RESEARCH

A Deep Dive Into ALPHV/BlackCat Ransomware

Executive summary

ALPHV/BlackCat is the first widely known ransomware written in Rust. The malware must run with an access token consisting of a 32-byte value (–access-token parameter), and other parameters can be specified. The ransomware comes with an encrypted configuration that contains a list of services/processes to be stopped, a list of whitelisted directories/files/file extensions, and a list of stolen credentials from the victim environment. It deletes all Volume Shadow Copies, performs privilege escalation using the CMSTPLUA COM interface, and enables “remote to local” and “remote to remote” symbolic links on the victim’s machine.

The files are encrypted using the AES algorithm, with the AES key being encrypted using the RSA public key contained in the configuration. The extension of the encrypted files is changed to uhwuvzu by the malware.

Analysis and findings

SHA256: 847fb7609f53ed334d5affbb07256c21cb5e6f68b1cc14004f5502d714d2a456

The malware can run with one of the following parameters:

SecurityScorecard Close

```
[OPTIONS] [SUBCOMMAND]

OPTIONS:
        --access-token <ACCESS_TOKEN>                Access Token
        --bypass <BYPASS>...
        --child                                      Run as child process
        --drag-and-drop                              Invoked with drag and drop
        --drop-drag-and-drop-target                  Drop drag and drop target batch file
        --extra-verbose                              Log more to console
    -h, --help                                       Print help information
        --log-file <LOG_FILE>                        Enable logging to specified file
        --no-net                                     Do not discover network shares on Windows
        --no-prop                                    Do not self propagate(worm) on Windows
        --no-prop-servers <NO_PROP_SERVERS>...       Do not propagate to defined servers
        --no-vm-kill                                 Do not stop VMs on ESXi
        --no-vm-kill-names <NO_VM_KILL_NAMES>...     Do not stop defined VMs on ESXi
        --no-vm-snapshot-kill                        Do not wipe VMs snapshots on ESXi
        --no-wall                                    Do not update desktop wallpaper on Windows
    -p, --paths <PATHS>...                           Only process files inside defined paths
        --propagated                                 Run as propagated process
        --ui                                         Show user interface
    -v, --verbose                                    Log to console
```

Figure 1
Whether the ransomware is running with no parameters or with an invalid access token, an error message is displayed:

```
Administrator: Administrator Command Prompt

C:\Users\    \Desktop>malware.exe
Invalid access token.

C:\Users\    \Desktop>malware.exe --access-token 123456
Invalid config.
```

Figure 2
By performing the dynamic analysis, we've found that the access token must be a 32-byte value that is not unique.

The binary registers a new top-level exception handler via a function call to SetUnhandledExceptionFilter:

```
00C51241   C7 04 24 A0 42 68 01   mov dword ptr ss:[esp],malware.16B42A0
EIP 00C51248   FF 15 4C 78 91 01   call dword ptr ds:[<&SetUnhandledExceptionFilter>]
dword ptr [0191784C <malware.&SetUnhandledExceptionFilter>]=<kernel32.SetUnhandledExceptionFilter>

.text:00C51248 malware.exe:$1248 #648

Default (stdcall)   5   Unlock
2: [esp+4] 00000002
3: [esp+8] 00000000
4: [esp+C] 00000000
005FFD00 016842A0 malware.016842A0
```

Figure 3
The AddVectoredExceptionHandler API is utilized to register a vectored exception handler:

```
01641139   68 70 48 65 01   push malware.1654870
0164113E   6A 00            push 0
EIP 01641140   E8 93 C9 06 00   call <malware.RtlAddVectoredExceptionHandler>
x87SW_SF 0   x87SW_P 0   x87SW_U 0
Default (stdcall)   5   Unlock
```

Figure 5
BlackCat opens the “SOFTWARE\Microsoft\Cryptography” registry key by calling the RegOpenKeyExW routine (0x80000002 = **HKEY_LOCAL_MACHINE**, 0x20019 = KEY_READ):

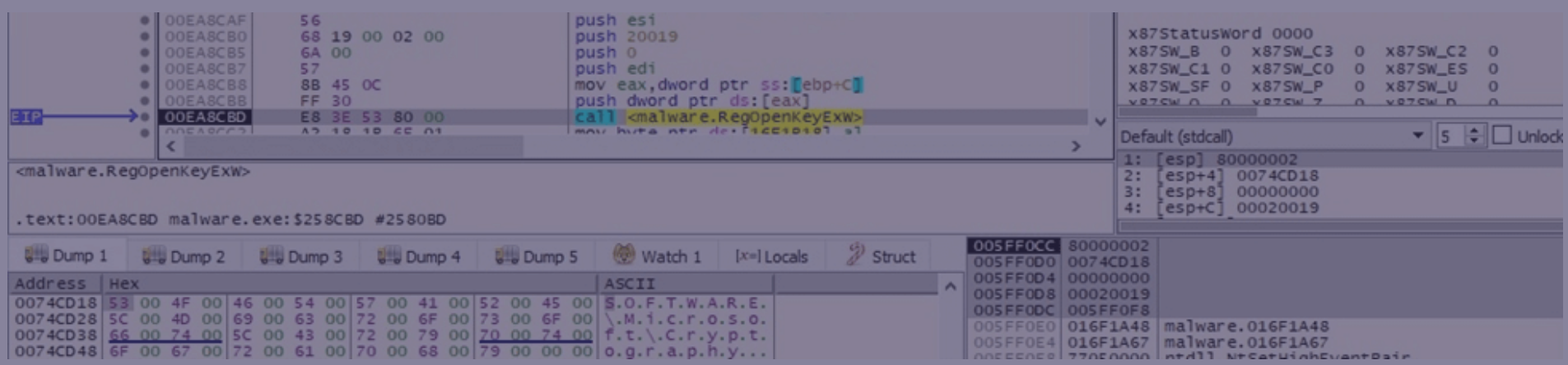


Figure 6
The binary extracts the MachineGUID value from the registry:

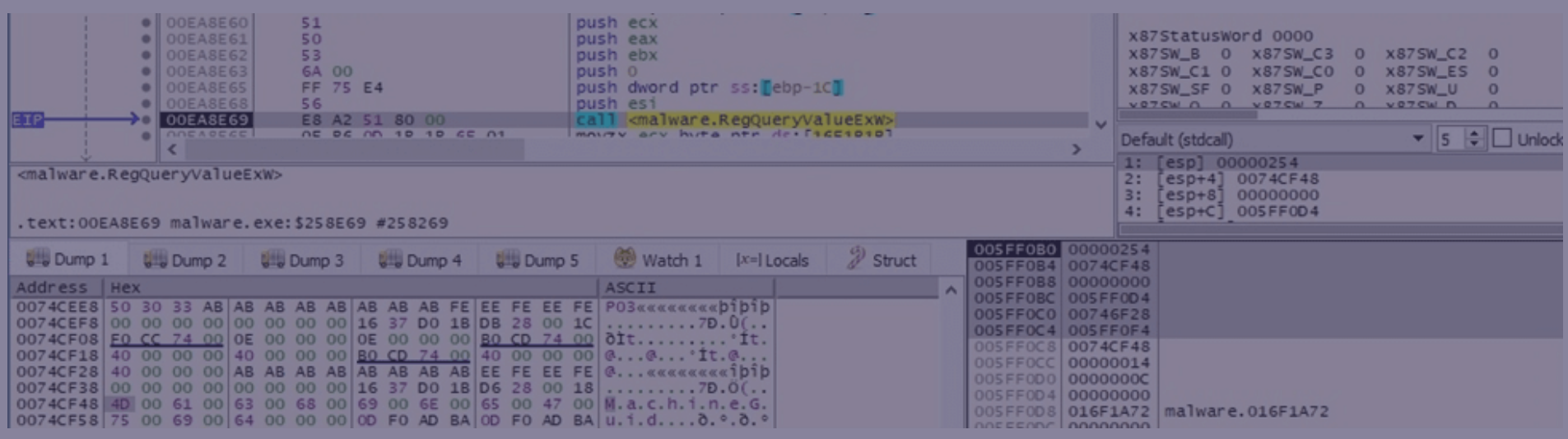
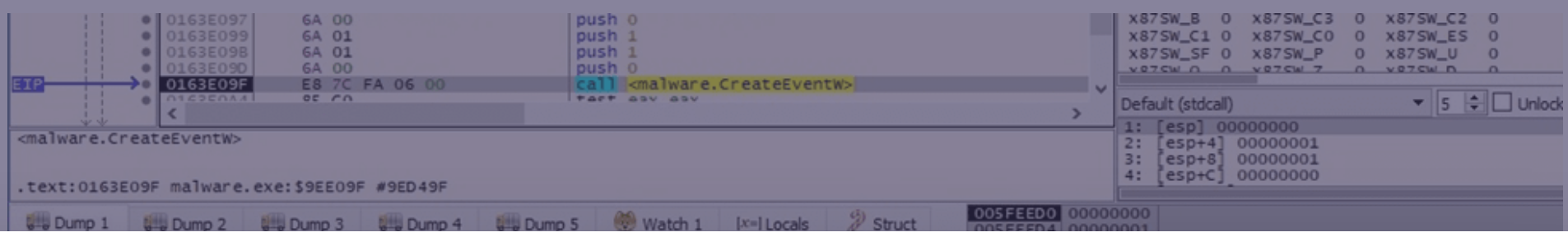


Figure 7
The malicious process searches for cmd.exe in the current directory and then in the System32 directory via a function call to CreateFileW (0x7 = **FILE_SHARE_DELETE | FILE_SHARE_WRITE | FILE_SHARE_READ**, 0x3 = **OPEN_EXISTING**, 0x2000000 = **FILE_FLAG_BACKUP_SEMANTICS**):



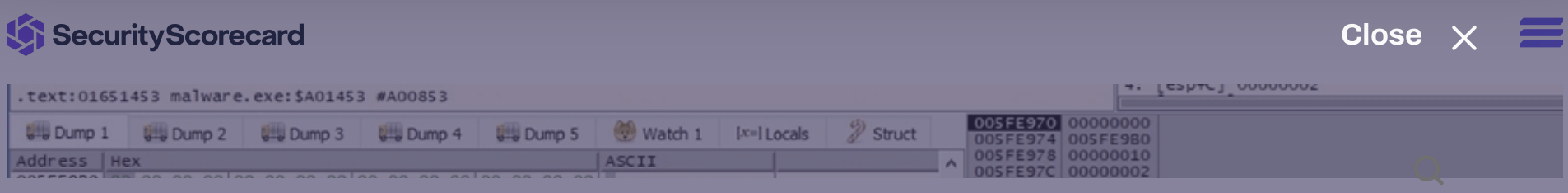


Figure 9
A named pipe whose name contains the current process ID and random bytes generated above is created using CreateNamedPipeW (0x40080001 = **FILE_FLAG_OVERLAPPED | FILE_FLAG_FIRST_PIPE_INSTANCE | PIPE_ACCESS_INBOUND**, 0x8 = **PIPE_REJECT_REMOTE_CLIENTS**):

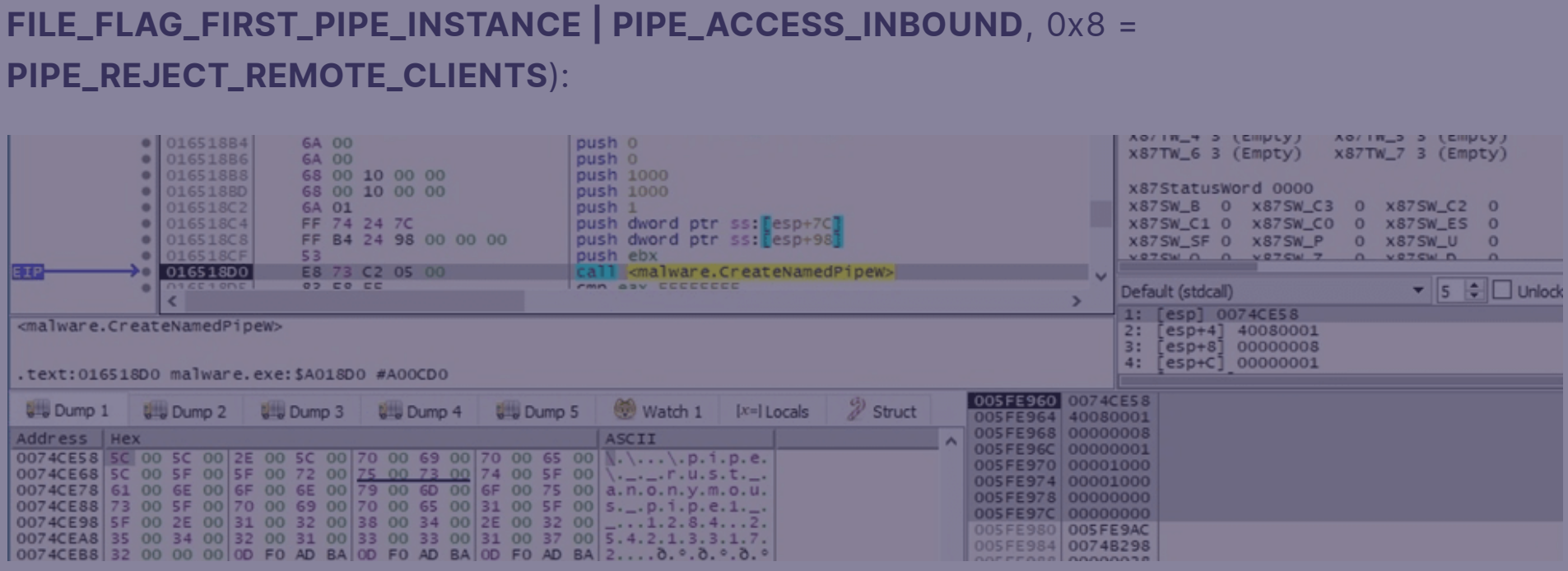


Figure 10
The process opens the named pipe for writing using the CreateFileW routine (0x40000000 = **GENERIC_WRITE**, 0x3 = **OPEN_EXISTING**):

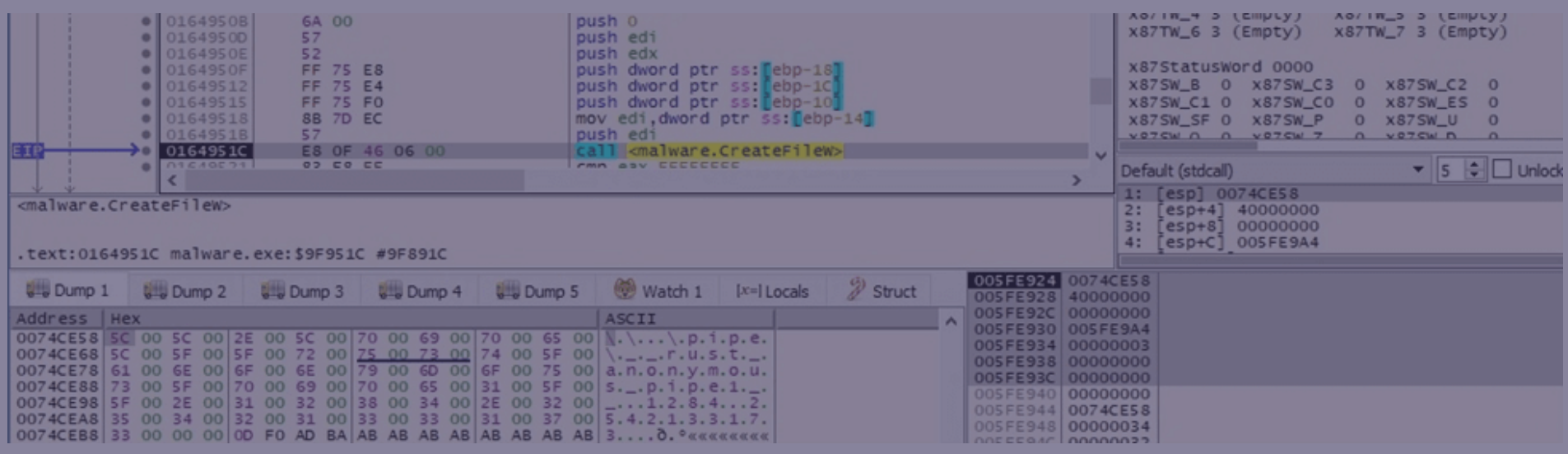


Figure 11
The ransomware creates a read and a write named pipe, respectively.

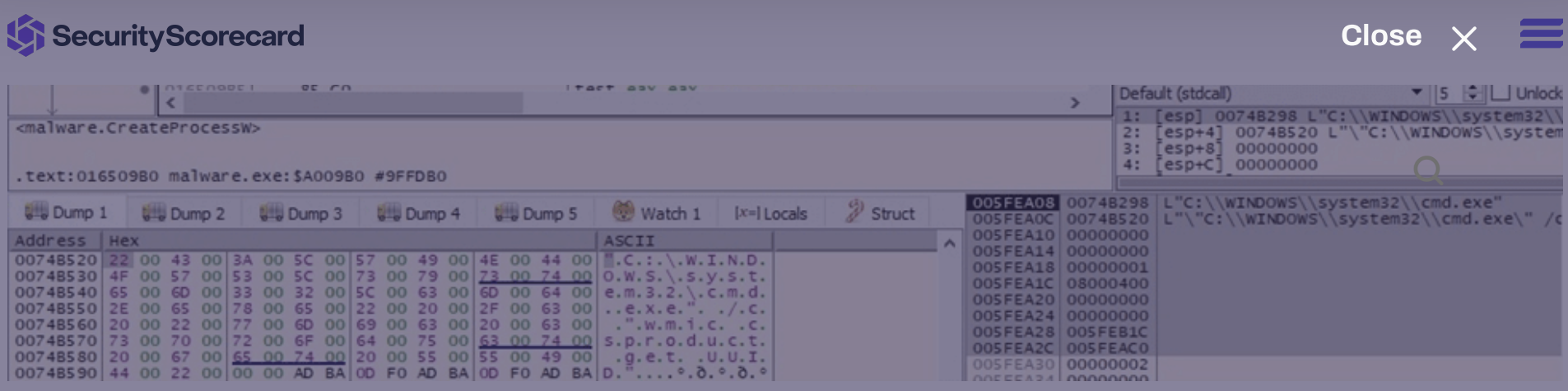


Figure 12
The CreateEventW API is utilized to create two unnamed event objects:

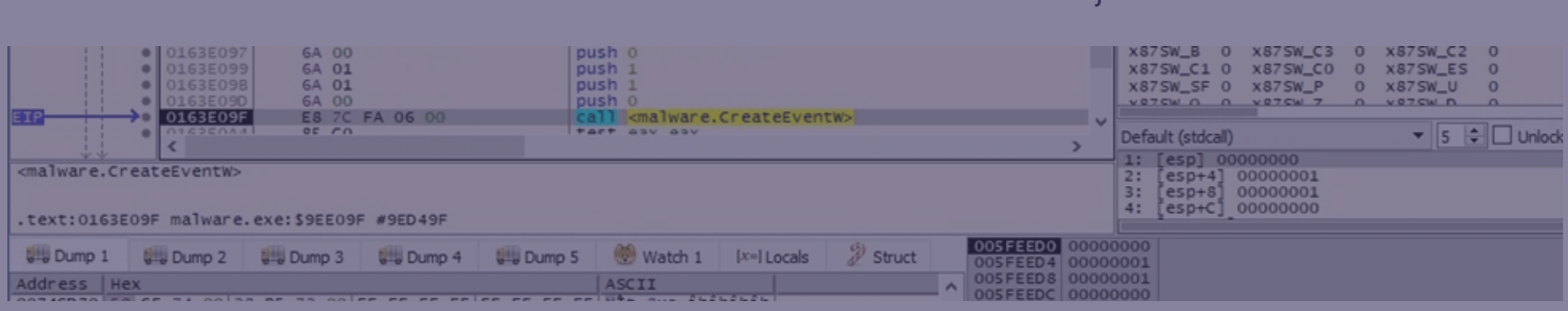


Figure 13
The binary waits until the event objects are in the signaled state by calling WaitForMultipleObjects:

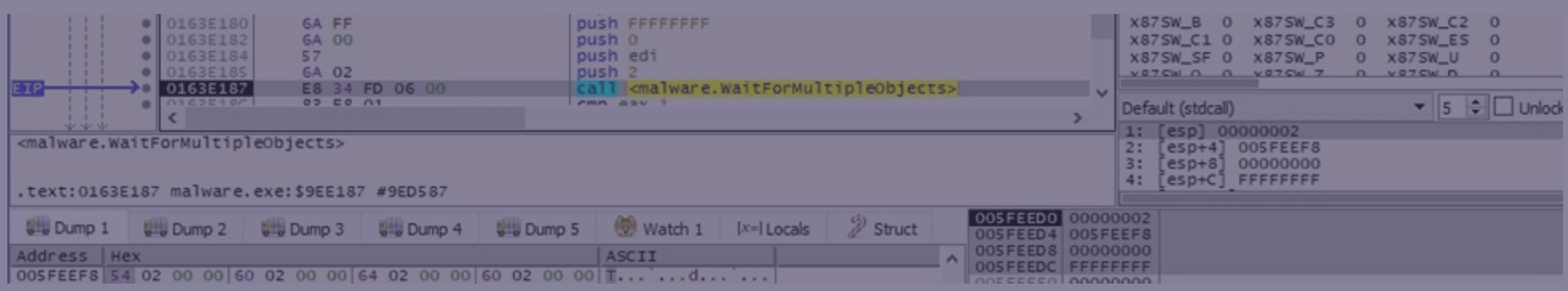
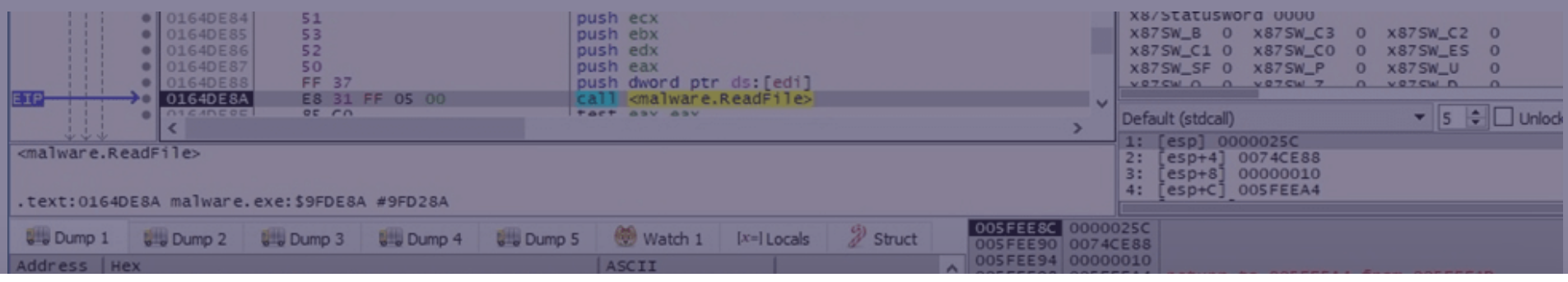


Figure 14
The output of the above process is read from the named pipe using the ReadFile routine:



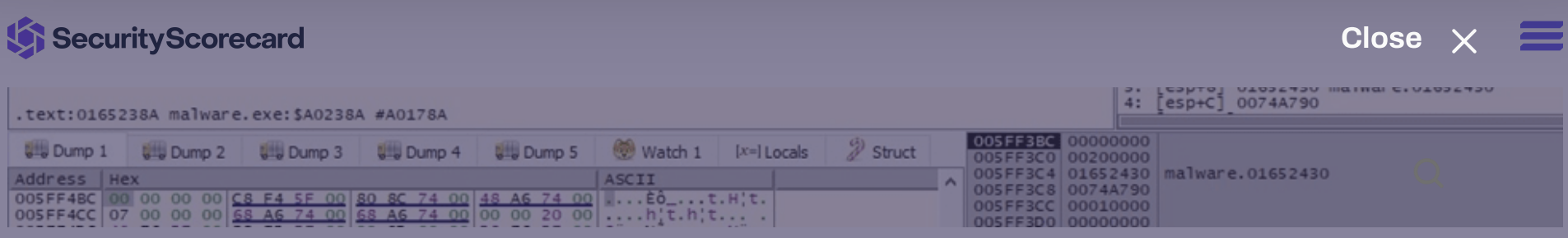


Figure 16

The content of the ransom note and the text that will appear on the Desktop Wallpaper are decrypted by the ransomware:

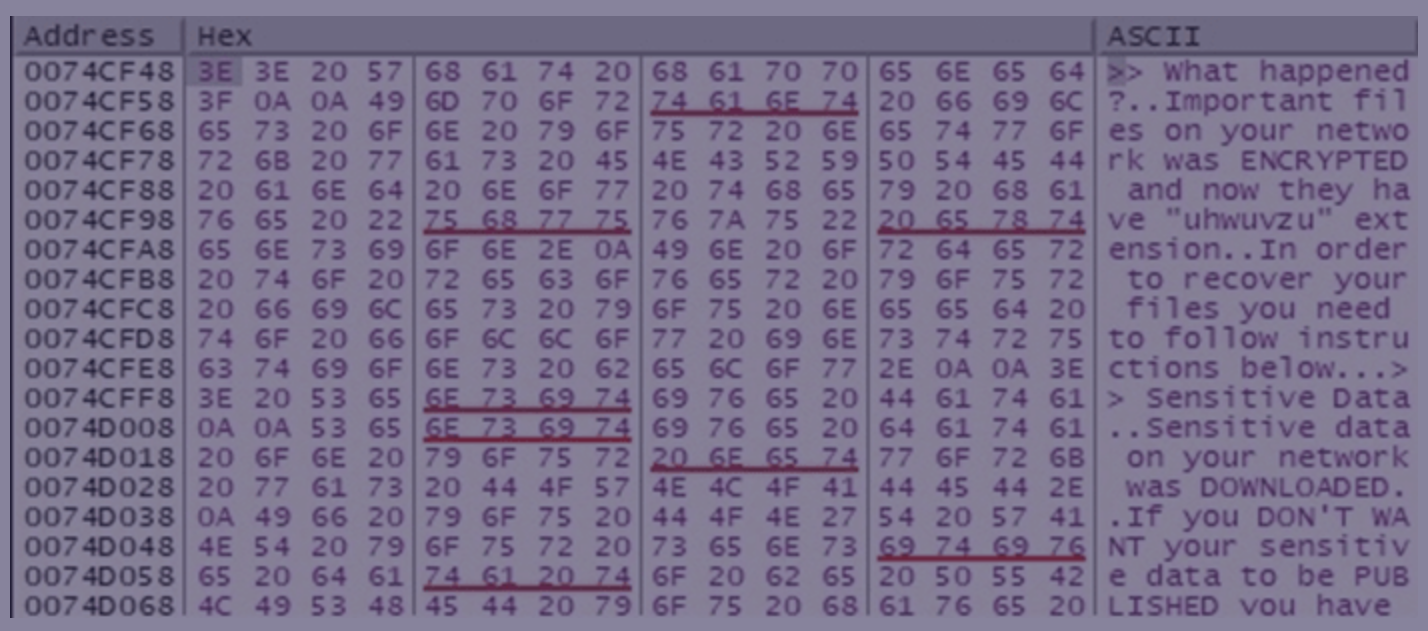


Figure 17

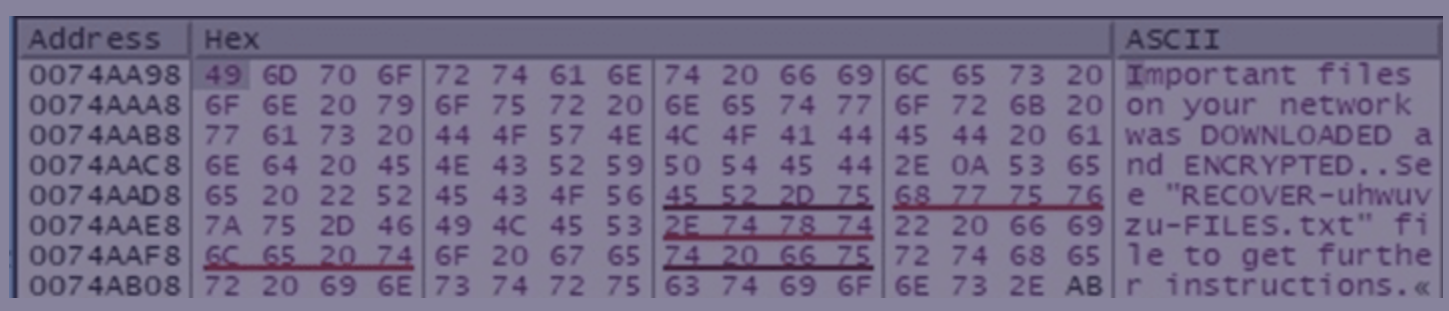
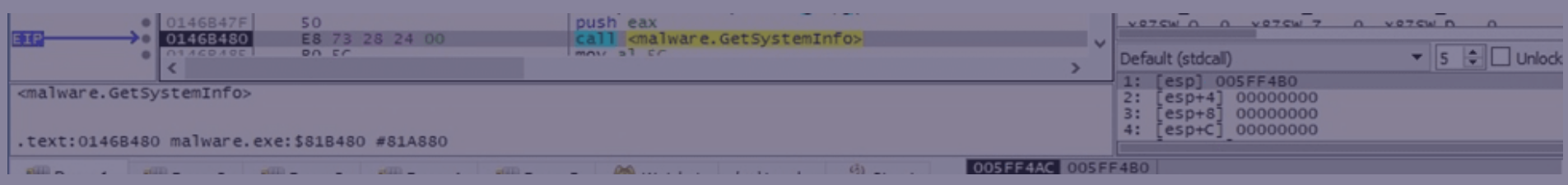


Figure 18

The malicious binary obtains information about the current system via a function call to GetSystemInfo:



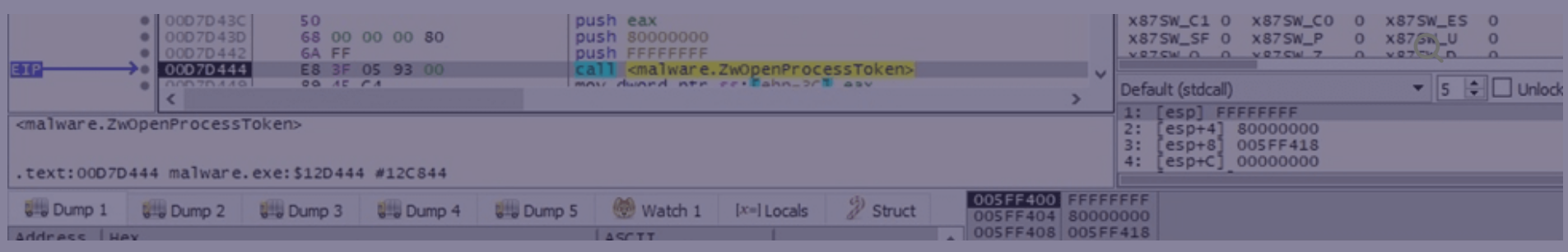


Figure 21
BlackCat extracts a TOKEN_GROUPS structure containing the group accounts associated with the above token using the NtQueryInformationToken function (0x2 = **TokenGroups**):

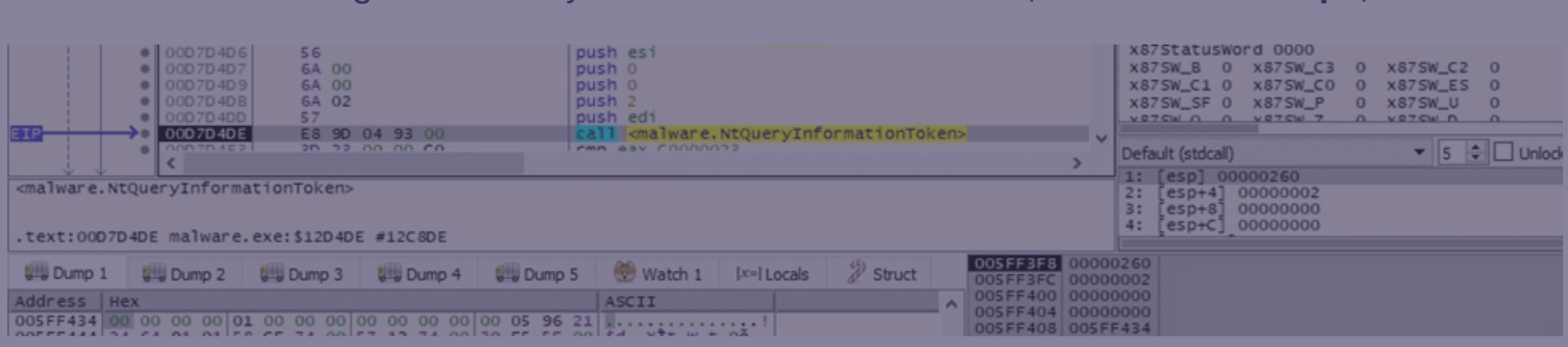


Figure 22
The OpenProcess API is utilized to open a local process object (0x438 = **PROCESS_QUERY_INFORMATION | PROCESS_VM_WRITE | PROCESS_VM_READ | PROCESS_VM_OPERATION**):

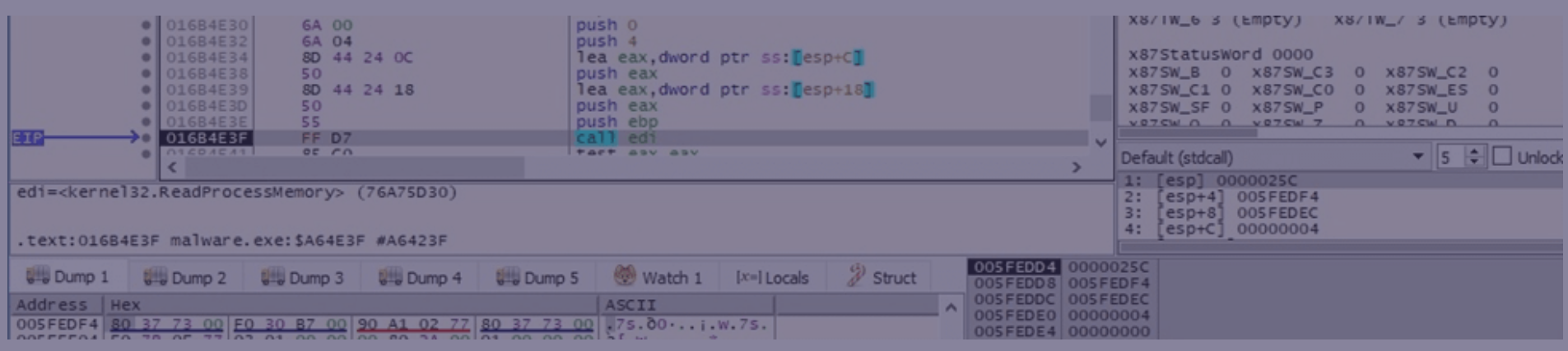
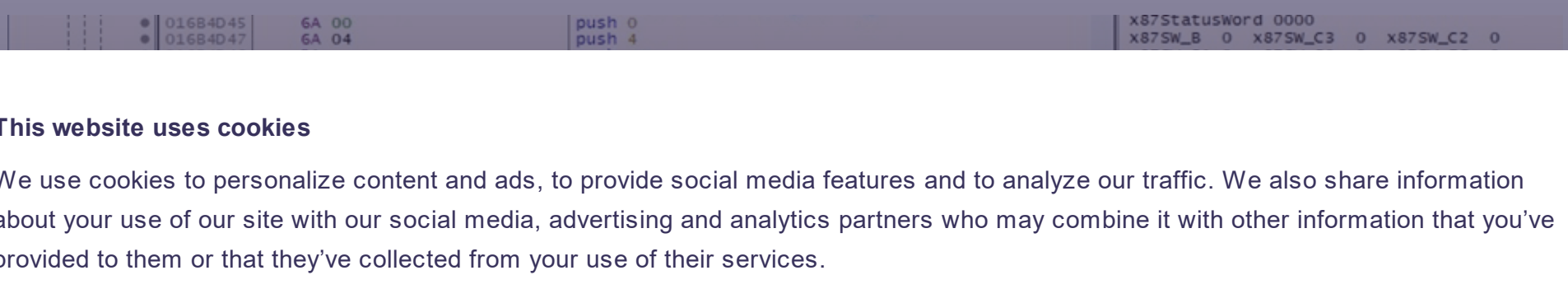


Figure 23
The malicious binary retrieves a pointer to a PEB structure using the ZwQueryInformationProcess routine (0x0 = **ProcessBasicInformation**):



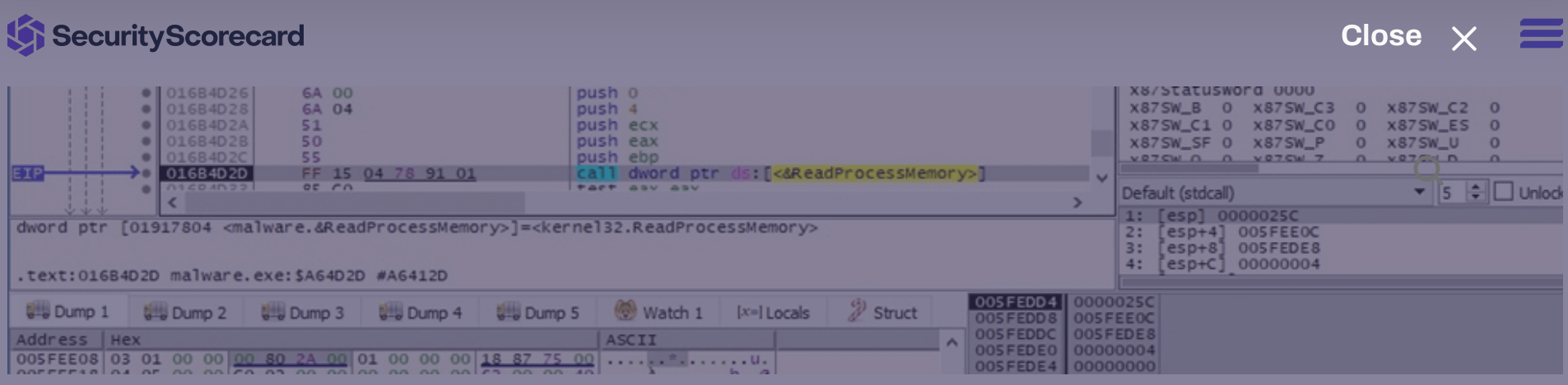


Figure 25

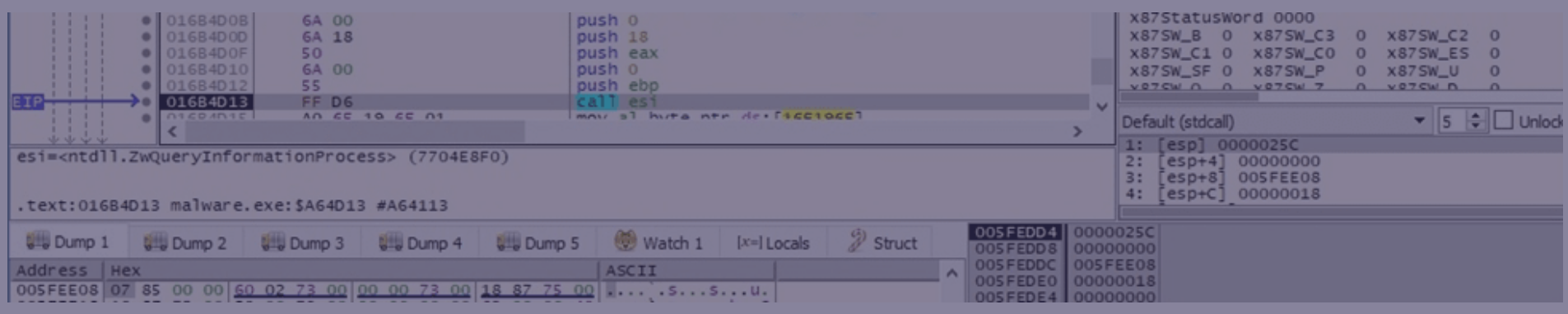


Figure 26

The path of the image file for the current process is retrieved using ReadProcessMemory:

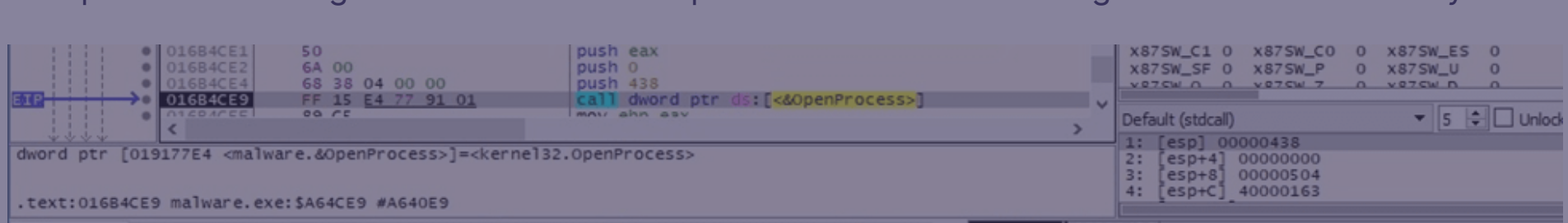
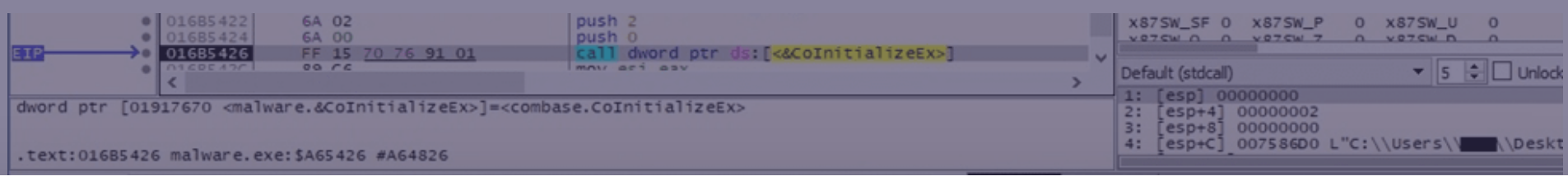


Figure 27

Privilege escalation via UAC bypass using CMSTPLUA COM interface

The ransomware initializes the COM library for use by the current thread via a call to CoInitializeEx (0x2 = **COINIT_APARTMENTTHREADED**):



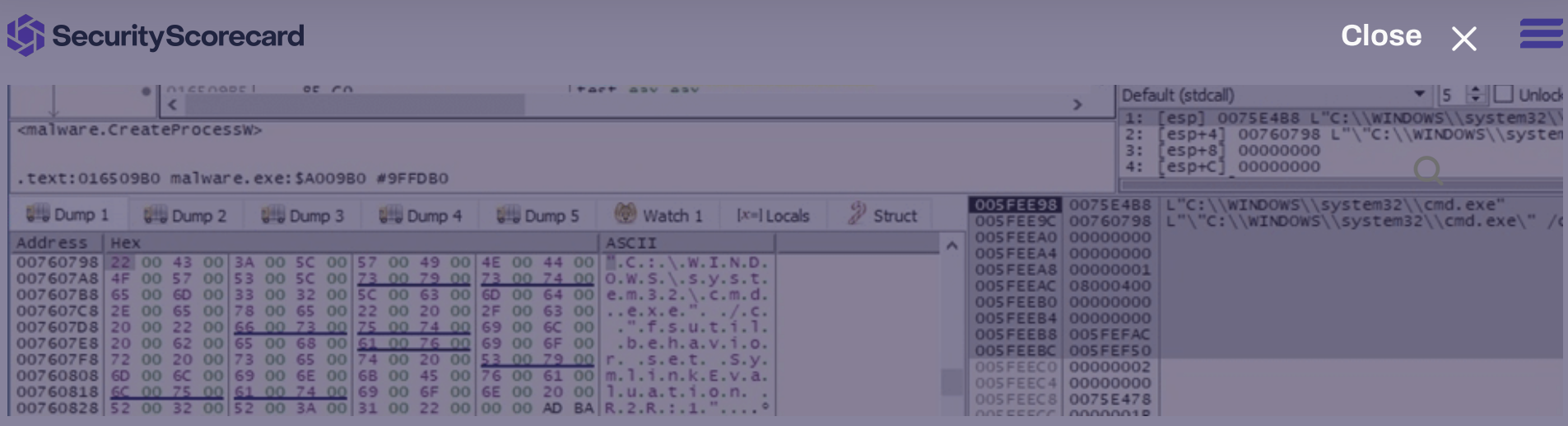


Figure 29
The initial executable is spawned with administrative privileges:

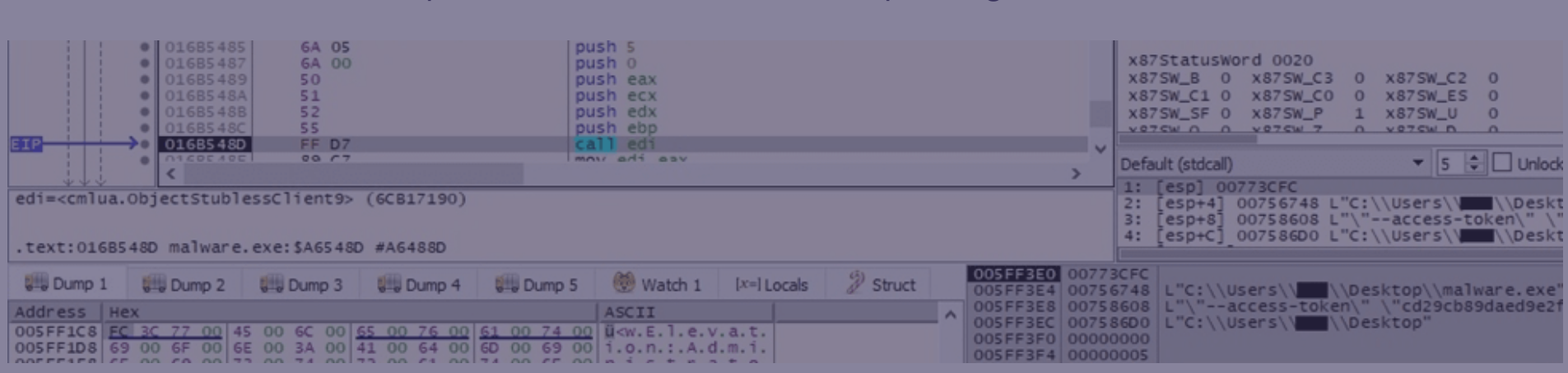


Figure 30
The LookupPrivilegeValueW routine is utilized to retrieve the locally unique identifier that represents the following privileges:

- SeIncreaseQuotaPrivilege SeSecurityPrivilege SeTakeOwnershipPrivilege
- SeLoadDriverPrivilege SeSystemProfilePrivilege SeSystemtimePrivilege
- SeProfileSingleProcessPrivilege SeIncreaseBasePriorityPrivilege
- SeCreatePagefilePrivilege SeBackupPrivilege SeRestorePrivilege
- SeShutdownPrivilege SeDebugPrivilege SeSystemEnvironmentPrivilege
- SeChangeNotifyPrivilege SeRemoteShutdownPrivilege SeUndockPrivilege
- SeManageVolumePrivilege SeImpersonatePrivilege SeCreateGlobalPrivilege
- SeIncreaseWorkingSetPrivilege SeTimeZonePrivilege
- SeCreateSymbolicLinkPrivilege SeDelegateSessionUserImpersonatePrivilege

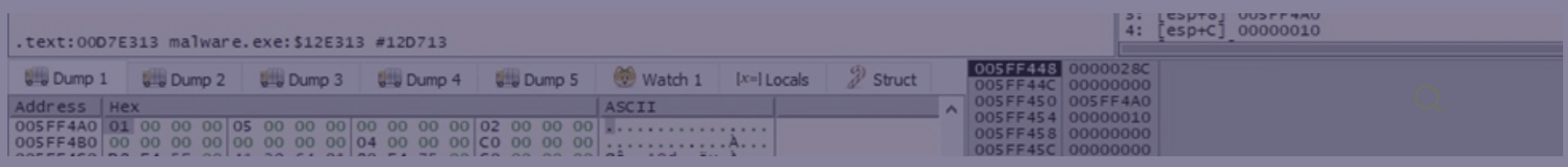


Figure 32

The binary creates the following processes that enable “remote to local” and “remote to remote” symbolic links on the local machine:

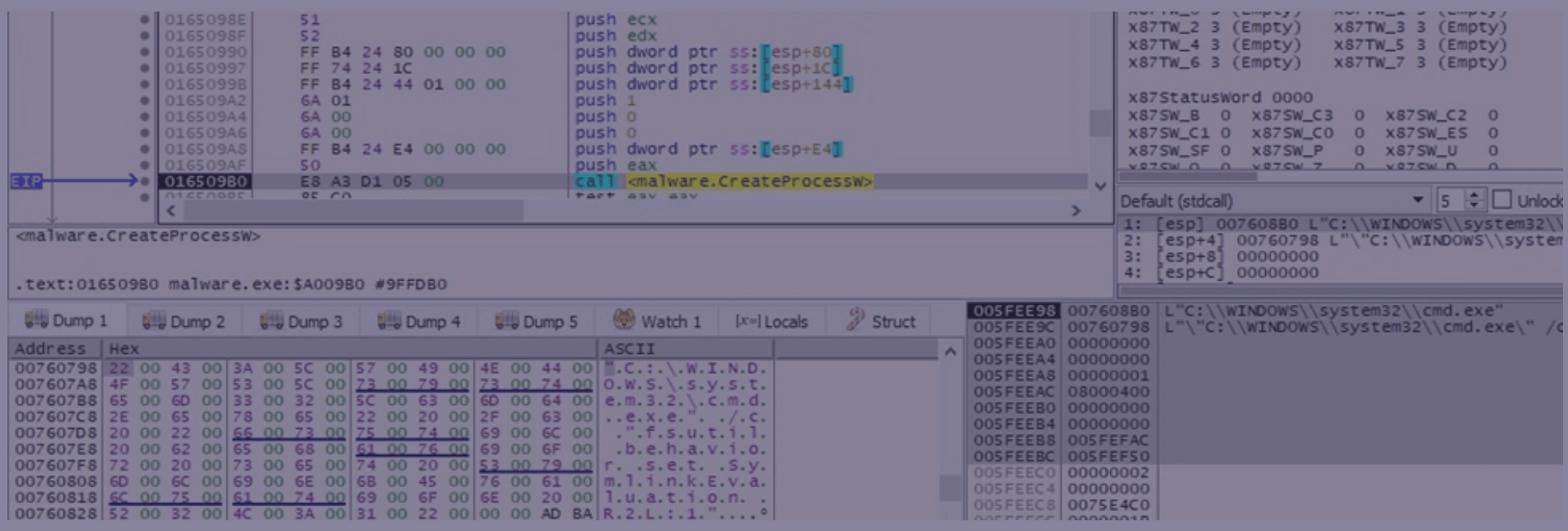


Figure 33

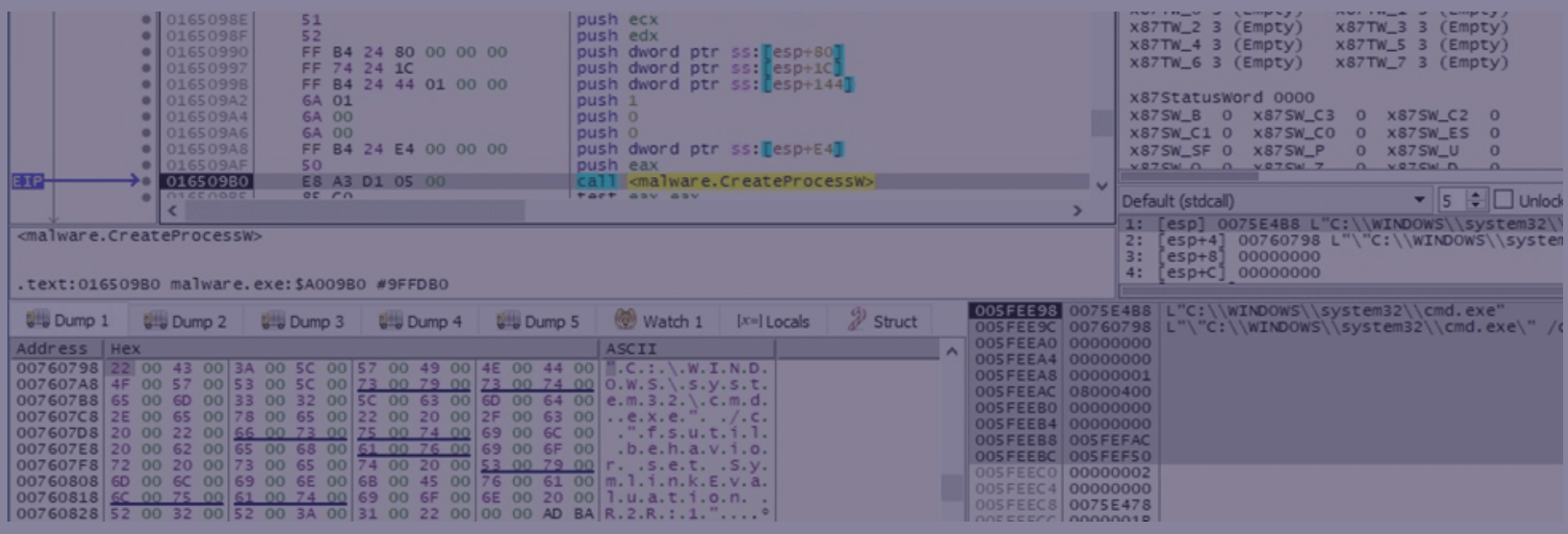


Figure 34

The malware tries to stop the Internet Information service (IIS) using IISReset.exe:

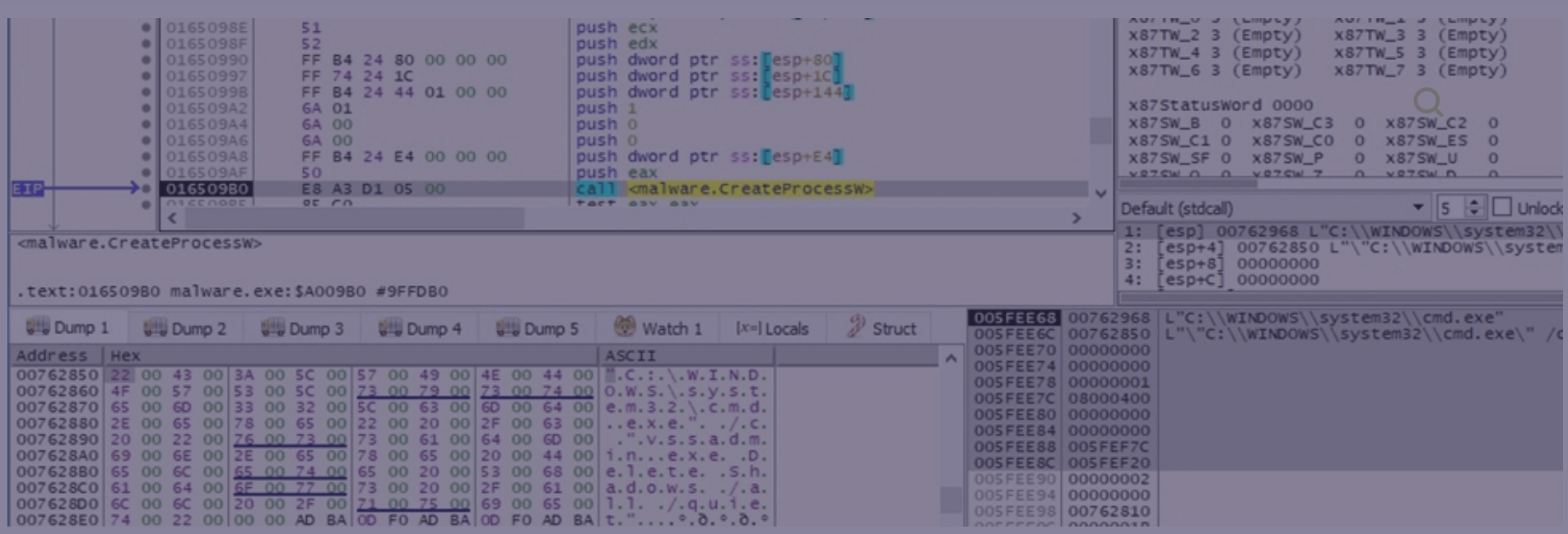


Figure 36

There is also a second process that is responsible for deleting all volume shadow copies with wmic:

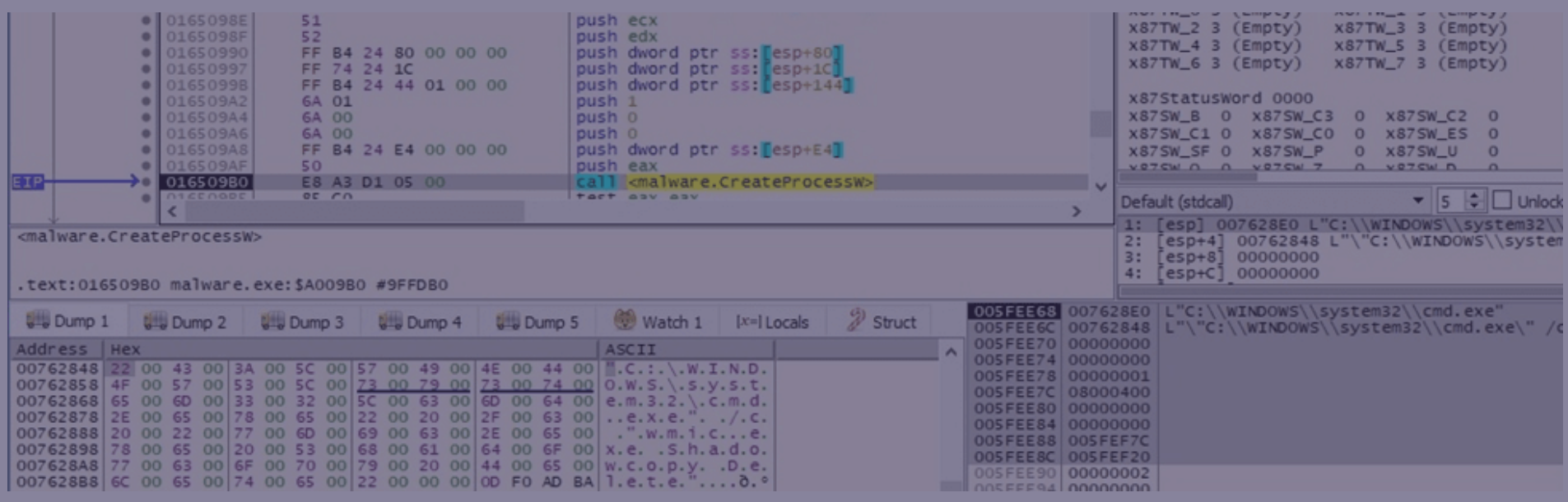
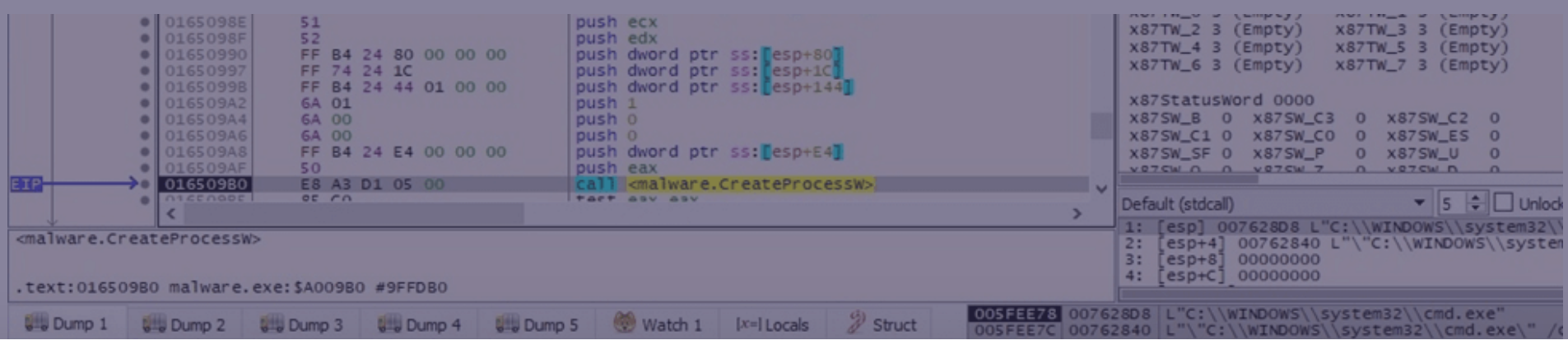


Figure 37

Interestingly, the malware runs the following command that is incomplete and returns an error:



```
007628C0  6D 65 74 65  72 20 69 73  20 69 6E 63  6F 72 72 65  meter is incorre
007628D0  63 74 2E 0D  0A F0 AD BA  0D F0 AD BA  0D F0 AD BA  ct...ð.°.ð.°.ð.°
```

Figure 39

The binary disables Automatic Repair using the bcdedit tool:

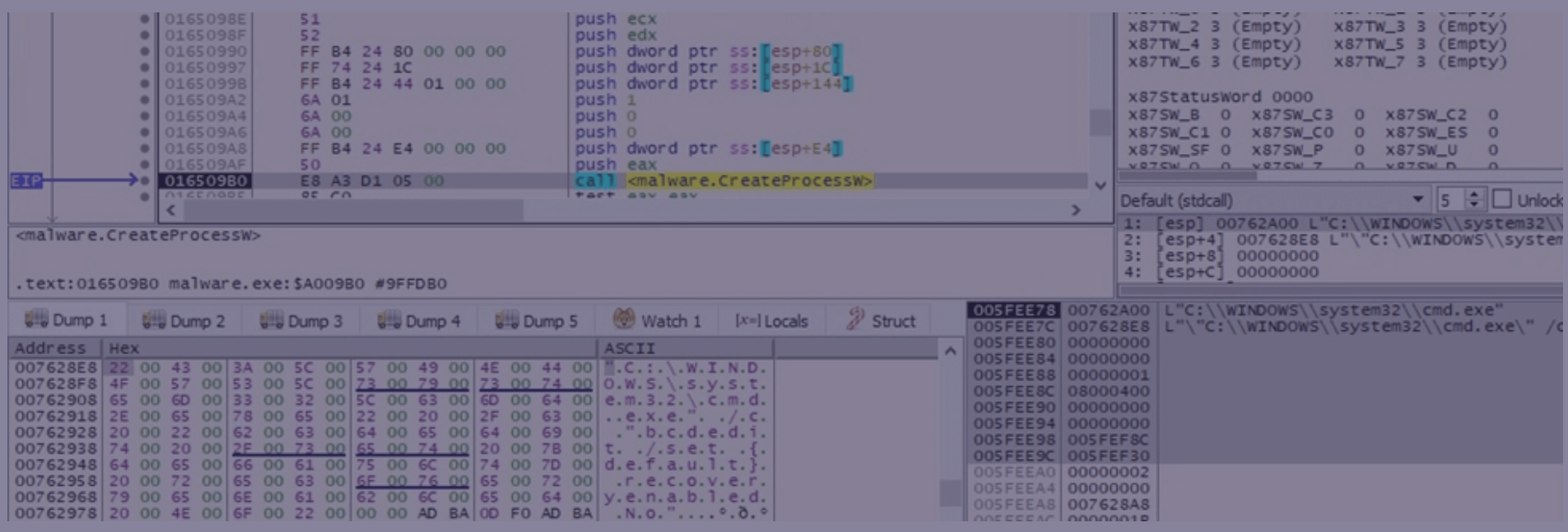


Figure 40

The ransomware tries to clear all event logs, however, the command is incorrect and returns an error, as highlighted below:

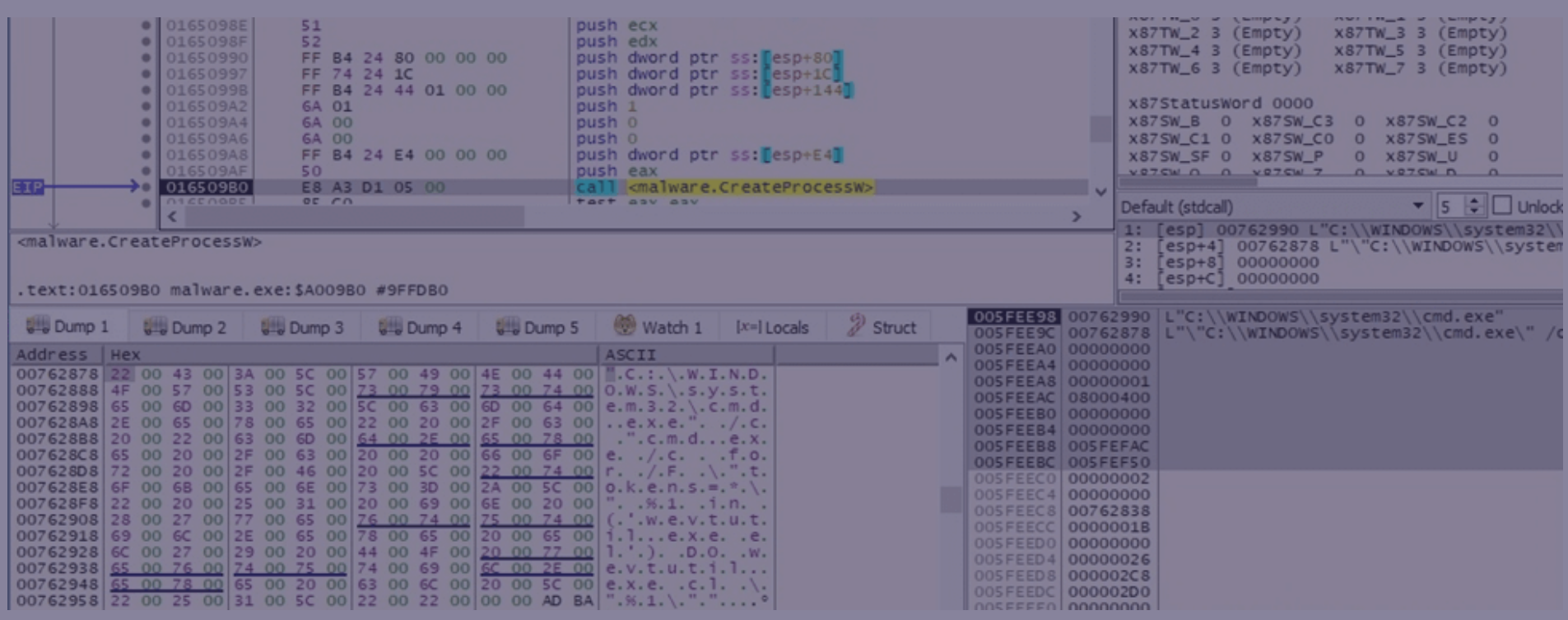


Figure 41

```
Address   Hex                                              ASCII
007628B8  15 37 D0 18  DB 28 00 18  5C 22 74 6F  6B 65 6E 73  .7Ð.Û(..\"tokens
007628C8  3D 2A 5C 22  20 77 61 73  20 75 6E 65  78 70 65 63  =*\" was unexpec
007628D8  74 65 64 20  61 74 20 74  68 69 73 20  74 69 6D 65  ted at this time
007628E8  2E 0D 0A BA  0D F0 AD BA  0D F0 AD BA  0D F0 AD BA  ...°.ð.°.ð.°.ð.°
```

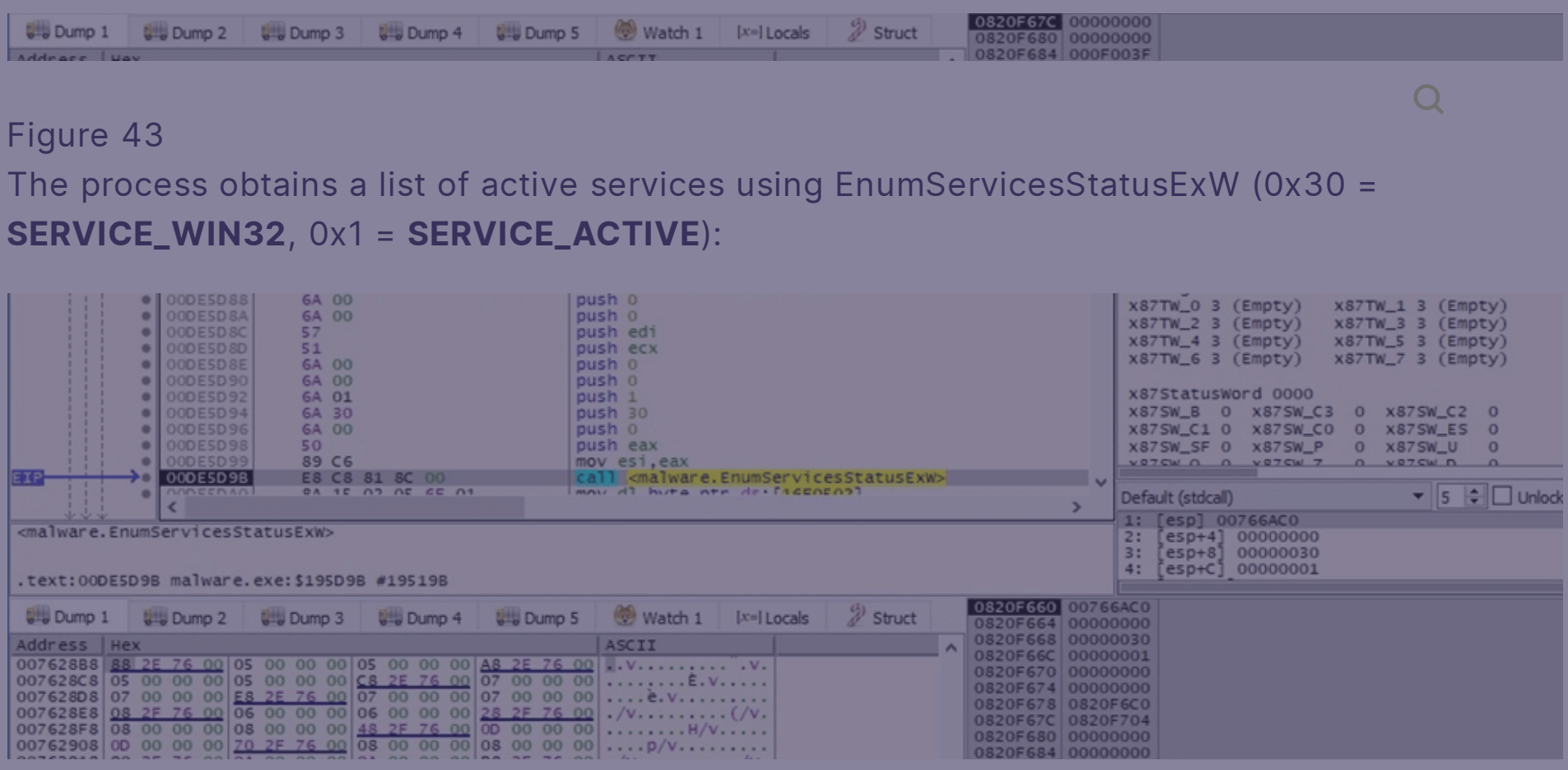


Figure 43
The process obtains a list of active services using EnumServicesStatusExW (0x30 = **SERVICE_WIN32**, 0x1 = **SERVICE_ACTIVE**):

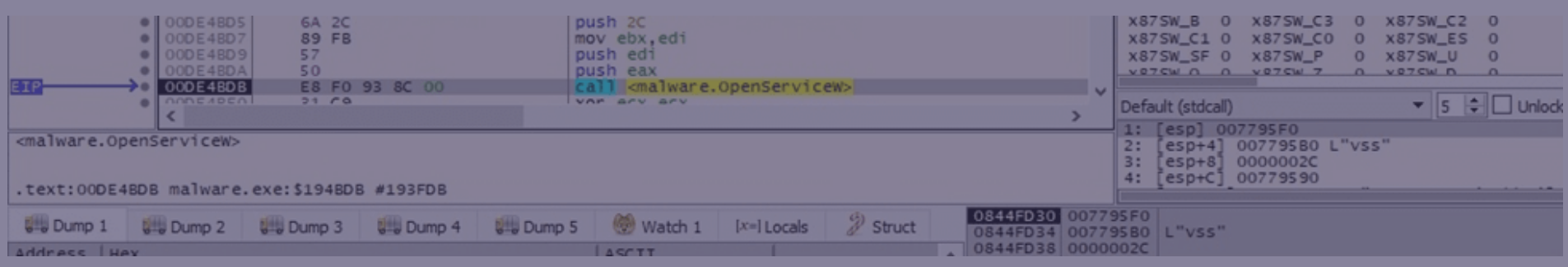


Figure 44
The malware targets the list of services from the kill_services element in the BlackCat configuration.

A targeted service is opened by calling the OpenServiceW routine (0x2c = **SERVICE_STOP | SERVICE_ENUMERATE_DEPENDENTS | SERVICE_QUERY_STATUS**):

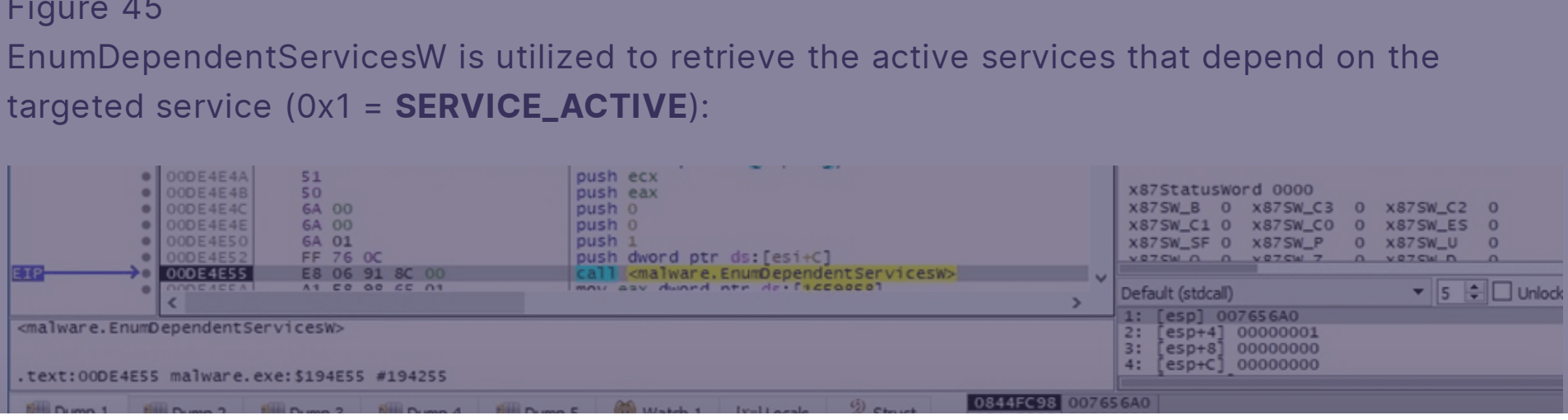


Figure 45
EnumDependentServicesW is utilized to retrieve the active services that depend on the targeted service (0x1 = **SERVICE_ACTIVE**):

Figure 47

Killing targeted processes

The executable takes a snapshot of all processes and threads in the system (0xF = **TH32CS_SNAPALL**):

Figure 48

The processes are enumerated using the Process32FirstW and Process32NextW APIs:

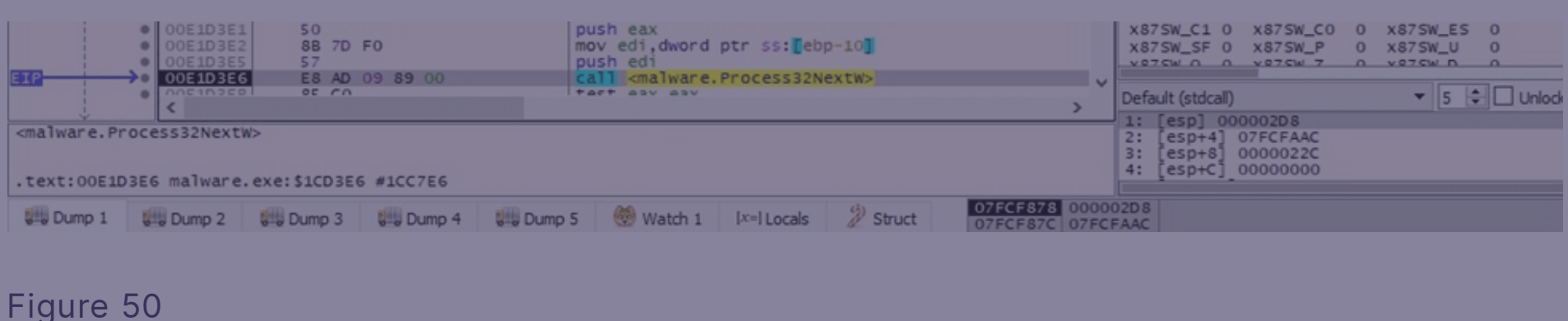
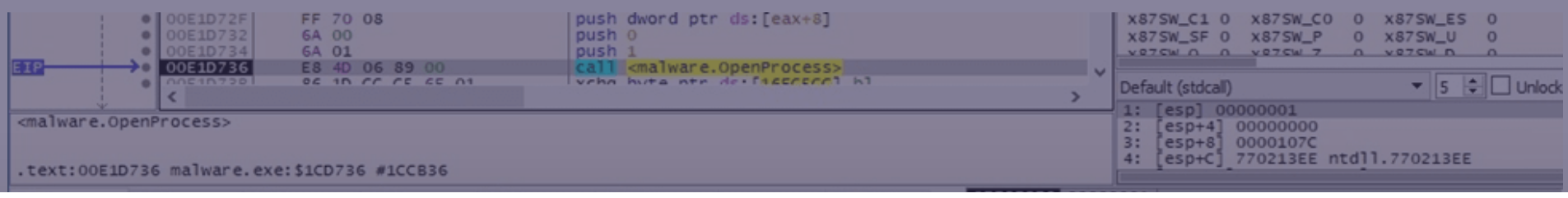


Figure 49

Figure 50

The malware targets the list of processes from the kill_processes element in the BlackCat configuration.

It opens a targeted process using OpenProcess (0x1 = **PROCESS_TERMINATE**):



specified in the credentials entry from the BlackCat configuration.

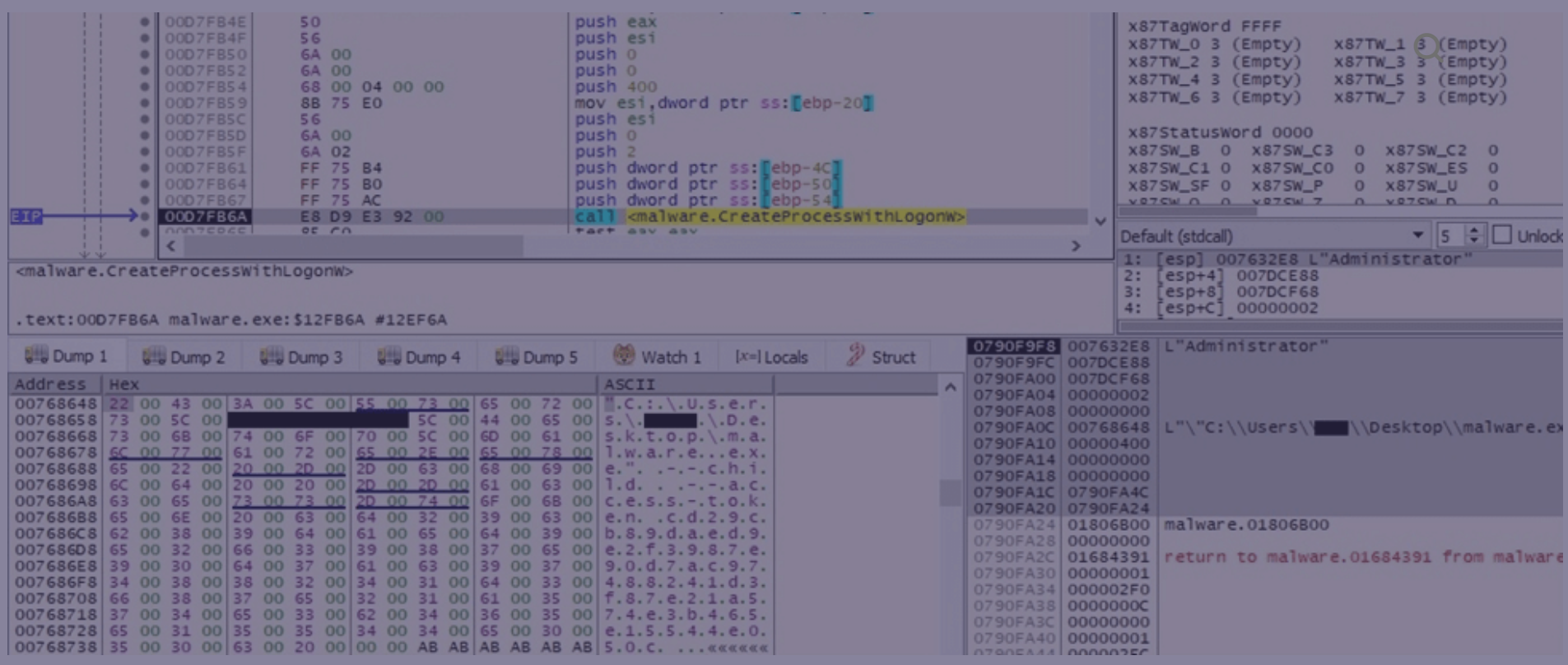


Figure 53

The number of network requests the Server Service can make is set to the maximum by modifying “HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\MaxMpxCt” Registry value:

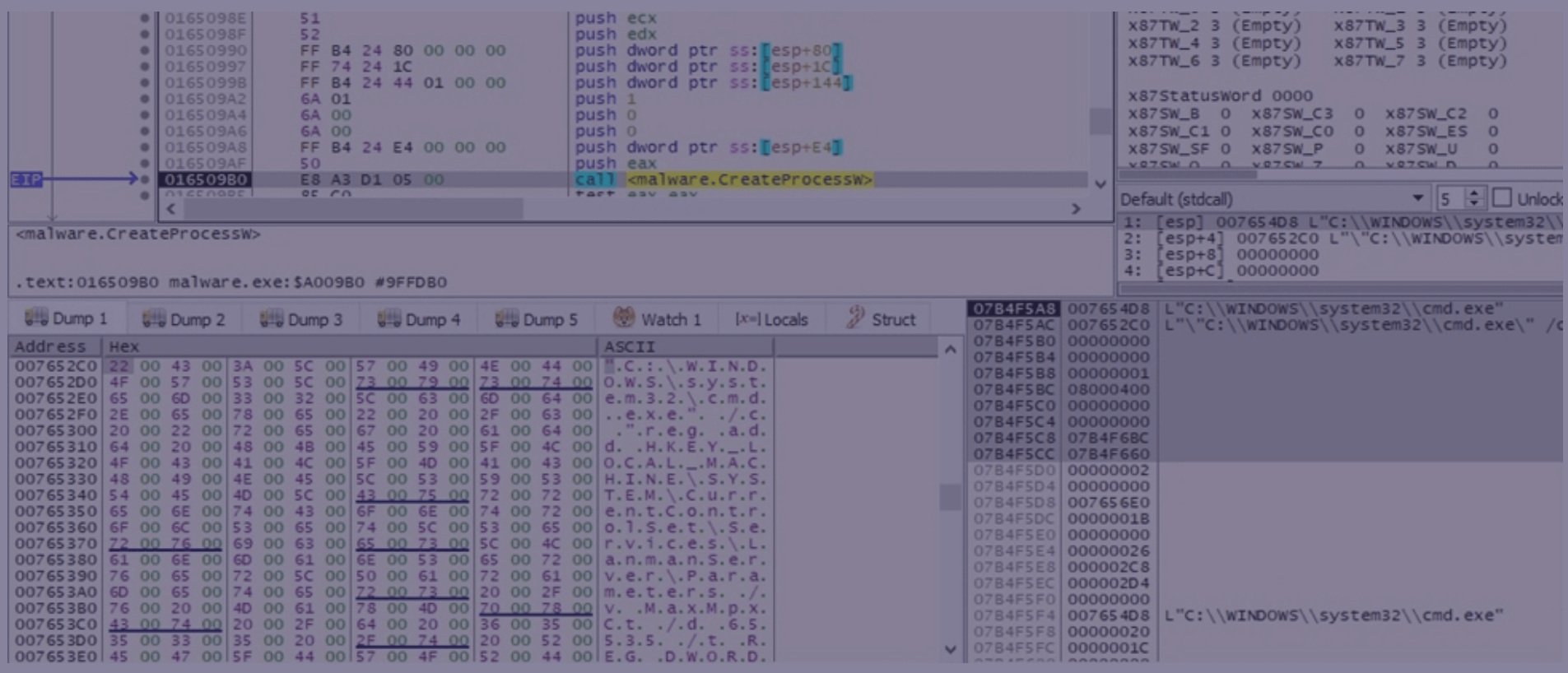


Figure 54

The malicious process obtains the ARP table using the arp command, as shown below:

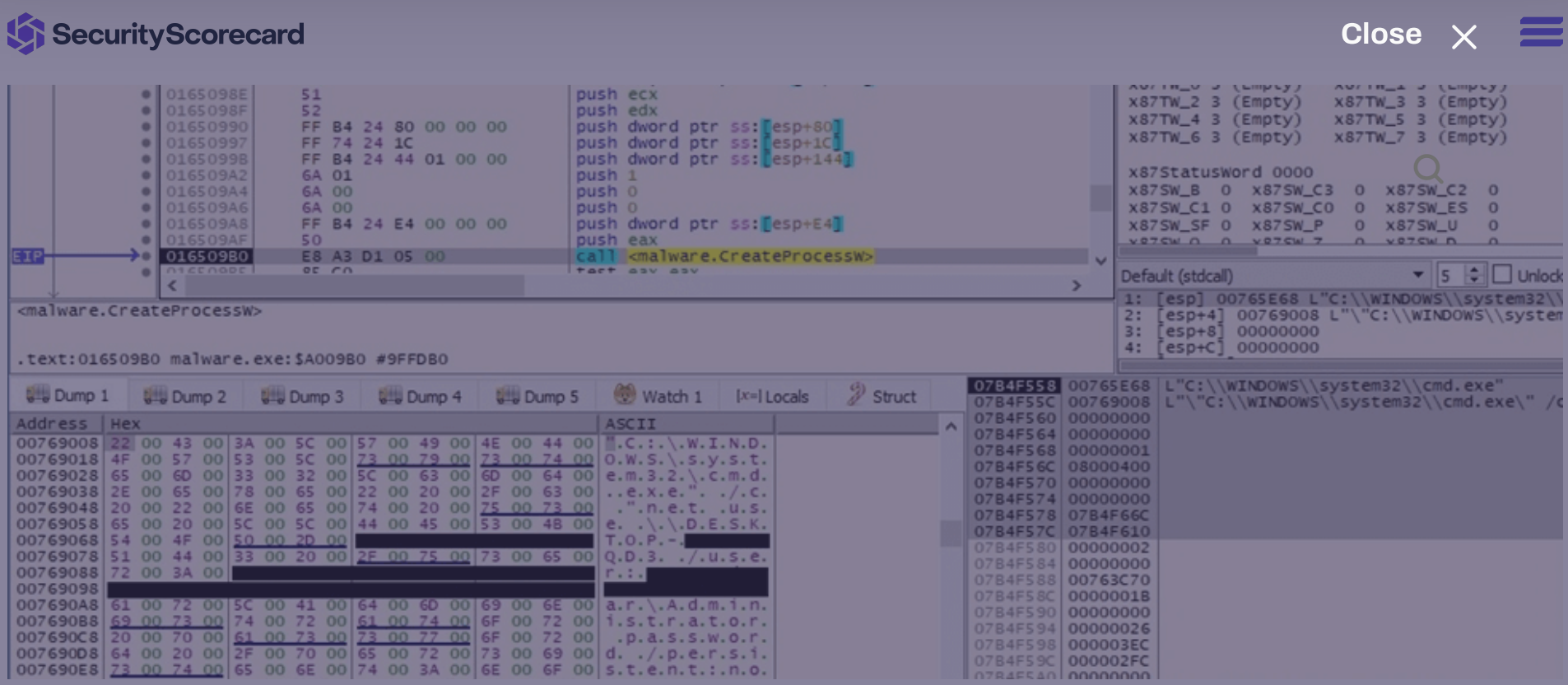


Figure 56

The malware retrieves the currently available disk drives by calling the GetLogicalDrives routine:

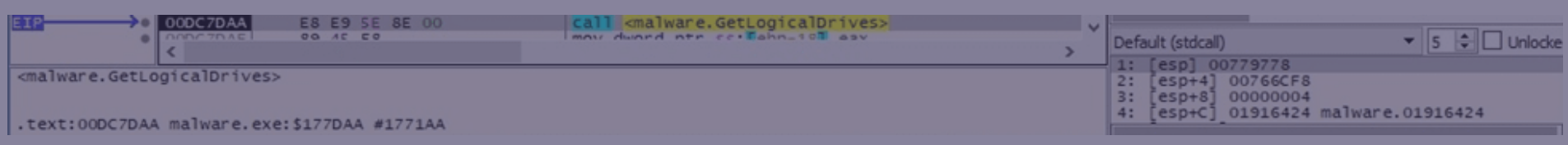


Figure 57

The GetDriveTypeW API is utilized to obtain the drive type:

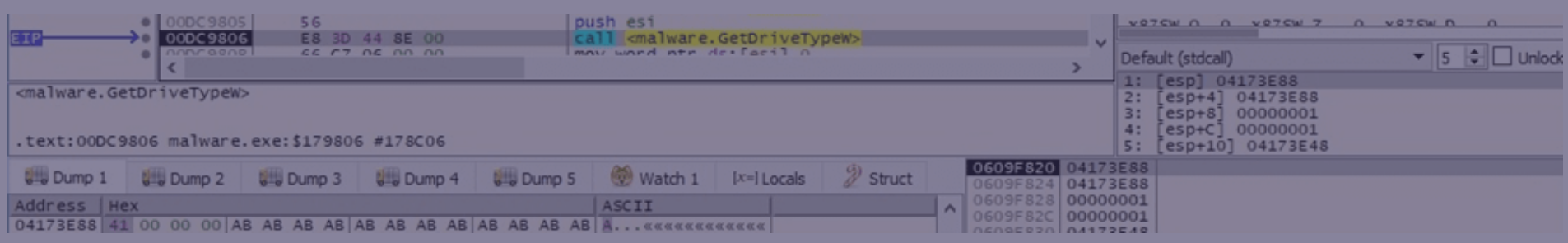


Figure 58

The ransomware starts scanning the volumes on the local machine using FindFirstVolumeW:

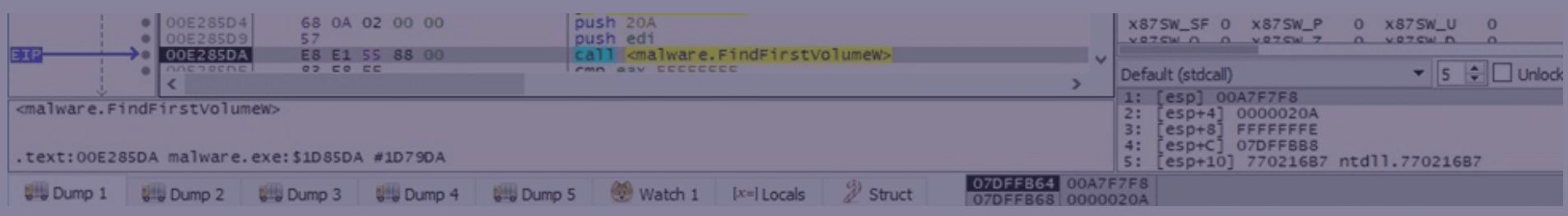


Figure 59

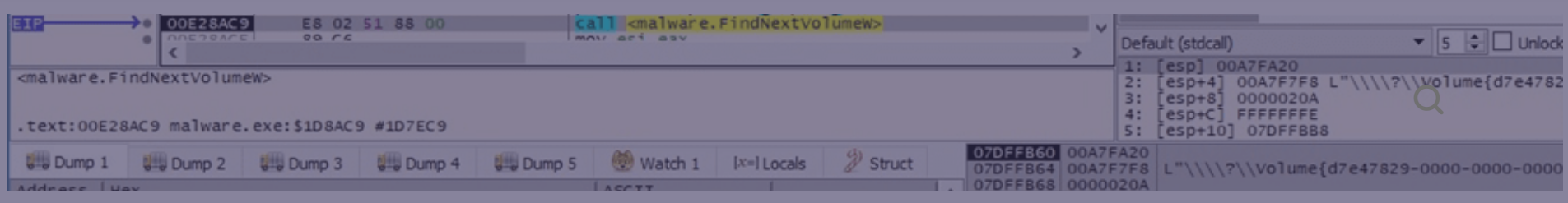


Figure 61

All unmounted volumes are mounted via a function call to SetVolumeMountPointW:

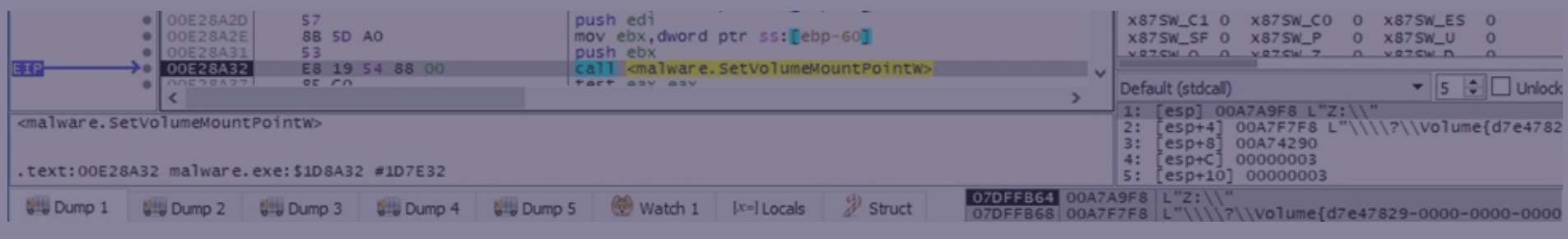


Figure 62

BlackCat traverses the file system using the FindFirstFileW and FindNextFileW APIs:

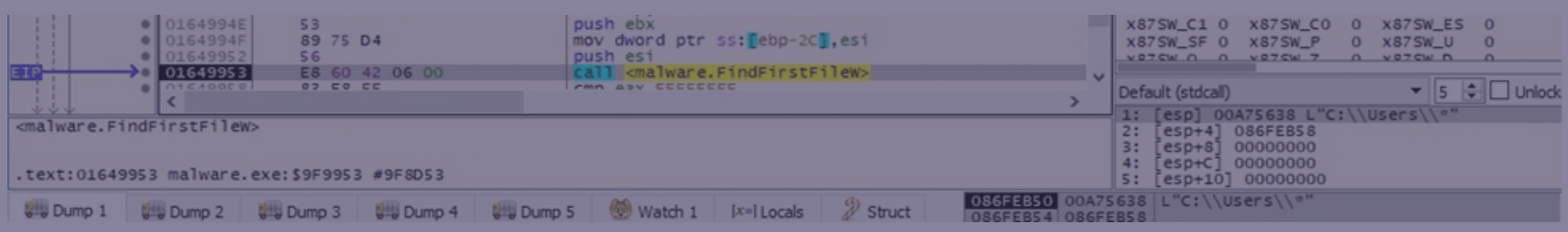


Figure 63

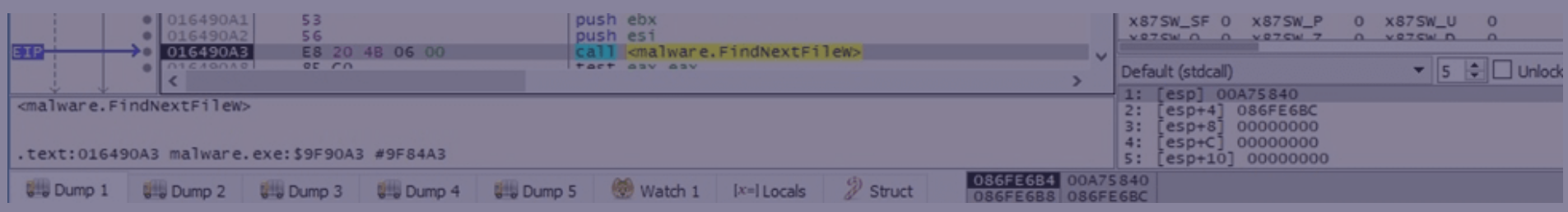


Figure 64

The BlackCat configuration is stored in JSON form and is decrypted at runtime. It contains:

- the extension appended to the encrypted files
- RSA public key that is used to encrypt the AES encryption key
- ransom note name and content
- stolen credentials specific to the victim's environment
- encryption cipher: AES

SecurityScorecard Close

Figure 65

Files encryption

The CreateFileW API is used to open a targeted file (0xC0000000 = **GENERIC_READ | GENERIC_WRITE**, 0x7 = **FILE_SHARE_DELETE | FILE_SHARE_WRITE | FILE_SHARE_READ**, 0x3 = **OPEN_EXISTING**):

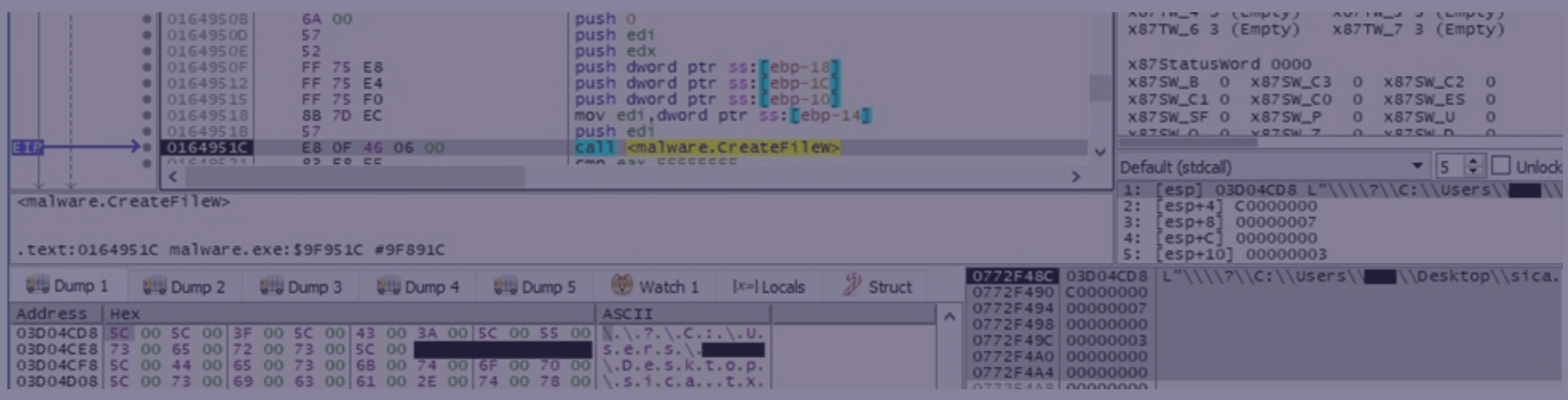
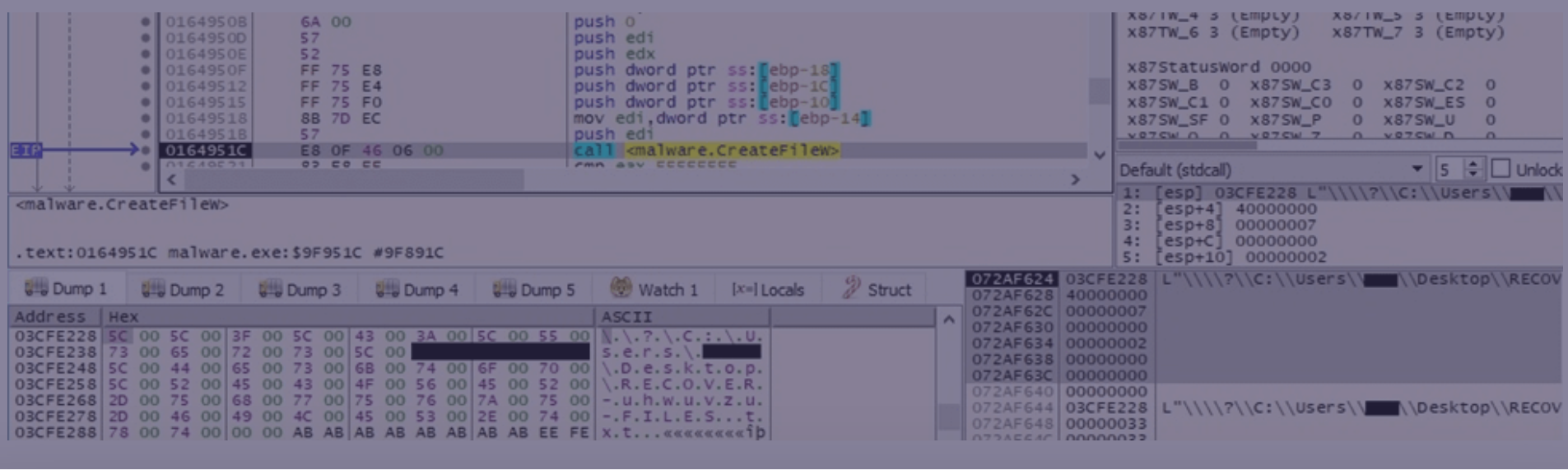


Figure 66

The ransom note is created in every traversed directory (0x40000000 = **GENERIC_WRITE**, 0x7 = **FILE_SHARE_DELETE | FILE_SHARE_WRITE | FILE_SHARE_READ**, 0x2 = **CREATE_ALWAYS**):



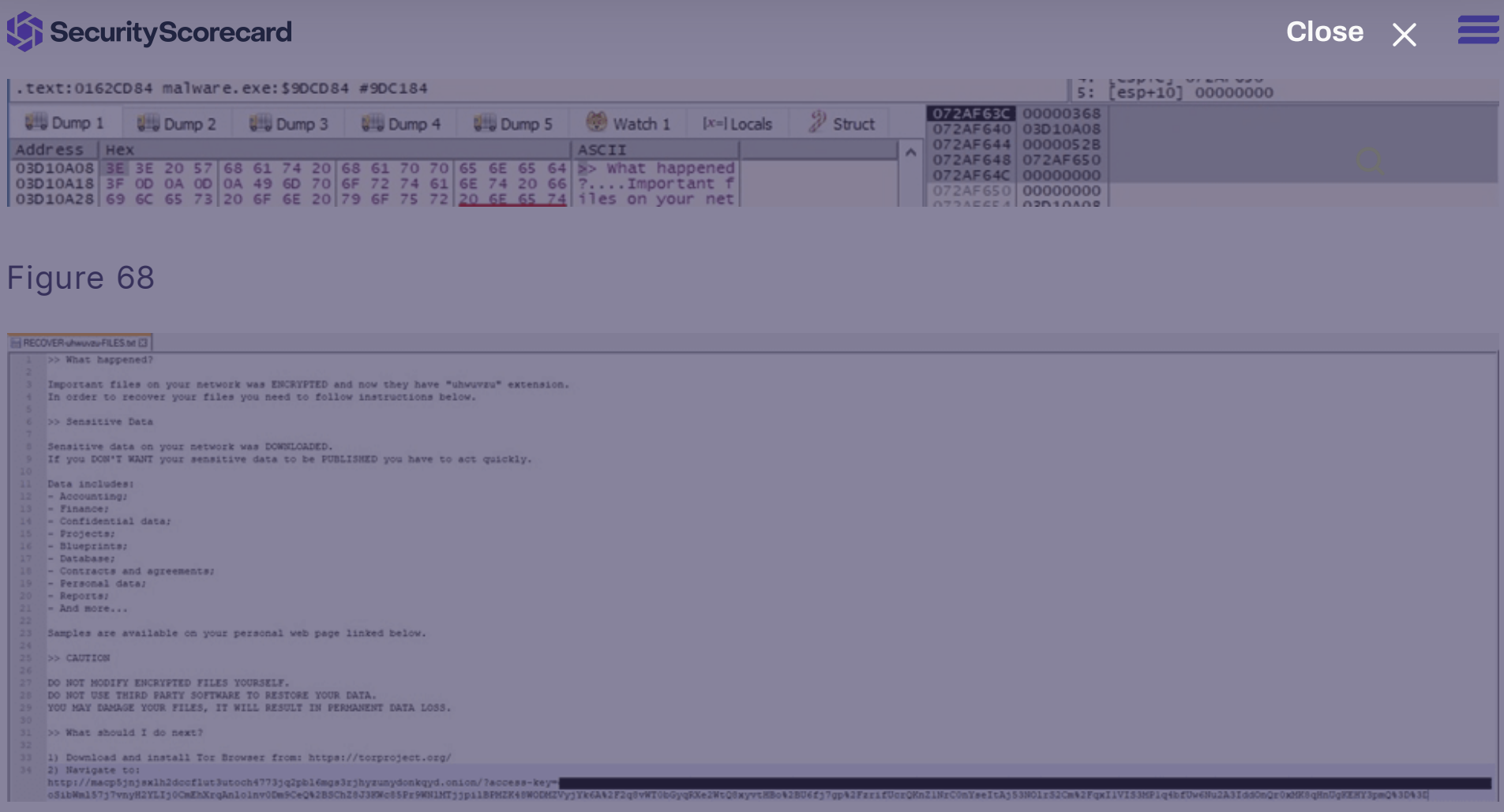


Figure 68

Figure 69

The file's extension is changed using the MoveFileExW function. The renamed file is opened using CreateFileW (0x7 = **FILE_SHARE_DELETE | FILE_SHARE_WRITE | FILE_SHARE_READ**, 0x3 = **OPEN_EXISTING**, 0x02000000 = **FILE_FLAG_BACKUP_SEMANTICS**):

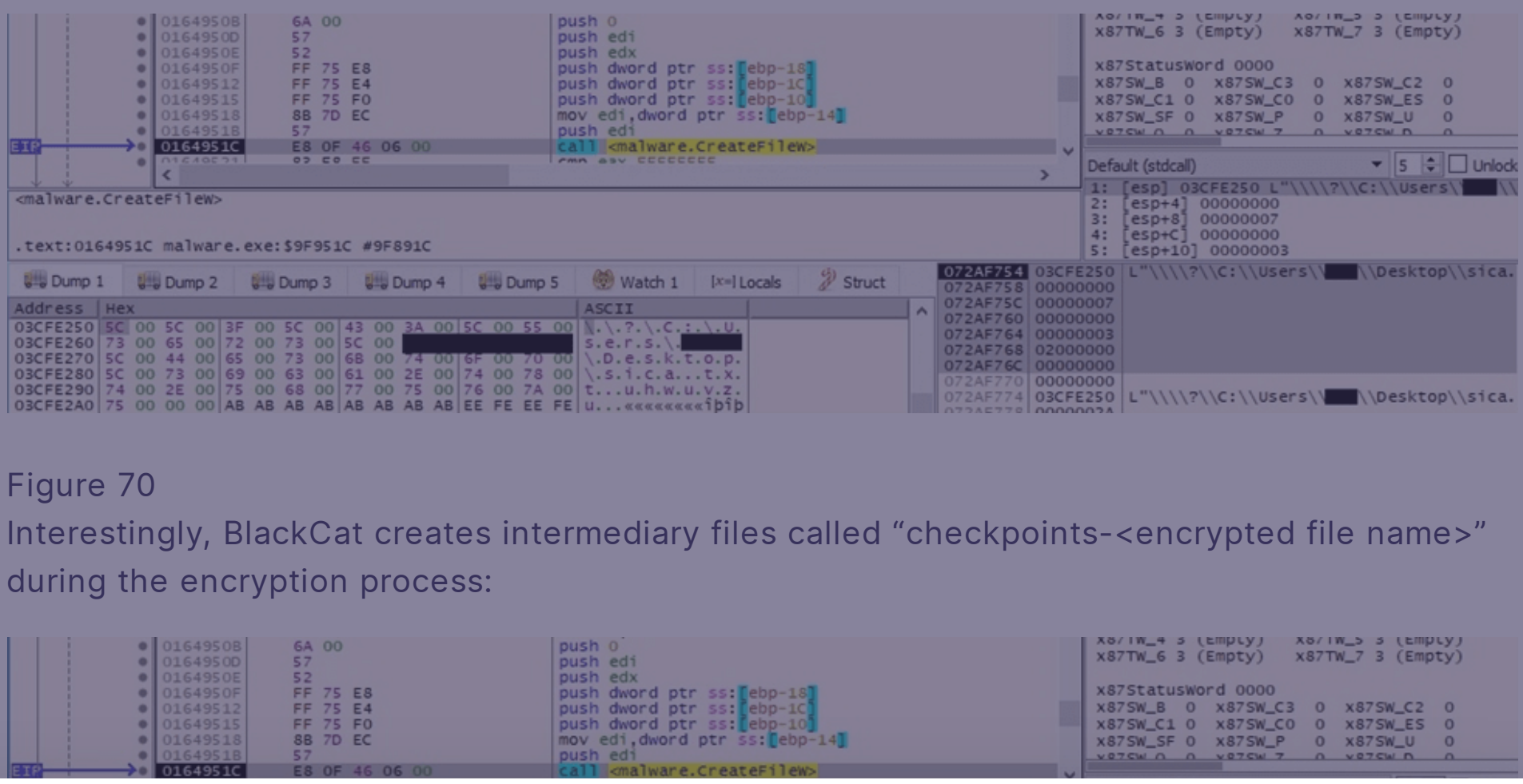


Figure 70

Interestingly, BlackCat creates intermediary files called "checkpoints-<encrypted file name>" during the encryption process:

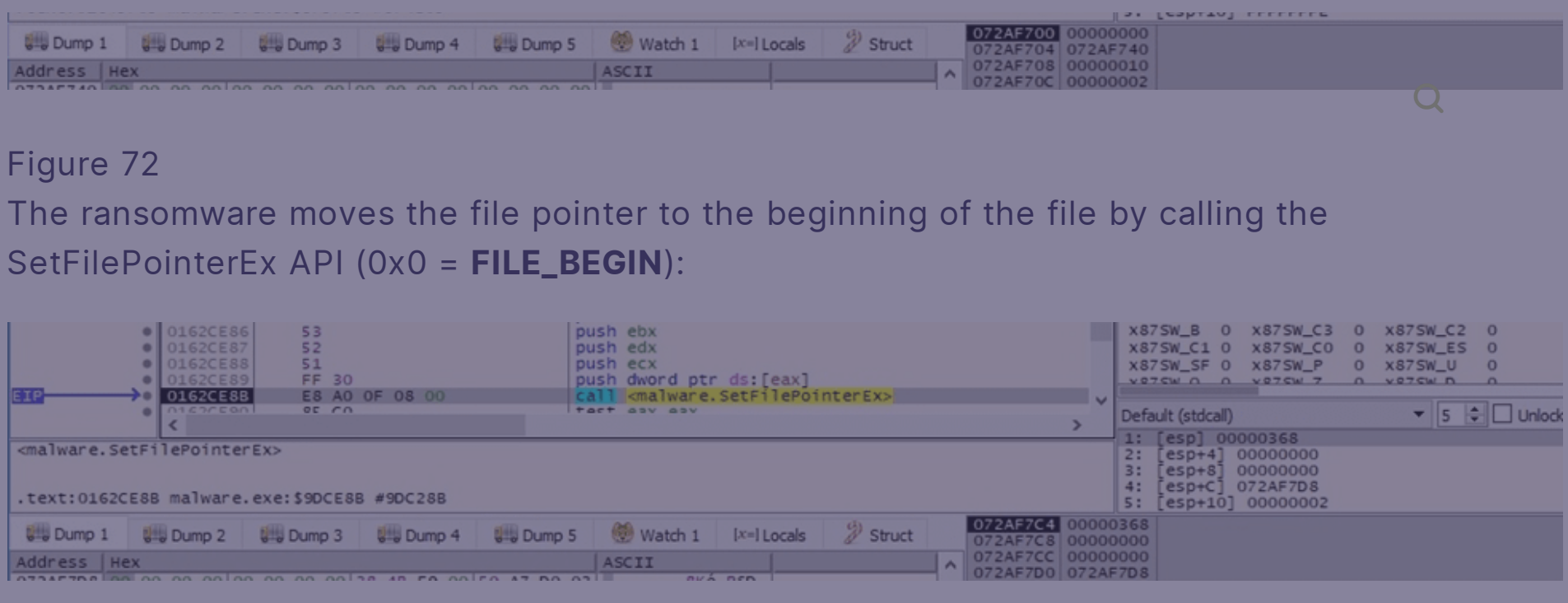


Figure 72

The ransomware moves the file pointer to the beginning of the file by calling the SetFilePointerEx API (0x0 = **FILE_BEGIN**):

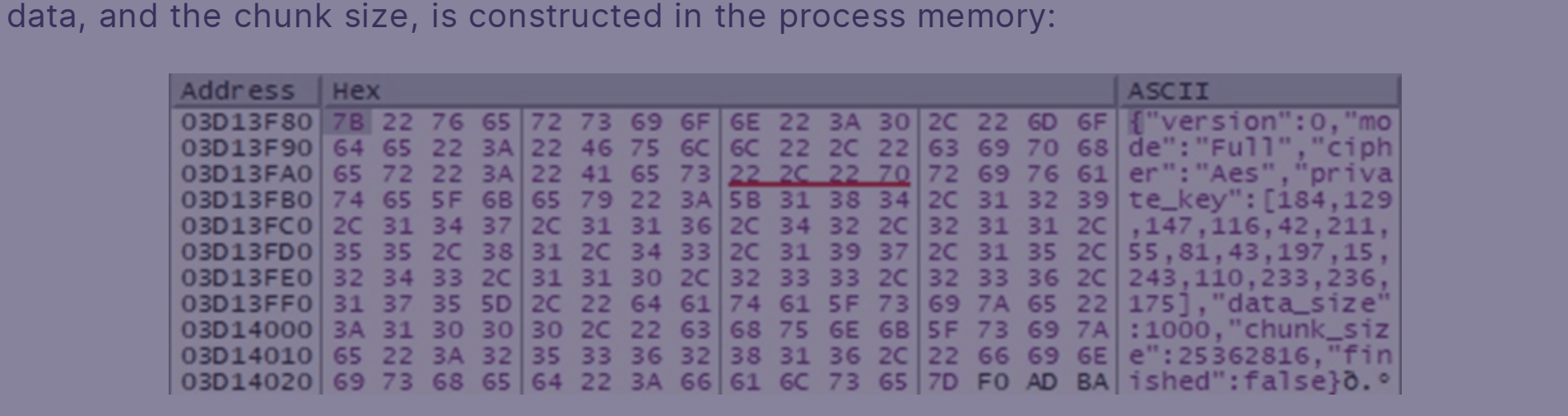


Figure 73

The process reads 4 bytes from the beginning of the file using ReadFile:

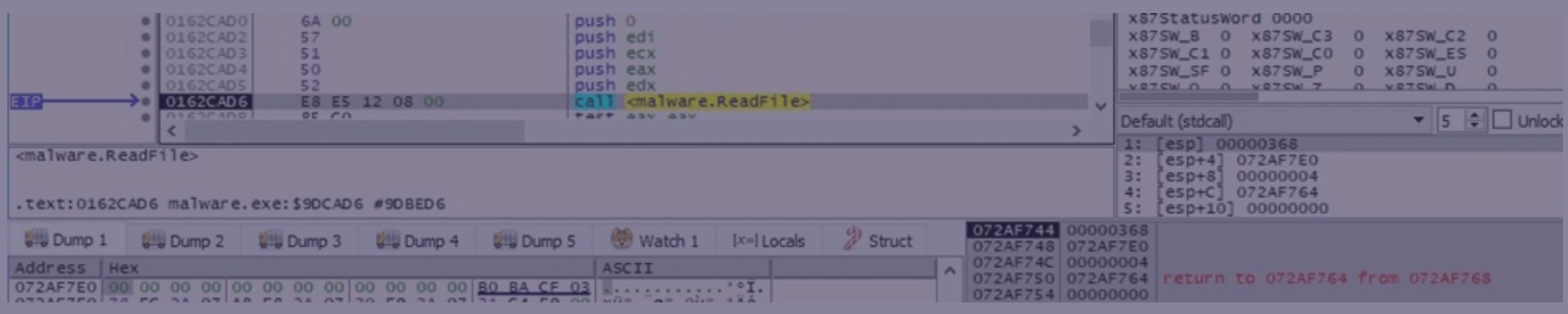


Figure 74

A JSON form containing the encryption cipher (AES), the AES key used to encrypt the file, the data, and the chunk size, is constructed in the process memory:

Address	Hex	ASCII
03D13F80	7B 22 76 65 72 73 69 6F 6E 22 3A 30 2C 22 6D 6F	{"version":0,"mo
03D13F90	64 65 22 3A 22 46 75 6C 6C 22 2C 22 63 69 70 68	de":"Full","ciph
03D13FA0	65 72 22 3A 22 41 65 73 22 2C 22 70 72 69 76 61	er":"Aes","priva
03D13FB0	74 65 5F 6B 65 79 22 3A 5B 31 38 34 2C 31 32 39	te_key":[184,129
03D13FC0	2C 31 34 37 2C 31 31 36 2C 34 32 2C 32 31 31 2C	,147,116,42,211,
03D13FD0	35 35 2C 38 31 2C 34 33 2C 31 39 37 2C 31 35 2C	55,81,43,197,15,
03D13FE0	32 34 33 2C 31 31 30 2C 32 33 33 2C 32 33 36 2C	243,110,233,236,
03D13FF0	31 37 35 5D 2C 22 64 61 74 61 5F 73 69 7A 65 22	175],"data_size"
03D14000	3A 31 30 30 30 2C 22 63 68 75 6E 6B 5F 73 69 7A	:1000,"chunk_siz
03D14010	65 22 3A 32 35 33 36 32 38 31 36 2C 22 66 69 6E	e":25362816,"fin
03D14020	69 73 68 65 64 22 3A 66 61 6C 73 65 7D F0 AD BA	ished":false}ð.°

Figure 75

The binary generates 0x50 (80) random bytes that are used to border the JSON form. The resulting buffer has a size of 256 bytes and is rotated using instructions such as pshuflw:

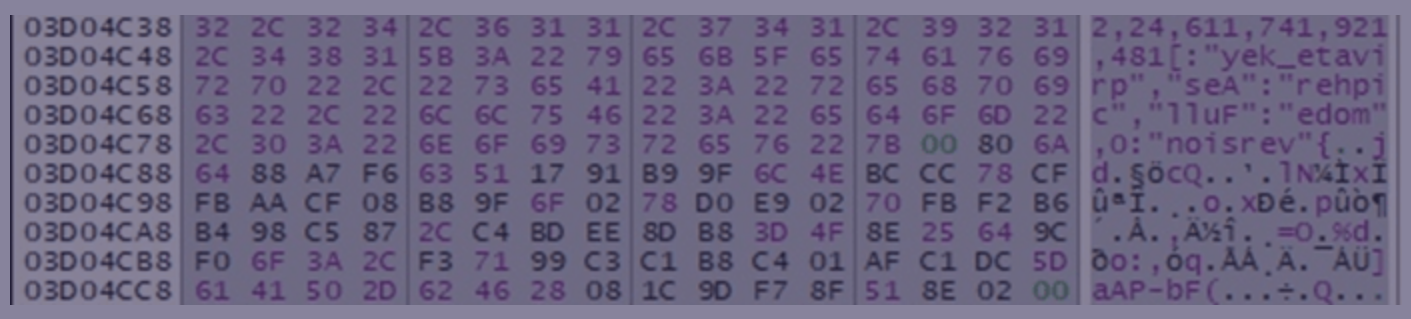


Figure 77

A 4-byte border “19 47 B2 CE” that separates the encrypted file content from the encrypted AES key is written to the file:

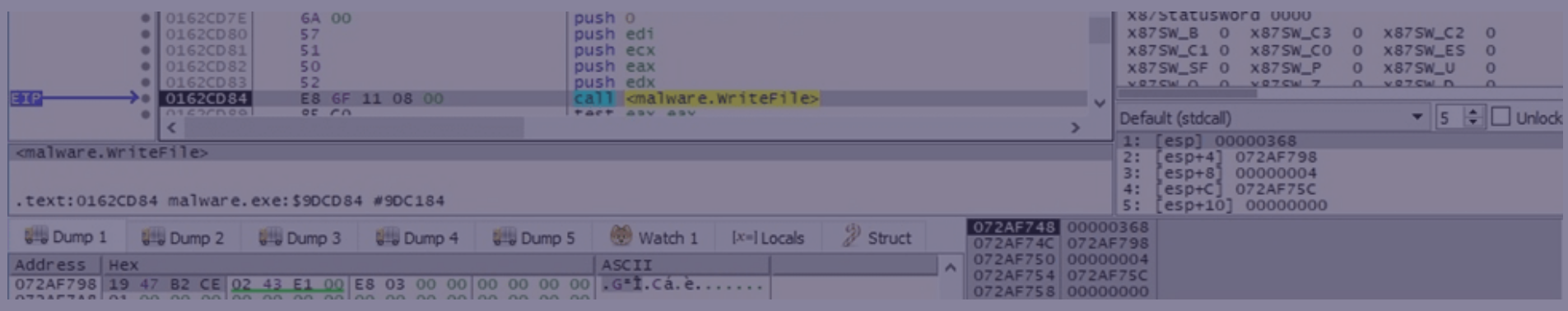


Figure 78

The buffer that contains the AES key presented in figure 77 is encrypted with the RSA public key from the BlackCat configuration. The result is written to the file using WriteFile:

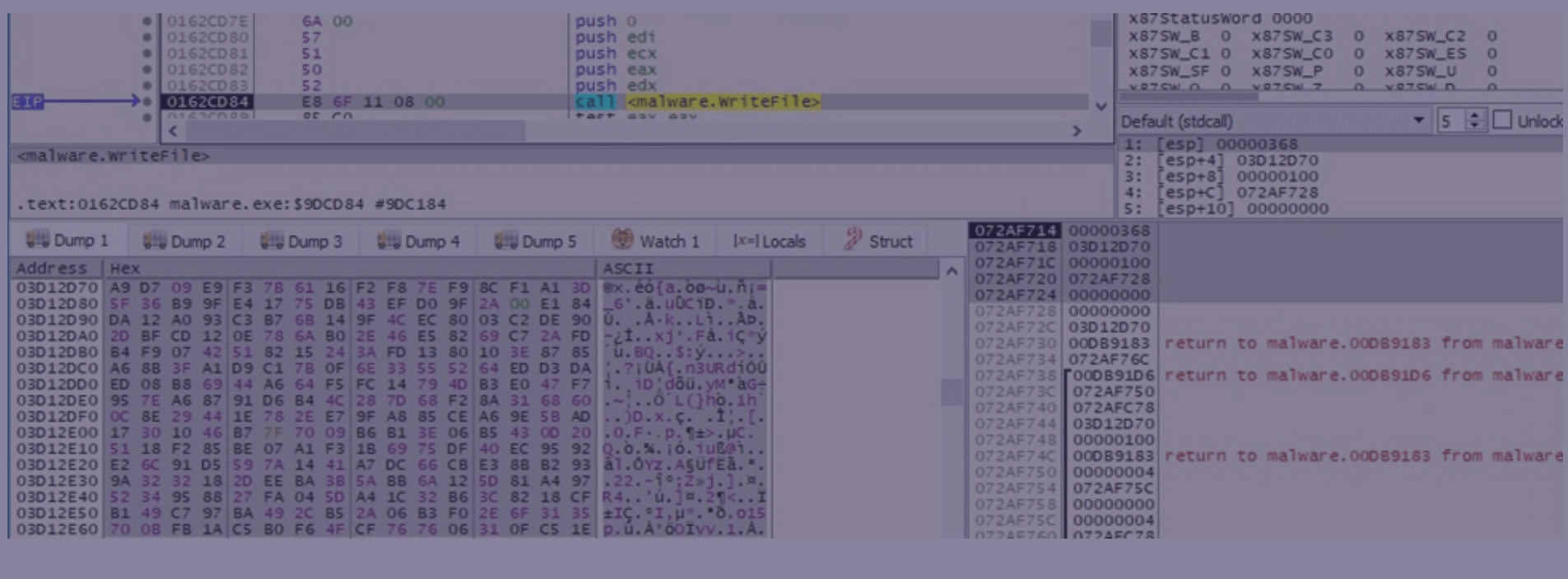
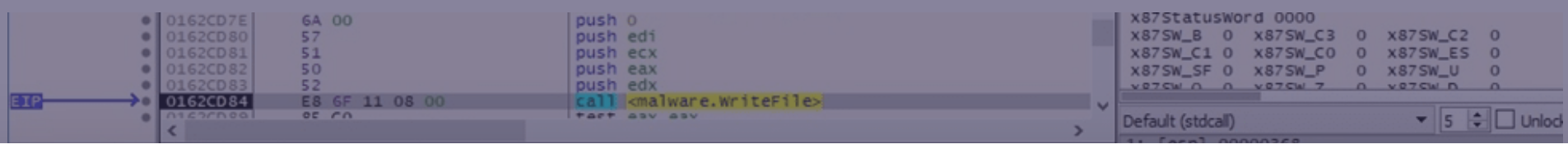


Figure 79

The size of encrypted key (0x100) is written to the file:



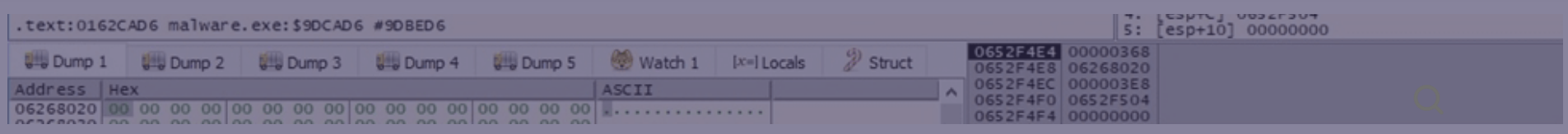


Figure 81

The file content is encrypted using the AES-128 algorithm. The malware uses the aesenc and aesenclast instructions for this purpose:

```
00DD84C7  66 0F 38 DC D0        aesenc xmm2,xmm0
00DD84CC  66 0F 38 DC D8        aesenc xmm3,xmm0
00DD84D1  66 0F 38 DC E0        aesenc xmm4,xmm0
00DD84D6  66 0F 38 DC E8        aesenc xmm5,xmm0
00DD84DB  66 0F 38 DC F0        aesenc xmm6,xmm0
00DD84E0  66 0F 38 DC F8        aesenc xmm7,xmm0
00DD84E5  66 0F 38 DC C8        aesenc xmm1,xmm0
00DD84EA  66 0F 7F 14 24        movdqa xmmword ptr ss:[esp],xmm2
00DD84EF  66 0F 6F 54 24 10     movdqa xmm2,xmmword ptr ss:[esp+10]
00DD84F5  66 0F 38 DC D0        aesenc xmm2,xmm0
00DD84FA  66 0F 6F 41 20        movdqa xmm0,xmmword ptr ds:[ecx+20]
00DD84FF  66 0F 7F 54 24 10     movdqa xmmword ptr ss:[esp+10],xmm2
00DD8505  66 0F 6F 14 24        movdqa xmm2,xmmword ptr ss:[esp]
00DD850A  66 0F 38 DC D8        aesenc xmm3,xmm0
00DD850F  66 0F 38 DC E0        aesenc xmm4,xmm0
00DD8514  66 0F 38 DC E8        aesenc xmm5,xmm0
00DD8519  66 0F 38 DC F0        aesenc xmm6,xmm0
00DD851E  66 0F 38 DC F8        aesenc xmm7,xmm0
00DD8523  66 0F 38 DC C8        aesenc xmm1,xmm0
00DD8528  66 0F 38 DC D0        aesenc xmm2,xmm0
00DD852D  66 0F 7F 14 24        movdqa xmmword ptr ss:[esp],xmm2
00DD8532  66 0F 6F 54 24 10     movdqa xmm2,xmmword ptr ss:[esp+10]
00DD8538  66 0F 38 DC D0        aesenc xmm2,xmm0
00DD853D  66 0F 6F 41 30        movdqa xmm0,xmmword ptr ds:[ecx+30]
00DD8542  66 0F 7F 54 24 10     movdqa xmmword ptr ss:[esp+10],xmm2
```

Figure 82

```
EIP
00DD87A0  55                    push ebp
00DD87A1  89 E5                 mov ebp,esp
00DD87A3  53                    push ebx
00DD87A4  8B 45 08              mov eax,dword ptr ss:[ebp+8]
00DD87A7  66 0F 6F 02           movdqa xmm0,xmmword ptr ds:[edx]
00DD87AB  B3 F1                 mov bl,F1
00DD87AD  66 0F EF 00           pxor xmm0,xmmword ptr ds:[eax]
00DD87B1  66 0F 38 DC 42 10     aesenc xmm0,xmmword ptr ds:[edx+10]
00DD87B7  66 0F 38 DC 42 20     aesenc xmm0,xmmword ptr ds:[edx+20]
00DD87BD  66 0F 38 DC 42 30     aesenc xmm0,xmmword ptr ds:[edx+30]
00DD87C3  66 0F 38 DC 42 40     aesenc xmm0,xmmword ptr ds:[edx+40]
00DD87C9  66 0F 38 DC 42 50     aesenc xmm0,xmmword ptr ds:[edx+50]
00DD87CF  66 0F 38 DC 42 60     aesenc xmm0,xmmword ptr ds:[edx+60]
00DD87D5  66 0F 38 DC 42 70     aesenc xmm0,xmmword ptr ds:[edx+70]
00DD87DB  66 0F 38 DC 82 80 00 00 00  aesenc xmm0,xmmword ptr ds:[edx+80]
00DD87E4  86 1D FE 8B 6E 01     xchg byte ptr ds:[16E8BFE],bl
00DD87EA  66 0F 38 DC 82 90 00 00 00  aesenc xmm0,xmmword ptr ds:[edx+90]
00DD87F3  66 0F 7F 00           movdqa xmmword ptr ds:[eax],xmm0
00DD87F7  66 0F 38 DD 82 A0 00 00 00  aesenclast xmm0,xmmword ptr ds:[edx+A0]
00DD8800  66 0F 7F 01           movdqa xmmword ptr ds:[ecx],xmm0
```

Figure 83

The encrypted file content is written back to the file using WriteFile:

```
0162CD7E  6A 00                 push 0
0162CD80  57                    push edi
0162CD81  51                    push ecx
0162CD82  50                    push eax
0162CD83  52                    push edx
EIP
0162CD84  E8 6F 11 08 00        call <malware.WriteFile>
```

x87StatusWord 0000
x87SW_B 0 x87SW_C3 0 x87SW_C2 0
x87SW_C1 0 x87SW_C0 0 x87SW_ES 0
x87SW_SF 0 x87SW_P 0 x87SW_U 0
Default (stdcall) 5 Unlock

Figure 85

The ransomware creates a PNG image called “RECOVER-uhwuvzu-FILES.txt.png”:

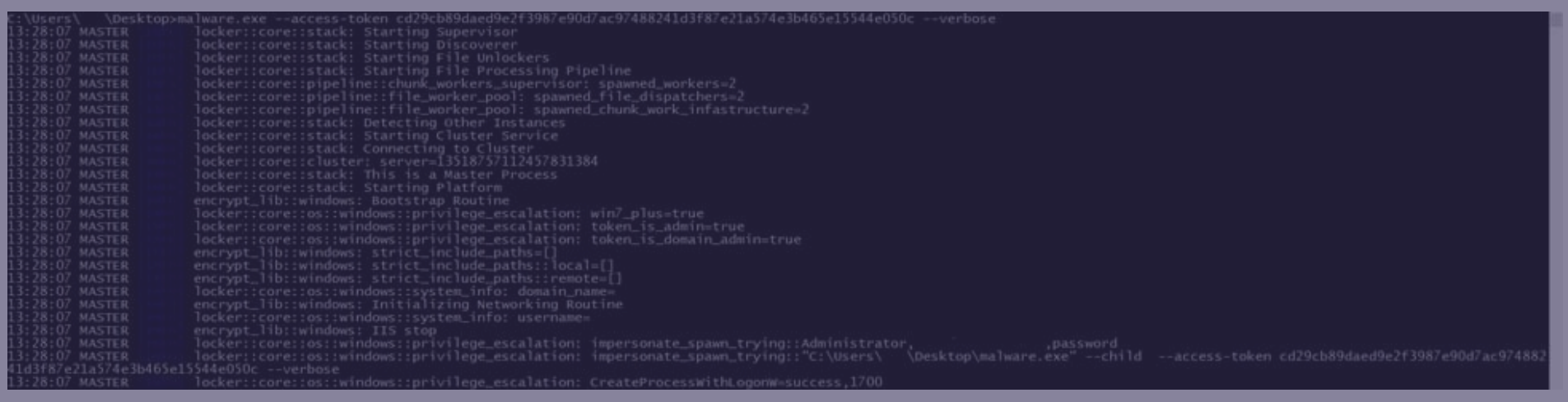


Figure 86

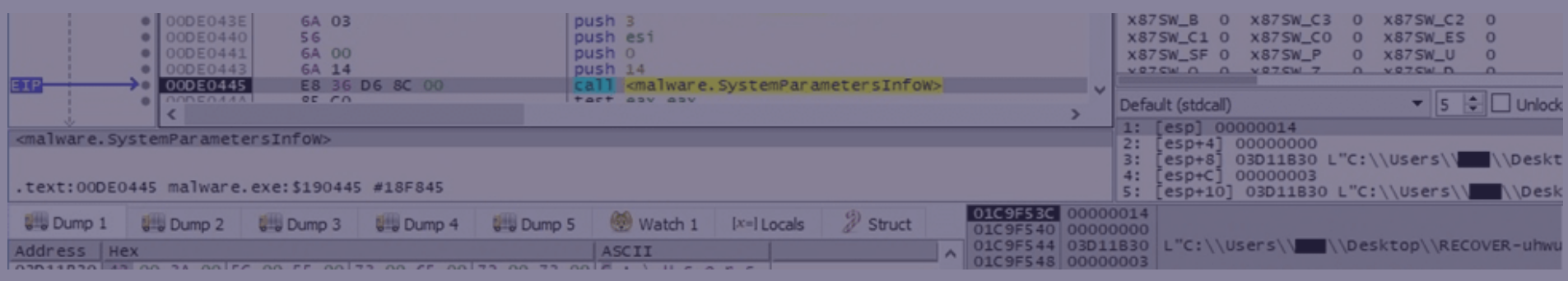
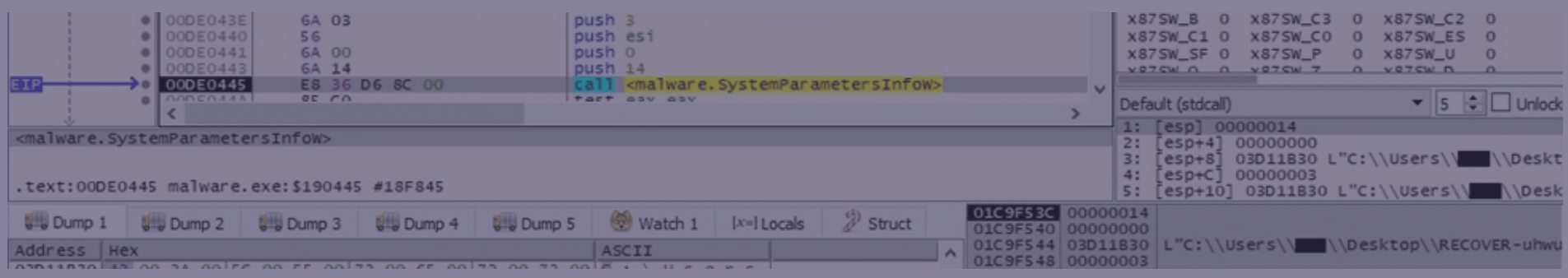


Figure 87

The Desktop wallpaper is changed to the above image by calling the SystemParametersInfoW API (0x14 = **SPI_SETDESKWALLPAPER**, 0x3 = **SPIF_UPDATEINIFILE | SPIF_SENDCHANGE**):



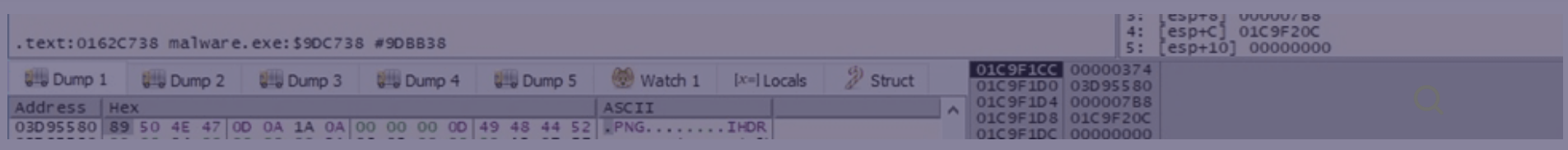


Figure 89

Running with the –extra-verbose –ui parameters

The malware presents the relevant information in the following window:

sica.txt.uhwuvzu

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
000001B0   DD 63 AC D7 EB 79 F6 CA 2E 3C CE 98 12 84 78 C0   Ýc¬×ëyöÊ.<Î˜.„xÀ
000001C0   4B 8C BB F0 38 76 CD CE AA CA 46 4D 87 56 59 08   KŒ»ð8vÍÎªÊFM‡VY.
000001D0   E7 51 F8 C7 3A 9F C4 4E 88 8A F2 1E 41 5F B7 7D   çQøÇ:ŸÄNˆŠò.A_·}
000001E0   C0 69 43 E9 05 09 30 29 40 B9 95 CB 78 EE D9 A8   ÀiCé..0)@¹•ËxîÙ¨
000001F0   91 26 2A 78 2E 38 6E 91 5A 43 E9 4F B1 A7 AA CB   '&*x.8n'ZCéO±§ªË
00000200   68 31 A7 1C 32 2C 59 16 9C 53 C8 2A 06 2D B6 0F   h1§.2,Y.œSÈ*.-¶.
00000210   9E 51 9B 53 1B D8 19 50 EE FA 61 2F 41 0F CE 84   žQ›S.Ø.Pîúa/A.Î„
00000220   27 66 F4 E8 1A F1 E7 8E 9E C8 1A D0 B9 83 C0 51   'fôè.ñçŽžÈ.Ð¹ƒÀQ
00000230   69 84 2C 8B 20 0C E8 79 FD 01 17 42 24 AA 61 CD   i„,‹ .èyý..B$ªaÍ
00000240   55 01 4E C4 D5 B2 85 52 63 03 F0 B9 1D 8D E8 A1   U.NÄÕ²…Rc.ð¹..è¡
00000250   69 A1 E4 2C F3 30 7E B1 87 F8 7B 9D AD 4D 8A 75   i¡ä,ó0~±‡ø{..MŠu
00000260   84 E2 B6 2A 66 18 03 3A FB 0A 1C 66 69 56 E8 75   „â¶*f..:û..fiVèu
00000270   AC 7B 83 B7 46 05 19 63 2B A4 8D B4 A9 27 18 22   ¬{ƒ·F..c+¤.´©'."
00000280   42 EE 8E 76 B3 D4 67 8D B4 7E 70 3C C9 51 04 10   BîŽv³Ôg.´~p<ÉQ..
00000290   08 0A 99 9B 6A 33 5C A2 DF 9F 81 3E 4A 2F 7E 7A   ..™›j3\¢ßŸ.>J/~z
000002A0   8C 48 8C B7 C8 16 D3 95 8A 83 BB 90 15 7E EC 20   ŒHŒ·È.Ó•Šƒ»..~ì
000002B0   A7 CF C5 BC 09 7E C8 19 83 ED CD 2C 42 76 1E 94   §ÏÅ¼.~È.ƒíÍ,Bv.”
000002C0   5B 77 C3 D9 C5 32 19 D3 91 6A 7A 16 F2 AE D1 94   [wÃÙÅ2.Ó'jz.ò®Ñ”
000002D0   55 23 C6 FC F6 AA FA 22 1A 8A EB CF DB E9 71 47   U#Æüöªú".ŠëÏÛéqG
000002E0   A2 CF 37 C8 33 BC C9 2D 75 1F 2F A4 E1 3D 02 A9   ¢Ï7È3¼É-u./¤á=.©
000002F0   EE F9 0A 7E 6C 3A 1D 51 D8 8A 53 E4 2D 2A A4 C1   îù.~l:.QØŠSä-*¤Á
00000300   32 38 D6 15 D4 3C 94 2A C8 1E 55 82 71 D8 63 10   28Ö.Ô<”*È.U‚qØc.
00000310   C2 A6 60 D9 6F 9B 55 63 A1 3B A7 39 DB A3 C6 00   Â¦`Ùo›Uc¡;§9Û£Æ.
00000320   87 1E E8 7F EA 8E 31 83 1C D1 1B B3 EF 3C 90 27   ‡.è.êŽ1ƒ.Ñ.³ï<.'
00000330   B0 A1 05 9A 92 30 F1 60 50 E7 25 C9 25 50 BA 37   °¡.š’0ñ`Pç%É%Pº7
00000340   9C 08 35 D6 66 A0 CB 1F CE 36 5B E1 0B 1C FE 09   œ.5Öf Ë.Î6[á..þ.
00000350   CA 41 AC D4 8C 8C E5 52 71 1F B6 11 EA 2F 5B C9   ÊA¬ÔŒŒåRq.¶.ê/[É
00000360   70 7C 63 88 7C AC 84 F0 EB A0 25 13 56 63 91 F6   p|cˆ|¬„ðë %.Vc'ö
00000370   EC 00 DF EF A1 0C 60 71 4C C4 10 82 44 D7 9A 0F   ì.ßï¡.`qLÄ.‚D×š.
00000380   04 31 09 56 59 49 CD 1F 0D 65 33 34 0E F7 99 13   .1.VYIÍ..e34.÷™.
00000390   14 F5 22 14 F1 0D CA C5 35 8D 2A 70 B6 60 AA A3   .õ".ñ.ÊÅ5.*p¶`ª£
000003A0   B6 4E 07 83 C8 83 73 25 37 42 65 87 FA 91 09 81   ¶N.ƒÈƒs%7Be‡ú'..
000003B0   04 69 6D 4A 5A F3 83 94 1D 58 E0 FE 2C E1 FD 89   .imJZóƒ”.Xàþ,áý‰
000003C0   E7 9C E4 72 5E AA CB 76 74 27 F1 99 28 98 7D 13   çœär^ªËvt'ñ™(˜}.
000003D0   E2 A5 EF 53 1A 57 B6 BA 1F 38 12 07 EF BD 0E 71   â¥ïS.W¶º.8..ï½.q
000003E0   0E 0E 5A 09 64 F8 06 93 19 47 B2 CE AF 69 00 0B   ..Z.dø.“.G²Î¯i..
000003F0   A6 20 03 E7 79 0C 0D AC FD 3B 98 6B DD 1B 90 BE   ¦ .çy..¬ý;˜kÝ..¾
00000400   97 30 3B 44 D2 43 1D 0C 04 2D 34 74 8A F8 6E FD   —0;DÒC...-4tŠøný
00000410   48 19 DF AE 9E 74 A0 F7 B5 8B CF 8C 8C 44 8E 9E   H.ß®žt ÷µ‹ÏŒŒDŽž
00000420   EA 70 EE 36 75 80 A0 2D DD F6 BB B2 CA 05 BB 03   êpî6u€ -Ýö»²Ê.».
00000430   A4 F2 A0 78 13 35 89 95 EF 6F 86 26 99 F3 66 9F   ¤ò x.5‰•ïo†&™ófŸ
00000440   FE 7F BE 14 28 20 90 F9 5C 28 B2 0C 7F 63 38 5B   þ.¾.( .ù\(².c8[
00000450   F0 5F 03 98 7D F8 87 75 9A 23 EA 3A CD 75 94 18   ð_.˜}ø‡uš#ê:Íu”.
00000460   55 C3 61 9A F8 5C 69 94 AD 1A E7 69 5B 02 9A F5   UÃašø\i”..çi[.šõ
00000470   53 B3 D0 F0 34 0F C9 F2 76 57 A8 66 B6 17 AA E5   S³Ðð4.ÉòvW¨f¶.ªå
00000480   6F 29 3E DE 02 60 0A 5C A6 19 37 D7 E7 45 B1 CE   o)>Þ.`.\¦.7×çE±Î
00000490   76 DC E4 DC DA 2B 9A 1D 16 A3 CC AE E6 64 0B B0   vÜäÜÚ+š..£Ì®æd.°
000004A0   BD 2C 27 E0 2B 17 13 88 6F 80 7E 1C 3F 66 9B 75   ½,'à+..ˆo€~.?f›u
000004B0   C0 0C 3F 9A 4C 6C 04 7B 1B 18 89 29 46 AD 25 DD   À.?šLl.{..‰)F.%Ý
```

SecurityScorecard

RECOVER-uhwuvzu-FILES.txt

Files created

checkpoints-<Filename>.uhwuvzu

RECOVER-uhwuvzu-FILES.txt.png

Processes spawned

cmd.exe /c “wmic csproduct get UUID”

cmd.exe /c “fsutil behavior set SymlinkEvaluation R2L:1”

cmd.exe /c “fsutil behavior set SymlinkEvaluation R2R:1”

cmd.exe /c “iisreset.exe /stop”

cmd.exe /c “vssadmin.exe Delete Shadows /all /quiet”

cmd.exe /c “wmic.exe Shadowcopy Delete”

cmd.exe /c “bcdedit /set {default}”

cmd.exe /c “bcdedit /set {default} recoveryenabled No”

cmd.exe /c for /F “tokens=*” %1 in (‘wevtutil.exe el’) DO wevtutil.exe cl %1

cmd.exe
/c “reg add
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
/v MaxMpxCt /d 65535 /t REG_DWORD /f”

cmd.exe /c “arp -a”