SIT282 Computer Crime and Digital Forensics
ASSIGNMENT 2

Report Title: Forensic Analysis of Suspected Drug Manufacturing Operation

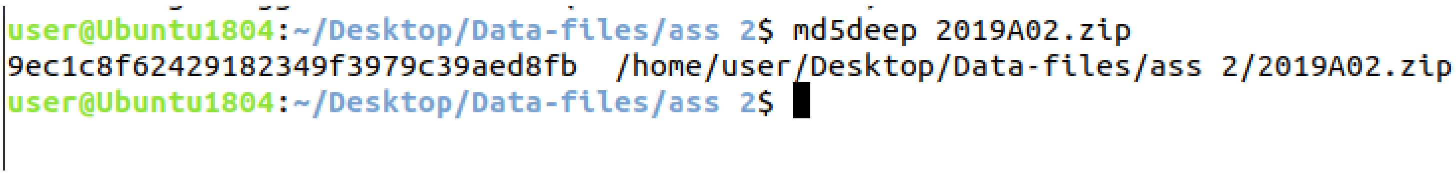
Table of Contents

Investigator Name: ***Nirosh Ravindran***

DIGITAL FORENSIC PROCEDURE

1. Explain how you downloaded the file, what precautions you took, and how you ensured its integrity.

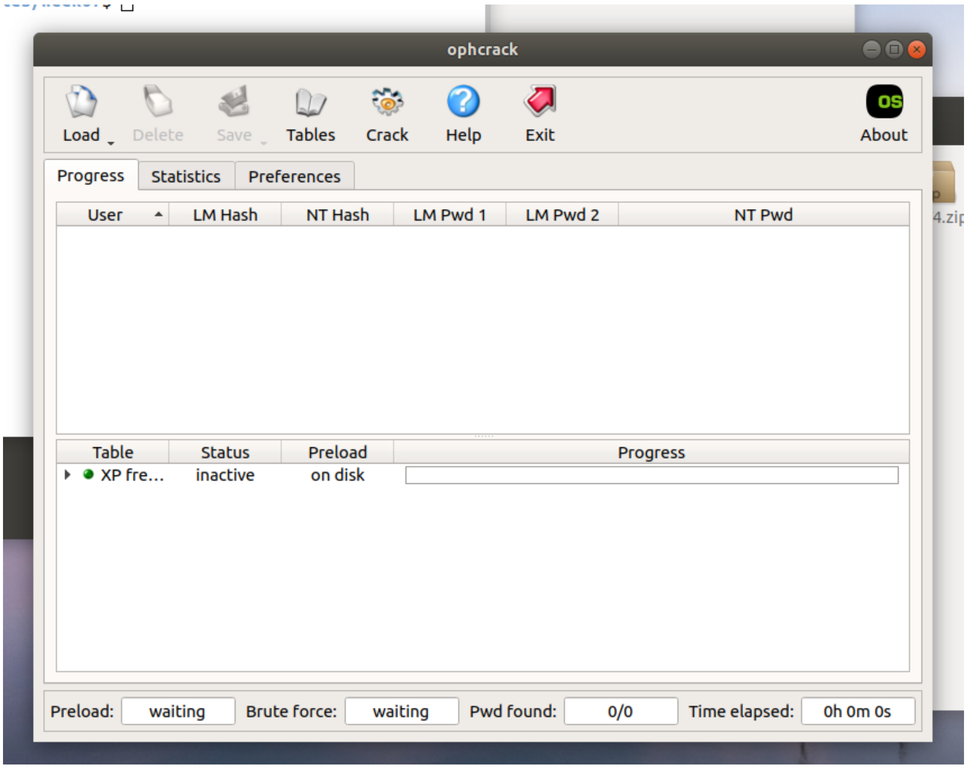
File Download Procedure	*I downloaded the zip file from the link* [*http://www.deakin.edu.au/~zoidberg/2019A02.zip*](http://www.deakin.edu.au/~zoidberg/2019A02.zip)
Precautions Applied	*I made sure my antivirus software was up-to-date and was active during the download process.* *I ensured the download took place over a secure network connection.* *Used a dedicated, isolated lab environment to prevent potential contamination and any security risks to my main system.*
Method used to ensure Integrity	*I verified the md5 hash value of the executable file within the ZIP archive to be* 9ec1c8f62429182349f3979c39aed8fb, ensuring the file was not tampered with during the download.



```
user@Ubuntu1804:~/Desktop/Data-files/ass 2$ md5deep 2019A02.zip
9ec1c8f62429182349f3979c39aed8fb  /home/user/Desktop/Data-files/ass 2/2019A02.zip
user@Ubuntu1804:~/Desktop/Data-files/ass 2$ 
```

2. Describe how you decrypt two given NTLM hash values by using OphCrack, including screen shots.

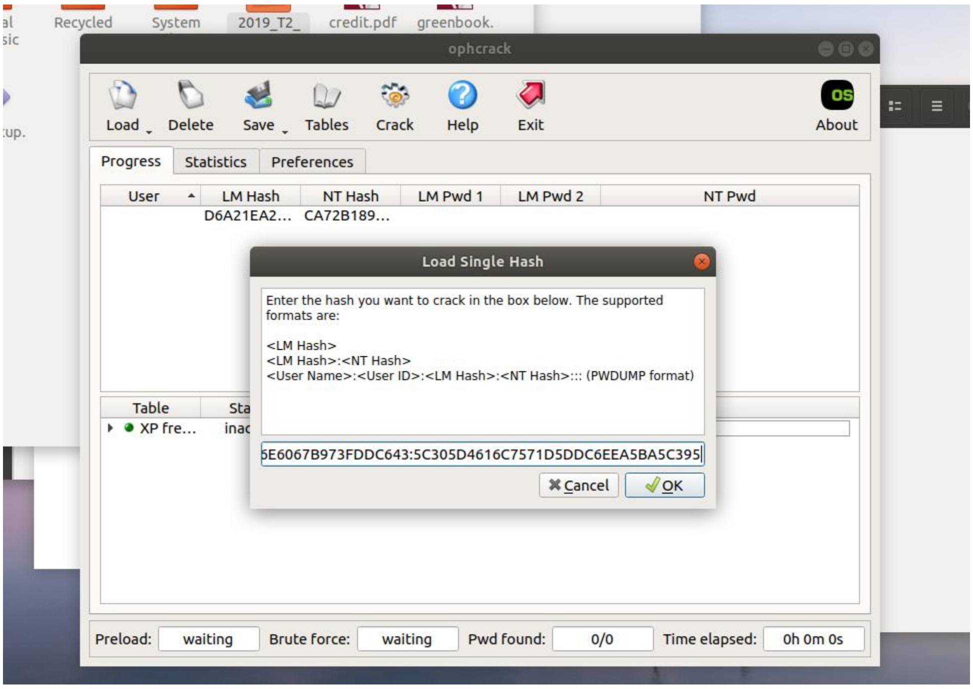
- *First, I loaded the NTLM hashed into the OphCrack tool.*
- *Then, I used the rainbow table in that came default with the OphCrack tool.*
- *Finally, I initiated the decryption process and waited for the hashes to be cracked.*
- *the first hash value D6A21EA26063C42FC9876E4B0C51BC82:CA72B189F412A384D96B785A08176773 was decrypted into* ***DickCheney***
- *the second hash value* 8282461A2BDAF626E6067B973FDDC643:5C305D4616C7571D5DDC6EEA5BA5C395 was decrypted into **DonaldRumsfeld**



User	LM Hash	NT Hash	LM Pwd 1	LM Pwd 2	NT Pwd

Table	Status	Preload	Progress
XP fre…	inactive	on disk	

Preload: waiting Brute force: waiting Pwd found: 0/0 Time elapsed: 0h 0m 0s



User	LM Hash	NT Hash	LM Pwd 1	LM Pwd 2	NT Pwd
	D6A21EA2…	CA72B189…			

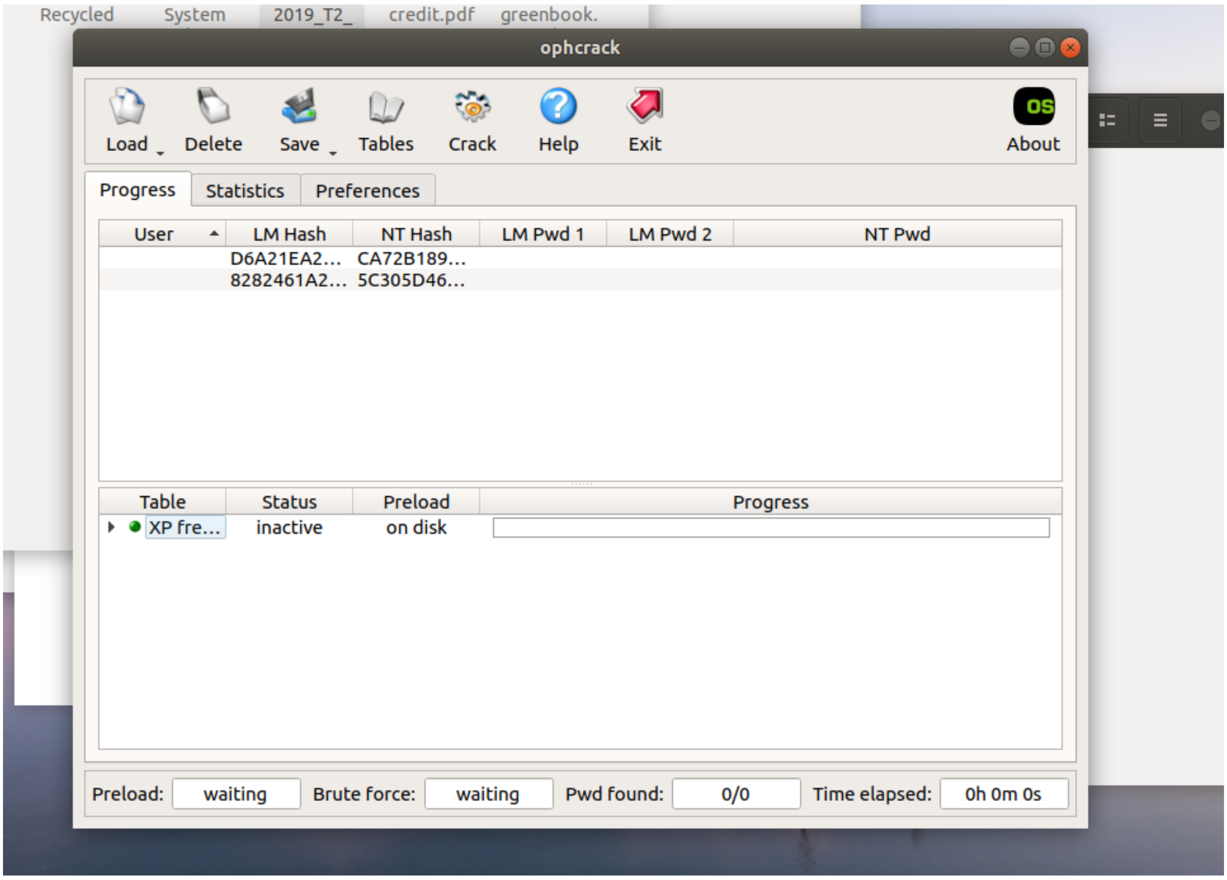
Table	Sta
XP fre…	inac

Enter the hash you want to crack in the box below. The supported formats are:

<LM Hash>
<LM Hash>:<NT Hash>
<User Name>:<User ID>:<LM Hash>:<NT Hash>::: (PWDUMP format)

5E6067B973FDDC643:5C305D4616C7571D5DDC6EEA5BA5C395

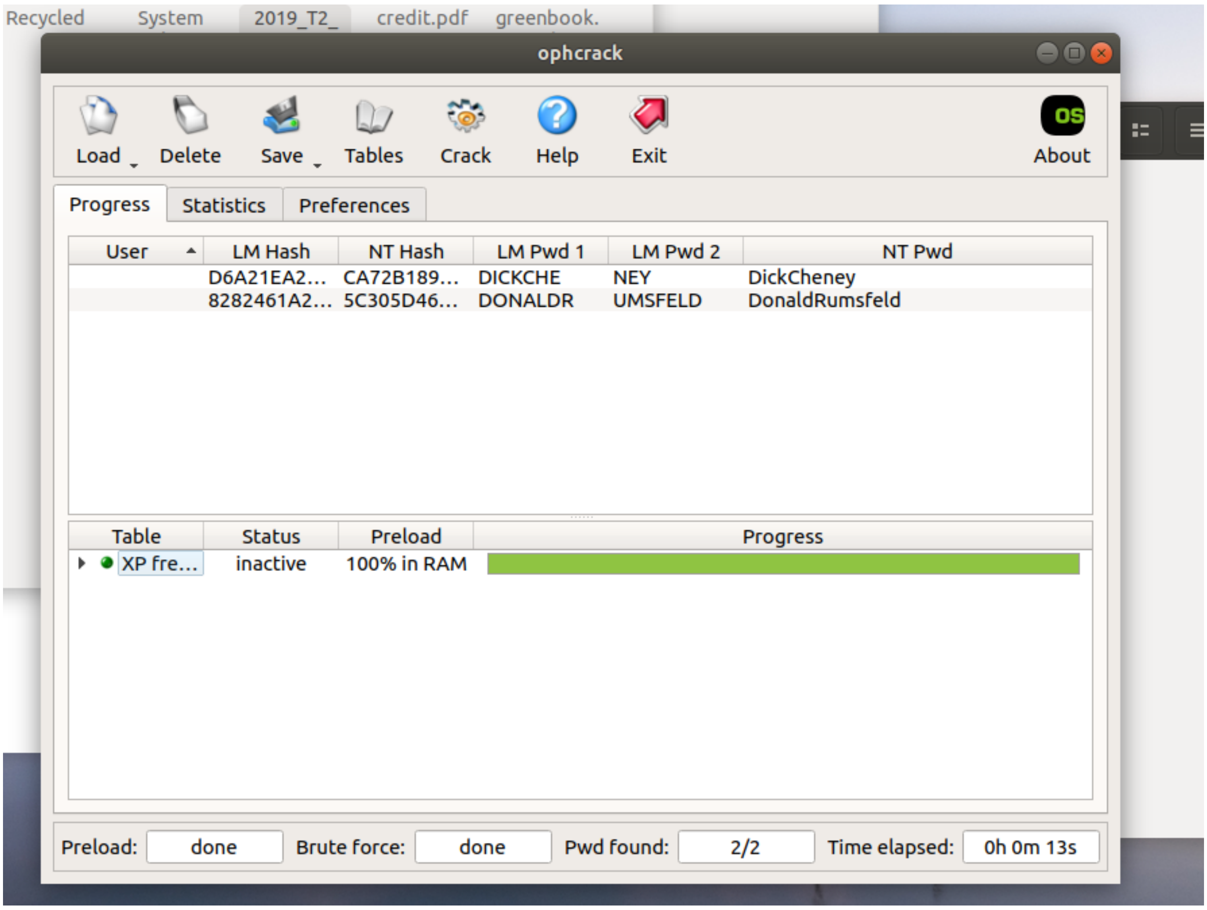
Preload: waiting Brute force: waiting Pwd found: 0/0 Time elapsed: 0h 0m 0s



User	LM Hash	NT Hash	LM Pwd 1	LM Pwd 2	NT Pwd
	D6A21EA2…	CA72B189…			
	8282461A2…	5C305D46…			

Table	Status	Preload	Progress
● XP fre…	inactive	on disk	

Preload: waiting Brute force: waiting Pwd found: 0/0 Time elapsed: 0h 0m 0s



User	LM Hash	NT Hash	LM Pwd 1	LM Pwd 2	NT Pwd
	D6A21EA2…	CA72B189…	DICKCHE	NEY	DickCheney
	8282461A2…	5C305D46…	DONALDR	UMSFELD	DonaldRumsfeld

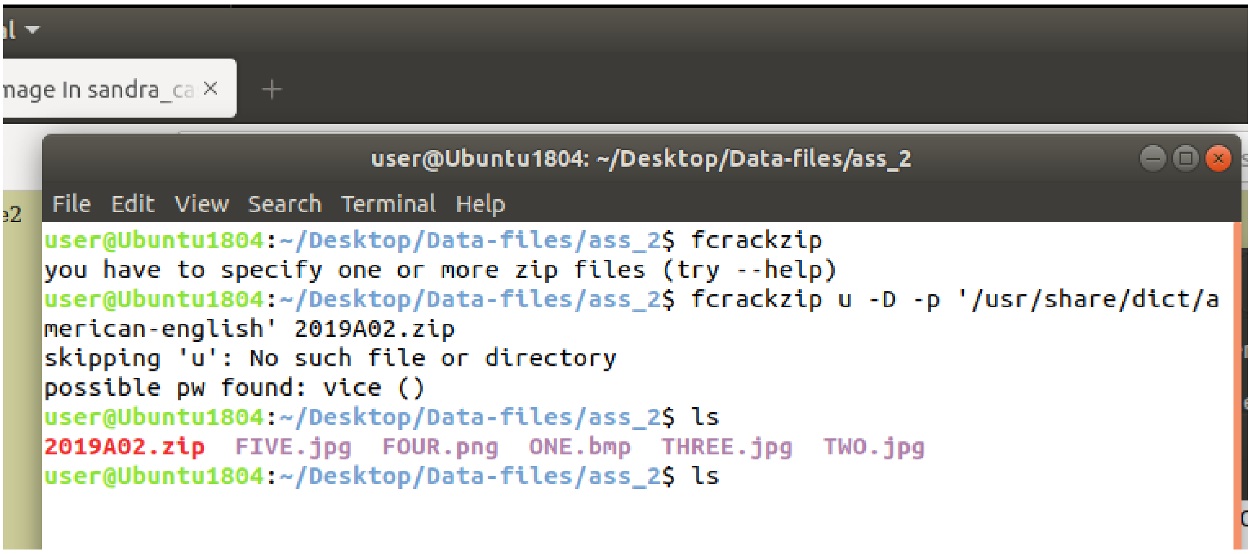
Table	Status	Preload	Progress
● XP fre…	inactive	100% in RAM	🟩

Preload: done Brute force: done Pwd found: 2/2 Time elapsed: 0h 0m 13s

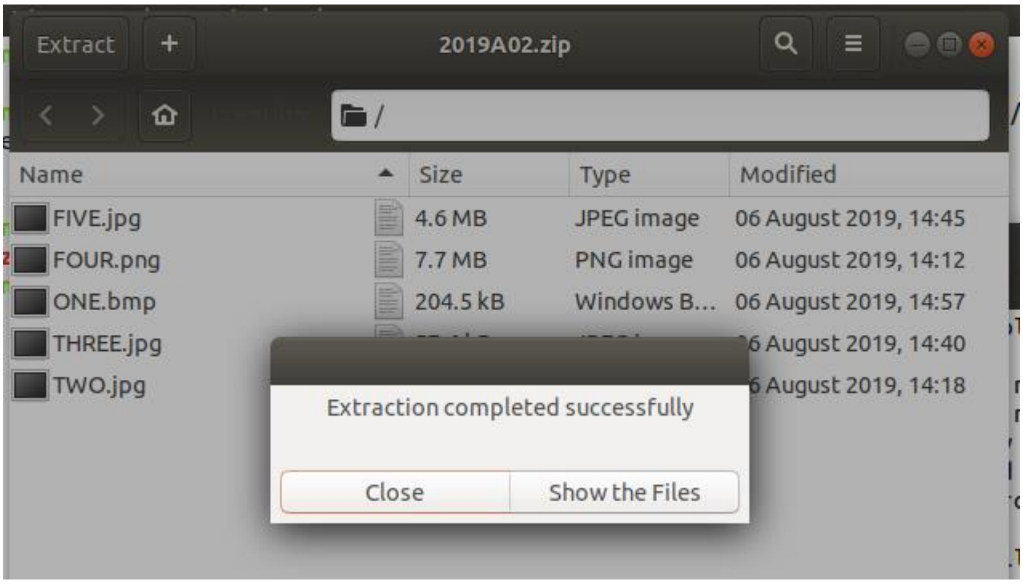
3. Describe the process that you apply to open the downloaded file. Describe whether there is a relationship between this process and the information obtained in Step 2.

Steps performed to open the file were:	1. *I extracted the contents of the zip file using fcrackzip tool.* 2. *For that, I first, brute forced the ZIP file using the word list american-english.* 3. *Then the password was found using brute forcing. It was **vice*** 4. *Then, I opened the ZIP file, it asked for a password since it was password protected. I typed the password vice.* 5. *Finally, the files inside the ZIP file got extracted and was in a viewable format.*

There is a relationship between the decrypted NTLM hashes and how to open the downloaded file because of the thorough technique used in the investigation. Even though the passwords derived from the NTLM hashes weren't used to unlock the ZIP file, recording this process is essential since it demonstrates a comprehensive investigation of all potential entry points and there are possibilities of these passwords being used later. In the end, brute forcing was required to open the ZIP file, highlighting the importance of adaptability and a variety of techniques in digital forensic investigations.

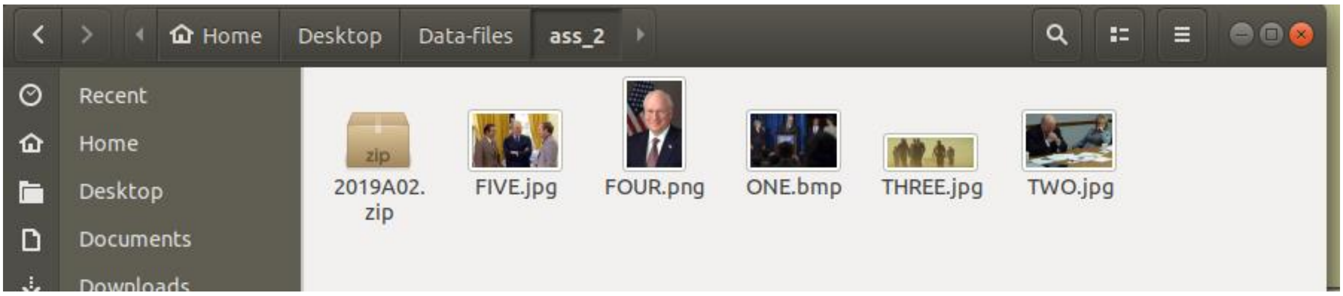

nage In sandra_ca
user@Ubuntu1804: ~/Desktop/Data-files/ass_2
File Edit View Search Terminal Help

```
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ fcrackzip
you have to specify one or more zip files (try --help)
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ fcrackzip u -D -p '/usr/share/dict/a
merican-english' 2019A02.zip
skipping 'u': No such file or directory
possible pw found: vice ()
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ ls
2019A02.zip  FIVE.jpg  FOUR.png  ONE.bmp  THREE.jpg  TWO.jpg
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ ls
```



Name	Size	Type	Modified
FIVE.jpg	4.6 MB	JPEG image	06 August 2019, 14:45
FOUR.png	7.7 MB	PNG image	06 August 2019, 14:12
ONE.bmp	204.5 kB	Windows B…	06 August 2019, 14:57
THREE.jpg			6 August 2019, 14:40
TWO.jpg			6 August 2019, 14:18

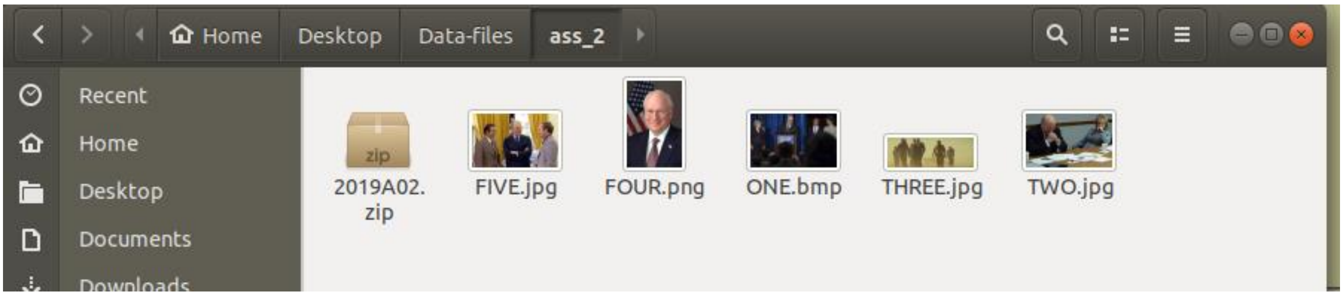
Extraction completed successfully

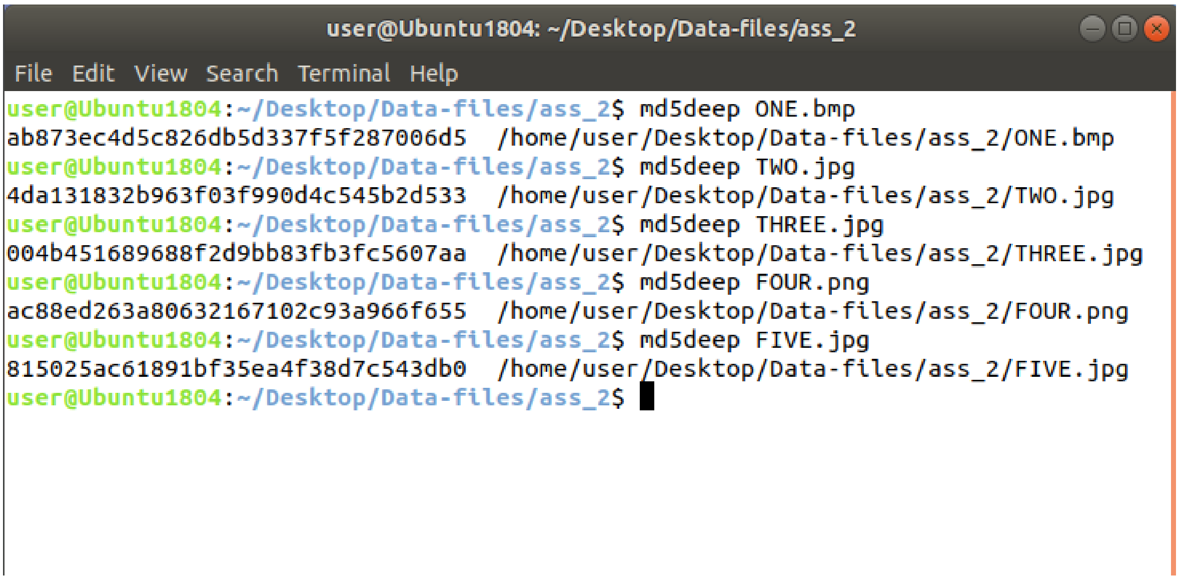


2019A02.zip FIVE.jpg FOUR.png ONE.bmp THREE.jpg TWO.jpg

4. Describe the actual content of the encrypted file that you identified in Step 3. If there are multiple files, list their file names, types and MD5 hash values. Describe the visual contents in each file.

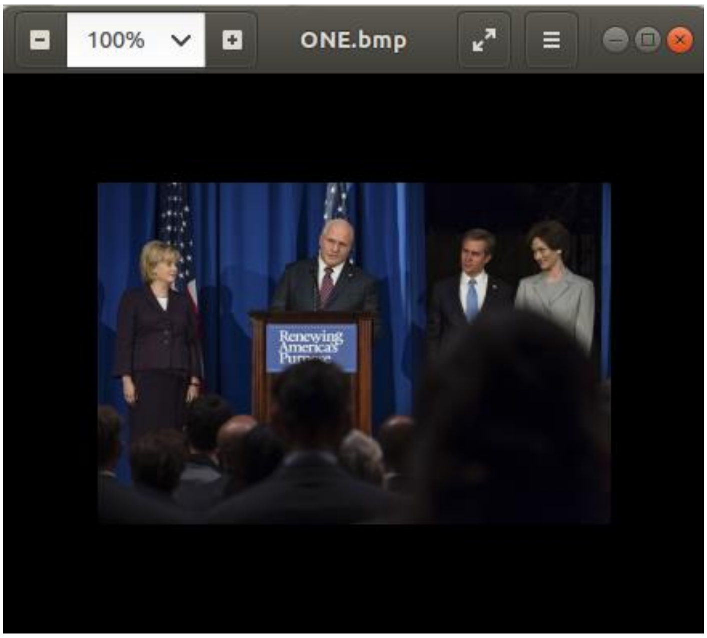
Content description	*The actual files that were extracted from the downloaded ZIP files were image files. There were totally 5 image files extracted.* *In the **ONE.bmp** file it was a group of people on a stage. Looks like white house's speech giving stage. Then I observed that it was from the movie called Vice.* *In the **TWO.jpg** file there were two people sitting near a table being anxious about something while looking down at a file on the table. This was also a scene from the movie Vice.* *In the **THREE.jpg** file I saw a group of people walking on a sandy place. This was also a scene from the movie called Vice.* *In the **FOUR.png** file, I saw an elderly person face.* *In the **FIVE.jpg**, there were a group of three people discussing something.*



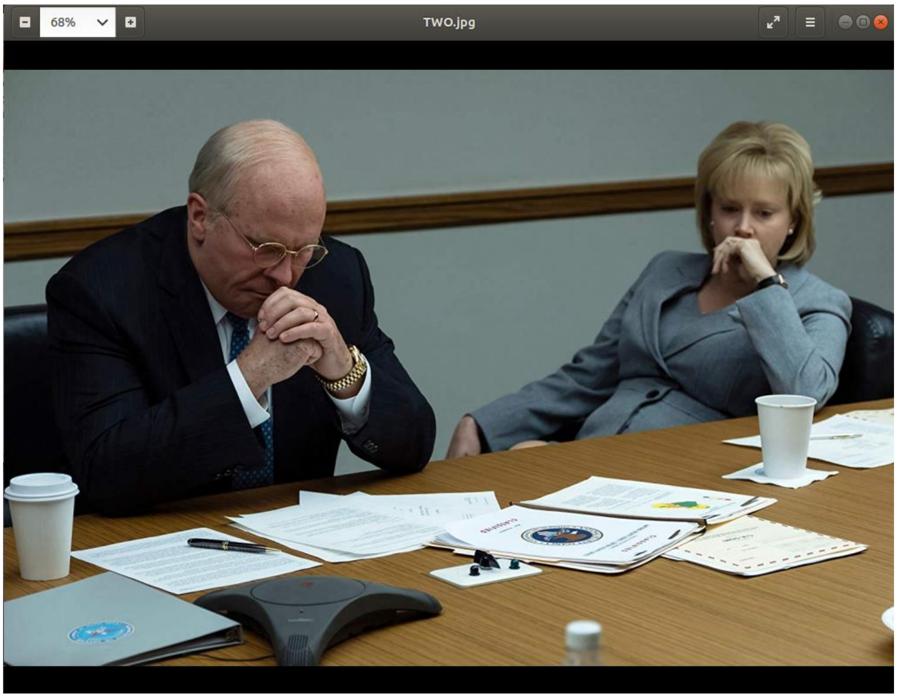

user@Ubuntu1804: ~/Desktop/Data-files/ass_2
File Edit View Search Terminal Help

```
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ md5deep ONE.bmp
ab873ec4d5c826db5d337f5f287006d5  /home/user/Desktop/Data-files/ass_2/ONE.bmp
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ md5deep TWO.jpg
4da131832b963f03f990d4c545b2d533  /home/user/Desktop/Data-files/ass_2/TWO.jpg
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ md5deep THREE.jpg
004b451689688f2d9bb83fb3fc5607aa  /home/user/Desktop/Data-files/ass_2/THREE.jpg
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ md5deep FOUR.png
ac88ed263a80632167102c93a966f655  /home/user/Desktop/Data-files/ass_2/FOUR.png
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ md5deep FIVE.jpg
815025ac61891bf35ea4f38d7c543db0  /home/user/Desktop/Data-files/ass_2/FIVE.jpg
user@Ubuntu1804:~/Desktop/Data-files/ass_2$
```

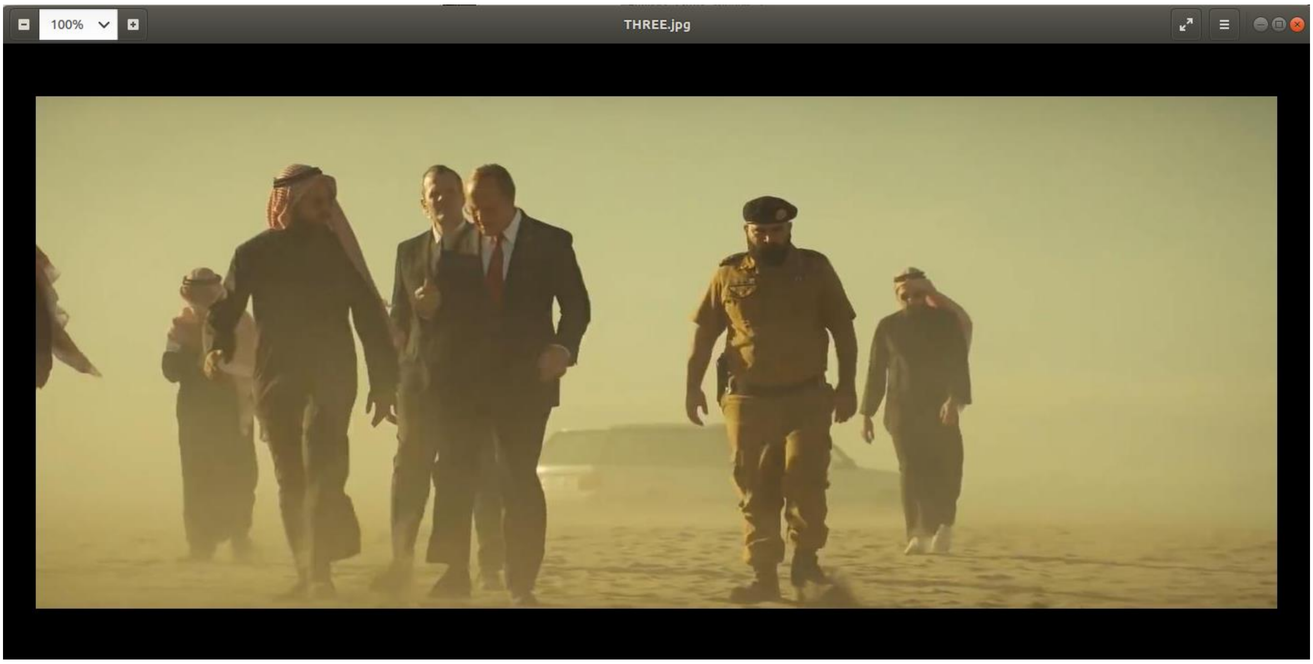
File Name	*File Type*	*MD5 Hash Value*
ONE	*.bmp*	*ab873ec4d5c826db5d337f5f287006d5*



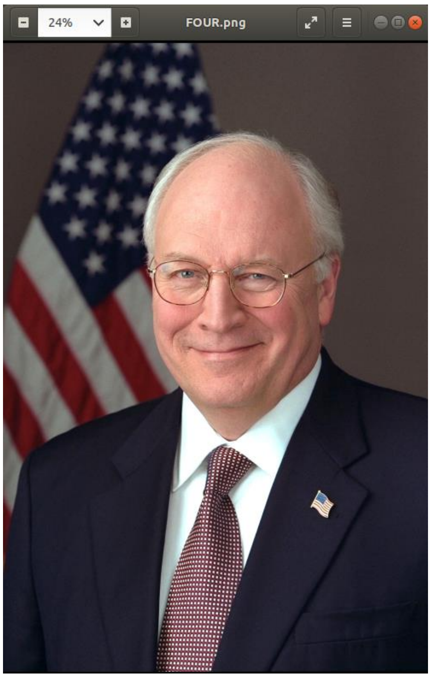
File Name	*File Type*	*MD5 Hash Value*
TWO	*.jpg*	*4da131832b963f03f990d4c545b2d533*



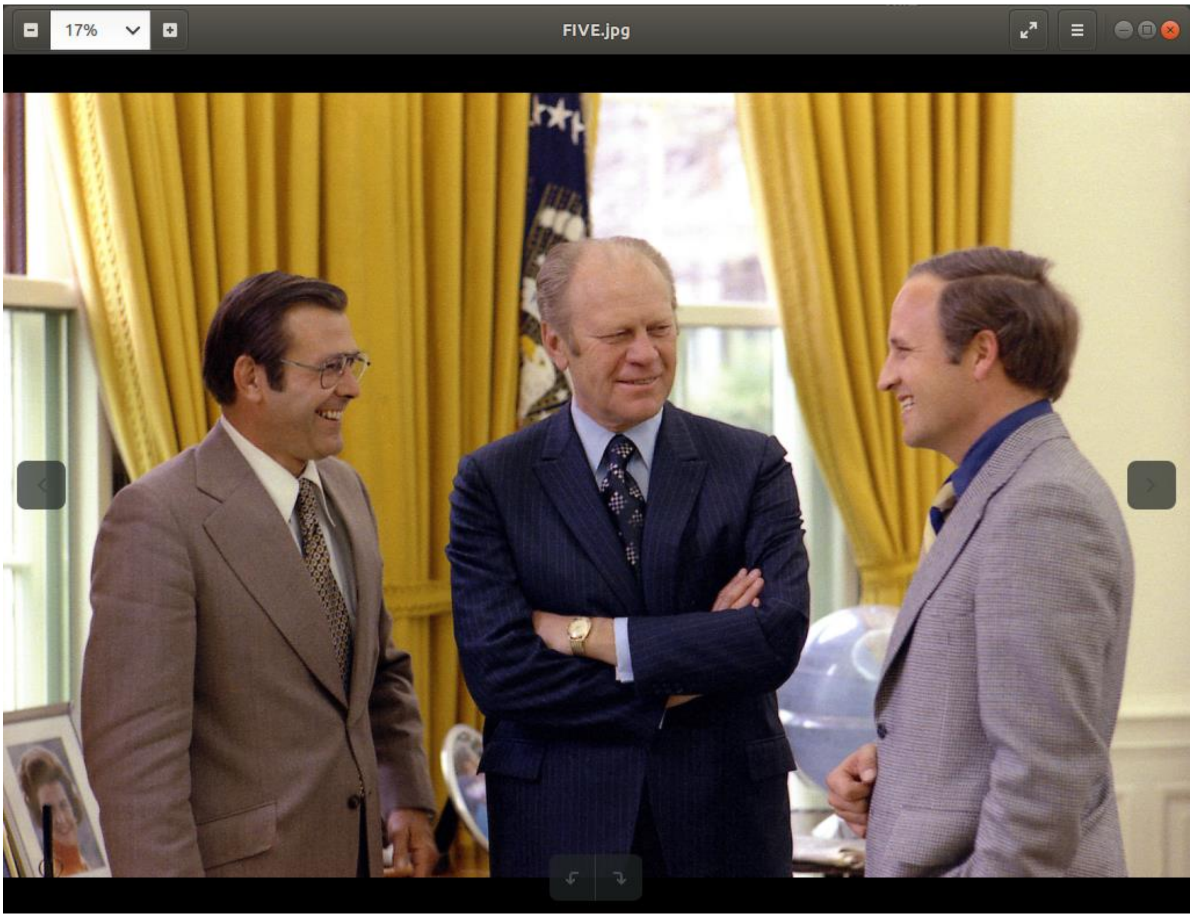
File Name	File Type	MD5 Hash Value
THREE	.jpg	004b451689688f2d9bb83fb3fc5607aa



File Name	File Type	MD5 Hash Value
FOUR	.png	ac88ed263a80632167102c93a966f655



File Name	*File Type*	*MD5 Hash Value*
FIVE	*.jpg*	*815025ac61891bf35ea4f38d7c543db0*



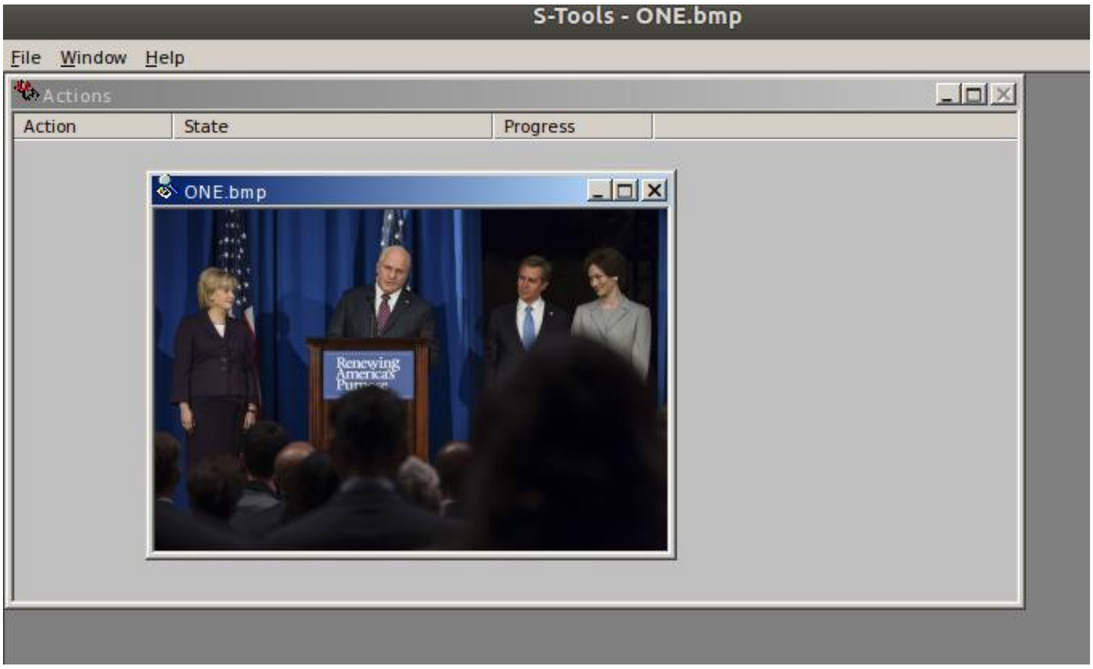
5. What tools will you now use to proceed your investigation and why?

Tool	*Reason*
S-Tools	*S-Tools is a steganographic analysis tool designed specifically for BMP files. Because BMP files are uncompressed and have a consistent data structure, they are useful for data concealment and are frequently used in steganography. In this instance, ONE.bmp was analyzed using S-Tools in order to find and extract any hidden information that might be present in the picture and be important for the inquiry.*
jpseek	*JPG picture steganographic content analysis is the specialty of jpseek. Because JPG files are so widely used, there is less reason to suspect them when it comes to data concealing. TWO.jpg was examined using jpseek to make sure that no hidden information was missed, as it is necessary to carefully check all JPG images for hidden messages or data.*
Foremost	*With the help of its own data structures, headers, and footers, the data recovery utility Foremost is able to extract files. It works especially well for extracting files that might have been buried or erased. Foremost was utilized on THREE.jpg in this experiment to retrieve any possibly embedded or obscured data that conventional viewing techniques could overlook. This guarantees the discovery of all relevant evidence.*
OpenPuff	*OpenPuff is a flexible steganography application that works with a variety of file formats, including PNG files, and enables data extraction and concealing. Because of their lossless compression, which maintains the concealed data without sacrificing image quality, PNG files are utilized in steganography. Because it is capable of efficiently locating hidden data that could be essential to the inquiry, OpenPuff was selected to examine FOUR.png for any hidden information.*
CrypTool	*A powerful tool for cryptography analysis, CrypTool can perform a wide range of encryption and decryption operations. Although its main purpose is not steganography, it is capable of deciphering encrypted data buried in files. In this instance, CrypTool was used to examine FIVE.jpg in order to locate and unlock any encrypted data that may have been contained. This allowed investigators to completely access any information that might have been hidden in the picture.*
HxD	*HxD's ability to perform detailed examination and manipulation of the FIVE.jpg's binary data provides a deeper level of analysis. It helps in identifying and extracting hidden data, verifying the integrity of the file, and uncovering any embedded content that other tools might miss.*

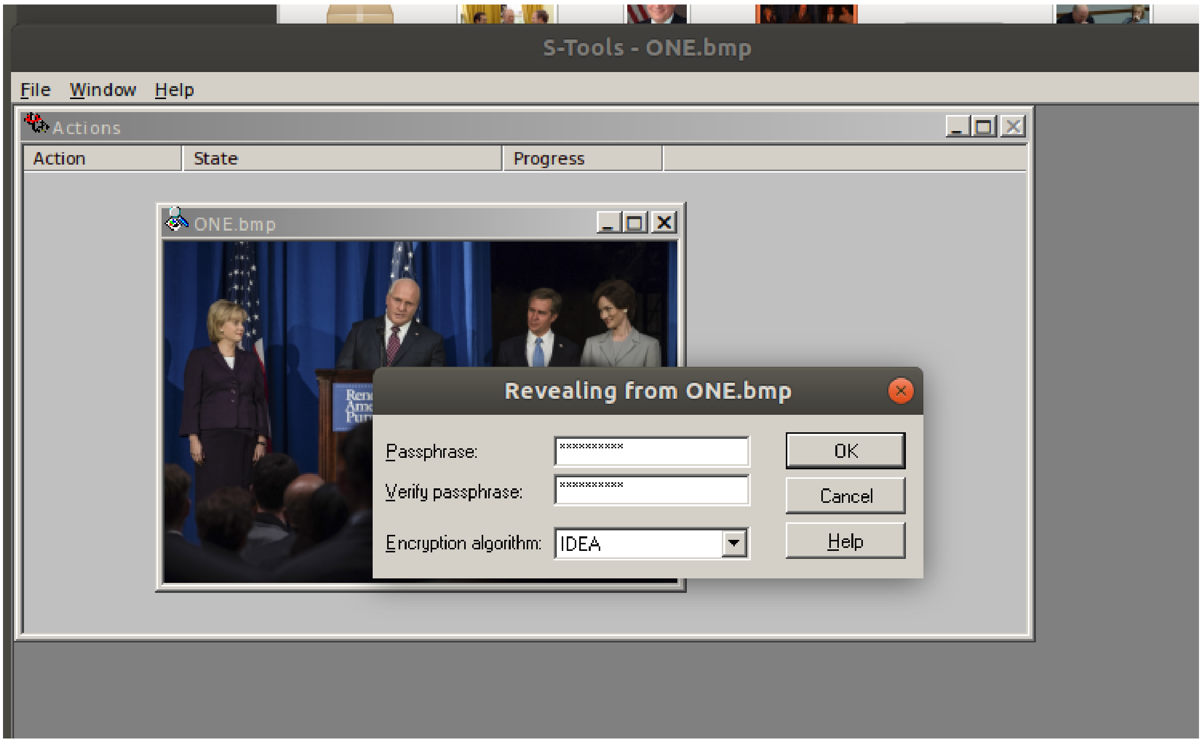
6. Describe how your investigation proceeded at this point, including screen shots.

1. *For the first image file, ONE.bmp, I used the S-Tools to reveal any hidden files inside it.*

*For that I first opened the tool using the command **wine ~/Desktop/win-tools/jphide\ and Stegbreak/S-tools/S-Tools.exe**. then drag and dropped the ONE.bmp file inside it*



*Then I right clicked the image file and selected the reveal option. It prompted for a password, since I have extracted two passwords using the NTLM hashes, I used the first password, **DickCheney** on this to reveal the hidden file.*

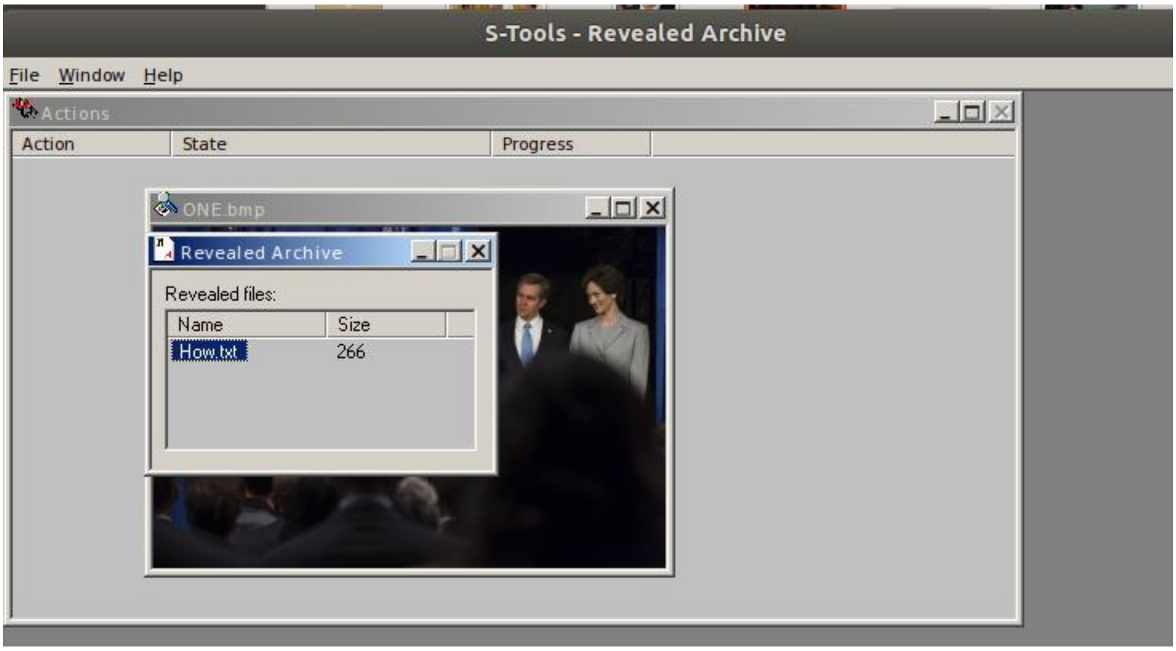


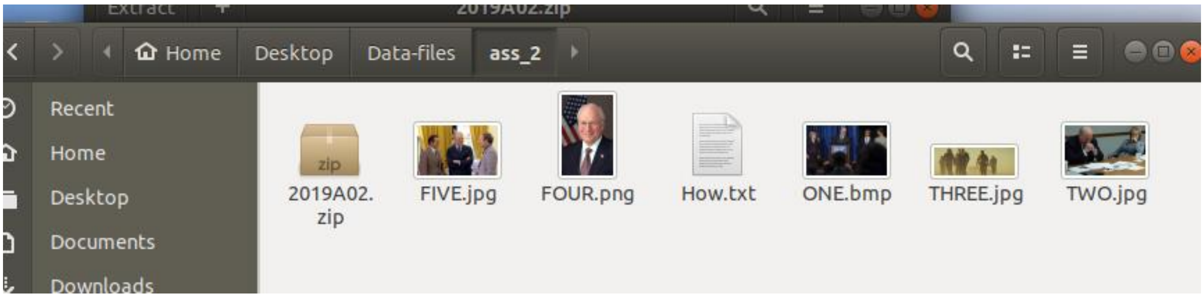
Passphrase: ***********

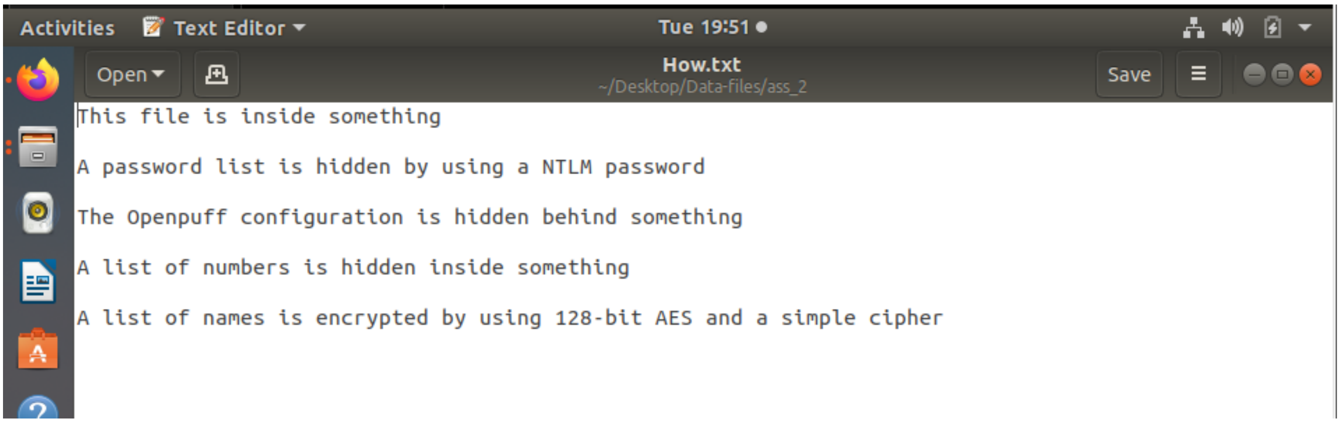
Verify passphrase: ***********

Encryption algorithm: IDEA

After entering the password, a text file was revealed, How.txt. I opened it using the text editor and found some clues on how to find the rest of the hidden files and evidences.





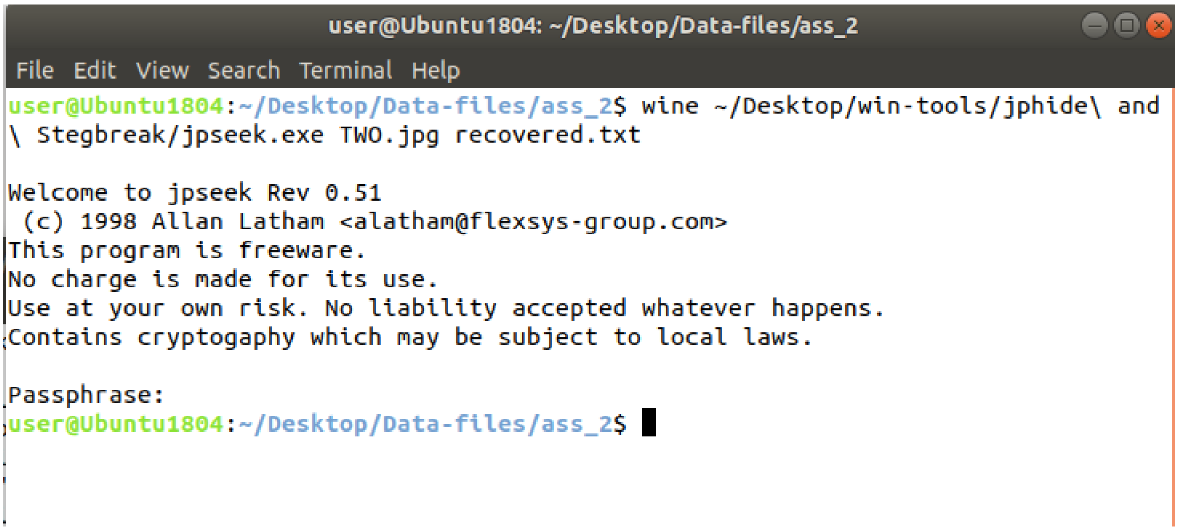


```
This file is inside something

A password list is hidden by using a NTLM password

The Openpuff configuration is hidden behind something

A list of numbers is hidden inside something

A list of names is encrypted by using 128-bit AES and a simple cipher
```

2. *Since I have now found the hidden file in the ONE.bmp, I'll now move on to the TWO.jpg file. The How.txt file says that a password list is hidden using the NTLM hash password. Using this clue, I started to analyse the TWO.jpg.*

   For that I used the tool jpseek to recover the hidden password list inside it. In order to do that, I typed in the command ***wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/jpseek.exe TWO.jpg recovered.txt.***

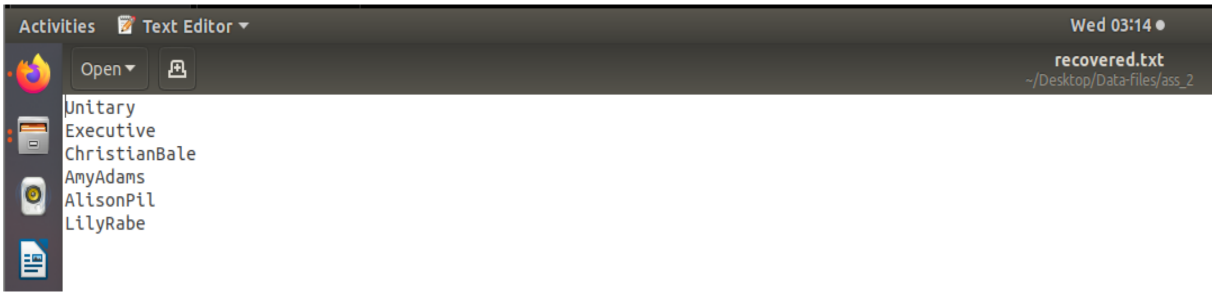
   When prompted for the passphrase, I entered the second NTLM password ***DonaldRumsfeld.***



user@Ubuntu1804: ~/Desktop/Data-files/ass_2
File Edit View Search Terminal Help

```
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ wine ~/Desktop/win-tools/jphide\ and
\ Stegbreak/jpseek.exe TWO.jpg recovered.txt

Welcome to jpseek Rev 0.51
 (c) 1998 Allan Latham <alatham@flexsys-group.com>
This program is freeware.
No charge is made for its use.
Use at your own risk. No liability accepted whatever happens.
Contains cryptogaphy which may be subject to local laws.

Passphrase:
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ 
```

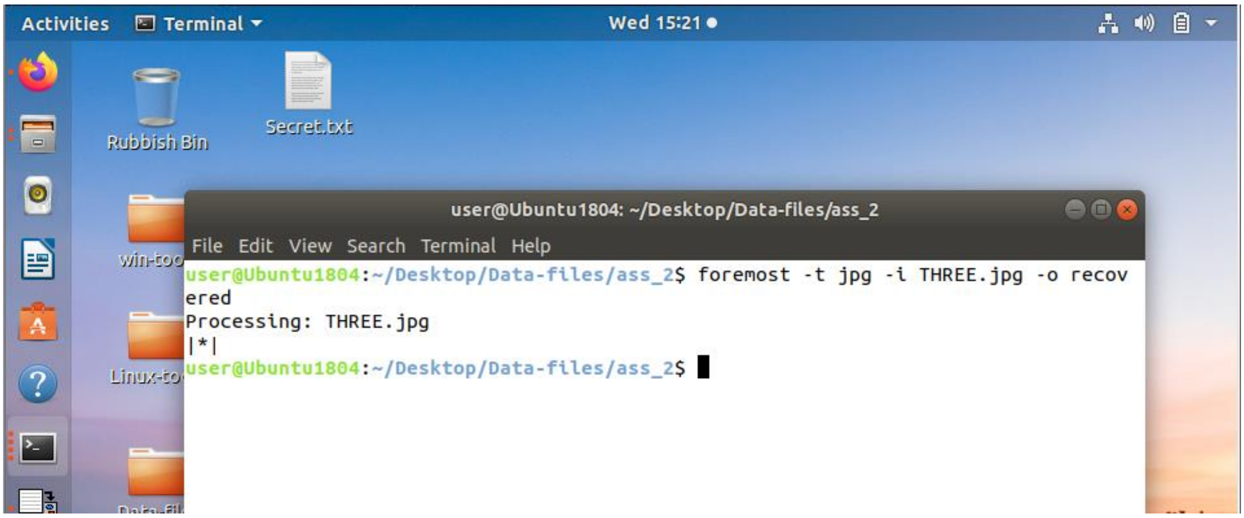
A text file was extracted, recovered.txt. it contained a list of passwords, as mentioned in the How.txt file.



```
Unitary
Executive
ChristianBale
AmyAdams
AlisonPil
LilyRabe
```

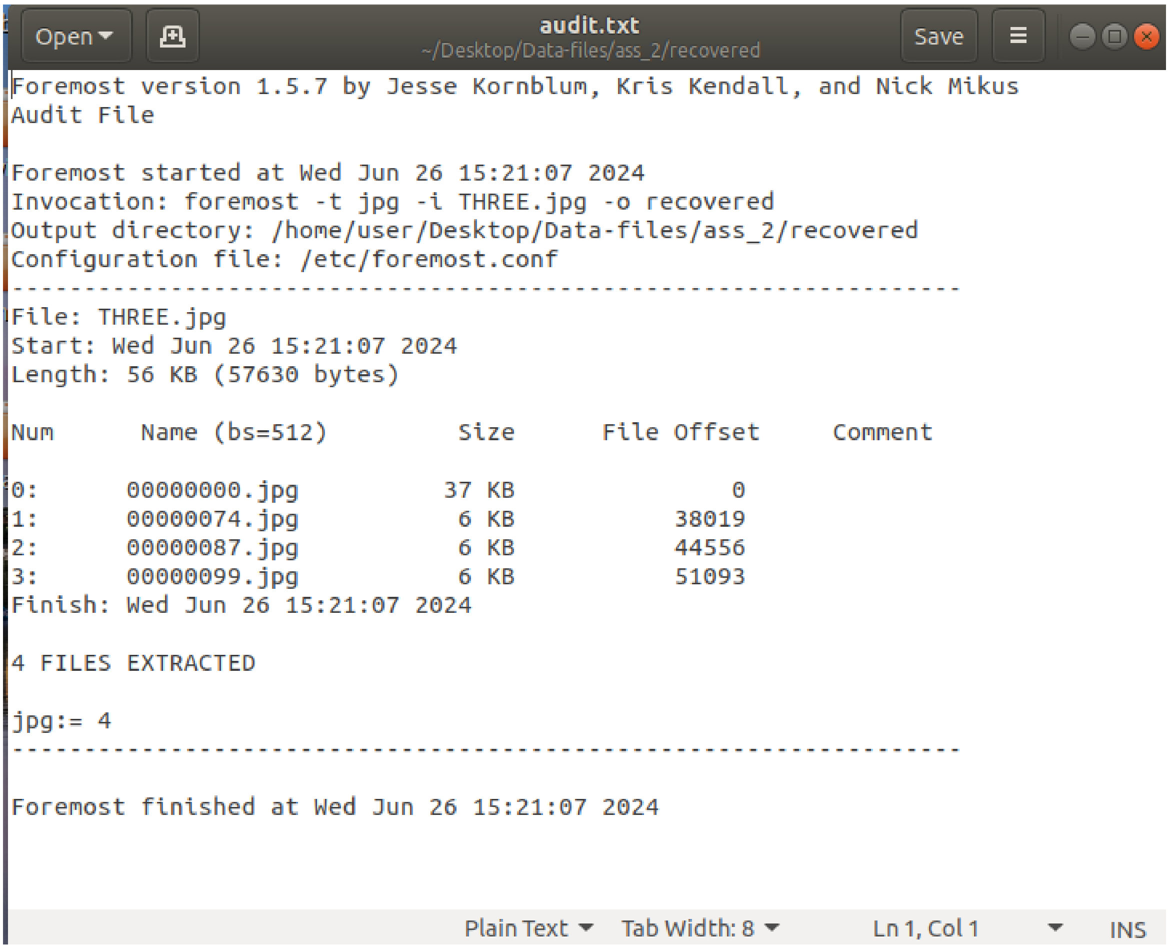
3. *According to the How.txt, now we should find the OpenPuff configuration. For that let's analyse the THREE.jpg. it was said that the OpenPuff file configuration has been hidden. So, I decided to use the tool called Foremost on THREE.jpg to recover any hidden OpenPuff configuration files.*

   I used the command ***foremost -t jpg -I THREE.jpg -o recovered.***

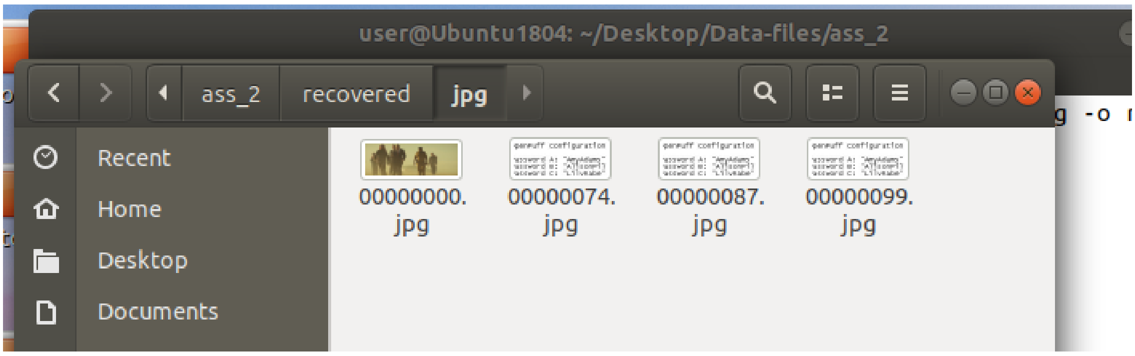

Activities
Terminal
Wed 15:21
Rubbish Bin
Secret.txt
win-too
Linux-to
Data-fil
user@Ubuntu1804: ~/Desktop/Data-files/ass_2
File Edit View Search Terminal Help

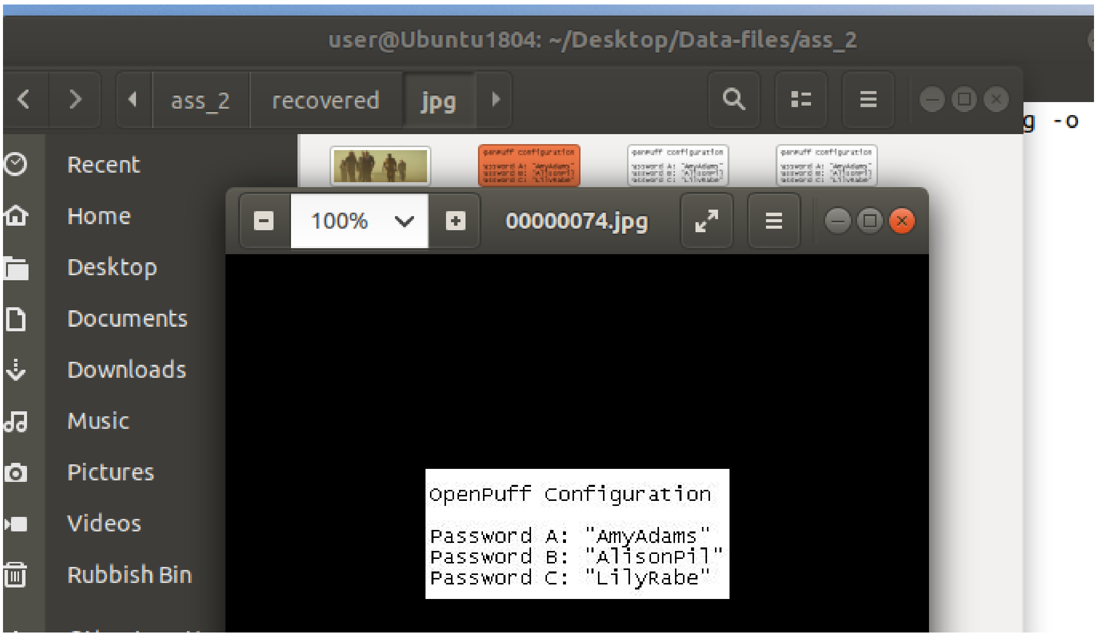
```
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ foremost -t jpg -i THREE.jpg -o recov
ered
Processing: THREE.jpg
|*|
user@Ubuntu1804:~/Desktop/Data-files/ass_2$
```

There were four text files extracted in the folder recover. Three of them contained the same OpenPuff file configuration and the fourth one contained an audit.txt file.



```
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus
Audit File

Foremost started at Wed Jun 26 15:21:07 2024
Invocation: foremost -t jpg -i THREE.jpg -o recovered
Output directory: /home/user/Desktop/Data-files/ass_2/recovered
Configuration file: /etc/foremost.conf
------------------------------------------------------------------
File: THREE.jpg
Start: Wed Jun 26 15:21:07 2024
Length: 56 KB (57630 bytes)

Num      Name (bs=512)         Size      File Offset     Comment

0:      00000000.jpg          37 KB              0
1:      00000074.jpg           6 KB          38019
2:      00000087.jpg           6 KB          44556
3:      00000099.jpg           6 KB          51093
Finish: Wed Jun 26 15:21:07 2024

4 FILES EXTRACTED

jpg:= 4
------------------------------------------------------------------

Foremost finished at Wed Jun 26 15:21:07 2024
```

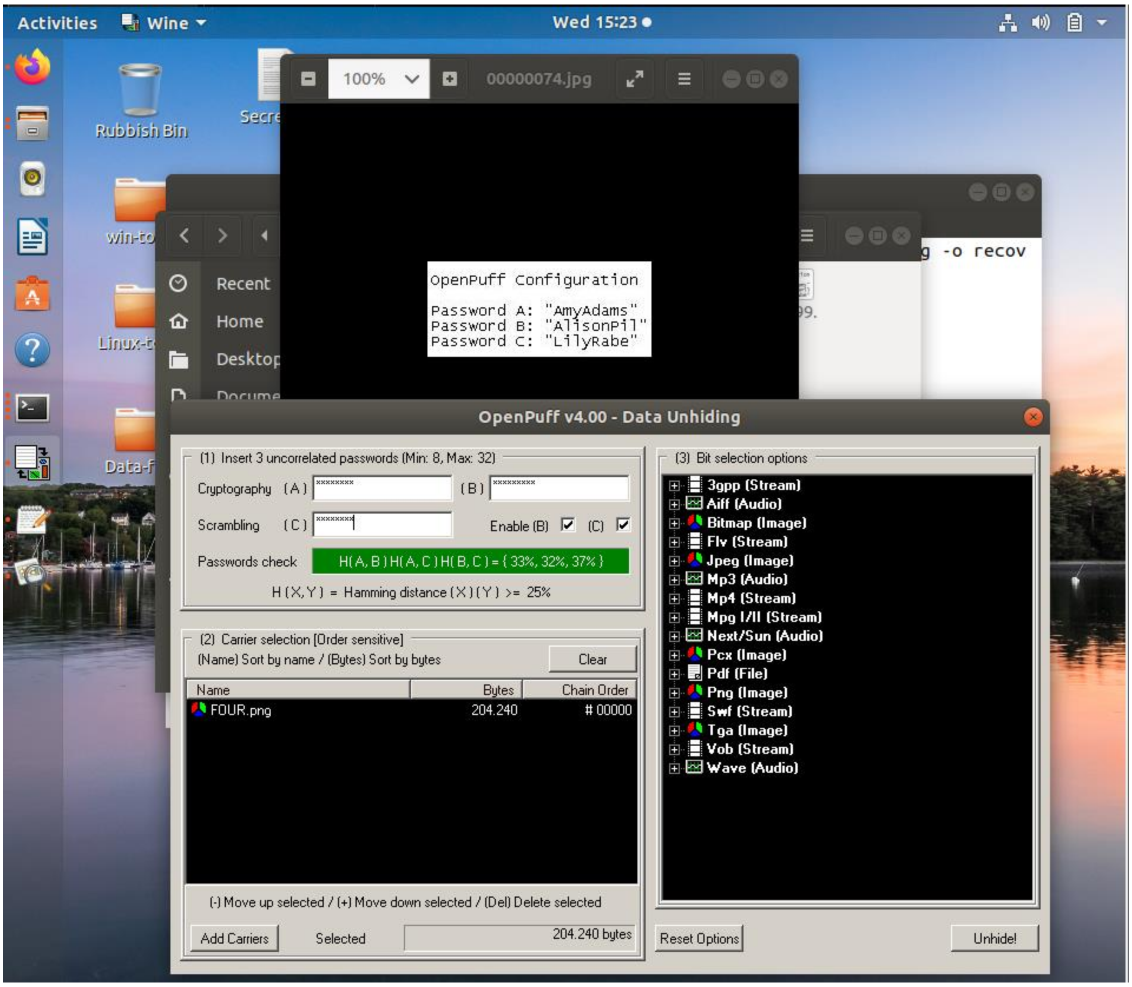




```
OpenPuff Configuration

Password A: "AmyAdams"
Password B: "AlisonPil"
Password C: "LilyRabe"
```

4. *Now moving on to the FOUR.png file. Since we have now got the OpenPuff configurations, we can use that to extract the hidden list of numbers from the FOUR.png file as mentioned in the How.txt file.*

   For that I opened the OpenPuff tool, entered the OpenPuff configuration that I got in the previous step. Then added the carrier as FOUR.png. now I clicked the Unhide button.



```
OpenPuff Configuration

Password A: "AmyAdams"
Password B: "AlisonPil"
Password C: "LilyRabe"
```

OpenPuff v4.00 - Data Unhiding

(1) Insert 3 uncorrelated passwords (Min: 8, Max: 32)

Cryptography (A) ******** (B) ********

Scrambling (C) ******** Enable (B) ☑ (C) ☑

Passwords check H(A,B) H(A,C) H(B,C) = (33%, 32%, 37%)

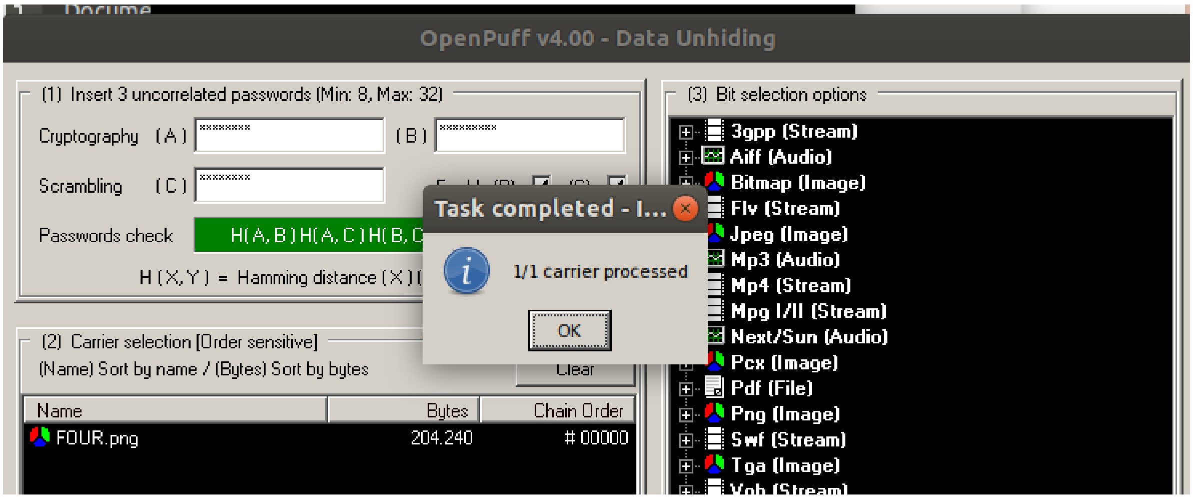
H(X,Y) = Hamming distance (X)(Y) >= 25%

(2) Carrier selection [Order sensitive]

Name	Bytes	Chain Order
FOUR.png	204.240	# 00000

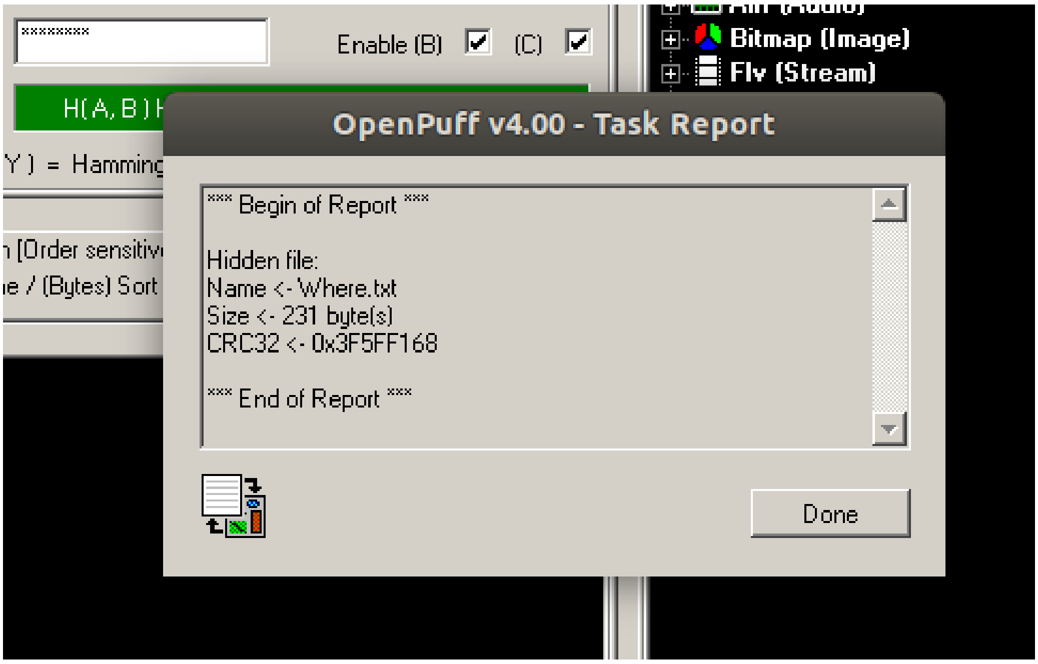
Selected 204.240 bytes

After unhiding, a text file, Where.txt got extracted containing a list of numbers. As I suspect, these could be the phone numbers of the members involved in the drug warehouse

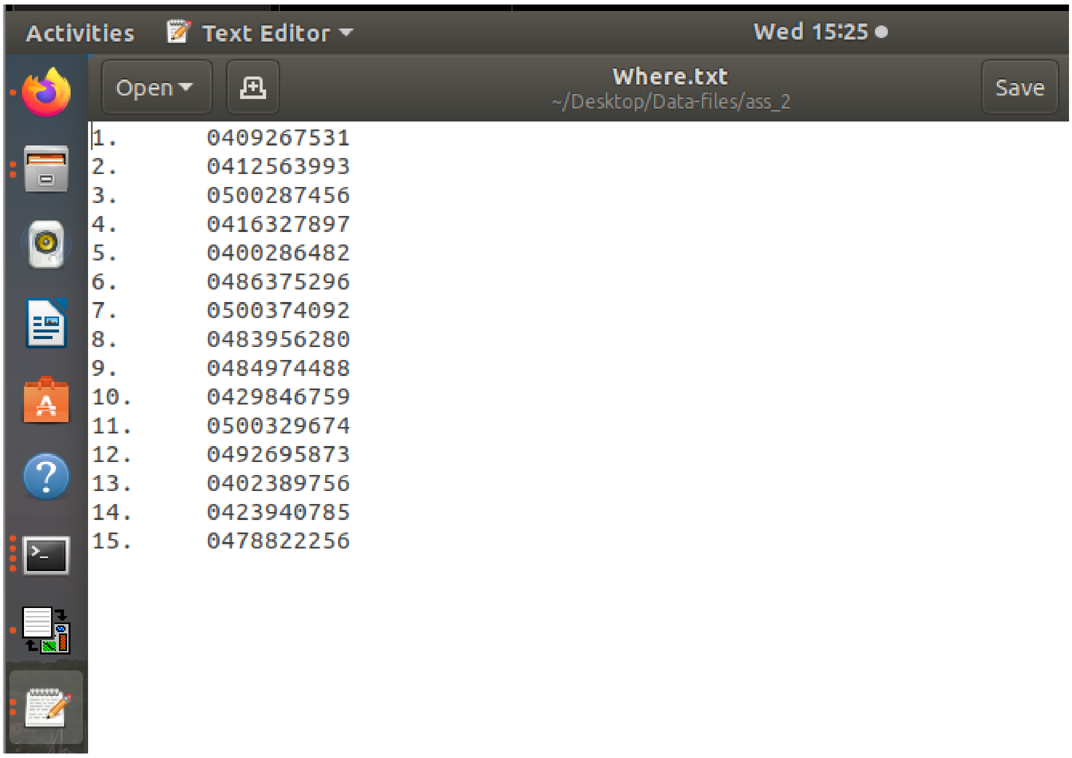


Task completed - I...

1/1 carrier processed



```
*** Begin of Report ***

Hidden file:
Name <- Where.txt
Size <- 231 byte(s)
CRC32 <- 0x3F5FF168

*** End of Report ***
```

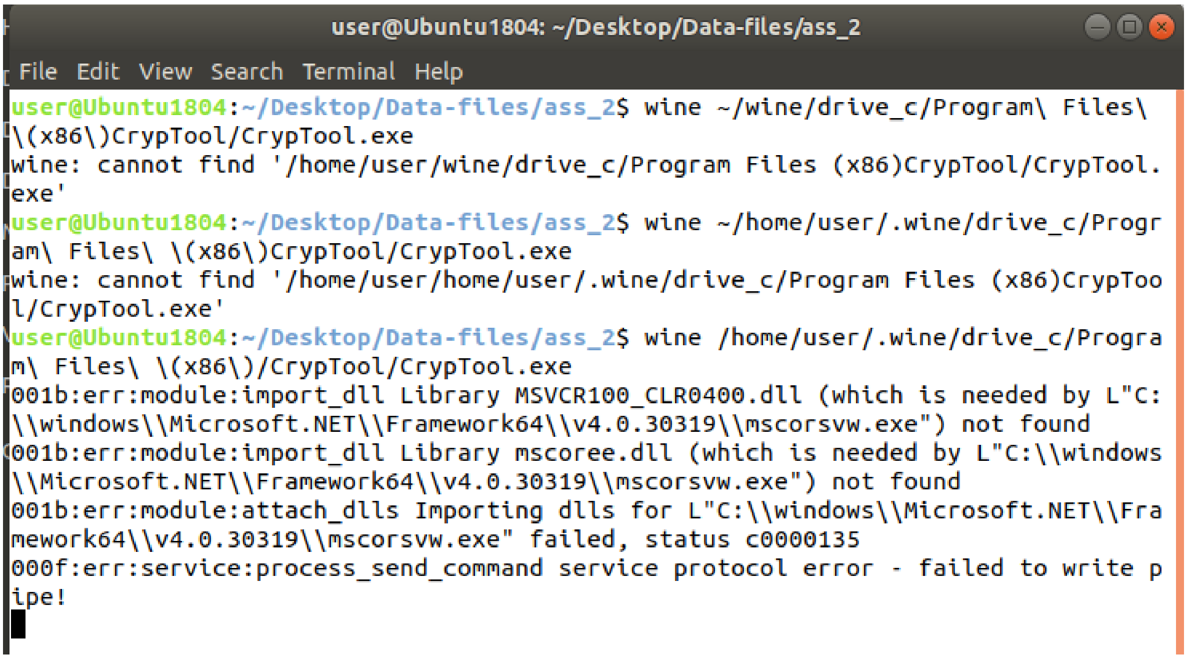


```
1.      0409267531
2.      0412563993
3.      0500287456
4.      0416327897
5.      0400286482
6.      0486375296
7.      0500374092
8.      0483956280
9.      0484974488
10.     0429846759
11.     0500329674
12.     0492695873
13.     0402389756
14.     0423940785
15.     0478822256
```

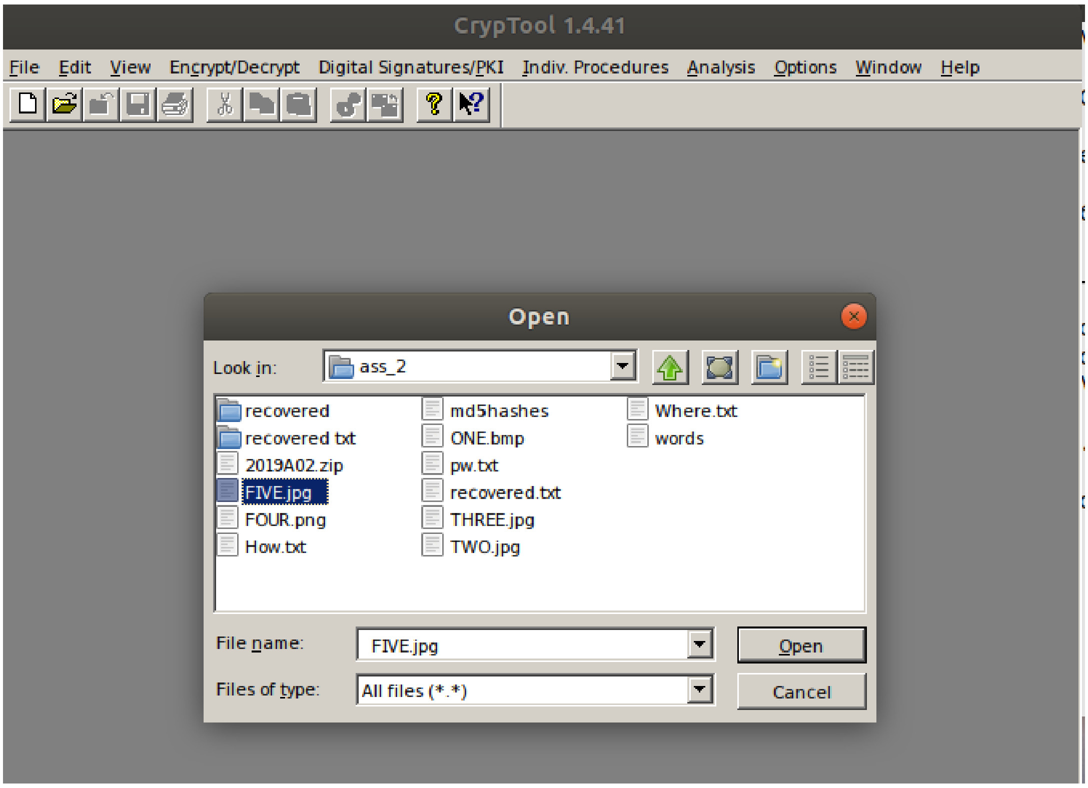
5. *Now according to the How.txt, the last clue states that a list of names are hidden, so in order to find that, I'll have to now analyse the FIVE.jpg.*

   To do that, I used a tool called CrypTool to decrypt and extract any hidden information in the FIVE.jpg image file. I entered the command ***wine /home/user/.wine/drive_c/Program\ Files\ \(x86\)/CrypTool.exe***

   For that I first loaded the FIVE.jpg in the Cryptool and decrypted it using the AES encryption and then tried to decode it using Base64 decoding. But unfortunately, it didn't work. For the AES decryption key, I converted the name ChristianBale that was found from the list of passwords found earlier into hex value using an online tool duplichecker.com.


user@Ubuntu1804: ~/Desktop/Data-files/ass_2
File Edit View Search Terminal Help

```
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ wine ~/wine/drive_c/Program\ Files\
\(x86\)CrypTool/CrypTool.exe
wine: cannot find '/home/user/wine/drive_c/Program Files (x86)CrypTool/CrypTool.
exe'
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ wine ~/home/user/.wine/drive_c/Progr
am\ Files\ \(x86\)CrypTool/CrypTool.exe
wine: cannot find '/home/user/home/user/.wine/drive_c/Program Files (x86)CrypToo
l/CrypTool.exe'
user@Ubuntu1804:~/Desktop/Data-files/ass_2$ wine /home/user/.wine/drive_c/Progra
m\ Files\ \(x86\)/CrypTool/CrypTool.exe
001b:err:module:import_dll Library MSVCR100_CLR0400.dll (which is needed by L"C:
\\windows\\Microsoft.NET\\Framework64\\v4.0.30319\\mscorsvw.exe") not found
001b:err:module:import_dll Library mscoree.dll (which is needed by L"C:\\windows
\\Microsoft.NET\\Framework64\\v4.0.30319\\mscorsvw.exe") not found
001b:err:module:attach_dlls Importing dlls for L"C:\\windows\\Microsoft.NET\\Fra
mework64\\v4.0.30319\\mscorsvw.exe" failed, status c0000135
000f:err:service:process_send_command service protocol error - failed to write p
ipe!
```

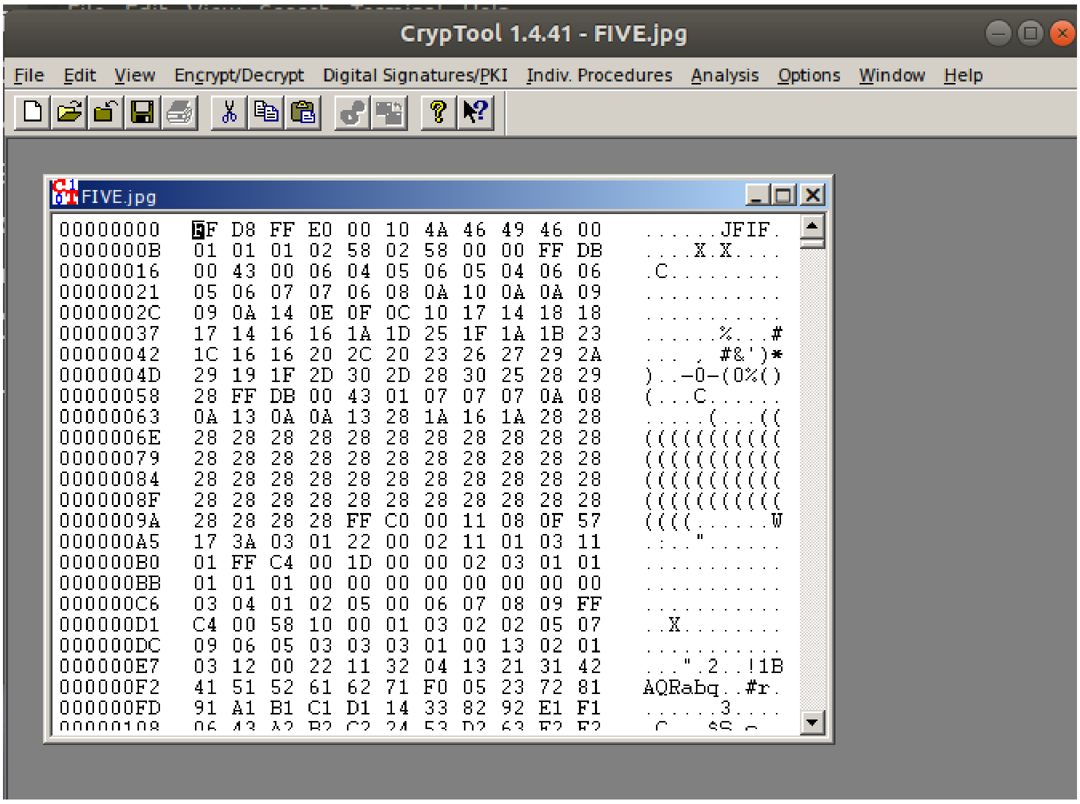

CrypTool 1.4.41
File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv. Procedures Analysis Options Window Help
Open

Look in: ass_2

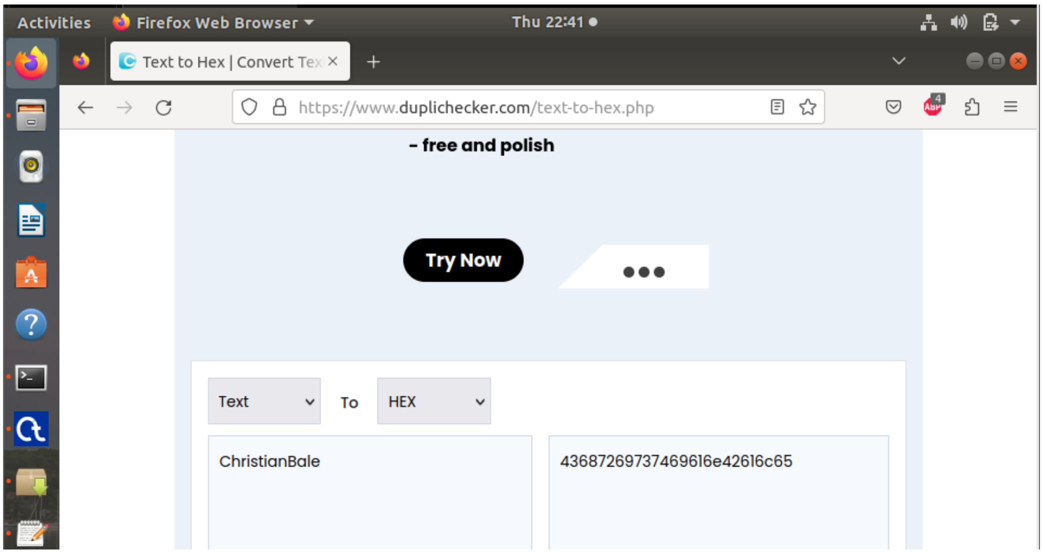
```
recovered        md5hashes      Where.txt
recovered txt    ONE.bmp        words
2019A02.zip      pw.txt
FIVE.jpg         recovered.txt
FOUR.png         THREE.jpg
How.txt          TWO.jpg
```

File name: FIVE.jpg

Files of type: All files (*.*)


CrypTool 1.4.41 - FIVE.jpg
File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv. Procedures Analysis Options Window Help
FIVE.jpg

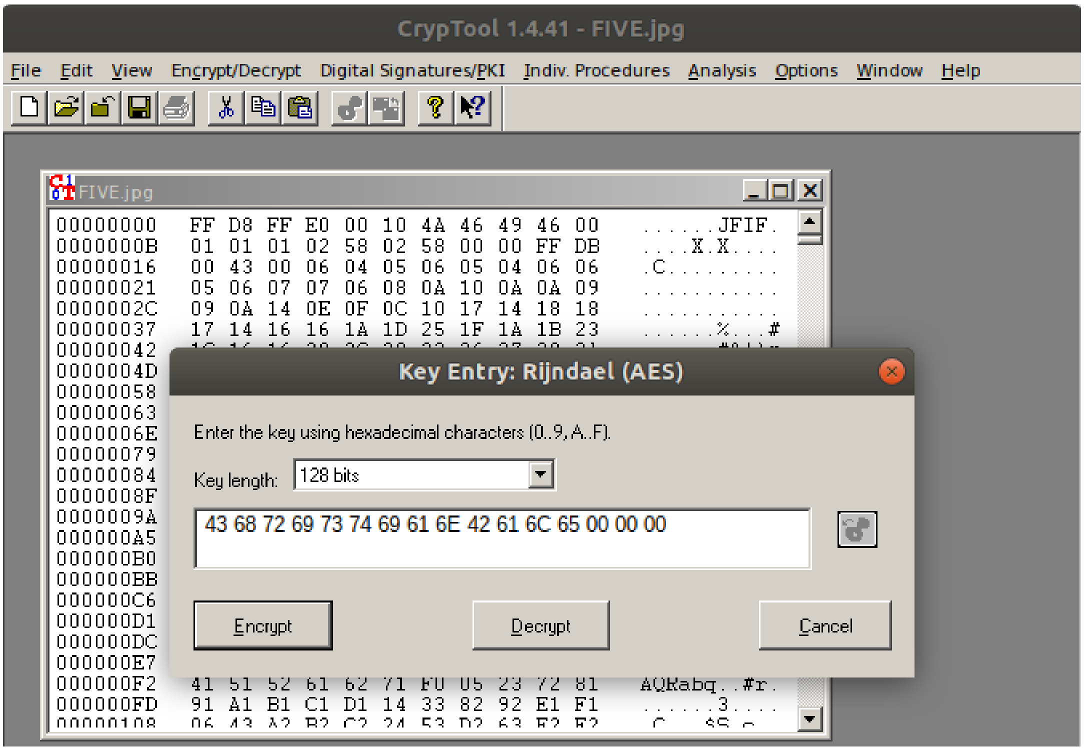
```
00000000   FF D8 FF E0 00 10 4A 46 49 46 00    ......JFIF.
0000000B   01 01 01 02 58 02 58 00 00 FF DB    ....X.X....
00000016   00 43 00 06 04 05 06 05 04 06 06    .C.........
00000021   05 06 07 07 06 08 0A 10 0A 0A 09    ...........
0000002C   09 0A 14 0E 0F 0C 10 17 14 18 18    ...........
00000037   17 14 16 16 1A 1D 25 1F 1A 1B 23    ......%...#
00000042   1C 16 16 20 2C 20 23 26 27 29 2A    ... , #&')*
0000004D   29 19 1F 2D 30 2D 28 30 25 28 29    )..-0-(0%()
00000058   28 FF DB 00 43 01 07 07 07 0A 08    (...C......
00000063   0A 13 0A 0A 13 28 1A 16 1A 28 28    .....(...((
0000006E   28 28 28 28 28 28 28 28 28 28 28    (((((((((((
00000079   28 28 28 28 28 28 28 28 28 28 28    (((((((((((
00000084   28 28 28 28 28 28 28 28 28 28 28    (((((((((((
0000008F   28 28 28 28 28 28 28 28 28 28 28    (((((((((((
0000009A   28 28 28 28 FF C0 00 11 08 0F 57    ((((......W
000000A5   17 3A 03 01 22 00 02 11 01 03 11    .:.."......
000000B0   01 FF C4 00 1D 00 00 02 03 01 01    ...........
000000BB   01 01 01 00 00 00 00 00 00 00 00    ...........
000000C6   03 04 01 02 05 00 06 07 08 09 FF    ...........
000000D1   C4 00 58 10 00 01 03 02 02 05 07    ..X........
000000DC   09 06 05 03 03 03 01 00 13 02 01    ...........
000000E7   03 12 00 22 11 32 04 13 21 31 42    ...".2..!1B
000000F2   41 51 52 61 62 71 F0 05 23 72 81    AQRabq..#r.
000000FD   91 A1 B1 C1 D1 14 33 82 92 E1 F1    ......3....
00000108   06 43 A2 B2 C2 24 53 D2 63 F2 F2    .C...$S.c..
```


Activities
Firefox Web Browser
Thu 22:41
Text to Hex | Convert Tex
https://www.duplichecker.com/text-to-hex.php
- free and polish
Try Now

Text To HEX

ChristianBale

43687269737469616e42616c65


CrypTool 1.4.41 - FIVE.jpg
File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv. Procedures Analysis Options Window Help
FIVE.jpg

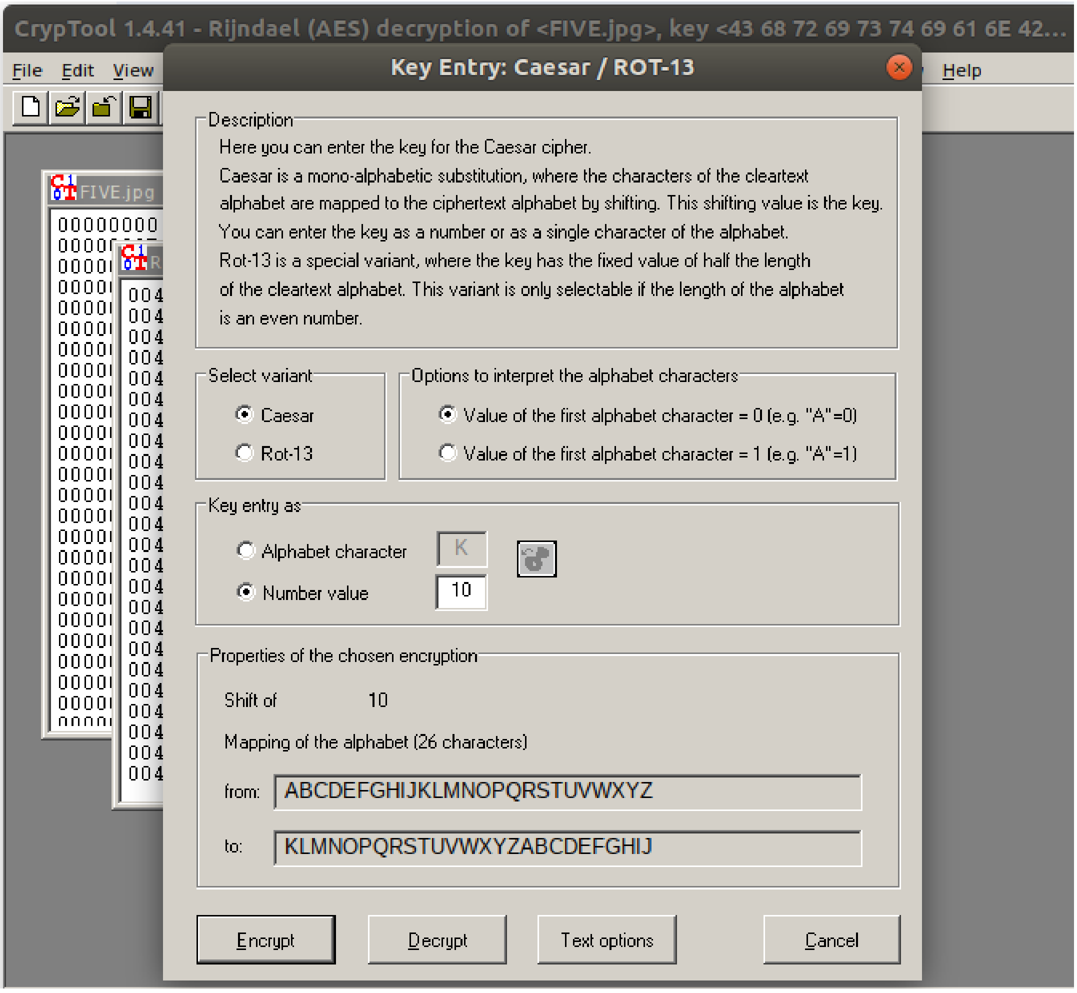
```
00000000   FF D8 FF E0 00 10 4A 46 49 46 00    ......JFIF.
0000000B   01 01 01 02 58 02 58 00 00 FF DB    ....X.X....
00000016   00 43 00 06 04 05 06 05 04 06 06    .C.........
00000021   05 06 07 07 06 08 0A 10 0A 0A 09    ...........
0000002C   09 0A 14 0E 0F 0C 10 17 14 18 18    ...........
00000037   17 14 16 16 1A 1D 25 1F 1A 1B 23    ......%...#
00000042
0000004D
00000058
00000063
0000006E
00000079
00000084
0000008F
0000009A
000000A5
000000B0
000000BB
000000C6
000000D1
000000DC
000000E7
000000F2   41 51 52 61 62 71 F0 05 23 72 81    AQRabq..#r.
000000FD   91 A1 B1 C1 D1 14 33 82 92 E1 F1    ......3....
00000108   06 43 A2 B2 C2 24 53 D2 63 F2 F2    .C...$S.c..
```

Key Entry: Rijndael (AES)

Enter the key using hexadecimal characters (0..9, A..F).

Key length: 128 bits

43 68 72 69 73 74 69 61 6E 42 61 6C 65 00 00 00



CrypTool 1.4.41 - Rijndael (AES) decryption of <FIVE.jpg>, key <43 68 72 69 73 74 69 61 6E 42...

Key Entry: Caesar / ROT-13

Description

Here you can enter the key for the Caesar cipher.
Caesar is a mono-alphabetic substitution, where the characters of the cleartext alphabet are mapped to the ciphertext alphabet by shifting. This shifting value is the key.
You can enter the key as a number or as a single character of the alphabet.
Rot-13 is a special variant, where the key has the fixed value of half the length of the cleartext alphabet. This variant is only selectable if the length of the alphabet is an even number.

Select variant
● Caesar
○ Rot-13

Options to interpret the alphabet characters
● Value of the first alphabet character = 0 (e.g. "A"=0)
○ Value of the first alphabet character = 1 (e.g. "A"=1)

Key entry as
○ Alphabet character: K
● Number value: 10

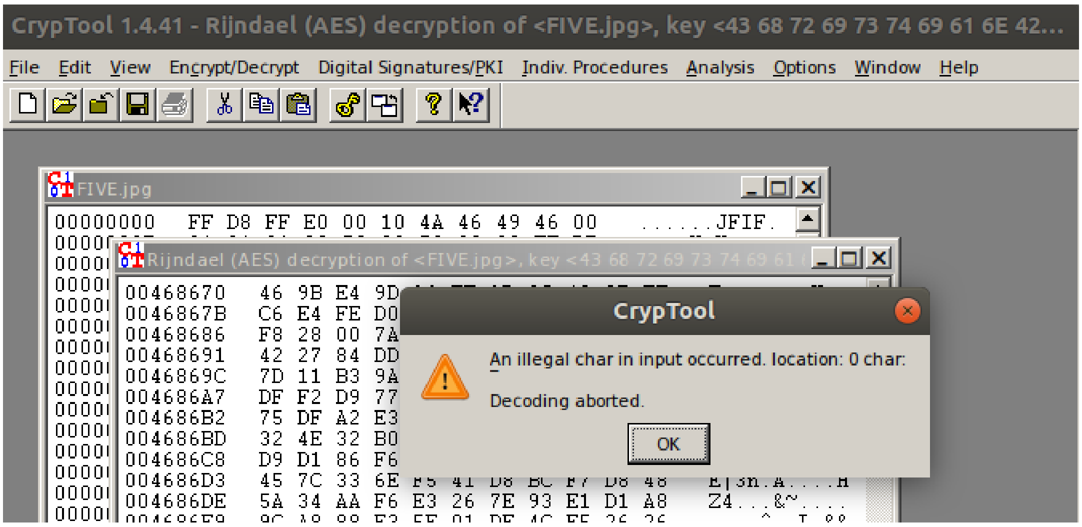
Properties of the chosen encryption

Shift of 10

Mapping of the alphabet (26 characters)

from: ABCDEFGHIJKLMNOPQRSTUVWXYZ

to: KLMNOPQRSTUVWXYZABCDEFGHIJ



CrypTool 1.4.41 - Rijndael (AES) decryption of <FIVE.jpg>, key <43 68 72 69 73 74 69 61 6E 42...

```
00000000   FF D8 FF E0 00 10 4A 46 49 46 00    ......JFIF.
```

Rijndael (AES) decryption of <FIVE.jpg>, key <43 68 72 69 73 74 69 61

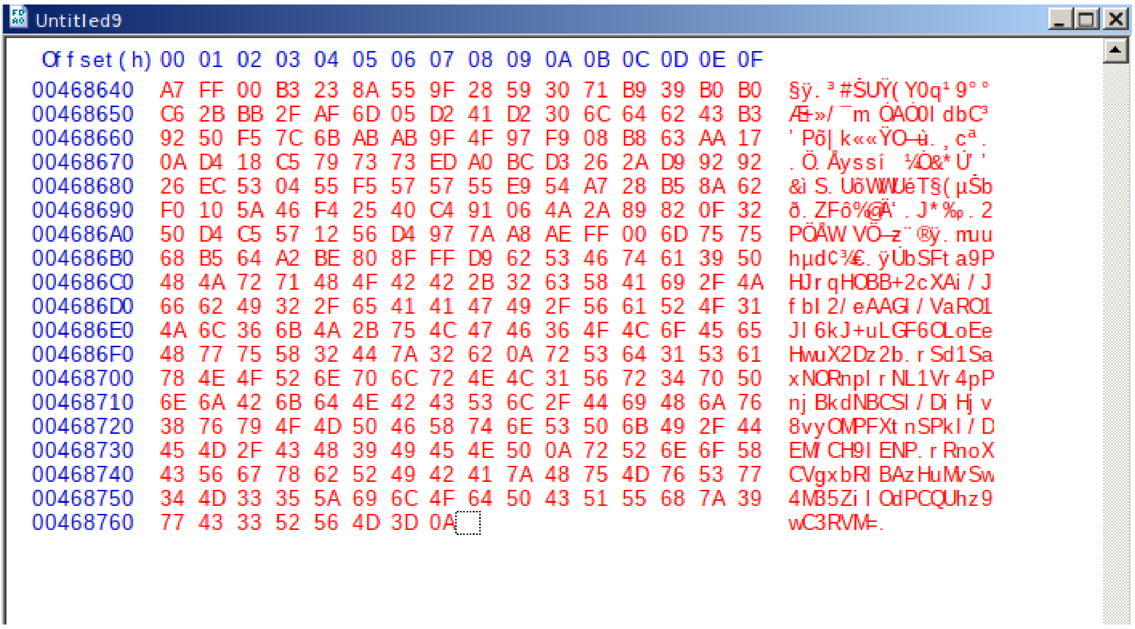
```
00468670   46 9B E4 9D
0046867B   C6 E4 FE D0
00468686   F8 28 00 7A
00468691   42 27 84 DD
0046869C   7D 11 B3 9A
004686A7   DF F2 D9 77
004686B2   75 DF A2 E3
004686BD   32 4E 32 B0
004686C8   D9 D1 86 F6
004686D3   45 7C 33 6E                                E|3n.A....H
004686DE   5A 34 AA F6 E3 26 7E 93 E1 D1 A8    Z4...&~....
```

CrypTool

An illegal char in input occurred. location: 0 char:

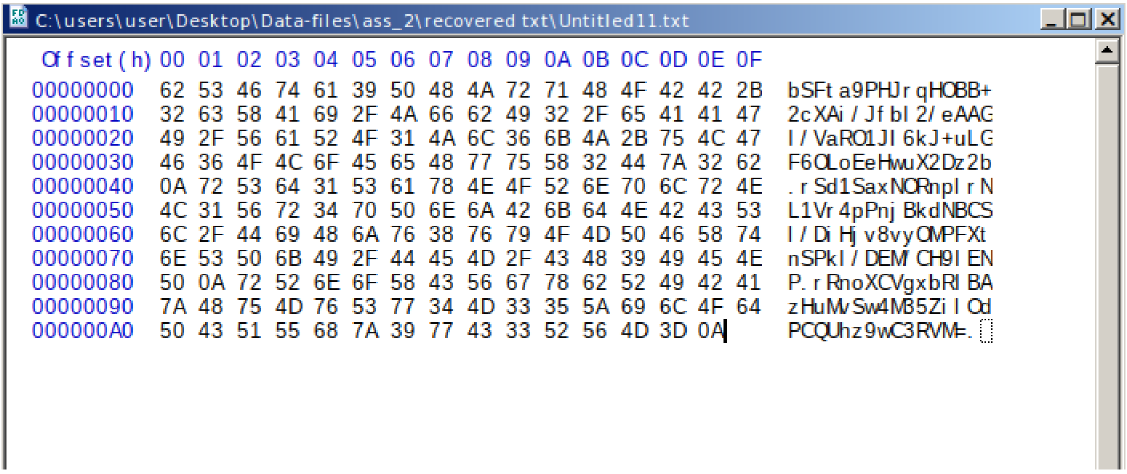
Decoding aborted.

So, to resolve it, I used the Hxd hex editor to analyse the FIVE.jpg. In that I separated the last part of the hex values, because they were a bit odd than others. That was without any jumbled words or symbols. So, I cut and pasted it as a new hex document.



Untitled9

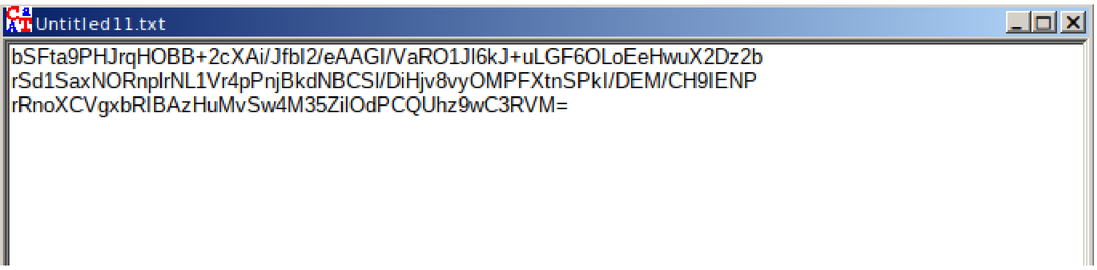
```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00468640  A7 FF 00 B3 23 8A 55 9F 28 59 30 71 B9 39 B0 B0  §ÿ.³#ŠUŸ(Y0q¹9°°
00468650  C6 2B BB 2F AF 6D 05 D2 41 D2 30 6C 64 62 43 B3  Æ+»/¯m.ÒAÒ0ldbC³
00468660  92 50 F5 7C 6B AB AB 9F 4F 97 F9 08 B8 63 AA 17  'Põ|k««ŸO—ù.¸cª.
00468670  0A D4 18 C5 79 73 73 ED A0 BC D3 26 2A D9 92 92  .Ô.Åyssí ¼Ó&*Ù''
00468680  26 EC 53 04 55 F5 57 57 55 E9 54 A7 28 B5 8A 62  &ìS.UõWWUéT§(µŠb
00468690  F0 10 5A 46 F4 25 40 C4 91 06 4A 2A 89 82 0F 32  ð.ZFô%@Ä'.J*‰‚.2
004686A0  50 D4 C5 57 12 56 D4 97 7A A8 AE FF 00 6D 75 75  PÔÅW.VÔ—z¨®ÿ.muu
004686B0  68 B5 64 A2 BE 80 8F FF D9 62 53 46 74 61 39 50  hµd¢¾€.ÿÙbSFta9P
004686C0  48 4A 72 71 48 4F 42 42 2B 32 63 58 41 69 2F 4A  HJrqHOBB+2cXAi/J
004686D0  66 62 49 32 2F 65 41 41 47 49 2F 56 61 52 4F 31  fbI2/eAAGI/VaRO1
004686E0  4A 6C 36 6B 4A 2B 75 4C 47 46 36 4F 4C 6F 45 65  Jl6kJ+uLGF6OLoEe
004686F0  48 77 75 58 32 44 7A 32 62 0A 72 53 64 31 53 61  HwuX2Dz2b.rSd1Sa
00468700  78 4E 4F 52 6E 70 6C 72 4E 4C 31 56 72 34 70 50  xNORnplrNL1Vr4pP
00468710  6E 6A 42 6B 64 4E 42 43 53 6C 2F 44 69 48 6A 76  njBkdNBCSl/DiHjv
00468720  38 76 79 4F 4D 50 46 58 74 6E 53 50 6B 49 2F 44  8vyOMPFXtnSPkI/D
00468730  45 4D 2F 43 48 39 49 45 4E 50 0A 72 52 6E 6F 58  EM/CH9IENP.rRnoX
00468740  43 56 67 78 62 52 49 42 41 7A 48 75 4D 76 53 77  CVgxbRIBAzHuMvSw
00468750  34 4D 33 35 5A 69 6C 4F 64 50 43 51 55 68 7A 39  4M35ZilOdPCQUhz9
00468760  77 43 33 52 56 4D 3D 0A                          wC3RVM=.
```



C:\users\user\Desktop\Data-files\ass_2\recovered txt\Untitled11.txt

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  62 53 46 74 61 39 50 48 4A 72 71 48 4F 42 42 2B  bSFta9PHJrqHOBB+
00000010  32 63 58 41 69 2F 4A 66 62 49 32 2F 65 41 41 47  2cXAi/JfbI2/eAAG
00000020  49 2F 56 61 52 4F 31 4A 6C 36 6B 4A 2B 75 4C 47  I/VaRO1Jl6kJ+uLG
00000030  46 36 4F 4C 6F 45 65 48 77 75 58 32 44 7A 32 62  F6OLoEeHwuX2Dz2b
00000040  0A 72 53 64 31 53 61 78 4E 4F 52 6E 70 6C 72 4E  .rSd1SaxNORnplrN
00000050  4C 31 56 72 34 70 50 6E 6A 42 6B 64 4E 42 43 53  L1Vr4pPnjBkdNBCS
00000060  6C 2F 44 69 48 6A 76 38 76 79 4F 4D 50 46 58 74  l/DiHjv8vyOMPFXt
00000070  6E 53 50 6B 49 2F 44 45 4D 2F 43 48 39 49 45 4E  nSPkI/DEM/CH9IEN
00000080  50 0A 72 52 6E 6F 58 43 56 67 78 62 52 49 42 41  P.rRnoXCVgxbRIBA
00000090  7A 48 75 4D 76 53 77 34 4D 33 35 5A 69 6C 4F 64  zHuMvSw4M35ZilOd
000000A0  50 43 51 55 68 7A 39 77 43 33 52 56 4D 3D 0A     PCQUhz9wC3RVM=.
```

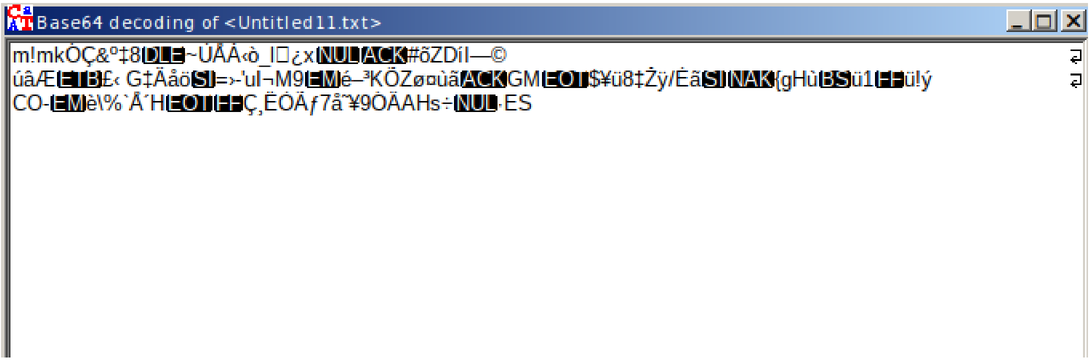
Then I saved it as a new text file loaded it in the CrypTool again and was able to observe encrypted words.



Untitled11.txt

```
bSFta9PHJrqHOBB+2cXAi/JfbI2/eAAGI/VaRO1Jl6kJ+uLGF6OLoEeHwuX2Dz2b
rSd1SaxNORnplrNL1Vr4pPnjBkdNBCSl/DiHjv8vyOMPFXtnSPkI/DEM/CH9IENP
rRnoXCVgxbRIBAzHuMvSw4M35ZilOdPCQUhz9wC3RVM=
```

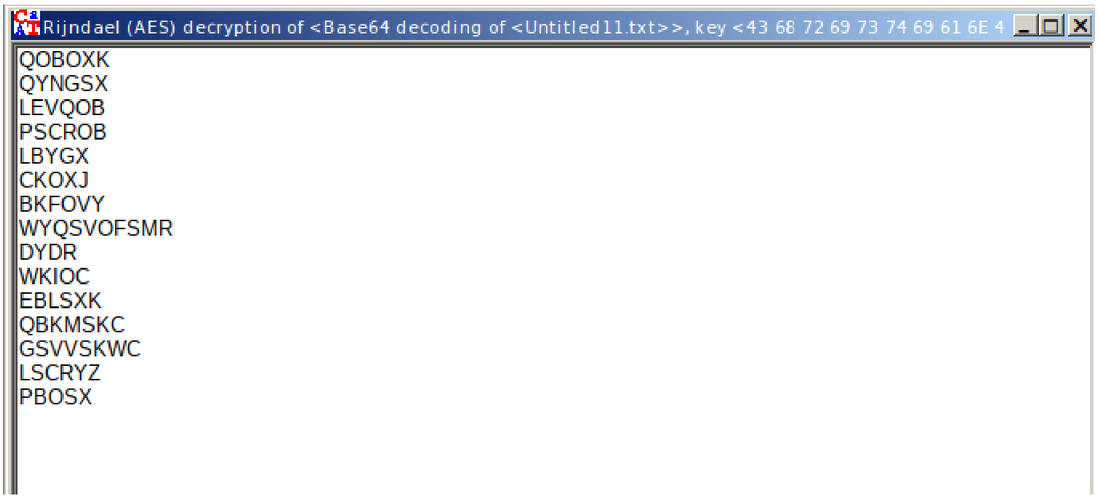
Now I first applied the Base64 decoding in it.



Base64 decoding of <Untitled11.txt>

```
m!mkÒÇ&º‡8DLE~ÙÅÀ‹ò_I□¿xNULACK#õZDíl—©
úâ/ÆETB£‹ G‡Äåö SI=›-'ul¬M9EMé–³KÖZø¤ùãACKGMEOT$¥ü8‡Żÿ/ĖãSINAK{gHùBSü1FFü!ý
CO-EMè\%`Å´HEOTFFÇ¸ËÒÄƒ7å˜¥9ÒÄAHs÷NUL·ES
```

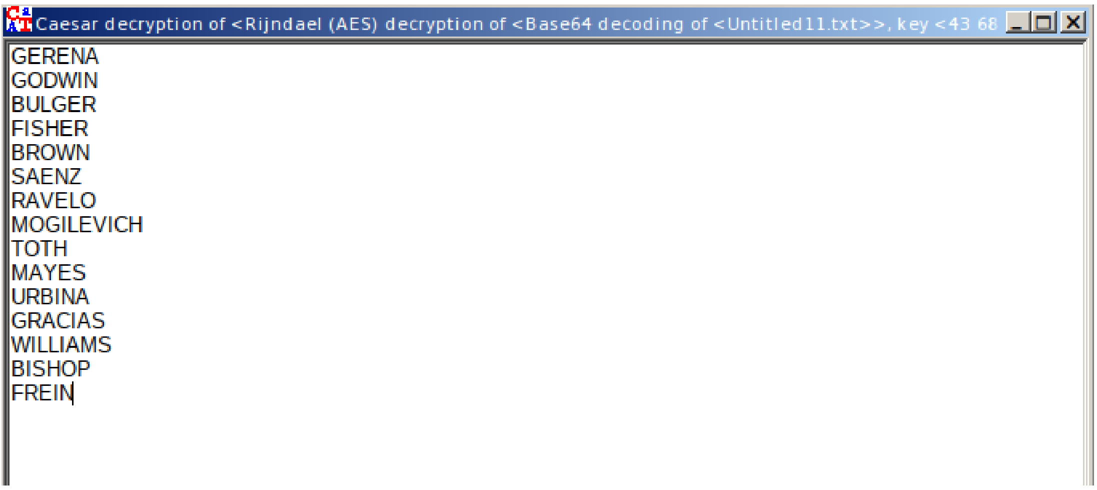
Then, I decrypted it using the AES decryption, the key I used was the hex value that I converted from ChristianBale. Then I was able to see some words but still a bit jumbled



Rijndael (AES) decryption of <Base64 decoding of <Untitled11.txt>>, key <43 68 72 69 73 74 69 61 6E 4

```
QOBOXK
QYNGSX
LEVQOB
PSCROB
LBYGX
CKOXJ
BKFOVY
WYQSVOFSMR
DYDR
WKIOC
EBLSXK
QBKMSKC
GSVVSKWC
LSCRYZ
PBOSX
```

Now to decrypt this, I used the Caeser cipher decryption, with some random keys. Finally, I was able to get a decrypted set of words that were mentioned in the How.txt. these words look like the names of the gang members that involved in the drug manufacturing.



Caesar decryption of <Rijndael (AES) decryption of <Base64 decoding of <Untitled11.txt>>, key <43 68

```
GERENA
GODWIN
BULGER
FISHER
BROWN
SAENZ
RAVELO
MOGILEVICH
TOTH
MAYES
URBINA
GRACIAS
WILLIAMS
BISHOP
FREIN
```

So, this is the end of my investigation procedure.

DIGITAL FORENSIC REPORT

7. Write a two-page report for Sandra listing your findings and recommendations. Make appropriate suggestions on how a further investigation should proceed. Construct and complete a single-item evidence form as part of your report.

Findings and Recommendations

Case summary:

Five picture files were discovered after downloading and extracting a password-protected ZIP file that was discovered during the examination of a possible drug manufacturing activity. With the use of several steganographic technologies, every image was examined for hidden content. These are the investigation's conclusions and suggestions.

Summary of steps:

1. *Downloaded the ZIP file and verified its integrity using the MD5 hash.*
2. *Decrypted NTLM hash values to attempt accessing encrypted content.*
3. *Brute-forced the zip file to extract the image files.*
4. *Analysed each image file using appropriate steganographic tools:*
    - *ONE.bmp: analysed with S-Tools*
    - *TWO.jpg: analysed with jpseek*
    - *THREE.jpg: analysed with Foremost*
    - *FOUR.png: analysed with OpenPuff*
    - *FIVE.jpg: analysed with Cryptool and HxD.*

5. *Description of recovered items:*
    - *ONE.bmp: contained a hidden text file, How.txt providing clues for further analysis.*
    - *TWO.jpg: contained a password list hidden using NTLM hash password.*
    - *THREE.jpg: contained OpenPuff configuration files essential for further analysis.*
    - *FOUR.png: contained a text file, where.txt with a list of numbers, suspected to be phone number*
    - *FIVE.jpg: contained a list of names, decrypted using CrypTool and HxD, likely belonging to gang members involved in the drug operation.*

6. *Recommendations for further investigation:*
    1. *Further analysis of recovered data:*
        - *Verify the authenticity and relevance of the phone number and name recovered.*
        - *Cross-reference the names and numbers with existing criminal databases.*
    2. *Malware analysis:*
        - *Upload the executable file to virustotal to check for known malware signatures.*
        - *Conduct a detailed analysis of the executable to understand its function and potential threats.*

3. *Extended steganographic analysis:*
    - *Use additional steganographic tools to ensure no hidden content is missed.*
    - *Perform a deeper analysis on the extracted images for any overlooked data.*
4. *Collaboration with law enforcement:*
    - *Share findings with law enforcement agencies for a coordinated effort in dismantling the drug operation.*
    - *Utilize law enforcement resources for tracking and apprehending suspects based on the recovered data.*

Evidence Form *(Figure 1-11 of the text)*

The hazardous materials team This form is to be used for only one piece of evidence. Fill out a separate form for each piece of evidence.			
Case No:	Case_2024_00002	Unit Number:	Unit_0002
Investigator:	Nirosh Ravindran		
Nature of Case:	Suspected Drug Manufacturing		
Location where evidence was obtained:	warehouse behind Roma St station in Brisbane		
Item # ID	Description of evidence	Vendor Name	Model No/Serial No.
ID_0002_0001	CD drives	Unknown	Unknown
Evidence Recovered by:	Moti	Date & Time:	10/03 @ 3.17 am
Evidence Placed in Locker:	Moti	Date & Time	10/03 @ 4.00 am
Evidence Processed by	Description of Evidence		Date & Time
Nirosh Ravindran	Downloaded a password protected zip file, 2019A02.zip from a secure link		10/03 @ 7.00 am
	Decrypted the passwords from the NTLM hashes given		10/03 @ 7.15 am
	Extracted the contents in them which were 5 image files		10/03 @ 8.00 am
	Started investigating them		10/03 @ 8.15 am
	Finished the investigation		10/03 @ 6.19 pm
	Sent the Digital Forensic report to Sandra		10/03 @ 8.00 pm
			Page _1_ of _1_

-end of template -