MrPhisher

> **Instructions**
> I received a suspicious email with a very weird-looking attachment. It keeps on asking me to "enable macros". What are those? Access this challenge by deploying the machine attached to this task by pressing the green "Start Machine" button. The files you need are located in **/home/ubuntu/mrphisher** on the VM.

I transferred those files from the Target Machine to my own local:

```
┌──(kali㉿kali)-[~/TryHackMe/MrPhisher]
└─$ ls -l
total 128
-rwxrwxrwx 1 kali kali 63346 Jan 29  2022 MrPhisher.docm
-rw-r--r-- 1 kali kali 58947 Mar 11  2022 mr-phisher.zip
```

The `MrPhiser.docm` is an Office File → `oletools` could help to analyze this file

Installation:

```
┌──(kali㉿kali)-[~/TryHackMe/MrPhisher]
└─$ pip3 install -U oletools
Defaulting to user installation because normal site-packages is not writeable
[REDACTED...]
Installing collected packages: easygui, pyparsing, msoffcrypto-tool, colorclass, pcodedmp,
oletools
  WARNING: The script msoffcrypto-tool is installed in '/home/kali/.local/bin' which is no
t on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use -
-no-warn-script-location.
  WARNING: The script pcodedmp is installed in '/home/kali/.local/bin' which is not on PAT
H.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use -
-no-warn-script-location.
```

```
  WARNING: The scripts ezhexviewer, ftguess, mraptor, msodde, olebrowse, oledir, olefile,
 oleid, olemap, olemeta, oleobj, oletimes, olevba, pyxswf and rtfobj are installed in '/ho
me/kali/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use -
-no-warn-script-location.
Successfully installed colorclass-2.2.2 easygui-0.98.3 msoffcrypto-tool-5.1.1 oletools-0.6
0.1 pcodedmp-1.2.6 pyparsing-2.4.7
```

If you see the `WARNING` message, read it and notice on the directory that contains the tools: `/.local/bin`

Use the module `olevba` to extract and analyze VBA Macro source code from MS Office documents (OLE and OpenXML).

```
┌──(kali㉿kali)-[~/TryHackMe/MrPhisher]
└─$ /home/kali/.local/bin/olevba MrPhisher.docm -c
olevba 0.60.1 on Python 3.11.2 - http://decalage.info/python/oletools
===============================================================================
FILE: MrPhisher.docm
Type: OpenXML
WARNING  For now, VBA stomping cannot be detected for files in memory
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin - OLE stream: 'VBA/ThisDocument'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(empty macro)
-------------------------------------------------------------------------------
VBA MACRO NewMacros.bas
in file: word/vbaProject.bin - OLE stream: 'VBA/NewMacros'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Sub Format()
Dim a()
Dim b As String
a = Array(102, 109, 99, 100, 127, 100, 53, 62, 105, 57, 61, 106, 62, 62, 55, 110, 113, 11
4, 118, 39, 36, 118, 47, 35, 32, 125, 34, 46, 46, 124, 43, 124, 25, 71, 26, 71, 21, 88)
For i = 0 To UBound(a)
b = b & Chr(a(i) Xor i)
Next
End Sub
```

- `Dim a()` and `Dim b As String`: declare the variable `a()` and `b`
- `UBound(a)`: returns the index of the last element.
- `b & Chr(a(i) Xor i)`: concatenate `b` with the result of `Chr()` function from the `Xor` operation

The above **VBA** code could be simply transformed into **Python** like this:

```
a = [102, 109, 99, 100, 127, 100, 53, 62, 105, 57, 61, 106, 62, 62, 55, 110, 113, 114, 11
8, 39, 36, 118, 47, 35, 32, 125, 34, 46, 46, 124, 43, 124, 25, 71, 26, 71, 21, 88]
b = ''
for i in range(len(a)):
  b = b + chr(a[i] ^ i)

print(b)
```

Run this code and get the flag:

```
┌──(kali㉿kali)-[~/TryHackMe/MrPhisher]
└─$ python3 run.py
flag{a39a07a239aacd40c948d852a5c9f8d1}
```