****

# Practices for Secure Software Report

Table of Contents

[Practices for Secure Software Report 1](#_Toc211697573)

[Document Revision History 3](#_Toc211697574)

[Client 3](#_Toc211697575)

[Instructions 3](#_Toc211697576)

[Developer 4](#_Toc211697577)

[1. Algorithm Cipher 4](#_Toc211697578)

[2. Certificate Generation 4](#_Toc211697579)

[3. Deploy Cipher 4](#_Toc211697580)

[4. Secure Communications 4](#_Toc211697581)

[5. Secondary Testing 5](#_Toc211697582)

[6. Functional Testing 6](#_Toc211697583)

[7. Summary 6](#_Toc211697584)

[8. Industry Standard Best Practices 6](#_Toc211697585)

[9. References 6](#_Toc211697586)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **10/18/2025** | **Michael Peck** | **Initial Revision** |

## 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

Michael R Peck Jr

## Algorithm Cipher

Based on new emerging technologies, the best option for an algorithm cipher would be the Advanced Encryption Standard (AES). It has long been considered the gold standard for ciphers in recent years. The AES is a symmetric style cipher which uses the same key for decrypting and encrypting, and comes in varieties such as 128-, 192-, and 256-bit key sizes (Porter, 2025). It strikes a balance between speed and strength, making it suitable for diverse applications, from secure communications to cloud storage and data backups. (Splashtop, 2024). The reason that AES is also recommended is that it has a collision-proof system in place. By that I mean that each “block” of data (size is determined by the bit-level) that is being decrypted or encrypted must be unique, even if the data is identical in both blocks, by passing the output to an XOR (exclusive or function),

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen with text

AI-generated content may be incorrect.

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A close-up of a computer screen

AI-generated content may be incorrect.

## Secure Communications

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

A screenshot of a computer

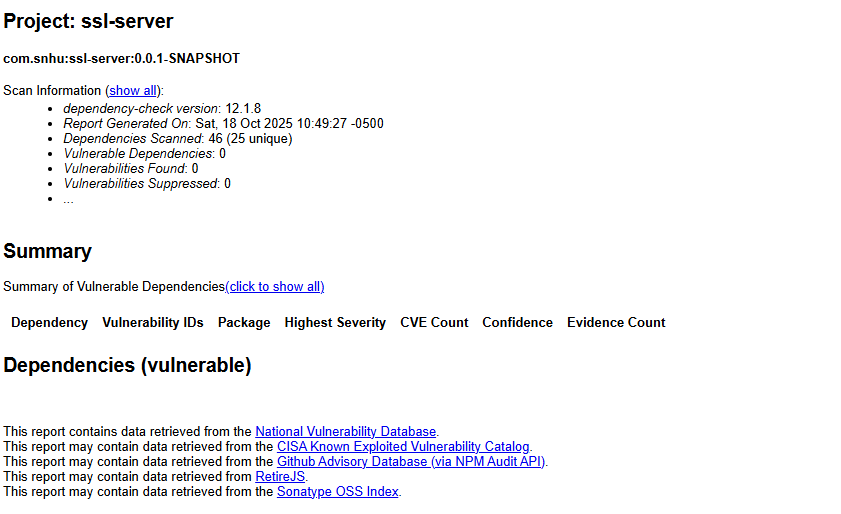
AI-generated content may be incorrect.

## Secondary Testing

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

A screenshot of a computer screen

AI-generated content may be incorrect.



## Functional Testing

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

To analyze the syntactical, logical or security vulnerabilities, each section should be analyzed piece by piece and then as a module, then a whole. Luckily, the amount of code performed during this project was minimal so each module can get extra attention so no known or common vulnerabilities would go unnoticed. Syntactically speaking, the main Java class has no errors that the IDE could catch, and all of the keywords correspond correctly with the logic, based on the results we received from the “/hash” mapping screenshot above. There is no user input required yet, so no need to validate any user input, other than to validate and test what is hard-coded or created during runtime, including the hash function. Logically speaking, there are potentially other better ways to create the hash function, including using a cloud service API or other API to handle Certification Authentication, etc., rather than hard coding it and it’s not good practice to have a local certificate attached to the project other than to test the checksum. Security speaking, it is also not good practice to keep the certificate next to the project without putting it behind a secure server, or through a secure API from a CA, but for testing it can do its job, but it ideally will be removed before deployment. Before this project was finished, getting it to run without errors required updating out-of-date libraries, and thus in the end, several, if not all of the known vulnerabilities have been cleared, according to the dependency report.

## Summary

If we are to assess the vulnerability of this project based on the vulnerability assessment process flow, we can start with the first step, which is input validation. Currently, this project has no need for input validation, as all of the variables are hard-coded or processed locally, so no user input is expected and thus do not need validation. APIs are not included in this project but can be used to handle CA authentication since we do not want to be using a local certificate for security purposes, and Artemis Financial does not want to be responsible for handling old or outdated certificate handling. Moving on to cryptography where we get into the actual encryption algorithm, where this project uses the current gold standard of AES with the SHA 256-bit hash function. The project uses a locally stored certificate for testing purposes only and will be removed before deployment. The client server model is represented by having the client request the mapping for the site and therefore it’s algorithm processing module. None of the algorithm is done by the client and therefore the hash function is processed by the server, behind more secure software and firewalls. Code Error follows up and even though the current build is error-free, the project has no exception handling currently, but there are conditionals that do prevent unauthorized behaviors, including if-else’s and try/catch blocks. Code Quality comes up next, and each module is broken up into smaller chunks that perform a single action and nothing else. Finally, encapsulation was not required for the current data that is present as all variables are localized to their individual modules or passed as parameters when called from other functions. There are no variables that need to be shared among multiple modules that need to be protected behind accessibility levels.

## Industry Standard Best Practices

Before I could apply the industry’s best practices for this project, I first needed to run the project to evaluate its behavior to see how it could potentially misbehave or cause issues elsewhere. The first step was to first solve the errors I was receiving, and most of them were related to the current running version of the JDK, or with the plugins for the project. The code base that was given was so out of date, that the project was designed for a much earlier version of Java. Before upgrading all of the dependencies and plugins, there were roughly a couple hundred known vulnerabilities according to the National Vulnerability Database. But when these out-of-date packages were updated to the most current version, all of the known vulnerabilities have been patched so there are no known vulnerabilities listed in the newest dependency report. By performing these updates and checking the reports for new vulnerabilities, we have provided as much protection as possible for Artemis Financial as we can, without creating new vulnerabilities.

## References

National Institute of Standards and Technology. (2025). Hash Functions | CSRC. Nist.gov. <https://csrc.nist.gov/projects/hash-functions#:~:text=Table_title:%20Security%20Strengths%20of%20Approved%20Hash%20Functions,Preimage%20Resistance%20Strength%20in%20bits:%20224%20%7C>

Porter, C. (2025, March 24). Encryption Best Practices 2025: Guide to Data Protection. Training Camp - IT Training & Certification Boot Camps. <https://trainingcamp.com/articles/encryption-best-practices-2025-complete-guide-to-data-protection-standards-and-implementation/#:~:text=AES%2D256%20remains%20the%20recommended,from%202048%20to%204096%20bits>.

Splashtop. (2024, November 12). AES Encryption: How it works, Benefits, and Use Cases. Splashtop.com. <https://www.splashtop.com/blog/aes-encryption#:~:text=AES%20is%20the%20standard%20choice,cloud%20storage%20and%20data%20backups>.