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@intradeff.gouv.fr
§DGA-MI, barbara.pilastre@intradeff.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 Backtracer [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.

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].
false 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, makes approaches unpractical.

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


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.

TABLE II

RESULTS OF APPROACHES ON STREAMSPOT. (*: AFTER OPTIMIZATION)

<table>
<thead>
<tr>
<th>Title</th>
<th>Year</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANUBIS [14]</td>
<td>2022</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>0.99</td>
</tr>
<tr>
<td>MAGIC [6]</td>
<td>2023</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>0.99</td>
</tr>
</tbody>
</table>

TABLE III

RESULTS OF APPROACHES ON DARPA OP/TC. (x: NO INFORMATION)

<table>
<thead>
<tr>
<th>Title</th>
<th>Year</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPKACH [10]</td>
<td>2022</td>
<td>0.99</td>
<td>x</td>
<td>1.00</td>
<td>0.99</td>
</tr>
<tr>
<td>ANUBIS [14]</td>
<td>0.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KAIROS [3]</td>
<td>0.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE I

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

<table>
<thead>
<tr>
<th>Title</th>
<th>Year</th>
<th>Detection level</th>
<th>Datasets</th>
<th>Graphs</th>
<th>Embedding</th>
<th>AI Processing</th>
<th>Supervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNICORN [2]</td>
<td>2020</td>
<td>Graph</td>
<td>DARPA TC, StreamSpot</td>
<td>Provenance graph</td>
<td>Histogram</td>
<td>0.987 (0.995*)</td>
<td>Yes</td>
</tr>
<tr>
<td>ANUBIS [14]</td>
<td>2022</td>
<td>Event/Files</td>
<td>DARPA Op/TC</td>
<td>Provenance graph</td>
<td>Contextual/Causal and neighborhood</td>
<td>0.997</td>
<td>No</td>
</tr>
<tr>
<td>PPKACH [10]</td>
<td>2022</td>
<td>Events</td>
<td>DARPA Op/TC, LAN</td>
<td>Provenance graph</td>
<td>Stigmaria</td>
<td>ORCA*</td>
<td>Yes</td>
</tr>
<tr>
<td>SHADOWWATCHER [14]</td>
<td>2023</td>
<td>Events</td>
<td>DARPA Trace (TC), custom</td>
<td>Provenance graph, Biparite</td>
<td>Neighborhood</td>
<td>GNU</td>
<td>Add human</td>
</tr>
<tr>
<td>PROGRAPHER [12]</td>
<td>2023</td>
<td>Graph</td>
<td>DARPA/ DARPAT Engage, ATLAS, StreamSpot, EDR</td>
<td>Provenance graph</td>
<td>Graph2vec</td>
<td>RCNN</td>
<td>No</td>
</tr>
<tr>
<td>GCA-SA [13]</td>
<td>2023</td>
<td>Graph</td>
<td>StreamSpot</td>
<td>Provenance graph</td>
<td>GCS</td>
<td>AutoEncoders</td>
<td>No</td>
</tr>
<tr>
<td>EDGETORRENT [7]</td>
<td>2023</td>
<td>Graph</td>
<td>StreamSpot, UNICORN, DARPA TCES</td>
<td>Provenance graph</td>
<td>MPNN</td>
<td>GAN</td>
<td>No</td>
</tr>
</tbody>
</table>