Joseph Valle

Cs405

Portfolio Reflection

3/3/24

When it came to developing coding throughout my time here at SNHU, surprisingly secure coding was rarely brought up nor any online coding “bootcamps” you do to learn certain languages. A lot had to do with the readability of the code. Taking this course was great as it consistently just engraved secured coding into your head. Not only did it make sure that you knew secure coding was extremely important to the development of your application but it also different types of secure coding and how each can all intertwine with each other. This was extremely important because of all the code we’ve written here at SNHU I have never thought of or added even simple secure coding techniques to the code. So now looking back on adding secure coding techniques into the small codes we developed I can only imagine what the process would take to do it into a full tack application. It also may have saved me some time on original projects as it would have caught errors in the moment rather than when I would run the code and would have to decipher multiples of errors.

When evaluating risk vs, the cost benefit of mitigation we usually look for what the potential of a risk can cause and the overall impact it may have on us if not handled properly. How much time and labor would it take to analyze, correct, and fix the error or attack. What’s the possible losses if the error or attack can impact our business or application. This is also thought of when developing a code. What’s the worth of user data if it is breached. What would of it costed us to add an additional layer of defense. These are the questions we set a standard for at the beginning of the development rather when we already got into this issue.

Zero Trust is a way of thinking that trusting is naïve and always verifying is the proper step in securing your code. Yes, internal attacks do happen, but also the possibility of having an internal account be compromised roaming around the development ecosystem freely. I’ve experienced this as a company I had worked for had an internal account compromised that ultimately was able to infiltrate user data. This happened to a user that was left at rest for months that nobody seemed to deactivate or delete causing this issue. So here we could have put the concept of Zero Trust to play as first we would strip credentials off this rest account, add multiple factors of authentication to access data. Even though it’s an internal account, you never know how an attacker is thinking of infiltrating which is why no trusting any account is first then making sure they are verified I second.

Implementation and recommendations of security policies should come at the begging of development. Standards should be outlined and given to all developers. They should be a part of the organization requirements while also following industry best practices. These should also not be forgotten about once the application is deployed but should also be adopted into the maintain of the applications code for its lifecycle.