****

# CS 305 Project Two

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **2.0** | **2/17/2021** | **Matthew Marinelli** | **Final Version 2.0** |

## Client



## 

## Developer

Matthew Marinelli

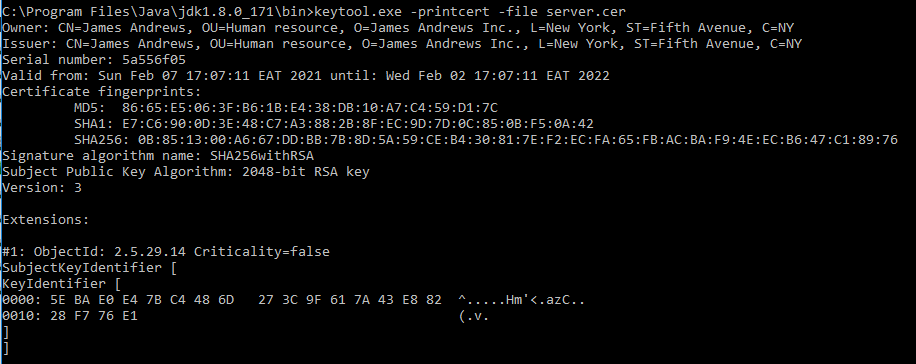
## 1. Algorithm Cipher

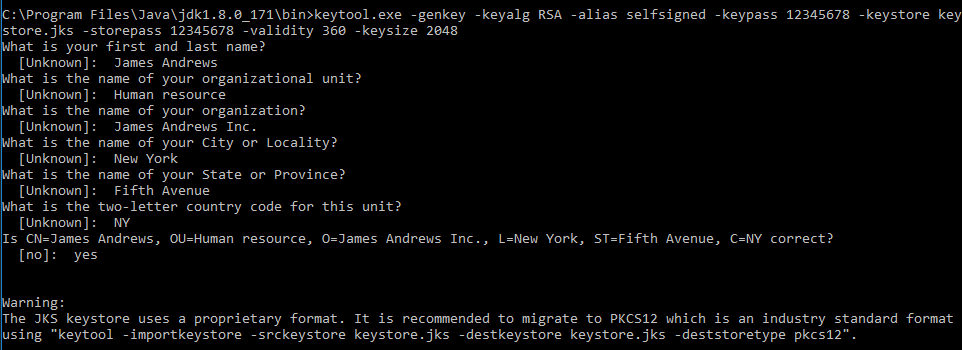
I would recommend the encryption algorithm AES. AES is a symmetric algorithm that encrypts data in 128 bit blocks of data. AES can be classified into three categories based on how many bits are in the key. (128, 192 and 256) I would recommend AES because it’s fast and efficient, requiring one single pass through the data as well as the ability to avoid collision attacks as well as Birthday attacks. MD5 for example, is notorious for being one of the first algorithms that created a hash collision thus being extremely vulnerable.

AES is a symmetric algorithm that encrypts data in 128 bit blocks of data. AES can be classified into three categories based on how many bits are in the key. (128, 192 and 256) I would recommend AES because it’s fast and efficient, requiring one single pass through the data as well as the ability to avoid collision attacks as well as Birthday attacks. MD5 for example, is notorious for being one of the first algorithms that created a hash collision thus being extremely vulnerable.

## 2. Certificate Generation

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





[Insert screenshot(s) here.]

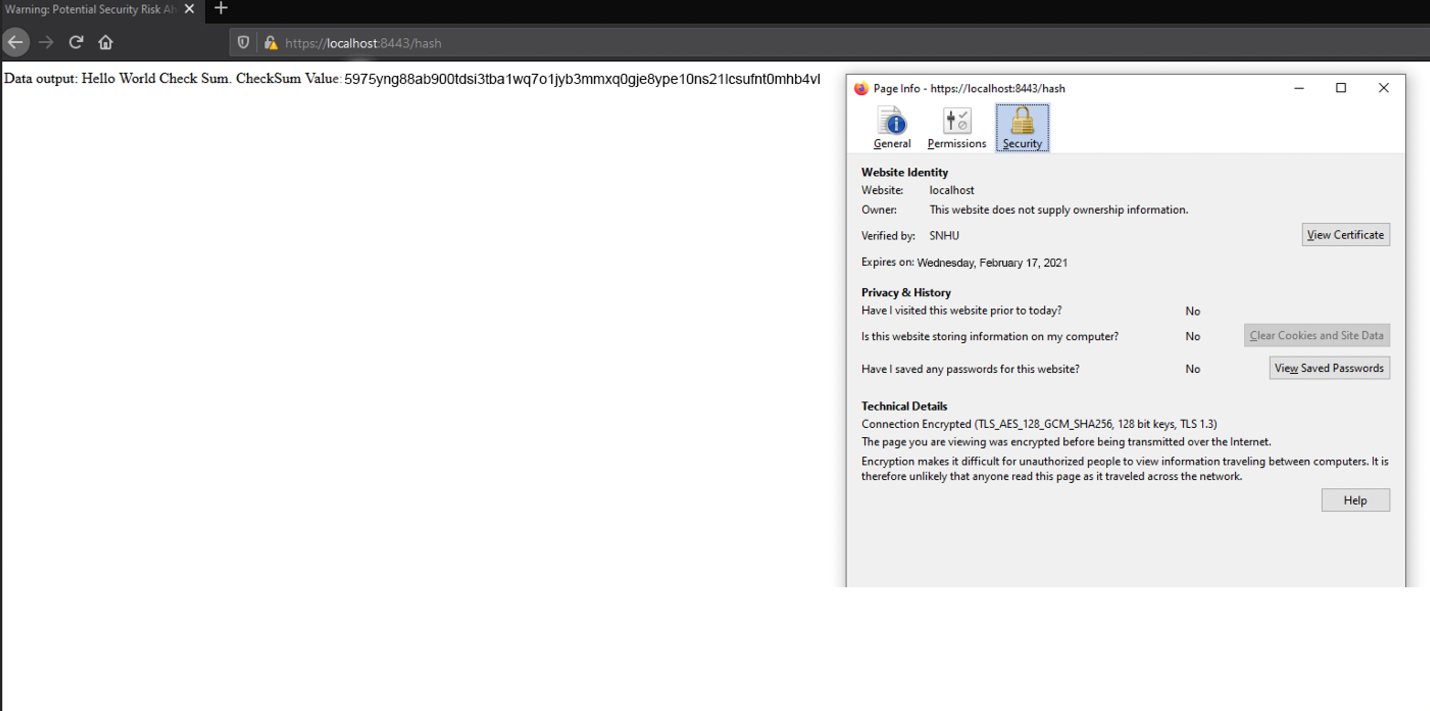
## 3. Deploy Cipher

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.

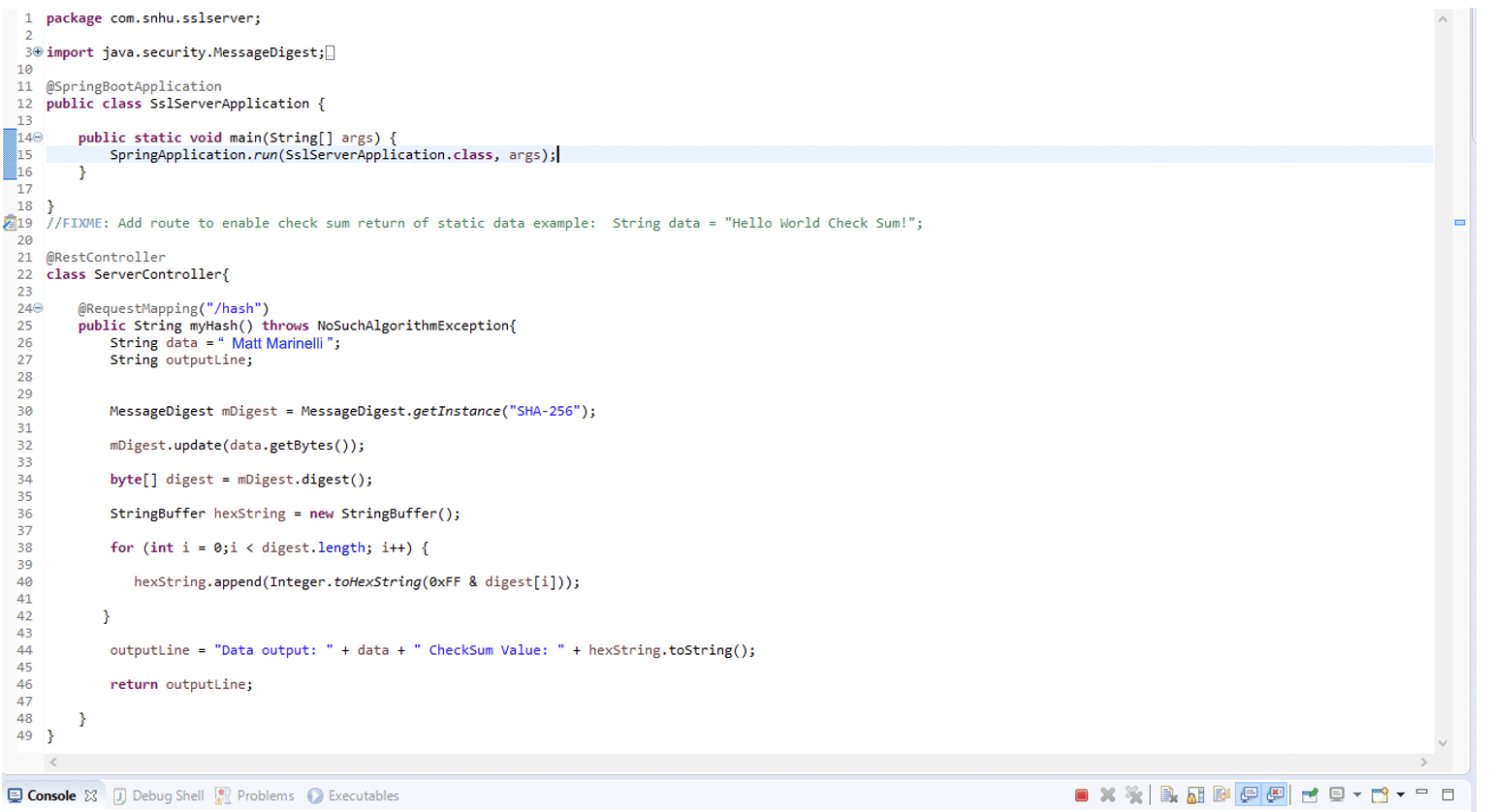
* Insert a screenshot below of the checksum verification. The screenshot must show your name and a unique data string that has been created.

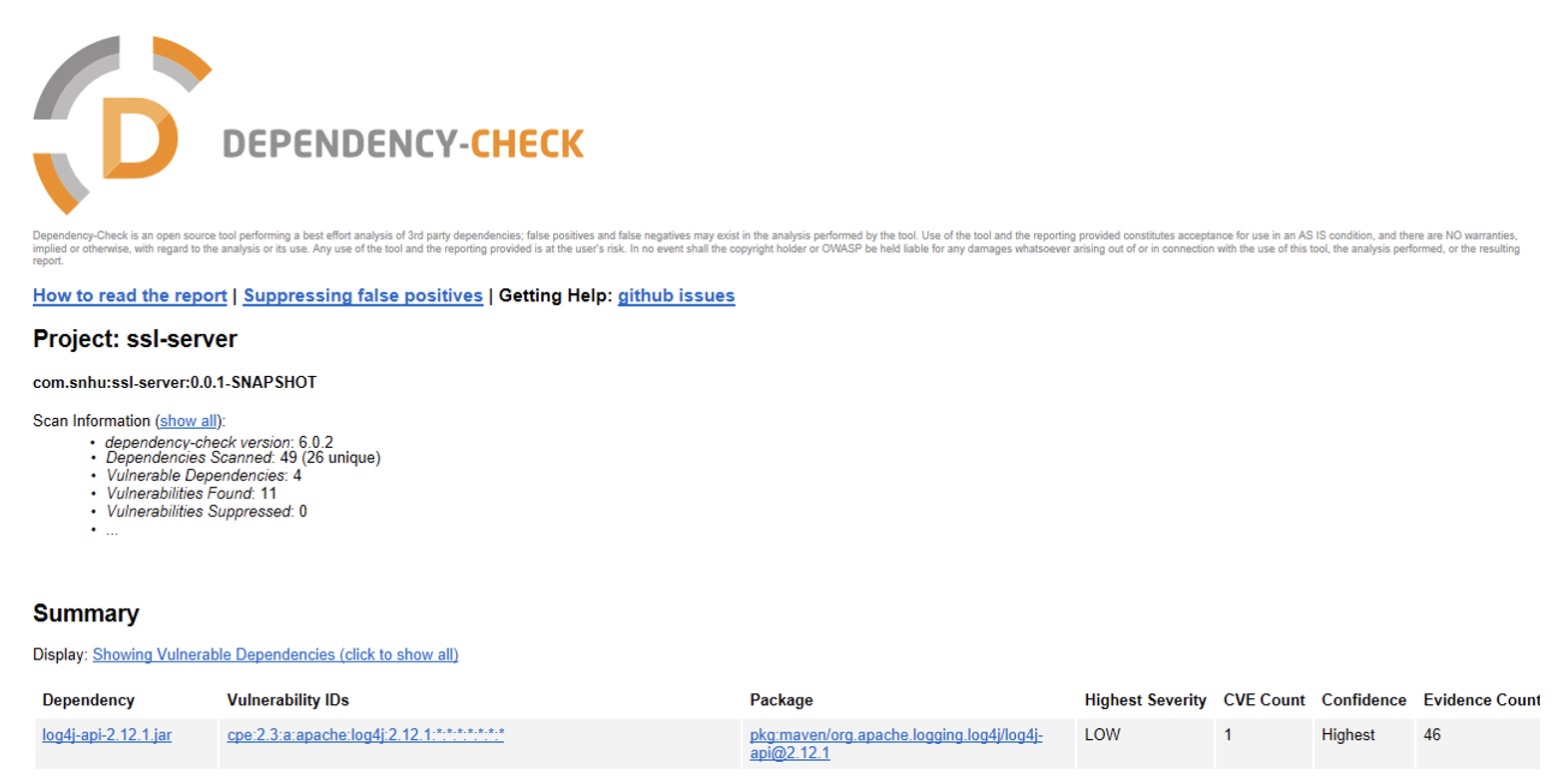
[Insert screenshot(s) here.]

## 4. Secure Communications



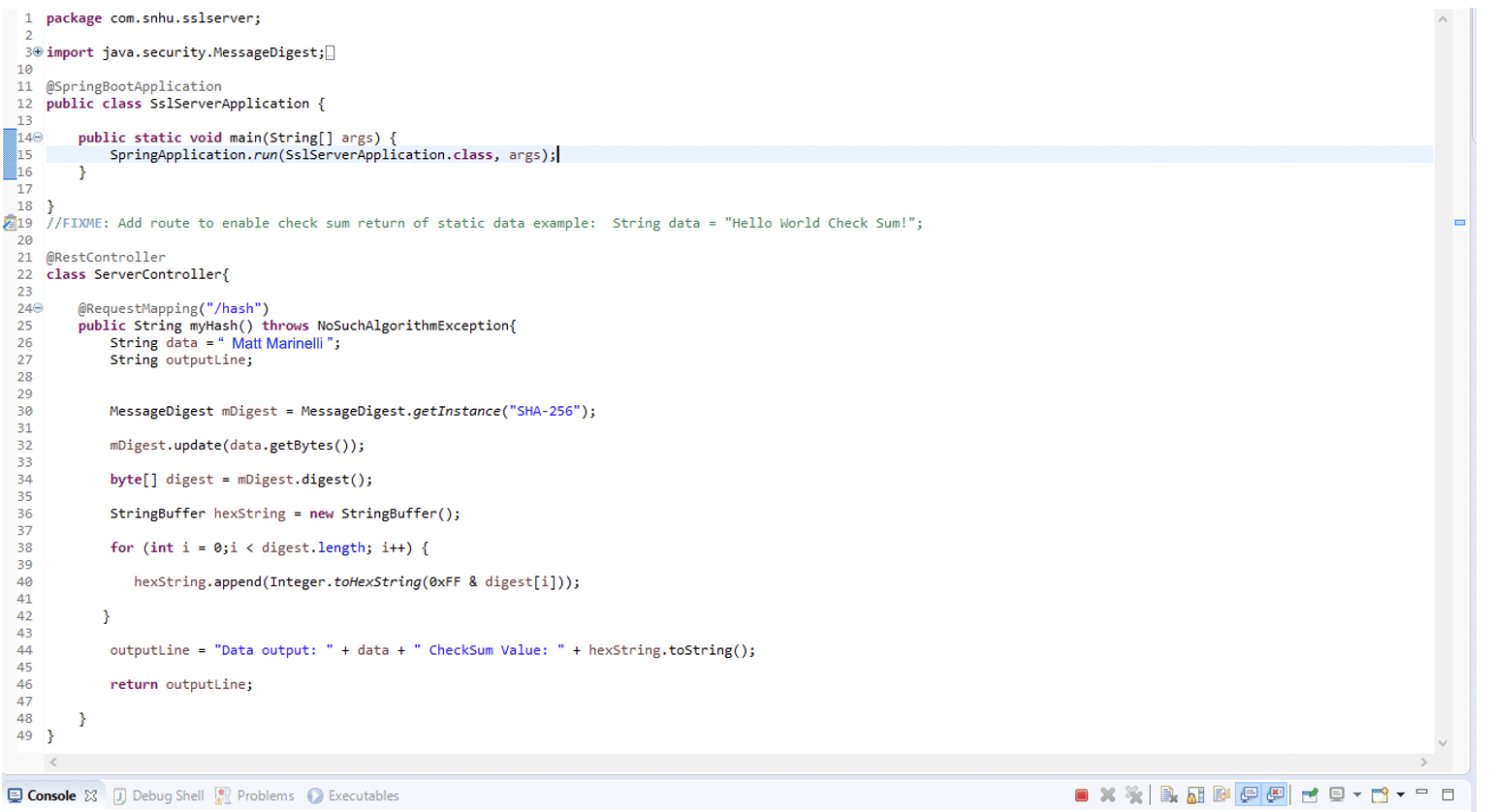
## 5. Secondary Testing





## 6. Functional Testing

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



## 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.

[Include your findings here.]