****

# 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** | **August 24, 2025** | **Yaumel Betancourt** |  |

## 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

Yaumel Betancourt

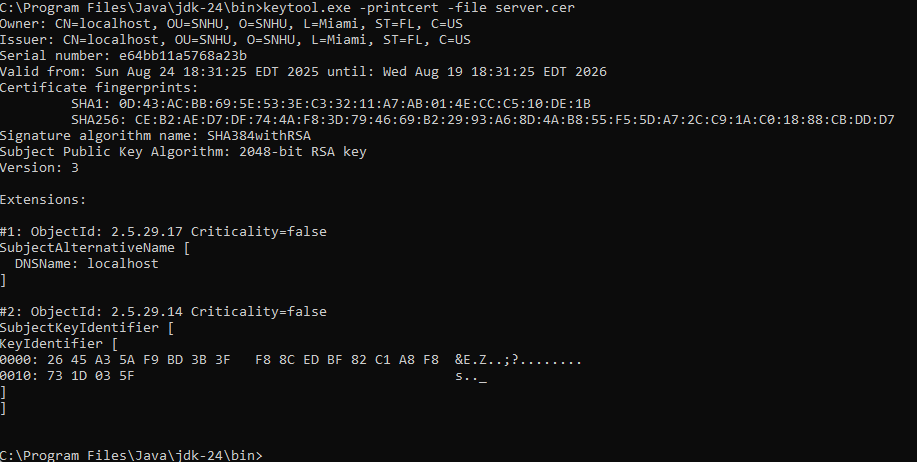
## Algorithm Cipher

Since Artemis Financial wants a security solution for archived files, then the strategy that needs to be implemented is for data at rest. The cipher I would recommend for the encryption of the archived files is AES/GCM/NoPadding. If remote access or file transport is necessary, then the cipher suite TLS\_AES\_128\_GCM\_SHA256 is an effective solution. There are several factors to consider, the first of which ensuring that strong keys are used, ideally 256-bit keys. A reliable key management system needs to be used to keep track of keys and any key rotations. Access management is another important aspect to lower the likelihood of unauthorized access. It is imperative that secure hashing is used to prevent any breaches due to predictable or reversible hashing algorithms. The purpose of the cipher’s hash functions is to take in data and produce a fixed-sized string of bytes. It is a deterministic process meaning that the same input always produces the same string of bytes. This is powerful for many applications, in this case including verifying file integrity to ensure no unauthorized changes were made and that only authorized users can decrypt the contents of the file. The bit level generally refers to the key size. The larger the key, the more secure the encryption. In this case the size is 256 bits. The idea behind random numbers is to create a truly random seed for key generation so they are truly unique and irreversible. When it comes to symmetric and asymmetric keys, the difference lies in the encryption/decryption process. With symmetric keys, one key is responsible for encrypting and decrypting. With asymmetric keys, there is a public key used for encrypting and a private key for decrypting. Symmetric encryption is faster and is ideal for encrypting data in bulk, which is the case in this scenario. This is the reason AES is being used.

The history of cryptography is a fascinating one primarily since it was a crucial part of human history from the Caesar cipher during the Roman republic to the Engima machine during World War II. The current state of cryptography includes hundreds if not thousands of algorithms and cipher suites to encrypt all kinds of data such as text files, images, etc. Algorithms have been researched and tested to ensure the strongest and safest approaches to security for many industries that rely on the safe storage and transport of sensitive data. The future looks promising, but equally troubling, with the advancements being made in quantum computing. Although it offers more secure possibilities, it also breaks current algorithms since the vast improvement in computational power and efficiency means it is easier to figure out hashes and decrypt data. It is important to note that measures are being taken by NIST to counter this.

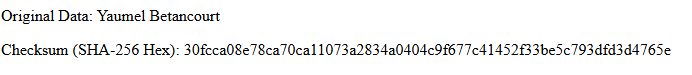
## Certificate Generation

Insert a screenshot below of the CER file.



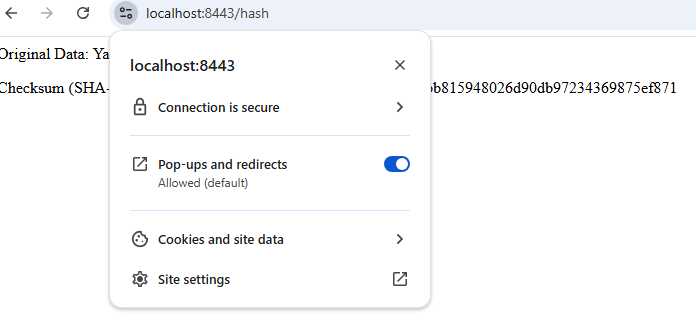
## Deploy Cipher

Insert a screenshot below of the checksum verification.



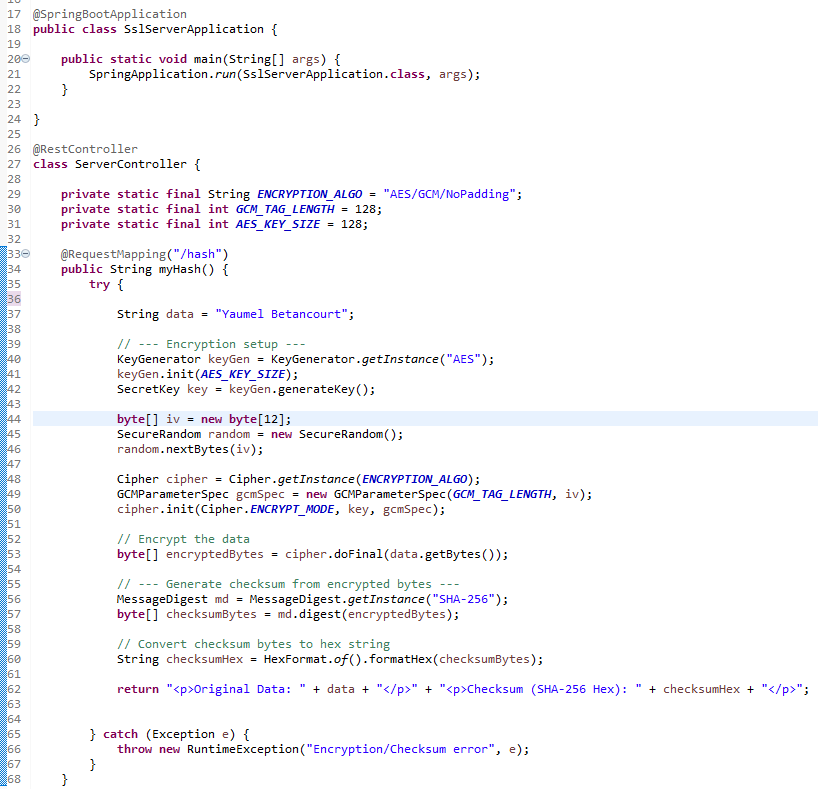
## Secure Communications

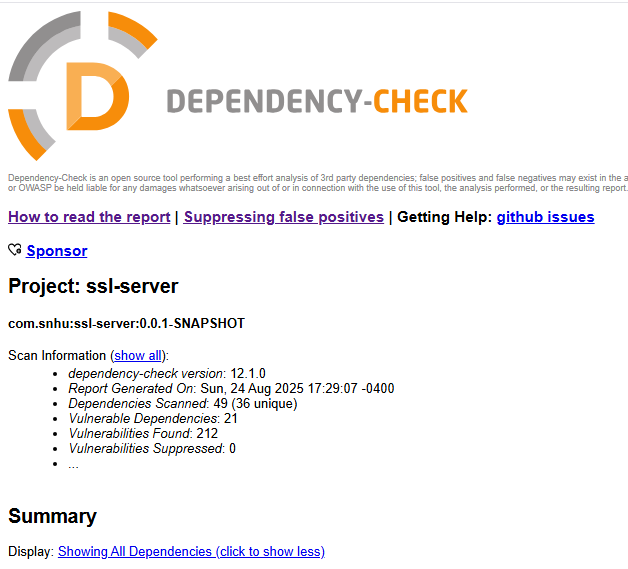
Insert a screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

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

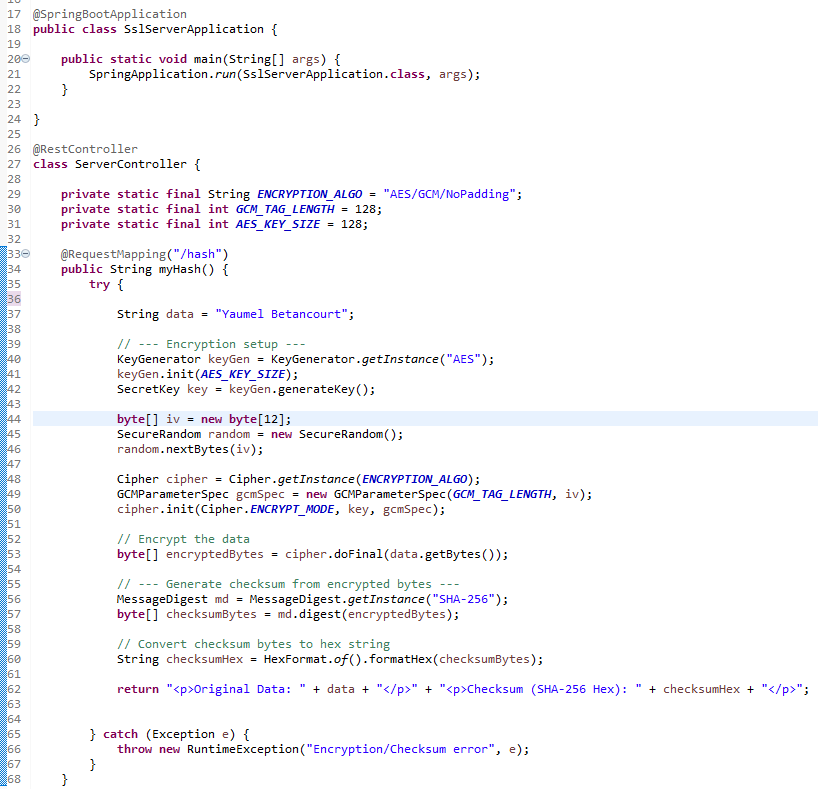


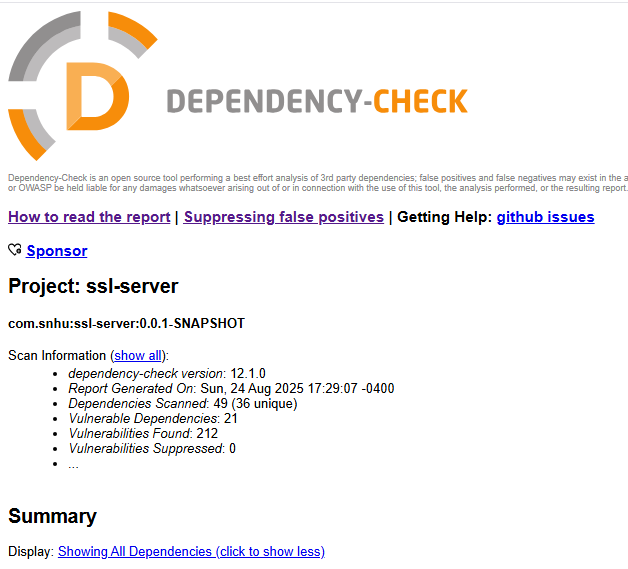


It is important to note that the vulnerabilities are mostly related to version changes which can be changed even though it seems as if there are an overwhelming number of them. Many can even be suppressed if necessary.

## Functional Testing

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





The vulnerability, in this case data integrity, has been addressed via the use of the cipher suite in the method used when the hash endpoint is called. Since the current implementation being used uses a hard-coded string, there are no risks of injections or dangerous external code execution. Should there be a need to use a more dynamic approach where a string can be taken from a user, then the proper steps must be taken to sanitize data to avoid certain attacks.

## Summary

The code has been refactored to use the TLS\_AES\_128\_GCM\_SHA256 cipher suite when calling the hash endpoint. It works by taking a data string in the method. AS the name implies it uses AES with a key size of 128 bits to allow for symmetric encryption. This suite is used for both encryption and authentication while also producing a unique 256-bit hash to ensure integrity of messages between server and client while also being used to establish the initial connection between the parties involved.

The major benefit of the addition of this suite is it makes the communication secure so sensitive information, such as file transfers in this case, cannot be hijacked or altered by a nefarious third-party causing data leaks which can result in private company, or user, information being released to the public. In addition, self-signed certificates were produced to make use of https to add yet another layer to ensure data integrity and authentication of parties, and combat against spoofing or hijacking attacks. The refactor covers several areas that are susceptible to security vulnerabilities starting with client server interactions. This solution makes sure that both parties verify who they are and that the data shared between them is not visible to outside entities attempting to eavesdrop. Secure error handling was used in case there were any errors when encrypting or creating a hash without releasing sensitive data in the message that hacker could use to try to decrypt messages. Using the cipher suite, a secure random number was used to make it practically impossible to figure out the encryption and gain access to sensitive data. In this case the base string is hardcoded, but in the future input validation can be used to sanitize a string if a more dynamic approach is preferred.

## Industry Standard Best Practices

The use of the cipher suite and the certificates is considered best practice since it offers robust security methods that have been tried and tested to be used together to ensure both authentication and data integrity. Both are necessary for the application to be truly secure. The cipher suite helps secure the data so both parties agree on how to encrypt and decrypt data when communicating as well as establishing the initial connection. However, this does not prevent attacks such as a man in the middle attack where a bad actor can fool both sides and established separate connections with each party intercepting communications. This is where the certificate is paramount to avoid such attacks because they authenticate that the public key received is legitimate and not an outside party seeking to trick the client and steal data. Following these best practices is paramount to the success and safety of Artemis Financial since the files will contain sensitive data. The nature of the business requires that authentication and data integrity remain central to the application’s development.