Memory Dump Analysis of WannaCry Ransomware

Memory Analysis

To understand malware behavior during infection, security researcher and analyst will analyze the volatile memory of infected computer. This process is called memory analysis. With memory analysis there are several information analysts can retrieve useful information such as running processes, connection established and more detailed indicators of compromise (IOC). Especially in analyzing ransomware, memory analysis gives greater detail on how the ransomware works and behave.

Based on SANS (2015) *"Memory Forensics: Essential in Effective Incident Response Today"*, there are 6 (six) steps on doing memory analysis, identify rogue processes, analyze process DLLs and handles, review network artifacts, look for evidence of code injection, check for signs of rootkit and dump suspicious processes and drivers. In this section, analyst will perform analysis of a memory dump from a machine which was infected with WannaCry ransomware.

Tools and Environment

- Environment

  Analysis will be done in an isolated virtual machine running Ubuntu 20.04.

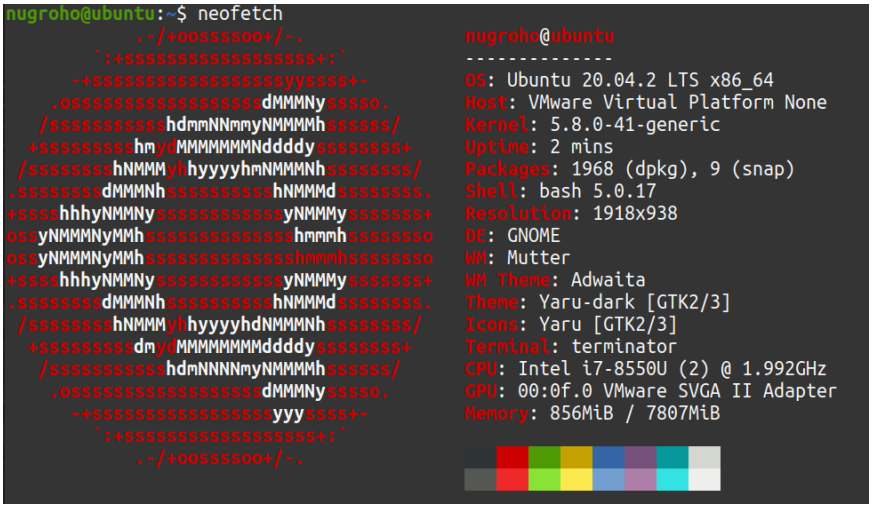


Figure 1: Analysis Environment

- Tools

  This analysis will utilize Volatility Framework as the main memory analysis tool.

Volatility is a tool developed back in 2007 and was firstly released in Black Hat DC. The software was developed based on tons of academic research into advanced forensics. Volatility introduced the power of analyzing volatile memory in the era where most of the investigation focus on hard drive images.

Analysis

Firstly, it is important to understand the profile of the machine which memory has been dumped. Volatility has a plugin to retrieve this information which is “imageinfo”.

```
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/nugroho/Documents/acs/mem/files/wcry.raw)
                      PAE type : No PAE
                           DTB : 0x39000L
                          KDBG : 0x8054cf60L
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2017-05-12 21:26:32 UTC+0000
     Image local date and time : 2017-05-13 02:56:32 +0530
```

Figure 1: Getting Profile

Now that the profile has been retrieved, further investigation will be faster. The next step is to analyze the process running during the ransomware infection. To do that, researcher will use ‘pslist’ plugin from volatility.

```
nugroho@ubuntu:~/Documents/acs/mem/files$ volatility -f wcry.raw --profile=WinXPSP3x86 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x823c8830 System                    4      0     51      244 ------      0
0x82169020 smss.exe                348      4      3       19 ------      0 2017-05-12 21:21:55 UTC+0000
0x82161da0 csrss.exe               596    348     12      352      0      0 2017-05-12 21:22:00 UTC+0000
0x8216e020 winlogon.exe            620    348     23      536      0      0 2017-05-12 21:22:01 UTC+0000
0x821937f0 services.exe            664    620     15      265      0      0 2017-05-12 21:22:01 UTC+0000
0x82191658 lsass.exe               676    620     23      353      0      0 2017-05-12 21:22:01 UTC+0000
0x8221a2c0 svchost.exe             836    664     19      211      0      0 2017-05-12 21:22:02 UTC+0000
0x821b5230 svchost.exe             904    664      9      227      0      0 2017-05-12 21:22:03 UTC+0000
0x821af7e8 svchost.exe            1024    664     79     1366      0      0 2017-05-12 21:22:03 UTC+0000
0x8203b7a8 svchost.exe            1084    664      6       72      0      0 2017-05-12 21:22:03 UTC+0000
0x821bea78 svchost.exe            1152    664     10      173      0      0 2017-05-12 21:22:06 UTC+0000
0x821e2da0 spoolsv.exe            1484    664     14      124      0      0 2017-05-12 21:22:09 UTC+0000
0x821d9da0 explorer.exe           1636   1608     11      331      0      0 2017-05-12 21:22:10 UTC+0000
0x82218da0 tasksche.exe           1940   1636      7       51      0      0 2017-05-12 21:22:14 UTC+0000
0x82231da0 ctfmon.exe             1956   1636      1       86      0      0 2017-05-12 21:22:14 UTC+0000
0x81fb95d8 svchost.exe             260    664      5      105      0      0 2017-05-12 21:22:18 UTC+0000
0x81fde308 @WanaDecryptor@         740   1940      2       70      0      0 2017-05-12 21:22:22 UTC+0000
0x81f747c0 wuauclt.exe            1768   1024      7      132      0      0 2017-05-12 21:22:52 UTC+0000
0x82010020 alg.exe                 544    664      6      101      0      0 2017-05-12 21:22:55 UTC+0000
0x81fea8a0 wscntfy.exe            1168   1024      1       37      0      0 2017-05-12 21:22:56 UTC+0000
```

Figure 2: Listing Running Processes

Here in the process list, researcher could identify a suspicious process called “@WanaDecryptor@”. This process is a parent process of “tasckssche.exe”. These two processes can be considered as a clear indicator of compromise (IOC). To dig deeper to these

processes, researcher will list out all processes including terminated processes with the Volatility plugin, 'psscan'.

```
nugroho@ubuntu:~/Documents/acs/mem/files$ volatility -f wcry.raw --profile=WinXPSP3x86 psscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P)          Name                PID   PPID PDB        Time created                   Time exited
------------------ ---------------- ------ ------ ---------- ------------------------------ ------------------------------
0x0000000001f4daf0 taskdl.exe          860   1940 0x199f6000 2017-05-12 21:26:23 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001f53d18 taskse.exe          536   1940 0x1986c000 2017-05-12 21:26:22 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001f69b50 @WanaDecryptor@     424   1940 0x18fa2000 2017-05-12 21:25:52 UTC+0000   2017-05-12 21:25:53 UTC+0000
0x0000000001f747c0 wuauclt.exe        1768   1024 0x11629000 2017-05-12 21:22:52 UTC+0000
0x0000000001f8ba58 @WanaDecryptor@     576   1940 0x19671000 2017-05-12 21:26:22 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001fb95d8 svchost.exe         260    664 0x0ce48000 2017-05-12 21:22:18 UTC+0000
0x0000000001fde308 @WanaDecryptor@     740   1940 0x0de3a000 2017-05-12 21:22:22 UTC+0000
0x0000000001fea8a0 wscntfy.exe        1168   1024 0x12217000 2017-05-12 21:22:56 UTC+0000
0x0000000002010020 alg.exe             544    664 0x1238d000 2017-05-12 21:22:55 UTC+0000
0x000000000203b7a8 svchost.exe        1084    664 0x0838c000 2017-05-12 21:22:03 UTC+0000
0x0000000002161da0 csrss.exe           596    348 0x07752000 2017-05-12 21:22:00 UTC+0000
0x0000000002169020 smss.exe            348      4 0x0683e000 2017-05-12 21:21:55 UTC+0000
0x000000000216e020 winlogon.exe        620    348 0x07957000 2017-05-12 21:22:01 UTC+0000
0x0000000002191658 lsass.exe           676    620 0x07bb7000 2017-05-12 21:22:01 UTC+0000
0x00000000021937f0 services.exe        664    620 0x07bad000 2017-05-12 21:22:01 UTC+0000
0x00000000021af7e8 svchost.exe        1024    664 0x081f7000 2017-05-12 21:22:03 UTC+0000
0x00000000021b5230 svchost.exe         904    664 0x08131000 2017-05-12 21:22:03 UTC+0000
0x00000000021bea78 svchost.exe        1152    664 0x08a15000 2017-05-12 21:22:06 UTC+0000
0x00000000021d9da0 explorer.exe       1636   1608 0x0add4000 2017-05-12 21:22:10 UTC+0000
0x00000000021e2da0 spoolsv.exe        1484    664 0x0a462000 2017-05-12 21:22:09 UTC+0000
0x0000000002218da0 tasksche.exe       1940   1636 0x0c0a2000 2017-05-12 21:22:14 UTC+0000
0x000000000221a2c0 svchost.exe         836    664 0x07e3e000 2017-05-12 21:22:02 UTC+0000
0x0000000002231da0 ctfmon.exe         1956   1636 0x0c01f000 2017-05-12 21:22:14 UTC+0000
0x00000000023c8830 System                4      0 0x00039000
```

Figure 3: Listing All Processes

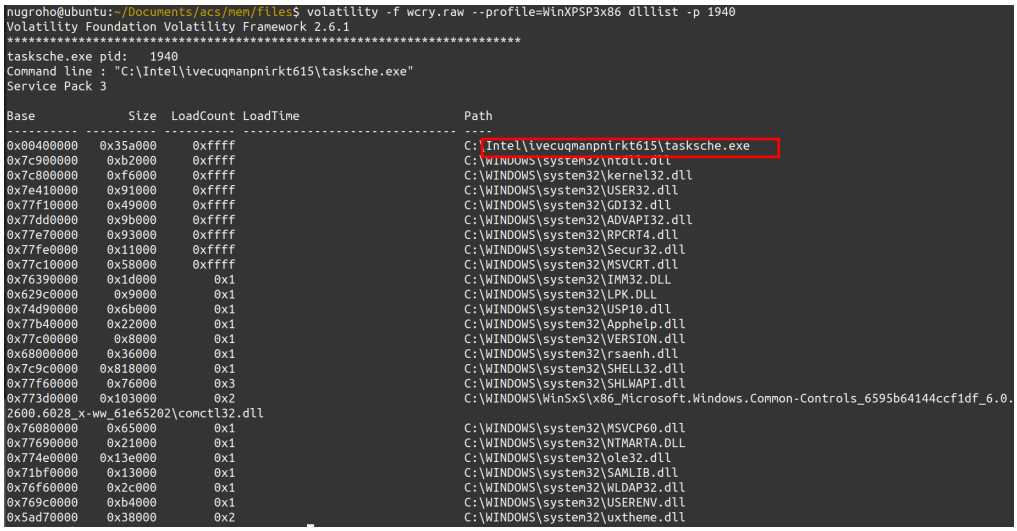
Several related processes which are terminated is listed out. The related terminated processes include *"taskdl.exe"*, *"taskse.exe"* and multiple *"@WanaDecryptor@"* processes. All processes mentioned are under the same parent process with PID 1940. To understand the timeline of the execution, researcher will sort the date and time of execution by using 'sort' command.

```
nugroho@ubuntu:~/Documents/acs/mem/files$ volatility -f wcry.raw --profile=WinXPSP3x86 psscan | grep 1940 | tee PPID1940.txt
Volatility Foundation Volatility Framework 2.6.1
0x0000000001f4daf0 taskdl.exe          860   1940 0x199f6000 2017-05-12 21:26:23 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001f53d18 taskse.exe          536   1940 0x1986c000 2017-05-12 21:26:22 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001f69b50 @WanaDecryptor@     424   1940 0x18fa2000 2017-05-12 21:25:52 UTC+0000   2017-05-12 21:25:53 UTC+0000
0x0000000001f8ba58 @WanaDecryptor@     576   1940 0x19671000 2017-05-12 21:26:22 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001fde308 @WanaDecryptor@     740   1940 0x0de3a000 2017-05-12 21:22:22 UTC+0000
0x0000000002218da0 tasksche.exe       1940   1636 0x0c0a2000 2017-05-12 21:22:14 UTC+0000
nugroho@ubuntu:~/Documents/acs/mem/files$ ls
24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c.bin.gz  PPID1940.txt  wannacry.7z  wcry.raw
nugroho@ubuntu:~/Documents/acs/mem/files$ cat PPID1940.txt
0x0000000001f4daf0 taskdl.exe          860   1940 0x199f6000 2017-05-12 21:26:23 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001f53d18 taskse.exe          536   1940 0x1986c000 2017-05-12 21:26:22 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001f69b50 @WanaDecryptor@     424   1940 0x18fa2000 2017-05-12 21:25:52 UTC+0000   2017-05-12 21:25:53 UTC+0000
0x0000000001f8ba58 @WanaDecryptor@     576   1940 0x19671000 2017-05-12 21:26:22 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001fde308 @WanaDecryptor@     740   1940 0x0de3a000 2017-05-12 21:22:22 UTC+0000
0x0000000002218da0 tasksche.exe       1940   1636 0x0c0a2000 2017-05-12 21:22:14 UTC+0000
nugroho@ubuntu:~/Documents/acs/mem/files$ sort -k 7,7 PPID1940.txt
0x0000000002218da0 tasksche.exe       1940   1636 0x0c0a2000 2017-05-12 21:22:14 UTC+0000
0x0000000001fde308 @WanaDecryptor@     740   1940 0x0de3a000 2017-05-12 21:22:22 UTC+0000
0x0000000001f69b50 @WanaDecryptor@     424   1940 0x18fa2000 2017-05-12 21:25:52 UTC+0000   2017-05-12 21:25:53 UTC+0000
0x0000000001f53d18 taskse.exe          536   1940 0x1986c000 2017-05-12 21:26:22 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001f8ba58 @WanaDecryptor@     576   1940 0x19671000 2017-05-12 21:26:22 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001f4daf0 taskdl.exe          860   1940 0x199f6000 2017-05-12 21:26:23 UTC+0000   2017-05-12 21:26:23 UTC+0000
```

Figure 4: Sorting Related Processes Execution Time

The sorted list shows that the parent process is executed and create multiple other processes which are *"@WanaDecryptor@"*, *"taskse.exe"* (terminated), and *"taskdl.exe"* (terminated). From this point, it can be seen that these are the processes that could be considered

as IOC. The next step is to identify the corresponding DLLs involved in those processes. To be able to do that, researcher will use the 'dlllist' plugin from Volatility.

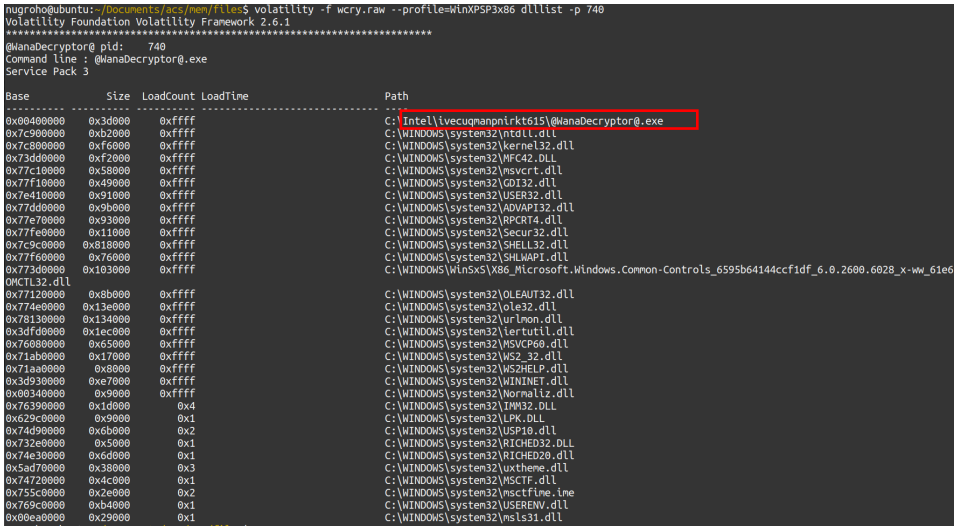


```
nugroho@ubuntu:~/Documents/acs/mem/files$ volatility -f wcry.raw --profile=WinXPSP3x86 dlllist -p 1940
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
tasksche.exe pid:   1940
Command line : "C:\Intel\ivecuqmanpnirkt615\tasksche.exe"
Service Pack 3

Base             Size  LoadCount LoadTime                       Path
---------- ---------- ---------- ------------------------------ ----
0x00400000   0x35a000     0xffff                                C:\Intel\ivecuqmanpnirkt615\tasksche.exe
0x7c900000    0xb2000     0xffff                                C:\WINDOWS\system32\ntdll.dll
0x7c800000    0xf6000     0xffff                                C:\WINDOWS\system32\kernel32.dll
0x7e410000    0x91000     0xffff                                C:\WINDOWS\system32\USER32.dll
0x77f10000    0x49000     0xffff                                C:\WINDOWS\system32\GDI32.dll
0x77dd0000    0x9b000     0xffff                                C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000    0x93000     0xffff                                C:\WINDOWS\system32\RPCRT4.dll
0x77fe0000    0x11000     0xffff                                C:\WINDOWS\system32\Secur32.dll
0x77c10000    0x58000     0xffff                                C:\WINDOWS\system32\MSVCRT.dll
0x76390000    0x1d000        0x1                                C:\WINDOWS\system32\IMM32.DLL
0x629c0000     0x9000        0x1                                C:\WINDOWS\system32\LPK.DLL
0x74d90000    0x6b000        0x1                                C:\WINDOWS\system32\USP10.dll
0x77b40000    0x22000        0x1                                C:\WINDOWS\system32\Apphelp.dll
0x77c00000     0x8000        0x1                                C:\WINDOWS\system32\VERSION.dll
0x68000000    0x36000        0x1                                C:\WINDOWS\system32\rsaenh.dll
0x7c9c0000   0x818000        0x1                                C:\WINDOWS\system32\SHELL32.dll
0x77f60000    0x76000        0x3                                C:\WINDOWS\system32\SHLWAPI.dll
0x773d0000   0x103000        0x2                                C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.
2600.6028_x-ww_61e65202\comctl32.dll
0x76080000    0x65000        0x1                                C:\WINDOWS\system32\MSVCP60.dll
0x77690000    0x21000        0x1                                C:\WINDOWS\system32\NTMARTA.DLL
0x774e0000   0x13e000        0x1                                C:\WINDOWS\system32\ole32.dll
0x71bf0000    0x13000        0x1                                C:\WINDOWS\system32\SAMLIB.dll
0x76f60000    0x2c000        0x1                                C:\WINDOWS\system32\WLDAP32.dll
0x769c0000    0xb4000        0x1                                C:\WINDOWS\system32\USERENV.dll
0x5ad70000    0x38000        0x2                                C:\WINDOWS\system32\uxtheme.dll
```

Figure 5: DLLs Loaded for *"tasksche.exe"*

From the output, research could spot a suspicious binary path used to execute the parent process of all related processes. It was executed from *"C:\Intel\ivecuqmanpnirkt615\"*. Looking at the *"@WanaDecryptor@"* process DLLs, the same path can be found.



```
nugroho@ubuntu:~/Documents/acs/mem/files$ volatility -f wcry.raw --profile=WinXPSP3x86 dlllist -p 740
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
@WanaDecryptor@ pid:    740
Command line : @WanaDecryptor@.exe
Service Pack 3

Base             Size  LoadCount LoadTime                       Path
---------- ---------- ---------- ------------------------------ ----
0x00400000    0x3d000     0xffff                                C:\Intel\ivecuqmanpnirkt615\@WanaDecryptor@.exe
0x7c900000    0xb2000     0xffff                                C:\WINDOWS\system32\ntdll.dll
0x7c800000    0xf6000     0xffff                                C:\WINDOWS\system32\kernel32.dll
0x73dd0000    0xf2000     0xffff                                C:\WINDOWS\system32\MFC42.DLL
0x77c10000    0x58000     0xffff                                C:\WINDOWS\system32\msvcrt.dll
0x77f10000    0x49000     0xffff                                C:\WINDOWS\system32\GDI32.dll
0x7e410000    0x91000     0xffff                                C:\WINDOWS\system32\USER32.dll
0x77dd0000    0x9b000     0xffff                                C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000    0x93000     0xffff                                C:\WINDOWS\system32\RPCRT4.dll
0x77fe0000    0x11000     0xffff                                C:\WINDOWS\system32\Secur32.dll
0x7c9c0000   0x818000     0xffff                                C:\WINDOWS\system32\SHELL32.dll
0x77f60000    0x76000     0xffff                                C:\WINDOWS\system32\SHLWAPI.dll
0x773d0000   0x103000     0xffff                                C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.6028_x-ww_61e6
OMCTL32.dll
0x77120000    0x8b000     0xffff                                C:\WINDOWS\system32\OLEAUT32.dll
0x774e0000   0x13e000     0xffff                                C:\WINDOWS\system32\ole32.dll
0x78130000   0x134000     0xffff                                C:\WINDOWS\system32\urlmon.dll
0x3dfd0000   0x1ec000     0xffff                                C:\WINDOWS\system32\iertutil.dll
0x76080000    0x65000     0xffff                                C:\WINDOWS\system32\MSVCP60.dll
0x71ab0000    0x17000     0xffff                                C:\WINDOWS\system32\WS2_32.dll
0x71aa0000     0x8000     0xffff                                C:\WINDOWS\system32\WS2HELP.dll
0x3d930000    0xe7000     0xffff                                C:\WINDOWS\system32\WININET.dll
0x00340000     0x9000     0xffff                                C:\WINDOWS\system32\Normaliz.dll
0x76390000    0x1d000        0x4                                C:\WINDOWS\system32\IMM32.DLL
0x629c0000     0x9000        0x1                                C:\WINDOWS\system32\LPK.DLL
0x74d90000    0x6b000        0x2                                C:\WINDOWS\system32\USP10.dll
0x732e0000     0x5000        0x1                                C:\WINDOWS\system32\RICHED32.DLL
0x74e30000    0x6d000        0x1                                C:\WINDOWS\system32\RICHED20.dll
0x5ad70000    0x38000        0x3                                C:\WINDOWS\system32\uxtheme.dll
0x74720000    0x4c000        0x1                                C:\WINDOWS\system32\MSCTF.dll
0x755c0000    0x2e000        0x2                                C:\WINDOWS\system32\msctfime.ime
0x769c0000    0xb4000        0x1                                C:\WINDOWS\system32\USERENV.dll
0x00ea0000    0x29000        0x1                                C:\WINDOWS\system32\msls31.dll
```

Figure 6: DLLs Loaded for *"@WanaDecryptor@"*

By looking at DLLs loaded by *"@WanaDecryptor@"* or PID 740, the functionality of the ransomware could be identified. It loaded *"WS2_32.DLL"* which related to socket connection creation (Doevan, 2018). It also utilizes *"WINNET.DLL"* to establish high privilege network interactions, *"SECURE32.DLL"* for encryption and *"URLMON.DLL"* to

interact with browsers. Information mentioned above will be useful during the malware analysis process.

The next step is to analyze the process' handles. By investigating handles, some additional information about threads, and mutex (Mutual Exclusion) could be found. Mutex is a mechanism which allows a program to access resources (GeeksforGeeks, 2020). In a malware, a mutex is used to prevent multiple instance of the same malware running on the victim machine.

```
nugroho@ubuntu:~/Documents/acs/mem/files$ volatility -f wcry.raw --profile=WinXPSP3x86 handles -p 1940 -t key
Volatility Foundation Volatility Framework 2.6.1
Offset(V)     Pid     Handle     Access Type             Details
---------- ------ ---------- ---------- ---------------- -------
0xe1a05938   1940       0x30  0x20f003f Key              MACHINE
0xe1b978d0   1940       0xc4  0x20f003f Key              USER\S-1-5-21-602162358-764733703-1957994488-1003
nugroho@ubuntu:~/Documents/acs/mem/files$ volatility -f wcry.raw --profile=WinXPSP3x86 handles -p 1940 -t Mutant
Volatility Foundation Volatility Framework 2.6.1
Offset(V)     Pid     Handle     Access Type             Details
---------- ------ ---------- ---------- ---------------- -------
0x821883e8   1940       0x40   0x120001 Mutant           ShimCacheMutex
0x8224f180   1940       0x54   0x1f0001 Mutant           MsWinZonesCacheCounterMutexA
0x822e3b08   1940       0x58   0x1f0001 Mutant           MsWinZonesCacheCounterMutexA0
```

Figure 7: Mutex for PID 1940

It can be seen from Figure 7, the parent process with PID 1940 have mutex *"MsWinZonesCacheCounterMutexA"*. This mutex prevents the system to run multiple malware instances. The next step is to analyze the network connections established during infections. This can be done with Volatility plugin 'connections' and 'connscan'.

```
nugroho@ubuntu:~/Documents/acs/mem/files$ volatility -f wcry.raw --profile=WinXPSP3x86 connections
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
nugroho@ubuntu:~/Documents/acs/mem/files$ volatility -f wcry.raw --profile=WinXPSP3x86 connscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
```

Figure 8: Network Connections from Memory Dump

There is no valuable information from the network connections. This could be due to the inactivity of the server which is supposed to interact with the ransomware. Next thing to do is to dump interesting files that can later be analyzed through static or dynamic reverse engineering.

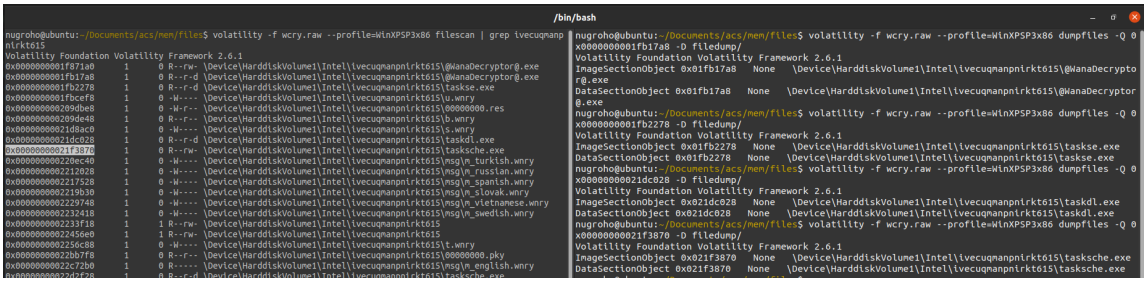


Figure 9: Dumping Files Related to Ransomware

Running a simple ‘strings’ command will show some interesting strings inside the dumped files. It includes some command, links and even bitcoin address.

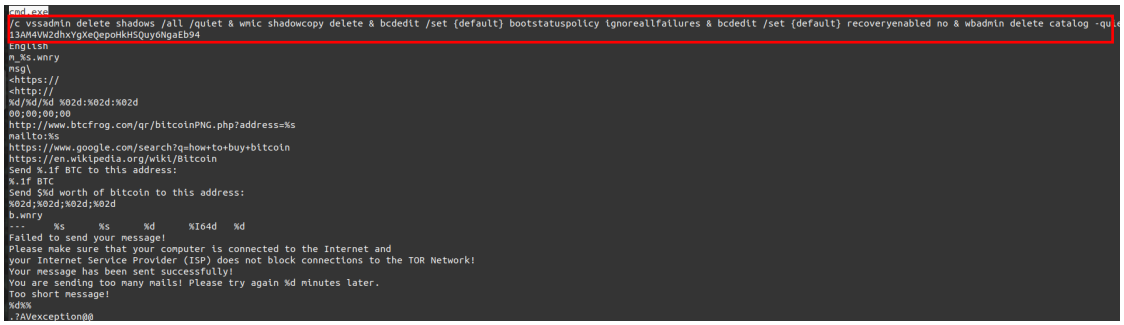


Figure 10: Inside Dumped Files

As part of the memory analysis, researcher will also dump the memory of specific processes. This includes the *‘tasksche.exe’* process with PID 1940 and *‘@WanaDecryptor@’* PID 740. The memory dump could reveal more information regarding the execution of the ransomware.

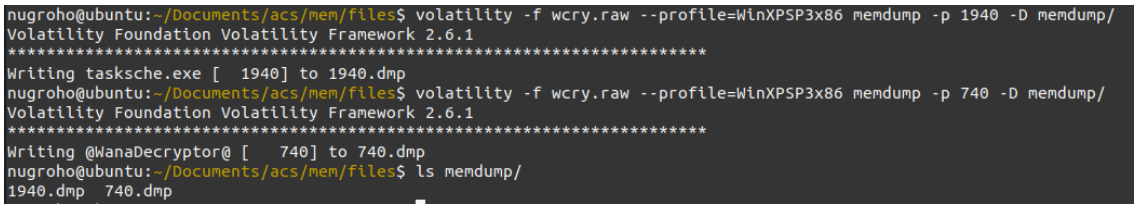


Figure 11: Dumping Memory of Specific Process

Inside the memory dump of PID 1940, researcher could find the execution of the binary *‘@WanaDecryptor@’* using command line. Looking at the memory dump of *‘@WanaDecryptor@’*, listed out several ‘.onion’ domain. It also shows the bitcoin address.

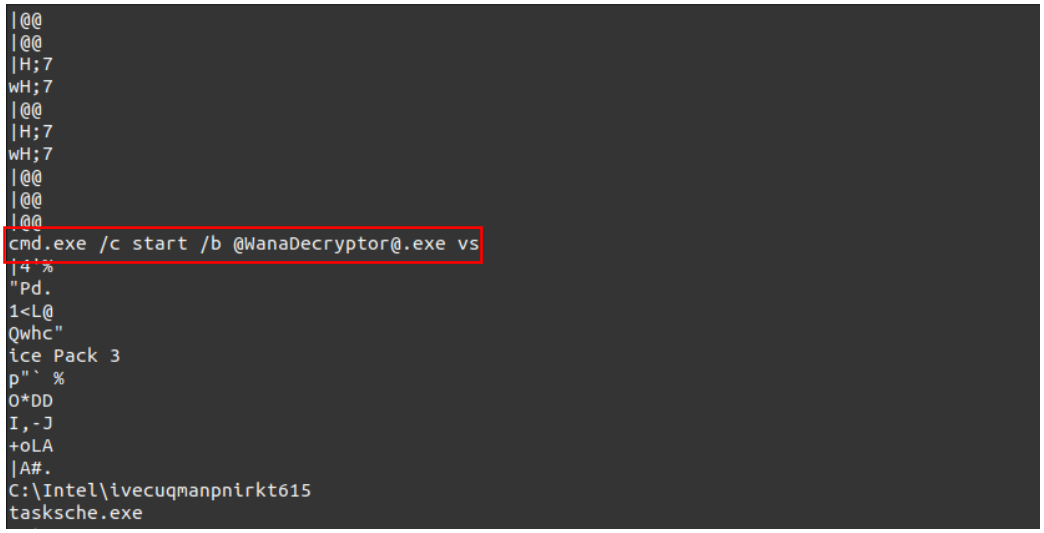


Figure 12: Memory Dump of PID 1940

Figure 13: Memory Dump of PID 740

Digging deeper, researcher found the ransom note with links to the bitcoin address. http://www.btcfrog.com/qr/bitcoinPNG.php?address=12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw.

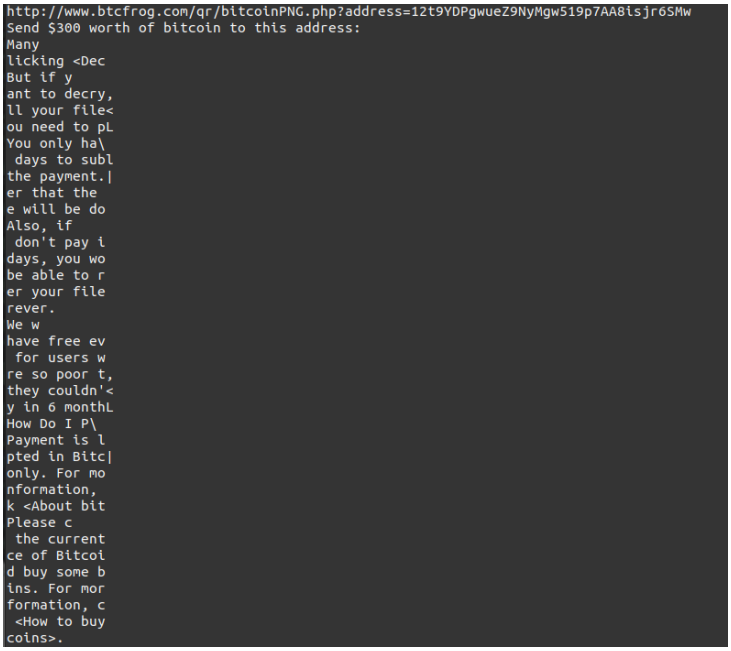


Figure 14: Ransom Note

It also contains the mutex generation and registry key creation for the ransomware.

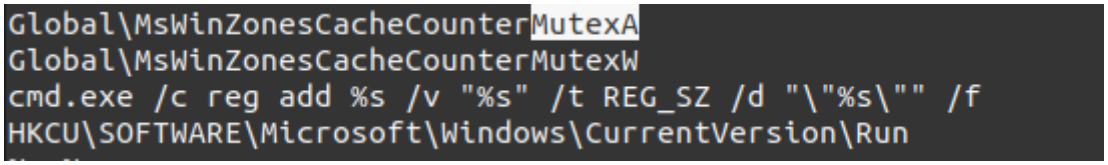


Figure 15: Mutex and Registry Key

2.2.3 Conclusion

To sum it up, memory analysis is done to understand the way of a malware behaves. It uses the memory dump of infected computer to investigate using memory analysis tools such as Volatility. There are several steps on doing memory analysis including identify rogue processes, analyze process DLLs and handles, review network artifacts, look for evidence of code injection, check for signs of rootkit and dump suspicious processes and drivers. In this analysis, information gathered from the memory related to its infection of WannaCry ransomware are as follow:

Indicators of Compromise (IOC):

- tasksche.exe
- taskse.exe
- taskdl.exe

- @WanaDecryptor@
- MsWinZonesCacheCounterMutexA
- MsWinZonesCacheCounterMutexA0
- 12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
- 13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94
- gx7ekbenv2riucmf.onion
- 76jdd2ir2embyv47.onion
- cwwnhwhlz52maqm7.onion
- 57g7spgrzlojinas.onion
- xxlvbrloxvriy2c5.onion

Suspicious files:

- tasksche.exe
- taskse.exe
- taskdl.exe
- @WanaDecryptor@.exe

References

Doevan, J., 2018. *What is ws2_32.dll? Should I remove it?.* [Online]
Available at: https://www.2-spyware.com/file-ws2_32-dll.html
[Accessed 10 February 2021].

GeeksforGeeks, 2020. *Mutex vs Semaphore.* [Online]
Available at: https://www.geeksforgeeks.org/mutex-vs-semaphore/
[Accessed 10 February 2021].