RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Inria

A look back at the SecGen associate team

Collaboration Inria-CISPA from 2022 to 2025

Pierre-François Gimenez, Inria researcher

Context

Context

Who am I?

- PhD in machine learning in 2018
- Now a researcher at Inria in Rennes
- Research keywords: anomaly-based intrusion detection, synthetic data generation

Collaboration with CISPA

- I visited Mario Fritz's research group for 3 months in 2022
- Mario and I coordinated an "associate team" between 2023 and 2025
- SecGen focuses on synthetic data generation and anomaly detection
- Website `https://files.inria.fr/secgen/`

Associate Team program

What is it?

- A way to quick-start collaborations between Inria and labs abroad
- A mobility funding from Inria's international relations management
- $\sim$ 10000€ per year for three years
- A lightweight application file
- The perfect tool to explore collaboration with minimal risk

How much can we do with a "small" funding?

Scientific output

Scientific stays

- Adrien Schoen: 2 months at CISPA
- Joscha Cüppers: 2 months at Inria
- Lénaïg Cornanguer: 1 week at Inria

SecGen's achievements

- PF Gimenez, S Sivaprasad, M Fritz, *Certifiably robust malware detectors by design*, at IFIP SEC (B-rank), 2025
- J Cüppers, A Schoen, G Blanc, PF Gimenez, *Flowchronicle: synthetic network flow generation through pattern set mining*, at CoNEXT (A-rank), 2024, best paper
- L Cornanguer, PF Gimenez, *TADAM: Learning Timed Automata from Noisy Observations*, at SDM (A-rank), 2025

The team

The team

From Inria

- **Pierre-François Gimenez** (PI, tenured researcher)
- Frédéric Majorczyk (PhD, engineer)
- Adrien Schoen (PhD student)

From CISPA

- **Mario Fritz** (PI, faculty)
- Jilles Vreeken (faculty)
- Lénaïg Cornanguer (post-doc)
- Sarath Sivaprasad (PhD student)
- Joscha Cüppers (PhD student)

Scientific Contributions

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

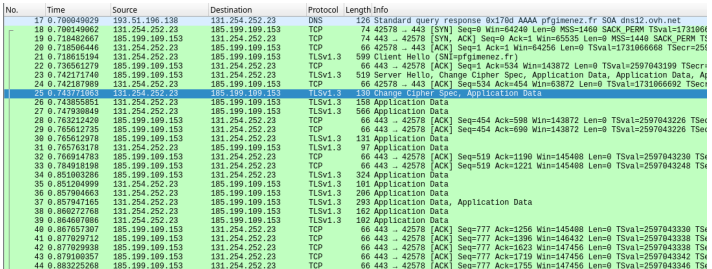
$\Rightarrow$ we cannot confidently evaluate intrusion detection systems because of this dubious quality

The goal of SecGen: **use AI to generate security data**

Network data example

Network data

- Raw data consist of packets, regrouped in conversation
- Cybersecurity analysis typically rely on network flow records that describe conversations statistically




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

Synthetic network traffic generators

What is synthetic data?

Data generated without simulation or emulation

Why synthetic data?

Some advantages:

- Much faster generation
- Allows to quickly iterate over dataset quality
- Ensure generation replicability
- Can have better scalability

But generating high-quality data is difficult

Related work

What generation techniques are used?

Deep learning techniques:

- Many, many GAN (generative adversarial network) and variations
- VAE (variational auto-encoder) and variations
- Diffusion models
- LLM (of course)

Other techniques:

- SMOTE
- Bayesian networks

In our approach, we focus on **statistical, explainable** AI techniques for **fast, trustworthy** generated data

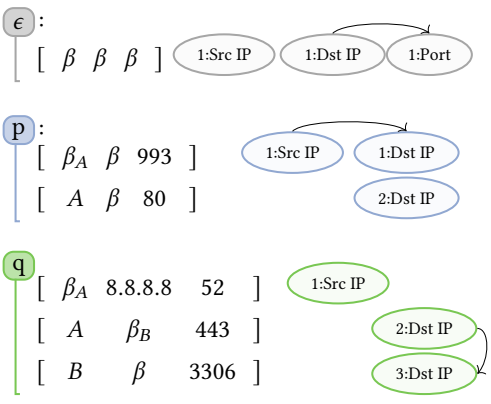
FlowChronicle (CoNEXT'24, best paper)

General idea

- Joint work with Joscha Cüppers
- 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

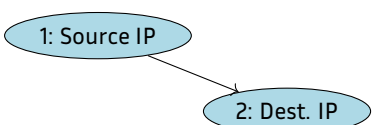
Each pattern has two part: a partially defined flow, and a Bayesian network

- 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:

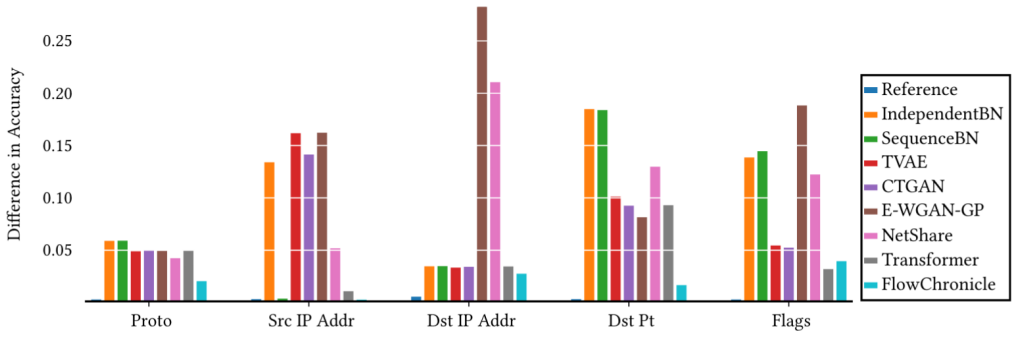
- Realism : Are the generated data part of the target distribution?
- Diversity : can any point in the target distribution be generated?
- Novelty : can the generator create data absent from the training set?
- Compliance : do the generated data comply with the technical specifications?

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!

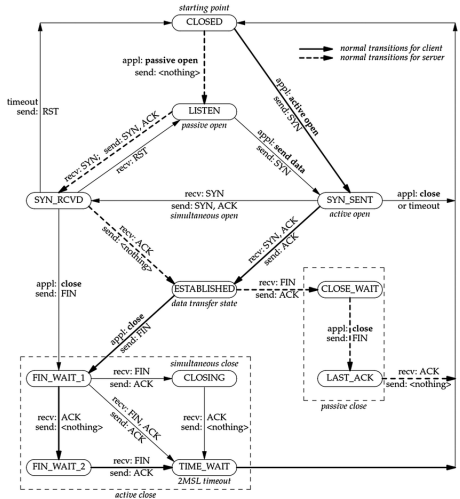
Next step

- 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)

Automata learning

Why not use the specification?

- Network protocols typically rely on finite state automata
- Official automata from RFC do not tell about the actual behavior
- Not all protocols are documented with a finite state automaton
- It is very easy to sample new sequences from a probabilistic automata



State of the art

Automaton learning

- Active learning: requires a "teacher" to verify the automaton during learning
- Passing learning: requires observations
  - Learning from positive and negative examples
  - Learning from positive examples only (TAG, RTI+)

Limitations of TAG and RTI+

- By design, all observations will be included in the automaton
- For example, if a client cannot connect and send many "SYN" packets to the server, this trace will be included in the protocol automaton
- But network data can be noisy! So it leads to bloated automata

We introduce TADAM, an automata learning that is robust to noise

Learning process

Learning process

- We start with an automaton initialized like a Markov chain
- We apply elementary operations (split a node or an edge, merge two nodes, delete an edge)
- We remove the noise of the data so it fits the model
- We evaluate the new automata with a **score** and keep the best

Learning process

Learning process

- We start with an automaton initialized like a Markov chain
- We apply elementary operations (split a node or an edge, merge two nodes, delete an edge)
- We remove the noise of the data so it fits the model
- We evaluate the new automata with a **score** and keep the best

The score (intuition)

This an MDL score based on compression: $L(D, A) = L(D \mid A) + L(A)$

- $L(D, A)$ is the score of automaton A for some data D
- $L(D \mid A)$ is the length of describing the data encoded with the automaton
- $L(A)$ is the length of describing the automaton

The full score

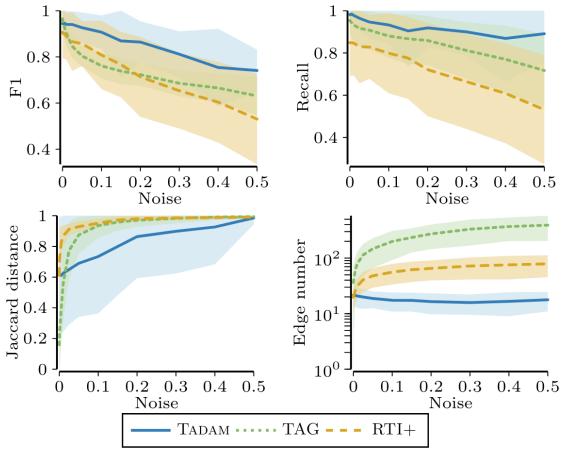
$$L(A) = L_{\mathbb{N}}(|\mathcal{Q}|) + L_{\mathbb{N}}(|\Sigma|) + \sum_{e \in \mathcal{E}} \left(2\log_2(|\mathcal{Q}|) + \log_2(|\Sigma|) + L_{\mathbb{N}}(\lfloor \mu_e \rfloor) + L_{\mathbb{N}}(\lfloor \sigma_e^2 \rfloor) \right) + 2\log_2(|\mathcal{Q}|)$$

$$L(D|A) = \sum_{(o,e,d) \in R} -\log_2(p(o)) + \sum_{(o,e,d) \in R_{\{\mathrm{tr,fo,ad}\}}} -\log_2 p(e \mid q_s(e))$$
$$+ \sum_{(o,e,d) \in R_{\{\mathrm{tr,fo,de}\}}} -\log_2 p(d \mid e) + \sum_{(o,e,d) \in R_{\{\mathrm{sk}\}}} (L_{\mathbb{N}}(d) + \log_2(|\Sigma|))$$

The originality is that we explicitely model the noise. Train set can have:

- missing packets (due to concurrent routes)
- swapped packets (due to system latency in probes)
- duplicated packets (due to TCP retransmission)

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.

Conclusion

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
- We achieved very promising results

A look back at SecGen

- Overall, a very positive experience
- High-quality scientific articles, a ”best paper” award
- SecGen reached a natural conclusion with these works
- While SecGen ended, I still build upon these works