

Explainable Sequential ML for Cybersecurity: A case of attacker strategy discovery



Azqa Nadeem^{1Φ} Sicco Verwer¹ Stephen Moskal² Shanchieh Jay Yang²
¹ Delft University of Technology ² Rochester Institute of Technology ^Φ azqa.nadeem@tudelft.nl



Background

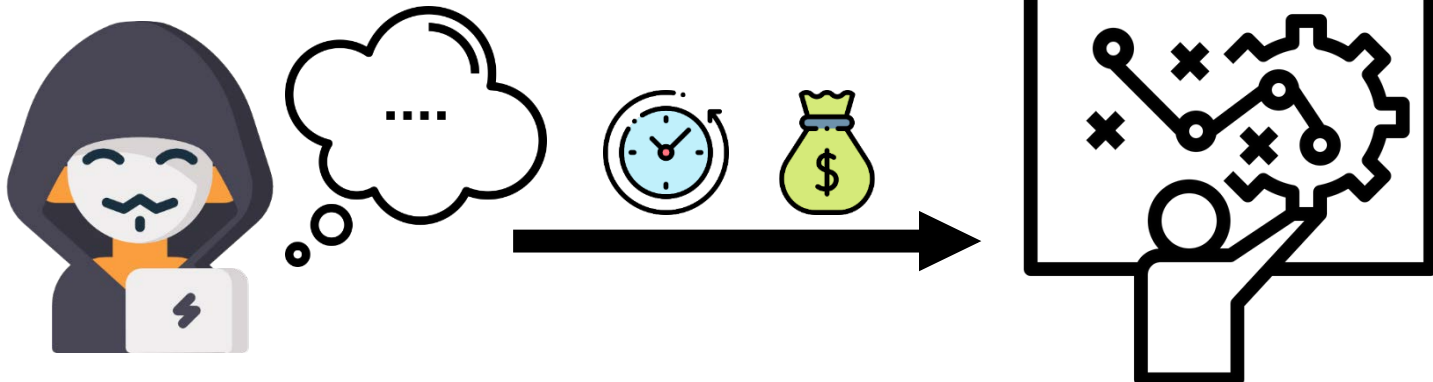
How to discover & display attacker strategies from intrusion alerts?

Too many alerts → alert fatigue

1 million alerts/day!



Attacker strategy identification is manual & labor-intensive

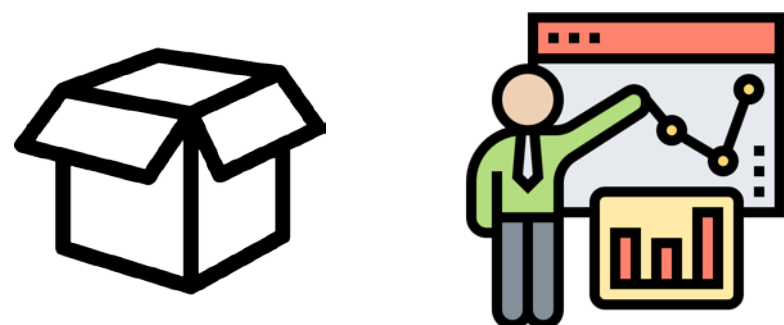


Want to answer questions like:

- How did an attack happen?
- Were multiple attackers involved?
- Were their strategies similar?

Design challenges

Need an explainable approach



Severe alerts are rare; non-severe are frequent but also interesting



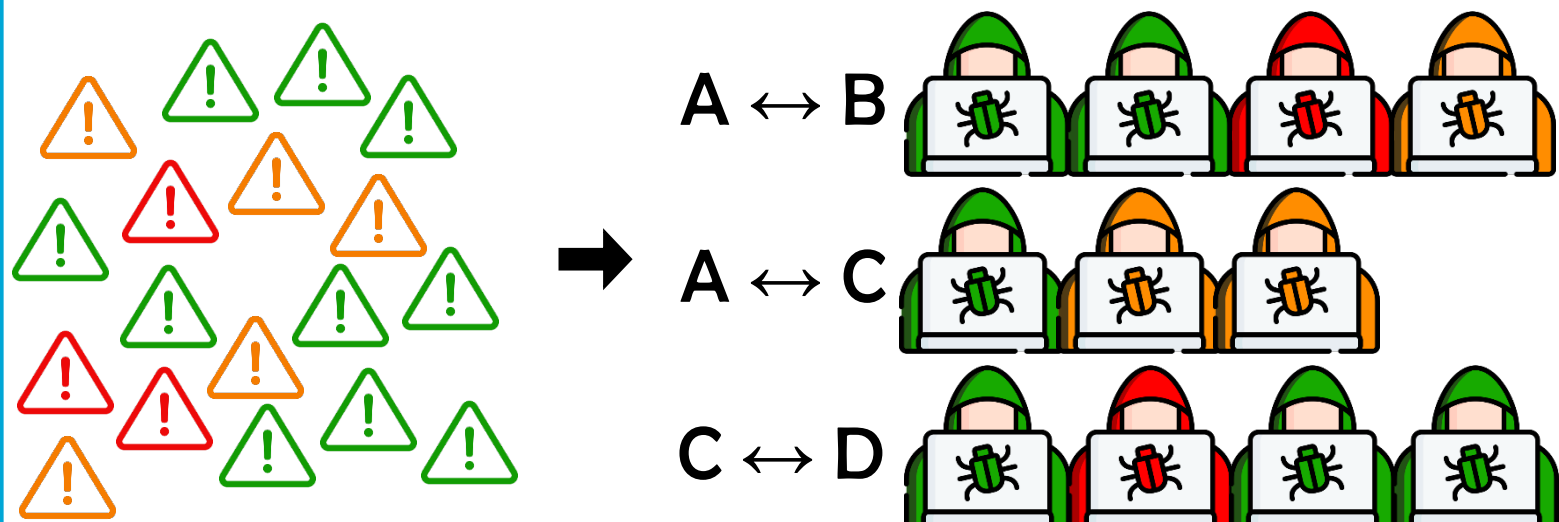
Same alert can have different semantics, depending on when it happened



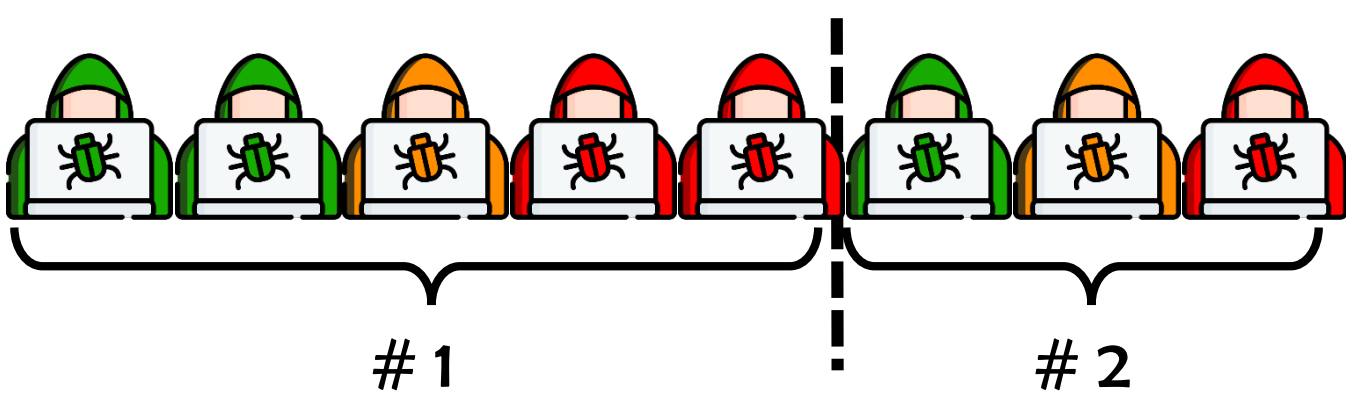
Proposed method

SAGE discovers attacker strategies from alerts, by learning a suffix probabilistic-DFA.

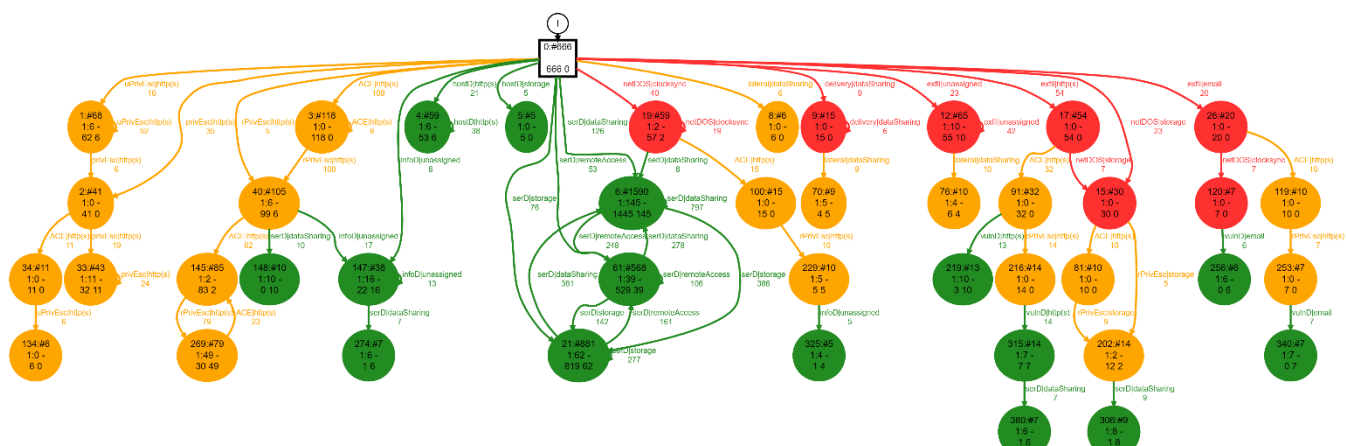
Alerts to Action sequences



Split each attack attempt

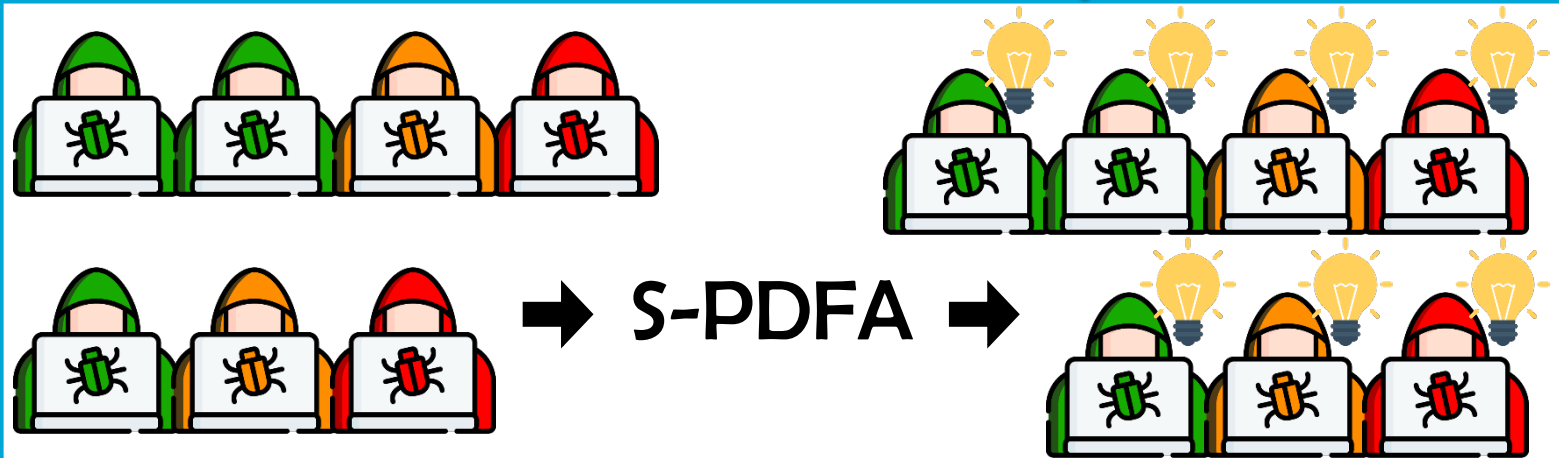


Suffix Probabilistic-DFA learning

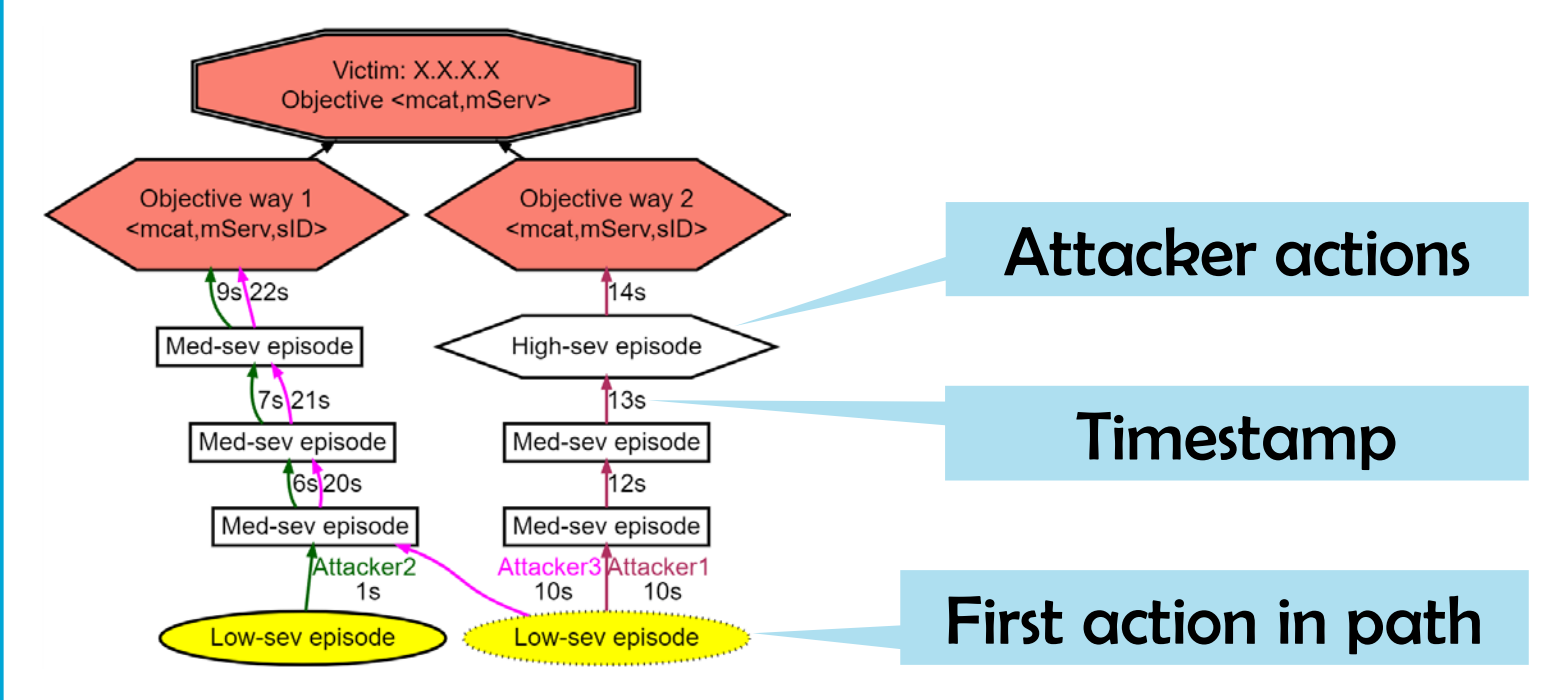


1. Highly interpretable
2. Highlights infrequent actions
3. Models semantics

Add semantics to sequences



Attack graph (AG) construction



Key Results

On CPTC-2018, containing

300,270 alerts

SAGE generates

93 AGs

in

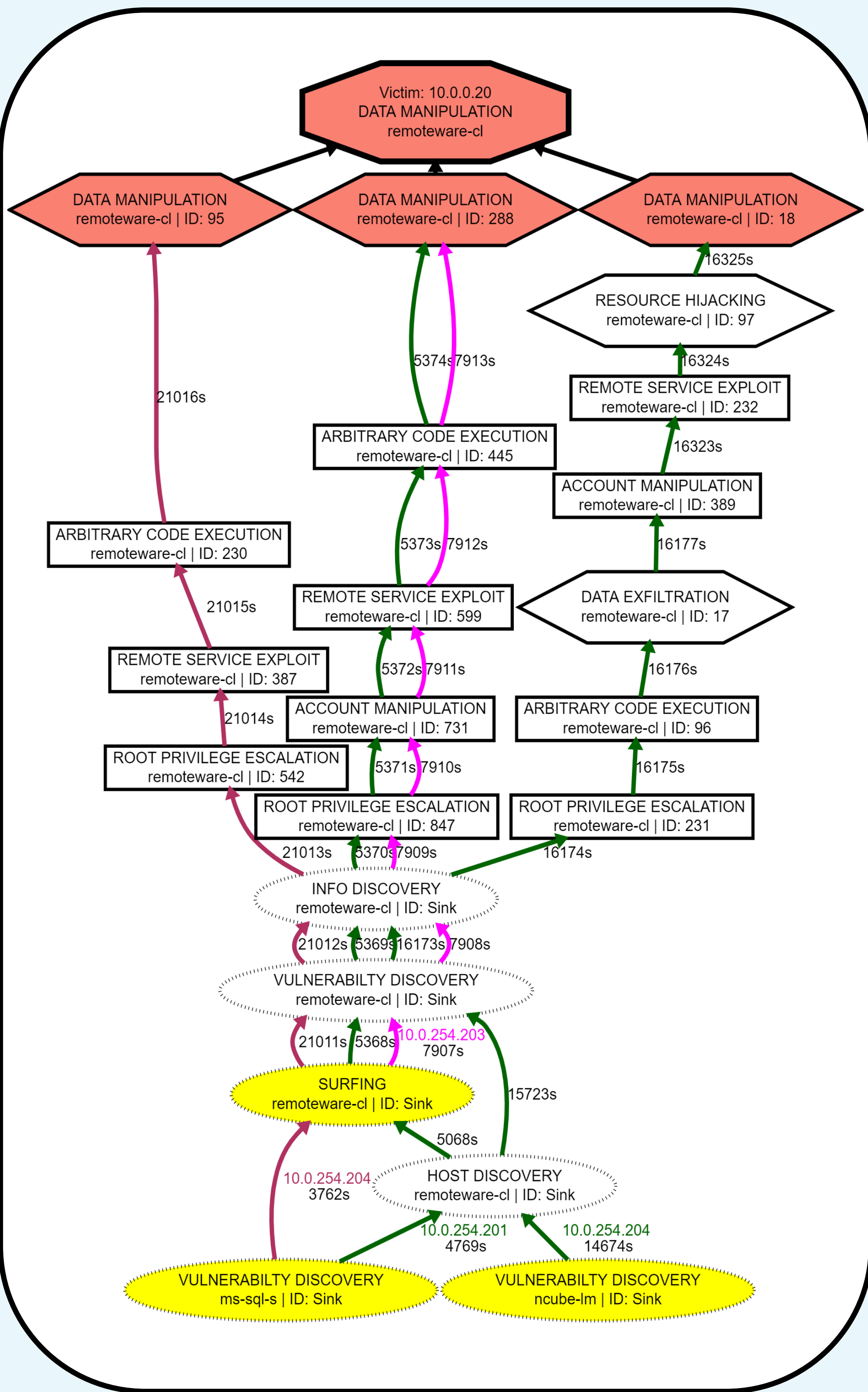
< 1 min

AGs show how an attack transpires

AGs show concrete similarities between attacker strategies

S-PDFA discovers 3 ways to reach the objective

SAGE finds 29 fingerprintable paths for attacker re-identification



Takeaways

SAGE extracts AGs without expert input!

S-PDFA is critical in modeling semantics & highlighting infrequent patterns.

SAGE is interpretable & transparent, enhancing analysts' productivity.



SAGE is open-source!

Code: <https://github.com/tudelft-cda-lab/SAGE>

Paper: <https://arxiv.org/abs/2107.02783?context=cs.LG>