****

# 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** | **Feb 25,2024** | **Josh Cantu** | **1.0** |

## Client



## Instructions

Submit this 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

Josh Cantu

## Algorithm Cipher

Artemis data should be protected and encrypted. Being a financial company makes them a target for users with malicious intentions. It could be someone trying to gain access to the user account or “prank” this institution. My recommendation would be AES 256. Many government agencies utilize AES. Depending on the data 128-bit can be utilized but I would determine certain requirements to be met to determine if 128-bit or 256-bit should be utilized.

Since these files, we are working with and encrypting are archives, we should not worry about performance. It is best to play it safe and utilize the top encryption with this technology. AES 256 should be used. It will take longer to decrypt if they need to be referenced but if they are being stored in the archive, the thought is they are there for long-term storage.

These two types of encryption types will be nearly impossible to attack by “brute force.” 256-bit will require more processing power due to the length of the encryption. This will need to be taken into consideration when working in the environment of Artemis. Normally, malicious users will try to find a weak point to expose and use as an entry point. This includes utilizing cloud storage. You should know where the “cloud” is located. You should be able to visit the servers and know where your data is being stored.

AES utilizes symmetric encryption. This means that this protocol converts plaintext to ciphertext using the same secret key. This key is used to decrypt and encrypt information. This also means that there are two keys when using this encryption technology. A private key and a public key. If information is being transferred outside of geographic boundaries, then RSA will need to be added to the environment. It is ideal to use both technologies. You get the protection from RSA with the performance from AES.

This does not mean that AES is the end-all-be-all encryption technology. There are some threats to be aware of. If the system is configured incorrectly then there is an opening in the keys. You can use generic keys to be able to gain access. A second known opening is a minor threat. It deals with 128-bit encryption. There is an 8-round version and the attack deals with that version. It is best to utilize the 10-round version.

Also, consider educating the staff on how to protect themselves and the information stored. This can be done by having “Information Security Awareness Training”. Have the IT staff implement strong password requirements for user accounts. Multifactor authentication should be set up. Education is important because it prevents users from falling victim to social engineering and phishing attacks. The best wall can be built but if the malicious users are let in by the people inside it defeats the purpose of having built this wall. (Bernstein & Cobb, 2021)

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

Description automatically generated

A screenshot of a certificate

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screen shot of a computer program

Description automatically generated

## Secure Communications

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

[Insert screenshots here.]

## Secondary Testing



## Functional Testing

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

[Insert screenshots here.]

## Summary

The web traffic was sent over utilizing HTTPS. This prevents websites from having spillage and malicious users acquiring any information that may be broadcasted. This connection was secured using 128-bit cipher and signed with a personal certificate. This is a temporary measure and should be replaced with a trusted CA root-type certificate.

The API was designed to return a checksum. The checksum may be used to check the footprint of a digital object. It is pushed through a hashing algorithm which generates a hash. An optimal designed hash will have no issues or collisions. There should be no matching hash. Each hash produced by the algorithm should be unique. Hash is used by a person to verify the legitimacy of the file and ensure it has not been manipulated in any way.

If a collision were to occur, it would mean that the checksum has been spoofed. The code was also passed through a static analyzer to check for vulnerability.

## Industry Standard Best Practices

There is no single list of best practices for web-based code. There are things you can do to better protect yourself from malicious users.

* Ensure all user inputs are validated and sanitized to prevent malicious code injection.
* Utilize proper validation techniques. This will prevent buffer overflow, SQL injection, and other different types of attacks.
* Implement authentication for users.
* Protect user data by utilizing encryptions and hashing when they are maneuvering through network storage.
* Ensure software has important security patches.
* Follow proper and secure coding.
* Review code often. Testing the code should be done as the code is being developed.

Maintaining a secure mindset will allow best practice to be followed while the code is being developed. Security flaws will be reduced, these flaws will be found early during the development of the software. This will reduce the time and resources required to go back into the code to patch them. For some more senior developers, this may be a hard habit to break but it will be worth it in the back end to avoid any security flaws that may be missed during development.

# References

Bernstein, C., & Cobb, M. (2021, September). *Advanced Encryption Standard (AES)*. Retrieved from Tech Target: https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard