DIGITAL FORENSICS LAB

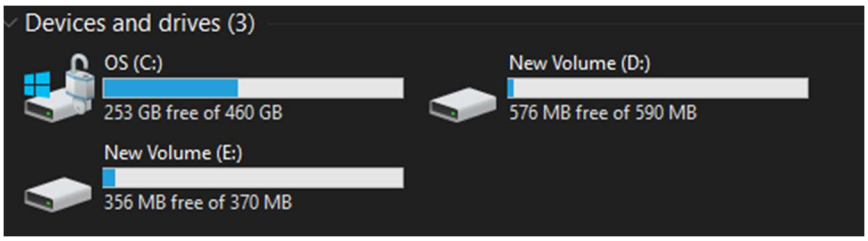
Exercise 13	
Name	S Shyam Sundaram
Registration Number	19BCE1560
Slot	L39+L40
Faculty	Dr. Seshu Babu Pulagara
Date	23rd November, 2021

AIM

Exploring various methods of data hiding.

Partition Hiding

The drive currently has partitions as shown below:



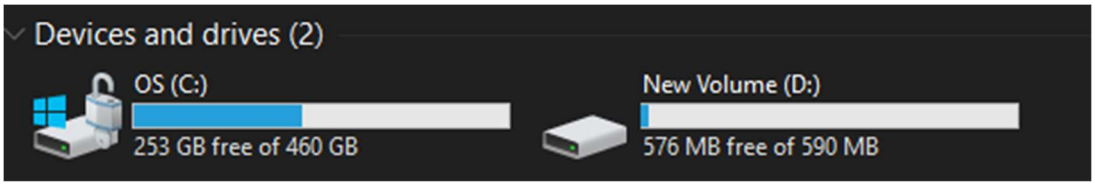
To hide partition E:, open the terminal in Admin mode. Then type in 'Diskpart' and press enter. Then enter 'list volume' to list the volumes present.

```
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         DELLSUPPORT  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.

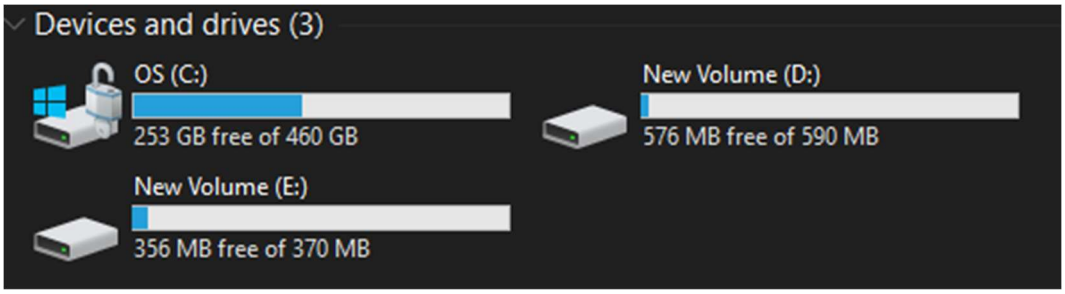
```
DISKPART> select volume 6

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.

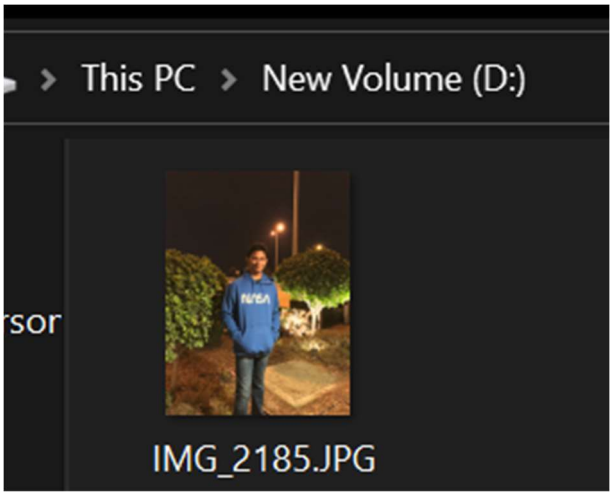
```
DISKPART> list volume

  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.
```



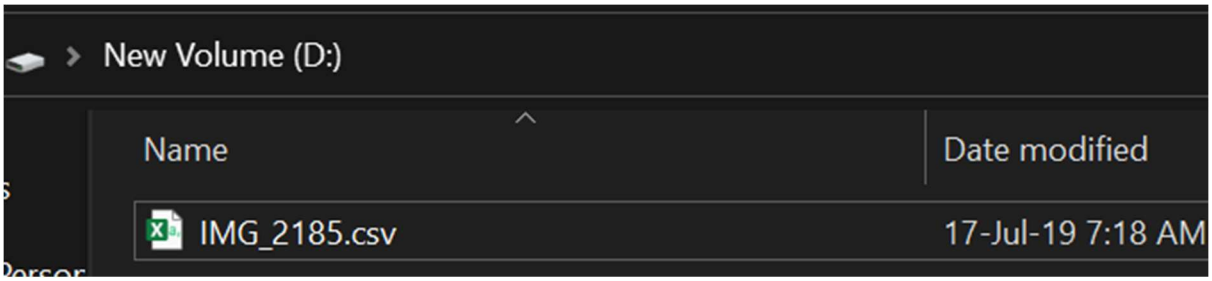
The partition is now unhidden.

Changing file extensions

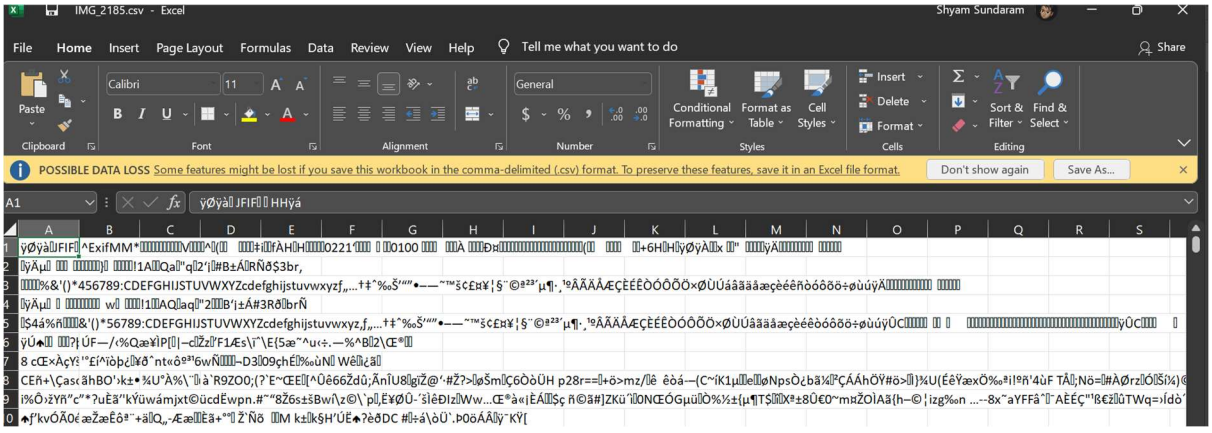
Here, we just change the extension of the file t be hidden. Say I wish to hide the JPG file below.



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



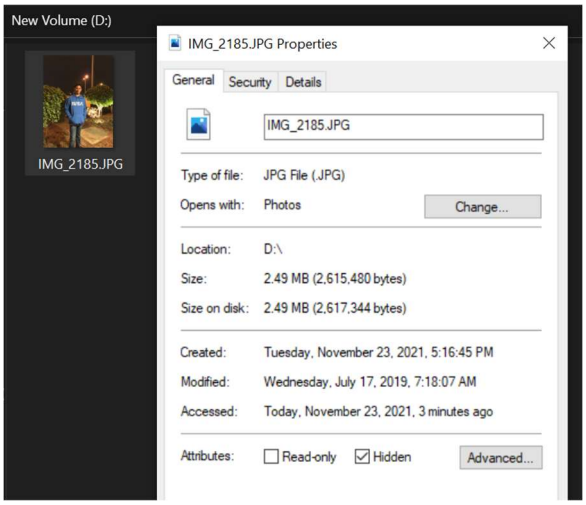
But the file would appear to be gibberish:

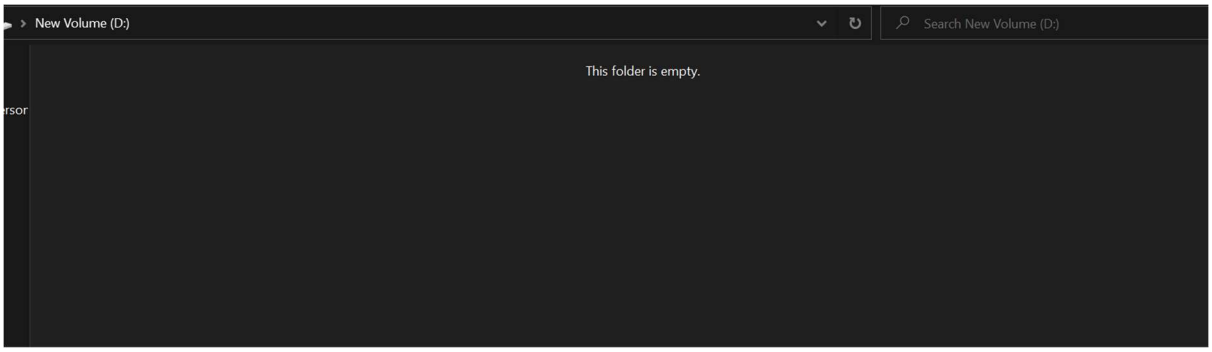


We can obtain it back by changing the extension back to JPG.

Setting File attributes to Hidden

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



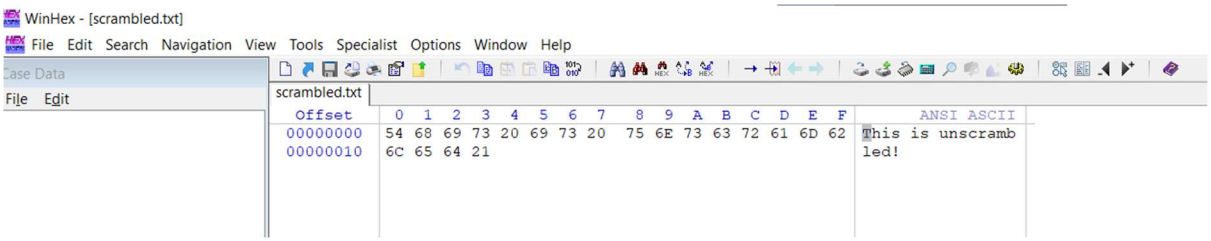


To unhide, click on View in file explorer, then select 'Hidden Item' in 'Show/Hide' group. The file will be visible and then the 'Hidden' attribute can be unchecked.

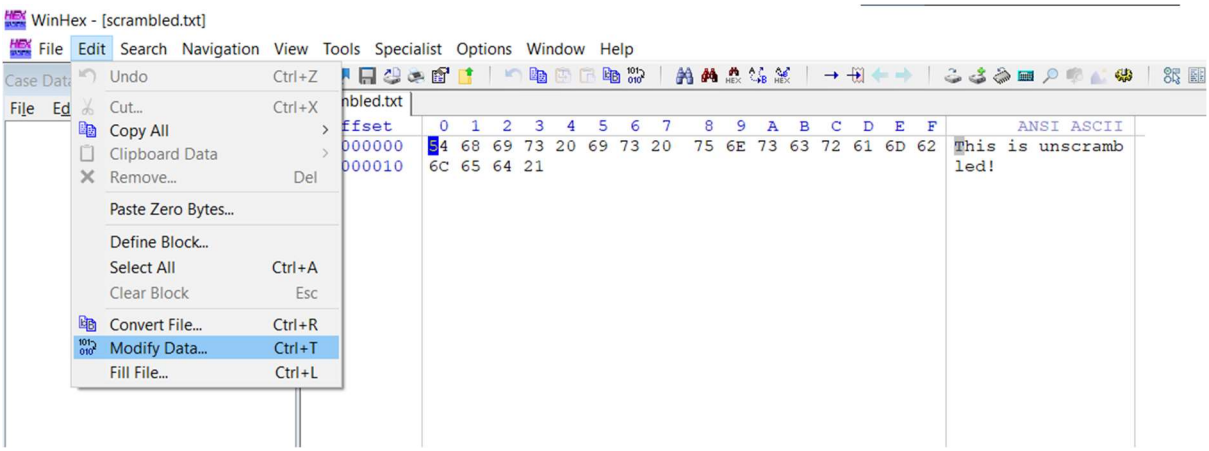
Bit Shifting

WinHex can be used for this.

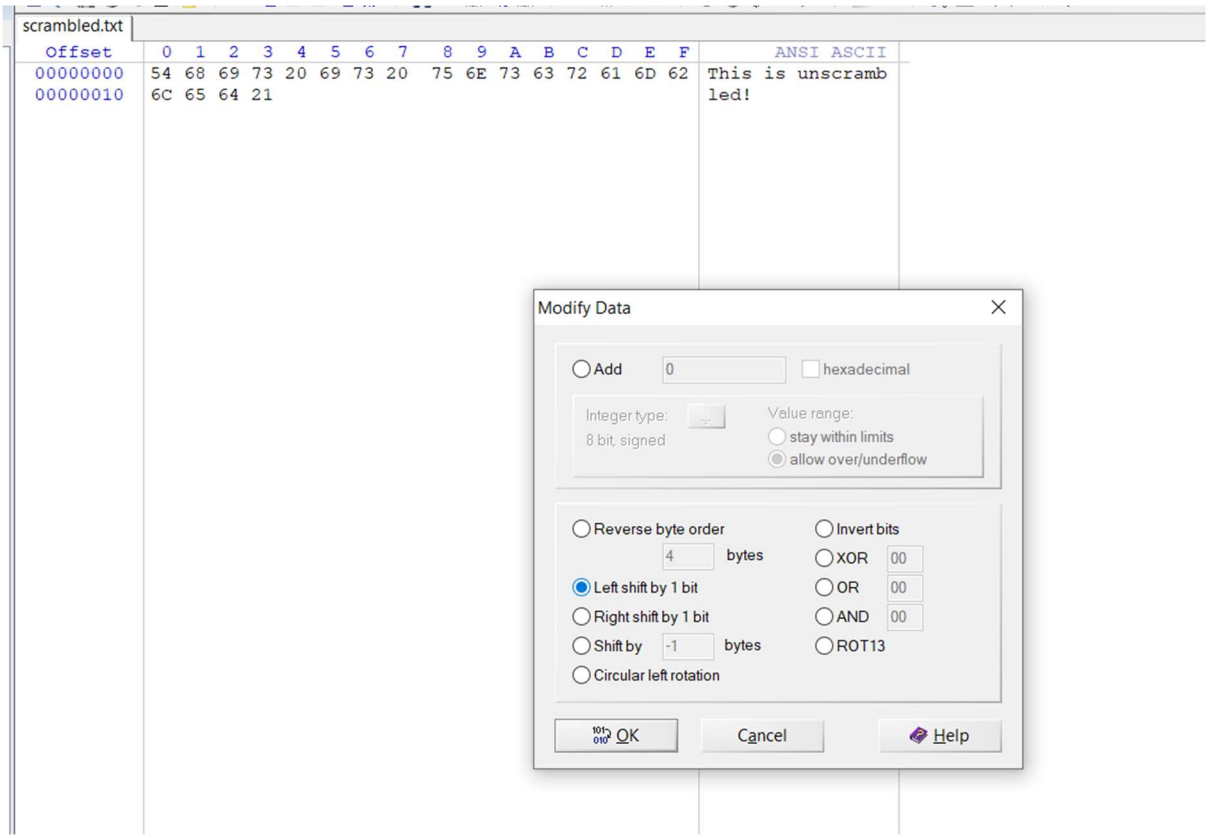
First, open a file in WinHex. Here, a text file is opened as shown.



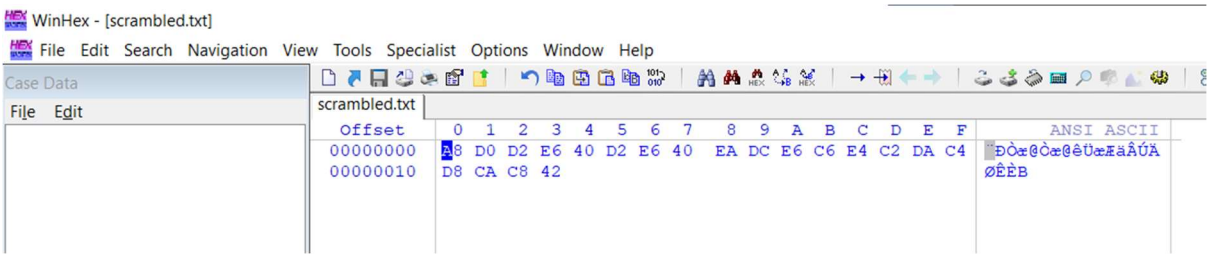
Now, go to Edit->Modify data.



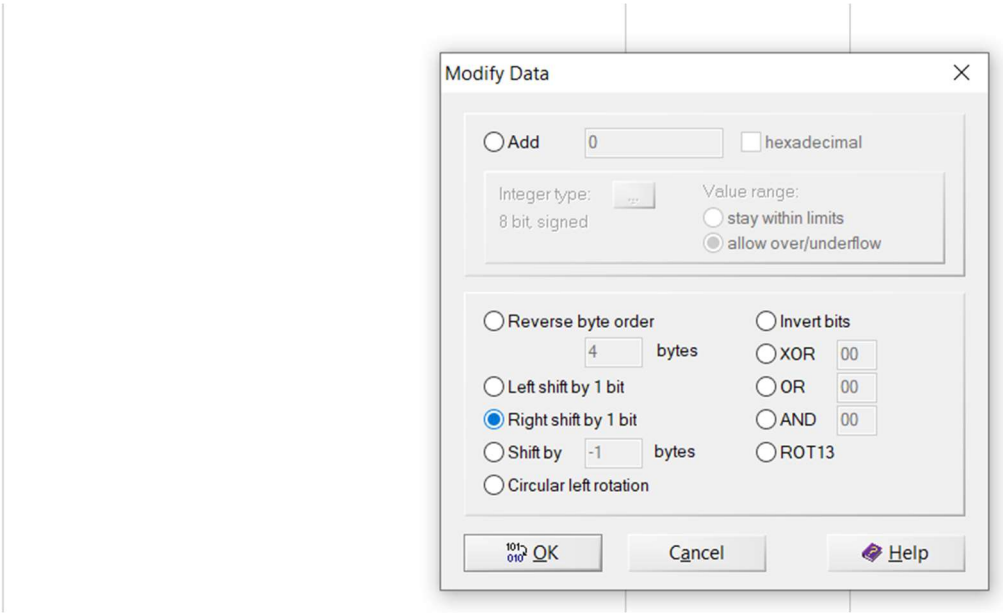
Now, select how you wish to shift the bits. Here I shift bits to the left once.



The file is now scrambled and unreadable.

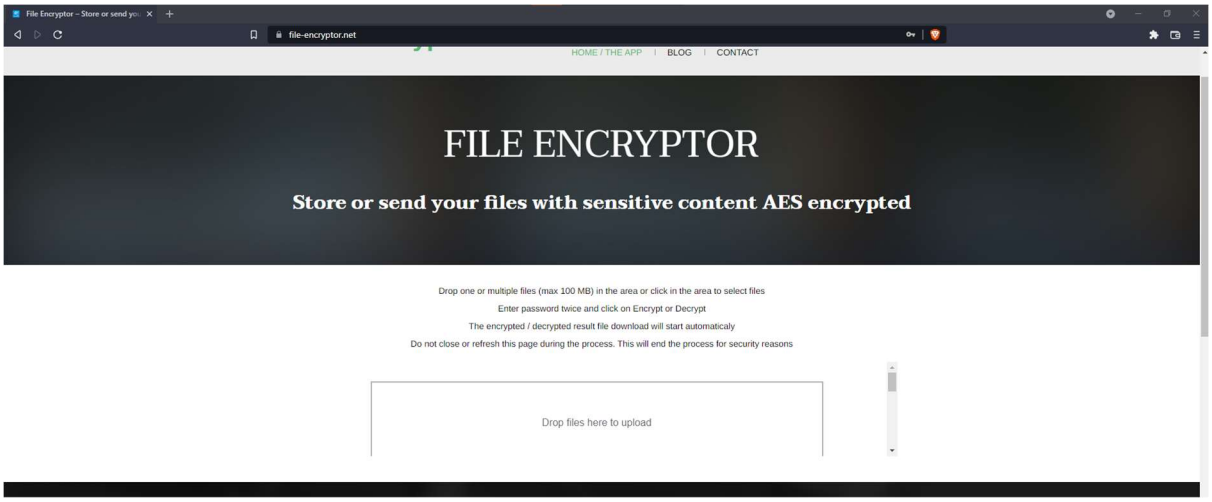


To unscramble, shift the file bits in the other direction. Since we shifted it to the left by one place, we now shift the bits to the right by one place to unscramble.

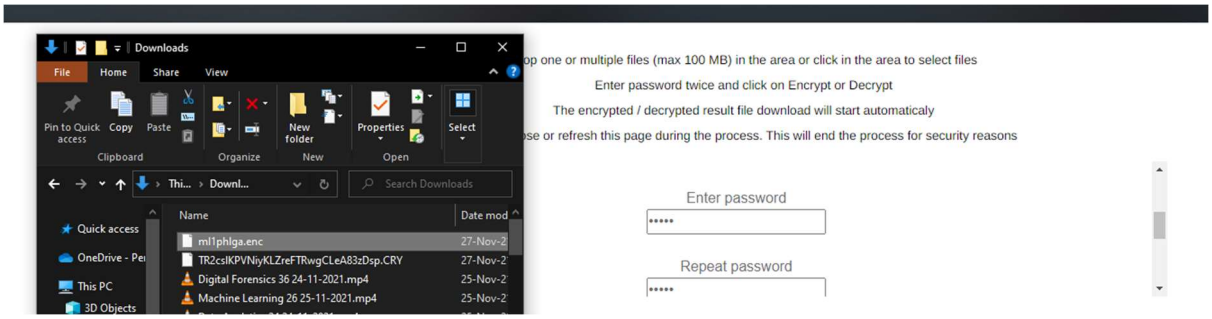


Encryption

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



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

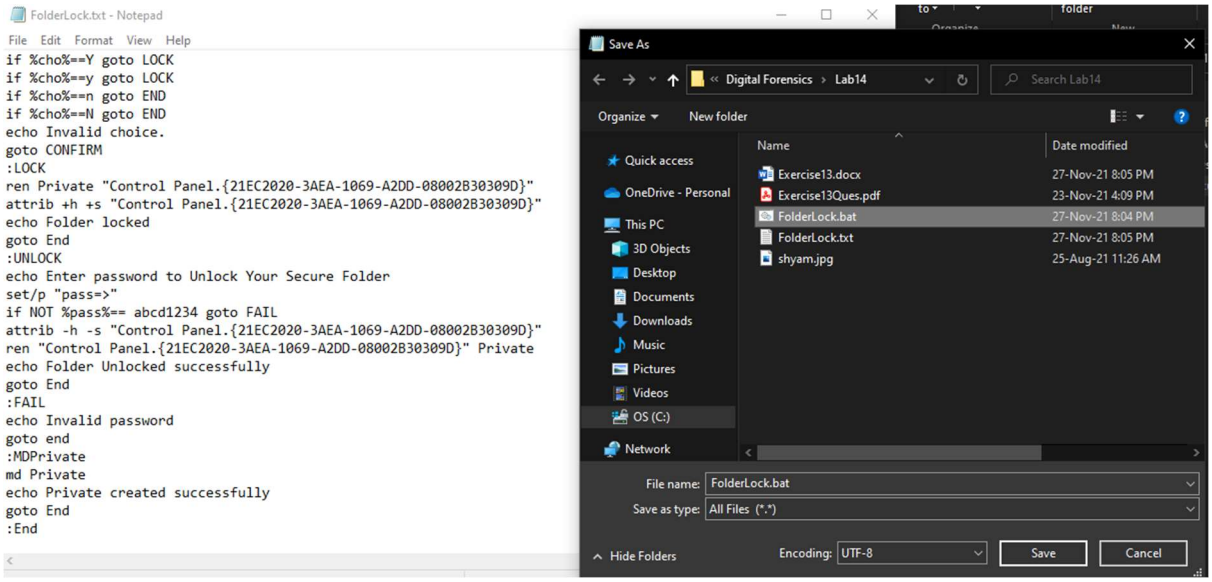
Windows Home editions do not have this feature available for use. So, we will have to rely on third party tools or find some other way as follows:

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

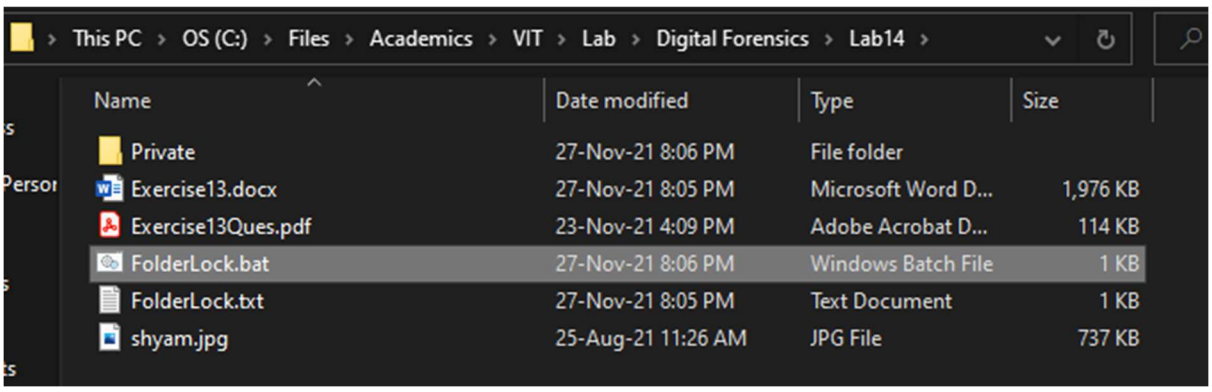
```
@ECHO OFF
if EXIST "Control Panel.{21EC2020-3AEA-1069-A2DD-08002B30309D}"
goto UNLOCK
if NOT EXIST Private goto MDPrivate
:CONFIRM
echo Are you sure to lock this folder? (Y/N)
set/p "cho=>"
if %cho%==Y goto LOCK
if %cho%==y goto LOCK
if %cho%==n goto END
if %cho%==N goto END
echo Invalid choice.
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
```

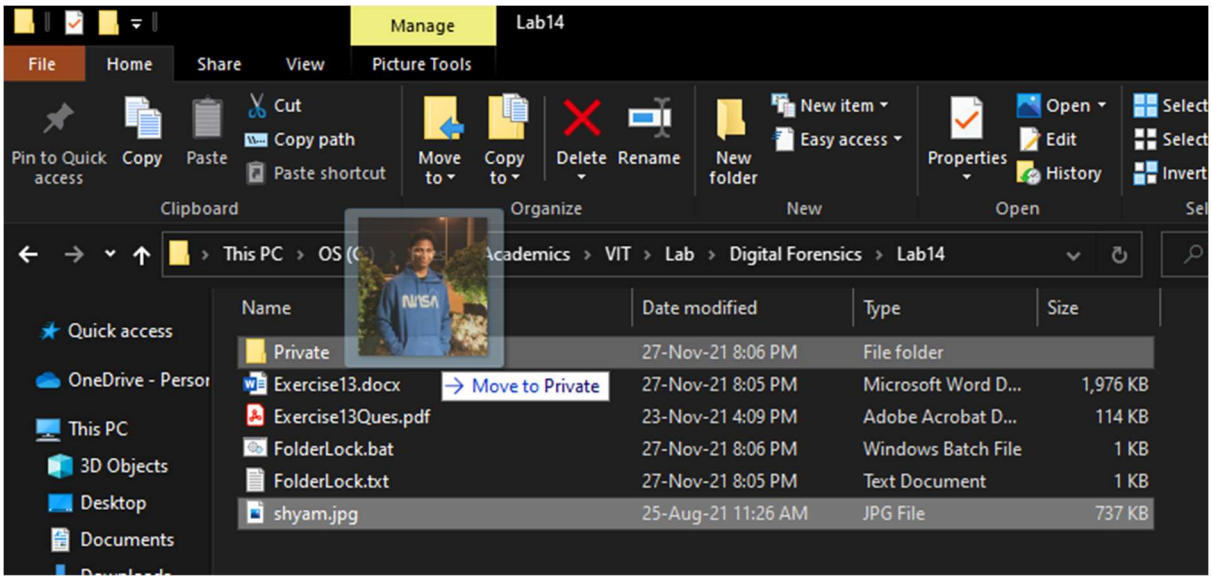
2. In the code above, replace ‘YOUR PASSWORD’ with the password you wish to use.
3. Then save the txt file as a .bat file.



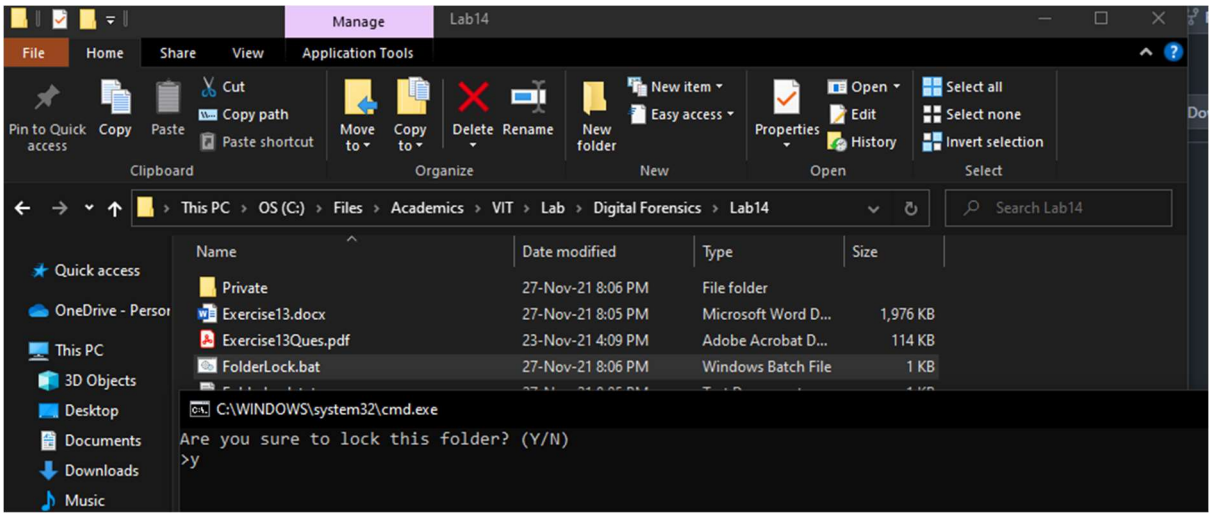
4. Now, double click on the bat file. A folder called ‘Private’ will be created.



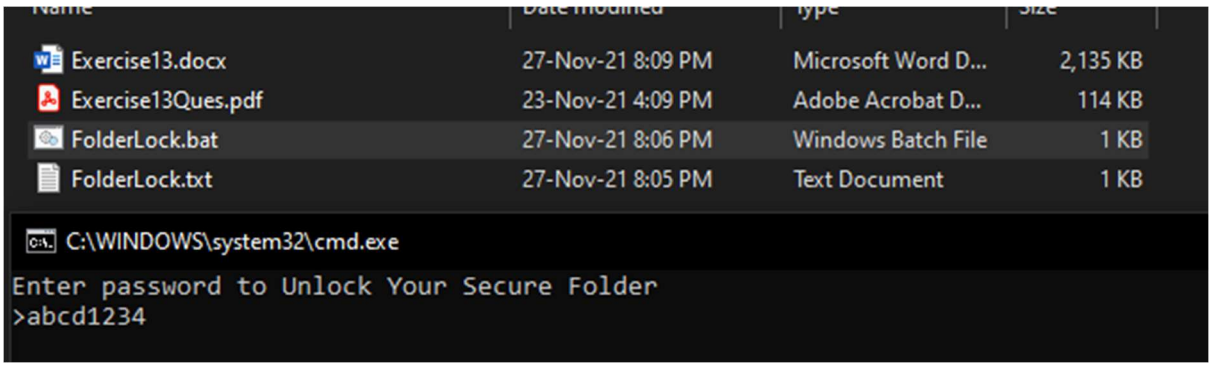
5. Drag and drop the files you want to lock into this ‘Private’ folder.



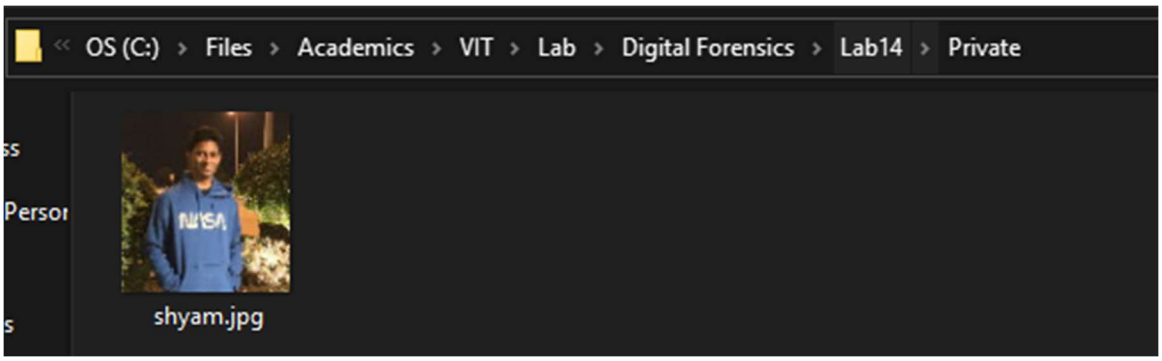
6. Then, again double click on the bat file. A prompt likthis will be presented. Press ‘y’ and Enter.



7. The Private folder now disappears. To view the folder and it’s 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

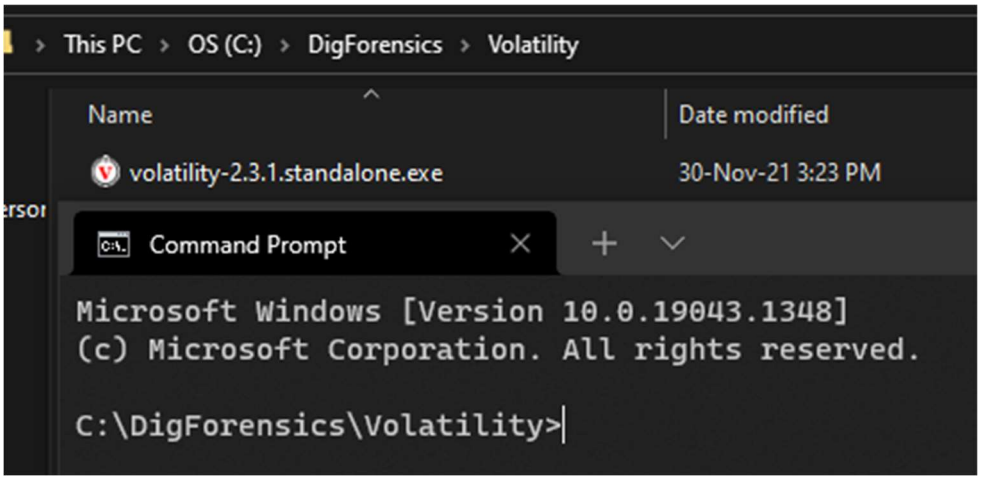
Exercise 14	
Name	S Shyam Sundaram
Registration Number	19BCE1560
Slot	L39+L40
Faculty	Dr. Seshu Babu Pulagara
Date	30th November, 2021

AIM

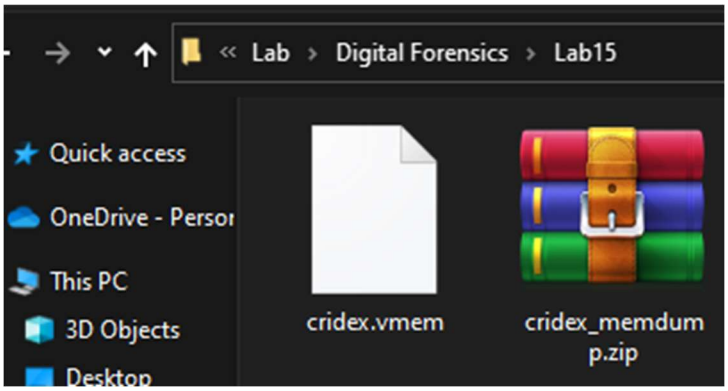
Employing memory forensic tool to analyse a memory dump.

PROCEDRE

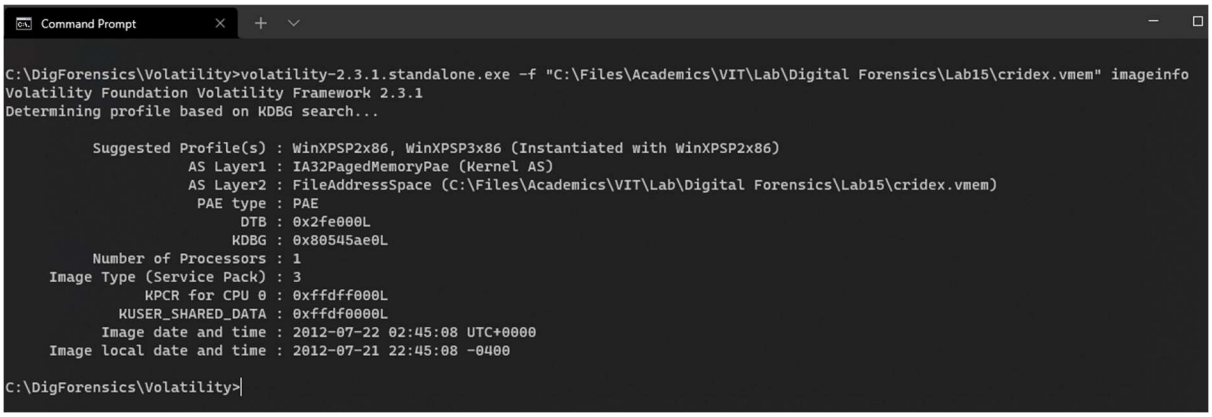
1. Go to the folder where Volatility is downloaded and open the terminal there.



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.



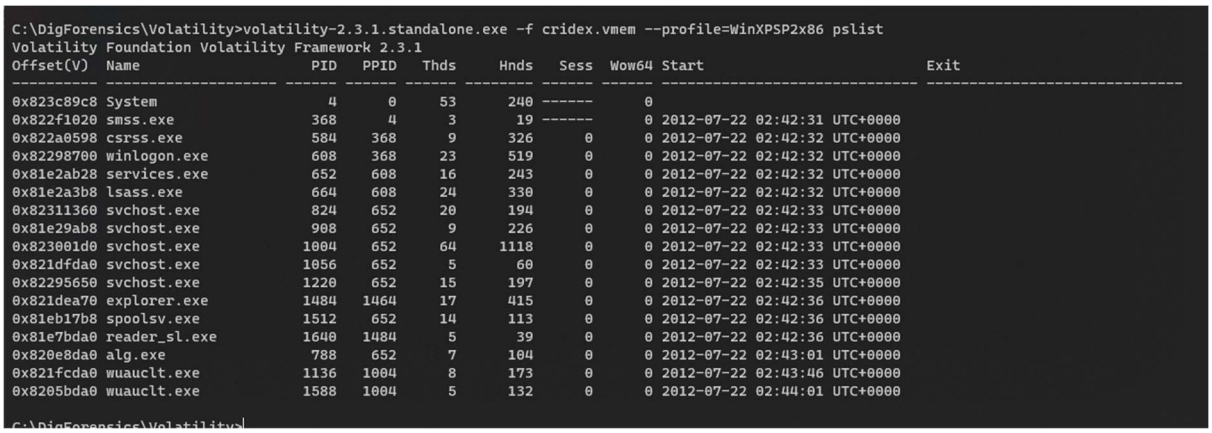
3. Now run the following command to get more info about the memory file.



```
Command Prompt

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 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           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>
```

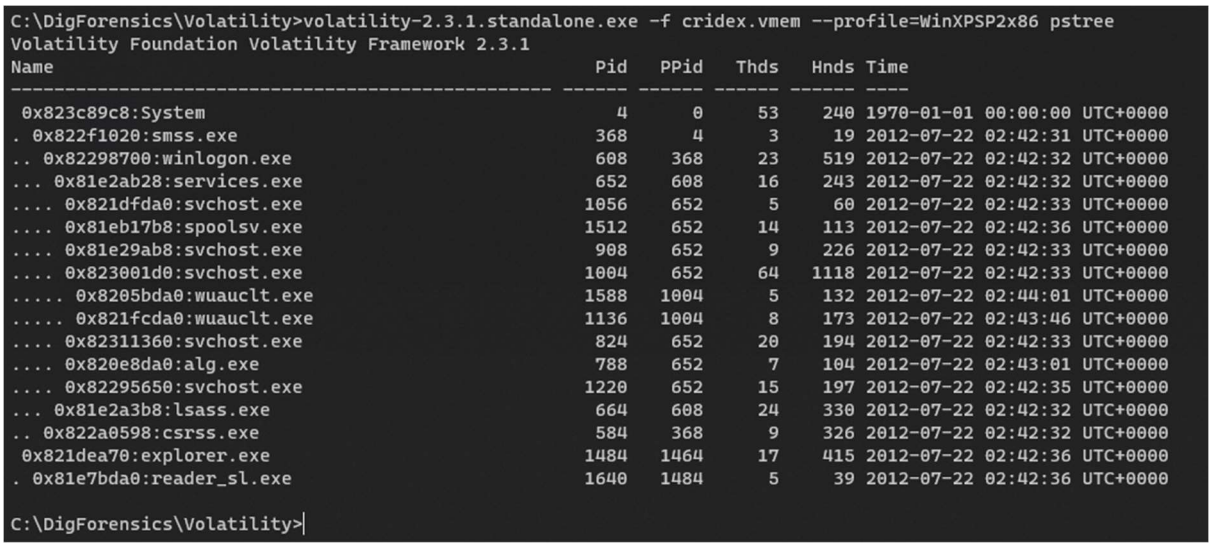
This gives us a list of suggested profiles to use for the cridex.vmem image.

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



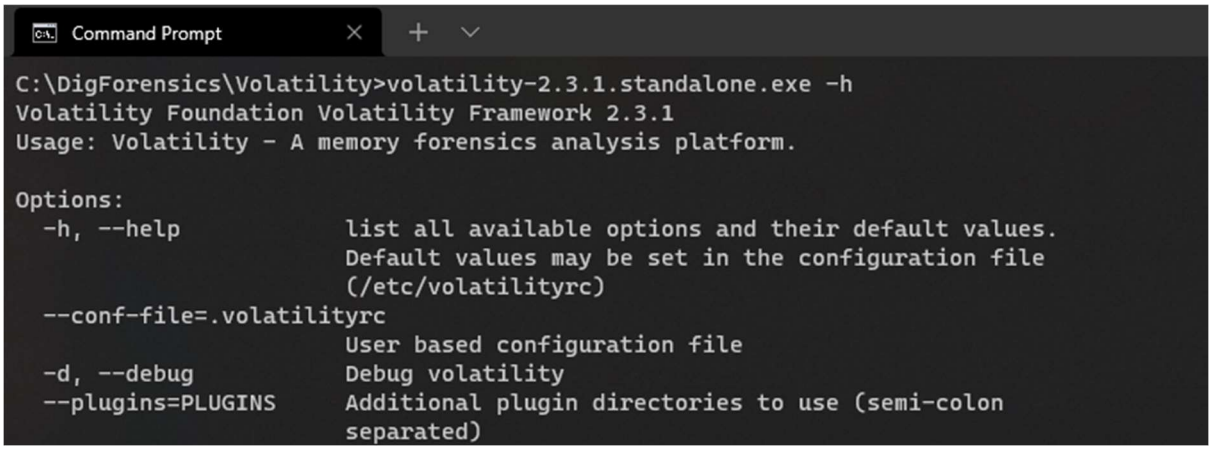
```
C:\DigForensics\Volatility>volatility-2.3.1.standalone.exe -f cridex.vmem --profile=WinXPSP2x86 pslist
Volatility Foundation Volatility Framework 2.3.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x823c89c8 System                    4      0     53      240 ------      0
0x822f1020 smss.exe                368      4      3       19 ------      0 2012-07-22 02:42:31 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 svchost.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 svchost.exe            1004    652     64     1118      0      0 2012-07-22 02:42:33 UTC+0000
0x821dfda0 svchost.exe            1056    652      5       60      0      0 2012-07-22 02:42:33 UTC+0000
0x82295650 svchost.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>
```

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

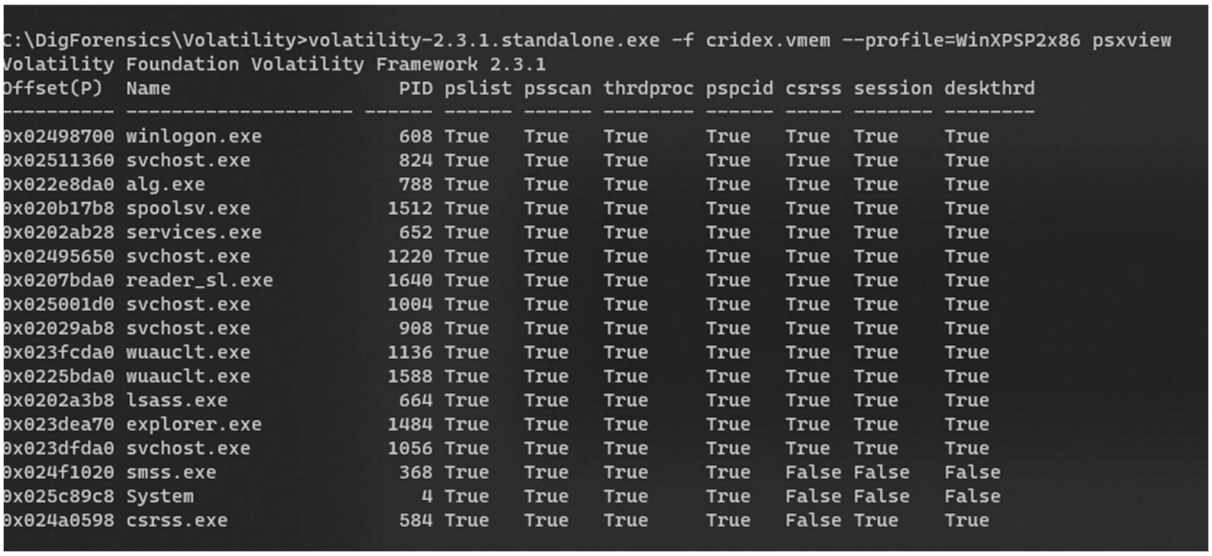


```
C:\DigForensics\Volatility>volatility-2.3.1.standalone.exe -f cridex.vmem --profile=WinXPSP2x86 pstree
Volatility Foundation Volatility Framework 2.3.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 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
.... 0x821dfda0:svchost.exe                          1056    652      5     60 2012-07-22 02:42:33 UTC+0000
.... 0x81eb17b8:spoolsv.exe                          1512    652     14    113 2012-07-22 02:42:36 UTC+0000
.... 0x81e29ab8:svchost.exe                           908    652      9    226 2012-07-22 02:42:33 UTC+0000
.... 0x823001d0:svchost.exe                          1004    652     64   1118 2012-07-22 02:42:33 UTC+0000
..... 0x8205bda0:wuauclt.exe                         1588   1004      5    132 2012-07-22 02:44:01 UTC+0000
..... 0x821fcda0:wuauclt.exe                         1136   1004      8    173 2012-07-22 02:43:46 UTC+0000
.... 0x82311360:svchost.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
.... 0x82295650:svchost.exe                          1220    652     15    197 2012-07-22 02:42:35 UTC+0000
... 0x81e2a3b8:lsass.exe                              664    608     24    330 2012-07-22 02:42:32 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>
```

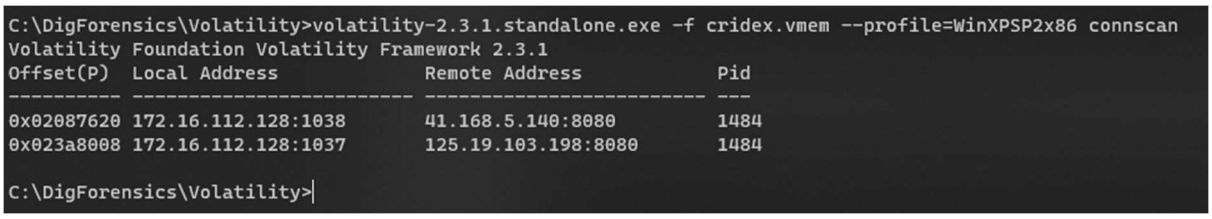
6. To see more about the tool's options and get some help, we execute this:



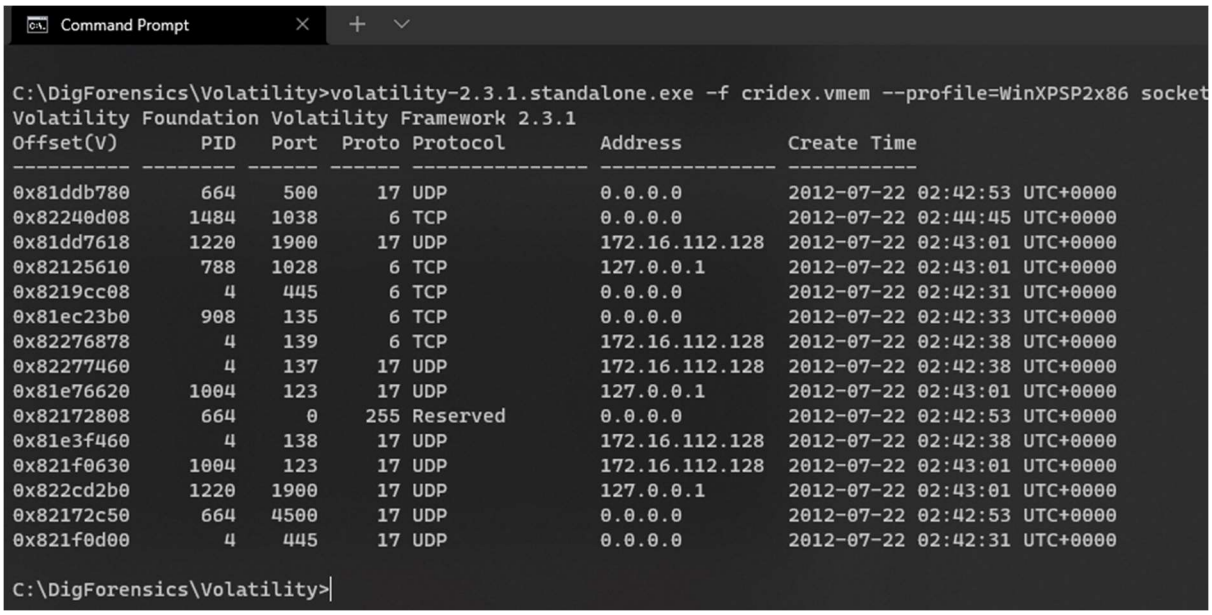
7. To uncover any hidden process, use psxview as below.



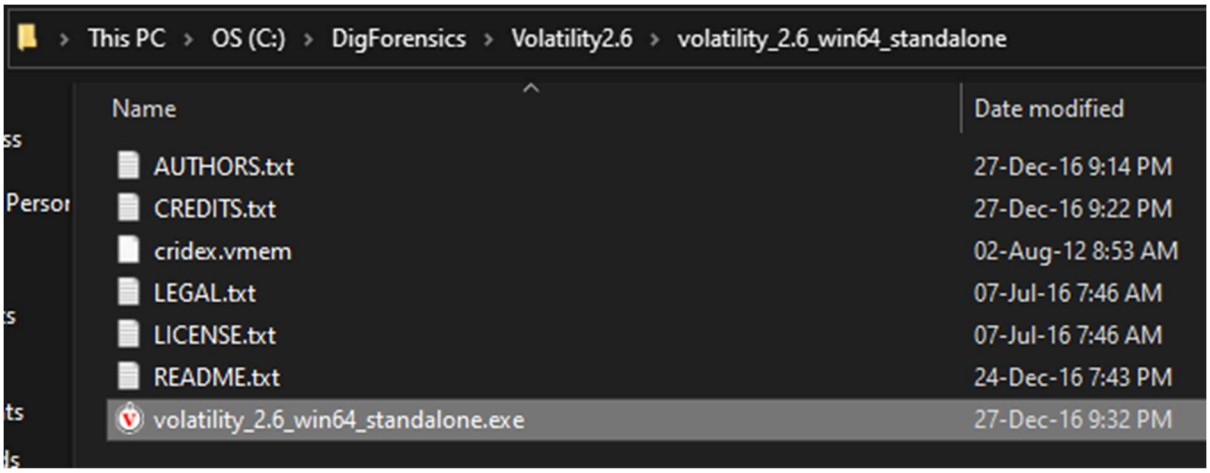
8. To check the running open TCP conections, we can use connscan.



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



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:

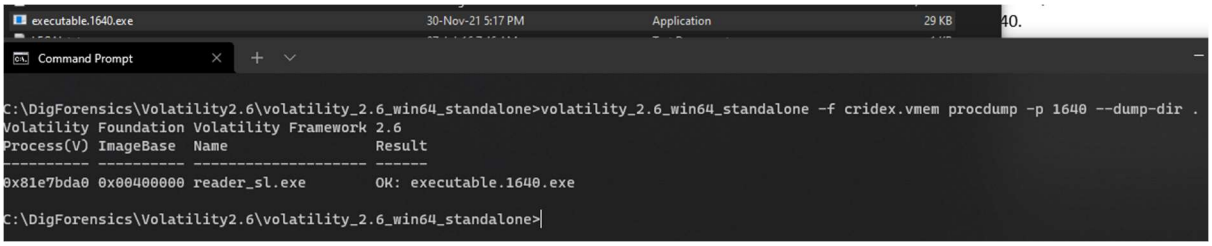
Scolling down we see the command line arguments and the path where reader_sl was stored. It is found using its process id 1640.

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

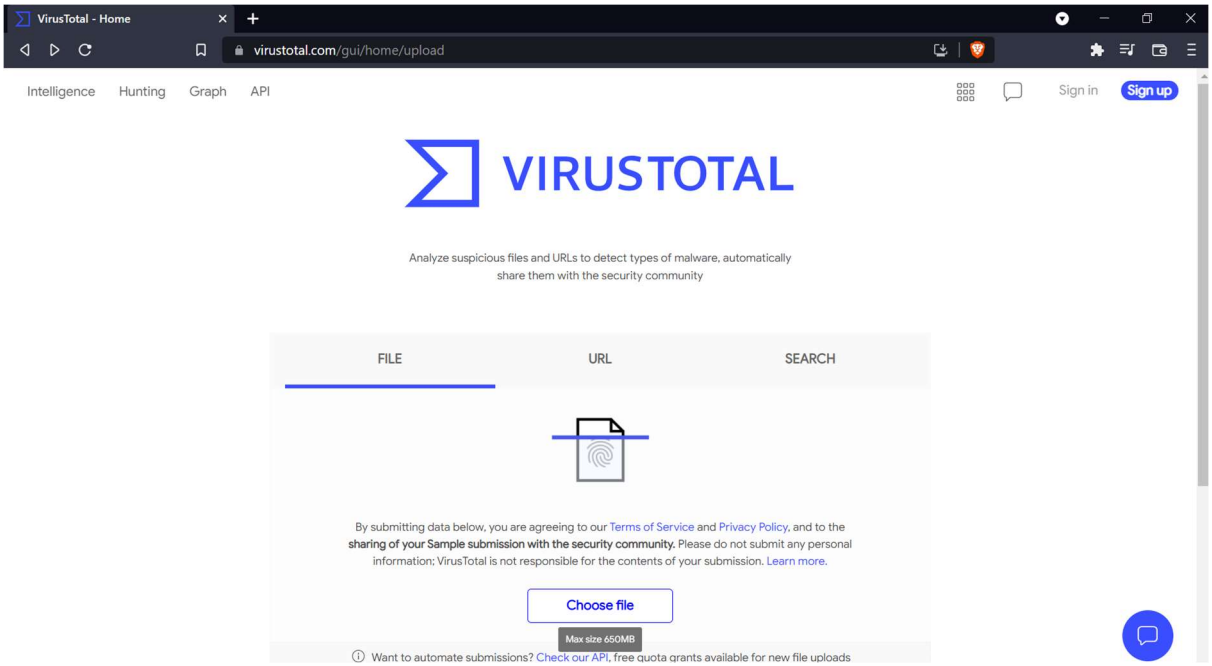
11. We now create a dump of this process and check it out. The command for this is given below using procdump and specifying the PID of the process.

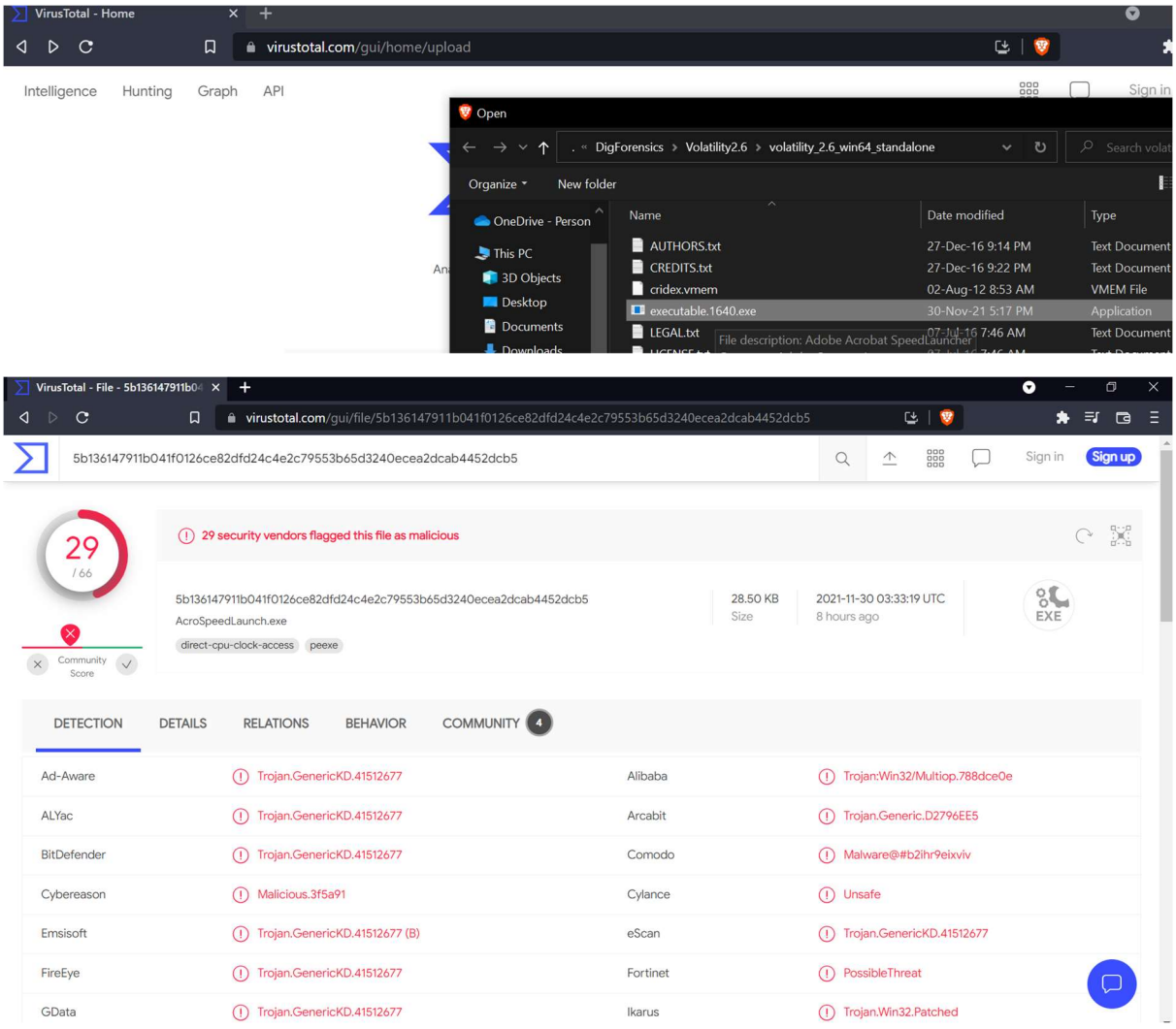
    Command run: volatility_2.6_win64_standalone -f cridex.vmem procdump -p 1640 --dump-dir .

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



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





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.