# Survey on system-level graph-based and anomaly-based intrusion detection

Fanny Dijoud, Pierre-François Gimenez, Michel Hurfin, Frédéric Majorczyk, Barbara Pilastre

# Survey on system-level graph-based and anomaly-based intrusion detection

Fanny Dijoud*, Pierre-François Gimenez†, Michel Hurfin*, Frédéric Majorczyk‡, Barbara Pilastre§
*Inria, Univ. Rennes, IRISA, {firstname.lastname}@inria.com
†CentraleSupélec, Univ. Rennes, IRISA, pierre-francois.gimenez@centralesupelec.fr
‡DGA-MI, Univ. Rennes, IRISA, frederic.majorczyk@intradef.gouv.fr
§DGA-MI, barbara.pilastre@intradef.gouv.fr

***Abstract*—Intrusion Detection Systems (IDS) are tools for monitoring a system, in order to identify potential malicious activities within it. This survey presents an analysis of graph-based anomaly-based IDS at system level. We also present open issues regarding those IDS, and propose a taxonomy of suitable features to compare them.**

***Index Terms*—intrusion detection system, anomaly detection, system security, machine learning, provenance graph**

## I. Introduction

Most widely used Intrusion Detection Systems (IDS) are signature-based such as Snort, but these IDS are not able to detect new attacks like zero-day attacks. Nowadays, the widely use of Artificial Intelligence (AI) in anomaly-based IDS makes new attacks detectable since these IDS do not rely on a characterization of attacks. Anomaly-based detection consists in profiling normal behavior from logs (i.e. recorded operations) and defining abnormal behavior by the degree of deviation from normal behavior. System-level logs involve APT-detection useful details: process identifier, file path, etc.

This survey provides system-level graph-based and anomaly-based IDS' state of the art, a taxonomy of suitable features and open issues.

## II. Focus on Ten Recent Solutions

We compare ten state-of-the-art approaches according to the following criteria: datasets, graph definition, graph embedding, detection level, machine learning processing, supervised or not, and the results of experiments (see Tables I, II and III).

### A. Datasets

Several datasets are used and each has its own specification, making IDS' comparison arduous. We focus on two very different widely-used datasets: StreamSpot and DARPA OpTC.

StreamSpot [11] contains logs of 6 scenarios (each reproduced 100 times and gathered in 100 graphs), including only 1 attack scenario. Thus this dataset is unrepresentative of the potential diversity of attacks and activities. However StreamSpot is popular because it is well labeled, gives first results for an approach, and its size is manageable (i.e. small).

DARPA Operationally Transparent Cyber (DARPA OpTC) [4] is 1.5+ TB of compressed JSON imprecise-labeled logs of the activity of 1,000 Windows 10 during 7 days. OpTC contains 3 red-activity days. Despite its scripted nature, the dataset is closer to real-world logs by its complexity.

### B. Building a graph

Most approaches turn logs to graphs according to the definition used in Backtraker [8], called *Provenance graphs*. They are directed graphs where nodes are system's objects and edges underline a source object, a sink object, and a timestamp. These graphs highlight causal/temporal links, resulting in a clearer intelligible vision of activities.

In Figure 1, *edge 1* represents process B doing an action (e.g. reading) on *file 1* at timestamp 1. Objects and edges sets may differ according to approaches: limited information into logs brings to limited objects/edges sets; non-obvious objects called abstract object (AO) can be added [2] (e.g. sensitive data object), etc.

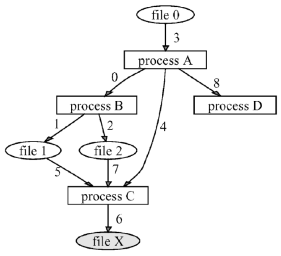

Fig. 1. Provenance graph (from [8])

These AO stress that sets' definition can significantly vary the intern structure of graphs: bringing in the same neighborhood, objects that would otherwise have been far apart. In Figure 1, if *file 0* and *file X* share sensitive data, they would be linked to a sensitive data AO, making them only two hops away from each other, which can be crucial when embedding.

### C. Embedding

The goal of the embedding is to transform essential graphs' information into an input for AI processing, while retaining the ability to scale up: classic information extraction methods such as adjacency matrix are out of scope.

Many different embeddings have been proposed, depending on the chosen detection level and AI processing. Approaches thus use graph objects or graphs/edges/nodes' features and neighborhood original/low-level embedding such as one-hot encoding, NLP methods, etc. (see Table I). Time can be included in the embedding but it is often taken into account by using time window or snapshots when defining the graphs.

### D. Anomaly detection

To compare the effectiveness of different IDS, we use common metrics: accuracy, precision, recall and F1-score [9].

In anomaly detection, the definition of normal behavior has a significant impact on the efficacy of the IDS. An attack

| Title | Year | Detection level | Datasets | Graphs | Embedding | AI Processing | Supervised |
|---|---|---|---|---|---|---|---|
| **UNICORN** [5] | 2020 | Graph | DARPA TC3, StreamSpot | Provenance graph | Histogram | Cluster | No |
| **ANUBIS** [1] | 2022 | Event traces | DARPA OpTC | Provenance graph | Contextual/Causal and neighborhood | LSTM, BNN | Yes |
| **PIKACHU** [10] | 2022 | Events | DARPA OpTC, LANL | Provenance graph | Skipgram | GRU-AE | No |
| **APT-KGL** [2] | 2022 | Node | DARPA TC, custom | Abstract object | Meta-path, HGAT | R-GCN | Yes |
| **SHADEWATCHER** [14] | 2022 | Events | DARPA Trace (TC), custom | Provenance graph, Bipartite | Neighborhood | GNN | Add human |
| **PROGRAPHER** [12] | 2023 | Graph | DARPA 3, DARPA Engagement ATLAS, StreamSpot, EDR | Provenance graph | Graph2vec | RCNN | No |
| **GCA(-SA)** [13] | 2023 | Graph | StreamSpot | Provenance graph | GCN | AutoEncoders | No |
| **EDGETORRENT** [7] | 2023 | Graph | StreamSpot, UNSW-NB15, DARPA TCE5 | Provenance graph | MPNN | GAN | No |
| **KAIROS** [3] | 2023 | Time window | DARPA3, DARPA5, DARPA OpTC, StreamSpot | Provenance graph | Feature hashing | AutoEncoder (TGN, MLP) | No |
| **MAGIC** [6] | 2023 | Node | StreamSpot, Unicorn Wget dataset, DARPA E3 | Provenance graph | Lookup | AE (GAT), Outlier (KNN) | No |

TABLE I
STATE-OF-ART SYSTEM-LEVEL GRAPH-BASED AND ANOMALY-BASED IDS.

| Title | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| **UNICORN** [5] | 0.96 | 0.98 | 0.93 | 0.94 |
| **GCA** [13] | 0.988 | 1.0 | 0.925 | 0.961 |
| **GCA-SA** [13] | 0.970 | 0.851 | 0.978 | 0.909 |
| **PROGRAPHER** [12] | 0.94 | 0.90 | 1.0 | 0.94 |
| **EDGETORRENT** [7] | 0.934 (1.0*) | 0.883 (1.0*) | 0.972 (1.0*) | 0.924 (1.0*) |
| **KAIROS** [3] | 1.0 | 1.0 | 1.0 | 1.0 |
| **MAGIC** [6] | 0.997 | 0.994 | 1.0 | 0.997 |

TABLE II
RESULTS OF APPROACHES ON STREAMSPOT. (*AFTER OPTIMIZATION)

| Title | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| **PIKACHU** [10] | × | × | 0.987 | × |
| **ANUBIS** [1] | 0.993 | 0.99 | 1.0 | 0.996 |
| **KAIROS** [3] | 0.987 (0.995*) | 0.579 (0.842*) | 1.0 | 0.733 (0.914*) |

TABLE III
RESULTS OF APPROACHES ON DARPA OPTC. (×: NO INFORMATION)

falsely considered as normal behavior, will not trigger alarms (false negative). In Table II and Table III, the recall is high: the normal behavior profile has been meticulously constructed and contains only normal behavior.

The profile of the normal behavior implies that all possible normal behaviors are represented in this profile. However it is difficult to do so because normal behavior is too vast and unstable to be represented in its entirety. Thus alerts can be raised wrongly (false positive FP). In Tables II and III, the precision highlights a quite high FP rate: real activity being more complex, precision less than 0.99 for these simple/scripted datasets, makes approaches unpractical.

AI-based detection approaches can be either supervised or unsupervised. Supervised learning needs labeled data (i.e. spot attacks' logs). However, even in a controlled setup, labeling data needs time from experts. So in practice the labeled data are often not sufficient to obtain a consistent dataset for learning. However unsupervised methods are currently less efficient than supervised ones: they start to be efficient on simple datasets, but for more realistic datasets like DARPA OpTC, the precision is too low: KAIROS [3], with no optimization, has 16 FP, means 16 time-windows of 15 minutes, which correspond to 4 hours of FP activity out of 3 days of activity.

## III. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Analysis of state-of-the-art approaches indicates that simple datasets are often used when more real-world scaled datasets should be the base of experiments. Moreover, time embedding, which is crucial to detect complex attacks, still need to be explored. Regarding experiments, the most effective approaches, on closer real-world datasets, are still supervised. However these approaches need labelled data, which is difficult to obtain in real environment. On the contrary, unsupervised approaches do not need labeled data and can be set up in a real environment more easily. However, those IDS are not yet fully mature: the FP rate is high which makes these approaches not suitable in a real-world environment for now.

The goal of my PhD is to build an unsupervised graph-based system-level anomaly-based approach that can deal with real-world datasets and perform anomaly detection in reasonable time with few false alarms.

## IV. ACKNOWLEDGEMENT

We thank the Brittany Region and the Directorate General of Armaments (DGA), in the context of CREACH LABS.

## REFERENCES

[1] MM Anjum, S Iqbal, and B Hamelin. Anubis: a provenance graph-based framework for advanced persistent threat detection. In *Proc. of the 37th ACM/SIGAPP Symp. on Applied Computing*, pages 1684–1693, 2022.

[2] T. Chen, C. Dong, M. Lv, Q. Song, H. Liu, T. Zhu, K. Xu, L. Chen, S. Ji, and Y. Fan. Apt-kgl: An intelligent apt detection system based on threat knowledge and heterogeneous provenance graph learning. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[3] Z. Cheng, Q. Lv, J. Liang, D. Wang, Y.and Sun, T. Pasquier, and X. Han. Kairos: Practical intrusion detection and investigation using whole-system provenance, 2023.

[4] Darpa optc. https://github.com/FiveDirections/OpTC-data, Accessed: 2024-01-20.

[5] X. Han, T. Pasquier, A. Bates, J. Mickens, and M. Seltzer. Unicorn: Runtime provenance-based detector for advanced persistent threats. *arXiv preprint arXiv:2001.01525*, 2020.

[6] Z. Jia, Y. Xiong, Y. Nan, Y. Zhang, J. Zhao, and M. Wen. Magic: Detecting advanced persistent threats via masked graph representation learning, 2023.

[7] I. J. King, X. Shu, J. Jang, K. Eykholt, T. Lee, and H. H. Huang. Edgetorrent: Real-time temporal graph representations for intrusion detection. In *Proc. of the 26th Int. Symposium on Research in Attacks, Intrusions and Defenses*, RAID '23, page 77–91, 2023.

[8] S. T. King and P. M. Chen. Backtracking intrusions. In *Proc. of the 19th ACM symp. on Operating systems principles*, pages 223–236, 2003.

[9] Wikipedia. https://en.wikipedia.org/wiki/Precision_and_recall, Accessed: 2024-01-20.

[10] R. Paudel and H. H. Huang. Pikachu: Temporal walk based dynamic graph embedding for network anomaly detection. In *IEEE/IFIP Network Operations and Management Symp. (NOMS)*, pages 1–7, 2022.

[11] Streamspot database. https://github.com/sbustreamspot/sbustreamspot-data, Accessed: 2024-01-20.

[12] F. Yang, J. Xu, C. Xiong, Z. Li, and K. Zhang. Prographer: An anomaly detection system based on provenance graph embedding. In *32nd USENIX Security Symposium*, pages 4355–4372, 2023.

[13] M. Ye, S. Men, L. Xie, and B. Chen. Detect advanced persistent threat in graph-level using competitive autoencoder. In *Proc. of the 2nd Int. Conf. on Networks, Communications and Information Technology*, pages 28–34, 2023.

[14] J. Zengy, X. Wang, J. Liu, Y. Chen, Z. Liang, T.S. Chua, and Z.L. Chua. Shadewatcher: Recommendation-guided cyber threat analysis using system audit records. In *IEEE Symp. on Security and Privacy (SP)*, pages 489–506. IEEE, 2022.