TrickBot Banking Trojan Adapts with New Module | Webroot

By Jason Davison

Published: 2018-03-21 · Archived: 2026-04-02 10:38:39 UTC

Since inception in late 2016, the TrickBot banking trojan has continually undergone updates and changes in attempts to stay one step ahead of defenders and internet security providers. While TrickBot has not always been the stealthiest trojan, its authors have remained consistent in the use of new distribution vectors and development of new features for their product. On March 15, 2018, Webroot observed a module (tabDll32 / tabDll64) being downloaded by TrickBot that has not been seen in the wild before this time.

It appears that the TrickBot authors are still attempting to leverage MS17-010 and other lateral movement methods coupled with this module in an attempt to create a new monetization scheme for the group.

You can teach an old bot older tricks

Analyzed samples

- 0058430e00d2ea329b98cbe208bc1dad – main sample (packed)
    - 0069430e00d2ea329b99cbe209bc1dad – bot 32 bit

Downloaded Modules

- 711287e1bd88deacda048424128bdfaf – systeminfo32.dll
- 58615f97d28c0848c140d5e78ffb2add – injectDll32.dll
- 30fc6b88d781e52f543edbe36f1ad03b – wormDll32.dll
- 5be0737a49d54345643c8bd0d5b0a79f – shareDll32.dll
- 88384ba81a89f8000a124189ed69af5c – importDll32.dll
- 3def0db658d9a0ab5b98bb3c5617afa3 – mailsearcher32.dll
- **311fdc24ce8dd700f951a628b805b5e5 – tabDll32.dll**

Behavioral Analysis

Upon execution, this iteration of TrickBot will install itself into the %APPDATA%\TeamViewer\ directory. If the bot has not been executed from its installation directory, it will restart itself from this directory and continue operation. Once running from its installation directory, TrickBot will write to the usual group_tag and client_id files along with creating a “Modules” folder used to store the encrypted plug and play modules and configuration files for the bot.

Image 1: TrickBot's plug and play modules used to extend the bots functionality

Many of the modules shown above have been previously documented. The systeminfo and injectDll module have been coupled with the bot since its inception. The mailsearcher module was added in December 2016 and the worm module was discovered in late July 2017. The module of interest here is tabDll32 as this module has been previously undocumented. Internally, the module is named spreader_x86.dll and exports four functions similar to the other TrickBot modules.

Image 2a: Peering inside tabDll.dll

Image 2b: Abnormally large .rdata section

The file has an abnormally large rdata section which proves to be quite interesting because it contains two additional files intended to be used by spreader_x86.dll. The spreader module contains an additional executable SsExecutor_x86.exe and an additional module screenLocker_x86.dll. Each module will be described in more detail in its respective section below.

Spreader_x86.dll

When loading the new TrickBot module in IDA, you are presented with the option of loading the debug symbol filename.

Image 3: Debug symbol filename of the downloaded module tabDll.dll

This gives us a preview of how the TrickBot developers structure new modules that are currently under development. When digging deeper into the module, it becomes evident that this module is used to spread laterally through an infected network making use of MS17-010.

Image 4: String references to EternalRomance exploit used for lateral movement

This module appears to make use of lateral movement in an attempt to set up the embedded executable as a service on the exploited system. Additionally, the TrickBot authors appear to be still developing this module as parts of the modules reflective dll injection mechanism are stolen from GitHub.

Image 5: Copied code from ImprovedReflectiveDLLInjection

Image 6: Printf statements from the copied project on GitHub

SsExecutor_x86.exe

The second phase of the new module comes in the form of an executable meant to run after post exploitation. Again, it was very nice of the TrickBot authors to give us a look at the debug symbols file path.

Image 7: Debug symbol filename of the embedded PE file.

When run, this executable will iterate over the use profiles in registry and goes to each profile to add a link to the copied binary to the start up path. This occurs after lateral movement takes place.

Image 8: Iterate over user profiles and create

Image 9: Execution of the copied binary

ScreenLocker_x86.dll

Similarly, to the other TrickBot modules, this module was written in Delphi. This is the first time TrickBot has shown any attempt at “locking” the victims machine.

Image 10: Peering inside screenLocker_x86.dll

This Module exports two functions, “MyFunction” and a reflective DLL loading function. “MyFunction” appears to be the work in progress:

Image 11: Peering inside “MyFunction”

Image 12: Creation of the Locker Window

If the TrickBot developers are attempting to complete this locking functionality, this generates interesting speculation around the group's business model. Locking a victim's computer before you are able to steal their banking credentials alerts the victim that they are infected, thus limiting the potential for credit card or bank theft. However, extorting victims to unlock their computer is a much simpler monetization scheme.

It is notable that this locking functionality is only deployed after lateral movement, meaning that it would be used to primarily target unpatched corporate networks. In a corporate setting (with unpatched machines) it is highly likely that backups would not exist as well. The authors appear to be getting to know their target audience and how to best extract money from them. On a corporate network, where users are unlikely to be regularly visiting targeted banking URLs, exfiltrating banking credentials is a less successful money-making model compared to the locking of potentially hundreds of machines.

The TrickBot authors continue to target various financial institutions across the world, using MS17-010 exploits in an attempt to successfully laterally move throughout a victim's network. This is being coupled with an unfinished "screenLocker" module in a new possible attempt to extort money from victims. The TrickBot banking trojan remains under continual development and testing in a constant effort by its developers to stay one step ahead of cybersecurity professionals.

About the Author

Jason Davison

Advanced Threat Research Analyst

Jason is a Malware Threat Researcher, investigating the latest techniques used in modern malware. Working for Webroot, he researches and reverses the latest malware families identifying new functionality and TTP’s.

-
