Marye Bierbaum

CS-405 Secure Coding

8-2 Journal: Portfolio Reflection

The adoption of a secure coding standard is important because it guides developers on practices that will limit the number of security issues we incur and provides a guidebook for how to handle security vulnerabilities once they are identified. The standard establishes an entities strategy for ensuring they build secure applications. This can help reinforce important ideas, like starting with security in the earliest stages of the DevSecOps lifecycle. Thinking about security while designing and building the code allows developers to identify vulnerabilities early, when they are the cheapest and easiest to fix.

Evaluation of issues is an important skill for developers and security professionals. Not all vulnerabilities are created equal, and instead they vary in severity, probability, and cost of remediation. Priority should be given to the issues that are the most likely to occur and that have the biggest impact. Issues should also be evaluated to ensure it is understood what the impact of leaving vulnerabilities open could have. What kind of attack can this issue spur? How would it be handled and what would be the associated cost? These are important questions to consider with assigning prioritization and making action plans.

Another important strategy in security is zero trust, in which each system assumes every other accessing system is hostile, even ones within the enterprise or on the same server. Assuming zero trust greatly reduces the risk of unauthorized access to systems.

Implementing a large security standard can be a challenge for any organization. Disseminating, explaining, and training to the standard should be given priority. Even the best policies in the world do nothing if they are not practiced. Managers and leadership should emphasize the importance of following secure standards as well.