DevSecOps

Institute of Technology

School of Computing

Department of Software Engineering

Software Engineering Tools and Practices

COURSE TITLE: SOFTWARE ENGINEERING TOOLS AND PRACTICE

COURSE CODE: SEng3051

INDIVIDUAL ASSIGNMENT

STUDENT NAME ID

1. TIENA ABEBAW ----------147400

SUBMITTED DATE: MAY /21/ 2016

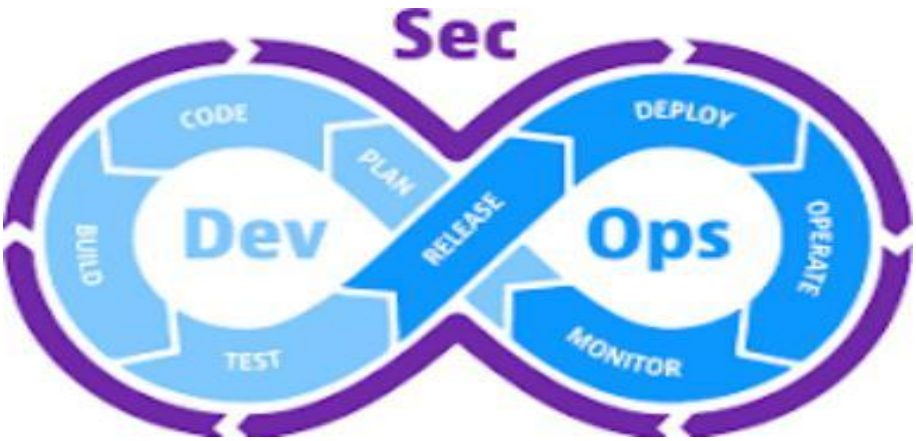
SUBMMITED TO: ISMAEL M.

Table of content

Content page

INTRODUCTION

DevSecOps is a methodology that combines software development (Dev), security (Sec), and IT operations (Ops) to integrate security into every phase of the software development pipeline. It aims to build security into the development process from the very beginning, rather than treating it as an afterthought. By incorporating security practices and tools early on, DevSecOps helps organizations build more secure and resilient software applications. This approach promotes collaboration between development, security, and operations teams to ensure that security is prioritized throughout the development lifecycle. DevSecOps works by integrating security practices into every stage of the software development and operations process, from planning and development to deployment and monitoring. DevSecOps aims to create a culture of security awareness and accountability within organizations, where security is not seen as a separate function but as an integral part of the software development and operations process.



DevSecOps

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

There are several software engineering problems that led to the initiation of DevSecOps. Some of these include:

1. Lack of security awareness: Traditionally, security has been an afterthought in the software development process, leading to vulnerabilities being introduced during development.

2. Siloed teams: In many organizations, security teams operate separately from development and operations teams, leading to a lack of communication and collaboration between these groups.

3. Slow security testing: Traditional security testing processes are often slow and manual, leading to delays in the release of secure software.

4. Lack of automation: Manual security processes are error-prone and time-consuming, making it difficult to maintain security across a large number of applications.

5. Compliance challenges: Meeting regulatory requirements and industry standards for security can be challenging without a coordinated approach to security across the development lifecycle.

6. Lack of Quality: Security is integral to quality. In our observation, lack of quality is often associated with the security team getting involved too late, a lack of confidence in the release, and system complexity.

7. Lack of Security Skills: Developers, architects, scrum masters, and other key players in an organization should have the right vocabularies and skills. By vocabularies, we mean some common knowledge or skillset, or a common understanding, such as a knowledge of how to write secure code.

8. Compliance and regulatory requirements : often add an extra layer of complexity to the software development process, making it difficult to ensure security and compliance at the same time.

9. Complexity: The complexity of modern software systems and the increasing use of third party components and libraries create new attack surfaces for potential security breaches.

2. *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 represents a natural and necessary evolution in the way development organizations approach security. In the past, security was 'tacked on' to software at the end of the development cycle, almost as an afterthought. A separate security team applied these security measures and then a separate quality assurance (QA) team tested these measures.

This ability to handle security issues was manageable when software updates were released just once or twice a year. But as software developers adopted Agile and DevOps practices, aiming to reduce software development cycles to weeks or even days, the traditional 'tacked-on' approach to security created an unacceptable bottleneck.

DevSecOps integrates application and infrastructure security seamlessly into Agile and DevOps processes and tools. It addresses security issues as they emerge, when they're easier, faster, and less expensive to fix, and before deployment into production.

Additionally, DevSecOps makes application and infrastructure security a shared responsibility of development, security and IT operations teams, rather than the sole responsibility of a security silo. It enables ―software, safer, sooner‖—the DevSecOps motto–by automating the delivery of secure software without slowing the software development cycle.

3. *Briefly Explain DevSecOps lifecycle?*

Steps in the Devsecops Lifecycle

DevSecOps is a software development methodology that emphasizes security and collaboration between development, security, and operations teams throughout the software development lifecycle. DevSecOps works best with teams that use CI/CD, or continuous integration and delivery process, meaning code changes are integrated and released as part of an automated process.

The DevSecOps lifecycle can be broken down into the following steps, with the

development, testing, and deployment stages often happening in a loop as software updates are made and new features are added:s

1. Plan

In the planning phase, development teams work with security and operations teams to identify potential security risks and develop a security strategy. This includes identifying security requirements, defining security policies, and selecting the appropriate security testing tools

2. Develop

During the development phase, development teams both build and test the application. This includes integrating automated security testing into the development process, conducting code reviews, and ensuring that security requirements are met.

Since development and testing happen together in the DevSecOps lifecycle, less secure components, such as third-party code, can be tested as they are put into place.

This is where the continuous integration part of the CI/CD process comes in. Code changes are automatically integrated into a shared repository on a regular basis, allowing developers to identify and address conflicts and issues early in the development process.

Optional: Test

Since testing happens during development, a separate testing phase is not necessary in a DevSecOps approach. When it is included, testing takes much less time than it does in a traditional testing process.

During the testing phase, security teams test the application for security weaknesses, vulnerabilities, and threats using penetration testing, vulnerability scanning, and other security testing techniques.

3. Deploy and Monitor

In a traditional process, the operation team would have deployed the application to production. However, the DevSecOps lifecycle follows the DevOps approach, which shifted the responsibility of deploying the application from operations teams to development teams.

The process of deploying to production includes configuring and securing the infrastructure, implementing access controls, and monitoring the environment for security threats.

DevSecOps

Today, many development teams trigger deployments using continuous delivery. This involves the use of tools and processes to automatically build, test, and deploy code changes to production environments.
After deployment, teams then monitor the application for security threats and respond to any incidents that occur.

4. How does DevSecOps works?

DevSecOps

DevSecOps works by integrating security practices into every stage of the software development and operations process, from planning and development to deployment and monitoring. The key principles and practices that guide DevSecOps implementation include:

1. ***Shift Left***: DevSecOps emphasizes shifting security practices and responsibilities to the left in the software development lifecycle, meaning that security is integrated early in the process. By addressing security considerations from the beginning, teams can identify and remediate vulnerabilities sooner, reducing the risk of security incidents in later stages.

2. ***Automation***: Automation plays a crucial role in DevSecOps by enabling teams to implement security controls consistently and at scale. Automated tools are used for tasks such as vulnerability scanning, code analysis, configuration management, and deployment, helping to streamline security processes and reduce manual errors.

3. ***Collaboration***: DevSecOps promotes collaboration and communication between development, operations, and security teams. By breaking down silos and fostering cross-functional teamwork, organizations can ensure that security is a shared responsibility and that all team members are aligned on security objectives and practices.

4. ***Continuous Improvement***: DevSecOps emphasizes continuous improvement through feedback
loops and iterative processes. By regularly assessing security practices, monitoring for vulnerabilities, and implementing lessons learned from security incidents, teams can adapt and enhance their security posture over time.

5. ***Security as Code*:** DevSecOps encourages treating security practices as code, meaning that security controls are defined, implemented, and managed using code-based configurations. This approach allows teams to version control security policies, automate security testing, and integrate

security into the same pipelines used for development and operations.

6. ***Risk Management***: DevSecOps incorporates risk management principles to prioritize security efforts based on the potential impact of vulnerabilities. By conducting risk assessments, threat modeling, and prioritizing security activities based on risk levels, teams can focus on addressing the most critical security issues first.

Overall, DevSecOps aims to create a culture of security awareness and accountability within organizations, where security is not seen as a separate function but as an integral part of the software development and operations process. By adopting DevSecOps practices, organizations can improve the security of their applications, reduce the likelihood of security incidents, and build more resilient and secure software systems.

5.Explain well known DevSecOps Tools?

Sure! Here are some well-known DevSecOps tools that are commonly used in the industry:

1. Git/GitHub/GitLab: Version control systems like Git, along with hosting platforms like GitHub and GitLab, are fundamental tools for collaborative development and version control. They enable teams to manage and track changes to source code, configurations, and infrastructure as code.
2. Jenkins: Jenkins is a popular open-source automation server that supports continuous integration and continuous deployment (CI/CD) pipelines. It allows teams to automate the build, test, and deployment processes, including security checks, and supports integration with various other tools.
3. SonarQube/SonarCloud: SonarQube and SonarCloud are widely used static code analysis tools that help identify code quality issues, security vulnerabilities, and maintainability problems. They provide actionable insights and recommendations to improve code quality and security.
4. OWASP ZAP: OWASP ZAP (Zed Attack Proxy) is a widely used open-source web application security scanner. It helps identify common security vulnerabilities, such as injection attacks, cross-site scripting (XSS), and insecure configurations in web applications.
5. Burp Suite: Burp Suite is a comprehensive web application testing tool that includes a proxy, scanner, and various other utilities. It is commonly used for manual and automated security testing of web applications, including vulnerability scanning and penetration testing.

6. Docker: Docker is a containerization platform that enables developers to package applications and their dependencies into portable containers. It helps ensure consistent and reproducible deployments across different environments, making security controls easier to manage.
7. Kubernetes: Kubernetes is a popular container orchestration platform that automates the deployment, scaling, and management of containerized applications. It provides features for securing containerized workloads, managing secrets, and enforcing access controls.
8. HashiCorp Vault: HashiCorp Vault is a tool for securely managing secrets, such as API keys, passwords, and certificates. It provides a centralized and encrypted storage for secrets, access control mechanisms, and audit logs.
9. Snyk: Snyk is a developer-first security platform that helps identify and fix vulnerabilities in open-source libraries and container images. It integrates with CI/CD pipelines to provide continuous security monitoring and vulnerability scanning.
10. Twistlock/Aqua Security/Sysdig : These are popular container security platforms that provide runtime protection, vulnerability scanning, compliance monitoring, and container image scanning capabilities. They help ensure the security of containerized applications in production environments.

These tools are just a selection from a wide range of DevSecOps tools available. The choice of tools may vary depending on specific requirements, programming languages, and infrastructure used by an organization.

6.what are the 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.

- ***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. This process becomes more efficient and cost effective since integrated security cuts out duplicative reviews and unnecessary rebuilds, resulting in more secure code.

- ***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.
Additionally, better collaboration between development, security and operations teams improves an organization's response to incidences and problems when they occur. DevSecOps practices reduce the
time to patch vulnerabilities and free up security teteams to focus on higher value work. These practices also ensure and simplify compliance, saving application development projects from having to be retrofitted for security.

- ***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***

Cybersecurity testing can be integrated into an automated test suite for operations teams if an organization uses a continuous integration/continuous delivery pipeline to ship their software. 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.

- ***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.

● *Best practices for DevSecOps*

DevSecOps should be the natural incorporation of security controls into your development, delivery and operational processes.

- ***Shift left***

'Shift left' is a DevSecOps mantra: It encourages software engineers to move security from the right (end) to the left (beginning) of the DevOps (delivery) process. In a DevSecOps environment, security is an integral part of the development process from the beginning.
An organization that uses DevSecOps brings in their cybersecurity architects and engineers as part of the development team. Their job is to ensure every component, and every configuration item in the stack is patched, configured securely, and documented.
Shifting left allows the DevSecOps team to identify security risks and exposures early and ensures that these security threats are addressed immediately. Not only is the development team thinking about building the product efficiently, but they are also implementing security as they build it.

- ***Security education***

Security is a combination of engineering and compliance. Organizations should form an alliance between the development engineers, operations teams and compliance teams to ensure that everyone in the organization understands the company's security posture and follows the same standards. Everyone involved with the delivery process should be familiar with the basic principles of application security. They should understand the Open Web Application Security Project (OWASP)

top 10, application security testing and other security engineering practices. Developers need to understand threat models, compliance checks and have a working knowledge of how to measure risks, exposure, and implement security controls

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

DevSecOps is a rapidly growing field that offers a wide range of career opportunities both locally and internationally. As organizations increasingly prioritize security in their software development and operations processes.Here are some insights into local and international DevSecOps career opportunities and potential career paths:

➢ Local DevSecOps Career Opportunities:

1. **Security Engineer:** Security engineers play a crucial role in implementing security measures, conducting security assessments, and ensuring the overall security of software systems. They work closely with development and operations teams to integrate security practices into the software development lifecycle.
2. **DevSecOps Engineer:** DevSecOps engineers are responsible for automating security processes, implementing security controls, and monitoring security metrics in the CI/CD pipeline. They collaborate with cross-functional teams to ensure that security is embedded throughout the development and operations process.
3. **Security Analyst:** Security analysts assess security vulnerabilities, conduct penetration testing, and analyze security incidents to identify potential threats and risks. They work to enhance the security posture of organizations by providing insights and recommendations for improving security practices.
4. **Security Consultant:** Security consultants provide advisory services to organizations on implementing DevSecOps practices, conducting security assessments, and developing security strategies. They help organizations identify security gaps, mitigate risks, and comply with industry regulations and standards.

➢ International DevSecOps Career Opportunities:

DevSecOps

1. **DevSecOps Architect:** DevSecOps architects design and implement secure software architectures, establish security best practices, and oversee the integration of security controls into the software development process. They play a strategic role in shaping the overall security strategy of organizations.
2. **Security Operations Center (SOC) Analyst:** SOC analysts monitor security alerts, investigate security incidents, and respond to cybersecurity threats in real-time. They work in SOC environments to detect and mitigate security incidents, analyze security logs, and maintain the security posture of organizations.
3. **Chief Information Security Officer (CISO):** CISOs are senior executives responsible for leading the organization's cybersecurity strategy, managing the information security program, and ensuring compliance with regulatory requirements. They oversee the implementation of DevSecOps practices to protect sensitive data and mitigate cyber risks.

DevSecOps Career Path:

- **Entry-Level:** Junior Security Analyst, Security Operations Analyst
- **Mid-Level:** DevSecOps Engineer, Security Engineer, Security Consultant
- **Senior-Level:** DevSecOps Architect, Chief Information Security Officer (CISO)

CONCLUSION

There are so many well-known DevSecOps tools that organizations can use to enhance their security practices throughout the software development and operations.These tools help organizations automate security processes, detect vulnerabilities, manage security configurations, and ensure compliance with security standards throughoutsoftware development lifecycle. By integrating these tools into their DevSecOps practices, organizations can strengthen their security posture and build more secure and resilient software systems. Also DevSecOps offers diverse career opportunities locally and internationally.

Reference

- ***https://aws.amazon.com***
- ***https://www.synopsys.com/glossary***
- ***https://www.ibm.com***
- ***https://www.veritis.com***
- ***https://www.browserstack.com***