ISSN 2522-9400 **European Modern Studies Journal** Vol 8 No 3
available at www.journal-ems.com
DOI: 10.59573/emsj.8(3).2024.30

An Approach to Enhance CI/CD Pipeline with Open-Source Security Tools

Hung Ho-Dac, Van-Len Vo
Digital Transformation Department,
Thu Dau Mot University, Vietnam

Abstract. Continuous Integration (CI) and Continuous Deployment (CD) are important aspects in software engineering today. In modern software production organizational models, CI/CD pipeline has become a mandatory element to improve speed and reduce team effort in developing, integrating, and deploying. In the context of increasing information security risks, deploying security tools for the CI/CD pipeline has become an inevitable trend. Deploying information security tools throughout the pipeline according to the "Shift Left" philosophy will help detect information security issues early for timely handling and reduce correction costs. In this research, we present an approach to improve the CI/CD pipeline by integrating information security tools introduced by the Open Source Foundation for Application Security Project (OWASP). In addition, we also present trade-offs when implementing information security into the CI/CD pipeline.

Keywords: CI/CD pipeline, DevSecOps, OWASP

Introduction

DevOps was conceived with the aim of accelerating the delivery of high-quality software by leveraging automation, speed, rigorous feedback, and cross-functional collaboration to the software development lifecycle (Myrbakken et al., 2017). DevSecOps leverages these same philosophies combined with information security elements to make the software development process more secure (Figure 1).

Pipeline is essentially a software development process through a roadmap of core phases including but not limited to installation, testing and deployment phases (Zampetti et al., 2021). By automating part or all of the pipeline, the basic goal is to minimize individual errors and maintain a consistent process in implementation. Tools used in the pipeline typically include source code compilation, unit testing, source code analysis, safety testing, and packaging tools.

Literature Review

Essentially, software development platforms and pipelines have some default support for information security. However, the default support does not cover all information security requirements in the current context. Therefore, nowadays, pipelines are supplemented with many information security tools to achieve information security requirements. Depending on each specific software field, different tools are considered for use to achieve the necessary efficiency.

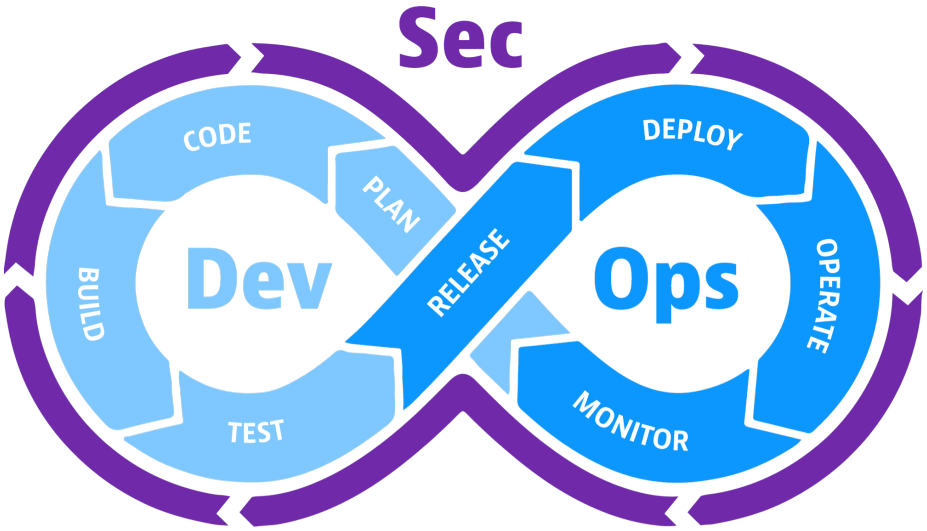

Figure 1. DevSecOps life cycle
Source: www.atlassian.com

Currently, there are many tools to deploy CI/CD pipelines. Some prominent tools include Bitbucket Pipelines (Atlassian, n.d.), Jenkins (Jenkins, n.d.), AWS CodePipeline (Amazon, n.d.), CircleCI (CircleCI, n.d.), Azure Pipelines (Microsoft, n.d.), GitLab (GitLab, n.d.), Atlassian Bamboo (Atlassian, n.d.). Tools have unique characteristics (Table 1) and depending on the nature of each project, the project team will choose the appropriate tool.

Table 1. Description of CI/CD pipeline deployment tools

Tool	Description	Feature
Bitbucket Pipelines	Bitbucket Pipelines CI is integrated directly into Bitbucket powered by Atlassian. Bitbucket Pipelines is an easy next step to enable CI for projects already available on Bitbucket. Bitbucket Pipelines are managed as code so it's easy to define pipeline definitions and start building. Bitbucket Pipelines also offers CDs.	Easy to set up and configure Unified experience with Bitbucket Cloud services provided by 3rd parties
Jenkins	Jenkins is a time-proven CI tool. Jenkins is open-source and maintained by the community. Jenkins is especially suitable for projects built in on-premise environments. Amazon Web Service (AWS) is a prominent cloud infrastructure provider. AWS provides a complete set of tools related to software development and infrastructure. CodePipeline is AWS's primary CI tool. CodePipeline unifies the experience with other AWS services.	On-premise Open-source Addon/Plugin
AWS CodePipeline	Amazon Web Service (AWS) is a prominent cloud infrastructure provider. AWS provides a complete set of tools related to software development and infrastructure. CodePipeline is AWS's primary CI tool. CodePipeline unifies the experience with other AWS services.	Entire deployment in the cloud Integrate with other AWS services Plugin
CircleCI	CircleCI is a CI Tool combined with Github. CircleCI is one of the most flexible CI Tools with	Notification notifies events

	the ability to support a matrix of version control systems, container systems, and deployment mechanisms. CircleCI can be deployed on-premise or in the cloud.	Optimize performance Easy debugging Control performance metrics
Azure Pipelines	Azure is Microsoft's cloud infrastructure platform, Microsoft's equivalent to Amazon Web Services. Like the aforementioned AWS CodePipeline, Azure offers CI Tools that are fully integrated into the Azure storage toolkit.	Azure platform integration Windows platform integration Github integration
GitLab	Gitlab is a new CI Tool that provides a full DevOps experience. Gitlab was created with the intention of improving the overall experience of Github. Gitlab provides modern UX with container support.	On-premise and cloud support Support information security testing
Atlassian Bamboo	Another CI product from Atlassian. While Bitbucket Pipelines is purely a cloud-hosted option, Bamboo offers an on-premise alternative.	Integrate Atlassian productsAddons/Plugins Container support with docker

Integration

Besides the basic objective of promoting integration and deployment; The pipeline must also ensure information security factors. By integrating information security tools into the pipeline, project teams can improve the information security factor of the pipeline. OWASP (Raghu Vamsi et al., 2023) recommends that pipelines can integrate tool groups including but not limited to (Table 2): static source code analysis tool group, dynamic source code analysis tool group, source code quality analysis tool group, dependency analysis tool group, interaction analysis tool group.

Static Application Security Testing (SAST) tools can help analyze raw or compiled source code to identify potential information security vulnerabilities. These tools can be run independently or embedded into the IDE as a plugin to help developers inspect source code directly while building them. Static source code analysis tools can help significantly limit the project team's time and effort related to information security because they help detect vulnerabilities at very early stage.

Table 2. Describe information security testing tools

Tool group	Tool	Feature
Static source code analysis tool	.NET Security Guard	Open-source or free IDE integration
	ABOM Scanner	Free SaaS
	Agnitio	Open-source or free Windows OS compatible
Dynamic source code analysis tool	AppTrana	Free SaaS
	Arachni	Free Cross-platform support
	CI Fuzz CLI	Open-source or free Cross-platform support

Source code quality analysis tool	PVS-Studio	Paid, free for 1 year Pipeline compatible
	SonarQube	Paid, free Pipeline, IDE compatible
	Crucible	Paid Pipeline compatible

Currently, modern web application architectures are broken down into many independent components for easy development, maintenance and extension. A user action can call on many independent components to execute functions to produce the final output. Therefore, some features are difficult to fully test using static source code analysis tools. To scan for vulnerabilities during the execution time of a web application, dynamic source code analysis tools are applied. These tools, in addition to controlling the source code, also control requests and responses to find potential vulnerabilities.

In addition, a pipeline also needs to be added to dependency analysis tools to check constraints and 3rd party source code. 3rd party source code is considered one of the sources of high-risk vulnerabilities. Currently, the use of third-party source code is very popular due to the software development philosophy of "standing on the shoulders of giants". Therefore, dependency analysis tools are often a required component in pipelines.

Static source code analysis tools are often used as plugins that plug into IDEs and serve primarily a developer audience. On build servers, static source code analysis is performed by another group of tools called source code quality analysis tools. In terms of core functionality, these two groups of tools have the same functions. However, the group of source code analysis tools is built to be compatible and provide stronger support for the build process on the build server.

Another group of tools that often appear in pipelines is the group of interactive analytics tools. This group of tools focuses on analyzing potential vulnerabilities through communication channels between components such as RESTful APIs and queues.

Results and Discussion

In this study, we built the pipeline with the main tool Jenkins, the static source code analysis tool "Security Code Scan" (Security Code Scan, n.d.), the dynamic source code analysis tool "OWASP ZAP" (Jakobsson et al., 2022), source code quality analysis tool “SonarQube” (Campbell et al., 2013), dependency analysis tool “OWASP Dependency Check” (OWAS, n.d.), monitoring tool “Prometheus” (An et al., 2021).

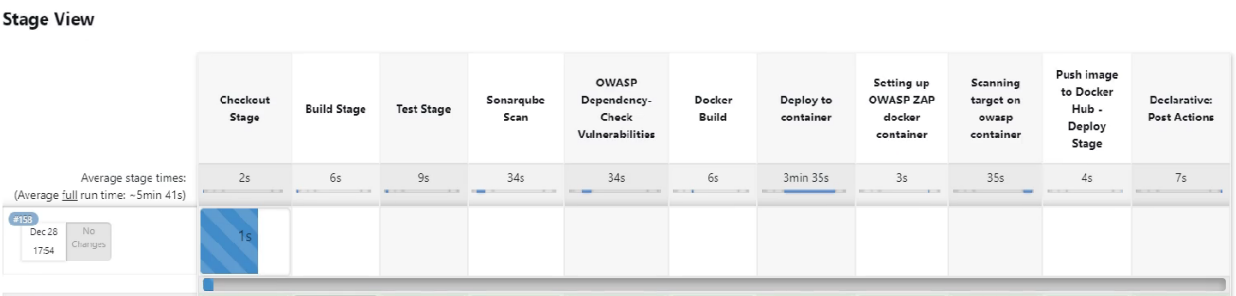

Figure 2. Pipeline integrates security testing tools as recommended by OWASP

Through testing the pipeline, the results show that the pipeline has basically tested information security aspects in the CI/CD process. However, pipeline execution time increases significantly when information security tools are deployed. Therefore, depending on resources

and severity of the project, the project team can consider increasing or decreasing information security tools to achieve the desired effect.

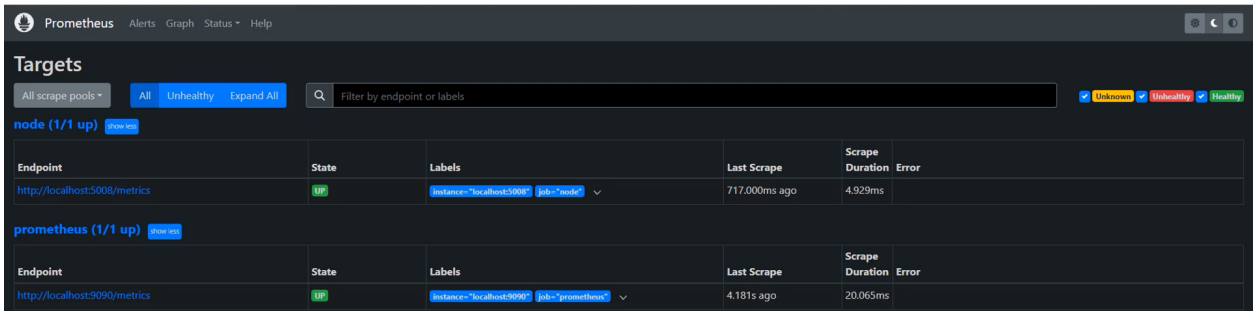

Figure 3. Monitoring tool shows results and logs of builds

Conclusion

Integrating information security tools into the pipeline is an inevitable trend. Information security organizations recommend a variety of toolkits with very different characteristics. Therefore, depending on the actual situation, project teams can flexibly choose appropriate tools. However, there is still a need to ensure tools fall within the core groups as recommended by OWASP.

References

Amazon. (n.d.). CI/CD Pipeline - AWS CodePipeline - AWS. Amazon AWS. Retrieved May 5, 2024, from http://aws.amazon.com/codepipeline/

An, Seong Yeol, et al. (2021). A pre-study on the open source prometheus monitoring system. *Smart Media Journal, 10*(2), 110-118.

Atlassian. (n.d.). Bamboo: Continuous Integration & Deployment. Atlassian. Retrieved May 5, 2024, from http://www.atlassian.com/software/bamboo

Atlassian. (n.d.). Bitbucket Pipelines - Continuous Delivery. Bitbucket. Retrieved May 5, 2024, from http://bitbucket.org/product/features/pipelines

Campbell, G. A., & Papapetrou, P. P. (2013). *SonarQube in action*. Manning Publications Co.

CircleCI. (n.d.). CircleCI: Continuous Integration and Delivery. Retrieved May 5, 2024, from http://circleci.com/

GitLab. (n.d.). The most-comprehensive AI-powered DevSecOps platform. Retrieved May 5, 2024, from http://about.gitlab.com/

Jakobsson, A., & Häggström, I. (2022). *Study of the techniques used by OWASP ZAP for analysis of vulnerabilities in web applications*.

Jenkins. (n.d.). Jenkins. Retrieved May 5, 2024, from http://www.jenkins.io/

Microsoft. (n.d.). Microsoft Azure: Cloud Computing Services. Retrieved May 5, 2024, from http://azure.microsoft.com/en-us

Myrbakken, H., & Colomo-Palacios, R. (2017). DevSecOps: a multivocal literature review. *Software Process Improvement and Capability Determination: 17th International Conference,* SPICE 2017, Palma de Mallorca, Spain, Proceedings. Springer International Publishing.

OWASP. (n.d.). OWASP Dependency-Check. OWASP Foundation. Retrieved May 5, 2024, from http://owasp.org/www-project-dependency-check/

Raghu Vamsi, P., Ahmad, A., & Dwivedi, V. (2023). Application for Simulating OWASP Vulnerabilities. *International Conference on Data Science and Communication*. Singapore: Springer Nature Singapore.

Security Code Scan. (n.d.). Security Code Scan. Retrieved May 5, 2024, from http://security-code-scan.github.io/

Zampetti, F., et al. (2021). Ci/cd pipelines evolution and restructuring: A qualitative and quantitative study. *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE.