Nate Bennett

****

# 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** | **18 June 2023** | **Nate Bennett** |  |

## 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

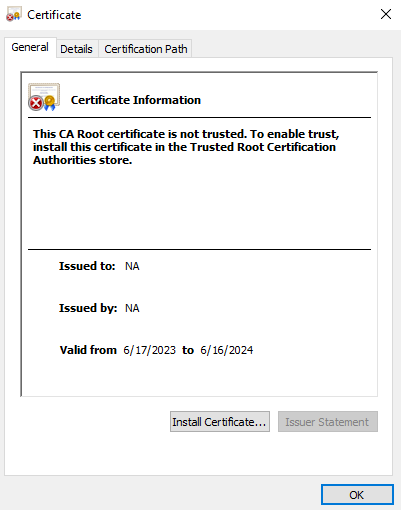
Nate Bennett

## Algorithm Cipher

I advise using the Advanced Encryption Standard (AES) algorithm encryption to satisfy Artemis Financial's security needs. The widely used and extremely secure encryption technique AES is ideal for encrypting and decrypting sensitive data. There are several key lengths available for AES encryption, including AES-128, AES-192, and AES-256. AES-256 offers the maximum level of security and is represented by the number as the key size in bits. The same key is utilized for both encryption and decryption in the symmetric encryption algorithm known as AES. Compared to asymmetric encryption, this makes the implementation process simpler and reduces performance overhead. AES encrypts data using a sequence of substitution and permutation operations on fixed-size blocks of data, usually 128 bits. Depending on the key size, it applies numerous rounds of these procedures. Using a key expansion method, AES creates a series of round keys from the initial encryption key. Each time the data is encrypted, these round keys are utilized to protect the data. The cryptographic community has examined and scrutinized AES in great detail. It is a trusted and dependable encryption algorithm because it has weathered multiple cryptanalytic assaults. AES itself is not a hash function in the sense of hashing algorithms. Asymmetric encryption is used in this approach. However, to guarantee data integrity and authenticity, AES can be paired with a hash function, such as SHA-256 (Secure Hash Algorithm 256-bit). A strong source of randomness is necessary for AES to generate secure encryption keys when using random integers. Since they avoid predictability and strengthen resilience to attacks, random numbers are essential for the security of cryptographic systems. The American National Institute of Standards and Technology (NIST) chose AES as the encryption standard in 2001 following a public competition. This information relates to the past and present condition of encryption algorithms. Due to growing security flaws, it superseded the outdated Data Encryption Standard (DES). Since being chosen, AES has taken the lead as the encryption algorithm of choice for a number of industries, including secure communications, banking, and government. It has shown to be reliable and is still one of the most extensively used encryption systems globally. In conclusion, the AES encryption algorithm cipher is a reliable and popular option for safeguarding sensitive data during transmission. It is the perfect suggestion for the web application of Artemis Financial because of its durability, symmetric encryption nature, and demonstrated security.

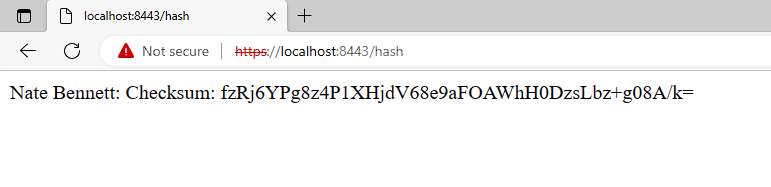
## Certificate Generation

Insert a screenshot below of the CER file.



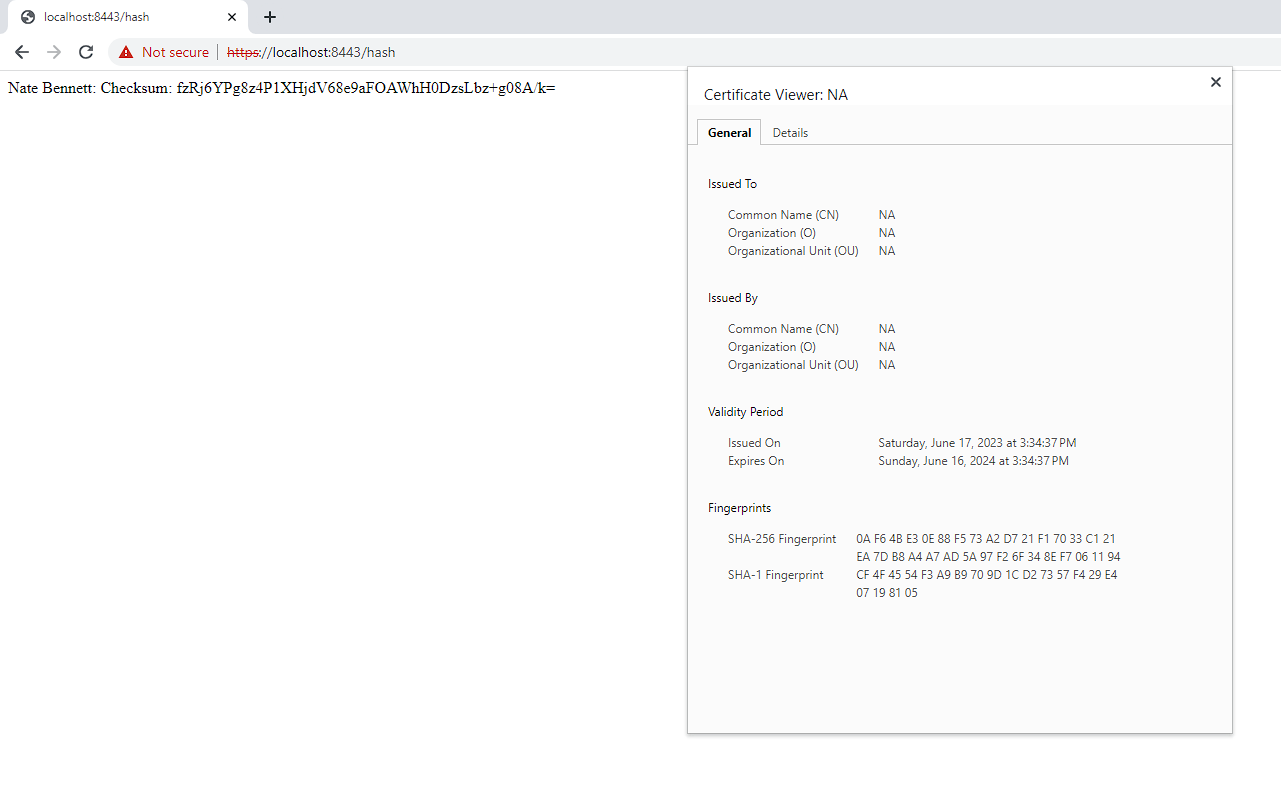
## Deploy Cipher

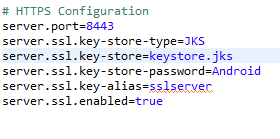
Insert a screenshot below of the checksum verification.



## Secure Communications

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

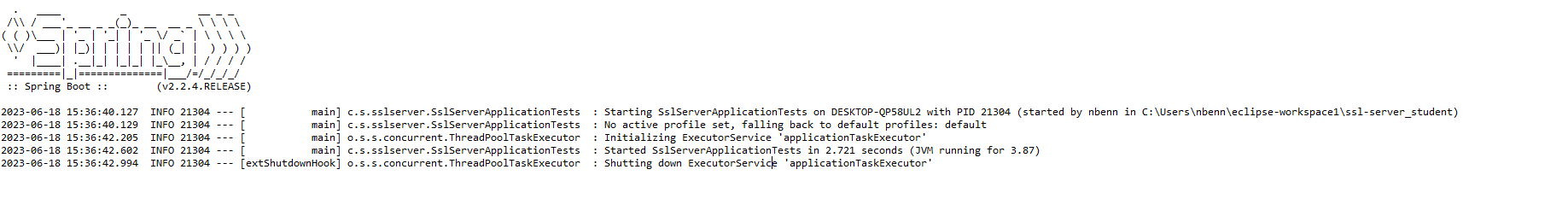


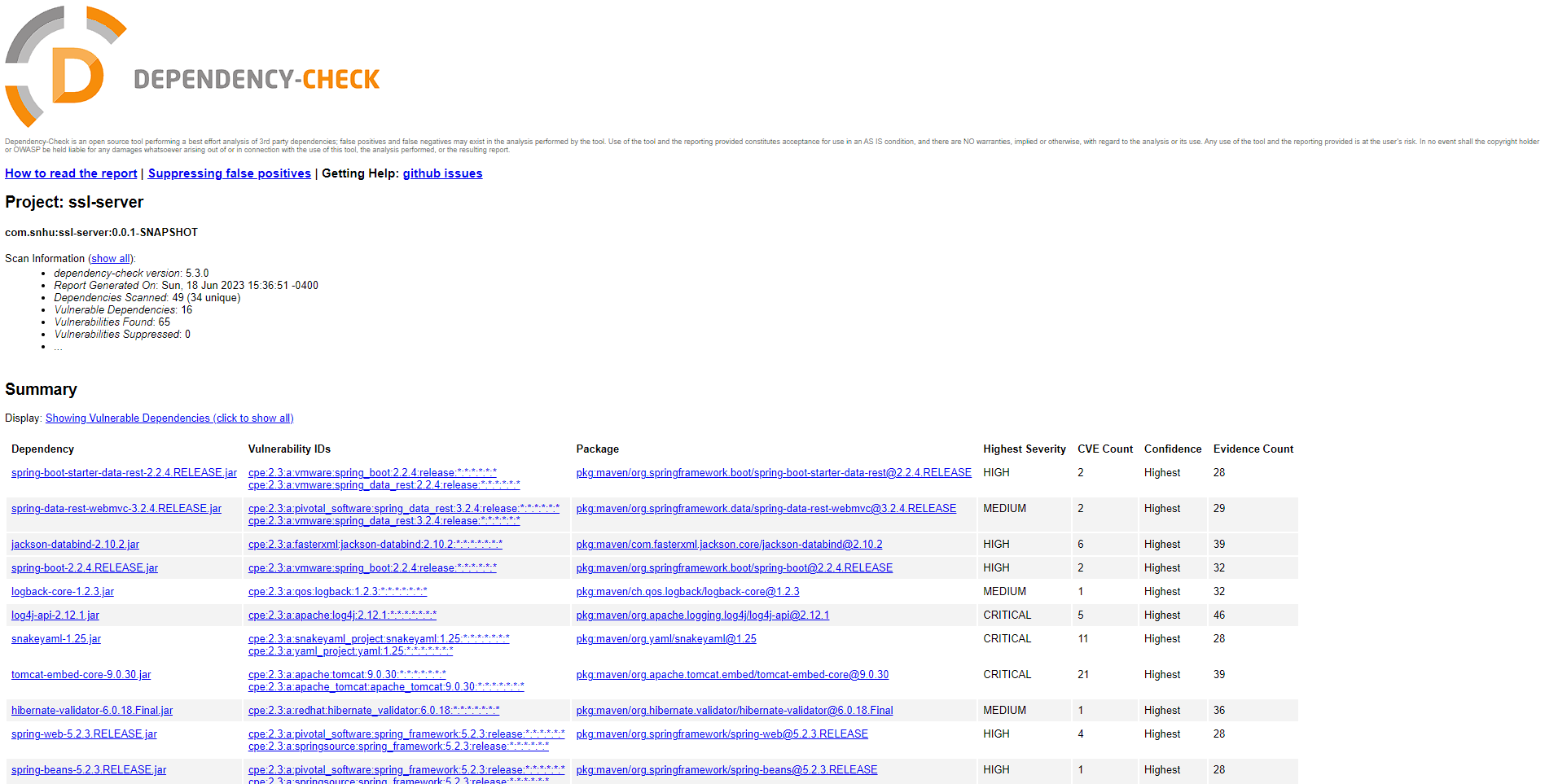


I’m not too sure why the browser still says it is not secure, I also tried to add the certification to the trusted certifications list in my browser, but after it said that the import was successful, for some reason I didn't see it in the list of the trusted certifications. You can see that it seems to recognize that the certification is present, but it won't secure the browser for some reason.

## Secondary Testing

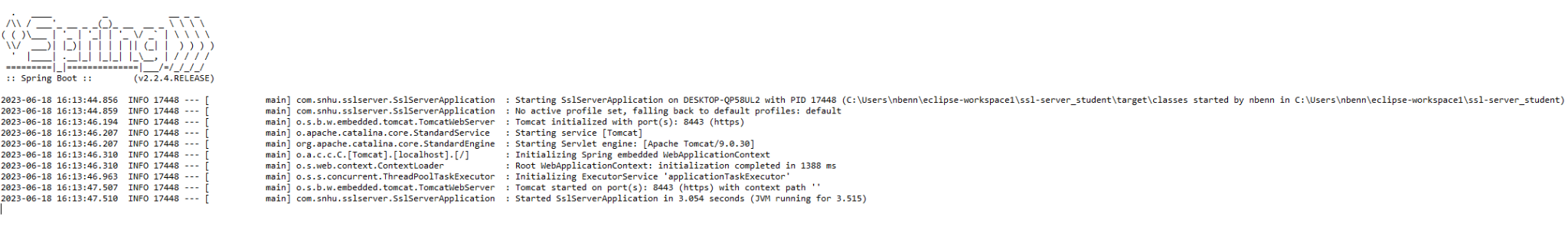
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

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



## Summary

Secure communication over HTTPS is enabled by the code, which also contains the SSL/TLS settings. The code makes use of SSL/TLS to guarantee that all data transferred between the server and client is encrypted and secure against interception and alteration. Sensitive data in the code is encrypted using the AES (Advanced Encryption Standard) standard. Widely employed for data security, AES is a powerful symmetric encryption technique. The code further fortifies the security of the sensitive data by encrypting it with AES. To create a checksum of the encrypted data, the algorithm uses the Secure Hash Algorithm, SHA-256, a 256-bit hash function. Data integrity checks and checksum creation are both made possible by hashing. As a result of its security features, SHA-256 is a well-known cryptographic hash algorithm.

## Industry Standard Best Practices

The code hashes data using SHA-256 and encrypts it using AES. Both the AES and SHA-256 cryptographic algorithms are well-known for their power and resistance to well-known attacks. These industry-standard algorithms are used by the code to guarantee a greater level of security. The code uses SHA-256 hashing to create an AES key from the supplied data. For encrypted data to remain confidential and authentic, key management is essential. The code adheres to best practices by making use of a robust key derivation function SHA-256 and handling the creation and storage of the AES key in an appropriate manner. In order to enable secure communication over HTTPS, the code configures SSL/TLS. Encrypted transport layer security offered by HTTPS guards against unauthorized access and data eavesdropping. The code creates a secure link between the server and clients by following accepted industry best practices for SSL/TLS configuration. I also implemented comments throughout the code to potentially increase readability and comprehension of the code.