****

# 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** | **[10/15/2023]** | **Jacob Senior** | **Conducted security report** |

## 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

Jacob Senior

## Algorithm Cipher

Artemis Financial handles sensitive financial information when doing business with its customers. Due to the sensitive nature of the data being handled, attackers will try to gain unauthorized access to the data. For this reason, Artemis Financial should employ safety measures to prevent unauthorized access to user data. There are two times when an attacker may try to extort sensitive data; when data is in transit, or when data is at rest. When data is in transit, attackers may employ eavesdropping attacks such as man-in-the-middle attacks. When data is at rest, attackers may employ attacks such as SQL injections. In either case, we can protect sensitive data by using encryption. We should encrypt both data in transit as well as data at rest. In fact, many data privacy laws require companies that handle sensitive data to use encryption (Crane, 2019). Additionally, the U.S. government requires sensitive data to “be encrypted using 192- or 256-bit encryption methods” (Rouse, 2023). Given these constraints, I would recommend that Artemis Financial uses the Advanced Encryption Standard (AES Encryption) for encrypting data at rest and data in transit. AES-256 encryption has become known today as the best cipher for encrypting data. AES encryption provides 128, 192, and 256-bit encryption allowing for more flexibility. AES-256 has been adopted as the standard for encryption by the U.S. government, military, and other institutions that store sensitive information due to the strength of AES-256 encryption (Everything, n.d.).

The different bit levels used in AES encryption describe the size of the key used to encrypt and decrypt the data. Since AES is a symmetric algorithm, the same key is used for encryption and decryption which also makes the algorithm more efficient (Everything, n.d.). The larger the size of the key, the more difficult it is to crack the algorithm. The bit level of the encryption also determines how many rounds of modification the algorithm goes through when encrypting the data (Everything, n.d.). These keys are generated using random numbers to make it difficult for hackers to crack the algorithm by brute force. The previous standard for encryption was the Data Encryption Standard (DES) which used a 56-bit key length and a 64-bit block size (AES, 2022). DES was developed in 1977 and by 1999 could be brute forced in less than 24 hours (AES, 2022). AES instead uses 128, 192, or 256-bit key lengths and 128-bit block sizes. Using the 256-bit key length, it would take millions of years to brute force with today's hardware capabilities (Everything, n.d.). To date AES is considered the most secure encryption cipher, however, like DES, there is no guarantee that it will remain secure forever.

AES vs DES Encryption: Why Advanced Encryption Standard (AES) has replaced DES, 3DES and TDEA. (2022, November 14). Precisely. https://www.precisely.com/blog/data-security/aes-vs-des-encryption-standard-3des-tdea

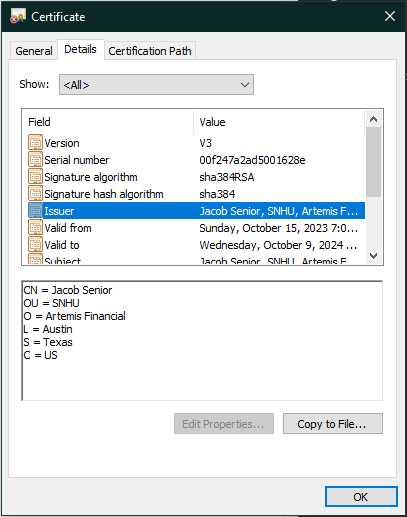
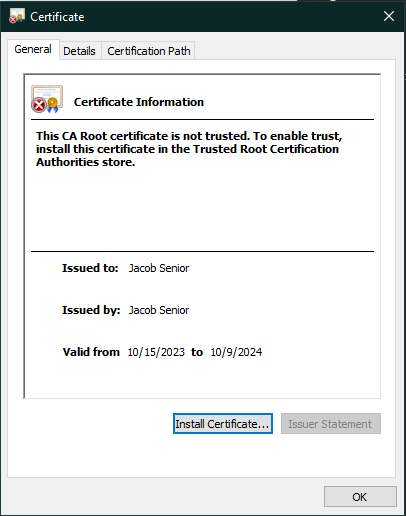
Crane, C. (2019, June 28). 10 Data Privacy and Encryption Laws Every Business Needs to Know. The SSL Store. https://www.thesslstore.com/blog/10-data-privacy-and-encryption-laws-every-business-needs-to-know/

Everything You Need to Know About AES-256 Encryption. (n.d.). Kiteworks. https://www.kiteworks.com/risk-compliance-glossary/aes-256-encryption/#:~:text=AES%2D256%20encryption%20is%20extremely,operating%20in%20highly%20regulated%20industries.

Rouse, M. (2023, June 26). 256-Bit Encryption. Techopedia. https://www.techopedia.com/definition/29703/256-bit-encryption#:~:text=However%2C%20it%20is%20also%20implemented,or%20256%2Dbit%20encryption%20methods.

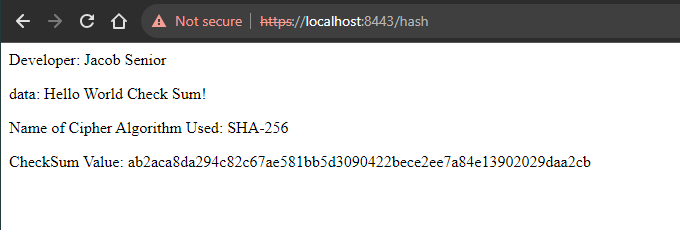
## Certificate Generation

Insert a screenshot below of the CER file.



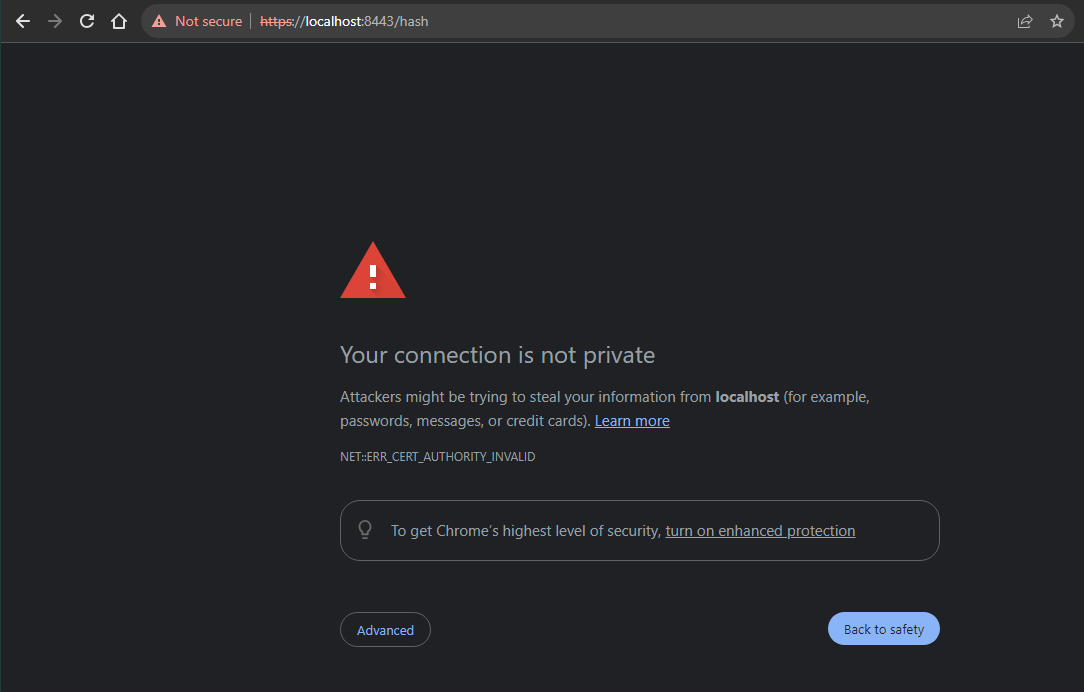
## Deploy Cipher

Insert a screenshot below of the checksum verification.



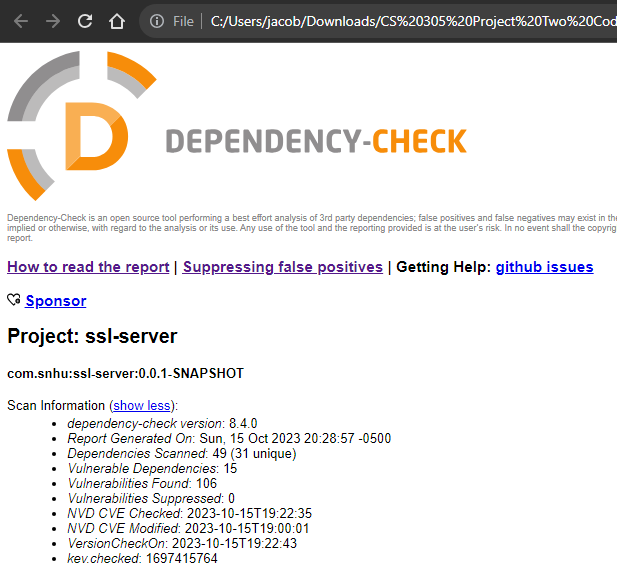
## Secure Communications

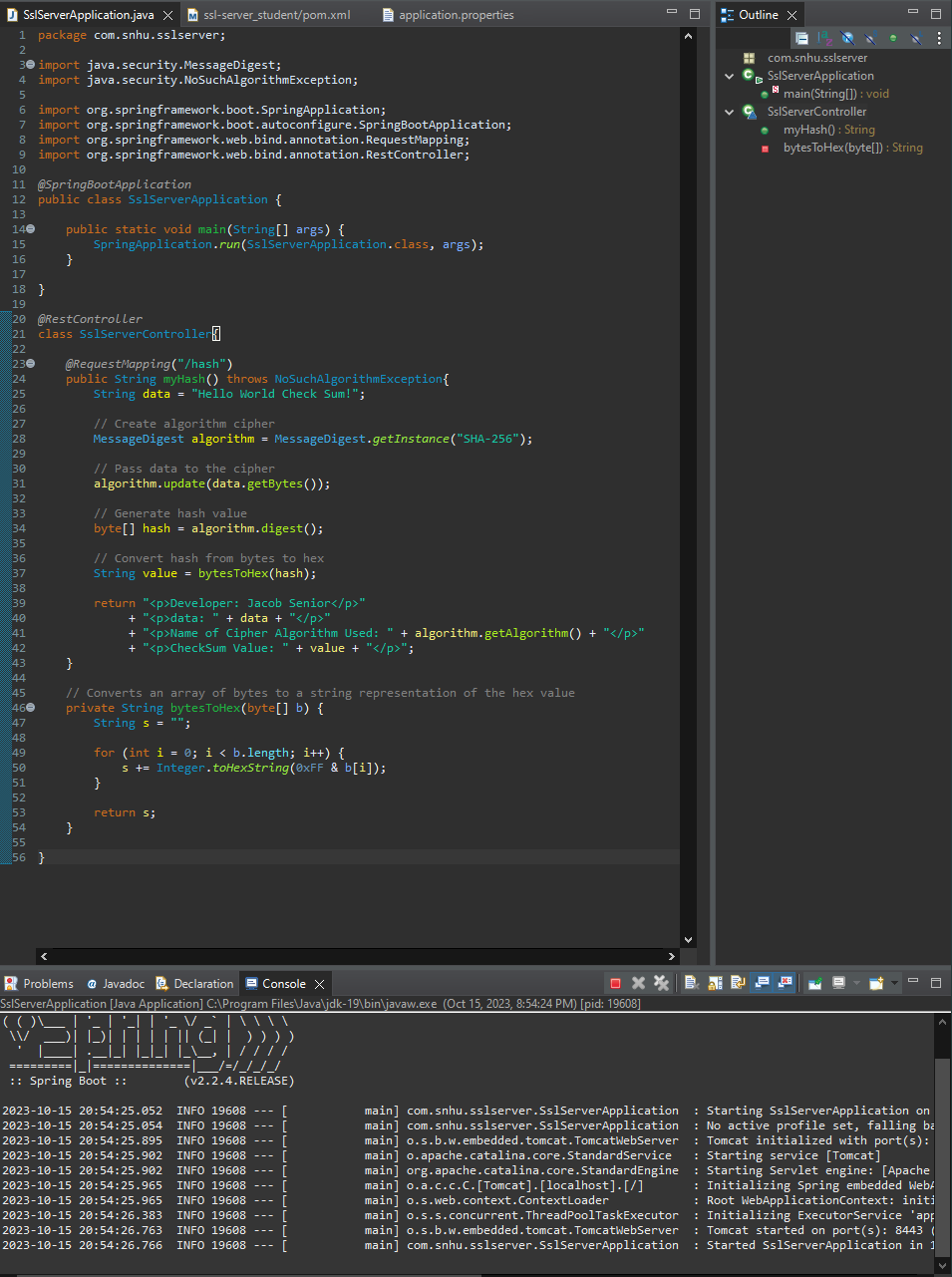
Insert a screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

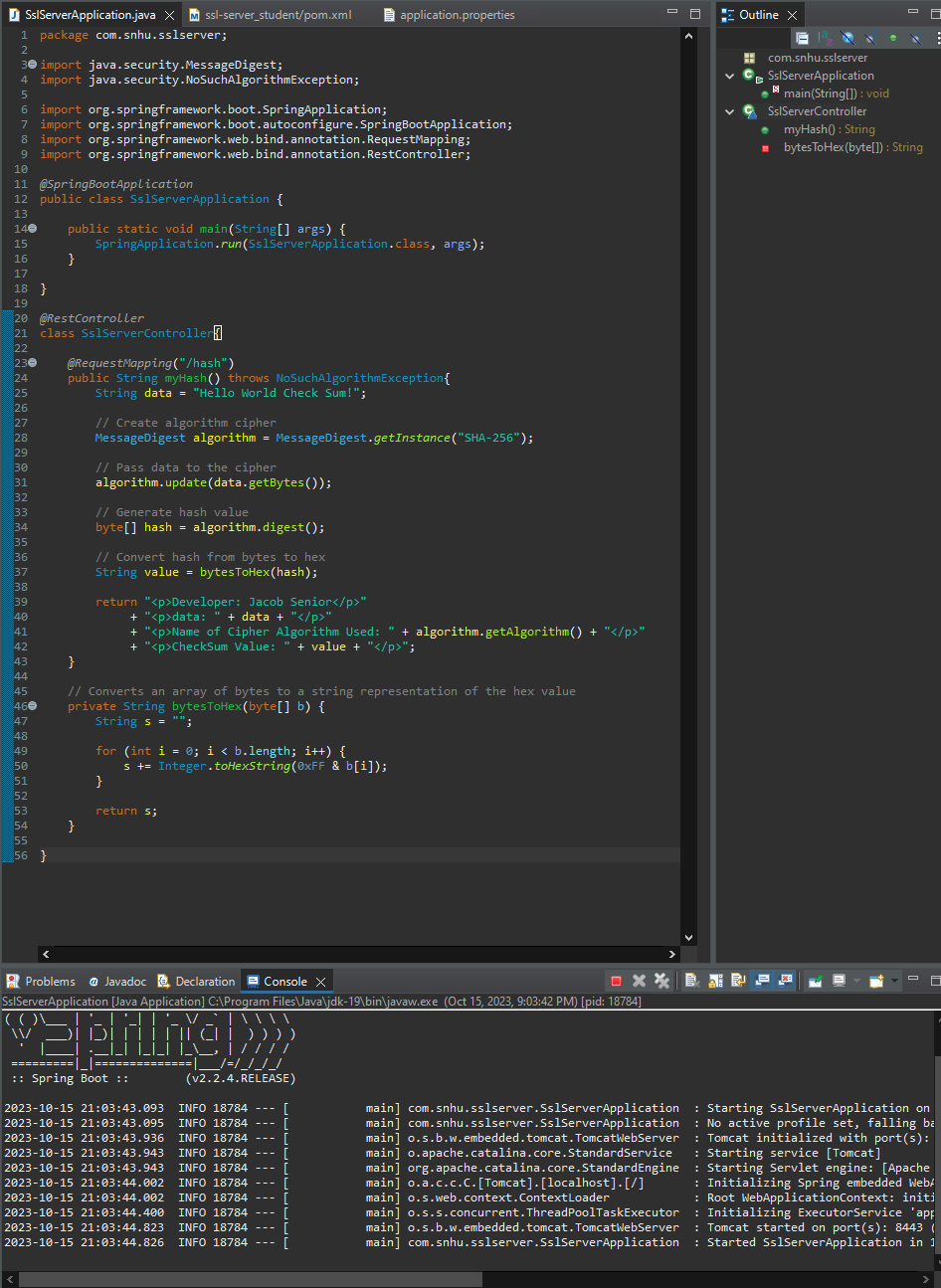
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



## Summary

The areas of security I have addressed in this code are cryptography, secure distributed composing, and code quality. The first step when performing data encryption is to find an encryption algorithm that is secure and fits your program’s needs. I chose SHA-256 because it is still very secure, performant, and flexible. We can then apply this encryption algorithm to our program, providing a safe method of storing data. In addition to protecting data at rest, it is important that we establish a secure connection to protect data in transit. We did so in this program by generating a self-signed certificate to use for an SSL (Secure Sockets Layer) connection. Using this certificate, we were able to create an HTTPS connection with Tomcat by entering our certificate’s information in the application.properties file. When implementing these security features, it is important we use secure coding practice so we do not create new vulnerabilities in the process. It is good practice to perform manual code reviews after refactoring to double-check for the use of good coding practice. A good way to test for new vulnerabilities is using a dependency checker. This can help to show any insecure dependencies we have used in our program. Frequent updates to vulnerable dependencies can be an exhaustive use of resources but is essential for maintaining a secure environment for our application.

## Industry Standard Best Practices

One way I used industry standard best practices to maintain the application’s current security is to apply secure coding practices when adding new code. Guidelines such as OWASP provide strategies we can use to mitigate coding flaws that lead to new vulnerabilities. I also ran a dependency check before and after refactoring the code to ensure that my design did not introduce new dependency vulnerabilities. Additionally, manual code reviews are an important part of the secure programming process to double-check all relevant areas of security after refactoring our code. It is essential for Artemis Financial to apply industry-standard best practices for secure coding for many reasons. Without applying these secure coding practices it is inevitable that we will introduce vulnerabilities into our systems. For example, improper bounds checking for an array could lead to leaking sensitive information or executing remote code. Such attacks not only could damage our system but could also damage our trust with customers. Neglecting proper security practices could also result in legal suits due to violations of data privacy laws. A good system to implement that could elevate the company’s security is DevSecOps. DevSecOps results in a more secure overall product by implementing security practices into the development cycle from the start. This would not only result in a more secure final product but also create a more efficient development cycle. Security is an essential part of software development and should be made a priority from the start of development.