Institute of Technology

School of Computing

Department of Software Engineering

Software Engineering Tools and Practices

Individual Assignment

Name: ABENEZER TEWODROS

ID: 145267

Table of Contents

Introduction

The introduction for the questions presented in the image. These questions touch upon various aspects of DevSecOps, a critical practice that combines development, security, and operations to enhance software security. Here's a brief overview: DevSecOps emerged as a response to several challenges faced by organizations during software development. These challenges include cultural resistance to change, integrating security seamlessly, tool-centricity, automation maintenance, and defining meaningful metrics. DevSecOps aims to address these issues by embedding security practices throughout the software development lifecycle. DevSecOps extends the principles of DevOps by emphasizing security at every stage of the SDLC. It promotes collaboration, automation, and shared responsibility among development, security, and operations teams. The goal is to create a secure and efficient software delivery pipeline.

The DevSecOps lifecycle involves continuous integration, continuous delivery, and continuous monitoring. It begins with code development, followed by automated testing, deployment, and ongoing monitoring for security vulnerabilities. This iterative process ensures that security is an integral part of software development Some popular DevSecOps tools include:

- SAST (Static Application Security Testing)tools like SonarQube and Checkmarx.
- DAST (Dynamic Application Security Testing) tools like OWASP ZAP and Burp Suite.
- Container Security Tools like Clair and Anchore.
- Infrastructure as Code (IaC) Tools like Terraform and AWS CloudFormation.
- Security Orchestration and Automation Platforms like Demisto and Phantom.

DevSecOps offers several advantages, including improved security posture, faster vulnerability detection, and reduced time to market, better collaboration, and increased overall efficiency. DevSecOps professionals are in high demand globally. Opportunities include roles such as DevSecOps engineers, security architects, penetration testers, and security analysts. Organizations across industries seek experts who can bridge the gap between development and security.

Remember, embracing DevSecOps isn't just about tools—it's a mindset shift that prioritizes security alongside agility and innovation.

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

Software engineering problems that led to the initiation of DevSecOps:

- Lack of collaboration between development and security teams: This often resulted in security vulnerabilities being introduced into software during development, which could only be discovered and fixed later in the software development lifecycle (SDLC).

- Manual and time-consuming security processes: Security testing and vulnerability management were often performed manually, which was time-consuming and error-prone. This made it difficult to keep up with the pace of software development and could lead to security vulnerabilities being overlooked.

- Lack of automation: The lack of automation in security processes made it difficult to scale security to meet the growing demands of software development. This could lead to security vulnerabilities being missed, or to security processes being bypassed altogether.

- Cultural and organizational barriers: There were often cultural and organizational barriers between development and security teams. This could make it difficult to communicate and collaborate effectively, and could lead to security concerns being ignored or dismissed.

- Security and Compliance Knowledge Gap: Developers often lack security and compliance expertise, which is a common challenge in DevSecOps. Similarly, security and operations teams may not be familiar with both infrastructure and software development environments.
- Insufficient Automation: Lack of sufficient automation, technical challenges, and organizational silos were the main reasons behind the slowdown in software development.
- Project-Level Security: Historically, software security has been addressed at the project level, emphasizing code scanning, penetration testing, and reactive approaches for incident response. This approach often fails to align security with business objectives.
- Lack of Continuous Practices: Issues related to DevOps or continuous practices, such as difficulties in integrating manual security practices into DevSecOps, have been identified.

• Tool-Related Issues: Challenges related to tools utilized in DevSecOps, their usage scenarios, and the pipelines, such as vulnerabilities in containers, have been reported.

The initiation of DevSecOps aimed to integrate security practices into the DevOps approach, promoting a more proactive and holistic view of security throughout the software development lifecycle3. This includes practices like iterative and incremental development, continuous feedback, metrics and measurement, automation in every phase of the Software Development Lifecycle (SDLC), and complete engagement with all stakeholders.

DevSecOps addresses these problems by:

• Integrating security into the software development lifecycle: DevSecOps integrates security into all stages of the SDLC, from planning and design to development, testing, and deployment. This helps to identify and fix security vulnerabilities early in the SDLC, when they are less costly and time-consuming to fix.

• Automating security processes: DevSecOps automates security processes, such as security testing and vulnerability management. This helps to improve the speed and accuracy of security processes, and to reduce the risk of security vulnerabilities being missed.

• Breaking down cultural and organizational barriers: DevSecOps promotes collaboration between development and security teams, and helps to break down cultural and organizational barriers. This creates a more cohesive and effective team, and helps to ensure that security concerns are taken seriously.

2. What is DevSecOps?

DevSecOps is a software development approach that integrates security into the software development lifecycle (SDLC) from the very beginning. It is a collaborative approach that involves development, security, and operations teams working together to build secure software.

DevSecOps, short for development, security, and operations, is an application development practice that integrates security practices into every phase of the software development lifecycle.

Some key points about DevSecOps: such as:

• Integration of Security: DevSecOps automates the integration of security at every phase of the software development lifecycle, from initial design through integration, testing, delivery, and deployment.

• Shared Responsibility: It makes application and infrastructure security a shared responsibility of development, security, and IT operations teams.

• Continuous Practices: DevSecOps introduces cyber security processes from the beginning of the development cycle. Throughout the development cycle, the code is reviewed, audited, scanned, and tested for security issues.

• Benefits: The main benefits of DevSecOps are speed and security. Development teams can deliver better, more secure code faster and cheaper.

• Shift Left Security: Another name for this approach is shift left security, which means teams discuss security implications during planning and begin testing for security issues in development environments, rather than waiting until the end.

The goal of DevSecOps is to build on the mindset that everyone is responsible for security, aiming to safely distribute security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required.

Key principles of DevSecOps:

• Security is everyone's responsibility: All members of the software development team, from developers to security engineers, are responsible for security.

• Security should be integrated into the SDLC: Security should not be an afterthought, but rather should be integrated into all stages of the SDLC.

• Automation is key: Security processes should be automated as much as possible to improve speed and accuracy.

- Collaboration is essential: Development, security, and operations teams must work together closely to build secure software.

3. Briefly explain DevSecOps lifecycle?

The DevSecOps lifecycle is a set of practices and processes that integrate security into all stages of the software development lifecycle (SDLC). It is a collaborative approach that involves development, security, and operations teams working together to build secure software.

The DevSecOps lifecycle typically includes the following stages:

- Plan: This phase involves defining the requirements and establishing the goals of the project.

Code: Developers write code in small chunks that can be easily integrated, tested, monitored, and deployed.

- Build: The code is compiled and packaged into a build that can be tested.
- Test: Automated tests are run to ensure the build is functioning as expected.
- Release: The build is released into a staging environment where it undergoes further testing.
- Deploy: If the build passes all tests, it is deployed into the production environment.
- Operate: The team monitors the operation of the software, looking for any issues that may arise.
- Monitor: The software and its environment are continuously monitored to ensure they are functioning correctly.

In each of these stages, security practices are integrated. For example, during the coding phase, developers might use secure coding practices. During the testing phase, automated security tests could be run. And during the operation and monitoring phases, the team would be on the lookout for any security issues that arise3. This approach ensures that security is considered at every step of the software development lifecycle.

The DevSecOps lifecycle is an iterative process. The team should continuously monitor the software for security vulnerabilities and make updates as needed.

Key practices of the DevSecOps lifecycle:

- Continuous integration: The team integrates code changes into the main branch of the code repository frequently, typically several times per day.
- Continuous delivery: The team automates the build, test, and deployment process to deliver software changes to production quickly and reliably.
- Continuous monitoring: The team monitors the software in production for security vulnerabilities and other issues.
- Security testing: The team performs security testing throughout the SDLC to identify and fix vulnerabilities.
- Collaboration: The team collaborates closely to share knowledge and expertise, and to ensure that security is everyone's responsibility.

4. How dose DevSecOps works?

DevSecOps works by integrating security practices into every phase of the software development lifecycle. Here’s a brief explanation of how it works:

How DevSecOps works in practice:

- Security Policies and Automation Tools: DevSecOps implements 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.
- Continuous Integration: With continuous integration, developers commit their code to a central repository multiple times a day. Then the code is automatically integrated and tested. This approach enables teams to catch integration issues and bugs early in the process rather than waiting until the end.

• Shared Responsibility: In DevSecOps, not only does the entire team take responsibility for quality assurance and code integration but also security. In practice, this means teams discuss security implications during planning and begin testing for security issues in development environments, rather than waiting until the end.

• Cultural Transformation: DevSecOps brings cultural transformation that makes security a shared responsibility for everyone who is building the software.

• Reducing Risk: The goal of DevSecOps is to promote the fast development of a secure codebase. Core to DevSecOps is integrating security into every part of the SDLC—from build to production.

By implementing DevSecOps, organizations can ensure that they are developing software that is both efficient and secure.

5. Explain well known DevSecOps tools?

Well-known DevSecOps tools:

• Static code analysis tools: These tools analyze source code to identify potential security vulnerabilities. Examples include SonarQube, Codacy, and Checkmarx.

• Dynamic application security testing (DAST) tools: These tools test running applications to identify security vulnerabilities. Examples include OWASP ZAP, Burp Suite, and Acunetix.

• Software composition analysis (SCA) tools: These tools identify and analyze open source components used in software to identify potential security vulnerabilities. Examples include Black Duck, WhiteSource, and FOSSA.

• Container security tools: These tools help to secure containerized applications. Examples include Docker Security Scanner, Aqua Security, and NeuVector.

• Cloud security tools: These tools help to secure cloud-based applications and infrastructure. Examples include AWS CloudTrail, Azure Sentinel, and Google Cloud Security Command Center.

- Security orchestration, automation, and response (SOAR) tools: These tools help to automate security processes and workflows. Examples include Splunk Phantom, IBM Resilient, and ServiceNow Security Orchestrator.

Other popular DevSecOps tools:

- GitLab: A DevOps platform that includes security features such as static code analysis, DAST, and SCA.
- Jenkins: A continuous integration and continuous delivery (CI/CD) tool that can be integrated with security tools.
- Ansible: A configuration management tool that can be used to automate security tasks.
- Terraform: An infrastructure-as-code tool that can be used to provision and manage secure infrastructure.
- Kubernetes: A container orchestration tool that includes security features such as role-based access control (RBAC) and network policies.

These are just a few of the many DevSecOps tools available. The best tools for a particular organization will depend on the specific needs and requirements of the organization.

6. What are the benefits of DevSecOps?

Benefits of DevSecOps:

- Increased Security: DevSecOps helps in developing high-quality products without compliance issues. It helps developers think critically, understand security requirements, and design the software properly from the beginning.
- Improved Workflow Efficiency: DevSecOps integrates security into every phase of the software development lifecycle, which can lead to improved workflow efficiency.
- Faster Product Releases: With DevSecOps, software teams can automate security tests and reduce human errors, leading to faster product releases4.

• Better Customer Experience: By ensuring the security and quality of software products, DevSecOps can lead to a better customer experience.

• Proactive Security: DevSecOps introduces cyber security processes from the beginning of the development cycle. Throughout the development cycle, the code is reviewed, audited, scanned, and tested for security issues.

• Reduced Time to Patch Vulnerabilities: DevSecOps practices reduce the time to patch vulnerabilities and free up security teams to focus on higher value work.

• Simplified Compliance: These practices also ensure and simplify compliance, saving application development projects from having to be retrofitted for security.

• Security-Aware Culture: DevSecOps helps build a security-aware culture within the organization.

• Improved security posture: DevSecOps helps organizations to build more secure software by integrating security into all stages of the software development lifecycle (SDLC). This helps to identify and fix security vulnerabilities early in the SDLC, when they are less costly and time-consuming to fix.

• Reduced risk of security vulnerabilities: DevSecOps helps to reduce the risk of security vulnerabilities being introduced into production by automating security processes and testing software throughout the SDLC. This helps to ensure that security vulnerabilities are identified and fixed before they can be exploited by attackers.

• Faster time to market: DevSecOps helps organizations to deliver software to market faster by automating security processes and reducing the time it takes to identify and fix security vulnerabilities. This allows organizations to stay ahead of the competition and meet the demands of the modern market.

• Improved collaboration and communication between teams: DevSecOps promotes collaboration and communication between development, security, and operations teams. This helps to break down cultural and organizational barriers and creates a more cohesive and effective team.

• Increased efficiency and productivity: DevSecOps helps organizations to increase efficiency and productivity by automating security processes and reducing the time it takes to identify and fix security vulnerabilities. This allows teams to focus on other tasks, such as developing new features and improving the user experience.

In addition to these benefits, DevSecOps can also help organizations to:

• Meet compliance requirements: DevSecOps can help organizations to meet compliance requirements by providing evidence of security controls and processes.

• Improve customer satisfaction: DevSecOps can help organizations to improve customer satisfaction by delivering more secure software that meets the needs of customers.

• Reduce the cost of security: DevSecOps can help organizations to reduce the cost of security by automating security processes and reducing the time it takes to identify and fix security vulnerabilities.

Overall, DevSecOps is a valuable approach that can help organizations to improve their security posture, reduce the risk of security vulnerabilities, and deliver software to market faster.

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

DevSecOps career opportunities:

DevSecOps is a rapidly growing field, with a high demand for skilled professionals. There are many different career opportunities available in DevSecOps, both locally and internationally.

Local DevSecOps career opportunities

In many countries, there is a high demand for DevSecOps engineers, architects, and managers. These professionals can find work in a variety of industries, including technology, finance, healthcare, and government.

In Ethiopia, there are opportunities for DevSecOps professionals. For instance, Leuwint Technologies is hiring a full-time DevOps engineer.

International DevSecOps career opportunities

DevSecOps is a global field, and there are many opportunities for professionals to work abroad. Many multinational companies have DevSecOps teams in different countries, and there is also a growing demand for DevSecOps professionals in emerging markets.

Globally, there are numerous opportunities for DevSecOps professionals. Companies are increasingly recruiting skilled DevSecOps engineers to their teams. For example, Triple Point Security in Bethesda, MD, USA, is hiring a DevSecOps Engineer II. There are also remote positions available, such as a Sales force DevOps Administrator at Ever light Solar.

DevSecOps career path:

There are many different career paths available in DevSecOps. Some common career paths include:

- Starting Point: Most DevSecOps professionals start their careers in software development or system administration4. Working as a software developer can help you build experience with coding and developing applications. Working in operations or a security role will provide you with experience with the business tools, systems, and processes used to manage and secure software applications.
- Transition to DevSecOps: After gaining experience in development, operations, or security, professionals often transition into DevSecOps. In this role, you'll work with operations staff and developers to ensure that teams design security into the software from the start and that the software environment is secure and monitored continuously.
- Certifications: Various DevSecOps certifications are available that can provide your resume with a boost to help you get onto a DevSecOps career path. For example, Certified DevSecOps Professional (CDP), Certified DevSecOps Expert (CDE), and Certified DevSecOps Leader certification (CDL) are some of the certifications you can pursue.
- Skills Required: To become a pro-DevSecOps engineer, aspiring individuals must have different technical and soft skills in combination. These include a strong understanding of security concepts, knowledge of the SDLC, familiarity with automation tools and scripting

languages like Python and PowerShell, understanding cloud security principles, and knowledge of container security principles.

A career in DevSecOps can offer you the chance to work with cutting-edge technologies, learn valuable workplace skills, and help organizations streamline and enhance their development processes. With the increasing cyber security threats emerging in the corporate landscape, the demand for DevSecOps professionals is expected to grow.

How to get started in a DevSecOps career?

There are many different ways to get started in a DevSecOps career. Some common paths include:

- Earn a degree in computer science or a related field. Many DevSecOps professionals have a bachelor's or master's degree in computer science, software engineering, or a related field.

- Get certified in DevSecOps. There are a number of different DevSecOps certifications available, such as the Certified DevSecOps Engineer (CDSE) certification from the DevOps Institute.

- Gain experience in software development and security. Many DevSecOps professionals have experience in both software development and security. This experience can be gained through internships, personal projects, or work experience.

Conclusion

DevSecOps emerged as a solution to challenges faced during software development. These challenges include cultural resistance to change, seamless security integration, tool-centricity, automation maintenance, and defining meaningful metrics. DevSecOps addresses these issues by embedding security practices throughout the software development lifecycle.

DevSecOps extends DevOps principles by emphasizing security at every stage of the SDLC. It promotes collaboration, automation, and shared responsibility among development, security, and operations teams. The goal is to create a secure and efficient software delivery pipeline. The DevSecOps lifecycle involves continuous integration, continuous delivery, and continuous monitoring. It starts with code development, followed by automated testing, deployment, and ongoing security monitoring. This iterative process ensures security is an integral part of software development. DevSecOps integrates security practices into existing DevOps workflows. It involves using security tools, automating security checks, and fostering a security-first mindset. Collaboration and communication among teams are crucial for successful implementation.

Notable tools include Static Application Security Testing (SAST) tools like SonarQube and Checkmarx, Dynamic Application Security Testing (DAST) tools like OWASP ZAP and Burp Suite, container security tools, Infrastructure as Code (IaC) tools, and security orchestration platforms.

DevSecOps offers improved security posture, faster vulnerability detection, reduced time to market, better collaboration, and increased overall efficiency.

DevSecOps professionals are in demand globally. Roles include DevSecOps engineers, security architects, penetration testers, and security analysts. Organizations seek experts who bridge development and security.

References

(1). https://www.coursehero.com/tutors-problems/Sociology/40310063-Here-are-8-visual-contents-from-the-internet-Identify-their/.

(2). https://www.studocu.com/en-us/document/morehead-state-university/biology-for-your-life/6-chapter-6-questions-and-answers-exam-study-guide/5983784.

(3). https://www.chegg.com/homework-help/questions-and-answers/selecting-mutually-exclusive-projects-technique-used-choose-best-project-payback-period-in-q120174268.

(4). https://slideplayer.com/slide/10857506/.

(5). https://www.chegg.com/homework-help/questions-and-answers/required-information-pr-14-46-algo-introducing-new-product-lo-14-4-14-5-following-informat-q120056841.