****

# CS 305 Project Two

**Practices for Secure Software Report**

Table of Contents

[CS 305 Project Two 1](#_Toc1296619603)

[Table of Contents 1](#_Toc446855935)

[Document Revision History 2](#_Toc1862145699)

[Client 3](#_Toc1216908323)

[Instructions 3](#_Toc1877395133)

[Developer 3](#_Toc565273649)

[1. Algorithm Cipher 4](#_Toc186358228)

[2. Certificate Generation 4](#_Toc609846914)

[3. Deploy Cipher 6](#_Toc1004637104)

[4. Secure Communications 6](#_Toc1339140961)

[5. Secondary Testing 7](#_Toc289973029)

[6. Functional Testing 8](#_Toc1148719330)

[7. Summary 9](#_Toc261551045)

[8. References 10](#_Toc67004685)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **4/17/2022** | **Shawn Neal** |  |

## Client



## Developer

Shawn Neal

## 1. Algorithm Cipher

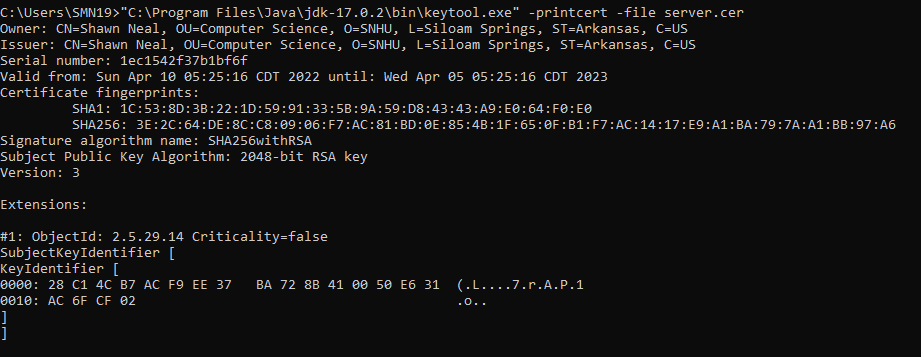
There are several factors to consider when recommending an encryption algorithm for a financial institution. Artemis Financial must consider a diverse range of potential threats, from social engineering, malware, data manipulation, and unencrypted data. (Packetlabs, 2021). As a financial company they also must meet a wide range of regulations and standards on data protection and security ranging from national level regulations such as the UK-GDPR, Canada’s OSFI, of the various regulations in the US such as NIST 800-53, SOX Act, Banking Security Act, Gramm-Leach-Biley Act, or FINRA to international standards such as the ISO 27000, PCI DSS, PSD 2 or the EU-GRPR. (Packetlabs, 2021; Kost,2022).

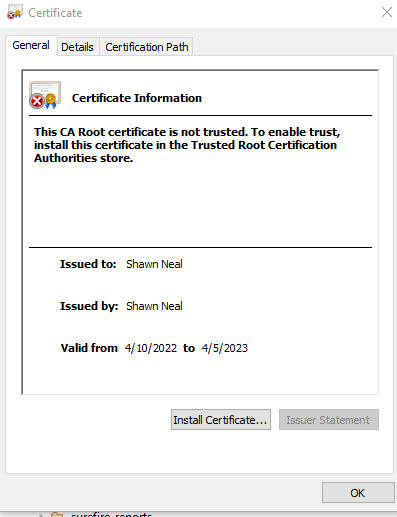
With all this in mind, for an encryption algorithm to preserve financial planning, I recommend the SHA-256 algorithm. This algorithm is part of the SHA-2 (or Secure Hashing Algorithm 2) family of algorithms co-developed by the National Security Agency (NSA) and the National Institute of Standards and Technology (NIST) as part of the SHA-2 family of hashing algorithms that was released in 2002. (Penard & Werkhoven,2008). The algorithm is a hash function, or an algorithm that takes a plaintext message and converts it into a length of characters a specific length, or bit level. (Schatten, 2021) The SHA-256 is a hashing algorithm that converts messages of any length less than2^64-1 bits into a message of 512 bits through a 256-hash value, a type of digital signature, through 64 rounds of encryption. (Penard & Werkhoven,2008; Bashir,2018). One key advantage to the SHA-2 family of algorithms is the incredibly minimal risk of collisions, or unencrypted messages having the same hash value once they are encrypted. (Penard & Werkhoven,2008)

One way that encryption algorithms help prevent collisions and bad faith decryption is the use of padding, or randomized random bits added into the message randomly to help confuse attempts at decryption without the proper key. (Schatten, 2021) Random values are used to introduce unpredictability to the algorithms. (Graham-Cumming, 2020) There are two types of keys in encryption, symmetric and asymmetric. (Daniel, 2021) Symmetric key encryption is where both the encryption and decryption use the same key, whereas asymmetric keys use two distinct keys that are mathematically connected to encrypt or decrypt. Symmetric keys have more security vulnerability, while asymmetrical keys are slower. (Daniel, 2021)

Encrypting sensitive information is an ancient practice, with roots dating back to use by Egyptian religious leaders four thousand years ago, with the Assyrians using to protect financial information a few centuries later and the Greeks using for military purposes a few centuries after that. (Jakob, 2001). In the earliest forms, encryption was simply replacing letters and symbols with different ones, and while the “how” has significantly changed, that remains at the heart of how encryption is still performed today. (Jakob, 2001).

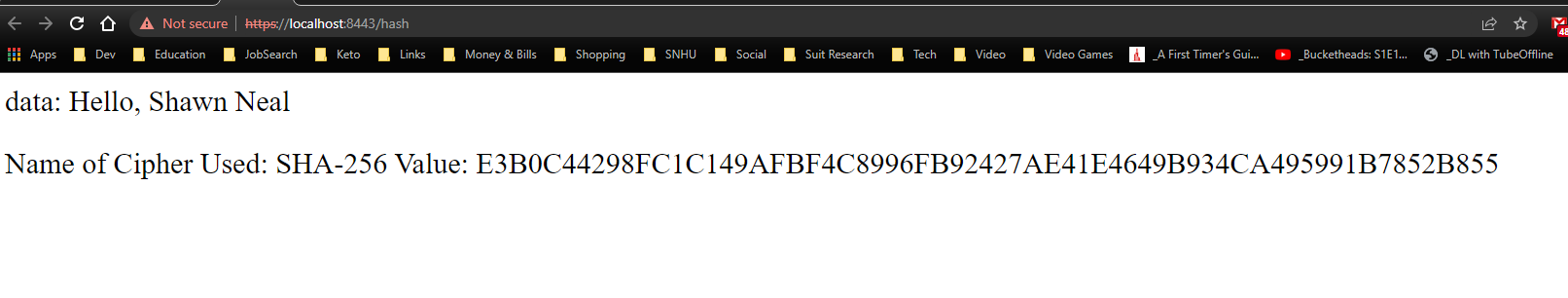
## 2. Certificate Generation

* CER File
* Certificate



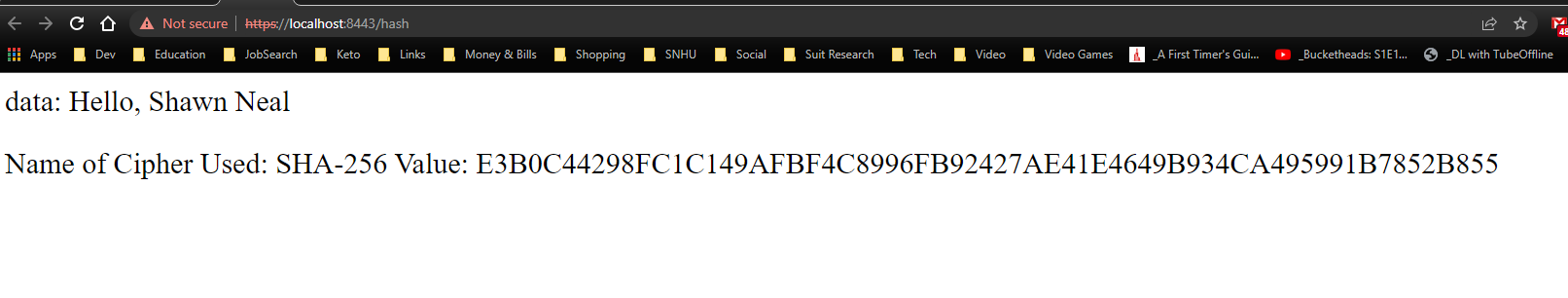
## 3. Deploy Cipher

* Checksum verification.

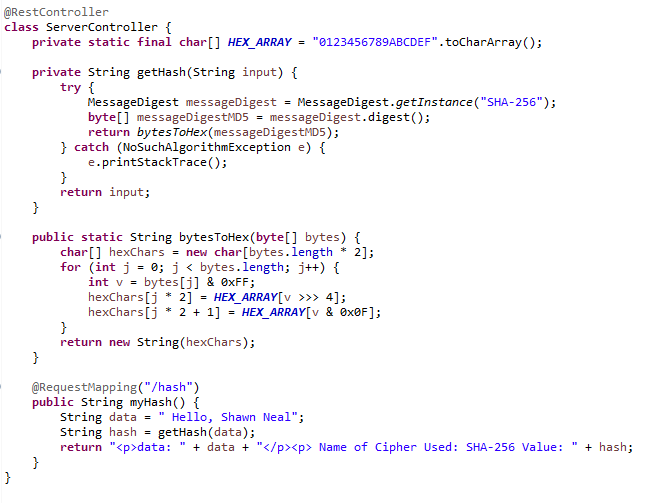
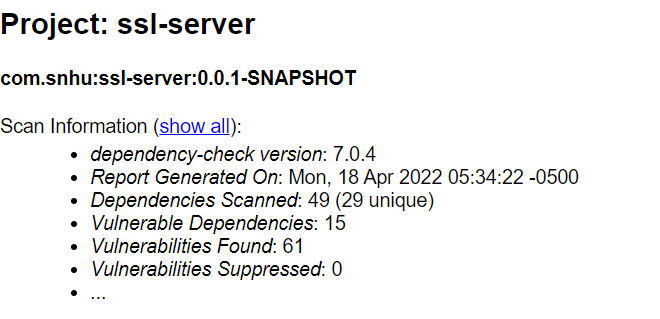


## 4. Secure Communications

* Secure webpage, with an untrusted certificate

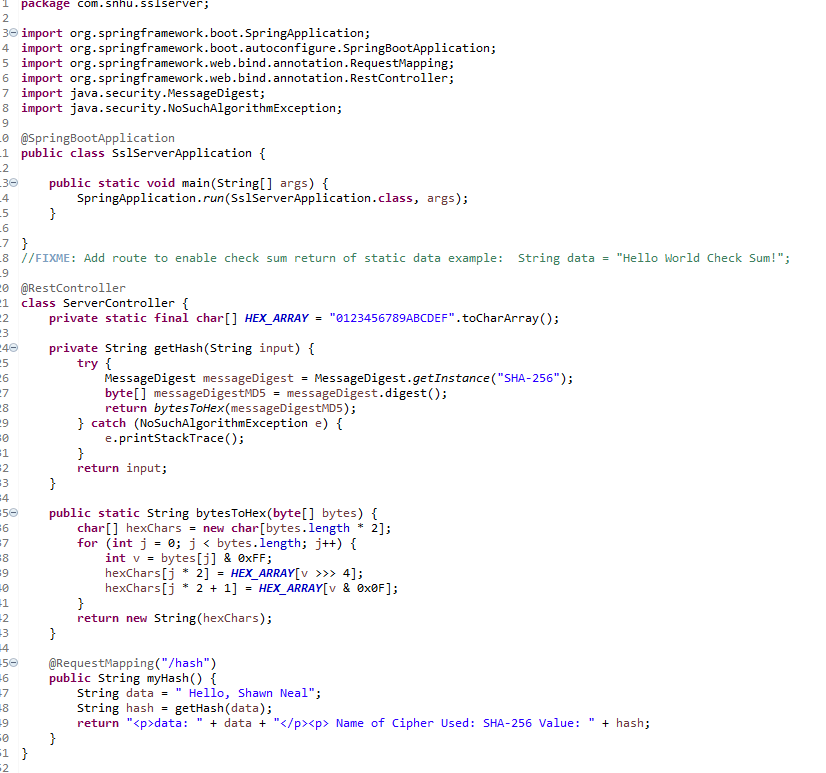


## 5. Secondary Testing

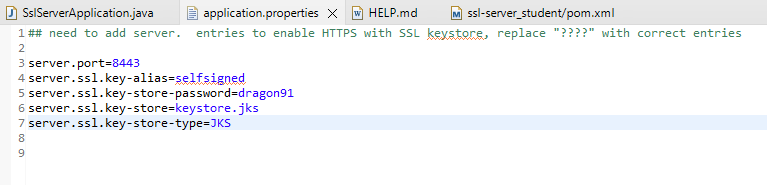
* + A screenshot of the refactored code executed without errors
  + A screenshot of the dependency check report

## 6. Functional Testing

* Full Code



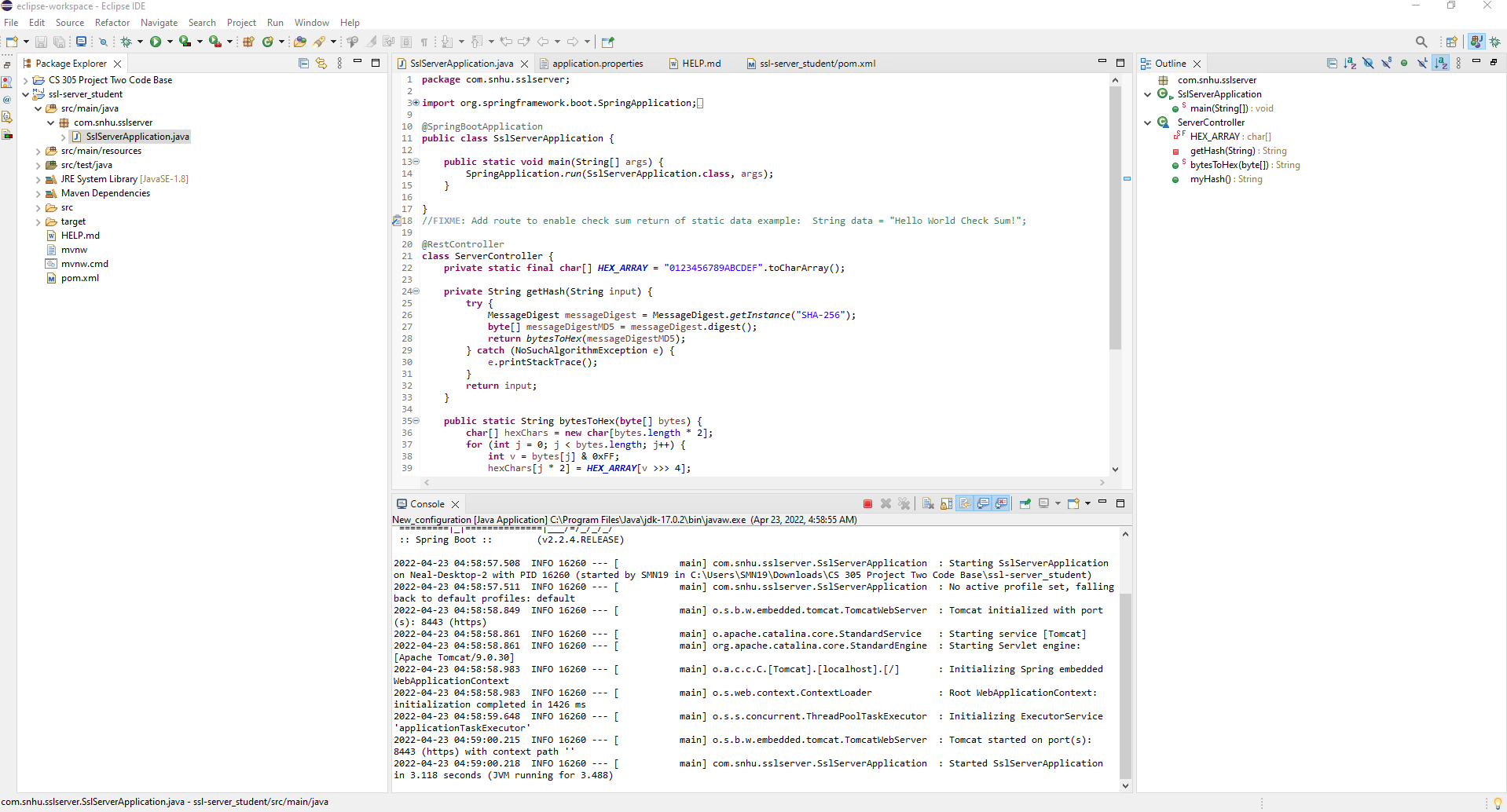
* Application Properties Code



* Pom File Updated Code Section



* App running in Eclipse



## 7. Summary

For this project I primarily address the areas of cryptography, server/client, and code quality in the Vulnerability Assessment Process Flow Diagram. I implemented the SHA-256 hashing algorithm to secure the data before it is stored on the Spring-based server by adding a REST controller to the SSLServerApplication file, which is further protected using an SSL certificate within the application.properties file, and finally I reviewed all dependencies to ensure no false positives remained by updating the version used of Maven within the pom.xml file.

The SHA-256 algorithm is the first layer of security. Being a financial institution, Artemis Financial will experience digital security breaches. This is a statical fact, with 71% of all data breaches being financially motivated. (Dautovic,2022) By encrypting the data first, there is less risk when that data is stolen. The second layer of security is the inclusion of an SSL certificate. This helps the client ensure a secure connection between them and the server.

There are a handful of things that Artemis Financial can do to ensure software security as they continue to build and then support this application. Implementation of secure coding standards are one thing that needs to be considered as this application continues to grow, as well as implementing input validation from any users, and encapsulating all data structures used with the program. Maintaining code quality alongside error free code while ensuring secure API interactions is another thing at should be considered while building and maintaining the application. Finally, maintaining a secure application includes the maintenance and regular review of the cryptographical strategies used. While SHA-256 is an incredibly strong hashing algorithm, as computing and encryption techniques evolve, new vulnerabilities may develop, as well as more secure algorithms. Regular reviews of dependency management should be implemented alongside security reviews to ensure the best, most up-to-date protection is used providing peace of mind for the customers of Artemis Financial and minimizing the risk for their financial information.

## 8. References

ClickSSL. (2012, January 27). *What is bit encryption, and how does it work?* ClickSSL Blog - Information about SSL Certificates & Infosec. Retrieved April 3, 2022, from <https://www.clickssl.net/blog/what-is-bit-encryption-and-how-does-it-work>

Daniel, B. (2021, May 24). *Symmetric vs. asymmetric encryption: What's the difference?* Trusted Computing Innovator. Retrieved April 3, 2022, from <https://www.trentonsystems.com/blog/symmetric-vs-asymmetric-encryption#:~:text=Unlike%20symmetric%20encryption%2C%20which%20uses,'%20and%20recipients'%20sensitive%20data>.

Dautovic, G. (2022, February 11). *Top 25 Financial Data Breach Statistics*. Fortunly. Retrieved April 17, 2022, from <https://fortunly.com/statistics/data-breach-statistics/#gref>

Graham-Cumming, J. (2020, August 17). *Why* *secure systems require random* *numbers*. The Cloudflare Blog. Retrieved April 3, 2022, from <https://blog.cloudflare.com/why-randomness-matters/>

Jackob, M. (2001, August 8). *History of Encryption*. SANS Institute. Retrieved April 3, 2022, from <https://www.sans.org/white-papers/730/>

Kost, E. (2022, March 30). *Top 12 Cybersecurity Regulations for Financial Services: Upguard*. RSS. Retrieved April 3, 2022, from <https://www.upguard.com/blog/cybersecurity-regulations-financial-industry>

Packetlabs. (2021, November 26). *Cybersecurity threats in the banking sector.* Retrieved April 3, 2022, from <https://www.packetlabs.net/posts/banking-and-cybersecurity/>

Penard, W., & van Werkhoven, T. (2008, February). *On the Secure Hash Algorithm family*. Cryptography in Context. Retrieved April 17, 2022, from <https://webspace.science.uu.nl/~tel00101/liter/Books/CrypCont.pdf>

Schatten, J. (2021, March 24). *Rehashing Hashing: What is SHA-256?* SSLTrust. Retrieved April 23, 2022, from <https://www.ssltrust.com.au/blog/what-is-sha-256>

TutorialsPoint. (n.d.). *Cryptography Hash functions*. Cryptography hash functions. Retrieved April 3, 2022, from <https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm>