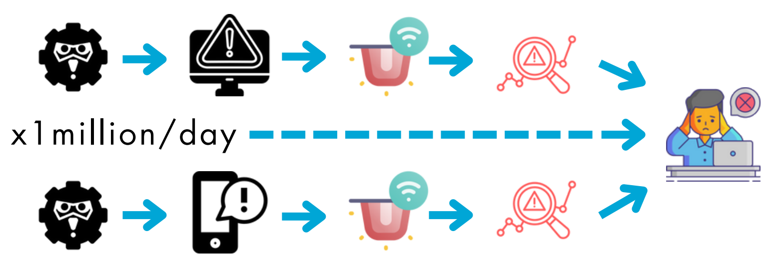
INVESTIGATING THE IMPACT OF SINK STATE MERGING ON ALERT-DRIVEN ATTACK GRAPHS

The effects of allowing the sink states to merge with other sink states

Author: Alexandru Dumitriu
a.dumitriu-1@student.tudelft.nl
Responsible Professor: Sicco Verwer
Supervisor: Azqa Nadeem
Delft University of Technology

1. Background

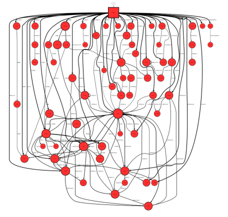
Alert fatigue due to too many alerts



SAGE compresses these alerts into Attack Graphs

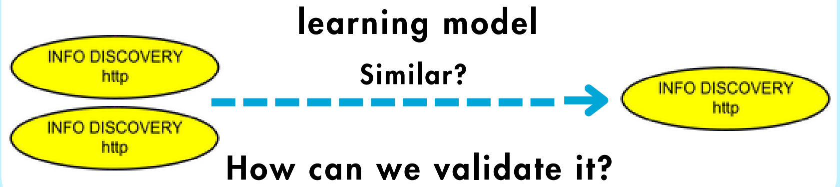
SAGE captures infrequent behaviour

- Learns with an S-PDFA model.
- Does not include infrequent severe states.
- Those are stored in sink states and are added afterwards.



2. Problem

SAGE uses an interpretable unsupervised sequence learning model



Learn a new and different model

Try allowing merges between sink states

Why...

- Sink states are not used in the learning process.
- Merged sink states can increase the quality of Attack Graphs.
- The algorithm might do incorrect merges between sinks due to their infrequent nature.

3. Methodology

Literature Study

Experiment Setup

→

Metrics

Size: # nodes

Compelxity:
- #nodes/#edges
- Linear regression

Interpretability:
- Global & Local Density
- Readability Protocol
- Relative Context Loss

Completeness: alerts represented in Attack Graph

→

Experiment
- 2 variants of SAGE
- datasets: CPTC-2017, 2018

→

Analysis
- Stats on Size, Complexity & Completeness
- H2H Analysis - Interpretability

Proposed Merging Criterion

4. Results

Size

22%
- 41 AGs suffered changes to the node count

4%
- overall number of nodes did not decrease substantially

Complexity

- 5 AGs transitioned from simple to complex
- overall decrease of less than 1%

Completeness

- not affected because alerts are not altered
- absolute value of around 80%, due to discarding of episodes with len<3

80%

Interpretability

- 25 pairs of AGs analysed Head to Head
- Consistent results after merging sinks:
  - Protocol takes longer to complete
  - Global Density Higher
  - Local Density Lower
  - Loss of Context

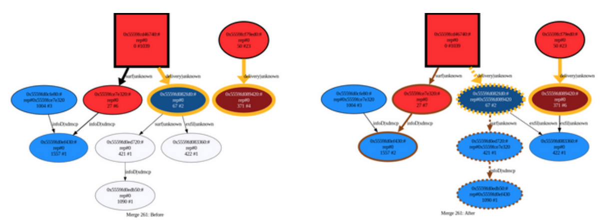


Figure 1: FlexFringe representation of the AG before and after the merge between two DATA DELIVERY unknown nodes

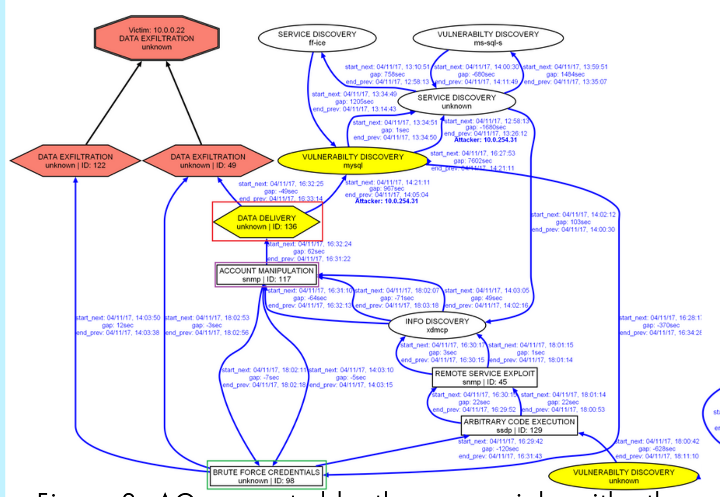


Figure 2: AG generated by the merge sinks with other sinks implementation, important merges highlighted

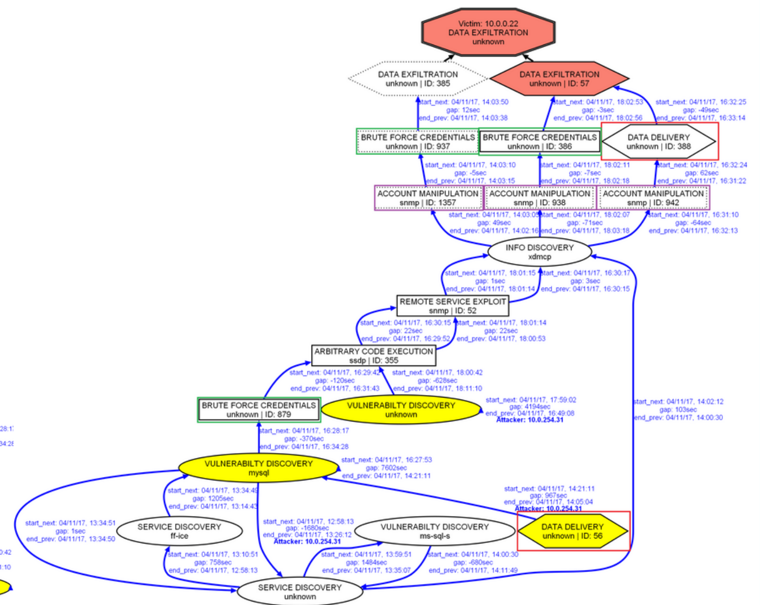


Figure 3: AG generated by the baseline implementation, unmerged sinks highlighted

Proposed Merging Criterion

Loss of context
↓ Why
Merges at diff levels
↓ Then
Add constraint:
Only merge if at equal distance to the victim

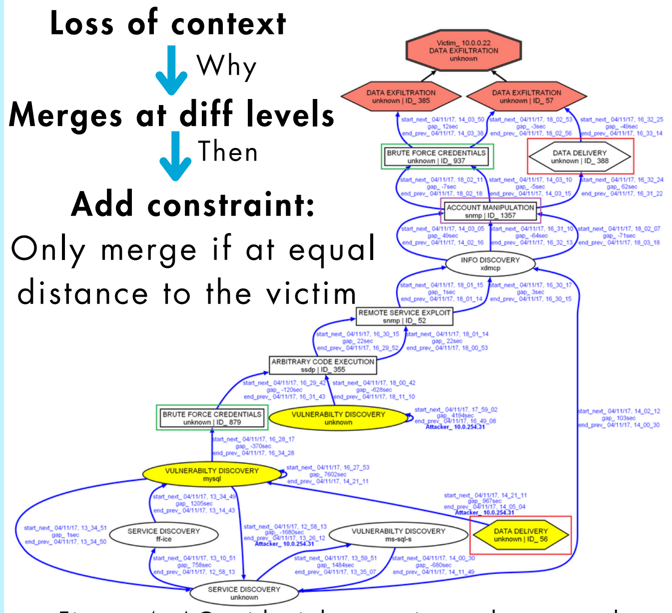


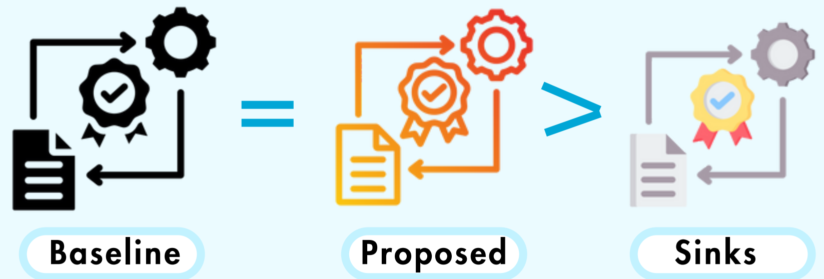
Figure 4: AG with sinks merging only at equal distance to the victim

	Baseline	Merge Sinks	Eq Dist
Global Density	0.081	0.153	0.116
Local Density	0.494	0.421	0.505

Table 1: Density differences between the 3 variants

5. Conclusions

- All **sink states** transformed into normal states.
- A small overall deficit in the average **size** of attack graphs.
- Baseline implementation is consistently less or equally **complex** to the merge sinks implementation.
- **Interpretability** has decreased substantially in all AGs affected by the merging of sink states.
- **Completeness** remained consistent at a level of approximately 80% because the extra merges happening are not affecting the episodes' processing.



Limitations:
- Manual analysis of alerts - error-prone
- Qualitative analysis of AGs - bias

Future Work:
- Add a small delta to the constraint
- Consider constraints based on the start state