Daniel Anderson

Secure Coding

12/14/2023

**Journal: Reflection**

When starting any development cycle there are many things to consider that need to be considered before the project can start. There are meetings with the clients that need to happen, planning on how the development cycle is going to look alone with the timeframe that must be created. One thing that is should also be discussed at the beginning is when security should be implemented into the development cycle. Merging security into your development cycle is best from the very start of the cycle to ensure that your application will be secure when the cycle is complete. Maintaining a high level of security and coding standards will add some time to the development cycle but will create less when the project is completed and have less upkeep when it is released. Coding standards are policies that should be maintained and upheld throughout every part of the development process to ensure your code is not only clean but manageable in the long run of the project. With every project there is a risk/cost benefit that must be taken into consideration to ensure that everything with your project gets done correctly and properly during the project’s lifecycle. Taking risks in the beginning of the project without implementing security will have a low cost in the beginning but very high costs at the end when you go into postproduction and keep the project up to date. This low-cost high-risk approach is good if the project needs to be done quickly but will have adverse effects when the project needs to be maintained after completion. The other side of the coin is the high-cost low-risk approach which can increase the development time but allows your team to put security in on the ground floor, which lowers the project upkeep after it has been completed. Both sides have their positives and negatives that need to be considered before starting the project. When discussing security there are different levels you can merge into your project but for the best type of security you want to implement a zero-trust policy into the life cycle. A zero-trust policy requires triple-A treatment (authorization, authentication, and accounting) throughout every stage and aspect of your development. Zero-trust is the ultimate shield because it requires constant verification at every point of entry and follows the highest security policies such as two-way authentication and privilege of least access. This brings us to security policies and which ones should be followed throughout the project development process. Security policies should not change from project to project, even though some of them may seem less important depending on the project you should always follow every security policy to its core and merge them into every project that you are working on or involved in. Security policies do not just apply to the development cycle but also to the employees that are working on said projects. Securing not only your project but your employees should always be handled swiftly and at the highest standards to ensure there is as little outside errors as possible to keep not only your application safe but also the clients and users that will eventually be utilizing your application.