Sarah Snow

CS405 Secure Coding

Module 8 Journal

Adopting a secure coding standard should be a large part of a company’s approach to programing. While readability, organization, and quality are important, it doesn’t mean much if the program is full of vulnerabilities that can be used by malicious users to access and exploit sensitive data. Additionally, leaving security to the end would be a huge misstep when designing a project. That can lengthen the time it would be until the product can launch, which can lead to a rushed approach to security. Incorporating security into every part of the project ensures a well-rounded and thorough defense system with fewer vulnerabilities falling through the cracks.

While security should be on the same level of consideration as design and functionality, the coding standard must be balanced. Zero trust is the best approach to implementing a secure coding standard; however, the user experience must be considered as well. Assessing the likelihood, severity, and remediation cost of potential breaches can guide the implementation of security measures. Keeping things simple can help with a zero-trust policy, such as multi-factor authentication and straightforward ways on how to reset credentials should the user forget their password.

The best approach to recommending and implementing a secure coding standard starts with a thorough assessment and understanding of the project and what kind of data it entails. That will be a solid foundation on which to build the standard, because after understanding comes the ability to assess what kind of risks there would be to the data, and so forth. Ultimately, security is about balance.