FORENSIC

Legal Informatics and Multimedia

Forensic Analysis for Computer Systems

Plan of the Labs:

1. Electronic data acquisition
2. Computer Forensics technics and tools
3. Volatility Framework
4. Autopsy (**Sleuthkit**)
5. FTK (Forensic ToolKit)
6. Guidance software EnCase & ProDiscover Forensic

Course Lab3: Volatility Framework

3.1 Introduction

3.2 Memory Format Support

3.3 Installing Volatility

3.4 Volatility commands for Windows profile

3.5 Important online resources

3.1 Introduction

- ✓ Memory forensics is the process of capturing the running memory of a device and then analyzing the captured output for evidence of malicious software.

- ✓ Unlike hard-disk forensics where the file system of a device is cloned and every file on the disk can be recovered and analyzed, memory forensics focuses on the actual programs that were running on a device when the memory dump was captured.

- ✓ When an application is opened on a device, a certain amount of RAM will be allocated to that application so it can run, if applications are opened then more RAM will need to be allocated by the device.

3.1 Introduction

→ Thus one of the important parts of Malware analysis is Random Access Memory (RAM) analysis within the compromised system.

→ It helps to identify the running malicious processes, network activities, open connections etc.

→ "Volatility" is a security tool for volatile memory analysis.

→ It can be used for both 32/64 bit systems RAM analysis and it supports analysis of Windows, Linux, Mac & Android systems.

→ The Volatility Framework is implemented in Python scripting language and it can be easily used on Linux and Windows operating systems. It is used to analyze crash dumps, raw dumps, VMware & VirtualBox dumps.

3.1 Introduction

What is a Memory Dump?

→ A memory dump or RAM dump is a snapshot of memory that has been captured for memory analysis. When a RAM dump is captured it will contain data relating to any running processes at the time the capture was taken.

→ The analysis techniques are performed completely independent of the system being investigated and give complete visibility into the runtime state of the system.

→ When responding to a cybersecurity incident the memory forensics is the first step we can adopt. By capturing the memory of a compromised device we can quickly perform some analysis to identify potential malware and gather IOC’s (Indicators of Compromise) which can then be used to identify other compromised devices.

3.2 Memory Format Support

The following memory format is supported by the latest Volatility release:

- Raw/Padded Physical Memory
- Expert Witness Format from ASR Data
- 32- and 64-bit Windows Crash Dump
- 32- and 64-bit Mac files
- 32- and 64-bit Linux
- Virtualbox Core Dumps
- VMware Saved State (.vmss) and Snapshot (.vmsn)
- HPAK Format (FastDump, HBGray Software)
- QEMU memory dumps

3.3 Installing Volatility (Windows)

The Volatility tool is available for Windows, Linux and Mac operating systems.

For Windows and Mac OSes, standalone executables are available and it can be used :

1. Download the executable file (according to the OS installed) from : http://www.volatilityfoundation.org/ For example : volatility_2.6_win64_standalone.exe for Win7.
2. Copy the standalone file in a drive (e.g., C:\ or another one) and the Volatility framework is ready to use!
3. In the next step we use Windows prompt (CMD) to start analysis operation using this line:

```
"C:\Volatility_2.6_win64_standalone  imageinfo  -f   dump_file"
```

Where "dump_file" is the dumped memory file which can have .raw or .mem extension.

3.3 Installing Volatility (Linux)

On Linux- Ubuntu we can install Volatility using following steps:

Step 1: **Update system packages** (This will update the list of newest versions of packages and its dependencies on our Linux system).

```
~$ sudo apt-get update
```

Step 2: **Installing Volatility***

```
~$ sudo apt-get install volatility
```

3.3 Installing Volatility (Linux)

Step 3: **Using Volatility**

To use volatility we should mount to it’s installation directory using a Terminal and enter the following command:

“python vol.py imageinfo -f dump_file“

* If Volatility is not installed then `dpkg -L volatility` will give the following error:

```
~$ dpkg -L volatility
dpkg-query: package 'volatility' is not installed
Use dpkg --info (= dpkg-deb --info) to examine archive files,
and dpkg --contents (= dpkg-deb --contents) to list their contents.
```

3.4 Volatility commands for Windows profile

`volatility -h` : This command displays volatility available options

`imageinfo` : This command allows to recuperate image profile

Example:

```
$ volatility -f memdump.mem imageinfo
Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86
AS Layer1 : IA32PagedMemoryPae (Kernel AS)
AS Layer2 : FileAddressSpace (/home/memdump.mem)
PAE type : PAE
DTB : 0x185000L
KDBG : 0x82b39c28L
Number of Processors : 1
Image Type (Service Pack) : 1
KPCR for CPU 0 : 0x82b3ac00L
KUSER_SHARED_DATA : 0xffdf0000L
Image date and time : 2019-07-31 13:45:40 UTC+0000
Image local date and time : 2019-07-31 15:45:40 +0200
```

Profiles (OS versions) are used for the dump analysis with “--profile=“ option

3.4 Volatility commands for Windows profile (Cont.)

`pslist` : This command displays processes list contained in the dump

Example:

```
$ volatility -f memdump.mem --profile=Win7SP1x86 pslist
Offset(V)  Name                    PID   PPID   Thds   Hnds  Sess   Wow64  Start                    Exit

---------- -------------------- ---- ----   ---- -------- ------ ------ -------------------------

0x848338e8 System                  4  0      90     547            0   2019-07-31 11:49:59 UTC+0000
0x87002020 smss.exe              272  4      2      29             0   2019-07-31 11:49:59 UTC+0000
0x8673f030 csrss.exe             360  352    9      381   0        0   2019-07-31 11:50:00 UTC+0000
0x867f7030 wininit.exe           412  352    3      74    0        0   2019-07-31 11:50:01 UTC+0000
0x867f55f8 csrss.exe             420  404    11     388   1        0   2019-07-31 11:50:01 UTC+0000
0x86879d40 winlogon.exe          468  404    3      110   1        0   2019-07-31 11:50:01 UTC+0000
0x84e19030 rad5163B.tmp.exe     1416  3544   11     472   1        0    2019-07-31 12:45:47 UTC+0000
```

Where :

`offset` : Process memory address ; `name` : Process name; `PID` : Process Identification; `PPID` : Parent Process ID ; `start` : Launching date and hour of process.

→ Note that process name is not sufficient to say that process is legal or not but it's a first sign

3.4 Volatility commands for Windows profile (Cont.)

`pstree` : This command displays processes list in tree format (Parent/child). This will show the linking of the process with the parent process. It will help us to identify the parent process of the malicious program.

Example:

```
$ volatility -f memdump.mem --profile=Win7SP1x86 pstree
```

`dlllist` : This command lists DLL (Dynamic Link Library) functions contained in the dump, which are available on the Windows API and used to manipulate DATA, achieve network connections and files creation.

Example:

```
$ volatility -f memdump.mem --profile=Win7SP1x86 dlllist  -p 1416
```

3.4 Volatility commands for Windows profile (Cont.)

`hivelist` : This command extracts register information with the corresponding files, Windows register contains several parameters and configurations of OS, it we allows to determine recent executed programs, extract pass words hashes and analyze Keys and introduced values by a malware code in the OS.

Example:

```
$ volatility -f memdump.mem --profile=Win7SP1x86 hivelist
Virtual     Physical    Name
---------- ----------  ----
0x95281008 0x675df008
\??\C:\Users\johnoc\AppData\Local\Microsoft\Windows\UsrClass.dat
0x95289008 0x6c12a008  \??\C:\Users\johnoc\ntuser.dat

[...]
```

3.4 Volatility commands for Windows profile (Cont.)

`hashdump` : This command allows us to display dumped pass words hashes for Windows accounts

Example:

```
$ volatility -f memdump.mem --profile=Win7SP1x86 hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
johnoc:1000:aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c:::
```

3.4 Volatility commands for Windows profile (Cont.)

`netscan` : This command displays active network connections

Example:

```
$ volatility -f memdump.mem --profile=Win7SP1x86 netscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P) Proto Local Address Foreign Address State Pid Owner Created
0x19472208 UDPv4 0.0.0.0:3702 *:* 1488 svchost.exe 2019-07-31 12:43:12 UTC+0000
0x19472208 UDPv6 :::3702 *:* 1488 svchost.exe 2019-07-31 12:43:12 UTC+0000
0x24d773a0 UDPv4 0.0.0.0:5355 *:* 1240 svchost.exe 2019-07-31 13:43:08 UTC+0000
0x5d30e208 UDPv4 0.0.0.0:3702 *:* 1488 svchost.exe 2019-07-31 12:43:12 UTC+0000
0x5d30e208 UDPv6 :::3702 *:* 1488 svchost.exe 2019-07-31 12:43:12 UTC+0000
0x6ec043a0 UDPv4 0.0.0.0:5355 *:* 1240 svchost.exe 2019-07-31 13:43:08 UTC+0000
0x7beddf50 UDPv6 ::1:1900 *:* 1488 svchost.exe 2019-07-31 12:43:10 UTC+0000
0x7ce3fe20 UDPv6 fe80::94b5:ad60:33d0:3773:1900 *:* 1488 svchost.exe 2019-07-31
12:43:10 UTC+0000
0x7d0028d8 UDPv4 0.0.0.0:0 *:* 948 svchost.exe 2019-07-31 11:50:02 UTC+0000
```

3.4 Volatility commands for Windows profile (Cont.)

Where :

- Offset : Location in memory
- Proto : Network protocol used by process
- LocalAddr : Source address of network connection
- LocalPort : Source port of network connection
- ForeignAddr : Destination address of network connection
- ForeignPort : Destination address of network connection
- State : State of network connection i.e. established, closed or listening
- PID : Process ID of associated process
- Owner : Account associated with process
- Created : Time network connection has initiated

3.4 Volatility commands for Windows profile (Cont.)

`malfind` : This command displays a list of processes that Volatility suspects may contain injected code based on the header information displayed in hex, the permissions, and some extracted assembly code, just because a process is listed it doesn't mean the process is 100% malware.

Example:

```
$ volatility -f memdump.mem --profile=Win7SP1x86 malfind
```

Example: In this case malfind is used to detect malicious DLL's in the process PID = 1416 and saves them in output directory (helps to dump the malicious process and analyze it)

```
$ volatility.exe --profile=Win7SP1x86 malfind -D E:\output -p 1416 -f memdump.mem
```

3.4 Volatility commands for Windows profile (Cont.)

`dumpfiles` : This command extracts a determined process with it’s all DLL files and saves them into a specified directory.

Example:

```
$ volatility_ -f memdump.mem --profile=Win7SP1x86 dumpfiles -p 1416 –D  directory_path
```

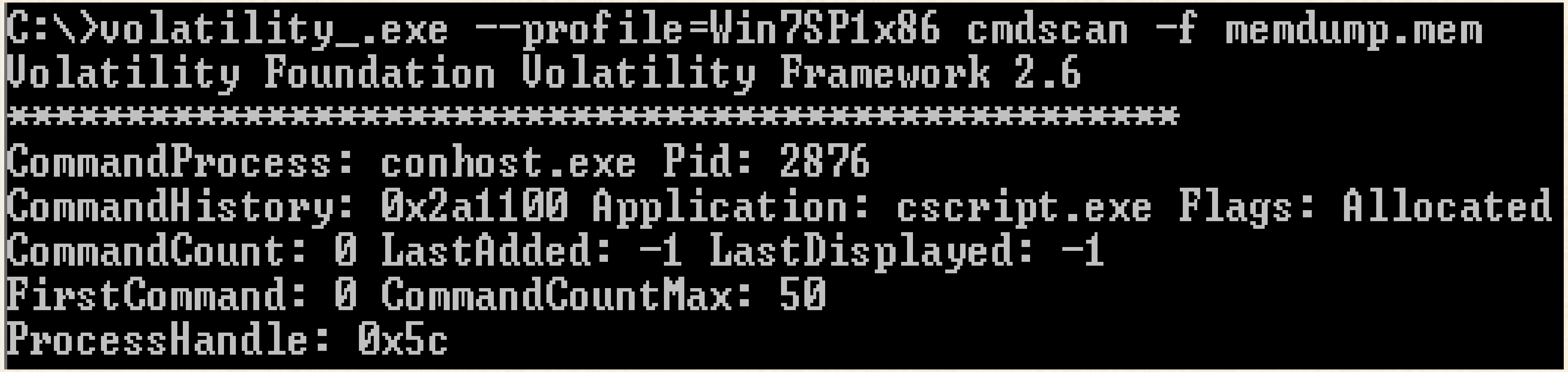
`procdump` : it works like dumpfiles.

```
$ volatility -f memdump.mem --profile=Win7SP1x86 procdump -D dir/ -p 1416
```

3.4 Volatility commands for Windows profile (Cont.)

cmdscan: Used to list the last commands on the compromised machine

```
$ volatility -f memdump.mem --profile=Win7SP1x86 cmdscan
```

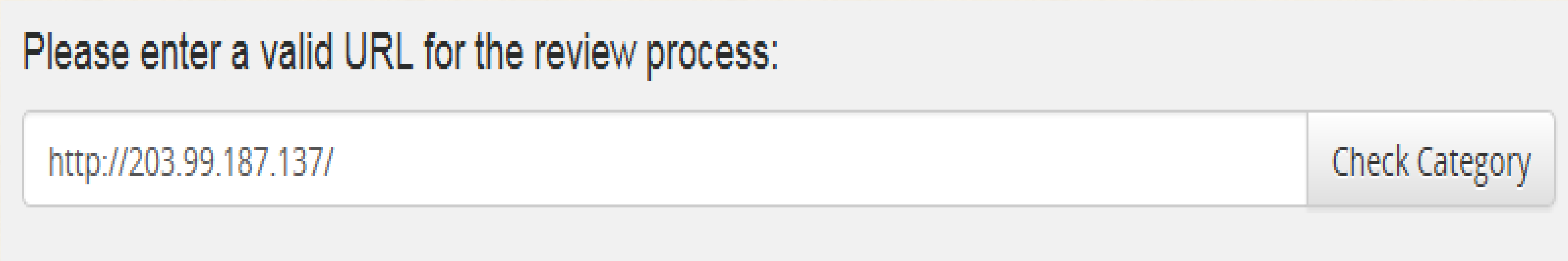


```
C:\>volatility_.exe --profile=Win7SP1x86 cmdscan -f memdump.mem
Volatility Foundation Volatility Framework 2.6
**************************************************
CommandProcess: conhost.exe Pid: 2876
CommandHistory: 0x2a1100 Application: cscript.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
```

3.5 Important online resources

In a forensic investigation scenario, online resources such "**virustotal**" and "**payload security**" website will be used to verify the results. In the end, Windows Defender and Malware Bytes can be used to scan the malicious programs.
For verifying malicious connections we can use :

http://sitereview.symantec.com/#/


Please enter a valid URL for the review process:

http://203.99.187.137/

For more plugins see: https://github.com/volatilityfoundation/volatility/