****

# 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** | **08/13/2023** | **Harshilkumar Jayswal** |  |

## 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

Harshilkumar Jayswal

## Algorithm Cipher

Artemis Financial is a company that creates personalized financial plans for its clients. They want to modernize their operations and are currently using a RESTful web application programming interface (API). They're seeking ways to defend against external threats and secure their data.

To achieve this, I’m recommend using SHA-256, which stands for Secure Hash Algorithm 256-bit, a strong encryption method to protect their data. It is a widely recognized tool in the world of encryption. It's not exactly an encryption method, but rather a hash function. Its main job is to convert a given message into a fixed-size code, or hash value, regardless of the message's length. The hash value generated by SHA-256 consists of 256 bits.

They are used for purposes like verifying data integrity. Hash functions take input data and turn it into a unique hash value, often referred to as a "digest." The 256-bit length of SHA-256's hash value ensures a strong level of security due to the incredibly vast number of possible outcomes. SHA-256 is more commonly involved in activities like digital signatures, securing passwords, and ensuring the integrity of data.

SHA-256 belongs to the SHA-2 family of hash functions, developed by the National Security Agency (NSA) in the early 2000s. This was seen as an upgrade over its predecessor, SHA-1, which was found to have vulnerabilities. SHA-256 and its counterparts, such as SHA-384 and SHA-512, were created to address these concerns.

To increase insecurity for files, I recommend using AES-256 cipher algorithm. Unlike SHA-256, AES-256 is a full-fledged encryption algorithm designed to safeguard data. It utilizes a symmetric key approach, where the same key is employed for both encryption and decryption operations.

Random numbers play a crucial role in the generation of strong encryption keys, especially in AES-256. These keys are essential for ensuring data confidentiality. In AES-256, symmetric keys are used, meaning the same key is used for both encryption and decryption. This means that the same secret key is employed to both lock and unlock the data.

The U.S. government introduced AES in the early 2000s, selecting Rijndael as the winner of a competition to find a strong encryption algorithm. Since its introduction, AES-256 has been widely adopted as a trustworthy encryption standard. Currently, AES-256 is considered one of the most secure encryption methods available. Its strength lies in its key length and the complexity of its operations, making it a preferred choice for securing sensitive data.

## Certificate Generation

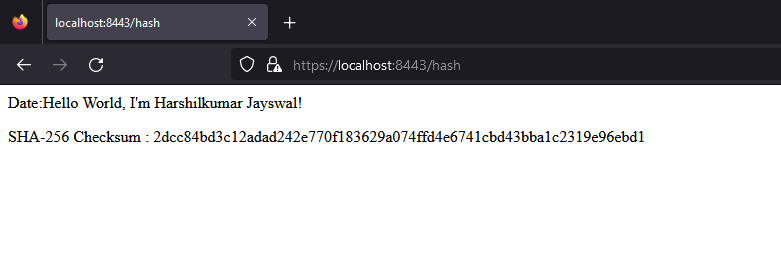
Insert a screenshot below of the CER file.

A computer screen with white text

Description automatically generated

## 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.

A screenshot of a computer

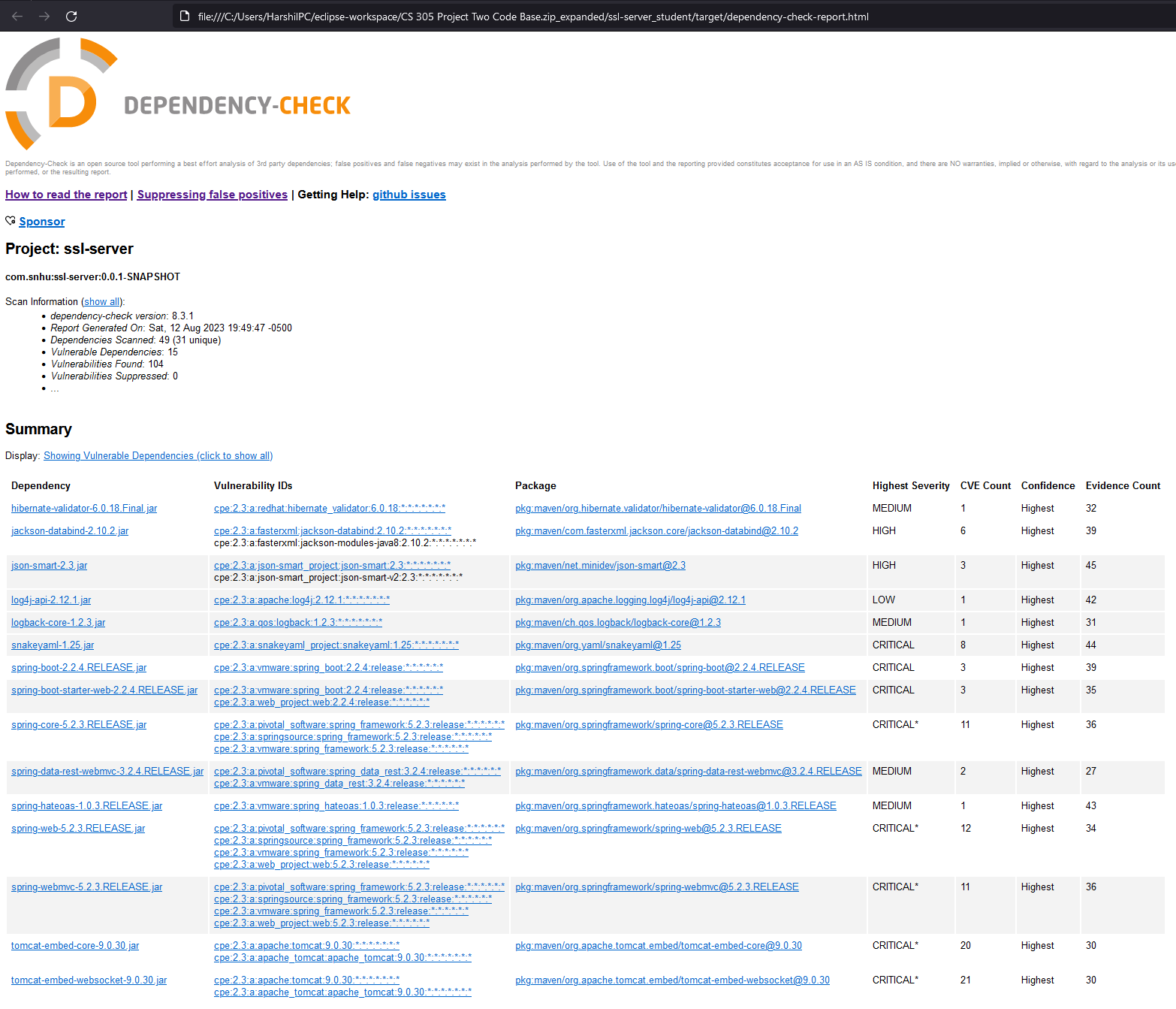
Description automatically generated

## Secondary Testing

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

A screenshot of a computer program

Description automatically generated

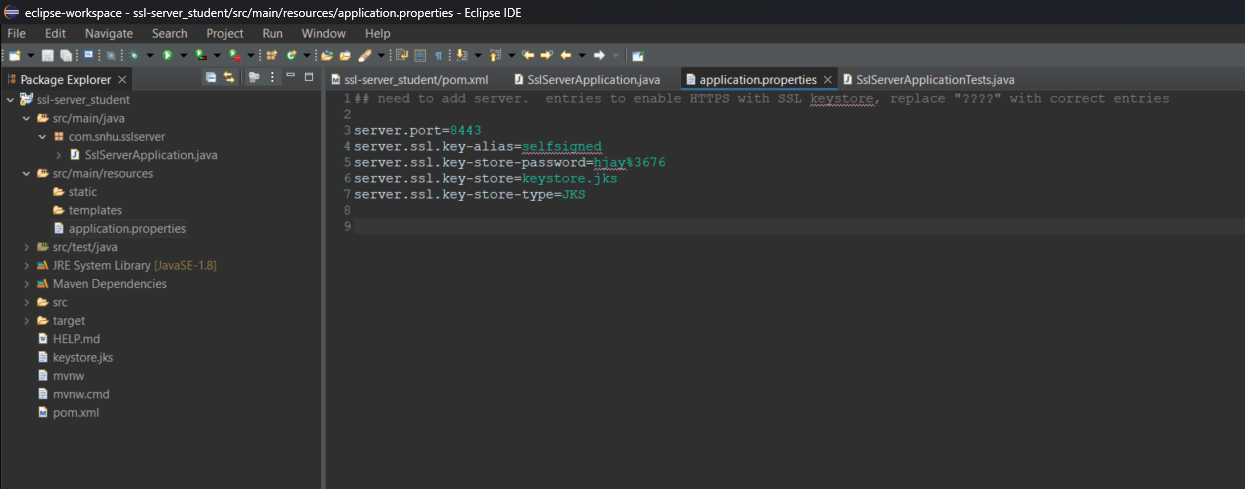


## Functional Testing

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

A screenshot of a computer program

Description automatically generated



## Summary

The refactored code is a Spring Boot application that implements a simple server that uses hash function SHA-256 to generate a hash for a string value and returns the hash value along with the original string. This is a secure approach for generating hash values and is suitable for data integrity checks. The controller class ServerController is annotated with “@RestController”, and it handles the /hash endpoint.

I added code handler exception “NoSuchAlgorithmException” that may arise from attempting to use the SHA-256 algorithm because it is important to handle exceptions to avoid potential vulnerabilities. The “calculateHash” method calculates the hash value, converts it into a hexadecimal string.

The use of SHA-256 ensures a strong and secure hash function for generating checksums. The code is minimal and simple to avoid potential attack surface. The data is encoded using a specified character encoding, reducing the likelihood of vulnerabilities related to character encoding mismatches. The code provides a clear response structure and ensures that no sensitive information is exposed.

## Industry Standard Best Practices

I applied industry-standard best practices for secure coding to maintain the software application's current security. I used the latest version of Maven Dependency check 8.3.1 in the pom.xml. This ensures that the newly discovered vulnerabilities have visibility. I used the industry-standard SHA-256 hashing algorithm for generating hash values. This algorithm is widely recognized for its security and resistance to various attacks, ensuring data integrity and authenticity. I followed the best practice of graceful error handling, preventing unexpected crashes and minimizing the potential attack surface.

By implementing these practices, the company reduces the risk of various security vulnerabilities, such as injection attacks, data breaches, and unauthorized access. This helps safeguard sensitive customer data and business-critical information. It helps the company stay compliant with governmental regulations, avoiding legal penalties and reputational damage. Companies that prioritize secure coding gain a competitive edge by offering customers safer products and services.

**References:**

Chen, L. (2022, May 26). The cornerstone of cybersecurity – Cryptographic Standards and a 50-year evolution. NIST. https://www.nist.gov/blogs/cybersecurity-insights/cornerstone-cybersecurity-cryptographic-standards-and-50-year-evolution