Learning Intrusion Tracking from Simulations

Nikolaos Kakouros, Pontus Johnson, Mathias Ekstedt
Division of Network and Systems Engineering
KTH Royal Institute of Technology
Stockholm, Sweden
Email: {nkak, pontusj, mekstedt}@kth.se

***Abstract*—This paper presents an approach to using Graph Neural Networks (GNNs) for intrusion detection, leveraging the MAL simulator for generating synthetic attack data. The proposed system utilizes telemetry events and their associations to predict the state of compromise in a network.**

***Index Terms*—cyber security, graph learning**

I. Introduction

A. Intrusion detection has a long history

Discuss the evolution and importance of intrusion detection systems (IDS) in cybersecurity.

B. A lack of labeled training data

Highlight the challenge of obtaining labeled data for training IDS, crucial for machine learning approaches.

C. MAL: We have a simulator

Introduce the MAL simulator developed for generating synthetic attack data.

D. GNNs are suitable DNNs

Explain why Graph Neural Networks (GNNs) are appropriate for this application due to their ability to handle graph-structured data.

E. Outline

II. Related Work

A. Intrusion detection

Review existing intrusion detection techniques and their limitations.

B. MAL

Discuss the features and capabilities of the MAL simulator.

C. GNNs

Overview of GNNs and their applications in various domains, including cybersecurity.

III. Requirements for GNN Training

A. GNNs need a graph

Explain the necessity of graph structures for GNNs.

B. The information needed should be in the neighborhood

Detail the requirement for relevant information to be accessible within the graph neighborhood.

C. Additional requirements...

IV. MAL 2.0

Explain how MAL 2.0 extends MAL 1.0

A. Telemetry events

- Define telemetry events and their role in the proposed system.
- Telemetry events are connected to each other sequentially through a next association.
- Telemetry events need to be connected to attack steps through a trigger association, which needs to include an accuracy estimate.
- Telemetry events can also be associated to assets, in the case of our prototype, accounts and IP addresses.

Here are examples of telemetry events:

```
asset VM {
    | root_shell(IP ip)
        e meterpreter_detected(IP ip) [0.1]
}
```

where the event meterpreter_detected will be emitted by some threat detection system in 10% of the cases where the attacker has gained root shell access to the VM. Note that we no longer have false positives, because they will be generated by user emulators. Here is another example:

```
asset Bucket {
    | list_objects(Account account, IP ip)
        e list_object(Account account, IP ip) [1.0
}
```

where the event list_object will be logged with 100

B. Next associations

- Describe the sequential linking of telemetry events.
- One good thing with creating a dedicated time step graph as suggested above, is that it does not need to include the complete attack graph, as it can be limited to the nodes that are N-neighbors to the log events.

C. Trigger associations

- Explain the associations between telemetry events and attack steps, including accuracy estimates.
- Note that we no longer need confusion matrices as the false positives will be generated by the legitimate events generator (user emulation).

D. *Information association*

- Discuss how telemetry events are connected to assets such as accounts and IP addresses.
- Telemetry events need to be connected to attack steps through a trigger association, which needs to include an accuracy estimate.

V. Example

A. *A simple MAL spec*

Provide a simplified specification example for the MAL simulator. This example could be for a generic cloud provider, or even to an on-prem system. It is very important that it dose not contain anything links it to any particular provider.

B. *A simple instance model*

Describe a basic instance model used for simulations.

```
Project project_26
Bucket bucket_49
Bucket bucket_75
BucketObject data_113
BucketObject data_118
BucketObject data_231
BucketObject data_236
BucketObject data_321
BucketObject data_546

Account pontusj, mathiase
IP 17_233_12_98, 130_237_44_15

owns(project_26, bucket_49)
owns(project_26, bucket_75)
contains(bucket_49, data_113)
contains(bucket_49, data_118)
contains(bucket_75, data_231)
contains(bucket_75, data_236)
contains(bucket_75, data_321)
contains(bucket_75, data_546)
```

C. *The resulting attack graph*

From the attack graph and an event log (synthetic or real), at each time step, we can generate a graph to present to the GNN 1.

D. *The resulting GNN*

Show the graph structure fed into the GNN for training 2.

E. *Results from a small training round*

Provide preliminary results from a small-scale training session with the GNN.

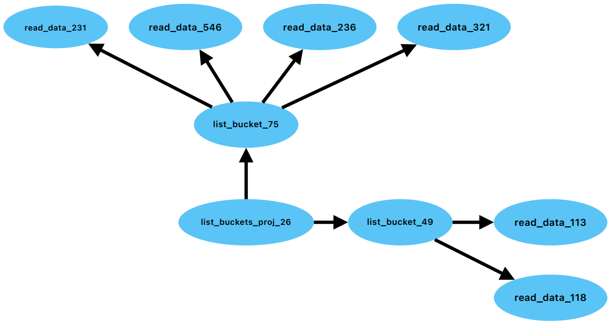

Fig. 1. The resulting attack graph.

VI. Simulation

Now, at each simulation step, events are triggered according to the specification. For instance, when the attacker compromises bucket_75.list_objects (list_bucket_75 in Figure 2), the attacker now needs to provide the employed account and IP. A list_object telemetry event is automatically produced, containing information about the concerned asset and attack step, as well as the account and the IP address. In this manner, the simulated log is generated.

VII. Graph Neural Network Training

From the attack graph and an event log (synthetic or real), at each time step, we can generate a graph to present to the GNN. Each event is connected with a trigger association to the associated attack step (as defined in the MAL specification), with an information associations to the employed account and IP, and a next association to the most recently triggered event, thus producing a graph like the one in Figure 2.

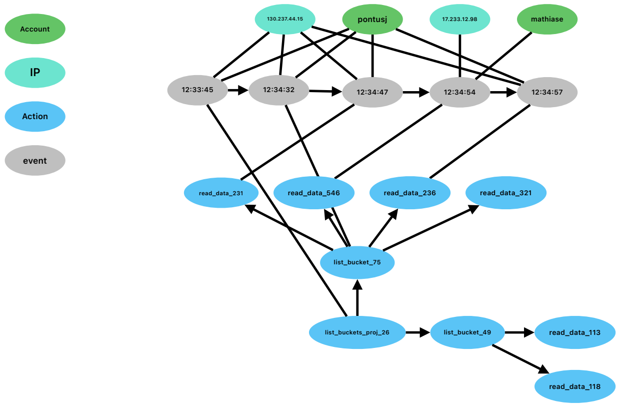

Fig. 2. Input graph for GNN training.

As mentioned, this graph should be useful to the GNN, as all the information relevant to assess the state of compromise of a node is within the neighborhood.

A. Future work

Suggest directions for future research and improvements.

VIII. Conclusion

A. Summary

Recap the main points and findings of the paper.

B. Impact

Discuss the potential impact of the proposed system on the field of intrusion detection.

C. Next steps

Outline immediate next steps for further development and research.

IX. Discussion

A. Challenges and limitations

Discuss potential challenges and limitations of the proposed approach.

- One consequence of this encoding is that the fan-out of certain nodes will become quite large.

B. Scalability

Address concerns about scalability and how they are mitigated.

- Parts of the graph where no events are recorded during the time window can therefore be omitted.