

# 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** | **[2/25/2023]** | **Kenneth Fancher** |  |

## 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

Kenneth Fancher

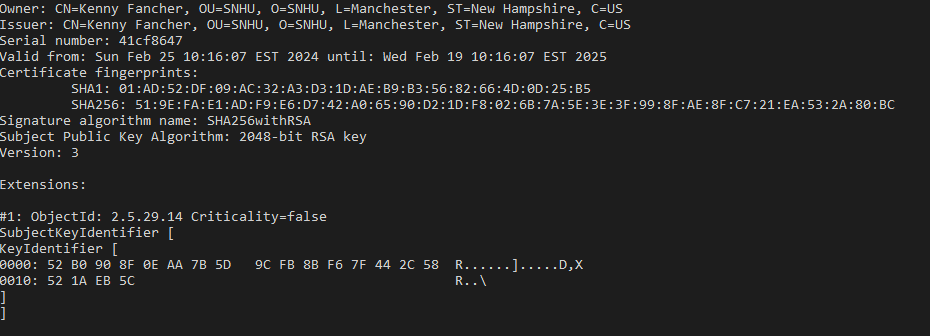
## Algorithm Cipher

Artemis Financial is company working with highly sensitive financial records that are linked to companies internationally, any breach of information would do irreparable damage to the reputation and business. Taking this in mind, we need to find the encryption that is just as graded, this needing to be something that is Ironclad and unbreakable. This leading us to our chosen method of security, SHA, the Secure Hash Algorithm. This is the industry standard for hashing, and good reason. Artemis would be utilizing this to ensure consumer and company data are all guaranteed as safe from any potential breaches. Any risks would be attached to somewhat external factors like firewall security or system information leaks. This hashing algorithm is generally viewed as the most secure and this is why we would primarily be using it.

The encryption utilizes key lengths like 128, 192, and 256 bits, with 256 bits chosen for this scenario due to its higher security against brute-force attacks. These key lengths impact the encryption's strength, making longer keys like 256 bits more resistant to decryption attempts. The 256-bit level refers to the binary representation of the encryption key. Random numbers enhance security by adding unpredictability. Symmetric encryption uses a single key for encryption and decryption, while asymmetric encryption employs a pair of public and private keys for secure communication.

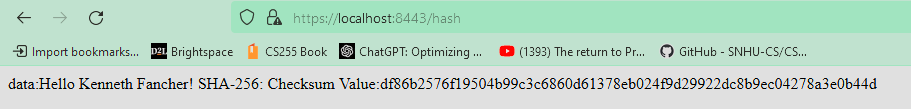
## Certificate Generation

Insert a screenshot below of the CER file.



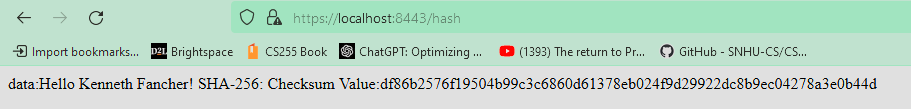
## Deploy Cipher

Insert a screenshot below of the checksum verification.



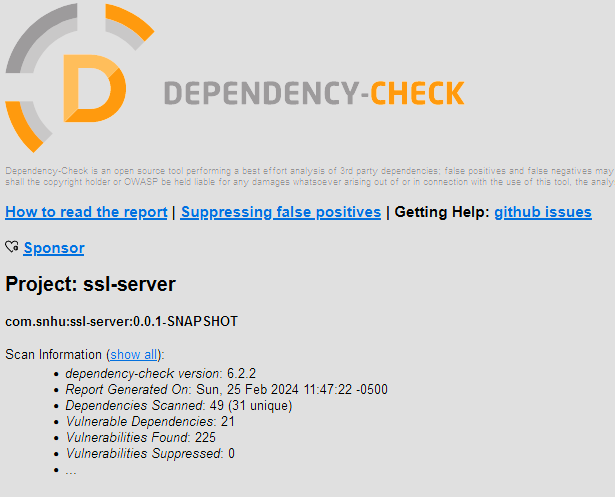
## Secure Communications

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



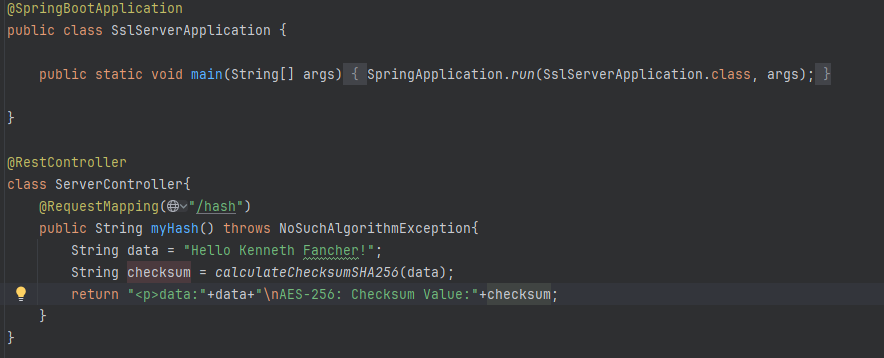
## Secondary Testing

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

In Summary, we will be moving forward with our SHA-256 hashing algorithm to protect against bad actors in the transfer data and confidentiality. This has been displayed through the above referenced images, showing proper use of SHA-256 to give a checksum verification of integrity. Our web page was certified with an in-house generated certificate, as shown by our HTTPS prefixed web page.

## 