Dominic Drury

06/10/2024

CS-405

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

One of the most important best practices when it comes to coding software is to work security into every measure of your program. You cannot leave security to the end of a project for a multitude of reasons. Taking security into account throughout the project helps prevent threats from being forgotten, reduce bugs from unlikely user input, and it saves time and money from being spent pouring over a project to try and cover all the aspects of security. Protecting yourself from all the different kinds of threats targeting software and databases is a full-time task that starts at the beginning of the project and goes until the software is no longer supported and is shut down.

There are a lot of different ways you can help protect yourself from threats. Staying up to date on the third-party libraries you use and ensuring that they practice safe coding, limiting the scope of your variables so they are harder to attack or alter, ensuring input validation is key to making sure a user doesn’t overflow the system or input the wrong type of code, limiting access to users and validating their credentials when giving access to more powerful areas of the program (like the CRUD methods and access level giving), and protecting your code from automated actions with Captchya style checks or other kind of bot prevention.

I think that one of the best examples of security that can be easy to succeed in if you perform it as you go, but very challenging if you wait until the end is user input validation. In a large project there can be a whole host of data that is received, altered, passed through functions, returned etc. If you validate the data as it moves through the program, you can protect yourself from countless types of attacks and at the same time reduce bugs and glitches in your program.