# **CS 405 6-1 Journal: Don’t Leave Security to the End**

# **Austin Donaubauer**

# **Southern New Hampshire University**

# **CS-405 Secure Coding**

# **Wasim Alim**

# **August 7th, 2025**

# **CS 405 6-1 Journal: Don’t Leave Security to the End**

Security in the development of any product should be at the forefront. Thinking ahead and developing coding standards and secure policies should be a major focus when beginning a new project. The phrase “Don’t Leave Security to the End” means just that. Developers should be thinking about secure programming and good standards while developing. You don’t want the developers to create the product and then wait until the end for testing to be done because it will cause delays when inevitably the code is found to be insecure. Security needs to be integrated throughout the whole software development lifecycle to be able to create the best product possible, and to be efficient in development. Waiting until the end to test and implement security can lead to delays and many refactors to the project. Time is very important to most companies so these delays would cost them money, and in turn would cost the developers money too when companies decide to get a new developer to create the product instead.

To protect against attacks and vulnerabilities, a developer should use coding standards and security policies. Some good standards and policies would be encrypting data, exception handling, and runtime errors like buffer overflows and numeric overflows. Developers that code with these things in mind help prevent threats to products. Some other threats like SQL Injection also need to be handled. Preventing unintended queries is important so that access to applications and access to data is not allowed. Validating the input is a key way to prevent SQL Injections from being successful. Static analysis tools are also a good way to analyze your code so that you can find potential issues that don’t always get caught by the console.

An example of something I could include in Project Two presentation to ensure security is not an afterthought would be to use static analysis tools to check the code. Static analysis can pick up on potential issues like incorrect function calls, unused libraries, style issues, and provide warnings for issues. Another potential example would be unit testing. Unit testing is a great way to validate your code with tests to make sure the intended results are correct. Writing and running unit tests throughout development is important so that you aren’t waiting for the end of development to test and then have to refactor everything and redo a test. Testing during development allows you to refactor in real time to be more efficient.