Kerrian Offermann

CS-450: Secure Coding

8-3 Journal: Portfolio Reflection

April 20, 2023

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

Throughout this course, something that has been impressed on me over and over is that treating security like an afterthought of software development can lead to major problems later on down the line. By adopting a secure coding standard to serve as a guide, and by including security throughout the entire process of development, programmers can keep their work safer from threats than if they waited until the end.

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

Evaluating and assessing risks as well as planning mitigation has the major benefit of eliminating/reducing surprises. If the team is aware of the things that can go wrong and how much it would cost to fix these issues then that same team can prepare in advance.

**Zero trust**

There are many lessons I will take from this course; however, I think one of the greatest lessons I will take is that human error plays a huge part in data breaches. The concept of zero trust is a logical one because, in a similar vein as the ‘least privilege’ approach, trust with the database has to be earned rather than handed out to every user who is added to it. I believe that the zero-trust approach is a way to protect the system as well as users who might otherwise make mistakes in the system without having the proper training first.

**Implementation and recommendations of security policies**

There are many different security policies that one could implement and recommend. It can even be overwhelming thinking of all the possible options. However, the one that I would stick to is the one I also chose to focus on for my project. I strongly believe that making regular training sessions and system updates a part of a company’s routine can safeguard against threats. Building this routine will also make following security policies the norm in a company; therefore, it is not seen as something strictly done at the start or end of the programming process but something done constantly.