****

# 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** | **[Date]** | **Alexander DeMarco** | **Initial** |

## Client



## Developer

Alexander DeMarco

## Algorithm Cipher

Artemis Financial has requested Global Rain to modernize their operations. They are seeking Global Rain’s expertise in securing their financial information throughout their systems by adding a checksum mechanism to secure their communication and meet software security requirements.

The Advanced Encryption Standards is a symmetric encryption and widely adopted for securing various databases worldwide. National Institute of Standards and Technology (NIST) developed the AES and has replaced older algorithms such as the Data Encryption Standard (DES) (Daemen & Rijmen, 2002)

Due to Artemis Financial dealings, it is recommended that Global Rain provides the Advanced Encryption Standards using the Secure Hash Algorithm (SHA) – 256bit. The SHA-256 produces a 256-bit hash value, considered the best security for verifying data integrity and authenticity (Eastlake & Hansen, 2011).

In cryptography, generating random numbers is crucial for generating secure keys and enhances security through randomization. SHA-256 provides one of the most secure keys available due to the length of key that it can provide, making it ideal for Artemis Financial needs.

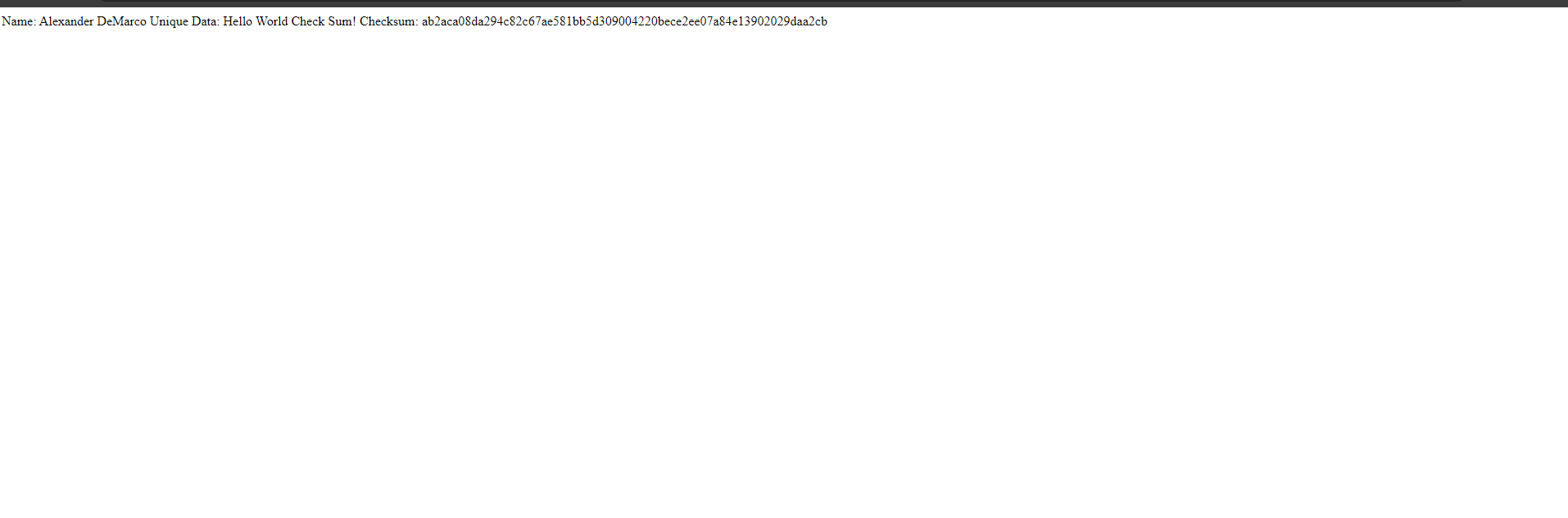
To provide context, symmetric encryption is generally faster and more efficient. Symmetric encryption is generally ideal for encrypting large datasets. It is efficient and generally faster as opposed to asymmetric styles. As such, symmetric provides a pair of keys, a public and private key. Though it is not as efficient as the Symmetric style, it does provide another layer of protection as the process is a key exchange between systems, such as the Rivest-Shamir-Adleman (RSA) algorithms (Rivest, Shamir, & Adleman, 1978).

Encryption algorithms can be simple to complex constructs and are designed to withstand attacks. As mentioned previously, the DES was one of the first developed in the 1970’s and widely used standards for security. However, the DES became susceptible to brute-force attacks as computing power advanced. This became a concern that led to the AES algorithm which quickly became the global standards due to its superior security over DES (Coppersmith, 1994). Today, AES remains the superior algorithm of encryption technology. AES is not only highly secure and efficient, but it is also scalable to protect sensitive data (Ferguson, Schneier, & Kohno, 2010).

## Certificate Generation

Insert a screenshot below of the CER file.



**Connected:  
**

## Secure Communications

## Showing connection with certificate:

A screenshot of a computer

Description automatically generated

## Secondary Testing

## Checksum Controller:

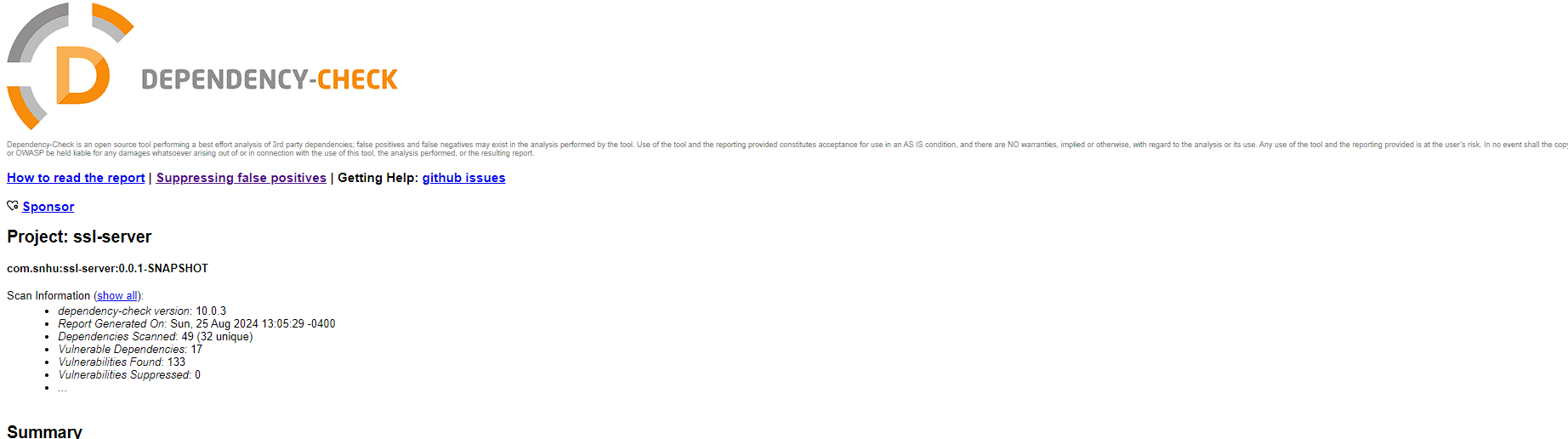
A screen shot of a computer

Description automatically generated

HashUtil:

A computer screen shot of a program

Description automatically generated

Dependency-Check without suppression:  


## Functional Testing

## *Note:* Warnings are present due to JRE version discrepancies (code was designed for JRE 1.8 but running on JDK 22).

A screen shot of a computer

Description automatically generated

## Summary

In summary, Artemis Financials’ software is beginning to shape up by implementing the AES SHA-256 security keys to handle many clients and their data. Certificate generation was self-signed and stored in the KeyStore for the development. The certificate ensures an encrypted environment between the clients and server. SSL was successfully implemented, and connection was established, showing the checksum. The Checksum controller and HashUtil were tested multiple times by connecting with web browsers (Chrome & Edge), both pulling the same result. A Screenshot is provided showing the connection and the self-signed certificate. The refactored code is executed without errors, ensuring that the keys are successfully integrated. Although some warnings are present, it was related to the JDK version differences and did not impact the overall functionality.

## Industry Standard Best Practices

Industry Standard Best Practices included:

* Implementation of the AES with 256 bits
* SHA-256-bit algorithm implemented for hashing purposes to ensure data integrity.
* Self-signed certificate was generated. It is recommended that is changed and issued by a Certificate Authority (CA) to enhance overall security.
* Code was tested and verified through different web browsers. This will ensure that users/clients can have a secure connection through their preferred web browsing methods.

**References**

Coppersmith, D. (1994). The Data Encryption Standard (DES) and its strength against attacks. *IBM Journal of Research and Development*, *38*(3), 243-250. https://doi.org/10.1147/rd.383.0243

Daemen, J., & Rijmen, V. (2002). *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer.

Eastlake, D., & Hansen, T. (2011). *US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)*. Internet Engineering Task Force (IETF). https://doi.org/10.17487/RFC6234

Ferguson, N., Schneier, B., & Kohno, T. (2010). *Cryptography Engineering: Design Principles and Practical Applications*. Wiley.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120-126. https://doi.org/10.1145/359340.359342