FRANK BOLDEWIN'S WWW.RECONSTRUCTER.ORG

```
push     2
call     sub_672B3730
add      esp, 0Ch
test     eax, eax
jnz      short loc_672B5428
lea      edx, [esp+110h+LibFileName]
push     edx
call     sub_672B35F0
mov      edi, off_672CA058
or       ecx, 0FFFFFFFFh
xor      eax, eax
lea      edx, [esp+114h+LibFileName]
repne scasb
not      ecx
sub      edi, ecx
mov      esi, edi
mov      ebx, ecx
cmp      eax, 7Eh
jnz      loc_672B5455
lea      ecx, [esp+110h+LibFileName]
push     104h
push     ecx
push     2
call     sub_672B3730
add      esp, 0Ch
test     eax, eax
jnz      short loc_672B5428
lea      edx, [esp+110h+LibFi
push     edx
call     sub_672B35F0
mov      edi, off_672CA058
or       ecx, 0FFFFFFFFh
xor      eax, eax
lea      edx, [esp+114h+LibFi
repne scasb
not      ecx
sub      edi, ecx
mov      esi, edi
mov      ebx, ecx
```

Hunting malware with Volatility v2.0

Frank Boldewin

CAST Forum

December 2011

(English edition)

What is Volatility?

- Forensics framework to acquire digital artifacts from memory dumps
- Completely written in Python
- Current stable version is 2.0.1
- Easy to use plugin interface
- Supports the following x86 Windows versions
  - Windows XP SP 2, 3
  - Windows 2003 Server SP 0, 1, 2
  - Windows Vista SP 0, 1, 2
  - Windows 2008 Server SP 1, 2
  - Windows 7 SP 0, 1

How does Volatility work?

- Volatility versions <=1.3 only supported Windows XP and searched for hardcoded values, e.g. to detect the Kernel Processor Control Region (KPCR)
- Starting with version 2.0 advanced scanning techniques are being used to detect the KPCR

```
If KPCR.baseaddr == *(baseaddr +10)
    Then Start_Sanity_checks()
or _DBGKD_DEBUG_DATA_HEADER64 Scan
```

- For details on these scanning techniques read the following articles

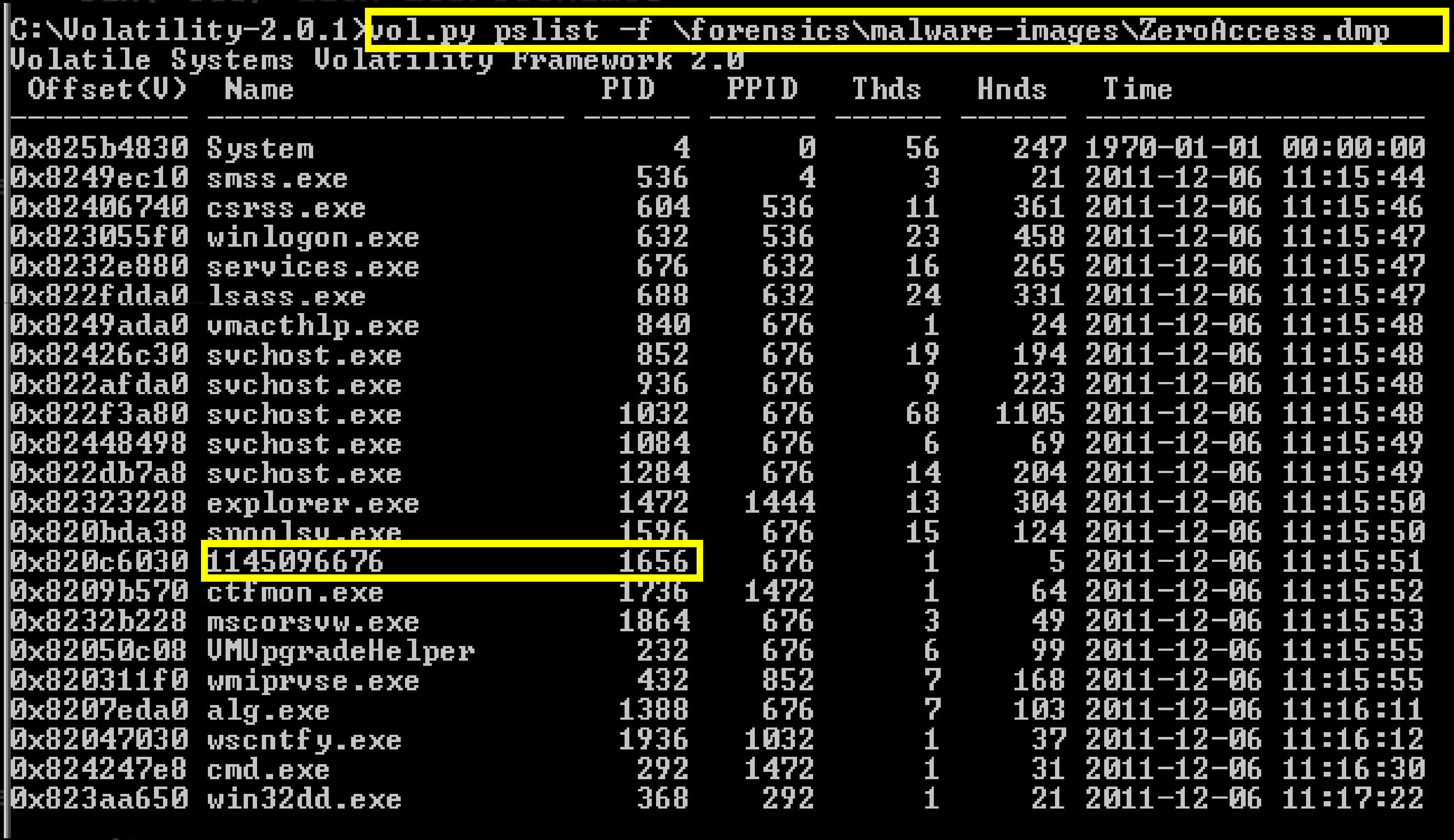
http://blog.schatzforensic.com.au/2010/07/finding-object-roots-in-vista-kpcr/

http://gleeda.blogspot.com/2010/12/identifying-memory-images.html

How does Volatility work?

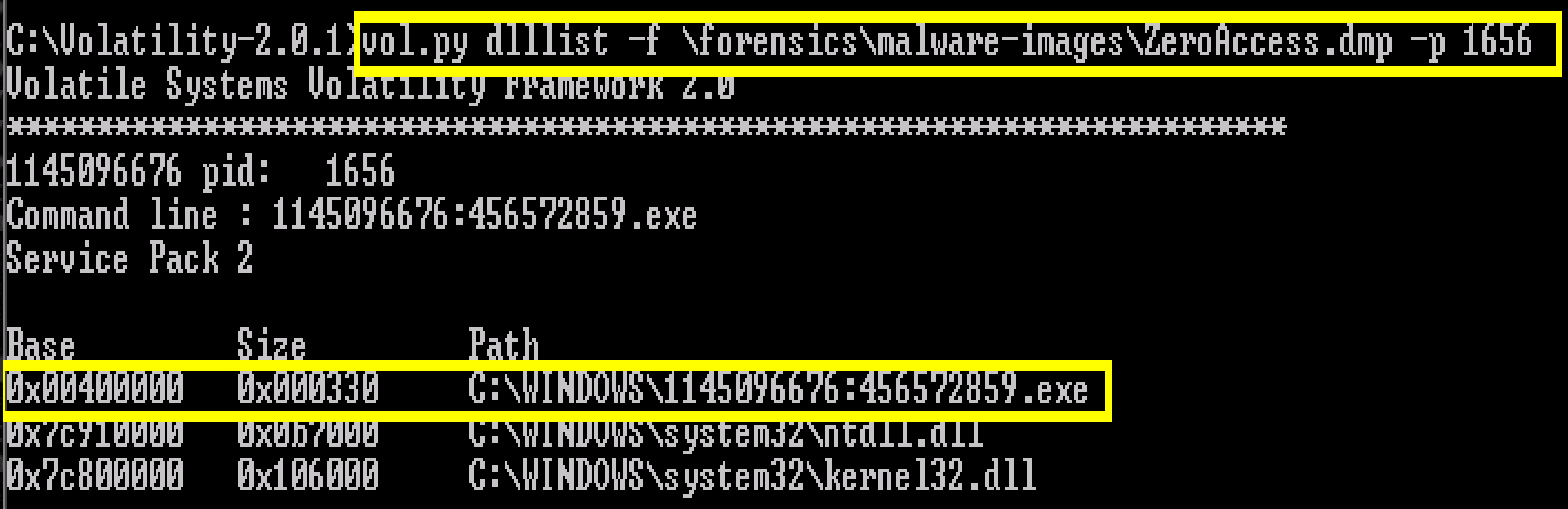
- After detecting the right Windows version and its KPCR, volatility scans for dozens of other structures inside a dump file. Additional plugins like malware.py hunt for malicious activities by using strong heuristics or comparing results from different structures
- Typical structures being parsed are:
  - _EPROCESS und _KPROCESS
  - _KTIMER
  - _TCPT_OBJECT
  - _ETHREAD und _KTHREAD
  - _CMHIVE
  - _LDR_DATA_TABLE_ENTRY
  - _KMUTANT .....

Show active processes via _EPROCESS list parsing



```
C:\Volatility-2.0.1>vol.py pslist -f \forensics\malware-images\ZeroAccess.dmp
Volatile Systems Volatility Framework 2.0
 Offset(V)  Name                    PID    PPID   Thds   Hnds   Time
---------- -------------------- ------ ------ ------ ------ -------------------
0x825b4830 System                    4      0     56    247 1970-01-01 00:00:00
0x8249ec10 smss.exe                536      4      3     21 2011-12-06 11:15:44
0x82406740 csrss.exe               604    536     11    361 2011-12-06 11:15:46
0x823055f0 winlogon.exe            632    536     23    458 2011-12-06 11:15:47
0x8232e880 services.exe            676    632     16    265 2011-12-06 11:15:47
0x822fdda0 lsass.exe               688    632     24    331 2011-12-06 11:15:47
0x8249ada0 vmacthlp.exe            840    676      1     24 2011-12-06 11:15:48
0x82426c30 svchost.exe             852    676     19    194 2011-12-06 11:15:48
0x822afda0 svchost.exe             936    676      9    223 2011-12-06 11:15:48
0x822f3a80 svchost.exe            1032    676     68   1105 2011-12-06 11:15:48
0x82448498 svchost.exe            1084    676      6     69 2011-12-06 11:15:49
0x822db7a8 svchost.exe            1284    676     14    204 2011-12-06 11:15:49
0x82323228 explorer.exe           1472   1444     13    304 2011-12-06 11:15:50
0x820bda38 spoolsv.exe            1596    676     15    124 2011-12-06 11:15:50
0x820c6030 1145096676             1656    676      1      5 2011-12-06 11:15:51
0x8209b570 ctfmon.exe             1736   1472      1     64 2011-12-06 11:15:52
0x8232b228 mscorsvw.exe           1864    676      3     49 2011-12-06 11:15:53
0x82050c08 VMUpgradeHelper         232    676      6     99 2011-12-06 11:15:55
0x820311f0 wmiprvse.exe            432    852      7    168 2011-12-06 11:15:55
0x8207eda0 alg.exe                1388    676      7    103 2011-12-06 11:16:11
0x82047030 wscntfy.exe            1936   1032      1     37 2011-12-06 11:16:12
0x824247e8 cmd.exe                 292   1472      1     31 2011-12-06 11:16:30
0x823aa650 win32dd.exe             368    292      1     21 2011-12-06 11:17:22
```

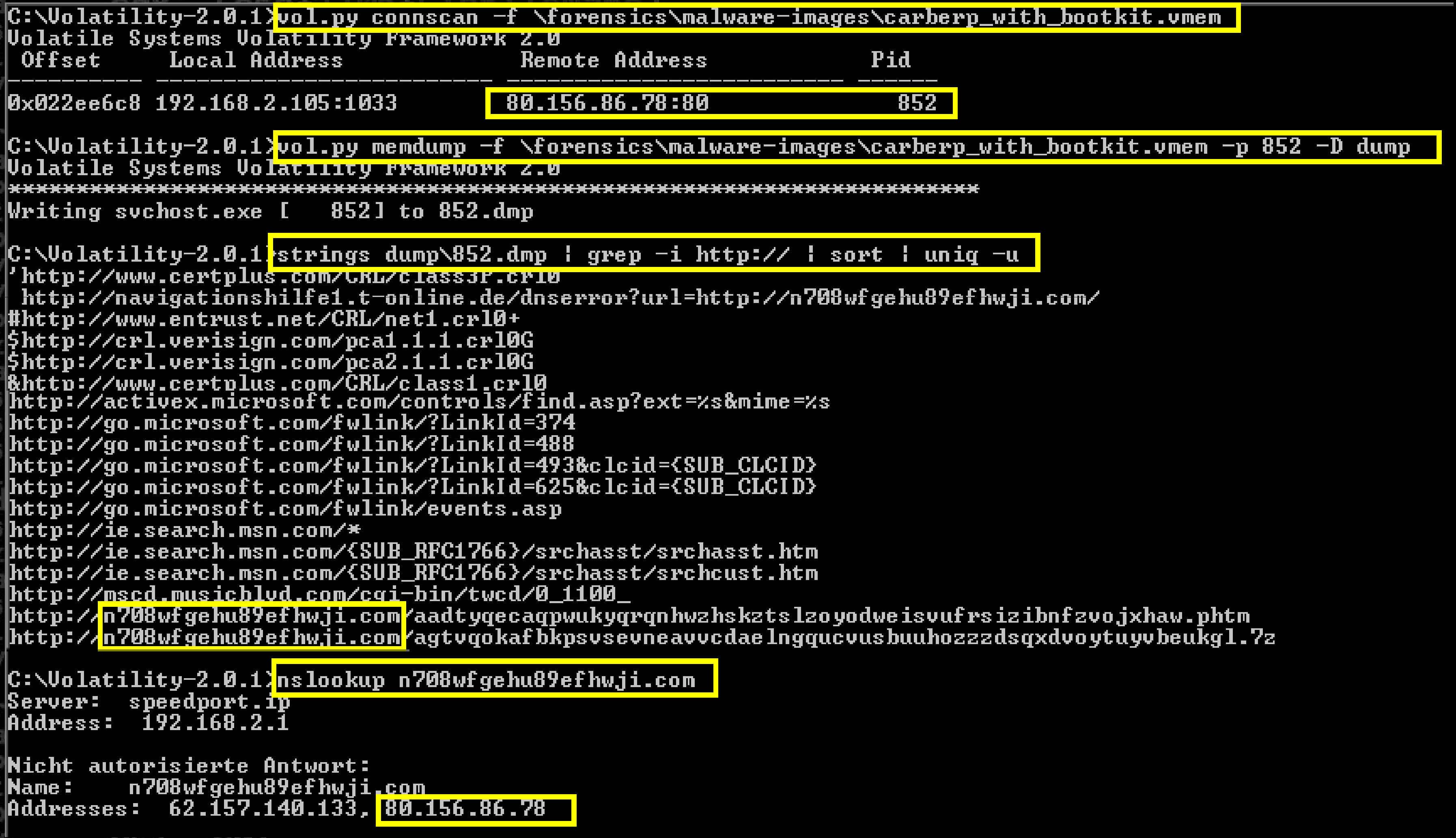
Show running modules/libraries to processes via Process Environment Block parsing



```
C:\Volatility-2.0.1>vol.py dlllist -f \forensics\malware-images\ZeroAccess.dmp -p 1656
Volatile Systems Volatility Framework 2.0
************************************************************************
1145096676 pid:   1656
Command line : 1145096676:456572859.exe
Service Pack 2

Base         Size         Path
0x00400000   0x000330     C:\WINDOWS\1145096676:456572859.exe
0x7c910000   0x0b7000     C:\WINDOWS\system32\ntdll.dll
0x7c800000   0x106000     C:\WINDOWS\system32\kernel32.dll
```

FRANK BOLDEWIN'S WWW.RECONSTRUCTER.ORG

Hunting for the C&C server with the connscan feature via _TCPT_OBJECT parsing



```
C:\Volatility-2.0.1>vol.py connscan -f \forensics\malware-images\carberp_with_bootkit.vmem
Volatile Systems Volatility Framework 2.0
 Offset     Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ------
0x022ee6c8 192.168.2.105:1033         80.156.86.78:80              852

C:\Volatility-2.0.1>vol.py memdump -f \forensics\malware-images\carberp_with_bootkit.vmem -p 852 -D dump
Volatile Systems Volatility Framework 2.0
************************************************************************
Writing svchost.exe [   852] to 852.dmp

C:\Volatility-2.0.1>strings dump\852.dmp | grep -i http:// | sort | uniq -u
'http://www.certplus.com/CRL/class3P.crl0
 http://navigationshilfe1.t-online.de/dnserror?url=http://n708wfgehu89efhwji.com/
#http://www.entrust.net/CRL/net1.crl0+
$http://crl.verisign.com/pca1.1.1.crl0G
$http://crl.verisign.com/pca2.1.1.crl0G
&http://www.certplus.com/CRL/class1.crl0
http://activex.microsoft.com/controls/find.asp?ext=%s&mime=%s
http://go.microsoft.com/fwlink/?LinkId=374
http://go.microsoft.com/fwlink/?LinkId=488
http://go.microsoft.com/fwlink/?LinkId=493&clcid={SUB_CLCID}
http://go.microsoft.com/fwlink/?LinkId=625&clcid={SUB_CLCID}
http://go.microsoft.com/fwlink/events.asp
http://ie.search.msn.com/*
http://ie.search.msn.com/{SUB_RFC1766}/srchasst/srchasst.htm
http://ie.search.msn.com/{SUB_RFC1766}/srchasst/srchcust.htm
http://mscd.musicblvd.com/cgi-bin/twcd/0_1100_
http://n708wfgehu89efhwji.com/aadtyqecaqpwukyqrqnhwzhskztslzoyodweisvufrsizibnfzvojxhaw.phtm
http://n708wfgehu89efhwji.com/agtvqokafbkpsvsevneavvcdaelngqucvusbuuhozzzdsqxdvoytuyvbeukgl.7z

C:\Volatility-2.0.1>nslookup n708wfgehu89efhwji.com
Server:  speedport.ip
Address:  192.168.2.1

Nicht autorisierte Antwort:
Name:    n708wfgehu89efhwji.com
Addresses:  62.157.140.133, 80.156.86.78
```

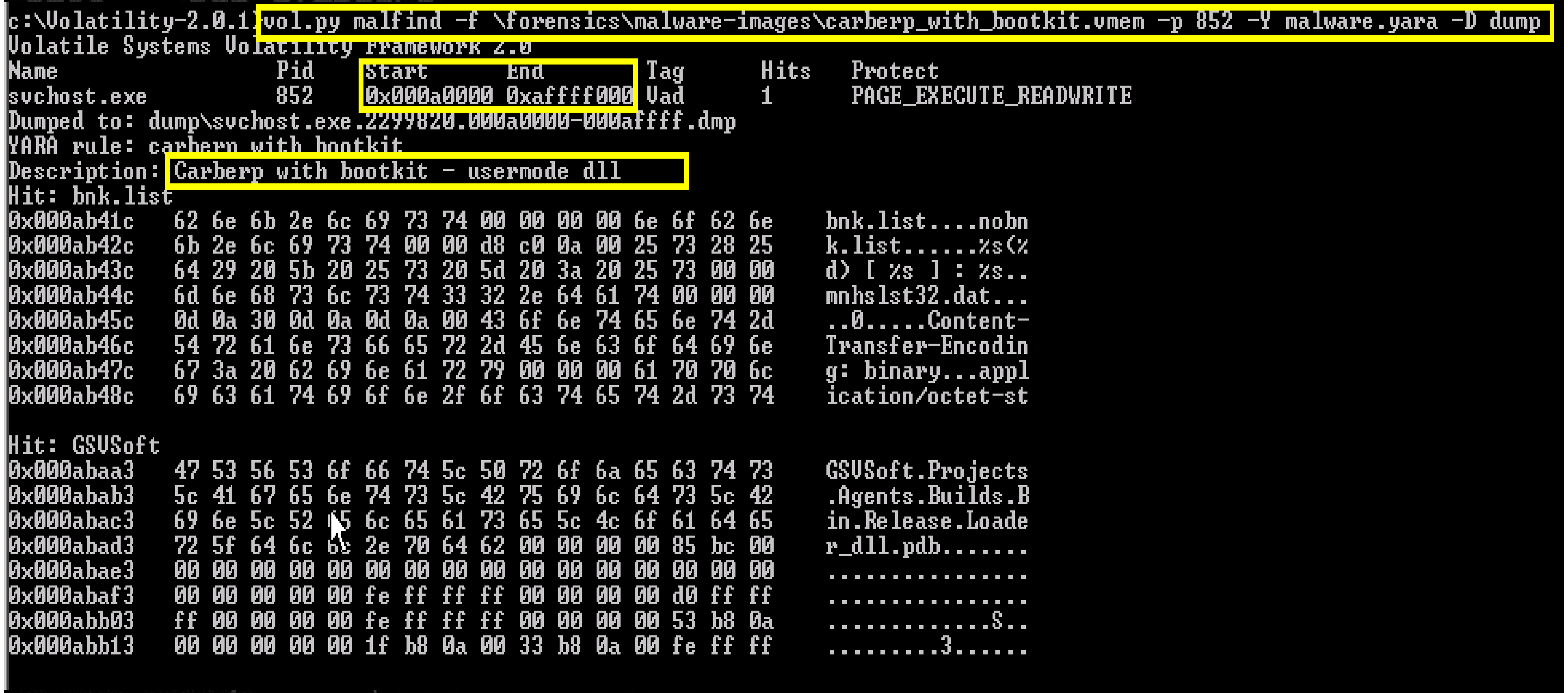
Virtual Address Descriptor (VAD)

- The VAD is a kernel data structure that describes the allocated memory pages of a process, e.g. loaded modules, mapped files or private heap
- A very often used malware technique is to inject its malicious code into trusted/privileged processes, e.g. Services.exe, Svchost.exe, Winlogon.exe

VAD parsing to find injected code with “malfind”

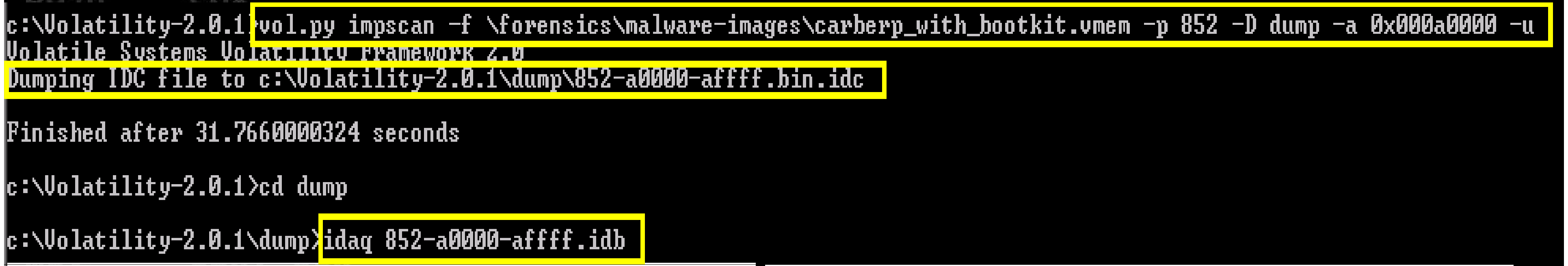
- Regular loaded libraries in the address space of a process are of type _MMVAD or _MMVAD_LONG
- Dynamically allocated memory pages created via VirtualAllocEx/WriteProcessMemory are of type _MMVAD_SHORT
- If these memory pages additionally are marked as PAGE_EXECUTE_READWRITE, this is a good indication for the malfind feature to write this page to a dump directory
- With the YARA library in combination further malware indicators could be detected

Hunting for injected code inside trusted/privileged processes and scan for typical malware pattern with YARA

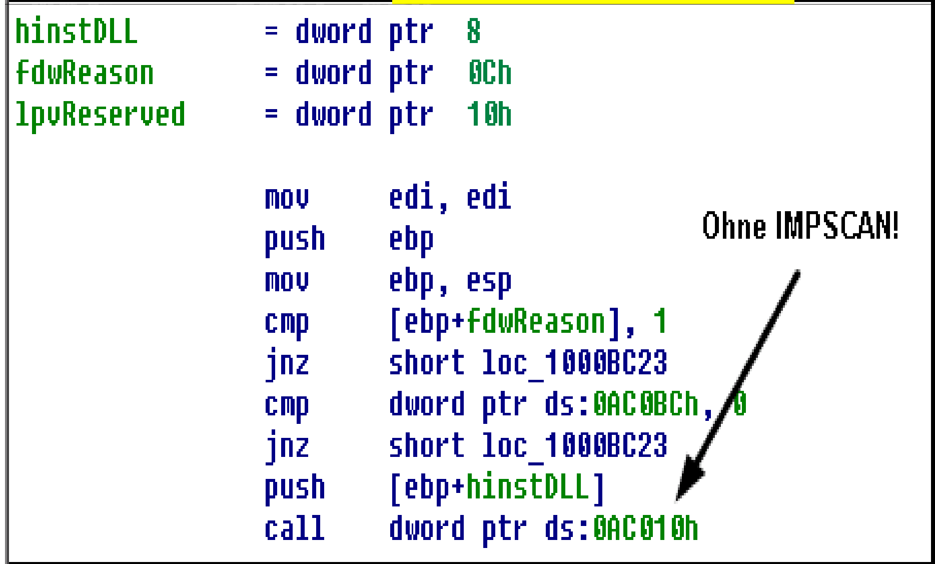


```
c:\Volatility-2.0.1>vol.py malfind -f \forensics\malware-images\carberp_with_bootkit.vmem -p 852 -Y malware.yara -D dump
Volatile Systems Volatility Framework 2.0
Name                 Pid    Start      End        Tag      Hits    Protect
svchost.exe          852    0x000a0000 0xaffff000 Vad      1       PAGE_EXECUTE_READWRITE
Dumped to: dump\svchost.exe.2299820.000a0000-000affff.dmp
YARA rule: carberp_with_bootkit
Description: Carberp with bootkit - usermode dll
Hit: bnk.list
0x000ab41c   62 6e 6b 2e 6c 69 73 74 00 00 00 00 6e 6f 62 6e    bnk.list....nobn
0x000ab42c   6b 2e 6c 69 73 74 00 00 d8 c0 0a 00 25 73 28 25    k.list......%s(%
0x000ab43c   64 29 20 5b 20 25 73 20 5d 20 3a 20 25 73 00 00    d) [ %s ] : %s..
0x000ab44c   6d 6e 68 73 6c 73 74 33 32 2e 64 61 74 00 00 00    mnhslst32.dat...
0x000ab45c   0d 0a 30 0d 0a 0d 0a 00 43 6f 6e 74 65 6e 74 2d    ..0.....Content-
0x000ab46c   54 72 61 6e 73 66 65 72 2d 45 6e 63 6f 64 69 6e    Transfer-Encodin
0x000ab47c   67 3a 20 62 69 6e 61 72 79 00 00 00 61 70 70 6c    g: binary...appl
0x000ab48c   69 63 61 74 69 6f 6e 2f 6f 63 74 65 74 2d 73 74    ication/octet-st

Hit: GSVSoft
0x000abaa3   47 53 56 53 6f 66 74 5c 50 72 6f 6a 65 63 74 73    GSVSoft.Projects
0x000abab3   5c 41 67 65 6e 74 73 5c 42 75 69 6c 64 73 5c 42    .Agents.Builds.B
0x000abac3   69 6e 5c 52 65 6c 65 61 73 65 5c 4c 6f 61 64 65    in.Release.Loade
0x000abad3   72 5f 64 6c 6c 2e 70 64 62 00 00 00 00 85 bc 00    r_dll.pdb.......
0x000abae3   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0x000abaf3   00 00 00 00 00 fe ff ff ff 00 00 00 00 d0 ff ff    ................
0x000abb03   ff 00 00 00 00 fe ff ff ff 00 00 00 00 53 b8 0a    .............S..
0x000abb13   00 00 00 00 00 1f b8 0a 00 33 b8 0a 00 fe ff ff    .........3......
```

PE-File fixing via impscan to have clean importnames inside IDA Pro

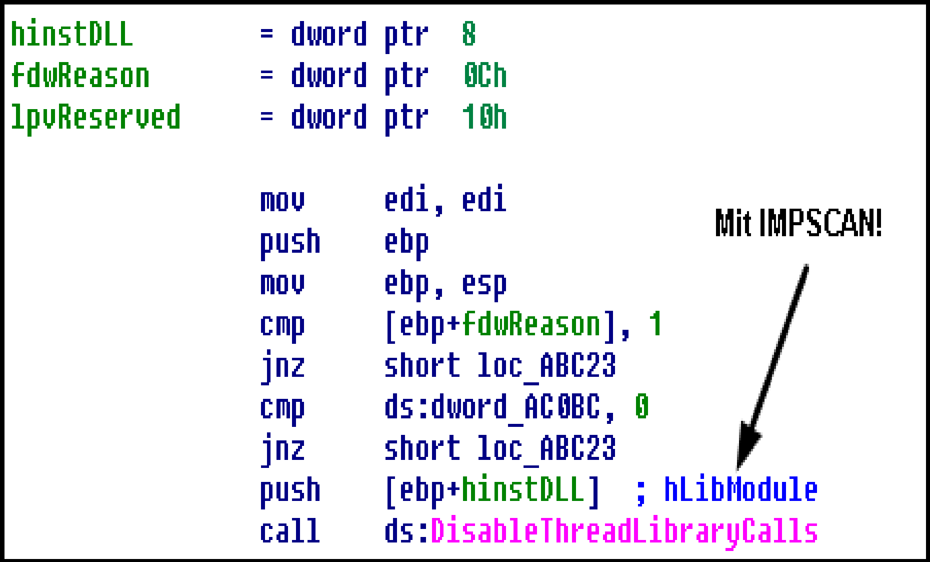


```
c:\Volatility-2.0.1>vol.py impscan -f \forensics\malware-images\carberp_with_bootkit.vmem -p 852 -D dump -a 0x000a0000 -u
Volatile Systems Volatility Framework 2.0
Dumping IDC file to c:\Volatility-2.0.1\dump\852-a0000-affff.bin.idc

Finished after 31.7660000324 seconds

c:\Volatility-2.0.1>cd dump

c:\Volatility-2.0.1\dump>idaq 852-a0000-affff.idb
```



```
hinstDLL        = dword ptr  8
fdwReason       = dword ptr  0Ch
lpvReserved     = dword ptr  10h

                mov     edi, edi
                push    ebp
                mov     ebp, esp
                cmp     [ebp+fdwReason], 1
                jnz     short loc_1000BC23
                cmp     dword ptr ds:0AC0BCh, 0
                jnz     short loc_1000BC23
                push    [ebp+hinstDLL]
                call    dword ptr ds:0AC010h
```

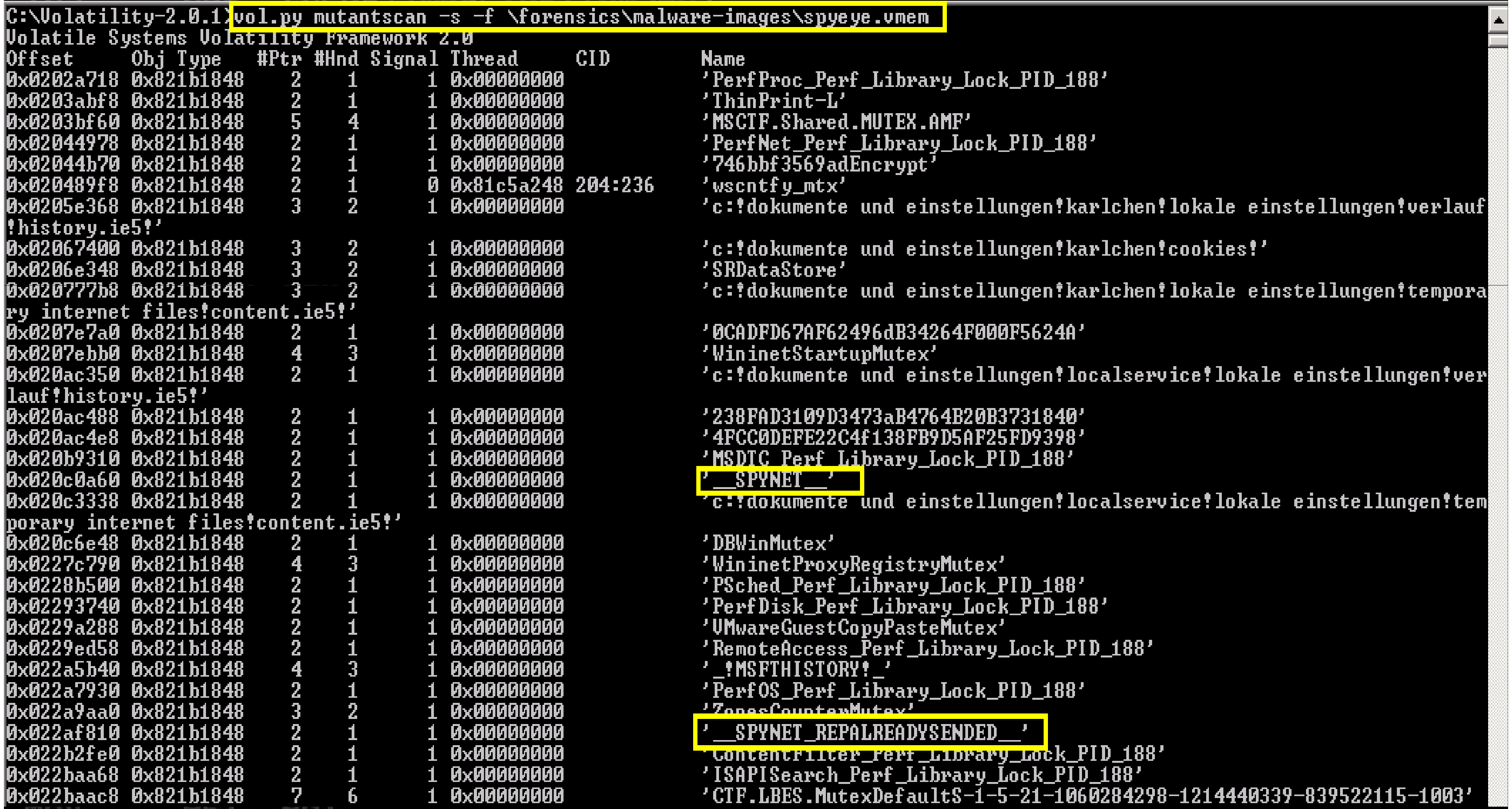
Ohne IMPSCAN!



```
hinstDLL        = dword ptr  8
fdwReason       = dword ptr  0Ch
lpvReserved     = dword ptr  10h

                mov     edi, edi
                push    ebp
                mov     ebp, esp
                cmp     [ebp+fdwReason], 1
                jnz     short loc_ABC23
                cmp     ds:dword_AC0BC, 0
                jnz     short loc_ABC23
                push    [ebp+hinstDLL]  ; hLibModule
                call    ds:DisableThreadLibraryCalls
```

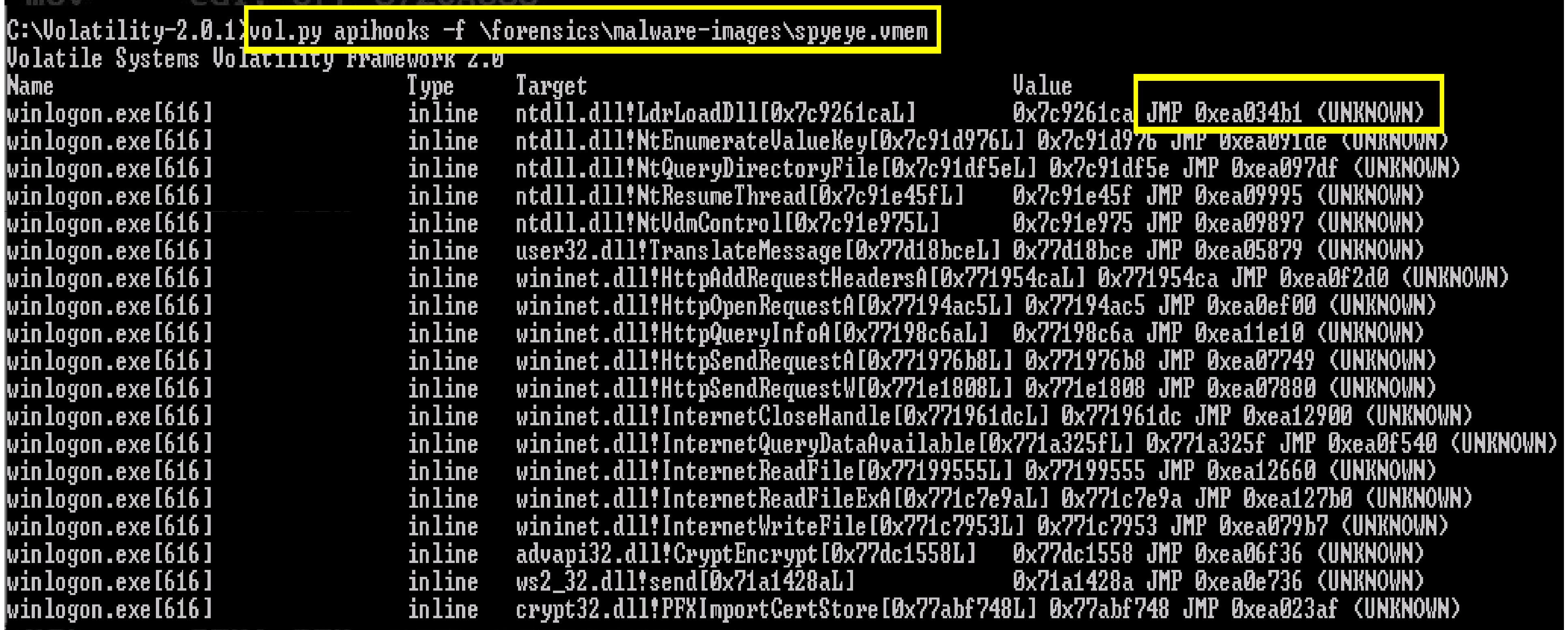
Mit IMPSCAN!

View of named mutexes to identify typical malware pattern



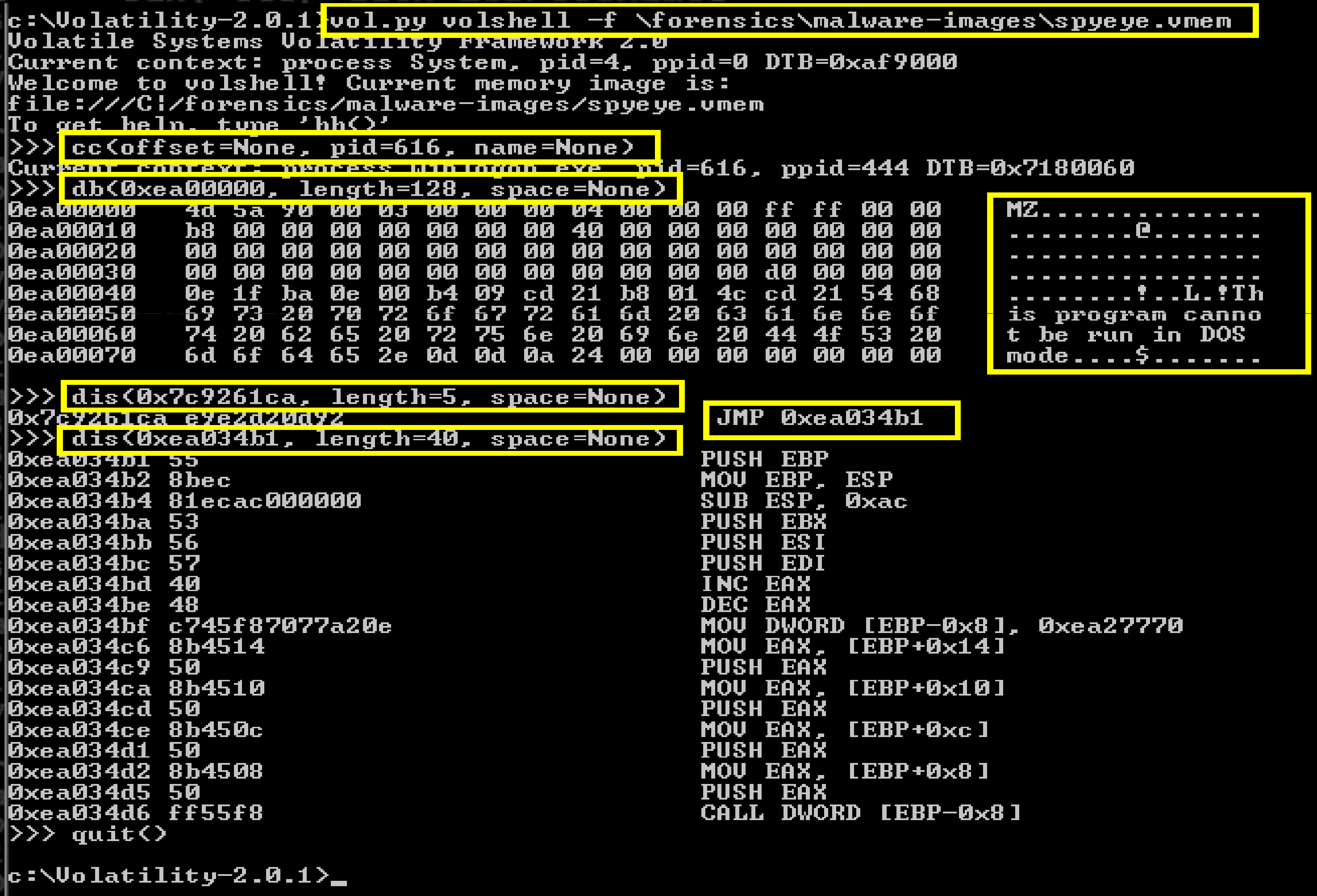
```
C:\Volatility-2.0.1>vol.py mutantscan -s -f \forensics\malware-images\spyeye.vmem
Volatile Systems Volatility Framework 2.0
Offset     Obj Type   #Ptr #Hnd Signal Thread     CID        Name
0x0202a718 0x821b1848    2    1      1 0x00000000            'PerfProc_Perf_Library_Lock_PID_188'
0x0203abf8 0x821b1848    2    1      1 0x00000000            'ThinPrint-L'
0x0203bf60 0x821b1848    5    4      1 0x00000000            'MSCTF.Shared.MUTEX.AMF'
0x02044978 0x821b1848    2    1      1 0x00000000            'PerfNet_Perf_Library_Lock_PID_188'
0x02044b70 0x821b1848    2    1      1 0x00000000            '746bbf3569adEncrypt'
0x020489f8 0x821b1848    2    1      0 0x81c5a248 204:236    'wscntfy_mtx'
0x0205e368 0x821b1848    3    2      1 0x00000000            'c:!dokumente und einstellungen!karlchen!lokale einstellungen!verlauf
!history.ie5!'
0x02067400 0x821b1848    3    2      1 0x00000000            'c:!dokumente und einstellungen!karlchen!cookies!'
0x0206e348 0x821b1848    3    2      1 0x00000000            'SRDataStore'
0x020777b8 0x821b1848    3    2      1 0x00000000            'c:!dokumente und einstellungen!karlchen!lokale einstellungen!tempora
ry internet files!content.ie5!'
0x0207e7a0 0x821b1848    2    1      1 0x00000000            '0CADFD67AF62496dB34264F000F5624A'
0x0207ebb0 0x821b1848    4    3      1 0x00000000            'WininetStartupMutex'
0x020ac350 0x821b1848    2    1      1 0x00000000            'c:!dokumente und einstellungen!localservice!lokale einstellungen!ver
lauf!history.ie5!'
0x020ac488 0x821b1848    2    1      1 0x00000000            '238FAD3109D3473aB4764B20B3731840'
0x020ac4e8 0x821b1848    2    1      1 0x00000000            '4FCC0DEFE22C4f138FB9D5AF25FD9398'
0x020b9310 0x821b1848    2    1      1 0x00000000            'MSDTC_Perf_Library_Lock_PID_188'
0x020c0a60 0x821b1848    2    1      1 0x00000000            '__SPYNET__'
0x020c3338 0x821b1848    2    1      1 0x00000000            'c:!dokumente und einstellungen!localservice!lokale einstellungen!tem
porary internet files!content.ie5!'
0x020c6e48 0x821b1848    2    1      1 0x00000000            'DBWinMutex'
0x0227c790 0x821b1848    4    3      1 0x00000000            'WininetProxyRegistryMutex'
0x0228b500 0x821b1848    2    1      1 0x00000000            'PSched_Perf_Library_Lock_PID_188'
0x02293740 0x821b1848    2    1      1 0x00000000            'PerfDisk_Perf_Library_Lock_PID_188'
0x0229a288 0x821b1848    2    1      1 0x00000000            'VMwareGuestCopyPasteMutex'
0x0229ed58 0x821b1848    2    1      1 0x00000000            'RemoteAccess_Perf_Library_Lock_PID_188'
0x022a5b40 0x821b1848    4    3      1 0x00000000            '_!MSFTHISTORY!_'
0x022a7930 0x821b1848    2    1      1 0x00000000            'PerfOS_Perf_Library_Lock_PID_188'
0x022a9aa0 0x821b1848    3    2      1 0x00000000            'ZonesCounterMutex'
0x022af810 0x821b1848    2    1      1 0x00000000            '__SPYNET_REPALREADYSENDED__'
0x022b2fe0 0x821b1848    2    1      1 0x00000000            'ContentFilter_Perf_Library_Lock_PID_188'
0x022baa68 0x821b1848    2    1      1 0x00000000            'ISAPISearch_Perf_Library_Lock_PID_188'
0x022baac8 0x821b1848    7    6      1 0x00000000            'CTF.LBES.MutexDefaultS-1-5-21-1060284298-1214440339-839522115-1003'
```

Hunting for code hooks to detect manipulated system functions



```
C:\Volatility-2.0.1>vol.py apihooks -f \forensics\malware-images\spyeye.vmem
Volatile Systems Volatility Framework 2.0
Name                            Type     Target                                    Value
winlogon.exe[616]               inline   ntdll.dll!LdrLoadDll[0x7c9261caL]         0x7c9261ca JMP 0xea034b1 (UNKNOWN)
winlogon.exe[616]               inline   ntdll.dll!NtEnumerateValueKey[0x7c91d976L] 0x7c91d976 JMP 0xea091de (UNKNOWN)
winlogon.exe[616]               inline   ntdll.dll!NtQueryDirectoryFile[0x7c91df5eL] 0x7c91df5e JMP 0xea097df (UNKNOWN)
winlogon.exe[616]               inline   ntdll.dll!NtResumeThread[0x7c91e45fL]     0x7c91e45f JMP 0xea09995 (UNKNOWN)
winlogon.exe[616]               inline   ntdll.dll!NtVdmControl[0x7c91e975L]       0x7c91e975 JMP 0xea09897 (UNKNOWN)
winlogon.exe[616]               inline   user32.dll!TranslateMessage[0x77d18bceL] 0x77d18bce JMP 0xea05879 (UNKNOWN)
winlogon.exe[616]               inline   wininet.dll!HttpAddRequestHeadersA[0x771954caL] 0x771954ca JMP 0xea0f2d0 (UNKNOWN)
winlogon.exe[616]               inline   wininet.dll!HttpOpenRequestA[0x77194ac5L] 0x77194ac5 JMP 0xea0ef00 (UNKNOWN)
winlogon.exe[616]               inline   wininet.dll!HttpQueryInfoA[0x77198c6aL]  0x77198c6a JMP 0xea11e10 (UNKNOWN)
winlogon.exe[616]               inline   wininet.dll!HttpSendRequestA[0x771976b8L] 0x771976b8 JMP 0xea07749 (UNKNOWN)
winlogon.exe[616]               inline   wininet.dll!HttpSendRequestW[0x771e1808L] 0x771e1808 JMP 0xea07880 (UNKNOWN)
winlogon.exe[616]               inline   wininet.dll!InternetCloseHandle[0x771961dcL] 0x771961dc JMP 0xea12900 (UNKNOWN)
winlogon.exe[616]               inline   wininet.dll!InternetQueryDataAvailable[0x771a325fL] 0x771a325f JMP 0xea0f540 (UNKNOWN)
winlogon.exe[616]               inline   wininet.dll!InternetReadFile[0x77199555L] 0x77199555 JMP 0xea12660 (UNKNOWN)
winlogon.exe[616]               inline   wininet.dll!InternetReadFileExA[0x771c7e9aL] 0x771c7e9a JMP 0xea127b0 (UNKNOWN)
winlogon.exe[616]               inline   wininet.dll!InternetWriteFile[0x771c7953L] 0x771c7953 JMP 0xea079b7 (UNKNOWN)
winlogon.exe[616]               inline   advapi32.dll!CryptEncrypt[0x77dc1558L]   0x77dc1558 JMP 0xea06f36 (UNKNOWN)
winlogon.exe[616]               inline   ws2_32.dll!send[0x71a1428aL]             0x71a1428a JMP 0xea0e736 (UNKNOWN)
winlogon.exe[616]               inline   crypt32.dll!PFXImportCertStore[0x77abf748L] 0x77abf748 JMP 0xea023af (UNKNOWN)
```

Memory, disassembler and structures view via the interactive shell



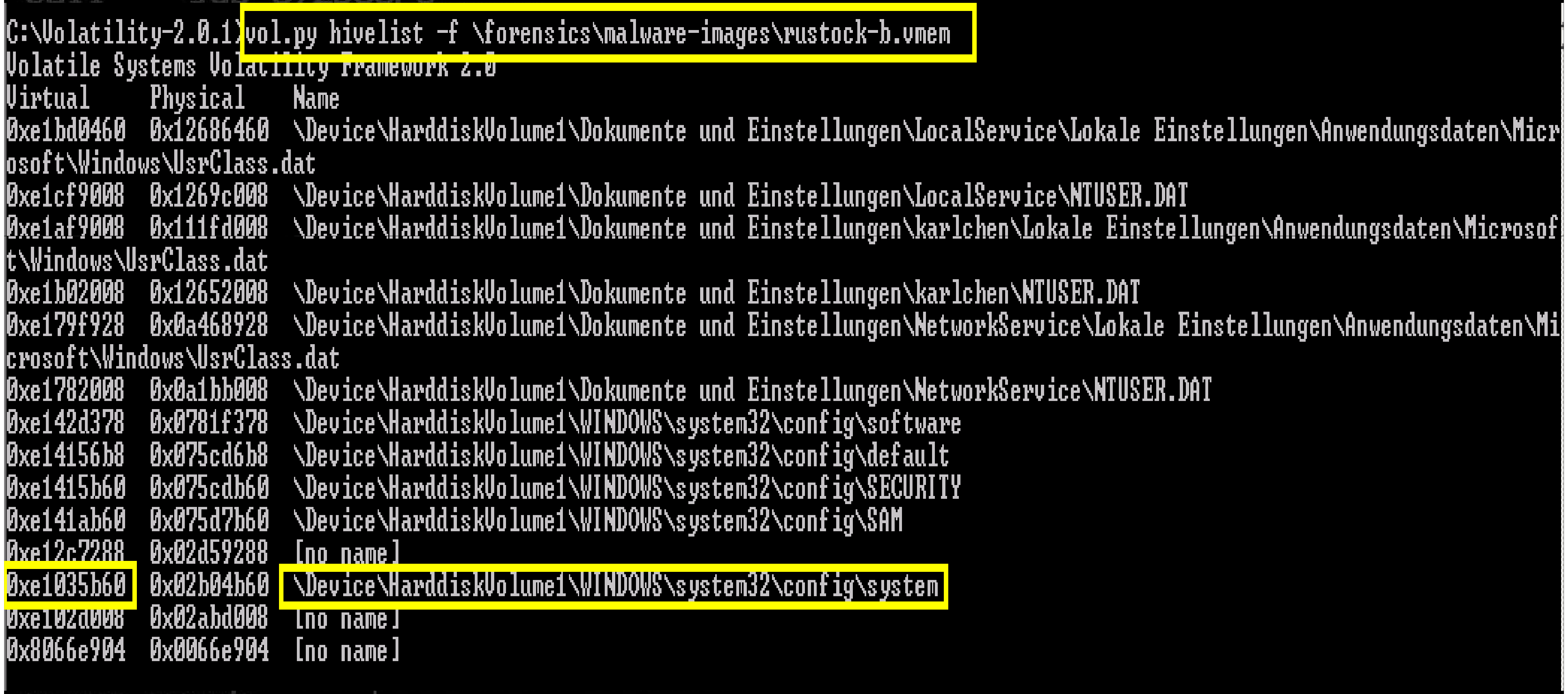
```
c:\Volatility-2.0.1>vol.py volshell -f \forensics\malware-images\spyeye.vmem
Volatile Systems Volatility Framework 2.0
Current context: process System, pid=4, ppid=0 DTB=0xaf9000
Welcome to volshell! Current memory image is:
file:///C:/forensics/malware-images/spyeye.vmem
To get help, type 'hh()'
>>> cc(offset=None, pid=616, name=None)
Current context: process winlogon.exe, pid=616, ppid=444 DTB=0x7180060
>>> db(0xea00000, length=128, space=None)
0ea00000   4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0ea00010   b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0ea00020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0ea00030   00 00 00 00 00 00 00 00 00 00 00 00 d0 00 00 00   ................
0ea00040   0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 68   ........!..L.!Th
0ea00050   69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f   is program canno
0ea00060   74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20   t be run in DOS
0ea00070   6d 6f 64 65 2e 0d 0d 0a 24 00 00 00 00 00 00 00   mode....$.......

>>> dis(0x7c9261ca, length=5, space=None)
0x7c9261ca e9e2d20d92                                 JMP 0xea034b1
>>> dis(0xea034b1, length=40, space=None)
0xea034b1 55                                          PUSH EBP
0xea034b2 8bec                                        MOV EBP, ESP
0xea034b4 81ecac000000                                SUB ESP, 0xac
0xea034ba 53                                          PUSH EBX
0xea034bb 56                                          PUSH ESI
0xea034bc 57                                          PUSH EDI
0xea034bd 40                                          INC EAX
0xea034be 48                                          DEC EAX
0xea034bf c745f87077a20e                              MOV DWORD [EBP-0x8], 0xea27770
0xea034c6 8b4514                                      MOV EAX, [EBP+0x14]
0xea034c9 50                                          PUSH EAX
0xea034ca 8b4510                                      MOV EAX, [EBP+0x10]
0xea034cd 50                                          PUSH EAX
0xea034ce 8b450c                                      MOV EAX, [EBP+0xc]
0xea034d1 50                                          PUSH EAX
0xea034d2 8b4508                                      MOV EAX, [EBP+0x8]
0xea034d5 50                                          PUSH EAX
0xea034d6 ff55f8                                      CALL DWORD [EBP-0x8]
>>> quit()

c:\Volatility-2.0.1>_
```

Registry Hives

- Table of standard hives and their supporting files

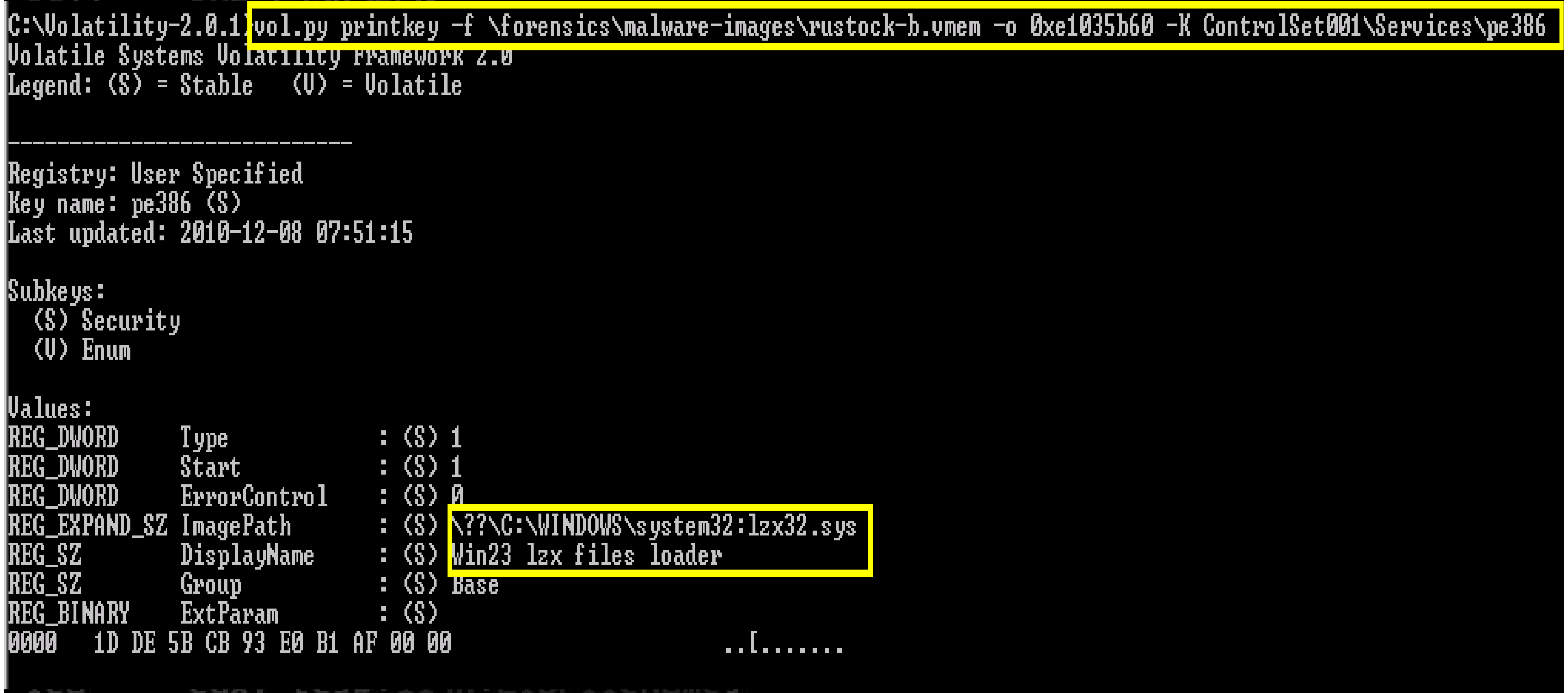
Registry hive	Supporting files
HKEY_CURRENT_CONFIG	System, System.alt, System.log, System.sav
HKEY_CURRENT_USER	Ntuser.dat, Ntuser.dat.log
HKEY_LOCAL_MACHINE\SAM	Sam, Sam.log, Sam.sav
HKEY_LOCAL_MACHINE\Security	Security, Security.log, Security.sav
HKEY_LOCAL_MACHINE\Software	Software, Software.log, Software.sav
HKEY_LOCAL_MACHINE\System	System, System.alt, System.log, System.sav
HKEY_USERS\.DEFAULT	Default, Default.log, Default.sav

Show registry hives of a system via _CMHIVE parsing, e.g. …\config\system points to registered services on a windows system



```
C:\Volatility-2.0.1>vol.py hivelist -f \forensics\malware-images\rustock-b.vmem
Volatile Systems Volatility Framework 2.0
Virtual     Physical    Name
0xe1bd0460  0x12686460  \Device\HarddiskVolume1\Dokumente und Einstellungen\LocalService\Lokale Einstellungen\Anwendungsdaten\Micr
osoft\Windows\UsrClass.dat
0xe1cf9008  0x1269c008  \Device\HarddiskVolume1\Dokumente und Einstellungen\LocalService\NTUSER.DAT
0xe1af9008  0x111fd008  \Device\HarddiskVolume1\Dokumente und Einstellungen\karlchen\Lokale Einstellungen\Anwendungsdaten\Microsof
t\Windows\UsrClass.dat
0xe1b02008  0x12652008  \Device\HarddiskVolume1\Dokumente und Einstellungen\karlchen\NTUSER.DAT
0xe179f928  0x0a468928  \Device\HarddiskVolume1\Dokumente und Einstellungen\NetworkService\Lokale Einstellungen\Anwendungsdaten\Mi
crosoft\Windows\UsrClass.dat
0xe1782008  0x0a1bb008  \Device\HarddiskVolume1\Dokumente und Einstellungen\NetworkService\NTUSER.DAT
0xe142d378  0x0781f378  \Device\HarddiskVolume1\WINDOWS\system32\config\software
0xe14156b8  0x075cd6b8  \Device\HarddiskVolume1\WINDOWS\system32\config\default
0xe1415b60  0x075cdb60  \Device\HarddiskVolume1\WINDOWS\system32\config\SECURITY
0xe141ab60  0x075d7b60  \Device\HarddiskVolume1\WINDOWS\system32\config\SAM
0xe12c7288  0x02d59288  [no name]
0xe1035b60  0x02b04b60  \Device\HarddiskVolume1\WINDOWS\system32\config\system
0xe102d008  0x02abd008  [no name]
0x8066e904  0x0066e904  [no name]
```

Show registry key that looks suspicious or was hidden through API hooking on a live system

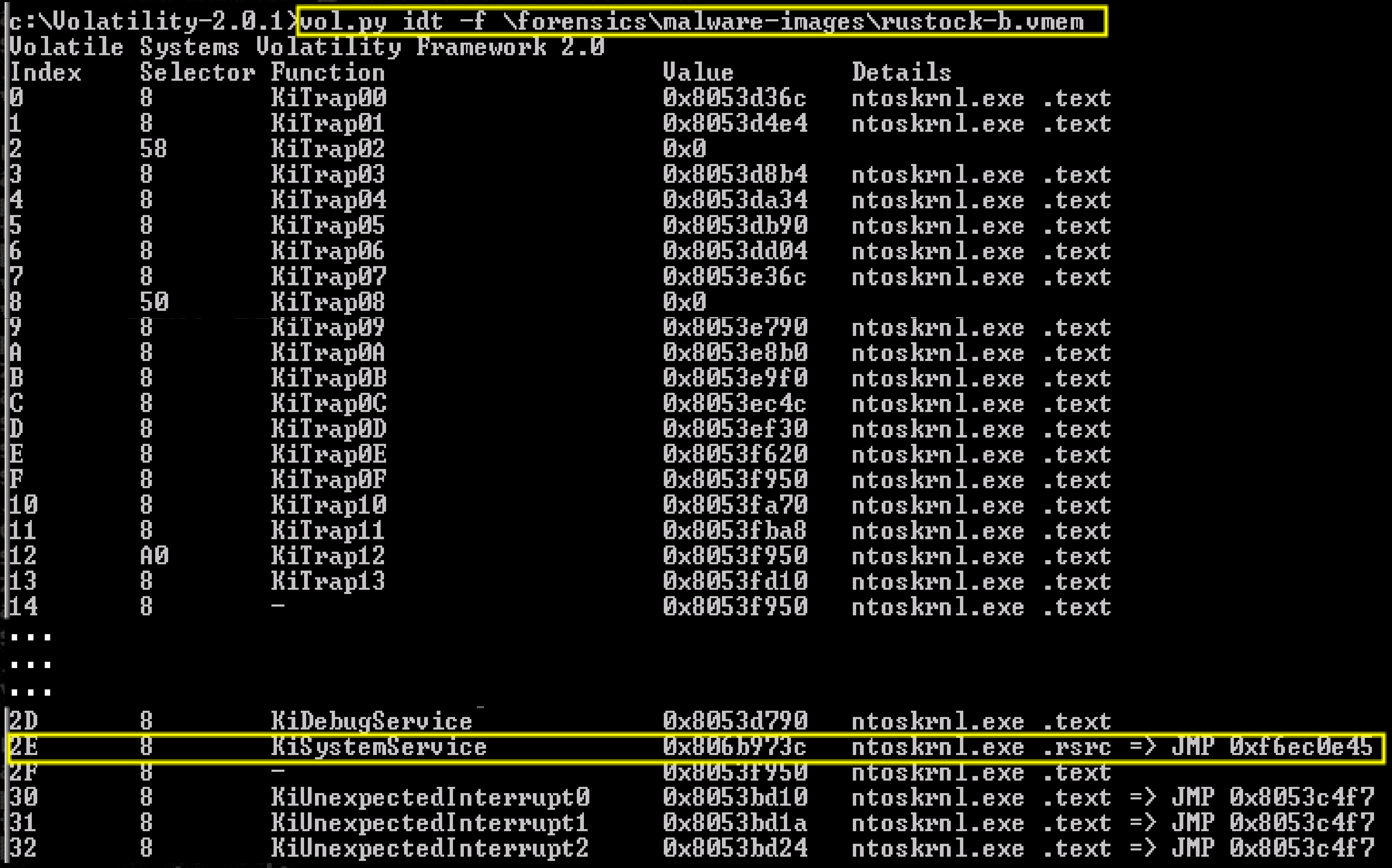


```
C:\Volatility-2.0.1>vol.py printkey -f \forensics\malware-images\rustock-b.vmem -o 0xe1035b60 -K ControlSet001\Services\pe386
Volatile Systems Volatility Framework 2.0
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: User Specified
Key name: pe386 (S)
Last updated: 2010-12-08 07:51:15

Subkeys:
  (S) Security
  (V) Enum

Values:
REG_DWORD     Type             : (S) 1
REG_DWORD     Start            : (S) 1
REG_DWORD     ErrorControl     : (S) 0
REG_EXPAND_SZ ImagePath        : (S) \??\C:\WINDOWS\system32:lzx32.sys
REG_SZ        DisplayName      : (S) Win23 lzx files loader
REG_SZ        Group            : (S) Base
REG_BINARY    ExtParam         : (S)
0000   1D DE 5B CB 93 E0 B1 AF 00 00                          ..[.......
```

Interrupt Descriptor Table (IDT)

- The Interrupt Descriptor Table (IDT) is a structure which is used when dispatching interrupts
- Interrupts can interrupt an execution of a program to to handle an event
- Interrupts could be a result of a hardware signal or software based using the INT instruction
- The IDT descriptor table can handle 256 entries
- The descriptor to the table can be written with the instruction LIDT and read with SIDT

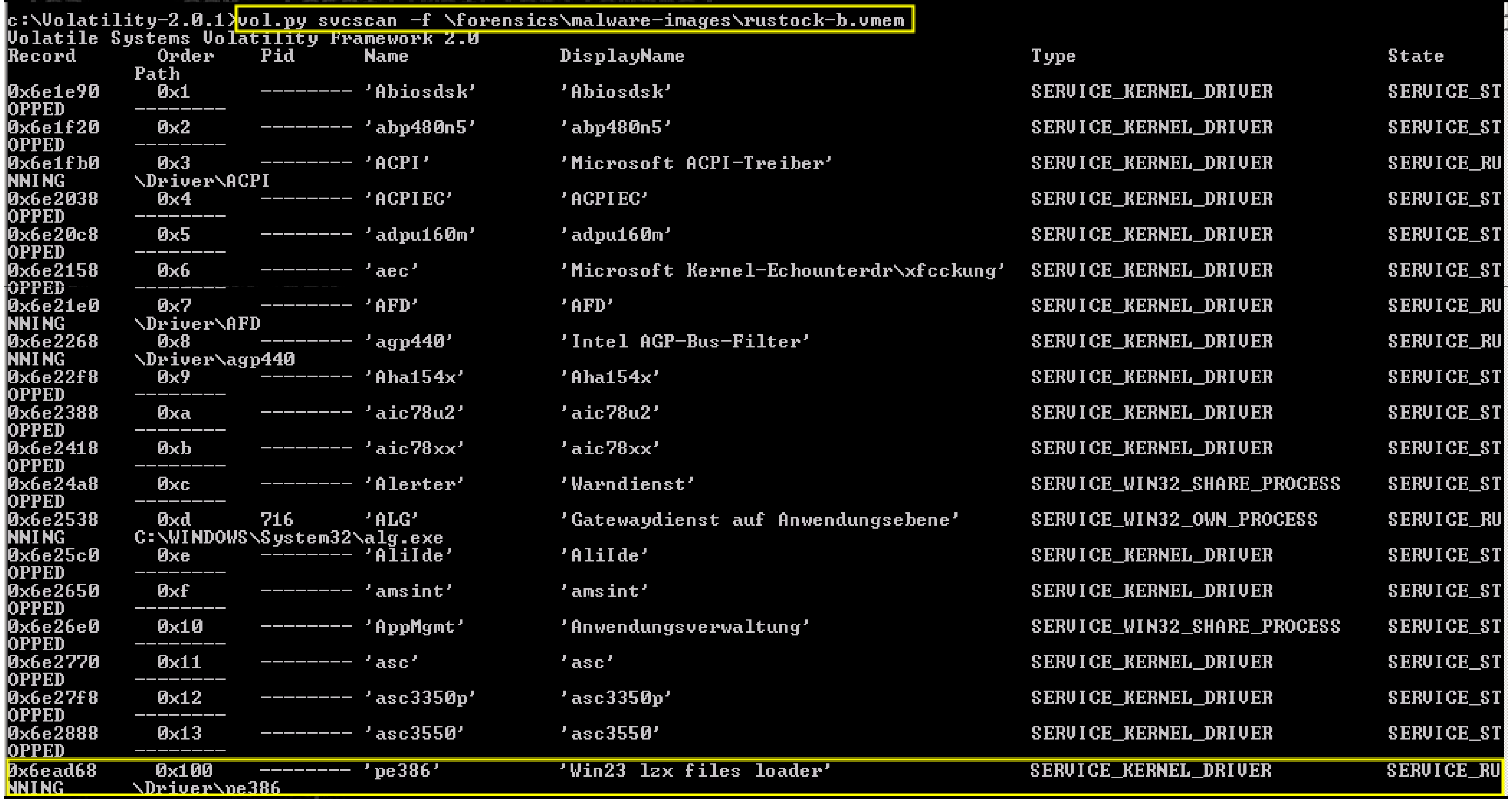
Show IDT to detect manipulated interrupts



```
c:\Volatility-2.0.1>vol.py idt -f \forensics\malware-images\rustock-b.vmem
Volatile Systems Volatility Framework 2.0
Index     Selector Function                    Value          Details
0         8        KiTrap00                    0x8053d36c     ntoskrnl.exe .text
1         8        KiTrap01                    0x8053d4e4     ntoskrnl.exe .text
2         58       KiTrap02                    0x0
3         8        KiTrap03                    0x8053d8b4     ntoskrnl.exe .text
4         8        KiTrap04                    0x8053da34     ntoskrnl.exe .text
5         8        KiTrap05                    0x8053db90     ntoskrnl.exe .text
6         8        KiTrap06                    0x8053dd04     ntoskrnl.exe .text
7         8        KiTrap07                    0x8053e36c     ntoskrnl.exe .text
8         50       KiTrap08                    0x0
9         8        KiTrap09                    0x8053e790     ntoskrnl.exe .text
A         8        KiTrap0A                    0x8053e8b0     ntoskrnl.exe .text
B         8        KiTrap0B                    0x8053e9f0     ntoskrnl.exe .text
C         8        KiTrap0C                    0x8053ec4c     ntoskrnl.exe .text
D         8        KiTrap0D                    0x8053ef30     ntoskrnl.exe .text
E         8        KiTrap0E                    0x8053f620     ntoskrnl.exe .text
F         8        KiTrap0F                    0x8053f950     ntoskrnl.exe .text
10        8        KiTrap10                    0x8053fa70     ntoskrnl.exe .text
11        8        KiTrap11                    0x8053fba8     ntoskrnl.exe .text
12        A0       KiTrap12                    0x8053f950     ntoskrnl.exe .text
13        8        KiTrap13                    0x8053fd10     ntoskrnl.exe .text
14        8        -                           0x8053f950     ntoskrnl.exe .text
...
...
...
2D        8        KiDebugService              0x8053d790     ntoskrnl.exe .text
2E        8        KiSystemService             0x806b973c     ntoskrnl.exe .rsrc => JMP 0xf6ec0e45
2F        8        -                           0x8053f950     ntoskrnl.exe .text
30        8        KiUnexpectedInterrupt0      0x8053bd10     ntoskrnl.exe .text => JMP 0x8053c4f7
31        8        KiUnexpectedInterrupt1      0x8053bd1a     ntoskrnl.exe .text => JMP 0x8053c4f7
32        8        KiUnexpectedInterrupt2      0x8053bd24     ntoskrnl.exe .text => JMP 0x8053c4f7
```

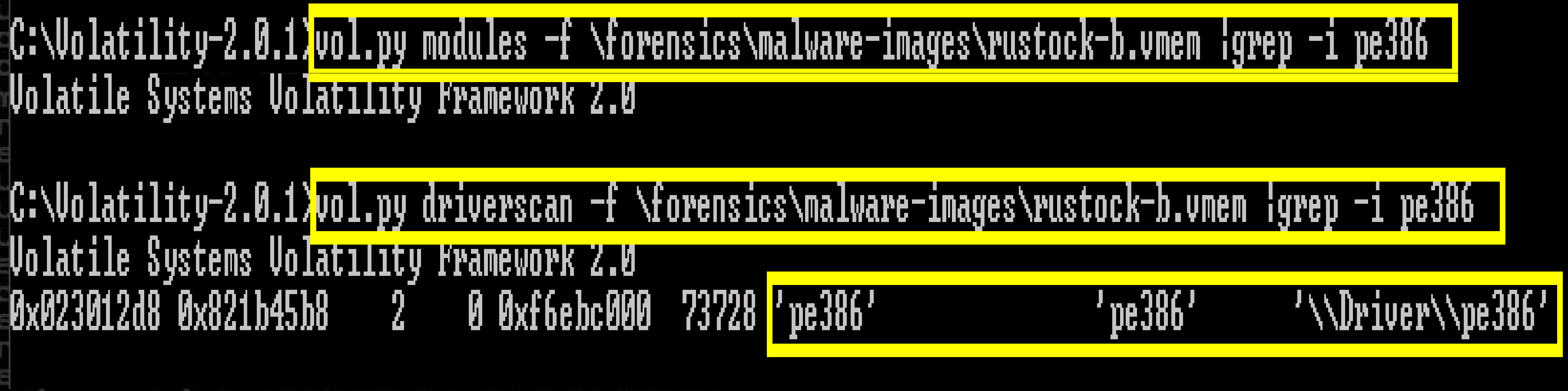
FRANK BOLDEWIN'S WWW.RECONSTRUCTER.ORG

Show registered services (incl. hidden) and status via _SERVICE* records



```
c:\Volatility-2.0.1>vol.py svcscan -f \forensics\malware-images\rustock-b.vmem
Volatile Systems Volatility Framework 2.0
Record          Order     Pid       Name                DisplayName                                    Type                          State
            Path
0x6e1e90        0x1       --------- 'Abiosdsk'          'Abiosdsk'                                     SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e1f20        0x2       --------- 'abp480n5'          'abp480n5'                                     SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e1fb0        0x3       --------- 'ACPI'              'Microsoft ACPI-Treiber'                       SERVICE_KERNEL_DRIVER         SERVICE_RU
NNING       \Driver\ACPI
0x6e2038        0x4       --------- 'ACPIEC'            'ACPIEC'                                       SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e20c8        0x5       --------- 'adpu160m'          'adpu160m'                                     SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e2158        0x6       --------- 'aec'               'Microsoft Kernel-Echounterdr\xfcckung'        SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e21e0        0x7       --------- 'AFD'               'AFD'                                          SERVICE_KERNEL_DRIVER         SERVICE_RU
NNING       \Driver\AFD
0x6e2268        0x8       --------- 'agp440'            'Intel AGP-Bus-Filter'                         SERVICE_KERNEL_DRIVER         SERVICE_RU
NNING       \Driver\agp440
0x6e22f8        0x9       --------- 'Aha154x'           'Aha154x'                                      SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e2388        0xa       --------- 'aic78u2'           'aic78u2'                                      SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e2418        0xb       --------- 'aic78xx'           'aic78xx'                                      SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e24a8        0xc       --------- 'Alerter'           'Warndienst'                                   SERVICE_WIN32_SHARE_PROCESS   SERVICE_ST
OPPED       ---------
0x6e2538        0xd       716       'ALG'               'Gatewaydienst auf Anwendungsebene'            SERVICE_WIN32_OWN_PROCESS     SERVICE_RU
NNING       C:\WINDOWS\System32\alg.exe
0x6e25c0        0xe       --------- 'AliIde'            'AliIde'                                       SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e2650        0xf       --------- 'amsint'            'amsint'                                       SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e26e0        0x10      --------- 'AppMgmt'           'Anwendungsverwaltung'                         SERVICE_WIN32_SHARE_PROCESS   SERVICE_ST
OPPED       ---------
0x6e2770        0x11      --------- 'asc'               'asc'                                          SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e27f8        0x12      --------- 'asc3350p'          'asc3350p'                                     SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6e2888        0x13      --------- 'asc3550'           'asc3550'                                      SERVICE_KERNEL_DRIVER         SERVICE_ST
OPPED       ---------
0x6ead68        0x100     --------- 'pe386'             'Win23 lzx files loader'                       SERVICE_KERNEL_DRIVER         SERVICE_RU
NNING       \Driver\pe386
```

Comparing the results of function "modules" via PsLoadedModuleList and function "driverscan" via _DRIVER_OBJECT parsing. Driverscan shows the hidden driver

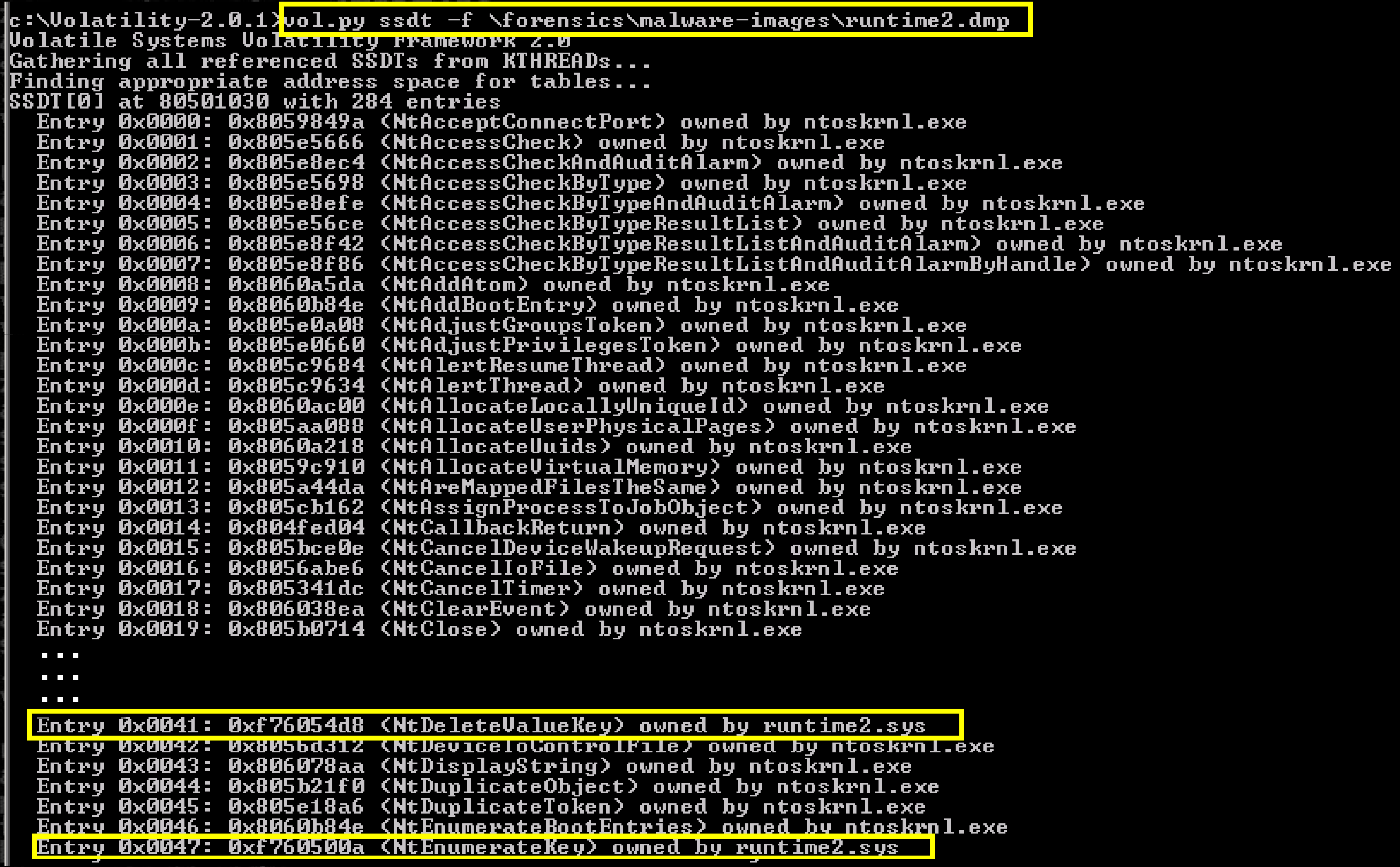


```
C:\Volatility-2.0.1>vol.py modules -f \forensics\malware-images\rustock-b.vmem |grep -i pe386
Volatile Systems Volatility Framework 2.0

C:\Volatility-2.0.1>vol.py driverscan -f \forensics\malware-images\rustock-b.vmem |grep -i pe386
Volatile Systems Volatility Framework 2.0
0x023012d8 0x821b45b8    2    0 0xf6ebc000   73728 'pe386'            'pe386'     '\\Driver\\pe386'
```

SSDT and Shadow SSDT

- The SSDT is a data array in kernel memory, that stores pointers to the native API functions of Windows, e.g. NtCreateFile, NtEnumerateKey
- These functions are handled in NTOSKRNL
- Some older rootkits hooked some distinctive functions to hide its files or registry entries when queried from usermode
- Another data array is the Shadow SSDT, pointing to native graphic and windows related functions, handled in Win32k.sys

Finding manipulated SSDT und Shadow SSDT entries

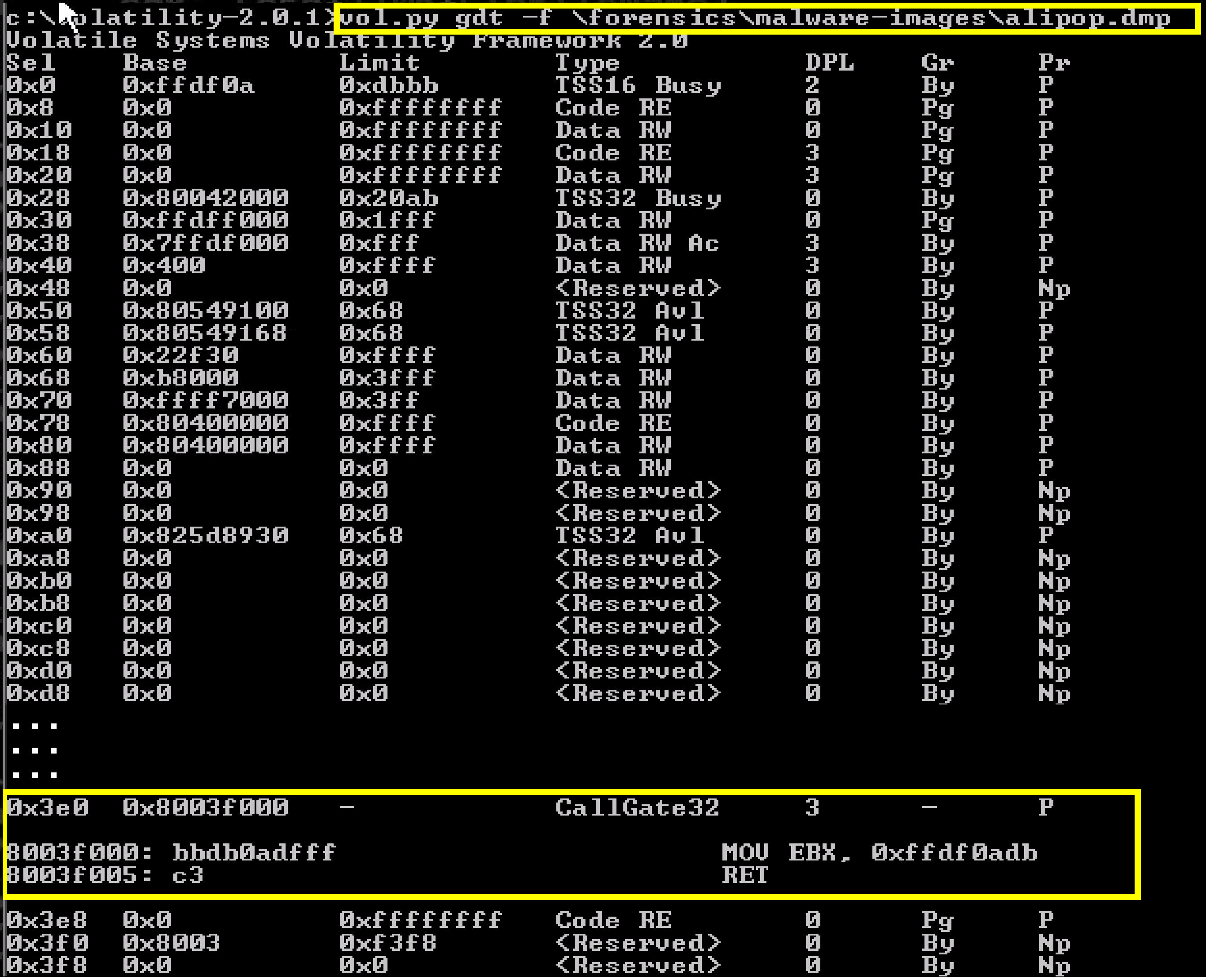


```
c:\Volatility-2.0.1>vol.py ssdt -f \forensics\malware-images\runtime2.dmp
Volatile Systems Volatility Framework 2.0
Gathering all referenced SSDTs from KTHREADs...
Finding appropriate address space for tables...
SSDT[0] at 80501030 with 284 entries
  Entry 0x0000: 0x8059849a (NtAcceptConnectPort) owned by ntoskrnl.exe
  Entry 0x0001: 0x805e5666 (NtAccessCheck) owned by ntoskrnl.exe
  Entry 0x0002: 0x805e8ec4 (NtAccessCheckAndAuditAlarm) owned by ntoskrnl.exe
  Entry 0x0003: 0x805e5698 (NtAccessCheckByType) owned by ntoskrnl.exe
  Entry 0x0004: 0x805e8efe (NtAccessCheckByTypeAndAuditAlarm) owned by ntoskrnl.exe
  Entry 0x0005: 0x805e56ce (NtAccessCheckByTypeResultList) owned by ntoskrnl.exe
  Entry 0x0006: 0x805e8f42 (NtAccessCheckByTypeResultListAndAuditAlarm) owned by ntoskrnl.exe
  Entry 0x0007: 0x805e8f86 (NtAccessCheckByTypeResultListAndAuditAlarmByHandle) owned by ntoskrnl.exe
  Entry 0x0008: 0x8060a5da (NtAddAtom) owned by ntoskrnl.exe
  Entry 0x0009: 0x8060b84e (NtAddBootEntry) owned by ntoskrnl.exe
  Entry 0x000a: 0x805e0a08 (NtAdjustGroupsToken) owned by ntoskrnl.exe
  Entry 0x000b: 0x805e0660 (NtAdjustPrivilegesToken) owned by ntoskrnl.exe
  Entry 0x000c: 0x805c9684 (NtAlertResumeThread) owned by ntoskrnl.exe
  Entry 0x000d: 0x805c9634 (NtAlertThread) owned by ntoskrnl.exe
  Entry 0x000e: 0x8060ac00 (NtAllocateLocallyUniqueId) owned by ntoskrnl.exe
  Entry 0x000f: 0x805aa088 (NtAllocateUserPhysicalPages) owned by ntoskrnl.exe
  Entry 0x0010: 0x8060a218 (NtAllocateUuids) owned by ntoskrnl.exe
  Entry 0x0011: 0x8059c910 (NtAllocateVirtualMemory) owned by ntoskrnl.exe
  Entry 0x0012: 0x805a44da (NtAreMappedFilesTheSame) owned by ntoskrnl.exe
  Entry 0x0013: 0x805cb162 (NtAssignProcessToJobObject) owned by ntoskrnl.exe
  Entry 0x0014: 0x804fed04 (NtCallbackReturn) owned by ntoskrnl.exe
  Entry 0x0015: 0x805bce0e (NtCancelDeviceWakeupRequest) owned by ntoskrnl.exe
  Entry 0x0016: 0x8056abe6 (NtCancelIoFile) owned by ntoskrnl.exe
  Entry 0x0017: 0x805341dc (NtCancelTimer) owned by ntoskrnl.exe
  Entry 0x0018: 0x806038ea (NtClearEvent) owned by ntoskrnl.exe
  Entry 0x0019: 0x805b0714 (NtClose) owned by ntoskrnl.exe
  ...
  ...
  ...
  Entry 0x0041: 0xf76054d8 (NtDeleteValueKey) owned by runtime2.sys
  Entry 0x0042: 0x8056d312 (NtDeviceIoControlFile) owned by ntoskrnl.exe
  Entry 0x0043: 0x806078aa (NtDisplayString) owned by ntoskrnl.exe
  Entry 0x0044: 0x805b21f0 (NtDuplicateObject) owned by ntoskrnl.exe
  Entry 0x0045: 0x805e18a6 (NtDuplicateToken) owned by ntoskrnl.exe
  Entry 0x0046: 0x8060b84e (NtEnumerateBootEntries) owned by ntoskrnl.exe
  Entry 0x0047: 0xf760500a (NtEnumerateKey) owned by runtime2.sys
```

Global Descriptor Table (GDT) and callgates

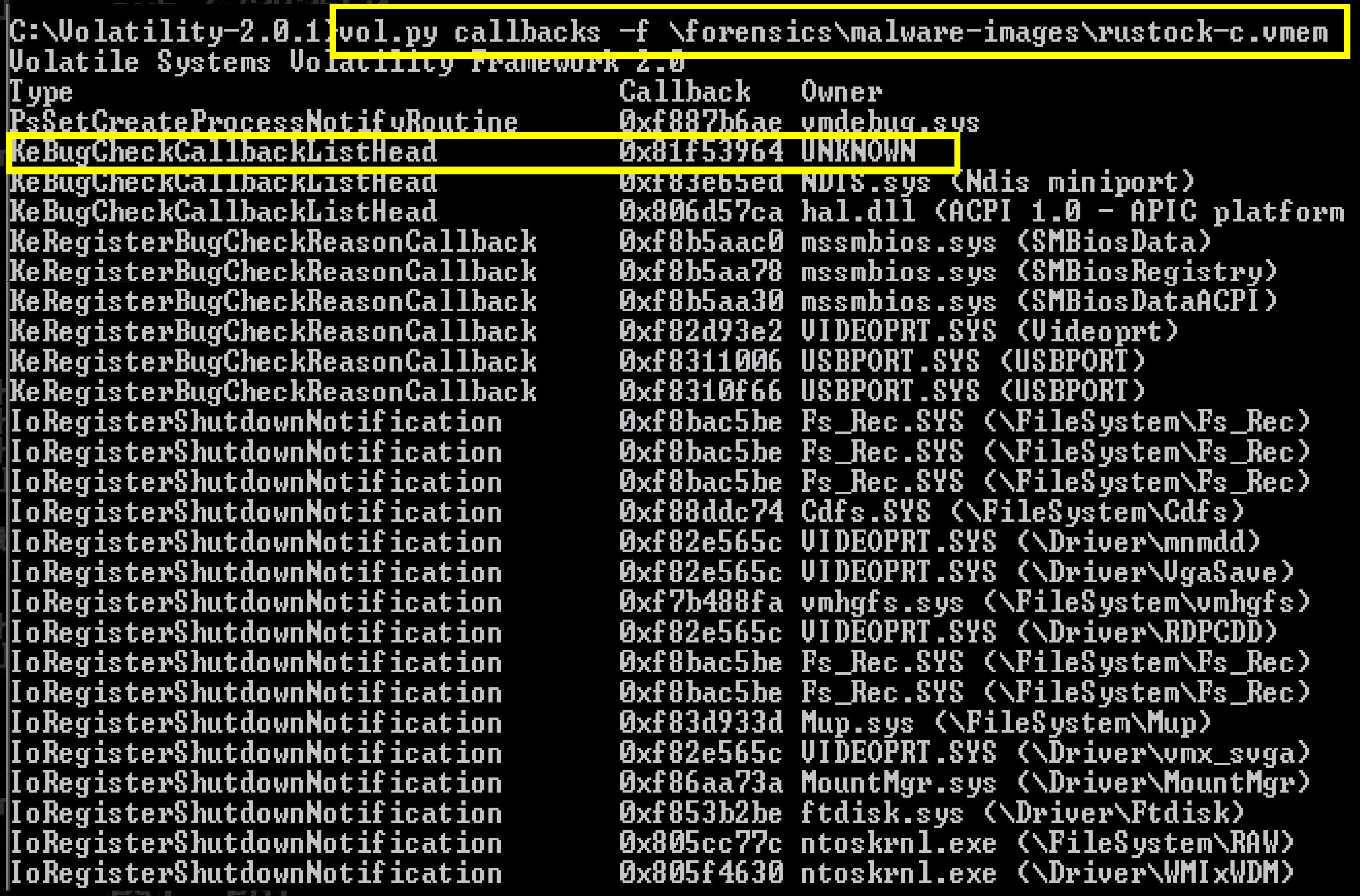
- The GDT is a table used in protected mode of a x86 CPU to manage memory, multitasking and different callgates
- A callgate is a mechanism in Intel x86 arch to change privilege level of the CPU
- Some rootkits install such callgates to execute code with the highest privilege (Ring 0) from usermode (Ring 3) without the need to have a driver, e.g. by calling DeviceIOControl
- Callgate usage works by executing "call far ptr <addr>" from usermode code

Show Global Descriptor Table to detect installed callgates



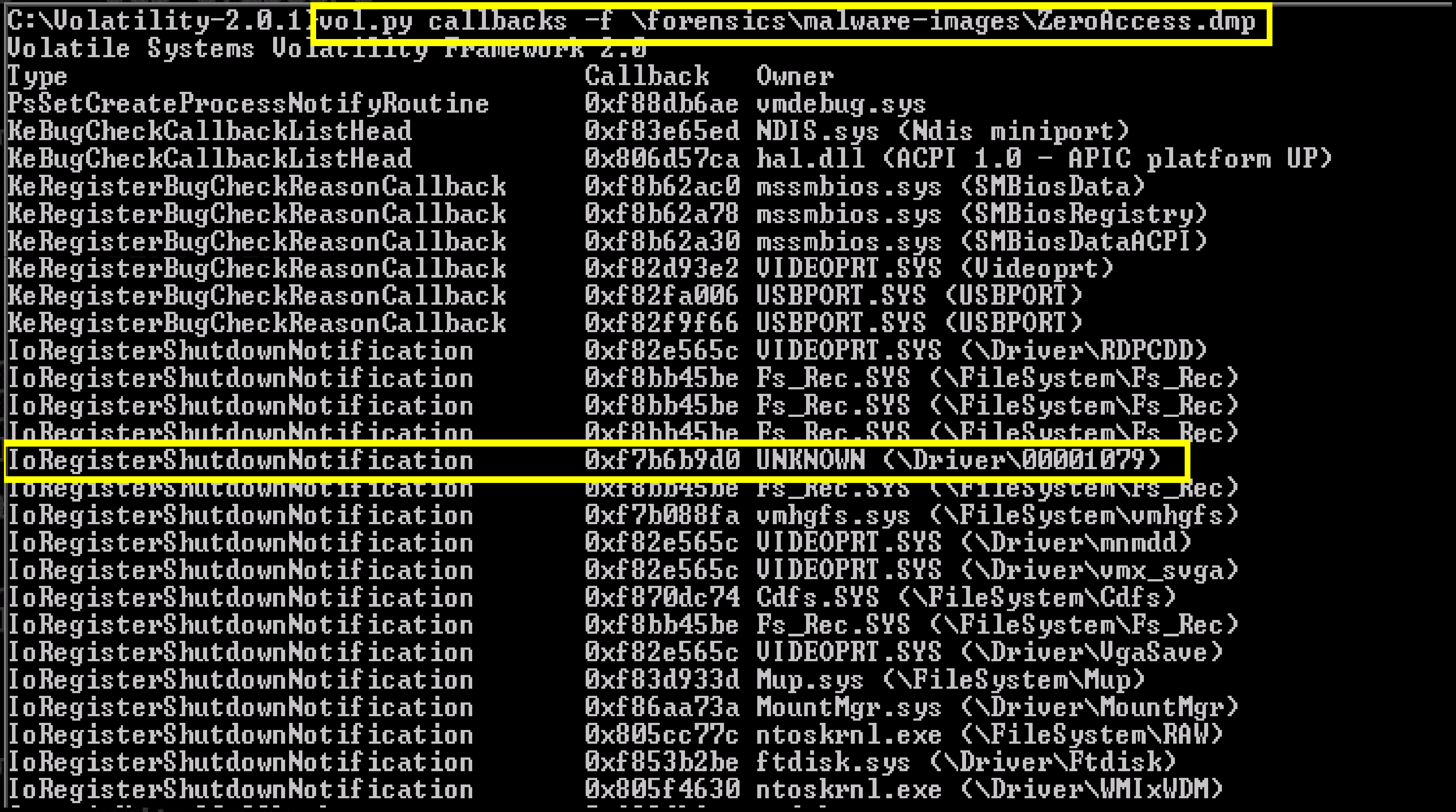
```
c:\volatility-2.0.1>vol.py gdt -f \forensics\malware-images\alipop.dmp
Volatile Systems Volatility Framework 2.0
Sel    Base        Limit       Type          DPL    Gr     Pr
0x0    0xffdf0a    0xdbbb      TSS16 Busy    2      By     P
0x8    0x0         0xffffffff  Code  RE      0      Pg     P
0x10   0x0         0xffffffff  Data  RW      0      Pg     P
0x18   0x0         0xffffffff  Code  RE      3      Pg     P
0x20   0x0         0xffffffff  Data  RW      3      Pg     P
0x28   0x80042000  0x20ab      TSS32 Busy    0      By     P
0x30   0xffdff000  0x1fff      Data  RW      0      Pg     P
0x38   0x7ffdf000  0xfff       Data  RW Ac   3      By     P
0x40   0x400       0xffff      Data  RW      3      By     P
0x48   0x0         0x0         <Reserved>    0      By     Np
0x50   0x80549100  0x68        TSS32 Avl     0      By     P
0x58   0x80549168  0x68        TSS32 Avl     0      By     P
0x60   0x22f30     0xffff      Data  RW      0      By     P
0x68   0xb8000     0x3fff      Data  RW      0      By     P
0x70   0xffff7000  0x3ff       Data  RW      0      By     P
0x78   0x80400000  0xffff      Code  RE      0      By     P
0x80   0x80400000  0xffff      Data  RW      0      By     P
0x88   0x0         0x0         Data  RW      0      By     P
0x90   0x0         0x0         <Reserved>    0      By     Np
0x98   0x0         0x0         <Reserved>    0      By     Np
0xa0   0x825d8930  0x68        TSS32 Avl     0      By     P
0xa8   0x0         0x0         <Reserved>    0      By     Np
0xb0   0x0         0x0         <Reserved>    0      By     Np
0xb8   0x0         0x0         <Reserved>    0      By     Np
0xc0   0x0         0x0         <Reserved>    0      By     Np
0xc8   0x0         0x0         <Reserved>    0      By     Np
0xd0   0x0         0x0         <Reserved>    0      By     Np
0xd8   0x0         0x0         <Reserved>    0      By     Np
...
...
...
0x3e0  0x8003f000  -           CallGate32    3      -      P

8003f000: bbdb0adfff                           MOV EBX, 0xffdf0adb
8003f005: c3                                   RET

0x3e8  0x0         0xffffffff  Code  RE      0      Pg     P
0x3f0  0x8003      0xf3f8      <Reserved>    0      By     Np
0x3f8  0x0         0x0         <Reserved>    0      By     Np
```

Kernel callback which is being called when a bugcheck occurs and possibly a crashdump is being created, e.g. to clean up malicious code pages

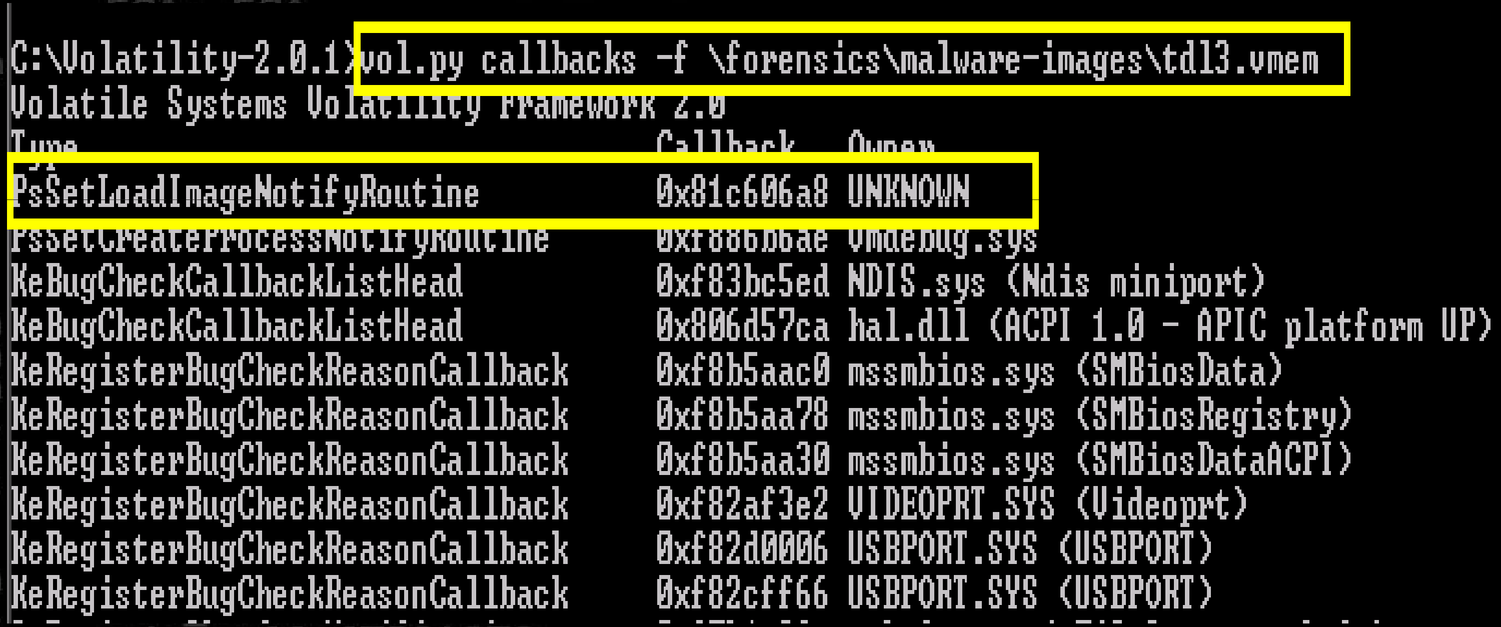


```
C:\Volatility-2.0.1>vol.py callbacks -f \forensics\malware-images\rustock-c.vmem
Volatile Systems Volatility Framework 2.0
Type                                 Callback   Owner
PsSetCreateProcessNotifyRoutine      0xf887b6ae vmdebug.sys
KeBugCheckCallbackListHead           0x81f53964 UNKNOWN
KeBugCheckCallbackListHead           0xf83e65ed NDIS.sys (Ndis miniport)
KeBugCheckCallbackListHead           0x806d57ca hal.dll (ACPI 1.0 - APIC platform UP)
KeRegisterBugCheckReasonCallback     0xf8b5aac0 mssmbios.sys (SMBiosData)
KeRegisterBugCheckReasonCallback     0xf8b5aa78 mssmbios.sys (SMBiosRegistry)
KeRegisterBugCheckReasonCallback     0xf8b5aa30 mssmbios.sys (SMBiosDataACPI)
KeRegisterBugCheckReasonCallback     0xf82d93e2 VIDEOPRT.SYS (Videoprt)
KeRegisterBugCheckReasonCallback     0xf8311006 USBPORT.SYS (USBPORT)
KeRegisterBugCheckReasonCallback     0xf8310f66 USBPORT.SYS (USBPORT)
IoRegisterShutdownNotification       0xf8bac5be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf8bac5be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf8bac5be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf88ddc74 Cdfs.SYS (\FileSystem\Cdfs)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\mnmdd)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\VgaSave)
IoRegisterShutdownNotification       0xf7b488fa vmhgfs.sys (\FileSystem\vmhgfs)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\RDPCDD)
IoRegisterShutdownNotification       0xf8bac5be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf8bac5be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf83d933d Mup.sys (\FileSystem\Mup)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\vmx_svga)
IoRegisterShutdownNotification       0xf86aa73a MountMgr.sys (\Driver\MountMgr)
IoRegisterShutdownNotification       0xf853b2be ftdisk.sys (\Driver\Ftdisk)
IoRegisterShutdownNotification       0x805cc77c ntoskrnl.exe (\FileSystem\RAW)
IoRegisterShutdownNotification       0x805f4630 ntoskrnl.exe (\Driver\WMIxWDM)
```

Kernel callback which is being called when a system is about to shut down, e.g. to check if MBR is still properly infected

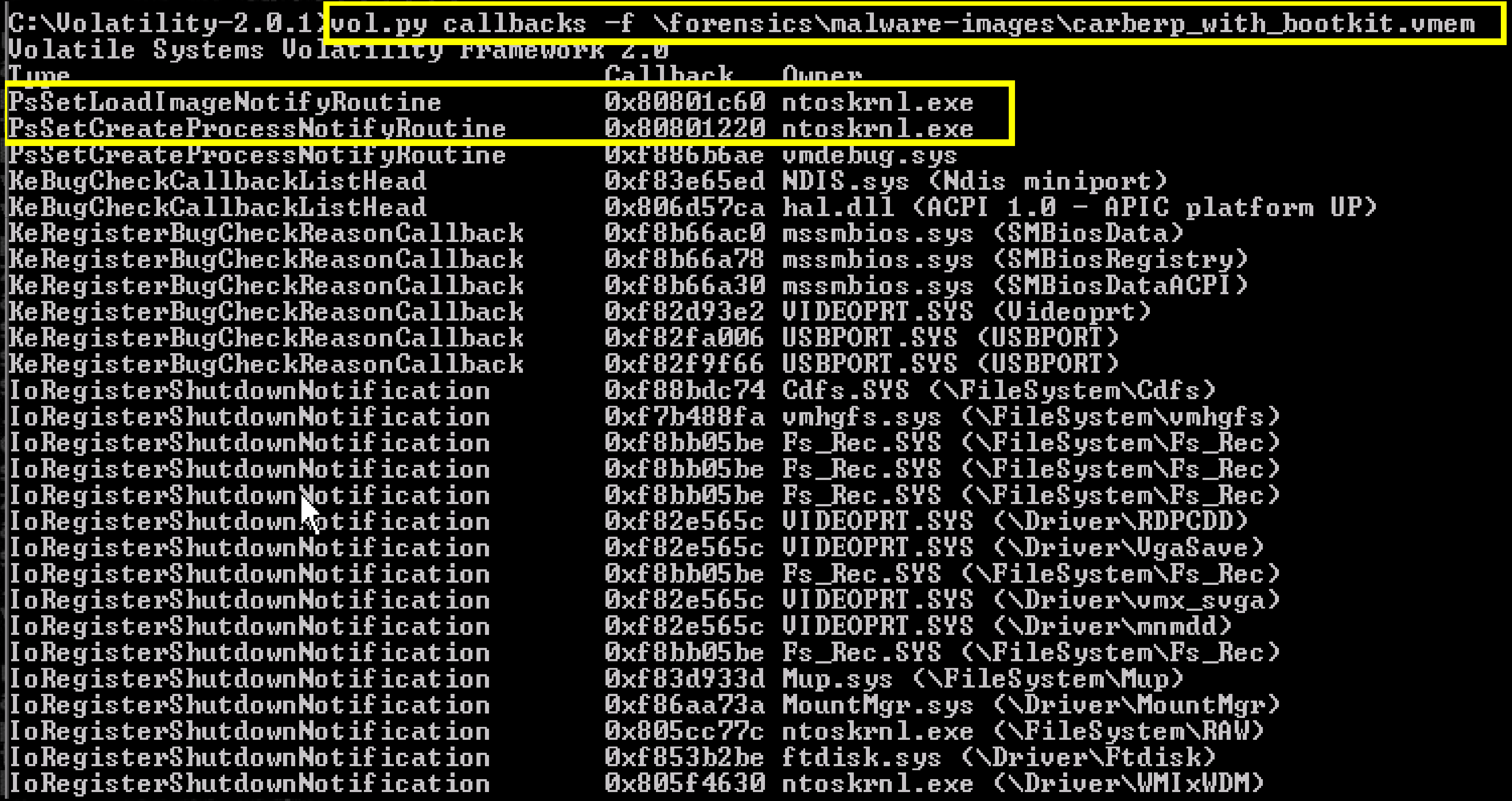

```
C:\Volatility-2.0.1>vol.py callbacks -f \forensics\malware-images\ZeroAccess.dmp
Volatile Systems Volatility Framework 2.0
Type                                 Callback   Owner
PsSetCreateProcessNotifyRoutine      0xf88db6ae vmdebug.sys
KeBugCheckCallbackListHead           0xf83e65ed NDIS.sys (Ndis miniport)
KeBugCheckCallbackListHead           0x806d57ca hal.dll (ACPI 1.0 - APIC platform UP)
KeRegisterBugCheckReasonCallback     0xf8b62ac0 mssmbios.sys (SMBiosData)
KeRegisterBugCheckReasonCallback     0xf8b62a78 mssmbios.sys (SMBiosRegistry)
KeRegisterBugCheckReasonCallback     0xf8b62a30 mssmbios.sys (SMBiosDataACPI)
KeRegisterBugCheckReasonCallback     0xf82d93e2 VIDEOPRT.SYS (Videoprt)
KeRegisterBugCheckReasonCallback     0xf82fa006 USBPORT.SYS (USBPORT)
KeRegisterBugCheckReasonCallback     0xf82f9f66 USBPORT.SYS (USBPORT)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\RDPCDD)
IoRegisterShutdownNotification       0xf8bb45be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf8bb45be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf8bb45be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf7b6b9d0 UNKNOWN (\Driver\00001079)
IoRegisterShutdownNotification       0xf8bb45be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf7b088fa vmhgfs.sys (\FileSystem\vmhgfs)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\mnmdd)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\vmx_svga)
IoRegisterShutdownNotification       0xf870dc74 Cdfs.SYS (\FileSystem\Cdfs)
IoRegisterShutdownNotification       0xf8bb45be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\VgaSave)
IoRegisterShutdownNotification       0xf83d933d Mup.sys (\FileSystem\Mup)
IoRegisterShutdownNotification       0xf86aa73a MountMgr.sys (\Driver\MountMgr)
IoRegisterShutdownNotification       0x805cc77c ntoskrnl.exe (\FileSystem\RAW)
IoRegisterShutdownNotification       0xf853b2be ftdisk.sys (\Driver\Ftdisk)
IoRegisterShutdownNotification       0x805f4630 ntoskrnl.exe (\Driver\WMIxWDM)
```

Kernel callback which is being called whenever a new module (Kernel+Usermode) gets loaded, e.g. to inject usermode code into the target process



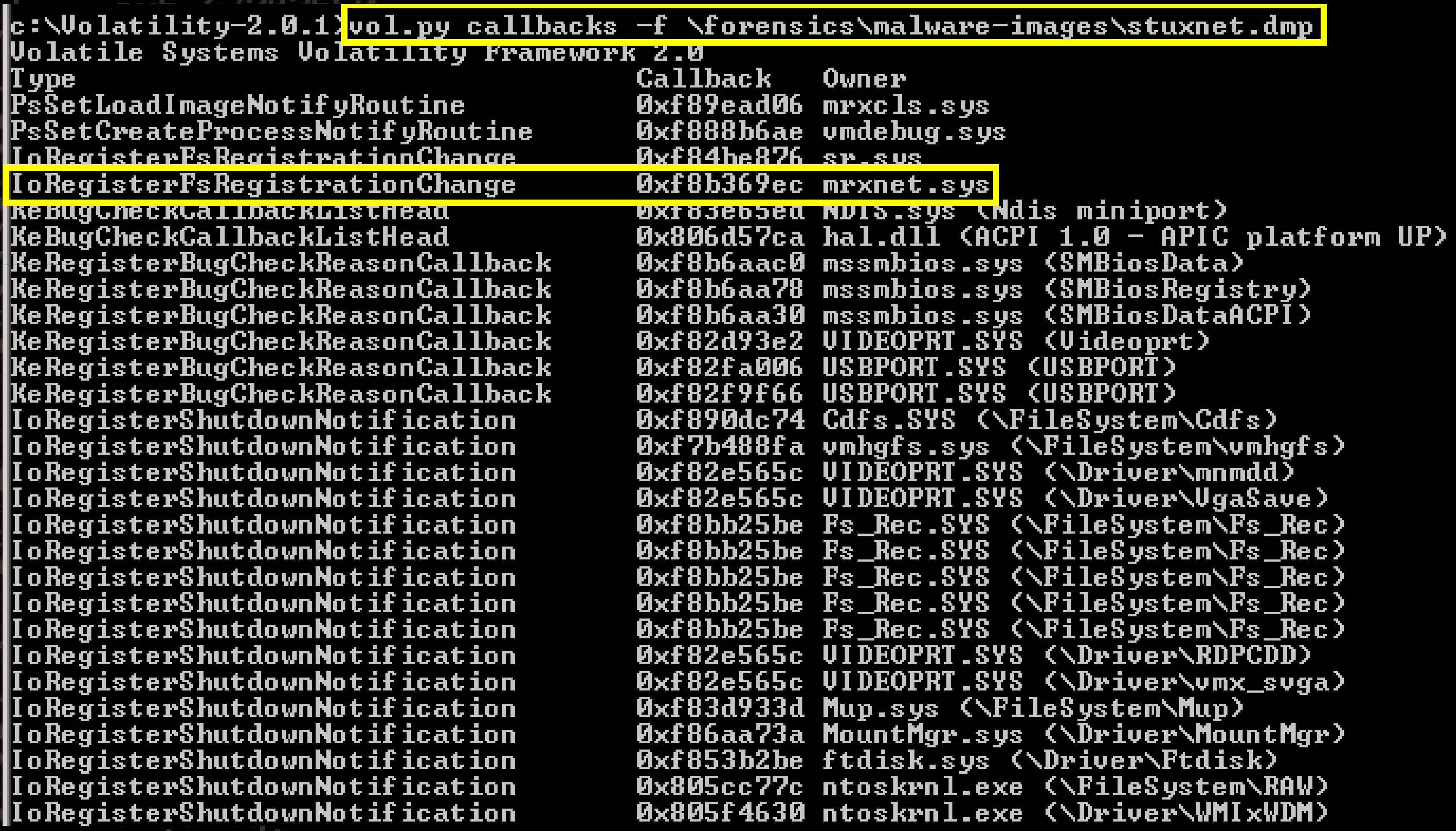
```
C:\Volatility-2.0.1>vol.py callbacks -f \forensics\malware-images\tdl3.vmem
Volatile Systems Volatility Framework 2.0
Type                                 Callback   Owner
PsSetLoadImageNotifyRoutine          0x81c606a8 UNKNOWN
PsSetCreateProcessNotifyRoutine      0xf886b6ae vmdebug.sys
KeBugCheckCallbackListHead           0xf83bc5ed NDIS.sys (Ndis miniport)
KeBugCheckCallbackListHead           0x806d57ca hal.dll (ACPI 1.0 - APIC platform UP)
KeRegisterBugCheckReasonCallback     0xf8b5aac0 mssmbios.sys (SMBiosData)
KeRegisterBugCheckReasonCallback     0xf8b5aa78 mssmbios.sys (SMBiosRegistry)
KeRegisterBugCheckReasonCallback     0xf8b5aa30 mssmbios.sys (SMBiosDataACPI)
KeRegisterBugCheckReasonCallback     0xf82af3e2 VIDEOPRT.SYS (Videoprt)
KeRegisterBugCheckReasonCallback     0xf82d0006 USBPORT.SYS (USBPORT)
KeRegisterBugCheckReasonCallback     0xf82cff66 USBPORT.SYS (USBPORT)
```

Kernel callbacks to fake NTOSKRNL.EXE, which is being called whenever a new module (Kernel+Usermode) gets loaded and a new process is created



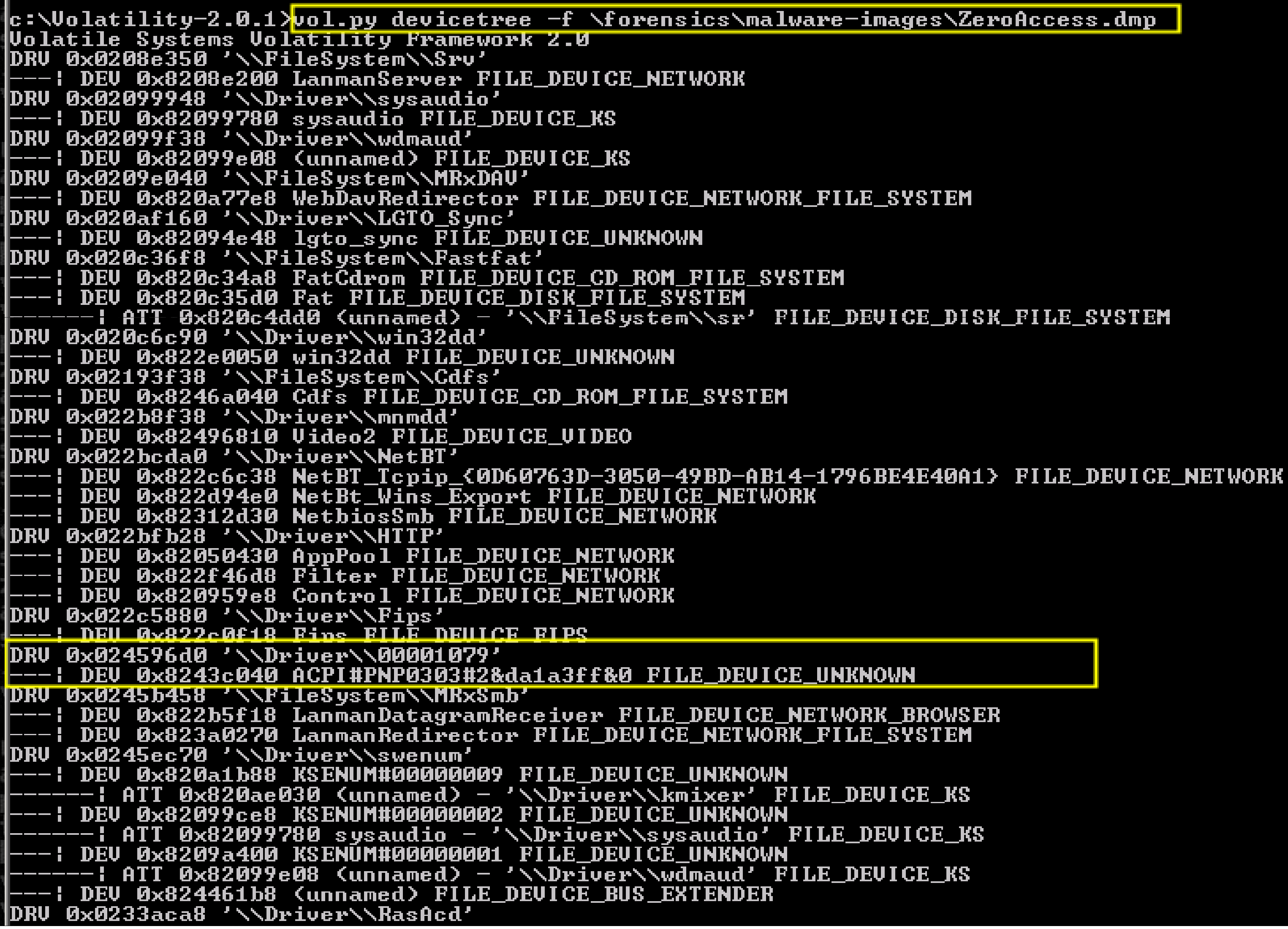
```
C:\Volatility-2.0.1>vol.py callbacks -f \forensics\malware-images\carberp_with_bootkit.vmem
Volatile Systems Volatility Framework 2.0
Type                                 Callback   Owner
PsSetLoadImageNotifyRoutine          0x80801c60 ntoskrnl.exe
PsSetCreateProcessNotifyRoutine      0x80801220 ntoskrnl.exe
PsSetCreateProcessNotifyRoutine      0xf886b6ae vmdebug.sys
KeBugCheckCallbackListHead           0xf83e65ed NDIS.sys (Ndis miniport)
KeBugCheckCallbackListHead           0x806d57ca hal.dll (ACPI 1.0 - APIC platform UP)
KeRegisterBugCheckReasonCallback     0xf8b66ac0 mssmbios.sys (SMBiosData)
KeRegisterBugCheckReasonCallback     0xf8b66a78 mssmbios.sys (SMBiosRegistry)
KeRegisterBugCheckReasonCallback     0xf8b66a30 mssmbios.sys (SMBiosDataACPI)
KeRegisterBugCheckReasonCallback     0xf82d93e2 VIDEOPRT.SYS (Videoprt)
KeRegisterBugCheckReasonCallback     0xf82fa006 USBPORT.SYS (USBPORT)
KeRegisterBugCheckReasonCallback     0xf82f9f66 USBPORT.SYS (USBPORT)
IoRegisterShutdownNotification       0xf88bdc74 Cdfs.SYS (\FileSystem\Cdfs)
IoRegisterShutdownNotification       0xf7b488fa vmhgfs.sys (\FileSystem\vmhgfs)
IoRegisterShutdownNotification       0xf8bb05be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf8bb05be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf8bb05be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\RDPCDD)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\VgaSave)
IoRegisterShutdownNotification       0xf8bb05be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\vmx_svga)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\mnmdd)
IoRegisterShutdownNotification       0xf8bb05be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf83d933d Mup.sys (\FileSystem\Mup)
IoRegisterShutdownNotification       0xf86aa73a MountMgr.sys (\Driver\MountMgr)
IoRegisterShutdownNotification       0x805cc77c ntoskrnl.exe (\FileSystem\RAW)
IoRegisterShutdownNotification       0xf853b2be ftdisk.sys (\Driver\Ftdisk)
IoRegisterShutdownNotification       0x805f4630 ntoskrnl.exe (\Driver\WMIxWDM)
```

Kernel callback to get notified whenever a filesystem registers, e.g. to attach to filesystems as filterdriver and control/intercept IRP packets



```
c:\Volatility-2.0.1>vol.py callbacks -f \forensics\malware-images\stuxnet.dmp
Volatile Systems Volatility Framework 2.0
Type                                 Callback   Owner
PsSetLoadImageNotifyRoutine          0xf89ead06 mrxcls.sys
PsSetCreateProcessNotifyRoutine      0xf888b6ae vmdebug.sys
IoRegisterFsRegistrationChange       0xf84be876 sr.sys
IoRegisterFsRegistrationChange       0xf8b369ec mrxnet.sys
KeBugCheckCallbackListHead           0xf83e65ed NDIS.sys (Ndis miniport)
KeBugCheckCallbackListHead           0x806d57ca hal.dll (ACPI 1.0 - APIC platform UP)
KeRegisterBugCheckReasonCallback     0xf8b6aac0 mssmbios.sys (SMBiosData)
KeRegisterBugCheckReasonCallback     0xf8b6aa78 mssmbios.sys (SMBiosRegistry)
KeRegisterBugCheckReasonCallback     0xf8b6aa30 mssmbios.sys (SMBiosDataACPI)
KeRegisterBugCheckReasonCallback     0xf82d93e2 VIDEOPRT.SYS (Videoprt)
KeRegisterBugCheckReasonCallback     0xf82fa006 USBPORT.SYS (USBPORT)
KeRegisterBugCheckReasonCallback     0xf82f9f66 USBPORT.SYS (USBPORT)
IoRegisterShutdownNotification       0xf890dc74 Cdfs.SYS (\FileSystem\Cdfs)
IoRegisterShutdownNotification       0xf7b488fa vmhgfs.sys (\FileSystem\vmhgfs)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\mnmdd)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\VgaSave)
IoRegisterShutdownNotification       0xf8bb25be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf8bb25be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf8bb25be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf8bb25be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf8bb25be Fs_Rec.SYS (\FileSystem\Fs_Rec)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\RDPCDD)
IoRegisterShutdownNotification       0xf82e565c VIDEOPRT.SYS (\Driver\vmx_svga)
IoRegisterShutdownNotification       0xf83d933d Mup.sys (\FileSystem\Mup)
IoRegisterShutdownNotification       0xf86aa73a MountMgr.sys (\Driver\MountMgr)
IoRegisterShutdownNotification       0xf853b2be ftdisk.sys (\Driver\Ftdisk)
IoRegisterShutdownNotification       0x805cc77c ntoskrnl.exe (\FileSystem\RAW)
IoRegisterShutdownNotification       0x805f4630 ntoskrnl.exe (\Driver\WMIxWDM)
```

Show device tree via _DEVICE_OBJECT parsing, e.g. to detect unknown file devices

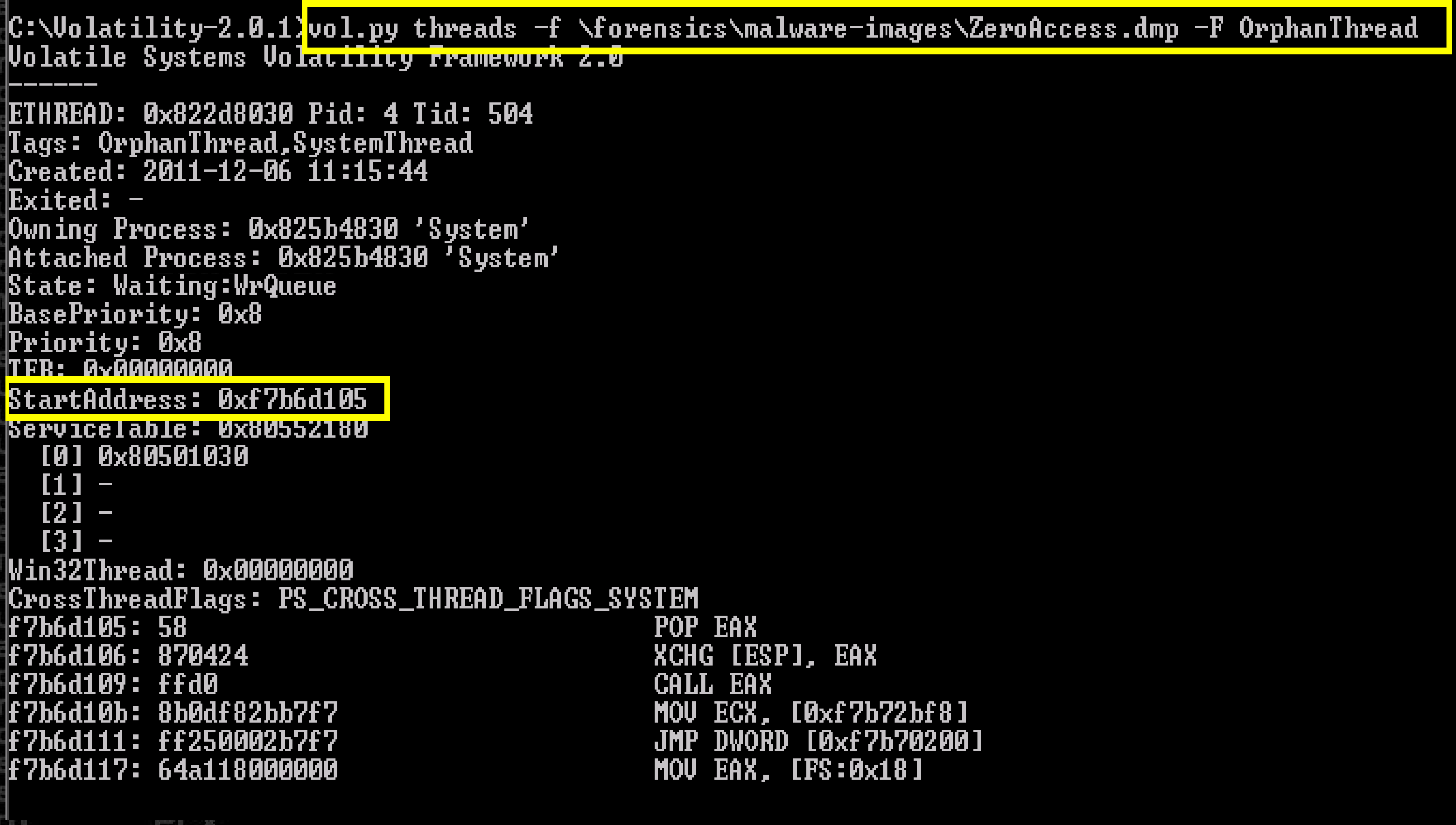


```
c:\Volatility-2.0.1>vol.py devicetree -f \forensics\malware-images\ZeroAccess.dmp
Volatile Systems Volatility Framework 2.0
DRV 0x0208e350 '\\FileSystem\\Srv'
---| DEV 0x8208e200 LanmanServer FILE_DEVICE_NETWORK
DRV 0x02099948 '\\Driver\\sysaudio'
---| DEV 0x82099780 sysaudio FILE_DEVICE_KS
DRV 0x02099f38 '\\Driver\\wdmaud'
---| DEV 0x82099e08 (unnamed) FILE_DEVICE_KS
DRV 0x0209e040 '\\FileSystem\\MRxDAV'
---| DEV 0x820a77e8 WebDavRedirector FILE_DEVICE_NETWORK_FILE_SYSTEM
DRV 0x020af160 '\\Driver\\LGTO_Sync'
---| DEV 0x82094e48 lgto_sync FILE_DEVICE_UNKNOWN
DRV 0x020c36f8 '\\FileSystem\\Fastfat'
---| DEV 0x820c34a8 FatCdrom FILE_DEVICE_CD_ROM_FILE_SYSTEM
---| DEV 0x820c35d0 Fat FILE_DEVICE_DISK_FILE_SYSTEM
------| ATT 0x820c4dd0 (unnamed) - '\\FileSystem\\sr' FILE_DEVICE_DISK_FILE_SYSTEM
DRV 0x020c6c90 '\\Driver\\win32dd'
---| DEV 0x822e0050 win32dd FILE_DEVICE_UNKNOWN
DRV 0x02193f38 '\\FileSystem\\Cdfs'
---| DEV 0x8246a040 Cdfs FILE_DEVICE_CD_ROM_FILE_SYSTEM
DRV 0x022b8f38 '\\Driver\\mnmdd'
---| DEV 0x82496810 Video2 FILE_DEVICE_VIDEO
DRV 0x022bcda0 '\\Driver\\NetBT'
---| DEV 0x822c6c38 NetBT_Tcpip_{0D60763D-3050-49BD-AB14-1796BE4E40A1} FILE_DEVICE_NETWORK
---| DEV 0x822d94e0 NetBt_Wins_Export FILE_DEVICE_NETWORK
---| DEV 0x82312d30 NetbiosSmb FILE_DEVICE_NETWORK
DRV 0x022bfb28 '\\Driver\\HTTP'
---| DEV 0x82050430 AppPool FILE_DEVICE_NETWORK
---| DEV 0x822f46d8 Filter FILE_DEVICE_NETWORK
---| DEV 0x820959e8 Control FILE_DEVICE_NETWORK
DRV 0x022c5880 '\\Driver\\Fips'
---| DEV 0x822c0f18 Fips FILE_DEVICE_FIPS
DRV 0x024596d0 '\\Driver\\00001079'
---| DEV 0x8243c040 ACPI#PNP0303#2&da1a3ff&0 FILE_DEVICE_UNKNOWN
DRV 0x0245b458 '\\FileSystem\\MRxSmb'
---| DEV 0x822b5f18 LanmanDatagramReceiver FILE_DEVICE_NETWORK_BROWSER
---| DEV 0x823a0270 LanmanRedirector FILE_DEVICE_NETWORK_FILE_SYSTEM
DRV 0x0245ec70 '\\Driver\\swenum'
---| DEV 0x820a1b88 KSENUM#00000009 FILE_DEVICE_UNKNOWN
------| ATT 0x820ae030 (unnamed) - '\\Driver\\kmixer' FILE_DEVICE_KS
---| DEV 0x82099ce8 KSENUM#00000002 FILE_DEVICE_UNKNOWN
------| ATT 0x82099780 sysaudio - '\\Driver\\sysaudio' FILE_DEVICE_KS
---| DEV 0x8209a400 KSENUM#00000001 FILE_DEVICE_UNKNOWN
------| ATT 0x82099e08 (unnamed) - '\\Driver\\wdmaud' FILE_DEVICE_KS
---| DEV 0x824461b8 (unnamed) FILE_DEVICE_BUS_EXTENDER
DRV 0x0233aca8 '\\Driver\\RasAcd'
```

Hunting for orphan threads

- **Drivers requiring delayed processing usually use a work item, using IoQueueWorkItem with a pointer to its callback routine**
- **When a system worker thread processes the queued item it gets removed and the callback gets invoked**
- **System worker threads run in the system process context (PID 4)**
- **Whenever work items have been processed or other system threads have been created this leaves traces on the callstack**
- **Modern rootkits often map themself into the non paged kernel pool, start this code as system thread and unload the original driver. These system threads without an existing driver entry can be detected with the Volatility “OrphanThread” function**

System Worker Threads parsing (SYSTEM process) to detect orphan threads

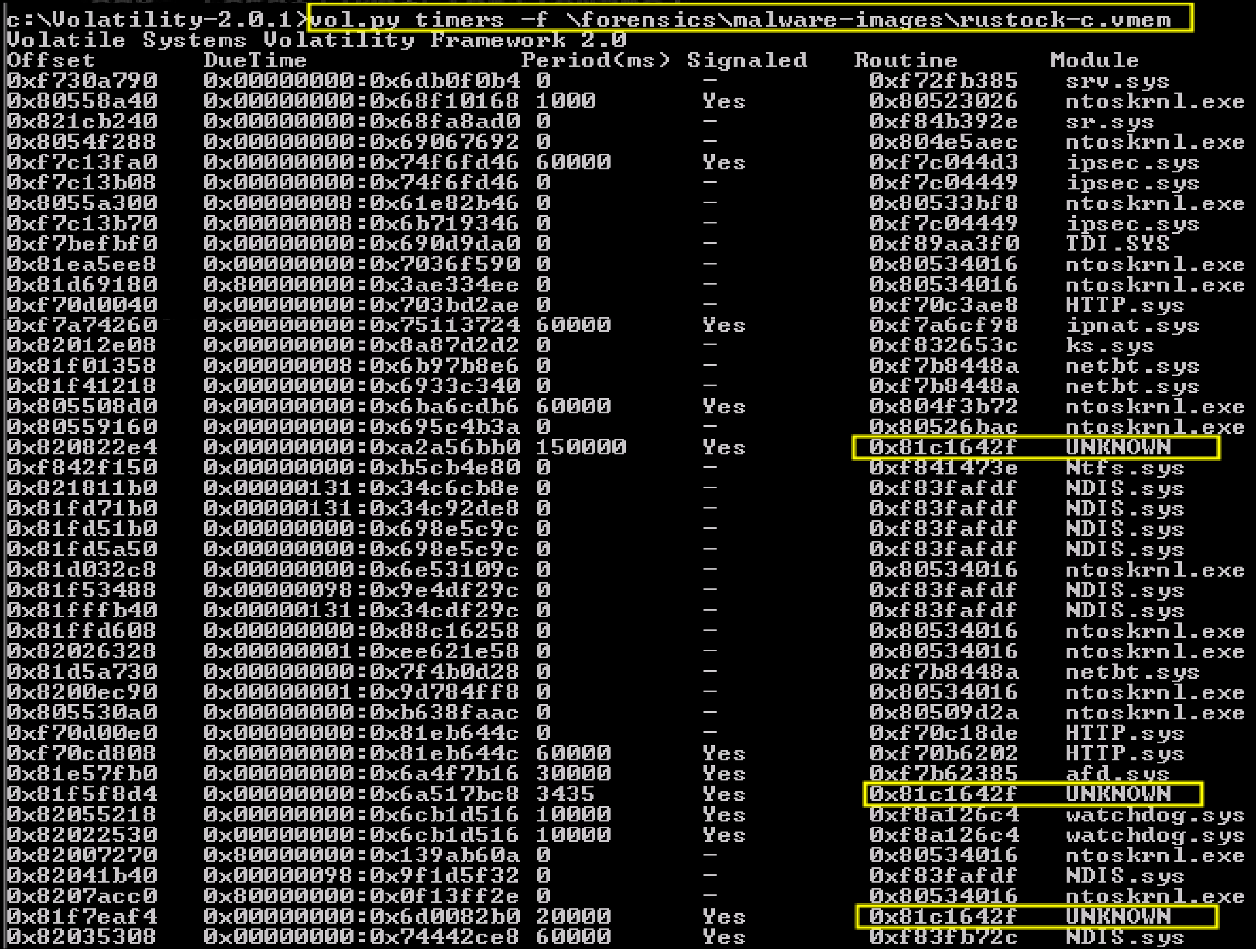


```
C:\Volatility-2.0.1>vol.py threads -f \forensics\malware-images\ZeroAccess.dmp -F OrphanThread
Volatile Systems Volatility Framework 2.0
------
ETHREAD: 0x822d8030 Pid: 4 Tid: 504
Tags: OrphanThread,SystemThread
Created: 2011-12-06 11:15:44
Exited: -
Owning Process: 0x825b4830 'System'
Attached Process: 0x825b4830 'System'
State: Waiting:WrQueue
BasePriority: 0x8
Priority: 0x8
TEB: 0x00000000
StartAddress: 0xf7b6d105
ServiceTable: 0x80552180
  [0] 0x80501030
  [1] -
  [2] -
  [3] -
Win32Thread: 0x00000000
CrossThreadFlags: PS_CROSS_THREAD_FLAGS_SYSTEM
f7b6d105: 58                                POP EAX
f7b6d106: 870424                            XCHG [ESP], EAX
f7b6d109: ffd0                              CALL EAX
f7b6d10b: 8b0df82bb7f7                      MOV ECX, [0xf7b72bf8]
f7b6d111: ff250002b7f7                      JMP DWORD [0xf7b70200]
f7b6d117: 64a118000000                      MOV EAX, [FS:0x18]
```

Hunting for suspicious functions in kernel timers

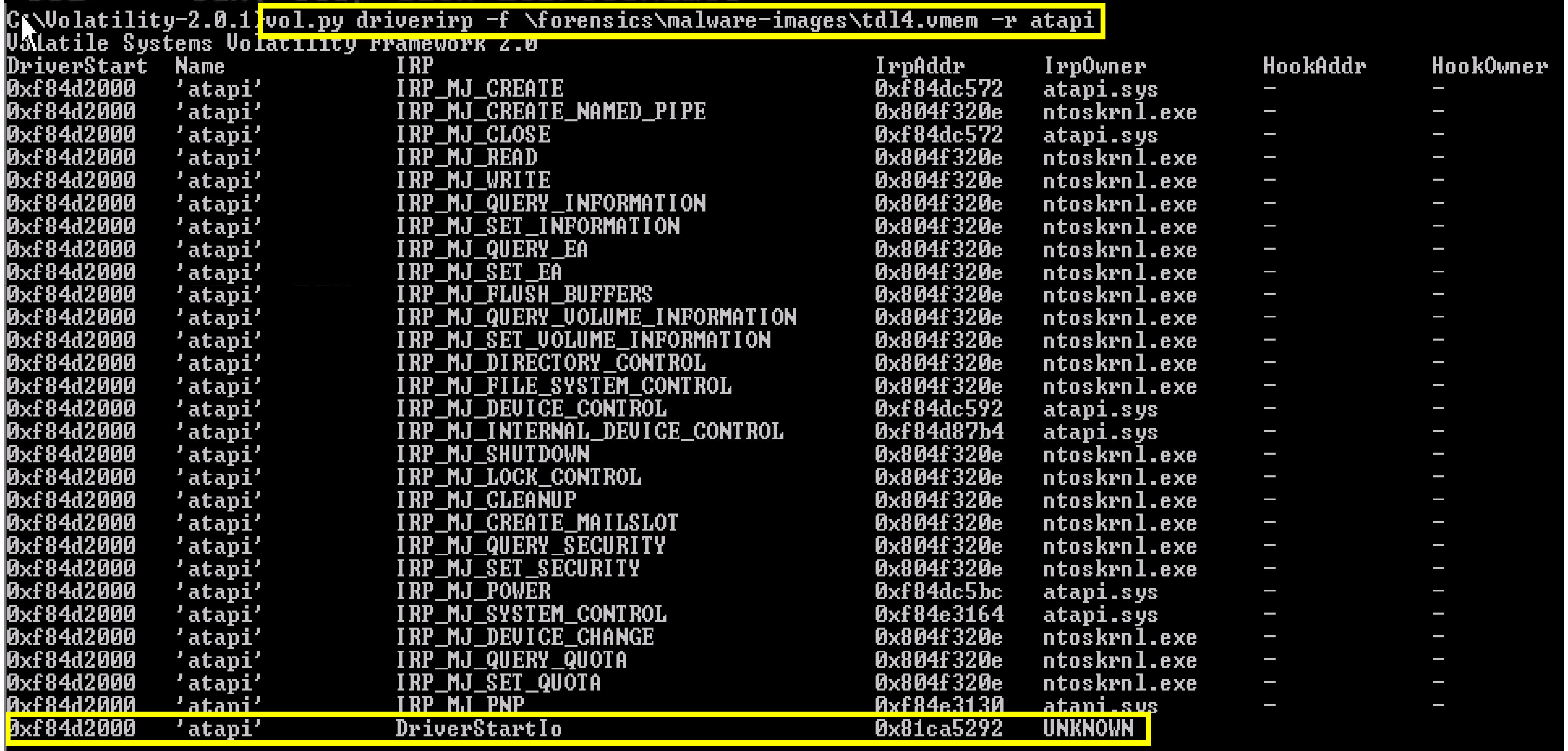
- Kernel timer DPCs are being used to schedule an execution of a function to a particular time
- Some rootkits install timers, e.g. to start C&C communication after an elapsed time or to check if the system is currently being traced or debugged

Show installed kernel timer routines and its owners via _KTIMER parsing



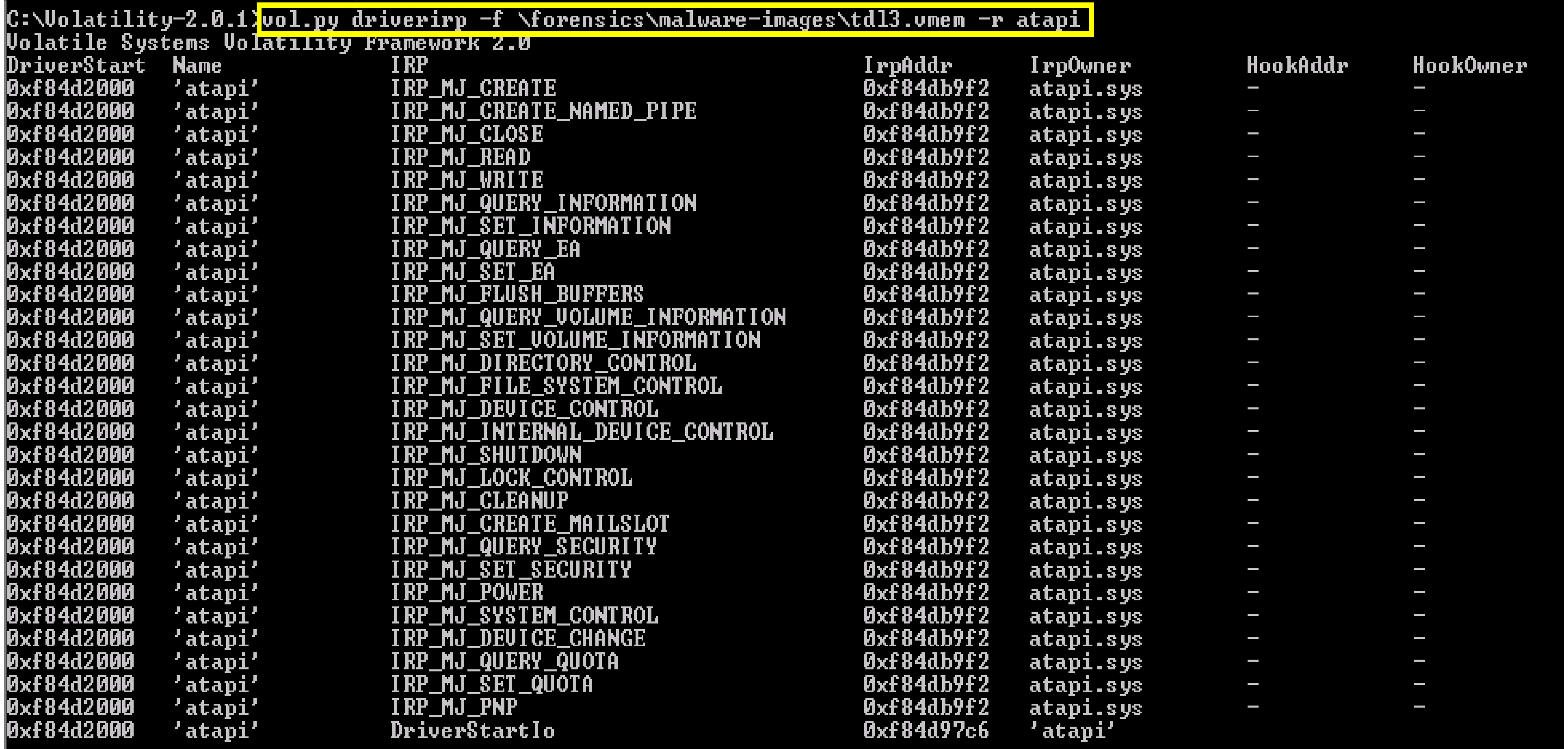
```
c:\Volatility-2.0.1>vol.py timers -f \forensics\malware-images\rustock-c.vmem
Volatile Systems Volatility Framework 2.0
Offset      DueTime                  Period(ms) Signaled   Routine     Module
0xf730a790  0x00000000:0x6db0f0b4 0             -           0xf72fb385  srv.sys
0x80558a40  0x00000000:0x68f10168 1000          Yes         0x80523026  ntoskrnl.exe
0x821cb240  0x00000000:0x68fa8ad0 0             -           0xf84b392e  sr.sys
0x8054f288  0x00000000:0x69067692 0             -           0x804e5aec  ntoskrnl.exe
0xf7c13fa0  0x00000000:0x74f6fd46 60000         Yes         0xf7c044d3  ipsec.sys
0xf7c13b08  0x00000000:0x74f6fd46 0             -           0xf7c04449  ipsec.sys
0x8055a300  0x00000008:0x61e82b46 0             -           0x80533bf8  ntoskrnl.exe
0xf7c13b70  0x00000008:0x6b719346 0             -           0xf7c04449  ipsec.sys
0xf7befbf0  0x00000000:0x690d9da0 0             -           0xf89aa3f0  TDI.SYS
0x81ea5ee8  0x00000000:0x7036f590 0             -           0x80534016  ntoskrnl.exe
0x81d69180  0x80000000:0x3ae334ee 0             -           0x80534016  ntoskrnl.exe
0xf70d0040  0x00000000:0x703bd2ae 0             -           0xf70c3ae8  HTTP.sys
0xf7a74260  0x00000000:0x75113724 60000         Yes         0xf7a6cf98  ipnat.sys
0x82012e08  0x00000000:0x8a87d2d2 0             -           0xf832653c  ks.sys
0x81f01358  0x00000008:0x6b97b8e6 0             -           0xf7b8448a  netbt.sys
0x81f41218  0x00000000:0x6933c340 0             -           0xf7b8448a  netbt.sys
0x805508d0  0x00000000:0x6ba6cdb6 60000         Yes         0x804f3b72  ntoskrnl.exe
0x80559160  0x00000000:0x695c4b3a 0             -           0x80526bac  ntoskrnl.exe
0x820822e4  0x00000000:0xa2a56bb0 150000        Yes         0x81c1642f  UNKNOWN
0xf842f150  0x00000000:0xb5cb4e80 0             -           0xf841473e  Ntfs.sys
0x821811b0  0x00000131:0x34c6cb8e 0             -           0xf83fafdf  NDIS.sys
0x81fd71b0  0x00000131:0x34c92de8 0             -           0xf83fafdf  NDIS.sys
0x81fd51b0  0x00000000:0x698e5c9c 0             -           0xf83fafdf  NDIS.sys
0x81fd5a50  0x00000000:0x698e5c9c 0             -           0xf83fafdf  NDIS.sys
0x81d032c8  0x00000000:0x6e53109c 0             -           0x80534016  ntoskrnl.exe
0x81f53488  0x00000098:0x9e4df29c 0             -           0xf83fafdf  NDIS.sys
0x81fffb40  0x00000131:0x34cdf29c 0             -           0xf83fafdf  NDIS.sys
0x81ffd608  0x00000000:0x88c16258 0             -           0x80534016  ntoskrnl.exe
0x82026328  0x00000001:0xee621e58 0             -           0x80534016  ntoskrnl.exe
0x81d5a730  0x00000000:0x7f4b0d28 0             -           0xf7b8448a  netbt.sys
0x8200ec90  0x00000001:0x9d784ff8 0             -           0x80534016  ntoskrnl.exe
0x805530a0  0x00000000:0xb638faac 0             -           0x80509d2a  ntoskrnl.exe
0xf70d00e0  0x00000000:0x81eb644c 0             -           0xf70c18de  HTTP.sys
0xf70cd808  0x00000000:0x81eb644c 60000         Yes         0xf70b6202  HTTP.sys
0x81e57fb0  0x00000000:0x6a4f7b16 30000         Yes         0xf7b62385  afd.sys
0x81f5f8d4  0x00000000:0x6a517bc8 3435          Yes         0x81c1642f  UNKNOWN
0x82055218  0x00000000:0x6cb1d516 10000         Yes         0xf8a126c4  watchdog.sys
0x82022530  0x00000000:0x6cb1d516 10000         Yes         0xf8a126c4  watchdog.sys
0x82007270  0x80000000:0x139ab60a 0             -           0x80534016  ntoskrnl.exe
0x82041b40  0x00000098:0x9f1d5f32 0             -           0xf83fafdf  NDIS.sys
0x8207acc0  0x80000000:0x0f13ff2e 0             -           0x80534016  ntoskrnl.exe
0x81f7eaf4  0x00000000:0x6d0082b0 20000         Yes         0x81c1642f  UNKNOWN
0x82035308  0x00000000:0x74442ce8 60000         Yes         0xf83fb72c  NDIS.sys
```

Show driver IRPs to detect manipulated dispatcher functions (Example: DriverStartIo hook)



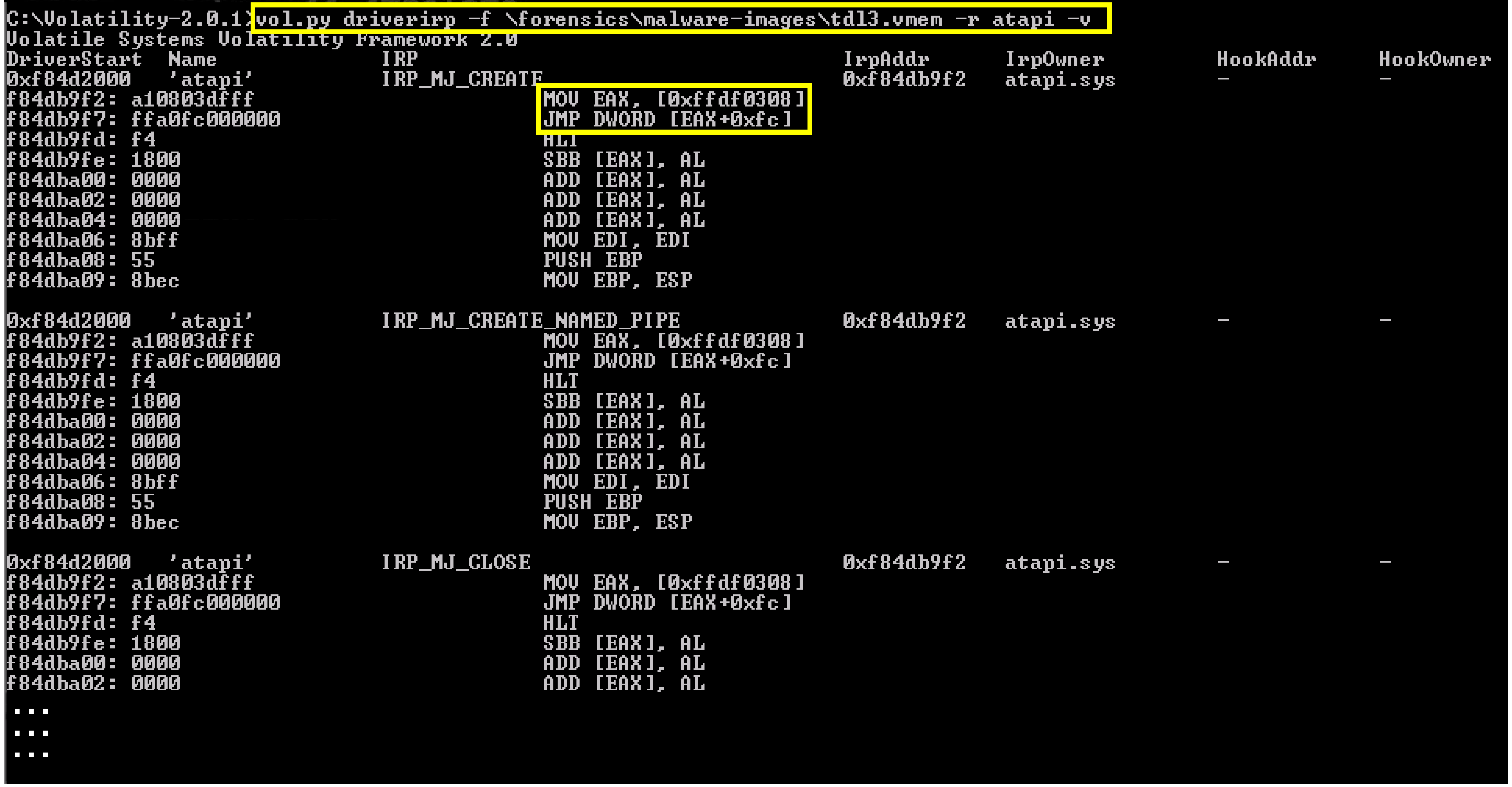
```
C:\Volatility-2.0.1>vol.py driverirp -f \forensics\malware-images\tdl4.vmem -r atapi
Volatile Systems Volatility Framework 2.0
DriverStart   Name        IRP                                IrpAddr      IrpOwner         HookAddr     HookOwner
0xf84d2000    'atapi'     IRP_MJ_CREATE                      0xf84dc572   atapi.sys        -            -
0xf84d2000    'atapi'     IRP_MJ_CREATE_NAMED_PIPE           0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_CLOSE                       0xf84dc572   atapi.sys        -            -
0xf84d2000    'atapi'     IRP_MJ_READ                        0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_WRITE                       0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_QUERY_INFORMATION           0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_SET_INFORMATION             0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_QUERY_EA                    0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_SET_EA                      0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_FLUSH_BUFFERS               0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_QUERY_VOLUME_INFORMATION    0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_SET_VOLUME_INFORMATION      0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_DIRECTORY_CONTROL           0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_FILE_SYSTEM_CONTROL         0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_DEVICE_CONTROL              0xf84dc592   atapi.sys        -            -
0xf84d2000    'atapi'     IRP_MJ_INTERNAL_DEVICE_CONTROL     0xf84d87b4   atapi.sys        -            -
0xf84d2000    'atapi'     IRP_MJ_SHUTDOWN                    0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_LOCK_CONTROL                0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_CLEANUP                     0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_CREATE_MAILSLOT             0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_QUERY_SECURITY              0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_SET_SECURITY                0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_POWER                       0xf84dc5bc   atapi.sys        -            -
0xf84d2000    'atapi'     IRP_MJ_SYSTEM_CONTROL              0xf84e3164   atapi.sys        -            -
0xf84d2000    'atapi'     IRP_MJ_DEVICE_CHANGE               0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_QUERY_QUOTA                 0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_SET_QUOTA                   0x804f320e   ntoskrnl.exe     -            -
0xf84d2000    'atapi'     IRP_MJ_PNP                         0xf84e3130   atapi.sys        -            -
0xf84d2000    'atapi'     DriverStartIo                      0x81ca5292   UNKNOWN
```

Show driver IRPs to detect manipulated dispatcher functions
But where's the hook?



```
C:\Volatility-2.0.1>vol.py driverirp -f \forensics\malware-images\tdl3.vmem -r atapi
Volatile Systems Volatility Framework 2.0
DriverStart  Name          IRP                                 IrpAddr     IrpOwner        HookAddr    HookOwner
0xf84d2000   'atapi'       IRP_MJ_CREATE                       0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_CREATE_NAMED_PIPE            0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_CLOSE                        0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_READ                         0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_WRITE                        0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_QUERY_INFORMATION            0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_SET_INFORMATION              0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_QUERY_EA                     0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_SET_EA                       0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_FLUSH_BUFFERS                0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_QUERY_VOLUME_INFORMATION     0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_SET_VOLUME_INFORMATION       0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_DIRECTORY_CONTROL            0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_FILE_SYSTEM_CONTROL          0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_DEVICE_CONTROL               0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_INTERNAL_DEVICE_CONTROL      0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_SHUTDOWN                     0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_LOCK_CONTROL                 0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_CLEANUP                      0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_CREATE_MAILSLOT              0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_QUERY_SECURITY               0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_SET_SECURITY                 0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_POWER                        0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_SYSTEM_CONTROL               0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_DEVICE_CHANGE                0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_QUERY_QUOTA                  0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_SET_QUOTA                    0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       IRP_MJ_PNP                          0xf84db9f2  atapi.sys       -           -
0xf84d2000   'atapi'       DriverStartIo                       0xf84d97c6  'atapi'
```

Show driver IRPs including disassembly using the driverirp function in combination with the –v parameter. This shows the patched code and jump to the _KUSER_SHARED_DATA area



```
C:\Volatility-2.0.1>vol.py driverirp -f \forensics\malware-images\tdl3.vmem -r atapi -v
Volatile Systems Volatility Framework 2.0
DriverStart  Name            IRP                             IrpAddr     IrpOwner       HookAddr    HookOwner
0xf84d2000   'atapi'         IRP_MJ_CREATE                   0xf84db9f2  atapi.sys      -           -
f84db9f2: a10803dfff                  MOV EAX, [0xffdf0308]
f84db9f7: ffa0fc000000                JMP DWORD [EAX+0xfc]
f84db9fd: f4                          HLT
f84db9fe: 1800                        SBB [EAX], AL
f84dba00: 0000                        ADD [EAX], AL
f84dba02: 0000                        ADD [EAX], AL
f84dba04: 0000                        ADD [EAX], AL
f84dba06: 8bff                        MOV EDI, EDI
f84dba08: 55                          PUSH EBP
f84dba09: 8bec                        MOV EBP, ESP

0xf84d2000   'atapi'         IRP_MJ_CREATE_NAMED_PIPE        0xf84db9f2  atapi.sys      -           -
f84db9f2: a10803dfff                  MOV EAX, [0xffdf0308]
f84db9f7: ffa0fc000000                JMP DWORD [EAX+0xfc]
f84db9fd: f4                          HLT
f84db9fe: 1800                        SBB [EAX], AL
f84dba00: 0000                        ADD [EAX], AL
f84dba02: 0000                        ADD [EAX], AL
f84dba04: 0000                        ADD [EAX], AL
f84dba06: 8bff                        MOV EDI, EDI
f84dba08: 55                          PUSH EBP
f84dba09: 8bec                        MOV EBP, ESP

0xf84d2000   'atapi'         IRP_MJ_CLOSE                    0xf84db9f2  atapi.sys      -           -
f84db9f2: a10803dfff                  MOV EAX, [0xffdf0308]
f84db9f7: ffa0fc000000                JMP DWORD [EAX+0xfc]
f84db9fd: f4                          HLT
f84db9fe: 1800                        SBB [EAX], AL
f84dba00: 0000                        ADD [EAX], AL
f84dba02: 0000                        ADD [EAX], AL
...
...
...
```

Conclusion

- Volatility is a very powerful tool, which is able to detect even the most advanced rootkits if it's being used properly.
- The analyst should have good windows knowledge to combine the different functions in a smart way and draw the right conclusions
- False positives could be caused by security software like HIPS, AV or personal firewalls, as they act in a very similar way malware does. The only way to be 100% sure if the code is malicious or not the investigator has to disassemble the dumped code resp. alerted functions

```
push    2
call    sub_672B3730
add     esp, 0Ch
test    eax, eax
jnz     short loc_672B5428
lea     edx, [esp+110h+LibFileName]
push    edx
call    sub_672B35F0
mov     edi, off_672CA058
or      ecx, 0FFFFFFFFh
xor     eax, eax
lea     edx, [esp+114h+LibFileName]
repne scasb
not     ecx
sub     edi, ecx
mov     esi, edi
mov     ebx, ecx
cmp     eax, 7Eh
jnz     loc_672B5455
lea     ecx, [esp+110h+LibFileName]
push    104h
push    ecx
push    2
call    sub_672B3730
add     esp, 0Ch
test    eax, eax
jnz     short loc_672B5428
lea     edx, [esp+110h+LibFileName]
push    edx
call    sub_672B35F0
mov     edi, off_672CA058
or      ecx, 0FFFFFFFFh
xor     eax, eax
lea     edx, [esp+114h+LibFileName]
repne scasb
not     ecx
sub     edi, ecx
mov     esi, edi
mov     ebx, ecx
```

Questions?