CS-305-T1168 Software Security

3-2 Journal Reflection

Eric Wallace

Sep. 18, 2022

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

As a software engineer/developer my role in solving security concerns is to work with

the security team to determine the security requirements, run high-level RRA, plan and implement the standards set forth by the DevSecOps Lifecycle.

Adopting the Agile software development lifecycle framework, take the “Shift-Left/Test Everywhere” mindset in which developers test early and test often. Practice secure coding by using linters, utilizing SAST tools which are used to analyze code for coding and design conditions and use vulnerability assessment tools such as OWASP. Implementing 2FA, eliminate the use of outdated frameworks or programming languages, and develop unit tests that focus on security.

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

Traditionally security has always favored the right side or deploy side of the development pipeline. Security has often been addressed after the fact and conducted during specific testing phases of the SDLC.

Recently this has changed with the SDLC changing to Agile and Lean which push it to the left or to the forefront as much as possible so security is address as early as possible.

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

Weaponize unit tests: Instead of using positive use cases look at negative use cases to

find vulnerabilities and areas of code that could be subject to attacks.

Verify Security of Third-Party Components: Use vulnerable assessment tools to find

vulnerabilities in third-party components, libraries, application dependencies and other code that is part of an application.

Audit State of System and Settings: Verify the state of server’s or container’s patches

and configurations, encryption stating and headers. No application should be released with missing patches or poor encryption.

Adding Dynamic Application Security Testing (DAST): Launch coordinated attacks and

finding bugs using data injection against an application to find security issues.

Add Static Application Security Testing: Using SAST tools to search for one

vulnerability can potentially eliminate an entire set of bugs. Implementing this early in the pipeline can lead to faster and fewer fixes.

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

1. Plan: Security requirements and design are done for software development and security the entire DevOps life cycle
2. Code: Apply secure coding practices, integrate SAST tools and enforce industry-followed secure-code practices.
3. Build: Software modules are checked into source code repository and made available to package and bundle for deployments.
4. Test: Unit level testing is completed
5. Release and Deploy: Processes, tools and technologies where software modules are released to the lower and live environments on a defined schedule
6. Operate: Monitoring, analysis, and protection for live-run environments.

I believe this is a solid plan to secure a DevOps life cycle, it seems to adopt an Agile like approach to implementing such a plan, so I would recommend following it.