Introduction to Memory forensics

Cristina Bianca Niculescu

Objectives:

Understanding of digital forensics

Understanding of volatile data and which tools are used to capture volatile data

Understanding of tools that can or can not analyze volatile data

Analyzing data with Volatility

What is Forensics?

Finding and analyzing evidence -> fundamentals of forensic sciences

Digital Forensics -> every piece of evidence data that is stored on an electronic device (PC, phones, TVs, smart WC, smart Fridges, smart coffee makers, credit cards etc.)

- Recovery of digital data
- Finding evidence of badness
- Undeleting and recomposing pieces of data (for example if antiforensics tools were used)
- Plaintext logs, network traffic, file system metadata, full-system memory dumps

Memory Forensics -> volatile storage of digital data that can be pulled out and analyzed

Memory Forensics

Today we talk about memory capture and analysis on a Windows system

Memory contains the present state of a device, a live system

We capture the memory of a device to get a snapshot of the actual data that is stored at the moment of capturing (note: everything in memory runs unencrypted!)

Like freezing a system in a time capsule!

Memory forensic

We can examine the memory dumps for a lot of essential data:

- What accounts were logged on at the time of the capture
- What processes were running and where they are placed (see SANS cheat sheet)
- When and what files were accessed
- What network connections were made
- Are there encryption keys for encrypted storages
- Are there any passwords or credentials used
- Or any other information that can be pulled out for further analysis (malware)

Memory forensic tools

Capturing memory from any live digital system can be a pain sometimes and it feels tricky. It’s Volatile!

Therefore knowing how to use few tools to acquire the capture is essential

What do you need to know first:

File system of the Windows machine (FAT or NTFS or exFAT) and the size of the RAM you will acquire

The USB storage you will be capturing the data on to needs to be NTFS or exFAT formatted, and cleaned previously before acquiring any type of data (examination storage needs to exceed the size of actual captured data, because capture is almost all the time raw format - no compression)

Prepare your tools and prepare your examination storages every time before you acquire data!

Memory forensics professional capturing tools

Dirty tools that leave traces of usage onto memory:

- DUMPIT
- FTK Imager lite edition
- EnCase Imager

Clean tools that leave memory untouched:

- DMA (direct memory access) the FIREWIRE ATTACK (INCEPTION)
- COLDBOOT (freezing the memory? what?) ram2usb https://www.youtube.com/watch?v=TQP2IMnPw9c

*Why untouching memory is important?

In very high profile criminal cases (terorism, spying etc.) recovery of the encryption keys is the most important step to gain access to encrypted containers which have the essential data for saving lives, and not overwriting the memory while capturing. (Bottom line is choose your tools carefully and practice with them before the live usage! (ref link: https://www.ethicalhacker.net/features/root/using-cold-boot-attacks-forensic-techniques-penetration-tests/)

Memory forensic basic specifications

What is a PID, a PPID and why is this very important in many digital forensic analysis?

PID is the process ID number that is given to a running process inside an ongoing operating system

PPID is the parent process ID from where another running process might be spawned (legitimate relationship or rogue)

The process tree always gives us clues of rogue processes that are causing a bad behaviour (mostly malware)

Ref links: http://dfir.com.br/wp-content/uploads/2014/09/poster_2014_find_evil.pdf

https://sansorg.egnyte.com/dl/LVvF5jRNLK

Memory forensic - Analysis tools

Briefly, best free professional tools for analysis are:

Volatility - command line (most preferred)

Red Line - Mandiant GUI tool

Rekall - command line and GUI (fork of Volatility from Google rapid response)

(Linux command line commands: hexeditor, strings, grep, foremost etc.)

Costly professional tools:

EnCase 8, FTK, Xways, Magnet Forensics, Nuix

Memory forensic - Analysis downsides

Not fancy structured data like OS file systems

Hard to identify what are you looking for in a maze of strings

Manual trying to carve pieces of objects like documents, pictures, other type of files feels like you are lost in a haystack and you don’t know what to pick and where to start from because we do not save our files entirely into memory (but we can find for sure their metadata (MFT))

Mostly you wanna get contiguous blobs of data, but nothing can be perfect!

Linux command line: less, strings, binwalk, foremost, scalpel etc.

How about bulkextractor and grep?

Memory forensic - Analysis with Volatility

The Art of memory forensics https://www.elefant.ro/the-art-of-memory-forensics-detecting-malware-and-threats-in-windows-linux-and-mac-memory-paperback_251fb2d0-cf61-4a2d-8474-b1e1b4c80bac?gclid=Cj0KCQiAweaNBhDEARIsAJ5hwbdLGpEVoW9SwGEt7MNIrRVMkvEUVcUGTU-xzFrNYvjY6OcBVCjqLyUaAhmuEALw_wcB

Volatility project https://github.com/volatilityfoundation/volatility

Memory forensic - Analysis with Volatility

Memory dump analysis

We'll be analyzing the memory dump file (challenge.raw) using **Volatility 2.6.1** as it is best suited to our needs.

The very first thing that anyone needs to know before proceeding to forensic analysis of a memory dump is to determine the OS **profile** we are going to use.

The profile tells us the OS of the system or computer from which the dump was extracted. Volatility has a built-in plugin to help us determine the profile of the dump

Now, we'll be using the **imageinfo** plugin

$ volatility -f Challenge.raw imageinfo

Memory forensic - Analysis with Volatility

```
sansforensics@siftworkstation: ~/win10_volatility
$ python vol.py -f ../malware_course/Challenge.raw imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86_24000, Win7SP1x86
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/sansforensics/malware_course/Challenge.raw)
                      PAE type : PAE
                           DTB : 0x185000L
                          KDBG : 0x8273cb78L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0x80b96000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2018-10-23 08:30:51 UTC+0000
     Image local date and time : 2018-10-23 14:00:51 +0530
```

Memory forensic - Analysis with Volatility

Now as you can see, Volatility provides a lot of suggestions as to which profile you should use.

In some cases, all of the suggested profiles may not be correct.

To help get over this barrier, you may use another plugin called **kdbgscan**.

As far as this challenge is concerned, using kdbgscan (kernel debugger) isn't required.

Now as a forensic analyst, one of the most important things we would like to know from a system during analysis would be:

- Active processes
- Commands executed in the shell/terminal/Command prompt
- Hidden processes (if any) or Exited processes
- Browser History
- Malware (if any) or suspicious activity

And many more as the analysis is going towards narrowing down the important stuffz!

Memory forensic - Analysis with Volatility

Now, to list the active or running processes, we use the help of the plugin **pslist**.

$ volatility -f Challenge.raw --profile=Win7SP1x86 pslist

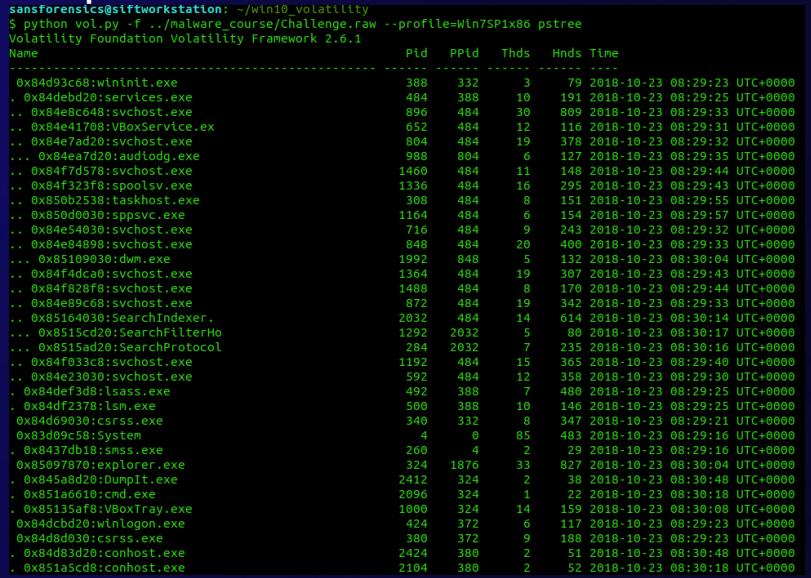
```
sansforensics@siftworkstation: ~/win10_volatility
$ python vol.py -f ../malware_course/Challenge.raw --profile=Win7SP1x86 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ----------
0x83d09c58 System                    4      0     85      483 ------      0 2018-10-23 08:29:16 UTC+0000
0x8437db18 smss.exe                260      4      2       29 ------      0 2018-10-23 08:29:16 UTC+0000
0x84d69030 csrss.exe               340    332      8      347      0      0 2018-10-23 08:29:21 UTC+0000
0x84d8d030 csrss.exe               380    372      9      188      1      0 2018-10-23 08:29:23 UTC+0000
0x84d93c68 wininit.exe             388    332      3       79      0      0 2018-10-23 08:29:23 UTC+0000
0x84dcbd20 winlogon.exe            424    372      6      117      1      0 2018-10-23 08:29:23 UTC+0000
0x84debd20 services.exe            484    388     10      191      0      0 2018-10-23 08:29:25 UTC+0000
0x84def3d8 lsass.exe               492    388      7      480      0      0 2018-10-23 08:29:25 UTC+0000
0x84df2378 lsm.exe                 500    388     10      146      0      0 2018-10-23 08:29:25 UTC+0000
0x84e23030 svchost.exe             592    484     12      358      0      0 2018-10-23 08:29:30 UTC+0000
0x84e41708 VBoxService.ex          652    484     12      116      0      0 2018-10-23 08:29:31 UTC+0000
0x84e54030 svchost.exe             716    484      9      243      0      0 2018-10-23 08:29:32 UTC+0000
0x84e7ad20 svchost.exe             804    484     19      378      0      0 2018-10-23 08:29:32 UTC+0000
0x84e84898 svchost.exe             848    484     20      400      0      0 2018-10-23 08:29:33 UTC+0000
0x84e89c68 svchost.exe             872    484     19      342      0      0 2018-10-23 08:29:33 UTC+0000
0x84e8c648 svchost.exe             896    484     30      809      0      0 2018-10-23 08:29:33 UTC+0000
0x84ea7d20 audiodg.exe             988    804      6      127      0      0 2018-10-23 08:29:35 UTC+0000
0x84f033c8 svchost.exe            1192    484     15      365      0      0 2018-10-23 08:29:40 UTC+0000
0x84f323f8 spoolsv.exe            1336    484     16      295      0      0 2018-10-23 08:29:43 UTC+0000
0x84f4dca0 svchost.exe            1364    484     19      307      0      0 2018-10-23 08:29:43 UTC+0000
0x84f7d578 svchost.exe            1460    484     11      148      0      0 2018-10-23 08:29:44 UTC+0000
0x84f828f8 svchost.exe            1488    484      8      170      0      0 2018-10-23 08:29:44 UTC+0000
0x850b2538 taskhost.exe            308    484      8      151      1      0 2018-10-23 08:29:55 UTC+0000
0x850d0030 sppsvc.exe             1164    484      6      154      0      0 2018-10-23 08:29:57 UTC+0000
0x85109030 dwm.exe                1992    848      5      132      1      0 2018-10-23 08:30:04 UTC+0000
0x85097870 explorer.exe            324   1876     33      827      1      0 2018-10-23 08:30:04 UTC+0000
0x85135af8 VBoxTray.exe           1000    324     14      159      1      0 2018-10-23 08:30:08 UTC+0000
0x85164030 SearchIndexer.         2032    484     14      614      0      0 2018-10-23 08:30:14 UTC+0000
0x8515ad20 SearchProtocol          284   2032      7      235      0      0 2018-10-23 08:30:16 UTC+0000
0x8515cd20 SearchFilterHo         1292   2032      5       80      0      0 2018-10-23 08:30:17 UTC+0000
0x851a6610 cmd.exe                2096    324      1       22      1      0 2018-10-23 08:30:18 UTC+0000
0x851a5cd8 conhost.exe            2104    380      2       52      1      0 2018-10-23 08:30:18 UTC+0000
0x845a8d20 DumpIt.exe             2412    324      2       38      1      0 2018-10-23 08:30:48 UTC+0000
0x84d83d20 conhost.exe            2424    380      2       51      1      0 2018-10-23 08:30:48 UTC+0000
```

Memory forensic - Analysis with Volatility

And lastly, we have to establish the parents of the processes, and check for suspicious processes:

$ volatility -f Challenge.raw --profile=Win7SP1x86 pstree

Let’s try those in the Laboratory!



Memory forensic - Analysis with Volatility

Exercise

Basic Forensic Win7mem.raw

Find:

Last modified time of "loveyou.png"

Physical offset of "loveletter"

Is there a "mega" link that was accessed?

Memory forensic - Analysis with Volatility

Challenge 1

My friend John is an "environmental" activist and a humanitarian. He hated the ideology of Thanos from the Avengers: Infinity War. He sucks at programming. He used too many variables while writing any program. One day, John gave me a memory dump and asked me to find out what he was doing while he took the dump. Can you figure it out for me?

CLUES: Environmental Activist (Since the word is quoted)

John hates Thanos (Maybe useless but let us see)

John sucks at programming and used too many variables.

Challenge 2

My sister's computer crashed. We were very fortunate to recover this memory dump. Your job is get all her important files from the system. From what we remember, we suddenly saw a black window pop up with some thing being executed. When the crash happened, she was trying to draw something. Thats all we remember from the time of crash.

CLUES:

computer crash

important files to recover

black window pop up (command line?)

date and time of a drawing application will identify around that time the process of black window pop?