УДК 004.056

Continuous Security in CI/CD Pipelines with Static Code Analysis Tools

Терро Моайад (ИТМО)
Научный руководитель – кандидат технических наук, доцент Коржук В.М. (ИТМО)

Introduction. DevOps or Development and Operations are considered as one of the promising software development methodologies in the industry. However, the adoption of it has presented a new challenge of ensuring secure software delivery and maintaining the agility of DevOps. As a solution to integrate security into DevOps, a new term has emerged, DevSecOps (Development, Security and Operations) which start to get more attention from industry and academics. The current CI/CD pipelines suffer from malicious code and severe vulnerabilities and people have not been fully aware of their attack surfaces and the corresponding impacts [1]. Using static code analysis tools within CI/CD pipelines significantly enhances security by enabling early detection of vulnerabilities and promoting secure coding practices. SAST tools play a crucial role in detecting vulnerabilities, widespread adoption has been hindered by usability issues, including high false positive rates and a lack of native pipeline support [2].

Main part. With the help of source code analysis tools, a pipeline can be created that supports continuous security by adding a new stage in the pipeline before starting the process of executing the source code and starting the building to create ready-to-use production environment. Open-source tools have been used to detect vulnerabilities early, making the mechanism proactive so that the source code can be analyzed, and possible vulnerabilities can be discovered as a first line of defense. With the spread of code generation tools using artificial intelligence (AI) tools and copying and pasting from different sources on the Internet, and considering that the developer is a human, he is the weakest element in the whole picture, code scanning tools will automate the process and work to detect in advance and give warnings at each build process. Jenkins is an open-source tool built in Java programing language to automate the process of creating CI/CD pipelines. In this research, tools such as Semgrep and SonarQube were used to scan a source code in which vulnerabilities were deliberately added, more than one pipeline build triggered to start the pipeline and test the experiment with a real case scenario. The results showed that using more than one tool in sequence may increase the accuracy of detection and achieve the more benefit in CI/CD pipelines.

Conclusion. A source code analysis was performed, the number of detected vulnerabilities has been calculated, and the methodology of vulnerabilities detection for CI/CD pipeline was developed.

References:

1. Z. Pan et al., "Ambush from All Sides: Understanding Security Threats in Open-Source Software CI/CD Pipelines," in IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 1, pp. 403-418, Jan.-Feb. 2024, Doi: 10.1109/TDSC.2023.3253572.
2. Z. Wadhams, A. M. Reinhold and C. Izurieta, "Automating Static Code Analysis Through CI/CD Pipeline Integration," 2024 IEEE International Conference on Software Analysis, Evolution and Reengineering - Companion (SANER-C), Rovaniemi, Finland, 2024, pp. 119-125, Doi: 10.1109/SANER-C62648.2024.00021.