DanaBot updated with new C&C communication

welivesecurity.com/2019/02/07/danabot-updated-new-cc-communication/

February 7, 2019



ESET researchers have discovered new versions of the DanaBot Trojan, updated with a more complicated protocol for C&C communication and slight modifications to architecture and campaign IDs



ESET Research 7 Feb 2019 - 12:00PM

ESET researchers have discovered new versions of the DanaBot Trojan, updated with a more complicated protocol for C&C communication and slight modifications to architecture and campaign IDs

The fast-evolving, modular Trojan <u>DanaBot</u> has undergone further changes, with the latest version featuring an entirely new communication protocol. The protocol, introduced to DanaBot at the end of January 2019, adds several layers of encryption to DanaBot's <u>C&C</u> communication.

Besides the changes in communication, DanaBot's architecture and campaign IDs have also been modified.

The evolution of DanaBot

After being <u>discovered</u> in May 2018 as part of Australia-targeted spam campaigns, DanaBot has had an eventful time since, appearing in malspam campaigns in Poland, <u>Italy, Germany, Austria and Ukraine</u>, as well as in <u>the United States</u>. The European campaigns have seen the Trojan expanding its capabilities with new plugins and <u>spam-sending features</u>.

In ESET telemetry on January 25, 2019, we noticed unusual DanaBot-related executables. Upon further inspection, these binaries were, indeed, revealed to be DanaBot variants, but using a different communication protocol to communicate with the C&C server. Starting January 26, 2019, DanaBot operators stopped building binaries with the old protocol.

At the time of writing, the new version is being distributed under two scenarios:

- As "updates" delivered to existing DanaBot victims
- Via malspam in Poland

The new communication protocol

In the communication protocol used before January 25, packets were not encrypted in any way, as seen in Figure 1.

| Wireshark | Foll | ow ' | ГСР | Stre | am | (tcp | stre | am e | eq 4) | • tcp | odur | np.p | сар | | | | |
|-----------|------|------|-----|------|----|------|------|------|-------|-------|------|------|-----|----|----|----|-------------------|
| | | | | | | | | | | | | | | | | | |
| 00000000 | 00 | 00 | 00 | 00 | ff | ff | ff | ff | 53 | 04 | 00 | 00 | 03 | 00 | 00 | 00 | S |
| 00000010 | 01 | 00 | 00 | 00 | 28 | 1e | 00 | 00 | 00 | 00 | 00 | 00 | 20 | 00 | 00 | 00 | (|
| 00000020 | ae | 75 | 67 | 24 | d1 | 07 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 40 | 00 | 00 | .ug\$@ |
| 00000030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 20 | 41 | 31 | 43 | 46 | 44 | 32 | 31 | A1CFD21 |
| 00000040 | 43 | 46 | 33 | 35 | 33 | 45 | 43 | 30 | 41 | 33 | 44 | 36 | 34 | 36 | 35 | 46 | CF353EC0 A3D6465F |
| 00000050 | 36 | 45 | 37 | 35 | 35 | 35 | 46 | 31 | 30 | 20 | 34 | 45 | 42 | 38 | 32 | 46 | 6E7555F1 0 4EB82F |
| 00000060 | 38 | 39 | 31 | 42 | 33 | 34 | 34 | 34 | 37 | 34 | 45 | 39 | 39 | 43 | 41 | 42 | 891B3444 74E99CAB |
| 00000070 | 35 | 39 | 35 | 35 | 46 | 30 | 37 | 35 | 44 | 37 | 20 | 37 | 46 | 43 | 35 | 36 | 5955F075 D7 7FC56 |
| 0800000 | 32 | 37 | 30 | 45 | 37 | 41 | 37 | 30 | 46 | 41 | 38 | 31 | 41 | 35 | 39 | 33 | 270E7A70 FA81A593 |
| 00000090 | 35 | 42 | 37 | 32 | 45 | 41 | 43 | 42 | 45 | 32 | 39 | 00 | 00 | 00 | 00 | 00 | 5B72EACB E29 |
| 000000A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 000000B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | | | | | | | | | | |

Figure 1 – Packet capture showing the old protocol with data in plaintext

Following the latest changes, DanaBot uses the AES and RSA encryption algorithms in its C&C communication. The new communication protocol is complicated, with several encryption layers being used, as seen in Figure 2.



Figure 2 – A diagram of DanaBot's new communication protocol

These changes break existing network-based signatures and make it more difficult to write new rules for Intrusion Detection and Prevention Systems. Also, without access to the corresponding RSA keys, it is impossible to decode sent or received packets; thus PCAP files from cloud-based analysis systems (such as <u>ANY.RUN</u>) become unusable for researchers.

| 📕 Wireshark · | Follow TCP Stream (tcp.stream eq 0) · downloader.pcap | |
|----------------------------------|--|-------------------------------|
| 00000000 00000010 | 24 01 00 00 00 00 00 00 65 a4 00 00 00 00 00 00 00 89 a5 00 00 00 00 00 00 00 00 | \$ e |
| 00000018 00000028 | cd 0b 66 ff 63 2e c8 ff fa d5 df c2 84 bc 3d 7c 32 eb 50 3b f0 bb 47 34 85 3e c1 24 71 2b 14 6d | f.c= 2.P;G4 .>.\$q+.m |
| 00000038 00000048 | eb 98 8a c8 15 72 29 a3 dd ff 57 db e9 71 e1 6b ce 5f b9 6e 52 f2 2a a5 e7 43 5e d3 dd fa c4 da 2f c6 62 35 70 b3 50 db c0 d5 05 82 23 26 82 fa | r)Wq.k nR.*C^ |
| 00000068 | b8 02 fa 73 ee 72 c9 6f dc 0c 12 50 93 eb 22 01 36 ab 83 a0 52 15 9f db a2 be 9a 9a a1 ce c1 e3 | s.r.oP". 6R |
| 00000088 00000098 | 62 b5 4b 42 dd 3d 52 dc e5 35 aa 23 b9 8f dd 10 38 41 41 f0 7c da ca a7 f0 7d 84 bf ed 0c 31 c6 | b.KB.=R5.# 8AA |
| 000000A8 000000B8 000000C8 | 32 bd 00 74 0d 9c 0f 18 64 7f b7 94 4c 4a fb 1d 0c 00 00 00 9e 03 f1 b5 c0 e5 9d 40 e5 5e 7f fd e1 1b e5 9f c3 d5 77 37 3d 91 ed 0c 4a e7 80 21 | 2t dLJ |
| 000000D8 000000E8 | 50 f6 01 17 0f c8 c4 12 50 48 75 54 a6 43 6a 5c 05 6a 6a 35 3f bd 90 df 8d 9a 03 ba bf c5 b6 4e | P PHuT.Cj\ .jj5?N |
| 000000F8 00000108 | 54 4e 5b b0 d2 95 12 11 db e4 08 1b 73 4b e8 b6 3b 9a 30 ba 20 86 f5 e7 9f 63 41 c5 f9 1c de f8 | TN[sK ;.0cA |
| 00000128 00000138 | 94 91 10 91 70 85 86 45 74 81 80 96 10 80 93 9d 54 b1 28 b1 e8 eb f5 87 d9 e5 f6 c4 5b 16 56 6c d5 7b 98 34 01 00 00 00 00 12 2c 00 00 | 4pe. et1. .T.([.V 1.{.4 |
| 00000148 00000158 | 00 00 00 00 46 2d 00 00 00 00 00 00 68 69 56 6b fb 32 47 ff 2a 64 80 d5 be 02 ba a6 1b fd 68 ea | FhiVk .2G.*dh. |
| 00000168 00000178 | 58 8a 61 85 8d 3e 98 42 3f 88 eb c9 1c 37 f2 65 bc ad ca 2c 6c 38 be da b9 63 1c 30 45 36 87 dd | X.a>.B ?7.e ,18c.0E6 |
| _ | small packet header packet payload, padding s | ize AES key, RSA encrypted |

Figure 3 – Packet capture with the new communication protocol in place

Each packet sent by the client has a 24 (0x18)-byte header:

| Offset | Size (bytes) | Meaning |
|--------|--------------|------------------------------------|
| 0x0 | 0x8 | Size of the data after this header |
| 0x8 | 0x8 | Random value |
| 0x10 | 0x8 | Sum of first two fields |

For each packet, the header is followed by AES-encrypted packet data, then a 4-byte value indicating AES padding size, and finally the RSA-encrypted AES key. Each packet is encrypted with a different AES key.

Server responses use the same format. Unlike in previous versions, packet data in server responses does not follow any specific layout (with some exceptions).

Packet data layout

Former packet data layout was detailed by <u>Proofpoint</u> in October 2018. In the latest version of DanaBot, the layout is slightly modified, as seen in Figure 4.

Previous layout

| Offset | Size (bytes) | Meaning | | | | | | |
|---------------------------------|------------------|-----------------------------|--|--|--|--|--|--|
| 0x0 | 0x4 | Random values (stack junk) | | | | | | |
| 0x4 | 0x4 | Hardcoded -1 | | | | | | |
| 0x8 | 0x4 | Command ID | | | | | | |
| 0xC | 0x4 | Campaign ID | | | | | | |
| 0x10 | 0x4 | Hardcoded 1 | | | | | | |
| 0x14 | 0x4 | Random value | | | | | | |
| 0x18 | 0x4 | Unknown counter variable | | | | | | |
| Ox1C | 0x4 | System architecture | | | | | | |
| 0x20 | 0x4 | Windows version information | | | | | | |
| 0x24 | 0x4 | Command parameter (0/32/64) | | | | | | |
| 0x28 | 0x4 | Admin status | | | | | | |
| 0x2C | 0x4 | Process integrity level | | | | | | |
| 0x30 | 0x8 | Payload length | | | | | | |
| 0x38 | 0x21 | Client ID | | | | | | |
| 0x59 | 0x21 | Command dependent | | | | | | |
| 0x7A | 0x21 | Checksum | | | | | | |
| Ox9B | 0x1C | Junk | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| Legend: | | | | | | | | |
| different field | | | | | | | | |
| same field | in a different p | osition | | | | | | |
| same field in the same position | | | | | | | | |

| Offset | Size (bytes) | Meaning |
|--------|--------------|----------------------------------|
| 0x0 | 0x4 | Size of the packet header (0xA7) |
| 0x4 | 0x8 | Random value |
| 0xC | 0x8 | Sum of first 2 fields |
| 0x14 | 0x4 | Campaign ID |
| 0x18 | 0x4 | Command ID |
| 0x1C | 0x4 | Command parameter (0/32/64) |
| 0x20 | 0x4 | Random value |
| 0x24 | 0x4 | Unknown counter variable |
| 0x28 | 0x4 | System architecture |
| 0x2C | Ox4 | Windows version information |
| 0x30 | 0x4 | Command dependent (0/0x3E9) |
| 0x34 | 0x4 | Admin status |
| 0x38 | 0x4 | Process integrity level |
| 0x3C | 0x8 | Payload length |
| 0x44 | 0x21 | Client ID |
| 0x65 | 0x21 | Command dependent |
| 0x86 | 0x21 | Checksum |

New layout

Figure 4 – Comparison of packet data layout in DanaBot's previous and latest version

Changes in DanaBot architecture

Besides the changed communication protocol, DanaBot has also undergone some changes in architecture. The previous versions of DanaBot included a component that downloaded and executed the main module. The main module then downloaded and executed plugins and configurations.

The latest version shifts both these responsibilities to a new loader component, which is used to download all plugins along with the main module. Persistence is achieved by registering the loader component as a service.



Figure 5 – Comparison of architecture in DanaBot's previous and latest version

Commands

According to our analysis, the loader component uses the following commands:

- 0x12C Hello. First command sent by client to server
- 0x12D Download 32/64-bit launcher component
- 0x12E Request list of plugins and configuration files
- 0x12F Download plugin/configuration files

Downloaded plugins and configuration files are encrypted using an AES key derived from the Client ID. In addition to that, plugins are compressed in ZIP format using LZMA compression, whereas configuration files are compressed using zlib.

Commands with ID numbers 0x130 – 0x134 are sent by the main module:

- 0x130 Upload collected information to C&C server (e.g., screenshot of a victim's computer; system information)
- 0x131 Upload collected information to C&C server (e.g., list of files on the victim's hard disk)
- 0x132 Ask C&C server for further commands; there are around 30 available commands typical
 of backdoors, including launching plugins, gathering detailed system information and modifying
 files on client system
- 0x133 Update C&C server list via Tor proxy
- 0x134 Exact purpose unknown; most likely used for communication between plugins and C&C

Changes in campaign IDs

Previous research has suggested that DanaBot is distributed under various "affiliate" or "campaign" IDs.

In the previous version of DanaBot, almost <u>20 different campaign IDs</u> were used. In the latest version, campaign IDs have changed slightly. As of February 5, 2019, we are seeing the following IDs in the wild:

- **ID=2** appears to be a test version, serving a limited number of configuration files and no webinjects
- **ID=3** is being actively spread, targeting users in both Poland and Italy, serving all configuration files and webinjects for both Polish and Italian targets
- ID=5 serves configuration files for Australian targets
- ID=7 is being spread only in Poland, serving webinjects for Polish targets
- **ID=9** appears to be another test version, with limited spread and no specific targeting, serving a limited number of configuration files and no webinjects

Conclusion

In 2018, we observed DanaBot expanding in both <u>distribution</u> and <u>functionality</u>. The beginning of 2019 has seen the Trojan undergo "internal" changes, indicating active development by its authors. The latest updates suggest the authors are making an effort to evade detection at the network level, and possibly paying attention to published research and making changes to stay ahead of defenders.

ESET systems detect and block all DanaBot components and plugins under detection names listed in the IoCs section.

This research was carried out by Kaspars Osis, Tomáš Procházka and Michal Kolář.

C&C servers used by the new version of DanaBot

- 84.54.37[.]102
- 89.144.25[.]243
- 89.144.25[.]104
- 178.209.51[.]211
- 185.92.222[.]238
- 192.71.249[.]51

Webinject and redirect servers

- 47.74.249[.]106
- 95.179.227[.]160
- 185.158.249[.]144

Example hashes

Note that since new builds of DanaBot's components are released regularly, we provide just a sampling of hashes.

| Component | SHA-1 | ESET detection name |
|--------------------------------------|--|-------------------------------|
| Dropper | 98C70361EA611BA33EE3A79816A88B2500ED7844 | Win32/TrojanDropper.Danabot.O |
| Loader (x86), campaign ID=3 | 0DF17562844B7A0A0170C9830921C3442D59C73C | Win32/Spy.Danabot.L |
| Loader (x64), campaign ID=3 | B816E90E9B71C85539EA3BB897E4F234A0422F85 | Win64/Spy.Danabot.G |
| Loader (x86), campaign ID=9 | 5F085B19657D2511A89F3172B7887CE29FC70792 | Win32/Spy.Danabot.I |
| Loader (x64), campaign ID=9 | 4075375A08273E65C223116ECD2CEF903BA97B1E | Win64/Spy.Danabot.F |
| Main module (x86) | 28139782562B0E4CAB7F7885ECA75DFCA5E1D570 | Win32/Spy.Danabot.K |
| Main module (x64) | B1FF7285B49F36FE8D65E7B896FCCDB1618EAA4B | Win64/Spy.Danabot.C |

Plugins

| Plugin | SHA-1 | ESET detection name |
|---------------|--|---------------------|
| RDPWrap | 890B5473B419057F89802E0B6DA011B315F3EF94 | Win32/Spy.Danabot.H |
| Stealer (x86) | E50A03D12DDAC6EA626718286650B9BB858B2E69 | Win32/Spy.Danabot.C |
| Stealer (x64) | 9B0EC454401023DF6D3D4903735301BA669AADD1 | Win64/Spy.Danabot.E |
| Sniffer | DBFD8553C66275694FC4B32F9DF16ADEA74145E6 | Win32/Spy.Danabot.B |
| VNC | E0880DCFCB1724790DFEB7DFE01A5D54B33D80B6 | Win32/Spy.Danabot.D |
| TOR | 73A5B0BEE8C9FB4703A206608ED277A06AA1E384 | Win32/Spy.Danabot.G |

7 Feb 2019 - 12:00PM

Sign up to receive an email update whenever a new article is published in our <u>Ukraine</u> <u>Crisis – Digital Security Resource Center</u>

Newsletter

Discussion