Encrypted File Extraction from Image

Scenario: During a forensic analysis of files extracted from a suspect's computer, your team has isolated a folder of images that have been flagged as suspicious. You have been tasked to analyze the images for the potential threat of steganography.

Files Included:

1. books.jpg
2. family.jpg
3. hellokitty.jpg
4. IMG_9008.jpg
5. AES_Encrypt.py
6. AES_Decrypt.py
7. fasttrack.txt

Installing Necessary Packages:

1. Pycryptodome
    a. pip install pycryptodome –break-system-packages
2. Stegcracker
    a. apt install stegcracker

Necessary Commands:

- To become superuser
    - sudo su
- When running Python in the terminal
    - python3 [python file]
- When using Stegcrack
    - stegcrack [image] [wordlist]

Understanding Wordlists

fasttrack.txt is a wordlist of possible passwords that will be used for finding the password for the steganographic image. If necessary, forensic analysts would generate a wordlist centered around the known facts about the suspect and add those words to the pre-existing wordlist.

To complete this challenge, you will need to know a little about how Python works and have to change variables. Below will be the code from the two Python files you will need for this challenge. Certain sections will be highlighted and explained in further detail so you understand what is happening and what you must do to run the program successfully.

AES_Encrypt.py

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes

with open ("text.txt", "rb") as data:
    plaintext = data.read()

secret_key = get_random_bytes(16)

file_key = open("AES_key.txt", "wb")
file_key.write(secret_key)
file_key.close()

cipher_AES = AES.new(secret_key, AES.MODE_CFB)
cipher_text = cipher_AES.encrypt(plaintext)

file_out = open("AES_Encrypted.bin", "wb")
file_out.write(cipher_AES.iv)
file_out.write(cipher_text)
file_out.close()
```

1. `with open ("text.txt", "rb") as data:`
    a. This line will open the file that needs to be encrypted and read in the data.
    b. "text.txt" is the name of the file that you want to encrypt.
2. `file_key = open("AES_key.txt", "wb")`
    a. This line creates a file that holds the AES key that is generated by the program.
    b. "AES_key.txt" is the name of the file that holds the key.
3. `file_out = open("AES_Encrypted.bin", "wb")`
    a. This line creates a file that holds the encrypted data, being the encrypted version of whatever plaintext you wanted to encrypt.

b. "AES_Encrypted.bin" is the name of the file that holds the encrypted data. This file should always be saved as .bin for binary.

AES_Decrypt.py

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes

file_in =  open("encrypted_file", "rb")

iv = file_in.read(16)

cipher_text = file_in.read()
file_in.close()

file_key = open("file", "rb")

## read also the secret key
secret_key = file_key.read()
file_key.close()
## read also the tag

cipher_AES = AES.new(secret_key, AES.MODE_CFB, iv=iv)
decrypted_message = cipher_AES.decrypt(cipher_text)

retrieval = open("decrypted_message.txt", "wb")
retrieval.write(decrypted_message)
retrieval.close()

print(decrypted_message)
```

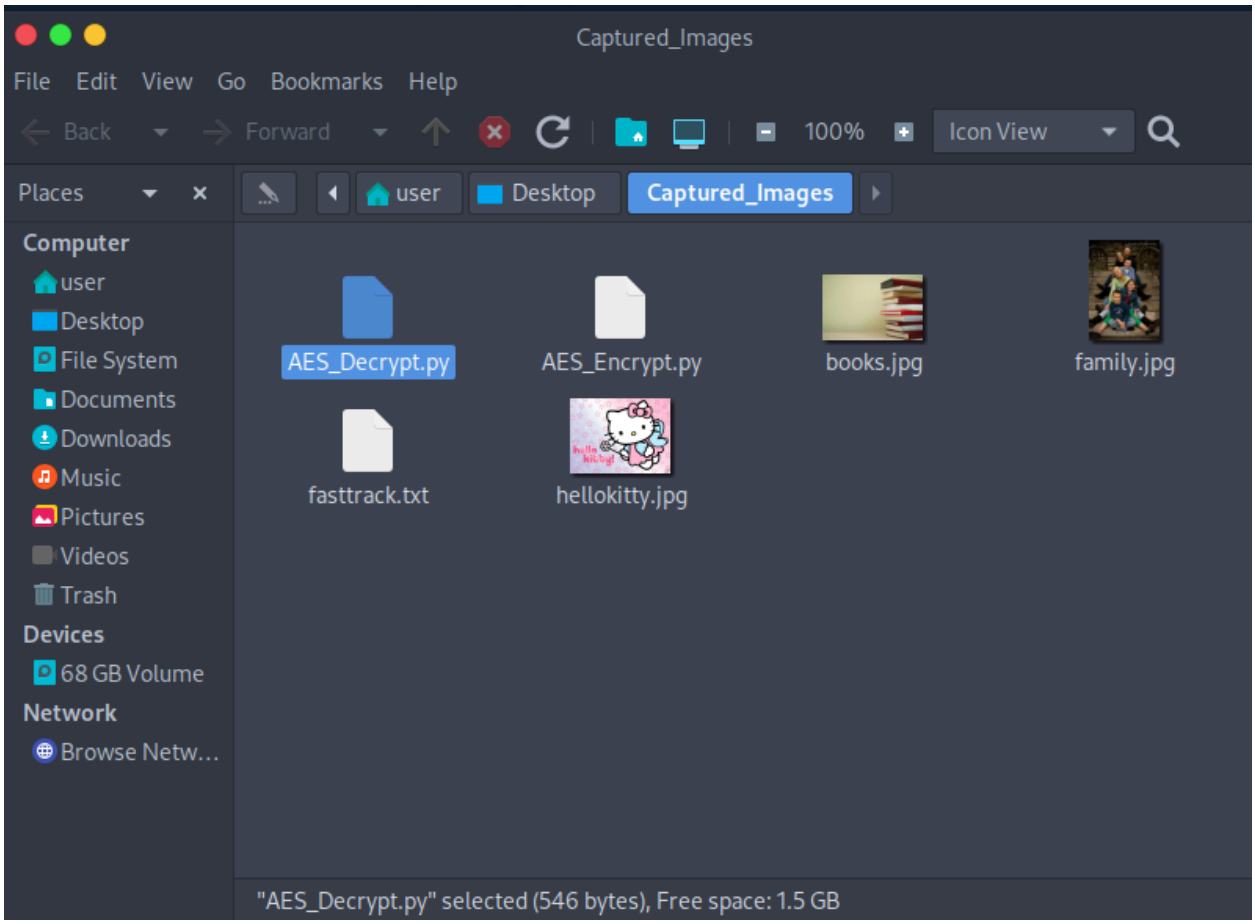
1. `file_in =  open("encrypted_file", "rb")`
   a. This line reads in the encrypted file in order to be processed by the program.
   b. "encrypted_file" will be the name of the file you wish to decrypt.
2. `file_key = open("file", "rb")`
   a. This line reads in the key file in order to be processed by the program.
   b. "file" is the name of the key file that you want to use to decrypt your encrypted file.

3. It is important to note that the encrypted file and key file may have unconventional names, however, they still will hold the same data.
4. This script is for files encrypted with CFB.

Instructions

These are the step-by-step instructions for how to complete this challenge in case you get lost. See how far you can get without needing to look here.

1. cd into the Captured_Images folder
2. Become the root user by typing sudo su
3. Install the necessary packages for this challenge
    a. pip install pycryptodome –break-system-packages
    b. apt install stegcracker
4. Open AES_Decrypt.py from the folder view



5. Minimize the text screen to be used later

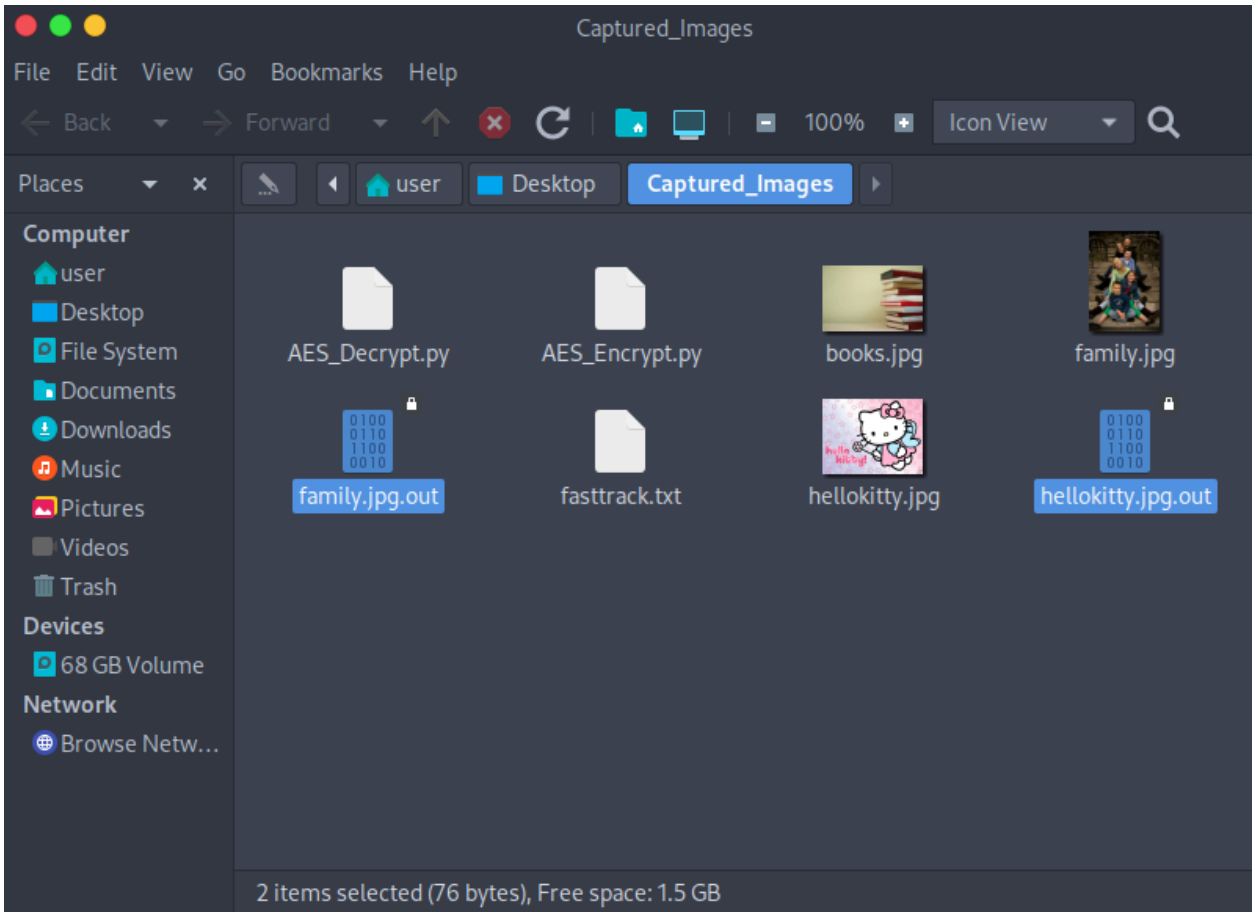
6. Run the command stegcracker hellokitty.jpg fasttrack.txt

```
Parrot Terminal
File  Edit  View  Search  Terminal  Help
┌─[root@parrot]─[/home/user/Desktop/Captured_Images]
└──╼ #stegcracker hellokitty.jpg fasttrack.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2024 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file 'hellokitty.jpg' with wordlist 'fasttrack.txt'..
Successfully cracked file with password: dragon
Tried 214 passwords
Your file has been written to: hellokitty.jpg.out
dragon
┌─[root@parrot]─[/home/user/Desktop/Captured_Images]
└──╼ #
```

7. Run the command stegcracker family.jpg fasttrack.txt

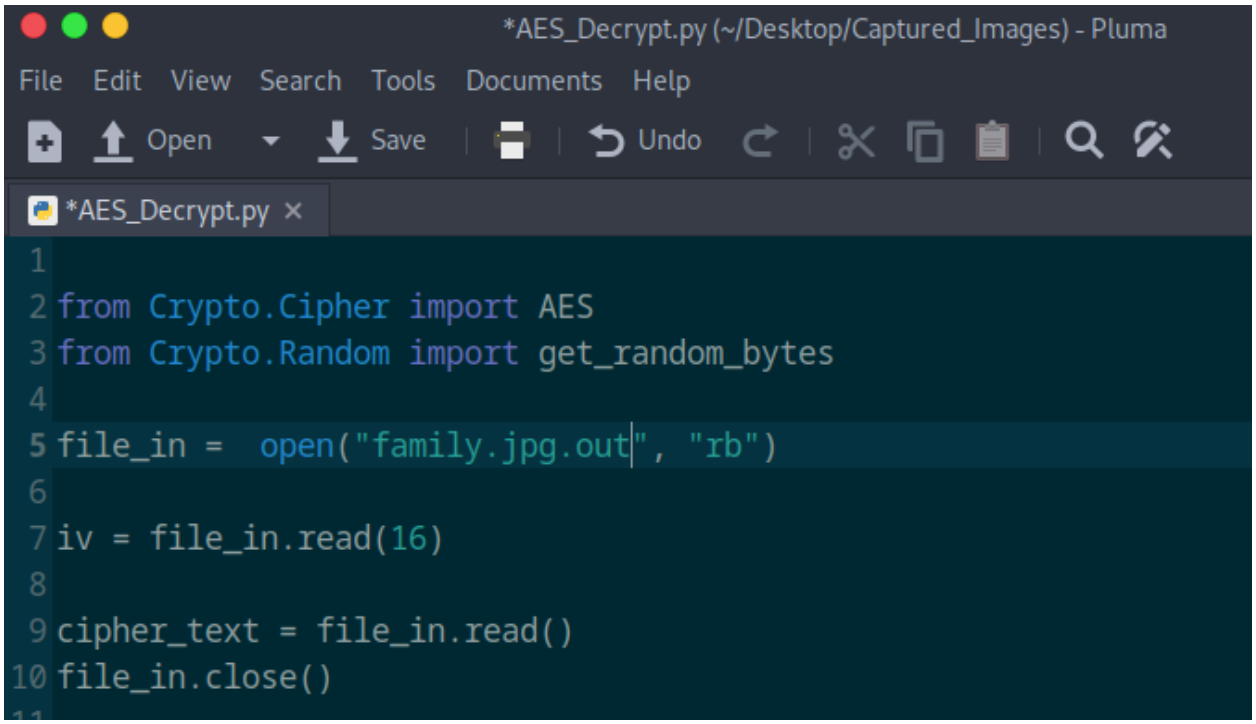
```
Parrot Terminal
File  Edit  View  Search  Terminal  Help
┌─[root@parrot]─[/home/user/Desktop/Captured_Images]
└──╼ #stegcracker family.jpg fasttrack.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2024 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file 'family.jpg' with wordlist 'fasttrack.txt'..
Successfully cracked file with password: princess
Tried 219 passwords
Your file has been written to: family.jpg.out
princess
┌─[root@parrot]─[/home/user/Desktop/Captured_Images]
└──╼ #
```

8. Open the folder view to see the generated files

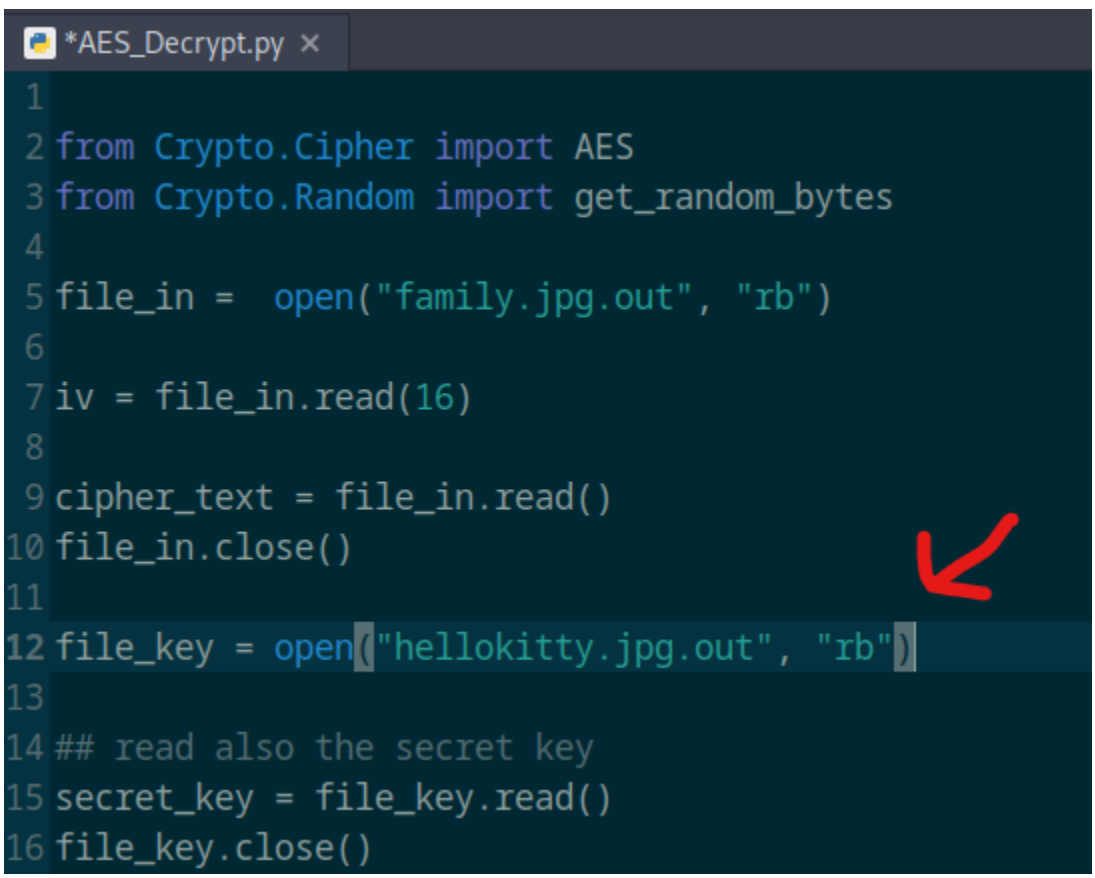


9. One of these two files is the encrypted message and one of these two files is the AES key.
10. Open the AES_Decrypt.py text that you minimized earlier

11. Change the file_in variable to specify the encrypted file



12. Change the file_key variable to specify the AES key



13. Save the file by clicking CTRL + S or the Save button at the top
14. Return to the terminal and prepare to decrypt the message
15. Run the command python3 AES_Decrypt.py

```
┌─[root@parrot]─[/home/user/Desktop/Captured_Images]
└──╼ #python3 AES_Decrypt.py
b'Meet at the bus stop on 76th ave for pickup\n'
┌─[root@parrot]─[/home/user/Desktop/Captured_Images]
└──╼ #
```

16. You have completed the challenge successfully.