****

# 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 6](#_Toc102040759)

[3. Deploy Cipher 6](#_Toc102040760)

[4. Secure Communications 7](#_Toc102040761)

[5. Secondary Testing 8](#_Toc102040762)

[6. Functional Testing 9](#_Toc102040763)

[7. Summary 9](#_Toc102040764)

[8. Industry Standard Best Practices 9](#_Toc102040765)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **08/09/2023** | **Derek Bamford** | **Security Check of Artemis Financial** |

## Client



## Instructions

Submit these completed practices for secure software report. Replace the bracketed text with the relevant information. You must document your process for writing secure communications and refactoring code that complies with software security testing protocols.

* Respond to the steps outlined below and include your findings.
* Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project Two Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Derek Bamford

## Algorithm Cipher

For Artemis Financial I would recommend the secure hash algorithm- SHA-256 to protect their data. The SHA-0 was originally developed by the NSA in 1993 for use in the Digital Signature standard. SHA-1 was part of the U.S. government’s Capstone project, which was a long-term project to develop cryptography standards for public and government use. SHA-2 was the next evolution and again was developed by the NSA in 2001, since then it has become widely use and is considered the industry standard.

An SHA-256 algorithm can be divided into five different steps.

1. Padding Bits: Add one to the first bit and zeros to all additional bits so that the length is exactly 64 bits short of a multiple of 512.
2. Padding Length: Applies a Modulus Value to make the final plaintext a multiple of 512.
3. Initializing the Buffers: Initializes values for eight buffers and stores 64 different keys in an array from K[0] to K[63]
4. Compression Functions: The message is broken down into multiple blocks of 512 bits. Each block is then put through 64 rounds of operation.
5. Output: The final round is then considered the final hash digest.

There are three main types of cypher algorithms -Symmetric, Asymmetric, and Hashing. Symmetric Encryption uses a “key” that requires the sender and receiver to have access to it. For closed systems this type of encryption works best and is also very fast, but the key must be securely stored by both parties and available only to the software that needs to use it. Asymmetric Encryption uses two keys for the encryption prosses instead of one, they are mathematically linked to each other, one is called a public and the other a private key. Blockchain crypto currencies commonly use this type of encryption. Finally, we have hashing which generates a unique mathematical signature called a “hash value” of fixed length for a data set or message. Each message has its own unique hash, and changes to the message can be easily traced by looking at the differences in hash values.

As far back as 600BC in ancient Sparta cryptography, the art of writing or solving codes, was used to write hidden messages. Today modern cryptography uses complex mathematical formulas to hide information from people that should not have access to it. A cypher algorithm is one such formula designed to obscure the value and content of data by generating a key that is used to encrypt the data, the original key or a complementary key is then needed to decrypt the data back its original form.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a certificate

Description automatically generatedA screenshot of a computer program

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

Code of algorithm cypher used to generate checksum.

![A screenshot of a computer

Description automatically generated]()

Checksum verification.

A screenshot of a computer

Description automatically generated

## Secure Communications

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

Setup to allow a secure connection.

![A screenshot of a computer code

Description automatically generated]()

A certificate authority (CA) is a trusted third-party entity that issues Secure Sockets Layer Certificates (SSL). SSLs are protocols for establishing authenticated and encrypted links between networked computers and works by binding the identities of websites to cryptographic key pairs via digital documents known as X.509 certificates. If a website is using a self-signed certificate instead of a certification from a trusted CA, Chrome will display an error since it cannot verify the authenticity of the certificate. As seen in section two of this document the generated certificate is not a trusted certificate and must be installed in the Trusted Root Certification Authority store.

## Secondary Testing

Insert screenshots below of the refactored code executed without errors and the dependency-check report.

Successful Maven Build

A screenshot of a computer

Description automatically generated

![A screenshot of a computer

Description automatically generated]()

## Functional Testing

Insert a screenshot below of the refactored code executed without errors.

I ran a code review using Snyk.io and no issues were found.

See Section 5 for Successful Maven Build

A screen shot of a computer

Description automatically generated

## Summary

The areas of security that I was able to address in the Artemis Financial code bank were as follows.

**Cryptography:** Used a SHA-256 encryption algorithm to secure that data.

**Client/Server:** Created a self-signed certificate for the purpose of enabling an SSL for secure access to the application.

**Code Error:** Added exception handling into the hash algorithm.

**Code Quality:** Performed a code review to look for possible malicious code injection sites. Updated all dependencies and program files.

Security in programming is a lot like securing a house, you would not lock the front door but leave all the windows open. In programming it is better to have layers of security, so that if one fails you can have redundancies in the system. That is why I added both an algorithm cypher and an SSC, so that Artemis would have multiple layers of security.

## Industry Standard Best Practices

The Open Wide Application Security Project (OWASP) maintains a checklist of industry standard best practices for maintaining software application security.

A list of information on a white background

Description automatically generated

While it was unnecessary to perform all these actions, I did manage to use a handful of them in the Artemis code base including Cryptographic Practices, Error Handling and logging, and data protection.

Companies rely heavily on their reputation with their customers and customers expect a certain amount of respect when it comes to their private data. Not only is it good practice to use secure coding when developing a website or application, it’s just an all-around good business. Numerous business entities have had their reputations severely tarnished because of customer data breaches, this is not good for the customer and even worse for the well-being and bottom line of the company.