**8-2 Journal: Portfolio Reflection**

Evan Bush

Southern New Hampshire University

CS-405: Secure Coding

Professor Mike Prasad

October 22, 2023

**8-2 Journal: Portfolio Reflection**

Secure coding, like SEI CERT guidelines, helps avoid software flaws. It's about embedding security within the Software Development Life Cycle (SDLC), not adding it later. This isn't a one-off task but an ongoing effort, echoing the 'shift-left' idea—bringing security tasks forward in the SDLC (DORA, n.d.).

Risk assessment in secure coding is about spotting software flaws from insecure code. Such defects let malicious actors create exploits (Morrow, 2023). Basic web security and strict secure coding are essential to combat these risks. A whole team's grasp and use of basic web security measures are vital to shield apps from common threats (Cairns, n.d.). Applying these principles throughout the SDLC cuts risks and makes a safer environment.

Mitigation strategies are critical in the security review process. However, firms must do a proper cost-benefit analysis of potential strategies. This detailed process ensures that financial and reputational gains and losses are well-considered. By looking at the upfront and ongoing costs of mitigation, firms can make wise choices, upping the chances of future savings and benefits while cutting their overall risk profile. A balanced risk mitigation and cost-effectiveness approach through cost-benefit analysis will promote an economically secure coding ecosystem.

Zero Trust is about not trusting anything inside and outside the firm's boundaries without checking. In secure coding, zero-trust means ongoing checks and validation of active software and users interacting with the system through the SDLC. Security measures like user checks, data scrambling, and regular security tests from the project start showing the zero-trust idea in the development cycle. Focusing on sensitive data, tracking data flows, and applying proper security controls around data is crucial (Kueh, 2020). This ensures that every piece of code is checked and sticks to solid security rules, cutting the chance of software flaws in the software.

Setting up security rules is critical to solid security in software development. However, the first step is identifying and classifying assets and potential rules. A security rule that gives a solid framework for a firm needs to be well-explained to build and keep it across its software infrastructure (Dunham, 2021).

# References

Cairns, C. (n.d.). The basics of web application Security. martinfowler.com.

https://martinfowler.com/articles/web-security-basics.html

DORA. (n.d.). Shifting left on security.

<https://dora.dev/devops-capabilities/technical/shifting-left-on-security/>

Dunham, R. (2021). Information Security Policies: Why They Are Important To Your

Organization. Linford & Company LLP. https://linfordco.com/blog/information-security-policies/

Kueh, T. (2020). A practical guide to Zero-Trust security. Threatpost.

https://threatpost.com/practical-guide-zero-trust-security/151912/

Morrow, S. (2023). What is Secure Coding and Why is It Important? VPNOverview.com.

https://vpnoverview.com/internet-safety/business/what-is-secure-coding/