Tofsee

TECHNICAL ANALYSIS REPORT

ZAYOTEM

ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

Contents

Preview

Tofsee is a malware family used as a botnet. This malware family can be used for different purposes such as sending spam emails, phishing attacks, downloading malware and forcing victims' computers to join other botnets.

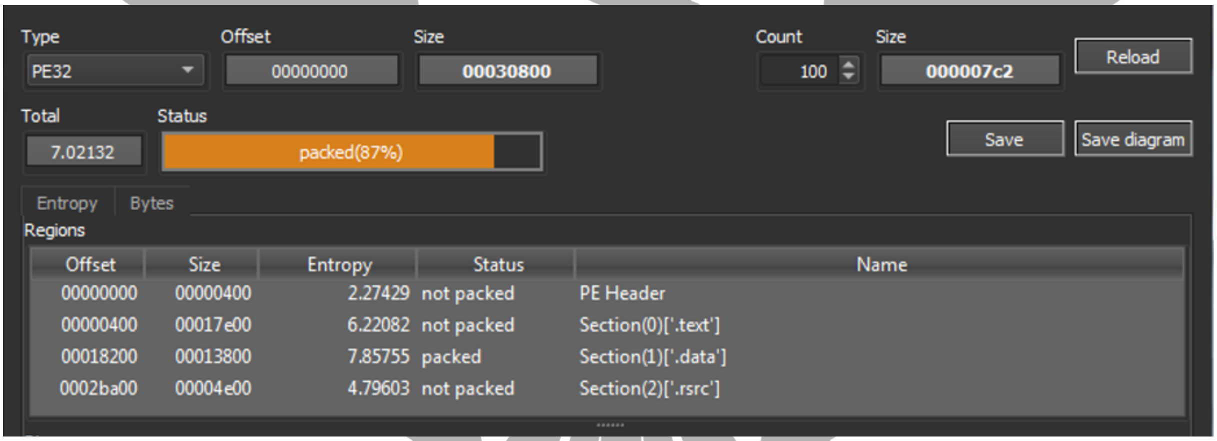
Tofsee was launched in 2013 and has been constantly updated and improved since then. It is used extensively in countries such as Russia and Ukraine.

The Tofsee family of malware is usually transmitted through spam email attachments or malicious links. Once placed on the victim's computer, it can perform many different actions such as downloading other malware and connecting to other botnets.

nightskywalker.exe Analysis

Adı	nightskywalker.exe
MD5	e5d88e4a2497a5f8219482d64d3b501b
SHA256	e16191d95969d7ae164c1dd4f5b0ac87a49a617e902743d204ffcc2ebc2fdf49
Dosya Türü	PE32 / EXE

Static Analysis



Type: PE32
Offset: 00000000
Size: 00030800
Count: 100
Size: 000007c2
Total: 7.02132
Status: packed(87%)

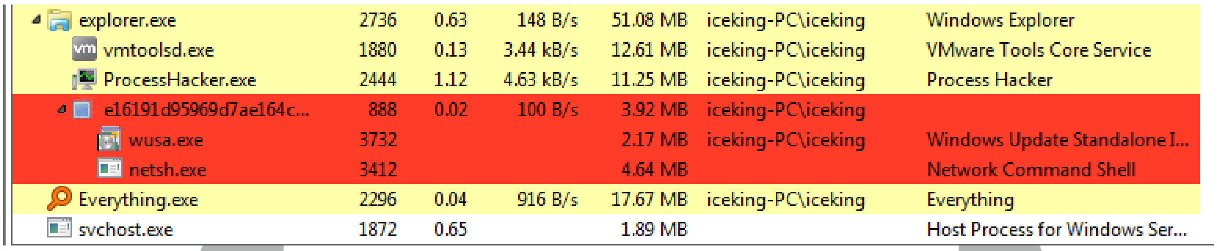
Regions

Offset	Size	Entropy	Status	Name
00000000	00000400	2.27429	not packed	PE Header
00000400	00017e00	6.22082	not packed	Section(0)['.text']
00018200	00013800	7.85755	packed	Section(1)['.data']
0002ba00	00004e00	4.79603	not packed	Section(2)['.rsrc']

Figure 1- Observing the packaging process in the malicious file

The malware appears to be packaged at first glance.

Dynamic Analysis



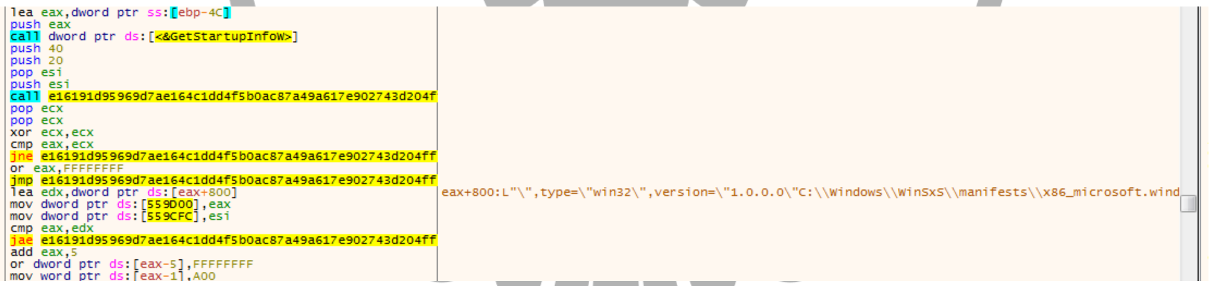
explorer.exe	2736	0.63	148 B/s	51.08 MB	iceking-PC\iceking	Windows Explorer
vmtoolsd.exe	1880	0.13	3.44 kB/s	12.61 MB	iceking-PC\iceking	VMware Tools Core Service
ProcessHacker.exe	2444	1.12	4.63 kB/s	11.25 MB	iceking-PC\iceking	Process Hacker
e16191d95969d7ae164c...	888	0.02	100 B/s	3.92 MB	iceking-PC\iceking	
wusa.exe	3732			2.17 MB	iceking-PC\iceking	Windows Update Standalone I...
netsh.exe	3412			4.64 MB		Network Command Shell
Everything.exe	2296	0.04	916 B/s	17.67 MB	iceking-PC\iceking	Everything
svchost.exe	1872	0.65		1.89 MB		Host Process for Windows Ser...

Figure 2- Examination of the malware with Process Monitor

In the first place, it is seen that the malware runs two child processes.

User Account Control (UAC) is used in the Windows operating system to prevent unauthorized changes to the computer. “wusa.exe” (Windows Update Standalone Installer) is one of the processes with autoElevate attribute as true in its manifest. It has the right to run itself as administrator without having UAC permission. By abusing this authority, it is possible to inject the malware into wusa.exe and run administrative privileges.

“netsh.exe” is a program for modifying or viewing the computer's network configuration. It seems that the malware may have changed the network settings.

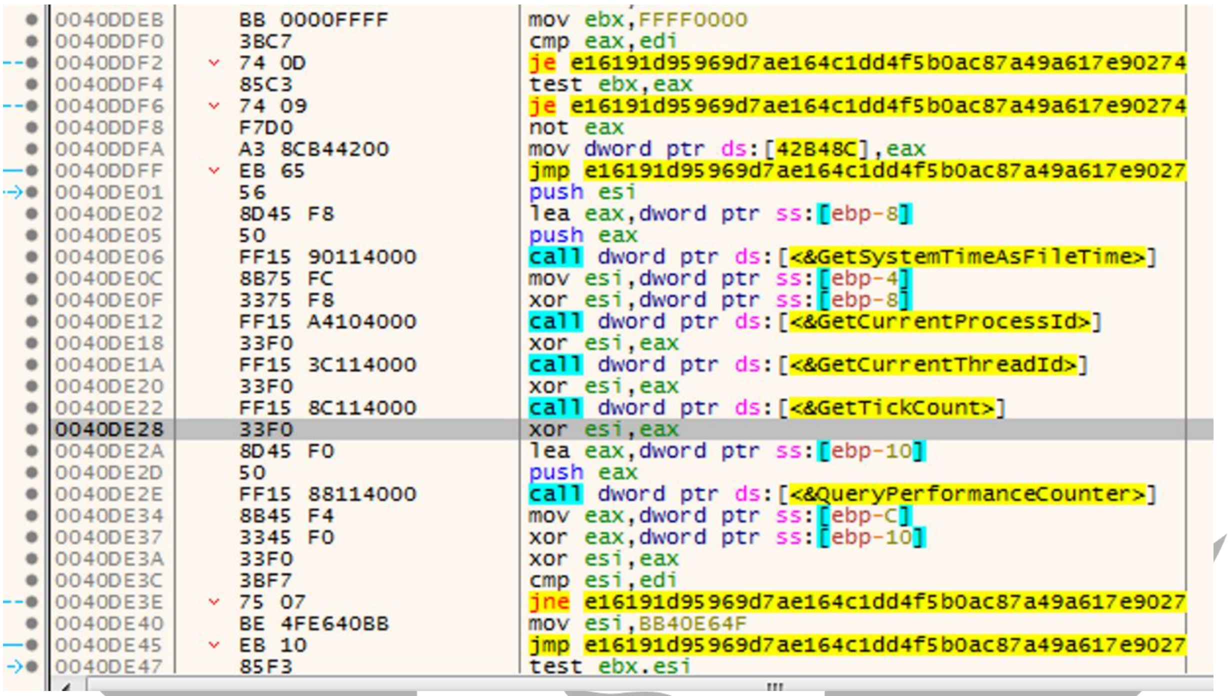


```
lea eax,dword ptr ss:[ebp-4C]
push eax
call dword ptr ds:[<&GetStartupInfoW>]
push 40
push 20
pop esi
push esi
call e16191d95969d7ae164c1dd4f5b0ac87a49a617e902743d204f
pop ecx
pop ecx
xor ecx,ecx
cmp eax,ecx
jne e16191d95969d7ae164c1dd4f5b0ac87a49a617e902743d204ff
or eax,FFFFFFFF
jmp e16191d95969d7ae164c1dd4f5b0ac87a49a617e902743d204ff
lea edx,dword ptr ds:[eax+800]                            eax+800:L"\",type=\"win32\",version=\"1.0.0.0\"C:\\Windows\\WinSxS\\manifests\\x86_microsoft.wind
mov dword ptr ds:[559D00],eax
mov dword ptr ds:[559CFC],esi
cmp eax,edx
jae e16191d95969d7ae164c1dd4f5b0ac87a49a617e902743d204ff
add eax,5
or dword ptr ds:[eax-5],FFFFFFFF
mov word ptr ds:[eax-1],A00
```

Figure 3- Information is collected using the GetStartupInfoW API

```
type=\"win32\",version=\"1.0.0.0\"C:\\Windows\\WinSxS\\manifests\\x86_microsoft.windows.isolationautomation_6595b64144ccf1df_1.0.0.0_none_35d357a66c38ade4.manifest
```

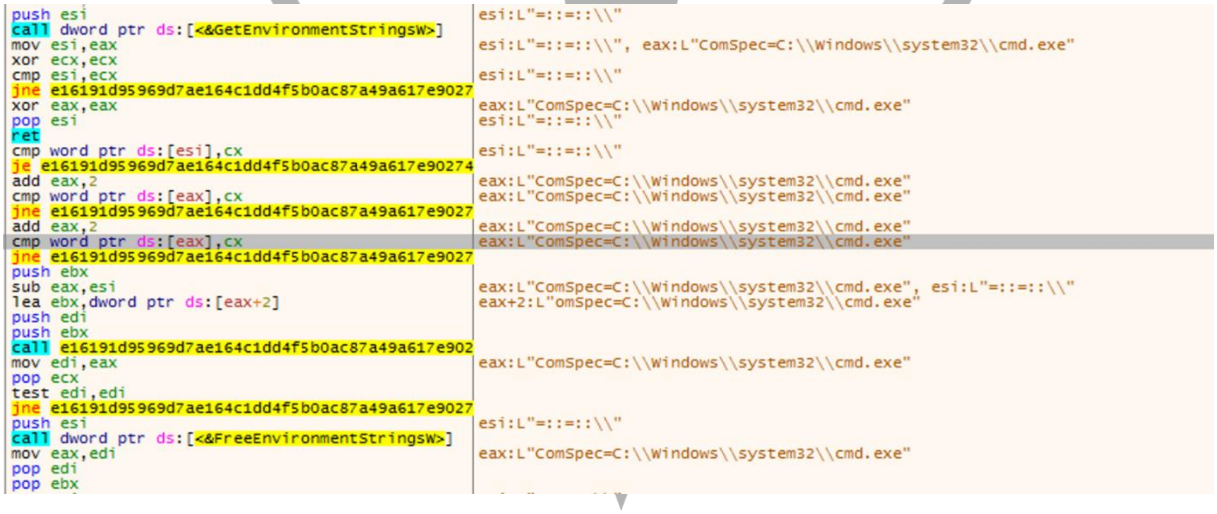
The malware collects system-related information with the API shown in Figure 3.



```
0040DDEB    BB 0000FFFF      mov ebx,FFFF0000
0040DDF0    3BC7             cmp eax,edi
0040DDF2    74 0D            je e16191d95969d7ae164c1dd4f5b0ac87a49a617e90274
0040DDF4    85C3             test ebx,eax
0040DDF6    74 09            je e16191d95969d7ae164c1dd4f5b0ac87a49a617e90274
0040DDF8    F7D0             not eax
0040DDFA    A3 8CB44200      mov dword ptr ds:[42B48C],eax
0040DDFF    EB 65            jmp e16191d95969d7ae164c1dd4f5b0ac87a49a617e9027
0040DE01    56               push esi
0040DE02    8D45 F8          lea eax,dword ptr ss:[ebp-8]
0040DE05    50               push eax
0040DE06    FF15 90114000    call dword ptr ds:[<&GetSystemTimeAsFileTime>]
0040DE0C    8B75 FC          mov esi,dword ptr ss:[ebp-4]
0040DE0F    3375 F8          xor esi,dword ptr ss:[ebp-8]
0040DE12    FF15 A4104000    call dword ptr ds:[<&GetCurrentProcessId>]
0040DE18    33F0             xor esi,eax
0040DE1A    FF15 3C114000    call dword ptr ds:[<&GetCurrentThreadId>]
0040DE20    33F0             xor esi,eax
0040DE22    FF15 8C114000    call dword ptr ds:[<&GetTickCount>]
0040DE28    33F0             xor esi,eax
0040DE2A    8D45 F0          lea eax,dword ptr ss:[ebp-10]
0040DE2D    50               push eax
0040DE2E    FF15 88114000    call dword ptr ds:[<&QueryPerformanceCounter>]
0040DE34    8B45 F4          mov eax,dword ptr ss:[ebp-C]
0040DE37    3345 F0          xor eax,dword ptr ss:[ebp-10]
0040DE3A    33F0             xor esi,eax
0040DE3C    3BF7             cmp esi,edi
0040DE3E    75 07            jne e16191d95969d7ae164c1dd4f5b0ac87a49a617e9027
0040DE40    BE 4FE640BB      mov esi,BB40E64F
0040DE45    EB 10            jmp e16191d95969d7ae164c1dd4f5b0ac87a49a617e9027
0040DE47    85F3             test ebx,esi
```

Figure 4- Collects system time information

The malware obtains system time information using APIs such as GetSystemTimeAsFileTime, GetCurrentProcessId, GetCurrentThreadId. It gets the current Process and Thread Ids.



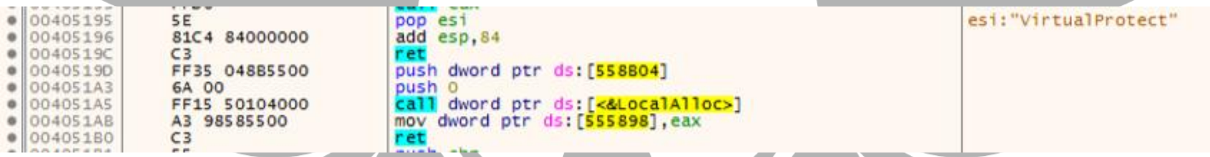
```
push esi                                                 esi:L"=::=::\\"
call dword ptr ds:[<&GetEnvironmentStringsW>]
mov esi,eax                                              esi:L"=::=::\\", eax:L"ComSpec=C:\\Windows\\system32\\cmd.exe"
xor ecx,ecx
cmp esi,ecx                                              esi:L"=::=::\\"
jne e16191d95969d7ae164c1dd4f5b0ac87a49a617e9027
xor eax,eax                                              eax:L"ComSpec=C:\\Windows\\system32\\cmd.exe"
pop esi                                                  esi:L"=::=::\\"
ret
cmp word ptr ds:[esi],cx                                 esi:L"=::=::\\"
je e16191d95969d7ae164c1dd4f5b0ac87a49a617e90274
add eax,2                                                eax:L"ComSpec=C:\\Windows\\system32\\cmd.exe"
cmp word ptr ds:[eax],cx                                 eax:L"ComSpec=C:\\Windows\\system32\\cmd.exe"
jne e16191d95969d7ae164c1dd4f5b0ac87a49a617e9027
add eax,2                                                eax:L"ComSpec=C:\\Windows\\system32\\cmd.exe"
cmp word ptr ds:[eax],cx                                 eax:L"ComSpec=C:\\Windows\\system32\\cmd.exe"
jne e16191d95969d7ae164c1dd4f5b0ac87a49a617e9027
push ebx
sub eax,esi                                              eax:L"ComSpec=C:\\Windows\\system32\\cmd.exe", esi:L"=::=::\\"
lea ebx,dword ptr ds:[eax+2]                             eax+2:L"omSpec=C:\\Windows\\system32\\cmd.exe"
push edi
push ebx
call e16191d95969d7ae164c1dd4f5b0ac87a49a617e902
mov edi,eax                                              eax:L"ComSpec=C:\\Windows\\system32\\cmd.exe"
pop ecx
test edi,edi
jne e16191d95969d7ae164c1dd4f5b0ac87a49a617e9027
push esi                                                 esi:L"=::=::\\"
call dword ptr ds:[<&FreeEnvironmentStringsW>]
mov eax,edi                                              eax:L"ComSpec=C:\\Windows\\system32\\cmd.exe"
pop edi
pop ebx
```

Figure 5- Collects system information

It is seen that the GetEnvironmentStringsW API obtains environment variable information for the current process. This information consists of sensitive information that includes information about the user, hardware and environment.Zararlının topladığı bazı bilgiler aşağıdaki tabloda verilmiştir.

ALLUSERSPROFILE=C:\\ProgramData	NUMBER_OF_PROCESSORS=4
LOGONSERVER=\\\\ICEKING-PC	LOCALAPPDATA=C:\\Users\\user\\AppData\\Local
FP_NO_HOST_CHECK=NO	COMPUTERNAME=ICEKING-PC
OS=Windows_NT	PROCESSOR_ARCHITECTURE=x86
HOMEDRIVE=C:	ComSpec=C:\\Windows\\system32\\cmd.exe
TEMP=C:\\Users\\user\\AppData\\Local\\Temp	PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC

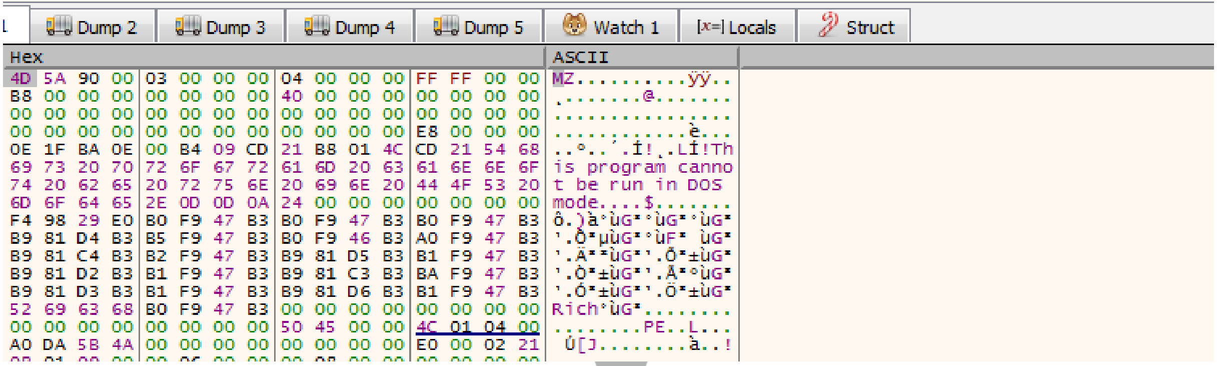
Figure 6- Information obtained by the malware



```
00405195   5E                 pop esi                              esi:"VirtualProtect"
00405196   81C4 84000000      add esp,84
0040519C   C3                 ret
0040519D   FF35 048B5500      push dword ptr ds:[558B04]
004051A3   6A 00              push 0
004051A5   FF15 50104000      call dword ptr ds:[<&LocalAlloc>]
004051AB   A3 98585500        mov dword ptr ds:[555898],eax
004051B0   C3                 ret
```

Figure 7- LocalAlloc appears to be used

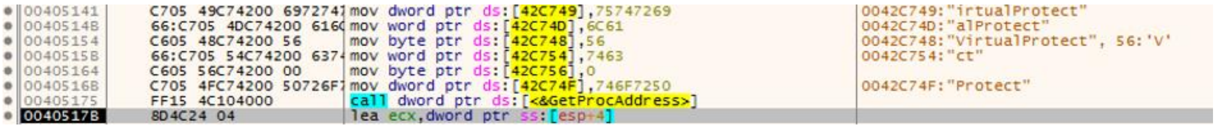
The malware allocates space using the **LocalAlloc** API.



```
Hex                                                ASCII
4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00   MZ..........ÿÿ..
B8 00 00 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 00 00 00 00 00 00 E8 00 00 00   ............è...
0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68   ..º..´.Í!¸.LÍ!Th
69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F   is program canno
74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20   t be run in DOS 
6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00   mode....$.......
F4 98 29 E0 B0 F9 47 B3 B0 F9 47 B3 B0 F9 47 B3   ô.)à°ùG³°ùG³°ùG³
B9 81 D4 B3 B5 F9 47 B3 B0 F9 46 B3 A0 F9 47 B3   ¹.Ô³µùG³°ùF³ ùG³
B9 81 C4 B3 B2 F9 47 B3 B9 81 D5 B3 B1 F9 47 B3   ¹.Ä³²ùG³¹.Õ³±ùG³
B9 81 D2 B3 B1 F9 47 B3 B9 81 C3 B3 BA F9 47 B3   ¹.Ò³±ùG³¹.Ã³ºùG³
B9 81 D3 B3 B1 F9 47 B3 B9 81 D6 B3 B1 F9 47 B3   ¹.Ó³±ùG³¹.Ö³±ùG³
52 69 63 68 B0 F9 47 B3 00 00 00 00 00 00 00 00   Rich°ùG³........
00 00 00 00 00 00 00 00 50 45 00 00 4C 01 04 00   ........PE..L...
A0 DA 5B 4A 00 00 00 00 00 00 00 00 E0 00 02 21    Ú[J........à..!
```

Figure 7- Hex code of the extracted file

The malware writes the packaged file in this reserved area.



```
00405141   C705 49C74200 6972747 mov dword ptr ds:[42C749],75747269     0042C749:"irtualProtect"
0040514B   66:C705 4DC74200 616C mov word ptr ds:[42C74D],6C61          0042C74D:"alProtect"
00405154   C605 48C74200 56      mov byte ptr ds:[42C748],56            0042C748:"VirtualProtect", 56:'V'
0040515B   66:C705 54C74200 6374 mov word ptr ds:[42C754],7463          0042C754:"ct"
00405164   C605 56C74200 00      mov byte ptr ds:[42C756],0
0040516B   C705 4FC74200 50726F7 mov dword ptr ds:[42C74F],746F7250     0042C74F:"Protect"
00405175   FF15 4C104000         call dword ptr ds:[<&GetProcAddress>]
0040517B   8D4C24 04             lea ecx,dword ptr ss:[esp+4]
```

Figure 9- API resolving is done using GetProcAddress.

The malware is performing API Resolving. Rather than importing all the required APIs, it just hides its name. It dynamically resolves APIs with GetProcAddress at runtime. It makes analysis difficult in this way.

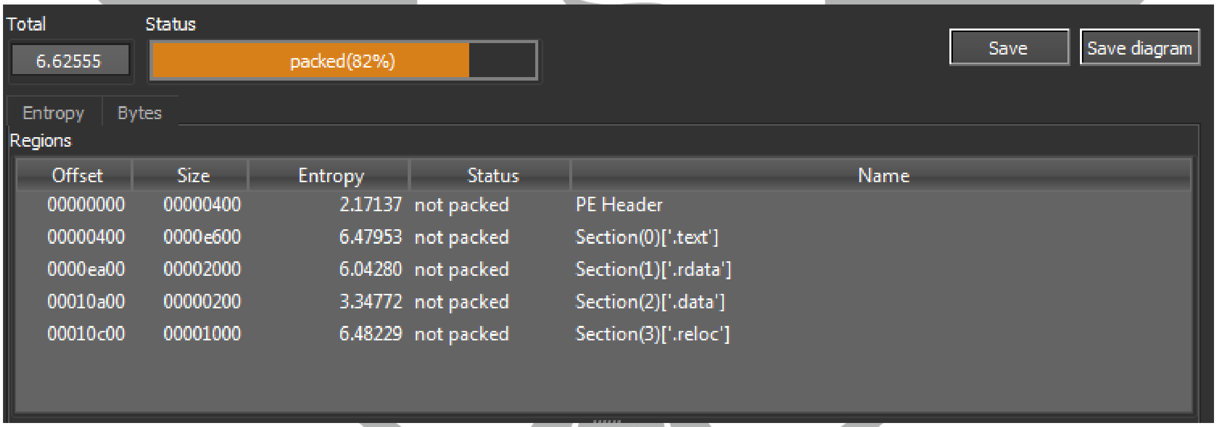
Resolving is being done to use the VirtualProtect API. After the DLL is given as a parameter to GetProcAddress, the address of the exported VirtualProtect API is returned. In this way, the resolving process is completed.

With the VirtualProtect API, it gives execution permission to the file in this area, which is allocated space. Then the file in the allocated space is run.

Stage 2 Analysis

Adı	-
MD5	92E466525E810B79AE23EAC344A52027
SHA256	96baba74a907890b995f23c7db21568f7bfb5dbf417ed90ca311482b99702b72
Dosya Türü	PE32 / EXE

Static Analysis



Total: 6.62555

Status: packed(82%)

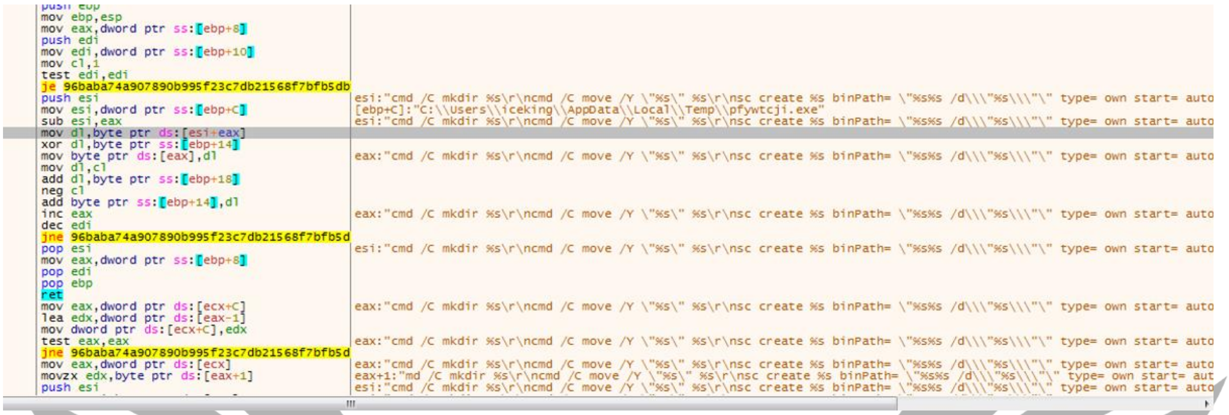
Regions

Offset	Size	Entropy	Status	Name
00000000	00000400	2.17137	not packed	PE Header
00000400	0000e600	6.47953	not packed	Section(0)['.text']
0000ea00	00002000	6.04280	not packed	Section(1)['.rdata']
00010a00	00000200	3.34772	not packed	Section(2)['.data']
00010c00	00001000	6.48229	not packed	Section(3)['.reloc']

Figure 10- There is no packaging process in the extracted file.

The file extracted from Malware is not packaged.

Dynamic Analysis

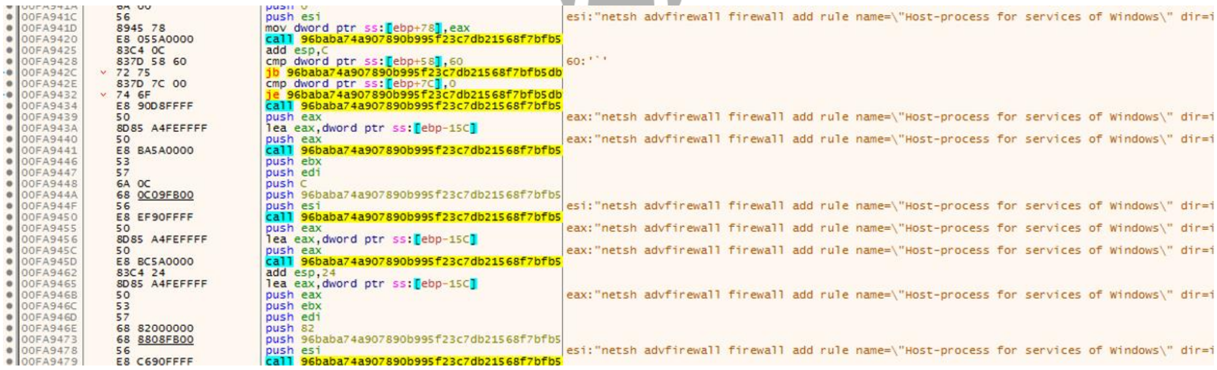


```
push ebp
mov ebp,esp
mov eax,dword ptr ss:[ebp+8]
push edi
mov edi,dword ptr ss:[ebp+10]
mov cl,1
test edi,edi
je 96baba74a907890b995f23c7db21568f7bfb5db
push esi                                    esi:"cmd /C mkdir %s\r\ncmd /C move /Y \"%s\" %s\r\nsc create %s binPath= \"%s%s /d\\\"%s\\\"\" type= own start= auto
mov esi,dword ptr ss:[ebp+C]                [ebp+C]:"C:\\Users\\iceking\\AppData\\Local\\Temp\\pfywtcji.exe"
sub esi,eax                                 esi:"cmd /C mkdir %s\r\ncmd /C move /Y \"%s\" %s\r\nsc create %s binPath= \"%s%s /d\\\"%s\\\"\" type= own start= auto
mov dl,byte ptr ds:[esi+eax]
xor dl,byte ptr ss:[ebp+14]
mov byte ptr ds:[eax],dl                    eax:"cmd /C mkdir %s\r\ncmd /C move /Y \"%s\" %s\r\nsc create %s binPath= \"%s%s /d\\\"%s\\\"\" type= own start= auto
mov dl,cl
add dl,byte ptr ss:[ebp+18]
neg cl
add byte ptr ss:[ebp+14],dl
inc eax                                     eax:"cmd /C mkdir %s\r\ncmd /C move /Y \"%s\" %s\r\nsc create %s binPath= \"%s%s /d\\\"%s\\\"\" type= own start= auto
dec edi
jne 96baba74a907890b995f23c7db21568f7bfb5d
pop esi                                     esi:"cmd /C mkdir %s\r\ncmd /C move /Y \"%s\" %s\r\nsc create %s binPath= \"%s%s /d\\\"%s\\\"\" type= own start= auto
mov eax,dword ptr ss:[ebp+8]
pop edi
pop ebp
ret
mov eax,dword ptr ds:[ecx+C]                eax:"cmd /C mkdir %s\r\ncmd /C move /Y \"%s\" %s\r\nsc create %s binPath= \"%s%s /d\\\"%s\\\"\" type= own start= auto
lea edx,dword ptr ds:[eax-1]
mov dword ptr ds:[ecx+C],edx
test eax,eax                                eax:"cmd /C mkdir %s\r\ncmd /C move /Y \"%s\" %s\r\nsc create %s binPath= \"%s%s /d\\\"%s\\\"\" type= own start= auto
jne 96baba74a907890b995f23c7db21568f7bfb5d
mov eax,dword ptr ds:[ecx]                  eax:"cmd /C mkdir %s\r\ncmd /C move /Y \"%s\" %s\r\nsc create %s binPath= \"%s%s /d\\\"%s\\\"\" type= own start= auto
movzx edx,byte ptr ds:[eax+1]               eax+1:"md /C mkdir %s\r\ncmd /C move /Y \"%s\" %s\r\nsc create %s binPath= \"%s%s /d\\\"%s\\\"\" type= own start= aut
push esi                                    esi:"cmd /C mkdir %s\r\ncmd /C move /Y \"%s\" %s\r\nsc create %s binPath= \"%s%s /d\\\"%s\\\"\" type= own start= auto
```

Figure 11- CMD commands appear

The malware keeps CMD commands in a String. This String it holds is run separately as a result of some manipulations.

```
cmd /C mkdir %s\r\n

cmd /C move /Y \"%s\" %s\r\n

sc create %s binPath= \"%s%s /d\\\"%s\\\"\" type= own start= auto DisplayName= \"wifi support\"\r\n

sc description %s \"wifi internet conection\"\r\n

sc start %s\r\n”
```

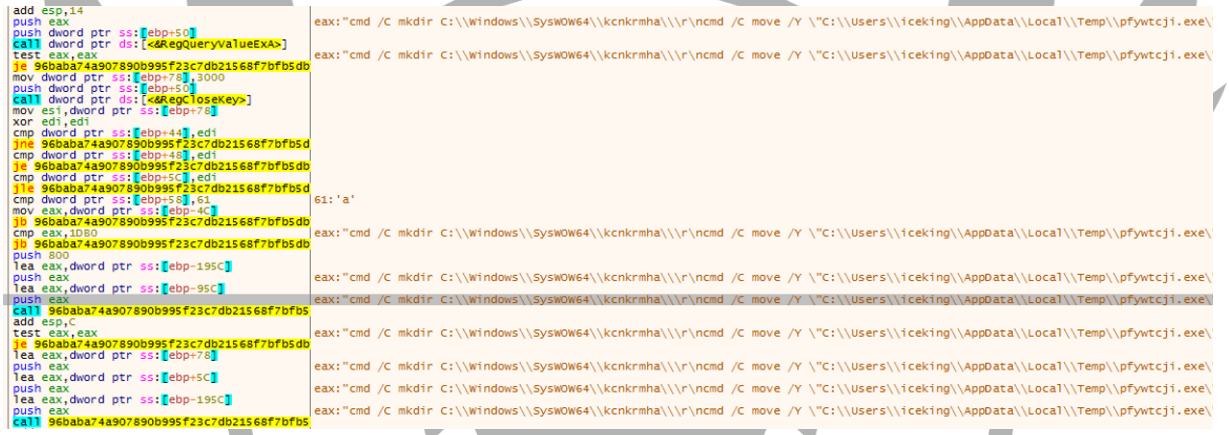


```
00FA941A   6A 00          push 0
00FA941C   56             push esi                                       esi:"netsh advfirewall firewall add rule name=\"Host-process for services of Windows\" dir=i
00FA941D   8945 78        mov dword ptr ss:[ebp+78],eax
00FA9420   E8 055A0000    call 96baba74a907890b995f23c7db21568f7bfb5
00FA9425   83C4 0C        add esp,C
00FA9428   837D 58 60     cmp dword ptr ss:[ebp+58],60                   60:'`'
00FA942C   72 75          jb 96baba74a907890b995f23c7db21568f7bfb5db
00FA942E   837D 7C 00     cmp dword ptr ss:[ebp+7C],0
00FA9432   74 6F          je 96baba74a907890b995f23c7db21568f7bfb5db
00FA9434   E8 90D8FFFF    call 96baba74a907890b995f23c7db21568f7bfb5
00FA9439   50             push eax                                       eax:"netsh advfirewall firewall add rule name=\"Host-process for services of Windows\" dir=i
00FA943A   8D85 A4FEFFFF  lea eax,dword ptr ss:[ebp-15C]
00FA9440   50             push eax                                       eax:"netsh advfirewall firewall add rule name=\"Host-process for services of Windows\" dir=i
00FA9441   E8 BA5A0000    call 96baba74a907890b995f23c7db21568f7bfb5
00FA9446   53             push ebx
00FA9447   57             push edi
00FA9448   6A 0C          push C
00FA944A   68 0C09FB00    push 96baba74a907890b995f23c7db21568f7bfb5
00FA944F   56             push esi                                       esi:"netsh advfirewall firewall add rule name=\"Host-process for services of Windows\" dir=i
00FA9450   E8 EF90FFFF    call 96baba74a907890b995f23c7db21568f7bfb5
00FA9455   50             push eax                                       eax:"netsh advfirewall firewall add rule name=\"Host-process for services of Windows\" dir=i
00FA9456   8D85 A4FEFFFF  lea eax,dword ptr ss:[ebp-15C]
00FA945C   50             push eax                                       eax:"netsh advfirewall firewall add rule name=\"Host-process for services of Windows\" dir=i
00FA945D   E8 BC5A0000    call 96baba74a907890b995f23c7db21568f7bfb5
00FA9462   83C4 24        add esp,24
00FA9465   8D85 A4FEFFFF  lea eax,dword ptr ss:[ebp-15C]
00FA946B   50             push eax                                       eax:"netsh advfirewall firewall add rule name=\"Host-process for services of Windows\" dir=i
00FA946C   53             push ebx
00FA946D   57             push edi
00FA946E   68 82000000    push 82
00FA9473   68 8808FB00    push 96baba74a907890b995f23c7db21568f7bfb5
00FA9478   56             push esi                                       esi:"netsh advfirewall firewall add rule name=\"Host-process for services of Windows\" dir=i
00FA9479   E8 C690FFFF    call 96baba74a907890b995f23c7db21568f7bfb5
```

Figure 12- Adding rule to Windows Firewall

```
netsh advfirewall firewall add rule name=\"Host-process for services of Windows\" dir=in
action=allow program=\"%s\" enable=yes>nul\r\n
```

The malware adds a rule to the Windows Firewall and allows its own traffic. The rule created with the name "host-process for services of Windows" allows inbound traffic. The program to which the rule will be applied is then determined as **“C:\\Users\\user\\AppData\\Local\\Temp\\pfywtcji.exe\”**.

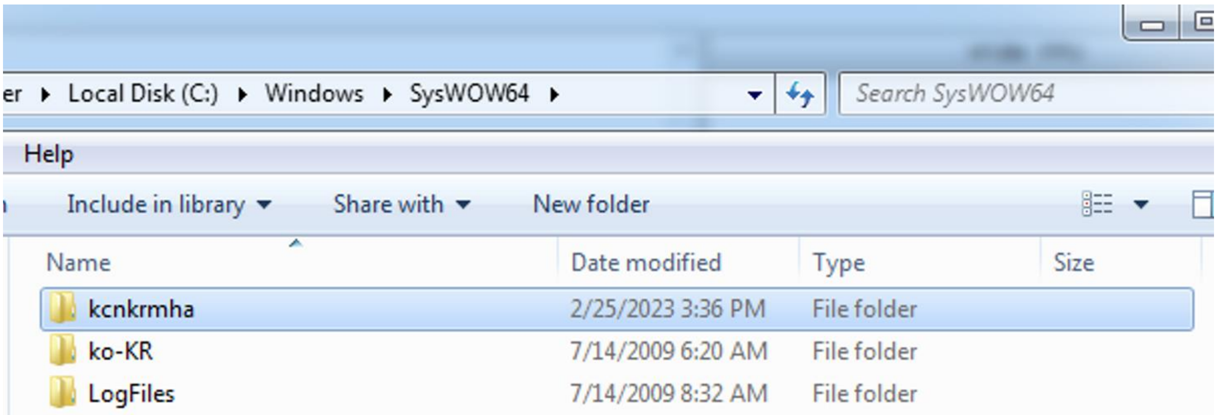


```
add esp,14
push eax                                              eax:"cmd /C mkdir C:\\Windows\\SysWOW64\\kcnkrmha\\\r\ncmd /C move /Y \"C:\\Users\\iceking\\AppData\\Local\\Temp\\pfywtcji.exe\
push dword ptr ss:[ebp+50]
call dword ptr ds:[<&RegQueryValueExA>]
test eax,eax                                          eax:"cmd /C mkdir C:\\Windows\\SysWOW64\\kcnkrmha\\\r\ncmd /C move /Y \"C:\\Users\\iceking\\AppData\\Local\\Temp\\pfywtcji.exe\
je 96baba74a907890b995f23c7db21568f7bfb5db
mov dword ptr ss:[ebp+78],3000
push dword ptr ss:[ebp+50]
call dword ptr ds:[<&RegCloseKey>]
mov esi,dword ptr ss:[ebp+78]
xor edi,edi
cmp dword ptr ss:[ebp+44],edi
jne 96baba74a907890b995f23c7db21568f7bfb5d
cmp dword ptr ss:[ebp+48],edi
je 96baba74a907890b995f23c7db21568f7bfb5db
cmp dword ptr ss:[ebp+5C],edi
jle 96baba74a907890b995f23c7db21568f7bfb5d
cmp dword ptr ss:[ebp+58],61                          61:'a'
mov eax,dword ptr ss:[ebp-4C]
jb 96baba74a907890b995f23c7db21568f7bfb5db
cmp eax,1DB0                                          eax:"cmd /C mkdir C:\\Windows\\SysWOW64\\kcnkrmha\\\r\ncmd /C move /Y \"C:\\Users\\iceking\\AppData\\Local\\Temp\\pfywtcji.exe\
jb 96baba74a907890b995f23c7db21568f7bfb5db
push 800
lea eax,dword ptr ss:[ebp-195C]
push eax                                              eax:"cmd /C mkdir C:\\Windows\\SysWOW64\\kcnkrmha\\\r\ncmd /C move /Y \"C:\\Users\\iceking\\AppData\\Local\\Temp\\pfywtcji.exe\
lea eax,dword ptr ss:[ebp-95C]
push eax                                              eax:"cmd /C mkdir C:\\Windows\\SysWOW64\\kcnkrmha\\\r\ncmd /C move /Y \"C:\\Users\\iceking\\AppData\\Local\\Temp\\pfywtcji.exe\
call 96baba74a907890b995f23c7db21568f7bfb5
add esp,C
test eax,eax                                          eax:"cmd /C mkdir C:\\Windows\\SysWOW64\\kcnkrmha\\\r\ncmd /C move /Y \"C:\\Users\\iceking\\AppData\\Local\\Temp\\pfywtcji.exe\
je 96baba74a907890b995f23c7db21568f7bfb5db
lea eax,dword ptr ss:[ebp+78]
push eax                                              eax:"cmd /C mkdir C:\\Windows\\SysWOW64\\kcnkrmha\\\r\ncmd /C move /Y \"C:\\Users\\iceking\\AppData\\Local\\Temp\\pfywtcji.exe\
lea eax,dword ptr ss:[ebp+5C]
push eax                                              eax:"cmd /C mkdir C:\\Windows\\SysWOW64\\kcnkrmha\\\r\ncmd /C move /Y \"C:\\Users\\iceking\\AppData\\Local\\Temp\\pfywtcji.exe\
lea eax,dword ptr ss:[ebp-195C]
push eax                                              eax:"cmd /C mkdir C:\\Windows\\SysWOW64\\kcnkrmha\\\r\ncmd /C move /Y \"C:\\Users\\iceking\\AppData\\Local\\Temp\\pfywtcji.exe\
call 96baba74a907890b995f23c7db21568f7bfb5
```

Figure 13- Copying itself to system files

Variable values are added to dynamically generated commands. The codes with variable values added are given below.

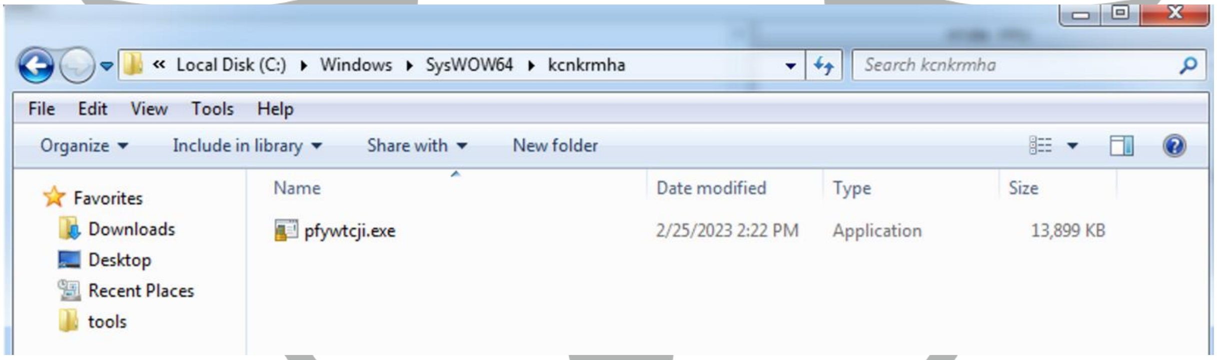
```
cmd /C mkdir C:\\Windows\\SysWOW64\\kcnkrmha\\\r\n

cmd /C move /Y \"C:\\Users\\user\\AppData\\Local\\Temp\\pfywtcji.exe\"
C:\\Windows\\SysWOW64\\kcnkrmha\\\r\n

sc create kcnkrmha binPath= \"C:\\Windows\\SysWOW64\\kcnkrmha\\pfywtcji.exe
/d\\\"C:\\Users\\user\\Downloads\\96baba74a907890b995f23c7db21568f7bfb5dbf417ed90ca3
11482b99702b72.exe\\\"\" type= own start= auto DisplayName= \"wifi support\"\r\n

sc description kcnkrmha \"wifi internet conection\"\r\n

sc start kcnkrmha\r\n
```



Name	Date modified	Type	Size
kcnkrmha	2/25/2023 3:36 PM	File folder	
ko-KR	7/14/2009 6:20 AM	File folder	
LogFiles	7/14/2009 8:32 AM	File folder	

Figure 8- Directory created in SysWOW64

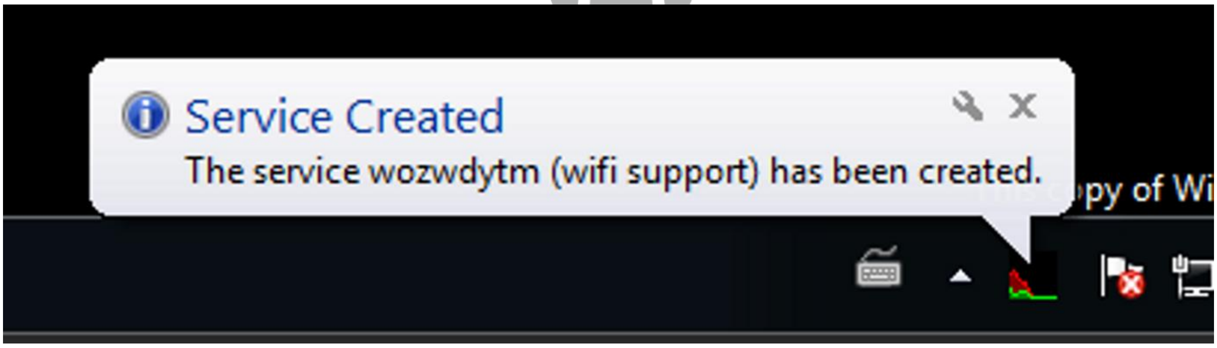
The malware creates a directory in **"C:\Windows\SysWOW64"**.



Name	Date modified	Type	Size
pfywtcji.exe	2/25/2023 2:22 PM	Application	13,899 KB

Figure 9- Malware moved into the created directory

The malware moves the malware file from the C:\Users\user\AppData\Local\Temp directory to C:\Windows\SysWOW64\kcnkrmha.



Service Created

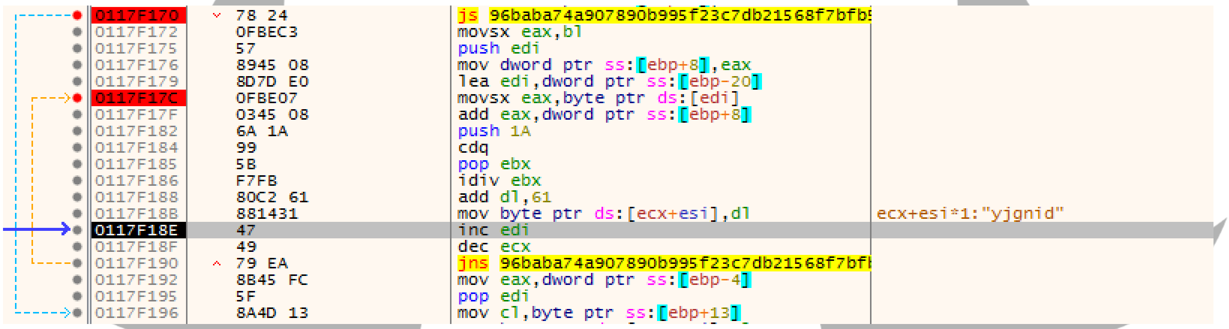
The service wozwdytm (wifi support) has been created.

Figure 10- A service called "wifi support" is being created.

It creates a service with the folder name it created. The service it created hides itself under the name "wifi support".

In the service, the malware gives its own path and runs itself automatically at system boot time. In addition, “wifi internet conection” is added to the service as an description.

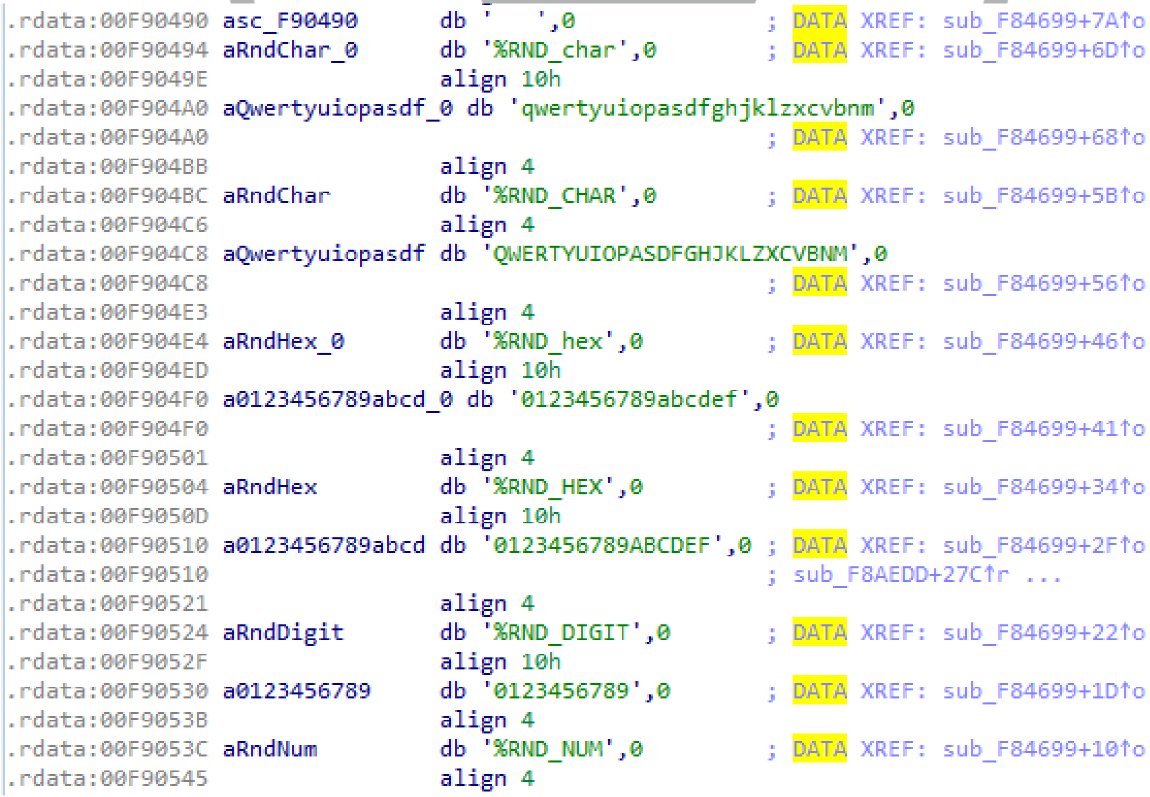
After the service creation is finished, it runs the service it created.



```
0117F170   78 24      js 96baba74a907890b995f23c7db21568f7bfb5
0117F172   0FBEC3     movsx eax,bl
0117F175   57         push edi
0117F176   8945 08    mov dword ptr ss:[ebp+8],eax
0117F179   8D7D E0    lea edi,dword ptr ss:[ebp-20]
0117F17C   0FBE07     movsx eax,byte ptr ds:[edi]
0117F17F   0345 08    add eax,dword ptr ss:[ebp+8]
0117F182   6A 1A      push 1A
0117F184   99         cdq
0117F185   5B         pop ebx
0117F186   F7FB       idiv ebx
0117F188   80C2 61    add dl,61
0117F18B   881431     mov byte ptr ds:[ecx+esi],dl       ecx+esi*1:"yjgnid"
0117F18E   47         inc edi
0117F18F   49         dec ecx
0117F190   79 EA      jns 96baba74a907890b995f23c7db21568f7bfb
0117F192   8B45 FC    mov eax,dword ptr ss:[ebp-4]
0117F195   5F         pop edi
0117F196   8A4D 13    mov cl,byte ptr ss:[ebp+13]
```

Figure 11- It appears to be one of the places where names are randomly generated.

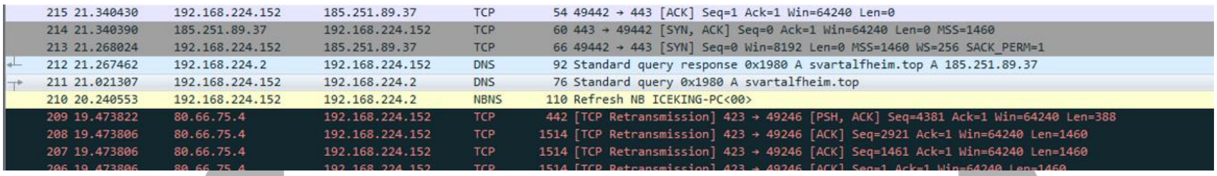
The names of the created backup files, services, folders, firewall rules are randomly generated.



```
.rdata:00F90490 asc_F90490      db '    ',0             ; DATA XREF: sub_F84699+7A↑o
.rdata:00F90494 aRndChar_0      db '%RND_char',0        ; DATA XREF: sub_F84699+6D↑o
.rdata:00F9049E                 align 10h
.rdata:00F904A0 aQwertyuiopasdf_0 db 'qwertyuiopasdfghjklzxcvbnm',0
.rdata:00F904A0                                         ; DATA XREF: sub_F84699+68↑o
.rdata:00F904BB                 align 4
.rdata:00F904BC aRndChar        db '%RND_CHAR',0        ; DATA XREF: sub_F84699+5B↑o
.rdata:00F904C6                 align 4
.rdata:00F904C8 aQwertyuiopasdf db 'QWERTYUIOPASDFGHJKLZXCVBNM',0
.rdata:00F904C8                                         ; DATA XREF: sub_F84699+56↑o
.rdata:00F904E3                 align 4
.rdata:00F904E4 aRndHex_0       db '%RND_hex',0         ; DATA XREF: sub_F84699+46↑o
.rdata:00F904ED                 align 10h
.rdata:00F904F0 a0123456789abcd_0 db '0123456789abcdef',0
.rdata:00F904F0                                         ; DATA XREF: sub_F84699+41↑o
.rdata:00F90501                 align 4
.rdata:00F90504 aRndHex         db '%RND_HEX',0         ; DATA XREF: sub_F84699+34↑o
.rdata:00F9050D                 align 10h
.rdata:00F90510 a0123456789abcd db '0123456789ABCDEF',0 ; DATA XREF: sub_F84699+2F↑o
.rdata:00F90510                                         ; sub_F8AEDD+27C↑r ...
.rdata:00F90521                 align 4
.rdata:00F90524 aRndDigit       db '%RND_DIGIT',0       ; DATA XREF: sub_F84699+22↑o
.rdata:00F9052F                 align 10h
.rdata:00F90530 a0123456789     db '0123456789',0       ; DATA XREF: sub_F84699+1D↑o
.rdata:00F9053B                 align 4
.rdata:00F9053C aRndNum         db '%RND_NUM',0         ; DATA XREF: sub_F84699+10↑o
.rdata:00F90545                 align 4
```

Figure 18- Variables used to create names

These variables are used in the name generation algorithm. With random names, it is difficult to catch with different file and service names at each runtime.



```
  215 21.340430    192.168.224.152    185.251.89.37      TCP        54 49442 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
  214 21.340390    185.251.89.37      192.168.224.152    TCP        60 443 → 49442 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
  213 21.268024    192.168.224.152    185.251.89.37      TCP        66 49442 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
  212 21.267462    192.168.224.2      192.168.224.152    DNS        92 Standard query response 0x1980 A svartalfheim.top A 185.251.89.37
  211 21.021307    192.168.224.152    192.168.224.2      DNS        76 Standard query 0x1980 A svartalfheim.top
  210 20.240553    192.168.224.152    192.168.224.2      NBNS      110 Refresh NB ICEKING-PC<00>
  209 19.473822    80.66.75.4         192.168.224.152    TCP       442 [TCP Retransmission] 423 → 49246 [PSH, ACK] Seq=4381 Ack=1 Win=64240 Len=388
  208 19.473806    80.66.75.4         192.168.224.152    TCP      1514 [TCP Retransmission] 423 → 49246 [ACK] Seq=2921 Ack=1 Win=64240 Len=1460
  207 19.473806    80.66.75.4         192.168.224.152    TCP      1514 [TCP Retransmission] 423 → 49246 [ACK] Seq=1461 Ack=1 Win=64240 Len=1460
  206 19.473806    80.66.75.4         192.168.224.152    TCP      1514 [TCP Retransmission] 423 → 49246 [ACK] Seq=1 Ack=1 Win=64240 Len=1460
```

Figure 19- Potential C2 server in network analysis

Network traffic contains “svartalfheim[.]top” C2 server.

YARA Rule

```
import “hash”

rule tofsee {

    meta:

        author = “Berkay Dogan”

    strings:

        $a1 = “loader_id”

        $a2 = “hi_id”

        $a3 = “born_date”

        $b = "svartalfheim.top"



        $crypt1 = {33 D2 8B C6 F7 F1 81 F6 61 61 61 61 80 C2 61 0F B6 C2}

        $crypt2 = {32 55 14 88 10 8A D1 02 55 18 F6 D9 00 55 14}



    condition:

        hash.md5(0,filesize)  ==  “92E466525E810B79AE23EAC344A52027”
or $a* or $b or $crypt*

}
```

MITRE ATTACK TABLE

Reconnaissance	Execution	Persistence	Discovery	Privilege Escalation	Defense Evasion	C&C	Exfliration
	T-1569 System Services	T-1547 Boot or Logon Autostart Execution	T-1082 System Information Discovery	T-1055 Process Injection	T-1027 Obfuscated Files or Information		
				T-1547 Boot or Logon Autostart Execution	T-1222 File and Directory Permissions Modification		
					T-1036 Creates files inside the user directory		

Solution Proposals

1. You can increase your system security by using good and up-to-date antivirus software.
2. By regularly updating your security software and operating system, you can strengthen its defenses against known attacks.
3. Use trusted websites and download from trusted sources to avoid exposure to malicious websites and downloads.
4. By backing up your important data, you can reduce the risk of data loss caused by malware.

PREPARED BY

Berkay DOĞAN

https://www.linkedin.com/in/berkay-dogan99/