Dennis James Stelmach

CS 405

Professor Olga Mill

June 30, 2024

Module Eight Journal: Portfolio Reflection

# Standards: Building a Secure Foundation

One of the most crucial lessons learned is the importance of adopting a secure coding standard from the very beginning of the development process. It's not something to be tacked on as an afterthought. Think of it as laying a solid foundation for your building – a secure coding standard provides a clear framework for developers to follow, ensuring they write code that is less susceptible to vulnerabilities. This proactive approach aligns with the concept of "Defense in Depth." Just like layering security measures in a physical building (alarms, security guards, etc.), a secure coding standard acts as the first line of defense, preventing vulnerabilities from becoming exploitable weaknesses.

# Risk Assessment: Balancing Security and Cost

The class emphasized the importance of evaluating and assessing risk. We learned to weigh the cost of implementing mitigation strategies against the potential benefits. It's about finding the optimal solution, just like navigating a complex problem. Should we invest heavily in securing a rarely used piece of code, or prioritize shoring up core functionalities? This requires careful analysis and understanding the threat landscape. It's a constant balancing act between ensuring high levels of security and keeping development costs under control.

# Zero Trust: A Security Mindset

The concept of "Zero Trust" resonated deeply. It challenges the assumption of trust within a system, emphasizing the need for constant verification. This principle directly translates to secure coding practices. We cannot blindly trust user inputs or external data. Every input needs to be validated and sanitized to prevent potential exploits. We can draw an analogy to a secure building – just because someone has a key doesn't mean they have access to all areas. Secure coding ensures that only authorized data can access specific functionalities within the program.

# Implementing Secure Policies: Putting Knowledge into Practice

The highlight of the course was the project where we designed a security policy for C++ and presented it. This practical experience underscored the importance of clear communication and implementation strategies. A well-defined security policy acts as a roadmap, guiding developers on best practices and fostering a culture of security within the team. Just like having clear guidelines for entering a secure building, a security policy ensures everyone involved in the development process understands their role in maintaining code safety.