**8-2 Journal: Portfolio Reflection**

Anthony Vigil

CS-405: Secure Coding

Professor: Mike Prasad

Southern New Hampshire University

**8-2 Journal: Portfolio Reflection**

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

Adopting the secure coding standard during the process of development is important as it can help save time and money in any future steps. The structure of the application should be built along with the security; This will ensure that every aspect of the application is secure as it moves forward. It would be time-consuming to go back and reinforce all the warnings or potential risks in the code. With time double checking, the resources and money will also be used to reinforce it along the way.

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

Evaluating the coding standards for the application is important as it ensures the code is compliant and meets the requirements of the client. You can always evaluate the code using the google test, g test, or using the window on Visual Studio Code that shows the errors, potential risks, and provides quick solutions. You can access the code which will benefit by finding the severity, remediation cost, likelihood, and severity.

* **Zero trust**

Practicing zero trust helped with assuming not only the original user will use the login. Others will try and get into accounts that are not theirs so steal information from them. Two-step authentication would help keep the user login credible to the original user. Assuming any information such as phishing scams are not real, the user will prevent themselves from being breached.

* **Implementation and recommendations of security policies**

Implementation and recommendation of the security policies would help in secure the user will not be hacked while also keeping it ethical. The four implementations and recommendations I would consider are: In-house (sensitive information), In transit (sending sensitive information), back-end security, and front-end security. Protect all sensitive information that is idle by encrypting it or storing it in a secure location. While sending information, it is best to protect the sensitive data by encrypting it and only giving the decryption key to the end user. Protect the server on the back end of the application. Constraints on the input for validation would help secure the front-end.