

## Digital Forensics

### A01 Zip file investigation

NOTE – This was produced as part of a Uni assignment. Some elements contained were included as they were required as part of the assignment brief and/or marking rubric. Additionally, the assignment had a strict word count, some elements had to be sacrificed.

This is my work, produced for the Bachelor of Cyber Security. Universities take plagiarism very seriously and automated tools are very effective at identifying the source of information. I am happy for this to be used as a source for learning. Keep in mind, I am learning also, some info may not be correct, you should always confirm with reputable sources. This information is likely out of date as it was produced some time between 2018 – 2021.

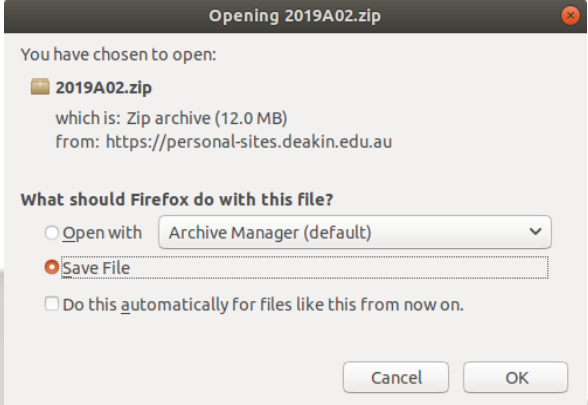
#### Table of Contents

DIGITAL FORENSIC PROCEDURE .....	3
1. Explain how you downloaded the file, what precautions you took, and how you ensured its integrity.....	3
File Download Procedure .....	3
Precautions Applied .....	3
Method used to ensure Integrity.....	3
2. Describe how you decrypt two given NTLM hash values by using OphCrack, including screen shots.....	4
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.....	5
Steps performed to open the file were:.....	5
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. ....	5
Content description .....	5
5. What tools will you now use to proceed your investigation and why?.....	6
6. Describe how your investigation proceeded at this point, including screen shots.....	6
Summary of findings during investigation of Zip file – A01.....	12
2. Recommendations. ....	12
3. Summary of steps performed. ....	12
4. Summary of what was recovered. ....	12
5. Relation to the case.....	13
6. Further investigation.....	13
<i>Evidence Form</i> .....	13



## DIGITAL FORENSIC PROCEDURE

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

<b>File Download Procedure</b>	<ol style="list-style-type: none"> <li>1. Accessed <a href="http://www.deakin.edu.au/XXXXXXXXX">http://www.deakin.edu.au/XXXXXXXXX</a> on UBUNTU Virtual Machine.</li> <li>2. Read disclaimer, clicked proceed. Saved file, '2019A02.zip' to downloads folder on UBUNTU machine</li> </ol>  <ol style="list-style-type: none"> <li>3. Moved the file to the working folder in preparation for copying</li> </ol>
<b>Precautions Applied</b>	<ol style="list-style-type: none"> <li>1. Checked URL was accurate</li> <li>2. Read disclaimer</li> <li>3. Checked MD5 hash value regularly once downloaded to Downloads folder to check for corruption.</li> <li>4. Zip was copied, all work was conducted on the copy. <pre> user@Ubuntu1804:~/Desktop/Assessment\$ sudo dd if=2019A02.zip of=2019A02v.dd 24607+1 records in 24607+1 records out 12598879 bytes (13 MB, 12 MiB) copied, 0.0983407 s, 128 MB/s user@Ubuntu1804:~/Desktop/Assessment\$ ls 2019A02v.dd  2019A02.zip  tempfile user@Ubuntu1804:~/Desktop/Assessment\$ md5sum 2019A02v.dd 9ec1c8f62429182349f3979c39aed8fb 2019A02v.dd user@Ubuntu1804:~/Desktop/Assessment\$ sudo dcfldd if=2019A02.zip vf=2019A02v.dd Total: Match  user@Ubuntu1804:~/Desktop/Assessment\$ md5sum 2019A02v.dd 9ec1c8f62429182349f3979c39aed8fb 2019A02v.dd </pre> </li> <li>5. File permissions for original and copy were changed to read only.</li> </ol>
<b>Method used to ensure Integrity</b>	<ol style="list-style-type: none"> <li>1. Initially checked MD5 hash value provided against the MD5 has produced by the linux command 'MD5sum 2019A02.zip'.</li> <li>2. This was checked in the Downloads folder after downloading.</li> <li>3. Checked again after moving the file to the working folder.</li> <li>4. Check again and a SHA1 hash was calculated after the zip file was copied. The original and copy had MD5 and SHA1 hashes calculated and compared using the bash command and the 'HASHCALC' tool.</li> </ol> <pre> user@Ubuntu1804:~\$ cd Downloads/ user@Ubuntu1804:~/Downloads\$ ls 2019A02.zip  jcryptool  jcryptool-0.9.9-linux.gtk.x86_64.tar.gz  tables_xp_free_small.zip user@Ubuntu1804:~/Downloads\$ md5sum 2019A02.zip 9ec1c8f62429182349f3979c39aed8fb 2019A02.zip user@Ubuntu1804:~/Downloads\$ mv 2019A02.zip ~/Desktop/Assessment/ user@Ubuntu1804:~/Downloads\$ cd .. user@Ubuntu1804:~\$ ls Desktop  Documents  Downloads  examples.desktop  Music  Pictures  Public  Templates  Videos  xp_free_small user@Ubuntu1804:~\$ cd Desktop/ user@Ubuntu1804:~/Desktop\$ cd Assessment/ user@Ubuntu1804:~/Desktop/Assessment\$ ls 2019A02.zip  tempfile user@Ubuntu1804:~/Desktop/Assessment\$ md5sum 2019A02.zip 9ec1c8f62429182349f3979c39aed8fb 2019A02.zip </pre>

```

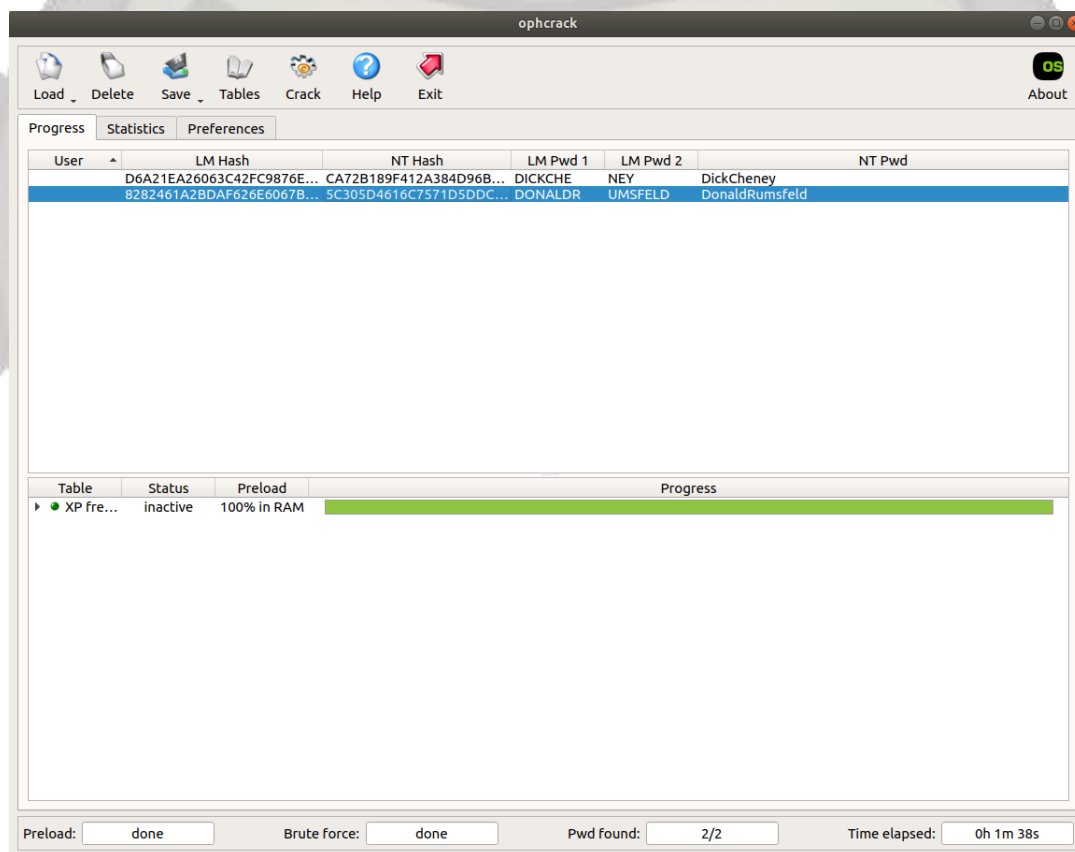
user@Ubuntu1804:~/Desktop/Assessment$ md5sum 2019A02.zip
9ec1c8f62429182349f3979c39aed8fb 2019A02.zip
user@Ubuntu1804:~/Desktop/Assessment$ sha1sum 2019A02.zip
dd91f418dcb81d16ba516bd7243b37cab8709438 2019A02.zip
user@Ubuntu1804:~/Desktop/Assessment$ md5sum 2019A02v.dd
9ec1c8f62429182349f3979c39aed8fb 2019A02v.dd
user@Ubuntu1804:~/Desktop/Assessment$ sha1sum 2019A02v.dd
dd91f418dcb81d16ba516bd7243b37cab8709438 2019A02v.dd

```

5. The original was retained and all further work was conducted on the copy (2019A02v.dd)

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


1. Open terminal
2. Open Ophcrack with the 'ophcrack' command
3. Select 'Load' then 'Single Hash'
4. Copied in the first NTLM hash, then selected 'Crack'
5. Successfully cracked, Password – **'DickCheney'**
6. Used the same process to load and crack second NTLM
7. Successfully cracked, Password – **'DonaldRumsfeld'**



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.

<b>Steps performed to open the file were:</b>	<ol style="list-style-type: none"> <li>1. cd into working directory where working file was located</li> <li>2. Used fcrackzip -u -D -p '/usr/share/dict/american-english' 2019A02v.dd command to conduct a dictionary attack on the encrypted zip drive.  <pre>user@Ubuntu1804:~/Desktop/Assessment\$ fcrackzip -u -D -p '/usr/share/dict/american-english' 2019A02v.dd</pre> <p>PASSWORD FOUND!!!!: pw == <b>vice</b></p> </li> <li>3. This is a loose thematic relationship between the password used to open the Zip file and the cracks NTLM passwords.</li> <li>4. The password 'vice' was then used to extract the files into a working folder.  <pre>user@Ubuntu1804:~/Desktop/Assessment\$ unzip 2019A02v.dd -d tempfile</pre> <p>Archive: 2019A02v.dd  [2019A02v.dd] FIVE.jpg password:  replace tempfile/FIVE.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: A  extracting: tempfile/FIVE.jpg  extracting: tempfile/FOUR.png  inflating: tempfile/ONE.bmp  inflating: tempfile/THREE.jpg  inflating: tempfile/TWO.jpg</p> <pre>user@Ubuntu1804:~/Desktop/Assessment\$ cd tempfile/</pre> <pre>user@Ubuntu1804:~/Desktop/Assessment/tempfile\$ ls</pre> <p>FIVE.jpg FOUR.png ONE.bmp THREE.jpg TWO.jpg</p> <pre>user@Ubuntu1804:~/Desktop/Assessment/tempfile\$</pre> </li> <li>5.</li> </ol>
---	--

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.

<b>Content description</b>	<p>The zip file contained Five images named ONE through to FIVE. Three file types .jpg, .bmp and .png. All images appear to have a consistent theme of United States political leaders.</p> <p>ONE.bmp – person presenting to crowd. 2M 2F on a stage all in suits. American flags behind against a blue curtain backdrop.</p> <p>TWO.jpg – 1M 1F in suits at a desk with 'classified' papers strewn across it. Looking concerned.</p> <p>THREE.jpg – 6M 1 unknown on L edge of frame. 3 are in Middle eastern garb, 1 in a tan military uniform, 2 in suits. Appears to be a desert setting. White or silver wagon in the background.</p> <p>FOUR.png – 1M suit and tie with US flag lapel pin. Professional style portrait photo. American flag in background.</p> <p>FIVE.jpg – 3M in suits. Grainy photo appears to be older. Yellow drapes in background with possibly a flag. Assorted items behind figures on tables.</p> <div data-bbox="413 1565 1337 1812">  <p>FIVE.jpg      FOUR.png      ONE.bmp      THREE.jpg      TWO.jpg</p> </div>
----------------------------	--



File Name	File Type	MD5 Hash Value
ONE.bmp	Image (.bmp)	ab873ec4d5c826db5d337f5f287006d5
TWO.jpg	Image (.jpg)	4da131832b963f03f990d4c545b2d533
THREE.jpg	Image (.jpg)	004b451689688f2d9bb83fb3fc5607aa
FOUR.png	Image (.png)	ac88ed263a80632167102c93a966f655
FIVE.jpg	Image (.jpg)	815025ac61891bf35ea4f38d7c543db0

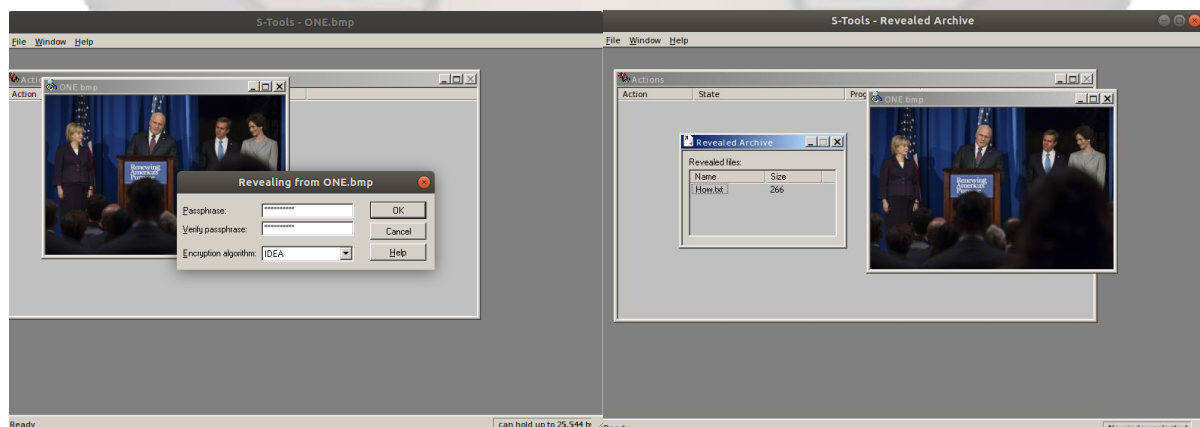
### 5. What tools will you now use to proceed your investigation and why?

Tool	Reason
Steg Detect	Working with .jpg images, it is worth attempting to see if Steg Detect finds anything
S-Tools	Again, identify if there are any images hidden inside the .bmp files Will require a password)
JPseek	Extract any hidden .txt files from .jpg (will require a password)
HxD	To inspect the images specifically identify any changes to the images.
Openpuff	This was listed as being identified on the source computer

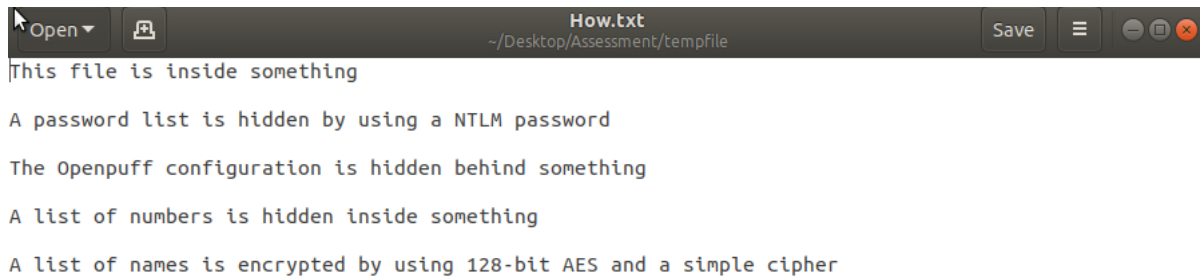
### 6. Describe how your investigation proceeded at this point, including screen shots.

With the Zip file extracted to the working folder, I initially began by opening each image and inspecting it for anything obvious. Initial inspection found nothing of note.

With no other information, I decided to start with image ONE.bmp as that seemed logical. I opened S-Tools using 'wine ~/Desktop/win-tools/jphide\ and \Stegbreak/S-tool/S-Tools.exe' then opened image ONE.bmp from the working folder. Right clicking the image and selecting reveal asked for a password. I currently had three passwords, **DickCheney**, **DonaldRumsfeld** and **vice**. My intent was to try all 3 in various combinations. Entering **DickCheney** into the two passphrases revealed a .txt file called How.txt. This was then saved to the working folder for inspection.



Opening How.txt revealed a series of cryptic instructions. These instructions lined up with the number of images so this became my guide going forward.



The instructions for the next step described a password list being hidden using a NTLM password. As I had used **DickCheney** already I assumed it would be **DonaldRumsfeld**, but didn't discount **DickCheney**.

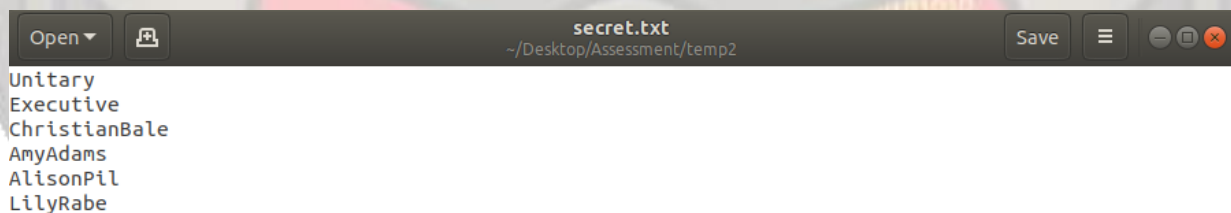
Image TWO.jpg was a jpeg so I opened ran jpseek, 'wine ~/Desktop/win-tools/jphide\ and \Stegbreak/jpseek.exe TWO.jpg secret.txt', against it, knowing I would be using one of the NTLM passwords.

```
user@Ubuntu1804:~/Desktop/Assessment/temp2$ wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/jpseek.exe TW0.jpg secret.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 cryptography which may be subject to local laws.
```

Passphrase:

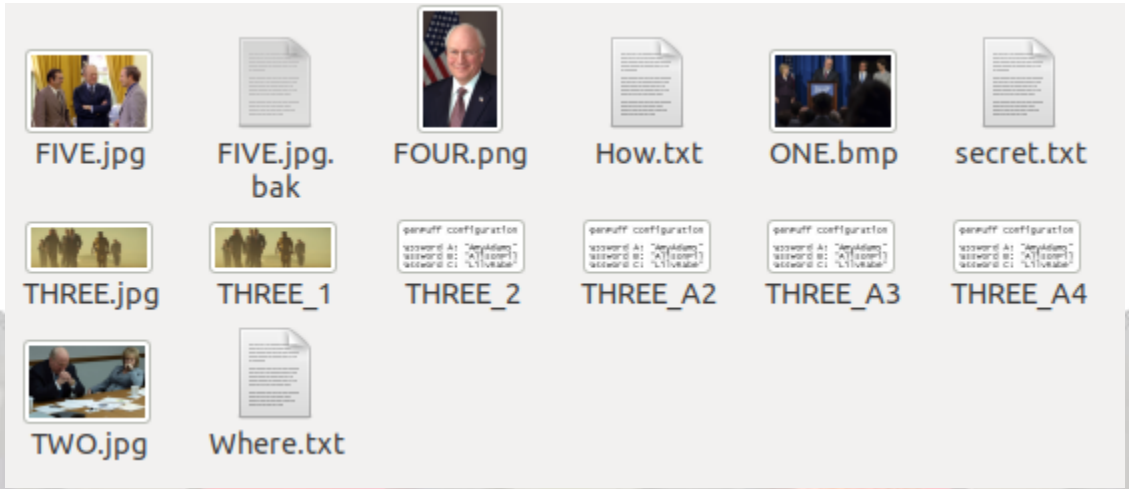
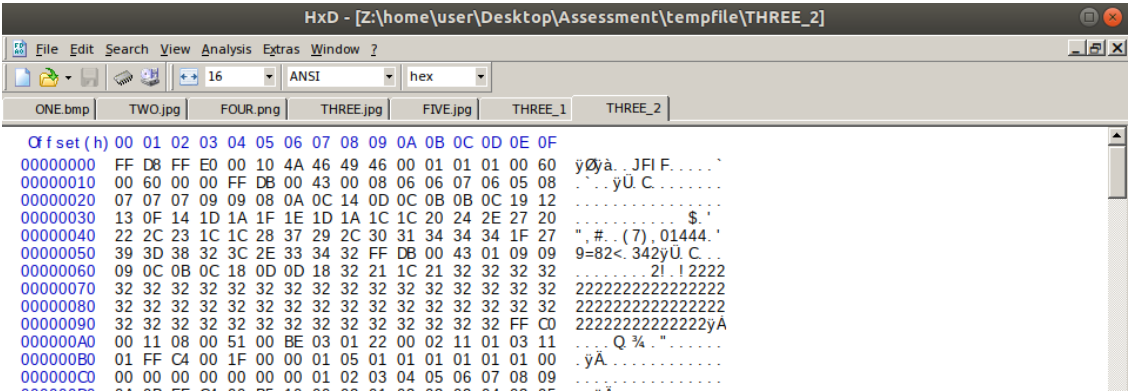
The passphrase DonaldRumsfeld was successful and the secret.txt file was created in the working folder. Opening the secret.txt file revealed a list of six words/names. No additional information was provided about their use.



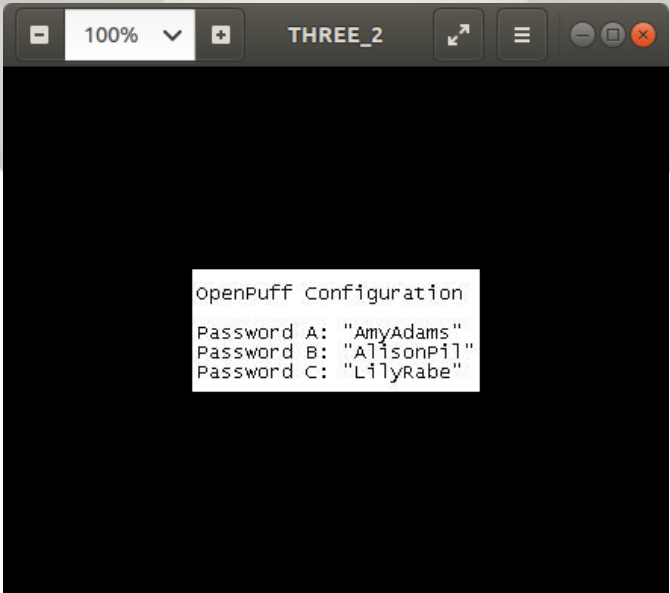
With THREE.jpg, the clue was that openpuff configuration was hidden behind something. This gave me the impression the image was being masked by another image. I opened THREE.jpg in HxD Hex editor and had an initial scan of file. The header and End of File appeared correct. During an additional scan I noticed there was an additional EOF and header within the image body at line 00009470.

00009470	D9	B5	EB	E6	FF	00	B0	00	0F	A5	E0	FD	71	E6	EF	FB	Ùeay. °... Yàÿqàí ù
00009480	57	FF	D9	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	WÜÿÿä... JFI F...
00009490	01	00	60	00	60	00	00	FF	DB	00	43	00	08	06	06	07	...yÜ. C...
000094A0	06	05	08	07	07	07	09	09	08	0A	0C	14	0D	0C	0B	0B	
000094B0	0C	19	12	13	0F	14	1D	1A	1F	1E	1D	1A	1C	1C	20	24	\$
000094C0	2E	27	20	22	2C	23	1C	1C	28	37	29	2C	30	31	34	34	" , #. ( 7 ) , 0144
000094D0	34	1F	27	39	3D	38	32	3C	2E	33	34	32	FF	DB	00	43	4. ' 9=82< . 342yÜ. C
000094E0	01	09	09	09	0C	0B	0C	18	0D	0D	18	32	21	1C	21	32	2! !2
000094F0	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	2222222222222222
00009500	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	2222222222222222
00009510	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	2222222222222222
00009520	32	FF	C0	00	11	08	00	51	00	BE	03	01	22	00	02	11	2yA... Q ¾. "...

Investigating this produced a second image. This was copied out into its own HEX editor and saved to the working folder so it could be reviewed. The Scan function was used to search for FFD9 to identify any possible other EOFs and in the end the hidden image was found stored within the masking image three times.

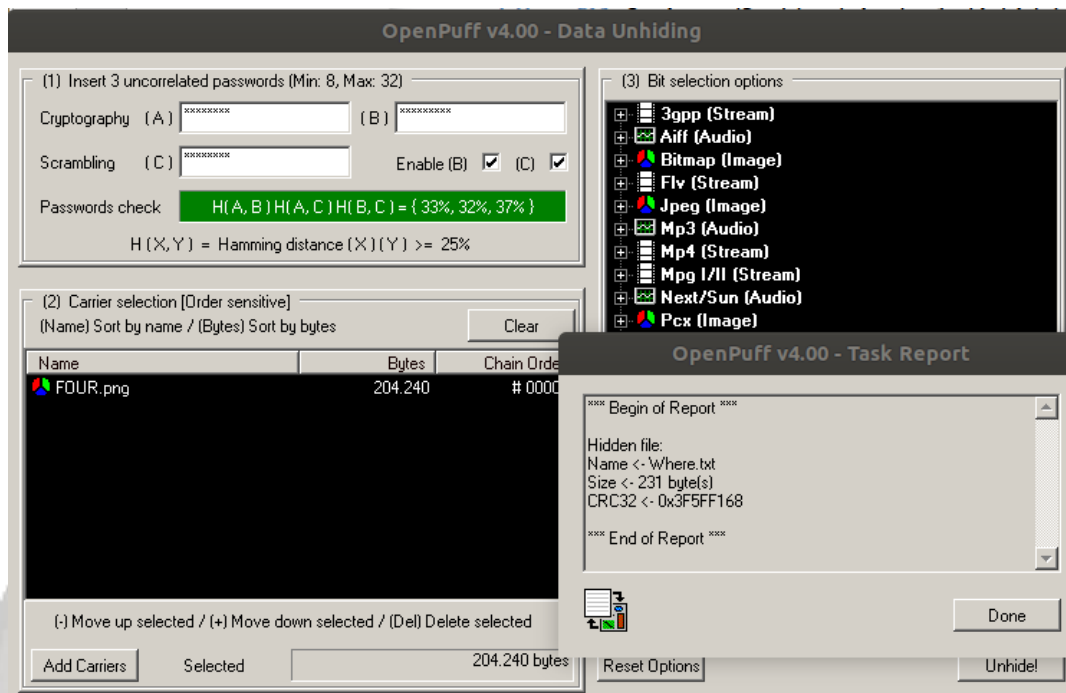


The mask image THREE\_1 was the original image images THREE\_A2 to A4 were the three copies of the hidden image. All were checked and identical. The hidden image contained three passwords under the title openpuff configuration.

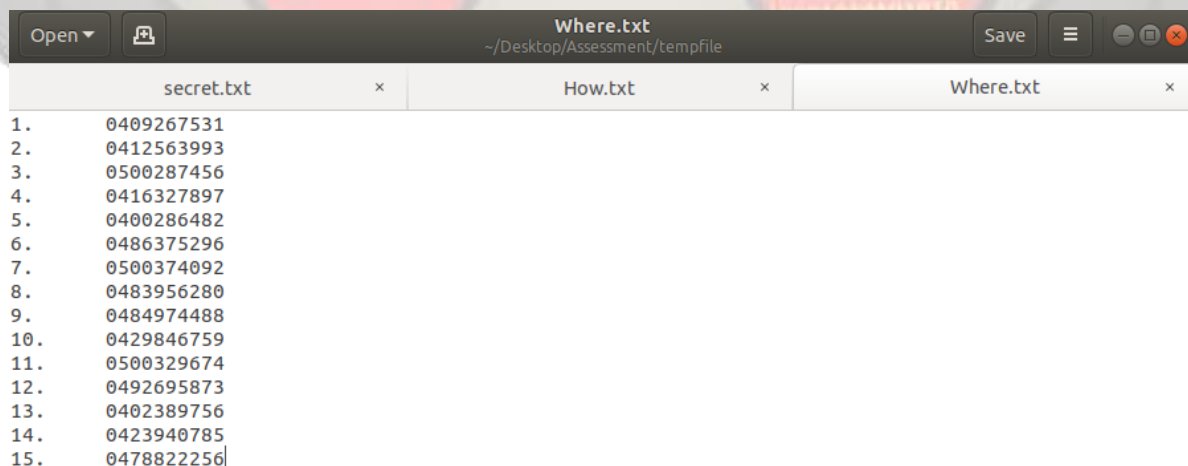




At this point I had openpuff crednetials and the How.txt described a list of numbers inside something, so I opened openpuff, 'wine ~/Desktop/win-tools/OpenPuff/OpenPuff.exe', and entered the passwords as detailed in the above image. I selected FOUR.png as the carrier and selected unhide.



This identified a file called Where.txt which was saved to the working folder. Inspection of the file revealed a list of 15 10 digit numbers. No other information was identified.

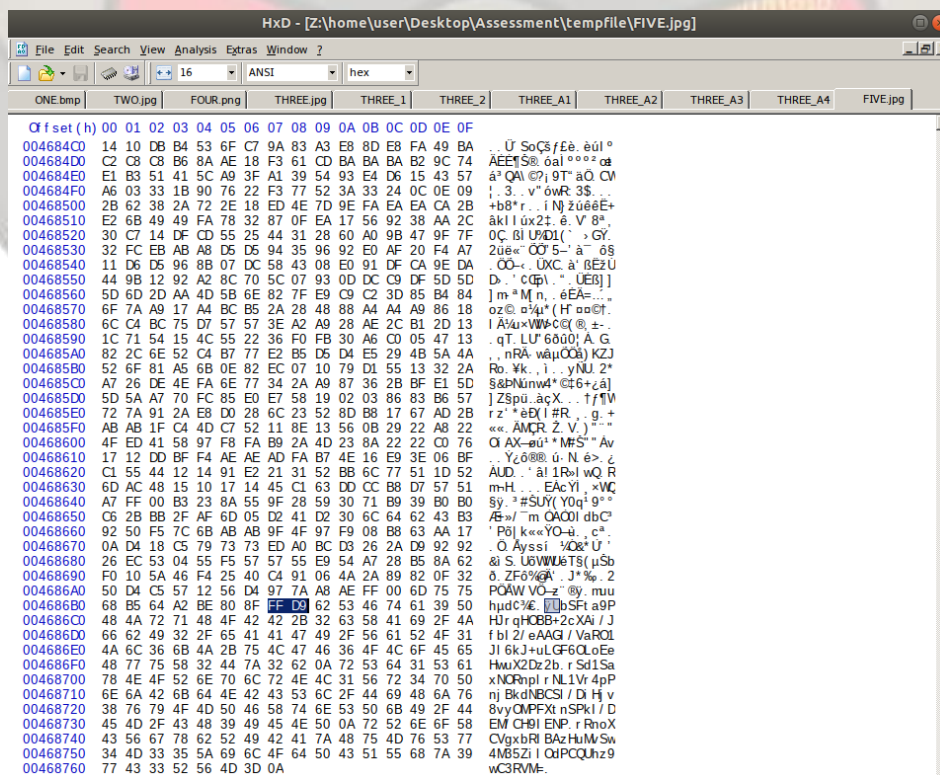
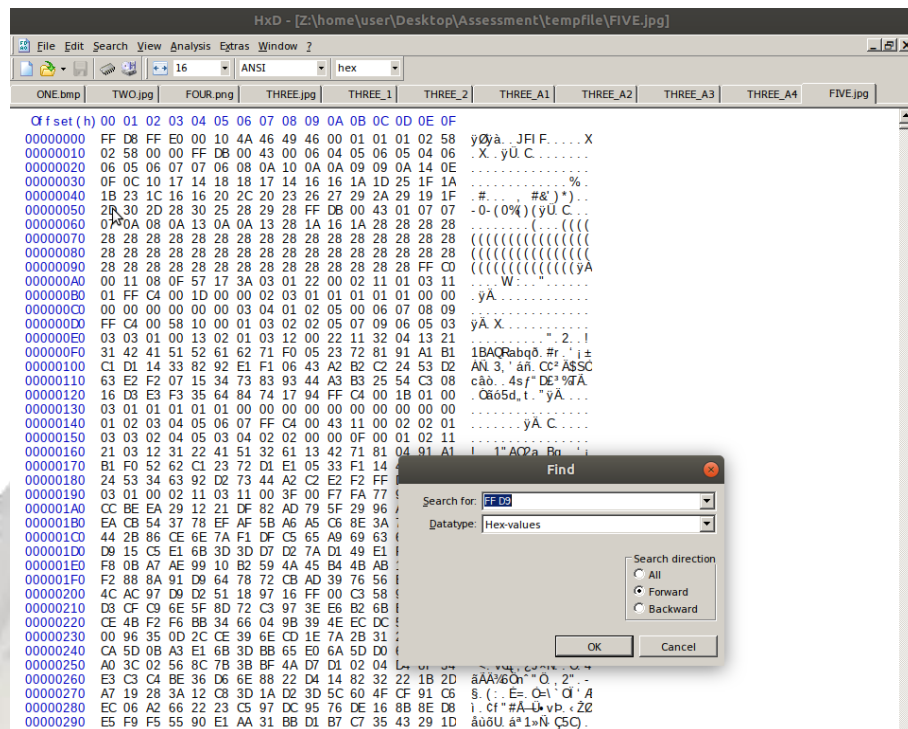


The numbers don't appear to have a pattern.

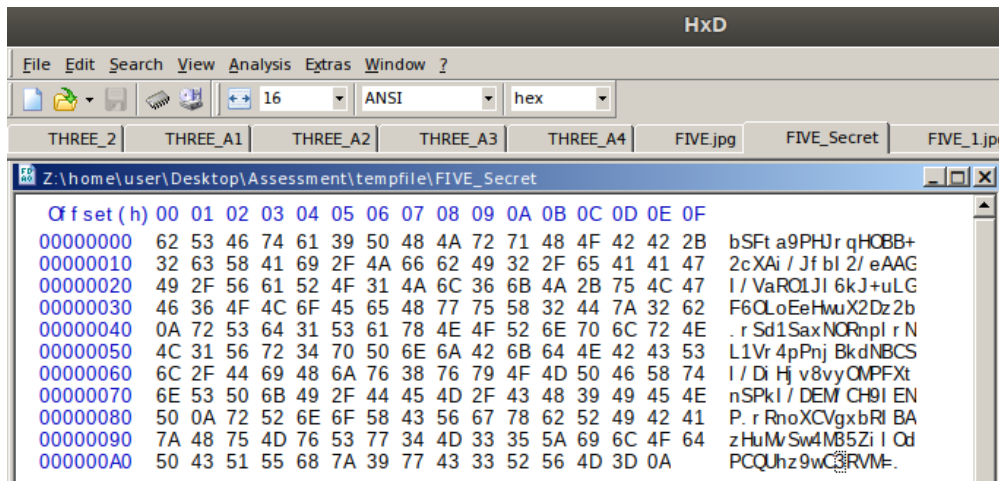
I moved on to FIVE.jpg and initially tried to use steg detect as I didn't think the whole image would be encrypted as it still opens.

```
user@ubuntu1804:~/Desktop/Assessment/tempfile$ wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/stegdetect/stegdetect.exe FIVE.jpg
Corrupt JPEG data: bad Huffman codeFIVE.jpg : negativeuser@ubuntu1804:~/Desktop/Assessment/tempfile$
```

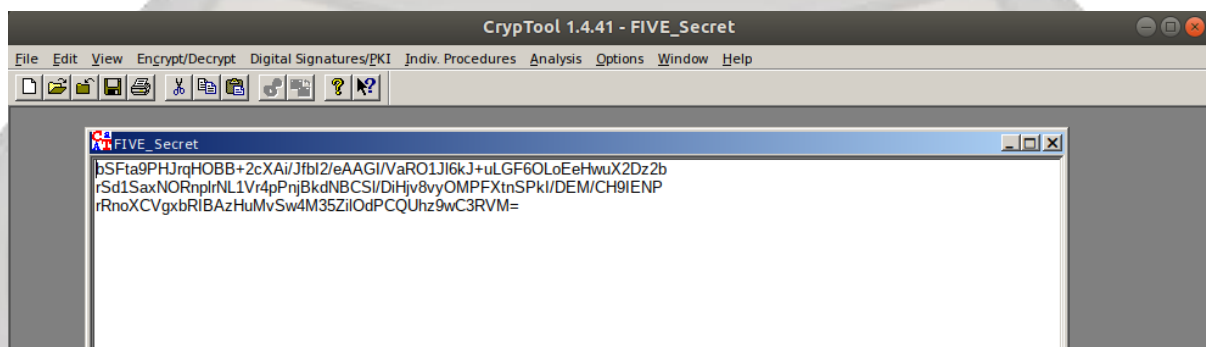
Steg detect revealed the file was corrupt. This didn't seem right as the image opened as was viewable. So I moved on to inspecting the image in HxD. I checked the Header and it appeared normal. I scrolled to the bottom of the image and noticed the EOF (FF D9) was missing. Using the search function the EOF was identified a few rows up, indicating additional information had been appended to the end of the image.



This additional data was copied out and saved as its own .txt file.



I believed this was the encrypted date the How.txt was referring to, so I opened CryptTool, 'wine /home/user/.wine/drive\_c/Program\ Files\ \(\x86\)/CrypTool/CrypTool.exe', and opened FIVE\_Secret.txt.



I then proceeded to try many methods to break the encryption and reveal the message but was unable to solve the problem. I initially shifted the characters using the Rot13 decryption as the How.txt described it as a basic cypher and I did not have a key for a Caesar cypher.

My thoughts were the Where.txt aligned to the require 16 bits AES key. 15 numbers were provided so the first or last could have been '00'. I attempted to convert the 10 digit numbers into a two digit number then convert that to hex and tried a few combinations but that did not work.

FIVE.jpg remains unsolved.

### Summary of findings during investigation of Zip file – A01

1. The following is a summary of the investigation of a Zip file recovered on May 10 from a warehouse behind Roma st Station in Brisbane QLD. The Zip file was extracted from a CD captured with CDs and a Laptop.
2. **Recommendations.** The procedure I used and my findings should be corroborated by repeating the process, ensuring that all information is accurate and no hidden information was missed. If possible, alternate programs should be utilised for the extraction of data.
3. **Summary of steps performed.**
  - Opened provided link in Virtual Machine (VM) and downloaded the Encrypted Zip (2019A02.zip) to the VMs downloads folder. MD5s were generated and check before and after moving to the working folders.
  - Conducted a bit stream copy using dd, changed both original and copy to read only permissions then MD5 and SHA1s were calculated and computed using the Linux command and HASHCALC to ensure integrity of the copy. Original was retained and all further works were completed on the copy.
  - Cracked both the provided NTLM hashes with Ophcrack, which provided **DickCheney** and **DonaldRumsfeld**.
  - The Zip was encrypted so I utilised fcrackzip to successfully brute-force the password. This returned the password **vice**. I was then able to extract the contents of the Zip into a working folder. Hashes were generated and recorded for the five images that were extracted.
  - The Zip contained five images, these were each inspected and all opened as would be expected, none appeared corrupted. Each image was then investigated in order of their naming convention.
  - **ONE.bmp** was loaded in S-Tools and. Selecting reveal and entering the password **DickCheney** revealed a hidden text file, **How.txt**. Investigating this file provided semi cryptic instruction so extracting further data from the other images.
  - Using jpseek and the other NTLM password as guided by the **How.txt**, I was able to extract a password list from **TWO.jpg**. Investigating the list revealed six words/names; **Unitary, Executive, ChristianBale, AmyAdams, AlisonPil, and LilyRabe**.
  - Hint three indicated some data was hidden behind **THREE.jpg**, so I reveied the image in the HxD hex editor and found multiple headers and EOFs. This resulted in being able to extract 3 copies of the same image. The image contained the key for openpuff, **AmyAdams, AlisonPil, and LilyRabe**.
  - With the key from **THREE.jpg**, I launched openpuff and entered the three key values, then loaded **FOUR.png** as the carrier. Selecting 'unhide' revealed a text file labelled **Where.txt**. Inspecting Where.txt revealed a list of 15 10-digit numbers.
  - Investigating FIVE.jpg in HxD revealed additional data appended after the EOF. I extracted this data into its own file. I attempted to decrypt this data, but was unsuccessful.
4. **Summary of what was recovered.** In total I recovered the five original images then from these three additional images (all the same image), three text files and an encrypted file.



- ONE.bmp, TWO.jpg, THREE.jpg, FOUR.png and FIVE.jpg are all benign looking images with a US presidential theme.
- How.txt provided loose guidance for extracting data from the other images
- Secret.txt contains the list of passwords three of which were the key for use in openpuff.
- THREE\_A2, THREE\_A3 and THREE\_A4 are the duplicate image hidden behind THREE.jpg and outlined the key for openpuff.
- Where.txt is a list of numbers likely the clue for solving the AES encryption key to be used on the encrypted data appended to the end of FIVE.jpg.
- FIVE\_Secret is possibly the encrypted file that How.txt describes as containing a list of names.

5. **Relation to the case.** The level the operator went to, to conceal this data using cryptography suggests the unidentified names were important enough to protect. It is possible, as this data was found in a drug lab, that the file containing the names will have criminal links. The last file must be fully investigated before a concrete relation can be made.

6. **Further investigation.** FIVE.jpg has not been fully exploited, I was unable to crack the final encryption. Once my previous findings have been corroborated, efforts should be made to crack the AES encryption by finding the required key. According to the information provided by the extracted How.txt, this should reveal a list of names. The list of names should be forward to the investigators to establish any persons of interest. The names and passwords should be cross referenced with all extracted data from the other captured devices. There was a general theme to the images and passwords and this may help build a profile of the user.

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

UNIT 66			
This form is to be used for only one piece of evidence. Fill out a separate form for each piece of evidence.			
Case No:	A01	Unit Number:	Unit_66
Investigator:	XXXXXXXX		
Nature of Case:	Suspected drug manufacturing lab		
Location where evidence was obtained:	http://www.deakin.edu.au/XXXXXXXX		
Item # ID	Description of evidence	Vendor Name	Model No/Serial No.
001	Encrypted .zip file named 2019A02.zip	N/A	N/A
Evidence Recovered by:	XXXXXX	Date & Time:	12 Sep 2020 @ 14:45
Evidence Placed in Locker:	Copy stored on Virtual machine	Date & Time	12 Sep 2020 @ 14:50
Evidence Processed by	Description of Evidence	Date & Time	
XXXXXX	Downloaded evidence, read brief from Sandra	12 Sep 2020 @ 1445	
XXXXXX	Created working copy, secured original.	12 Sep 2020 @ 1450	
XXXXXX	Processed Evidence and extracted data from copy	12 Sep 2020 @ 1450	
XXXXXX	Processed Evidence and extracted data from copy	06 Oct 2020 @ 1930	
XXXXXX	Processed Evidence and extracted data from copy	08 Oct 2020 @ 2000	
			Page 1 of 1