GIUSEPPE DE ROSA

+39 342 733 1188 ◇ Naples (NA)
giuseppe.de.rosa@outlook.it ◇ linkedin.com/GiuseppeDeRosa

EDUCATION

National Ph.D. in Cybersecurity, Scuola IMT Alti Studi Lucca, Italy — Expected 2026
Cycle XXXIX, advisor prof. Domenico Cotroneo

Master of Security Computer Engineering, University of Naples Federico II, Italy — 2021-2023
Final Grade: 110/110 cum laude (Italian grading system)

Bachelor of Computer Engineering, University of Naples Federico II, Italy — 2017 - 2020

SKILLS

Programming	Python, C, C++, Java, Assembly
Cybersecurity	Risk Assessment, Vulnerability Testing (fuzz testing, pentesting)
	ISO 2700x, NIST, GDPR, OWASP Standards
Machine Learning	Dataset Creation, Data Engineering, NLP
Virtualization	Hypervisor Configuration, VM Management, Nested Virtualization
Soft Skills	Problem Solving, Teamwork, Client Management
Languages	Italian (Native), English (B2), French (A2)

PROJECTS & PUBBLICATIONS

How does Python software systems fail? Led the development of PyResBugs, a comprehensive dataset containing 5,040 real-world residual bug pairs (faulty and fixed code) from major Python frameworks. This research presents the first dataset specifically designed for Software Fault Injection (SFI) in Python applications, focusing exclusively on residual faults that evaded traditional testing approaches. Each bug pair is annotated with multi-level natural language descriptions, facilitating the training of AI-based solutions for automated fault detection and generation. Our analysis, based on the extended ODC, of real-world defects revealed that Missing Parameter in Function Calls (10.41%) and Missing IF Constructs (10.08%) are the most prevalent issues and provides insights on the main class of faults for weakly-typed programming languages. (Publication in preparation)

How to generate Python bugs? I am currently working on a project focused on training a model (using CodeT5+) to identify fault patterns in complex Python software. This project involves examining various technologies, including Django for web development, Pandas for data analysis, Keras for machine learning, and OpenStack for cloud computing. The objective is to utilize this model for conducting software fault injection campaigns to evaluate the dependability of complex systems against residual faults, i.e. non-trivial faults. A significant challenge in this work is constructing and structuring a dataset of residual faults collected from GitHub. This involves preparing a large dataset, ensuring it contains a variety of interesting, repeatable, and useful real-world faults for effective model training. The project also includes designing and implementing experiments, and collaborating with team members to refine the model and enhance its accuracy and impact on fault injection. (Publication in preparation)

COSMOS. The complexity and size of today's hypervisors pose significant challenges for software dependability and security. COSMOS is a project, initiated during my master's thesis, that enables testing the hardware fault-handling capabilities of a hypervisor through nested virtualization technology in a fault injection campaign. The software has been tested on general-purpose hypervisors, such as KVM and Xen, as well as partitioning hypervisors like Jailhouse. The related paper is under review for the Transactions on Dependability and Security Computing (TDSC) journal.

RESEARCH PROPOSAL

My research activity aims to develop an innovative framework for automated program repair (APR), using large language models (LLMs) and the hypervisor's features. The goal is to create an intelligent self-healing system that automatically detects and corrects software bugs with minimal downtime. The core of this framework consists of LLMs fine-tuned for code comprehension and repair, coupled with algorithms for patch optimization and correctness. The

framework incorporates hypervisor technology to enable the seamless application and validation of LLM-generated patches without interrupting software execution. This approach minimizes downtime for mission-critical applications while allowing just-in-time testing in an isolated environment. By combining LLM and hypervisor technology, this research aims to create a next-generation repair system that not only fixes current bugs but also prevents possible future software defects, significantly improving software reliability and maintainability.

OTHER ACTIVITIES

- Worked in Critiware between 2020 and 2021, a start-up ICT company providing services and tools to support the engineering of critical computer systems. I worked with key frameworks and standards including ISO 2700x, NIST, and GDPR, developing a comprehensive understanding of cybersecurity best practices. My role involved conducting risk assessments and penetration testing for major clients like Trenitalia, Italy's leading railway operator, providing hands-on experience in identifying and mitigating security vulnerabilities in railway systems. I also contributed to cybersecurity requirement analysis for Hitachi's WMATA metro project, a global leader in transportation solutions.
- Worked, during the Bachelor Degree, in the Unina Corse E-Team to the design, development, and coordination of activities for the first Electronic Control Unit (ECU) of the T-ONE, the first electric racing car in Southern Italy, adhering to AUTOSAR standards. This involved real-time development using the ChibiOS operating system for controlling the ECU and car sensors, implementing key control mechanisms based on priority mechanics and static deadline scheduling. The project utilized STM32 microcontrollers, particularly the F7 and F4 models.