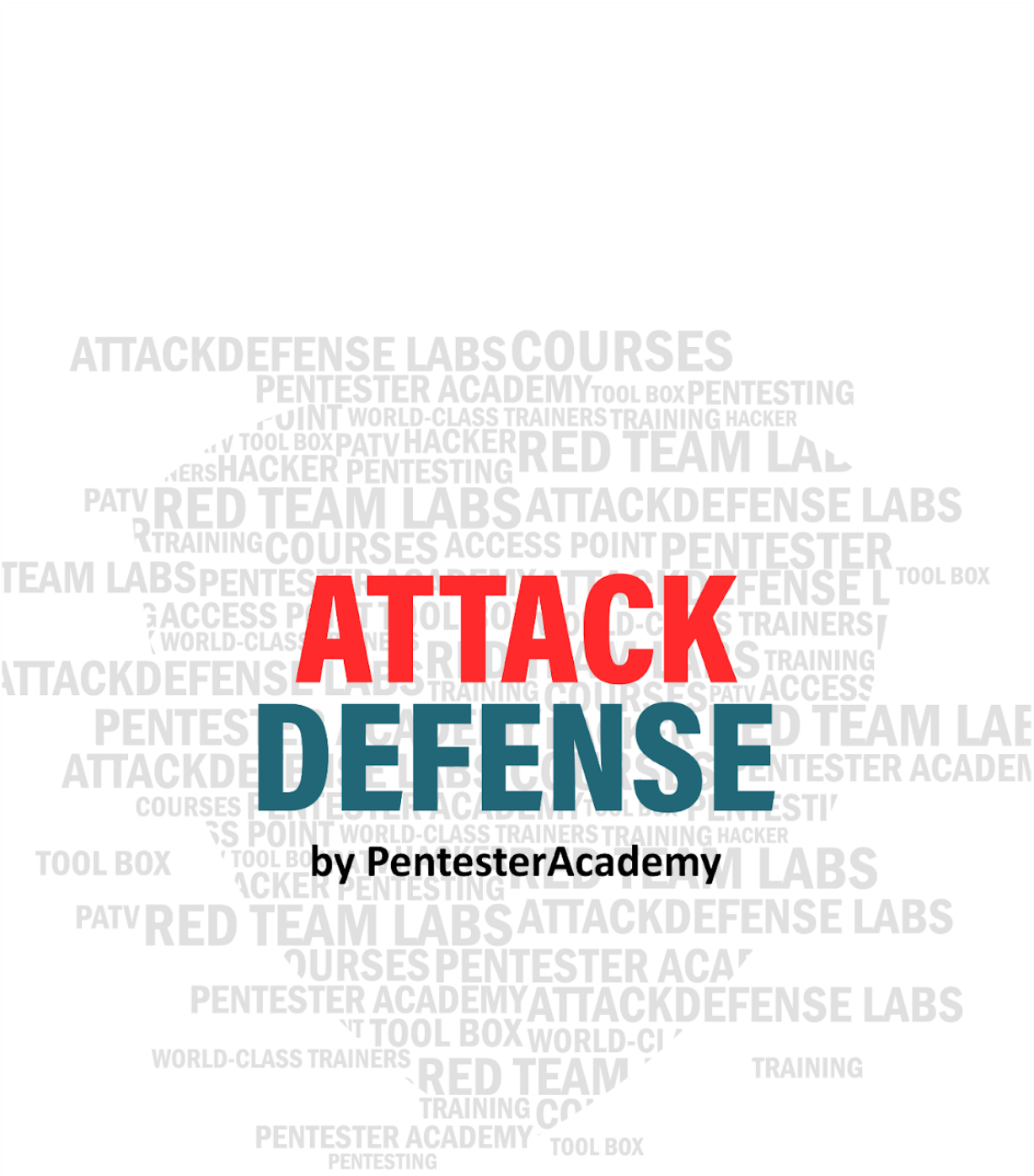
ATTACK
DEFENSE
by PentesterAcademy

Name	Volatility: Binary I
URL	https://attackdefense.com/challengedetails?cid=1130
Type	Forensics: Memory Forensics

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

A suspicious program named after a popular website is executed from a non-standard directory.. The memory dump of that machine is given to you. You have to use Volatility to analyze the memory dump and answer the following questions:

Q1. What is the name of the executed binary?

Answer: YoutubeDownloader.exe

Command: vol.py -f memory_dump.mem --profile=Win81U1x64 consoles

```
CommandHistory: 0x1c8d7f5be0 Application: YoutubeDownloader.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x1c8d7c6920
----
CommandHistory: 0x1c8d7f5a00 Application: YoutubeDownloader.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x1c8d7c6500
----
```

Q2. What is the name of the directory the binary was kept in?

Answer: Downloads

Command: vol.py -f memory_dump.mem --profile=Win81U1x64 userassist

```
REG_BINARY    C:\Users\IEUser\Downloads\YoutubeDownloader.exe :
Count:          2
Focus Count:    3
Time Focused:   0:00:32.140000
Last updated:   2019-06-29 18:30:44 UTC+0000
Raw Data:
0x00000000  00 00 00 00 02 00 00 00 03 00 00 00 98 7b 00 00   .............{..
0x00000010  00 00 80 bf 00 00 80 bf 00 00 80 bf 00 00 80 bf   ................
0x00000020  00 00 80 bf 00 00 80 bf 00 00 80 bf 00 00 80 bf   ................
0x00000030  00 00 80 bf 00 00 80 bf ff ff ff ff c0 cf 80 c3   ................
0x00000040  a8 2e d5 01 00 00 00 00                           ........
---------------------------
```

Q3. How many times it was executed by the user?

Answer: 2

Command: vol.py -f memory_dump.mem --profile=Win81U1x64 userassist

```
REG_BINARY    C:\Users\IEUser\Downloads\YoutubeDownloader.exe :
Count:          2
Focus Count:    3
Time Focused:   0:00:32.140000
Last updated:   2019-06-29 18:30:44 UTC+0000
Raw Data:
0x00000000  00 00 00 00 02 00 00 00 03 00 00 00 98 7b 00 00   .............{..
0x00000010  00 00 80 bf 00 00 80 bf 00 00 80 bf 00 00 80 bf   ................
0x00000020  00 00 80 bf 00 00 80 bf 00 00 80 bf 00 00 80 bf   ................
0x00000030  00 00 80 bf 00 00 80 bf ff ff ff ff c0 cf 80 c3   ................
0x00000040  a8 2e d5 01 00 00 00 00                           ........
---------------------------
```

Q4. How many handles are opened by the processes of this binary in total (exclude conhost.exe)?

Answer: 80

Solution:

Get the PID of the process from the process list

Command: vol.py -f memory_dump.mem --profile=Win81U1x64 pslist | grep Youtube

```
root@attackdefense:~# vol.py -f memory_dump.mem --profile=Win81U1x64 pslist | grep Youtube
Volatility Foundation Volatility Framework 2.6.1
0xffffe000403c92c0 YoutubeD              3744   1580      1        0      1      0 2019-06-29 18:30:44 UTC+0000

0xffffe00040493080 YoutubeD               680   3744      1        0      1      0 2019-06-29 18:30:44 UTC+0000

root@attackdefense:~#
```

Extract the binary from the process using PID

Command: vol.py -f memory_dump.mem --profile=Win81U1x64 handles -p 680

```
root@attackdefense:~# vol.py -f memory_dump.mem --profile=Win81U1x64 handles -p 680
Volatility Foundation Volatility Framework 2.6.1
Offset(V)          Pid             Handle             Access Type            Details
------------------ ------ ------------------ ---------- --------------- -------
0xffffe00041dc7490    680                0x4   0x12019f File            \Device\ConDrv\Reference
0xffffc001bea9dc60    680                0x8        0x3 Directory       KnownDlls
0xffffe00040428070    680                0xc   0x100020 File            \Device\HarddiskVolume1\Users\IEUser\Downloads
0xffffe00041c93980    680               0x10   0x12019f File            \Device\ConDrv\Connect
0xffffe00041bb6180    680               0x14   0x100003 Semaphore
0xffffe000401d12c0    680               0x24   0x1f0001 ALPC Port
```

Do the same for the second process

Command: vol.py -f memory_dump.mem --profile=Win81U1x64 handles -p 3744

Count the total number of entries by appending wc -l to command.

Commands:
vol.py -f memory_dump.mem --profile=Win81U1x64 handles -p 680 | wc -l
vol.py -f memory_dump.mem --profile=Win81U1x64 handles -p 3744 | wc -l

It is important to note that there are 2 extra counts due to the top description line in each case. Hence, in total, there are 4 extra in the count for these two commands. Subtract that from the count and the answer will be 80.

Q5. One of the handles opened by binary is a mutex. Provide the full details of that mutex?

Answer: testmutex_{D0E858DF-985E-4907-B7FB-8D732C3FC3B9}

Command: vol.py -f memory_dump.mem --profile=Win81U1x64 handles -p 680 -t mutant

```
root@attackdefense:~# vol.py -f memory_dump.mem --profile=Win81U1x64 handles -p 680 -t mutant
Volatility Foundation Volatility Framework 2.6.1
Offset(V)            Pid             Handle             Access Type             Details
------------------ ------ ------------------ ------------------ --------------- -------
0xffffe00041bec210    680              0x100          0x1f0001 Mutant          testmutex_{D0E858DF-985E-4907-B7FB-8D732C3FC3B9}
root@attackdefense:~#
```

References:

1. Volatility (https://github.com/volatilityfoundation/volatility)