INSTITUTE OF TECHNOLOGY SCHOOL OF COMPUTING DEPARTMENT OF SOFTWARE ENGINEERING

COURSE TITLE: SOFTWARE ENGINEERING TOOLS AND PRACTICE

COURSE CODE: SEng3051

STUDENT NAME	ID
1.Dagm Woldekidan --	1304693

SUBMMITED TO: Esmail M.

Table of contents

Topics **page**

Introduction

In the ever-evolving landscape of software development, security has become a paramount concern due to the increasing frequency and sophistication of cyber threats. Traditional software engineering practices often treated security as an afterthought, leading to vulnerabilities and breaches in applications. To address these challenges, the concept of DevSecOps emerged as a proactive approach to integrating security into the entire software development lifecycle. By combining Development, Security, and Operations practices, DevSecOps aims to ensure that security is not just a separate layer but an intrinsic part of the development process. Let's delve deeper into the key aspects of DevSecOps, including its definition, lifecycle, working mechanisms, tools, benefits, and career opportunities.

DevSecOps represents a transformative approach to software development that integrates security practices into every phase of the development lifecycle, ensuring that security is not an afterthought but an integral part of the process. In response to traditional software engineering problems that often led to security vulnerabilities and breaches, the initiation of DevSecOps was a strategic shift towards fostering collaboration between development, security, and operations teams. By embracing the core principles of DevSecOps, organizations aim to enhance their security posture, reduce risks, and accelerate the delivery of secure and resilient applications.

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

Increasing cybercrime and cybersecurity threats in recent years have brought about the new term DevSecOps in the software industry. To keep up with the modern application and software development needs, it is critical for developers and enterprises to adopt DevSecOps.

As technology builders and maintainers, we build, deploy, and maintain applications in order to help our end users by making their day-to-day existence a little bit easier and more streamlined. These end users trust us with their time and data, so it's important that we take all necessary steps to protect them in their online journeys.

- DevSecOps was initiated in response to the growing need for integrating security practices into the software development lifecycle. Traditionally, security was often treated as an afterthought, leading to vulnerabilities and breaches in applications.

  - According to "The State of DevSecOps Report" by GitLab, some of the key software engineering problems that led to the initiation of DevSecOps include lack of collaboration between development, security, and operations teams, slow and manual security processes, and inadequately secured code deployments.

- Explore specific incidents or breaches that highlighted the need for integrating security early in the development process.

  - Discuss common challenges faced by software engineering teams that prompted the adoption of DevSecOps practices.

2. What is DevSecOps?

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?

The DevSecOps lifecycle encompasses a series of stages where security practices are integrated into the software development process from planning to deployment and beyond. Here is an overview of the typical DevSecOps lifecycle stages:

Plan

The planning phases of DevSecOps integration are the least automated, involving collaboration, discussion, review, and a strategy for security analysis. Teams must conduct a security analysis and develop a schedule for security testing that specifies where, when, and how it will carry it out. IriusRisk, a collaborative threat modeling tool, is a well-liked DevSecOps planning tool. There are also tools for collaboration and conversation, like Slack, and solutions for managing and tracking issues, like Jira Software.

Code

Developers can produce better secure code using DevSecOps technologies during the code phase. Code reviews, static code analysis, and pre-commit hooks are essential code-phase security procedures. Every commit and merge automatically starts a security test or review when security technologies are directly integrated into developers’ existing Git workflow. These technologies support different integrated development environments and many programming languages. Some popular security tools include PMD, Gerrit, SpotBugs, CheckStyle, Phabricator, and Find Security Bugs.

Build

The ‘ build ‘ step begins once developers develop code for the source repository. The primary objective of DevSecOps build tools is automated security analysis of the build output artifact. Static application software testing (SAST), unit testing, and software component analysis are crucial security procedures. Tools can be implemented into an existing CI/CD pipeline to automate these tests. Dependencies on third-party code, which may come from an unidentified or unreliable source, are frequently installed and built upon by developers. In addition, dependencies on external code may unintentionally or maliciously involve vulnerabilities and exploits. Therefore, reviewing and checking these dependencies for potential security flaws during the development phase is crucial. The most popular tools to create build phase analysis include Checkmarx, SourceClear, Retire.js, SonarQube, OWASP Dependency-Check, and Snyk.

Test

The test phase is initiated once a build artifact has been successfully built and delivered to staging or testing environments. Execution of a complete test suite requires a significant amount of time. Therefore, this stage should fail quickly to save the more expensive test tasks for the final stage. Dynamic application security testing (DAST) tools are used throughout the testing process to detect application flows such as authorization, user authentication, endpoints connected to APIs, and SQL injection. Multiple open-source and paid testing tools are available in the current market. Support functionality and language ecosystems include BDD Automated Security Tests, Boofuzz, JBro Fuzz, OWASP ZAP, SecApp suite, GAUNTLET, IBM AppScan, and Arachi.

Release

The application code should have undergone extensive testing when the DevSecOps cycle is released. The stage focuses on protecting the runtime environment architecture by reviewing environment configuration values, including user access control, network firewall access, and personal data management. One of the main concerns of the release stage is the principle of least privilege (PoLP). PoLP signifies that each program, process, and user needs the minimum access to carry out its task. This combines checking access tokens and API keys to limit owner access. Without this audit, a hacker can come across a key that grants access to parts of the system that are not intended. In the release phase, configuration management solutions are a crucial security component. Reviewing and auditing the system configuration is then possible in this stage. As a result, commits to a configuration management repository may use to change the configuration, which becomes immutable. Some well-liked configuration management tools include HashiCorp Terraform, Docker, Ansible, Chef, and Puppet.

Deploy

If the earlier process goes well, it's the proper time to deploy the build artifact to the production phase. The security problems affecting the live production system should be addressed during deployment. For instance, it is essential to carefully examine any configuration variations between the current production environment and the initial staging and development settings. In addition, production TLS and DRM certificates should be checked over and validated in preparation for upcoming renewal. The deploy stage is a good time for runtime verification tools such as Osquery, Falco, and Tripwire. It can gather data from an active system to assess if it functions as intended. Organizations can also apply chaos engineering principles by testing a system to increase their confidence in its resilience to turbulence. Replicating real-world occurrences such as hard disc crashes, network connection loss, and server crashes is possible.

Operation

Another critical phase is operation, and operations personnel frequently do periodic maintenance. Zero-day vulnerabilities are terrible. Operation teams should monitor them frequently. DevSecOps integration can use IaC tools to protect the organization's infrastructure while swiftly and effectively preventing human error from slipping in.

Monitor

A breach can be avoided if security is constantly being monitored for abnormalities. As a result, it's crucial to put in place a robust continuous monitoring tool that operates in real-time to maintain track of system performance and spot any exploits at an early stage.

In this article, we will cover the top 10 DevSecOps tools for application security and explore their key features such as application security testing, vulnerability scanning, integration, and reporting. DevSecOps tools is a broad category of solutions, and so in this article we will look at a range of services, including platforms which may cover DevSecOps capabilities as well as havig other capabilities.

Key Phases in the DevSecOps Lifecycle

1. Planning and Security Integration

Define Security Requirements: Lay down the foundational security requirements and objectives.

Integrate Security Controls: Incorporate security controls early in the planning stage to align with overarching security goals.

2. Continuous Integration and Security Testing

Automated Security Testing: Integrate robust security testing tools into the development pipeline for automated assessments.

Vulnerability Identification: Conduct frequent security assessments to pinpoint vulnerabilities and address them promptly.

3. Deployment and Configuration Security

Secure Deployment Practices: Implement secure deployment protocols and robust configuration management practices.

Infrastructure as Code (IaC): Employ IaC principles for consistent, secure deployments across environments.

4. Monitoring and Incident Response

Real-time Monitoring: Establish vigilant monitoring mechanisms to detect security events and anomalies promptly.

Incident Response Protocols: Define clear incident response procedures and conduct post-incident analyses for continual enhancement.

4. How dose DevSecOps works?

To implement DevSecOps, software teams must first implement DevOps and continuous integration.

DevOps

DevOps culture is a software development practice that brings development and operations teams together. It uses tools and automation to promote greater collaboration, communication, and transparency between the two teams. As a result, companies reduce software development time while still remaining flexible to changes.

Continuous integration

Continuous integration and continuous delivery (CI/CD) is a modern software development practice that uses automated build-and-test steps to reliably and efficiently deliver small changes to the application. Developers use CI/CD tools to release new versions of an application and quickly respond to issues after the application is available to users. For example, AWS CodePipeline is a tool that you can use to deploy and manage applications.

DevSecOps

DevSecOps introduces security to the DevOps practice by integrating security assessments throughout the CI/CD process. It makes security a shared responsibility among all team members who are involved in building the software. The development team collaborates with the security team before they write any code. Likewise, operations teams continue to monitor the software for security issues after deploying it. As a result, companies deliver secure software faster while ensuring compliance.

DevSecOps compared to DevOps

DevOps focuses on getting an application to the market as fast as possible. In DevOps, security testing is a separate process that occurs at the end of application development, just before it is deployed. Usually, a separate team tests and enforces security on the software. For example, security teams set up a firewall to test intrusion into the application after it has been built.

DevSecOps, on the other hand, makes security testing a part of the application development process itself. Security teams and developers collaborate to protect the users from software vulnerabilities. For example, security teams set up firewalls, programmers design the code to prevent vulnerabilities, and testers test all changes to prevent unauthorized third-party access.

- DevSecOps incorporates security practices throughout the software development lifecycle, from planning to deployment and operation.

- It emphasizes automation of security controls and testing processes to identify and remediate security vulnerabilities early in the development cycle.

- Collaboration between development, security, and operations teams is essential to ensure timely security assessments and responses.

DevSecOps relies on automation tools for security testing, configuration management, vulnerability scanning, and continuous monitoring of applications.

- By automating security processes and integrating security tools into the development pipeline, DevSecOps enables rapid detection and remediation of security issues.

5. Explain well known DevSecOps tools.

DevSecOps tools are essential in coding, especially when you're dealing with complex projects. They're not just about keeping your code safe; they're also about making your whole development process more efficient.

DevSecOps tools are like the support crew in your development process, handling a lot of the technical and security details so you can focus more on the creative coding part.

DevSecOps tools is a broad category of solutions, and so in this article we will look at a range of services, including platforms which may cover DevSecOps capabilities as well as havig other capabilities.

1. Aikido Security

Aikido Security is an automatic web application security platform, designed specifically for software development teams. It consolidates various application scanning tools within a single platform, with key features including cloud posture management, open source dependency scanning, secrets detection, static code analysis, infrastructure as code scanning, and container scanning. In addition, the platform provides continuous surface monitoring, open source license scanning, malware detection in dependencies, and end-of-life runtime scanning.

The platform is designed to integrate seamlessly into your existing tech stacks and language, offering versatility to adapt to any configuration. Aikido can be integrated with your pre-existing task management tools, messaging utilities, compliance suites, and continuous integration systems, making it possible to monitor and address issues within your current toolset.Aikido provides comprehensive vulnerability alerting, while reducing false positives. It automates alert prioritization with deduplication of recurring alerts, automatic triaging, and customizable rules engine to sift out irrelevant alerts. Aikido also converts Common Vulnerabilities & Exposures data into plain language, facilitating rapid, precise threat response.

Aikido ensures data privacy by conducting scans within temporary environments, and deleting them post-analysis. The platform is unable to alter source code and requires read-only access to ensure protection for your code base. Aikido is compliant with AICPA's SOC 2 Type II & ISO 27001:2022. Aikido provides a reliable security tool for software development teams requiring comprehensive web application security screening.

2. Acunetix

Acunetix is an application security testing solution used by over 2,300 companies of various sizes to automate web application security. The software creates a comprehensive list of websites, applications, and APIs to ensure no potential entry points are left unscanned and, therefore, vulnerable to attack.

Acunetix is capable of crawling and scanning even the most complex web applications, including those built with HTML5 and JavaScript. Its advanced detection features can identify over 7,000 vulnerabilities, including zero-day threats. The software is designed for fast, efficient scanning that alerts users to vulnerabilities the moment they are found, providing more complete coverage with blended Dynamic Application Security Testing (DAST) and Interactive Application Security Testing (IAST) methods.

In addition to detection, Acunetix offers practical tools for resolving vulnerabilities quickly. By automating manual tasks and reducing guesswork, security professionals can save time and resources. Acunetix minimizes false positives with proof of exploit and helps pinpoint the exact lines of code that need to be fixed, enabling developers to address security issues independently.

3. Aqua Security

Aqua Security is a unified cloud security company that offers protection for the entire development lifecycle. The platform discovers and remediates vulnerabilities, malware, exposed secrets, and other risks in code, build tools, and delivery pipelines. With Aqua, users can gain visibility into every resource and risk across the development lifecycle, enabling them to understand their security posture, make informed security decisions, and provide compliance reports to auditors and management.

Aqua Security's platform is compatible with various environments, including clouds, containers, serverless platforms, CI/CD pipelines, registries, and DevOps tools. It also supports multiple compliance frameworks, such as PCI and SOC2, simplifying the process of achieving and maintaining compliance. Aqua Security is trusted by Fortune 1000 customers in over 40 countries.

The Aqua Cloud Native Application Protection Platform (CNAPP) provides total lifecycle visibility, risk reduction, and attack prevention with its fully integrated system. Founded in 2015, with headquarters in Boston, MA, and Ramat Gan, IL, Aqua Security helps clients reduce risk and build a secure future for their businesses.

4. Checkmarx One

Checkmarx One is a comprehensive application security platform designed to help companies secure their digital transformations throughout the entire application development process. This platform is suitable for CISOs, AppSec teams, and developers, ensuring secure application development without compromising speed.The platform offers a complete suite of application security testing (AST) solutions, including Static Application Security Testing (SAST), Software Composition Analysis (SCA), Supply Chain Security (SCS), API Security, Dynamic Application Security Testing (DAST), Container Security, and Infrastructure as Code (IaC) Security. Checkmarx One uses its Fusion engine to seamlessly secure applications by correlating findings between AST solutions, identifying the most critical vulnerabilities, and reducing management overhead.Developers benefit from a seamless experience with Checkmarx, featuring IDE integration, bug ticketing, guided remediation, and security learning. The platform allows developers to efficiently fix security issues and receive just-in-time learning via Checkmarx Codebashing, all without leaving their preferred IDE. Checkmarx, the Enterprise Application Security provider, serves over 1,800 customers, including 60 percent of Fortune 100 organizations.

5. Codacy Quality

Codacy Quality is used by 600,000 developers worldwide to improve code quality, security, and performance. The company offers a suite of products designed to help developers optimize their code and create efficient solutions.

Codacy streamlines the code review process by monitoring and enforcing code quality, test coverage, and security standards. It provides developers with actionable insights to fix potential issues before they arise. It also monitors, maintains, and improves test coverage. Additionally, its AI-assisted features suggest fixes that developers can directly apply in their Git workflows.

The platform integrates seamlessly with developers' existing Git tools, such as GitHub, BitBucket, and GitLab, and offers full visibility of all applications in a single dashboard for easy benchmarking and performance assessment. Codacy also includes security and risk management dashboards to help users identify, prioritize, and fix critical security issues. With a focus on keeping customer data protected, Codacy Quality provides an effective solution for increasing code quality, security, and performance for developers and engineering teams.

6. Fortify by OpenText

Fortify by OpenText offers a comprehensive and extensible application security platform, designed to integrate seamlessly with various tools within the software development life cycle (SDLC). The platform provides extensive DevSecOps integrations, scalable application security, and flexible deployment options, including managed services, cloud-hosted solutions, and on-premises data centers.

Core capabilities include secure developer training, an extensive AppSec ecosystem, AppSec orchestration, Fortify Insight (which provides a single-pane-of-glass view of enterprise security), and automated results auditing using machine learning-assisted technology. Fortify solutions cater to different customers' needs, including Fortify on Demand for security testing and vulnerability management, Software Security Center for managing software security activities, Fortify Hosted for dedicated cloud deployment, and Fortify Insight for effective application security program management.Recognized as a market leader by industry analysts, Fortify by

OpenText continues to expand its offerings to cover critical use cases, from DevSecOps and cloud transformation to securing the software supply chain.

7. GitLab

GitLab is a comprehensive DevOps platform. GitLab contributes to faster software delivery by reducing cycle time from weeks to minutes, cutting development costs and time to market while enhancing overall developer productivity. GitLab's platform is AI-powered, boosting the efficiency of users across the software development lifecycle, from planning, code creation, testing, security to monitoring. This all-in-one DevSecOps solution includes integrated security throughout its single data model, offering insights across the entire lifecycle.

GitLab's deployment options include SaaS, self-managed, and GitLab Dedicated for clients seeking data isolation and residency. GitLab's multi-cloud strategy avoids vendor lock-in and allows deployment anywhere.

GitLab supports various features, including artificial intelligence and machine learning, software supply chain security, value stream management, source code management, continuous integration and delivery, GitOps, and agile project and portfolio management. GitLab is used by over 30 million users, including 50% of Fortune 100 companies.

8. Palo Alto Networks Prisma Cloud

Palo Alto Networks Prisma Cloud is a comprehensive Cloud Native Application Protection Platform (CNAPP) that provides extensive security and compliance coverage for infrastructure, workloads, and applications throughout the development lifecycle in hybrid and multicloud environments. With over 1,900 customers, Prisma Cloud secures more than 4 billion cloud resources and processes over 1 trillion cloud events daily.

Prisma Cloud offers a broad range of security capabilities, including code security, cloud security posture management, cloud workload protection, web application and API security, and cloud infrastructure entitlement management, to ensure comprehensive coverage for cloud-native architectures and toolkits.

The platform simplifies security management by integrating multiple security features into a single solution, such as prevention-first protection and enhanced application delivery. The solution addresses the challenges caused by point security tool sprawl and enables security and DevOps teams to collaborate effectively, accelerating secure cloud-native application development.

9. Snyk Logo

Snyk is a developer security platform designed to support the modern development landscape by integrating directly into development tools, workflows, and automation pipelines. The platform allows teams to easily discover, prioritize, and fix security vulnerabilities in code, dependencies, containers, and infrastructure as code. Snyk's industry-leading security intelligence ensures a high level of accuracy in addressing various security concerns.

The Snyk platform provides a unified solution for securing proprietary code, open source dependencies, container images, and cloud infrastructure. Its developer-first approach empowers developers to maintain code security throughout the development process, while its DeepCode AI enables increased accuracy and productivity in scans and suggested code fixes. Snyk also supports seamless integration with DevSecOps, automating security tasks to save time and reduce human error.

In addition to its powerful security tools, Snyk offers easy integration throughout the Software Development Life Cycle (SDLC) by weaving security expertise into existing tools and workflows. This enables developers to find and fix vulnerabilities without the need for additional applications. Snyk also provides governance at scale, allowing organizations to standardize security protocols and enforce best practices across all applications. Snyk delivers a comprehensive security platform that adapts to the changing needs of application and cloud developers.

10. Veracode

Veracode is a software security platform that utilizes artificial intelligence to identify and rectify flaws and vulnerabilities throughout the software development lifecycle. The platform is trusted by security teams, developers, and business leaders from thousands of leading global organizations.

Veracode’s security tools integrate seamlessly into existing development toolchains, providing fast, accurate, and reliable results with minimal interference in the development process. Veracode offers a comprehensive suite of solutions, including Static Analysis, Static Analysis IDE Scan, Static Analysis Pipeline Scan, Software Composition Analysis, and Secure Code Training, to help developers create secure software with confidence.

The platform also aids in delivering a successful DevSecOps program by unifying development and security features. This includes providing security teams with a holistic view of their organization’s security posture, continuous scanning throughout the software development process, and meeting various data residency requirements. Veracode’s cloud-native SaaS architecture offers added benefits such as elastic scalability, high performance, and cost savings. With a proven track record and a global customer base, Veracode is a reliable choice for organizations aiming to improve their software security and development efficiency.

All of the best DevSecOps tools integrate well with CI/CD, encounter a good community, and promise scalability. Though they do differ in some aspects. Let’s break down their prowess with a quick DevSecOps tools comparison table.

Must-have features in DevSecOps tools

When you're diving into the sea of DevSecOps tools and techniques, it's crucial to know what floats and what sinks. Here's a list of features to absolutely look for in the first place:

– Integration: The MVPs of DevSecOps tools play nice with your existing tech stack. Look for tools that easily integrate into your development pipeline, ensuring a smooth workflow without the headache of compatibility issues.

– Automatic web application security checks: Time is money, and in the coding universe, it's also the key to staying ahead of the game. Top-notch DevSecOps tools automate security checks like a silent guardian. They catch vulnerabilities on the fly, saving you from late-night debugging sessions.

– Real-time threat intelligence: You need tools with radar and threat modeling. Opt for those armed with real-time threat intelligence, so you're not just reacting to yesterday's threats but staying one step ahead.

– User-friendly interface: Let's keep it real — nobody has time for a tool that requires a PhD to operate. Your ideal DevSecOps security tools are user-friendly, with an interface that even your coffee-deprived coder at 3 AM can navigate without a hitch.

– Scalability: Your code is destined for greatness, so your tools better grow with it. Choose DevSecOps tools that scale effortlessly as your projects evolve, ensuring they're not just for now but for the next big thing.

– Compliance: With so many regulations and standards, your tools should make compliance quick and painless. Look for those that understand and align with industry standards, saving you from regulatory headaches down the road.

Effective DevSecOps tools quietly fortify your code. Keep an eye on these features, and your toolkit will be the envy of every developer on the block.

<u>6. What are the benefits of DevSecOps?</u>

"The purpose and intent of DevSecOps is to build on the mindset that everyone is responsible for security with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required," describes Shannon Lietz, co-author of the "DevSecOps Manifesto."

Rapid, cost-effective software delivery

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

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 teams 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 server less compute environments.

Reduced security error and associated costs

By addressing security issues throughout the development lifecycle rather than after release, organizations can catch and address issues earlier, when the time and cost to resolve them is much lower. These costs include lost dollars,additional effort and rework ,and customer goodwill

Reduced time to deploy

Catching flaws and vulnerabilities through constant testing during the lifecycle reduces time to deploy, reduces time spent after deployment on error response, and improves your readiness to deploy .

- Improved security posture with proactive identification and mitigation of security vulnerabilities.

- Faster delivery of secure software through automation and integration of security processes.

- Enhanced collaboration between development, security, and operations teams.

- Reduced security risks and compliance issues through continuous security monitoring and testing.

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

A DevSecOps career 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 different routes into this career, you'll find various DevSecOps certifications available that can provide your resume with a boost to help you get onto a DevSecOps career path.

DevSecOps combines information security best practices with the ability to integrate and deploy software changes continuously. The combination of DevOps and Sec can improve software reliability, security, and quality. DevSecOps is an approach to development that grew out of DevOps. Rather than considering security in late development and post-development phases, DevSecOps makes security integral to development activities through the development lifecycle.

What does a DevSecOps professional do?

A DevSecOps professional is responsible for the security of the software development process, including automating scans, code verification, and developing security protocols. 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.

Professional Certificate

Microsoft Cybersecurity Analyst

Launch your career as a cybersecurity analyst. Build job-ready skills for an in-demand career in the field of cybersecurity in as little as 6 months. No prior experience required to get started.Skills you'll build:

Cloud Computing Security, Computer Security Incident Management, Network Security, Penetration Test, Threat mitigation, Computer Architecture, Cybersecurity, Cloud Computing, Operating Systems, Network Monitoring, Computer Network, Information Security (INFOSEC), Encryption techniques, threat intelligence, Compliance techniques, Authentication Methods, Access Management, Enterprise security, Identity governance, Event Management, Security Response, System Testing, Security Testing, Cybersecurity planning, Record management, Data Management, Cloud Architecture, Threat Model, Access Control, Asset Management, Cybersecurity strategies, Regulatory Compliance, Security Analysis

How do you start a career in DevSecOps?

Experience is highly prized when employers are looking at DevSecOps job applicants. You'll find different routes to working in this function. You can take various jobs to help you prepare for a DevSecOp role. The important thing is to get some valuable experience before moving into the pressure of a security-focused role.

For example, working as a software developer can help you build experience with coding and developing applications. This job can give you experience in the Dev side of the role. 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.

Should you opt to pursue a college degree, research which major would be most beneficial for your career goals. Depending on the roles you're targeting, you might choose a degree that focuses on cybersecurity or a degree that is more software development-focused.

Attending conferences and workshops can demonstrate that you're keeping up with the latest security trends. Additionally, you can enhance your resume by taking courses and certifications. You'll want to make your resume as appealing as possible to potential employers.

Types of jobs in DevSecOps

You’ll find many types of jobs in which you can build a career in DevSecOps. For example, you could become a developer, a tester, an operations engineer, or a security analyst. Here are some roles advertised in DevSecOps environments and their average annual salaries.

- DevSecOps engineer:
- DevSecOps software engineer:
- Cloud security engineer:
- Cloud and DevSecOps architect:
- Senior DevSecOps engineer:
- DevSecOps lead:

Skills needed in DevSecOps jobs

When you work in DevSecOps, you'll bring security to the heart of software development and deployment. You'll need an understanding of the organization’s development and operational side and will have programming and infrastructure knowledge to ensure that security becomes a vital part of the software lifecycle. To get a DevSecOps job, you'll need to demonstrate both technical and workplace competencies that map to your target role.

Technical skills

You must quickly adapt and learn new technologies in the ever-changing business and technology landscape. Having the capacity to troubleshoot and resolve technical issues fast is critical in this role. Here are some of the top DevSecOps skills you'll see in job advertisements.

- Understanding of code development and scripting languages like Java, C++, XML, and JSON
- Familiarity with automation tools like Puppet, Chef, and Ansible
- Experience with cloud technologies for cloud DevSecOps
- Working knowledge of security concepts and tools like firewalls, intrusion detection/prevention systems, and encryption
- Configuration management expertise
- Familiarity with basic Linux commands

- A keen understanding of networking concepts
- Cloud computing
- Continuous integration and continuous delivery (CI/CD)
- Coding skills in at least one common scripting language, such as Python or Ruby
- Ability to use a text editor, such as Vim or Emacs
- Familiarity with basic Linux commands
- Ability to use a terminal emulator, such as PuTTY or iTerm2

Workplace skills

It's also crucial that you have strong workplace skills. The following skills can help you be more successful in your DevSecOps career and help you positively impact your organization.

- Strong communication and interpersonal skills
- Ability to manage and prioritize tasks
- Knowledge of top-level cybersecurity subjects and issues
- Ability to research threats and draw up logical conclusions through well-thought-out, unbiased processes
- Ability to troubleshoot and solve problems
- Ability to learn new technologies quickly
- Ability to bring together data from diverse sources and articulate it into simple and concise information

What is the future of DevSecOps?

With the ever-growing need for speed and agility, organizations are turning to DevSecOps to help deliver software with greater security and get it to the market faster. By automating security controls, integrating them into the software development process, and taking a more strategic approach to security, companies can mitigate the increasing risk posed by cyber threats.

Career paths

1. DevSecOps Engineer:

- DevSecOps Engineers are responsible for designing, implementing, and maintaining secure development practices within the organization.

- They work closely with development, security, and operations teams to integrate security tools, automate security processes, and monitor security vulnerabilities throughout the development lifecycle.

- Skills required: Strong knowledge of security principles, proficiency in scripting and automation tools, expertise in cloud security, and experience with DevOps practices.

2. Security Automation Specialist:

- Security Automation Specialists focus on automating security tasks, processes, and controls to improve efficiency and reduce manual errors.

- They develop scripts, tools, and integrations to automate security testing, compliance checks, and incident response workflows.

- Skills required: Proficiency in scripting languages (e.g., Python, Ruby), experience with automation tools (e.g., Ansible, Puppet), knowledge of security controls and compliance standards.

3. Security Analyst (DevSecOps):

- Security Analysts in DevSecOps roles analyze security threats, vulnerabilities, and incidents within the development pipeline.

- They conduct security testing, vulnerability assessments, and code reviews to identify and remediate security issues proactively.

- Skills required: Knowledge of penetration testing tools, security testing methodologies, understanding of secure coding practices, and experience with threat modeling.

4. Security Architect (DevSecOps):

- Security Architects design and implement secure architecture for applications, infrastructure, and cloud environments within the DevOps framework.

- They establish security controls, define security policies, and provide guidance on security best practices to ensure a robust security posture.

- Skills required: Expertise in security architecture design, experience with cloud security solutions, understanding of security frameworks (e.g., OWASP), and knowledge of compliance requirements.

5. DevSecOps Manager/Director:

- DevSecOps Managers or Directors oversee the implementation of security practices across the organization's development teams.

- They define security strategies, manage security initiatives, and ensure compliance with security policies and regulations.

- Skills required: Leadership and project management skills, strategic planning abilities, understanding of security governance, risk management, and compliance.

6. Compliance Analyst (DevSecOps):

- Compliance Analysts in DevSecOps roles focus on ensuring that security practices align with regulatory requirements and industry standards.

- They conduct audits, assessments, and reviews to verify compliance with security regulations and provide recommendations for improvement.

- Skills required: Knowledge of security compliance frameworks (e.g., GDPR, HIPAA), experience with compliance audits, understanding of data protection laws.

Conclusion

In conclusion, DevSecOps addresses the need for integrating security practices into software development, with a focus on collaboration, automation, and continuous testing. By following the core principles of DevSecOps, implementing the DevSecOps lifecycle, leveraging technical tools, and reaping the benefits of improved security, organizations can enhance their overall security posture and create rewarding career opportunities in the field of DevSecOps.

DevSecOps stands at the intersection of development, security, and operations, offering a holistic approach to building and maintaining secure software systems. By following the DevSecOps lifecycle, leveraging automation tools, and embracing a culture of shared responsibility for security, organizations can effectively mitigate risks, identify vulnerabilities early, and respond to security threats proactively. The benefits of DevSecOps extend beyond improved security to include enhanced collaboration, agility, and overall efficiency in the software development process. As DevSecOps continues to gain prominence in the industry, it presents both challenges and opportunities for professionals to make a significant impact in creating a more secure and resilient digital ecosystem.

Reference

[1]. https://doi.org/10.58012/fywc-yq50

[2]. https://www.veritis.com/solutions/devops/

[3]. https://expertinsights.com/insights/the-top-10-devsecops-tools-for-application-security/

[4]. https://aws.amazon.com/what-is/devsecops/

[5]. https://www.coursera.arg/articles/devsecops/