Matthew Trembley

12/4/2022

Southern New Hampshire University

CS-405

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

As a best practice in secure coding, “Don’t leave security to the end” means that secure code should be at the start of the software development lifecycle. It should be considered with system and business requirements, and not an afterthought. Automated testing should be done as the project continues, and not to the very end where bugs and flaws may be more expensive to fix. To prevent threats, developers working on a project can include industry standard practices of secure coding. Also, they can test their code not only for extreme edge cases, but many various, normal, use cases that a normal user may stumble upon. Developers can continue to use unit testing like Google unit tests. Another step that can be taken is static code analysis, where a developer should have the compiler on the highest level of warning sensitivity. Also, this can be paired with a second static analysis tool like CppCheck for C++ programs. There are many tools at a developer’s disposable to help them create strong, secure code. One way to keep security as a natural habit for developers is to create a policy *requirement* of using an external static code analysis tool. This will eventually become a habit for a developer to use and will keep any issues at bay, as they will be caught frequently during development rather than the end of the development. By keeping security at the forefront of the mind, it will become natural for a developer to start thinking securely, thus producing more secure code. Constant, in-depth static analysis and unit testing will ensure that the project is as secure as possible.