PROGRAPHER: An Anomaly Detection System based on Provenance Graph Embedding

32ND USENIX SECURITY SYMPOSIUM

Fan Yang, Jiacen Xu, Chunlin Xiong, Zhou Li, Kehuan Zhang

Advanced Persistent Threats Attacks

New APT Group Red Stinger Targets Military and Critical Infrastructure in Eastern Europe

May 11, 2023 Ravie Lakshmanan Advanced Persistent Threat

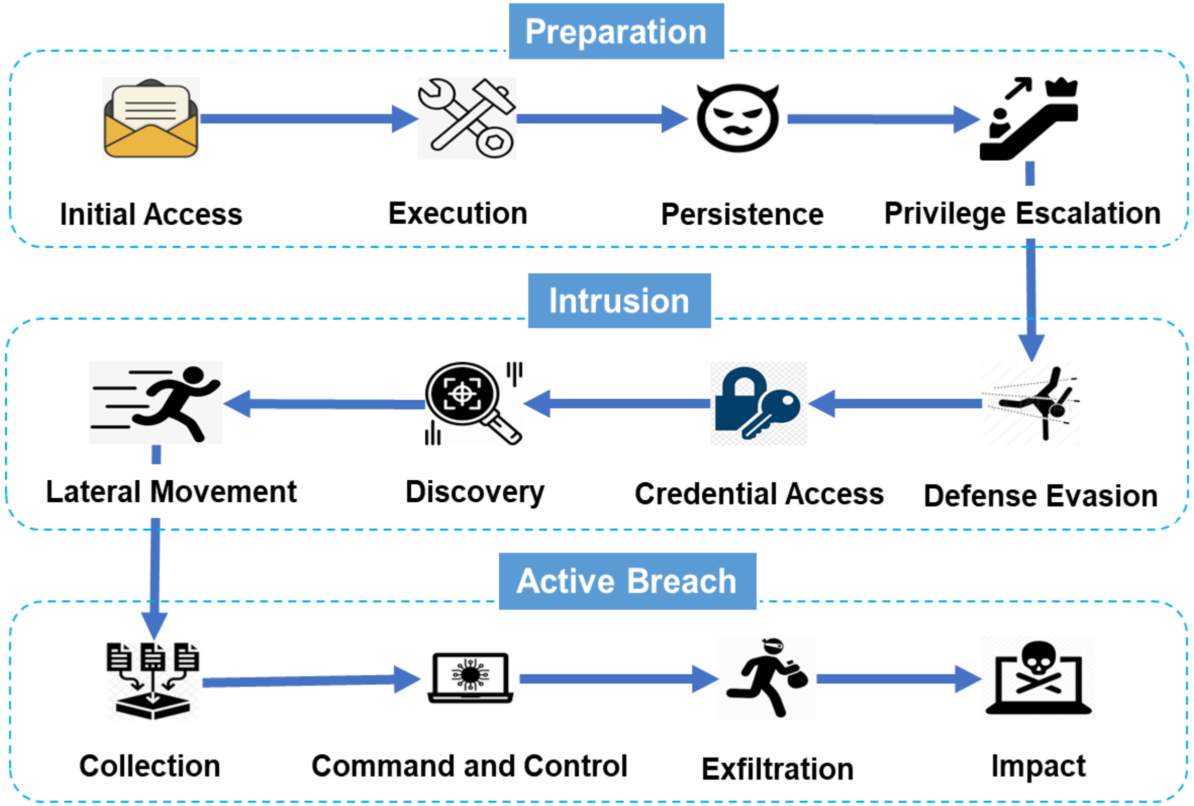
NEWS 3 NOV 2022

Threat Actor "OPERA1ER" Steals Millions from Banks and Telcos

Home > News > Security > Dark Pink hackers continue to target govt and military organizations

Dark Pink hackers continue to target govt and military organizations

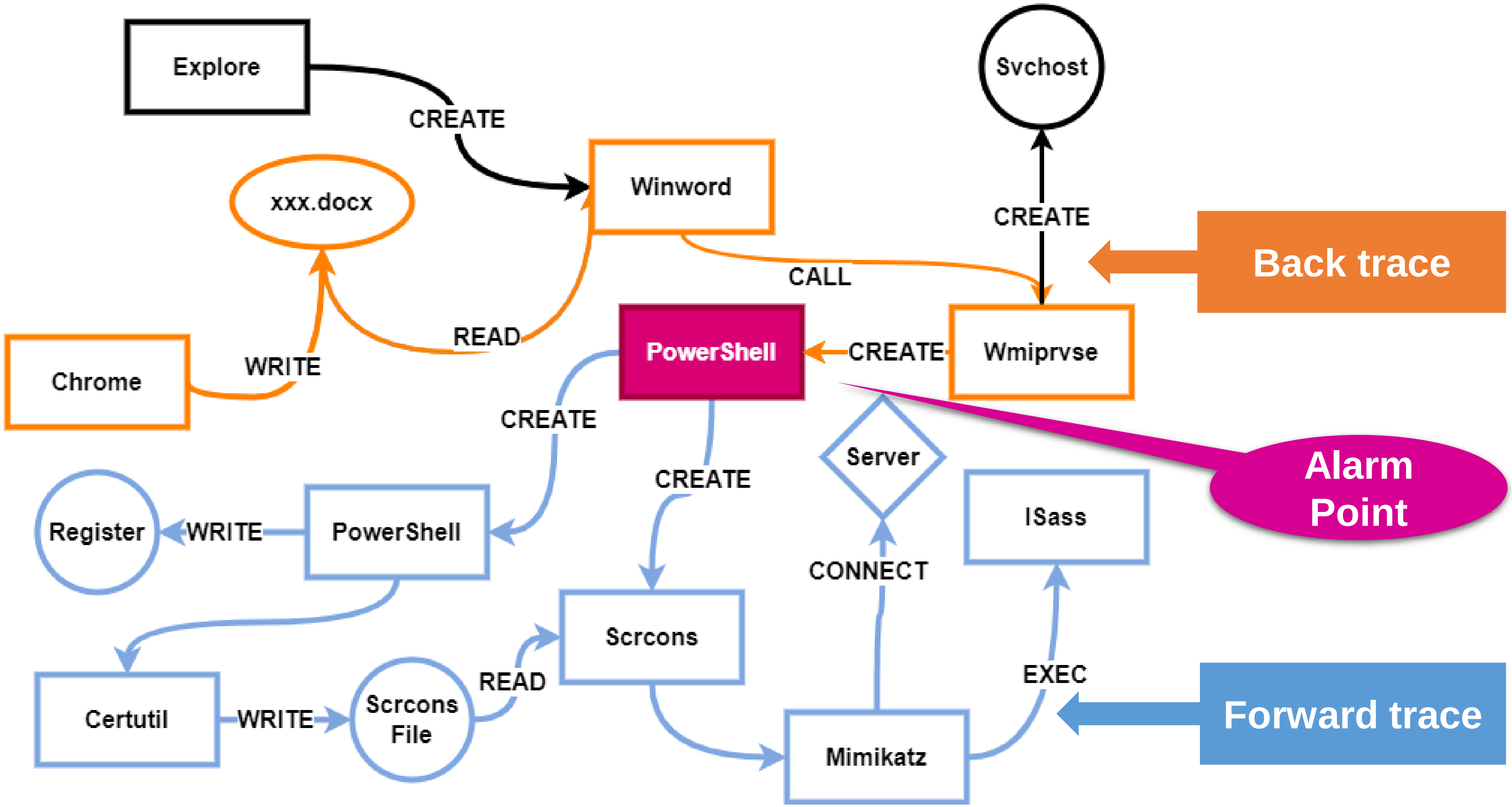
By Bill Toulas May 31, 2023



A Big Problem Affecting Many Nations and Industries

Long Duration and Stealthy

Detect APT Attacks with Provenance Graph



With data provenance, we can capture **full historical context** and all **casual relationships** among system subjects (e.g., process) and objects (e.g., files).

Previous Provenance-based Approaches

Heuristics-based provenance analysis:

- Leverage the knowledge of experts and known attack behaviors to search the attack patterns or prioritize investigations.
- **Require considerable effort** from the experts and **can be vulnerable to zero-day exploits.**

Learning-based provenance analysis:

- Apply machine learning methods to classify system entities of different granularities into benign or not.
- Show promising detection performance, but can not achieve a good **balance** between **efficiency, accuracy, and granularity.**

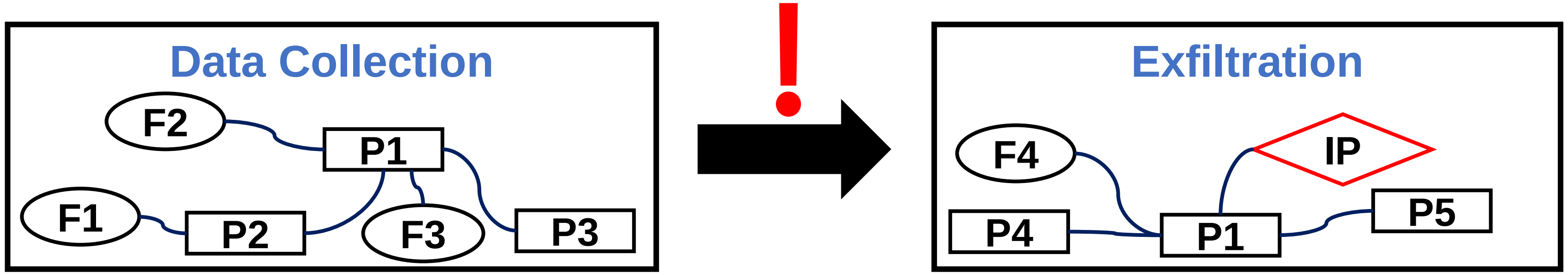
Our Motivation

Observation:

a. Each subgraph could represent one behavior of the system.

b. APT attacks can be exposed by evaluating the likelihood of a system's behavior interacting with historical behaviors.

- At least one phase of APT attacks is likely to exhibit unusual behaviors compared to normal system behaviors.



Normally, the behavior of communication with a public network **should not** happen after the local file collection

We can detect such attacks by estimating this likelihood with the causal relationships among system provenance graphs at different times.

PROGRAPHER : Goals

We envision **three design goals (G1 to G3)** to be fulfilled by **ProGrapher**:

a. It should learn the normal behavior patterns from the benign logs. → **Accuracy**
   - It should be built with **unsupervised-learning** fashion.

b. PROGRAPHER should be able to process **subgraphs of the whole provenance graph** that are separated by periods, and leverage the temporal dynamics between periods for detection. → **Efficiency**

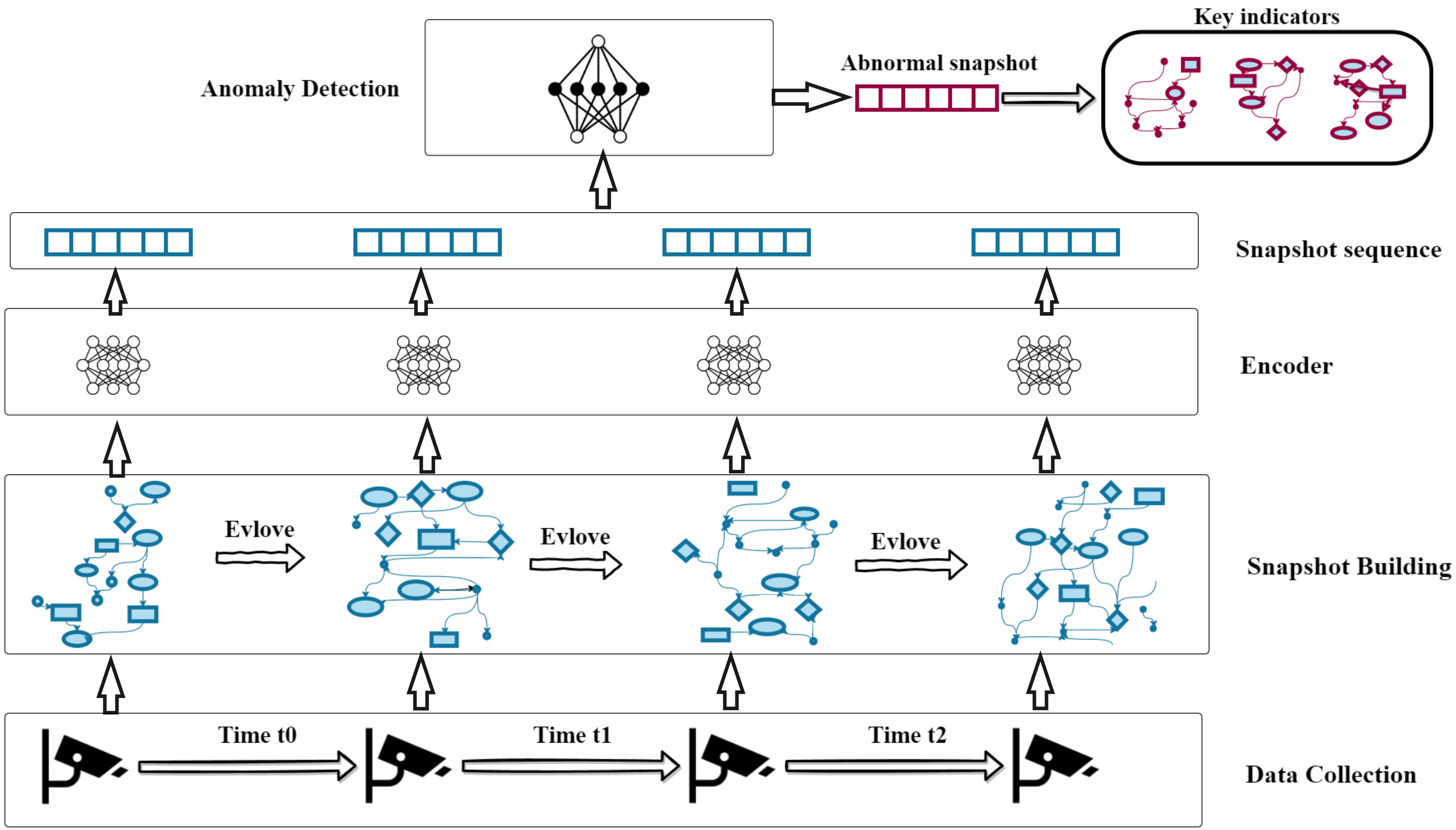
c. PROGRAPHER should be able to accurately identify the subgraphs with abnormal activities and **point out** most suspicious entities. → **Granularity**

PROGRAPHER : Overview

Input Takes as input system audit logs.

Training Minimize the prediction loss between normal behavior interaction.

Output Abnormal snapshots and directly related system entities.



Snapshot Building

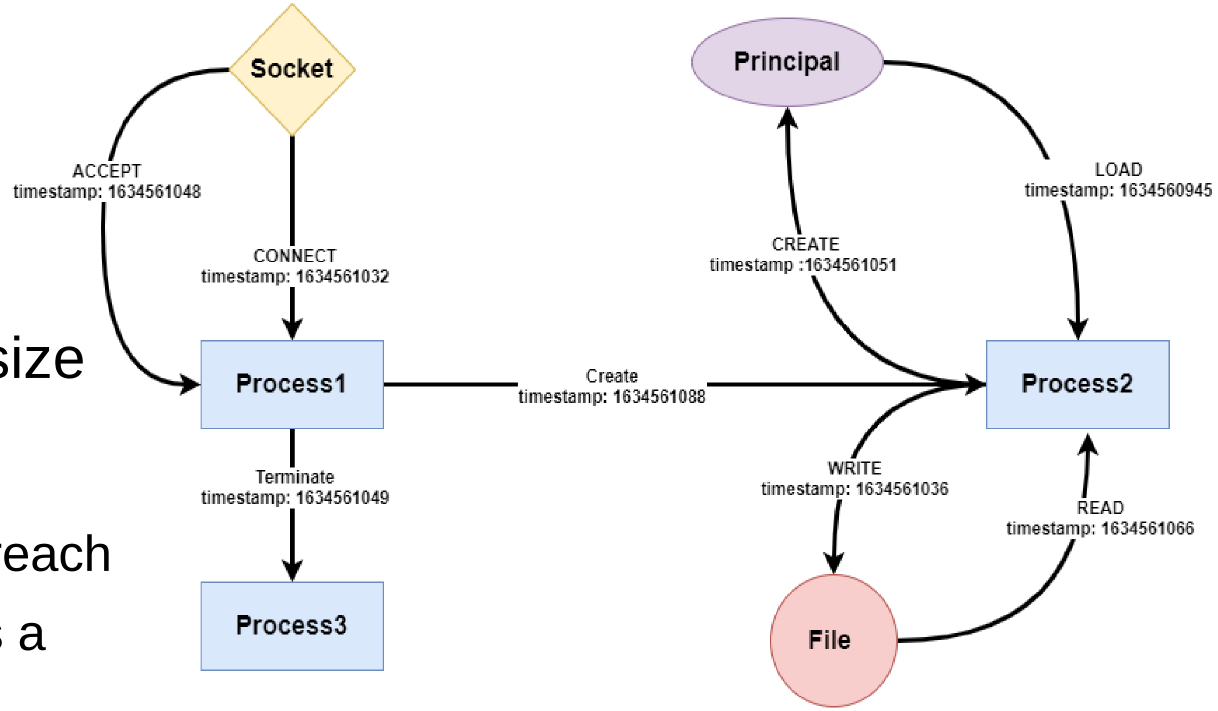
a. Maintaining a cache provenance graph

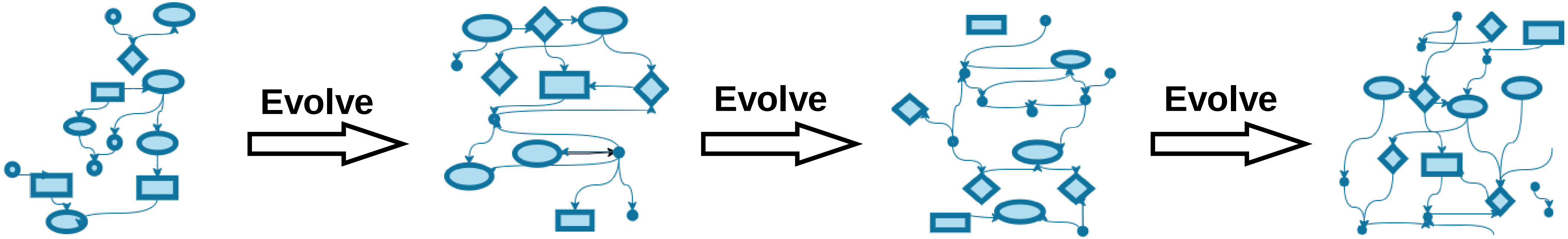
b. Vertices: system entities

c. Edges: relations between system entities

d. Sampling provenance graph every constant size

- Forgetting rate fr
- Forgetting $n * fr$ oldest nodes when graph size reach $n * (1 + fr)$ and output old provenance graph as a snapshot.

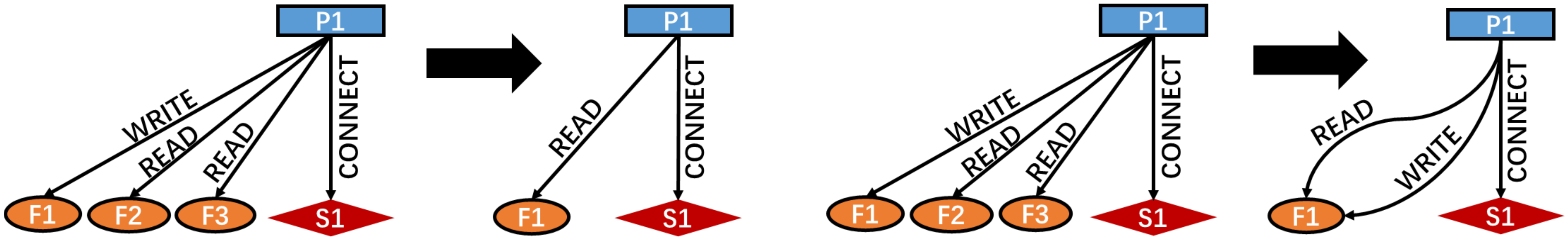




Encoder

- **Whole graph embedding representation**
  a. Convert whole graph into a latent vectors based on ***Graph2Vec*** model.
    - Extract **rooted subgraphs** from snapshot that could represent behaviors.
    - Train graph2vec to acquire representation of each snapshot.
      - Maximize the likelihood of co-occurred rooted subgraphs and corresponding snapshot.

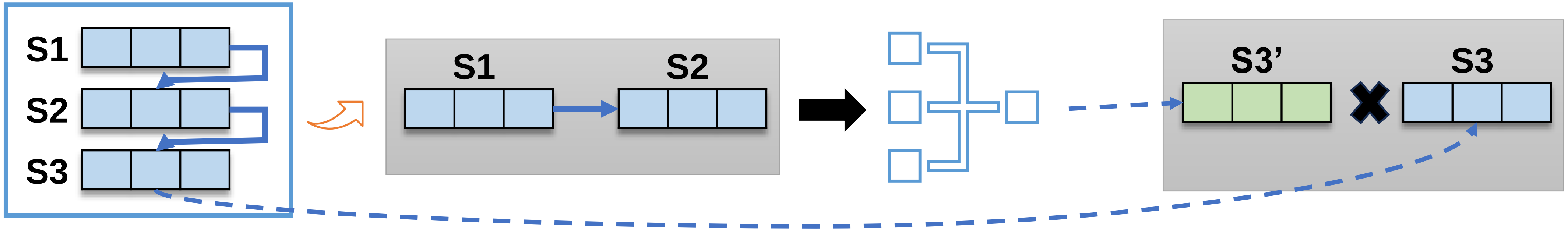
  b. Rooted subgraph optimization.
    - Faster the training speed.
    - Remove duplicate edges and nodes.



Anomaly Detection

Training:

- Given benign snapshot sequences, we apply ***TextRCNN*** model on system **snapshot embeddings** to learn the behavior interaction between each snapshot and historical snapshots happened before it.



Testing:

- During testing, according to the **predication loss and threshold**, we detect the abnormal snapshot.

Key Indicator Generation

- After getting the result from the anomaly detection model, we could further use **the ground-truth vector**, **generated prediction vector**, and **well-trained graph2vec model** to infer the **potential malicious root subgraph.**

Evaluation Datasets

- **Four simulated datasets and one real-world EDR dataset** for our evaluation.

  a. **StreamSpot**
     - The StreamSpot dataset contains 600 benign and attack graphs derived from 6 scenarios.

  b. **ATLAS**
     - Data is collected in a manually controlled environment and contains ten types of APT attacks separately

  c. **DARPA3 (CADETS, CLEARSCOPE, THEIA)**
     - Data obtained during a red-team vs blue-team adversarial engagement with various provenance capture systems.

  d. **DARPA4 (TRACE)**
     - It has more system events and entities per unit of time than the previous three datasets, reflecting the vast diversity of user behavior patterns and background activities

  e. **Real-World Deployment**
     - 9-day EDR production data collected from SANGFOR company.

Performance in APT Detection

Comparison Study:

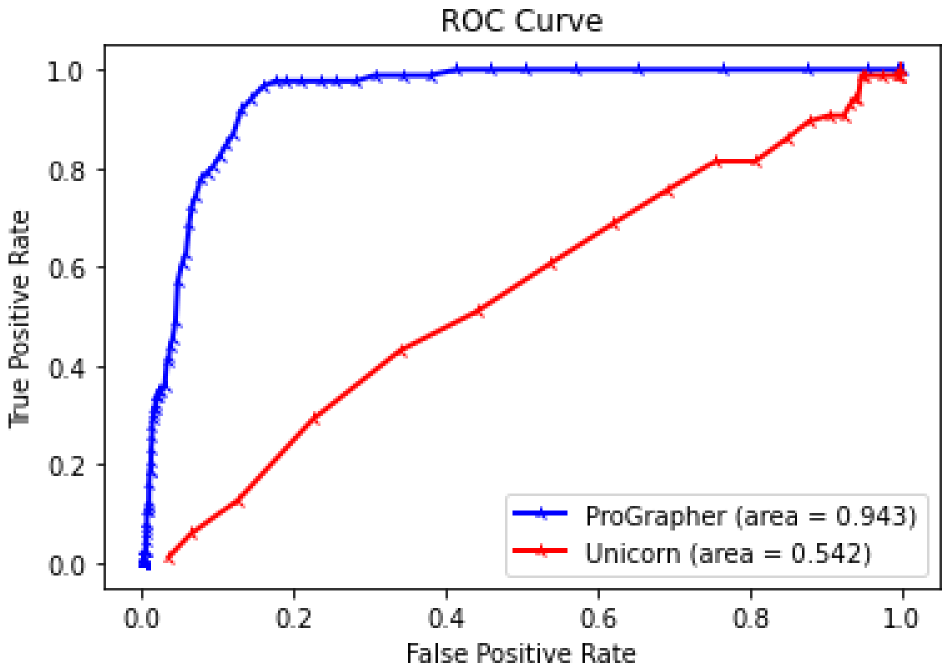
ProGrapher outperforms the state-of-the-art detection system with better F1 score and fewer false positives. It reveals that graph embedding and temporal modeling employed by ***ProGrapher*** are important to provenance analysis.

Dataset	System	Precision	Recall	Accuracy	F1
StreamSpot -DS	Unicorn	0.85	1.00	0.91	0.92
	ProGrapher	**0.90**	**1.00**	**0.94**	**0.94**
CADET	Unicorn	0.31	1.00	0.44	0.47
	ProGrapher	**1.00**	**1.00**	**1.00**	**1.00**
CLEARSCOPE	Unicorn	1.00	0.75	0.93	0.86
	ProGrapher	**0.80**	**1.00**	**0.93**	**0.89**
THEIA	Unicorn	0.67	0.67	0.80	0.67
	ProGrapher	**1.00**	**1.00**	**1.00**	**1.00**

Performance in APT Detection (cont.)

Real-World EDR dataset:

Even in a production environment, **PROGRAPHER** can achieve reasonable accuracy e.g., **94% TPR** and **14% FPR.**



Other Datasets:

PROGRAPHER is able to detect different APT attacks with various situations.

Dataset	Precision	Recall	Accuracy	F1
ATLAS	1.0	1.0	1.0	1.0
DARPA4	1.0	1.0	1.0	1.0

Effectiveness in key indicators

- Select **top K RSGs** from an abnormal snapshot and return **all nodes** matching these RSGs as the indicators and evaluate their effectiveness based on three metrics.

With an appropriate value of K, by providing nodes that matches top K RSGs, *PROGRAPHER* can:

- Provide effective indicators for analyst to reason about the attack trace.
- Cover most of the nodes the are directly related to attacks.
- Significantly reduce the workload of security analysts.

Effectiveness of indicators

Dataset	Effectiveness Rate				
	K=1	K=2	K=3	K=4	K=5
CADETS	0.88	0.94	0.94	1	1
THEIA	0.89	1	1	1	1
CLEARSCOPE	1	1	1	1	1

Effectiveness in key indicators

- Select **top K RSGs** from an abnormal snapshot and return **all nodes** matching these RSGs as the indicators and evaluate their effectiveness based on three metrics.

With an appropriate value of K, by providing nodes that matches top K RSGs, *PROGRAPHER* can:

- Provide effective indicators for analyst to reason about the attack trace.
- Cover most of the nodes the are directly related to attacks.
- Significantly reduce the workload of security analysts.

Coverage of attack

Dataset	Coverage Rate					
	Total	K=1	K=2	K=3	K=4	K=5
CADETS	28	0.61	0.67	0.85	0.96	0.96
THEIA	18	1	1	1	1	1
CLEARSCOPE	28	1	1	1	1	1

Effectiveness in key indicators

- Select **top K RSGs** from an abnormal snapshot and return **all nodes** matching these RSGs as the indicators and evaluate their effectiveness based on three metrics.

With an appropriate value of K, by providing nodes that matches top K RSGs, *PROGRAPHER* can:

- Provide effective indicators for analyst to reason about the attack trace.
- Cover most of the nodes the are directly related to attacks.
- Significantly reduce the workload of security analysts.

Workload Reduction (K = 4)

Dataset	Covered	Total	Reduction	Unicorn
CADETS	6794	16200	58.1%	51,029
CLEARSCOPE	3460	7500	53.9%	21,853
THEIA	6988	17100	59.2%	51,147
Average	5748	13600	57.7%	41343

Other Experiments

- **Runtime Performance**
  - Measure the runtime overhead of each component.
  - Discuss how ProGrapher scales with the input volume.

- **Impact of Key Parameters**
  - Analyze the impact of key parameters on the effectiveness of ProGrapher using StreamSpot-DS, including *Snapshot size(n), Forgetting rate (fr), Snapshot sequence length (L).*

- **Robustness**
  - Conduct a new experiment on ATLAS-DS by inserting many random events before and after the attack events in order to hide the real attack.

Conclusion

■ **We present PROGRAPHER:**

- **Anomaly**-based detection system
- A novel combination of **graph embedding**, **sequence learning**, and **indicator extraction techniques** to model normal behavior patterns for accurate and **unsupervised** anomaly detection at graph level.
- It is able to achieve **high accuracy** in finding abnormal snapshots and significantly reduce analysts' workload in **pinpointing the root cause** of the anomalies.

ProGrapher: An Anomaly Detection System based on Provenance Graph Embedding

32ND USENIX

Security Symposium

Thank you for your time and attention!

yf020@cuhk.edu.hk