Portfolio Reflection

This course taught me a lot about secure coding and how to incorporate secure coding policies and best practices in my code. To be honest, in the beginning I was quite worried about this course because the first assignment dove directly into starting to code securely and I had only a little experience with secure coding. Nevertheless, I feel as though it taught be a lot as I persevered through the first assignment. I picked up the pace through the following weeks as well. I learned a lot from creating and adopting a secure coding standard. Being introduced to the SEI CERT guidelines is going to be something I’ll remember throughout my career. Not leaving security for the end is engraved in my mind now. If you don’t use secure coding strategies when you build code, then you will inevitably require a redesign to become secure and this will take even more time and money to do. Think of it this way, locking your car and setting the alarm system while you have your windows down won’t make it more secure. An effective way to incorporate security into your code is automation, developing tools that execute tasks without human intervention. Automation helps to include security early and often in the software development process and enables fast feedback (Johnson, 2020).

The cost and time benefit of mitigating security risks early is also significant. Along with the SEI CERT coding standards, we have a lot of tools, such as cppcheck and even the IDE compiler integrated tools, that helps us to evaluate these security risks. Unit testing is an important part of the development process which also provides a significant reduction in these risks. The cost to fix detected bugs during testing are less in comparison to the defects at higher levels (Gill, 2021). Considering how everything is moving towards cloud-based systems, it is important to utilize the many tools that the cloud providers offer, especially Amazon Web Services. This also means that we need to reconsider how we view security, as many have already after switching to a cloud-based system.

Traditional security methods broadly classify everything, such as users, devices, and applications, inside the corporate network as trustworthy (Kueh, 2020). This method of security is known as the “castle and moat” method, but it has proven over time that it is quite ineffective once a bad actor gets access inside the “castle”. It is because of this that the zero-trust security policy came about, and it has changed the entire view of security policies. Establishing roles and policies for each user, device, session, application, and data, ensures that there is continuous verification of trust across all of these. This model considers all resources to be external and continuously verifies trust before granting only the required access (Kueh, 2020).

I think that the implementation of the policies we had done for our project, and even the zero-trust policy, should be done without questioning why, and instead how? Changing the environment in which we work in by introducing these policies isn’t necessarily easy, but it shouldn’t be extremely difficult either. As developers we should always practice secure coding and we should change our coding practices to best practices available because that mitigates the risk on our work. I recommend that we always stay up to date on security policies wherever we work, even if its just for a small personal project. When we begin coding something new, our first thought should always be, “How can I make my code more secure while achieving the job I need it to do?”.

Work Cited

Gill, N. S. (2021, October 28). *Unit testing techniques and best practices*. XenonStack. Retrieved February 20, 2022, from https://www.xenonstack.com/insights/what-is-unit-testing#:~:text=Concluding%20Unit%20Testing-,What%20is%20Unit%20Testing%3F,and%20produces%20a%20single%20output.

Johnson, P. (2020, June 13). *Secure coding: A practical guide*. WhiteSource. Retrieved February 9, 2022, from https://www.whitesourcesoftware.com/resources/blog/secure-coding/

Kueh, T. (2020, January 15). *A practical guide to zero-trust security*. Threatpost English Global threatpostcom. Retrieved February 21, 2022, from https://threatpost.com/practical-guide-zero-trust-security/151912/