Chase Carter

Tuesday, June 24, 2025

Professor Aaron Demory

CS-405 Secure Coding

8-2 Journal: Portfolio Reflection

“Zero trust” and “no one is safe” makes me see security differently. Instead of trusting anyone or anything by default, the zero-trust model says everything must be verified. It does not matter if a person is already inside the network or using a company device. This means I now see security as something that must be checked at every point, not just at the login screen. Access needs to be limited and constantly verified. As a user this means I will need to get used to more checks and security steps. I will have to sign in more often or use multi factor authentication. Even if it’s annoying, I understand that these steps protect my data and the system. The mentality of zero trust will be shown to fellow developers by explaining that threats can come from inside the system, not just from outside. I will show real cases where zero trust could have stopped data breach. Developers care about building strong systems, so showing them how zero trust helps makes it easier to get their support. I will also recommend secure coding habits like input validation to make sure each user has the right permission for each task.

This course overall showed the need to use a secure coding standard from the start. Leaving security to the end causes problems. This class has shown me how to evaluate risk and find the cost benefit of mitigation. It described a risk matrix to compare threat likelihood and fix cost. We should focus on top level threats first, because that gives the highest return on investment. We should write clear rules that assign each step to a person. Every team member must be trained in these rules before they code. We must run regular audits to make sure rules are followed. We use tools like CPPCHECK and others to spot policy breaches and alert the team if one occurs. For example, we can set rules for encryption using strong algorithms and secure key lengths. We can review these rules every three months to keep them current. These steps help build a strong security culture.