West Chester University

CSC 471

Dr. Si Chen

Spring 2024 Final Project

Submitted by

Sean Berlin

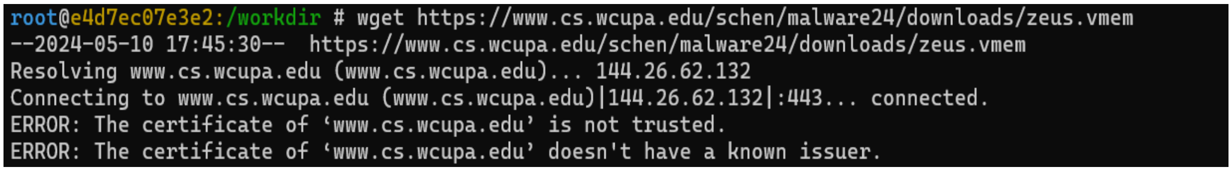
5/10/24

1. Introduction:

The purpose of this lab is to undertake a comprehensive analysis of the notorious Trojan malware known as Zeus. Utilizing the Volatility framework, we aim to dissect the intricacies of this malware by examining the provided memory dump file named zeus.vmem. Drawing parallels with the material covered in class, which delved into the subject of memory forensics, our objective is to decipher the underlying operations of the virus. Through meticulous scrutiny of the memory dump file, we seek to identify functions and lines of code indicative of malicious activity, unraveling the inner workings of this nefarious entity.

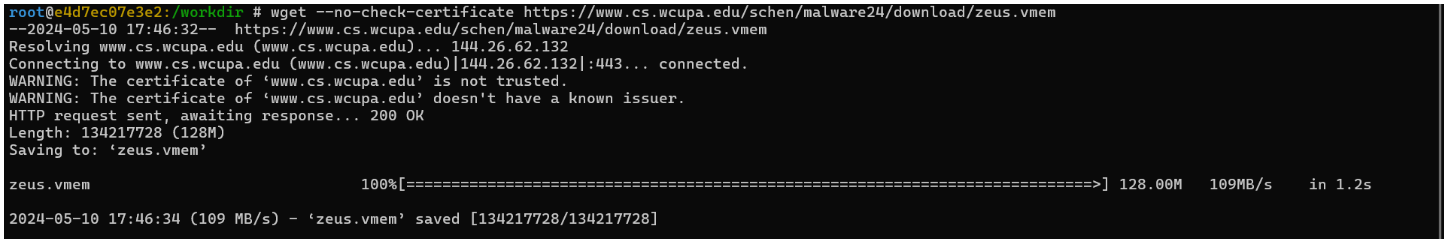
2. Analysis and Results:

The very first step taken was to download the “zeus.vmem” image file from the class website. Using the standard “wget” request caused the image to not be able to download due to the certificate on the class website not being trusted.



```
root@e4d7ec07e3e2:/workdir # wget https://www.cs.wcupa.edu/schen/malware24/downloads/zeus.vmem
--2024-05-10 17:45:30--  https://www.cs.wcupa.edu/schen/malware24/downloads/zeus.vmem
Resolving www.cs.wcupa.edu (www.cs.wcupa.edu)... 144.26.62.132
Connecting to www.cs.wcupa.edu (www.cs.wcupa.edu)|144.26.62.132|:443... connected.
ERROR: The certificate of ‘www.cs.wcupa.edu’ is not trusted.
ERROR: The certificate of ‘www.cs.wcupa.edu’ doesn't have a known issuer.
```

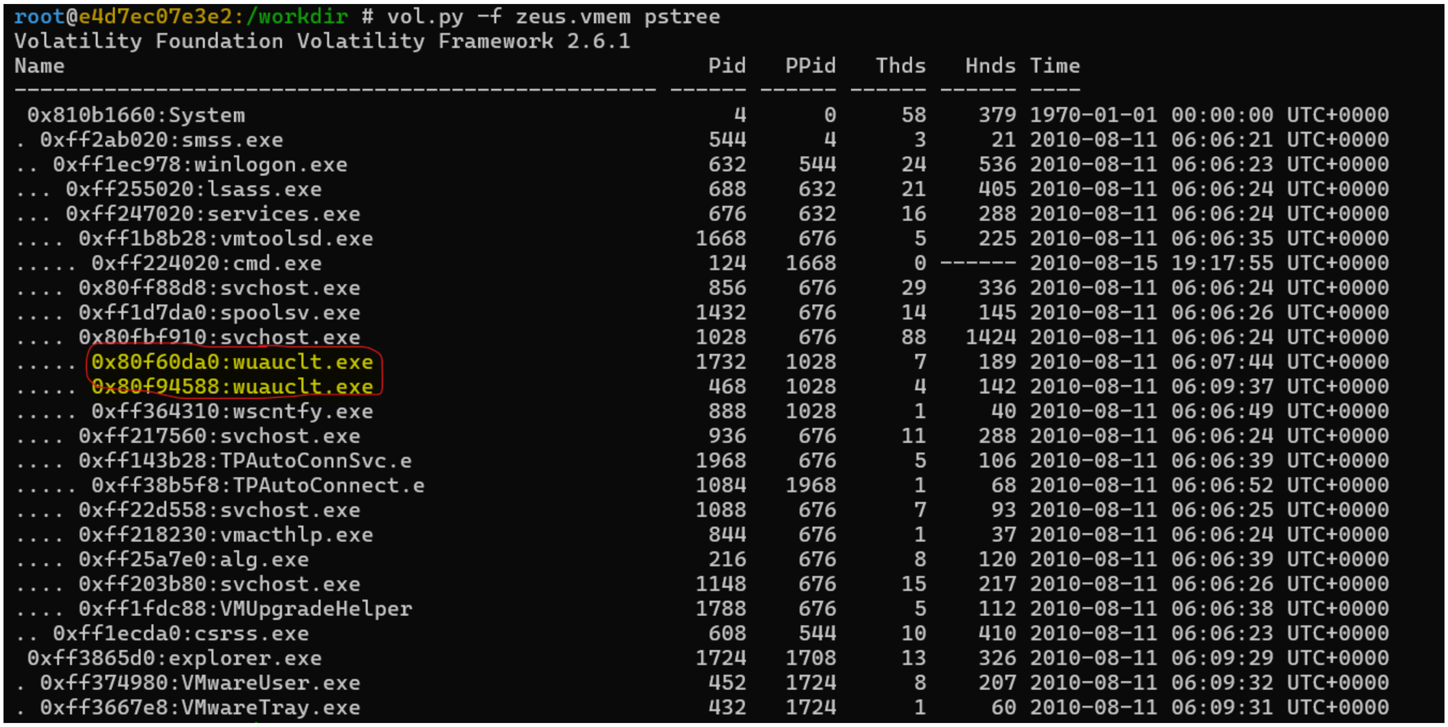
To work around this a “--no-check-certificate” flag was added to the previous command and the download was successful.



```
root@e4d7ec07e3e2:/workdir # wget --no-check-certificate https://www.cs.wcupa.edu/schen/malware24/download/zeus.vmem
--2024-05-10 17:46:32--  https://www.cs.wcupa.edu/schen/malware24/download/zeus.vmem
Resolving www.cs.wcupa.edu (www.cs.wcupa.edu)... 144.26.62.132
Connecting to www.cs.wcupa.edu (www.cs.wcupa.edu)|144.26.62.132|:443... connected.
WARNING: The certificate of ‘www.cs.wcupa.edu’ is not trusted.
WARNING: The certificate of ‘www.cs.wcupa.edu’ doesn't have a known issuer.
HTTP request sent, awaiting response... 200 OK
Length: 134217728 (128M)
Saving to: ‘zeus.vmem’

zeus.vmem                              100%[=============================================================================>] 128.00M   109MB/s    in 1.2s

2024-05-10 17:46:34 (109 MB/s) - ‘zeus.vmem’ saved [134217728/134217728]
```

The next step was to list all the devices using the Volatility framework with the command “vol.py -f zeus.vmem devicetree”. The complete result of the command is not shown in the following screenshot due to the extended length.

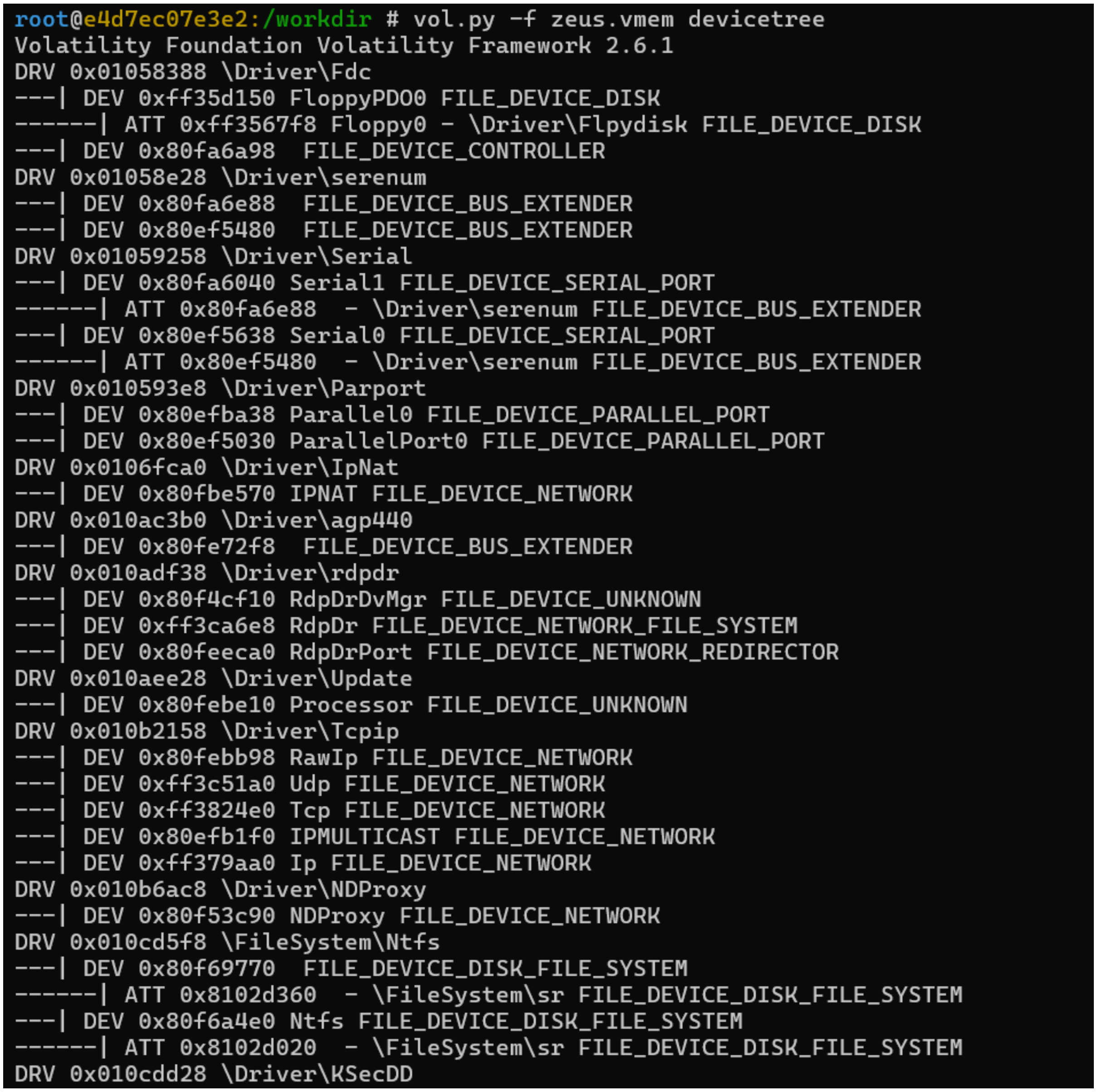
Next, the “vol.py -f zeuz.vmem pstree” was run to list all the processes in the system on the given memory dump file. Additionally, after analyzing the processes we encounter that a new file called “sr” is being installed.



```
root@e4d7ec07e3e2:/workdir # vol.py -f zeus.vmem pstree
Volatility Foundation Volatility Framework 2.6.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x810b1660:System                                      4      0     58    379 1970-01-01 00:00:00 UTC+0000
. 0xff2ab020:smss.exe                                 544      4      3     21 2010-08-11 06:06:21 UTC+0000
.. 0xff1ec978:winlogon.exe                            632    544     24    536 2010-08-11 06:06:23 UTC+0000
... 0xff255020:lsass.exe                              688    632     21    405 2010-08-11 06:06:24 UTC+0000
... 0xff247020:services.exe                           676    632     16    288 2010-08-11 06:06:24 UTC+0000
.... 0xff1b8b28:vmtoolsd.exe                         1668    676      5    225 2010-08-11 06:06:35 UTC+0000
..... 0xff224020:cmd.exe                              124   1668      0 ------ 2010-08-15 19:17:55 UTC+0000
.... 0x80ff88d8:svchost.exe                           856    676     29    336 2010-08-11 06:06:24 UTC+0000
.... 0xff1d7da0:spoolsv.exe                          1432    676     14    145 2010-08-11 06:06:26 UTC+0000
.... 0x80fbf910:svchost.exe                          1028    676     88   1424 2010-08-11 06:06:24 UTC+0000
..... 0x80f60da0:wuauclt.exe                         1732   1028      7    189 2010-08-11 06:07:44 UTC+0000
..... 0x80f94588:wuauclt.exe                          468   1028      4    142 2010-08-11 06:09:37 UTC+0000
..... 0xff364310:wscntfy.exe                          888   1028      1     40 2010-08-11 06:06:49 UTC+0000
.... 0xff217560:svchost.exe                           936    676     11    288 2010-08-11 06:06:24 UTC+0000
.... 0xff143b28:TPAutoConnSvc.e                      1968    676      5    106 2010-08-11 06:06:39 UTC+0000
..... 0xff38b5f8:TPAutoConnect.e                     1084   1968      1     68 2010-08-11 06:06:52 UTC+0000
.... 0xff22d558:svchost.exe                          1088    676      7     93 2010-08-11 06:06:25 UTC+0000
.... 0xff218230:vmacthlp.exe                          844    676      1     37 2010-08-11 06:06:24 UTC+0000
.... 0xff25a7e0:alg.exe                               216    676      8    120 2010-08-11 06:06:39 UTC+0000
.... 0xff203b80:svchost.exe                          1148    676     15    217 2010-08-11 06:06:26 UTC+0000
.... 0xff1fdc88:VMUpgradeHelper                      1788    676      5    112 2010-08-11 06:06:38 UTC+0000
.. 0xff1ecda0:csrss.exe                               608    544     10    410 2010-08-11 06:06:23 UTC+0000
 0xff3865d0:explorer.exe                             1724   1708     13    326 2010-08-11 06:09:29 UTC+0000
. 0xff374980:VMwareUser.exe                           452   1724      8    207 2010-08-11 06:09:32 UTC+0000
. 0xff3667e8:VMwareTray.exe                           432   1724      1     60 2010-08-11 06:09:31 UTC+0000
```

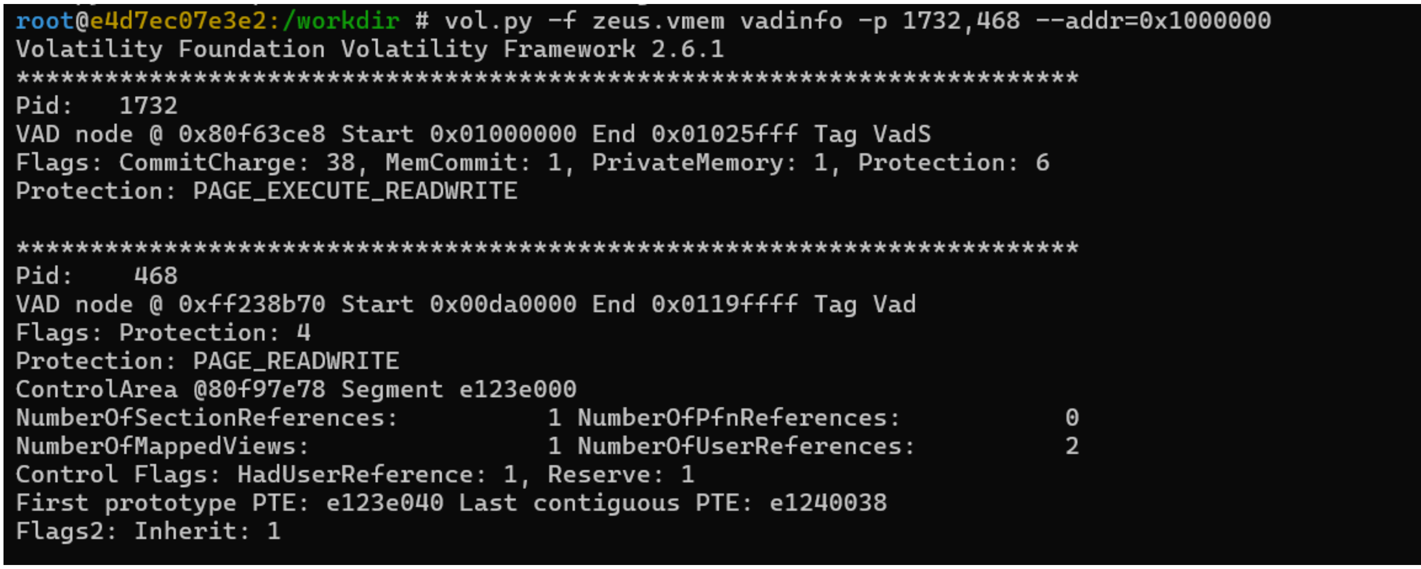
Following an in-depth analysis, the realization is that there are two processes with the same executable “wuaclt.exe” as highlighted and circled. This executable file is associated with Windows operating systems, specifically Windows Update. To make it clear, it stands for "Windows Update Automatic Updates Client." It appears the primary function of wuaclt.exe is to manage and control Windows Update operations.

Next, to find out which “wuaclt.exe” is the copy we and not legitimate, the “vol.py -f zeus.vmem vadinfo -p 1732,468” command is used to list their Pid. It is important to note this command is using the Virtual address descriptor.



```
root@e4d7ec07e3e2:/workdir # vol.py -f zeus.vmem devicetree
Volatility Foundation Volatility Framework 2.6.1
DRV 0x01058388 \Driver\Fdc
---| DEV 0xff35d150 FloppyPDO0 FILE_DEVICE_DISK
------| ATT 0xff3567f8 Floppy0 - \Driver\Flpydisk FILE_DEVICE_DISK
---| DEV 0x80fa6a98  FILE_DEVICE_CONTROLLER
DRV 0x01058e28 \Driver\serenum
---| DEV 0x80fa6e88  FILE_DEVICE_BUS_EXTENDER
---| DEV 0x80ef5480  FILE_DEVICE_BUS_EXTENDER
DRV 0x01059258 \Driver\Serial
---| DEV 0x80fa6040 Serial1 FILE_DEVICE_SERIAL_PORT
------| ATT 0x80fa6e88  - \Driver\serenum FILE_DEVICE_BUS_EXTENDER
---| DEV 0x80ef5638 Serial0 FILE_DEVICE_SERIAL_PORT
------| ATT 0x80ef5480  - \Driver\serenum FILE_DEVICE_BUS_EXTENDER
DRV 0x010593e8 \Driver\Parport
---| DEV 0x80efba38 Parallel0 FILE_DEVICE_PARALLEL_PORT
---| DEV 0x80ef5030 ParallelPort0 FILE_DEVICE_PARALLEL_PORT
DRV 0x0106fca0 \Driver\IpNat
---| DEV 0x80fbe570 IPNAT FILE_DEVICE_NETWORK
DRV 0x010ac3b0 \Driver\agp440
---| DEV 0x80fe72f8  FILE_DEVICE_BUS_EXTENDER
DRV 0x010adf38 \Driver\rdpdr
---| DEV 0x80f4cf10 RdpDrDvMgr FILE_DEVICE_UNKNOWN
---| DEV 0xff3ca6e8 RdpDr FILE_DEVICE_NETWORK_FILE_SYSTEM
---| DEV 0x80feeca0 RdpDrPort FILE_DEVICE_NETWORK_REDIRECTOR
DRV 0x010aee28 \Driver\Update
---| DEV 0x80febe10 Processor FILE_DEVICE_UNKNOWN
DRV 0x010b2158 \Driver\Tcpip
---| DEV 0x80febb98 RawIp FILE_DEVICE_NETWORK
---| DEV 0xff3c51a0 Udp FILE_DEVICE_NETWORK
---| DEV 0xff3824e0 Tcp FILE_DEVICE_NETWORK
---| DEV 0x80efb1f0 IPMULTICAST FILE_DEVICE_NETWORK
---| DEV 0xff379aa0 Ip FILE_DEVICE_NETWORK
DRV 0x010b6ac8 \Driver\NDProxy
---| DEV 0x80f53c90 NDProxy FILE_DEVICE_NETWORK
DRV 0x010cd5f8 \FileSystem\Ntfs
---| DEV 0x80f69770  FILE_DEVICE_DISK_FILE_SYSTEM
------| ATT 0x8102d360  - \FileSystem\sr FILE_DEVICE_DISK_FILE_SYSTEM
---| DEV 0x80f6a4e0 Ntfs FILE_DEVICE_DISK_FILE_SYSTEM
------| ATT 0x8102d020  - \FileSystem\sr FILE_DEVICE_DISK_FILE_SYSTEM
DRV 0x010cdd28 \Driver\KSecDD
```

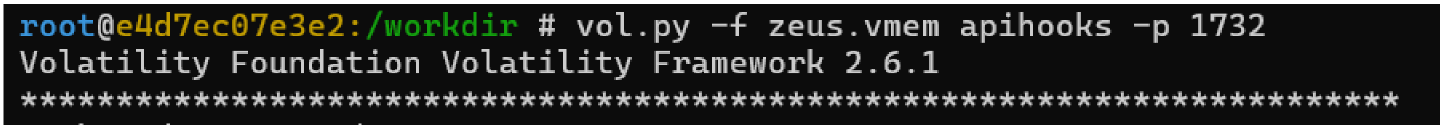
To make a more precise analysis, the following command is used to add some restrictions: “vol.py -f zeus.vmem vadinfo -p 1732,468 --addr=0x1000000”



```
root@e4d7ec07e3e2:/workdir # vol.py -f zeus.vmem vadinfo -p 1732,468 --addr=0x1000000
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
Pid:   1732
VAD node @ 0x80f63ce8 Start 0x01000000 End 0x01025fff Tag VadS
Flags: CommitCharge: 38, MemCommit: 1, PrivateMemory: 1, Protection: 6
Protection: PAGE_EXECUTE_READWRITE

************************************************************************
Pid:    468
VAD node @ 0xff238b70 Start 0x00da0000 End 0x0119ffff Tag Vad
Flags: Protection: 4
Protection: PAGE_READWRITE
ControlArea @80f97e78 Segment e123e000
NumberOfSectionReferences:           1 NumberOfPfnReferences:           0
NumberOfMappedViews:                 1 NumberOfUserReferences:          2
Control Flags: HadUserReference: 1, Reserve: 1
First prototype PTE: e123e040 Last contiguous PTE: e1240038
Flags2: Inherit: 1
```

To analyze the returned information, it can be concluded that process 1732 is not legitimate because it has some missing information which includes: ControlArea, NumberOfSectionReferences, NumberOfMappedViews, ControlFlags, First prototype, and Flags2.

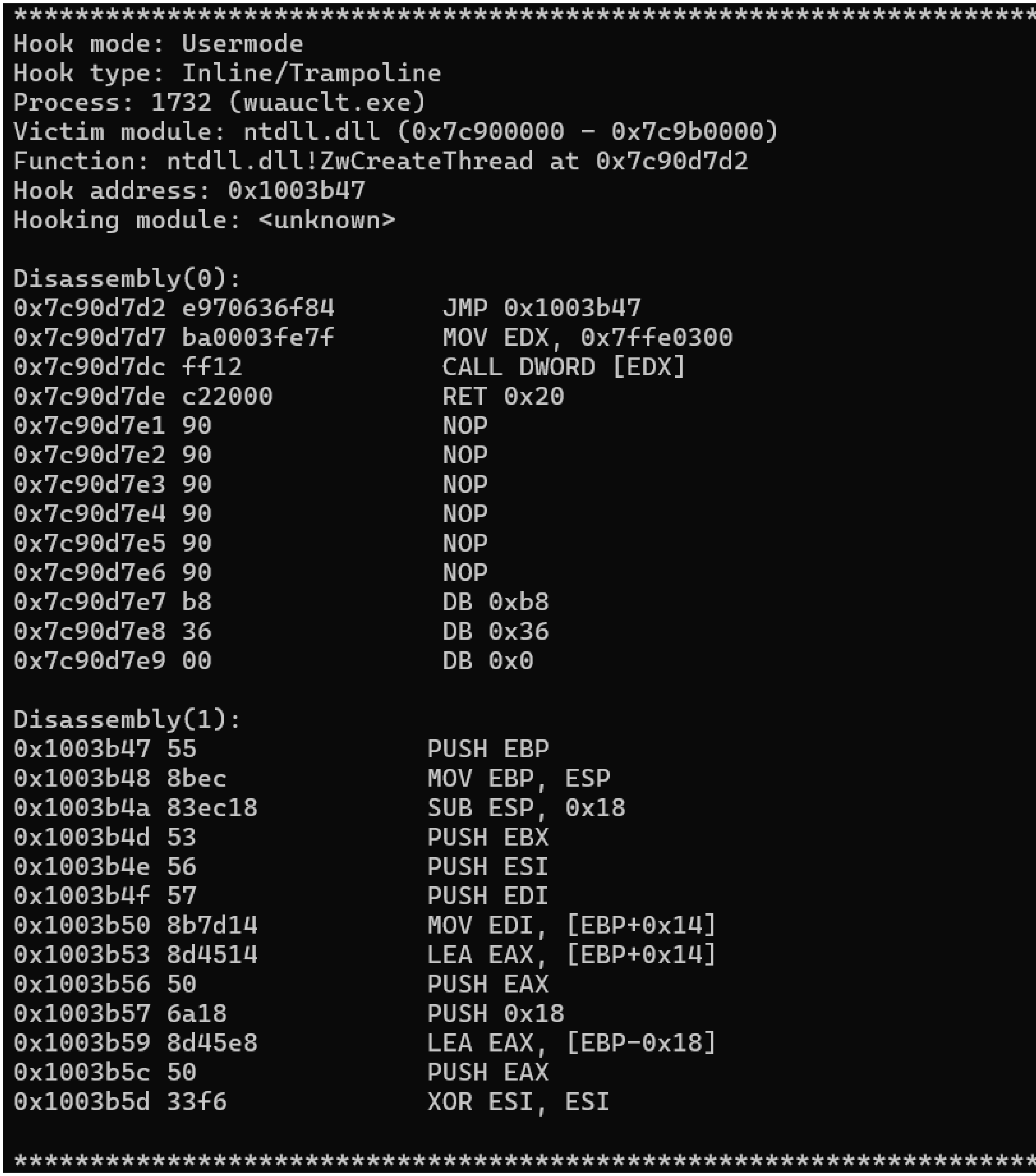
The next step is to identify the API hooks used by the given memory dump file. This can be performed by running the “vol.py -f zeus.vmem apihooks -p 1732” command.



```
root@e4d7ec07e3e2:/workdir # vol.py -f zeus.vmem apihooks -p 1732
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
```

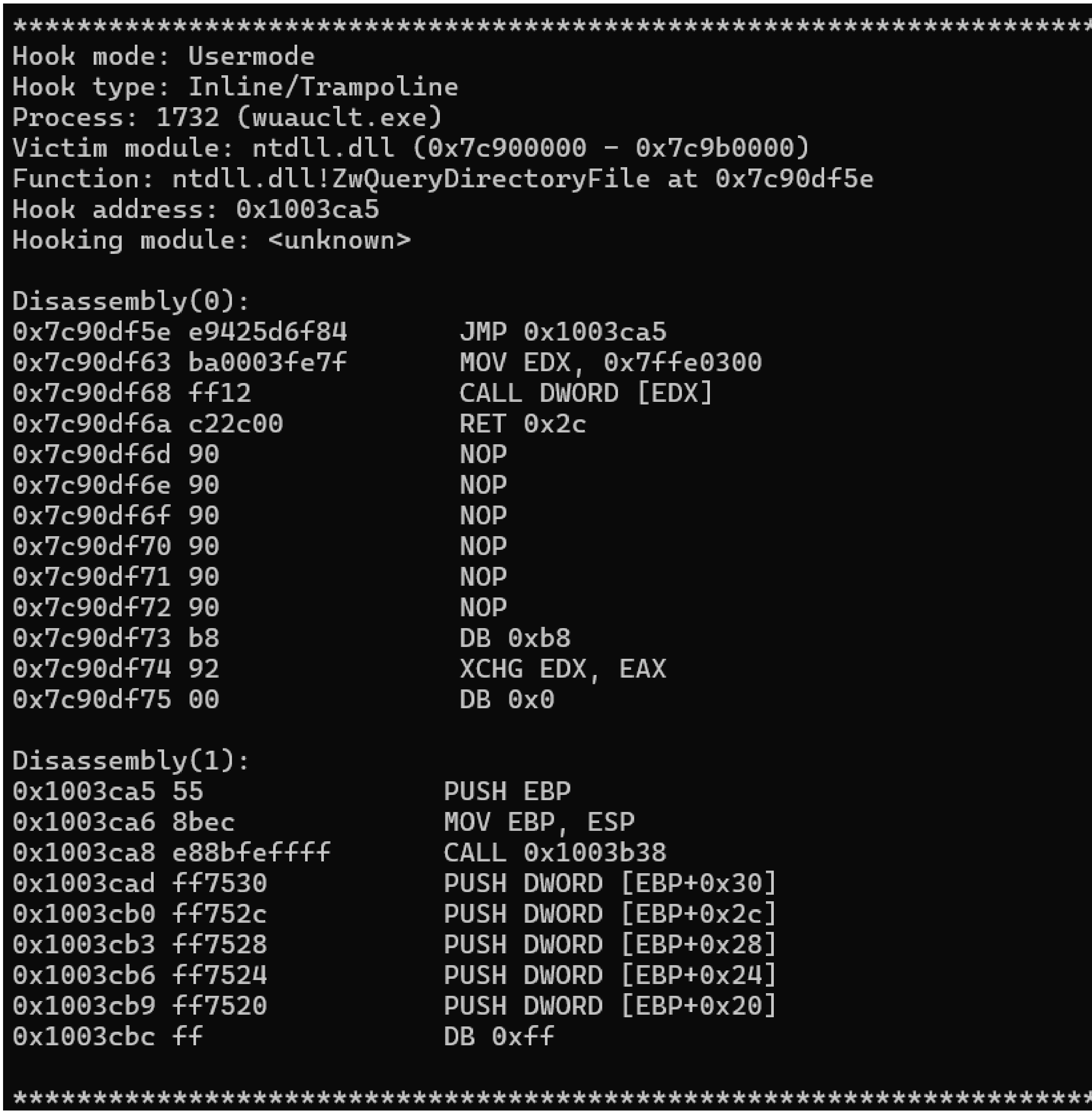
This reveals some suspicious API hooks:

ZwCreateThread:



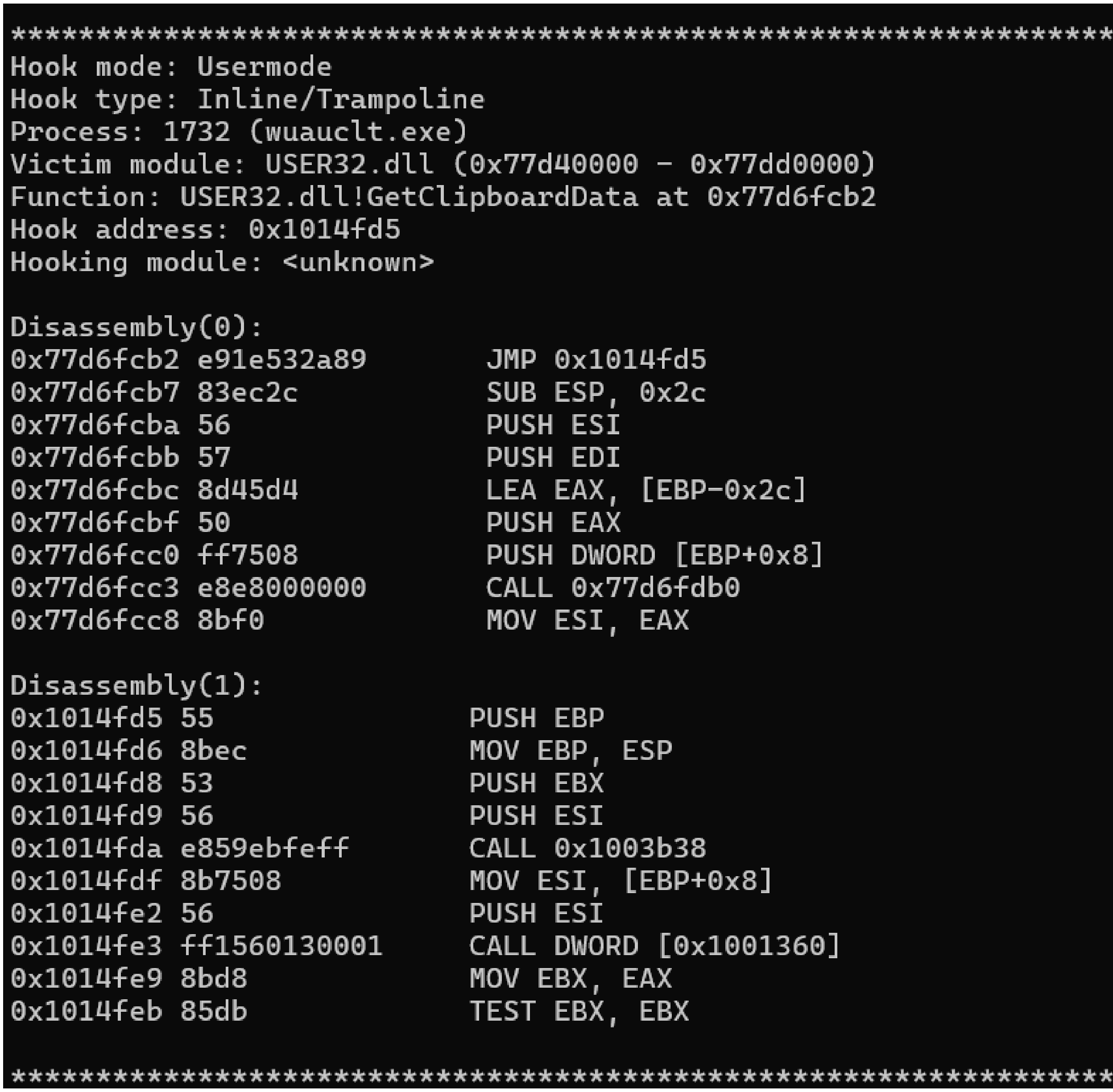
```
*******************************************************************
Hook mode: Usermode
Hook type: Inline/Trampoline
Process: 1732 (wuauclt.exe)
Victim module: ntdll.dll (0x7c900000 - 0x7c9b0000)
Function: ntdll.dll!ZwCreateThread at 0x7c90d7d2
Hook address: 0x1003b47
Hooking module: <unknown>

Disassembly(0):
0x7c90d7d2 e970636f84       JMP 0x1003b47
0x7c90d7d7 ba0003fe7f       MOV EDX, 0x7ffe0300
0x7c90d7dc ff12             CALL DWORD [EDX]
0x7c90d7de c22000           RET 0x20
0x7c90d7e1 90               NOP
0x7c90d7e2 90               NOP
0x7c90d7e3 90               NOP
0x7c90d7e4 90               NOP
0x7c90d7e5 90               NOP
0x7c90d7e6 90               NOP
0x7c90d7e7 b8               DB 0xb8
0x7c90d7e8 36               DB 0x36
0x7c90d7e9 00               DB 0x0

Disassembly(1):
0x1003b47 55               PUSH EBP
0x1003b48 8bec             MOV EBP, ESP
0x1003b4a 83ec18           SUB ESP, 0x18
0x1003b4d 53               PUSH EBX
0x1003b4e 56               PUSH ESI
0x1003b4f 57               PUSH EDI
0x1003b50 8b7d14           MOV EDI, [EBP+0x14]
0x1003b53 8d4514           LEA EAX, [EBP+0x14]
0x1003b56 50               PUSH EAX
0x1003b57 6a18             PUSH 0x18
0x1003b59 8d45e8           LEA EAX, [EBP-0x18]
0x1003b5c 50               PUSH EAX
0x1003b5d 33f6             XOR ESI, ESI

*******************************************************************
```

ZwQueryDirectoryFile:



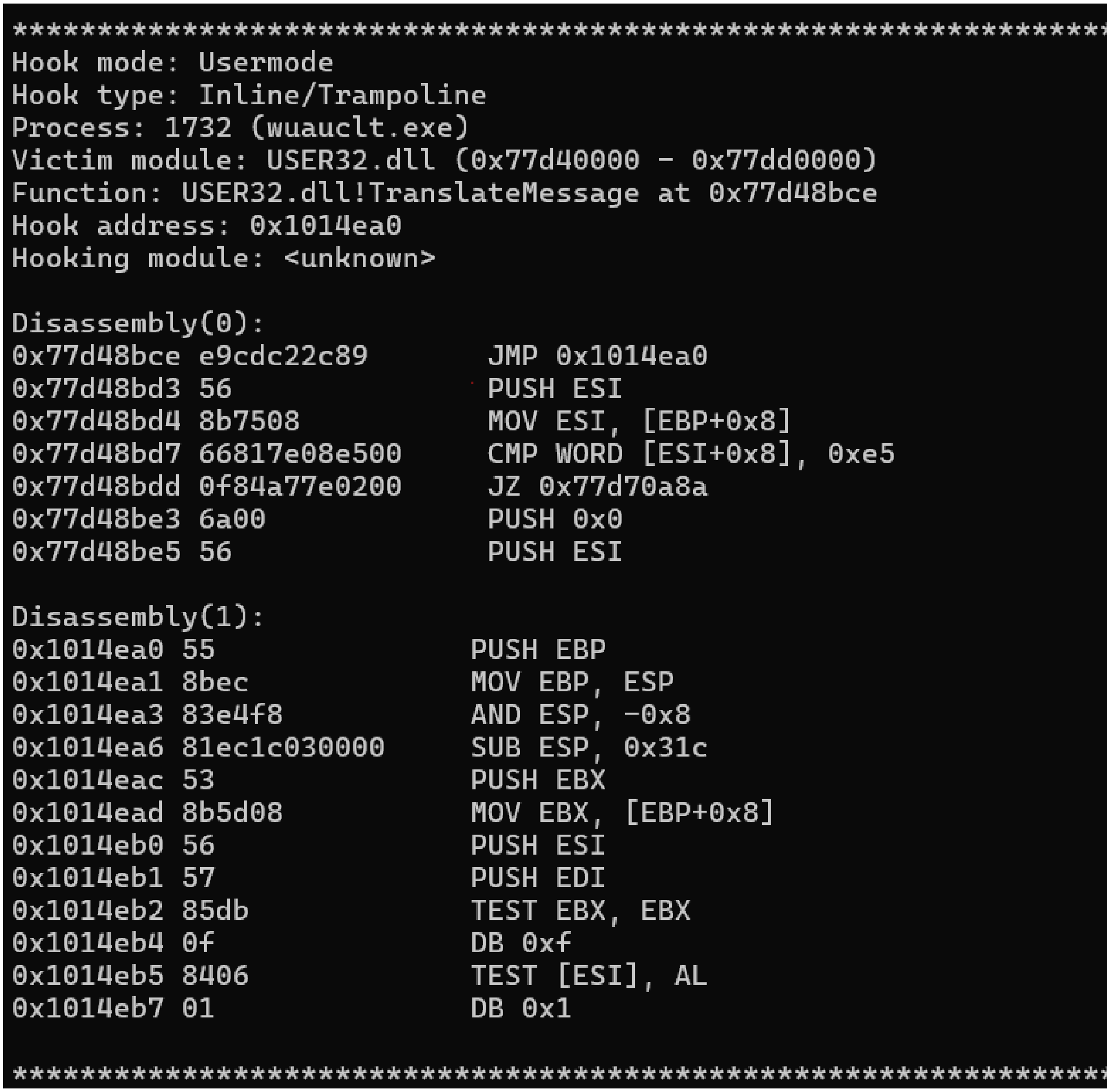
```
*******************************************************************
Hook mode: Usermode
Hook type: Inline/Trampoline
Process: 1732 (wuauclt.exe)
Victim module: ntdll.dll (0x7c900000 - 0x7c9b0000)
Function: ntdll.dll!ZwQueryDirectoryFile at 0x7c90df5e
Hook address: 0x1003ca5
Hooking module: <unknown>

Disassembly(0):
0x7c90df5e e9425d6f84       JMP 0x1003ca5
0x7c90df63 ba0003fe7f       MOV EDX, 0x7ffe0300
0x7c90df68 ff12             CALL DWORD [EDX]
0x7c90df6a c22c00           RET 0x2c
0x7c90df6d 90               NOP
0x7c90df6e 90               NOP
0x7c90df6f 90               NOP
0x7c90df70 90               NOP
0x7c90df71 90               NOP
0x7c90df72 90               NOP
0x7c90df73 b8               DB 0xb8
0x7c90df74 92               XCHG EDX, EAX
0x7c90df75 00               DB 0x0

Disassembly(1):
0x1003ca5 55               PUSH EBP
0x1003ca6 8bec             MOV EBP, ESP
0x1003ca8 e88bfeffff       CALL 0x1003b38
0x1003cad ff7530           PUSH DWORD [EBP+0x30]
0x1003cb0 ff752c           PUSH DWORD [EBP+0x2c]
0x1003cb3 ff7528           PUSH DWORD [EBP+0x28]
0x1003cb6 ff7524           PUSH DWORD [EBP+0x24]
0x1003cb9 ff7520           PUSH DWORD [EBP+0x20]
0x1003cbc ff               DB 0xff

*******************************************************************
```

GetClipboardData:



```
*****************************************************************
Hook mode: Usermode
Hook type: Inline/Trampoline
Process: 1732 (wuauclt.exe)
Victim module: USER32.dll (0x77d40000 - 0x77dd0000)
Function: USER32.dll!GetClipboardData at 0x77d6fcb2
Hook address: 0x1014fd5
Hooking module: <unknown>

Disassembly(0):
0x77d6fcb2 e91e532a89       JMP 0x1014fd5
0x77d6fcb7 83ec2c           SUB ESP, 0x2c
0x77d6fcba 56               PUSH ESI
0x77d6fcbb 57               PUSH EDI
0x77d6fcbc 8d45d4           LEA EAX, [EBP-0x2c]
0x77d6fcbf 50               PUSH EAX
0x77d6fcc0 ff7508           PUSH DWORD [EBP+0x8]
0x77d6fcc3 e8e8000000       CALL 0x77d6fdb0
0x77d6fcc8 8bf0             MOV ESI, EAX

Disassembly(1):
0x1014fd5 55               PUSH EBP
0x1014fd6 8bec             MOV EBP, ESP
0x1014fd8 53               PUSH EBX
0x1014fd9 56               PUSH ESI
0x1014fda e859ebfeff       CALL 0x1003b38
0x1014fdf 8b7508           MOV ESI, [EBP+0x8]
0x1014fe2 56               PUSH ESI
0x1014fe3 ff1560130001     CALL DWORD [0x1001360]
0x1014fe9 8bd8             MOV EBX, EAX
0x1014feb 85db             TEST EBX, EBX

*****************************************************************
```

TranslateMessage:

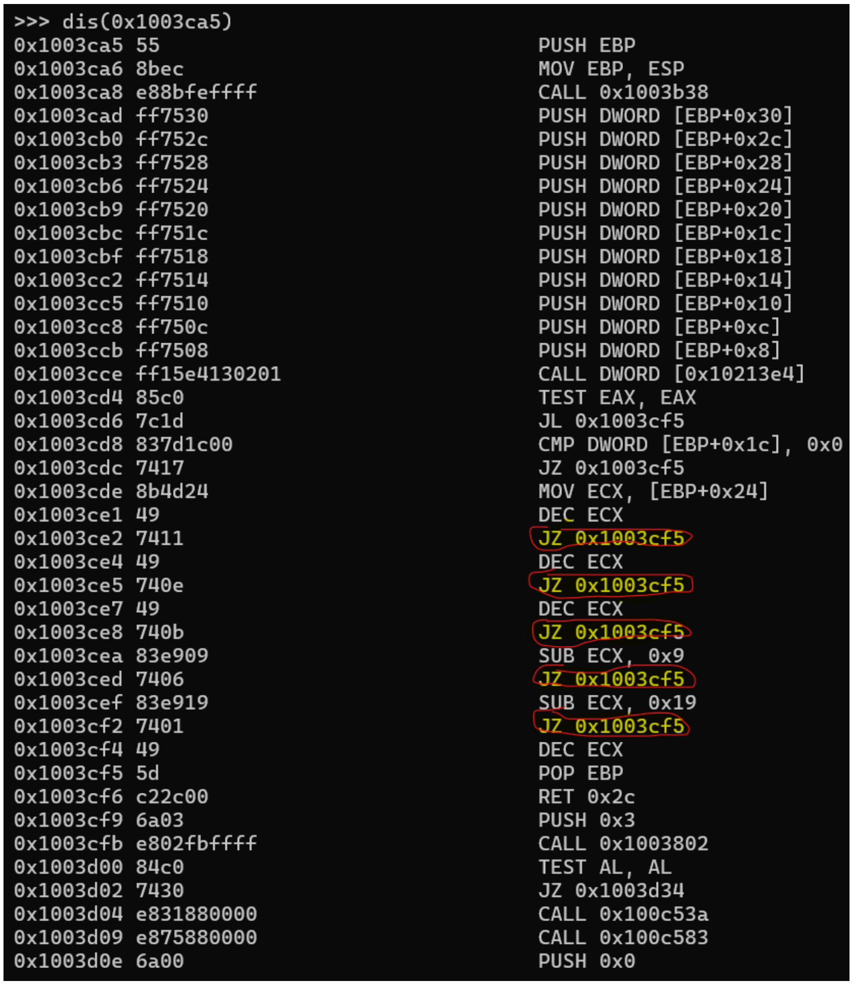


```
****************************************************************
Hook mode: Usermode
Hook type: Inline/Trampoline
Process: 1732 (wuauclt.exe)
Victim module: USER32.dll (0x77d40000 - 0x77dd0000)
Function: USER32.dll!TranslateMessage at 0x77d48bce
Hook address: 0x1014ea0
Hooking module: <unknown>

Disassembly(0):
0x77d48bce e9cdc22c89       JMP 0x1014ea0
0x77d48bd3 56               PUSH ESI
0x77d48bd4 8b7508           MOV ESI, [EBP+0x8]
0x77d48bd7 66817e08e500     CMP WORD [ESI+0x8], 0xe5
0x77d48bdd 0f84a77e0200     JZ 0x77d70a8a
0x77d48be3 6a00             PUSH 0x0
0x77d48be5 56               PUSH ESI

Disassembly(1):
0x1014ea0 55               PUSH EBP
0x1014ea1 8bec             MOV EBP, ESP
0x1014ea3 83e4f8           AND ESP, -0x8
0x1014ea6 81ec1c030000     SUB ESP, 0x31c
0x1014eac 53               PUSH EBX
0x1014ead 8b5d08           MOV EBX, [EBP+0x8]
0x1014eb0 56               PUSH ESI
0x1014eb1 57               PUSH EDI
0x1014eb2 85db             TEST EBX, EBX
0x1014eb4 0f               DB 0xf
0x1014eb5 8406             TEST [ESI], AL
0x1014eb7 01               DB 0x1

****************************************************************
```

To analyze, in the ZwQueryDirectoryFile after jumping to 0x1003ca5 a function named “0x1003b38” is called, which is also not listed. In addition, this is called on GetClipboardData in the instruction 0x1014fda.

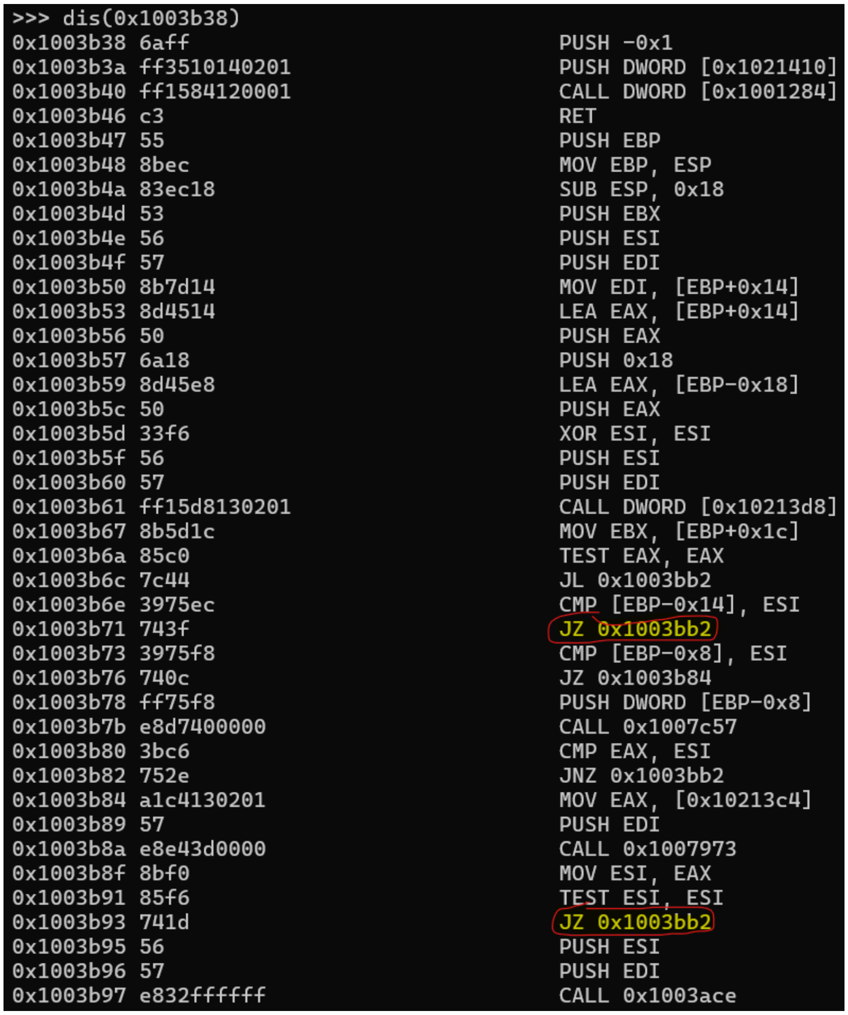
The next step was to further analyze the memory addresses that were jumped to, them being 0x1003ca5 and 0x1014fda. To accomplish this, the following commands were run in order:
- vol.py -f zeus.vmem volshell
- cc(pid=1732)
- dis(0x1003ca5)
- dis(0x1003b38)

dis(0x1003ca5):



```
>>> dis(0x1003ca5)
0x1003ca5 55                               PUSH EBP
0x1003ca6 8bec                             MOV EBP, ESP
0x1003ca8 e88bfeffff                       CALL 0x1003b38
0x1003cad ff7530                           PUSH DWORD [EBP+0x30]
0x1003cb0 ff752c                           PUSH DWORD [EBP+0x2c]
0x1003cb3 ff7528                           PUSH DWORD [EBP+0x28]
0x1003cb6 ff7524                           PUSH DWORD [EBP+0x24]
0x1003cb9 ff7520                           PUSH DWORD [EBP+0x20]
0x1003cbc ff751c                           PUSH DWORD [EBP+0x1c]
0x1003cbf ff7518                           PUSH DWORD [EBP+0x18]
0x1003cc2 ff7514                           PUSH DWORD [EBP+0x14]
0x1003cc5 ff7510                           PUSH DWORD [EBP+0x10]
0x1003cc8 ff750c                           PUSH DWORD [EBP+0xc]
0x1003ccb ff7508                           PUSH DWORD [EBP+0x8]
0x1003cce ff15e4130201                     CALL DWORD [0x10213e4]
0x1003cd4 85c0                             TEST EAX, EAX
0x1003cd6 7c1d                             JL 0x1003cf5
0x1003cd8 837d1c00                         CMP DWORD [EBP+0x1c], 0x0
0x1003cdc 7417                             JZ 0x1003cf5
0x1003cde 8b4d24                           MOV ECX, [EBP+0x24]
0x1003ce1 49                               DEC ECX
0x1003ce2 7411                             JZ 0x1003cf5
0x1003ce4 49                               DEC ECX
0x1003ce5 740e                             JZ 0x1003cf5
0x1003ce7 49                               DEC ECX
0x1003ce8 740b                             JZ 0x1003cf5
0x1003cea 83e909                           SUB ECX, 0x9
0x1003ced 7406                             JZ 0x1003cf5
0x1003cef 83e919                           SUB ECX, 0x19
0x1003cf2 7401                             JZ 0x1003cf5
0x1003cf4 49                               DEC ECX
0x1003cf5 5d                               POP EBP
0x1003cf6 c22c00                           RET 0x2c
0x1003cf9 6a03                             PUSH 0x3
0x1003cfb e802fbffff                       CALL 0x1003802
0x1003d00 84c0                             TEST AL, AL
0x1003d02 7430                             JZ 0x1003d34
0x1003d04 e831880000                       CALL 0x100c53a
0x1003d09 e875880000                       CALL 0x100c583
0x1003d0e 6a00                             PUSH 0x0
```

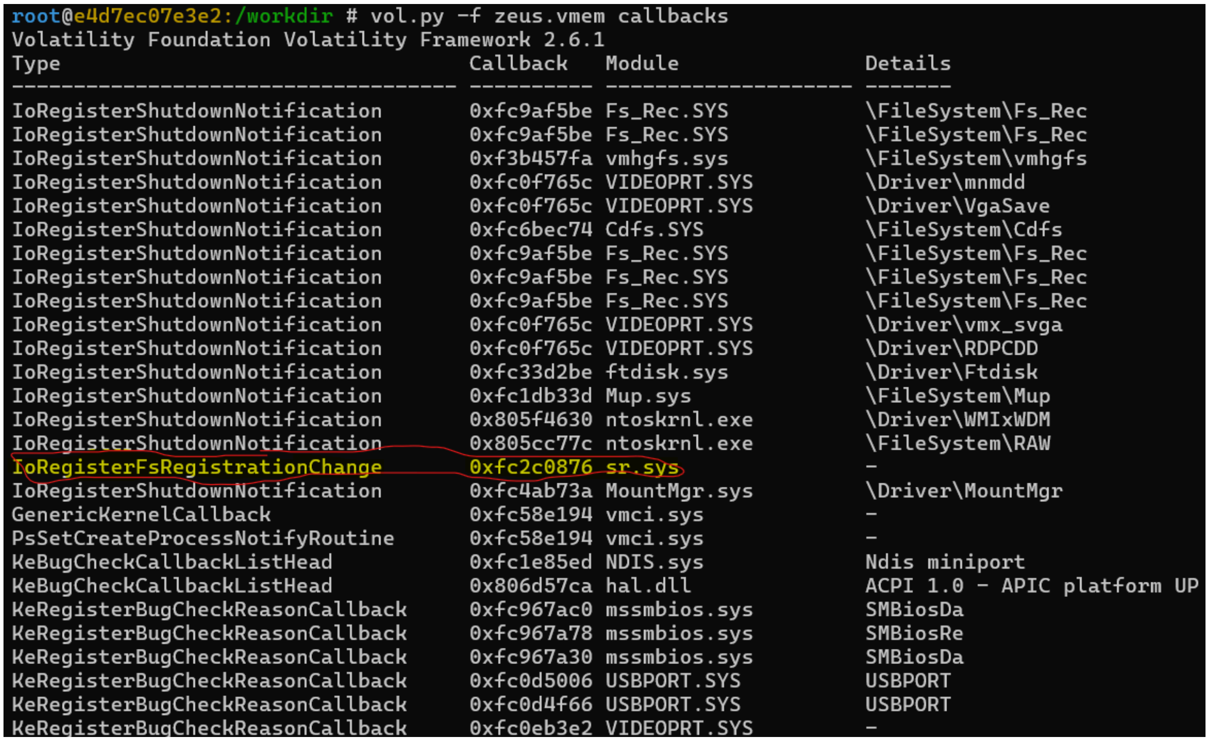
dis(0x1003b38):



```
>>> dis(0x1003b38)
0x1003b38 6aff                             PUSH -0x1
0x1003b3a ff3510140201                     PUSH DWORD [0x1021410]
0x1003b40 ff1584120001                     CALL DWORD [0x1001284]
0x1003b46 c3                               RET
0x1003b47 55                               PUSH EBP
0x1003b48 8bec                             MOV EBP, ESP
0x1003b4a 83ec18                           SUB ESP, 0x18
0x1003b4d 53                               PUSH EBX
0x1003b4e 56                               PUSH ESI
0x1003b4f 57                               PUSH EDI
0x1003b50 8b7d14                           MOV EDI, [EBP+0x14]
0x1003b53 8d4514                           LEA EAX, [EBP+0x14]
0x1003b56 50                               PUSH EAX
0x1003b57 6a18                             PUSH 0x18
0x1003b59 8d45e8                           LEA EAX, [EBP-0x18]
0x1003b5c 50                               PUSH EAX
0x1003b5d 33f6                             XOR ESI, ESI
0x1003b5f 56                               PUSH ESI
0x1003b60 57                               PUSH EDI
0x1003b61 ff15d8130201                     CALL DWORD [0x10213d8]
0x1003b67 8b5d1c                           MOV EBX, [EBP+0x1c]
0x1003b6a 85c0                             TEST EAX, EAX
0x1003b6c 7c44                             JL 0x1003bb2
0x1003b6e 3975ec                           CMP [EBP-0x14], ESI
0x1003b71 743f                             JZ 0x1003bb2
0x1003b73 3975f8                           CMP [EBP-0x8], ESI
0x1003b76 740c                             JZ 0x1003b84
0x1003b78 ff75f8                           PUSH DWORD [EBP-0x8]
0x1003b7b e8d7400000                       CALL 0x1007c57
0x1003b80 3bc6                             CMP EAX, ESI
0x1003b82 752e                             JNZ 0x1003bb2
0x1003b84 a1c4130201                       MOV EAX, [0x10213c4]
0x1003b89 57                               PUSH EDI
0x1003b8a e8e43d0000                       CALL 0x1007973
0x1003b8f 8bf0                             MOV ESI, EAX
0x1003b91 85f6                             TEST ESI, ESI
0x1003b93 741d                             JZ 0x1003bb2
0x1003b95 56                               PUSH ESI
0x1003b96 57                               PUSH EDI
0x1003b97 e832ffffff                       CALL 0x1003ace
```

Redundancies are present in the previous two screenshots, thus pointing to suspicious activity. To add, after disassembling “0x1003b38” it can see that it firsts pushes to - 0x1 and then calls 0x1001284. Furthermore, as shown JZ is a recurring instruction. To analyze, it appears to be an instruction that stands for "Jump if Zero." In conditional branching, it is usually used to change the program flow according to the value of the zero flag (ZF) in the processor's status flags. To clarify, it is used to check if the result of a previous operation or comparison yielded a zero value. If the zero flag is set, the program execution will jump to a specified target location in the code. If the zero flag is not set, the program execution will continue with the next sequential instruction. This is listed after DEC which Decrements ECX.

The next step was to run a callback function designed to handle specific events or conditions that occur within the operating system kernel or device drivers.



```
root@e4d7ec07e3e2:/workdir # vol.py -f zeus.vmem callbacks
Volatility Foundation Volatility Framework 2.6.1
Type                                 Callback   Module               Details
------------------------------------ ---------- -------------------- -------
IoRegisterShutdownNotification       0xfc9af5be Fs_Rec.SYS           \FileSystem\Fs_Rec
IoRegisterShutdownNotification       0xfc9af5be Fs_Rec.SYS           \FileSystem\Fs_Rec
IoRegisterShutdownNotification       0xf3b457fa vmhgfs.sys           \FileSystem\vmhgfs
IoRegisterShutdownNotification       0xfc0f765c VIDEOPRT.SYS         \Driver\mnmdd
IoRegisterShutdownNotification       0xfc0f765c VIDEOPRT.SYS         \Driver\VgaSave
IoRegisterShutdownNotification       0xfc6bec74 Cdfs.SYS             \FileSystem\Cdfs
IoRegisterShutdownNotification       0xfc9af5be Fs_Rec.SYS           \FileSystem\Fs_Rec
IoRegisterShutdownNotification       0xfc9af5be Fs_Rec.SYS           \FileSystem\Fs_Rec
IoRegisterShutdownNotification       0xfc9af5be Fs_Rec.SYS           \FileSystem\Fs_Rec
IoRegisterShutdownNotification       0xfc0f765c VIDEOPRT.SYS         \Driver\vmx_svga
IoRegisterShutdownNotification       0xfc0f765c VIDEOPRT.SYS         \Driver\RDPCDD
IoRegisterShutdownNotification       0xfc33d2be ftdisk.sys           \Driver\Ftdisk
IoRegisterShutdownNotification       0xfc1db33d Mup.sys              \FileSystem\Mup
IoRegisterShutdownNotification       0x805f4630 ntoskrnl.exe         \Driver\WMIxWDM
IoRegisterShutdownNotification       0x805cc77c ntoskrnl.exe         \FileSystem\RAW
IoRegisterFsRegistrationChange       0xfc2c0876 sr.sys               -
IoRegisterShutdownNotification       0xfc4ab73a MountMgr.sys         \Driver\MountMgr
GenericKernelCallback                0xfc58e194 vmci.sys             -
PsSetCreateProcessNotifyRoutine      0xfc58e194 vmci.sys             -
KeBugCheckCallbackListHead           0xfc1e85ed NDIS.sys             Ndis miniport
KeBugCheckCallbackListHead           0x806d57ca hal.dll              ACPI 1.0 - APIC platform UP
KeRegisterBugCheckReasonCallback     0xfc967ac0 mssmbios.sys         SMBiosDa
KeRegisterBugCheckReasonCallback     0xfc967a78 mssmbios.sys         SMBiosRe
KeRegisterBugCheckReasonCallback     0xfc967a30 mssmbios.sys         SMBiosDa
KeRegisterBugCheckReasonCallback     0xfc0d5006 USBPORT.SYS          USBPORT
KeRegisterBugCheckReasonCallback     0xfc0d4f66 USBPORT.SYS          USBPORT
KeRegisterBugCheckReasonCallback     0xfc0eb3e2 VIDEOPRT.SYS         -
```

As highlighted above there is a module called “sr.sys” which registers if there is a change, injects, and sends a notification. As a result, it can be inferred that it can copy, hide, or spread sensitive information.

3. Discussion and Conclusion:

Generated by ChatGPT- In conclusion, the project aimed to perform a comprehensive analysis of the notorious Zeus Trojan malware. The investigation explored memory forensics using the robust Volatility framework and looking through the supplied zeus.vmem memory dump file. Determining the malware's malicious activity and understanding its inner workings were the objectives. An extensive examination of the memory dump file provided important information about the virus's characteristics. Important features and code segments that support the malware's malicious activities were found during the investigation. Knowing Zeus's subtleties helps us to lessen possible dangers and better withstand its effects. It is important to note that a careful examination showed the goal of the virus was likely theft of private data such as personal identifiable information (PII), financial information, passwords/login credentials, etc.