

Goodbye Mworm, Hello Nworm: TrickBot Updates Propagation Module

By Brad Duncan

Published: 2020-05-28 · Archived: 2026-04-05 13:46:40 UTC

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)

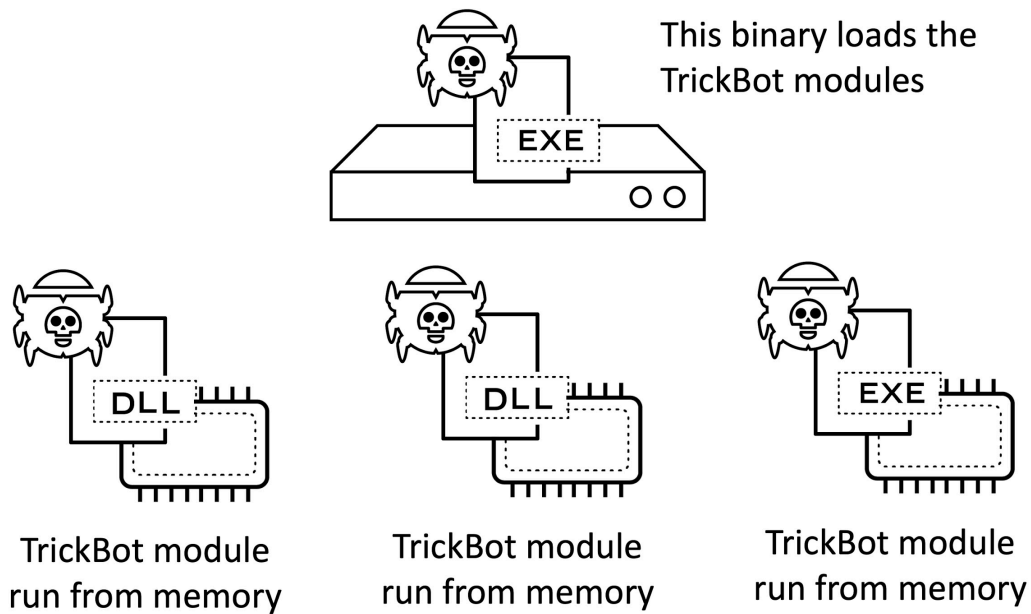


Figure 1. A visual representation of TrickBot and its modules.

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](#).

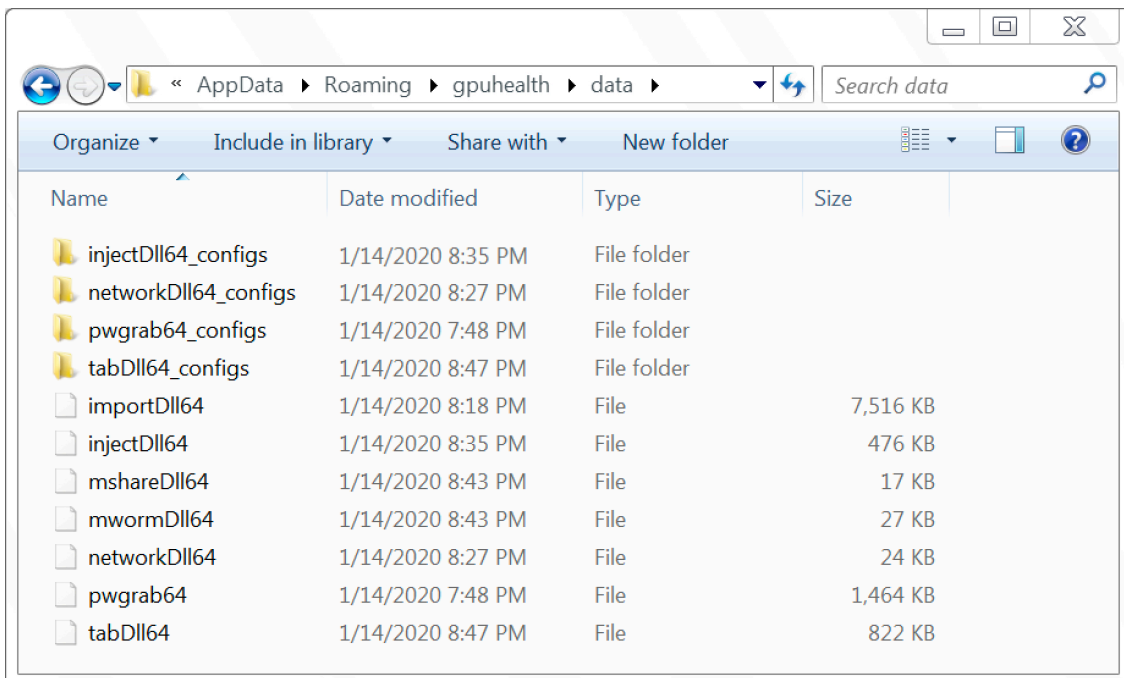


Figure 2. Example of artifacts for TrickBot modules on an infected Windows 7 client.

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

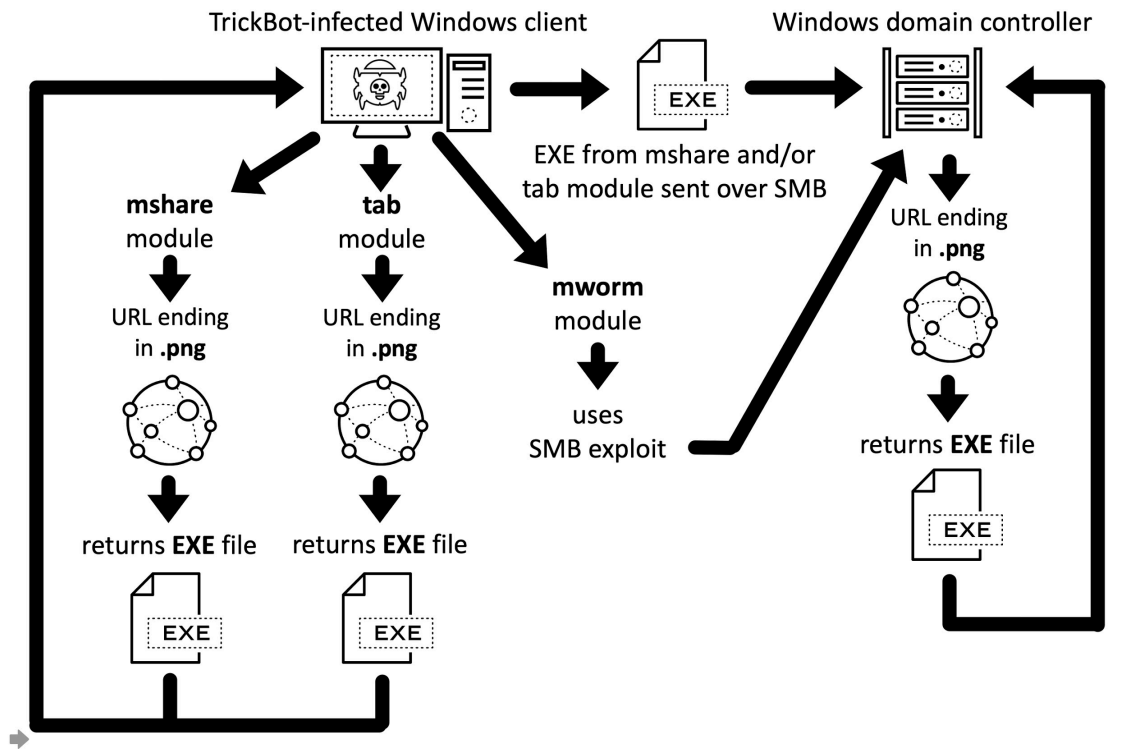


Figure 3. TrickBot propagation flow chart from September 2019 through March 2020.

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.

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.

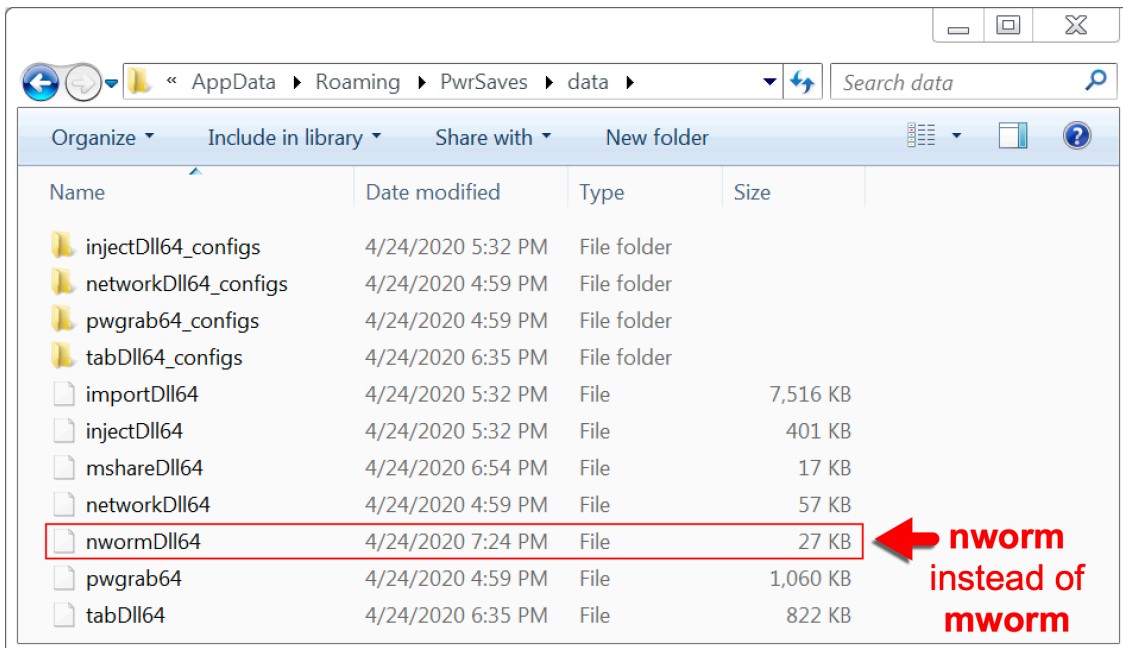


Figure 5. New nworm module found from an infection on April 24, 2020.

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](#).

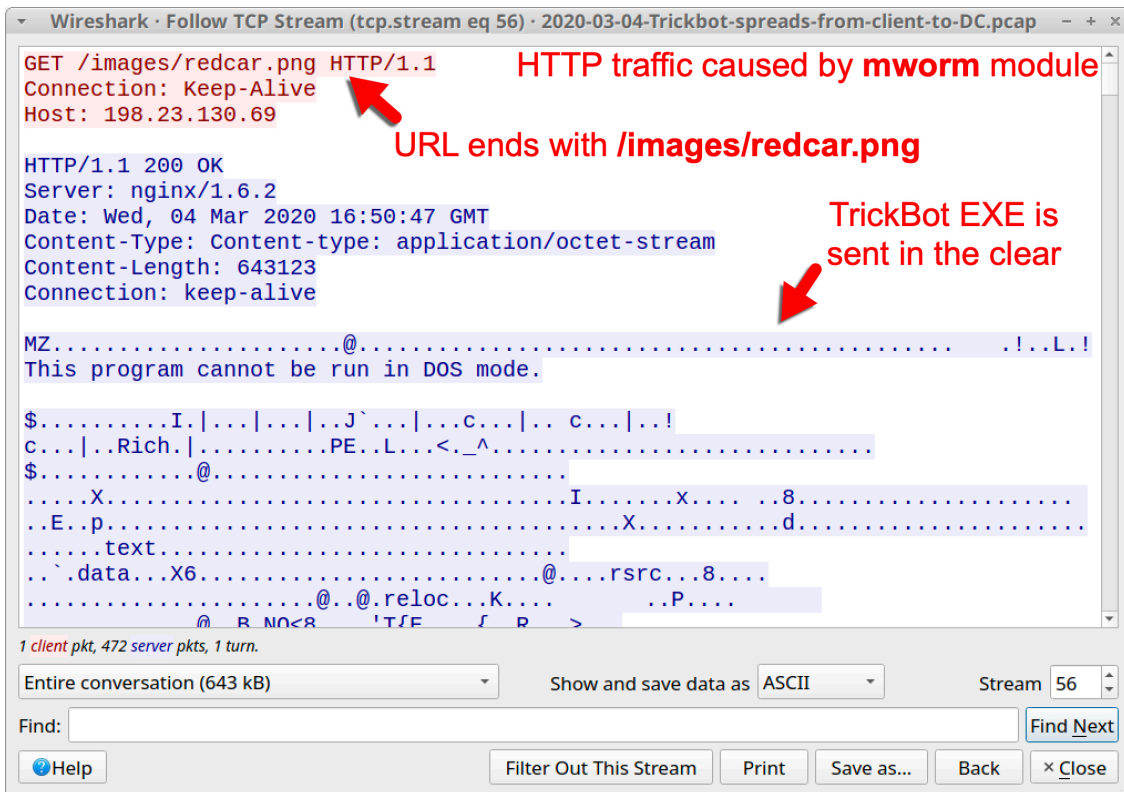


Figure 6. TCP stream showing HTTP traffic caused by the mworm module in March 2020.

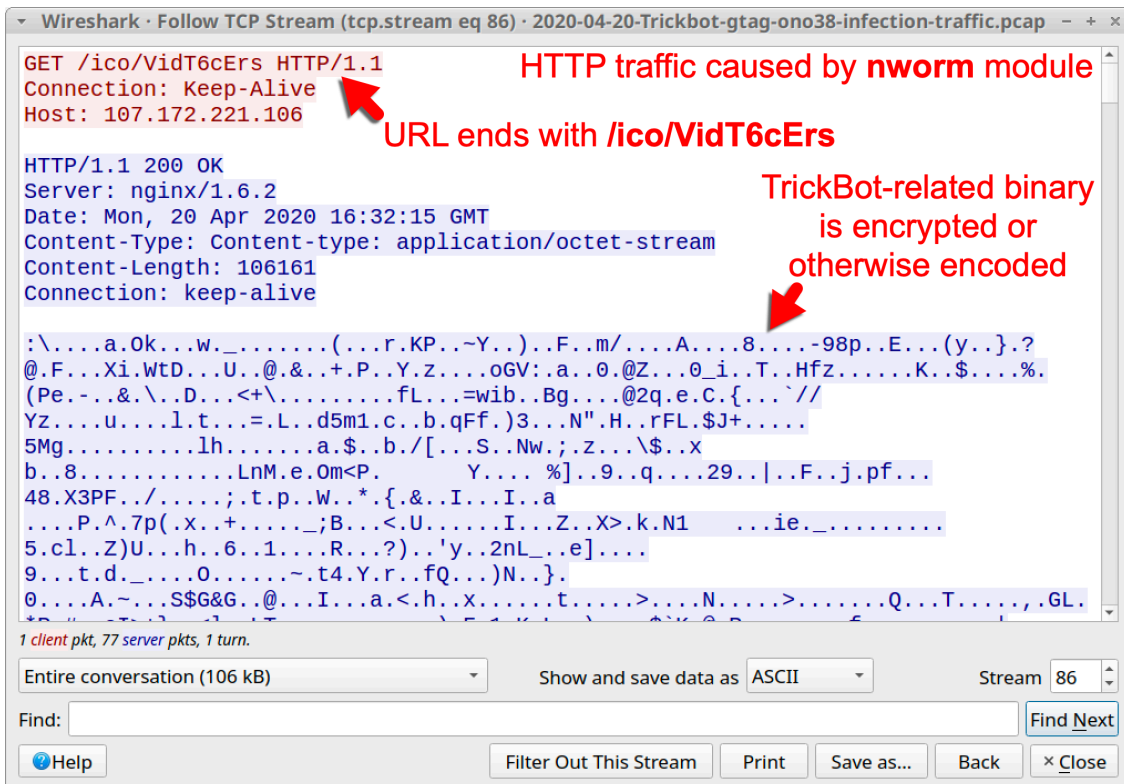


Figure 7. TCP stream showing HTTP traffic caused by the nworm module April 2020.

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

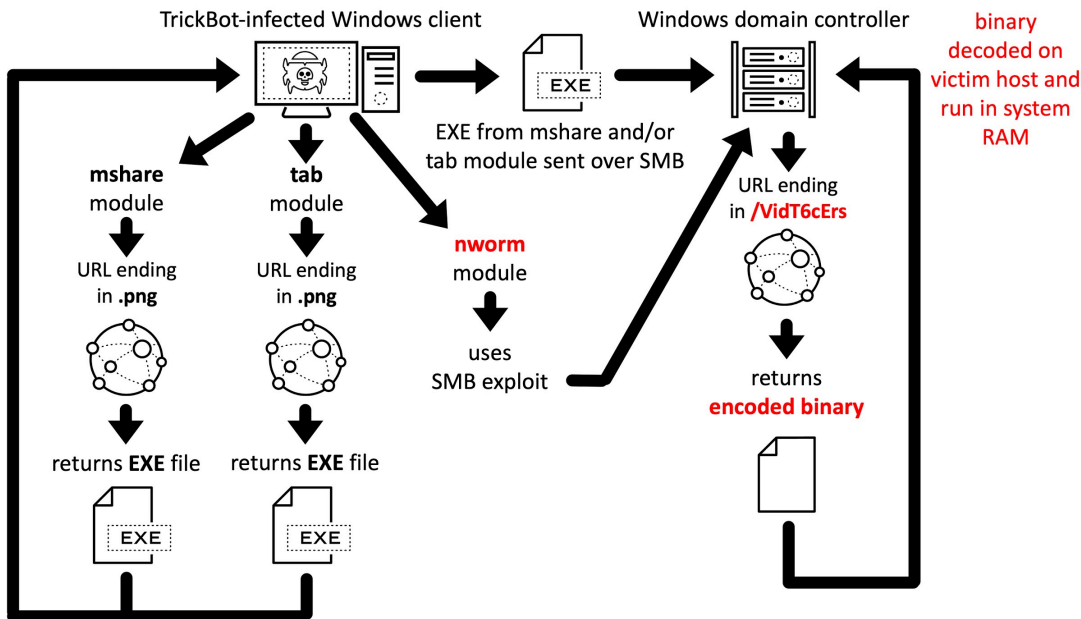


Figure 8. TrickBot propagation flow chart since April 2020.

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.

- o 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:

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

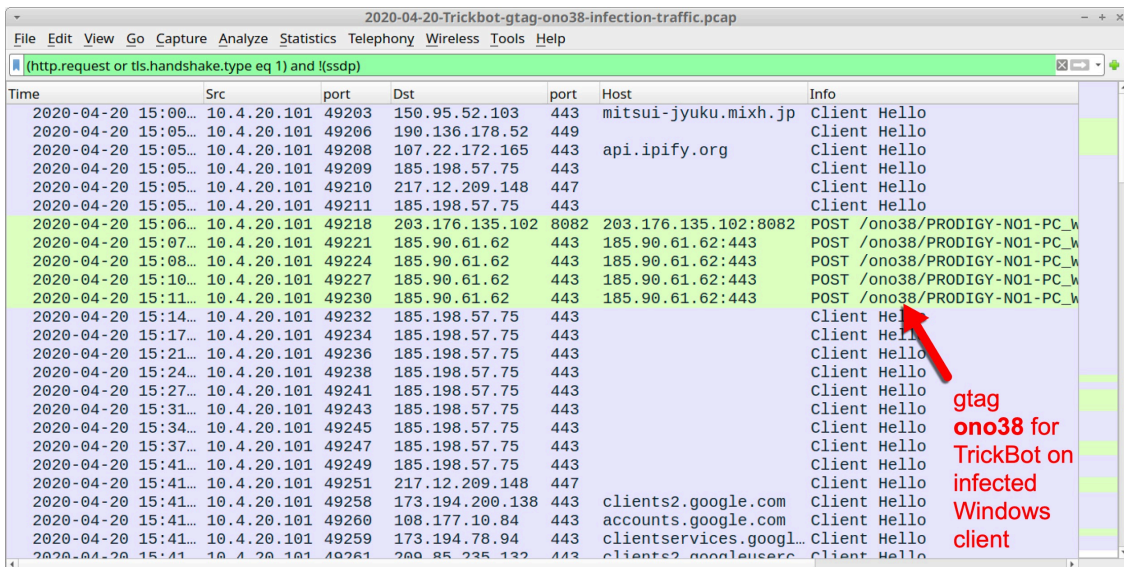


Figure 9. The initial TrickBot infection, where HTTP traffic from an infected client at 10.4.20.101 shows gtag ono38.

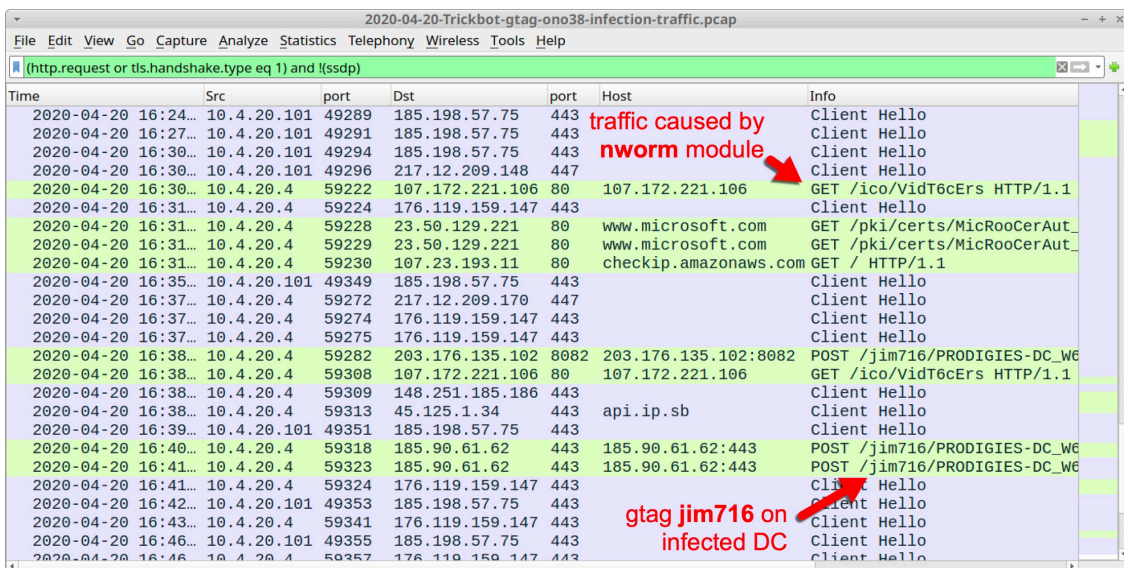


Figure 10. TrickBot spreads to the DC where we see gtag jim716 from an infection caused by the nworm module.

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

Source: <https://unit42.paloaltonetworks.com/goodbye-nworm-hello-nworm-trickbot-updates-propagation-module/>