INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING DEPARTMENT OF SOFTWARE ENGINEERING

COURSE TITLE: SOFTWARE ENGINEERING TOOLS AND PRACTICE COURSE

CODE: SEng3051

INDIVIDUAL ASSIGNMENT

NAME **ID**

1. **Daniel Wasihun ----------145817**

SUBMITTED DATE : May/29/ 2024

SUBMMITED TO: Mr. Esmail M.

Contents

I. Introduction

The emergence of DevSecOps was catalyzed by critical software engineering challenges, necessitating a paradigm shift in software development practices to address security concerns effectively. This integrated approach acknowledges the need for security measures to be seamlessly integrated into the development lifecycle, thus mitigating vulnerabilities and enhancing overall software resilience. Evolution of DevSecOps was catalyzed by a series of software engineering challenges that demanded a paradigm shift in approach. These challenges included security vulnerabilities, fragmented security practices, slow and reactive security measures, compliance burdens, and the need for seamless integration within CI/CD pipelines. DevSecOps emerged as a holistic solution to address these issues, emphasizing the integration of security practices into every stage of the software development lifecycle. .

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

Develops challenges refer to the obstacles and difficulties that organizations face when implementing DevOps practices in their software development and IT operations. These challenges can vary depending on the organization's size, industry, existing processes, and cultural factors.

Every successful security plan rests on three pillars:

- ✓ **People,**
- ✓ **Processes, and**
- ✓ **Technology.**

The DevSecOps approach is no different. Its successful implementation relies on better collaboration between Development, Security, and Operations. Nonetheless, a rift between the DevSecOps security and development teams is inevitable in most cases while implementing this strategy.

1.1 DevSecOps Challenges

1. **Lack of Security Assurance**
   - Implement security testing tools and processes early in the development lifecycle to identify and address security vulnerabilities.
   - Conduct regular security assessments and penetration testing to ensure the security of applications.
   - Provide security training and awareness programs for developers and stakeholders to increase security assurance.
2. **Organizational Barriers**
   - Foster a culture of collaboration between development, operations, and security teams by promoting cross-functional teams and communication.
   - Invest in DevSecOps tooling that enables seamless integration of security practices into the CI/CD pipeline.

3. **Lack of Security Skills**
    - Encourage continuous learning and professional development in security for both developers and business stakeholders.
    - Provide training and resources for developers to enhance their security skills, such as secure coding practices and vulnerability detection.
    - Collaborate with security experts or hire external consultants to bridge the gap in security skills within the organization.
4. **Lack of Security Resources**
    - Establish security standards and guidelines within the organization to ensure consistent security practices are followed.
    - Allocate budget and resources for implementing security measures, such as security tools, training, and hiring security professionals.
    - Leverage open-source security tools and resources to supplement existing security capabilities and overcome resource constraints.

2. What is DevSecOps?

Introduced to overcome the flaws of traditional security processes implemented in DevOps, DevSecOps helped remove the silos between the development, security, and operations team. It initiated a transformational shift enabling teams to incorporate secure culture, practices, and tools that drive collaboration, agility, and visibility of security into each phase of the DevOps pipeline, mitigating any potential security risks.

The technology uses security as a code culture, where the security tools are embedded within the DevOps lifecycle to automate the process and decrease the issue remediation time to make the product safer and reliable. DevSecOps follows a process where the emphasis is on building a secure foundation into DevOps initiatives, with everyone being responsible for the product's security.

Moreover, it follows the shift-left approach wherein security processes are embedded early into the design/plan phases of software development to provide complete security awareness to the development and operations teams, fulfilling the critical cybersecurity requirements.

DevSecOps is the practice of integrating security testing at every stage of the software development process. 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.

2.1 What does DevSecOps stand for?

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

- **Development**

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

- **Security**

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.

Software development lifecycle

The software development lifecycle (SDLC) is a structured process that guides software teams to produce high-quality applications. Software teams use the SDLC to reduce costs, minimize mistakes, and ensure the software aligns with the project's objectives at all times. The software development life cycle takes software teams through these stages:

SDLC includes:

- Requirement analysis
- Planning
- Architectural design
- Software development
- Testing
- Deployment

2.2 DevSecOps in the SDLC

In conventional software development methods, security testing was a separate process from the SDLC. The security team discovered security flaws only after they built the software. The DevSecOps framework improves the SDLC by detecting vulnerabilities throughout the software development and delivery process.

3. Briefly explain DevSecOps lifecycle?

With the increasing advancement in technology, the need for enhanced security is also growing drastically. To cater to this requirement, organizations are rapidly adopting new and far more sophisticated technologies that help them tackle various security issues and ensure the quality, reliability, and dependency of the software products/features delivered.

DevSecOps Life Cycle:

DevOps follows a traditional development cycle that involves phases like Plan, Code, Build, Test, Release, Deploy, Operate, and Monitor. Whereas, in DevSecOps, some distinct security steps are integrated into each of the DevOps development phases for thorough security checks, which help organizations build and deliver increasingly secure products at an accelerated rate.

- **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.

- **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).

- **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.

- **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.

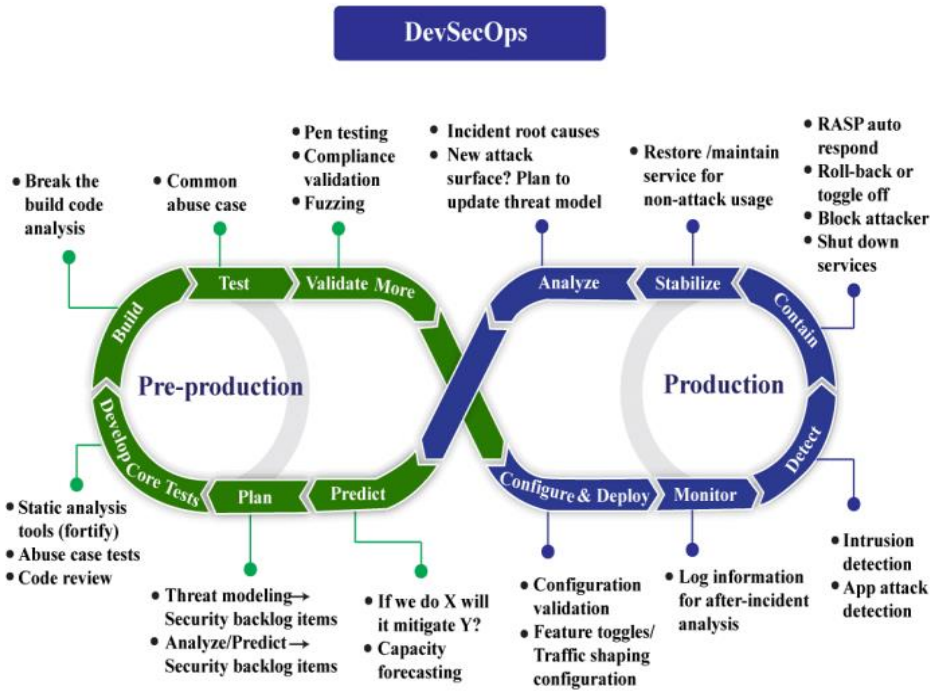
This makes it easier for the team to address and rectify the security issues as they arise.

- **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. How dose DevSecOps works?

DevSecOps operates by embedding security practices into the entire development pipeline, leveraging automation and collaboration to streamline security processes. It involves implementing security controls, integrating security testing tools into CI/CD pipelines, and fostering a shared responsibility for security among development, security, and operations teams. DevSecOps works by embedding security into every aspect of the software development process, from design to deployment. It involves implementing security controls and automation tools, fostering a culture of shared responsibility, and integrating security testing into the CI/CD pipeline. DevSecOps works by integrating security practices and principles into the entire software development lifecycle, from planning and coding to testing, deployment, and operations. Here is a detailed explanation of how DevSecOps functions:

1. Shift Left Approach: DevSecOps emphasizes the "shift left" approach, which means addressing security concerns early in the software development process. By incorporating security practices at the beginning of the development cycle, teams can identify and mitigate security vulnerabilities before they become more costly and time-consuming to fix later in the process.

2. Automation: Automation plays a crucial role in DevSecOps by streamlining security processes and ensuring consistency across development, testing, and deployment stages. Automated security tools can help scan code for vulnerabilities, enforce security policies, monitor infrastructure for potential threats, and facilitate rapid response to security incidents.

3. Collaboration and Communication: DevSecOps promotes collaboration and communication among cross-functional teams, including developers, security professionals, operations engineers, and other stakeholders. By fostering a culture of shared responsibility and transparency, teams can work together to address security issues effectively and proactively.

4. Continuous Integration/Continuous Deployment (CI/CD): DevSecOps leverages CI/CD pipelines to automate the building, testing, and deployment of software applications. Security checks and controls are integrated into these pipelines to ensure that code changes are thoroughly vetted for security vulnerabilities before being deployed to production environments.

5. Security as Code: DevSecOps advocates for treating security configurations, policies, and controls as code that can be version-controlled, tested, and deployed alongside application code. This approach enables teams to manage security configurations programmatically and ensure consistency across environments.

6. Monitoring and Incident Response: DevSecOps emphasizes continuous monitoring of applications and infrastructure for security threats and anomalies. Teams use monitoring tools to detect suspicious activity, respond to security incidents promptly, and implement remediation measures to mitigate risks.

Overall, DevSecOps aims to create a culture of security awareness, collaboration, and automation within organizations to build secure, resilient software applications. By integrating security practices into the DevOps workflow, teams can deliver high-quality software that meets stringent security requirements while maintaining agility and efficiency in the development process.

5. Explain well known DevSecOps tools.

5.1 What is DevSecOps Tools?

DevSecOps tools are a set of software and applications that facilitate the integration of security practices into the software development and operations lifecycle. These tools play a pivotal role in ensuring that security measures are seamlessly woven into every step of the development process – from code creation to deployment and beyond.

5.2 Best DevSecOps Tools List for 2024

The need for robust security tools that integrate seamlessly into the development process has become paramount. Here are some of the best DevSecOps tools list you can choose to deploy

1. **Veracode:** Veracode is an amazing cloud-based security tool created to simplify developer security testing. It provides comprehensive visibility into your application's security posture and offers remediation tips for any vulnerabilities it detects.

2. **Checkmarx:** Checkmarx provides AI-powered software security solutions that help identify and remediate code vulnerabilities. It integrates easily into your development pipeline and provides actionable insights into your security posture.
3. **OWASP ZAP :** OWASP ZAP is a free and open-source web application security scanner. It is highly customizable and can identify vulnerabilities in your application and works by intercepting and modifying HTTP and HTTPS traffic between the web application and client. ZAP has the capability to scan for a range of security issues and includes automated and manual scanning modes.
4. **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.
5. **SonarQube :** SonarQube is a popular code quality tool that offers security-focused plugins to help identify code vulnerabilities during development, provides continuous feedback on your code, and enables you to maintain high code quality.
6. **Fortify :** Fortify is an industry-leading application security tool that offers comprehensive testing capabilities, including static, dynamic, and interactive application security testing. It also offers integrations with leading tools for seamless DevSecOps.
7. **Snyk :** Snyk is a popular developer-first application security tool that integrates directly into your development tools and workflows. It supports multiple languages and offers actionable insight into your app's security posture.
8. **Coverity :**Coverity is a static analysis tool that detects and helps you remediate critical software defects that could impact the security of your application. It also offers integrations with all the leading DevSecOps tools, making it a popular choice for large organizations.
9. **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.
10. **GitLab Secure:** GitLab Secure is a suite of security tools integrated into the GitLab CI/CD platform, including static application security testing (SAST), dynamic application security testing (DAST), dependency scanning, and container scanning.

6. What are the benefits of DevSecOps?

DevSecOps offers several benefits, including improved software quality, faster time-to-market, reduced security risks, enhanced collaboration between teams, and increased overall efficiency and productivity. By integrating security into the development process, organizations can proactively identify and mitigate security threats, thus safeguarding their systems and data

- **Keep pace with modern development methods**

Customers and business stakeholders demand software that is fast, reliable, and secure. To keep up, development teams need to leverage the latest in collaborative and security technology, including automated security testing, continuous integration and continuous delivery (CI/CD), and vulnerability patching. DevSecOps is all about improving collaboration between development, security, and operations teams to improve organizational efficiency and free up teams to focus on work that drives value for the business.

- **Improved Software Security:**

By integrating security practices into every stage of the development lifecycle, DevSecOps helps identify and mitigate security vulnerabilities early in the process. This proactive approach reduces the likelihood of security breaches and ensures that software is built with security in mind from the outset.

- **Faster Time-to-Market:**

DevSecOps streamlines development and deployment processes through automation and collaboration, resulting in shorter development cycles and faster delivery of software updates. By removing bottlenecks and reducing manual intervention, organizations can accelerate their time-to-market without sacrificing security.

- **Reduced Security Risks:**

Continuous security testing and monitoring in DevSecOps enable organizations to detect and remediate security threats in realtime. This proactive stance minimizes exposure to security risks and helps organizations stay ahead of evolving threats, enhancing overall security posture.

- **Enhanced Collaboration:**

DevSecOps promotes collaboration between development, security, and operations teams by breaking down silos and fostering a shared responsibility for security. Collaboration ensures that security considerations are integrated into every aspect of the development process, leading to more robust and secure software.

- **Proactively find and fix vulnerabilities**

Unlike traditional approaches where security is often left to the end, DevSecOps shifts security to earlier in the software development lifecycle. By reviewing, scanning, and testing code for security issues throughout the development process, teams can identify security concerns proactively and address them immediately, before additional dependencies are introduced or code is released to customers.

- **Release more secure software, faster**

If security vulnerabilities aren't detected until the end of a project, the result can be major delays as development teams scramble to address the issues at the last minute. But with a DevSecOps approach, developers can remediate vulnerabilities while they're coding, which teaches secure code writing and reduces back and forth during security reviews. Not only does this help organizations release software faster, it ensures that their software is more secure and cost efficient.

- **Speeding Up Application Development**

In environments without DevSecOps, security issues can cause significant delays in programming. The DevSecOps method removes these obstacles, leading to quicker application development. This approach makes securing code more efficient and cost-effective than traditional methods.

- **Quick Resolution of Security Flaws**

A key benefit of DevSecOps is its quick response to security weaknesses. Dealing with common vulnerabilities during the development phase reduces the risks linked to flaws in development frameworks.

- **Proactive Security Practices**

DevSecOps best practices are to tackle the constantly changing security challenges in software projects. It integrates security throughout the Software Development Life Cycle (SDLC), ensuring continuous evaluation and analysis of code for security risks. This forward-looking strategy helps in early detection and resolution of security issues, preventing them from becoming significant problems.

- **Automated Security Monitoring and Testing**

DevSecOps enhances security monitoring and testing through automation. This method uses automated testing to check and compare actual results with expected ones, either through automated test scripts or testing tools. It also ensures thorough code testing and validation with static and dynamic assessments before integration into the development cycle.

- **Adaptable and Consistent Processes**

As organizations grow, their security needs change. DevSecOps offers flexible and repeatable cycles for consistent security across different environments, even as requirements shift. It encourages collaboration among development, safety, and IT teams, creating a shared responsibility for security. This leads to a more robust and efficient process.

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

DevSecOps professionals have a wide range of career opportunities both locally and internationally, given the increasing demand for individuals with expertise in integrating security practices into the DevOps process. Some of the common career paths and opportunities for DevSecOps professionals include:

7.1 Local DevSecOps Career Opportunities:

1. **Security Engineer:** Security engineers focus on implementing security measures in software development processes, including code reviews, vulnerability assessments, and security testing.
2. **DevSecOps Engineer:** DevSecOps engineers specialize in integrating security practices into the DevOps pipeline, automating security testing, and ensuring compliance with security standards.
3. **Security Analyst:** Security analysts monitor and analyze security threats, conduct risk assessments, and provide recommendations for improving security practices within an organization.
4. **Application Security Specialist**: Application security specialists focus on securing applications by implementing secure coding practices, conducting security assessments, and addressing vulnerabilities.

7.2 International DevSecOps Career Opportunities:

1. **Security Architect:** Security architects design and implement secure systems and applications, develop security policies and procedures, and provide guidance on security best practices.
2. **Cybersecurity Consultant:** Cybersecurity consultants offer expertise in evaluating and enhancing an organization's cybersecurity posture, conducting security assessments, and developing security strategies.

3. **Chief Information Security Officer (CISO):** CISOs are responsible for overseeing an organization's information security program, managing security initiatives, and ensuring compliance with security regulations.
4. **Security Operations Center (SOC) Analyst:** SOC analysts monitor and investigate security incidents, analyze security logs, and respond to cyber security threats in real-time.

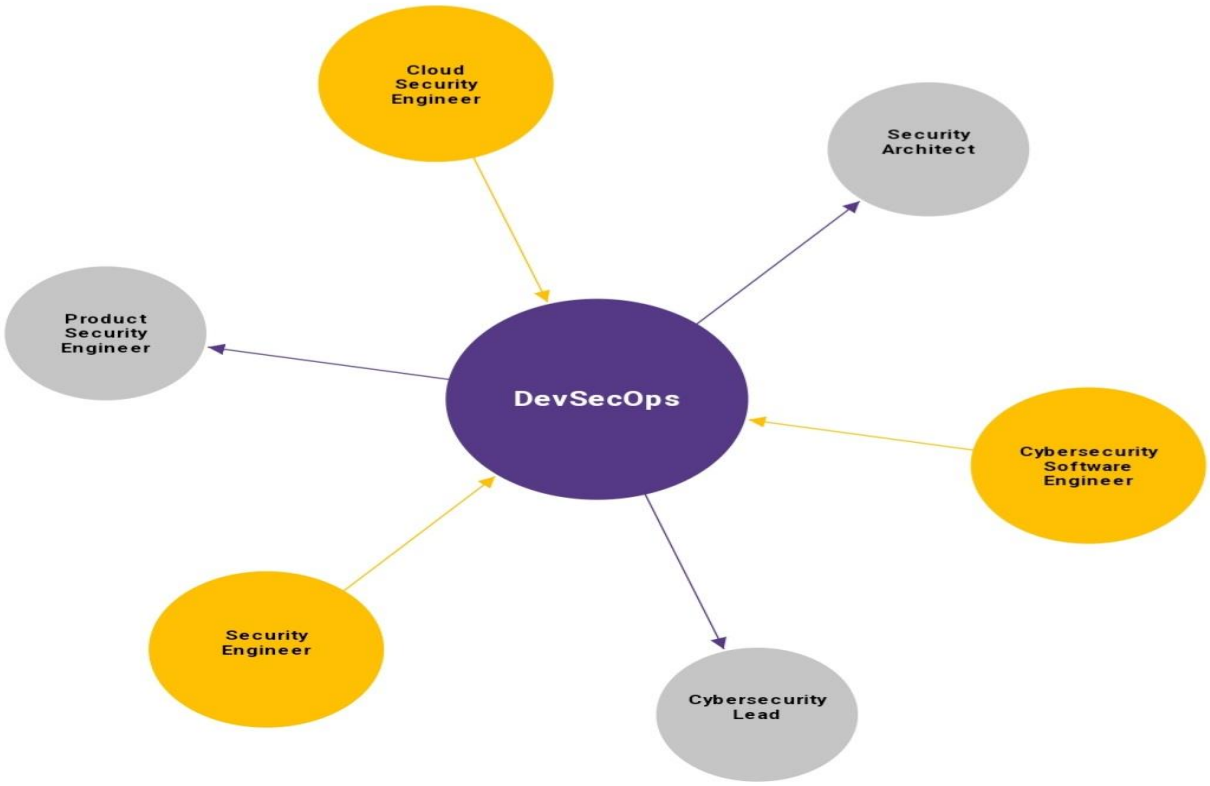


Fig. DevSecOps Careers

7.3 DevSecOps Career Path:

1. **Entry-Level:** Start as a Security Analyst, Junior DevSecOps Engineer, or Security Intern to gain foundational knowledge in security practices and tools.
2. **Mid-Level:** Progress to roles such as DevSecOps Engineer, Security Engineer, or Application Security Specialist, focusing on integrating security into the development process and automating security testing.
3. **Senior-Level:** Advance to positions like Security Architect, CISO, or Cybersecurity Consultant, where you lead strategic security initiatives, design secure systems, and provide guidance on security governance.

To advance in a DevSecOps career path, professionals can pursue certifications such as Certified DevSecOps Professional (CDP), Certified Information Systems Security Professional (CISSP), or Certified Cloud Security Professional (CCSP) to demonstrate expertise in security practices and technologies. Continuous learning, hands-on experience, and staying updated on industry trends are essential for success in the dynamic field of DevSecOps.

CONCLUSION

DevSecOps is a transformative practice that integrates security into every stage of the software development process, promoting collaboration between development, security, and operations teams. By emphasizing shared responsibility for security and implementing tools and processes to enhance security assurance, DevSecOps enables organizations to build software that is not only efficient but also secure. Overcoming challenges such as lack of security assurance, organizational barriers, skills, and resources is essential for successful DevSecOps implementation. By addressing these challenges proactively and adopting strategies to improve security practices, organizations can strengthen their DevSecOps approach and enhance the overall security posture of their software development processes.

References

1. www.microsoftoffice.com
2. www.ibm.com/topics/devsecops
3. https://about.gitlab.com/topics/devsecops
4. https://techvify-software.com/what-is-devsecops/
5. www.atlassian.com/devops-tools/devsecops-tools
6. www.practical-devsecops.com/devsecops-life-cycle/
7. https://www.browserstack.com/guide/devops-lifecycle
8. https://www.practical-devsecops.com/devsecops-tools/
9. https://fossa.com/blog/must-have-devsecops-tools/