DumpMe Lab (Cyber Defenders) - Walkthrough

Sunday, September 22, 2024 7:34 AM

Story:
A SOC analyst took a memory dump from a machine infected with a meterpreter malware. As a Digital Forensicators, your job is to analyze the dump, extract the available indicators of compromise (IOCs) and answer the provided questions.

Q1: What is the SHA1 hash of Triage-Memory.mem (memory dump)?

```
┌──(kali㉿kali)-[~/Desktop]
└─$ sha1sum Triage-Memory.mem
c95e8cc8c946f95a109ea8e47a6800de10a27abd  Triage-Memory.mem
```

Q2: What volatility profile is the most appropriate for this machine? (ex: Win10x86_14393)

- To address this question, I used 'Imageinfo' plugin by Volatility2.

```
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_24000, Win2008R2SP
418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/kali/Desktop/Triage-Memory.mem)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf800029f80a0L
          Number of Processors : 2
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff800029f9d00L
                KPCR for CPU 1 : 0xfffff880009ee000L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2019-03-22 05:46:00 UTC+0000
     Image local date and time : 2019-03-22 01:46:00 -0400
```

Q3: What was the process ID of notepad.exe?

- **I used 'Pstree' plugin to identify the PID of notepad.exe**

```
.... 0xfffffa8005bb0060:cmd.exe                4660   3496      1     33 2019-03-22 05:35:36 UTC+0000
. 0xfffffa80054f9060:notepad.exe               3032   1432      1     60 2019-03-22 05:32:22 UTC+0000
. 0xfffffa8005b49890:vmtoolsd.exe              1828   1432      6    144 2019-03-22 05:32:10 UTC+0000
. 0xfffffa800474fb30:taskmgr.exe               3792   1432      6    134 2019-03-22 05:34:38 UTC+0000
```

Q4: Name the child process of wscript.exe.

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python2 vol.py -f ../Triage-Memory.mem --profile=Win7SP1x64 pstree | grep wscript
Volatility Foundation Volatility Framework 2.6.1
.. 0xfffffa8005a80060:wscript.exe                5116   3952      8    312 2019-03-22 05:35:32 UTC+0000

┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python2 vol.py -f ../Triage-Memory.mem --profile=Win7SP1x64 pstree | grep 5116
Volatility Foundation Volatility Framework 2.6.1
.. 0xfffffa8005a80060:wscript.exe                5116   3952      8    312 2019-03-22 05:35:32 UTC+0000
... 0xfffffa8005a1d9e0:UWkpjFjDzM.exe            3496   5116      5    109 2019-03-22 05:35:33 UTC+0000
```

Q5: What was the IP address of the machine at the time the RAM dump was created?

- To address this question, I used 'Netscan' plugin and checked the 'Local Address'

```
Offset(P)          Proto    Local Address                  Foreign Address      State            Pid      Owner          Created
0x13e057300        UDPv4    10.0.0.101:55736               *:*                                   2888     svchost.exe    2019-03-22 05:32:20 UTC+0000
0x13e05b4f0        UDPv6    ::1:55735                      *:*                                   2888     svchost.exe    2019-03-22 05:32:20 UTC+0000
0x13e05b790        UDPv6    fe80::7475:ef30:be18:7807:55734 *:*                                  2888     svchost.exe    2019-03-22 05:32:20 UTC+0000
0x13e05d4b0        UDPv6    fe80::7475:ef30:be18:7807:1900 *:*                                   2888     svchost.exe    2019-03-22 05:32:20 UTC+0000
0x13e05dec0        UDPv4    127.0.0.1:55737                *:*                                   2888     svchost.exe    2019-03-22 05:32:20 UTC+0000
0x13e05e3f0        UDPv4    10.0.0.101:1900                *:*                                   2888     svchost.exe    2019-03-22 05:32:20 UTC+0000
0x13e05eab0        UDPv6    ::1:1900                       *:*                                   2888     svchost.exe    2019-03-22 05:32:20 UTC+0000
0x13e064d70        UDPv4    127.0.0.1:1900                 *:*                                   2888     svchost.exe    2019-03-22 05:32:20 UTC+0000
```

Q6: Based on the answer regarding the infected PID, can you determine the IP of the attacker?

- We identified an established connection from a suspicious process, 'UWkpjFjDzM.exe', which is communicating with the IP address '10.0.0.106' over port 4444, the default port used by the Metasploit Framework.

```
0x13e2348a0        TCPv4    -:49366                        192.168.206.181:389  CLOSED           504
0x13e397190        TCPv4    10.0.0.101:49217               10.0.0.106:4444      ESTABLISHED      3496     UWkpjFjDzM.exe
0x13e3986d0        TCPv4    -:49378                        213.209.1.129:25     CLOSED           504
0x13e3abae0        TCPv4    -:49226                        72.51.60.132:443     CLOSED           4048     POWERPNT.EXE
```

Q7: How many processes are associated with VCRUNTIME140.dll?

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python2 vol.py -f ../Triage-Memory.mem --profile=Win7SP1x64 dlllist | grep 'VCRUNTIME140.dll'
Volatility Foundation Volatility Framework 2.6.1
0x000007fefa5c0000          0x16000          0xffff 4168440                    C:\Program Files\Common Files\Microsoft Shared\ClickToRun\VCRUNTIME140.dll
0x0000000745f0000           0x15000          0xffff 47552144                   C:\Program Files (x86)\Microsoft Office\root\Office16\VCRUNTIME140.dll
0x0000000745f0000           0x15000          0xffff 35953048                   C:\Program Files (x86)\Microsoft Office\root\Office16\VCRUNTIME140.dll
0x0000000745f0000           0x15000             0x3 7109480                    C:\Program Files (x86)\Microsoft Office\root\Office16\VCRUNTIME140.dll
0x0000000745f0000           0x15000          0xffff 5871200                    C:\Program Files (x86)\Microsoft Office\root\Office16\VCRUNTIME140.dll
```

Q8: After dumping the infected process, what is its md5 hash?

- I used 'procdump' to dump the malicious process via the command:
  python2 vol.py -f ../Triage-Memory.mem --profile=Win7SP1x64 procdump --pid 3496 --dump-dir .

```
Process(V)         ImageBase          Name                 Result
------------------ ------------------ -------------------- ------
0xfffffa8005a1d9e0 0x0000000000400000 UWkpjFjDzM.exe       OK: executable.3496.exe

┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ md5sum executable.3496.exe
690ea20bc3bdfb328e23005d9a80c290  executable.3496.exe
```

Q9: What is the LM hash of Bob's account?

- To address this question, I used voltaility3 via 'Hashdump' to extract the LM hash

```
┌──(kali㉿kali)-[~/Desktop/volatility3]
└─$ python3 vol.py -f ../Triage-Memory.mem windows.hashdump
Volatility 3 Framework 2.9.0
Progress:  100.00               PDB scanning finished
User    rid     lmhash  nthash

Administrator   500     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
Guest   501     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
Bob     1000    aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
```

Q10: What memory protection constants does the VAD node at 0xfffffa800577ba10 have?

- **I used 'Vadinfo' plugin to** display information about Virtual Address Descriptors (VADs) in a Windows memory image.
  I used grep utility by '0xfffffa800577ba10' keyword:

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python2 vol.py -f ../Triage-Memory.mem --profile=Win7SP1x64 vadinfo | grep -C10 '0xfffffa800577ba10'
Volatility Foundation Volatility Framework 2.6.1
Flags: Protection: 4
Protection: PAGE_READWRITE
Vad Type: VadNone
ControlArea @fffffa8005740740 Segment fffff8a001021f30
NumberOfSectionReferences:          1 NumberOfPfnReferences:          0
NumberOfMappedViews:                2 NumberOfUserReferences:         3
Control Flags: Commit: 1
First prototype PTE: fffff8a001021f78 Last contiguous PTE: fffff8a001021ff0
Flags2:

VAD node @ 0xfffffa800577ba10 Start 0x0000000000030000 End 0x000000000033fff Tag Vad
Flags: NoChange: 1, Protection: 1
Protection: PAGE_READONLY
Vad Type: VadNone
ControlArea @fffffa8005687a50 Segment fffff8a000c4f870
NumberOfSectionReferences:          1 NumberOfPfnReferences:          0
NumberOfMappedViews:               29 NumberOfUserReferences:        30
Control Flags: Commit: 1
First prototype PTE: fffff8a000c4f8b8 Last contiguous PTE: fffff8a000c4f8d0
Flags2: Inherit: 1, SecNoChange: 1
```

Q11: What memory protection did the VAD starting at 0x00000000033c0000 and ending at 0x00000000033dffff have?

- Same method like before

```
VAD node @ 0xfffffa80052652b0 Start 0x00000000033c0000 End 0x00000000033dffff Tag VadS
Flags: CommitCharge: 32, PrivateMemory: 1, Protection: 24
Protection: PAGE_NOACCESS
Vad Type: VadNone
```

Q12: There was a VBS script that ran on the machine. What is the name of the script? (submit without file extension)

- **I used 'Cmdline' plugin to check the name of the script that executed by 'Wscript.exe'**

```
************************************************************************
wscript.exe pid:   5116
Command line : "C:\Windows\System32\wscript.exe" //B //NOLOGO %TEMP%\VhjReUDEuumrX.vbs
************************************************************************
```

Q13: An application was run at 2019-03-07 23:06:58 UTC. What is the name of the program? (Include extension)

- To address this question, I used voltaility3 'Shimcache' plugin which indicates about the executions that performed on the OS.

```
┌──(kali㉿kali)-[~/Desktop/volatility3]
└─$ python3 vol.py -f ../Triage-Memory.mem windows.shimcachemem.ShimcacheMem | grep '2019-03-07 23:06:58'
247gress2019-03-07 23:06:58.000000 UTC  N/A     True    N/A     \??\C:\Program Files (x86)\Microsoft\Skype for Desktop\Skype.exe
```

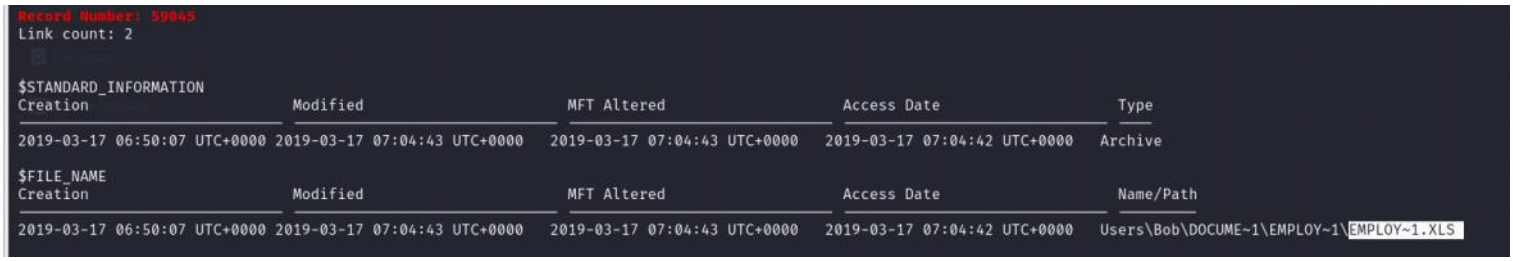
Q14: What was written in notepad.exe at the time when the memory dump was captured?

- Firstly, I used 'memdump' plugin to extract the memory of Notepad (PID:3032).
  Subsequently, I used strings via '-e l' flag to see the readable strings.

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ strings -e l 3032.dmp | grep flag
flag<REDBULL_IS_LIFE>
```

Q15: What is the short name of the file at file record 59045?

- I used 'MFTparser' plugin and searched the file record number by grep utility.



Q16: This box was exploited and is running meterpreter. What was the infected PID?

- We already found the answer at Q6, **3496**