Michael Antoniazzi

CS 405 Secure Coding

**8-2 Journal: Portfolio Reflection**

In areas like software security and cybersecurity, throughout this course we have learned how important it is to be proactive when trying to secure systems and data. From the first week of this class when we read the different textbooks and articles about the importance of secure coding with C and C++ it became clear what kind of vulnerabilities exist in computer programming and how those vulnerabilities are taken advantage of. This made it very clear that security should be a part of the development process from the earliest possible stage. When you introduce security at the end, not only is it more difficult to implement, but also you end up with patchy solutions that can leave many different vulnerabilities and exploits unchecked. The better solution will always be to follow secure coding standards from the beginning to save time and costs that come with fixing issues later in the software development cycle.

When evaluating and assessing risk, having the ability to identify, assess, and prioritize risks allows companies to understand the vulnerabilities in their systems and properly allocate resources to help in mitigating those risks. It is important to be able to identify risks because mitigation cannot begin without understanding what the vulnerabilities are whether they are software, hardware, or even with employees. There are many tools available to test for and identify some of the more common vulnerabilities that should always be used. Once you have identified these vulnerabilities and taken steps to understand the impacts they may have as well as the probability of occurring, you can begin to assess the cost of mitigating them. A cost-benefit analysis is done to weigh the risks and costs of either addressing these issues or just accepting the risks. This process is continuous as the landscape is always changing and evolving so regular reviews should be done to keep a proper assessment of security.

Traditional security models work in such a way that once a user has gained access to the network, they generally remain trusted until they have logged out. With a zero-trust policy the assumption is to always be on watch from both threats outside and inside the network. Due to that assumption every request is untrusted and must be authenticated and verified before being allowed access to ensure security remains a top priority. Secure policies are an important part of software and cybersecurity frameworks. These policies should never remain static and should constantly evolve with the threats that exist now as well as in the future.