Analyzing Memory with Volatility

Volatility [1] is a free, open source framework written in python. It is a command line tool used to analyze memory images from Linux, OS X, and Windows systems. Since it is written in python, it is capable of being run on any system that supports python. There is also a standalone Windows executable version. This document will assume the .exe version is being used. The only difference in running the two, assuming you have python installed, is how you first run the tool.

`volatility.exe` vs `python volatility.py (may also be vol.py)`

Procedure

1. Open command prompt
2. View Volatility help file
3. Run Volatility with appropriate INPUTFILE, PROFILE, and PLUGIN

Volatility Help

```
C:\tools>volatility.exe -h
Volatile Systems Volatility Framework 2.3_beta
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (semi-colon
                        separated)
  --info                Print information about all registered objects
  --cache-directory=C:\Users\Dae/.cache\volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the timezone for displaying timestamps
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
                        Name of the profile to load
  -l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
  -w, --write           Enable write support
  --dtb=DTB             DTB Address
  --cache-dtb           Cache virtual to physical mappings
  --output=text         Output in this format (format support is module
                        specific)
  --output-file=OUTPUT_FILE
                        write output in this file
  -v, --verbose         Verbose information
  --shift=SHIFT         Mac KASLR shift address
  -g KDBG, --kdbg=KDBG  Specify a specific KDBG virtual address
  -k KPCR, --kpcr=KPCR  Specify a specific KPCR address
```

Figure 1. Volatility Options

Running volatility with the *-h* flag lists the help file displayed in figure 1. The help file also lists all available plugins for the specified profile. We'll focus on a few of the above options.

`--info` lists all the available profiles and plugins available in that version of volatility.

`--tz=` lets sets the timezone. If the image was acquired in a CST timezone during daylight savings time, then *--tz=-0500* would be used so any times displayed by Volatility would appear as the correct local time.

`-f` or `--filename=` is used to specify the filename of the image to analyze.

`--profile=` specifies the profile to load. The default profile is WinXPSP2x86. Other profiles can be specified. Explained in more detail on next page.

In order to run Volatility, a filename and profile are required.

Volatility Profiles

```
Profiles
--------
VistaSP0x64     - A Profile for Windows Vista SP0 x64
VistaSP0x86     - A Profile for Windows Vista SP0 x86
VistaSP1x64     - A Profile for Windows Vista SP1 x64
VistaSP1x86     - A Profile for Windows Vista SP1 x86
VistaSP2x64     - A Profile for Windows Vista SP2 x64
VistaSP2x86     - A Profile for Windows Vista SP2 x86
Win2003SP0x86   - A Profile for Windows 2003 SP0 x86
Win2003SP1x64   - A Profile for Windows 2003 SP1 x64
Win2003SP1x86   - A Profile for Windows 2003 SP1 x86
Win2003SP2x64   - A Profile for Windows 2003 SP2 x64
Win2003SP2x86   - A Profile for Windows 2003 SP2 x86
Win2008R2SP0x64 - A Profile for Windows 2008 R2 SP0 x64
Win2008R2SP1x64 - A Profile for Windows 2008 R2 SP1 x64
Win2008SP1x64   - A Profile for Windows 2008 SP1 x64
Win2008SP1x86   - A Profile for Windows 2008 SP1 x86
Win2008SP2x64   - A Profile for Windows 2008 SP2 x64
Win2008SP2x86   - A Profile for Windows 2008 SP2 x86
Win7SP0x64      - A Profile for Windows 7 SP0 x64
Win7SP0x86      - A Profile for Windows 7 SP0 x86
Win7SP1x64      - A Profile for Windows 7 SP1 x64
Win7SP1x86      - A Profile for Windows 7 SP1 x86
WinXPSP1x64     - A Profile for Windows XP SP1 x64
WinXPSP2x64     - A Profile for Windows XP SP2 x64
WinXPSP2x86     - A Profile for Windows XP SP2 x86
WinXPSP3x86     - A Profile for Windows XP SP3 x86
```

Figure 2. Volatility Profiles

Profiles are what Volatility uses to understand a memory image. Memory is structured differently across different operating systems (Windows, Linux, OSX), different versions (XP, Vista, Lion, Mountain Lion), and different architectures (x86, x64).

The profiles shown above can be listed by using the --info flag: `volatility.exe --info`. If you wanted to analyze a 64-bit Windows Vista Service Pack 1 memory image, you would specify `--profile=VistaSP1x64`. If the image was from a 32-bit Windows Server 2003, Service Pack 2 system then you would use `--profile=Win2003SP2x86`. Whatever system the memory sample was pulled from must match the proper profile(s).

Note: x86 implies a 32-bit system

Volatility Plugins

```
Supported Plugin Commands:

        apihooks        Detect API hooks in process and kernel memory
        atoms           Print session and window station atom tables
        atomscan        Pool scanner for _RTL_ATOM_TABLE
        bioskbd         Reads the keyboard buffer from Real Mode memory
        callbacks       Print system-wide notification routines
        clipboard       Extract the contents of the windows clipboard
        cmdscan         Extract command history by scanning for _COMMAND_HISTORY
        connections     Print list of open connections [Windows XP and 2003 Only]
        connscan        Scan Physical memory for _TCPT_OBJECT objects (tcp connections)
        consoles        Extract command history by scanning for _CONSOLE_INFORMATION
        crashinfo       Dump crash-dump information
        deskscan        Poolscaner for tagDESKTOP (desktops)
        devicetree      Show device tree
        dlldump         Dump DLLs from a process address space
        dlllist         Print list of loaded dlls for each process
        driverirp       Driver IRP hook detection
        driverscan      Scan for driver objects _DRIVER_OBJECT
        dumpcerts       Dump RSA private and public SSL keys
        envars          Display process environment variables
        eventhooks      Print details on windows event hooks
        evtlogs         Extract Windows Event Logs (XP/2003 only)
        filescan        Scan Physical memory for _FILE_OBJECT pool allocations
```

Figure 3. Volatility Plugins

Figure 3 is a portion of the plugins available for use when the WindowsXPSP2x86 profile specified. Plugins are how Volatility accomplishes various tasks: listing processes, finding active network connections, or accessing registry keys in memory, just to name a few.

If you are using Volatility to analyze a Windows XP memory image, then you could use the `connections` plugin to list all open (active) connections that existed at the time of the memory acquisition. If you wanted to see traces of connections that might have already closed but still exist in memory, then use `connscan`.

If the image were taken from a Windows 7 system then `connections` and `connscan` would not appear in the list. Instead you would use the `netscan` plugin, which returns similar results.

To see available plugins for analyzing a 32-bit Windows 7 Service Pack 0 memory image, use

```
volatility.exe --profile=Win7SP0x86 -h
```

For a more detailed description of the plugins, see [2].

Volatility Plugin Options

```
-h, --help            list all available options and their default values.
                      Default values may be set in the configuration file
                      (/etc/volatilityrc)
--conf-file=.volatilityrc
                      User based configuration file
-d, --debug           Debug volatility
--plugins=PLUGINS     Additional plugin directories to use (semi-colon
                      separated)
--info                Print information about all registered objects
--cache-directory=C:\Users\Dae/.cache\volatility
                      Directory where cache files are stored
--cache               Use caching
--tz=<volatility.timefmt.OffsetTzInfo object at 0x040121D0>
                      Sets the timezone for displaying timestamps
-f FILENAME, --filename=FILENAME
                      Filename to use when opening an image
--profile=Win7SP1x86  Name of the profile to load
-l file:///C:/tools/WIN-KS7VOUQA69E-20130701-154352.raw, --location=file:/
                      A URN location from which to load an address space
-w, --write           Enable write support
--dtb=DTB             DTB Address
--cache-dtb           Cache virtual to physical mappings
--output=text         Output in this format (format support is module
                      specific)
--output-file=OUTPUT_FILE
                      write output in this file
-v, --verbose         Verbose information
--shift=SHIFT         Mac KASLR shift address
-g KDBG, --kdbg=KDBG  Specify a specific KDBG virtual address
-k KPCR, --kpcr=KPCR  Specify a specific KPCR address
-o OFFSET, --offset=OFFSET
                      EPROCESS offset (in hex) in the physical address spa
-p PID, --pid=PID     Operate on these Process IDs (comma-separated)
-P, --physical-offset
                      Physical Offset
-t OBJECT_TYPE, --object-type=OBJECT_TYPE
                      Show these object types (comma-separated)
-s, --silent          Suppress less meaningful results
```

Figure 4. Volatility Plugin Options

Plugins have their own set of options in addition to the general volatility options. The listing above shows the general options first (`-f, --tz, ,-v, etc..`). `-p` and `--pid=` are the first of the plugin specific options for the `handles` plugin.

In order to display the plugin specific help options the general form is:

```
volatility.exe PLUGIN -h
```

Where PLUGIN is whichever plugin you choose.
This document will cover a few plugins. Consult the Volatility CheatSheet [3] and your local help file for further information.

Volatility Plugin – imageinfo

```
     Suggested Profile(s) : Win7SP0x86, Win7SP1x86
                AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                AS Layer2 : FileAddressSpace (C:\tools\my_memory_dump.raw)
                 PAE type : PAE
                      DTB : 0x185000L
                     KDBG : 0x8296fc28L
     Number of Processors : 1
Image Type (Service Pack) : 1
           KPCR for CPU 0 : 0x82970c00L
        KUSER_SHARED_DATA : 0xffdf0000L
      Image date and time : 2013-07-01 15:43:54 UTC+0000
Image local date and time : 2013-07-01 10:43:54 -0500
```

Figure 4. Volatility Plugin - imageinfo

Figure 4 is the result of running `volatility.exe -f my_memory_dump.raw imageinfo`

The suggested profiles lists the profiles Volatility thinks will work for analyzing this memory image, in this case Win7SP0x86 or Win7SP1x86. The "Image local date and time" line contains the local time of acquisition. This one was acquired at UTC -0500 (CST/CDT).

Volatility Plugin – pslist

```
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
---------- -------------------- ------ ------ ------ -------- ------ ------ -------------------
<snip>
0x85dddd40 explorer.exe           1412   1284     65     1401      1      0 2013-06-08 21:09:28
<snip>
0x84e0f060 notepad++.exe          2936   1412      0 --------      1      0 2013-06-28 15:36:45
0x862ce1f8 notepad++.exe           672   1412      0 --------      1      0 2013-06-28 16:06:47
0x85c5cad8 cmd.exe                 460   1412      1       93      1      0 2013-06-28 21:05:11
0x84aef030 conhost.exe            3400    396      2       50      1      0 2013-06-28 21:05:11
0x861ff3f8 SearchProtocol         2456   2120      6      318      0      0 2013-07-01 10:42:38
0x86203490 SearchFilterHo         3740   2120      5      120      0      0 2013-07-01 10:42:38
0x84ca7d40 audiodg.exe            3692    740      7      143      0      0 2013-07-01 10:42:55
0x85cc19b8 wmplayer.exe           1880   1412     37      745      1      0 2013-07-01 10:43:37
0x86106030 DumpIt.exe             1128   1412      2       37      1      0 2013-07-01 10:43:52
0x84b81978 conhost.exe            2240    396      2       49      1      0 2013-07-01 10:43:52
```

Figure 5. Volatility Plugin - pslist

Figure 5 is the result of running `volatility.exe -f my_memory_dump.raw --profile=Win7SP1x86 pslist`

The volatility plugin `pslist` lists the processes that were running at the time of acquisition. There are other plugins for listing processes as well: `psscan`, `psxview`, and `pstree`

Volatility Plugin – handles

Offset(V)	Pid	Handle	Access	Type	Details
<snip>					
0x84bd8188	1880	0x868	0x120089	File	\Device\HarddiskVolume1\Users\Public\Music\Sample Music\Kalimba.mp3
0x84d90520	1880	0x884	0x100001	File	\Device\KsecDD
0x860ae4f8	1880	0x8f4	0x12019f	File	\Device\NamedPipe\samr
0x8603f440	1880	0x970	0x16019f	File	\Device\Afd\Endpoint
0x86287f80	1880	0x974	0x16019f	File	\Device\Afd\Endpoint
<snip>					

Figure 6. Volatility Plugin - handles

Figure 6 is the result of running `volatility.exe -f my_memory_dump.raw --profile=Win7SP1x86 handles -p 1880 -t File`

The volatility plugin `handles` lists the handles a process (or processes) has open. In this case, `handles` was used with two additional options: `-p` and `-t`

`-p` is used to specify the Pid of the process to run handles against. `-t` specifies the type of handles. In this case, volatility only returns handles of type File for the process with id 1880.

Volatility Plugin – netscan

Offset(P)	Proto	Local Address	Foreign Address	State	Pid	Owner
<snip>						
0x83badf8	TCPv4	192.168.102.138:49200	157.56.59.250:80	CLOSED	1880	wmplayer.exe
0x7aa74df8	TCPv4	192.168.102.138:49179	65.55.87.65:80	CLOSED	1880	wmplayer.exe
0x7e481c28	TCPv4	192.168.102.138:49208	65.55.87.65:80	CLOSED	1880	wmplayer.exe
0x7e65c8c0	TCPv4	192.168.102.138:49178	65.55.87.65:80	CLOSED	1880	wmplayer.exe
0x7e6b2888	TCPv4	192.168.102.138:49176	72.18.206.203:80	ESTABLISHED	1880	wmplayer.exe
0x7fa00538	TCPv4	192.168.102.138:49206	65.55.87.65:80	CLOSED	1880	wmplayer.exe
0x7fd1edf8	TCPv4	192.168.102.138:49175	54.230.90.178:80	ESTABLISHED	1880	wmplayer.exe
0x7fda0630	TCPv4	192.168.102.138:49207	157.56.59.250:80	CLOSED	1880	wmplayer.exe
<snip>						

Figure 7. Volatility Plugin - netscan

Figure 6 is the result of running `volatility.exe -f my_memory_dump.raw --profile=Win7SP1x86 netscan`

The volatility plugin `netscan` lists the active and recently closed network connections in a memory sample.

References

[1] https://code.google.com/p/volatility/
[2] https://code.google.com/p/volatility/wiki/CommandReference23adsf

[3] https://volatility.googlecode.com/files/CheatSheet_v2.3.nothing