

A screenshot of a computer

Description automatically generated

A screenshot of a computer message

Description automatically generated

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generated

A screen shot of a computer

Description automatically generated

A white background with black text

Description automatically generated

A screen shot of a computer code

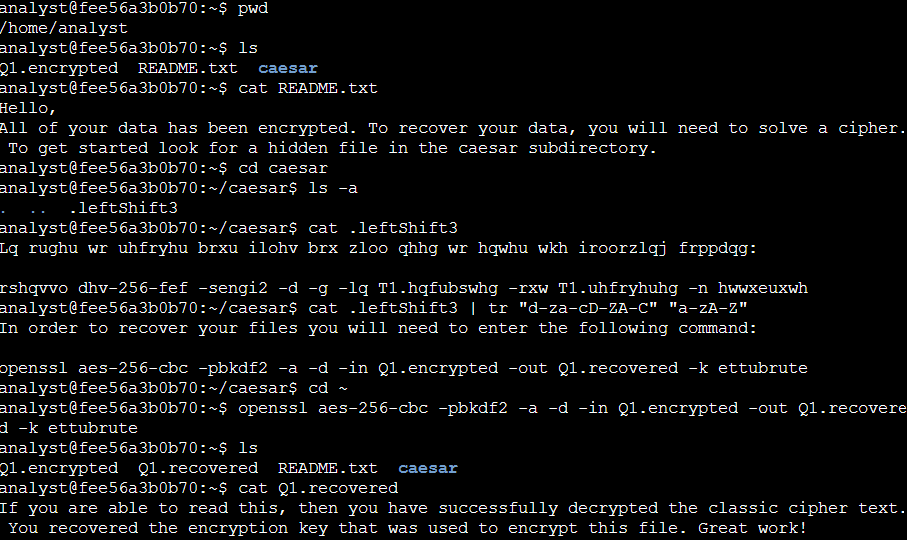
Description automatically generated

A screen shot of a computer

Description automatically generatedA close up of a text

Description automatically generated

* Here, the openssl command reverses the encryption of the file with a secure symmetric cipher, as indicated by AES-256-CBC.
* The -pbkdf2 option is used to add extra security to the key, and -a indicates the desired encoding for the output.
* The -d indicates decrypting, while -in specifies the input file and -out specifies the output file. The -k specifies the password, which in this example is ettubrute.

**Activity completed Using Linux bash shell**

**Explanation**

First, I checked what files were in my home directory by using ls /home/analyst. This showed files like README.txt and Q1.encrypted, and a directory named caesar.

then I opened README.txt with the command cat README.txt to look for any initial instructions or information.

Next, I changed directories to the caesar folder using cd caesar because the README suggested that more information could be found there. Inside the caesar directory, I listed all files, including hidden ones, with ls -a. This revealed a hidden file called .leftShift3.

then I used cat .leftShift3 to display the contents of the hidden file, which appeared to be encrypted with a Caesar cipher. To decode it, I used the tr command. tr "d-za-cD-ZA-C" "a-zA-Z" shifts each letter by three places in the alphabet, effectively reversing the Caesar cipher encryption. tr stands for translate. It replaces or removes specific characters in its input data set, which is useful for tasks like decoding.

The decrypted instructions told me to use a specific OpenSSL command to decrypt the file Q1.encrypted.

Later I moved to my home directory using cd ~, I ran the OpenSSL command provided by the decoded message: openssl aes-256-cbc -pbkdf2 -a -d -in Q1.encrypted -out Q1.recovered -k ettubrute. This command uses OpenSSL, a toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. aes-256-CBC specifies the encryption method, -d stands for decrypt, -in specifies the input file, -out specifies the output file, and -k provides the encryption key.

I listed the files again with ls to see the newly created Q1.recovered. I opened it using cat Q1.recovered to verify that the decryption was successful and to read the hidden message.