Valentin Manès

Software Engineer

I am a Senior Software Engineer with expertise in full-stack development, focusing on JavaScript and Go. I excel at building robust backend services and modern web applications, with a particular strength in backend architecture. With extensive experience leading technical initiatives, I consistently deliver high-quality solutions using best practices. Passionate about clean code, maintainable architectures, and creating efficient, scalable systems.

Website
jiliac.com

Links
@Jilyac
Jiliac
valentinmanes

Programming
Node JS
React
Go
Python
C++

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

Interests
Card Games
Languages
Books
Travel

Experience

2024 **AI Content Creation Startup** Paris, France

Co-founded a startup focused on automating long-form YouTube content creation (20-90 minutes) targeting niche markets. The goal was to enable content creators to efficiently produce high-quality videos and monetize through ad revenue.

- Built prototype using **Node.js**, **React**, and **Python** with **Firebase Functions** backend
- Integrated multiple AI services: **GPT**, **Eleven Labs** for voice synthesis, and **Stability AI** for visual content
- Project reached prototype stage before concluding due to strategic differences between co-founders

2022-23 **Violet** Berlin, Germany

Violet is a Web3 company specializing in digital identity and KYC solutions. As a software engineer, I focused on wallet integration and frontend development.

- Developed user interfaces using **React** and backend services with **Node.js**
- Implemented wallet integrations and smart contract interactions
- Contributed to KYC workflows and identity verification systems

2021 **Qonto** Paris, France

At Qonto, a European neobank for professionals, I was part of the Ledger team maintaining the core banking system. Worked on micro-services using **Go** and **PostgreSQL**, and contributed to the billing system in Ruby on Rails. The infrastructure consisted of 80+ micro-services deployed with **Kubernetes** and **Argo CD**.

2020 **PacketAI** Paris, France

Developed an IT infrastructure monitoring platform with ML capabilities for incident prediction. Built multiple microservices in **Go**, implemented data collection agents, and designed the data pipeline using **ELK** stack and **Kafka**. **Mentored** junior team members.

2016-19 **Cyber Security Research Center - KAIST** Daejeon, South Korea

Led two major projects at KAIST's research center: Enhanced LLVM's dynamic testing tool (**C++**, merged by *Google*) and developed Ankou, a novel fuzzing tool (**Go**). Implemented ML-based improvements using **Python** stack (**TensorFlow**, **Keras**). Managed distributed experiments using **Docker** and **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 that boosts fuzzing performance by optimizing seed selection based on information gain. **The work was evaluated and integrated by Google into LibFuzzer**, making it part of the LLVM project’s mainline fuzzing engine. This implementation has since become a standard component of Google’s fuzzing infrastructure.

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 (**600+ citations**)
Companion website: fuzzing-survey.org
This comprehensive survey has become the definitive reference in the field of fuzzing. It presents a unified, general-purpose model that systematically categorizes fuzzing techniques. The paper’s impact is reflected in its extensive citation count and its companion website, which maintains an up-to-date database of fuzzing research. By identifying and analyzing key algorithmic stages of fuzzers, we established a framework that continues to guide research and development in the field.

2018 **Domain Isolated Kernel**
Elsevier Computer & Security
Code: github.com/Jiliac/DIKernel
A novel approach to kernel security that isolates driver extensions through memory access restrictions and privilege control. Implemented on Linux 4.13 with minimal performance overhead and no required changes to existing driver code.