****

# Practices for Secure Software Report

Table of Contents

[Document Revision History 3](#_Toc102040754)

[Client 3](#_Toc102040755)

[Instructions 3](#_Toc102040756)

[Developer 4](#_Toc102040757)

[1. Algorithm Cipher 4](#_Toc102040758)

[2. Certificate Generation 4](#_Toc102040759)

[3. Deploy Cipher 4](#_Toc102040760)

[4. Secure Communications 4](#_Toc102040761)

[5. Secondary Testing 4](#_Toc102040762)

[6. Functional Testing 4](#_Toc102040763)

[7. Summary 4](#_Toc102040764)

[8. Industry Standard Best Practices 4](#_Toc102040765)

## Document Revision History

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | **4/26/2025** | **Matthew Bridegroom** | **Revision One** |

## Client



## Instructions

Submit this completed practices for secure software report. Replace the bracketed text with the relevant information. You must document your process for writing secure communications and refactoring code that complies with software security testing protocols.

* Respond to the steps outlined below and include your findings.
* Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project Two Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

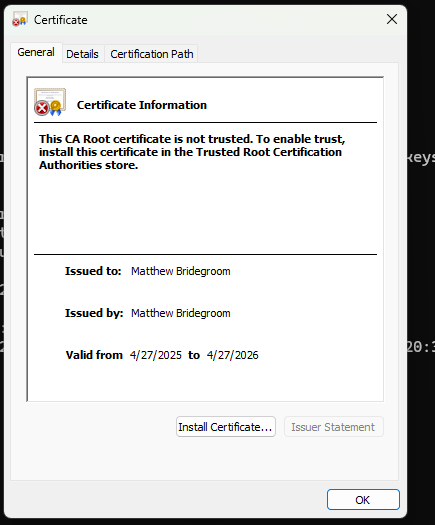
Matthew Bridegroom

## Algorithm Cipher

When implementing an algorithm cipher for Artemis Financial, I chose to pursue the encryption algorithm of SHA-256. SHA-256 is a one-way hashing algorithm that is widely used in order to make sure the integrity of the data is secure. The algorithm works by taking any input and creating a fixed-sized hash value from the data. The way that a hash function such as SHA-256 works is by taking any data input and turning it into a hash value of fixed length. The length of the hash is determined by the bit level of the hash. In this case, a 256 bit has a length of 32 bytes. This helps to ensure that there is a high degree of security.

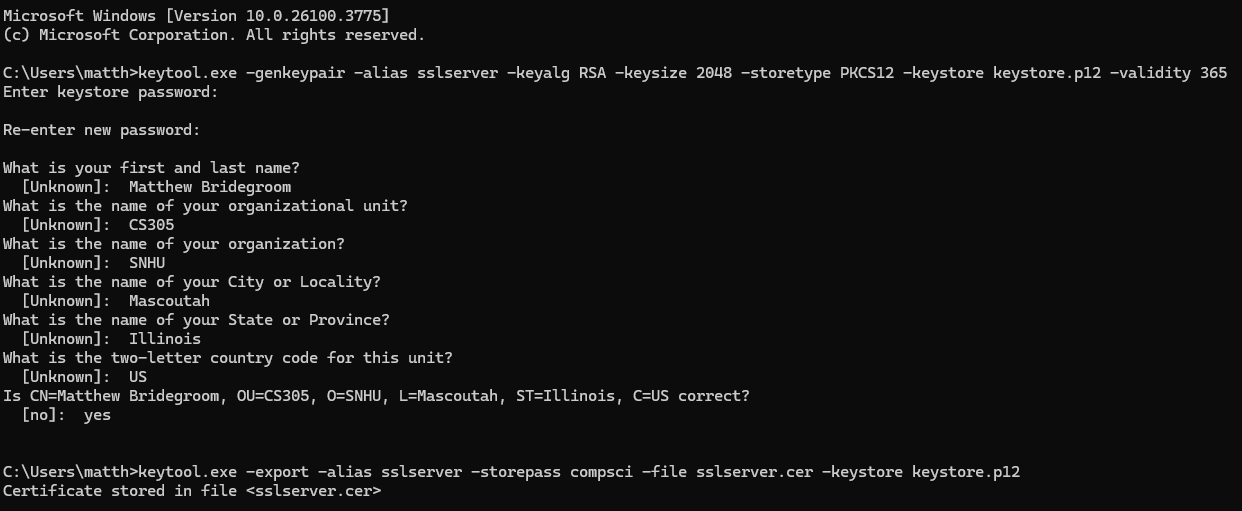
Looking at a separate algorithm such as AES or Advanced Encryption Standard, it is a symmetric encryption algorithm, which means there is a single shared key that is used for both encrypting and decrypting. In doing this, it is important that random numbers are used for generating the keys so that they are secure. However, in SHA-256, no keys are needed as it is a hashing algorithm, therefore it will always generate the same hash for a certain input. This helps to ensure the integrity of the input. SHA-256 was first used back in 2001 and was designed to replace the older SHA-1 algorithm. This was because there had been vulnerabilities found when performing collision attacks. Still to this day, SHA-256 is still being used across many different industries and is recognized by the National Institute of Standards and Technologies as it meets the different standards needed.

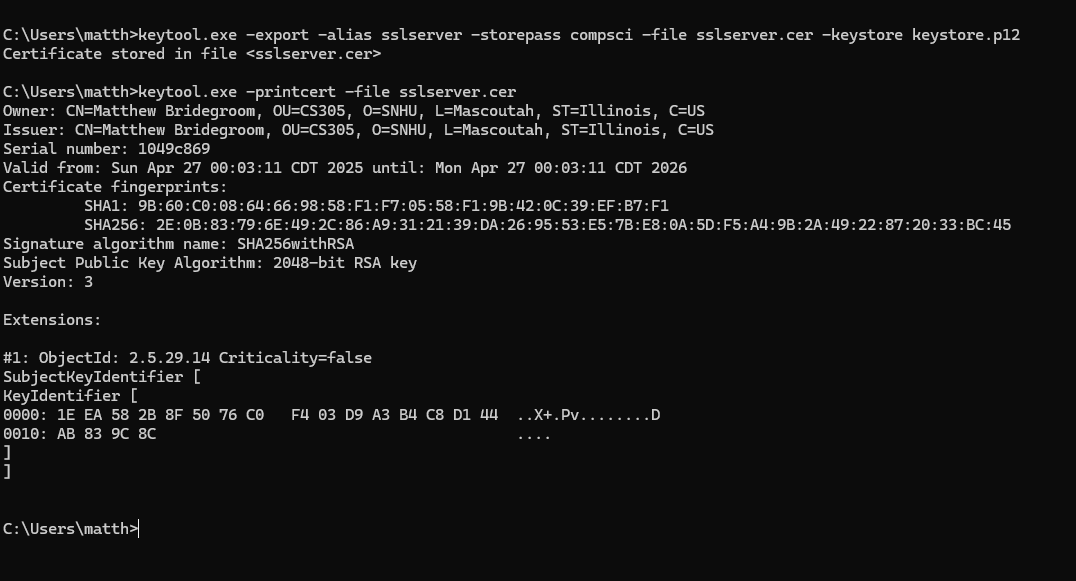
## Certificate Generation



## Deploy Cipher

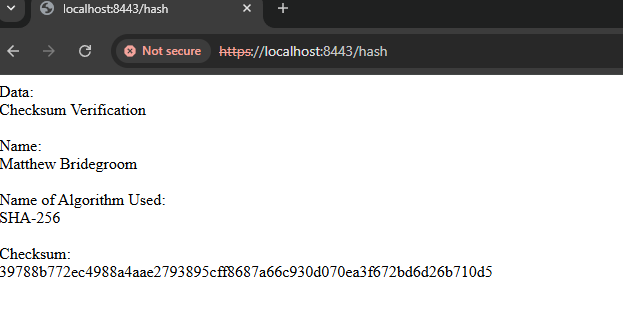
Insert a screenshot below of the checksum verification.





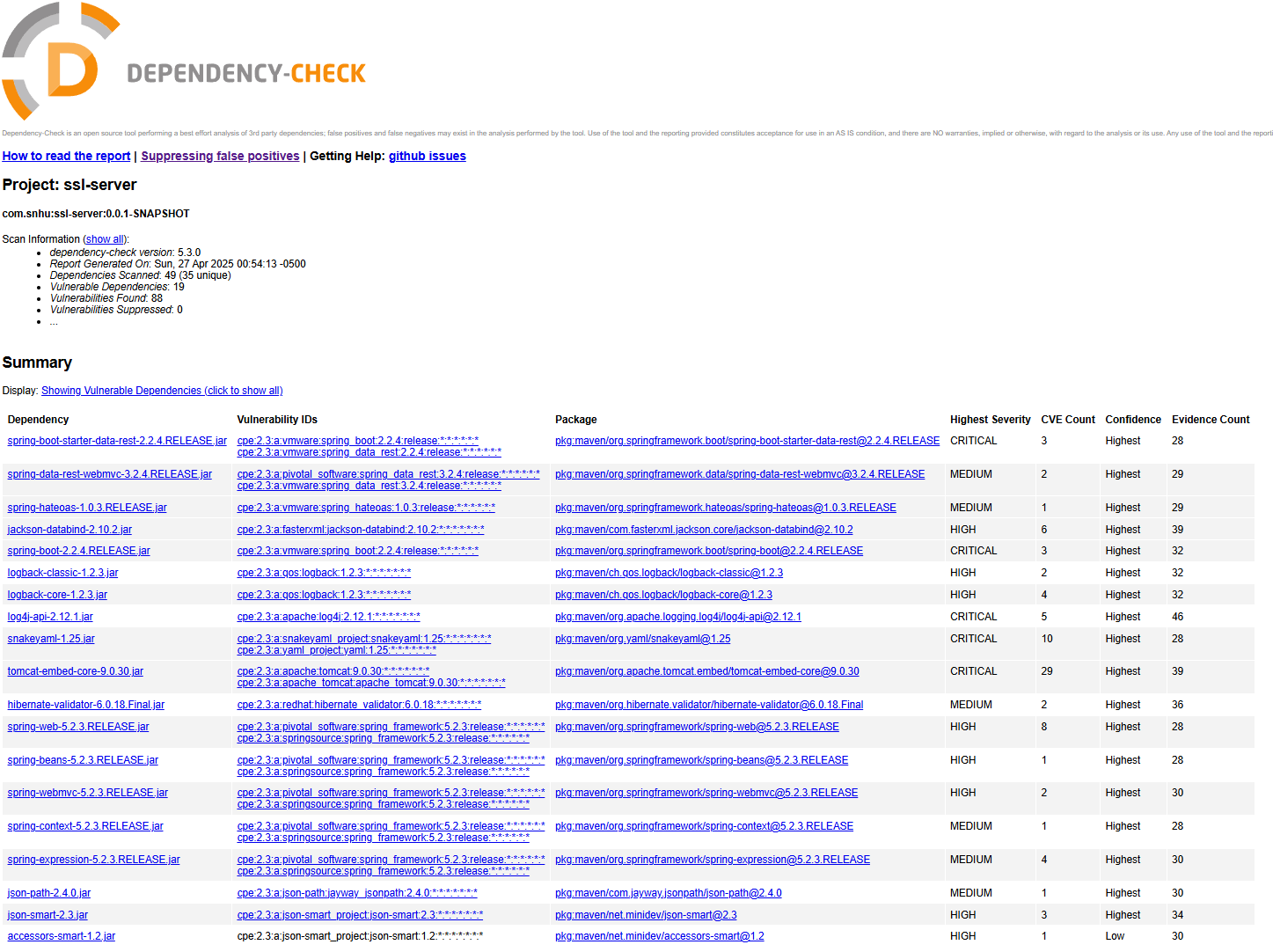
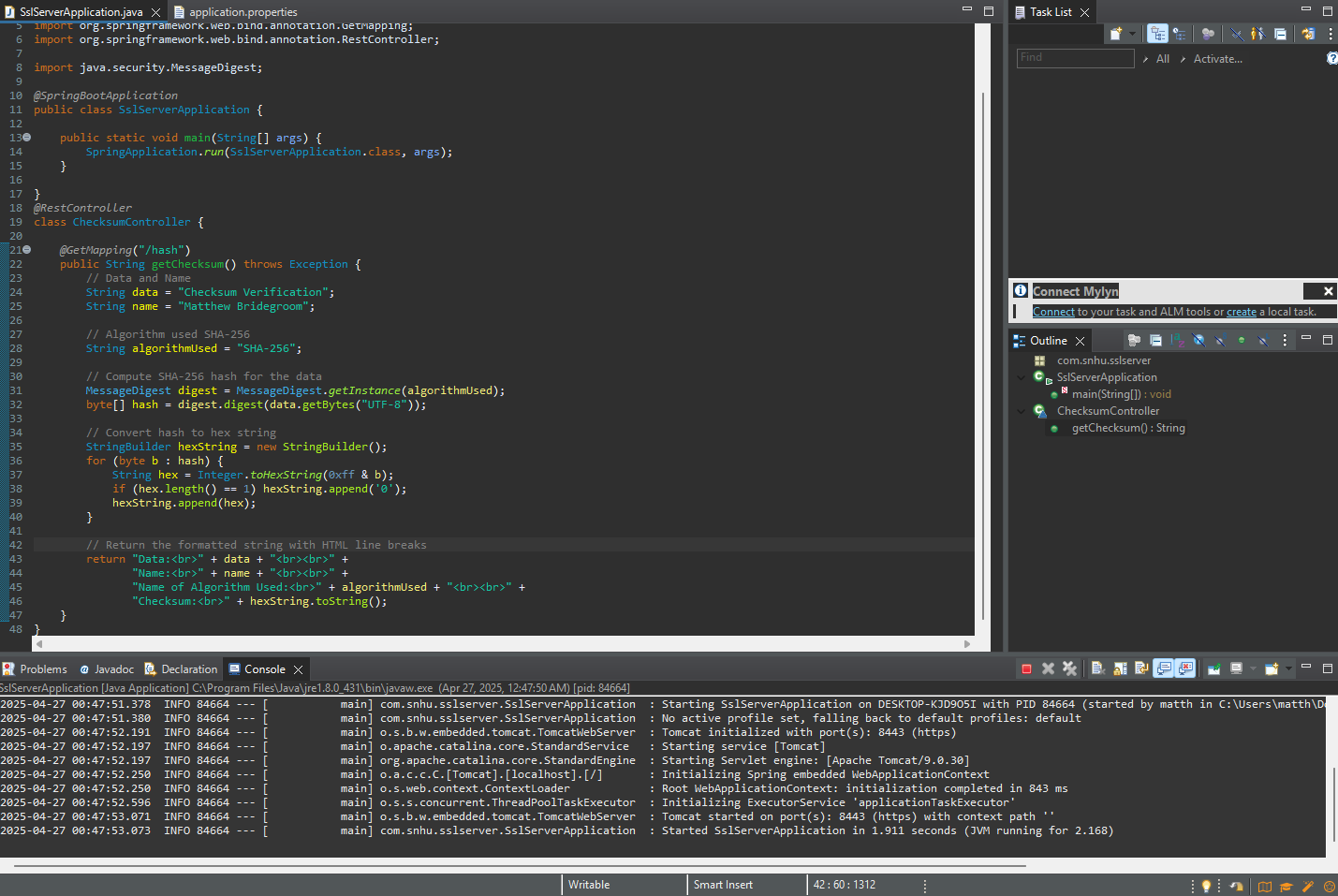
## Secure Communications

Insert a screenshot below of the web browser that shows a secure webpage.



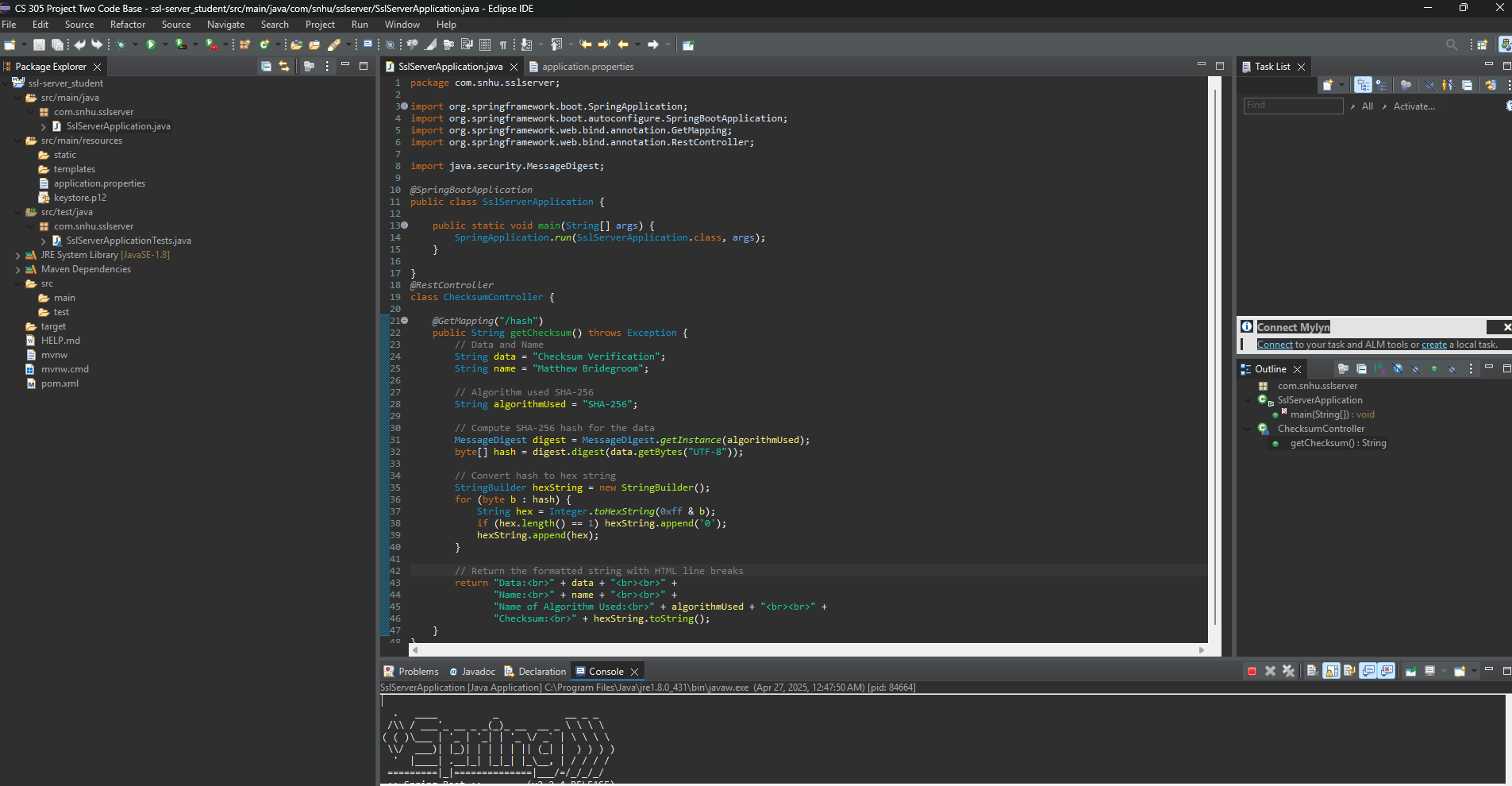
## Secondary Testing

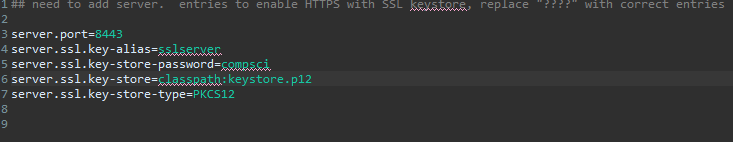
Insert screenshots below of the refactored code executed without errors and the dependency-check report.



## Functional Testing

Insert a screenshot below of the refactored code executed without errors.





## Summary

When looking back on the vulnerability assessment diagram, there were two main ways that I addressed security when refactoring the code. The first way was to run the OWASP Dependency check. This was done in order to ensure that once the code had been written, no new vulnerabilities were placed into the code itself. The second way that I tackled security was by implementing SHA-256 for the encryption method. In doing this, I created a new security layer as SHA-256 helps to ensure data integrity. A second security layer was the use of HTTPS for more secure communications as it encrypts the data that is being transmitted. The last security layer was again the use of the dependency check before and after the code was written in order to ensure that no new vulnerabilities were added.

## Industry Standard Best Practices

I did my best to use industry standard best practices when working on refactoring the code. One area of this was the choice of SHA-256 for the encryption method. This is because it is in line with the industry standard practices as it helps to ensure the data’s integrity and uses a secure hashing algorithm. Another area where I applied these practices was with the use of HTTPS encryption in order to make sure that any data being transmitted is secure. A third area was the use of Dependency Checks. Running an OWASP Dependency check before and after the code was written helped me to make sure that not only were no new vulnerabilities introduced, but that the software's existing security was maintained.

The value of applying the different industry standard best practices is not only to mitigate the different security risks, but it is used to also protect the customer data and build trust in the system. Without these practices, vulnerabilities could pop up all over the system and this would result in a major loss of trust from the customer, especially in the financial sector, and this could lead to major problems for any business down the road.