Briefly summarize your client, Artemis Financial, and their software requirements. Who was the client? What issue did they want you to address?

* Artemis Financial is a hypothetical company for the sake of immersion. They are a multinational corporation that deals with government and overseas operations. Their area of business encompasses portfolio management, savings, and retirement accounts.

Why is it important to code securely? What value does software security add to a company’s overall well-being?

* Writing secure code saves time and money. There is a trickle-down effect when had code is implemented. It starts customer trust which is incalculable to a company’s wellbeing. The monetary losses from data breaches are only getting higher as the EU shapes cyber law and one of these suits could bring down a company. Even if it is caught without outward-facing damage there is still the process of fixing issues which is also time intensive. Debugging, ticketing, assignment, and refactoring all cost a great amount of time, money, and energy.

What part of the vulnerability assessment was challenging or helpful to you?

* The most beneficial knowledge I have gained from this project is the differences between hashing, symmetric and asymmetric encryption. Before I thought hashing was salting or padding and asymmetric encryption was hashing. Now I understand that hashing irreversible changes data into a fixed-size block of text. This can be used to store passwords or check the integrity of downloads. This helped un-muddle the concept of asymmetric encryption having two keys and the fact that symmetric only has one.

How did you increase layers of security? In the future, what would you use to assess vulnerabilities and decide which mitigation techniques to use?

* Because Artemis Financial has valuable personal information, I decided to use RSA-4096 which is a bit slower than RSA-2048, but since it is a Financial company I think it is worthy of having a slightly slower and more secure cipher. I paired that with a SHA-256 hashing algorithm which currently has now publicly known vulnerabilities.

How did you make certain the code and software application were functional and secure? After refactoring the code, how did you check to see whether you introduced new vulnerabilities?

* Using best practices, I accounted for injections which is one of the most common attacks. I also reran the dependency check with OWASP. Beyond that, I manually went and checked the domain for http and https. I also ran the hashed data through Kleopatra which reaffirmed the hashing was working correctly.

What resources, tools, or coding practices did you use that might be helpful in future assignments or tasks?

* I used several tools like Java.security module, Javas’ Keytool.exe, Kleopatra, and a Maven OWASP plugin. The algorithms I used were RSA-4096 and SHA-256. I also used fundamental concepts like Least Privileges, the 7 levels of cybersecurity, CIA, and best coding practices to spot areas of concern in the code.

Employers sometimes ask for examples of work that you have completed to show your skills, knowledge, and experience. What might you show future employers from this assignment?

* This project provides a glimpse of my skills, knowledge, and experience by demonstrating the following. My knowledge of secure communication. The technical skill to be able to implement encryption and hashing algorithms to secure communications and validate their integrity. It also shows my experience in the CI/CD life cycle from writing code to debugging and even offering support in previous roles