INSTITUTE OF TECHNOLOGY SCHOOL OF COMPUTING DEPARTMENTOF SOFTWARE ENGINEERING

COURSE TITLE: SOFTWARE ENGINEERING TOOLS AND PRACTICE

COURSE CODE: SEng3051

STUDENT NAME **ID**

1.Haylemariam semre ----------1306338

SUBMMITED TO: Esmail M.

Table of contents

Topics Pages

Introduction

DevSecOps is short for development, security and operations. It is a software development model in which these three teams work in close collaboration and in a synchronized fashion. One may say that a DevSecOps team is an agile, cross-functional DevOps team that embeds security practices into their own processes to deliver secure software products and digital services. In other words, DevSecOps is DevOps done securely.

The intent of DevSecOps is to make everyone accountable for security while still operating at the same speed and scale as DevOps development CI/CD pipelines. Adding application security to DevOps is a major challenge because security practices are becoming a "bottleneck" for software development assembly line. However, as cyber threats continue to grow, secure software development processes have never been more important.
The solution is to reimagine and rebuild security practices integration into DevOps.

Security is meant to be an integral part of DevOps because of security practices such as secure design, code reviews, automated testing, penetration testing. In real life, these detached DevOps security practices cannot provide security, although they could help. DevOps teams should adopt a security mindset and apply security principles to their day-to-day agile activities.

Today software is released in short sprints and it becomes essential to perform application security testing during development cycles without slowing down the engineering processes. Ideally, security activities should be started as early as possible during the software lifecycle – we call it a security “shift-left”. DevSecOps model brings together DevOps and continuous security testing.

Q1.What are Software engineering problems which was cause for initiation of DevSecOps.

before the advent of DevSecOps, the software development world had its fair share of challenges,and these are some of the problems which was the cause for initiation of devsecops:-

1.**Siloed Development**

The Challenge:

- Development, security, and operations teams operated in isolated silos, leading to communication barriers and disjointed workflows.

The Consequence:

- This lack of collaboration often resulted in security practices being an afterthought, not integrated seamlessly into the development process.

2. Reactive Security Measures:

The Challenge:

- Security evaluations typically occurred post-development, identifying vulnerabilities late in the process.

The Consequence:

- Fixing security flaws reactively led to increased costs, delays, and potentially higher risks of security breaches.

3. Slow Feedback and Iteration:

The Challenge:

- Lengthy feedback loops between security assessments and development hindered the timely resolution of security issues.

The Consequence:

- Iterations were slow, impacting the agility and efficiency of the development process.

4. Compliance vs. Security:

The Challenge:

- Emphasis on compliance rather than security led to a checkbox mentality, where meeting regulatory requirements took precedence over addressing real security threats.

The Consequence:

- Organizations often focused on passing audits without a robust security posture, leaving them vulnerable to attacks.

5. Manual Security Checks:

The Challenge:

- Manual security assessments were time-consuming and error-prone, making it challenging to keep up with the fast-paced development cycles.

The Consequence:

- Critical security vulnerabilities could slip through the cracks due to manual oversight.

In response to these challenges, ***DevSecOps*** emerged as a paradigm shift in software development, aiming to address security concerns early and continuously throughout the development lifecycle. By breaking down silos, automating security processes, and fostering

collaboration, DevSecOps transformed the way security is integrated into software development.

Q2. What is DevSecOps?

DevSecOps process is a method for handling IT security with the mindset that "everyone is accountable for security." It combines injecting security into a company's DevOps pipeline. The aim is to involve security in all software development life cycle (SDLC) stages. DevSecOps framework indicates you shouldn't save security for the last stage of the SDLC, contrary to its predecessor development methods.

If your business already uses DevOps, consider upgrading it to DevSecOps integration. DevSecOps phases are primarily built on the DevOps services, which will guide your case for switching. By doing this, you'll be able to assemble talented specialists from several technical disciplines to improve your security procedures as they are now.

DevSecOps is a way of thinking or a culture that IT operations and developers' teams follow when creating and deploying software applications. Agile application development incorporates security audits and penetration testing that are both active and automated.

To implement the DevSecOps process, you need to:

- Reducing vulnerabilities in software programming incorporates the concept of security from the beginning of the SDLC.
- Please make sure everyone, including IT operations teams and developers, shares responsibility for adhering to security procedures in their tasks.
- Ensure DevOps workflow begins with the involvement of security controls, processes, and tools. This will allow for automatic security checks throughout the software delivery process.

DevOps managed services have always been about integrating security into the development and release process, quality assurance (QA), database management, and everyone else. DevSecOps process, on the other hand, is an extension of that process where security is always the crucial component of the procedure.

How to Adopt DevSecOps with Your Team?

In software development, DevOps best practices have sparked a revolution. It combines software development, deployment, and management into one process. Operations and development teams would merge into a single team; if not, the teams collaborate closely. Faster updates and improved cycle control for software releases are the advantages.

Likewise, there has been a growing understanding that security must be an essential component of the development process. Writing code takes longer and doesn't work well

before figuring out how to make it secure. The phrase “DevSecOps” was created due to the convergence of these trends.

The core concept of the DevSecOps process is that everyone is responsible for security. Management must take into consideration when defining requirements and developing schedules. Developers must incorporate it into every facet of code and specifications. QA professionals must test security in addition to functionality. Finally, operations teams must monitor software behavior and respond quickly to problems.

Each party must adopt a new way of thinking to implement DevSecOps. They must establish a strong line of communication because they each have specific tasks. No issues ought to be overlooked due to a lack of communication. Security teams have frequently been separated from other groups during the development cycle. With the DevSecOps model, they are included in each stage of the phases of devsecops process and available to offer inputs.

Q3. Briefly explain DevSecOps lifecycle?

Lifecycles of DevSecOps:-

1.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.

2.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.

3.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.

4.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.

5.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.

6.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.

7.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.

8.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.

Q4. How does DevSecOps works?

DevSecOps will integrate automation across your software release pipeline to reduce downtime, security attacks, and fix code issues faster.
Businesses who want to employ security into their DevOps framework need to utilize optimal DevSecOps processes and tools.

Here's how your DevOps and DevSecOps framework will look like:

- A developer writes code in a version control management system.
- Any changes made are applied throughout the version control management system.
- Other developer retrieves this code from the version control system to analyze the static code quality and look for any potential security bugs.
- Utilizing infrastructure-as-code DevSecOps tools, developers create an environment such as Chef. In this infrastructure, you can easily deploy applications and perform environment-specific configurations.
- Perform test automation suite on the newly deployed application consisting of security tests, API, UI, integrations, and back-end.
- If the application runs seamlessly in testing, it is sent forward to the production environment.
- The production environment is kept under continuous monitoring to detect any active security risks, bugs, or attacks.
- DevSecOps creates a test-driven development infrastructure that executes continuous integration and automated testing to create quality code, enhanced security, and compliance

Q5. Exline well known DevSecOps tools.

1. CodeAI

CodeAI can automatically find and fix security vulnerabilities in your source code. To achieve this, CodeAI uses deep learning technology to help developers find issues and solutions to each security problem. QbitLogic—the vendor behind CodeAI—trained the solution using millions of actual bug-fix samples.

2. Parasoft Tool Suite

Parasoft provides a suite of tools that automate a wide range of development security testing aspects, including:

- Parasoft C/C++test – Can identify defects early on in the development cycle.
- Parasoft Insure++ – Can find erratic programming and memory-access errors.
- Parasoft Jtest – Designed especially for Java software development testing
- Parasoft dotTEST – Complements Visual Studio tools with advanced coverage and deep static analysis.

3. Red Hat Ansible Automation

Ansible is an IT automation engine offered under an open-source license. You can use Ansible to significantly reduce the scope of repetitive, manual work. This level of automation can help you improve the consistency, reliability, and scalability of your IT environment.

Ansible can help you automate the following types of tasks:

Provisioning – Ansible can set up servers for your infrastructure.
Configuration management – Ansible lets you automate configuration changes for your applications, device, or operating system. It can start and stop services, implement security policies, update or install applications, and more.

Application deployment – Ansible can improve DevOps pipelines by automating the deployment of applications to production systems.

4. StackStorm

StackStorm is a platform for runbook automation. It is event-driven and supports infrastructure as code (IoC). StackStorm uses “if-then” rules to simplify your workflows. Once a trigger event occurs, StackStorm checks the rules, runs relevant instructions, executes the appropriate commands, and provides the results.

StackStorm lets you compartmentalize small tasks, which you can then orchestrate into larger tasks. The tool has a variety of use cases for site reliability engineering (SRE) teams, such as automated remediation and security responses.

Container Security Tools

Container security technology can help ensure containers, container images, and related components are securely configured and free of vulnerabilities. Here are several container security tools:

5. Calico Open Source

Project Calico is an open-source project with an active development and user community. Calico Open Source was born out of this project and has grown to be the most widely adopted solution for container networking and security, powering 1.5M+ nodes daily across 166 countries.

Calico Open Source is a networking and security solution for containers, virtual machines, and native host-based workloads. Calico supports a broad range of platforms including Kubernetes, OpenShift, Docker EE, OpenStack, and bare metal services.

6. Clair

Clair ingests information from multiple vulnerability data sources, including CVE databases like Ubuntu CVE Tracker, Red Hat Security Data, and Debian Security Bug Tracker. Using this data, it performs comprehensive static analysis of container vulnerabilities.

7. Notary

Most publishers, including container repositories, use TLS to secure their communications with web servers. However, TLS does not help protect against compromised servers. If the server is compromised, TLS cannot prevent it from substituting legitimate content with malicious content. Notary can help prevent these issues from occurring.

The Notary project is based on The Update Framework (TUF), a secure design that helps solve software distribution and update problems. The tool lets publishers sign their content offline by using keys that are kept highly secure.

Cloud Testing Tools

Cloud testing tools provide test environments especially for the cloud, including all requisite software-hardware configurations. Most cloud-based testing platforms offer integrations with DevSecOps tools and CI/CD workflows.

8. AppScan on Cloud

AppScan on Cloud provides a suite of security testing tools, such as dynamic, interactive, and static testing for mobile, open-source, and web applications. The tool can detect pervasive security vulnerabilities and facilitate remediation.

9. AWS Security Service

Amazon Web Services (AWS) offers a variety of security services. For example, their data protection offerings include encryption, key management, and continuous threat detection for your workloads and accounts. Additionally, AWS offers an identity management service that enables you to manage your identities, permissions, and resources at scale.

10. ThreatModeler

ThreatModeler aims to help enterprises effectively manage security risks. The solution offers a Cloud Edition that automatically builds threat models for your cloud infrastructures. It can manage potential threats for various clouds, including AWS and Microsoft Azure.

Application Security Testing Tools

Several types of application security (AppSec) tools can help DevSecOps teams verify that applications are secure before releasing them to production:

Static Application Security Testing (SAST) tools can analyze your source code or any compiled versions of your code and identify security flaws during early development phases.

Dynamic Application Security Testing Tools (DAST) tools can identify security flaws by performing realistic tests on applications running in testing or production environments.

Test automation software enables DevSecOps teams to define software testing tasks that reduce the amount of manual labor.

11. Veracode

Veracode Static Analysis is a SAST solution that can analyze software libraries in all major frameworks and languages without requiring access to the source code—making it possible to analyze proprietary code alongside components from external vendors.

Veracode provides an API, which lets you integrate static analysis with existing CI/CD tools. The solution also supports adding static analysis to IDEs, build systems, and task management systems. It provides a Pipeline Scan feature that lets you scan new code commits, identify and prioritize security flaws, and compare them to previous scans, to quickly identify which version introduced a new security issue.

12. Checkmarx CxSAST

CxSAST is a static analysis tool offered as part of the Checkmarx Software Exposure Platform. CxSAST aims to identify security vulnerabilities in custom code as well as open-source components. The tool supports more than 25 scripting and coding languages.

Here are notable features of CxSAST:

Helps organizations ensure coverage of industry compliance regulations and security standards.

Fixes vulnerabilities in the code.

Allows developers with different skill sets to utilize security features easily—there are no configuration changes, no complex wizard commands, and no learning curve when switching languages.

Provides an incremental scan capability that allows scanning only modified or new code.

13. SonarQube

The SonarQube platform applies continuous inspection to manage code quality. It is an open-source tool that supports more than 25 programming languages and integrates with existing workflows. It displays the health of your application and highlights detected new issues. DevSecOps teams can use this tool to quickly detect and remediate code errors to ensure both security and quality.

14. Fortify WebInspect

Fortify WebInspect is a dynamic application security testing (DAST) tool that can help you find and prioritize exploitable vulnerabilities in your web applications.

Key features include:

Functional Application Security Testing (FAST) – Able to run functional tests like IAST, but without being limited to a specific subset of functionality.

Black box testing insights – Scans a running application like a hacker would. This type of testing can identify client-side frameworks used, version numbers, and other vulnerabilities that attackers could easily detect and exploit.

Compliance management – Provides built-in policies and reports for many compliance standards, including PCI DSS, HIPAA, NIST 800-53, ISO 27000, and OWASP Top Ten.

API support – Can scan both SOAP and REST APIs, identifying API functionality using Swagger, OpenAPI, or Postman, to discover API security vulnerabilities.

15. New Relic

New Relic provides an observability platform that lets you bring in data from various sources. It helps you use this data to gain a comprehensive understanding of your software and learn how to improve it.

Here are several key advantages of New Relic:

Centralized data – New Relic can help you instrument all information and import data from the entire technology stack, using agents, APIs, and integrations.

Data analysis – The platform lets you analyze all of your data from a single UI, leveraging New Relic's query language to find the root causes of issues.

Threat detection – New Relic's machine learning solution can proactively detect and explain anomalies before they become critical.

16. ELK with Kibana

The ELK Stack includes three open-source tools—Elasticsearch, Logstash, and Kibana (ELK). The stack can help you identify problems with servers or applications. Logstash can centralize your logging offers, Elasticsearch lets you search this data, and Kibana offers data visualization.

The three tools in the ELK Stack complement each other. Kibana can help you visualize Elasticsearch documents. You can use Kibana to create dashboards that offer interactive diagrams, view geospatial data, and employ graphs to visualize complex queries. In addition to visualization, Kibana also lets you search and interact with data kept in your Elasticsearch directories.

Q6. What are the benefits of DevSecOps?

Benefits of DevSecOps

The two main benefits of DevSecOps are speed and security. Therefore, development teams deliver better, more-secure code faster and cheaper.
"The purpose and intent of DevSecOps is to build on the mindset that everyone is responsible for security with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required," describes Shannon Lietz, co-author of the "DevSecOps Manifesto."

1.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.

2.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.

3.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.

4.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 production1.

5.A repeatable and adaptive process

As organizations mature, their security postures mature. DevSecOps lends itself to repeatable and adaptive processes. DevSecOps ensures that security is applied consistently across the environment, as the environment changes and adapts to new requirements. A mature implementation of DevSecOps will have a solid automation, configuration management, orchestration, containers, immutable infrastructure and even serverless compute environments.

Q7. 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 environment.
• 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 moresuccessful 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.
More companies understand and seek the benefits of integrating security into their DevOps processes. The niche has an impressive predicted growth rate of 35 percent from 2021 to 2031, according to the US Bureau of Labor Statistics (BLS).

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.

References

http:/www.synopsy.com/glossary/what-is-DevscOps.html
http:/www.practical-devscops.com/devscops-life-cycle
https://www.datadoghq.com/knowledge-center/devsecops/
https://www.javapoint.com/what-is-devsecops
https://www.veritis.com/blog/what-are-the-phases-of-devsecops/