SIT282 Computer Crime and Digital Forensics
ASSIGNMENT 2

Digital Forensic Investigation #2

Written and completed by Connor Gent - 219501701

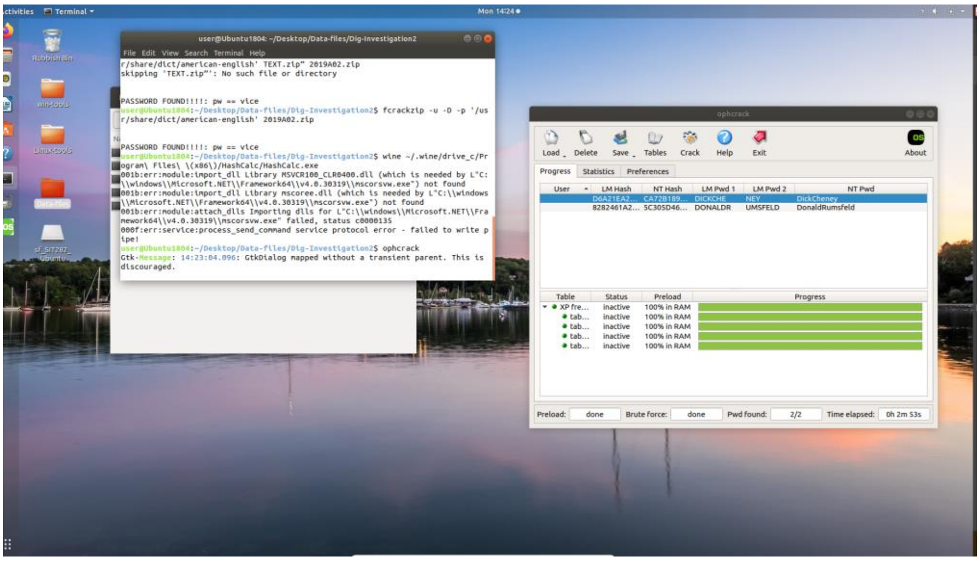
Table of Contents

DIGITAL FORENSIC PROCEDURE

1. How I downloaded the file, precautions taken, and how I ensured its integrity.

File Download Procedure	In order to download the executable files that have been provided to us by Sandra I used an FTP program. I could have also used a Web-browser (Firefox on VM) but that contains more risks. Once Downloaded I calculated the hash value using md5sum syntax to compare it with the given value in the case "9ec1c8f62429182349f3979c39aed8fb"
Precautions Applied	Precautions used included making a copy of the executable code, whilst also changing the permission of the file meaning I could only access and view the file just minimizing the chances of the file being altered and manipulated by unauthorised individuals.
Method used to ensure Integrity	I copied the executable file and locked the original file. A duplicate was made immediately after the download to have a backup copy and to minimize other risks. CHMOD 400 was used on the original file to protect the file against accidental overwrite. I also checked to see if there were any viruses or other malicious codes were within the file. This was done as it was a binary file. These steps/methods helped ensure integrity.

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



```
r/share/dict/american-english' TEXT.zip" 2019A02.zip
skipping 'TEXT.zip"': No such file or directory


PASSWORD FOUND!!!!: pw == vice
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ fcrackzip -u -D -p '/us
r/share/dict/american-english' 2019A02.zip


PASSWORD FOUND!!!!: pw == vice
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ wine ~/.wine/drive_c/Pr
ogram\ Files\ \(x86\)/HashCalc/HashCalc.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!
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ ophcrack
Gtk-Message: 14:23:04.096: GtkDialog mapped without a transient parent. This is
discouraged.
```

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	
tab...	inactive	100% in RAM	
tab...	inactive	100% in RAM	
tab...	inactive	100% in RAM	
tab...	inactive	100% in RAM	

Preload: done Brute force: done Pwd found: 2/2 Time elapsed: 0h 2m 53s

(Figure 0.1)

I used the program OphCrack to obtain the passwords for the two hash values, the steps required to decrypt/Crack the two given NTML hash values were:

- Clicked the "load" button then selected "Single Hash" from the list that dropped down.
- I then typed in the NTML hashes separately in the load single hash box:

  - D6A21EA26063C42FC9876E4B0C51BC82:CA72B189F412A384D96B785A08176773
  - 8282461A2BDAF626E6067B973FDDC643:5C305D4616C7571D5DDC6EEA5BA5C395

- Once both NTML hash values were typed I clicked the ‘crack’ button from the main menu
- I waited a few minutes as I wanted to fully load the rainbow tables in the memory. By waiting I was able to see the recovered passwords of these two NTML hash values.

NTML hash values were:

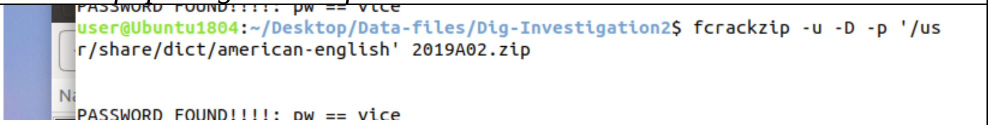
D6A21EA26063C42FC9876E4B0C51BC82:CA72B189F412A384D96B785A0817677

- Corresponded to alphabetical phrase: DickCheney (Password Acquired)

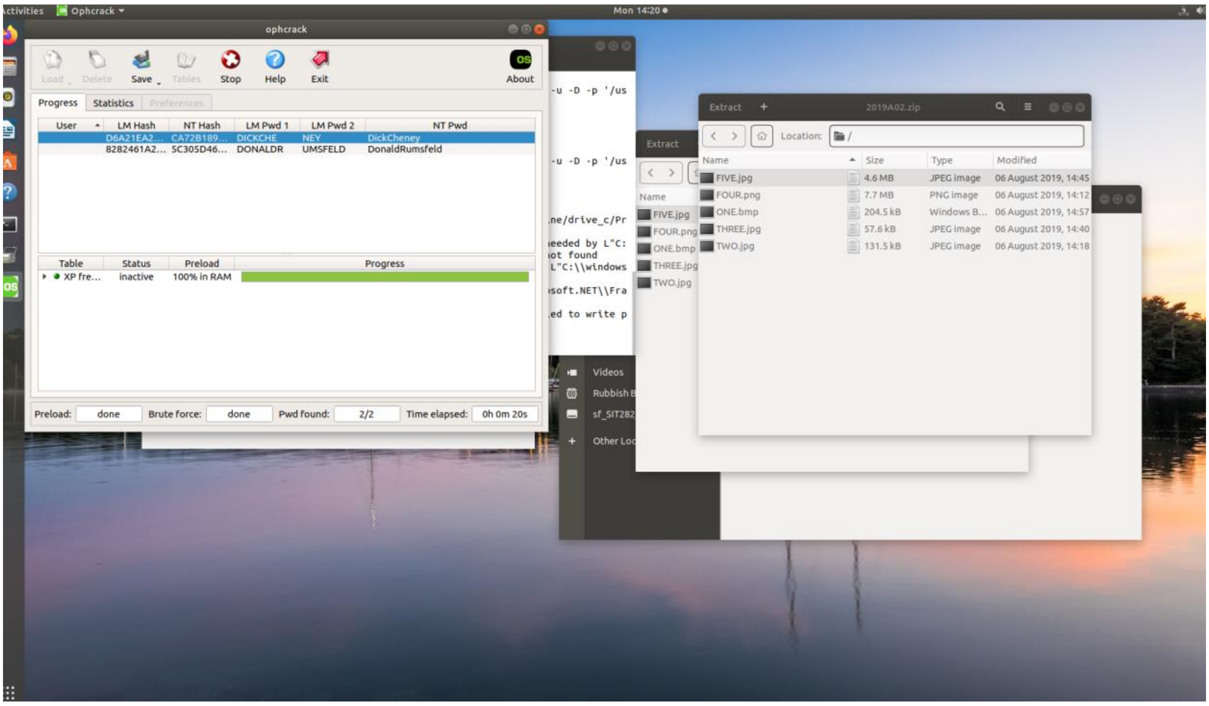
8282461A2BDAF626E6067B973FDDC643:5C305D4616C7571D5DDC6EEA5BA5C395

- Corresponded to Alphabetical phrase: DonaldRumsfield (Password Acquired)

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. During the investigation, I found that a password protected the ZIP file. To gain access to the file I used fcrazkzip tool to recover the encryption password. The NTML passwords led to the information contained in the zip file, so using a dictionary to attack the password was needed to understand where these passwords are required.*
	*2. The command I used was **frackzip – u -D -p ‘/usr/share/dict/ameican-english’ 2019AO2.zip** (Screenshot can be seen below).*
	3. After typing this command, a password was found – vice. The password had to be in the dictionary as the fcrazkzip was able to crack the password in quick succession. The information obtained in step 2 did not have a direct relation to the process of opening the ZIP file.
	4.  PASSWORD FOUND!!!!: pw == vice user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ fcrackzip -u -D -p '/usr/share/dict/american-english' 2019A02.zip PASSWORD FOUND!!!!: pw == vice *(Figure 0.2)*

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.



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 20s

```
-u -D -p '/us

-u -D -p '/us

.ne/drive_c/Pr
ieeded by L"C:
iot found
L"C:\\windows
isoft.NET\\Fra
.ed to write p
```

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	57.6 kB	JPEG image	06 August 2019, 14:40
TWO.jpg	131.5 kB	JPEG image	06 August 2019, 14:18

(Figure 0.3)

Content description	The encrypted file contained five images that varied in image types (BMP, JPG, PNG) and image size. The images within the file were encrypted with a password. The password 'vice' was required to access these images. They were last modified on August 6th of 2019. American politicians and a picture of army soldiers made up the images (Presumably US army troops in a war-torn country) and were all visible once these images were unlocked. The politician images featured Donald Rumsfeld, and Dick Cheney. It's clear there is more to the images as passwords found in step 2 were these individuals first and last names.

File Name	*File Type*	*MD5 Hash Value*
ONE.bmp	*Bitmap image*	*85a37e5db9bfd515de3d81e1e008aeb8*
Two.jpg	*JPEG Image*	*Ea2d39cc9bba7d2d803c97d44660f3ba*
THREE.jpg	*JPEG Image*	*E57ccee729362cf6749fc49f830faca5*
FOUR.png	*PNG image*	*9daf259b5250fe57e76a6ab79b1aead*
FIVE.jpg	*JPEG image*	*B1fdbbcf337d6098436831ea660ac03a*

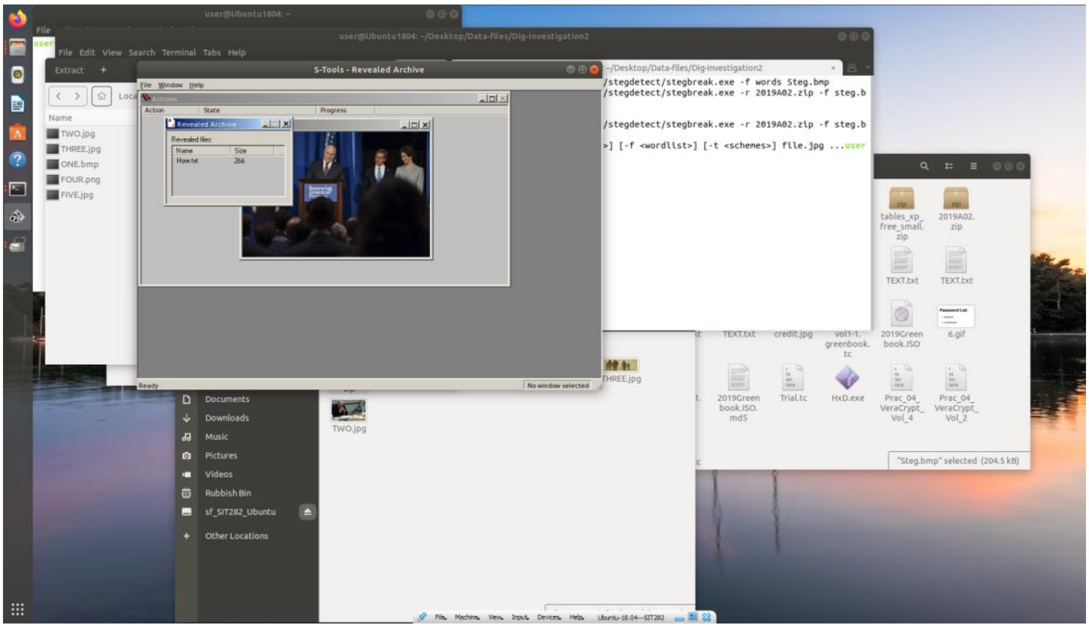
5. What tools will you now use to proceed your investigation and why?

Tool	*Reason*
Stegbreak	JPG can contain hidden information; I suspect hidden data in the JPEG files. Therefore I will use Stegbreak to locate and find the hidden data in the JPEG images.
Jpseek	Jpseek will be used to determine whether there are hidden contents concealed in other files found in the zip file.
S-Tools	S-tools are a GUI-based stegorgraphy tool. I will be using this tool in the

	investigation as it allows me to reveal contents in a graphic file
HxD	HxD is often used in digital investigation as it enables an investigator the opportunity to open any file to inspect the contents within. HxD can also be used to display and edit the memory used by running processes. Some of the main features of HxD include Data inspection, searching and replacing several data types, and calculating checksums and hashes.
Ophcrack	Ophcrack is a program that cracks windows log-in passwords by using LM hashes through rainbow tables. I will be using this tool to crack the NTML sent by Sandra.
Cryptool	Cryptool provides a bundle of Cryptgraphic tools. I will implement the decryption of Caesar Cipher which replaces each letter of the alphabet with a letter placed down or up according to the key given.

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

Steps undertaken during this investigation:


S-Tools - Revealed Archive
Revealed Archive
Revealed files
Name
Size
How.txt
266
~/Desktop/Data-files/Dig-investigation2

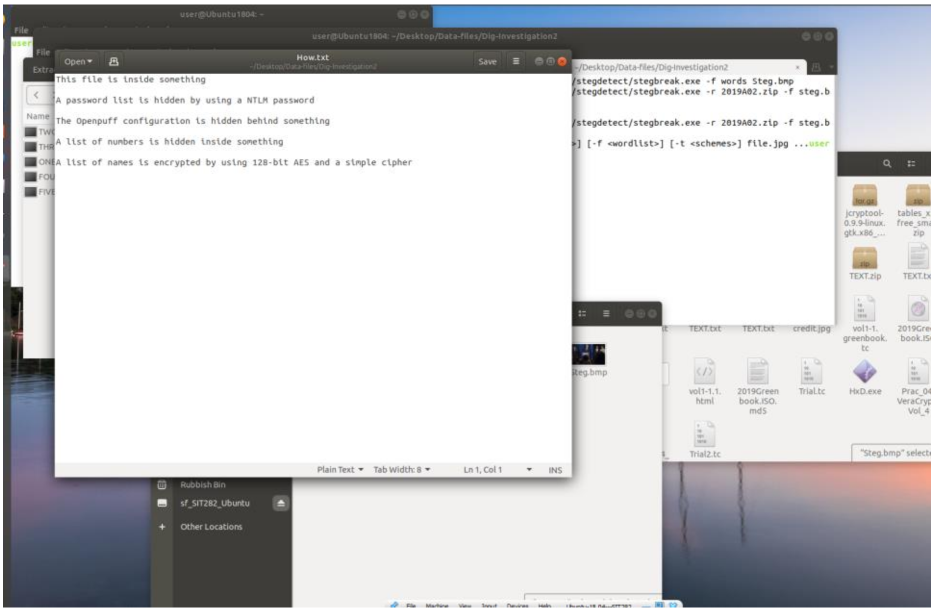
```
/stegdetect/stegbreak.exe -f words Steg.bmp
/stegdetect/stegbreak.exe -r 2019A02.zip -f steg.b

/stegdetect/stegbreak.exe -r 2019A02.zip -f steg.b

>] [-f <wordlist>] [-t <schemes>] file.jpg ...user
```

(Figure 0.4)

1: I used S-tools as I suspected there were hidden contents within these images. S-tools can reveal contents, I used the command **wine ~/Desktop/win-tools/jphide\ and \Stegbreak/S-tool/S-Tools.exe”** to open the tool. S-tools only uses BMP, GIF, and WAV file types, so I dragged the existing BMP image file into the ZIP file, ‘ONE.bmp’. I saved the image as Steg.bmp. Closing the opened file in the S-tools was done after saving

2: I then reopened the Steg.bmp file and right clicked the picture in S-tools selecting Reveal. I used the password DickCheney to recover the hidden TXT file. How.txt was saved in the ONE.bmp file, revealed with the password mentioned above.

3: Figure 0.5 shows the containments of the How.txt file, I immediately realised it was clues to find more hidden files, it helped me identify what steps I need to conduct in this forensic investigation.

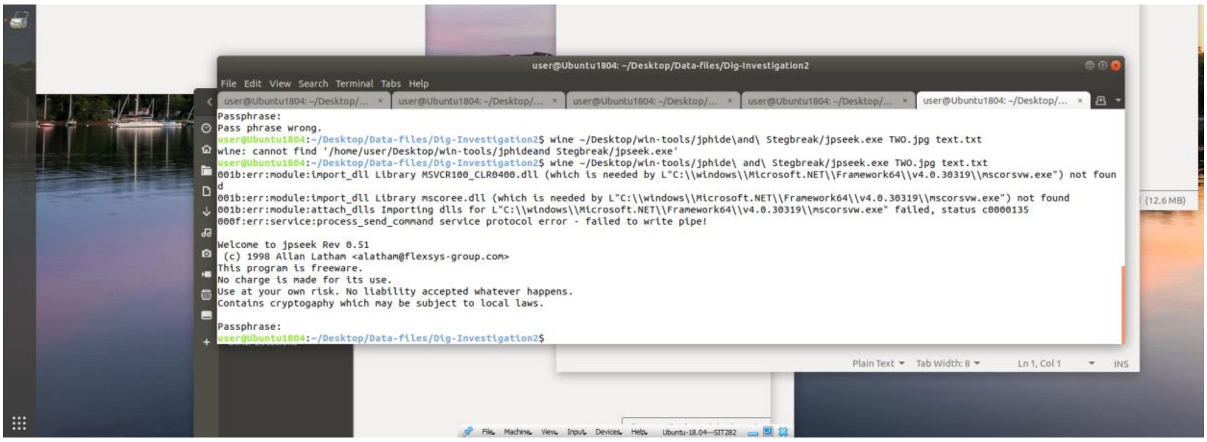


```
This file is inside something

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
```

```
/stegdetect/stegbreak.exe -f words Steg.bmp
/stegdetect/stegbreak.exe -r 2019A02.zip -f steg.b

/stegdetect/stegbreak.exe -r 2019A02.zip -f steg.b

>] [-f <wordlist>] [-t <schemes>] file.jpg ...user
```

(Figure 0.5)

4: Using the list in figure 0.5 I had the belief the tips were given in photo order. For the 2nd image file, I needed to use 'jpseek' to uncover the hidden password list. The command I used was "**wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/jpseek.exe TWO.jpg text.txt".** I was asked in the terminal for a password using the 2nd password found in step 2 (DonaldRumsfield).

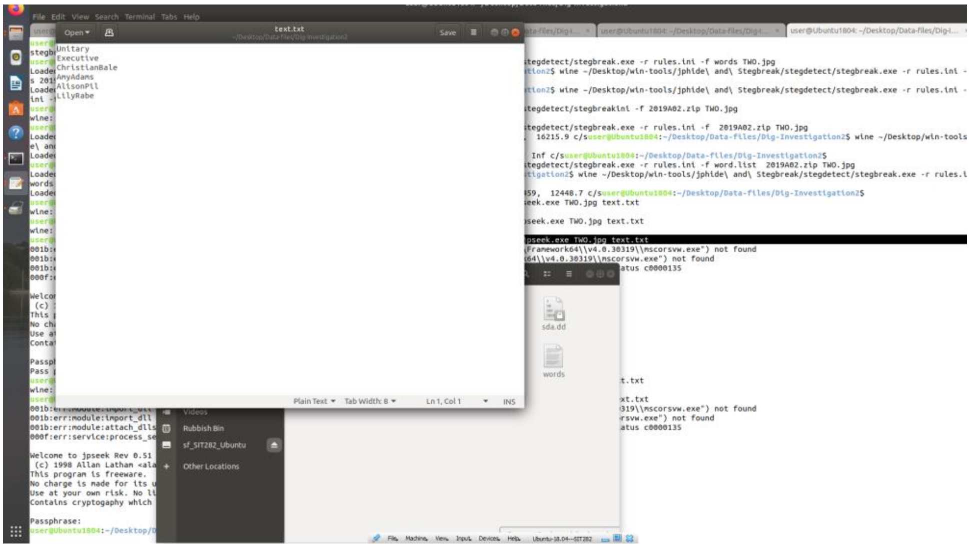


```
Passphrase:
Pass phrase wrong.
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ wine ~/Desktop/win-tools/jphide\and\ Stegbreak/jpseek.exe TWO.jpg text.txt
wine: cannot find '/home/user/Desktop/win-tools/jphideand Stegbreak/jpseek.exe'
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/jpseek.exe TWO.jpg text.txt
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 foun
d
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\\Framework64\\v4.0.30319\\mscorsvw.exe" failed, status c0000135
000f:err:service:process_send_command service protocol error - failed to write pipe!

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/Dig-Investigation2$
```

user@Ubuntu1804: ~/Desktop/Data-files/Dig-Investigation2
File Edit View Search Terminal Tabs Help
Plain Text
Tab Width: 8
Ln 1, Col 1
INS
File Machine View Input Devices Help Ubuntu-18.04--SIT282

(Figure 0.6)

5: After entering the password correctly, a new TXT was revealed, 'text.TXT', which contained a list of names that I assumed was the list of passwords (Seen in figure 0.7) hidden by the NTML password. This gave a clear indication I was on the right track. These passwords found were handy later on in my investigation.



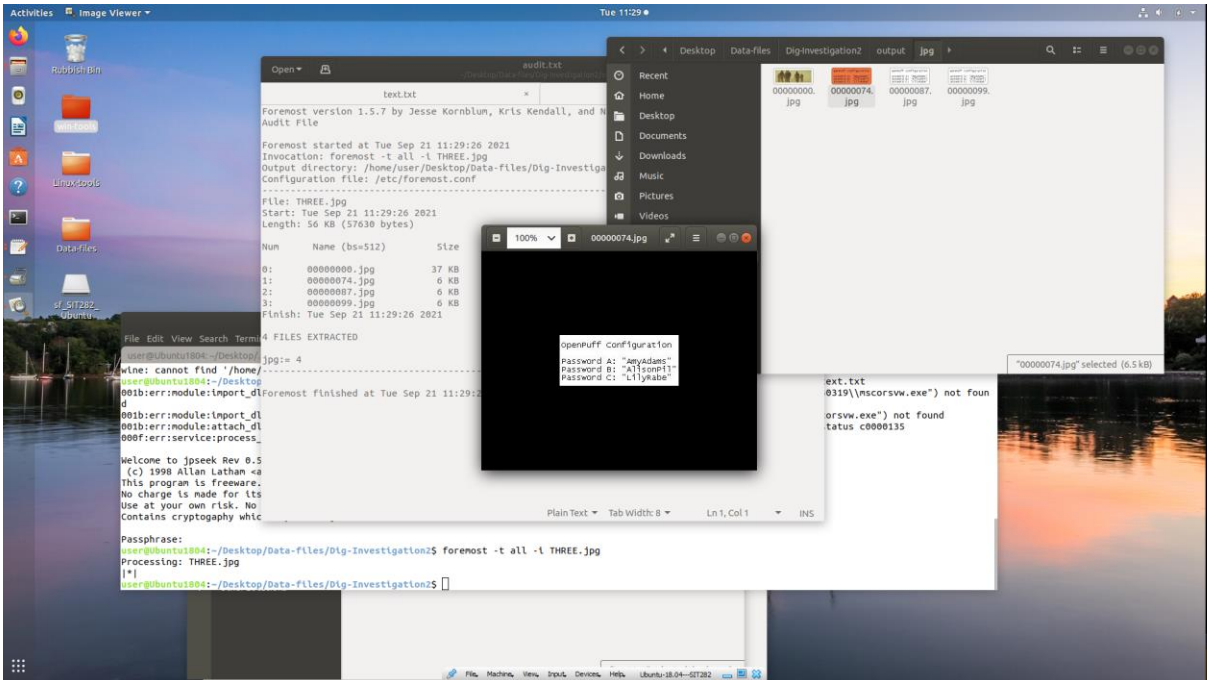
```
Unitary
Executive
ChristianBale
AmyAdams
AlisonPil
LilyRabe
```

```
tegdetect/stegbreak.exe -r rules.ini -f words TWO.jpg
tion2$ wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/stegdetect/stegbreak.exe -r rules.ini -

tion2$ wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/stegdetect/stegbreak.exe -r rules.ini -

tegdetect/stegbreakini -f 2019A02.zip TWO.jpg

tegdetect/stegbreak.exe -r rules.ini -f  2019A02.zip TWO.jpg
  16215.9 c/suser@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ wine ~/Desktop/win-tools

 Inf c/suser@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$
tegdetect/stegbreak.exe -r rules.ini -f word.list  2019A02.zip TWO.jpg
tigation2$ wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/stegdetect/stegbreak.exe -r rules.i

59,  12448.7 c/suser@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$
eek.exe TWO.jpg text.txt

seek.exe TWO.jpg text.txt

pseek.exe TWO.jpg text.txt
Framework64\\v4.0.30319\\mscorsvw.exe") not found
64\\v4.0.30319\\mscorsvw.exe") not found
atus c0000135
```

```
001b:err:module:import_dll
001b:err:module:attach_dlls
000f:err:service:process_se

Welcome to jpseek Rev 0.51
 (c) 1998 Allan Latham <ala
This program is freeware.
No charge is made for its u
Use at your own risk. No li
Contains cryptogaphy which

Passphrase:
user@Ubuntu1804:~/Desktop/D
```

(Figure 0.7)

6: I realised that I needed to obtain 3 sets of passwords to use OpenPuff's data unhiding feature, which was needed to complete the next step of the investigation. I decided to use foremost, which recovers and reconstructs files. The command **foremost -t all -i THREE.jpg** was used. A new JPG image was revealed containing three passwords required for the OpenPuff Configuration. I could now progress with the investigation once these passwords were found. The steps undertaken can be seen in figure 0.8



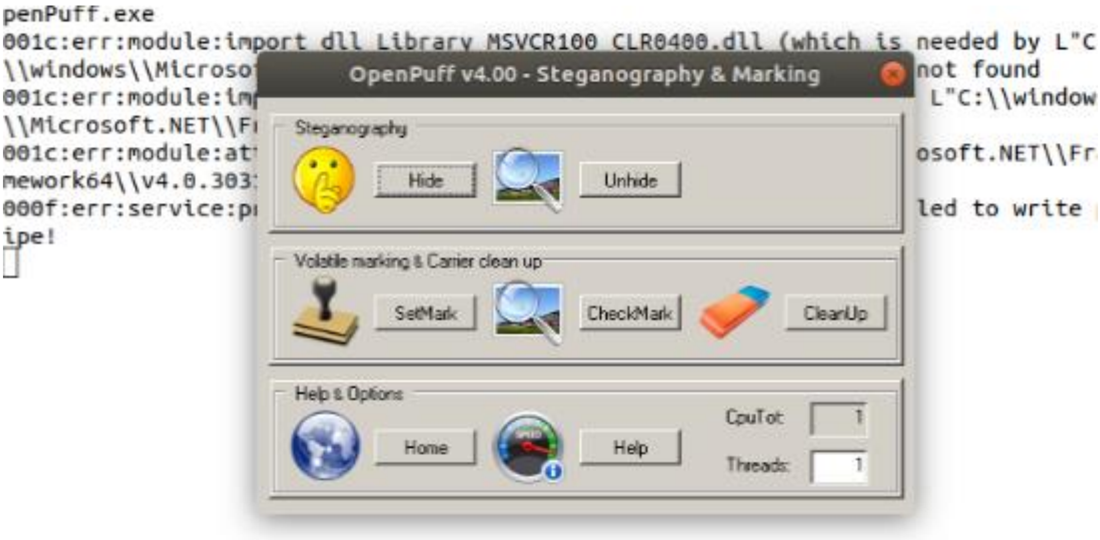
```
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and N
Audit File

Foremost started at Tue Sep 21 11:29:26 2021
Invocation: foremost -t all -i THREE.jpg
Output directory: /home/user/Desktop/Data-files/Dig-Investiga
Configuration file: /etc/foremost.conf
------------------------------------------------------------------
File: THREE.jpg
Start: Tue Sep 21 11:29:26 2021
Length: 56 KB (57630 bytes)

Num      Name (bs=512)         Size

0:       00000000.jpg          37 KB
1:       00000074.jpg           6 KB
2:       00000087.jpg           6 KB
3:       00000099.jpg           6 KB
Finish: Tue Sep 21 11:29:26 2021

4 FILES EXTRACTED

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

Foremost finished at Tue Sep 21 11:29:2
```

```
OpenPuff Configuration

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

```
Passphrase:
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ foremost -t all -i THREE.jpg
Processing: THREE.jpg
|*|
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$
```

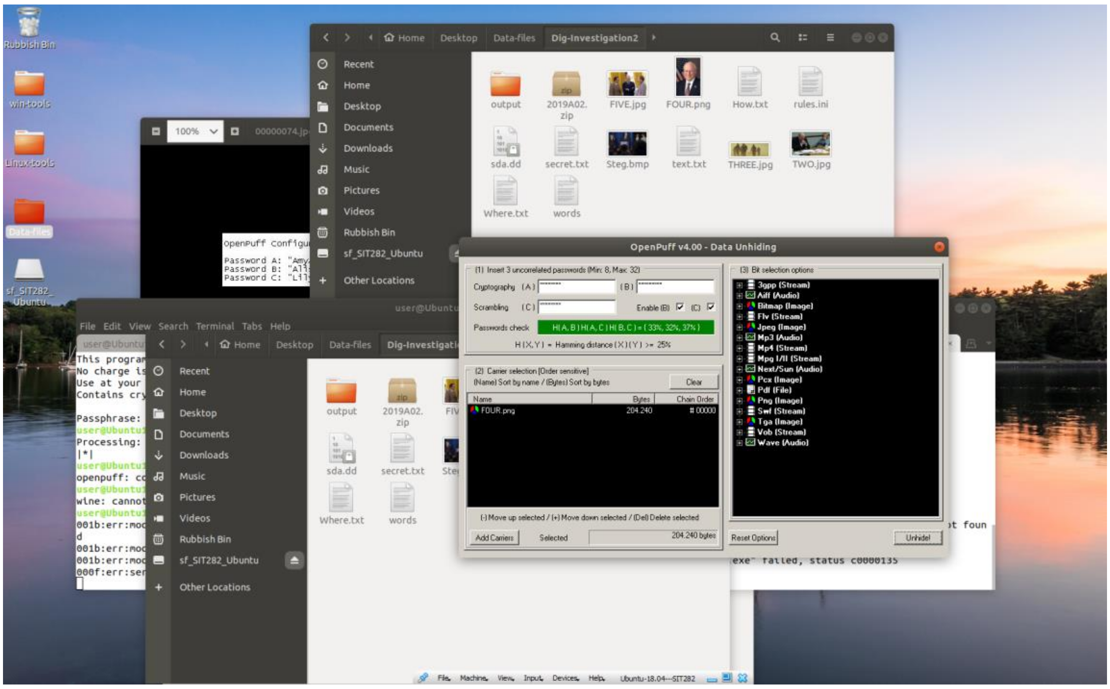
(Figure 0.8)

7: After finding these passwords, I open OpenPuff through the command **"wine ~/Desktop/win-tools/OpenPuff/OpenPuff.exe"** and selected the unhide option part of the user interface (figure 0.9 shows the user interface). Typed in the passwords seen in figure 0.8 and then utilized the OpenPuff configuration.



```
penPuff.exe
001c:err:module:import dll Library MSVCR100 CLR0400.dll (which is needed by L"C
\\windows\\Microso                                                   not found
001c:err:module:im                                                  L"C:\\window
\\Microsoft.NET\\F
001c:err:module:at                                                  osoft.NET\\Fr
mework64\\v4.0.303
000f:err:service:p                                                  led to write
ipe!
```

(Figure 0.9)



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) Sort by name / (Bytes) Sort by bytes

Name	Bytes	Chain Order
FOUR.png	204.240	# 00000

(-) Move up selected / (+) Move down selected / (Del) Delete selected

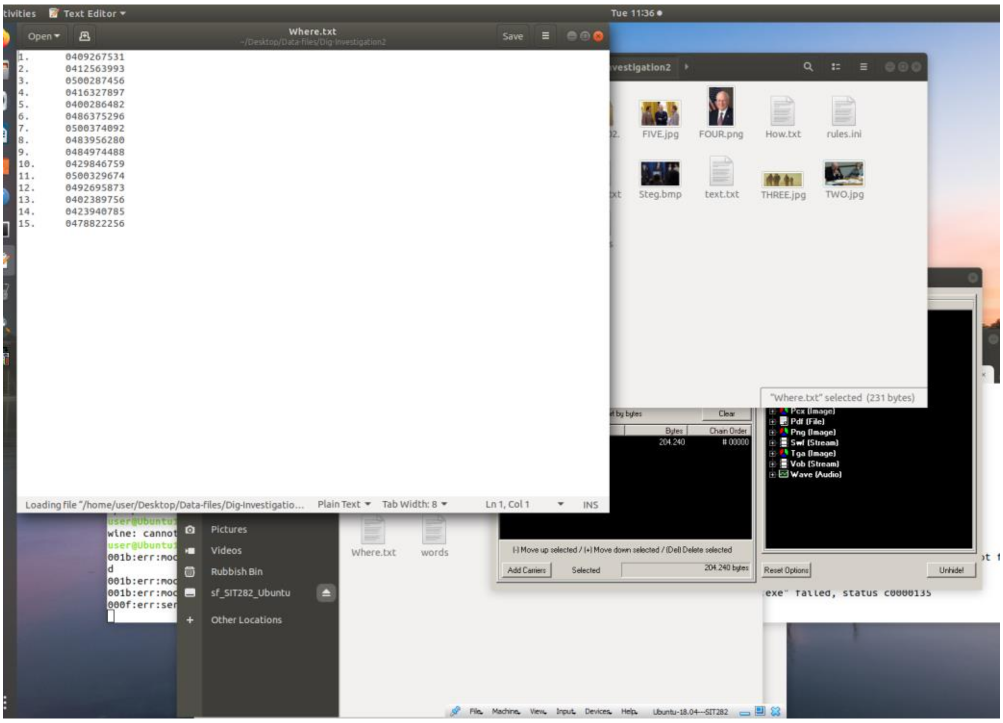
Selected 204.240 bytes

(3) Bit selection options

(Figure 1.0)

8: I placed the file FOUR.png into OpenPuff and began the unhiding configuration process to locate the list of numbers believed to be hidden within this file.

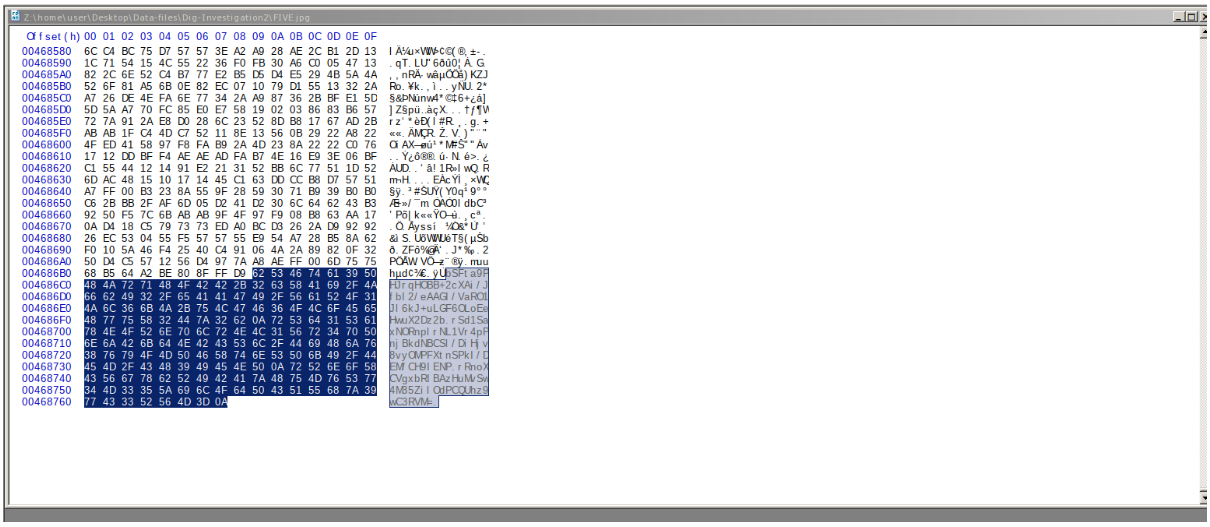
9: The process was successful, and the list of numbers (Figure 1.1) revealed in a TXT file titled Where.txt (File stored inside of FOUR.png). The list of numbers was another Key piece of information I can add to this forensics case.



```
Where.txt
~/Desktop/Data-files/Dig-Investigation2

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
```

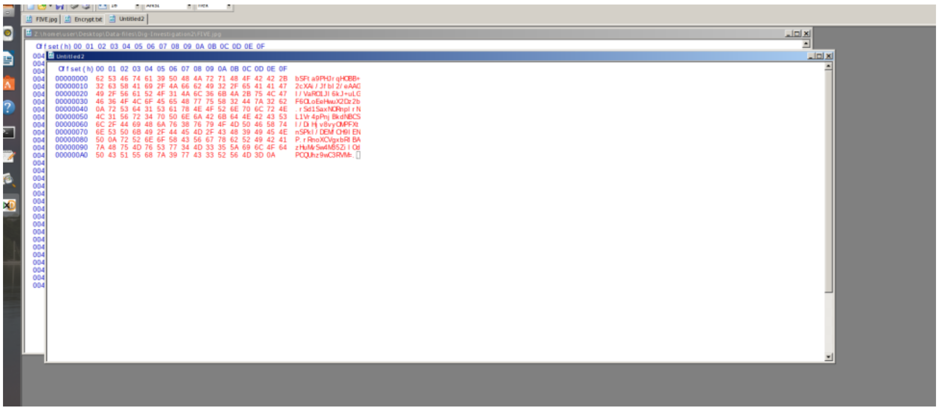
(Figure 1.1)

10: After retrieving the phone numbers seen in figure 1.1, the next task was finding the list of names in the last image file. I had to use Cryptool in conjunction with HxD programs to get the final results as data manipulation was required. I opened HxD using the code **wine ~/Desktop/win-tools/HxD.exe,** then opened FIVE.jpg in HxD to inspect the file closely. After inspecting, I found hidden contents in FIVE.jpg after its trailer. I copied this chunk of data and stored it on a new HxD page. Then saving the file as 'Encrypt.txt', I made a duplicate copy to lower the risk.



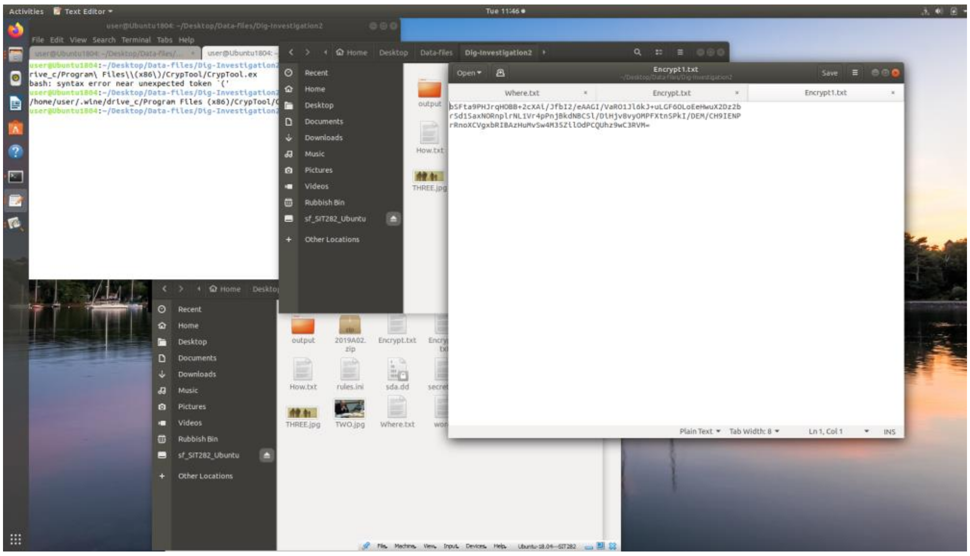
```
Z:\home\user\Desktop\Data-files\Dig-Investigation2\FIVE.jpg

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

00468580  6C C4 BC 75 D7 57 57 3E A2 A9 28 AE 2C B1 2D 13
00468590  1C 71 54 15 4C 55 22 36 F0 FB 30 A6 C0 05 47 13
004685A0  82 2C 6E 52 C4 B7 77 E2 B5 D5 D4 E5 29 4B 5A 4A
004685B0  52 6F 81 A5 6B 0E 82 EC 07 10 79 D1 55 13 32 2A
004685C0  A7 26 DE 4E FA 6E 77 34 2A A9 87 36 2B BF E1 5D
004685D0  5D 5A A7 70 FC 85 E0 E7 58 19 02 03 86 83 B6 57
004685E0  72 7A 91 2A E8 D0 28 6C 23 52 8D B8 17 67 AD 2B
004685F0  AB AB 1F C4 4D C7 52 11 8E 13 56 0B 29 22 A8 22
00468600  4F ED 41 58 97 F8 FA B9 2A 4D 23 8A 22 22 C0 76
00468610  17 12 DD BF F4 AE AE AD FA B7 4E 16 E9 3E 06 BF
00468620  C1 55 44 12 14 91 E2 21 31 52 BB 6C 77 51 1D 52
00468630  6D AC 48 15 10 17 14 45 C1 63 DD CC B8 D7 57 51
00468640  A7 FF 00 B3 23 8A 55 9F 28 59 30 71 B9 39 B0 B0
00468650  C6 2B BB 2F AF 6D 05 D2 41 D2 30 6C 64 62 43 B3
00468660  92 50 F5 7C 6B AB AB 9F 4F 97 F9 08 B8 63 AA 17
00468670  0A D4 18 C5 79 73 73 ED A0 BC D3 26 2A D9 92 92
00468680  26 EC 53 04 55 F5 57 57 55 E9 54 A7 28 B5 8A 62
00468690  F0 10 5A 46 F4 25 40 C4 91 06 4A 2A 89 82 0F 32
004686A0  50 D4 C5 57 12 56 D4 97 7A A8 AE FF 00 6D 75 75
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=.
```

(figure 1.2)



```
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=.
```

(**Figure 1.3**)

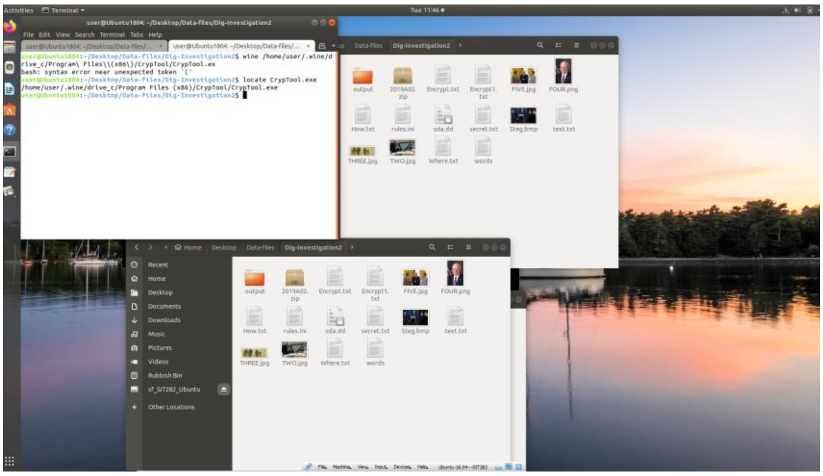


```
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2
rive_c/Program\ Files\\(x86\)/CrypTool/CrypTool.ex
bash: syntax error near unexpected token `('
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2
/home/user/.wine/drive_c/Program Files (x86)/CrypTool/C
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2
```

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

(**Figure 1.4 – Containments of Encrypt.TXT**)

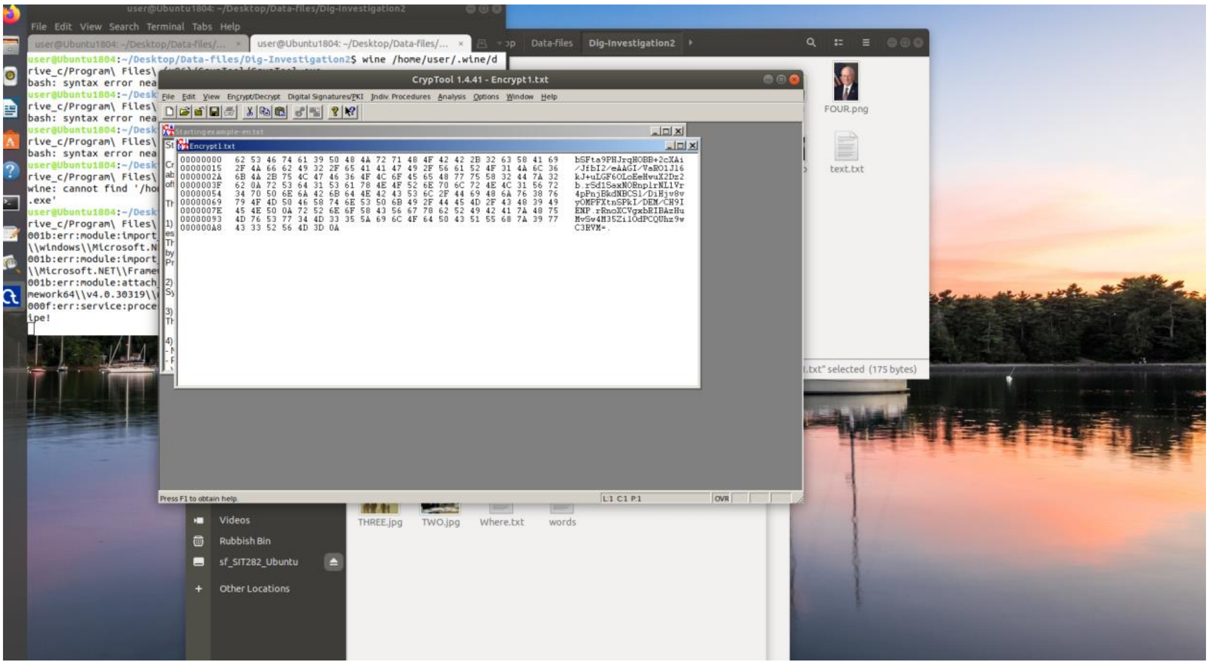
11: As the last instruction said, this file was encrypted using a simple cipher, I decided to use Cryptool to decrypt Encrypt.txt. I chose Cryptool as it is a program that can encrypt and decrypt messages using Caesar encryption. I first located Cryptool in the terminal with the command ‘**locate CrypTool.exe’.** Opening CrypTool with the command **wine /home/user/.wine/drive_c/Program\ Files\ (x86)/CrypTool/CrypTool.exe**



```
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ wine /home/user/.wine/d
rive_c/Program\ Files\\(x86\)/CrypTool/CrypTool.ex
bash: syntax error near unexpected token `('
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ locate CrypTool.exe
/home/user/.wine/drive_c/Program Files (x86)/CrypTool/CrypTool.exe
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$
```

(Figure 1.5)

12: I opened the Encrypt1.txt file in Cryptool, choosing Base 64 Decoding of the file which was an option available in Cryptool.

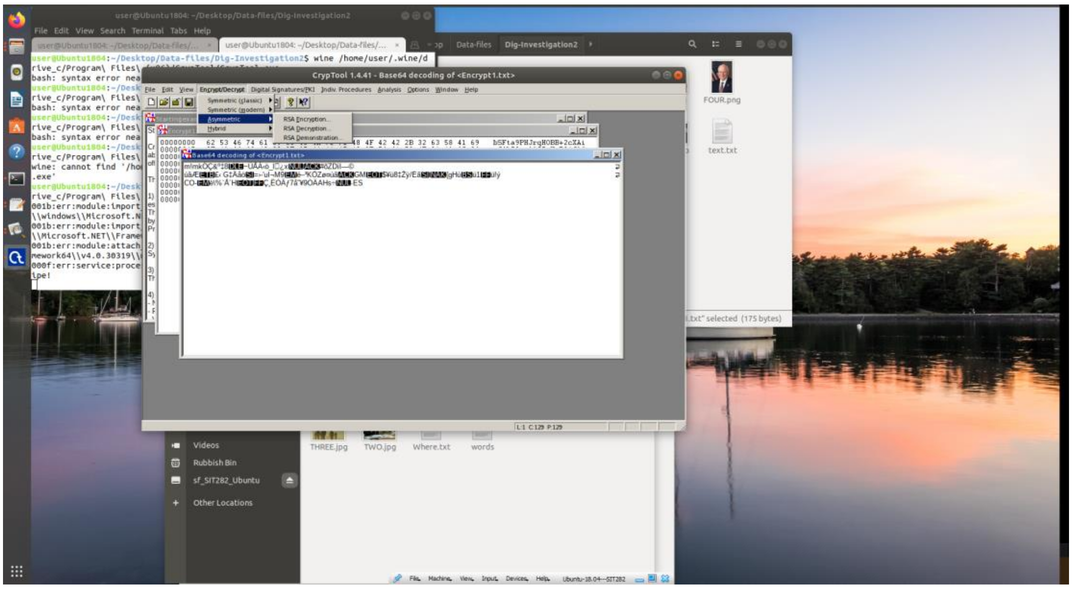


```
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ wine /home/user/.wine/d
rive_c/Program\ Files\
bash: syntax error nea
user@Ubuntu1804:~/Desk
rive_c/Program\ Files\
bash: syntax error nea
user@Ubuntu1804:~/Desk
rive_c/Program\ Files\
bash: syntax error nea
user@Ubuntu1804:~/Desk
rive_c/Program\ Files\
wine: cannot find '/ho
.exe'
user@Ubuntu1804:~/Desk
rive_c/Program\ Files\
001b:err:module:import
\\windows\\Microsoft.N
001b:err:module:import
\\Microsoft.NET\\Frame
001b:err:module:attach
mework64\\v4.0.30319\\
000f:err:service:proce
ipe!
```

CrypTool 1.4.41 - Encrypt1.txt

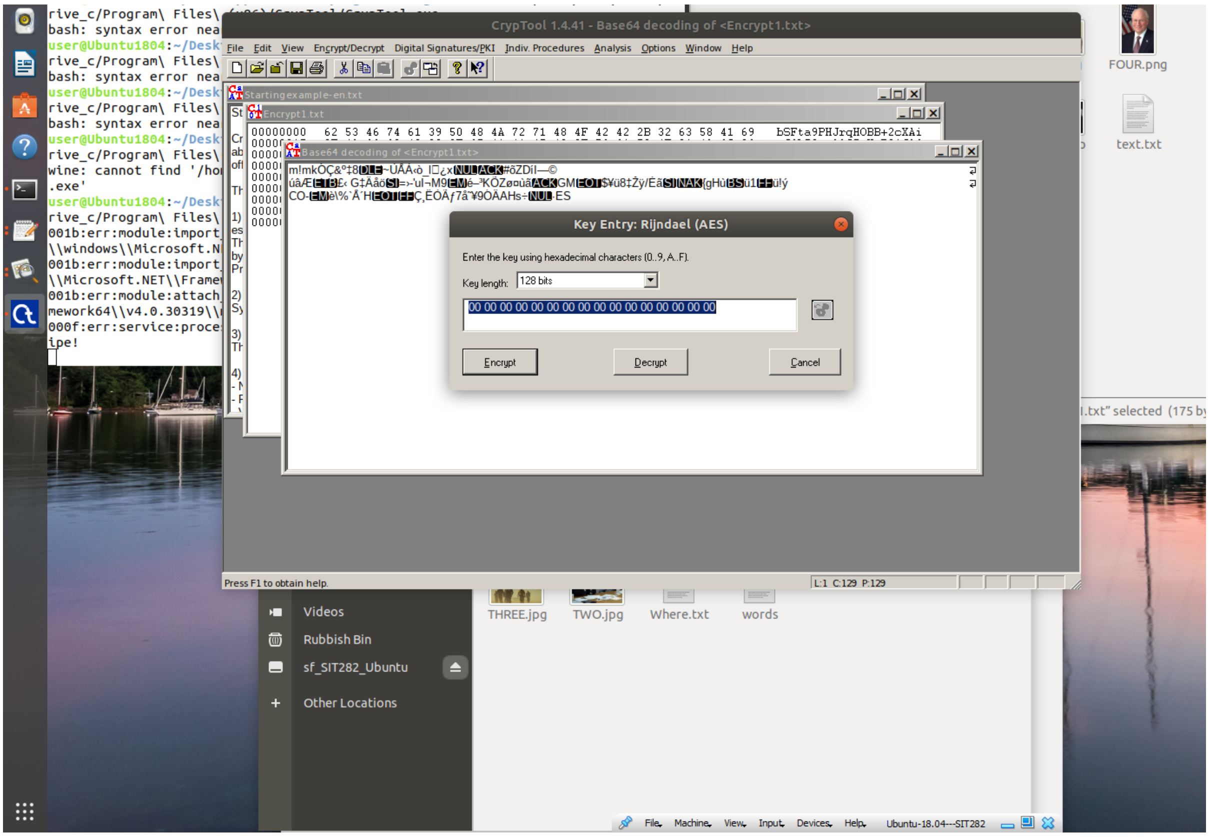
```
00000000   62 53 46 74 61 39 50 48 4A 72 71 48 4F 42 42 2B 32 63 58 41 69   bSFta9PHJrqHOBB+2cXAi
00000015   2F 4A 66 62 49 32 2F 65 41 41 47 49 2F 56 61 52 4F 31 4A 6C 36   /JfbI2/eAAGI/VaRO1Jl6
0000002A   6B 4A 2B 75 4C 47 46 36 4F 4C 6F 45 65 48 77 75 58 32 44 7A 32   kJ+uLGF6OLoEeHwuX2Dz2
0000003F   62 0A 72 53 64 31 53 61 78 4E 4F 52 6E 70 6C 72 4E 4C 31 56 72   b.rSd1SaxNORnplrNL1Vr
00000054   34 70 50 6E 6A 42 6B 64 4E 42 43 53 6C 2F 44 69 48 6A 76 38 76   4pPnjBkdNBCSl/DiHjv8v
00000069   79 4F 4D 50 46 58 74 6E 53 50 6B 49 2F 44 45 4D 2F 43 48 39 49   yOMPFXtnSPkI/DEM/CH9I
0000007E   45 4E 50 0A 72 52 6E 6F 58 43 56 67 78 62 52 49 42 41 7A 48 75   ENP.rRnoXCVgxbRIBAzHu
00000093   4D 76 53 77 34 4D 33 35 5A 69 6C 4F 64 50 43 51 55 68 7A 39 77   MvSw4M35ZilOdPCQUhz9w
000000A8   43 33 52 56 4D 3D 0A                                              C3RVM=.
```

(Figure 1.6)



(**Figure 1.7**)

13: From there, it was selecting Asymmetric Decryption > AES(CBC). I chose AES as its one of the most robust and widely used ciphers. Next, the user interface box appeared asking for a password. It had to be a 128-bit password in Hex values. Referring to Figure 0.7, this was a list of passwords that had yet to be used during this investigation. However, they were text values. Therefore, conversion to hex values was required to proceed with the investigation.



```
00000000   62 53 46 74 61 39 50 48 4A 72 71 48 4F 42 42 2B 32 63 58 41 69    bSFta9PHJrqHOBB+2cXAi
```

Key Entry: Rijndael (AES)

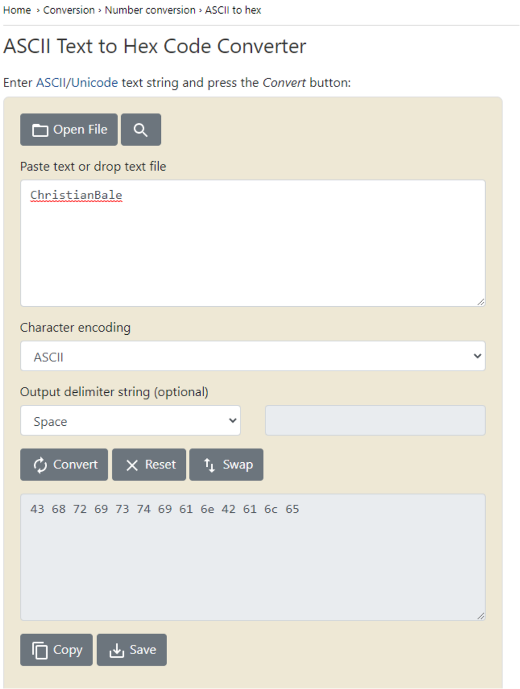
Enter the key using hexadecimal characters (0..9, A..F).

Key length: 128 bits

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

(**Figure 1.8**)

14: I used an online converter and tried every password in the list discovered earlier until I found the correct key values that allowed me to progress with the investigation. This converter changed plain text to Hex code which was required in order to continue with Cryptool.



Home › Conversion › Number conversion › ASCII to hex
ASCII Text to Hex Code Converter
Enter ASCII/Unicode text string and press the Convert button:
Open File
Paste text or drop text file
Character encoding
Output delimiter string (optional)
Convert
Reset
Swap
Copy
Save

```
ChristianBale
```

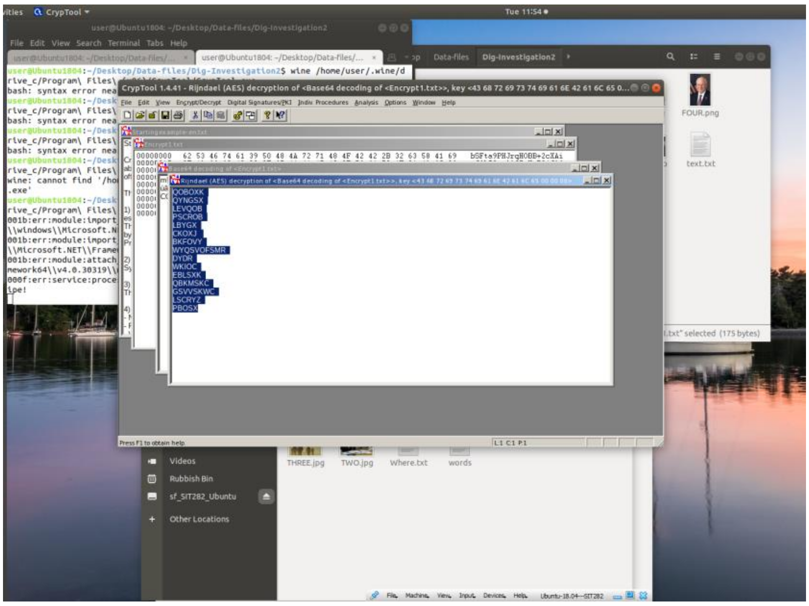
Character encoding: ASCII

Output delimiter string (optional): Space

```
43 68 72 69 73 74 69 61 6e 42 61 6c 65
```

(**Figure 1.9**)

15: Once I entered these as they were the encryption key values, the decryption began once entered. The decryption showed a new list with text values all scrambled. I knew I wasn’t finished yet as I had to convert these with Caesar Cipher (Substitution) decryption. This means each letter of the text in the file will be replaced by a letter some fixed number of positions down the alphabet.



CrypTool 1.4.41 - Rijndael (AES) decryption of <Base64 decoding of <Encrypt1.txt>>, key <43 68 72 69 73 74 69 61 6E 42 61 6C 65 0...

```
00000000   62 53 46 74 61 39 50 48 4A 72 71 48 4F 42 42 2B 32 63 58 41 69   bSFta9PHJrqHOBB+2cXAi
```

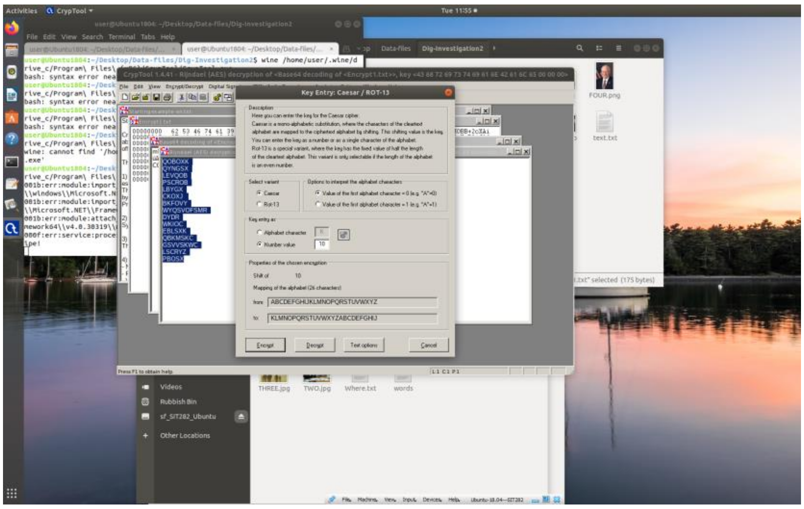
```
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ wine /home/user/.wine/d
```

Rijndael (AES) decryption of <Base64 decoding of <Encrypt1.txt>>, key <43 68 72 69 73 74 69 61 6E 42 61 6C 65 00 00 00>

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

(**Figure 2.0**)

16: When the User interface options appeared, I selected Caesar cipher as the variant, changed the numeric value to 10. Caesar Cipher involves replacing each letter of the alphabet with a letter placed down or up according to the key given. The images below show the options I chose, I then selected decrypt and a new list of names appeared in the correct order. The names appear to be of suspects linked to the case so I saved the file and made a duplicate copy, then changing the permissions of the original file in order to preserve and mitigate risks.



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

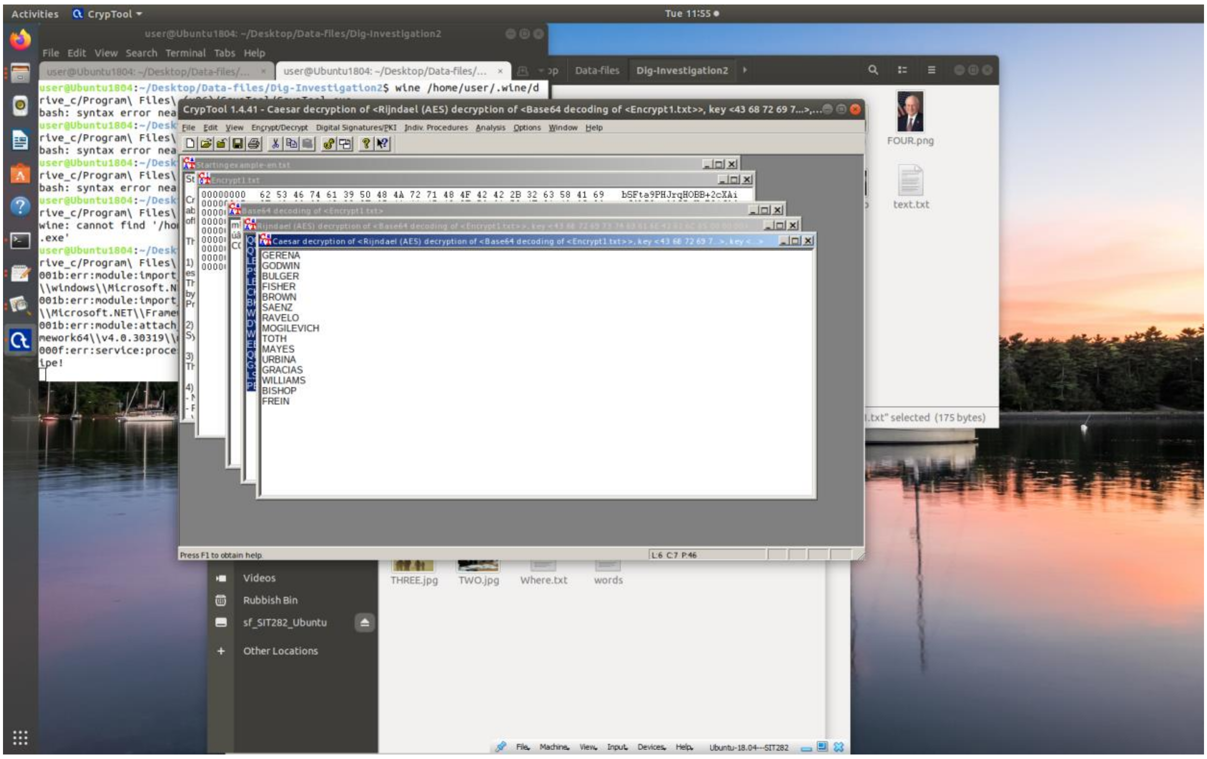
Shift of: 10

Mapping of the alphabet (26 characters)

from: ABCDEFGHIJKLMNOPQRSTUVWXYZ

to: KLMNOPQRSTUVWXYZABCDEFGHIJ

(**Figure 2.1**)



```
user@Ubuntu1804:~/Desktop/Data-files/Dig-Investigation2$ wine /home/user/.wine/d
rive_c/Program\ Files\
bash: syntax error nea
user@Ubuntu1804:~/Desk
rive_c/Program\ Files\
bash: syntax error nea
user@Ubuntu1804:~/Desk
rive_c/Program\ Files\
bash: syntax error nea
user@Ubuntu1804:~/Desk
rive_c/Program\ Files\
wine: cannot find '/ho
.exe'
user@Ubuntu1804:~/Desk
rive_c/Program\ Files\
001b:err:module:import
\\windows\\Microsoft.N
001b:err:module:import
\\Microsoft.NET\\Frame
001b:err:module:attach
mework64\\v4.0.30319\\
000f:err:service:proce
ipe!
```

CrypTool 1.4.41 - Caesar decryption of <Rijndael (AES) decryption of <Base64 decoding of <Encrypt1.txt>>, key <43 68 72 69 7...>,...

Encrypt1.txt

```
00000000   62 53 46 74 61 39 50 48 4A 72 71 48 4F 42 42 2B 32 63 58 41 69   bSFta9PHJrqHOBB+2cXAi
```

Caesar decryption of <Rijndael (AES) decryption of <Base64 decoding of <Encrypt1.txt>>, key <43 68 72 69 7...>, key <...>

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

(**Figure 2.2**)

17: This was the final piece of evidence found in this forensics investigation. Figure 2.2 showcases the last evidence found.

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.

Recommendations:

I would recommend going back and analysing the phone numbers found during the investigation at the lab. Access the last cell locations accessed by the mobile device. Gain a subpoena to collect history related to the mobile phones. When analysing the evidence, I have gathered please install write-blocking software to prevent any changes to the data. With the mobile phone numbers, you would be able to find analyse the call logs of these numbers. Analysing the call logs will enable you to obtain billing information from their respected mobile service provider, whilst also being able to see their time and date information from the call records. This will give you a time stamp which will help in providing more detailed evidence to the case. Background checks on the names discovered should be done in the lab. Background checks through the police database, and social media will provide more details on the individuals whilst also giving a clear indication of whether these individuals set off any red flags. As we have the names you could potentially obtain warrants, subpoenas to search these individuals' properties and bring them in for questioning. Analysis of the drugs should also be done in the lab to give a clear picture on what types of drugs we are dealing with in this investigation.

As we found out the suspect uses encryption, a more detailed search at the site should be done, the search should look for information relating to authors of the encryption program, passphrases, personal information (e.g., Family name, pet name etc..) about the subject as their may be more files on the computer that need passphrases to be unlocked. Autopsy should be used on other files found on the laptop in the lab. Analysing the search history on the suspects computer (Internet explorer is installed on the device). Use internet tools to identify users and internet connections. Tools such as Whois, traceroute, NSLookup and a ping search, and Netstat can all be conducted on the suspects computer which may provide more evidence in this case. The other CDs discovered also need to be thoroughly investigated as their may be more encrypted files as the subject used steganography throughout the zip file I had to analyse. I recommend decrypting other zip files found as data may have been concealed within the contents of those files.

Summary of investigation:

- The investigation was conducted using Linux (Ubuntu) which allowed me to utilise software/programs already installed on there.

- I used Zipattack to obtain the password to the locked zip file sent to me by Sandra. Fcrackzip is what I used in the terminal; it’s designed to recover the encryption passwords to unlock the contents within the zip file.
- Utilized OphCrack to analyse the NTML hashes provided by Sandra. Analysing these led me to discover more passwords associated with files we are investigating.
- S-tools were used to crack image files and gain more evidence. I used S-tools on the 1st image file. S-tools allowed me to reveal the contents in the image file.
- I used Jpseek to reveal the hidden contents in the 2nd image file. The password required was DonaldRumsfield which was discovered through OphCrack.
- Foremost tools were utilised to find the hidden contents within image file 3. Foremost can recover lost files based on headers, footers and can work on image files. This gave me the OpenPuff configuration passwords for the next step.
- OpenPuff program was used to acquire the evidence hiding in image number 4, OpenPuff enabled me to unhide the hidden file as I new the three secret keys.
- Data manipulation through HxD was conducted for image number 5. Data manipulation enabled a file to be reconstructed. This reconstructed file was then able to be analysed in Cryptool.
- Conversion of plain text to ASCII keys was needed to decrypt the file in Cryptool
- Cryptool was needed to decrypt the text of the reconstructed file, this meant using substitution ciphers being able to decrypt the file via Casear led me to the discover of names that are presumed to be suspects in the case.

Evidence Recovered:

The evidence recovered consisted of names of potential suspects that are linked to the drug crimes this police department is investigation. Names in conjunction with phone numbers were recovered through a throughout/deep forensic investigation. These findings are crucial to the case and will assist in gaining more evidence to make a stronger case in the coming days. The phone numbers and Names discovered can be analysed further in the lab. Links can be established through the names and phone numbers discovered in this forensic investigation. Based off the concealing of information and encryption of files it provides evidence that the subject has knowledge on steganography, cipher encryption, and how to use Linux tools to conceal files within other files. The lab needs to consider this when investigating more evidence

Relation to the case:

The names and phone numbers discovered I believe are related to the case. It creates a link to the people that hurriedly packaged loose powders when police investigated. They may have been employed by a drug king pin, by having these names we can bring them in for questioning to find more intel on the whole operation. By having the names and number concealed in other files it indicates these individuals are linked to the drug manufacturing location which was discovered by police in May.

Suggestions:

I suggest analysing all the evidence in the lab, locate the individuals whose names were on the list found in the investigation. Once located then tap their phones, get a search warrant to search their homes/workplace. Bring the individuals in for questioning once there is more evidence that ties them into this case. Any evidence found run a deep forensics analysis back at the lab. Deep background checks on the names found should be done. The subject has shown they use encryption and software tools to conceal files, investigators need to be wary.

Evidence Form (Figure 1-11 of the text)

SIT282 ***Computer Crime and Forensic Lab*** ***Single Evidence Form***			
Case No:	*001*	*Unit Number:*	*001*
Investigator:	***Connor Cameron Gent***		
Nature of Case:	*Potential drug trafficking case. Location of case has been identified as a drug manufacturing warehouse.*		
Location where evidence was obtained:	***Warehouse (Roma St, Brisbane)***		
Item # ID	*Description of evidence*	*Vendor Name*	*Model No/Serial No.*
	Laptop and four CD's found at the crime scene. The fourth CD contained a suspicious ZIP file.	*Not Specified*	*Not Specified.*
Evidence Recovered by:	***Police Officer Moti***	*Date & Time:*	*10th May 2019, 3AM*
Evidence Placed in Locker:	***Police Office Moti***	*Date & Time*	*10th May 2019, 3:30 Am*
Evidence Processed by	*Description of Evidence*		*Date & Time*
	CD and Disks found at scene		*10th May, 2019*
	Powder found on table		*10th May, 2019*
	Laptop Found		*10th May, 2019*
			Page __ of __<

Host Machine Information

Processor	*Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz*
Installed RAM	*8.00 GB (7.65 GB usable)*
Device ID	*5FBF6555-A506-4FD1-AAE5-7990690C0D8C*
Product ID	*00325-96744-43641-AAOEM*
System type	*64-bit operating system, x64-based processor*
Pen and touch	*No pen or touch input is available for this display*

***Virtual Machine Information*:**

Product: VMware Workstation 15 pro
Hard Disk: 50GB

Memory: 2GB
Version: 15.0.4 build-1299004
Virutal Machine Operation System

References:

TABLES, R. 2018. *ASCII Text to Hex Code Converter* [Online]. Available: https://www.rapidtables.com/convert/number/ascii-to-hex.html [Accessed 2021].