

First, run oledump on the file to reveal all streams of data:

```
alpine:/home/ogre/Downloads# python ../geany-prjs/D0cum3n7s.py
mlsc{B4$IX_M@CmR0}
alpine:/home/ogre/Downloads# python ../DidierStevensSuite/oledump.py totallySafeDoc.docm
A: word/vbaProject.bin
A1:      420 'PROJECT'
A2:      71 'PROJECTwm'
A3: M    1836 'VBA/NewMacros'
A4: m    1082 'VBA/ThisDocument'
A5:      2671 'VBA/_VBA_PROJECT'
A6:     1462 'VBA/_SRP_0'
A7:      122 'VBA/_SRP_1'
A8:      169 'VBA/_SRP_2'
A9:      156 'VBA/_SRP_3'
A10:     568 'VBA/dir'
alpine:/home/ogre/Downloads#
```

A3 looks suspicious. It is a VB macro and its name seems fishy. Let's decompress this script and see what it has inside:

```
alpine:/home/ogre/Downloads# python ../DidierStevensSuite/oledump.py -s 3 --vbadecompressskipattribute totallySafeDoc.docm
Sub flag()
    Dim fl As String
    Dim flag As String
    fl = "ZB(,+-?FGHJ[Ksl[];'.\A1SD340XCvWEmRTYU#s%~*6*IOPl2NM5c67}890-=/!@\"
    flag = (Left(Mid(fl, 35), 1) + Left(Mid(fl, 24), 1) + Left(Mid(fl, 15), 1) + Left(Mid(fl, 54), 1) + Left(Mid(fl, 13), 1) + Left(Mid(fl, 2), 1) + Left(Mid(fl, 28), 1) + Left(Mid(fl, 25),
1) + Left(Mid(fl, 46), 1) + Left(Mid(fl, 31), 1) + Left(Mid(fl, 5), 1) + Left(Mid(fl, 52), 1) + Left(Mid(fl, 65), 1) + Left(Mid(fl, 31), 1) + Left(Mid(fl, 36), 1) + Left(Mid(fl, 60), 1) + Le
ft(Mid(fl, 57), 1))
End Sub
```

```
D0cum3n7s.vb x D0cum3n7s.py x
1 Module VBAModule
2
3     Sub main()
4         Dim fl As String
5         Dim flag As String
6         fl = "ZB() +: ?FGHJ{KsL[]; ',.AlSD34QXCvWEmRTYU#$$%^&*IOP12NM5c67}890-=/!@"
7         flag = (Left(Mid(fl, 35), 1) +
8             Left(Mid(fl, 24), 1) +
9             Left(Mid(fl, 15), 1) +
10            Left(Mid(fl, 54), 1) +
11            Left(Mid(fl, 13), 1) +
12            Left(Mid(fl, 2), 1) +
13            Left(Mid(fl, 28), 1) +
14            Left(Mid(fl, 25), 1) +
15            Left(Mid(fl, 46), 1) +
16            Left(Mid(fl, 31), 1) +
17            Left(Mid(fl, 5), 1) +
18            Left(Mid(fl, 52), 1) +
19            Left(Mid(fl, 65), 1) +
20            Left(Mid(fl, 31), 1) +
21            Left(Mid(fl, 36), 1) +
22            Left(Mid(fl, 60), 1) +
23            Left(Mid(fl, 57), 1))
24     End Sub
25 End Module
26
```

There's a function (Sub) called flag which computes the flag based on the fl variable. Now, we can recreate this VB code in Python as below:

```
D0cum3n7s.vb x D0cum3n7s.py x
1 fl = "ZB()_+:?FGHJ{KsL[];','.,A\SD34QXCWEmRTYU#$$^&*IOP12NM5c67}890-=/!@"
2 flag = ""
3
4
5 flag = fl[34] + fl[23] + fl[14] + fl[53] + fl[12] + fl[1] + fl[27] + \
6       fl[24] + fl[45] + fl[29] + fl[4] + fl[51] + fl[64] + fl[30] + fl[34] + \
7       fl[35] + fl[59] + fl[56]
8
9
10 print(flag)
11
12
13
```

On a side note, VB can be zero and one indexed (it does not make sense, right?). And, Mid and Left functions are one-indexed. Thereby, those indexes in the Python code have been decremented by one. When we run the Python code, we will get the flag:

```
alpine:/home/ogre/Downloads# python ../geany-prjs/D0cum3n7s.py  
mlsc{B4SIX_M@CmR0}  
alpine:/home/ogre/Downloads#
```