WOLDIA UNIVERSITY

COLLEGE OF TECHNOLOGY
SCHOOL OF COMPUTING
DEPARTMENT OF SOFTWARE ENGNEERING

COURSE TITLE: Software Engineering Tools and practice
COURSE CODE : SEng3051
Individual ASSIGNMENT 1

DevSecOps

Name= Abenezer Tariku
ID=1300134
Email= abenimom1994@gmail.com
Github=https://github.com/abenezerTariku

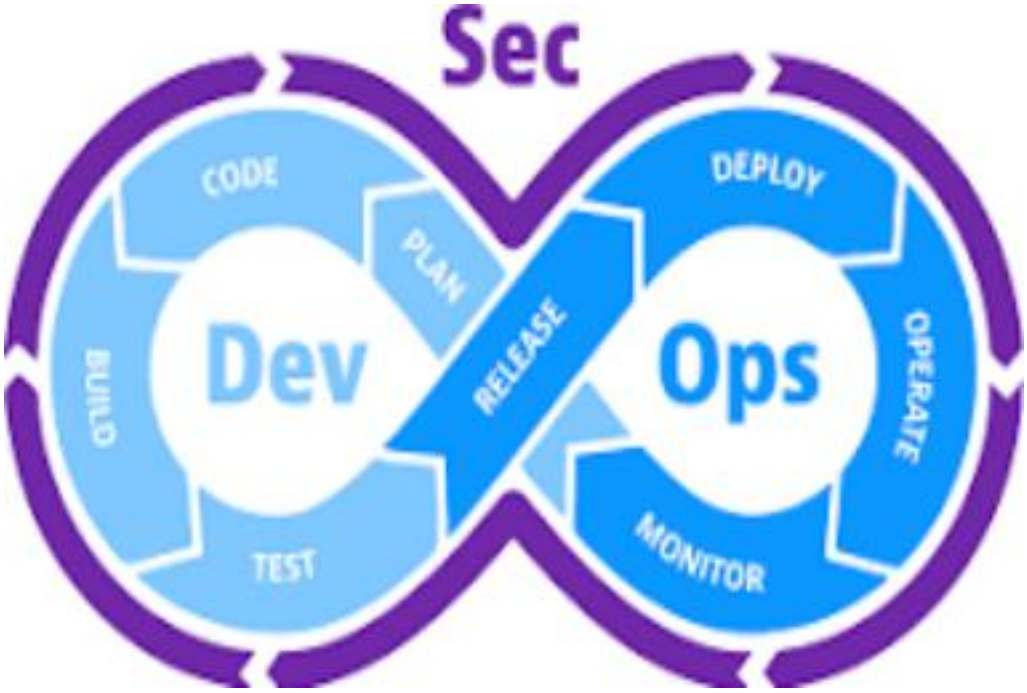
INSTRUCTOR :Esmail M.

Table content

Topics **Page No**

Introduction

DevSecOps, which stands for Development, Security, and Operations, is a method of integrating security principles into the software development lifecycle. It emphasizes the need of addressing security risks early in the development process, rather than as an afterthought. Because of the increased requirement for enterprises to emphasize security in an era of regular cyber-attacks and data breaches, DevSecOps has gained prominence in the software development scene. DevSecOps strives to achieve a balance between agility and security by embedding security into the DevOps approach, allowing teams to release software quickly while ensuring it is strong and resilient against potential security vulnerabilities. DevSecOps arose from the desire to foster a culture of shared responsibility and collaboration among developers.



What are Software engineering problems which was cause for initiation of DevSecOps.?

The DevOps movement is a method to software development that places an emphasis on teamwork, communication, and automation between IT operations teams (Ops) and software development teams (Dev). It attempts to eliminate the conventional silos that have existed between these two functions and advance a culture of shared accountability, continuous improvement, and quicker software delivery.

Businesses frequently struggle to strike a balance between security and speed when implementing software development techniques. In order to meet customer requests and maintain competitiveness, it is important to deploy software rapidly, yet security precautions must not be compromised. Here are a few obstacles that businesses frequently confront when trying to strike the correct balance.

The need for a security-focused approach within the DevOps framework.

For several reasons, a security-focused strategy within the DevOps framework is essential

1. Growing challenges to security
2. Security that shifts left
3. Quick release iterations
4. Requirements for conformity
5. Maintaining client confidence
6. Cooperation and shared accountability
7. Constant security development

What is DevSecOps ?

DevSecOps, which is short for *development*, *security* and *operations*, is an application development practice that automates the integration of security and security practices at every phase of the software development lifecycle, from initial design through integration, testing, delivery and deployment.

DevSecOps means thinking about application and infrastructure security from the start. It also means automating some security gates to keep the DevOps work-flow from slowing down. Selecting the right tools to continuously integrate security, like agreeing on an integrated development environment (IDE) with security features, can help meet these goals. However, effective DevOps security requires more than new tools—it builds on the cultural changes of DevOps to integrate the work of security teams sooner rather than later.

What are DevSecOps life cycles?

Plan

The planning phases of DevSecOps integration are the least automated, involving collaboration, discussion, review, and a strategy for security analysis. Teams must conduct a security analysis and develop a schedule for security testing that specifies where, when, and how it will carry it out.

IriusRisk, a collaborative threat modeling tool, is a well-liked DevSecOps planning tool. There are also tools for collaboration and conversation, like Slack, and solutions for managing and tracking issues, like Jira Software.

Code

Developers can produce better secure code using DevSecOps technologies during the code phase. Code reviews, static code analysis, and pre-commit hooks are essential code-phase security procedures.

Every commit and merge automatically starts a security test or review when security technologies are directly integrated into developers' existing Git workflow. These technologies support different integrated development environments and many programming languages. Some

popular security tools include PMD, Gerrit, SpotBugs, CheckStyle, Phabricator, and Find Security Bugs.

Build

The ‘ build ‘ step begins once developers develop code for the source repository. The primary objective of DevSecOps build tools is automated security analysis of the build output artifact. Static application software testing (SAST), unit testing, and software component analysis are crucial security procedures. Tools can be implemented into an existing CI/CD pipeline to automate these tests.

Dependencies on third-party code, which may come from an unidentified or unreliable source, are frequently installed and built upon by developers. In addition, dependencies on external code may unintentionally or maliciously involve vulnerabilities and exploits. Therefore, reviewing and checking these dependencies for potential security flaws during the development phase is crucial.

The most popular tools to create build phase analysis include Checkmarx, SourceClear, Retire.js, SonarQube, OWASP Dependency-Check, and Snyk.

Test

The test phase is initiated once a build artifact has been successfully built and delivered to staging or testing environments. Execution of a complete test suite requires a significant amount of time. Therefore, this stage should fail quickly to save the more expensive test tasks for the final stage.

Dynamic application security testing (DAST) tools are used throughout the testing process to detect application flows such as authorization, user authentication, endpoints connected to APIs, and SQL injection.

Multiple open-source and paid testing tools are available in the current market. Support functionality and language ecosystems include BDD Automated Security Tests, Boofuzz, JBro Fuzz, OWASP ZAP, SecApp suite, GAUNTLET, IBM AppScan, and Arachi.

Release

The application code should have undergone extensive testing when the DevSecOps cycle is released. The stage focuses on protecting the runtime environment architecture by reviewing environment

configuration values, including user access control, network firewall access, and personal data management.

One of the main concerns of the release stage is the principle of least privilege (PoLP). PoLP signifies that each program, process, and user needs the minimum access to carry out its task. This combines checking access tokens and API keys to limit owner access. Without this audit, a hacker can come across a key that grants access to parts of the system that are not intended.

In the release phase, configuration management solutions are a crucial security component. Reviewing and auditing the system configuration is then possible in this stage. As a result, commits to a configuration management repository may use to change the configuration, which becomes immutable. Some well-liked configuration management tools include HashiCorp Terraform, Docker, Ansible, Chef, and Puppet.

Deploy

If the earlier process goes well, it's the proper time to deploy the build artifact to the production phase. The security problems affecting the live production system should be addressed during deployment. For instance, it is essential to carefully examine any configuration variations between the current production environment and the initial staging and development settings. In addition, production TLS and DRM certificates should be checked over and validated in preparation for upcoming renewal.

The deploy stage is a good time for run time verification tools such as Osquery, Falco, and Tripwire. It can gather data from an active system to assess if it functions as intended. Organizations can also apply chaos engineering principles by testing a system to increase their confidence in its resilience to turbulence. Replicating real-world occurrences such as hard disc crashes, network connection loss, and server crashes is possible.

Operation

Another critical phase is operation, and operations personnel frequently do periodic maintenance. Zero-day vulnerabilities are terrible. Operation teams should monitor them frequently. DevSecOps integration can use IaC tools to protect the organization's infrastructure while swiftly and effectively preventing human error from slipping in.

Monitor

A breach can be avoided if security is constantly being monitored for abnormalities. As a result, it's crucial to put in place a robust continuous monitoring tool that operates in real-time to maintain track of system performance and spot any exploits at an early stage.

How DevSecOps works?

To implement DevSecOps, software teams must first implement DevOps and continuous integration.

DevOps

DevOps culture is a software development practice that brings development and operations teams together. It uses tools and automation to promote greater collaboration, communication, and transparency between the two teams. As a result, companies reduce software development time while still remaining flexible to changes.

Continuous integration

Continuous integration and continuous delivery (CI/CD) is a modern software development practice that uses automated build-and-test steps to reliably and efficiently deliver small changes to the application. Developers use CI/CD tools to release new versions of an application and quickly respond to issues after the application is available to users. For example, AWS CodePipeline is a tool that you can use to deploy and manage applications.

DevSecOps

DevSecOps introduces security to the DevOps practice by integrating security assessments throughout the CI/CD process. It makes security a shared responsibility among all team members who are involved in building the software. The development team collaborates with the security team before they write any code. Likewise, operations teams continue to monitor the software for security issues after deploying it. As a result, companies deliver secure software faster while ensuring compliance.

DevSecOps compared to DevOps

DevOps focuses on getting an application to the market as fast as possible. In DevOps, security testing is a separate process that occurs at the end of application development, just before it is deployed. Usually, a separate team tests and enforces security on the software. For example, security teams set up a firewall to test intrusion into the application after it has been built.

DevSecOps, on the other hand, makes security testing a part of the application development process itself. Security teams and developers collaborate to protect the users from software vulnerabilities. For example, security teams set up firewalls, programmers design the code to prevent vulnerabilities, and testers test all changes to prevent unauthorized third-party access.

Well known DevSecOps Tools

There are so many devscops tools here are some:

1. **GitLab**: GitLab offers a comprehensive DevSecOps platform that includes features for source code management, CI/CD, security scanning, and collaboration tools. It provides built-in security features such as container scanning, dependency scanning, and static code analysis.
2. **Jenkins**: A popular automation server that can be used for continuous integration and continuous deployment (CI/CD) pipelines in a DevSecOps environment.
3. . **OWASP ZAP** (Zed Attack Proxy): An open-source web application security testing tool that helps identify vulnerabilities in web applications.
4. **Chef**: Another configuration management tool that can help automate infrastructure management tasks while ensuring security and compliance.
5. **Docker**: Docker is a containerization platform that allows for the creation and deployment of lightweight, portable containers. DevSecOps

teams use Docker to package applications along with their dependencies, providing a consistent environment for testing and deployment.
6. **SonarQube**: SonarQube is a static code analysis tool that helps identify and fix security vulnerabilities, code smells, and bugs in the codebase. It provides continuous inspection of code quality to ensure that security standards are met.

What are the benefits of DevSecOps?

The value of DevOps is big. Nearly all (99%) of respondents said DevOps has had a positive impact on their organization, according to the 2020 DevOps Trends Survey. Teams that practice DevOps ship better work faster, streamline incident responses, and improve collaboration and communication across teams.

1.Collaboration and trust

Building a culture of shared responsibility, transparency, and faster feedback is the foundation of every high-performing DevOps team. In fact, collaboration and problem-solving ranked as the most important elements of a successful DevOps culture, according to our 2020 DevOps Trends survey.

Teams that work in silos often don't adhere to the systems thinking DevOps espouses. Systems thinking is being aware of how your actions not only affect your team, but all the other teams involved in the release process. Lack of visibility and shared goals means lack of dependency planning, misaligned priorities, finger pointing, and “not our problem” mentality, resulting in slower velocity and substandard quality. DevOps is that change in mindset of looking at the development process holistically and breaking down the barrier between development and operations.

2.Release faster and work smarter

Speed is everything. Teams that practice DevOps release deliverables more frequently, with higher quality and stability. In fact, the DORA “2019 State of

DevOps" report found that elite teams deploy 208 times more frequently and 106 times faster than low-performing teams.

A lack of automated test and review cycles slow the release to production, while poor incident response time kills velocity and team confidence. Disparate tools and processes increase operating costs, lead to context switching, and can slow down momentum. Yet with tools that drive automation and new processes, teams can increase productivity and release more frequently with fewer hiccups.

3.Accelerate time-to-resolution

The team with the fastest feedback loop is the team that thrives. Full transparency and seamless communication enable DevOps teams to minimize downtime and resolve issues faster.

If critical issues aren't resolved quickly, customer satisfaction tanks. Key issues slip through the cracks in the absence of open communication, resulting in increased tension and frustration among teams. Open communication helps development and operations teams swarm on issues, fix incidents, and unblock the release pipeline faster.

4.Better manage unplanned work

Unplanned work is a reality that every team faces–a reality that most often impacts team productivity. With established processes and clear prioritization, development and operations teams can better manage unplanned work while continuing to focus on planned work.

Transitioning and prioritizing unplanned work across different teams and systems is inefficient and distracts from work at hand. However, through raised visibility and proactive retrospection, teams can better anticipate and share unplanned work.

Teams who fully embrace DevOps practices work smarter and faster, and deliver better quality to their customers. The increased use of automation

and cross-functional collaboration reduces complexity and errors, which in turn improves the Mean Time to Recovery (MTTR) when incidents and outages occur.

About Local and international DevSecOps career opportunities, career path.

In the recent years DevSecOPs is highly needed field in the software development process, due to this need demand for professionals who can integrate security practices into the DevOps is highly increasing

1. Local Opportunities: In many regions, including the US, Europe, Asia, and Australia, there is a growing demand for DevSecOps professionals. Companies of all sizes, from startups to large enterprises, are looking for individuals who can help them secure their software development processes. Local job boards, recruitment agencies, and networking events can be good places to find DevSecOps job opportunities in your area.

2. International Opportunities: DevSecOps skills are in demand globally, and many companies are open to hiring remote workers or sponsoring work visas for talented professionals. International job boards, networking platforms like LinkedIn, and specialized websites for tech jobs can help you explore opportunities in other countries.

3. Career Path: A typical career path in DevSecOps may start with roles such as Security Analyst, DevOps Engineer, or Software Developer with a focus on security. As you gain experience and expertise in integrating security practices into the development lifecycle, you may progress to roles like DevSecOps Engineer, Security Engineer, Security Architect, or even Chief Information Security Officer (CISO). Continuous learning, obtaining relevant certifications (such as Certified DevSecOps Professional), and staying updated on industry trends are key to advancing your career in DevSecOps.

Conclusion

In summary, the specific software engineering problems that led to the initiation of DevSecOps include the need to address security vulnerabilities in applications through proactive security measures and practices integrated into the development process.

The initiation of DevSecOps can be attributed to software engineering problems that businesses face when trying to strike a balance between security and speed in software development. The DevOps movement aims to eliminate the silos between IT operations and software development teams and promote a culture of shared accountability and continuous improvement.

Some of the obstacles that businesses commonly confront in achieving this balance include growing security challenges, the need for security to shift left in the development process, the pressure for quick release iterations, the requirement for conformity to security standards, the need to maintain client confidence, the importance of cooperation and shared accountability among teams, and the necessity for constant security development.The benefits of DevSecOps include improved collaboration and trust among teams, faster release and smarter work processes, accelerated time-to-resolution of issues, better management of unplanned work, and enhanced customer satisfaction. DevSecOps professionals are in high demand globally, and there are opportunities for both local and international career growth in this field. The career path in DevSecOps typically starts with roles like Security Analyst or DevOps Engineer and progresses to roles such as DevSecOps Engineer, Security Engineer, Security Architect, or Chief Information Security Officer.

References

1. eForensics Magazine. (n.d.). History of DevSecOps. eForensics Magazine. https://eforensicsmag.com/history-of-devsecops
2. Red Hat. (n.d.). What is DevSecOps? Red Hat. https://www.redhat.com/en/topics/devops/what-is-devsecops
3. Veritis. (n.d.). What are the phases of DevSecOps? Veritis. https://www.veritis.com/blog/what-are-the-phases-of-devsecops/#: :text=Throughout
4. Amazon Web Services. (n.d.). What is DevSecOps? Amazon Web Services. https://aws.amazon.com/what-is/devsecops/#: :text=DevSecOps