Daniel Olson

CS-405 Secure Coding

Southern New Hampshire University

21 April 2023

8-2 Journal: Portfolio Reflection

Adopting a secure coding standard from the start of the development process is required for developing safe software. This standard specifies how tools, services, and systems should be used across the software development life cycle to ensure that security best practices are implemented. To identify common vulnerabilities and mitigation techniques, it is critical to create communication channels between the security and development teams. Companies may emphasize security throughout the development process by creating a secure coding standard, avoiding leaving security measures till the conclusion of the project.

Developing safe software necessitates a thorough strategy that considers risk assessment as well as cost-benefit analysis of mitigation. Because software vulnerabilities are continually being found, it is critical to assess the dangers of leaving them open. Prioritizing vulnerabilities based on the potential damage they may do and the time it would take to repair them is critical. The expense of putting security measures in place should also be evaluated. To enable rapid product development while maintaining security, a methodology that incorporates security at every phase of the SDLC is required. DevSecOps is a strategy that incorporates security into the existing DevOps architecture. This method may be used to automate duplicate and time-consuming activities while also facilitating continuous security integration. It is also critical to teach developers on proper practices for security and to leverage IDE security features such as static analysis tools to detect vulnerabilities early on.

The zero-trust security strategy is an effective way to secure software development that prioritizes users above networks. It assumes that without adequate identification and authorisation, no person or system can be trusted. SSO and biometric access may be used to improve the user experience while ensuring security. A good beginning point for preventing vulnerabilities is to adopt a methodology that integrates security into every SDLC phase. Prioritize dynamic testing and automated security checks, as well as frequent security training sessions for the development team to remain up to speed on the newest vulnerabilities and security best practices. Companies may guarantee that their software is resistant to possible threats by implementing a thorough approach to safe software development.