Joseph Les

CS-405 Professor Hodde

Don’t leave Security to the End

"Don’t leave security to the end" is exactly as it sounds. When designing a new system and/or software, it is crucial to integrate security throughout the entire software development lifecycle. This ensures the safety of both the company and its customers. This approach encompasses various categories, such as threat modeling, secure development practices, continuous security testing, and auditing. By addressing security from the start, costs are reduced by avoiding the need to backtrack and fix vulnerabilities, and it ensures the company meets compliance requirements.

Incorporating security principles from the very beginning of the software development process sets the stage for maintaining security throughout the entire process. Designing systems with security in mind involves considering potential threats and planning appropriate defenses from the start. This can be achieved by conducting threat modeling early in the project to identify and understand potential threats and vulnerabilities. This allows the team to design countermeasures and mitigate risks before they become more difficult to address. Additionally, it is important to conduct regular testing and perform audits frequently to identify any potential gaps sooner rather than later.

The leading benefits of early security integration are reduced costs, enhanced security posture, and meeting compliance requirements. Identifying and fixing security issues early in the development process is significantly cheaper than addressing them after the product is completed or after a security breach has occurred. A proactive approach to security leads to more robust and resilient software. Moreover, many industries have regulatory requirements for security. Integrating security early helps ensure compliance with these regulations, building trust with users, clients, and stakeholders.

To ensure security is at the forefront of Project Two, I will integrate input validation unit tests throughout the development process. Every function or module that accepts user input will include unit tests to verify that the input meets the expected format and constraints. Using a unit testing framework such as JUnit to automate the validation process will help ensure the tests are run frequently and are repeatable. By integrating these unit tests from the beginning and running them continuously, we ensure that security is a fundamental part of the development process, reducing the risk of vulnerabilities being introduced into the codebase.