Memory Hunting with volatility

Mohammad Khorram

www.SecureMind.se

WHOAMI

Threat Detection Engineer at SecureMind

https://www.linkedin.com/in/mohammad-khorram-608430199

CERTIFICATION

EC-Council Certified Security Analyst (CSA)

MEMORY HUNTING

Memory Hunting is the process of finding malicious artifacts in memory.
In memory hunting you should answer some questions like:

- On the time of infection what processes were running on the suspect system?
- Is there any suspicious network connection from abnormal process?
- Is there any artifacts from existed process?
- Are there any suspicious DLL loaded by processes?
- Are there any suspicious strings associated with a particular processes?

WHAT IS VOLATILITY

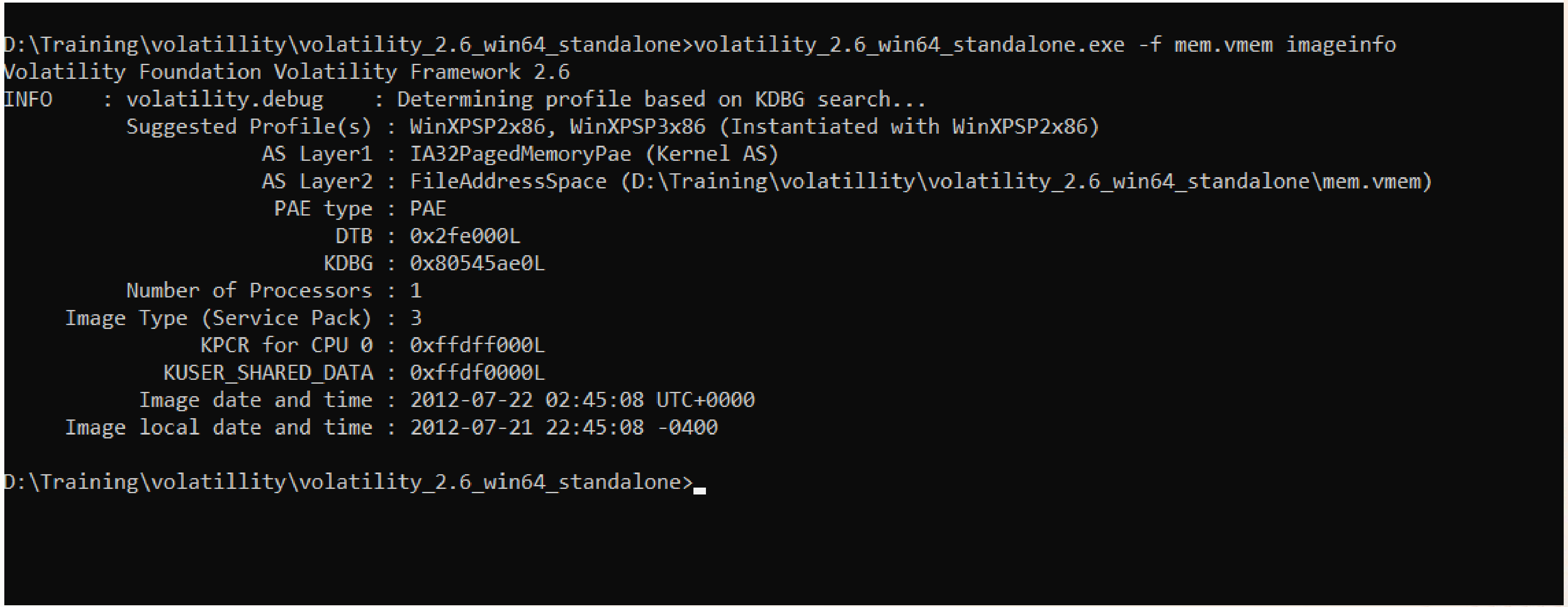
Volatility is one of the best open source software programs for analyzing RAM in 32 bit/64 bit systems. It supports analysis for Linux, Windows, Mac, and Android systems. It is based on Python and can be run on Windows, Linux, and Mac systems. It can analyze raw dumps, crash dumps, VMware dumps (.vmem), virtual box dumps, and many others.

https://www.volatilityfoundation.org/

I SEE CYBER RISKS
EVERYWHERE

STEP 1: FIND THE MEMORY IMAGE PROFILE

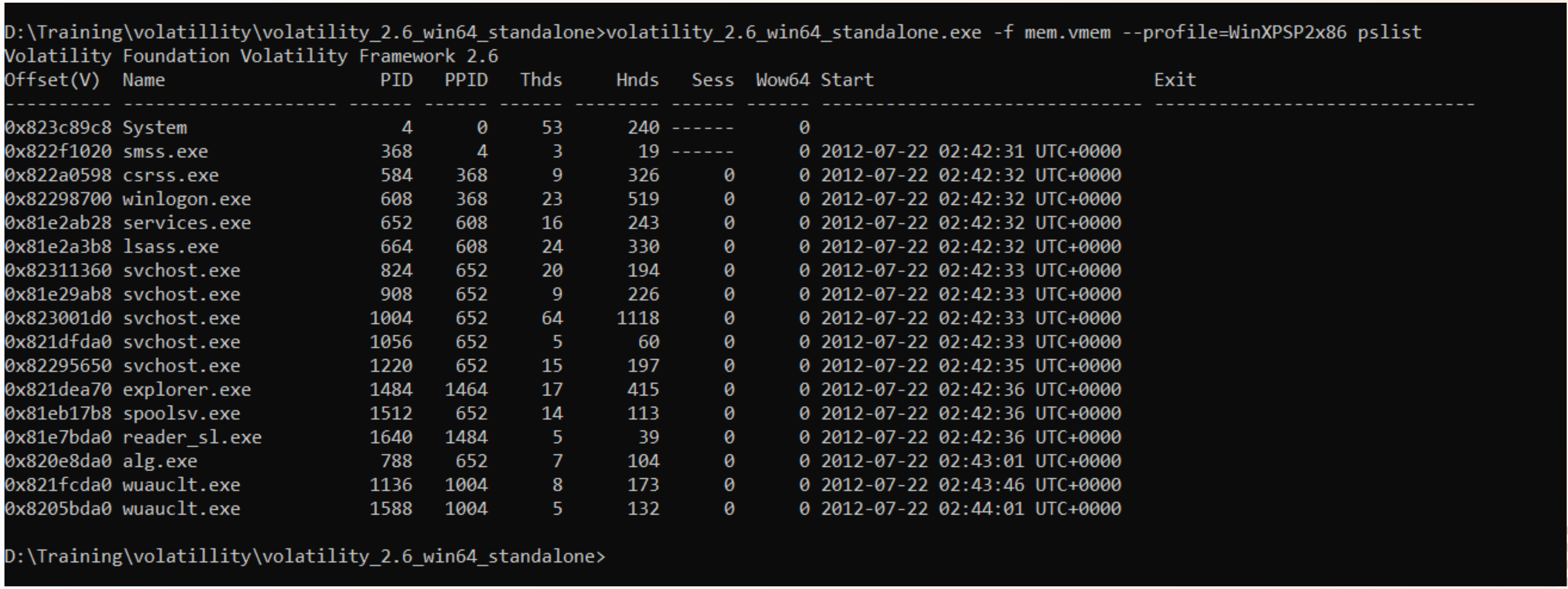
In the first step you should find the memory image profile. identifying the profile is important when certain plugins may be OS dependent. In this example the best profile is WinXPSP2x86



```
D:\Training\volatillity\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone.exe -f mem.vmem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (D:\Training\volatillity\volatility_2.6_win64_standalone\mem.vmem)
                      PAE type : PAE
                           DTB : 0x2fe000L
                          KDBG : 0x80545ae0L
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2012-07-22 02:45:08 UTC+0000
     Image local date and time : 2012-07-21 22:45:08 -0400

D:\Training\volatillity\volatility_2.6_win64_standalone>
```

STEP 2: WHAT PROCESSES WERE RUNNING ON THE SUSPECT SYSTEM AT THE TIME OF THE MEMORY ACQUISITION?

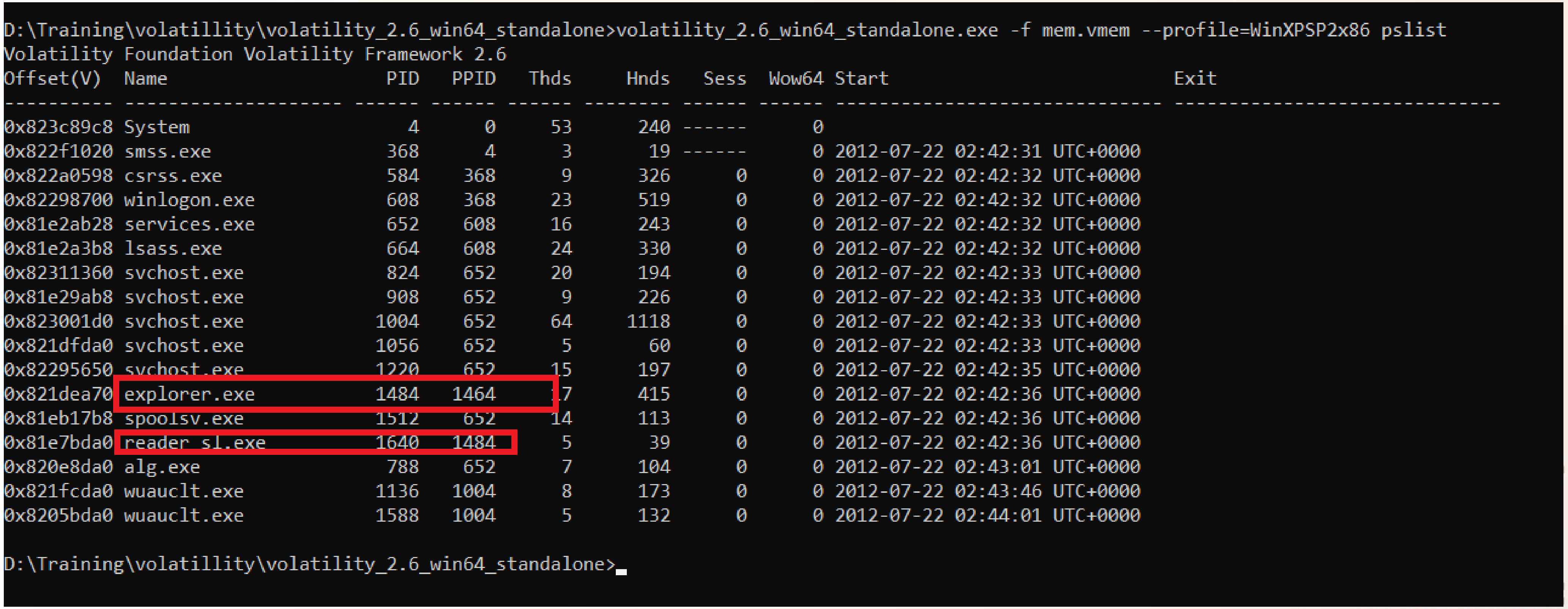
In this example we can use pslist or pstree plugins to list the processes that were running at the time of the memory acquisition .



```
D:\Training\volatillity\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone.exe -f mem.vmem --profile=WinXPSP2x86 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x823c89c8 System                    4      0     53      240 ------      0
0x822f1020 smss.exe                368      4      3       19 ------      0 2012-07-22 02:42:31 UTC+0000
0x822a0598 csrss.exe               584    368      9      326      0      0 2012-07-22 02:42:32 UTC+0000
0x82298700 winlogon.exe            608    368     23      519      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2ab28 services.exe            652    608     16      243      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2a3b8 lsass.exe               664    608     24      330      0      0 2012-07-22 02:42:32 UTC+0000
0x82311360 svchost.exe             824    652     20      194      0      0 2012-07-22 02:42:33 UTC+0000
0x81e29ab8 svchost.exe             908    652      9      226      0      0 2012-07-22 02:42:33 UTC+0000
0x823001d0 svchost.exe            1004    652     64     1118      0      0 2012-07-22 02:42:33 UTC+0000
0x821dfda0 svchost.exe            1056    652      5       60      0      0 2012-07-22 02:42:33 UTC+0000
0x82295650 svchost.exe            1220    652     15      197      0      0 2012-07-22 02:42:35 UTC+0000
0x821dea70 explorer.exe           1484   1464     17      415      0      0 2012-07-22 02:42:36 UTC+0000
0x81eb17b8 spoolsv.exe            1512    652     14      113      0      0 2012-07-22 02:42:36 UTC+0000
0x81e7bda0 reader_sl.exe          1640   1484      5       39      0      0 2012-07-22 02:42:36 UTC+0000
0x820e8da0 alg.exe                 788    652      7      104      0      0 2012-07-22 02:43:01 UTC+0000
0x821fcda0 wuauclt.exe            1136   1004      8      173      0      0 2012-07-22 02:43:46 UTC+0000
0x8205bda0 wuauclt.exe            1588   1004      5      132      0      0 2012-07-22 02:44:01 UTC+0000

D:\Training\volatillity\volatility_2.6_win64_standalone>
```

At first glance , you may see all things normal. However if you look closely you will see

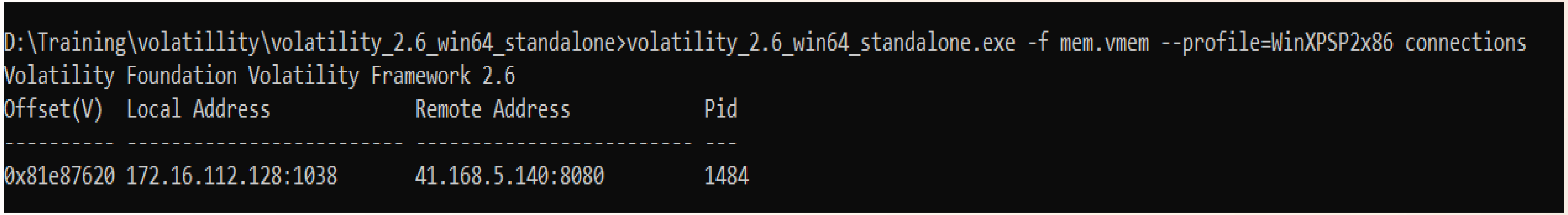
Explorer.exe with PID 1484 has a parent process with PPID 1464 that’s exited and PID 1484 it self spawned some other processes like PID 1640



```
D:\Training\volatillity\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone.exe -f mem.vmem --profile=WinXPSP2x86 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x823c89c8 System                    4      0     53      240 ------      0
0x822f1020 smss.exe                368      4      3       19 ------      0 2012-07-22 02:42:31 UTC+0000
0x822a0598 csrss.exe               584    368      9      326      0      0 2012-07-22 02:42:32 UTC+0000
0x82298700 winlogon.exe            608    368     23      519      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2ab28 services.exe            652    608     16      243      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2a3b8 lsass.exe               664    608     24      330      0      0 2012-07-22 02:42:32 UTC+0000
0x82311360 svchost.exe             824    652     20      194      0      0 2012-07-22 02:42:33 UTC+0000
0x81e29ab8 svchost.exe             908    652      9      226      0      0 2012-07-22 02:42:33 UTC+0000
0x823001d0 svchost.exe            1004    652     64     1118      0      0 2012-07-22 02:42:33 UTC+0000
0x821dfda0 svchost.exe            1056    652      5       60      0      0 2012-07-22 02:42:33 UTC+0000
0x82295650 svchost.exe            1220    652     15      197      0      0 2012-07-22 02:42:35 UTC+0000
0x821dea70 explorer.exe           1484   1464     17      415      0      0 2012-07-22 02:42:36 UTC+0000
0x81eb17b8 spoolsv.exe            1512    652     14      113      0      0 2012-07-22 02:42:36 UTC+0000
0x81e7bda0 reader_sl.exe          1640   1484      5       39      0      0 2012-07-22 02:42:36 UTC+0000
0x820e8da0 alg.exe                 788    652      7      104      0      0 2012-07-22 02:43:01 UTC+0000
0x821fcda0 wuauclt.exe            1136   1004      8      173      0      0 2012-07-22 02:43:46 UTC+0000
0x8205bda0 wuauclt.exe            1588   1004      5      132      0      0 2012-07-22 02:44:01 UTC+0000

D:\Training\volatillity\volatility_2.6_win64_standalone>
```

STEP 3: FIND SUSPICIOUS NETWORK CONNECTIONS

With **connection** command we will see the active network connections at the time of memory acquisition.

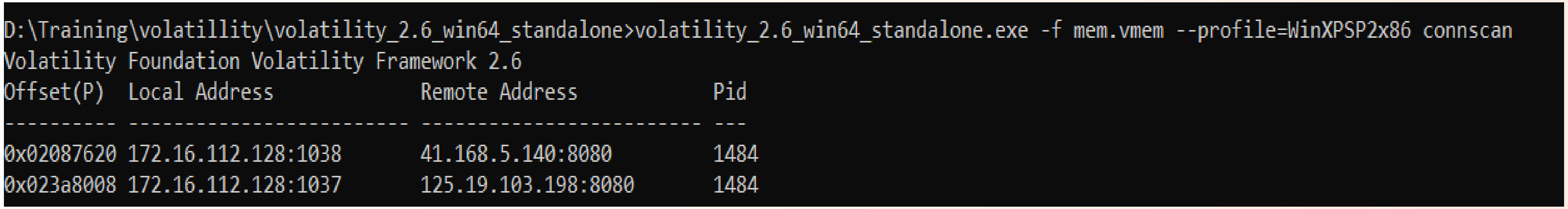


```
D:\Training\volatillity\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone.exe -f mem.vmem --profile=WinXPSP2x86 connections
Volatility Foundation Volatility Framework 2.6
Offset(V)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ---
0x81e87620 172.16.112.128:1038       41.168.5.140:8080         1484
```

In this example we see suspicious network connection to 41.168.5.140 from PID 1484 that it is associated with explorer.exe

STEP 3: FIND SUSPICIOUS NETWORK CONNECTIONS

With **connscan** command we will see that several connections were made.

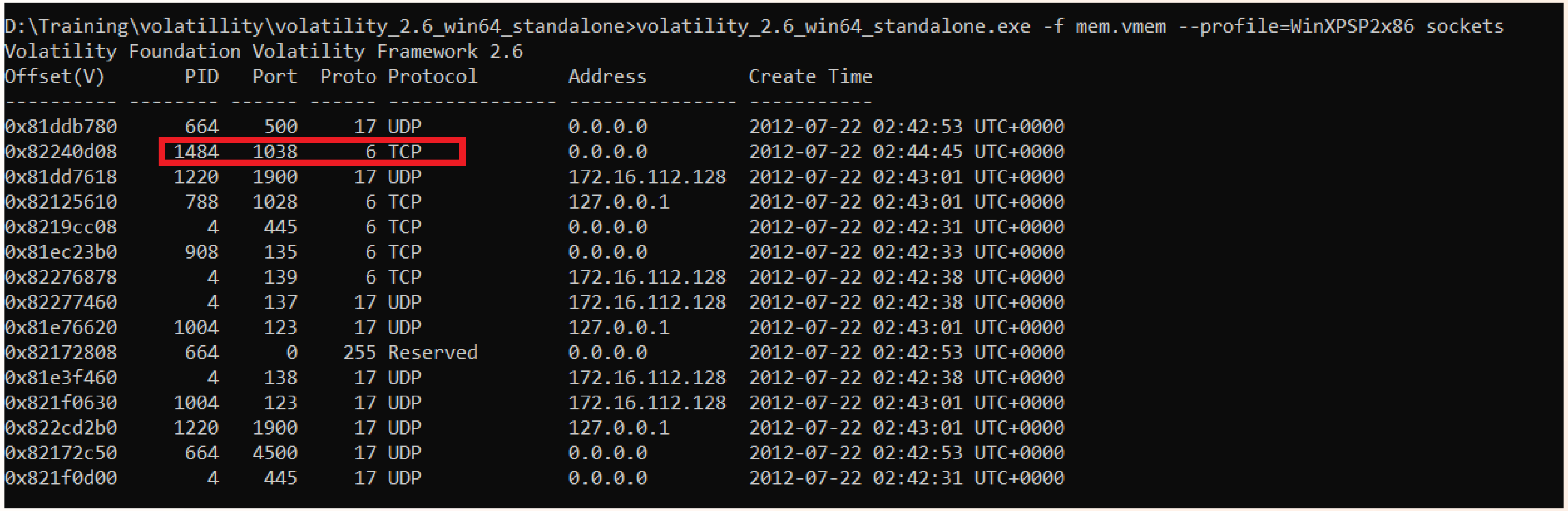


```
D:\Training\volatillity\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone.exe -f mem.vmem --profile=WinXPSP2x86 connscan
Volatility Foundation Volatility Framework 2.6
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ---
0x02087620 172.16.112.128:1038       41.168.5.140:8080         1484
0x023a8008 172.16.112.128:1037       125.19.103.198:8080       1484
```

In this example we see another suspicious network connection to 125.19.103.198 from PID 1484 that it is associated with explorer.exe

STEP 4: CHECK THE SOCKETS

With **sockets** command we can see the active sockets on the suspected system at the time of memory acquisition.



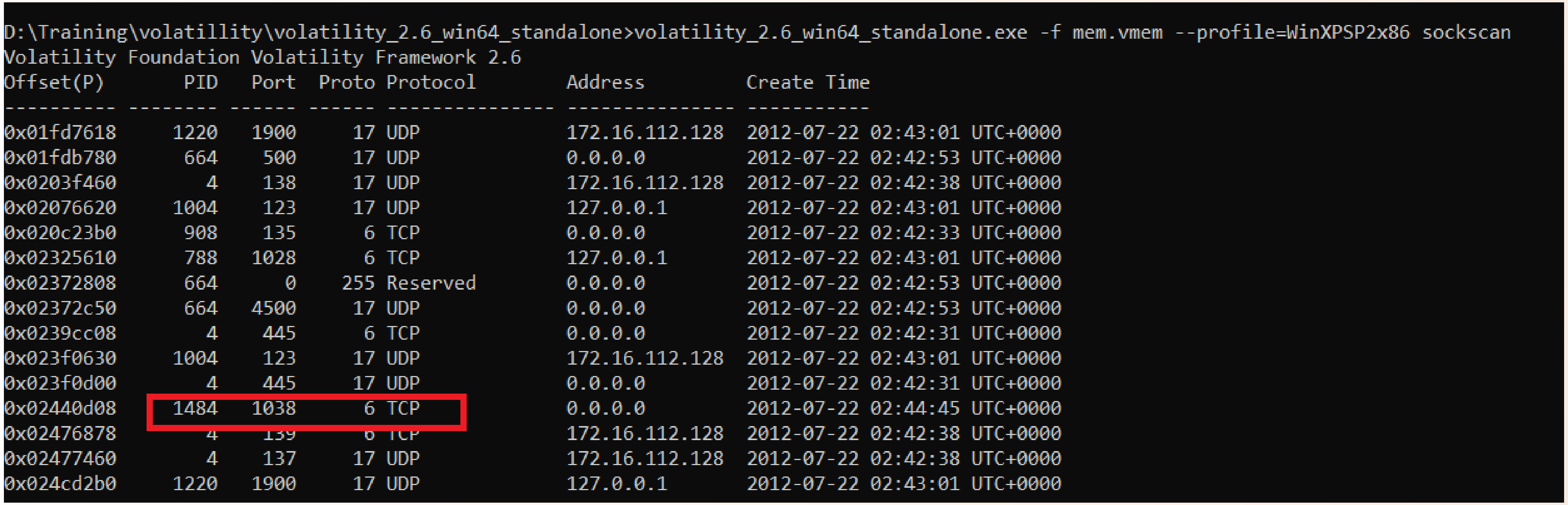
```
D:\Training\volatillity\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone.exe -f mem.vmem --profile=WinXPSP2x86 sockets
Volatility Foundation Volatility Framework 2.6
Offset(V)       PID   Port  Proto Protocol        Address         Create Time
---------- -------- ------ ------ --------------- --------------- -----------
0x81ddb780      664    500     17 UDP             0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x82240d08     1484   1038      6 TCP             0.0.0.0         2012-07-22 02:44:45 UTC+0000
0x81dd7618     1220   1900     17 UDP             172.16.112.128  2012-07-22 02:43:01 UTC+0000
0x82125610      788   1028      6 TCP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x8219cc08        4    445      6 TCP             0.0.0.0         2012-07-22 02:42:31 UTC+0000
0x81ec23b0      908    135      6 TCP             0.0.0.0         2012-07-22 02:42:33 UTC+0000
0x82276878        4    139      6 TCP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x82277460        4    137     17 UDP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x81e76620     1004    123     17 UDP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x82172808      664      0    255 Reserved        0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x81e3f460        4    138     17 UDP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x821f0630     1004    123     17 UDP             172.16.112.128  2012-07-22 02:43:01 UTC+0000
0x822cd2b0     1220   1900     17 UDP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x82172c50      664   4500     17 UDP             0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x821f0d00        4    445     17 UDP             0.0.0.0         2012-07-22 02:42:31 UTC+0000
```

We can see that there is a socket with Source port 1038 that is associated with PID 1484.

We saw this source port in connection plugin output, so there is nothing new here

STEP 4: CHECK THE SOCKETS

With **sockscan** command we can see the sockets were created on the suspected system.



```
D:\Training\volatillity\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone.exe -f mem.vmem --profile=WinXPSP2x86 sockscan
Volatility Foundation Volatility Framework 2.6
Offset(P)       PID   Port  Proto Protocol        Address         Create Time
---------- -------- ------ ------ --------------- --------------- -----------
0x01fd7618     1220   1900     17 UDP             172.16.112.128  2012-07-22 02:43:01 UTC+0000
0x01fdb780      664    500     17 UDP             0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x0203f460        4    138     17 UDP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x02076620     1004    123     17 UDP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x020c23b0      908    135      6 TCP             0.0.0.0         2012-07-22 02:42:33 UTC+0000
0x02325610      788   1028      6 TCP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x02372808      664      0    255 Reserved        0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x02372c50      664   4500     17 UDP             0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x0239cc08        4    445      6 TCP             0.0.0.0         2012-07-22 02:42:31 UTC+0000
0x023f0630     1004    123     17 UDP             172.16.112.128  2012-07-22 02:43:01 UTC+0000
0x023f0d00        4    445     17 UDP             0.0.0.0         2012-07-22 02:42:31 UTC+0000
0x02440d08     1484   1038      6 TCP             0.0.0.0         2012-07-22 02:44:45 UTC+0000
0x02476878        4    139      6 TCP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x02477460        4    137     17 UDP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x024cd2b0     1220   1900     17 UDP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
```

According to the output there were no other suspicious sockets associated with PID 1484

STEP 5: ANALYZE THE IP ADDRESSES

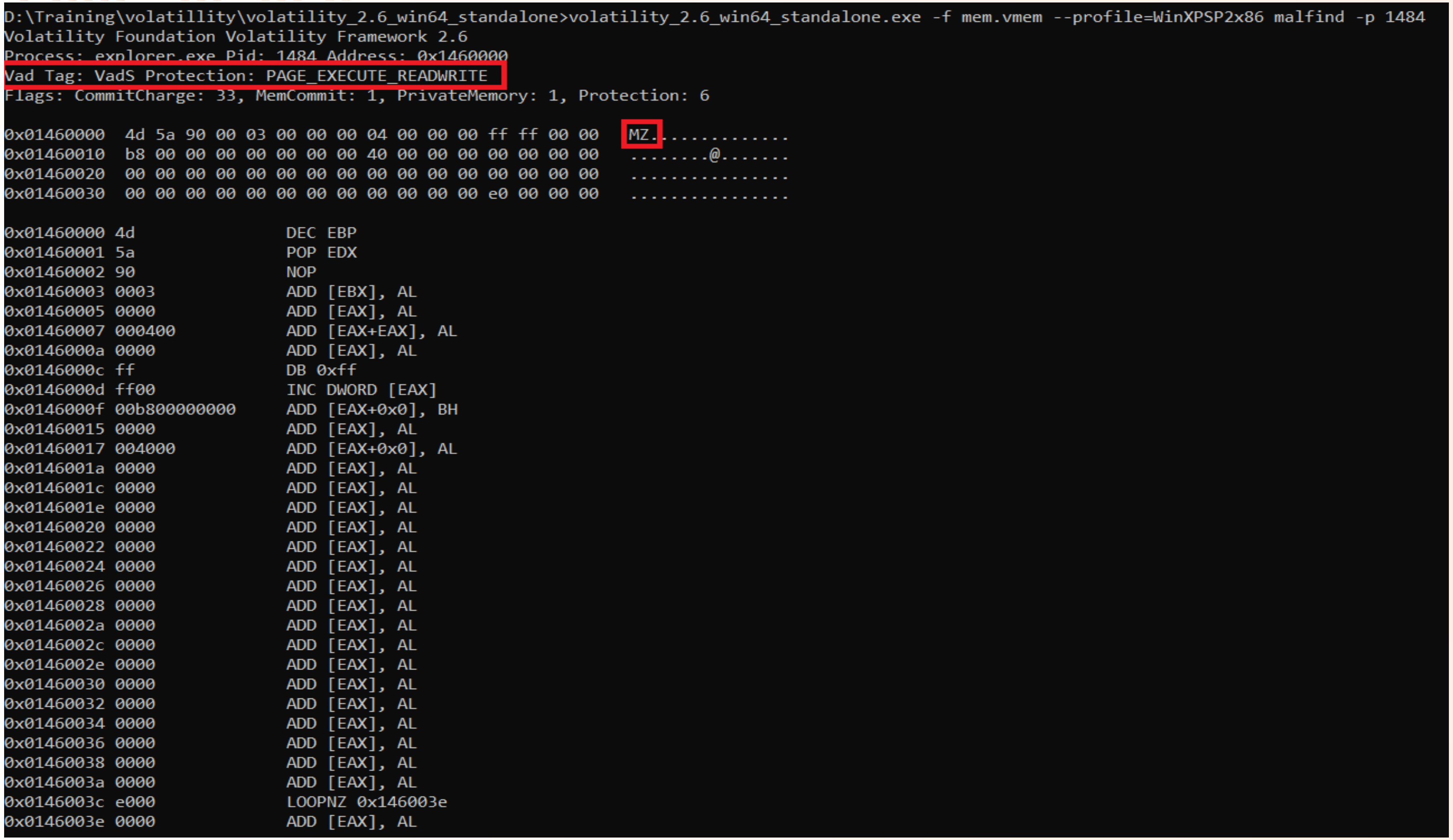
At this step you can check the IP address artifacts with online OSINT services like:

- Virustotal
- Whois
- IBM xforce
- Talos intellgence

STEP 6: FINDING THE REMOTE CODE EXECUTION

At this step we will check if any remote code execution is done on PID 1484.
In remote code execution you will have readable, writeable, and executable private memory region. This region will contain PE file header or valid CPU instruction that can indicate a shellcode.

With **malfind** plugin you check the protection on this private memory region

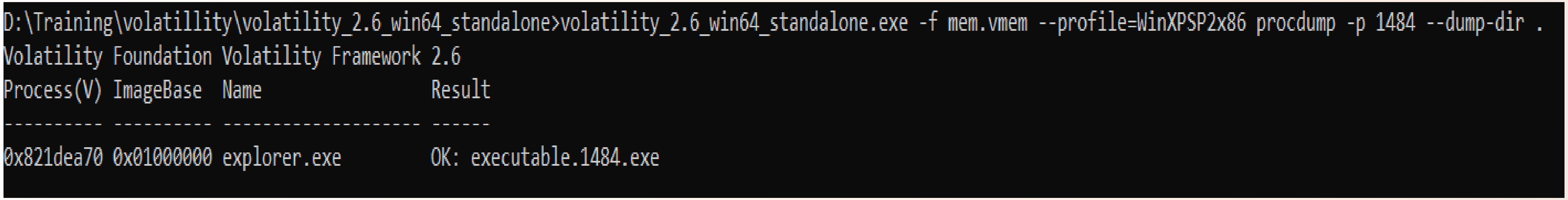


```
D:\Training\volatillity\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone.exe -f mem.vmem --profile=WinXPSP2x86 malfind -p 1484
Volatility Foundation Volatility Framework 2.6
Process: explorer.exe Pid: 1484 Address: 0x1460000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 33, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x01460000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x01460010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x01460020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x01460030  00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00   ................

0x01460000 4d               DEC EBP
0x01460001 5a               POP EDX
0x01460002 90               NOP
0x01460003 0003             ADD [EBX], AL
0x01460005 0000             ADD [EAX], AL
0x01460007 000400           ADD [EAX+EAX], AL
0x0146000a 0000             ADD [EAX], AL
0x0146000c ff               DB 0xff
0x0146000d ff00             INC DWORD [EAX]
0x0146000f 00b800000000     ADD [EAX+0x0], BH
0x01460015 0000             ADD [EAX], AL
0x01460017 004000           ADD [EAX+0x0], AL
0x0146001a 0000             ADD [EAX], AL
0x0146001c 0000             ADD [EAX], AL
0x0146001e 0000             ADD [EAX], AL
0x01460020 0000             ADD [EAX], AL
0x01460022 0000             ADD [EAX], AL
0x01460024 0000             ADD [EAX], AL
0x01460026 0000             ADD [EAX], AL
0x01460028 0000             ADD [EAX], AL
0x0146002a 0000             ADD [EAX], AL
0x0146002c 0000             ADD [EAX], AL
0x0146002e 0000             ADD [EAX], AL
0x01460030 0000             ADD [EAX], AL
0x01460032 0000             ADD [EAX], AL
0x01460034 0000             ADD [EAX], AL
0x01460036 0000             ADD [EAX], AL
0x01460038 0000             ADD [EAX], AL
0x0146003a 0000             ADD [EAX], AL
0x0146003c e000             LOOPNZ 0x146003e
0x0146003e 0000             ADD [EAX], AL
```

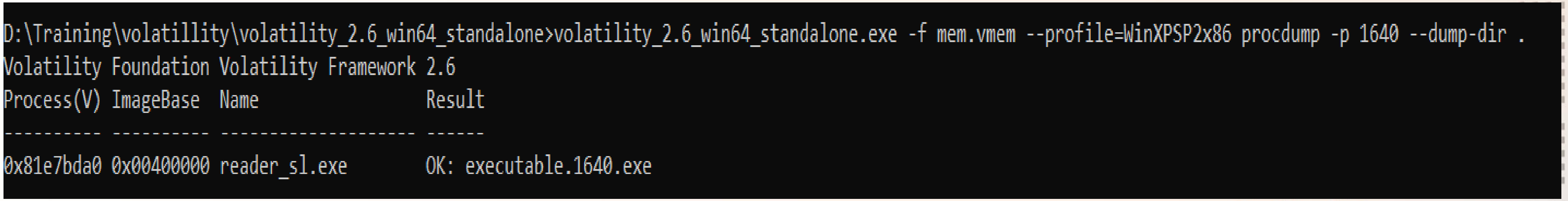
At this step you see VAD with Vads protection PAGE_EXECUTE_READWRITE
You can also see MZ character that indicates the PE header.

STEP 7: DUMP THE PROCESS EXECUTABLE

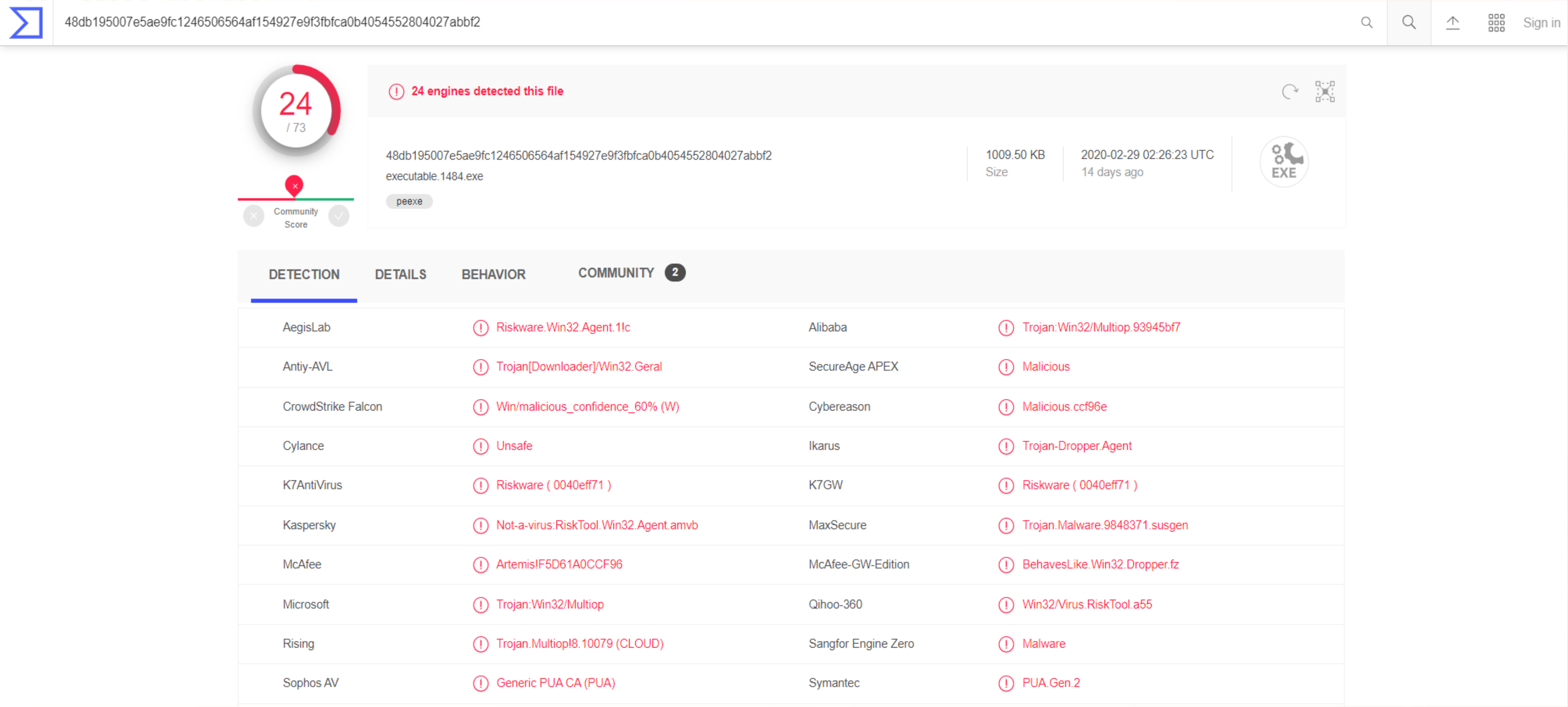
At this time based on the parent/child relation ship between PID 1484 and 1640
Our hypothesis is that some sort of remote code execution is performed by PID 1640 on PID 1484
we will dump the executables of PID 1484 and 1640 and test it on virustotal



```
D:\Training\volatillity\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone.exe -f mem.vmem --profile=WinXPSP2x86 procdump -p 1484 --dump-dir .
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x821dea70 0x01000000 explorer.exe         OK: executable.1484.exe
```



```
D:\Training\volatillity\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone.exe -f mem.vmem --profile=WinXPSP2x86 procdump -p 1640 --dump-dir .
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x81e7bda0 0x00400000 reader_sl.exe        OK: executable.1640.exe
```



48db195007e5ae9fc1246506564af154927e9f3fbfca0b4054552804027abbf2

24 / 73

24 engines detected this file

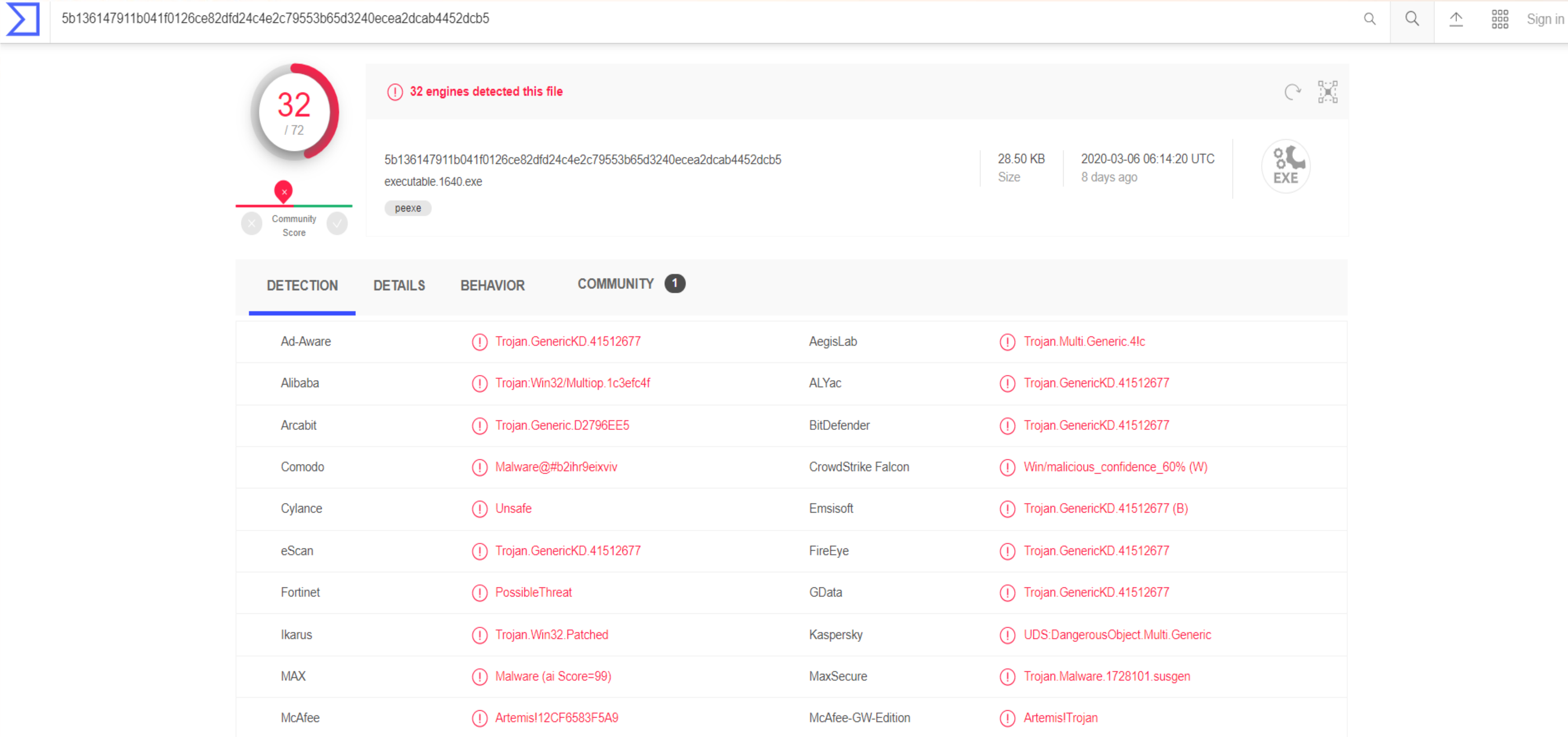
48db195007e5ae9fc1246506564af154927e9f3fbfca0b4054552804027abbf2
executable.1484.exe
peexe

1009.50 KB
Size

2020-02-29 02:26:23 UTC
14 days ago

AegisLab	Riskware.Win32.Agent.1!c	Alibaba	Trojan:Win32/Multiop.93945bf7
Antiy-AVL	Trojan[Downloader]/Win32.Geral	SecureAge APEX	Malicious
CrowdStrike Falcon	Win/malicious_confidence_60% (W)	Cybereason	Malicious.ccf96e
Cylance	Unsafe	Ikarus	Trojan-Dropper.Agent
K7AntiVirus	Riskware (0040eff71)	K7GW	Riskware (0040eff71)
Kaspersky	Not-a-virus:RiskTool.Win32.Agent.amvb	MaxSecure	Trojan.Malware.9848371.susgen
McAfee	Artemis!F5D61A0CCF96	McAfee-GW-Edition	BehavesLike.Win32.Dropper.fz
Microsoft	Trojan:Win32/Multiop	Qihoo-360	Win32/Virus.RiskTool.a55
Rising	Trojan.Multiop!8.10079 (CLOUD)	Sangfor Engine Zero	Malware
Sophos AV	Generic PUA CA (PUA)	Symantec	PUA.Gen.2

SECURE MIND



5b136147911b041f0126ce82dfd24c4e2c79553b65d3240ecea2dcab4452dcb5

Sign in

32 / 72

Community Score

32 engines detected this file

5b136147911b041f0126ce82dfd24c4e2c79553b65d3240ecea2dcab4452dcb5
executable.1640.exe
peexe

28.50 KB
Size

2020-03-06 06:14:20 UTC
8 days ago

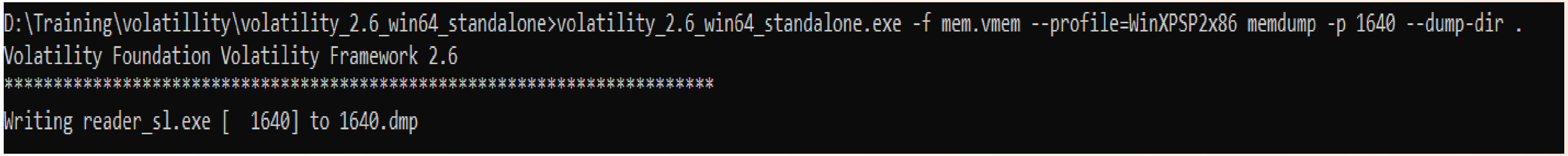
EXE

DETECTION DETAILS BEHAVIOR COMMUNITY 1

Ad-Aware	Trojan.GenericKD.41512677	AegisLab	Trojan.Multi.Generic.4!c
Alibaba	Trojan:Win32/Multiop.1c3efc4f	ALYac	Trojan.GenericKD.41512677
Arcabit	Trojan.Generic.D2796EE5	BitDefender	Trojan.GenericKD.41512677
Comodo	Malware@#b2ihr9eixviv	CrowdStrike Falcon	Win/malicious_confidence_60% (W)
Cylance	Unsafe	Emsisoft	Trojan.GenericKD.41512677 (B)
eScan	Trojan.GenericKD.41512677	FireEye	Trojan.GenericKD.41512677
Fortinet	PossibleThreat	GData	Trojan.GenericKD.41512677
Ikarus	Trojan.Win32.Patched	Kaspersky	UDS:DangerousObject.Multi.Generic
MAX	Malware (ai Score=99)	MaxSecure	Trojan.Malware.1728101.susgen
McAfee	Artemis!12CF6583F5A9	McAfee-GW-Edition	Artemis!Trojan

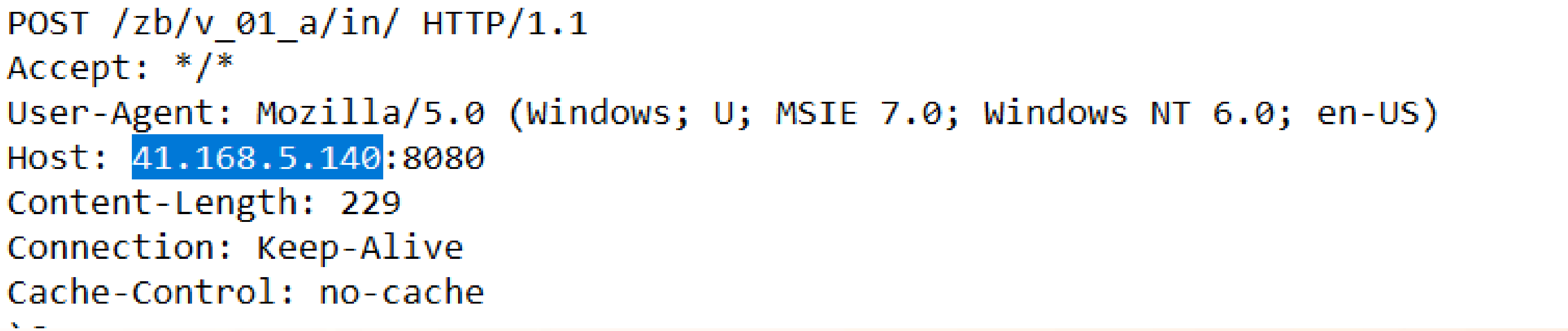
STEP 8: DUMP THE PROCESS MEMORY ADDRESS

At this step we will dump the memory address and use it to find suspicious strings

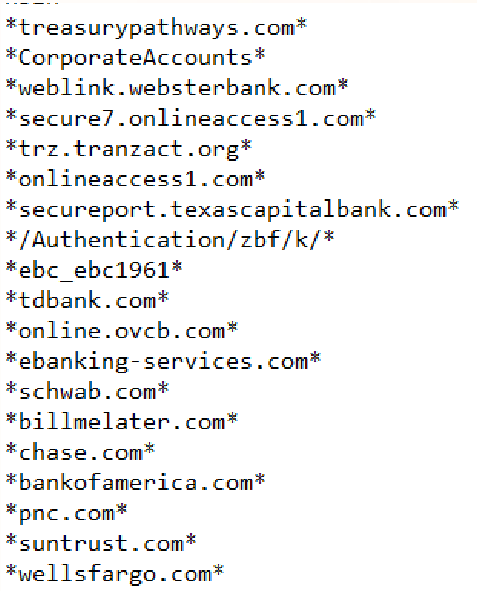


```
D:\Training\volatillity\volatility_2.6_win64_standalone>volatility_2.6_win64_standalone.exe -f mem.vmem --profile=WinXPSP2x86 memdump -p 1640 --dump-dir .
Volatility Foundation Volatility Framework 2.6
************************************************************************
Writing reader_sl.exe [  1640] to 1640.dmp
```

If we search our IP address artifact from **connection** plugin you can see that its Communicating over HTTP protocol with specified user agent. also if we look carefully through The output you can see list of banking domains associated with this process



```
POST /zb/v_01_a/in/ HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0 (Windows; U; MSIE 7.0; Windows NT 6.0; en-US)
Host: 41.168.5.140:8080
Content-Length: 229
Connection: Keep-Alive
Cache-Control: no-cache
```



```
*treasurypathways.com*
*CorporateAccounts*
*weblink.websterbank.com*
*secure7.onlineaccess1.com*
*trz.tranzact.org*
*onlineaccess1.com*
*secureport.texascapitalbank.com*
*/Authentication/zbf/k/*
*ebc_ebc1961*
*tdbank.com*
*online.ovcb.com*
*ebanking-services.com*
*schwab.com*
*billmelater.com*
*chase.com*
*bankofamerica.com*
*pnc.com*
*suntrust.com*
*wellsfargo.com*
```

- For more information about latest threats and free detection rues please visit our website and LinkedIn page:
- https://www.threathunting.se/
- https://linkedin.com/company/threat-hunting

Thank you

www.SecureMind.se