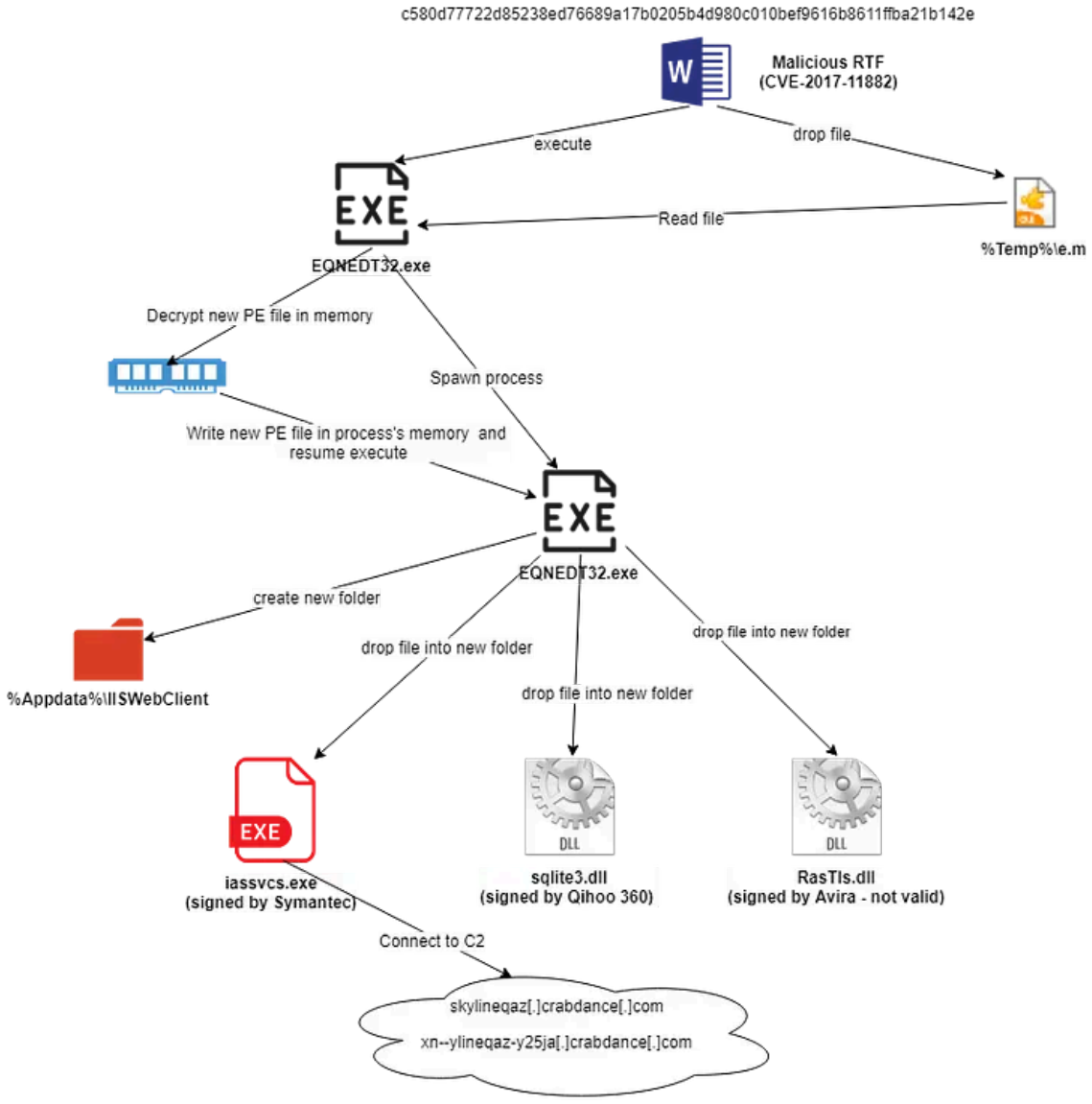
Another malicious document with CVE-2017–11882

By m4n0w4r

Published: 2019-01-03 · Archived: 2026-04-05 15:35:16 UTC

Press enter or click to view image in full size



Overview

Nhờ người em hỗ trợ, tôi có được một sample mới ***c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e*** sử dụng **CVE-2017–11882**. Sample này ***có thay đổi chút về OLE object, init_key để decrypt binary, cũng như các dropped binary*** so với mẫu tôi đã viết tại đây https://tradahacking.vn/l%C3%A0-1937cn-hay-oceanlotus-hay-lazarus-6ca15fe1b241

1. Stage 1 — Phân tích sơ bộ

Kiểm tra thấy đây là một file RTF:

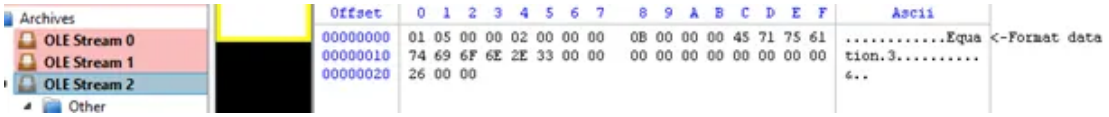
property	value
md5	30528DC0C1E123DFF51F40301CC03204
sha1	398FB04CE9B2E30BCE932590E0B86B594C8A97EA
sha256	C580D77722D85238ED76689A17B0205B4D980C010BEF9616B8611FFBA21B142E
first-bytes (hex)	7B 5C 72 74 66 31 5C 61 64 65 66 6C 61 6E 67 31 30 32 35 5C 61 6E 73 69 5C 61 6E 73 69 63 70 67 31
first-bytes (text)	{\rtf1\adeflang1025\ansi\ansicpg1
size	1042891 bytes
entropy	3.824

Sử dụng **rtfobj** để xem có các embedded objects không, thấy có **4 objects**:

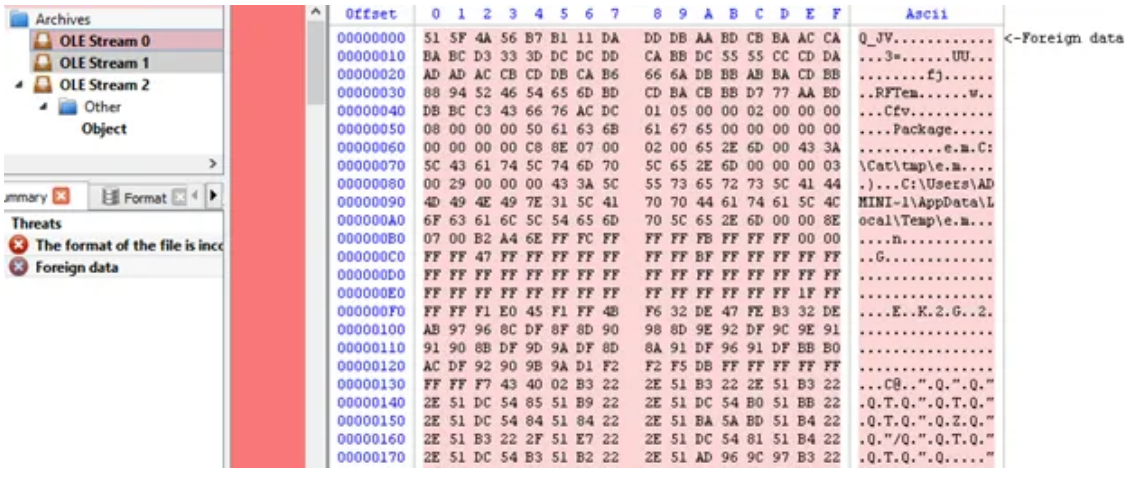
```
C:\Users\Administrator\Desktop\Sample>rtfobj c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e
rtfobj 0.54dev1 on Python 2.7.10 - http://decalage.info/python/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

===============================================================================
File: 'c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e' - size: 1042891 bytes
---+----------+--------------------------------------------------------------
id |index     |OLE Object
---+----------+--------------------------------------------------------------
0  |00007BA4h |Not a well-formed OLE object
---+----------+--------------------------------------------------------------
1  |00007850h |Not a well-formed OLE object
---+----------+--------------------------------------------------------------
2  |000F99E4h |Not a well-formed OLE object
---+----------+--------------------------------------------------------------
3  |000F99D2h |Not a well-formed OLE object
---+----------+--------------------------------------------------------------
```

Thông qua Profiler, có được thông tin sau:

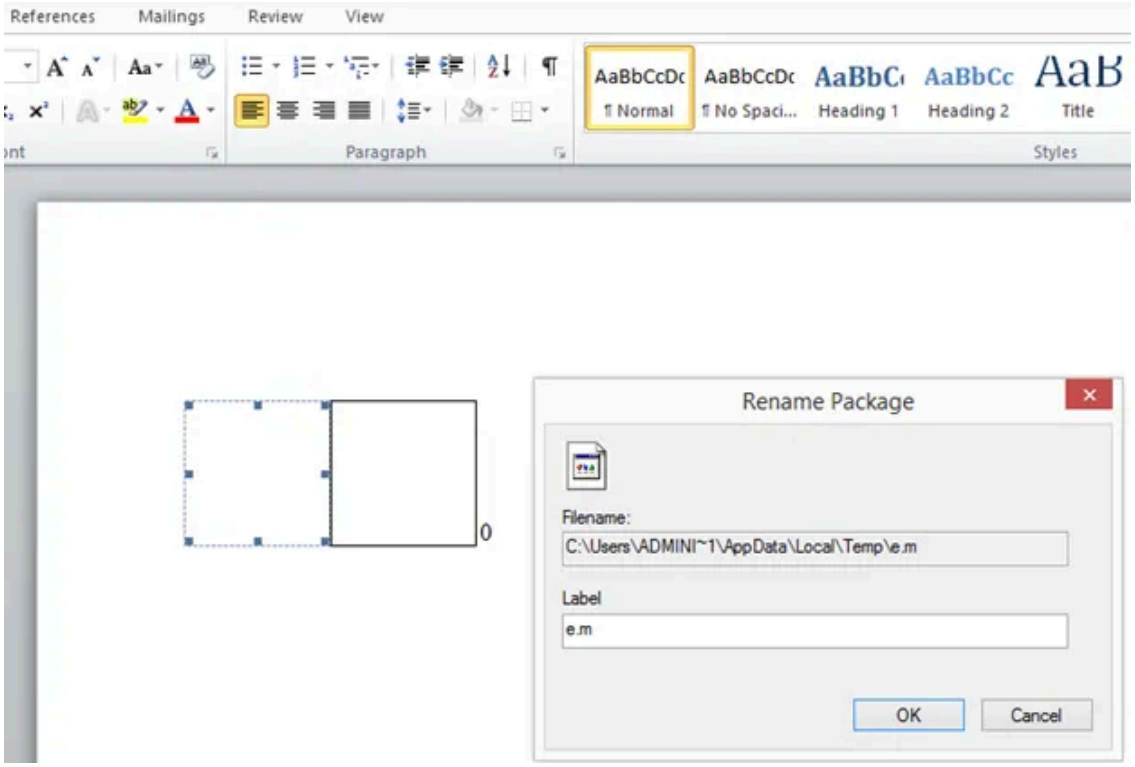


CVE-2017–11882 Signature



Embedded file

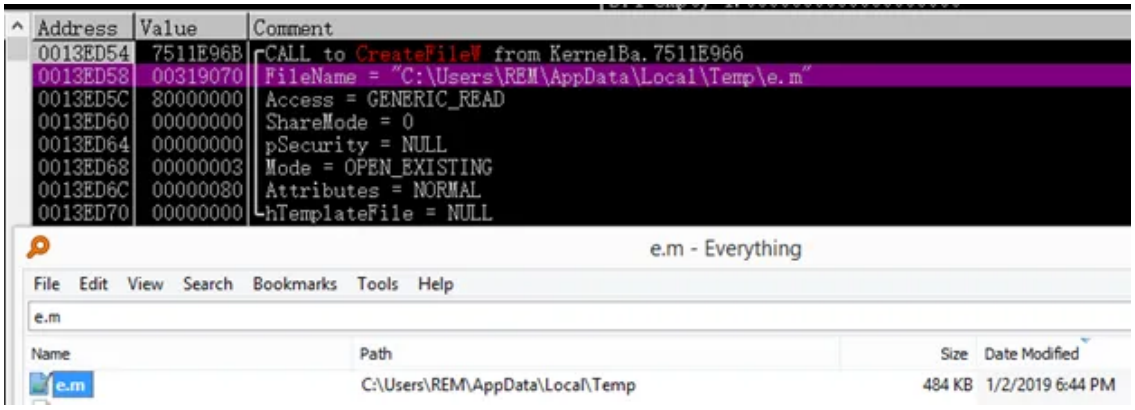
Mở file bằng ứng dụng Word không thấy có nội dung gì (*theo đánh giá cá nhân, tụi này làm mẫu không bằng các mẫu nhắm vào VN, không viết nổi một nội dung cho tử tế :D*). Nó sẽ drop file **e.m** vào thư mục Temp . File này sẽ có nội dung như trên hình:

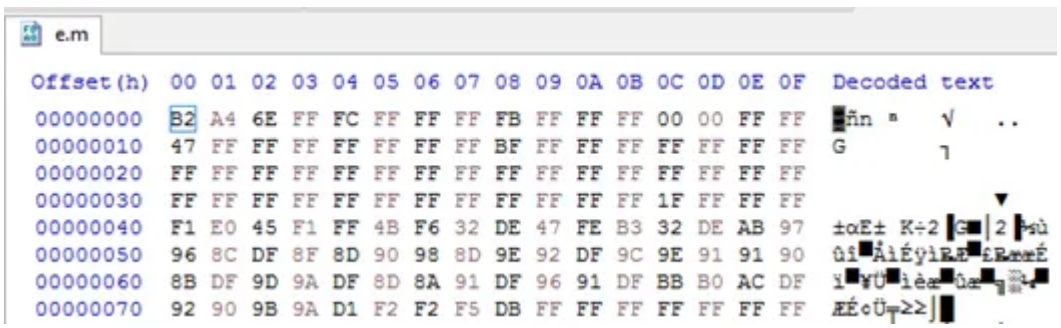


2. Stage 2 — Lấy binary được giải mã

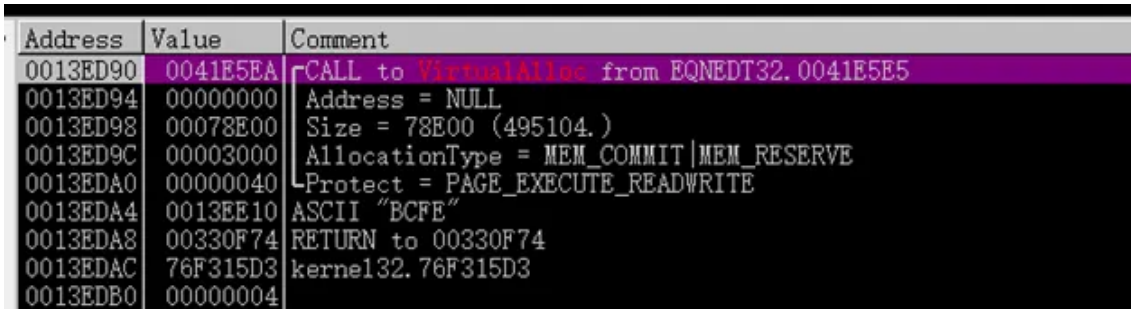
Với sample này, tôi không áp dụng được dụng tính năng **Image File Execution Options (IFEO)** nên tôi dùng HxD để patch Entry Point của **EQNEDT32.exe** thành **0xEB 0xEF**.

Sau đó mở file bằng ứng dụng Word, dùng OllyDBG tiến hành attach tiến trình **EQNEDT32.exe**. Sau khi attach xong khôi phục lại các bytes gốc đã patch bằng HxD. Đặt một bp tại **CreatFileW**:

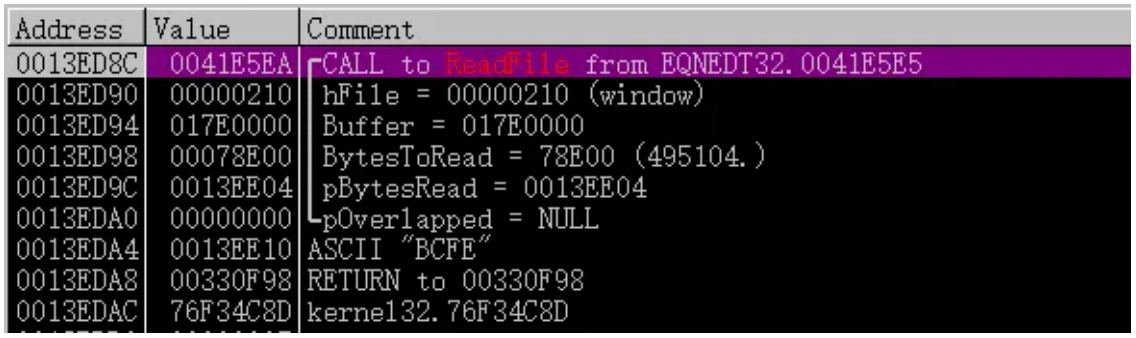


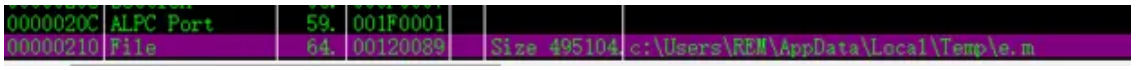


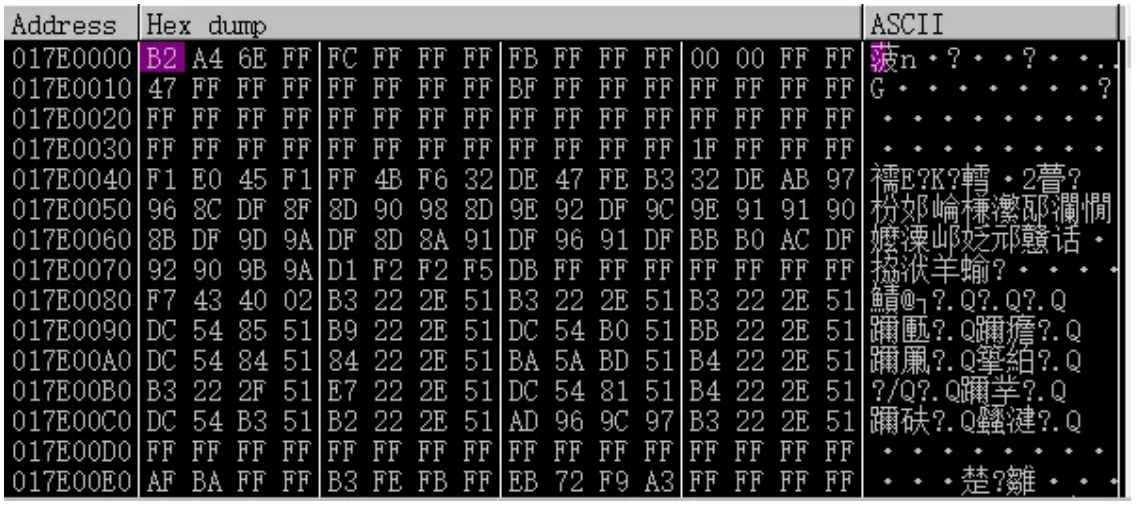
Thông qua shellcode gọi hàm **VirtualAlloc** để cấp phát vùng nhớ phục vụ cho việc lưu nội dung của file e.m:



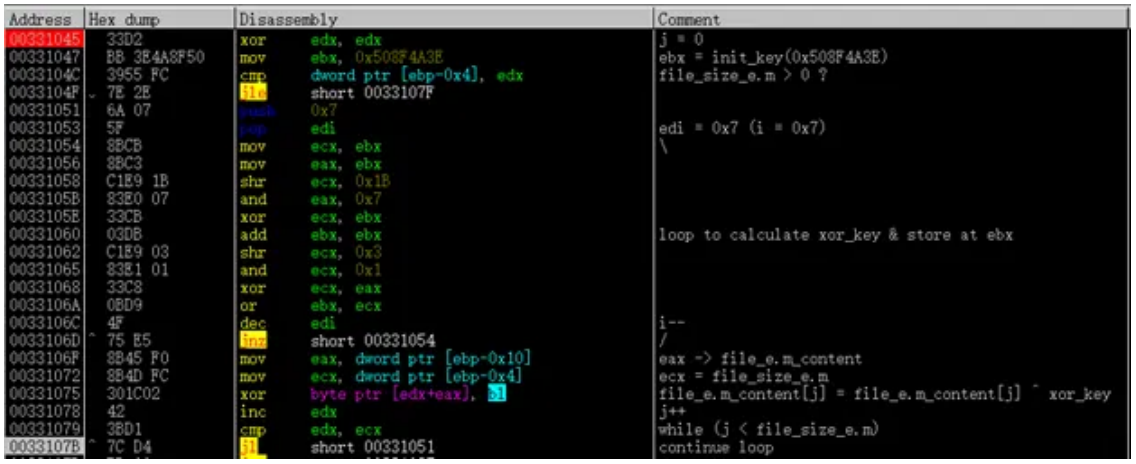
Tiếp theo gọi hàm **ReadFile** để đọc nội dung từ e.m và lưu vào vùng nhớ ở trên:



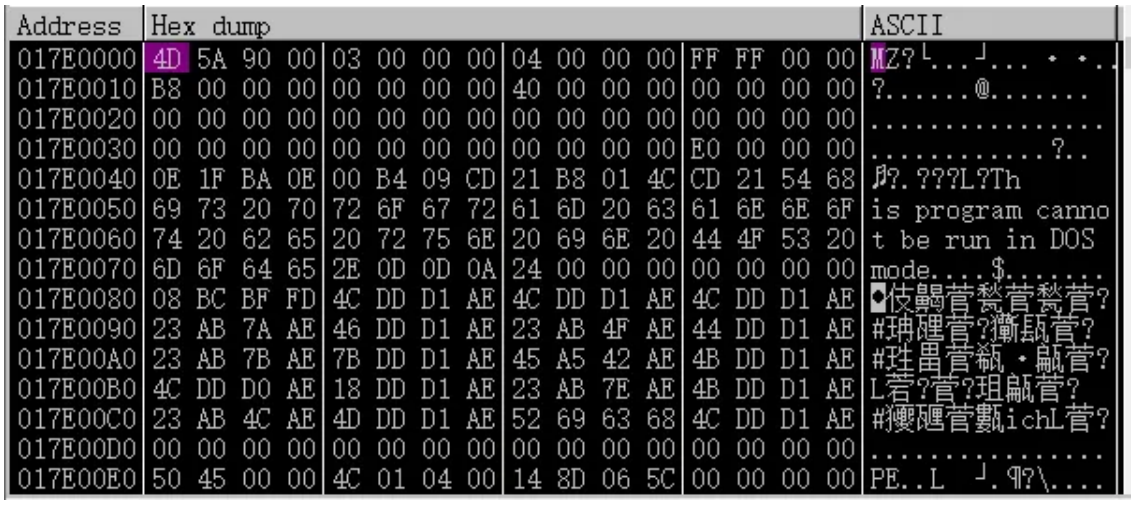




Dùng vòng lặp xor để giải mã dữ liệu tại vùng nhớ trên (*thuật toán tương tự như bài viết https://tradahacking.vn/l%C3%A0-1937cn-hay-oceanlotus-hay-lazarus-6ca15fe1b241, chỉ khác* ***init_key***):



Sau vòng lặp trên có được một PE file mới như sau:



Dump vùng nhớ này ra đĩa để phân tích:

Property	Value
File Name	C:\Users\REM\Desktop_02330000.mem
File Type	Portable Executable 32
File Info	Microsoft Visual C++ 8
File Size	484.00 KB (495616 bytes)
PE Size	483.50 KB (495104 bytes)
Created	Thursday 03 January 2019, 09.16.52
Modified	Thursday 03 January 2019, 09.16.52
Accessed	Thursday 03 January 2019, 09.16.52
MD5	1D34E10BE052156F030EC93345C40B74
SHA-1	8D7425AE30FD2D5196EC4DCD2540B31A0D26772F

_02330000.mem

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	73	00007B30	00000000	00000000	00007DE6	00006014
ADVAPI32.dll	4	00007B1C	00000000	00000000	00007E36	00006000
SHELL32.dll	1	00007C58	00000000	00000000	00007E56	0000613C

3. Stage 3 — Phân tích binary đã dump

Load binary ở trên vào IDA, nó thực hiện tạo thư mục có tên **IISWebClient** tại **%appdata%**:

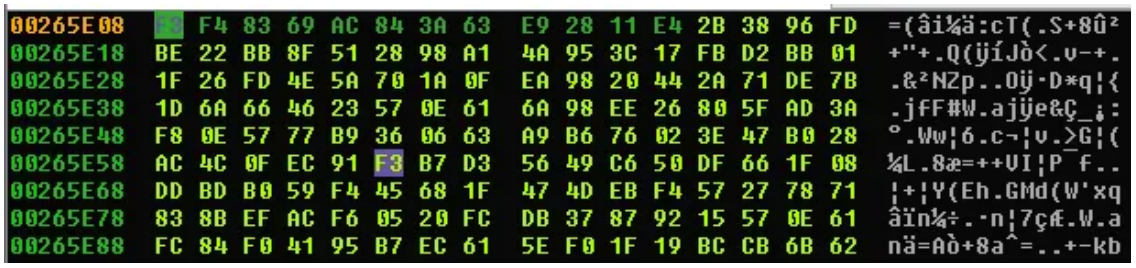
```
strcpy(sz_iassvcs_exe, "iassvcs.exe");
sz_RasTls_dll[0] = 'TsaR';                    // RasTls.dll
sz_RasTls_dll[1] = 'd.sl';
sz_RasTls_dll[2] = 'll';
strcpy(sz_sqlite3_dll, "sqlite3.dll");
GetCurrentDirectoryA(0x104u, &lpCurDir);
memset(&lpPathName, 0, 0x104u);
ExpandEnvironmentStringsA("%appdata%", &Dst, 0x104u);
lstrcatA(&lpPathName, &Dst);
lstrcatA(&lpPathName, "\\IISWebClient");
if ( CreateDirectoryA(&lpPathName, 0) || GetLastError() == ERROR_ALREADY_EXISTS )
{
  lstrcatA(&lpPathName, "\\");
  SetCurrentDirectoryA(&lpPathName);
```

```
szCUsersRemAp_1 db 'C:\Users\REM\AppData\Roaming\IISWebClient',0
```

Thực hiện giải mã một buffer:

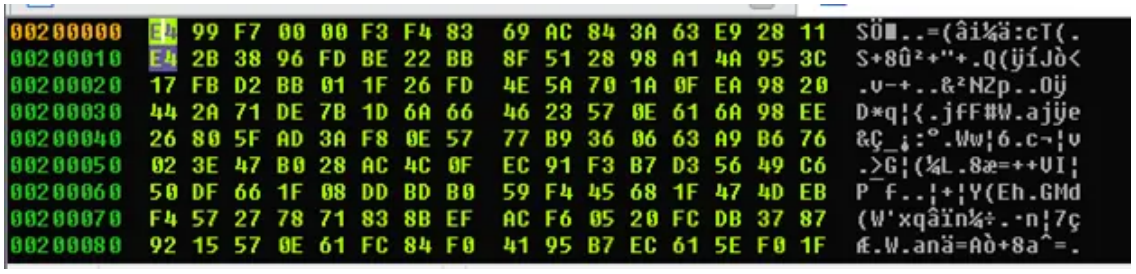
Press enter or click to view image in full size

```
i = 0;
mem_alloc = VirtualAlloc(0, 0xF83Bu, MEM_COMMIT, PAGE_READWRITE);
Sleep(1u);
GetSystemTime(&SystemTime);
v2 = SystemTime.wMilliseconds % 0xFFu + 1;
*(_BYTE *)mem_alloc = v2;
*(_DWORD *)((char *)mem_alloc + 1) = 0xF799;
pmem_alloc = (char *)mem_alloc + 5;
// decrypt_loop
do
{
  v2 = (0xD * (unsigned int)v2 + 7) % 0xFF;
  byte_265E08[i] ^= v2;
  ++i;
}
while ( i < 0xF82B );
v4 = (char *)(byte_265E08 - pmem_alloc);
```



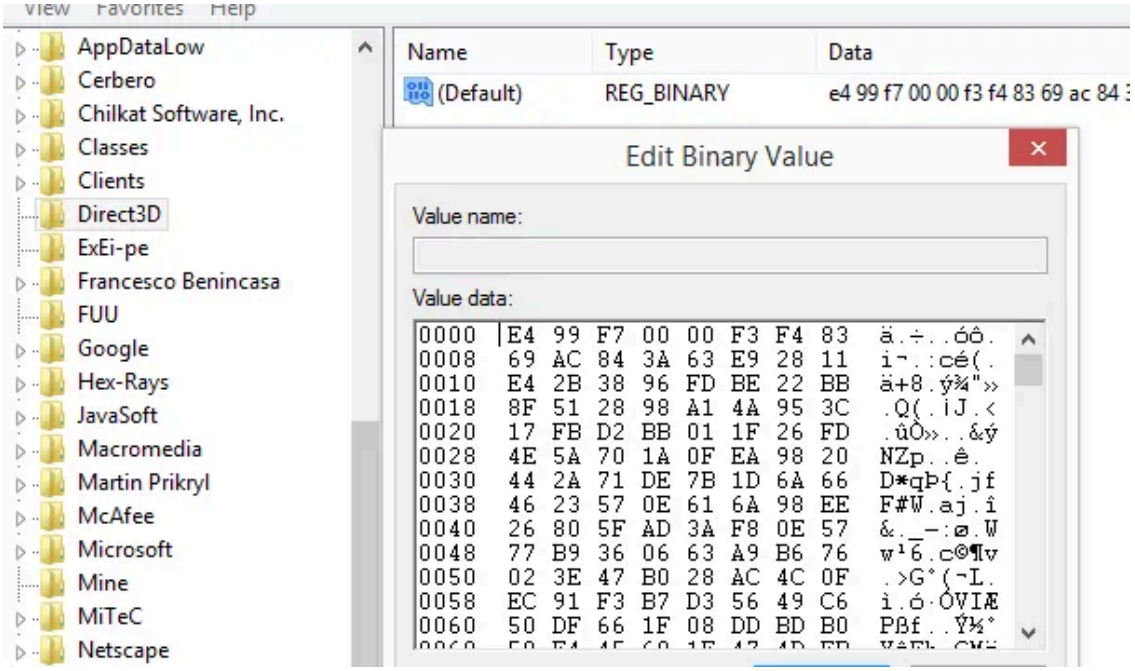
Sau đó copy toàn bộ các bytes đã giải mã ở trên vào vùng nhớ đã được cấp phát:

```
j = 0xF82B;
// copy decrypted bytes to allocated mem
do
{
  *pmem_alloc = pmem_alloc[(_DWORD)v4];
  ++pmem_alloc;
  --j;
}
while ( j );
result_1 = Query_Registry_Key((BYTE *)mem_alloc);
VirtualFree(mem_alloc, 0, MEM_RELEASE);
return result_1;
```



Tạo một key là “**Direct3D**” tại **HKEY_CURRENT_USER\Software** & lưu toàn bộ decrypted bytes:

```
result_1 = 0;
if ( !RegOpenKeyExA(HKEY_CURRENT_USER, "Software", 0, KEY_CREATE_SUB_KEY, &phkResult)
  && !RegCreateKeyExA(phkResult, "Software", 0, 0, 0, KEY_WRITE, 0, &hKey, &dwDisposition) )
{
  RegSetValueExA(hKey, 0, 0, REG_BINARY, lpData + 0x10, 0x10u);
  if ( !RegCreateKeyExA(phkResult, "Direct3D", 0, 0, 0, KEY_WRITE, 0, &hkey, &dwDisposition)
    && !RegSetValueExA(hkey, 0, 0, REG_BINARY, lpData, 0xF830u) )
  {
    result_1 = 1;
  }
}
```



Tiếp theo, drop 3 files vào thư mục IISWebClient đã tạo ở trên:

Get m4n0w4r’s stories in your inbox

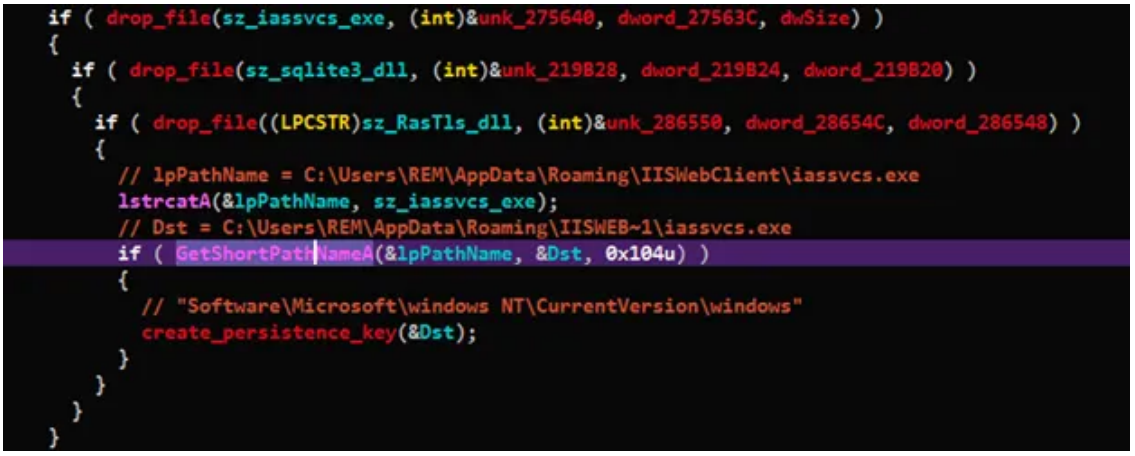
Join Medium for free to get updates from this writer.

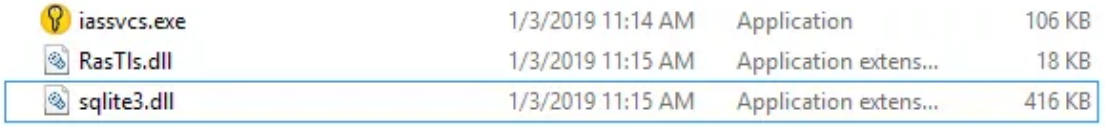
Remember me for faster sign in

· iassvcs.exe (signed by *Symantec*).

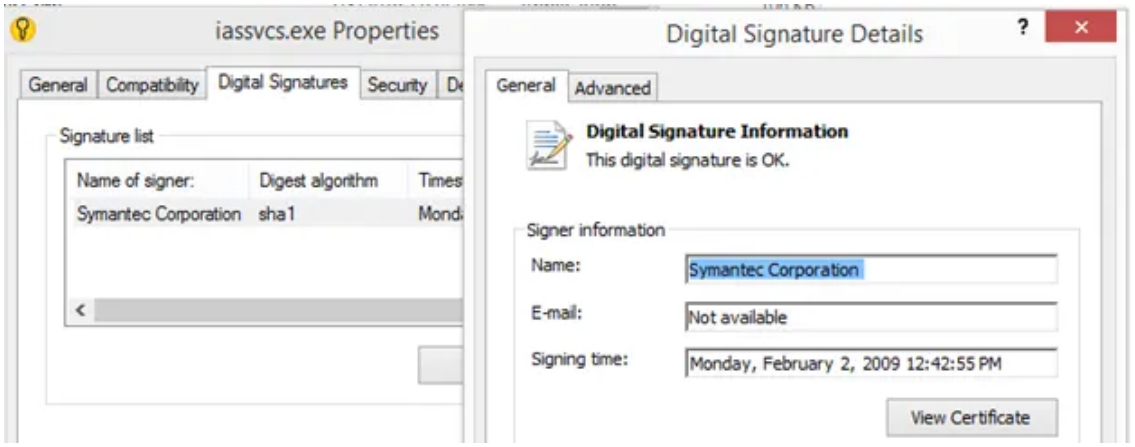
· sqlite3.dll (signed by *Qihoo 360*).

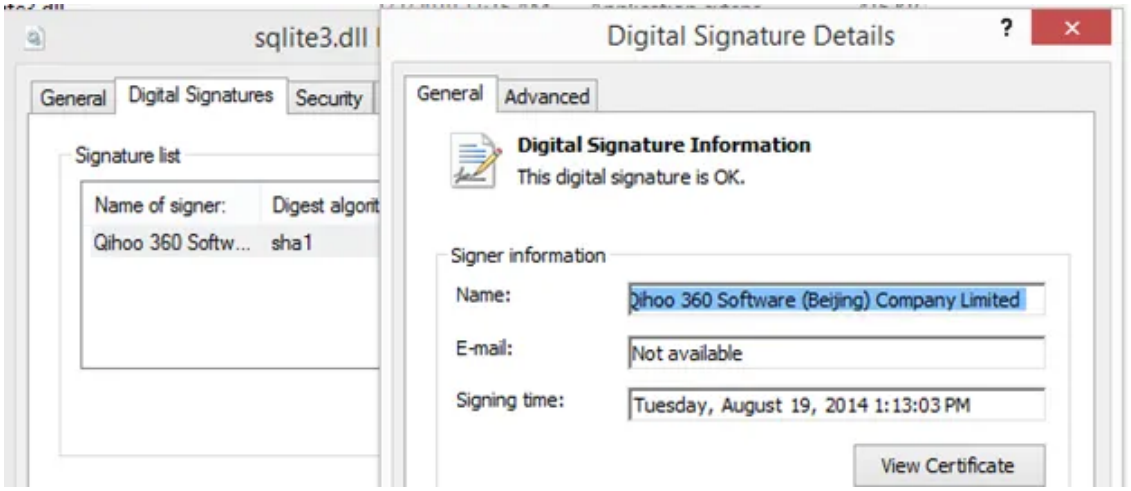
· RasTls.dll (signed by *Avira* — not valid cert).

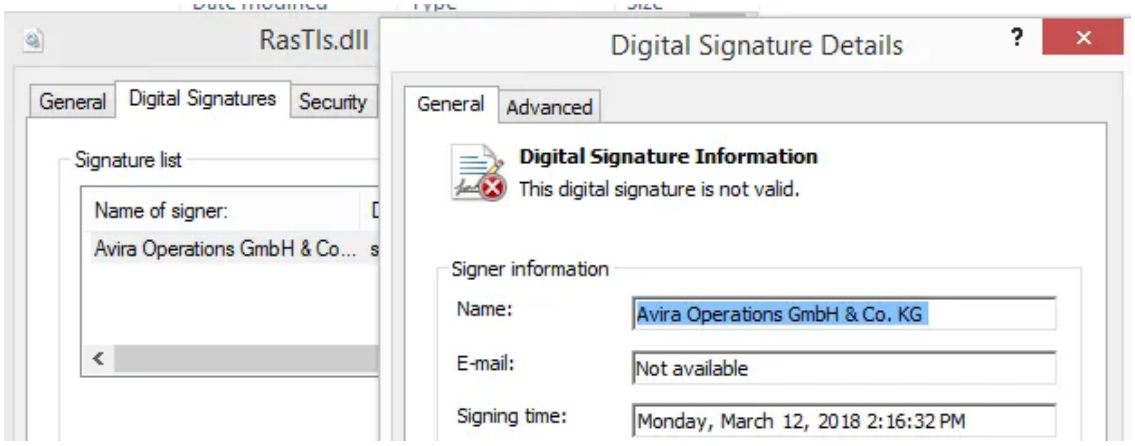




Thông tin Digital Signatures của các files:

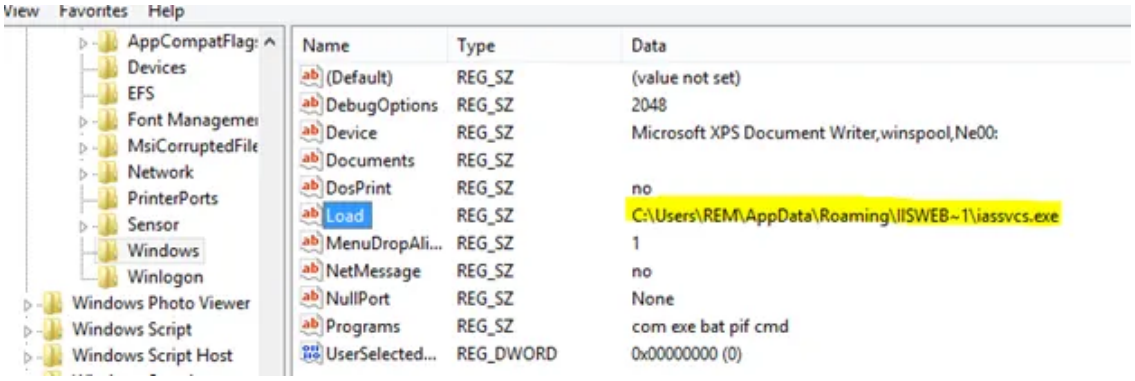




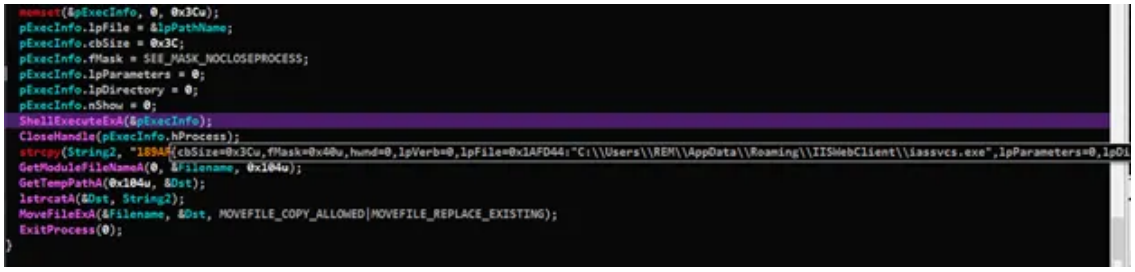


Tạo persistence key để tự động chạy tại **“Software\Microsoft\windows NT\CurrentVersion\windows”:**

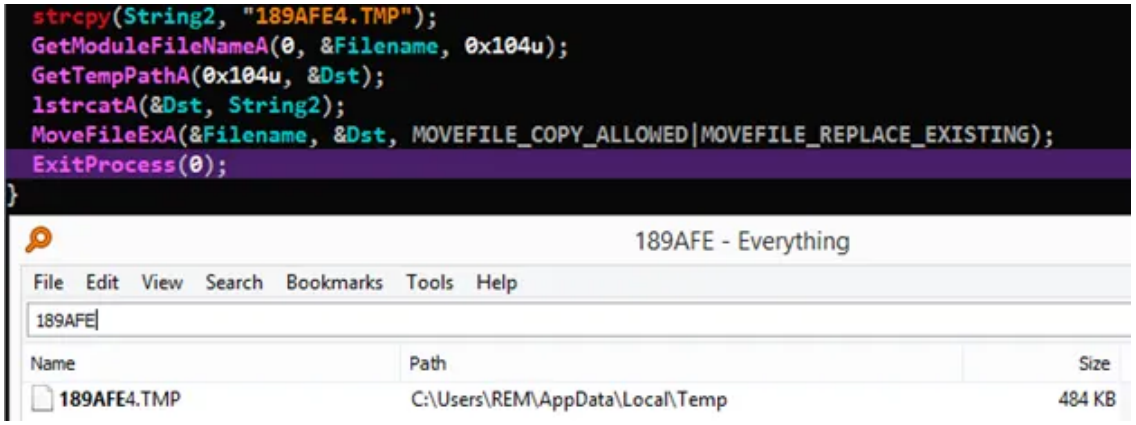
```
phkResult = 0;
result_1 = 0;
if ( RegOpenKeyExA(
       HKEY_CURRENT_USER,
       "Software\\Microsoft\\windows NT\\CurrentVersion\\windows",
       0,
       KEY_WRITE,
       &phkResult) )
{
  GetLastError();
}
else
{
  ValueName = *(_DWORD *)"Load";
  v6 = szLoad[4];
  cbData = lstrlenA(lpData);
  if ( !RegSetValueExA(phkResult, (LPCSTR)&ValueName, 0, 1u, (const BYTE *)lpData, cbData) )
  {
    result_1 = 1;
  }
}
if ( phkResult )
{
  RegCloseKey(phkResult);
}
return result_1 != 0;
```



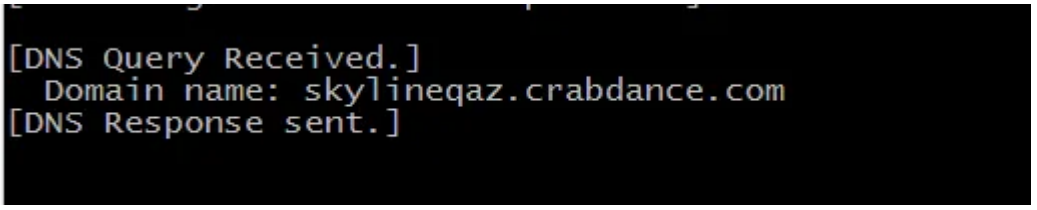
Sau khi tạo key trong Registry xong, thực thi file **iassvcs.exe**, file này sẽ load các đã drop cùng thư mục:



Binary cuối cùng được lưu thành file **189AFE4.TMP**:



Tiến trình **iassvcs.exe** sau khi thực thi sẽ kết nối tới C2 tại:



4. IOCs

· **Malicious RTF:** c580d77722d85238ed76689a17b0205b4d980c010bef9616b8611ffba21b142e

· **Decrypted binary:** 8D7425AE30FD2D5196EC4DCD2540B31A0D26772F

· **Dropped binary:**

o %appdata%\IISWebClient\iassvcs.exe: 62944E26B36B1DCACE429AE26BA66164

o %appdata%\IISWebClient\sqlite3.dll: FEE0B982AF421FF8C16C0187B376B086

o %appdata%\IISWebClient\RasTls.dll: C6A73E29C770065B4911EF46285D6557

· **C2:**

o Name: skylineqaz[.]crabdance[.]com

o Name: xn — ylineqaz-y25ja[.]crabdance[.]com

· **Registry:**

o “HKCU\Software\Microsoft\windows NT\CurrentVersion\windows”; Value name “Load”; Data: C:\Users\{username}\AppData\Roaming\IISWEB~1\iassvcs.exe

o “HKEY_CURRENT_USER\Software\Direct3D”
