Daulton Truett

03/03/2024

CS-405: Secure Coding

Southern New Hampshire University

8-2 Journal: Portfolio Reflection

Security is a critical component to implement early and throughout development of any piece of software and system architecture. The attack vector for software has grown significantly over time and this trend will continue as long as malicious actors formulate creative means for exploitation for a multitude of motives. One of the key methods for implementing security early and throughout the development process for software is developing a secure coding standard. A successful coding standard provides a strong foundation for developers to follow during the development process to prevent writing security vulnerabilities into the code for future exploitation. Another benefit of developing a coding standard is the early adoption of security into the development process, which prevents a common software development mistake of leaving security to the end.

An important step when developing software is evaluating and assessing risk. The level of risk each company/software is willing to accept, as well as the level of risk the company/software may face will vary and must be carefully calculated to meet the required security requirements to prevent exploitation. Cost benefit of mitigation strategies pairs with assessment and evaluation of risk as it frames how and when to implement security measures. Common methods to mitigate risk are “acting now” and “acting later”. Acting now is preventing security vulnerabilities from making their way into production, therefore reducing the potential attack vector and probability of exploitation. Acting later is an assessment on how to respond to exploitation of unforeseen vulnerabilities. It’s an important security measure to practice since unforeseen vulnerabilities happen frequently and are seen as some of the most damaging exploitations due to the unknown consequences they hold.

Strict security measures can be taken with the concept of a “Zero Trust” policy. As the name implies, trust is not inherently given to any device, user, data, or resource as it must be earned by consistent verification. There are major benefits with this approach, with the most notable being the reduction in speed and the increase in complexity for malicious actors to exploit potential vulnerabilities. A problem with a Zero Trust policy is the need for constant verification that users will be expected to provide that they may likely find a nuisance. However, the benefit of severely reducing the potential attack vector is worth the inconvenience that users will experience with consistent verification, and one that many users will likely appreciate in the long term.

Finally, a security policy should be created to create a consistent plan that all team members will follow to ensure all established security measures are followed during development. Security policies provide a comprehensive path that all developers/team members utilize during development to guide decisions, design, code, and overall architecture to ensure the implementation of security best practices. An important note on security policies, they should NEVER become static as security is dynamic and the policies must be able to adapt to changing threats. Security is one of the most important components in software to be forward-thinking and adaptable in anticipation of newly discovered threats and exploitations, therefore, it’s wise to review security policies frequently to expand or contract them as needed.