**Derek G. Braun**

**SNHU CS405: Secure Coding**

**8-2 Journal: Final Reflection**

**04/24/2022**

For the development of a system, code, or project, it is vital to have very specific and effective standards. In CS405 Secure Coding we not only learned secure coding practices, but also created security coding standards for the start-up of a project. Our main goal while writing our secure coding standards is to develop a system with high security, making it impossible or extremely difficult for anyone to penetrate.

This is when our secure coding standards come into play. We must follow our 10 “golden rules”, just like any other standards during a development process. My favorite was “Don’t leave security to the end”. This mainly means to test well and test often. Do not build something great but have vulnerabilities easy to bypass for hackers to steal or destroy data. In today’s world of technology where everyone is connected, we must make sure all vulnerabilities are covered from the ground up. A building cannot be tested when it’s complete after 20 stories. No, that building needs to be inspected on every floor before moving on to the next.

Confluence was able to show me how threats are reviewed and evaluated. This helped divide the threats into specific levels of risk, priority and threat. Being able to categorize and prioritize threats allows a team of developers to manage their time more efficiently, tackling the highest threat first or whichever threat makes the “most sense” first. This can affect the outcome of future threats, current threats, as well as the budget. Time is money!

Having many years of experience in the Aerospace industry (defense private sector/programs), I have been a part of many zero-trust programs. A program that is in development will always trust no one. You are not trusted until specific backgrounds are done, clauses are signed, and access granted. Access is also only granted to those that NEED TO KNOW. “Need to know” is only allowing access or information to those who need to know in order to do their job properly. Today you are seeing more and more corporations implement this zero-trust standard along with Triple A Policies, and encryptions for example.

All these standards and practices are to protect a system’s data. Once a security policy is in place you cannot sit back and ignore it, thinking it will be good for all of eternity. Security policies and systems need to constantly be upgraded, maintained and improved. Developers and security analysts need to be on top of all the newest technologies to be prepared. Knowing a hacker’s tools for example, can be used to properly update/patch these new polices and code. Secure coding and policies will be a constant improvement with all our data being digital, on networks, and who knows where else! Data must always be protected.

**Resources:**

* 1. Bellairs, R. (n.d.). What is secure coding? Perforce Software. From <https://www.perforce.com/blog/sca/what-secure-coding>