Using Sequential Traces for Attacker Behavior Analysis

Azqa Nadeem

PhD candidate

Cyber Analytics Lab

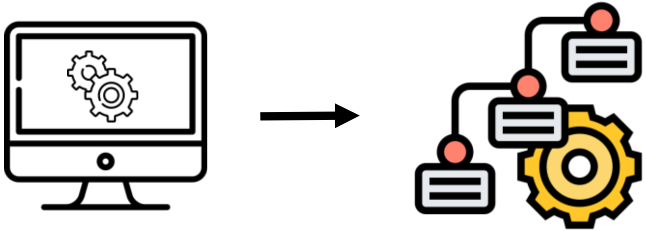
TUDelft

15 June 2021

Cyber Analytics Lab

Dynamic observables

- Program execution → observable data
- Network traffic, software logs, intrusion alerts, …

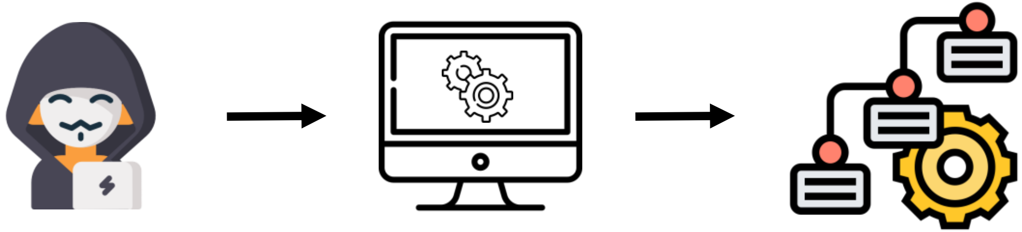


TU Delft

Dynamic observables

- Program execution → observable data
- Network traffic, software logs, intrusion alerts, …

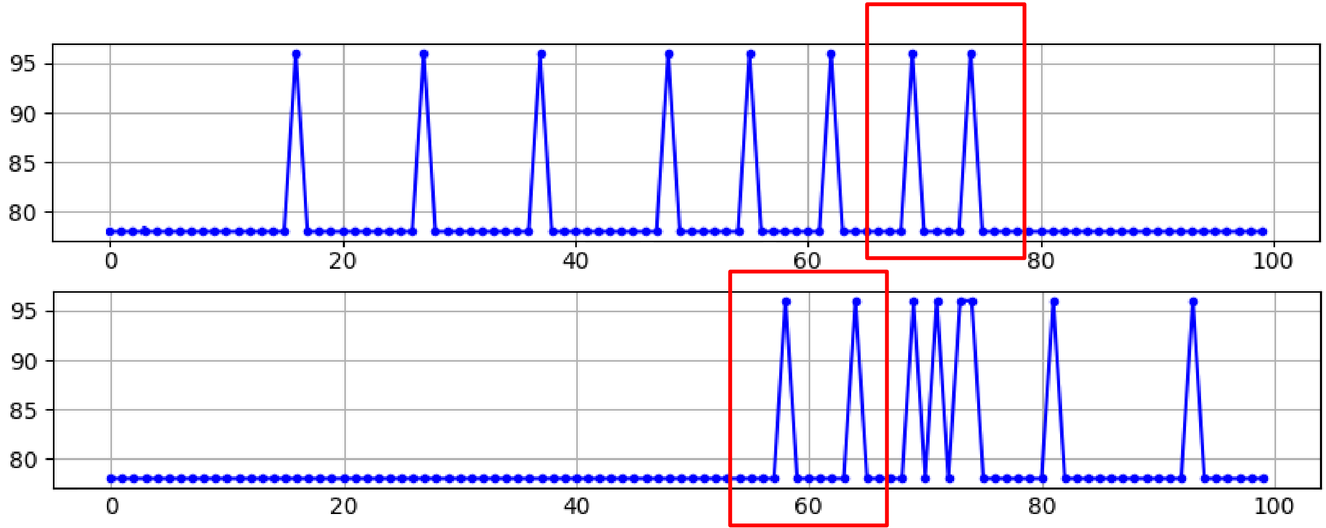
- Proxy to attacker intent



Sequential traces (Dynamic)

- Patterns in temporal data
- Limited data required → insightful patterns

Challenges:
- Curse of dimensionality
- Visualization?
- Distance measure?
- Performance
- Outliers are interesting
- …



TU Delft

Use case I

Problem scenario

- Alert fatigue: Security analysts handle >1M intrusion alerts/day*

- How to make alert analysis easier?
  - By answering *"How did an attack happen?"*

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

What’s already out there?

- “Alert correlation” groups related alerts
  - But how did the attack happen?

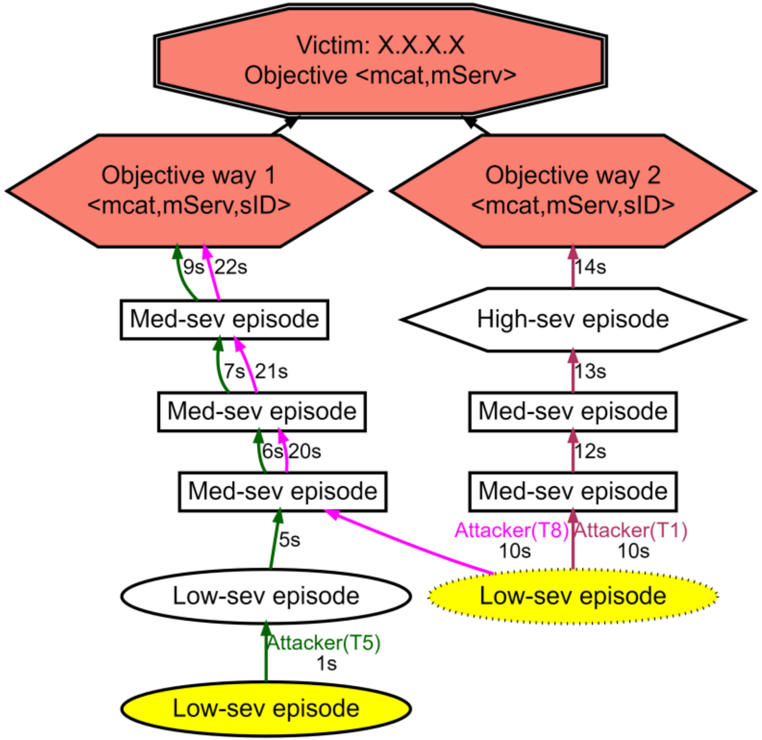
- Attack graph generation (MulVAL*)
  - Require: network structure + vulnerability reports

- Attack model generation (Process mining^)
  - Visual summary of alerts

*X. Ou, S. Govindavajhala, and A. W. Appel, “Mulval: A logic based network security analyzer.” in USENIX security symposium, 2005.
^De Alvarenga, S. C., Barbon Jr, S., Miani, R. S., Cukier, M., & Zarpelão, B. B., “Process mining and hierarchical clustering to help intrusion alert visualization.” in Computers & Security, 2018.

SAGE: Attack graph generator

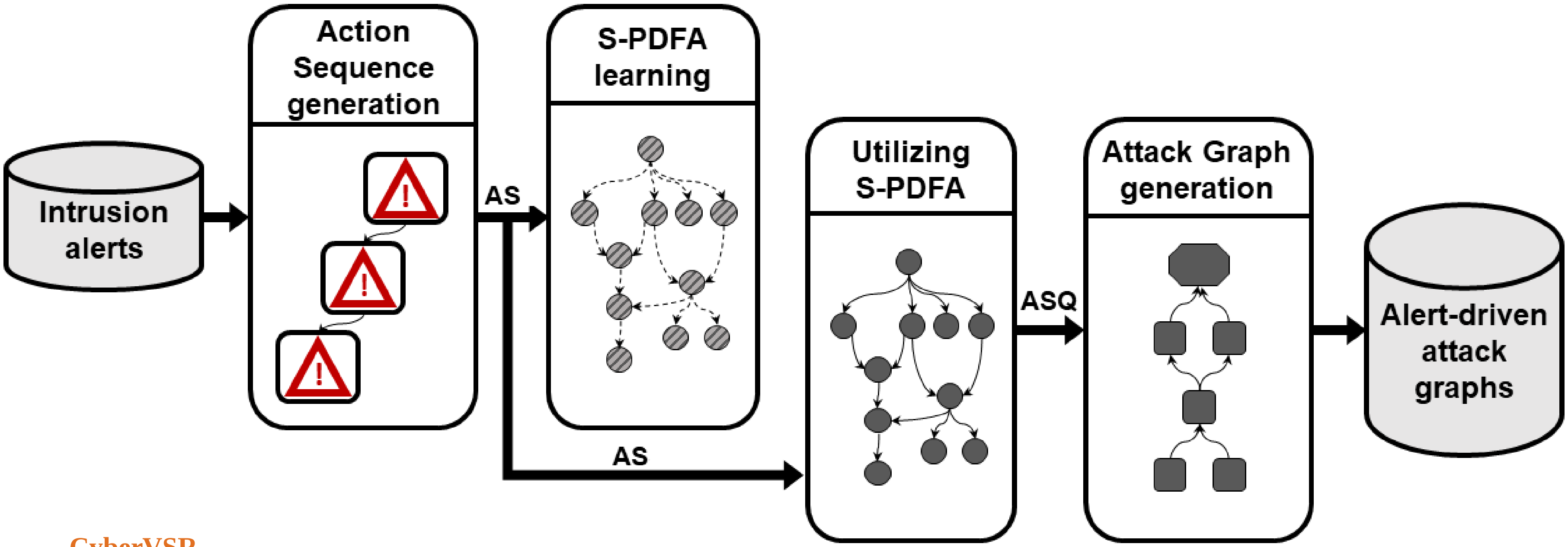
- *<u>Goal:</u> Visualize attacker strategies from intrusion alerts*
- Extract targeted attack graphs
- Discover attacker strategies
  - Without prior knowledge
  - From heaps of alerts
  - Without losing alerts



TU Delft

Alert-driven attack graphs using S-PDFA. Azqa Nadeem, Sicco Verwer, Stephen Moskal, Shanchieh Jay Yang. In IEEE Transactions on Dependable and Secure Computing, 2021. (Submitted)

SAGE: Pipeline



Alert-driven attack graphs using S-PDFA. *Azqa Nadeem, Sicco Verwer, Stephen Moskal, Shanchieh Jay Yang. In IEEE Transactions on Dependable and Secure Computing, 2021. (Submitted)*

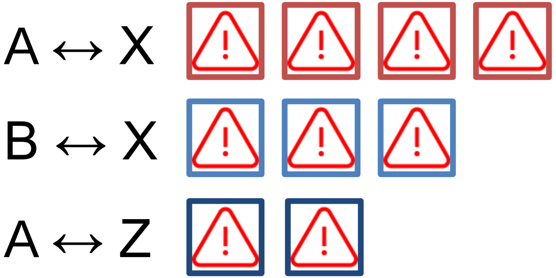
Alerts → Actions


```
{   '_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'}}
```


IDS alerts

$$\mathit{Action} = \left\langle \begin{array}{c} \mathit{start\ time}, \\ \mathit{end\ time}, \\ \mathit{attack\ stage}, \\ \mathit{targeted\ service} \end{array} \right\rangle$$

Alert Sequences



sorted by *start time*

Alerts → Actions


```
{   '_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'}}
```


IDS alerts

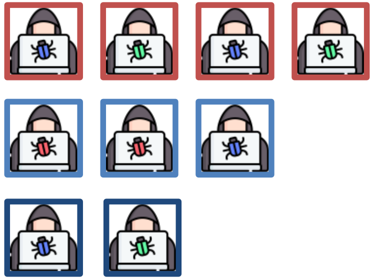
$$Action = \left\langle \begin{array}{c} start\ time, \\ end\ time, \\ attack\ stage, \\ targeted\ service \end{array} \right\rangle$$

Action Sequences

A ↔ X

B ↔ X

A ↔ Z



TU Delft

sorted by *start time*

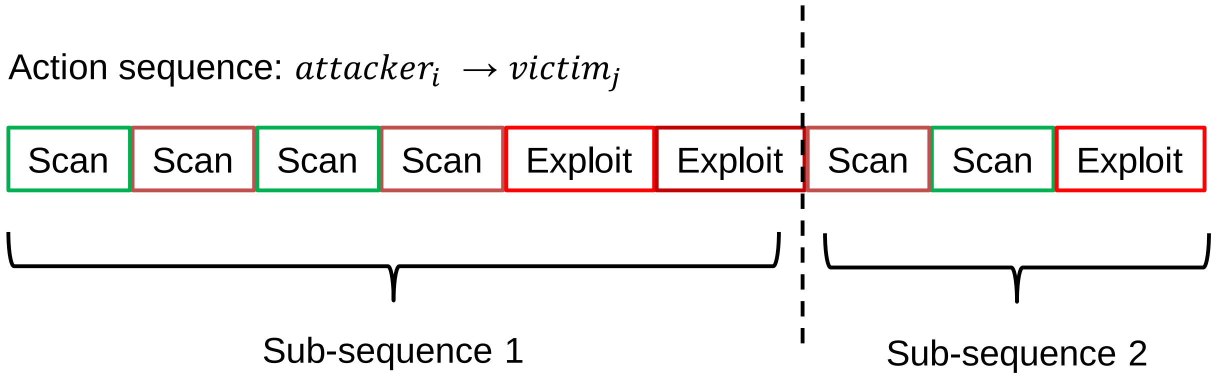
Action sub-sequences

Action sequence: $attacker_i \rightarrow victim_j$

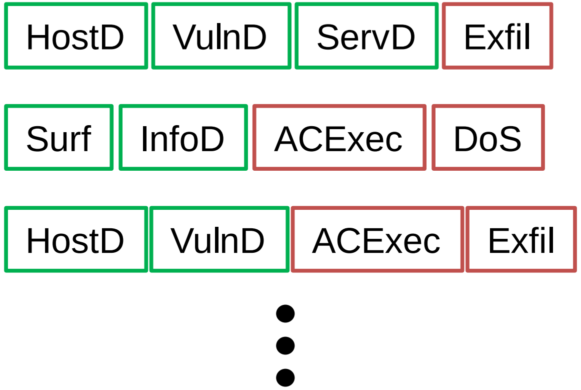
Scan	Scan	Scan	Scan	Exploit	Exploit	Scan	Scan	Exploit

TUDelft

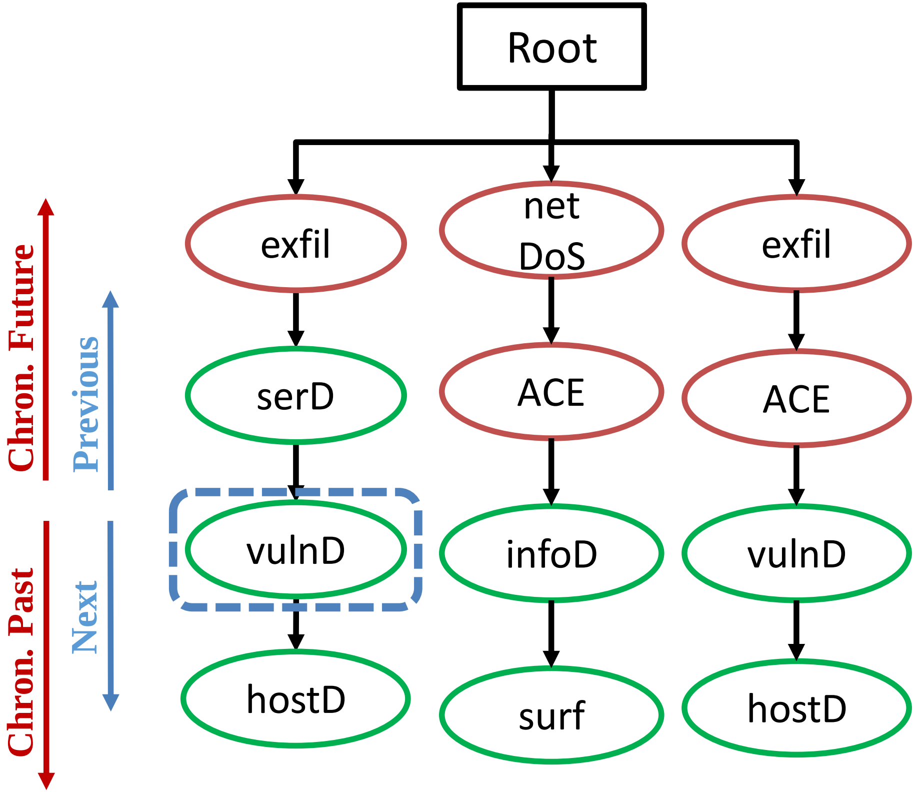
Action sub-sequences



Suffix Tree



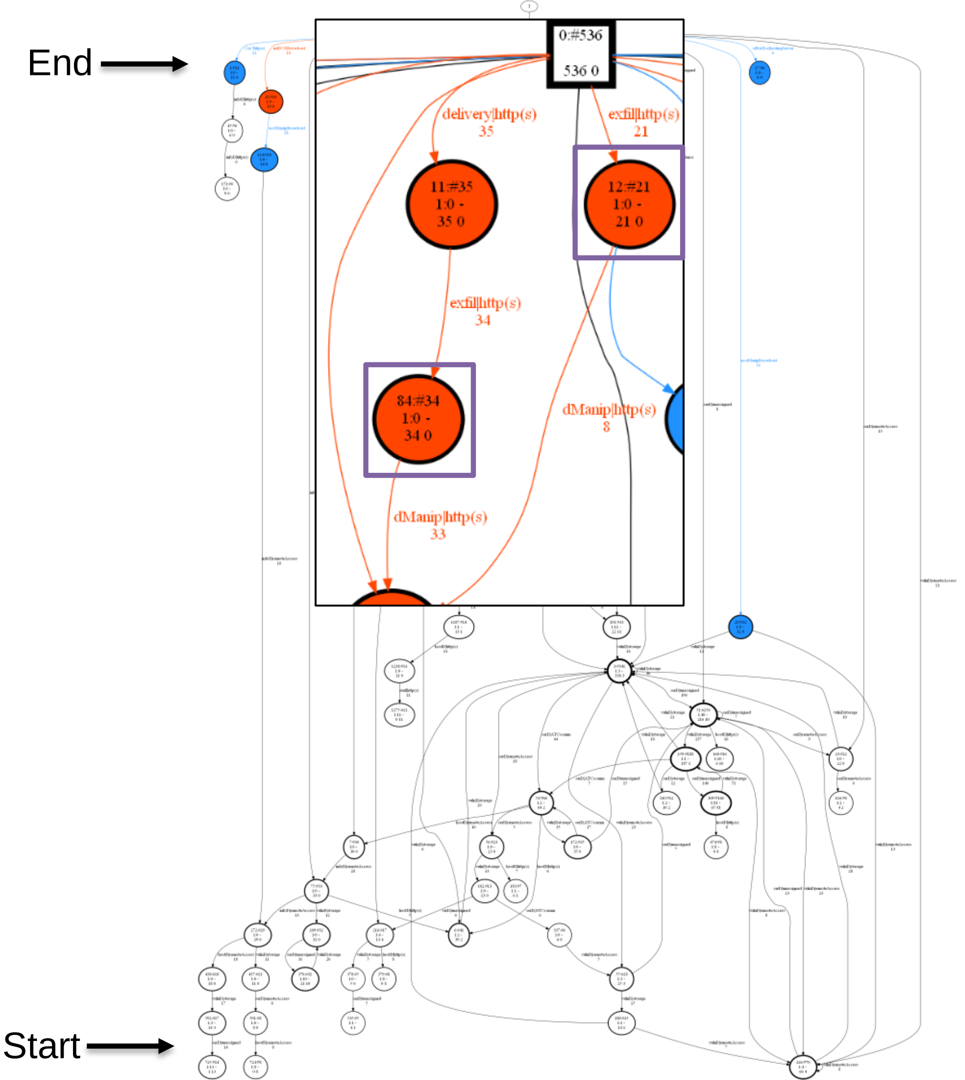
… all sub-sequences!



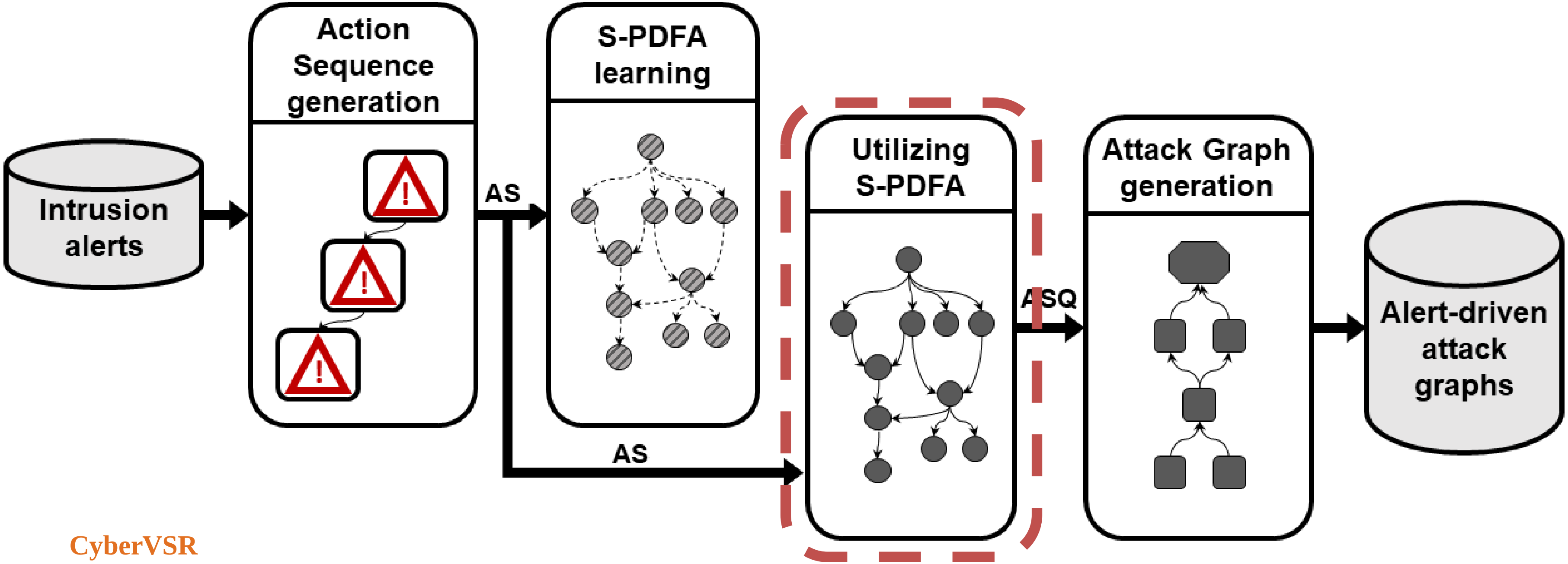
TUDelft

S-PDFA

- *Suffix-based Probabilistic Deterministic Finite Automaton*
- State colors
  - Severe | Medium | Low
- Context modelling



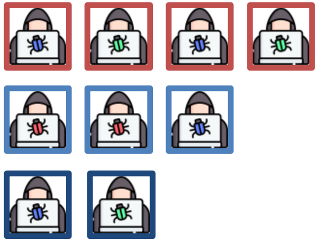
SAGE: Pipeline

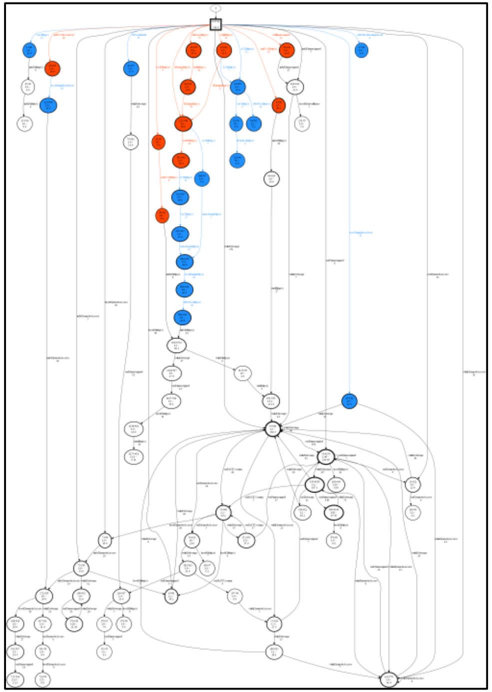


Alert-driven attack graphs using S-PDFA. *Azqa Nadeem, Sicco Verwer, Stephen Moskal, Shanchieh Jay Yang. In IEEE Transactions on Dependable and Secure Computing, 2021. (Submitted)*

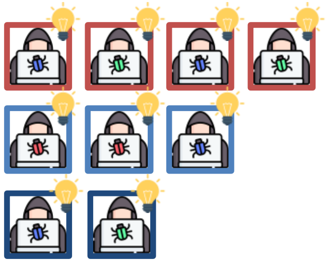
Encoding action sequences

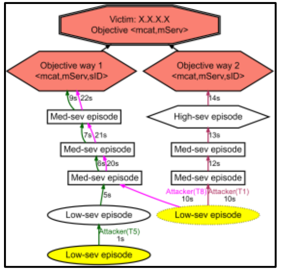
Action sequences





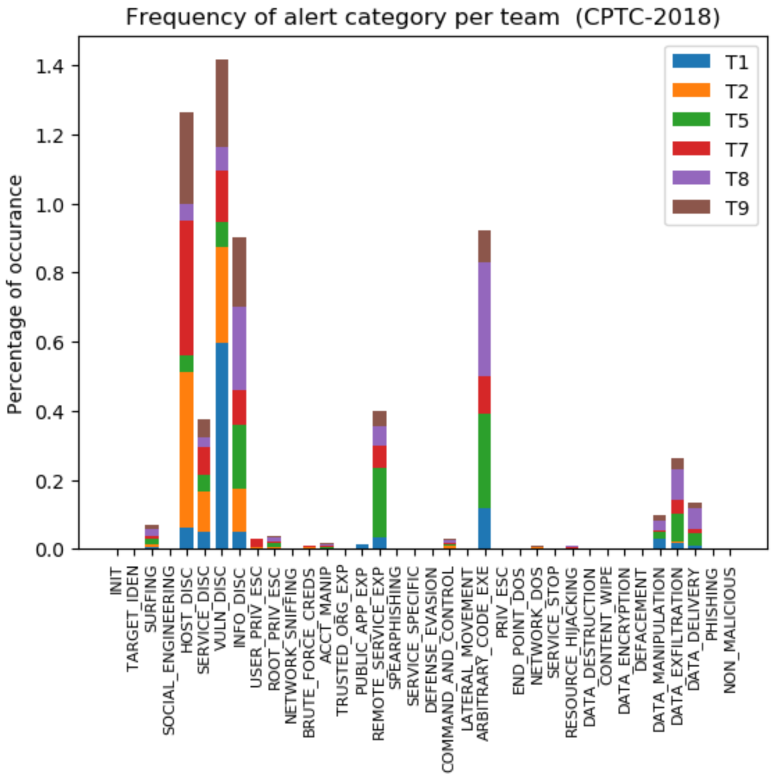
State sequences





Threat model and Dataset

- CPTC '18: Pen. testing competition[1]
- Moskal's Attack-Intent framework[2]
  - Alert signature → Attack stage
- Distributed multi-stage attacks



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

Results: Workload reduction

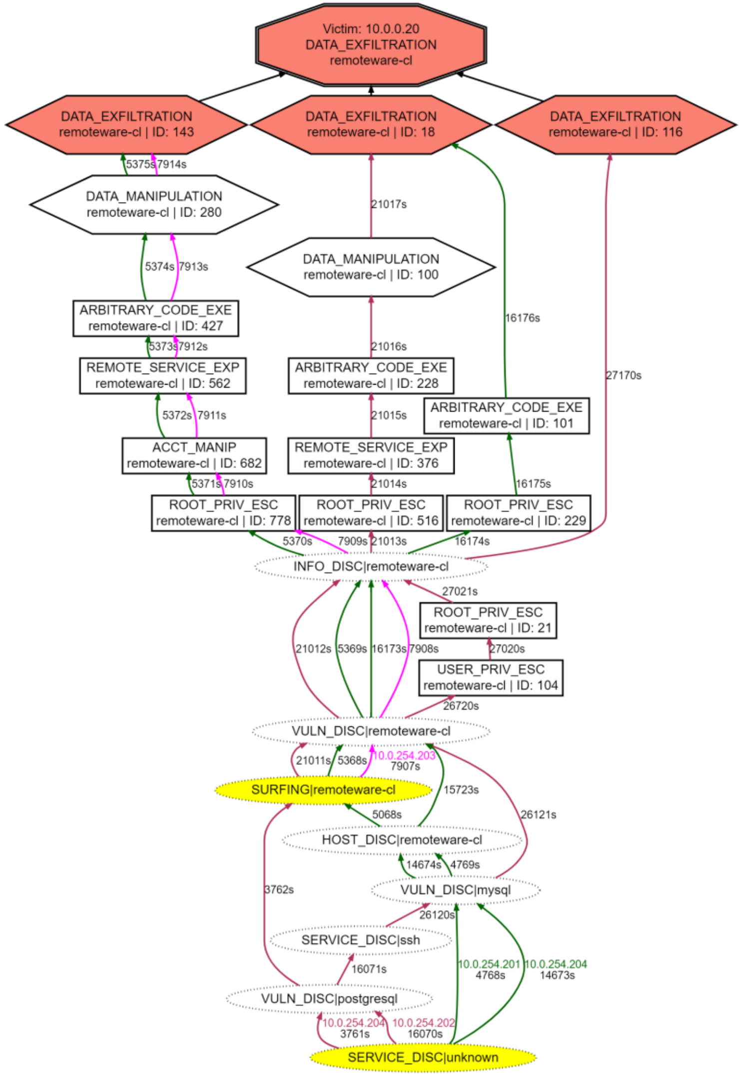
Table 1: Workload reduction in the CPTC-2018 dataset.

	# alerts (raw)	# alerts (filtered)	#actions	#AS/ #ASQ	#ASS	#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

330,270 alerts → 93 AGs!

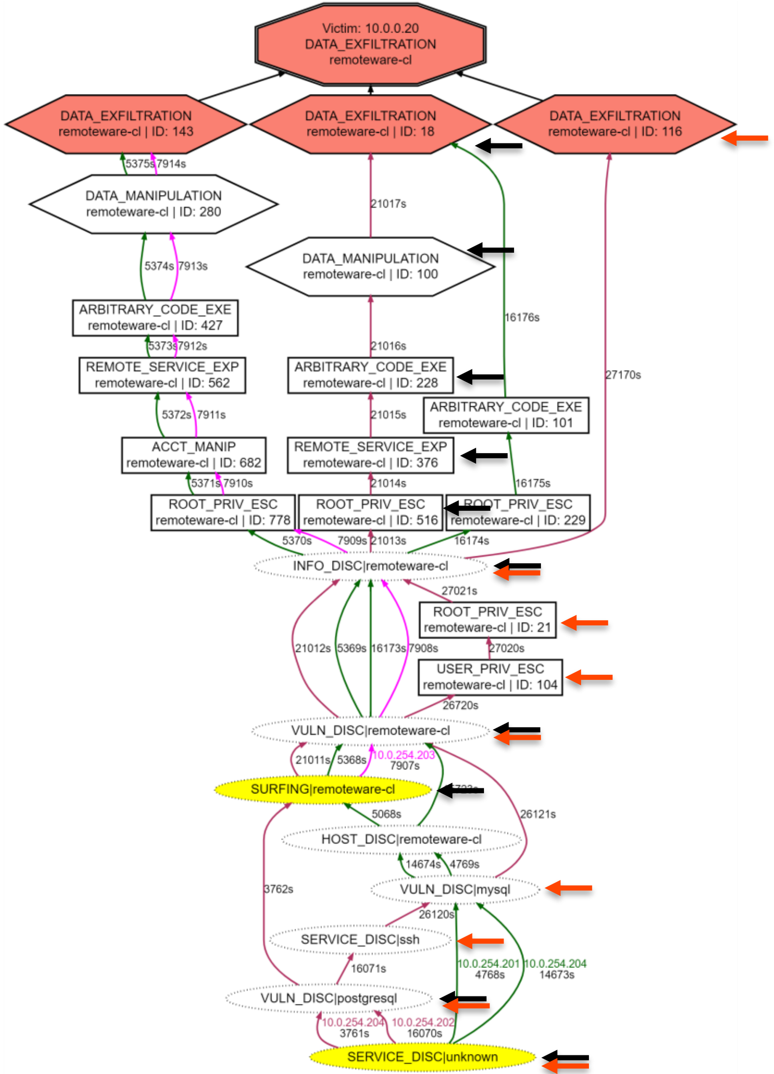
TU Delft

AG Analysis [1/3]



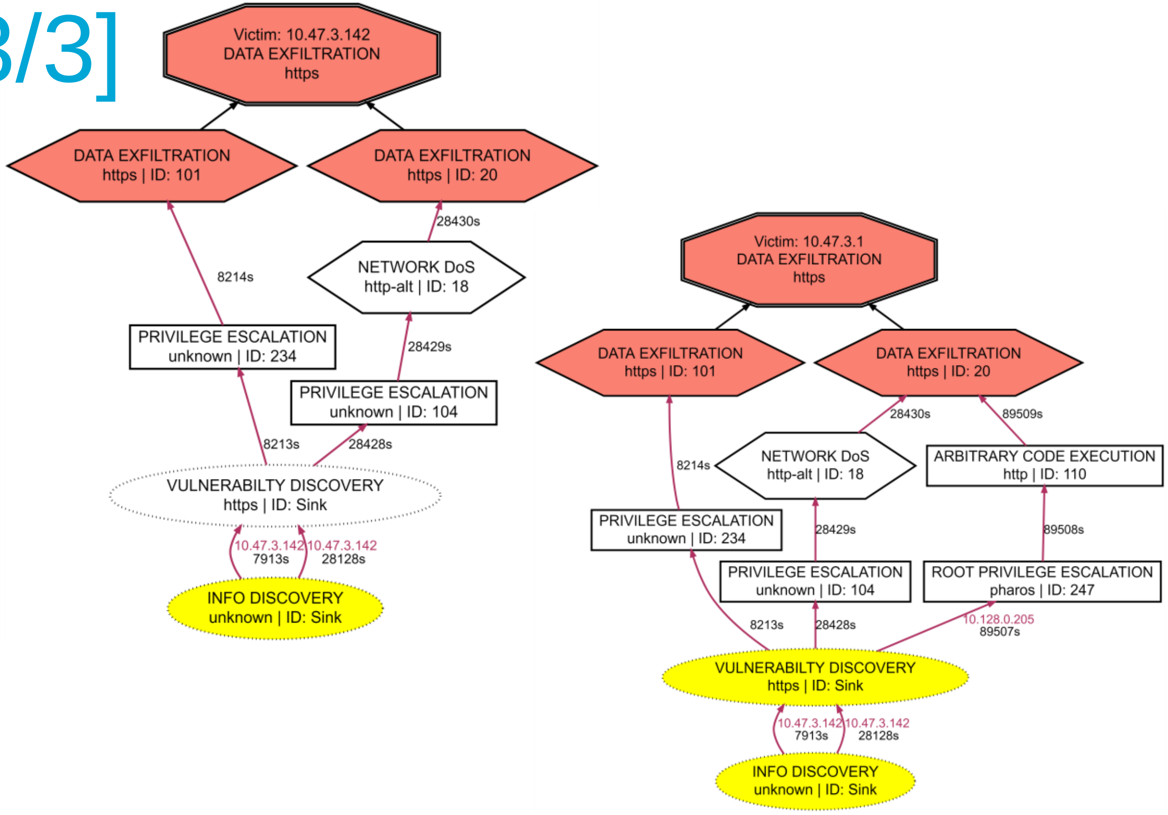
AG Analysis [2/3]

- Attackers follow shorter paths after discovering longer ones



AG Analysis [3/3]

- Near-identical strategies appear as highly similar AGs



SAGE: Open issues

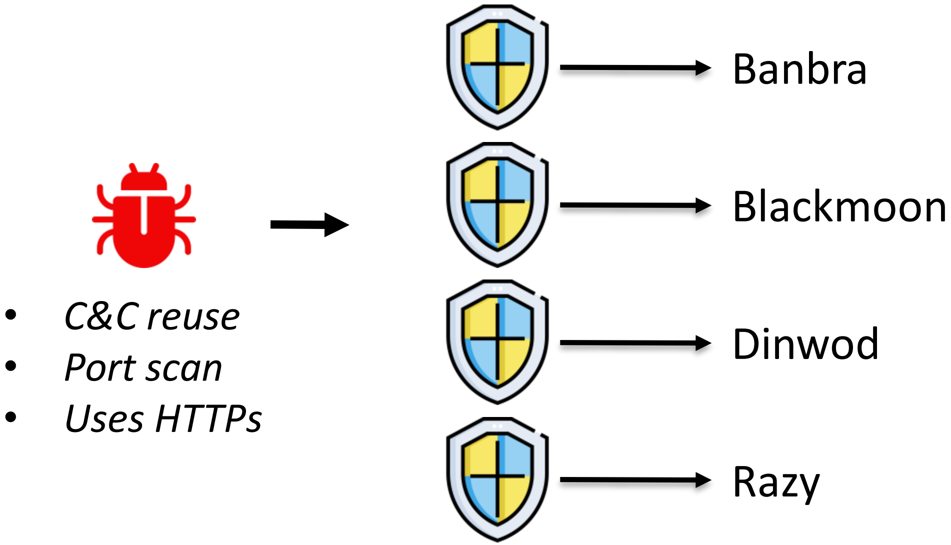
- Attack path prioritization
- Missing paths in AGs
- Adversarial robustness(?)

Alert-driven attack graphs using S-PDFA. Azqa Nadeem, Sicco Verwer, Stephen Moskal, Shanchieh Jay Yang. In IEEE Transactions on Dependable and Secure Computing, 2021. (Submitted)

Use case II

Problem scenario

- Malware labels are inconsistent and black-box



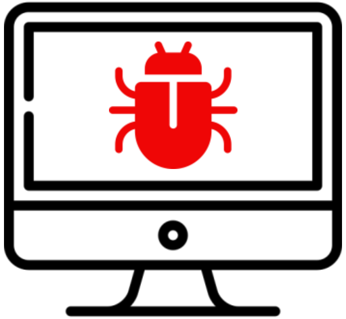
Problem scenario

- Malware labels are inconsistent and black-box
- How to discover behaviors?

Beyond Labeling: Using Clustering to Build Network Behavioral Profiles of Malware Families. *Azqa Nadeem, Christian Hammerschmidt, Carlos H. Ganan, Sicco Verwer. In Malware Analysis using Artificial Intelligence and Deep Learning, Springer, 2021.*

Network trace collection

- Malware infected machine generates network traffic



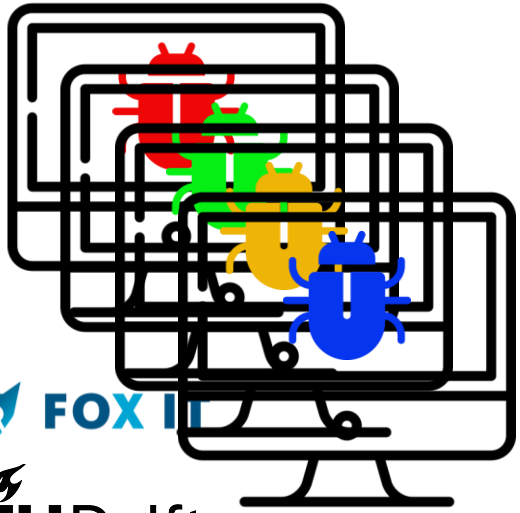
No.	Source	Destination	Protocol	Length	Info
40	192.168.1.2	192.168.1.110	ICMP	82	Redirect (Redirect for host)
41	CzNicZSP_00:0…	PcsCompu_7c:9…	ARP	60	192.168.1.1 is at d8:58:d7:00:0f:72
42	192.168.1.110	203.153.165.21	TCP	182	49191 → 8343 [PSH, ACK] Seq=1 Ack=1 Win=65700 Len=128
43	203.153.165.21	192.168.1.110	TCP	60	8343 → 49191 [ACK] Seq=1 Ack=129 Win=15744 Len=0
44	203.153.165.21	192.168.1.110	TCP	1188	8343 → 49191 [PSH, ACK] Seq=1 Ack=129 Win=15744 Len=1134
45	192.168.1.110	203.153.165.21	TCP	380	49191 → 8343 [PSH, ACK] Seq=129 Ack=1135 Win=64564 Len=326
46	192.168.1.2	192.168.1.110	ICMP	408	Redirect (Redirect for host)
47	203.153.165.21	192.168.1.110	TCP	113	8343 → 49191 [PSH, ACK] Seq=1135 Ack=455 Win=16768 Len=59
48	fd2d:ab8c:225…	fd2d:ab8c:225…	DNS	110	Standard query 0xb554 A www.download.windowsupdate.com

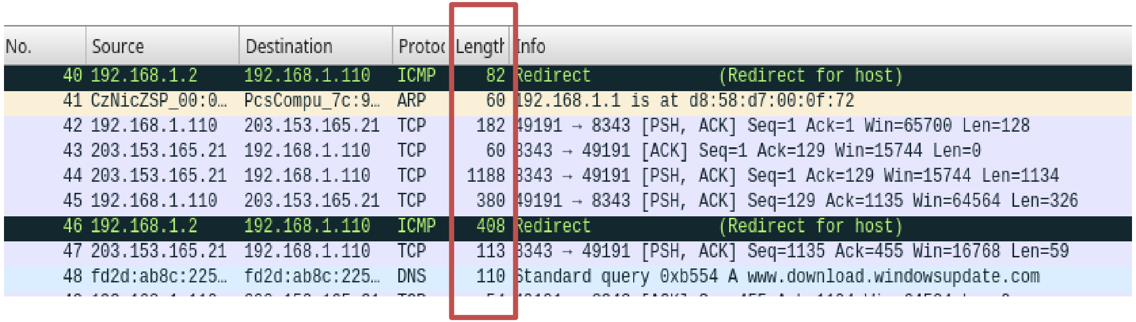
FOX IT

TU Delft

Network trace collection

- Malware infected machine generates network traffic





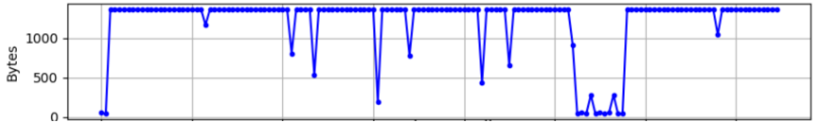
No.	Source	Destination	Protoc	Length	Info
40	192.168.1.2	192.168.1.110	ICMP	82	Redirect (Redirect for host)
41	CzNicZSP_00:0…	PcsCompu_7c:9…	ARP	60	192.168.1.1 is at d8:58:d7:00:0f:72
42	192.168.1.110	203.153.165.21	TCP	182	49191 → 8343 [PSH, ACK] Seq=1 Ack=1 Win=65700 Len=128
43	203.153.165.21	192.168.1.110	TCP	60	8343 → 49191 [ACK] Seq=1 Ack=129 Win=15744 Len=0
44	203.153.165.21	192.168.1.110	TCP	1188	8343 → 49191 [PSH, ACK] Seq=1 Ack=129 Win=15744 Len=1134
45	192.168.1.110	203.153.165.21	TCP	380	49191 → 8343 [PSH, ACK] Seq=129 Ack=1135 Win=64564 Len=326
46	192.168.1.2	192.168.1.110	ICMP	408	Redirect (Redirect for host)
47	203.153.165.21	192.168.1.110	TCP	113	8343 → 49191 [PSH, ACK] Seq=1135 Ack=455 Win=16768 Len=59
48	fd2d:ab8c:225…	fd2d:ab8c:225…	DNS	110	Standard query 0xb554 A www.download.windowsupdate.com

FOX IT

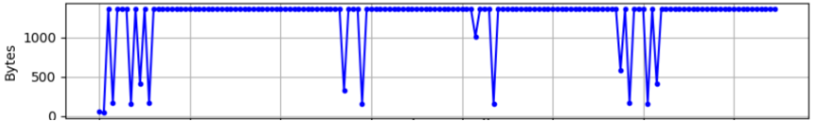
TUDelft

Network trace collection

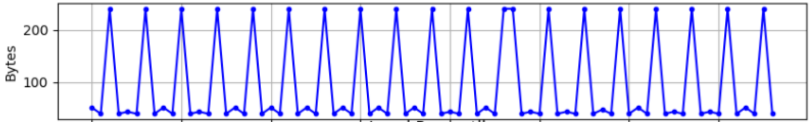
Zeus-738f →



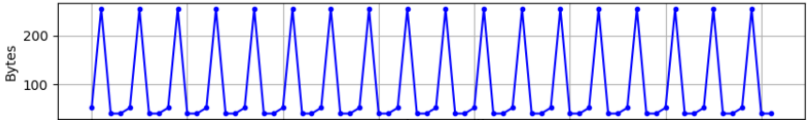
Gozi-4bd7 →



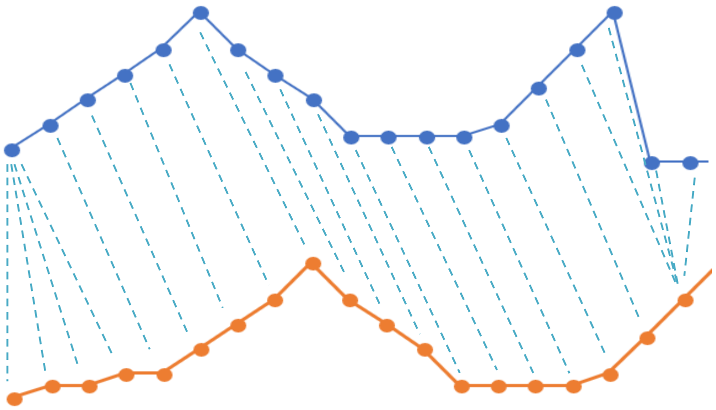
Zeus-78de →



Zeus-6631 →



Behavior discovery

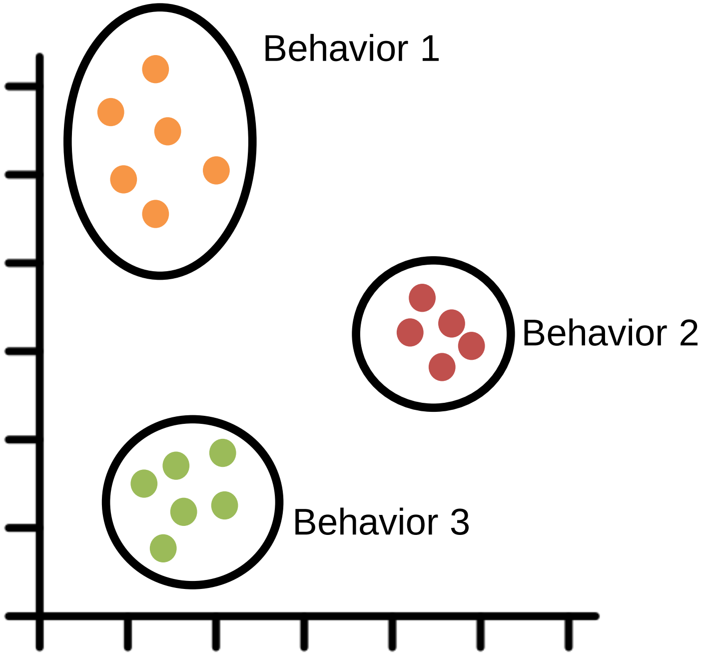


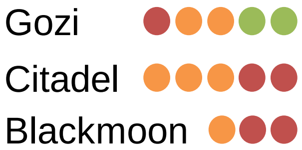
Dynamic Time Warping

$$D(i,j) = |A_i - B_j| + \min(D(i-1,j), D(i,j-1), D(i-1,j-1))$$

TU Delft

Behavior discovery





TU Delft

Malware Behavior Profiles

	B	C	D	DL	GE	GI	R	Z	ZP	ZPa	Zv1	ZVA
SSDP traffic	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	-	✓
Broadcast traffic	✓	✓	-	✓	-	✓	✓	-	✓	-	✓	✓
LLMNR traffic	✓	✓	-	✓	-	✓	-	-	-	-	-	-
System. port scan	✓	✓	-	-	-	✓	✓	-	-	-	-	✓
Random. port scan	✓	✓	-	-	-	✓	✓	-	-	-	-	✓
In conn spam	-	-	-	-	-	✓	-	-	-	-	-	-
Out conn spam	-	-	-	-	-	✓	-	-	-	-	-	-
Malicious Subnet	-	-	-	-	-	-	-	-	-	-	-	✓
In HTTPs	-	✓	-	✓	-	✓	-	-	-	✓	-	-
Out HTTPs	-	-	-	-	-	✓	-	-	-	✓	-	-
C&C reuse	✓	-	-	-	-	-	-	-	-	✓	-	-
Misc.	✓	✓	-	✓	-	✓	-	✓	-	✓	-	✓
# Clusters	7	11	*1*	8	*1*	*16*	4	2	*1*	7	*1*	7

Beyond Labeling: Using Clustering to Build Network Behavioral Profiles of Malware Families. Azqa Nadeem, Christian Hammerschmidt, Carlos H. Ganan, Sicco Verwer. In Malware Analysis using Artificial Intelligence and Deep Learning, Springer, 2021.

Wrap-up

- Sequence of dynamic observables → attacker intent
- 2 use-cases
  - Intrusion alerts → Attacker strategy attack graphs
  - Network traffic → Malware behavior profiles

- Input: observables | Output: Intelligence

- Unsupervised setting with limited prior knowledge

Thank you!
Questions?

Sequence of dynamic observables → attacker intent

2 use-cases

- Intrusion alerts → Attacker strategy attack graphs
- Network traffic → Malware behavior profiles

Input: observables | Output: Intelligence

Unsupervised setting with limited prior knowledge

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

TU Delft

Icons courtesy Freepik, Those Icons, Dinosoft Labs, Pixel Perfect, Flat Icons, Kiranshastry from www.flaticon.com

Action extraction