A Deeper Look Into TrickBot's Machinations

securityintelligence.com/tricks-of-the-trade-a-deeper-look-into-trickbots-machinations/

November 9, 2016

Home / Advanced Threats

Tricks of the Trade: A Deeper Look Into TrickBot’s Machinations

Advanced Threats November 9, 2016
By Lior Keshet 6 min read

TrickBot is a new banking Trojan. It appears to be a Dyre successor that emerged in the wild in October 2016. TrickBot's code has been in progressive testing since August 2016. It continues to see ongoing updates and, now, actual infection campaigns and fraud attacks.

Internally, there is more to TrickBot than meets the eye. In this research post, we'll cover some of the most notable points about this malware's capabilities, including:

- An uncommon method of performing man-in-the-browser (MitB) attacks;
- TrickBot's buggy webinjection mechanism;
- The developer's elegant application program interface (API) obfuscation, borrowed from Carberp; and
- Our two cents about the suspected TrickBot-Dyre connection.

For analysis, the sample we used was: 5e363a42d019fc6535850a2867548f5b968d68952e1cddd49240d1f426debb73.

An Unusual Man-in-the-Browser Technique

Nowadays, most modern financial malware families are capable of injecting malicious code into ongoing browser sessions (e.g., MitB or a webinjection attack). The most common way malware developers implement injections is by setting them up locally at the victim's machine. The malware keeps a local configuration file for the injections, specifying exactly when and how the malware will modify the contents of targeted bank webpages.

A more advanced and less common method to achieve the same result is to fetch the injection instructions from the attacker's server in real time. This is the method TrickBot's developers opted to use. It is also known as serverside injections.

For this purpose, and much like other advanced banking Trojans, TrickBot deploys a browser-hooking engine designed to intercept communications to and from the victim's internet browser. With the real-time fetching trick, the malicious code injections themselves are kept securely on the attacker's server, not in a file on the victim's endpoint.

When a victim browses one of TrickBot's target URLs, this is what actually happens:

1. TrickBot's financial module intercepts the original HTTP response before it is rendered to the victim. The browser does not display the "clean" response.
2. TrickBot sends a multipart HTTP packet to its C2 with the following sections:
    1. "sourcelink," the complete URL that triggered the attack;
    2. "sourcequery," the browser's complete HTTP query; and
    3. "sourcehtml," the original HTML as would be displayed by an uninfected browser.
3. C2 replies with full HTML content to be rendered by the victim's browser, including injected parts.
4. Finally, TrickBot's financial module replaces the original response that would normally come from the bank with the C2's response, and the injected page is displayed on the victim's end.

The serverside injection method has advantages over the standard, local mechanism used by most financial malware today. Notably, it allows for enhanced obscurity and flexibility. The malware's author can keep the injection code out of sight until it is needed. The actor can turn the webinjections on or off on the fly, easily modify the injections and then push an update to some or all the infected victims instantaneously.

Figure 1: TrickBot hooking FireFox's network functions to enable MitB interception.

Figure 2: TrickBot's Server Side Web-Injects — Top Level Flow.

An Elegant Choice for API Obfuscation

When it comes to keeping malware alive longer, it is common practice for malware authors to add protection layers to their code to ward off reverse engineering. As expected, we identified one such technique employed by TrickBot: API obfuscation.

Having analyzed TrickBot's obfuscation method, we found it very similar to — and likely borrowed from — the Carberp Trojan's API obfuscation. Carberp's source code was leaked in 2013, giving rise to other malware based on its sophisticated DNA.

We found that TrickBot does not apply the API obfuscation to all the APIs; it only applies it to the more sensitive APIs that the developer wants to be hidden. This is a sneaky method, since researchers may believe they already know all the APIs being used, when in reality more APIs are covertly part of the game.

The obfuscation process here is based on precalculated hash values of the APIs. Calling an API function only includes a hash value instead of the function name, making static analysis harder unless the researcher applies an additional method to resolve the APIs.

Figure 3: WSAStartup hash from Carberp's source code.

Figure 4: Resolving an API by hash — WSAStartup.

A simple way of overcoming this obfuscation is by using an Interactive Disassembler (IDA) Python script, especially since the hashed values themselves are already available within Carberp's leaked source code.

A Bug in the Wild

TrickBot has been in testing since summer 2016, even before it was equipped with financial malware features. Initially, TrickBot's developers appeared to struggle with the malware's webinjection mechanism, since we found a few TrickBot samples in the wild that presented strangely erratic behavior. At first, we suspected TrickBot was up to some anti-research wiles, but in reality, it was just bugged.

Per our analysis, TrickBot's webinjection malfunction caused the malware to constantly inject the same code over and over again, sabotaging the malware's own functionality. As this behavior was inconsistent across some samples, we had to manually apply a fix to continue researching the mechanism.

We won't get into further detail at this point, since this bug actually prevented TrickBot from performing fraud. We will say, however, that since the malware is under constant development, the developers may have already addressed the bug and fixed it in the newer samples, enabling TrickBot to operate more smoothly.

The TrickBot-Dyre Connection

Speculation over the TrickBot-Dyre connection emerged as soon as the malware was discovered and has been the source of much debate ever since. Even though this subject was mentioned in several other TrickBot analysis blogs, we would like to contribute several key points from our own research into this new threat:

> TrickBot's serverside webinjection method is uncommon in today's malware. The other malware family that used it, as you can guess, was Dyre.
>
> Packets sent to the attack server during the serverside webinjects consist of three parts, titled "sourcelink," "sourcequery" and "sourcehtml." These exact names were also used in Dyre's webinjection mechanism.

Figure 5: TrickBot and Dyre both use "sourcelink" and "sourcequery" for their communications.

> Targeted URLs and command-and-control (C&C) addresses are kept encrypted on the infected machine. Dyre did the same. While the encryption schemes for TrickBot are not identical, they appear to be too similar to be a mere coincidence.
>
> TrickBot passes the target URLs list to its financial module, which is injected into the browser using pipes communication. This, again, is a Dyre hallmark.
>
> The structure of the targeted URLs in the configuration is typically consistent for each malware, and TrickBot's target specification — you guessed it — sure looks a lot like Dyre's.

Related: An Aggressive Launch: TrickBot Trojan Rises With Redirection Attacks in the UK
Although the similarities are there, keep in mind that most of them are relatively simply to imitate. For example, even though the serverside webinjection technique is common to both Trojans, the code implementing this capability and the coding style itself are actually different.

This point is important because we can see how rapidly and effectively new and advanced malware is being developed, either by the same actors or by newcomers inspired by one of the most nefarious gangs in cybercrime history.

A Last-Minute Update: Redirection Attacks!

TrickBot's developers must be hard at work these days, pushing to enhance the malware for a live campaign targeting banks. Just as we were about to publish this post, we detected a new infection campaign with a new configuration targeting U.K. banks. Until now, TrickBot

only targeted banks in Australia. Furthermore, some of these new U.K. targets are set up for redirection attacks, while TrickBot only employed the serverside webinjection attack described above until now.

A redirection attack, in short, means that instead of injecting malicious code into the original webpage, the victim is now redirected to a new site forged by the fraudsters. This site looks and feels exactly like the original website, and the browser indicates a Secure Sockets Layer (SSL) connection based on the original site's certificate.

To learn more about redirection attacks and their purpose, read our blogs about Dridex and GozNym.

A Newcomer to the Malware Arena

TrickBot is undoubtedly the work of professionals who have been around the banking Trojan scene for some time. These experienced fraudsters are apparently well-versed in the modern features common to the types of malware banks reckon with nowadays. We expect to see this Trojan evolve its anti-security and anti-research techniques and pop up in more infection campaigns as the year comes to a close.

Lior Keshet
Malware Research Technical Lead, IBM Trusteer

Lior is a malware research technical lead at IBM Security's Trusteer's group. He has been a core member of the Trusteer cybercrime labs for the past four yea...