Analysis of MS Word to drop Remcos RAT

muha2xmad.github.io/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 fromat 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 eddition 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.

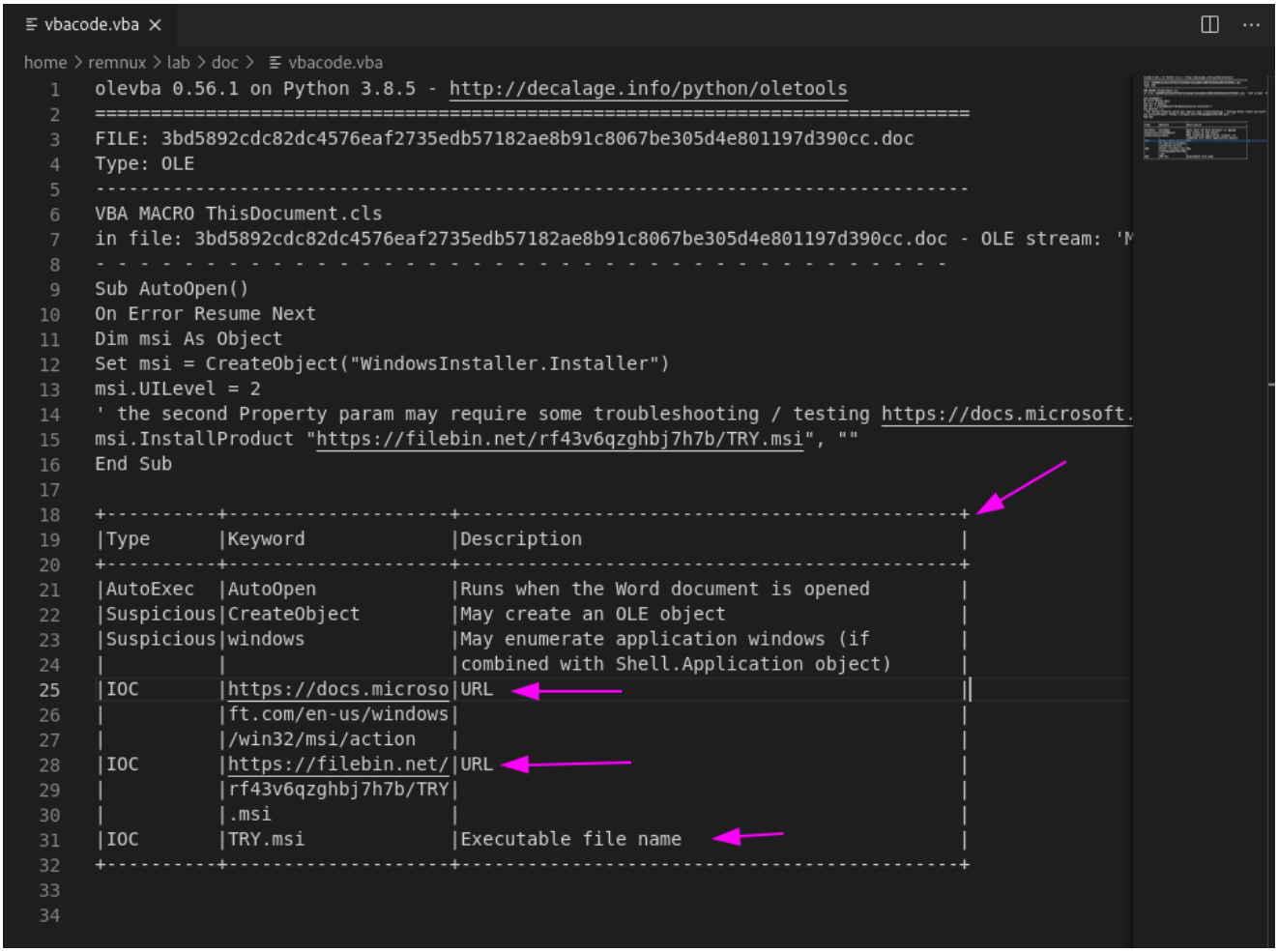
```
remnux@remnux:~/lab/doc$ oledump.py 3bd5892cdc82dc4576eaf2735edb57182ae8b91c8067be305d4e801197d390cc
.doc
  1:       114 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:      4096 '\x05SummaryInformation'
  4:      7133 '1Table'
  5:     32978 'Data'
  6:       367 'Macros/PROJECT'
  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()`.



Figure(3): Extraction of the VBA code

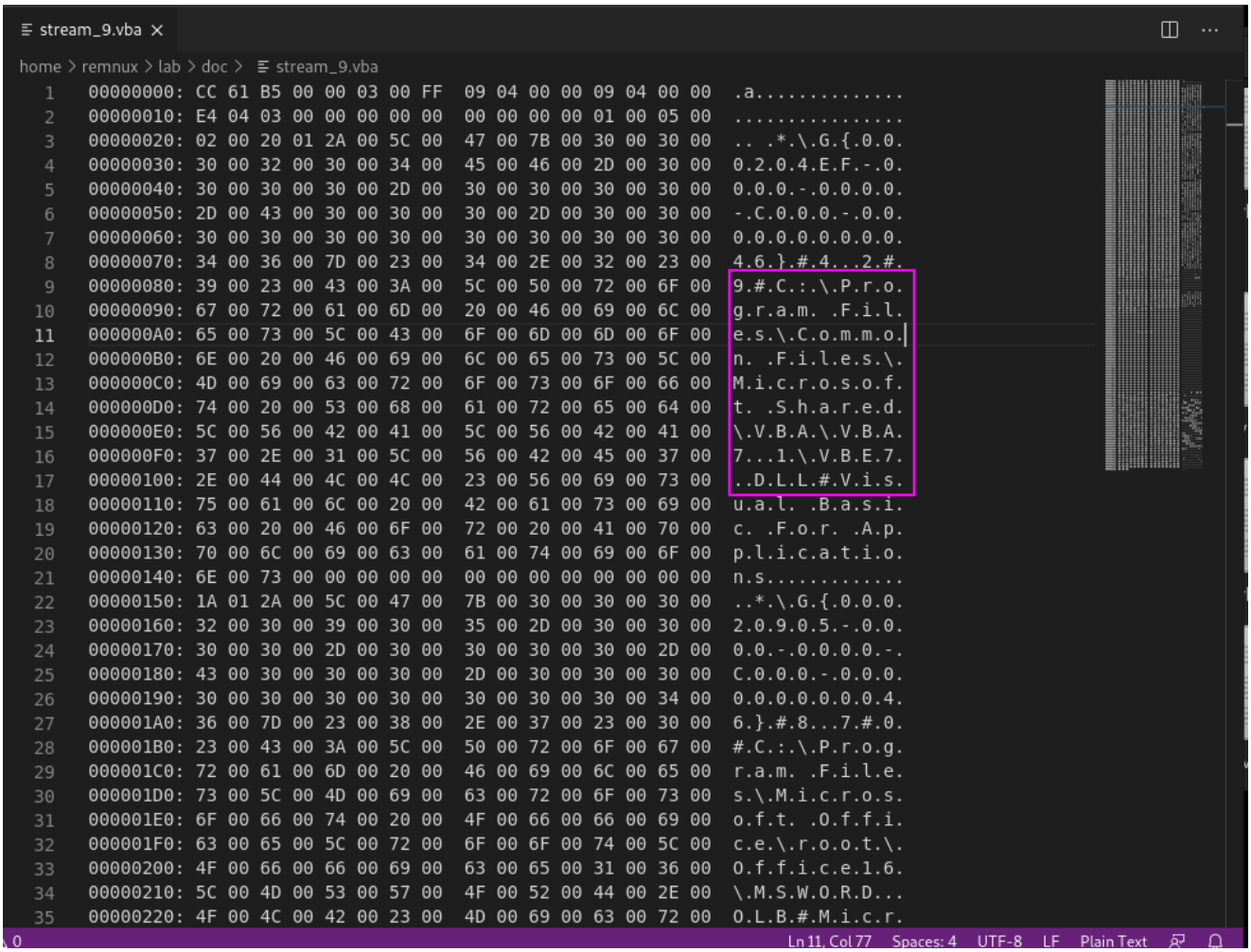
But this is not enough. We will try to extract much info about the Doc by using `oledump.py` and extract the content of all the streams but if you want to short your time extract only the streams `9` and `10`.

```
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.

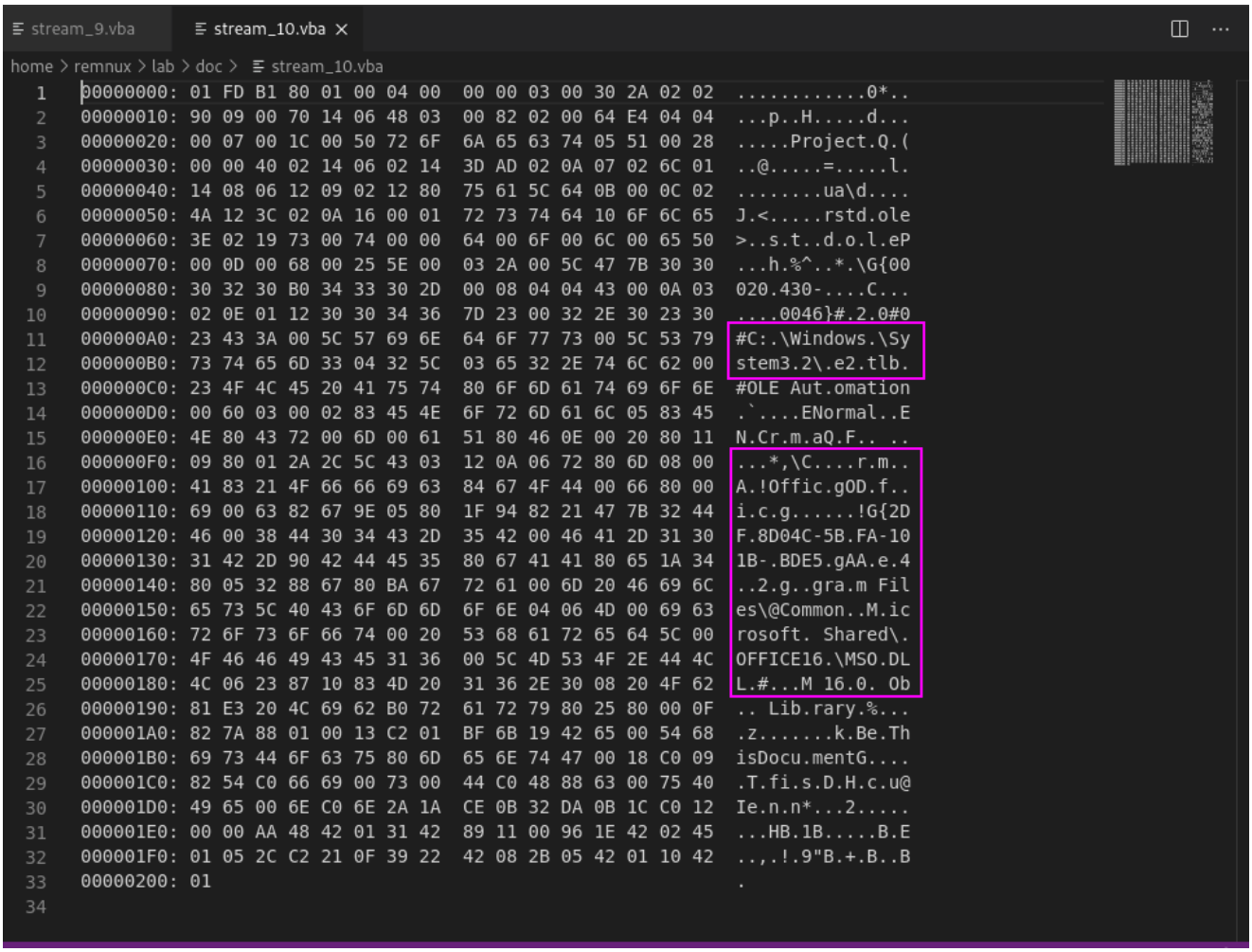
```
C:\Program files\Common files\Microsoft shared\VBA\VBA7.1\VBE7.dll
C:\Windows\System32\stdole2.tlb
C:\Program files\Microsoft Office\root\Office1.6\MSWORD
ObjectLibrary
C:\Program files\Common files\Microsoft shared\OFFICE16\MSO.DLL
autoOpen
CreateObject
InstallProduct
```



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

And in stream `10` which has less IoCs than stream `9`.

```
C:.\Windows.\System32\e2tlb
C:\Program files\Common files\Microsoft shared\OFFICE1.6\MSO.DLL
```



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