**Module 8 Journal**

**Justin Hancock**

**CS-405 Secure Coding**

**10.21.2024**

Throughout this course, the importance of embedding security at every stage of the development process became increasingly clear. By adopting secure coding standards, assessing risks early, and implementing a zero trust model, I’ve come to appreciate the proactive approach to security. Below are reflections on key topics covered during the course, drawing from both my projects and readings.

One of the primary takeaways from this course is the value of adopting a secure coding standard from the start, rather than treating security as an afterthought. By incorporating practices like input validation and encryption during development, vulnerabilities can be significantly reduced early on, minimizing costly rework later. In my projects, I implemented secure input handling and encryption mechanisms, such as XOR encryption, which reinforced the idea that securing code as it's written is far more effective than retroactively addressing vulnerabilities. By shifting my mindset to view security as an integral part of development, I’ve become more diligent in how I structure code. This approach aligns with industry best practices, where addressing security early not only saves time and resources but also strengthens the overall resilience of the system. Security is not a feature to be bolted on later, but a continuous process embedded into every line of code.

Risk assessment played a central role in my secure coding projects, helping me prioritize vulnerabilities and decide on appropriate mitigations. For instance, when implementing input validation to prevent SQL injection, I considered both the potential risks of a successful attack and the cost-effectiveness of mitigation techniques. This balance between risk and cost was vital for determining how to best allocate resources without compromising security. Conducting threat assessments allowed me to understand the practicalities of securing code, especially when faced with limited time or resources. The cost-benefit analysis was a key factor in deciding which security measures to implement, reminding me that not all risks can be eliminated, but they can be mitigated effectively through strategic planning.

The zero trust model, which assumes no user or device is inherently trustworthy, reshaped how I think about system security. In adopting this approach, I realized the importance of continuous authentication, access control, and network segmentation. Every request or interaction must be verified, limiting access to only those who absolutely need it. In my projects, I applied zero trust principles by strengthening authentication processes and ensuring that all user interactions were validated through role-based access controls. This model challenged the more traditional security practices I had previously worked with, reinforcing the need for constant vigilance and validation, regardless of a user’s network location.

Developing and implementing security policies throughout the course gave me a clearer understanding of how best practices can be translated into organizational policies. For example, in my security policy projects, I focused on areas like input validation, access control, and encryption to protect sensitive data. These policies reflected the core principles of least privilege and defense-in-depth, ensuring that each layer of the system remained protected. Recommending security policies involves not only understanding the technical aspects of secure coding but also the operational challenges of enforcing them. By aligning these policies with the secure coding standards I practiced, I was able to create a more comprehensive approach to addressing security across different layers of the system. My recommendations emphasized simplicity and practicality, focusing on solutions that were both effective and feasible for developers to adopt.

This course reinforced the importance of treating security as a continuous process that begins at the earliest stages of development. By adopting secure coding standards, evaluating risks, implementing zero trust, and developing sound security policies, I now approach development with a deeper understanding of how to proactively mitigate threats. Moving forward, these lessons will guide my coding practices, ensuring that security is always a priority, not an afterthought.