****

# Practices for Secure Software Report

**Table of Contents**

[**Document Revision History 3**](#_30j0zll)

[**Client 3**](#_1fob9te)

[**Instructions 3**](#_3znysh7)

[**Developer 4**](#_2et92p0)

[**1. Algorithm Cipher 4**](#_tyjcwt)

[**2. Certificate Generation 4**](#_3dy6vkm)

[**3. Deploy Cipher 4**](#_1t3h5sf)

[**4. Secure Communications 4**](#_4d34og8)

[**5. Secondary Testing 4**](#_2s8eyo1)

[**6. Functional Testing 4**](#_17dp8vu)

[**7. Summary 4**](#_3rdcrjn)

[**8. Industry Standard Best Practices 4**](#_26in1rg)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **June 22, 2024** | **Andrew Emilio DiStefano** | **This document describes the changes made to the code base of the Artemis Financial application. The following changes to the code base we received reflect the security needs of our client Artemis Financial.** |

## Client



## Developer

Andrew Emilio DiStefano

## Algorithm Cipher

* **Provide a brief, high-level overview of the encryption algorithm cipher.**

We recommend the use of the SHA256 algorithm for this scenario. The SHA family of algorithms have a variety of highly beneficial specifications for this scenario including the fact that this family of algorithm ciphers implement the avalanche effect, which means that “even the slightest change in the input message completely changes the output” (Encryption Consulting, 2024). This prevents malicious actors from deciphering the original text and assists in informing the user whether or not the original text has been changed” (Encryption Consulting, 2024).  
 While a variety of sources describe SHA1 as the checksum generation algorithm which performs best in terms of speed (Awati, 2024), SHA256 is considered much more resistant to hash collision attacks since “SHA1 is an older and weaker hash function than SHA256, and it is vulnerable to collision attacks” (KeyCDN, 2023). “A hash collision attack is a type of attack where an attacker generates two different input messages that produce the same hash output” (KeyCDN, 2023). This means that it is possible for malicious actors to find two unique messages that result in the same hash value when using the SHA1 algorithm (KeyCDN, 2023). “This type of attack…can allow an attacker to forge digital signatures, spoof identities, or create malicious software that can bypass security checks” (KeyCDN, 2023). This of course means that SHA256 would be a better choice than SHA256 for checksum generation.  
 While SHA512 is technically more secure than SHA256, the use of SHA512 involves slower speeds and requires higher computational power. SHA256 is widely considered secure enough that the chances of a successful collision attack are astronomically low. SHA256 is currently used in a variety of professional applications and systems including SSL certificate generation for websites and use in a variety of popular blockchains (Rhodes, 2023). Considering the fact that SHA256 is considered secure enough for these types of applications by the industry, and the speed and resource-intensity benefits afforded by the use of SHA256 ove SHA512, we believe that SHA256 would be a better choice than SHA512 for checksum generation in this scenario.

* **Discuss the hash functions and bit levels of the cipher.**

“In encryption, data is transformed into a secure format that is unreadable unless the recipient has a key. In its encrypted form, the data may be of unlimited size, often just as long as when unencrypted. In hashing, by contrast, data of arbitrary size is mapped to data of fixed size. For example, a 512-bit string of data would be transformed into a 256-bit string through SHA-256 hashing” (N-able, 2019).  
 When describing an algorithm, the term ‘bit level’ refers to the level of complexity of the algorithm and how much computing power is required for the algorithm to be used in a given application or function. A higher bit level implies a more complex operation which requires more computing power to implement, while a lower bit level implies the opposite.  
 “SHA-256…generates a fixed-size 256-bit (32-byte) hash value from input data of arbitrary length” (Securiti, 2024). The SHA256 algorithm is known for its reliable level of security and significant resistance to collision attacks, which makes it a popular choice for [digital signatures](https://securiti.ai/glossary/digital-signatures/), authentication, and password [hashing](https://securiti.ai/glossary/hashing/) (Securiti, 2024).

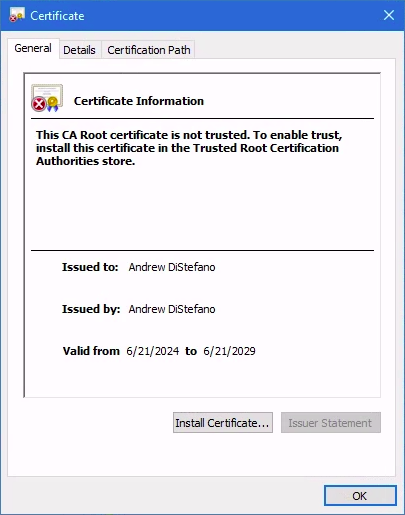
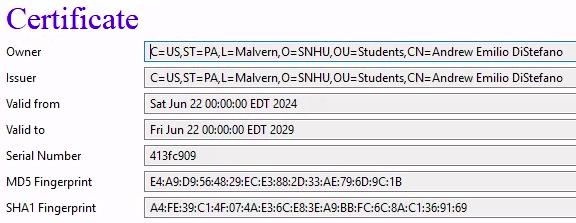
* **Explain the use of random numbers, symmetric versus non-symmetric keys, and so on.**

Cipher algorithms use random numbers produced by random number generators to create secret keys which can be used to decrypt encrypted information. The use of random numbers is a foundational part of the process by which algorithm ciphers encrypt data. The complex mathematics used by these ciphers to generate private keys needs to be based on a randomized initial value or values which result(s) in a highly abstracted series of characters uninterpretable to any person or machine which does not have access to this key.   
 Ciphers which use non-symmetric key encryption use a public key for encryption and a private key for decryption. Access to the public key does not give one the ability to decrypt data encrypted with the private key, which means that the public key can be shared openly without compromising the encrypted data. This makes non-symmetric key encryption ideal for authentication and the encryption of relatively small amounts of data, but adds a significant amount of complexity which makes it less ideal for the encryption of large data sets.   
 Ciphers which use symmetric key encryption use only one key for both encryption and decryption. This method is less complex than non-symmetric key encryption, and can make the process much faster and more efficient. This makes symmetric key encryption ideal for large data sets. Since there is only one key and therefore no public key involved in symmetric key encryption, the key must be shared securely and can not be made public to parties which are not intended to decrypt data.

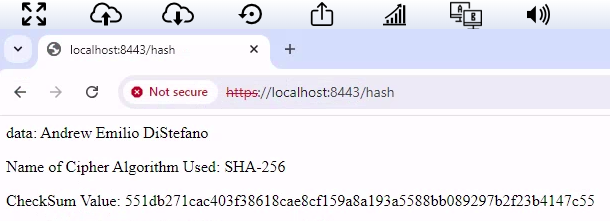
* **Describe the history and current state of encryption algorithms.**

The history of encryption algorithms could be said to have begun with the use of cryptography which dates back to 1900 BC (Sidhpurwala, 2023) and has been used throughout history for a number of applications where security was of the utmost importance including espionage and wartime correspondence involving sensitive information. In the 1970s, the National Bureau of Standards (now called NIST) put out an RFP for a block cipher to use as the national standard which resulted in the acceptance of the cipher algorithm Lucifer (Sidhpurwala, 2023) which given the official name DES, which stands for Data Encryption Standard. Since the encryption key used by DES was much smaller than those used today, advances in computation led to the breaking of DES which led to another RFQ from NIST. In order to address the risks posed by potential malicious use of the incredibly powerful computational systems (and the increased economic accessibility of relatively powerful computers) today, incredibly robust algorithms such as SHA and AES are used by both private and public government entities to protect sensitive information including customer information, trade secrets, and even information which could compromise the safety of US citizens if it were to fall into the wrong hands.

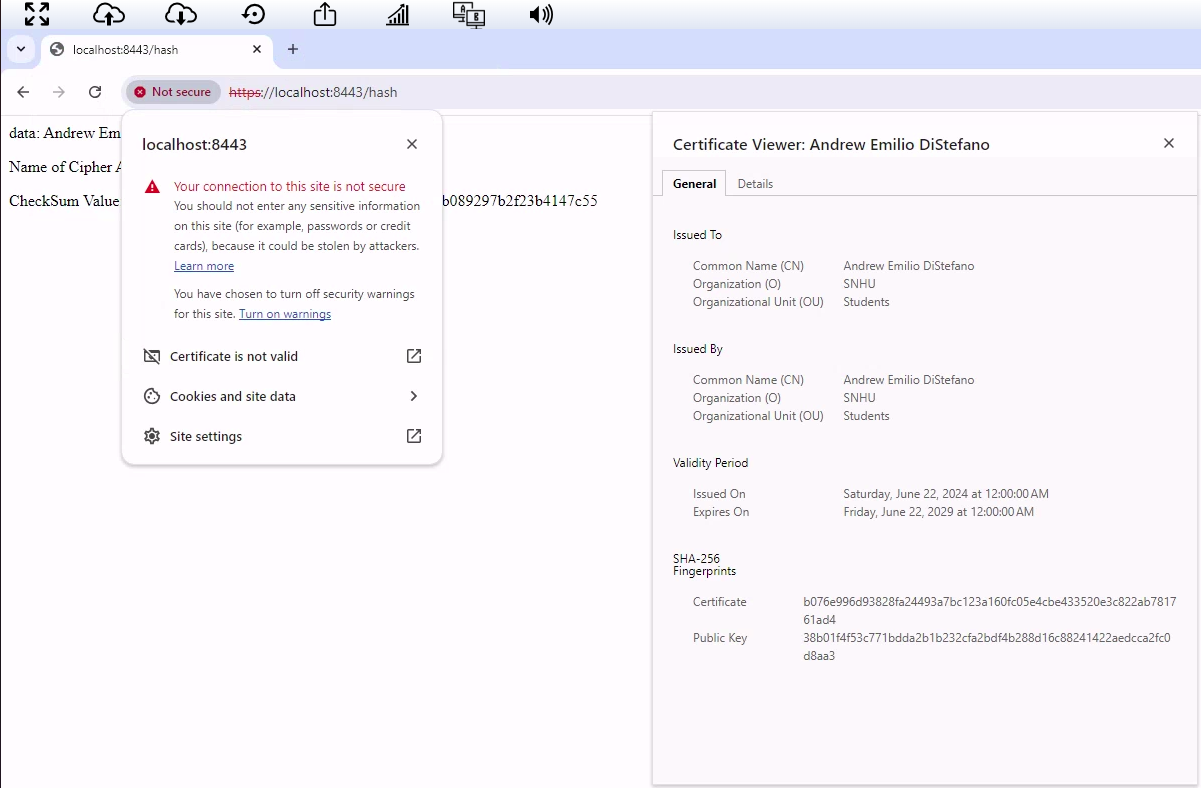
## Certificate Generation

## Deploy Cipher

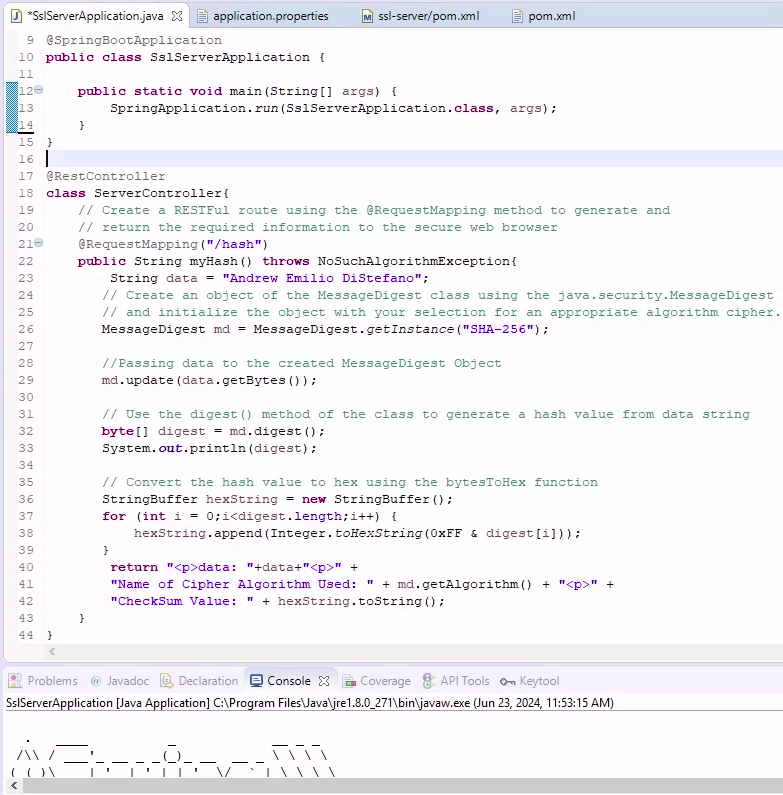


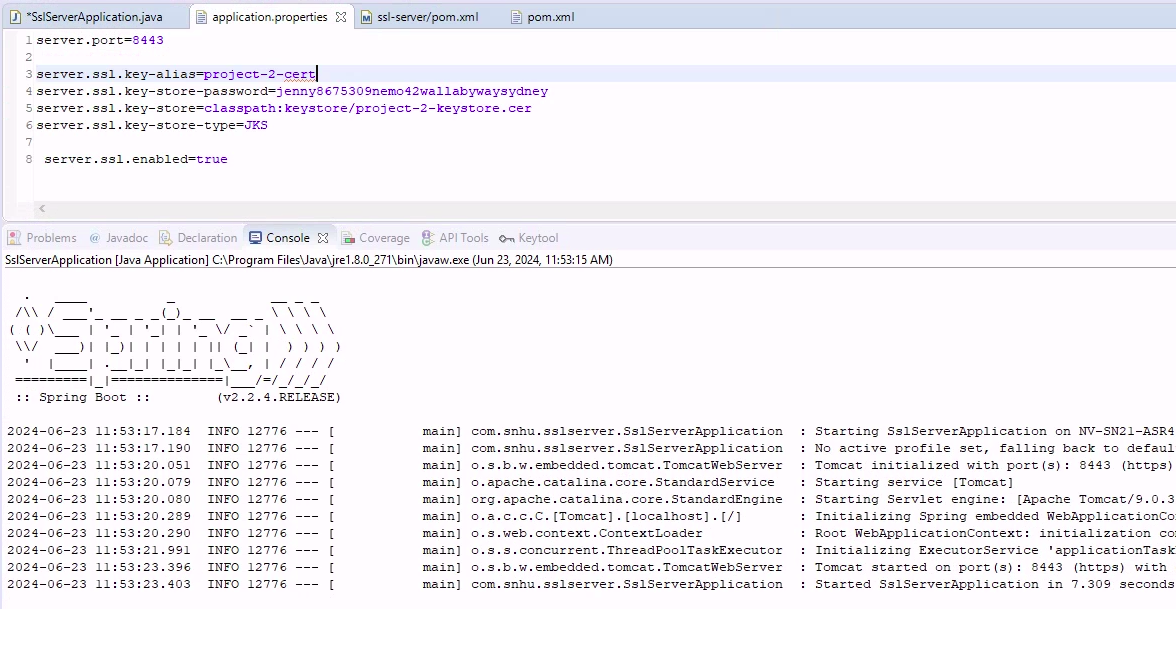
## Secure Communications

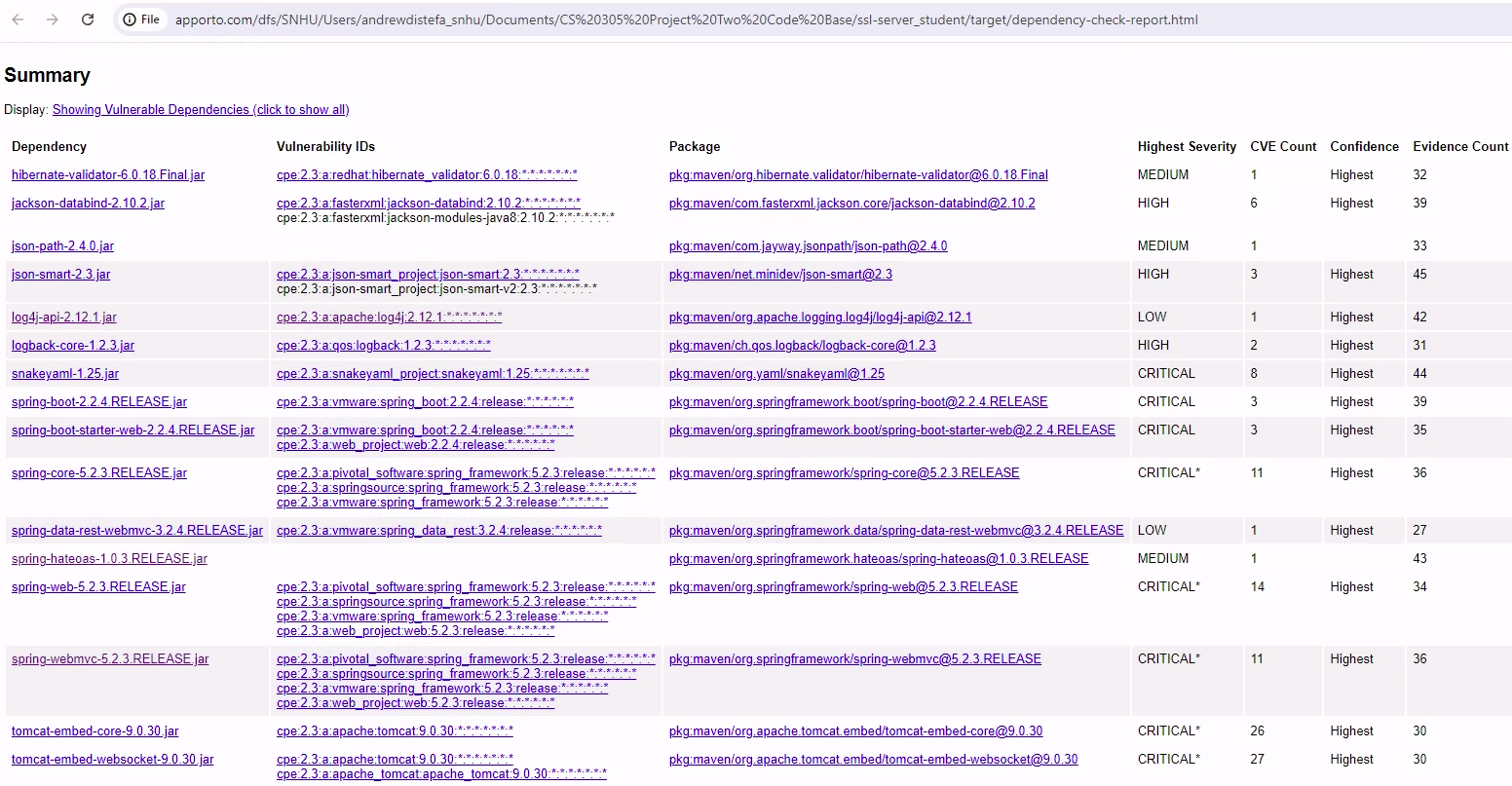


In order for the URL bar to depict a ‘lock’ icon rather than the ‘not secure’ alert, my certificate would have had to have been added to the device’s directory meant to store CA certificates. Due to the fact that this functionality is not permitted to students using the Apporto Windows environment, I was unable to add my self-signed certificate to this directory.

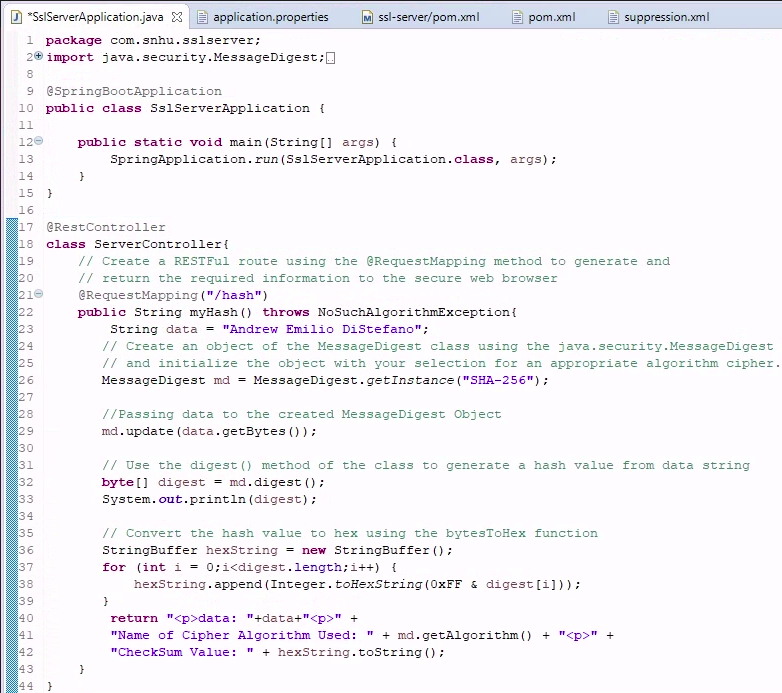
## Secondary Testing

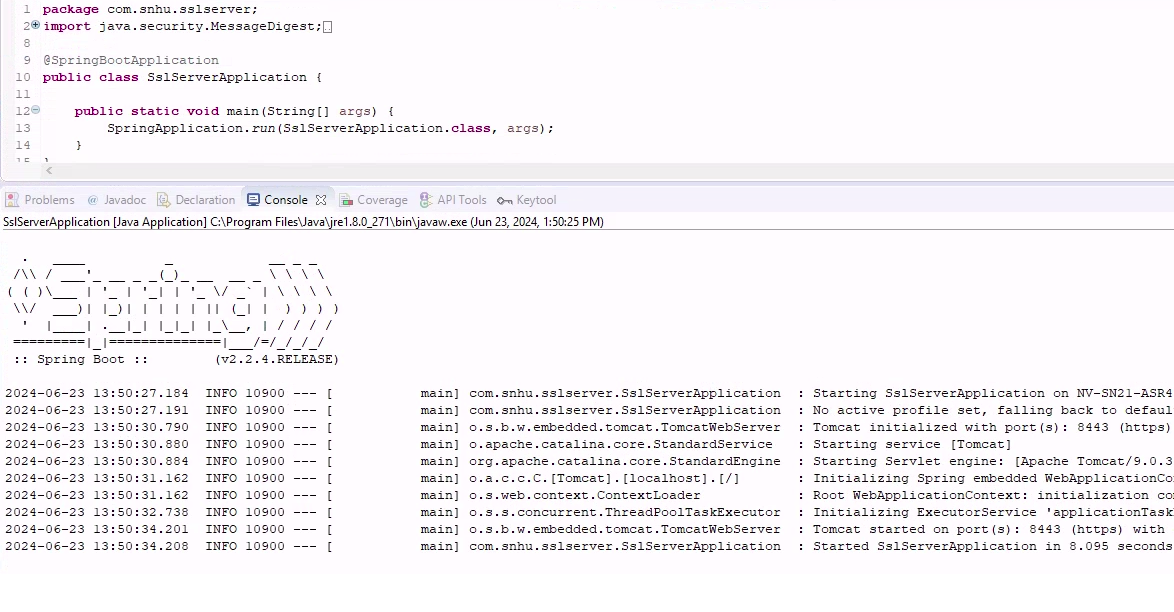






## Functional Testing





## Summary

**Discuss how the code has been refactored and complies with security testing protocols. In the summary of your practices for secure software report, be certain to address the following items:**

* **Refer to the vulnerability assessment process flow diagram in the Supporting Materials section. Highlight the areas of security that you addressed by refactoring the code.**

The areas of security which I addressed by refactoring the original code base include cryptography, client/server, code error, and code quality. In order to address the encryption area of security, I implemented the SHA256 algorithm cipher to secure data and communications which will be used by the Artemis Financial application. In order to address the client/server area of security, I implemented a self-signed certificate system which can be used to verify that clients are in fact connecting to the right application and that those connecting to the system are in fact clients of Artemis Financial. In order to address the code error area of security, I made sure that the application is able to successfully run with no errors and that information is able to successfully be encrypted, then decrypted and depicted on the screens of Artemis FInanclai’s clients.  
 In order to address the code quality area of security, I ran the OWASP maven dependency check to see if there were any vulnerabilities associated with my refactoring of the code in the original code base. In order to detect vulnerabilities as accurately as possible, I made sure to include the latest version of the OWASP dependency check Maven tool (version 9.2.0) rather than using the version already included in the ‘pom.xml’ file upon receival of the code base (which was version 5.3.0). I then suppressed the false positives included in the first iteration of the report. I identified false positives in a number of ways, including by checking to see if the CPE identifiers matched and if the vulnerabilities would be fixed by using an algorithm from the SHA family of algorithms (which we will be using to encrypt the data and communications of Artemis Financial which will be used in this application).

* **Discuss your process for adding layers of security to the software application.**

Each step described in this document effectively added another layer of security to the software application. TLS/SSL establishes a secure connection between systems is an incredibly important consideration in terms of security. The use of a cipher algorithm allows sensitive data and communications to be encrypted and kept from being taken advantage of by malicious actors. The use of HTTPS rather than HTTP further encrypts data sent across or between networks to further obscure it from potential attackers. Static testing helps confirm the strength and robustness of code, including in terms of security. Functional testing assures that a program can run without throwing errors and can achieve the goals for which it was created.

## Industry Standard Best Practices

**Explain how you applied industry standard best practices for secure coding to mitigate known security vulnerabilities. Be sure to address the following items:**

* **Explain how you used industry standard best practices to maintain the software application’s existing security.**

I took several important measures to assure the use of industry standard best practices for secure coding to mitigate known security vulnerabilities. My use of the SHA256 algorithm cipher, which is “one of the most widely used hash algorithms” (Simplilearn, 2024), reflects decisions made by some of the largest players in the tech industry in pursuit of similar goals to those of Artemis Financial in this scenario. “The standard way to protect against the network-based threats mentioned earlier is to use the cryptographic protocol Secure Sockets Layer (SSL) or its modern incarnation, Transport Layer Security (TLS)” (Detlefsen, 2014). I used an SSL certificate in order to secure data in transit to and from the Artemis Financial application system. OWASP standards and best practices were also taken into account in the development of this system. Use of the OWASP dependency check maven tool assured our consciousness of any vulnerabilities related to our code or to our dependencies, and therefore to our ability to address them.   
 In my experience, most industry leaders suggest that code should be reviewed by multiple individuals or at least by one person other than the developer themself. Having received this code base written by another developer, my review of it adds an alternative viewpoint based on my own personal skills and experience to the system.

* **Explain the value of applying industry standard best practices for secure coding to the company’s overall well-being.**

Industry standards and best practices “are incredibly powerful business drivers for security that have painful financial consequences if not followed” (Detlefsen, 2014). Standard best practices are methods which have been tried and tested throughout the history of a given industry. Since attackers often use similar methods to maliciously penetrate systems, the use of industry best practices is especially relevant in the software development industry in terms of security. It is of the utmost importance that organizations communicate and set standards based on their past experience defending against hackers and cybercriminals so that they can pool their knowledge and use it to all of their benefit. The development of industry standard best practices is a great way for organizations to do this in tandem, and adherence to these industry standard best practices is an incredibly important way for them to significantly decrease the chances of a successful attack being perpetrated against them.

**Sources:**  
  
Awati, R. (2024). *Cryptographic Checksum.* TechTarget.  
 <https://www.techtarget.com/searchsecurity/definition/cryptographic-checksum>   
  
Detlefsen, A. & Manico, J. (2014). *Iron-Clad Java: Building Secure Web Applications.* McGraw Hill Computing.   
  
Encryption Consulting. (2024). *What is SHA? What is SHA used for?*   
 <https://www.encryptionconsulting.com/education-center/what-is-sha/>   
  
KeyCDN. (2023, February 22). *SHA1 vs SHA256.* <https://www.keycdn.com/support/sha1-vs-sha256>   
  
N-able. (2019, September 12). Sha-256 Algorithm Overview.   
 <https://www.n-able.com/it/blog/sha-256-encryption>   
  
Rhodes, D. (2023, December 14). *SHA-256 Cryptographic Hash Algorithms.* Komodo Platform. <https://komodoplatform.com/en/academy/sha-256-algorithm>   
  
Securiti. (2024). *SHA-256 (Secure Hash Algorithm 256-bit).*   
 <https://securiti.ai/glossary/secure-hash-algorithm-sha-256-bit>   
  
Sidhpurwala, (2023, January 12). *A Brief History of Cryptography.* Red Hat.   
 <https://www.redhat.com/en/blog/brief-history-cryptography>   
  
Simplilearn. (2024). A Definitive Guide to Learn The SHA-256 (Secure Hash Algorithms).   
 <https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm>