****

# 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** | **2024-04-20** | **John Vowcicefski** | **Initial document crreation** |

## 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

John Vowcicefski

## Algorithm Cipher

For the financial data and the necessity for great security protocols, my recommendation for Artemis Financials’ encryption needs is the Advanced Encryption Standard (AES) cipher:

* **Overview of AES**: The AES cipher is a symmetric block cipher chosen by the U.S. government to protect classified information. It is widely used across the globe due to its strength and efficiency in securing digital data.
* **Hash Functions and Bit Levels**: AES supports a lot of key lengths—128, 192, and 256 bits—with the 256-bit key size being the most secure option. It is often used in conjunction with secure hash algorithms like SHA-256, which provides a secure way to maintain the integrity of the data by generating a unique hash value.
* **Symmetric Keys and Randomness**: AES uses symmetric keys, meaning the same key is applied for both encrypting and decrypting information. The keys are derived using secure random number generators, which are critical to prevent attackers from guessing or predicting the key.
* **Historical Context and Current Relevance**: AES was established as an encryption standard by the National Institute of Standards and Technology (NIST) in 2001. Since then, it has been subjected to intense scrutiny and has withstood all known forms of cryptanalytic attacks. Today, AES is still considered secure and is used in various security protocols, including Secure Sockets Layer (SSL)/Transport Layer Security (TLS), which safeguard internet communications.
* **Implementation in Artemis Financial**: Deploying AES within Artemis Financial will bolster our security infrastructure, ensuring that all data in transit and at rest is encrypted with a tried-and-tested algorithm. Its fast encryption speeds and low overhead also mean that the performance impact on Artemis Financial's systems will be minimal, maintaining efficiency in client transactions and data processing.

The choice of Advanced Encryption Standard (AES) for Artemis Financial is based on its proven security strength and efficiency, making it ideal for protecting sensitive financial data. AES is endorsed by the National Institute of Standards and Technology (NIST) and is the de facto encryption standard used globally. It offers a resistance to a variety of attacks, including all known brute-force, cryptanalytic, and side-channel attacks, ensuring that financial transactions and personal client data remain confidential and secure.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

keytool -genkeypair -alias sslserver -keyalg RSA -keysize 2048 -storetype PKCS12 -keystore "C:\Users\vowci\eclipse-workspace\CS 305 Project Two Code Base\ssl-server\_student\src\main\resources\keystore.p12" -validity 3650 -storepass 123456 -dname "CN=John.Vowcicefski, OU=SNHU, O=SNHU, L=Philly, ST=PA, C=US"

keytool -list -keystore "C:\Users\vowci\eclipse-workspace\CS 305 Project Two Code Base\ssl-server\_student\src\main\resources\keystore.p12" -storepass 123456

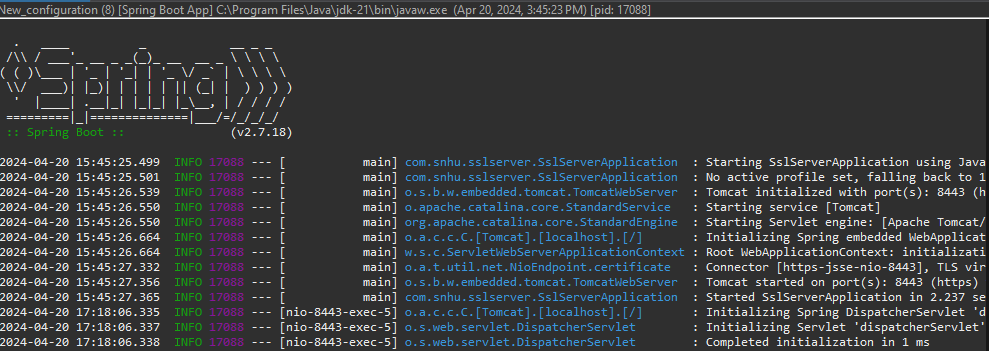
Self-signed certificates play a crucial role in setting up a secure environment within internal systems by establishing a rudimentary trust layer. They are particularly useful in closed ecosystems where external trust is not a requirement. For Artemis Financial, using these certificates helps secure internal communications and data transfers without relying on external certificate authorities. However, it's important to note their limitations, such as the lack of validation by third-party authorities, which might make them less suitable for public-facing applications without additional trust measures.

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screen shot of a computer program

Description automatically generated



## Secure Communications

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

A screenshot of a computer

Description automatically generated

Using HTTPS is essential for keeping data safe as it moves between the user’s device and the server. This secure way of sending data helps prevent outsiders from sneaking a peek at what’s being sent or altering the data before it reaches its destination. At Artemis Financial, HTTPS helps protect customers' financial details during transactions, showing the company's dedication to keeping personal and financial information safe. This protection is possible through SSL/TLS protocols, which scramble the data in a way that only the intended receiver can understand.

## 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

Description automatically generated

## Functional Testing

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

A screenshot of a computer

Description automatically generated

## Summary

I made several important security improvements to Artemis Financial's application. These changes include:

* **AES Encryption**: I added AES encryption to protect the data. AES is a powerful encryption method that helps keep sensitive financial information safe from unauthorized access.
* **Certificate Management**: I created and added a self-signed SSL certificate to the system. This helps make sure that all data sent over the network is encrypted and secure.
* **HTTPS Enforcement**: I set up the application to only use HTTPS, which protects all data transfers from being intercepted or tampered with.

## Industry Standard Best Practices

To strengthen the security setup for Artemis Financial, I used these key best practices:

* **AES Encryption**: AES is a security method recommended by NIST. It’s designed to protect data that’s sensitive but not classified. By using AES, especially the most secure version, AES-256, Artemis Financial ensures that their data handling meets high security standards. This is crucial for tasks that involve sensitive financial information.
* **HTTPS and SSL/TLS Configurations:** Artemis Financial follows the recommended guidelines by OWASP, a leading body on web application security, by using HTTPS. This ensures that all data sent over their networks is protected against unauthorized access. Proper setup of SSL/TLS, the technology behind HTTPS, is key to making sure that financial transactions and personal data are transmitted securely. This is also important to meet financial regulations and standards for data security.
* **Dependency Checks:** Using the OWASP Dependency-Check helps Artemis Financial spot potential security risks in the software components they use. This tool checks against a comprehensive list of known security issues, helping developers fix problems before they can harm the system. This is an important part of making sure that the software is as secure as possible, safeguarding it against attacks that exploit known vulnerabilities.

References

National Institute of Standards and Technology. (2020). NIST Special Publication 800-57. Retrieved from <https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final>

Oracle. (n.d.). Keytool - Key and Certificate Management Tool. Retrieved from <https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html>

National Institute of Standards and Technology. (2001). FIPS PUB 197: The Advanced Encryption Standard. Retrieved from <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf>

file:///C:/Users/vowci/eclipse-workspace/CS%20305%20Project%20Two%20Code%20Base/ssl-server\_student/target/dependency-check-report.html