8-2 Journal: Portfolio Reflection

CS-405

Corey Roth

When I think about adoption of a secure coding standard and not leaving security to the end, I’m reminded that this isn’t a new idea for me. As I’m getting close to completing my degree, I can remember in previous courses re-iterating the fact that you should have a security mindset from the very beginning of the software development life cycle and that this mindset should continue on throughout the project. This is the safest way to ensure your code is secure as well as the least time consuming. Imagine getting to the end of a project only to find out you have a major security vulnerability with it, and you have to go back and try and fix this without affecting the rest of the code. This becomes even more difficult with large projects that are worked on in a group.

Security also has to be implemented based on what the project is and what data or information is being stored in it. Financial institutions obviously need to have security on the forefront to ensure their customer’s financials are secure and safe. On the other hand, a company that doesn’t have such critical information probably doesn’t need to spend a fortune or the extra time for a team to develop a extremely complex security program for their code. Life is about balance, just like you determine the amount of insurance you need for your vehicle businesses need to determine the amount of time and money that is going to be spent on security.

When it comes to zero trust, this is a practice that all good developers will follow because they understand the repercussions of writing insecure code and what could happen if an attack would occur. While it an often cause additional steps for the user, having a zero-trust mindset will make you more understanding of why you must take these extra steps to keep the system safe.

Because of this course as well as a few prior courses I plan to implement a zero-trust mindset into all of my projects as well as the best practice of not leaving security to the end. What I have learnt along the way is that a little more time and effort early on can prevent a lot of work later on should an attack happen.