Dylan Coulter February 16, 2025

ID#1896062

C405

# **CS405 6-1 Journal: Don't Leave Security to the End**

**Explain what the following statement means as a best practice in secure coding: “Don’t leave security to the end.”**

Don’t leave security to the end means not considering or implementing security measures during the creation of a project and pushing the research, identification, and mitigation of possible vulnerabilities till the end or final stages of a project. Leaving security till the end can have major impacts on a project such as developing something that has many vulnerabilities that can easily be exploited. This can have massive cost and reputation implications for organizations that utilize the developed product. Not only can this have a large cost if there is a security or data breach, but the cost to back track through a project to identify vulnerabilities can be very costly. Ensuring that we are considering security threats at every stage of development is not only cost effective, but more efficient. Not only is it important to identify security threats at every stage it is important to continuously be considering new threat throughout the life of the product as new threats can arise at any time.

**Describe the steps you can take to prevent threats.**

There are different steps that can be taken to prevent threats. These include using a multi-layer security system. Multiple layers of defense are more effective than a single set of defenses. With this it is also important to consider the most efficient number of layers of defense for the situation. What is being protected should be considered. For example, a complex defense system for data that is not sensitive would be counterproductive as with complexity comes greater costs, higher maintenance, and performance implications. There is also the use of white-box, gray-box, and black-box testing. These three forms of testing will help identify risks and vulnerabilities from different angles. For example, black-box testing would test the vulnerabilities of our system from the viewpoint of and external hacker who has no knowledge of the innerworkings of the system. The use of peer reviews before the submission of code is also important as this allows for a larger possibility for issue to be identified and corrected before submitting to the main project. Lastly the use of multiple automated tests is crucial in identifying issue. Unit tests are key in identifying issues and there are many unit testing tools to be utilized such as JUnit tests, NUnit tests, and DBUnit tests to name a few. There is also Cppcheck which is a static code analysis tool which can help detect issues.

**Provide one example that you can include in your Project Two presentation of how you plan to ensure that security is addressed intrinsically and not left until an issue is discovered—for instance, the use of unit testing.**

A great example that I plan on including in my project two presentation of how I plan to ensure that security is addressed intrinsically and not left until an issue is discovered is the inclusion of unit testing. In previous courses in the program, I worked with JUnit testing which really showed me how important testing is to creating a complete well function and secure project. In this class we worked with other forms of tests to help us identify issues that the IDE did not detect. One of these testing tools was Cppcheck which was able to find more warnings and errors within our code than the IDE Visual Studio. In our case Visual Studio recognized two warnings while Cppcheck was able to recognize three errors and seven warnings in our code. The use of different forms of testing will help us identify mistakes and issues in our code that may not be found by other test or programs.