Portfolio Reflection

Nathanael Burke

CS-405 Secure Coding

I will say that this course has been more in-depth than most I have taken so far. In this journal I am going to be reflecting on the adoption of a secure coding standard, evaluation and assessment of risk, zero trust, and implementation and recommendations of security policies. The adoption of a secure coding standard came about from our work with the ten core coding principles. I quickly learned that there were many things that were crucial to providing security to a system. Some of the principles that I found to be more important that the rest included adopting a coding system that would address security throughout the process of the project. I worked on creating a strategy where before code was even created security was being addressed. This was through the means of trying to understand the motive behind why a user may attack and what it was that they were going to attack.

The way that I addressed the cost benefit mitigation strategy was by taking a look at what might happen if security was not addressed until the end of the project. When security is addressed at the forefront of a project you spend less time trying to patch the system and add security later. What I found to be a helpful guide in how this process works is by the DevSecOps pipeline. The risk of failing to evaluate and assess comes down to how much it will cost later down the road when the issue occurs. This is where the coding standards vulnerability assessment was helpful as there was a section dedicated to the cost of the project if not treated immediately.

Zero trust was something that I knew very little about before this week. The funny thing is that it is something that I have used and seen and interacted with on a daily basis for years now. Implementing a zero trust policy simply makes sense in a project that is heavily reliant on the people interacting with the system. My approach was to add things such as default deny and factors of authentication and reauthentication in order to prevent users accounts from being hacked. I have a lot of experience on the front end of what it looks like to use various forms of authentication and although it can be a lot of work to get through it is well worth it.

Lastly, my strategies concluded with implementation and recommendations. I will always recommend that any company that uses tech or has an email system teach their employees basic safety practices. One of the main examples I use around this was the loss of patient data from a hospital due to an employee falling for a phishing attack. (For more info: <https://www.databreachtoday.com/malware-leads-to-health-data-breach-a-6258>). It is highly important that users remain aware of the technology and how to use it properly. That being said, I recommend the approach again of default deny as this prevents users from accessing things that they should not have access to. This also means that if their accounts are corrupted their access will prevent further attack. Regarding implementation, I always recommend following the DevSecOps pipeline and include security best practices throughout the course of the project. This includes reoccurring static code analysis and other accounting approaches.