****

# Practices for Secure Software Report

**Table of Contents**

[**Document Revision History 3**](#_heading=h.30j0zll)

[**Client 3**](#_heading=h.1fob9te)

[**Instructions 3**](#_heading=h.3znysh7)

[**Developer 4**](#_heading=h.2et92p0)

[**1. Algorithm Cipher 4**](#_heading=h.tyjcwt)

[**2. Certificate Generation 4**](#_heading=h.3dy6vkm)

[**3. Deploy Cipher 4**](#_heading=h.1t3h5sf)

[**4. Secure Communications 4**](#_heading=h.4d34og8)

[**5. Secondary Testing 4**](#_heading=h.2s8eyo1)

[**6. Functional Testing 4**](#_heading=h.17dp8vu)

[**7. Summary 4**](#_heading=h.3rdcrjn)

[**8. Industry Standard Best Practices 4**](#_heading=h.26in1rg)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **[Date]** | **[Your Name]** |  |

## 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

Zachery Irvin

## Algorithm Cipher

### **Recommended Cipher: AES-256**

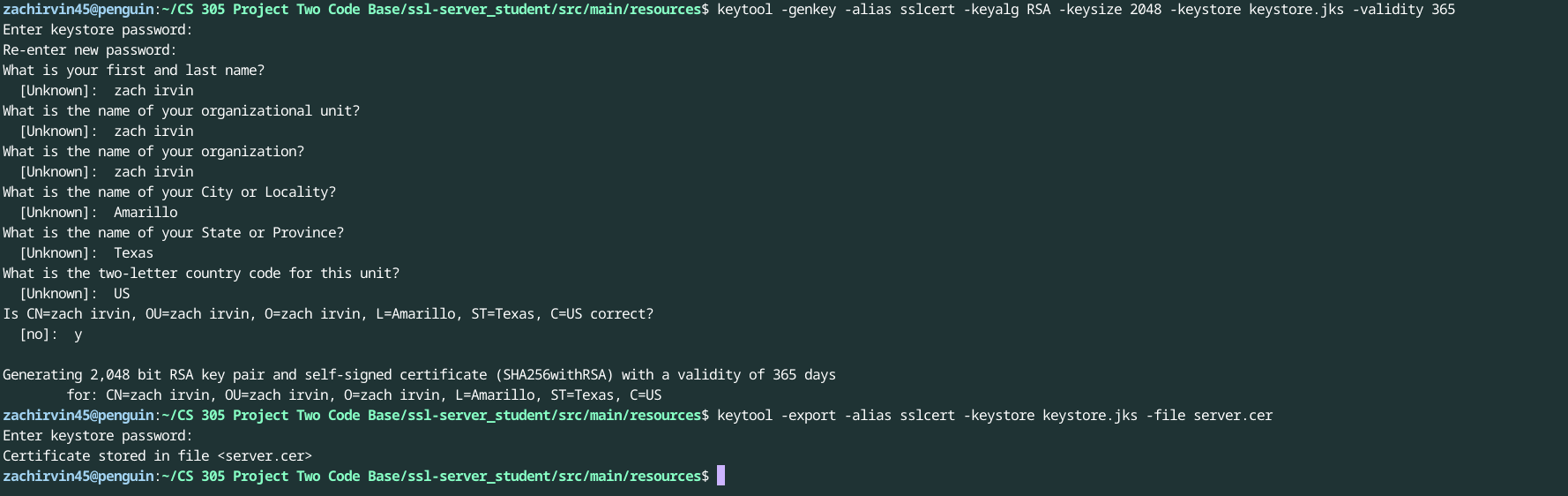
My recommendation for the cipher is the AES-256. It is widely regarded as the de-facto for security when it comes to symmetric encryption. It provides a high level of security due to the 256 bit key length. This makes it very resilient to brute force attacks. It has been optimized extensively making its performance ideal. As an added bonus, it is widely accepted and used in secure communications.

The bit levels are directly correlated to the length of the has and as such affect the overall security of the hash. The AES has a bit length of 256 which produces a 256 bit output regardless of the input.

There are two different methods for encrypted data. One is symmetric which uses a single key for encryption and decryption, and the other is asymmetric that utilizes a public and private key. AES-256 uses symmetric as it is faster and more efficient when still maintaining industry standard security.

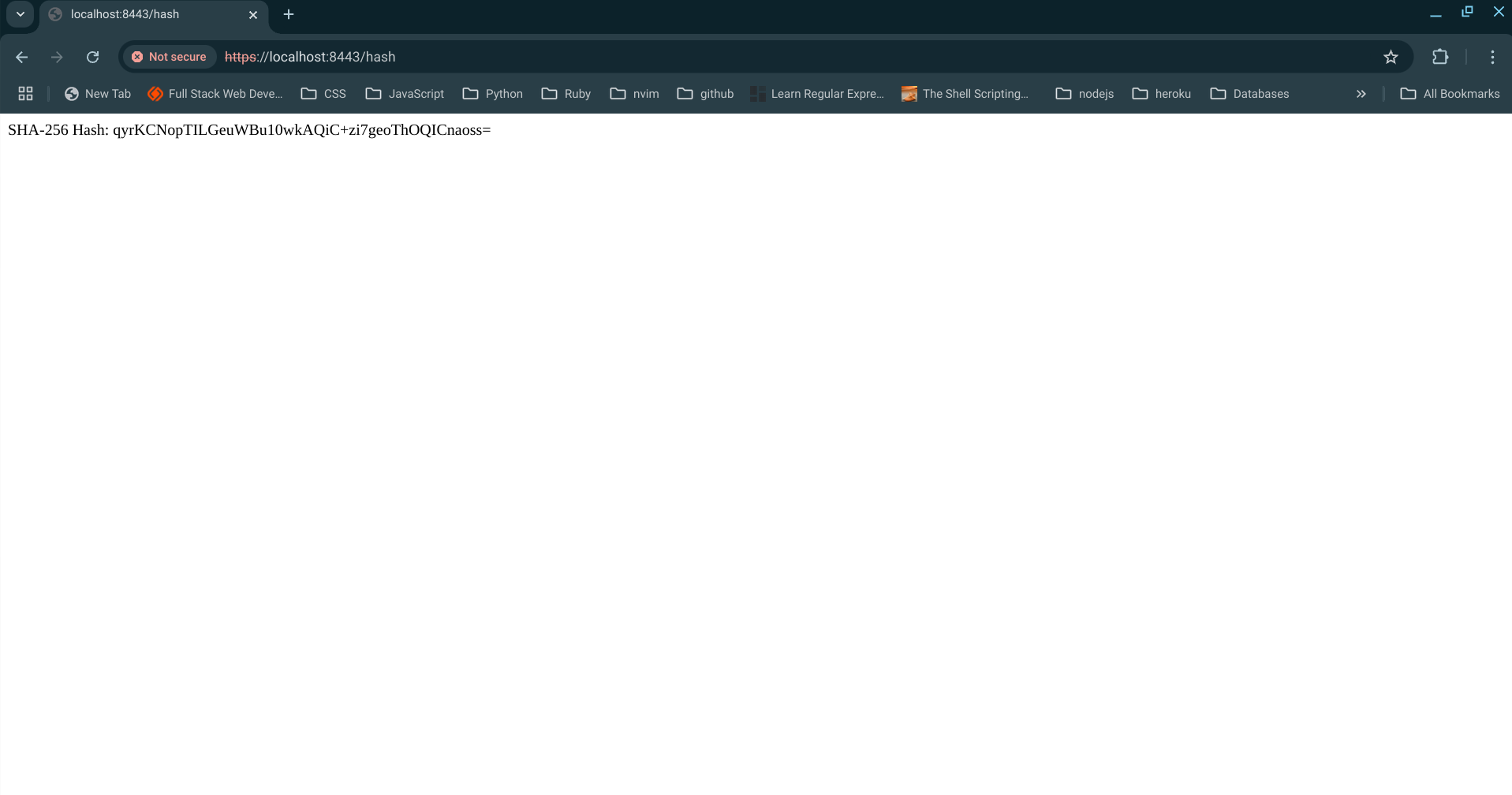
## Certificate Generation

Insert a screenshot below of the CER file.



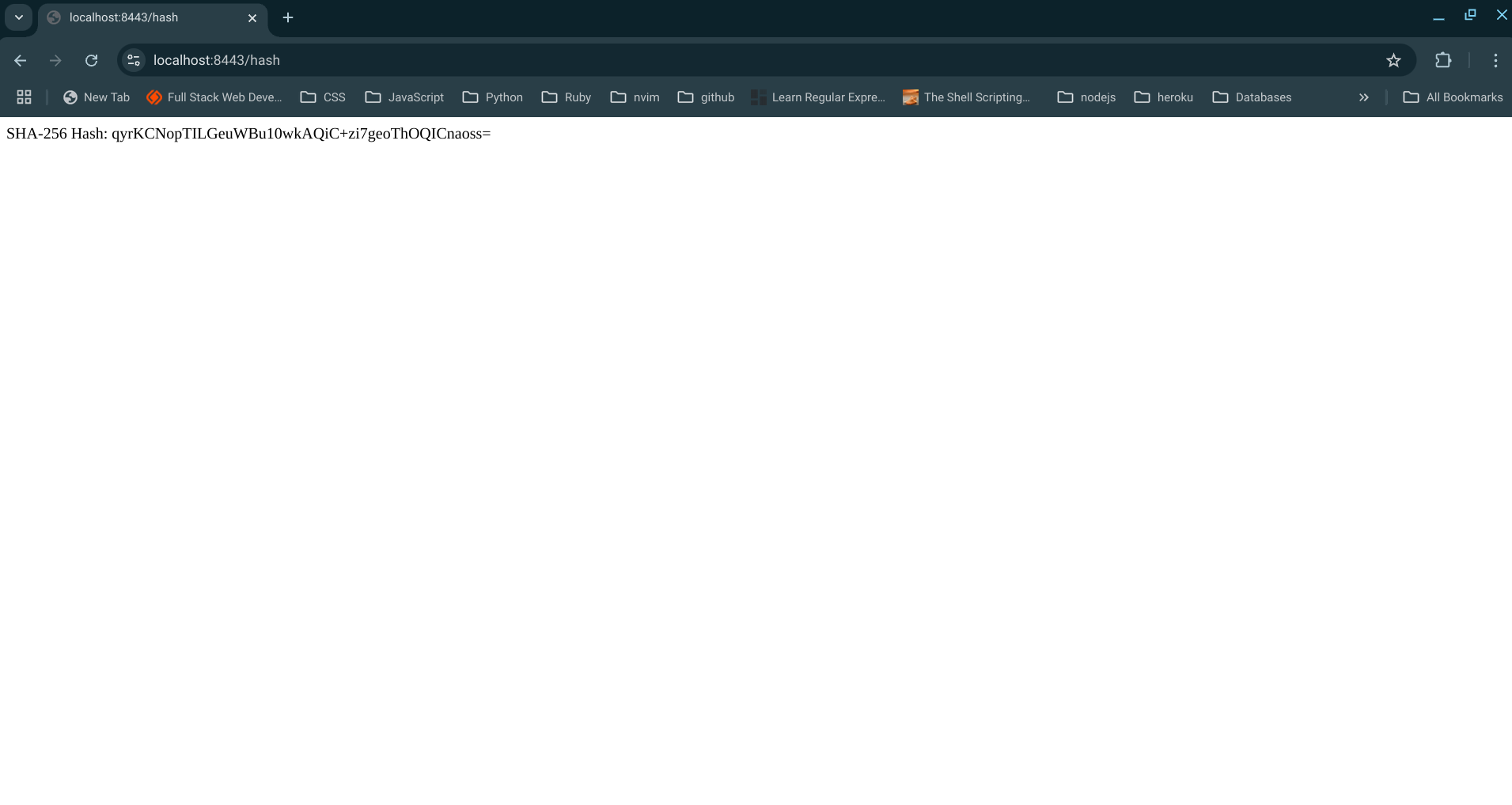
## Deploy Cipher

Insert a screenshot below of the checksum verification.



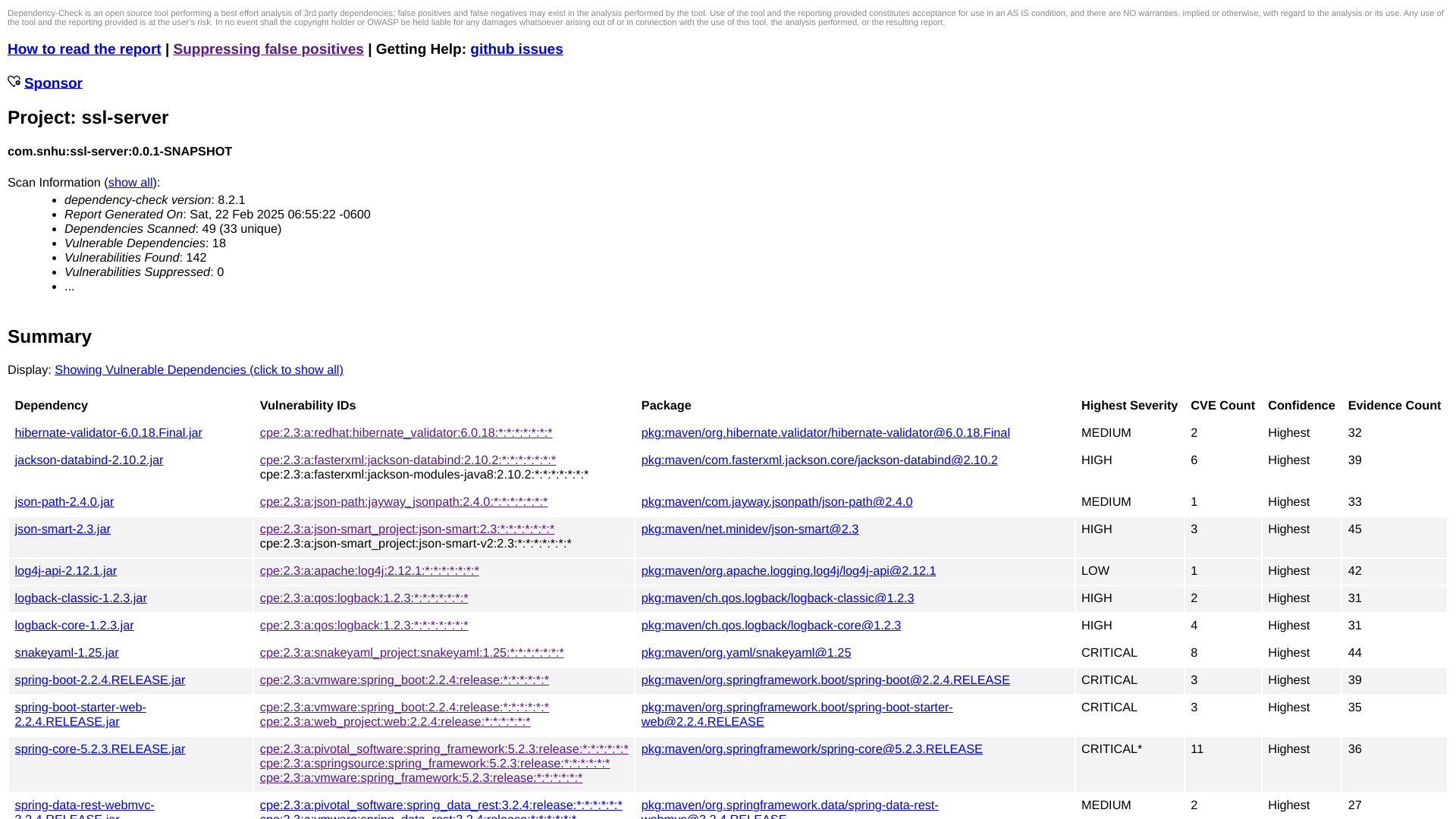
## Secure Communications

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

In summary the following security enhancements were added to the existing code base. First the AES-256 encryption was selected for the protection of the data. Second, the SHA-256 was implemented for verification of the data and to maintain its integrity. Third, for implementation of secure communications, HTTPS protocol was used. Lastly the use of the OWASP dependency check was used for static security analysis. These four different practices make up the layers of security for this particular application.

## Industry Standard Best Practices

The above layers are also considered industry best practices. The AES-256 provides very strong encryption. This goes hand in hand with the SHA-256 that assures the integrity of the data has not been corrupted. The next layers of HTTPS solidifies secure communication and finally the dependency check tool works well for static analysis of any vulnerabilities. All of these work together to protect sensitive information and conform to industry security standards.