable Secure Comput. (2023)
18. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
19. Liu, F., Wen, Y., Zhang, D., Jiang, X., Xing, X., Meng, D.: Log2vec: a heterogeneous graph embedding based approach for detecting cyber threats within enterprise. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 1777–1794 (2019)
20. Liu, X., Masana, M., Herranz, L., Van de Weijer, J., Lopez, A.M., Bagdanov, A.D.: Rotate your networks: Better weight consolidation and less catastrophic forgetting. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 2262–2268. IEEE (2018)
21. Manzoor, E., Milajerdi, S.M., Akoglu, L.: Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In: Proceedings of the 22nd ACM SIGKDD

International Conference on Knowledge Discovery and Data Mining, pp. 1035–1044 (2016)
22. Milajerdi, S.M., Eshete, B., Gjomemo, R., Venkatakrishnan, V.: POIROT: aligning attack behavior with kernel audit records for cyber threat hunting. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 1795–1812 (2019)
23. Milajerdi, S.M., Gjomemo, R., Eshete, B., Sekar, R., Venkatakrishnan, V.: HOLMES: real-time apt detection through correlation of suspicious information flows. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 1137–1152. IEEE (2019)
24. Mining, W.I.D.: Data Mining: Concepts and Techniques, vol. 10, no. 559–569, p. 4. Morgan Kaufinann (2006)
25. Pei, K., et al.: HERCULE: attack story reconstruction via community discovery on correlated log graph. In: Proceedings of the 32Nd Annual Conference on Computer Security Applications, pp. 583–595 (2016)
26. Rehman, M.U., Ahmadi, H., Hassan, W.U.: FLASH: a comprehensive approach to intrusion detection via provenance graph representation learning. In: 2024 IEEE Symposium on Security and Privacy (SP), p. 139. IEEE Computer Society (2024)
27. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
28. Wang, L., et al.: Incorporating gradients to rules: Towards lightweight, adaptive provenance-based intrusion detection (2024)
29. Wang, S., et al.: ThreatRace: detecting and tracing host-based threats in node level through provenance graph learning. IEEE Trans. Inf. Forensics Secur. **17**, 3972–3987 (2022)
30. Wang, X., et al.: Heterogeneous graph attention network. In: The World Wide Web Conference, pp. 2022–2032 (2019)
31. Wang, X., et al.: Traffic flow prediction via spatial temporal graph neural network. In: Proceedings of the Web Conference 2020, pp. 1082–1092 (2020)
32. Wu, W., et al.: Brewing vodka: distilling pure knowledge for lightweight threat detection in audit logs. In: Proceedings of the ACM on Web Conference 2025, pp. 2172–2182 (2025)
33. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121 (2019)
34. Yang, F., Xu, J., Xiong, C., Li, Z., Zhang, K.: {PROGRAPHER}: an anomaly detection system based on provenance graph embedding. In: 32nd USENIX Security Symposium (USENIX Security 2023), pp. 4355–4372 (2023)
35. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875 (2017)
36. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., Liu, Y.: Deep learning: a generic approach for extreme condition traffic forecasting. In: Proceedings of the 2017 SIAM International Conference on Data Mining, pp. 777–785. SIAM (2017)
37. Zengy, J., et al.: SHADEWATCHER: recommendation-guided cyber threat analysis using system audit records. In: 2022 IEEE Symposium on Security and Privacy (SP), pp. 489–506. IEEE (2022)
38. Zheng, C., Fan, X., Wang, C., Qi, J.: GMAN: a graph multi-attention network for traffic prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 1234–1241 (2020)