CS405: 8-2 Final Reflection Journal

Michael Ross

Department of Computer Science, Southern New Hampshire University

CS405: Secure Coding

Prof Prasad

June 25th, 2023

# Journal – Final Reflection

## Adoption of a secure coding standard, and not leaving security to the end

By adopting a secure coding standard, it integrates security into the earliest phases of code generation. The standard can be integrated into the pipeline for easier checking of compliance with the standard. When a standard is present, it can even be accounted for in the planning process for a piece of software. In this way, security is not left to the end.

## Evaluation and assessment of risk and cost benefit of mitigation

Not all vulnerabilities or code standards are created equally. There is a level of severity to a vulnerability that must be considered. The more severe a vulnerability is, the sooner it should be addressed. In addition to the severity, the ease of remediation should also be considered. For vulnerabilities which are not critical, the ones that are easier to fix should be taken first. The rest can be planned and prioritized based on the severity, and ease of remediation.

Some vulnerabilties which have a low level of impact and a high cost for remediation may never be addressed if the costs are too high.

## Zero trust

By assuming no systems are trusted and going from there, a piece of software can be made secure. Not allowing any access unless specifically allowed ensures that attackers cannot bypass security by exploiting a bad default security rule because all default rules will be denied. This increases complexity but adds a large amount of security to a project.

## Implementation and recommendations of security policies

Implementation of the security policies and recommendations should be handled using a phased approach. This ensures that the highest priority recommendations can be handled first, and the rest can be queued up for remediation when it makes sense.