DevSecOps

One goal of DevOps is to deliver rapid, high-quality software updates. And for software to be high quality, it must meet stringent cybersecurity requirements. That's where DevOps security, or DevSecOps, comes in. In this lesson, you'll explore DevSecOps, its security checks, and its reliance on automation.

What is DevSecOps?

DevSecOps is an extension of DevOps that automates security checks throughout the software development lifecycle (SDLC) to prevent vulnerabilities in the final product. A **vulnerability** is a potential weakness, like missing data encryption, that someone can exploit in a system.

Traditionally, developers have written most production code without security in mind, and only at the end of the SDLC would a security team test the software. This approach works when updates come just a few times a year, but DevOps teams produce updates every few weeks or sooner.

With DevSecOps, developers incorporate security into each step of the SDLC. Teams consider and plan for potential security threats early on, and they test, scan, audit, and review code throughout development.

Components

Key components of DevSecOps include shared responsibility, speed and quality, security checks, and automation.

Shared responsibility

In DevSecOps, everyone—the development, operations, and security teams—shares responsibility for security.

- Team members should understand basic application security and mitigation strategies.
- Team members should follow updates to the Open Web Application Security Project (OWASP) Top 10, an industry-standard list of critical security risks for web applications.
- Developers should agree on and follow secure coding practices.

Speed and quality

Security problems in the application require time and money to fix, especially those found late in the SDLC, and can significantly delay a release.

With DevSecOps, teams account for security from the planning stage onward, and they identify and address security issues early and quickly. That way, they can continue delivering small, continuous, high-quality updates.

Security checks

With DevSecOps, software undergoes numerous security checks throughout the SDLC. Let's discuss some standard checks.

- **Threat modeling** is a process in which teams identify and categorize security threats to account for in software development and support. Threat modeling typically occurs during the design or planning stage of development before developers write the code.
- **Vulnerability scans** identify vulnerabilities in the application and from libraries (collections of reusable code) on which the application depends. Teams can automate patching to address vulnerabilities as soon as possible. Two common vulnerability scans are static application security testing and dynamic application security testing.
- **Static application security testing (SAST)** tools scan for vulnerabilities inside the code and its libraries. "Static" means that the application is not executing; it's at rest.
- **Dynamic application security testing (DAST)** tools detect vulnerabilities observable outside the code while the application runs. To do so, these tools simulate real hacking techniques like SQL injection to discover weak spots that cybercriminals can exploit.
- **Secrets detection** scans search for secrets that developers accidentally leave in code or configuration files. Secrets are sensitive credentials like passwords and encryption keys, and organizations must protect them from leaks. If secrets find their way into the application's code base, cybercriminals may find them.
- **Unit tests** are tests that evaluate a single component, or unit, of an application to verify that the component performs correctly. These tests run when developers submit newly written code for integration into the software's main code base. In DevSecOps, developers design additional unit tests, called **security unit tests,** that check for security issues.
- **Security monitoring** tools monitor live applications for security issues like cyberattacks and send immediate alerts when such activity occurs. That way, personnel can respond quickly to minimize harm and patch the application if needed.

Automation

Automation is essential for DevOps, and DevSecOps is no different.

Tools for continuous integration and continuous delivery (CI/CD) can automate security checks through nearly every stage of the SDLC, freeing everyone to focus on other tasks.

- Automation tools verify that the code passes security unit tests and that software dependencies are on their latest patches.
- SAST tools detect vulnerabilities in new code before developers incorporate it into their code base.
- DAST tools evaluate updates in a pre-production environment.
- Tools can automate configuration of systems and services, ensuring security compliance and reducing human error.