****

# 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** | **08/17/2024** | **Taylor Stapus** |  |

## 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

Taylor Stapus

## Algorithm Cipher

The recommended encryption algorithm that should be used to protect Artemis Financial is

SHA-256. SHA, which stands for Secure Hash Algorithm, is a family of cryptographic hash functions. A hash function takes data of any size and produces a fixed-size bit string that represents the unique ‘fingerprint’ of the original data. In this case a 256-bit length as suggested in the name. SHA-256 can be applied in multiple settings such as, password-based key derivatives, certificate authorities, digital signatures, password storage and file integrity checking. SHA-256 provides robust security and integrity in these various applications by safeguarding data and ensuring trust by enabling secure digital interactions.

Artemis Financial stores private information such as customer/ banking information and banking transactions making it worthwhile to invest in implementing one of the best encryption algorithms available. A security breach could potentially cost the company a significant amount in financial losses as well as losing credibility to the customers and the loss of business. Regarding the risk of what a security breach would entail, it is imperative that Artemis Financial uses SHA- 256 as it provides the best security measures that fit the company’s needs.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen shot of a program

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer program

Description automatically generated

## Secure Communications

I haven’t been able to connect to the browser page even with turning off my firewall and any security measures in place. I have added a screenshot of the check sum verification code which includes the section where it should add my name and a unique data string.

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.

Screenshot of changing the maven dependency version and a successful build:

A screenshot of a computer

Description automatically generated

Screenshot of the dependency check report summary:

A screenshot of a computer error

Description automatically generated

List of dependencies:

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.MessageDigest;

**import** java.security.NoSuchAlgorithmException;

@SpringBootApplication

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

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

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

}

}

//**FIXME**: Add route to enable check sum return of static data example: String data = "Hello World Check Sum!";

@RestController

**class** ServerController {

**private** **static** **final** **char** [] ***HEX\_ARRAY*** = "012345567ABCDEF".toCharArray();

**private** String getHash(String input) {

**try** {

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

**byte** [] messageDigestMD5 = messageDigest.digest();

**return** *bytesToHex* (messageDigestMD5);

}

**catch** (NoSuchAlgorithmException e) {

e.printStackTrace();

}

**return** input;

}

**public** **static** String bytesToHex(**byte** [] bytes) {

**char** [] hexChars = **new** **char** [bytes.length \* 2];

**for** (**int** j = 0; j<bytes.length; j++) {

**int** v = bytes [j] & 0xFF;

hexChars [j\*2] = ***HEX\_ARRAY*** [v>>>4];

hexChars [j\*2+1] = ***HEX\_ARRAY*** [v & 0x0F];

}

**return** **new** String(hexChars);

}

@RequestMapping ("/hash")

**public** String myHash() {

String data = "Hello Taylor Stapus";

String hash = getHash(data);

**return** "<p>data: " + data + "</p><p> Name of Cipher Used: SHA-256 Value:" + hash;

}

}

## Summary

The application code has been refactored by adding security measures to RestController. I chose to implement the encryption algorithm SHA-256 because it is one of the most secure cyphers currently at 256 bits. The first step in adding layers of security to this application was to create a certificate to ensure authentication and secure internet access of the user. I then made sure that the maven dependency was up to date by changing the version it was running to the latest addition. To maintain security for the application, dependency checks would need to be run often and thoroughly. While following the assessment process flow diagram, the area that are highlighted with this program would be, Input Validation, as there currently none, APIs, Code Error due to missing code and syntax error and Cryptography because I needed to add SHA-256 to ensure safety.

## Industry Standard Best Practices

Applying industry standards best practices help prevent security vulnerabilities. Following secure coding standards can reduce vulnerabilities before the deployment of the software which can lead to potential security attacks. Best practices are important because they help ensure reliability, safety, and security. For this software application, I implemented AES (advanced encryption standard) as it is a symmetric block cipher and essential to the functionality of the code.