

**CS 305 Project Two**

**Practices for Secure Software Report**

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| **1.0** | **6/19/2020** | **Stephen** |  |

**Client**



**Instructions**

Deliver this completed Practices for Secure Software Report documenting your process for writing secure communications and refactoring code that complies with software security testing protocols.

Respond to the steps outlined below and replace the bracketed text with your findings in your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

**Developer**

Stephen Cardone

**1. Algorithm Cipher**

Determine an appropriate encryption algorithm cipher to deploy given the security vulnerabilities, justifying your reasoning. Be sure to address the following:

* Provide a brief, high-level overview of the encryption algorithm cipher.
* Discuss the hash functions and bit levels of the cipher.
* Explain the use of random numbers, symmetric vs non-symmetric keys, and so on.
* Describe the history and current state of encryption algorithms.

The encryption algorithm ciper was has been a key aspect of computing in the modern era. The primary purpose of an encryption algorithm is to secure data. The encryption alogrithm does this by using some extreme mathematics to scramble data to the point where it is unrecognisible. The data can not be interpreted unless someone has the proper "key". The key is typically a hexidecimal number and can vary in size. When the scrabled up information is decoded using the key, the content is returned to a readable state. There are 2 primary cases where encryption is used. Firstly, any data communication. While in transit, there is nothing stopping someone from intercepting your message. Using a encryption cipher means the interceptor can not interpret the data. The second case is storing data. When you store sensitive data, you do not want to store it in plain text. It should be stored in an encrypted state that requires a key to read it.

Hash functions are used to abstract the data itself to ensure that the data cannot be read without the proper key. The bit levels of the cipher indicate the size of the key, and in turn represent the complexity and length of calculations to crack the code. The Higher the bit level the better.

There are 2 common key systems: symmetric and non-symmetric. With a symmetric key, the same key is used to encrypt, and decrypt the data. Anyone who has access to the key will be able to read the data encrypted by that key. On the contrary, non-symmetric keys do not use the same key for encryption and decryption. One key is used to encrypt the data, while another key is needed to decrypt the data. Non-symmetric keys have a private and a public key. The public key is used to encrypt the data and only the private key can decrypt the data. This ensures higher security assuming you are able to keep the private key from being compromised.

As computing power increases, existing encryption algorithms become less and less secure due to the fact that the computers are more efficent at "cracking" the cipher. Because of this, encryption is an ever-changing technology. An example of this is the SHA encryption technology. SHA1 has been used for a long time. Eventually a conflict was found in SHA1. To resolve this, SHA2 was created which uses the same algorithm as SHA1 but increses the sizes of the input and output to make them more complex. That is a fine solution, but since computers improve rapidly, it's only a matter of time before SHA2 will also have conflicts. A newer alternative is SHA3 which is a completly new encryption algorithm that is very secure.

Artemis Financial's goal is to "apply the most current and effective software security". Because of this, I am reccomending that Artemis Fincncial use the SHA3-512 encryption algorithm.

**2. Certificate Generation**

Generate appropriate self-signed certificates using the Java Keytool, which is used through the command line.

* To demonstrate that the keys were effectively generated, export your certificates (CER file) and submit a screenshot of the CER file below.

[Insert screenshot(s) here.]

**3. Deploy Cipher**

Refactor the code and use security libraries to deploy and implement the encryption algorithm cipher to the software application. Verify this additional functionality with a checksum.

* Insert a screenshot below of the checksum verification. The screenshot must show your name and a unique data string that has been created.

[Insert screenshot(s) here.]

**4. Secure Communications**

Refactor the code to convert HTTP to the HTTPS protocol. Compile and run the refactored code to verify secure communication by typing [**https://localhost:8443/hash**](https://localhost:8443/hash) in a new browser window to demonstrate that the secure communication works successfully.

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

[Insert screenshot(s) here.]

**5. Secondary Testing**

Complete a secondary static testing of the refactored code using the dependency check tool to ensure code complies with software security enhancements. You only need to focus on the code you have added as part of the refactoring. Complete the dependency check and review the output to ensure you did not introduce additional security vulnerabilities.

* Include the following below:
* A screenshot of the refactored code executed without errors
* A screenshot of the dependency check report

[Insert screenshots here.]

**6. Functional Testing**

Identify syntactical, logical, and security vulnerabilities for the software application by manually reviewing code.

* Complete this functional testing and include a screenshot below of the refactored code executed without errors.

[Insert screenshot(s) here.]

**7. Summary**

Discuss how the code has been refactored and how it complies with security testing protocols. Be sure to address the following:

* Refer to the Vulnerability Assessment Process Flow Diagram and highlight the areas of security that you addressed by refactoring the code.
* Discuss your process for adding layers of security to the software application and the value that security adds to the company’s overall wellbeing.
* Point out best practices for maintaining the current security of the software application to your customer.

[Include your findings here.]