****

# 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** | **[Date]** | **Happy Dooley** |  |

## Client



## Developer

Happy Dooley

## Algorithm Cipher

### Recommended Encryption Algorithm Cipher

In response to Artemis Financial's request for secure data transmission, the appropriate encryption algorithm for file verification and data security is **AES (Advanced Encryption Standard)** with **RSA** for key exchange. AES is widely used and considered one of the most secure encryption algorithms for protecting sensitive data in modern systems. It operates as a **symmetric encryption** algorithm, meaning the same key is used for both encryption and decryption. The AES algorithm has different key lengths, typically 128-bit, 192-bit, and 256-bit. In this case, **AES-256** would be ideal for ensuring high security.

**RSA**, a **non-symmetric** (asymmetric) encryption algorithm, will be used to securely exchange the keys for AES encryption. RSA uses a pair of keys, a public key for encryption and a private key for decryption, which is vital for securely sharing the symmetric AES key.

This combination leverages the strengths of both symmetric and asymmetric encryption, providing both efficiency and strong security. AES-256 encryption is fast and secure for encrypting large amounts of data, while RSA ensures the safe exchange of the encryption key without exposing it to unauthorized users.

### Hash Functions and Bit Levels

AES operates at the **block level** with data blocks of 128 bits, and the cipher key size can be chosen at 128, 192, or 256 bits. In this case, AES-256 will be employed for the highest security level, providing a 256-bit key length and 128-bit data blocks.

For data verification via checksum, **SHA-256** (Secure Hash Algorithm 256-bit) is recommended. SHA-256 produces a fixed-length 256-bit hash value and is resistant to collision attacks, meaning it is computationally infeasible to find two different inputs that hash to the same output. The use of SHA-256 for checksum verification ensures that any tampering with the file's content can be detected by comparing the hash value sent with the file to the hash value computed from the received file.

### Random Numbers, Symmetric vs. Non-Symmetric Keys

* **Random Numbers**: In the context of AES encryption, **random number generation** is essential for securely creating the initialization vector (IV) for the cipher. AES requires an IV to ensure that the same plaintext encrypted multiple times will yield different ciphertexts. Random number generation can also be used for generating keys and salt values to prevent attacks like rainbow table attacks.
* **Symmetric Encryption (AES)**: Symmetric encryption uses a single key for both encryption and decryption. The key must be kept secret, and since it is computationally efficient, AES is suitable for encrypting large files or streams of data. However, the key exchange must be secure to prevent interception.
* **Asymmetric Encryption (RSA)**: Asymmetric encryption uses two keys: a **public key** for encryption and a **private key** for decryption. RSA is slower than AES, so it is typically used to securely exchange the symmetric AES key. The RSA public key can be used by Artemis Financial to encrypt the AES key, which is then securely transmitted to the intended recipient, who uses their private key to decrypt the AES key for further data encryption.

### History and Current State of Encryption Algorithms

* **History**: The AES algorithm replaced the older **DES (Data Encryption Standard)** after the U.S. National Institute of Standards and Technology (NIST) conducted a competition in the late 1990s. AES, which uses a block size of 128 bits and key sizes of 128, 192, or 256 bits, was designed to be secure against a variety of attack vectors, including brute-force and cryptanalysis attacks.
* **Current State**: AES is one of the most widely adopted encryption standards and is used globally to secure data. It is the default encryption algorithm for various security protocols, including **TLS/SSL** for secure web communications, and is also employed in military and government applications due to its high level of security. AES has remained unbroken for decades, with AES-256 being considered highly resistant to modern cryptanalytic techniques.

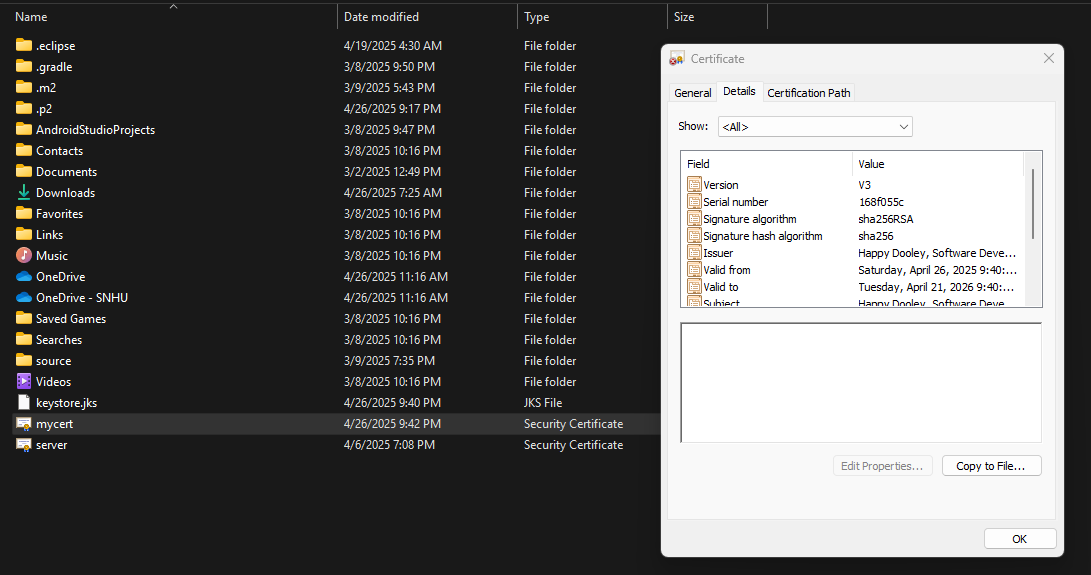
**RSA** remains the most commonly used asymmetric encryption algorithm, especially for encrypting keys and digital signatures. While RSA is still secure, it is relatively slower than modern elliptic curve cryptography (ECC) algorithms. For Artemis Financial’s use case, RSA is still an appropriate choice for key exchange due to its widespread support and adequate security.

### Conclusion

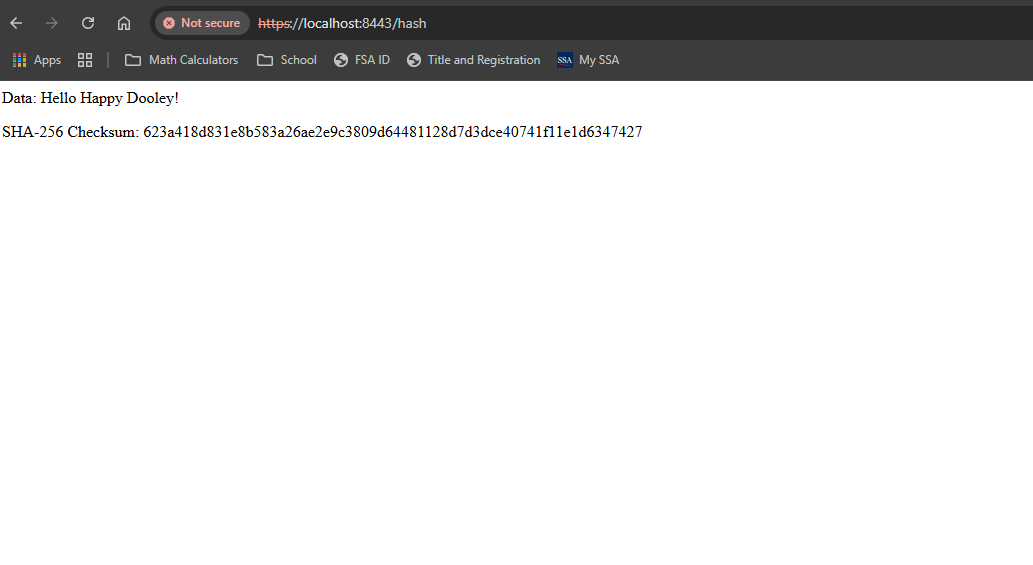
To secure Artemis Financial’s web application and provide robust data protection, the combination of **AES-256** for encrypting file transfers and **RSA** for secure key exchange is recommended. This approach leverages both symmetric and asymmetric encryption methods to provide strong security, high efficiency, and secure communication channels. Additionally, implementing **SHA-256** for checksum verification ensures the integrity of transmitted files, confirming that the data remains intact and unaltered during transmission.

This encryption solution not only meets the modern security standards but also aligns with best practices in cryptography, ensuring that Artemis Financial’s client data and financial information are protected.

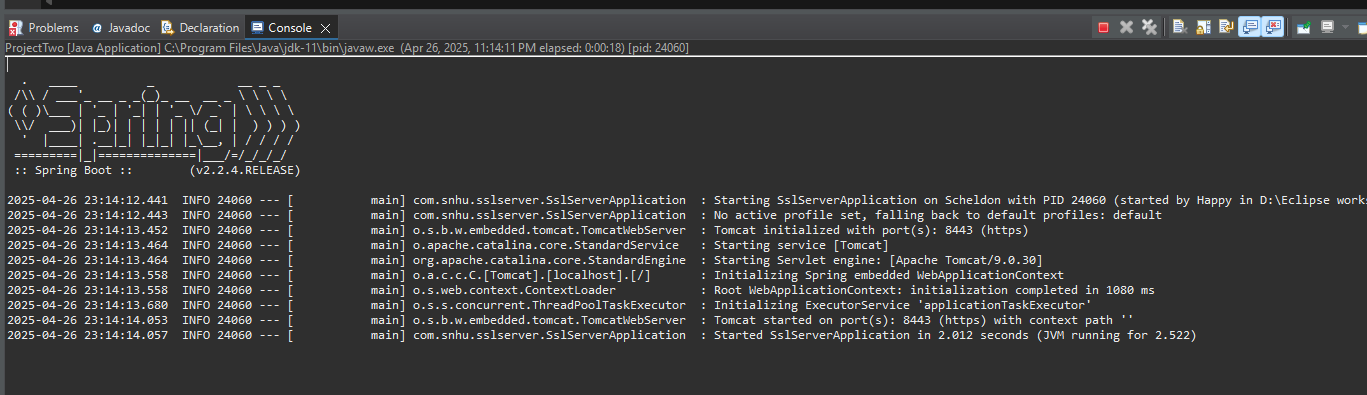
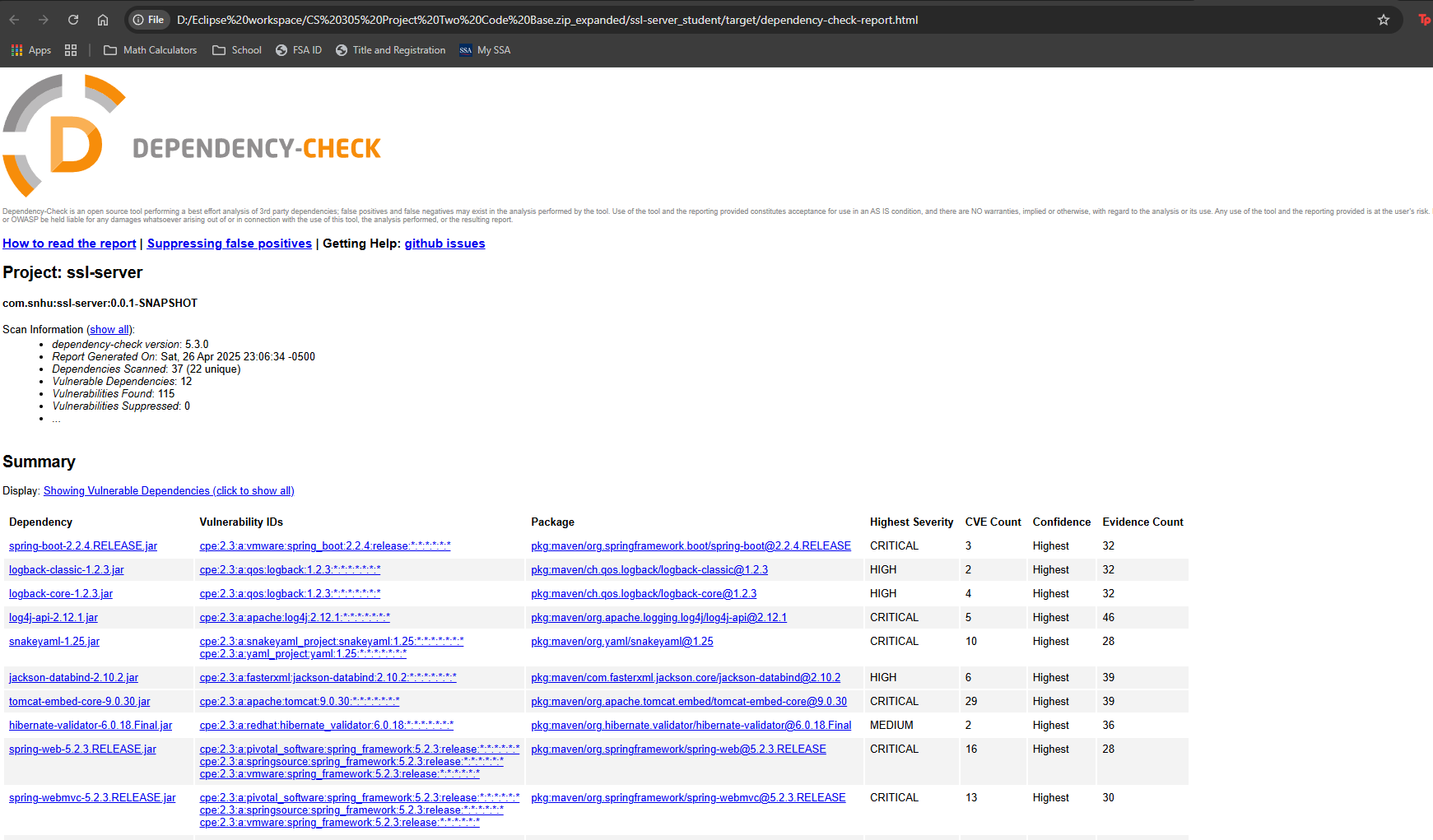
## Certificate Generation



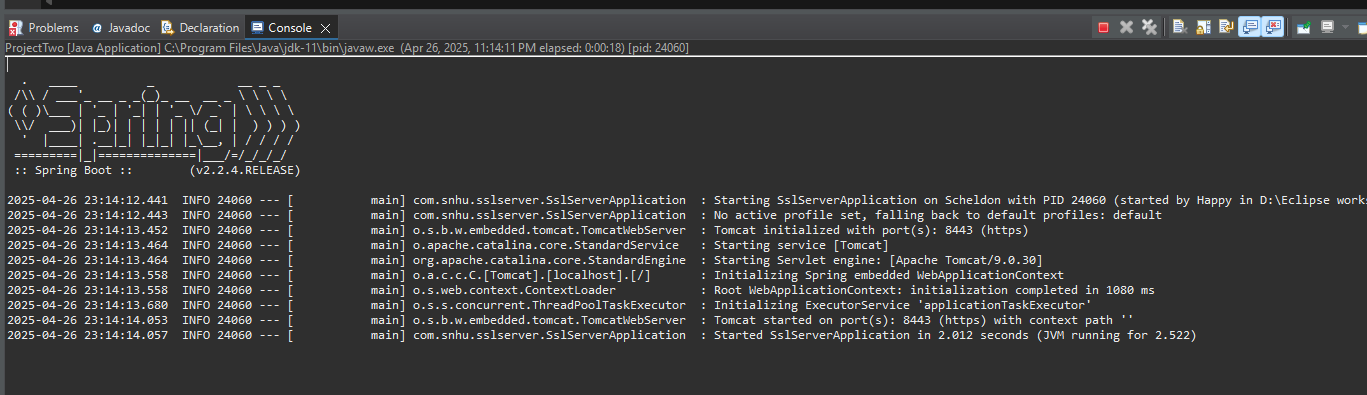
## Deploy Cipher & Secure communications



## Secondary Testing

## Functional Testing



## Summary

### Vulnerability Assessment and Refactoring Process:

Referring to the **vulnerability assessment process flow diagram**, the refactoring activities primarily addressed the **Identification**, **Analysis**, and **Mitigation** phases of the vulnerability management lifecycle. Initially, the application lacked secure communication, strong hashing mechanisms, and proper SSL/TLS configuration, potentially exposing it to data breaches and man-in-the-middle attacks.

The refactored code introduces strong cryptographic hashing using the SHA-256 algorithm to verify data integrity, mitigating risks associated with data tampering and ensuring secure file verification. SSL/TLS configuration was introduced through the integration of a Java keystore with a self-signed certificate, shifting the application’s communication from HTTP to secure HTTPS protocol. This step specifically targets vulnerabilities in communication security and protects data during transmission.

### Process for Adding Layers of Security:

The approach taken for adding layers of security involved several key phases:

* **Cryptographic Security:** Implemented SHA-256 hashing using the java.security.MessageDigest class to securely generate a checksum for sensitive data verification. SHA-256 ensures data integrity and protection from tampering attempts.
* **Secure Communication:** Deployed SSL/TLS by generating and configuring a self-signed SSL certificate using the Java Keytool. Refactored the application's configuration (application.properties) to enforce HTTPS protocol, thereby protecting data in transit.
* **Dependency Security:** Integrated OWASP Dependency-Check into the Maven build lifecycle, ensuring automated detection of vulnerable libraries and dependencies. This static testing ensures that no new security vulnerabilities are introduced through external components.

By adding multiple layers—cryptographic hashing, secure communication protocols, and dependency security—the application now demonstrates defense-in-depth, significantly strengthening the overall security posture.

## Industry Standard Best Practices

### Maintaining Existing Application Security:

To maintain and enhance the existing security standards, several industry best practices were adopted:

* **Use of Strong Cryptographic Algorithms:** SHA-256 hashing was employed, following current cryptographic recommendations and avoiding outdated or compromised algorithms such as MD5 or SHA-1.
* **Secure Protocol Enforcement:** Transitioning to HTTPS and configuring SSL/TLS using secure standards ensured data confidentiality and integrity during data exchanges.
* **Secure Key Management:** Managed SSL/TLS keys securely through Java Keystores, enforcing proper access controls and configurations to prevent unauthorized key exposure.
* **Static Security Analysis:** Utilized OWASP Dependency-Check within the build process for continuous detection and remediation of vulnerabilities in third-party libraries, adhering strictly to industry guidelines for continuous security monitoring.

### Value of Applying Industry Standard Best Practices:

Adopting industry-standard best practices for secure coding delivers substantial benefits to the company’s overall well-being:

* **Enhanced Data Protection:** Protecting sensitive client information significantly reduces the risk of data breaches, safeguarding the organization's reputation and customer trust.
* **Compliance and Risk Management:** By aligning with accepted industry standards (OWASP guidelines, secure coding standards, and encryption protocols), the application meets regulatory and compliance requirements more effectively, reducing potential legal and financial liabilities.
* **Operational Stability and Trust:** A well-secured software application minimizes vulnerabilities, reduces downtime, and ensures smoother operations, enhancing the company’s reliability and market competitiveness.