

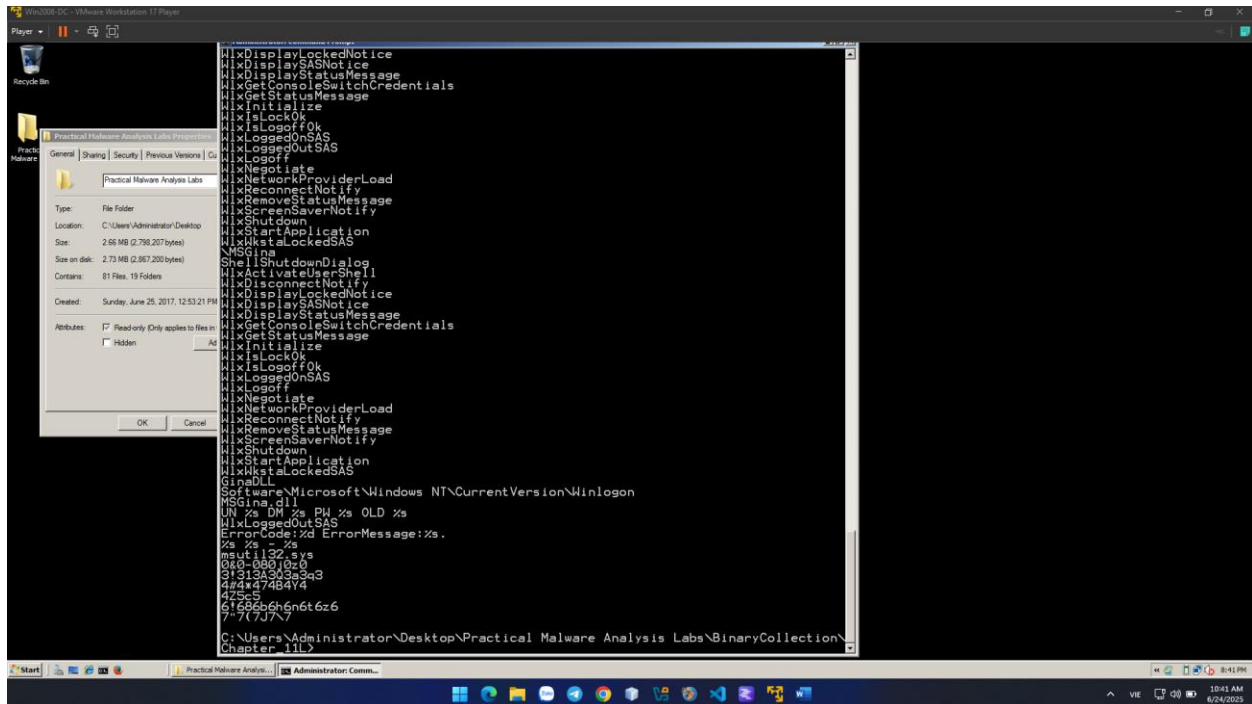
# LAB 12 Dynamic Analysis Tools

## **LAB 1:**

What you need: The Windows 2008 Server virtual machine we have been using. Purpose: Analyze malware behavior

## Static Analysis with Strings

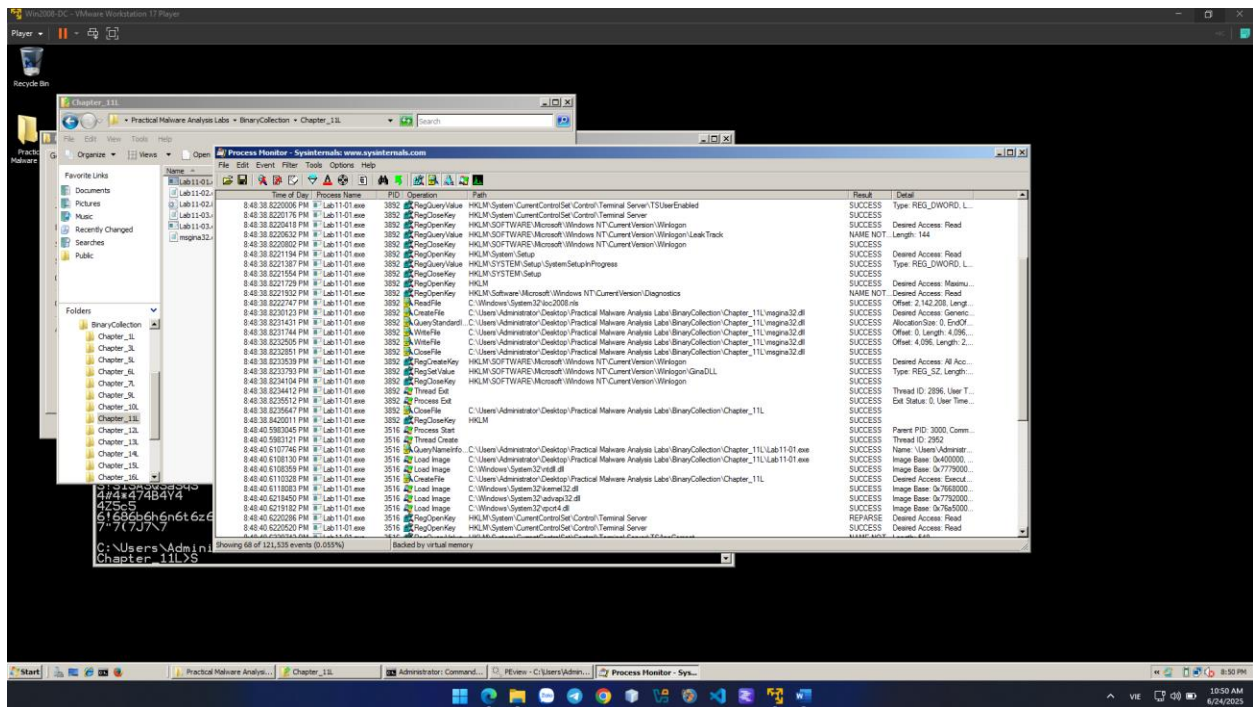
Examine the strings in Lab11-01.exe.



Static Analysis with PView Examine the Lab11-01.exe file in PView. Find the items below.

- RegSetValueExA • RegCreateKeyExA • SizeofResource • LockResource • LoadResource



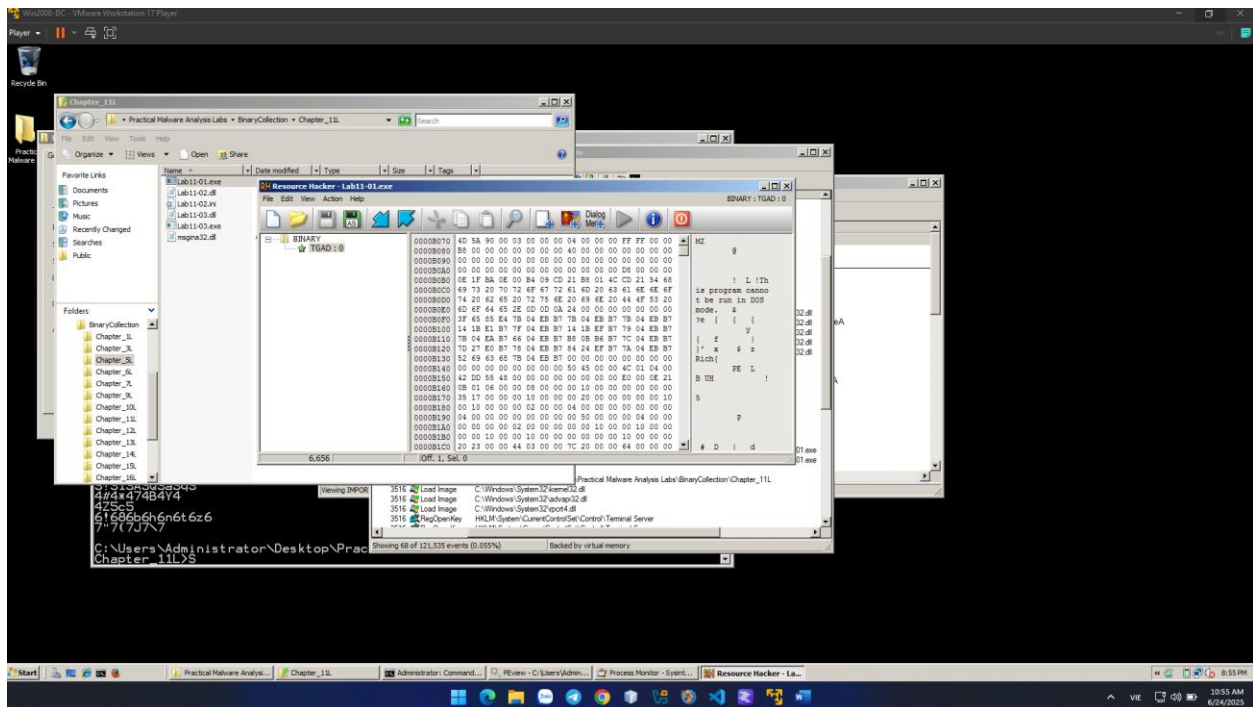


8:48:38.8221932 PM	Lab11-01.exe	3892	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnos	NAME NOT
8:48:38.8222747 PM	Lab11-01.exe	3892	ReadFile	C:\Windows\System32\loc2008.nls	SUCCESS
8:48:38.8230123 PM	Lab11-01.exe	3892	CreateFile	C:\Users\Administrator\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_11\msgina32.dll	SUCCESS
8:48:38.8231431 PM	Lab11-01.exe	3892	QueryStandard...	C:\Users\Administrator\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_11\msgina32.dll	SUCCESS
8:48:38.8231744 PM	Lab11-01.exe	3892	WriteFile	C:\Users\Administrator\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_11\msgina32.dll	SUCCESS
8:48:38.8232505 PM	Lab11-01.exe	3892	WriteFile	C:\Users\Administrator\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_11\msgina32.dll	SUCCESS
8:48:38.8232851 PM	Lab11-01.exe	3892	CloseFile	C:\Users\Administrator\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_11\msgina32.dll	SUCCESS
8:48:38.8233539 PM	Lab11-01.exe	3892	RegCreateKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS
8:48:38.8233793 PM	Lab11-01.exe	3892	RegSetValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL	SUCCESS
8:48:38.8234104 PM	Lab11-01.exe	3892	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS
8:48:38.8234412 PM	Lab11-01.exe	3892	Thread Exit		SUCCESS
8:48:38.8235512 PM	Lab11-01.exe	3892	Process Exit		SUCCESS

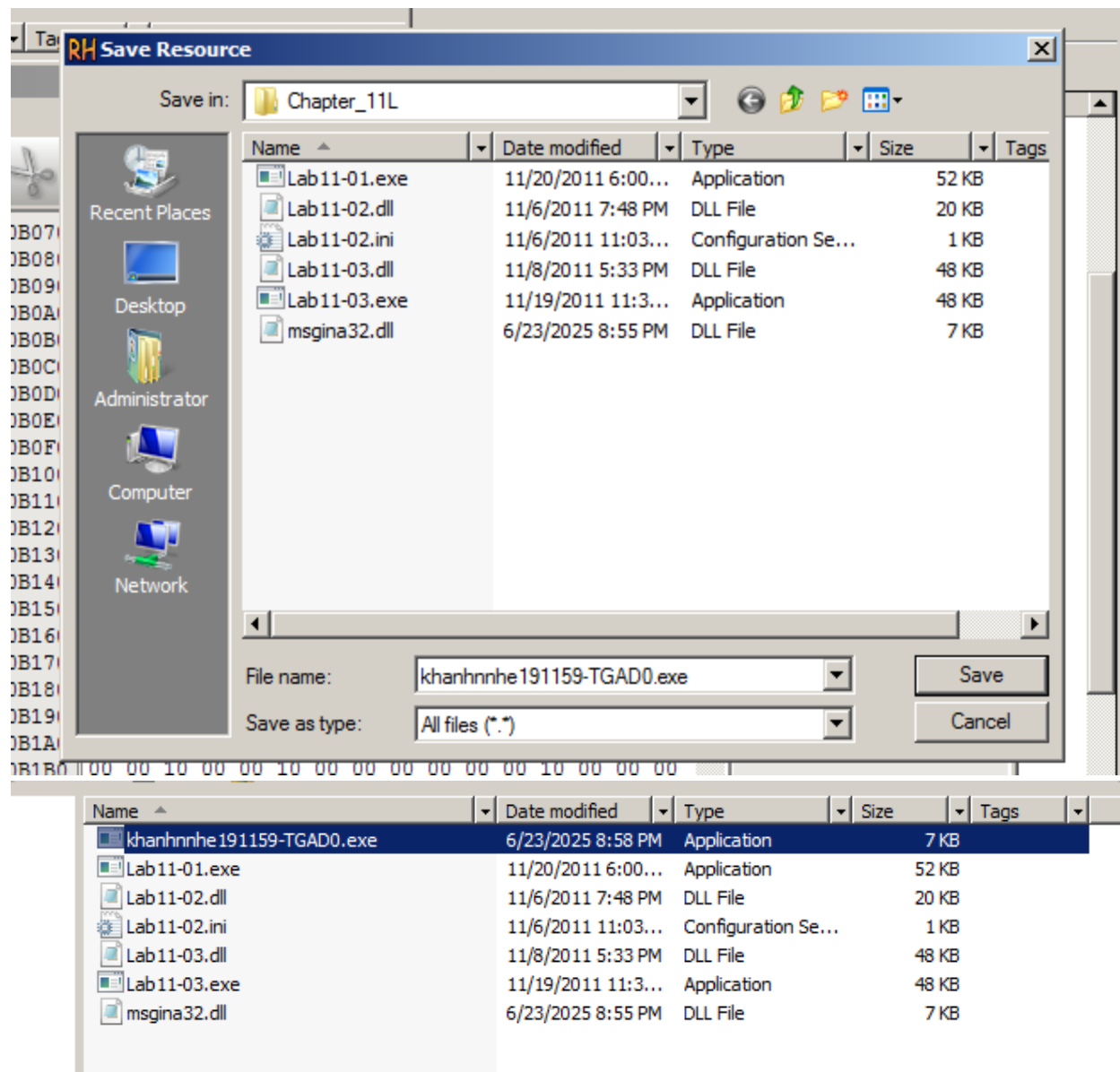
## Resource Hacker

Open Lab11-01.exe in Resource Hacker. The "BINARY TGAD 0" starts with MZ and contains the telltale text "This program cannot be run in DOS mode", as shown below--this is an EXE

file.

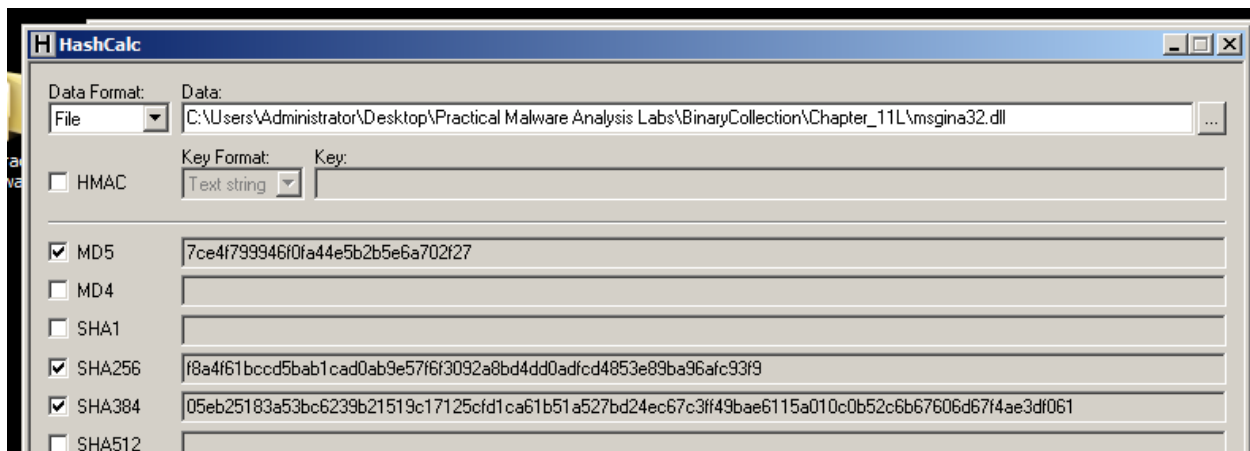


In Resource Hacker, in the left pane, click 0 ti to highlight it, as shown above. Click Action, Save Resource as a binary file...". Save the file as YOURNAME-TGAD0.exe, replacing the text "YOURNAME" with your own name.

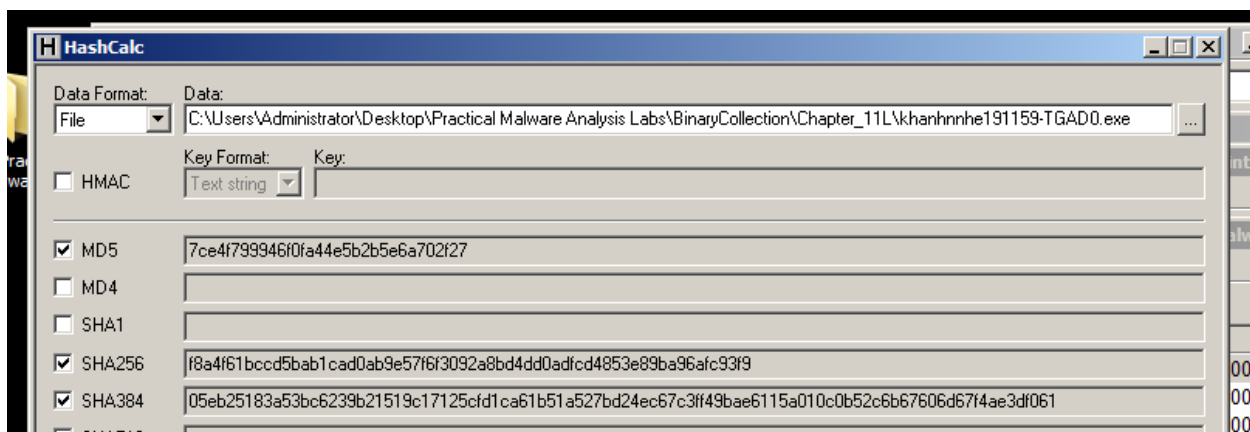


## HashCalc

Calculate the MD5 hash of the msgina32.dll file created by running the malware. The MD5 hash begins with 7ce4, as shown below.



Calculate the MD5 hash of the khanhnnhe191159-TGAD0.exe file, as shown below.



## **LAB 2:**

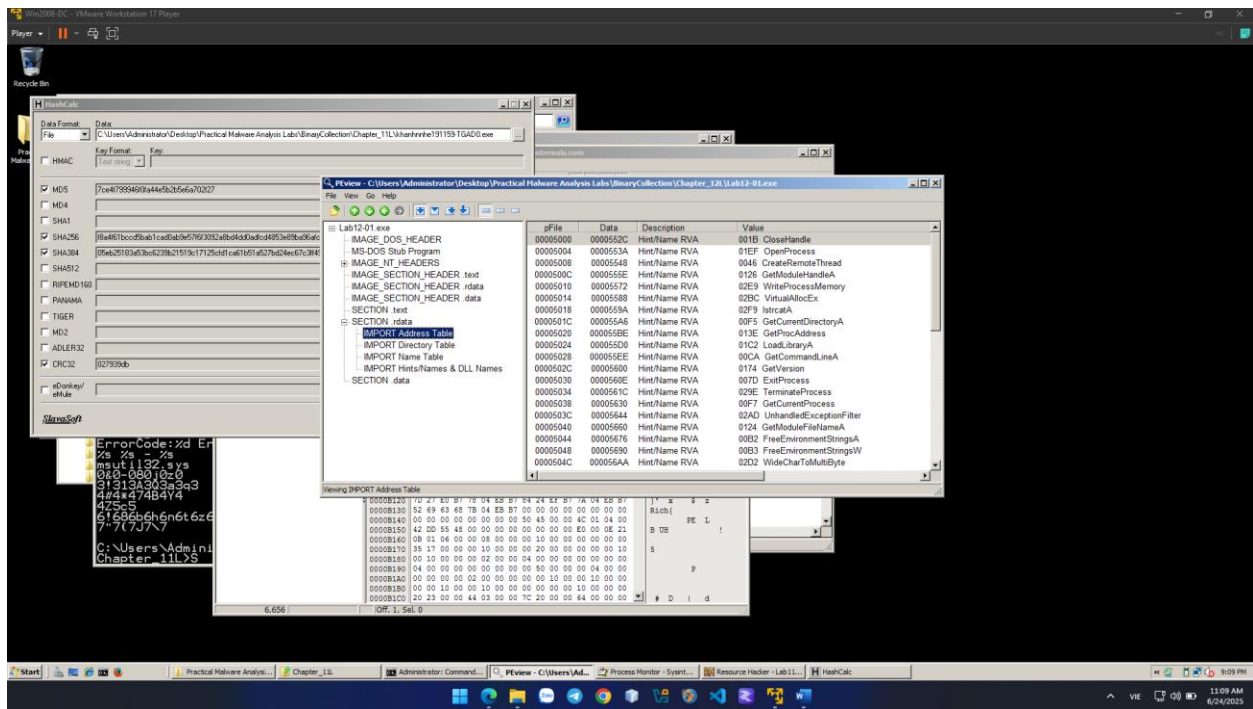
What you need: The Windows 2008 Server virtual machine we have been using.

Purpose: Analyze malware behavior

Imports

Examine Lab12-01.exe in PEView. Find these three imports, which are used in process injection:

- CreateRemoteThread
- WriteProcessMemory
- VirtualAllocEx



	pFile	Data	Description	Value
...IMAGE_DOS_HEADER	00005000	0000552C	Hint/Name RVA	001B CloseHandle
...MS-DOS Stub Program	00005004	0000553A	Hint/Name RVA	01EF OpenProcess
...IMAGE_NT_HEADERS	00005008	00005548	Hint/Name RVA	0046 CreateRemoteThread
...IMAGE_SECTION_HEADER .text	0000500C	0000555E	Hint/Name RVA	0126 GetModuleHandleA
...IMAGE_SECTION_HEADER .rdata	00005010	00005572	Hint/Name RVA	02E9 WriteProcessMemory
...IMAGE_SECTION_HEADER .data	00005014	00005588	Hint/Name RVA	02BC VirtualAllocEx
...SECTION .text	00005018	0000559A	Hint/Name RVA	02F9 IstrcatA

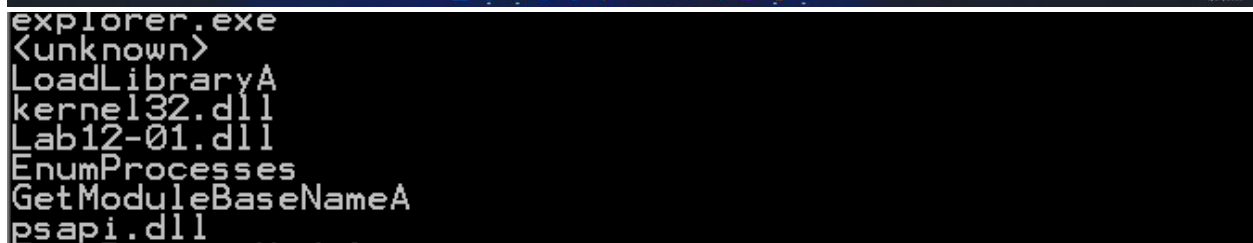
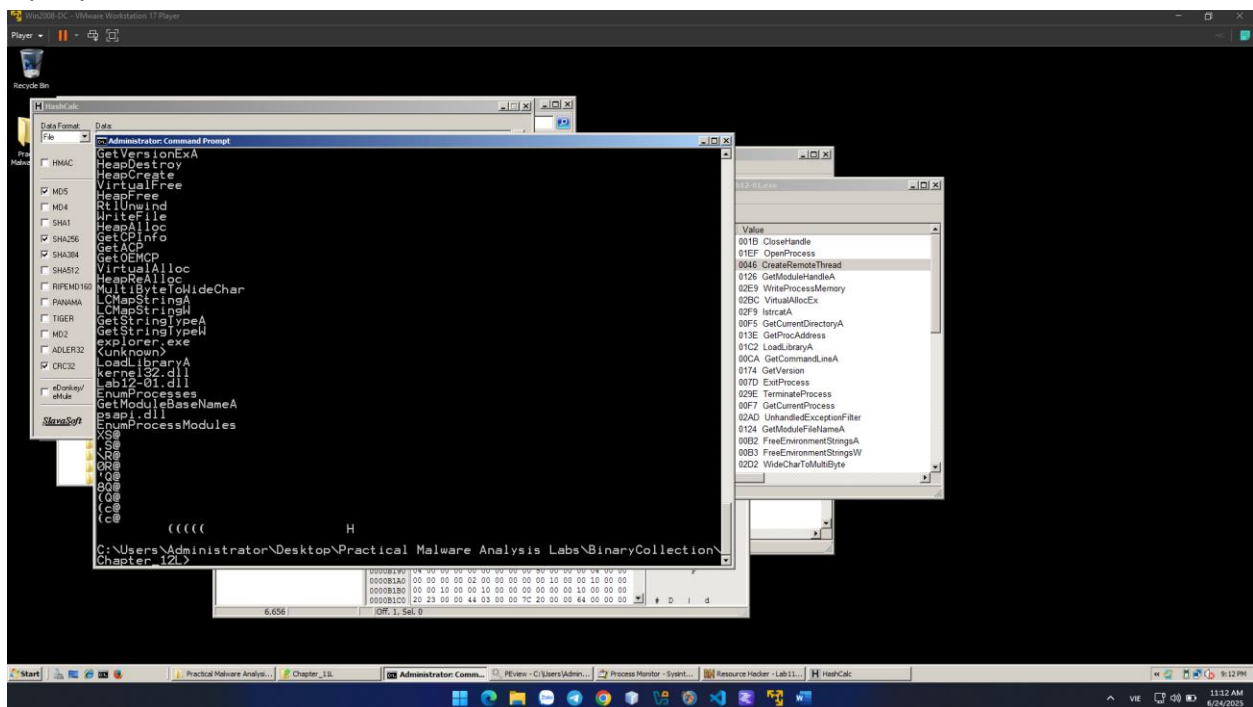
## Strings

Examine the strings in Lab12-01.exe. Find these three strings, which show the process being injected, the DLL file used, and psapi.dll, which is used for process enumeration:

- explorer.exe
- Lab12-01.dll



- psapi.dll



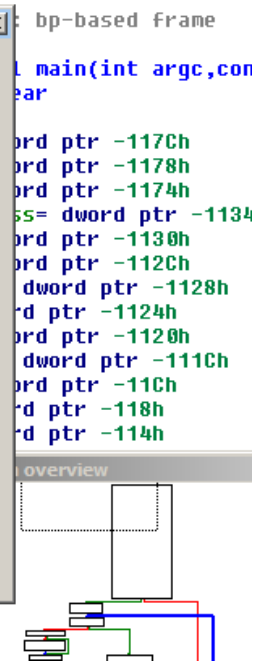
IDA Pro

Load Lab12-01.exe in IDA Pro Free.

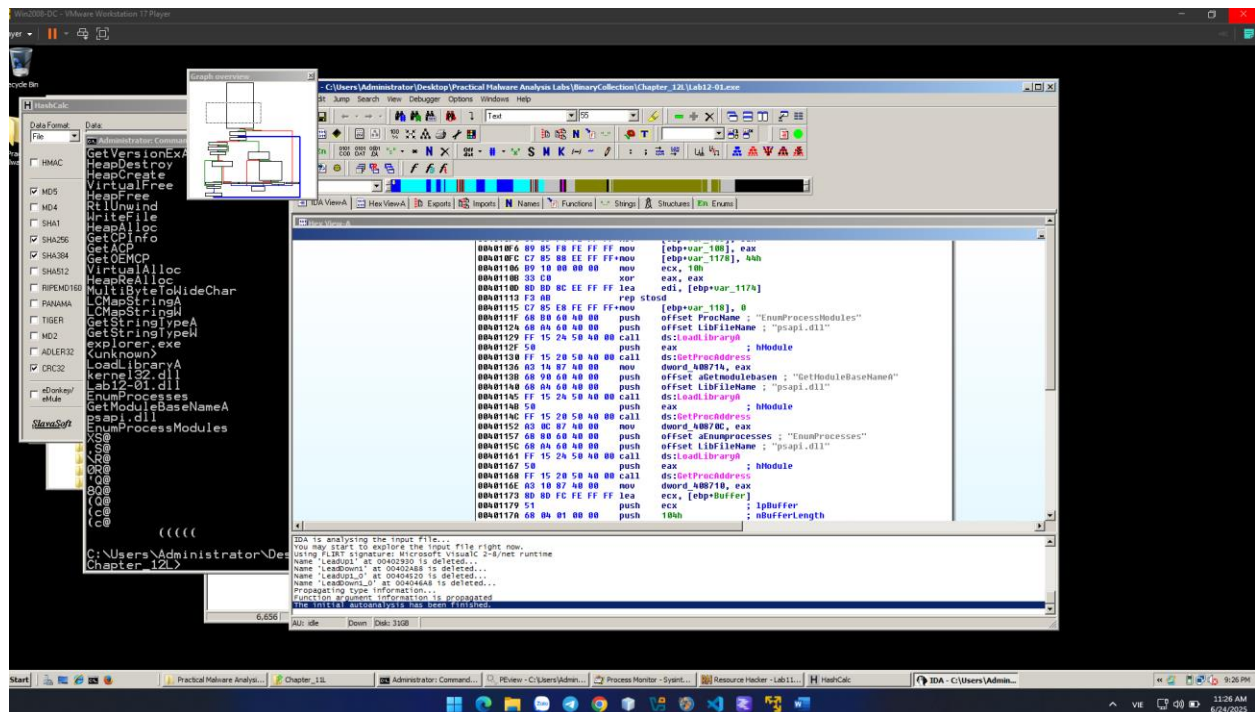
Click Options, General.

Check "Line Prefixes" and set the "Number of opcode bytes" to 6, as shown below.





B8	7C	11	00	00	mov
E8	73	02	00	00	call

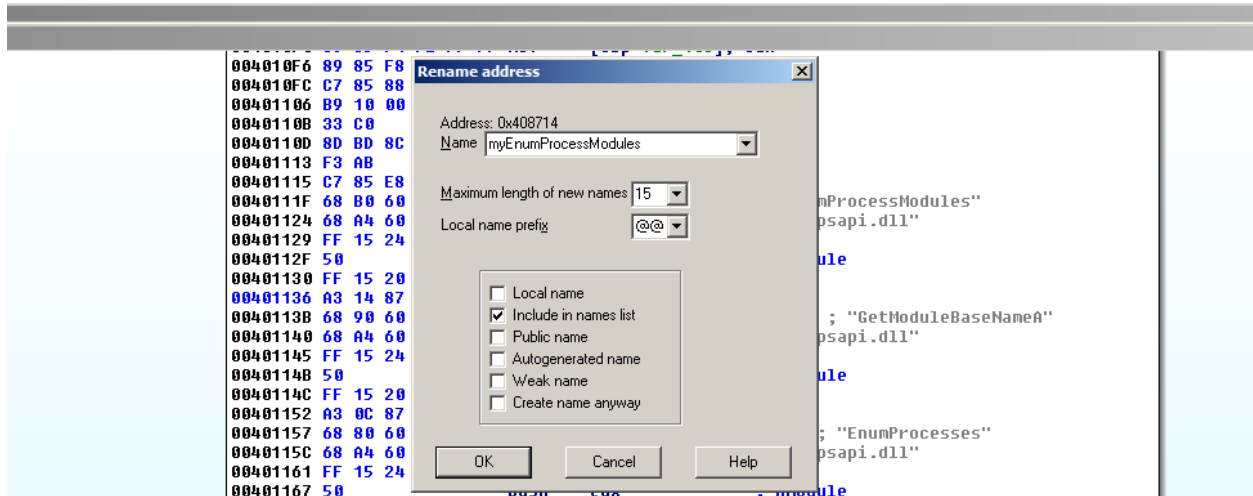


This code uses psapi three times to locate a Windows API function and store its address in a numerical address.

This obfuscates the code, so later calls to these functions will be difficult to recognize. We'll assign labels to these memory addresses in IDA Pro to make later analysis easier.

The first section of code assigns a pointer to the function EnumProcessModules.

In the line starting with address 00401136, right-click dword\_408714 and click Rename. Enter a new Name of myEnumProcessModules in the box, as shown below. Click OK. Increase the length limit when you are prompted to.

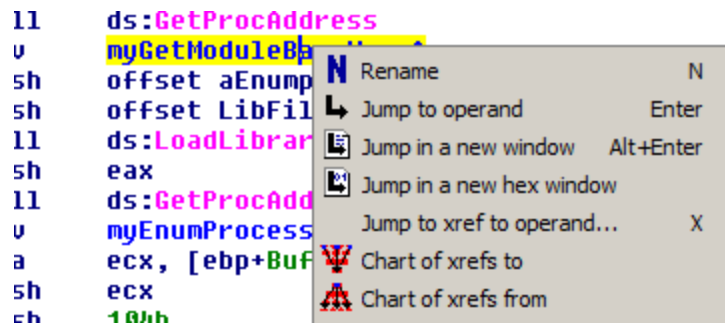


Repeat the process to rename dword\_40870C to myGetModuleBaseNameA

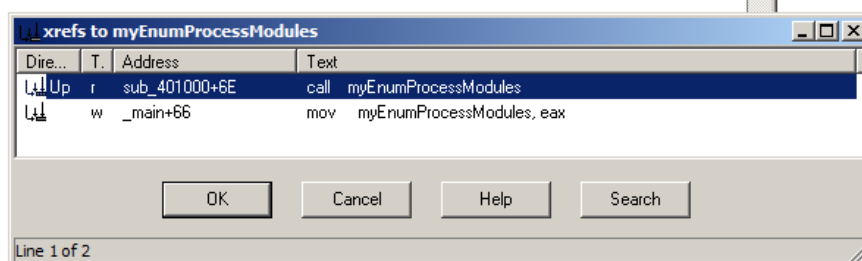
Repeat the process to rename dword\_408710 to myEnumProcesses



Right-click myGetModuleBaseNameA and click "Jump tp xrefs of operand", as shown below:



An xrefs box pops up, as shown below, showing that this address is only used once, in sub\_401000.

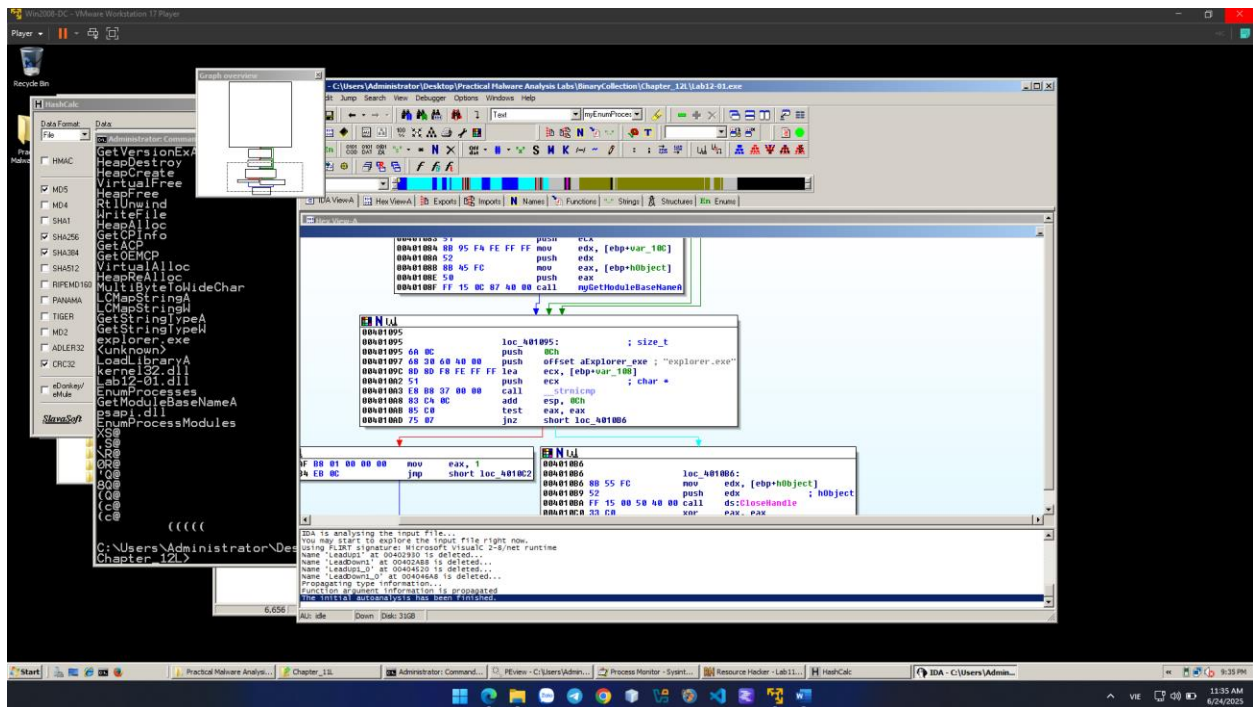


In the xrefs box, click OK.

This routine enumerates the modules and compares each module name to "explorer.exe", to find the module into which to inject code.

Make sure you can see these three items on your screen, as shown below:

- call myGetModuleBaseNameA
- "explorer.exe"
- call \_\_strnicmp



```

0040108B 8B 45 FC      mov     ecx, [ebp+hObject]
0040108E 50             push    ecx
0040108F FF 15 0C 87 40 00 call    myGetModuleBaseNameA

```

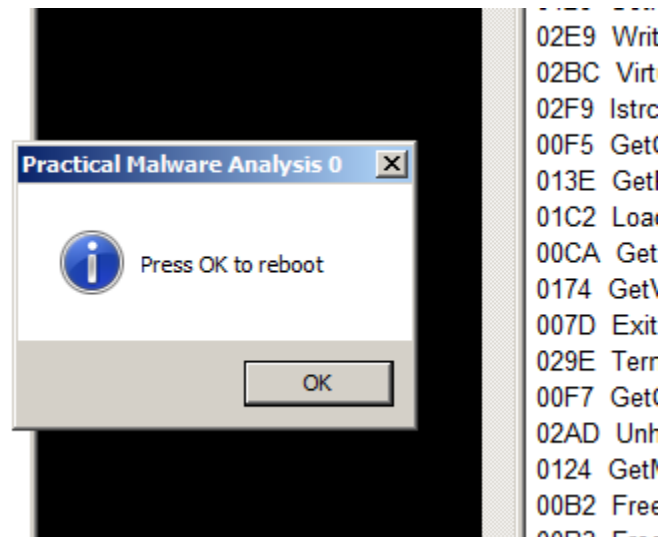
```

Nt!
00401095
00401095      loc_401095:                ; size_t
00401095 6A 0C      push    0Ch
00401097 68 30 60 40 00 push    offset aExplorer_exe ; "explorer.exe"
0040109C 8D 8D F8 FE FF FF lea     ecx, [ebp+var_108]
004010A2 51      push    ecx                ; char *
004010A3 E8 B8 37 00 00 call    __strnicmp
004010A8 83 C4 0C      add     esp, 0Ch

```

Process Explorer

Close IDA Pro. Double-click Lab12-01.exe to run the malware. A box pops up saying "Press OK to reboot". as shown below. Drag this box out of the way.



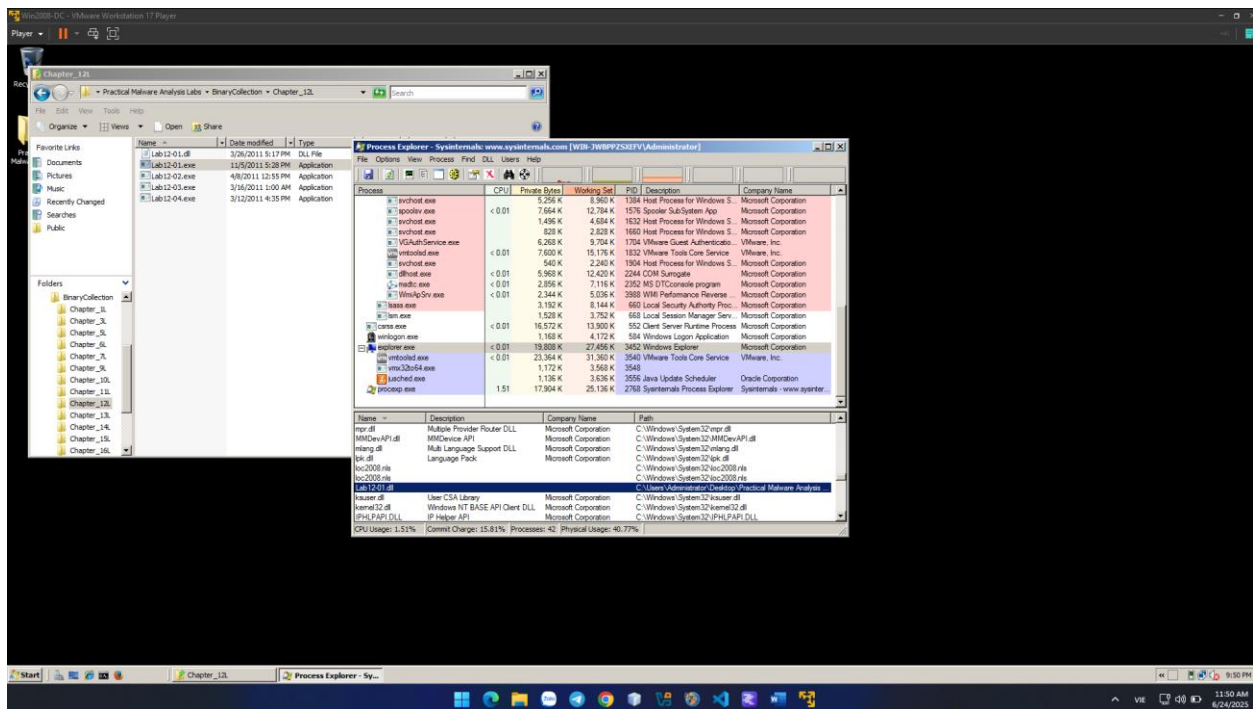
Open Process Explorer.

In the upper pane, scroll to the bottom of the list. Click explorer.exe to select it.

In Process Explorer, from the menu bar, click View and make sure "Show Lower Pane" is checked.

In Process Explorer, from the menu bar, click View, "Lower Pane View", DLLs.

In the lower pane, find the Lab12-01.dll that has been injected into explorer.exe, as shown below.



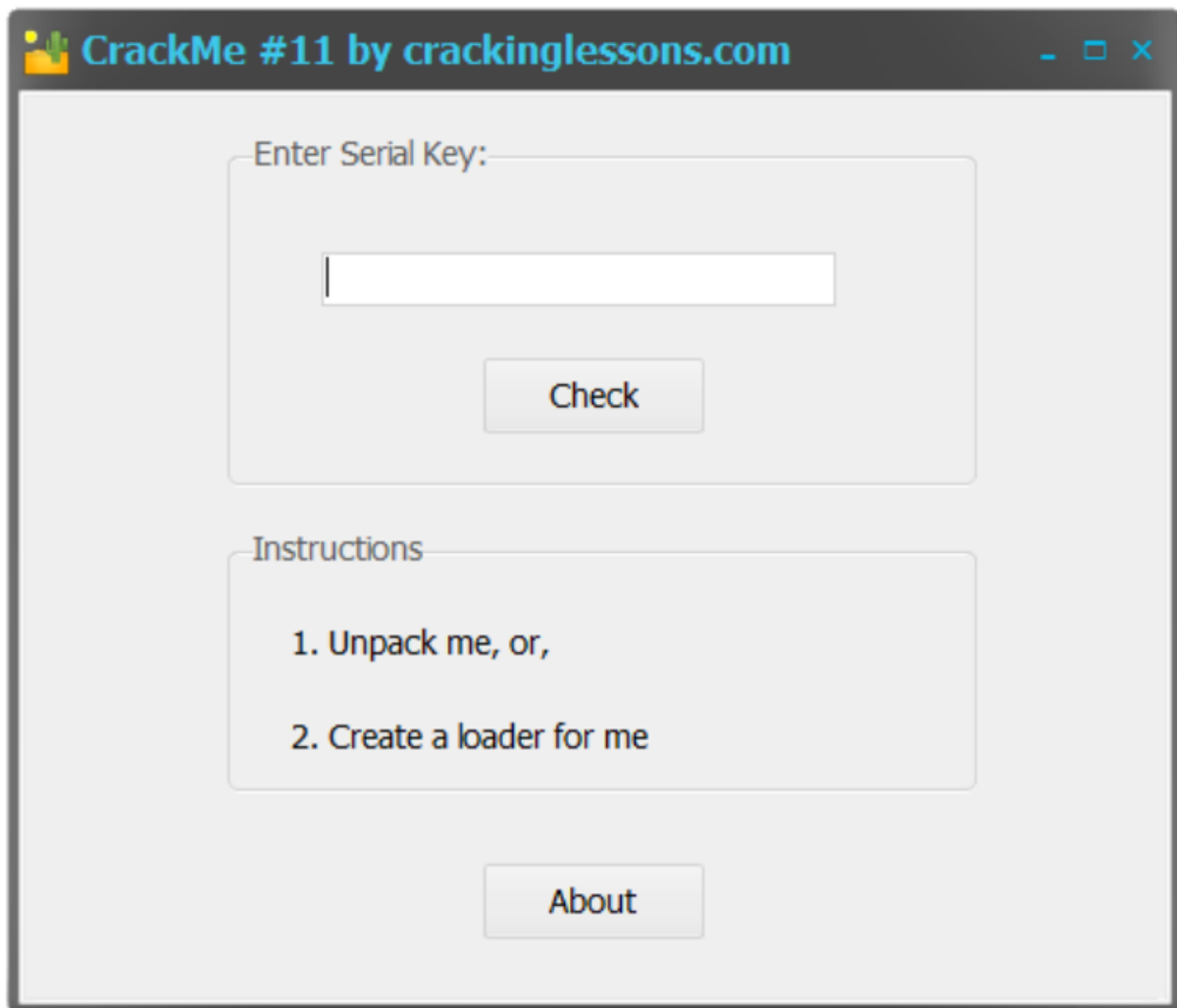
explorer.exe	< 0.01	19,808 K	27,456 K	3452	Windows Explorer	Microsoft Corporation
vmtoolsd.exe	< 0.01	23,364 K	31,360 K	3540	VMware Tools Core Service	VMware, Inc.
vmx32to64.exe		1,172 K	3,568 K	3548		
usched.exe		1,136 K	3,636 K	3556	Java Update Scheduler	Oracle Corporation
procexp.exe	1.54	17,900 K	25,132 K	2768	Sysinternals Process Explorer	Sysinternals - www.sysinter...

Name	Description	Company Name	Path
mpr.dll	Multiple Provider Router DLL	Microsoft Corporation	C:\Windows\System32\mpr.dll
MMDevAPI.dll	MMDevice API	Microsoft Corporation	C:\Windows\System32\MMDevAPI.dll
mlang.dll	Multi Language Support DLL	Microsoft Corporation	C:\Windows\System32\mlang.dll
lpk.dll	Language Pack	Microsoft Corporation	C:\Windows\System32\lpk.dll
loc2008.nls			C:\Windows\System32\loc2008.nls
loc2008.nls			C:\Windows\System32\loc2008.nls
Lab 12-01.dll			C:\Users\Administrator\Desktop\Practical Malware Analysis ...
ksuser.dll	User CSA Library	Microsoft Corporation	C:\Windows\System32\ksuser.dll
kemsel32.dll	Windows NT BASE API Client DLL	Microsoft Corporation	C:\Windows\System32\kernbase.dll
IPHLPAPI.DLL	IP Helper API	Microsoft Corporation	C:\Windows\System32\IPHLPAPI.DLL
CPU Usage: 1.54%   Commit Charge: 15.79%   Processes: 42   Physical Usage: 40.73%			

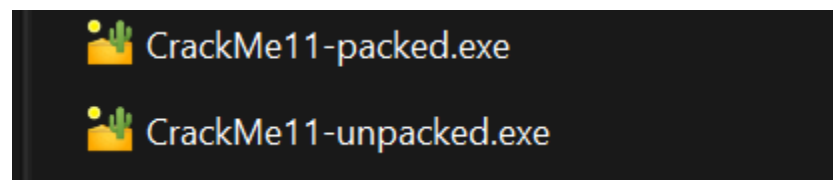
# CRACKME 11:

This CrackMe is packed with UPX 3.91 packer. Your task is to :

1. Unpack it and then patch the unpacked file, or,
2. Create a loader for it



In this challenge, I dont focus on how to crack file, I want try to unpacked the file one successfully.





The second file can be cracked and patched successfully. But the first one is packed by UPX 3.91 packer.

For CrackMe11-unpacked.exe

I find string references to view the code related (Sorry wrong key)

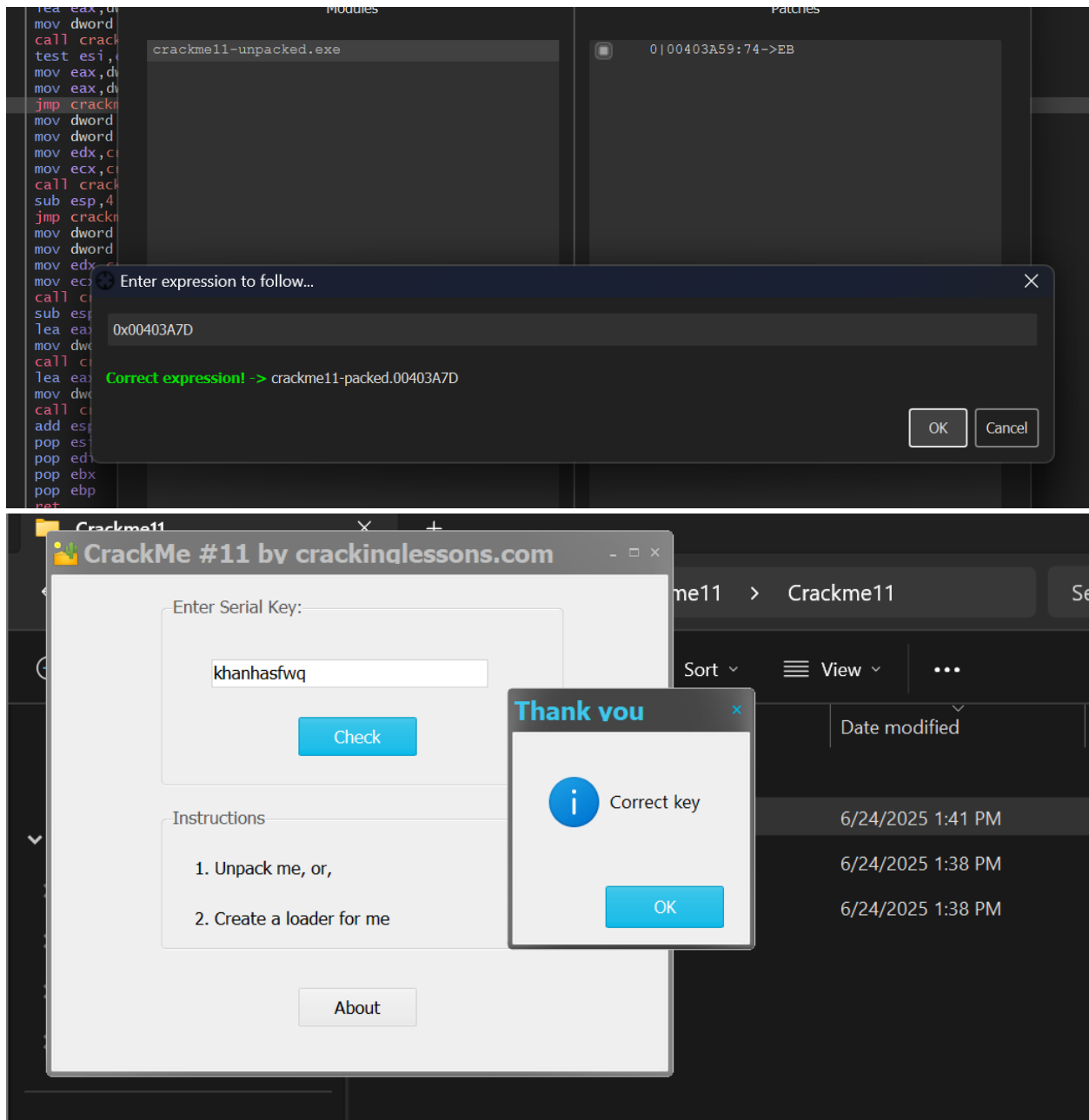
User Modules (Strings)			
Address	Disassembly	String Address	String
004003F6	xor eax,20000	00020000	"??"
00403171	mov dword ptr ss:[esp+4],crackme11-unpacked.718154	00718154	"Turquoise Gray"
004031C0	mov dword ptr ss:[esp+4],crackme11-unpacked.718163	00718163	"CrackMe #8"
004031FE	mov ecx,dword ptr ds:[7255EC]	007255EC	"ar"
004039B5	mov edx,crackme11-unpacked.7181E0	007181E0	L"coded by crackinglessons.com"
004039BA	mov ecx,crackme11-unpacked.71821A	0071821A	L>About"
00403A1D	mov dword ptr ss:[esp+4],crackme11-unpacked.718226	00718226	"ABC-123456"
00403A69	mov edx,crackme11-unpacked.71825E	0071825E	L"Sorry wrong key"
00403A6E	mov ecx,crackme11-unpacked.71827E	0071827E	L"Error"
00403A8B	mov edx,crackme11-unpacked.718232	00718232	L"Correct key"
00403A90	mov ecx,crackme11-unpacked.71824A	0071824A	L"Thank you"
00403F0B	push 17261	00017261	"??"
004041B9	imul ebp,dword ptr ds:[esi+67],20000	00020000	"??"
00404462	imul esp,dword ptr ss:[ebp+73],20002	00020002	"??"
004052AA	add bh,byte ptr ds:[edx+edx*2+20040]	00020040	"??"
0040542E	imul esi,dword ptr ds:[esi+67],crackme11-unpacked.403A7D	00405528	"ug"
00405A2E	imul ebp,dword ptr ds:[edi+ebp*2+72],crackme11-unpacked.403A7D	004041AC	&"\nAnsiString"
004066B7	imul edi,dword ptr ds:[edx+65],20002	00020002	"??"
00406702	imul edi,dword ptr ds:[edx+65],20002	00020002	"??"
0040734D	add byte ptr ds:[edi+edi+20040],b1	00020040	"??"
00407418	add byte ptr ds:[edi+edi+20040],b1	00020040	"??"
00408086	mov word ptr ds:[eax+20040],es	00020040	"??"
00408090	mov edi,10+60	00010160	"??"

Look at the JE command can jump to Correct status.

85F0	test esi,esi		
A1 78567200	mov eax,dword ptr ds:[725678]		
8B00	mov eax,dword ptr ds:[eax]		
74 22	je crackme11-unpacked.403A7D		
C745 BC 04000000	mov dword ptr ss:[ebp-44],4		[ebp-44]:LdrInitShimEngineDynamic+6A9
C70424 10000000	mov dword ptr ss:[esp],10		
BA 5E827100	mov edx,crackme11-unpacked.71825E		71825E:L"Sorry wrong key"
B9 7E827100	mov ecx,crackme11-unpacked.71827E		71827E:L"Error"
E8 012B3100	call crackme11-unpacked.716579		
83EC 04	sub esp,4		
EB 20	jmp crackme11-unpacked.403A9D		
C745 BC 03000000	mov dword ptr ss:[ebp-44],3		[ebp-44]:LdrInitShimEngineDynamic+6A9
C70424 40000000	mov dword ptr ss:[esp],40		40:'@'
BA 32827100	mov edx,crackme11-unpacked.718232		718232:L"Correct key"
B9 4A827100	mov ecx,crackme11-unpacked.71824A		71824A:L"Thank you"
E8 DF2A3100	call crackme11-unpacked.716579		
83EC 04	sub esp,4		
8045 5C	lea ecx,dword ptr ss:[ebp-14]		

Change to JMP command to always jump to Correct status. And pack it!

A1 78567200	mov eax,dword ptr ds:[725678]	
8B00	mov eax,dword ptr ds:[eax]	
EB 22	jmp crackme11-unpacked.403A7D	
C745 BC 04000000	mov dword ptr ss:[ebp-44],4	
C70424 10000000	mov dword ptr ss:[esp],10	
BA 5E827100	mov edx,crackme11-unpacked.71825E	
B9 7E827100	mov ecx,crackme11-unpacked.71827E	
E8 012B3100	call crackme11-unpacked.716579	
83EC 04	sub esp,4	
EB 20	jmp crackme11-unpacked.403A9D	
C745 BC 03000000	mov dword ptr ss:[ebp-44],3	
C70424 40000000	mov dword ptr ss:[esp],40	
BA 32827100	mov edx,crackme11-unpacked.718232	



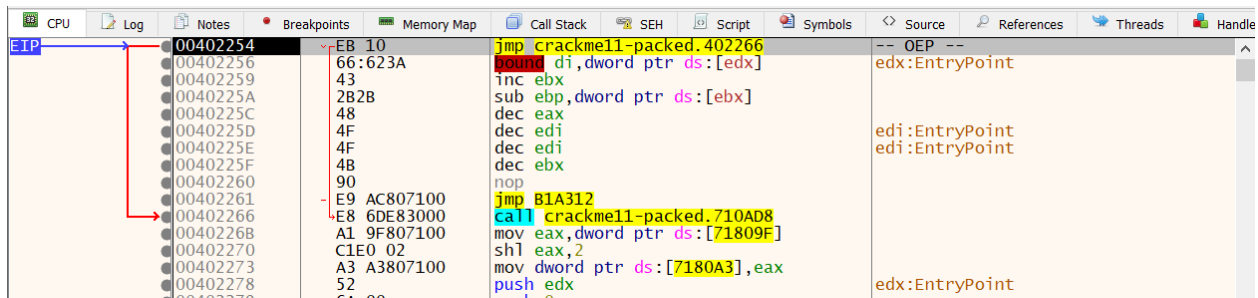
Success!!!

For CrackMe11-packed.exe – manually unpack the file packed by UPX:

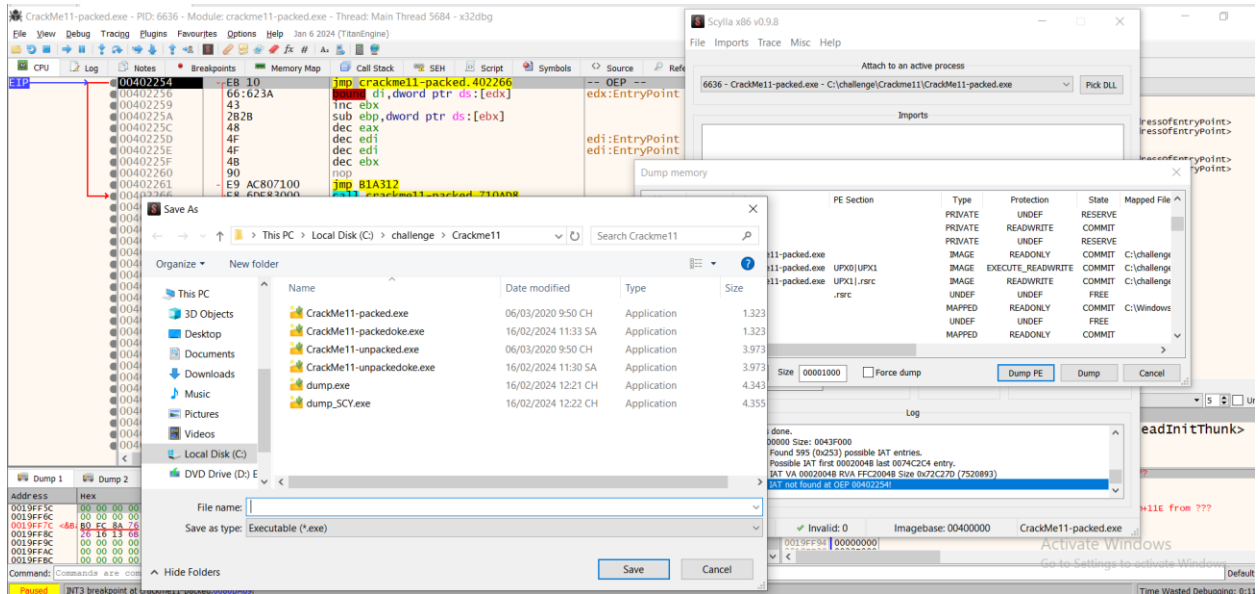
This file is packed so I don't see the Wrong serial key String. The codes are also hidden too.

So we need to unpack this file to do the same crack way as CrackMe11-unpacked.exe.

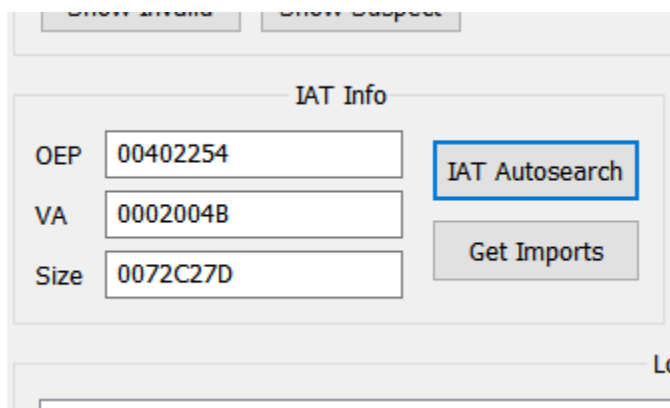




Use plugin to dump memory of file:



If only dump memory dll library is lacked and file doesnt run so we need to import by IAT info



Run file packed by UPX successfully.

