****

# CS 305 Project Two

**Practices for Secure Software Report**

**Project Two**

**Braxton Morrow**

**SNHU**

**Vivian Lyon**

**Due: 4/17/2022**

Table of Contents

[CS 305 Project Two 1](#_Toc101026431)

[CS 305 1](#_Toc101026432)

[Document Revision History 3](#_Toc101026433)

[Client 3](#_Toc101026434)

[Developer 4](#_Toc101026435)

[1. Algorithm Cipher 4](#_Toc101026436)

[2. Certificate Generation 6](#_Toc101026437)

[3. Deploy Cipher 7](#_Toc101026438)

[4. Secure Communications 7](#_Toc101026439)

[5. Secondary Testing 7](#_Toc101026440)

[6. Functional Testing 9](#_Toc101026441)

[7. Summary 9](#_Toc101026442)

8.Sources 10

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **4-10-2022** | **Braxton Morrow** | **Made Security recommendations and designed and implemented recommended code** |

## Client



## Developer

Braxton Morrow

## 1. Algorithm Cipher

I am choosing the Advanced Encryption Standard (AES-256) to secure Artemis Financial against harmful attacks that may want to steal their or their client’s personal information. It is of the upmost importance for Artemis Financial to have secure data as a financial Institution, AES is currently the most secure algorithm, “This standard may be used by Federal departments and agencies when an agency determines that sensitive information requires cryptographic information” (Dworkin et al., 2001). This algorithm has been used by a variety of different companies, because of its ability to withstand attacks. It has been attacked in the past, however only systems that were incorrectly configured received issues. As well there have been several attacks on the higher bit versions. According to Bruce Schneider, a graduate and lecturer at Harvard, as well as the Chief of Security Architecture for Inrupt Inc. “AES is the best known and most widely used block cipher… There is no known attack which is faster than the 2^128 exhaustive search” (Schneider, 2009). What this means is it will take a regular computer 2^128 years to crack the cipher, or the use of a supercomputer. Artemis Financial, can rest easy known its data is safe behind this encryption. To note as well, it will be important to back this encryption with strong passwords, multifactor authentication, as well as firewalls and antimalware scans, run on a frequent schedule.

Like I said before AES implements a few different cyphers each having a higher bit count than the one previous, 128, 192, and 256 bits. AES 256-bit is the strongest encryption, however if on site-security of your data is an issue I would recommend using 128 over 256. Utilizing some basic knowledge here, 128 bit is going to be faster than 256 bit and has a stronger key schedule (the timeframe for which keys circulate and change), meaning on site tampering with data becomes almost impossible. However, 256 is much more resistant to brute force attacks, seeing as it has twice the bits. AES-256 is going to be preferable in most situations and is thus my recommendation.

The way AES works is, it utilizes symmetric encryption which involves converting, “Data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back into its original form, called plaintext” (Dworken et al., 2001). As well, symmetric encryption is desired when encrypting your own data as it has a singular private key to encrypt and decrypt data, meaning the private key never gets shared with the public allowing some data to be potentially compromised, whereas asymmetric uses two, one private and one public. So, using AES 256 if on site-security is sufficient or AES 128 if not, will allow Artemis Financial to have the most secure data possible for their clients.

However, AES-256 alone is not enough to secure Artemis Financial against all types of data theft and tamper. The Secure Hashing Algorithm, or SHA-256 needs to be used in conjunction with AES-256 to completely secure all of Artemis Financials’ data. The way SHA works is, it takes data input as plaintext, then creates a cyphertext a number of bits long which is 256 for SHA-256. Two of the most popular, secure, and time efficient options are Message Digest (MD5) and SHA-256. “In conclusion, we know that the complexity of the MD5 algorithm and SHA256 is equal and the value is Ɵ(N), but the running time of MD5 is faster than SHA256” (Rachmawati et al., 2018) True as this may be MD5 does have known vulnerabilities, as Artemis Financial is a financial institution. I must recommend using SHA-256.

## 

## 2. Certificate Generation

## I am including the actual certificate with my submission file

## Graphical user interface, text, application, email Description automatically generatedA computer screen capture Description automatically generated with medium confidence

## 

## 3. Deploy Cipher

Graphical user interface, text, application

Description automatically generated

## 

## 4. Secure Communications

Graphical user interface, text, application

Description automatically generated

## 5. Secondary Testing

Graphical user interface, text, application, email

Description automatically generated

This screenshot shows an error free execution run, and I also updated Spring Boot and Maven to the correct current version as well.

Graphical user interface, text, application

Description automatically generated**Pre Code Refactor**

**Post Code Refactor**

A screenshot of a computer

Description automatically generated with medium confidence

As you can see both runs worked error free, had different ports so you know it was ran separately, mostly I did this to show that my code introduced no vulnerabilities, since the dependency-check reports show no new vulnerabilities before and after my code was added, as well utilizing the application.properties file allows for us to add HTTPS to further encode data. Shown below. I also added both dependency check reports to my submission file.Graphical user interface, text, application

Description automatically generated

## 

## 6. Functional TestingGraphical user interface, text, application Description automatically generated

There are not any syntax errors in my refactored code, comments are laid out appropriately. There are no logic errors in my code, each piece does one thing and sends it along the line, each variable is called appropriately minimizing security risks, exposing as little endpoints as possible.

## 7. Summary

After reviewing the Security Vulnerability Assessment Process Flow diagram, I am recommending APIs, Input Validation, Cryptography for review (SNHU, NA). According to Paulo Silva who has a PhD in Information Sciences and Technology from the University of Coimbra and an OWASP contributor, Attackers can exploit API endpoints that are vulnerable to broken object level authorization by manipulating the ID of an object… because the server component usually does not fully track the client’s state.” (Silva, 2019). Input Validation becomes important because of the hash function and the checksum value. Finally, cryptography, although it is addressed using RSA with a software certificate in combination with SHA-256. When it comes to adding layers of security to a system its best to think large and then slowly narrow your field of vision as you proceed. Initially highlighting large areas of concern and slowly finding smaller and smaller details to fix until your system remains as secure as possible. For me this usually means thinking about the needs of the customer, in this case a financial institution. The nature of their business ensures that an airtight security is needed.

There are a few common practices to ensure that your system stays secure. The first one being, update your systems and dependencies frequently on a schedule, “Many attackers exploit known vulnerabilities associated with old or out-of-date software” (Synopsis.com, 2020). Finally like I said before its important to always be thinking about security, according to Hyperproof.io “Think security from the beginning: Before creating single lines of code, begin planning how you will integrate security into every phase of the SDLC.” (hyperproof.io, 2022). It’s a concept *Iron-Clad Java* our textbook mentions a lot. When building any system especially more complex systems, every step of the way, security must be at the forefront of a developer’s mind, whilst designing software.

**Sources**

Dworkin, M. , Barker, E. , Nechvatal, J. , Foti, J. , Bassham, L. , Roback, E. and Dray, J. (2001*), Advanced Encryption Standard (AES)*, Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.FIPS.197 (Accessed March 26, 2022)

Schneier, B. (2009, July 30). *Schneier on Security*. Retrieved March 26, 2022, from https://www.schneier.com/blog/archives/2009/07/another\_new\_aes.html.

Rachmawati, D., Tarigan, J. T., & Ginting, A. B. (2018). *A comparative study of message digest 5(MD5) and sha256 algorithm*. Journal of Physics: Conference Series, 978, 012116. https://doi.org/10.1088/1742-6596/978/1/012116 197 (Accessed April 10, 2022)

S. (na). *Vulnerability Assessment Process Flow* [PDF]. Manchester: SNHU.

Manico, J., & Detlefsen, A. (2015). *Iron-clad java building secure web applications*. McGraw-Hill Education.

Silva, P. A. (2019, October 7). *Api-security/0XA1-broken-object-level-authorization.md at master · OWASP/API-security*. GitHub. Retrieved April 15, 2022, from https://github.com/OWASP/API-Security/blob/master/2019/en/src/0xa1-broken-object-level-authorization.md

N.A. (2021, July 1). *Top 10 software security best practices: Synopsys*. Software Integrity Blog. Retrieved April 17, 2022, from https://www.synopsys.com/blogs/software-security/top-10-software-security-best-practices/

Hyperproof Team, (2022, January 27). *Secure software development best practices*. Hyperproof. Retrieved April 15, 2022, from https://hyperproof.io/resource/secure-software-development-best-practices/