****

Haley Moyer

# CS 305 Project Two

**Practices for Secure Software Report**

Table of Contents

[Document Revision History 3](#_Toc33111302)

[Client 3](#_Toc33111303)

[Instructions 3](#_Toc33111304)

[Developer 4](#_Toc33111305)

[1. Algorithm Cipher 4](#_Toc33111306)

[2. Certificate Generation 4](#_Toc33111307)

[3. Deploy Cipher 4](#_Toc33111308)

[4. Secure Communications 4](#_Toc33111309)

[5. Secondary Testing 4](#_Toc33111310)

[6. Functional Testing 5](#_Toc33111311)

[7. Summary 5](#_Toc33111312)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **4/24/2022** | **Haley Moyer** |  |

## Client



## Instructions

Deliver this completed Practices for Secure Software Report documenting your process for writing secure communications and refactoring code that complies with software security testing protocols.

Respond to the steps outlined below and replace the bracketed text with your findings in your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

Haley Moyer

## 1. Algorithm Cipher

Determine an appropriate encryption algorithm cipher to deploy given the security vulnerabilities, justifying your reasoning. Be sure to address the following:

* Provide a brief, high-level overview of the encryption algorithm cipher.
* Discuss the hash functions and bit levels of the cipher.
* Explain the use of random numbers, symmetric vs non-symmetric keys, and so on.
* Describe the history and current state of encryption algorithms.

When reviewing the needs of Artemis Financial, the most appropriate algorithm cipher to deploy would be the Advanced Encryption Standard (AES) using a 128-bit encryption key. This algorithm is considered to be the most secure and is used by the US government to protect classified information. It is considered safe against brute force attacks due to how large the encryption key is. The 256-bit encryption key is more secure, as it encrypts information to 256-bits, but it takes a lot more processing power to run and will slow down applications, especially smaller ones. (Bernstein & Cobb, 2021) This encryption algorithm also complies with Federal Information Processing Standards (FIPS) which are required when working with a government agency. (Compliance faqs: Federal Information Processing Standards (FIPS) 2021)

In the past, DES was the standard for encryption, and it is still used today. This would not be a good fit for Artemis Financial since it only uses a 56-bit encryption key and is much more susceptible to brute force attacks by hackers. Hash functions ensure that the encrypted data is completely different than the original data, and the encrypted data cannot easily be traced back to the original. The bit length of an encryption algorithm determines the length of the encrypted data, therefore a key with a larger bitrate would be more secure. Using a 128-bit key, it is nearly impossible to gain the key with brute force using today’s computing power. (Bernstein & Cobb, 2021)

Another aspect of encryption to consider is whether to use a symmetric or asymmetric key. With a symmetric key, the same key is used to encrypt and decrypt data. This is the method that is used for large amounts of information, because only one private key is needed. Asymmetric encryption uses a public key to encrypt data, then a private key is needed to decrypt the data. This is generally used for only small amounts of data, because two keys are required, and if just one key is lost, then it is impossible to get the data back. With asymmetric encryption, a key would also need to be sent to the party who is receiving the encrypted data, so that would need to be done securely as well. Keeping these things in mind, it would be best for Artemis Financial to use symmetric encryption. (Mailfence Team, 2021)

## 2. Certificate Generation

Generate appropriate self-signed certificates using the Java Keytool, which is used through the command line.

* To demonstrate that the keys were effectively generated, export your certificates (CER file) and submit a screenshot of the CER file below.

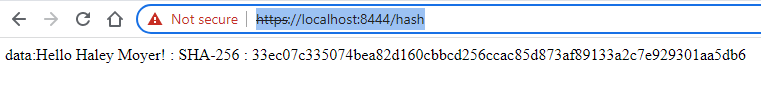
Text

Description automatically generated

## 3. Deploy Cipher

Refactor the code and use security libraries to deploy and implement the encryption algorithm cipher to the software application. Verify this additional functionality with a checksum.

* Insert a screenshot below of the checksum verification. The screenshot must show your name and a unique data string that has been created.



## 4. Secure Communications

Refactor the code to convert HTTP to the HTTPS protocol. Compile and run the refactored code to verify secure communication by typing **https://localhost:8443/hash** in a new browser window to demonstrate that the secure communication works successfully.

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

[Insert screenshot(s) here.]

## 5. Secondary Testing

Complete a secondary static testing of the refactored code using the dependency check tool to ensure code complies with software security enhancements. You only need to focus on the code you have added as part of the refactoring. Complete the dependency check and review the output to ensure you did not introduce additional security vulnerabilities.

* Include the following below:
  + A screenshot of the refactored code executed without errors
  + A screenshot of the dependency check report

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## 6. Functional Testing

Identify syntactical, logical, and security vulnerabilities for the software application by manually reviewing code.

* Complete this functional testing and include a screenshot below of the refactored code executed without errors.

Graphical user interface, text

Description automatically generated

## 7. Summary

Discuss how the code has been refactored and how it complies with security testing protocols. Be sure to address the following:

* Refer to the Vulnerability Assessment Process Flow Diagram and highlight the areas of security that you addressed by refactoring the code.
* Discuss your process for adding layers of security to the software application and the value that security adds to the company’s overall wellbeing.
* Point out best practices for maintaining the current security of the software application to your customer.

In adding more security to the program, I have added an encryption algorithm and checksum verification. I have also generated a security certificate and ensured that there were no security vulnerabilities with a maven dependency check. Security is important for the company’s overall wellbeing because people will not trust a company, especially a financial one, without them knowing that their information will be secure. To maintain the current security, it is imperative to ensure that all the dependencies used are up to date and functioning properly, as well as regularly running security checks.

Citations

Bernstein, C., & Cobb, M. (2021, September 24). *What is the Advanced Encryption Standard (AES)?* SearchSecurity. Retrieved April 3, 2022, from <https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard>

*Compliance faqs: Federal Information Processing Standards (FIPS)*. NIST. (2021, July 8). Retrieved April 3, 2022, from https://www.nist.gov/standardsgov/compliance-faqs-federal-information-processing-standards-fips#:~:text=are%20FIPS%20developed%3F-,What%20are%20Federal%20Information%20Processing%20Standards%20(FIPS)%3F,by%20the%20Secretary%20of%20Commerce.

Mailfence Team. (2021, March 17). *Symmetric vs asymmetric encryption: What's the difference?* Mailfence Blog. Retrieved April 24, 2022, from https://blog.mailfence.com/symmetric-vs-asymmetric-encryption/#:~:text=Symmetric%20encryption%20uses%20the%20same,private%20key%20to%20decrypt%20information.