****

# 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** | **04/10/2023** | **Anthony Mink** |  |

## 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

Anthony Mink

## Algorithm Cipher

To select an appropriate encryption algorithm cipher, it is important to consider the level of security required and the type of data being encrypted. A commonly used encryption algorithm is Advanced Encryption Standard or (AES). AES is a symmetric key algorithm that uses a block cipher. AES also provides strong encryption and is widely used in various applications. AES is also supported in TLS 1.3 or Transport Layer Security, which is recommended because it is newer and better that SSL or Secure Sockets Layer.

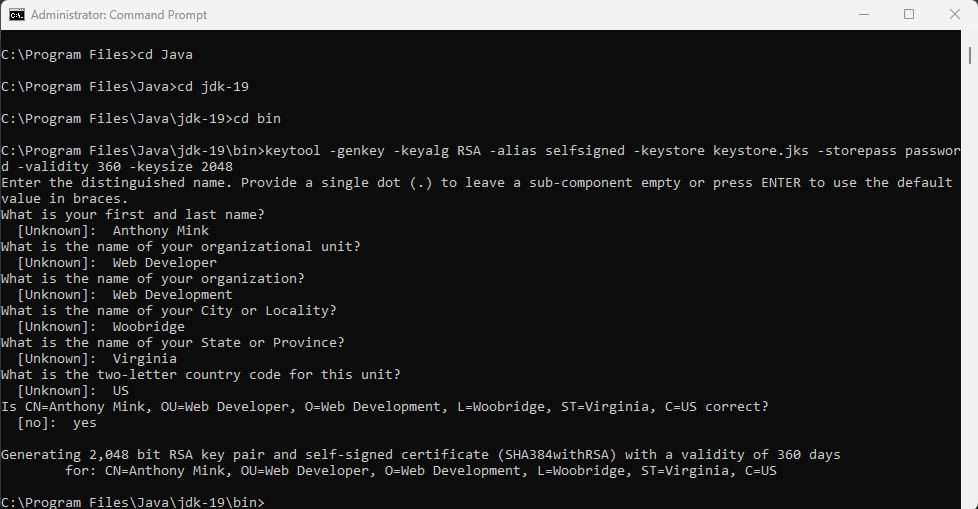
Hash functions are used in encryption algorithms to transform data into a fixed-size output. This allows them to be used as digital fingerprints or signatures for data transfers. The bit level of a cipher refers to the size of the key used for encryption. The larger the key size the more secure the encryption. AES can use several key sizes from 128 to 256, which is relatively high and can prove very secure.

Random numbers are often used to generate cryptographic keys or to add randomness to encryption algorithms. Symmetric keys are use the same key for encryption and decryption, which is what AES uses. Non-symmetric or asymmetric keys use different keys for encryption and decryption.

Encryption algorithms have been around for a long time, with various techniques used to protect information from unauthorized access. Modern encryption algorithms use complex mathematical functions and are continually evolving to provide stronger security.

## Certificate Generation

Insert a screenshot below of the CER file.



## Deploy Cipher

Insert a screenshot below of the checksum verification.

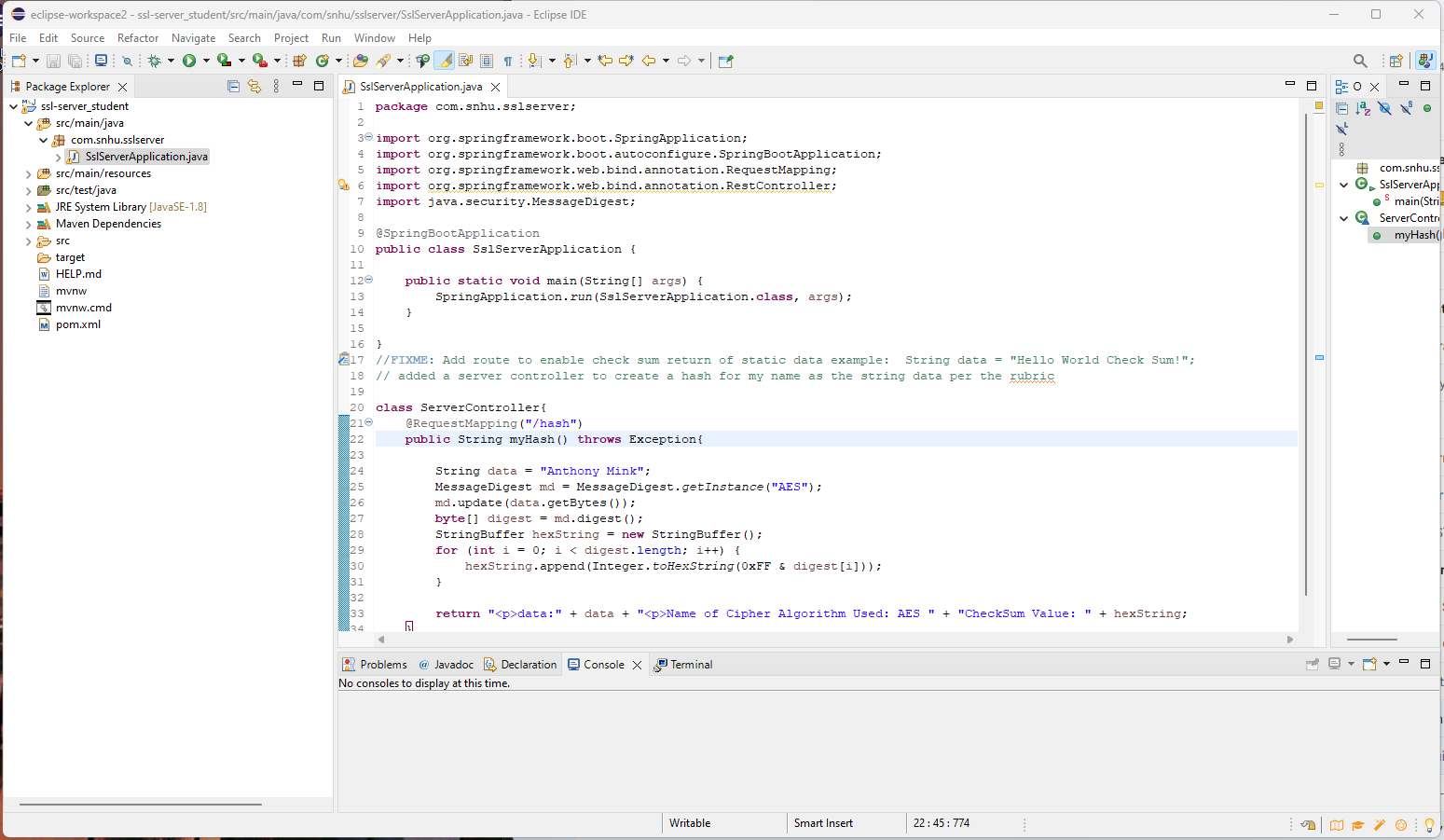
## Secure Communications

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

[Insert screenshots here.]

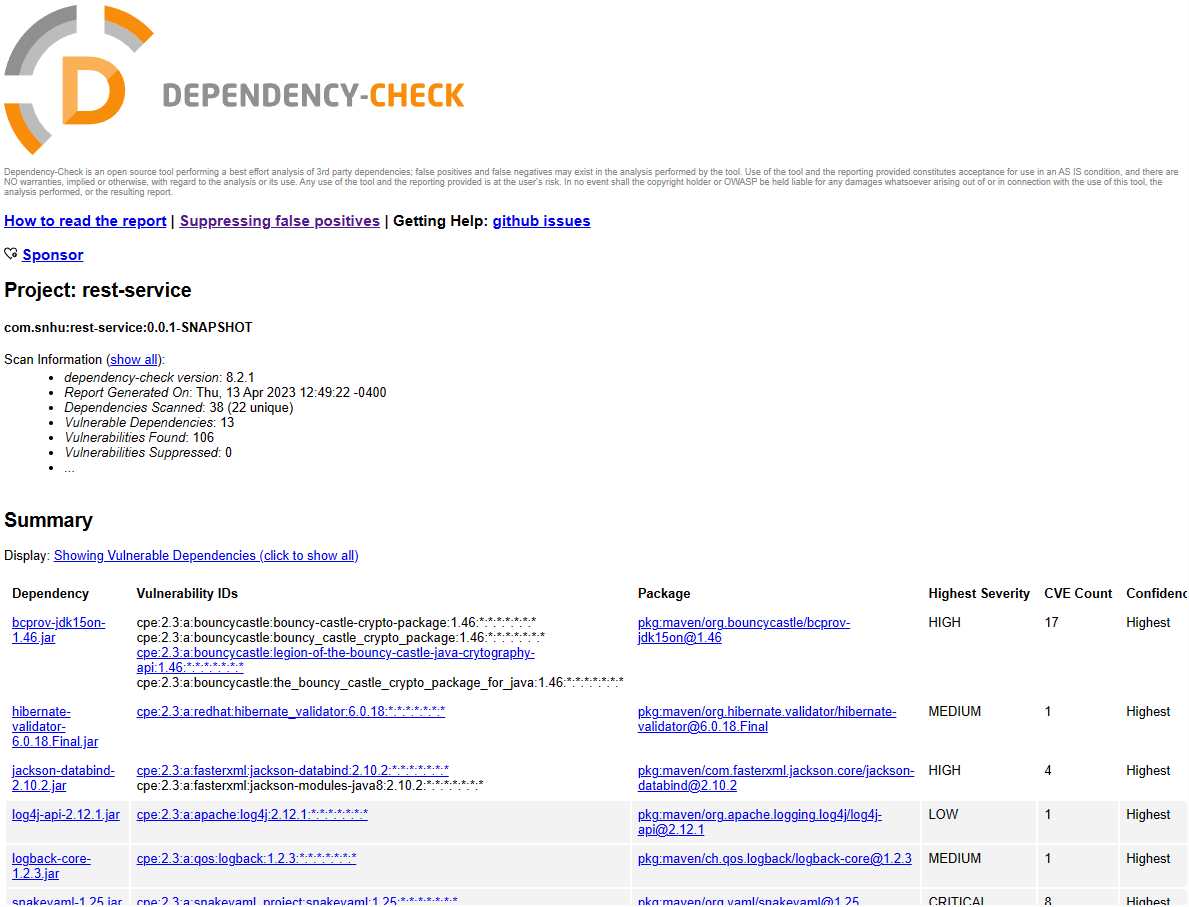
## Secondary Testing

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



## Functional Testing

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



## Summary

In this project I tried to address Artemis Financials’ security needs with cryptography, server/client, and code quality. I used AES cipher algorithm to the best of my ability to handle any coding errors that they may have had in their base code. The certificate verification as well as the checksum was to show that Artemis Financial could send messages that would be encrypted. For the project it was supposed to be just my name, but I was unable to accomplish this goal. I also tried to implement client/server needs by showing the certificate verification for their security while using their website. Finally, I was able to review the code and create a dependency check for the code that they gave, and after when I “refactored” it. Not sure if I accomplished that goal, but I believe that I was on the right track.

The end goal of what needed to be accomplished was a multi-layer form of security. The first layer would be the SSL certificate to ensure that the site is secure and that the users of the site know that their communication is encrypted. The second layer of security would be the AES encryption algorithm itself. The final layer of security would be the enforcement of HTTPS on the application instead of HTTP. This allows an encrypted session for both the user (client) and the server during transactions on the website. Poorly secured connections can allow vulnerabilities such as man-in-the-middle attacks.

## Industry Standard Best Practices

Best practices for maintaining the current security that I tried to apply to their application would involve regular dependency checks to look out for new vulnerabilities that are being discovered. Secure coding practices when adding anything additional is a must in any corporation. Other safe security practices such as input validation, solid API builds, and making sure the client knows how the security of their application works and how to do regular maintenance on it is important.

To meet the needs of Artemis Financial moving forward to achieve these best practices could be quite simple. They would need to make sure that they update the cipher algorithm as it may change in the future. Instead of AES they may have to move to another cipher such as SHA-256. Input validation can be performed by only allowing specific types of input and whitelisting available inputs rather than trying to blacklist every new threat that comes up. Regular dependency checks can highlight any new threats discovered by the NIST or any other governing bodies that show new threats that arise in the industry. A secure coding plan for the company should be implemented to make sure when they want to add anything new it will be able to coincide with the code that is in place now. Also, when they create the API make sure they create with caution so that users cannot perform any action s not explicitly authorized for their level of access. Taking these best practices into account, the application should operate smoothly for many years to come.