****

# 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** | **4/15/2024** | **Alex Mehr** |  |

## 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

Alex Mehr

## Algorithm Cipher

Artemis Financial is requesting for additional security on the web application to secure their system and prevent unauthorized communications. The encryption is the best protection recommendation to protect their application from attackers who want to access their data and retrieve financial information. By adding encryption, the system will secure the connections and prevent unauthorized access to the system. Another recommendation is to use SHA-256 cipher algorithm because the SHA-256 provides a high level of encryption which included many different key combination possibilities of 256 bits long. The 256-SHA which uses Java random number generator, ensures the highest level of encryption security by creating a checksum that verifies the file validity and hash function would use the cipher and provide a message.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Secure Communications

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

A screenshot of a computer

Description automatically generated

## Functional Testing

A computer screen shot of a program

Description automatically generated

## Summary

The @RestController that I added to the refactored code, serving the program as a secure controller for the RESTful hash endpoint. This class is provided to address the vulnerability concerns and ensure the requirements are met by choosing SHA-256 for the algorithm cipher and with a little change to the code, the vulnerability of the program minimized, and data is more protected. Also, the version on Maven dependency check has been updated to 9.0.10 from 5.3.0 which helps with identifying the vulnerabilities and mitigate them.

## Industry Standard Best Practices

Mitigating known security vulnerabilities and maintaining a secure software application requires industry standard practices applied in the program. I applied some security measures which has been described in previous section. Aside from those, we should have proper security implementation to maintain a proper and secure application. The following security measures are required for a secure application.

* Password management and authentication: Implementing MFA (Multi-factor authentication) and cipher algorithm and strict password policies.
* Input Validation: Validating user inputs to prevent XSS (Cross-site Scripting) and Data injections.
* Data Encryption: Implementing algorithm ciphers and using HTTPS protocol instead of HTTP to secure data transmission.
* Error Handling: Implementing a proper error-handling code to catch the errors and prevent the application from leaking sensitive data and secure the application from attacks.

These are some of the security measures we should take to keep the application secure and unharmed which will help with cost reduction, high-level security and complied with rules and regulations.

**Resources:**

Manico, J., &amp; Detlefsen, A. (n.d.). Iron-clad Java - Chapter 6. O'Reilly Online Learning.

Retrieved March 20, 2023, from https://learning.oreilly.com/library/view/iron-clad-

java/9780071835886/ch06.html#ch06lev2sec3

baeldung, W. by: (2021, November 14). Java AES encryption and decryption. Baeldung.

Retrieved March 20, 2023, from https://www.baeldung.com/java-aes-encryption-decryption

Types of encryption: Symmetric or asymmetric? RSA or AES? Prey Blog. (2021, June 15).

Retrieved March 20, 2023, from https://preyproject.com/blog/types-of-encryption-symmetric-or-

asymmetric-rsa-or-aes

Bernstein, C., &amp; Cobb, M. (2021, September 24). What is the Advanced Encryption

Standard (AES)? definition from searchsecurity. Security. Retrieved March 20, 2023, from

https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard

9

Manico, J., &amp; Detlefsen, A. (n.d.). Iron-clad Java -Chapter 6. O’Reilly Online Learning.

Retrieved march 20, 2023, from <https://learning.oreilly.com/liberary/view/iron-clad-java/9780071835886/ch06.html#ch06lev2sec3>

Bernstein, C., &amp; Cobb, M. (2021, September 24). What is the advanced Encryption Standard (AES)?

Definition from searchsecurity. Security. Retrieved March 20, 2023, from <https://www.techtarget.com/searchsecurity/definition/advanced-Encryption-Standard>