

Goblin Panda changes the dropper and reuses the old infrastructure

medium.com/@Sebdraven/goblin-panda-changes-the-dropper-and-reused-the-old-infrastructure-a35915f3e37a

Sebdraven

December 28, 2018

Sebdraven

Dec 28, 2018

I found a new dropper



c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e with the same signature of others. But it changes the exploitation. Before, it uses the ole package to drop a 8.t file in %appdata% and decode two files, a legit file and an backdoor (PlugX, newcore rat, sysfader...).

Now it's a big object ole mapped in memory and one PE is used to drop the files.

RTF exploit

Now we have four object oles in the RTF.

```
slarntier@chise:~/Documents/projet/Invest/201812275 rtfobj c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e
rtfobj 0.52.1 on python 2.7.15 - http://decalage.info/python/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

=====
File: 'c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e' - size: 1042891 bytes
-----
Id |Index |OLE Object
-----
0 |00078A4h |Not a well-formed OLE object
-----
3 |0007859h |Not a well-formed OLE object
-----
2 |000F99E4h |Not a well-formed OLE object
-----
3 |000F99D2h |Not a well-formed OLE object
-----

slarntier@chise:~/Documents/projet/Invest/201812275 ls -lth c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e_object_000*
-rw-r--r-- 1 slarntier slarntier 484K dec. 27 10:56 c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e_object_0007850.raw
-rw-r--r-- 1 slarntier slarntier 1 dec. 27 10:56 c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e_object_00078A4.raw
-rw-r--r-- 1 slarntier slarntier 9,6K dec. 27 10:56 c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e_object_000F99D2.raw
-rw-r--r-- 1 slarntier slarntier 35 dec. 27 10:56 c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e_object_000F99E4.raw
```

this RTF exploits again the CVE-2017_1882 on eqnedt32.exe.

The biggest object is a new exe ptr overwritten the first eqnedt32.exe . This PE is decoded by the shellcode of the exploit and executes this PE.

```
001EE955 33D2 xor edx,edx
001EE957 BB 3E4A8F50 mov ebx,508F4A3E
001EE95C 3955 FC cmp dword ptr ss:[ebp-4],edx
001EE95F 7E 0E jlt 1EE98F
001EE961 6A 07 push 7
001EE963 5F pop edi
001EE964 8BCB mov ecx,ebx
001EE966 8BC3 mov ebx,ebx
001EE968 C1E9 18 shr ecx,18
001EE96B 83E0 07 and eax,7
001EE96E 33CB xor ecx,ebx
001EE970 03DB add ebx,ebx
001EE972 C1E9 03 shr ecx,3
001EE975 83E1 01 and ecx,1
001EE978 33C8 xor ecx,eax
001EE97A 0BD9 or ebx,ecx
001EE97C 4F dec edi
001EE97D 75 E5 jlt 1EE964
001EE97F 8B45 F0 mov eax,dword ptr ss:[ebp-10]
001EE982 8B4D FC mov ecx,dword ptr ss:[ebp-4]
001EE985 301C02 xor byte ptr ds:[edx+eax],01
001EE988 42 inc edx
001EE989 3BD1 cmp edx,ecx
001EE98B 7C D4 jlt 1EE964
```

It's the same algorithm of the exploit targeting Vietnam.

0066C82A	B8 63 8E F4 7B	mov eax,7BF48E63
0066C82F	39 55 FC	cmp dword ptr ss:[ebp-4],edx
0066C832	7E 22	jle 66C856
0066C834	6A 07	push 7
0066C836	5F	pop edi
0066C837	8B C8	mov ecx,eax
0066C839	C1 E9 1B	shr ecx,1B
0066C83C	33 C8	xor ecx,eax
0066C83E	C1 E9 03	shr ecx,3
0066C841	33 C8	xor ecx,eax
0066C843	03 C0	add eax,eax
0066C845	83 E1 01	and ecx,1
0066C848	0B C1	or eax,ecx
0066C84A	4F	dec edi
0066C84B	75 EA	jne 66C837
0066C84D	30 04 1A	xor byte ptr ds:[edx+ebx],al

Just the init key has changed.

Dropper

the PE creates a directory in %appdata% named IISWebClient

```

.text:004011B5      push     esi          ; nSize
.text:004011B6      lea     eax, [ebp+Dat]
.text:004011BC      push     eax          ; lpDst
.text:004011BD      push     offset Src   ; "%appdata%"
.text:004011C2      call    ds:ExpandEnvironmentStringsA
.text:004011C8      mov     edi, ds:lstrcatA
.text:004011CE      lea     eax, [ebp+Dat]
.text:004011D4      push     eax          ; lpString2
.text:004011D5      lea     eax, [ebp+String1]
.text:004011DB      push     eax          ; lpString1
.text:004011DC      call    edi ; lstrcatA
.text:004011DE      push     offset String2 ; "\\IISWebClient"
.text:004011E3      lea     eax, [ebp+String1]
.text:004011E9      push     eax          ; lpString1
.text:004011EA      call    edi ; lstrcatA
.text:004011EC      push     ebx          ; lpSecurityAttributes
.text:004011ED      lea     eax, [ebp+String1]
.text:004011F3      push     eax          ; lpPathName
.text:004011F4      call    ds:CreateDirectoryA

```

After, it decrypt a buffer in the address space of the executable with a xor.

```

.text:00401223      push     eax ; lpPathName
.text:00401224      call    ds:SetCurrentDirectoryA
.text:0040122A      call    decrypt_loop

```

```

; Attributes: bp-based frame
decrypt_loop proc near
dwFreeType= dword ptr -20h
SystemTime= _SYSTEMTIME ptr -14h
var_4= dword ptr -4

push    ebp
mov     ebp, esp
sub     esp, 14h
mov     eax, ___security_cookie
xor     eax, ebp
mov     [ebp+var_4], eax
push    esi
push    edi
push    4           ; flProtect
push    1000h      ; flAllocationType
push    0F83Bh     ; dwSize
xor     edi, edi
push    edi        ; lpAddress
call    ds:VirtualAlloc
push    1           ; dwMilliseconds
mov     esi, eax
call    ds:$sleep
lea     eax, [ebp+SystemTime]
push    eax        ; lpSystemTime
call    ds:GetSystemTime
movzx   eax, [ebp+SystemTime.wMilliseconds]
xor     edx, edx
mov     ecx, 0FFh
div     ecx
lea     ecx, [esi+1]
push    ebx
lea     eax, [edx+1]
mov     [esi], al
mov     dword ptr [ecx], 0F799h
add     ecx, 4

```

```

loc_401540:
movzx   eax, al
imul   eax, 0Dh
add     eax, 7
xor     edx, edx
mov     ebx, 0FFh
div     ebx
mov     al, dl
xor     byte_455E08[edi], al
inc     edi
mov     edx, 0F82Bh
cmp     edi, edx
jnb     short loc_401540

```

```

mov     eax, offset byte_455E08
sub     eax, ecx
mov     edi, edx
pop     ebx

```

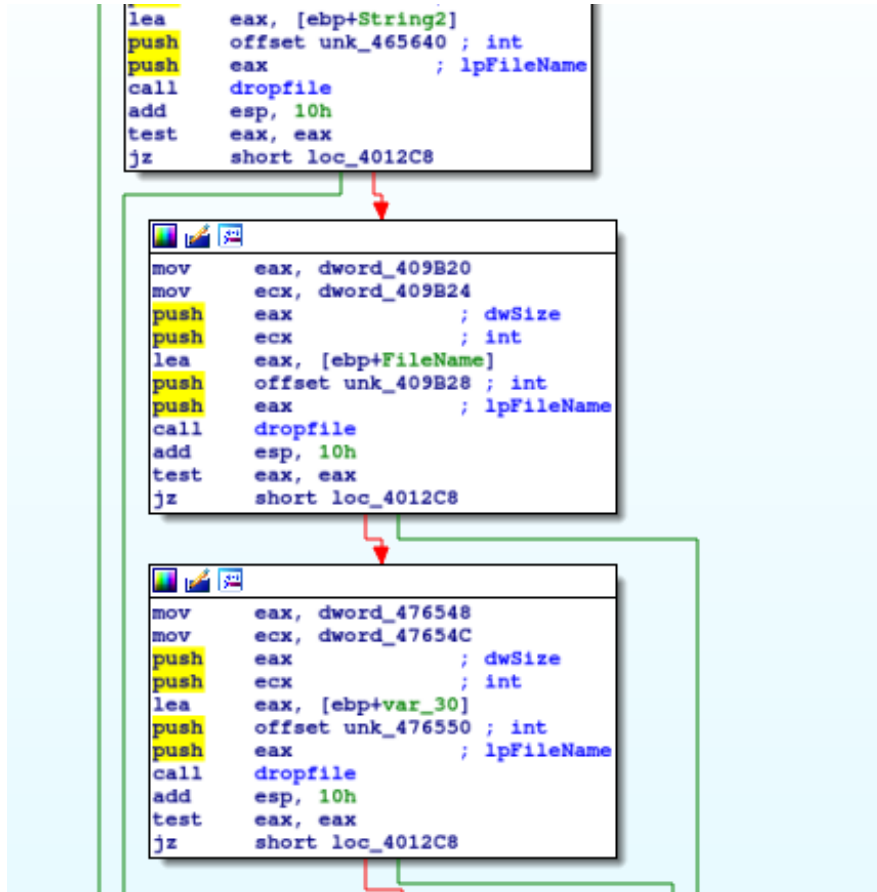
```

loc_40156E:
mov     dl, [eax+ecx]
mov     [ecx], dl
inc     ecx
dec     edi
jnz     short loc_40156E

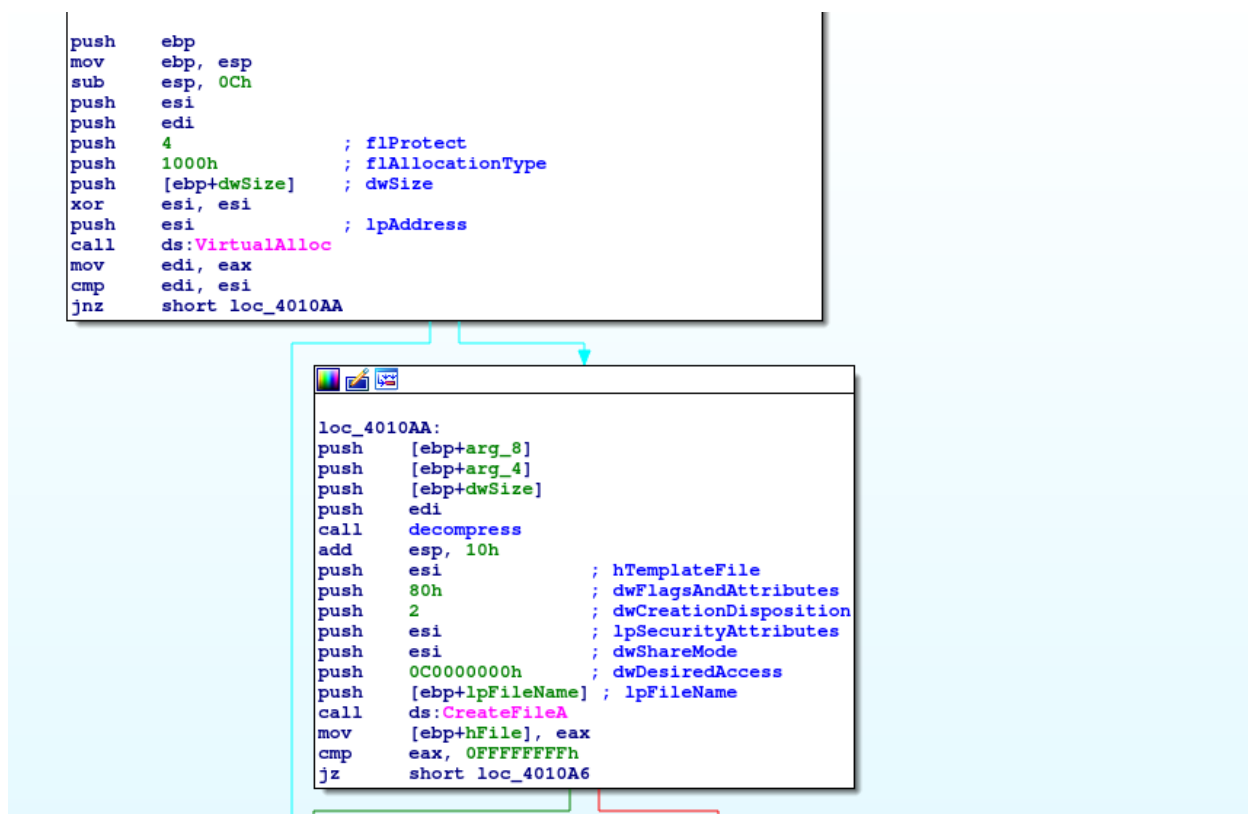
```

After the decoding, the PE decompresses and drops three files on the disk: iassvcs.exe developed and signed by Symantec, sqlite3.dll signed by the av 360 and RasTls.dll the backdoor.

The first step is to call dropfile



In this function the first step is to allocate at 0016000 a memory page and to decompress the file before writing on the disk.



```

push    edi
call    decompress
add     esp, 10h
push    esi           ; hTemplateFile
push    80h           ; dwFlagsAndAttributes
push    2             ; dwCreationDisposition
push    esi           ; lpSecurityAttributes
push    esi           ; dwShareMode
push    0C000000h     ; dwDesiredAccess
push    [ebp+lpFileName] ; lpFileName
call    ds:CreateFileA
mov     [ebp+hFile], eax
cmp     eax, 0FFFFFFFh
jz     short loc_4010A6

```

```

mov     eax, [ebp+dwSize]
push    ebx
xor     ebx, ebx
mov     [ebp+nNumberOfBytesToWrite], eax
cmp     eax, esi
jbe    short loc_40110F

```

```

loc_4010E9:           ; lpOverlapped
push    esi
lea     eax, [ebp+NumberOfBytesWritten]
push    eax           ; lpNumberOfBytesWritten
push    [ebp+nNumberOfBytesToWrite] ; nNumberOfBytesToWrite
lea     eax, [ebx+edi]
push    eax           ; lpBuffer
push    [ebp+hFile]   ; hFile
call    ds:WriteFile
test   eax, eax
jz     short loc_40112D

```

the function decompress loads dynamically the function RtlDecompressBuffer and use it.

```

decompress proc near
hLibModule= dword ptr -0Ch
var_8= byte ptr -8
uBytes= dword ptr 8
arg_0= dword ptr 8
arg_4= dword ptr 0Ch
arg_8= dword ptr 10h
arg_C= dword ptr 14h

push    ebp
mov     ebp, esp
sub     esp, 0Ch
push    edi
push    offset LibFileName ; "ntdll.dll"
call    ds:LoadLibraryA
mov     edi, eax
mov     [ebp+hLibModule], edi
test   edi, edi
jz     short loc_40107F

```

```

push    ebx
push    esi
mov     esi, ds:GetProcAddress
push    offset ProcName ; "RtlDecompressBuffer"
push    edi           ; hModule
call    esi ; GetProcAddress
push    offset aRtlGetcompress ; "RtlGetCompressionWorkSpaceSize"
push    edi           ; hModule
mov     ebx, eax
call    esi ; GetProcAddress
test   ebx, ebx
jz     short loc_401074

```

```

test   eax, eax
jz     short loc_401074

```

```

lea     ecx, [ebp+var_8]
push    ecx
lea     ecx, [ebp+uBytes]
push    ecx
mov     esi, 102h
push    esi
call    eax
push    [ebp+uBytes] ; uBytes
push    40h         ; uFlags
call    ds:LocalAlloc
mov     edi, eax
lea     eax, [ebp+var_8]
push    eax
push    [ebp+arg_C]
push    [ebp+arg_8]
push    [ebp+arg_4]
push    [ebp+arg_0]
push    esi
call    ebx
push    edi         ; hMem
call    ds:LocalFree

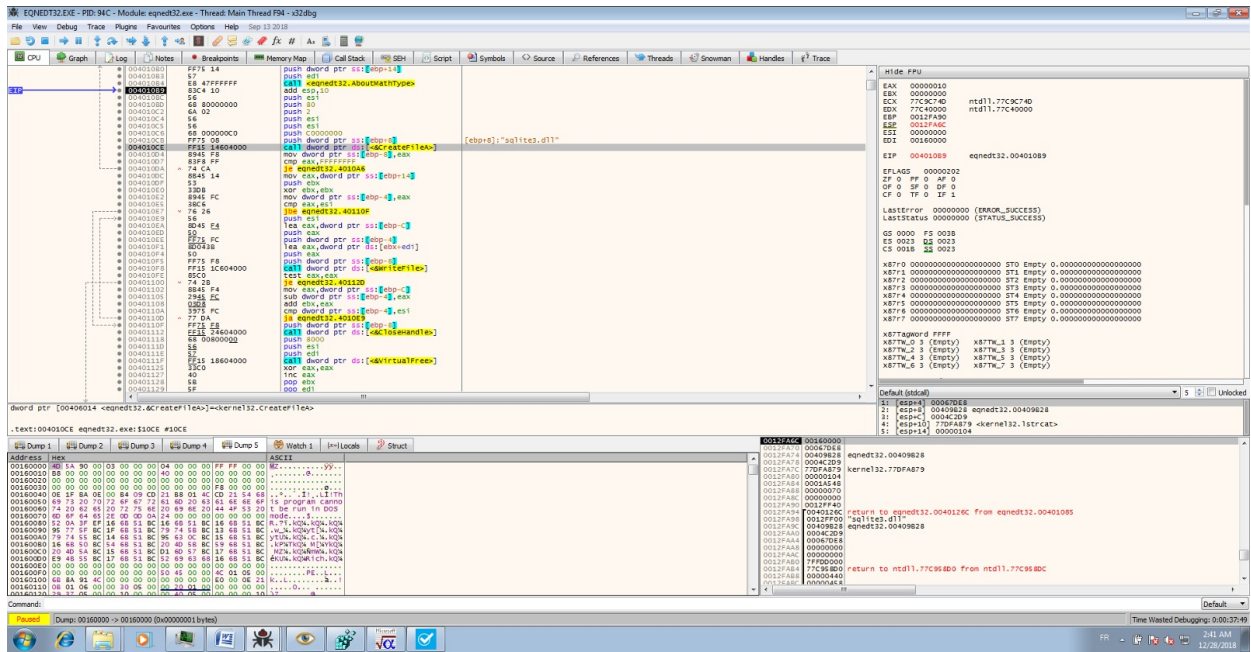
```

```

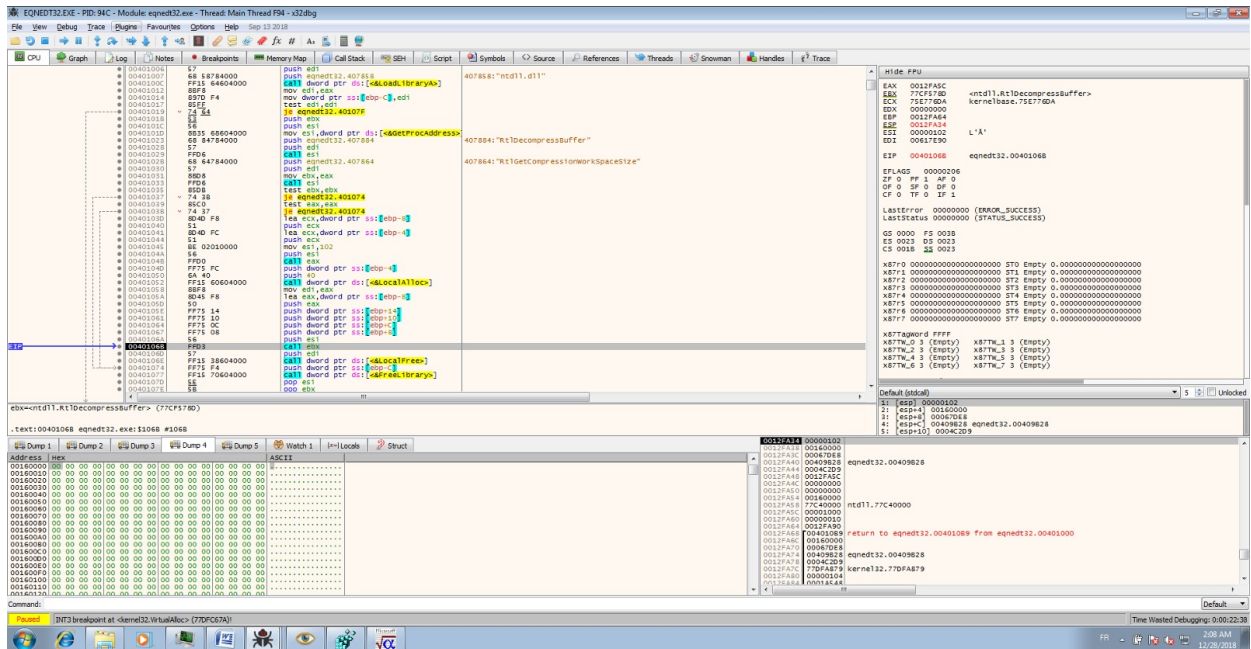
loc_401074:           ; hLibModule
push    [ebp+hLibModule]
call    ds:FreeLibrary
pop     esi
pop     ebx

```

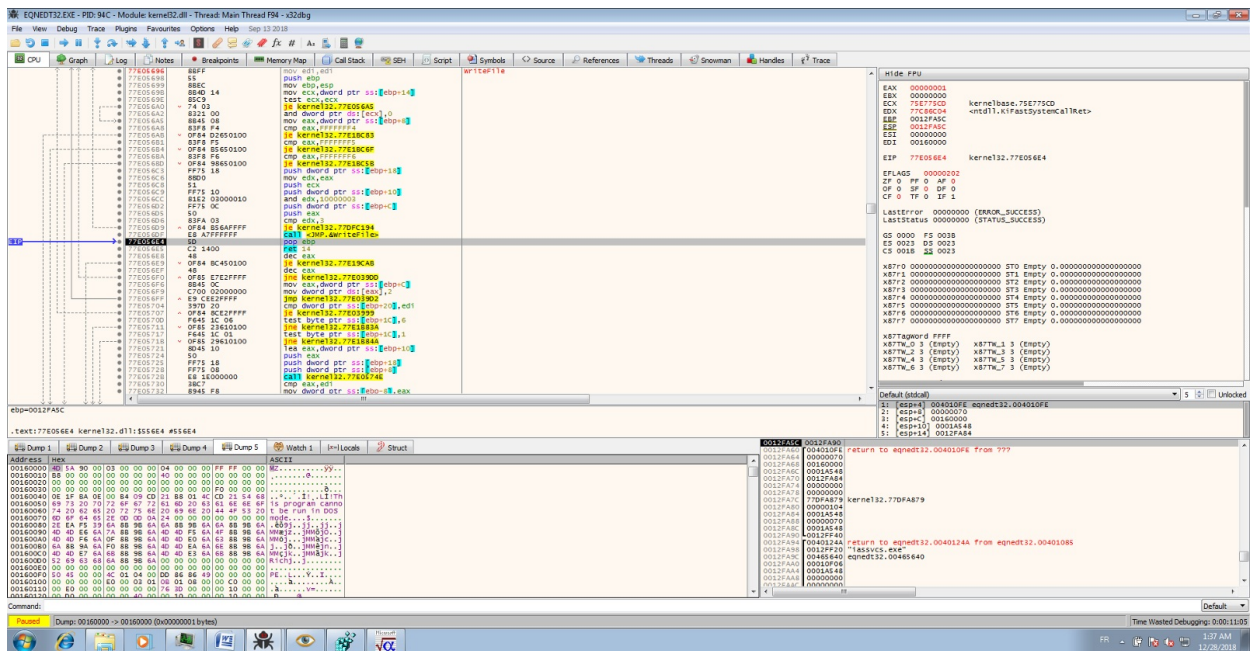
In the debugger, these steps are:



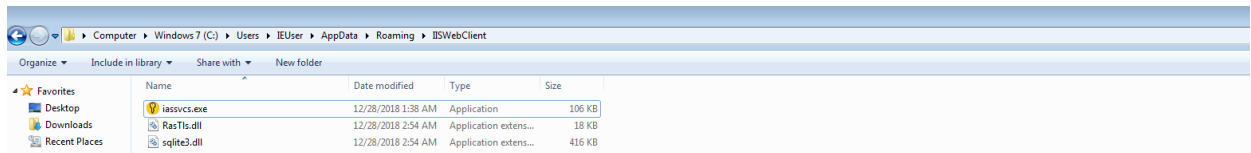
CreateFile



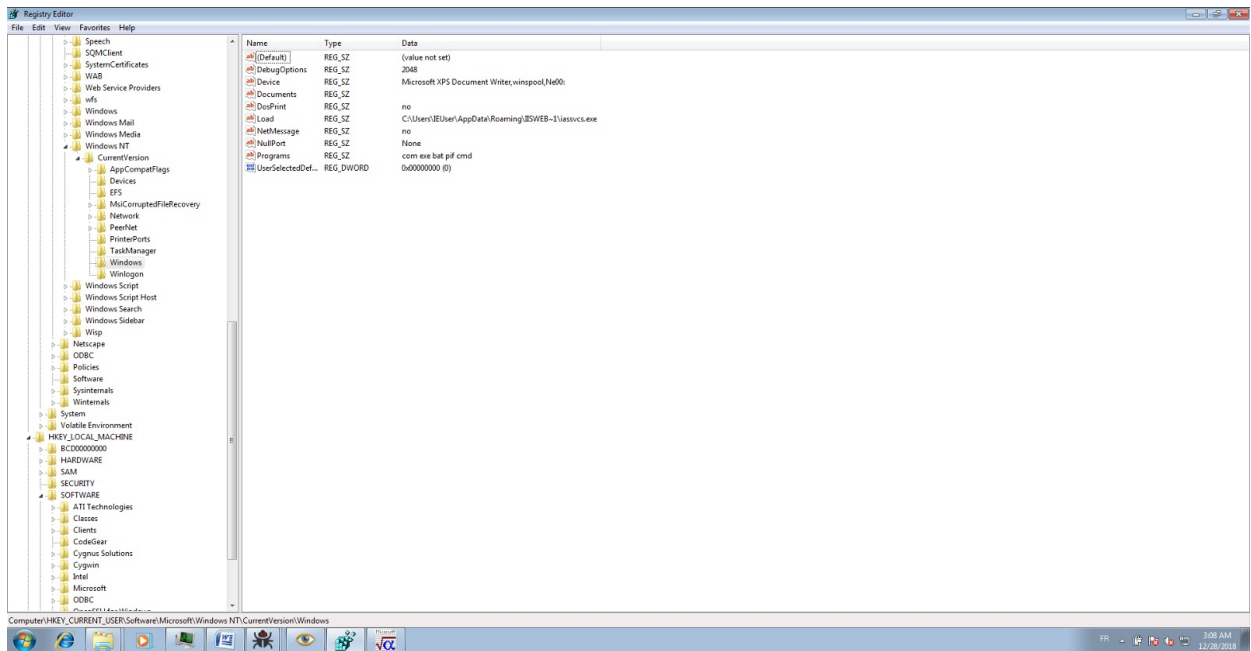
Decompress



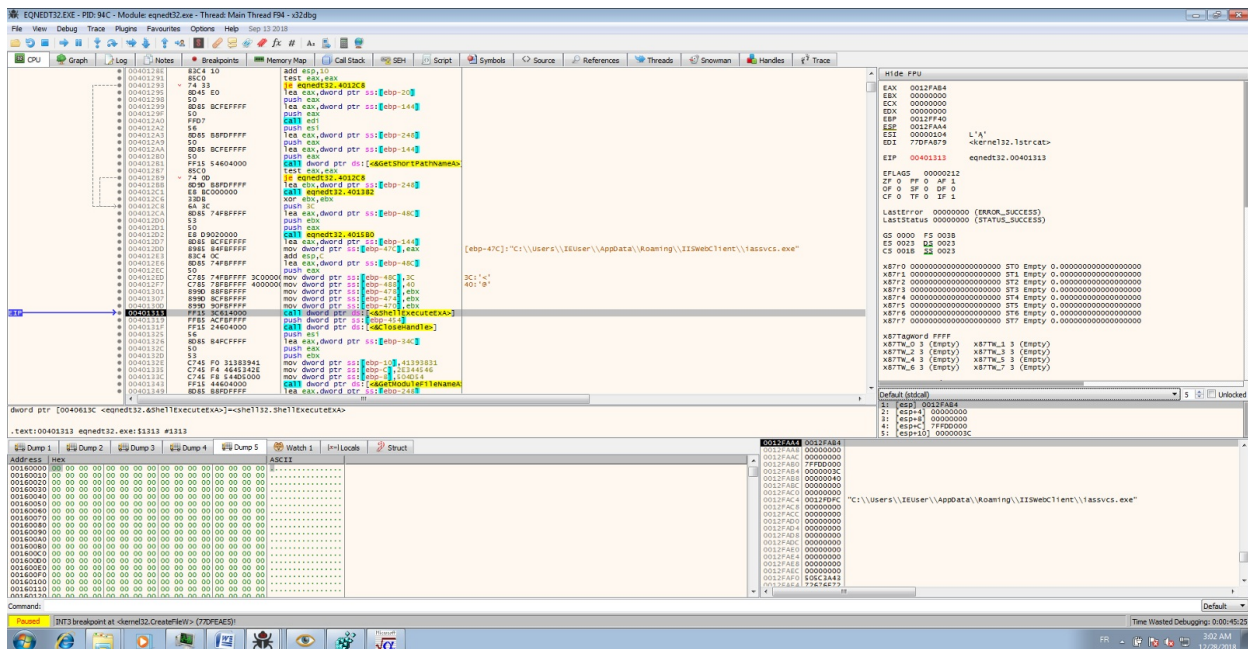
Write the buffer on the disk after decompressing



And the dropper execute the iassvcs.exe to make a side loading and make the persistence.



at each executable or cmd line, the executable is reloaded.



In a second part, I'll analysed the backdoor.

Threat Intelligence

the backdoor contacts [hxxps://skylineqaz.crabdance.com/](https://skylineqaz.crabdance.com/)

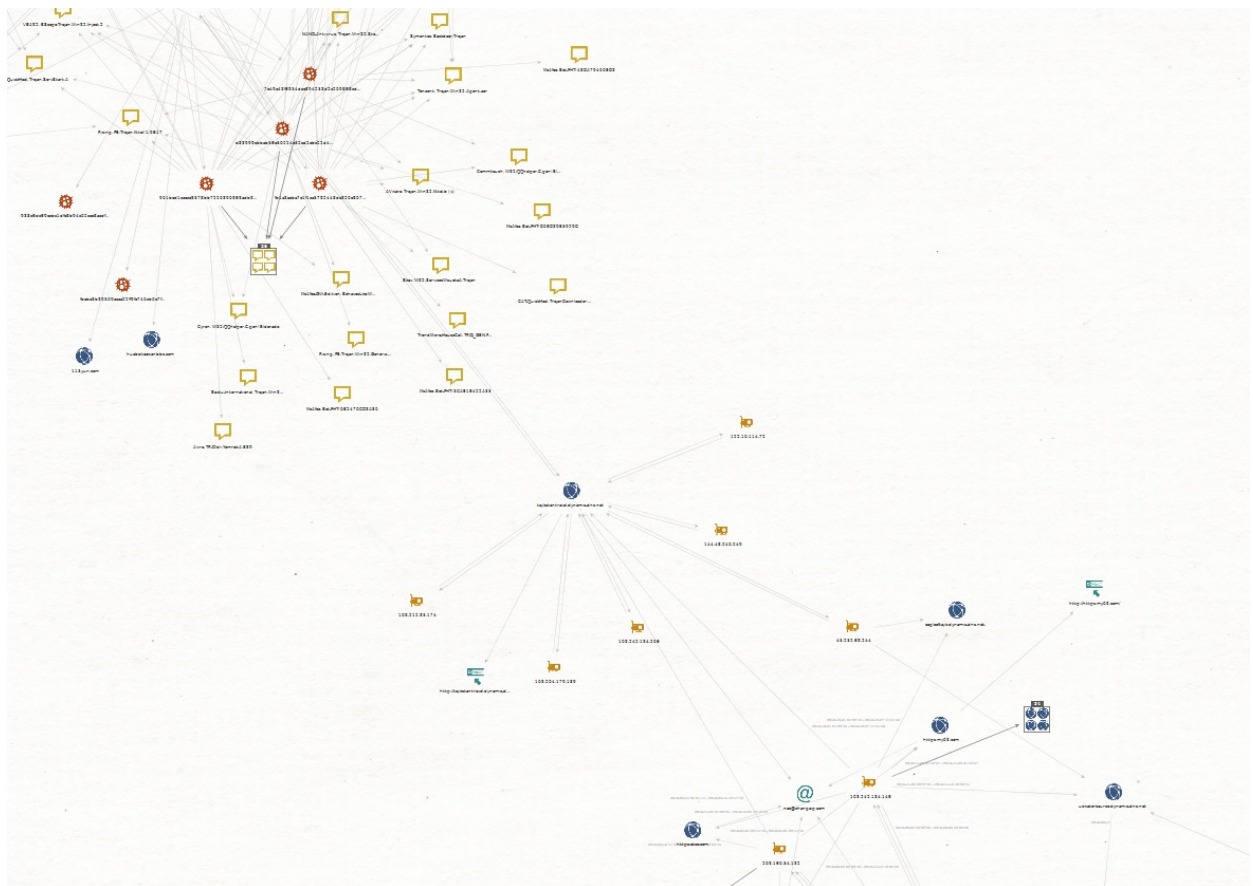
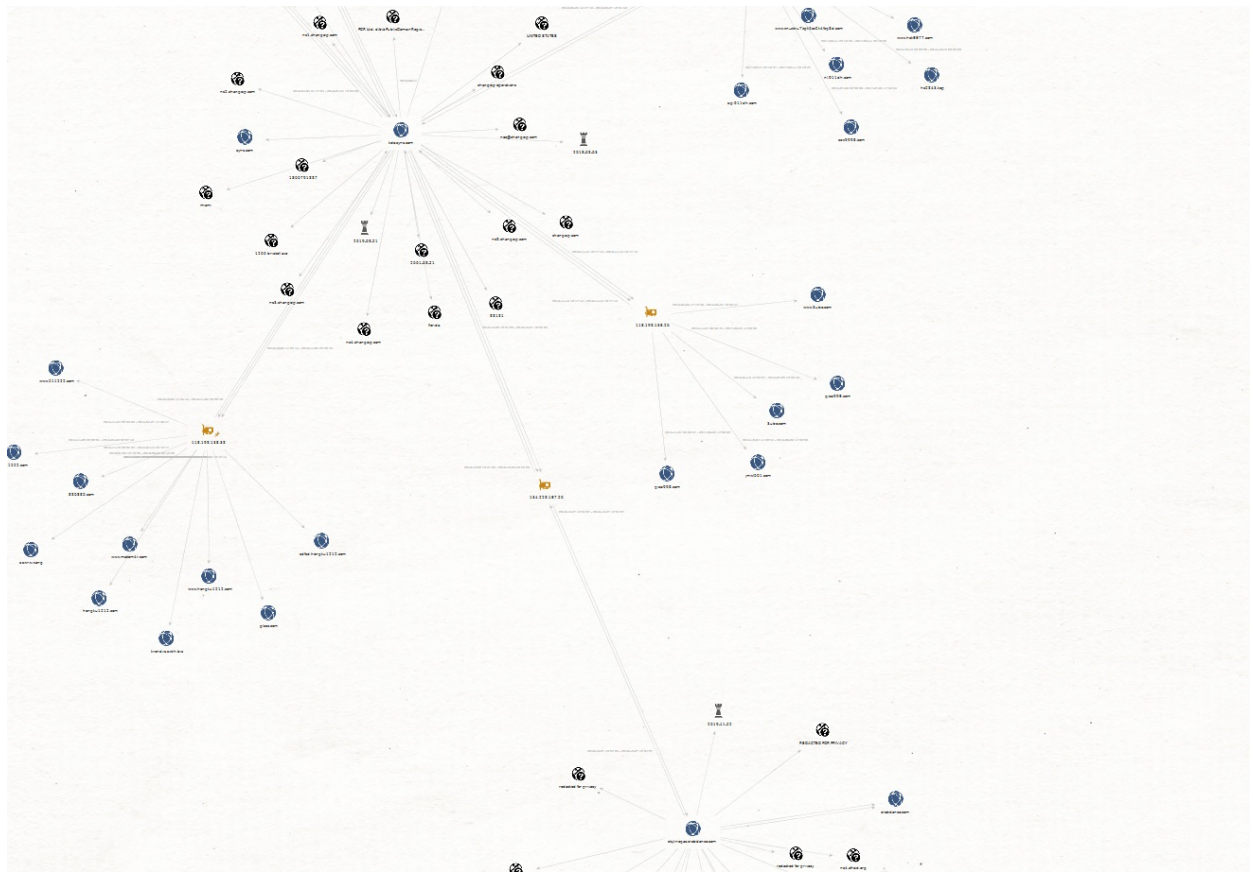
the domain resolves 154.223.167.20. This IP is very interesting because it connects with tele.zyns.com and old infrastructures used by chinese APT or DDOS Chinese team against the ancient soviet republics.

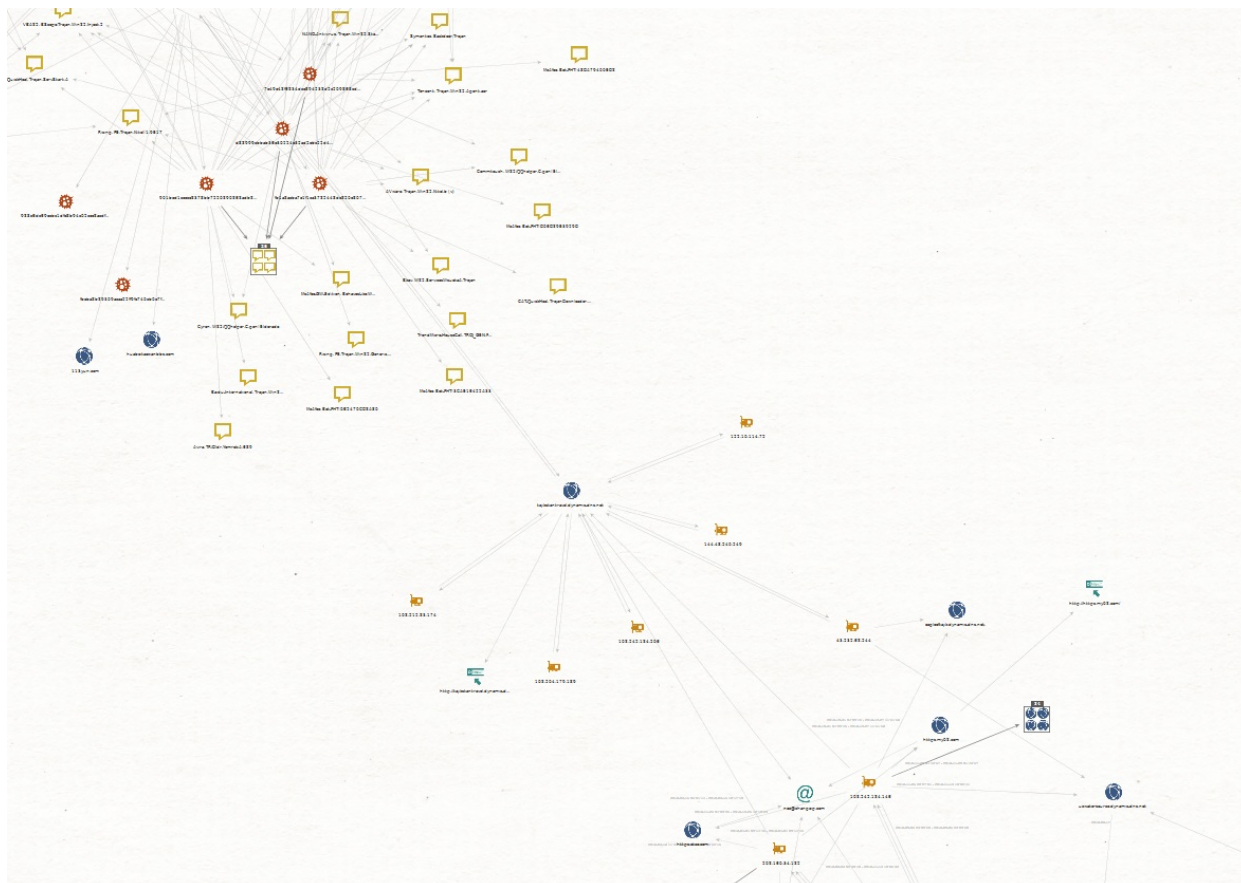
Also, the name of domains show the targets is energy and telecom sectors.

they find uzwatersource.dynamic-dns.net used by Icefog connected by the 150.129.80.184 to tele.zyns.com

uzwatersource.dynamic-dns.net connects to tajikstantravel.dynamic-dns.net by 45.252.63.244.

The domain tajikstantravel.dynamic-dns.net connected to ddos infrastructure chinese.





Another thing is the dropper what has submitted by an ID coming of the Kazakhstan. So Goblin Panda targets it ?

IOCs

the dropper:

dropper "c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e"
 sha256 c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e
 sha1 398fb04ce9b2e30bce932590e0b86b594c8a97ea
 md5 30528dc0c1e123dff51f40301cc03204

Dropped executable file

sha256 C:\Users\admin\AppData\Roaming\IISWebClient\RasTls.dll
 eb0b848f18d8002aaf59faca18b28941df67dc46891868b96fa4daf03018d148
 sha256 C:\Users\admin\AppData\Roaming\IISWebClient\iassvcs.exe
 f9ebf6aeb3f0fb0c29bd8f3d652476cd1fe8bd9a0c11cb15c43de33bbce0bf68
 sha256 C:\Users\admin\AppData\Roaming\IISWebClient\sqlite3.dll
 e342eefb43249a3a1b62b8622f7c94fc391c0488bdae7e1909e37cb125029f1c

DNS requests

domain skylineqaz.crabdance.com
 domain xn--ylineqaz-y25ja.crabdance.com

Connections

ip 154.223.167.20

HTTP/HTTPS requests

url <https://skylineqaz.crabdance.com/>

tele.zyns.com 103.242.134.146 150.129.80.184 (Goblin Panda, Icefog)

uzwatersource.dynamic-dns.net 150.129.80.184 (Icefog)