Jyorish Sainju

CS 405 – Secure Coding

Southern New Hampshire University

25 August 2024

8-2 Journal: Portfolio Reflection

Over the past eight weeks, we have focused on applying secure coding standards to protect software and applications from vulnerabilities. With hackers continuously developing new methods to exploit weaknesses, a reactive approach is insufficient for safeguarding personal and organizational data. Instead, adopting and consistently practicing secure coding standards allows developers to stay ahead of potential threats.

Implementing strategies such as defense in depth and the zero trust concept is crucial for maintaining data security. These practices ensure robust protection by incorporating various layers of security and continuous validation. Key aspects of secure coding include proper data input validation, data encryption, rigorous authentication, the principle of least privilege, role-based access control, and effective dependency management.

Defense in depth enhances security through multiple layers of protective measures, creating a comprehensive defense against potential breaches. The zero trust model requires continuous authorization at every step, ensuring that no access is granted without verification. Data input validation helps prevent attacks by verifying each piece of data or command before it is processed. Data encryption secures information by rendering it inaccessible without the correct decryption key, thus protecting it from unauthorized access. Meanwhile, the principle of least privilege minimizes the risk of exploitation by limiting permissions to only those necessary for the software’s operation.

By integrating these secure coding practices, developers can create more resilient and multi-layered security environment systems and protect against evolving security threats.