



**L**OVELY  
**P**ROFESSIONAL  
**U**NIVERSITY

**CA-3**

On

**(Volatility Framework)**

Submitted by

**Name of Student: Venkata Kireeti Savarala**

**Registration No: 11905507**

**Roll No: B53**

**Program Name: Open Source Technologies**

Under the Guidance of

**Dr. Manjot Kaur**

**School of Computer Science & Engineering**

**Lovely Professional University, Phagwara**

(March, 2023)

**Q) Investigate the system run time state of a device (RAM), extract the information present in RAM with the help of volatility Framework.**

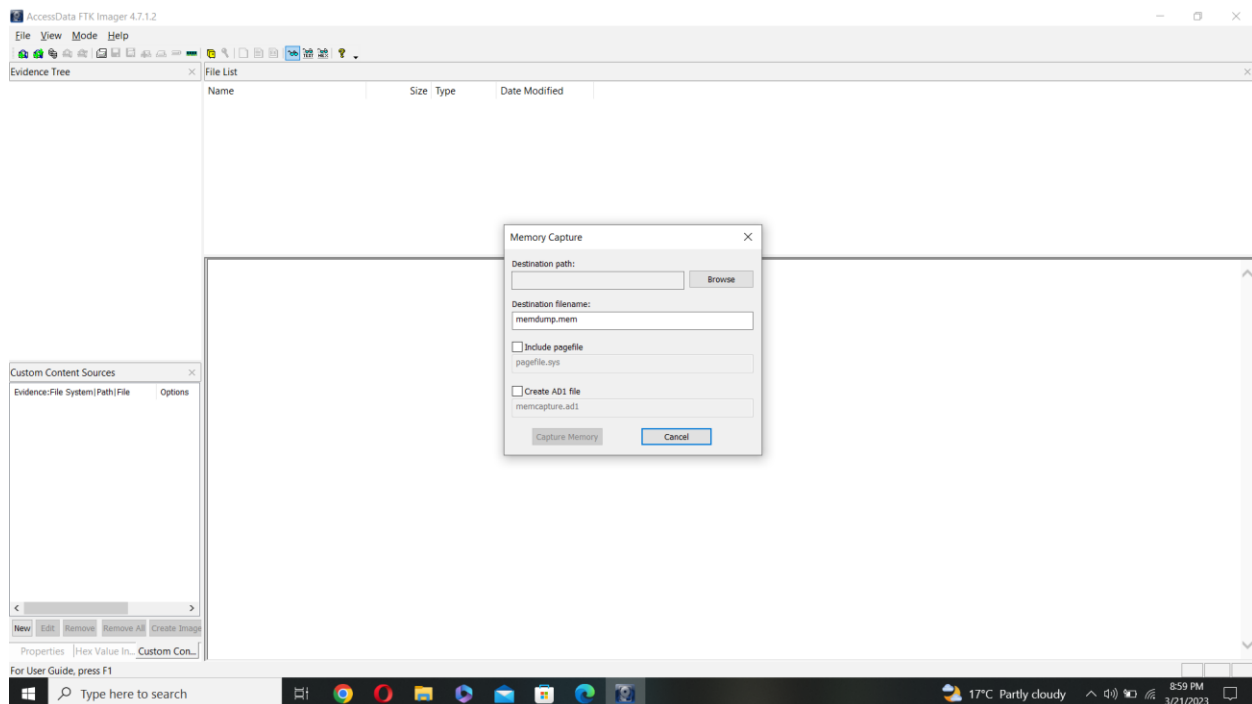
**GITHUB link:** [kireeeti2612002/INT-301\\_CA-3 \(github.com\)](https://github.com/kireeeti2612002/INT-301_CA-3)

### **Step – 1:**

Install Access Data FTK Imager by using below link and this FTK imager helps for getting memory dump into our PC. After installing open it and go to file and click on capture memory. It will look like below image.

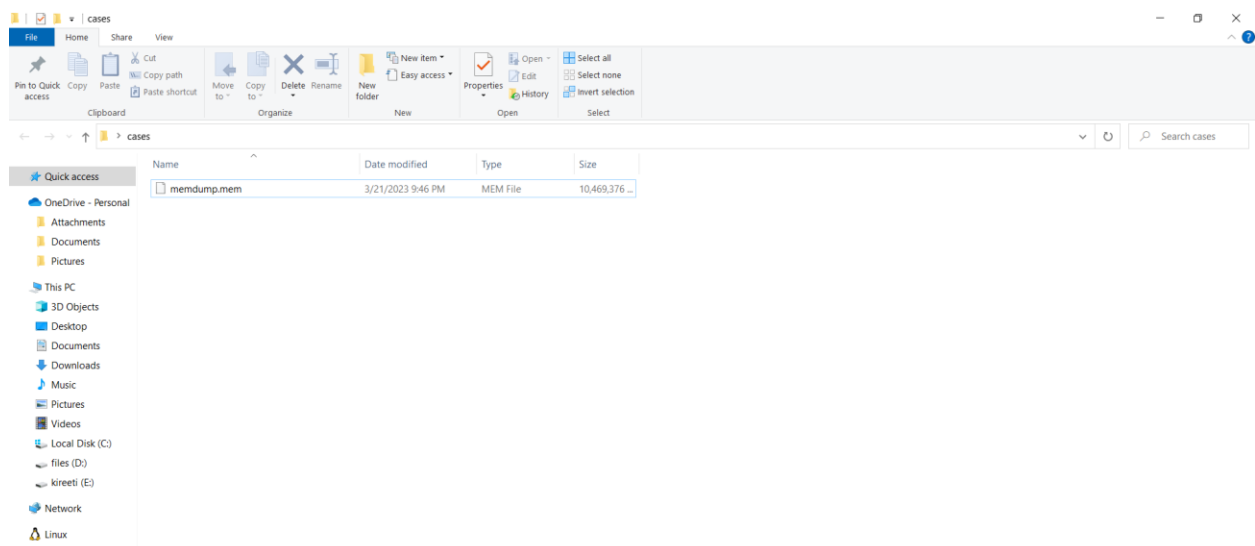
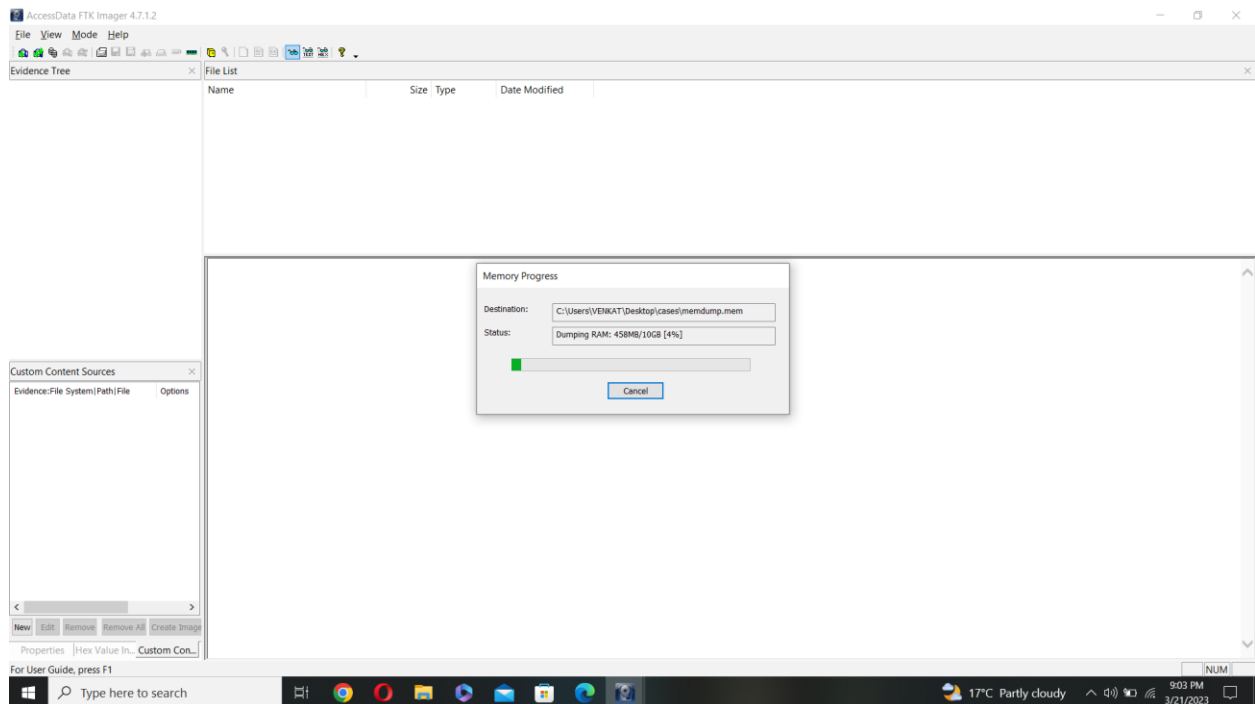
This memory dump of the device will help to investigate the RAM.

Link: <https://www.exterro.com/ftk-imager>



## Step – 2:

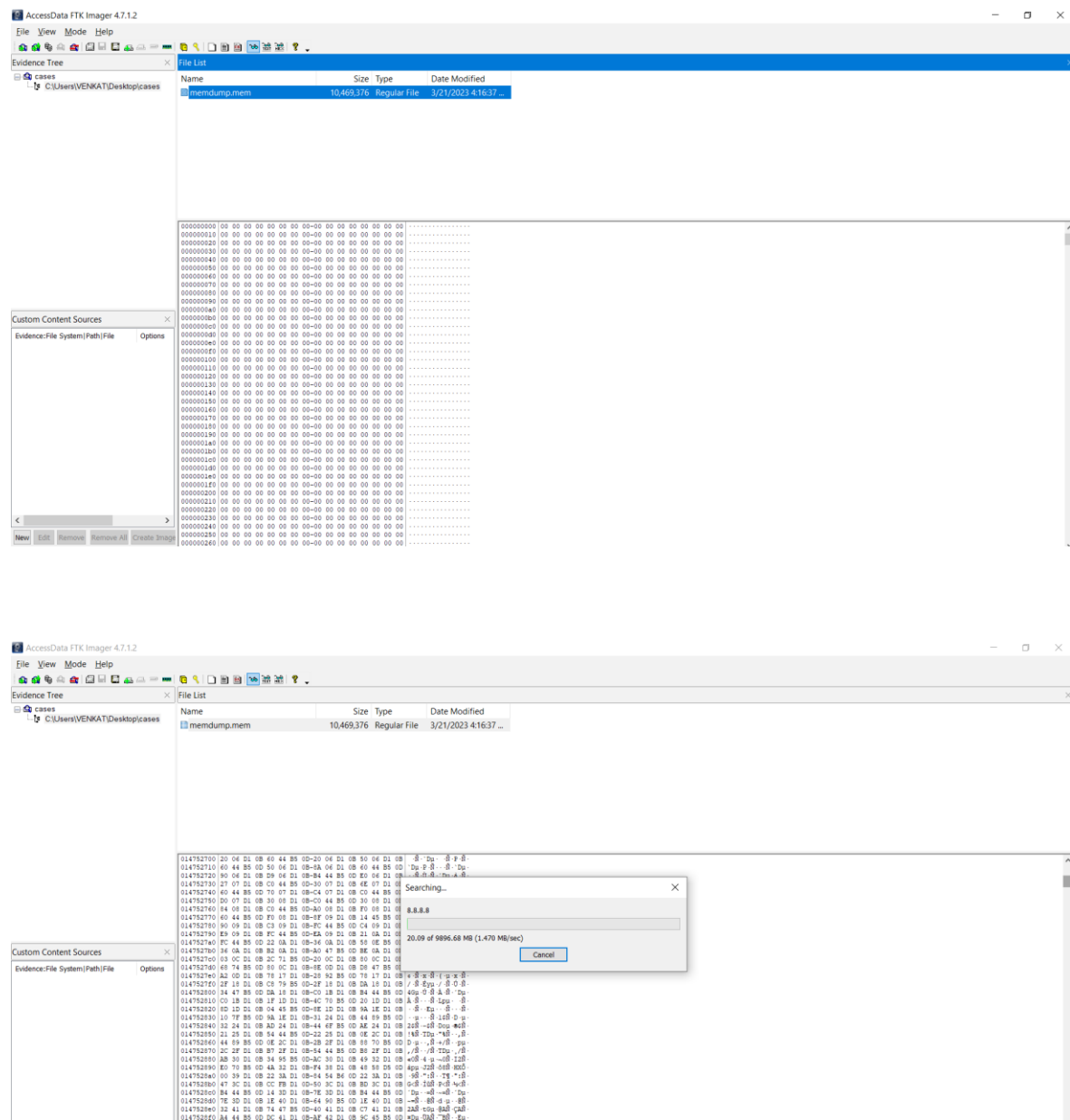
After completing step-1, give the path where to acquire the memory dump. I have given it in a folder called “cases” on Desktop. So after giving path click on capture memory and it will start acquiring memory (mine was 10 GB). It should look like below image and also the memory dump file looks like given image.



## Step – 3:

After completing the process of acquiring memory dump, go Access Data FTK imager, go to file and click on “Add evidence item”, then select “contents of a folder” and give the path of the memory dump file location.

You can see the file on the FTK imager. By clicking on the left side file path, you can view all the information present in it (usernames, network information, etc.). You can also search by using find option like below given image.



## **Step – 4:**

To extract information from RAM we use Volatility Framework, install it by link given below and also to use volatility framework we require Python, so install python by given link below. Also download symbols file to analyze the memory dump.

Volatility Framework link: <https://www.volatilityfoundation.org/>

Install Python link: <https://www.python.org/downloads/>

Symbols file link: <https://tinyurl.com/4xff653v>

## **Step – 5:**

After downloading both Volatility framework and python, put the volatility extracted file in disk (C or D disk) by giving a new folder. Check whether all the files are there in the folder.

The downloaded symbols zip file move it to the symbols folder of volatility3 folder inside volatility folder we created.

Also move the memory dump file into the “D:\volatility3-1.0.0”.

## **Step – 6:**

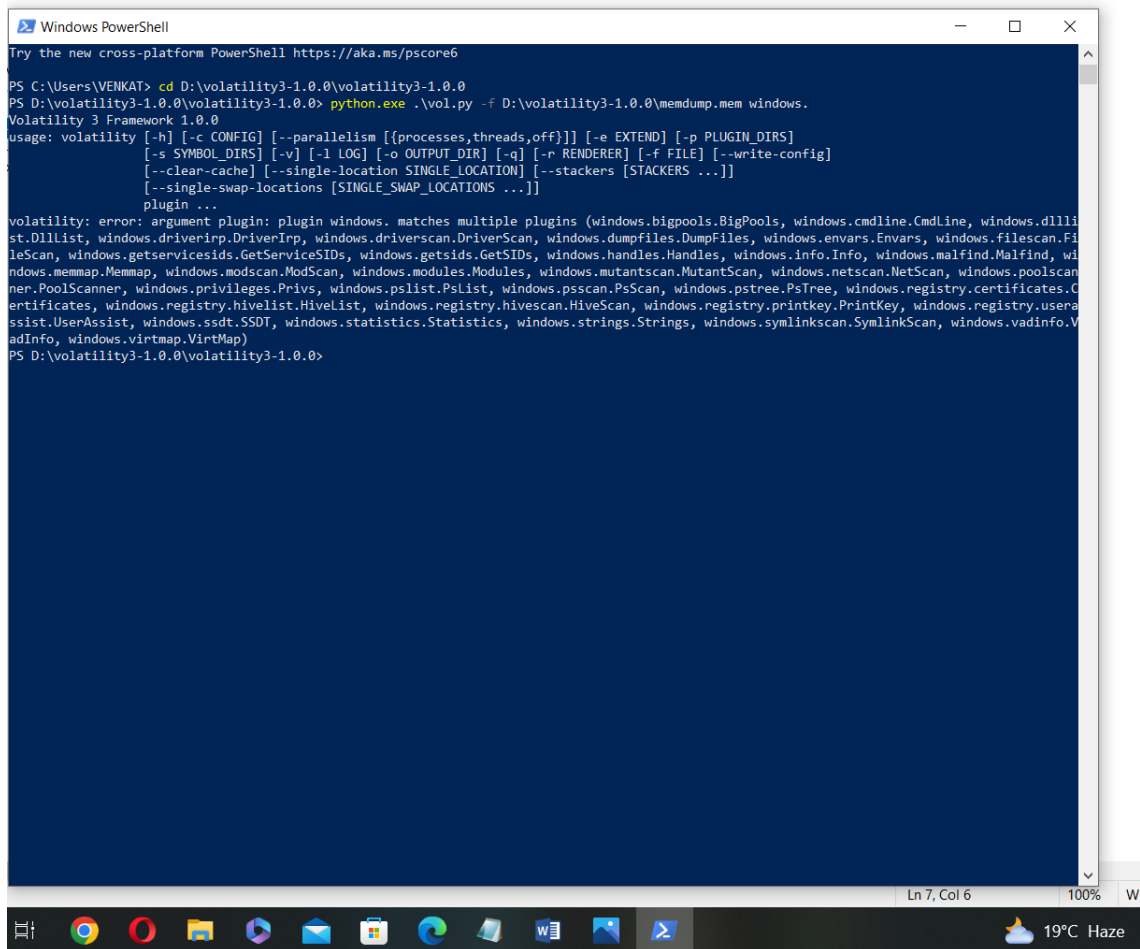
Open PowerShell in PC, first change the directory by using “cd” command. “cd D:\volatility3-1.0.0\volatility3-1.0.0” use this command so that you will be moved to the directory.

After this use “ls” command to list the files present inside the directory and check whether the “vol.py” file is present.

If all ok, then proceed by executing the below commands to extract information from RAM.

## Step – 7:

Use “python.exe .\vol.py -f D:\volatility3-1.0.0\memdump.mem windows.” command to get the plugins which we can use to extract information from memory dump, here you should give the path of volatility folder.



```
Windows PowerShell
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\VENKAT> cd D:\volatility3-1.0.0\volatility3-1.0.0
PS D:\volatility3-1.0.0\volatility3-1.0.0> python.exe .\vol.py -f D:\volatility3-1.0.0\memdump.mem windows.
Volatility 3 Framework 1.0.0
usage: volatility [-h] [-c CONFIG] [--parallelism [[processes,threads,off]]] [-e EXTEND] [-p PLUGIN_DIRS]
                  [-s SYMBOL_DIRS] [-v] [-l LOG] [-o OUTPUT_DIR] [-q] [-r RENDERER] [-f FILE] [--write-config]
                  [--clear-cache] [--single-location SINGLE_LOCATION] [--stackers [STACKERS ...]]
                  [--single-swap-locations [SINGLE_SWAP_LOCATIONS ...]]
                  plugin ...
volatility: error: argument plugin: plugin windows. matches multiple plugins (windows.bigpools.BigPools, windows.cmdline.Cmdline, windows.dlllist.DllList, windows.driverirp.DriverIrp, windows.driverscan.DriverScan, windows.dumpfiles.DumpFiles, windows.envvars.Envvars, windows.filescan.FileScan, windows.getservicesids.GetServiceSIDs, windows.getsids.GetSIDs, windows.handles.Handles, windows.info.Info, windows.malfind.Malfind, windows.memmap.Memmap, windows.modscan.ModScan, windows.modules.Modules, windows.mutantscan.MutantScan, windows.netscan.NetScan, windows.poolscan.PoolScanner, windows.privileges.Privs, windows.pslist.PsList, windows.psscan.PsScan, windows.pstree.PsTree, windows.registry.certificates.Certificates, windows.registry.hivelist.HiveList, windows.registry.hivescan.HiveScan, windows.registry.printkey.PrintKey, windows.registry.userassist.UserAssist, windows.ssdts.SSDT, windows.statistics.Statistics, windows.strings.Strings, windows.symblinkscan.SymblinkScan, windows.vadinfo.VadInfo, windows.virtmap.VirtMap)
PS D:\volatility3-1.0.0\volatility3-1.0.0>
```

### Step – 8:

Use “python.exe .\vol.py -f D:\volatility3-1.0.0\memdump.mem windows.info” plugin and command to get the full information about the windows of our PC from memory dump.

```

C:\Administrator\Windows PowerShell
PS C:\Users\Borat> cd C:\ProgramData\PowerShell
PS C:\ProgramData\PowerShell> cd .\Scripts
PS C:\ProgramData\PowerShell\Scripts> curl https://aka.ms/powershell

The new cross-platform Powershell https://aka.ms/powershell

PS C:\ProgramData\PowerShell\Scripts> Start-Transcript -Path "D:\volatility\OneWall\Franscript.txt"
Transcript started, output file is C:\Users\BORAT\Desktop\PowerShell\transcript.txt
PS C:\ProgramData\PowerShell\Scripts> cd D:\volatility\1.0.0
PS D:\volatility\1.0.0> python.exe .\vol.py -f D:\volatility\1.0.0\memdump.exe windows.info

Directory: D:\volatility\1.0.0\volatility\1.0.0


Mode                LastWriteTime         Length Name
----                -
d-----        2/1/2021   9:00 PM             github
d-----        2/1/2021   9:00 PM          development
d-----        2/1/2021   9:00 PM              doc
d-----        2/21/2021  8:13 PM      volatility3
d-----        2/1/2021   9:00 PM       104 gitignore
d-----        2/1/2021   9:00 PM    432 README.md
d-----        2/1/2021   9:00 PM     7680 style.pdf
d-----        2/1/2021   9:00 PM    3020 LICENSE.txt
d-----        2/1/2021   9:00 PM    180 NOTEST.txt
d-----        2/1/2021   9:00 PM       79 pyproject.toml
d-----        2/1/2021   9:00 PM    5061 README.md
d-----        2/1/2021   9:00 PM    2083 setup.py
d-----        2/1/2021   9:00 PM     280 vol.py
d-----        2/1/2021   9:00 PM    5423 vol.spec
d-----        2/1/2021   9:00 PM     297 volwall.py
d-----        2/1/2021   9:00 PM    2963 volwall.spec

PS D:\volatility\1.0.0\volatility\1.0.0> python.exe .\vol.py -f D:\volatility\1.0.0\memdump.exe windows.info
Volatility 3 Framework 1.0.0
Usage: vol.py [-h] PDB scanning finished
Variable Value
Kernel Base 0x80507000000
CPU k8LAME
Symbols file:///D:/volatility\1.0.0\volatility\1.0.0\volatility\symbols/windows/ntkernel.pdb/SF9CF30512F3833348B4AB2AC6596-1_jsoe.xz
Cache hit True
Cache size 0
Cache valid False
Library @ WindowsIntel12e
Memory layer 3 file layer
KDBG loaded 0x80507000000
Page offset 15.19041
Machine type 34484
Processor architecture amd64
System time 2023-02-21 15:36:15
System root C:\WINDOWS
Product type WinProductTypeNT
Patched version 10
Patched version 10
Patched version 0

```

### Step -9:

Use “python.exe .\vol.py -f D:\volatility3-1.0.0\memdump.mem windows.pslist” plugin and command to get the list all running processes on the system.

```
Administrator: Windows PowerShell
PS MajorOperatingSystemVersion 10
PS MinorOperatingSystemVersion 0
PS Machine
PS ImageDateStamp Sun Mar 4 22:56:30 2024
PS O_Volatility_1.8.0 Volatility_1.8.0 python.exe -d O:\volatility_1.8.0\memdump.new windows.psill
Volatility 3 Framework 1.8.0
Progress: 100.0%
PSD scanning finished
OffsetView Threads Handles SessionId Mon4d CreateTime ExitTime File output
1 0 System DacBfd6dc6d08 255 N/A False 2023-03-09 03:19:45.000000 N/A Disabled
1 124 4 Registry DacBfd6d6d15080 4 - N/A False 2023-03-09 03:18:39.000000 N/A Disabled
894 4 smss.exe DacBfd7f3080c 2 - N/A False 2023-03-09 03:19:48.000000 N/A Disabled
780 800 csrss.exe DacBfd7f30800 16 - N/A False 2023-03-09 03:19:42.000000 N/A Disabled
896 780 wininit.exe DacBfd4880000 0 - False 2023-03-09 03:14:37.000000 N/A Disabled
896 780 csrss.exe DacBfd4880000 0 - False 2023-03-09 03:14:37.000000 N/A Disabled
902 896 NotNull.exe DacBfd4880000 0 - False 2023-03-09 03:14:37.000000 2023-03-09 04:07:05.000000 Disabled
1620 896 lsass.exe DacBfd7f30800 12 - N/A False 2023-03-09 03:14:37.000000 N/A Disabled
1620 896 csrss.exe DacBfd7f30800 8 - N/A False 2023-03-09 03:14:37.000000 2023-03-09 05:27:54.000000 Disabled
1134 968 svchost.exe DacBfd7d4c080 29 - N/A False 2023-03-09 03:14:44.000000 N/A Disabled
1134 968 csrss.exe DacBfd7d4c080 2 - N/A False 2023-03-09 03:14:44.000000 N/A Disabled
1236 968 MDMHost.exe DacBfd0517080 11 - False 2023-03-09 03:14:44.000000 N/A Disabled
1276 968 svchost.exe DacBfd5142d0 16 - N/A False 2023-03-09 03:14:46.000000 N/A Disabled
1276 968 csrss.exe DacBfd5142d0 4 - N/A False 2023-03-09 03:14:46.000000 N/A Disabled
1604 968 svchost.exe DacBfd6650800 12 - False 2023-03-09 03:14:58.000000 N/A Disabled
1604 968 csrss.exe DacBfd6650800 6 - False 2023-03-09 03:14:58.000000 N/A Disabled
1644 968 svchost.exe DacBfd6650800 6 - False 2023-03-09 03:14:58.000000 N/A Disabled
1644 968 IntelIPCCPVC DacBfd66d0000 4 - N/A False 2023-03-09 03:14:58.000000 N/A Disabled
1808 968 svchost.exe DacBfd66d0000 4 - False 2023-03-09 03:14:59.000000 N/A Disabled
1808 968 svchost.exe DacBfd665080 5 - False 2023-03-09 03:14:58.000000 N/A Disabled
2008 968 svchost.exe DacBfd66d0000 2 - False 2023-03-09 03:14:52.000000 N/A Disabled
2008 968 IntelIPCCSvc DacBfd9250800 3 - False 2023-03-09 03:14:52.000000 N/A Disabled
968 968 Igf3CUIService DacBfd9260800 2 - False 2023-03-09 03:14:52.000000 N/A Disabled
1312 968 svchost.exe DacBfd9250800 4 - False 2023-03-09 03:14:55.000000 N/A Disabled
2220 968 svchost.exe DacBfd930800 9 - False 2023-03-09 03:14:56.000000 N/A Disabled
2220 968 csrss.exe DacBfd930800 7 - False 2023-03-09 03:14:51.000000 N/A Disabled
2464 968 svchost.exe DacBfd9310800 15 - False 2023-03-09 03:14:03.000000 N/A Disabled
2464 968 svchost.exe DacBfd9310800 3 - False 2023-03-09 03:14:06.000000 N/A Disabled
2464 968 csrss.exe DacBfd9310800 13 - False 2023-03-09 03:14:06.000000 N/A Disabled
2556 968 svchost.exe DacBfd9310800 16 - False 2023-03-09 03:14:07.000000 N/A Disabled
2556 968 csrss.exe DacBfd9310800 6 - False 2023-03-09 03:14:06.000000 N/A Disabled
2796 968 svchost.exe DacBfd9370800 6 - False 2023-03-09 03:14:09.000000 N/A Disabled
2836 968 MDIPlayerCont DacBfd9810800 10 - False 2023-03-09 03:14:09.000000 N/A Disabled
2836 968 svchost.exe DacBfd9830800 3 - False 2023-03-09 03:14:18.000000 N/A Disabled
2948 968 svchost.exe DacBfd9890800 2 - False 2023-03-09 03:14:18.000000 N/A Disabled
2948 968 csrss.exe DacBfd9890800 7 - False 2023-03-09 03:14:18.000000 N/A Disabled
2996 968 svchost.exe DacBfd9860800 3 - False 2023-03-09 03:14:18.000000 N/A Disabled
3696 968 svchost.exe DacBfd9860800 3 - False 2023-03-09 03:11.000000 N/A Disabled
3696 968 csrss.exe DacBfd9860800 7 - False 2023-03-09 03:11.000000 N/A Disabled
2092 4 MemCompression DacBfd98610d0 54 N/A False 2023-03-09 03:14:11.000000 N/A Disabled
3216 968 svchost.exe DacBfd98610d0 17 - False 2023-03-09 03:14:17.000000 N/A Disabled
3216 968 svchost.exe DacBfd98610d0 4 - False 2023-03-09 03:14:18.000000 N/A Disabled
3648 968 svchost.exe DacBfd013080 15 - False 2023-03-09 03:14:23.000000 N/A Disabled
3648 968 csrss.exe DacBfd013080 6 - False 2023-03-09 03:14:23.000000 N/A Disabled
3664 968 svchost.exe DacBfd0130800 10 - False 2023-03-09 03:14:23.000000 N/A Disabled
3892 968 svchost.exe DacBfd0130800 5 - False 2023-03-09 03:14:25.000000 N/A Disabled
```

We can use various plugins and commands available in the Volatility framework.

**pstree:** This command displays the running processes in a hierarchical tree view.

**psscan:** This command used to scan the memory for processes and provides us the detailed information about the each process.

**filescan:** This command scans the memory for open files and provides detailed information of each file.

**netscan:** This command lists all the network connections that currently active on the system.

**dumpfiles:** This command is used to extract files from memory to disk.

**malfind:** This command is used to scan the memory for suspicious code or malware.

These are some examples of plugins and commands available in the Volatility framework. To use any of these commands, simply specify the name of the plugin or command when running the Volatility framework.