**8-2 Journal: Portfolio Reflection**

Jessica Ayer

December 16th, 2022

CS 405: 22EW2

Instructor: Aaron Demory

Before taking this course, I had heard about many of the security risks covered, looked at the secure coding standards/policies, and studied the SDLC and DevSecOps pipeline in other courses but I didn’t know how to implement most of it. The security features I had figured out on my own, I typically added at the end once I’d completed a project and found flaws in the testing phase. Now that I have these tools, I will be able to utilize them from the start and not leave security until the end. The security policies we developed in this class seem like a great place to start and maintain as a living document.

Adopting a secure coding standard and not leaving security until the end is a practice all future developers need to learn to fight against the entourage of cyberattacks. In today’s world applications need to be developed with the mindset of zero trust. This is especially crucial when developing in the C/C++ language as its structure leave more room for dangerous “memory unsafety” errors (Gaynor, 2018). Writing code in C++ has become my favorite language to work with because of its flexibility but with that comes a lot of responsibility. I know I’m personally not the best at pointers yet but I’ve been learning how to set type parameters and with time comes knowledge. Having secure coding standards will help developers write more secure code from day one and not require excessive remediation on large scale projects.

As a student developer there are free tools at my disposal but for large scale organizations’ higher quality tools will need to be purchased to adopt a DevSecOps pipeline and incorporate automation into the development lifecycle. This means there will be some up-front expenses. It will also take developers time to accept the new process and learn how to use the new tools. It may be hard for developers to switch to a zero-trust mindset if they’ve already been working in the field for years without focusing on security so patients will be needed. In the business world time is money so that will add to the financial risk. Beyond financial risk, while the developers are learning to use the tools there will most likely be a few mistakes made. After taking this course I know I’ve still barely scratched the surface on how to use the tools introduced but it’s a start and practice will bring more knowledge and growth.

The expense of purchasing the tools and training the development staff will be counter balanced by the time and effort saved later in the software development lifecycle as there will be less remediating needed in the final steps and less vulnerabilities that may be exploited in the final project that gets deployed. By starting with security protecting clients’ confidential information becomes priority rather then an afterthought. Making conscientious choices on what data to collect, how long to hold onto it, and who has access to it and then utilizing automated tools to make sure the security measures taken to protect that data don’t have weak spots will, in the long run result in less cost than the initial upfront cost of mitigating to DevSecOps (Federal Trade Commission, 2015).

Though it will be challenging for many to change their process, switching to a Defense in Depth strategy, utilizing the Triple-A (Authentication, Authorization, and Accounting) policy is a great place to start. By leveraging layers of security measures starting with the base code and not just relying on user’s virus scan and firewalls an organizations asset will be less vulnerable (Fortinet, 2022). I recommend, and plan to slowly start adopting security policies and standards into the development one at a time. Soon it will feel natural to include the extra safety measures and code will feel naked or wrong without being wrapped up tightly in security blankets.

**References**

Federal Trade Commission (June 2015). *Start with Security: A Guide for Business.* https://www.ftc.gov/business-guidance/resources/start-security-guide-business#start

Fortinet (2022). *Defense in Depth*. Fortinet, Inc. https://www.fortinet.com/resources/cyberglossary/defense-in-depth

Gaynor, Alex (November 15, 2018). *The Internet Has a Huge C/C++ Problem and Developers Don’t Want to Deal With It*. https://www.vice.com/en/article/a3mgxb/the-internet-has-a-huge-cc-problem-and-developers-dont-want-to-deal-with-it