УДК 004.056

Automated Security Scanning in CI/CD Pipelines Using Large Language Models (LLMs)

Терро Моайад (ИТМО)

Научный руководитель - кандидат технических наук, доцент Д. А. Заколдаев (ИТМО)

Introduction. With the rising of Artificial Intelligence (AI) and its applications it became crucial for rapid development applications, companies start the adoption of AI assistance and plugin to help developers writing code faster, vibe coding is a new term that refer development of applications using prompts to generate code by conversation, although this help a lot it raise a new concern about the security of such applications [1], the need of automated vulnerabilities scanning it become essential, DevOps engineers start to implement static code analysis tools into pipelines, the main disadvantage of such tools that it produce large number of false alarms [2] and that killed the whole idea of continuous delivery, to solve the problem of false positive alarms this study suggest a new methodology that employee Large Language models (LLM) to work as static code analyzer by scanning the code base in early stage of CI/CD pipeline to ensure continuous safety.

Main part. This work suggests using LLM-driven static code analysis that scans codebase for security vulnerabilities, the model ingests code snippers or even complete files, understand the context and which part of third-party library that was used, where, and how then identify potentially vulnerable patterns, assess the actual risk and provide developers with full details report and even references about suggest solutions. Recent research on hybrid LLM assisted static analysis demonstrates that combining LLMs with traditional static analysis can increase detecting vulnerabilities compared to tools like CodeQL [3]. Our suggested methodology would be faster, more accurate, and more efficient, the code check stage in pipeline start by building an abstract syntax tree and a graph representation of the codebase and pass it to a fine-tuned model that already trained on code in the first place such as deepseek-coder, fine-tuning the model would compactly create an LLM expert that specialize in cyber security and able to make a decision whether this snippet is vulnerable or not in this context. This design will result more accurate analysis and reduce false alarms comparing with the traditional static code analysis tools.

Conclusion. This study suggested a new methodology to replace the normal rule-based static analysis tools and produce many false positives by a solution based on Large Language models; the new methodology would solve the problem of false alarms and lack of context in the traditional static code analysis.

References:

1. Zhao S. и др. Is Vibe Coding Safe? Benchmarking Vulnerability of Agent-Generated Code in Real-World Tasks. 2026.
2. Kang H. J., Aw K. L., Lo D. Detecting False Alarms from Automatic Static Analysis Tools: How Far are We? // Proceedings - International Conference on Software Engineering. IEEE Computer Society, 2022. T. 2022-May. C. 698–709.
3. Li Z., Dutta S., Naik M. IRIS: LLM-Assisted Static Analysis for Detecting Security Vulnerabilities // 13th International Conference on Learning Representations, ICLR 2025. International Conference on Learning Representations, ICLR, 2025. C. 16868–16891.