CoinMiner

TECHNICAL ANALYSIS REPORT

ZAYOTEM

ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

Table of Contents

Preview

Golfstikator.exe is a malware from the CoinMiner family. A Coin Miner is a type of malware that uses hardware elements of the victim's computer to mine. It uses high CPU mining on infected computers, causing slowdowns and errors in the computer. Most of the time, attackers controlling the CoinMiner malware strain target coins called Monero (XMR) or Litecoin because they are the easiest to mine. Computers infected with Golfstikator malware;

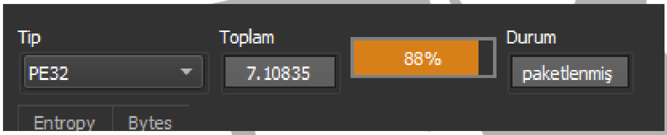
1. Computer resources,
2. Services, records and passwords on the computer,
3. Computer network and security software,
4. It allows them to access computer documents.

Golfstikator.exe Analysis

Name	golfstikator.exe
MD5	743fc0d22063e7ea97ca753280ff9f3e
SHA256	5c9051c7d3b4658cb635429f8644ed682bf23cf10f237b75d07cd5e74f86ba2
File Type	PE32/EXE

Static Analysis

When the Golfstikator.exe malware is analyzed with the DIE tool, it is seen that it is packaged.



Tip: PE32

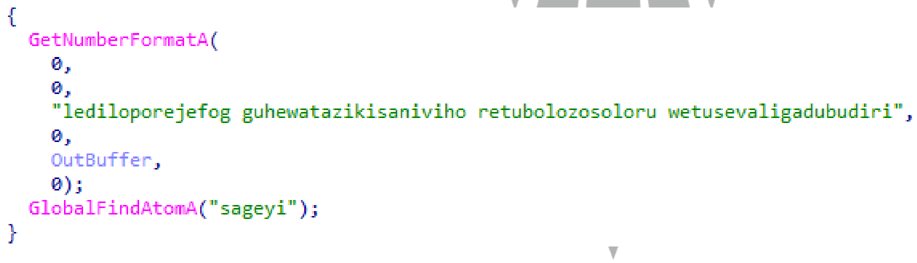
Toplam: 7.10835

88%

Durum: paketlenmiş

Figure 1- Entropy value

When the malware's codes are examined statically, dysfunctional and ineffective API calls are encountered, as seen in Figure 2. It appears that these API calls and gibberish strings are trying to make the code harder to analyze.

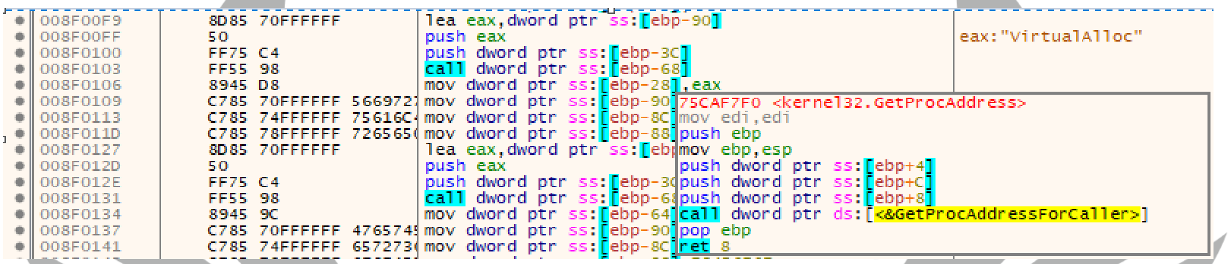


```
{
  GetNumberFormatA(
    0,
    0,
    "lediloporejefog guhewatazikisaniviho retubolozosoloru wetusevaligadubudiri",
    0,
    OutBuffer,
    0);
  GlobalFindAtomA("sageyi");
}
```

Figure 2- Dysfunctional Api calls

Dynamic Analysis

Various techniques were used in the dynamic analysis part of the Golfstikator.exe malware to make analysis difficult. It is understood that the **Dynamic API Resolution** technique was used when examining the malware.

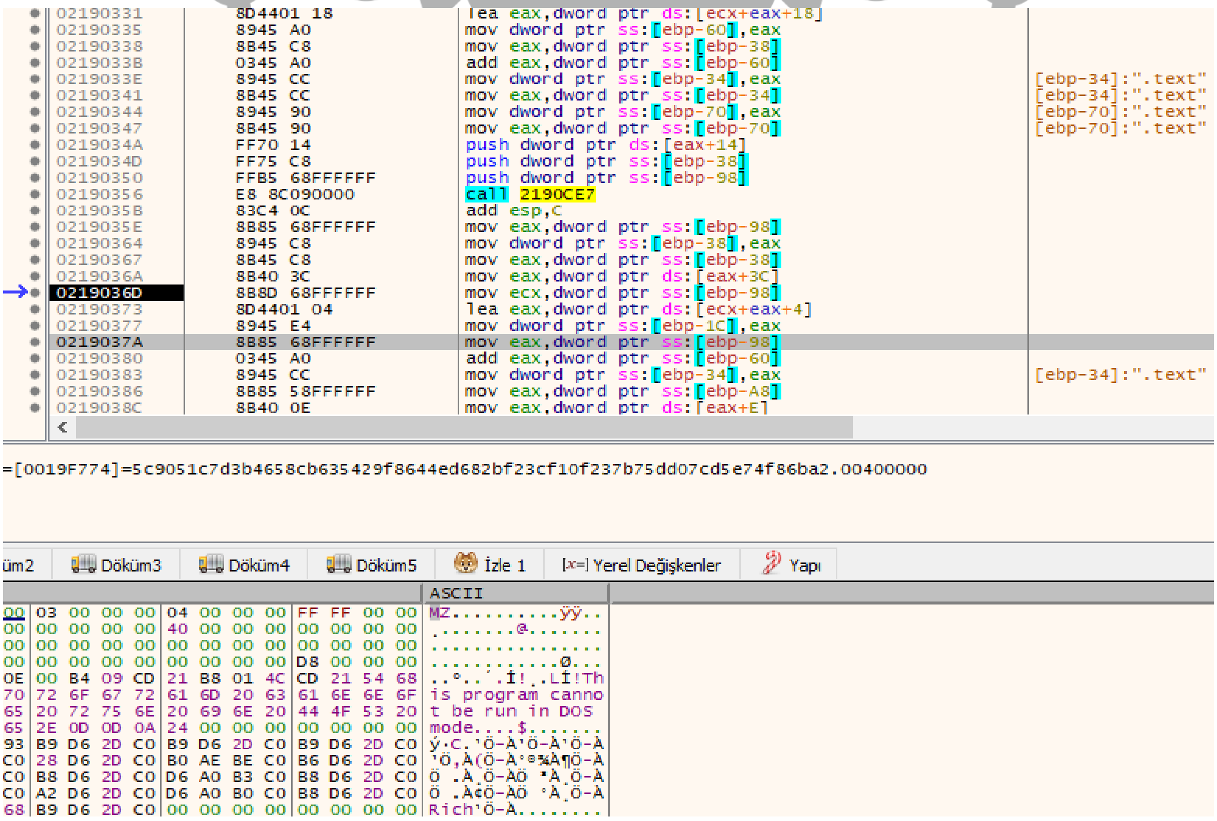


```
008F00F9   8D85 70FFFFFF            lea eax,dword ptr ss:[ebp-90]
008F00FF   50                       push eax                             eax:"VirtualAlloc"
008F0100   FF75 C4                  push dword ptr ss:[ebp-3C]
008F0103   FF55 98                  call dword ptr ss:[ebp-68]
008F0106   8945 D8                  mov dword ptr ss:[ebp-28],eax
008F0109   C785 70FFFFFF 569727     mov dword ptr ss:[ebp-90
008F0113   C785 74FFFFFF 75616C     mov dword ptr ss:[ebp-8C
008F011D   C785 78FFFFFF 726565     mov dword ptr ss:[ebp-88
008F0127   8D85 70FFFFFF            lea eax,dword ptr ss:[ebp
008F012D   50                       push eax
008F012E   FF75 C4                  push dword ptr ss:[ebp-3C
008F0131   FF55 98                  call dword ptr ss:[ebp-6
008F0134   8945 9C                  mov dword ptr ss:[ebp-64
008F0137   C785 70FFFFFF 476574     mov dword ptr ss:[ebp-90
008F0141   C785 74FFFFFF 657273     mov dword ptr ss:[ebp-8C
```

```
75CAF7F0 <kernel32.GetProcAddress>
mov edi,edi
push ebp
mov ebp,esp
push dword ptr ss:[ebp+4]
push dword ptr ss:[ebp+C]
push dword ptr ss:[ebp+8]
call dword ptr ds:[<&GetProcAddressForCaller>]
pop ebp
ret 8
```

Figure 3 - Resolution with GetProcAdrress

During the dynamic analysis, it is noticed that the malware produces a new malware using the self-modification technique. It is understood that the PE sections were created by writing them separately.



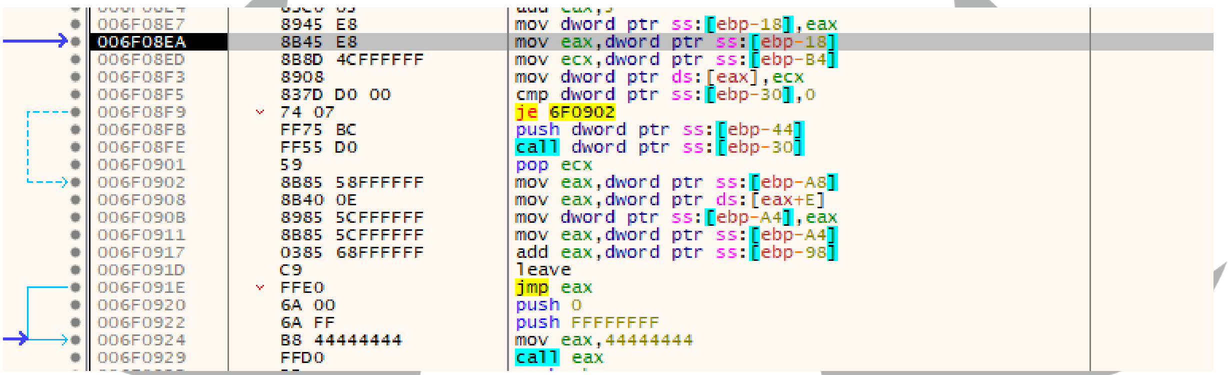
```
   02190331   8D4401 18          lea eax,dword ptr ds:[ecx+eax+18]
   02190335   8945 A0            mov dword ptr ss:[ebp-60],eax
   02190338   8B45 C8            mov eax,dword ptr ss:[ebp-38]
   0219033B   0345 A0            add eax,dword ptr ss:[ebp-60]
   0219033E   8945 CC            mov dword ptr ss:[ebp-34],eax     [ebp-34]:".text"
   02190341   8B45 CC            mov eax,dword ptr ss:[ebp-34]     [ebp-34]:".text"
   02190344   8945 90            mov dword ptr ss:[ebp-70],eax     [ebp-70]:".text"
   02190347   8B45 90            mov eax,dword ptr ss:[ebp-70]     [ebp-70]:".text"
   0219034A   FF70 14            push dword ptr ds:[eax+14]
   0219034D   FF75 C8            push dword ptr ss:[ebp-38]
   02190350   FFB5 68FFFFFF      push dword ptr ss:[ebp-98]
   02190356   E8 8C090000        call 2190CE7
   0219035B   83C4 0C            add esp,C
   0219035E   8B85 68FFFFFF      mov eax,dword ptr ss:[ebp-98]
   02190364   8945 C8            mov dword ptr ss:[ebp-38],eax
   02190367   8B45 C8            mov eax,dword ptr ss:[ebp-38]
   0219036A   8B40 3C            mov eax,dword ptr ds:[eax+3C]
-> 0219036D   8B8D 68FFFFFF      mov ecx,dword ptr ss:[ebp-98]
   02190373   8D4401 04          lea eax,dword ptr ds:[ecx+eax+4]
   02190377   8945 E4            mov dword ptr ss:[ebp-1C],eax
   0219037A   8B85 68FFFFFF      mov eax,dword ptr ss:[ebp-98]
   02190380   0345 A0            add eax,dword ptr ss:[ebp-60]
   02190383   8945 CC            mov dword ptr ss:[ebp-34],eax     [ebp-34]:".text"
   02190386   8B85 58FFFFFF      mov eax,dword ptr ss:[ebp-A8]
   0219038C   8B40 0E            mov eax,dword ptr ds:[eax+E]
```

```
=[0019F774]=5c9051c7d3b4658cb635429f8644ed682bf23cf10f237b75dd07cd5e74f86ba2.00400000
```

```
                                                      ASCII
00 | 03 00 00 00 | 04 00 00 00 | FF FF 00 00 | MZ..........ÿÿ..
00 | 00 00 00 00 | 40 00 00 00 | 00 00 00 00 | ........@.......
00 | 00 00 00 00 | 00 00 00 00 | 00 00 00 00 | ................
00 | 00 00 00 00 | 00 00 00 00 | D8 00 00 00 | ............Ø...
0E | 00 B4 09 CD | 21 B8 01 4C | CD 21 54 68 | ..º..´.Í!¸.LÍ!Th
70 | 72 6F 67 72 | 61 6D 20 63 | 61 6E 6E 6F | is program canno
65 | 20 72 75 6E | 20 69 6E 20 | 44 4F 53 20 | t be run in DOS 
65 | 2E 0D 0D 0A | 24 00 00 00 | 00 00 00 00 | mode....$.......
93 | B9 D6 2D C0 | B9 D6 2D C0 | B9 D6 2D C0 | ý·C.¹Ö-À¹Ö-À¹Ö-À
C0 | 28 D6 2D C0 | B0 AE BE C0 | B6 D6 2D C0 | ¹Ö,À(Ö-À°®¾À¶Ö-À
C0 | B8 D6 2D C0 | D6 A0 B3 C0 | B8 D6 2D C0 | Ö .À¸Ö-ÀÖ ³À¸Ö-À
C0 | A2 D6 2D C0 | D6 A0 B0 C0 | B8 D6 2D C0 | Ö .À¢Ö-ÀÖ °À¸Ö-À
68 | B9 D6 2D C0 | 00 00 00 00 | 00 00 00 00 | Rich¹Ö-À........
```

Figure 4 - Writing the sections of the new malware

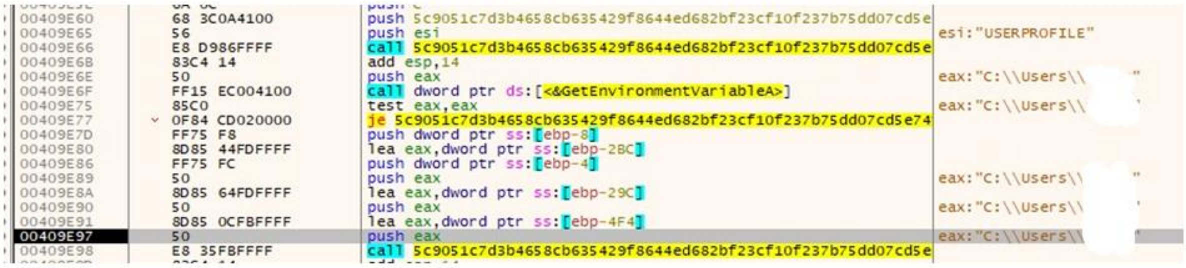
After the "section" is created, the address of the new malware is written with the **[jmp eax]** command.



```
006F08E7   8945 E8          mov dword ptr ss:[ebp-18],eax
006F08EA   8B45 E8          mov eax,dword ptr ss:[ebp-18]
006F08ED   8B8D 4CFFFFFF    mov ecx,dword ptr ss:[ebp-B4]
006F08F3   8908             mov dword ptr ds:[eax],ecx
006F08F5   837D D0 00       cmp dword ptr ss:[ebp-30],0
006F08F9   74 07            je 6F0902
006F08FB   FF75 BC          push dword ptr ss:[ebp-44]
006F08FE   FF55 D0          call dword ptr ss:[ebp-30]
006F0901   59               pop ecx
006F0902   8B85 58FFFFFF    mov eax,dword ptr ss:[ebp-A8]
006F0908   8B40 0E          mov eax,dword ptr ds:[eax+E]
006F090B   8985 5CFFFFFF    mov dword ptr ss:[ebp-A4],eax
006F0911   8B85 5CFFFFFF    mov eax,dword ptr ss:[ebp-A4]
006F0917   0385 68FFFFFF    add eax,dword ptr ss:[ebp-98]
006F091D   C9               leave
006F091E   FFE0             jmp eax
006F0920   6A 00            push 0
006F0922   6A FF            push FFFFFFFF
006F0924   B8 44444444      mov eax,44444444
006F0929   FFD0             call eax
```

Figure 5- The part where the address of the created malware is jumped to

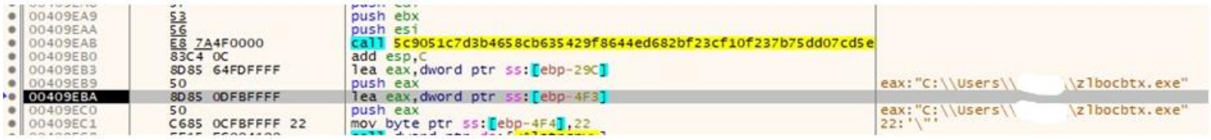
In order to record the malware to be created, environmental variables and path control are carried out.



```
00409E60   68 3C0A4100      push 5c9051c7d3b4658cb635429f8644ed682bf23cf10f237b75dd07cd5e
00409E65   56               push esi                                   esi:"USERPROFILE"
00409E66   E8 D986FFFF      call 5c9051c7d3b4658cb635429f8644ed682bf23cf10f237b75dd07cd5e
00409E6B   83C4 14          add esp,14
00409E6E   50               push eax                                   eax:"C:\\Users\\[redacted]"
00409E6F   FF15 EC004100    call dword ptr ds:[<&GetEnvironmentVariableA>]   eax:"C:\\Users\\[redacted]"
00409E75   85C0             test eax,eax
00409E77   0F84 CD020000    je 5c9051c7d3b4658cb635429f8644ed682bf23cf10f237b75dd07cd5e74
00409E7D   FF75 F8          push dword ptr ss:[ebp-8]
00409E80   8D85 44FDFFFF    lea eax,dword ptr ss:[ebp-2BC]
00409E86   FF75 FC          push dword ptr ss:[ebp-4]
00409E89   50               push eax                                   eax:"C:\\Users\\[redacted]"
00409E8A   8D85 64FDFFFF    lea eax,dword ptr ss:[ebp-29C]
00409E90   50               push eax                                   eax:"C:\\Users\\[redacted]"
00409E91   8D85 0CFBFFFF    lea eax,dword ptr ss:[ebp-4F4]
00409E97   50               push eax                                   eax:"C:\\Users\\[redacted]"
00409E98   E8 35FBFFFF      call 5c9051c7d3b4658cb635429f8644ed682bf23cf10f237b75dd07cd5e
```

Figure 6 - Path control

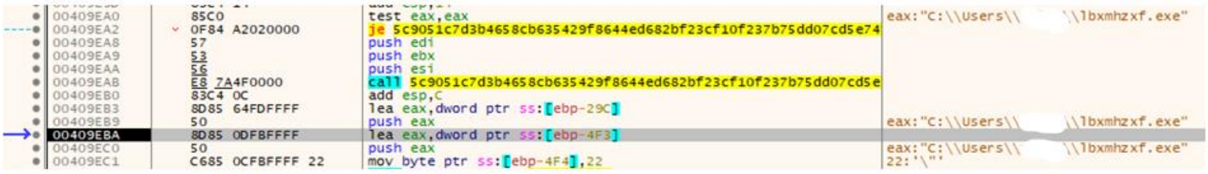
After the control is achieved, the malware is saved under the **"C:\\Users\\UserName"** folder.



```
00409EA9   53               push ebx
00409EAA   56               push esi
00409EAB   E8 7A4F0000      call 5c9051c7d3b4658cb635429f8644ed682bf23cf10f237b75dd07cd5e
00409EB0   83C4 0C          add esp,C
00409EB3   8D85 64FDFFFF    lea eax,dword ptr ss:[ebp-29C]
00409EB9   50               push eax                          eax:"C:\\Users\\[redacted]\zlbocbtx.exe"
00409EBA   8D85 0DFBFFFF    lea eax,dword ptr ss:[ebp-4F3]
00409EC0   50               push eax                          eax:"C:\\Users\\[redacted]\zlbocbtx.exe"
00409EC1   C685 0CFBFFFF 22 mov byte ptr ss:[ebp-4F4],22      22:'\"'
```

Figure 7 - The folder where the malware was created

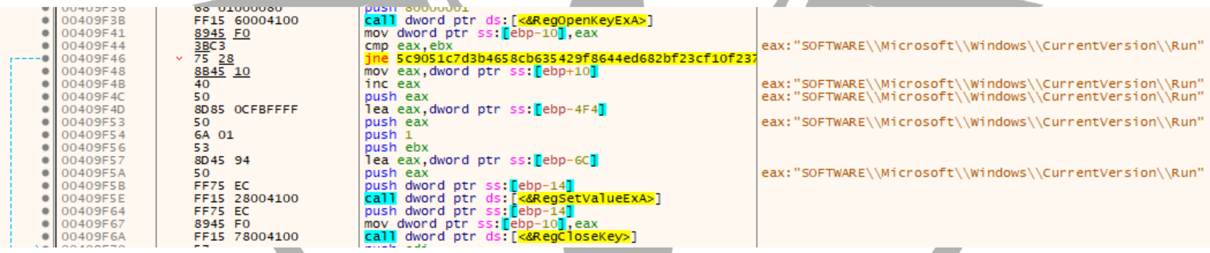
In addition, it is observed that it produces the same malware with random names every time. Another example is shown in the screen shot.



```
00409EA0    85C0               test eax,eax                                                eax:"C:\\Users\\       \\lbxmhzxf.exe"
00409EA2    0F84 A2020000      je 5c9051c7d3b4658cb635429f8644ed682bf23cf10f237b75dd07cd5e74
00409EA8    57                 push edi
00409EA9    53                 push ebx
00409EAA    56                 push esi
00409EAB    E8 7A4F0000        call 5c9051c7d3b4658cb635429f8644ed682bf23cf10f237b75dd07cd5e
00409EB0    83C4 0C            add esp,C
00409EB3    8D85 64FDFFFF      lea eax,dword ptr ss:[ebp-29C]
00409EB9    50                 push eax                                                    eax:"C:\\Users\\       \\lbxmhzxf.exe"
00409EBA    8D85 0DFBFFFF      lea eax,dword ptr ss:[ebp-4F3]
00409EC0    50                 push eax                                                    eax:"C:\\Users\\       \\lbxmhzxf.exe"
00409EC1    C685 0CFBFFFF 22   mov byte ptr ss:[ebp-4F4],22                                22:'\"'
```

Figure 8- Creating the malware in the same folder with other names

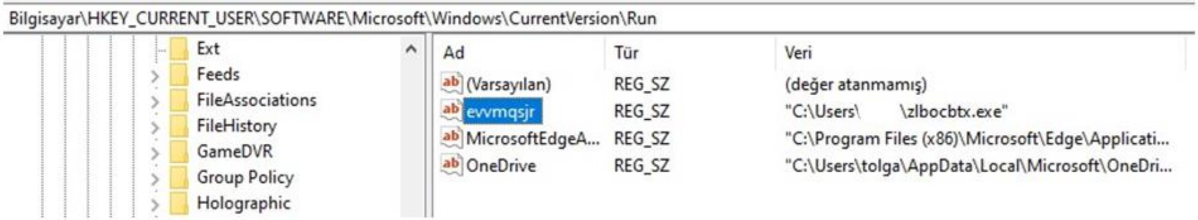
After the new malware is created, it is registered at the registry address in the path "Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" so that it can run every time the computer is restarted **RegOpenKeyExA**, **RegSetValueExA** and **RegCloseKey** APIs are used to manipulate registry records.



```
00409F3B    FF15 60004100      call dword ptr ds:[<&RegOpenKeyExA>]
00409F41    8945 F0            mov dword ptr ss:[ebp-10],eax
00409F44    3BC3               cmp eax,ebx                            eax:"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run"
00409F46    75 28              jne 5c9051c7d3b4658cb635429f8644ed682bf23cf10f237
00409F48    8B45 10            mov eax,dword ptr ss:[ebp+10]
00409F4B    40                 inc eax                                eax:"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run"
00409F4C    50                 push eax                               eax:"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run"
00409F4D    8D85 0CFBFFFF      lea eax,dword ptr ss:[ebp-4F4]
00409F53    50                 push eax                               eax:"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run"
00409F54    6A 01              push 1
00409F56    53                 push ebx
00409F57    8D45 94            lea eax,dword ptr ss:[ebp-6C]
00409F5A    50                 push eax                               eax:"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run"
00409F5B    FF75 EC            push dword ptr ss:[ebp-14]
00409F5E    FF15 28004100      call dword ptr ds:[<&RegSetValueExA>]
00409F64    FF75 EC            push dword ptr ss:[ebp-14]
00409F67    8945 F0            mov dword ptr ss:[ebp-10],eax
00409F6A    FF15 78004100      call dword ptr ds:[<&RegCloseKey>]
```

Figure 9 - Saving in the Run folder for persistence

The values in the Run folder are indicated in Figure-10.

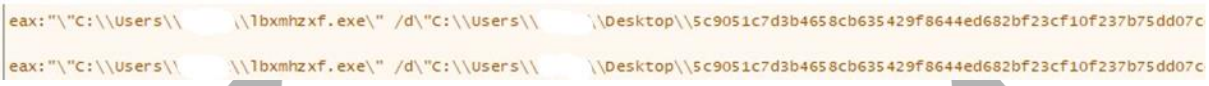


Bilgisayar\HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Ad	Tür	Veri
(Varsayılan)	REG_SZ	(değer atanmamış)
evvmqsjr	REG_SZ	"C:\Users\ \zlbocbtx.exe"
MicrosoftEdgeA...	REG_SZ	"C:\Program Files (x86)\Microsoft\Edge\Applicati...
OneDrive	REG_SZ	"C:\Users\tolga\AppData\Local\Microsoft\OneDri...

Figure 10- The image of the Run folder

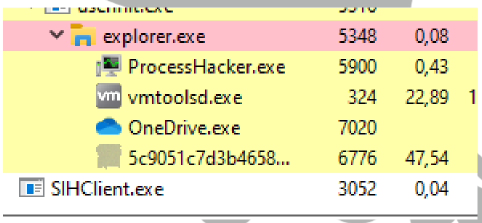
At the end of the process, the new malware is successfully created. After the new malware is created, the Golfstikator malware runs the new malware with the **"/d"** and **"/f"** parameters, deletes itself, and ensures its permanence by saving it in the registry.



```
eax:"\"C:\\Users\\        \\lbxmhzxf.exe\" /d\"C:\\Users\\        \\Desktop\\5c9051c7d3b4658cb635429f8644ed682bf23cf10f237b75dd07c
eax:"\"C:\\Users\\        \\lbxmhzxf.exe\" /d\"C:\\Users\\        \\Desktop\\5c9051c7d3b4658cb635429f8644ed682bf23cf10f237b75dd07c
```

Figure 11- Running the created malware with parameters

When the malware is examined with Process Hacker, it is examined that it consumes 47% of CPU.



```
    explorer.exe              5348    0,08
      ProcessHacker.exe       5900    0,43
      vmtoolsd.exe             324   22,89   1
      OneDrive.exe            7020
      5c9051c7d3b4658...      6776   47,54
  SIHClient.exe               3052    0,04
```

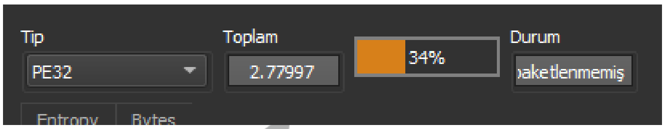
Figure 12- CPU consumption of the malware

Stage 2 Analysis

Name	-
MD5	71d1f08703f6d940c3b2f88f811d792b
SHA256	E5D60C81A634C00C8C1861EF260D71810D1BCA6294D8542598F664B260075FD8
File Type	PE32/EXE

Static Analysis

In Stage-2, when the malware is thrown into the DIE tool and analyzed, it is seen that it is not packaged. Following the static analysis, it is understood that it is almost the same as the Golfstikator malware and that this malware will benefit from various dynamic analyses.

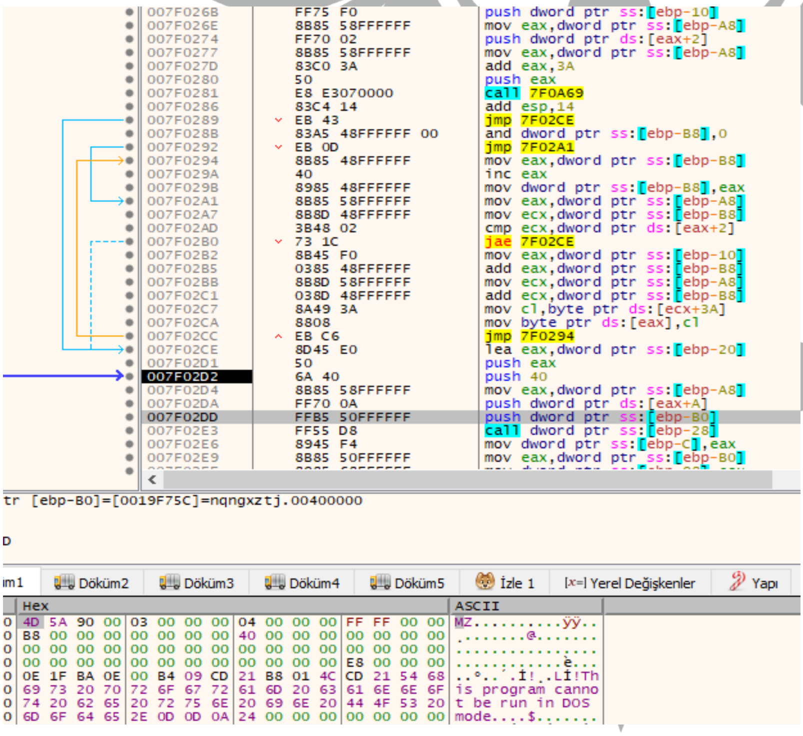


Tip: PE32
Toplam: 2.77997
34%
Durum: paketlenmemiş

Şekil 13-Entropy değeri

Dynamic Analysis

In the analysis of Stage-2, there is no finding different from the Golfstikator malware. Since no findings were found, a "dump" was taken from the region specified in Figure-14 and it was deemed appropriate to continue the analysis on the "dump".



```
007F026B   FF75 F0            push dword ptr ss:[ebp-10]
007F026E   8B85 58FFFFFF      mov eax,dword ptr ss:[ebp-A8]
007F0274   FF70 02            push dword ptr ds:[eax+2]
007F0277   8B85 58FFFFFF      mov eax,dword ptr ss:[ebp-A8]
007F027D   83C0 3A            add eax,3A
007F0280   50                 push eax
007F0281   E8 E3070000        call 7F0A69
007F0286   83C4 14            add esp,14
007F0289   EB 43              jmp 7F02CE
007F028B   83A5 48FFFFFF 00   and dword ptr ss:[ebp-B8],0
007F0292   EB 0D              jmp 7F02A1
007F0294   8B85 48FFFFFF      mov eax,dword ptr ss:[ebp-B8]
007F029A   40                 inc eax
007F029B   8985 48FFFFFF      mov dword ptr ss:[ebp-B8],eax
007F02A1   8B85 58FFFFFF      mov eax,dword ptr ss:[ebp-A8]
007F02A7   8B8D 48FFFFFF      mov ecx,dword ptr ss:[ebp-B8]
007F02AD   3B48 02            cmp ecx,dword ptr ds:[eax+2]
007F02B0   73 1C              jae 7F02CE
007F02B2   8B45 F0            mov eax,dword ptr ss:[ebp-10]
007F02B5   0385 48FFFFFF      add eax,dword ptr ss:[ebp-B8]
007F02BB   8B8D 58FFFFFF      mov ecx,dword ptr ss:[ebp-A8]
007F02C1   038D 48FFFFFF      add ecx,dword ptr ss:[ebp-B8]
007F02C7   8A49 3A            mov cl,byte ptr ds:[ecx+3A]
007F02CA   8808               mov byte ptr ds:[eax],cl
007F02CC   EB C6              jmp 7F0294
007F02CE   8D45 E0            lea eax,dword ptr ss:[ebp-20]
007F02D1   50                 push eax
007F02D2   6A 40              push 40
007F02D4   8B85 58FFFFFF      mov eax,dword ptr ss:[ebp-A8]
007F02DA   FF70 0A            push dword ptr ds:[eax+A]
007F02DD   FFB5 50FFFFFF      push dword ptr ss:[ebp-B0]
007F02E3   FF55 D8            call dword ptr ss:[ebp-28]
007F02E6   8945 F4            mov dword ptr ss:[ebp-C],eax
007F02E9   8B85 50FFFFFF      mov eax,dword ptr ss:[ebp-B0]

tr [ebp-B0]=[0019F75C]=nqngxztj.00400000

D
```

```
  Hex                                                ASCII
0 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
0 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0 00 00 00 00 00 00 00 00 00 00 00 00 E8 00 00 00  ............è...
0 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
0 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
0 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
0 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
```

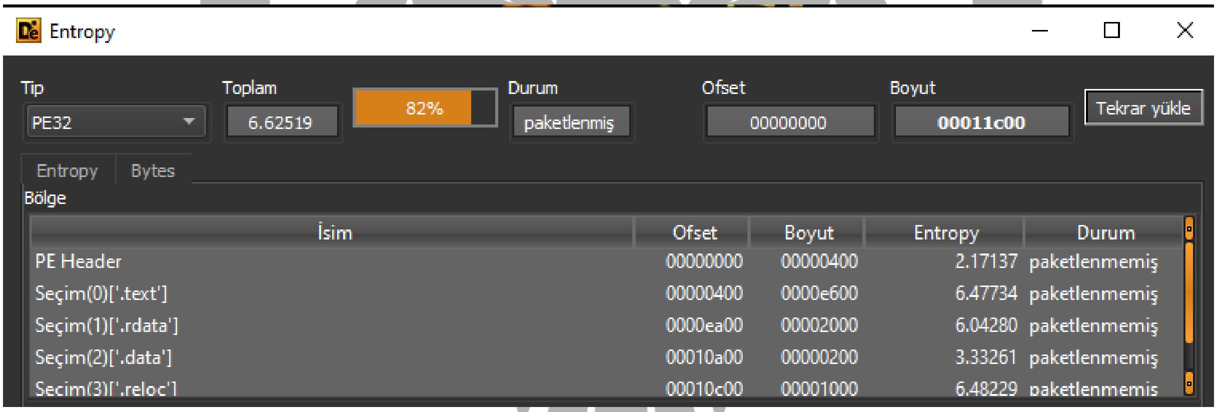
Figure 14- Loop in which Sections are written

Stage-3 Analysis

Name	-
MD5	A050f3c88055b70ddf52d04747d4f527
SHA256	ca07ed841c430fedf79b2696148963cc5c5c989641e40aa34c022d4685e8ba3e
File Type	PE32/EXE

Static Analysis

When the "dump" is examined with the DIE tool, it is seen that it is packaged due to its entropy value, but when the sections are examined, it is understood that it is not packaged.



Tip: PE32
Toplam: 6.62519
82%
Durum: paketlenmiş
Ofset: 00000000
Boyut: 00011c00

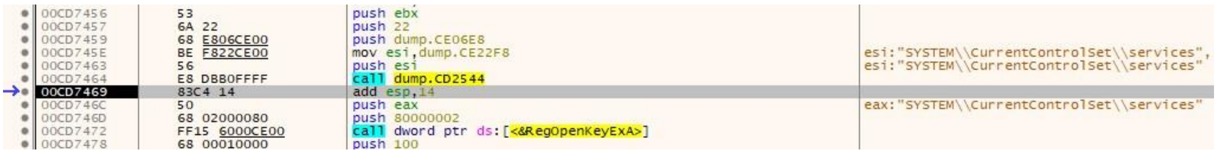
Bölge

İsim	Ofset	Boyut	Entropy	Durum
PE Header	00000000	00000400	2.17137	paketlenmemiş
Seçim(0)['.text']	00000400	0000e600	6.47734	paketlenmemiş
Seçim(1)['.rdata']	0000ea00	00002000	6.04280	paketlenmemiş
Seçim(2)['.data']	00010a00	00000200	3.33261	paketlenmemiş
Seçim(3)['.reloc']	00010c00	00001000	6.48229	paketlenmemiş

Figure 15- Entropy value

Dynamic Analysis

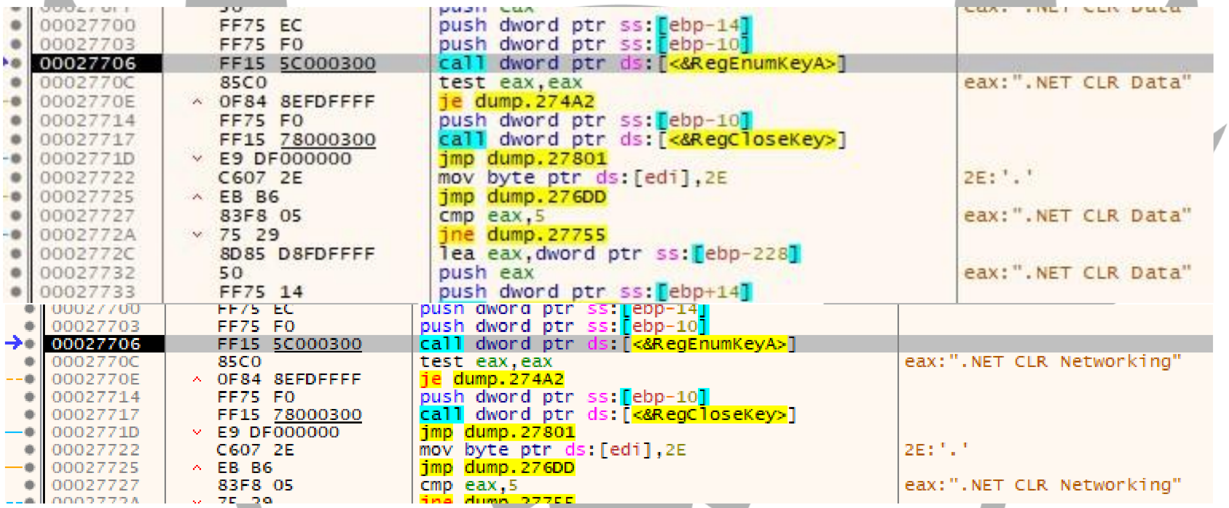
When the Stage-3 dump is examined, it is seen that various manipulations are made in the registry and the information of the system services is reached.



```
00CD7456    53              push ebx
00CD7457    6A 22           push 22
00CD7459    68 E806CE00     push dump.CE06E8
00CD745E    BE F822CE00     mov esi,dump.CE22F8                     esi:"SYSTEM\\CurrentControlSet\\services",
00CD7463    56              push esi                                esi:"SYSTEM\\CurrentControlSet\\services"
00CD7464    E8 DBB0FFFF     call dump.CD2544
00CD7469    83C4 14         add esp,14
00CD746C    50              push eax                                eax:"SYSTEM\\CurrentControlSet\\services"
00CD746D    68 02000080     push 80000002
00CD7472    FF15 6000CE00   call dword ptr ds:[<&RegOpenKeyExA>]
00CD7478    68 00010000     push 100
```

Figure 16 - Service check from registry addresses

As the investigations continue, it appears that the .NET service in the registry can access various records.

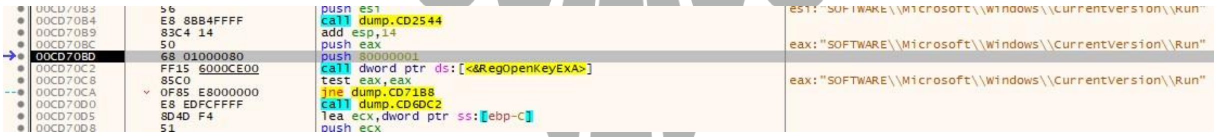


```
00027700    FF75 EC          push dword ptr ss:[ebp-14]
00027703    FF75 F0          push dword ptr ss:[ebp-10]
00027706    FF15 5C000300    call dword ptr ds:[<&RegEnumKeyA>]
0002770C    85C0             test eax,eax                           eax:".NET CLR Data"
0002770E    0F84 8EFDFFFF    je dump.274A2
00027714    FF75 F0          push dword ptr ss:[ebp-10]
00027717    FF15 78000300    call dword ptr ds:[<&RegCloseKey>]
0002771D    E9 DF000000      jmp dump.27801
00027722    C607 2E          mov byte ptr ds:[edi],2E               2E:'.'
00027725    EB B6            jmp dump.276DD
00027727    83F8 05          cmp eax,5                              eax:".NET CLR Data"
0002772A    75 29            jne dump.27755
0002772C    8D85 D8FDFFFF    lea eax,dword ptr ss:[ebp-228]
00027732    50               push eax                               eax:".NET CLR Data"
00027733    FF75 14          push dword ptr ss:[ebp+14]
```

```
00027700    FF75 EC          push dword ptr ss:[ebp-14]
00027703    FF75 F0          push dword ptr ss:[ebp-10]
00027706    FF15 5C000300    call dword ptr ds:[<&RegEnumKeyA>]
0002770C    85C0             test eax,eax                           eax:".NET CLR Networking"
0002770E    0F84 8EFDFFFF    je dump.274A2
00027714    FF75 F0          push dword ptr ss:[ebp-10]
00027717    FF15 78000300    call dword ptr ds:[<&RegCloseKey>]
0002771D    E9 DF000000      jmp dump.27801
00027722    C607 2E          mov byte ptr ds:[edi],2E               2E:'.'
00027725    EB B6            jmp dump.276DD
00027727    83F8 05          cmp eax,5                              eax:".NET CLR Networking"
```

Figure 17 - Records of .NET services

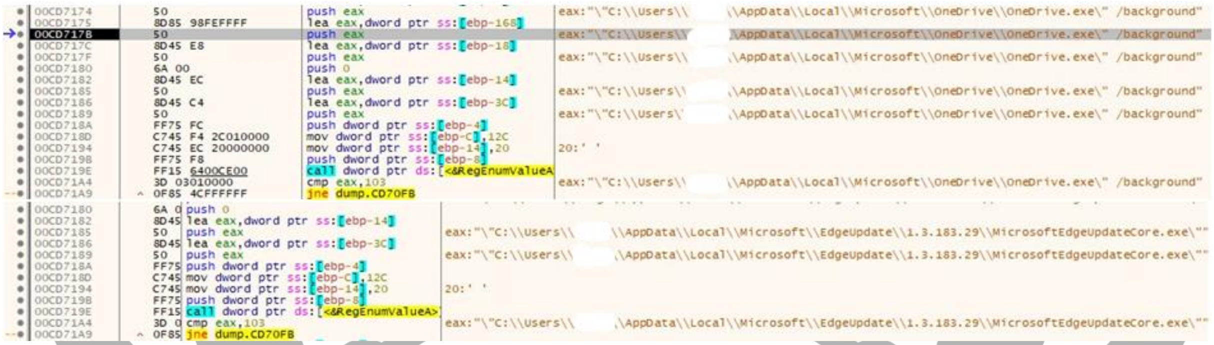
Later, it is observed that it accesses the **"Current Version\Run"** records that contain the startup services.



```
00CD70B3    56              push esi                                esi:"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run"
00CD70B4    E8 8BB4FFFF     call dump.CD2544
00CD70B9    83C4 14         add esp,14
00CD70BC    50              push eax                                eax:"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run"
00CD70BD    68 01000080     push 80000001
00CD70C2    FF15 6000CE00   call dword ptr ds:[<&RegOpenKeyExA>]
00CD70C8    85C0            test eax,eax                            eax:"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run"
00CD70CA    0F85 E8000000   jne dump.CD71B8
00CD70D0    E8 EDFCFFFF     call dump.CD6DC2
00CD70D5    8D4D F4         lea ecx,dword ptr ss:[ebp-C]
00CD70D8    51              push ecx
```

Figure 18- Accessing the Run folder from the registry

Continuing the investigation, it is observed that it investigates Windows applications such as OneDrive and Microsoft Edge.

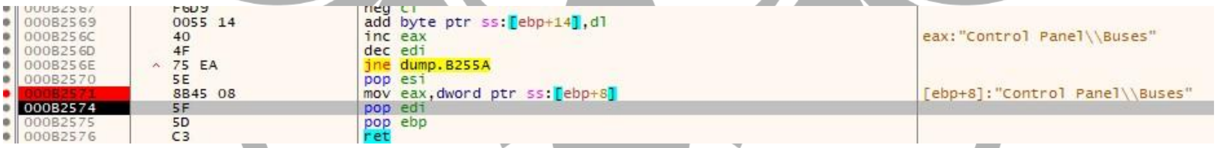


```
00CD7174    50                   push eax                                eax:"\"C:\\Users\\[redacted]\\AppData\\Local\\Microsoft\\OneDrive\\OneDrive.exe\" /background"
00CD7175    8D85 98FEFFFF        lea eax,dword ptr ss:[ebp-168]
00CD717B    50                   push eax                                eax:"\"C:\\Users\\[redacted]\\AppData\\Local\\Microsoft\\OneDrive\\OneDrive.exe\" /background"
00CD717C    8D45 E8              lea eax,dword ptr ss:[ebp-18]
00CD717F    50                   push eax                                eax:"\"C:\\Users\\[redacted]\\AppData\\Local\\Microsoft\\OneDrive\\OneDrive.exe\" /background"
00CD7180    6A 00                push 0
00CD7182    8D45 EC              lea eax,dword ptr ss:[ebp-14]
00CD7185    50                   push eax                                eax:"\"C:\\Users\\[redacted]\\AppData\\Local\\Microsoft\\OneDrive\\OneDrive.exe\" /background"
00CD7186    8D45 C4              lea eax,dword ptr ss:[ebp-3C]
00CD7189    50                   push eax                                eax:"\"C:\\Users\[redacted]\\AppData\\Local\\Microsoft\\OneDrive\\OneDrive.exe\" /background"
00CD718A    FF75 FC              push dword ptr ss:[ebp-4]
00CD718D    C745 F4 2C010000     mov dword ptr ss:[ebp-C],12C
00CD7194    C745 EC 20000000     mov dword ptr ss:[ebp-14],20            20:' '
00CD719B    FF75 F8              push dword ptr ss:[ebp-8]
00CD719E    FF15 6400CE00        call dword ptr ds:[<&RegEnumValueA
00CD71A4    3D 03010000          cmp eax,103                             eax:"\"C:\\Users\\[redacted]\\AppData\\Local\\Microsoft\\OneDrive\\OneDrive.exe\" /background"
00CD71A9  ^ 0F85 4CFFFFFF        jne dump.CD70FB

00CD7180    6A 0 push 0
00CD7182    8D45 lea eax,dword ptr ss:[ebp-14]
00CD7185    50   push eax                     eax:"\"C:\\Users\\[redacted]\\AppData\\Local\\Microsoft\\EdgeUpdate\\1.3.183.29\\MicrosoftEdgeUpdateCore.exe\""
00CD7186    8D45 lea eax,dword ptr ss:[ebp-3C]
00CD7189    50   push eax                     eax:"\"C:\\Users\\[redacted]\\AppData\\Local\\Microsoft\\EdgeUpdate\\1.3.183.29\\MicrosoftEdgeUpdateCore.exe\""
00CD718A    FF75 push dword ptr ss:[ebp-4]
00CD718D    C745 mov dword ptr ss:[ebp-C],12C
00CD7194    C745 mov dword ptr ss:[ebp-14],20     20:' '
00CD719B    FF75 push dword ptr ss:[ebp-8]
00CD719E    FF15 call dword ptr ds:[<&RegEnumValueA>]
00CD71A4    3D 0 cmp eax,103                  eax:"\"C:\\Users\\[redacted]\\AppData\\Local\\Microsoft\\EdgeUpdate\\1.3.183.29\\MicrosoftEdgeUpdateCore.exe\""
00CD71A9  ^ 0F85 jne dump.CD70FB
```

Figure 19- Microsoft Edge folder control

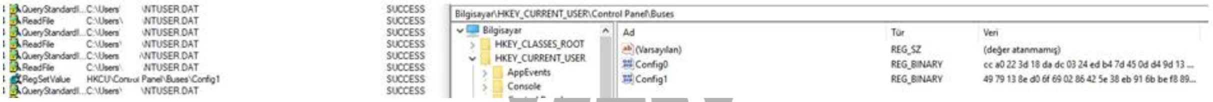
In the **" Control Panel\\Buses_"** records, it is seen that a new value has been created and meaningless string assignments have been made to it.



```
000B2567    F6D9        neg cl
000B2569    0055 14     add byte ptr ss:[ebp+14],dl
000B256C    40          inc eax                      eax:"Control Panel\\Buses"
000B256D    4F          dec edi
000B256E  ^ 75 EA       jne dump.B255A
000B2570    5E          pop esi
000B2571    8B45 08     mov eax,dword ptr ss:[ebp+8] [ebp+8]:"Control Panel\\Buses"
000B2574    5F          pop edi
000B2575    5D          pop ebp
000B2576    C3          ret
```

Figure 20 - Creating a new record in the registry

The values created in the registry are shown in Figure-23.



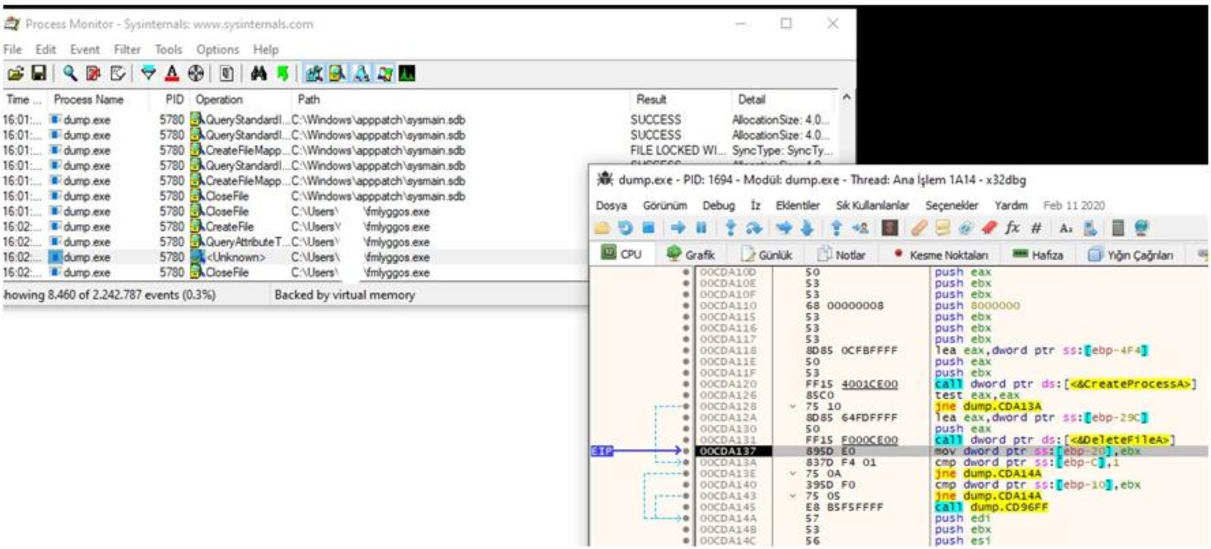
```
QueryStandardI... C:\Users\[redacted]\NTUSER.DAT          SUCCESS
ReadFile          C:\Users\[redacted]\NTUSER.DAT          SUCCESS
QueryStandardI... C:\Users\[redacted]\NTUSER.DAT          SUCCESS
ReadFile          C:\Users\[redacted]\NTUSER.DAT          SUCCESS
QueryStandardI... C:\Users\[redacted]\NTUSER.DAT          SUCCESS
ReadFile          C:\Users\[redacted]\NTUSER.DAT          SUCCESS
RegSetValue       HKCU\Control Panel\Buses\Config1        SUCCESS
QueryStandardI... C:\Users\[redacted]\NTUSER.DAT          SUCCESS
```

Bilgisayar\HKEY_CURRENT_USER\Control Panel\Buses

Ad	Tür	Veri
(Varsayılan)	REG_SZ	(değer atanmamış)
Config0	REG_BINARY	cc a0 22 3d 18 da dc 03 24 ed b4 7d 45 0d d4 9d 13 ...
Config1	REG_BINARY	49 79 13 8e d0 6f 69 02 86 42 5e 38 eb 91 6b be f8 89...

Figure 21 - Generated record values

It is examined that the malware creates a new process with the CreateProcessA API and saves it in the **" C:\\Users\UserName "** directory and then deletes this process using **DeleteFileA**, but this process is reported as an unknown process in Process Monitor.



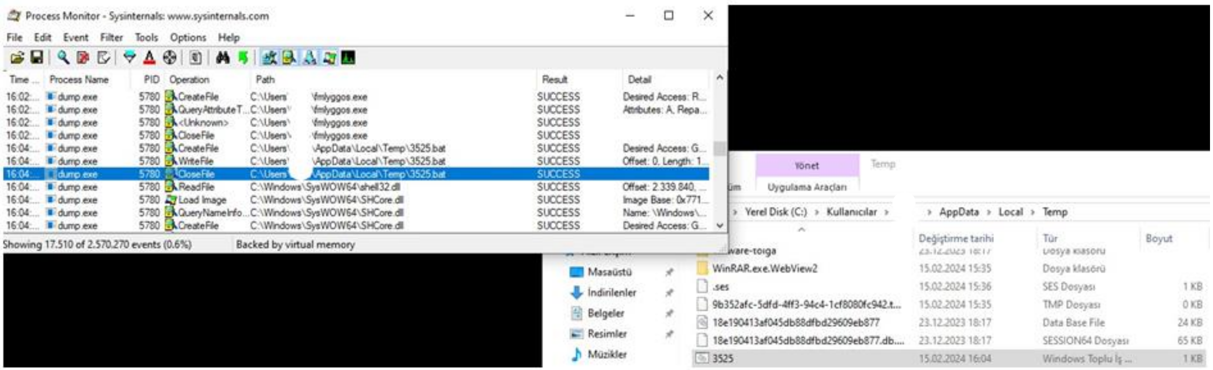
```
Process Monitor - Sysinternals: www.sysinternals.com
Time ...  Process Name  PID   Operation          Path                                   Result           Detail
16:01:... dump.exe      5780  QueryStandardI...  C:\Windows\apppatch\sysmain.sdb         SUCCESS          AllocationSize: 4.0...
16:01:... dump.exe      5780  QueryStandardI...  C:\Windows\apppatch\sysmain.sdb         SUCCESS          AllocationSize: 4.0...
16:01:... dump.exe      5780  CreateFileMapp...  C:\Windows\apppatch\sysmain.sdb         FILE LOCKED WI...  SyncType: SyncTy...
16:01:... dump.exe      5780  QueryStandardI...  C:\Windows\apppatch\sysmain.sdb
16:01:... dump.exe      5780  CreateFileMapp...  C:\Windows\apppatch\sysmain.sdb
16:01:... dump.exe      5780  CloseFile          C:\Windows\apppatch\sysmain.sdb
16:01:... dump.exe      5780  CloseFile          C:\Users\[redacted]\fmlyggos.exe
16:02:... dump.exe      5780  CreateFile         C:\Users\[redacted]\fmlyggos.exe
16:02:... dump.exe      5780  QueryAttributeT... C:\Users\[redacted]\fmlyggos.exe
16:02:... dump.exe      5780  <Unknown>          C:\Users\[redacted]\fmlyggos.exe
16:02:... dump.exe      5780  CloseFile          C:\Users\[redacted]\fmlyggos.exe
howing 8.460 of 2.242.787 events (0.3%)     Backed by virtual memory
```

```
dump.exe - PID: 1694 - Modül: dump.exe - Thread: Ana İşlem 1A14 - x32dbg
Dosya  Görünüm  Debug  İz  Eklentiler  Sık Kullanılanlar  Seçenekler  Yardım  Feb 11 2020
CPU  Grafik  Günlük  Notlar  Kesme Noktaları  Hafıza  Yığın Çağrıları

    00CDA10D   50              push eax
    00CDA10E   53              push ebx
    00CDA10F   53              push ebx
    00CDA110   68 00000008     push 8000000
    00CDA115   53              push ebx
    00CDA116   53              push ebx
    00CDA117   53              push ebx
    00CDA118   8D85 0CFBFFFF   lea eax,dword ptr ss:[ebp-4F4]
    00CDA11E   50              push eax
    00CDA11F   53              push ebx
    00CDA120   FF15 4001CE00   call dword ptr ds:[<&CreateProcessA>]
    00CDA126   85C0            test eax,eax
    00CDA128   75 10           jne dump.CDA13A
    00CDA12A   8D85 64FDFFFF   lea eax,dword ptr ss:[ebp-29C]
    00CDA130   50              push eax
    00CDA131   FF15 F000CE00   call dword ptr ds:[<&DeleteFileA>]
EIP 00CDA137   895D E0         mov dword ptr ss:[ebp-20],ebx
    00CDA13A   837D F4 01      cmp dword ptr ss:[ebp-C],1
    00CDA13E   75 0A           jne dump.CDA14A
    00CDA140   395D F0         cmp dword ptr ss:[ebp-10],ebx
    00CDA143   75 05           jne dump.CDA14A
    00CDA145   E8 B5F5FFFF     call dump.CD96FF
    00CDA14A   57              push edi
    00CDA14B   53              push ebx
    00CDA14C   56              push esi
```

Figure 22- Process image of the deletion of the malware in the Procmon and Debugger tool

When the investigations continued, it was understood that he created a file with bat extension named 3525 in the **\Temp** directory and then deleted the dump file.

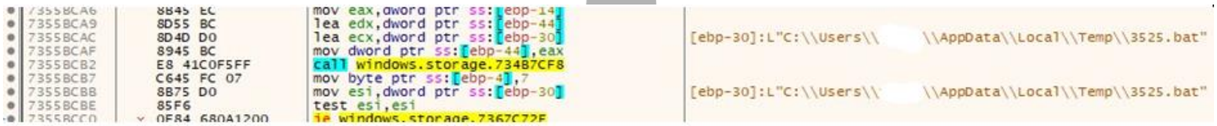


```
Process Monitor - Sysinternals: www.sysinternals.com
Time ...  Process Name  PID   Operation          Path                                                Result   Detail
16:02:... dump.exe      5780  CreateFile         C:\Users\[redacted]\fmlyggos.exe                    SUCCESS  Desired Access: R...
16:02:... dump.exe      5780  QueryAttributeT... C:\Users\[redacted]\fmlyggos.exe                    SUCCESS  Attributes: A, Repa...
16:02:... dump.exe      5780  <Unknown>          C:\Users\[redacted]\fmlyggos.exe                    SUCCESS
16:02:... dump.exe      5780  CloseFile          C:\Users\[redacted]\fmlyggos.exe                    SUCCESS
16:04:... dump.exe      5780  CreateFile         C:\Users\[redacted]\AppData\Local\Temp\3525.bat     SUCCESS  Desired Access: G...
16:04:... dump.exe      5780  WriteFile          C:\Users\[redacted]\AppData\Local\Temp\3525.bat     SUCCESS  Offset: 0, Length: 1...
16:04:... dump.exe      5780  CloseFile          C:\Users\[redacted]\AppData\Local\Temp\3525.bat     SUCCESS
16:04:... dump.exe      5780  ReadFile           C:\Windows\SysWOW64\shell32.dll                     SUCCESS  Offset: 2.339.840, ...
16:04:... dump.exe      5780  Load Image         C:\Windows\SysWOW64\SHCore.dll                      SUCCESS  Image Base: 0x771...
16:04:... dump.exe      5780  QueryNameInfo...   C:\Windows\SysWOW64\SHCore.dll                      SUCCESS  Name: \Windows\...
16:04:... dump.exe      5780  CreateFile         C:\Windows\SysWOW64\SHCore.dll                      SUCCESS  Desired Access: G...
Showing 17.510 of 2.570.270 events (0.6%)     Backed by virtual memory
```

```
Yönet    Temp
Uygulama Araçları
› Yerel Disk (C:) › Kullanıcılar › [redacted] › AppData › Local › Temp

                                              Değiştirme tarihi   Tür                   Boyut
...ware-tolga                                 23.12.2023 18:17    Dosya klasörü
WinRAR.exe.WebView2                           15.02.2024 15:35    Dosya klasörü
.ses                                          15.02.2024 15:36    SES Dosyası           1 KB
9b352afc-5dfd-4ff3-94c4-1cf8080fc942.t...     15.02.2024 15:35    TMP Dosyası           0 KB
18e190413af045db88dfbd29609eb877              23.12.2023 18:17    Data Base File        24 KB
18e190413af045db88dfbd29609eb877.db....       23.12.2023 18:17    SESSION64 Dosyası     65 KB
3525                                          15.02.2024 16:04    Windows Toplu İş ...  1 KB
```

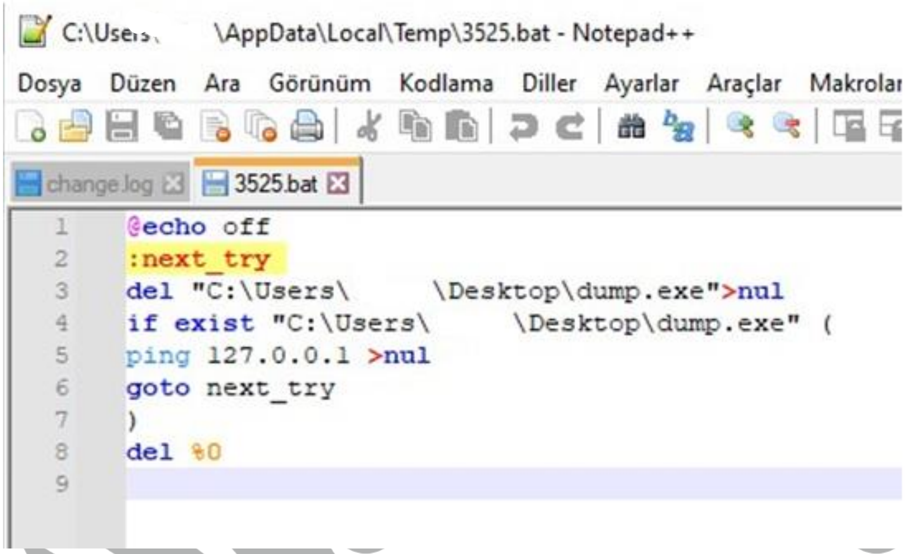
Figure 23- File with .bat extension created for deletion



```
7355BCA6   8B45 EC         mov eax,dword ptr ss:[ebp-14]
7355BCA9   8D55 BC         lea edx,dword ptr ss:[ebp-44]
7355BCAC   8D4D D0         lea ecx,dword ptr ss:[ebp-30]     [ebp-30]:L"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\3525.bat"
7355BCAF   8945 BC         mov dword ptr ss:[ebp-44],eax
7355BCB2   E8 41C0F5FF     call windows.storage.734B7CF8
7355BCB7   C645 FC 07      mov byte ptr ss:[ebp-4],7
7355BCBB   8B75 D0         mov esi,dword ptr ss:[ebp-30]     [ebp-30]:L"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\3525.bat"
7355BCBE   85F6            test esi,esi
7355BCC0   0F84 680A1200   je windows.storage.7367C72E
```

Figure 24- Checking the relevant file in the Debugger

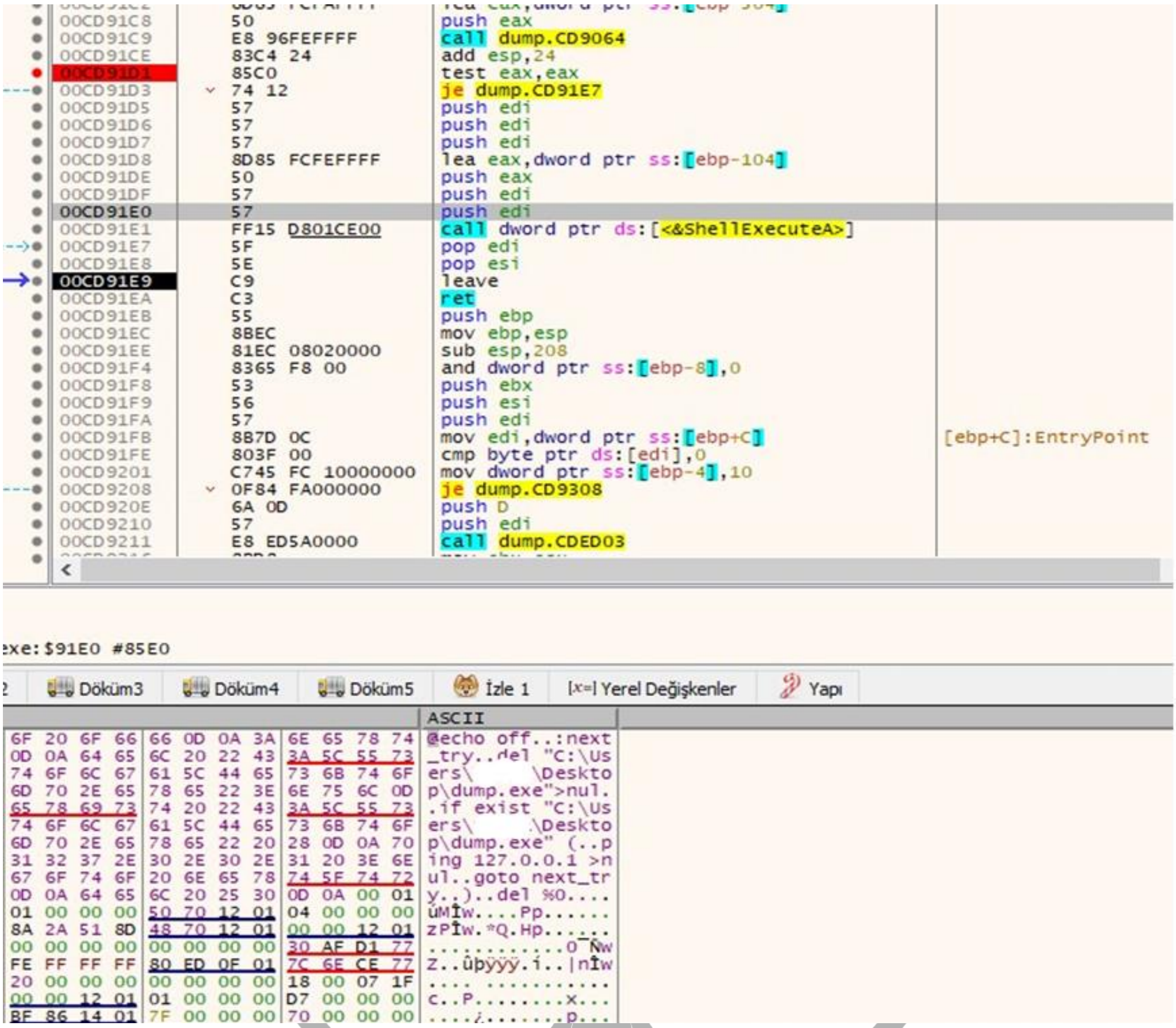
When the relevant file is examined, the script used to delete it is displayed as follows.


C:\Users\   \AppData\Local\Temp\3525.bat - Notepad++
Dosya Düzen Ara Görünüm Kodlama Diller Ayarlar Araçlar Makrolar
change.log
3525.bat

```
@echo off
:next_try
del "C:\Users\      \Desktop\dump.exe">nul
if exist "C:\Users\      \Desktop\dump.exe" (
ping 127.0.0.1 >nul
goto next_try
)
del %0
```

Figure 25- Contents of the file with .bat extension

When examined from the debugger, the .It is displayed that the file with the bat extension is run with the **ShellExecuteA** API.



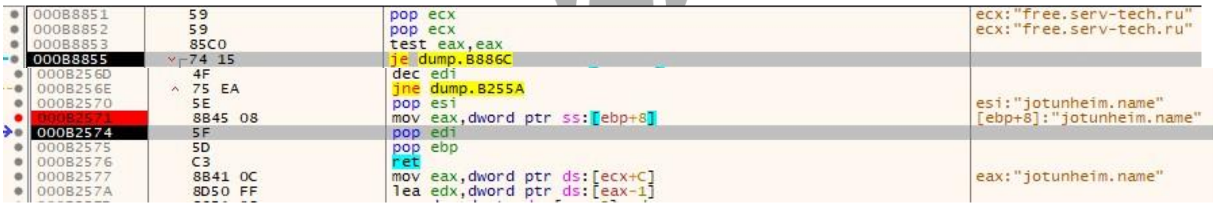
```
00CD91C8   50              push eax
00CD91C9   E8 96FEFFFF     call dump.CD9064
00CD91CE   83C4 24         add esp,24
00CD91D1   85C0            test eax,eax
00CD91D3   74 12           je dump.CD91E7
00CD91D5   57              push edi
00CD91D6   57              push edi
00CD91D7   57              push edi
00CD91D8   8D85 FCFEFFFF   lea eax,dword ptr ss:[ebp-104]
00CD91DE   50              push eax
00CD91DF   57              push edi
00CD91E0   57              push edi
00CD91E1   FF15 D801CE00   call dword ptr ds:[<&ShellExecuteA>]
00CD91E7   5F              pop edi
00CD91E8   5E              pop esi
00CD91E9   C9              leave
00CD91EA   C3              ret
00CD91EB   55              push ebp
00CD91EC   8BEC            mov ebp,esp
00CD91EE   81EC 08020000   sub esp,208
00CD91F4   8365 F8 00      and dword ptr ss:[ebp-8],0
00CD91F8   53              push ebx
00CD91F9   56              push esi
00CD91FA   57              push edi
00CD91FB   8B7D 0C         mov edi,dword ptr ss:[ebp+C]         [ebp+C]:EntryPoint
00CD91FE   803F 00         cmp byte ptr ds:[edi],0
00CD9201   C745 FC 10000000  mov dword ptr ss:[ebp-4],10
00CD9208   0F84 FA000000   je dump.CD9308
00CD920E   6A 0D           push D
00CD9210   57              push edi
00CD9211   E8 ED5A0000     call dump.CDED03

exe:$91E0 #85E0
```

```
                                                 ASCII
6F 20 6F 66 66 0D 0A 3A 6E 65 78 74   @echo off..:next
0D 0A 64 65 6C 20 22 43 3A 5C 55 73   _try..del "C:\Us
74 6F 6C 67 61 5C 44 65 73 6B 74 6F   ers\[redacted]\Deskto
6D 70 2E 65 78 65 22 3E 6E 75 6C 0D   p\dump.exe">nul.
65 78 69 73 74 20 22 43 3A 5C 55 73   .if exist "C:\Us
74 6F 6C 67 61 5C 44 65 73 6B 74 6F   ers\[redacted]\Deskto
6D 70 2E 65 78 65 22 20 28 0D 0A 70   p\dump.exe" (..p
31 32 37 2E 30 2E 30 2E 31 20 3E 6E   ing 127.0.0.1 >n
67 6F 74 6F 20 6E 65 78 74 5F 74 72   ul..goto next_tr
0D 0A 64 65 6C 20 25 30 0D 0A 00 01   y..)..del %0....
01 00 00 00 50 70 12 01 04 00 00 00   úMÎw....Pp......
8A 2A 51 8D 48 70 12 01 00 00 12 01   zPÎw.*Q.Hp......
00 00 00 00 00 00 00 00 30 AF D1 77   ............0¯Ñw
FE FF FF FF 80 ED 0F 01 7C 6E CE 77   z..ûþÿÿÿ.í..|nÎw
20 00 00 00 00 00 00 00 18 00 07 1F   ................
00 00 12 01 01 00 00 00 D7 00 00 00   c..P........x...
BF 86 14 01 7F 00 00 00 70 00 00 00   ....¿.......p...
```

Figure 26- Creating the file with the .bat extension in the Debugger

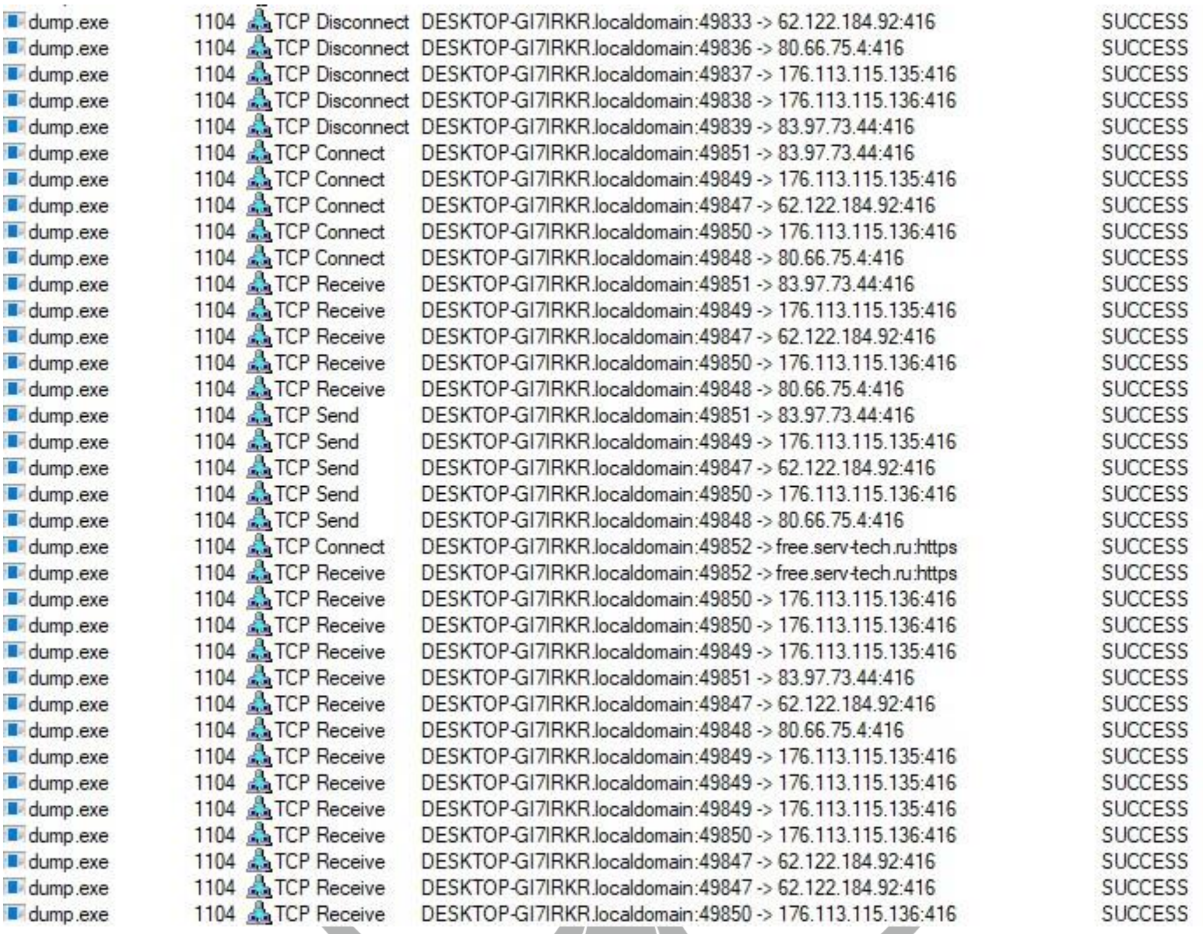
In the continuation of the dynamic analysis, it is determined that it communicates **with dns addresses named** "vanheim.cn**", "jotunheim.name" and "**free.serv-tech.ru".



```
000B8851   59         pop ecx                              ecx:"free.serv-tech.ru"
000B8852   59         pop ecx                              ecx:"free.serv-tech.ru"
000B8853   85C0       test eax,eax
000B8855   74 15      je dump.B886C
000B256D   4F         dec edi
000B256E   75 EA      jne dump.B255A
000B2570   5E         pop esi                              esi:"jotunheim.name"
000B2571   8B45 08    mov eax,dword ptr ss:[ebp+8]         [ebp+8]:"jotunheim.name"
000B2574   5F         pop edi
000B2575   5D         pop ebp
000B2576   C3         ret
000B2577   8B41 0C    mov eax,dword ptr ds:[ecx+C]         eax:"jotunheim.name"
000B257A   8D50 FF    lea edx,dword ptr ds:[eax-1]
```

Figure 27 - Communicating DNS addresses

When the other connections made by the malware are examined with the ProcMon tool, it is understood that it connects with random ip addresses.



```
dump.exe    1104  TCP Disconnect  DESKTOP-GI7IRKR.localdomain:49833 -> 62.122.184.92:416        SUCCESS
dump.exe    1104  TCP Disconnect  DESKTOP-GI7IRKR.localdomain:49836 -> 80.66.75.4:416           SUCCESS
dump.exe    1104  TCP Disconnect  DESKTOP-GI7IRKR.localdomain:49837 -> 176.113.115.135:416      SUCCESS
dump.exe    1104  TCP Disconnect  DESKTOP-GI7IRKR.localdomain:49838 -> 176.113.115.136:416      SUCCESS
dump.exe    1104  TCP Disconnect  DESKTOP-GI7IRKR.localdomain:49839 -> 83.97.73.44:416          SUCCESS
dump.exe    1104  TCP Connect     DESKTOP-GI7IRKR.localdomain:49851 -> 83.97.73.44:416          SUCCESS
dump.exe    1104  TCP Connect     DESKTOP-GI7IRKR.localdomain:49849 -> 176.113.115.135:416      SUCCESS
dump.exe    1104  TCP Connect     DESKTOP-GI7IRKR.localdomain:49847 -> 62.122.184.92:416        SUCCESS
dump.exe    1104  TCP Connect     DESKTOP-GI7IRKR.localdomain:49850 -> 176.113.115.136:416      SUCCESS
dump.exe    1104  TCP Connect     DESKTOP-GI7IRKR.localdomain:49848 -> 80.66.75.4:416           SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49851 -> 83.97.73.44:416          SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49849 -> 176.113.115.135:416      SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49847 -> 62.122.184.92:416        SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49850 -> 176.113.115.136:416      SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49848 -> 80.66.75.4:416           SUCCESS
dump.exe    1104  TCP Send        DESKTOP-GI7IRKR.localdomain:49851 -> 83.97.73.44:416          SUCCESS
dump.exe    1104  TCP Send        DESKTOP-GI7IRKR.localdomain:49849 -> 176.113.115.135:416      SUCCESS
dump.exe    1104  TCP Send        DESKTOP-GI7IRKR.localdomain:49847 -> 62.122.184.92:416        SUCCESS
dump.exe    1104  TCP Send        DESKTOP-GI7IRKR.localdomain:49850 -> 176.113.115.136:416      SUCCESS
dump.exe    1104  TCP Send        DESKTOP-GI7IRKR.localdomain:49848 -> 80.66.75.4:416           SUCCESS
dump.exe    1104  TCP Connect     DESKTOP-GI7IRKR.localdomain:49852 -> free.serv-tech.ru:https  SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49852 -> free.serv-tech.ru:https  SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49850 -> 176.113.115.136:416      SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49850 -> 176.113.115.136:416      SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49849 -> 176.113.115.135:416      SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49851 -> 83.97.73.44:416          SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49847 -> 62.122.184.92:416        SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49848 -> 80.66.75.4:416           SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49849 -> 176.113.115.135:416      SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49849 -> 176.113.115.135:416      SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49849 -> 176.113.115.135:416      SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49850 -> 176.113.115.136:416      SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49847 -> 62.122.184.92:416        SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49847 -> 62.122.184.92:416        SUCCESS
dump.exe    1104  TCP Receive     DESKTOP-GI7IRKR.localdomain:49850 -> 176.113.115.136:416      SUCCESS
```

Figure 28 - Displaying the addresses connected with the Procmon tool

YARA Rule

```
import "hash"
rule golfstikator
{
    meta:
            author = "Tolga Yılmaz"
    strings:
           $a1 = "C:\\jupivulehu.pdb"
           $a2 = "sageyi"
           $a3 = "puduvikajicezodezofut"
           $b = { 6C 65 64 69 6C 6F 70 6F 72 65 6A 65 66 6F 67 20 67 75 68 65 77 61 74 61 7A 69 6B 69 73 61 6E 69 76 69 68 6F }

    condition:
           hash.md5(0, filesize) == "743fc0d22063e7ea97ca753280ff9f3e" or
           2 ($a*) or $b
}
```

```
import "hash"

rule tofsee

{

    meta:

            author = "Tolga Yılmaz"

    strings:

           $a1 = "\\\\.\\pipe\\"

           $a2 = "smtp_herr"

           $a3 = "lid_file_upd"

           $a4 = "loader_id"

           $a5 = "12:08:32"

           $b = { 77 74 6D 5F }



    condition:

           hash.md5(0, filesize) == "a050f3c88055b70ddf52d04747d4f527" or

           3 of ($a*) and $b

}
```

MITRE ATTACK TABLE

Execution	Persistence	Privilege Escalation	Defense Evasion	Discovery	C&C
Command and Scripting Interpreter (T1059)	**Windows Service** (T1543.003)	**Windows Service** (T1543.003)	**Software Packing** (T1027.002)	**Security Software Discovery** (T1518.001)	**Application Layer Protocols** (T1071)
Native API (T1106)	**Registry Run Keys / Startup Folder** (T1547.001)	**Process Injection** (T1055)	**File Deletion** (T1070.0040)	**System Time Discovery** (T1124)	**Non-Application Layer Protocol** (T1095)
	Create or Modify System Process (T1543)		**Virtualization/Sandbox Evasion** (T1497)	**File and Directory Discovery** (T1083)	
			Process Injection (T1055)	**System Owner/User Discovery** (T1033)	
				Modify Registry (T1112)	

Solution Suggestions

1. System security should be increased by using up-to-date antivirus software.
2. By regularly updating your security software and operating system, its defenses against known attacks should be strengthened.
3. To avoid being exposed to malicious websites and downloads, one should use trusted websites and downloads should be made from reliable sources.
4. By backing up your important data, the risk of data loss that can be caused by malware should be reduced.
5. Second-layer security measures such as two-factor authentication (2FA) or multi-factor authentication (MFA) should be enabled for your accounts.

Prepared By

Tolga Yılmaz

https://www.linkedin.com/in/tolga-ylmz/