

**Practices for Secure Software Report**

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| **1.0** | **2/25/2023** | **Avery Sulker** |  |

**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**

Avery Sulker

* **Algorithm Cipher**

Non-symmetric keys employ two keys, one for data encryption and the other for data decryption.

A key to one of these

generally public, allowing system users from outside the system to make messages or questions.

The

Responses are always protected since they typically involve the usage of a second key, which generally remains private.

sensitive information about a person or business.

Symmetric keys employ a single key for both encryption and decryption.

and data decryption.

Both the client and the server must have private keys in order to use a symmetric key.

information.

Given the size of the application we are working on and the security flaws it has, the

The Advanced Encryption Standard (AES) is the most suited encryption algorithm cipher to use in

in combination with a SHA-256 hashing algorithm.

AES is the most reliable encryption method available because it is

by the majority of financial institutions, even the federal government of the United States.

There are several

Applications that employ this cipher technique for encryption are commonplace nowadays.

Symmetric keys are used by AES.

To encrypt and decode data, use a key (which can be 128, 192, or 256 bit).

There are 2256 possible results when using Secure Hashing Algorithms(SHA)SHA-256 as a hash function.

The encryption algorithm cipher utilized is AES, making it nearly hard for any attempts to

Collisions are extremely infrequent, and important data is hindered.

This is not just the safest option for

The option that is ideal for both customers and programmers.

(The number of possible hash combinations is 1.1579209e+77.)

(There are 1.1579209e+77 potential hash combinations.)

One of the most secure options for data encryption right now is AES.

Common Data Encryption

The United States used to utilize (DES) as their default encryption method, but multiple

getting broken repeatedly.

It is said that this is because the key is just 56 bits long.

This amounts to far less

compared to AES.

* **Certificate Generation**



I had been preemptively generated this in preperation for the assignment as I had taken the course before and did not want to invalidate my stores

* **Deploy Cipher**

Insert a screenshot below of the checksum verification.

[Insert screenshots here.]

* **Secure Communications**



* **Secondary Testing**







some problems I could see where there was no controller in the sslServiceApplication.java file and no binding to make sure it was running when using the /hash subpage. implementation of the SHA-256 cipher as well, owing to its longevity and dependability.

⦁ Summary

While refactoring the code base, I was able to handle a number of operational security concerns and guarantee the codebase's quality.

I started out by making sure API interactions were done securely using HTTPS communication protocols. After that, I was the driving force behind the provision of safe error management so I could use the algorithm cipher. By storing sensitive information in environmental variables instead of storing it, I am adhering to the safe coding practices I use in the workplace, and refrain from saving them as simple text.

instead. The addition of multilayered protection was simple to do. I can verify what the application program interface (API) returned against the expected outputs after launching the app and calling the API endpoints. I then conducted a manual code evaluation by surveying the code base. Here, it was crucial to apply a particularly sharp lens to issues like application properties, which, in the case provided, were distributed as plain text but should, instead, have been stored in environment variables.

The Secure Hash Algorithm 256 bit was chosen as the best cipher for these particular requirements after much consideration from a previous assignment, so it was necessary to build an endpoint in ServerController. This specific endpoint will accept data but will output a checksum number generated by the SHA-256 hash function.

Industry Standard Best Practices

Best practices frequently imply the difference between an unavoidable security breach and a secure, dependable codebase. The goal of best practices should be to eliminate or reduce threats to sensitive or essential information. This will improve the company's standing with customers while also avoiding the legal and financial difficulties that may result from a security breach. For best practices, we can turn to industry standards and occasionally to federal regulatory standards. One example of this that comes to mind is the fact that almost all businesses must use the data from our credit and debit cards. If at all possible, stringent guidelines must be implemented worldwide to ensure the security of this data.

We can refer to the Payment card industry's data protection standard in this industry's situation. Which was created with these pressing requirements in mind. Additionally, we ought to strive to promote these practices and a work environment that is supportive of them. Making people aware of the serious harm that compromised data can do to their business can be the first step in doing this.