WOLDIA UNIVERSITY

INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF SOFTWARE ENGINEERING

SOFTWARE ENGINEERING TOOLS AND PRACTICES

INDIVIUAL ASSIGNMENT

DevSecOps

Name: Gebremeskel Molla

Id: 149181

SUBMITTED TO: MR ESMAEL

SUBMISSION DATE: 5/29/2024 GC

Table of Contents

INTRODUCTION

Getting started with DevSecOps involves shifting security requirements and execution to the earliest possible stage in the development process. It ultimately creates a shift in culture where security becomes everyone's responsibility, not only the security teams. You may have heard teams talking about a "shift left." If you flatten the development pipeline into a horizontal line to include the key stages of the product evolution—from initiation to design, building, testing, and finally to operating—the goal of a security is to be involved as early as possible. This allows the risks to be better evaluated, socialized, and mitigated by design. The "shift-left" mentality is about moving this engagement far left in this pipeline. The goal of DevSecOps is to improve customer outcomes and mission value through the automation, monitoring, and application of security at every phase of the software lifecycle.

Practicing DevSecOps requires an array of purpose-built tools and a wide range of activities that rely on those tools. This document conveys the relationship between each DevSecOps phase, a taxonomy of supporting tools for a given phase, and the set of activities that occur at each phase cross referenced to the tool(s) that support the specific activity.

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

While DevSecOps can benefit your organization in many ways, we've observed several challenges to its adoption, the most common of which are the following:

1.lack of security assurance at the business and project levels

2.organizational barriers related to collaboration, tooling, and culture

3.impact to quality because security is not a priority while systems are getting more complex

4.lack of security skills for developers, business stakeholders, and auditors

5.insufficient security guidance due to lack of resources, standards, and data

1. Lack of Security Assurance at the Business and Project Levels:

- Organizations often prioritize speed and time-to-market over security considerations, leading to a lack of emphasis on security assurance at the business and project levels. This results in inadequate security requirements, limited security testing, and a general lack of oversight throughout the software development lifecycle.

2. Organizational Barriers Related to Collaboration, Tooling, and Culture:

- Siloed teams, resistance to change, and a lack of collaboration between development, operations, and security teams create barriers that hinder the adoption of security practices. Outdated tools and processes, as well as a culture that does not prioritize security, further exacerbate these challenges.

3. Impact to Quality Because Security is Not a Priority While Systems Are Getting More Complex:

- As systems become increasingly complex and interconnected, the impact of security vulnerabilities on software quality becomes more significant. Neglecting security as a priority can lead to poor quality software with vulnerabilities that can be exploited by attackers, resulting in data breaches, system downtime, and reputational damage.

4. Lack of Security Skills for Developers, Business Stakeholders, and Auditors:

- A lack of security skills among developers, business stakeholders, and auditors can result in insecure code, inadequate risk assessments, and compliance gaps. Without the necessary expertise in security

best practices, threat modeling, secure coding practices, and security auditing, organizations are at risk of introducing vulnerabilities into their software products.

5. Insufficient Security Guidance Due to Lack of Resources, Standards, and Data:

- Organizations may struggle to provide adequate security guidance to development teams due to resource constraints, a lack of established security standards, and limited access to relevant security data. The absence of clear guidelines, best practices, and benchmarks for security implementation can impede effective risk management and mitigation efforts.

2. What is DevSecOps?

DevSecOps, which is short for *development*, *security* and *operations*, is an application development practice that makes security an integral part of the entire software development lifecycle. [1]

It emerged as a response to the escalating cybersecurity threats faced by organizations. Rather than treating security as an afterthought, DevSecOps makes security an integral part of the entire software development lifecycle.

It is an extension of the DevOps practice. Each development, security and operations term define different roles and responsibilities of software teams when they are building software applications.

Development: is the process of planning, coding, building, and testing the application.

Security: means introducing security earlier in the software development cycle. For example, programmers ensure that the code is free of security vulnerabilities, and security practitioners test the software further before the company releases it.

Operations: The operations team releases, monitors, and fixes any issues that arise from the software.

It includes tools and processes that encourage collaboration between developers, security specialists, and operation teams to build software that is both efficient and secure. DevSecOps brings cultural transformation that makes security a shared responsibility for everyone who is building the software.[2]

3. DevSecOps lifecycle

Certainly! Let's delve into the DevSecOps lifecycle, a crucial framework that seamlessly integrates security practices throughout the software development process.

Understanding the Imperative of DevSecOps

The **DevSecOps lifecycle** serves as the backbone of security enhancement within the software development continuum. It embodies a structured flow of stages that enables organizations to embed security practices from inception to deployment, fostering a security-centric culture across teams. Here are the key aspects:

1. **Security by Design**: This phase involves embedding security principles at the core of software development processes. By defining foundational security requirements and objectives, teams lay the groundwork for a secure application.
2. **Collaborative Synergy**: Nurturing collaboration among diverse teams is essential. Developers, security experts, and operations personnel work together to enhance the security posture. This collaborative approach ensures that security considerations are woven into every aspect of development.
3. **Risk Mitigation Strategies**: Proactively identifying and mitigating security risks during the lifecycle is critical. Regular security assessments help pinpoint vulnerabilities, allowing timely remediation.
4. **Iterative Security Enhancements**: DevSecOps emphasizes continuous improvement. Teams iteratively enhance security resilience by learning from incidents, adapting processes, and staying vigilant.

Key Phases in the DevSecOps Lifecycle

Let's explore the pivotal phases that define the DevSecOps lifecycle:

1. **Planning and Security Integration**:
    - **Define Security Requirements**: Lay down foundational security requirements and objectives during the planning stage.
    - **Integrate Security Controls**: Incorporate security controls early to align with overarching security goals.
2. **Continuous Integration and Security Testing**:
    - **Automated Security Testing**: Integrate robust security testing tools into the development pipeline for automated assessments. ○ **Vulnerability Identification**: Conduct frequent security assessments to pinpoint vulnerabilities and address them promptly.
3. **Deployment and Configuration Security**:
    - **Secure Deployment Practices**: Implement secure deployment protocols and robust configuration management practices.

- **Infrastructure as Code (IaC)**: Employ IaC principles for consistent, secure deployments across environments.

4. **Monitoring and Incident Response**:
   - **Real-time Monitoring**: Establish vigilant monitoring mechanisms to detect security events and anomalies promptly.
   - **Incident Response Protocols**: Define clear incident response procedures and conduct post-incident analyses for continual enhancement.

- **Real-World Scenario: DevSecOps Lifecycle in Action**

Imagine a tech-savvy software development team launching a new application. By weaving security controls into the planning phase, rigorously testing for vulnerabilities during development, and orchestrating robust monitoring post-deployment, the team successfully fortifies the application against potential threats. This ensures a robust security posture throughout the lifecycle.

In summary, embracing the essence of the DevSecOps lifecycle empowers organizations to build resilient and secure digital solutions. [3]

4.How does DevSecOps works?

DevSecOps works by implementing security policies and automation tools that detect and identify security issues and vulnerabilities while code is being written. These automated processes include security scans, code quality checks, and automated security checks.

As part of the DevSecOps process, the security team also trains the dev and ops teams to interpret the output of these tools. When security tools are integrated into the IaC (Infrastructure-as-Code) pipeline, developers will receive automated output on the application security status, detailing what issues need to be fixed. If there are none, the pipeline will deploy and release the application. [7] Companies use the following approaches to support digital transformation with DevSecOps.

Shift left: is the process of checking for vulnerabilities in the earlier stages of software development. By following the process, software teams can prevent undetected security issues when they build the application. For example, developers create secure code in a DevSecOps process.

Shift right: indicates the importance of focusing on security after the application is deployed. Some vulnerabilities might escape earlier security checks and become apparent only when customers use the software.

Use automated security tools: DevSecOps teams might need to make multiple revisions in a day. To do that, they need to integrate security scanning tools into the CI/CD process. This prevents security evaluations from slowing down development.

Promote security awareness: Companies make security awareness a part of their core values when building software. Every team member who plays a role in developing applications must share the responsibility of protecting software users from security threats.

Continuous Integration (CI): Developers commit their code to a central repository multiple times a day.

Automated integration and testing occur, catching issues early in the process.

This approach ensures that integration problems and bugs are addressed promptly. [2]

5. well known DevSecOps tools.

Most popular tools to apply DevSecOps

As organizations adopt DevSecOps practices, selecting the right tools becomes crucial.

GitLab

GitLab is an end-to-end DevOps platform that integrates source code repositories, CI/CD pipelines, and security scanning tools. Its built-in security features include static application security testing (SAST), dynamic application security testing (DAST), and dependency scanning. **Key Features:**

- Integrated code repository and CI/CD pipelines.
- Automated security testing throughout the development process.
- Scanning containers for Docker images.

Jenkins

An open-source automation server, Jenkins is widely used to build, test, and deploy software. With a wide range of plugins, Jenkins can be extended to include security scanning tools, making it a versatile choice for DevSecOps. **Key Features:**

- Extensive ecosystem of plugins to integrate security tools.
- Pipeline as code to define and manage implementation processes.
- Continuous tracking and reporting capabilities.

OWASP dependency check

The Open Web Application Security Project (OWASP) dependency check is a tool that identifies project dependencies and checks for known and publicly disclosed vulnerabilities. It supports multiple programming languages and integrates well with build systems. **Key Features:**

- Automatic identification of vulnerable dependencies.
- Integration with popular build tools like Maven and Gradle.
- Periodic updates of vulnerability databases.

SonarQube

SonarQube is a platform for continuous inspection of code quality and security. Provides static code analysis and identifies security vulnerabilities, code smells, and bugs.

Key Features:

- Security hotspots to focus on the most critical issues.

- Integration with popular CI/CD tools.
- Real-time feedback to developers.

HashiCorp

HashiCorp Vault is a tool for managing secrets and protecting sensitive data. In a DevSecOps context, it ensures secure storage and access control to secrets used in application development and deployment.

Key Features:

- Centralized secret management.
- Dynamic secret generation.
- Audit log for compliance.

The tools mentioned above provide a solid foundation for integrating security into the DevOps process, allowing organizations to deliver software quickly and securely. [4]

AppScan

AppScan is a popular application security tool produced by HCL Technologies, a leader in the cybersecurity field. Its AI-powered solution is easy-to-use and supports both static and dynamic applications.

Burp Suite

Burp Suite is a leading platform for web application security testing. It offers a variety of tools to help you identify and remediate vulnerabilities and integrates seamlessly into your DevSecOps pipeline.[6]

And there are many tools in DevSecOps.

6.DevSecOps Benefits

DevSecOps aims to help development teams address security issues efficiently. It is an alternative to older software security practices that could not keep up with tighter timelines and rapid software updates.[2]

The purpose and intent of DevSecOps is to build on the mindset that everyone is responsible for security with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required," describes Shannon Lietz, coauthor of the "DevSecOps Manifesto." [1]

"Security is not just the responsibility of a single department or a single individual; it takes a village," said Feferman. "Everyone has to be involved with their security hat on to make sure things are done properly."

More importantly, DevSecOps automates how code is transferred between developers and IT teams, so that they stay in communication continuously, and so that any vulnerabilities in the code are immediately flagged for developers to rectify. [7] DevSecOps provides the following benefits:

- **Fast, cost-effective delivery**—traditional software development methods often result in huge bottlenecks and delays due to security issues. Addressing security flaws and fixing code is often time-consuming and costly. DevSecOps enables faster, secure software delivery to save time and reduce technical debt, thus lowering costs by reducing the need for repeated processes at the end of the delivery cycle.

- **A proactive approach to security**—DevSecOps introduces security processes at the beginning of the software development cycle and ensures the code passes continued reviews, audits, tests, and scans throughout the development pipeline. Development teams can address security issues immediately when discovered, remediating problems before they introduce more dependencies. This approach makes security more effective and less expensive.

- **Fast vulnerability remediation**—DevSecOps helps teams identify security vulnerabilities quickly and apply patches early. It integrates vulnerability detection and patching into the development cycle to prevent the release of the vulnerable software. Early patching also reduces the opportunity for threat actors to exploit vulnerabilities, especially for publicly exposed common vulnerabilities and exposures (CVEs).

- **Automation-driven development**—DevSecOps teams can integrate security testing into automated test suites, enabling streamlined operations. Organizations can leverage continuous integration/continuous delivery (C/CI) pipelines to automate development and security processes. [5]

- **A repeatable and adaptive process--** As organizations mature, their security postures mature. DevSecOps lends itself to repeatable and adaptive processes. DevSecOps ensures that security

is applied consistently across the environment, as the environment changes and adapts to new requirements. A mature implementation of DevSecOps will have a solid automation, configuration management, orchestration, containers, immutable infrastructure and even serverless compute environments. [1]

The main objective of DevSecOps is to introduce security processes early in the development lifecycle, helping reduce vulnerabilities and aligning IT and business objectives with security requirements. [5]

7.About Local and international DevSecOps career opportunities, career path.

Hey there! DevSecOps, a combination of Development, Security, and Operations, is a hot field with plenty of career opportunities both locally and internationally. Companies worldwide are increasingly adopting DevSecOps practices to ensure faster and more secure software delivery.

Local Opportunities:

- **Entry-Level:** Starting as a Junior DevSecOps Engineer or Security Analyst could be your initial step. You'll work on security tasks within agile development teams.

- **Mid-Level:** Progress to positions like DevSecOps Engineer or Security Consultant, where you focus on integrating security practices into the development pipeline.

- **Senior-Level:** At this stage, you could be a DevSecOps Architect, responsible for designing and implementing security strategies across entire organizations.

International Opportunities:

- **Diverse Roles**: In other countries, you may find niche roles like Cloud Security Engineer, Application Security Specialist, or Compliance Analyst.

- **Global Companies:** International corporations often look for DevSecOps professionals to secure their software development processes.

- **Consulting Opportunities:** Consulting firms worldwide offer opportunities to work on various projects, which can enhance your skills and knowledge.

Types of jobs in DevSecOps

You'll find many types of jobs in which you can build a career in DevSecOps. For example, you could become a developer, a tester, an operations engineer, or a security analyst. Here are some roles advertised in DevSecOps environments and their average annual salaries.

- DevSecOps engineer: $116,2351

- DevSecOps software engineer: $124,195
- Cloud security engineer: $102,939
- Cloud and DevSecOps architect: $133,059
- Senior DevSecOps engineer: $124,258
- DevSecOps lead: $126,731

Career Path:

1. Foundation: Start by understanding basic security principles, tools, and processes. Familiarize yourself with popular DevOps practices.

2. Skills Development: Learn scripting (e.g., Python, Bash), automation tools (e.g., Ansible, Terraform), and continuous integration/continuous deployment (CI/CD) pipelines.

3. Certifications: Consider certifications like Certified DevSecOps Engineer or Certified Information Systems Security Professional (CISSP) to enhance your credibility.

4. Experience: Gain hands-on experience by working on projects involving security assessments, code analysis, and security tool implementation.

5. Specialization: Focus on areas like cloud security, container security, or compliance to carve out a niche for yourself.

DevSecOps Engineers typically begin their careers with a strong foundation in software development or security, often as a DevOps or Security Engineer. As they specialize in integrating security into the DevOps process, they may progress to Senior DevSecOps Engineer roles, where they take on more complex projects and mentor junior staff. Advancement can lead to positions like DevSecOps Team Lead or Manager, overseeing multidisciplinary teams and initiatives. With strategic vision and leadership skills, they may ascend to roles such as Chief Information Security Officer (CISO) or VP of Engineering, where they shape security practices and infrastructure at the organizational level. Career growth involves evolving from technical execution to strategic planning and organizational leadership. [8]

Those with skills in DevSecOps will enjoy a long and profitable career. Organizations can’t simply “buy” a DevSecOps software solution. They have to hire people who understand the DevSecOps philosophy, and who can lead teams geared towards greater collaboration and more rapid software delivery.[7]

Remember, the DevSecOps field is dynamic, so staying updated with the latest trends and technologies is key to succeeding in this exciting career path. Good luck on your DevSecOps journey!

CONCLUSION

Practicing DevSecOps requires an array of purpose-built tools and a wide range of activities that rely on those tools. This document conveys the relationship between each DevSecOps phase, there are supporting tools for a given phase, and the set of activities that occur at each phase cross-referenced to the tool(s) that support the specific activity.

REFERENCES

1.What is DevSecOps? | IBM

2. What is DevSecOps? - Developer Security Operations Explained - AWS (amazon.com)

3. DevSecOps Lifecycle - Understand Key Phases (practical-devsecops.com)

4. Best Devsecops Tools | Rootstack

5. DevSecOps: Quick Guide to Process, Tools, and Best Practices | HackerOne

6. Best DevSecOps Tools List in 2024 (practical-devsecops.com)

7. What Is DevSecOps? Exploring the Benefit & Role of DevSecOps (springboard.com)

8. What is a DevSecOps Engineer? Explore the DevSecOps Engineer Career Path in 2024 (tealhq.com)