****

# 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** | **2023-16-06** | **Gerardo Gonzalez** | **Initial project.** |

## Client



## Instructions

Submit these completed practices for secure software reports. 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

Gerardo Gonzalez.

## Algorithm Cipher

This paper discusses the recommended cipher algorithm that will best fit Artemis Financial’s needs for a current and efficient method for encrypting their sensitive data. After reviewing the client’s needs, I recommend using the encryption algorithm AES. I think that AES best fulfills the client’s needs for a current and effective cipher algorithm. AES is the current industry standard cipher used for handling sensitive data and reaches a balance for encrypting data in terms of speed and capability, which makes it a great algorithm to use.

AES also known as the Rijndael algorithm is a symmetric block cipher, which means that it produces a single key for both encryption and decryption of the data. The length of the key can one of three sizes 128, 192, and 256 bits. AES encrypts data by blocks and each block uses each of these blocks can then be transmitted and decrypted after transmission.

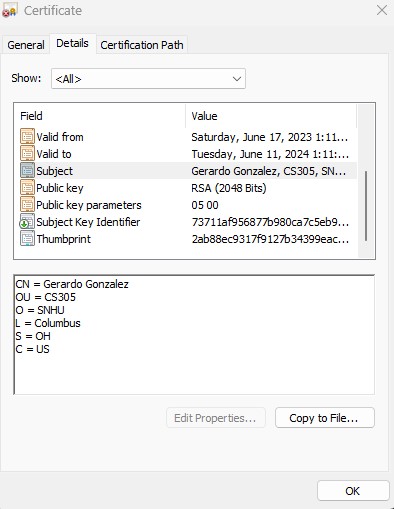
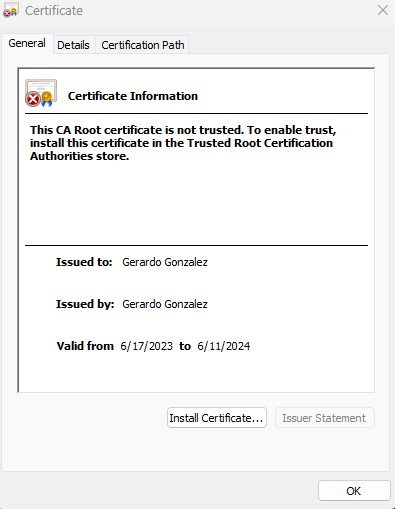
The hash functions are used based on the block ciphers one way that hash functions work is by using it as a method for authenticating the data that is coming in, since hashes are not able to be decrypted, they can be used to authenticate the data. This works by producing a specific hash value and comparing it after transmission and if any part of the hash value is different then that is a sign that the block has been tampered with. In general, the has the number of bits can impact the strength of the encryption, the larger the bit level the more difficult it is to access the data.

Random numbers are used everywhere in cryptography and are used in a variety of ways, one way it is used in AES is by generating keys and bits. AES uses pseudo-random number generations which can help generate a mathematically random number for its encryption and keys. In cryptography, the use of symmetric and non-symmetric keys is used by the type of algorithm used. For example, we discussed how AES uses symmetric keys for both encryption and decryption of data, while non-symmetric or also known as asymmetric keys work by having two separate keys one for encryption and one for decryption. Both have pros and cons when used in an algorithm.

The history of encryption can be seen in hieroglyphics as these create the process of how to transform data in this case visually and let people who understand the language know what it means while preserving the privacy of the hieroglyphics from individuals who don’t know the meaning of the symbols. The current state of encryption algorithms has advanced as the world grows. Today cyber security is one of the most in-demand services and with the world transmitting sensitive data every second it’s important to keep this data private on a need-to-know basis which has developed a user authentication process to validate that the person who has the data is an authorized individual. Also as mentioned earlier AES is one of the most used algorithms and is the industry standard.

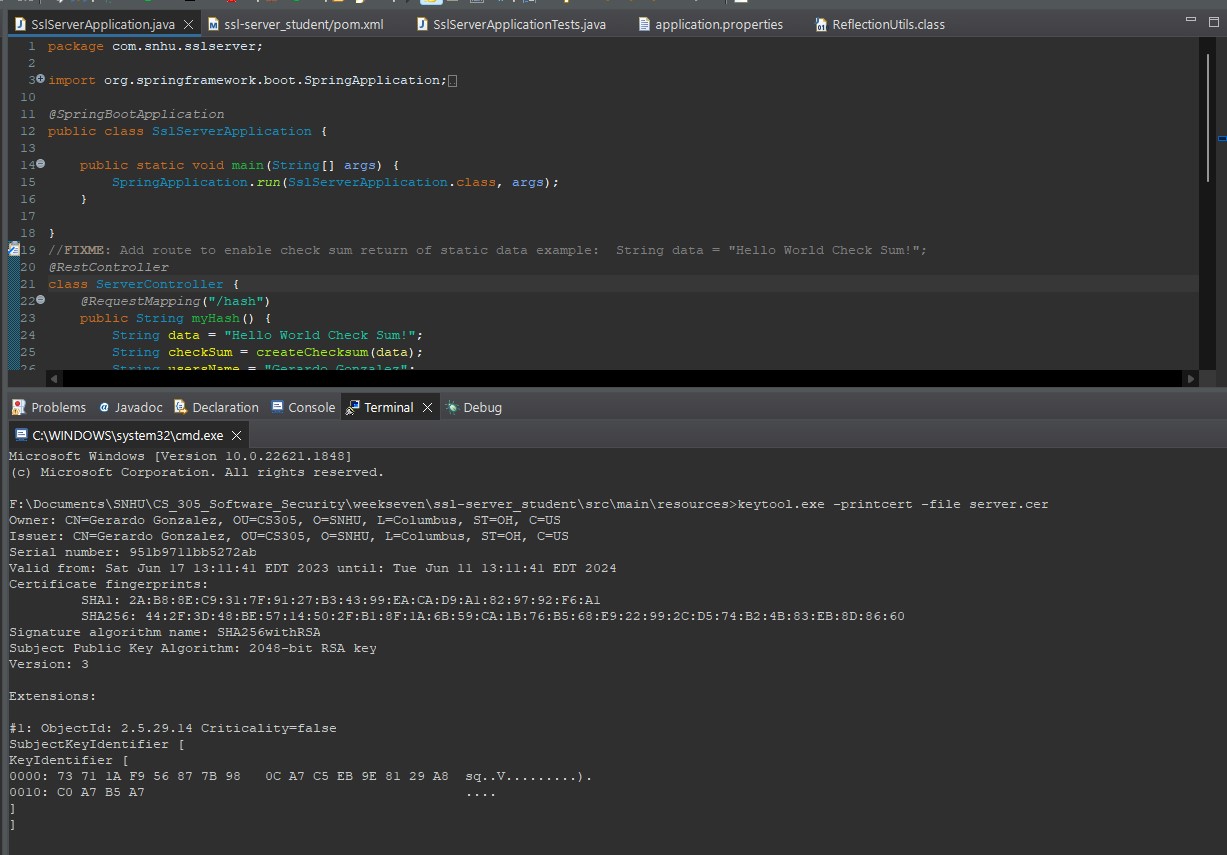
## Certificate Generation

A screenshot below of the CER file.



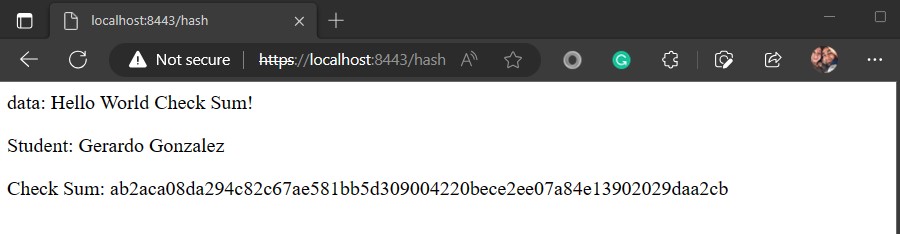
## Deploy Cipher

A screenshot below of the checksum verification.



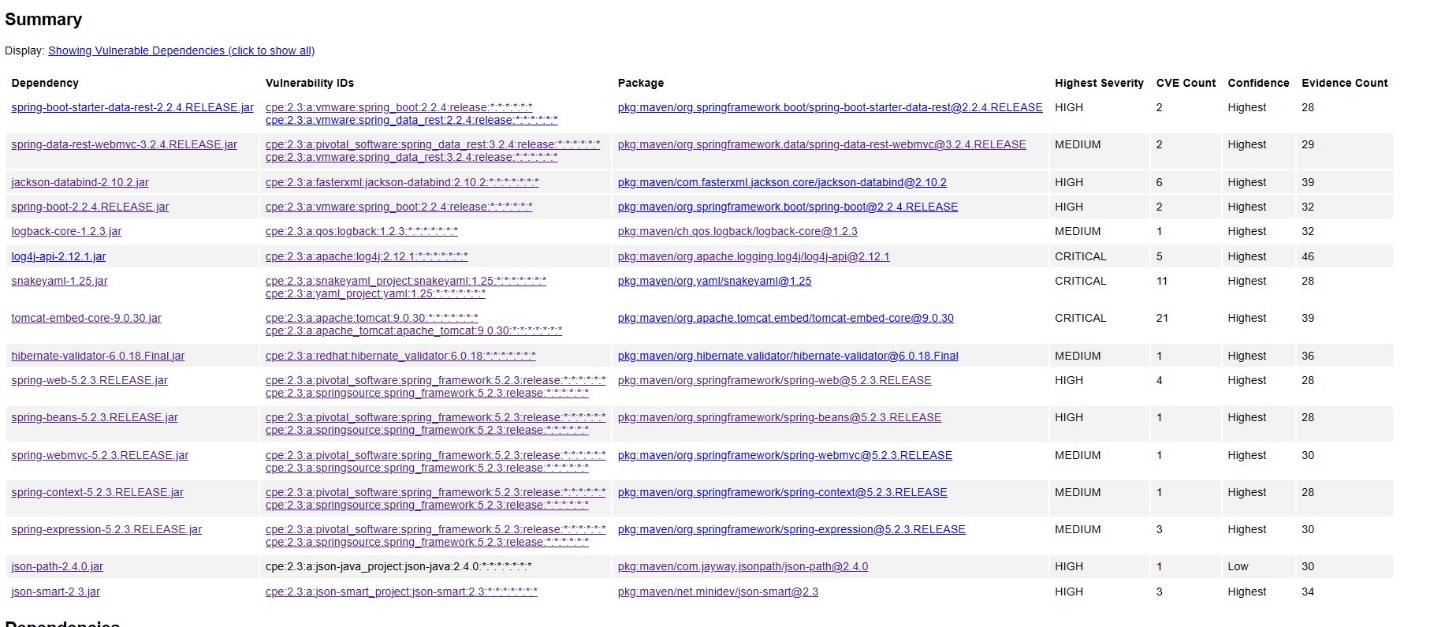
## Secure Communications

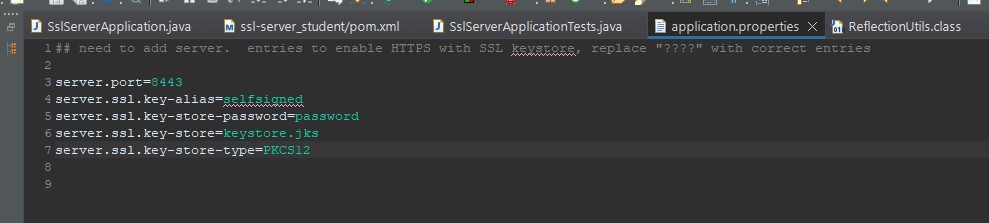
A screenshot below of the web browser that shows a secure webpage.

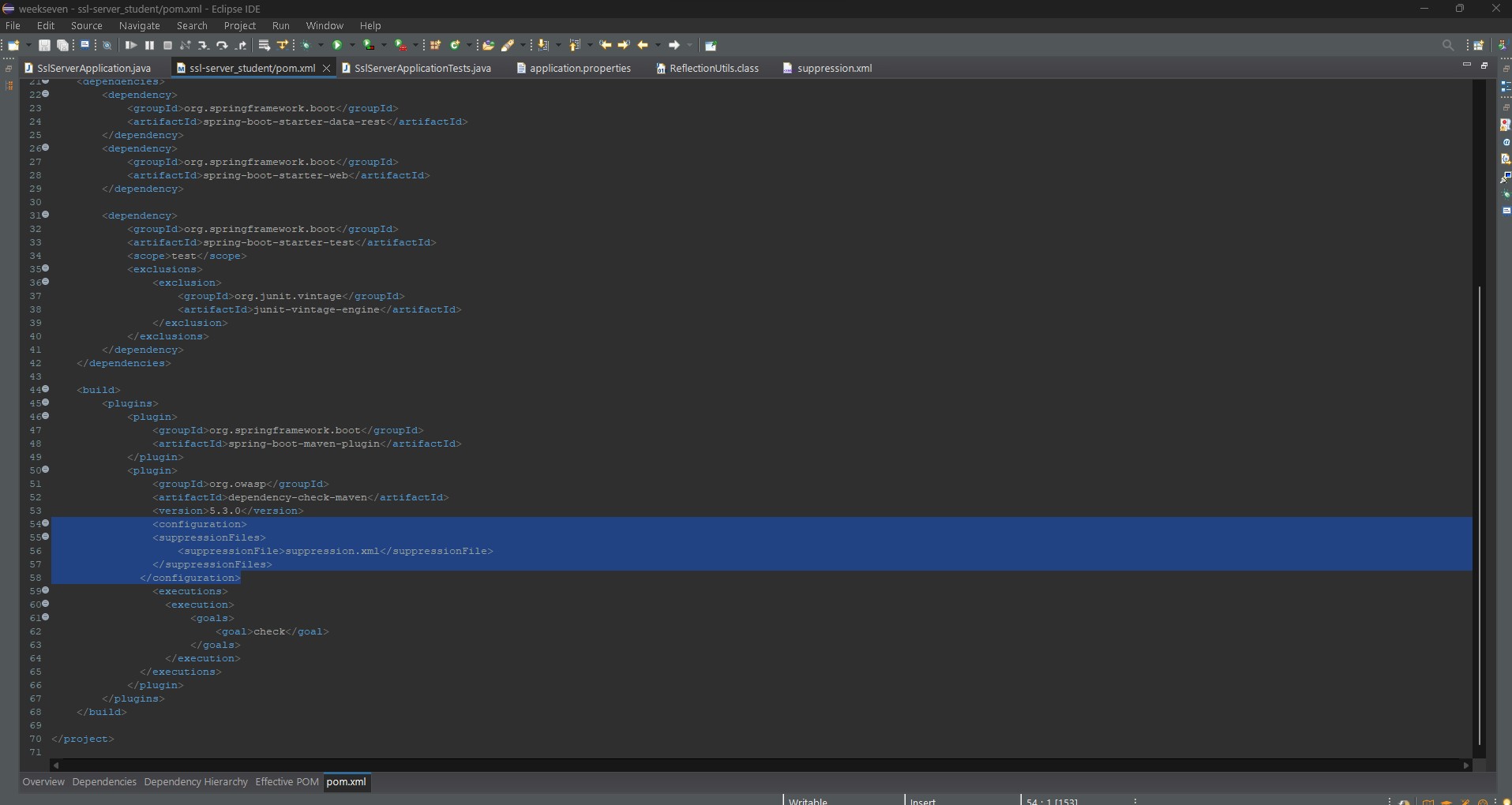


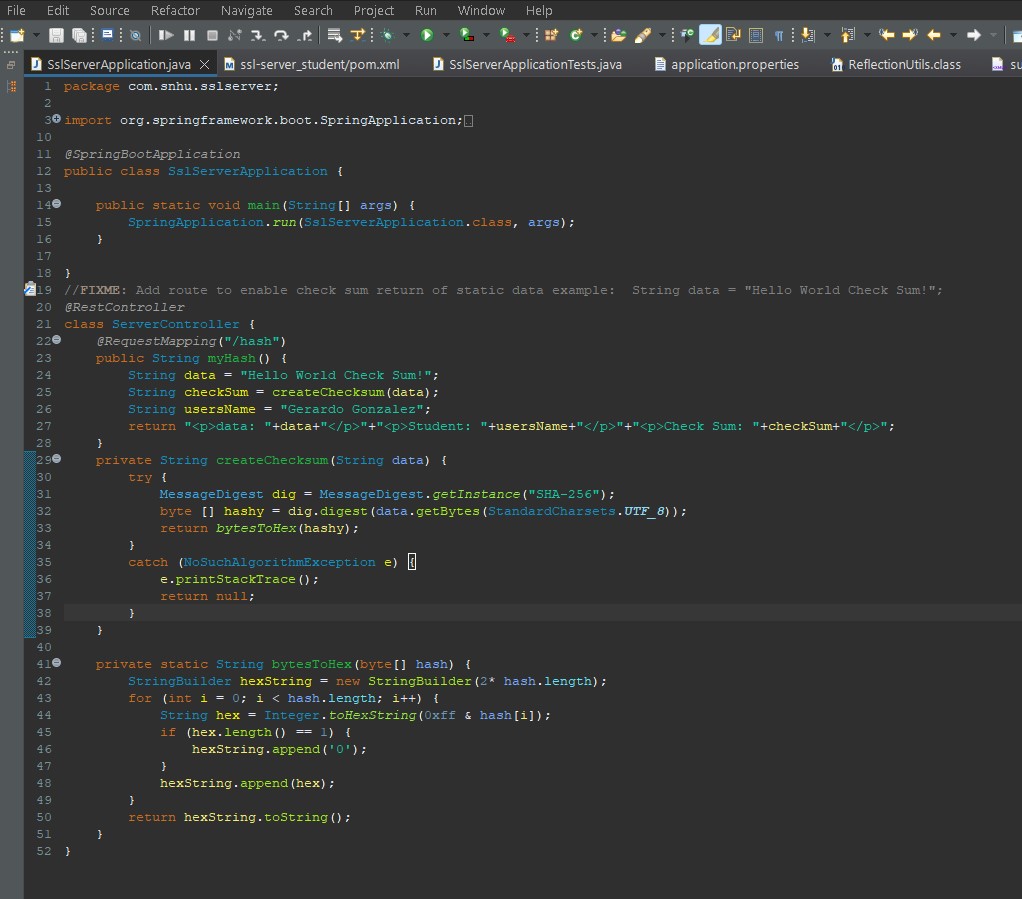
## Secondary Testing

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



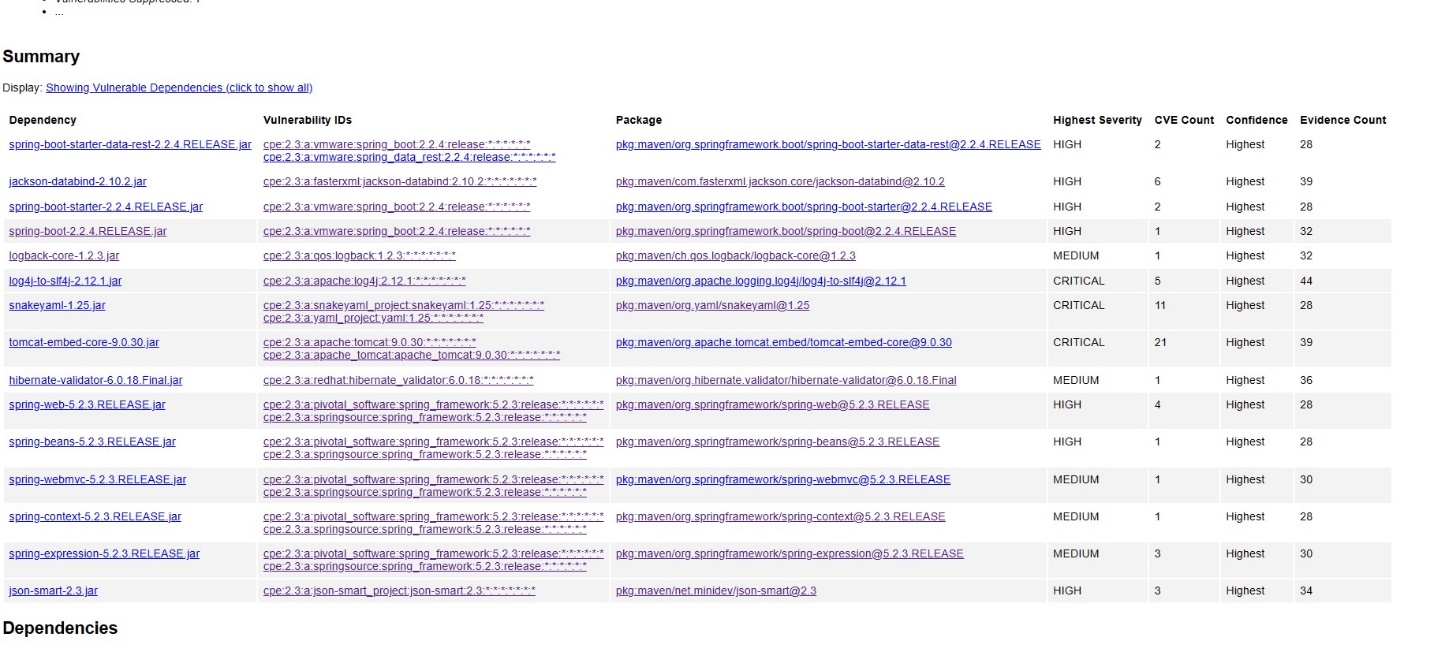


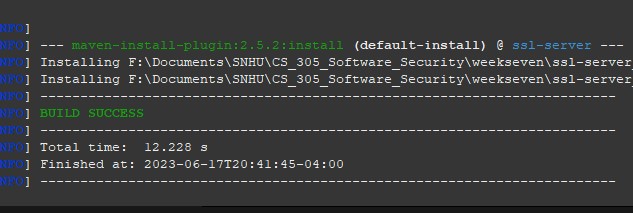
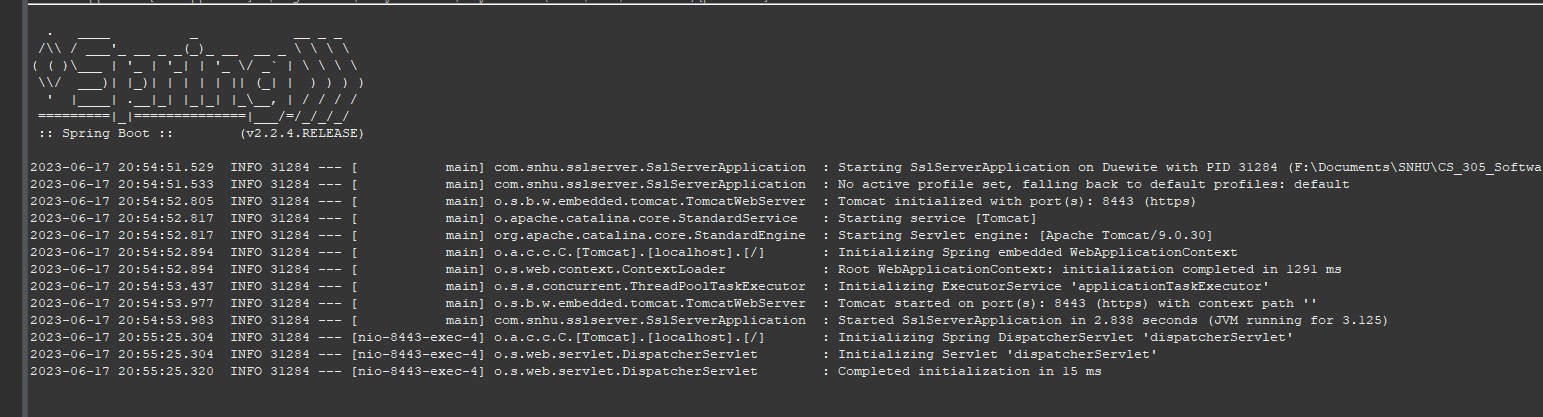




## Functional Testing

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





## Summary

During the code implementation and the given code, we refactored the POM.XML file to include suppression for false positives, on top of this we created a suppression xml file for the false vulnerabilities. Another refactor that I did was the *“application.properties”* file as it needed the key alias, store password, the store name, and the store type. This help make sure the program communicated properly to the https local host.

Additionally, we added the “*ServerController”* class to map the static data. During the development, I created a checksum method to hash out the given data and then convert that data into a hex string. I focused on the Input Validation, API, Cryptography, and Client Server aspects of the vulnerability assessment process flow diagram. This way securely develops data access and the APIs that were used. These include the Spring Boot API and Data such as the hash value and name original data message.

The Process I took to add layers to the security of the application was to use parameters for the data variables and suppress false positives in the dependency check and make the report as accurate as possible. And in conclusion, utilizing the methods demonstrated with the corresponding screenshots attached above shows the process of designing and implementing best practices for encrypting data from a certified source to secure sensitive data and shows an example of how Artemis data will be processed.

## Industry Standard Best Practices

Some industry standards that I followed were the practice of the AES algorithm and using https as both help improve and follow modern industry standards. Another example of how security practices were used was using certificates and key stores. The part had me create a key using the java keytool.exe and using that keystore to send the server to encrypt the data, as you can see in the “*application. properties”* file we added this property and during the initial lunch this was left out and an error was received stating that the transmission of the data was not properly encrypted/decrypted.

By applying the most current security practices we can create an application that follows the most secure practice currently and will benefit a company by keeping sensitive data safe while also reducing the potential damage if a data breach or cyber attack takes place. This could overall save the company thousands if not millions of dollars and since Artemis is a financial company it will be more targeted for its data.

Sources.

Java security standard algorithm names. (n.d.). <https://docs.oracle.com/javase/9/docs/specs/security/standard-names.html#cipher-algorithm-names>

Hougen, A. (2021, May 26). *What is AES encryption & how does it work in 2023? [256-bit VS 128-bit]*. Cloudwards. <https://www.cloudwards.net/what-is-aes/#:~:text=The%20AES%20algorithm%20is%20the%20industry-standard%20encryption%20protocol,better%20performance%20during%20the%20encryption%20and%20decryption%20process%29>.

Graham-Cumming, J. (2020, August 17). *Why secure systems require random numbers*. The Cloudflare Blog. <https://blog.cloudflare.com/why-randomness-matters/>

ClickSSL. (n.d.). *128-bit VS 256-bit SSL encryption – what is the difference?* ClickSSL. <https://www.clickssl.net/blog/128-bit-ssl-encryption-vs-256-bit-ssl-encryption>