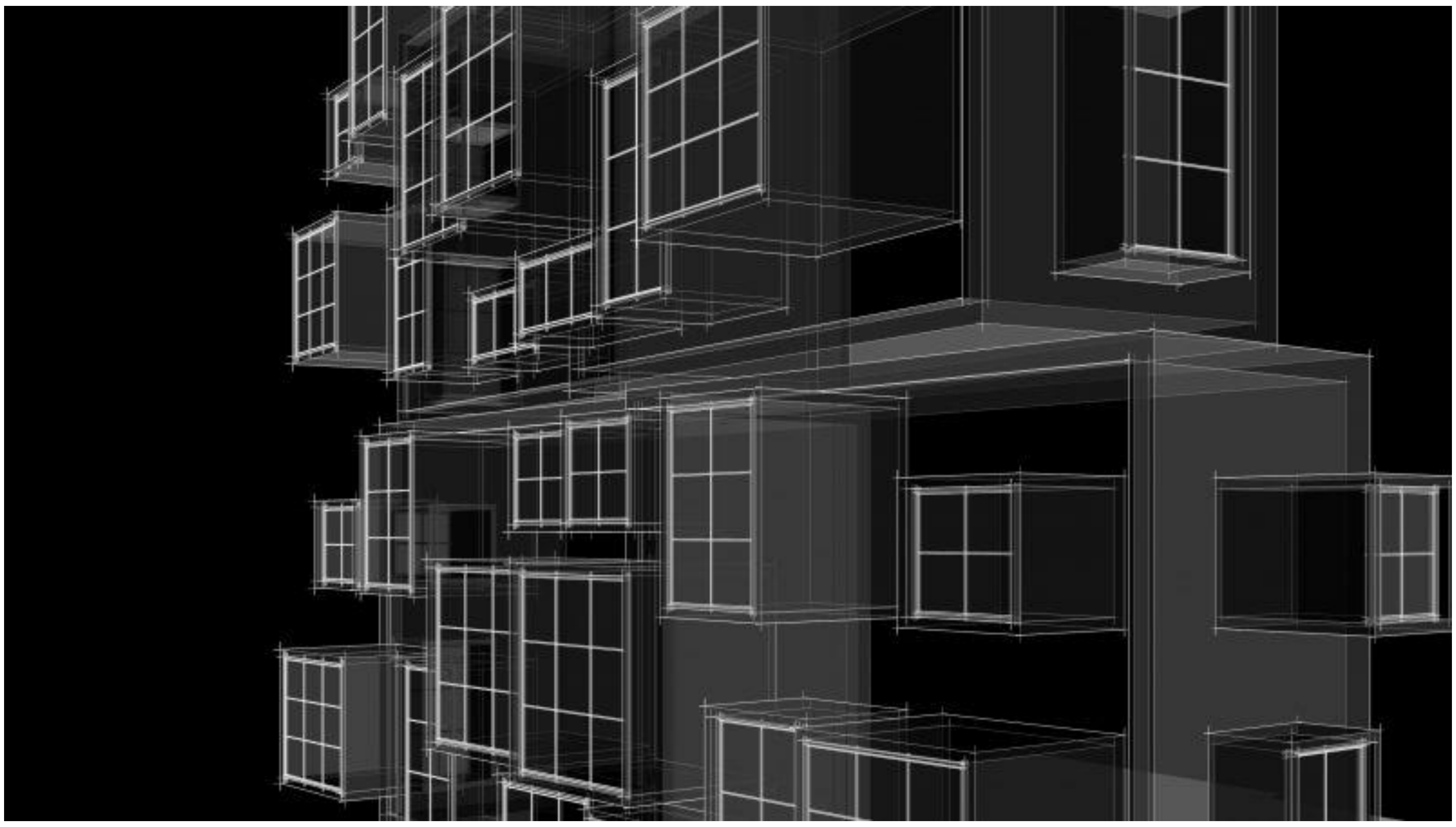


Malware descriptions

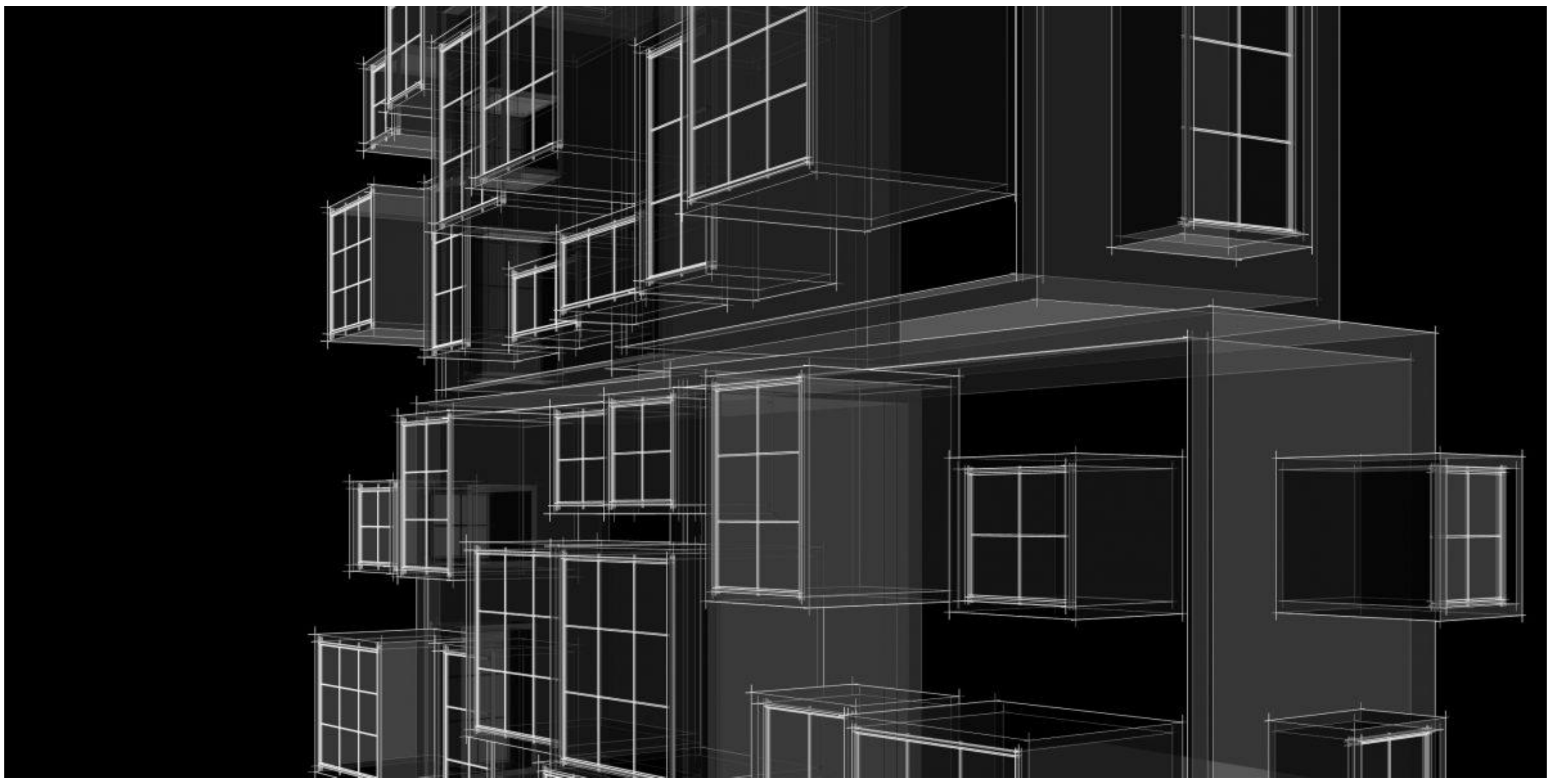
Emotet modules and recent attacks

Malware descriptions

13 Apr 2022

7 minute read

Table of Contents



Authors

- Expert AMR

Anti-Malware Research

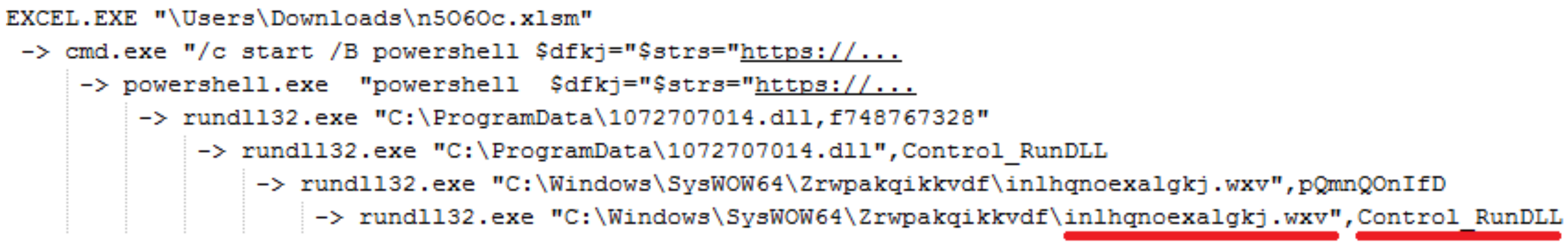
Emotet was first found in the wild in 2014. Back then its main functionality was stealing user banking credentials. Since then it has survived numerous transformations, started delivering other malware and finally became a powerful botnet. In January 2021 Emotet was disrupted by a joint effort of different countries' authorities. It took the threat actors almost 10 months to rebuild the infrastructure, whereupon Emotet returned in November. At that time, Trickbot malware was used to deliver Emotet. Now, Emotet is spreading by itself in malicious spam campaigns.

Based on recent Emotet protocol analysis and C2 responses, we can say that now Emotet can download 16 additional modules. We were able to retrieve 10 of them (including two different copies of the Spam module), used by Emotet for Credential/Password/Account/E-mail stealing and spamming. In this post, we provide a brief analysis of these modules, as well as statistics on recent Emotet attacks.

Emotet technical analysis

Infection chain

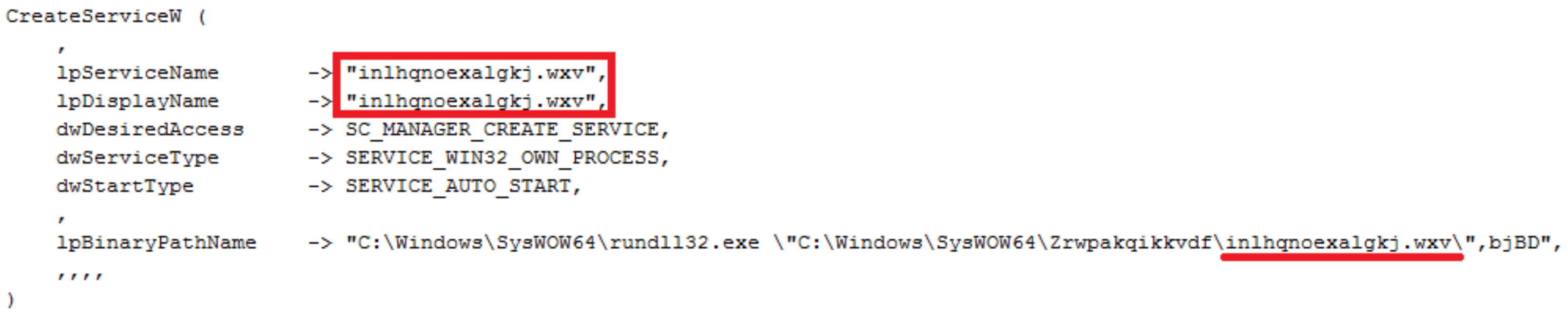
A typical Emotet infection begins with spam e-mails delivered with Microsoft Office (Word, Excel) attachments. Malicious macros are used to start PowerShell, and download and execute an Emotet DLL. Depending on the available access, Emotet creates a subdirectory with a random name in the %Windows%\SysWOW64\ or %User%\AppData\Local\ directory, and copies itself there under a randomly generated name and extension. The exported Control_RunDLL function is used to run the main activity of the Emotet DLL.



```
EXCEL.EXE "\Users\Downloads\n5060c.xlsm"
 -> cmd.exe "/c start /B powershell $dfkj="$strs="https://...
     -> powershell.exe  "powershell  $dfkj="$strs="https://...
         -> rundll32.exe "C:\ProgramData\1072707014.dll,f748767328"
             -> rundll32.exe "C:\ProgramData\1072707014.dll",Control_RunDLL
                 -> rundll32.exe "C:\Windows\SysWOW64\Zrwpakqikkvdf\inlhqnoexalgkj.wxv",pQmnQOnIfD
                     -> rundll32.exe "C:\Windows\SysWOW64\Zrwpakqikkvdf\inlhqnoexalgkj.wxv",Control_RunDLL
```

Emotet infection execution chain

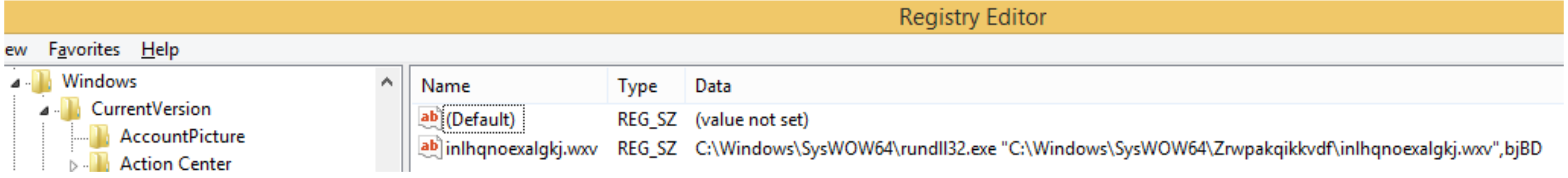
After being run, the Emotet malware creates a service by calling the CreateServiceW() function. A randomly generated name and extension, which were used to create a copy, act as service names.



```
CreateServiceW (
    ,
    lpServiceName       -> "inlhqnoexalgkj.wxv",
    lpDisplayName       -> "inlhqnoexalgkj.wxv",
    dwDesiredAccess     -> SC_MANAGER_CREATE_SERVICE,
    dwServiceType       -> SERVICE_WIN32_OWN_PROCESS,
    dwStartType         -> SERVICE_AUTO_START,
    ,
    lpBinaryPathName    -> "C:\Windows\SysWOW64\rundll32.exe \"C:\Windows\SysWOW64\Zrwpakqikkvdf\inlhqnoexalgkj.wxv\",bjBD",
    ,,,,
)
```

CreateServiceW() function with arguments

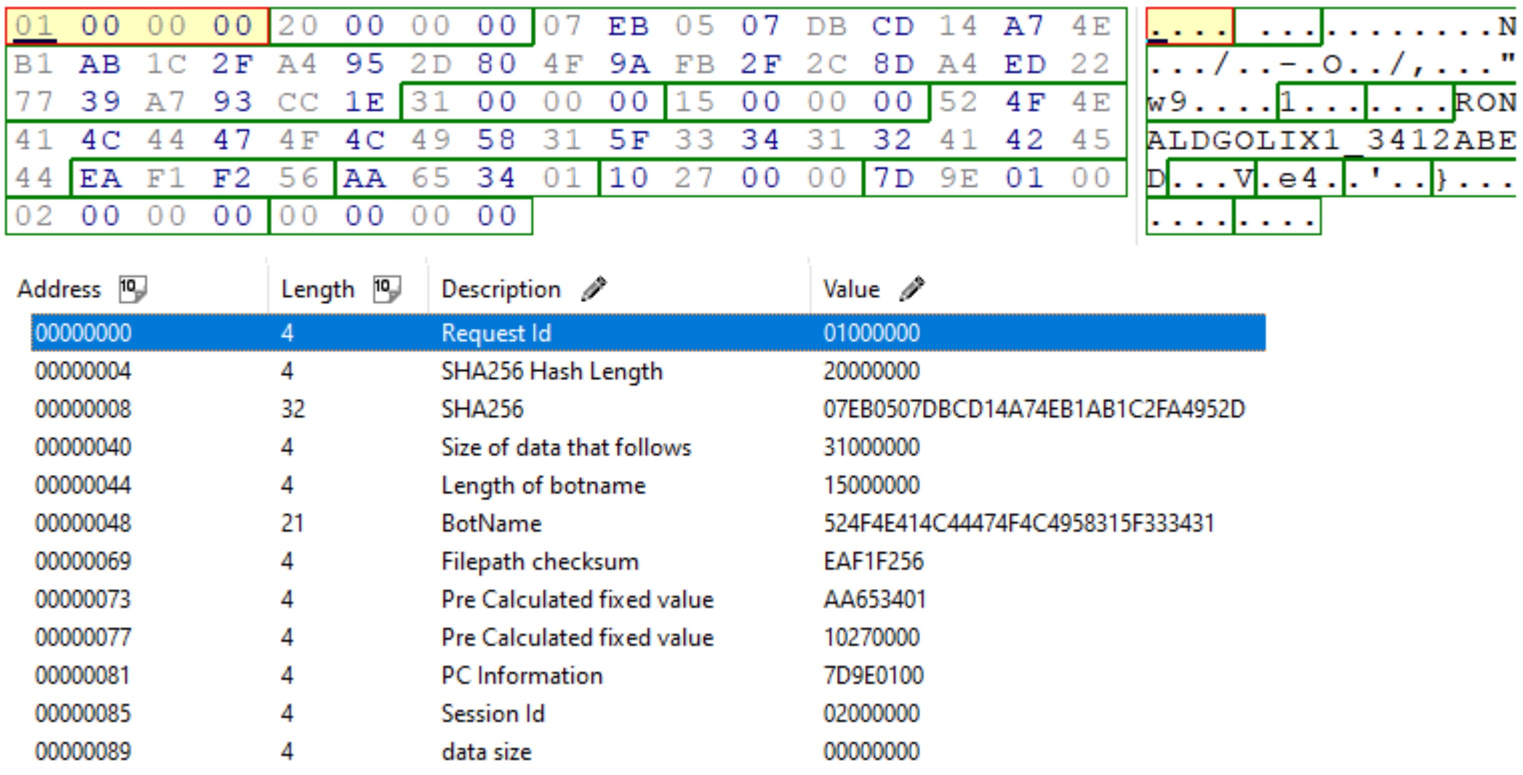
If the attempt to create a new service fails, Emotet creates a new registry key in HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run with the same names that were used when creating the service.



Name	Type	Data
(Default)	REG_SZ	(value not set)
inlhqnoexalgkj.wxv	REG_SZ	C:\Windows\SysWOW64\rundll32.exe "C:\Windows\SysWOW64\Zrwpakqikkvdf\inlhqnoexalgkj.wxv",bjBD

Autostart key in registry

As soon as the Emotet DLL is launched, it registers with one of the 20 C2 IPs that are hardcoded in encrypted form into the malware body. Downloaded modules can also include additional C2 IPs. The following data is used for bot registration:



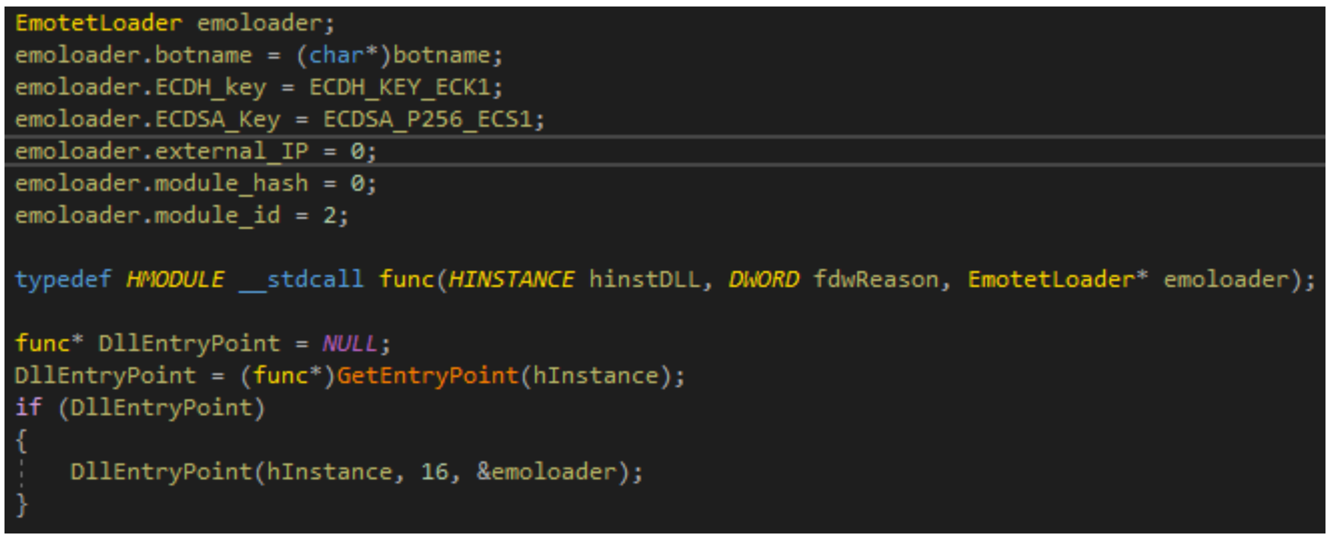
```
01 00 00 00 20 00 00 00 07 EB 05 07 DB CD 14 A7 4E   .... ...........N
B1 AB 1C 2F A4 95 2D 80 4F 9A FB 2F 2C 8D A4 ED 22   .../..-.O../,..."
77 39 A7 93 CC 1E 31 00 00 00 15 00 00 00 52 4F 4E   w9....1.......RON
41 4C 44 47 4F 4C 49 58 31 5F 33 34 31 32 41 42 45   ALDGOLIX1_3412ABE
44 EA F1 F2 56 AA 65 34 01 10 27 00 00 7D 9E 01 00   D...V.e4..'..}...
02 00 00 00 00 00 00 00                              ........
```

Address	Length	Description	Value
00000000	4	Request Id	01000000
00000004	4	SHA256 Hash Length	20000000
00000008	32	SHA256	07EB0507DBCD14A74EB1AB1C2FA4952D
00000040	4	Size of data that follows	31000000
00000044	4	Length of botname	15000000
00000048	21	BotName	524F4E414C44474F4C4958315F333431
00000069	4	Filepath checksum	EAF1F256
00000073	4	Pre Calculated fixed value	AA653401
00000077	4	Pre Calculated fixed value	10270000
00000081	4	PC Information	7D9E0100
00000085	4	Session Id	02000000
00000089	4	data size	00000000

Registration data

Together with the registration data, the victim's public key that is generated in every run is also sent to the C2. Unlike previous versions that used RSA to encrypt the generated AES key, this newest Emotet sample uses the ECDH (Elliptic curve Diffie–Hellman) algorithm, using the victim's generated key pair together with Emotet's public key hardcoded into the code to derive the AES key for encrypting the communication. This is done with use of the Windows API BCryptSecretAgreement.

During our monitoring we have observed that after registration the C2 replies with the Process List module payload. The module comes in the form of a DLL that is parsed and loaded directly into the Rundll32 process. Its entry point is called by passing a specific structure to its DllMain function. It is also worth noting that Emotet uses the ECDSA (Elliptic Curve Digital Signature Algorithm) to verify the payload integrity before loading it.



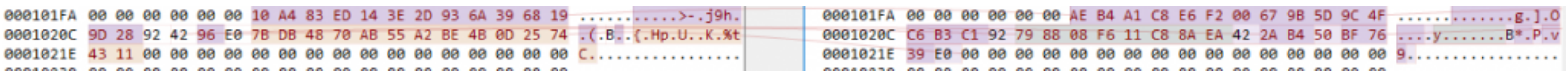
```
EmotetLoader emoloader;
emoloader.botname = (char*)botname;
emoloader.ECDH_key = ECDH_KEY_ECK1;
emoloader.ECDSA_Key = ECDSA_P256_ECS1;
emoloader.external_IP = 0;
emoloader.module_hash = 0;
emoloader.module_id = 2;

typedef HMODULE __stdcall func(HINSTANCE hinstDLL, DWORD fdwReason, EmotetLoader* emoloader);

func* DllEntryPoint = NULL;
DllEntryPoint = (func*)GetEntryPoint(hInstance);
if (DllEntryPoint)
{
    DllEntryPoint(hInstance, 16, &emoloader);
}
```

Pseudo code to load Emotet's second-stage DLL directly into memory

Aside from loading the DLL into memory, there are other ways to run the payload. For example:

- write the DLL payload to disk and run it through regsvr32.exe -s "%s" or rundll32.exe "%s",Control_RunDLL
- write the payload to disk and attempt to call CreateProcess or duplicate the user token to call CreateProcessAsUser

During communication, C2 returns the module bodies and configuration. Based on the configuration, the malware selects the way to run the payload module. During our research, all the modules we retrieved were launched in the parent process, but a separate thread is started for each new module. Each module has its own numeric ID, and contains its own C2 list. However, all the modules we retrieved contained the same list of C2, except the Spam module. Emotet modules are delivered on demand, and there are always a few junk bytes that vary in different samples of the same module. This is likely to avoid cloud scanning or file hash detection.



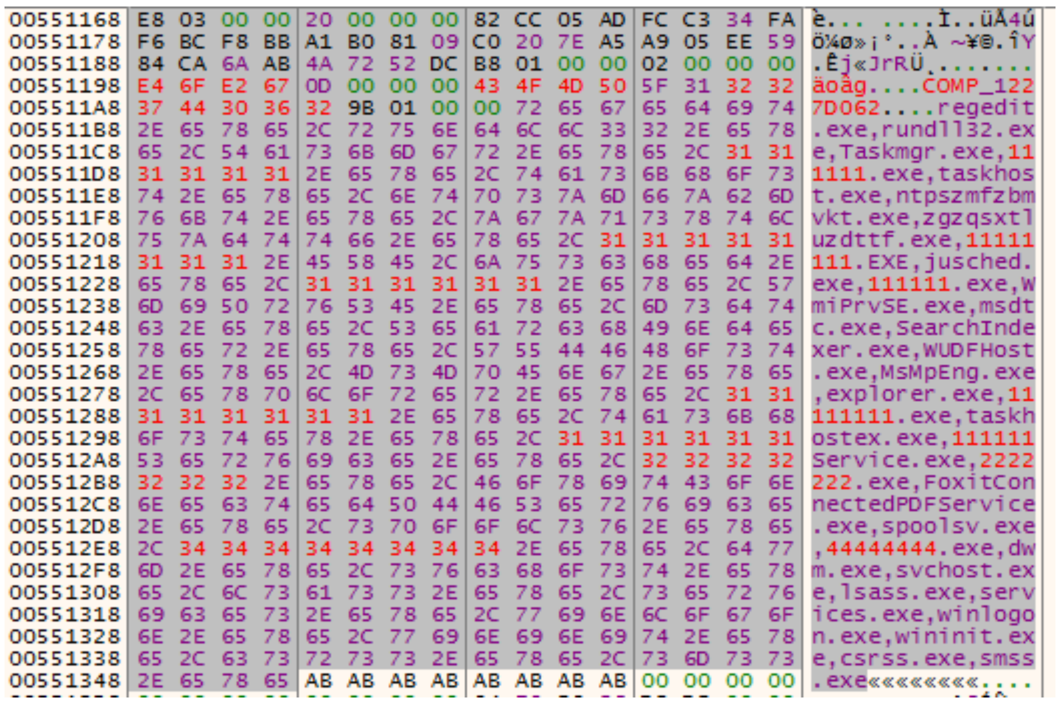
```
000101FA 00 00 00 00 00 00 10 A4 83 ED 14 3E 2D 93 6A 39 68 19 ............>-.j9h.
0001020C 9D 28 92 42 96 E0 7B DB 48 70 AB 55 A2 BE 4B 0D 25 74 .(.B..{.Hp.U..K.%t
0001021E 43 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C.................
```

```
000101FA 00 00 00 00 00 00 AE B4 A1 C8 E6 F2 00 67 9B 5D 9C 4F .............g.].O
0001020C C6 B3 C1 92 79 88 08 F6 11 C8 8A EA 42 2A B4 50 BF 76 ....y.......B*.P.v
0001021E 39 E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 9.................
```

Random bytes changed between "Process List" module binaries

Process List module

This module sends the list of running processes back to C2. Usually C2 does not send any other modules until it gets a response from this one.

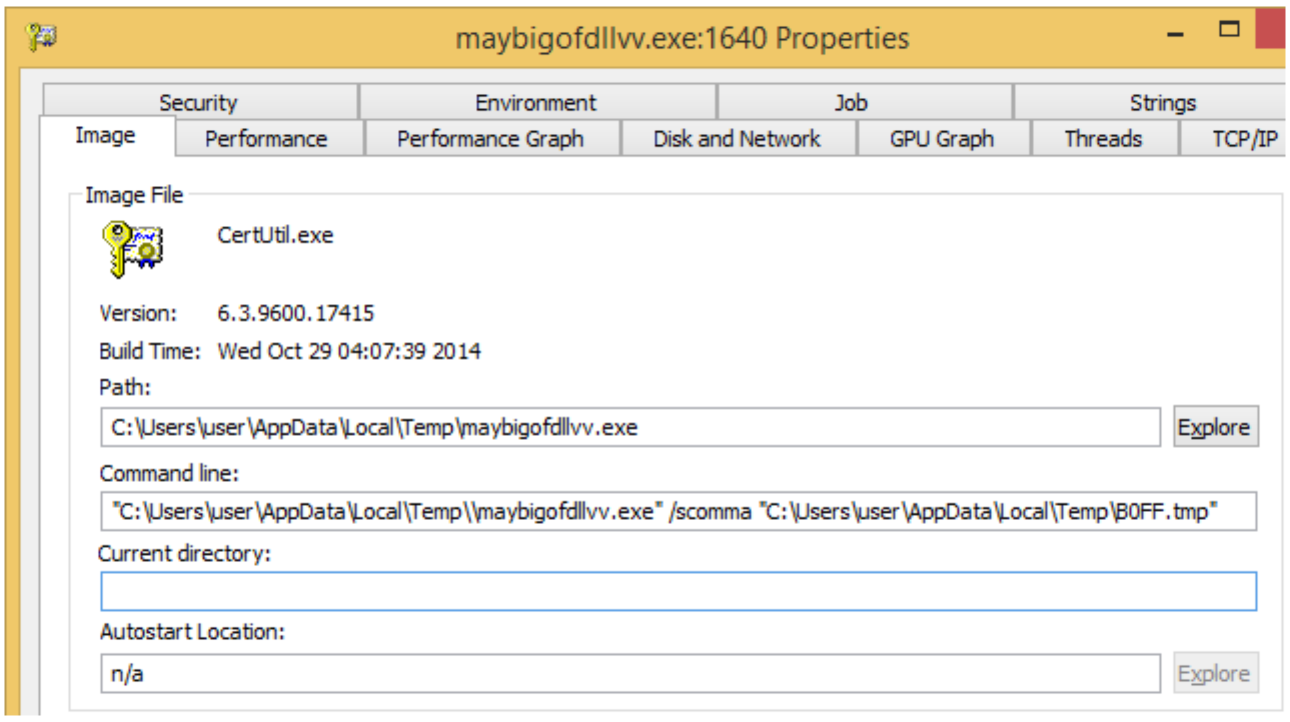


```
00551168 E8 03 00 00 20 00 00 00 82 CC 05 AD FC C3 34 FA è... ....Ì..üÃ4ú
00551178 F6 BC F8 BB A1 B0 81 09 C0 20 7E A5 A9 05 EE 59 ö¼ø»¡°..À ~¥©.îY
00551188 84 CA 6A AB 4A 72 52 DC B8 01 00 00 02 00 00 00 .Êj«JrRÜ¸.......
00551198 E4 6F E2 67 0D 00 00 00 43 4F 4D 50 5F 31 32 32 äoâg....COMP_122
005511A8 37 44 30 36 32 9B 01 00 00 72 65 67 65 64 69 74 7D062....regedit
005511B8 2E 65 78 65 2C 72 75 6E 64 6C 6C 33 32 2E 65 78 .exe,rundll32.ex
005511C8 65 2C 54 61 73 6B 6D 67 72 2E 65 78 65 2C 31 31 e,Taskmgr.exe,11
005511D8 31 31 31 31 2E 65 78 65 2C 74 61 73 6B 68 6F 73 1111.exe,taskhos
005511E8 74 2E 65 78 65 2C 6E 74 70 73 7A 6D 66 7A 62 6D t.exe,ntpszmfzbm
005511F8 76 6B 74 2E 65 78 65 2C 7A 67 7A 71 73 78 74 6C vkt.exe,zgzqsxtl
00551208 75 7A 64 74 74 66 2E 65 78 65 2C 31 31 31 31 31 uzdttf.exe,11111
00551218 31 31 31 2E 45 58 45 2C 6A 75 73 63 68 65 64 2E 111.EXE,jusched.
00551228 65 78 65 2C 31 31 31 31 31 31 2E 65 78 65 2C 57 exe,111111.exe,W
00551238 6D 69 50 72 76 53 45 2E 65 78 65 2C 6D 73 64 74 miPrvSE.exe,msdt
00551248 63 2E 65 78 65 2C 53 65 61 72 63 68 49 6E 64 65 c.exe,SearchInde
00551258 78 65 72 2E 65 78 65 2C 57 55 44 46 48 6F 73 74 xer.exe,WUDFHost
00551268 2E 65 78 65 2C 4D 73 4D 70 45 6E 67 2E 65 78 65 .exe,MsMpEng.exe
00551278 2C 65 78 70 6C 6F 72 65 72 2E 65 78 65 2C 31 31 ,explorer.exe,11
00551288 31 31 31 31 31 31 2E 65 78 65 2C 74 61 73 6B 68 111111.exe,taskh
00551298 6F 73 74 65 78 2E 65 78 65 2C 31 31 31 31 31 31 ostex.exe,111111
005512A8 53 65 72 76 69 63 65 2E 65 78 65 2C 32 32 32 32 Service.exe,2222
005512B8 32 32 32 2E 65 78 65 2C 46 6F 78 69 74 43 6F 6E 222.exe,FoxitCon
005512C8 6E 65 63 74 65 64 50 44 46 53 65 72 76 69 63 65 nectedPDFService
005512D8 2E 65 78 65 2C 73 70 6F 6F 6C 73 76 2E 65 78 65 .exe,spoolsv.exe
005512E8 2C 34 34 34 34 34 34 34 34 2E 65 78 65 2C 64 77 ,44444444.exe,dw
005512F8 6D 2E 65 78 65 2C 73 76 63 68 6F 73 74 2E 65 78 m.exe,svchost.ex
00551308 65 2C 6C 73 61 73 73 2E 65 78 65 2C 73 65 72 76 e,lsass.exe,serv
00551318 69 63 65 73 2E 65 78 65 2C 77 69 6E 6C 6F 67 6F ices.exe,winlogo
00551328 6E 2E 65 78 65 2C 77 69 6E 69 6E 69 74 2E 65 78 n.exe,wininit.ex
00551338 65 2C 63 73 72 73 73 2E 65 78 65 2C 73 6D 73 73 e,csrss.exe,smss
00551348 2E 65 78 65 AB AB AB AB AB AB AB AB 00 00 00 00 .exe««««««««....
```

Emotet Process List module request

Mail PassView module

The module contains an embedded executable called Nir Sofer's Mail PassView, a password recovery tool that reveals passwords and account details for various e-mail clients. In order to execute the password recovery tool, the Emotet module copies certutil.exe into a %Temp% directory under a random name with the .exe extension, starts the copied executable and uses the process hollowing technique to inject the password recovery tool executable into the newly created process. The CertUtil process is started with command line arguments to force the recovery tool to save the results to file.



maybigofdllvv.exe:1640 Properties

CertUtil.exe

Version: 6.3.9600.17415

Build Time: Wed Oct 29 04:07:39 2014

Path:

C:\Users\user\AppData\Local\Temp\maybigofdllvv.exe

Command line:

"C:\Users\user\AppData\Local\Temp\\maybigofdllvv.exe" /scomma "C:\Users\user\AppData\Local\Temp\B0FF.tmp"

Current directory:

Autostart Location:

n/a

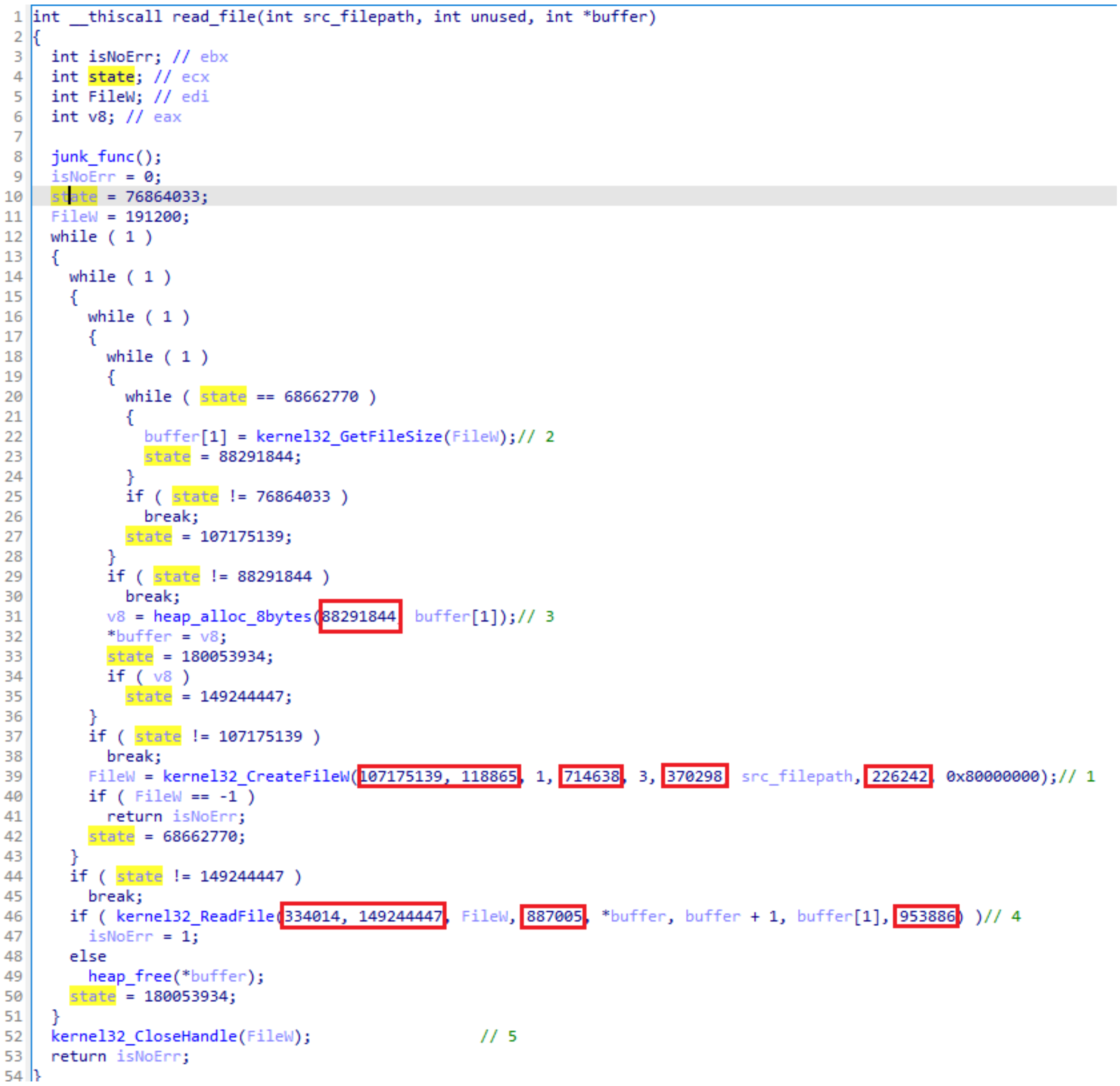
CertUtil with command line for password recovery tool

According to the official website, the utility is capable of revealing passwords and other account details for various e-mail clients, including Outlook and Thunderbird.

WebBrowser PassView module

This module is mostly the same as the previous one, except it uses the Nir Sofer's WebBrowser PassView password recovery tool for revealing passwords and account details in browsers.

According to the official website, the utility is capable of revealing passwords and other account details in various web browsers, including Internet Explorer, Mozilla Firefox, Google Chrome, Safari and Opera.



```
int __thiscall read_file(int src_filepath, int unused, int *buffer)
{
  int isNoErr; // ebx
  int state; // ecx
  int FileW; // edi
  int v8; // eax

  junk_func();
  isNoErr = 0;
  state = 76864033;
  FileW = 191200;
  while ( 1 )
  {
    while ( 1 )
    {
      while ( 1 )
      {
        while ( 1 )
        {
          while ( state == 68662770 )
          {
            buffer[1] = kernel32_GetFileSize(FileW);// 2
            state = 88291844;
          }
          if ( state != 76864033 )
            break;
          state = 107175139;
        }
        if ( state != 88291844 )
          break;
        v8 = heap_alloc_8bytes(88291844, buffer[1]);// 3
        *buffer = v8;
        state = 180053934;
        if ( v8 )
          state = 149244447;
      }
      if ( state != 107175139 )
        break;
      FileW = kernel32_CreateFileW(107175139, 118865, 1, 714638, 3, 370298, src_filepath, 226242, 0x80000000);// 1
      if ( FileW == -1 )
        return isNoErr;
      state = 68662770;
    }
    if ( state != 149244447 )
      break;
    if ( kernel32_ReadFile(334014, 149244447, FileW, 887005, *buffer, buffer + 1, buffer[1], 953886) )// 4
      isNoErr = 1;
    else
      heap_free(*buffer);
    state = 180053934;
  }
  kernel32_CloseHandle(FileW);                  // 5
  return isNoErr;
}
```

Pseudocode of function from WebBrowser PassView module

Emotet has used code obfuscation for years, and this module is no exception. In the figure above, we can see that the control flow obfuscation technique is used with the variable 'state' (yellow-colored). Also, all API calls are resolved during runtime. This is why this API resolution layer can use junk arguments (red-framed). Code listings can be larger and more obfuscated, which is why it makes no sense to show them for all modules.

Outlook Address Grabber module

A data exfiltration module for Outlook. The module uses the Outlook Messaging API interface, iterates through Outlook profiles and extracts all displayed names and mail addresses from each found mail. It then sends the collected e-mail addresses to C2.

Outlook E-mails Grabber module

A data exfiltration module for Outlook. The module uses the Outlook Messaging API interface, iterates through all personal folders (Inbox, Sent items, Deleted Items, etc), extracts all displayed names and mail addresses of sender and recipient, and extracts the e-mail subject and body. It then sends the collected e-mails to C2.

Thunderbird Address Grabber module

A data exfiltration module for Thunderbird. The module iterates through Thunderbird profiles located in %AppData%\Roaming\Thunderbird\Profiles\, parses Thunderbird data files and extracts displayed names and mail addresses. It then sends the collected e-mail addresses to C2.

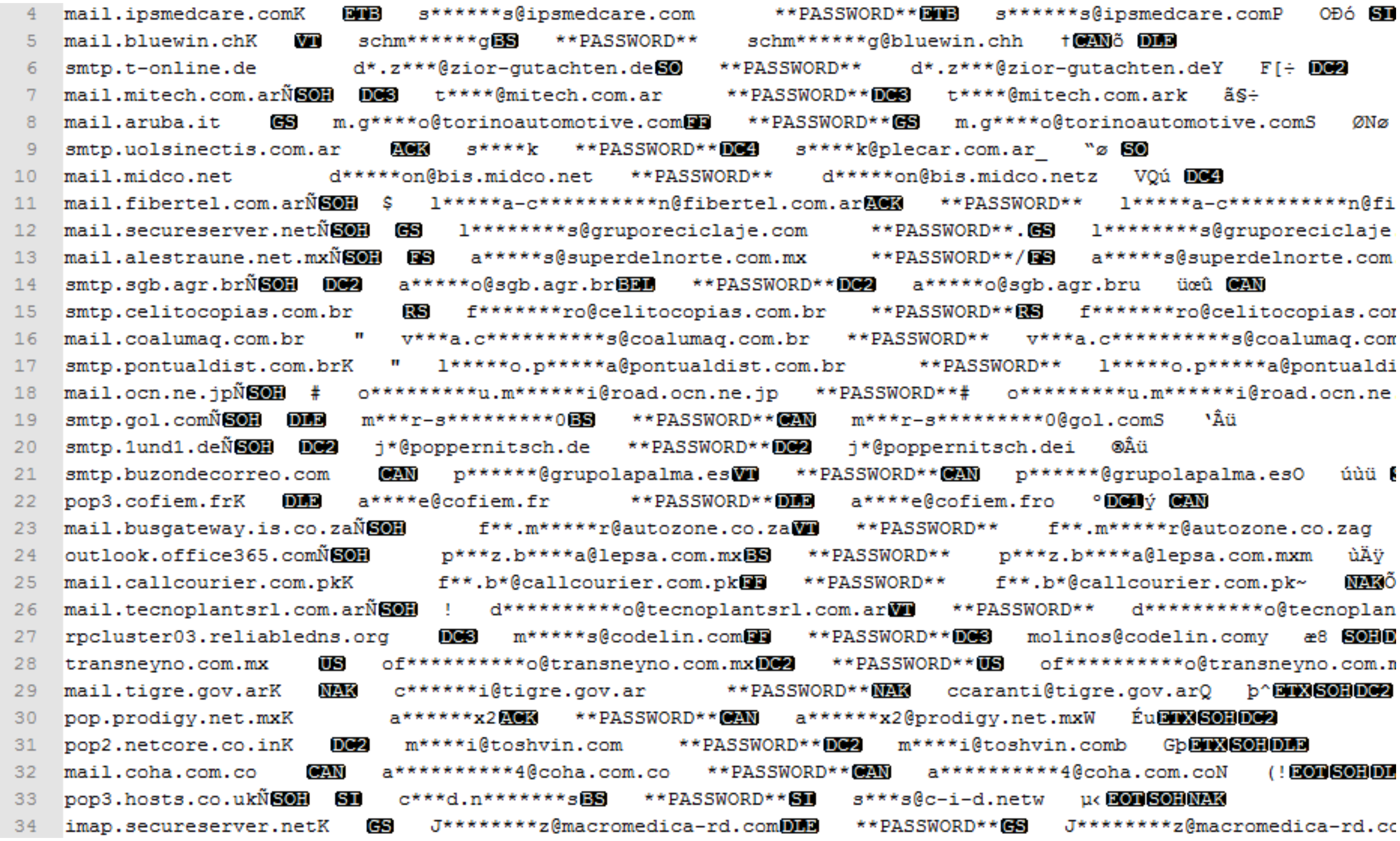
Thunderbird E-mails Grabber module

A data exfiltration module for Thunderbird. The module iterates through Thunderbird profiles located in %AppData%\Roaming\Thunderbird\Profiles\, parses Thunderbird data files and extracts displayed names and mail addresses of sender and recipient, and extracts the e-mail subject and body. It then sends the collected e-mails to C2.

Spam module

The module is responsible for sending spam. It queries C2 until it receives a response with a spam task that usually consist of three parts:

- A list of e-mail servers and compromised accounts to be used to send spam; dozens of compromised accounts are stored in a single task.
- A list of targeted e-mails, recipient e-mail and name, sender e-mail and name.
- A spam template with subject, body and attachments.



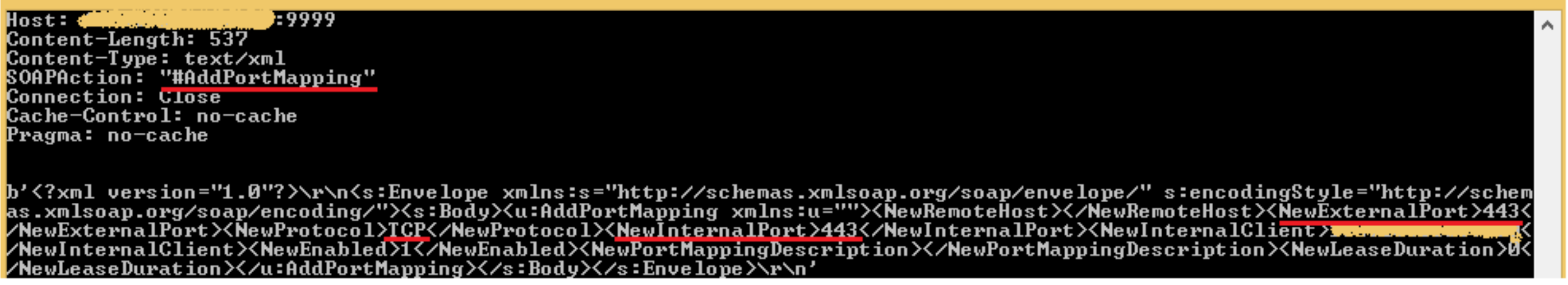
```
mail.ipsmedcare.comK   ETB   s******s@ipsmedcare.com        **PASSWORD**ETB   s******s@ipsmedcare.comP   OÐó SI
mail.bluewin.chK   VT   schm******gBS   **PASSWORD**    schm******g@bluewin.chh   †CANõ DLE
smtp.t-online.de        d*.z***@zior-gutachten.deSO   **PASSWORD**    d*.z***@zior-gutachten.deY   F[÷ DC2
mail.mitech.com.arÑSOH  DC3   t****@mitech.com.ar      **PASSWORD**DC3   t****@mitech.com.ark   ã§÷
mail.aruba.it     GS   m.g****o@torinoautomotive.comFF   **PASSWORD**GS   m.g****o@torinoautomotive.comS   ØNø
smtp.uolsinectis.com.ar     ACK   s****k   **PASSWORD**DC4   s****k@plecar.com.ar_   “ø SO
mail.midco.net        d*****on@bis.midco.net   **PASSWORD**    d*****on@bis.midco.netz   VQú DC4
mail.fibertel.com.arÑSOH  $   l*****a-c**********n@fibertel.com.arACK   **PASSWORD**   l*****a-c**********n@fi
mail.secureserver.netÑSOH  GS   l********s@gruporeciclaje.com      **PASSWORD**.GS   l********s@gruporeciclaje
mail.alestraune.net.mxÑSOH  FS   a*****s@superdelnorte.com.mx      **PASSWORD**/FS   a*****s@superdelnorte.com
smtp.sgb.agr.brÑSOH  DC2   a*****o@sgb.agr.brBEL   **PASSWORD**DC2   a*****o@sgb.agr.bru   üœû CAN
smtp.celitocopias.com.br     RS   f*******ro@celitocopias.com.br     **PASSWORD**RS   f*******ro@celitocopias.com
mail.coalumaq.com.br    "   v***a.c**********s@coalumaq.com.br   **PASSWORD**   v***a.c**********s@coalumaq.com
smtp.pontualdist.com.brK   "   l*****o.p*****a@pontualdist.com.br       **PASSWORD**   l*****o.p*****a@pontualdi
mail.ocn.ne.jpÑSOH  #   o*********u.m******i@road.ocn.ne.jp   **PASSWORD**#   o*********u.m******i@road.ocn.ne
smtp.gol.comÑSOH  DLE   m***r-s*********0BS   **PASSWORD**CAN   m***r-s*********0@gol.comS   ‘Âü
smtp.1und1.deÑSOH  DC2   j*@poppernitsch.de   **PASSWORD**DC2   j*@poppernitsch.dei   ®Âü
smtp.buzondecorreo.com     CAN   p******@grupolapalma.esVT   **PASSWORD**CAN   p******@grupolapalma.esO   úùü
pop3.cofiem.frK   DLE   a****e@cofiem.fr        **PASSWORD**DLE   a****e@cofiem.fro   °DC1ý CAN
mail.busgateway.is.co.zaÑSOH      f**.m*****r@autozone.co.zaVT   **PASSWORD**    f**.m*****r@autozone.co.zag
outlook.office365.comÑSOH      p***z.b****a@lepsa.com.mxBS   **PASSWORD**    p***z.b****a@lepsa.com.mxm   ùÄÿ
mail.callcourier.com.pkK      f**.b*@callcourier.com.pkFF   **PASSWORD**    f**.b*@callcourier.com.pk~    NAKÕ
mail.tecnoplantsrl.com.arÑSOH  !   d*********o@tecnoplantsrl.com.arVT   **PASSWORD**   d*********o@tecnoplan
rpcluster03.reliabledns.org     DC3   m*****s@codelin.comFF   **PASSWORD**DC3   molinos@codelin.comy   æ8 SOHD
transneyno.com.mx    US   of*********o@transneyno.com.mxDC2   **PASSWORD**US   of*********o@transneyno.com.m
mail.tigre.gov.arK   NAK   c******i@tigre.gov.ar        **PASSWORD**NAK   ccaranti@tigre.gov.arQ   þ^ETXSOHDC2
pop.prodigy.net.mxK         a******x2ACK   **PASSWORD**CAN   a******x2@prodigy.net.mxW   ÉuETXSOHDC2
pop2.netcore.co.inK   DC2   m****i@toshvin.com     **PASSWORD**DC2   m****i@toshvin.comb   GþETXSOHDLE
mail.coha.com.co     CAN   a**********4@coha.com.co   **PASSWORD**CAN   a**********4@coha.com.coN   (!EOTSOHDL
pop3.hosts.co.ukÑSOH  SI   c***d.n*******sBS   **PASSWORD**SI   s***s@c-i-d.netw   µ‹EOTSOHNAK
imap.secureserver.netK   GS   J********z@macromedica-rd.comDLE   **PASSWORD**GS   J********z@macromedica-rd.co
```

Redacted list of email servers, compromised accounts used for spamming

Two of the 10 modules we were able to obtain were spam modules. Their functionality is one and the same, but the module IDs differ.

UPnP module

An auxiliary module for testing the possibility of connecting to the infected system from the outside. In the settings of this module, which are sent by C2, together with the module itself, the external IP address of the infected system is transmitted. The first thing this module does is enumerate the network interfaces and compare their addresses with the IP address obtained from the module's configuration settings. If a suitable network interface is found, the module opens ports for listening and waits for an incoming connection. The module can open the following ports: 80, 443, 8080, 8090, 7080, 8443, 20, 21, 22, 53, 143, 465, 990, 993, 995. If a suitable network interface is not found, it uses the SSDP protocol to find devices (modem, router, etc.) with Internet access. If suitable devices are found, the module tries to reconfigure them using AddPortMapping to allow port forwarding.



```
Host: [redacted]:9999
Content-Length: 537
Content-Type: text/xml
SOAPAction: "#AddPortMapping"
Connection: Close
Cache-Control: no-cache
Pragma: no-cache


b'<?xml version="1.0"?>\r\n<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" s:encodingStyle="http://schem
as.xmlsoap.org/soap/encoding/"><s:Body><u:AddPortMapping xmlns:u=""><NewRemoteHost></NewRemoteHost><NewExternalPort>443<
/NewExternalPort><NewProtocol>TCP</NewProtocol><NewInternalPort>443</NewInternalPort><NewInternalClient>[redacted]<
/NewInternalClient><NewEnabled>1</NewEnabled><NewPortMappingDescription></NewPortMappingDescription><NewLeaseDuration>0<
/NewLeaseDuration></u:AddPortMapping></s:Body></s:Envelope>\r\n'
```

Example of AddPortMapping for 443 port forwarding

Statistics

Since Emotet's return in November 2021, we have observed its activity gradually increase. In March 2022, however, based on our telemetry, the number of attacked users shot up from 2,847 in February to 9,086 —— more than threefold growth.

Dynamics of the number of attacked users in recent Emotet attacks, November 2021—March 2022 (download)

A similar upsurge we observed in March in the number of Emotet detections.

Dynamics of the number of Emotet detections, November 2021—March 2022 (download)

Victimology

Emotet infects computers of companies and individual users all over the world. In Q1 2022, according to our telemetry, users of the following countries were most often targeted by Emotet: Italy (10.04%), Russia (9.87%), Japan (8.55%), Mexico (8.36%), Brazil (6.88%), Indonesia (4.92%), India (3.21%), Vietnam (2.70%), China (2.62), Germany (2.19%) and Malaysia (2.13%).

Geographical distribution of Emotet targets, Q1 2022 (download)

Conclusion

The current set of modules is capable of performing a large set of malicious actions: stealing e-mails, passwords and login data from various sources; sending spam. All these modules, except those for Thunderbird, in one form or another, have been used before by Emotet. However, there are still modules that we have not been able to obtain yet. In addition, our telemetry shows significant growth in the number of attacked users in March. We continue to actively monitor the Emotet family. More information about the malware we provide in our private reports on Kaspersky Threat Intelligence Portal.

Indicators of Compromise

Note: Because Emotet is polymorphic malware, there are no IOC hashes.

C2 IP addresses

70[.]36.102.35:443 197[.]242.150.244:8080 188[.]44.20.25:443 45[.]118.135.203:7080 92[.]240.254.110:8080 103[.]43.46.182:443
1[.]234.2.232:8080 50[.]116.54.215:443 51[.]91.76.89:8080 206[.]188.212.92:8080 153[.]126.146.25:7080 178[.]79.147.66:8080
217[.]182.25.250:8080 196[.]218.30.83:443 51[.]91.7.5:8080 72[.]15.201.15:8080 119[.]193.124.41:7080 5[.]9.116.246:8080
151[.]106.112.196:8080 101[.]50.0.91:8080 45[.]142.114.231:8080 185[.]157.82.211:8080 46[.]55.222.11:443 103[.]75.201.2:443
176[.]56.128.118:443 176[.]104.106.96:8080 107[.]182.225.142:8080 31[.]24.158.56:8080 51[.]254.140.238:7080 159[.]65.88.10:8080
82[.]165.152.127:8080 146[.]59.226.45:443 173[.]212.193.249:8080 212[.]24.98.99:8080 212[.]237.17.99:8080 110[.]232.117.186:8080
131[.]100.24.231:80 209[.]250.246.206:443 195[.]201.151.129:8080 138[.]185.72.26:8080

- Botnets
- Emotet
- Malware
- Malware Descriptions
- Malware Statistics
- Malware Technologies
- Trojan Banker

Authors

- AMR