****

# 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** | **12/07/23** | **Gabrielle Maitland** | **Project Two** |

## Client



## Developer

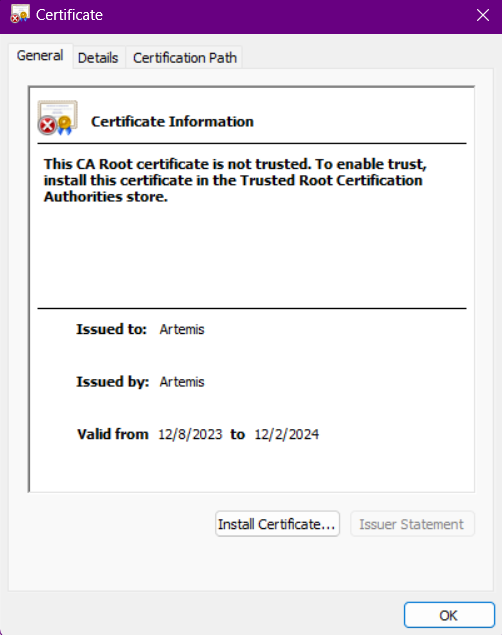
Gabrielle Maitland

## Algorithm Cipher

Given Artemis Financial’s security vulnerabilities, an appropriate algorithm to deploy would be AES Encryption. This algorithm will be strong enough to encrypt the company’s long-term archive files. AES is heralded as a multi-industry standard due to its legacy of being uncrackable. Hash functions are used to convert data into an output of fixed characters, meaning even if your message is 500 characters long, if the hash function is limited to 20 characters, your message will be output as 20 characters. With bit levels, its strength lies in how high the level is. If the bit level is higher, there are more possible key combinations, rendering it more difficult to intercept. Bit levels essentially add an extra layer of difficulty in the cipher. Symmetric encryption, which is used by AES, uses one private key to encrypt and decrypt information. On the other hand, asymmetric encryption uses a private key to encrypt the data and a public key to decrypt data. Randomized numbers help to add unpredictability to a cipher, and therefore more security.

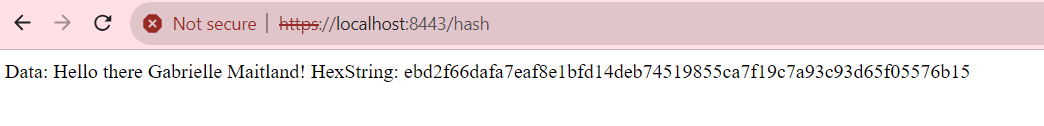
Cryptography got its start centuries ago, with examples of early use dotting the timeline of history. Humans have consistently had a desire to send secret messages to others, whether for political purposes or otherwise, so we see examples even in 1900 BC Egypt. The first known cipher with an encryption key is attributed to Blaise de Vignere, who created his own in the 16th century. Today, the NIST constitutes AES as the standard for symmetric encryption. However, because we are continuously advancing in mathematics and technology, there is the potential for quantum resistant algorithms to be very useful for our future.

## Certificate Generation



## Deploy Cipher

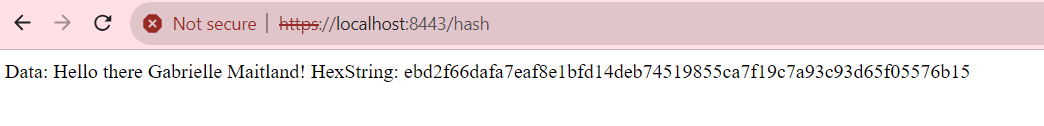
Insert a screenshot below of the checksum verification.



## Secure Communications

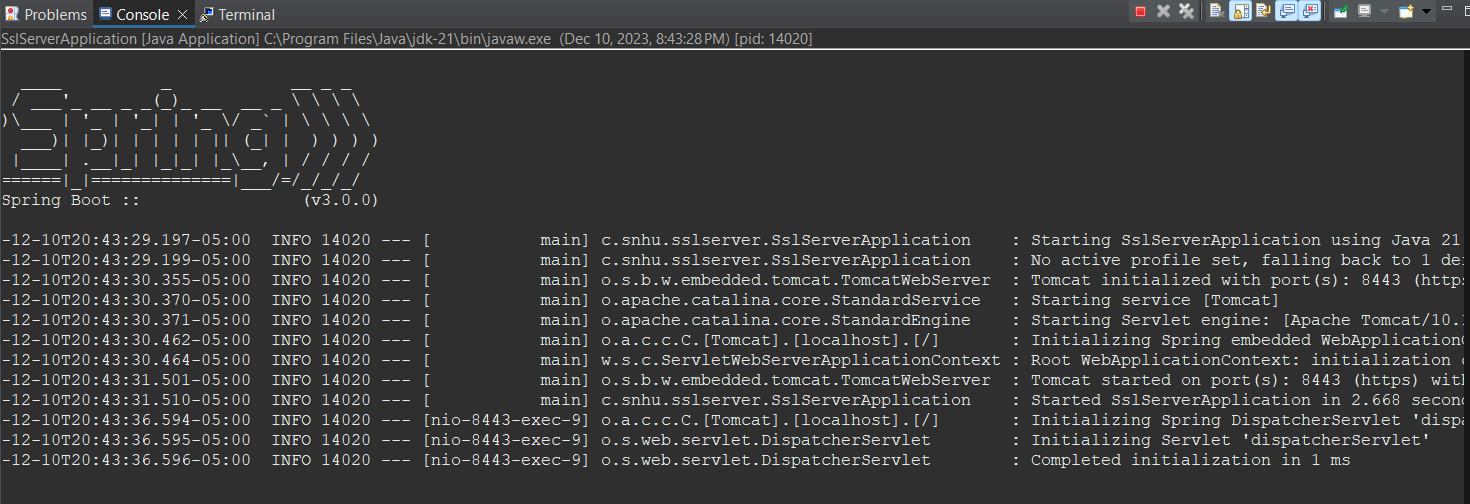
Insert a screenshot below of the web browser that shows a secure webpage.

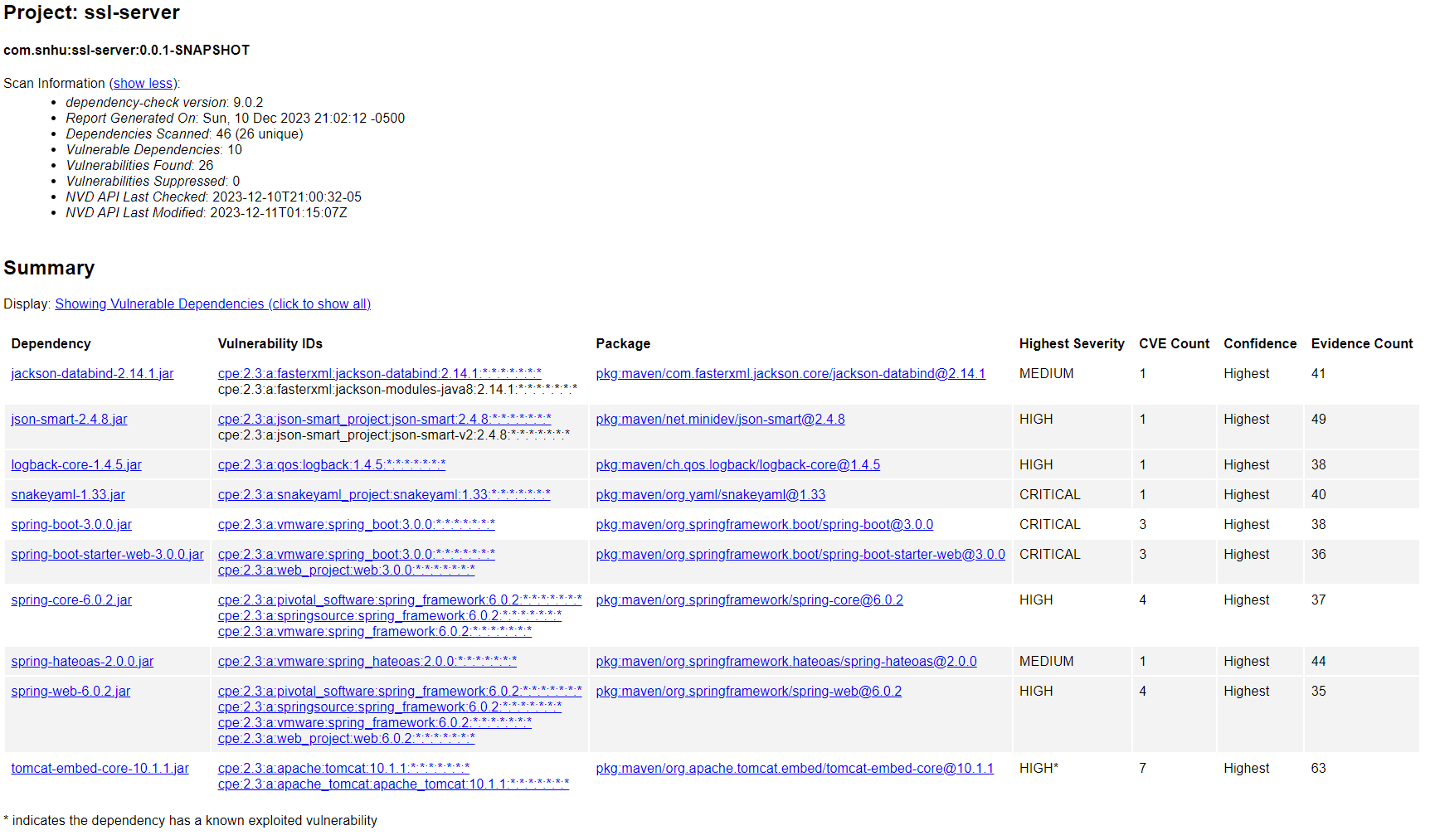
*Still shows as insecure because the certificate is not valid.*



## Secondary Testing

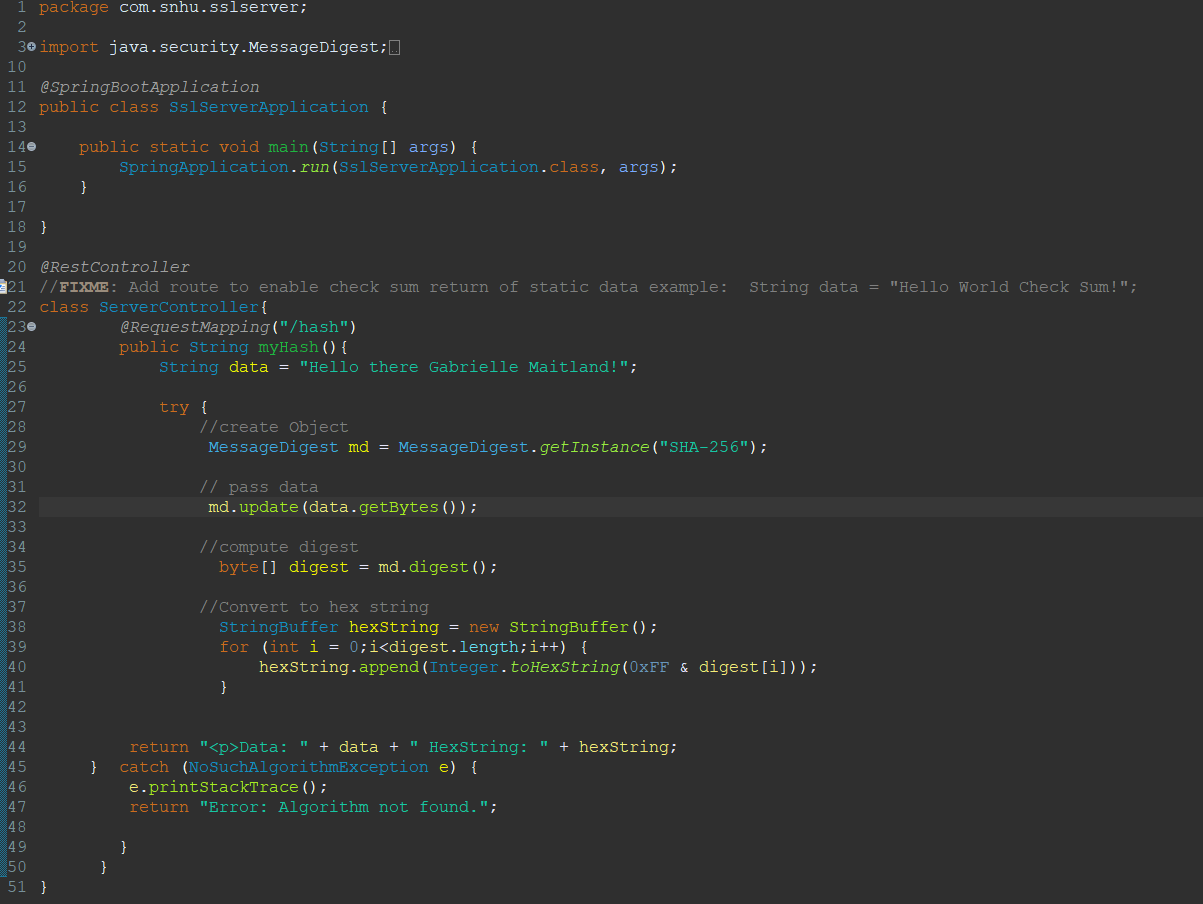
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



## Summary

The code has been refactored to encrypt private information from a user. This is useful for security testing so that there are no potential leaks of data occurring when information is passed back and forth between the site and the user. The areas of security addressed by refactoring the code are Cryptography and Code Quality Layers of security have been added to the application by upgrading dependency packages to stay up to date, as well as identifying any areas that are susceptible to attacks. Please see attached ZIP file of the refactored code base for further discovery.

## Industry Standard Best Practices

To mitigate against known security vulnerabilities, I used industry standard practices through implementing a MessageDigest function to properly encrypt information input by the user. With MessageDigest, I specified the use of SHA-256, which is widely recommended by professionals for hashing. I also used TLS to ensure a secure connection for passing information back and forth. The value of applying these practices benefits the company’s wellbeing because it helps to build a trustworthy reputation for Artemis Financial. Customers want to be sure that their information will not be intercepted and that there will not be a breach of sensitive information from the company’s side. By applying these practices, we strengthen Artemis Financial’s security measures to make these sorts of attacks less and less likely. Artemis Financial can uphold this reputation by regularly completing dependency checks.

**Resources:**

*16 CFR 314.4 -- elements.* (n.d.). https://www.ecfr.gov/current/title-16/chapter-I/subchapter-C/part-314/section-314.4

AppSealing. (2023, October 7). *How AES Encryption is Used in Cybersecurity and Why it Matters*. https://www.appsealing.com/aes-encryption/

*Java Security Standard Algorithm Names*. (n.d.). https://docs.oracle.com/javase/9/docs/specs/security/standard-names.html#cipher-algorithm-names

Kryptall Secure Communication Services. (n.d.). *How Safe is AES Encryption?*https://www.kryptall.com/index.php/information/how-safe-is-aes-encryption

*Levels of Encryption | Tech Talk | Simms International*. (n.d.). https://www.simms.co.uk/tech-talk/understanding-the-levels-of-encryption/

Loo, A. (2023, November 7). *Hash function*. Corporate Finance Institute. <https://corporatefinanceinstitute.com/resources/cryptocurrency/hash-function/>

Maitland, G. (2023). *Module Four Assignment* [Journal Assignment]. Southern New Hampshire University.

Manico, J., & Detlefsen, A. (2014). *Iron-Clad java: Building Secure Web Applications*. McGraw Hill Professional.

Robin, E. (2023, September 3). How do banks encrypt data? - Newsoftwares.net blog. *Newsoftwares.net Blog*. https://www.newsoftwares.net/blog/how-do-banks-encrypt-data/

Sidhpurwala, A. |. H. (2023, April 19). Understanding random number generators, and their limitations, in Linux. *Red Hat*. https://www.redhat.com/en/blog/understanding-random-number-generators-and-their-limitations-linux

Sidhpurwala, H. (2023, June 16). A Brief History of Cryptography. *Red Hat*. https://www.redhat.com/en/blog/brief-history-cryptography