Isaac Trudeau

**8-2 Journal: Portfolio Reflection**

Adopting a secure coding standard and not leaving security to the end is very important because it allows us to implement security throughout the software development life cycle rather than trying to implement it at the end and failing to cover all our vulnerabilities. Using established guidelines like SEI CERT standards and industry standards we can avoid the possibility of introducing common vulnerabilities into our code such as buffer overflow or SQL injection.

Evaluating and assessing risks helps us understand where our security needs to be prioritized and how to best mitigate security risks. It can also keep us in the mindset of a hacker to determine what security vulnerabilities exist and need to be protected against. We can determine high priority risks like the ones that allow for data breaches and low priority that may just give us bad results in data.

The zero-trust model tells us that we should treat everything as a risk and use proper validation to determine whether it can be trusted. Nothing should be automatically trusted. This works with the principle of least privilege where users are only given the necessary permissions and nothing more. This type of model keeps systems secure and makes it more difficult for hackers to exploit.

Implementation and recommendations of security policies involves creating clear and direct guidelines for how security should be managed. These policies should follow industry best practices and industry standards. Creating a security policy document that can be reviewed and followed is important to keep everyone at a level of understanding about security integration in the development lifecycle.