New Year, New Version of DanaBot

proofpoint.com/us/blog/threat-insight/new-year-new-version-danabot

January 26, 2021

Blog
Threat Insight
New Year, New Version of DanaBot

January 26, 2021 Dennis Schwarz, Axel F., and Brandon Murphy

Proofpoint researchers discovered an updated version of DanaBot in the wild. DanaBot is a banking/stealer malware first discovered by Proofpoint in May 2018. There have been at least three significant versions of the malware:

- Version 1: DanaBot - A new banking Trojan surfaces Down Under
- Version 2: DanaBot Gains Popularity and Targets US Organizations in Large Campaigns
- Version 3: ESET's DanaBot updated with new C&C communication

This will be the fourth major update.

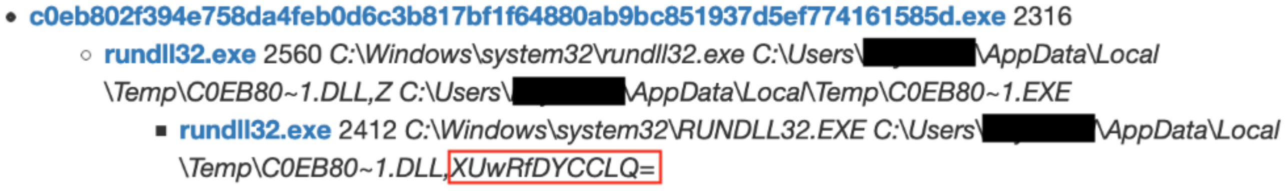
From May 2018 to June 2020, DanaBot was a fixture in the crimeware threat landscape. Proofpoint researchers observed multiple threat actors with at least 12 affiliate IDs in version 2 and 38 IDs in version 3. These affiliate identifications (IDs) represent the threat actors the DanaBot operators serve. Distribution has typically targeted financial institutions predominantly located in the United States, Canada, Germany, United Kingdom, Australia, Italy, Poland, Mexico, and Ukraine. After June 2020, there was a sharp decline in DanaBot activity in Proofpoint's data and in public threat intel repositories (e.g. MalwareBazaar and #DanaBot). It disappeared from the threat landscape without a clear cause.

Starting in late October 2020, we observed a significant update to DanaBot samples appearing in VirusTotal. At the time of publication, Proofpoint researchers spotted two affiliate IDs using this latest version with at least one distribution method. While it has not returned to its former scale, DanaBot is malware that defenders should put back on their radar.

Malware Analysis

The sample with a SHA-256 hash of c0eb802f394e758da4feb0d6c3b817bf1f64880ab9bc851937d5ef774161585d was used for this analysis.

Like previous versions of DanaBot, version 4 is a large, multithreaded, modular malware written in the Delphi programming language. A loader component (EXE) decrypts, decompresses, and executes a secondary component (DLL) seen in Figure 1:



```
• c0eb802f394e758da4feb0d6c3b817bf1f64880ab9bc851937d5ef774161585d.exe 2316
    ○ rundll32.exe 2560 C:\Windows\system32\rundll32.exe C:\Users\[redacted]\AppData\Local
      \Temp\C0EB80~1.DLL,Z C:\Users\[redacted]\AppData\Local\Temp\C0EB80~1.EXE
        ▪ rundll32.exe 2412 C:\Windows\system32\RUNDLL32.EXE C:\Users\[redacted]\AppData\Local
          \Temp\C0EB80~1.DLL,XUwRfDYCCLQ=
```

Figure 1: Malware execution

The secondary component removes the loader and reruns itself using a specially crafted export name highlighted above in red in Figure 1. The export name is base64 decoded and the first three bytes are subtracted from each other (i.e., running_mode = byte_0 – byte_1 – byte_2). This value determines the running mode of the secondary component, with four options available:

Running Mode	Description
0	Main component
1	TOR component
2	Used for process injection of downloaded files
3	Module component

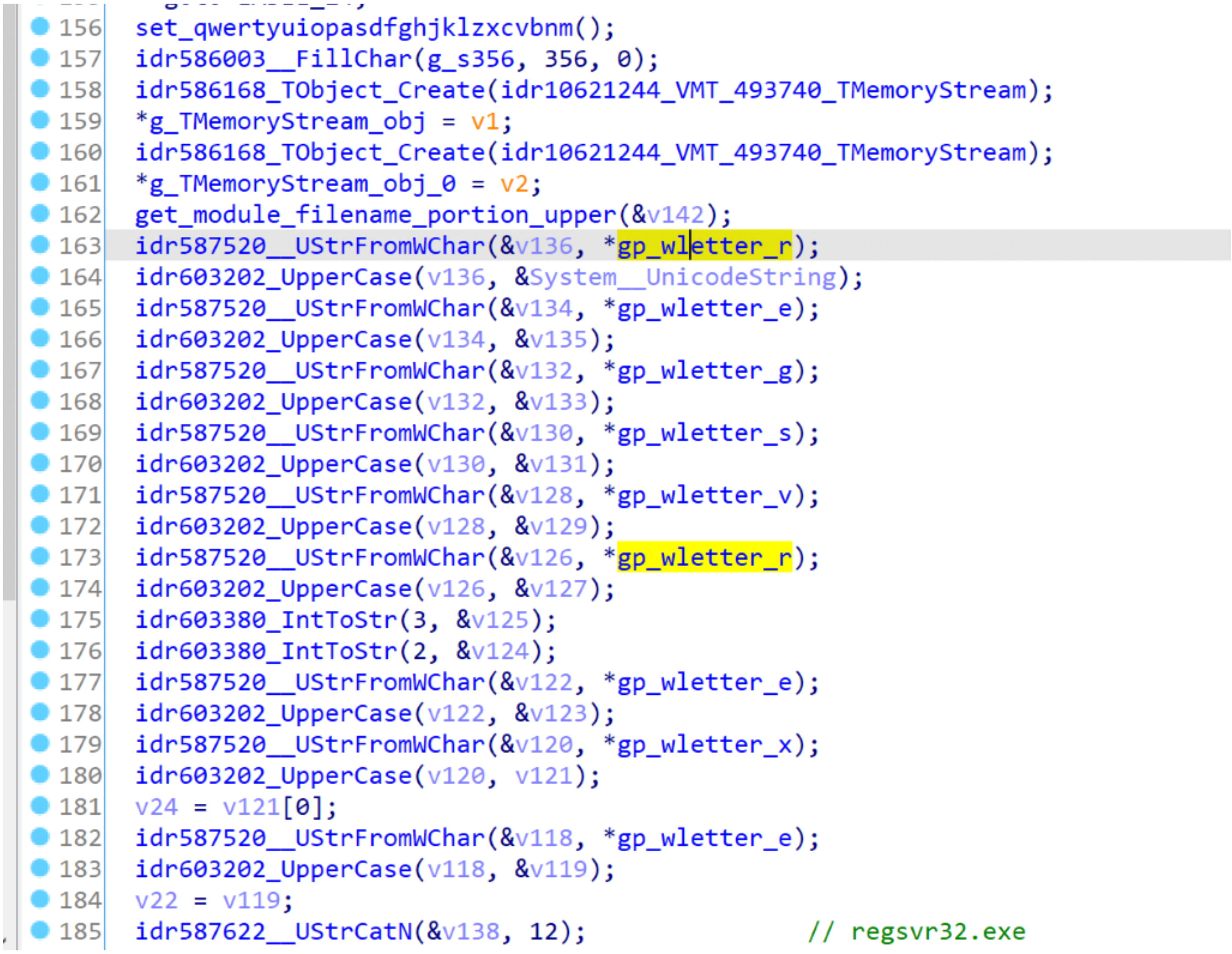
This analysis will mostly focus on mode 0, the main component.

Anti-Analysis

Besides being written in Delphi there are a few other anti-analysis features in the malware:

Some strings are constructed one character at a time (Figure 2)

- Some Windows API functions are resolved at run-time
- When a malware-related file is read or written to the filesystem, it is done in the middle of benign decoy file reads or writes
- Persistence is maintained by creating an LNK file that executes the main component in the user’s Startup directory. This file is only written once a WM_QUERYENDSESSION Windows event is received when the user logs off

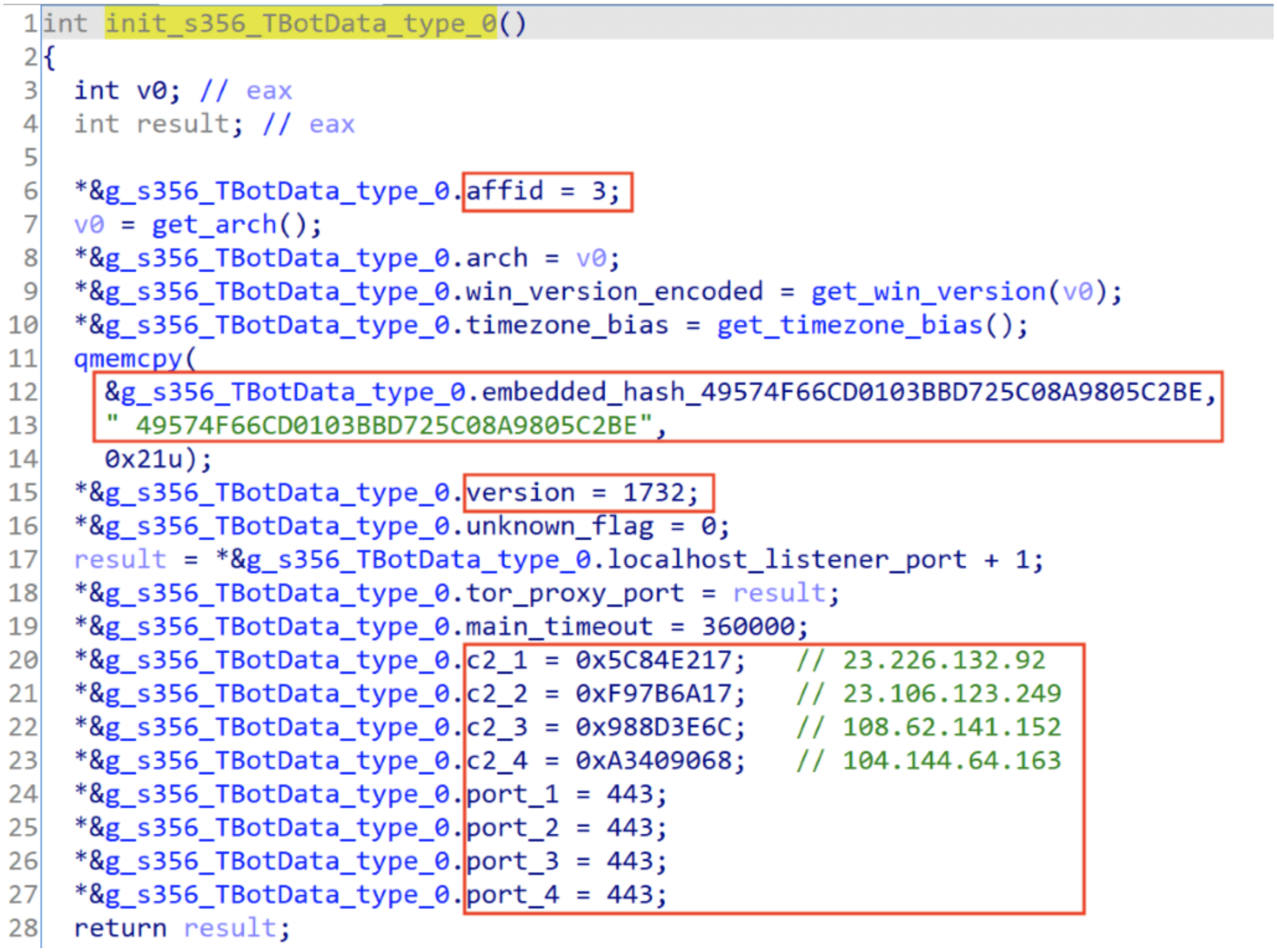


```
set_qwertyuiopasdfghjklzxcvbnm();
idr586003__FillChar(g_s356, 356, 0);
idr586168_TObject_Create(idr10621244_VMT_493740_TMemoryStream);
*g_TMemoryStream_obj = v1;
idr586168_TObject_Create(idr10621244_VMT_493740_TMemoryStream);
*g_TMemoryStream_obj_0 = v2;
get_module_filename_portion_upper(&v142);
idr587520__UStrFromWChar(&v136, *gp_wletter_r);
idr603202_UpperCase(v136, &System__UnicodeString);
idr587520__UStrFromWChar(&v134, *gp_wletter_e);
idr603202_UpperCase(v134, &v135);
idr587520__UStrFromWChar(&v132, *gp_wletter_g);
idr603202_UpperCase(v132, &v133);
idr587520__UStrFromWChar(&v130, *gp_wletter_s);
idr603202_UpperCase(v130, &v131);
idr587520__UStrFromWChar(&v128, *gp_wletter_v);
idr603202_UpperCase(v128, &v129);
idr587520__UStrFromWChar(&v126, *gp_wletter_r);
idr603202_UpperCase(v126, &v127);
idr603380_IntToStr(3, &v125);
idr603380_IntToStr(2, &v124);
idr587520__UStrFromWChar(&v122, *gp_wletter_e);
idr603202_UpperCase(v122, &v123);
idr587520__UStrFromWChar(&v120, *gp_wletter_x);
idr603202_UpperCase(v120, v121);
v24 = v121[0];
idr587520__UStrFromWChar(&v118, *gp_wletter_e);
idr603202_UpperCase(v118, &v119);
v22 = v119;
idr587622__UStrCatN(&v138, 12);                // regsvr32.exe
```

Figure 2: String obfuscation example, where strings are constructed one character at a time

Configuration

DanaBot’s configuration is hardcoded into a 356-byte structure (Figure 3):



```
int init_s356_TBotData_type_0()
{
  int v0; // eax
  int result; // eax

  *&g_s356_TBotData_type_0.affid = 3;
  v0 = get_arch();
  *&g_s356_TBotData_type_0.arch = v0;
  *&g_s356_TBotData_type_0.win_version_encoded = get_win_version(v0);
  *&g_s356_TBotData_type_0.timezone_bias = get_timezone_bias();
  qmemcpy(
    &g_s356_TBotData_type_0.embedded_hash_49574F66CD0103BBD725C08A9805C2BE,
    " 49574F66CD0103BBD725C08A9805C2BE",
    0x21u);
  *&g_s356_TBotData_type_0.version = 1732;
  *&g_s356_TBotData_type_0.unknown_flag = 0;
  result = *&g_s356_TBotData_type_0.localhost_listener_port + 1;
  *&g_s356_TBotData_type_0.tor_proxy_port = result;
  *&g_s356_TBotData_type_0.main_timeout = 360000;
  *&g_s356_TBotData_type_0.c2_1 = 0x5C84E217;    // 23.226.132.92
  *&g_s356_TBotData_type_0.c2_2 = 0xF97B6A17;    // 23.106.123.249
  *&g_s356_TBotData_type_0.c2_3 = 0x988D3E6C;    // 108.62.141.152
  *&g_s356_TBotData_type_0.c2_4 = 0xA3409068;    // 104.144.64.163
  *&g_s356_TBotData_type_0.port_1 = 443;
  *&g_s356_TBotData_type_0.port_2 = 443;
  *&g_s356_TBotData_type_0.port_3 = 443;
  *&g_s356_TBotData_type_0.port_4 = 443;
  return result;
}
```

Figure 3: Configuration structure of DanaBot

Key configuration items are highlighted in red in Figure 3 and include the following:

Affiliate ID

As previously reported in DanaBot control panel revealed, we believe DanaBot is set up as a “malware as a service” in which one threat actor controls a global command and control (C&C) panel and infrastructure then sells access to other threat actors known as affiliates.

This field likely represents the ID of the affiliate associated with the sample. At the time of publication, only two IDs were found: 3 and 21. It is currently unclear whether version 4 affiliate IDs will overlap with previous version affiliate IDs, though they did change between versions 2 and 3.

Embedded Hash

It is currently unclear what the following embedded hash values represent:

E1D3580C52F82AF2B3596E20FB85D9F4

- DE420A65BFC5F29167A85A5199065A0E
- E0ECDBB46B59DFAB6F7CB1136E7496F5
- 429B39BF421C0F74463EF2A17209ADAA
- 6266E79288DFE2AE2C2DB47563C7F93A

DE6DF8FA2198DD77CFD93D89D8ECC62D

Version

This field below likely represents a version number that increments in newer samples:

- 1650
- 1701
- 1705
- 1732
- 1755

C&C IP Addresses and Ports

The IP addresses are hardcoded as DWORD values and are set to the following in the analyzed sample:

- 23[.]226.132.92
- 23[.]106.123.249
- 108[.]62.141.152
- 104[.]144.64.163

Version 3 of DanaBot mixed in decoy C&C addresses, but it does not appear version 4 is making use of them.

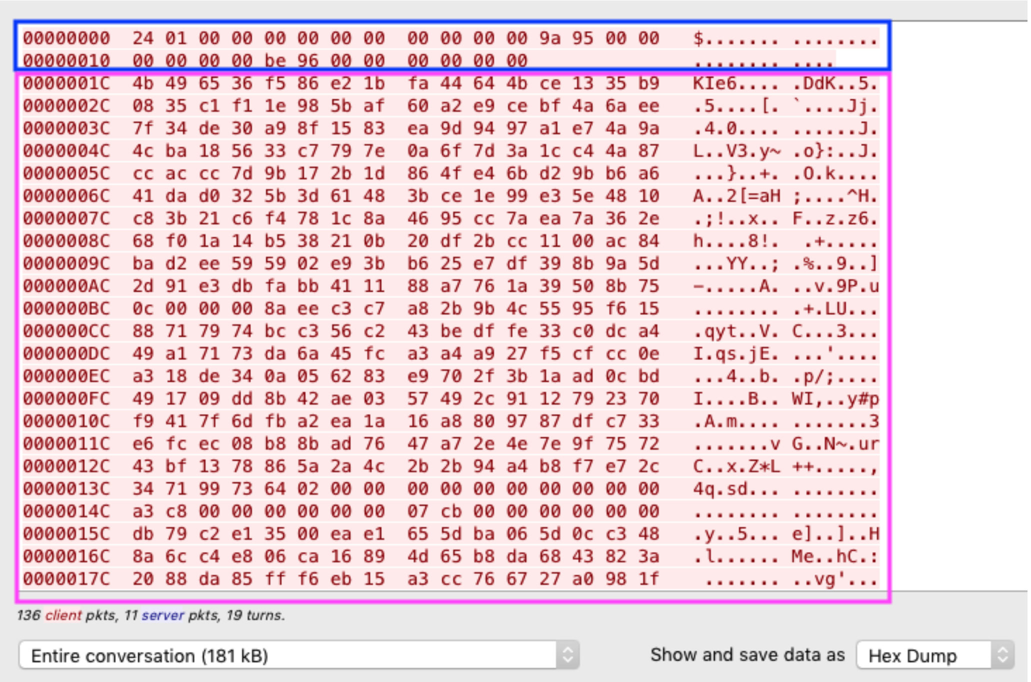
TOR

DanaBot has functionality to switch to TOR-based C&C. The analyzed sample contains the following hardcoded onion hostname:

5jjsgjephjcua63go2o5donzw5x4hiwn6wh2dennmyq65pbhk6qflzyd\.onion

Command and Control

The C&C protocol in version 4 is similar to version 3. An example request is shown in Figure 4:



```
00000000  24 01 00 00 00 00 00 00  00 00 00 00 9a 95 00 00   $....... ........
00000010  00 00 00 00 be 96 00 00  00 00 00 00               ........ ....
0000001C  4b 49 65 36 f5 86 e2 1b  fa 44 64 4b ce 13 35 b9   KIe6.... .DdK..5.
0000002C  08 35 c1 f1 1e 98 5b af  60 a2 e9 ce bf 4a 6a ee   .5....[. `....Jj.
0000003C  7f 34 de 30 a9 8f 15 83  ea 9d 94 97 a1 e7 4a 9a   .4.0.... ......J.
0000004C  4c ba 18 56 33 c7 79 7e  0a 6f 7d 3a 1c c4 4a 87   L..V3.y~ .o}:..J.
0000005C  cc ac cc 7d 9b 17 2b 1d  86 4f e4 6b d2 9b b6 a6   ...}..+. .O.k....
0000006C  41 da d0 32 5b 3d 61 48  3b ce 1e 99 e3 5e 48 10   A..2[=aH ;....^H.
0000007C  c8 3b 21 c6 f4 78 1c 8a  46 95 cc 7a ea 7a 36 2e   .;!..x.. F..z.z6.
0000008C  68 f0 1a 14 b5 38 21 0b  20 df 2b cc 11 00 ac 84   h....8!.  .+.....
0000009C  ba d2 ee 59 59 02 e9 3b  b6 25 e7 df 39 8b 9a 5d   ...YY..; .%..9..]
000000AC  2d 91 e3 db fa bb 41 11  88 a7 76 1a 39 50 8b 75   -.....A. ..v.9P.u
000000BC  0c 00 00 00 8a ee c3 c7  a8 2b 9b 4c 55 95 f6 15   ........ .+.LU...
000000CC  88 71 79 74 bc c3 56 c2  43 be df fe 33 c0 dc a4   .qyt..V. C...3...
000000DC  49 a1 71 73 da 6a 45 fc  a3 a4 a9 27 f5 cf cc 0e   I.qs.jE. ...'....
000000EC  a3 18 de 34 0a 05 62 83  e9 70 2f 3b 1a ad 0c bd   ...4..b. .p/;....
000000FC  49 17 09 dd 8b 42 ae 03  57 49 2c 91 12 79 23 70   I....B.. WI,..y#p
0000010C  f9 41 7f 6d fb a2 ea 1a  16 a8 80 97 87 df c7 33   .A.m.... .......3
0000011C  e6 fc ec 08 b8 8b ad 76  47 a7 2e 4e 7e 9f 75 72   .......v G..N~.ur
0000012C  43 bf 13 78 86 5a 2a 4c  2b 2b 94 a4 b8 f7 e7 2c   C..x.Z*L ++.....,
0000013C  34 71 99 73 64 02 00 00  00 00 00 00 00 00 00 00   4q.sd... ........
0000014C  a3 c8 00 00 00 00 00 00  07 cb 00 00 00 00 00 00   ........ ........
0000015C  db 79 c2 e1 35 00 ea e1  65 5d ba 06 5d 0c c3 48   .y..5... e]..]..H
0000016C  8a 6c c4 e8 06 ca 16 89  4d 65 b8 da 68 43 82 3a   .l...... Me..hC.:
0000017C  20 88 da 85 ff f6 eb 15  a3 cc 76 67 27 a0 98 1f    ....... ..vg'...
```

136 client pkts, 11 server pkts, 19 turns.

Figure 4: Example C&C request

It is still a binary protocol using mostly TCP port 443. Requests and responses have a plaintext header (highlighted in blue in Figure 4) followed by command data (highlighted in purple). The header is 28-bytes and has the following fields:

Offset	Size	Name	Notes
0x00	8-bytes	Data length	
0x08	4-bytes	Data compression/encryption mode	Four modes, described below
0x0c	8-bytes	Random value	
0x14	8-bytes	Checksum	Value = data length + random value

The command data structure is:

- AES-encrypted data
- Padding length (4-bytes)
- RSA-encrypted session key

  RSA Signature (in responses)

Depending on the command, data can be compressed using zlib (mode 1), ZIP (mode 2), or not compressed (modes 0 and 3).

Data is encrypted with AES-256 in CBC mode using a generated session key. In modes 0, 1, and 2, the session key is randomly generated and encrypted with RSA. For requests to the C&C server, an embedded public RSA key is used. For responses from the C&C server, a generated RSA key is used (see below). For mode 3, the session key is CryptDeriveKey'd based on the MD5 uppercase hex digest of the bot ID.

Responses from the C&C also contain an RSA signature which is verified using an embedded public RSA key.

The first request to the C&C server is a key exchange where an RSA key pair is generated by the malware, and the public key is sent to the C&C server. There is no response from the C&C for this request. Session keys used in future responses from the C&C server will be encrypted using this key.

The second request is an initial beacon to the C&C server. The data is a 479-byte structure containing:

Offset	**Size**	**Name**	**Notes**
0x00	4-bytes	Length	
0x04	8-bytes	Random value	
0x0C	8-bytes	Checksum	Value = data length + random value
0x14	4-bytes	Affiliate ID	See Configuration section above
0x18	4-bytes	Command	Described below
0x1c	4-bytes	Sub-command	Described below
0x20	4-bytes	Version	See Configuration section above
0x24	4-bytes	Is admin flag	
0x28	4-bytes	Process integrity level	
0x2c	4-bytes	Architecture	

0x30	4-bytes	Windows version	Encoded into a DWORD value
0x34	4-bytes	Time zone bias	
0x38	36-bytes	Unknown null bytes	
0x5c	41-bytes	Bot ID	Prepended with string length and CRC32 value. MD5 uppercase hex digest of hardware profile GUID
0x85	41-bytes	Embedded hash value	Prepended with string length and CRC32 value. See Configuration section above
0xae	41-bytes	Checksum 2	Prepended with string length and CRC32 value. MD5 uppercase hex digest of affiliate ID, bot ID, and embedded hash values concatenated together
0xd7	41-bytes	MD5 uppercase hex digest of three random values	Prepended with string length and CRC32 value
0x100	remaining	Unknown null bytes	

Once decrypted (this particular response uses mode 0 and an extra layer of mode 3), the response from the C&C for the initial beacon is an echo of the request.

Other commands use similar structures but will not be detailed in this post.

Commands

Some of the main C&C commands we have identified include:

Command 1024, Sub-command 0

The initial beacon described in the C&C section above.

Command 2048, Sub-command 0

This command returns three hash values. It is unclear what the hashes are of, but they represent:

Current set of “CommandRecord”s

- Current set of modules and/or files to download and execute
- Current set of “OnlineRec”s

Command 2048, Sub-command 1

Get updated list of C&C IP addresses.

Command 2048, Sub-command 2

This command returns a list of hash values. The hash values represent individual “CommandRecord”s.

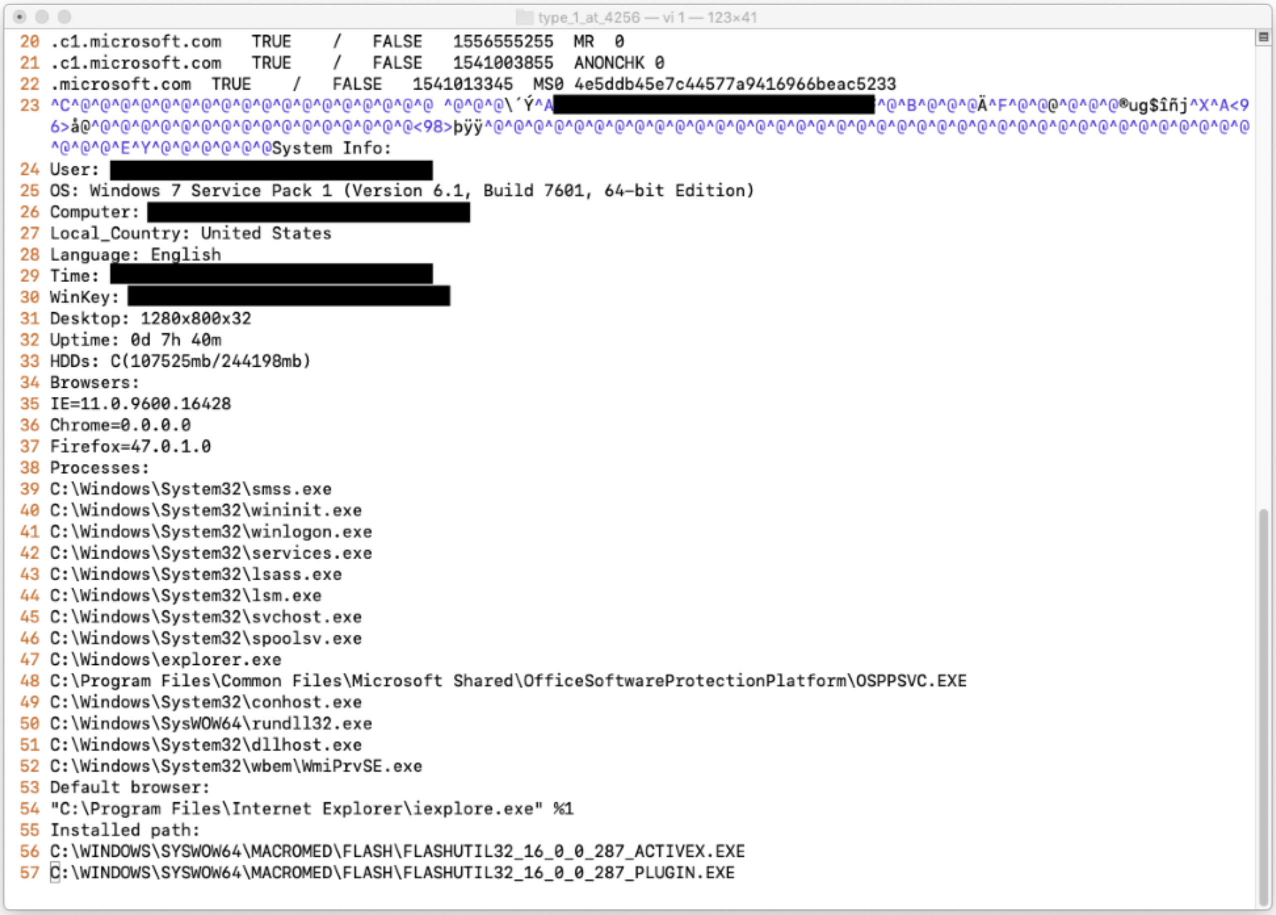
Command 2048, Sub-command 3

This command is used to fetch a “CommandRecord”. These records map to three commands:

- Switch to TOR C&C communications
- Set C&C sleep value

Enable the built-in stealer and system information components

The stealer component steals credentials from various software such as web browsers and File Transfer Protocol (FTP) clients. The system information component collects system information seen below in Figure 5:



```
.c1.microsoft.com   TRUE    /   FALSE   1556555255  MR  0
.c1.microsoft.com   TRUE    /   FALSE   1541003855  ANONCHK 0
.microsoft.com  TRUE    /   FALSE   1541013345  MS0 4e5ddb45e7c44577a9416966beac5233
^C^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@ ^@^@^@\´Ý^A[redacted]^@^B^@^@^@Ä^F^@^@@^@^@^@®ug$îñj^X^A<9
6>å@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@<98>þÿÿ^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@
^@^@^@^E^Y^@^@^@^@^@System Info:
User: [redacted]
OS: Windows 7 Service Pack 1 (Version 6.1, Build 7601, 64-bit Edition)
Computer: [redacted]
Local_Country: United States
Language: English
Time: [redacted]
WinKey: [redacted]
Desktop: 1280x800x32
Uptime: 0d 7h 40m
HDDs: C(107525mb/244198mb)
Browsers:
IE=11.0.9600.16428
Chrome=0.0.0.0
Firefox=47.0.1.0
Processes:
C:\Windows\System32\smss.exe
C:\Windows\System32\wininit.exe
C:\Windows\System32\winlogon.exe
C:\Windows\System32\services.exe
C:\Windows\System32\lsass.exe
C:\Windows\System32\lsm.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\spoolsv.exe
C:\Windows\explorer.exe
C:\Program Files\Common Files\Microsoft Shared\OfficeSoftwareProtectionPlatform\OSPPSVC.EXE
C:\Windows\System32\conhost.exe
C:\Windows\SysWOW64\rundll32.exe
C:\Windows\System32\dllhost.exe
C:\Windows\System32\wbem\WmiPrvSE.exe
Default browser:
"C:\Program Files\Internet Explorer\iexplore.exe" %1
Installed path:
C:\WINDOWS\SYSWOW64\MACROMED\FLASH\FLASHUTIL32_16_0_0_287_ACTIVEX.EXE
C:\WINDOWS\SYSWOW64\MACROMED\FLASH\FLASHUTIL32_16_0_0_287_PLUGIN.EXE
```

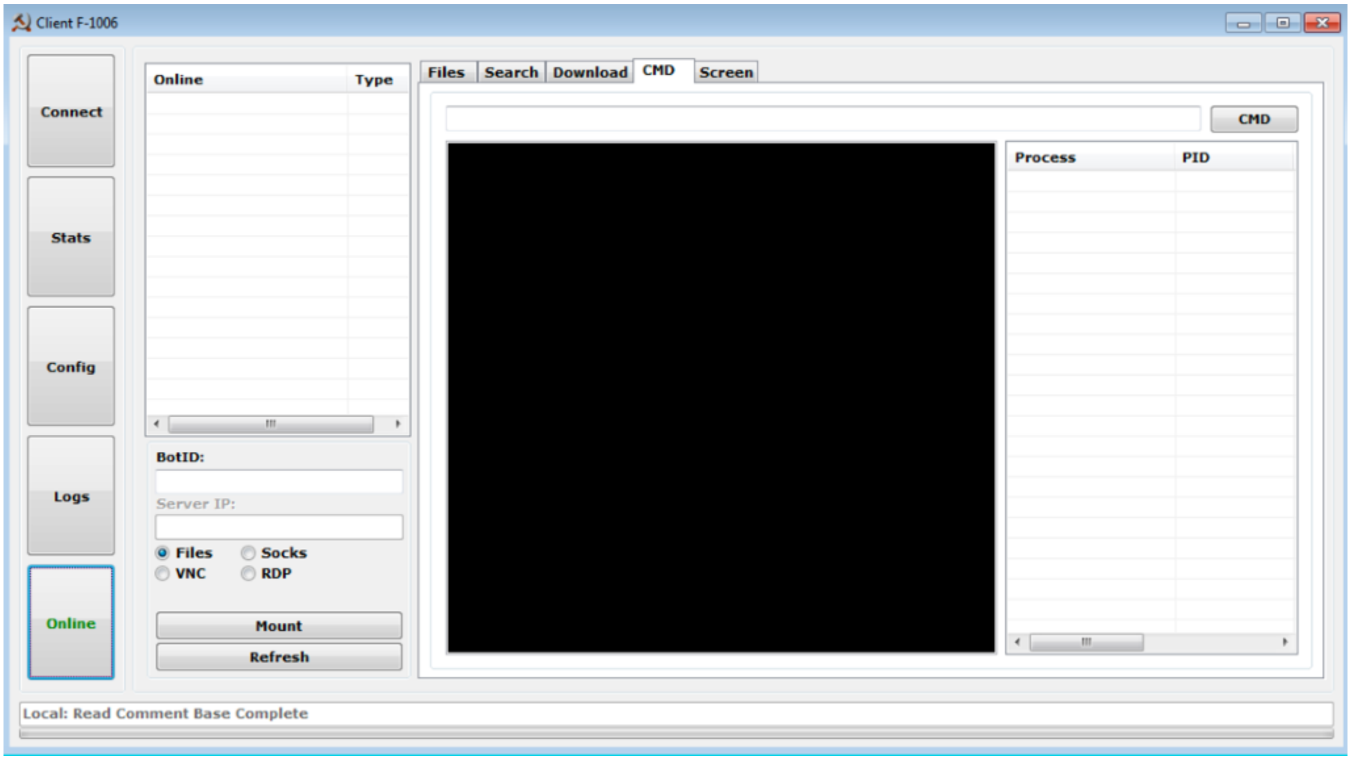
Figure 5: Example output from stealer/system information report

Command 2048, Sub-command 4

This command is used to send arbitrary data such as an initial screenshot and report from the stealer/system information components to the C&C server.

Command 2048, Sub-command 6

This command returns a set of "OnlineRec"s. Referencing our DanaBot control panel revealed blog post, these records seem to map to similar functionality accessible from the "Online" tab of the version 3 control panel seen below in Figure 6:



● Files ○ Socks
○ VNC ○ RDP

Local: Read Comment Base Complete

Figure 6: "Online" tab of version 3 control panel

This includes functionality such as command shell, file system access, screen/keyboard/mouse access, and SOCKS proxy.

Command 2048, Sub-command 8

Note: all run modes detailed below reference the malware execution section above

This command returns a list of hashes. The hash values represent individual file records. File records are downloaded with command 2048, sub-command 9. File records are used to download files or modules to execute. Executable files can be run using:

regsvr32.exe

- rundll32.exe
- CreateProcess
- Process injection into secondary component using running mode 2

Modules are known as "MLocalProcess"s and are loaded into secondary components using run mode 3. They communicate with the main component over a localhost connection. At the time of publication, we have not seen any modules being distributed by the C&Cs. Based on previous versions of DanaBot, we suspect that the modules will enable the following functionality:

Person-in-the-browser functionality along with web injects

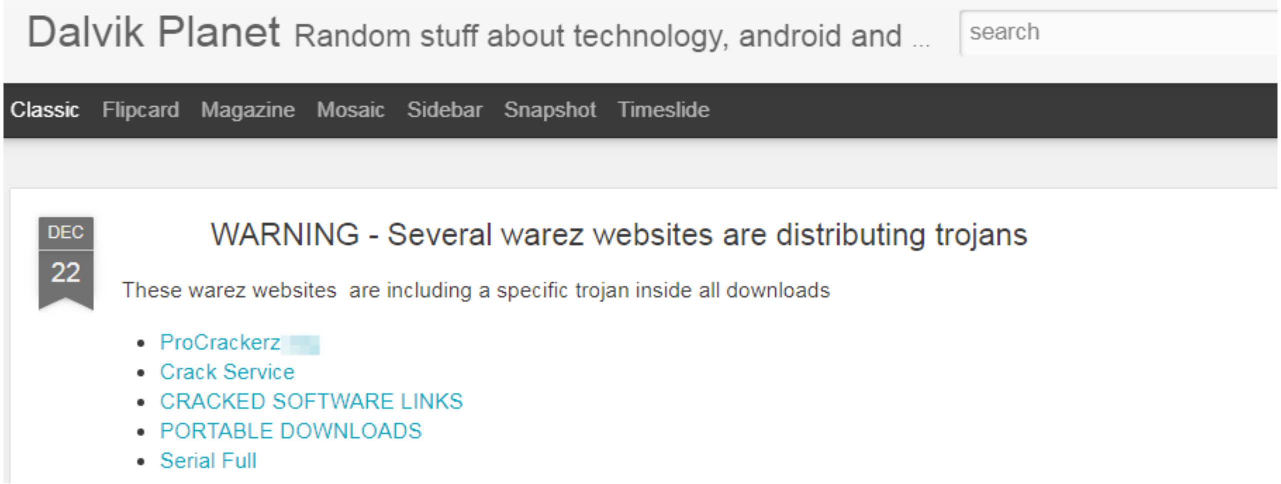
- Video recording of the screen
- Keylogging
- VNC/RDP

Command 2048, Sub-command 10

Used to download TOR. The TOR client will be loaded into a secondary component using running mode 1.

Distribution Via Cracks Websites

Proofpoint researchers were able to narrow down at least one of the DanaBot distribution methods to various software warez and cracks websites that supposedly offer software keys and cracks for a free download, including anti-virus programs, VPNs, graphics editors, document editors, and games. However, the files distributed by these sites are a bundle of several different malware, including DanaBot.

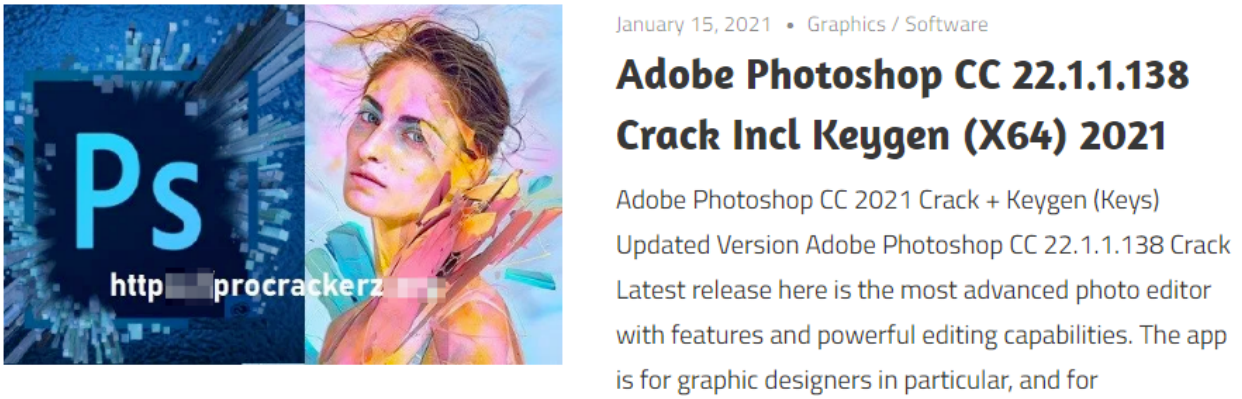


WARNING - Several warez websites are distributing trojans

These warez websites are including a specific trojan inside all downloads

- ProCrackerz[redacted]
- Crack Service
- CRACKED SOFTWARE LINKS
- PORTABLE DOWNLOADS
- Serial Full

Figure 7: The investigation into warez sites started from a December 22, 2020 blog on Dalvik Planet



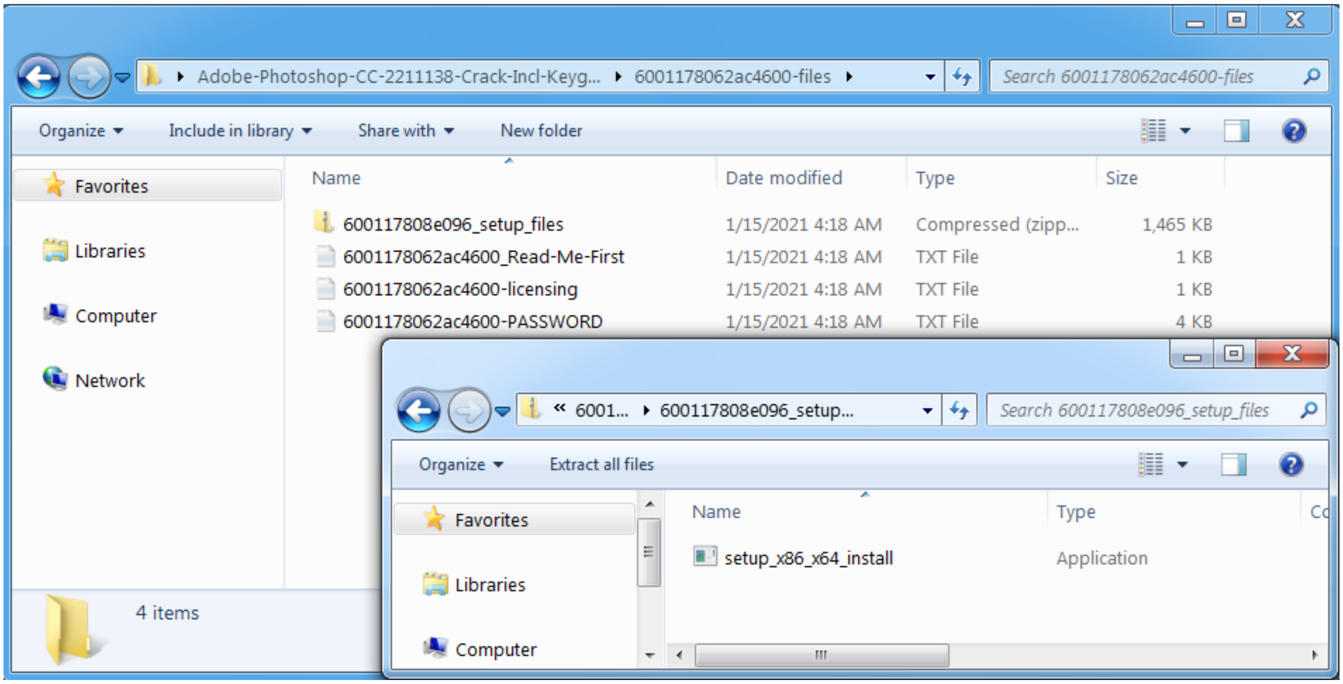
Ps
http[redacted]procrackerz[redacted]
January 15, 2021 • Graphics / Software

Adobe Photoshop CC 22.1.1.138 Crack Incl Keygen (X64) 2021

Adobe Photoshop CC 2021 Crack + Keygen (Keys) Updated Version Adobe Photoshop CC 22.1.1.138 Crack Latest release here is the most advanced photo editor with features and powerful editing capabilities. The app is for graphic designers in particular, and for

Figure 8: Example of a cracks site offering a popular graphics editor keygen for download

A random file “600117809bae5__Adobe-Photoshop-CC-2211138-Crack-Incl-Keygen-X64-2021.zip” was downloaded and analyzed from one of the sites. It contained several “README” files and a password-protected archive containing the initial dropper for the malware bundle, “setup_x86_x64_install.exe.”



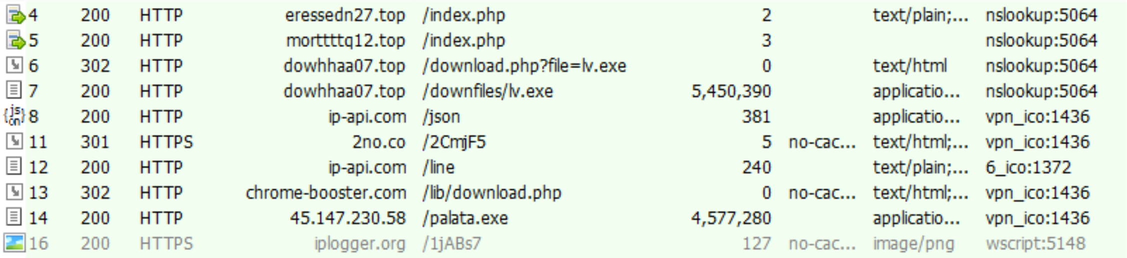
Adobe-Photoshop-CC-2211138-Crack-Incl-Keyg... ▸ 6001178062ac4600-files ▸
Search 6001178062ac4600-files
Organize ▾
Include in library ▾
Share with ▾
New folder
Favorites
Libraries
Computer
Network
4 items
« 6001... ▸ 600117808e096_setup...
Search 600117808e096_setup_files
Organize ▾
Extract all files
Favorites
Libraries
Computer

Name	Date modified	Type	Size
600117808e096_setup_files	1/15/2021 4:18 AM	Compressed (zipp...	1,465 KB
6001178062ac4600_Read-Me-First	1/15/2021 4:18 AM	TXT File	1 KB
6001178062ac4600-licensing	1/15/2021 4:18 AM	TXT File	1 KB
6001178062ac4600-PASSWORD	1/15/2021 4:18 AM	TXT File	4 KB

Name	Type
setup_x86_x64_install	Application

Figure 9: A zip archive downloaded from a warez site containing the initial dropper for the malware bundle

Running this executable generated the following traffic.



```
4    200  HTTP        eressedn27.top   /index.php                        2         text/plain;...  nslookup:5064
5    200  HTTP       morttttq12.top   /index.php                        3                         nslookup:5064
6    302  HTTP       dowhhaa07.top   /download.php?file=lv.exe          0         text/html       nslookup:5064
7    200  HTTP       dowhhaa07.top   /downfiles/lv.exe          5,450,390         applicatio...   nslookup:5064
8    200  HTTP            ip-api.com   /json                            381         applicatio...   vpn_ico:1436
11   301  HTTPS              2no.co   /2CmjF5                            5  no-cac...  text/html;...  vpn_ico:1436
12   200  HTTP            ip-api.com   /line                            240         text/plain;...  6_ico:1372
13   302  HTTP     chrome-booster.com   /lib/download.php                 0  no-cac...  text/html;...  vpn_ico:1436
14   200  HTTP         45.147.230.58   /palata.exe                4,577,280         applicatio...   vpn_ico:1436
16   200  HTTPS          iplogger.org   /1jABs7                          127  no-cac...  image/png       wscript:5148
```

Figure 10: Network traffic resulting from running “setup_x86_x64_install.exe”

A brief description of this traffic and the malware components is below, but we have not conducted a full analysis of the files.

1. Stage 1: drops and runs a stealer component and downloads stage 2

1.a. Stealer: the first two network requests, hxxp[:]//eressedn27[.]top/index.php and hxxp://morttttq12[.]top/index.php, belong to the stealer component. The stealer collects and uploads a zip with information about the infected machine, including:

- Browser's information: saved username and password values, saved forms, credit card-related information, and cookies—from browsers including Chrome, Brave, Vivaldi, Opera, Avast, Firefox

- Screenshot: screenshot of the Desktop

- System Information: Operating System, language, keyboard languages, local time, username, CPU, RAM, video card, display resolution, installed software

- Cryptocurrency / wallets: we observed strings (but did not confirm exact functionality) related to cryptocurrency wallets and exchanges such as: Coinomi, waves-exchange, Ledger Live, Electrum, Electron Cash, Jaxx, Exodus, MultiBitHD, and Atomic.

1.b. Downloads stage 2: the following network request is a download of the next stage, “lv.exe”

2. Stage 2: drops a miner and downloads DanaBot

2.a. Miner: a file is dropped (not analyzed) that appears to be an AutoIT cryptocurrency miner

2.b. Download DanaBot: DanaBot is downloaded from hxxp[:]//45.147.230[.]58/palata.exe

Research performed by CSIS (Center for Strategic and International Studies) appears consistent with the same actor that Proofpoint researchers found. CSIS described a different malware bundle that may include AZORult, Predator the Thief, Smoke Loader, Redline Stealer, Amadey, Ficker Stealer, and Raccoon Stealer. However, this is likely due to the involvement of a Traffic Direction System (TDS) serving different payloads depending on factors such as geographic location of the victim. The Indicators of Compromise (IOC) reported by CSIS included the domain chrome-booster[.]com which Proofpoint also observed in our network traffic screenshot (above) leading to the download of DanaBot via a 302 redirect. Finally, the CSIS research described the number of infections in hundreds of thousands in a span of approximately 1 month. The infections focused on quantity instead of quality and ranged across many countries including United States, Canada, India, Turkey, Brazil, and others.

Conclusion

For almost two years, DanaBot was one of the top banking malwares being used in the crimeware threat landscape. Multiple threat actors were distributing and using it to target financials in many countries. In the middle of 2020, DanaBot activity dropped off. Some of the affiliates that were using it have continued their campaigns using other banking malware (e.g. Ursnif and Zloader). It is unclear whether COVID-19, competition from other banking malware, redevelopment time, or something else caused the dip, but it looks like DanaBot is back and trying to regain its foothold in the threat landscape. We assess the number of DanaBot affiliates will grow and that DanaBot will once again be distributed via phishing campaigns within the next few months.

Indicators of Compromise

Indicator	Type	Notes
c0eb802f394e758da4feb0d6c3b817bf1f64880ab9bc851937d5ef774161585d	SHA-256	Analyzed sample, affiliate ID 3
23.226.132.92	IP Address	C2 of analyzed sample
23.106.123.249	IP Address	C2 of analyzed sample
108.62.141.152	IP Address	C2 of analyzed sample
104.144.64.163	IP Address	C2 of analyzed sample
5jjsgjephjcua63go2o5donzw5x4hiwn6wh2dennmyq65pbhk6qflzyd\.onion	Hostname	TOR C2 of analyzed sample
83a67ecd166b919255b264718993c284a3238971a24c939c45e0c525f3361a43	SHA-256	Affiliate ID 21
149.129.212.179	IP Address	C2 of affiliate ID 21 sample
47.254.247.133	IP Address	C2 of affiliate ID 21 sample
159.89.114.62	IP Address	C2 of affiliate ID 21 sample
138.197.139.56	IP Address	C2 of affiliate ID 21 sample
ab3c72aaacbe2c99646bf4d91e177585631b164f8cd9e9e5eb7a180ce7d945d5	SHA-256	600117809bae5__Adobe-Photoshop-CC-2211138-Crack-Incl-Keygen-X64-2021.zip

ceb0ad27aaf97a5a33664f49aa107ca421c3f0a6e0b9a3c37f93455a258f3c04	SHA-256	DanaBot downloaded from hxxp[:]//45.147.230[.]58/palata.exe

Emerging Threats Signatures

ETPRO TROJAN Danabot Key Exchange Request

ETPRO TROJAN Danabot Command Beacon Request

Is your institution protected against malware attacks? Learn about Cybersecurity for Financial Services Firms.

Subscribe to the Proofpoint Blog