****

# 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** | **8/17/2024** | **Ivette Cerpa** |  |



## Developer

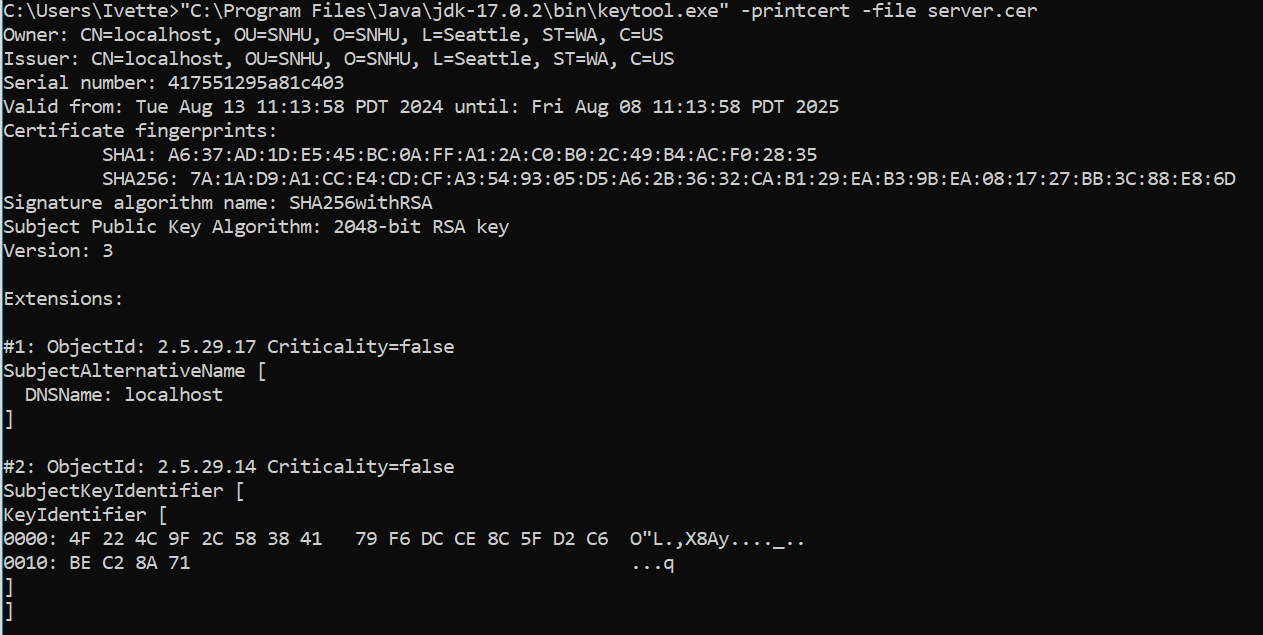
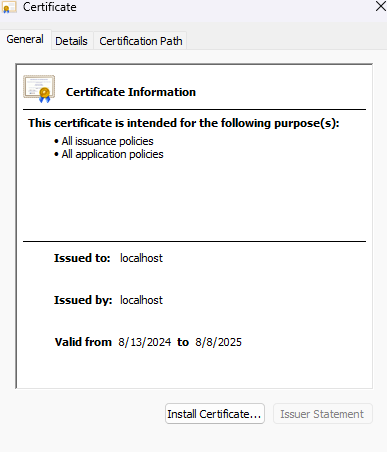
Ivette Cerpa

## Algorithm Cipher

AES, a symmetric-key algorithm, operates through a series of linked operations involving substitution and permutation, transforming plaintext into a form that is difficult to reverse without the key (Dennis, 2024). The same key is used for both encryption and decryption, simplifying key management while emphasizing the importance of key secrecy. The use of random numbers in key generation adds an additional layer of security, making it harder for an attacker to predict the key. Additionally, AES supports key sizes of 128, 192, and 256 bits, providing different levels of security based on the required strength.

Originally known as Rijndael and declared a standard by NIST in 2001, AES superseded the Data Encryption Standard (DES) due to its improved security and efficiency (Veritas Technologies, n.d.). By implementing AES, Artemis Financial can ensure that client data and financial information are protected against unauthorized access and potential breaches.

## Certificate Generation



## Deploy Cipher

A close up of a computer screen

Description automatically generated

## Secure Communications

A screenshot of a phone

Description automatically generated

## Secondary Testing

**Before refactoring code:**

A screenshot of a computer error

Description automatically generated

**After refactoring code:**

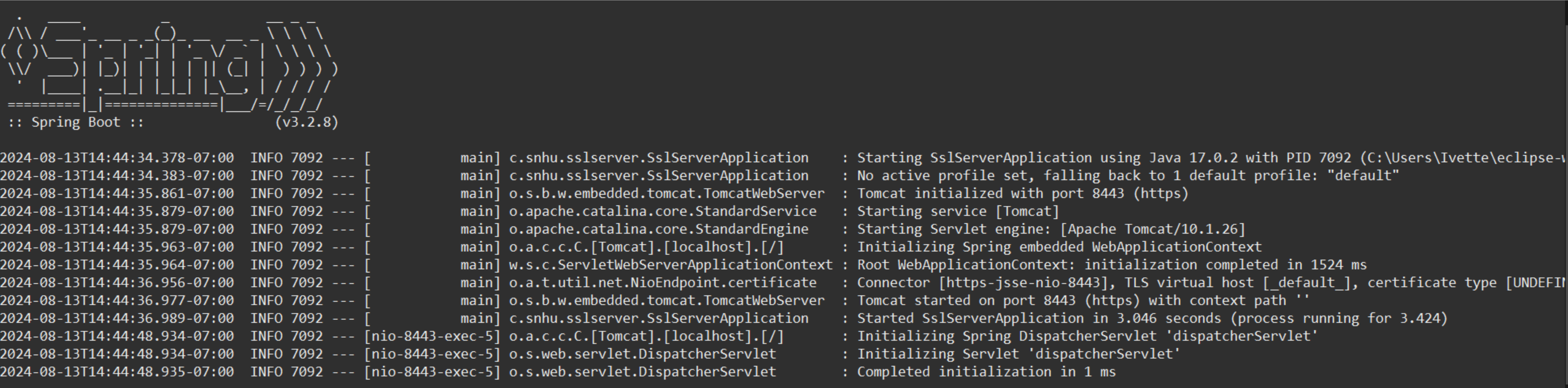
**Refactored code running without errors:**

A computer screen with a black background

Description automatically generated

## Functional Testing

**Refactored code executed without errors:**



## Summary

1. **Dependency Updates**: All dependencies were updated to their latest secure versions. This ensures the application leverages the newest features and improvements while mitigating known vulnerabilities. This change enhances code quality and maintains compatibility with the latest API changes.
2. **Environment Variables**: Hardcoded passwords in the application.properties file were replaced with environment variables. This protects sensitive information and improves security in client-server communication. Additionally, this approach encapsulates sensitive data, making it easier to manage and secure.
3. **Exception Handling:** Exception handling in the generateChecksum method was improved by logging exceptions and returning user-friendly error messages. This enhancement boosts code readability, maintainability, and debugging capabilities, while encapsulating the error handling logic to make the code more secure.
4. Updates were made to controllers, services, and the addition of the dotenv library for secure environment variable management.

## Industry Standard Best Practices

In this project for Artemis Financial, industry-standard best practices for secure coding were applied to mitigate known security vulnerabilities and maintain the software application’s existing security. One of the key practices was updating dependencies to the latest secure versions so that the application leverages the newest features and improvements while addressing any known vulnerabilities. Additionally, hardcoded passwords were replaced with environment variables, a best practice that protects sensitive information and enhances security in client-server communication. This method encapsulates sensitive data, making it easier to manage and secure, and aligns with the principle of least privilege by ensuring that sensitive information is not exposed in the codebase. Additionally, exception handling was improved by logging exceptions and returning user-friendly error messages, which will enhance code readability, maintainability, and debugging capabilities. This practice ensures that errors are properly managed and logged, providing valuable insights for future troubleshooting and security audits.

Applying industry-standard best practices for secure coding is essential for the overall well-being of Artemis Financial. As a consulting firm that develops individualized financial plans for its customers, the security of client data and financial information is paramount. By modernizing their operations and implementing the most current and effective software security measures, Artemis Financial can protect its clients’ sensitive information from potential threats and breaches. The addition of a file verification step, such as the checksum we added, will allow for secure communications and data integrity, further enhancing the trust and reliability of their services. By prioritizing secure coding practices, Artemis Financial demonstrates its commitment to protecting client data and maintaining the highest standards of security in its operations.

Sources:

Dennis. (2024, April 19). What is AES encryption and how does it work? WEZOM. <https://wezom.com/blog/what-is-aes-encryption-and-how-does-it-work>

Veritas Technologies. (n.d.). Fort Knox for your data: How AES Encryption Secures Your business. Veritas. <https://www.veritas.com/information-center/aes-encryption>