### Southern New Hampshire University

### CS 405– Secure Coding

### 8-2 Journal: Portfolio Reflection

### Kenneth Wilkerson

### Professor Andujar

October 26, 2025

**Journal**

In this course, I have learned that adopting a secure coding standard early in the development process makes sure that security is built into the foundation of the code rather than treated as an afterthought. Standards such as CERT C/C++ or OWASP provide clear rules that help prevent common vulnerabilities such as buffer overflows, improper input handling or insecure API usage. As highlighted in our textbook Secure Coding in C and C++, consistently applying secure coding standards throughout the software development life cycle reduces the long-term cost of fixing security issues after deployment.

The evaluation and assessment of risk are crucial in determining how to use resources effectively. There is a cost that comes with every mitigation measure and understanding the tradeoff helps decide where to focus. Using structured risk frameworks allows developers to measure likelihood and impact which supports informed security decisions. In terms of the zero trust model, I have learned that nothing should automatically be trusted. Zero trust means that there is always authentication, authorization and validation happening at every request. This aligns with secure coding practices like input validation, least-privilege design and strict session management. Implementing zero trust also encourages developers to code defensively and assume that breaches can happen internally as well as externally.

Lastly, the implementation and recommendation of security policies ensure that secure practices become part of the standard operating procedures instead of being left to individual choice. Policies that define encryption standards and the Triple A framework (Authentication, Authorization and Accounting) provide structure for users and developers. Overall, secure software development depends on strong standards and realistic risk assessments. These practices create systems that are resilient in a constantly evolving threat environment.