****

# 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** | **8/13/2024** | **Danielle Sousa** |  |

## 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

Danielle Sousa

## Algorithm Cipher

To provide the most current and effective software security, the AES Cipher, also known as Advanced Encryption Standard, is recommended. AES uses a 128-bit block that supports the keys of 128,192 and 256 bits. There are two types of key encryption, Symmetric and Asymmetric. Symmetric key encryption uses the same key to encrypt and decrypt the data. AES is a symmetric key encryption algorithm. Asymmetric encryption uses a pair of keys for encryption and decryption, such as a public key and a private key. Symmetric encryption is considered the most secure and is widely used for securing internet communications, protecting sensitive data, and encrypting files. Using a secure hash algorithm (SHA-256) is a hash function that converts any text length into a fixed string of 256 bits. SHA-256 is made to be a one-way function and is impossible to reverse the process to obtain the original input from the hash. SHA-256 is fit for hashing passwords, messages, and files. It is considered fast and reliable and provides high-level security against attacks. The very first cipher was used circa 600 BC by the Spartans, which would involve wrapping a string of parchment around a rod to reveal the hidden message. Over decades it has evolved into something more advanced. Since the rise of technology and the introduction of computers, encryption algorithms now use advanced mathematical fundamentals that have become critical for protecting sensitive data in various computer applications.

## Certificate Generation

Insert a screenshot below of the CER file.

* Generate certificate and store it in cer file.

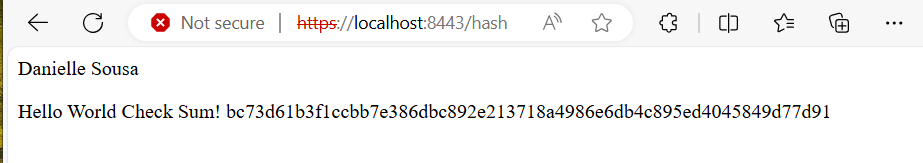
A screenshot of a computer

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

* Deployed and implemented the cryptographic hash algorithm by refactoring the code. Below is an image of the checksum verification.



## Secure Communications

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

* As you can see below, I installed the self-signed certificate into the trusted root once the certificate was generated.

A screenshot of a certificate

Description automatically generated A screenshot of a computer

Description automatically generatedA screenshot of a certificate

Description automatically generatedA screenshot of a computer

Description automatically generated

* Even after installing the certificate in the trusted root, I was unable to get a secure connection.

A screenshot of a computer

Description automatically generated

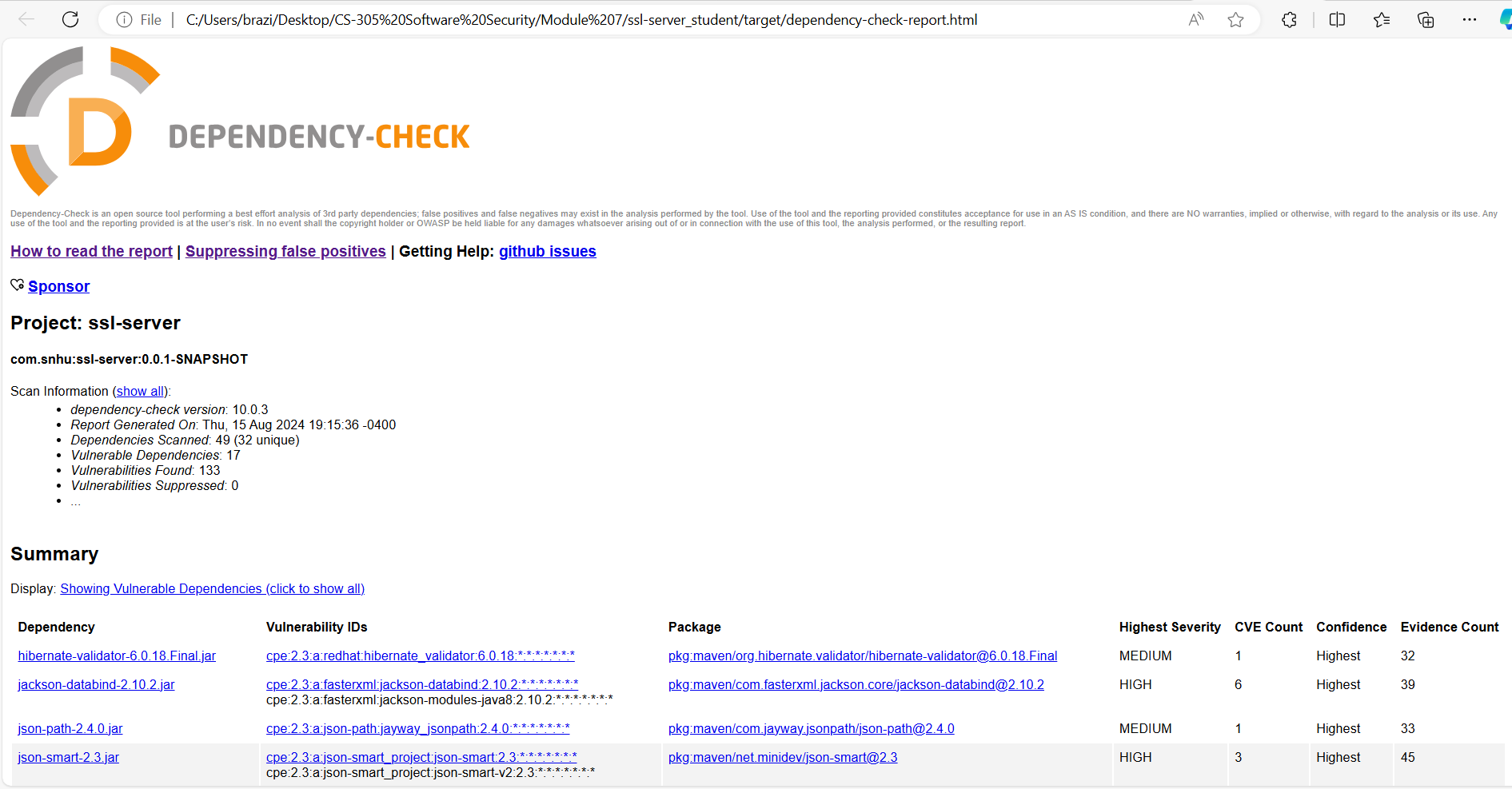
## Secondary Testing

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

* Code is refactored below with no errors and the dependency check report. Successful

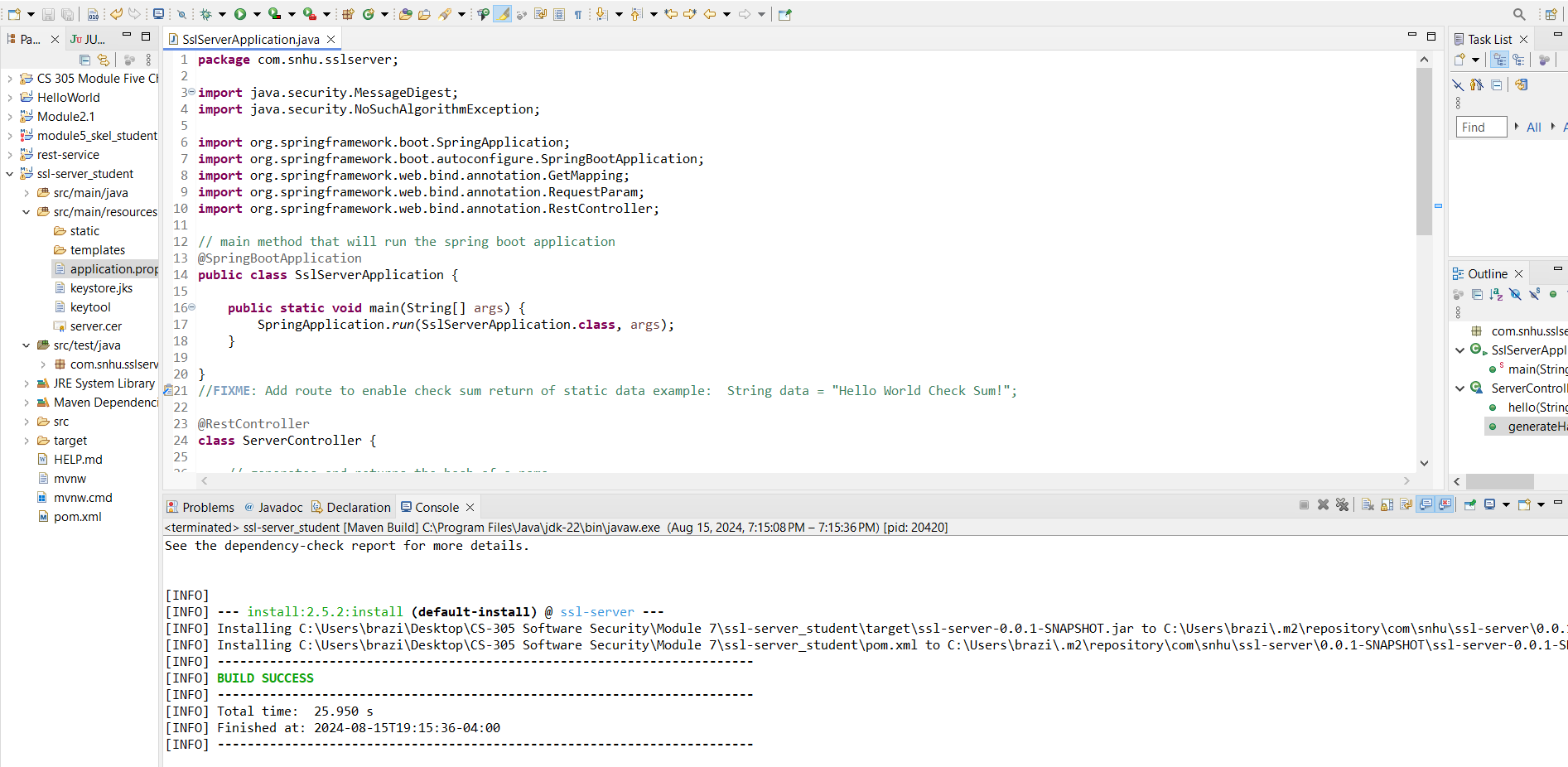
A screenshot of a computer

Description automatically generated





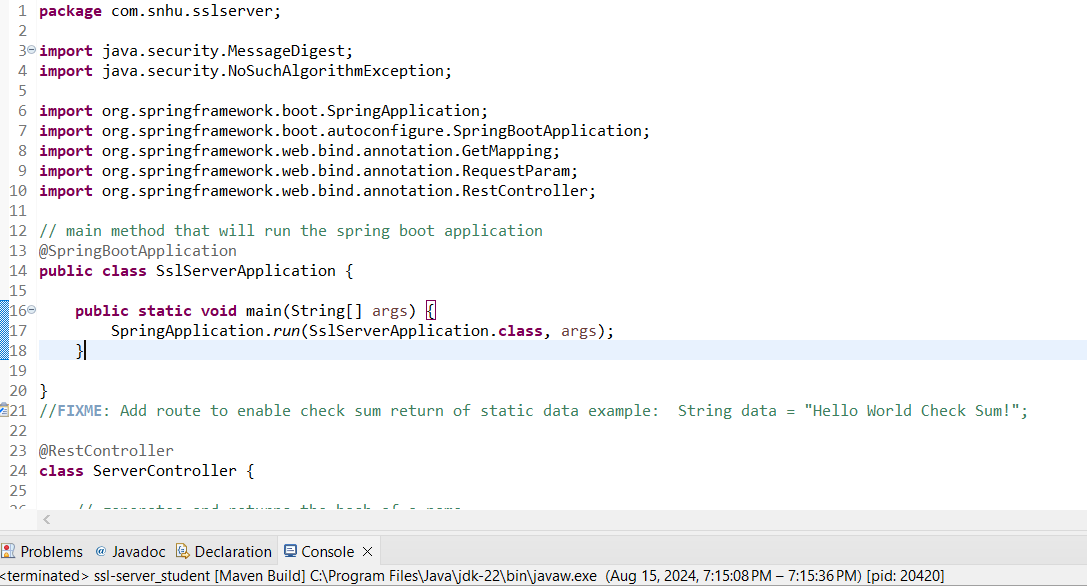
* Maven builds are also successful with no errors.



## Functional Testing

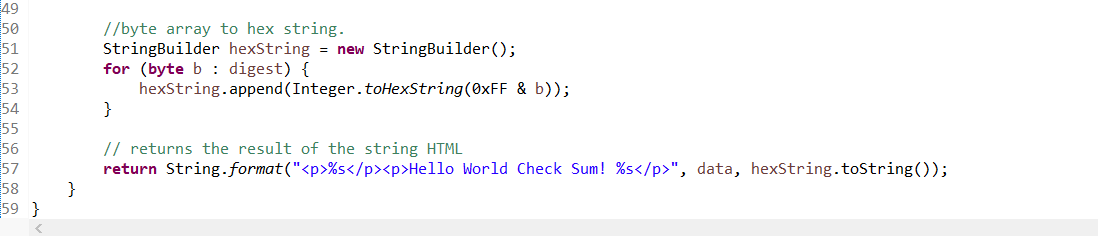
Insert a screenshot below of the refactored code executed without errors. Updated the POM.xml file to the latest version of OWASP dependency check.

* Code refactored and executed without errors.



A screenshot of a computer

Description automatically generated



## Summary

According to the Vulnerability assessment process flow diagram, I could address areas of security by refactoring my code to address areas of security such as cryptography, by implementing a hash function for the checksum verification. Then I address API’s interactions by implementing Rest Controller code built using the Spring framework. Lastly, the client/server was addressed by generating a self-signed certificate. Using the hash function in cryptography applications helps to ensure data integrity and security by making it impossible to reverse the hash function or get two different inputs that generate the same hash value. Through API interactions, HTTPS instead of HTTP is used to send the data between the web browser and the secured website. The self-signed certificate encrypts the data between the server and the client. This helps the customer to verify that the data can be trusted. I generated a check dependency report to make sure there were no other vulnerabilities associated with my code. These security measures help to give the customer peace of mind to know that Artemis Financial is a trusted business.

## Industry Standard Best Practices

To make sure my code adhered to Artemis security requirements, following industry standards best practices such as comments for maintainability. Exception Handling that helps with debugging. Avoids hardcoding sensitive data and avoids redundant code. The value of applying best industry-standard best practices for secure coding helps to reduce vulnerabilities. Keeps the company name and reputation intact. Keeps the company in regulatory compliance, which if violated can be costly to the company. Integrating these best practices helps to maintain the trust with the customer, make sure the company experiences continued success, and protect the company's assets.

**References**

Adetunji, D. (2023, April 5). *Symmetric and Asymmetric Key Encryption – Explained in Plain English*. FreeCodeCamp.org. <https://www.freecodecamp.org/news/encryption-explained-in-plain-english/>

*Encryption: The Past, Present, and Future - AXEL.org*. (2021, May 28). Www.axel.org. https://www.axel.org/2021/05/28/history-of-encryption/

‌

Jena, B. K. (2023, August 29). *What Is SHA-256 Algorithm: How it Works & Applications | Simplilearn*. Simplilearn.com. <https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm>

Website Administrator. (2024, February 19). *The Importance of Secure Coding Standards*. Sec1. https://sec1.io/blog/the-importance-of-secure-coding-standards/

‌

‌