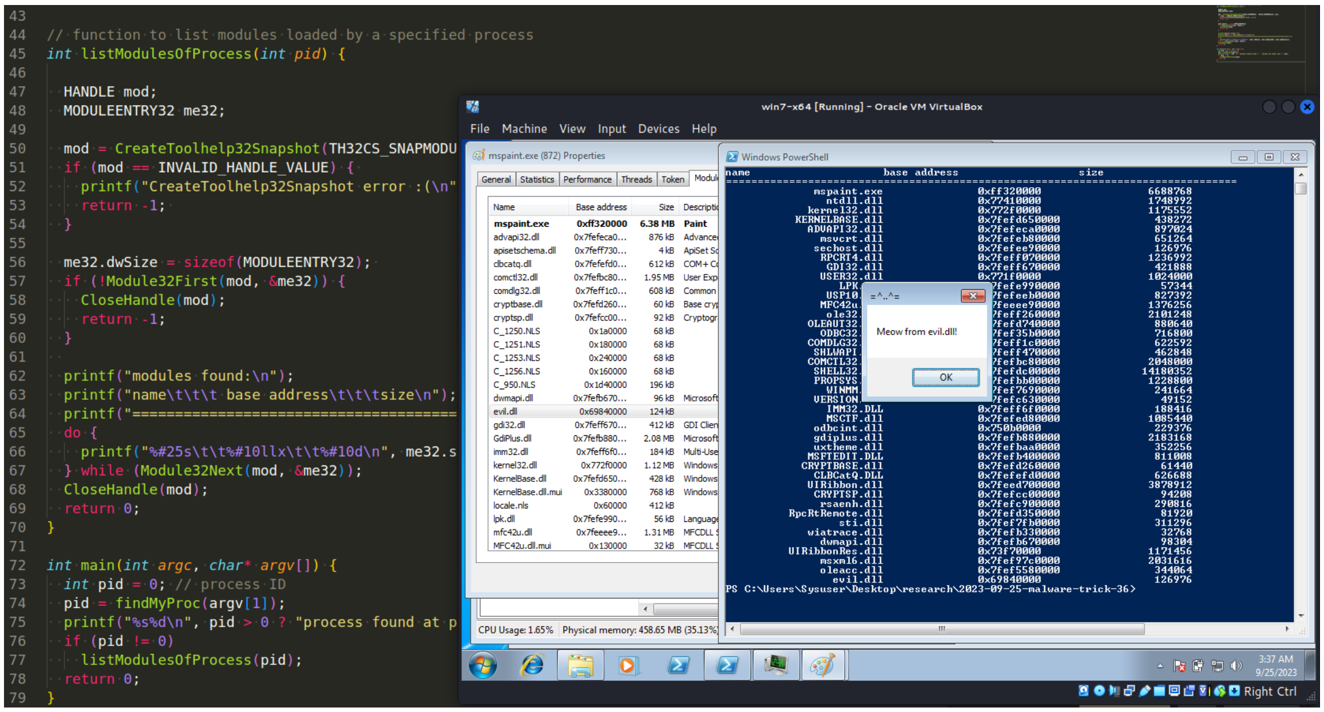
Malware development trick - part 36: Enumerate process modules. Simple C++ example.

cocomelonc.github.io/malware/2023/09/25/malware-trick-36.html

September 25, 2023

3 minute read

Hello, cybersecurity enthusiasts and white hackers!



```c

// function to list modules loaded by a specified process
int listModulesOfProcess(int pid) {

  HANDLE mod;
  MODULEENTRY32 me32;

  mod = CreateToolhelp32Snapshot(TH32CS_SNAPMODU
  if (mod == INVALID_HANDLE_VALUE) {
    printf("CreateToolhelp32Snapshot error :(\n"
    return -1;
  }

  me32.dwSize = sizeof(MODULEENTRY32);
  if (!Module32First(mod, &me32)) {
    CloseHandle(mod);
    return -1;
  }

  printf("modules found:\n");
  printf("name\t\t\t base address\t\t\tsize\n");
  printf("====================================
  do {
    printf("%#25s\t\t%#10llx\t\t%#10d\n", me32.s
  } while (Module32Next(mod, &me32));
  CloseHandle(mod);
  return 0;
}

int main(int argc, char* argv[]) {
  int pid = 0; // process ID
  pid = findMyProc(argv[1]);
  printf("%s%d\n", pid > 0 ? "process found at p
  if (pid != 0)
    listModulesOfProcess(pid);
  return 0;
}
```

win7-x64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
mspaint.exe (872) Properties
General Statistics Performance Threads Token Modul

Name	Base address	Size	Descriptio
mspaint.exe	**0xff320000**	**6.38 MB**	**Paint**
advapi32.dll	0x7fefeca0...	876 kB	Advance
apisetschema.dll	0x7feff730...	4 kB	ApiSet Sc
clbcatq.dll	0x7fefefd0...	612 kB	COM+ C
comctl32.dll	0x7fefbc80...	1.95 MB	User Exp
comdlg32.dll	0x7feff1c0...	608 kB	Common
cryptbase.dll	0x7fefd260...	60 kB	Base cryp
cryptsp.dll	0x7fefcc00...	92 kB	Cryptogr
C_1250.NLS	0x1a0000	68 kB	
C_1251.NLS	0x180000	68 kB	
C_1253.NLS	0x240000	68 kB	
C_1256.NLS	0x160000	68 kB	
C_950.NLS	0x1d40000	196 kB	
dwmapi.dll	0x7fefb670...	96 kB	Microsoft
evil.dll	0x69840000	124 kB	
gdi32.dll	0x7feff670...	412 kB	GDI Clien
GdiPlus.dll	0x7fefb880...	2.08 MB	Microsoft
imm32.dll	0x7feff6f0...	184 kB	Multi-Use
kernel32.dll	0x772f0000	1.12 MB	Windows
KernelBase.dll	0x7fefd650...	428 kB	Windows
KernelBase.dll.mui	0x3380000	768 kB	Windows
locale.nls	0x60000	412 kB	
lpk.dll	0x7fefe990...	56 kB	Language
mfc42u.dll	0x7feeee9...	1.31 MB	MFCDLL
MFC42u.dll.mui	0x130000	32 kB	MFCDLL

CPU Usage: 1.65% Physical memory: 458.65 MB (35.13%)

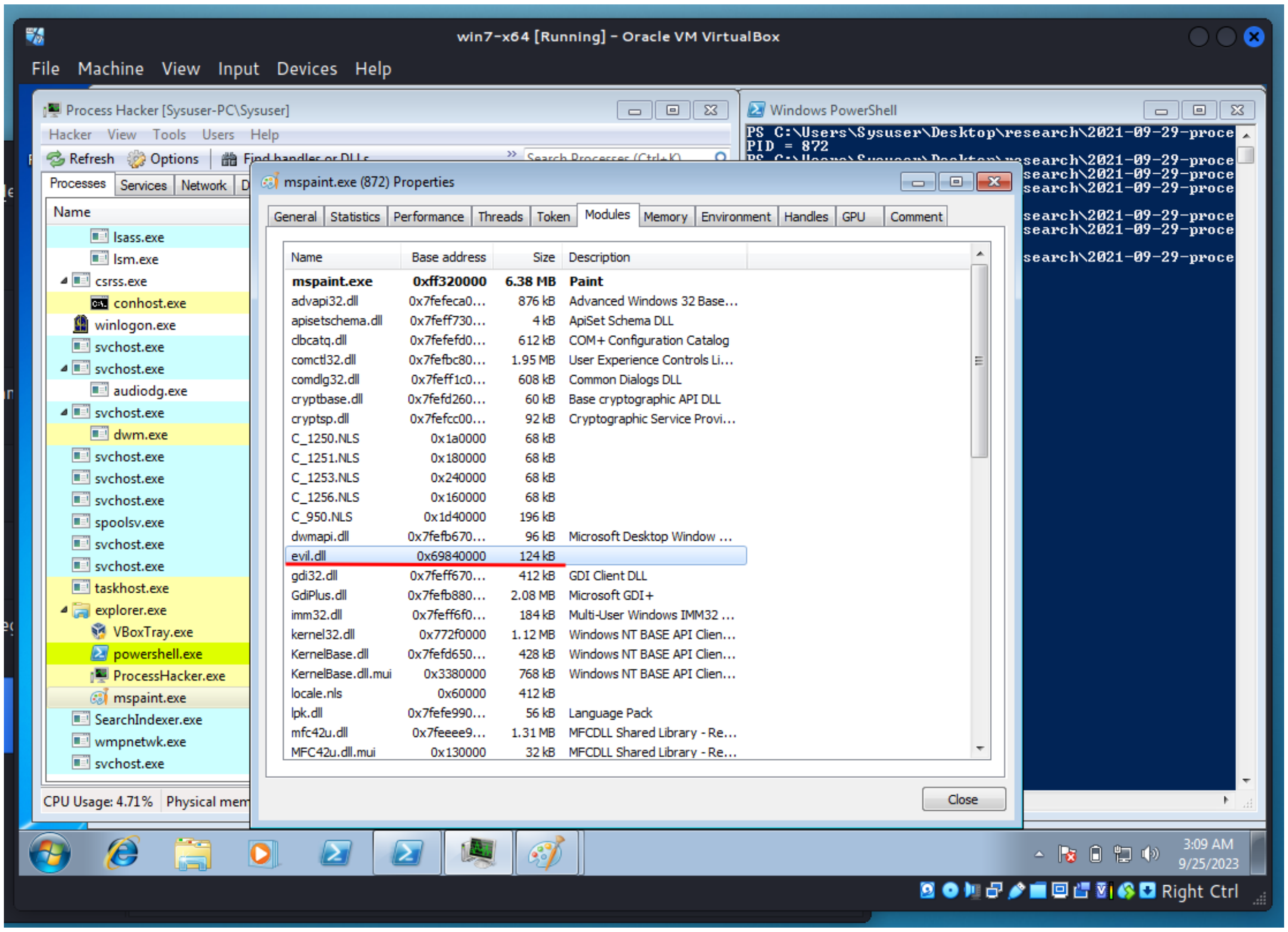
```
name                    base address                    size
=================================================================================
             mspaint.exe              0xff320000                 6688768
               ntdll.dll              0x77410000                 1748992
            kernel32.dll              0x772f0000                 1175552
          KERNELBASE.dll              0x7fefd650000               438272
            ADVAPI32.dll              0x7fefeca0000               897024
              msvcrt.dll              0x7fefeb80000               651264
             sechost.dll              0x7fefee90000               126976
              RPCRT4.dll              0x7feff070000              1236992
               GDI32.dll              0x7feff670000               421888
              USER32.dll              0x771f0000                 1024000
                    LPK.              7fefe990000                  57344
                  USP10.              7fefeeb0000                 827392
                 MFC42u.              7feeee90000                1376256
                  ole32.              7feff260000                2101248
               OLEAUT32.              7fefd740000                 880640
                 ODBC32.              7fef35b0000                 716800
               COMDLG32.              7feff1c0000                 622592
                SHLWAPI.              7feff470000                 462848
               COMCTL32.              7fefbc80000                2048000
                SHELL32.              7fefdc00000               14180352
                PROPSYS.              7fefbb00000                1228800
                  WINMM.              7fef7690000                 241664
                VERSION.              7fefc630000                  49152
               IMM32.DLL              0x7feff6f0000               188416
               MSCTF.dll              0x7fefed80000              1085440
             odbcint.dll              0x750b0000                  229376
             gdiplus.dll              0x7fefb880000              2183168
             uxtheme.dll              0x7fefbaa0000               352256
            MSFTEDIT.DLL              0x7fefb400000               811008
           CRYPTBASE.dll              0x7fefd260000                61440
             CLBCatQ.DLL              0x7fefefd0000               626688
            UIRibbon.dll              0x7feed700000              3878912
             CRYPTSP.dll              0x7fefcc00000                94208
              rsaenh.dll              0x7fefc900000               290816
          RpcRtRemote.dll             0x7fefd350000                81920
                 sti.dll              0x7fef7fb0000               311296
            wiatrace.dll              0x7fefb330000                32768
              dwmapi.dll              0x7fefb670000                98304
         UIRibbonRes.dll              0x73f70000                 1171456
              msxml6.dll              0x7fef97c0000              2031616
              oleacc.dll              0x7fef5580000               344064
                evil.dll              0x69840000                  126976
PS C:\Users\Sysuser\Desktop\research\2023-09-25-malware-trick-36>
```

=^..^=

Meow from evil.dll!

Today, this post is the result of my own research on another popular malware development trick: get list of modules of target process.

Let’s say we created successfully DLL injection to process. How to check if DLL in list of modules of our process?



mspaint.exe (872) Properties

Name	Base address	Size	Description
mspaint.exe	**0xff320000**	**6.38 MB**	**Paint**
advapi32.dll	0x7fefeca0...	876 kB	Advanced Windows 32 Base...
apisetschema.dll	0x7feff730...	4 kB	ApiSet Schema DLL
clbcatq.dll	0x7fefefd0...	612 kB	COM+ Configuration Catalog
comctl32.dll	0x7fefbc80...	1.95 MB	User Experience Controls Li...
comdlg32.dll	0x7feff1c0...	608 kB	Common Dialogs DLL
cryptbase.dll	0x7fefd260...	60 kB	Base cryptographic API DLL
cryptsp.dll	0x7fefcc00...	92 kB	Cryptographic Service Provi...
C_1250.NLS	0x1a0000	68 kB	
C_1251.NLS	0x180000	68 kB	
C_1253.NLS	0x240000	68 kB	
C_1256.NLS	0x160000	68 kB	
C_950.NLS	0x1d40000	196 kB	
dwmapi.dll	0x7fefb670...	96 kB	Microsoft Desktop Window ...
evil.dll	0x69840000	124 kB	
gdi32.dll	0x7feff670...	412 kB	GDI Client DLL
GdiPlus.dll	0x7fefb880...	2.08 MB	Microsoft GDI+
imm32.dll	0x7feff6f0...	184 kB	Multi-User Windows IMM32 ...
kernel32.dll	0x772f0000	1.12 MB	Windows NT BASE API Clien...
KernelBase.dll	0x7fefd650...	428 kB	Windows NT BASE API Clien...
KernelBase.dll.mui	0x3380000	768 kB	Windows NT BASE API Clien...
locale.nls	0x60000	412 kB	
lpk.dll	0x7fefe990...	56 kB	Language Pack
mfc42u.dll	0x7feeee9...	1.31 MB	MFCDLL Shared Library - Re...
MFC42u.dll.mui	0x130000	32 kB	MFCDLL Shared Library - Re...

```
PS C:\Users\Sysuser\Desktop\research\2021-09-29-proce
PID = 872
PS C:\Users\Sysuser\Desktop\research\2021-09-29-proce
search\2021-09-29-proce
search\2021-09-29-proce

search\2021-09-29-proce
search\2021-09-29-proce

search\2021-09-29-proce
```

practical example

First of all, we just use one of the methods to find target process PID. For example I used this one:

```
typedef NTSTATUS (NTAPI * fNtGetNextProcess)(
  _In_ HANDLE ph,
  _In_ ACCESS_MASK DesiredAccess,
  _In_ ULONG HandleAttributes,
  _In_ ULONG Flags,
  _Out_ PHANDLE Newph
);

int findMyProc(const char * procname) {
  int pid = 0;
  HANDLE current = NULL;
  char procName[MAX_PATH];

  // resolve function address
  fNtGetNextProcess myNtGetNextProcess = (fNtGetNextProcess)
GetProcAddress(GetModuleHandle("ntdll.dll"), "NtGetNextProcess");

  // loop through all processes
  while (!myNtGetNextProcess(current, MAXIMUM_ALLOWED, 0, 0, &current)) {
    GetProcessImageFileNameA(current, procName, MAX_PATH);
    if (lstrcmpiA(procname, PathFindFileName((LPCSTR) procName)) == 0) {
      pid = GetProcessId(current);
      break;
    }
  }

  return pid;
}
```

Then, just use `Module32First` and `Module32Next` functions from Windows API.

```
// function to list modules loaded by a specified process
int listModulesOfProcess(int pid) {

  HANDLE mod;
  MODULEENTRY32 me32;

  mod = CreateToolhelp32Snapshot(TH32CS_SNAPMODULE | TH32CS_SNAPMODULE32, pid);
  if (mod == INVALID_HANDLE_VALUE) {
    printf("CreateToolhelp32Snapshot error :(\n");
    return -1;
  }

  me32.dwSize = sizeof(MODULEENTRY32);
  if (!Module32First(mod, &me32)) {
    CloseHandle(mod);
    return -1;
  }

  printf("modules found:\n");
  printf("name\t\t\t base address\t\t\tsize\n");

printf("=================================================================================\n");
  do {
    printf("%#25s\t\t%#10llx\t\t%#10d\n", me32.szModule, me32.modBaseAddr,
me32.modBaseSize);
  } while (Module32Next(mod, &me32));
  CloseHandle(mod);
  return 0;
}
```

As you can see, the code is a bit similar to the PID search logic with `CreateToolHelp32Snapshot`, `Process32First` and `Process32Next`.

So, the full source code is looks like this (`hack.c`):

```
/*
 * hack.c - get the list of modules of the process. C++ implementation
 * @cocomelonc
 * https://cocomelonc.github.io/malware/2023/09/25/malware-tricks-36.html
*/
#include <windows.h>
#include <stdio.h>
#include <winternl.h>
#include <tlhelp32.h>
#include <shlwapi.h>
#include <psapi.h>

#pragma comment(lib, "ntdll.lib")
#pragma comment(lib, "shlwapi.lib")

typedef NTSTATUS (NTAPI * fNtGetNextProcess)(
  _In_ HANDLE ph,
  _In_ ACCESS_MASK DesiredAccess,
  _In_ ULONG HandleAttributes,
  _In_ ULONG Flags,
  _Out_ PHANDLE Newph
);

int findMyProc(const char * procname) {
  int pid = 0;
  HANDLE current = NULL;
  char procName[MAX_PATH];

  // resolve function address
  fNtGetNextProcess myNtGetNextProcess = (fNtGetNextProcess)
GetProcAddress(GetModuleHandle("ntdll.dll"), "NtGetNextProcess");

  // loop through all processes
  while (!myNtGetNextProcess(current, MAXIMUM_ALLOWED, 0, 0, &current)) {
    GetProcessImageFileNameA(current, procName, MAX_PATH);
    if (lstrcmpiA(procname, PathFindFileName((LPCSTR) procName)) == 0) {
      pid = GetProcessId(current);
      break;
    }
  }

  return pid;
}

// function to list modules loaded by a specified process
int listModulesOfProcess(int pid) {

  HANDLE mod;
  MODULEENTRY32 me32;

  mod = CreateToolhelp32Snapshot(TH32CS_SNAPMODULE | TH32CS_SNAPMODULE32, pid);
  if (mod == INVALID_HANDLE_VALUE) {
    printf("CreateToolhelp32Snapshot error :(\n");
    return -1;
  }

  me32.dwSize = sizeof(MODULEENTRY32);
  if (!Module32First(mod, &me32)) {
    CloseHandle(mod);
    return -1;
  }

  printf("modules found:\n");
  printf("name\t\t\t base address\t\t\tsize\n");

printf("=====================================================================
====\n");
  do {
    printf("%#25s\t\t%#10llx\t\t%#10d\n", me32.szModule, me32.modBaseAddr,
me32.modBaseSize);
  } while (Module32Next(mod, &me32));
  CloseHandle(mod);
  return 0;
}

int main(int argc, char* argv[]) {
  int pid = 0; // process ID
  pid = findMyProc(argv[1]);
  printf("%s%d\n", pid > 0 ? "process found at pid = " : "process not found. pid = ",
pid);
  if (pid != 0)
    listModulesOfProcess(pid);
  return 0;
}
```

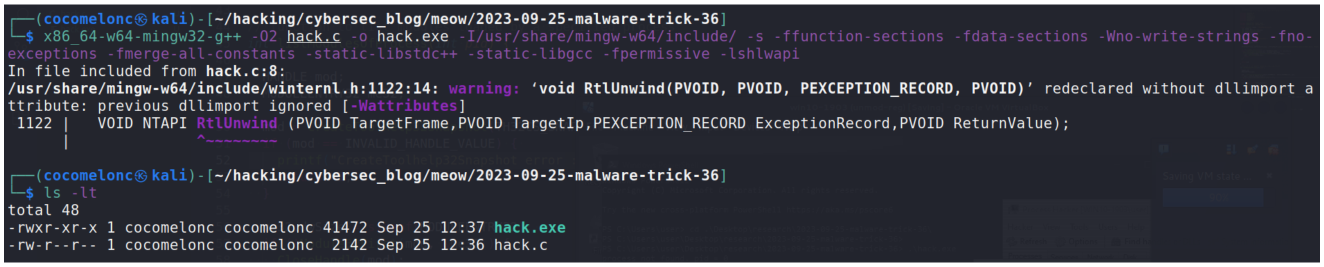
You can use this code to check if a DLL is in the list of modules of the target process.

demo

Let's go to see this logic in action.

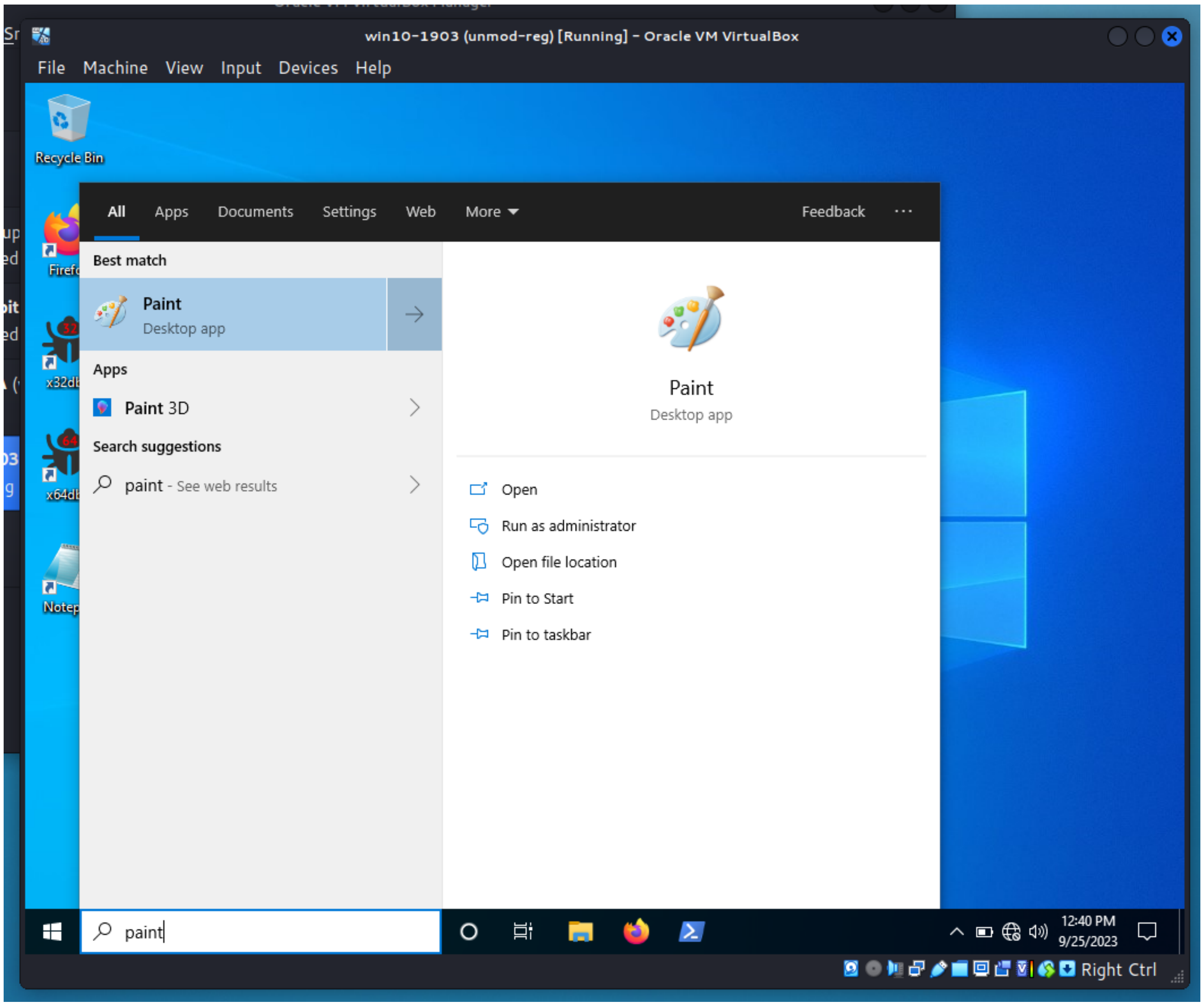
Compile it:

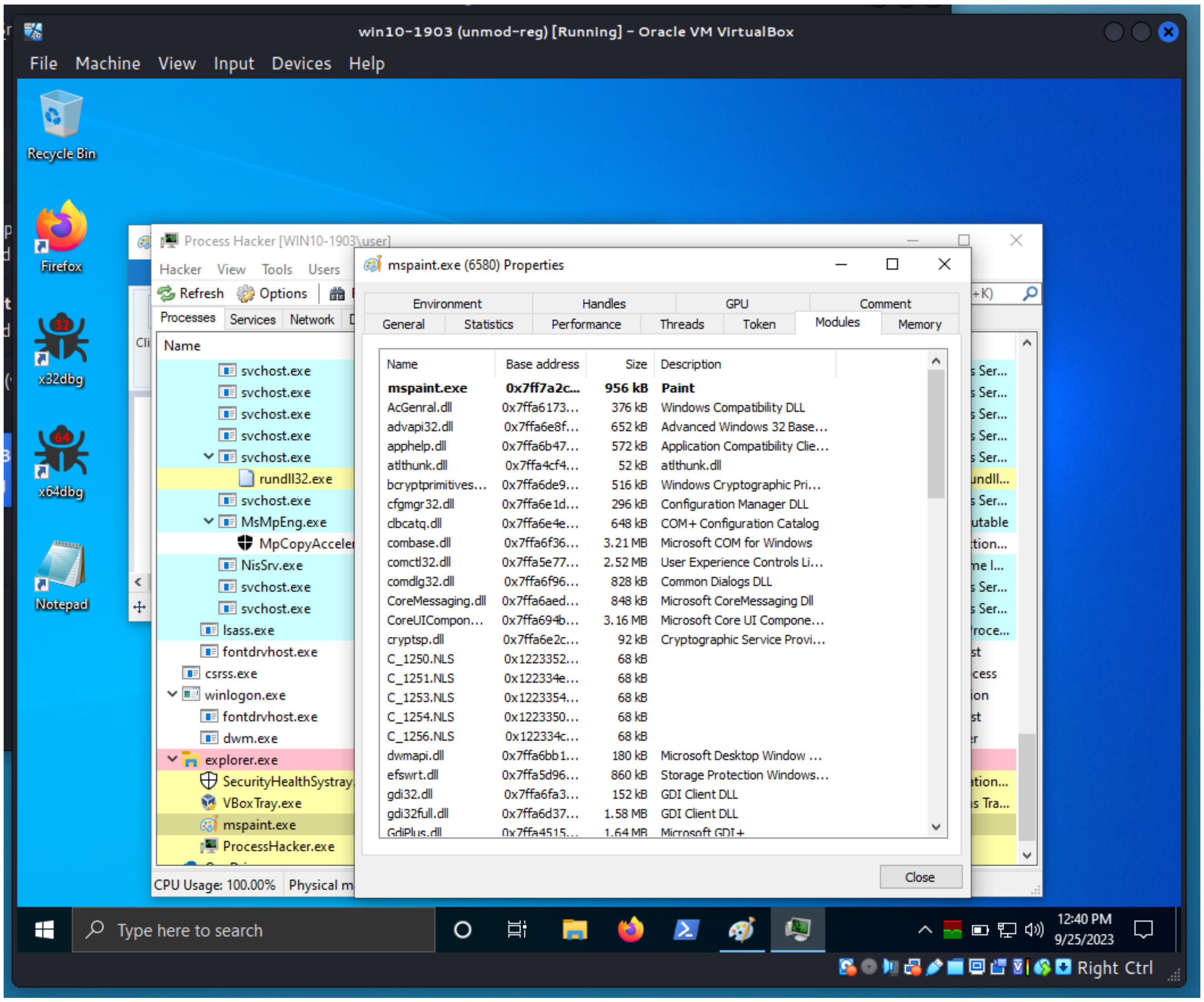
```
x86_64-w64-mingw32-g++ -O2 hack.c -o hack.exe -I/usr/share/mingw-w64/include/ -s -
ffunction-sections -fdata-sections -Wno-write-strings -fno-exceptions -fmerge-all-
constants -static-libstdc++ -static-libgcc -fpermissive -lshlwapi
```



```
┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/meow/2023-09-25-malware-trick-36]
└─$ 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 -lshlwapi
In file included from hack.c:8:
/usr/share/mingw-w64/include/winternl.h:1122:14: warning: 'void RtlUnwind(PVOID, PVOID, PEXCEPTION_RECORD, PVOID)' redeclared without dllimport a
ttribute: previous dllimport ignored [-Wattributes]
 1122 |   VOID NTAPI RtlUnwind (PVOID TargetFrame,PVOID TargetIp,PEXCEPTION_RECORD ExceptionRecord,PVOID ReturnValue);
      |              ^~~~~~~~~

┌──(cocomelonc㉿kali)-[~/hacking/cybersec_blog/meow/2023-09-25-malware-trick-36]
└─$ ls -lt
total 48
-rwxr-xr-x 1 cocomelonc cocomelonc 41472 Sep 25 12:37 hack.exe
-rw-r--r-- 1 cocomelonc cocomelonc  2142 Sep 25 12:36 hack.c
```

Then, open target process in the victim's machine:



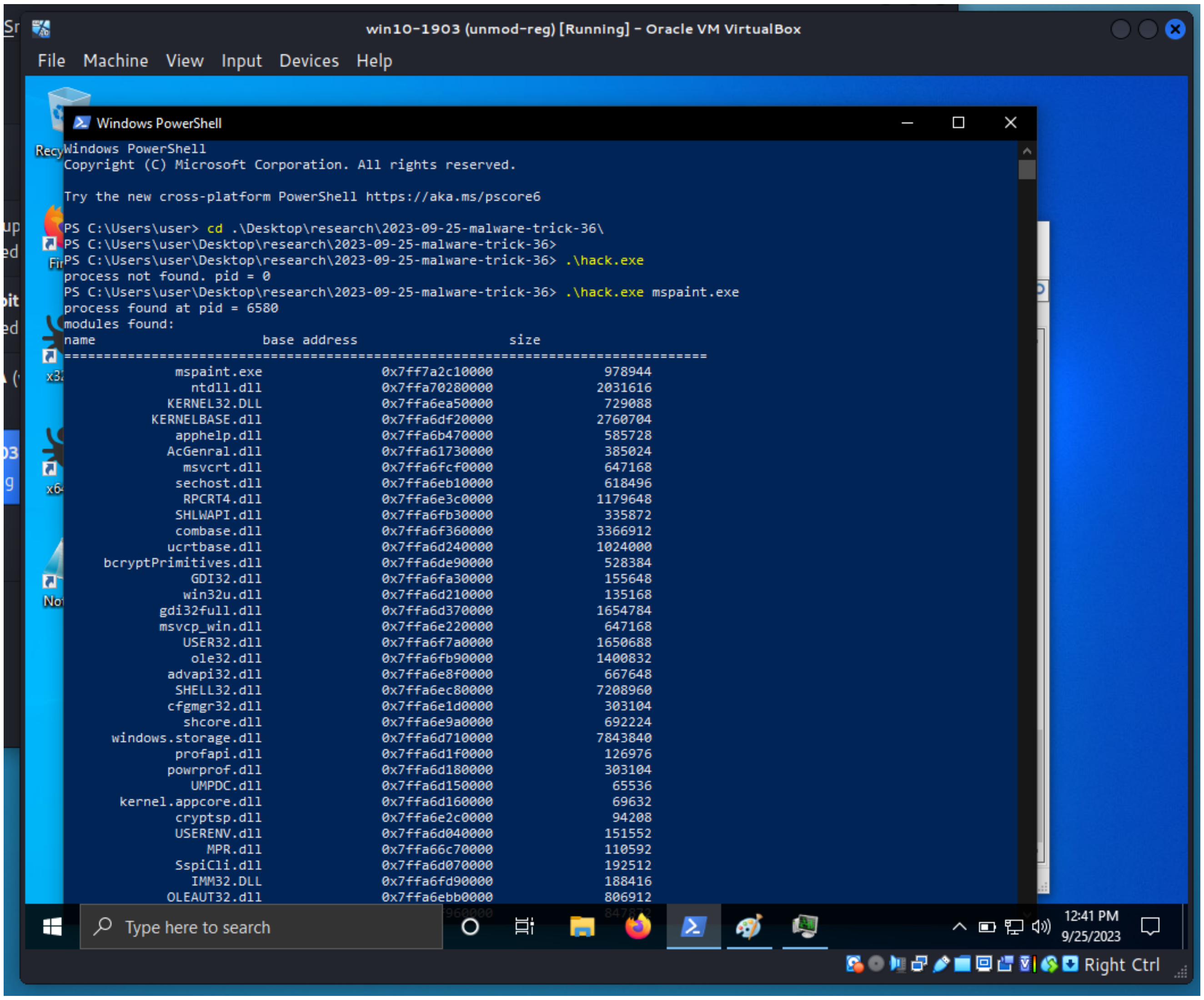

win10-1903 (unmod-reg) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Recycle Bin
Firefox
x32dbg
x64dbg
Notepad
Process Hacker [WIN10-1903\user]
Hacker View Tools Users
Refresh Options
Processes Services Network
mspaint.exe (6580) Properties
Environment Handles GPU Comment
General Statistics Performance Threads Token Modules Memory
Close
Type here to search
12:40 PM
9/25/2023
Right Ctrl

```
svchost.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe
  rundll32.exe
svchost.exe
MsMpEng.exe
  MpCopyAcceler
NisSrv.exe
svchost.exe
svchost.exe
lsass.exe
fontdrvhost.exe
csrss.exe
winlogon.exe
  fontdrvhost.exe
  dwm.exe
explorer.exe
  SecurityHealthSystray
  VBoxTray.exe
  mspaint.exe
  ProcessHacker.exe
CPU Usage: 100.00%   Physical m
```

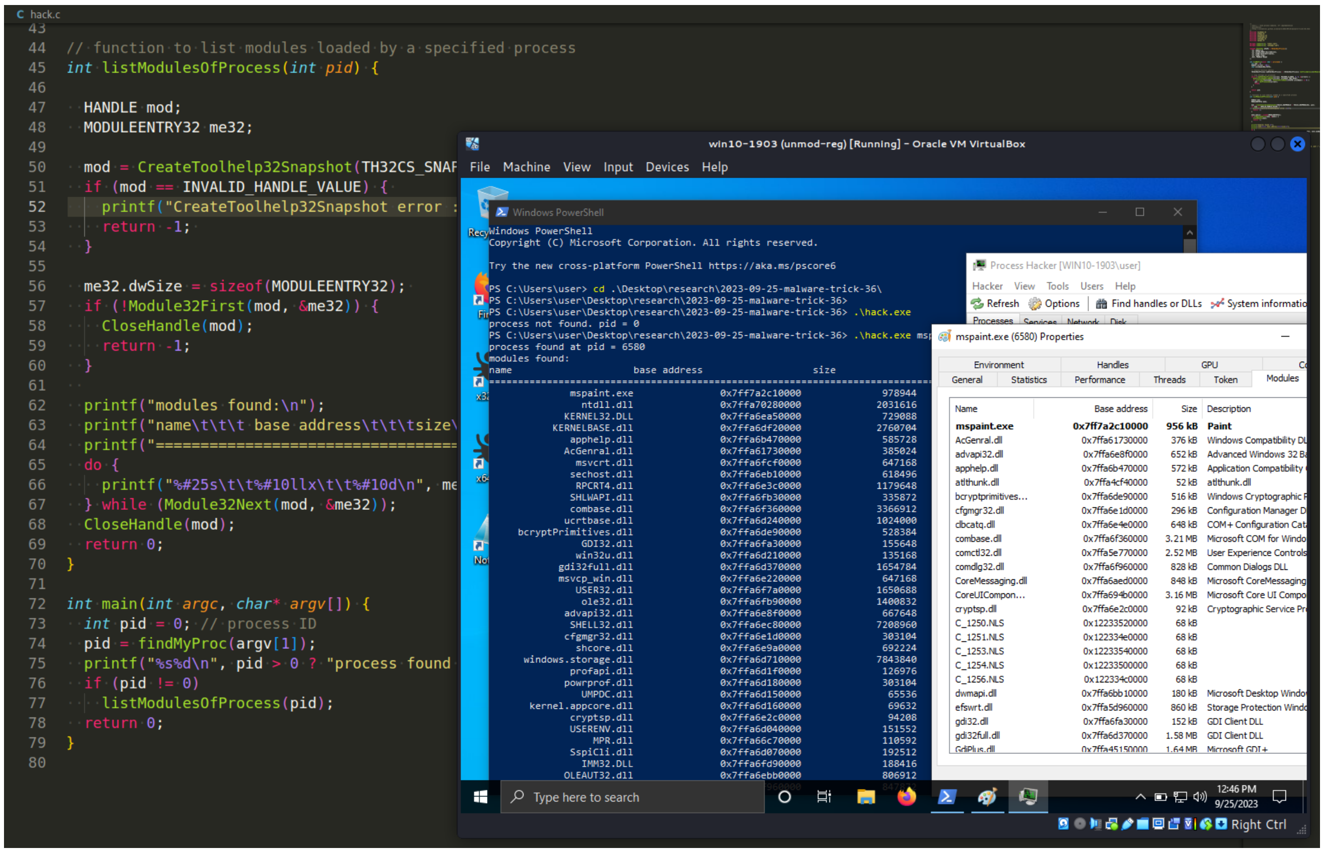
Name	Base address	Size	Description
mspaint.exe	**0x7ff7a2c...**	**956 kB**	**Paint**
AcGenral.dll	0x7ffa6173...	376 kB	Windows Compatibility DLL
advapi32.dll	0x7ffa6e8f...	652 kB	Advanced Windows 32 Base...
apphelp.dll	0x7ffa6b47...	572 kB	Application Compatibility Clie...
atlthunk.dll	0x7ffa4cf4...	52 kB	atlthunk.dll
bcryptprimitives...	0x7ffa6de9...	516 kB	Windows Cryptographic Pri...
cfgmgr32.dll	0x7ffa6e1d...	296 kB	Configuration Manager DLL
clbcatq.dll	0x7ffa6e4e...	648 kB	COM+ Configuration Catalog
combase.dll	0x7ffa6f36...	3.21 MB	Microsoft COM for Windows
comctl32.dll	0x7ffa5e77...	2.52 MB	User Experience Controls Li...
comdlg32.dll	0x7ffa6f96...	828 kB	Common Dialogs DLL
CoreMessaging.dll	0x7ffa6aed...	848 kB	Microsoft CoreMessaging Dll
CoreUIComponen...	0x7ffa694b...	3.16 MB	Microsoft Core UI Compone...
cryptsp.dll	0x7ffa6e2c...	92 kB	Cryptographic Service Provi...
C_1250.NLS	0x1223352...	68 kB	
C_1251.NLS	0x122334e...	68 kB	
C_1253.NLS	0x1223354...	68 kB	
C_1254.NLS	0x1223350...	68 kB	
C_1256.NLS	0x122334c...	68 kB	
dwmapi.dll	0x7ffa6bb1...	180 kB	Microsoft Desktop Window ...
efswrt.dll	0x7ffa5d96...	860 kB	Storage Protection Windows...
gdi32.dll	0x7ffa6fa3...	152 kB	GDI Client DLL
gdi32full.dll	0x7ffa6d37...	1.58 MB	GDI Client DLL
GdiPlus.dll	0x7ffa4515...	1.64 MB	Microsoft GDI+

And just run our `hack.exe`:

```
.\hack.exe mspaint.exe
```


win10-1903 (unmod-reg) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
Type here to search
12:41 PM
9/25/2023
Right Ctrl

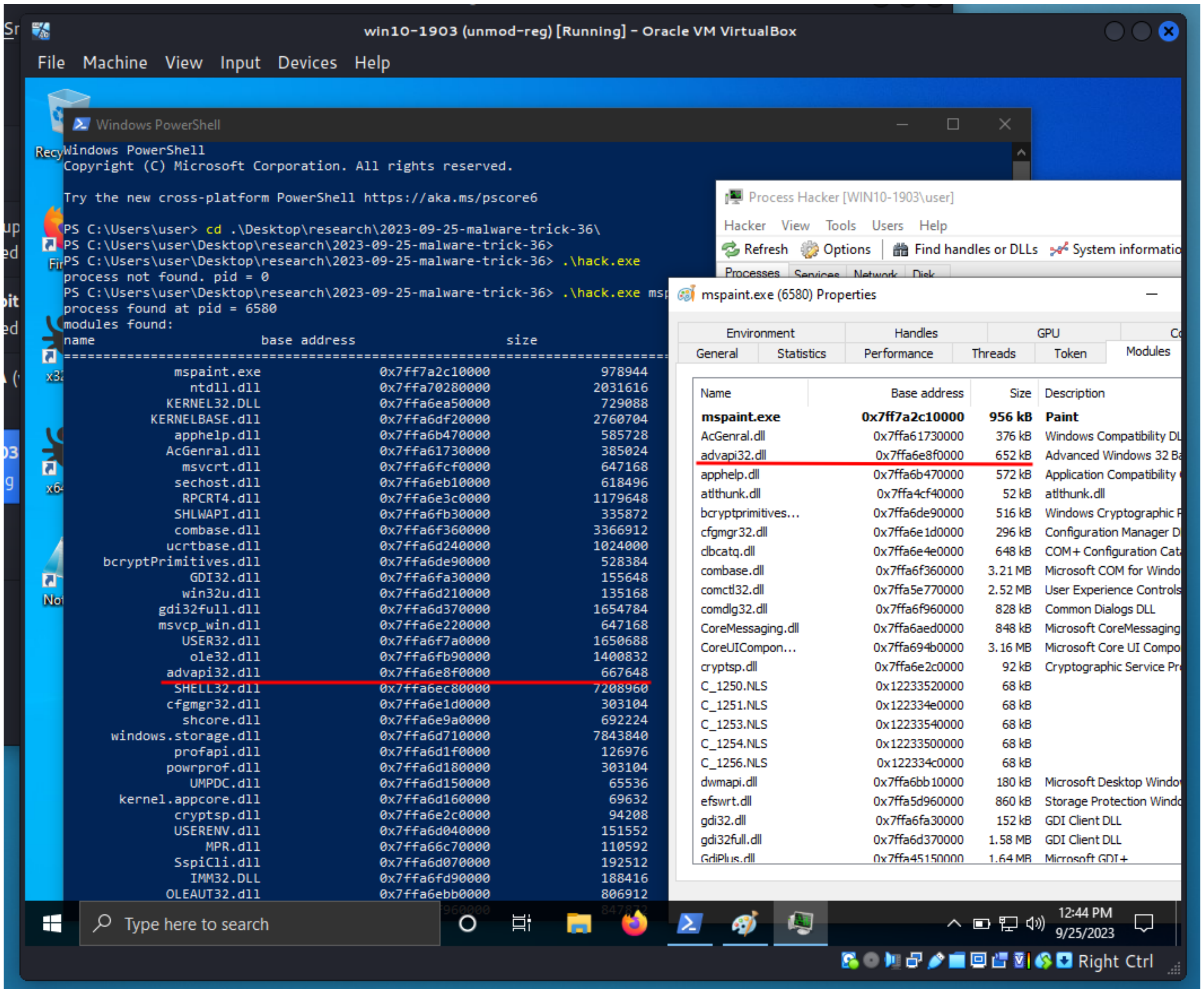
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\user> cd .\Desktop\research\2023-09-25-malware-trick-36\
PS C:\Users\user\Desktop\research\2023-09-25-malware-trick-36>
PS C:\Users\user\Desktop\research\2023-09-25-malware-trick-36> .\hack.exe
process not found. pid = 0
PS C:\Users\user\Desktop\research\2023-09-25-malware-trick-36> .\hack.exe mspaint.exe
process found at pid = 6580
modules found:
name                         base address                    size
=================================================================================
              mspaint.exe              0x7ff7a2c10000             978944
                ntdll.dll              0x7ffa70280000            2031616
             KERNEL32.DLL              0x7ffa6ea50000             729088
           KERNELBASE.dll              0x7ffa6df20000            2760704
              apphelp.dll              0x7ffa6b470000             585728
             AcGenral.dll              0x7ffa61730000             385024
               msvcrt.dll              0x7ffa6fcf0000             647168
              sechost.dll              0x7ffa6eb10000             618496
               RPCRT4.dll              0x7ffa6e3c0000            1179648
              SHLWAPI.dll              0x7ffa6fb30000             335872
              combase.dll              0x7ffa6f360000            3366912
             ucrtbase.dll              0x7ffa6d240000            1024000
     bcryptPrimitives.dll              0x7ffa6de90000             528384
                GDI32.dll              0x7ffa6fa30000             155648
               win32u.dll              0x7ffa6d210000             135168
            gdi32full.dll              0x7ffa6d370000            1654784
            msvcp_win.dll              0x7ffa6e220000             647168
               USER32.dll              0x7ffa6f7a0000            1650688
                ole32.dll              0x7ffa6fb90000            1400832
             advapi32.dll              0x7ffa6e8f0000             667648
              SHELL32.dll              0x7ffa6ec80000            7208960
             cfgmgr32.dll              0x7ffa6e1d0000             303104
               shcore.dll              0x7ffa6e9a0000             692224
      windows.storage.dll              0x7ffa6d710000            7843840
              profapi.dll              0x7ffa6d1f0000             126976
             powrprof.dll              0x7ffa6d180000             303104
                UMPDC.dll              0x7ffa6d150000              65536
      kernel.appcore.dll              0x7ffa6d160000              69632
              cryptsp.dll              0x7ffa6e2c0000              94208
              USERENV.dll              0x7ffa6d040000             151552
                  MPR.dll              0x7ffa66c70000             110592
              SspiCli.dll              0x7ffa6d070000             192512
                IMM32.DLL              0x7ffa6fd90000             188416
             OLEAUT32.dll              0x7ffa6ebb0000             806912
```


hack.c
win10-1903 (unmod-reg) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
Process Hacker [WIN10-1903\user]
Hacker View Tools Users Help
Refresh Options Find handles or DLLs System informatio
Processes Services Network Disk
mspaint.exe (6580) Properties
Environment Handles GPU
General Statistics Performance Threads Token Modules
Type here to search
12:46 PM
9/25/2023
Right Ctrl

```c

// function to list modules loaded by a specified process
int listModulesOfProcess(int pid) {

  HANDLE mod;
  MODULEENTRY32 me32;

  mod = CreateToolhelp32Snapshot(TH32CS_SNAF
  if (mod == INVALID_HANDLE_VALUE) {
    printf("CreateToolhelp32Snapshot error :
    return -1;
  }

  me32.dwSize = sizeof(MODULEENTRY32);
  if (!Module32First(mod, &me32)) {
    CloseHandle(mod);
    return -1;
  }

  printf("modules found:\n");
  printf("name\t\t\t base address\t\t\tsize\
  printf("====================================
  do {
    printf("%#25s\t\t%#10llx\t\t%#10d\n", me
  } while (Module32Next(mod, &me32));
  CloseHandle(mod);
  return 0;
}

int main(int argc, char* argv[]) {
  int pid = 0; // process ID
  pid = findMyProc(argv[1]);
  printf("%s%d\n", pid > 0 ? "process found
  if (pid != 0)
    listModulesOfProcess(pid);
  return 0;
}

```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\user> cd .\Desktop\research\2023-09-25-malware-trick-36\
PS C:\Users\user\Desktop\research\2023-09-25-malware-trick-36>
PS C:\Users\user\Desktop\research\2023-09-25-malware-trick-36> .\hack.exe
process not found. pid = 0
PS C:\Users\user\Desktop\research\2023-09-25-malware-trick-36> .\hack.exe msp
process found at pid = 6580
modules found:
name                         base address                  size
==================================================================================
                  mspaint.exe               0x7ff7a2c10000             978944
                    ntdll.dll               0x7ffa70280000            2031616
                 KERNEL32.DLL               0x7ffa6ea50000             729088
               KERNELBASE.dll               0x7ffa6df20000            2760704
                  apphelp.dll               0x7ffa6b470000             585728
                  AcGenral.dll              0x7ffa61730000             385024
                   msvcrt.dll               0x7ffa6fcf0000             647168
                  sechost.dll               0x7ffa6eb10000             618496
                   RPCRT4.dll               0x7ffa6e3c0000            1179648
                  SHLWAPI.dll               0x7ffa6fb30000             335872
                  combase.dll               0x7ffa6f360000            3366912
                 ucrtbase.dll               0x7ffa6d240000            1024000
       bcryptPrimitives.dll               0x7ffa6de90000             528384
                    GDI32.dll               0x7ffa6fa30000             155648
                   win32u.dll               0x7ffa6d210000             135168
                gdi32full.dll               0x7ffa6d370000            1654784
                msvcp_win.dll               0x7ffa6e220000             647168
                   USER32.dll               0x7ffa6f7a0000            1650688
                    ole32.dll               0x7ffa6fb90000            1400832
                 advapi32.dll               0x7ffa6e8f0000             667648
                  SHELL32.dll               0x7ffa6ec80000            7208960
                 cfgmgr32.dll               0x7ffa6e1d0000             303104
                   shcore.dll               0x7ffa6e9a0000             692224
         windows.storage.dll               0x7ffa6d710000            7843840
                  profapi.dll               0x7ffa6d1f0000             126976
                 powrprof.dll               0x7ffa6d180000             303104
                    UMPDC.dll               0x7ffa6d150000              65536
           kernel.appcore.dll               0x7ffa6d160000              69632
                  cryptsp.dll               0x7ffa6e2c0000              94208
                  USERENV.dll               0x7ffa6d040000             151552
                      MPR.dll               0x7ffa66c70000             110592
                  SspiCli.dll               0x7ffa6d070000             192512
                   IMM32.DLL               0x7ffa6fd90000             188416
                 OLEAUT32.dll               0x7ffa6ebb0000             806912
```

Name	Base address	Size	Description
mspaint.exe	**0x7ff7a2c10000**	**956 kB**	**Paint**
AcGenral.dll	0x7ffa61730000	376 kB	Windows Compatibility DL
advapi32.dll	0x7ffa6e8f0000	652 kB	Advanced Windows 32 Ba
apphelp.dll	0x7ffa6b470000	572 kB	Application Compatibility
atlthunk.dll	0x7ffa4cf40000	52 kB	atlthunk.dll
bcryptprimitives...	0x7ffa6de90000	516 kB	Windows Cryptographic F
cfgmgr32.dll	0x7ffa6e1d0000	296 kB	Configuration Manager D
clbcatq.dll	0x7ffa6e4e0000	648 kB	COM+ Configuration Cata
combase.dll	0x7ffa6f360000	3.21 MB	Microsoft COM for Windo
comctl32.dll	0x7ffa5e770000	2.52 MB	User Experience Controls
comdlg32.dll	0x7ffa6f960000	828 kB	Common Dialogs DLL
CoreMessaging.dll	0x7ffa6aed0000	848 kB	Microsoft CoreMessaging
CoreUICompon...	0x7ffa694b0000	3.16 MB	Microsoft Core UI Compo
cryptsp.dll	0x7ffa6e2c0000	92 kB	Cryptographic Service Pr
C_1250.NLS	0x12233520000	68 kB	
C_1251.NLS	0x122334e0000	68 kB	
C_1253.NLS	0x12233540000	68 kB	
C_1254.NLS	0x12233500000	68 kB	
C_1256.NLS	0x122334c0000	68 kB	
dwmapi.dll	0x7ffa6bb10000	180 kB	Microsoft Desktop Windo
efswrt.dll	0x7ffa5d960000	860 kB	Storage Protection Windo
gdi32.dll	0x7ffa6fa30000	152 kB	GDI Client DLL
gdi32full.dll	0x7ffa6d370000	1.58 MB	GDI Client DLL
GdiPlus.dll	0x7ffa45150000	1.64 MB	Microsoft GDI+

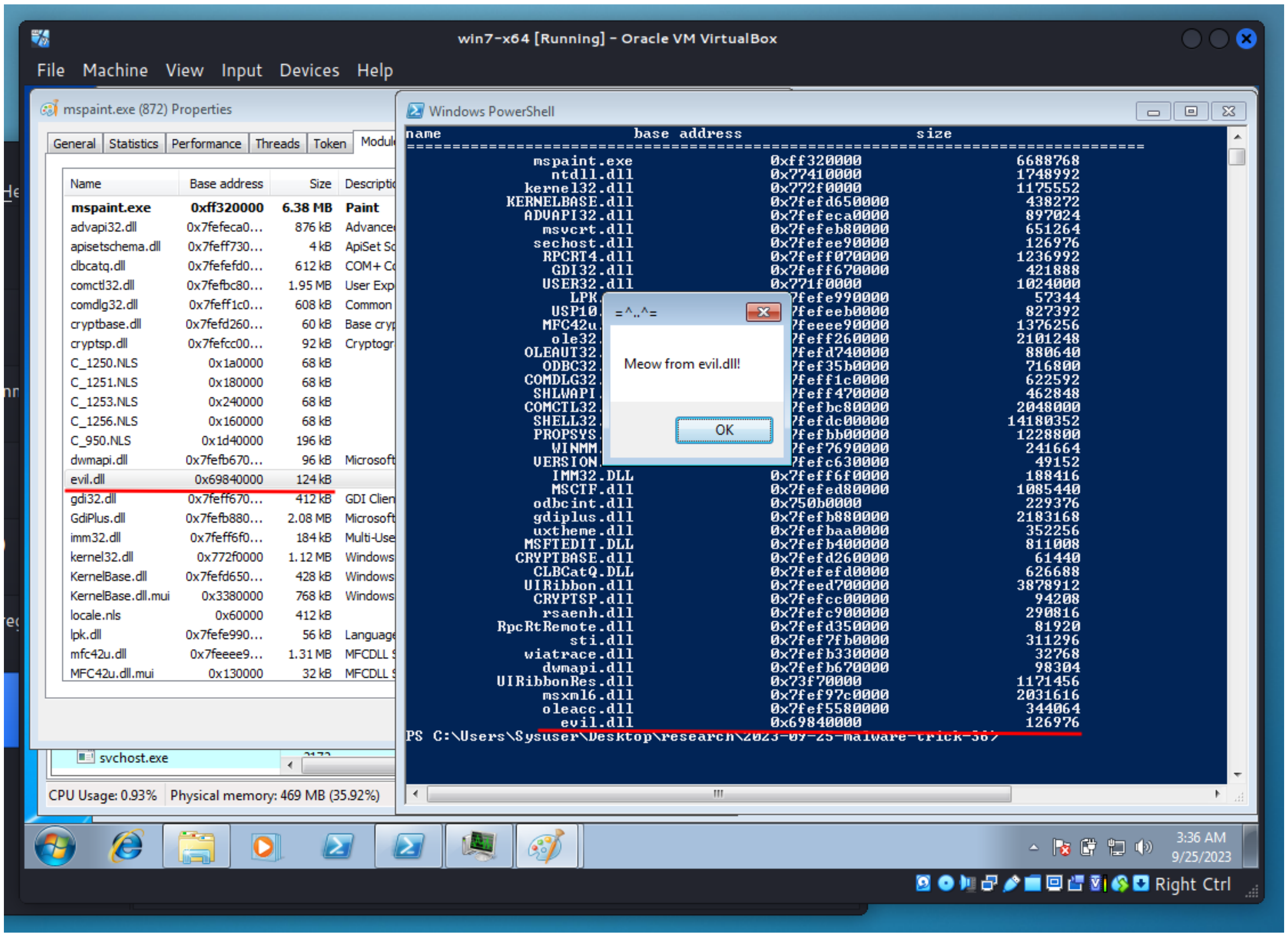


win10-1903 (unmod-reg) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
Process Hacker [WIN10-1903\user]
Hacker View Tools Users Help
Refresh Options Find handles or DLLs System informatio
mspaint.exe (6580) Properties
Environment Handles GPU
General Statistics Performance Threads Token Modules
Type here to search
12:44 PM
9/25/2023
Right Ctrl

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\user> cd .\Desktop\research\2023-09-25-malware-trick-36\
PS C:\Users\user\Desktop\research\2023-09-25-malware-trick-36>
PS C:\Users\user\Desktop\research\2023-09-25-malware-trick-36> .\hack.exe
process not found. pid = 0
PS C:\Users\user\Desktop\research\2023-09-25-malware-trick-36> .\hack.exe msp
process found at pid = 6580
modules found:
name                         base address                  size
=============================================================================
              mspaint.exe              0x7ff7a2c10000            978944
                ntdll.dll              0x7ffa70280000           2031616
             KERNEL32.DLL              0x7ffa6ea50000            729088
           KERNELBASE.dll              0x7ffa6df20000           2760704
              apphelp.dll              0x7ffa6b470000            585728
             AcGenral.dll              0x7ffa61730000            385024
               msvcrt.dll              0x7ffa6fcf0000            647168
              sechost.dll              0x7ffa6eb10000            618496
               RPCRT4.dll              0x7ffa6e3c0000           1179648
              SHLWAPI.dll              0x7ffa6fb30000            335872
              combase.dll              0x7ffa6f360000           3366912
             ucrtbase.dll              0x7ffa6d240000           1024000
      bcryptPrimitives.dll              0x7ffa6de90000            528384
                GDI32.dll              0x7ffa6fa30000            155648
               win32u.dll              0x7ffa6d210000            135168
             gdi32full.dll             0x7ffa6d370000           1654784
             msvcp_win.dll             0x7ffa6e220000            647168
               USER32.dll              0x7ffa6f7a0000           1650688
                ole32.dll              0x7ffa6fb90000           1400832
             advapi32.dll              0x7ffa6e8f0000            667648
               SHELL32.dll             0x7ffa6ec80000           7208960
             cfgmgr32.dll              0x7ffa6e1d0000            303104
               shcore.dll              0x7ffa6e9a0000            692224
       windows.storage.dll              0x7ffa6d710000           7843840
              profapi.dll              0x7ffa6d1f0000            126976
             powrprof.dll              0x7ffa6d180000            303104
                UMPDC.dll              0x7ffa6d150000             65536
        kernel.appcore.dll              0x7ffa6d160000             69632
              cryptsp.dll              0x7ffa6e2c0000             94208
              USERENV.dll              0x7ffa6d040000            151552
                  MPR.dll              0x7ffa66c70000            110592
               SspiCli.dll              0x7ffa6d070000            192512
                IMM32.DLL              0x7ffa6fd90000            188416
             OLEAUT32.dll              0x7ffa6ebb0000            806912
```

Name	Base address	Size	Description
mspaint.exe	**0x7ff7a2c10000**	**956 kB**	**Paint**
AcGenral.dll	0x7ffa61730000	376 kB	Windows Compatibility DL
advapi32.dll	0x7ffa6e8f0000	652 kB	Advanced Windows 32 Ba
apphelp.dll	0x7ffa6b470000	572 kB	Application Compatibility
atlthunk.dll	0x7ffa4cf40000	52 kB	atlthunk.dll
bcryptprimitives…	0x7ffa6de90000	516 kB	Windows Cryptographic P
cfgmgr32.dll	0x7ffa6e1d0000	296 kB	Configuration Manager D
clbcatq.dll	0x7ffa6e4e0000	648 kB	COM+ Configuration Cata
combase.dll	0x7ffa6f360000	3.21 MB	Microsoft COM for Windo
comctl32.dll	0x7ffa5e770000	2.52 MB	User Experience Controls
comdlg32.dll	0x7ffa6f960000	828 kB	Common Dialogs DLL
CoreMessaging.dll	0x7ffa6aed0000	848 kB	Microsoft CoreMessaging
CoreUICompon…	0x7ffa694b0000	3.16 MB	Microsoft Core UI Compo
cryptsp.dll	0x7ffa6e2c0000	92 kB	Cryptographic Service Pr
C_1250.NLS	0x12233520000	68 kB	
C_1251.NLS	0x122334e0000	68 kB	
C_1253.NLS	0x12233540000	68 kB	
C_1254.NLS	0x12233500000	68 kB	
C_1256.NLS	0x122334c0000	68 kB	
dwmapi.dll	0x7ffa6bb10000	180 kB	Microsoft Desktop Windo
efswrt.dll	0x7ffa5d960000	860 kB	Storage Protection Windo
gdi32.dll	0x7ffa6fa30000	152 kB	GDI Client DLL
gdi32full.dll	0x7ffa6d370000	1.58 MB	GDI Client DLL
GdiPlus.dll	0x7ffa45150000	1.64 MB	Microsoft GDI+

Also, check with DLL injection logic:


win7-x64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
mspaint.exe (872) Properties
General Statistics Performance Threads Token Modul
svchost.exe
CPU Usage: 0.93% Physical memory: 469 MB (35.92%)
Windows PowerShell
=^..^=
OK
3:36 AM
9/25/2023
Right Ctrl

Name	Base address	Size	Descriptio
mspaint.exe	**0xff320000**	**6.38 MB**	**Paint**
advapi32.dll	0x7fefeca0...	876 kB	Advance
apisetschema.dll	0x7feff730...	4 kB	ApiSet Sc
clbcatq.dll	0x7fefefd0...	612 kB	COM+ Co
comctl32.dll	0x7fefbc80...	1.95 MB	User Exp
comdlg32.dll	0x7feff1c0...	608 kB	Common
cryptbase.dll	0x7fefd260...	60 kB	Base cryp
cryptsp.dll	0x7fefcc00...	92 kB	Cryptogr
C_1250.NLS	0x1a0000	68 kB	
C_1251.NLS	0x180000	68 kB	
C_1253.NLS	0x240000	68 kB	
C_1256.NLS	0x160000	68 kB	
C_950.NLS	0x1d40000	196 kB	
dwmapi.dll	0x7fefb670...	96 kB	Microsoft
evil.dll	0x69840000	124 kB	
gdi32.dll	0x7feff670...	412 kB	GDI Clien
GdiPlus.dll	0x7fefb880...	2.08 MB	Microsoft
imm32.dll	0x7feff6f0...	184 kB	Multi-Use
kernel32.dll	0x772f0000	1.12 MB	Windows
KernelBase.dll	0x7fefd650...	428 kB	Windows
KernelBase.dll.mui	0x3380000	768 kB	Windows
locale.nls	0x60000	412 kB	
lpk.dll	0x7fefe990...	56 kB	Language
mfc42u.dll	0x7feeee9...	1.31 MB	MFCDLL S
MFC42u.dll.mui	0x130000	32 kB	MFCDLL S

```
name                      base address                     size
============================================================================
              mspaint.exe              0xff320000                  6688768
                ntdll.dll              0x77410000                  1748992
             kernel32.dll              0x772f0000                  1175552
           KERNELBASE.dll              0x7fefd650000                438272
             ADVAPI32.dll              0x7fefeca0000                897024
               msvcrt.dll              0x7fefeb80000                651264
              sechost.dll              0x7fefee90000                126976
               RPCRT4.dll              0x7feff070000               1236992
                GDI32.dll              0x7feff670000                421888
               USER32.dll              0x771f0000                  1024000
                  LPK.                 7fefe990000                  57344
                USP10.                 7fefeeb0000                 827392
               MFC42u.                 7feeee90000                1376256
                ole32.                 7feff260000                2101248
             OLEAUT32.                 7fefd740000                 880640
               ODBC32.                 7fef35b0000                 716800
             COMDLG32.                 7feff1c0000                 622592
              SHLWAPI.                 7feff470000                 462848
             COMCTL32.                 7fefbc80000                2048000
              SHELL32.                 7fefdc00000               14180352
              PROPSYS.                 7fefbb00000                1228800
                WINMM.                 7fef7690000                 241664
              VERSION.                 7fefc630000                  49152
                IMM32.DLL              0x7feff6f0000                188416
                MSCTF.dll              0x7fefed80000               1085440
              odbcint.dll              0x750b0000                  229376
              gdiplus.dll              0x7fefb880000               2183168
              uxtheme.dll              0x7fefbaa0000                352256
             MSFTEDIT.DLL              0x7fefb400000                811008
            CRYPTBASE.dll              0x7fefd260000                 61440
             CLBCatQ.DLL               0x7fefefd0000                626688
             UIRibbon.dll              0x7feed700000               3878912
              CRYPTSP.dll              0x7fefcc00000                 94208
               rsaenh.dll              0x7fefc900000                290816
           RpcRtRemote.dll             0x7fefd350000                 81920
                  sti.dll              0x7fef7fb0000                311296
             wiatrace.dll              0x7fefb330000                 32768
               dwmapi.dll              0x7fefb670000                 98304
           UIRibbonRes.dll             0x73f70000                  1171456
               msxml6.dll              0x7fef97c0000               2031616
               oleacc.dll              0x7fef5580000                344064
                 evil.dll              0x69840000                   126976
PS C:\Users\Sysuser\Desktop\research\2023-09-25-malware-trick-36>
```

Meow from evil.dll!

As you can see, everything is worked perfectly! =^..^=

Keep in mind that this code may have limitations and dependencies on specific Windows APIs. Additionally, it relies on the process name for identification, which may not be unique.

This trick is used by 4H RAT and Aria-body in the wild.

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

Find process ID by name and inject to it
Find PID via NtGetNextProcess
4H RAT
Aria-body
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