****

# 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** | **20 Feb 2024** | **Tyler Barnes** |  |

## 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

Tyler Barnes

## Algorithm Cipher

## Recommend an appropriate encryption algorithm cipher to deploy, given the security vulnerabilities, and justify your reasoning. Review the scenario and the supporting materials to support your recommendation. In your practices for secure software report, be sure to address the following:

* 1. Provide a brief, high-level overview of the encryption algorithm cipher.
  2. Discuss the hash functions and bit levels of the cipher.
  3. Explain the use of random numbers, symmetric versus non-symmetric keys, and so on.
  4. Describe the history and current state of encryption algorithms.

Artemis Financial can benefit from AES with SHA-256 as a hash function for their algorithm cipher. This cipher has been proven to provide fortification against all types of attacks. An exception would be brute force attacks because it has shown vulnerable to it, however, the likelihood of this decreases as the byte count becomes more complicated with the cipher. This cipher has been utilized by multiple platforms and institutions, including the government and banking systems. This shows how effective this cipher is.

Encryption algorithms have been around for centuries. An example of this is when the Spartans used scytale which allowed soldiers to send messages in battle using a wooden rod and leather that had the message on it. A more recent encryption tool was a machine called the Enigma in the early 1900s made by Arthur Scherbius. This device used rotating disks where keys were embedded with an encoded substitution table. Every time a character was typed, the table changed. This was used by the Nazi’s until the table was cracked. Today, encryption is utilized everyday with ATM machines, or Alexa from Amazon.

## Certificate Generation

Generate appropriate self-signed certificates using the Java Keytool in Eclipse.

1. To demonstrate that the certificate was correctly generated:
   1. Export your certificates (CER file).
   2. Submit a screenshot of the CER file in your practices for secure software report.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screen shot of a computer program

Description automatically generated

## Secure Communications

Insert a screenshot below of the web browser that shows a secure webpage.



I tried and couldn’t get it secure. I don’t know what I’m doing wrong.

## Secondary Testing

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

Run a secondary static testing of the refactored codeusing the OWASP Dependency-Check Maven (see Supporting Materials) to ensure code complies with software security enhancements. You need to focus on only the code you have added as part of the refactoring. Complete the dependency check and review the outputto ensure you did not introduce additional security vulnerabilities. Include the following in your practices for secure software report:

1. A screenshot of the refactored code executed without errors
2. A screenshot of the report of the output from the dependency-check static tester

A screen shot 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 screen

Description automatically generated

A screen shot of a computer

Description automatically generated

## Summary

Discuss how the code has been refactored and how it complies with security testing protocols. In the summary of your practices for secure software report, be sure to address the following:

1. Refer to the Vulnerability Assessment Process Flow Diagram. Highlight the areas of security that you addressed by refactoring the code.
2. Discuss your process for adding layers of security to the software application.

By encrypting the data by using a hash function, this bolstered the security of the API, since this is a RESTful application. To make sure the data can be exchanged securely, I used a certificate. This improved security of the client and by using a try and catch clause reaffirmed the integrity of the code would be secure. I had to make sure the system was up to date on the Spring Boot and Tomcat versions improved security of the application tremendously and provided safeguard against known vulnerabilities.

The security of a system is vital in today’s climate of technology. There are so many potential threats daily and making sure your consumer’s data is the most important. Constantly monitoring encryption methods used, maintaining dependencies, and reviewing code base, you can ensure the system is as secure as it can be.