Journal: Portfolio Reflection

CS 405 SNHU

Kevin Bristow

This class has opened my eyes to the importance of not leaving security to the end and the adoption of a secure coding standard. From my perspective, all I would hear about in the news was how a hack occurred and that the exposed company was going to work on a solution to prevent future attacks from happening. My assumption back then was that security was added to the software after the code was put into production i.e., you could just pay a company to protect your software. However, this reactive security is rather expensive and is not an exhaustive security plan that is unbreachable; the true solution is to build software with security in mind so that the end product is naturally secure which reduces the chances of dealing with the consequences of being hacked. Often times, implementing security from the beginning starts with adopting a secure coding standard to act as a guideline for those to follow. A secure coding standard ensures uniformity within the project so that the software is developed in the same secure manner.

Through this course I have learned to appreciate the benefit of risk evaluation and assessment and the cost benefit that results from it. Understanding potential risks associated with aspects of code helps me as a developer to write code that protects against those known vulnerabilities. A major bug that has dire consequences that I am aware of every time I write code is when using an array of specified length. Because accessing elements beyond the provided length gives access to unwanted areas of memory, damages can occur. Being able to assess this risk before the code is put into production is extremely valuable and will save lots of money down the road if a hacker were to utilize this error in a breach.

The concept of zero trust is interesting to me because from my experience with a home computer (not entirely the same thing), once signed into an account on the computer, you can do pretty much whatever you want. There are resources that require you to be the admin, but for the most part, the authenticated user can do whatever they want. This obviously raises some concerns once a bad actor has entered the system, which makes the concept of zero trust quite ideal. With zero trust, it implements the principle of “Never trust, always verify” to reduce the risk of a bad actor on the network; users are constantly being verified when they make actions, their devices are validated, and access to resources follows least-privilege. This framework is an ideal security practice for modern technology.

When it comes to security policies, they are vital for any organization wanting to establish a standard for protecting the companies’ assets and implementing robust security measures. From a developer’s point of view, a security policy informs us of the standards we need to be following and the risks associated with them if they were to not be implemented correctly. It provides a guideline for how to approach the writing of the code so that the code is written in a secure and uniform way throughout its entirety. The end goal with the security policy developed in this class is to provide a guideline for the company so they can ensure the implementation of secure coding principles throughout all their applications.