MAGIC: Detecting Advanced Persistent Threats via Masked Graph Representation Learning

<u>Zian Jia</u>, Yun Xiong, Yuhong Nan, Yao Zhang, Jinjing Zhao, Mi Wen

上海市数据科学重点实验室
Shanghai Key Laboratory of Data Science

Advanced Persistent Threats (APTs)

Void Banshee APT Exploits Microsoft MHTML Flaw to Spread Atlantida Stealer

Jul 16, 2024 · Ravie Lakshmanan · Data Security / Vulnerability

Greece's Land Registry agency breached in wave of 400 cyberattacks

NEWS 1 AUG 2024

RansomEXX Group Targets Indian Banking With New Tactics

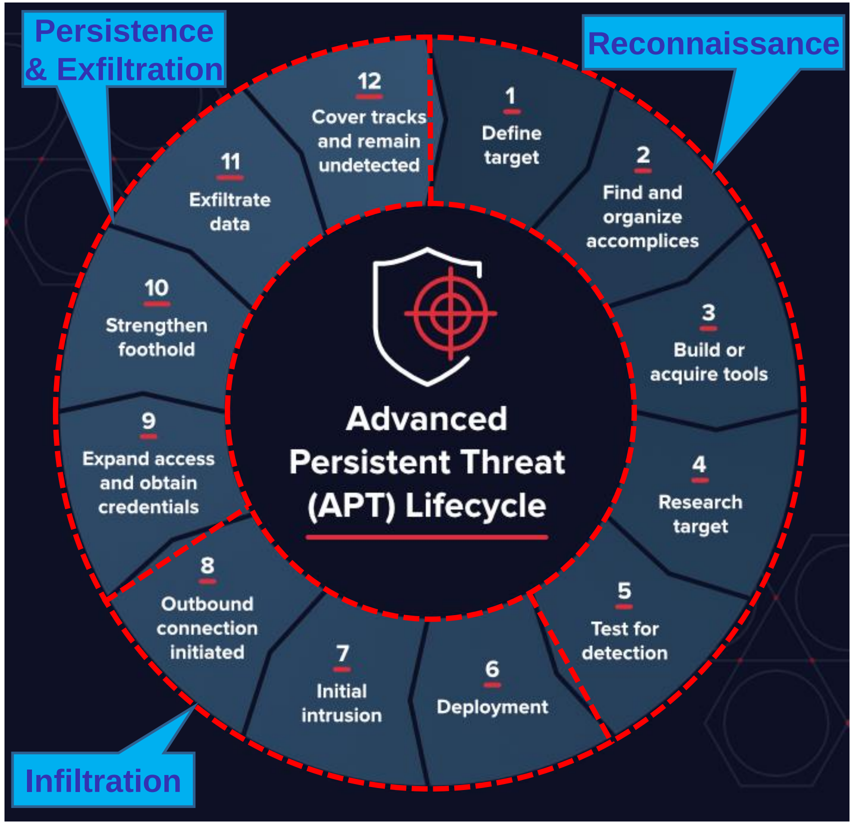
New 'HrServ.dll' Web Shell Detected in APT Attack Targeting Afghan Government

Nov 25, 2023 · Ravie Lakshmanan · Cyber Attack / Threat Intelligence

Plugins on WordPress.org backdoored in supply chain attack

NEWS 11 JUN 2024

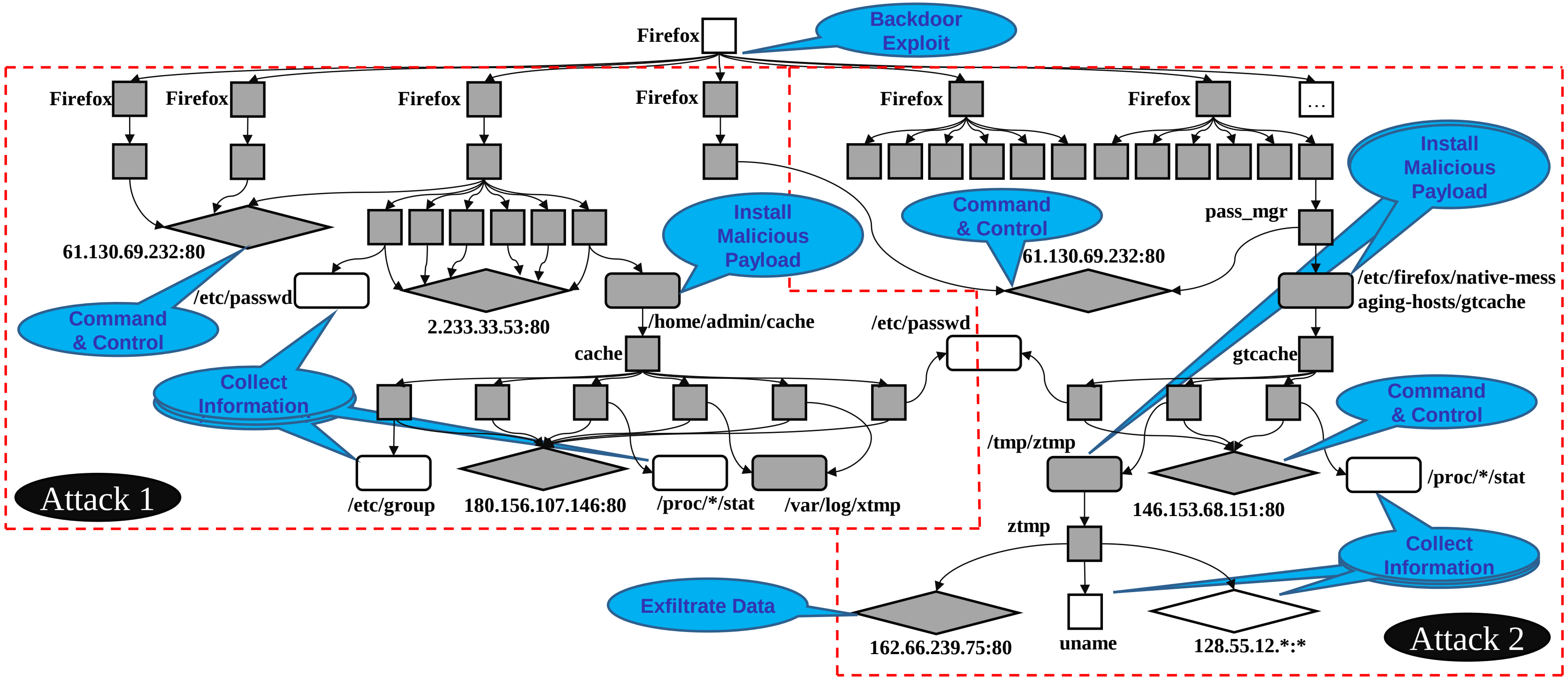
Threat Actor Breaches Snowflake Customers, Victims Extorted



[1] https://thehackernews.com/2024/07/void-banshee-apt-exploits-microsoft.html
[2] https://www.bleepingcomputer.com/news/security/greeces-land-registry-agency-breached-in-wave-of-400-cyberattacks/
[3] https://www.infosecurity-magazine.com/news/ransomexx-targets-indian-banking/
[4] https://thehackernews.com/2023/11/new-hrservdll-web-shell-detected-in-apt.html
[5] https://www.bleepingcomputer.com/news/security/plugins-on-wordpressorg-backdoored-in-supply-chain-attack/
[6] https://www.infosecurity-magazine.com/news/threat-actor-breaches-snowflake/

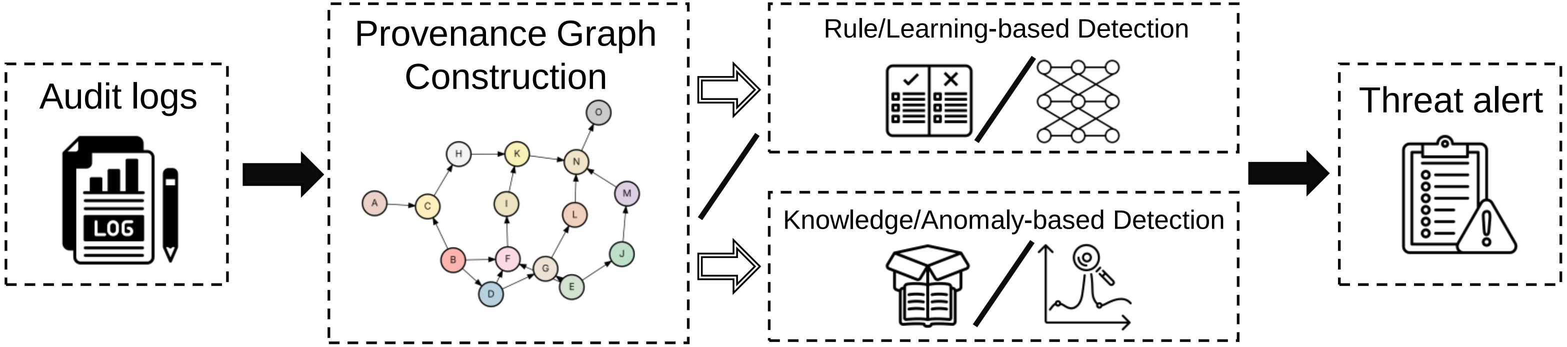
Provenance-based Intrusion Detection

- The construction of **provenance graphs** from **audit logs.**
  - System entities as *nodes* (e.g. processes, files and network flows);
  - System events between entities as *edges* (e.g. read, write, execute).



Provenance-based Intrusion Detection (cont.)

- **Rule**-based v.s. **Learning**-based Detection
  - Balance between *feature extraction* and *performance overhead*.
- **Attack-knowledge**-based v.s. **Anomaly**-based Detection
  - Attack knowledge ensures precise detection on *known attacks*.
  - Anomaly-based detection covers *unknown attacks* or *zero-day exploits*.



Existing Challenges and Design Goals

1

Reliance on attack knowledge

- Avoid *expert knowledge* or *extensive* attack data.
- Require robustness against *unknown* attacks.

MAGIC should be an *unsupervised anomaly-based* detector that identifies anomalous system behaviors as alerts.

2

Performance Overhead

- Balance between *deep* feature extraction and a *reasonable* performance overhead.

MAGIC should be able to extract *deep features* from provenance graphs with *minimum overhead*.

3

Lack of flexibility and scalability

- Call for detection in *finer granularities*.
- *Adapt to* new data and concept drift.

MAGIC should be a *flexible* solution with the capability of *multi-granularity detection* and *online adaptation*.

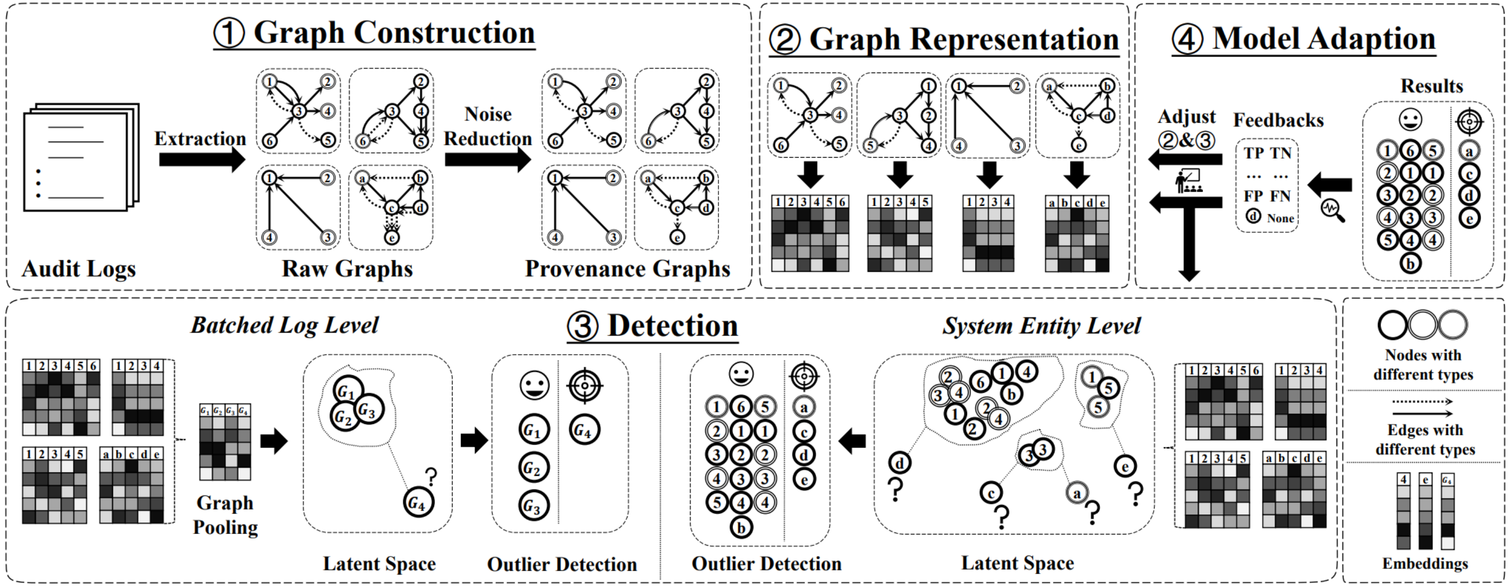
MAGIC Overview

① Construct provenance graphs from audit logs;

② Model system behaviors with Graph Representation Module (Multi-granularity);

③ Detect and alert anomalous behaviors with Outlier Detection (Multi-granularity);

④ Adapt MAGIC to false positives newly-arrived data.



Provenance Graph Construction

■Log parsing

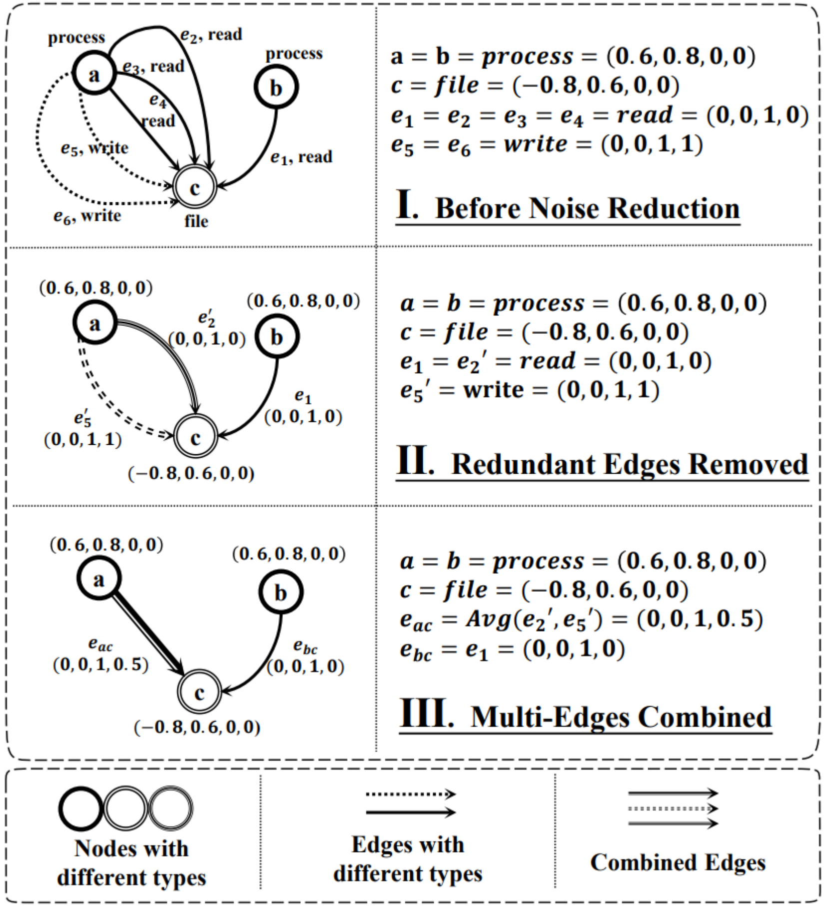
- ➢ System entities as *Nodes* and events as *Edges*;
- ➢ *Multi-label hashing* for Node and Edge types.

■Noise Reduction

- ➢ Keep first occurrence of *unique* triplet (SrcNode, EdgeType, DstNode);
- ➢ *Merge* triplets between node pairs as *final edges*.

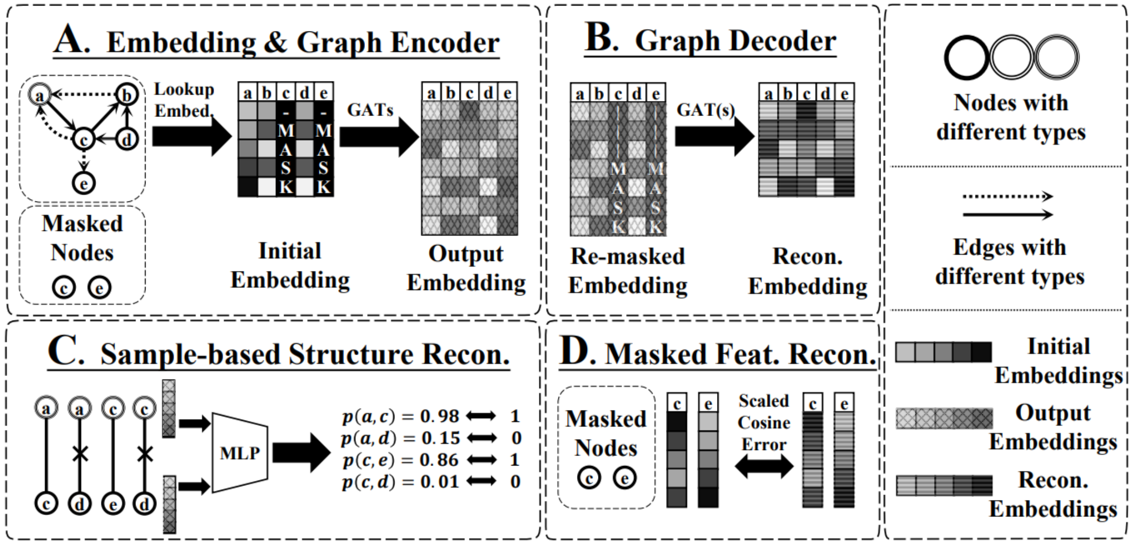
■Feature Embedding

- ➢ *Lookup Embedding* for Node and Edge types;
- ➢ Embeddings *summed up* for merged edges.



Graph Representation Module

- **(A/B/D) Graph Masked Auto-Encoder (GMAE)**
  - GAT Encoder + Decoder that *reconstructs node features*.
  - Excels at *efficiency* but misses *structural* information.
- **(C) Sample-based Structure Reconstruction**
  - Incorporates *structural* information with little increase in overhead.
- **(A) Output**
  - Node Embeddings (*at Entity-level*).
  - Graph Embeddings *after Pooling* (*at Batch-level*).



Detection Module

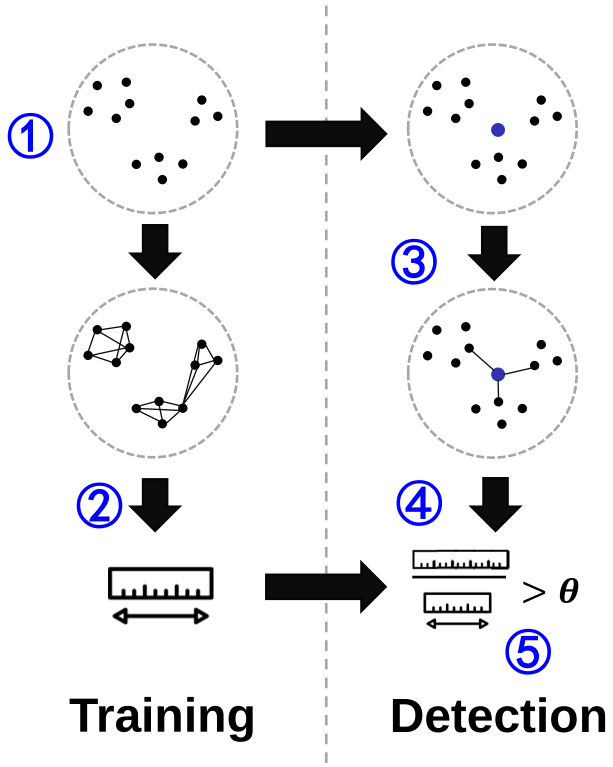
■ Simple outlier detection

■ Training

- ① *Memorizing* the benign embedding *distribution*;
- ② Computing the *standard dispersion* $\overline{dist}$ of the learnt distribution.

■ Detection

- ③ Finding *KNN* of the new embedding *within the learned distribution*;
- ④ Computing the *average distance* to its KNN relative to $\overline{dist}$ as anomaly score;
- ⑤ Raising alert when anomaly score above *threshold θ*.



Model Adaptation

- Adapt the **Graph Representation Module** with **any new data**
  - Improve *graph representation ability* with incremental training.

- Adapt the **Detection Module** with **false positives** and **new benign data**
  - Memorize new *benign behaviors*;
  - *Forget* old data;
  - *Adjust* the learned benign *distribution*.

Evaluation Setup

■Batch-level Detection Datasets

➢ *Streamspot*[1] and *Unicorn Wget*[2] dataset.

Dataset	# Attack batches	# Benign Batches	Avg. #Entity	Avg. #Event
StreamSpot	100	500	8,410	149,618
Unicorn Wget	25	125	264,046	971,003

■Entity-level Detection Datasets

➢ DARPA Transparent Computing[3] sub-datasets *E3-Trace*, *E3-THEIA* and *E3-CADETS.*

Dataset	# Malicious Entity	# Benign Entity	# Event
E3-Trace	68,082	3,220,594	4,080,457
E3-THEIA	25,319	1,598,647	2,874,821
E3-CADETS	12,846	1,614,189	3,303,264

[1] https://github.com/sbustreamspot/sbustreamspot-data.
[2] https://dataverse.harvard.edu/dataverse/unicorn-wget.
[3] https://github.com/darpa-i2o/Transparent-Computing.

Evaluation Results

Granularity	Dataset	Recall	False Positive Rate	Precision	F1-Score	AUC
Batch	**Streamspot**	100.00%	0.59%	99.41%	99.71%	99.95%
	Unicorn Wget	96.00%	2.00%	98.02%	96.98%	96.32%
Entity	**E3-Trace**	99.98%	0.09%	99.17%	99.57%	99.99%
	E3-THEIA	99.99%	0.14%	98.23%	99.11%	99.87%
	E3-CADETS	99.77%	0.22%	94.40%	97.01%	99.77%

MAGIC yields high recall and low FPR on different datasets and various granularities of detection, supporting the effectiveness and universality of **MAGIC**'s "**behavioral modeling, then outlier detection**" **detection framework**.

Evaluation Results (cont.)

Dataset	System	Supervision	F1-Score	Recall	FPR	Precision
StreamSpot	Unicorn	Benign	0.96	0.93	0.016	0.95
	Prov-Gem	All	0.97	0.94	**0.000**	**1.00**
	ThreaTrace	Benign	0.99	0.99	0.004	0.98
	MAGIC	Benign	**0.99**	**1.00**	0.006	0.99
Unicorn Wget	Unicorn	Benign	0.90	0.95	0.155	0.86
	Prov-Gem	All	0.89	0.80	**0.000**	**1.00**
	ThreaTrace	Benign	0.95	**0.98**	0.074	0.93
	MAGIC	Benign	**0.97**	0.96	0.020	0.98
E3-Trace	ShadeWatcher	Semi	0.99	**0.99**	0.003	0.97
	ThreaTrace	Benign	0.83	0.99	0.011	0.72
	MAGIC	Benign	**0.99**	0.99	**0.001**	**0.99**
E3-THEIA	ThreaTrace	Benign	0.93	0.99	**0.001**	0.87
	MAGIC	Benign	**0.99**	**0.99**	0.001	**0.98**
E3-CADETS	ThreaTrace	Benign	0.95	**0.99**	0.002	0.94
	MAGIC	Benign	**0.97**	0.99	**0.002**	**0.97**

MAGIC outperforms previous works with only benign data for training.

Evaluation Results (cont.)

<table>
<tr><th rowspan="2">Phase</th><th rowspan="2">Component</th><th colspan="2">Time(s)</th><th rowspan="2">Memory(MB)</th></tr>
<tr><th>GPU</th><th>CPU</th></tr>
<tr><td>Graph Construction</td><td>N/A</td><td colspan="2">642</td><td>2,610</td></tr>
<tr><td rowspan="2">Training</td><td>Graph Representation</td><td>151</td><td>685</td><td>1,564</td></tr>
<tr><td>Detection Module</td><td colspan="2">78</td><td>1,320</td></tr>
<tr><td rowspan="2">Detection</td><td>Graph Representation</td><td>5</td><td>10</td><td>2,108</td></tr>
<tr><td>Detection Module</td><td colspan="2">825</td><td>1,667</td></tr>
</table>

MAGIC operates with **minimum overhead**, times faster than state-of-the-art, granting it applicability under various conditions.

Train Ratio	Adaptation	FPR
80%	N/A	0.00089
20%	N/A	0.00426
20%	FP & TN in Next 40%	0.00173
20%	FP in Next 20%	0.00272
20%	FP & TN in Next 20%	0.00220

MAGIC adapts to changes in benign behaviors by incremental training on new benign data.

Other Experiments

■ Ablation Study

- Compare the effect of different *reconstruction principles* on overall performance.
- Evaluate the impact of different *hyperparameters*, including the embedding dimension d, the number of GMAE encoder layers l, and the mask rate r.

■ Sensitivity Analysis

- Discuss the *sensitivity* of the detection threshold θ and the *separation* between anomaly scores.

■ Robustness against Adversarial Attacks

- Evaluate MAGIC's *robustness* against adversarial attacks, including *evasion* (*mimicry*) and *poison attacks*.

Conclusion

- **MAGIC, an unsupervised, provenance-based APT detection approach**
  - **Simple detection pipeline of “behavioral modeling, then outlier detection”**
    - *Unsupervised behavior-based* Detection.
    - *Multi-granularity* Detection.
    - *Adaptation to changes* in benign behaviors.
  - **Efficiency-oriented design**
    - *Masked Graph Representation Learning* with *sample-based structure learning*.
    - *CPU-friendly detection module*.
  - **Evaluation results over various datasets**
    - *Effectively* detects APTs in different granularities and situations, *with minimum overhead*.

MAGIC: Detecting Advanced Persistent Threats via Masked Graph Representation Learning

Thank you for listening!

https://github.com/FDUDSDE/MAGIC

Zian Jia, jimmyokokok@gmail.com

FUDAN UNIVERSITY 1905

Datalogy
上海市数据科学重点实验室
Shanghai Key Laboratory of Data Science

中山大學 SUN YAT-SEN UNIVERSITY 1924