A Deep Dive Into Black Basta Ransomware

securityscorecard.com/research/a-deep-dive-into-black-basta-ransomware

Skip to main content

- Support
- Login
- Contact
- Blog
- Languages
    - English
    - Français
    - 日本語
- Request a Demo

Interested in reading the report later? Download it.

Download Now

Prepared by: Vlad Pasca, Senior Malware & Threat Analyst

Executive summary

Black Basta ransomware is a recent threat that compiled its first malware samples in February 2022. The ransomware deletes all Volume Shadow Copies, creates a new JPG image set as the Desktop Wallpaper and an ICO file representing the encrypted files. Unlike

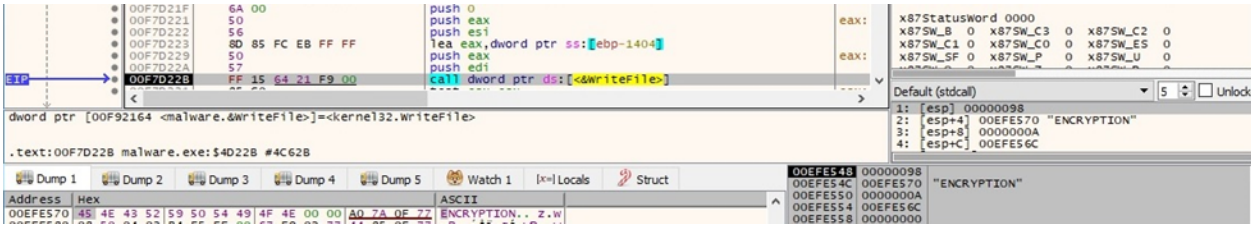
other ransomware families, the malware doesn’t skip files based on their extensions. However, it doesn’t encrypt critical folders that would make the system inoperable.

The files are encrypted using the ChaCha20 algorithm, with the key and nonce being encrypted using the RSA public key that is hard-coded in the sample. The malware can fully or partially encrypt a file depending on its size. The extension of the encrypted files is changed to .basta by the ransomware.

Analysis and findings

SHA256: ae7c868713e1d02b4db60128c651eb1e3f6a33c02544cc4cb57c3aa6c6581b6e

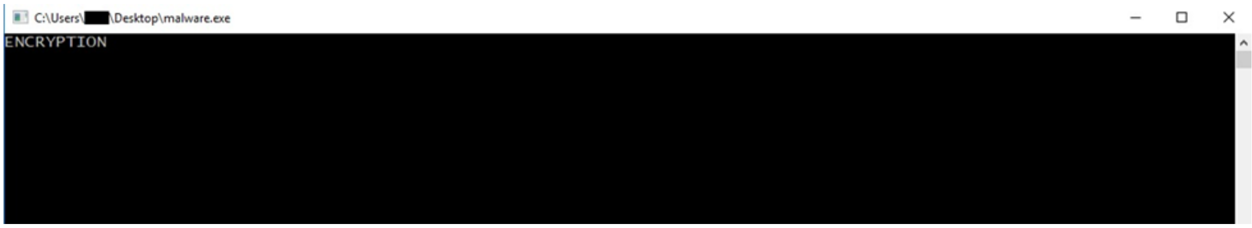
The process displays "ENCRYPTION" in the program window using WriteFile:



```
00F7D21F   6A 00              push 0
00F7D221   50                 push eax                              eax:
00F7D222   56                 push esi
00F7D223   8D 85 FC EB FF FF  lea eax,dword ptr ss:[ebp-1404]
00F7D229   50                 push eax                              eax:
00F7D22A   57                 push edi
EIP 00F7D22B   FF 15 64 21 F9 00  call dword ptr ds:[<&WriteFile>]

dword ptr [00F92164 <malware.&WriteFile>]=<kernel32.WriteFile>

.text:00F7D22B malware.exe:$4D22B #4C62B

Address   Hex                                           ASCII
00EFE570  45 4E 43 52 59 50 54 49 4F 4E 00 00 A0 7A 0F 77  ENCRYPTION.. z.w
```

```
x87StatusWord 0000
x87SW_B   0  x87SW_C3  0  x87SW_C2  0
x87SW_C1  0  x87SW_C0  0  x87SW_ES  0
x87SW_SF  0  x87SW_P   0  x87SW_U   0

Default (stdcall)   5   □ Unlock
1: [esp] 00000098
2: [esp+4] 00EFE570 "ENCRYPTION"
3: [esp+8] 0000000A
4: [esp+C] 00EFE56C

00EFE548  00000098
00EFE54C  00EFE570  "ENCRYPTION"
00EFE550  0000000A
00EFE554  00EFE56C
00EFE558  00000000
```

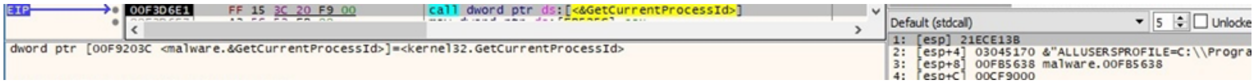
Figure 1



```
C:\Users\[redacted]\Desktop\malware.exe
ENCRYPTION
```

Figure 2

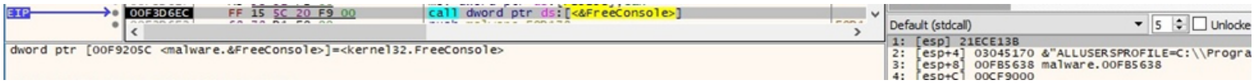
The binary retrieves the process ID via a function call to GetCurrentProcessId:



```
EIP 00F3D6E1   FF 15 3C 20 F9 00   call dword ptr ds:[<&GetCurrentProcessId>]

dword ptr [00F9203C <malware.&GetCurrentProcessId>]=<kernel32.GetCurrentProcessId>

Default (stdcall)   5   □ Unlocke
1: [esp] 21ECE13B
2: [esp+4] 03045170 &"ALLUSERSPROFILE=C:\\Progra
3: [esp+8] 00FB5638 malware.00FB5638
4: [esp+C] 00CF9000
```

Figure 3

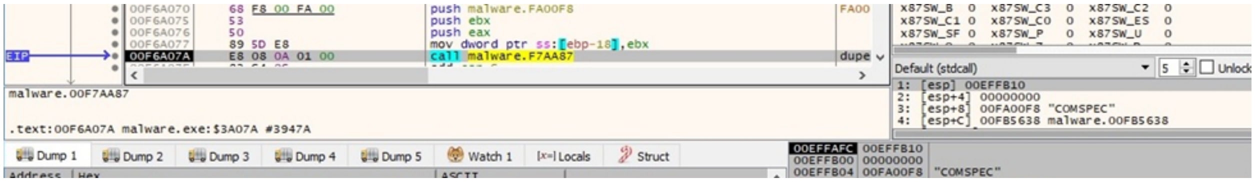
The malicious process detaches itself from its console by calling the FreeConsole API:



```
EIP 00F3D6EC   FF 15 5C 20 F9 00   call dword ptr ds:[<&FreeConsole>]

dword ptr [00F9205C <malware.&FreeConsole>]=<kernel32.FreeConsole>

Default (stdcall)   5   □ Unlocke
1: [esp] 21ECE13B
2: [esp+4] 03045170 &"ALLUSERSPROFILE=C:\\Progra
3: [esp+8] 00FB5638 malware.00FB5638
4: [esp+C] 00CF9000
```

Figure 4

The executable obtains the "COMSPEC" environment variable value, which points to the command line:

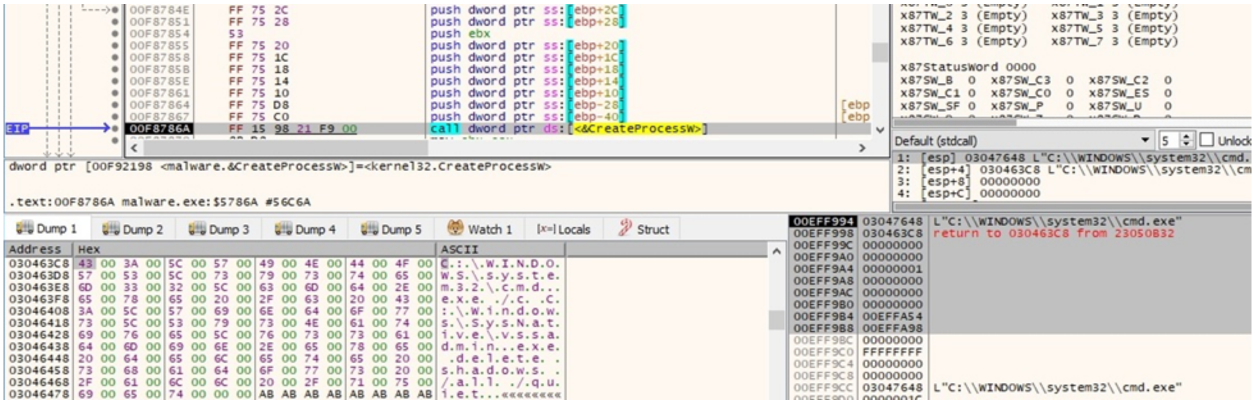


```
00F6A070   68 F8 00 FA 00   push malware.FA00F8                FA00
00F6A075   53               push ebx
00F6A076   50               push eax
00F6A077   89 5D E8         mov dword ptr ss:[ebp-18],ebx
EIP 00F6A07A   E8 08 0A 01 00   call malware.F7AA87                dupe

malware.00F7AA87

.text:00F6A07A malware.exe:$3A07A #3947A
```

```
x87SW_B   0  x87SW_C3  0  x87SW_C2  0
x87SW_C1  0  x87SW_C0  0  x87SW_ES  0
x87SW_SF  0  x87SW_P   0  x87SW_U   0

Default (stdcall)   5   □ Unlock
1: [esp] 00EFFB10
2: [esp+4] 00000000
3: [esp+8] 00FA00F8 "COMSPEC"
4: [esp+C] 00FB5638 malware.00FB5638

00EFFAFC  00EFFB10
00EFFB00  00000000
00EFFB04  00FA00F8  "COMSPEC"
```

Figure 5

The ransomware deletes all Volume Shadow Copies by running the “C:\Windows\SysNative\vssadmin.exe delete shadows /all /quiet” command, as highlighted below:



```
00F8784E   FF 75 2C           push dword ptr ss:[ebp+2C]
00F87851   FF 75 28           push dword ptr ss:[ebp+28]
00F87854   53                 push ebx
00F87855   FF 75 20           push dword ptr ss:[ebp+20]
00F87858   FF 75 1C           push dword ptr ss:[ebp+1C]
00F8785B   FF 75 18           push dword ptr ss:[ebp+18]
00F8785E   FF 75 14           push dword ptr ss:[ebp+14]
00F87861   FF 75 10           push dword ptr ss:[ebp+10]
00F87864   FF 75 D8           push dword ptr ss:[ebp-28]
00F87867   FF 75 C0           push dword ptr ss:[ebp-40]
00F8786A   FF 15 98 21 F9 00  call dword ptr ds:[<&CreateProcessW>]

dword ptr [00F92198 <malware.&CreateProcessW>]=<kernel32.CreateProcessW>

.text:00F8786A malware.exe:$5786A #56C6A
```

```
x87TW_2 3 (Empty)   x87TW_3 3 (Empty)
x87TW_4 3 (Empty)   x87TW_5 3 (Empty)
x87TW_6 3 (Empty)   x87TW_7 3 (Empty)

x87StatusWord 0000
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   0  x87SW_U   0
```

Default (stdcall) 5 □ Unlock

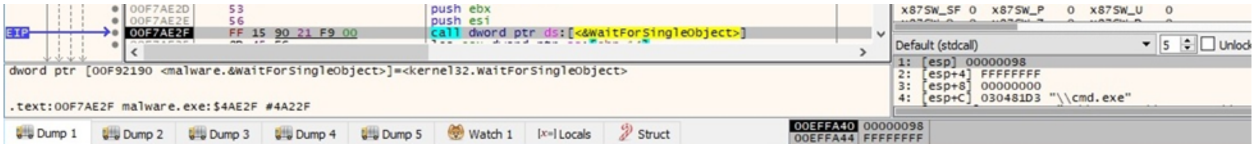
```
1: [esp] 03047648 L"C:\\WINDOWS\\system32\\cmd.
2: [esp+4] 030463C8 L"C:\\WINDOWS\\system32\\cm
3: [esp+8] 00000000
4: [esp+C] 00000000
```

```
Address  Hex                                              ASCII
030463C8 43 00 3A 00 5C 00 57 00 49 00 4E 00 44 00 4F 00  C.:.\.W.I.N.D.O.
030463D8 57 00 53 00 5C 00 73 00 79 00 73 00 74 00 65 00  W.S.\.s.y.s.t.e.
030463E8 6D 00 33 00 32 00 5C 00 63 00 6D 00 64 00 2E 00  m.3.2.\.c.m.d...
030463F8 65 00 78 00 65 00 20 00 2F 00 63 00 20 00 43 00  e.x.e. ./.c. .C.
03046408 3A 00 5C 00 57 00 69 00 6E 00 64 00 6F 00 77 00  :.\.W.i.n.d.o.w.
03046418 73 00 5C 00 53 00 79 00 73 00 4E 00 61 00 74 00  s.\.S.y.s.N.a.t.
03046428 69 00 76 00 65 00 5C 00 76 00 73 00 73 00 61 00  i.v.e.\.v.s.s.a.
03046438 64 00 6D 00 69 00 6E 00 2E 00 65 00 78 00 65 00  d.m.i.n...e.x.e.
03046448 20 00 64 00 65 00 6C 00 65 00 74 00 65 00 20 00   .d.e.l.e.t.e. .
03046458 73 00 68 00 61 00 64 00 6F 00 77 00 73 00 20 00  s.h.a.d.o.w.s. .
03046468 2F 00 61 00 6C 00 6C 00 20 00 2F 00 71 00 75 00  /.a.l.l. ./.q.u.
03046478 69 00 65 00 74 00 00 00 AB AB AB AB AB AB AB AB  i.e.t...««««««««
```

```
00EFF994 03047648 L"C:\\WINDOWS\\system32\\cmd.exe"
00EFF998 030463C8 return to 030463C8 from 23050B32
00EFF99C 00000000
00EFF9A0 00000000
00EFF9A4 00000001
00EFF9A8 00000000
00EFF9AC 00000000
00EFF9B0 00000000
00EFF9B4 00EFFA54
00EFF9B8 00EFFA98
00EFF9BC 00000000
00EFF9C0 FFFFFFFF
00EFF9C4 00000000
00EFF9C8 00000000
00EFF9CC 03047648 L"C:\\WINDOWS\\system32\\cmd.exe"
```

Figure 6

The sample waits until the spawned process finishes using the WaitForSingleObject routine:



```
00F7AE2D   53                 push ebx
00F7AE2E   56                 push esi
00F7AE2F   FF 15 90 21 F9 00  call dword ptr ds:[<&WaitForSingleObject>]

dword ptr [00F92190 <malware.&WaitForSingleObject>]=<kernel32.WaitForSingleObject>

.text:00F7AE2F malware.exe:$4AE2F #4A22F
```

```
x87SW_SF 0  x87SW_P   0  x87SW_U   0
```

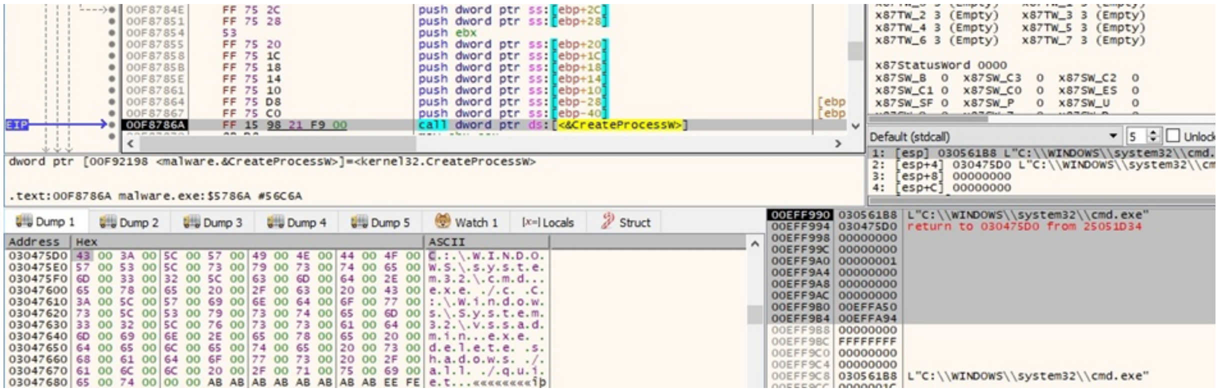
Default (stdcall) 5 □ Unlock

```
1: [esp] 00000098
2: [esp+4] FFFFFFFF
3: [esp+8] 00000000
4: [esp+C] 030481D3 "\\cmd.exe"
```

```
00EFFA40 00000098
00EFFA44 FFFFFFFF
```

Figure 7

A similar process as above that deletes the Volume Shadow Copies is spawned:



```
00F8784E   FF 75 2C           push dword ptr ss:[ebp+2C]
00F87851   FF 75 28           push dword ptr ss:[ebp+28]
00F87854   53                 push ebx
00F87855   FF 75 20           push dword ptr ss:[ebp+20]
00F87858   FF 75 1C           push dword ptr ss:[ebp+1C]
00F8785B   FF 75 18           push dword ptr ss:[ebp+18]
00F8785E   FF 75 14           push dword ptr ss:[ebp+14]
00F87861   FF 75 10           push dword ptr ss:[ebp+10]
00F87864   FF 75 D8           push dword ptr ss:[ebp-28]
00F87867   FF 75 C0           push dword ptr ss:[ebp-40]
00F8786A   FF 15 98 21 F9 00  call dword ptr ds:[<&CreateProcessW>]

dword ptr [00F92198 <malware.&CreateProcessW>]=<kernel32.CreateProcessW>

.text:00F8786A malware.exe:$5786A #56C6A
```

```
x87TW_2 3 (Empty)   x87TW_3 3 (Empty)
x87TW_4 3 (Empty)   x87TW_5 3 (Empty)
x87TW_6 3 (Empty)   x87TW_7 3 (Empty)

x87StatusWord 0000
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   0  x87SW_U   0
```

Default (stdcall) 5 □ Unlock

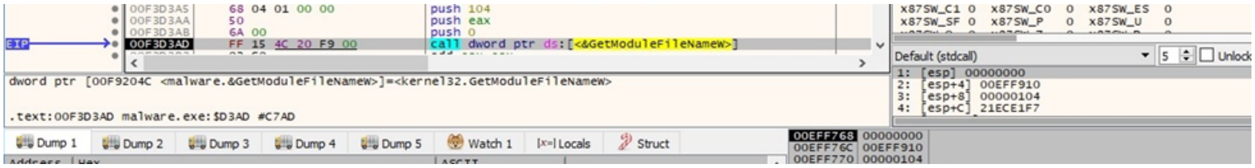
```
1: [esp] 030561B8 L"C:\\WINDOWS\\system32\\cmd.
2: [esp+4] 030475D0 L"C:\\WINDOWS\\system32\\cm
3: [esp+8] 00000000
4: [esp+C] 00000000
```

```
Address  Hex                                              ASCII
030475D0 43 00 3A 00 5C 00 57 00 49 00 4E 00 44 00 4F 00  C.:.\.W.I.N.D.O.
030475E0 57 00 53 00 5C 00 73 00 79 00 73 00 74 00 65 00  W.S.\.s.y.s.t.e.
030475F0 6D 00 33 00 32 00 5C 00 63 00 6D 00 64 00 2E 00  m.3.2.\.c.m.d...
03047600 65 00 78 00 65 00 20 00 2F 00 63 00 20 00 43 00  e.x.e. ./.c. .C.
03047610 3A 00 5C 00 57 00 69 00 6E 00 64 00 6F 00 77 00  :.\.W.i.n.d.o.w.
03047620 73 00 5C 00 53 00 79 00 73 00 74 00 65 00 6D 00  s.\.S.y.s.t.e.m.
03047630 33 00 32 00 5C 00 76 00 73 00 73 00 61 00 64 00  3.2.\.v.s.s.a.d.
03047640 6D 00 69 00 6E 00 2E 00 65 00 78 00 65 00 20 00  m.i.n...e.x.e. .
03047650 64 00 65 00 6C 00 65 00 74 00 65 00 20 00 73 00  d.e.l.e.t.e. .s.
03047660 68 00 61 00 64 00 6F 00 77 00 73 00 20 00 2F 00  h.a.d.o.w.s. ./.
03047670 61 00 6C 00 6C 00 20 00 2F 00 71 00 75 00 69 00  a.l.l. ./.q.u.i.
03047680 65 00 74 00 00 00 AB AB AB AB AB AB AB AB EE FE  e.t...««««««««îþ
```

```
00EFF990 030561B8 L"C:\\WINDOWS\\system32\\cmd.exe"
00EFF994 030475D0 return to 030475D0 from 25051D34
00EFF998 00000000
00EFF99C 00000000
00EFF9A0 00000001
00EFF9A4 00000000
00EFF9A8 00000000
00EFF9AC 00000000
00EFF9B0 00EFFA50
00EFF9B4 00EFFA94
00EFF9B8 00000000
00EFF9BC FFFFFFFF
00EFF9C0 00000000
00EFF9C4 00000000
00EFF9C8 030561B8 L"C:\\WINDOWS\\system32\\cmd.exe"
```

Figure 8

The binary extracts the path of the executable of the current process via a call to GetModuleFileNameW:



```
00F3D3A5   68 04 01 00 00     push 104
00F3D3AA   50                 push eax
00F3D3AB   6A 00              push 0
00F3D3AD   FF 15 4C 20 F9 00  call dword ptr ds:[<&GetModuleFileNameW>]

dword ptr [00F9204C <malware.&GetModuleFileNameW>]=<kernel32.GetModuleFileNameW>

.text:00F3D3AD malware.exe:$D3AD #C7AD
```

```
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   0  x87SW_U   0
```

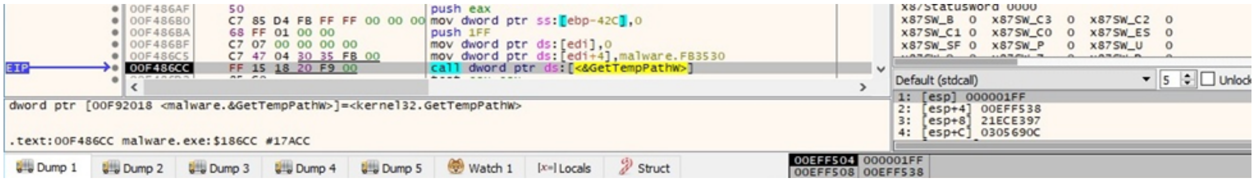
Default (stdcall) 5 □ Unlock

```
1: [esp] 00000000
2: [esp+4] 00EFF910
3: [esp+8] 00000104
4: [esp+C] 21ECE1F7
```

```
00EFF768 00000000
00EFF76C 00EFF910
00EFF770 00000104
```

Figure 9

The GetTempPathW API is utilized to retrieve the path of the Temp directory:

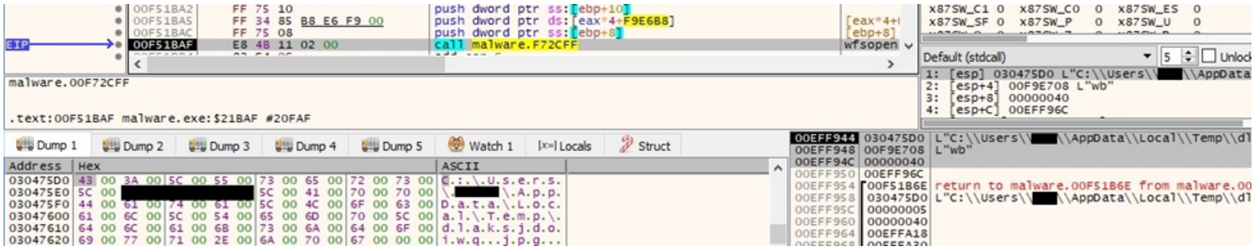


```
00F486AF   50                          push eax
00F486B0   C7 85 D4 FB FF FF 00 00 00  mov dword ptr ss:[ebp-42C],0
00F486BA   68 FF 01 00 00              push 1FF
00F486BF   C7 07 00 00 00 00           mov dword ptr ds:[edi],0
00F486C5   C7 47 04 30 35 FB 00        mov dword ptr ds:[edi+4],malware.FB3530
EIP → 00F486CC   FF 15 18 20 F9 00     call dword ptr ds:[<&GetTempPathW>]

dword ptr [00F92018 <malware.&GetTempPathW>]=<kernel32.GetTempPathW>

.text:00F486CC malware.exe:$186CC #17ACC
```

```
x87StatusWord 0000
x87SW_B   0  x87SW_C3  0  x87SW_C2  0
x87SW_C1  0  x87SW_C0  0  x87SW_ES  0
x87SW_SF  0  x87SW_P   0  x87SW_U   0

Default (stdcall)   5
1: [esp] 000001FF
2: [esp+4] 00EFF538
3: [esp+8] 21ECE397
4: [esp+C] 0305690C

00EFF504 000001FF
00EFF508 00EFF538
```

Figure 10

A file called “dlaksjdoiwq.jpg” is created in the Temp directory (0x40 = **_SH_DENYNO**):



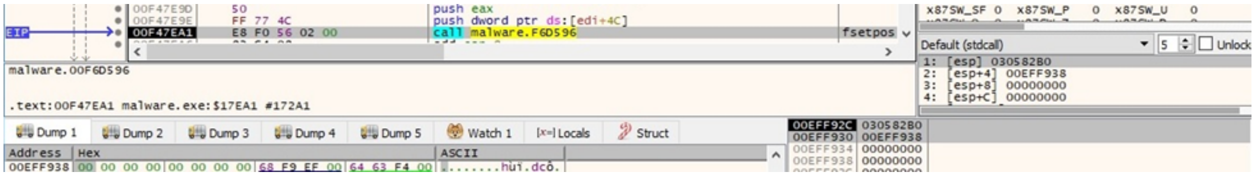
```
00F51BA2   FF 75 10          push dword ptr ss:[ebp+10]
00F51BA5   FF 34 85 B8 E6 F9 00   push dword ptr ds:[eax*4+F9E6B8]   [eax*4+
00F51BAC   FF 75 08          push dword ptr ss:[ebp+8]              [ebp+8]
EIP → 00F51BAF   E8 4B 11 02 00    call malware.F72CFF              wfsopen

malware.00F72CFF

.text:00F51BAF malware.exe:$21BAF #20FAF
```

```
Address   Hex                                              ASCII
030475D0  43 00 3A 00 5C 00 55 00 73 00 65 00 72 00 73 00  C.:.\.U.s.e.r.s.
030475E0  5C 00 [redacted] 5C 00 41 00 70 00 70 00  \.[redacted]\.A.p.p.
030475F0  44 00 61 00 74 00 61 00 5C 00 4C 00 6F 00 63 00  D.a.t.a.\.L.o.c.
03047600  61 00 6C 00 5C 00 54 00 65 00 6D 00 70 00 5C 00  a.l.\.T.e.m.p.\.
03047610  64 00 6C 00 61 00 6B 00 73 00 6A 00 64 00 6F 00  d.l.a.k.s.j.d.o.
03047620  69 00 77 00 71 00 2E 00 6A 00 70 00 67 00 00 00  i.w.q...j.p.g...
```

```
x87SW_C1  0  x87SW_C0  0  x87SW_ES  0
x87SW_SF  0  x87SW_P   0  x87SW_U   0

Default (stdcall)   5
1: [esp] 030475D0 L"C:\\Users\\[redacted]\\AppData
2: [esp+4] 00F9E708 L"wb"
3: [esp+8] 00000040
4: [esp+C] 00EFF96C

00EFF944 030475D0 L"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\dl
00EFF948 00F9E708 L"wb"
00EFF94C 00000040
00EFF950 00EFF96C
00EFF954 00F51B6E return to malware.00F51B6E from malware.00
00EFF958 030475D0 L"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\dl
00EFF95C 00000005
00EFF960 00000040
00EFF964 00EFFA18
```

Figure 11

The process moves the file position indicator to the beginning of the file using the fsetpos function:

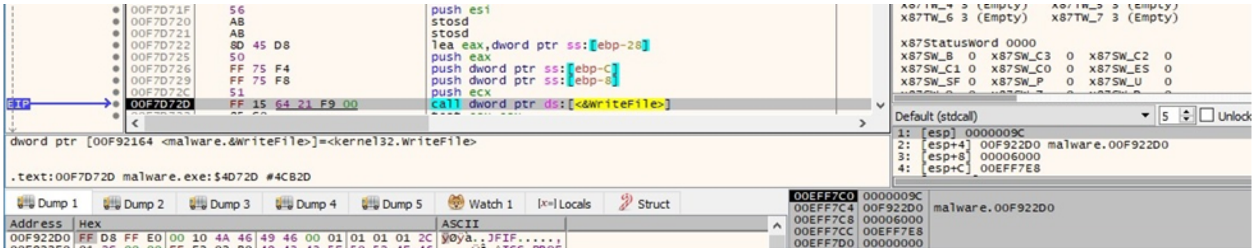


```
00F47E9D   50                push eax
00F47E9E   FF 77 4C          push dword ptr ds:[edi+4C]
EIP → 00F47EA1   E8 F0 56 02 00    call malware.F6D596           fsetpos

malware.00F6D596

.text:00F47EA1 malware.exe:$17EA1 #172A1

Address   Hex                                              ASCII
00EFF938  00 00 00 00 00 00 00 00 68 F9 EF 00 64 63 F4 00  ........hùï.dcô.
```

```
x87SW_SF  0  x87SW_P   0  x87SW_U   0

Default (stdcall)   5
1: [esp] 030582B0
2: [esp+4] 00EFF938
3: [esp+8] 00000000
4: [esp+C] 00000000

00EFF92C 030582B0
00EFF930 00EFF938
00EFF934 00000000
00EFF938 00000000
```

Figure 12

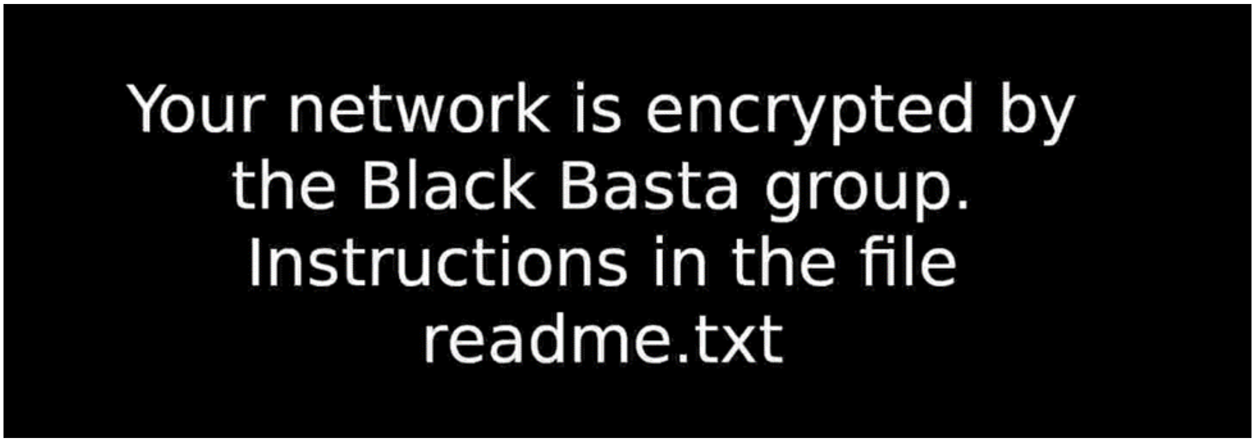
The WriteFile routine is used to populate the JPG file, which contains instructions from the threat actor:



```
00F7D71F   56                push esi
00F7D720   AB                stosd
00F7D721   AB                stosd
00F7D722   8D 45 D8          lea eax,dword ptr ss:[ebp-28]
00F7D725   50                push eax
00F7D726   FF 75 F4          push dword ptr ss:[ebp-C]
00F7D729   FF 75 F8          push dword ptr ss:[ebp-8]
00F7D72C   51                push ecx
EIP → 00F7D72D   FF 15 64 21 F9 00   call dword ptr ds:[<&WriteFile>]

dword ptr [00F92164 <malware.&WriteFile>]=<kernel32.WriteFile>

.text:00F7D72D malware.exe:$4D72D #4CB2D

Address   Hex                                              ASCII
00F922D0  FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 01 2C  ÿØÿà..JFIF.....,
```

```
x87TW_6 3 (Empty)   x87TW_7 3 (Empty)

x87StatusWord 0000
x87SW_B   0  x87SW_C3  0  x87SW_C2  0
x87SW_C1  0  x87SW_C0  0  x87SW_ES  0
x87SW_SF  0  x87SW_P   0  x87SW_U   0

Default (stdcall)   5
1: [esp] 0000009C
2: [esp+4] 00F922D0 malware.00F922D0
3: [esp+8] 00006000
4: [esp+C] 00EFF7E8

00EFF7C0 0000009C
00EFF7C4 00F922D0 malware.00F922D0
00EFF7C8 00006000
00EFF7CC 00EFF7E8
00EFF7D0 00000000
```

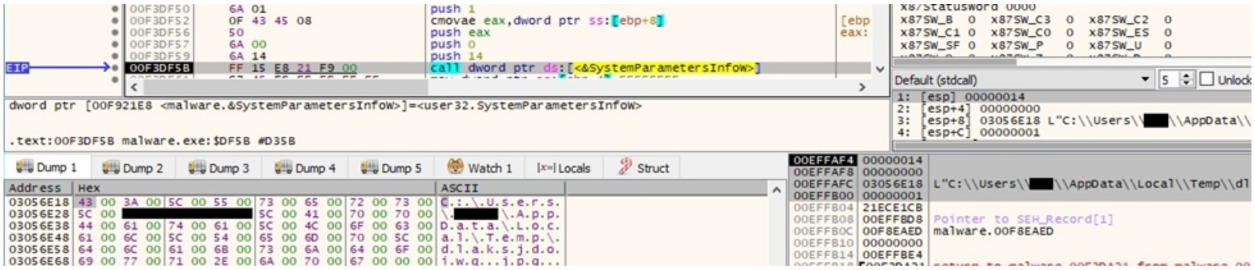
Figure 13



Your network is encrypted by the Black Basta group. Instructions in the file readme.txt

Figure 14

The newly created image is set as the Desktop Wallpaper using SystemParametersInfoW (0x14 = **SPI_SETDESKWALLPAPER**, 0x1 = **SPIF_UPDATEINIFILE**):



```
00F3DF50   6A 01               push 1
00F3DF52   0F 43 45 08         cmovae eax,dword ptr ss:[ebp+8]      [ebp
00F3DF56   50                  push eax                             eax:
00F3DF57   6A 00               push 0
00F3DF59   6A 14               push 14
00F3DF5B   FF 15 E8 21 F9 00   call dword ptr ds:[<&SystemParametersInfoW>]

dword ptr [00F921E8 <malware.&SystemParametersInfoW>]=<user32.SystemParametersInfoW>

.text:00F3DF5B malware.exe:$DF5B #D35B
```

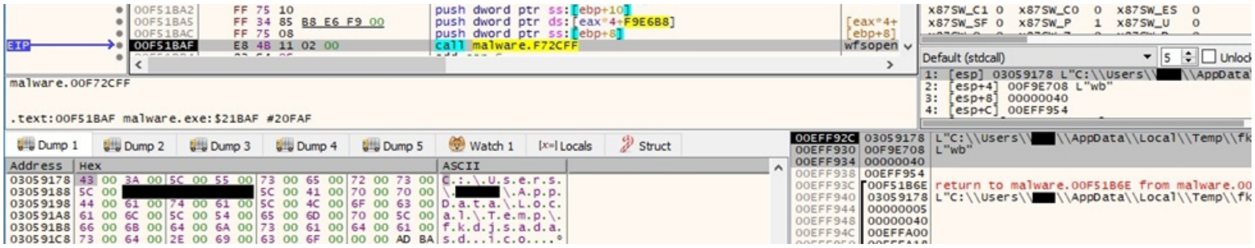
```
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   0  x87SW_U   0

1: [esp] 00000014
2: [esp+4] 00000000
3: [esp+8] 03056E18 L"C:\\Users\\[redacted]\\AppData\\
4: [esp+C] 00000001
```

```
Address   Hex                                              ASCII
03056E18  43 00 3A 00 5C 00 55 00 73 00 65 00 72 00 73 00  C.:.\.U.s.e.r.s.
03056E28  5C 00 [redacted] 5C 00 41 00 70 00 70 00  \.[redacted]\.A.p.p.
03056E38  44 00 61 00 74 00 61 00 5C 00 4C 00 6F 00 63 00  D.a.t.a.\.L.o.c.
03056E48  61 00 6C 00 5C 00 54 00 65 00 6D 00 70 00 5C 00  a.l.\.T.e.m.p.\.
03056E58  64 00 6C 00 61 00 6B 00 73 00 6A 00 64 00 6F 00  d.l.a.k.s.j.d.o.
03056E68  69 00 77 00 71 00 2E 00 6A 00 70 00 67 00 00 00  i.w.q...j.p.g...
```

```
00EFFAF4  00000014
00EFFAF8  00000000
00EFFAFC  03056E18  L"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\dl
00EFFB00  00000001
00EFFB04  21ECE1CB
00EFFB08  00EFFBD8  Pointer to SEH_Record[1]
00EFFB0C  00F8EAED  malware.00F8EAED
00EFFB10  00000000
00EFFB14  00EFFBE4
```

Figure 15

The executable creates an ICO file called “fkdjsadasd.ico” in the Temp directory:



```
00F51BA2   FF 75 10            push dword ptr ss:[ebp+10]
00F51BA5   FF 34 85 B8 E6 F9 00   push dword ptr ds:[eax*4+F9E6B8]   [eax*4+
00F51BAC   FF 75 08            push dword ptr ss:[ebp+8]            [ebp+8]
00F51BAF   E8 4B 11 02 00      call malware.F72CFF                  wfsopen

malware.00F72CFF

.text:00F51BAF malware.exe:$21BAF #20FAF
```

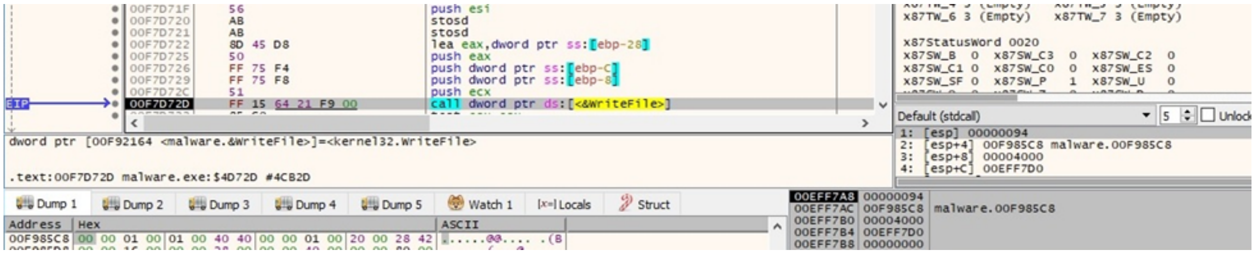
```
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

1: [esp] 03059178 L"C:\\Users\\[redacted]\\AppData
2: [esp+4] 00F9E708 L"wb"
3: [esp+8] 00000040
4: [esp+C] 00EFF954
```

```
Address   Hex                                              ASCII
03059178  43 00 3A 00 5C 00 55 00 73 00 65 00 72 00 73 00  C.:.\.U.s.e.r.s.
03059188  5C 00 [redacted] 5C 00 41 00 70 00 70 00  \.[redacted]\.A.p.p.
03059198  44 00 61 00 74 00 61 00 5C 00 4C 00 6F 00 63 00  D.a.t.a.\.L.o.c.
030591A8  61 00 6C 00 5C 00 54 00 65 00 6D 00 70 00 5C 00  a.l.\.T.e.m.p.\.
030591B8  66 00 6B 00 64 00 6A 00 73 00 61 00 64 00 61 00  f.k.d.j.s.a.d.a.
030591C8  73 00 64 00 2E 00 69 00 63 00 6F 00 00 00 AD BA  s.d...i.c.o....°
```

```
00EFF92C  03059178  L"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\fk
00EFF930  00F9E708  L"wb"
00EFF934  00000040
00EFF938  00EFF954
00EFF93C  00F51B6E  return to malware.00F51B6E from malware.00
00EFF940  03059178  L"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\fk
00EFF944  00000005
00EFF948  00000040
00EFF94C  00EFFA00
```

Figure 16

The ransomware writes content to the ICO file, which will represent the icon of the encrypted files:



```
00F7D71F   56                  push esi
00F7D720   AB                  stosd
00F7D721   AB                  stosd
00F7D722   8D 45 D8            lea eax,dword ptr ss:[ebp-28]
00F7D725   50                  push eax
00F7D726   FF 75 F4            push dword ptr ss:[ebp-C]
00F7D729   FF 75 F8            push dword ptr ss:[ebp-8]
00F7D72C   51                  push ecx
00F7D72D   FF 15 64 21 F9 00   call dword ptr ds:[<&WriteFile>]

dword ptr [00F92164 <malware.&WriteFile>]=<kernel32.WriteFile>

.text:00F7D72D malware.exe:$4D72D #4CB2D
```

```
x87TW_6 3 (Empty)   x87TW_7 3 (Empty)

x87StatusWord 0020
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

1: [esp] 00000094
2: [esp+4] 00F985C8 malware.00F985C8
3: [esp+8] 00004000
4: [esp+C] 00EFF7D0
```

```
Address   Hex                                              ASCII
00F985C8  00 00 01 00 01 00 40 40 00 00 01 00 20 00 28 42  ......@@.... .(B
```

```
00EFF7A8  00000094
00EFF7AC  00F985C8  malware.00F985C8
00EFF7B0  00004000
00EFF7B4  00EFF7D0
00EFF7B8  00000000
```

Figure 17

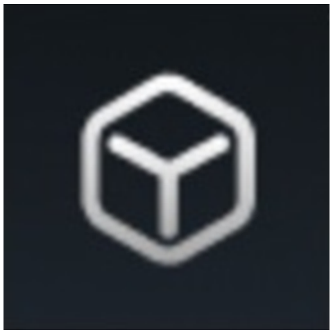
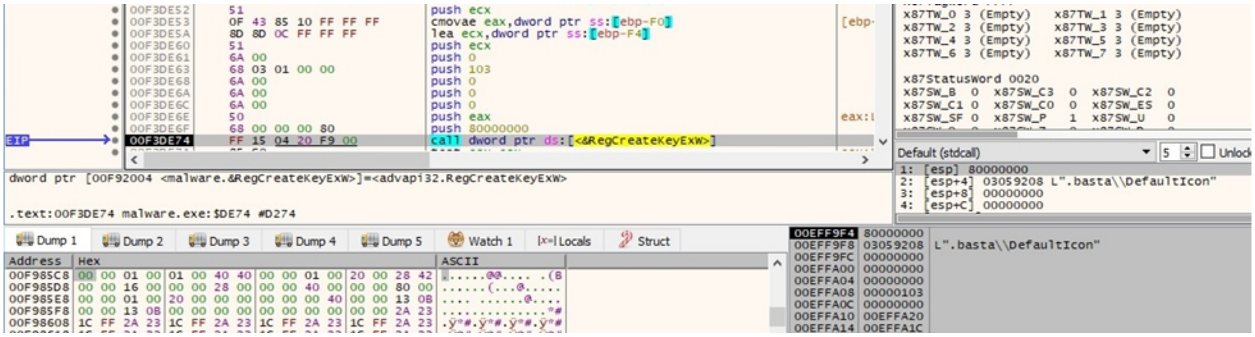


Figure 18

Black Basta ransomware creates the ".basta\DefaultIcon" registry key using RegCreateKeyExW (0x80000000 = **HKEY_CLASSES_ROOT**, 0x103 = **KEY_WOW64_64KEY | KEY_SET_VALUE | KEY_QUERY_VALUE**):



```
00F3DE52   51                      push ecx
00F3DE53   0F 43 85 10 FF FF FF    cmovae eax,dword ptr ss:[ebp-F0]
00F3DE5A   8D 8D 0C FF FF FF       lea ecx,dword ptr ss:[ebp-F4]
00F3DE60   51                      push ecx
00F3DE61   6A 00                   push 0
00F3DE63   68 03 01 00 00          push 103
00F3DE68   6A 00                   push 0
00F3DE6A   6A 00                   push 0
00F3DE6C   6A 00                   push 0
00F3DE6E   50                      push eax
00F3DE6F   68 00 00 00 80          push 80000000
00F3DE74   FF 15 04 20 F9 00       call dword ptr ds:[<&RegCreateKeyExW>]

dword ptr [00F92004 <malware.&RegCreateKeyExW>]=<advapi32.RegCreateKeyExW>

.text:00F3DE74 malware.exe:$DE74 #D274
```

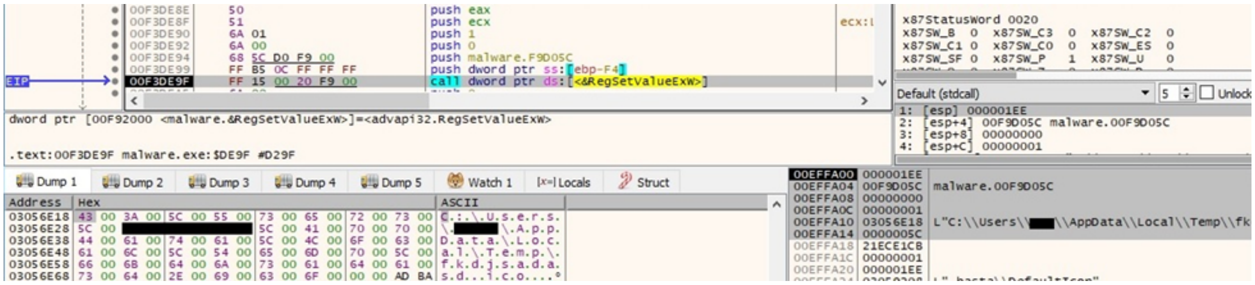
```
x87TW_0 3 (Empty)   x87TW_1 3 (Empty)
x87TW_2 3 (Empty)   x87TW_3 3 (Empty)
x87TW_4 3 (Empty)   x87TW_5 3 (Empty)
x87TW_6 3 (Empty)   x87TW_7 3 (Empty)

x87StatusWord 0020
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)
1: [esp] 80000000
2: [esp+4] 03059208 L".basta\\DefaultIcon"
3: [esp+8] 00000000
4: [esp+C] 00000000
```

```
Address  Hex                                              ASCII
00F985C8 00 00 01 00 01 00 40 40 00 00 01 00 20 00 28 42 ......@@.... .(B
00F985D8 00 00 16 00 00 00 28 00 00 00 40 00 00 00 80 00 ......(...@.....
00F985E8 00 00 01 00 20 00 00 00 00 00 00 40 00 00 13 0B .... ......@....
00F985F8 00 00 13 0B 00 00 00 00 00 00 00 00 00 00 2A 23 ..............*#
00F98608 1C FF 2A 23 1C FF 2A 23 1C FF 2A 23 1C FF 2A 23 .ÿ*#.ÿ*#.ÿ*#.ÿ*#
```

```
00EFF9F4 80000000
00EFF9F8 03059208 L".basta\\DefaultIcon"
00EFF9FC 00000000
00EFFA00 00000000
00EFFA04 00000000
00EFFA08 00000103
00EFFA0C 00000000
00EFFA10 00EFFA20
00EFFA14 00EFFA1C
```

Figure 19

The “(Default)” value of the above key is set to the path of the ICO file:



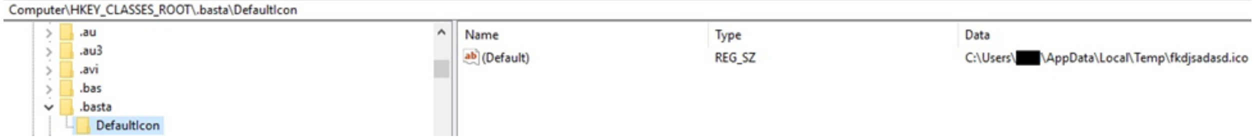
```
00F3DE8E   50                      push eax
00F3DE8F   51                      push ecx
00F3DE90   6A 01                   push 1
00F3DE92   6A 00                   push 0
00F3DE94   68 5C D0 F9 00          push malware.F9D05C
00F3DE99   FF B5 0C FF FF FF       push dword ptr ss:[ebp-F4]
00F3DE9F   FF 15 00 20 F9 00       call dword ptr ds:[<&RegSetValueExW>]

dword ptr [00F92000 <malware.&RegSetValueExW>]=<advapi32.RegSetValueExW>

.text:00F3DE9F malware.exe:$DE9F #D29F
```

```
x87StatusWord 0020
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)
1: [esp] 000001EE
2: [esp+4] 00F9D05C malware.00F9D05C
3: [esp+8] 00000000
4: [esp+C] 00000001
```

```
Address  Hex                                              ASCII
03056E18 43 00 3A 00 5C 00 55 00 73 00 65 00 72 00 73 00 C.:.\.U.s.e.r.s.
03056E28 5C 00 [redacted] 5C 00 41 00 70 00 70 00 \.[redacted]\.A.p.p.
03056E38 44 00 61 00 74 00 61 00 5C 00 4C 00 6F 00 63 00 D.a.t.a.\.L.o.c.
03056E48 61 00 6C 00 5C 00 54 00 65 00 6D 00 70 00 5C 00 a.l.\.T.e.m.p.\.
03056E58 66 00 6B 00 64 00 6A 00 73 00 61 00 64 00 61 00 f.k.d.j.s.a.d.a.
03056E68 73 00 64 00 2E 00 69 00 63 00 6F 00 00 00 AD BA s.d...i.c.o....°
```

```
00EFFA00 000001EE
00EFFA04 00F9D05C malware.00F9D05C
00EFFA08 00000000
00EFFA0C 00000001
00EFFA10 03056E18 L"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\fk
00EFFA14 0000005C
00EFFA18 21ECE1CB
00EFFA1C 00000001
00EFFA20 000001EE
```

Figure 20



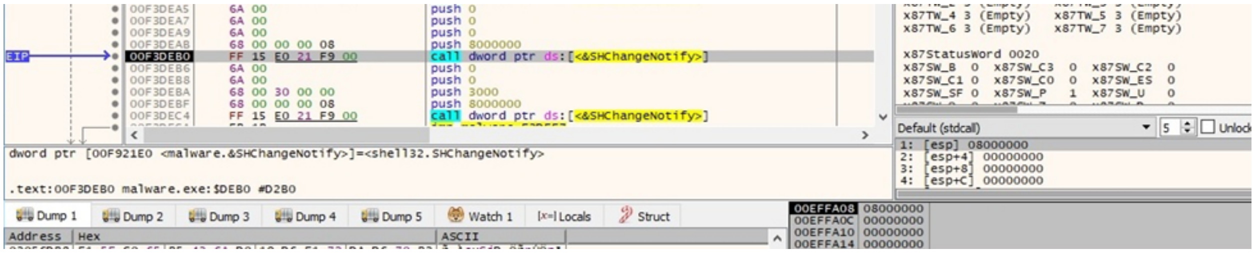
Computer\HKEY_CLASSES_ROOT\.basta\DefaultIcon

- .au
- .au3
- .avi
- .bas
- .basta
  - DefaultIcon

Name	Type	Data
(Default)	REG_SZ	C:\Users\[redacted]\AppData\Local\Temp\fkdjsadasd.ico

Figure 21

The malicious binary notifies the system that the icon has been changed by calling the SHChangeNotify function (0x08000000 = **SHCNE_ASSOCCHANGED**, 0x3000 = **SHCNF_FLUSHNOWAIT**):



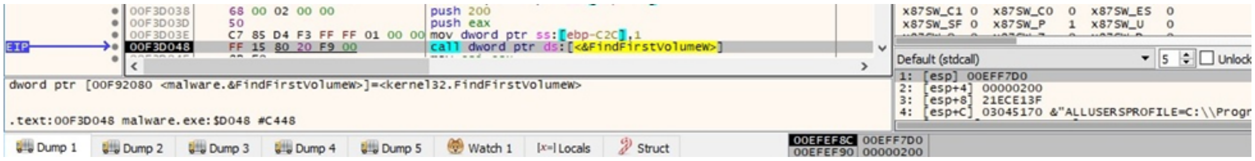
```
00F3DEA5   6A 00                   push 0
00F3DEA7   6A 00                   push 0
00F3DEA9   6A 00                   push 0
00F3DEAB   68 00 00 00 08          push 8000000
00F3DEB0   FF 15 E0 21 F9 00       call dword ptr ds:[<&SHChangeNotify>]
00F3DEB6   6A 00                   push 0
00F3DEB8   6A 00                   push 0
00F3DEBA   68 00 30 00 00          push 3000
00F3DEBF   68 00 00 00 08          push 8000000
00F3DEC4   FF 15 E0 21 F9 00       call dword ptr ds:[<&SHChangeNotify>]

dword ptr [00F921E0 <malware.&SHChangeNotify>]=<shell32.SHChangeNotify>

.text:00F3DEB0 malware.exe:$DEB0 #D2B0
```

```
x87TW_4 3 (Empty)   x87TW_5 3 (Empty)
x87TW_6 3 (Empty)   x87TW_7 3 (Empty)

x87StatusWord 0020
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)
1: [esp] 08000000
2: [esp+4] 00000000
3: [esp+8] 00000000
4: [esp+C] 00000000
```

```
00EFFA08 08000000
00EFFA0C 00000000
00EFFA10 00000000
00EFFA14 00000000
```

Figure 22

The malware starts scanning for volumes on the system using FindFirstVolumeW:



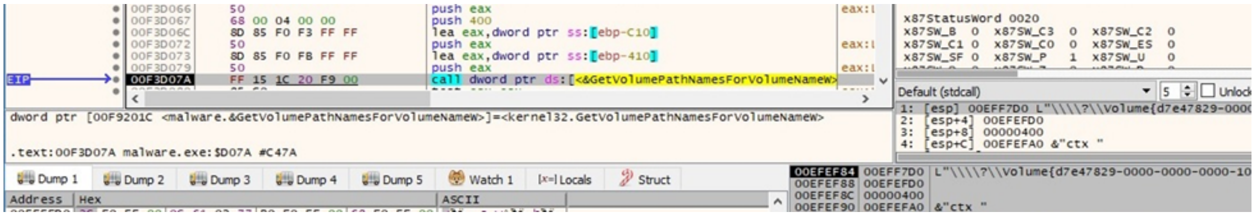
```
00F3D038   68 00 02 00 00              push 200
00F3D03D   50                          push eax
00F3D03E   C7 85 D4 F3 FF FF 01 00 00  mov dword ptr ss:[ebp-C2C],1
00F3D048   FF 15 80 20 F9 00           call dword ptr ds:[<&FindFirstVolumeW>]

dword ptr [00F92080 <malware.&FindFirstVolumeW>]=<kernel32.FindFirstVolumeW>

.text:00F3D048 malware.exe:$D048 #C448
```

```
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)
1: [esp] 00EFF7D0
2: [esp+4] 00000200
3: [esp+8] 21ECE13F
4: [esp+C] 03045170 &"ALLUSERSPROFILE=C:\\Progr
```

```
00EFEF8C 00EFF7D0
00EFEF90 00000200
```

Figure 23

GetVolumePathNamesForVolumeNameW is utilized to obtain the list of drive letters and mounted folder paths for the volume:

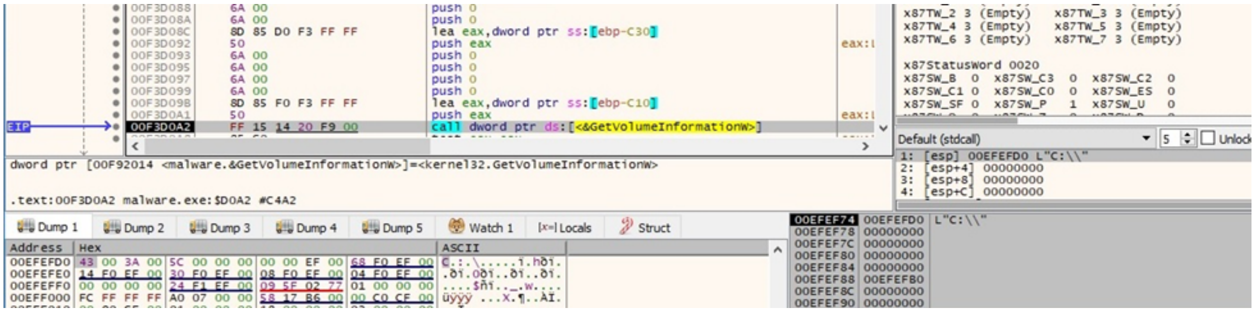


```
00F3D066   50                  push eax
00F3D067   68 00 04 00 00      push 400
00F3D06C   8D 85 F0 F3 FF FF   lea eax,dword ptr ss:[ebp-C10]
00F3D072   50                  push eax
00F3D073   8D 85 F0 FB FF FF   lea eax,dword ptr ss:[ebp-410]
00F3D079   50                  push eax
00F3D07A   FF 15 1C 20 F9 00   call dword ptr ds:[<&GetVolumePathNamesForVolumeNameW>

dword ptr [00F9201C <malware.&GetVolumePathNamesForVolumeNameW>]=<kernel32.GetVolumePathNamesForVolumeNameW>

.text:00F3D07A malware.exe:$D07A #C47A
```

```
x87StatusWord 0020
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)  5
1: [esp] 00EFF7D0 L"\\\\?\\Volume{d7e47829-000
2: [esp+4] 00EFEFD0
3: [esp+8] 00000400
4: [esp+C] 00EFEFA0 &"ctx "

00EFEF84 00EFF7D0 L"\\\\?\\Volume{d7e47829-0000-0000-0000-10
00EFEF88 00EFEFD0
00EFEF8C 00000400
00EFEF90 00EFEFA0 &"ctx "
```

Figure 24

For each drive found, the process performs a call to the GetVolumeInformationW API (see figure 25). As opposed to other ransomware families, Black Basta only targets the mounted volumes and doesn't mount the hidden volumes.



```
00F3D088   6A 00               push 0
00F3D08A   6A 00               push 0
00F3D08C   8D 85 D0 F3 FF FF   lea eax,dword ptr ss:[ebp-C30]
00F3D092   50                  push eax
00F3D093   6A 00               push 0
00F3D095   6A 00               push 0
00F3D097   6A 00               push 0
00F3D099   6A 00               push 0
00F3D09B   8D 85 F0 F3 FF FF   lea eax,dword ptr ss:[ebp-C10]
00F3D0A1   50                  push eax
00F3D0A2   FF 15 14 20 F9 00   call dword ptr ds:[<&GetVolumeInformationW>]

dword ptr [00F92014 <malware.&GetVolumeInformationW>]=<kernel32.GetVolumeInformationW>

.text:00F3D0A2 malware.exe:$D0A2 #C4A2
```

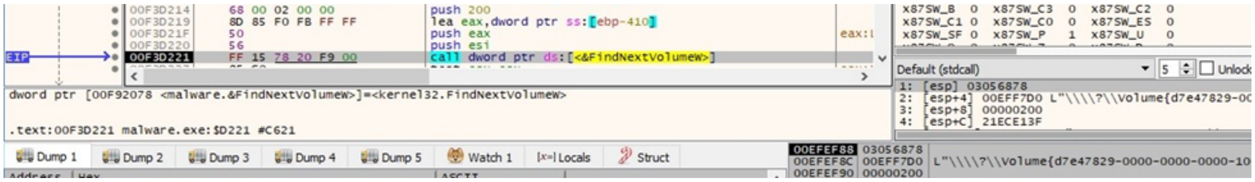
```
x87TW_2 3 (Empty)   x87TW_3 3 (Empty)
x87TW_4 3 (Empty)   x87TW_5 3 (Empty)
x87TW_6 3 (Empty)   x87TW_7 3 (Empty)

x87StatusWord 0020
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)  5
1: [esp] 00EFEFD0 L"C:\\"
2: [esp+4] 00000000
3: [esp+8] 00000000
4: [esp+C] 00000000
```

```
Address   Hex                                              ASCII
00EFEFD0  43 00 3A 00 5C 00 00 00 00 00 EF 00 68 F0 EF 00  C.:.\.....ï.hðï.
00EFEFE0  14 F0 EF 00 30 F0 EF 00 08 F0 EF 00 04 F0 EF 00  .ðï.0ðï..ðï..ðï.
00EFEFF0  00 00 00 00 24 F1 EF 00 09 5F 02 77 01 00 00 00  ....$ñï.._.w....
00EFF000  FC FF FF FF A0 07 00 00 58 17 B6 00 00 C0 CF 00  üÿÿÿ ...X.¶..ÀÏ.
```

```
00EFEF74 00EFEFD0 L"C:\\"
00EFEF78 00000000
00EFEF7C 00000000
00EFEF80 00000000
00EFEF84 00000000
00EFEF88 00EFEFB0
00EFEF8C 00000000
00EFEF90 00000000
```

Figure 25

The volume's enumeration continues by calling the FindNextVolumeW routine:

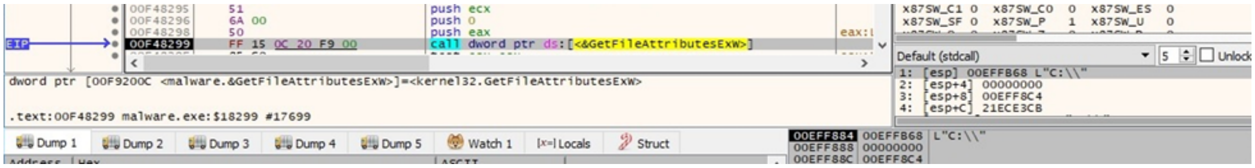


```
00F3D214   68 00 02 00 00      push 200
00F3D219   8D 85 F0 FB FF FF   lea eax,dword ptr ss:[ebp-410]
00F3D21F   50                  push eax
00F3D220   56                  push esi
00F3D221   FF 15 78 20 F9 00   call dword ptr ds:[<&FindNextVolumeW>]

dword ptr [00F92078 <malware.&FindNextVolumeW>]=<kernel32.FindNextVolumeW>

.text:00F3D221 malware.exe:$D221 #C621
```

```
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)  5
1: [esp] 03056878
2: [esp+4] 00EFF7D0 L"\\\\?\\Volume{d7e47829-0C
3: [esp+8] 00000200
4: [esp+C] 21ECE13F

00EFEF88 03056878
00EFEF8C 00EFF7D0 L"\\\\?\\Volume{d7e47829-0000-0000-0000-10
00EFEF90 00000200
```

Figure 26

The ransomware extracts a standard set of attribute information from the drives found via a function call to GetFileAttributesExW (0x0 = **GetFileExInfoStandard**):

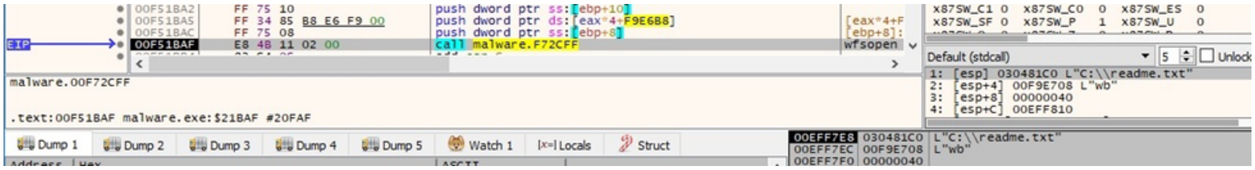


```
00F48295   51                  push ecx
00F48296   6A 00               push 0
00F48298   50                  push eax
00F48299   FF 15 0C 20 F9 00   call dword ptr ds:[<&GetFileAttributesExW>]

dword ptr [00F9200C <malware.&GetFileAttributesExW>]=<kernel32.GetFileAttributesExW>

.text:00F48299 malware.exe:$18299 #17699
```

```
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)  5
1: [esp] 00EFFB68 L"C:\\"
2: [esp+4] 00000000
3: [esp+8] 00EFF8C4
4: [esp+C] 21ECE3CB

00EFF884 00EFFB68 L"C:\\"
00EFF888 00000000
00EFF88C 00EFF8C4
```

Figure 27

The ransomware creates a ransom note called "readme.txt" in every directory that is traversed, as highlighted in figure 28:

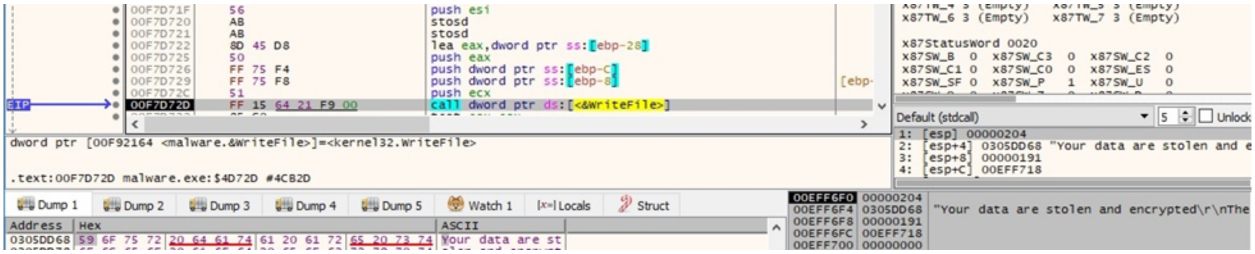


```
00F51BA2   FF 75 10            push dword ptr ss:[ebp+10]
00F51BA5   FF 34 85 B8 E6 F9 00   push dword ptr ds:[eax*4+F9E6B8]   [eax*4+F
00F51BAC   FF 75 08            push dword ptr ss:[ebp+8]          [ebp+8]:
00F51BAF   E8 4B 11 02 00      call malware.F72CFF                wfsopen

malware.00F72CFF

.text:00F51BAF malware.exe:$21BAF #20FAF
```

```
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)  5
1: [esp] 030481C0 L"C:\\readme.txt"
2: [esp+4] 00F9E708 L"wb"
3: [esp+8] 00000040
4: [esp+C] 00EFF810

00EFF7E8 030481C0 L"C:\\readme.txt"
00EFF7EC 00F9E708 L"wb"
00EFF7F0 00000040
```

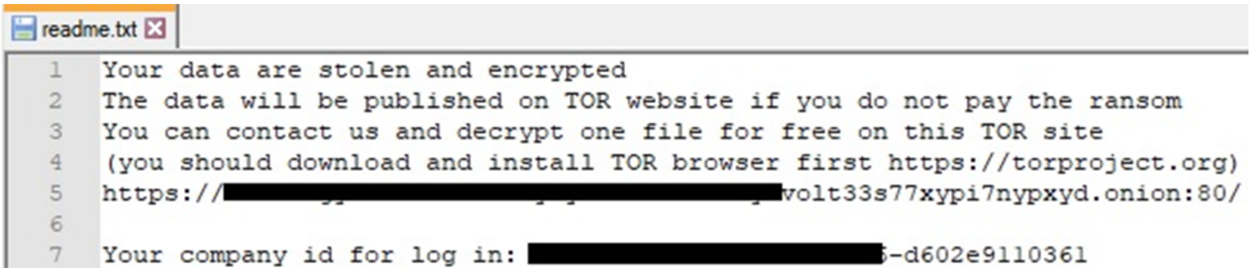
Figure 28

WriteFile is used to populate the ransom note:



```
00F7D71F   56                 push esi
00F7D720   AB                 stosd
00F7D721   AB                 stosd
00F7D722   8D 45 D8           lea eax,dword ptr ss:[ebp-28]
00F7D725   50                 push eax
00F7D726   FF 75 F4           push dword ptr ss:[ebp-C]
00F7D729   FF 75 F8           push dword ptr ss:[ebp-8]
00F7D72C   51                 push ecx
00F7D72D   FF 15 64 21 F9 00  call dword ptr ds:[<&WriteFile>]

dword ptr [00F92164 <malware.&WriteFile>]=<kernel32.WriteFile>

.text:00F7D72D malware.exe:$4D72D #4CB2D

x87TW_6 3 (Empty)   x87TW_7 3 (Empty)

x87StatusWord 0020
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)  5  Unlock
1: [esp] 00000204
2: [esp+4] 0305DD68 "Your data are stolen and e
3: [esp+8] 00000191
4: [esp+C] 00EFF718

Address   Hex                                              ASCII
0305DD68  59 6F 75 72 20 64 61 74 61 20 61 72 65 20 73 74  Your data are st

00EFF6F0 00000204
00EFF6F4 0305DD68  "Your data are stolen and encrypted\r\nThe
00EFF6F8 00000191
00EFF6FC 00EFF718
00EFF700 00000000
```

Figure 29

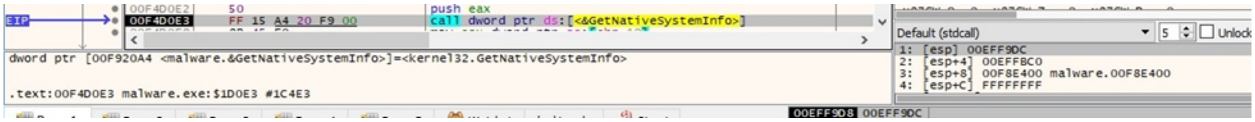


readme.txt

```
1  Your data are stolen and encrypted
2  The data will be published on TOR website if you do not pay the ransom
3  You can contact us and decrypt one file for free on this TOR site
4  (you should download and install TOR browser first https://torproject.org)
5  https://[redacted]volt33s77xypi7nypxyd.onion:80/
6
7  Your company id for log in: [redacted]-d602e9110361
```

Figure 30

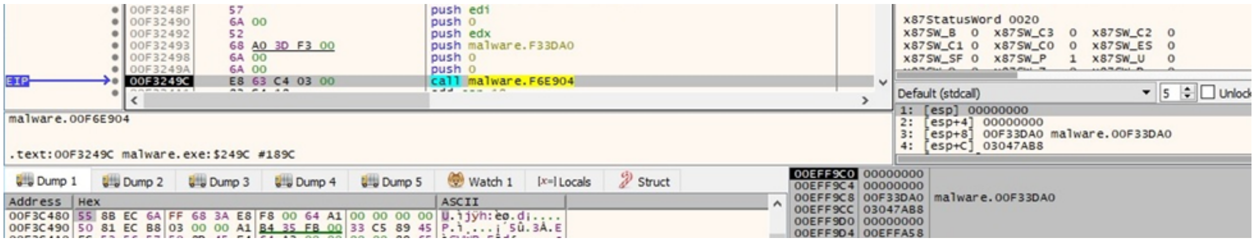
The binary retrieves information about the current system by calling the GetNativeSystemInfo function:



```
00F4D0E2   50                 push eax
00F4D0E3   FF 15 A4 20 F9 00  call dword ptr ds:[<&GetNativeSystemInfo>]

dword ptr [00F920A4 <malware.&GetNativeSystemInfo>]=<kernel32.GetNativeSystemInfo>

.text:00F4D0E3 malware.exe:$1D0E3 #1C4E3

Default (stdcall)  5  Unlock
1: [esp] 00EFF9DC
2: [esp+4] 00EFFBC0
3: [esp+8] 00F8E400 malware.00F8E400
4: [esp+C] FFFFFFFF

00EFF9D8 00EFF9DC
```

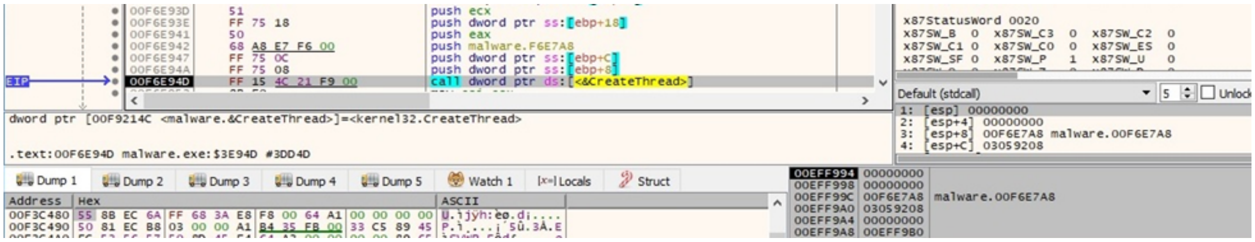
Figure 31

The malware creates multiple threads that will handle the file encryption. The function responsible for encryption is sub_F33DA0 and not the starting address of the thread:



```
00F3248F   57                 push edi
00F32490   6A 00              push 0
00F32492   52                 push edx
00F32493   68 A0 3D F3 00     push malware.F33DA0
00F32498   6A 00              push 0
00F3249A   6A 00              push 0
00F3249C   E8 63 C4 03 00     call malware.F6E904

malware.00F6E904

.text:00F3249C malware.exe:$249C #189C

x87StatusWord 0020
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)  5  Unlock
1: [esp] 00000000
2: [esp+4] 00000000
3: [esp+8] 00F33DA0 malware.00F33DA0
4: [esp+C] 03047AB8

Address   Hex                                              ASCII
00F3C480  55 8B EC 6A FF 68 3A E8 F8 00 64 A1 00 00 00 00  U.ìjÿh:èø.d¡....
00F3C490  50 81 EC B8 03 00 00 A1 B4 35 FB 00 33 C5 89 45  P.ì¸...¡´5û.3Å.E

00EFF9C0 00000000
00EFF9C4 00000000
00EFF9C8 00F33DA0  malware.00F33DA0
00EFF9CC 03047AB8
00EFF9D0 00000000
00EFF9D4 00EFFA58
```

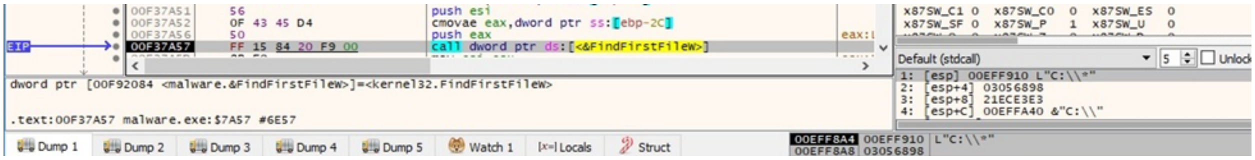
Figure 32



```
00F6E93D   51                 push ecx
00F6E93E   FF 75 18           push dword ptr ss:[ebp+18]
00F6E941   50                 push eax
00F6E942   68 A8 E7 F6 00     push malware.F6E7A8
00F6E947   FF 75 0C           push dword ptr ss:[ebp+C]
00F6E94A   FF 75 08           push dword ptr ss:[ebp+8]
00F6E94D   FF 15 4C 21 F9 00  call dword ptr ds:[<&CreateThread>]

dword ptr [00F9214C <malware.&CreateThread>]=<kernel32.CreateThread>

.text:00F6E94D malware.exe:$3E94D #3DD4D

x87StatusWord 0020
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)  5  Unlock
1: [esp] 00000000
2: [esp+4] 00000000
3: [esp+8] 00F6E7A8 malware.00F6E7A8
4: [esp+C] 03059208

Address   Hex                                              ASCII
00F3C480  55 8B EC 6A FF 68 3A E8 F8 00 64 A1 00 00 00 00  U.ìjÿh:èø.d¡....
00F3C490  50 81 EC B8 03 00 00 A1 B4 35 FB 00 33 C5 89 45  P.ì¸...¡´5û.3Å.E

00EFF994 00000000
00EFF998 00000000
00EFF99C 00F6E7A8  malware.00F6E7A8
00EFF9A0 03059208
00EFF9A4 00000000
00EFF9A8 00EFF9B0
```

Figure 33

The malicious process starts enumerating the files on the drive using FindFirstFileW:

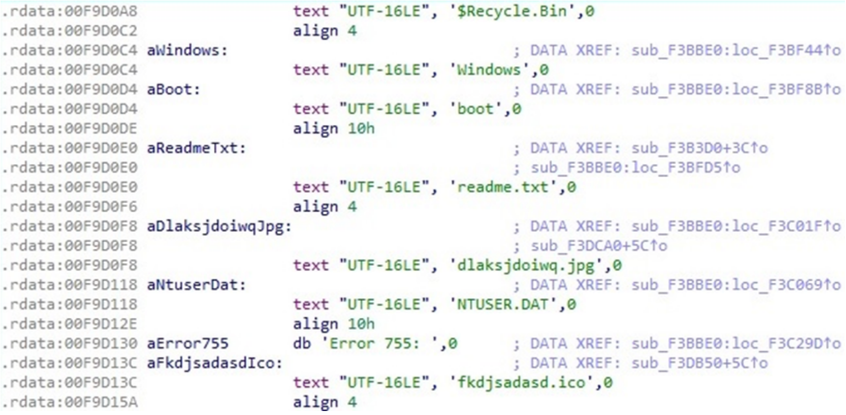


```
00F37A51   56                 push esi
00F37A52   0F 43 45 D4        cmovae eax,dword ptr ss:[ebp-2C]
00F37A56   50                 push eax
00F37A57   FF 15 84 20 F9 00  call dword ptr ds:[<&FindFirstFileW>]

dword ptr [00F92084 <malware.&FindFirstFileW>]=<kernel32.FindFirstFileW>

.text:00F37A57 malware.exe:$7A57 #6E57

x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)  5  Unlock
1: [esp] 00EFF910 L"C:\\*"
2: [esp+4] 03056898
3: [esp+8] 21ECE3E3
4: [esp+C] 00EFFA40 &"C:\\"

00EFF8A4 00EFF910  L"C:\\*"
00EFF8A8 03056898
```

Figure 34

As shown in figure 35, the following files/directories will be skipped:

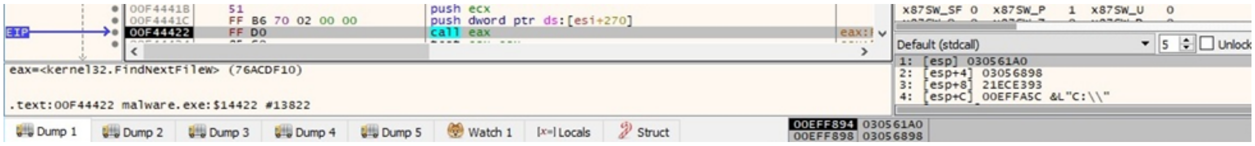
- $Recycle.Bin
- Windows
- boot
- readme.txt
- dlaksjdoiwq.jpg
- NTUSER.DAT
- fkdjsadasd.ico



```
.rdata:00F9D0A8                 text "UTF-16LE", '$Recycle.Bin',0
.rdata:00F9D0C2                 align 4
.rdata:00F9D0C4 aWindows:                               ; DATA XREF: sub_F3BBE0:loc_F3BF44↑o
.rdata:00F9D0C4                 text "UTF-16LE", 'Windows',0
.rdata:00F9D0D4 aBoot:                                  ; DATA XREF: sub_F3BBE0:loc_F3BF8B↑o
.rdata:00F9D0D4                 text "UTF-16LE", 'boot',0
.rdata:00F9D0DE                 align 10h
.rdata:00F9D0E0 aReadmeTxt:                             ; DATA XREF: sub_F3B3D0+3C↑o
.rdata:00F9D0E0                                         ; sub_F3BBE0:loc_F3BFD5↑o
.rdata:00F9D0E0                 text "UTF-16LE", 'readme.txt',0
.rdata:00F9D0F6                 align 4
.rdata:00F9D0F8 aDlaksjdoiwqJpg:                        ; DATA XREF: sub_F3BBE0:loc_F3C01F↑o
.rdata:00F9D0F8                                         ; sub_F3DCA0+5C↑o
.rdata:00F9D0F8                 text "UTF-16LE", 'dlaksjdoiwq.jpg',0
.rdata:00F9D118 aNtuserDat:                             ; DATA XREF: sub_F3BBE0:loc_F3C069↑o
.rdata:00F9D118                 text "UTF-16LE", 'NTUSER.DAT',0
.rdata:00F9D12E                 align 10h
.rdata:00F9D130 aError755       db 'Error 755: ',0      ; DATA XREF: sub_F3BBE0:loc_F3C29D↑o
.rdata:00F9D13C aFkdjsadasdIco:                         ; DATA XREF: sub_F3DB50+5C↑o
.rdata:00F9D13C                 text "UTF-16LE", 'fkdjsadasd.ico',0
.rdata:00F9D15A                 align 4
```

Figure 35

The FindNextFileW routine is utilized to continue the files enumeration:

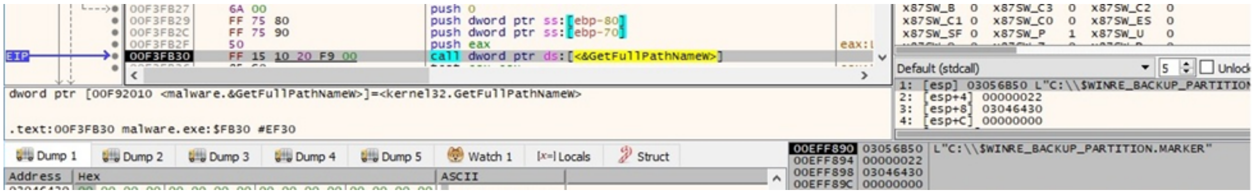


```
00F4441B   51                  push ecx
00F4441C   FF B6 70 02 00 00   push dword ptr ds:[esi+270]
00F44422   FF D0               call eax

eax=<kernel32.FindNextFileW> (76ACDF10)

.text:00F44422 malware.exe:$14422 #13822
```

```
x87SW_SF 0  x87SW_P   1  x87SW_U  0
Default (stdcall)  5  Unlock
1: [esp] 030561A0
2: [esp+4] 03056898
3: [esp+8] 21ECE393
4: [esp+C] 00EFFA5C &L"C:\\"

00EFF894 030561A0
00EFF898 03056898
```

Figure 36

Black Basta ransomware calls the GetFullPathNameW API with a targeted file as a parameter:

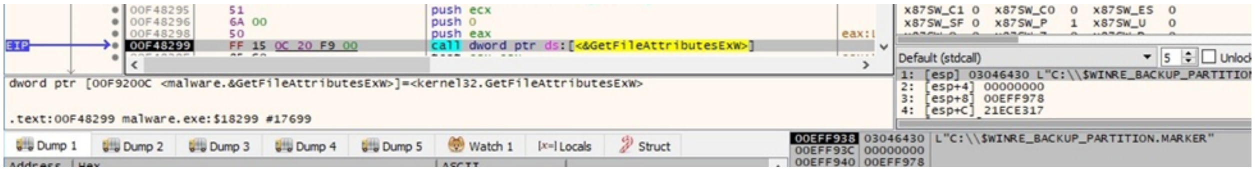


```
00F3FB27   6A 00               push 0
00F3FB29   FF 75 80            push dword ptr ss:[ebp-80]
00F3FB2C   FF 75 90            push dword ptr ss:[ebp-70]
00F3FB2F   50                  push eax
00F3FB30   FF 15 10 20 F9 00   call dword ptr ds:[<&GetFullPathNameW>]

dword ptr [00F92010 <malware.&GetFullPathNameW>]=<kernel32.GetFullPathNameW>

.text:00F3FB30 malware.exe:$FB30 #EF30
```

```
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0
Default (stdcall)  5  Unlock
1: [esp] 03056B50 L"C:\\$WINRE_BACKUP_PARTITION
2: [esp+4] 00000022
3: [esp+8] 03046430
4: [esp+C] 00000000

00EFF890 03056B50  L"C:\\$WINRE_BACKUP_PARTITION.MARKER"
00EFF894 00000022
00EFF898 03046430
00EFF89C 00000000
```

Figure 37

The process obtains a standard set of attribute information for the file via a call to GetFileAttributesExW:



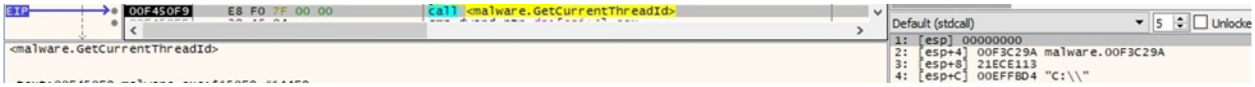
```
00F48295   51            push ecx
00F48296   6A 00         push 0
00F48298   50            push eax                                       eax:l
00F48299   FF 15 0C 20 F9 00   call dword ptr ds:[<&GetFileAttributesExW>]

dword ptr [00F9200C <malware.&GetFileAttributesExW>]=<kernel32.GetFileAttributesExW>

.text:00F48299 malware.exe:$18299 #17699
```

```
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)  5  Unlock
1: [esp] 03046430 L"C:\\$WINRE_BACKUP_PARTITIO
2: [esp+4] 00000000
3: [esp+8] 00EFF978
4: [esp+C] 21ECE317

00EFF938 03046430 L"C:\\$WINRE_BACKUP_PARTITION.MARKER"
00EFF93C 00000000
00EFF940 00EFF978
```

Figure 38

The ransomware has embedded a list of extensions (.exe, .cmd, .bat, and .com) in a section; however, it still encrypts these file extensions.

The executable retrieves the thread identifier of the calling thread using GetCurrentThreadId:

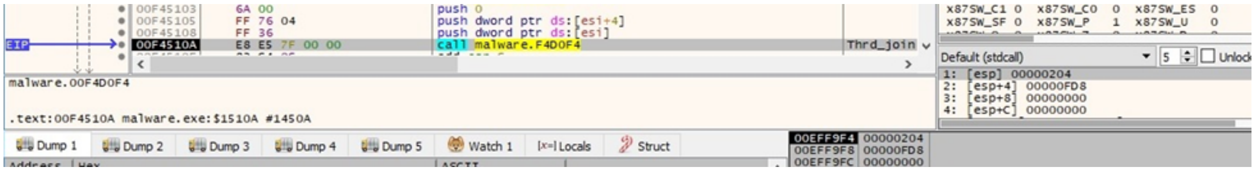


```
00F450F9   E8 F0 7F 00 00   call <malware.GetCurrentThreadId>

<malware.GetCurrentThreadId>
```

```
Default (stdcall)  5  Unlocke
1: [esp] 00000000
2: [esp+4] 00F3C29A malware.00F3C29A
3: [esp+8] 21ECE113
4: [esp+C] 00EFFBD4 "C:\\"
```

Figure 39

The malicious process blocks the main thread until all encryption threads finish execution (see figure 40).



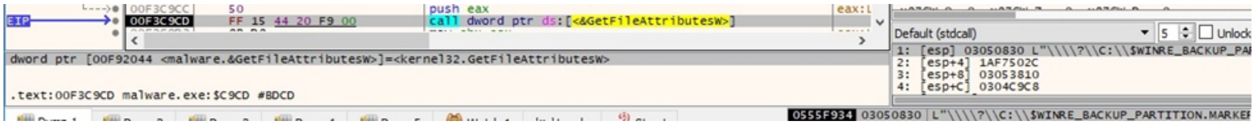
```
00F45103   6A 00          push 0
00F45105   FF 76 04       push dword ptr ds:[esi+4]
00F45108   FF 36          push dword ptr ds:[esi]
00F4510A   E8 E5 7F 00 00 call malware.F4D0F4                  Thrd_join

malware.00F4D0F4

.text:00F4510A malware.exe:$1510A #1450A
```

```
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0

Default (stdcall)  5  Unlock
1: [esp] 00000204
2: [esp+4] 00000FD8
3: [esp+8] 00000000
4: [esp+C] 00000000

00EFF9F4 00000204
00EFF9F8 00000FD8
00EFF9FC 00000000
```

Figure 40

Thread activity – sub_F33DA0 function

The GetFileAttributesW API is utilized to retrieve file system attributes for a targeted file:

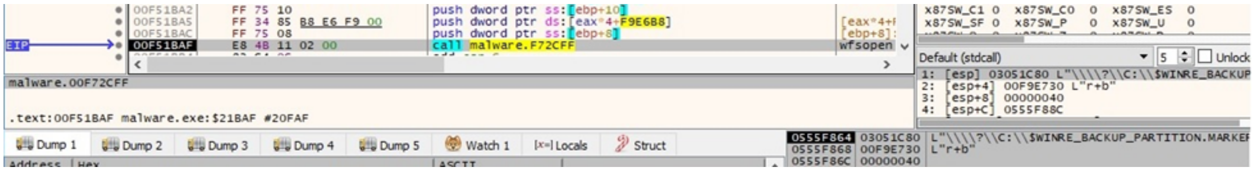


```
00F3C9CC   50                  push eax                                 eax:L
00F3C9CD   FF 15 44 20 F9 00   call dword ptr ds:[<&GetFileAttributesW>]

dword ptr [00F92044 <malware.&GetFileAttributesW>]=<kernel32.GetFileAttributesW>

.text:00F3C9CD malware.exe:$C9CD #BDCD
```

```
Default (stdcall)  5  Unlock
1: [esp] 03050830 L"\\\\?\\C:\\$WINRE_BACKUP_PA
2: [esp+4] 1AF7502C
3: [esp+8] 03053810
4: [esp+C] 0304C9C8

0555F934 03050830 L"\\\\?\\C:\\$WINRE_BACKUP_PARTITION.MARKEF
```

Figure 41

The malicious process opens a file for reading using wfsopen:

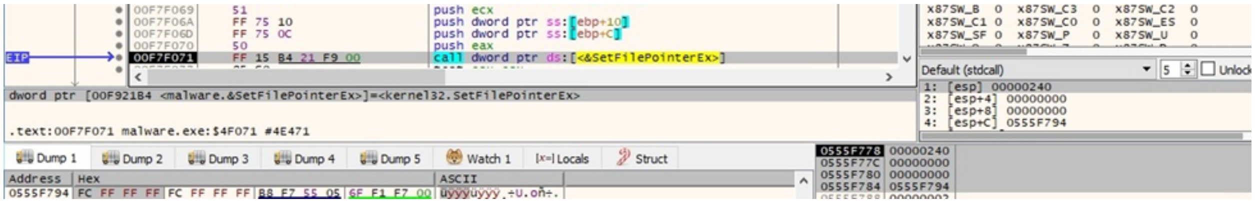


```
00F51BA2   FF 75 10            push dword ptr ss:[ebp+10]
00F51BA5   FF 34 85 B8 E6 F9 00   push dword ptr ds:[eax*4+F9E6B8]     [eax*4+F
00F51BAC   FF 75 08            push dword ptr ss:[ebp+8]               [ebp+8]:
00F51BAF   E8 4B 11 02 00      call malware.F72CFF                     wfsopen

malware.00F72CFF

.text:00F51BAF malware.exe:$21BAF #20FAF
```

```
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   0  x87SW_U   0

Default (stdcall)  5  Unlock
1: [esp] 03051C80 L"\\\\?\\C:\\$WINRE_BACKUP
2: [esp+4] 00F9E730 L"r+b"
3: [esp+8] 00000040
4: [esp+C] 0555F88C

0555F864 03051C80 L"\\\\?\\C:\\$WINRE_BACKUP_PARTITION.MARKEF
0555F868 00F9E730 L"r+b"
0555F86C 00000040
```

Figure 42

The ransomware moves the file pointer to the position of the last 4 bytes. Whether the file would be encrypted, these would represent the length of the encrypted ChaCha20 key and nonce, as we'll see later on:

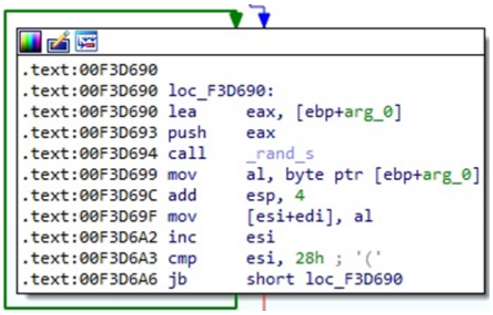


```
00F7F069   51                  push ecx
00F7F06A   FF 75 10            push dword ptr ss:[ebp+10]
00F7F06D   FF 75 0C            push dword ptr ss:[ebp+C]
00F7F070   50                  push eax
00F7F071   FF 15 B4 21 F9 00   call dword ptr ds:[<&SetFilePointerEx>]

dword ptr [00F921B4 <malware.&SetFilePointerEx>]=<kernel32.SetFilePointerEx>

.text:00F7F071 malware.exe:$4F071 #4E471

Address   Hex                                              ASCII
0555F794  FC FF FF FF FC FF FF FF B8 F7 55 05 6F F1 F7 00  üÿÿÿüÿÿÿ¸÷U.oñ÷.
```

```
x87SW_B  0  x87SW_C3 0  x87SW_C2 0
x87SW_C1 0  x87SW_C0 0  x87SW_ES 0
x87SW_SF 0  x87SW_P  0  x87SW_U  0

Default (stdcall)  5  Unlock
1: [esp] 00000240
2: [esp+4] 00000000
3: [esp+8] 00000000
4: [esp+C] 0555F794

0555F778 00000240
0555F77C 00000000
0555F780 00000000
0555F784 0555F794
```

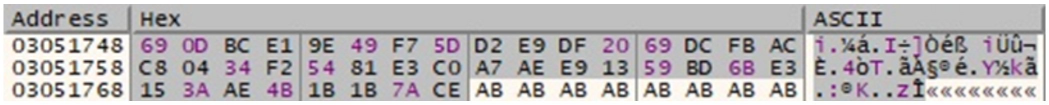
Figure 43

Black Basta ransomware generates 32 random bytes representing the ChaCha20 key and then 8 bytes representing the nonce using rand_s:



```
.text:00F3D690
.text:00F3D690 loc_F3D690:
.text:00F3D690 lea     eax, [ebp+arg_0]
.text:00F3D693 push    eax
.text:00F3D694 call    _rand_s
.text:00F3D699 mov     al, byte ptr [ebp+arg_0]
.text:00F3D69C add     esp, 4
.text:00F3D69F mov     [esi+edi], al
.text:00F3D6A2 inc     esi
.text:00F3D6A3 cmp     esi, 28h ; '('
.text:00F3D6A6 jb      short loc_F3D690
```

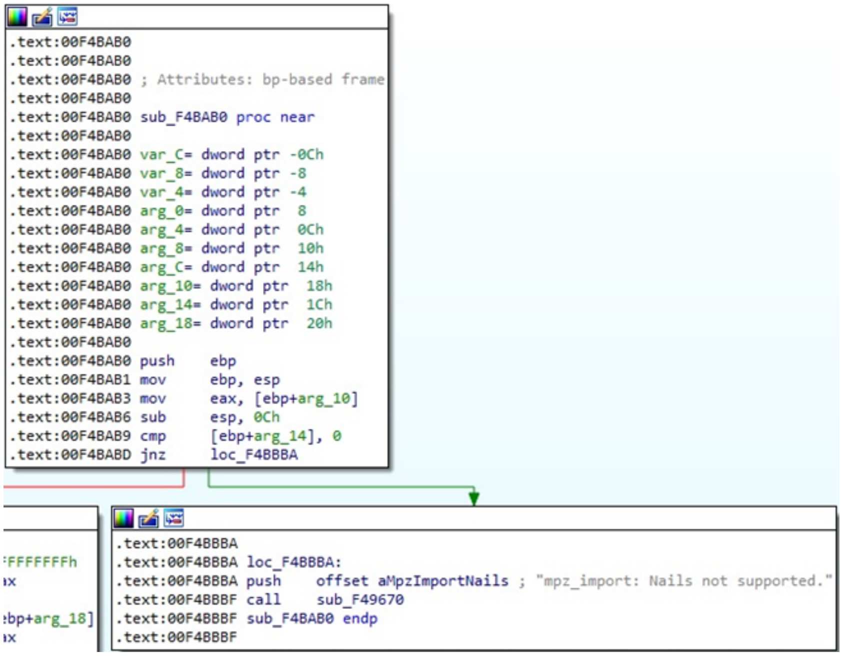
Figure 44



```
Address  | Hex                                             | ASCII
03051748 | 69 0D BC E1 9E 49 F7 5D D2 E9 DF 20 69 DC FB AC | i.¼á.I÷]Òéß iÜû¬
03051758 | C8 04 34 F2 54 81 E3 C0 A7 AE E9 13 59 BD 6B E3 | È.4òT.ãÀ§®é.Y½kã
03051768 | 15 3A AE 4B 1B 1B 7A CE AB AB AB AB AB AB AB AB | .:®K..zÎ««««««««
```

Figure 45

The binary implements the RSA algorithm using the Mini-GMP library, which is fully available on Github:

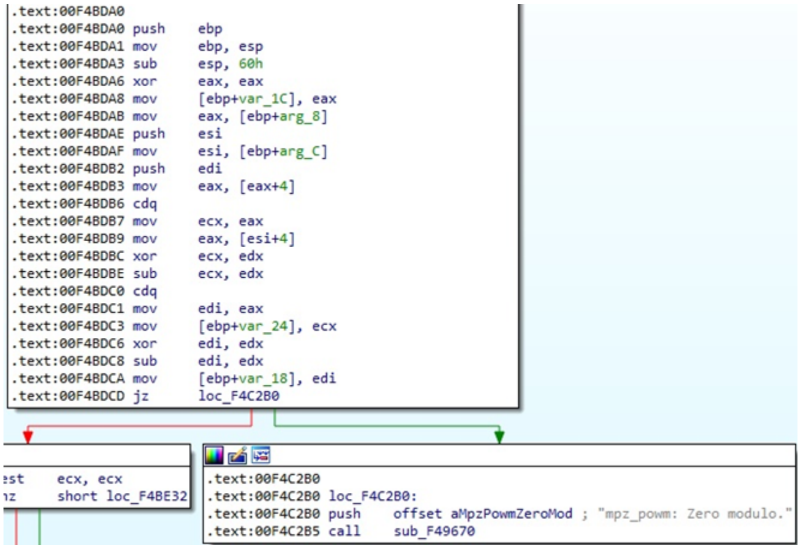


```
.text:00F4BAB0
.text:00F4BAB0
.text:00F4BAB0 ; Attributes: bp-based frame
.text:00F4BAB0
.text:00F4BAB0 sub_F4BAB0 proc near
.text:00F4BAB0
.text:00F4BAB0 var_C= dword ptr -0Ch
.text:00F4BAB0 var_8= dword ptr -8
.text:00F4BAB0 var_4= dword ptr -4
.text:00F4BAB0 arg_0= dword ptr  8
.text:00F4BAB0 arg_4= dword ptr  0Ch
.text:00F4BAB0 arg_8= dword ptr  10h
.text:00F4BAB0 arg_C= dword ptr  14h
.text:00F4BAB0 arg_10= dword ptr  18h
.text:00F4BAB0 arg_14= dword ptr  1Ch
.text:00F4BAB0 arg_18= dword ptr  20h
.text:00F4BAB0
.text:00F4BAB0 push    ebp
.text:00F4BAB1 mov     ebp, esp
.text:00F4BAB3 mov     eax, [ebp+arg_10]
.text:00F4BAB6 sub     esp, 0Ch
.text:00F4BAB9 cmp     [ebp+arg_14], 0
.text:00F4BABD jnz     loc_F4BBBA
```

```
FFFFFFFh
ax

ebp+arg_18]
ax
```

```
.text:00F4BBBA
.text:00F4BBBA loc_F4BBBA:
.text:00F4BBBA push    offset aMpzImportNails ; "mpz_import: Nails not supported."
.text:00F4BBBF call    sub_F49670
.text:00F4BBBF sub_F4BAB0 endp
.text:00F4BBBF
```

Figure 46



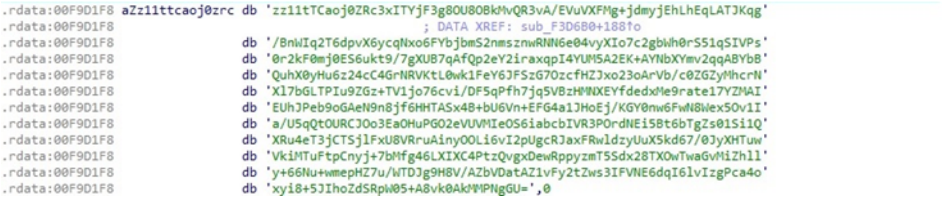
```
.text:00F4BDA0
.text:00F4BDA0 push    ebp
.text:00F4BDA1 mov     ebp, esp
.text:00F4BDA3 sub     esp, 60h
.text:00F4BDA6 xor     eax, eax
.text:00F4BDA8 mov     [ebp+var_1C], eax
.text:00F4BDAB mov     eax, [ebp+arg_8]
.text:00F4BDAE push    esi
.text:00F4BDAF mov     esi, [ebp+arg_C]
.text:00F4BDB2 push    edi
.text:00F4BDB3 mov     eax, [eax+4]
.text:00F4BDB6 cdq
.text:00F4BDB7 mov     ecx, eax
.text:00F4BDB9 mov     eax, [esi+4]
.text:00F4BDBC xor     ecx, edx
.text:00F4BDBE sub     ecx, edx
.text:00F4BDC0 cdq
.text:00F4BDC1 mov     edi, eax
.text:00F4BDC3 mov     [ebp+var_24], ecx
.text:00F4BDC6 xor     edi, edx
.text:00F4BDC8 sub     edi, edx
.text:00F4BDCA mov     [ebp+var_18], edi
.text:00F4BDCD jz      loc_F4C2B0
```

```
est    ecx, ecx
nz     short loc_F4BE32
```

```
.text:00F4C2B0
.text:00F4C2B0 loc_F4C2B0:
.text:00F4C2B0 push    offset aMpzPowmZeroMod ; "mpz_powm: Zero modulo."
.text:00F4C2B5 call    sub_F49670
```

Figure 47

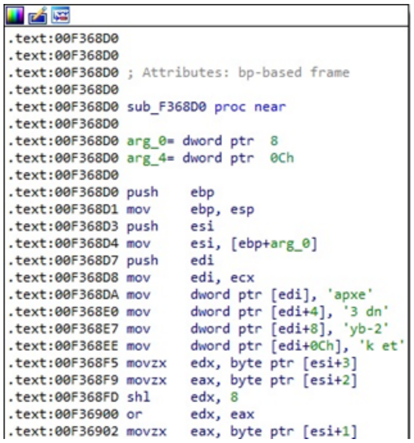
The RSA public key used to encrypt the randomly generated ChaCha20 key and the nonce is presented in the figure below:



```
.rdata:00F9D1F8 aZz11ttcaoj0zrc db 'zz11tTCaoj0ZRc3xITYjF3g8OU8OBkMvQR3vA/EVuVXFMg+jdmyjEhLhEqLATJKqg'
.rdata:00F9D1F8                                         ; DATA XREF: sub_F3D6B0+188↑o
.rdata:00F9D1F8                 db '/BnWIq2T6dpvX6ycqNxo6FYbjbmS2nmsznwRNN6e04vyXIo7c2gbWh0rS51qSIVPs'
.rdata:00F9D1F8                 db '0r2kF0mj0ES6ukt9/7gXUB7qAfQp2eY2iraxqpI4YUM5A2EK+AYNbXYmv2qqABYbB'
.rdata:00F9D1F8                 db 'QuhX0yHu6z24cC4GrNRVKtL0wk1FeY6JFSzG7OzcfHZJxo23oArVb/c0ZGZyMhcrN'
.rdata:00F9D1F8                 db 'Xl7bGLTPIu9ZGz+TV1jo76cvi/DF5qPfh7jq5VBzHMNXEYfdedxMe9rate17YZMAI'
.rdata:00F9D1F8                 db 'EUhJPeb9oGAeN9n8jf6HHTASx4B+bU6Vn+EFG4a1JHoEj/KGY0nw6FwN8Wex5Ov1I'
.rdata:00F9D1F8                 db 'a/U5qQtOURCJOo3EaOHuPGO2eVUVMIeOS6iabcbIVR3POrdNEi5Bt6bTgZs01Si1Q'
.rdata:00F9D1F8                 db 'XRu4eT3jCTSjlFxU8VRruAinyOOLi6vI2pUgcRJaxFRwldzyUuX5kd67/0JyXHTuw'
.rdata:00F9D1F8                 db 'VkiMTuFtpCnyj+7bMfg46LXIXC4PtzQvgxDewRppyzmT5Sdx28TXOwTwaGvMiZhll'
.rdata:00F9D1F8                 db 'y+66Nu+wmepHZ7u/WTDJg9H8V/AZbVDatAZ1vFy2tZws3IFVNE6dqI6lvIzgPca4o'
.rdata:00F9D1F8                 db 'xyi8+5JIhoZdSRpW05+A8vk0AkMMPNgGU=',0
```

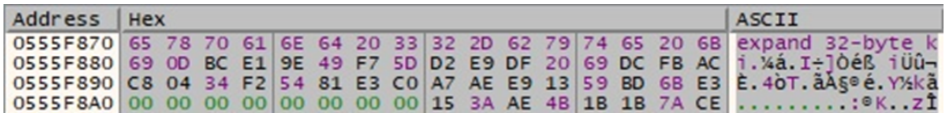
Figure 48

The process constructs the initial state of ChaCha20 using the key, the nonce, and some constant values:



```
.text:00F368D0
.text:00F368D0
.text:00F368D0 ; Attributes: bp-based frame
.text:00F368D0
.text:00F368D0 sub_F368D0 proc near
.text:00F368D0
.text:00F368D0 arg_0= dword ptr  8
.text:00F368D0 arg_4= dword ptr  0Ch
.text:00F368D0
.text:00F368D0 push    ebp
.text:00F368D1 mov     ebp, esp
.text:00F368D3 push    esi
.text:00F368D4 mov     esi, [ebp+arg_0]
.text:00F368D7 push    edi
.text:00F368D8 mov     edi, ecx
.text:00F368DA mov     dword ptr [edi], 'apxe'
.text:00F368E0 mov     dword ptr [edi+4], '3 dn'
.text:00F368E7 mov     dword ptr [edi+8], 'yb-2'
.text:00F368EE mov     dword ptr [edi+0Ch], 'k et'
.text:00F368F5 movzx   edx, byte ptr [esi+3]
.text:00F368F9 movzx   eax, byte ptr [esi+2]
.text:00F368FD shl     edx, 8
.text:00F36900 or      edx, eax
.text:00F36902 movzx   eax, byte ptr [esi+1]
```

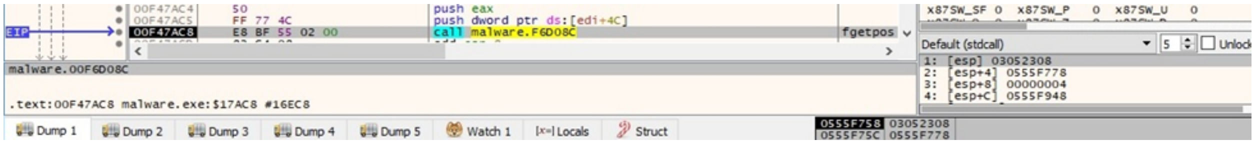
Figure 49



```
Address  | Hex                                             | ASCII
0555F870 | 65 78 70 61 6E 64 20 33 32 2D 62 79 74 65 20 6B | expand 32-byte k
0555F880 | 69 0D BC E1 9E 49 F7 5D D2 E9 DF 20 69 DC FB AC | i.¼á.I÷]Òéß iÜû¬
0555F890 | C8 04 34 F2 54 81 E3 C0 A7 AE E9 13 59 BD 6B E3 | È.4òT.ãÀ§®é.Y½kã
0555F8A0 | 00 00 00 00 00 00 00 00 15 3A AE 4B 1B 1B 7A CE | .........:®K..zÎ
```

Figure 50

The sample obtains the current position in the targeted file by calling the fgetpos function:

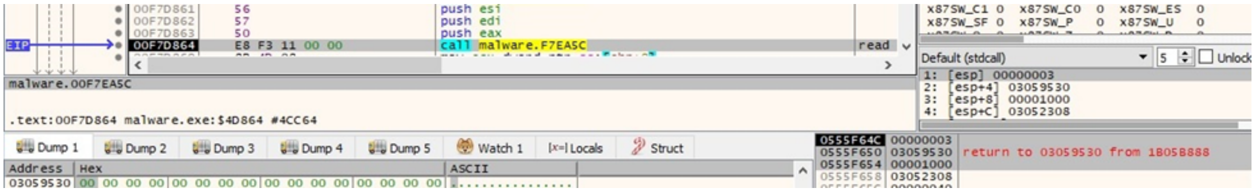


```
00F47AC4   50              push eax
00F47AC5   FF 77 4C        push dword ptr ds:[edi+4C]
00F47AC8   E8 BF 55 02 00  call malware.F6D08C              fgetpos
malware.00F6D08C

.text:00F47AC8 malware.exe:$17AC8 #16EC8
```

```
x87SW_SF 0  x87SW_P  0  x87SW_U  0
Default (stdcall)   5   Unlock
1: [esp] 03052308
2: [esp+4] 0555F778
3: [esp+8] 00000004
4: [esp+C] 0555F948
0555F758 03052308
0555F75C 0555F778
```

Figure 51

The file content is read by the process via a call to the _read function:



```
00F7D861   56              push esi
00F7D862   57              push edi
00F7D863   50              push eax
00F7D864   E8 F3 11 00 00  call malware.F7EA5C              read
malware.00F7EA5C

.text:00F7D864 malware.exe:$4D864 #4CC64
```

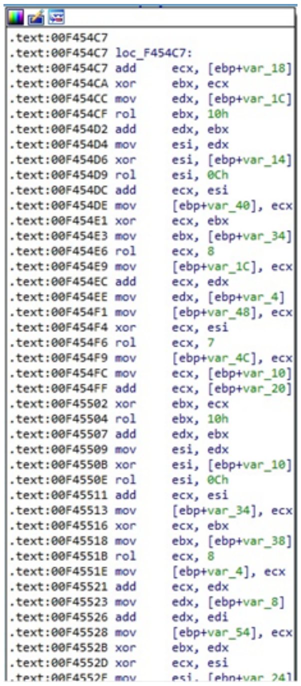
```
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   0  x87SW_U   0
Default (stdcall)   5   Unlock
1: [esp] 00000003
2: [esp+4] 03059530
3: [esp+8] 00001000
4: [esp+C] 03052308
```

```
Address  | Hex                                             | ASCII
03059530 | 00 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00 | ................
```

```
0555F64C 00000003
0555F650 03059530 return to 03059530 from 1805B888
0555F654 00001000
0555F658 03052308
```

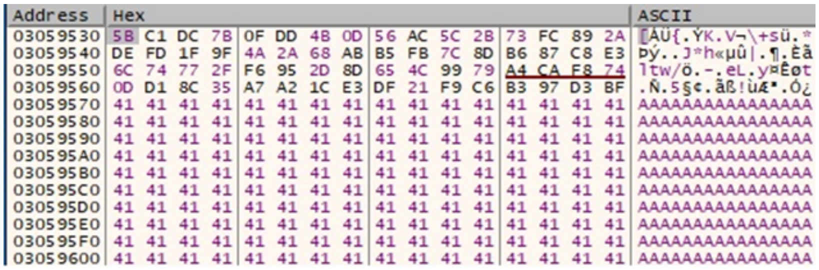
Figure 52

The content is encrypted by the ChaCha20 algorithm 64 bytes at a time:



```
.text:00F454C7
.text:00F454C7 loc_F454C7:
.text:00F454C7 add     ecx, [ebp+var_18]
.text:00F454CA xor     ebx, ecx
.text:00F454CC mov     edx, [ebp+var_1C]
.text:00F454CF rol     ebx, 10h
.text:00F454D2 add     edx, ebx
.text:00F454D4 mov     esi, edx
.text:00F454D6 xor     esi, [ebp+var_14]
.text:00F454D9 rol     esi, 0Ch
.text:00F454DC add     ecx, esi
.text:00F454DE mov     [ebp+var_40], ecx
.text:00F454E1 xor     ecx, ebx
.text:00F454E3 mov     ebx, [ebp+var_34]
.text:00F454E6 rol     ecx, 8
.text:00F454E9 mov     [ebp+var_1C], ecx
.text:00F454EC add     ecx, edx
.text:00F454EE mov     edx, [ebp+var_4]
.text:00F454F1 mov     [ebp+var_48], ecx
.text:00F454F4 xor     ecx, esi
.text:00F454F6 rol     ecx, 7
.text:00F454F9 mov     [ebp+var_4C], ecx
.text:00F454FC mov     ecx, [ebp+var_10]
.text:00F454FF add     ecx, [ebp+var_20]
.text:00F45502 xor     ebx, ecx
.text:00F45504 rol     ebx, 10h
.text:00F45507 add     edx, ebx
.text:00F45509 mov     esi, edx
.text:00F4550B xor     esi, [ebp+var_10]
.text:00F4550E rol     esi, 0Ch
.text:00F45511 add     ecx, esi
.text:00F45513 mov     [ebp+var_34], ecx
.text:00F45516 xor     ecx, ebx
.text:00F45518 mov     ebx, [ebp+var_38]
.text:00F4551B rol     ecx, 8
.text:00F4551E mov     [ebp+var_4], ecx
.text:00F45521 add     ecx, edx
.text:00F45523 mov     edx, [ebp+var_8]
.text:00F45526 add     edx, edi
.text:00F45528 mov     [ebp+var_54], ecx
.text:00F4552B xor     ebx, edx
.text:00F4552D xor     ecx, esi
.text:00F4552F mov     esi, [ebp+var_24]
```

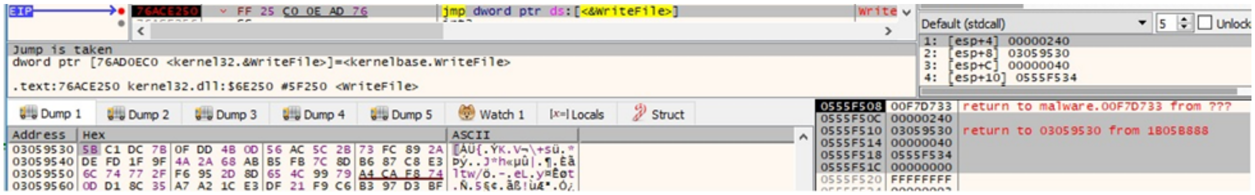
Figure 53



```
Address  | Hex                                             | ASCII
03059530 | 5B C1 DC 7B|0F DD 4B 0D|56 AC 5C 2B|73 FC 89 2A | [ÁÜ{.ÝK.V¬\+sü.*
03059540 | DE FD 1F 9F|4A 2A 68 AB|B5 FB 7C 8D|B6 87 C8 E3 | Þý..J*h«µû|.¶.Èã
03059550 | 6C 74 77 2F|F6 95 2D 8D|65 4C 99 79|A4 CA F8 74 | ltw/ö.-.eL.y¤Êøt
03059560 | 0D D1 8C 35|A7 A2 1C E3|DF 21 F9 C6|B3 97 D3 BF | .Ñ.5§¢.ãß!ùÆ³.Ó¿
03059570 | 41 41 41 41|41 41 41 41|41 41 41 41|41 41 41 41 | AAAAAAAAAAAAAAAA
03059580 | 41 41 41 41|41 41 41 41|41 41 41 41|41 41 41 41 | AAAAAAAAAAAAAAAA
03059590 | 41 41 41 41|41 41 41 41|41 41 41 41|41 41 41 41 | AAAAAAAAAAAAAAAA
030595A0 | 41 41 41 41|41 41 41 41|41 41 41 41|41 41 41 41 | AAAAAAAAAAAAAAAA
030595B0 | 41 41 41 41|41 41 41 41|41 41 41 41|41 41 41 41 | AAAAAAAAAAAAAAAA
030595C0 | 41 41 41 41|41 41 41 41|41 41 41 41|41 41 41 41 | AAAAAAAAAAAAAAAA
030595D0 | 41 41 41 41|41 41 41 41|41 41 41 41|41 41 41 41 | AAAAAAAAAAAAAAAA
030595E0 | 41 41 41 41|41 41 41 41|41 41 41 41|41 41 41 41 | AAAAAAAAAAAAAAAA
030595F0 | 41 41 41 41|41 41 41 41|41 41 41 41|41 41 41 41 | AAAAAAAAAAAAAAAA
03059600 | 41 41 41 41|41 41 41 41|41 41 41 41|41 41 41 41 | AAAAAAAAAAAAAAAA
```

Figure 54

The encrypted data is written back to the file using the WriteFile API:



```
76ACE250   FF 25 C0 0E AD 76   jmp dword ptr ds:[<&WriteFile>]      Write
Jump is taken
dword ptr [76AD0EC0 <kernel32.&WriteFile>]=<kernelbase.WriteFile>

.text:76ACE250 kernel32.dll:$6E250 #5F250 <WriteFile>
```

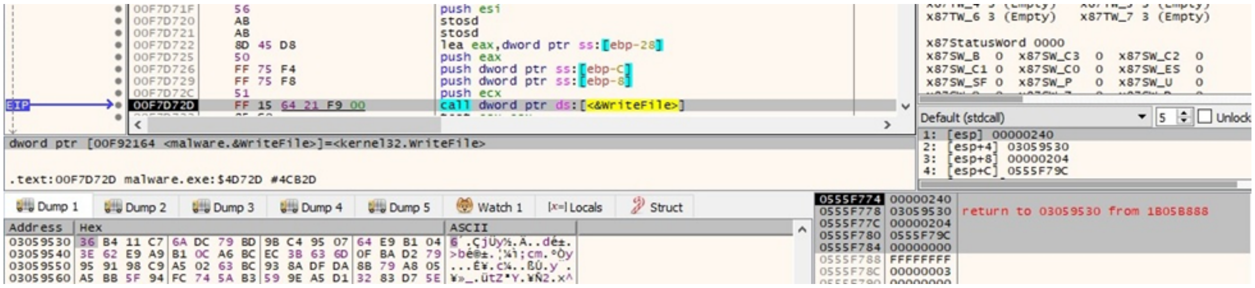
```
Default (stdcall)   5   Unlock
1: [esp+4] 00000240
2: [esp+8] 03059530
3: [esp+C] 00000040
4: [esp+10] 0555F534
```

```
Address  | Hex                                             | ASCII
03059530 | 5B C1 DC 7B|0F DD 4B 0D|56 AC 5C 2B|73 FC 89 2A | [ÁÜ{.ÝK.V¬\+sü.*
03059540 | DE FD 1F 9F|4A 2A 68 AB|B5 FB 7C 8D|B6 87 C8 E3 | Þý..J*h«µû|.¶.Èã
03059550 | 6C 74 77 2F|F6 95 2D 8D|65 4C 99 79|A4 CA F8 74 | ltw/ö.-.eL.y¤Êøt
03059560 | 0D D1 8C 35|A7 A2 1C E3|DF 21 F9 C6|B3 97 D3 BF | .Ñ.5§¢.ãß!ùÆ³.Ó¿
```

```
0555F508 00F7D733 return to malware.00F7D733 from ???
0555F50C 00000240
0555F510 03059530 return to 03059530 from 1805B888
0555F514 00000040
0555F518 0555F534
0555F51C 00000000
0555F520 FFFFFFFF
```

Figure 55

The buffer containing the RSA encrypted ChaCha20 key and nonce is appended to the encrypted file. The length of the encrypted information (0x200 = 512) is added as well:



```
00F7D71F   56            push esi
00F7D720   AB            stosd
00F7D721   AB            stosd
00F7D722   8D 45 D8      lea eax,dword ptr ss:[ebp-28]
00F7D725   50            push eax
00F7D726   FF 75 F4      push dword ptr ss:[ebp-C]
00F7D729   FF 75 F8      push dword ptr ss:[ebp-8]
00F7D72C   51            push ecx
00F7D72D   FF 15 64 21 F9 00   call dword ptr ds:[<&WriteFile>]

dword ptr [00F92164 <malware.&WriteFile>]=<kernel32.WriteFile>

.text:00F7D72D malware.exe:$4D72D #4CB2D
```

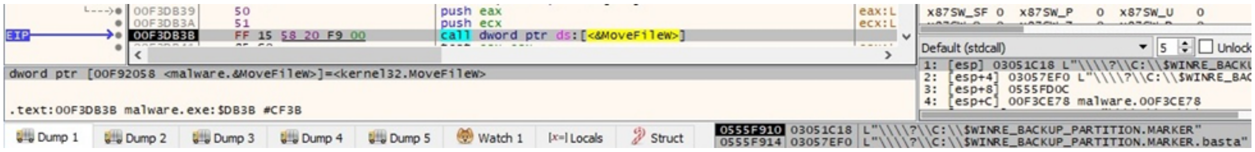
```
x87TW_6 3 (Empty)   x87TW_7 3 (Empty)

x87StatusWord 0000
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   0  x87SW_U   0

Default (stdcall)   5   Unlock
1: [esp] 00000240
2: [esp+4] 03059530
3: [esp+8] 00000204
4: [esp+C] 0555F79C
```

```
Address  Hex                                              ASCII
03059530 36 B4 11 C7 6A DC 79 BD 9B C4 95 07 64 E9 B1 04  6´.ÇjÜy½.Ä..dé±.
03059540 3E 62 E9 A9 B1 0C A6 BC EC 3B 63 6D 0F BA D2 79  >bé©±.¦¼ì;cm.ºÒy
03059550 95 91 98 C9 A5 02 63 BC 93 8A DF DA 8B 79 A8 05  ...É¥.c¼..ßÚ.y¨.
03059560 A5 BB 5F 94 FC 74 5A B3 59 9E A5 D1 32 83 D7 5E  ¥»_.ütZ³Y.¥Ñ2.×^
```

```
0555F774 00000240
0555F778 03059530  return to 03059530 from 1B05B888
0555F77C 00000204
0555F780 0555F79C
0555F784 00000000
0555F788 FFFFFFFF
0555F78C 00000003
```

Figure 56

The encrypted file extension is changed to “.basta” using MoveFileW:



```
00F3DB39   50            push eax
00F3DB3A   51            push ecx
00F3DB3B   FF 15 58 20 F9 00   call dword ptr ds:[<&MoveFileW>]

dword ptr [00F92058 <malware.&MoveFileW>]=<kernel32.MoveFileW>

.text:00F3DB3B malware.exe:$DB3B #CF3B
```

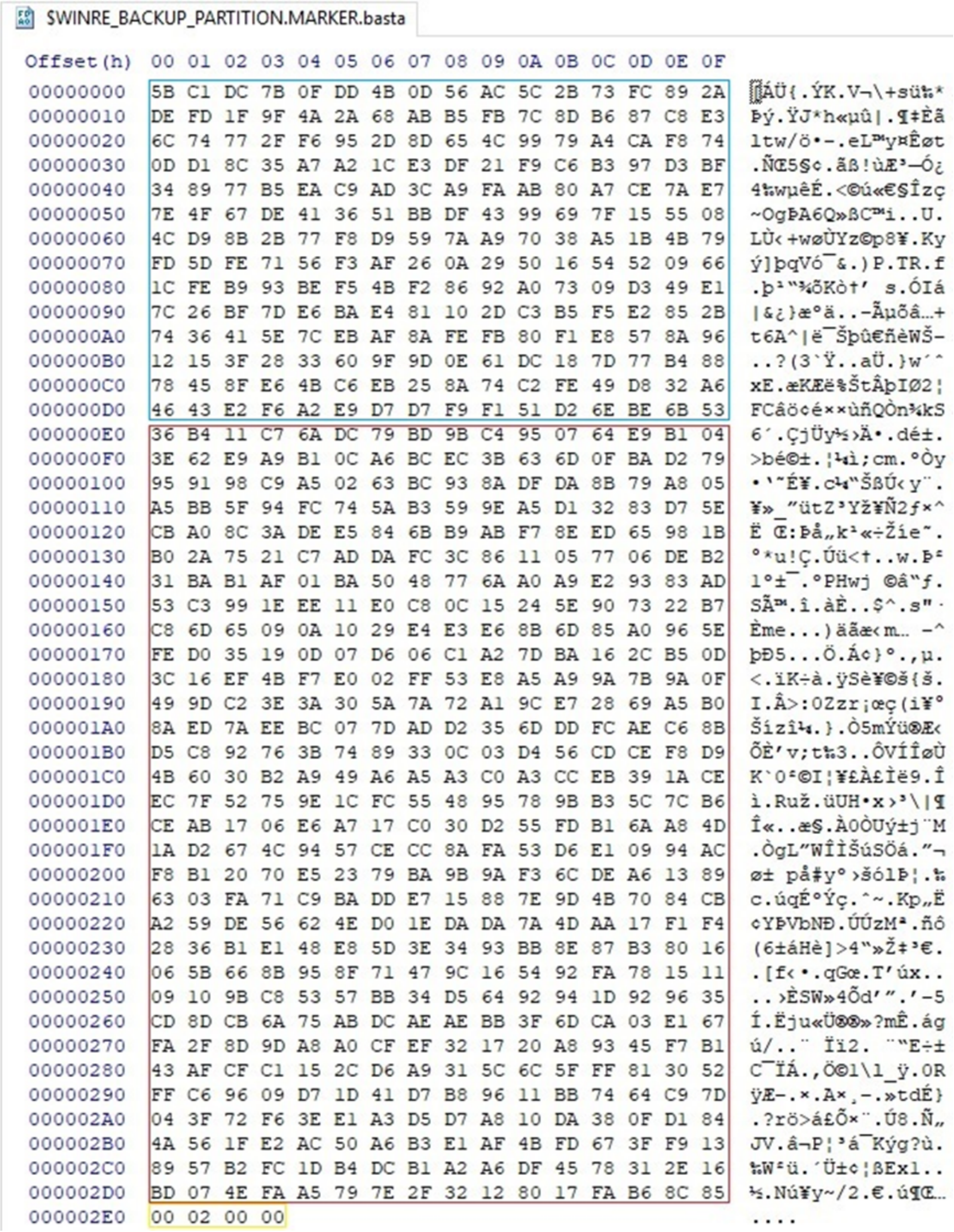
```
x87SW_SF 0  x87SW_P   0  x87SW_U   0

Default (stdcall)   5   Unlock
1: [esp] 03051C18 L"\\\\?\\C:\\$WINRE_BACKU
2: [esp+4] 03057EF0 L"\\\\?\\C:\\$WINRE_BAC
3: [esp+8] 0555FD0C
4: [esp+C] 00F3CE78 malware.00F3CE78
```

```
0555F910 03051C18 L"\\\\?\\C:\\$WINRE_BACKUP_PARTITION.MARKER"
0555F914 03057EF0 L"\\\\?\\C:\\$WINRE_BACKUP_PARTITION.MARKER.basta"
```

Figure 57

Case 1 – File size < 704 bytes

In this case, the entire file content is encrypted by the ransomware:

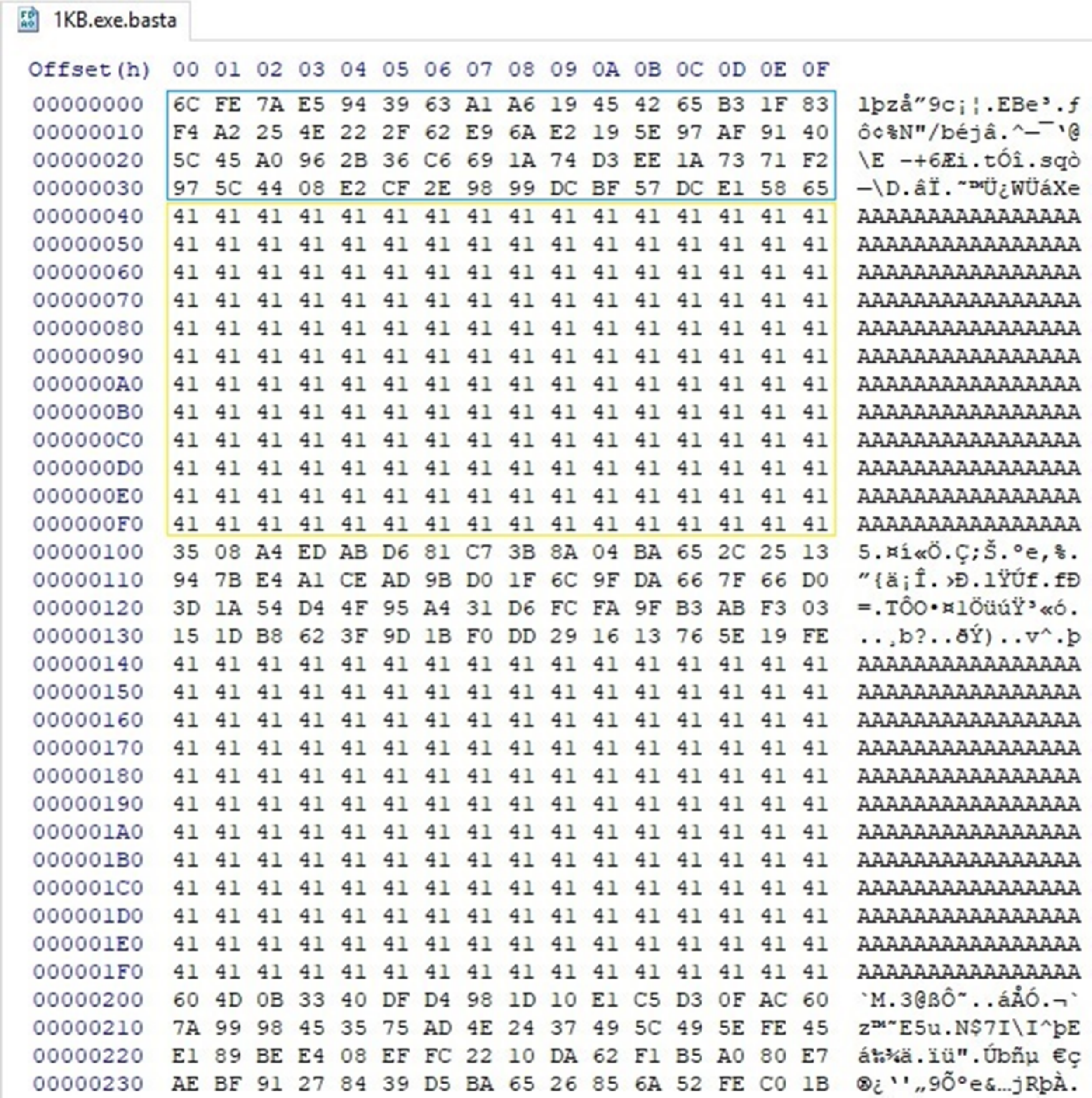


```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  5B C1 DC 7B 0F DD 4B 0D 56 AC 5C 2B 73 FC 89 2A  [ÁÜ{.ÝK.V¬\+sü‰*
00000010  DE FD 1F 9F 4A 2A 68 AB B5 FB 7C 8D B6 87 C8 E3  Þý.ŸJ*h«µû|.¶‡Èã
00000020  6C 74 77 2F F6 95 2D 8D 65 4C 99 79 A4 CA F8 74  ltw/ö•-.eL™y¤Êøt
00000030  0D D1 8C 35 A7 A2 1C E3 DF 21 F9 C6 B3 97 D3 BF  .ÑŒ5§¢.ãß!ùÆ³—Ó¿
00000040  34 89 77 B5 EA C9 AD 3C A9 FA AB 80 A7 CE 7A E7  4‰wµêÉ.<©ú«€§Îzç
00000050  7E 4F 67 DE 41 36 51 BB DF 43 99 69 7F 15 55 08  ~OgÞA6Q»ßC™i..U.
00000060  4C D9 8B 2B 77 F8 D9 59 7A A9 70 38 A5 1B 4B 79  LÙ‹+wøÙYz©p8¥.Ky
00000070  FD 5D FE 71 56 F3 AF 26 0A 29 50 16 54 52 09 66  ý]þqVó¯&.)P.TR.f
00000080  1C FE B9 93 BE F5 4B F2 86 92 A0 73 09 D3 49 E1  .þ¹“¾õKò†’ s.ÓIá
00000090  7C 26 BF 7D E6 BA E4 81 10 2D C3 B5 F5 E2 85 2B  |&¿}æºä..-Ãµõâ…+
000000A0  74 36 41 5E 7C EB AF 8A FE FB 80 F1 E8 57 8A 96  t6A^|ë¯Šþû€ñèWŠ–
000000B0  12 15 3F 28 33 60 9F 9D 0E 61 DC 18 7D 77 B4 88  ..?(3`Ÿ..aÜ.}w´ˆ
000000C0  78 45 8F E6 4B C6 EB 25 8A 74 C2 FE 49 D8 32 A6  xE.æKÆë%ŠtÂþIØ2¦
000000D0  46 43 E2 F6 A2 E9 D7 D7 F9 F1 51 D2 6E BE 6B 53  FCâö¢é××ùñQÒn¾kS
000000E0  36 B4 11 C7 6A DC 79 BD 9B C4 95 07 64 E9 B1 04  6´.ÇjÜy½›Ä•.dé±.
000000F0  3E 62 E9 A9 B1 0C A6 BC EC 3B 63 6D 0F BA D2 79  >bé©±.¦¼ì;cm.ºÒy
00000100  95 91 98 C9 A5 02 63 BC 93 8A DF DA 8B 79 A8 05  •‘˜É¥.c¼“ŠßÚ‹y¨.
00000110  A5 BB 5F 94 FC 74 5A B3 59 9E A5 D1 32 83 D7 5E  ¥»_”ütZ³Yž¥Ñ2ƒ×^
00000120  CB A0 8C 3A DE E5 84 6B B9 AB F7 8E ED 65 98 1B  Ë Œ:Þå„k¹«÷Žíe˜.
00000130  B0 2A 75 21 C7 AD DA FC 3C 86 11 05 77 06 DE B2  °*u!Ç.Úü<†..w.Þ²
00000140  31 BA B1 AF 01 BA 50 48 77 6A A0 A9 E2 93 83 AD  1º±¯.ºPHwj ©â“ƒ.
00000150  53 C3 99 1E EE 11 E0 C8 0C 15 24 5E 90 73 22 B7  SÃ™.î.àÈ..$^.s"·
00000160  C8 6D 65 09 0A 10 29 E4 E3 E6 8B 6D 85 A0 96 5E  Ème...)äãæ‹m… –^
00000170  FE D0 35 19 0D 07 D6 06 C1 A2 7D BA 16 2C B5 0D  þÐ5...Ö.Á¢}º.,µ.
00000180  3C 16 EF 4B F7 E0 02 FF 53 E8 A5 A9 9A 7B 9A 0F  <.ïK÷à.ÿSè¥©š{š.
00000190  49 9D C2 3E 3A 30 5A 7A 72 A1 9C E7 28 69 A5 B0  I.Â>:0Zzr¡œç(i¥°
000001A0  8A ED 7A EE BC 07 7D AD D2 35 6D DD FC AE C6 8B  Šízî¼.}.Ò5mÝü®Æ‹
000001B0  D5 C8 92 76 3B 74 89 33 0C 03 D4 56 CD CE F8 D9  ÕÈ’v;t‰3..ÔVÍÎøÙ
000001C0  4B 60 30 B2 A9 49 A6 A5 A3 C0 A3 CC EB 39 1A CE  K`0²©I¦¥£À£Ìë9.Î
000001D0  EC 7F 52 75 9E 1C FC 55 48 95 78 9B B3 5C 7C B6  ì.Ruž.üUH•x›³\|¶
000001E0  CE AB 17 06 E6 A7 17 C0 30 D2 55 FD B1 6A A8 4D  Î«..æ§.À0ÒUý±j¨M
000001F0  1A D2 67 4C 94 57 CE CC 8A FA 53 D6 E1 09 94 AC  .ÒgL”WÎÌŠúSÖá.”¬
00000200  F8 B1 20 70 E5 23 79 BA 9B 9A F3 6C DE A6 13 89  ø± på#yº›šólÞ¦.‰
00000210  63 03 FA 71 C9 BA DD E7 15 88 7E 9D 4B 70 84 CB  c.úqÉºÝç.ˆ~.Kp„Ë
00000220  A2 59 DE 56 62 4E D0 1E DA DA 7A 4D AA 17 F1 F4  ¢YÞVbNÐ.ÚÚzMª.ñô
00000230  28 36 B1 E1 48 E8 5D 3E 34 93 BB 8E 87 B3 80 16  (6±áHè]>4“»Ž‡³€.
00000240  06 5B 66 8B 95 8F 71 47 9C 16 54 92 FA 78 15 11  .[f‹•.qGœ.T’úx..
00000250  09 10 9B C8 53 57 BB 34 D5 64 92 94 1D 92 96 35  ..›ÈSW»4Õd’”.’–5
00000260  CD 8D CB 6A 75 AB DC AE AE BB 3F 6D CA 03 E1 67  Í.Ëju«Ü®®»?mÊ.ág
00000270  FA 2F 8D 9D A8 A0 CF EF 32 17 20 A8 93 45 F7 B1  ú/..¨ Ïï2. ¨“E÷±
00000280  43 AF CF C1 15 2C D6 A9 31 5C 6C 5F FF 81 30 52  C¯ÏÁ.,Ö©1\l_ÿ.0R
00000290  FF C6 96 09 D7 1D 41 D7 B8 96 11 BB 74 64 C9 7D  ÿÆ–.×.A×¸–.»tdÉ}
000002A0  04 3F 72 F6 3E E1 A3 D5 D7 A8 10 DA 38 0F D1 84  .?rö>á£Õ×¨.Ú8.Ñ„
000002B0  4A 56 1F E2 AC 50 A6 B3 E1 AF 4B FD 67 3F F9 13  JV.â¬P¦³á¯Kýg?ù.
000002C0  89 57 B2 FC 1D B4 DC B1 A2 A6 DF 45 78 31 2E 16  ‰W²ü.´Ü±¢¦ßEx1..
000002D0  BD 07 4E FA A5 79 7E 2F 32 12 80 17 FA B6 8C 85  ½.Nú¥y~/2.€.ú¶Œ…
000002E0  00 02 00 00                                      ....
```

Figure 58

Case 2 – File size < 4KB

In this case, the file is partially encrypted. The ransomware encrypts 64 bytes, skips 192 bytes, encrypts 64 bytes again, and so on.



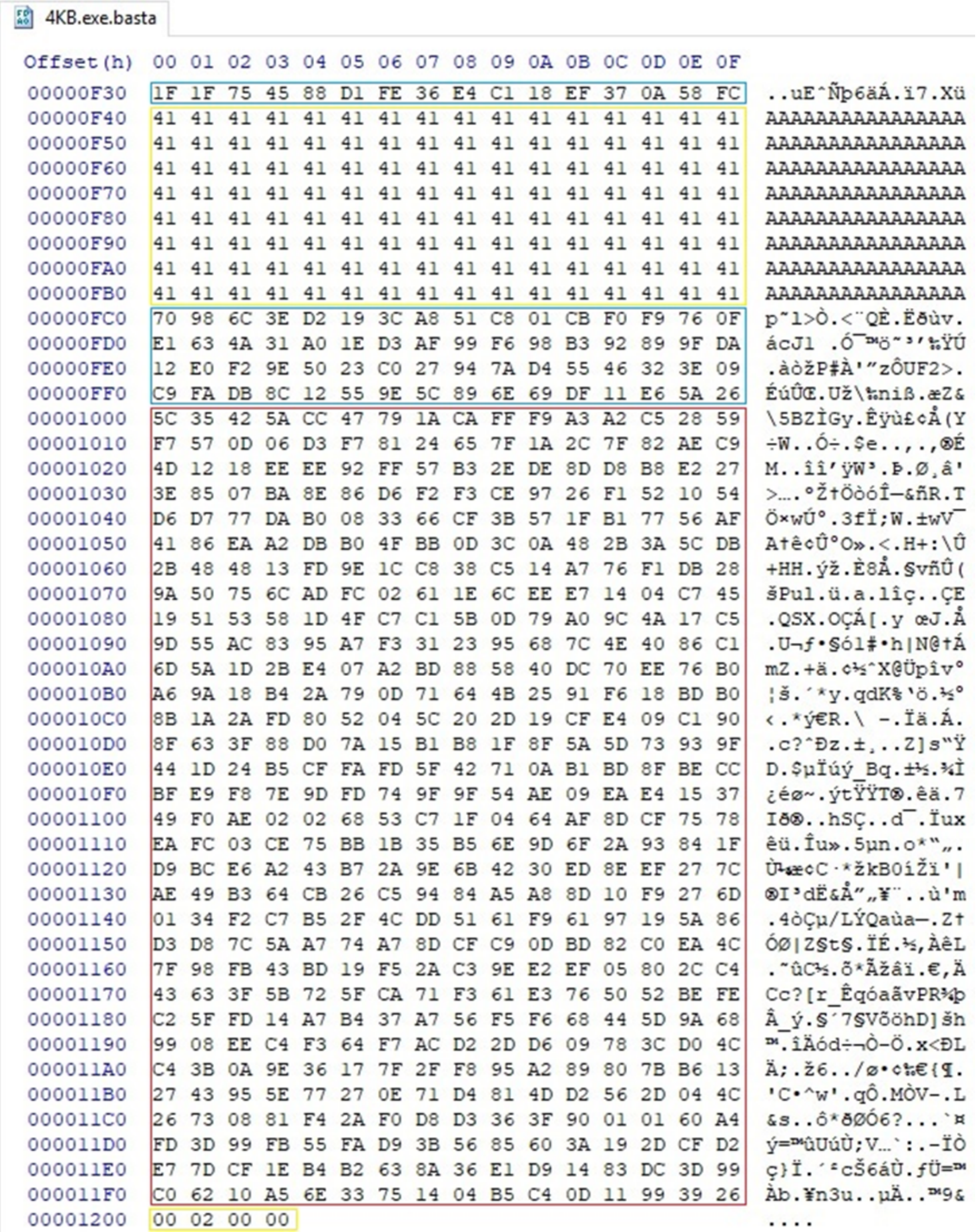
1KB.exe.basta

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  6C FE 7A E5 94 39 63 A1 A6 19 45 42 65 B3 1F 83  lþzå”9c¡¦.EBe³.ƒ
00000010  F4 A2 25 4E 22 2F 62 E9 6A E2 19 5E 97 AF 91 40  ô¢%N"/béjâ.^—¯‘@
00000020  5C 45 A0 96 2B 36 C6 69 1A 74 D3 EE 1A 73 71 F2  \E –+6Æi.tÓî.sqò
00000030  97 5C 44 08 E2 CF 2E 98 99 DC BF 57 DC E1 58 65  —\D.âÏ.˜™Ü¿WÜáXe
00000040  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000050  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000060  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000070  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000080  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000090  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000000A0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000000B0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000000C0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000000D0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000000E0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000000F0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000100  35 08 A4 ED AB D6 81 C7 3B 8A 04 BA 65 2C 25 13  5.¤í«Ö.Ç;Š.ºe,%.
00000110  94 7B E4 A1 CE AD 9B D0 1F 6C 9F DA 66 7F 66 D0  ”{ä¡Î.›Ð.lŸÚf.fÐ
00000120  3D 1A 54 D4 4F 95 A4 31 D6 FC FA 9F B3 AB F3 03  =.TÔO•¤1ÖüúŸ³«ó.
00000130  15 1D B8 62 3F 9D 1B F0 DD 29 16 13 76 5E 19 FE  ..¸b?..ðÝ)..v^.þ
00000140  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000150  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000160  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000170  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000180  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000190  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000001A0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000001B0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000001C0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000001D0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000001E0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
000001F0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000200  60 4D 0B 33 40 DF D4 98 1D 10 E1 C5 D3 0F AC 60  `M.3@ßÔ˜..áÅÓ.¬`
00000210  7A 99 98 45 35 75 AD 4E 24 37 49 5C 49 5E FE 45  z™˜E5u.N$7I\I^þE
00000220  E1 89 BE E4 08 EF FC 22 10 DA 62 F1 B5 A0 80 E7  á‰¾ä.ïü".Úbñµ €ç
00000230  AE BF 91 27 84 39 D5 BA 65 26 85 6A 52 FE C0 1B  ®¿‘'„9Õºe&…jRþÀ.
```

Figure 59

Case 3 – File size > 4KB

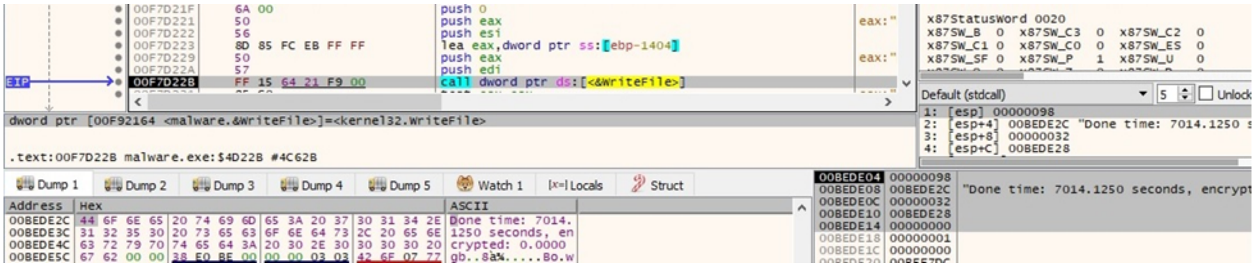
In this case, the file is partially encrypted. The ransomware encrypts 64 bytes, skips 128 bytes, encrypts 64 bytes again, and so on.



```
4KB.exe.basta

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000F30  1F 1F 75 45 88 D1 FE 36 E4 C1 18 EF 37 0A 58 FC  ..uE^Ñþ6äÁ.ï7.Xü
00000F40  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000F50  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000F60  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000F70  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000F80  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000F90  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000FA0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000FB0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000FC0  70 98 6C 3E D2 19 3C A8 51 C8 01 CB F0 F9 76 0F  p˜l>Ò.<¨QÈ.Ëðùv.
00000FD0  E1 63 4A 31 A0 1E D3 AF 99 F6 98 B3 92 89 9F DA  ácJ1 .Ó¯™ö˜³’‰ŸÚ
00000FE0  12 E0 F2 9E 50 23 C0 27 94 7A D4 55 46 32 3E 09  .àòžP#À'”zÔUF2>.
00000FF0  C9 FA DB 8C 12 55 9E 5C 89 6E 69 DF 11 E6 5A 26  ÉúÛŒ.Už\‰niß.æZ&
00001000  5C 35 42 5A CC 47 79 1A CA FF F9 A3 A2 C5 28 59  \5BZÌGy.Êÿù£¢Å(Y
00001010  F7 57 0D 06 D3 F7 81 24 65 7F 1A 2C 7F 82 AE C9  ÷W..Ó÷.$e..,.‚®É
00001020  4D 12 18 EE EE 92 FF 57 B3 2E DE 8D D8 B8 E2 27  M..îî’ÿW³.Þ.Ø¸â'
00001030  3E 85 07 BA 8E 86 D6 F2 F3 CE 97 26 F1 52 10 54  >….ºŽ†ÖòóÎ—&ñR.T
00001040  D6 D7 77 DA B0 08 33 66 CF 3B 57 1F B1 77 56 AF  Ö×wÚ°.3fÏ;W.±wV¯
00001050  41 86 EA A2 DB B0 4F BB 0D 3C 0A 48 2B 3A 5C DB  A†ê¢Û°O».<.H+:\Û
00001060  2B 48 48 13 FD 9E 1C C8 38 C5 14 A7 76 F1 DB 28  +HH.ýž.È8Å.§vñÛ(
00001070  9A 50 75 6C AD FC 02 61 1E 6C EE E7 14 04 C7 45  šPul.ü.a.lîç..ÇE
00001080  19 51 53 58 1D 4F C7 C1 5B 0D 79 A0 9C 4A 17 C5  .QSX.OÇÁ[.y œJ.Å
00001090  9D 55 AC 83 95 A7 F3 31 23 95 68 7C 4E 40 86 C1  .U¬ƒ•§ó1#•h|N@†Á
000010A0  6D 5A 1D 2B E4 07 A2 BD 88 58 40 DC 70 EE 76 B0  mZ.+ä.¢½ˆX@Üpîv°
000010B0  A6 9A 18 B4 2A 79 0D 71 64 4B 25 91 F6 18 BD B0  ¦š.´*y.qdK%‘ö.½°
000010C0  8B 1A 2A FD 80 52 04 5C 20 2D 19 CF E4 09 C1 90  ‹.*ý€R.\ -.Ïä.Á.
000010D0  8F 63 3F 88 D0 7A 15 B1 B8 1F 8F 5A 5D 73 93 9F  .c?ˆÐz.±¸..Z]s“Ÿ
000010E0  44 1D 24 B5 CF FA FD 5F 42 71 0A B1 BD 8F BE CC  D.$µÏúý_Bq.±½.¾Ì
000010F0  BF E9 F8 7E 9D FD 74 9F 9F 54 AE 09 EA E4 15 37  ¿éø~.ýtŸŸT®.êä.7
00001100  49 F0 AE 02 02 68 53 C7 1F 04 64 AF 8D CF 75 78  Ið®..hSÇ..d¯.Ïux
00001110  EA FC 03 CE 75 BB 1B 35 B5 6E 9D 6F 2A 93 84 1F  êü.Îu».5µn.o*“„.
00001120  D9 BC E6 A2 43 B7 2A 9E 6B 42 30 ED 8E EF 27 7C  Ù¼æ¢C·*žkB0íŽï'|
00001130  AE 49 B3 64 CB 26 C5 94 84 A5 A8 8D 10 F9 27 6D  ®I³dË&Å”„¥¨..ù'm
00001140  01 34 F2 C7 B5 2F 4C DD 51 61 F9 61 97 19 5A 86  .4òÇµ/LÝQaùa—.Z†
00001150  D3 D8 7C 5A A7 74 A7 8D CF C9 0D BD 82 C0 EA 4C  ÓØ|Z§t§.ÏÉ.½‚ÀêL
00001160  7F 98 FB 43 BD 19 F5 2A C3 9E E2 EF 05 80 2C C4  .˜ûC½.õ*ÃžâïĀ€,Ä
00001170  43 63 3F 5B 72 5F CA 71 F3 61 E3 76 50 52 BE FE  Cc?[r_ÊqóaãvPR¾þ
00001180  C2 5F FD 14 A7 B4 37 A7 56 F5 F6 68 44 5D 9A 68  Â_ý.§´7§VõöhD]šh
00001190  99 08 EE C4 F3 64 F7 AC D2 2D D6 09 78 3C D0 4C  ™.îÄód÷¬Ò-Ö.x<ÐL
000011A0  C4 3B 0A 9E 36 17 7F 2F F8 95 A2 89 80 7B B6 13  Ä;.ž6../ø•¢‰€{¶.
000011B0  27 43 95 5E 77 27 0E 71 D4 81 4D D2 56 2D 04 4C  'C•^w'.qÔ.MÒV-.L
000011C0  26 73 08 81 F4 2A F0 D8 D3 36 3F 90 01 01 60 A4  &s..ô*ðØÓ6?...`¤
000011D0  FD 3D 99 FB 55 FA D9 3B 56 85 60 3A 19 2D CF D2  ý=™ûUúÙ;V…`:.-ÏÒ
000011E0  E7 7D CF 1E B4 B2 63 8A 36 E1 D9 14 83 DC 3D 99  ç}Ï.´²cŠ6áÙ.ƒÜ=™
000011F0  C0 62 10 A5 6E 33 75 14 04 B5 C4 0D 11 99 39 26  Àb.¥n3u..µÄ..™9&
00001200  00 02 00 00                                      ....
```

Figure 60

Finally, the ransomware tries to write the time spent during the execution and the total size of encrypted files to the console; however, it raises an error because the process was detached from its console:



```
00F7D21F   6A 00              push 0
00F7D221   50                 push eax                              eax:"
00F7D222   56                 push esi
00F7D223   8D 85 FC EB FF FF  lea eax,dword ptr ss:[ebp-1404]
00F7D229   50                 push eax                              eax:"
00F7D22A   57                 push edi
00F7D22B   FF 15 64 21 F9 00  call dword ptr ds:[<&WriteFile>]
```

```
dword ptr [00F92164 <malware.&WriteFile>]=<kernel32.WriteFile>

.text:00F7D22B malware.exe:$4D22B #4C62B
```

```
x87StatusWord 0020
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_ES  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0
```

Default (stdcall) 5 □ Unlock

```
1: [esp] 00000098
2: [esp+4] 00BEDE2C "Done time: 7014.1250 s
3: [esp+8] 00000032
4: [esp+C] 00BEDE28
```

```
Address   Hex                                              ASCII
00BEDE2C  44 6F 6E 65 20 74 69 6D 65 3A 20 37 30 31 34 2E  Done time: 7014.
00BEDE3C  31 32 35 30 20 73 65 63 6F 6E 64 73 2C 20 65 6E  1250 seconds, en
00BEDE4C  63 72 79 70 74 65 64 3A 20 30 2E 30 30 30 30 20  crypted: 0.0000
00BEDE5C  67 62 00 00 38 E0 BE 00 00 00 03 03 42 6F 07 77  gb..8à¾.....Bo.w
```

```
00BEDE04  00000098
00BEDE08  00BEDE2C  "Done time: 7014.1250 seconds, encrypt
00BEDE0C  00000032
00BEDE10  00BEDE28
00BEDE14  00000000
00BEDE18  00000001
00BEDE1C  00000000
```

Figure 61

Indicators of Compromise

Black Basta Ransom Note

readme.txt

Files created

%Temp%\fkdjsadasd.ico

%Temp%\dlaksjdoiwq.jpg

Processes spawned

cmd.exe /c “C:\Windows\SysNative\vssadmin.exe delete shadows /all /quiet”

cmd.exe /c “C:\Windows\System32\vssadmin.exe delete shadows /all /quiet”

Registry key created

HKEY_CLASSES_ROOT\.basta