Goodbye Mworm, Hello Nworm: TrickBot Updates Propagation Module

unit42.paloaltonetworks.com/goodbye-mworm-hello-nworm-trickbot-updates-propagation-module/

Brad Duncan

May 28, 2020

By Brad Duncan

May 28, 2020 at 12:00 PM

Category: Malware, Unit 42

Tags: Mworm, Nworm, Trickbot

This post is also available in: 日本語 (Japanese)

Executive Summary

First discovered in 2016, TrickBot is an information stealer that provides backdoor access sometimes used by criminal groups to distribute other malware. TrickBot uses modules to perform different functions, and one key function is propagating from an infected Windows client to a vulnerable Domain Controller (DC). TrickBot currently uses three modules for propagation. As early as April 2020, TrickBot updated one of its propagation modules known as "mworm" to a new module called "nworm." Infections caused through nworm leave no artifacts on an infected DC, and they disappear after a reboot or shutdown.

Other key differences of the new nworm module include:

- It retrieves an encrypted, or otherwise encoded binary, over network traffic that represents a TrickBot executable file (the old mworm module sent it as an executable file without any sort of encryption/encoding).
- A TrickBot infection caused by the new mworm module is run from system RAM and does not appear to remain persistent on an infected host.
- This is a much better method of evading detection on an infected DC.

TrickBot is a significant threat that has received high-profile coverage in recent years, and this is a notable evolution. This blog reviews TrickBot modules, and it covers characteristics of the new nworm module in greater detail.

TrickBot Modules

TrickBot is modular, meaning it uses various binaries to perform different functions during an infection. In most cases, the basis of a TrickBot infection is a malicious Windows executable (EXE) file saved to disk. This EXE is often called a "TrickBot loader" because it loads the TrickBot modules. TrickBot modules are dynamic link libraries (DLLs) or EXEs run from system memory. See Figure 1 for a visualization of TrickBot modules.

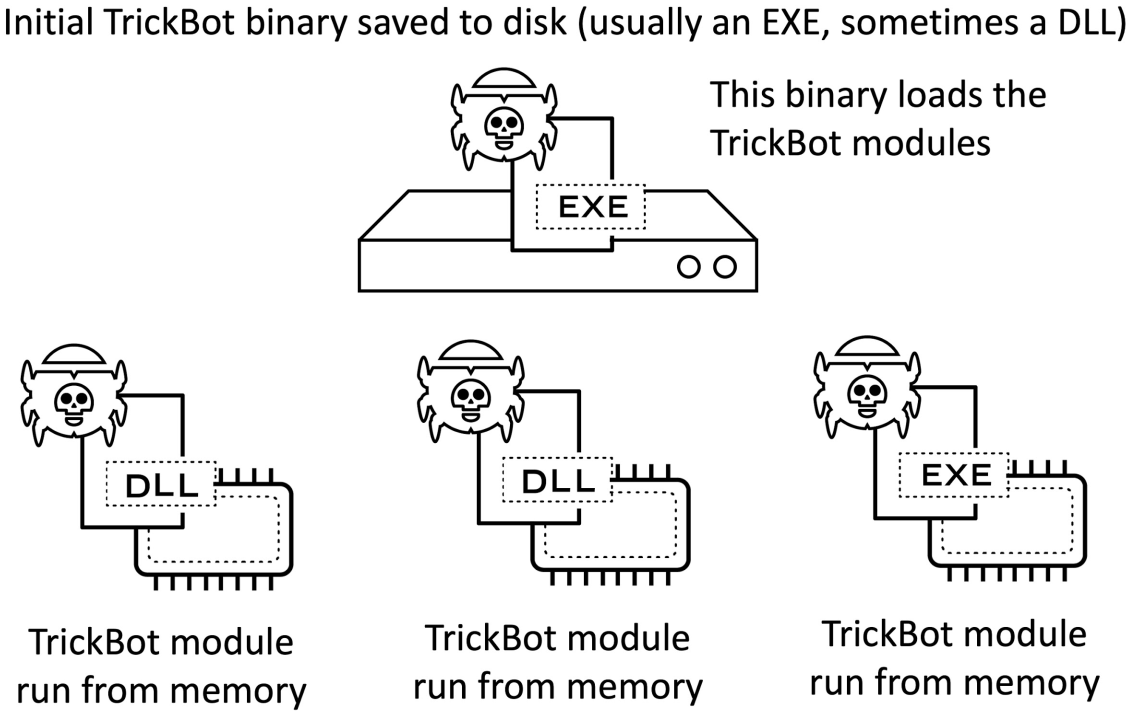
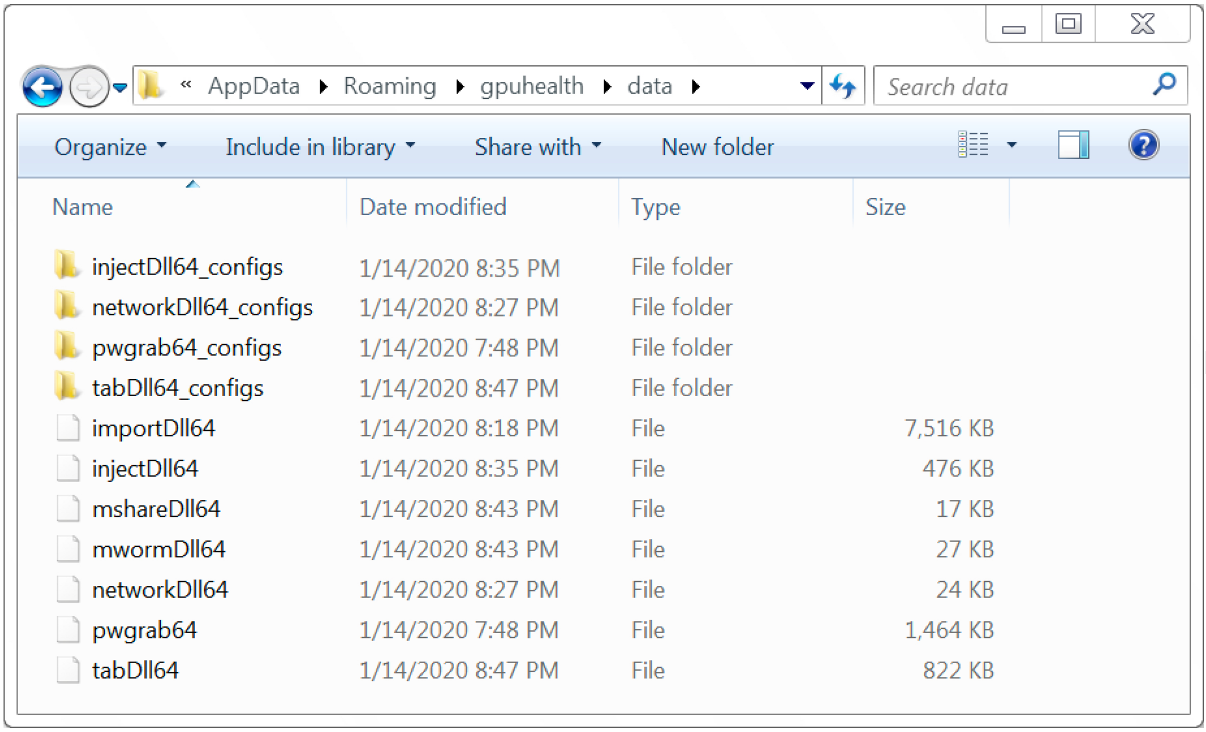

Initial TrickBot binary saved to disk (usually an EXE, sometimes a DLL)
This binary loads the TrickBot modules
EXE
DLL
DLL
EXE
TrickBot module run from memory
TrickBot module run from memory
TrickBot module run from memory

Figure 1. A visual representation of TrickBot and its modules.

On an infected Windows 10 host, TrickBot modules are only found in system memory. But on an infected Windows 7 host, we also see artifacts related to the modules stored on the disk. These artifacts are encrypted binaries. During a TrickBot infection, these encrypted binaries are decrypted and run from system memory as TrickBot modules. Figure 2 shows an example of artifacts for TrickBot modules from an infection on a Windows 7 client in January 2020.



Name	Date modified	Type	Size
injectDll64_configs	1/14/2020 8:35 PM	File folder	
networkDll64_configs	1/14/2020 8:27 PM	File folder	
pwgrab64_configs	1/14/2020 7:48 PM	File folder	
tabDll64_configs	1/14/2020 8:47 PM	File folder	
importDll64	1/14/2020 8:18 PM	File	7,516 KB
injectDll64	1/14/2020 8:35 PM	File	476 KB
mshareDll64	1/14/2020 8:43 PM	File	17 KB
mwormDll64	1/14/2020 8:43 PM	File	27 KB
networkDll64	1/14/2020 8:27 PM	File	24 KB
pwgrab64	1/14/2020 7:48 PM	File	1,464 KB
tabDll64	1/14/2020 8:47 PM	File	822 KB

Figure 2. Example of artifacts for TrickBot modules on an infected Windows 7 client.

As seen in Figure 2, the artifact names end with 64, meaning this host is running a 64-bit version of Windows 7. If the infection happens on a 32-bit Windows 7 host, these artifact names would end in 32 instead of 64.

Figure 2 also reveals three modules TrickBot uses to spread to a DC in an Active Directory (AD) environment. They are:

- mwormDll64 (the "mworm" module)
- mshareDll64 (the "mshare" module)
- tabDll64 (the "tab" module)

Note: The tab module has a propagation function, but it also includes different capabilities not applicable to this blog.

Modules for Propagation

Starting in September 2019, TrickBot modules with propagation capabilities have been mworm, mshare, and tab. They generate distinct activity when propagating to a vulnerable DC.

For the mshare and tab modules:

- An infected Windows client retrieves a new TrickBot EXE using an HTTP URL.
- The infected Windows client sends this new TrickBot EXE over SMB traffic to the vulnerable DC.

For the mworm module:

- The infected Windows client uses an SMB exploit targeting the vulnerable DC.
- The vulnerable DC retrieves a new TrickBot EXE using an HTTP URL and infects itself with it.

Of note, the mworm module did not usually appear unless the TrickBot infection happened in an AD environment with a DC.

Figure 3 shows a flow chart of propagation traffic caused by these three TrickBot modules.

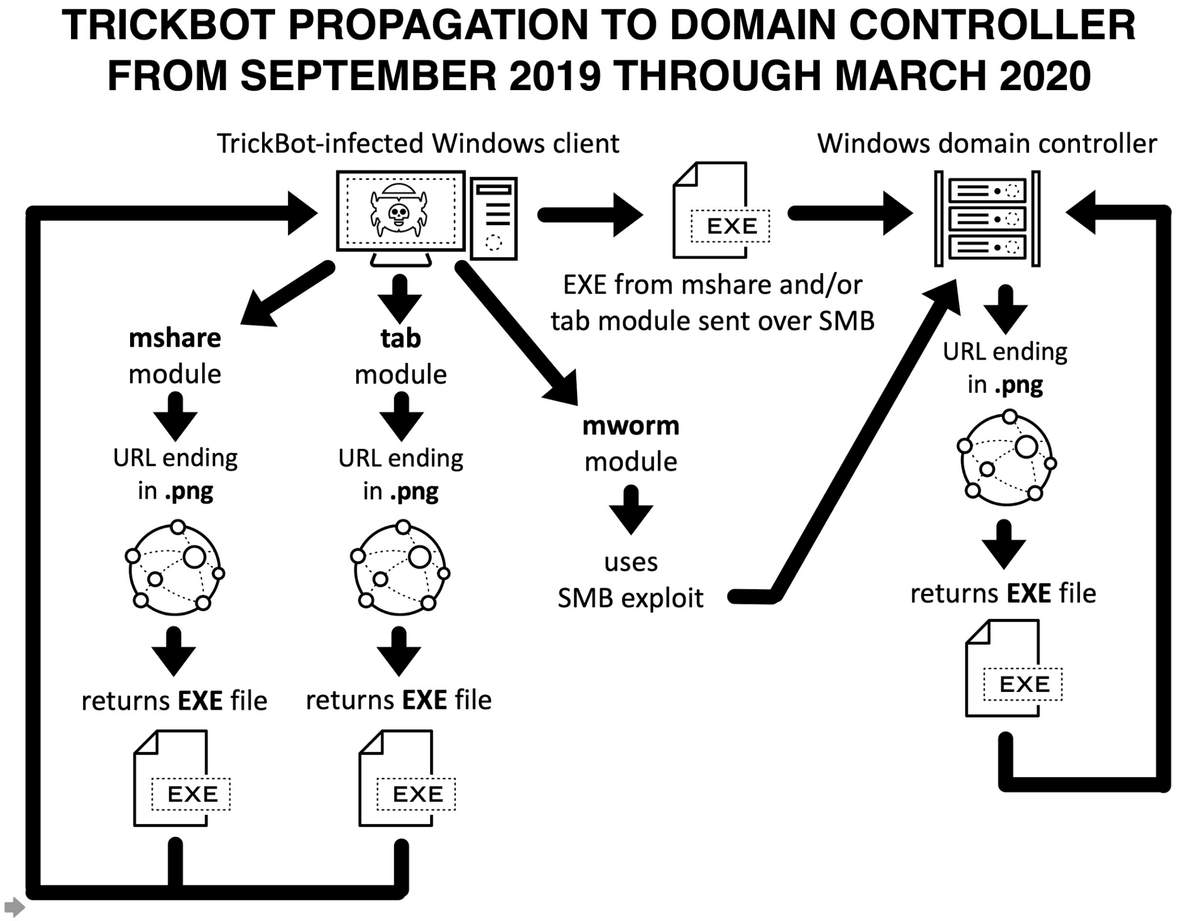
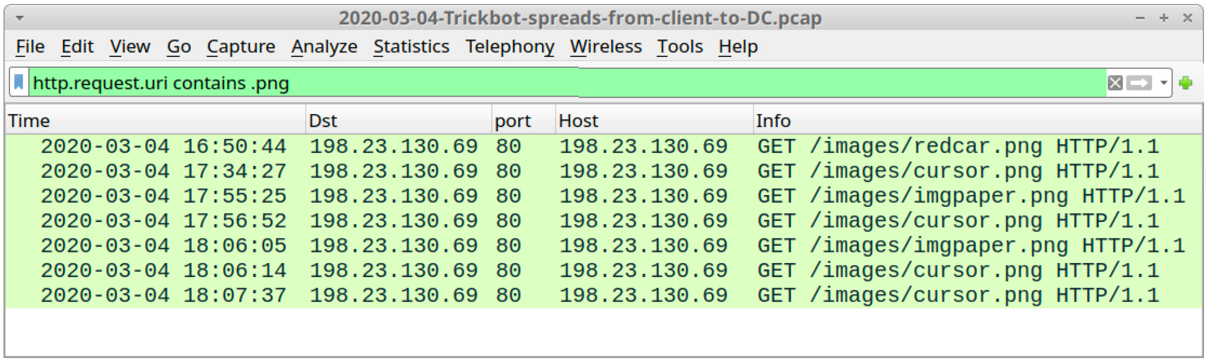

TRICKBOT PROPAGATION TO DOMAIN CONTROLLER
FROM SEPTEMBER 2019 THROUGH MARCH 2020
TrickBot-infected Windows client
Windows domain controller
EXE
EXE from mshare and/or tab module sent over SMB
mshare module
tab module
mworm module
URL ending in .png
URL ending in .png
uses SMB exploit
URL ending in .png
returns EXE file
returns EXE file
returns EXE file
EXE
EXE
EXE

Figure 3. TrickBot propagation flow chart from September 2019 through March 2020.

Since February 2020, URLs generated by these modules to retrieve follow-up TrickBot EXE files used the following patterns:

- URL generated by mshare module ends with /images/cursor.png
- URL generated by mworm module ends with /images/redcar.png
- URL generated by tab module ends with /images/imgpaper.png

These URLs use IP addresses instead of domains. Figure 4 shows an example of the traffic filtered in Wireshark from a pcap of a TrickBot infection in March 2020.


2020-03-04-Trickbot-spreads-from-client-to-DC.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request.uri contains .png

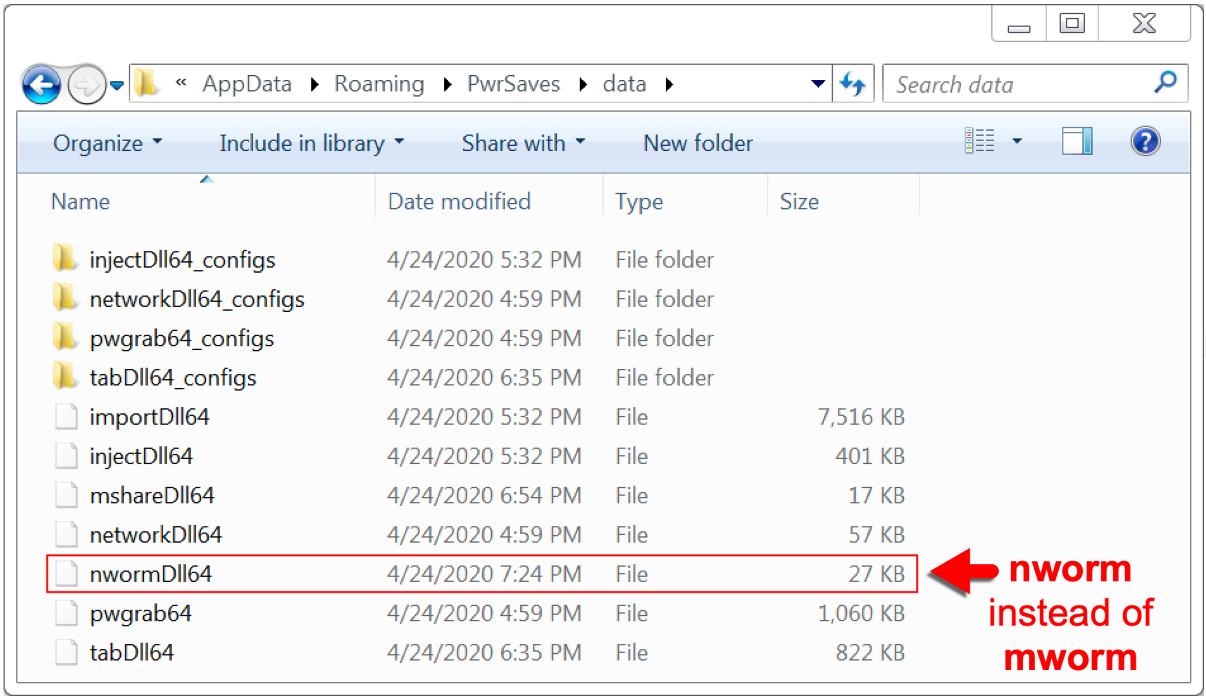
```
Time                 Dst            port Host           Info
2020-03-04 16:50:44 198.23.130.69  80   198.23.130.69  GET /images/redcar.png HTTP/1.1
2020-03-04 17:34:27 198.23.130.69  80   198.23.130.69  GET /images/cursor.png HTTP/1.1
2020-03-04 17:55:25 198.23.130.69  80   198.23.130.69  GET /images/imgpaper.png HTTP/1.1
2020-03-04 17:56:52 198.23.130.69  80   198.23.130.69  GET /images/cursor.png HTTP/1.1
2020-03-04 18:06:05 198.23.130.69  80   198.23.130.69  GET /images/imgpaper.png HTTP/1.1
2020-03-04 18:06:14 198.23.130.69  80   198.23.130.69  GET /images/cursor.png HTTP/1.1
2020-03-04 18:07:37 198.23.130.69  80   198.23.130.69  GET /images/cursor.png HTTP/1.1
```

Figure

4. HTTP GET requests caused by TrickBot’s mshare, mworm and tab modules.

Goodbye Mworm: Hello Nworm

In April 2020 while generating a TrickBot infection in a lab environment, TrickBot stopped using the mworm module. In its place, a new artifact named "nworm" appeared on an infected Windows 7 client. Figure 5 shows an example of this new nworm artifact.



Name	Date modified	Type	Size
injectDll64_configs	4/24/2020 5:32 PM	File folder	
networkDll64_configs	4/24/2020 4:59 PM	File folder	
pwgrab64_configs	4/24/2020 4:59 PM	File folder	
tabDll64_configs	4/24/2020 6:35 PM	File folder	
importDll64	4/24/2020 5:32 PM	File	7,516 KB
injectDll64	4/24/2020 5:32 PM	File	401 KB
mshareDll64	4/24/2020 6:54 PM	File	17 KB
networkDll64	4/24/2020 4:59 PM	File	57 KB
nwormDll64	4/24/2020 7:24 PM	File	27 KB
pwgrab64	4/24/2020 4:59 PM	File	1,060 KB
tabDll64	4/24/2020 6:35 PM	File	822 KB

nworm instead of mworm

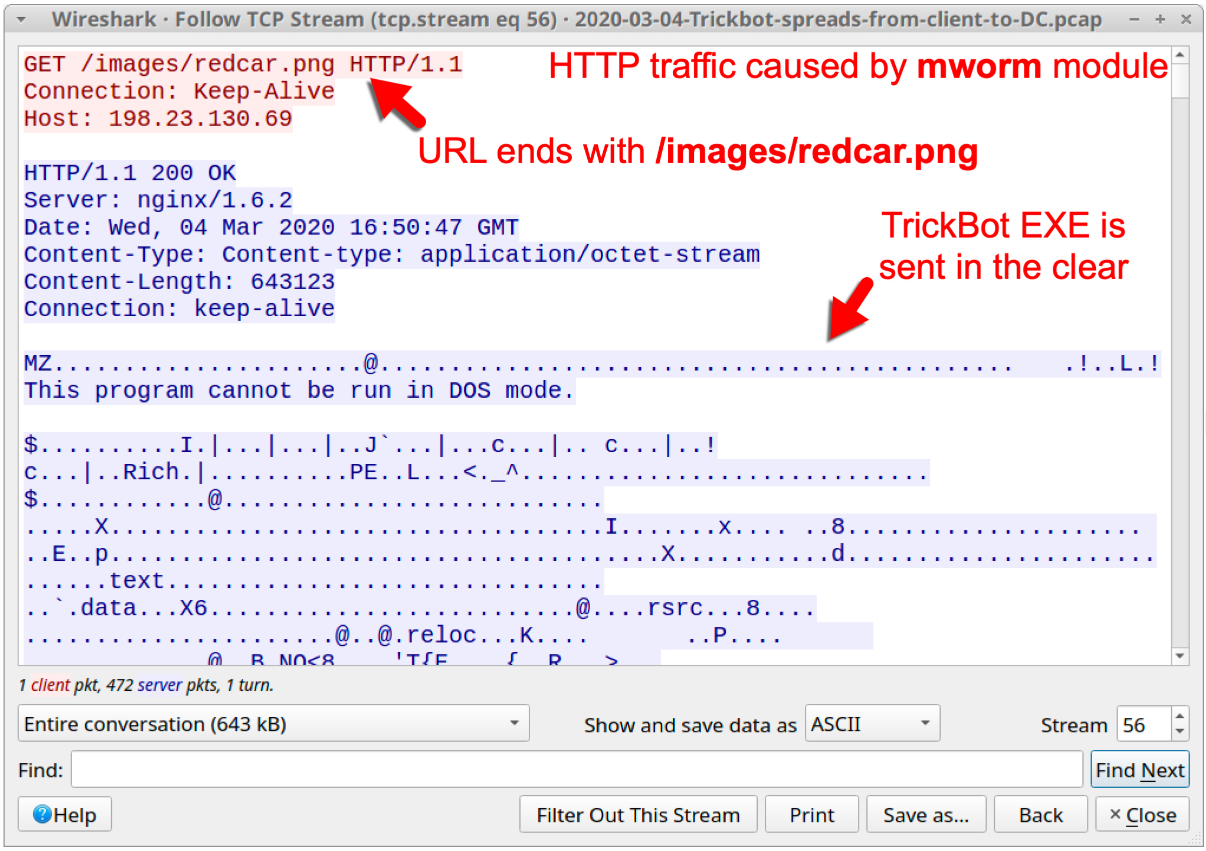
Figure

5. New nworm module found from an infection on April 24, 2020.

HTTP traffic for follow-up TrickBot EXEs caused by nworm is noticeably different than traffic caused by mworm. The differences are:

-
    - mworm: URL for TrickBot EXE ends with /images/redcar.png
    - nworm: URL for TrickBot EXE ends with /ico/VidT6cErs
-
    - mworm: Follow-up TrickBot EXE is returned unencrypted in the HTTP traffic
    - nworm: Follow-up TrickBot EXE is returned as an encrypted or otherwise encoded binary in the HTTP traffic

By using Wireshark and examining TCP streams, we can easily spot the differences in HTTP traffic caused by the old mworm module and the new nworm module. Figure 6 shows traffic from the mworm module in March 2020, and Figure 7 shows traffic from the nworm module in April 2020.



Wireshark · Follow TCP Stream (tcp.stream eq 56) · 2020-03-04-Trickbot-spreads-from-client-to-DC.pcap

```
GET /images/redcar.png HTTP/1.1
Connection: Keep-Alive
Host: 198.23.130.69

HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Wed, 04 Mar 2020 16:50:47 GMT
Content-Type: Content-type: application/octet-stream
Content-Length: 643123
Connection: keep-alive

MZ......................@......................................   .!..L.!
This program cannot be run in DOS mode.

$.........I.|...|...|..J`...|...c...|.. c...|..!
c...|..Rich.|..........PE..L...<._^..........................
$...........@.........................
.....X.................................I.......x....  ..8....................
..E..p.....................................X..........d....................
......text............................
..`.data...X6.........................@....rsrc...8....
......................@..@.reloc...K....       ..P....
            @  B.NQ<8     'T{F     {  R    >
```

HTTP traffic caused by **mworm** module

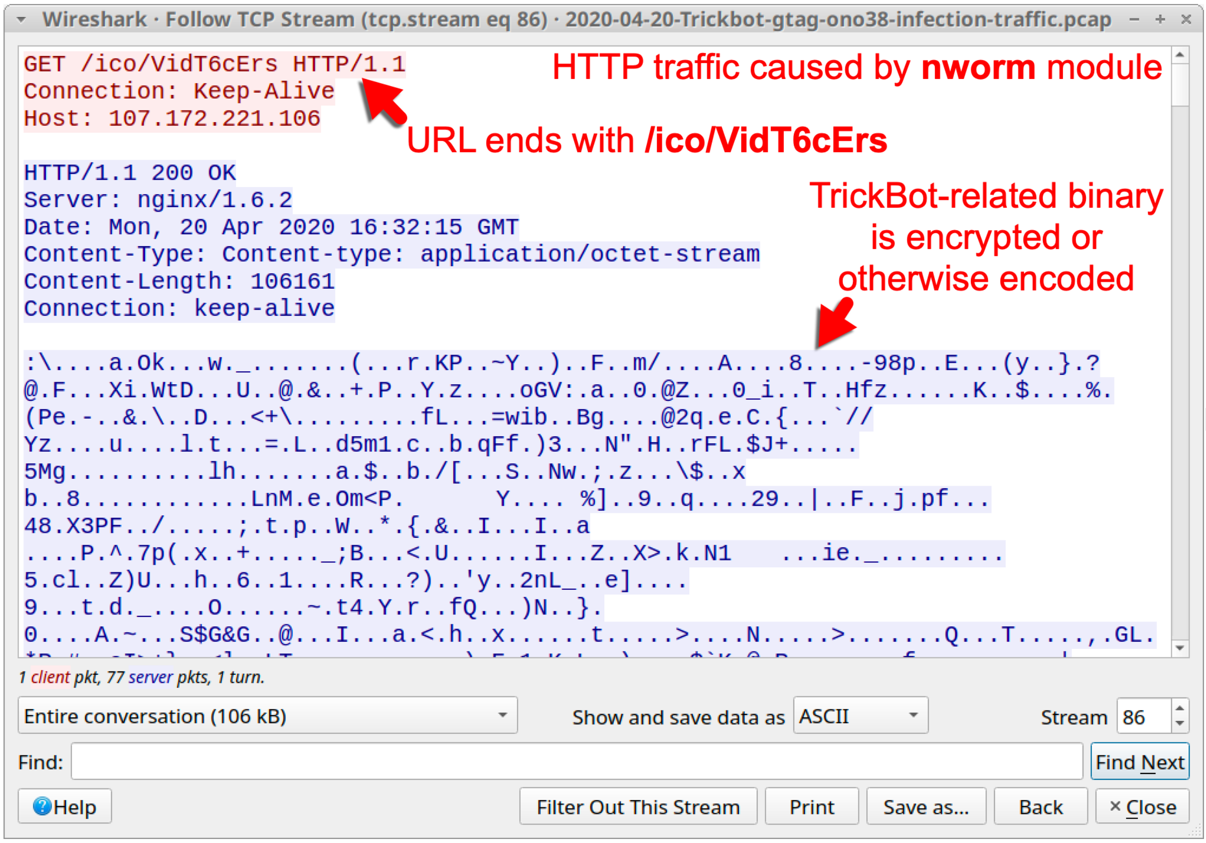
URL ends with **/images/redcar.png**

TrickBot EXE is sent in the clear

1 client pkt, 472 server pkts, 1 turn.

Entire conversation (643 kB) — Show and save data as: ASCII — Stream: 56

Figure 6. TCP stream showing HTTP traffic caused by the mworm module in March 2020.



Wireshark · Follow TCP Stream (tcp.stream eq 86) · 2020-04-20-Trickbot-gtag-ono38-infection-traffic.pcap
HTTP traffic caused by **nworm** module
URL ends with **/ico/VidT6cErs**
TrickBot-related binary is encrypted or otherwise encoded

```
GET /ico/VidT6cErs HTTP/1.1
Connection: Keep-Alive
Host: 107.172.221.106

HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Mon, 20 Apr 2020 16:32:15 GMT
Content-Type: Content-type: application/octet-stream
Content-Length: 106161
Connection: keep-alive

:\....a.Ok...w._.......(...r.KP..~Y..)..F..m/....A....8....-98p..E...(y..}.?
@.F...Xi.WtD...U..@.&..+.P..Y.z....oGV:.a..0.@Z...0_i..T..Hfz......K..$....%.
(Pe.-..&.\..D...<+\.........fL...=wib..Bg....@2q.e.C.{...`//
Yz....u....l.t...=.L..d5m1.c..b.qFf.)3...N".H..rFL.$J+.....
5Mg..........lh.......a.$..b./[...S..Nw.;.z...\$..x
b..8............LnM.e.Om<P.       Y.... %]..9..q....29..|..F..j.pf...
48.X3PF../.....;.t.p..W..*.{.&..I...I..a
....P.^.7p(.x..+....._;B...<.U......I...Z..X>.k.N1    ...ie._.........
5.cl..Z)U...h..6..1....R...?)..'y..2nL_..e]....
9...t.d._....O......~.t4.Y.r..fQ...)N..}.
0....A.~...S$G&G..@...I...a.<.h..x......t.....>....N.....>.......Q...T.....,.GL.
```

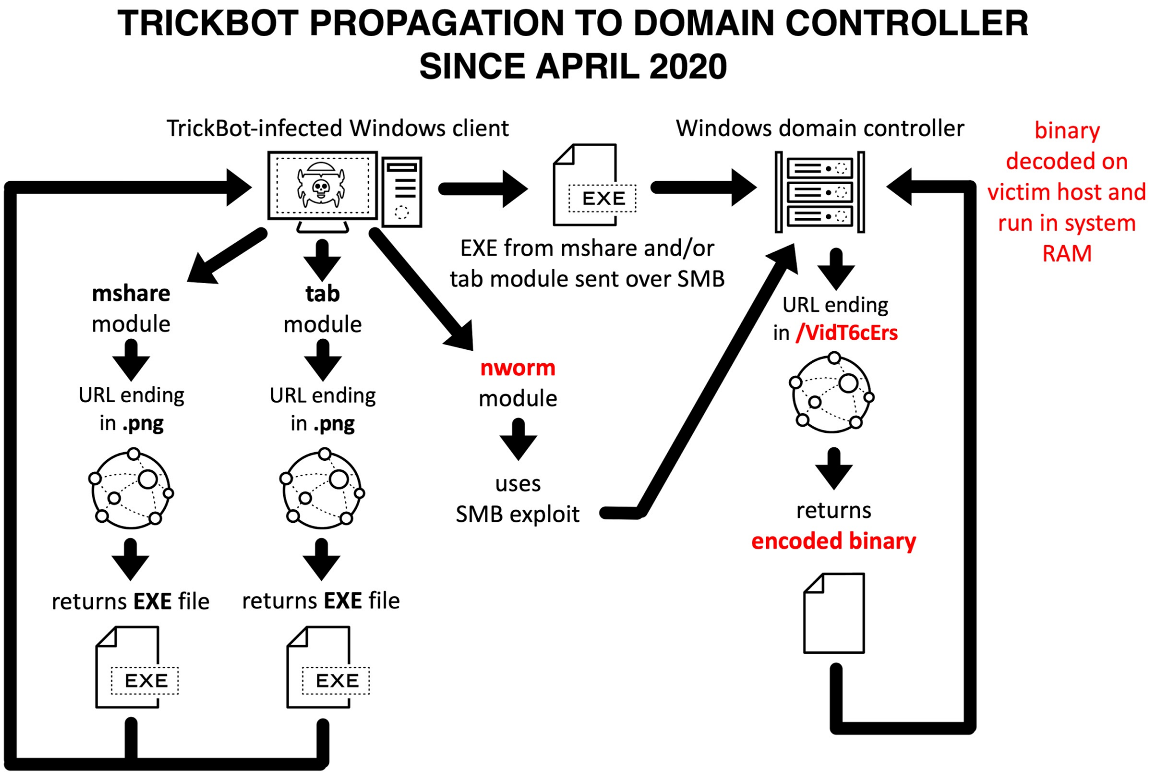
1 client pkt, 77 server pkts, 1 turn.

Entire conversation (106 kB) | Show and save data as: ASCII | Stream: 86

Figure

7. TCP stream showing HTTP traffic caused by the nworm module April 2020.

Figure 8 shows the current propagation flowchart, highlighting changes seen with the nworm module since April 2020.



TRICKBOT PROPAGATION TO DOMAIN CONTROLLER SINCE APRIL 2020
TrickBot-infected Windows client
Windows domain controller
EXE
EXE from mshare and/or tab module sent over SMB
binary decoded on victim host and run in system RAM
mshare module
tab module
nworm module
URL ending in **.png**
URL ending in **.png**
uses SMB exploit
URL ending in **/VidT6cErs**
returns **EXE** file
returns **EXE** file
returns **encoded binary**
EXE
EXE

Figure

8. TrickBot propagation flow chart since April 2020.

Like mworm, the new nworm module does not appear unless the TrickBot infection happens in an AD environment with a DC.

TrickBot Caused By Nworm: Not Persistent

When nworm infects a vulnerable DC, the malware is run from memory. No artifacts are found on the infected DC and TrickBot on the DC doesn't survive a reboot.

In cases where mshare and tab infect a vulnerable DC with TrickBot, these infections remain persistent on the DC, but TrickBot caused by nworm is not persistent. This shouldn't be an issue for the malware, because the DC is a server and servers rarely shut down or reboot like a Windows client.

Post-Infection Gtag from TrickBot Caused By Nworm

Every TrickBot binary has an identifier called a gtag. This is found in configuration data extracted from a TrickBot binary. Gtags can also be found in HTTP traffic during a TrickBot infection. They indicate the specific campaign or source of infection used for a TrickBot binary.

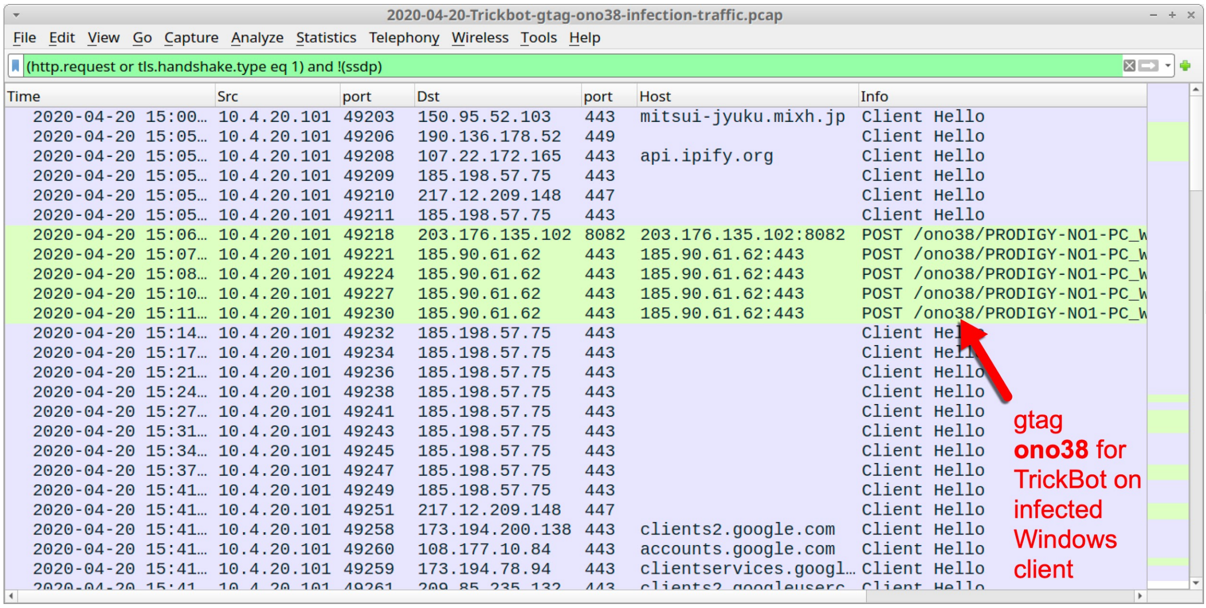
The gtag is a short alphabetic string followed by a number representing a one-up serialization. Examples follow:

- mor-series gtag: TrickBot caused by an Emotet infection, for example: TrickBot gtag mor84 caused by Emotet on January 27th, 2020.
- ono-series gtag: various TrickBot infections initiated through malicious Microsoft Office documents like Word documents or Excel spreadsheets, distributed through English-language emails.
- red-series gtag: TrickBot distributed as a DLL file instead of an EXE, for example: TrickBot gtag red5 documented on March 17th, 2020.

Gtags for TrickBot binaries used by TrickBot modules are unique. They break out as:

- tot-series gtag: TrickBot binaries used by mshare module
- jim-series gtag: TrickBot binaries used by nworm (and the old mworm) module
- lib-series gtag: TrickBot binaries used by tab module

Figure 9 and Figure 10 show gtags from traffic filtered in Wireshark from an infection on April 20th, 2020. In these images, the Windows client is at 10.4.20.101, and the DC is at 10.4.20.4.



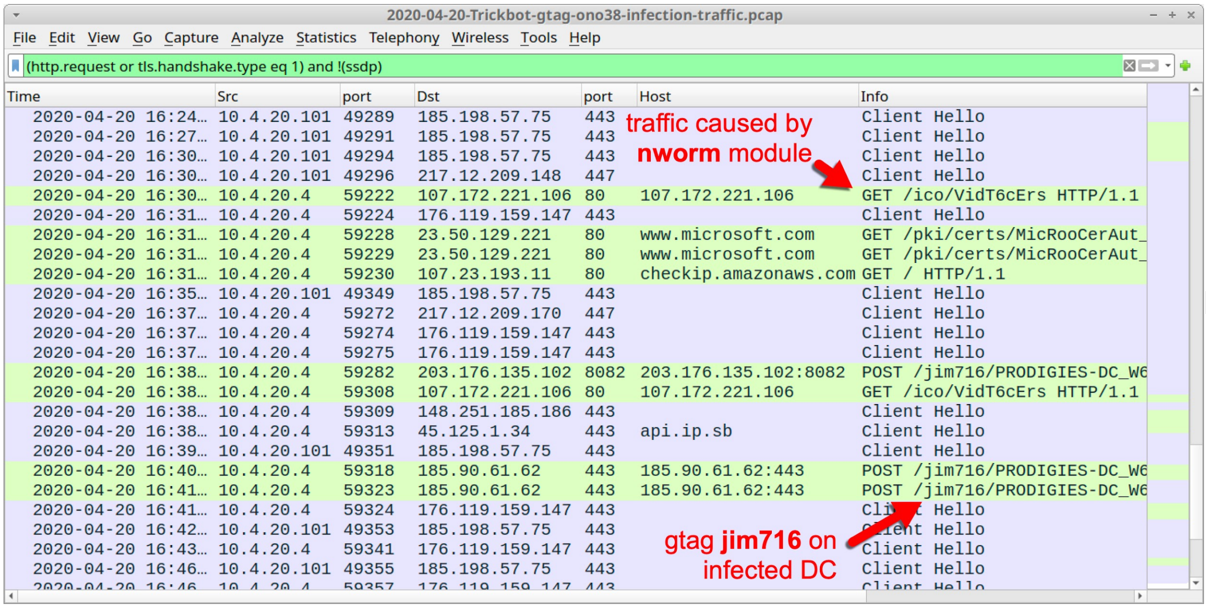
2020-04-20-Trickbot-gtag-ono38-infection-traffic.pcap

(http.request or tls.handshake.type eq 1) and !(ssdp)

Time	Src	port	Dst	port	Host	Info
2020-04-20 15:00…	10.4.20.101	49203	150.95.52.103	443	mitsui-jyuku.mixh.jp	Client Hello
2020-04-20 15:05…	10.4.20.101	49206	190.136.178.52	449		Client Hello
2020-04-20 15:05…	10.4.20.101	49208	107.22.172.165	443	api.ipify.org	Client Hello
2020-04-20 15:05…	10.4.20.101	49209	185.198.57.75	443		Client Hello
2020-04-20 15:05…	10.4.20.101	49210	217.12.209.148	447		Client Hello
2020-04-20 15:05…	10.4.20.101	49211	185.198.57.75	443		Client Hello
2020-04-20 15:06…	10.4.20.101	49218	203.176.135.102	8082	203.176.135.102:8082	POST /ono38/PRODIGY-NO1-PC_W
2020-04-20 15:07…	10.4.20.101	49221	185.90.61.62	443	185.90.61.62:443	POST /ono38/PRODIGY-NO1-PC_W
2020-04-20 15:08…	10.4.20.101	49224	185.90.61.62	443	185.90.61.62:443	POST /ono38/PRODIGY-NO1-PC_W
2020-04-20 15:10…	10.4.20.101	49227	185.90.61.62	443	185.90.61.62:443	POST /ono38/PRODIGY-NO1-PC_W
2020-04-20 15:11…	10.4.20.101	49230	185.90.61.62	443	185.90.61.62:443	POST /ono38/PRODIGY-NO1-PC_W
2020-04-20 15:14…	10.4.20.101	49232	185.198.57.75	443		Client Hello
2020-04-20 15:17…	10.4.20.101	49234	185.198.57.75	443		Client Hello
2020-04-20 15:21…	10.4.20.101	49236	185.198.57.75	443		Client Hello
2020-04-20 15:24…	10.4.20.101	49238	185.198.57.75	443		Client Hello
2020-04-20 15:27…	10.4.20.101	49241	185.198.57.75	443		Client Hello
2020-04-20 15:31…	10.4.20.101	49243	185.198.57.75	443		Client Hello
2020-04-20 15:34…	10.4.20.101	49245	185.198.57.75	443		Client Hello
2020-04-20 15:37…	10.4.20.101	49247	185.198.57.75	443		Client Hello
2020-04-20 15:41…	10.4.20.101	49249	185.198.57.75	443		Client Hello
2020-04-20 15:41…	10.4.20.101	49251	217.12.209.148	447		Client Hello
2020-04-20 15:41…	10.4.20.101	49258	173.194.200.138	443	clients2.google.com	Client Hello
2020-04-20 15:41…	10.4.20.101	49260	108.177.10.84	443	accounts.google.com	Client Hello
2020-04-20 15:41…	10.4.20.101	49259	173.194.78.94	443	clientservices.googl…	Client Hello
2020-04-20 15:41	10.4.20.101	49261	209.85.235.132	443	clients2.googleuserc	Client Hello

Figure

9. The initial TrickBot infection, where HTTP traffic from an infected client at 10.4.20.101 shows gtag ono38.



2020-04-20-Trickbot-gtag-ono38-infection-traffic.pcap

(http.request or tls.handshake.type eq 1) and !(ssdp)

Time	Src	port	Dst	port	Host	Info
2020-04-20 16:24…	10.4.20.101	49289	185.198.57.75	443		Client Hello
2020-04-20 16:27…	10.4.20.101	49291	185.198.57.75	443		Client Hello
2020-04-20 16:30…	10.4.20.101	49294	185.198.57.75	443		Client Hello
2020-04-20 16:30…	10.4.20.101	49296	217.12.209.148	447		Client Hello
2020-04-20 16:30…	10.4.20.4	59222	107.172.221.106	80	107.172.221.106	GET /ico/VidT6cErs HTTP/1.1
2020-04-20 16:31…	10.4.20.4	59224	176.119.159.147	443		Client Hello
2020-04-20 16:31…	10.4.20.4	59228	23.50.129.221	80	www.microsoft.com	GET /pki/certs/MicRooCerAut_
2020-04-20 16:31…	10.4.20.4	59229	23.50.129.221	80	www.microsoft.com	GET /pki/certs/MicRooCerAut_
2020-04-20 16:31…	10.4.20.4	59230	107.23.193.11	80	checkip.amazonaws.com	GET / HTTP/1.1
2020-04-20 16:35…	10.4.20.101	49349	185.198.57.75	443		Client Hello
2020-04-20 16:37…	10.4.20.4	59272	217.12.209.170	447		Client Hello
2020-04-20 16:37…	10.4.20.4	59274	176.119.159.147	443		Client Hello
2020-04-20 16:37…	10.4.20.4	59275	176.119.159.147	443		Client Hello
2020-04-20 16:38…	10.4.20.4	59282	203.176.135.102	8082	203.176.135.102:8082	POST /jim716/PRODIGIES-DC_W6
2020-04-20 16:38…	10.4.20.4	59308	107.172.221.106	80	107.172.221.106	GET /ico/VidT6cErs HTTP/1.1
2020-04-20 16:38…	10.4.20.4	59309	148.251.185.186	443		Client Hello
2020-04-20 16:38…	10.4.20.4	59313	45.125.1.34	443	api.ip.sb	Client Hello
2020-04-20 16:39…	10.4.20.101	49351	185.198.57.75	443		Client Hello
2020-04-20 16:40…	10.4.20.4	59318	185.90.61.62	443	185.90.61.62:443	POST /jim716/PRODIGIES-DC_W6
2020-04-20 16:41…	10.4.20.4	59323	185.90.61.62	443	185.90.61.62:443	POST /jim716/PRODIGIES-DC_W6
2020-04-20 16:41…	10.4.20.4	59324	176.119.159.147	443		Client Hello
2020-04-20 16:42…	10.4.20.101	49353	185.198.57.75	443		Client Hello
2020-04-20 16:43…	10.4.20.4	59341	176.119.159.147	443		Client Hello
2020-04-20 16:46…	10.4.20.101	49355	185.198.57.75	443		Client Hello
2020-04-20 16:46	10.4.20.4	59357	176.119.159.147	443		Client Hello

Figure

10. TrickBot spreads to the DC where we see gtag jim716 from an infection caused by the nworm module.

Conclusion

An infection caused by nworm is run from system memory, leaves no artifacts on an infected DC and disappears after a reboot or shutdown. Furthermore, the TrickBot binary used by nworm is encrypted or otherwise encoded when it is retrieved over the Internet. These characteristics are likely an attempt by TrickBot developers to avoid detection.

This is the latest in a series of changes in TrickBot as it evolves within our current threat landscape.

However, best security practices like running fully-patched and up-to-date versions of Microsoft Windows will hinder or prevent TrickBot infections. Palo Alto Networks customers are further protected from TrickBot by our threat prevention platform. AutoFocus users can track TrickBot activity by using the TrickBot tag.

Indicators of Compromise

Recent HTTP URLs for TrickBot binaries for propagation to vulnerable DC

(Read: First seen YYYY-MM-DD - module name - URL)

2020-04-20 - nworm - hxxp://107.172.221[.]106/ico/VidT6cErs

2020-04-20 - mshare - hxxp://107.172.221[.]106/images/cursor.png

2020-04-20 - tab - hxxp://107.172.221[.]106/images/imgpaper.png

2020-05-08 - nworm - hxxp://23.95.227[.]159/ico/VidT6cErs

2020-05-08 - mshare - hxxp://23.95.227[.]159/images/cursor.png

2020-05-08 - tab - hxxp://23.95.227[.]159/images/imgpaper.png

SHA256 hash for nwormDll64 artifact (encrypted binary) from an infected Windows 7 client on April 24th 2020:

900aa025bf770102428350e584e8110342a70159ef2f92a9bfd651c5d8e5f76b

SHA256 hash for nwormDll64 artifact (encrypted binary) from an infected Windows 7 client on May 8th 2020:

85d88129eab948d44bb9999774869449ab671b4d1df3c593731102592ce93a70