Ericka Resendez

CS-405 Secure Coding

Southern New Hampshire University

April 25, 2025

Portfolio Reflection

April 25, 2025 Portfolio Reflection A critical lesson from this course is the necessity of integrating security into every phase of the software development lifecycle (SDLC) rather than addressing it as a late-stage concern. Security integration should be recurring and not simply an afterthought. Secure coding involves writing software that is protected from the outset to prevent vulnerabilities, thereby avoiding costly remediation late in the development cycle. (Bellairs, 2018). The Open Web Application Security Project (OWASP, n.d.) provides critical secure coding practices that help developers avoid common vulnerabilities like SQL injection and cross-site scripting during initial programming phases. The National Institute of Standards and Technology's SP 800-30 (2012) framework offers vital guidance for conducting comprehensive risk assessments throughout development. This approach aligns with Foster's (2018) argument that adopting security standards early creates more robust systems while reducing long-term costs. By identifying potential threats during design rather than deployment, organizations can implement cost-effective mitigation strategies when changes are least expensive to implement. Zero Trust Architecture, as defined in NIST SP 800-207 (2020), is a paradigm shift from traditional perimeter-based security models. This "never trust, always verify" approach requires continuous authentication for all users and devices, significantly reducing potential attack surfaces (Cloudfare, n.d.). This strategy proves especially useful in modern environments where employees access systems from various locations and devices. Incorporating security considerations throughout development produces more stable and secure systems than bolting on protections after completion (Ryther, n.d.). This proactive approach, combining secure coding standards, continuous risk assessment, and Zero Trust principles, creates a comprehensive security posture that adapts to evolving threats while minimizing technical debt and remediation costs.

References

Bellairs, R. (2018). What is secure coding? Perforce Software <https://www.perforce.com/blog/sca/what-secure-coding>

Cloudflare. (n.d.). What is Zero Trust?  <https://www.cloudflare.com/learning/security/glossary/what-is-zero-trust/>

Foster, S. (2018). What are security standards? Secure coding standards overview. Perforce Software. <https://www.perforce.com/blog/qac/secure-coding-standards>

National Institute of Standards and Technology. (2012). NIST SP 800-30: Guide for conducting risk assessments. <https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final>

National Institute of Standards and Technology. (2020). NIST SP 800-207: Zero Trust architecture. <https://csrc.nist.gov/publications/detail/sp/800-207/final>

Open Web Application Security Project. (n.d.). Secure coding practices. <https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/>

Ryther, J. (n.d.). Don't leave security to the end. Medium. <https://medium.com/@jryther91/dont-leave-security-to-the-end-ca13b107d4>