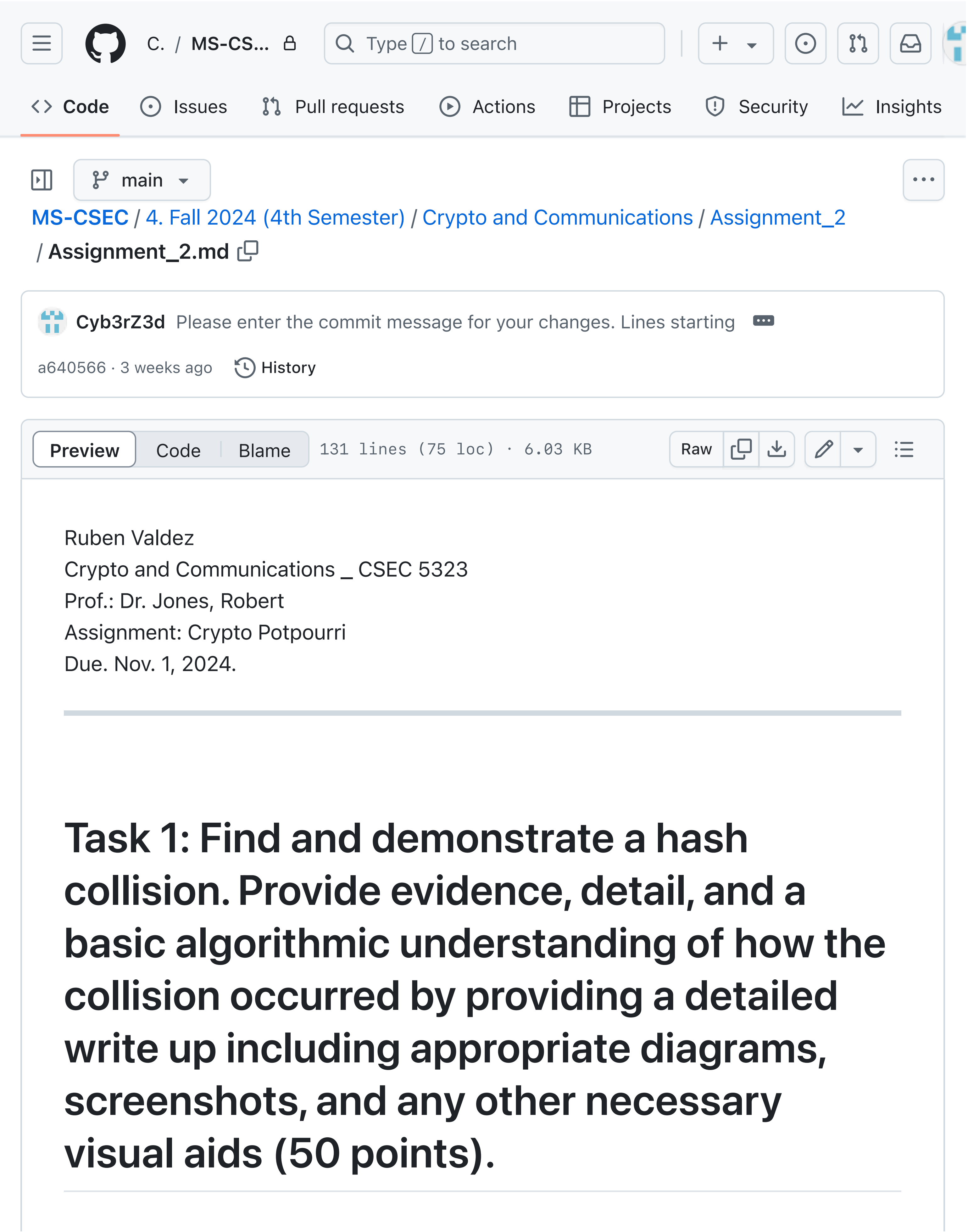

C. / MS-CS...
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security
Insights
main
MS-CSEC / 4. Fall 2024 (4th Semester) / Crypto and Communications / Assignment_2
/ Assignment_2.md
Cyb3rZ3d Please enter the commit message for your changes. Lines starting
a640566 · 3 weeks ago
History
Preview
Code
Blame
131 lines (75 loc) · 6.03 KB
Raw

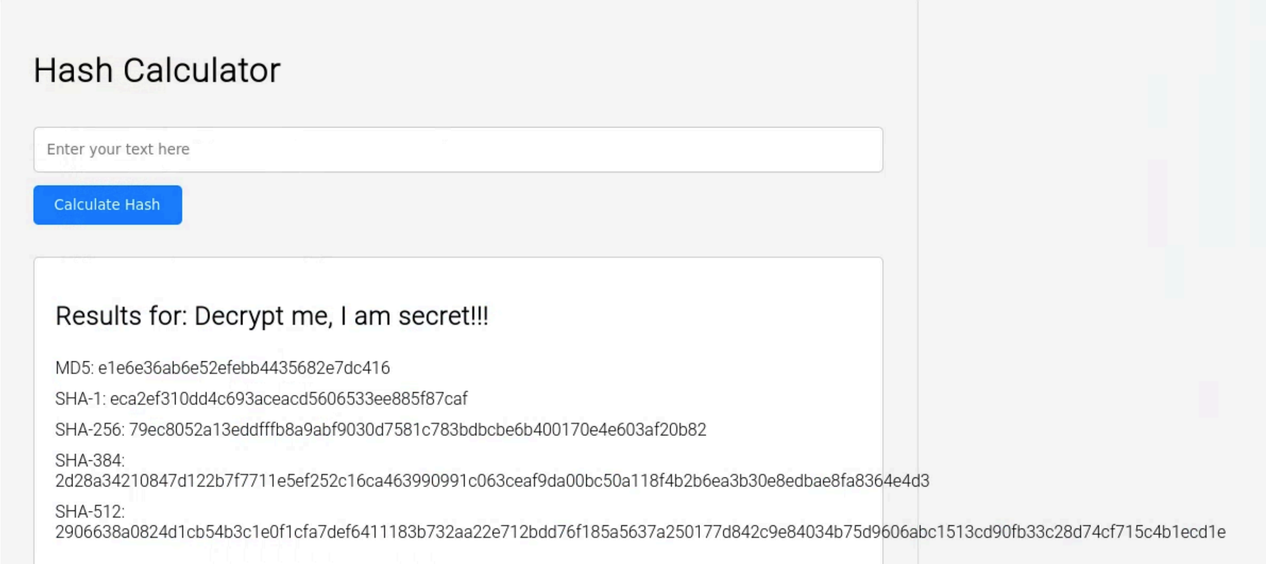
Ruben Valdez
Crypto and Communications _ CSEC 5323
Prof.: Dr. Jones, Robert
Assignment: Crypto Potpourri
Due. Nov. 1, 2024.

Task 1: Find and demonstrate a hash collision. Provide evidence, detail, and a basic algorithmic understanding of how the collision occurred by providing a detailed write up including appropriate diagrams, screenshots, and any other necessary visual aids (50 points).

- I attempted to use your hash.py program. wasn't sure exactly what else to do after i manually attempted to calculate the text i entered `Decrypt me, I am secret!!!`.

  http://cl.xytify.net/cgi-bin/hash.py

  Results for: Decrypt me, I am secret!!! MD5: e1e6e36ab6e52efebb4435682e7dc416 SHA-1: eca2ef310dd4c693aceacd5606533ee885f87caf SHA-256: 79ec8052a13eddfffb8a9abf9030d7581c783bdbcbe6b400170e4e603af20b82 SHA-384: 2d28a34210847d122b7f7711e5ef252c16ca463990991c063ceaf9da00bc50a118f4b2b6ea3b30e8edbae8fa8364e4d3 SHA-512: 2906638a0824d1cb54b3c1e0f1cfa7def6411183b732aa22e712bdd76f185a5637a250177d842c9e84034b75d9606abc1513cd90fb33c28d74cf715c4b1ecd1e

  Hash Calculator

  ```
  Results for: Decrypt me, I am secret!!!

  MD5: e1e6e36ab6e52efebb4435682e7dc416
  SHA-1: eca2ef310dd4c693aceacd5606533ee885f87caf
  SHA-256: 79ec8052a13eddfffb8a9abf9030d7581c783bdbcbe6b400170e4e603af20b82
  SHA-384:
  2d28a34210847d122b7f7711e5ef252c16ca463990991c063ceaf9da00bc50a118f4b2b6ea3b30e8edbae8fa8364e4d3
  SHA-512:
  2906638a0824d1cb54b3c1e0f1cfa7def6411183b732aa22e712bdd76f185a5637a250177d842c9e84034b75d9606abc1513cd90fb33c28d74cf715c4b1ecd1e
  ```

- Since I am visual I wanted to create my own python script using the same text `Decrypt me, I am secret!!!` and then modifying the text by adding a space at the end `Decrypt me, I am secret!!! `.

  i. Imports:
     - Used Python's built-in hashlib library, which provides hash functions, including MD5.

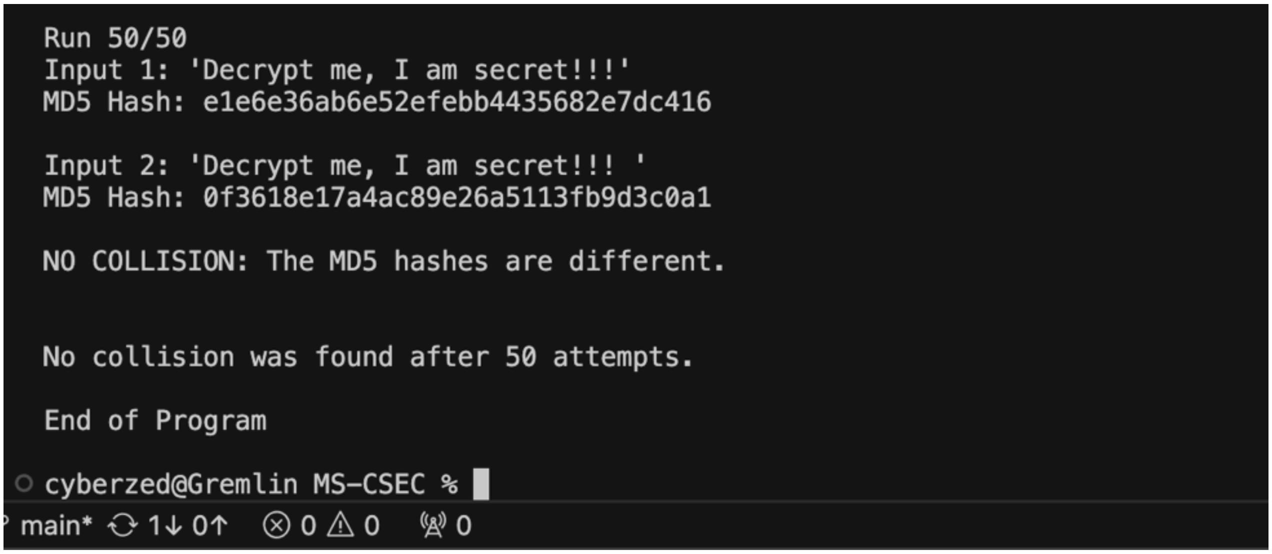
  ii. Function: test_for_collision():
     - This function defines two inputs:
       - input1 is the original string: "Decrypt me, I am secret!!!".
       - input2 is a slightly modified version with an extra space at the end: "Decrypt me, I am secret!!! ".

        - Both inputs are encoded to bytes, as required by the hashlib.md5() function.
        - The function calculates the MD5 hash of each input using hashlib.md5(input).hexdigest(), which returns the hash in hexadecimal format.
        - It prints each input string and its respective MD5 hash.
        - The function then checks if the two MD5 hashes are identical:
            - If they match, it indicates an MD5 collision, prints a message confirming this, and returns True.
            - If they do not match, it prints a message stating no collision was found and returns False.

    iii. Function: main(): This function orchestrates the testing process, running test_for_collision() up to 50 times. It keeps track of whether a collision has been found, initializing collision_found to False. For each attempt, it prints the current attempt number, calls test_for_collision(), and checks if it returned True. If a collision is detected (True), the loop breaks immediately, indicating success. If the loop completes all 50 attempts without finding a collision, it prints a summary message indicating no collision was detected after 50 attempts.

    iv. Execution: The script includes a conditional if **name** == "**main**": block, which ensures the main() function only runs if the script is executed directly.

  Result:



```
Run 50/50
Input 1: 'Decrypt me, I am secret!!!'
MD5 Hash: e1e6e36ab6e52efebb4435682e7dc416

Input 2: 'Decrypt me, I am secret!!! '
MD5 Hash: 0f3618e17a4ac89e26a5113fb9d3c0a1

NO COLLISION: The MD5 hashes are different.


No collision was found after 50 attempts.

End of Program

cyberzed@Gremlin MS-CSEC %
```

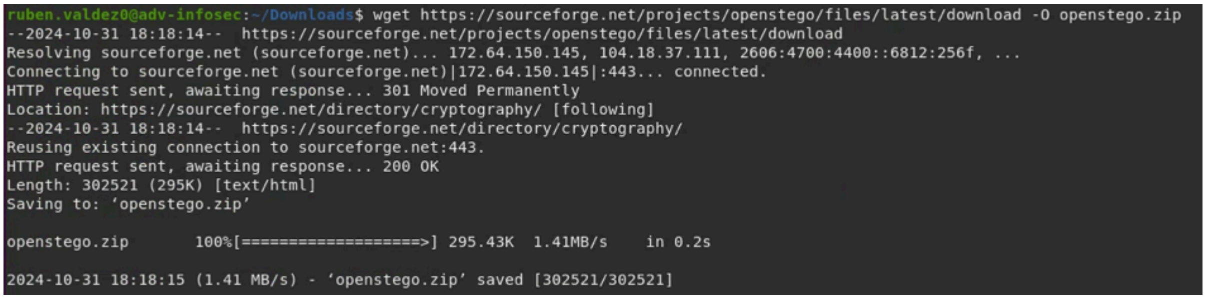
- Summary,

Wasn't sure exactly how to perform this task. I just remember doing a similary project in python in Python Security Programming with Prof. Logher. I know i deviated from using your python program but just decided to create a quick script to run up to 50 attempts if a collision had not been found before stopping the program.

Task 2: Generate your own nested steganographic solution. It must include some secondary authentication mechanic. Provide the base message, the cover file, and methodology as a detailed write up (50 points).

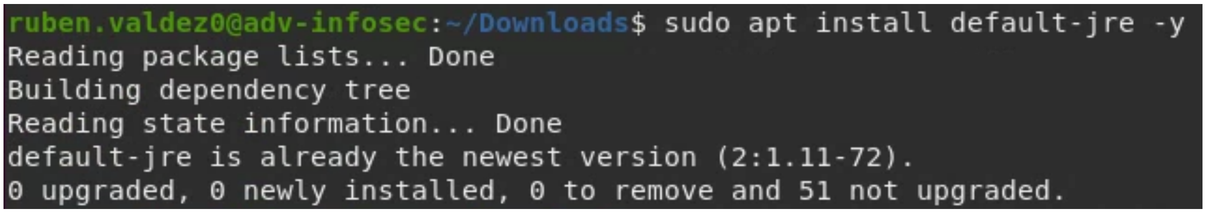
1. Installed `OpenStego_0.8.60-1_all.deb` on my Ubuntu GCP instance.

    - Installed OpenStego:

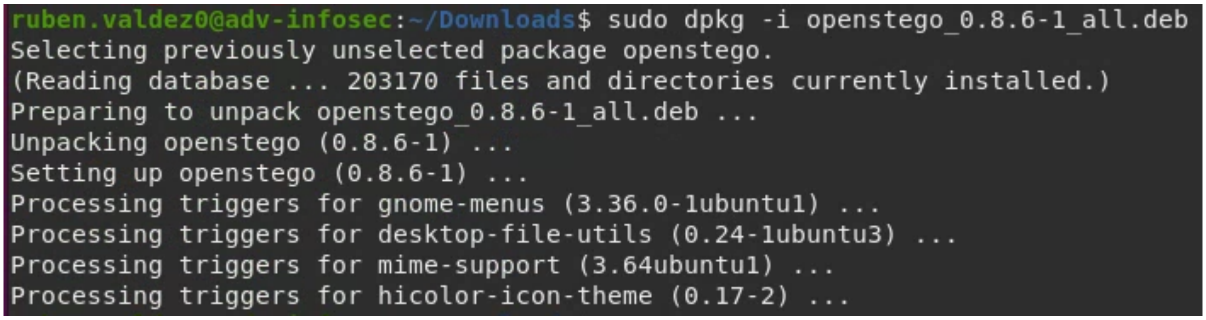
      ```
      ruben.valdez0@adv-infosec:~/Downloads$ wget https://sourceforge.net/projects/openstego/files/latest/download -O openstego.zip
      --2024-10-31 18:18:14--  https://sourceforge.net/projects/openstego/files/latest/download
      Resolving sourceforge.net (sourceforge.net)... 172.64.150.145, 104.18.37.111, 2606:4700:4400::6812:256f, ...
      Connecting to sourceforge.net (sourceforge.net)|172.64.150.145|:443... connected.
      HTTP request sent, awaiting response... 301 Moved Permanently
      Location: https://sourceforge.net/directory/cryptography/ [following]
      --2024-10-31 18:18:14--  https://sourceforge.net/directory/cryptography/
      Reusing existing connection to sourceforge.net:443.
      HTTP request sent, awaiting response... 200 OK
      Length: 302521 (295K) [text/html]
      Saving to: ‘openstego.zip’

      openstego.zip       100%[===================>] 295.43K  1.41MB/s    in 0.2s

      2024-10-31 18:18:15 (1.41 MB/s) - ‘openstego.zip’ saved [302521/302521]
      ```

    - Installed Java:

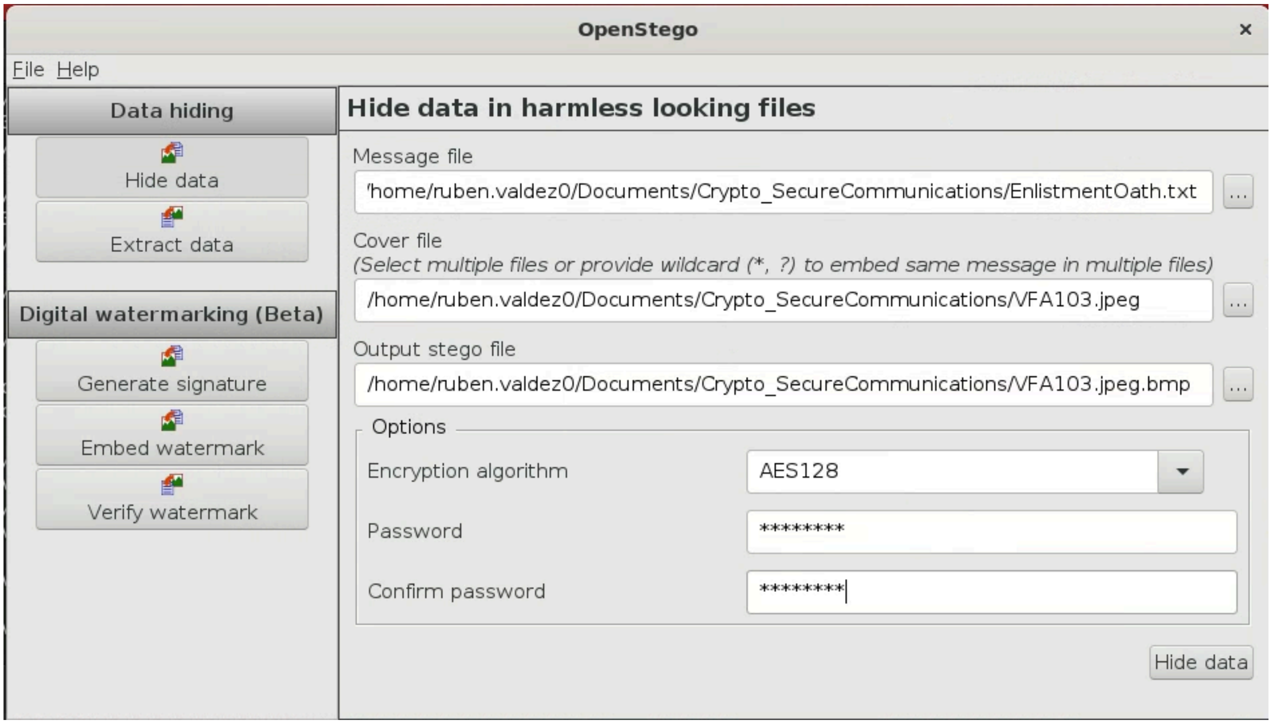
      ```
      ruben.valdez0@adv-infosec:~/Downloads$ sudo apt install default-jre -y
      Reading package lists... Done
      Building dependency tree
      Reading state information... Done
      default-jre is already the newest version (2:1.11-72).
      0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.
      ```

    - Installed the OpenStego .deb package using dpkg



```
ruben.valdez0@adv-infosec:~/Downloads$ sudo dpkg -i openstego_0.8.6-1_all.deb
Selecting previously unselected package openstego.
(Reading database ... 203170 files and directories currently installed.)
Preparing to unpack openstego_0.8.6-1_all.deb ...
Unpacking openstego (0.8.6-1) ...
Setting up openstego (0.8.6-1) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
```

2. Text File prep and locating the cover image.
   - Created a text file namedd `EnlistedOath.txt`.
   - Copy/Pasted in the text file the Enlisted Oath, The Sailor's Creed, and the Navy Song - Anchors Aweigh.
     - Source: https://www.navy.mil/About/Our-Heritage/
3. We can start the process to embed/hide the file using OpenStego



Hide data in harmless looking files

Message file: 'home/ruben.valdez0/Documents/Crypto_SecureCommunications/EnlistmentOath.txt

Cover file
(Select multiple files or provide wildcard (, ?) to embed same message in multiple files)*
/home/ruben.valdez0/Documents/Crypto_SecureCommunications/VFA103.jpeg

Output stego file: /home/ruben.valdez0/Documents/Crypto_SecureCommunications/VFA103.jpeg.bmp

Options

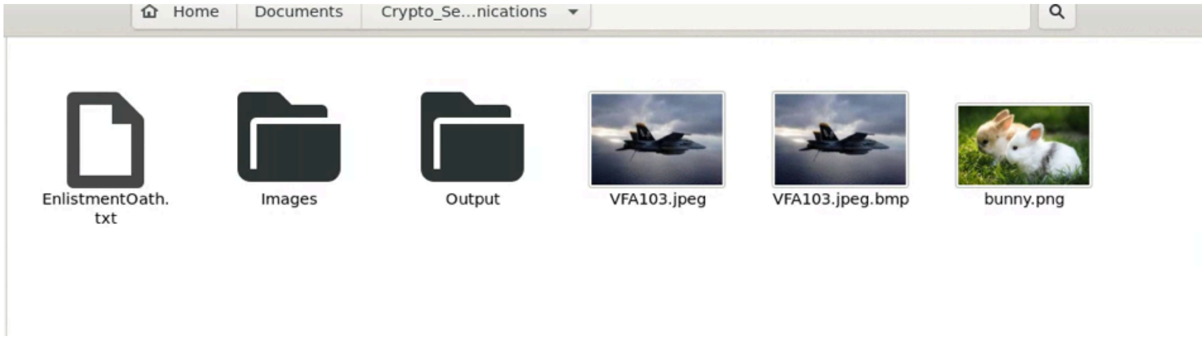
Encryption algorithm: AES128

Password: *********

Confirm password: *********

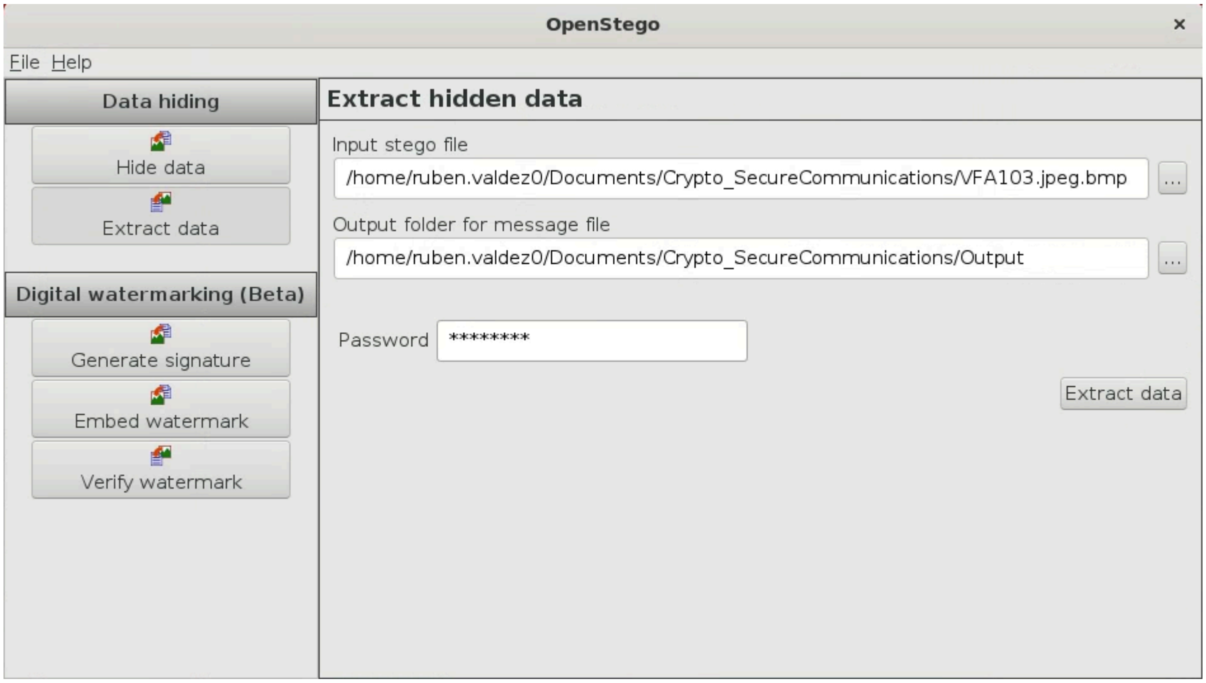
   - Message File: I uploaded the text file I created.
   - Cover File: I used a downloaded image of a VFA-103 SuperHornet fighter jet I used to maintain back from my Navy days.

- Output Stego File: Provided the file path to save my output file. This is the updated file with the cover image listed with the text file hidden in the image.



4. Extracting the Hidden File

- enter the stego file:

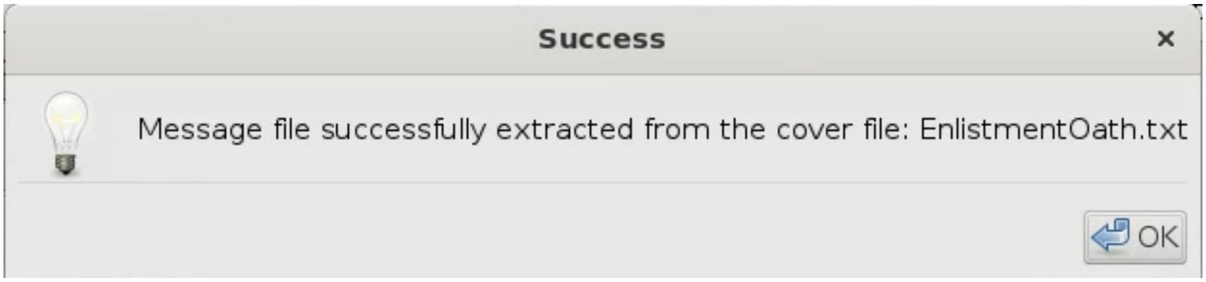


Input stego file: /home/ruben.valdez0/Documents/Crypto_SecureCommunications/VFA103.jpeg.bmp

Output folder for message file: /home/ruben.valdez0/Documents/Crypto_SecureCommunications/Output

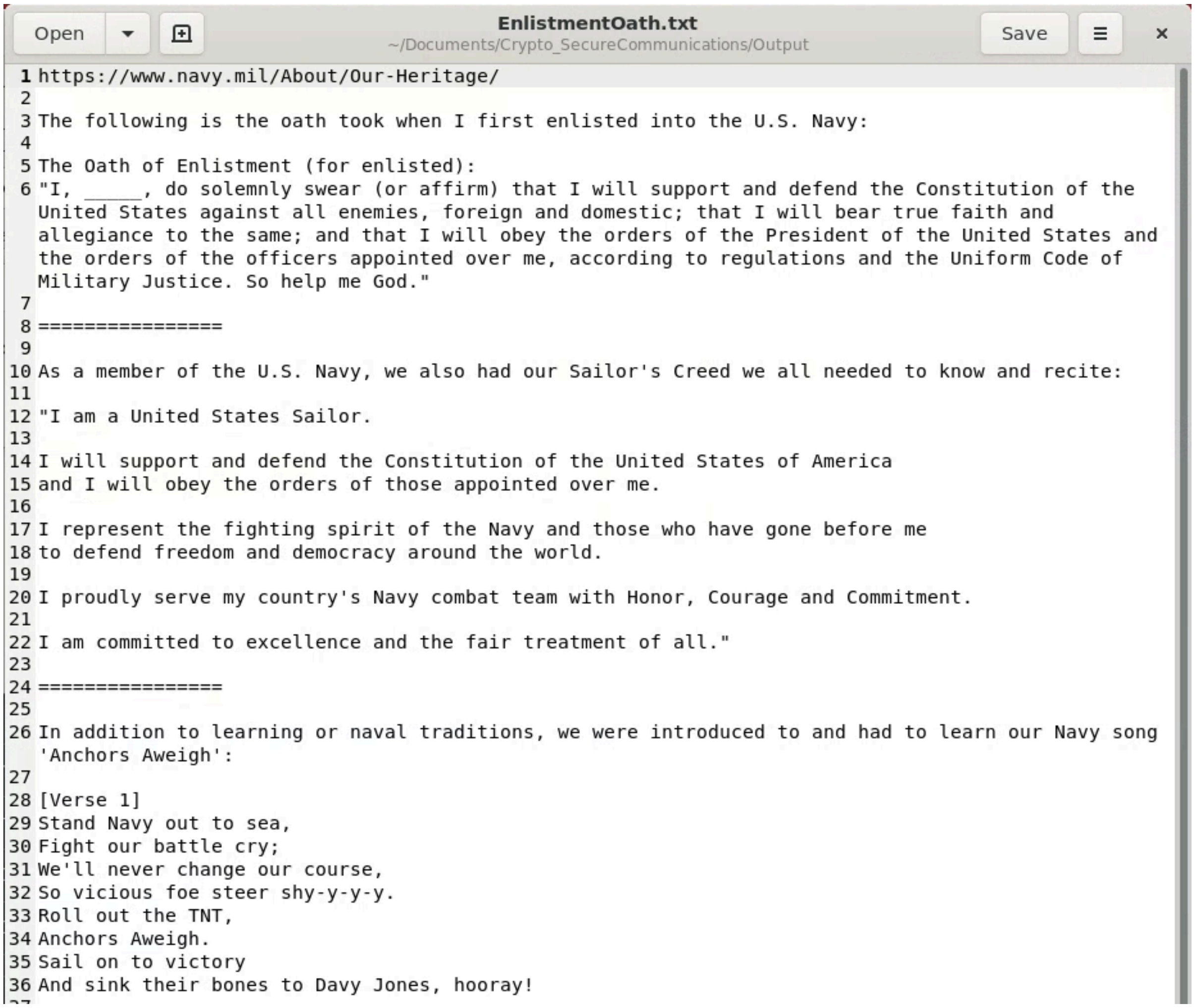
Password: ********

- Select the output folder:



Message file successfully extracted from the cover file: EnlistmentOath.txt

- Enter the password: ^YHN6yhn



```
https://www.navy.mil/About/Our-Heritage/

The following is the oath took when I first enlisted into the U.S. Navy:

The Oath of Enlistment (for enlisted):
"I, _____, do solemnly swear (or affirm) that I will support and defend the Constitution of the
United States against all enemies, foreign and domestic; that I will bear true faith and
allegiance to the same; and that I will obey the orders of the President of the United States and
the orders of the officers appointed over me, according to regulations and the Uniform Code of
Military Justice. So help me God."

================

As a member of the U.S. Navy, we also had our Sailor's Creed we all needed to know and recite:

"I am a United States Sailor.

I will support and defend the Constitution of the United States of America
and I will obey the orders of those appointed over me.

I represent the fighting spirit of the Navy and those who have gone before me
to defend freedom and democracy around the world.

I proudly serve my country's Navy combat team with Honor, Courage and Commitment.

I am committed to excellence and the fair treatment of all."

================

In addition to learning or naval traditions, we were introduced to and had to learn our Navy song
'Anchors Aweigh':

[Verse 1]
Stand Navy out to sea,
Fight our battle cry;
We'll never change our course,
So vicious foe steer shy-y-y-y.
Roll out the TNT,
Anchors Aweigh.
Sail on to victory
And sink their bones to Davy Jones, hooray!
```

Summary:

```
This was a nice exercise.  Although, as I was performing this
task, I wasn't sure what you meant by `It must include some
secondary authentication mechanic.`.  I didn't see there being an
option to add  secondary authentication.
```

Task 3: Sign your name on the list at dfw.xytify.net by hashing the value at dfw.xytify.net/hash.txt, ssh'ing in to the machine, and editing the file named editme. The password is also the username. If you need a hint, the combo is used in the Oath of Office taken by United States government officials (50 points).

So in this task i was completely lost in how to navigate resolving this. I was unable to resolve this task.

I wasn't sure if I had to break the hash using hashcat or a different program. In total i just couldn't figure out what i had to do.