DevSecOps with SAST

Sudhir Singhal
Application Security Manager
Tavant Technologies India Private Limited

Chandra Pushpanjali Patel
Indraprastha Engineering College

ABSTRACT
DevSecOps is a process of automated security tool integration into all phases of the software development life cycle. It's an approach and requires a change in process, and automation. In agile, every team member has a role to play in building security into continuous integration and continuous delivery. It improves knowledge and understanding of security practices, tools, and processes within the context of software development while working in Scrum. DevSecOps improves so many things like the development of effective measures, reduced vulnerability, turnaround time, faster security checks, early vulnerability detection, increased visibility, improved collaboration, and enhanced customer trust.

Keywords
DevSecOps, SAST

1. INTRODUCTION
DevSecOps (Development Security and Operation) work best when an organization focus on agility, through enable continuous integration with security we can develop better, high performing, and more secure software faster and with less cost and less efforts, it can be a excited journey that will deliver better business value and outcome, but when executed correctly. Every day there is new challenges, DevSecOps best practices speed up time to market and lowers cost for the business. in agile process identify the issues as early and integrate application security into DevOps and agile processes. It addresses security issues as they occur. The team of DevSecOps deliver secure code. This code runs faster at lower costs. It's easier and less costly to catch and fix vulnerabilities before they go into production release.

Security needs in SDLC
1.1 Key business value of implementing the DevSecOps Process

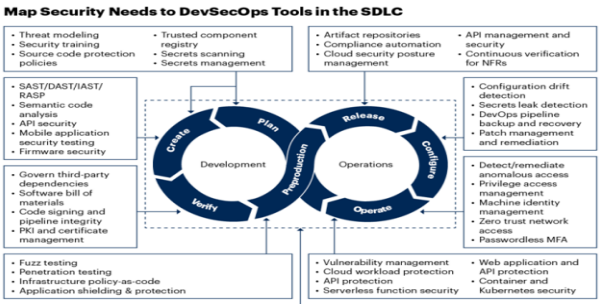

Fig 1: DevSecOps in the SDLC

Delivering the value of DevOps : The incorporation of security principles into DevOps creates a culture of shared accountability, which improves overall security posture.The overall security posture can be improved by the incorporation of security principles into DevOps produces a culture of shared accountability.

Increasing (Rising) the Likelihood of Overall Business Success
Increased revenue growth and expanded (enlarged) business offerings are enabled (set up) by increased faith in the security of developed software and the adoption of new technologies.

Automation Compatible with Modern Development
The Business growth and goals have a huge impact on automated security check. Automated application security testing can verify that incorporated software dependencies are patched to the appropriate levels.

2. SECURITY TOOLS IN DEVSECOPS
DevSecOps enhances security testing at every step of the software development process. It works with tools and process efficiently. DevSecOps brings cultural transformation and shared the responsibilities to everyone who is responsible for software. It includes the following tools.

2.1 Static application security testing (SAST)

This tool checks the security without executing the application. It focuses on potential vulnerabilities. SAST tools are crucial in the software development space since they detect vulnerabilities that leave systems open to attacks such as coding errors, leakage of private data, and design flaws that could lead to exploitable weaknesses, SAST can identify a range of vulnerabilities without running the application.

2.1.1 Commonly used SAST tools
- *Checkmarx:* Checkmarx is a widely used SAST tool that helps identify and remediate security vulnerabilities in source code.
- *Fortify Static Code Analyzer:* Offered by Micro Focus, Fortify is another popular SAST tool that provides advanced security scanning capabilities.
- *Veracode:* Veracode is a cloud-based application security platform that includes SAST capabilities to identify and fix security issues in code.
- *SonarQube:* While primarily known for its code quality analysis, SonarQube also includes security scanning features to identify vulnerabilities in code.

2.2 Software Composition Analysis (SCA).
SCA tools identify project dependencies and check if there are any known, publicly disclosed, vulnerabilities within a codebase. In addition, they can be integrated seamlessly into a CI/CD process as well.

2.2.1 SCA tools are as follows:

- *OWASP Dependency-Check:* An open-source tool from OWASP, Dependency-Check scans project dependencies

for known vulnerabilities and provides reports on the findings. Features to identify vulnerabilities in code.

- *NPM Audit:* This is a built-in tool for Node.js projects that checks for vulnerabilities in JavaScript packages using the npm package manager.
- *JFrog Xray:* JFrog Xray is an SCA tool that integrates with JFrog Artifactory to scan artifacts and dependencies for vulnerabilities, licensing issues, and compliance violations.
- *Nexus Repository (Sonatype OSS):* Sonatype Nexus Repository, in addition to its lifecycle component, also provides a repository manager that can help you store and manage open-source dependencies securely.

2.3 Dynamic application security testing (DAST).

DAST is an automated and black-box security testing process in which tests are performed by attacking an application to the front end to find vulnerabilities. DAST tools do not require access to source code but require a running application.

1.1.1 Using DAST scanners we can find vulnerabilities such as SQL injection commands, cross-site scripting (XSS), and unusual inputs that might uncover issues in input validation or memory management.

2.3.1 DAST tools:

- *Burp Suite:* Burp Suite is a popular web vulnerability scanner and proxy tool that helps identify security issues in web applications.
- *OWASP ZAP (Zed Attack Proxy):* ZAP is an open-source security testing tool designed to find web application vulnerabilities.
- *Acunetix:* Acunetix is a web vulnerability scanner that checks web applications for a wide range of security issues, providing detailed reports and remediation guidance.
- *AppSpider*: AppSpider by Rapid7 is a DAST tool that scans web applications for vulnerabilities and offers integration with other security tools.
- *WebInspect:* WebInspect by Micro Focus is a DAST tool that identifies security vulnerabilities in web applications.

3. WORKING MODEL OF SAST

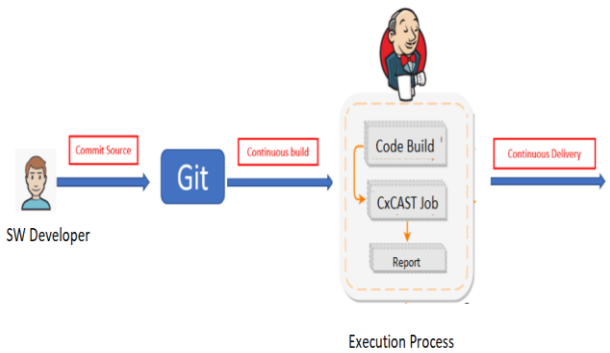

Fig 2:SAST working

Git: Git is a distributed version control system that tracks changes to source code and enables collaborative development by multiple contributors.

Continuous build: Often referred to as "Continuous Integration" (CI), is a software development practice in which code changes are automatically built, tested, and verified whenever new code is added to a project's version control system. The goal is to ensure that code changes do not introduce errors or break existing functionality, promoting early detection and resolution of issues in the development process.

IDE: IDE stands for Integrated Development Environment. It is a software application that provides a comprehensive and integrated set of tools and features for software developers to streamline and facilitate the process of writing, testing, and debugging code.

3.1 SAST view during build

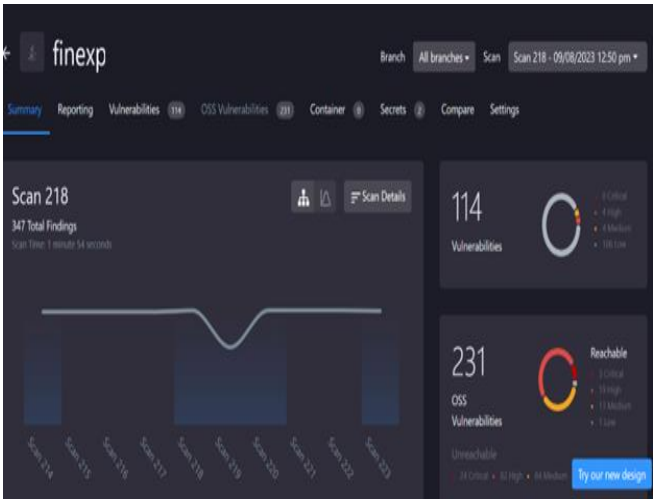

Fig 3 . Testing the Data

And after fixing the few vulnerabilities, vulnerabilities count came down, refer below image

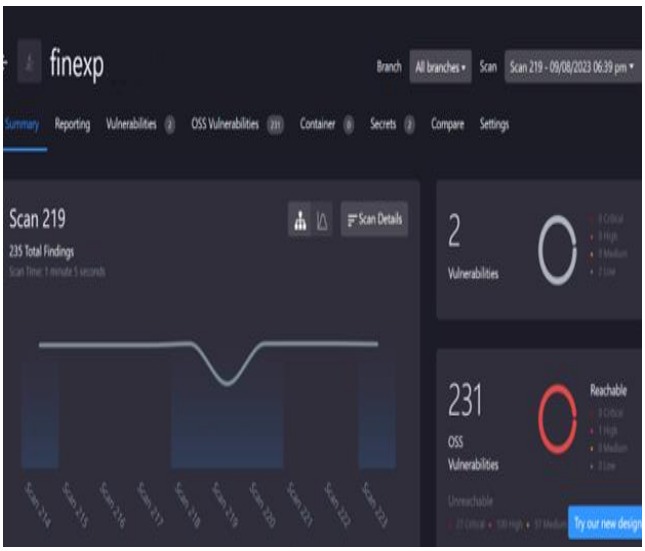

Fig 4. Testing the Data

Integration of SAST into DevSecOps

This tool checks the source code for security defects. It is one of the many checks in an application security assurance

program. SAST identifies and mitigates security vulnerabilities early in the DevSecOps process. It is integrated into the DevSecOps pipeline with so many questions like:

- Work with false positives?
- Identify new issues
- How much time taken to complete the scanning of code,
- What do you mean by "baseline scan"?

4. CONCLUSIONS

We highlight the need for developer-centered application security testing tools that target continuous practices in DevSecOps. More research is needed on how the traditionally manual security practices can be automated to suit rapid software deployment cycles. Finally, achieving a suitable balance between the speed of delivery and security is a significant issue practitioners face in the DevSecOps paradigm.

5. REFERENCES

[1] https://www.sciencedirect.com/science/article/abs/pii/S0950584921001543

[2] https://www.linkedin.com/pulse/devsecops-tools-secure-software-delivery-alexandre-wagner/

[3] https://www.synopsys.com/glossary/what-is-devsecops.html

[4] https://www.tavant.com/

[5] https://www.researchgate.net/figure/Figure-6-Scrum-software-development-Security-Framework_fig5_303679659