Creation and use of a representative dataset for Advanced Persistent Threats detection

Tommaso Puccetti[1], Simona De Vivo[2], Davide Zhang[1], Pietro Liguori[2], Roberto Natella[2] and Andrea Ceccarelli[1]

1 - Department of Mathematics and Informatics, University of Florence
2 - Federico II University of Naples, Naples (Italy)

UNIVERSITÀ DEGLI STUDI FIRENZE
DIMAI
DIPARTIMENTO DI MATEMATICA E INFORMATICA "ULISSE DINI"

SAFECOMP 2025

RCL
RESILIENT COMPUTING LAB

Advanced Persistent Threats

- Advanced Persistent Threats (APTs):
  - **Advanced**: combine a full spectrum of cyber attacks and intelligence-gathering techniques.
  - **Persistent**: highly determined and persistent attackers.
  - **Threats**: extract information and lead to big-scale damage.
- Target industrial or critical infrastructures, potentially impacting safety (Stuxnet attack [1].)

[1] Langner, R.: Stuxnet: Dissecting a Cyberwarfare Weapon. IEEE Secur. Priv. 9, 49–51 (2011). https://doi.org/10.1109/MSP.2011.67.

APT and Network Intrusion Detection

- **Anomaly-based Intrusion Detection systems** are a promising means to protect against APTs.
- They can derive complex rules to identify attacks by learning from data.
- They can detect **unknown attacks and zero-day day**.
- **Network Intrusion Detection (NID)**: network traffic encoded as **structured data**.

Protocol	Flow_Duration	Total_Fwd_Packets	Total_Length_of_Fwd_Packets	Label
17	49670	1	50	normal
6	17	1	0	normal
17	77141	1	49	normal
17	52050	1	48	normal
6	86003050	8	302	attack
6	99277732	5	380	attack
6	98353922	7	349	attack
6	2069	3	380	attack

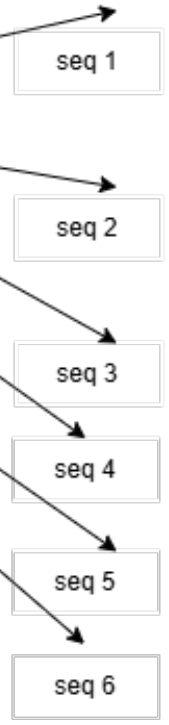
Motivations

- NIDs depend on high-quality **datasets**, which often **fail to represent APT complexity** and the evolution of the attacker strategies.
- Further, state-of-the-art NID datasets:
  - Structured in a flow-based approach, does not present alternating sequences of normal and attack network packets.
  - are not suitable for evaluating the detector's capabilities to interrupt an attack path;
  - are not suitable to measure the time to detect an attack (**attack latency).**

Motivations

SOTA Flow-based approach

	timestamp	protocol	src ip	dst ip	duration (seconds)	label
Flow 1	Tue, 24 Sep 2024 10:59:18	TCP	10.0.0.1	10.0.0.2	52	attack
Flow 2	Tue, 24 Sep 2024 11:30:00	TCP	10.0.0.3	10.0.0.2	150	normal
Flow 3	Tue 24 Sep 2024 12:40:10	UDP	10.0.01	10.0.0.2	18	attack
Flow 4	Tue, 24 Sep 2024 12:53:28	MQTT	10.0.0.5	10.0.0.2	6	normal
Flow 5	Tue, 24 Sep 2024 13:40:10	UDP	10.0.0.1	10.0.0.2	804	attack
Flow 6	Tue, 24 Sep 2024 13:53:34	MQTT	10.0.0.3	10.0.0.2	24	normal



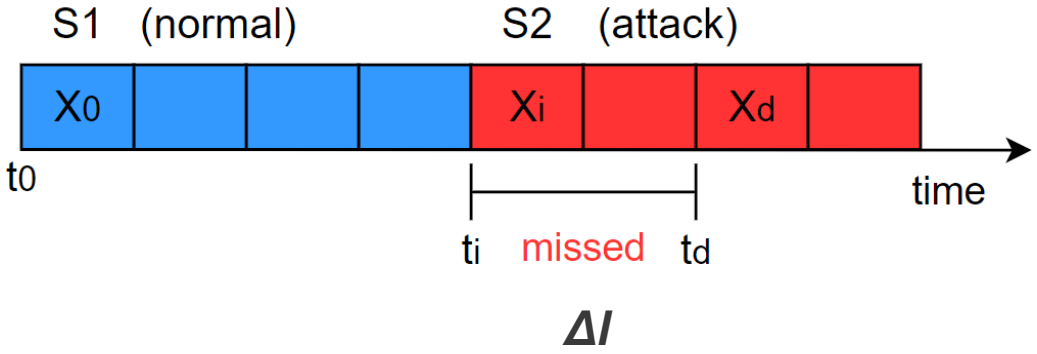
Packet-based dataset

timestamp	...	label
Tue, 24 Sep 2024 10:59:18	...	attack
Tue, 24 Sep 2024 10:59:53	...	attack
Tue, 24 Sep 2024 10:11:00	...	attack
...	...	attack
Tue, 24 Sep 2024 11:30:00	...	normal
timestamp	...	normal
timestamp	...	normal
timestamp	...	normal
Tue 24 Sep 2024 12:40:10	...	attack
timestamp	...	attack
timestamp	...	attack
Tue, 24 Sep 2024 12:53:28	...	normal
timestamp	...	normal
Tue, 24 Sep 2024 13:40:10	...	attack
timestamp	...	attack
timestamp	...	attack
Tue, 24 Sep 2024 13:53:34	...	normal
timestamp	...	normal

Evaluate a Detector using Latency

- We apply an **evaluation approach** for NIDs that considers **attack latency** [5].
- In practice, **we measure latency as a time interval**, or as the number of data points **between** the two data points x_i and x_d.

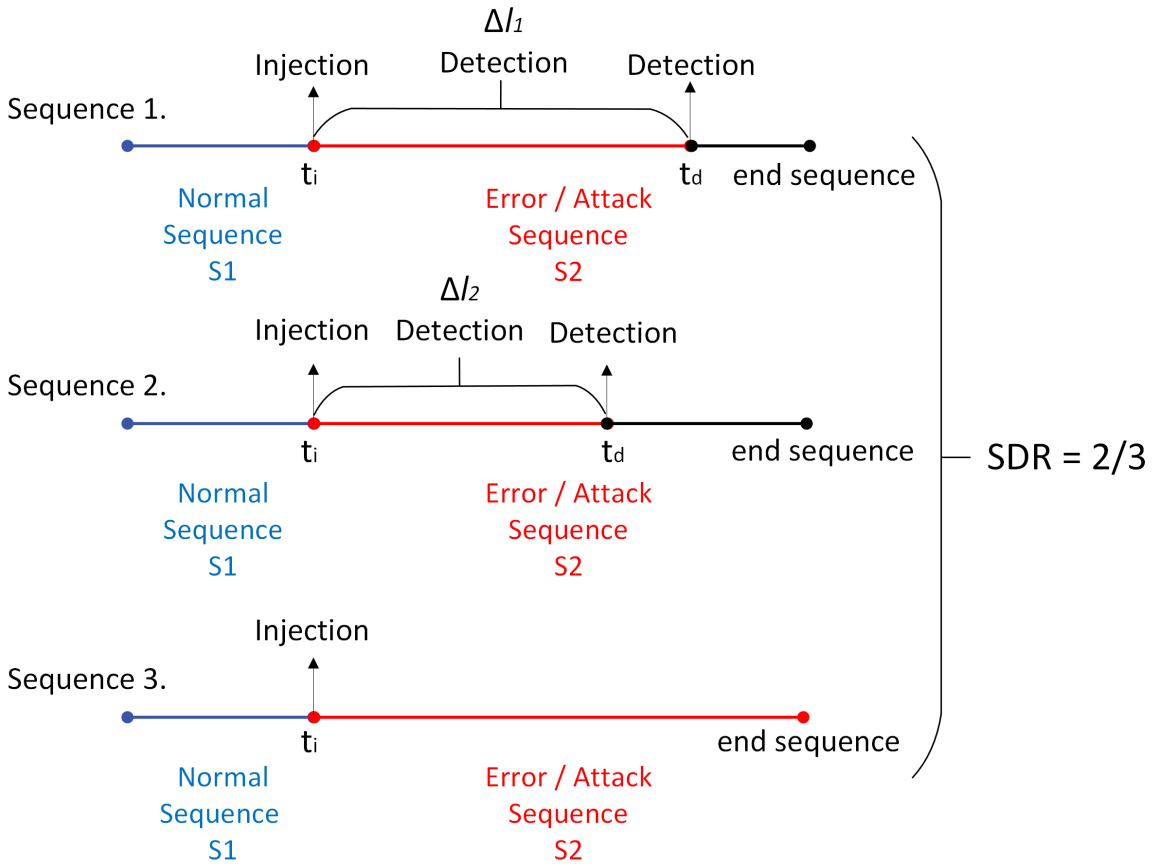
- $\Delta l = t_d - t_i$.
- Average Latency = $\Delta L = \sum_{i=0}^{N} \Delta l_i / N$
- ***SDR** = detected sequences / total sequences.*



[5] T. Puccetti and A. Ceccarelli, "Detection Latencies of Anomaly Detectors - An Overlooked Perspective?," *2024 IEEE 35th International Symposium on Software Reliability Engineering (**ISSRE**)*, Tsukuba, Japan, 2024, pp. 37-48, doi: 10.1109/ISSRE62328.2024.00015.(Link).

Evaluate a Detector using Latency

- $\Delta l = t_d - t_i$.
- Average Latency = $\Delta L = \sum_{i=0}^{N} \Delta l_i / N$
- ***SDR =*** *detected sequences / total sequences.*



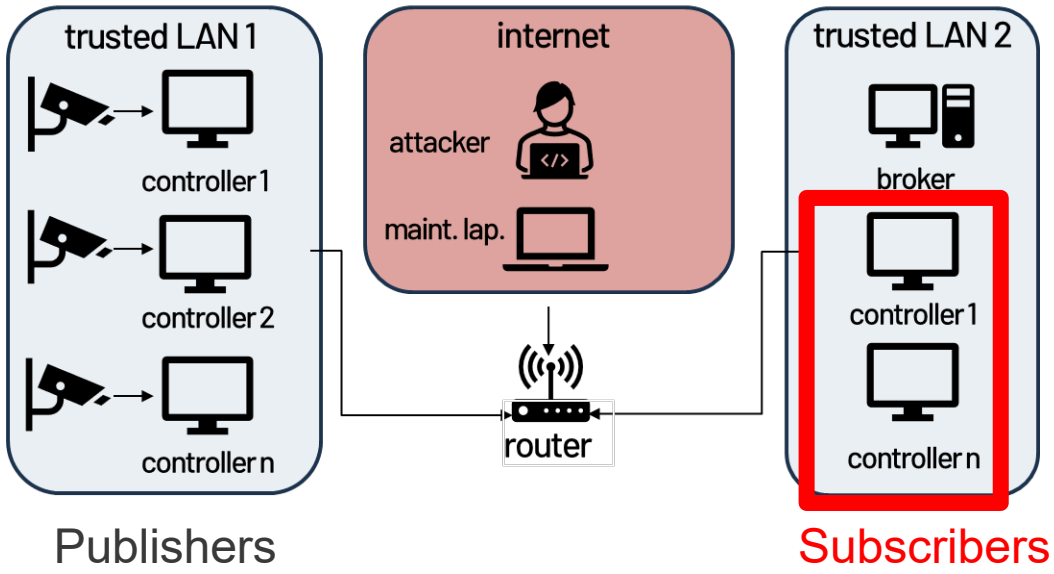
[5] T. Puccetti and A. Ceccarelli, "Detection Latencies of Anomaly Detectors - An Overlooked Perspective?," *2024 IEEE 35th International Symposium on Software Reliability Engineering (**ISSRE**)*, Tsukuba, Japan, 2024, pp. 37-48, doi: 10.1109/ISSRE62328.2024.00015.(Link).

What we do

- We propose a methodology to create a semi-synthetic dataset.
- We reproduce APTs against a simulated industrial network.
- The dataset includes APT paths combining several techniques and covering multiple stages of the APT lifecycle.
- Different from most works, the dataset reproduces publish/subscribe communication traffic from a real network.
- The dataset keeps a detailed track of the attack stages within the network traffic. This allows us to:
  - Measure the Network Intrusion Detectors' ability to interrupt the APT path
  - The attacker's persistence until detection (**time to detect** a step of the APT path).

Simulate Normal Traffic

1. We simulate an industrial system (**DoS/DDoS-MQTT-IoT** dataset [2]).
2. Analyze the dataset to understand network topology (IP addresses, roles).
3. Replicate the dataset network topology.
4. Replay publish messages.
5. We simulate 23 devices (IP 10.0.0.1/23) for **2 days of normal operation**.

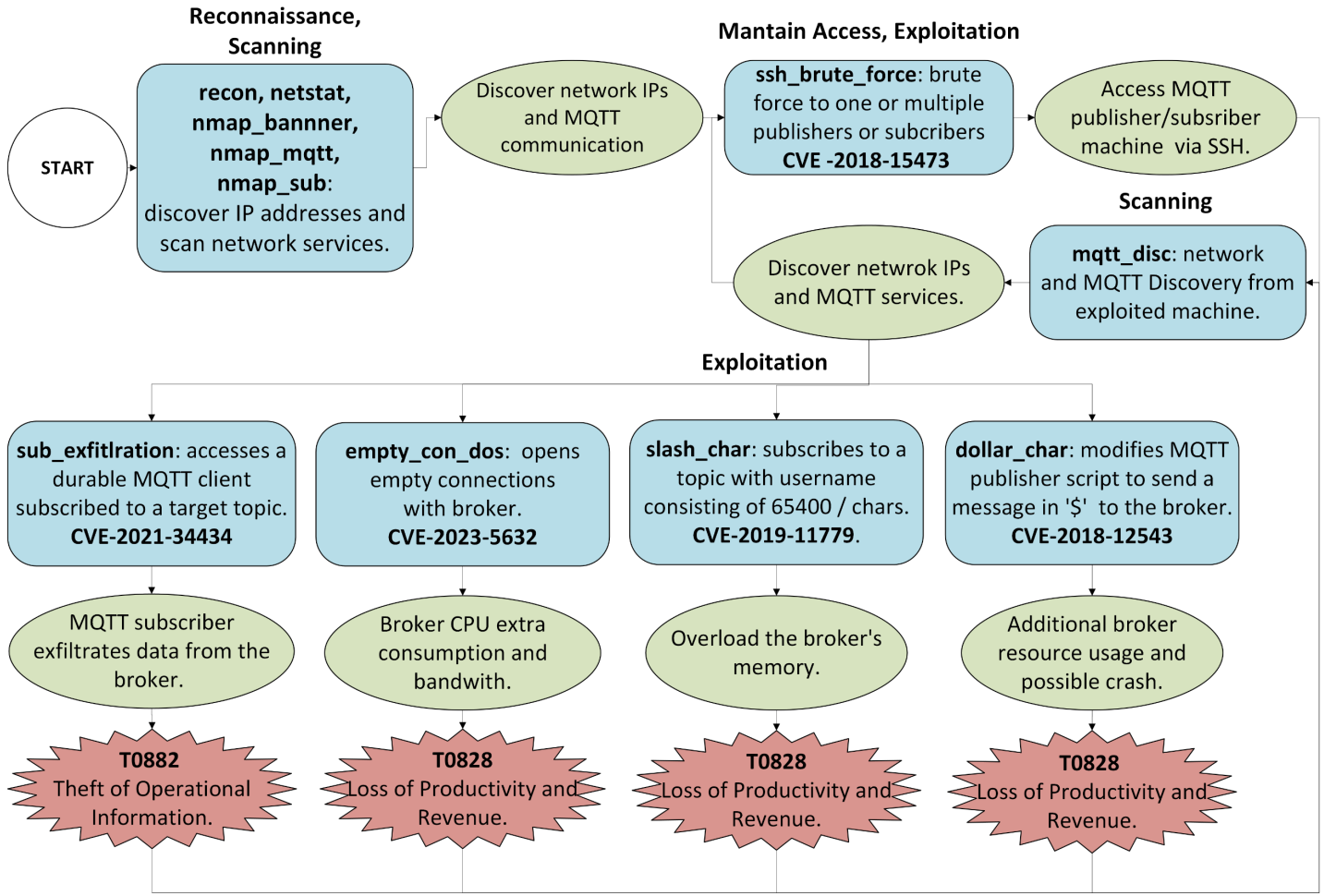


[2] Alatram, A., Sikos, L.F., Johnstone, M., Szewczyk, P., Kang, J.J.: DoS/DDoS-MQTT-IoT: A dataset for evaluating intrusions in IoT networks using the MQTT protocol. Comput. Netw. 231, 109809 (2023). https://doi.org/10.1016/j.comnet.2023.109809.

Define APT Scenarios

Define APTs:

- We define an attack graph based on the MITRE ATT&CK [3].
- Each attack step executes a real exploit from CVE [4].
- We select 4 different attack paths from the attack graph, i.e., we combine different attack steps.

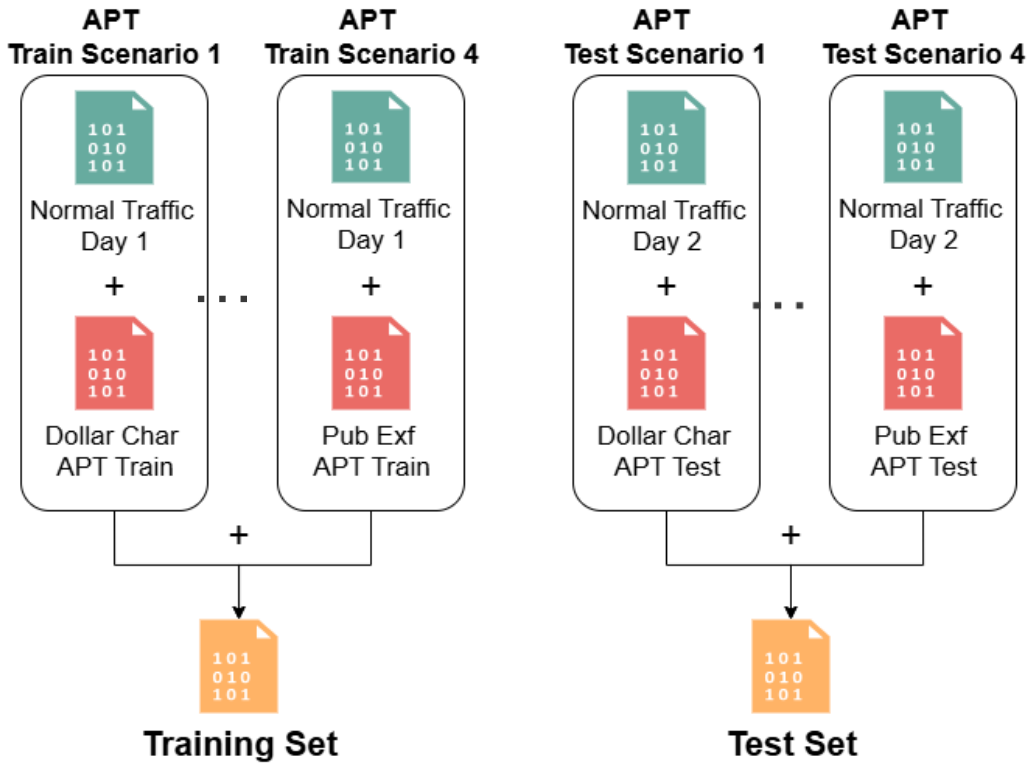


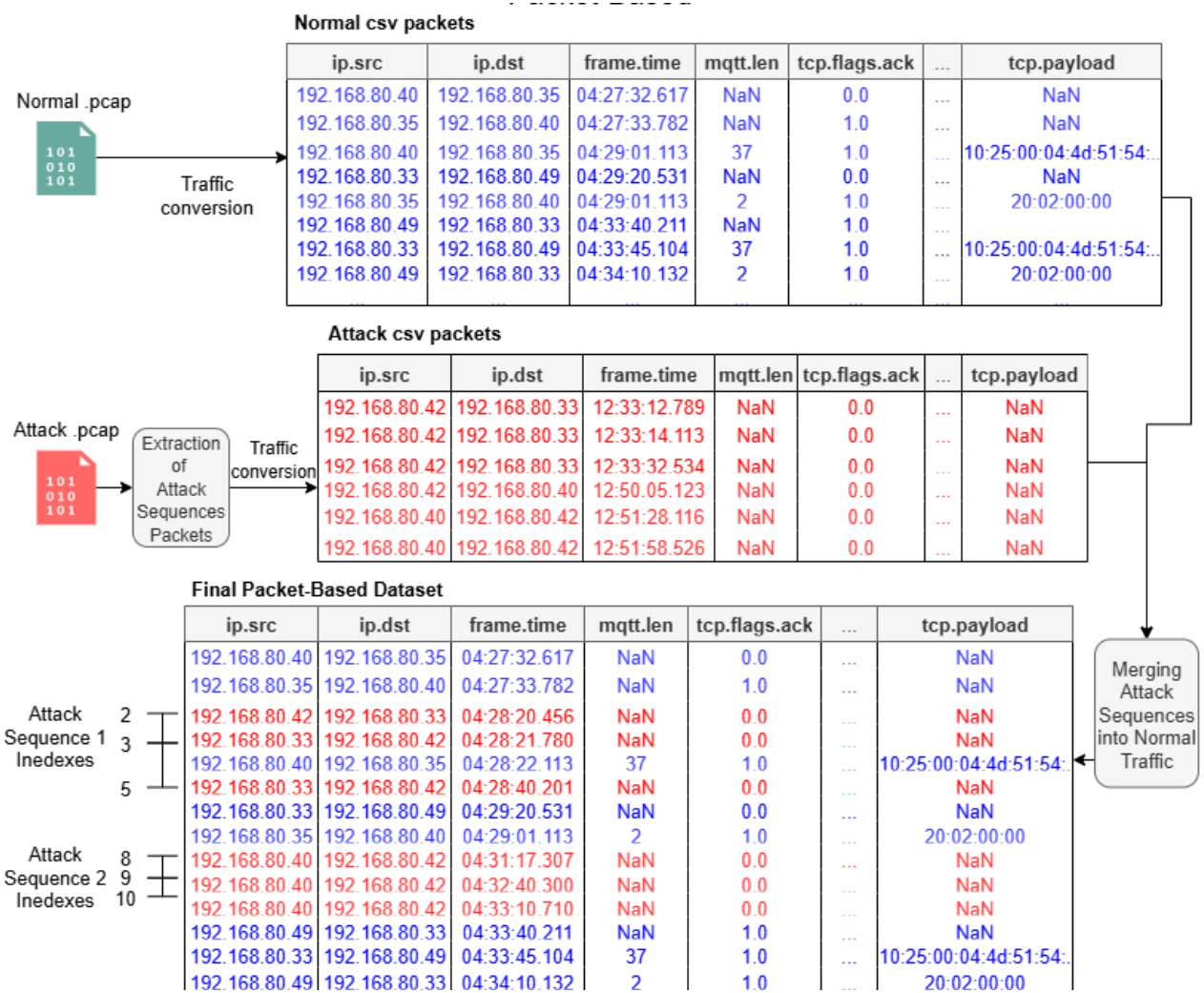
[3] MITRE ATT&CK - https://attack.mitre.org/
[4] CVE - https://cve.mitre.org/

Simulate APT Scenarios

Timestamp	Attack Phase	Step
04:27:32.802	Reconnaissance	**nmap_banner**
04:29:20.175		pause
04:33:15.278		**nmap sub**
04:33:45.482		pause
04:36:12.122		**nmap_banner**
04:38:12.122		pause
04:41:45.014	**Maintain access**	**ssh_bruteforce**
05:25:23.456		pause
05:27:10.446	Scanning	**mqtt_disc**
05:27:57.122		pause
05:28:51.111		**mqtt_disc**
05:32:38.221		pause
05:34:36.765	**Exploitation**	**dollar_char**
05:38:38.821		pause
05:40.25.669		**dollar_char**

Compose the Dataset





ip.src	ip.dst	frame.time	mqtt.len	tcp.flags.ack	...	tcp.payload
192.168.80.40	192.168.80.35	04:27:32.617	NaN	0.0	...	NaN
192.168.80.35	192.168.80.40	04:27:33.782	NaN	1.0	...	NaN
192.168.80.40	192.168.80.35	04:29:01.113	37	1.0	...	10:25:00:04:4d:51:54:..
192.168.80.33	192.168.80.49	04:29:20.531	NaN	0.0	...	NaN
192.168.80.35	192.168.80.40	04:29:01.113	2	1.0	...	20:02:00:00
192.168.80.49	192.168.80.33	04:33:40.211	NaN	1.0	...	
192.168.80.33	192.168.80.49	04:33:45.104	37	1.0	...	10:25:00:04:4d:51:54:..
192.168.80.49	192.168.80.33	04:34:10.132	2	1.0	...	20:02:00:00

ip.src	ip.dst	frame.time	mqtt.len	tcp.flags.ack	...	tcp.payload
192.168.80.42	192.168.80.33	12:33:12.789	NaN	0.0	...	NaN
192.168.80.42	192.168.80.33	12:33:14.113	NaN	0.0	...	NaN
192.168.80.42	192.168.80.33	12:33:32.534	NaN	0.0	...	NaN
192.168.80.42	192.168.80.40	12:50.05.123	NaN	0.0	...	NaN
192.168.80.40	192.168.80.42	12:51:28.116	NaN	0.0	...	NaN
192.168.80.40	192.168.80.42	12:51:58.526	NaN	0.0	...	NaN

	ip.src	ip.dst	frame.time	mqtt.len	tcp.flags.ack	...	tcp.payload
	192.168.80.40	192.168.80.35	04:27:32.617	NaN	0.0	...	NaN
	192.168.80.35	192.168.80.40	04:27:33.782	NaN	1.0	...	NaN
2	192.168.80.42	192.168.80.33	04:28:20.456	NaN	0.0	...	NaN
3	192.168.80.33	192.168.80.42	04:28:21.780	NaN	0.0	...	NaN
	192.168.80.40	192.168.80.35	04:28:22.113	37	1.0	...	10:25:00:04:4d:51:54:..
5	192.168.80.33	192.168.80.42	04:28:40.201	NaN	0.0	...	NaN
	192.168.80.33	192.168.80.49	04:29:20.531	NaN	0.0	...	NaN
	192.168.80.35	192.168.80.40	04:29:01.113	2	1.0	...	20:02:00:00
8	192.168.80.40	192.168.80.42	04:31:17.307	NaN	0.0	...	NaN
9	192.168.80.40	192.168.80.42	04:32:40.300	NaN	0.0	...	NaN
10	192.168.80.40	192.168.80.42	04:33:10.710	NaN	0.0	...	NaN
	192.168.80.49	192.168.80.33	04:33:40.211	NaN	1.0	...	NaN
	192.168.80.33	192.168.80.49	04:33:45.104	37	1.0	...	10:25:00:04:4d:51:54:..
	192.168.80.49	192.168.80.33	04:34:10.132	2	1.0	...	20:02:00:00

The dataset

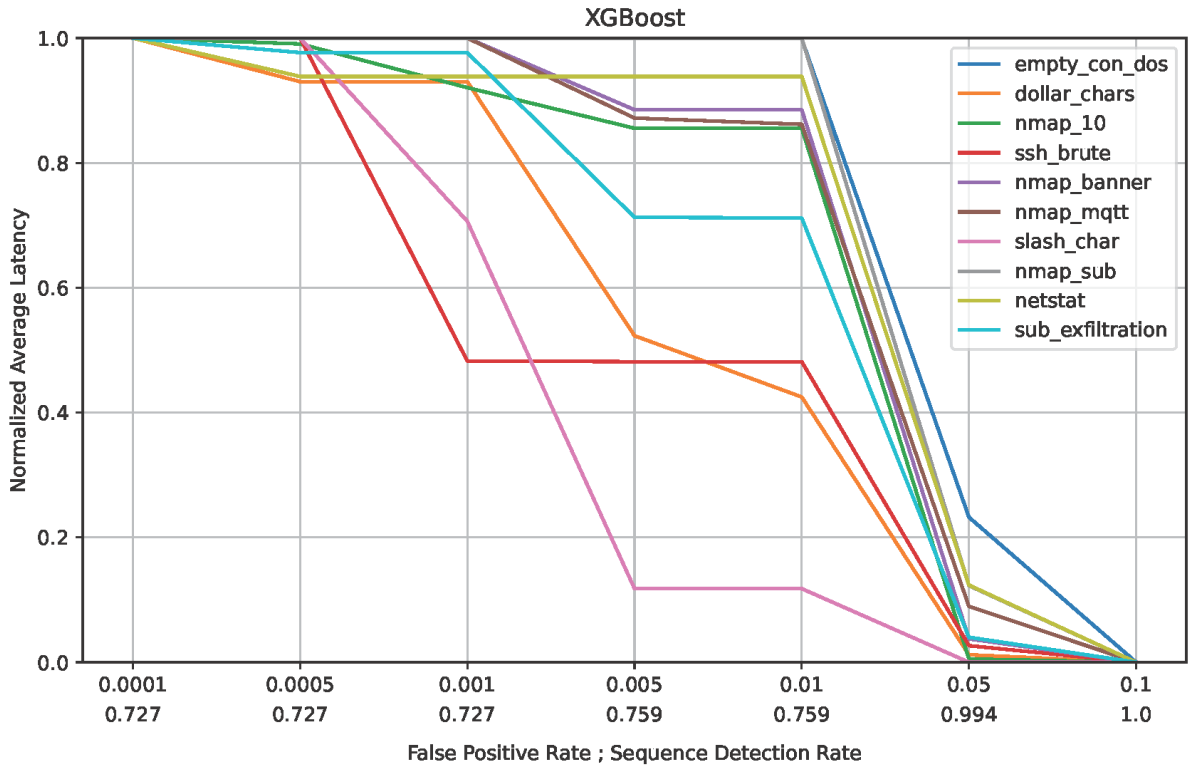
Table 1: For each attack step, we report the number of repetitions (iter column), the duration of repetition in seconds, and its length (number of packets).

attack	iter	average duration	minimum duration	maximum duration	average length	minimum length	maximum length
empty_con_dos	83	149.68	84.71	524.74	675.05	30	2120.5
dollar_char	71	601.5	86	1260.97	5468.16	637	11448
nmap_10	15	1040.36	1034.69	1045.12	44417.37	43927	44720
ssh_brute	24	140.41	118.72	194.36	2219.91	110	2706
nmap_banner	24	253.37	244.99	258.76	1181.47	612	2266
nmap_mqtt	24	258.95	250.69	271.18	1001.65	87	2378
slash_char	60	21.05	14.6	26.15	537.3	395	721
nmap_sub	10	61.66	55.65	67.11	627.5	237	770
netstat	28	56.84	38.70	67.88	233.27	22	370
sub_exfiltration	10	18.23	13.24	20.71	2355.6	2224	2425

Results

- We train XGBoost using the dataset.
- **False Positive Rate (FPR) = 0.001**.

Attack step sequences	R	F1	SDR	ΔL
empty_con_dos	0.43	0.60	0.53	69.6
dollar_char	0.70	0.82	0.90	101.8
nmap_10	0.94	0.97	0.75	375.3
ssh_brute	0.64	0.78	0.91	15.1
nmap_banner	0.24	0.38	0.36	162.8
nmap_mqtt	0.16	0.28	0.36	166.6
slash_char	0.36	0.53	1.00	4.9
nmap_sub	0.28	0.44	0.60	26.1
netstat	0.48	0.65	0.64	25.1
sub_exfiltration	0.95	0.97	0.66	10.6
all	**0.80**	**0.89**	**0.72**	**95.8**



Conclusions

- We propose a methodology to realize a semi-synthetic APT detection dataset
- We define the APT attack by composing different attack steps that resemble the APT lifecycle
- We provide the **indices for accessing the packets belonging to each attack step** iteration, allowing for computing the attack latency and sequence detection rate.
- This allows us to **measure the probability and the time required to interrupt an attack step**. This is important when defending from APT as it quantifies the ability of the defenses to avoid the attacker completing the attack graph and reaching its goal.

Future Work

- Compose a refined version of the dataset:
  A. The attack steps will be enriched.
  B. The time of the simulation will be augmented.
- Study of **detection algorithms** and **feature augmentation** techniques that can leverage the structure of the dataset.
- Combines the probability of detecting attack steps to compute the probability of detecting the whole APT and the related **attack latency**.

Questions