NGUYEN NAM KHANH – HE191159 – IA1902 – IAM302

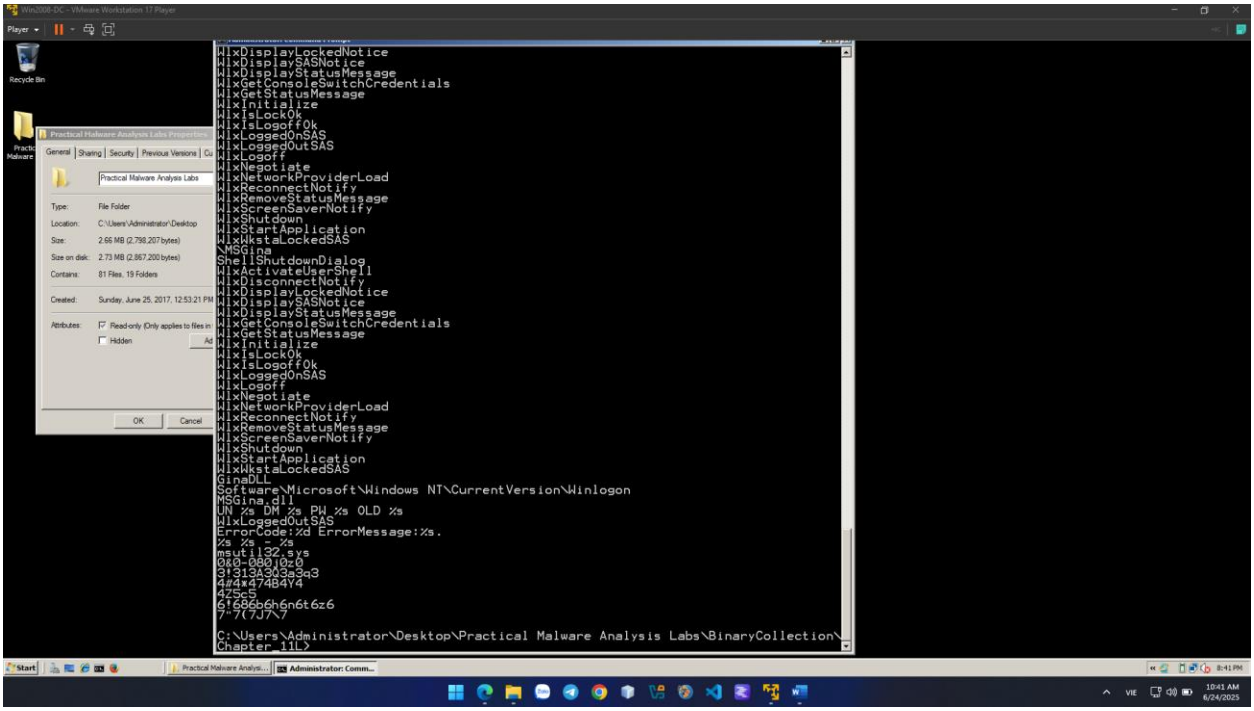
LAB 12 Dynamic Analysis Tools

LAB 1:

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

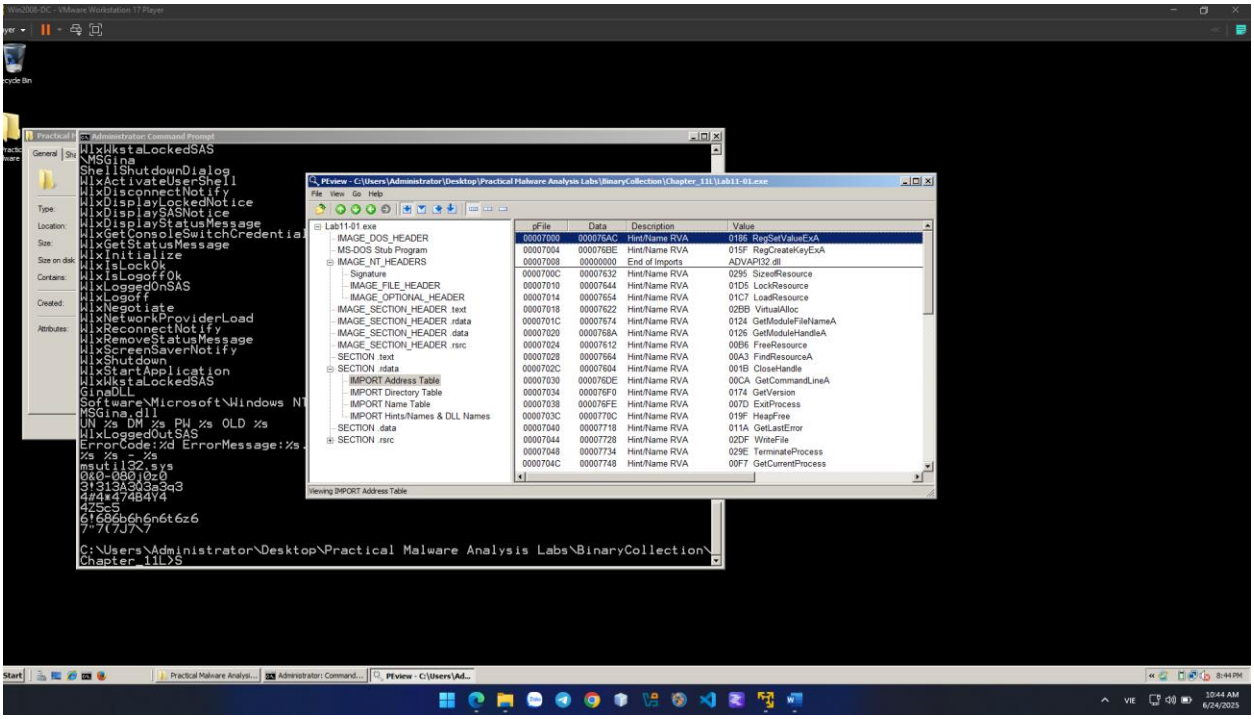
Static Analysis with Strings

Examine the strings in Lab11-01.exe.



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

• RegSetValueExA • RegCreateKeyExA • SizeofResource • LockResource • LoadResource

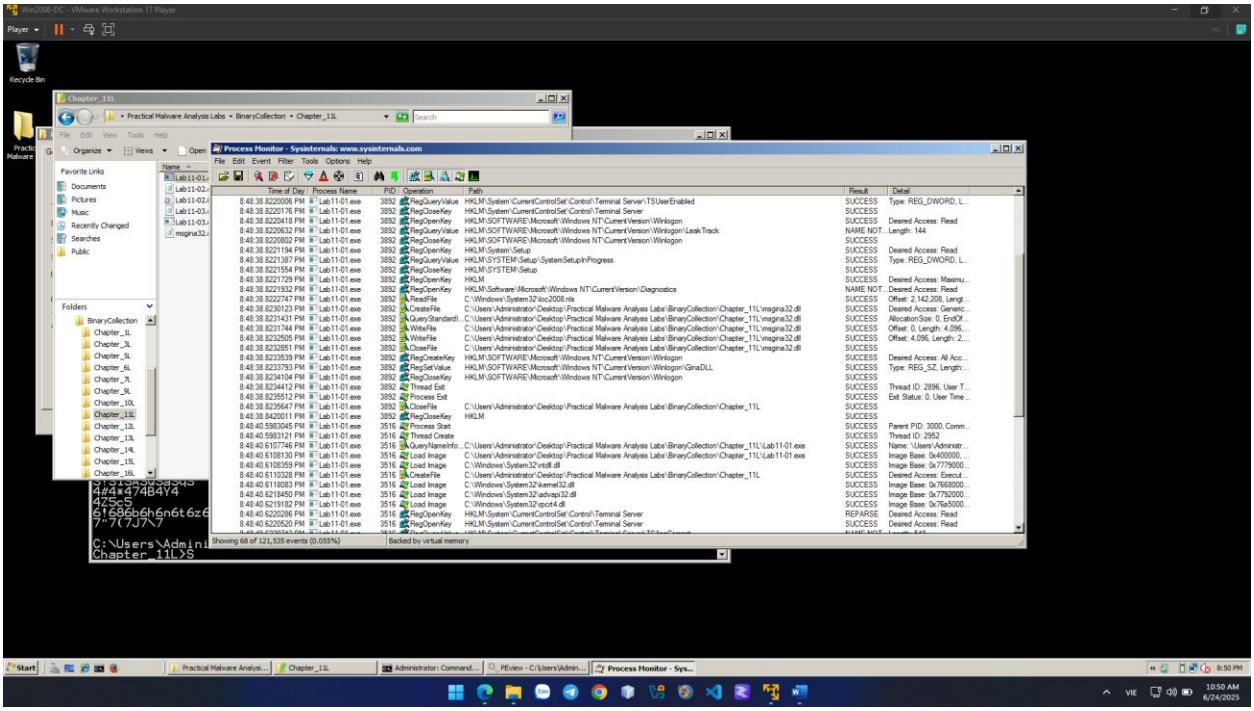


Dynamic Analysis with Procmon Run the malware in a virtual machine, while running Procmon to see what it does.

In Procmon, click Filter, "Reset Filter".

Click Filter, Filter. Filter for a "Process Name" of Lab11-01.exe.

- CreateFile ... msgina32.dll
- RegCreateKey HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
- RegSetValue HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL

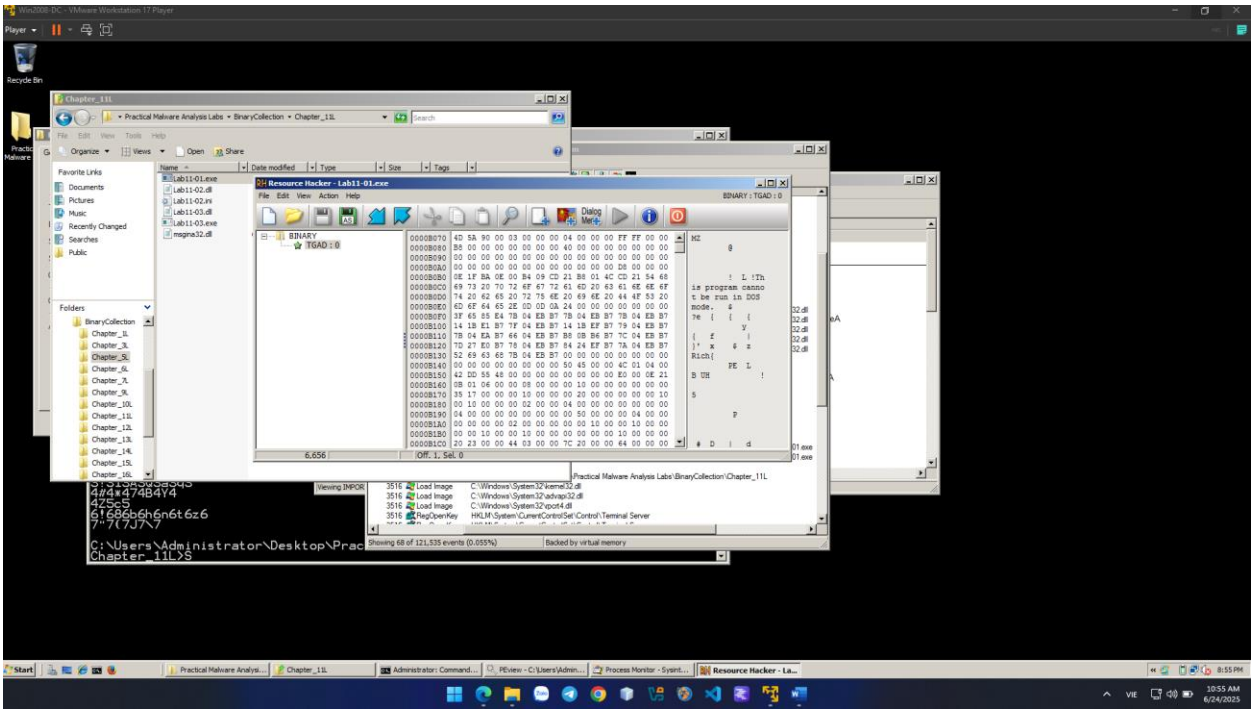


8:48:38.8221932 PM	Lab11-01.exe	3892	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics	NAME NOT
8:48:38.8222747 PM	Lab11-01.exe	3892	ReadFile	C:\Windows\System32\loc2008.nls	SUCCESS
8:48:38.8230123 PM	Lab11-01.exe	3892	CreateFile	C:\Users\Administrator\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_11L\msgina32.dll	SUCCESS
8:48:38.8231431 PM	Lab11-01.exe	3892	QueryStandardI...	C:\Users\Administrator\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_11L\msgina32.dll	SUCCESS
8:48:38.8231744 PM	Lab11-01.exe	3892	WriteFile	C:\Users\Administrator\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_11L\msgina32.dll	SUCCESS
8:48:38.8232505 PM	Lab11-01.exe	3892	WriteFile	C:\Users\Administrator\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_11L\msgina32.dll	SUCCESS
8:48:38.8232851 PM	Lab11-01.exe	3892	CloseFile	C:\Users\Administrator\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_11L\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

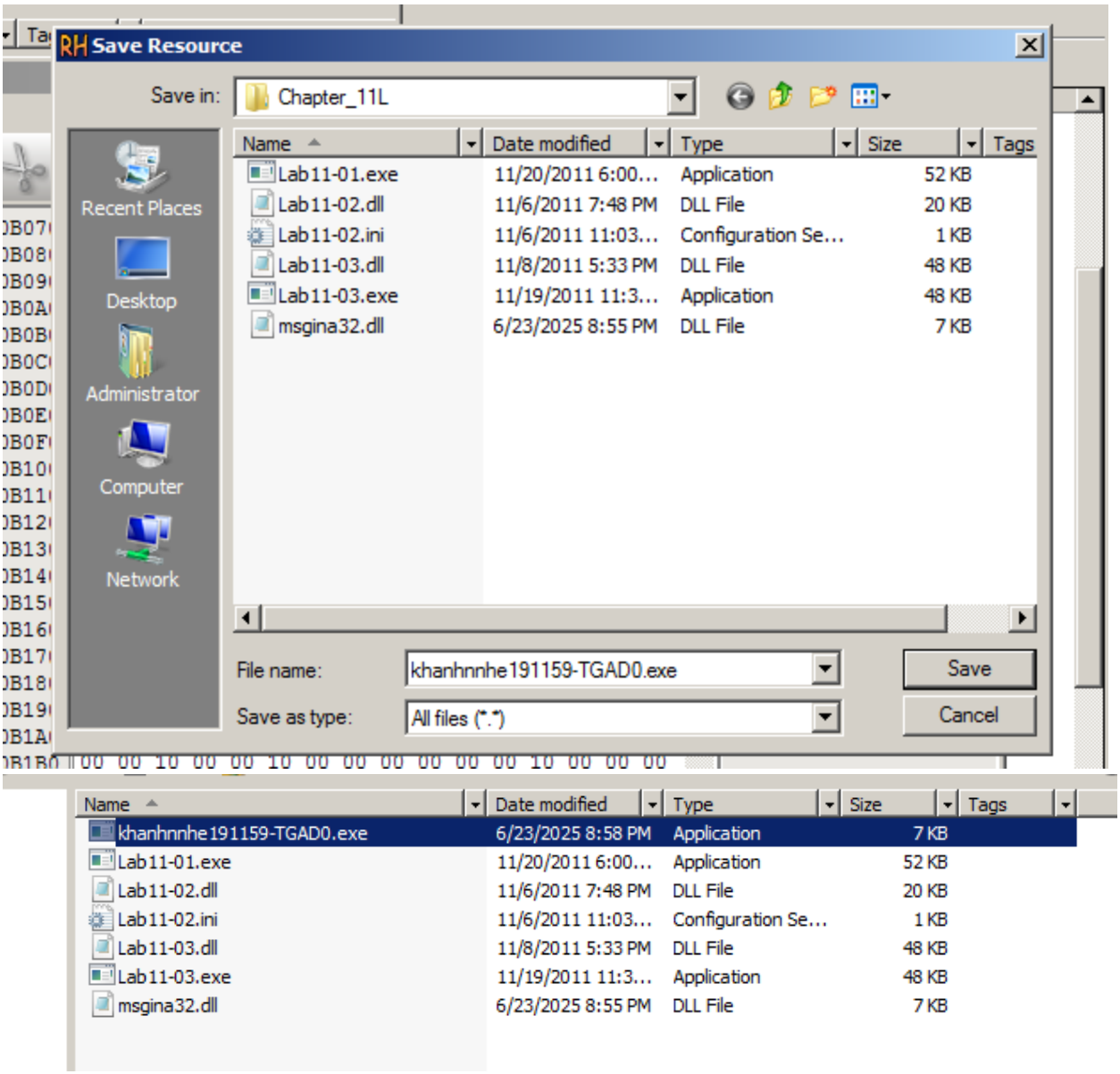
Resource Hacker

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.

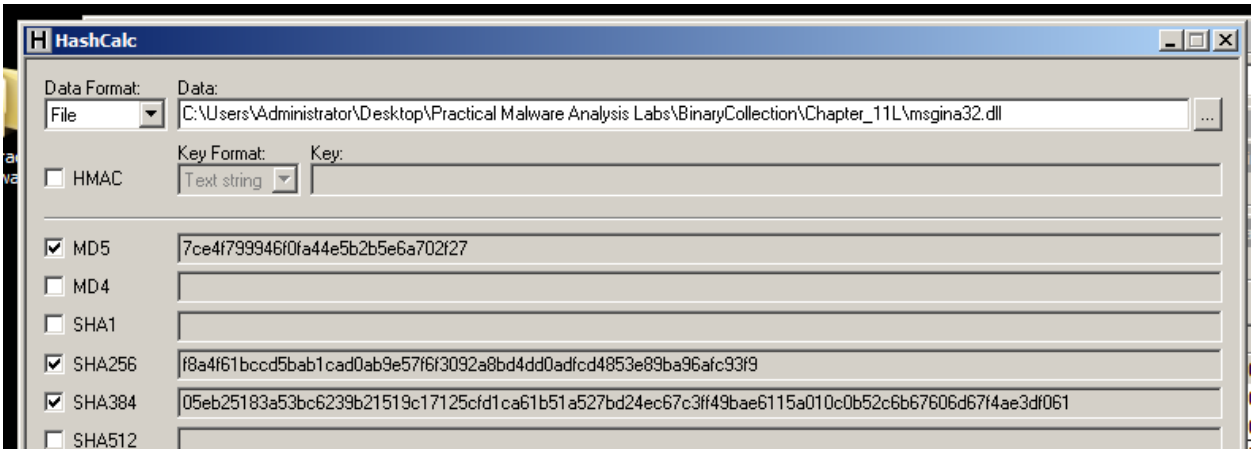


In Resource Hacker, in the left pane, click 0 ti highlight it, as shown above. Click Action, Save Resource as a binary file...". Save the file as YOURNAME-TGAD0.exe, replacing the text "YOURNAME" with your own name.

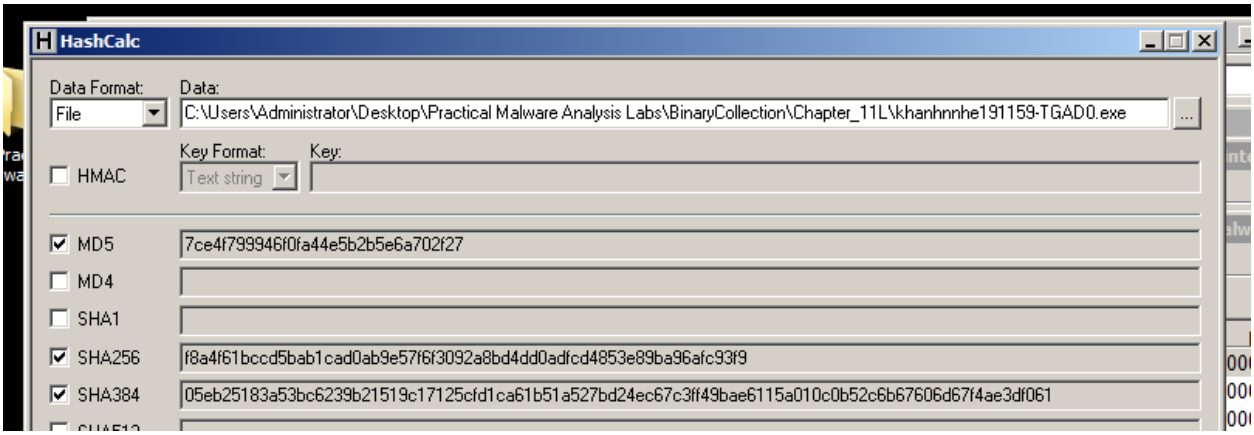


HashCalc

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:

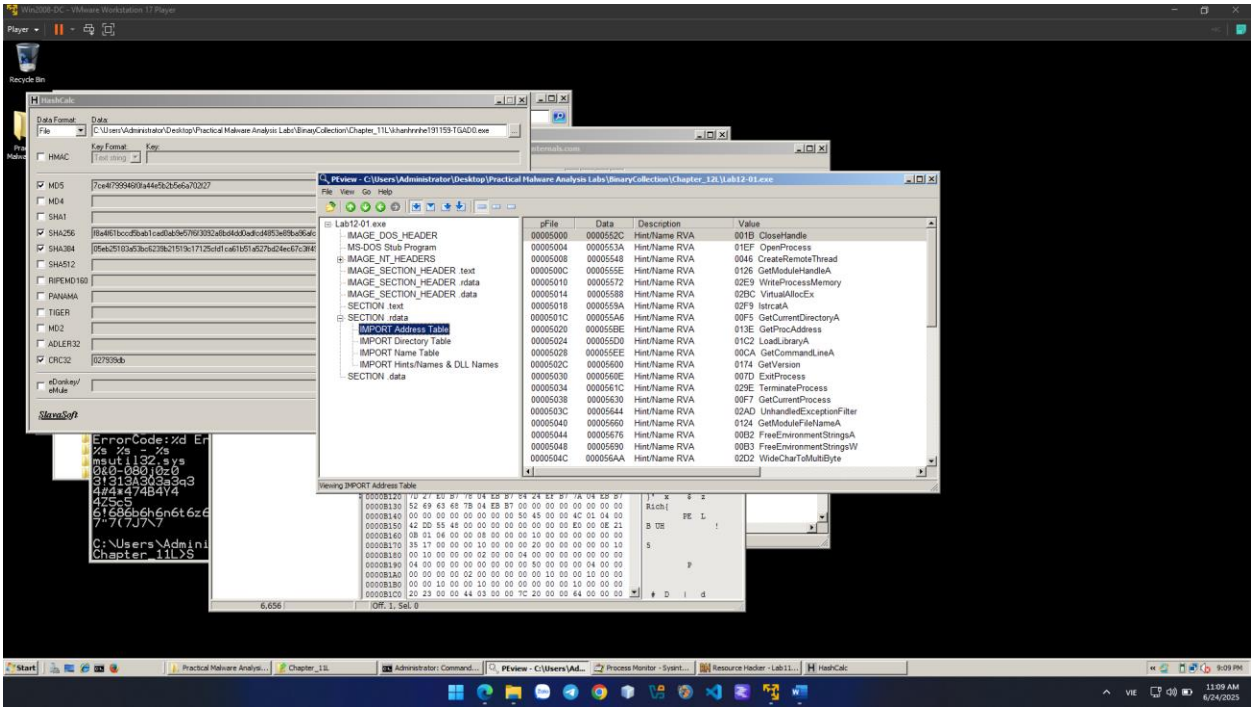
What you need: The Windows 2008 Server virtual machine we have been using.

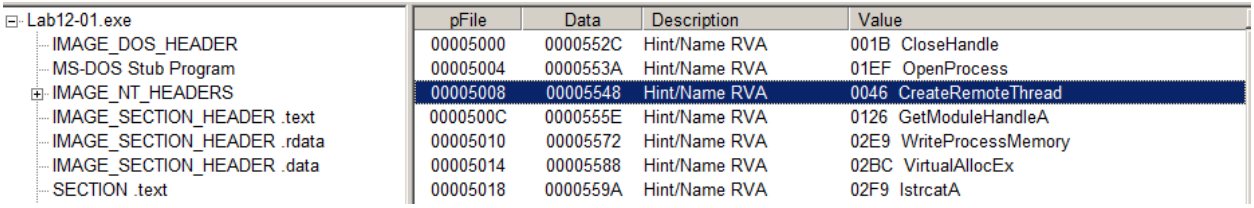
Purpose: Analyze malware behavior

Imports

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

- CreateRemoteThread
- WriteProcessMemory
- VirtualAllocEx



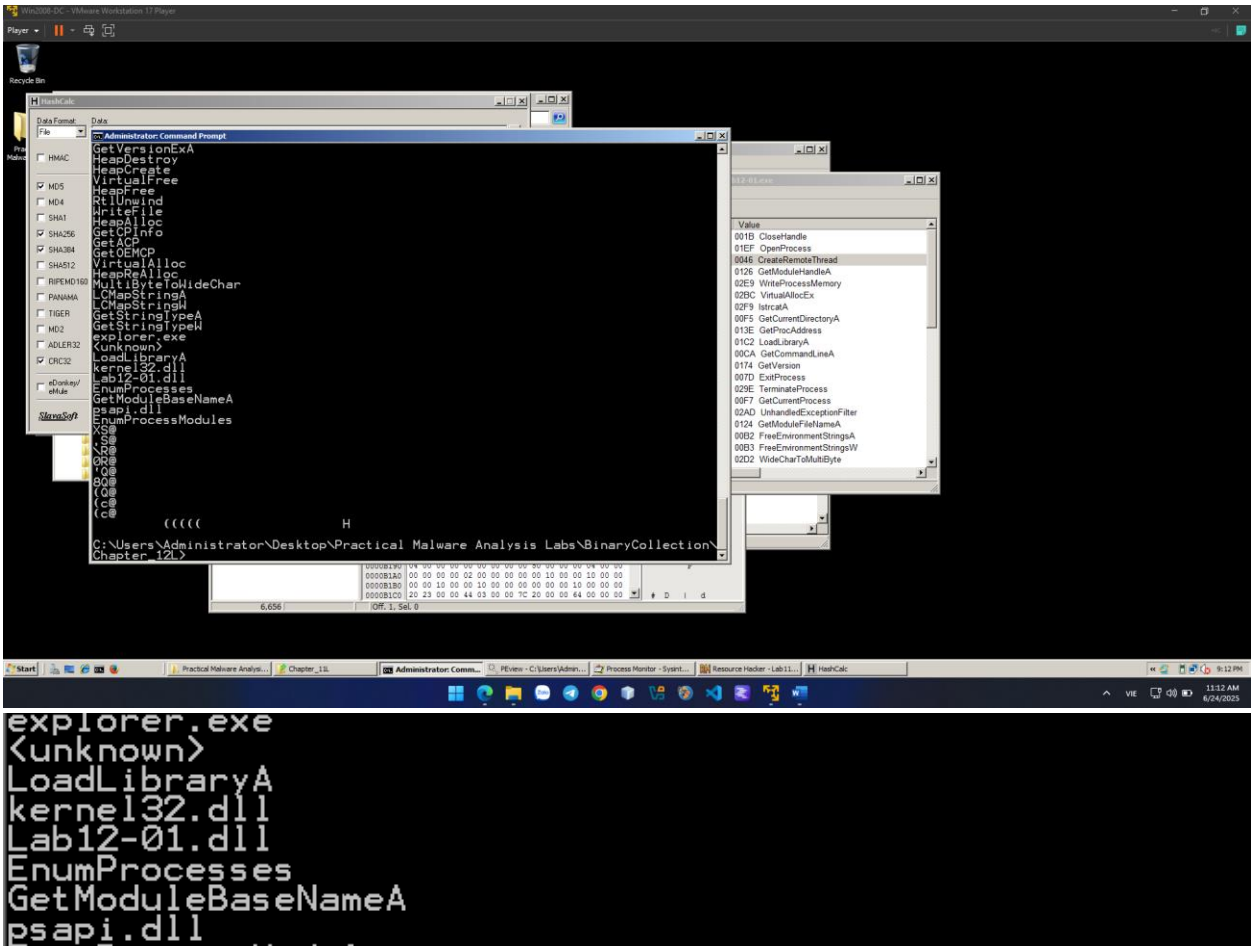


Strings

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

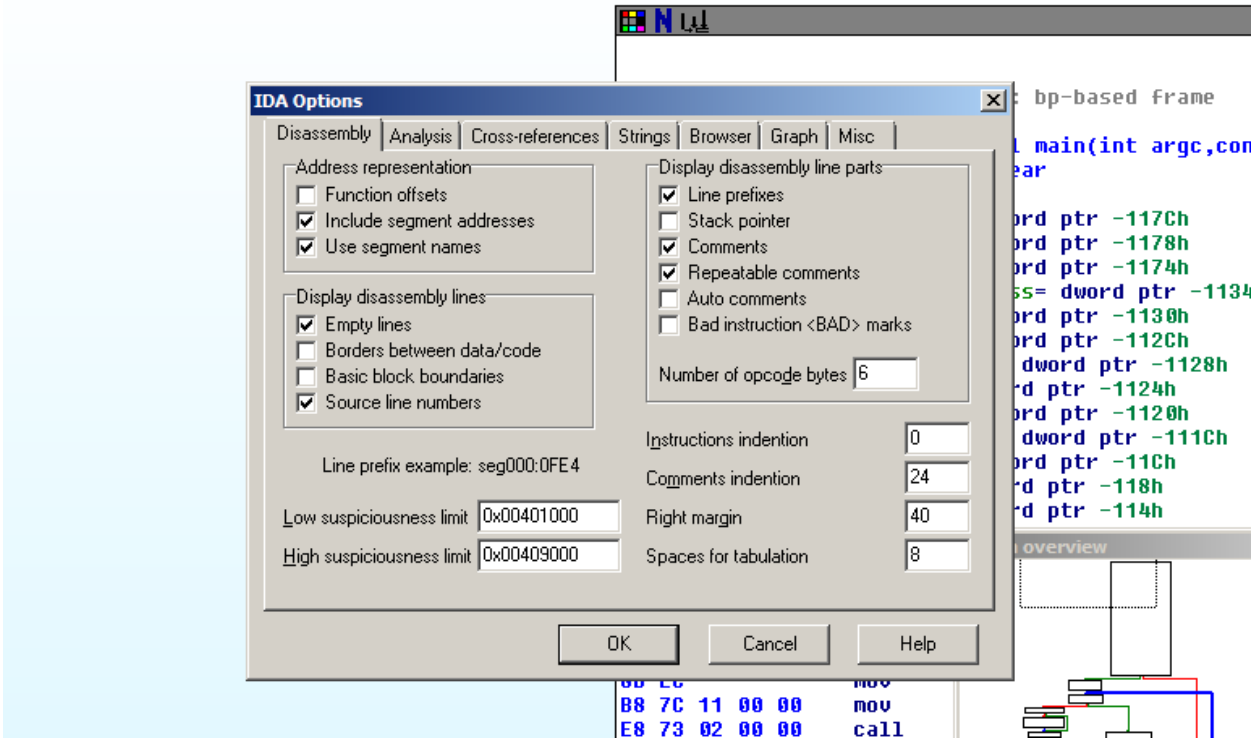


IDA Pro

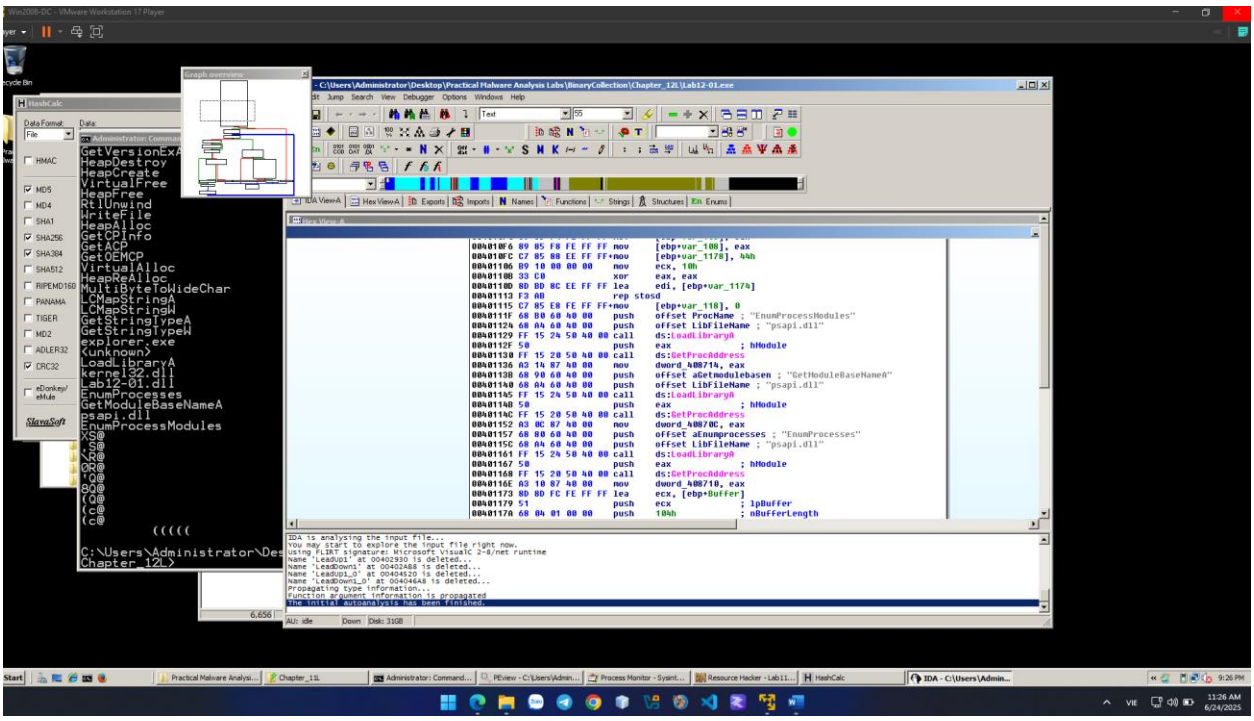
Load Lab12-01.exe in IDA Pro Free.

Click Options, General.

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



Find the code shown below, near the start of main():

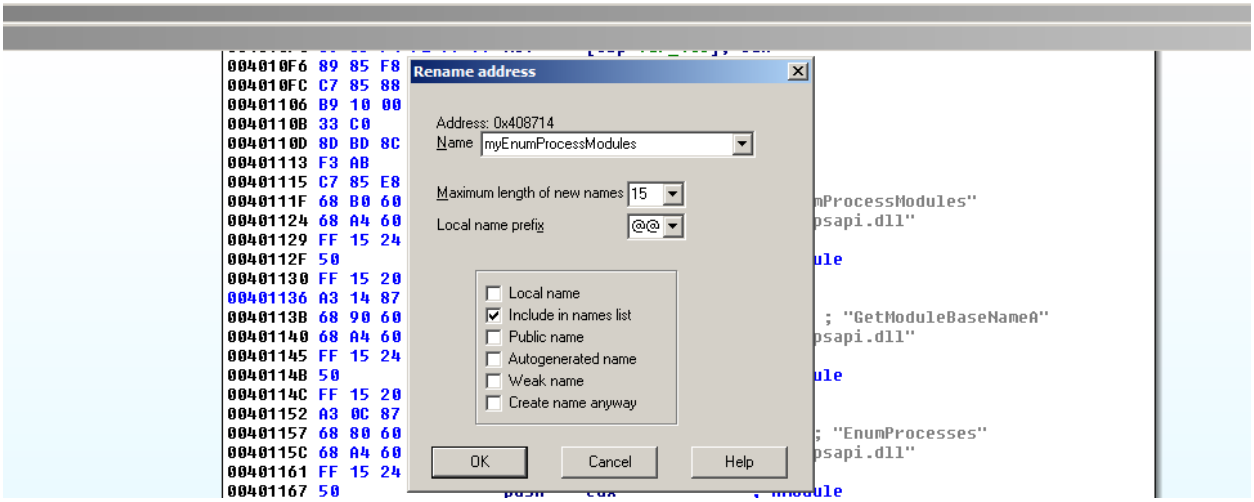


This code uses psapi three times to locate a Windows API function and store its address in a numerical address.

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.

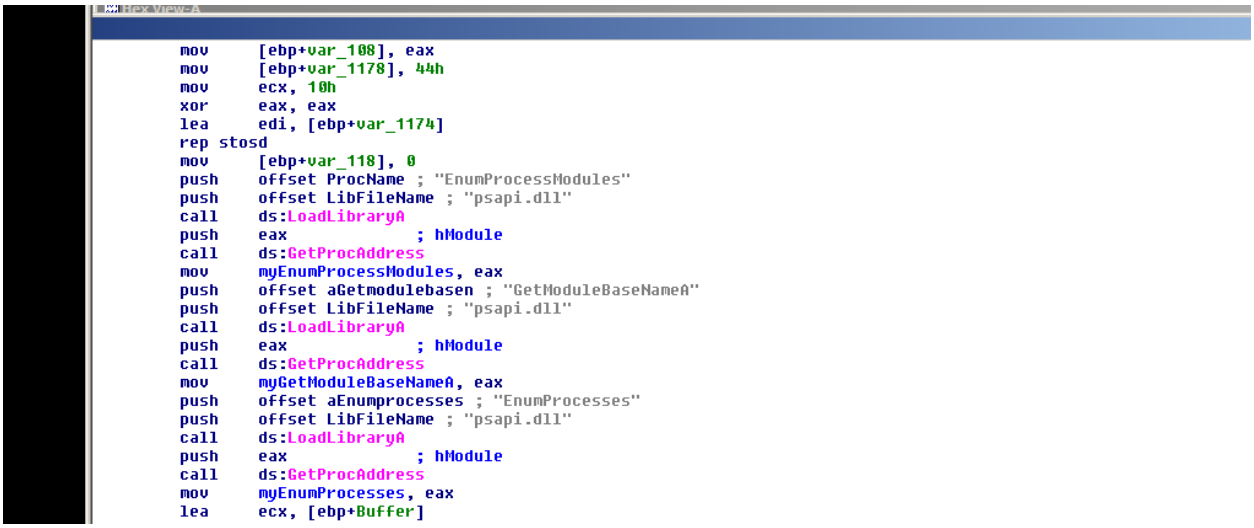
The first section of code assigns a pointer to the function EnumProcessModules.

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.

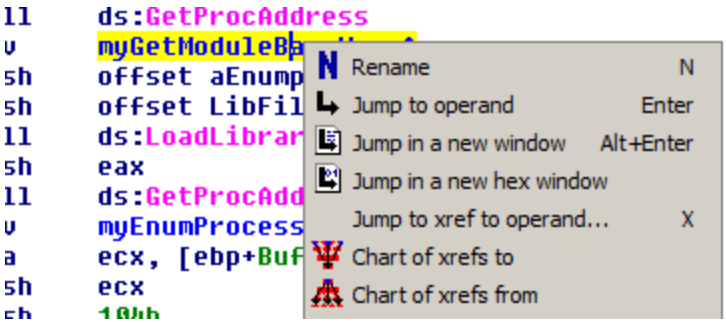


Repeat the process to rename dword_40870C to myGetModuleBaseNameA

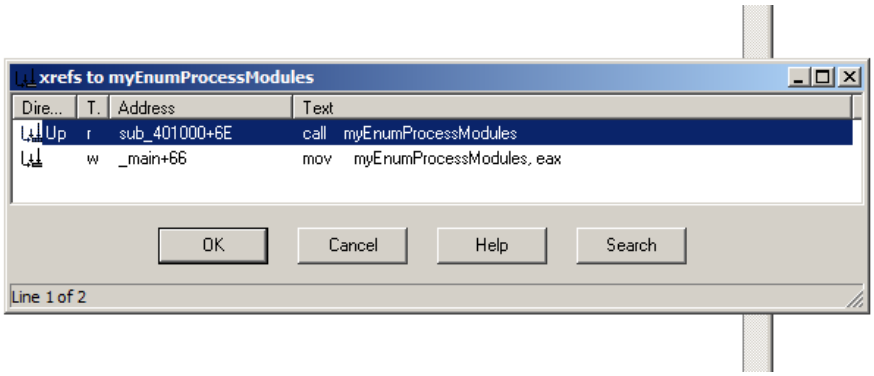
Repeat the process to rename dword_408710 to myEnumProcesses



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.

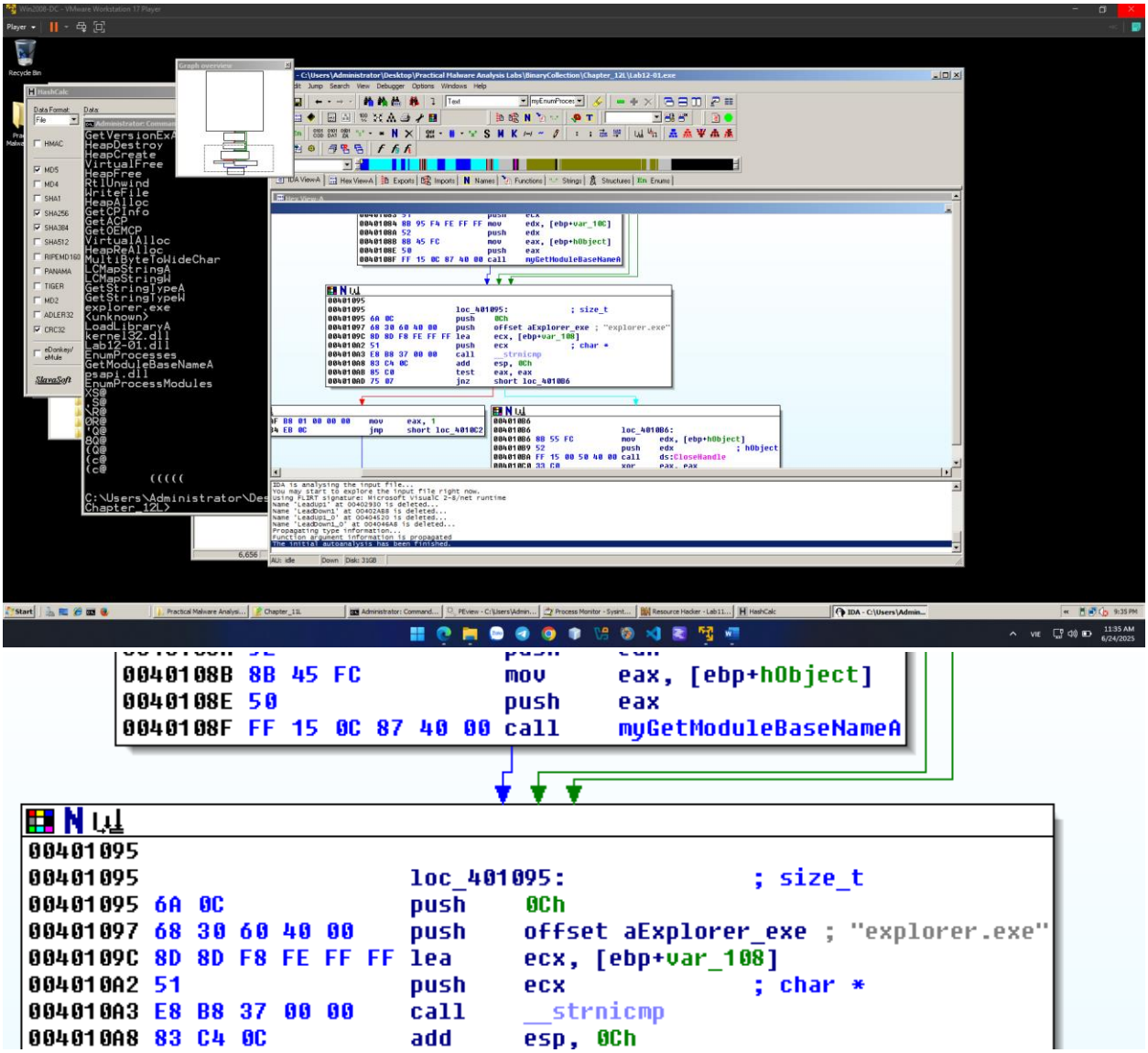


In the xrefs box, click OK.

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

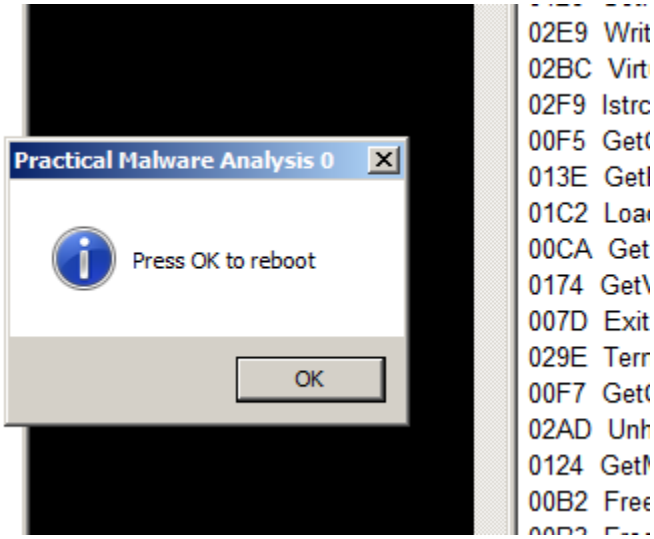
Make sure you can see these three items on your screen, as shown below:

- call myGetModuleBaseNameA
- "explorer.exe"
- call __strnicmp



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.



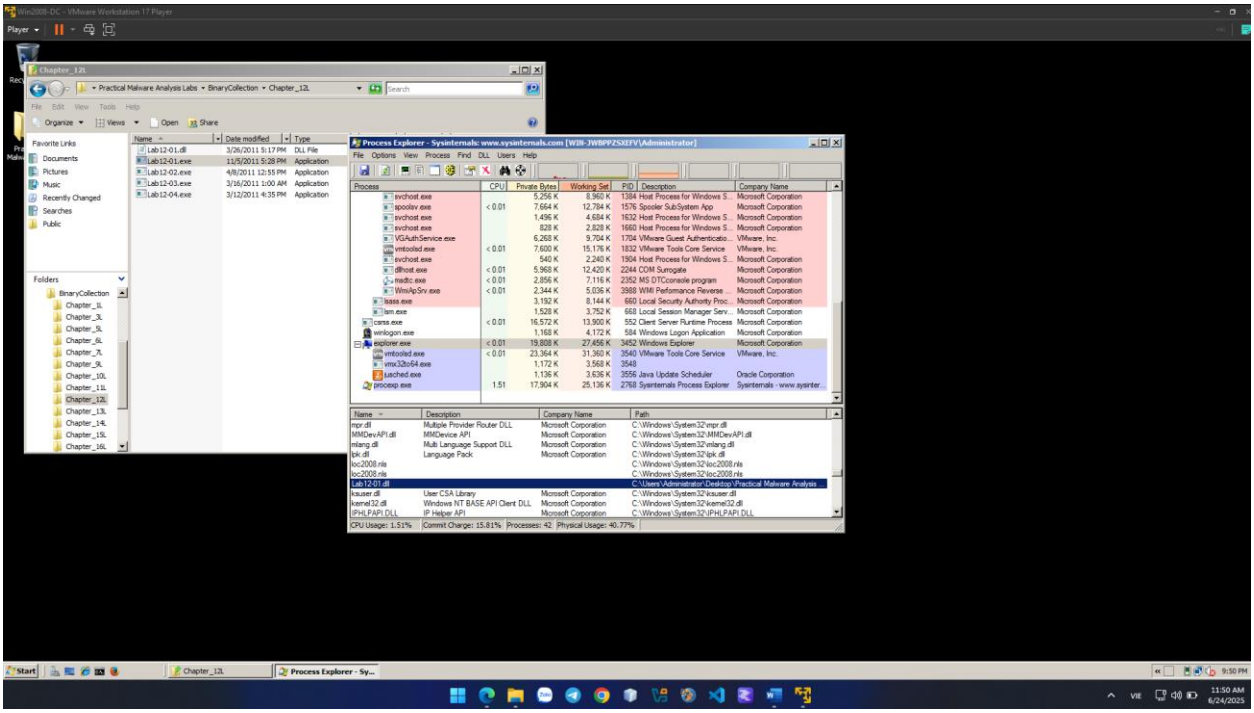
Open Process Explorer.

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	
jusched.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
Lab12-01.dll			C:\Users\Administrator\Desktop\Practical Malware Analysis ...
ksuser.dll	User CSA Library	Microsoft Corporation	C:\Windows\System32\ksuser.dll
kernel32.dll	Windows NT BASE API Client DLL	Microsoft Corporation	C:\Windows\System32\kernel32.dll
IPHLPAPI.DLL	IP Helper API	Microsoft Corporation	C:\Windows\System32\IPHLPAPI.DLL

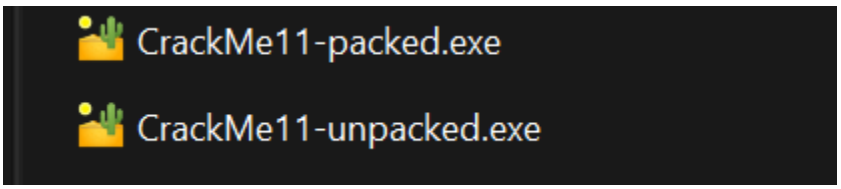
CPU Usage: 1.54% | Commit Charge: 15.79% | Processes: 42 | Physical Usage: 40.73%

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

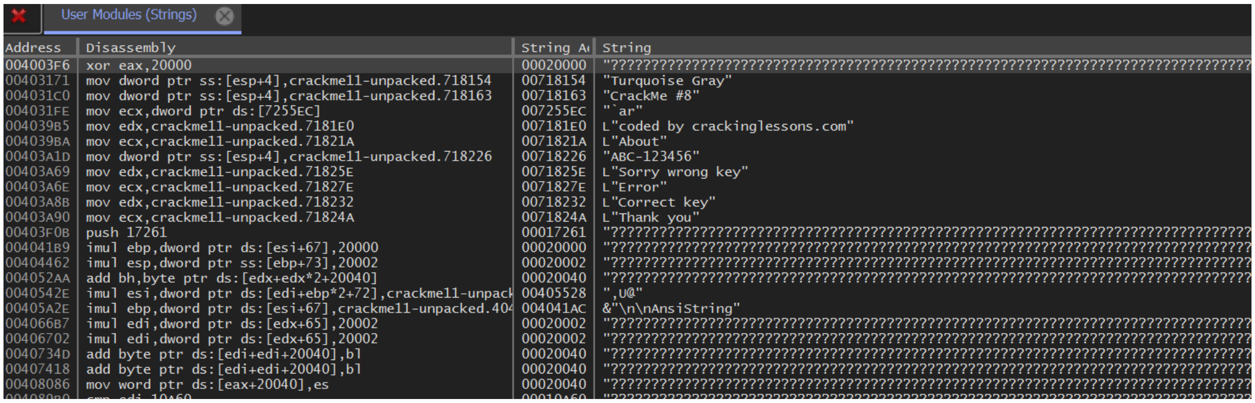
In this challenge, I dont focus on how to crack file, I want try to unpacked the file one successfully.



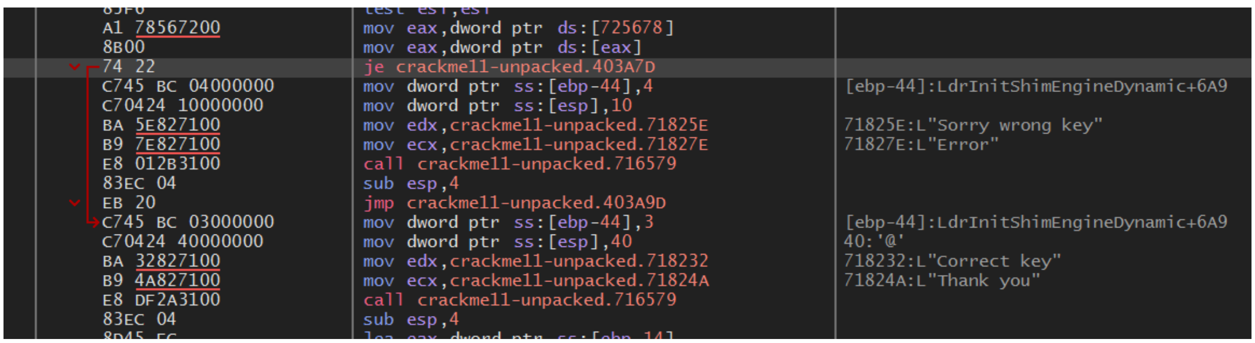
The second file can be cracked and patched successfully. But the first one is packed by UPX 3.91 packer.

For CrackMe11-unpacked.exe

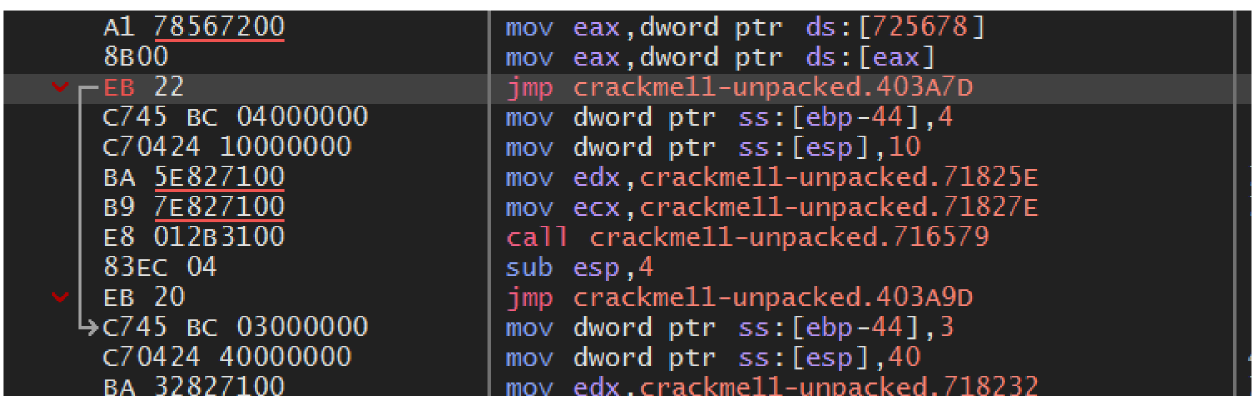
I find string references to view the code related (Sorry wrong key)

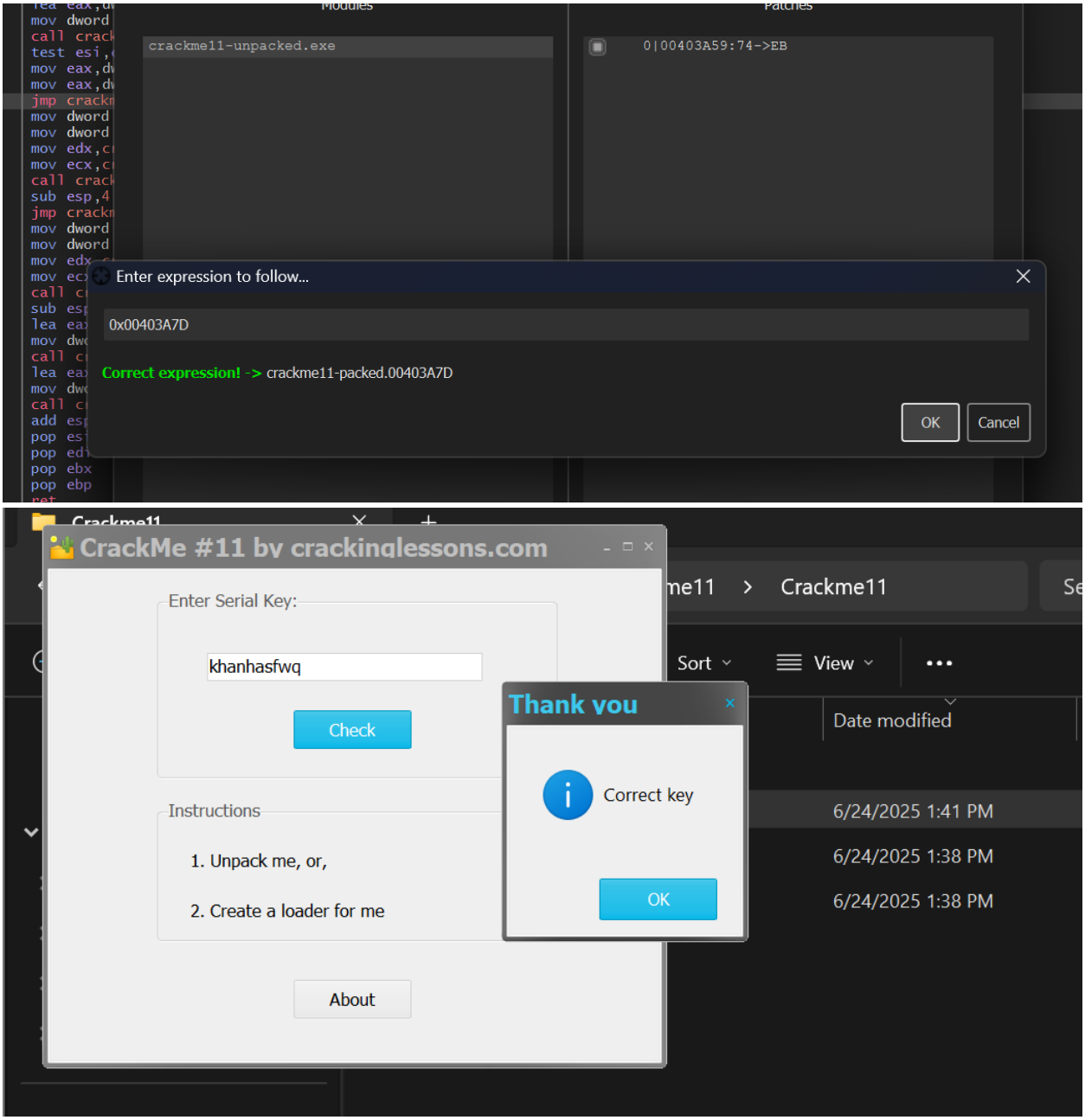


Look at the JE command can jump to Correct status.



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



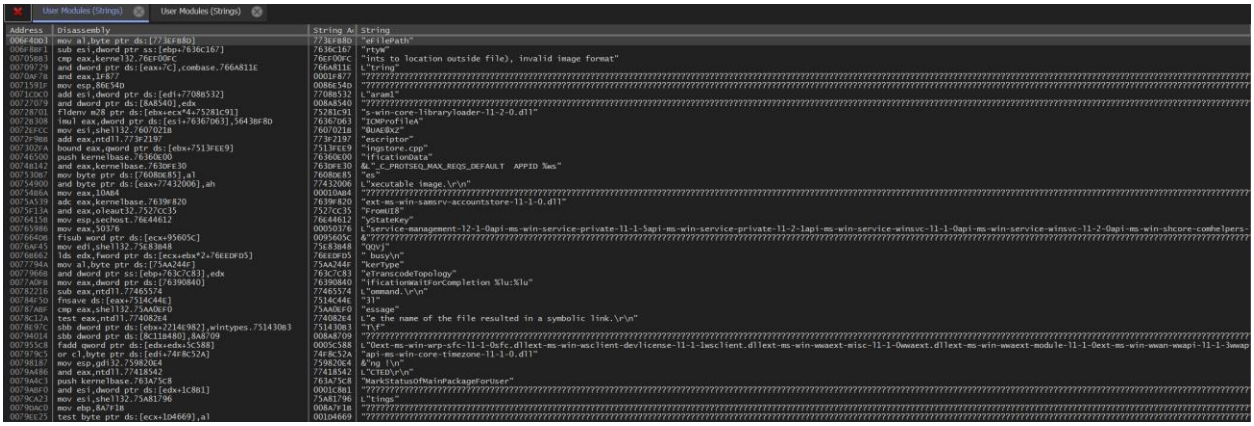


Success!!!

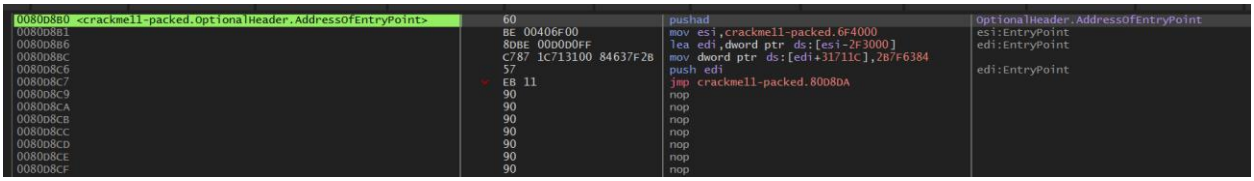
For CrackMe11-packed.exe – manually unpack the file packed by UPX:

This file is packed so I dont see the Wrong serial key String. The codes are also hidden too.

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

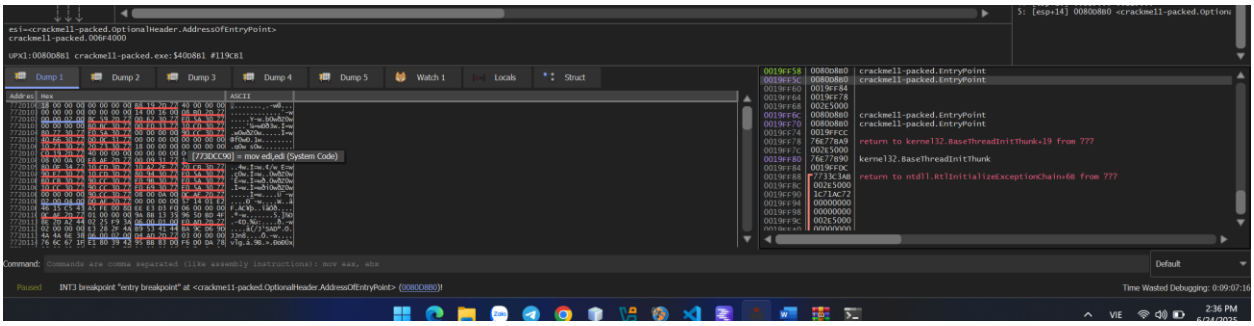


Find pushad

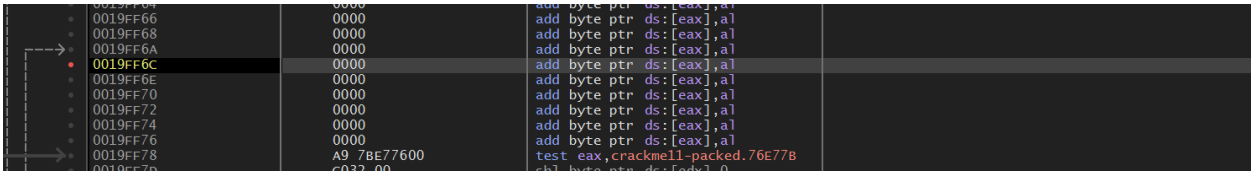


Notice DBP

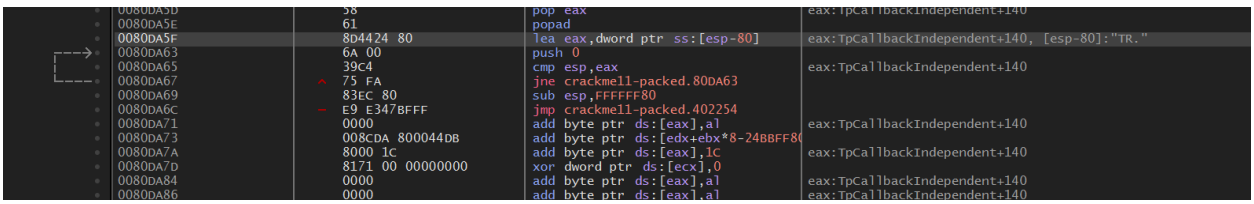
Find address of EBP:



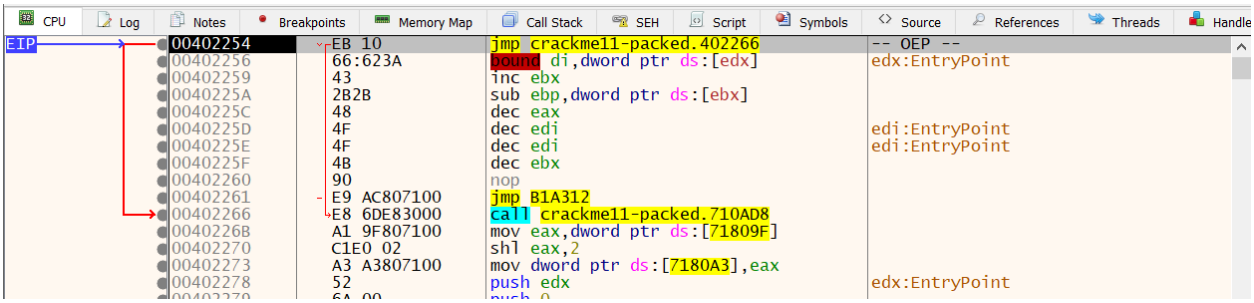
Set breakpoint:



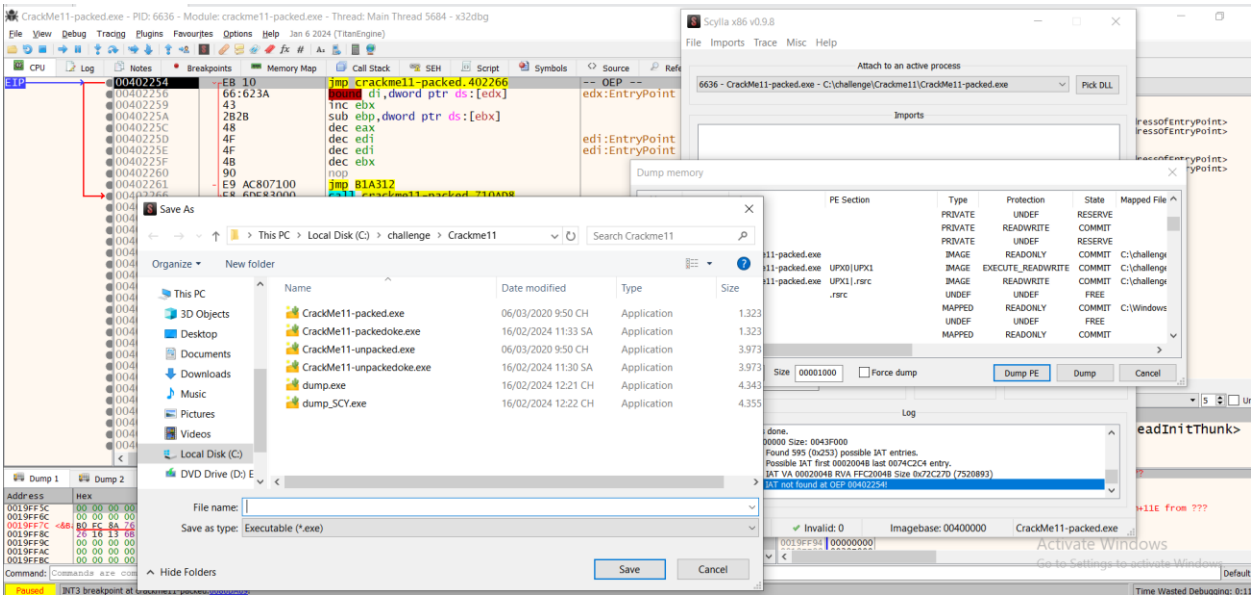
Run to reach breakpoint and find popad command:



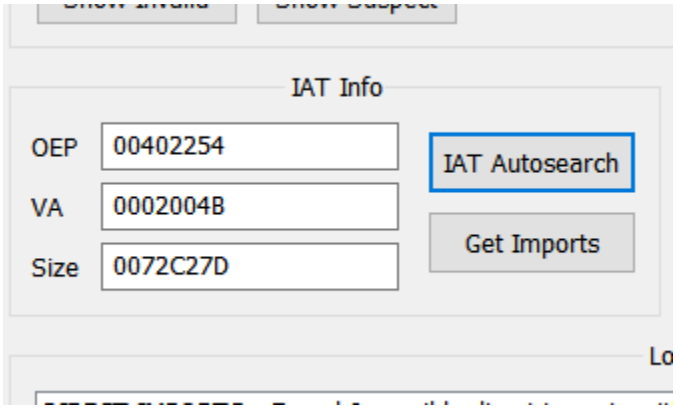
Find JMP neareast popad press f8 to display OEP (orgin entry point)



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.