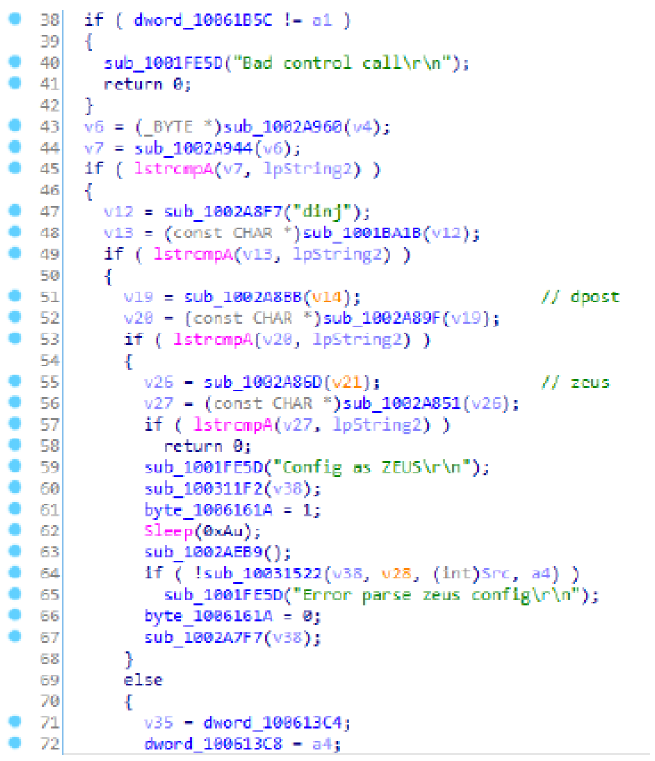
TrickBot and Zeus - Kryptos Logic

kryptoslogic.com/blog/2021/07/trickbot-and-zeus/



```
if ( dword_10061B5C != a1 )
{
  sub_1001FE5D("Bad control call\r\n");
  return 0;
}
v6 = (_BYTE *)sub_1002A960(v4);
v7 = sub_1002A944(v6);
if ( lstrcmpA(v7, lpString2) )
{
  v12 = sub_1002A8F7("dinj");
  v13 = (const CHAR *)sub_1001BA1B(v12);
  if ( lstrcmpA(v13, lpString2) )
  {
    v19 = sub_1002A8BB(v14);                    // dpost
    v20 = (const CHAR *)sub_1002A89F(v19);
    if ( lstrcmpA(v20, lpString2) )
    {
      v26 = sub_1002A86D(v21);                  // zeus
      v27 = (const CHAR *)sub_1002A851(v26);
      if ( lstrcmpA(v27, lpString2) )
        return 0;
      sub_1001FE5D("Config as ZEUS\r\n");
      sub_100311F2(v38);
      byte_1006161A = 1;
      Sleep(0xAu);
      sub_1002AEB9();
      if ( !sub_10031522(v38, v28, (int)Src, a4) )
        sub_1001FE5D("Error parse zeus config\r\n");
      byte_1006161A = 0;
      sub_1002A7F7(v38);
    }
    else
    {
      v35 = dword_100613C4;
      dword_100613C8 = a4;
```

TrickBot and Zeus

Authored by: Kryptos Logic Vantage Team on Thursday, July 1, 2021
Tags: TrickBot

Overview

TrickBot is an established and widespread multi-purpose trojan. Active since 2016 and modular in nature, it can accomplish a variety of goals ranging from credential theft to lateral movement. Many of the malware's capabilities come as self-contained modules, which the malware is instructed to download from the C2. Initially, TrickBot's main focus was bank fraud, but this later shifted toward corporate targetted ransomware attacks, eventually resulting in the discontinuation of their fraud operation.

In June 2021, Kryptos Logic Threat Intelligence team began observing new developments to the TrickBot webinject module. TrickBot's webinject module supports both static and dynamic configuration for injects. The static inject type causes the victim to be redirected to an attacker-controlled replica of the intended destination site, where credentials can then be harvested. The dynamic inject type transparently forwards the server response to the TrickBot C2, where the source is then modified to contain malicious components, before being returned to the victim as though it came from the legitimate site. In the current iteration of the webinject module, injects are achieved by proxying traffic through a local SOCKS server, if the traffic matches a list of target URLs the traffic is modified accordingly.

Through our monitoring, we were able to obtain a debug version of the module, which contained new features being tested. In this evolution of their webinject capabilities, TrickBot has added support for Zeus-style webinject configs. During development the module was served under a test name and supported parsing configs named `zeus`. We have since observed the updated module being pushed out to real victims under the name `injectDll`, superseding their old webinject module. Previously, webinject configs were stored in two files named `sinj` (static injects) and `dinj` (dynamic injects). Now, there exists a single config file named `winj` containing Zeus-style webinjects.

InjectDLL

We began observing new changes to their webinject module when we came across a debug build uploaded to [1]VirusTotal. This module was different from the previous known webinjects, since it makes reference to a config by the name of `winj`.

However, we were not able to pull down such a debug build from the C2 servers, since we were not aware of the base name that was given to the module. Base names are the names given to modules by TrickBot, and are used by the main bot to fetch these modules from TrickBot's plugin servers.

Currently we are seeing the bot pull down the released versions of webinject module as `injectDll32` and `injectDll64` for the 32-bit and 64-bit module respectively.

An example for a module request made by the main bot is as follows:
`https://<c2_address>/<gtag>/<unique_bot_id>/5/injectDll64/`

In this blog we will not go into detail on how the main bot communicates to its C2, since it has already been covered[2].

winj Config

In this module we see the introduction of Zeus webinject configs, in addition to the regular webinject configs that TrickBot previously used. A good description on how the webinject config works can be found in its leaked manual[3].

Due to Zeus having been the gold standard for banking malware, Zeus-style webinjects are extremely popular. It is not uncommon for other malware families to support Zeus-style webinject syntax for cross-compatibility ([4]Zloader, [5]Citadel, to name a few).

In the debug version, the module referred to this new config simply as `zeus`; however in the release version this has been renamed to `winj`. As with the previous webinjects, the module is able to parse the usual webinject configs `dinj` and `sinj`. Additionally, they continue to use the `dpost` configs in order to send the results of the webjects to the handler C2s.

Example `winj`:

```
set_url <target-website-url> GP
data_before
data_end
data_after
</head>
data_end
data_inject
<script></script>
data_end
```

MitM

As previously reported[6], the module shares substantial code with IcedID/Bokbot's Man-in-the-Browser webinject module. To MitM TLS connections, it creates a self-signed TLS certificate and adds it to the certificate store. The module contains a packed payload that is injected into the victim's browser, where it hooks socket APIs to redirect traffic to a locally listening SOCKS proxy, it also hooks `CertVerifyCertificateChainPolicy` and `CertGetCertificateChain` to ensure no certificate errors are shown to the victim.

Conclusion

The resumption of development of the webinject module indicates that TrickBot intends to revive its bank fraud operation, which appears to have been shelved for over a year. The addition of Zeus-style webinjects may suggest expansion of their Malware-as-a-Service platform, enabling users to bring their own webinjects.

IOCs

SHA256	Module Base Name	Description
dd268740958fe3829c927054b900f6287235662cd93c1e91d51c38c44eb2571b	n/a	32-bit Webinject Debug Module
3bd5117466a9bcd539a482343957f8c9a74ad2fa0d5da959fcfa0d42beb9133d	n/a	64-bit Webinject Debug Module
c79f016996b45cd7cc88aa3ce6c4d1ab247a1d803de4b742f64f3bf1e183ffb0	injectDll32	32-bit Webinject Module
19f4f1ca50b3306c9d953e12e573b1e65670b110b358d0240e55331c7ed0d76f	injectDll64	64-bit Webinject Module

YARA

```
rule TrickBot__Webinject
{
    meta:
        id = "7eKHCSumVp7QRXF0z7BC1i"
        fingerprint = "8a66f290d84a54e8ee148c461513627e83b8f092acebe8251c6098a88d8c4eba"
        version = "1.0"
        first_imported = "2021-07-01"
        last_modified = "2021-07-01"
        status = "RELEASED"
        sharing = "TLP:WHITE"
        source = "KRYPTOS LOGIC"
        author = "KRYPTOS LOGIC"
        description = "Detects TrickBot updated Webinject module"
        category = "MALWARE"
        malware = "BOT"
        hash = "c79f016996b45cd7cc88aa3ce6c4d1ab247a1d803de4b742f64f3bf1e183ffb0"

    strings:
        $a = "c:\\developer\\webinject\\http-lib\\parser.c" wide
        $zeus1 = "data_before"
        $zeus2 = "data_after"
        $zeus3 = "data_inject"
        $zeus4 = "data_end"
        $name = /wbi-x(86|64).dll/

    condition:
        all of them
}
```

References