****

# CS 305 Project Two

**Practices for Secure Software Report**

**Table of Contents**

[Document Revision History 3](#_heading=h.1fob9te)

[Client 3](#_heading=h.3znysh7)

[Instructions 3](#_heading=h.2et92p0)

[Developer 4](#_heading=h.tyjcwt)

[1. Algorithm Cipher 4](#_heading=h.3dy6vkm)

[2. Certificate Generation](#_heading=h.1t3h5sf) 5

[3. Deploy Cipher](#_heading=h.4d34og8) 5

[4. Secure Communications](#_heading=h.2s8eyo1) 6

[5. Secondary Testing](#_heading=h.17dp8vu) 6

[6. Functional Testing](#_heading=h.3rdcrjn) 7

[7. Summary](#_heading=h.26in1rg) 7

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **[2/10/2022]** | **Randy M** | **Updated the Algorithm Cipher, Certificate Generation, and Deploy Cipher sections.** |
| **1.01** | **[2/11/2022]** | **Randy M** | **Updated Secure Communications, Secondary Testing, Functional Testing, and Summary.** |

## Client



## Instructions

Deliver this completed Practices for Secure Software Report documenting your process for writing secure communications and refactoring code that complies with software security testing protocols.

Respond to the steps outlined below and replace the bracketed text with your findings in your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

Randy Marcelino

## 1. Algorithm Cipher

Determine an appropriate encryption algorithm cipher to deploy given the security vulnerabilities, justifying your reasoning. Be sure to address the following:

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

When it comes to the client Artemis Financial’s needs of encrypting their archive files I would recommend AES or Advanced Encryption Standard. The choice behind AES-256 bit encryption is because of the following “Considering that AES is the only hardware-accelerated encryption algorithm in all reasonably modern processors, choosing any encryption algorithm other than AES-256 will unnecessarily slow down your reads and writes (expect a difference of 2 to 3 orders of magnitude in theoretical RAM-to-RAM encryption speeds) without providing any additional security benefit” (Afonin, 2020). AES is widely accepted and used even by the government and it is one of the most secure encryption methods that also retains its speed of decryption. There are more secure ciphers such as quantum cryptography that transcends binary cryptography; the downfall is that it would take longer to crack and get the data necessary. Of course, no cipher is unbreakable; the AES functions by having not only 256-bit encryption but a public key and a private key; now if an attacker gains access to the private key they can have access to any and all data that you can access with that private key. AES is a complex cipher and one with an interesting history starting in 1997.

AES’s beginnings are as follows “The National Institute of Standards and Technology (NIST) started development of AES in 1997 when it announced the need for an alternative to the Data Encryption Standard (DES), which was starting to become vulnerable to brute-force attacks'' (Bernstein & Cobb, 2021). AES was invented to keep up with the rapidly advancing computer technology and will eventually as CPU’s continue to advance in processing power become obsolete and be replaced by an even more complex cipher just as DES was.

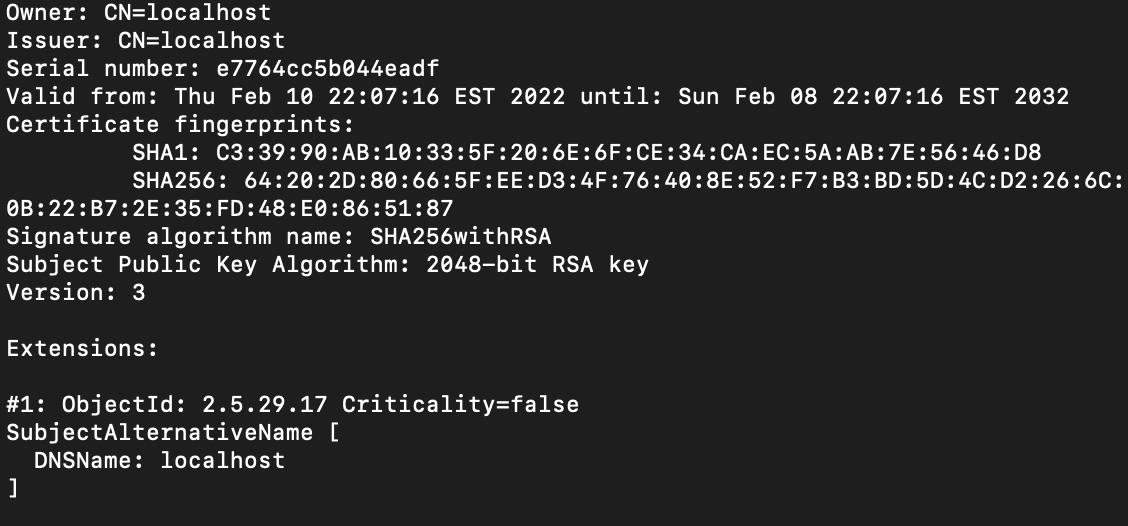
As of now AES can be used in 128, 192, and 256 gigs the more gigs the more secure. The gig differentiator of AES defines how many blocks the data gets encrypted in it also defines the amount of rounds that the text goes through processing which includes substitution, transposition and mixing. Now apart from bit ciphers there is hashing which is in other words a signature signifying that a document has not been altered since its original hash code was created. A way to know that a document has not been interfered with is to check that the hash key is the same as the original that was sent with the file this is were built in decryptors like bitlocker built into the drive come in to play that can quickly decipher files know the internal company hash codes.

AES is considered an asymmetric encryption which is where both parties have a public and private key. The public key is known by all and can be accessed by anyone that is where encrypted information is sent but it can only be accessed by the person who has the private key which is how it is kept secure. On the other hand synchronous encryption is where there is a single shared key amongst the users which is known by all who would use the encryption and only by them. Now where random numbers come into play is with the creation of these keys they allow the keys to remain anonymous by creating random keys. All of these encryption methods are a solid foundation to keeping data safe and secure like archive files for Artemis Financial.

## 2. Certificate Generation

Generate appropriate self-signed certificates using the Java Keytool, which is used through the command line.

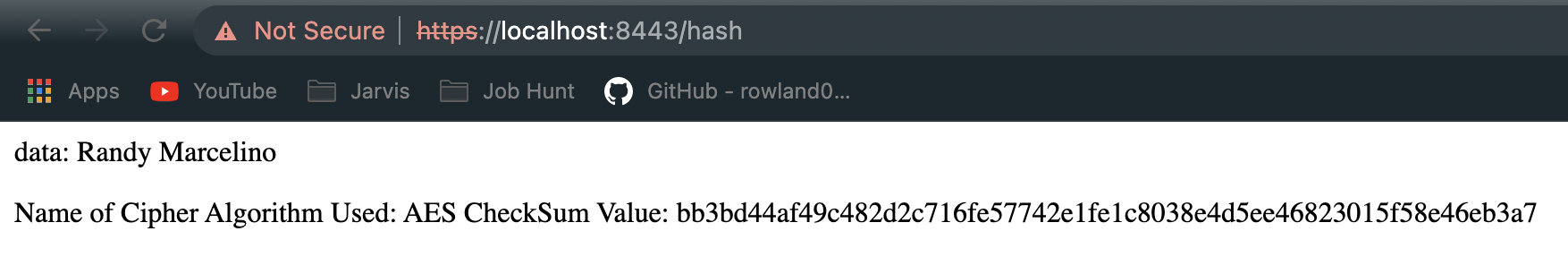
* To demonstrate that the keys were effectively generated, export your certificates (CER file) and submit a screenshot of the CER file below.



## 3. Deploy Cipher

Refactor the code and use security libraries to deploy and implement the encryption algorithm cipher to the software application. Verify this additional functionality with a checksum.

* Insert a screenshot below of the checksum verification. The screenshot must show your name and a unique data string that has been created.



## 

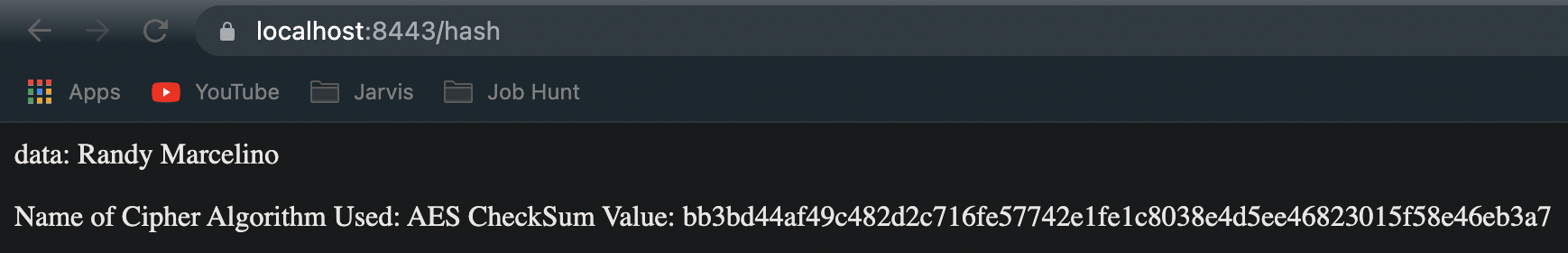
## 

## 

## 4. Secure Communications

Refactor the code to convert HTTP to the HTTPS protocol. Compile and run the refactored code to verify secure communication by typing **https://localhost:8443/hash** in a new browser window to demonstrate that the secure communication works successfully.

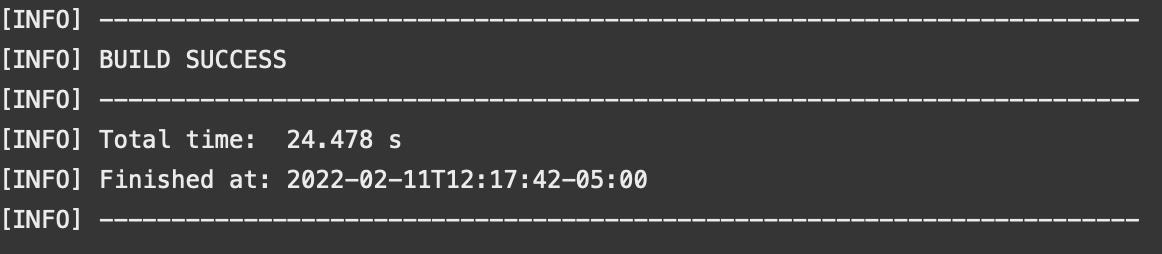
* Insert a screenshot below of the web browser that shows a secure webpage.

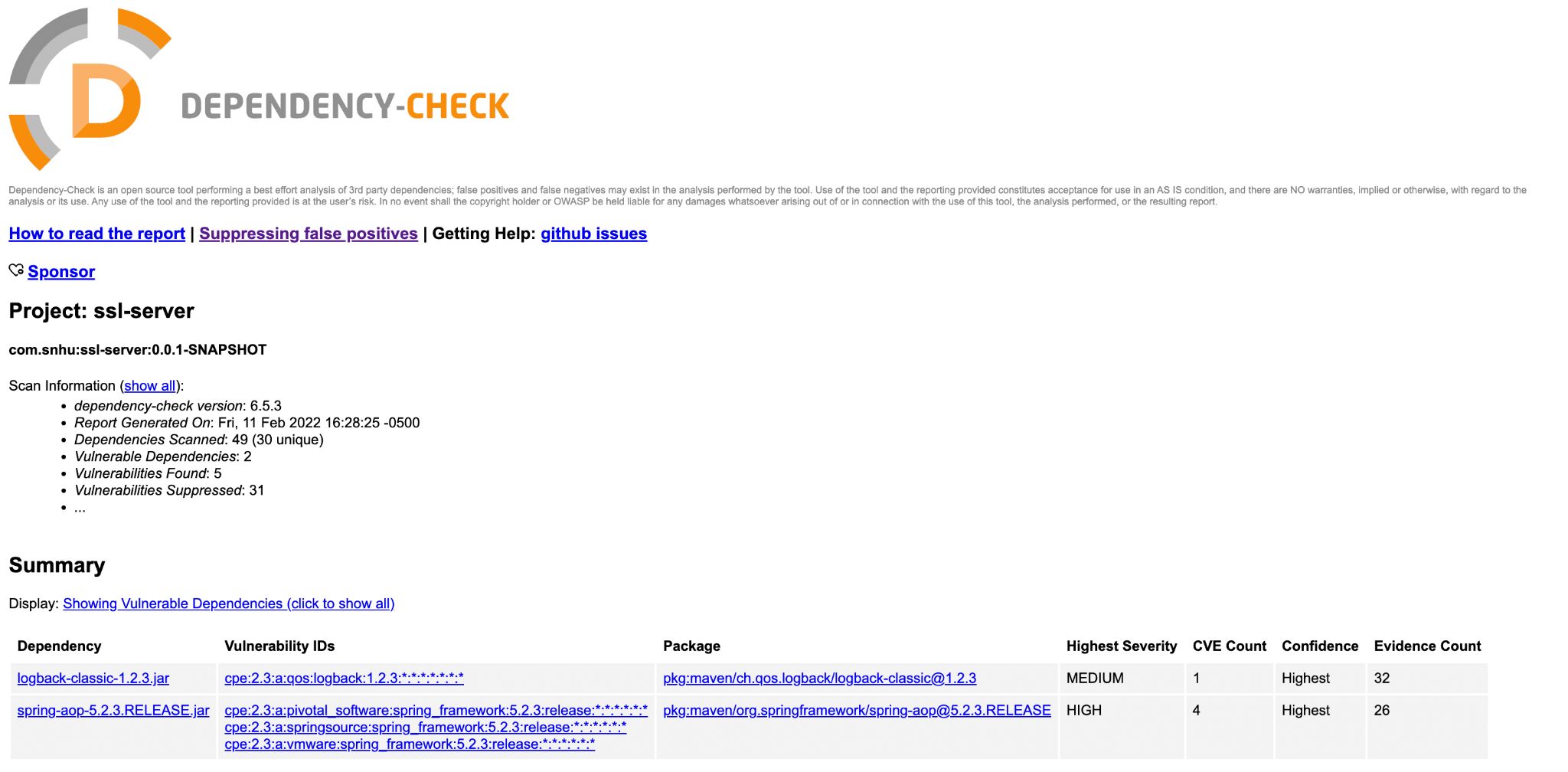


## 5. Secondary Testing

Complete a secondary static testing of the refactored code using the dependency check tool to ensure code complies with software security enhancements. You only need to focus on the code you have added as part of the refactoring. Complete the dependency check and review the output to ensure you did not introduce additional security vulnerabilities.

* Include the following below:
  + A screenshot of the refactored code executed without errors
  + A screenshot of the dependency check report

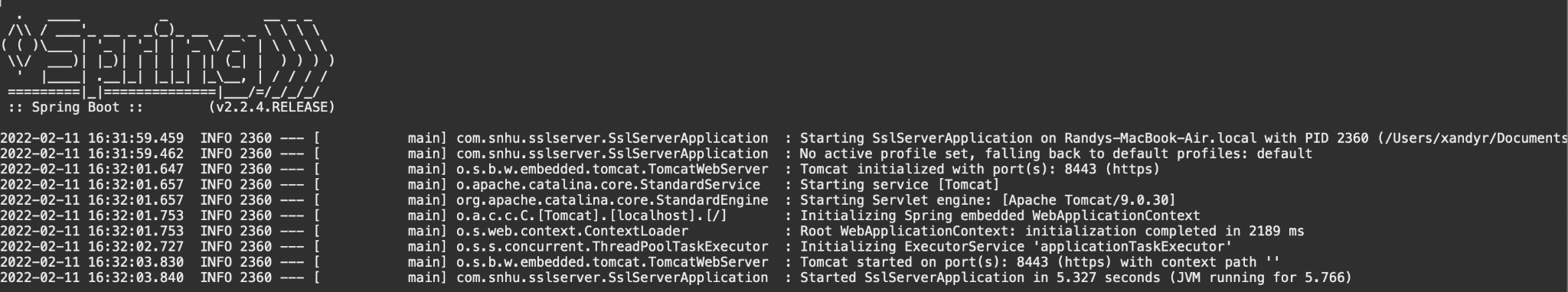




## 6. Functional Testing

Identify syntactical, logical, and security vulnerabilities for the software application by manually reviewing code.

* Complete this functional testing and include a screenshot below of the refactored code executed without errors.



## 7. Summary

Discuss how the code has been refactored and how it complies with security testing protocols. Be sure to address the following:

* Refer to the Vulnerability Assessment Process Flow Diagram and highlight the areas of security that you addressed by refactoring the code.
* Discuss your process for adding layers of security to the software application and the value that security adds to the company’s overall wellbeing.
* Point out best practices for maintaining the current security of the software application to your customer.

Through the refactoring of the code I was able to enhance the code quality, cryptography, code error, client/server, and API’s. By refactoring the code and addressing certain areas of security extra layers of protection are added which can give the users an ease of mind and the company as well since the customers data is safe the companies reputation and integrity is up kept. To keep up this level of security the certificate has to be renewed and kept up to date so as to keep the security of data fresh. The checksum validation should be kept up to date with evolving and changing security requirements. And finally the dependency check should be run periodically to make sure the software is not in danger of new threats.