John Munguia

CS 405

August 17, 2025

**Journal Response**

In secure coding, the phrase **“Don’t leave security to the end”** emphasizes the need to integrate security practices throughout the entire software development life cycle rather than treating it as an afterthought. If developers postpone security until the testing or deployment stage, it often leads to costly rework, missed vulnerabilities, or rushed fixes. By contrast, embedding security considerations into each phase such as requirements gathering, design, implementation, and testing helps ensure that the final product is both functional and resilient against threats.

To prevent threats effectively, developers can take several proactive steps. One of the most important is **threat modeling** early in the design phase, which helps identify possible attack vectors before any code is written. Another step is **input validation**, ensuring that all user data is checked for malicious patterns to prevent attacks like SQL injection or buffer overflows. Developers should also follow the principle of least privilege, giving systems and users only the access they require. Finally, incorporating **regular code reviews and unit tests** that focus on security-related scenarios can catch issues long before they become exploitable vulnerabilities.

For my **Project Two presentation**, one example I plan to highlight is the use of **unit testing to validate input handling**. Instead of assuming that user input will be clean or valid, I will write tests that deliberately attempt to break the application by passing in malformed or dangerous data. This ensures that the system is resilient against threats such as buffer overflows and injection attacks. By integrating these tests into the regular development cycle, security is not treated as a one-time activity but as a continuous process built into the foundation of the code. This approach supports the best practice of addressing security intrinsically, ensuring that issues are discovered and resolved before they can pose a real risk.