

# **Excel 4.0 Macros [Let's Defend – Write-up]**

One of the employees has received a suspicious document attached in the email. When the e-mail flow is examined, it is seen that there is a suspicious Office file. Employees forward the email to the security team for analysis.

When L1 analysts scan the suspicious file with several different scanning tools, they see that it does not contain VBA macros. Since the file format is similar to phishing, they forwarded the suspicious Office file to you for detailed analysis.

\*\* Since the 2nd payload download addresses are closed, the 2nd payload is in the zip. Please start your analysis from the Office file.

## | Start investigation

It is important to note that olevba is specifically designed to scan VBA streams, meaning it will not detect XLM-based macros. These types of macros are often concealed within spreadsheets using deceptive cell names like Auto\_Open or other renamed cells intended to trigger execution.

To handle these cases, you need a specialized tool called **XLMMacroDeobfuscator**.

## Installation & Execution Guide

Installation: If the tool is missing, use the following command to force an install: pip install XLMMacroDeobfuscator --force

Deobfuscation: To analyze a specific Excel file (e.g., research-1646684671.xls), run: **xlmdeobfuscator --file research-1646684671.xls**

```
root@ip-172-31-6-133:~/Desktop/ChallengeFiles/l1f44531fb888d31307d87b01e8abff# xmdeobfuscator --file research-1646684671.xls
XLMMacroDeobfuscator: pywin32 is not installed (only is required if you want to use MS Excel)

[Unrecognized file format]
[Unencrypted xls file]

[[Loading Cells]]
Error [deobfuscator.py:3195 process_file(**vars(args))]

root@ip-172-31-6-133:~/Desktop/ChallengeFiles/l1f44531fb888d31307d87b01e8abff#
```

## Fixing Assertion Error in deobfuscator.py

If you encounter an error at deobfuscator.py:3195 while processing a file, it is likely due to a version mismatch in the formula parser. To resolve this, you need to manually patch the formula.py file within the package.

### Step 1: Locate the formula.py file

```
Processing triggers for man-db (2.9.1-1) ...
root@ip-172-31-6-133:~/Desktop/ChallengeFiles/11f44531fb088d31307d87b01e8eabff# locate formula.py
/usr/local/lib/python3.8/dist-packages/pvxlbsb2/formula.py
/usr/local/lib/python3.8/dist-packages/xlrd2/formula.py
root@ip-172-31-6-133:~/Desktop/ChallengeFiles/11f44531fb088d31307d87b01e8eabff#
```

### Editing formula.py using Vim

To apply the fix, follow these steps to locate and modify the function within the xlrd2 library:

#### 1. Open the file with Root Privileges

Run the following command to open the specific file in the Vim editor:

```
sudo vim /usr/local/lib/python3.8/dist-packages/xlrd2/formula.py
```

#### 2. Search for the dump\_formula Function

Once the file is open in Vim, use the search feature to jump to the correct line:

**Start Search:** Press / then type dump\_formula and hit Enter.

**Navigate Matches:** \* Press n to jump to the next match.

Press N to go back to the previous match.

#### 3. Modify the Line

After locating assert bv >= 80, press i to enter Insert Mode, change the value to 70, then press Esc followed by :wq to save and exit

```
#### under deconstruction ####
def dump_formula(bk, data, fmlalen, bv, reldelta, blah=0, isname=0):
    if blah:
        print("dump_formula", fmlalen, bv, len(data), file=bk.logfile)
        hex_char_dump(data, 0, fmlalen, fout=bk.logfile)
    assert bv >= 70 ##### this function needs updating #####
    sztab = szdict[bv]
    pos = 0
    stack = []
    any_rel = 0
    any_err = 0
```

After fixing the error i run this again;

xlmdeobfuscator — file research-1646684671.xls

**Q1: Attackers use a function to make the malicious VBA macros they have prepared run when the document is opened. What do attackers change the cell name to to make Excel 4.0 macros work to provide the same functionality?**

**Answer:** auto\_open

**Q2: What is the address of the first cell where Excel 4.0 macros will run in the malicious Office document you are analyzing? (Example: {doc1!ab3})**

**Answer:** doc4!ba7

**Q3: Which function is used to start a process in the operating system in the document you are analyzing?**

**Answer:** exec

**Q4: Which LOLBAS tool was used in the Excel 4.0 macros you analyzed? (Format: {xxxx.exe})**

**LOLBAS (Living-Off-the-Land Binaries-And-Scripts) is a known technique hackers can use to stay under the radar by utilizing legitimate tools for malicious activities.**

```
*"""\.\iroto.dll""***,Doc3AW14)"  
CELL:BG21 , FullEvaluation , FORMULA("=EXEC("regsvr32 -s ""&""..\iroto.dll"),",Doc3AW15)  
CELL:BG22 , FullEvaluation , BD12()  
CELL:BD15 , PartialEvaluation , =B74526348672131073295890118132856725104642837213024234875168688591210916395323297595260510582865920==ACOS(78  
709969853406929708102759735505844224)==ACOSH(8769769797886054592382096123330822144)==B7452634867213107329589011813285672510464283721302423487516  
2329759526051058265920=(879475612348767955979096853406929782176257973555684424)==ACOSH(8769769797886054592382096123330822144)-874526348  
8576521046428372130242348751686885912109163953232975562051058265920==ACOS(78947561234875795599709968534069297821072579735555084424)==ACOSH(87  
382096123330822144)==B7452634867213107329589011813285672510464283721302423487516868859121091639532329759526051058265920==ACOS(7894756123487579559  
67259735550844224)==ACOSH(8769769797886054592382096123330822144)==B74526348672131073295890118132856725104642837213024234875168688591210916395323297595  
260510582865920=
```

**Answer:** regsvr32.exe

### **Q5: What is the name of the registered DLL?**

```
260510582865920==AC05(78947561234876795599769969853406929702810725797355505844224)==AC05(87697697978978605454923820961  
&"""\..iroto.dll""",Doc3AW14)  
CELL:BG21 , FullEvaluation , FORMULA("=EXEC("""regsvr32 -s """&"\" .\irotol.dll """)",Doc3AW15)  
CELL:BG22 , FullEvaluation , BD12()  
CELL:BD15 , PartialEvaluation , "8745263486721310732958901181328567251046428327213024234875168688591210916395932
```

**Answer:** iroto.dll

**Q6: What is the username that made the last change to the malicious document?**

```
Processing triggers for libtiff-bin (4.3.1-0ubuntu0.13) ...
root@ip-172-31-6-133:~/Desktop/ChallengeFiles/11f44531fb088d31307d87b01e8eabff# exiftool research-1646684671.xls
+ ifTool Version Number          : 11.88
+ file Name                   : research-1646684671.xls
Directory                         :
File Size                          : 648 kB
File Modification Date/Time       : 2021:06:13 09:24:55+00:00
File Access Date/Time             : 2026:01:31 04:09:11+00:00
File Inode Change Date/Time      : 2026:01:31 04:35:29+00:00
File Permissions                 : rw-rw-r--
File Type                         : XLS
File Type Extension              : xls
MIME Type                         : application/vnd.ms-excel
Author                            :
Last Modified By                 : Amanda
Software                          : Microsoft Excel
Create Date                       : 2015:06:05 18:17:20
Modify Date                        : 2021:06:13 09:24:55
Security                           : None
Code Page                         : Windows Latin 1 (Western European)
Company                           :
Scale Crop                        : No
Links Up To Date                 : No
Title Of Parts                   : Doc1, Doc2, Doc3, Doc4
Heading Pairs                     : Worksheets, 1, Excel 4.0 Macros, 3
root@ip-172-31-6-133:~/Desktop/ChallengeFiles/11f44531fb088d31307d87b01e8eabff#
```

**Answer:** amanda

## **Summary**

While there are always deeper layers to explore when analyzing macro behavior and execution flows, this walkthrough has covered the essential steps for effective detection and deobfuscation.