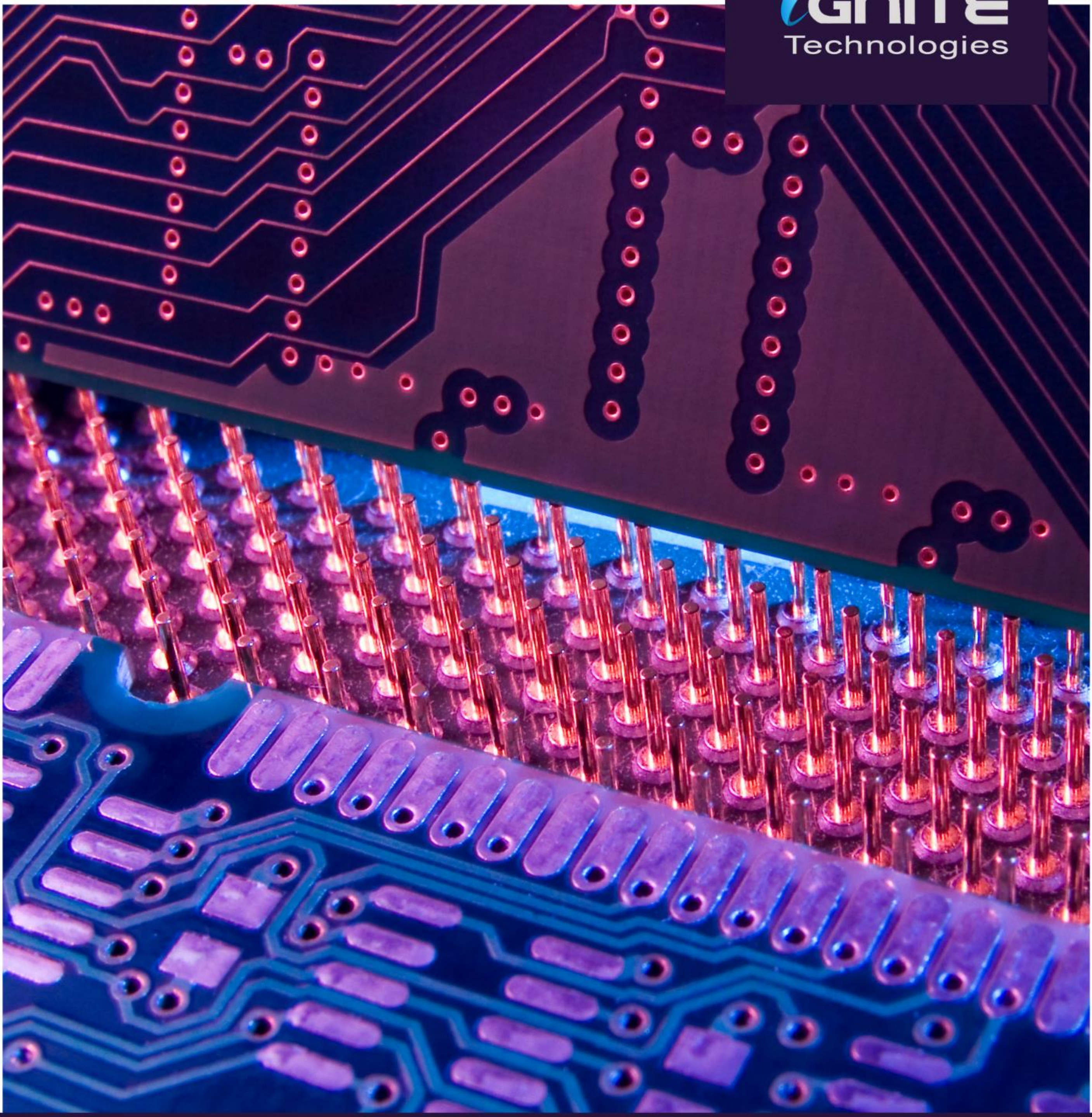
iGNITE Technologies



MEMORY FORENSICS VOLATILITY

FRAMEWORK & WORKBENCH

WWW.HACKINGARTICLES.IN

Table of Contents

Abstract

Cyber Criminals and **attackers** have become so creative in their crime type that they have started finding methods to hide data in the **volatile memory** of the systems. Today, in this article we are going to have a greater understanding of live **memory acquisition** and its **forensic analysis**. Live Memory acquisition is a method that is used to collect data when the system is found in an active state at the scene of the crime.

Memory forensics is a division of digital forensics that generally emphasizes extracting **artefacts** from the volatile memory of a system that was compromised. This domain is speedily spreading in cybercrime investigations. The main reason for this is that certain artefacts are extracted from system memory only and cannot be found anywhere else.

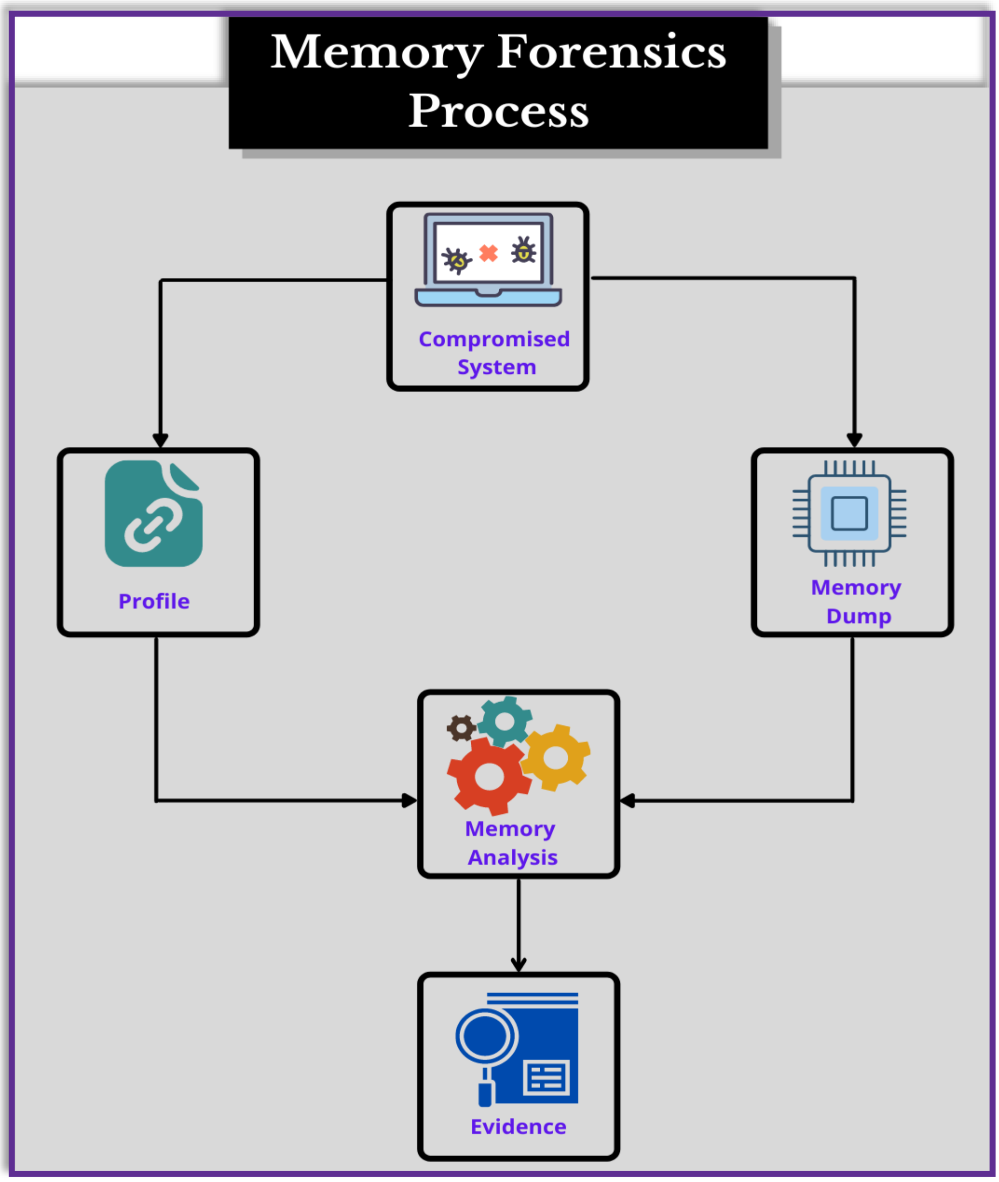
Analysing memory after **capturing the ram** is extremely important when it comes to collecting information on ports that were in use, the number of processes running, and the path of certain executables on the system while carrying out the investigation. The **Volatility Framework** is one such memory analysis tool that works on **command-line** on **Windows** and **Linux** systems.

Volatility Workbench is a **GUI version** of one of the same tool Volatility for analysing the artefacts from a memory dump. It is available free of cost, open-source, and runs on the Windows Operating system.

Introduction

Memory Forensics

Memory Forensics is a budding field in Digital Forensics Investigation which involves recovering, extracting and analysing evidence such as images, documents, or chat histories etc from the structured volatile memory into non-volatile devices like Hard-drives or USB drives.



NOTE: We have taken a memory dump of a Windows7 system using the Belkasoft RAM Capturer, which can be downloaded from here.

Memory Acquisition

- It is the method of capturing and dumping the contents of a volatile content into a non-volatile storage device to preserve it for further investigation.
- A ram analysis can only be successfully conducted when the acquisition has been performed accurately without corrupting the image of the volatile memory.
- In this phase, the investigator has to be careful about his decisions to collect the volatile data as it won't exist after the system undergoes a reboot.
- The volatile memory can also be prone to alteration of any sort due to the continuous processes running in the background.
- Any external move made on the suspect system may impact the device's ram adversely.

Importance of Memory Acquisition

When a volatile memory is captured, the following artefacts can be discovered which can be useful to the investigation:

- On-going processes and recently terminated processes
- Files mapped in the memory (.exe, .txt, shared files, etc.)
- Any open TCP/UDP ports or any active connections
- Caches (clipboard data, SAM databases, edited files, passwords, web addresses, commands)
- Presence of hidden data, malware, etc.

Memory Analysis

Once the dump is available, we will begin with the forensic analysis of the memory using the Volatility Memory Forensics Framework which can be downloaded from here. The volatility framework support analysis of **memory dump** from all the versions and services of Windows from **XP** to **Windows 10**. It also supports **Server 2003** to **Server 2016**. In this article, we will be analysing the memory dump in Kali Linux where Volatility comes pre-installed.

NOTE: **Dump Format Supported-** Raw format, Hibernation File, VM snapshot, Microsoft crash dump

Volatility Framework

Volatility Framework processes RAM dumps in various formats which can be used to process crash dumps, hibernation files and, page files that may be found on dumps of storage drives. RAM dumps from virtual machines or hypervisors can also be processed.

Data found using Volatility Framework

A huge amount of data can be availed on analysing volatile memory. It includes data like processes, information on open files, registry handles, information on the network and open ports, passwords and cryptographic keys, hidden data, worms and rootkits etc.

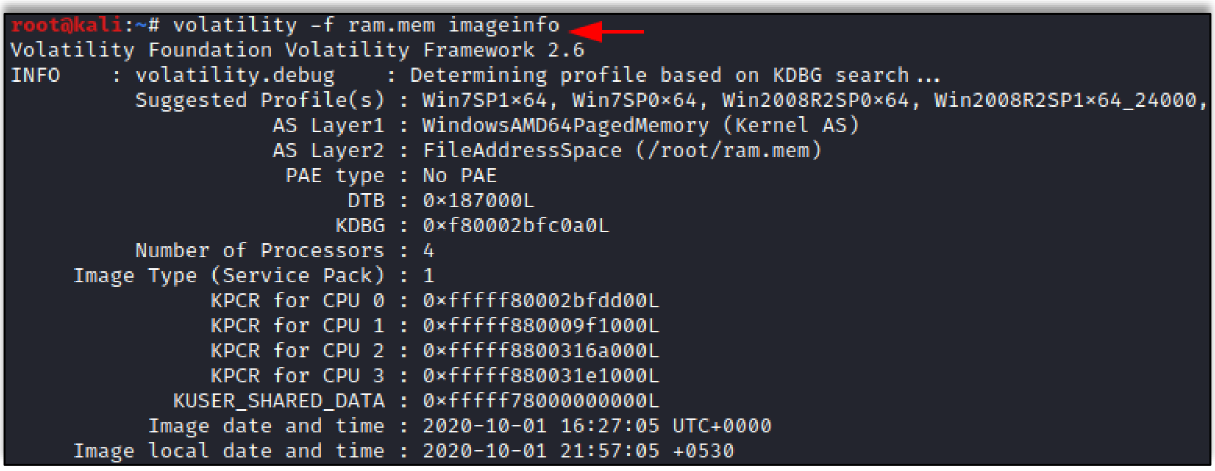
Switch on your Kali Linux Machines, and to get a basic list of all the available options, plugins, and flags to use in the analysis, you can type:

```
volatility -h
```

1. Imageinfo

When a Memory dump is taken, it is extremely important to know the information about the operating system that was in use. Volatility will try to read the image and suggest the related profiles for the given memory dump. The image info plugin displays the date and time of the sample that was collected, the number of CPUs present, etc. To obtain the details of the ram, you can type;

```
volatility -f ram.mem imageinfo
```



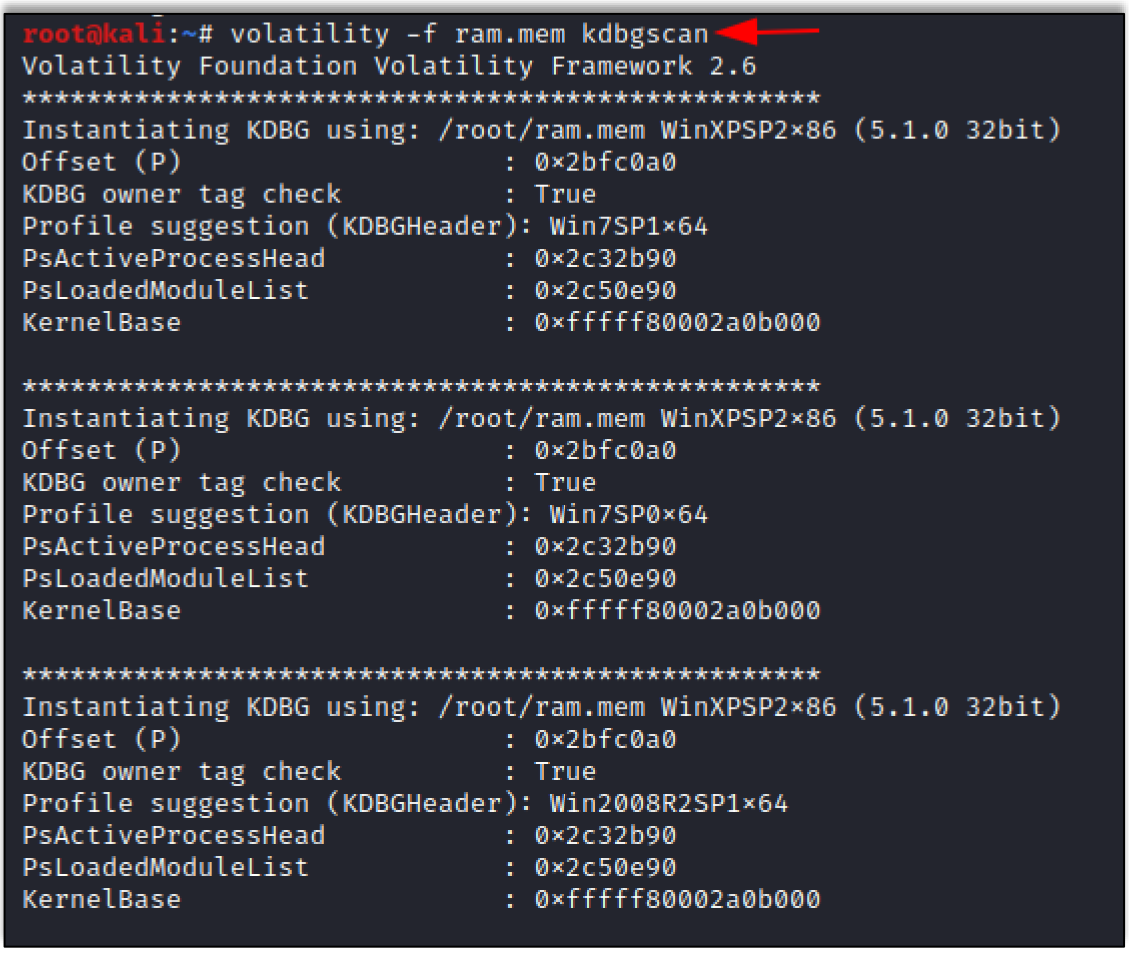
```
root@kali:~# volatility -f ram.mem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search ...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_24000,
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/root/ram.mem)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf80002bfc0a0L
          Number of Processors : 4
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff80002bfdd00L
                KPCR for CPU 1 : 0xfffff880009f1000L
                KPCR for CPU 2 : 0xfffff8800316a000L
                KPCR for CPU 3 : 0xfffff880031e1000L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2020-10-01 16:27:05 UTC+0000
     Image local date and time : 2020-10-01 21:57:05 +0530
```

A profile is a categorization of specific operating systems, versions and their hardware architecture, A profile generally includes metadata information, system call information, etc. You may notice multiple profiles would be suggested to you.

2. Kdbgscan

This plugin finds and analyses the profiles based on the Kernel debugger data block. The Kdbgscan thus provides the correct profile related to the raw image. It is extremely important to get the right profile for memory analysis. To supply the correct profile for the memory analysis, type

```
volatility -f ram.mem kdbgscan
```



```
root@kali:~# volatility -f ram.mem kdbgscan
Volatility Foundation Volatility Framework 2.6
**************************************************
Instantiating KDBG using: /root/ram.mem WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x2bfc0a0
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win7SP1x64
PsActiveProcessHead           : 0x2c32b90
PsLoadedModuleList            : 0x2c50e90
KernelBase                    : 0xfffff80002a0b000

**************************************************
Instantiating KDBG using: /root/ram.mem WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x2bfc0a0
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win7SP0x64
PsActiveProcessHead           : 0x2c32b90
PsLoadedModuleList            : 0x2c50e90
KernelBase                    : 0xfffff80002a0b000

**************************************************
Instantiating KDBG using: /root/ram.mem WinXPSP2x86 (5.1.0 32bit)
Offset (P)                    : 0x2bfc0a0
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win2008R2SP1x64
PsActiveProcessHead           : 0x2c32b90
PsLoadedModuleList            : 0x2c50e90
KernelBase                    : 0xfffff80002a0b000
```

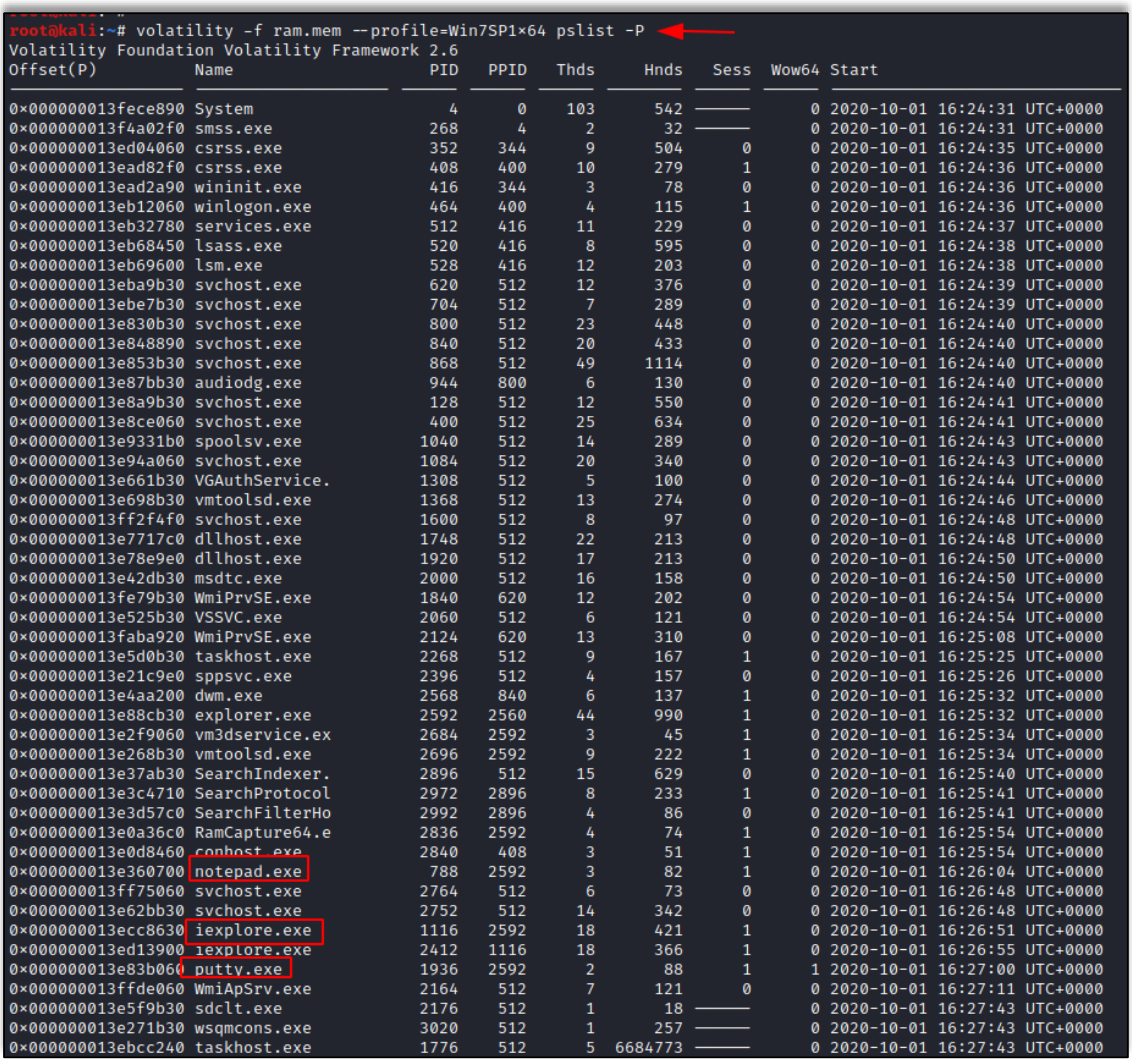
3. Processes

When a system is in an active state it is normal for it to have multiple processes running in the background and can be found in the volatile memory. It consists of executable program code, imported libraries, allocated memory, execution threads. The presence of any hidden process can also be parsed out of a memory dump. The recently terminated processes before the reboot can also be recorded and analysed in the memory dump. There are a few plugins that can be used to list the processes to carry out forensic Investigation.

PSlist

To identify the presence of any rogue processes and to view any high-level running processes.
On executing this command, the list of processes running is displayed, their respective process ID assigned to them and the parent process ID is also displayed along. The details about the threads, sessions, handles are also mentioned. The timestamp according to the start of the process is also displayed. This helps to identify whether an unknown process is running or was running at an unusual time. It will not give information about processes that were hidden by removing themselves from the process list or the ones that were terminated before

```
volatility -f ram.mem --profile=Win7SP1x64 pslist -P
```

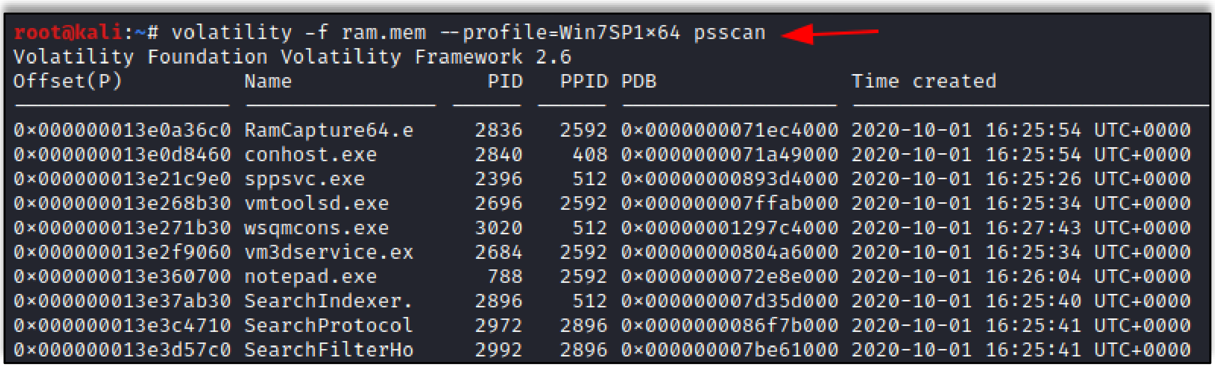


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 pslist -P
Volatility Foundation Volatility Framework 2.6
Offset(P)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------

0x000000013fece890 System                    4      0    103      542 ------      0 2020-10-01 16:24:31 UTC+0000
0x000000013f4a02f0 smss.exe                268      4      2       32 ------      0 2020-10-01 16:24:31 UTC+0000
0x000000013ed04060 csrss.exe               352    344      9      504      0      0 2020-10-01 16:24:35 UTC+0000
0x000000013ead82f0 csrss.exe               408    400     10      279      1      0 2020-10-01 16:24:36 UTC+0000
0x000000013ead2a90 wininit.exe             416    344      3       78      0      0 2020-10-01 16:24:36 UTC+0000
0x000000013eb12060 winlogon.exe            464    400      4      115      1      0 2020-10-01 16:24:36 UTC+0000
0x000000013eb32780 services.exe            512    416     11      229      0      0 2020-10-01 16:24:37 UTC+0000
0x000000013eb68450 lsass.exe               520    416      8      595      0      0 2020-10-01 16:24:38 UTC+0000
0x000000013eb69600 lsm.exe                 528    416     12      203      0      0 2020-10-01 16:24:38 UTC+0000
0x000000013eba9b30 svchost.exe             620    512     12      376      0      0 2020-10-01 16:24:39 UTC+0000
0x000000013ebe7b30 svchost.exe             704    512      7      289      0      0 2020-10-01 16:24:39 UTC+0000
0x000000013e830b30 svchost.exe             800    512     23      448      0      0 2020-10-01 16:24:40 UTC+0000
0x000000013e848890 svchost.exe             840    512     20      433      0      0 2020-10-01 16:24:40 UTC+0000
0x000000013e853b30 svchost.exe             868    512     49     1114      0      0 2020-10-01 16:24:40 UTC+0000
0x000000013e87bb30 audiodg.exe             944    800      6      130      0      0 2020-10-01 16:24:40 UTC+0000
0x000000013e8a9b30 svchost.exe             128    512     12      550      0      0 2020-10-01 16:24:41 UTC+0000
0x000000013e8ce060 svchost.exe             400    512     25      634      0      0 2020-10-01 16:24:41 UTC+0000
0x000000013e9331b0 spoolsv.exe            1040    512     14      289      0      0 2020-10-01 16:24:43 UTC+0000
0x000000013e94a060 svchost.exe            1084    512     20      340      0      0 2020-10-01 16:24:43 UTC+0000
0x000000013e661b30 VGAuthService.         1308    512      5      100      0      0 2020-10-01 16:24:44 UTC+0000
0x000000013e698b30 vmtoolsd.exe           1368    512     13      274      0      0 2020-10-01 16:24:46 UTC+0000
0x000000013ff2f4f0 svchost.exe            1600    512      8       97      0      0 2020-10-01 16:24:48 UTC+0000
0x000000013e7717c0 dllhost.exe            1748    512     22      213      0      0 2020-10-01 16:24:48 UTC+0000
0x000000013e78e9e0 dllhost.exe            1920    512     17      213      0      0 2020-10-01 16:24:50 UTC+0000
0x000000013e42db30 msdtc.exe              2000    512     16      158      0      0 2020-10-01 16:24:50 UTC+0000
0x000000013fe79b30 WmiPrvSE.exe           1840    620     12      202      0      0 2020-10-01 16:24:54 UTC+0000
0x000000013e525b30 VSSVC.exe              2060    512      6      121      0      0 2020-10-01 16:24:54 UTC+0000
0x000000013faba920 WmiPrvSE.exe           2124    620     13      310      0      0 2020-10-01 16:25:08 UTC+0000
0x000000013e5d0b30 taskhost.exe           2268    512      9      167      1      0 2020-10-01 16:25:25 UTC+0000
0x000000013e21c9e0 sppsvc.exe             2396    512      4      157      0      0 2020-10-01 16:25:26 UTC+0000
0x000000013e4aa200 dwm.exe                2568    840      6      137      1      0 2020-10-01 16:25:32 UTC+0000
0x000000013e88cb30 explorer.exe           2592   2560     44      990      1      0 2020-10-01 16:25:32 UTC+0000
0x000000013e2f9060 vm3dservice.ex         2684   2592      3       45      1      0 2020-10-01 16:25:34 UTC+0000
0x000000013e268b30 vmtoolsd.exe           2696   2592      9      222      1      0 2020-10-01 16:25:34 UTC+0000
0x000000013e37ab30 SearchIndexer.         2896    512     15      629      0      0 2020-10-01 16:25:40 UTC+0000
0x000000013e3c4710 SearchProtocol         2972   2896      8      233      1      0 2020-10-01 16:25:41 UTC+0000
0x000000013e3d57c0 SearchFilterHo         2992   2896      4       86      0      0 2020-10-01 16:25:41 UTC+0000
0x000000013e0a36c0 RamCapture64.e         2836   2592      4       74      1      0 2020-10-01 16:25:54 UTC+0000
0x000000013e0d8460 conhost.exe            2840    408      3       51      1      0 2020-10-01 16:25:54 UTC+0000
0x000000013e360700 notepad.exe             788   2592      3       82      1      0 2020-10-01 16:26:04 UTC+0000
0x000000013ff75060 svchost.exe            2764    512      6       73      0      0 2020-10-01 16:26:48 UTC+0000
0x000000013e62bb30 svchost.exe            2752    512     14      342      0      0 2020-10-01 16:26:48 UTC+0000
0x000000013ecc8630 iexplore.exe           1116   2592     18      421      1      0 2020-10-01 16:26:51 UTC+0000
0x000000013ed13900 iexplore.exe           2412   1116     18      366      1      0 2020-10-01 16:26:55 UTC+0000
0x000000013e83b060 putty.exe              1936   2592      2       88      1      1 2020-10-01 16:27:00 UTC+0000
0x000000013ffde060 WmiApSrv.exe           2164    512      7      121      0      0 2020-10-01 16:27:11 UTC+0000
0x000000013e5f9b30 sdclt.exe              2176    512      1       18 ------      0 2020-10-01 16:27:43 UTC+0000
0x000000013e271b30 wsqmcons.exe           3020    512      1      257 ------      0 2020-10-01 16:27:43 UTC+0000
0x000000013ebcc240 taskhost.exe           1776    512      5  6684773 ------      0 2020-10-01 16:27:43 UTC+0000
```

PSscan

This plugin can be used to give a detailed list of processes found in the memory dump. On executing this command, the list of processes running is displayed, their respective process ID assigned to them and the parent process ID is also displayed along. The details about the threads, sessions, handles are also mentioned. The timestamp according to the start of the process is also displayed. This helps to identify whether an unknown process is running or was running at an unusual time

```
volatility -f ram.mem --profile=Win7SP1x64 psscan
```

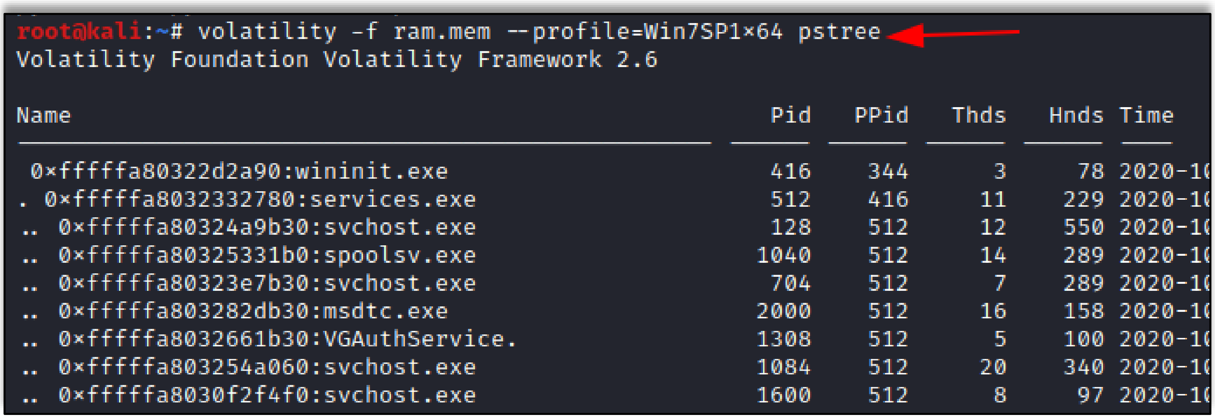


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 psscan
Volatility Foundation Volatility Framework 2.6
Offset(P)          Name                PID   PPID PDB                Time created
------------------ ---------------- ------ ------ ------------------ ------------------------------

0x000000013e0a36c0 RamCapture64.e     2836   2592 0x0000000071ec4000 2020-10-01 16:25:54 UTC+0000
0x000000013e0d8460 conhost.exe        2840    408 0x0000000071a49000 2020-10-01 16:25:54 UTC+0000
0x000000013e21c9e0 sppsvc.exe         2396    512 0x00000000893d4000 2020-10-01 16:25:26 UTC+0000
0x000000013e268b30 vmtoolsd.exe       2696   2592 0x000000007ffab000 2020-10-01 16:25:34 UTC+0000
0x000000013e271b30 wsqmcons.exe       3020    512 0x00000001297c4000 2020-10-01 16:27:43 UTC+0000
0x000000013e2f9060 vm3dservice.ex     2684   2592 0x00000000804a6000 2020-10-01 16:25:34 UTC+0000
0x000000013e360700 notepad.exe         788   2592 0x0000000072e8e000 2020-10-01 16:26:04 UTC+0000
0x000000013e37ab30 SearchIndexer.     2896    512 0x000000007d35d000 2020-10-01 16:25:40 UTC+0000
0x000000013e3c4710 SearchProtocol     2972   2896 0x0000000086f7b000 2020-10-01 16:25:41 UTC+0000
0x000000013e3d57c0 SearchFilterHo     2992   2896 0x000000007be61000 2020-10-01 16:25:41 UTC+0000
```

PStree

In this plugin, the process list is represented with a child-parent relationship and shows any unknown or abnormal processes. The child process is represented by indention and periods.

```
volatility -f ram.mem --profile=Win7SP1x64 pstree
```



```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 pstree
Volatility Foundation Volatility Framework 2.6

Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0xfffffa80322d2a90:wininit.exe                       416    344      3     78 2020-1
. 0xfffffa8032332780:services.exe                     512    416     11    229 2020-1
.. 0xfffffa80324a9b30:svchost.exe                     128    512     12    550 2020-1
.. 0xfffffa80325331b0:spoolsv.exe                    1040    512     14    289 2020-1
.. 0xfffffa80323e7b30:svchost.exe                     704    512      7    289 2020-1
.. 0xfffffa803282db30:msdtc.exe                      2000    512     16    158 2020-1
.. 0xfffffa8032661b30:VGAuthService.                 1308    512      5    100 2020-1
.. 0xfffffa803254a060:svchost.exe                    1084    512     20    340 2020-1
.. 0xfffffa8030f2f4f0:svchost.exe                    1600    512      8     97 2020-1
```

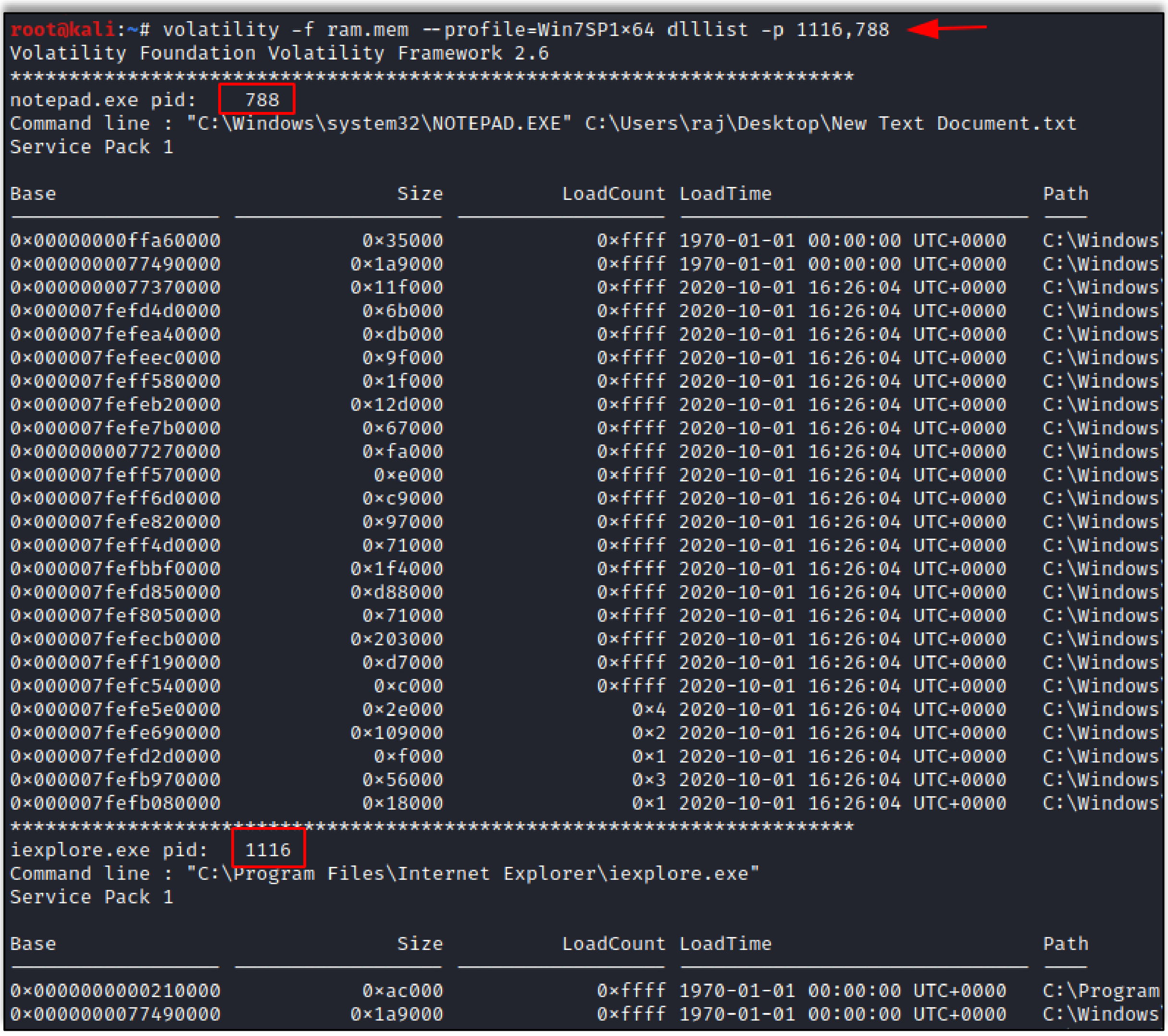
4. DLL

It is extremely important to know which DLLs (Dynamic Linked Libraries) are imported into the process while analysing the memory dump. A DLL can contain malicious executable code that may have a benign process to introduce malicious activity. Therefore, examining the various processes for the presence of malicious DLLs or similar code injections is crucial for analysis. Volatility has various types of plugins for this analysis.

DLLList

Various tools only have the potential to detect the DLLs which are used by a process by consulting the first of the three DLL lists stored in the PEB, which tracks the order in which each DLL is loaded. As a result, malware will sometimes modify that list to hide the presence of a DLL. Volatility has a plugin that also parses this same list, which can be run with the following command:

```
volatility -f ram.mem --profile=Win7SP1x64 dlllist -p 116,788
```

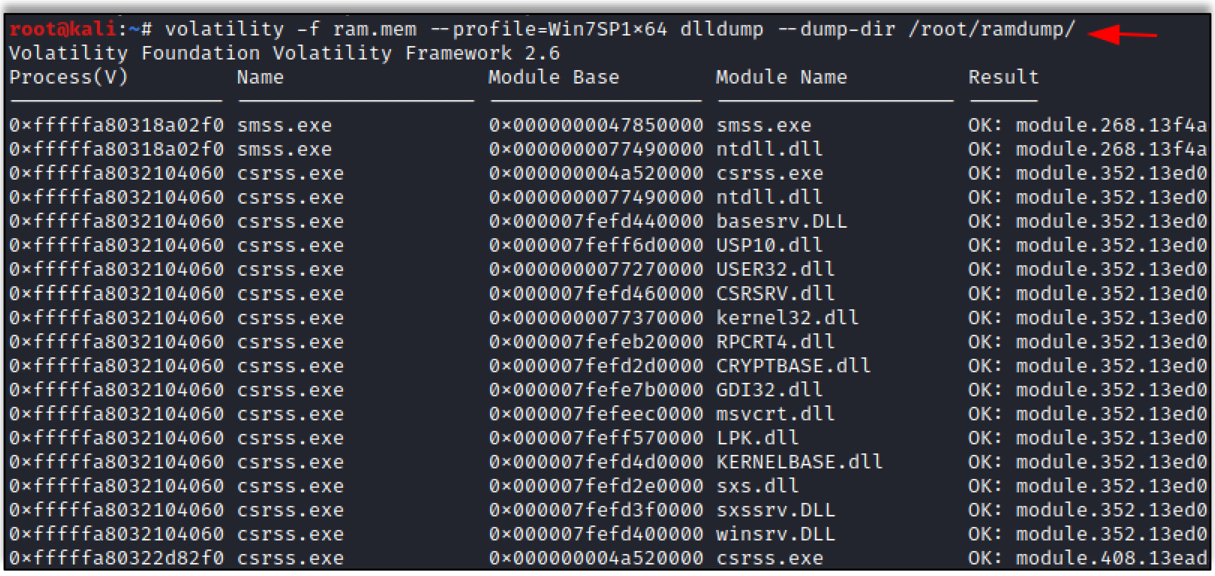


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 dlllist -p 1116,788
Volatility Foundation Volatility Framework 2.6
************************************************************************
notepad.exe pid:    788
Command line : "C:\Windows\system32\NOTEPAD.EXE" C:\Users\raj\Desktop\New Text Document.txt
Service Pack 1

Base                             Size          LoadCount  LoadTime                       Path
------------------ ------------------ ------------------ ------------------------------ ----
0x00000000ffa60000            0x35000             0xffff 1970-01-01 00:00:00 UTC+0000   C:\Windows
0x0000000077490000           0x1a9000             0xffff 1970-01-01 00:00:00 UTC+0000   C:\Windows
0x0000000077370000           0x11f000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefd4d0000            0x6b000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefea40000            0xdb000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefeec0000            0x9f000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007feff580000            0x1f000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefeb20000           0x12d000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefe7b0000            0x67000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x0000000077270000            0xfa000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007feff570000             0xe000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007feff6d0000            0xc9000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefe820000            0x97000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007feff4d0000            0x71000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefbbf0000           0x1f4000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefd850000           0xd88000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fef8050000            0x71000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefecb0000           0x203000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007feff190000            0xd7000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefc540000             0xc000             0xffff 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefe5e0000            0x2e000                0x4 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefe690000           0x109000                0x2 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefd2d0000             0xf000                0x1 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefb970000            0x56000                0x3 2020-10-01 16:26:04 UTC+0000   C:\Windows
0x000007fefb080000            0x18000                0x1 2020-10-01 16:26:04 UTC+0000   C:\Windows
************************************************************************
iexplore.exe pid:   1116
Command line : "C:\Program Files\Internet Explorer\iexplore.exe"
Service Pack 1

Base                             Size          LoadCount  LoadTime                       Path
------------------ ------------------ ------------------ ------------------------------ ----
0x0000000000210000            0xac000             0xffff 1970-01-01 00:00:00 UTC+0000   C:\Program
0x0000000077490000           0x1a9000             0xffff 1970-01-01 00:00:00 UTC+0000   C:\Windows
```

DLLDump

This plugin is used to dump the DLLs from the memory space of the processes into another location to analyze it. To take a dump of the DLLs you can type,

```
volatility -f ram.mem --profile=Win7SP1x64 dlldump -dump-dir
```

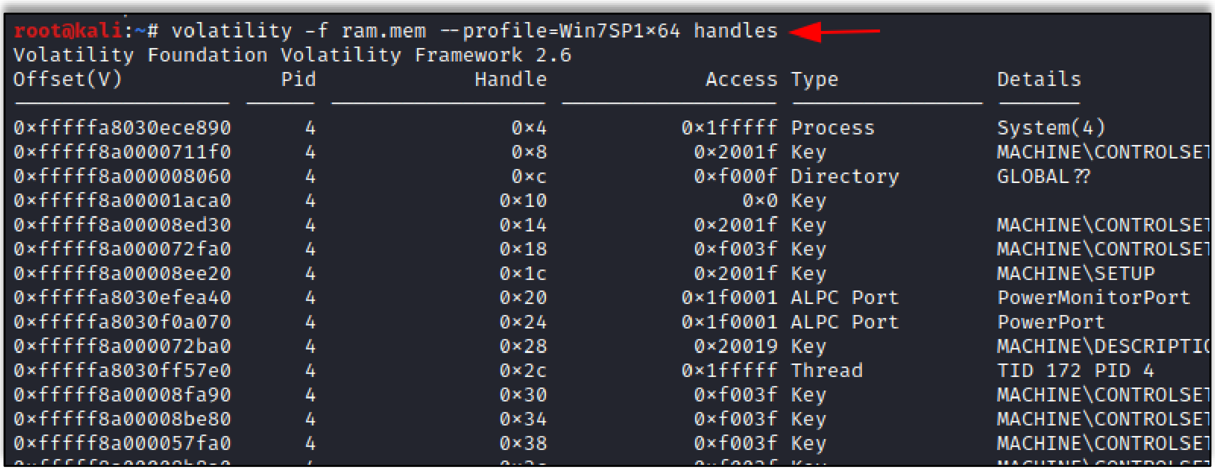


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 dlldump --dump-dir /root/ramdump/
Volatility Foundation Volatility Framework 2.6
Process(V)         Name                 Module Base        Module Name          Result
------------------ -------------------- ------------------ -------------------- ------
0xfffffa80318a02f0 smss.exe             0x0000000047850000 smss.exe             OK: module.268.13f4a
0xfffffa80318a02f0 smss.exe             0x0000000077490000 ntdll.dll            OK: module.268.13f4a
0xfffffa8032104060 csrss.exe            0x000000004a520000 csrss.exe            OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x0000000077490000 ntdll.dll            OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007fefd440000 basesrv.DLL          OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007feff6d0000 USP10.dll            OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x0000000077270000 USER32.dll           OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007fefd460000 CSRSRV.dll           OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x0000000077370000 kernel32.dll         OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007fefeb20000 RPCRT4.dll           OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007fefd2d0000 CRYPTBASE.dll        OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007fefe7b0000 GDI32.dll            OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007fefeec0000 msvcrt.dll           OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007feff570000 LPK.dll              OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007fefd4d0000 KERNELBASE.dll       OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007fefd2e0000 sxs.dll              OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007fefd3f0000 sxssrv.DLL           OK: module.352.13ed0
0xfffffa8032104060 csrss.exe            0x000007fefd400000 winsrv.DLL           OK: module.352.13ed0
0xfffffa80322d82f0 csrss.exe            0x000000004a520000 csrss.exe            OK: module.408.13ead
```

5. Handles

This plugin is used to display the open handles that are present in a process. This plugin applies to files, registry keys, events, desktops, threads, and all other types of objects. To see the handles, present in the dump, you can type,

```
volatility -f ram.mem --profile=Win7SP1x64 handles
```

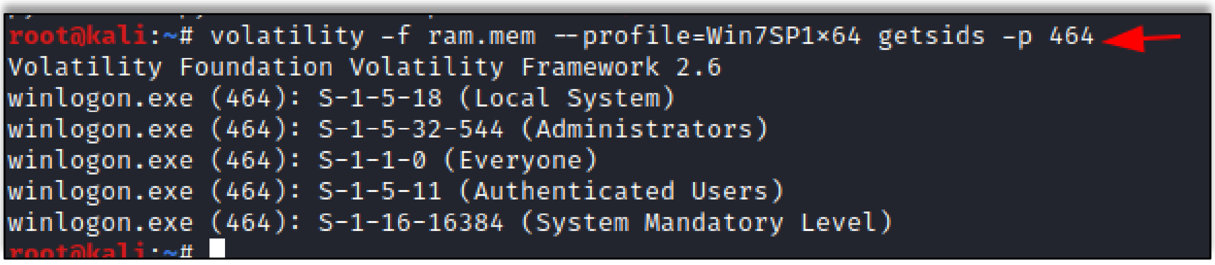


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 handles
Volatility Foundation Volatility Framework 2.6
Offset(V)             Pid             Handle             Access Type             Details
------------------ ------ ------------------ ---------- ---------------- -------
0xfffffa8030ece890      4                0x4    0x1ffff Process          System(4)
0xfffff8a0000711f0      4                0x8    0x2001f Key              MACHINE\CONTROLSET
0xfffff8a000008060      4                0xc    0xf000f Directory        GLOBAL??
0xfffff8a00001aca0      4               0x10        0x0 Key
0xfffff8a00008ed30      4               0x14    0x2001f Key              MACHINE\CONTROLSET
0xfffff8a000072fa0      4               0x18    0xf003f Key              MACHINE\CONTROLSET
0xfffff8a00008ee20      4               0x1c    0x2001f Key              MACHINE\SETUP
0xfffffa8030efea40      4               0x20   0x1f0001 ALPC Port        PowerMonitorPort
0xfffffa8030f0a070      4               0x24   0x1f0001 ALPC Port        PowerPort
0xfffff8a000072ba0      4               0x28    0x20019 Key              MACHINE\DESCRIPTI
0xfffffa8030ff57e0      4               0x2c    0x1ffff Thread           TID 172 PID 4
0xfffff8a00008fa90      4               0x30    0xf003f Key              MACHINE\CONTROLSET
0xfffff8a00008be80      4               0x34    0xf003f Key              MACHINE\CONTROLSET
0xfffff8a000057fa0      4               0x38    0xf003f Key              MACHINE\CONTROLSET
```

6. Getsids

This plugin is used to view the SIDs stands for Security Identifiers that are associated with a process. This plugin can help in identifying processes that have maliciously escalated privileges and which processes belong to specific users. To get detail on a particular process id, you can type:

```
volatility -f ram.mem --profile=Win7SP1x64 getsids -p 464
```

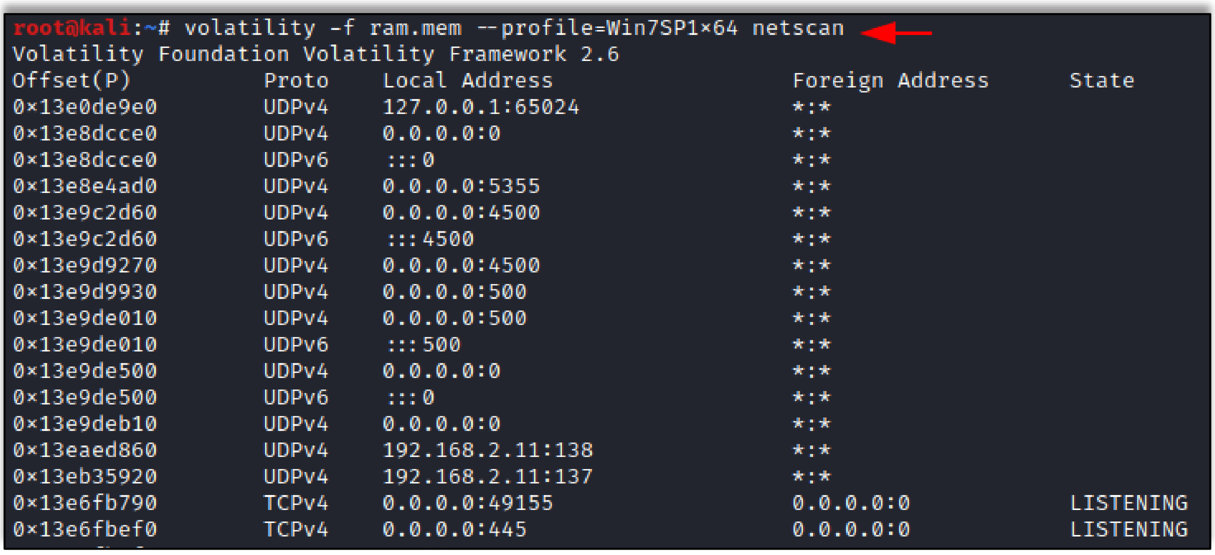


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 getsids -p 464
Volatility Foundation Volatility Framework 2.6
winlogon.exe (464): S-1-5-18 (Local System)
winlogon.exe (464): S-1-5-32-544 (Administrators)
winlogon.exe (464): S-1-1-0 (Everyone)
winlogon.exe (464): S-1-5-11 (Authenticated Users)
winlogon.exe (464): S-1-16-16384 (System Mandatory Level)
root@kali:~#
```

7. Netscan

This plugin helps in finding network-related artifacts present in the memory dump. It makes use of pool tag scanning. This plugin finds all the TCP endpoints, TCP listeners, UDP endpoints, and UDP listeners. It provides details about the local and remote IP and also about the local and remote port. To get details on the network artifacts, you can type:

```
volatility -f ram.mem --profile=Win7SP1x64 netscan
```

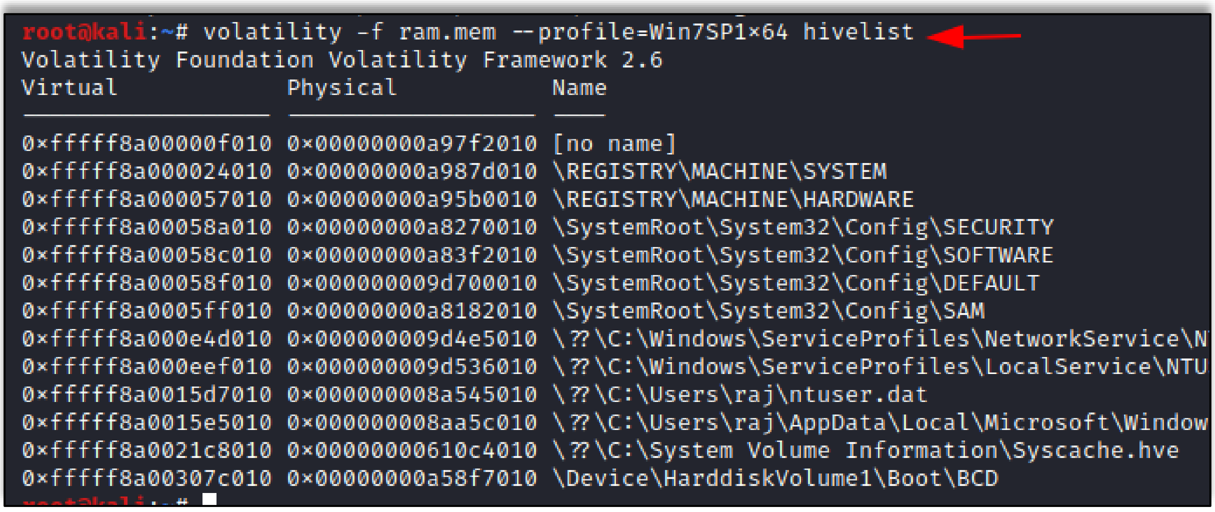


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 netscan
Volatility Foundation Volatility Framework 2.6
Offset(P)          Proto    Local Address                  Foreign Address      State
0x13e0de9e0        UDPv4    127.0.0.1:65024                *:*
0x13e8dcce0        UDPv4    0.0.0.0:0                      *:*
0x13e8dcce0        UDPv6    :::0                           *:*
0x13e8e4ad0        UDPv4    0.0.0.0:5355                   *:*
0x13e9c2d60        UDPv4    0.0.0.0:4500                   *:*
0x13e9c2d60        UDPv6    :::4500                        *:*
0x13e9d9270        UDPv4    0.0.0.0:4500                   *:*
0x13e9d9930        UDPv4    0.0.0.0:500                    *:*
0x13e9de010        UDPv4    0.0.0.0:500                    *:*
0x13e9de010        UDPv6    :::500                         *:*
0x13e9de500        UDPv4    0.0.0.0:0                      *:*
0x13e9de500        UDPv6    :::0                           *:*
0x13e9deb10        UDPv4    0.0.0.0:0                      *:*
0x13eaed860        UDPv4    192.168.2.11:138               *:*
0x13eb35920        UDPv4    192.168.2.11:137               *:*
0x13e6fb790        TCPv4    0.0.0.0:49155                  0.0.0.0:0            LISTENING
0x13e6fbef0        TCPv4    0.0.0.0:445                    0.0.0.0:0            LISTENING
```

8. Hivelist

This plugin can be used to locate the virtual addresses present in the registry hives in memory, and their entire paths to hive on the disk. To obtain the details on the hivelist from the memory dump, you can type:

```
volatility -f ram.mem --profile=Win7SP1x64 hivelist
```

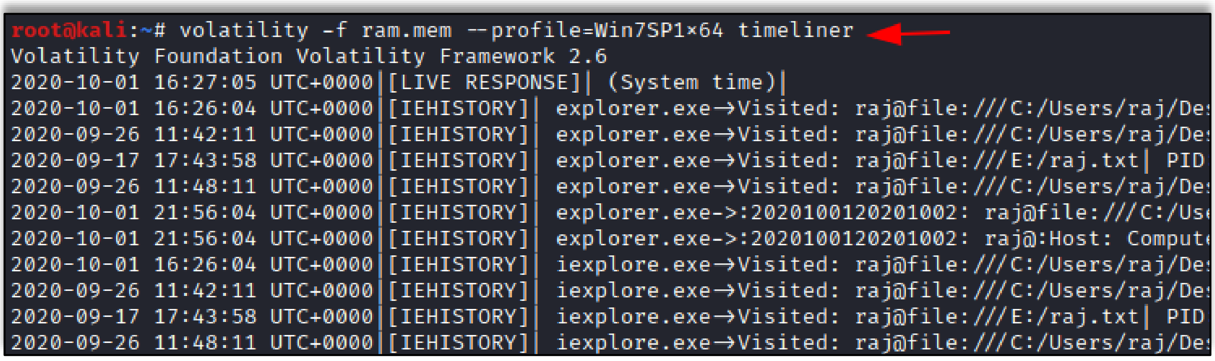


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual            Physical           Name
------------------ ------------------ ----
0xfffff8a00000f010 0x00000000a97f2010 [no name]
0xfffff8a000024010 0x00000000a987d010 \REGISTRY\MACHINE\SYSTEM
0xfffff8a000057010 0x00000000a95b0010 \REGISTRY\MACHINE\HARDWARE
0xfffff8a00058a010 0x00000000a8270010 \SystemRoot\System32\Config\SECURITY
0xfffff8a00058c010 0x00000000a83f2010 \SystemRoot\System32\Config\SOFTWARE
0xfffff8a00058f010 0x000000009d700010 \SystemRoot\System32\Config\DEFAULT
0xfffff8a0005ff010 0x00000000a8182010 \SystemRoot\System32\Config\SAM
0xfffff8a000e4d010 0x000000009d4e5010 \??\C:\Windows\ServiceProfiles\NetworkService\N
0xfffff8a000eef010 0x000000009d536010 \??\C:\Windows\ServiceProfiles\LocalService\NTU
0xfffff8a0015d7010 0x000000008a545010 \??\C:\Users\raj\ntuser.dat
0xfffff8a0015e5010 0x000000008aa5c010 \??\C:\Users\raj\AppData\Local\Microsoft\Window
0xfffff8a0021c8010 0x00000000610c4010 \??\C:\System Volume Information\Syscache.hve
0xfffff8a00307c010 0x00000000a58f7010 \Device\HarddiskVolume1\Boot\BCD
```

9. Timeliner

This plugin usually creates a timeline from the various artifacts found in the memory dump. To locate the artifacts according to the timeline, you can use the following command:

```
volatility -f ram.mem --profile=Win7SP1x64 timeliner
```

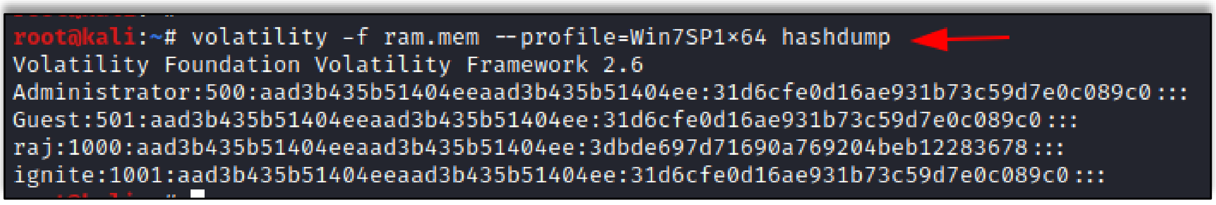


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 timeliner
Volatility Foundation Volatility Framework 2.6
2020-10-01 16:27:05 UTC+0000|[LIVE RESPONSE]| (System time)|
2020-10-01 16:26:04 UTC+0000|[IEHISTORY]| explorer.exe->Visited: raj@file:///C:/Users/raj/De
2020-09-26 11:42:11 UTC+0000|[IEHISTORY]| explorer.exe->Visited: raj@file:///C:/Users/raj/De
2020-09-17 17:43:58 UTC+0000|[IEHISTORY]| explorer.exe->Visited: raj@file:///E:/raj.txt| PID
2020-09-26 11:48:11 UTC+0000|[IEHISTORY]| explorer.exe->Visited: raj@file:///C:/Users/raj/De
2020-10-01 21:56:04 UTC+0000|[IEHISTORY]| explorer.exe->:2020100120201002: raj@file:///C:/Use
2020-10-01 21:56:04 UTC+0000|[IEHISTORY]| explorer.exe->:2020100120201002: raj@:Host: Compute
2020-10-01 16:26:04 UTC+0000|[IEHISTORY]| iexplore.exe->Visited: raj@file:///C:/Users/raj/De
2020-09-26 11:42:11 UTC+0000|[IEHISTORY]| iexplore.exe->Visited: raj@file:///C:/Users/raj/De
2020-09-17 17:43:58 UTC+0000|[IEHISTORY]| iexplore.exe->Visited: raj@file:///E:/raj.txt| PID
2020-09-26 11:48:11 UTC+0000|[IEHISTORY]| iexplore.exe->Visited: raj@file:///C:/Users/raj/De
```

10. HashDump

This plugin can be used to extract and decrypt cached domain credentials stored in the registry which can be availed from the memory dump. The hashes that are availed from the memory dump can be cracked using John the Ripper, Hashcat, etc. To gather the hashdump, you can use the command:

```
volatility -f ram.mem --profile=Win7SP1x64 hashdump
```

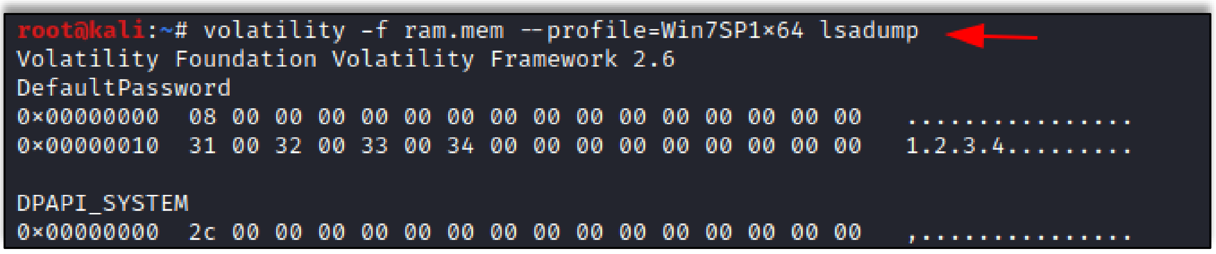


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 hashdump
Volatility Foundation Volatility Framework 2.6
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
raj:1000:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
ignite:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

11. Lsadump

This plugin is used to dump LSA secrets from the registry in the memory dump. This plugin gives out information like the default password, the RDP public key, etc. To perform a lsadump, you can type the following command:

```
volatility -f ram.mem --profile=Win7SP1x64 lsadump
```

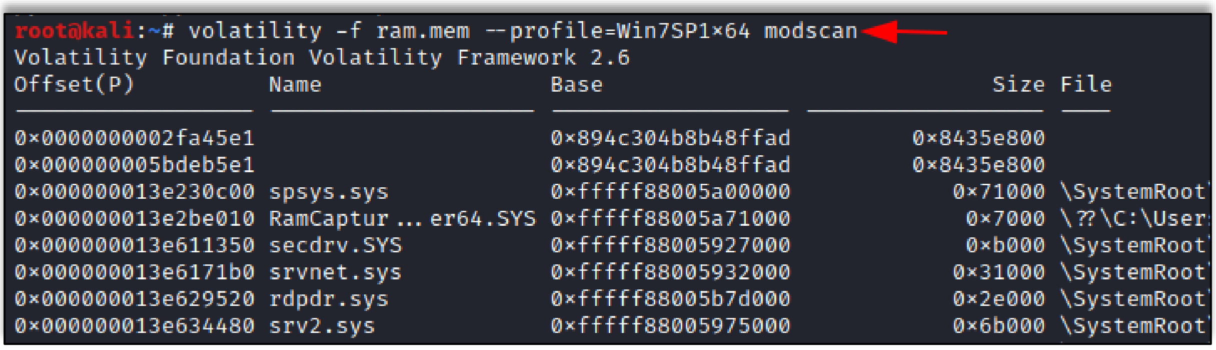


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 lsadump
Volatility Foundation Volatility Framework 2.6
DefaultPassword
0x00000000  08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00000010  31 00 32 00 33 00 34 00 00 00 00 00 00 00 00 00   1.2.3.4.........

DPAPI_SYSTEM
0x00000000  2c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ,...............
```

12. Modscan

This plugin is used to locate kernel memory and its related objects. It can pick up all the previously unloaded drivers and also those drivers that have been hidden or have been unlinked by rootkits in the system.

```
volatility -f ram.mem --profile=Win7SP1x64 modscan
```

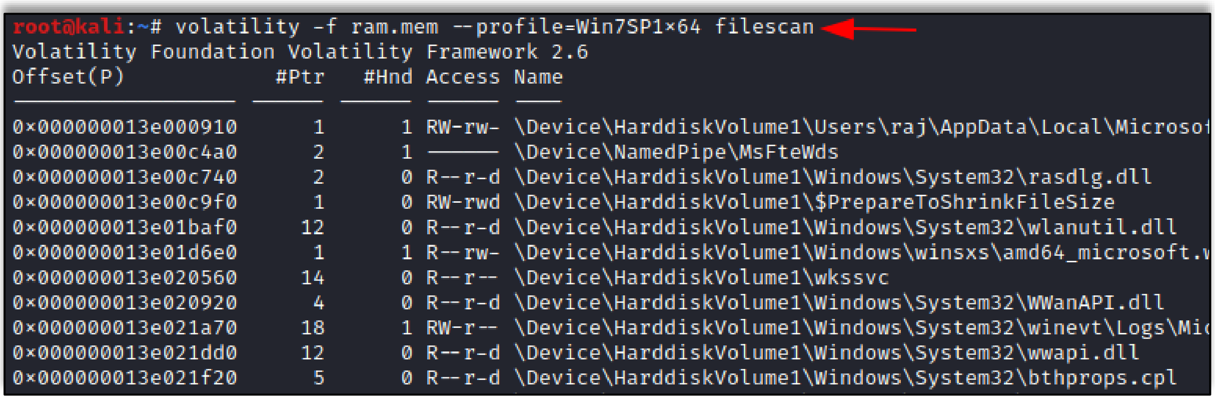


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 modscan
Volatility Foundation Volatility Framework 2.6
Offset(P)          Name                 Base                             Size File
------------------ -------------------- ------------------ ------------------ ----

0x0000000002fa45e1                      0x894c304b8b48ffad         0x8435e800
0x000000005bdeb5e1                      0x894c304b8b48ffad         0x8435e800
0x000000013e230c00 spsys.sys            0xfffff88005a00000            0x71000 \SystemRoot
0x000000013e2be010 RamCaptur...er64.SYS 0xfffff88005a71000             0x7000 \??\C:\User
0x000000013e611350 secdrv.SYS           0xfffff88005927000             0xb000 \SystemRoot
0x000000013e6171b0 srvnet.sys           0xfffff88005932000            0x31000 \SystemRoot
0x000000013e629520 rdpdr.sys            0xfffff88005b7d000            0x2e000 \SystemRoot
0x000000013e634480 srv2.sys             0xfffff88005975000            0x6b000 \SystemRoot
```

13. FileScan

This plugin is used to find FILE_OBJECTs present in the physical memory by using pool tag scanning. It can find open files even if there is a hidden rootkit present in the files. To make use of this plugin, you can type the following command:

```
volatility -f ram.mem --profile=Win7SP1x64 filescan
```

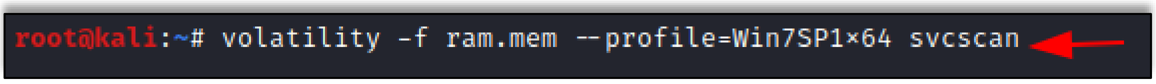


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 filescan
Volatility Foundation Volatility Framework 2.6
Offset(P)            #Ptr   #Hnd Access Name
------------------ ------ ------ ------ ----
0x000000013e000910      1      1 RW-rw- \Device\HarddiskVolume1\Users\raj\AppData\Local\Microsof
0x000000013e00c4a0      2      1 ------ \Device\NamedPipe\MsFteWds
0x000000013e00c740      2      0 R--r-d \Device\HarddiskVolume1\Windows\System32\rasdlg.dll
0x000000013e00c9f0      1      0 RW-rwd \Device\HarddiskVolume1\$PrepareToShrinkFileSize
0x000000013e01baf0     12      0 R--r-d \Device\HarddiskVolume1\Windows\System32\wlanutil.dll
0x000000013e01d6e0      1      1 R--rw- \Device\HarddiskVolume1\Windows\winsxs\amd64_microsoft.w
0x000000013e020560     14      0 R--r-- \Device\HarddiskVolume1\wkssvc
0x000000013e020920      4      0 R--r-d \Device\HarddiskVolume1\Windows\System32\WWanAPI.dll
0x000000013e021a70     18      1 RW-r-- \Device\HarddiskVolume1\Windows\System32\winevt\Logs\Mic
0x000000013e021dd0     12      0 R--r-d \Device\HarddiskVolume1\Windows\System32\wwapi.dll
0x000000013e021f20      5      0 R--r-d \Device\HarddiskVolume1\Windows\System32\bthprops.cpl
```

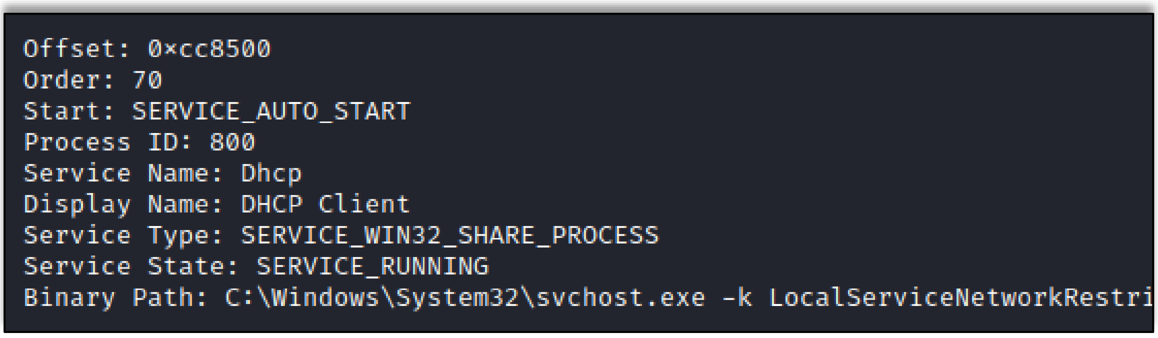
14. Svcscan

This plugin is used to see the services are registered on your memory image, use the svcscan command. The output shows the process ID of each service the service name, service name, display name, service type, service state, and also shows the binary path for the registered service – which will be a .exe for user mode services and a driver name for services that run from kernel mode. To find the details on the services

```
volatility -f ram.mem --profile=Win7SP1x64 svcscan
```



```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 svcscan
```

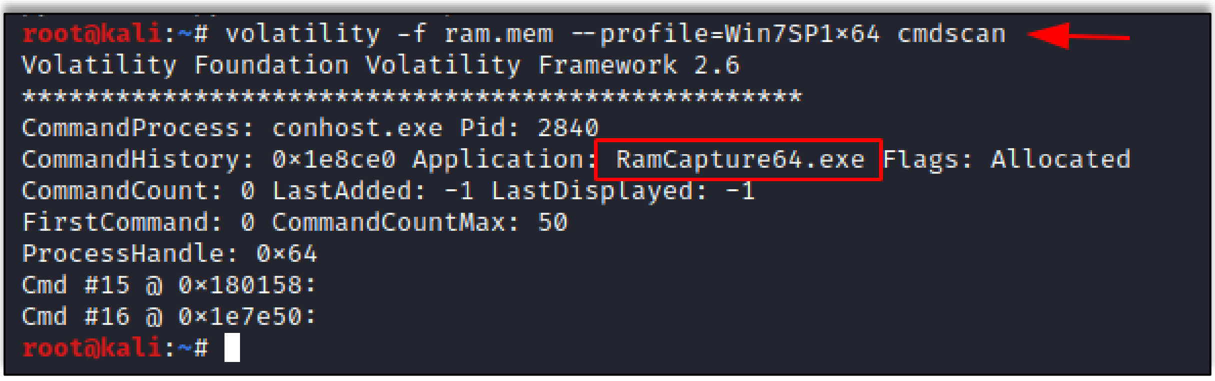


```
Offset: 0xcc8500
Order: 70
Start: SERVICE_AUTO_START
Process ID: 800
Service Name: Dhcp
Display Name: DHCP Client
Service Type: SERVICE_WIN32_SHARE_PROCESS
Service State: SERVICE_RUNNING
Binary Path: C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestri
```

15. Cmdscan

This plugin searches the memory dump of XP/2003/Vista/2008 and Windows 7 for commands that the attacker might have entered through a command prompt (cmd.exe). It is one of the most powerful commands that one can use to gain visibility into an attacker's actions on a victim system. To conduct a cmdscan, you can make use of the following command:

```
volatility -f ram.mem --profile=Win7SP1x64 cmdscan
```

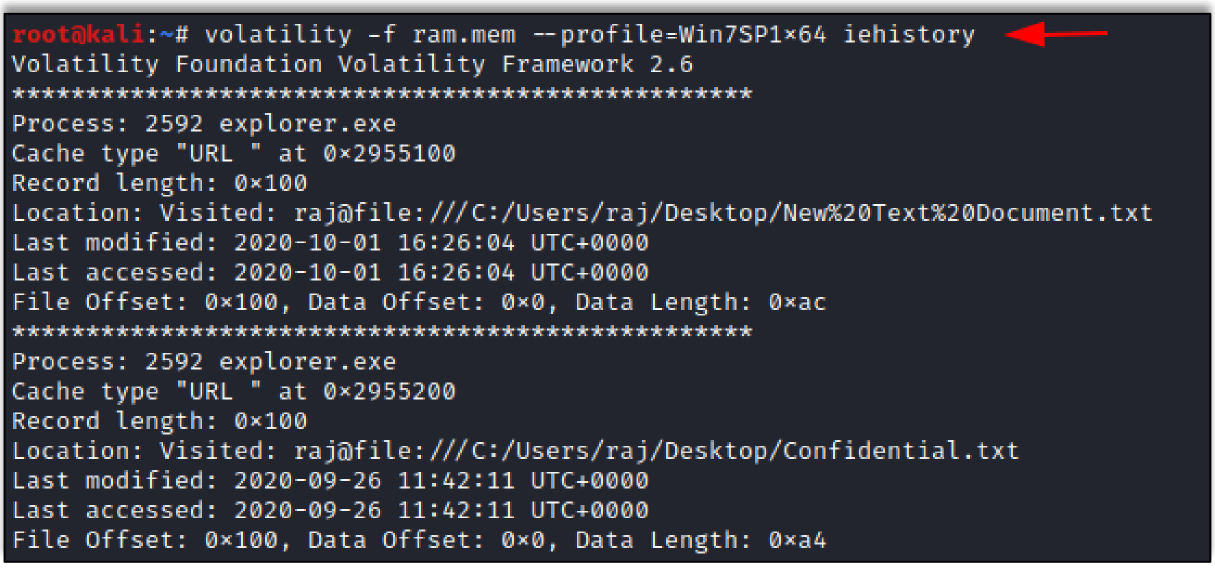


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 cmdscan
Volatility Foundation Volatility Framework 2.6
**************************************************
CommandProcess: conhost.exe Pid: 2840
CommandHistory: 0x1e8ce0 Application: RamCapture64.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x64
Cmd #15 @ 0x180158:
Cmd #16 @ 0x1e7e50:
root@kali:~#
```

16. Iehistory

This plugin recovers the fragments of Internet Explorer history by finding index.dat cache file. To find iehistory files, you can type the following command:

```
volatility -f ram.mem --profile=Win7SP1x64 iehistory
```

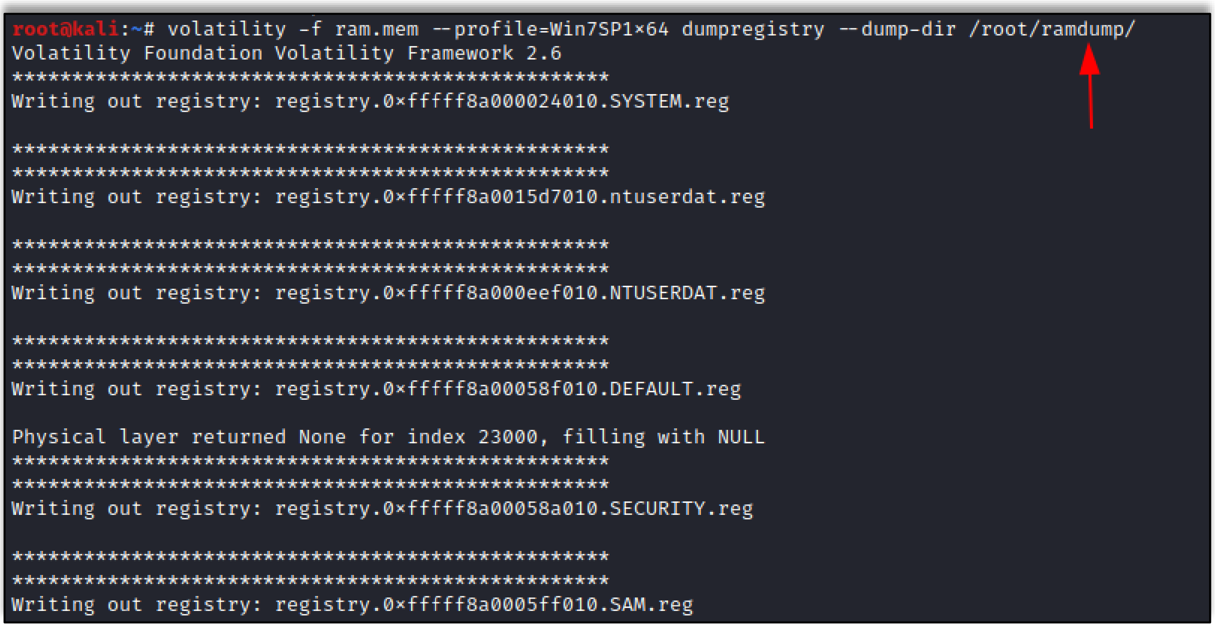


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 iehistory
Volatility Foundation Volatility Framework 2.6
**************************************************
Process: 2592 explorer.exe
Cache type "URL " at 0x2955100
Record length: 0x100
Location: Visited: raj@file:///C:/Users/raj/Desktop/New%20Text%20Document.txt
Last modified: 2020-10-01 16:26:04 UTC+0000
Last accessed: 2020-10-01 16:26:04 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0xac
**************************************************
Process: 2592 explorer.exe
Cache type "URL " at 0x2955200
Record length: 0x100
Location: Visited: raj@file:///C:/Users/raj/Desktop/Confidential.txt
Last modified: 2020-09-26 11:42:11 UTC+0000
Last accessed: 2020-09-26 11:42:11 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0xa4
```

17. Dumpregistry

This plugin allows one to dump a registry hive into a disk location. To dump the registry hive, you use the following command.

```
volatility -f ram.mem --profile=Win7SP1x64 dumpregistry --dump-dir
/root/ramdump/
```

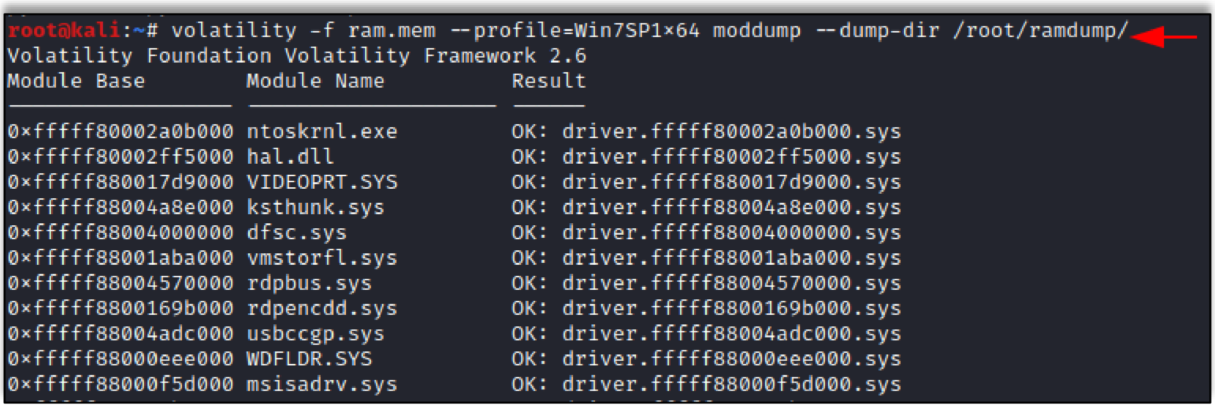


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 dumpregistry --dump-dir /root/ramdump/
Volatility Foundation Volatility Framework 2.6
**************************************************
Writing out registry: registry.0xfffff8a000024010.SYSTEM.reg

**************************************************
**************************************************
Writing out registry: registry.0xfffff8a0015d7010.ntuserdat.reg

**************************************************
**************************************************
Writing out registry: registry.0xfffff8a000eef010.NTUSERDAT.reg

**************************************************
**************************************************
Writing out registry: registry.0xfffff8a00058f010.DEFAULT.reg

Physical layer returned None for index 23000, filling with NULL
**************************************************
**************************************************
Writing out registry: registry.0xfffff8a00058a010.SECURITY.reg

**************************************************
**************************************************
Writing out registry: registry.0xfffff8a0005ff010.SAM.reg
```

18. Moddump

This plugin is used to extract a kernel driver to a file, you can do this by using the following command:

```
volatility -f ram.mem --profile=Win7SP1x64 moddump --dump-dir
/root/ramdump/
```

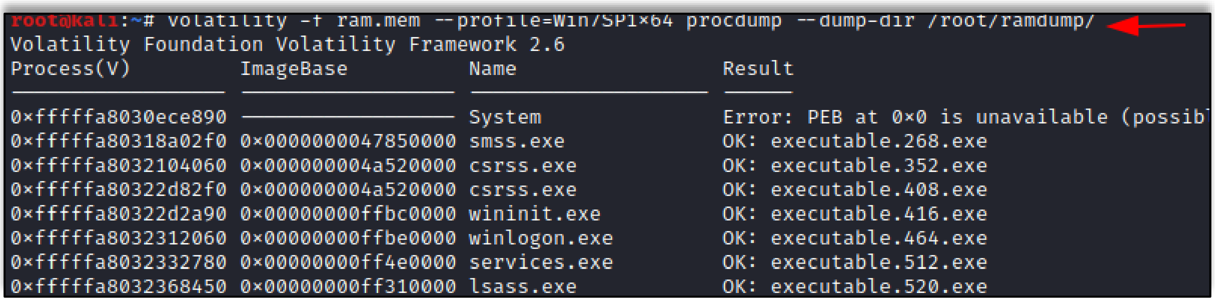


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 moddump --dump-dir /root/ramdump/
Volatility Foundation Volatility Framework 2.6
Module Base        Module Name          Result
------------------ -------------------- ------
0xfffff80002a0b000 ntoskrnl.exe         OK: driver.fffff80002a0b000.sys
0xfffff80002ff5000 hal.dll              OK: driver.fffff80002ff5000.sys
0xfffff880017d9000 VIDEOPRT.SYS         OK: driver.fffff880017d9000.sys
0xfffff88004a8e000 ksthunk.sys          OK: driver.fffff88004a8e000.sys
0xfffff88004000000 dfsc.sys             OK: driver.fffff88004000000.sys
0xfffff88001aba000 vmstorfl.sys         OK: driver.fffff88001aba000.sys
0xfffff88004570000 rdpbus.sys           OK: driver.fffff88004570000.sys
0xfffff8800169b000 rdpencdd.sys         OK: driver.fffff8800169b000.sys
0xfffff88004adc000 usbccgp.sys          OK: driver.fffff88004adc000.sys
0xfffff88000eee000 WDFLDR.SYS           OK: driver.fffff88000eee000.sys
0xfffff88000f5d000 msisadrv.sys         OK: driver.fffff88000f5d000.sys
```

19. Procdump

This plugin is used to dump the executable processes in a single location, If there is malware present it will intentionally forge size fields in the PE header for the memory dumping tool to fail. To collect the dump on processes, you can type:

```
volatility -f ram.mem --profile=Win7SP1x64 procdump --dump-dir /root/ramdump/
```

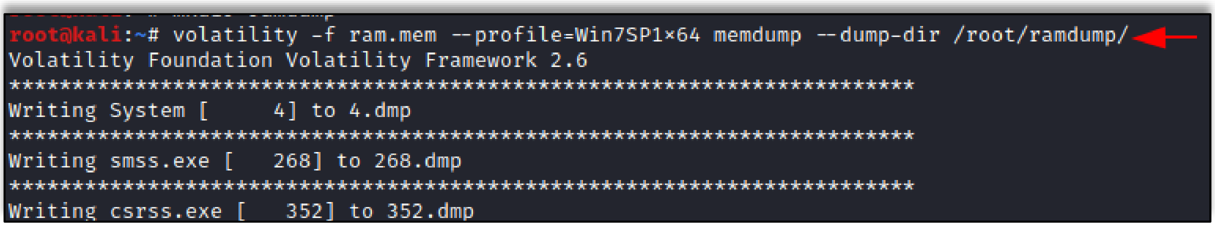


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 procdump --dump-dir /root/ramdump/
Volatility Foundation Volatility Framework 2.6
Process(V)         ImageBase          Name                 Result
------------------ ------------------ -------------------- ------
0xfffffa8030ece890 ------------------ System               Error: PEB at 0x0 is unavailable (possib
0xfffffa80318a02f0 0x0000000047850000 smss.exe             OK: executable.268.exe
0xfffffa8032104060 0x000000004a520000 csrss.exe            OK: executable.352.exe
0xfffffa80322d82f0 0x000000004a520000 csrss.exe            OK: executable.408.exe
0xfffffa80322d2a90 0x00000000ffbc0000 wininit.exe          OK: executable.416.exe
0xfffffa8032312060 0x00000000ffbe0000 winlogon.exe         OK: executable.464.exe
0xfffffa8032332780 0x00000000ff4e0000 services.exe         OK: executable.512.exe
0xfffffa8032368450 0x00000000ff310000 lsass.exe            OK: executable.520.exe
```

20. Memdump

The memdump plugin is used to dump the memory-resident pages of a process into a separate file. You can also lookup a particular process using -p and provide it with a directory path -D to generate the output. To take a dump on memory-resident pages, you can use the following command:

```
volatility -f ram.mem --profile=Win7SP1x64 memdump --dump-dir /root/ramdump/
```

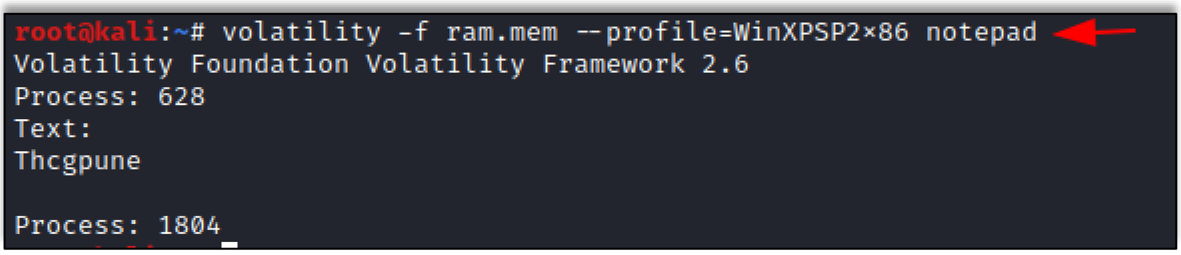


```
root@kali:~# volatility -f ram.mem --profile=Win7SP1x64 memdump --dump-dir /root/ramdump/
Volatility Foundation Volatility Framework 2.6
************************************************************************
Writing System [      4] to 4.dmp
************************************************************************
Writing smss.exe [    268] to 268.dmp
************************************************************************
Writing csrss.exe [    352] to 352.dmp
```

21. Notepad

Notepad files are usually highly looked up files in the ram dump. To find the contents present in the notepad file, you can use the following command:

```
volatility -f ram.mem --profile=Win7SP1x64 notepad
```



```
root@kali:~# volatility -f ram.mem --profile=WinXPSP2x86 notepad
Volatility Foundation Volatility Framework 2.6
Process: 628
Text:
Thcgpune

Process: 1804
```

PassMark Volatility Workbench

Volatility Workbench is a GUI version of one of the most popular tool Volatility for analyzing the artifacts from a memory dump. It is available free of cost, open-source, and runs on the Windows Operating system. You can download it from Here.

Features of Volatility Workbench

1. A forensic investigator does not have to worry about remembering the parameters of the command line.
2. It has made it easier to store dump information to a file on disk.
3. There is a drop-down list that contains the commands and its brief description.
4. It records the time stamp of the commands that were previously executed.

Download the tool and run it. Now choose the dump file that you have previously created and select the profile of the image that was created which could be used in place of imageinfo command. Now click on Refresh Process List and you can run all the commands.

1. Hunting rootkits and malicious code

It tends to run a scan on the memory dump and looks around for the presence of a rootkit or a malicious code that would not be easily seen in the system but could be running in the background.

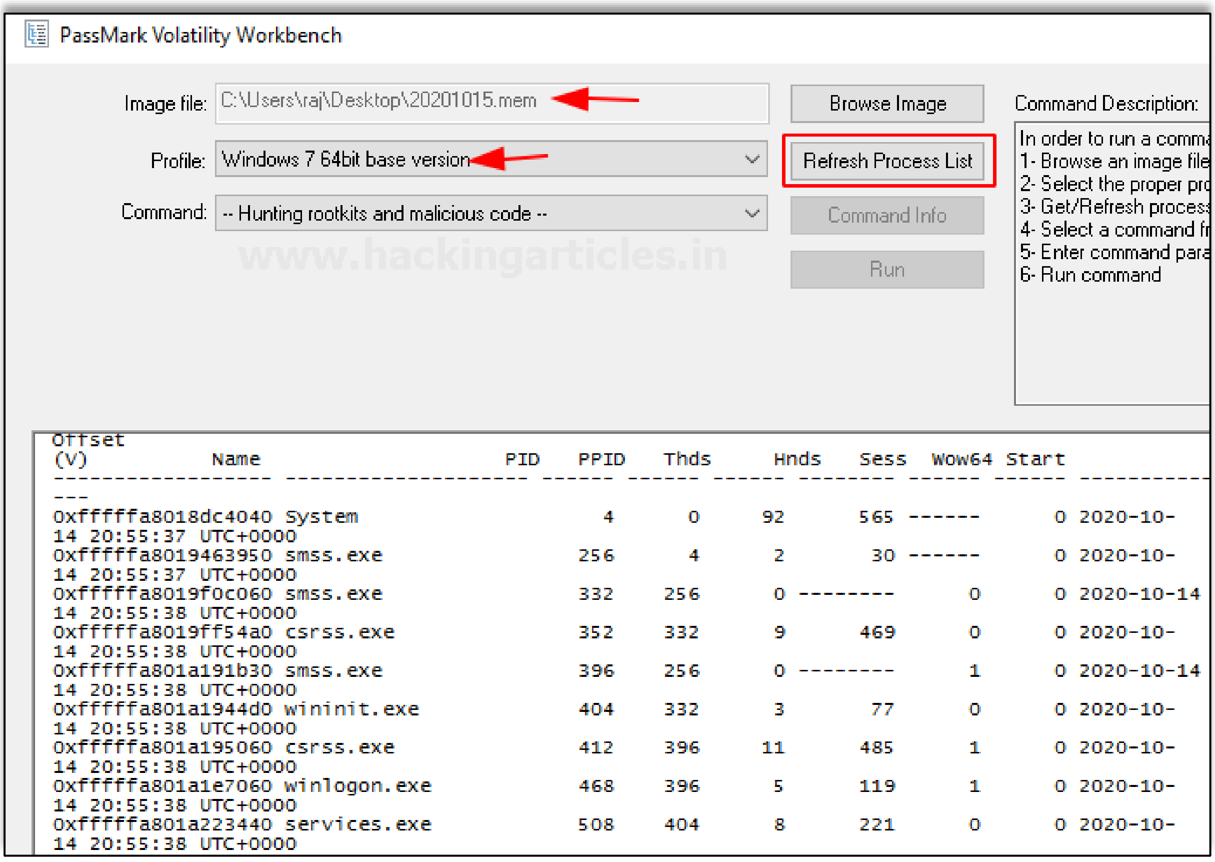


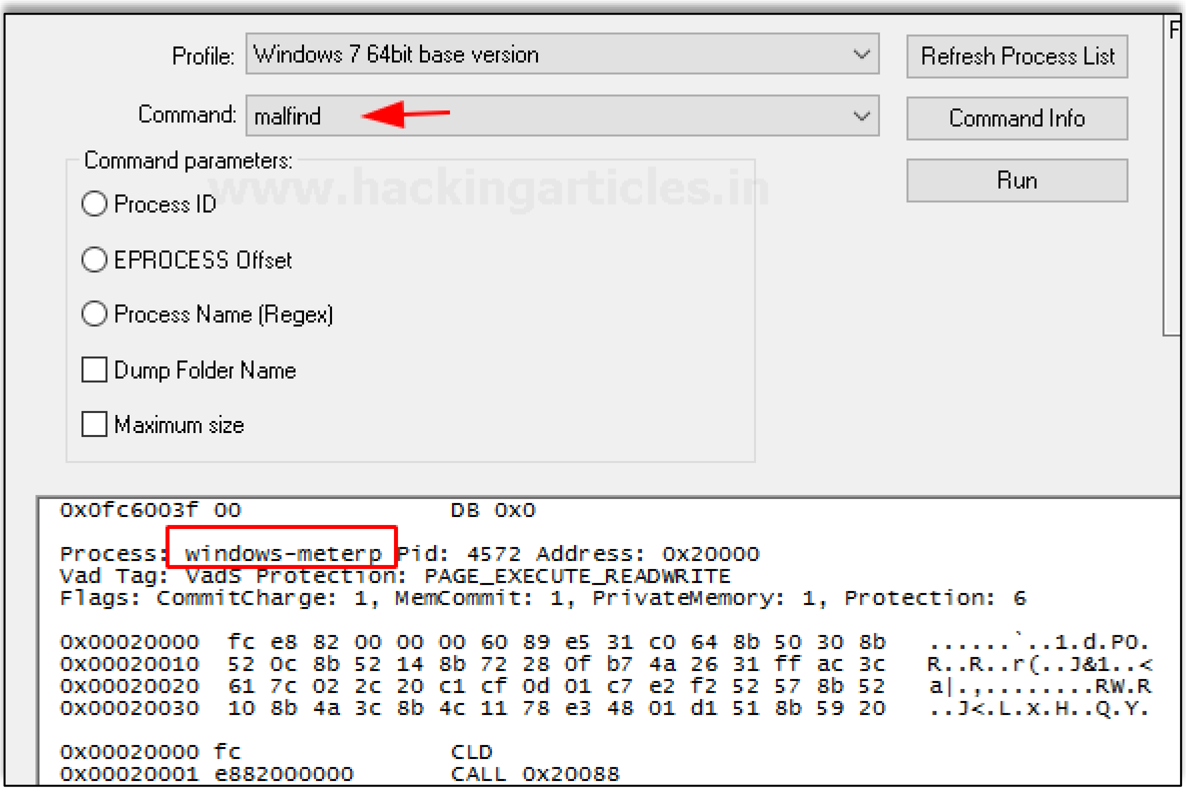
Image file: C:\Users\raj\Desktop\20201015.mem
Profile: Windows 7 64bit base version
Command: -- Hunting rootkits and malicious code --

In order to run a comma
1- Browse an image file
2- Select the proper pro
3- Get/Refresh process
4- Select a command fr
5- Enter command para
6- Run command

```
Offset
(V)               Name                     PID    PPID   Thds     Hnds   Sess  Wow64  Start
------------------ -------------------- ------ ------ ------ -------- ------ ------ ----------
---
0xfffffa8018dc4040 System                    4      0     92      565 ------      0 2020-10-
14 20:55:37 UTC+0000
0xfffffa8019463950 smss.exe                256      4      2       30 ------      0 2020-10-
14 20:55:37 UTC+0000
0xfffffa8019f0c060 smss.exe                332    256      0 --------      0      0 2020-10-14
14 20:55:38 UTC+0000
0xfffffa8019ff54a0 csrss.exe               352    332      9      469      0      0 2020-10-
14 20:55:38 UTC+0000
0xfffffa801a191b30 smss.exe                396    256      0 --------      1      0 2020-10-14
14 20:55:38 UTC+0000
0xfffffa801a1944d0 wininit.exe             404    332      3       77      0      0 2020-10-
14 20:55:38 UTC+0000
0xfffffa801a195060 csrss.exe               412    396     11      485      1      0 2020-10-
14 20:55:38 UTC+0000
0xfffffa801a1e7060 winlogon.exe            468    396      5      119      1      0 2020-10-
14 20:55:38 UTC+0000
0xfffffa801a223440 services.exe            508    404      8      221      0      0 2020-10-
14 20:55:38 UTC+0000
```

2. Malfind

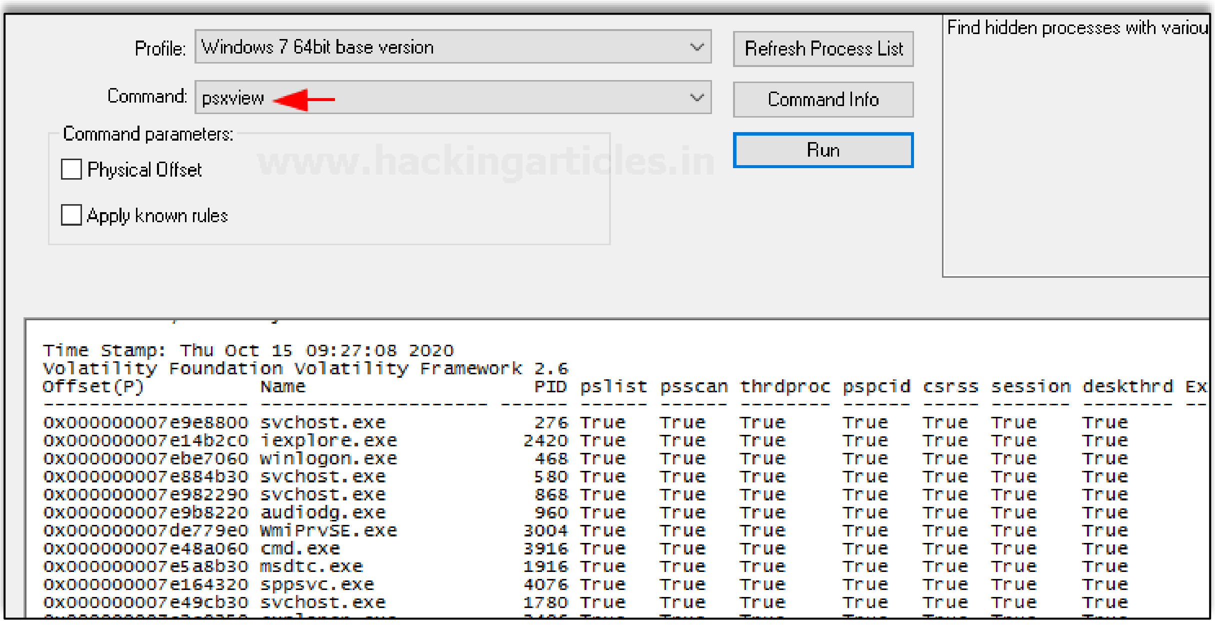
It is a command which helps in finding a hidden code or a code that has been injected into the user's memory. It doesn't generally detect the presence of a DLL in a process but instead locates them.



```
0x0fc6003f 00                    DB 0x0

Process: windows-meterp Pid: 4572 Address: 0x20000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00020000  fc e8 82 00 00 00 60 89 e5 31 c0 64 8b 50 30 8b   ......`..1.d.P0.
0x00020010  52 0c 8b 52 14 8b 72 28 0f b7 4a 26 31 ff ac 3c   R..R..r(..J&1..<
0x00020020  61 7c 02 2c 20 c1 cf 0d 01 c7 e2 f2 52 57 8b 52   a|.,........RW.R
0x00020030  10 8b 4a 3c 8b 4c 11 78 e3 48 01 d1 51 8b 59 20   ..J<.L.x.H..Q.Y.

0x00020000 fc                    CLD
0x00020001 e882000000            CALL 0x20088
```

3. psxview

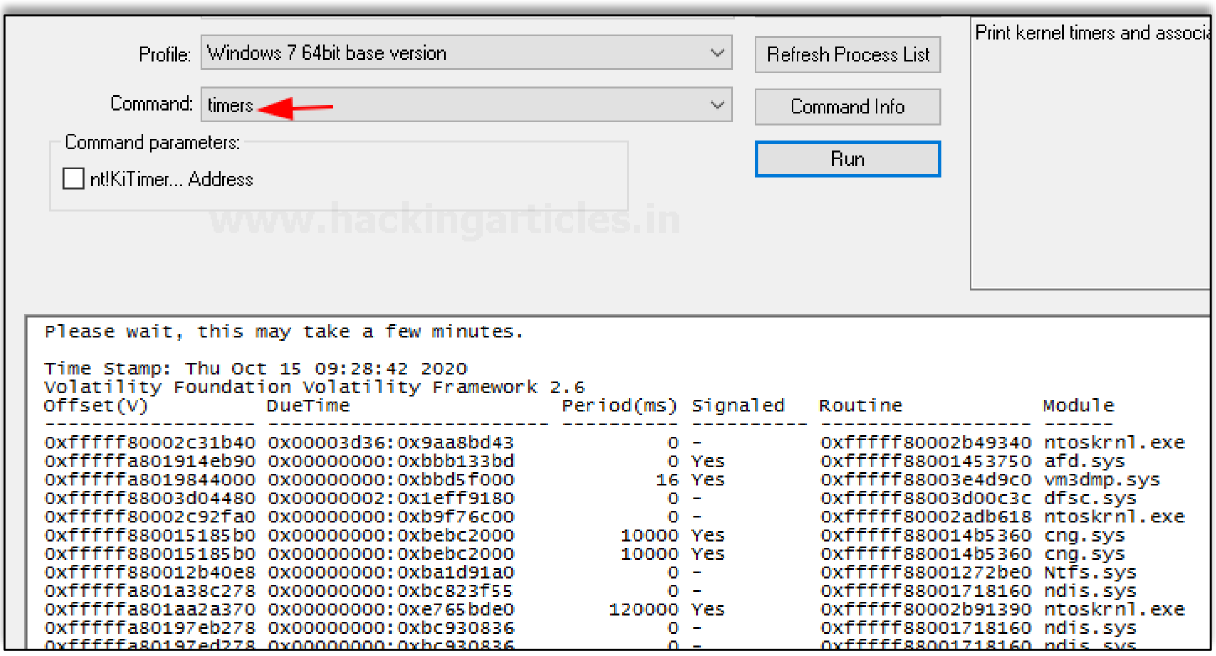
This command usually helps in discovering any hidden processes in the plugin present in the memory dump.



```
Time Stamp: Thu Oct 15 09:27:08 2020
Volatility Foundation Volatility Framework 2.6
Offset(P)          Name                    PID pslist psscan thrdproc pspcid csrss session deskthrd Ex
------------------ -------------------- ------ ------ ------ -------- ------ ----- ------- -------- --
0x000000007e9e8800 svchost.exe             276 True   True   True     True   True  True    True
0x000000007e14b2c0 iexplore.exe           2420 True   True   True     True   True  True    True
0x000000007ebe7060 winlogon.exe            468 True   True   True     True   True  True    True
0x000000007e884b30 svchost.exe             580 True   True   True     True   True  True    True
0x000000007e982290 svchost.exe             868 True   True   True     True   True  True    True
0x000000007e9b8220 audiodg.exe             960 True   True   True     True   True  True    True
0x000000007de779e0 WmiPrvSE.exe           3004 True   True   True     True   True  True    True
0x000000007e48a060 cmd.exe                3916 True   True   True     True   True  True    True
0x000000007e5a8b30 msdtc.exe              1916 True   True   True     True   True  True    True
0x000000007e164320 sppsvc.exe             4076 True   True   True     True   True  True    True
0x000000007e49cb30 svchost.exe            1780 True   True   True     True   True  True    True
```

4. Timers

It displays the timer of the kernel and all the associated timers present in the memory dump of the system.

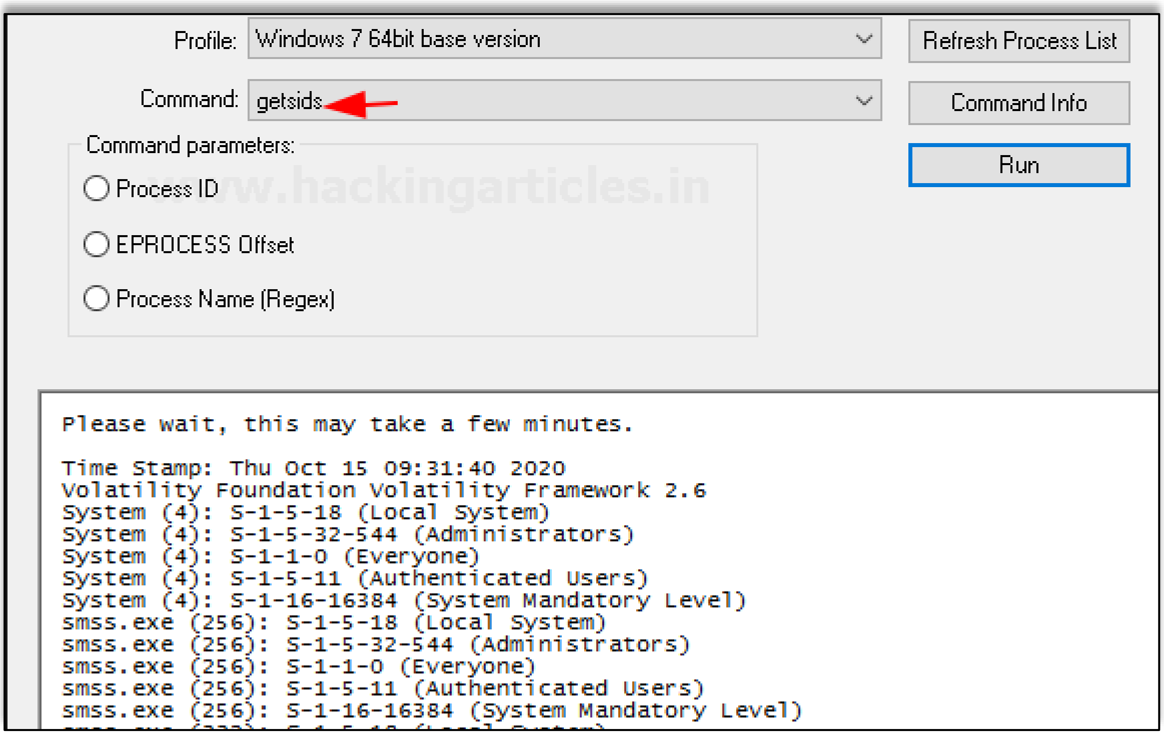


Profile: Windows 7 64bit base version
Command: timers
Command parameters:
☐ nt!KiTimer... Address

```
Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:28:42 2020
Volatility Foundation Volatility Framework 2.6
Offset(V)          DueTime                   Period(ms) Signaled   Routine            Module
------------------ ------------------------- ---------- ---------- ------------------ ------
0xfffff80002c31b40 0x00003d36:0x9aa8bd43              0 -          0xfffff80002b49340 ntoskrnl.exe
0xfffffa801914eb90 0x00000000:0xbbb133bd              0 Yes        0xfffff88001453750 afd.sys
0xfffffa8019844000 0x00000000:0xbbd5f000             16 Yes        0xfffff88003e4d9c0 vm3dmp.sys
0xfffff88003d04480 0x00000002:0x1eff9180              0 -          0xfffff88003d00c3c dfsc.sys
0xfffff80002c92fa0 0x00000000:0xb9f76c00              0 -          0xfffff80002adb618 ntoskrnl.exe
0xfffff880015185b0 0x00000000:0xbebc2000          10000 Yes        0xfffff880014b5360 cng.sys
0xfffff880015185b0 0x00000000:0xbebc2000          10000 Yes        0xfffff880014b5360 cng.sys
0xfffff880012b40e8 0x00000000:0xba1d91a0              0 -          0xfffff8800127be0 Ntfs.sys
0xfffffa801a38c278 0x00000000:0xbc823f55              0 -          0xfffff88001718160 ndis.sys
0xfffffa801aa2a370 0x00000000:0xe765bde0         120000 Yes        0xfffff80002b91390 ntoskrnl.exe
0xfffffa80197eb278 0x00000000:0xbc930836              0 -          0xfffff88001718160 ndis.sys
0xfffffa80197ed278 0x00000000:0xbc930836              0 -          0xfffff88001718160 ndis.sys
```

5. Getsids

This command can be used to view the Security Identifiers that are associated with a particular process. With the help of this command, you can identify if any malicious process has taken any privilege escalation.

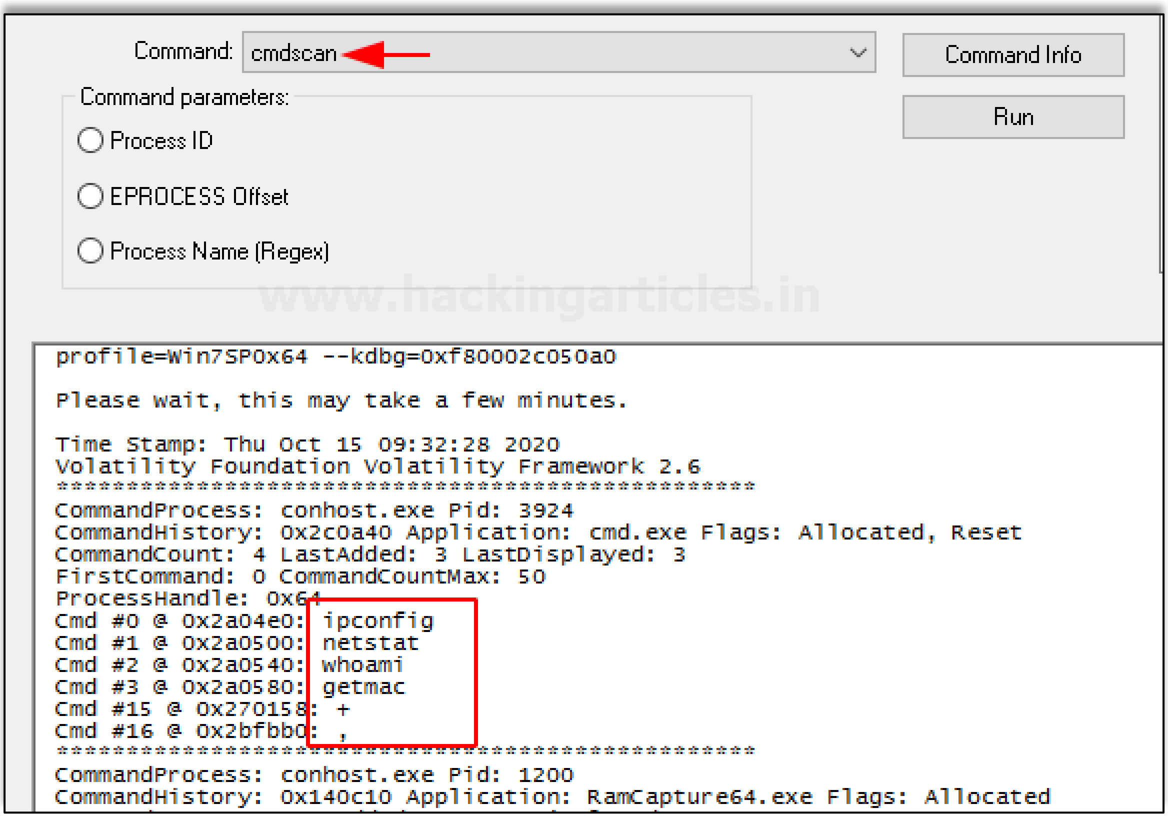


Profile: Windows 7 64bit base version
Command: getsids
Command parameters:
○ Process ID
○ EPROCESS Offset
○ Process Name (Regex)

```
Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:31:40 2020
Volatility Foundation Volatility Framework 2.6
System (4): S-1-5-18 (Local System)
System (4): S-1-5-32-544 (Administrators)
System (4): S-1-1-0 (Everyone)
System (4): S-1-5-11 (Authenticated Users)
System (4): S-1-16-16384 (System Mandatory Level)
smss.exe (256): S-1-5-18 (Local System)
smss.exe (256): S-1-5-32-544 (Administrators)
smss.exe (256): S-1-1-0 (Everyone)
smss.exe (256): S-1-5-11 (Authenticated Users)
smss.exe (256): S-1-16-16384 (System Mandatory Level)
```

6. Cmdscan

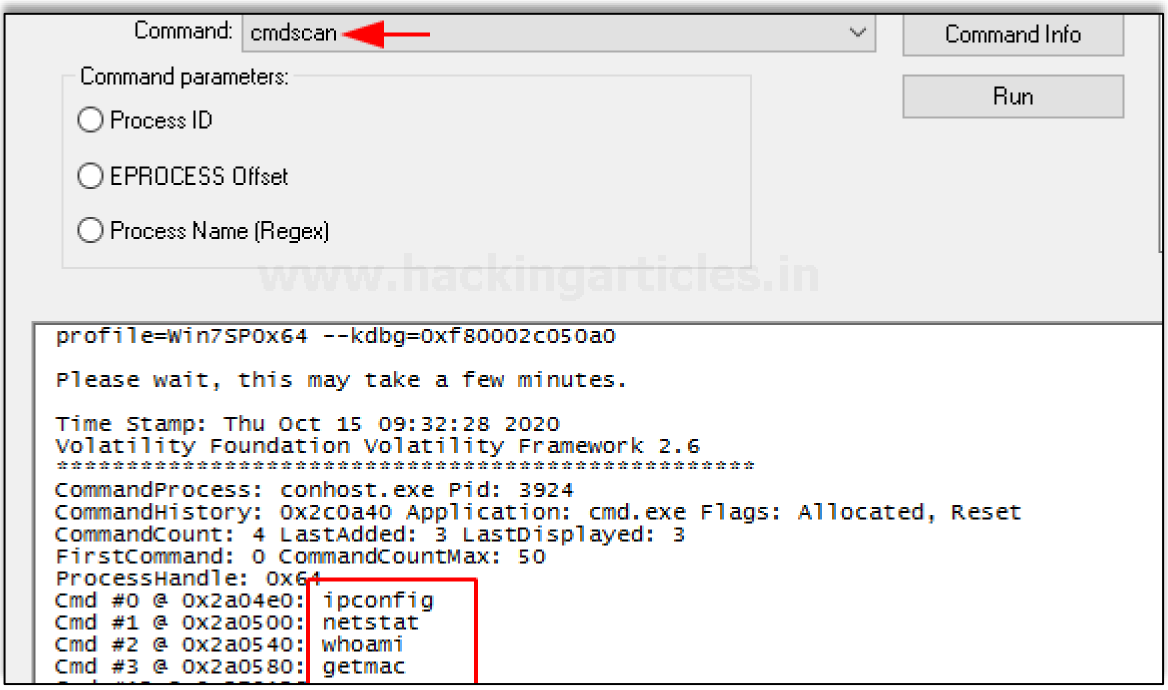
This plugin helps in searching the memory dump for the command the user must have used the cmd.exe application. This command is highly used if the attacker's command activity is to be traced.



```
profile=Win7SP0x64 --kdbg=0xf80002c050a0

Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:32:28 2020
Volatility Foundation Volatility Framework 2.6
**************************************************
CommandProcess: conhost.exe Pid: 3924
CommandHistory: 0x2c0a40 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 4 LastAdded: 3 LastDisplayed: 3
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x64
Cmd #0 @ 0x2a04e0: ipconfig
Cmd #1 @ 0x2a0500: netstat
Cmd #2 @ 0x2a0540: whoami
Cmd #3 @ 0x2a0580: getmac
Cmd #15 @ 0x270158: +
Cmd #16 @ 0x2bfbb0: ,
**************************************************
CommandProcess: conhost.exe Pid: 1200
CommandHistory: 0x140c10 Application: RamCapture64.exe Flags: Allocated
```

7. Consoles

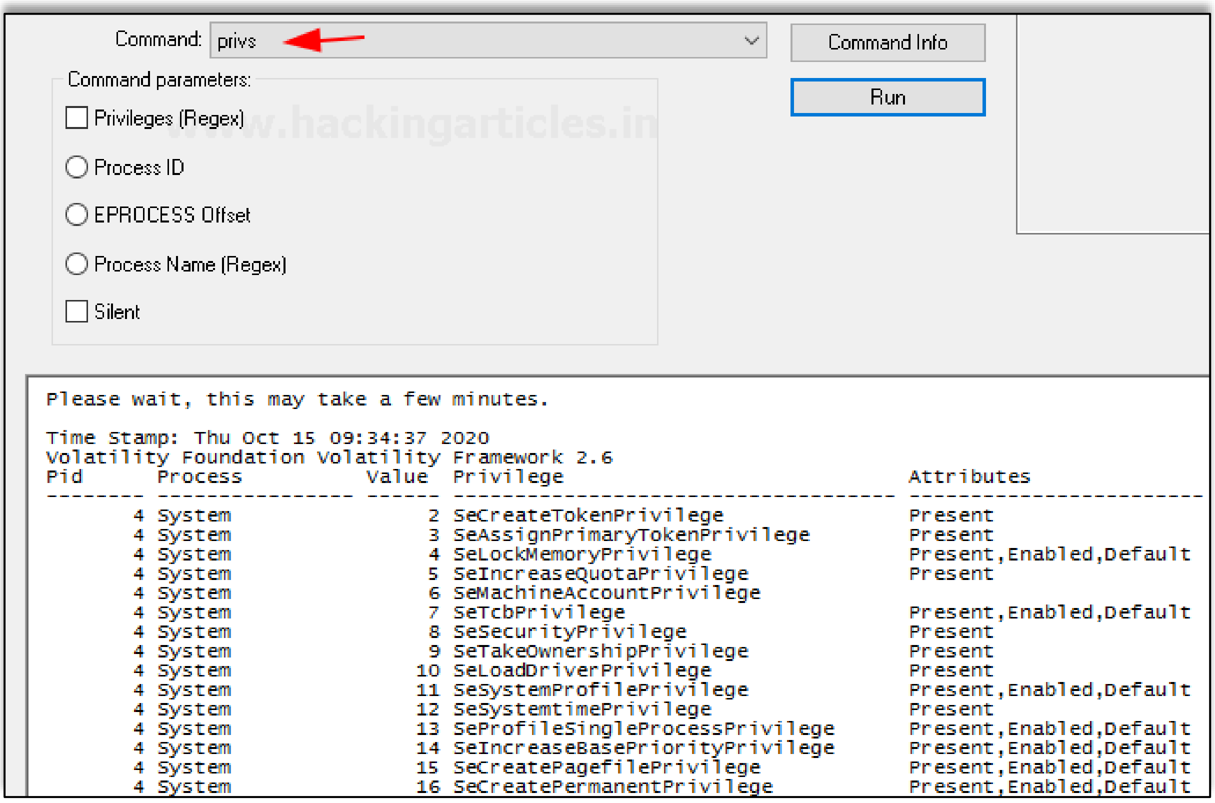
This command is similar to cmdscan and helps to find if the attacker had typed anything in cmd or had executed anything via the backdoor.



```
profile=Win7SP0x64 --kdbg=0xf80002c050a0

Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:32:28 2020
Volatility Foundation Volatility Framework 2.6
**************************************************
CommandProcess: conhost.exe Pid: 3924
CommandHistory: 0x2c0a40 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 4 LastAdded: 3 LastDisplayed: 3
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x64
Cmd #0 @ 0x2a04e0: ipconfig
Cmd #1 @ 0x2a0500: netstat
Cmd #2 @ 0x2a0540: whoami
Cmd #3 @ 0x2a0580: getmac
```

8. Privs

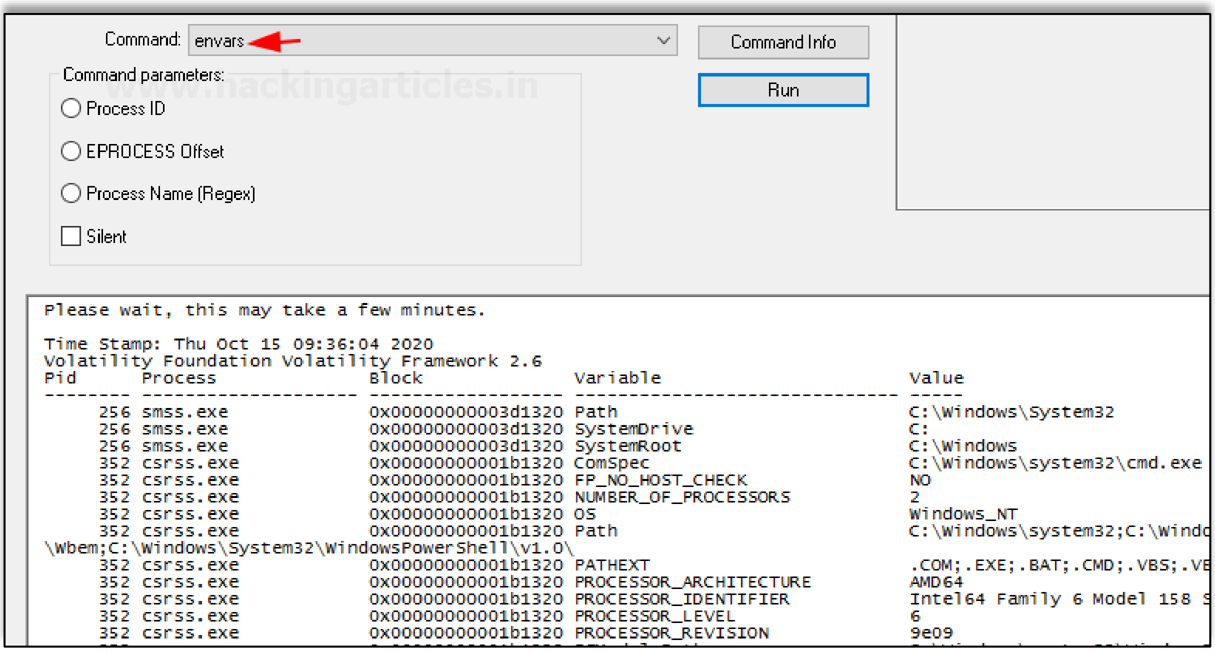
This command displays the privileges assigned to the processes that are enabled or not enabled by default.



```
Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:34:37 2020
Volatility Foundation Volatility Framework 2.6
Pid      Process          Value  Privilege                            Attributes
-------- ---------------- ------ ------------------------------------ ------------------------
       4 System                2 SeCreateTokenPrivilege               Present
       4 System                3 SeAssignPrimaryTokenPrivilege        Present
       4 System                4 SeLockMemoryPrivilege                Present,Enabled,Default
       4 System                5 SeIncreaseQuotaPrivilege             Present
       4 System                6 SeMachineAccountPrivilege
       4 System                7 SeTcbPrivilege                       Present,Enabled,Default
       4 System                8 SeSecurityPrivilege                  Present
       4 System                9 SeTakeOwnershipPrivilege             Present
       4 System               10 SeLoadDriverPrivilege                Present
       4 System               11 SeSystemProfilePrivilege             Present,Enabled,Default
       4 System               12 SeSystemtimePrivilege                Present
       4 System               13 SeProfileSingleProcessPrivilege      Present,Enabled,Default
       4 System               14 SeIncreaseBasePriorityPrivilege      Present,Enabled,Default
       4 System               15 SeCreatePagefilePrivilege            Present,Enabled,Default
       4 System               16 SeCreatePermanentPrivilege           Present,Enabled,Default
```

9. Envars

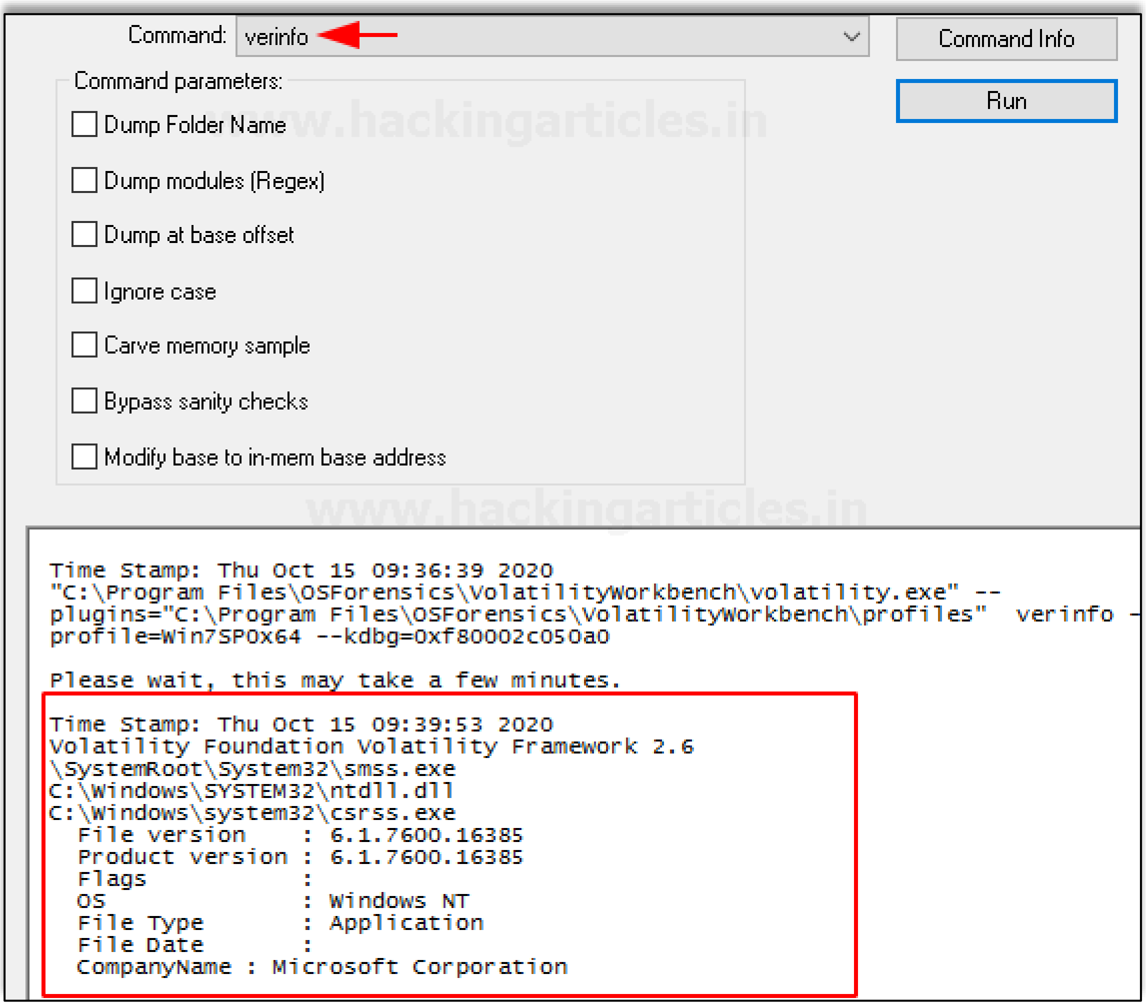
This command displays all the variables in the process, its environment along with its current directory.



```
Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:36:04 2020
Volatility Foundation Volatility Framework 2.6
Pid      Process              Block              Variable                       Value
-------- -------------------- ------------------ ------------------------------ -----
     256 smss.exe             0x00000000003d1320 Path                           C:\Windows\System32
     256 smss.exe             0x00000000003d1320 SystemDrive                    C:
     256 smss.exe             0x00000000003d1320 SystemRoot                     C:\Windows
     352 csrss.exe            0x00000000001b1320 ComSpec                        C:\Windows\system32\cmd.exe
     352 csrss.exe            0x00000000001b1320 FP_NO_HOST_CHECK               NO
     352 csrss.exe            0x00000000001b1320 NUMBER_OF_PROCESSORS           2
     352 csrss.exe            0x00000000001b1320 OS                             Windows_NT
     352 csrss.exe            0x00000000001b1320 Path                           C:\Windows\system32;C:\Windo
\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\
     352 csrss.exe            0x00000000001b1320 PATHEXT                        .COM;.EXE;.BAT;.CMD;.VBS;.VE
     352 csrss.exe            0x00000000001b1320 PROCESSOR_ARCHITECTURE         AMD64
     352 csrss.exe            0x00000000001b1320 PROCESSOR_IDENTIFIER           Intel64 Family 6 Model 158 S
     352 csrss.exe            0x00000000001b1320 PROCESSOR_LEVEL                6
     352 csrss.exe            0x00000000001b1320 PROCESSOR_REVISION             9e09
```

10. Verinfo

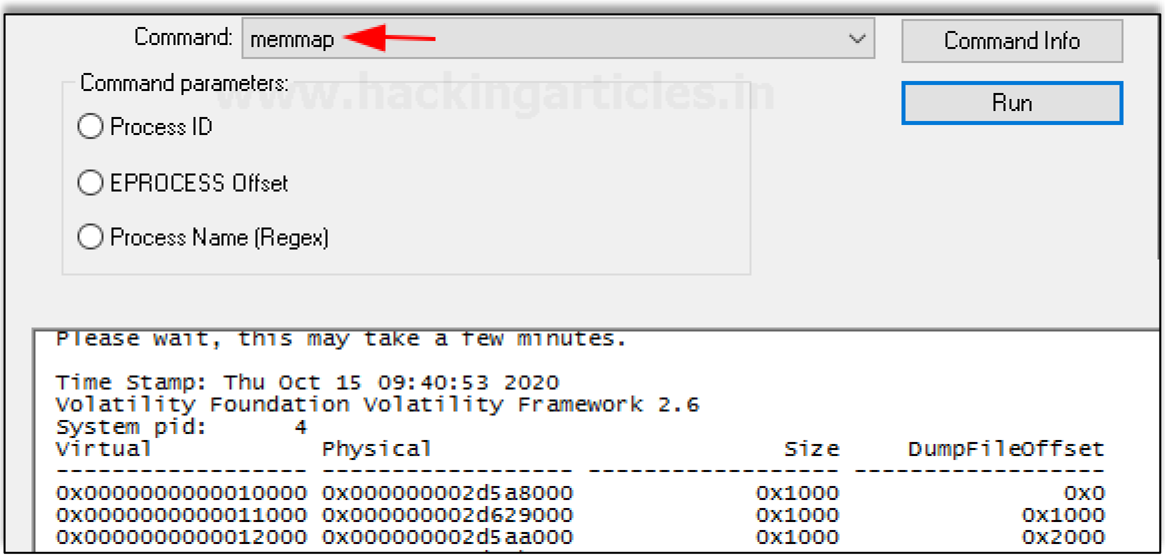
This command displays the version information that is present in the PE files. It helps identify any binaries and also correlates with other files.



```
Time Stamp: Thu Oct 15 09:36:39 2020
"C:\Program Files\OSForensics\VolatilityWorkbench\volatility.exe" --
plugins="C:\Program Files\OSForensics\VolatilityWorkbench\profiles"  verinfo -
profile=Win7SP0x64 --kdbg=0xf80002c050a0

Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:39:53 2020
Volatility Foundation Volatility Framework 2.6
\SystemRoot\System32\smss.exe
C:\Windows\SYSTEM32\ntdll.dll
C:\Windows\system32\csrss.exe
  File version    : 6.1.7600.16385
  Product version : 6.1.7600.16385
  Flags           :
  OS              : Windows NT
  File Type       : Application
  File Date       :
  CompanyName : Microsoft Corporation
```

11. Memmap

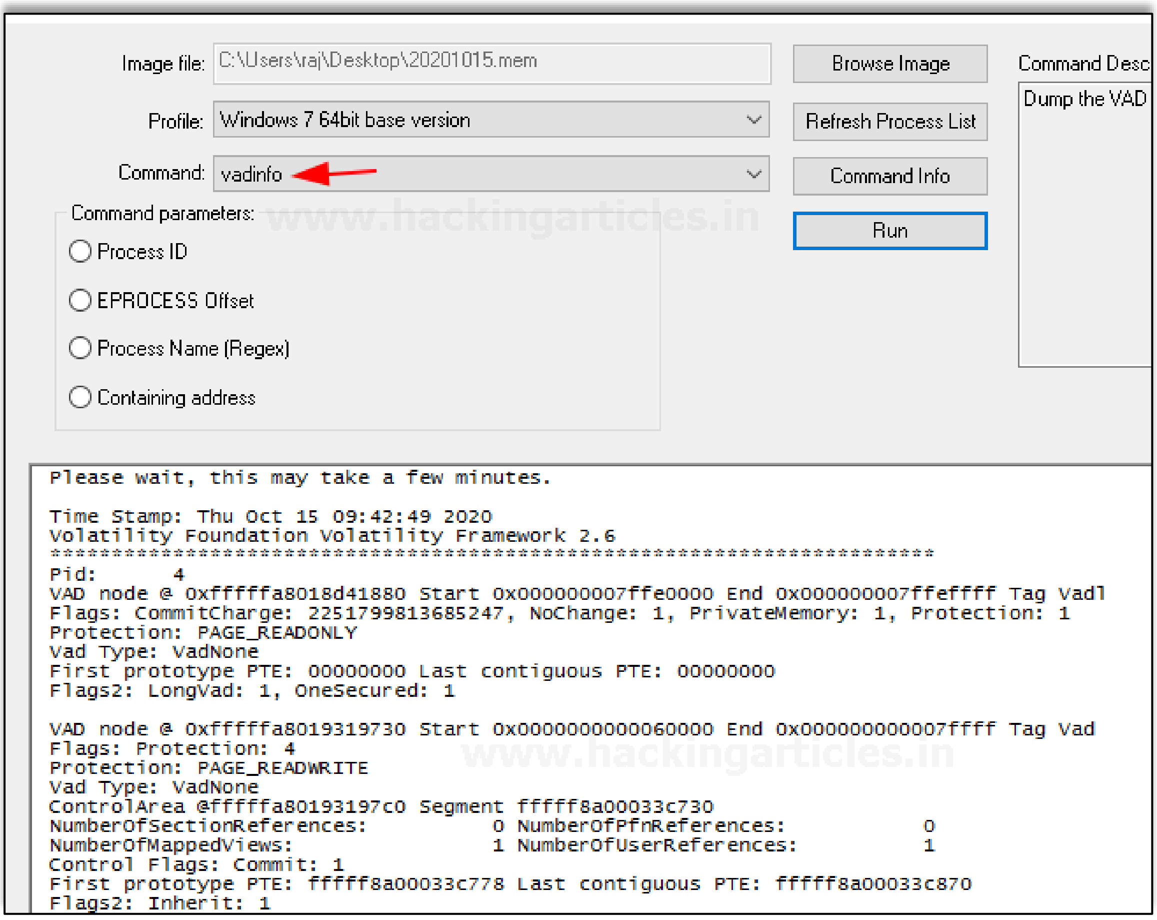
This command shows the exact pages that are present on the page of a specific process. It also shows the virtual address of the page and the size of its page.



```
Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:40:53 2020
Volatility Foundation Volatility Framework 2.6
System pid:       4
Virtual             Physical                          Size     DumpFileOffset
------------------  ------------------  ------------------  ------------------
0x0000000000010000  0x000000002d5a8000              0x1000                 0x0
0x0000000000011000  0x000000002d629000              0x1000              0x1000
0x0000000000012000  0x000000002d5aa000              0x1000              0x2000
```

12. Vadinfo

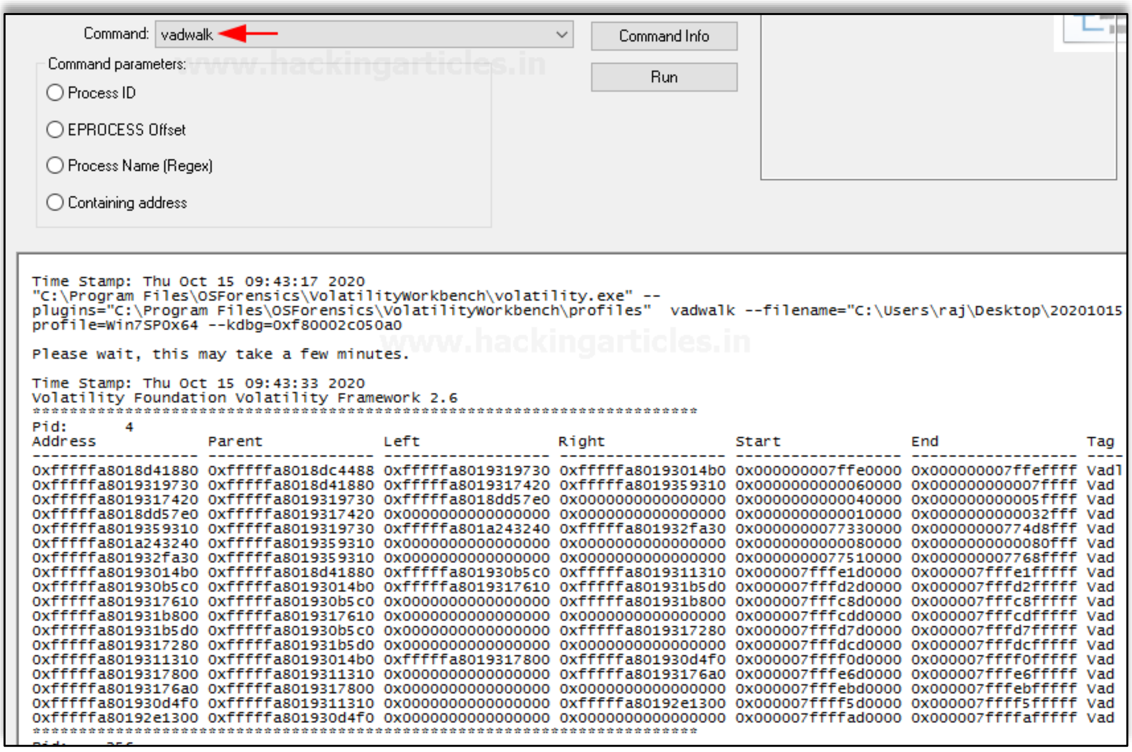
This command usually displays information about a particular process's VAD nodes. It displays the VAD Flags control flags, VAD tags.



```
Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:42:49 2020
Volatility Foundation Volatility Framework 2.6
************************************************************************
Pid:      4
VAD node @ 0xfffffa8018d41880 Start 0x000000007ffe0000 End 0x000000007ffeffff Tag Vadl
Flags: CommitCharge: 2251799813685247, NoChange: 1, PrivateMemory: 1, Protection: 1
Protection: PAGE_READONLY
Vad Type: VadNone
First prototype PTE: 00000000 Last contiguous PTE: 00000000
Flags2: LongVad: 1, OneSecured: 1

VAD node @ 0xfffffa8019319730 Start 0x0000000000060000 End 0x000000000007ffff Tag Vad
Flags: Protection: 4
Protection: PAGE_READWRITE
Vad Type: VadNone
ControlArea @fffffa80193197c0 Segment fffff8a00033c730
NumberOfSectionReferences:          0 NumberOfPfnReferences:           0
NumberOfMappedViews:                1 NumberOfUserReferences:          1
Control Flags: Commit: 1
First prototype PTE: fffff8a00033c778 Last contiguous PTE: fffff8a00033c870
Flags2: Inherit: 1
```

13. Vadwalk

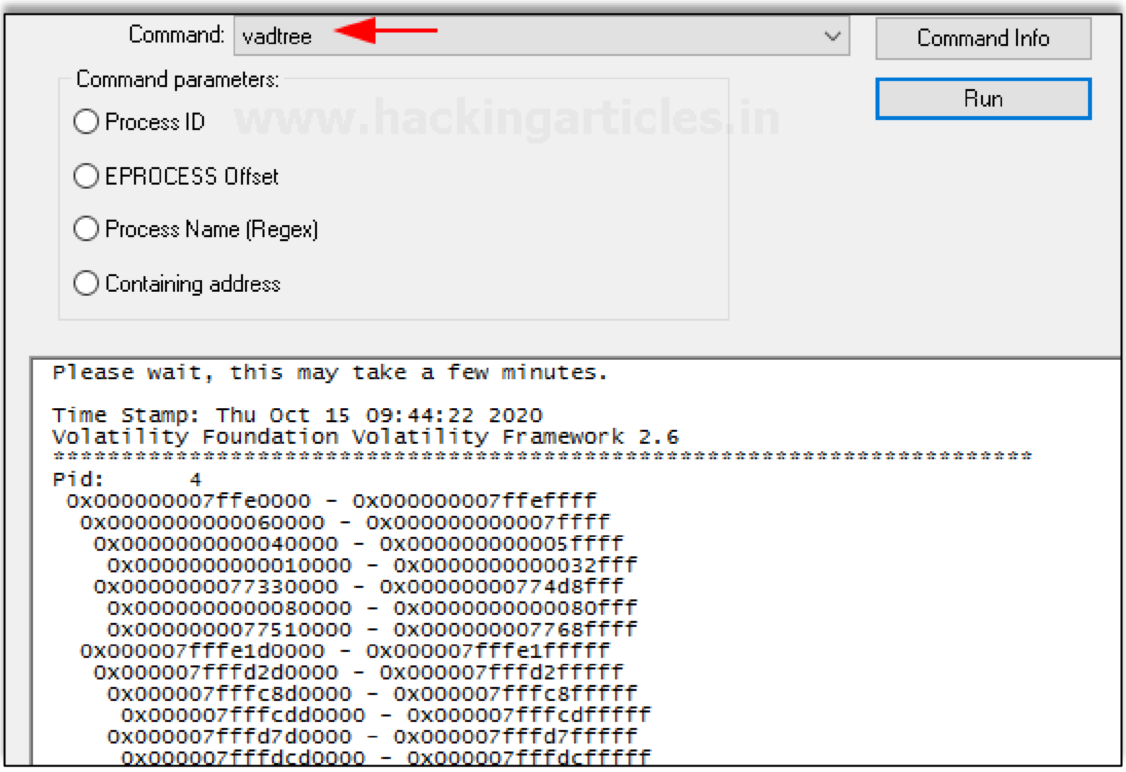
It is a command that is used to display all the VAD nodes in a tabular form.



```
Time Stamp: Thu Oct 15 09:43:17 2020
"C:\Program Files\OSForensics\VolatilityWorkbench\volatility.exe" --
plugins="C:\Program Files\OSForensics\VolatilityWorkbench\profiles"  vadwalk --filename="C:\Users\raj\Desktop\20201015
profile=Win7SP0x64 --kdbg=0xf80002c050a0

Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:43:33 2020
Volatility Foundation Volatility Framework 2.6
************************************************************************
Pid:      4
Address            Parent             Left               Right              Start              End                Tag
------------------ ------------------ ------------------ ------------------ ------------------ ------------------ ----
0xfffffa8018d41880 0xfffffa8018dc4488 0xfffffa8019319730 0xfffffa80193014b0 0x000000007ffe0000 0x000000007ffeffff Vadl
0xfffffa8019319730 0xfffffa8018d41880 0xfffffa8019317420 0xfffffa8019359310 0x0000000000060000 0x000000000007ffff Vad
0xfffffa8019317420 0xfffffa8019319730 0xfffffa8018dd57e0 0x0000000000000000 0x0000000000040000 0x000000000005ffff Vad
0xfffffa8018dd57e0 0xfffffa8019317420 0x0000000000000000 0x0000000000000000 0x0000000000010000 0x000000000032ffff Vad
0xfffffa8019359310 0xfffffa8019319730 0xfffffa801a243240 0xfffffa801932fa30 0x0000000077330000 0x00000000774d8fff Vad
0xfffffa801a243240 0xfffffa8019359310 0x0000000000000000 0x0000000000000000 0x0000000000080000 0x0000000000080fff Vad
0xfffffa801932fa30 0xfffffa8019359310 0x0000000000000000 0x0000000000000000 0x0000000077510000 0x000000007768ffff Vad
0xfffffa80193014b0 0xfffffa8018d41880 0xfffffa801930b5c0 0xfffffa8019311310 0x000007fffe1d0000 0x000007fffe1fffff Vad
0xfffffa801930b5c0 0xfffffa80193014b0 0xfffffa8019317610 0xfffffa801931b5d0 0x000007fffd2d0000 0x000007fffd2fffff Vad
0xfffffa8019317610 0xfffffa801930b5c0 0x0000000000000000 0xfffffa801931b800 0x000007fffc8d0000 0x000007fffc8fffff Vad
0xfffffa801931b800 0xfffffa8019317610 0x0000000000000000 0x0000000000000000 0x000007fffcdd0000 0x000007fffcdfffff Vad
0xfffffa801931b5d0 0xfffffa801930b5c0 0x0000000000000000 0xfffffa8019317280 0x000007fffd7d0000 0x000007fffd7fffff Vad
0xfffffa8019317280 0xfffffa801931b5d0 0x0000000000000000 0x0000000000000000 0x000007fffdcd0000 0x000007fffdcfffff Vad
0xfffffa8019311310 0xfffffa80193014b0 0xfffffa8019317800 0xfffffa801930d4f0 0x000007ffff0d0000 0x000007ffff0fffff Vad
0xfffffa8019317800 0xfffffa8019311310 0x0000000000000000 0xfffffa80193176a0 0x000007fffe6d0000 0x000007fffe6fffff Vad
0xfffffa80193176a0 0xfffffa8019317800 0x0000000000000000 0x0000000000000000 0x000007fffebd0000 0x000007fffebfffff Vad
0xfffffa801930d4f0 0xfffffa8019311310 0x0000000000000000 0xfffffa80192e1300 0x000007ffff5d0000 0x000007ffff5fffff Vad
0xfffffa80192e1300 0xfffffa801930d4f0 0x0000000000000000 0x0000000000000000 0x000007ffffad0000 0x000007ffffafffff Vad
************************************************************************
```

14. Vadtree

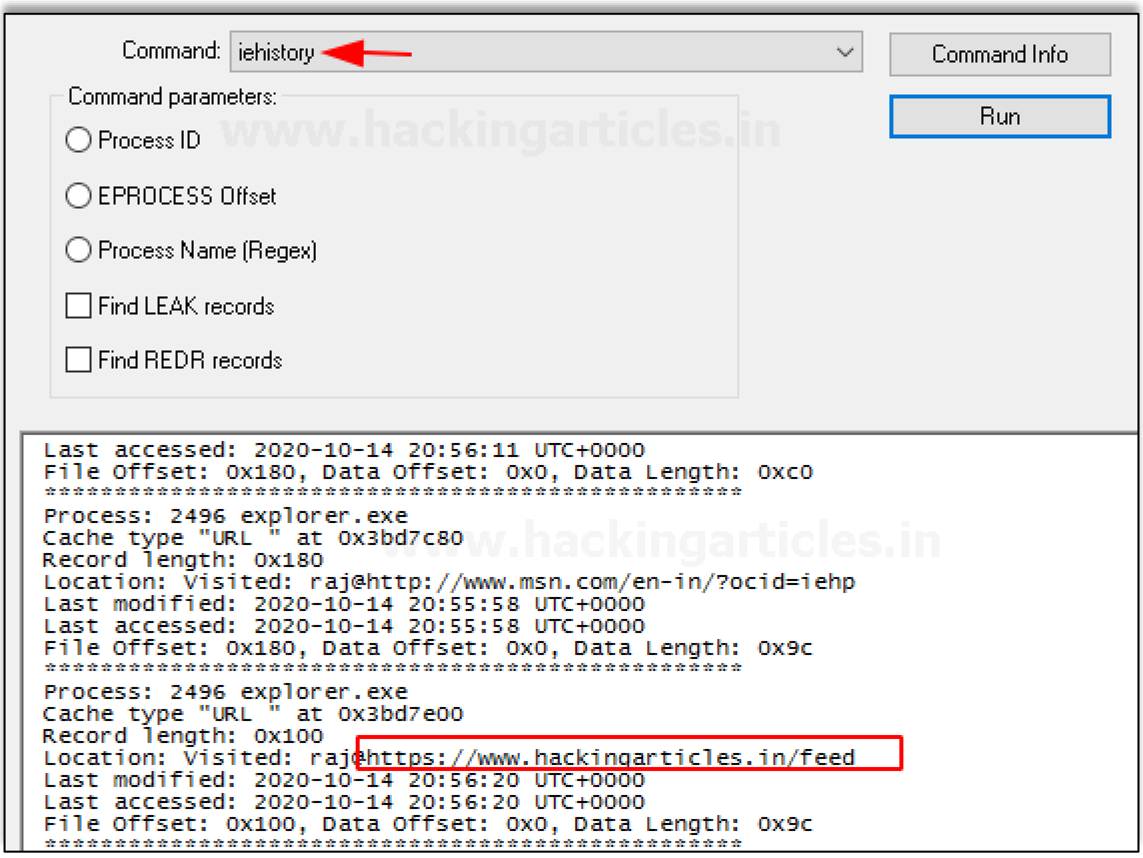
This process displays the VAD nodes in a tree form.



```
Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:44:22 2020
Volatility Foundation Volatility Framework 2.6
************************************************************************
Pid:      4
 0x000000007ffe0000 - 0x000000007ffeffff
  0x0000000000060000 - 0x000000000007ffff
   0x0000000000040000 - 0x000000000005ffff
    0x0000000000010000 - 0x000000000032ffff
   0x0000000077330000 - 0x00000000774d8fff
    0x0000000000080000 - 0x0000000000080fff
    0x0000000077510000 - 0x000000007768ffff
  0x000007fffe1d0000 - 0x000007fffe1fffff
   0x000007fffd2d0000 - 0x000007fffd2fffff
    0x000007fffc8d0000 - 0x000007fffc8fffff
     0x000007fffcdd0000 - 0x000007fffcdfffff
    0x000007fffd7d0000 - 0x000007fffd7fffff
     0x000007fffdcd0000 - 0x000007fffdcfffff
```

15. iehistory

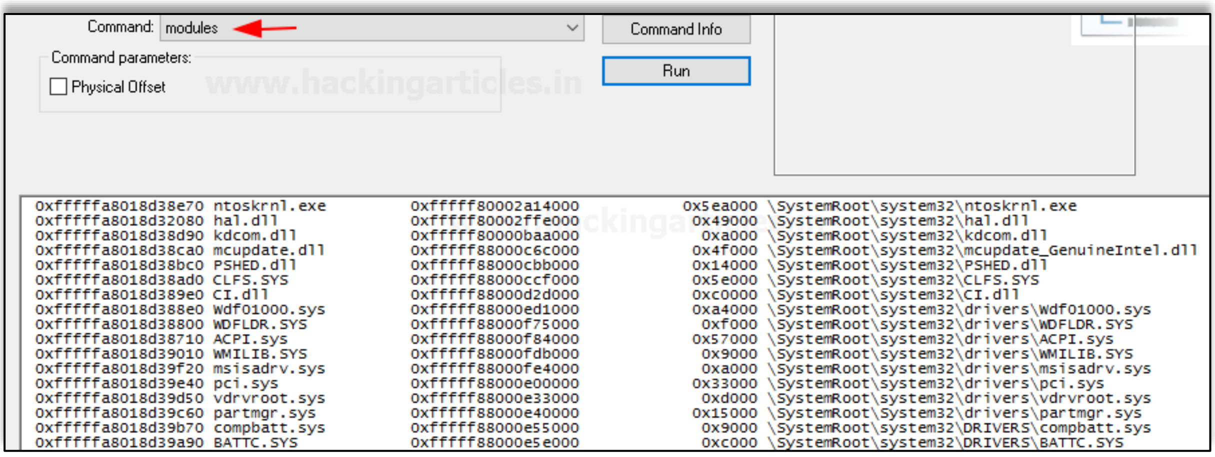
This Plugin helps in recovering the fragments of the Internet explore history index.dat named cache files. It displays FTP and HTTP links that were accessed, links that were redirected, any deleted entries.



```
Last accessed: 2020-10-14 20:56:11 UTC+0000
File Offset: 0x180, Data Offset: 0x0, Data Length: 0xc0
**************************************************************
Process: 2496 explorer.exe
Cache type "URL " at 0x3bd7c80
Record length: 0x180
Location: Visited: raj@http://www.msn.com/en-in/?ocid=iehp
Last modified: 2020-10-14 20:55:58 UTC+0000
Last accessed: 2020-10-14 20:55:58 UTC+0000
File Offset: 0x180, Data Offset: 0x0, Data Length: 0x9c
**************************************************************
Process: 2496 explorer.exe
Cache type "URL " at 0x3bd7e00
Record length: 0x100
Location: Visited: raj@https://www.hackingarticles.in/feed
Last modified: 2020-10-14 20:56:20 UTC+0000
Last accessed: 2020-10-14 20:56:20 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0x9c
**************************************************************
```

16. Modules

This command is used to list the kernel drivers that are present in the system.



```
0xfffffa8018d38e70 ntoskrnl.exe       0xfffff80002a14000   0x5ea000 \SystemRoot\system32\ntoskrnl.exe
0xfffffa8018d32080 hal.dll            0xfffff80002ffe000    0x49000 \SystemRoot\system32\hal.dll
0xfffffa8018d38d90 kdcom.dll          0xfffff80000baa000     0xa000 \SystemRoot\system32\kdcom.dll
0xfffffa8018d38ca0 mcupdate.dll       0xfffff88000c6c000    0x4f000 \SystemRoot\system32\mcupdate_GenuineIntel.dll
0xfffffa8018d38bc0 PSHED.dll          0xfffff88000cbb000    0x14000 \SystemRoot\system32\PSHED.dll
0xfffffa8018d38ad0 CLFS.SYS           0xfffff88000ccf000    0x5e000 \SystemRoot\system32\CLFS.SYS
0xfffffa8018d389e0 CI.dll             0xfffff88000d2d000    0xc0000 \SystemRoot\system32\CI.dll
0xfffffa8018d388e0 Wdf01000.sys       0xfffff88000ed1000    0xa4000 \SystemRoot\system32\drivers\Wdf01000.sys
0xfffffa8018d38800 WDFLDR.SYS         0xfffff88000f75000     0xf000 \SystemRoot\system32\drivers\WDFLDR.SYS
0xfffffa8018d38710 ACPI.sys           0xfffff88000f84000    0x57000 \SystemRoot\system32\drivers\ACPI.sys
0xfffffa8018d39010 WMILIB.SYS         0xfffff88000fdb000     0x9000 \SystemRoot\system32\drivers\WMILIB.SYS
0xfffffa8018d39f20 msisadrv.sys       0xfffff88000fe4000     0xa000 \SystemRoot\system32\drivers\msisadrv.sys
0xfffffa8018d39e40 pci.sys            0xfffff88000e00000    0x33000 \SystemRoot\system32\drivers\pci.sys
0xfffffa8018d39d50 vdrvroot.sys       0xfffff88000e33000     0xd000 \SystemRoot\system32\drivers\vdrvroot.sys
0xfffffa8018d39c60 partmgr.sys        0xfffff88000e40000    0x15000 \SystemRoot\System32\drivers\partmgr.sys
0xfffffa8018d39b70 compbatt.sys       0xfffff88000e55000     0x9000 \SystemRoot\system32\DRIVERS\compbatt.sys
0xfffffa8018d39a90 BATTC.SYS          0xfffff88000e5e000     0xc000 \SystemRoot\system32\DRIVERS\BATTC.SYS
```

17. SSDT

This command is used to list the functions present in the original and GUI SSDTs. It displays the index, the name of the function, and the owner of the driver of each entry in the SSDT.

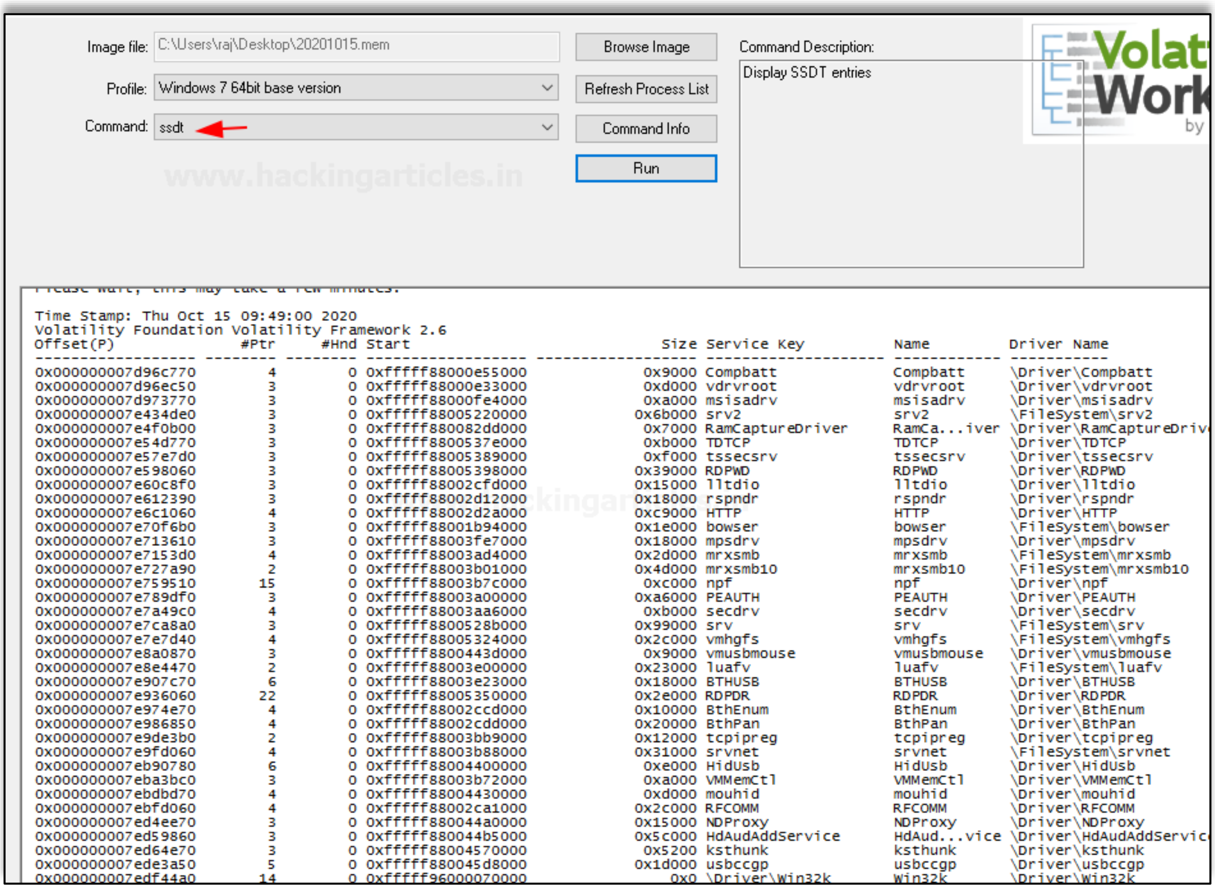
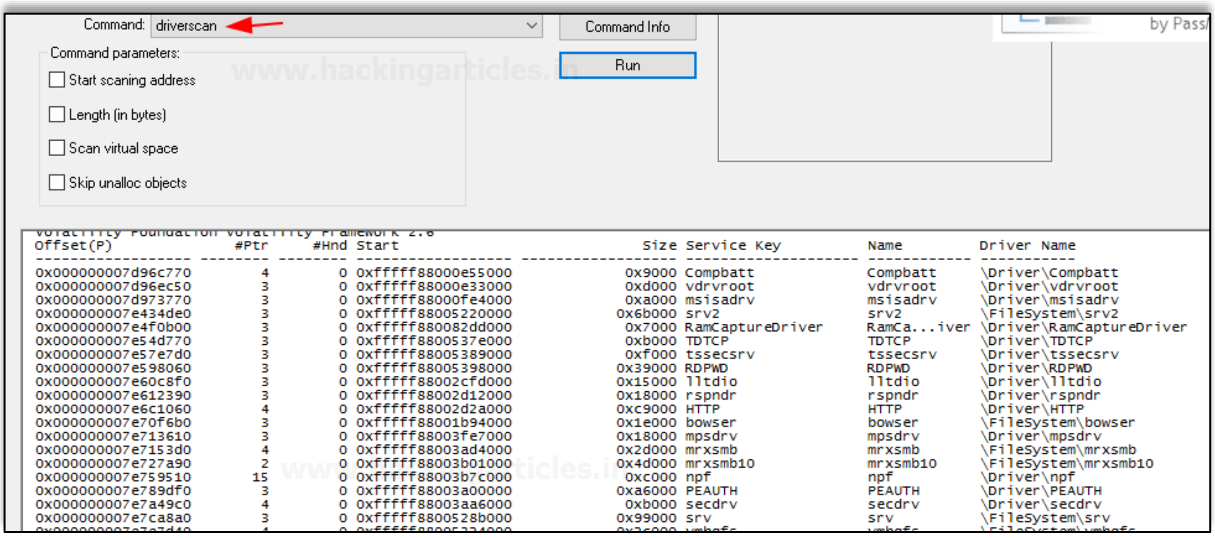


Image file: C:\Users\raj\Desktop\20201015.mem
Profile: Windows 7 64bit base version
Command: ssdt
Command Description:
Display SSDT entries

```
[illegible]

Time Stamp: Thu Oct 15 09:49:00 2020
Volatility Foundation Volatility Framework 2.6
Offset(P)              #Ptr     #Hnd Start                            Size Service Key           Name         Driver Name
------------------ -------- -------- ------------------ ------------------ -------------------- ------------ -----------
0x000000007d96c770        4        0 0xfffff88000e55000             0x9000 Compbatt             Compbatt     \Driver\Compbatt
0x000000007d96ec50        3        0 0xfffff88000e33000             0xd000 vdrvroot             vdrvroot     \Driver\vdrvroot
0x000000007d973770        3        0 0xfffff88000fe4000             0xa000 msisadrv             msisadrv     \Driver\msisadrv
0x000000007e434de0        3        0 0xfffff88005220000            0x6b000 srv2                 srv2         \FileSystem\srv2
0x000000007e4f0b00        3        0 0xfffff880082dd000             0x7000 RamCaptureDriver     RamCa...iver \Driver\RamCaptureDriv
0x000000007e54d770        3        0 0xfffff8800537e000             0xb000 TDTCP                TDTCP        \Driver\TDTCP
0x000000007e57e7d0        3        0 0xfffff88005389000             0xf000 tssecsrv             tssecsrv     \Driver\tssecsrv
0x000000007e598060        3        0 0xfffff88005398000            0x39000 RDPWD                RDPWD        \Driver\RDPWD
0x000000007e60c8f0        3        0 0xfffff88002cfd000            0x15000 lltdio               lltdio       \Driver\lltdio
0x000000007e612390        3        0 0xfffff88002d12000            0x18000 rspndr               rspndr       \Driver\rspndr
0x000000007e6c1060        4        0 0xfffff88002d2a000            0xc9000 HTTP                 HTTP         \Driver\HTTP
0x000000007e70f6b0        3        0 0xfffff88001b94000            0x1e000 bowser               bowser       \FileSystem\bowser
0x000000007e713610        3        0 0xfffff88003fe7000            0x18000 mpsdrv               mpsdrv       \Driver\mpsdrv
0x000000007e7153d0        4        0 0xfffff88003ad4000            0x2d000 mrxsmb               mrxsmb       \FileSystem\mrxsmb
0x000000007e727a90        2        0 0xfffff88003b01000            0x4d000 mrxsmb10             mrxsmb10     \FileSystem\mrxsmb10
0x000000007e759510       15        0 0xfffff88003b7c000             0xc000 npf                  npf          \Driver\npf
0x000000007e789df0        3        0 0xfffff88003a00000            0xa6000 PEAUTH               PEAUTH       \Driver\PEAUTH
0x000000007e7a49c0        4        0 0xfffff88003aa6000             0xb000 secdrv               secdrv       \Driver\secdrv
0x000000007e7ca8a0        3        0 0xfffff8800528b000            0x99000 srv                  srv          \FileSystem\srv
0x000000007e7e7d40        4        0 0xfffff88005324000            0x2c000 vmhgfs               vmhgfs       \FileSystem\vmhgfs
0x000000007e8a0870        3        0 0xfffff8800443d000             0x9000 vmusbmouse           vmusbmouse   \Driver\vmusbmouse
0x000000007e8e4470        2        0 0xfffff88003e00000            0x23000 luafv                luafv        \FileSystem\luafv
0x000000007e907c70        6        0 0xfffff88003e23000            0x18000 BTHUSB               BTHUSB       \Driver\BTHUSB
0x000000007e936060       22        0 0xfffff88005350000            0x2e000 RDPDR                RDPDR        \Driver\RDPDR
0x000000007e974e70        4        0 0xfffff88002ccd000            0x10000 BthEnum              BthEnum      \Driver\BthEnum
0x000000007e986850        4        0 0xfffff88002cdd000            0x20000 BthPan               BthPan       \Driver\BthPan
0x000000007e9de3b0        2        0 0xfffff88003bb9000            0x12000 tcpipreg             tcpipreg     \Driver\tcpipreg
0x000000007e9fd060        4        0 0xfffff88003b88000            0x31000 srvnet               srvnet       \FileSystem\srvnet
0x000000007eb90780        6        0 0xfffff88004400000             0xe000 HidUsb               HidUsb       \Driver\HidUsb
0x000000007eba3bc0        3        0 0xfffff88003b72000             0xa000 VMMemCtl             VMMemCtl     \Driver\VMMemCtl
0x000000007ebdbd70        4        0 0xfffff88004430000             0xd000 mouhid               mouhid       \Driver\mouhid
0x000000007ebfd060        4        0 0xfffff88002ca1000            0x2c000 RFCOMM               RFCOMM       \Driver\RFCOMM
0x000000007ed4ee70        3        0 0xfffff880044a0000            0x15000 NDProxy              NDProxy      \Driver\NDProxy
0x000000007ed59860        3        0 0xfffff880044b5000            0x5c000 HdAudAddService      HdAud...vice \Driver\HdAudAddServic
0x000000007ed64e70        3        0 0xfffff88004570000             0x5200 ksthunk              ksthunk      \Driver\ksthunk
0x000000007ede3a50        5        0 0xfffff880045d8000            0x1d000 usbccgp              usbccgp      \Driver\usbccgp
0x000000007edf44a0       14        0 0xfffff96000070000                0x0 \Driver\Win32k       win32k       \Driver\Win32k
```

18. Driverscan

This command can be used to find the DRIVER_OBJECT present in the physical memory by making use of a pool tag scan.

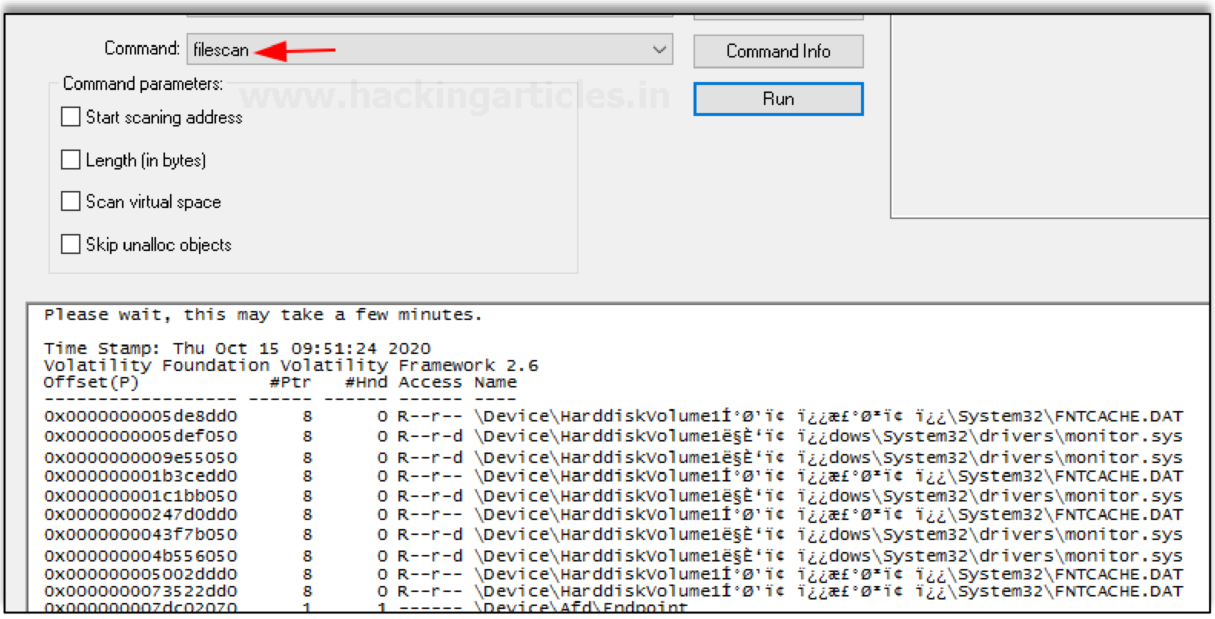


Command: driverscan
Command parameters:
☐ Start scaning address
☐ Length (in bytes)
☐ Scan virtual space
☐ Skip unalloc objects

```
Volatility Foundation Volatility Framework 2.6
Offset(P)              #Ptr     #Hnd Start                            Size Service Key          Name         Driver Name
------------------ -------- -------- ------------------ ------------------ -------------------- ------------ -----------
0x000000007d96c770        4        0 0xfffff88000e55000             0x9000 Compbatt             Compbatt     \Driver\Compbatt
0x000000007d96ec50        3        0 0xfffff88000e33000             0xd000 vdrvroot             vdrvroot     \Driver\vdrvroot
0x000000007d973770        3        0 0xfffff88000fe4000             0xa000 msisadrv             msisadrv     \Driver\msisadrv
0x000000007e434de0        3        0 0xfffff88005220000            0x6b000 srv2                 srv2         \FileSystem\srv2
0x000000007e4f0b00        3        0 0xfffff880082dd000             0x7000 RamCaptureDriver     RamCa...iver \Driver\RamCaptureDriver
0x000000007e54d770        3        0 0xfffff8800537e000             0xb000 TDTCP                TDTCP        \Driver\TDTCP
0x000000007e57e7d0        3        0 0xfffff88005389000             0xf000 tssecsrv             tssecsrv     \Driver\tssecsrv
0x000000007e598060        3        0 0xfffff88005398000            0x39000 RDPWD                RDPWD        \Driver\RDPWD
0x000000007e60c8f0        3        0 0xfffff88002cfd000            0x15000 lltdio               lltdio       \Driver\lltdio
0x000000007e612390        3        0 0xfffff88002d12000            0x18000 rspndr               rspndr       \Driver\rspndr
0x000000007e6c1060        4        0 0xfffff88002d2a000            0xc9000 HTTP                 HTTP         \Driver\HTTP
0x000000007e70f6b0        3        0 0xfffff88001b94000            0x1e000 bowser               bowser       \FileSystem\bowser
0x000000007e713610        3        0 0xfffff88003fe7000            0x18000 mpsdrv               mpsdrv       \Driver\mpsdrv
0x000000007e7153d0        4        0 0xfffff88003ad4000            0x2d000 mrxsmb               mrxsmb       \FileSystem\mrxsmb
0x000000007e727a90        2        0 0xfffff88003b01000            0x4d000 mrxsmb10             mrxsmb10     \FileSystem\mrxsmb10
0x000000007e759510       15        0 0xfffff88003b7c000             0xc000 npf                  npf          \Driver\npf
0x000000007e789df0        3        0 0xfffff88003a00000            0xa6000 PEAUTH               PEAUTH       \Driver\PEAUTH
0x000000007e7a49c0        4        0 0xfffff88003aa6000             0xb000 secdrv               secdrv       \Driver\secdrv
0x000000007e7ca8a0        3        0 0xfffff8800528b000            0x99000 srv                  srv          \FileSystem\srv
```

19. File Scan

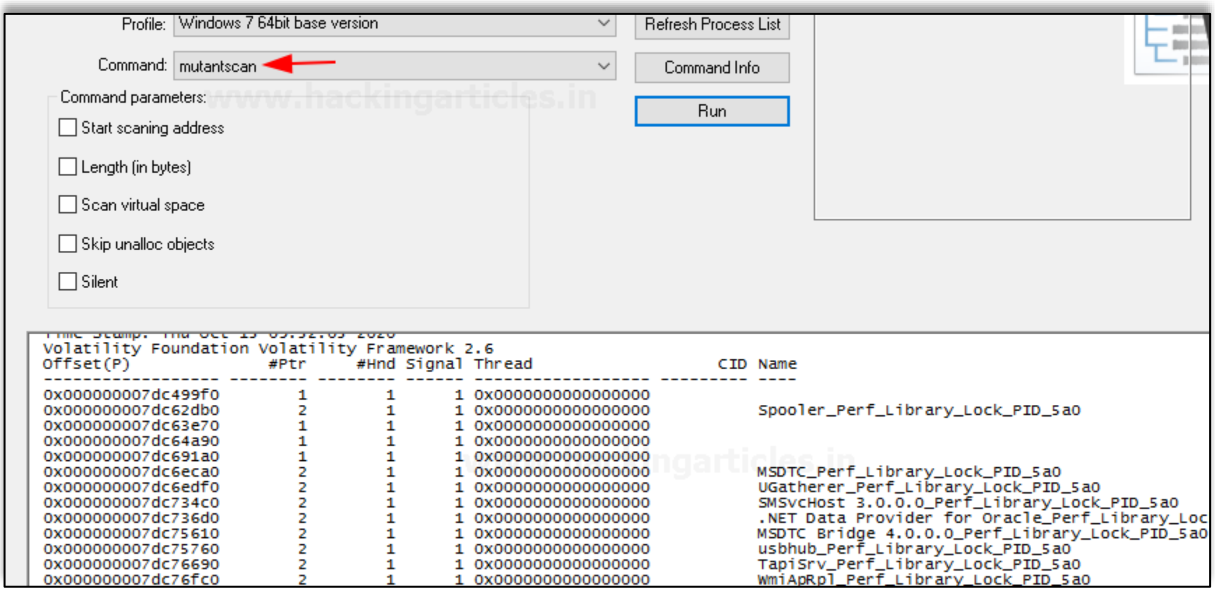
This command can be used to find File_object that is present in the physical memory by making use of a pool tag scan. This command will help in finding open files in the system dump even if they are hidden with the help of rootkit.



```
Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:51:24 2020
Volatility Foundation Volatility Framework 2.6
Offset(P)            #Ptr   #Hnd Access Name
------------------ ------ ------ ------ ----
0x000000005de8dd0       8      0 R--r-- \Device\HarddiskVolume1Î˚Ø'ï¢ ï¿¿æ£˚Ø*ï¢ ï¿¿\System32\FNTCACHE.DAT
0x000000005def050       8      0 R--r-d \Device\HarddiskVolume1ë§È'ï¢ ï¿¿dows\System32\drivers\monitor.sys
0x000000009e55050       8      0 R--r-d \Device\HarddiskVolume1ë§È'ï¢ ï¿¿dows\System32\drivers\monitor.sys
0x00000001b3cedd0       8      0 R--r-- \Device\HarddiskVolume1Î˚Ø'ï¢ ï¿¿æ£˚Ø*ï¢ ï¿¿\System32\FNTCACHE.DAT
0x00000001c1bb050       8      0 R--r-d \Device\HarddiskVolume1ë§È'ï¢ ï¿¿dows\System32\drivers\monitor.sys
0x0000000247d0dd0       8      0 R--r-- \Device\HarddiskVolume1Î˚Ø'ï¢ ï¿¿æ£˚Ø*ï¢ ï¿¿\System32\FNTCACHE.DAT
0x0000000043f7b050      8      0 R--r-d \Device\HarddiskVolume1ë§È'ï¢ ï¿¿dows\System32\drivers\monitor.sys
0x000000004b556050      8      0 R--r-d \Device\HarddiskVolume1ë§È'ï¢ ï¿¿dows\System32\drivers\monitor.sys
0x000000005002ddd0      8      0 R--r-- \Device\HarddiskVolume1Î˚Ø'ï¢ ï¿¿æ£˚Ø*ï¢ ï¿¿\System32\FNTCACHE.DAT
0x0000000073522dd0      8      0 R--r-- \Device\HarddiskVolume1Î˚Ø'ï¢ ï¿¿æ£˚Ø*ï¢ ï¿¿\System32\FNTCACHE.DAT
0x000000007dc02070      1      1 ------ \Device\Afd\Endpoint
```

20. Mutant scan

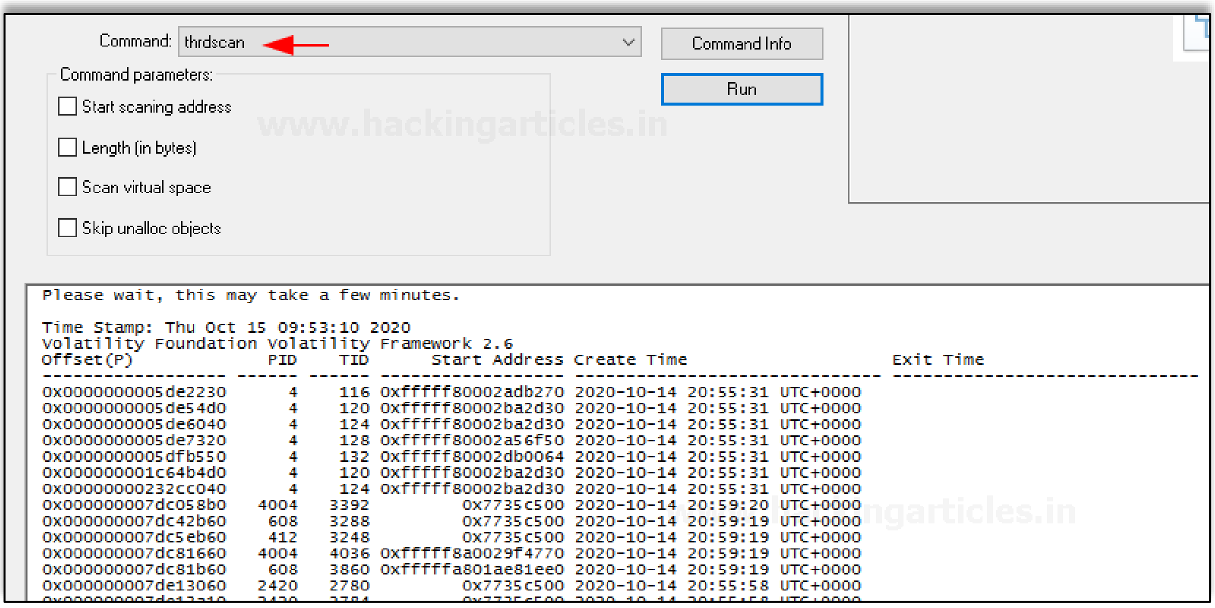
This command is used to scan the physical memory of mutant objects by making use of pool tag scanning.



```
Volatility Foundation Volatility Framework 2.6
Offset(P)              #Ptr     #Hnd Signal Thread                   CID Name
------------------ -------- -------- ------ ------------------ --------- ----
0x000000007dc499f0        1        1      1 0x0000000000000000
0x000000007dc62db0        2        1      1 0x0000000000000000           Spooler_Perf_Library_Lock_PID_5a0
0x000000007dc63e70        1        1      1 0x0000000000000000
0x000000007dc64a90        1        1      1 0x0000000000000000
0x000000007dc691a0        1        1      1 0x0000000000000000
0x000000007dc6eca0        2        1      1 0x0000000000000000           MSDTC_Perf_Library_Lock_PID_5a0
0x000000007dc6edf0        2        1      1 0x0000000000000000           UGatherer_Perf_Library_Lock_PID_5a0
0x000000007dc734c0        2        1      1 0x0000000000000000           SMSvcHost 3.0.0.0_Perf_Library_Lock_PID_5a0
0x000000007dc736d0        2        1      1 0x0000000000000000           .NET Data Provider for Oracle_Perf_Library_Loc
0x000000007dc75610        2        1      1 0x0000000000000000           MSDTC Bridge 4.0.0.0_Perf_Library_Lock_PID_5a0
0x000000007dc75760        2        1      1 0x0000000000000000           usbhub_Perf_Library_Lock_PID_5a0
0x000000007dc76690        2        1      1 0x0000000000000000           TapiSrv_Perf_Library_Lock_PID_5a0
0x000000007dc76fc0        2        1      1 0x0000000000000000           WmiApRpl_Perf_Library_Lock_PID_5a0
```

21. Thrdscan

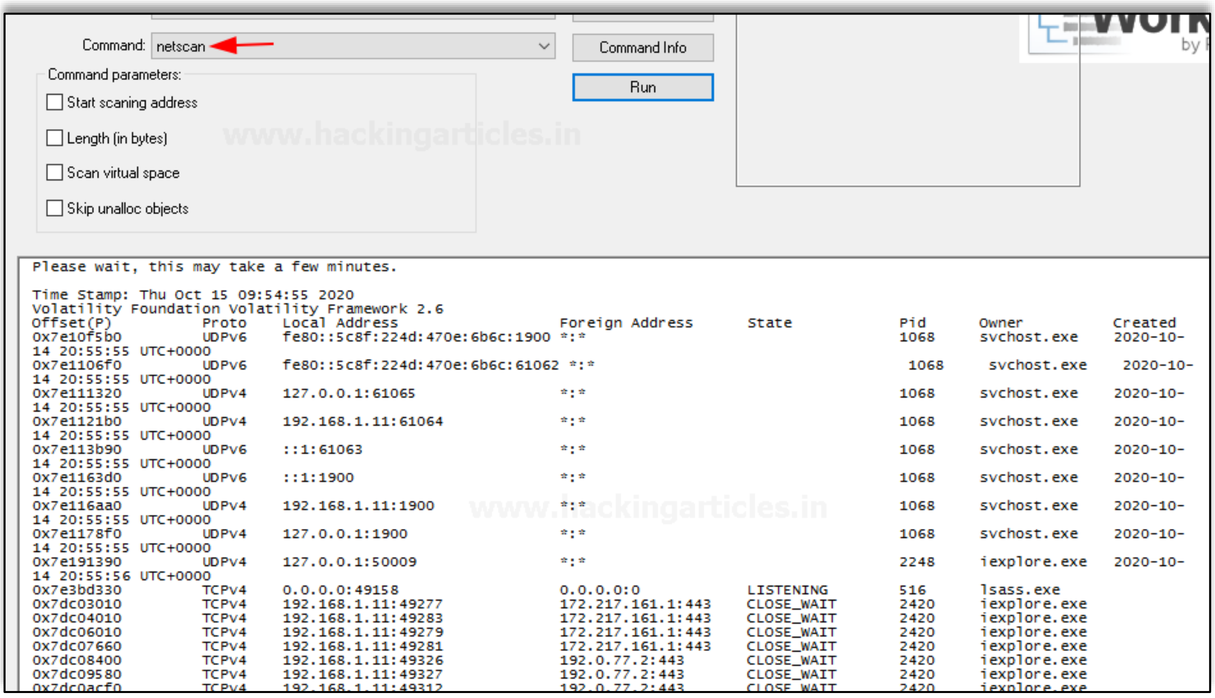
This command is used to find the thread objects that are present in the physical memory with the help of a pool tag scan. It contains certain fields that can identify its parent processes which can help in finding hidden processes.



```
Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:53:10 2020
Volatility Foundation Volatility Framework 2.6
Offset(P)             PID    TID      Start Address Create Time                    Exit Time
------------------ ------ ------ ------------------ ------------------------------ ------------------------------
0x000000005de2230       4    116 0xfffff80002adb270 2020-10-14 20:55:31 UTC+0000
0x000000005de54d0       4    120 0xfffff80002ba2d30 2020-10-14 20:55:31 UTC+0000
0x000000005de6040       4    124 0xfffff80002ba2d30 2020-10-14 20:55:31 UTC+0000
0x000000005de7320       4    128 0xfffff80002a56f50 2020-10-14 20:55:31 UTC+0000
0x000000005dfb550       4    132 0xfffff80002db0064 2020-10-14 20:55:31 UTC+0000
0x000000001c64b4d0      4    120 0xfffff80002ba2d30 2020-10-14 20:55:31 UTC+0000
0x00000000232cc040      4    124 0xfffff80002ba2d30 2020-10-14 20:55:31 UTC+0000
0x000000007dc058b0   4004   3392         0x7735c500 2020-10-14 20:59:20 UTC+0000
0x000000007dc42b60    608   3288         0x7735c500 2020-10-14 20:59:19 UTC+0000
0x000000007dc5eb60    412   3248         0x7735c500 2020-10-14 20:59:19 UTC+0000
0x000000007dc81660   4004   4036 0xfffff8a0029f4770 2020-10-14 20:59:19 UTC+0000
0x000000007dc81b60    608   3860 0xfffffa801ae81ee0 2020-10-14 20:59:19 UTC+0000
0x000000007de13060   2420   2780         0x7735c500 2020-10-14 20:55:58 UTC+0000
```

22. Netscan

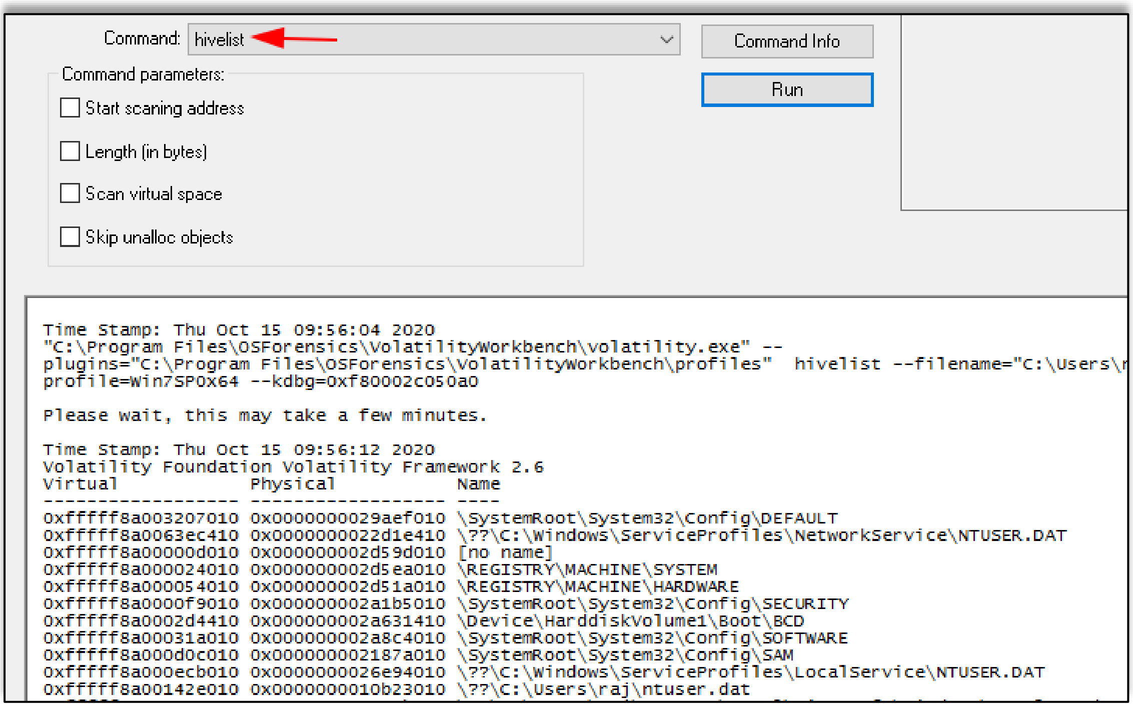
This plugin helps in finding network-related artefacts present in the memory dump. It makes use of pool tag scanning. This plugin finds all the TCP endpoints, TCP listeners, UDP endpoints, and UDP listeners. It provides details about the local and remote IP and also about the local and remote port



```
Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:54:55 2020
Volatility Foundation Volatility Framework 2.6
Offset(P)          Proto    Local Address                  Foreign Address      State            Pid      Owner          Created
0x7e10f5b0         UDPv6    fe80::5c8f:224d:470e:6b6c:1900 *:*                                   1068     svchost.exe    2020-10-
14 20:55:55 UTC+0000
0x7e1106f0         UDPv6    fe80::5c8f:224d:470e:6b6c:61062 *:*                                   1068     svchost.exe    2020-10-
14 20:55:55 UTC+0000
0x7e111320         UDPv4    127.0.0.1:61065                *:*                                   1068     svchost.exe    2020-10-
14 20:55:55 UTC+0000
0x7e1121b0         UDPv4    192.168.1.11:61064             *:*                                   1068     svchost.exe    2020-10-
14 20:55:55 UTC+0000
0x7e113b90         UDPv6    ::1:61063                      *:*                                   1068     svchost.exe    2020-10-
14 20:55:55 UTC+0000
0x7e1163d0         UDPv6    ::1:1900                       *:*                                   1068     svchost.exe    2020-10-
14 20:55:55 UTC+0000
0x7e116aa0         UDPv4    192.168.1.11:1900              *:*                                   1068     svchost.exe    2020-10-
14 20:55:55 UTC+0000
0x7e1178f0         UDPv4    127.0.0.1:1900                 *:*                                   1068     svchost.exe    2020-10-
14 20:55:55 UTC+0000
0x7e191390         UDPv4    127.0.0.1:50009                *:*                                   2248     iexplore.exe   2020-10-
14 20:55:56 UTC+0000
0x7e3bd330         TCPv4    0.0.0.0:49158                  0.0.0.0:0            LISTENING        516      lsass.exe
0x7dc03010         TCPv4    192.168.1.11:49277             172.217.161.1:443    CLOSE_WAIT       2420     iexplore.exe
0x7dc04010         TCPv4    192.168.1.11:49283             172.217.161.1:443    CLOSE_WAIT       2420     iexplore.exe
0x7dc06010         TCPv4    192.168.1.11:49279             172.217.161.1:443    CLOSE_WAIT       2420     iexplore.exe
0x7dc07660         TCPv4    192.168.1.11:49281             172.217.161.1:443    CLOSE_WAIT       2420     iexplore.exe
0x7dc08400         TCPv4    192.168.1.11:49326             192.0.77.2:443       CLOSE_WAIT       2420     iexplore.exe
0x7dc09580         TCPv4    192.168.1.11:49327             192.0.77.2:443       CLOSE_WAIT       2420     iexplore.exe
0x7dc0acf0         TCPv4    192.168.1.11:49312             192.0.77.2:443       CLOSE_WAIT       2420     iexplore.exe
```

23. Hivelist

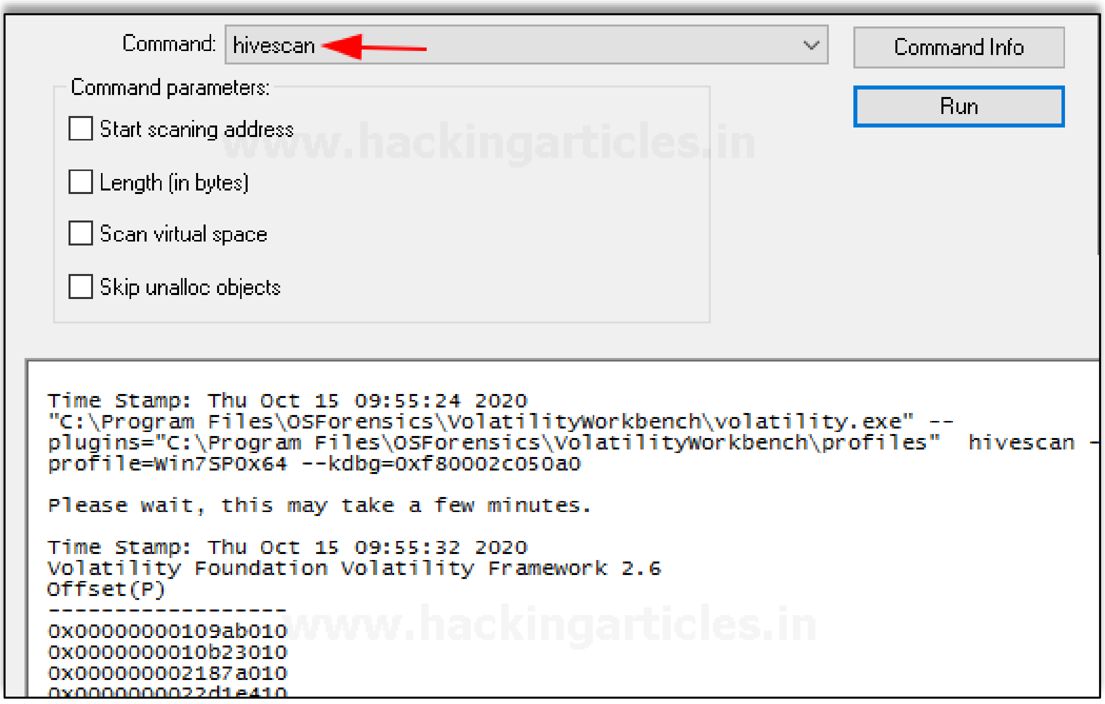
This command can be used to locate the virtual addresses present in the registry hives in memory, and their entire paths to hive on the disk.



```
Time Stamp: Thu Oct 15 09:56:04 2020
"C:\Program Files\OSForensics\VolatilityWorkbench\volatility.exe" --
plugins="C:\Program Files\OSForensics\VolatilityWorkbench\profiles"  hivelist --filename="C:\Users\r
profile=Win7SP0x64 --kdbg=0xf80002c050a0

Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:56:12 2020
Volatility Foundation Volatility Framework 2.6
Virtual            Physical           Name
------------------ ------------------ ----
0xfffff8a003207010 0x0000000029aef010 \SystemRoot\System32\Config\DEFAULT
0xfffff8a0063ec410 0x0000000022d1e410 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0xfffff8a00000d010 0x000000002d59d010 [no name]
0xfffff8a000024010 0x000000002d5ea010 \REGISTRY\MACHINE\SYSTEM
0xfffff8a000054010 0x000000002d51a010 \REGISTRY\MACHINE\HARDWARE
0xfffff8a0000f9010 0x000000002a1b5010 \SystemRoot\System32\Config\SECURITY
0xfffff8a0002d4410 0x000000002a631410 \Device\HarddiskVolume1\Boot\BCD
0xfffff8a00031a010 0x000000002a8c4010 \SystemRoot\System32\Config\SOFTWARE
0xfffff8a000d0c010 0x000000002187a010 \SystemRoot\System32\Config\SAM
0xfffff8a000ecb010 0x0000000026e94010 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0xfffff8a00142e010 0x0000000010b23010 \??\C:\Users\raj\ntuser.dat
```

24. Hivescan

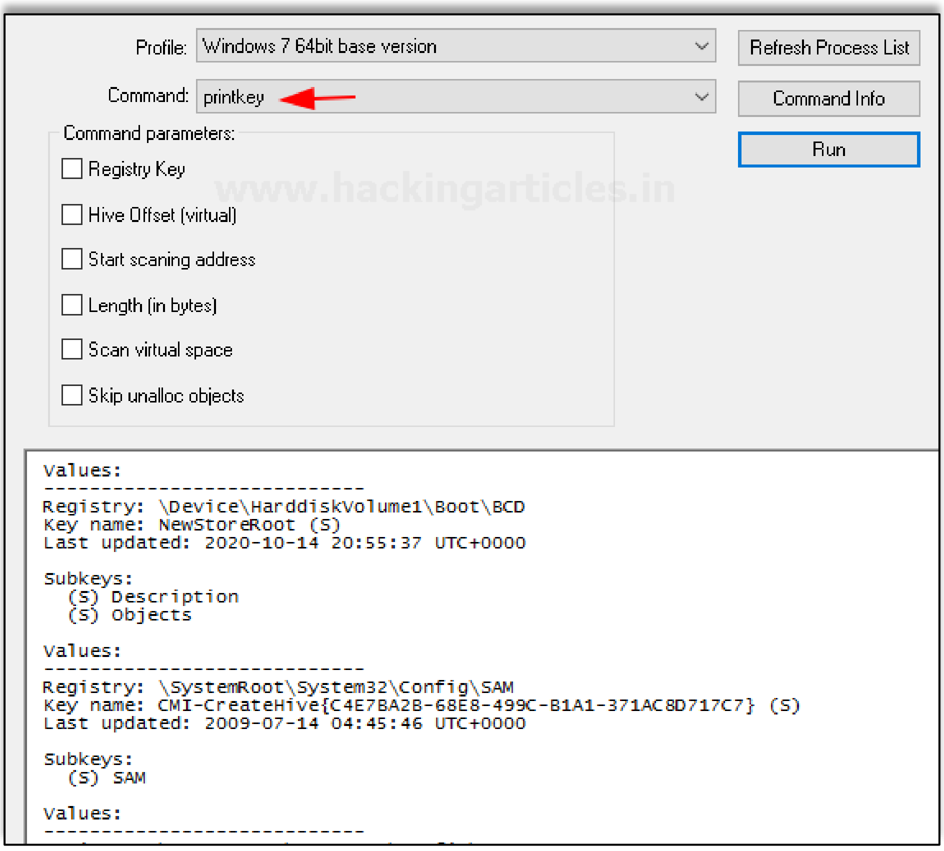
This command is used to find the physical address of the registry hives that are present in the memory. It is there to support the hivelist.



```
Time Stamp: Thu Oct 15 09:55:24 2020
"C:\Program Files\OSForensics\VolatilityWorkbench\volatility.exe" --
plugins="C:\Program Files\OSForensics\VolatilityWorkbench\profiles"  hivescan -
profile=Win7SP0x64 --kdbg=0xf80002c050a0

Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:55:32 2020
Volatility Foundation Volatility Framework 2.6
Offset(P)
------------------
0x00000000109ab010
0x0000000010b23010
0x000000002187a010
0x0000000022d1e410
```

25. Printkey

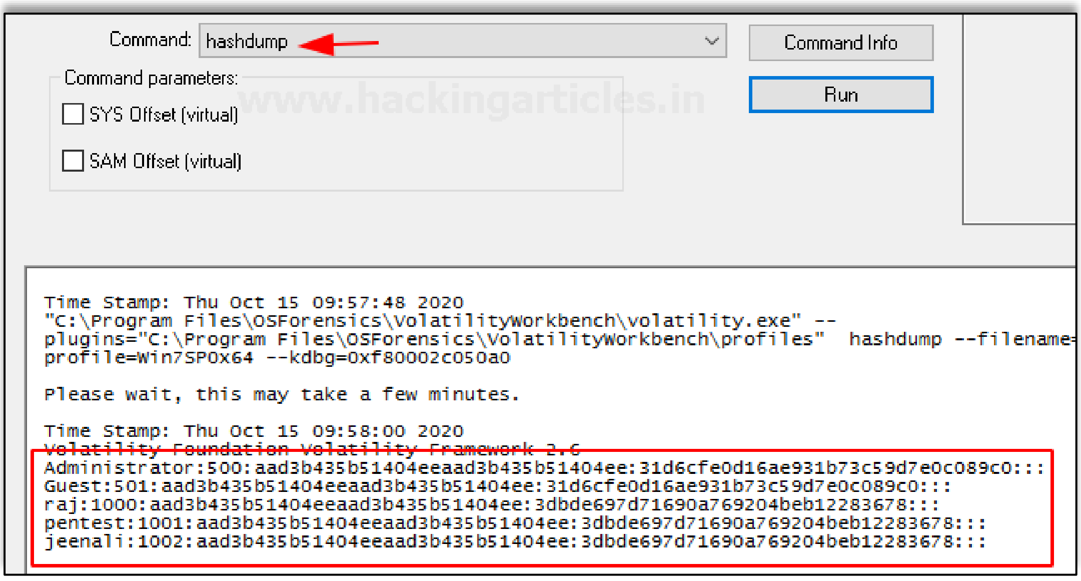
This command is used to display the values, data, subkeys, and data types that are present in a specified registry.



```
Values:
----------------------------
Registry: \Device\HarddiskVolume1\Boot\BCD
Key name: NewStoreRoot (S)
Last updated: 2020-10-14 20:55:37 UTC+0000

Subkeys:
  (S) Description
  (S) Objects

Values:
----------------------------
Registry: \SystemRoot\System32\Config\SAM
Key name: CMI-CreateHive{C4E7BA2B-68E8-499C-B1A1-371AC8D717C7} (S)
Last updated: 2009-07-14 04:45:46 UTC+0000

Subkeys:
  (S) SAM

Values:
----------------------------
```

26. Hashdump

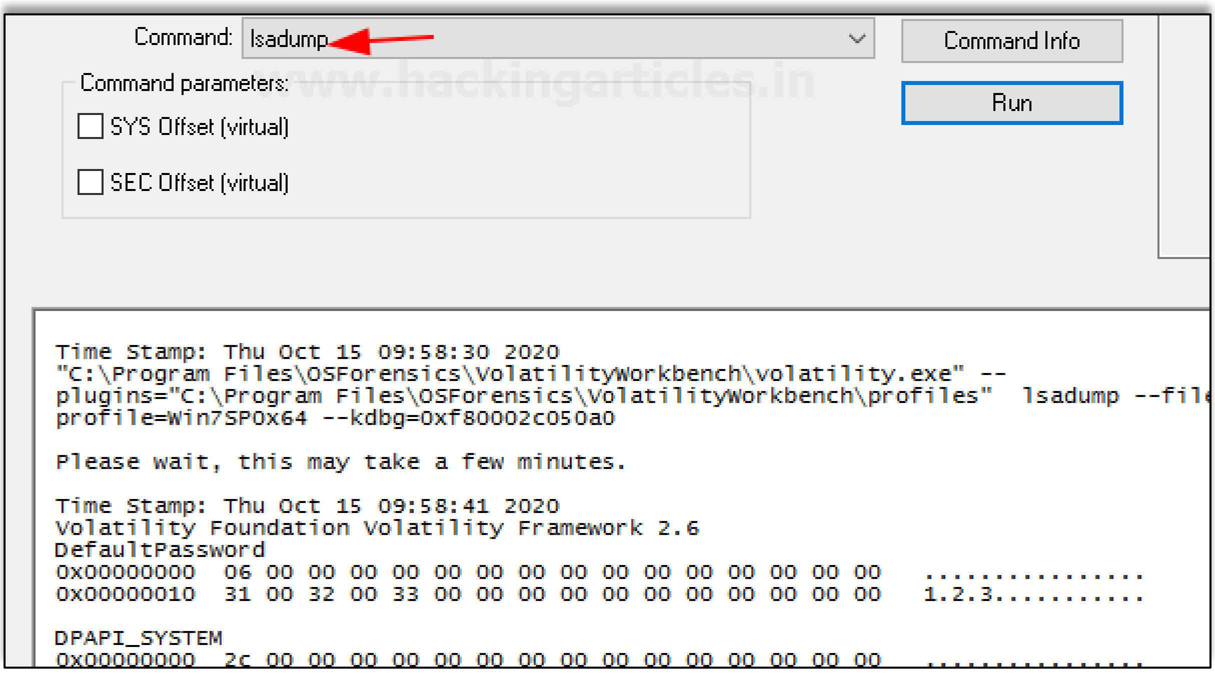
This command can be used to extract and decrypt cached domain credentials stored in the registry which can be availed from the memory dump. The hashes that are availed from the memory dump can be cracked using John the Ripper, Hashcat, etc



```
Time Stamp: Thu Oct 15 09:57:48 2020
"C:\Program Files\OSForensics\VolatilityWorkbench\volatility.exe" --
plugins="C:\Program Files\OSForensics\VolatilityWorkbench\profiles"  hashdump --filename=
profile=Win7SP0x64 --kdbg=0xf80002c050a0

Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:58:00 2020
Volatility Foundation Volatility Framework 2.6
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
raj:1000:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
pentest:1001:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
jeenali:1002:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
```

27. Lsadump

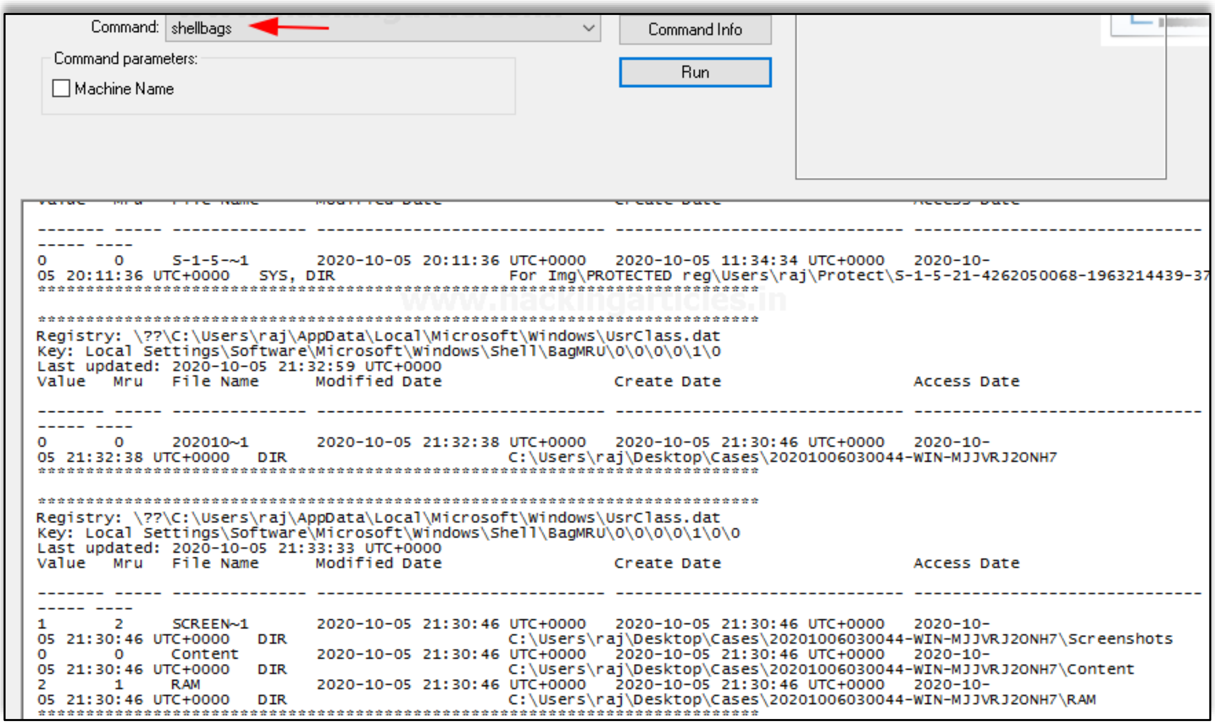
This command is used to dump LSA secrets from the registry in the memory dump. This plugin gives out information like the default password, the RDP public key, etc.



```
Time Stamp: Thu Oct 15 09:58:30 2020
"C:\Program Files\OSForensics\VolatilityWorkbench\volatility.exe" --
plugins="C:\Program Files\OSForensics\VolatilityWorkbench\profiles"  lsadump --file
profile=Win7SP0x64 --kdbg=0xf80002c050a0

Please wait, this may take a few minutes.

Time Stamp: Thu Oct 15 09:58:41 2020
Volatility Foundation Volatility Framework 2.6
DefaultPassword
0x00000000   06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00000010   31 00 32 00 33 00 00 00 00 00 00 00 00 00 00 00   1.2.3...........

DPAPI_SYSTEM
0x00000000   2c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

28. Shellbags

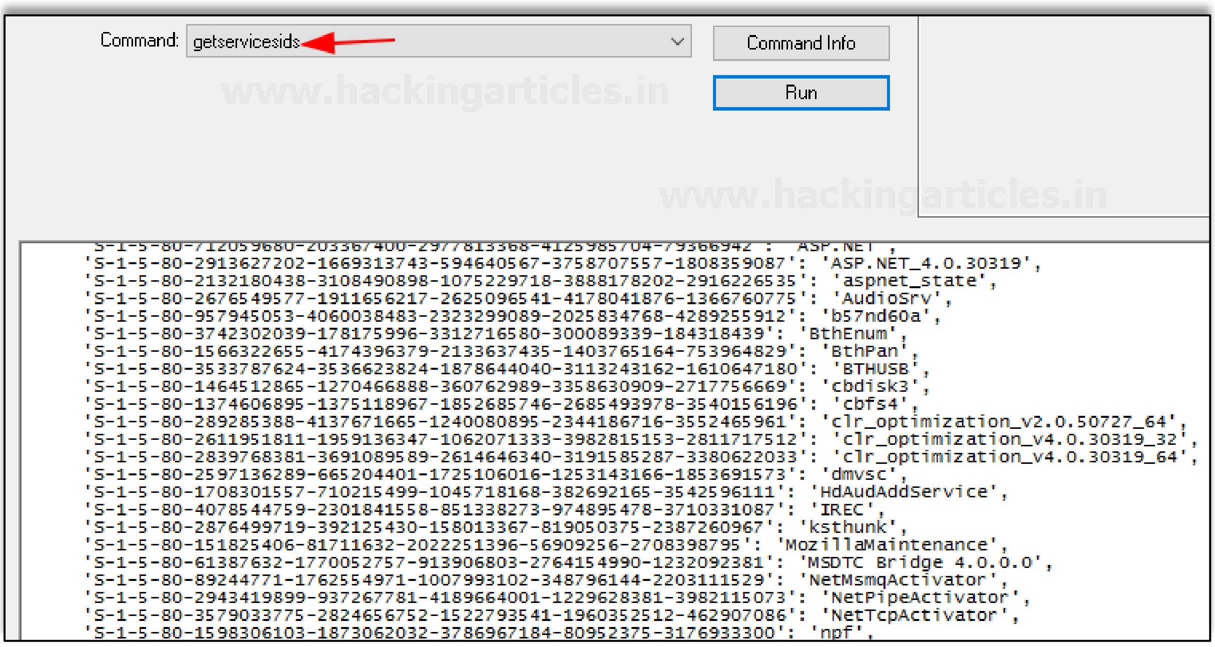
This command usually parses and prints the shellbag information that is obtained from the registry.



```
------- ----- -------------- ------------------------------ ------------------------------ ------------------------------
----- ----
0       0     S-1-5-~1       2020-10-05 20:11:36 UTC+0000   2020-10-05 11:34:34 UTC+0000   2020-10-
05 20:11:36 UTC+0000   SYS, DIR                   For Img\PROTECTED reg\Users\raj\Protect\S-1-5-21-4262050068-1963214439-37
***************************************************************************

***************************************************************************
Registry: \??\C:\Users\raj\AppData\Local\Microsoft\Windows\UsrClass.dat
Key: Local Settings\Software\Microsoft\Windows\Shell\BagMRU\0\0\0\0\1\0
Last updated: 2020-10-05 21:32:59 UTC+0000
Value   Mru   File Name      Modified Date                  Create Date                    Access Date

------- ----- -------------- ------------------------------ ------------------------------ ------------------------------
----- ----
0       0     202010~1       2020-10-05 21:32:38 UTC+0000   2020-10-05 21:30:46 UTC+0000   2020-10-
05 21:32:38 UTC+0000   DIR                        C:\Users\raj\Desktop\Cases\20201006030044-WIN-MJJVRJ2ONH7
***************************************************************************

***************************************************************************
Registry: \??\C:\Users\raj\AppData\Local\Microsoft\Windows\UsrClass.dat
Key: Local Settings\Software\Microsoft\Windows\Shell\BagMRU\0\0\0\0\1\0\0
Last updated: 2020-10-05 21:33:33 UTC+0000
Value   Mru   File Name      Modified Date                  Create Date                    Access Date

------- ----- -------------- ------------------------------ ------------------------------ ------------------------------
----- ----
1       2     SCREEN~1       2020-10-05 21:30:46 UTC+0000   2020-10-05 21:30:46 UTC+0000   2020-10-
05 21:30:46 UTC+0000   DIR                        C:\Users\raj\Desktop\Cases\20201006030044-WIN-MJJVRJ2ONH7\Screenshots
0       0     Content        2020-10-05 21:30:46 UTC+0000   2020-10-05 21:30:46 UTC+0000   2020-10-
05 21:30:46 UTC+0000   DIR                        C:\Users\raj\Desktop\Cases\20201006030044-WIN-MJJVRJ2ONH7\Content
2       1     RAM            2020-10-05 21:30:46 UTC+0000   2020-10-05 21:30:46 UTC+0000   2020-10-
05 21:30:46 UTC+0000   DIR                        C:\Users\raj\Desktop\Cases\20201006030044-WIN-MJJVRJ2ONH7\RAM
***************************************************************************
```

29. Getservicesids

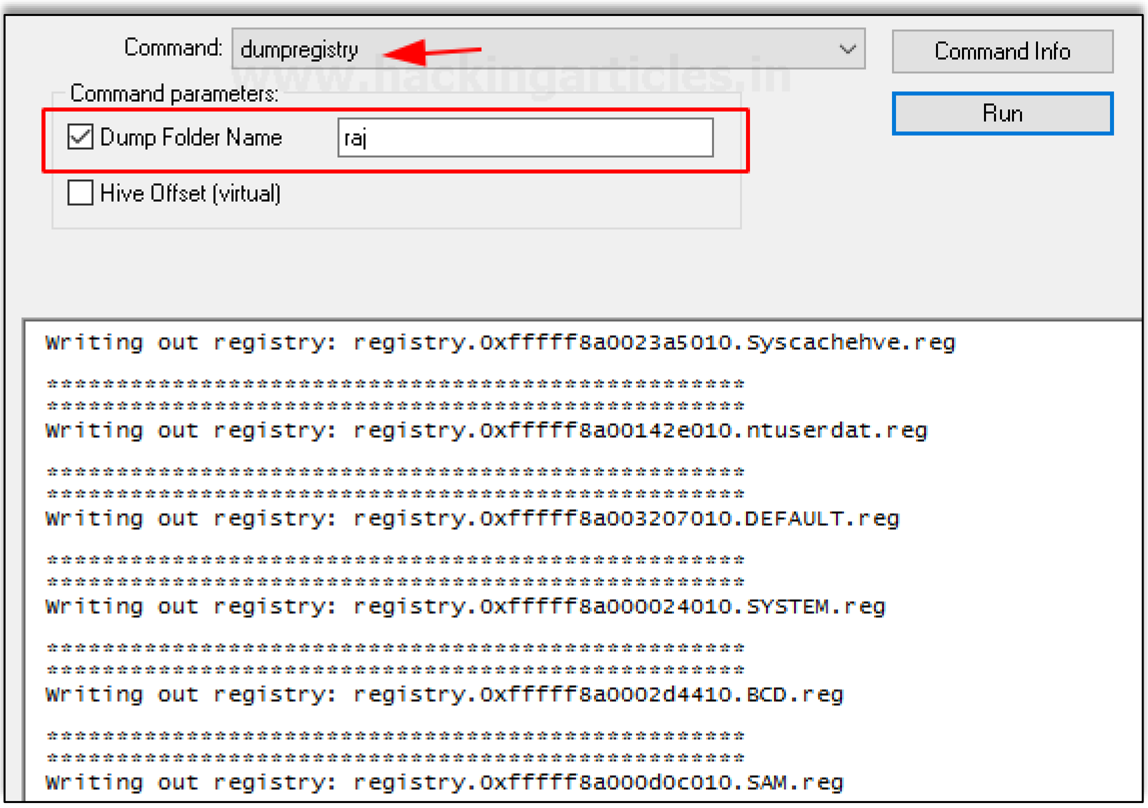
This command does the work of calculating the SIDz for the services that are present on the machine. The name of the services has been taken from the registry.



```
     'S-1-5-80-712059680-2033367400-2977813368-4125985704-79366942': 'ASP.NET',
     'S-1-5-80-2913627202-1669313743-594640567-3758707557-1808359087': 'ASP.NET_4.0.30319',
     'S-1-5-80-2132180438-3108490898-1075229718-3888178202-2916226535': 'aspnet_state',
     'S-1-5-80-2676549577-1911656217-2625096541-4178041876-1366760775': 'AudioSrv',
     'S-1-5-80-957945053-4060038483-2323299089-2025834768-4289255912': 'b57nd60a',
     'S-1-5-80-3742302039-178175996-3312716580-300089339-184318439': 'BthEnum',
     'S-1-5-80-1566322655-4174396379-2133637435-1403765164-753964829': 'BthPan',
     'S-1-5-80-3533787624-3536623824-1878644040-3113243162-1610647180': 'BTHUSB',
     'S-1-5-80-1464512865-1270466888-360762989-3358630909-2717756669': 'cbdisk3',
     'S-1-5-80-1374606895-1375118967-1852685746-2685493978-3540156196': 'cbfs4',
     'S-1-5-80-289285388-4137671665-1240080895-2344186716-3552465961': 'clr_optimization_v2.0.50727_64',
     'S-1-5-80-2611951811-1959136347-1062071333-3982815153-2811717512': 'clr_optimization_v4.0.30319_32',
     'S-1-5-80-2839768381-3691089589-2614646340-3191585287-3380622033': 'clr_optimization_v4.0.30319_64',
     'S-1-5-80-2597136289-665204401-1725106016-1253143166-1853691573': 'dmvsc',
     'S-1-5-80-1708301557-710215499-1045718168-382692165-3542596111': 'HdAudAddService',
     'S-1-5-80-407854759-2301841558-851338273-974895478-3710331087': 'IREC',
     'S-1-5-80-2876499719-392125430-158013367-819050375-2387260967': 'ksthunk',
     'S-1-5-80-151825406-81711632-2022251396-56909256-2708398795': 'MozillaMaintenance',
     'S-1-5-80-61387632-1770052757-913906803-2764154990-1232092381': 'MSDTC Bridge 4.0.0.0',
     'S-1-5-80-89244771-1762554971-1007993102-348796144-2203111529': 'NetMsmqActivator',
     'S-1-5-80-2943419899-937267781-4189664001-1229628381-3982115073': 'NetPipeActivator',
     'S-1-5-80-3579033775-2824656752-1522793541-1960352512-462907086': 'NetTcpActivator',
     'S-1-5-80-1598306103-1873062032-3786967184-80952375-3176933300': 'npf',
```

30. Dumpregistry

This plugin allows one to dump a registry hive into a disk location.

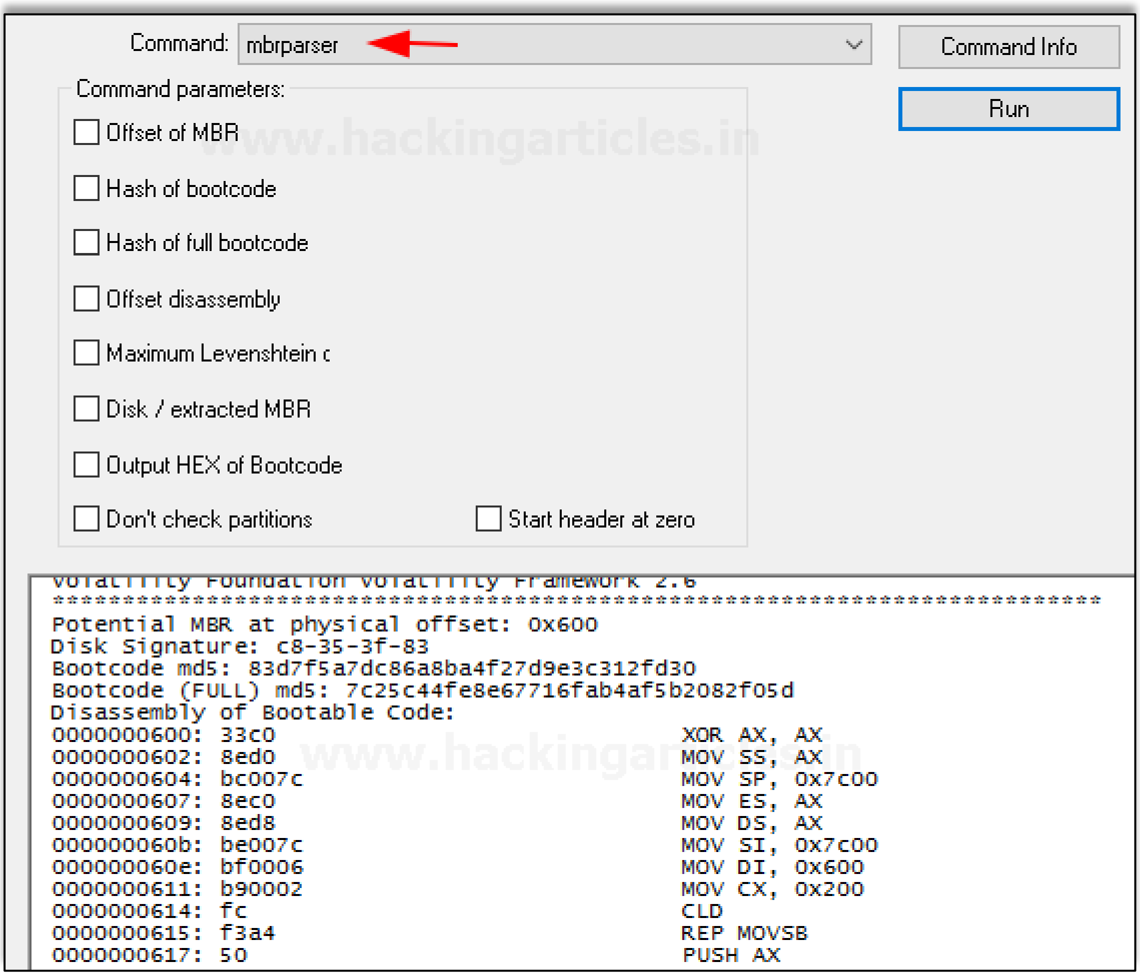


☑ Dump Folder Name: raj
☐ Hive Offset (virtual)

```
Writing out registry: registry.0xfffff8a0023a5010.Syscachehve.reg

**************************************************
**************************************************
Writing out registry: registry.0xfffff8a00142e010.ntuserdat.reg

**************************************************
**************************************************
Writing out registry: registry.0xfffff8a003207010.DEFAULT.reg

**************************************************
**************************************************
Writing out registry: registry.0xfffff8a000024010.SYSTEM.reg

**************************************************
**************************************************
Writing out registry: registry.0xfffff8a0002d4410.BCD.reg

**************************************************
**************************************************
Writing out registry: registry.0xfffff8a000d0c010.SAM.reg
```

31. Mbrparser

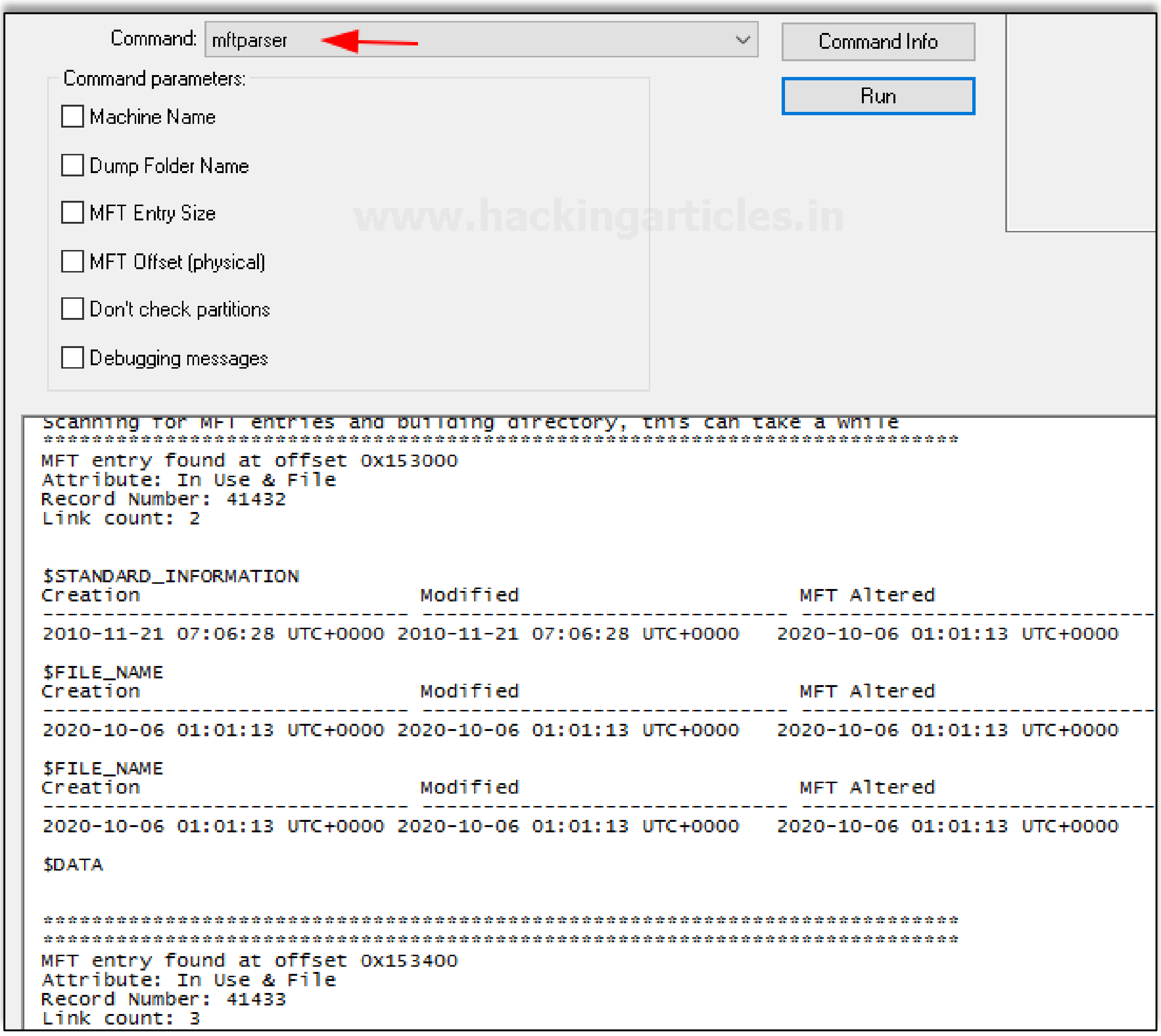
This command scans and parses potential MBR from the memory dump. There are various ways to find MBR and the way of filtering it.



```
Volatility Foundation Volatility Framework 2.6
******************************************************************************
Potential MBR at physical offset: 0x600
Disk Signature: c8-35-3f-83
Bootcode md5: 83d7f5a7dc86a8ba4f27d9e3c312fd30
Bootcode (FULL) md5: 7c25c44fe8e67716fab4af5b2082f05d
Disassembly of Bootable Code:
0000000600: 33c0                           XOR AX, AX
0000000602: 8ed0                           MOV SS, AX
0000000604: bc007c                         MOV SP, 0x7c00
0000000607: 8ec0                           MOV ES, AX
0000000609: 8ed8                           MOV DS, AX
000000060b: be007c                         MOV SI, 0x7c00
000000060e: bf0006                         MOV DI, 0x600
0000000611: b90002                         MOV CX, 0x200
0000000614: fc                             CLD
0000000615: f3a4                           REP MOVSB
0000000617: 50                             PUSH AX
```

32. Mftparser

This command is used to scan the MFT entries in the memory dump and prints out the information for certain types of file attributes.



```
Scanning for MFT entries and building directory, this can take a while
***************************************************************************
MFT entry found at offset 0x153000
Attribute: In Use & File
Record Number: 41432
Link count: 2


$STANDARD_INFORMATION
Creation                       Modified                       MFT Altered
------------------------------ ------------------------------ ------------------------------
2010-11-21 07:06:28 UTC+0000 2010-11-21 07:06:28 UTC+0000   2020-10-06 01:01:13 UTC+0000

$FILE_NAME
Creation                       Modified                       MFT Altered
------------------------------ ------------------------------ ------------------------------
2020-10-06 01:01:13 UTC+0000 2020-10-06 01:01:13 UTC+0000   2020-10-06 01:01:13 UTC+0000

$FILE_NAME
Creation                       Modified                       MFT Altered
------------------------------ ------------------------------ ------------------------------
2020-10-06 01:01:13 UTC+0000 2020-10-06 01:01:13 UTC+0000   2020-10-06 01:01:13 UTC+0000

$DATA


***************************************************************************
***************************************************************************
MFT entry found at offset 0x153400
Attribute: In Use & File
Record Number: 41433
Link count: 3
```

References

- https://www.hackingarticles.in/memory-forensics-using-volatility-framework/
- https://www.hackingarticles.in/memory-forensics-using-volatility-workbench/

About Us

“Simple training makes Deep Learning”

“IGNITE” is a worldwide name in IT field. As we provide high-quality cybersecurity training and consulting services that fulfil students, government and corporate requirements.
We are working towards the vision to “Develop India as a Cyber Secured Country”. With an outreach to over eighty thousand students and over a thousand major colleges, Ignite Technologies stood out to be a trusted brand in the Education and the Information Security structure.

We provide training and education in the field of Ethical Hacking & Information Security to the students of schools and colleges along with the corporate world. The training can be provided at the client’s location or even at Ignite’s Training Center.
We have trained over 10,000 + individuals across the globe, ranging from students to security experts from different fields. Our trainers are acknowledged as Security Researcher by the Top Companies like - Facebook, Google, Microsoft, Adobe, Nokia, Paypal, Blackberry, AT&T and many more. Even the trained students are placed into a number of top MNC's all around the globe. Over with this, we are having International experience of training more than 400+ individuals.

The two brands, Ignite Technologies & Hacking Articles have been collaboratively working from past 10+ Years with about more than 100+ security researchers, who themselves have been recognized by several research paper publishing organizations, The Big 4 companies, Bug Bounty research programs and many more.

Along with all these things, all the major certification organizations recommend Ignite's training for its resources and guidance.
Ignite's research had been a part of number of global Institutes and colleges, and even a multitude of research papers shares Ignite's researchers in their reference.

What We Offer

Ethical Hacking

The Ethical Hacking course has been structured in such a way that a technical or a non-technical applicant can easily absorb its features and indulge his/her career in the field of IT security.

Bug Bounty 2.0

A bug bounty program is a pact offered by many websites and web developers by which folks can receive appreciation and reimbursement for reporting bugs, especially those affecting to exploits and vulnerabilities.
Over with this training, an indivisual is thus able to determine and report bugs to the authorized before the general public is aware of them, preventing incidents of widespread abuse.

Network Penetration Testing 2.0

The Network Penetration Testing training will build up the basic as well advance skills of an indivisual with the concept of Network Security & Organizational Infrastructure. Thereby this course will make the indivisual stand out of the crowd within just 45 days.

Red Teaming

This training will make you think like an "Adversary" with its systematic structure & real Environment Practice that contains more than 75 practicals on Windows Server 2016 & Windows 10. This course is especially designed for the professionals to enhance their Cyber Security Skills

CTF 2.0

The CTF 2.0 is the latest edition that provides more advance module connecting to real infrastructure organization as well as supporting other students preparing for global certification. This curriculum is very easily designed to allow a fresher or specialist to become familiar with the entire content of the course.

Infrastructure Penetration Testing

This course is designed for Professional and provides an hands-on experience in Vulnerability Assessment Penetration Testing & Secure configuration Testing for Applications Servers, Network Deivces, Container and etc.

Digital Forensic

Digital forensics provides a taster in the understanding of how to conduct investigations in order for business and legal audien ces to correctly gather and analyze digital evidence.