ATTACK DEFENSE

by PentesterAcademy

Name	Volatility: Basic (Windows)
URL	https://attackdefense.com/challengedetails?cid=1117
Type	Forensics: Memory Forensics

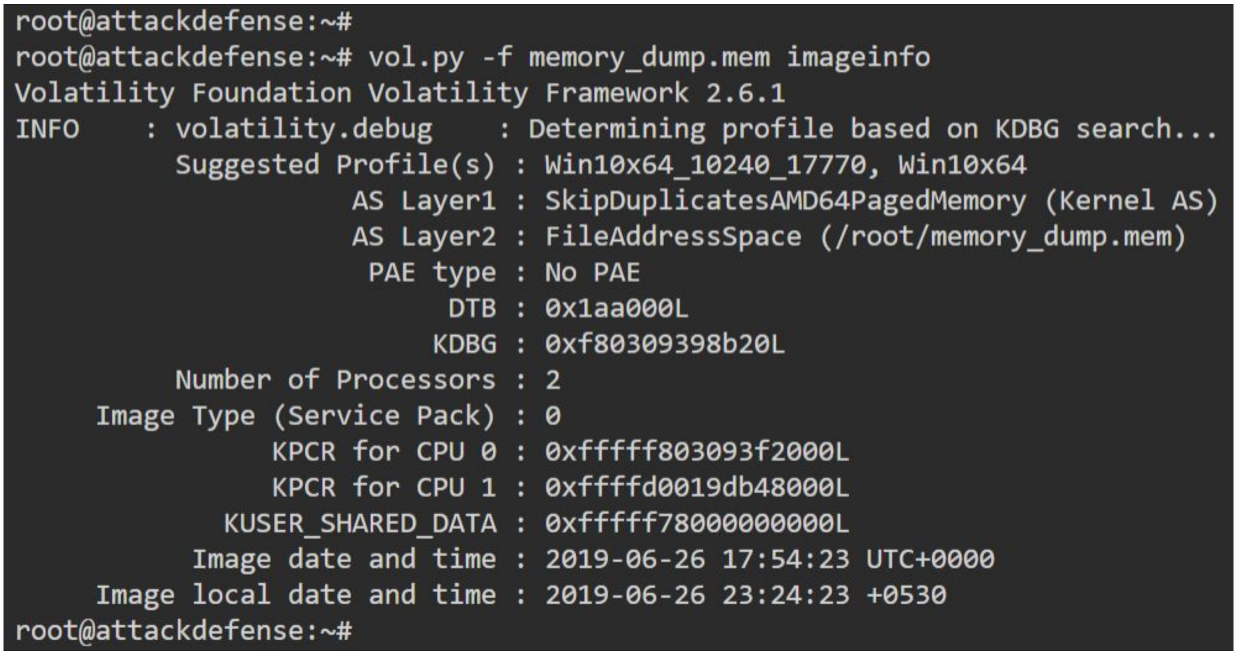
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this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

A memory dump of a Windows machine is provided in the home directory of the root user. You have to use Volatility to analyze the memory dump and answer the following questions:

Q1. Which profile is suitable for the given memory dump?

Answer: Win10x64_10240_17770

Command: vol.py -f memory_dump.mem imageinfo

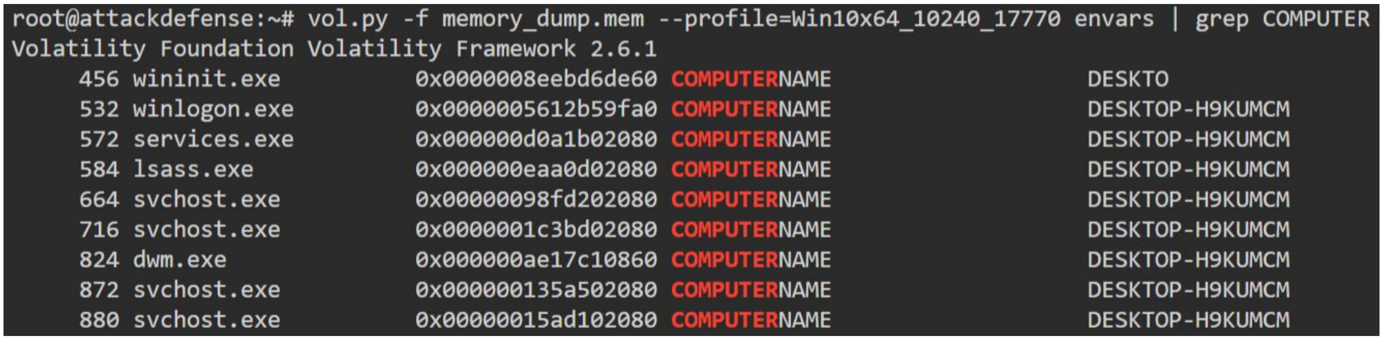


```
root@attackdefense:~#
root@attackdefense:~# vol.py -f memory_dump.mem imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win10x64_10240_17770, Win10x64
                     AS Layer1 : SkipDuplicatesAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/root/memory_dump.mem)
                      PAE type : No PAE
                           DTB : 0x1aa000L
                          KDBG : 0xf80309398b20L
          Number of Processors : 2
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0xfffff803093f2000L
                KPCR for CPU 1 : 0xffffd0019db48000L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2019-06-26 17:54:23 UTC+0000
     Image local date and time : 2019-06-26 23:24:23 +0530
root@attackdefense:~#
```

Q2. What is the name of the machine (i.e. COMPUTERNAME)?

Answer: DESKTOP-H9KUMCM

Command: vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 envars | grep COMPUTER

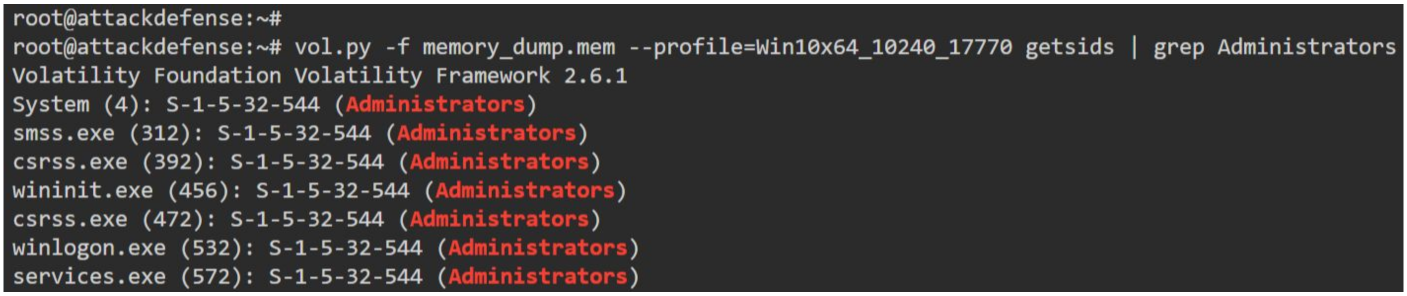


```
root@attackdefense:~# vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 envars | grep COMPUTER
Volatility Foundation Volatility Framework 2.6.1
     456 wininit.exe        0x0000008eebd6de60 COMPUTERNAME                   DESKTO
     532 winlogon.exe       0x0000005612b59fa0 COMPUTERNAME                   DESKTOP-H9KUMCM
     572 services.exe       0x000000d0a1b02080 COMPUTERNAME                   DESKTOP-H9KUMCM
     584 lsass.exe          0x000000eaa0d02080 COMPUTERNAME                   DESKTOP-H9KUMCM
     664 svchost.exe        0x00000098fd202080 COMPUTERNAME                   DESKTOP-H9KUMCM
     716 svchost.exe        0x0000001c3bd02080 COMPUTERNAME                   DESKTOP-H9KUMCM
     824 dwm.exe            0x000000ae17c10860 COMPUTERNAME                   DESKTOP-H9KUMCM
     872 svchost.exe        0x000000135a502080 COMPUTERNAME                   DESKTOP-H9KUMCM
     880 svchost.exe        0x00000015ad102080 COMPUTERNAME                   DESKTOP-H9KUMCM
```

Q3. What is the SID associated with the running process winlogon.exe?

Answer: S-1-5-32-544

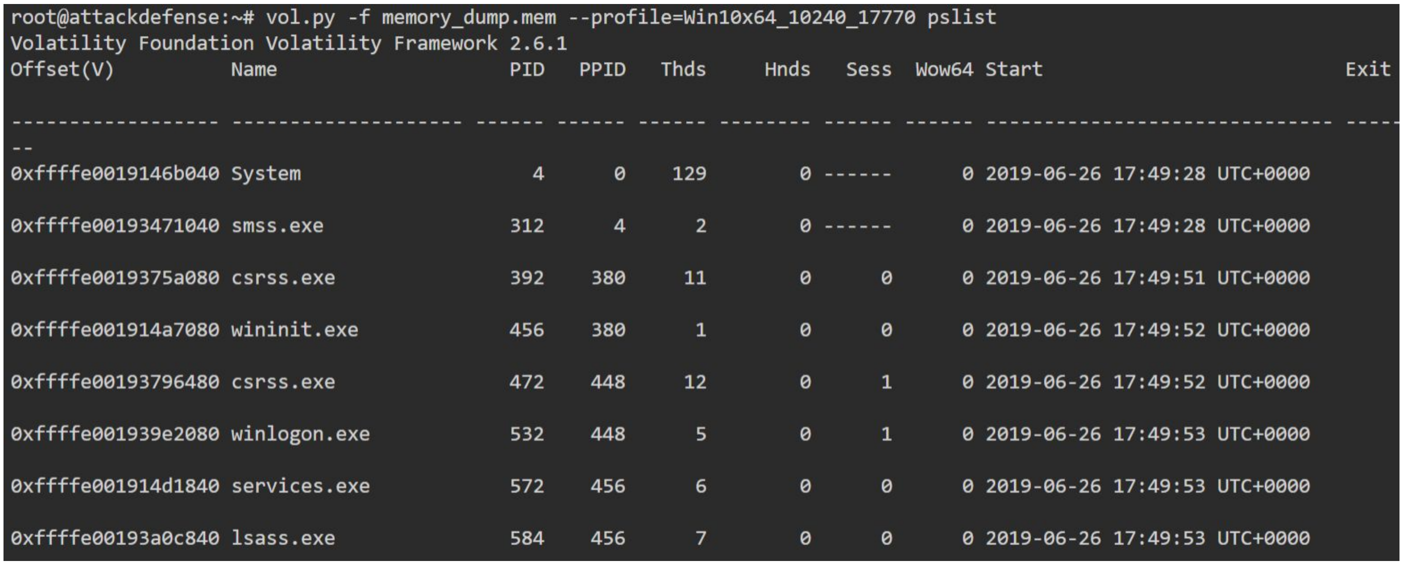
Command: vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 getsids | grep Administrators



```
root@attackdefense:~#
root@attackdefense:~# vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 getsids | grep Administrators
Volatility Foundation Volatility Framework 2.6.1
System (4): S-1-5-32-544 (Administrators)
smss.exe (312): S-1-5-32-544 (Administrators)
csrss.exe (392): S-1-5-32-544 (Administrators)
wininit.exe (456): S-1-5-32-544 (Administrators)
csrss.exe (472): S-1-5-32-544 (Administrators)
winlogon.exe (532): S-1-5-32-544 (Administrators)
services.exe (572): S-1-5-32-544 (Administrators)
```

Q4. Which command is used to view the list of running processes?

Command: vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 pslist

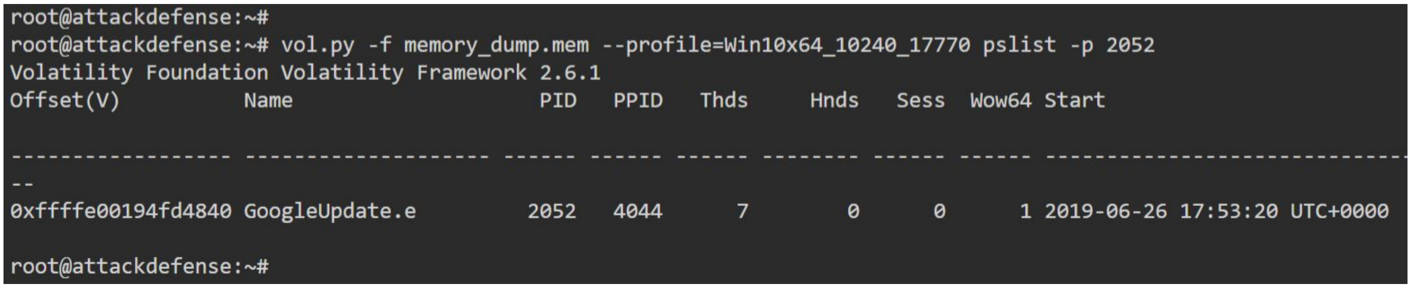


```
root@attackdefense:~# vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit

------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ -----
--
0xffffe0019146b040 System                    4      0    129        0 ------      0 2019-06-26 17:49:28 UTC+0000

0xffffe00193471040 smss.exe                312      4      2        0 ------      0 2019-06-26 17:49:28 UTC+0000

0xffffe0019375a080 csrss.exe               392    380     11        0      0      0 2019-06-26 17:49:51 UTC+0000

0xffffe001914a7080 wininit.exe             456    380      1        0      0      0 2019-06-26 17:49:52 UTC+0000

0xffffe00193796480 csrss.exe               472    448     12        0      1      0 2019-06-26 17:49:52 UTC+0000

0xffffe001939e2080 winlogon.exe            532    448      5        0      1      0 2019-06-26 17:49:53 UTC+0000

0xffffe001914d1840 services.exe            572    456      6        0      0      0 2019-06-26 17:49:53 UTC+0000

0xffffe00193a0c840 lsass.exe               584    456      7        0      0      0 2019-06-26 17:49:53 UTC+0000
```

Q5. What is the Offset for the process with PID 2052?

Answer: 0xffffe00194fd4840

Command: vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 pslist -p 2052

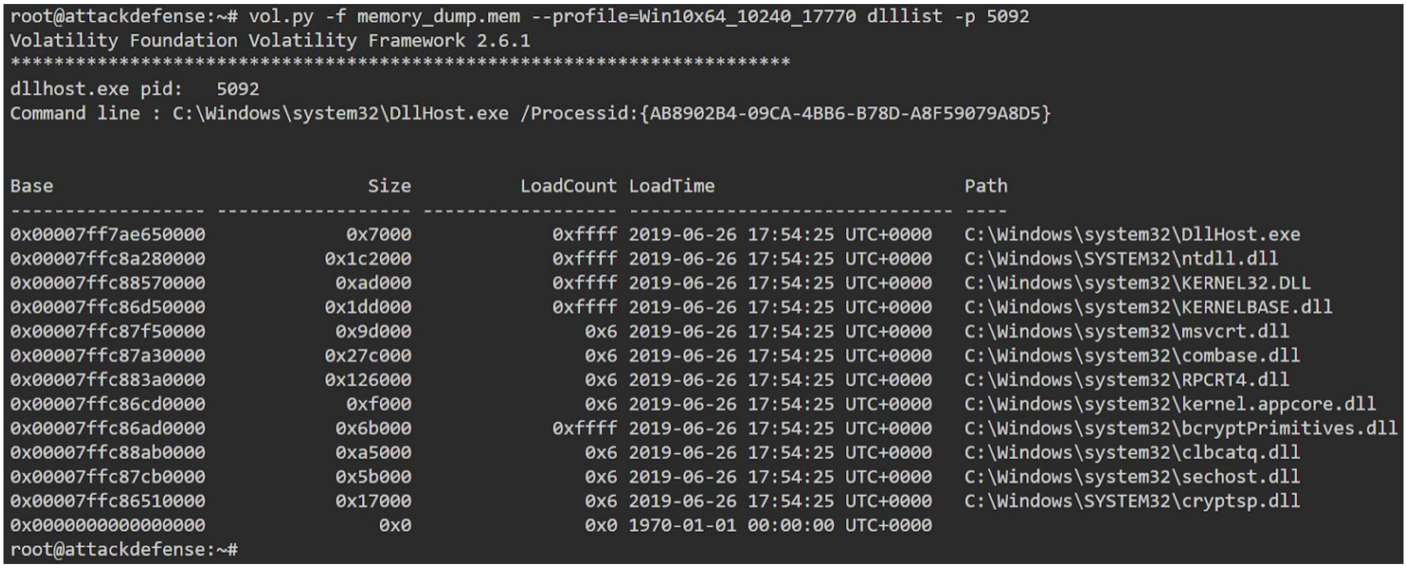


```
root@attackdefense:~#
root@attackdefense:~# vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 pslist -p 2052
Volatility Foundation Volatility Framework 2.6.1
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start

------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------
--
0xffffe00194fd4840 GoogleUpdate.e         2052   4044      7        0      0      1 2019-06-26 17:53:20 UTC+0000

root@attackdefense:~#
```

Q6. How many DLLs were loaded by the process with PID 5092?

Answer: 11

Command: vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 dlllist -p 5092



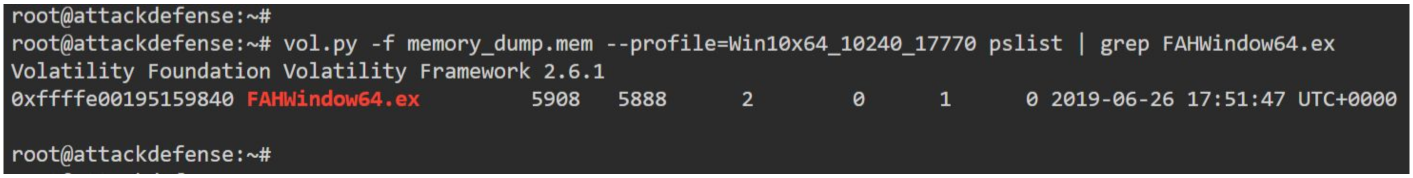
```
root@attackdefense:~# vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 dlllist -p 5092
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
dllhost.exe pid:   5092
Command line : C:\Windows\system32\DllHost.exe /Processid:{AB8902B4-09CA-4BB6-B78D-A8F59079A8D5}


Base                             Size          LoadCount LoadTime                       Path
------------------ ------------------ ------------------ ------------------------------ ----
0x00007ff7ae650000             0x7000             0xffff 2019-06-26 17:54:25 UTC+0000   C:\Windows\system32\DllHost.exe
0x00007ffc8a280000           0x1c2000             0xffff 2019-06-26 17:54:25 UTC+0000   C:\Windows\SYSTEM32\ntdll.dll
0x00007ffc88570000            0xad000             0xffff 2019-06-26 17:54:25 UTC+0000   C:\Windows\system32\KERNEL32.DLL
0x00007ffc86d50000           0x1dd000             0xffff 2019-06-26 17:54:25 UTC+0000   C:\Windows\system32\KERNELBASE.dll
0x00007ffc87f50000            0x9d000                0x6 2019-06-26 17:54:25 UTC+0000   C:\Windows\system32\msvcrt.dll
0x00007ffc87a30000           0x27c000                0x6 2019-06-26 17:54:25 UTC+0000   C:\Windows\system32\combase.dll
0x00007ffc883a0000           0x126000                0x6 2019-06-26 17:54:25 UTC+0000   C:\Windows\system32\RPCRT4.dll
0x00007ffc86cd0000             0xf000                0x6 2019-06-26 17:54:25 UTC+0000   C:\Windows\system32\kernel.appcore.dll
0x00007ffc86ad0000            0x6b000             0xffff 2019-06-26 17:54:25 UTC+0000   C:\Windows\system32\bcryptPrimitives.dll
0x00007ffc88ab0000            0xa5000                0x6 2019-06-26 17:54:25 UTC+0000   C:\Windows\system32\clbcatq.dll
0x00007ffc87cb0000            0x5b000                0x6 2019-06-26 17:54:25 UTC+0000   C:\Windows\system32\sechost.dll
0x00007ffc86510000            0x17000                0x6 2019-06-26 17:54:25 UTC+0000   C:\Windows\SYSTEM32\cryptsp.dll
0x0000000000000000                0x0                0x0 1970-01-01 00:00:00 UTC+0000
root@attackdefense:~#
```

Q7. What command line argument was passed to 'FAHWindow64.exe' binary?

Answer: register

Check the PID of process

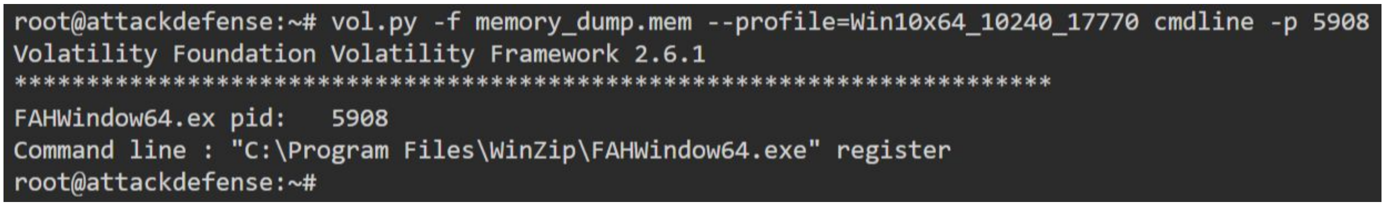
Command: vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 pslist | grep FAHWindow64



```
root@attackdefense:~#
root@attackdefense:~# vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 pslist | grep FAHWindow64.ex
Volatility Foundation Volatility Framework 2.6.1
0xffffe00195159840 FAHWindow64.ex         5908   5888      2        0      1      0 2019-06-26 17:51:47 UTC+0000

root@attackdefense:~#
```

Use PID to get command line argument

Command: vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 cmdline -p 5908

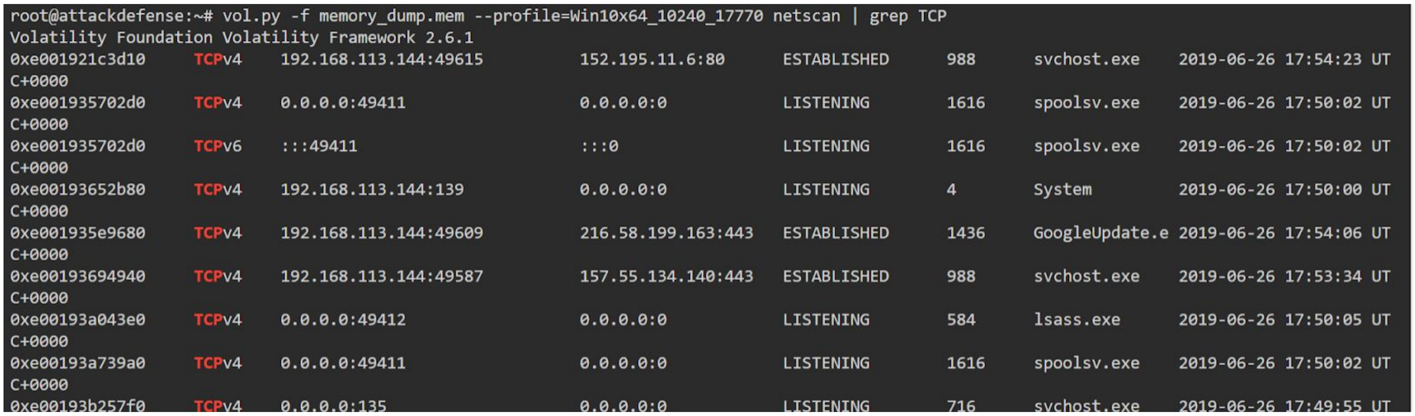


```
root@attackdefense:~# vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 cmdline -p 5908
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
FAHWindow64.ex pid:   5908
Command line : "C:\Program Files\WinZip\FAHWindow64.exe" register
root@attackdefense:~#
```

Q8. GoogleUpdate.exe is connected to a remote machine. What is the IP address of that remote machine?

Answer: 216.58.199.163

Command: vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 netscan | grep TCP



```
root@attackdefense:~# vol.py -f memory_dump.mem --profile=Win10x64_10240_17770 netscan | grep TCP
Volatility Foundation Volatility Framework 2.6.1
0xe001921c3d10     TCPv4    192.168.113.144:49615        152.195.11.6:80      ESTABLISHED      988      svchost.exe    2019-06-26 17:54:23 UT
C+0000
0xe001935702d0     TCPv4    0.0.0.0:49411                0.0.0.0:0            LISTENING        1616     spoolsv.exe    2019-06-26 17:50:02 UT
C+0000
0xe001935702d0     TCPv6    :::49411                     :::0                 LISTENING        1616     spoolsv.exe    2019-06-26 17:50:02 UT
C+0000
0xe00193652b80     TCPv4    192.168.113.144:139          0.0.0.0:0            LISTENING        4        System         2019-06-26 17:50:00 UT
C+0000
0xe001935e9680     TCPv4    192.168.113.144:49609        216.58.199.163:443   ESTABLISHED      1436     GoogleUpdate.e 2019-06-26 17:54:06 UT
C+0000
0xe00193694940     TCPv4    192.168.113.144:49587        157.55.134.140:443   ESTABLISHED      988      svchost.exe    2019-06-26 17:53:34 UT
C+0000
0xe00193a043e0     TCPv4    0.0.0.0:49412                0.0.0.0:0            LISTENING        584      lsass.exe      2019-06-26 17:50:05 UT
C+0000
0xe00193a739a0     TCPv4    0.0.0.0:49411                0.0.0.0:0            LISTENING        1616     spoolsv.exe    2019-06-26 17:50:02 UT
C+0000
0xe00193b257f0     TCPv4    0.0.0.0:135                  0.0.0.0:0            LISTENING        716      svchost.exe    2019-06-26 17:49:55 UT
```

References:

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