**Portfolio Reflection**

**Journal 8-2**

David Faulkner

Department of Computer Science, Southern New Hampshire University

CS-405: Secure Coding

Prof. Michael Alesso

October 27, 2024

**Portfolio Reflection**

Building on CS-305, this course has opened my eyes to quite a few new concepts, not only in C++ specifically, but also in software development in general.

**Adoption of a Secure Coding Standard**

One of those concepts is the codifying and adoption of a secure coding standard. It’s important to develop and adopt a *written* policy that can be maintained and referred to throughout the entire organization to ensure that all departments and team members are on the same page when it comes to general security principles and specific coding standards. This helps remove ambiguity and misunderstandings so that downtime is minimized.

**Don’t Leave Security to the End**

Another concept this course built upon is “shift left” security. “Shift left” refers to the practice of mitigating security issues as soon as possible in the software development life cycle since correcting vulnerabilities and breaches cost significantly more to correct later rather than sooner (*Start With Security: A Guide for Business*, 2024).

**Evaluation and Assessment of Risk and the Cost Benefit of Mitigation**

One skill that was particularly helpful that I learned in this course was the use of third-party tools such as the SEI CERT C++ Coding Standard in order to help evaluate and assess the level of risk associated with particular vulnerabilities and coding standards (*SEI CERT C++ Coding Standard - SEI CERT C++ Coding Standard - Confluence*, n.d.). This tool also helps determine the potential cost of specific vulnerabilities so that mitigation action plans can prioritize patching based on risk and cost so that the most critical vulnerabilities are patched first.

**Portfolio Reflection**

**Zero Trust**

Zero trust is another concept discussed in this course, which addresses the need to treat every user, network, application, and piece of data as unsafe, by default, regardless of where it originated. Remote work is here to stay, so gone are the days of implicitly trusting users, applications, devices, and data once they’ve gotten past the company firewall (Kueh, 2020).

**Implementation and Recommendations of Security Policies**

Implementation of security policies needs to happen early and often in the DevSecOps pipeline since the concept of shift left security applies to formal written policies, as well. Not only do formal policies get everyone on the same page, they communicate to the organization that security is a constant priority, and vigilance needs to be practiced continuously.

**References**

Kueh, T. (2020, January 17). A Practical Guide to Zero-Trust Security. *Threat Post*. <https://threatpost.com/practical-guide-zero-trust-security/151912/>

*SEI CERT C++ Coding Standard - SEI CERT C++ Coding Standard - Confluence.* (n.d.). <https://wiki.sei.cmu.edu/confluence/pages/viewpage.action?pageId=88046682>

*Start with Security: A Guide for Business*. (2024, February 14). Federal Trade Commission. <https://www.ftc.gov/business-guidance/resources/start-security-guide-business>