CS-405-10861-M01 Secure Coding 2024 C-3

6-1 Journal: Don't Leave Security to the End

Computer Science Department

Southern New Hampshire University

Eric Farkas

6/16/2024

The statement “Don’t leave security to the end” reminds system and software developers that security should be an ongoing and ever-present element in their work. Addressing security elements at the end of development leads to rework, oversights, and vulnerabilities discovered after deployment. A holistic approach to security defines threats and sensitive data during planning and early development. Thinking about these elements early allows the developer to properly format data elements and determine if the data stored is sensitive. These practices ensure that encryption is used where necessary and validate that any input contains only the information the software or system expects. This is often part of a comprehensive risk assessment performed before or early in the development. This assessment helps developers identify security risks, and testers write plans to ensure the risk has been mitigated within the code. Systems requiring users will include access levels or groups that should be analyzed to ensure the least permissions are extended to users at each level to accomplish their tasks. Extending unnecessary permissions to users can introduce security risks should their accounts be compromised. Password criteria should be determined by the sensitivity level of the accessed data. Strong passwords reduce the risk of an unauthorized user gaining access to an authorized user's account.

Testing will be performed as development progresses. Code reviews will help testers focus on the interactions that introduce security risks and their mitigation strategies. Static, unit, and penetration testing are common testing elements when considering security risks. Static testing is a highly automated testing method that reviews code for security issues before its first execution. Unit testing is customizable and helps development teams align with specific company practices or policies for many elements, including security. Penetration testing requires the code to be executable and thus begins later in development. This testing simulates various security attack vectors. The results are recorded and placed in a report. Successful attempts must be reviewed to determine whether the success is an actual vulnerability, a configuration issue, or a false positive.

While no process will create impenetrable software and systems, considering security during planning and early development will significantly improve their security. Keeping security in mind when performing testing is a secondary way to enhance security and ensure the mitigation strategies developers use are effective. These proactive development steps are essential to launching products without security flaws, but a maintenance plan should also be established before launch. This plan outlines code elements that may evolve and documents the framework and third-party integrations used in the system. It will serve as a blueprint for developing and testing future versions and a checklist when vulnerabilities are identified in supporting modules.

Project Two will include the coding standards, encryption, Triple A framework, unit testing, and automation discussed herein. This layered defense approach results in a high-quality, low-vulnerability product that complies with best practices. The product will be easily maintained through future versions and documented using methods that convey strategies to alternate developers tasked with supporting the system in the future.