Throughout this course, I learned the importance of security and how essential it is to keep security at the forefront of development. It should be ingrained in the process of development and embraced as part of the agile approach. In effect, this would involve adopting coding standards that are mutually agreed on by the development team. Then, coding becomes a process where the secure coding standards are constantly in consideration and adhered to as development progresses. The idea is that, at the end of development, you aren’t scrambling to include secure practices into your work because it was always being considered and tested for. Often, mistakes are made when developing without security in mind. You simply can’t rush to solve security at the end of the process.

Implementing zero trust as a coding policy helps to align with “keep it simple”, one of my secure coding standards. By modeling zero trust policy in your coding, you are requiring authentication from everyone by trusting nobody. You are simplifying the process by saying “I don’t trust any user, I’m going to require all users to authenticate”. This puts a lot of the burden on the user to get the correct permissions setup with their profiles.

If I were to pick a single security policy to implement, it would be protecting against SQL injection. This can be done by validating user input so users can’t simply enter any string of data into a database. If malicious intent is present, buffer overflow could occur and SQL injection commences. This is a common attack vector on poorly secured systems and should be addressed during development, not at the end.