SAGE: Intrusion Alert-driven Attack Graph Extractor

Azqa Nadeem[*], Sicco Verwer[*], Shanchieh Jay Yang[^]

[*]Delft University of Technology, The Netherlands
[^]Rochester Institute of Technology, USA

IEEE Symposium on Visualization for Cyber Security (VizSec)

Cyber Analytics Lab
TU Delft

RIT

27 October 2021

Background

- Attacker strategy identification requires manual effort
  - How?
  - Multiple attackers?
  - Strategic similarity?

- Answers via cybersec data + expert input

RIT
TUDelft

* https://www.imperva.com/blog/27-percent-of-it-professionals-receive-more-than-1-million-security-alerts-daily/
https://www.researchgate.net/publication/220202986_Measuring_Security_Risk_of_Networks_Using_Attack_Graphs

Background

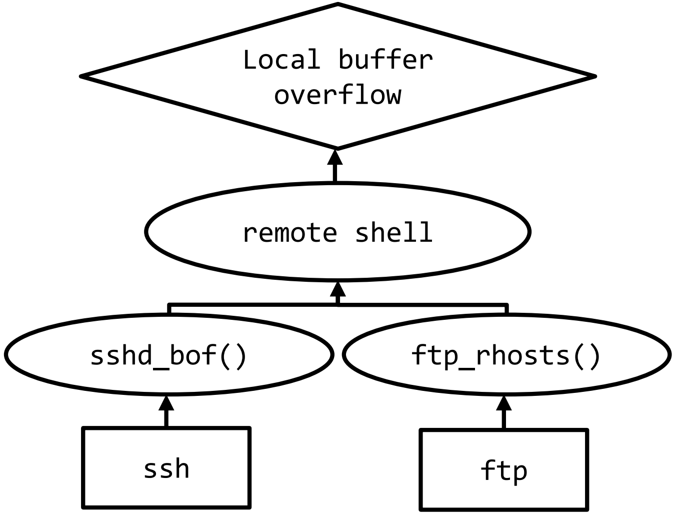
- Security analysts receive > 1M intrusion alerts/day*


```
{   '_sourcetype': 'suricata:alert',
    'alert': {   'category': 'Attempted Information Leak',
                 'severity': 2,
                 'signature': 'ET POLICY Python-urllib\\/ '
                              'Suspicious User Agent'},
    'dest_ip': '169.254.169.254',
    'dest_port': 80,
    'src_ip': '10.0.0.20',
    'src_port': 56952,
    'timestamp': '2018-11-03T13:51:58.205548+0000'}}
```


* *https://www.imperva.com/blog/27-percent-of-it-professionals-receive-more-than-1-million-security-alerts-daily/*

Background

- Automate attacker strategy identification
- via *Alert-driven* Attack Graphs



"Measuring Security Risk of Networks Using Attack Graphs". Steven Noel et al. (Int. J. Next Gener. Comput. 2010)

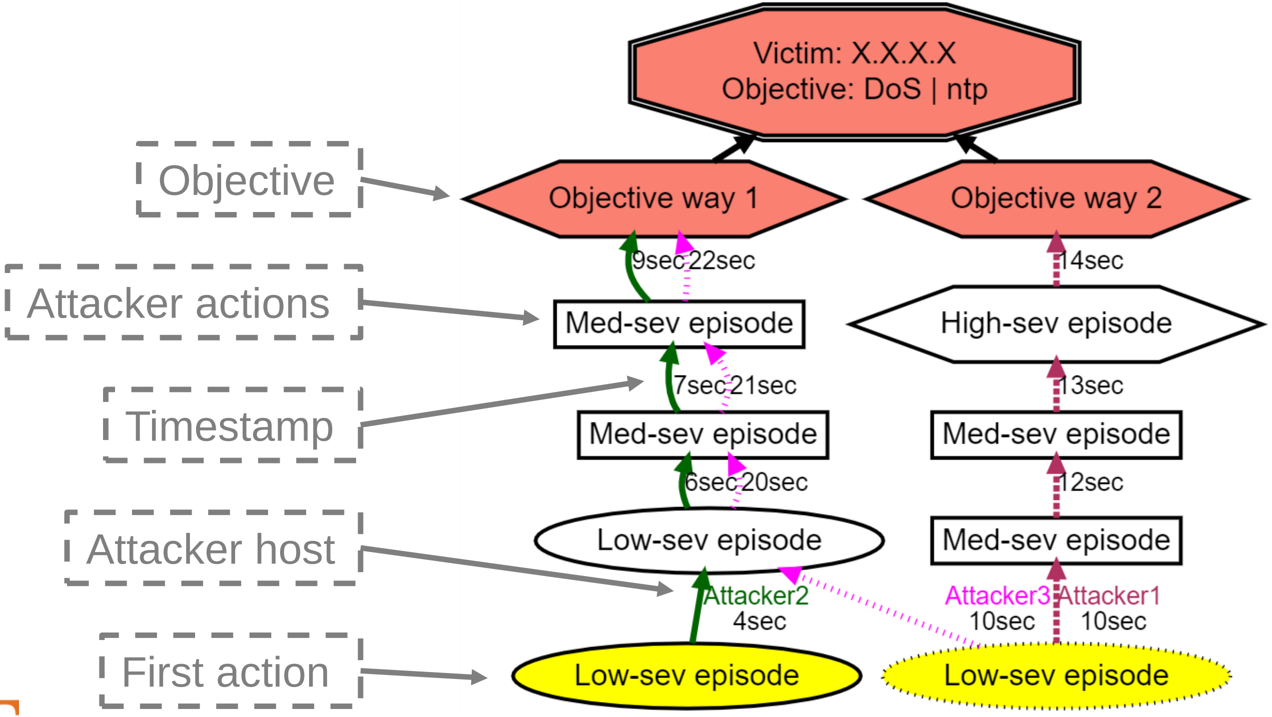
Traditional approaches

- Topological Vulnerability Analysis (TVA)
  - Network topology + Vulnerability reports
  - `MulVal` by Ou *et al.* (USENIX '05)

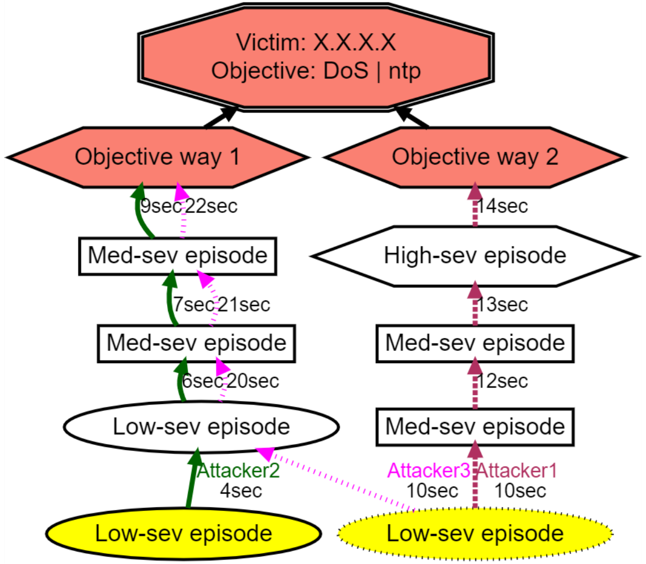
- Alert-driven attack scenario modelling
  - `Causal analysis` by Ning *et al.* (CCS '02)
  - `Visual summary` by De Alvarenga *et al.* (Computers & Security '18)
  - `Strategy discovery` by Moskal *et al.* (ISI '18)

RIT
TUDelft

Anatomy of an Alert-driven Attack Graph

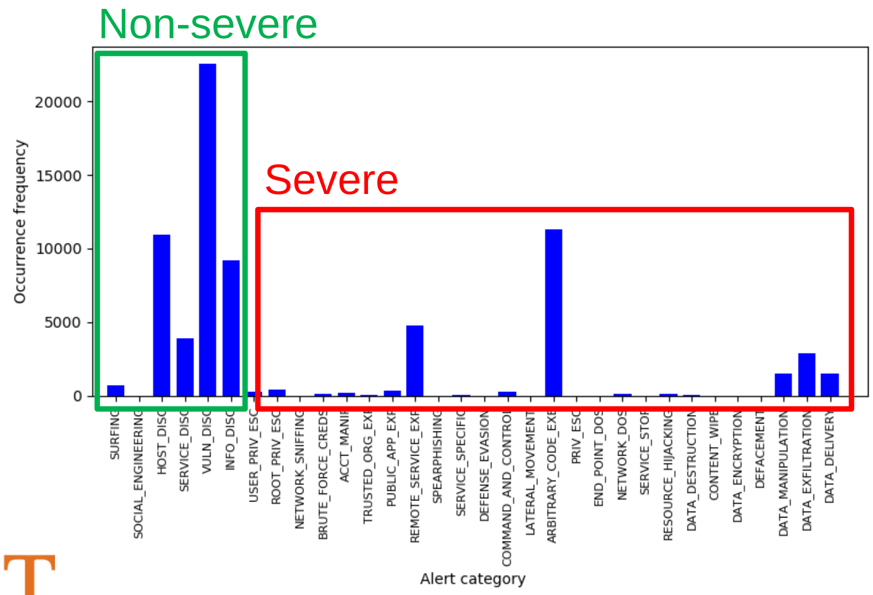


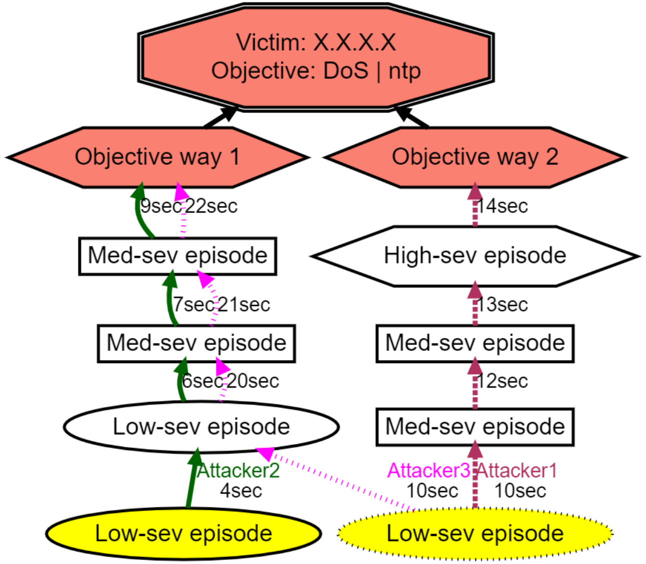
Key design challenges



Key design challenges

1. Alert-type imbalance

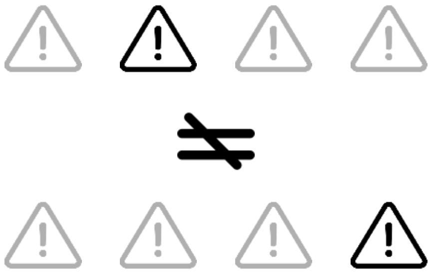


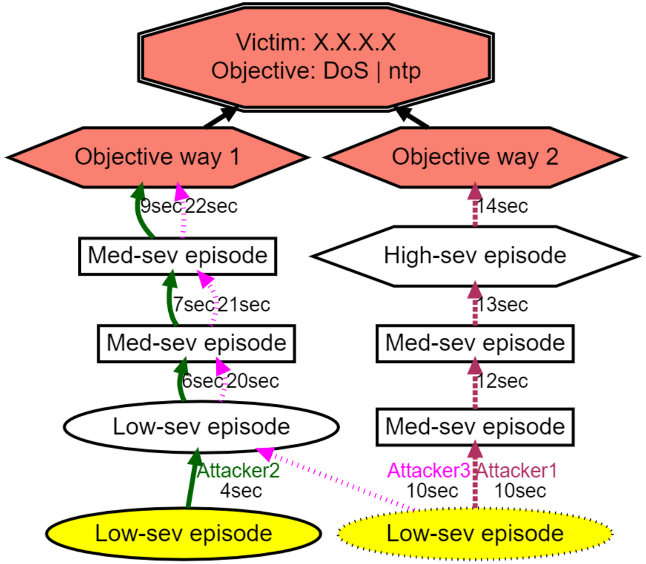


RIT
TU Delft

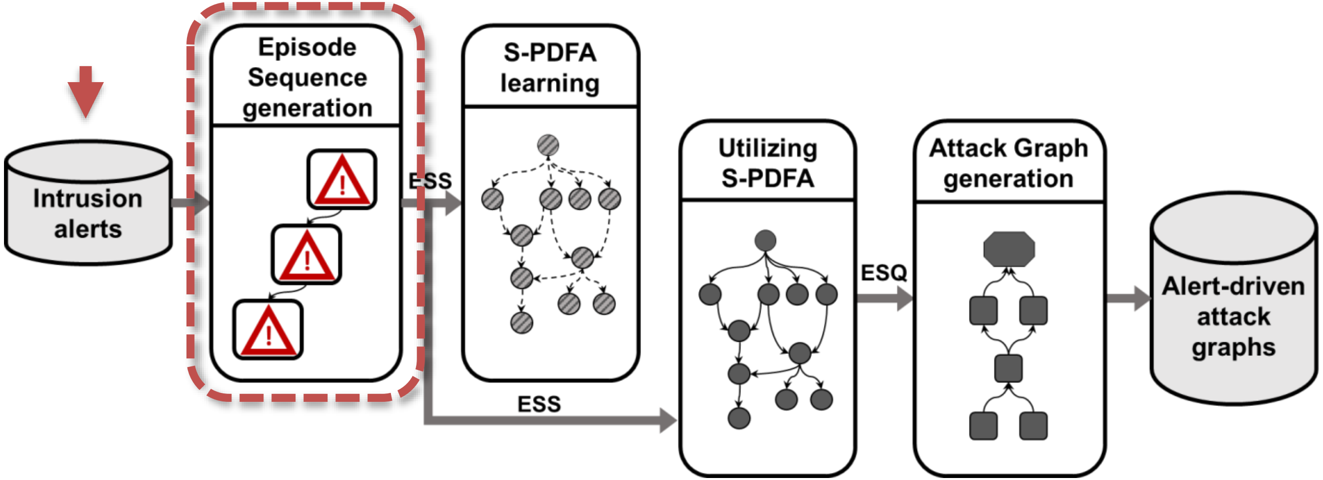
Key design challenges

1. Alert-type imbalance
2. Context modelling



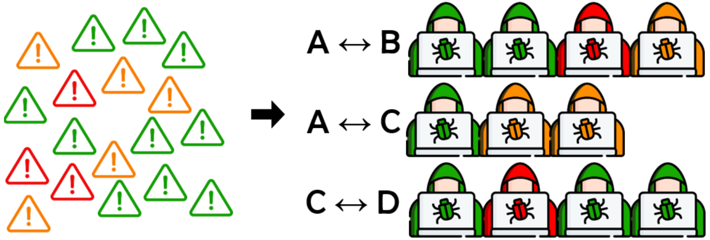


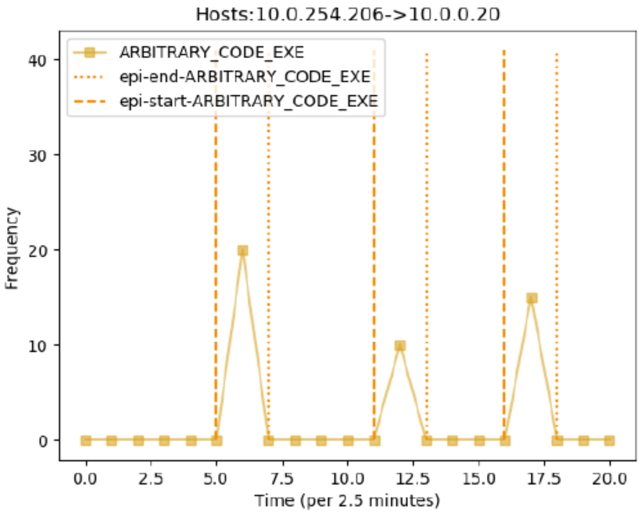
SAGE: IntruSion alert-driven Attack Graph Extractor

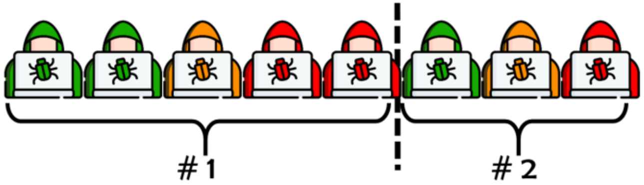


Alert → Episode sequences

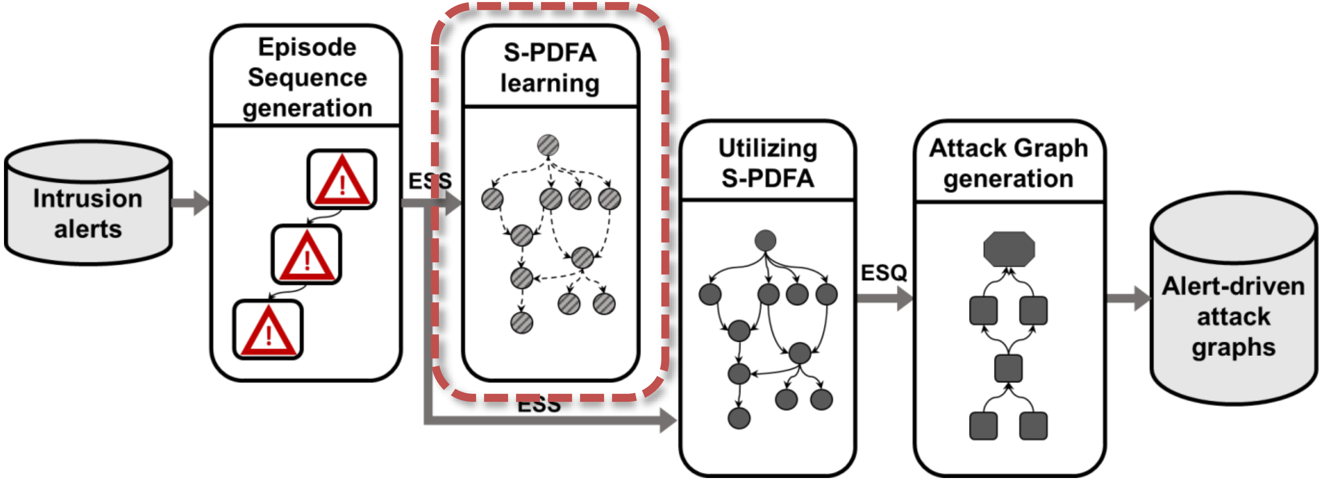

```
{   '_sourcetype': 'suricata:alert',
    'alert': {   'category': 'Attempted Information Leak',
                 'severity': 2,
                 'signature': 'ET POLICY Python-urllib\\/
                              'Suspicious User Agent'},
    'dest_ip': '169.254.169.254',
    'dest_port': 80,
    'src_ip': '10.0.0.20',
    'src_port': 56952,
    'timestamp': '2018-11-03T13:51:58.205548+0000'}}
```





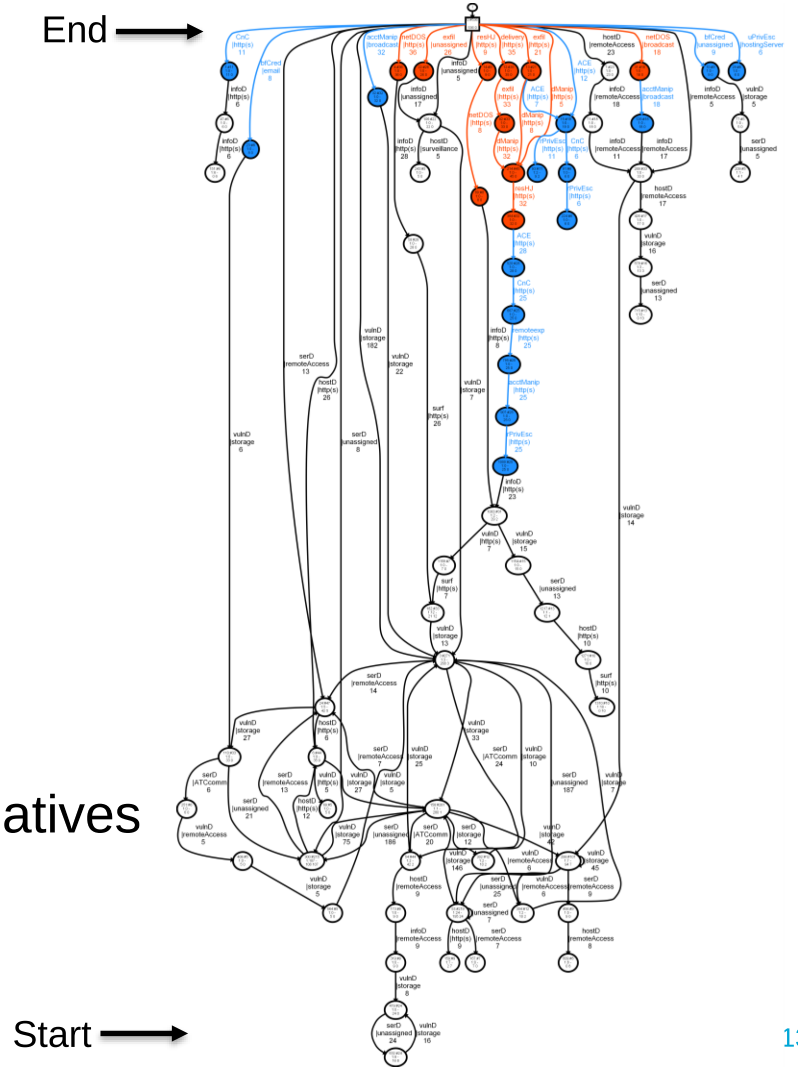
SAGE: IntruSion alert-driven Attack Graph Extractor



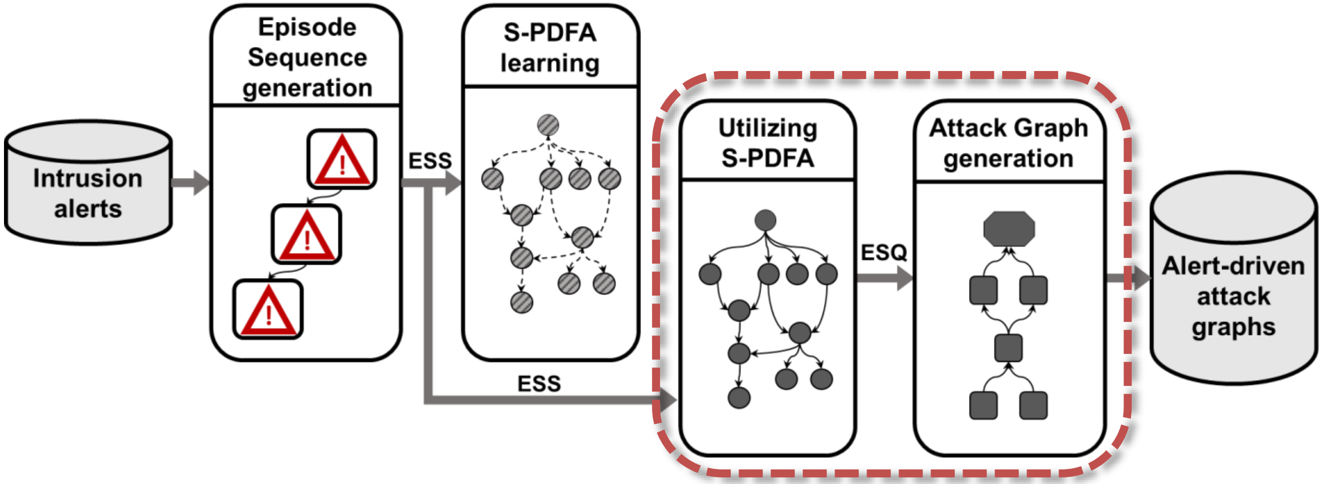
“Flexfringe: a passive automaton learning package”. Sicco Verwer, Christian. A. Hammerschmidt (ICSME 2017)

Suffix-based PDFA

- Summarizes attack paths
- Brings infrequent episodes to the top
  - Red → Severe | Blue → Medium severity
- States → milestones with context
- Good model quality compared to alternatives
  - via `Perplexity`

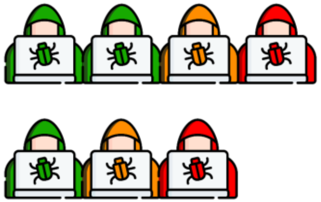


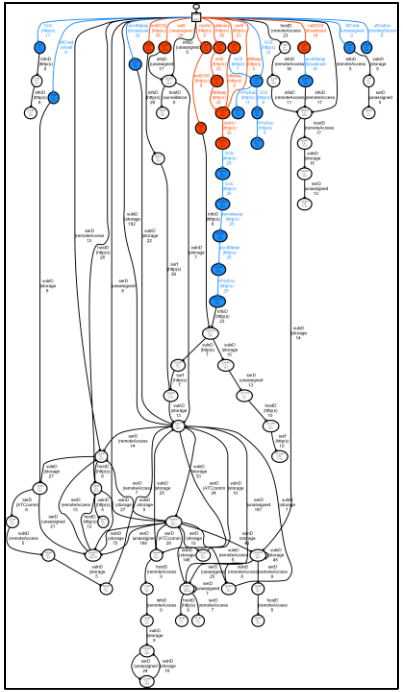
SAGE: IntruSion alert-driven Attack Graph Extractor



Adding context & AG formation

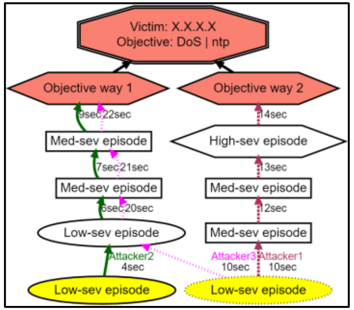
Episode sequences





State sequences

Vertex: Group of alerts
Edge: Temporal order

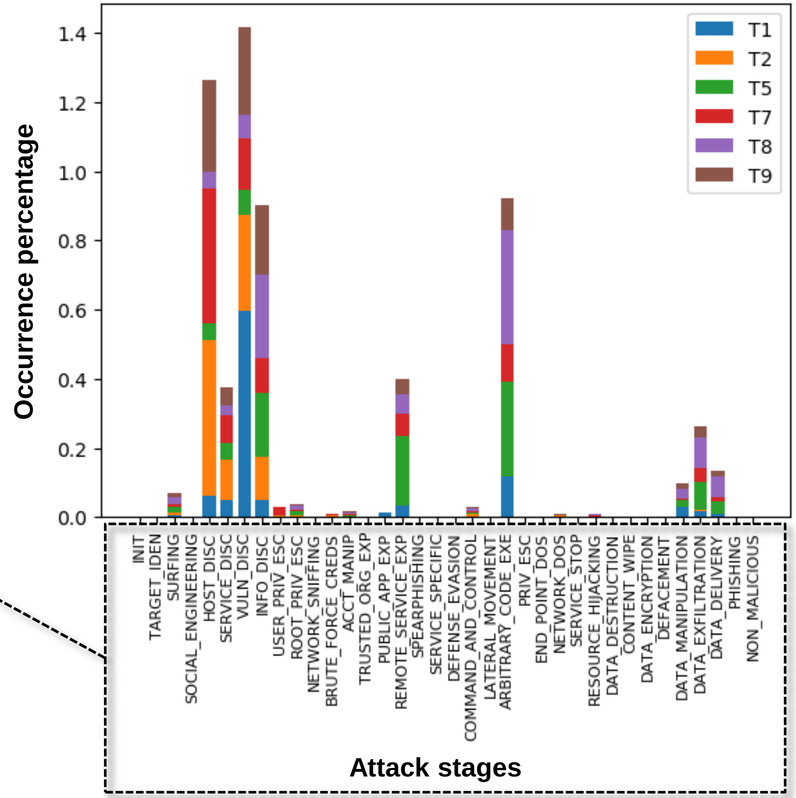


On a *per-victim, per-objective* basis

Experimental dataset

- Suricata alerts from Collegiate Penetration Testing Competition[1]
  - 6 multi-attacker teams
  - 1 fictitious network
  - 330,270 alerts

- Moskal's Action-Intent framework[2]
  - Alert signature → Attack stage
  - Based on MITRE ATT&CK



RIT TU Delft

1. CPTC dataset: https://www.globalcptc.org/
2. S. Moskal and S. J. Yang, "Framework to describe intentions of a cyber attack action," arXiv preprint arXiv:2002.07838, 2020.

[1] Alert triaging

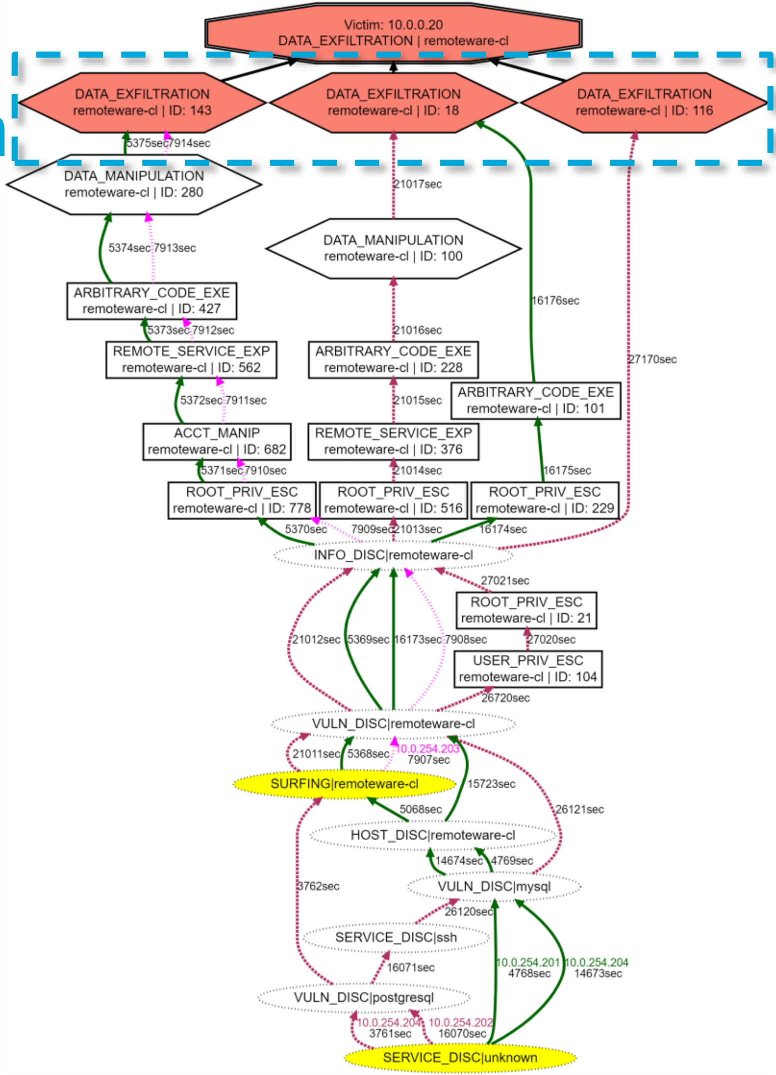
- 330,270 alerts → 93 alert-driven AGs
- ~500 alerts in < 25 vertices
- Average simplicity = 0.81

	# alerts (raw)	# alerts (filtered)	#episodes	#ES/ #ESQ	#ESS	#AGs
T1	81373	26651	655	103	108	53
T2	42474	4922	609	86	92	7
T5	52550	11918	622	69	74	51
T7	47101	8517	576	63	73	23
T8	55170	9037	439	67	79	33
T9	51602	10081	1042	69	110	30

RIT
TUDelft

[2] Attacker strategy visualization

- Shows how the attack transpired
- 3 teams, 5 attempts
- 3 ways to reach objective
  - Discovered by S-PDFA

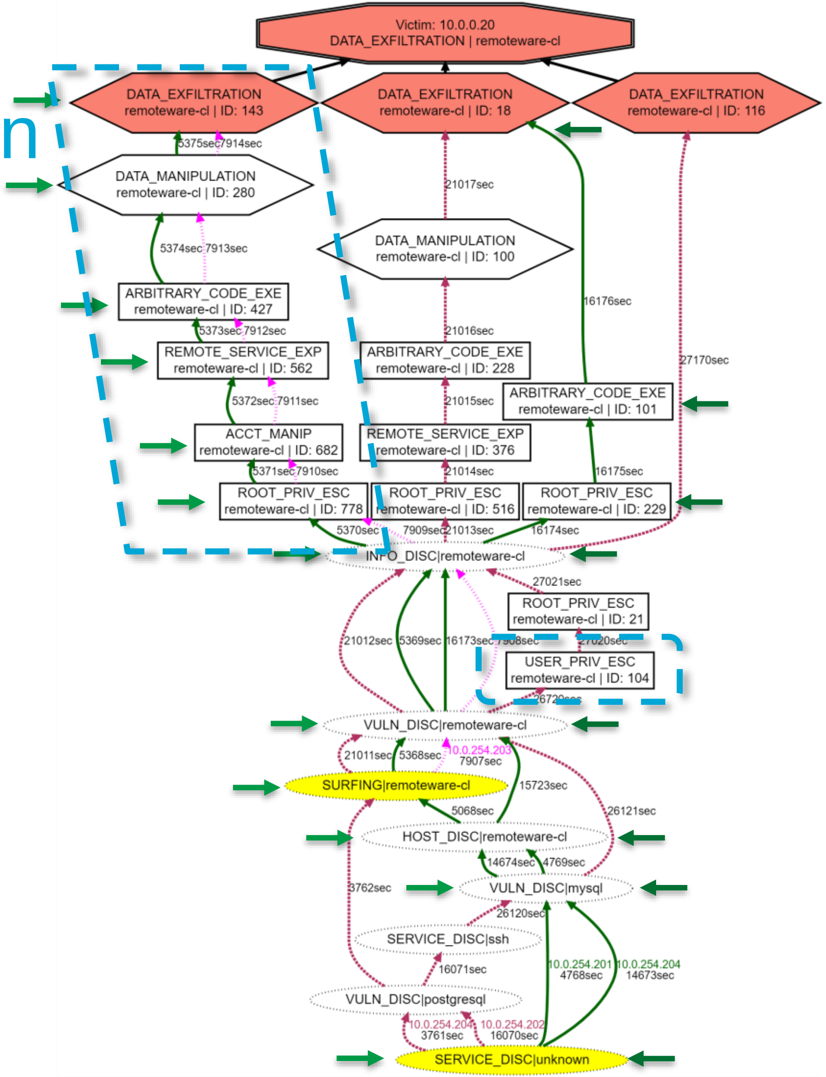


[3] Attacker strategy comparison

- T5 and T8 share a common strategy
- Only T1 does user privilege escalation

- Some paths are shorter than others

- Attackers follow shorter paths to re-exploit an objective in 84.5% cases



Future research directions

- Attack graph prioritization
- Advanced comparative visual analysis for strategy comparison
- Applications
  - Improving IDS signatures
  - Suggesting additional sources for evidence collection

Take aways

- SAGE uses sequence learning to extract attacker strategies
  - Builds attack graphs from intrusion alerts without expert input

- The S-PDFA is critical for
  - Accentuating infrequent severe actions,
  - Identifying contextually different actions

- Alert-driven attack graphs
  - Compress thousands of alerts in a few AGs
  - Provide insights into attacker strategies
  - Capture attackers’ behavior dynamics

Thank you! Questions?

- SAGE uses sequence learning to extract attacker strategies
  - Builds attack graphs from intrusion alerts without expert input
- The S-PDFA is critical for
  - Accentuating infrequent severe actions,
  - Identifying contextually different actions
- Alert-driven attack graphs
  - Compress thousands of alerts in a few AGs
  - Provide insights into attacker strategies
  - Capture attackers' behavior dynamics

SAGE is open-source!

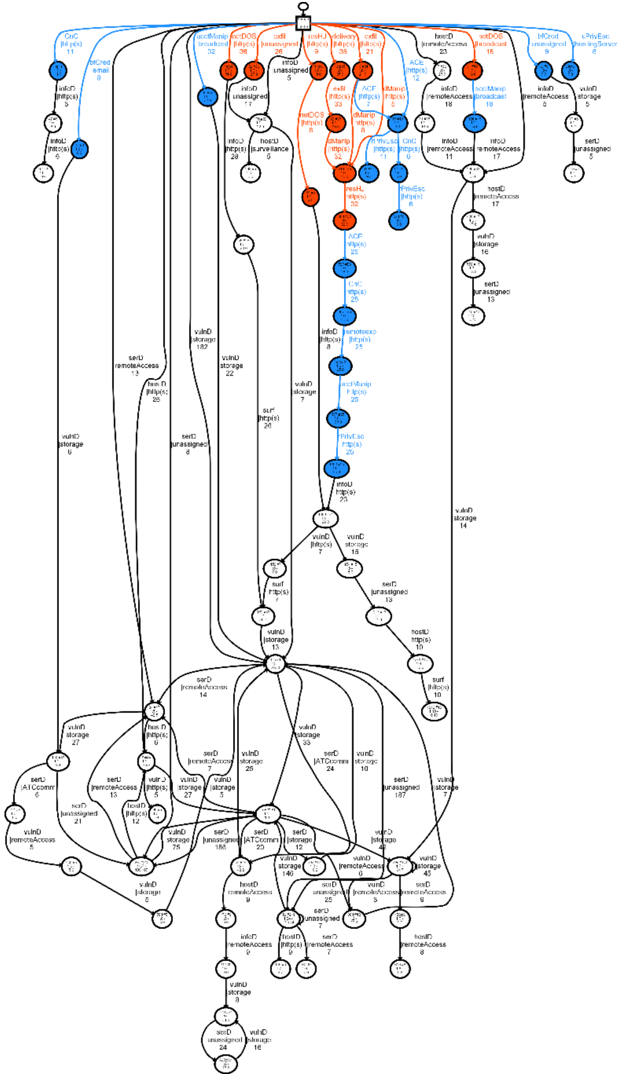
azqa.nadeem@tudelft.nl @azqa_nadeem https://cyber-analytics.nl/

RIT TUDelft

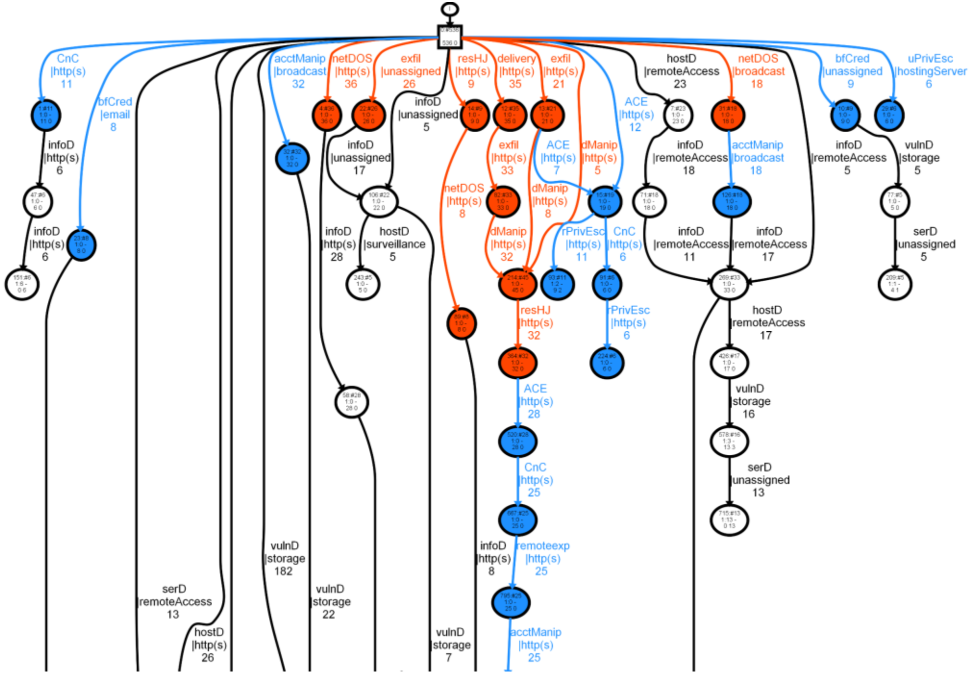
Extra: S-PDFA specifics

- $A = \langle Q, \Sigma, \Delta, P, q_0 \rangle \rightarrow$ model structure
- $Q \rightarrow$ finite set of states
- $\Sigma \rightarrow$ finite alphabet of symbols
- $\Delta \rightarrow$ finite set of transitions
- $q_0 \in Q \rightarrow$ final state (suffix model)

- $\langle q, q', a \rangle \in \Delta \rightarrow$ a transition, where $q, q' \in Q$ and $a \in \Sigma$
- $\{P : \Delta \rightarrow [0,1]\} \rightarrow$ transition probability function
- $P(s) = \prod_{0 \le i < n} P(\langle q_i, q_{i+1}, a_{n-i} \rangle) \rightarrow$ sequence probability
- $\sum_{q,a} P(\langle q, q', a \rangle) = 1$



Extra: S-PDFA specifics

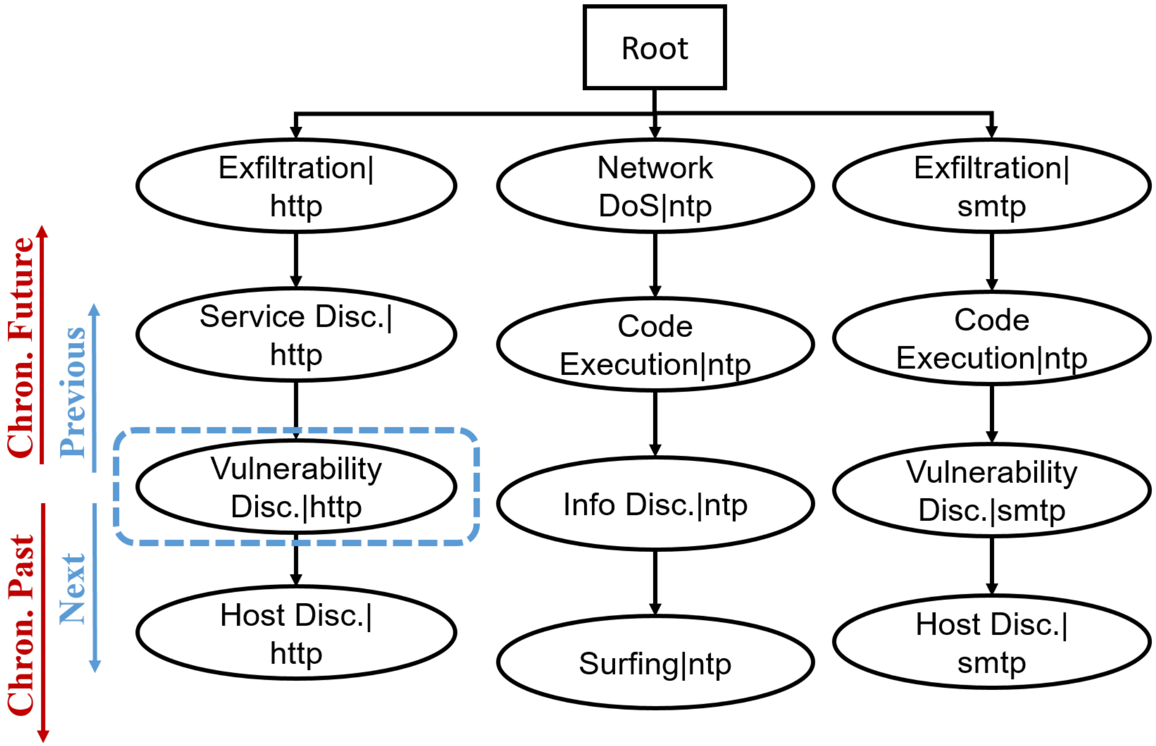


Extra: S-PDFA evaluation

- $Perplexity(M) = 2^{-\frac{1}{N}\sum_{i=1}^{N} log_2 P(x_i)}$
- $P(x_i) \rightarrow$ probability of trace
- $N \rightarrow$ Number of traces

	Suffix tree	Markov chain	SAGE S-PDFA
Training set	1265.4*	13659.6	2397.8
Holdout test set	13020.7	11617.8	9884.6*

Extra: Suffix-tree specifics



Extra: Suricata alert specifics


```
{   '_sourcetype': 'suricata:alert',
    'alert': {   'category': 'Attempted Information Leak',
                 'severity': 2,
                 'signature': 'ET POLICY Python-urllib\\/ '
                              'Suspicious User Agent'},
    'dest_ip': '169.254.169.254',
    'dest_port': 80,
    'src_ip': '10.0.0.20',
    'src_port': 56952,
    'timestamp': '2018-11-03T13:51:58.205548+0000'}}
```

Extra: Episode creation specifics