Volatility: Plugins

Written by

Dan Doonan and Catherine Stamm

Researched by

Dan Doonan, Connor Hicks, David Lebelfinger, and Catherine Stamm

The Senator Patrick Leahy Center for Digital Investigation

Champlain College

November 5, 2012

Disclaimer:

This document contains information based on research that has been gathered by employee(s) of The Senator Patrick Leahy Center for Digital Investigation (LCDI). The data contained in this project is submitted voluntarily and is unaudited. Every effort has been made by LCDI to assure the accuracy and reliability of the data contained in this report. However, LCDI nor any of our employees make no representation, warranty or guarantee in connection with this report and hereby expressly disclaims any liability or responsibility for loss or damage resulting from use of this data. Information in this report can be downloaded and redistributed by any person or persons. Any redistribution must maintain the LCDI logo and any references from this report must be properly annotated.

Contents

1 Introduction

Volatility is a forensic framework that utilizes multiple tools in order to analyze memory images. This Python-based tool aids investigators in finding out more about volatile memory on a system by extracting running processes, computer profiles, open network connections, hidden injections, possible malware, and more.

RAM can hold traces of malicious code, data that may have been taken from the system, usernames and passwords, contents of an open window, registry keys, and other pieces of data that can be used in an investigation. Since RAM is volatile, the data is gone as soon as the system powers off. To save the contents of RAM, certain forensic tools can be used to acquire the memory, and from there, Volatility can be used to analyze what was captured, presenting the investigator with all sorts of evidence. Running processes, passwords, network connections and numerous lists will be displayed to help an examiner piece together what could have happened within a system. The evidence provided by Volatility can make all the difference to a case and, if used to its fullest potential, can present enough information to develop a solid understanding of how a system was being used during the time of acquisition.

1.1 Background

Because Volatility is an open source tool, developments are continuing over time. There are numerous blogs dedicated to Volatility's functions regarding different types of situations, such as examining hiberfil.sys files or analyzing rootkits. September was the Month of Volatility, as a lot of new plugins were added to the framework. These new plugins are currently be researched by ourselves and others in the industry.

1.2 Terminology

This report will outline the plugins that are most frequently used in an investigation, as well as the plugins that were added to the framework in September. Dan has created a list of these plugins, tested them, and given a brief description of how to use them and why they are important to a forensic investigation.

Below are some important keywords that may be unfamiliar:

Volatile: Data that is not permanent; it will be lost once power is cut from a system.

Plugins: Software that makes a larger piece of software more capable.

Framework: A structure or set of forensic tools that support an investigation.

1.3 Research Questions

What Volatility plugins are used most often?

What are their commands and functions?

How is Volatility installed and used?

How can Volatility's findings aid an investigation?

2 Basic Commands

Before getting started with Volatility, the framework must be downloaded and installed. A list of Volatility downloads can be found here: http://code.google.com/p/volatility/downloads/list. For a Windows user, it is easiest to use the Standalone version, which is what we used for the basis of this research. There is also a download for the source code to aide in developing plugins for Volatility or to look into how the program actually works. Volatility is a part of the SIFT Workstation, which can be found here: http://computer-forensics.sans.org/community/downloads#locations.

Once Volatility is downloaded, it is recommended that you put it in an easily accessible area on your system, such as the C drive or a folder on your desktop. To get the Standalone version of Volatility to work, you can run the command prompt as an administrator and change directories to the location of Volatility. If you moved Volatility to your C drive, then to get it running you would change directories to the C drive using the cd.. command. Next, type in “volatility-2.2.standalone.exe –h” (omitting the quotes). This will list the help options, along with the commands for different plugins. From there, you can input whatever it is you would like Volatility to do. Typically, the next step would be to have Volatility gather information on the memory image. To do this, input “volatility-2.2.standalone.exe –f <path to memory image> imageinfo”(again omitting quotes). This will display what operating system the memory image game from, when the image was taken, how many processors the system has, and other information that can be used in the investigation

3 Frequently Used Plugins

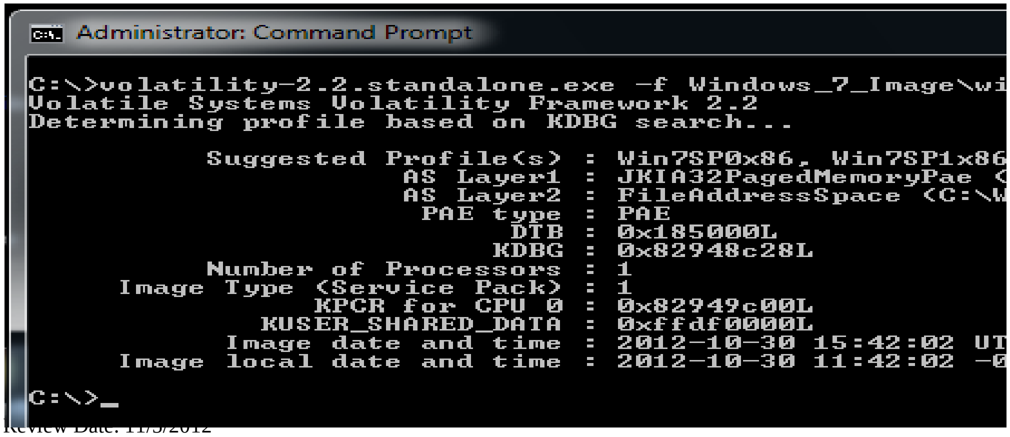
Part one of this Volatility project was spent researching the plugins that law enforcement and examiners would most likely use in a case. It is vital to know how to run these commands and understand when to use them during an investigation, as Volatility can be a tricky program to use.

Images

Plugins relating to this section identify the memory image being analyzed and provide a basic understanding of what the image contains.

a. **Imageinfo**: Imageinfo identifies the memory image and suggests a profile to use. Volatility requires that you specify what operating system the memory image came from. This command identifies the operating system so that you can run other commands.

    i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> imageinfo



```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\wi
Volatile Systems Volatility Framework 2.2
Determining profile based on KDBG search...

          Suggested Profile(s) : Win7SP0x86, Win7SP1x86
                     AS Layer1 : JKIA32PagedMemoryPae (
                     AS Layer2 : FileAddressSpace (C:\W
                      PAE type : PAE
                           DTB : 0x185000L
                          KDBG : 0x82948c28L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0x82949c00L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2012-10-30 15:42:02 UT
     Image local date and time : 2012-10-30 11:42:02 -0

C:\>_
```

b. **Crashinfo**: This plugin displays information stored in a crashdump header.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> crashinfo

   ii. Displays:

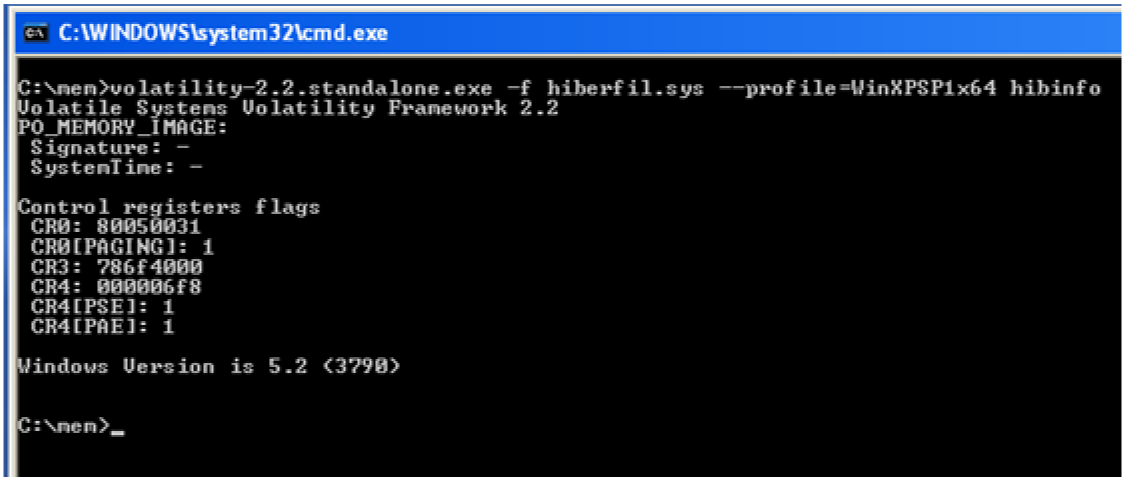
   - MajorVersion
   - MinorVersion
   - KdSecondaryVersion
   - DirectoryTableBase
   - PfnDataBase
   - PsLoadedModuleList
   - PsActiveProcessHead
   - MachineImageType
   - NumberProcessors
   - BugCheckCode
   - KdDebuggerDataBlock
   - ProductType
   - SuiteMask
   - WriterStatus
   - Comment
   - Physical Memory Description

c. **Hibinfo**: This plugin dumps hibernation file information if the system was ever in that mode.

   i. Usage: volatility-2.2.standalone.exe -f <path to image> --profile=<profile> hibinfo

   ii. Displays:

   - Signature
   - System Time
   - Control registers flags
   - Windows Version



```
C:\mem>volatility-2.2.standalone.exe -f hiberfil.sys --profile=WinXPSP1x64 hibinfo
Volatile Systems Volatility Framework 2.2
PO_MEMORY_IMAGE:
 Signature: -
 SystemTime: -

Control registers flags
 CR0: 80050031
 CR0[PAGING]: 1
 CR3: 786f4000
 CR4: 000006f8
 CR4[PSE]: 1
 CR4[PAE]: 1

Windows Version is 5.2 (3790)


C:\mem>_
```

d. **Imagecopy**: Imagecopy copies a physical address space out as a raw drive image (dd)

  i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> imagecopy –O <output file>

e. **Raw2dmp**: This plugin converts a physical memory sample to a windbg crash dump.

  i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> raw2dmp –O <output file>

Processes and DLLs

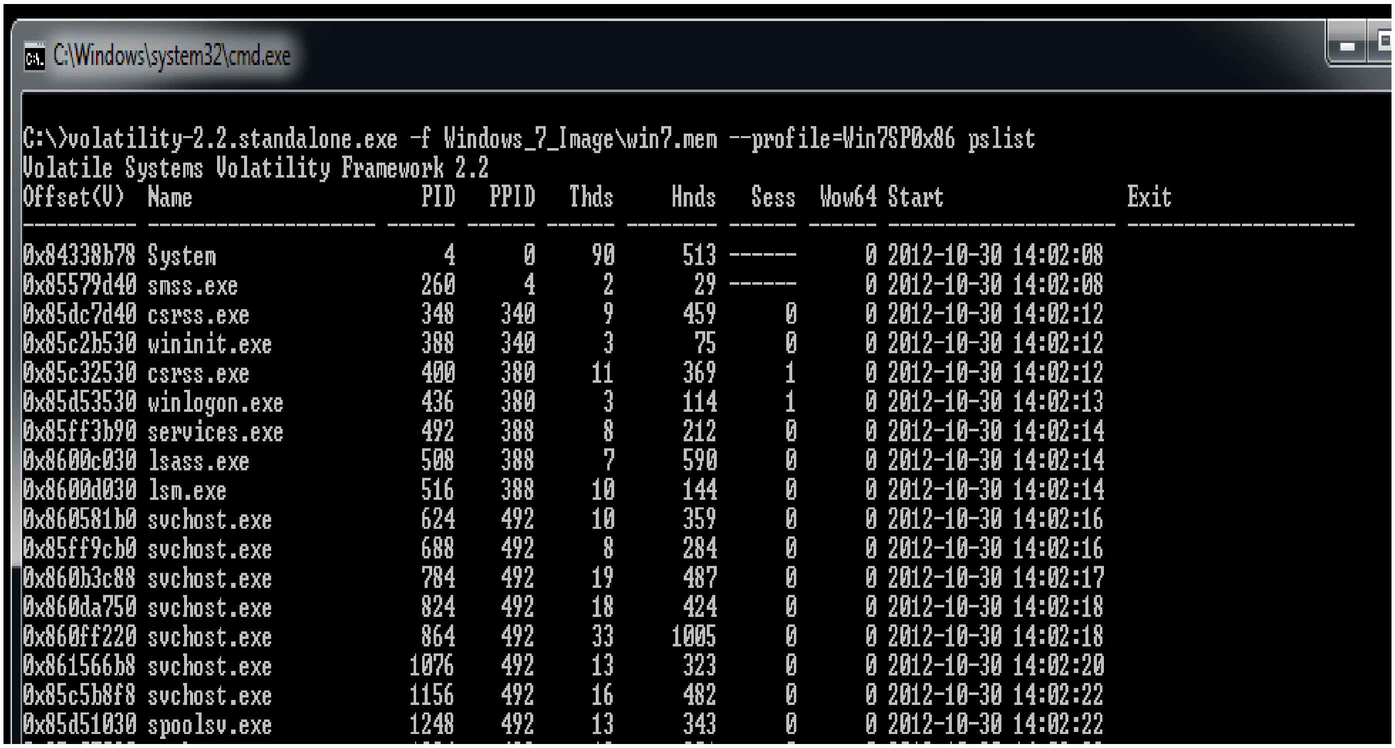
Plugins relating to this section determine running processes at the time of memory capture and can find hidden DLLs.

a. **Pslist**: Pslist prints all running processes by following the EPROCESS lists. This command will display every running process on a system and could be used to prove that a specific process was open, or to look for a suspicious process in an investigation.

  i. Usage: volatility-2.2.standalone.exe –f <path to image> pslist

  ii. Displays:

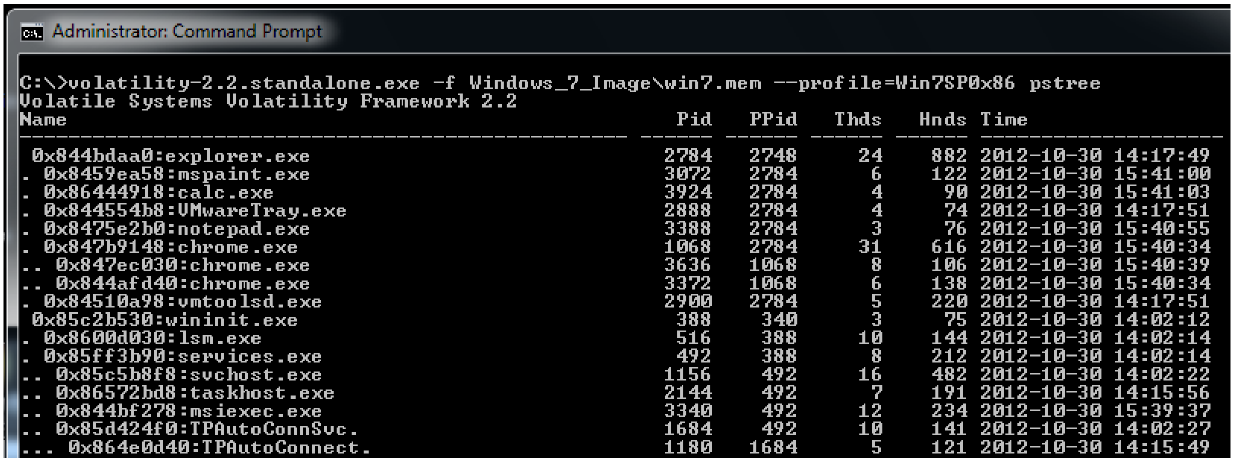
- Offset (By default Virtual Offset, -P for Physical)
- Name
- PID
- PPID
- Threads
- Number of Handles
- Session ID (System and smss.exe will not have a Session ID)
- If it is a Wow64 process
- Start/Exit time



```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 pslist
Volatile Systems Volatility Framework 2.2
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ -------------------- --------------------
0x84338b78 System                    4      0     90      513 ------      0 2012-10-30 14:02:08
0x85579d40 smss.exe                260      4      2       29 ------      0 2012-10-30 14:02:08
0x85dc7d40 csrss.exe               348    340      9      459      0      0 2012-10-30 14:02:12
0x85c2b530 wininit.exe             388    340      3       75      0      0 2012-10-30 14:02:12
0x85c32530 csrss.exe               400    380     11      369      1      0 2012-10-30 14:02:12
0x85d53530 winlogon.exe            436    380      3      114      1      0 2012-10-30 14:02:13
0x85ff3b90 services.exe            492    388      8      212      0      0 2012-10-30 14:02:14
0x8600c030 lsass.exe               508    388      7      590      0      0 2012-10-30 14:02:14
0x8600d030 lsm.exe                 516    388     10      144      0      0 2012-10-30 14:02:14
0x860581b0 svchost.exe             624    492     10      359      0      0 2012-10-30 14:02:16
0x85ff9cb0 svchost.exe             688    492      8      284      0      0 2012-10-30 14:02:16
0x860b3c88 svchost.exe             784    492     19      487      0      0 2012-10-30 14:02:17
0x860da750 svchost.exe             824    492     18      424      0      0 2012-10-30 14:02:18
0x860ff220 svchost.exe             864    492     33     1005      0      0 2012-10-30 14:02:18
0x861566b8 svchost.exe            1076    492     13      323      0      0 2012-10-30 14:02:20
0x85c5b8f8 svchost.exe            1156    492     16      482      0      0 2012-10-30 14:02:22
0x85d51030 spoolsv.exe            1248    492     13      343      0      0 2012-10-30 14:02:22
```

b. **Pstree**: Pstree prints the process list as a tree. This command displays the same information as pslist, only in tree form. This allows you to see which parent process everything belongs to. This could be used to see if a process is attempting to hide as something else.

   i. Usage: volatility-2.2.standalone.exe –f<path to image> --profile=<profile> pstree



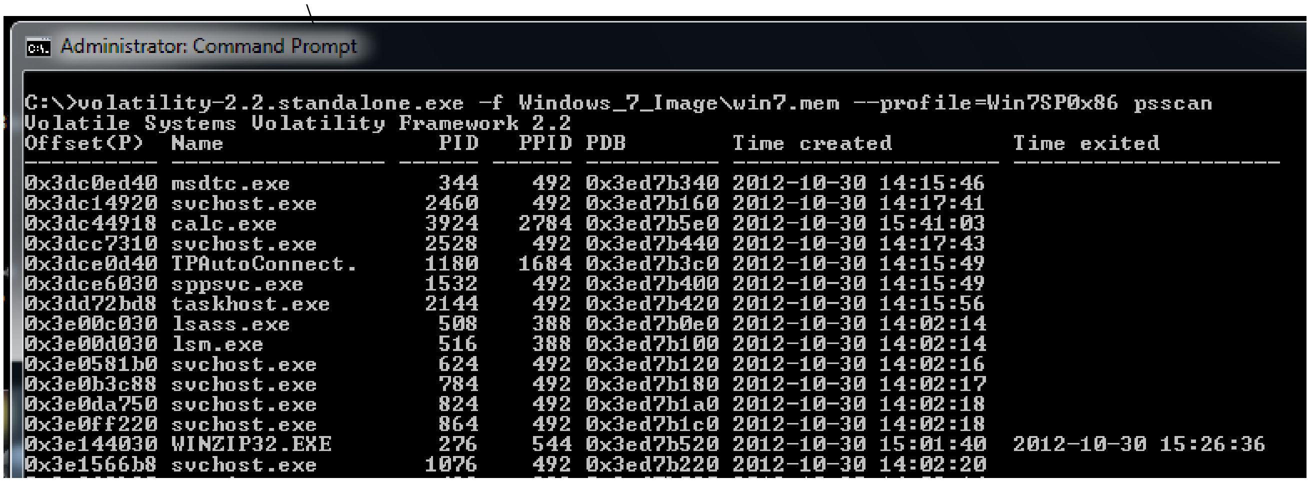
```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 pstree
Volatile Systems Volatility Framework 2.2
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ --------------------
 0x844bdaa0:explorer.exe                             2784   2748     24    882 2012-10-30 14:17:49
. 0x8459ea58:mspaint.exe                             3072   2784      6    122 2012-10-30 15:41:00
. 0x86444918:calc.exe                                3924   2784      4     90 2012-10-30 15:41:03
. 0x844554b8:VMwareTray.exe                          2888   2784      4     74 2012-10-30 14:17:51
. 0x8475e2b0:notepad.exe                             3388   2784      3     76 2012-10-30 15:40:55
. 0x847b9148:chrome.exe                              1068   2784     31    616 2012-10-30 15:40:34
.. 0x847ec030:chrome.exe                             3636   1068      8    106 2012-10-30 15:40:39
.. 0x844afd40:chrome.exe                             3372   1068      6    138 2012-10-30 15:40:34
. 0x84510a98:vmtoolsd.exe                            2900   2784      5    220 2012-10-30 14:17:51
 0x85c2b530:wininit.exe                               388    340      3     75 2012-10-30 14:02:12
. 0x8600d030:lsm.exe                                  516    388     10    144 2012-10-30 14:02:14
. 0x85ff3b90:services.exe                             492    388      8    212 2012-10-30 14:02:14
.. 0x85c5b8f8:svchost.exe                            1156    492     16    482 2012-10-30 14:02:22
.. 0x86572bd8:taskhost.exe                           2144    492      7    191 2012-10-30 14:15:56
.. 0x844bf278:msiexec.exe                            3340    492     12    234 2012-10-30 15:39:37
.. 0x85d424f0:TPAutoConnSvc.                         1684    492     10    141 2012-10-30 14:02:27
... 0x864e0d40:TPAutoConnect.                        1180   1684      5    121 2012-10-30 14:15:49
```

c. **Psscan**: This plugin can find processes that were previously terminated or unlinked by a rootkit. This command lists processes running on a system, but it also has the ability to list hidden/unlinked processes. This command can be used in an investigation to discover hidden malicious software such as keyloggers or rootkits.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> psscan

   ii. Displays**:**

   - Offset
   - Name
   - PID
   - PPID
   - PDB
   - Time Created
   - Time exited



```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 psscan
Volatile Systems Volatility Framework 2.2
Offset(P)  Name                PID   PPID PDB        Time created         Time exited
---------- ---------------- ------ ------ ---------- -------------------- --------------------
0x3dc0ed40 msdtc.exe           344    492 0x3ed7b340 2012-10-30 14:15:46
0x3dc14920 svchost.exe        2460    492 0x3ed7b160 2012-10-30 14:17:41
0x3dc44918 calc.exe           3924   2784 0x3ed7b5e0 2012-10-30 15:41:03
0x3dcc7310 svchost.exe        2528    492 0x3ed7b440 2012-10-30 14:17:43
0x3dce0d40 TPAutoConnect.     1180   1684 0x3ed7b3c0 2012-10-30 14:15:49
0x3dce6030 sppsvc.exe         1532    492 0x3ed7b400 2012-10-30 14:15:49
0x3dd72bd8 taskhost.exe       2144    492 0x3ed7b420 2012-10-30 14:15:56
0x3e00c030 lsass.exe           508    388 0x3ed7b0e0 2012-10-30 14:02:14
0x3e00d030 lsm.exe             516    388 0x3ed7b100 2012-10-30 14:02:14
0x3e0581b0 svchost.exe         624    492 0x3ed7b120 2012-10-30 14:02:16
0x3e0b3c88 svchost.exe         784    492 0x3ed7b180 2012-10-30 14:02:17
0x3e0da750 svchost.exe         824    492 0x3ed7b1a0 2012-10-30 14:02:18
0x3e0ff220 svchost.exe         864    492 0x3ed7b1c0 2012-10-30 14:02:18
0x3e144030 WINZIP32.EXE        276    544 0x3ed7b520 2012-10-30 15:01:40  2012-10-30 15:26:36
0x3e1566b8 svchost.exe        1076    492 0x3ed7b220 2012-10-30 14:02:20
```

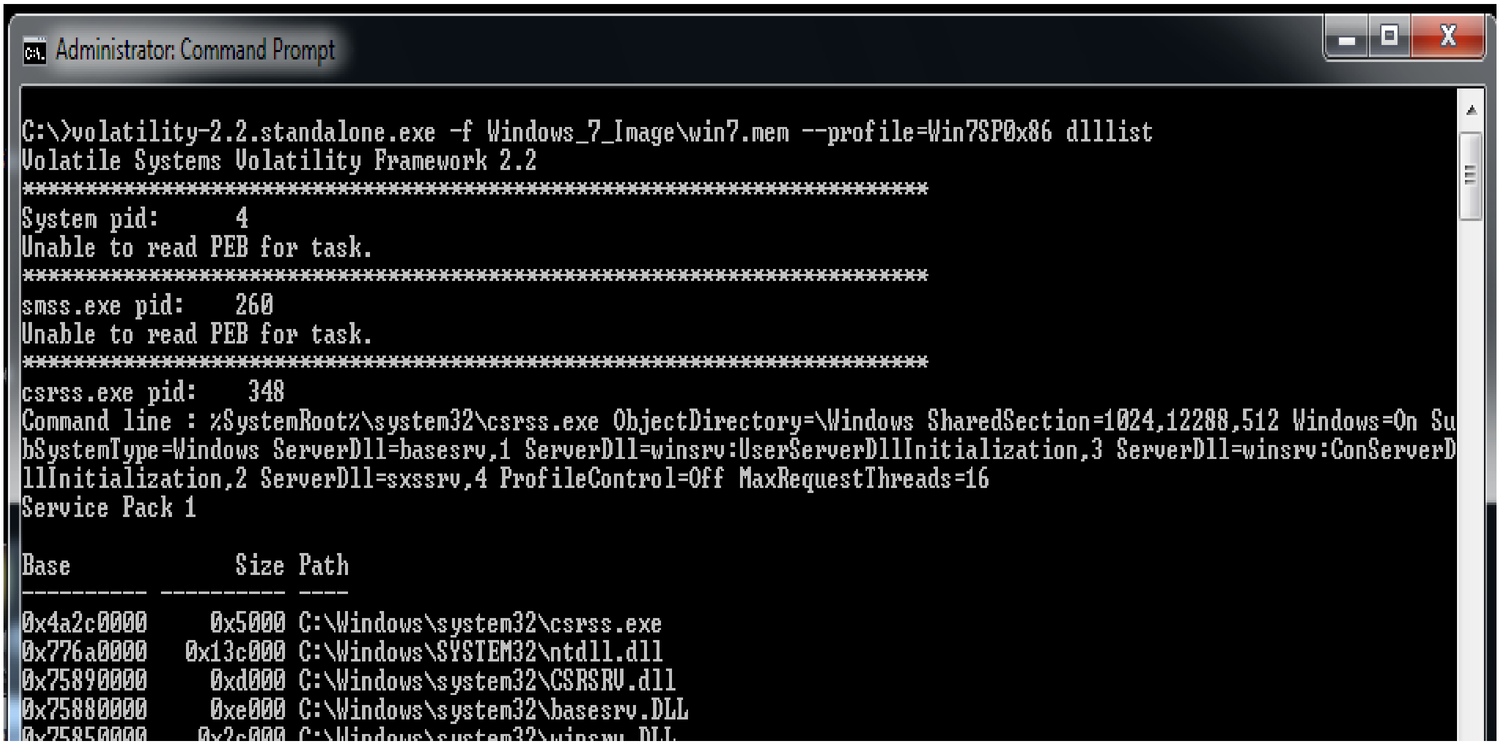
d. **Dllist**: **Dllist** displays a process's loaded DLLs. You can use the -p or -pid switch to filter. This command will display every DLL that a process calls and can be useful in an investigation by discovering if a process is calling DLLs that it should not be calling. For example, malware that is hiding as a system process and calling non-system DLLs.

i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> dllist

1. Filter using –p or –pid

ii. Displays:

- Base
- Size
- Path



```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 dlllist
Volatile Systems Volatility Framework 2.2
************************************************************************
System pid:      4
Unable to read PEB for task.
************************************************************************
smss.exe pid:    260
Unable to read PEB for task.
************************************************************************
csrss.exe pid:    348
Command line : %SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,12288,512 Windows=On Su
bSystemType=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=winsrv:ConServerD
llInitialization,2 ServerDll=sxssrv,4 ProfileControl=Off MaxRequestThreads=16
Service Pack 1

Base             Size Path
---------- ---------- ----
0x4a2c0000     0x5000 C:\Windows\system32\csrss.exe
0x776a0000   0x13c000 C:\Windows\SYSTEM32\ntdll.dll
0x75890000     0xd000 C:\Windows\system32\CSRSRV.dll
0x75880000     0xe000 C:\Windows\system32\basesrv.DLL
0x75850000    0x2c000 C:\Windows\system32\winsrv.DLL
```

e. **Dlldump**: Dlldump dumps the DLL to disk. This command will extract a specified DLL from the memory image, and the DLL can then be investigated further using other programs.

i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> dlldump

1. No Arguments: dumps all DLLs from all processes
2. -pid=<PID>: Dumps all DLLs from a specific process
3. --offset=<OFFSET>: all DLLs from a hidden/unlinked process
4. --base=<BASEADDR>: Dump a PE from anywhere in process memory
5. --regex=<REGEX>: Dump DLLs that match a regular expression--dump-dir=<DIR> or –d <DIR>: specify output directory

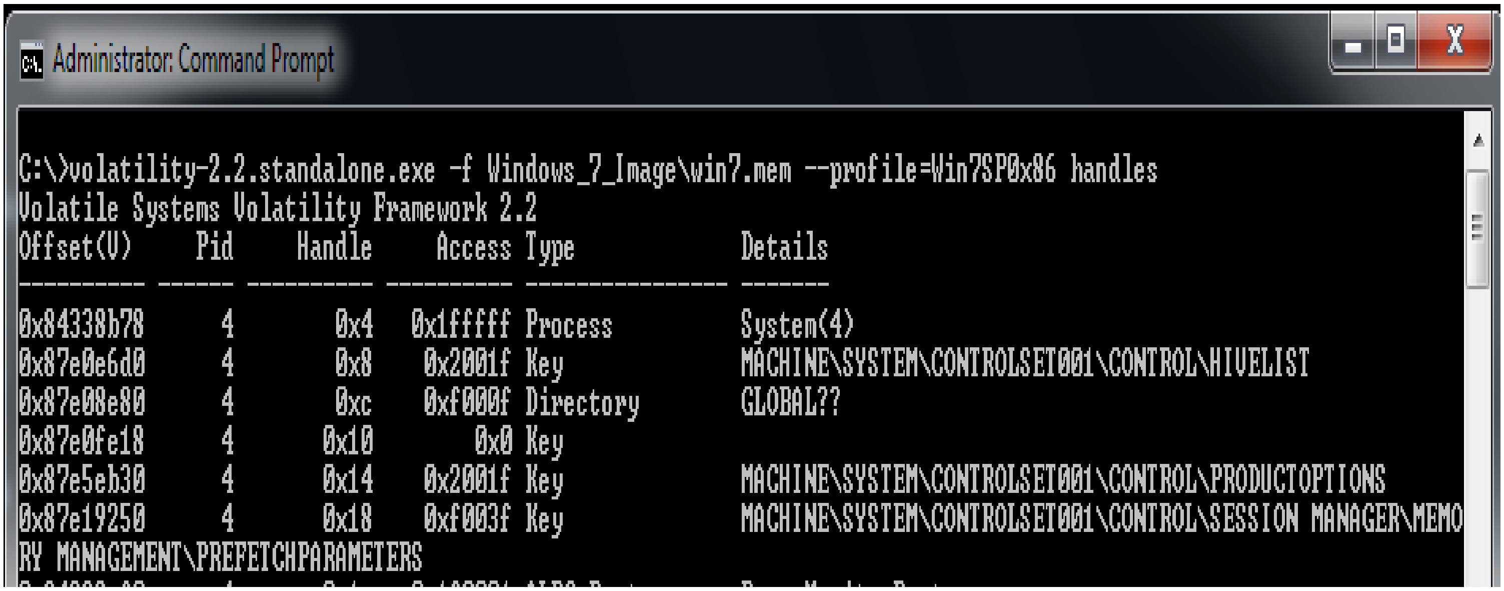
f. **Handles**: This plugin displays the open handles in a process.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> handles

      1. --pid=<PID>: filter by PID
      2. --physical-offset=<OFFSET>: filter by physical offset
      3. -t <OBJECTTYPE>: filter by object type
      4. --object-type=<OBJECTTYPE>: filter by object type

   ii. Displays:

   - Offset
   - PID
   - Handle
   - Access
   - Object Type



Administrator: Command Prompt

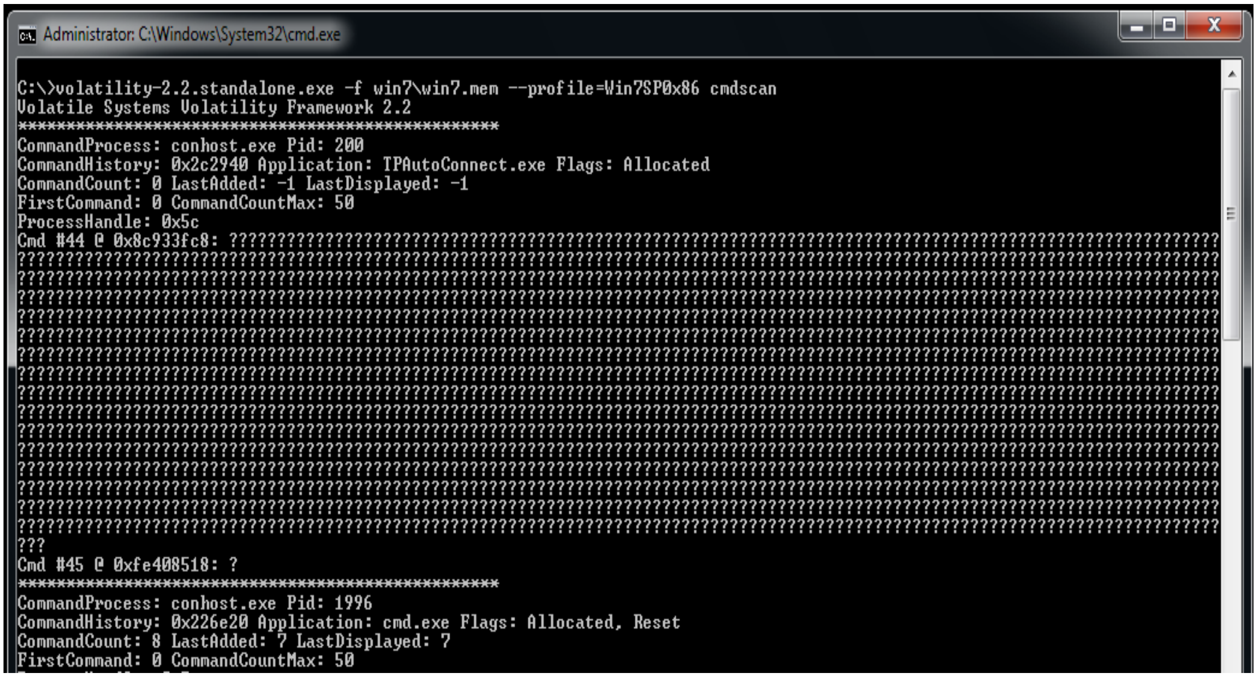
```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 handles
Volatile Systems Volatility Framework 2.2
Offset(V)     Pid     Handle     Access Type             Details
---------- ------ ---------- ---------- ---------------- -------
0x84338b78      4        0x4   0x1fffff Process          System(4)
0x87e0e6d0      4        0x8    0x2001f Key              MACHINE\SYSTEM\CONTROLSET001\CONTROL\HIVELIST
0x87e08e80      4        0xc    0xf000f Directory        GLOBAL??
0x87e0fe18      4       0x10        0x0 Key
0x87e5eb30      4       0x14    0x2001f Key              MACHINE\SYSTEM\CONTROLSET001\CONTROL\PRODUCTOPTIONS
0x87e19250      4       0x18    0xf003f Key              MACHINE\SYSTEM\CONTROLSET001\CONTROL\SESSION MANAGER\MEMO
RY MANAGEMENT\PREFETCHPARAMETERS
```

g. **Cmdscan**: This plugin shows every command entered through a console shell. This can be useful to an investigation in that it will show commands that a user entered into command prompt or those that an intruder executed remotely.

   i. Usage: volatility-2.2.standalone.exe –f<path to image> --profile=<profile> cmdscan

   ii. Displays:

   - The name of the console host process
   - Application using the console
   - Location of command history buffs, current buffer count, last added command and last displayed command
   - Process Handle



```
C:\>volatility-2.2.standalone.exe -f win7\win7.mem --profile=Win7SP0x86 cmdscan
Volatile Systems Volatility Framework 2.2
**************************************************
CommandProcess: conhost.exe Pid: 200
CommandHistory: 0x2c2940 Application: TPAutoConnect.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
Cmd #44 @ 0x8c933fc8: ?????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
???
Cmd #45 @ 0xfe408518: ?
**************************************************
CommandProcess: conhost.exe Pid: 1996
CommandHistory: 0x226e20 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 8 LastAdded: 7 LastDisplayed: 7
FirstCommand: 0 CommandCountMax: 50
```

Memory and Kernel Objects

Plugins relating to this section extract slack space, display kernel drivers, and provide a list of open files on the system.

a. **Procmemdump**: This plugin dumps a process to an executable memory sample. This command will extract a process, including slack space, from a memory image. This would allow you to then investigate the suspect process further using other tools.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> procmemdump –D <output location> -p <PID>

      1. --unsafe or -u to by bypass sanity checks

b. **Procexedump**: This plugin dumps a process to an executable file sample. This command will extract a process from a memory image and would allow you to then investigate the suspect process further using other tools.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> procmemdump –D <output location> -p <PID>

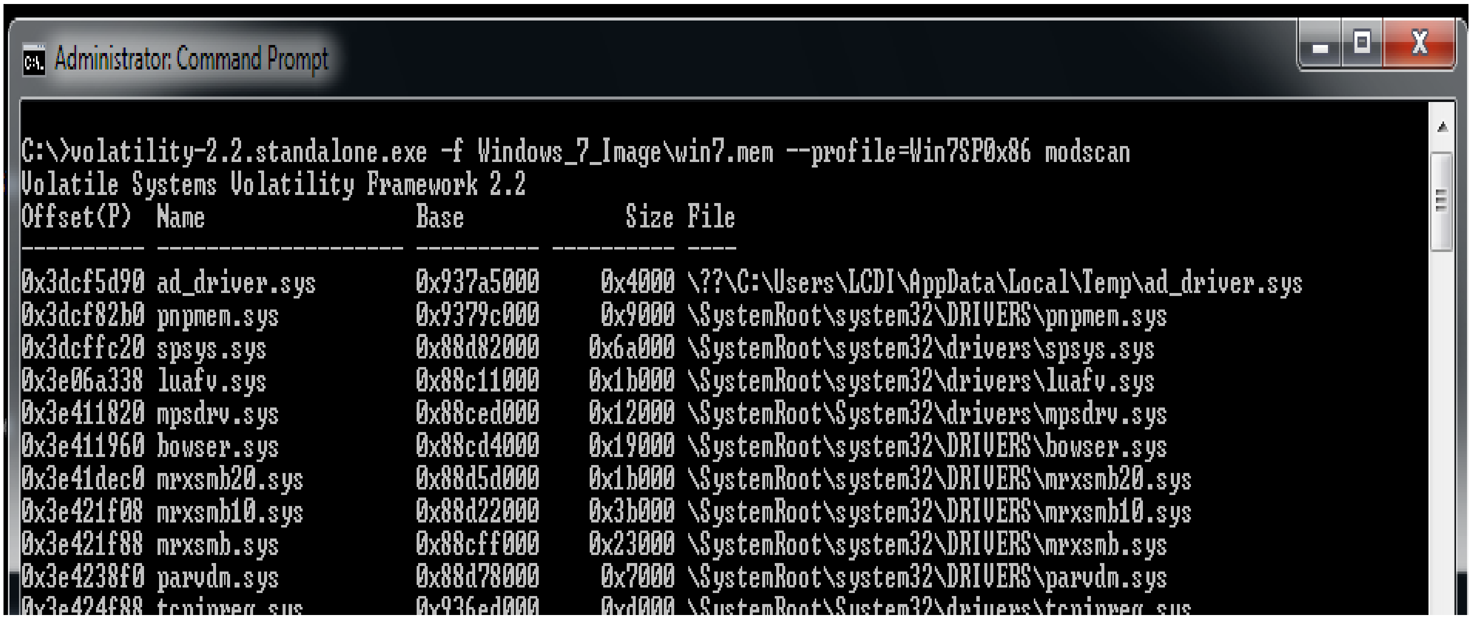
      1. --unsafe or -u to by bypass sanity checks

c. **Modscan**: Modscan scans physical memory for _LDR_DATA_TABLE_ENTRY objects. This command will display kernel drivers, including ones that have been hidden/unlinked.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> modscan

   ii. Display:

   - Offset (By default Virtual Offset, -P for Physical)
   - Name
   - Base
   - Size
   - File



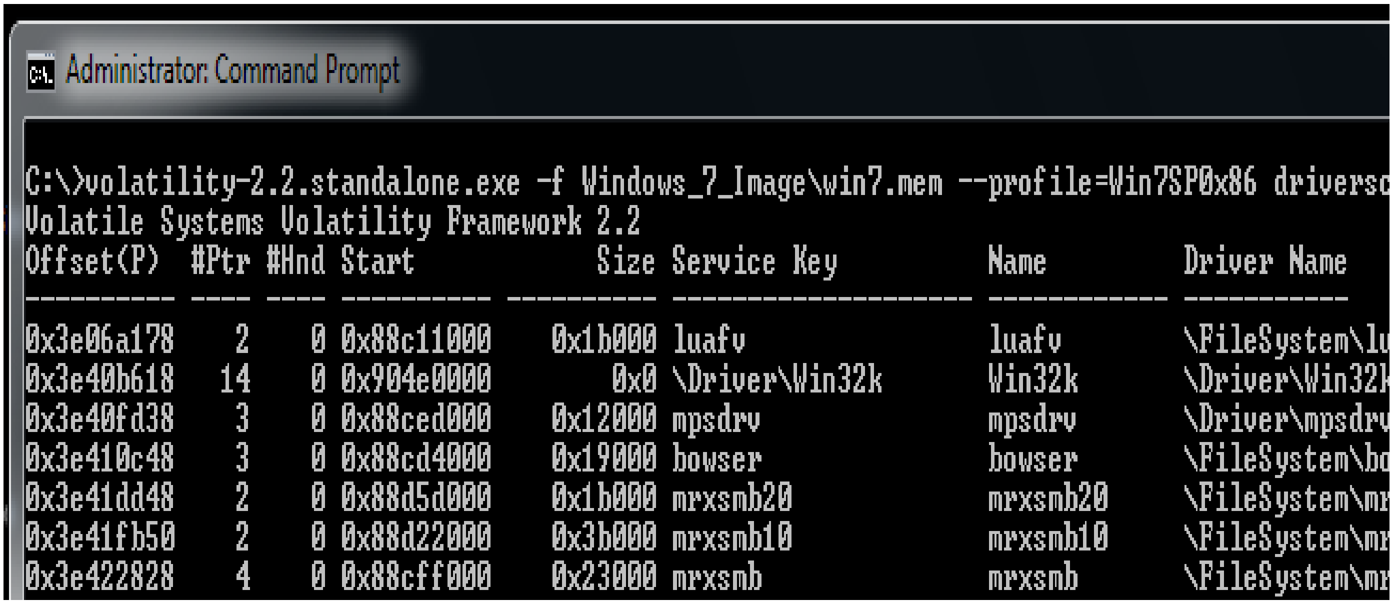
```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 modscan
Volatile Systems Volatility Framework 2.2
Offset(P)  Name                 Base             Size File
---------- -------------------- ---------- ---------- ----
0x3dcf5d90 ad_driver.sys        0x937a5000     0x4000 \??\C:\Users\LCDI\AppData\Local\Temp\ad_driver.sys
0x3dcf82b0 pnpmem.sys           0x9379c000     0x9000 \SystemRoot\system32\DRIVERS\pnpmem.sys
0x3dcffc20 spsys.sys            0x88d82000    0x6a000 \SystemRoot\system32\drivers\spsys.sys
0x3e06a338 luafv.sys            0x88c11000    0x1b000 \SystemRoot\system32\drivers\luafv.sys
0x3e411820 mpsdrv.sys           0x88ced000    0x12000 \SystemRoot\System32\drivers\mpsdrv.sys
0x3e411960 bowser.sys           0x88cd4000    0x19000 \SystemRoot\system32\DRIVERS\bowser.sys
0x3e41dec0 mrxsmb20.sys         0x88d5d000    0x1b000 \SystemRoot\system32\DRIVERS\mrxsmb20.sys
0x3e421f08 mrxsmb10.sys         0x88d22000    0x3b000 \SystemRoot\system32\DRIVERS\mrxsmb10.sys
0x3e421f88 mrxsmb.sys           0x88cff000    0x23000 \SystemRoot\system32\DRIVERS\mrxsmb.sys
0x3e4238f0 parvdm.sys           0x88d78000     0x7000 \SystemRoot\system32\DRIVERS\parvdm.sys
0x3e424f88 tcpipreg.sys         0x936ed000     0xd000 \SystemRoot\System32\drivers\tcpipreg.sys
```

d. **Driverscan**: Driverscan scans for driver objects in _DRIVER_OBJECT. This command will list kernel module driver objects.

i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> driverscan

ii. Displays:

- Offset
- Pointers
- Handles
- Start
- Size
- Service Key
- Name
- Driver Name



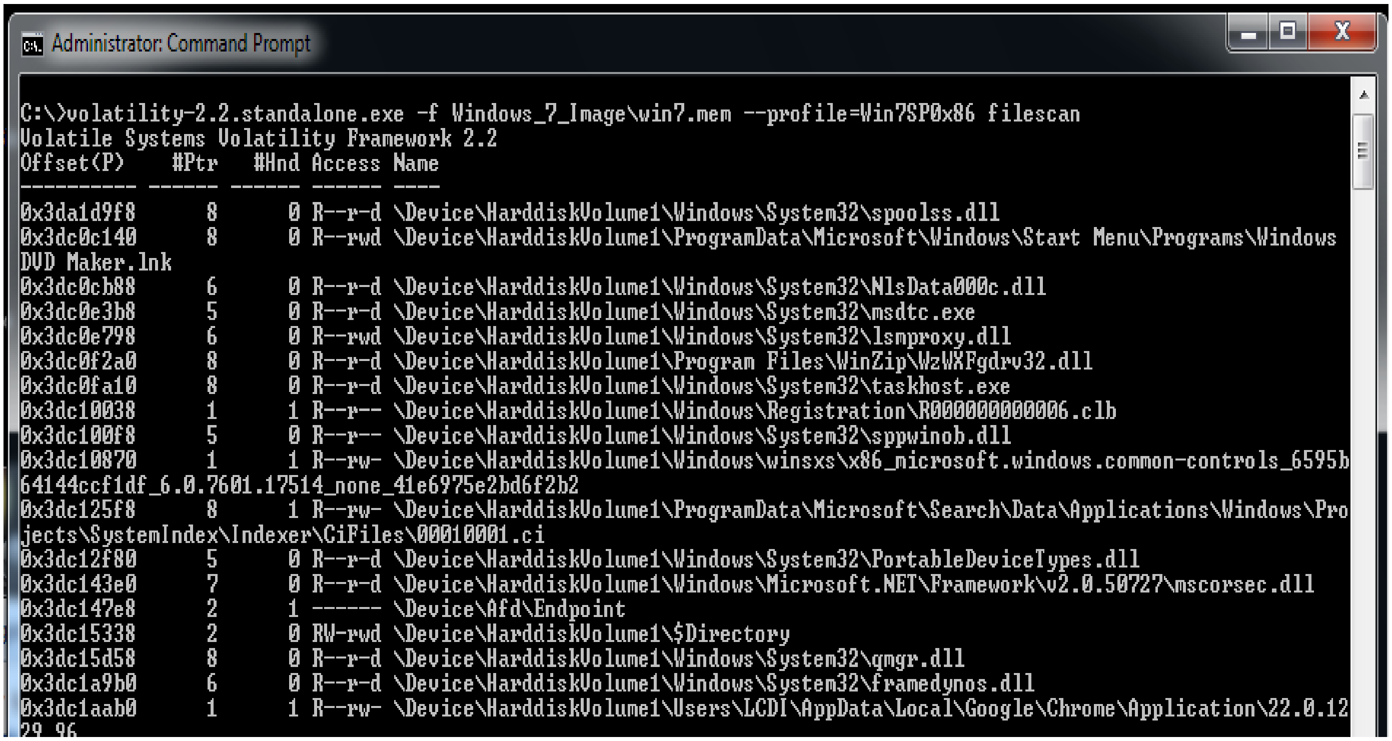
```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 driversc
Volatile Systems Volatility Framework 2.2
Offset(P)  #Ptr #Hnd Start            Size Service Key          Name         Driver Name
---------- ---- ---- ---------- ---------- -------------------- ------------ ------------
0x3e06a178    2    0 0x88c11000    0x1b000 luafv                luafv        \FileSystem\lu
0x3e40b618   14    0 0x904e0000        0x0 \Driver\Win32k       Win32k       \Driver\Win32k
0x3e40fd38    3    0 0x88ced000    0x12000 mpsdrv               mpsdrv       \Driver\mpsdrv
0x3e410c48    3    0 0x88cd4000    0x19000 bowser               bowser       \FileSystem\bo
0x3e41dd48    2    0 0x88d5d000    0x1b000 mrxsmb20             mrxsmb20     \FileSystem\mr
0x3e41fb50    2    0 0x88d22000    0x3b000 mrxsmb10             mrxsmb10     \FileSystem\mr
0x3e422828    4    0 0x88cff000    0x23000 mrxsmb               mrxsmb       \FileSystem\mr
```

e. **File scan**: File scan locates files from FILE_OBJECT in the physical memory. This command will display open files on the system, including files that have been hidden by malicious software.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> filescan

   ii. Displays:

   - Physical offset
   - File name
   - Points
   - Handles
   - Permissions



```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 filescan
Volatile Systems Volatility Framework 2.2
Offset(P)     #Ptr   #Hnd Access Name
---------- ------ ------ ------ ----
0x3da1d9f8      8      0 R--r-d \Device\HarddiskVolume1\Windows\System32\spoolss.dll
0x3dc0c140      8      0 R--rwd \Device\HarddiskVolume1\ProgramData\Microsoft\Windows\Start Menu\Programs\Windows
DVD Maker.lnk
0x3dc0cb88      6      0 R--r-d \Device\HarddiskVolume1\Windows\System32\NlsData000c.dll
0x3dc0e3b8      5      0 R--r-d \Device\HarddiskVolume1\Windows\System32\msdtc.exe
0x3dc0e798      6      0 R--rwd \Device\HarddiskVolume1\Windows\System32\lsmproxy.dll
0x3dc0f2a0      8      0 R--r-d \Device\HarddiskVolume1\Program Files\WinZip\WzWXFgdrv32.dll
0x3dc0fa10      8      0 R--r-d \Device\HarddiskVolume1\Windows\System32\taskhost.exe
0x3dc10038      1      1 R--r-- \Device\HarddiskVolume1\Windows\Registration\R000000000006.clb
0x3dc100f8      5      0 R--r-- \Device\HarddiskVolume1\Windows\System32\sppwinob.dll
0x3dc10870      1      1 R--rw- \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-controls_6595b
64144ccf1df_6.0.7601.17514_none_41e6975e2bd6f2b2
0x3dc125f8      8      1 R--rw- \Device\HarddiskVolume1\ProgramData\Microsoft\Search\Data\Applications\Windows\Pro
jects\SystemIndex\Indexer\CiFiles\00010001.ci
0x3dc12f80      5      0 R--r-d \Device\HarddiskVolume1\Windows\System32\PortableDeviceTypes.dll
0x3dc143e0      7      0 R--r-d \Device\HarddiskVolume1\Windows\Microsoft.NET\Framework\v2.0.50727\mscorsec.dll
0x3dc147e8      2      1 ------ \Device\Afd\Endpoint
0x3dc15338      2      0 RW-rwd \Device\HarddiskVolume1\$Directory
0x3dc15d58      8      0 R--r-d \Device\HarddiskVolume1\Windows\System32\qmgr.dll
0x3dc1a9b0      6      0 R--r-d \Device\HarddiskVolume1\Windows\System32\framedynos.dll
0x3dc1aab0      1      1 R--rw- \Device\HarddiskVolume1\Users\LCDI\AppData\Local\Google\Chrome\Application\22.0.12
29.96
```

Networking

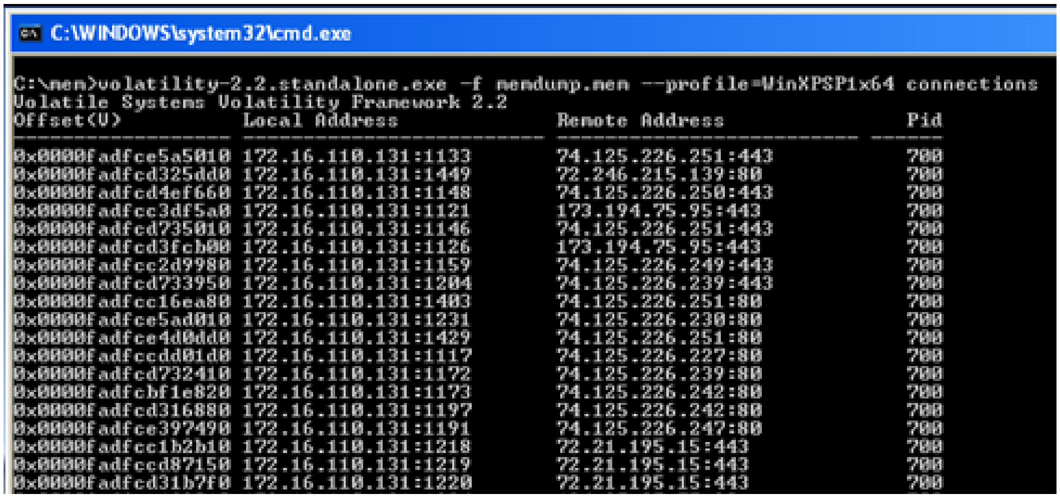
Plugins relating to this section identify open connections and sockets.

a. **Connections**: (x86 and x64 XP and 2003 Server) This plugin prints a list of open connections and will list active network connections. It would be useful in investigations to determine where traffic was coming from or going to and which application was generating it.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> connections

   ii. Displays:

   - Offset (Virtual by default, -P for physical)
   - local address
   - remote address
   - PID



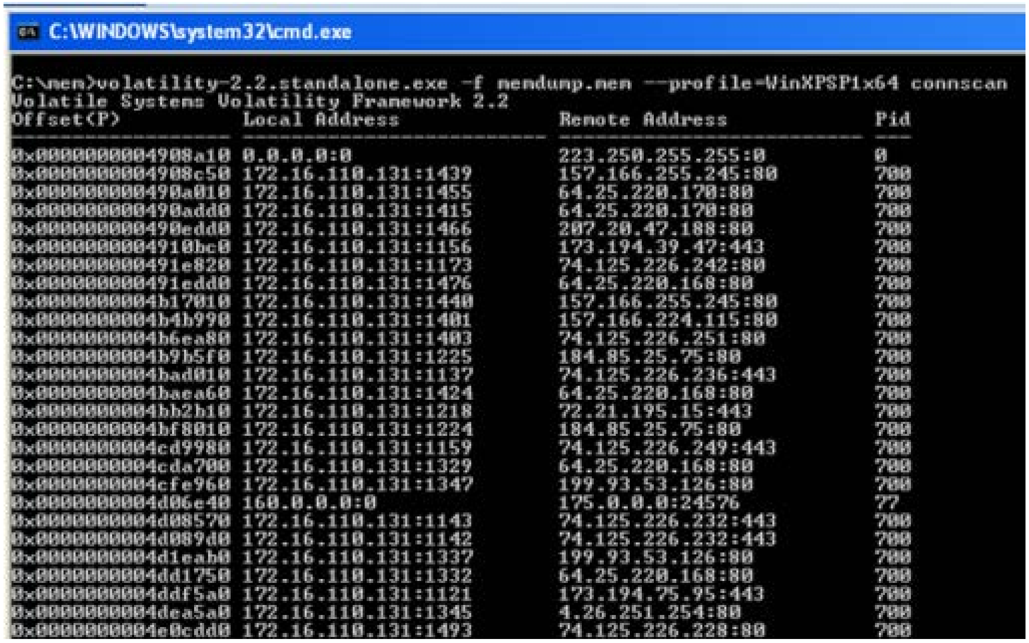
```
C:\mem>volatility-2.2.standalone.exe -f memdump.mem --profile=WinXPSP1x64 connections
Volatile Systems Volatility Framework 2.2
Offset(V)          Local Address             Remote Address            Pid
------------------ ------------------------- ------------------------- ------
0x0000fadfce5a5010 172.16.110.131:1133       74.125.226.251:443        700
0x0000fadfcd325dd0 172.16.110.131:1449       72.246.215.139:80         700
0x0000fadfcd4ef660 172.16.110.131:1148       74.125.226.250:443        700
0x0000fadfcc3df5a0 172.16.110.131:1121       173.194.75.95:443         700
0x0000fadfcd735010 172.16.110.131:1146       74.125.226.251:443        700
0x0000fadfcd3fcb00 172.16.110.131:1126       173.194.75.95:443         700
0x0000fadfcc2d9980 172.16.110.131:1159       74.125.226.249:443        700
0x0000fadfcd733950 172.16.110.131:1204       74.125.226.239:443        700
0x0000fadfcc16ea80 172.16.110.131:1403       74.125.226.251:80         700
0x0000fadfce5ad010 172.16.110.131:1231       74.125.226.230:80         700
0x0000fadfce4d0dd0 172.16.110.131:1429       74.125.226.251:80         700
0x0000fadfccdd01d0 172.16.110.131:1117       74.125.226.227:80         700
0x0000fadfcd732410 172.16.110.131:1172       74.125.226.239:80         700
0x0000fadfcbf1e820 172.16.110.131:1173       74.125.226.242:80         700
0x0000fadfcd316880 172.16.110.131:1197       74.125.226.242:80         700
0x0000fadfce397490 172.16.110.131:1191       74.125.226.247:80         700
0x0000fadfcc1b2b10 172.16.110.131:1218       72.21.195.15:443          700
0x0000fadfccd87150 172.16.110.131:1219       72.21.195.15:443          700
0x0000fadfcd31b7f0 172.16.110.131:1220       72.21.195.15:443          700
```

b. **Connscan**: (x86 and x64 XP and 2003 Server) Connscan is similar to connections, but this plugin can find artifacts from previous connections. This command will list active network connections, including connections that have been terminated. It would be useful in investigations to determine where traffic was coming from or going to and which application was generating it.

i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> connscan

ii. Displays:

- Offset
- Local address
- Remote Address
- PID



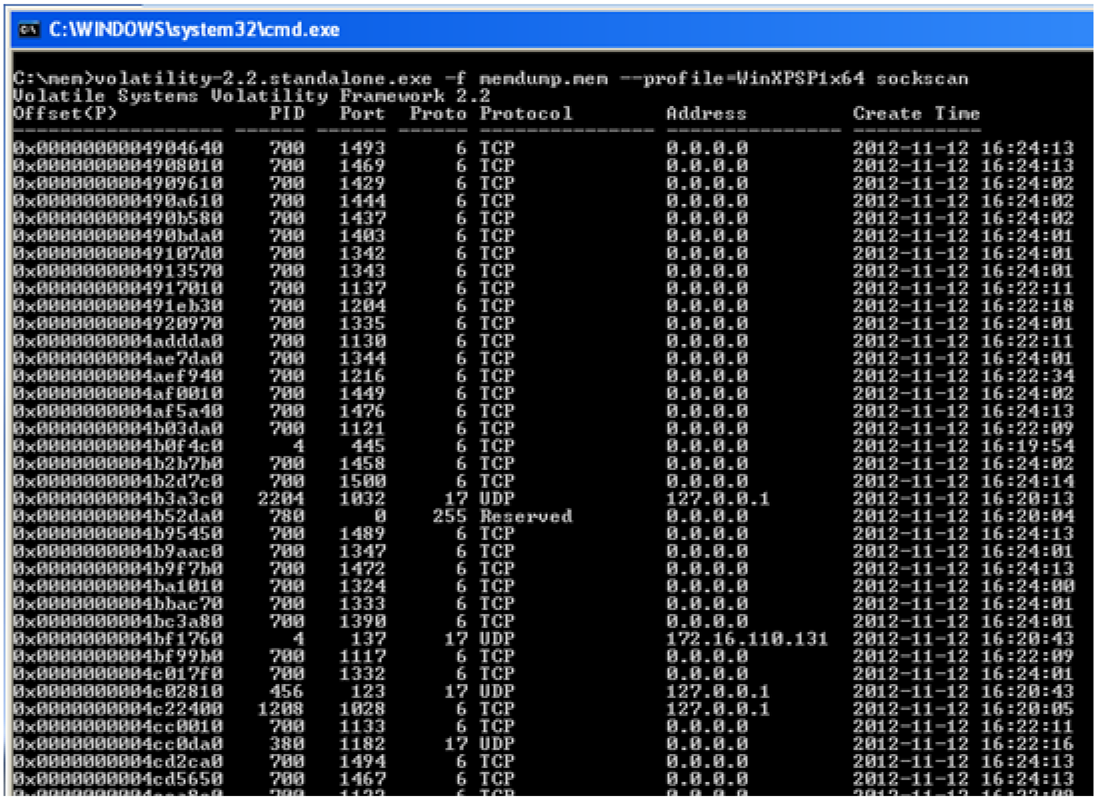
```
C:\mem>volatility-2.2.standalone.exe -f memdump.mem --profile=WinXPSP1x64 connscan
Volatile Systems Volatility Framework 2.2
Offset(P)          Local Address             Remote Address            Pid
------------------ ------------------------- ------------------------- ---
0x0000000004908a10 0.0.0.0:0                 223.250.255.255:0         0
0x0000000004908c50 172.16.110.131:1439       157.166.255.245:80        700
0x000000000490a010 172.16.110.131:1455       64.25.220.170:80          700
0x000000000490add0 172.16.110.131:1415       64.25.220.170:80          700
0x000000000490edd0 172.16.110.131:1466       207.20.47.188:80          700
0x0000000004910bc0 172.16.110.131:1156       173.194.39.47:443         700
0x000000000491e820 172.16.110.131:1173       74.125.226.242:80         700
0x000000000491edd0 172.16.110.131:1476       64.25.220.168:80          700
0x0000000004b17010 172.16.110.131:1440       157.166.255.245:80        700
0x0000000004b4b990 172.16.110.131:1401       157.166.224.115:80        700
0x0000000004b6ea80 172.16.110.131:1403       74.125.226.251:80         700
0x0000000004b9b5f0 172.16.110.131:1225       184.85.25.75:80           700
0x0000000004bad010 172.16.110.131:1137       74.125.226.236:443        700
0x0000000004baea60 172.16.110.131:1424       64.25.220.168:80          700
0x0000000004bb2b10 172.16.110.131:1218       72.21.195.15:443          700
0x0000000004bf8010 172.16.110.131:1224       184.85.25.75:80           700
0x0000000004cd9980 172.16.110.131:1159       74.125.226.249:443        700
0x0000000004cda700 172.16.110.131:1329       64.25.220.168:80          700
0x0000000004cfe960 172.16.110.131:1347       199.93.53.126:80          700
0x0000000004d06e40 160.0.0.0:0               175.0.0.0:24576           77
0x0000000004d08570 172.16.110.131:1143       74.125.226.232:443        700
0x0000000004d089d0 172.16.110.131:1142       74.125.226.232:443        700
0x0000000004d1eab0 172.16.110.131:1337       199.93.53.126:80          700
0x0000000004dd1750 172.16.110.131:1332       64.25.220.168:80          700
0x0000000004ddf5a0 172.16.110.131:1121       173.194.75.95:443         700
0x0000000004dea5a0 172.16.110.131:1345       4.26.251.254:80           700
0x0000000004e0cdd0 172.16.110.131:1493       74.125.226.228:80         700
```

c. **Sockscan**: (x86 and x64 XP and 2003 Server) Sockscan scans physical memory for _ADDRESS_OBJECT objects (TCP sockets). This command will display a list of sockets on the system and can find previous sockets. This command would be useful in an investigation by allowing you to see which processes are listening for network connections on which protocol.

i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> sockscan

ii. Displays:

- Offset
- PID
- Port
- Proto
- Protocol
- Address
- Create Time



C:\WINDOWS\system32\cmd.exe

```
C:\mem>volatility-2.2.standalone.exe -f memdump.mem --profile=WinXPSP1x64 sockscan
Volatile Systems Volatility Framework 2.2
Offset(P)             PID   Port  Proto Protocol         Address          Create Time
------------------ ------ ------ ------ ---------------- ---------------- --------------
0x0000000004904640    700   1493      6 TCP              0.0.0.0          2012-11-12 16:24:13
0x0000000004908010    700   1469      6 TCP              0.0.0.0          2012-11-12 16:24:13
0x0000000004909610    700   1429      6 TCP              0.0.0.0          2012-11-12 16:24:02
0x000000000490a610    700   1444      6 TCP              0.0.0.0          2012-11-12 16:24:02
0x000000000490b580    700   1437      6 TCP              0.0.0.0          2012-11-12 16:24:02
0x000000000490bda0    700   1403      6 TCP              0.0.0.0          2012-11-12 16:24:01
0x00000000049107d0    700   1342      6 TCP              0.0.0.0          2012-11-12 16:24:01
0x0000000004913570    700   1343      6 TCP              0.0.0.0          2012-11-12 16:24:01
0x0000000004917010    700   1137      6 TCP              0.0.0.0          2012-11-12 16:22:11
0x000000000491eb30    700   1204      6 TCP              0.0.0.0          2012-11-12 16:22:18
0x0000000004920970    700   1335      6 TCP              0.0.0.0          2012-11-12 16:24:01
0x0000000004addda0    700   1130      6 TCP              0.0.0.0          2012-11-12 16:22:11
0x0000000004ae7da0    700   1344      6 TCP              0.0.0.0          2012-11-12 16:24:01
0x0000000004aef940    700   1216      6 TCP              0.0.0.0          2012-11-12 16:22:34
0x0000000004af0010    700   1449      6 TCP              0.0.0.0          2012-11-12 16:24:02
0x0000000004af5a40    700   1476      6 TCP              0.0.0.0          2012-11-12 16:24:13
0x0000000004b03da0    700   1121      6 TCP              0.0.0.0          2012-11-12 16:22:09
0x0000000004b0f4c0      4    445      6 TCP              0.0.0.0          2012-11-12 16:19:54
0x0000000004b2b7b0    700   1458      6 TCP              0.0.0.0          2012-11-12 16:24:02
0x0000000004b2d7c0    700   1500      6 TCP              0.0.0.0          2012-11-12 16:24:14
0x0000000004b3a3c0   2204   1032     17 UDP              127.0.0.1        2012-11-12 16:20:13
0x0000000004b52da0    780      0    255 Reserved         0.0.0.0          2012-11-12 16:20:04
0x0000000004b95450    700   1489      6 TCP              0.0.0.0          2012-11-12 16:24:13
0x0000000004b9aac0    700   1347      6 TCP              0.0.0.0          2012-11-12 16:24:01
0x0000000004b9f7b0    700   1472      6 TCP              0.0.0.0          2012-11-12 16:24:13
0x0000000004ba1010    700   1324      6 TCP              0.0.0.0          2012-11-12 16:24:00
0x0000000004bbac70    700   1333      6 TCP              0.0.0.0          2012-11-12 16:24:01
0x0000000004bc3a80    700   1390      6 TCP              0.0.0.0          2012-11-12 16:24:01
0x0000000004bf1760      4    137     17 UDP              172.16.110.131   2012-11-12 16:20:43
0x0000000004bf99b0    700   1117      6 TCP              0.0.0.0          2012-11-12 16:22:09
0x0000000004c017f0    700   1332      6 TCP              0.0.0.0          2012-11-12 16:24:01
0x0000000004c02810    456    123     17 UDP              127.0.0.1        2012-11-12 16:20:43
0x0000000004c22400   1208   1028      6 TCP              127.0.0.1        2012-11-12 16:20:05
0x0000000004cc0010    700   1133      6 TCP              0.0.0.0          2012-11-12 16:22:11
0x0000000004cc0da0    380   1182     17 UDP              0.0.0.0          2012-11-12 16:22:16
0x0000000004cd2ca0    700   1494      6 TCP              0.0.0.0          2012-11-12 16:24:13
0x0000000004cd5650    700   1467      6 TCP              0.0.0.0          2012-11-12 16:24:13
```

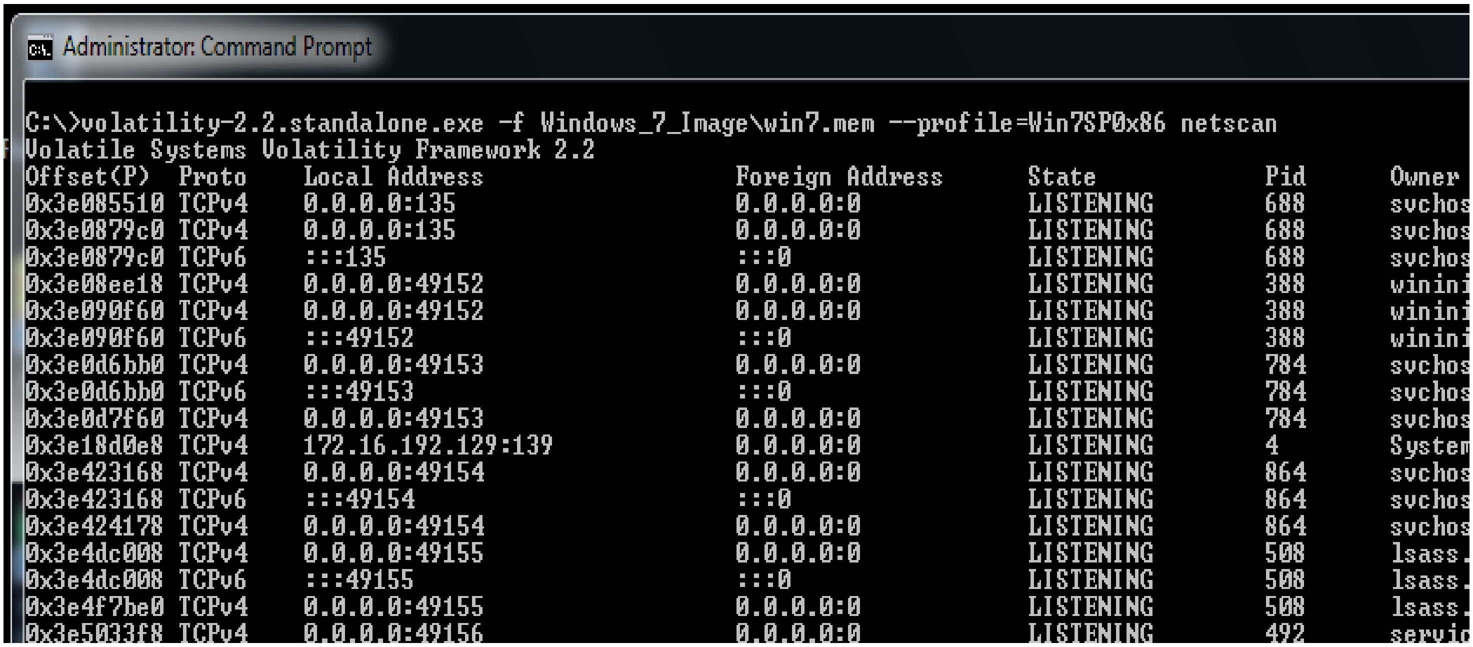
d. **Netscan**: (x86 and x64 Vista 2008 Server, Win7) Netscan finds TCP/UDP endpoints and listeners. This command will display a list of active network connections. This would be useful in investigations to determine where traffic was coming from or going to, over which protocol, and which application was generating it.

i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> netscan

ii. Displays:

- Offset
- Protocol
- Local Address

- Foreign Address
- State
- PID
- Owner
- Created



Administrator: Command Prompt

```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 netscan
Volatile Systems Volatility Framework 2.2
Offset(P)  Proto    Local Address                  Foreign Address      State            Pid      Owner
0x3e085510 TCPv4    0.0.0.0:135                    0.0.0.0:0            LISTENING        688      svchos
0x3e0879c0 TCPv4    0.0.0.0:135                    0.0.0.0:0            LISTENING        688      svchos
0x3e0879c0 TCPv6    :::135                         :::0                 LISTENING        688      svchos
0x3e08ee18 TCPv4    0.0.0.0:49152                  0.0.0.0:0            LISTENING        388      winini
0x3e090f60 TCPv4    0.0.0.0:49152                  0.0.0.0:0            LISTENING        388      winini
0x3e090f60 TCPv6    :::49152                       :::0                 LISTENING        388      winini
0x3e0d6bb0 TCPv4    0.0.0.0:49153                  0.0.0.0:0            LISTENING        784      svchos
0x3e0d6bb0 TCPv6    :::49153                       :::0                 LISTENING        784      svchos
0x3e0d7f60 TCPv4    0.0.0.0:49153                  0.0.0.0:0            LISTENING        784      svchos
0x3e18d0e8 TCPv4    172.16.192.129:139             0.0.0.0:0            LISTENING        4        System
0x3e423168 TCPv4    0.0.0.0:49154                  0.0.0.0:0            LISTENING        864      svchos
0x3e423168 TCPv6    :::49154                       :::0                 LISTENING        864      svchos
0x3e424178 TCPv4    0.0.0.0:49154                  0.0.0.0:0            LISTENING        864      svchos
0x3e4dc008 TCPv4    0.0.0.0:49155                  0.0.0.0:0            LISTENING        508      lsass.
0x3e4dc008 TCPv6    :::49155                       :::0                 LISTENING        508      lsass.
0x3e4f7be0 TCPv4    0.0.0.0:49155                  0.0.0.0:0            LISTENING        508      lsass.
0x3e5033f8 TCPv4    0.0.0.0:49156                  0.0.0.0:0            LISTENING        492      servic
```

Registry

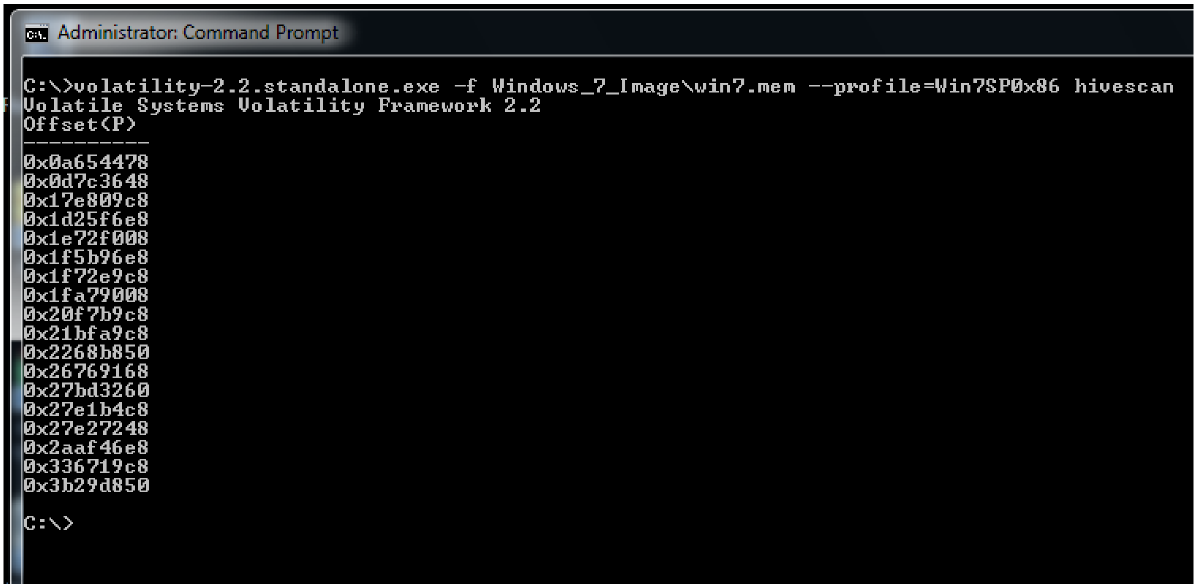
Plugins relating to this section print a list of registry hives and can dump password hashes from the memory image.

a. **Hivescan and Hivelist**: Both of these plugins find the physical addresses of registry hives and print the list of them. Hivelist gives the virtual offset and file system path, but these plugins essentially do the same thing. These commands would be useful in an investigation as the offset can be used to extract registry hives or for further analysis using other commands.

    i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> **hivescan** or **hivelist**

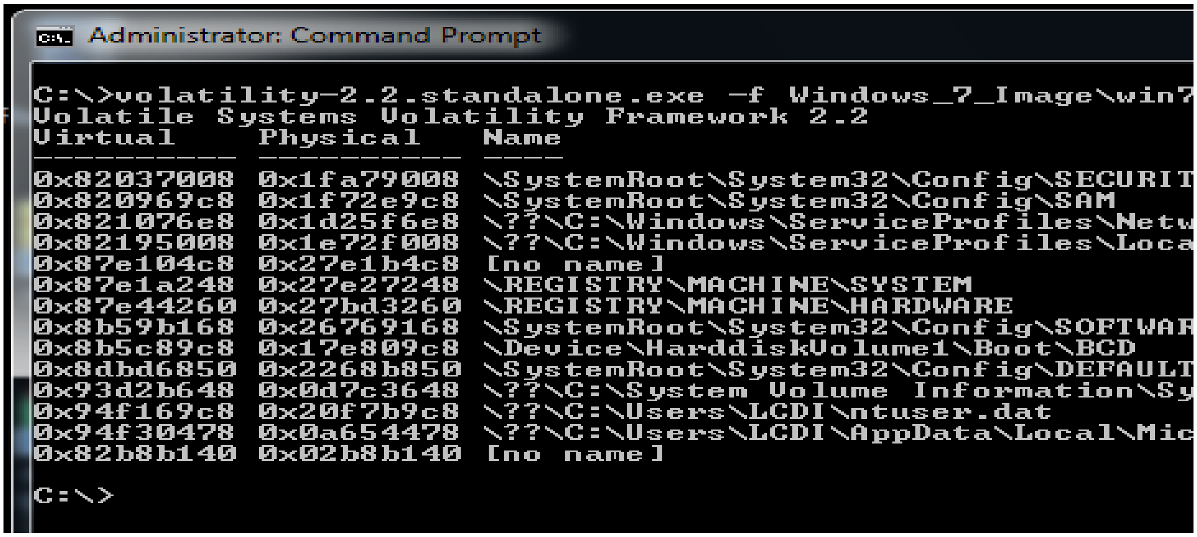
    ii. Displays:

    - Virtual/Physical Offset
    - Name



Administrator: Command Prompt

```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 hivescan
Volatile Systems Volatility Framework 2.2
Offset(P)
----------
0x0a654478
0x0d7c3648
0x17e809c8
0x1d25f6e8
0x1e72f008
0x1f5b96e8
0x1f72e9c8
0x1fa79008
0x20f7b9c8
0x21bfa9c8
0x2268b850
0x26769168
0x27bd3260
0x27e1b4c8
0x27e27248
0x2aaf46e8
0x336719c8
0x3b29d850

C:\>
```



Administrator: Command Prompt

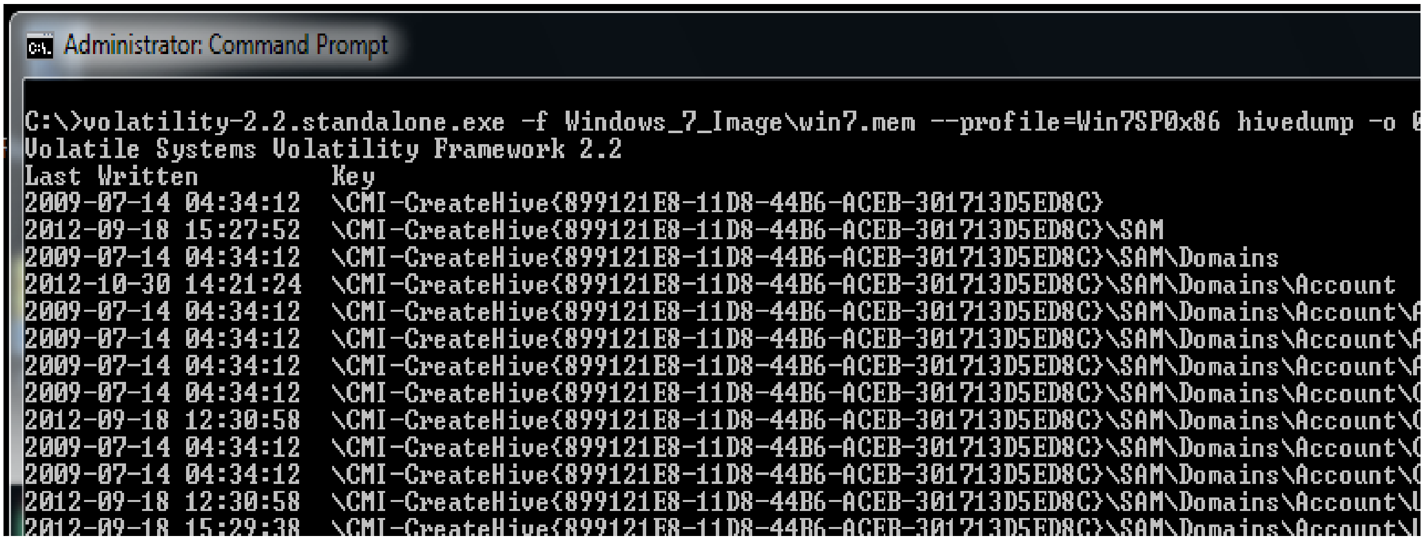
```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7
Volatile Systems Volatility Framework 2.2
Virtual     Physical     Name
----------  ----------   ----
0x82037008  0x1fa79008  \SystemRoot\System32\Config\SECURIT
0x820969c8  0x1f72e9c8  \SystemRoot\System32\Config\SAM
0x821076e8  0x1d25f6e8  \??\C:\Windows\ServiceProfiles\Netw
0x82195008  0x1e72f008  \??\C:\Windows\ServiceProfiles\Loca
0x87e104c8  0x27e1b4c8  [no name]
0x87e1a248  0x27e27248  \REGISTRY\MACHINE\SYSTEM
0x87e44260  0x27bd3260  \REGISTRY\MACHINE\HARDWARE
0x8b59b168  0x26769168  \SystemRoot\System32\Config\SOFTWAR
0x8b5c89c8  0x17e809c8  \Device\HarddiskVolume1\Boot\BCD
0x8dbd6850  0x2268b850  \SystemRoot\System32\Config\DEFAULT
0x93d2b648  0x0d7c3648  \??\C:\System Volume Information\Sy
0x94f169c8  0x20f7b9c8  \??\C:\Users\LCDI\ntuser.dat
0x94f30478  0x0a654478  \??\C:\Users\LCDI\AppData\Local\Mic
0x82b8b140  0x02b8b140  [no name]

C:\>
```

b. **Hivedump**: This plugin prints out a hive. This command displays all of the subkeys contained in a registry hive, as well as the last written time. This is useful as the presence of certain subkeys could be of evidentiary value, and the last written key can also show that a key was recently updated.

i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> hivedump –o <virtual address>

ii. Displays:

- Last Written time
- Key



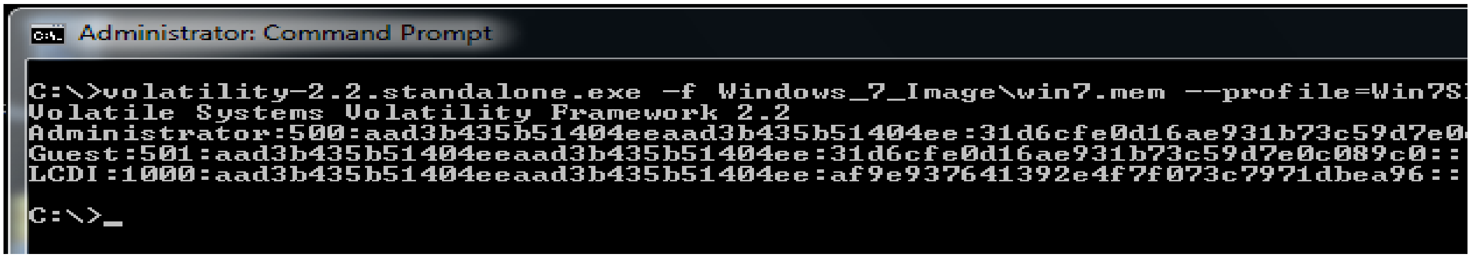
```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 hivedump -o 0
Volatile Systems Volatility Framework 2.2
Last Written         Key
2009-07-14 04:34:12  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}
2012-09-18 15:27:52  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM
2009-07-14 04:34:12  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM\Domains
2012-10-30 14:21:24  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM\Domains\Account
2009-07-14 04:34:12  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM\Domains\Account\A
2009-07-14 04:34:12  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM\Domains\Account\A
2009-07-14 04:34:12  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM\Domains\Account\A
2009-07-14 04:34:12  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM\Domains\Account\G
2012-09-18 12:30:58  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM\Domains\Account\G
2009-07-14 04:34:12  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM\Domains\Account\G
2009-07-14 04:34:12  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM\Domains\Account\G
2012-09-18 12:30:58  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM\Domains\Account\U
2012-09-18 15:29:38  \CMI-CreateHive{899121E8-11D8-44B6-ACEB-301713D5ED8C}\SAM\Domains\Account\U
```

c. **Hashdump**: Hashdump dumps passwords hashes (LM/NTLM) from memory. This command can be used to display the hashed credentials for user accounts, and these hashes can then be used in other tools to determine their account passwords.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> hashdump –y <virtual address of SYSTEM hive> -s <virtual address of SAM hive>

   ii. Displays:

   - Username
   - Domain Name
   - Hashed password



```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7S
Volatile Systems Volatility Framework 2.2
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
LCDI:1000:aad3b435b51404eeaad3b435b51404ee:af9e937641392e4f7f073c7971dbea96::

C:\>_
```

Malware Analysis

Plugins relating to this section aid in finding hidden malicious codes, as well as figuring out what malware is operating on the system.

a. **Malfind**: Malfind finds hidden or injected code. This command will find hidden or injected code/DLLs and would be useful in an investigation to discover/analyze malware.

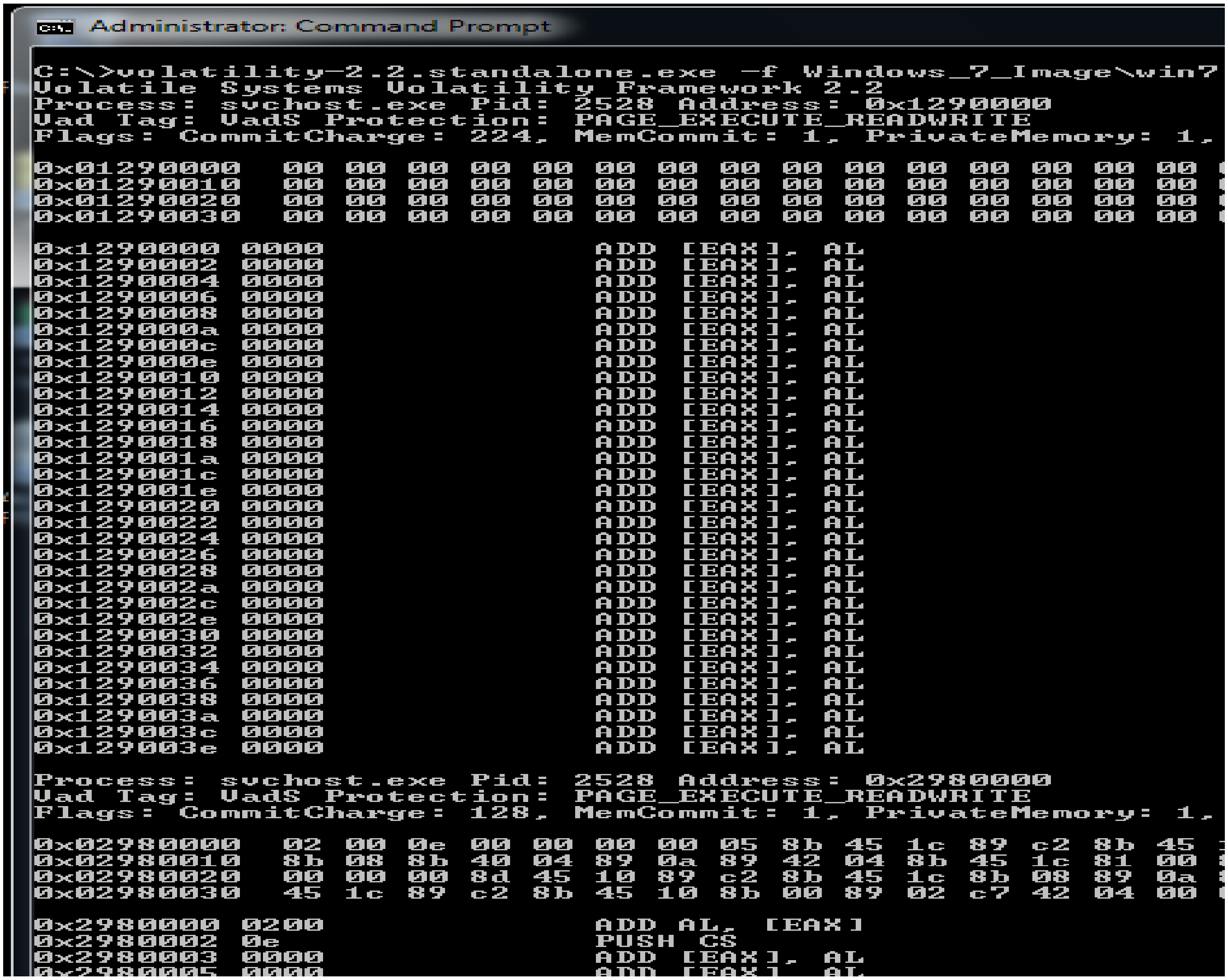
   i. Usage**:** volatility-2.2.standalone.exe –f <path to image> --profile=<profile> malfind –p <PID>

      1. -D <DIR>: Extracts copy of identified memory segment to disk
      2. --dump-dir=DIR: Extracts copy of identified memory segment to disk

   ii. Displays:

   - Process
   - Vad Tag

- Flags
- Memory segment



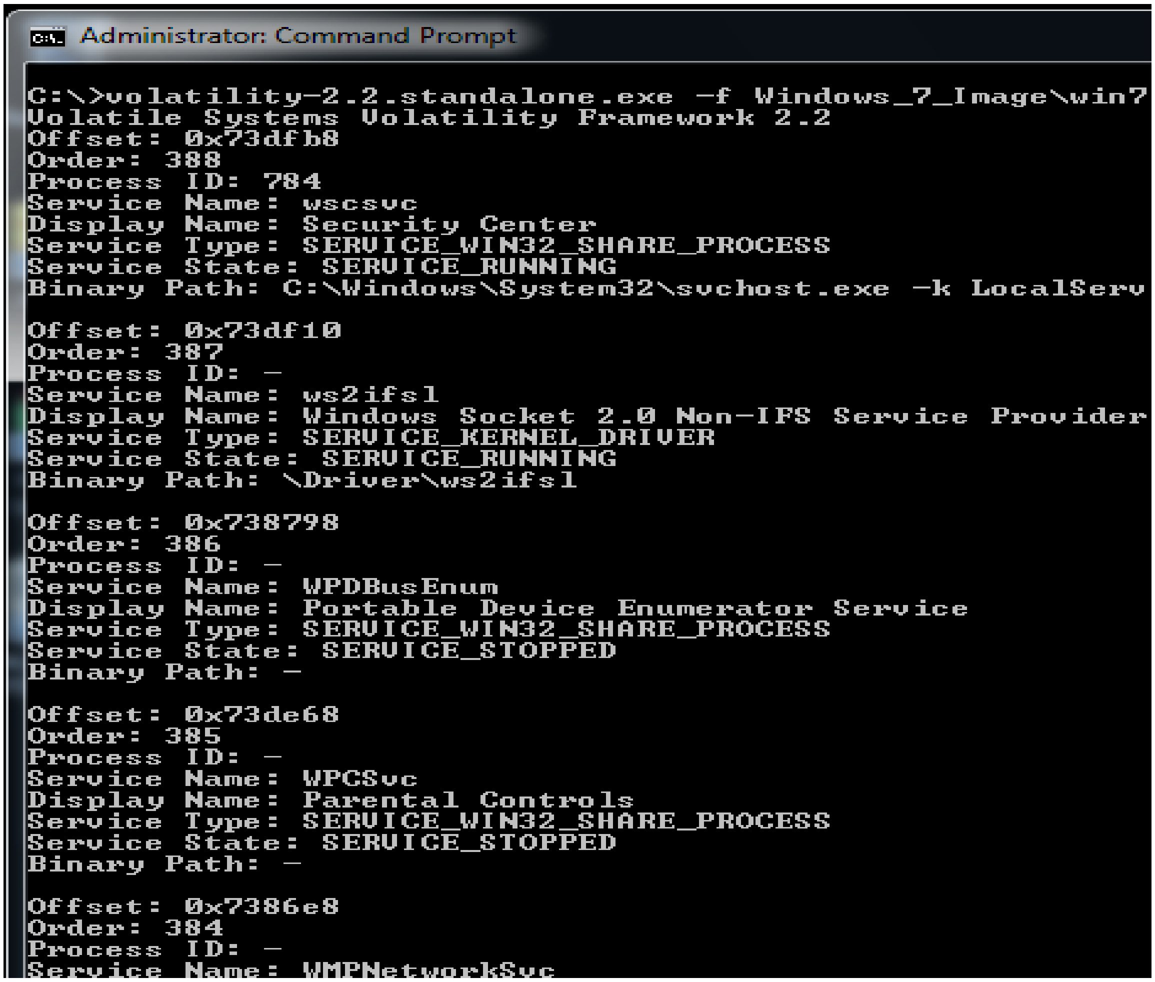
```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7
Volatile Systems Volatility Framework 2.2
Process: svchost.exe Pid: 2528 Address: 0x1290000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 224, MemCommit: 1, PrivateMemory: 1,

0x01290000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x01290010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x01290020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x01290030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0x1290000 0000                 ADD [EAX], AL
0x1290002 0000                 ADD [EAX], AL
0x1290004 0000                 ADD [EAX], AL
0x1290006 0000                 ADD [EAX], AL
0x1290008 0000                 ADD [EAX], AL
0x129000a 0000                 ADD [EAX], AL
0x129000c 0000                 ADD [EAX], AL
0x129000e 0000                 ADD [EAX], AL
0x1290010 0000                 ADD [EAX], AL
0x1290012 0000                 ADD [EAX], AL
0x1290014 0000                 ADD [EAX], AL
0x1290016 0000                 ADD [EAX], AL
0x1290018 0000                 ADD [EAX], AL
0x129001a 0000                 ADD [EAX], AL
0x129001c 0000                 ADD [EAX], AL
0x129001e 0000                 ADD [EAX], AL
0x1290020 0000                 ADD [EAX], AL
0x1290022 0000                 ADD [EAX], AL
0x1290024 0000                 ADD [EAX], AL
0x1290026 0000                 ADD [EAX], AL
0x1290028 0000                 ADD [EAX], AL
0x129002a 0000                 ADD [EAX], AL
0x129002c 0000                 ADD [EAX], AL
0x129002e 0000                 ADD [EAX], AL
0x1290030 0000                 ADD [EAX], AL
0x1290032 0000                 ADD [EAX], AL
0x1290034 0000                 ADD [EAX], AL
0x1290036 0000                 ADD [EAX], AL
0x1290038 0000                 ADD [EAX], AL
0x129003a 0000                 ADD [EAX], AL
0x129003c 0000                 ADD [EAX], AL
0x129003e 0000                 ADD [EAX], AL

Process: svchost.exe Pid: 2528 Address: 0x2980000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 128, MemCommit: 1, PrivateMemory: 1,

0x02980000  02 00 0e 00 00 00 00 05 8b 45 1c 89 c2 8b 45
0x02980010  8b 08 8b 40 04 89 0a 89 42 04 8b 45 1c 81 00
0x02980020  00 00 00 8d 45 10 89 c2 8b 45 1c 8b 08 89 0a
0x02980030  45 1c 89 c2 8b 45 10 8b 00 89 02 c7 42 04 00

0x2980000 0200                 ADD AL, [EAX]
0x2980002 0e                   PUSH CS
0x2980003 0000                 ADD [EAX], AL
0x2980005 0000                 ADD [EAX], AL
```

b. **Svcscan**: This plugin scans for Windows Services.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> svcscan

   ii. Displays:

- Offset
- Order
- Process ID
- Service Name
- Display Name
- Service Type
- Service State
- Binary Path



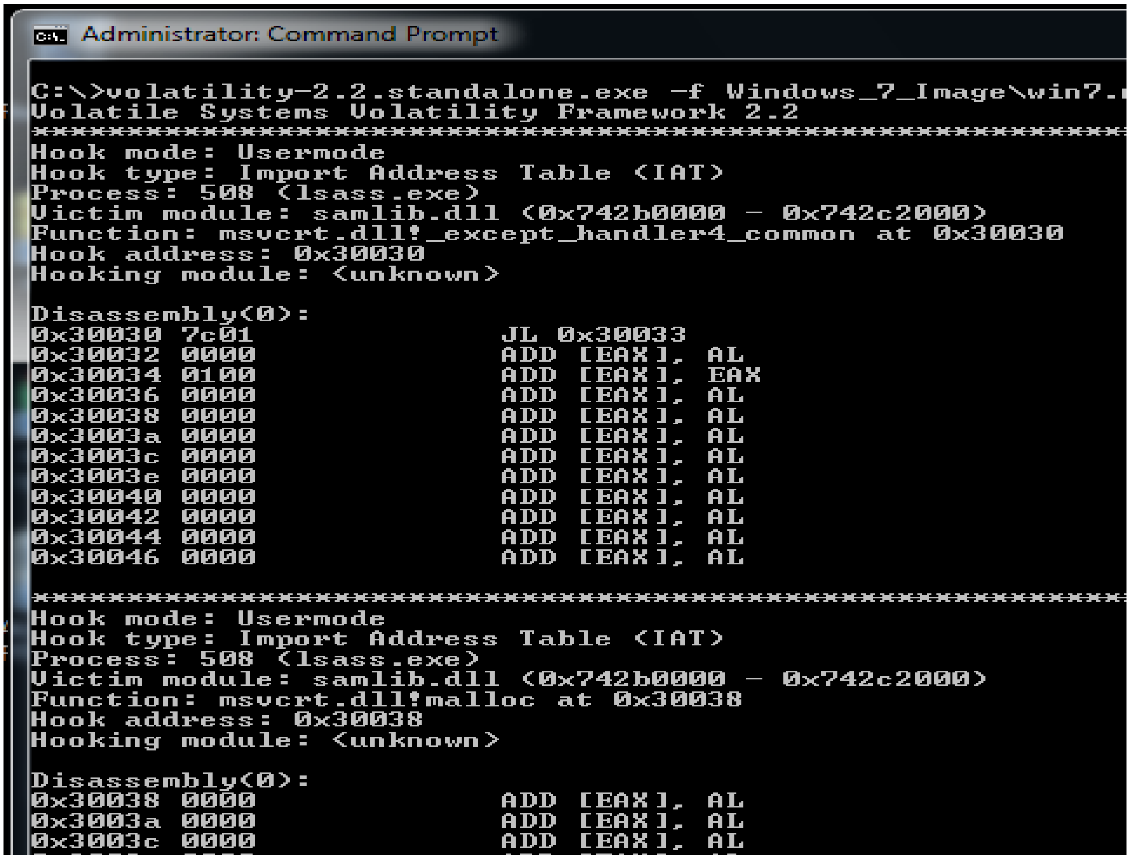
```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7
Volatile Systems Volatility Framework 2.2
Offset: 0x73dfb8
Order: 388
Process ID: 784
Service Name: wscsvc
Display Name: Security Center
Service Type: SERVICE_WIN32_SHARE_PROCESS
Service State: SERVICE_RUNNING
Binary Path: C:\Windows\System32\svchost.exe -k LocalServ

Offset: 0x73df10
Order: 387
Process ID: -
Service Name: ws2ifsl
Display Name: Windows Socket 2.0 Non-IFS Service Provider
Service Type: SERVICE_KERNEL_DRIVER
Service State: SERVICE_RUNNING
Binary Path: \Driver\ws2ifsl

Offset: 0x738798
Order: 386
Process ID: -
Service Name: WPDBusEnum
Display Name: Portable Device Enumerator Service
Service Type: SERVICE_WIN32_SHARE_PROCESS
Service State: SERVICE_STOPPED
Binary Path: -

Offset: 0x73de68
Order: 385
Process ID: -
Service Name: WPCSvc
Display Name: Parental Controls
Service Type: SERVICE_WIN32_SHARE_PROCESS
Service State: SERVICE_STOPPED
Binary Path: -

Offset: 0x7386e8
Order: 384
Process ID: -
Service Name: WMPNetworkSvc
```

c. **Apihooks**: This plugin detectsf API hooks in process and kernel memory. This command discovers instances of code hooking into other APIs. It would be useful in a malware investigation to determine how malicious software is operating.

i. volatility-2.2.standalone.exe –f <path to image> --profile=<profile> apihooks -p <PID>

ii. Displays:

- Hook mode
- Hook type
- Process
- Victim module
- Function
- Hook Address
- Hooking Module
- Disassembly



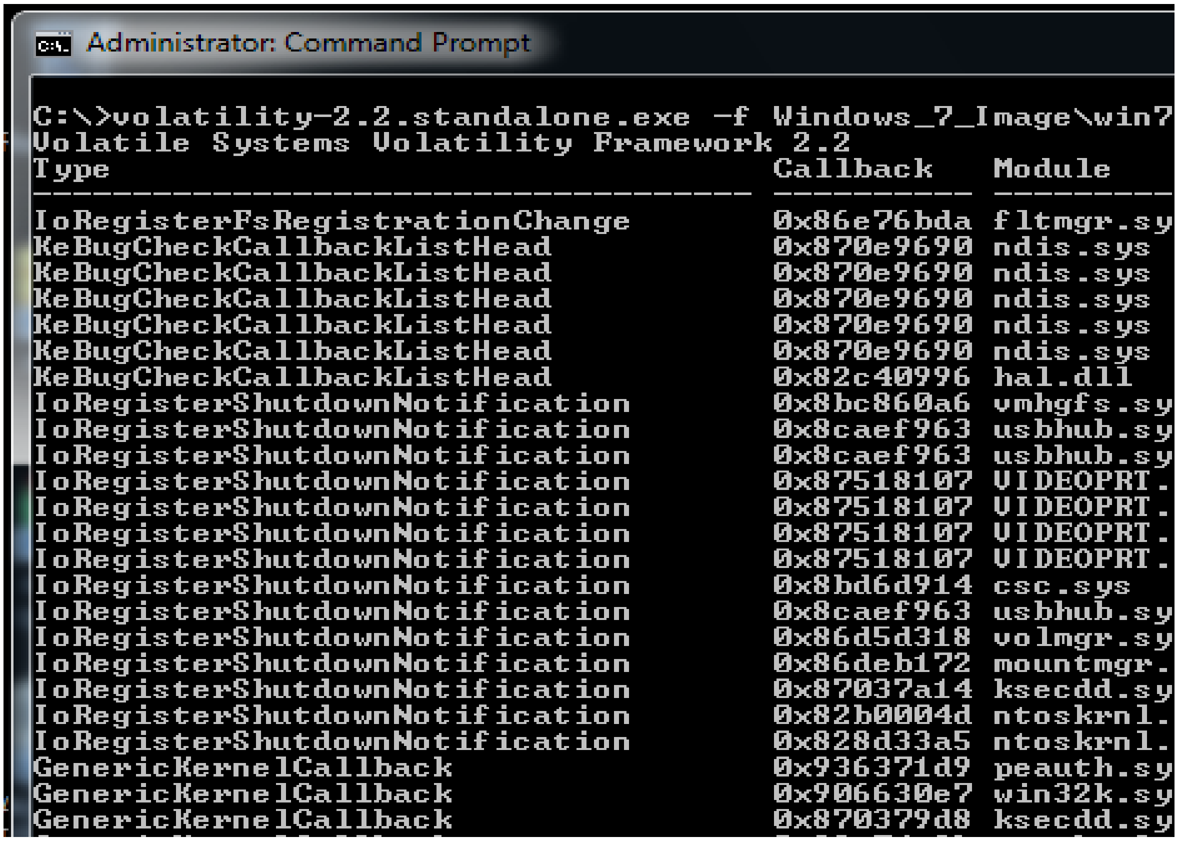
```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.
Volatile Systems Volatility Framework 2.2
************************************************************
Hook mode: Usermode
Hook type: Import Address Table (IAT)
Process: 508 (lsass.exe)
Victim module: samlib.dll (0x742b0000 - 0x742c2000)
Function: msvcrt.dll!_except_handler4_common at 0x30030
Hook address: 0x30030
Hooking module: <unknown>

Disassembly(0):
0x30030 7c01                    JL 0x30033
0x30032 0000                    ADD [EAX], AL
0x30034 0100                    ADD [EAX], EAX
0x30036 0000                    ADD [EAX], AL
0x30038 0000                    ADD [EAX], AL
0x3003a 0000                    ADD [EAX], AL
0x3003c 0000                    ADD [EAX], AL
0x3003e 0000                    ADD [EAX], AL
0x30040 0000                    ADD [EAX], AL
0x30042 0000                    ADD [EAX], AL
0x30044 0000                    ADD [EAX], AL
0x30046 0000                    ADD [EAX], AL

************************************************************
Hook mode: Usermode
Hook type: Import Address Table (IAT)
Process: 508 (lsass.exe)
Victim module: samlib.dll (0x742b0000 - 0x742c2000)
Function: msvcrt.dll!malloc at 0x30038
Hook address: 0x30038
Hooking module: <unknown>

Disassembly(0):
0x30038 0000                    ADD [EAX], AL
0x3003a 0000                    ADD [EAX], AL
0x3003c 0000                    ADD [EAX], AL
```

d. **Callbacks**: This plugin prints system-wide notification routines. This command will display instances of software listening for callbacks. This can be useful to a malware investigation and help the investigator determine what activities malicious software is monitoring.

i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> callbacks

ii. Displays:

- Type
    a. PsSetCreateProcessNotifyRoutine
    b. PsSetCreateThreadNotifyRoutine
    c. PsSetImageLoadNotifyRoutine
    d. IoRegisterFsRegistrationChange
    e. KeRegisterBugCheck
    f. KeRegisterBugCheckReasonCallback.
    g. CmRegisterCallback
    h. CmRegisterCallbackEx
    i. IoRegisterShutdownNotification
    j. DbgSetDebugPrintCallback
    k. DbgkLkmdRegisterCallback
- Owner
- Callback

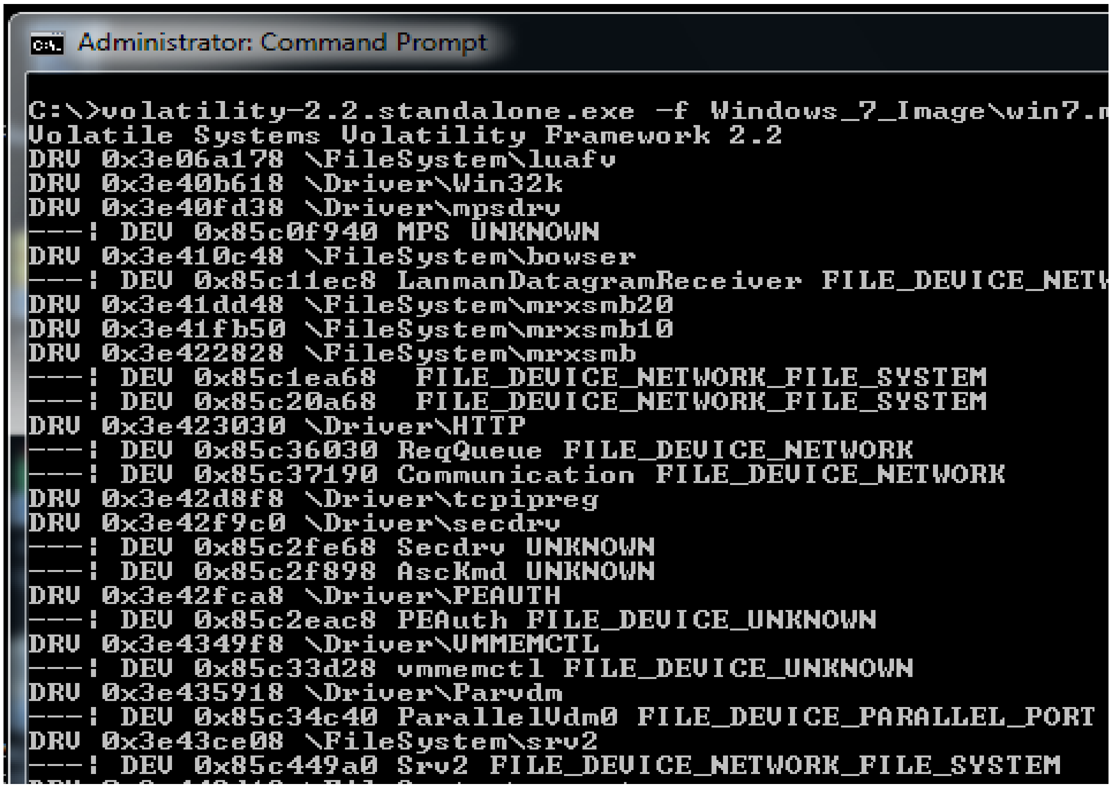


```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7
Volatile Systems Volatility Framework 2.2
Type                                 Callback   Module
------------------------------------ ---------- ---------
IoRegisterFsRegistrationChange       0x86e76bda fltmgr.sy
KeBugCheckCallbackListHead           0x870e9690 ndis.sys
KeBugCheckCallbackListHead           0x870e9690 ndis.sys
KeBugCheckCallbackListHead           0x870e9690 ndis.sys
KeBugCheckCallbackListHead           0x870e9690 ndis.sys
KeBugCheckCallbackListHead           0x870e9690 ndis.sys
KeBugCheckCallbackListHead           0x82c40996 hal.dll
IoRegisterShutdownNotification       0x8bc860a6 vmhgfs.sy
IoRegisterShutdownNotification       0x8caef963 usbhub.sy
IoRegisterShutdownNotification       0x8caef963 usbhub.sy
IoRegisterShutdownNotification       0x87518107 VIDEOPRT.
IoRegisterShutdownNotification       0x87518107 VIDEOPRT.
IoRegisterShutdownNotification       0x87518107 VIDEOPRT.
IoRegisterShutdownNotification       0x87518107 VIDEOPRT.
IoRegisterShutdownNotification       0x8bd6d914 csc.sys
IoRegisterShutdownNotification       0x8caef963 usbhub.sy
IoRegisterShutdownNotification       0x86d5d318 volmgr.sy
IoRegisterShutdownNotification       0x86deb172 mountmgr.
IoRegisterShutdownNotification       0x87037a14 ksecdd.sy
IoRegisterShutdownNotification       0x82b0004d ntoskrnl.
IoRegisterShutdownNotification       0x828d33a5 ntoskrnl.
GenericKernelCallback                0x936371d9 peauth.sy
GenericKernelCallback                0x906630e7 win32k.sy
GenericKernelCallback                0x870379d8 ksecdd.sy
```

e. **Devicetree**: Devicetree shows the relationship of a driver object to its devices and any attached devices. This command lists devices and driver objects in tree format. This is useful in malware investigations as malicious software were insert driver objects in order to intercept data.

    i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> devicetree

        1. DRV represents drivers
        2. DEV represents devices
        3. ATT represents attached devices



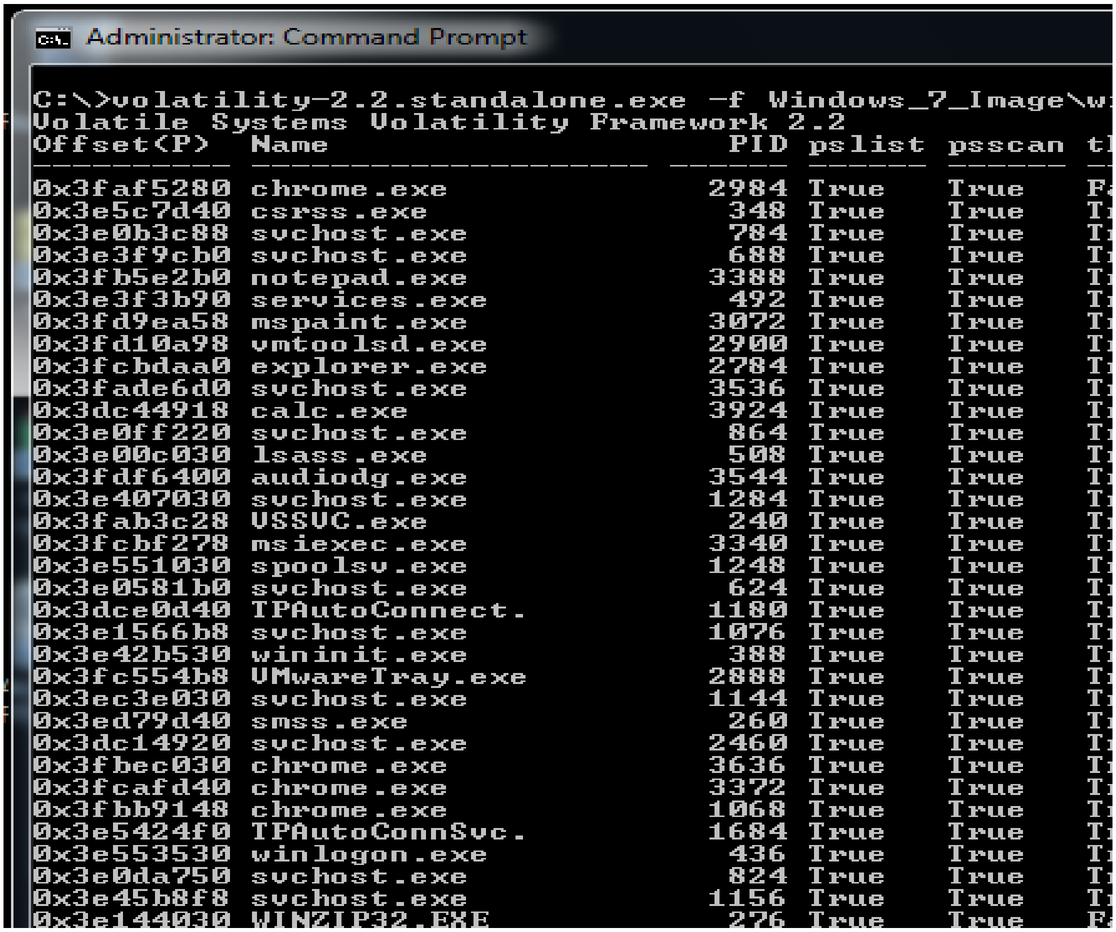
```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.m
Volatile Systems Volatility Framework 2.2
DRV 0x3e06a178 \FileSystem\luafv
DRV 0x3e40b618 \Driver\Win32k
DRV 0x3e40fd38 \Driver\mpsdrv
---| DEV 0x85c0f940 MPS UNKNOWN
DRV 0x3e410c48 \FileSystem\bowser
---| DEV 0x85c11ec8 LanmanDatagramReceiver FILE_DEVICE_NETW
DRV 0x3e41dd48 \FileSystem\mrxsmb20
DRV 0x3e41fb50 \FileSystem\mrxsmb10
DRV 0x3e422828 \FileSystem\mrxsmb
---| DEV 0x85c1ea68  FILE_DEVICE_NETWORK_FILE_SYSTEM
---| DEV 0x85c20a68  FILE_DEVICE_NETWORK_FILE_SYSTEM
DRV 0x3e423030 \Driver\HTTP
---| DEV 0x85c36030 ReqQueue FILE_DEVICE_NETWORK
---| DEV 0x85c37190 Communication FILE_DEVICE_NETWORK
DRV 0x3e42d8f8 \Driver\tcpipreg
DRV 0x3e42f9c0 \Driver\secdrv
---| DEV 0x85c2fe68 Secdrv UNKNOWN
---| DEV 0x85c2f898 AscKmd UNKNOWN
DRV 0x3e42fca8 \Driver\PEAUTH
---| DEV 0x85c2eac8 PEAuth FILE_DEVICE_UNKNOWN
DRV 0x3e4349f8 \Driver\VMMEMCTL
---| DEV 0x85c33d28 vmmemctl FILE_DEVICE_UNKNOWN
DRV 0x3e435918 \Driver\Parvdm
---| DEV 0x85c34c40 ParallelVdm0 FILE_DEVICE_PARALLEL_PORT
DRV 0x3e43ce08 \FileSystem\srv2
---| DEV 0x85c449a0 Srv2 FILE_DEVICE_NETWORK_FILE_SYSTEM
```

f. **Psxview**: This plugin finds hidden processes with various process listings. This command will list every process and whether or not the process is listed in different sources of process listings. The command can be useful in an investigation by aiding in discovering hidden processes.

i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> psxview

ii. Displays

- Offset (By default Virtual Offset, -P for Physical)
- Name
- PID
- Pslist
- Psscan
- Thrdproc
- Pspcdid
- Csrss



Administrator: Command Prompt

```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\w
Volatile Systems Volatility Framework 2.2
Offset(P)  Name                    PID pslist psscan t
---------- -------------------- ------ ------ ------ -
0x3faf5280 chrome.exe             2984 True   True   F
0x3e5c7d40 csrss.exe               348 True   True   T
0x3e0b3c88 svchost.exe             784 True   True   T
0x3e3f9cb0 svchost.exe             688 True   True   T
0x3fb5e2b0 notepad.exe            3388 True   True   T
0x3e3f3b90 services.exe            492 True   True   T
0x3fd9ea58 mspaint.exe            3072 True   True   T
0x3fd10a98 vmtoolsd.exe           2900 True   True   T
0x3fcbdaa0 explorer.exe           2784 True   True   T
0x3fade6d0 svchost.exe            3536 True   True   T
0x3dc44918 calc.exe               3924 True   True   T
0x3e0ff220 svchost.exe             864 True   True   T
0x3e00c030 lsass.exe               508 True   True   T
0x3fdf6400 audiodg.exe            3544 True   True   T
0x3e407030 svchost.exe            1284 True   True   T
0x3fab3c28 VSSVC.exe               240 True   True   T
0x3fcbf278 msiexec.exe            3340 True   True   T
0x3e551030 spoolsv.exe            1248 True   True   T
0x3e0581b0 svchost.exe             624 True   True   T
0x3dce0d40 TPAutoConnect.         1180 True   True   T
0x3e1566b8 svchost.exe            1076 True   True   T
0x3e42b530 wininit.exe             388 True   True   T
0x3fc554b8 VMwareTray.exe         2888 True   True   T
0x3ec3e030 svchost.exe            1144 True   True   T
0x3ed79d40 smss.exe                260 True   True   T
0x3dc14920 svchost.exe            2460 True   True   T
0x3fbec030 chrome.exe             3636 True   True   T
0x3fcafd40 chrome.exe             3372 True   True   T
0x3fbb9148 chrome.exe             1068 True   True   T
0x3e5424f0 TPAutoConnSvc.         1684 True   True   T
0x3e553530 winlogon.exe            436 True   True   T
0x3e0da750 svchost.exe             824 True   True   T
0x3e45b8f8 svchost.exe            1156 True   True   T
0x3e144030 WINZIP32.EXE            276 True   True   F
```

GUI Analysis

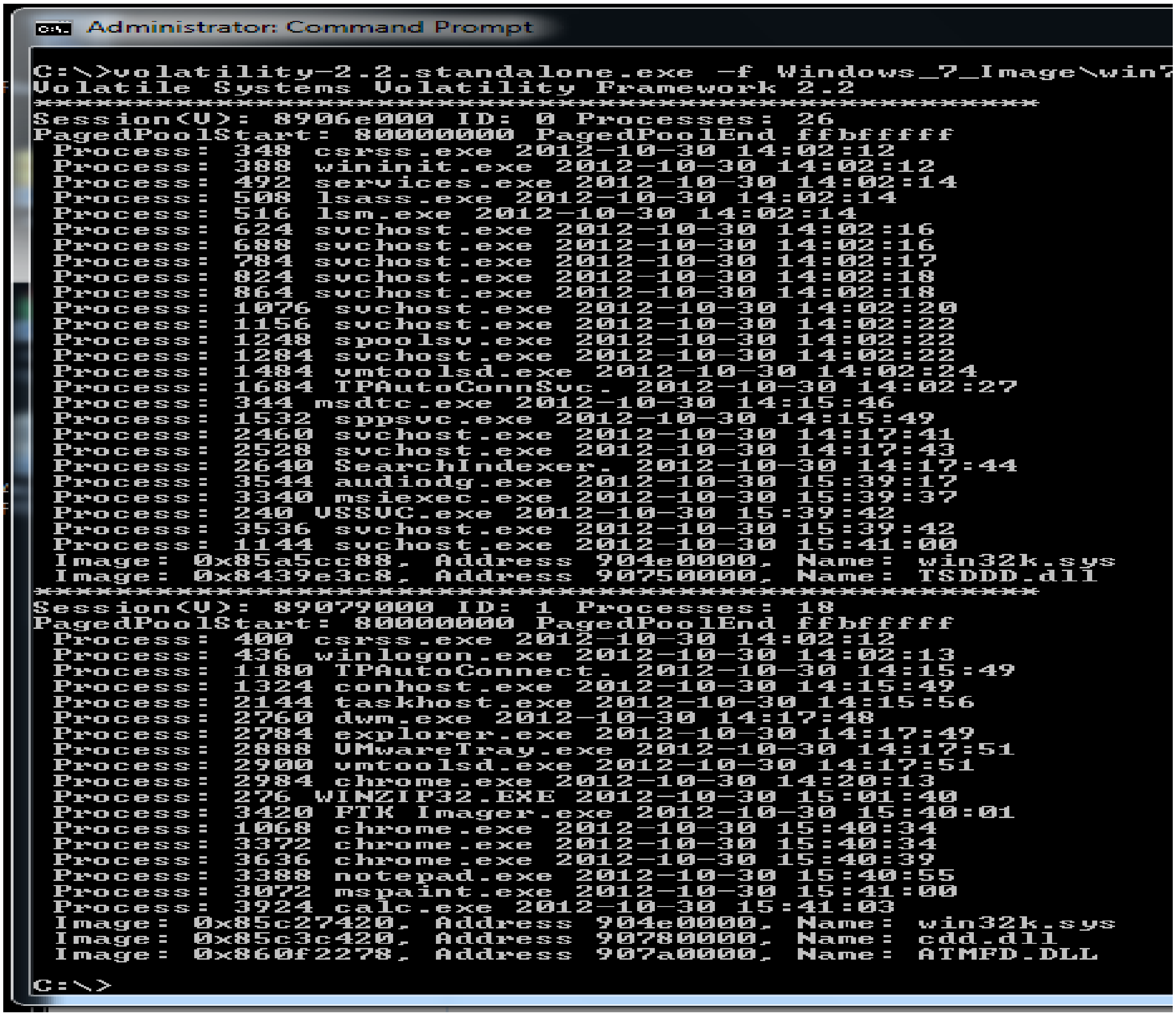
All the plugins mentioned below are new and were implemented during the Month of Volatility. They assist in recreating the graphical interface at the time a system's memory is dumped.

a. **Sessions**: Sessions lists details on _MM_SESSION_SPACE (user logon sessions). This command lists running processes, separated by which session they were launched in. This information is of evidentiary value because you can determine which session a process was started in. For example, you can see which commands were started from a remote session.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> sessions

   ii. Displays:

   - Session Number
   - Number of Processes
   - List of Processes
   - Image list



```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7
Volatile Systems Volatility Framework 2.2
**************************************************
Session(V): 8906e000 ID: 0 Processes: 26
PagedPoolStart: 80000000 PagedPoolEnd ffbfffff
 Process: 348 csrss.exe 2012-10-30 14:02:12
 Process: 388 wininit.exe 2012-10-30 14:02:12
 Process: 492 services.exe 2012-10-30 14:02:14
 Process: 508 lsass.exe 2012-10-30 14:02:14
 Process: 516 lsm.exe 2012-10-30 14:02:14
 Process: 624 svchost.exe 2012-10-30 14:02:16
 Process: 688 svchost.exe 2012-10-30 14:02:16
 Process: 784 svchost.exe 2012-10-30 14:02:17
 Process: 824 svchost.exe 2012-10-30 14:02:18
 Process: 864 svchost.exe 2012-10-30 14:02:18
 Process: 1076 svchost.exe 2012-10-30 14:02:20
 Process: 1156 svchost.exe 2012-10-30 14:02:22
 Process: 1248 spoolsv.exe 2012-10-30 14:02:22
 Process: 1284 svchost.exe 2012-10-30 14:02:22
 Process: 1484 vmtoolsd.exe 2012-10-30 14:02:24
 Process: 1684 TPAutoConnSvc. 2012-10-30 14:02:27
 Process: 344 msdtc.exe 2012-10-30 14:15:46
 Process: 1532 sppsvc.exe 2012-10-30 14:15:49
 Process: 2460 svchost.exe 2012-10-30 14:17:41
 Process: 2528 svchost.exe 2012-10-30 14:17:43
 Process: 2640 SearchIndexer. 2012-10-30 14:17:44
 Process: 3544 audiodg.exe 2012-10-30 15:39:17
 Process: 3340 msiexec.exe 2012-10-30 15:39:37
 Process: 240 VSSVC.exe 2012-10-30 15:39:42
 Process: 3536 svchost.exe 2012-10-30 15:39:42
 Process: 1144 svchost.exe 2012-10-30 15:41:00
 Image: 0x85a5cc88, Address 904e0000, Name: win32k.sys
 Image: 0x8439e3c8, Address 90750000, Name: TSDDD.dll
**************************************************
Session(V): 89079000 ID: 1 Processes: 18
PagedPoolStart: 80000000 PagedPoolEnd ffbfffff
 Process: 400 csrss.exe 2012-10-30 14:02:12
 Process: 436 winlogon.exe 2012-10-30 14:02:13
 Process: 1180 TPAutoConnect. 2012-10-30 14:15:49
 Process: 1324 conhost.exe 2012-10-30 14:15:49
 Process: 2144 taskhost.exe 2012-10-30 14:15:56
 Process: 2760 dwm.exe 2012-10-30 14:17:48
 Process: 2784 explorer.exe 2012-10-30 14:17:49
 Process: 2888 VMwareTray.exe 2012-10-30 14:17:51
 Process: 2900 vmtoolsd.exe 2012-10-30 14:17:51
 Process: 2984 chrome.exe 2012-10-30 14:20:13
 Process: 276 WINZIP32.EXE 2012-10-30 15:01:40
 Process: 3420 FTK Imager.exe 2012-10-30 15:40:01
 Process: 1068 chrome.exe 2012-10-30 15:40:34
 Process: 3372 chrome.exe 2012-10-30 15:40:34
 Process: 3636 chrome.exe 2012-10-30 15:40:39
 Process: 3388 notepad.exe 2012-10-30 15:40:55
 Process: 3072 mspaint.exe 2012-10-30 15:41:00
 Process: 3924 calc.exe 2012-10-30 15:41:03
 Image: 0x85c27420, Address 904e0000, Name: win32k.sys
 Image: 0x85c3c420, Address 90780000, Name: cdd.dll
 Image: 0x860f2278, Address 907a0000, Name: ATMFD.DLL

C:\>
```

b. **Wndscan**: Wndscan is a pool scanner for tagWINDOWSTATION (window stations). This command details information on window stations and which processes are interacting with the clipboard. This command could be used in an investigation to show that a specific process was using the clipboard.

   i. Volatility-2.2.standalone.exe –f <path to image> --profile=<profile> wndscan

   ii. Displays

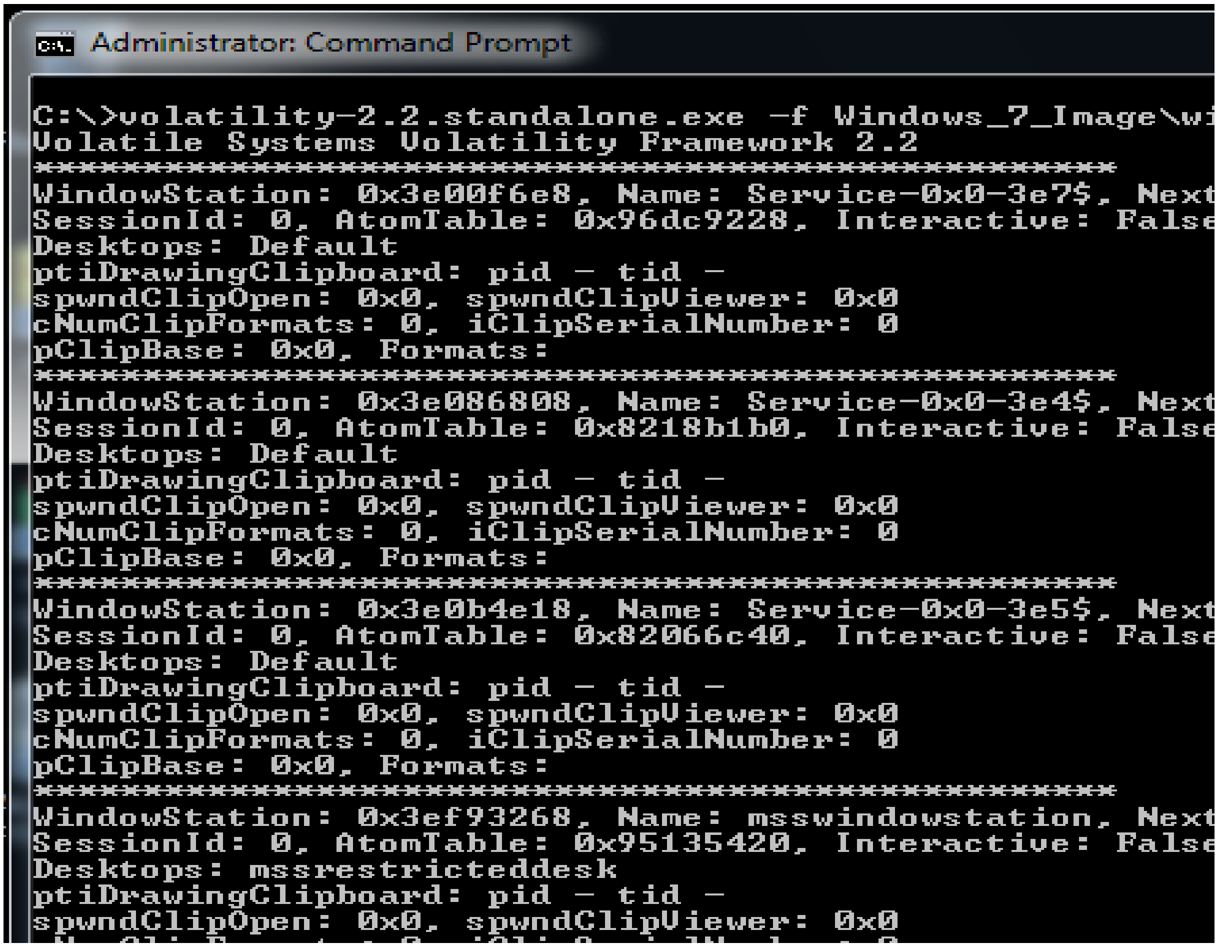
   - Window Station Name
   - Session ID
   - Atom Table
   - Desktops
   - The process viewing the clipboard
   - Number of items in the clipboard

c. **Atoms**: This plugin prints session and window station atom tables. This plugin will display atom table information and link each entry to the session and window station which own it. This information can be beneficial in malware investigations by discoveringq artifacts that many people would not think of in an attempt to cover their tracks.

   i. Usage: volatility-2.2.standalone.exe –f <path to image> --profile=<profile> atom

ii. Displays

- Offset
- Session
- WindowStation
- Atom
- RefCount
- HIndex
- Pinned
- Name



```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\wi
Volatile Systems Volatility Framework 2.2
**************************************************
WindowStation: 0x3e00f6e8, Name: Service-0x0-3e7$, Next
SessionId: 0, AtomTable: 0x96dc9228, Interactive: False
Desktops: Default
ptiDrawingClipboard: pid - tid -
spwndClipOpen: 0x0, spwndClipViewer: 0x0
cNumClipFormats: 0, iClipSerialNumber: 0
pClipBase: 0x0, Formats:
**************************************************
WindowStation: 0x3e086808, Name: Service-0x0-3e4$, Next
SessionId: 0, AtomTable: 0x8218b1b0, Interactive: False
Desktops: Default
ptiDrawingClipboard: pid - tid -
spwndClipOpen: 0x0, spwndClipViewer: 0x0
cNumClipFormats: 0, iClipSerialNumber: 0
pClipBase: 0x0, Formats:
**************************************************
WindowStation: 0x3e0b4e18, Name: Service-0x0-3e5$, Next
SessionId: 0, AtomTable: 0x82066c40, Interactive: False
Desktops: Default
ptiDrawingClipboard: pid - tid -
spwndClipOpen: 0x0, spwndClipViewer: 0x0
cNumClipFormats: 0, iClipSerialNumber: 0
pClipBase: 0x0, Formats:
**************************************************
WindowStation: 0x3ef93268, Name: msswindowstation, Next
SessionId: 0, AtomTable: 0x95135420, Interactive: False
Desktops: mssrestricteddesk
ptiDrawingClipboard: pid - tid -
spwndClipOpen: 0x0, spwndClipViewer: 0x0
```

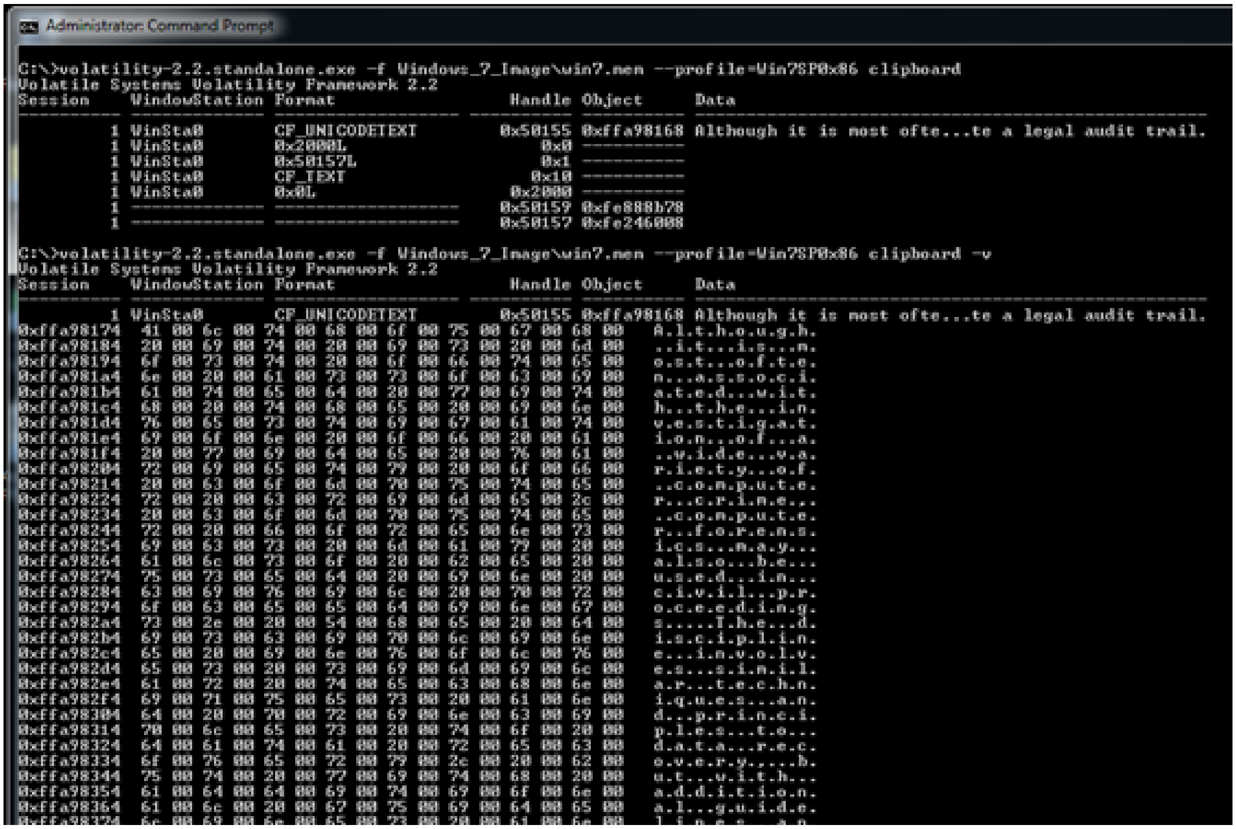
d. **Clipboard**: This command can extract the information stored in the clipboard.

i. volatility-2.2.standalone.exe –f <path to image> --profile=<profile> clipboard

1. -v: Displays the clipboard data in hex

ii. Displays

- Session
- Window Station
- Format
- Handle
- Object
- Data



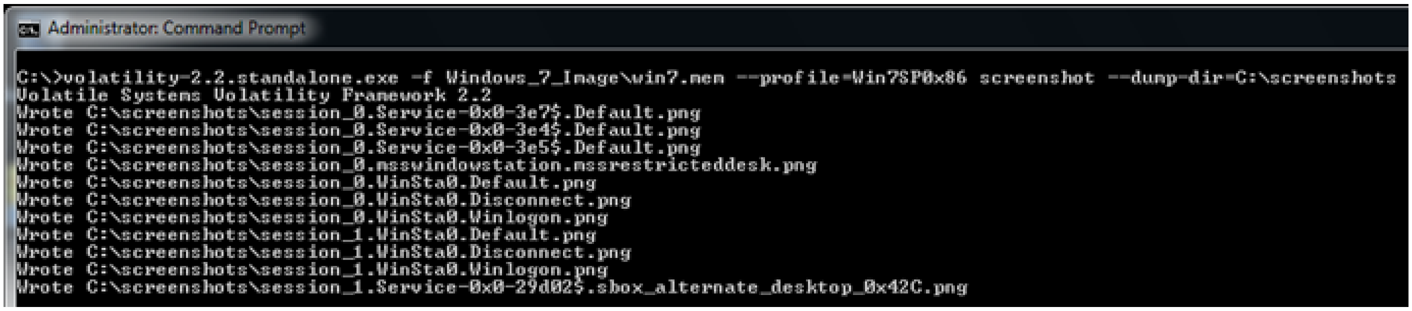
```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 clipboard
Volatile Systems Volatility Framework 2.2
Session    WindowStation Format                 Handle Object     Data
---------- ------------- ------------------ ---------- ---------- --------------------------------------------------
         1 WinSta0       CF_UNICODETEXT        0x50155 0xffa98168 Although it is most ofte...te a legal audit trail.
         1 WinSta0       0x2000L                   0x0 ----------
         1 WinSta0       0x50157L                  0x1 ----------
         1 WinSta0       CF_TEXT                  0x10 ----------
         1 WinSta0       0x0L                   0x2000 ----------
         1 ------------- ------------------    0x50159 0xfe888b78
         1 ------------- ------------------    0x50157 0xfe246008

C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 clipboard -v
Volatile Systems Volatility Framework 2.2
Session    WindowStation Format                 Handle Object     Data
---------- ------------- ------------------ ---------- ---------- --------------------------------------------------
         1 WinSta0       CF_UNICODETEXT        0x50155 0xffa98168 Although it is most ofte...te a legal audit trail.
0xffa98174  41 00 6c 00 74 00 68 00 6f 00 75 00 67 00 68 00   A.l.t.h.o.u.g.h.
0xffa98184  20 00 69 00 74 00 20 00 69 00 73 00 20 00 6d 00   ..i.t...i.s...m.
0xffa98194  6f 00 73 00 74 00 20 00 6f 00 66 00 74 00 65 00   o.s.t...o.f.t.e.
0xffa981a4  6e 00 20 00 61 00 73 00 73 00 6f 00 63 00 69 00   n...a.s.s.o.c.i.
0xffa981b4  61 00 74 00 65 00 64 00 20 00 77 00 69 00 74 00   a.t.e.d...w.i.t.
0xffa981c4  68 00 20 00 74 00 68 00 65 00 20 00 69 00 6e 00   h...t.h.e...i.n.
0xffa981d4  76 00 65 00 73 00 74 00 69 00 67 00 61 00 74 00   v.e.s.t.i.g.a.t.
0xffa981e4  69 00 6f 00 6e 00 20 00 6f 00 66 00 20 00 61 00   i.o.n...o.f...a.
0xffa981f4  20 00 77 00 69 00 64 00 65 00 20 00 76 00 61 00   ..w.i.d.e...v.a.
0xffa98204  72 00 69 00 65 00 74 00 79 00 20 00 6f 00 66 00   r.i.e.t.y...o.f.
0xffa98214  20 00 63 00 6f 00 6d 00 70 00 75 00 74 00 65 00   ..c.o.m.p.u.t.e.
0xffa98224  72 00 20 00 63 00 72 00 69 00 6d 00 65 00 2c 00   r...c.r.i.m.e...
0xffa98234  20 00 63 00 6f 00 6d 00 70 00 75 00 74 00 65 00   ..c.o.m.p.u.t.e.
0xffa98244  72 00 20 00 66 00 6f 00 72 00 65 00 6e 00 73 00   r...f.o.r.e.n.s.
0xffa98254  69 00 63 00 73 00 20 00 6d 00 61 00 79 00 20 00   i.c.s...m.a.y...
0xffa98264  61 00 6c 00 73 00 6f 00 20 00 62 00 65 00 20 00   a.l.s.o...b.e...
0xffa98274  75 00 73 00 65 00 64 00 20 00 69 00 6e 00 20 00   u.s.e.d...i.n...
0xffa98284  63 00 69 00 76 00 69 00 6c 00 20 00 70 00 72 00   c.i.v.i.l...p.r.
0xffa98294  6f 00 63 00 65 00 65 00 64 00 69 00 6e 00 67 00   o.c.e.e.d.i.n.g.
0xffa982a4  73 00 2e 00 20 00 54 00 68 00 65 00 20 00 64 00   s.....T.h.e...d.
0xffa982b4  69 00 73 00 63 00 69 00 70 00 6c 00 69 00 6e 00   i.s.c.i.p.l.i.n.
0xffa982c4  65 00 20 00 69 00 6e 00 76 00 6f 00 6c 00 76 00   e...i.n.v.o.l.v.
0xffa982d4  65 00 73 00 20 00 73 00 69 00 6d 00 69 00 6c 00   e.s...s.i.m.i.l.
0xffa982e4  61 00 72 00 20 00 74 00 65 00 63 00 68 00 6e 00   a.r...t.e.c.h.n.
0xffa982f4  69 00 71 00 75 00 65 00 73 00 20 00 61 00 6e 00   i.q.u.e.s...a.n.
0xffa98304  64 00 20 00 70 00 72 00 69 00 6e 00 63 00 69 00   d...p.r.i.n.c.i.
0xffa98314  70 00 6c 00 65 00 73 00 20 00 74 00 6f 00 20 00   p.l.e.s...t.o...
0xffa98324  64 00 61 00 74 00 61 00 20 00 72 00 65 00 63 00   d.a.t.a...r.e.c.
0xffa98334  6f 00 76 00 65 00 72 00 79 00 2c 00 20 00 62 00   o.v.e.r.y.....b.
0xffa98344  75 00 74 00 20 00 77 00 69 00 74 00 68 00 20 00   u.t...w.i.t.h...
0xffa98354  61 00 64 00 64 00 69 00 74 00 69 00 6f 00 6e 00   a.d.d.i.t.i.o.n.
0xffa98364  61 00 6c 00 20 00 67 00 75 00 69 00 64 00 65 00   a.l...g.u.i.d.e.
0xffa98374  6c 00 69 00 6e 00 65 00 73 00 20 00 61 00 6e 00   l.i.n.e.s...a.n.
```

e. **Screenshot**: Screenshot saves a pseudo-screenshot based on GDI windows. This command will create a wireframe outline of the window positioning for each window station. Starting in Volatility 2.3, this will include the titles of each window. These screenshots will be beneficial to a case because they will display the desktop as the user saw it.

   i. Volatility-2.2.standalone.exe –f <path to image> --profile=<profile> screenshot --dump-dir=<path to directory

   ii. Displays

      - Path to each screenshot of each session and desktop



```
C:\>volatility-2.2.standalone.exe -f Windows_7_Image\win7.mem --profile=Win7SP0x86 screenshot --dump-dir=C:\screenshots
Volatile Systems Volatility Framework 2.2
Wrote C:\screenshots\session_0.Service-0x0-3e7$.Default.png
Wrote C:\screenshots\session_0.Service-0x0-3e4$.Default.png
Wrote C:\screenshots\session_0.Service-0x0-3e5$.Default.png
Wrote C:\screenshots\session_0.msswindowstation.mssrestricteddesk.png
Wrote C:\screenshots\session_0.WinSta0.Default.png
Wrote C:\screenshots\session_0.WinSta0.Disconnect.png
Wrote C:\screenshots\session_0.WinSta0.Winlogon.png
Wrote C:\screenshots\session_1.WinSta0.Default.png
Wrote C:\screenshots\session_1.WinSta0.Disconnect.png
Wrote C:\screenshots\session_1.WinSta0.Winlogon.png
Wrote C:\screenshots\session_1.Service-0x0-29d02$.sbox_alternate_desktop_0x42C.png
```

4 Other Plugins

The following plugins also came from the Month of Volatility and are not categorized in any of the previous sub-headings, as they individually perform differently.

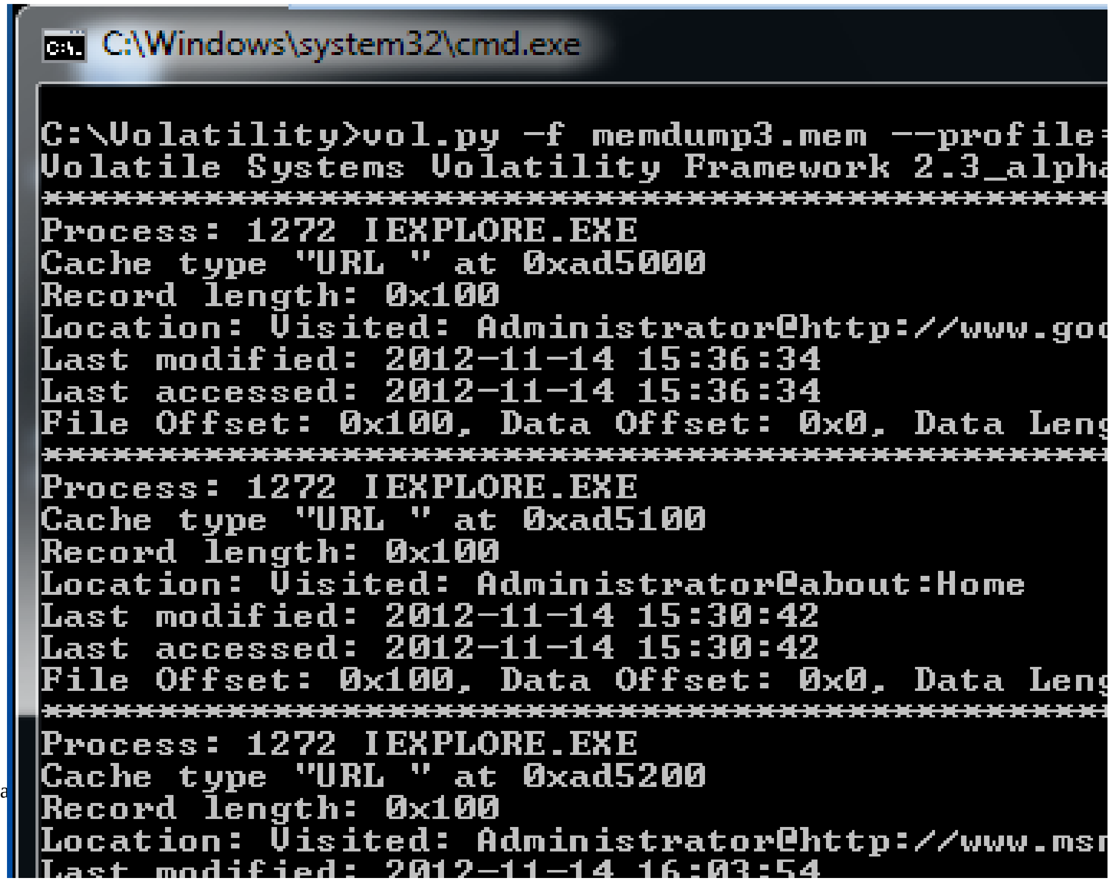
a. **Iehistory**: This plugin will reconstruct Internet Explorer cache/history. This can be useful in an investigation to examine a user's internet activity.

   i. Usage**:** volatility-2.2.standalone.exe -f <path to image> --profile=<profile> iehistory

      1. --pid: Filter by process
      2. --offset: Filter by offset
      3. --leak
      4. --redr

   ii. Displays

      - Process
      - Cache type
      - Record length
      - Location
      - Last modified
      - Last accessed
      - File Offset
      - File name

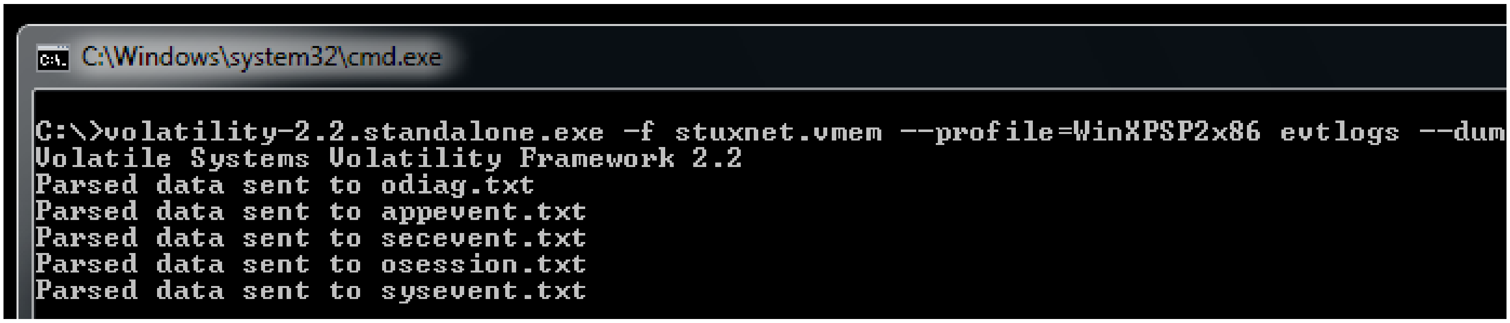


```
C:\Volatility>vol.py -f memdump3.mem --profile
Volatile Systems Volatility Framework 2.3_alpha
**************************************************
Process: 1272 IEXPLORE.EXE
Cache type "URL " at 0xad5000
Record length: 0x100
Location: Visited: Administrator@http://www.goo
Last modified: 2012-11-14 15:36:34
Last accessed: 2012-11-14 15:36:34
File Offset: 0x100, Data Offset: 0x0, Data Leng
**************************************************
Process: 1272 IEXPLORE.EXE
Cache type "URL " at 0xad5100
Record length: 0x100
Location: Visited: Administrator@about:Home
Last modified: 2012-11-14 15:30:42
Last accessed: 2012-11-14 15:30:42
File Offset: 0x100, Data Offset: 0x0, Data Leng
**************************************************
Process: 1272 IEXPLORE.EXE
Cache type "URL " at 0xad5200
Record length: 0x100
Location: Visited: Administrator@http://www.msn
Last modified: 2012-11-14 16:03:54
```

b. **Evtlogs**: This plugin extracts Windows Event Logs (XP/2003 only). This plugin can be useful in an investigation, as event logs can help understand when things happened on a system.

   i. Usage: volatility-2.2.standalone.exe -f <path to image> --profile=<profile> evtlogs -D <output>

      1. --save-evt: Saves the event logs (.evt)
      2. --verbose: SIDs are also evaluated



C:\Windows\system32\cmd.exe

```
C:\>volatility-2.2.standalone.exe -f stuxnet.vmem --profile=WinXPSP2x86 evtlogs --dum
Volatile Systems Volatility Framework 2.2
Parsed data sent to odiag.txt
Parsed data sent to appevent.txt
Parsed data sent to secevent.txt
Parsed data sent to osession.txt
Parsed data sent to sysevent.txt
```

c. **Deskscan**: Deskscan enumerates desktops, desktop heap allocations, and associated threads. It aids in finding rogue desktops used to hide applications from logged on users. It detects desktops created by ransomware and links threads to their desktops. It analyzes the desktop heap from memory corruptions and searches profile desktop heap allocations to locate USER objects.

4 References

Commands for image, processes, kernel memory, networking and registry plugins: https://code.google.com/p/volatility/wiki/CommandReference23

Commands for malware analysis plugins: https://code.google.com/p/volatility/wiki/CommandReferenceMal23

Commands for GUI analysis plugins: http://code.google.com/p/volatility/wiki/CommandReferenceGui22

A blog by the developers of Volatility: http://volatility-labs.blogspot.com/