8-2 Journal: Portfolio Reflection

Jillian R. Ortiz

SNHU

CS405 Secure Coding

Dr. Tam

10/27/2024

When practicing secure coding, one should not leave security practices until the end of the process. Secure coding practices should be applied throughout the entire software development life cycle (SDLC). From the design stage, where requirements are addressed, to the production stage, implementing unit tests or other test automation throughout each iterative portion of the program development will allow developers to address issues as they arise. Addressing vulnerabilities as they arise will be more cost-effective and manageable to remediate.

Risk assessment and cost/benefit analysis help to frame the risks, costs, and benefits of adopting a particular practice or standard. Most coding standards utilize a threat matrix, with threat level, severity, remediation costs, and overall importance shown, to help determine the priority. Many studies have also been done to compare the cost and benefit

associated with secure coding, and typically the result is the benefit of adopting early security, good coding standards, and overall limiting risk, will outweigh the cost.

Zero Trust is a security concept that operates under the principle of "never trust, always verify." Instead of assuming everything inside an organization's network is safe, Zero Trust ensures that every user, device, and application must be authenticated and continuously verified before gaining access to resources. This approach towards the interaction and connection between devices and users is highly preemptive, not presuming anything, whether within or outside a network or system.

In conclusion, the development of security policies and recommendations helps bridge the gaps between each of these concepts, and more, by attempting to adopt all aspects of secure coding in some fashion. The objective was to transition DevOps to DevSecOps by implementing security standards and practices throughout the development cycle.