**Matt Knutson**

**CS-305: Software Security**

**SNHU - Instructor North**

**April 19th, 2024**

**Week 8 - Portfolio Submission**

Artemis Financial is a large consulting firm that handles the financial portfolio of their customers. These portfolios include extremely sensitive customer data, which consists of their savings and retirement accounts, their personal investments, and their insurance policies. With that in mind, they are looking to upgrade and revamp their system in order to offer top notch security and a modern user experience. Specifically, their website and software need protocols for secure communication and API assembly, data (at rest and in transit) encryption and integrity, input validation, and user authentication and authorization.

For this project, I felt I was very thorough in identifying vulnerabilities and areas in the code that were lacking in security. It is extremely beneficial to implement security measures throughout the entire development process, starting with the first function created and ending with the last. This will ensure that input validation and error handling are integrated throughout the entire system. Likewise, it is highly probable that a multitude of security issues and vulnerabilities will go unnoticed if security is not dealt with until the very end of building a potentially massive software application. With the size of modern-day software, the amount of user traffic, and the intricate process of connecting a plethora of devices, it is extremely important for software to be secure in order for a company to be successful.

Overall, I had a good experience on my journey through the vulnerability assessment. I found generating a certificate to be an interesting exercise, but never could quite get my hashed data to connect to port 8843. I found the Dependency-Check to be an awesome new tool. At first the information concerning vulnerable dependencies was a bit overwhelming. But, after a few weeks of using the Dependency-Check, and studying the information, it became a valuable asset.

In the future, I will definitely use the OWASP Dependency-Check to scan for vulnerabilities in my software and the plugins I use. I will also use my knowledge of generating self-signed certificates and using cryptographic algorithms in future security related classes. The OWASP Dependency-Check can be run multiple times throughout the development life cycle to continually check for new vulnerabilities introduced in the system. By simply creating a suppression.xml file and adding a few lines of code, the false positives can be suppressed and hidden from the current list of vulnerable dependencies.

Although I was successful when running the dependency-check and generating a certificate, I could not get the internal SSL server to connect to the web browser. I tried refactoring the code multiple times and completely starting over a couple of times as well but couldn’t get it to run without errors. I documented the code in depth using comments to try and trace any mistakes in syntax or logic, to no avail. But I am still walking away with the knowledge of how to generate hashed data, use different algorithms for encryption, create a self-signed certificate, use a Certificate Authority, and run an OWASP Dependency-Check.