**CS 405 Journal**

Mitch Sfakianos

April 23, 2023

Adopting a secure coding standard is important for many different reasons. Coding standards such as CERT for C and CPP provide information on many different errors programmers could make and how to prevent them. Following the right coding standard depending on the language can lead to finding many vulnerabilities within code that may not have even been caught by a standard debugging tool in an IDE.

Not leaving security to the end of the programming process is a key concept within secure programming. Applications should be created not only with the core functionality in mind but also with security aspects in mind as both should be developed together. Programming software should be done with a mindset regarding how malicious actors could misuse it and then take preventative measures within the code itself to prevent them. This relates to a zero-trust policy when it comes to allowing access to privileged actions within the software. A hierarchy of privileges should exist for users with a robust user authentication system, and complex privileges should be denied by default if the user does not meet those requirements.

The evaluation and assessment of risk and cost benefit of the mitigation of those risks is something every programmer must consider when looking at vulnerabilities within their code. Typically, coding standards will have levels defining different standards' likeliness, severity if left unfixed, and the resource cost of fixing that error from high to low costs. Higher severity risks with lower costs of mitigation should definitely be prioritized.