****

# 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** | **08/16/24** | **Tyten Perez** | **Security analysis updated** |

## 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

Tyten Perez

## Algorithm Cipher

For Artemis Financial I recommend implementing AES-256 (Advanced Encryption Standard), a widely used and trusted encryption algorithm for data security. AES was originally established in 2001 by the United States National Institute of Standards and Technology (NIST). The algorithm is based on the Rijndael cipher, which was developed by Joan Daemon and Vincent Rijmen, two cryptographers. When implemented correctly it is virtually uncrackable as it comprehensively scrambles data requiring a secret key to unscramble it. Additionally, brute forcing the key is unfeasible as it creates billions upon billions of possible combinations given its 256-bit length. This level of security is why it is used by governments and financial institutions, making it a perfect fit for Artemis Financial.

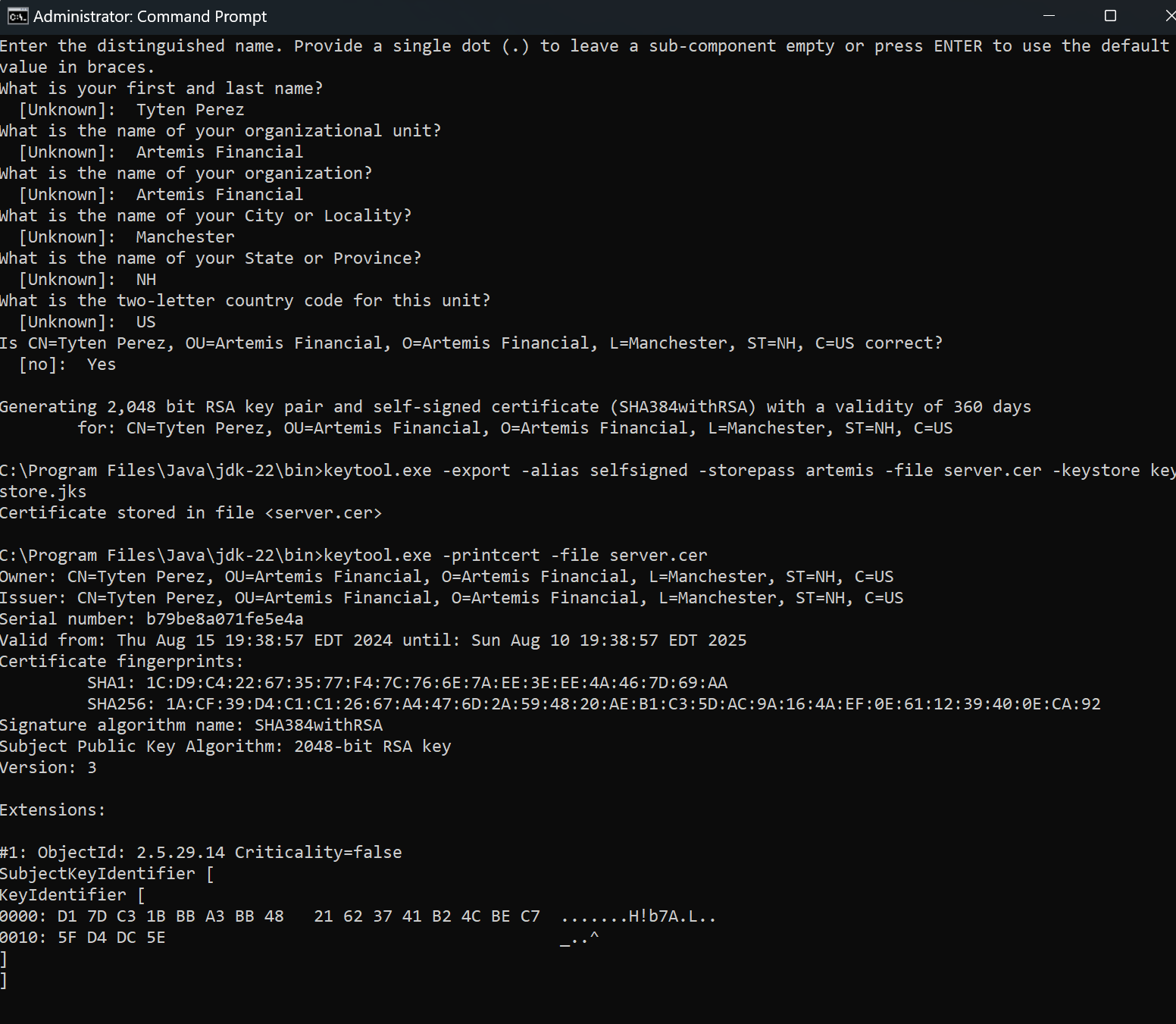
AES uses a symmetric key system, meaning that the same key is used for encryption and decryption of data. This requires that this key be shared securely when encrypting/decrypting to be effective. Asymmetric encryption, on the other hand, utilizes a pair of keys where one is a public key, and another is a private key. The public key can be shared to encrypt data while the private key is kept secret to securely decrypt the data. This asymmetric system is considered more secure but has the drawback of being slower than symmetric methods due to its increased computational needs, especially with larger volumes of data. This makes it better suited to facilitating secure communications and key exchanges than it would be with the large amounts of data encryption needed of Artemis. Random numbers are used in generating these keys in order to mitigate the risk of brute-forcing attacks. Random numbers are also used for an additional layer of security for the encrypted data. While AES-256 focuses on encryption/decryption of data, hash functions can help this process by verifying data integrity. This means that it can help reveal data alterations or invalid digital signatures, further securing Artemis Financial from unauthorized changes in its data.

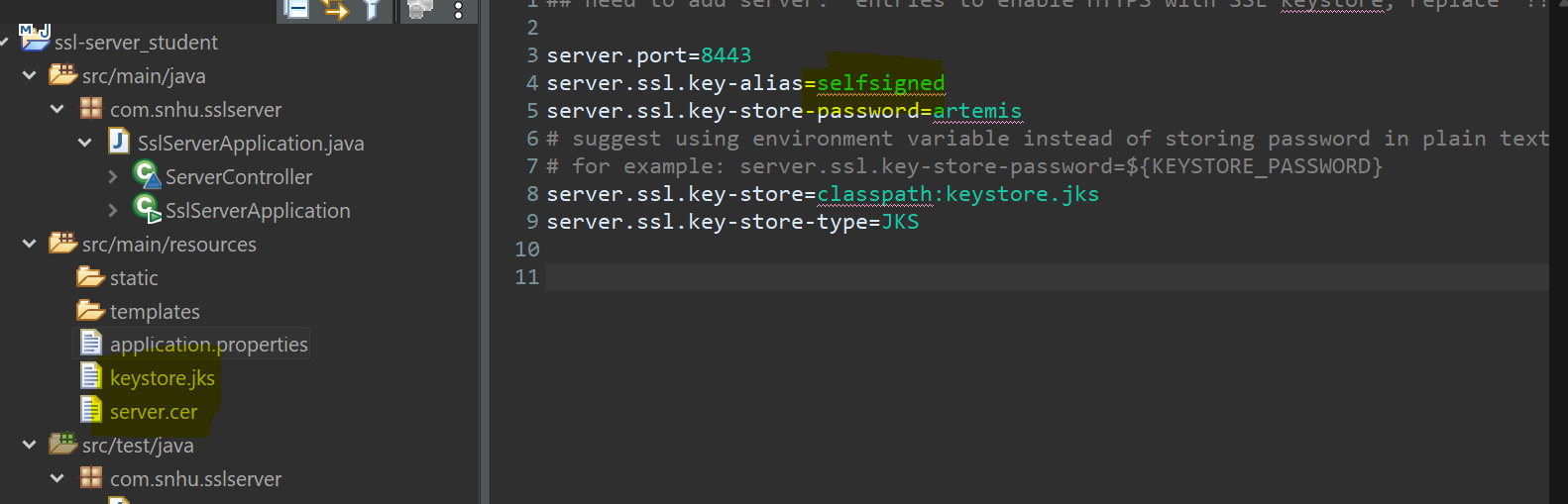
To properly integrate AES-256 and defend against security attacks, there are several practices that should be followed. For AES to be effective, it is crucial to safely manage all encryption keys, ideally using a dedicated key management service to store them. It is also important not to store this information anywhere else within the codebase, such as in a hard-coded value. Cryptographic dependencies and libraries should also be updated/patched regularly to protect against known vulnerabilities. Lastly, consistently performing automated security tests, audits and penetration tests help find security flaws before it is too late.

There are however certain considerations when using something like AES encryption. As mentioned before, if the key is compromised the encrypted data is at risk. Additionally, there is a performance cost to consider as encryption and decryption can be resource-intensive and may take time depending on the size of the data. From a regulatory standpoint, the data encrypted should comply with government standards such as GPDR in Europe, GLBA in United States, and PIPEDA in Canada.

## Certificate Generation

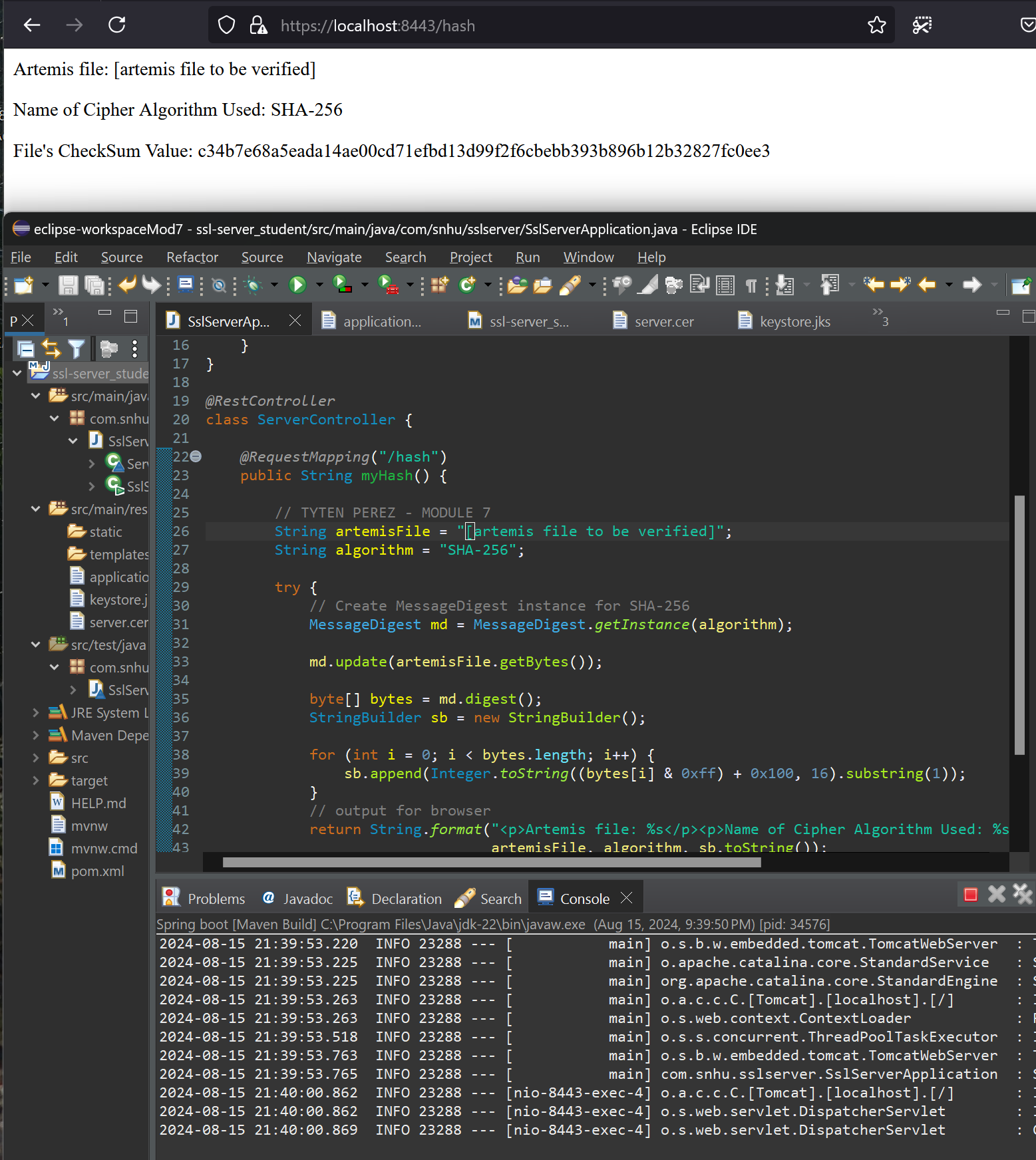
Screenshot below of the CER file.





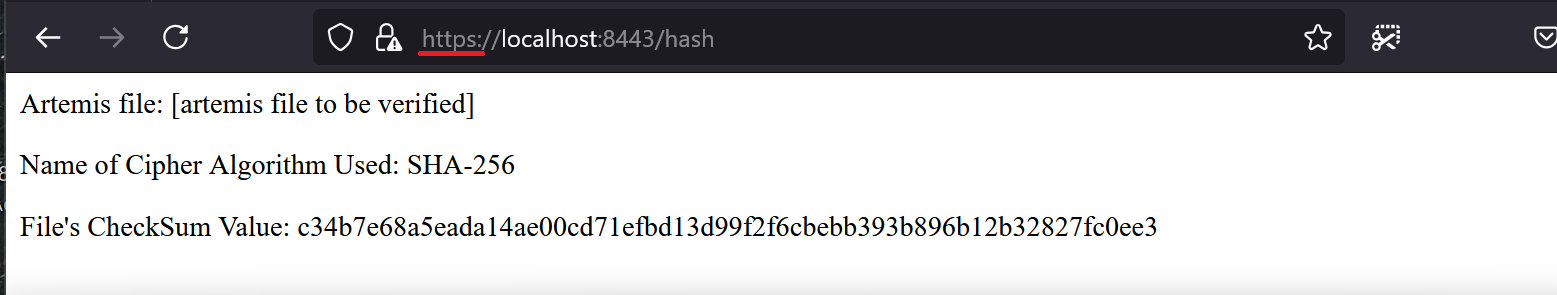
## Deploy Cipher

Screenshot below of the checksum verification.



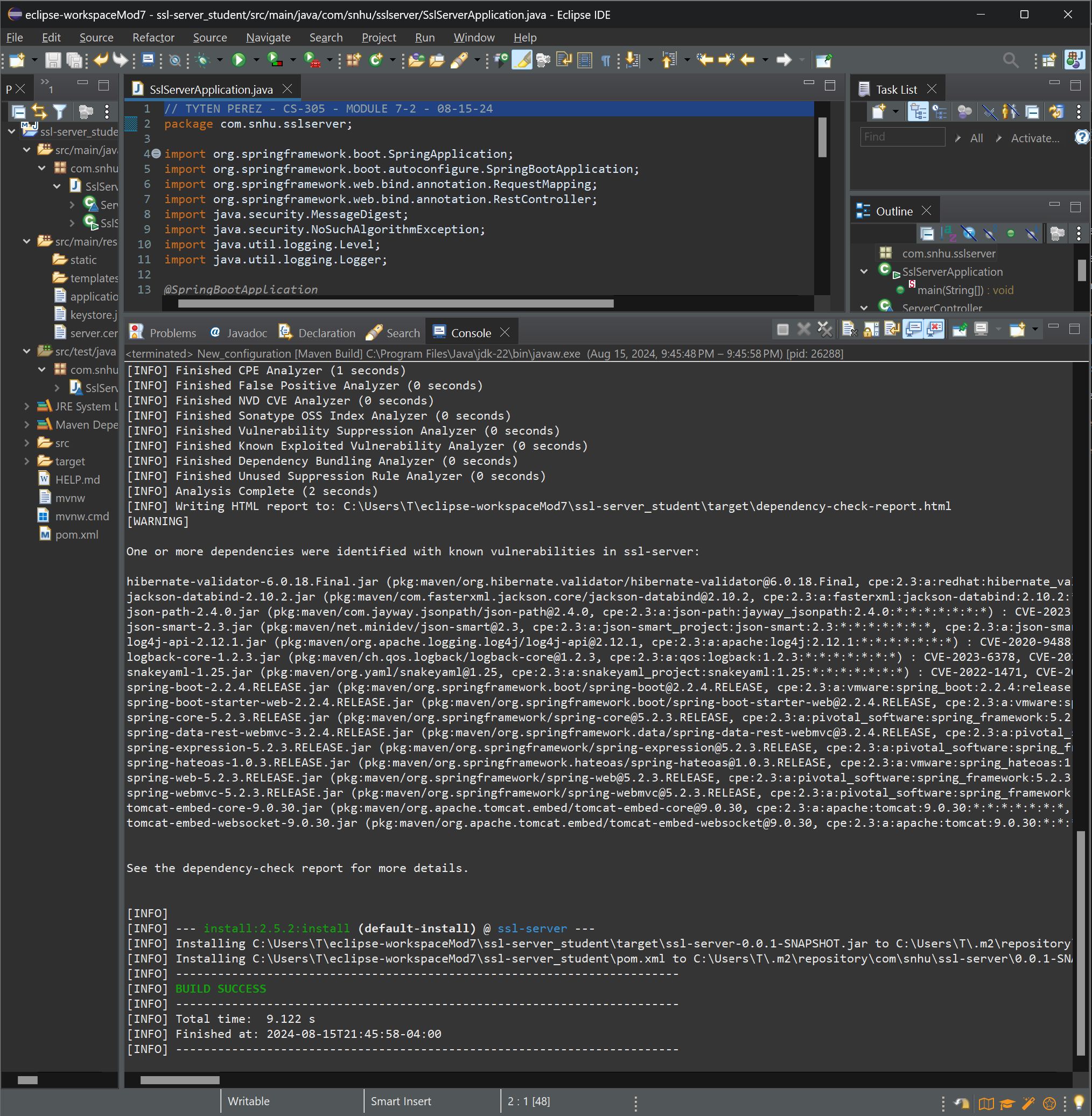
## Secure Communications

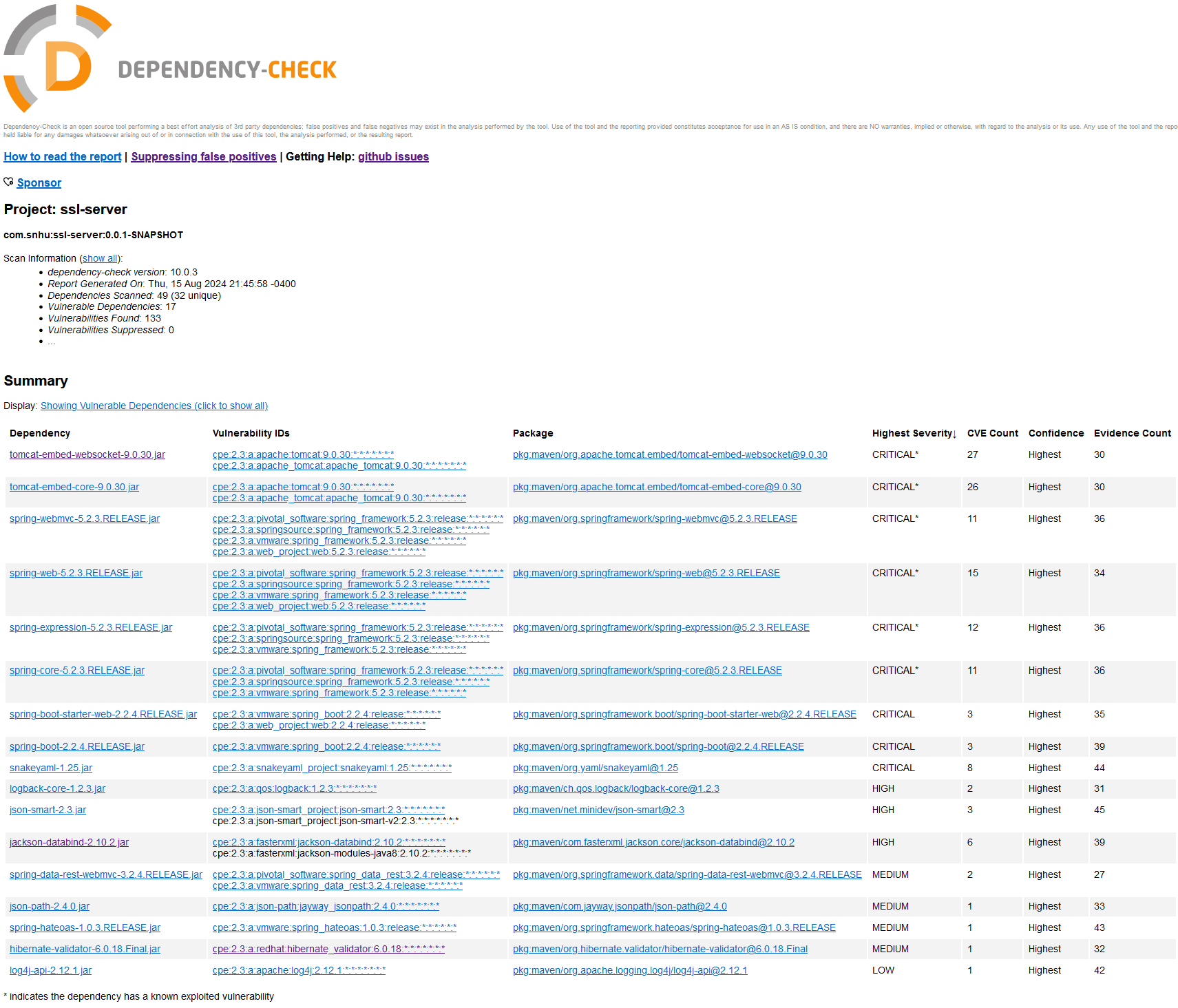
Screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

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





## Functional Testing

Screenshot below of the refactored code executed without errors.

A computer screen shot of a program

Description automatically generated

## Summary

When a malicious attacker attempts to exploit a target, they will likely need to gather as much technical information as possible about this target with details on the application server, libraries, and frameworks. One vector for this probing is in unhandled errors.

In refactoring the code, error handling was refined to ensure that any errors encountered during the hashing process does not reveal technical information to the end-user. This is important as the returned exception information could potentially be exploited by a malicious attacker and be used for future attacks like injection. Instead, the exceptions are logged at a severe level and return a generic error message to the end-user.

## Industry Standard Best Practices

This refactored code attempts to meet industry standard best practices by utilizing Open Web Application Security Project (OWASP) best practices and by utilizing dependency checks to identify and mitigate vulnerabilities.

For a financial company like Artemis Financial, these values protect against cyber-attacks which could compromise sensitive data they handle. The protection of this data is particularly important as it likely contains sensitive financial information or user data like social security numbers. Additionally, it protects the company’s assets and reputation. Going forward, the strategy of Artemis should integrate security in their continuous development life cycle as Attackers continuously try new methods over time. Being proactive can prevent data breaches, help comply with regulation, and build trust with clients, boosting the reliability of the company.