**8-2 Journal: Portfolio Reflection**

Caleb Leavell

caleb.leavell@snhu.edu

Southern New Hampshire University

**Reflection on Secure Coding Practices and Policies**

Adopting a secure coding standard is a fundamental practice in software development, ensuring that security is integrated from the beginning rather than being an afterthought. Secure coding standards, such as those outlined by OWASP and CERT, provide guidelines to prevent common vulnerabilities like SQL injection, buffer overflows, and authentication flaws. By embedding security principles into the development lifecycle, organizations mitigate risks before they become critical issues. One of the most crucial aspects of secure coding is ensuring that security is not postponed until the final stages of development. When security is left to the end, vulnerabilities may go undetected until deployment, increasing remediation costs and potential exposure to threats. A proactive approach, incorporating security testing throughout the software development lifecycle (SDLC), significantly reduces risks.

Risk evaluation and mitigation strategies are essential in making informed security decisions. Organizations must assess potential threats, their likelihood, and the impact they could have on operations. This assessment allows developers and security teams to prioritize risks based on severity and implement appropriate countermeasures. A cost-benefit analysis of mitigation strategies is crucial, ensuring that resources are allocated efficiently to address the most pressing vulnerabilities. Implementing security measures should be a balance between effectiveness and feasibility, preventing overspending on low-impact threats while ensuring critical risks are addressed.

The zero-trust security model has emerged as a best practice in modern cybersecurity, emphasizing the principle that no entity—internal or external—should be automatically trusted. Unlike traditional perimeter-based security approaches, zero trust enforces strict identity verification, least privilege access, and continuous monitoring. This model significantly reduces the risk of insider threats and unauthorized access, making it a necessary approach in an era of increasing cyberattacks. Implementing zero trust requires robust authentication mechanisms, such as multi-factor authentication (MFA), network segmentation, and real-time monitoring of user behavior.

Developing and maintaining effective security policies is essential for sustaining a secure environment. Security policies provide a structured approach to handling data protection, access control, incident response, and regulatory compliance. A well-documented security policy ensures that all team members understand their responsibilities and follow best practices to maintain system integrity. Recommendations for improving security policies include regular audits, employee training, and incorporating automated security tools for continuous compliance checks. Additionally, policies should be dynamic, evolving alongside emerging threats and technological advancements.

In conclusion, secure coding practices, risk evaluation, zero trust implementation, and strong security policies are integral to developing and maintaining secure software. Organizations that prioritize these elements protect their systems from cyber threats and ensure long-term resilience. By embedding security into every stage of development and enforcing strict security protocols, businesses can mitigate risks effectively and safeguard their digital assets.