УДК 004.056.54

ENHANCING PROACTIVE VULNERABILITY MANAGEMENT INTEGRATING DEVSECOPS

Driss Alaoui Soulimani, Kadatska Olha, Shaimae Mishish
e-mail: olha.kadatska@nure.ua
Kharkiv National University of Radio Electronics,
V.V. Popovskyy dep. ICE,
Kharkiv, Ukraine

The need for robust security throughout the software development life cycle has grown with the rise in complex software ecosystems. This paper discusses the adoption of DevSecOps practices into Continuous Integration/Continuous Deployment pipelines to achieve proactive vulnerability detection and mitigation. Embedding security at every step of the SDLC, therefore, helps in minimizing risks, improving operational efficiency, and streamlining secure software delivery.

Quickness and scalability are much emphasized in modern software development. However, most of the traditional development models overlook security and treat it as an end step. This brings systems that have a lot of critical flaws, a fact which the number of CVEs reported each year depicts, and we looks at how DevSecOps methodology fits security into pipelines of CI/CD for early detection of vulnerabilities, thereby addressing challenges in proactive security management.

DevSecOps extends the principles of DevOps by embedding security testing in all parts of the development lifecycle. It fosters collaboration between development, operations, and security teams by emphasizing automation, continuous feedback, and real-time vulnerability mitigation.

CI/CD pipelines automate the building, testing, and deployment of software. While this accelerates delivery, it also amplifies risks if security is not embedded within the pipeline. Traditional CI/CD processes lack mechanisms to detect vulnerabilities early, increasing the likelihood of critical issues reaching production.

Challenges in integrating security into CI/CD pipelines

1. Late-stage testing: vulnerabilities are often discovered post-deployment, resulting in high remediation costs and production risks.

2. Tool integration: security tools like SAST and DAST must seamlessly integrate into automated workflows without disrupting development speed.

3. False positives: automated tools can generate excessive false positives, creating noise that burdens developers.

Proposed solution DevSecOps framework for CI/CD pipelines include

Automated security testing:

- Static Application Security Testing (SAST): Identifies vulnerabilities in source code during the development phase, enabling early mitigation. Tools like

SonarQube analyze code for flaws such as SQL Injection and Cross-Site Scripting (XSS).

- Dynamic Application Security Testing (DAST): Simulates external attacks during runtime to detect vulnerabilities like authentication flaws and session management errors. OWASP ZAP is a widely used tool for this purpose.

Penetration testing serves as a manual validation layer to uncover vulnerabilities that automated tools might miss. This includes simulating real-world attack scenarios to validate the application’s resilience.

CI/CD integration integrates SAST and DAST tools directly into CI/CD pipelines. Vulnerabilities detected during code scans or runtime tests automatically halt the pipeline, ensuring only secure code progresses to production. Penetration tests are performed periodically to reinforce automated checks.

A comparison of detection rates highlighted the efficiency of SAST tools, which identified 10.65 vulnerabilities per minute, compared to DAST's 0.76 vulnerabilities per minute.

After implementing the proposed solutions in the real world and obtaining results that validate our solutions, we offer the following key recommendations:

1. Embedded security early: SAST tools provide significant benefits by identifying vulnerabilities during development.

2. Adopt a layered approach: combining SAST, DAST, and penetration testing ensures comprehensive security coverage.

3. Automate where possible: automated tools reduce the manual effort required for security testing, accelerating secure software delivery.

4. Regular audits: periodic penetration testing validates the effectiveness of automated tools and uncovers advanced vulnerabilities

As conclusion Integrating DevSecOps into CI/CD pipelines represents a paradigm shift in how organizations approach software security. By embedding security at every stage of the SDLC, organizations can proactively mitigate vulnerabilities, enhance operational efficiency, and deliver secure software faster. This paper underscores the importance of combining automated and manual testing methods to establish a robust security framework.

References:

1.Gupta, S. Implementing DevSecOps in Large Organizations. Springer, Berlin, 2022. 300 p.

2.SonarSource. SonarQube Documentation: Static Application Security Testing (SAST). 2024. URL: https://docs.sonarqube.org (accessed 2024-01-05).

3.OWASP Foundation. OWASP ZAP Documentation: Dynamic Application Security Testing (DAST). 2024. URL: https://owasp.org/www-project-zap/ (accessed 2024-01-05).