Valentin **Manès**
Software Engineer

I am a Software Engineer with experience in all levels of projects, including design and architecture, development and test, and the setup of reliable production. Skilled at writing well designed low-level system programs using best practices in Go, C, C++. Fast learner, hard worker, and team player with flexibility using various tools. Dedicated to streamlining processes and efficiently resolving project issues in hand using the most adapted technology.

Contact
valentinmanes@outlook.fr

Website
jiliac.com 🌐

Links
@Jilyac 🐦
Jiliac 🎧
valentinmanes 💼

Programming
Go, **C/C++**
Java, Python, julia

Languages
French: Mother Tongue
English: Near Native
Spanish: Intermediate
Korean: Basic

Interests
Card Games
Languages
Books
Travel

Experience

2021   **Qonto**   Paris, France
Qonto is a European neobank for professionals. To improve a higher quality of service for its clients, in 2018 it developed its own "Core Banking System", meaning that it maintains itself all its clients' accounts and process all their transactions. (Beforehand it was relying on an external partner to do so.) At Qonto, I was part of the Ledger team, which maintains the "source of truth" for all accounts and their transactions. These micro-services were implemented using a **Go** plus **PostgreSQL** stack. Moreover, I did multiple interventions to improve and maintain Qonto's billing system which is implemented using Ruby on Rails. Qonto uses an advanced micro-service architecture with over 80 services being **continuously deployed** relying on **Gitlab**, **Kubernetes** and **Argo CD**.

2020   **PacketAI**   Paris, France
PacketAI aims to develop an IT infrastructure monitoring platform, similar to Datadog and Dynatrace, but equipped with Machine Learning to predict incidents in advance and locate their root cause. At PacketAI, I started adding new features to the agent, a software running on the client hosts collecting events and metrics. I designed and developed from scratch all PacketAI microservices, all in **Go**, plus a **Logstash** node.
- PacketAI product is based on the **ELK** stack: The **Beats** to produce data and **Logstash** to transform and forward it to ElasticSearch.
- Data is streamed using **Kafka** pipes.
- **Mentored** the integration of an intern to the team.

2016-19 **Cyber Security Research Center - KAIST**   Daejeon, South Korea
CSRC is a publicly-funded research center within KAIST university. I was free to define the problems I worked on, and figure potential solutions, then develop and design their implementation, and finally test and evaluate these prototypes.
I completed two large projects. First, an improvement of the dynamic testing tool of LLVM, a compiler infrastructure project written in **C++**. This project was merged into the mainline by a team at *Google*. And then, Ankou, my largest project, is a fuzzer I developed from scratch in **Go**.
- Experiments setup in **Docker** containers to be reproducible and scalable to multiple servers. Command-line tools are invaluable: `htop`, `grep`, `find`, etc…
- Ankou (described below), I started as an investigation on the usage of machine learning techniques to improve fuzzers bug finding ability. For this, standard python libraries were used: **Keras**, **TensorFlow**, **Numpy**, **Pandas**. The two parts of the project, in Go and Python, were communicating via **RabbitMQ**.

Education

2015-16 **KAIST - Exchange**   Daejeon, South Korea
KAIST is considered the "MIT of Korea". It was a very different studying environment than I was used to: more centered around research. In particular, I focused on kernel hardening techniques and software security.

2013-16 **Telecom ParisTech - Master's degree**   Paris, France
Telecom ParisTech is one of France's top three graduate science schools (*grandes écoles*), and is considered the leading French school in Information and Communication Technology. I specialized in Information Security.

2011-13 **Lakanal - Preparatory School**   Sceaux, France

2006-11 **Lycée Franco-Méxicain**   Mexico City, Mexico

Publications

2020 **Boosting Fuzzer Efficiency: An Information Theoretic Perspective**
Foundations of Software Engineering (Second Author)
Code: github.com/llvm/llvm-project/commit/e2e38fca
Entropic is an information-theoretic power schedule implemented based on LibFuzzer. It boosts performance by changing weights assigned to the seeds in the corpus. Seeds revealing more "information" are assigned a higher weight. Entropic has been independently evaluated by a team at Google and invited for integration into mainline LibFuzzer @ LLVM (C++ code base), whereupon Entropic was subject to a substantial code reviewing process.

2020 **Ankou: Guiding Grey-box Fuzzing towards Combinatorial Difference**
International Conference on Software Engineering
Code: github.com/SoftSec-KAIST/ankou
Grey-box fuzzing search process is not expressive enough because it does not take *combinations* of software features into account. We propose a way to account for combinations. However, it is too computationally expensive, thus we reduce the dimensionality of the problem via a modified version of the Principal Component Analysis. This was a large engineering project: 15K lines of Go.

2019 **The Art, Science, and Engineering of Fuzzing: A Survey**
IEEE Transaction on Software Engineering
Companion website: fuzzing-survey.org
This survey presents a unified, general-purpose model. By identifying the key algorithmic stages of fuzzers, we could effectively summarize the literature.

2018 **Domain Isolated Kernel**
Elsevier Computer & Security
Code: github.com/Jiliac/DIKernel
Kernel extensions (i.e. drivers) are the weakest kernel part security-wise. DIKernel isolates extensions by lowering their memory access permission and their execution privilege. We keep our solution convenient for both the end-users, by ensuring a low-performance cost, and developers, by not requiring any change in the code of extensions. DIKernel was implemented on top of Linux 4.13 kernel with 1.5K lines of C.