WOLDIA UNIVERSITY

INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF SOFTWARE ENGINEERING

SOFTWARE ENGINEERING TOOLS AND PRACTICES

INDIVIDUAL ASSIGNMENT

NAME: - Kaleab Bayeh

ID: - 1301691

Submitted To: Esmail M.

Submitted Date: March 20, 2024

Software engineering problems which were cause for initiation of DevSecOps

There are several reasons for example

1. **Lack of Security Expertise**: Developers often lack deep security and compliance knowledge.

   Solution: DevSecOps aims to bridge this gap by integrating security practices into the development process.

2. **Knowledge Gap Between Teams**: Security and operations teams may not be familiar with both infrastructure and software development environments.

   Solution: DevSecOps encourages collaboration and knowledge sharing across multidisciplinary teams.

3. **Manual Security Practices Integration**: Integrating manual security practices into DevOps workflows can be difficult.

   Solution: DevSecOps emphasizes automation, making security checks seamless within the pipeline.

4. **Organizational Culture and Collaboration**: Inter-team collaboration issues hinder successful DevSecOps adoption.

   Solution: Cultivating a security-aware culture and promoting collaboration are essential.

5. **Quality Impact Due to Neglected Security**: As systems become more complex, security often takes a back seat, impacting overall quality.

   Solution: DevSecOps ensures security is a priority throughout the development lifecycle.

6. **Technical Challenges**: Overcoming technical hurdles related to integrating security tools and practices.

   Solution: DevSecOps leverages tools and practices that seamlessly fit into the development pipeline.

7. **Security Assurance at Business and Project Levels**: Ensuring security alignment at both business and project levels.

Solution: DevSecOps aligns security goals with business objectives.

8. **Infrastructure Diversity**: Adapting DevSecOps practices to various types of infrastructures.

   Solution: DevSecOps principles apply regardless of infrastructure type.

What is DevSecOps

DevSecOps, which stands for **development, security,** and **operations**, is a framework that integrates security into all phases of the software development lifecycle. DevSecOps is a variation of DevOps that injects security evaluations into all stages of software development and operations. This approach to building and supporting software promotes collaboration among the different teams that create, secure, and maintain applications. With DevSecOps, security concerns are consistently assessed and addressed as applications are created, deployed, and updated. This idea is illustrated in the image below.

DevSecOps is a trending practice in application security that involves introducing security earlier in the software development life cycle. It also expands the collaboration between development and operations teams to integrate security teams in the software delivery cycle. DevSecOps requires a change in culture, process, and tools across these core functional teams and makes security a shared responsibility. Everyone involved in the SDLC has a role to play in building security into the DevOps continuous integration and continuous delivery (CI/CD) workflow.

Ultimately, DevSecOps is important because it places security in the SDLC earlier and on purpose. When development organizations code with security in mind from the outset, it’s easier and less costly to catch and fix vulnerabilities before they go too far into production or after release. Organizations in a variety of industries can implement DevSecOps to break down silos between development, security, and operations.

DevSecOps lifecycle

key components and stages of the DevSecOps lifecycle:

1. **Continuous Integration (CI)**:

- Developers commit their code to a central repository multiple times a day.
- Automated integration and testing occur promptly.
- This approach helps catch integration issues and bugs early in the process, rather than waiting until the end.
- **Goal**: Improve code quality and streamline collaboration between teams.

2. **Continuous Delivery (CD)**:

- The process of automatically deploying code to production or staging environments.
- Ensures that code changes are consistently delivered to end-users.
- **Goal**: Accelerate software delivery while maintaining reliability.

3. **Automated Security Testing**:

- Throughout the development process, DevSecOps teams conduct:
  - **Code Reviews**: Collaborative examination of code to identify security flaws.
  - **Audits**: Assessments of code and configurations against security standards.
  - **Tests**: Verification of security controls, including vulnerability scanning.
  - **Scans**: Automated checks for vulnerabilities and misconfigurations.
  - **Debugging**: Addressing security issues at the code level.
- **Goal**: Ensure the application meets vital security standards without impeding development speed.

4. **Shift Left Security**:

- In DevSecOps, security is addressed from the very start of the project.
- Teams discuss security implications during planning and begin testing for security issues in development environments.

- **Goal**: Prevent security vulnerabilities by integrating security practices early.

5. **Collaboration and Shared Responsibility**:

   - DevSecOps emphasizes collective ownership of security.
   - All team members, including developers, operations, and security professionals, share responsibility.
   - **Goal**: Foster a culture of security awareness and proactive risk mitigation.

6. **Multicloud Security Strategy**:

   - DevSecOps is a critical component of a multicloud security approach.
   - It ensures consistent security practices across diverse cloud environments.
   - **Goal**: Protect applications and data regardless of the cloud platform.

How does DevSecOps work?

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) are 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 Code Pipeline 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.

well known DevSecOps tools

Edge Delta - Best for real-time observability and analytics

mabl - Best for AI-driven end-to-end testing

Headless forms - Best for creating form endpoints in minutes

Datadog - Best for full-stack application monitoring

GitLab - Best for comprehensive CI/CD in a single application

New Relic - Best for performance engineering and analysis

Flosum - Best for Salesforce-specific DevOps

Octopus Deploy - Best for automated deployments for .NET applications

VMware - Best for managing a multi-cloud environment

Wiz IaC Scanning - Best for identifying risks in Infrastructure as Code

Sysdig - Best for container and Kubernetes security

Honeycomb - Best for observability and debugging in production systems

Benefits of DevSecOps

The two main benefits of DevSecOps are speed and security. Therefore, development teams deliver better, more-secure code faster and cheaper.

“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, co-author of the “DevSecOps Manifesto.”

Rapid, cost-effective software delivery

When software is developed in a non-DevSecOps environment, security problems can lead to huge time delays. Fixing the code and security issues can be time-consuming and expensive. The rapid, secure delivery of DevSecOps saves time and reduces costs by minimizing the need to repeat a process to address security issues after the fact.

Improved, proactive security

DevSecOps introduces cybersecurity processes from the beginning of the development cycle. Throughout the development cycle, the code is reviewed, audited, scanned and tested for security issues. These issues are addressed as soon as they are identified. Security problems are fixed before additional dependencies are introduced. Security issues become less expensive to fix when protective technology is identified and implemented early in the cycle.

Accelerated security vulnerability patching

A key benefit of DevSecOps is how quickly it manages newly identified security vulnerabilities. As DevSecOps integrates vulnerability scanning and patching into the release cycle, the ability to identify and patch common vulnerabilities and exposures (CVE) is diminished. This capability limits the window that a threat actor has to take advantage of vulnerabilities in public-facing production systems.

Automation compatible with modern development

Automation of security checks depends strongly on the project and organizational goals. Automated testing can ensure that incorporated software dependencies are at appropriate patch levels, and confirm that software passes security unit testing. Plus, it can test and secure code with static and dynamic analysis before the final update is promoted to production.

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.

international DevSecOps career opportunities and path

- **What:** DevSecOps integrates security into software development.

- **Why:** Growing cybersecurity threats demand secure practices.
- **Skills**: Security concepts, SDLC knowledge, automation tools.
- **Roles**: Developer, tester, operations engineer, security analyst.
- **Global Market**:
  - **USA**: Competitive salaries, around $120,000 annually.
  - **Australia**: Opportunities in roles like DevSecOps engineer.
  - **Middle East**: Positions available in countries like Lebanon and India.

local DevSecOps career opportunities and path

DevSecOps is a promising field with rising demand, especially in developing countries like Ethiopia. DevSecOps Career Path Foundation must be Start with a solid background in software development. Many DevSecOps engineers begin as software developers or system administrators. The Job Opportunities in Ethiopia it was growing field Ethiopia and the future is secured.

Skills for DevSecOps Engineers

- **Security Concepts**: Understand threat modeling, risk assessment, and vulnerability management.
- **SDLC Knowledge**: Integrate security best practices at every stage of the development process.
- **Automation Tools**: Familiarity with scripting languages (e.g., Python, PowerShell) and automation tools.
- **Cloud Security**: Grasp secure architecture design and configuration management.
- **Container Security**: Learn about Docker and Kubernetes.
- **DevOps Practices**: Understand continuous integration and delivery (CI/CD) and infrastructure as code (IaC).