****

# 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** | **12/10/2023** | **Martin Granados** |  |

## 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

Martin Granados

## Algorithm Cipher

I would recommend the SHA-256 algorithm cipher. In 2001, the National Security Agency created SHA-256 to replace SHA-1. In addition, SHA-256 is one of the safest hashing algorithms on the market right now with a collision rate of under 0.01%. A collision occurs when an algorithm produces a similar hash result for two distinct pieces of data. The numbers 0 through 9 or lowercase letters are the characters produced by SHA256. It is extremely unlikely that two pieces of data will have the same hash value or result in a collision because there are 3664 potential possibilities (What Is SHA-256 Algorithm: How It Works & Applications | Simplilearn, n.d.). Bit levels in the Sha-256 hash algorithm are chosen at random. Before being employed, the input value is compressed during the hash function creation process. The name of the compressed data is the hash value. The bit levels dictate how long the encryption will take. The most basic type of encryption, symmetric keys encrypt plain text with a key. They comply with AES-256 requirements. Since asymmetric keys are used for internet connections more than anything else, they are thought to be more secure.

## Certificate Generation

Insert a screenshot below of the CER file.

A screen shot of a computer

Description automatically generated

A number of numbers on a white background

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer code

Description automatically generated

## 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

Description automatically generated

## Functional Testing

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

**package** com.snhu.sslserver;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RestController;

**import** java.security.GeneralSecurityException;

**import** java.security.MessageDigest;

**import** java.security.NoSuchAlgorithmException;

**import** java.math.BigInteger;

**import** java.nio.charset.StandardCharsets;

@SpringBootApplication

**public** **class** SslServerApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(SslServerApplication.**class**, args);

}

@RestController

**class** ServerController {

**public** String calculateHash(String name) **throws** NoSuchAlgorithmException {

**if** (name.isEmpty()) {

**return** "";

}

MessageDigest md = MessageDigest.*getInstance*("SHA-256");

**byte**[] hash = md.digest(name.getBytes(StandardCharsets.***UTF\_8***));

BigInteger number = **new** BigInteger(1, hash);

StringBuilder hexString = **new** StringBuilder(number.toString(16));

**while** (hexString.length() < 32) {

hexString.insert(0, '0');

}

**return** hexString.toString();

}

@RequestMapping("/hash")

**public** String myHash() **throws** NoSuchAlgorithmException {

String data = "Martin Granados";

String hash = calculateHash(data);

**return** String.*format*("data: %s; Name ol Cipher Algorithm Used : CheckSum Value: %s", data, hash);

}

}

}

## Summary

After reworking the code, I made the SSLServerApplication.java file have a secure RestController that acts as the secure controller for the hash RESTful API. This ServerController class satisfies the prerequisites and handles the secure coding concern in the Vulnerability Assessment Diagram. To lessen the possible attack surface, I've decided to utilize SHA-256 as the hashing cipher for this function. I also changed the OWASP version to the current version.

## Industry Standard Best Practices

Input validation is the process of examining and sanitizing user inputs to prevent attacks such as SQL injection, cross-site scripting (XSS), and command injection. Enhancing user access control, providing secure authentication, and managing passwords may be achieved by the use of multi-factor authentication (MFA), strong password regulations, and hashing algorithms. By allowing users and programs the smallest amount of permissions necessary to do their tasks, the idea of least privilege seeks to reduce the likelihood of unauthorized access or activity. Safe data transmission and storage: protecting sensitive data when it's at rest with industry-standard encryption techniques and secure communication protocols like HTTPS. Regular security patching and upgrades involve installing security updates on schedule, patching known vulnerabilities to lower the likelihood of exploitation, and upgrading the application and all of its dependencies. Error management and logging: ensuring secure logging protocols to detect potential security threats and implementing suitable error handling to prevent the leakage of confidential information. Using industry-standard best practices for secure coding reduces costs, protects sensitive data, satisfies regulatory requirements, and eventually wins over partners and customers. All of these benefits add significant value to a company. Additionally, it enhances the company's reputation and helps to create a good image.

Input Validation and Output Encoding: Adequate input validation must be implemented in order to prevent common vulnerabilities such as SQL injection and cross-site scripting (XSS). It is recommended to employ output encoding when presenting user-generated material in order to defend against XSS attacks.

Code Quality: Depending on the user's inner code quality, one can assume command of method access. For example, a user could only access their own data and not that of other users, the server, or any other third party.

APIs: Due to its dual internal and external operations, an API would need to be developed. It would make it possible to choose which data is allowed to access.

Code Error: It is evident which portions of the API need to be fixed by putting error handling in place. Artemis Financial wouldn't have to worry about customer data disclosure or illegal access in this way.

Cryptography: Since there would be other currencies involved, it would be crucial to add encryption to Artemis Financial in order to prevent users' information from being compromised by hackers operating from around the globe.

References

Geeksforgeeks. (2021, October 15). *Advanced Encryption Standard (AES)*. GeeksforGeeks. <https://www.geeksforgeeks.org/advanced-encryption-standard-aes/>

SSL2BUY. (2019, February 7). *Symmetric vs. Asymmetric Encryption – What Are differences?* SSL2BUY Wiki - Get Solution for SSL Certificate Queries. https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

What Is SHA-256 Algorithm: How it Works & Applications | Simplilearn. (n.d.). Simplilearn.com. https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm