Ongoing work on synthetic network traffic generation for IDS evaluation

Pierre-François Gimenez

Superviz meeting – March 11th

Information system security

Information system security

- Prevent the attack, detect it, and react
- Detection with **IDS**: *Intrusion Detection System*

Detection relies on observation

- System : OS and applications logs
- Network : network communications

Constraints

- Partial and heterogeneous observations
- Adversarial context: the attacker hides!

```
2024-05-06T23:24:16.806598+02:00
stellar-sheep sshd[16039]: Failed
password for pfg from 192.168.1.36
port 48650 ssh2
```


```
"ts": 1591367999.305988,
"id.orig_h": "192.168.4.76",
"id.resp_h": "192.168.4.1",
"id.resp_p": 53, "proto": "udp",
"service": "dns", "duration":
0.066851, "orig_bytes":
62, "resp_bytes": 141,
"conn_state": "SF", "orig_pkts":
2, "orig_ip_bytes": 118,
"resp_pkts": 2, "resp_ip_bytes":
197
```

The issue of data in security

Why do we need data?

- For evaluating security measures, most notably detection
- For using machine learning in cybersecurity

Current state of datasets

- Public datasets are typically run in testbed with no real users
- They can suffer from mislabelling, network and attack configurations errors, etc.
- We cannot access private data due to confidentiality and privacy reasons

$\Rightarrow$ we cannot confidently evaluate intrusion detection systems because of this dubious quality

My research project: **use AI to generate security data**

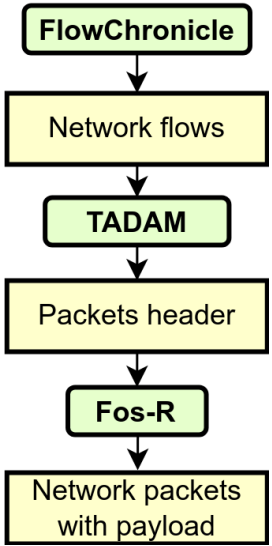
Approach

State of the part

- Several approaches have been tried to generate network flow records or pcap files: VAE, GAN, LLMs
- The results are not very good:
  - A significant portion of generated data do not comply with network protocols
  - Generated data do not reflect the diversity of the original data

Our approach: a three-step generation

- FlowChronicle (published): a network flow generator
- TADAM (accepted): a packet header generator
- Fos-R (ongoing work): full packet generator



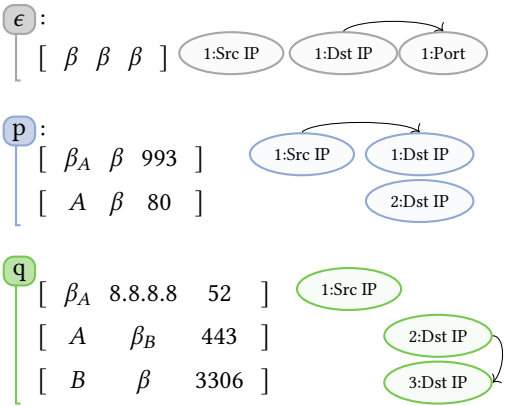
FlowChronicle (CoNEXT'24)

General idea

- Joint work with Joscha Cüppers from CISPA in the context of SecGen
- General approach: find patterns in the data and use them to generate new data
- We focus on temporal patterns of flows
  - DNS query then HTTP(S)
  - IMAP request then HTTP(S)
- These patterns are self-explanatory:
  - they can be verified by an expert
  - they can also be added manually

FlowChronicle

Model — Pattern and Bayesian Network:



Data and Pattern Windows:

Time	Src IP	Dst IP	Port
12	134.96.235.78	142.251.36.5	993
56	134.96.235.129	8.8.8.8	52
89	134.96.235.78	212.21.165.114	80
113	134.96.235.129	198.95.26.96	443
145	198.95.26.96	198.95.28.30	3306
156	134.96.235.78	134.96.234.5	21
178	134.96.235.36	185.15.59.224	993
206	134.96.235.36	128.93.162.83	80

Pattern Description

Pattern language

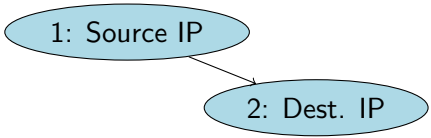
Each pattern has two part: a partially defined flow, and a Bayesian network

- Fixed values are defined in the partial flow
- the distribution of Free variables is defined in the Bayesian network
- Reused variables are always equal to some Free variable

Partial flows

Source IP	Dest. IP	Dest. Port
β_A	8.8.8.8	53
A	β	80

Bayesian Network



In reality there are more columns!

Data quality evaluation

Hard to evaluate

- No standard metrics
- Evaluation often partial

Proposition

A set of evaluating metrics:

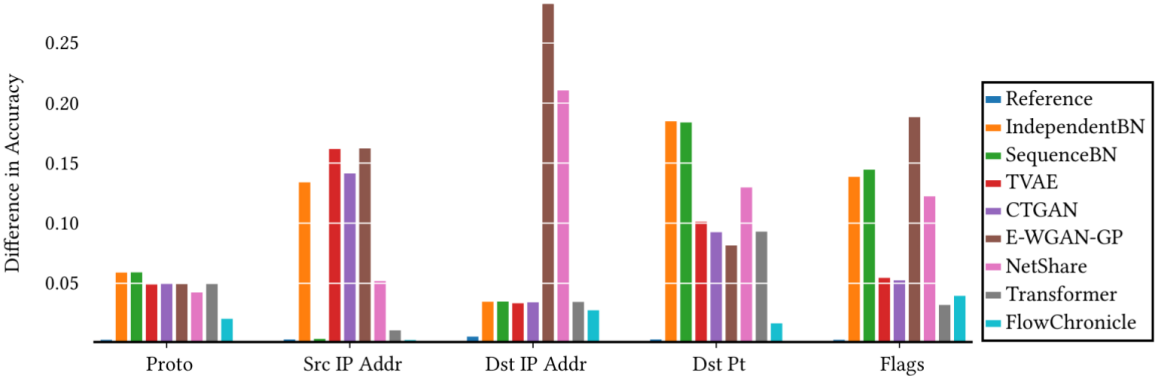
Realism	:	Are the generated data part of the target distribution?
Diversity	:	can any point in the target distribution be generated?
Novelty	:	can the generator create data absent from the training set?
Compliance	:	do the generated data comply with the technical specifications?

We do not consider privacy yet

FlowChronicle: generation quality

	Density	CMD	PCD	EMD	JSD	Coverage	DKC	MD	Rank
	Real. ↑	*Real.* ↓	*Real.* ↓	*Real./Div.* ↓	*Real./Div.* ↓	*Div.* ↑	*Comp.* ↓	*Nov.* =	*Average Ranking*
Reference	**0.69**	**0.06**	**1.38**	**0.00**	**0.15**	**0.59**	**0.00**	**6.71**	-
IndependentBN	0.24	0.22	2.74	*0.11*	0.27	0.38	0.05	5.47	5.25
SequenceBN	0.30	**0.13**	2.18	0.08	0.21	0.44	**0.02**	5.51	3.875
TVAE	0.49	0.18	1.84	**0.01**	0.30	0.33	0.07	5.17	4.125
CTGAN	**0.56**	0.15	**1.60**	0.01	**0.15**	**0.51**	*0.11*	**5.70**	**3.0**
E-WGAN-GP	*0.02*	0.34	*3.63*	0.02	0.38	*0.02*	0.07	4.66	7.0
NetShare	0.32	0.28	**1.47**	0.03	0.36	0.22	0.05	3.82	5.25
Transformer	**0.62**	*0.78*	3.62	**0.00**	*0.55*	0.03	0.05	*3.75*	*5.375*
FlowChronicle	0.41	**0.03**	2.06	0.02	**0.10**	**0.59**	**0.02**	**5.87**	**2.125**

FlowChronicle: temporal generation quality



Data generated with FlowChronicle

Output of FlowChronicle

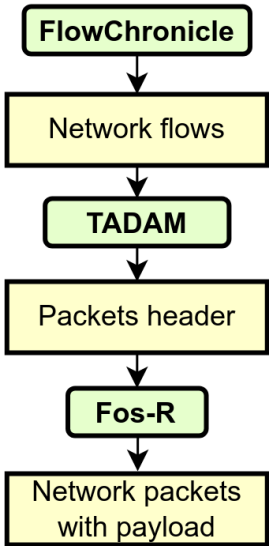
- FlowChronicle outputs network flow records, e.g:

```
ts,proto,src_ip,dst_ip,dst_port,fwd_pkts,bwd_pkts,fwd_bytes,bwd_bytes
1730800143,TCP,131.254.252.23,216.58.213.78,443,33,41,5988,1950
```

- But in the end, we want to generate packets!

Next intermediary step

- Before generating complete packets, we propose to first generate an intermediate representation
- More precisely, we generate for each packet a tuple with:
  - the direction (forward or backward)
  - the TCP flags
  - the size of the payload
  - the time since the last packet (i.e., the inter-arrival time)



TADAM (SDM'25)

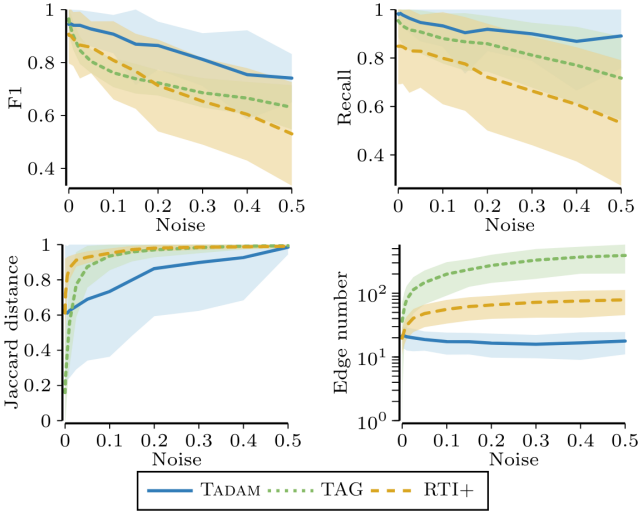
Learning

- Network protocols typically rely on finite state automata
- We propose to learn probabilistic timed automata to capture packet header sequences
- Existing automata learners from observations cannot handle noisy data
- We propose TADAM: a robust timed automata learner
- Two main contributions:
  - A compression-based score to avoid overfitting
  - An explicit modelization of the noise

Experimental results

- TADAM is far more robust to noise
- TADAM learns smaller models
- TADAM has better performance on real-world classification and anomaly detection tasks

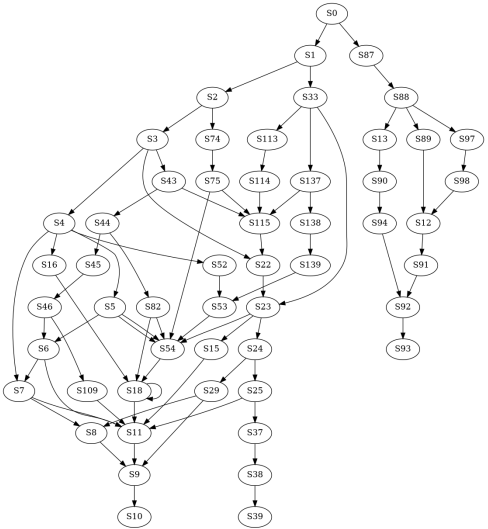
TADAM: experiments



Learner	AU-ROC	TPR	FPR	F1
TADAM	**0.982**	0.998	**0.025**	**0.705**
TAG	0.891	**1**	0.142	0.298
RTI+	0.790	**1**	0.292	0.171
HMM	0.608	0.640	0.085	0.288

Table 3: *Anomaly detection performance on HDFS_v1 dataset.* We report the TPR, FPR and F1-score for the threshold maximizing TPR-FPR.

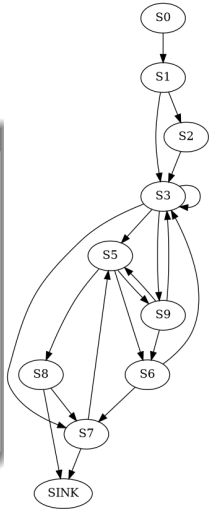
Example: Kerberos protocol



TAG

And for network protocols?

- We limit the observations to some data: TCP flags, direction, size and inter-arrival time
- In particular, we do not look at the payload, so no perspective on the semantics of the message
- In practice, it's not easy to interpret them



TADAM

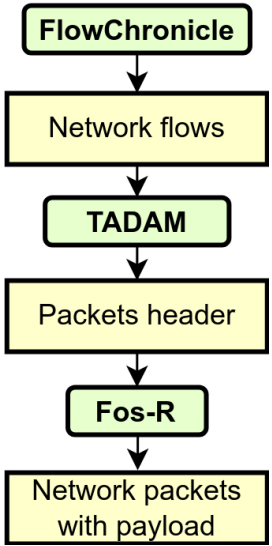
Data generated with TADAM

Output of TADAM

TADAM outputs tuples, e.g: (FWD, SYN, 0, 0), (BWD, SYN/ACK, 0, 2), (FWD, ACK, 0 3), (FWD, PUSH, 123, 10), ...

Fos-R: bridging the gaps

- Fos-R has a linear algorithm to sample from an automata according to constraints from FlowChronicle generation (the number of forward and backward packets in a flow)
- It creates the full packets:
  - The rest of the header is creating according to some rules (window size, checksum, etc.)
  - For now, the payload is replayed or random



Fos-R

Faster generation

Fos-R is a new, faster implementation in Rust (Python was too slow) with three modes:

- Static pcap creation
- Pcap replay on network *work in progress*
- Honeynet mode: the flow are played on the network without communication overhead, for honeynet and cyber range (deployed for BreizhCTF2025). Packet tainting with "evil bit"

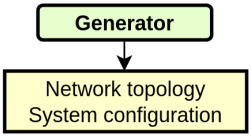
Challenges

Mostly engineering challenges that are mostly solved:

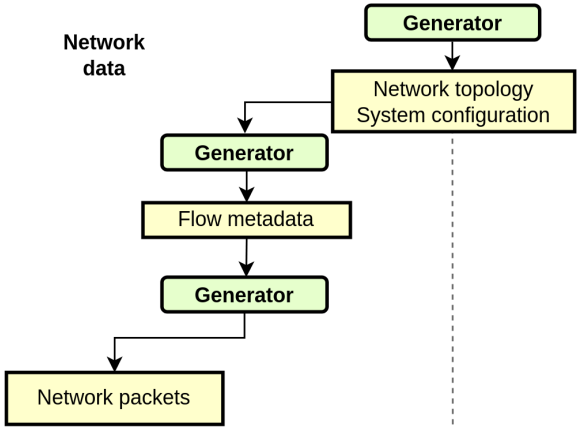
- How to ensure determinism between agents?
- How to parallelize the generation?
- How to make the kernel not interfere with the communications?

Demo time!

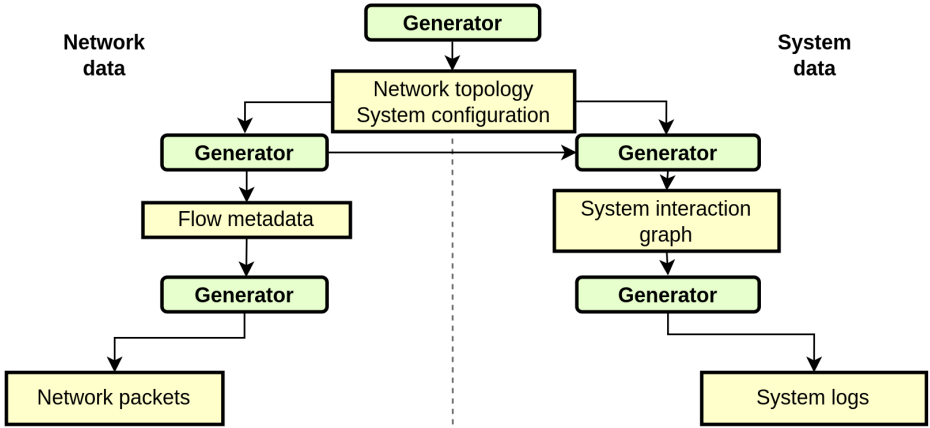
The next steps of my research project



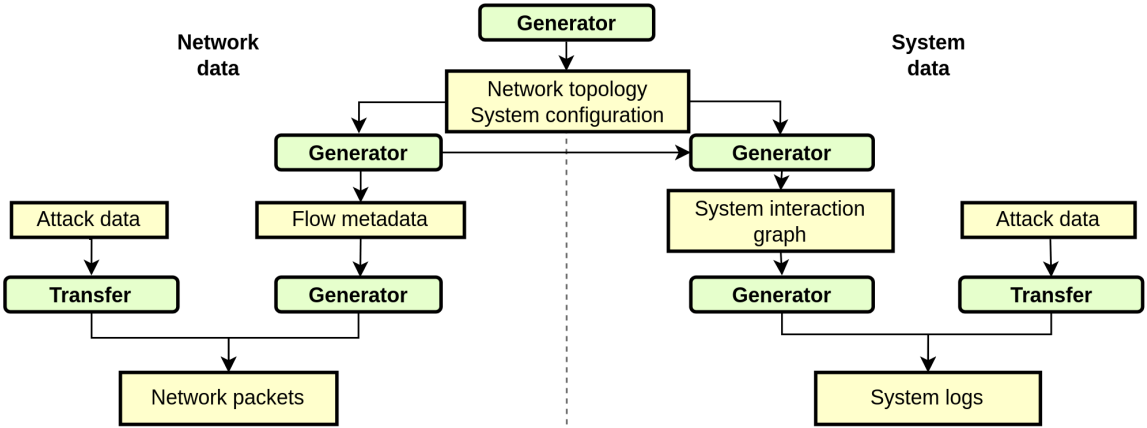
The next steps of my research project



The next steps of my research project



The next steps of my research project



Conclusion

The need of data

- Good quality data is of utmost importance for security system evaluation
- One way to achieve such quality is through generative AI

My research project

- Our experiments so far show that better generation quality with frugal & explainable AI than with deep learning
- Fos-R roadmap is available on `https://crates.io/crates/fosr`. Wait until Q4 2025 before testing.
- We will probably start a PhD on system data generation in 2025
- Beyond data generation: my long-term goal is to create an interactive, synthetic environment to learn and evaluate RL-based reaction to attacks