Coral Stewart

CS 405

Professor Andujar

22 April 2025

Portfolio Reflection

**Adoption of a secure coding standard, and not leaving security to the end**

By utilizing the DevSecOps pipeline and CI/CD, security becomes part of every stage of the software development lifecycle, and rightfully so. I appreciate this pipeline because security should not wait until the end- there should be mitigation and continuous checks throughout. Waiting until the end can have its consequences, such as cost and having to go back and make changes to the code itself. Automated testing used within the DevSecOps pipeline makes security easy and consistent.

**Evaluation and assessment of risk and cost benefit of mitigation**

Looking at the severity of risks, their likelihood of occurring ands their priority will paint a picture for developers so they know what must be avoided when enforcing the 10 core security principles. Having a clear picture of what risks there are, and the tools to use to avoid them, will save any company money in the end. Mitigation is important with security, because it means developers are actively preparing ands trying to prevent any security mishaps from occurring. There are so many tools at every developer’s disposal, especially when it comes to mitigating security.

**Zero trust**

I was introduced to the concept of zero trust and it instantly reminded me of the principle of least privilege and default deny. Although implementing zero trust may be inconvenient for developers, I think that it is exceptionally great for security. It is important to understand that threats can come from the inside, and we are not always protecting sensitive information and data from malicious users on the outside. Zero trust aims to keep users more honest by using different access methods, such as biometrics (fingerprint, face scan).

**Implementation and recommendations of security policies**

Implementing security practices may come as a learning curve to developers, but it is worth the extra training, worth the extra step, and worth the occasional headache. Some developers may not realize how sensitive the data is that they work with, and enforcing security policies continuously is the best chance of protecting sensitive data. I would recommend using DevSecOps to the fullest extent, as well as zero trust.