**8-2 Journal: Portfolio Reflection**

**Alex Sandoval**

**Southern New Hampshire University**

**CS-405-T2851 – Secure Coding**

**Prof. Aaron Demory**

**December 18, 2022**

This journal will reflect the authors thoughts for the course on secure coding regarding adopting a secure coding standard from the beginning of implementation on a project. The author will consider the many ways that companies consider their own security by evaluating their risks and the cost of justifying defense in depth to their networks. A zero-trust approach is quickly becoming a gold standard in the industry for security, which is leading to more acceptance withing the IT community to trust no person and trust no device. The author will go into more detail about zero-trust and explain his opinion for employing this model on the network. Finally, the author will mention implementations and recommendations for security policies.

The need to add secure coding standards within a company’s way of programming is vital for saving time for unnecessary errors, but ultimately for eradication of vulnerabilities that can be exploited. Each programming language has its own set of rules for producing solid code. For the C++ language, the SEI Cert C++ Coding Standard is a set of rules and recommendations to help users focus on security for their code (Ballman, 2018). The importance of adding security to the development of the code is producing a safer product than attempting to add code at the end of the project which has been proven to be less secure. DevSecOps routinely adds security at every phase of the software development process, facilitating improvement of secure software (What is DevSecOps?, 2020). In the past, security was added at the end of the development lifecycle by a secondary security team and then tested by a quality assurance team (What is DevSecOps?, 2020). The reason this was able to be done in the past was because software updates were merely distributed one or two times a year (What is DevSecOps?, 2020). This process is no longer the case with the use of Agile and DevOps methods that have increased the speed of development to mere weeks or days, which complicates the addition of security at the end of the project an impossibility for safeguarding the code from vulnerabilities (What is DevSecOps?, 2020).

The impossibility for safeguarding code from vulnerabilities leads into the discussion of assessing the risk and cost benefit of mitigating to a more thorough defense of security. There are various ways to address mitigation risk, but the approach can be different for every company’s situation of how they value their data in comparison to the cost to secure that data. The goal is to find the best version of risk mitigation to serve your company’s information based on your budget for expenses and importance of data. Risk mitigation includes a plan of action to diminish the occurrence or attack. If an attack does take place, then the mitigation plan will include instructions on how to act to limit the harm done (Risk Mitigation: Meaning, Types, Aspects, and Benefits, 2022). Several types of risk mitigation are performance testing, integration testing, unit testing, security testing, stress testing, and acceptance testing (Risk Mitigation: Meaning, Types, Aspects, and Benefits, 2022). All of these can significantly reduce issues before they occur and can show the effectiveness of the code prior to launch. Benefits with using risk mitigation are avoiding stress, protecting company reputation, avoiding losses, and increasing efficiency to the projects (Risk Mitigation: Meaning, Types, Aspects, and Benefits, 2022).

The inclusion of zero-trust networking for providing a more robust defense in security is quickly on the rise for many businesses. Previously, the conventional standard of assuming that every device on the company’s network can be trusted is quickly going away to the zero-trust model. Customarily the network perimeters were protected by authenticating a user’s identity only the first time a user or device entered the network (What Is Zero-Trust Networking?, 2022). Now the process has become more rigorous with various “small network perimeters”, each having their specific authenticating conditions (What Is Zero-Trust Networking?, 2022). The zero-trust model philosophy is “never trust, always verify” regardless of being on or off the network (What Is Zero-Trust Networking?, 2022). Common varieties of zero-trust are the addition of multi-factor authentication, virtual private network access, and the use of proxy servers to authenticate and authorize users and their privileges on the network.

The inclusion of secure coding standards with an emphasis on including security built within the code every step of the process is important to the wellbeing of the software development’s security and functionality. The problems that can be avoided are worth the efforts of adding extra time to preserve the code. The benefits of adding risk mitigation are that the company will have a set of instructions to deal with limiting attacks or dealing with the event of a breach. The relief of having such a plan in place can help to fix the problem quickly compared to not having a plan before an attack occurs. The benefits outweigh the financial expenses of protecting your company’s reputation and the data of its customers. Defense in depth, secure coding standards, risk mitigation, and zero-trust are all effective ways to protect from inside and outside threats to a company network. The recommendation is to use a combination of all security methods mentioned when considering financial cost, but if storing sensitive data, the author advocates using all the methods for a more complete defense in security. Providing the best possible security to your network is always going to be an ongoing battle against attackers. Employee security education is highly recommended to help lessen the possibilities of human error of opening malicious emails, downloading suspect executable files, and not locking their computer when stepping away for an extended period. IT security can do its part with network security, but every employee must be a participant in company security by completing annual security training to be aware of vulnerabilities that exist.

**Conclusion**

This journal reflected the authors thoughts for the course on secure coding regarding adopting a secure coding standard from the beginning of implementation on a project. The author mentioned the multitude of options that companies need to consider for their own security by evaluating the risks to their networks and the benefits of adding extra security. The author mentioned a zero-trust approach as a way of never trusting any device or user only one time, but the need to consistently test all devices on the network to provide the best level of security possible. Lastly, the author stated the significance of implementing and recommending security policies to protect sensitive data. The author explained how the benefits of additional security outweigh the decision to cut back on costs to preserve company and customer data.

**References**

Ballman, A. (2018, March 22). *Rules*. Retrieved from Sei.cmu.edu: https://wiki.sei.cmu.edu/confluence/display/cplusplus/2+Rules

*Risk Mitigation: Meaning, Types, Aspects, and Benefits*. (2022, June 24). Retrieved from eclipsesuite.com: https://www.eclipsesuite.com/risk-mitigation/

*What is DevSecOps?* (2020, July 30). Retrieved from IBM.com: https://www.ibm.com/cloud/learn/devsecops

*What Is Zero-Trust Networking?* (2022, December 01). Retrieved from Cisco.com: https://www.cisco.com/c/en/us/solutions/automation/what-is-zero-trust-networking.html