Portfolio Reflection

Tyler Hollingsworth

Computer Science Department, Southern New Hampshire University

CS-405: Secure Coding

Professor Aaron Demory

April 23, 2023

* **Adoption of a secure coding standard, and not leaving security to the end**

The adoption of secure coding standards is highly important for a company as it creates a universal set of rules and guidelines for developers to follow. By using secure coding standards, the company can prevent, detect, and eliminate security vulnerabilities in its code by setting the expectations that the company expects to be followed. These coding standards give descriptions, examples, and threat levels of possible vulnerabilities that could be introduced into the system. By creating and following these standards, the company is applying the principle of don’t leave security until the end. In modern agile development environments, it is crucial to incorporate security into the development life cycle. Using methods such as a DevSecOps framework, secure coding standards, and the correct use of tools for automation can ensure this.

* **Evaluation and assessment of risk and cost-benefit of mitigation**

Evaluation and assessment of risk is another important aspect of secure coding as some vulnerabilities will require a higher priority than others. Sometimes the cost-benefit of mitigation is well worth the extra effort to secure the vulnerability and other times the vulnerability is not severe enough to justify the increased time and money spent to patch the vulnerability as there isn’t a risk to the system. Market competitiveness is a huge priority in software development, like trying to beat the competition to release a product. It is important to focus on the security vulnerabilities that pose the largest risk to the system so that the release timeline is not pushed back. With that being said not every situation can be foreseen in advance so evaluating and assessing the risk along with the cost-benefit of mitigation needs to be an ongoing task as more vulnerabilities and exploits will be discovered over time.

* **Zero trust**

Zero trust was born from the modern complexities of remote access workers where the use of castle and moat strategies no longer work. Securing systems in the past was based on an on-premises location strategy such as trusting all users and accesses from within the network using methods like a firewall but nothing from outside the network. However modern companies have distributed their employees that work all over the world, so now it is more important to change from a mindset of trust but verify to one of never trust and always verify. Zero trust does this by securing every point of access on the system using methods such as single sign-on, multifactor sign-on, two-step verification, biometrics, and trusted devices. As the world continues to move into a more distributed environment and more employees become remote users of the system then zero trust will become more important than ever.

* **Implementation and recommendations of security policies**

Security policy implementation should be a total company buy-in from the leaders of the company down to the bottom and should be created to implement security into the development lifecycle. The best way to approach this is by incorporating the DevSecOps framework so that all the companies’ developers include security in every step of the development lifecycle. It’s also important to list the security principles and how they apply to company decisions such as in coding standards that give descriptions, examples, and threat levels. A good practice when creating a security policy is to include defense in depth, triple-A, and encryption requirements into it as well as these are very important for security and must not be overlooked. Lastly, it’s very important to list the consequences of not following the policy as well as what should be done in outlying cases so that there is a clear chain of command and a concise process for anything that may arise.