

**Practices for Secure Software Report**

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| **1.0** | **8/23** | **Grant Sorenson** | Implemented secure encryption and decryption via AES as well as checksum verification |

**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**

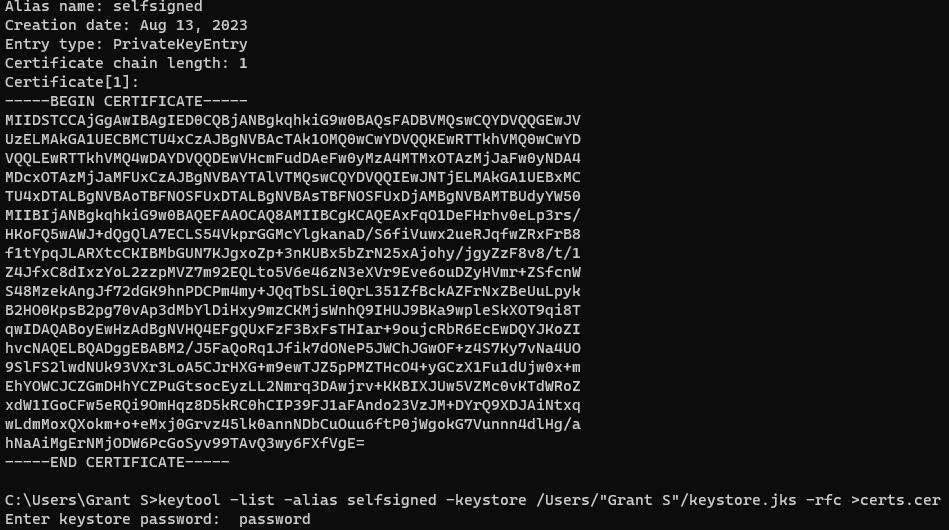
Grant Sorenson

* **Algorithm Cipher**

The Algorithm Cipher I would recommend to Artemis Financial is AES or Advanced Encryption Standard. AES is widely recognized as a reliable and secure cipher that supports multiple key lengths which positively correlate to security and strength. The Cipher can take information blocked into specified bit sizes, encrypt the information through the use of a key that utilizes random number generation and algorithms, and decrypt the same data into a readable format for secure file transfer protocol. The information stored in the specified encrypted data is run through multiple hash functions and transferred into obscure text. For example, imagine an empty vending machine, it has an x and y axis for selecting row and column, in which each x,y pair is named and can have an amount of items ranging from 0-n. When the data is hashed, the hash places the data into a slot in the vending machine which can be retrieved later using the value of that slot in the vending machine. More data is used to narrow down the exact position, and you can then extract it from the vending machine as the original data you sent. While this analogy may not be completely accurate, it goes to explain how hash functions are necessary when sorting through gigabytes of data and how they can be effective to obscure data while keeping access limited to 1 key. AES is a symmetric key algorithm, so the key used to encrypt the data is the only key that can be used to decrypt it\*(Unless it has been compromised). While algorithm ciphers have become more popular in the last few decades due to the reliance on computer technology and online banking and communications infrastructure, there is still a need to keep up to date with the most current security solutions. AES has yet to see any major infiltrations and is constantly updated due to it being the standard within many high-ranking government bodies. This makes it a reliable and consistent data encryption solution.

* **Certificate Generation**

Insert a screenshot below of the CER file.



The certificate was generated and saved to a file called certs.cer which appeared in the directory after the command.

* **Deploy Cipher**

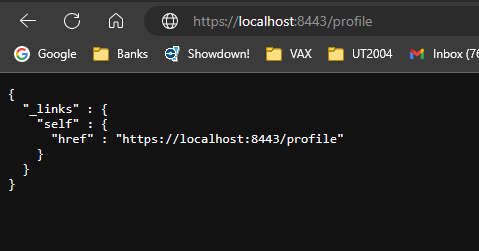
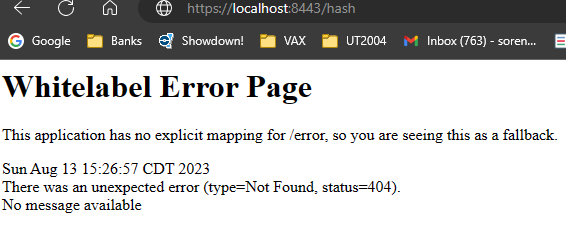
Insert a screenshot below of the checksum verification.



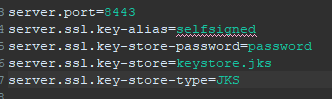
The Cipher has been implemented, protected data practices have not been used like making the classes and data private and using class functions to retrieve sensitive values. The data string is encrypted, decrypted, and a checksum is created.

* **Secure Communications**

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



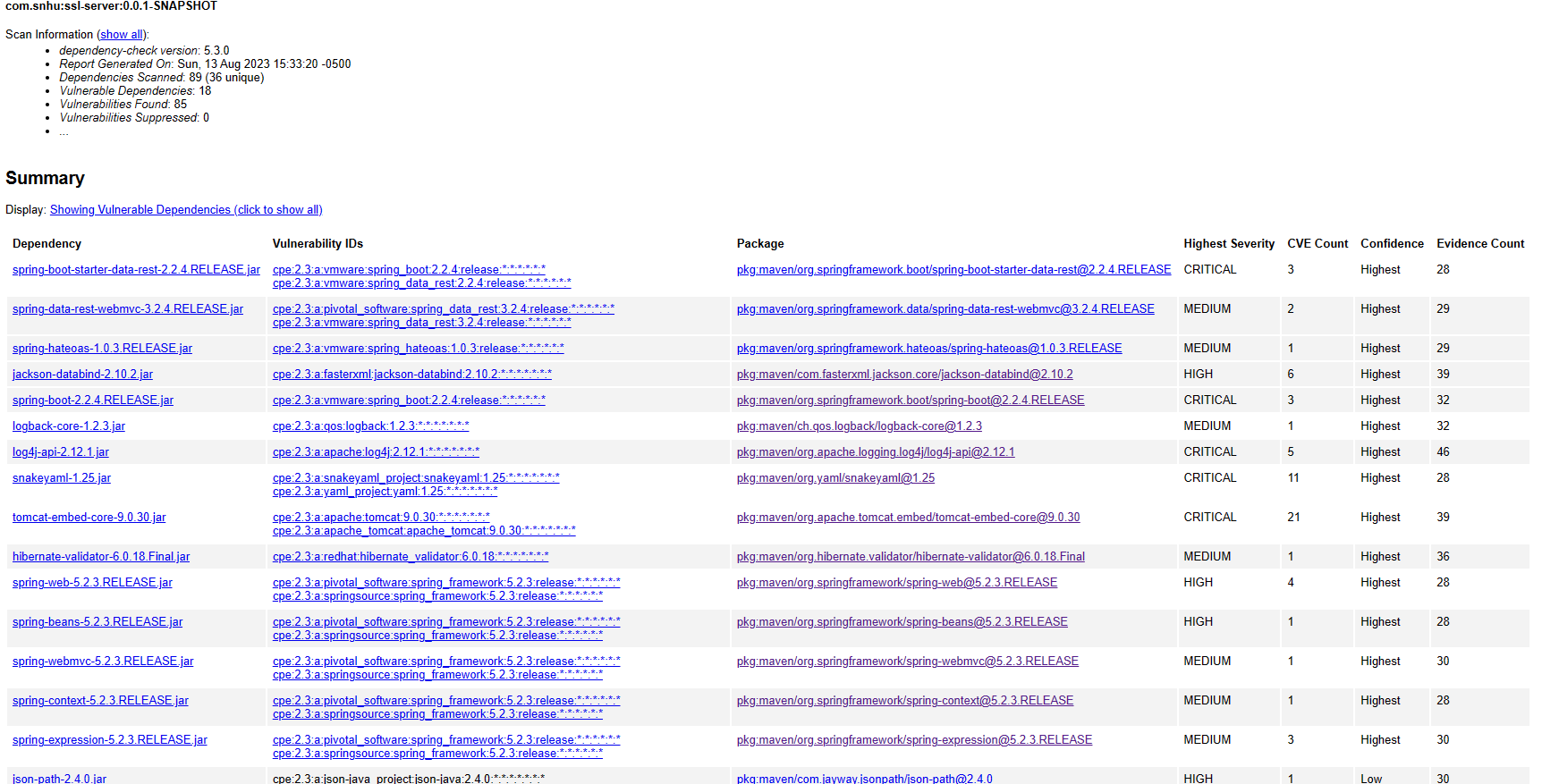
The Server is running and the connection is secure. The fallback page shows up when navigating to /hash but the profile page displays the general hierarchy. The applications file was modified with correct values and navigated to the correct page and accepted the keystore file.



* **Secondary Testing**

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

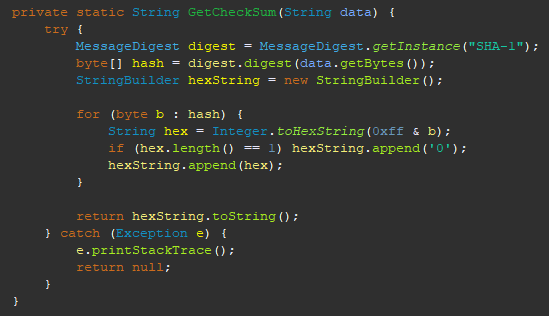
Dependency check report. No additional Errors were discovered after refactoring code and implementing algorithms.



main



checksum encrypt and decrypt

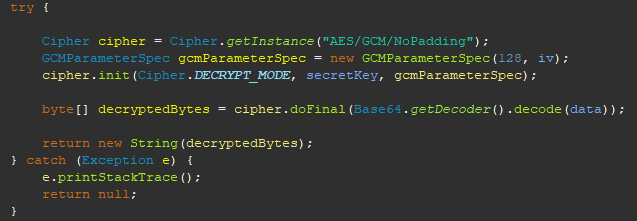


* **Functional Testing**

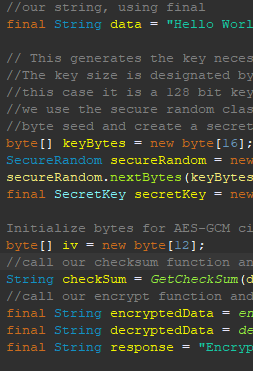
Insert a screenshot below of the refactored code executed without errors.

Originally while creating the encrypt and decrypt functions they were set to public. This was wrong because it would allow members of other classes to access them and not properly protect the data that was being used. By making them private, we restrict access to those calls to objects of the same type and ensure our data is properly protected. To further protect these, we would make the datatypes of the variables private and include getter and setter functions, proper constructors, or other methods for secure data encapsulation. Try and Catch blocks were also added for further error protection.





* **Summary**

While considering the safe transfer of data, security is of utmost importance and should be improved as much as possible. Some areas where I made security improvements while refactoring the code included areas like code quality, code error, cryptography, and encapsulation. I tried to organize and document my code so it was as clear as possible. Readability and enhancability go hand in hand while developing secure software. When a system is easy to read and can be broken down quickly, it will also be easy to see where problems may arise. Having good Code quality and error handling is important because it prevents problems later on and provides a strong foundation to build off of. I included error handling as well to improve the security and strength of the system while refactoring. Using accepted and standard algorithms is always a good way to encrypt data and by making the output string values final, it prevents other objects from changing them and secures the integrity of the return values.

Making this values final helps protect the programmer from preventing their values from being changed, but they could still be improved by preventing their values from being known by further encapsulating them in another object and adding another layer of protection. This is just one way a developer can improve the security of their system by adding layers. While it may impact speed, security is important when dealing with sensitive information because once its leaked its leaked forever. Making sure this kind of data is secure will be paramount to Artemis financial and they will need to meet federal and international standards if they plan on operating on a wider scale.

* **Industry Standard Best Practices**

By protecting data, using specific variable and function identifiers and running the refactored code through the OWASP dependency checker, the software is being maintained to a secure level. Data was protected by implementing java final identifier to variables to protect their data from being changed or modified. Private identifiers were added to sensitive functions so other objects could not interact or get data from them. The correct data was sent and transmitted between the functions and error handling was implemented through try, catch blocks when necessary. Overall all security standards were attempted to be met and any additional vulnerabilities presented by the dependency check were isolated and fixed. Further security enhancements were added near the end of development to further strenghten the security of the system.

Its important to maintain and improve the security of systems that need to send, receive, and augment sensitive data. These data practices are often managed or oversaw by government bodies because of the importance of data being transmitted. Adhering to these guidelines allows developers to feel at ease knowing the data they are using is being protected so long as they implement it in the correct way. By applying industry best practices and striving to exceed security standards, software can be made more secure and more trusted by people who need it most around the world.