****

# 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** | **12/10/2023** | **Dylan Cavazos** |  |

## 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

Dylan Cavazos

## Algorithm Cipher

For Artemis Financial, I recommend the Advanced Encryption Standard (AES) algorithm due to its widely used and robust cryptographic security. AES is a key block cipher that features various key sizes ranging from 128, 192, and 256 bits. AES has become the overall standard for encryption against cryptographic attacks and has even replaced the Data Encryption Standard since 2001 (GeeksforGeeks, 2020b).

Some of the major characteristics of AES is that can be utilized for both software and hardware, ensures high security due to its multi-bit implementation, and it’s also open source making it freely available for any company to utilize its secure algorithm. While it features many pivotal characteristics for hash algorithms, the AES does have some downsides, including that it can be difficult to implement in a software setting, and that the higher the bit used, the greater the toll on performance. This can be factored by companies when choosing what is the best hash algorithm to utilize for their software. In this case, I think AES is ideal for Artemis Financial, as protection and security are number one priority over performance of major applications.

AES uses a random number generator not implicitly generated through its own function but relies on an external source for its use of random number generation. Although this isn’t necessarily ideal the security measurement isn’t directly impacted. AES is also a symmetric key algorithm, so it uses the same key for both encryption and decryption.

Regarding the history of the AES, it was established by the U.S. National Institute of Standards and Technology (NIST) in 2001, immediately replacing the previous standard of Data Encryption Standard (DES). The current state of AES is that its vastly considered to be an incredibly reliable and robust security encryption algorithm that still can withhold against conventional cryptographic attacks and is often recommended as the ideal encryption standard throughout various institutions and regulatory entities.

## Certificate Generation

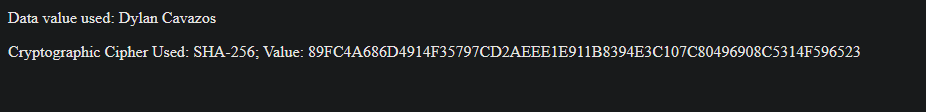
Insert a screenshot below of the CER file.

A screenshot of a computer

Description automatically generated

## Deploy Cipher

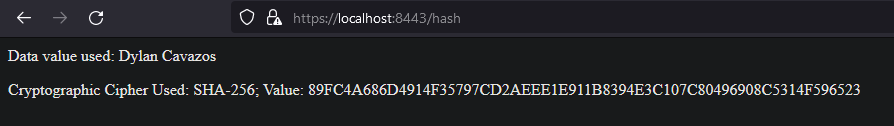
Insert a screenshot below of the checksum verification.



## Secure Communications

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

Although the certificate is self-signed, I was able to successfully connect to a secure website using HTTPS protocol.



**## need to add server. entries to enable HTTPS with SSL keystore, replace "????" with correct entries**

**server.port=8443**

**server.ssl.key-store=file:/C:/Users/dcava/OneDrive%20-%20SNHU/Documents/SCHOOL/2023/Software%20Security/Module%20Seven/ssl-server\_student/src/main/resources/myKeystore.jks**

**server.ssl.key-store-password=xxxxx**

**server.ssl.key-password=xxxxxx**

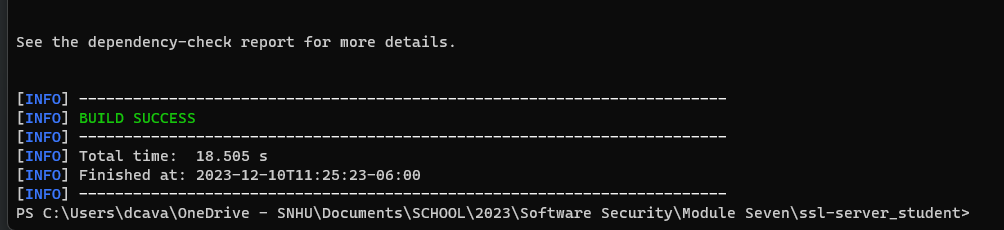
**server.ssl.keyStoreType=JKS**

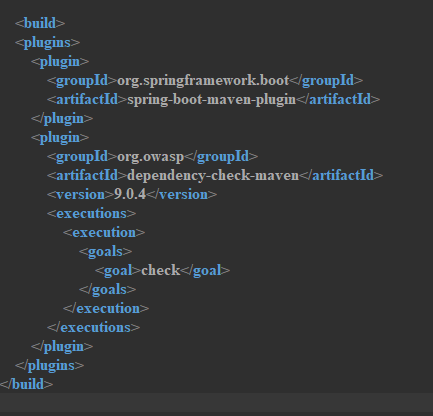
**server.ssl.keyAlias=myalias**

**logging.level.root=DEBUG**

## Secondary Testing

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





A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

## Functional Testing

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

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

## Summary

The refactored code focused primarily on addressing the potential vulnerabilities related to the generation of cryptographic hashes. To combat this, I implemented the use of the SHA-256 algorithm for hashing while also adding conversion for bytes to hexadecimal format. These illustrated best practices that align to secure principles for cryptographic operations. Relating to the vulnerability process flow diagram, the refactored code helps illustrate secure code quality, secure error handling, secure encryption uses and vulnerabilities, and secure input validation and API interaction. The vulnerability process flow diagram was a helpful tool in properly assessing the secure methods to implement into the functionality of the program while adhering to secure programming principles.

## Industry Standard Best Practices

Regarding industry standard best practices, the SHA-256 algorithm for cryptographic hashing is considered an industry best practice as it’s a widely used robust security measure used for data integrity and verification. By implementing best practices such as the SHA-256 and exception handling, the code will help in mitigating known vulnerabilities, cryptographic attacks, and vulnerabilities related to hash generation. For Artemis Financial, the use of an AES like the SHA-256 will help in maintaining the user’s privacy information while also enhancing the reputation of the company to the users and stakeholders.

GeeksforGeeks. (2020b, April 30). *AES full form*. https://www.geeksforgeeks.org/aes-full-form/