CS-305 Software Security

Huan Ai

7/21/2025

What is your role in solving security concerns as a developer? What might solving security concerns as a developer involve?

I used to see security as someone else’s problem—something the "security team" handled later. This article was a wake-up call. Every line of code I write could have been a potential vulnerability. If I skip input validation, I am leaving the door open for SQL injection. If I hardcode credentials, I’m risking a breach. One slip in code could leak customer data, melt down a business, or even endanger people. As developers, we own the first line of defense.

Solving security concerns as a developer involves writing secure code from day 1. We need to use parameterized SQL queries every time we touch a database. We need to validate all user inputs when we work on backend APIs, which helps to prevent XSS attacks. Solving security concerns as a developer also involves collaborating proactively with other team members. For example, running threat modeling sessions with security teams for new features.

Where does security fall within the software stack and development life cycle?

Security is everywhere within the software stack and development life cycle. For software stack, security starts from infrastructure (For example, securing cloud configs), to APIs (validating tokens/user inputs), to UI (sanitizing outputs). For SDLC, at planning phase, we define security requirements (e.g., "Payment data must be encrypted"). At coding phase, we write secure functions and use tools to scan for flaws. At testing phase, we run DAST/SAST scans and manual pen tests. At deployment phase, we scan containers for misconfigurations before release. At operation phase, we continue to monitor logs for suspicious activity after launch.

How might you add security measures to transform a DevOps pipeline into a DevSecOps pipeline?

Based on the article’s "continuous security pipeline" idea, I would add security into existing DevOps tools. I would add SAST/DAST scans to Jenkins/GitLab CI stages. Builds will fail if critical bugs are found. I would automate infrastructure checks (e.g., Terraform scans for open S3 buckets). In addition, I would shift security left culturally by making developers own security testing (e.g., train them to run OWASP ZAP scans).

The article suggests creating and following a plan to secure the entire DevOps life cycle. What is included in the suggested plan? Would you recommend following the plan?

The suggested plan is a comprehensive, phase-based security plan covering the entire DevOps lifecycle. It involves people & culture (security training, collaboration awareness), process integration (CI/CD pipeline that includes SAST, DAST, and IAST), and technology & tools, including IDE plugins (e.g., SonarQube for SAST), dependency scanners (e.g., Dependabot for vulnerable libraries), and Fuzz testing tools (e.g., OWASP ZAP). It also involves real-time monitoring (log analysis, intrusion detection) after launch. I would totally recommend following the plan.

Reference:

Jeganathan, S. (n.d.). The Issa Journal November 2019: DevSecOps: A systemic approach for secure software development. Content Delivery Platform. https://mydigitalpublication.com/article/DevSecOps:+A+Systemic+Approach+for+Secure+Software+Development/3524379/632044/article.html