WOLDIA UNIVERSITY

COLLEGE OF TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF SOFTWARE ENGNEERING

® COURSE TITLE:SOFTWARE ENGINEERING TOOLS AND PRACTICES.

® COURSE CODE : SEng3051

® INDIVIDUAL ASSIGNMENT

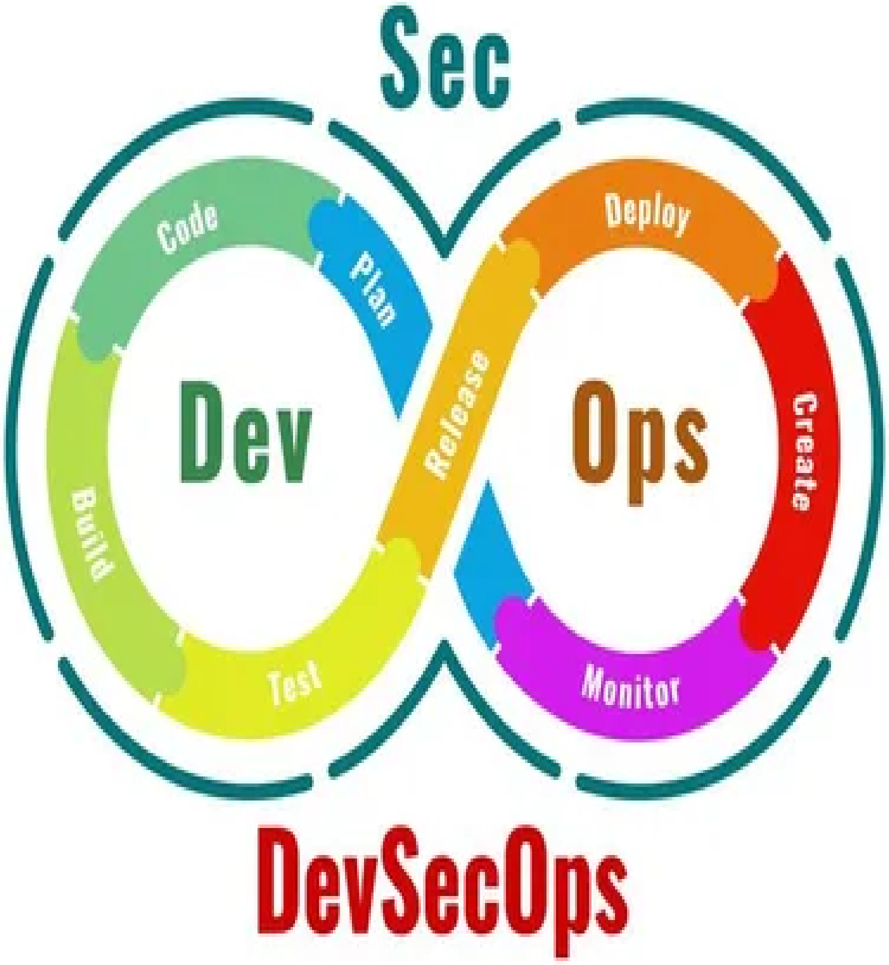
® ACADAMIC YEAR 2016E.C

THIRD YAER FIRST SEMESTER

INSTRUCTOR:- ESMAIL.M

THIS ASSIGNMENT IS DONE By:-

GETAHUN TAMIRAT ID=1301372.



TEBLE CONTENT :-

Assignment Title and question

TITLE :- DevSecOps.

Question:-

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

2, What is DevSecOps?

3 ,Briefly explain DevSecOps lifecycle?

4 ,How dose DevSecOps works?

5, Explain well known DevSecOps tools

6 ,What are the benefits of DevSecOps?

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

Introduction

This document disuse about devsecops.so:-

DevSecOps is a software development methodology that integrates security practices into every stage of the development lifecycle. By combining development, security, and operations processes, DevSecOps aims to proactively address security vulnerabilities and reduce the risk of breaches. Traditional software development lifecycles often treat security as an afterthought, leading to potential vulnerabilities that can be exploited by attackers. By implementing DevSecOps, organizations can improve their security posture, detect and respond to threats more quickly, enhance collaboration between teams, and align security goals with business objectives. This shift towards a more secure development approach has created career opportunities for professionals in roles such as DevSecOps engineer, security architect, security analyst, and security automation specialist. As organizations prioritize security in their software development processes, professionals with expertise in DevSecOps tools, automation, cloud security, and secure coding practices are in high demand.

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

Some of the software engineering problems that led to the initiation of DevSecOps include:

1. Lack of security considerations in the development process: Traditional software development processes often focused more on functionality and speed of delivery, neglecting security aspects. This led to vulnerabilities and security breaches in applications.

2. Soloed teams and lack of collaboration: In many organizations, security teams were separate from development and operations teams, leading to a lack of communication and collaboration. This resulted in security being an afterthought rather than a core part of the development process.

3. Slow response to security threats: Traditional security practices often involved manual security reviews and testing, which were time-consuming and reactive. This delayed the identification and mitigation of security vulnerabilities.

4. Compliance challenges: Meeting regulatory requirements and industry standards for security compliance was a complex and time-consuming process, especially when security was not integrated into the development lifecycle.

5. Inadequate tools and automation: Traditional security tools were not designed to integrate seamlessly with the development and operations processes, making it difficult to automate security testing and monitoring.

6. Lack of visibility and transparency: Without proper monitoring and visibility into the security posture of applications, organizations struggled to identify vulnerabilities and respond effectively to security incidents.

These challenges highlighted the need for a more integrated approach to security in software development, leading to the emergence of DevSecOps as a way to embed security practices into the entire software development lifecycle [1],[2]

2, What is DevSecOps?

DevSecOps is a methodology that integrates security practices into the DevOps process, aiming to build security into every stage of the software development lifecycle. It emphasizes collaboration, communication, and automation among development, operations, and security teams to ensure that security is not an afterthought but a core component of the development process. 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 evolved to address the need to build in security continuously across the SDLC so that DevOps teams could deliver secure applications with speed and quality. Incorporating testing, triage, and risk mitigation earlier in the CI/CD workflow prevents the time-intensive, and often costly, repercussions of making a fix postproduction. This concept is part of "shifting left," which moves security testing toward developers, enabling them to fix security issues in their code in near real time rather than "bolting on security" at the end of the SDLC. DevSecOps spans the entire SDLC, from planning and design to coding, building, testing, and release, with real-time continuous feedback loops and insights.

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]

3. Briefly explain DevSecOps lifecycle?

The DevSecOps lifecycle involves integrating security practices into every stage of the software development process, from planning and design to deployment and monitoring. Here is a brief overview of the DevSecOps lifecycle:

1,Threat Modeling:

The first phase of the DevSecOps lifecycle, thread modeling, helps the team assess an application and its surrounding environment to find as many vulnerabilities as possible before attackers do.

By implementing threat modeling within the traditional development process, teams are able to gather a summary of possible attack scenarios, outline the sensitive data workflow, identify vulnerabilities and potential mitigation options.

Like the majority of the processes in DevSecOps, this is also implemented with the help of tools like OWASP Threat Dragon, IriusRisk, ThreatModeler, etc.

2,Scan & Analyze:

After the threat modeling phase, the code is analyzed in the scanning phase to ensure it is secure from security vulnerabilities. This phase involves both manual and automated code review, which helps developers to identify security vulnerabilities and bugs earlier in the software development life cycle.

This phase involves the use of tools like Static Application Software Testing (SAST) and Dynamic Application Security Testing (DAST).

3,Identity:

After code analysis, the team reviews all the data and metrics collected from the previous phases to identify security risks. These risks are then compiled based on their severity and priority.

Tools like Klocwork can be used to identify security vulnerabilities within the data and metrics collected.

4 ,Remediate:

Once all the security vulnerabilities are identified and organized in the previous phases, the team moves on to the remediation phase, where steps are taken to rectify issues. This involves the use of various SAST tools that suggest solutions for the identified vulnerabilities, errors, and bugs.

This makes it easier for the team to address and rectify the security issues as they arise.

5,Monitor:

Though last, this is another critical phase of the DevSecOps lifecycle, where the team is responsible for tracking all the identified vulnerabilities, the steps taken to mitigate or eliminate those vulnerabilities, and the overall status of the application's security. This allows them to make informed data-driven decisions during the software development lifecycle, which further helps them deliver quality and secure products/features to the users.

Apart from tracking the aforementioned aspects, the team can also track and manage the differences between the actual and target metric values, which will allow the organization to experience advancement in operational efficiency across various departments.

Though there is no concrete process for implementing DevSecOps, these steps are usually present. Depending on the complexity and size of your project, your development lifecycle might include some other sequential steps.

[4],[5]

4. How dose DevSecOps works?

DevSecOps works by integrating security practices into every stage of the software development process, from planning and design to deployment and monitoring. Here are some key principles and practices that make DevSecOps effective:

1. Shift-left approach: DevSecOps emphasizes shifting security practices to the left, meaning that security considerations are introduced early in the development process. By addressing security issues at the planning and design stages, teams can proactively identify and mitigate risks before they become more costly and time-consuming to fix.

2. Automation: Automation is a key component of DevSecOps, enabling teams to automate security testing, code analysis, vulnerability scanning, and compliance checks throughout the development pipeline. Automated tools help identify security vulnerabilities quickly and consistently, allowing teams to address them in a timely manner.

3. Collaboration: DevSecOps promotes collaboration between development, operations, and security teams to ensure that security is everyone's responsibility. By breaking down silos and fostering communication between teams, organizations can build a culture of shared responsibility for security and enable faster response to security threats.

4. Continuous monitoring: DevSecOps emphasizes continuous monitoring of applications and infrastructure to detect security threats in real-time. Monitoring tools provide visibility into the security posture of applications, enabling teams to identify and

respond to security incidents promptly.

5. Security as code: DevSecOps encourages treating security policies, configurations, and controls as code that can be version-controlled, automated, and deployed alongside application code. This approach helps ensure that security controls are consistently applied across environments and can be easily audited and updated.

Overall, DevSecOps is a holistic approach to integrating security into the software development lifecycle, emphasizing collaboration, automation, and continuous improvement to deliver secure and resilient software products]

5. Explain well known DevSecOps tools.

There are several well-known DevSecOps tools that organizations use to integrate security practices into their software development processes. Here are some popular DevSecOps tools:

1. Software Composition Analysis (SCA)

Given the fact that open source software makes up over 90% of the codebase of modern applications, SCA has become an indispensable DevSecOps tool.

Software composition analysis (SCA) tools scan applications to detect and address issues (security vulnerabilities, problematic OSS licenses, and quality issues) in open source code. SCA solutions also offer reporting functionality, including the ability to generate a software bill of materials.

If and when SCA does identify a vulnerability, it provides a host of information (including severity score, inclusion path, and remediation guidance) to help users properly address the issue.

For the open source license compliance use case, SCA inventories the different licenses involved in your code, flagging any components with licenses that violate an organization’s compliance policies.

Finally, modern SCA tools also help teams implement the key DevSecOps principle of delivering *quality* software. SCA offers code quality and provenance checks, helping users identify and upgrade outdated and/or poorly maintained software components.

2. Static Application Security Testing (SAST)

SAST refers to a set of tools that scan codes (source code, binary code, byte code) in a non-running (read: static) state. SAST flags weaknesses in the code it scans, effectively surfacing common issues like CWE-79 (cross-site scripting), buffer overflow errors, SQL Injection, and more.

Much like SCA, SAST flags vulnerabilities and offers remediation guidance. Both tools analyze source code/binaries as opposed to running applications. And, both SCA and SAST are frequently used during the “build” stage of the software development lifecycle, in line with the “shift-left” principle of conducting security testing as early as possible in the SDLC.

There are several significant differences between SCA and SAST, however. While SCA identifies vulnerabilities in open source code, SAST detects vulnerabilities in proprietary code. And, as you might expect, open source license compliance is *not* a SAST use case. DevSecOps teams often use SCA and SAST in a complementary manner.

3. Dynamic Application Security Testing (DAST)

In contrast to SAST and SCA, DAST (Dynamic Application Security Testing) tests for vulnerabilities in a running application. As such, it’s used later in the software development lifecycle.

DAST does not require access to source code. Instead, DAST tools detect vulnerabilities in a running application by (safely) injecting malicious inputs to identify potential security vulnerabilities within the application. A DAST tool will make HTTP requests and uncover issues like SQL injections, OS injections, and cross-site scripting errors. It also finds bugs that are important to application security contexts, like security headers, cookie safety, content security policies, and X-Frame-Options.

There’s no language dependency with DAST tools because they test the running app, however you compile it. DAST also takes into account the context of how the application works: It tests the running application with bad inputs to see how the application behaves. Security teams often use DAST tools as part of their application security suites along with SAST, SCA, and more.

4. Automated Testing Tools

The days of large, dedicated QA teams are a thing of the past for organizations with successful DevSecOps implementations. As the U.S. government’s DevSecOps Fundamentals Guidebook puts it: ”Testing is about automation, and testers will need to become coders of that automation.”

Although some manual testing work will still be required — it's not possible to automate every part of every test — the majority can be automated. For example:

Unit tests: Unit tests analyze individual units of code to make sure they perform as expected. Unit testing tools tend to be language-specific.

Integration tests: Integration tests are performed after unit tests and deal with the interaction between units of code. Again, many of these tests are language-specific.

System tests: System tests are performed after integration tests and analyze the entire application. System testing tools analyze areas like usability, reliability, scalability, and more.

Performance testing, regression testing, and acceptance testing are also among the areas that can be automated.

5. Issue Tracking System

The final tool we'll discuss is one that most teams are likely already familiar with: issue tracking software. Issue tracking systems support several key DevSecOps phases and activities.

Key characteristics of issue tracking tools include:

Automation: Improves engineering efficiency by automating processes like closing issues, notifying customers, assigning issues, and more

Issue resolution tracking and history: Provides visibility and structure to enable efficient bug management. Also creates a record of activities related to issue resolution.

Change management: Equips stakeholders with visibility into new feature development. Offers interactive workflows and roadmaps to support planning and development.

Prioritization management: Enables teams to easily (i.e. drag and drop) prioritize different fixes and activities so that they continuously address

These tools help automate security testing, vulnerability scanning, compliance checks, and other security practices throughout the software development lifecycle, enabling organizations to build secure and resilient applications[7],[9],[11]

6. What are the benefits of DevSecOps?

DevSecOps, which integrates security practices into the DevOps workflow, offers several benefits to organizations. Some of the key advantages of adopting DevSecOps Include: -

1 ,Early Detection of Security Issues: DevSecOps promotes the identification and remediation of security vulnerabilities early in the development process. This helps in addressing issues when they are less costly and time-consuming to fix.

2, Improved Security Posture: By integrating security practices throughout the development lifecycle, DevSecOps helps organizations maintain a strong security posture. It reduces the likelihood of security breaches and data leaks.

3, Faster Response to Threats: DevSecOps encourages real-time monitoring of applications and infrastructure. This enables teams to respond quickly to security threats and incidents, minimizing the potential damage.

4,Automation: Automation is a key component of DevSecOps. Automated security testing, scanning, and compliance checks can significantly reduce the manual effort required for security assessments.

5,Collaboration: DevSecOps fosters collaboration among development, security, and operations teams. This collaboration helps break down silos and ensures that everyone is on the same page regarding security requirements and best practices.

6,Compliance and Auditability: DevSecOps makes it easier to maintain and demonstrate compliance with security standards and regulations. Automated testing and documentation can simplify the audit process.

7,Reduced Risk and Cost: By catching and fixing security issues early, organizations can reduce the risk of security incidents and minimize the potential costs associated with data breaches, regulatory fines, and damage to the company’s reputation.

8,Continuous Security Improvement: DevSecOps is a continuous process. It encourages ongoing security improvement rather than treating security as a one-time activity. This adaptability is crucial in an ever-evolving threat landscape.

9,Scalability: DevSecOps practices can be scaled to suit the needs of both small startups and large enterprises. It adapts to the specific requirements of the organization.

10,Security as Code: Treating security as code means that security policies, tests, and configurations are stored in version-controlled repositories. This ensures consistency and traceability in security practices.

11,Increased Resilience: DevSecOps promotes the design and implementation of applications and infrastructure with an emphasis on resilience. This can help systems withstand security incidents and continue to operate effectively.

12,Cultural Shift: DevSecOps can lead to a cultural shift within an organization, making security a shared responsibility rather than the exclusive domain of the security team.[10]

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

DevSecOps professionals are in high demand both locally and internationally, as organizations across industries recognize the importance of integrating security practices into their DevOps workflows. Here are some insights into local and international DevSecOps career opportunities and career paths:

Local DevSecOps Career Opportunities:

1. IT and Technology Companies: Local IT and technology companies often seek DevSecOps professionals to strengthen their security posture and ensure the secure development and deployment of software applications.

2. Financial Services Sector: Banks, financial institutions, and insurance companies prioritize security due to the sensitive nature of financial data. DevSecOps roles in this sector focus on securing financial systems and applications.

3. Healthcare Industry: Healthcare organizations handle sensitive patient data and must comply with strict regulations like HIPAA. DevSecOps professionals play a crucial role in securing healthcare systems and protecting patient information.

4. Government Agencies: Local government agencies and public sector organizations require DevSecOps experts to secure government systems, infrastructure, and citizen data.

5. Consulting Firms: Consulting firms offer opportunities for DevSecOps professionals to work with a variety of clients across different industries, providing security expertise and guidance on implementing DevSecOps practices.

International DevSecOps Career Opportunities:-

1. Global Technology Companies: International tech giants like Google, Amazon, Microsoft, and Facebook have extensive DevSecOps teams working on securing their platforms and services.

2. Cyber security Firms: International cyber security companies hire DevSecOps professionals to help clients secure their digital assets, conduct security assessments, and implement robust security measures.

3. Financial Institutions: Multinational banks, investment firms, and financial services companies look for DevSecOps experts to strengthen their security defenses and protect customer financial data.

4. Health-Tech Companies: International healthcare technology companies focus on securing healthcare systems, medical devices, and patient data, creating opportunities for DevSecOps professionals with healthcare security expertise.

5. Remote Work Opportunities: With the rise of remote work, DevSecOps professionals can explore international job opportunities with companies that offer remote work options or have distributed teams across different regions.

DevSecOps Career Path:

The career path for DevSecOps professionals typically involves the following progression:

1. Entry-Level Roles: Junior DevSecOps Engineer, Security Analyst, or Security Operations Center (SOC) Analyst roles that involve learning foundational security concepts and tools.

2. Mid-Level Roles: DevSecOps Engineer, Security Engineer, or Security Consultant positions that require hands-on experience with security tools, automation, and integration within the DevOps pipeline.

3. Senior-Level Roles: Senior DevSecOps Engineer, Security Architect, or Security Manager roles that involve leading security initiatives, designing secure systems, and managing security programs within organizations.[12]

Conclusion

DevSecOps represents a significant shift in software development practices by integrating security throughout the entire development lifecycle. This proactive approach addresses security vulnerabilities early on, reducing the risk of breaches and improving overall security posture. By aligning security goals with business objectives, organizations can enhance collaboration between development, security, and operations teams, leading to more secure and resilient applications. The adoption of DevSecOps not only mitigates security risks but also provides career opportunities for professionals in roles such as DevSecOps engineer, security architect, security analyst, and security automation specialist. As organizations prioritize security in their software development processes, professionals with expertise in DevSecOps tools, automation, cloud security, and secure coding practices are in high demand, making it a valuable skill set in today's cybersecurity landscape.

Reference

[1] Program Managers—The DevSecOps Pipeline Can Provide Actionable Data.

[2] An Infrastructure-Focused Framework for Adopting DevSecOps

[3] Explore IBM DevOps solutions

[4] *SANS 2023 DevSecOps Survey*

[5] Build security into DevOps intelligently with Synopsys

[6]www.synopsys.com/software-integrity/contct-devsecops

[8] DEPENDENCY HEAVEN>SOFTWARE COMPOSITION ANALYSIS18 January 2022

[9]https//fossa.com/blog/author/fossa.com

[10]https//www.skillvertex.com/blog/author/hridhya-manoj/

[11] U.S. government's *DevSecOps Fundamentals Guidebook: DevSecOps Tools & Activities*

[12]] from online books