RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Inria

Rust for Research: Feedback from the Fos-R Project

Rust Paris 2026

Pierre-François Gimenez, Inria

A quick presentation

Who am I?

- Researcher at Inria, a national research institute, in the PIRAT team
- PhD in machine learning in 2018
- Research in AI for cybersecurity since then

Research topics of my team

- Malware analysis (Windows and Android)
- Intrusion detection (EDR, NDR)
- Automatic vulnerable infrastructure deployment (digital twin, honeynets and CTF)

Software in research

Relation between software development and research

- Fundamental research with low TRL (1 to 4)
- We develop prototypes to validate ideas
- Software are almost always open source to allow reproducibility
- The output are the ideas and the experimental evaluation, the software implementation is a secondary deliverable
- The vast majority of our software has no update after initial release

The current practices

- Python is by far the most used language
- We also used a few more specific languages: R, OCaml, Rocq (Coq), etc.

Research project: Fos-R

Scientific context: data is key

- We need data to train ML/DL models
- We need data to evaluate our tools
- Background traffic for more realistic honeypot and CTF

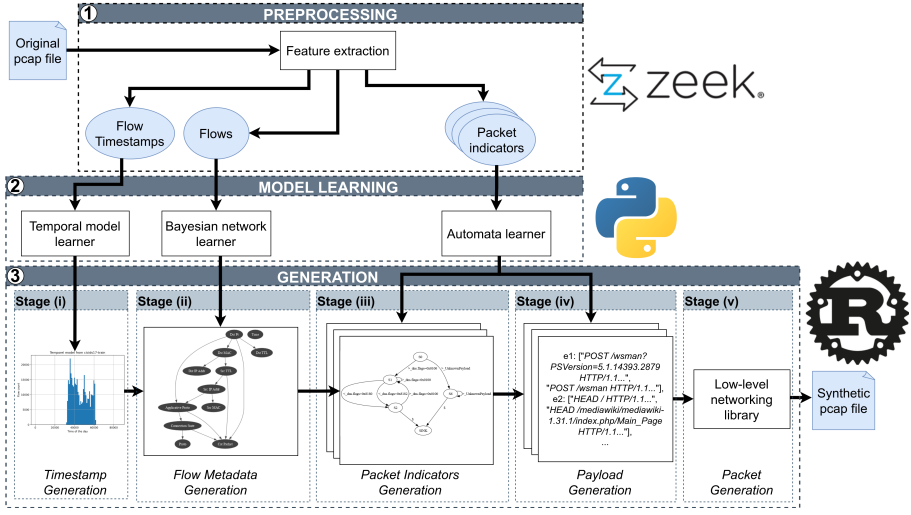
Sources of data

- We rely on public datasets created by research institutions
- They have many issues: obsolescence, errors, lack of realism, etc.
- It makes evaluation difficult: how to be sure our ideas are the correct ones?

Fos-R

I develop Fos-R, a software that generate network traffic (pcap) files with machine learning models, and inject it in a network

Fos-R: inner working overview



Fos-R specification

Functional specification

- Traffic generation with machine learning models
- Low-level networking
- Kernel-based routing
- User interface

Non-functional specification

- High performances
- Reproducibility
- Portability (Linux and Windows, ARM and x86)
- Easy to test without downloading anything

1. Rust popularity ☹

Why is it important?

All people working on a project must master its language

In research

- Many PhD students and engineers do not know Rust
- Most of our team have CDD (PhD students, most engineers, etc.), so many people work on a project
- More and more are learning it, but they are still in the minority

For Fos-R

- I am the main developer and I reimplement the work of PhD students
- I teach Rust at CentraleSupélec since 2020
- I can propose students to work on Fos-R as part of their teaching
- Luckily we have a Rust senior research engineer to help us!

2. Prototype-friendly ☹

Why is it important?

Agile method require quick iterations

In research

- In research we iterate at lot on our ideas (typically one iteration per week)
- Hackability is more important than maintability

For Fos-R

- Prototyping is mostly done in Python
- A modular architecture with traits allows for easier change of individual component

3. Library ecosystem 😐

Why is it important?

Library allows for faster development

In research

It highly depends on the types of library:

- There is much fewer ML library than is Python
- Libraries are less mature: their API are more unstable and they do not have as much features
- But there are some great library (e.g., GUI library), and even a full Rust SMT solver

For Fos-R

- Model learning is written in Python
- Inference is written in Rust
- I had to reimplement several machine learning algorithms manually
- I had to hack some libraries to use them

4. Dependency management ☺

Why is it important?

Anyone should be able to easily compile our code

In research

- Python is a mess, and code breaks quickly due to non-pinned dependency, breaking changes somewhere, and no unified project management system
- It makes projects very difficult to reproduce, which hamper comparison between methods and overall scientific progress

For Fos-R

- Cargo is a fantastic tool, crates.io is very convenient
- We know it won't break randomly in two months
- Features with conditional dependency is very useful as well

5. Cross-compilation ☺

Why is it important?

Cross-compilation allows to distribute easily binaries for multiple platforms

In research

We generally do not require cross-compilation

For Fos-R

- Fos-R requires cross-compilation to be deployed on Linux and Windows honeypots
- We cross-compile to x86 and ARM platform (e.g., Raspberry Pi)
- We also export to WASM for easy demo without downloading anything
- WASM export requires conditional compilation however

6. Performances ☺

Why is it important?

Nobody likes to wait, and RAM is too expansive to waste :)

In research

Prototypes are not optimized: as long as it takes a reasonable time, it is enough

For Fos-R

- State-of-the-art methods are really slow, so a fast generation is a way to differentiate
- We changed the default Linux allocator for a 5× improvement
- This is in strong contrast with Python

7. System programming language ☺

Why is it important?

Some projects require low level access to OS features

In research

Most research projects that require system programming use C or C++

For Fos-R

- Rust allows us to use raw sockets, to specify allocators, to use eBPF, etc.
- We need to use conditional compilation to disable some low-level features for the WASM export

8. Safety ☺

Why is it important?

Nobody wants to ship vulnerable software

In research

The software never leaves the lab: there is little to no security concern

For Fos-R

- Vulnerabilities are not considered
- But the borrow checker allows to identifies many issues during compilation
- If it compiles, there is a good chance it's correct
- Even if compilation is slower, iterations are faster
- Fearless concurrency is real: with the proper architecture, making Fos-R concurrent was effortless

Assessment for Fos-R

Functional specification

- Traffic generation with machine learning models ~ (manual reimplementation)
- Low-level networking ✓ (libpnet)
- Kernel-based routing ✓ (eBPF with Aya)
- User interface ✓ (egui)

Non-functional specification

- High performances ✓ (about 6.8 GB/s)
- Reproducibility ✓
- Portability (Linux and Windows, ARM and x86) ✓ (6 exports)
- Easy to test without downloading anything ✓ (WASM export)

Conclusion

Rust for Fos-R

- Overall, a very positive experience using Rust for Fos-R
- The biggest issue was a lack of ML models
- Part of the project is still in Python

Rust for research

- The drawbacks probably outweigh the benefits
- Some drawbacks may disappear (language popularity, ecosystem maturity)
- Lack of hackability is the most fundamental issue
- The hegemony of Python makes it easier to build on each other's work

Rust is helpful for my work, but I do not expect it to become popular within the computer science scientific community