8-2 Journal: Portfolio Reflection

Throughout this course we have learned of the importance of not leaving security until the end. By leaving security to the end, you open the door to the dangers of errors in your code that can lead to vulnerabilities. These vulnerabilities can be exploited and result in security breaches, data breaches, and other malicious attacks. These attacks can cause significant reputational harm to the organization and even financial losses. On top of these risks, the longer it takes to fix these vulnerabilities, the greater the cost and difficulty the process becomes.

This is why it is imperative to adopt a proactive defense in depth approach in applying security measures, such as the Green Pace Security Policy we have worked with throughout the course. By adhering to secure coding principles and standards throughout the development process, many of the dangers and costs of vulnerabilities can be mitigated. This approach, such as incorporation like the Green Pace DevSecOps toolchain provides continuous support of vulnerability scanning, testing, and addressing that lasts throughout the project lifecycle. While this approach can avoid the financial losses of addressing security issues late in the development process, it will imply an upfront cost to imply. Considerations should be made for the sensitivity of the data being utilized by the system and the upfront budget available for implementing security measures to protect that data.

Zero trust is another best practice security approach that involves not trusting anyone inside or outside of the network. Traditional security approaches used perimeters that monitored and protected anything entering or exiting the network, with anything inside of the perimeter considered trusted, while anything outside was considered untrusted. With this traditional approach, attacks that breached the network security perimeter caused great risk as attackers had trusted access. This is changed with the zero trust approach as the focus of the security perimeter is instead focused on the user rather than the network. In zero trust, authentication and verification is utilized for accessing any service within the network. This approach allows a better method of mitigating security risks of unauthorized access to sensitive data or services within the network in comparison to the traditional approach.

Overall, we have learned of the importance of prioritizing security. That defense in depth is key in providing a strong foundation for security, and that it should be supported throughout the lifecycle of the project through examples like the DevSecOps toolchain in the course project. We have learned of the many dangers associated with leaving security to the end, and the reputational and financial losses it may cause organizations. Secure coding policies and standards, such as the ones we constructed in this course, should be adhered to and utilized to mitigate vulnerabilities throughout the development process and provide secure code moving forward.