**3-2 Journal: Reflection**

Winnie Kwong

Southern New Hampshire University

CS-305 Software Security

Professor Foultz

September 16th, 2023

**Roles in Solving Security and Solving Security Concerns**

Development and security cannot exist without the other. Including security increases trust, reputation, and sales and reduces costs. As a developer, the roles to solve security concerns consists of focusing on the safety and security of the software. A way to solve security concerns is for developers to pass the code to DevOps/DevSecOps staff and security teams for deployment and evaluation to ensure any issues are identified as soon as possible.

**Where Security Falls in Software Stack and Development Life Cycles**

According to Jeganathan, “in traditional software development models, security is often viewed as an afterthought wherein security testing is mostly conducted during specific testing phases of the software development life cycle, which are usually planned far ahead (right) in the schedule...” (Jeganathan, 2019, p20). However, security is being pushed to begin in earlier stages with the cultural shift, but it can only occur when people, processes, and technological elements shift as well. Realistically, security applies to every phase of the software development cycle and should be at the forefront of a developer’s mind to ensure software requirements are met.

**Adding Security Measures from DevOps Pipelines into DevSecOps Pipelines**

Ways to add security measures to transform a DevOps pipeline into a DevSecOps pipeline involve security teams, automating security outcomes, and learning more about best practices around libraries and modules that can integrate with the monitoring stack. Having security professionals involved can raise possible security concerns and identify potential dangers. In these situations, security professionals can provide suggestions to mitigate the case that could also reduce the project's cost. By applying continuous integration/continuous delivery (CI/CD) pipelines, it will be more maintainable to handle input validation, integration testing, and behavior monitoring for efficient and high-quality performances. Also, using CI/CS pipelines also reduces the risk of human errors from manual processes that prevent developers from focusing on other tasks. Lastly, technology constantly changes, and developers should know what is happening. According to Vanbuskirk, "Configuration changes, new dependencies, and code deployments can all introduce new risks that need to be mitigated." (Vanbuskirk, 2023, para 26). If a developer is unaware of outdated security policies, exploits may occur and invite potential attackers to gain control of the system. By including security professionals, automating security outcomes, and utilizing best practices, security measures from DevOps Pipelines will influence DevSecOps Pipelines.

**Suggested Plan to Secure the Entire DevOps Life Cycle**

The suggestion for creating and following a plan to secure the entire DevOps life cycle is to start with a high-level rapid risk assessment, plan and secure the DevOps lifecycle tools, use privileged account security tools, and define infrastructure protection controls, and enforce segregation of duties. Within a cybersecurity risk assessment, it helps to determine its fundamental objective and to identify what tools are being used. By pursuing a risk assessment, developers can make rational decisions and determine what areas need more attention to reduce the overall risks. Including tools in the DevOps lifecycle adds extra layers of security to help boost protection. Adding additional layers will make users feel safer when providing sensitive data. Privileged access management (PAM) is a solution to identity security to help prevent unauthorized users from accessing confidential information. By managing PAM, it helps to prevent data breaches and accounts from being exploited. Defining infrastructure protection controls and enforcing segregation of duties (SoD) can be accomplished by breaking down tasks into multiple tasks so that one person manages all control. When jobs are distributed, it helps to minimize the occurrence of errors or fraud as well as reduce people from concealing problems within their duties. I recommend using this plan because it helps establish transparent and secure software.

**References**

Jeganathan, S. (2019). DevSecOps: A Systemic Approach for Secure Software Development. *ISSA Journal*, 17(11), 20–27.

Vanbuskirk, M. (2023, June 8). *5 ways to implement DevSecOps right now* [Review of *5 ways to implement DevSecOps right now*]. Pluralsight. <https://www.pluralsight.com/resources/blog/cloud/5-ways-to-implement-devsecops-right-now>

‌