**Journal: Don’t Leave Security to the End**

Justin Starr

Department of STEM

CS 405 – Secure Coding

Professor Mike Alesso

August 11, 2024

**Journal: Don’t Leave Security to the End**

As a best practice in secure coding, the statement: “Don’t leave security to the end” means that security should be taken into consideration and implemented as early as possible in the Software Development Life Cycle (SDLC). Unfortunately, in many cases in an SDLC, security is not thought about until the end of a project or SDLC, or even worse not until a project has been deployed and a vulnerability is discovered, which can oftentimes have costly side effects, such as the amount of time to remediate the issue. Sometimes the issue can prove to be so large that an entire project could end up becoming scrapped because of the cost associated with fixing the problem, especially in cases where a project does not have sufficient resources (monetarily or otherwise) to move forward with the project.

Steps that can be taken to prevent threats include adopting a DevSecOps pipeline for incorporating security practices all throughout the software development lifecycle. As a part of this DevSecOps pipeline, a secure coding policy should be implemented and strictly adhered to and should include following: applying industry standards and best practices and coding using secure coding standards, applying automation throughout the DevSecOps pipeline to help mitigate potential vulnerabilities, including the use of static analysis tools, applying defense-in-depth (layered defense) techniques, following the Triple-A framework, which includes Authentication, Authorization, and Accounting practices, and ensuring that data is encrypted at all stages, in rest, in flight, and while in use.

One example that can be included in the Project Two presentation for how I plan to ensure that security is addressed intrinsically and not left until an issue is discovered is to follow industry standards and best practices and keep security in mind while coding. This means that when I code an application I can ensure that I am addressing security concerns by doing things such as ensuring I validate user input to make sure that the input I am receiving is what is expected and anything else is handled appropriately. Recognizing when flaws in my code can lead to potential vulnerabilities is essential, and using static analysis tools, writing tests in my code, and handling errors will aid in my ability to ensure security is being addressed throughout a project.