

## DIGITAL FORENSICS LAB

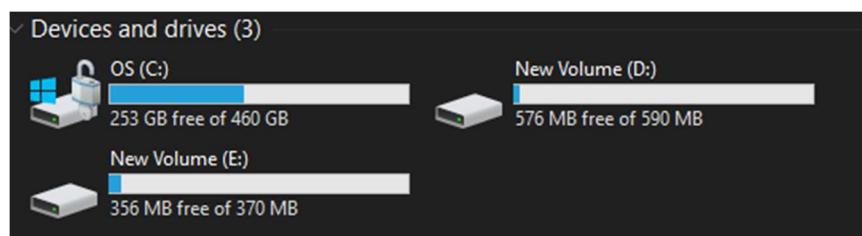
Exercise 13	
Name	S Shyam Sundaram
Registration Number	19BCE1560
Slot	L39+L40
Faculty	Dr. Seshu Babu Pulagara
Date	23 <sup>rd</sup> November, 2021

### AIM

Exploring various methods of data hiding.

### Partition Hiding

The drive currently has partitions as shown below:



To hide partition E:, open the terminal in Admin mode. Then type in 'Diskpart' and press enter. Then enter 'list volume' to list the volumes present.

```
C:\Users\mails>Diskpart

Microsoft DiskPart version 10.0.19041.964

Copyright (C) Microsoft Corporation.
On computer: SHYAM-DELL-2

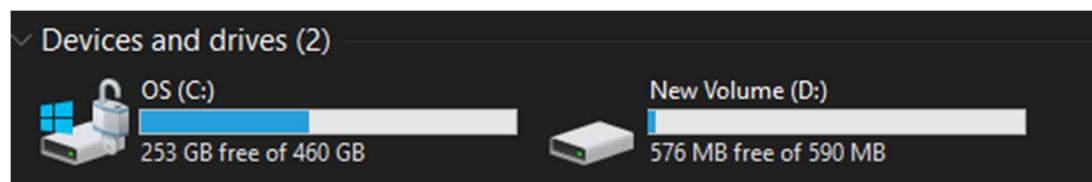
DISKPART> list volume

  Volume ###  Ltr  Label        Fs    Type        Size     Status      Info
  -----  ---  -----  -----  -----  -----  -----
  Volume 0            FAT32  Partition   512 MB  Healthy    Hidden
  Volume 1      C  OS          NTFS  Partition   460 GB  Healthy    Boot
  Volume 2            ESP         FAT32  Partition  150 MB  Healthy    System
  Volume 3            NTFS  Partition   990 MB  Healthy    Hidden
  Volume 4            Image  NTFS  Partition    13 GB  Healthy    Hidden
  Volume 5            DELL SUPPORT NTFS  Partition 1455 MB  Healthy    Hidden
  Volume 6      E  New Volume  NTFS  Removable  371 MB  Healthy
  Volume 7      D  New Volume  NTFS  Removable  591 MB  Healthy
```

Since we want to hide E:. enter 'select volume 6' and enter 'remove letter E'. The partition is now hidden.

```
DISKPART> select volume 6  
Volume 6 is the selected volume.  
  
DISKPART> remove letter E  
  
DiskPart successfully removed the drive letter or mount point.  
  
DISKPART> |
```

The partition and its contents are no longer visible.

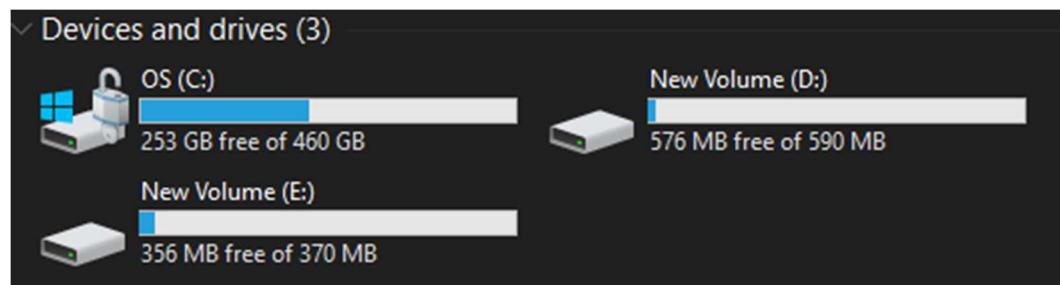


To unhide it and assign it the letter E, select the volume again and assign letter E as shown below.

```
DISKPART> list volume  


| Volume ### | Ltr | Label       | Fs    | Type      | Size    | Status  | Info   |
|------------|-----|-------------|-------|-----------|---------|---------|--------|
| Volume 0   |     |             | FAT32 | Partition | 512 MB  | Healthy | Hidden |
| Volume 1   | C   | OS          | NTFS  | Partition | 460 GB  | Healthy | Boot   |
| Volume 2   |     | ESP         | FAT32 | Partition | 150 MB  | Healthy | System |
| Volume 3   |     |             | NTFS  | Partition | 990 MB  | Healthy | Hidden |
| Volume 4   |     | Image       | NTFS  | Partition | 13 GB   | Healthy | Hidden |
| Volume 5   |     | DELLSUPPORT | NTFS  | Partition | 1455 MB | Healthy | Hidden |
| * Volume 6 |     | New Volume  | NTFS  | Removable | 371 MB  | Healthy |        |
| Volume 7   | D   | New Volume  | NTFS  | Removable | 591 MB  | Healthy |        |

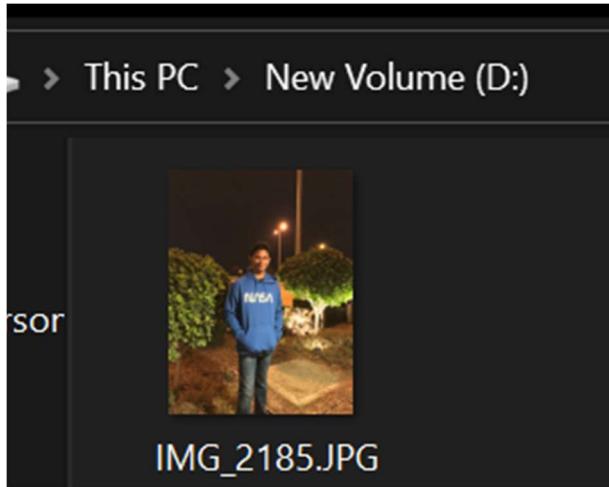
  
DISKPART> select volume 6  
Volume 6 is the selected volume.  
  
DISKPART> assign letter E  
  
DiskPart successfully assigned the drive letter or mount point.
```



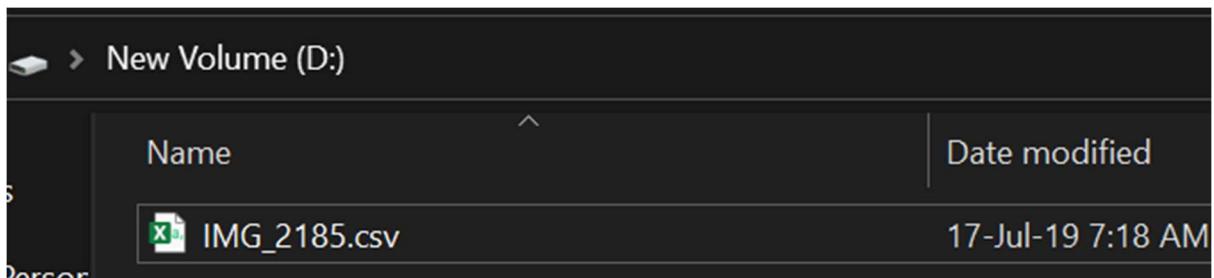
The partition is now unhidden.

## Changing file extensions

Here, we just change the extension of the file to be hidden. Say I wish to hide the JPG file below.



The extension, JPG, can be changed to something else.



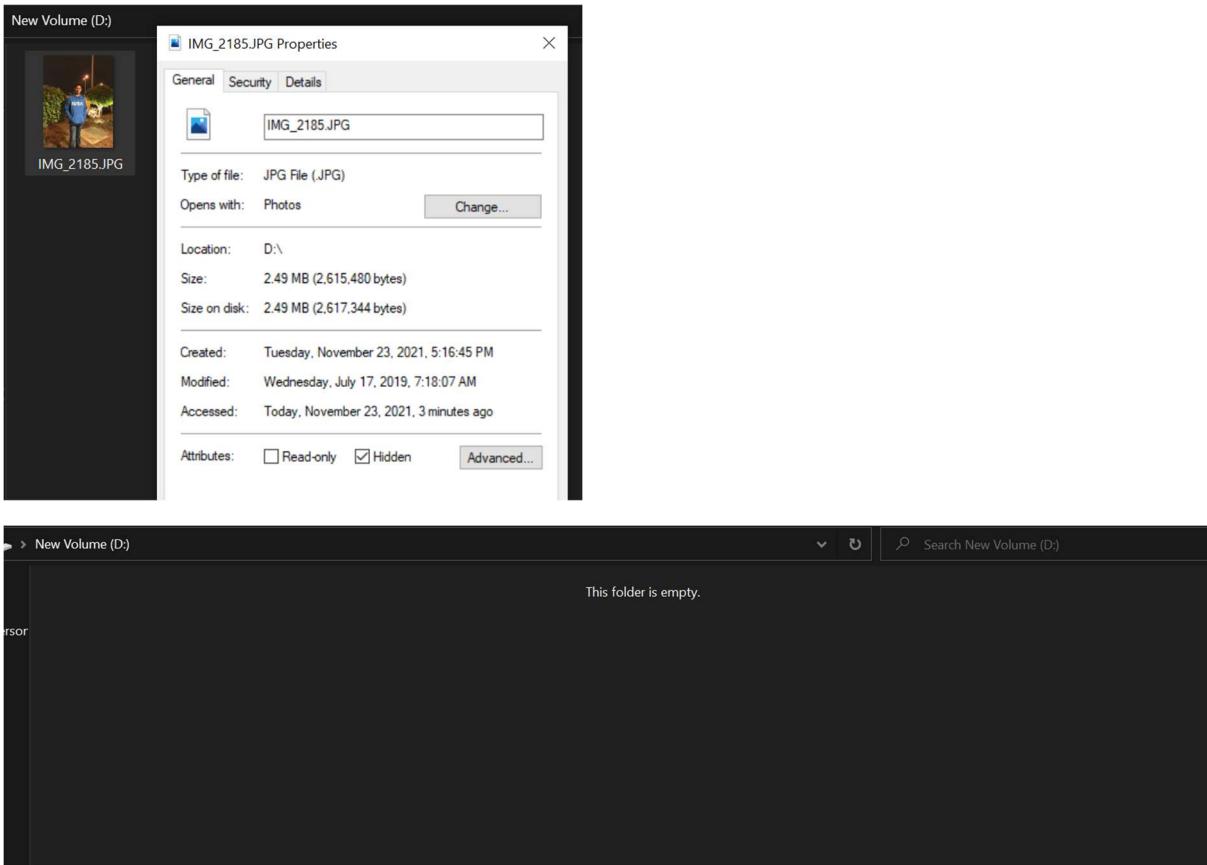
But the file would appear to be gibberish:

A screenshot of Microsoft Excel showing a single cell (A1) containing a large amount of binary-looking data. The cell is formatted with a general number format. A warning message at the top of the Excel window says: 'POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.' There are buttons for 'Don't show again' and 'Save As...'. The rest of the Excel interface is visible, including the ribbon, toolbars, and other cells which are also filled with similar binary data.

We can obtain it back by changing the extension back to JPG.

## Setting File attributes to Hidden

This is done by selecting the file's 'Hidden' attribute in properties.



To unhide, click on View in file explorer, then select 'Hidden Item' in 'Show/Hide' group. The file will be visible and then the 'Hidden' attribute can be unchecked.

## Bit Shifting

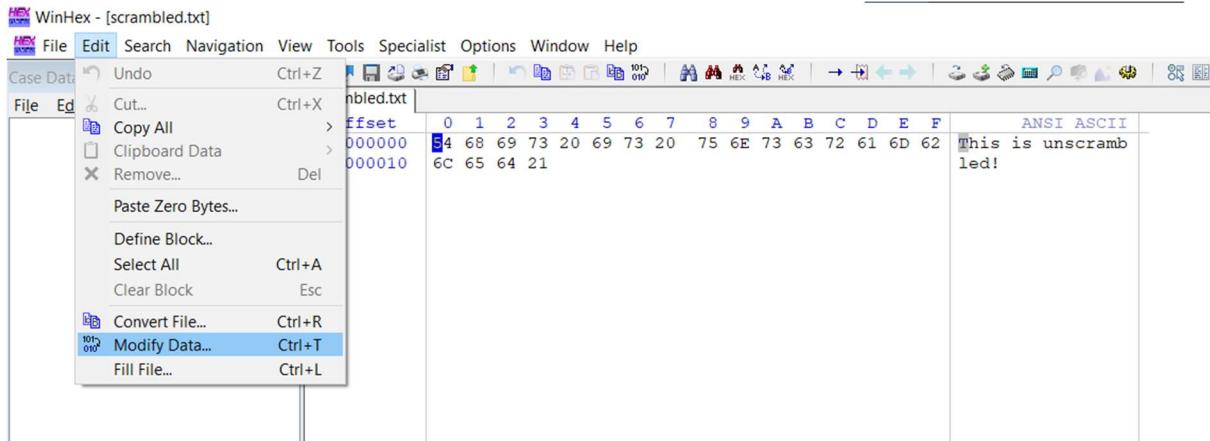
WinHex can be used for this.

First, open a file in WinHex. Here, a text file is opened as shown.

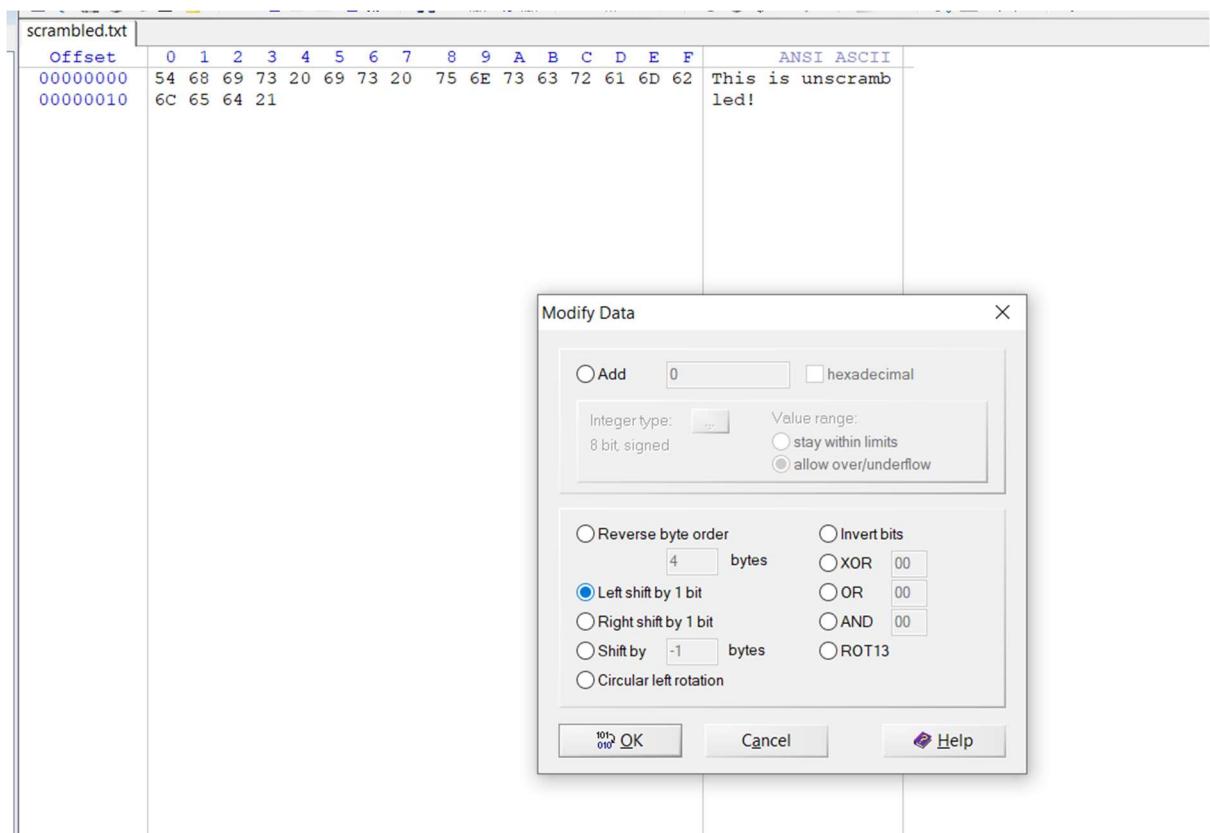
The screenshot shows the WinHex application interface. The menu bar includes File, Edit, Search, Navigation, View, Tools, Specialist, Options, Window, and Help. The toolbar contains various icons for file operations. The main window displays a hex dump of a file named 'scrambled.txt'. The left pane shows the file structure with sections for 'Case Data' and 'File'. The right pane shows the hex dump with columns for 'Offset', 'ANSI ASCII', and 'HEX'. The ASCII column shows the text 'This is unscrambled!'. The hex dump table is as follows:

Offset	0 1 2 3 4 5 6 7 8 9 A B C D E F	ANSI ASCII
00000000	54 68 69 73 20 69 73 20 75 6E 73 63 72 61 6D 62	This is unscrambled!
00000010	6C 65 64 21	

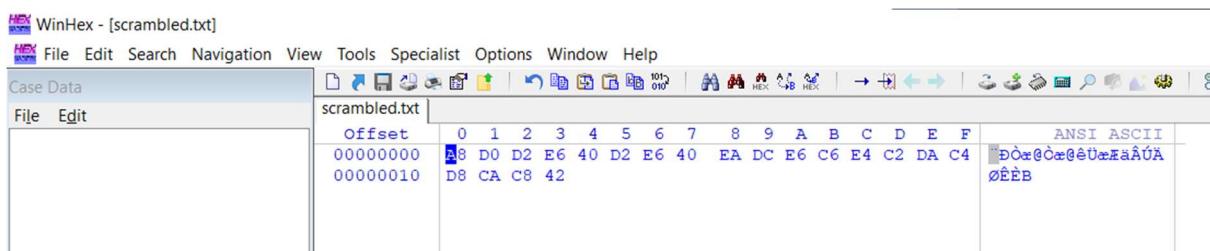
Now, go to Edit->Modify data.



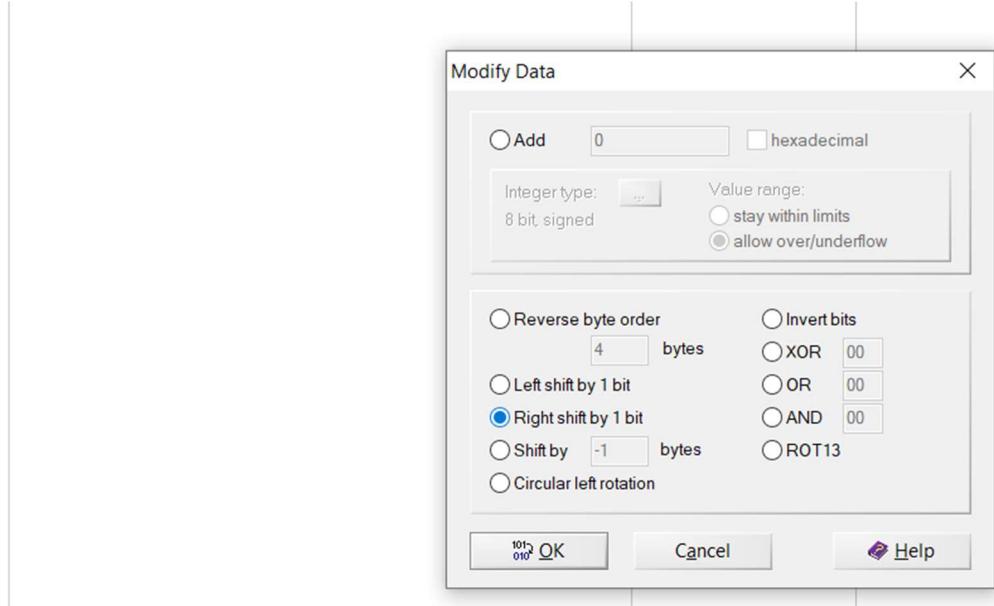
Now, select how you wish to shift the bits. Here I shift bits to the left once.



The file is now scrambled and unreadable.



To unscramble, shift the file bits in the other direction. Since we shifted it to the left by one place, we now shift the bits to the right by one place to unscramble.

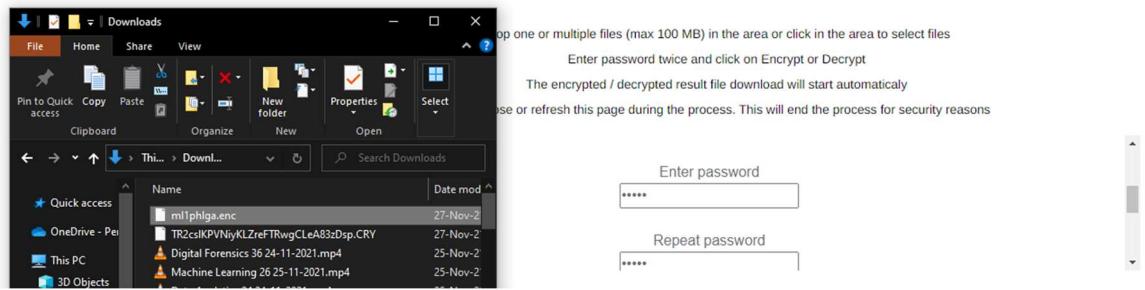


## Encryption

Visit any online website that does this. Here, file-encryptor.net was used as shown in image.

A screenshot of the file-encryptor.net website. The title bar shows 'File Encryptor - Store or send your files'. The URL bar shows 'file-encryptor.net'. The main header reads 'FILE ENCRYPTOR' and 'Store or send your files with sensitive content AES encrypted'. Below this is a dark area with instructions: 'Drop one or multiple files (max 100 MB) in the area or click in the area to select files', 'Enter password twice and click on Encrypt or Decrypt', 'The encrypted / decrypted result file download will start automatically', and 'Do not close or refresh this page during the process. This will end the process for security reasons'. There is a large rectangular file upload area with the placeholder text 'Drop files here to upload'.

Drop the file here then scroll down in the iframe to provide password for encryption. Then click Encrypt. A file of extension enc is created. To decrypt this, drop the enc file, enter the password used while encrypting then click Decrypt. The original image is gotten back.



## Password protection

Windows Home editions do not have this feature available for use. So, we will have to rely on third party tools or find some other way as follows:

1. Create a txt file and paste the following code:

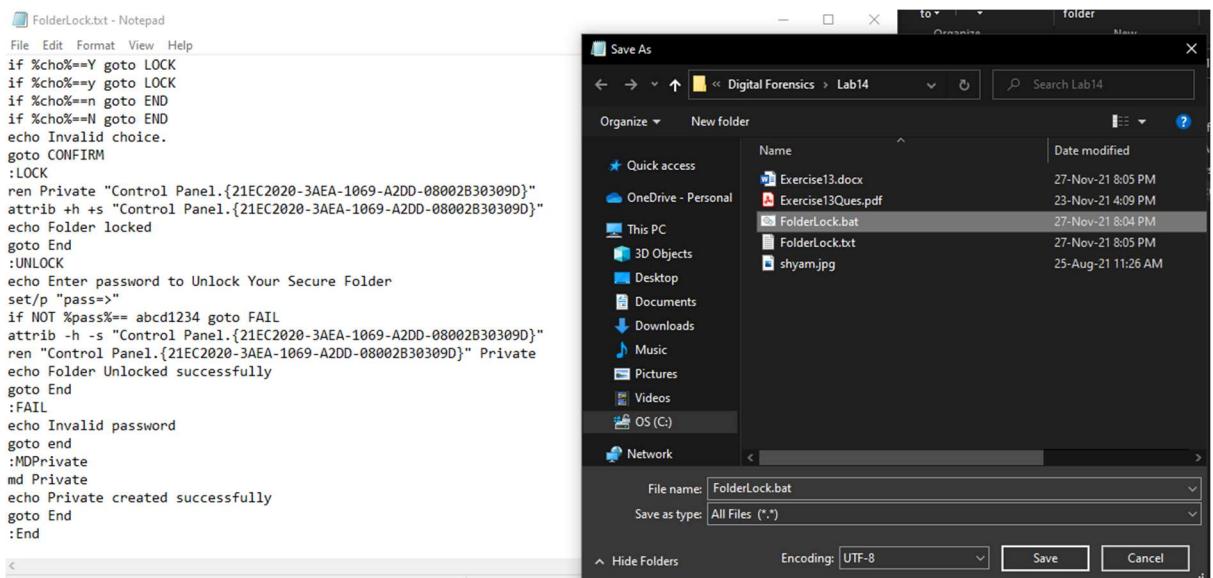
```
@ECHO OFF
if EXIST "Control Panel.{21EC2020-3AEA-1069-A2DD-08002B30309D}"
goto UNLOCK
if NOT EXIST Private goto MDPrivate
:CONFIRM
echo Are you sure to lock this folder? (Y/N)
set/p "cho=>"
if %cho%==Y goto LOCK
if %cho%==y goto LOCK
if %cho%==n goto END
if %cho%==N goto END
echo Invalid choice.
goto CONFIRM
:LOCK
ren Private "Control Panel.{21EC2020-3AEA-1069-A2DD-
08002B30309D}"
attrib +h +s "Control Panel.{21EC2020-3AEA-1069-A2DD-
08002B30309D}"
echo Folder locked
goto End
:UNLOCK
echo Enter password to Unlock Your Secure Folder
set/p "pass=>"
if NOT %pass%== YOUR-PASSWORD goto FAIL
attrib -h -s "Control Panel.{21EC2020-3AEA-1069-A2DD-08002B30309D}"
```

```

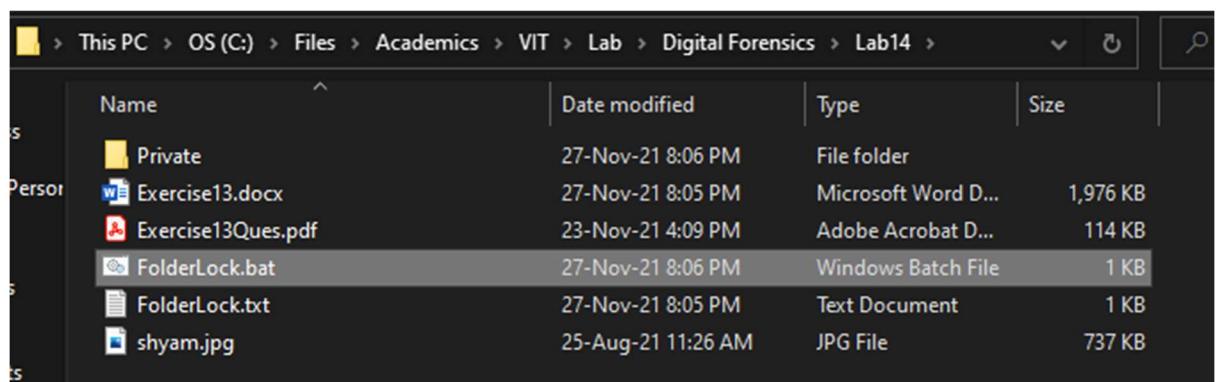
ren "Control Panel.{21EC2020-3AEA-1069-A2DD-08002B30309D}"
Private
echo Folder Unlocked successfully
goto End
:FAIL
echo Invalid password
goto end
:MDPrivate
md Private
echo Private created successfully
goto End
:End

```

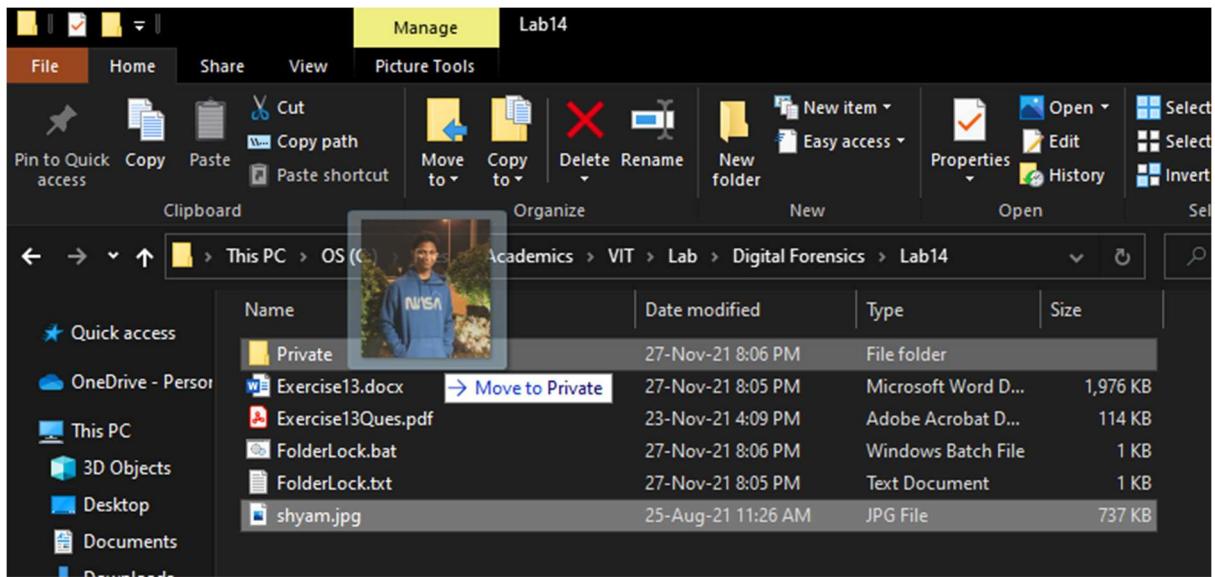
2. In the code above, replace 'YOUR PASSWORD' with the password you wish to use.
3. Then save the txt file as a .bat file.



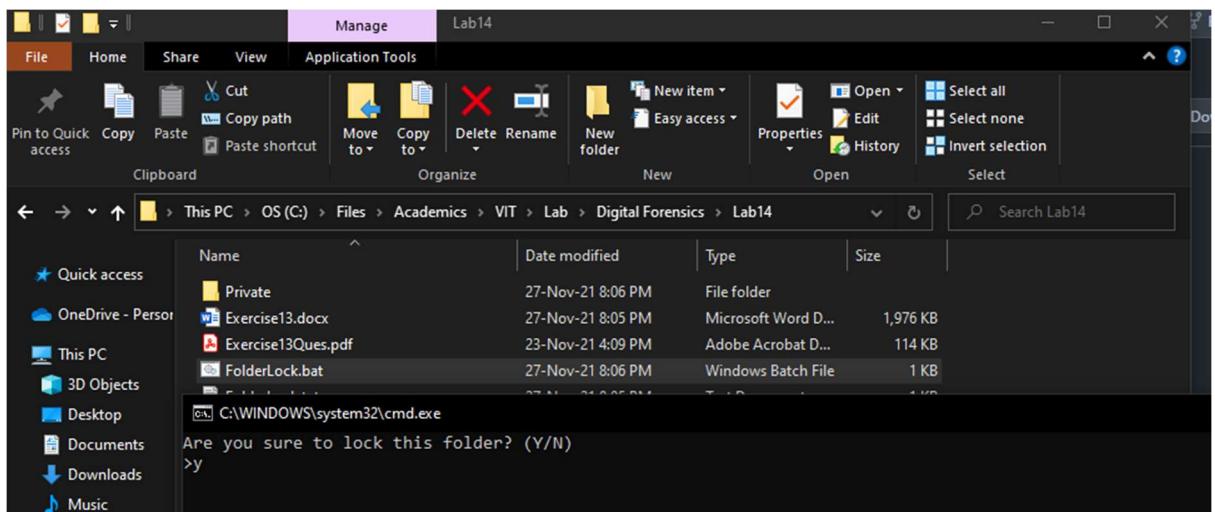
4. Now, double click on the bat file. A folder called 'Private' will be created.



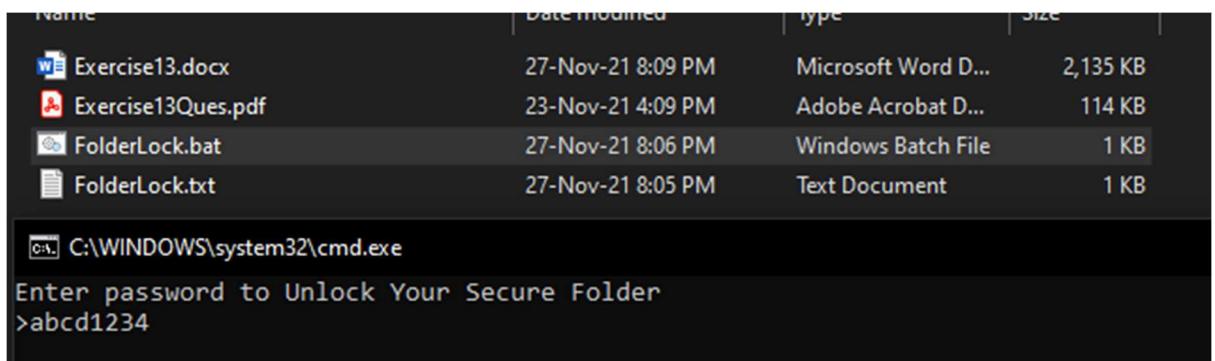
5. Drag and drop the files you want to lock into this 'Private' folder.



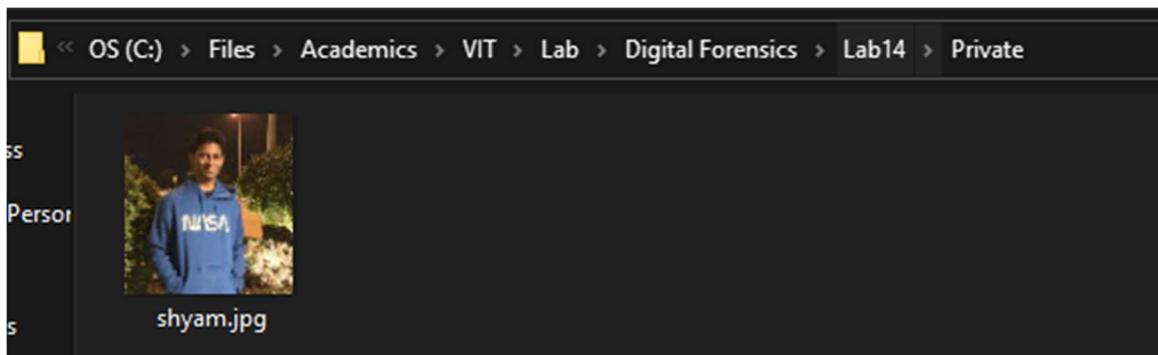
6. Then, again double click on the bat file. A prompt like this will be presented. Press 'y' and Enter.



7. The Private folder now disappears. To view the folder and its contents again, double click on the bat file and enter the password.



8. The folder and its contents are now visible again.



## **OBSERVATIONS**

In the methods used above, files are either completely hidden from view or were transformed into a form or file type that is not their own, rendering them unreadable. In some methods, the data within the file is scrambled to render them as nonsense to anyone else.

## **CONCLUSION**

Thus, we have explored the various ways to hide files.

## DIGITAL FORENSICS LAB

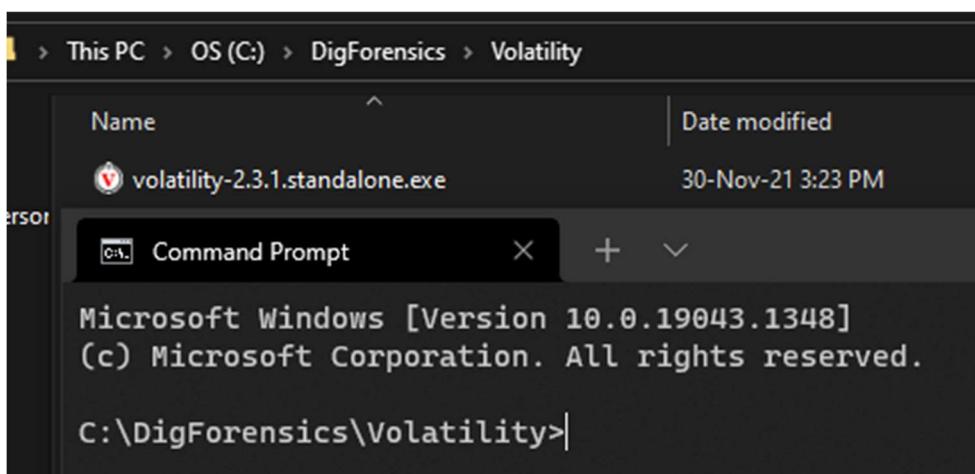
Exercise 14	
Name	S Shyam Sundaram
Registration Number	19BCE1560
Slot	L39+L40
Faculty	Dr. Seshu Babu Pulagara
Date	30 <sup>th</sup> November, 2021

### AIM

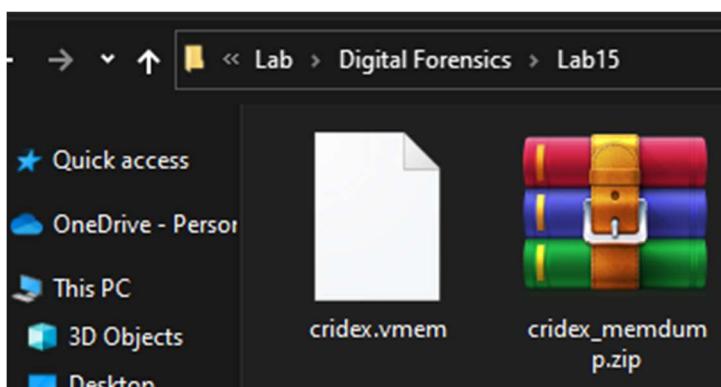
Employing memory forensic tool to analyse a memory dump.

### PROCEDURE

1. Go to the folder where Volatility is downloaded and open the terminal there.



2. Download a memory image file or create your own using 'Magnet RAM Capture'. Here I have downloaded a memory file with the Cridex malware.



- Now run the following command to get more info about the memory file.

```
C:\DigForensics\Volatility>volatility-2.3.1.standalone.exe -f "C:\Files\Academics\VIT\Lab\Digital Forensics\Lab15\cridex.vmem" imageinfo
Volatility Foundation Volatility Framework 2.3.1
Determining profile based on KDBG search...

Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
    AS Layer1 : IA32PagedMemoryPae (Kernel AS)
    AS Layer2 : FileAddressSpace (C:\Files\Academics\VIT\Lab\Digital Forensics\Lab15\cridex.vmem)
    PAE type : PAE
        DTB : 0x2fe000L
        KDBG : 0x80545ae0L
    Number of Processors : 1
    Image Type (Service Pack) : 3
        KPCR for CPU 0 : 0xffffd0000L
        KUSER_SHARED_DATA : 0xffffd0000L
    Image date and time : 2012-07-22 02:45:08 UTC+0000
    Image local date and time : 2012-07-21 22:45:08 -0400

C:\DigForensics\Volatility>
```

This gives us a list of suggested profiles to use for the cridex.vmem image.

- Now we see what were the processes running in the memory with the following command.

```
C:\DigForensics\Volatility>volatility-2.3.1.standalone.exe -f cridex.vmem --profile=WinXPSP2x86 pslist
Volatility Foundation Volatility Framework 2.3.1
Offset(V) Name PID PPID Thds Hnds Sess Wow64 Start Exit
0x823c89c8 System 4 0 53 240 ----- 0 2012-07-22 02:42:31 UTC+0000
0x822f1020 smss.exe 368 4 3 19 ----- 0 2012-07-22 02:42:32 UTC+0000
0x822a0598 csrss.exe 584 368 9 326 0 0 2012-07-22 02:42:32 UTC+0000
0x82298700 winlogon.exe 608 368 23 519 0 0 2012-07-22 02:42:32 UTC+0000
0x81e2ab28 services.exe 652 608 16 243 0 0 2012-07-22 02:42:32 UTC+0000
0x81e2a3b8 lsass.exe 664 608 24 330 0 0 2012-07-22 02:42:32 UTC+0000
0x82311360 svhost.exe 824 652 20 194 0 0 2012-07-22 02:42:33 UTC+0000
0x81e29ab8 svchost.exe 908 652 9 226 0 0 2012-07-22 02:42:33 UTC+0000
0x823001d0 svhost.exe 1004 652 64 1118 0 0 2012-07-22 02:42:33 UTC+0000
0x821dfda0 svhost.exe 1056 652 5 60 0 0 2012-07-22 02:42:33 UTC+0000
0x82295650 svhost.exe 1220 652 15 197 0 0 2012-07-22 02:42:35 UTC+0000
0x821dea70 explorer.exe 1484 1464 17 415 0 0 2012-07-22 02:42:36 UTC+0000
0x81eb17b8 spoolsv.exe 1512 652 14 113 0 0 2012-07-22 02:42:36 UTC+0000
0x81e7bda0 reader_sl.exe 1640 1484 5 39 0 0 2012-07-22 02:42:36 UTC+0000
0x820e8da0 alg.exe 788 652 7 104 0 0 2012-07-22 02:43:01 UTC+0000
0x821fcda0 wuauclt.exe 1136 1004 8 173 0 0 2012-07-22 02:43:46 UTC+0000
0x8205bda0 wuauclt.exe 1588 1004 5 132 0 0 2012-07-22 02:44:01 UTC+0000

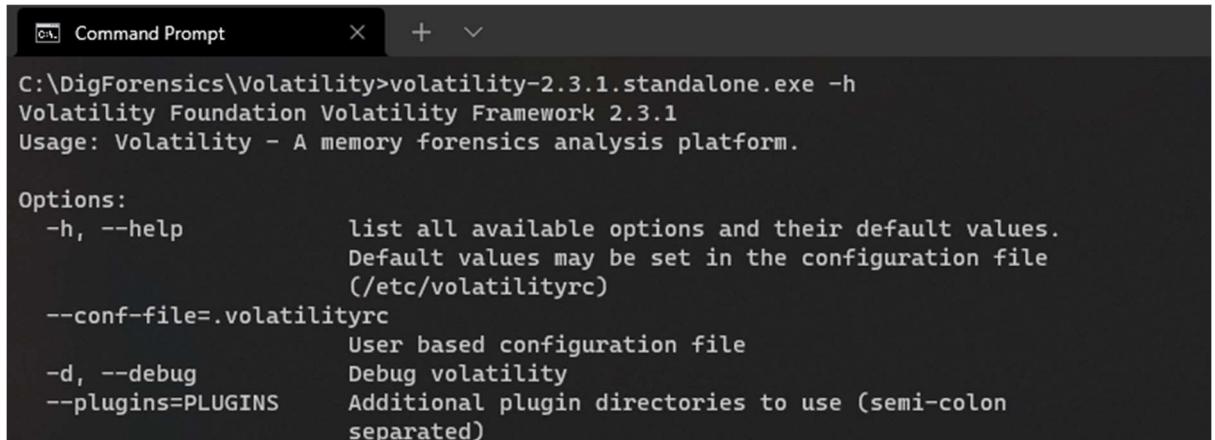
C:\DigForensics\Volatility>
```

- To see it in another format, we replace pslist with pstree.

```
C:\DigForensics\Volatility>volatility-2.3.1.standalone.exe -f cridex.vmem --profile=WinXPSP2x86 pstree
Volatility Foundation Volatility Framework 2.3.1
Name Pid PPid Thds Hnds Time
-----
0x823c89c8:System 4 0 53 240 1970-01-01 00:00:00 UTC+0000
. 0x822f1020:smss.exe 368 4 3 19 2012-07-22 02:42:31 UTC+0000
.. 0x82298700:winlogon.exe 608 368 23 519 2012-07-22 02:42:32 UTC+0000
... 0x81e2ab28:services.exe 652 608 16 243 2012-07-22 02:42:32 UTC+0000
.... 0x81e29ab8:svchost.exe 908 652 9 226 2012-07-22 02:42:33 UTC+0000
.... 0x823001d0:svhost.exe 1004 652 64 1118 2012-07-22 02:42:33 UTC+0000
.... 0x82295650:svhost.exe 1220 652 15 197 2012-07-22 02:42:35 UTC+0000
.... 0x821fcda0:wuauclt.exe 1136 1004 8 173 2012-07-22 02:43:46 UTC+0000
.... 0x82311360:svhost.exe 824 652 20 194 2012-07-22 02:42:33 UTC+0000
.... 0x820e8da0:alg.exe 788 652 7 104 2012-07-22 02:43:01 UTC+0000
.... 0x81e7bda0:wuauclt.exe 1588 1004 5 132 2012-07-22 02:44:01 UTC+0000
... 0x822a0598:csrss.exe 584 368 9 326 2012-07-22 02:42:32 UTC+0000
0x821dea70:explorer.exe 1484 1464 17 415 2012-07-22 02:42:36 UTC+0000
. 0x81e7bda0:reader_sl.exe 1640 1484 5 39 2012-07-22 02:42:36 UTC+0000

C:\DigForensics\Volatility>
```

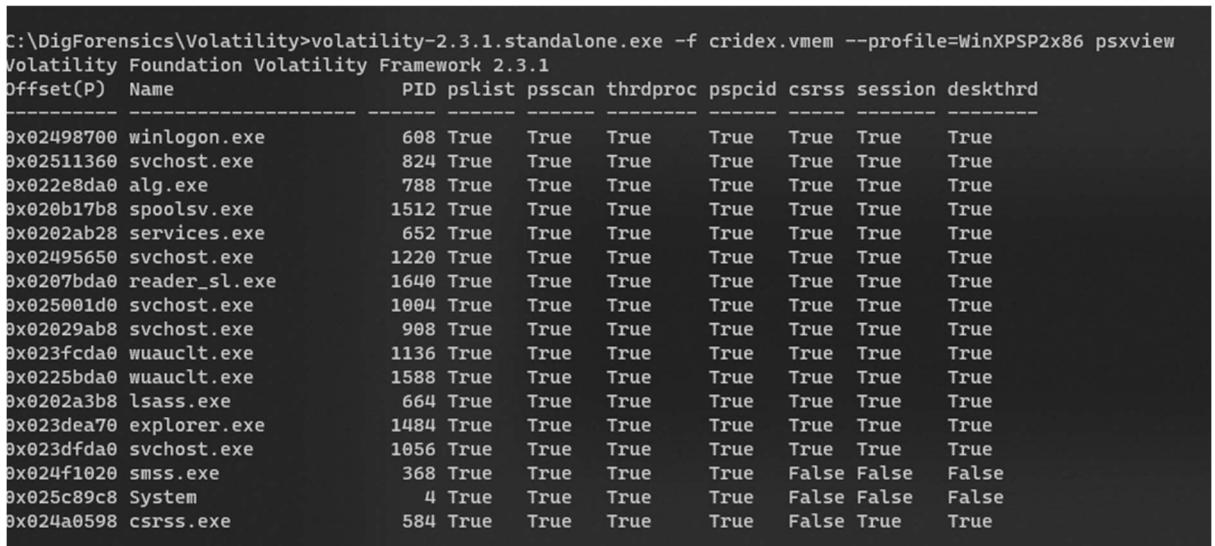
6. To see more about the tool's options and get some help, we execute this:



```
C:\DigForensics\Volatility>volatility-2.3.1.standalone.exe -h
Volatility Foundation Volatility Framework 2.3.1
Usage: Volatility - A memory forensics analysis platform.

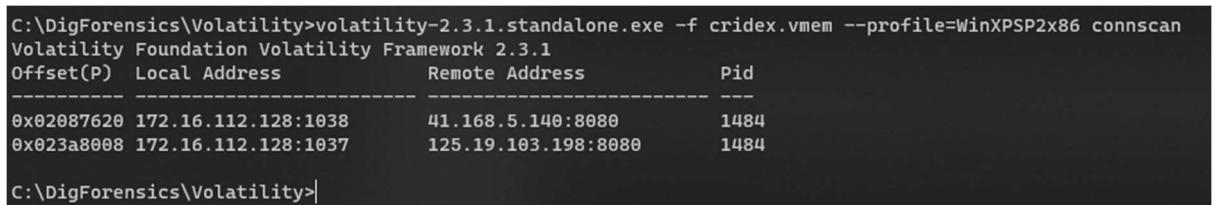
Options:
  -h, --help           list all available options and their default values.
                      Default values may be set in the configuration file
                      (/etc/volatilityrc)
  --conf-file=.volatilityrc
                      User based configuration file
  -d, --debug          Debug volatility
  --plugins=PLUGINS    Additional plugin directories to use (semi-colon
                      separated)
```

7. To uncover any hidden process, use psxview as below.



Offset(P)	Name	PID	pslist	psscan	thrdproc	pspcid	csrss	session	deskthrd
0x02498700	winlogon.exe	608	True	True	True	True	True	True	True
0x02511360	svchost.exe	824	True	True	True	True	True	True	True
0x022e8da0	alg.exe	788	True	True	True	True	True	True	True
0x020b17b8	spoolsv.exe	1512	True	True	True	True	True	True	True
0x0202ab28	services.exe	652	True	True	True	True	True	True	True
0x02495650	svchost.exe	1220	True	True	True	True	True	True	True
0x0207bda0	reader_sl.exe	1640	True	True	True	True	True	True	True
0x025001d0	svchost.exe	1004	True	True	True	True	True	True	True
0x02029ab8	svchost.exe	908	True	True	True	True	True	True	True
0x023fcda0	wuauctl.exe	1136	True	True	True	True	True	True	True
0x0225bda0	wuauctl.exe	1588	True	True	True	True	True	True	True
0x0202a3b8	lsass.exe	664	True	True	True	True	True	True	True
0x023dea70	explorer.exe	1484	True	True	True	True	True	True	True
0x023dfda0	svchost.exe	1056	True	True	True	True	True	True	True
0x024f1020	smss.exe	368	True	True	True	False	False	False	False
0x025c89c8	System	4	True	True	True	False	False	False	False
0x024a0598	csrss.exe	584	True	True	True	False	True	True	True

8. To check the running open TCP connections, we can use connscan.



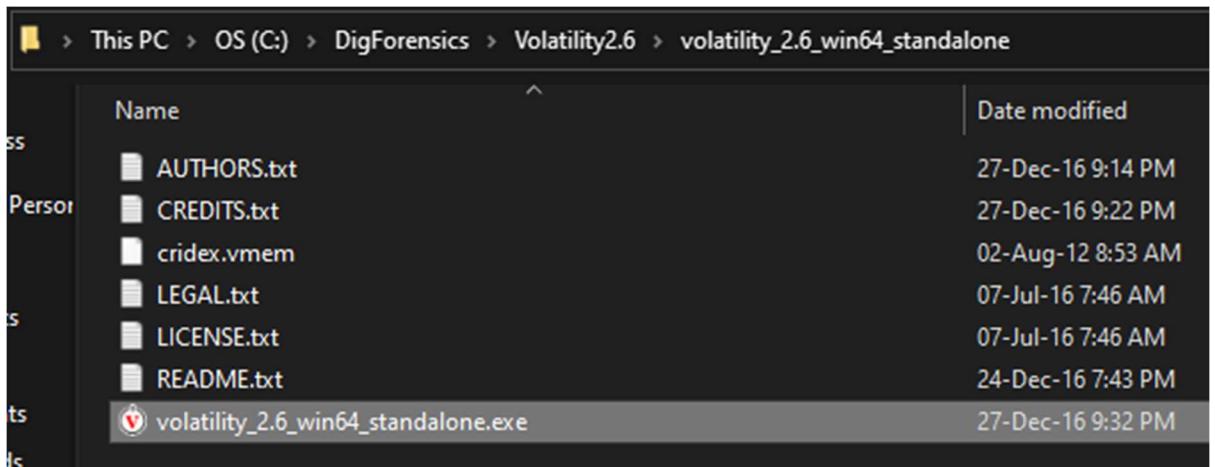
Offset(P)	Local Address	Remote Address	Pid
0x02087620	172.16.112.128:1038	41.168.5.140:8080	1484
0x023a8008	172.16.112.128:1037	125.19.103.198:8080	1484

9. To view both TCP and UDP connections, use 'sockets'

```
C:\DigForensics\Volatility>volatility-2.3.1.standalone.exe -f cridex.vmem --profile=WinXPSP2x86 socket
Volatility Foundation Volatility Framework 2.3.1
Offset(V)      PID  Port Proto Protocol      Address      Create Time
-----  -----
0x81ddb780    664   500   17 UDP       0.0.0.0      2012-07-22 02:42:53 UTC+0000
0x82240d08    1484   1038  6 TCP       0.0.0.0      2012-07-22 02:44:45 UTC+0000
0x81dd7618    1220   1900  17 UDP       172.16.112.128 2012-07-22 02:43:01 UTC+0000
0x82125610    788    1028  6 TCP       127.0.0.1      2012-07-22 02:43:01 UTC+0000
0x8219cc08     4     445   6 TCP       0.0.0.0      2012-07-22 02:42:31 UTC+0000
0x81ec23b0    908    135   6 TCP       0.0.0.0      2012-07-22 02:42:33 UTC+0000
0x82276878     4     139   6 TCP       172.16.112.128 2012-07-22 02:42:38 UTC+0000
0x82277460     4     137   17 UDP      172.16.112.128 2012-07-22 02:42:38 UTC+0000
0x81e76620    1004   123   17 UDP      127.0.0.1      2012-07-22 02:43:01 UTC+0000
0x82172808    664    0     255 Reserved  0.0.0.0      2012-07-22 02:42:53 UTC+0000
0x81e3f460     4     138   17 UDP      172.16.112.128 2012-07-22 02:42:38 UTC+0000
0x821f0630    1004   123   17 UDP      172.16.112.128 2012-07-22 02:43:01 UTC+0000
0x822cd2b0    1220   1900  17 UDP      127.0.0.1      2012-07-22 02:43:01 UTC+0000
0x82172c50    664    4500  17 UDP      0.0.0.0      2012-07-22 02:42:53 UTC+0000
0x821f0d00     4     445   17 UDP      0.0.0.0      2012-07-22 02:42:31 UTC+0000

C:\DigForensics\Volatility>
```

## 10. FOR THE FOLLOWING, USE VOLATILITY 2.6. This was saved in another folder called “Volatility2.6” in “DigForensics” folder.



To display the commandline arguments of each process, use cmdline:

```
C:\DigForensics\Volatility2.6\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone -f cridex.vmem cmdline
Volatility Foundation Volatility Framework 2.6
*****
System pid:        4
*****
smss.exe pid:     368
Command line : \SystemRoot\System32\smss.exe
*****
csrss.exe pid:    584
Command line : C:\WINDOWS\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,3072,512 Windows=On SubSystemType=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=winsrv:ConServerDllInitialization,2 ProfileControl=Off MaxRequestThreads=16
*****
winlogon.exe pid: 608
```

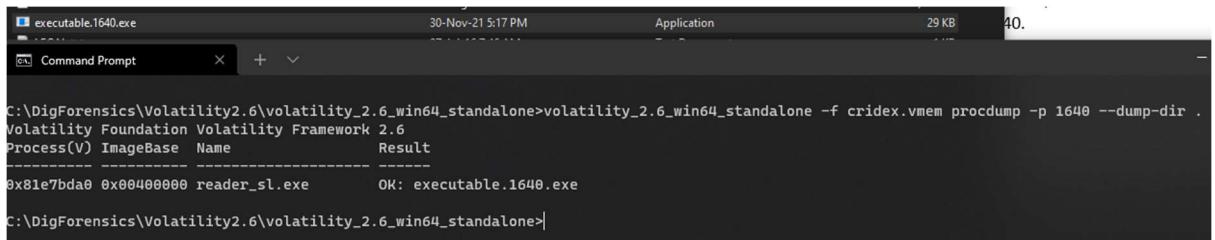
Scolling down we see the command line arguments and the path where reader\_sl was stored. It is found using its process id 1640.

```
*****
reader_sl.exe pid: 1640
Command line : "C:\Program Files\Adobe\Reader 9.0\Reader\Reader_sl.exe"
*****
alg.exe pid: 788
```

11. We now create a dump of this process and check it out. The command for this is given below using procdump and specifying the PID of the process.

**Command run:** volatility\_2.6\_win64\_standalone -f cridex.vmem procdump -p 1640 --dump-dir .

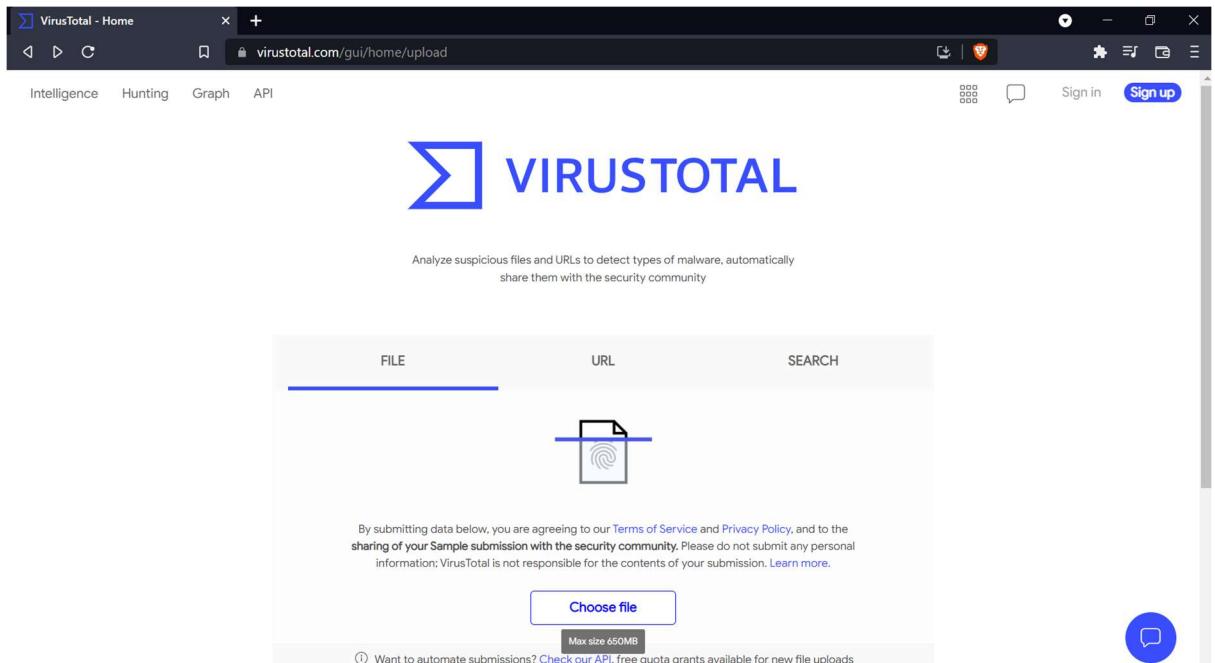
**NOTE: DO NOT DOUBLE CLICK OR RUN THIS NEWLY CREATED EXE/DUMP FILE!**



```
executable.1640.exe          30-Nov-21 5:17 PM      Application      29 KB      40.
Command Prompt      +  ▾

C:\DigForensics\Volatility2.6\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone -f cridex.vmem procdump -p 1640 --dump-dir .
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase Name             Result
-----
0x81e7bda0 0x00400000 reader_sl.exe     OK: executable.1640.exe
C:\DigForensics\Volatility2.6\volatility_2.6_win64_standalone>
```

12. Now, go to virustotal.com and upload this newly created exe file.



The screenshot shows two windows from the VirusTotal platform. The top window is a file upload interface where a memory dump file named 'volatility\_2.6\_win64\_standalone' has been uploaded. The bottom window is the detailed analysis page for the file '5b136147911b041f0126ce82df24c4e2c79553b65d3240ece2dcab4452dc5'. The analysis summary indicates 29 security vendors flagged it as malicious. The 'DETECTION' tab shows various vendor findings, all of which are Trojans or similar threats.

Vendor	Findings
Ad-Aware	Trojan.GenericKD.41512677
ALYac	Trojan.GenericKD.41512677
BitDefender	Trojan.GenericKD.41512677
Cyberason	Malicious.3f5a91
Emsisoft	Trojan.GenericKD.41512677 (B)
FireEye	Trojan.GenericKD.41512677
GData	Trojan.GenericKD.41512677
Alibaba	Trojan:Win32/Multiop.788dce0e
Arcabit	Trojan.Generic.D2796EE5
Comodo	Malware@#b2ihr9eixviv
Cylance	Unsafe
eScan	Trojan.GenericKD.41512677
Fortinet	PossibleThreat
Ikarus	Trojan.Win32.Patched

From the image above, we see that VirusTotal recognized this file as a Trojan malware. Thus, reader\_sl.exe is a malware.

## OBSERVATIONS

In the fifth image, we see that there is a process named “reader\_sl.exe” with “explorer.exe” as its parent process. Upon checking the connections and sockets, we see in image 8 that its parent process 1484 makes a connection to some location with address 41.168.140:8080. This is a bit suspicious as the name says it is a process of Adobe Reader but there is no reason why Adobe reader would have to make a connection to some remote location.

Thus, this process’s dump was made and uploaded to VirusTotal which recognized this file as a malware.

## CONCLUSION

We have used a memory image to figure out which process was the malware.