Jasmine Zeng

Professor Ahlam Alhweiti

CS 405

Module 8

**8-2 Journal: Portfolio Reflection**

ADOPTION OF A SECURE CODING STANDARD, AND NOT LEAVING SECURITY TO THE END

Not leaving security to the end is essential in secure coding. In each stage of the software development lifecycle (SDLC), security should be considered a critical factor before proceeding to the next stage. Programs should be designed, developed, tested, and deployed to prevent compromise of the system, its data, and its users. If security is an afterthought in the SDLC, it will reap many negative and even damaging repercussions to all parties involved. Consequences could result in ethical and economic damage and even harm to life or property. One secure coding standard is unit testing. Implementing dependencies to check on all factors or traits of a system, such as data validation, helps developers monitor the system for potential areas of harm.

EVALUATION AND ASSESSMENT OF RISK AND COST BENEFIT OF MITIGATION

Companies stand to benefit from the implementation of secure coding practices early on in development. When a strong foundation is built, the rest of the application follows suit. Bad practices can cause the system to be vulnerable to compromise. Data breaches can test the tide of an application and determine whether or not an application was built well from the start. It would cost more to implement security policies in later stages of the SDLC or even after the fact, since an application increases its risk and responsibility as it proceeds through the SDLC. This is especially important during application deployment, when it is live and responsible for managing and retaining sensitive information from users or the company.

ZERO TRUST

The zero trust security practice safeguards all users and data within a system by establishing barriers even at the lowest level. Assuming that “no one is safe” will ensure all parties and users are protected. This prevents data breaches from occurring more frequently since there are more layers to penetrate through. This also means that more layers of detection are possible when developers are analyzing for security.

IMPLEMENTATION AND RECOMMENDATIONS OF SECURITY POLICIES

Implementations and recommendations of security policies are included in the Green Pace security manual. My security policy is modeled after standard C/C++ coding standards, mainly for data validation and handling. Dependencies for each security standard will help detect any potential areas of improvement or vulnerability within the system. By abiding by the security policy manual, Green Pace will be able to construct applications with minimal struggle in security. It will not only build a robust system, it will promote trust in users and clients. Technical and non-technical factors are important when implementing and considering security protocols.