AI4VULN

AI enhanced vulnerability detector

AI4VULN is a static source code analyzer tool that can detect vulnerability problems in the source code. It uses symbolic execution, which means that it simulates the execution of the program without real execution and this way, it can detect runtime vulnerability issues in the code.

AI4CYBER

Component

AI4VULV is a static source code analyzer, but it uses symbolic execution to find runtime problems in the source code without real execution. Although symbolic execution is a powerful tool for larger projects, it requires a great amount of resources and can be very slow. To mitigate this weakness, LLMs are used to select the vulnerable methods that the algorithm focuses on. This way, AI4VULN achieves almost the same results with much less effort.

AI4VULN is a static analysis tool that utilizes symbolic execution, a special form of static program analysis. During regular execution, the variables of the program have concrete values, meaning that the program follows a specific executional path determined by these values.

In this scenario, only one path is taken, and all other possibilities are ignored. This limitation makes it hard to uncover issues that only arise under certain inputs or rare conditions. Moreover, in the case of static analysis, input values are not available, therefore, it is difficult to detect runtime problems in the code. During symbolic execution, the program is executed on symbolic values instead of concrete values. Symbolic variables represent a range of possible values.

Artificial intelligence offers additional possibilities to increase the efficiency of symbolic execution. By integrating AI, AI4VULN became more scalable and focused, making it more suitable for real-world software analysis. One possible application of AI is to guide the symbolic analysis towards the more relevant parts of the code. Instead of analyzing every function equally, the system prioritizes the code that is more likely to be vulnerable.

After extensive experimentation, Gemini was selected for AI4VULN. We use this LLM to analyze the codebase and produce a ranked list of methods based on their estimated risk. AI4VULN launches more precise, higher resolution executions from the methods on the list than from the other methods in the project. This means that more computing power is allocated to the methods most likely to contain vulnerabilities. This way, the execution time is reduced significantly, but the results, mainly the true positive warnings, remain the same.

In AI4VULN, we tried to find a solution to the challenges of classical symbolic execution using AI. Our hybrid approach is a scalable, targeted, and effective solution, even for large software projects. As AI models continue to evolve, tools like AI4VULN represent a promising direction in the ongoing effort to improve software security and reliability through static analysis.

YouTube Video Link

The solution is still a prototype and the future tool will have commercial license. AI4VULN is hosted as an open-source project in FrontEndART GitHub repository. https://github.com/FrontEndART/AI4Vuln

Istvan Siket (istvan.siket@frontendart.com)
FrontEndART Ltd

FRONTENDART

https://frontendart.com/en/

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070450. Disclaimer: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.