

# Chinese Actor APT target Ministry of Justice Vietnamese

 [medium.com/@Sebdraven/chineses-actor-apt-target-ministry-of-justice-vietnamese-14f13cc1c906](https://medium.com/@Sebdraven/chineses-actor-apt-target-ministry-of-justice-vietnamese-14f13cc1c906)

May 10, 2019



Sebdraven

May 11

With the new RTF exploit using 8.t to store their payloads, many malicious document have targeted to Vietnam. One document is very interesting because its target specifically the Ministry of Justice.



## Analyze

The recipients of the document

41f0757ca4367f22b0aece325208799135c96ebe1dcafc752d3f3c8dd4a5ccf 8.t are (at the end of the document):

the deputy minister;

- the units under the ministry;
- police of provinces and cities directly under the central government;
- Department of Inspection of Legal Documents of the Ministry of Justice;
- Official Journal, Government Electronic Portal, Ministry of Public Security Portal;
- Archive: VT, C06 (P1).

The document exploits Equation Editor starts application (CVE-2017-11882) to decode the 8.t in memory, after fork to install two files:

C:\Users\admin\AppData\Local\Temp\wsc.dll  
4e88f8a3c3be45e0a59a8868f2b2ace51754fcdbfa9ab618e3d9d0e17831990f

and

C:\Users\admin\AppData\Local\Temp\wsc\_proxy.exe  
1948bb0df11f768d6dd30ae7ecec5550db7c817d09cb31b5e2cee9b86a4047da

The malware is a dll, it seems to be Gh0st RAT.

<https://app.any.run/tasks/5715cfe3-2550-4808-aad0-1ea4c4fc7a88>

An to start the malware, it uses a side loading technics with a scheduled Task.

The exe call in the entry loads dynamically wsc.dll and call the function \_run@4

```

Decompile: entry - (wsc_proxy.bin)
1
2 void entry(void)
3
4 {
5     code *pcVar1;
6     HMODULE hModule;
7     FARPROC pVar2;
8     LPWSTR pwVar3;
9     UINT uExitCode;
10
11     hModule = GetModuleHandleW(L"kernel32.dll");
12     pVar2 = GetProcAddress(hModule,"SetDefaultDllDirectories");
13     SetDllDirectoryW(L"");
14     if (pVar2 != (FARPROC)0x0) {
15         (*pVar2)(0x1000);
16     }
17     hModule = LoadLibraryW(L"wsc.dll");
18     if (hModule == (HMODULE)0x0) {
19         uExitCode = GetLastError();
20     }
21     else {
22         pVar2 = GetProcAddress(hModule,"_run@4");
23         if (pVar2 == (FARPROC)0x0) {
24             uExitCode = GetLastError();
25             FreeLibrary(hModule);
26         }
27         else {
28             pwVar3 = GetCommandLine();
29             uExitCode = (*pVar2)(pwVar3);
30             FreeLibrary(hModule);
31         }
32     }
33     ExitProcess(uExitCode);
34     pcVar1 = (code *)swi(3);
35     (*pcVar1)();
36     return;
37 }
38

```

### Side Loading

The screenshot shows a debugger window with the following details:

- Register Window:**
  - eax=0013F668
  - dword ptr [esp+1D0]=0012F5F8
- Instruction List:**
  - 00401659: call SetDefaultDllDirectories
  - 00401660: push 0
  - 00401661: push ecx
  - 00401662: call SetDefaultDllDirectories
  - 00401663: pop ecx
  - 00401664: pop edi
  - 00401665: pop esi
  - 00401666: pop ebp
  - 00401667: xor eax, eax
  - 00401668: pop ebx
  - 00401669: add esp, 280
  - 00401670: jmp
  - 00401671: jmp
  - 00401672: jmp
  - 00401673: jmp
  - 00401674: jmp
  - 00401675: jmp
  - 00401676: jmp
  - 00401677: jmp
  - 00401678: jmp
- Right Pane (Assembly):**

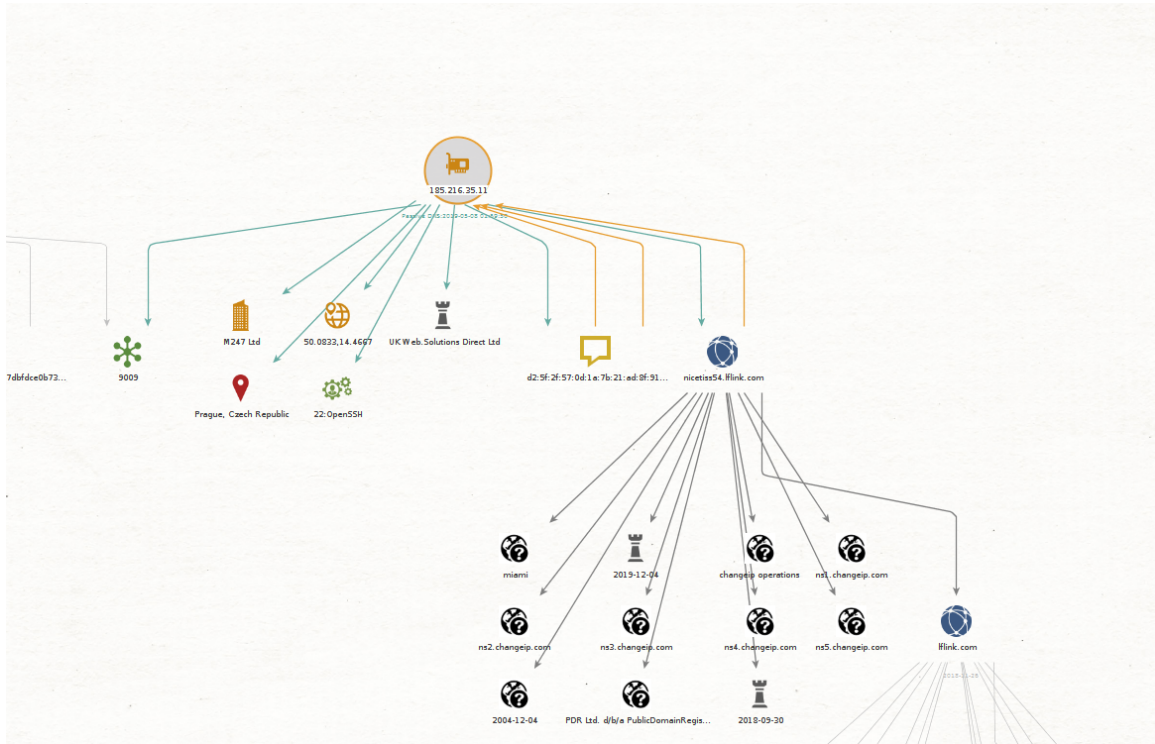
```

41D02C: "%s\%s\" /tr \"%s\" /mo s /f"
mov     esi, eax
add     esp, 1C
test    esi, esi
jz     eqned32.401668
push   esi
push   0A0
push   1
push   eqned32.400EC8
CALL   ebx
push   esi
CALL   esp
lea    edx, dword ptr ss:[esp+108]
push   edx
CALL   eqned32.4011A0
add     esp, 18
test   al, al
jz     eqned32.401668
lea    eax, dword ptr ss:[esp+90]
push   eax
CALL   eqned32.4011A0
add     esp, 1
test   al, al
jz     eqned32.401668
lea    ecx, dword ptr ss:[esp+90]
lea    edx, dword ptr ss:[esp+38]
push   ecx
lea    eax, dword ptr ss:[esp+6C]
push   eax
push   ecx
lea    ecx, dword ptr ss:[esp+1C8]
push   ecx
push   eqned32.41D02C
CALL   dword ptr ds:[!$$_printfs]
lea    edx, dword ptr ss:[esp+100]
push   0
push   edx
CALL   eqned32.401080
pop     edi
pop     esi
pop     ebp
xor     eax, eax
pop     ebx
add     esp, 280
jmp

```

### Scheduled Task

The RAT tries to connect to nicetiss54.lflink.com|185.216.35.11.



## Threat Intelligence Consideration

We have the same TTps and victimology like Goblin Panda:

- Officials Vietnamese
- Side Loading
- 8.t RTF kit exploit
- a dynamic dns name

But the payload has changed and the launch of the backdoor has changed.

It's used a scheduled task.

Usually this group uses NewCoreRat.

## IOCs

Main object-

"41f0757ca4367f22b0aece325208799135c96ebe1dcafc752d3f3c8dd4a5ccf"  
 sha256 41f0757ca4367f22b0aece325208799135c96ebe1dcafc752d3f3c8dd4a5ccf  
 sha1 6e670a837970a1fb4161d77d5f720d318d7e4dbc  
 md5 f34514118eb4689560cd6c0c654f26d9

Dropped executable file

sha256 C:\Users\admin\AppData\Local\Temp\wsc.dll  
 4e88f8a3c3be45e0a59a8868f2b2ace51754fcd9a9ab618e3d9d0e17831990f

sha256 C:\Users\admin\AppData\Local\Temp\wsc\_proxy.exe  
 1948bb0df11f768d6dd30ae7ecec5550db7c817d09cb31b5e2cee9b86a4047da

DNS requests

domain nicetiss54.lflink.com

Connections

ip 185.216.35.11