Valentin Manès

Contact
valentinmanes@outlook.fr

Website
https://jiliac.com

Profil
Software Engineer
Applied Mathematics

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

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

Interests
Languages
History
Travel

Experience

2020 **PacketAI** Paris, France

2016-19 **Cyber Security Research Center - KAIST** Daejeon, South Korea
During my first year, I worked on developing a kernel hardening solution by limiting the kernel attack surface. Then, I reoriented myself towards Automatic Software Testing (also called fuzzing). Fuzzers generate inputs with the intent of causing the software under test to misbehave. Specifically, I investigated the usage of machine learning techniques to improve fuzzers bug finding ability.

Education

2015-16 **KAIST - One Year Exchange** Daejeon, South Korea
KAIST 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 five 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 **Ankou: Guiding Grey-box Fuzzing towards Combinatorial Difference**
International Conference on Software Engineering
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
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
Kernel extensions (i.e. modules, drivers) are the weakest kernel part concerning its security. Thus, 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.