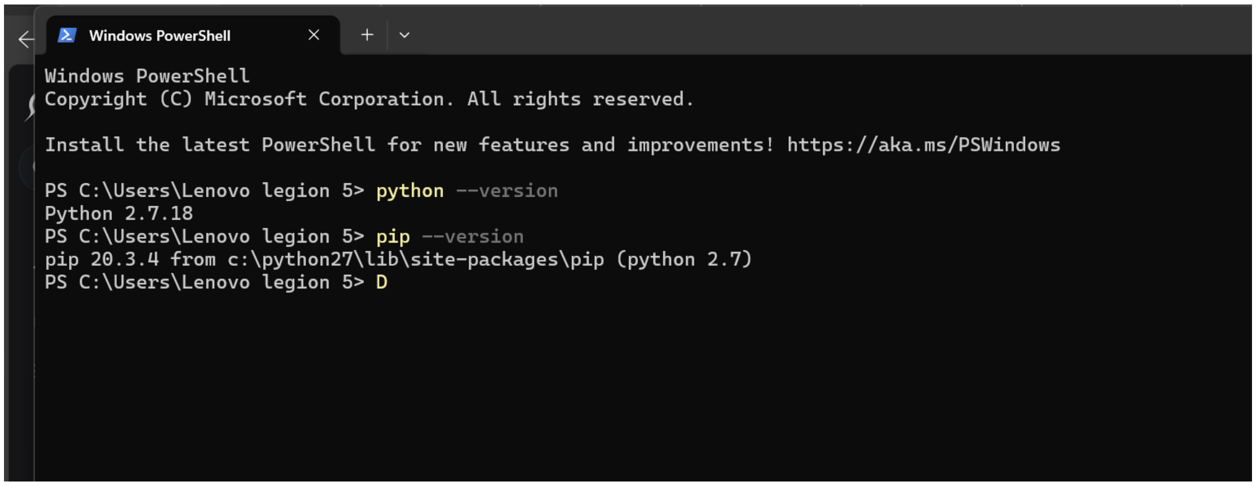
NGUYEN NAM KHANH – HE191159 – IA1902 – IAM302

LAB 13:

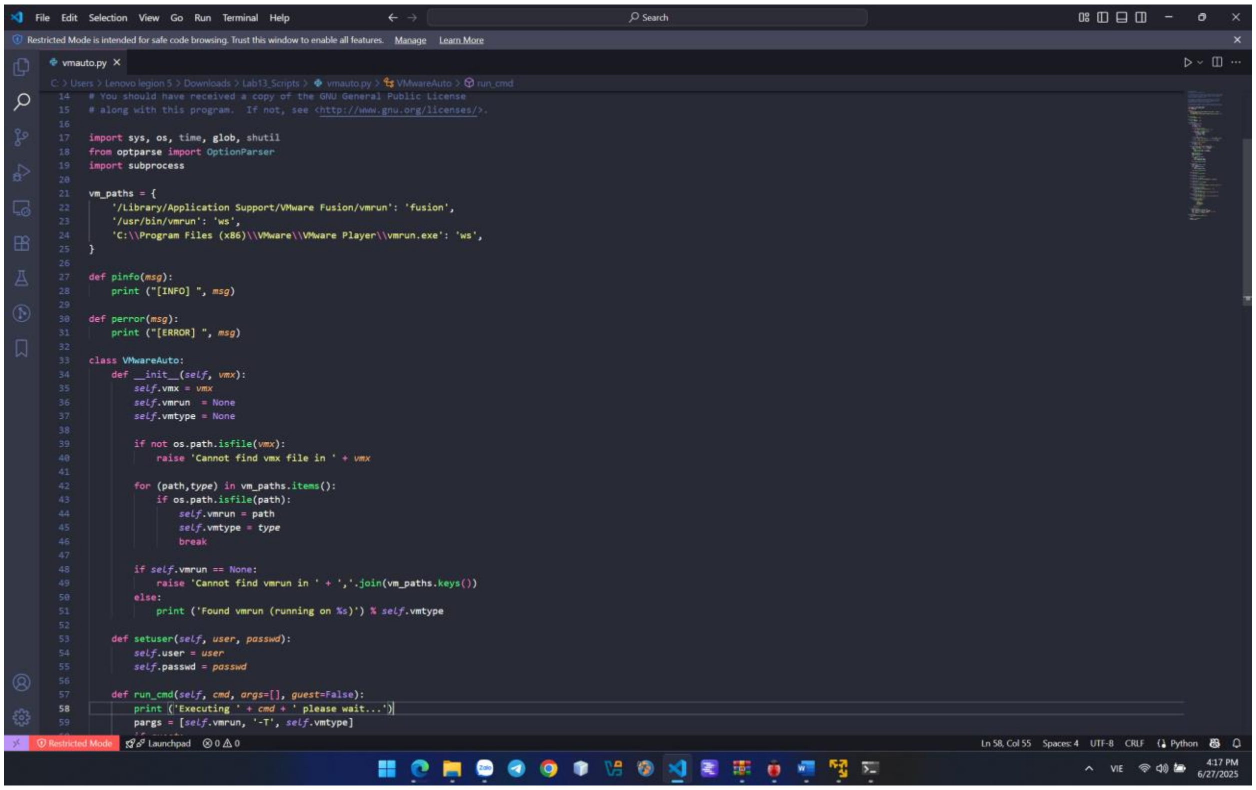
Automated Malware Analysis VMWARE Steps:

Setting python and pip:



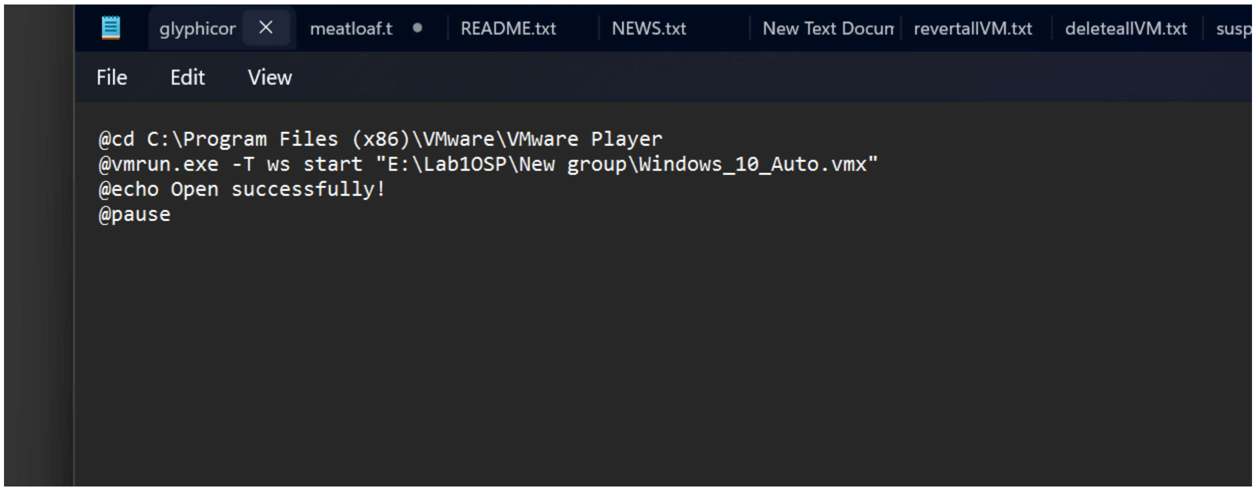
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Lenovo legion 5> python --version
Python 2.7.18
PS C:\Users\Lenovo legion 5> pip --version
pip 20.3.4 from c:\python27\lib\site-packages\pip (python 2.7)
PS C:\Users\Lenovo legion 5> D
```

Content of vmauto.py

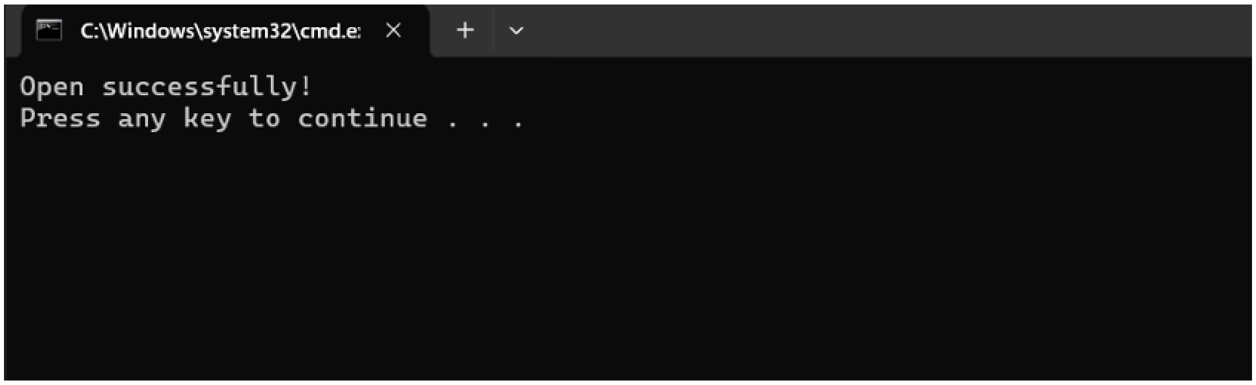


```python
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <http://www.gnu.org/licenses/>.

import sys, os, time, glob, shutil
from optparse import OptionParser
import subprocess

vm_paths = {
    '/Library/Application Support/VMware Fusion/vmrun': 'fusion',
    '/usr/bin/vmrun': 'ws',
    'C:\\Program Files (x86)\\VMware\\VMware Player\\vmrun.exe': 'ws',
}

def pinfo(msg):
    print ("[INFO] ", msg)

def perror(msg):
    print ("[ERROR] ", msg)

class VMwareAuto:
    def __init__(self, vmx):
        self.vmx = vmx
        self.vmrun  = None
        self.vmtype = None

        if not os.path.isfile(vmx):
            raise 'Cannot find vmx file in ' + vmx

        for (path,type) in vm_paths.items():
            if os.path.isfile(path):
                self.vmrun = path
                self.vmtype = type
                break

        if self.vmrun == None:
            raise 'Cannot find vmrun in ' + ','.join(vm_paths.keys())
        else:
            print ('Found vmrun (running on %s)') % self.vmtype

    def setuser(self, user, passwd):
        self.user = user
        self.passwd = passwd

    def run_cmd(self, cmd, args=[], guest=False):
        print ('Executing ' + cmd + ' please wait...')
        pargs = [self.vmrun, '-T', self.vmtype]
```

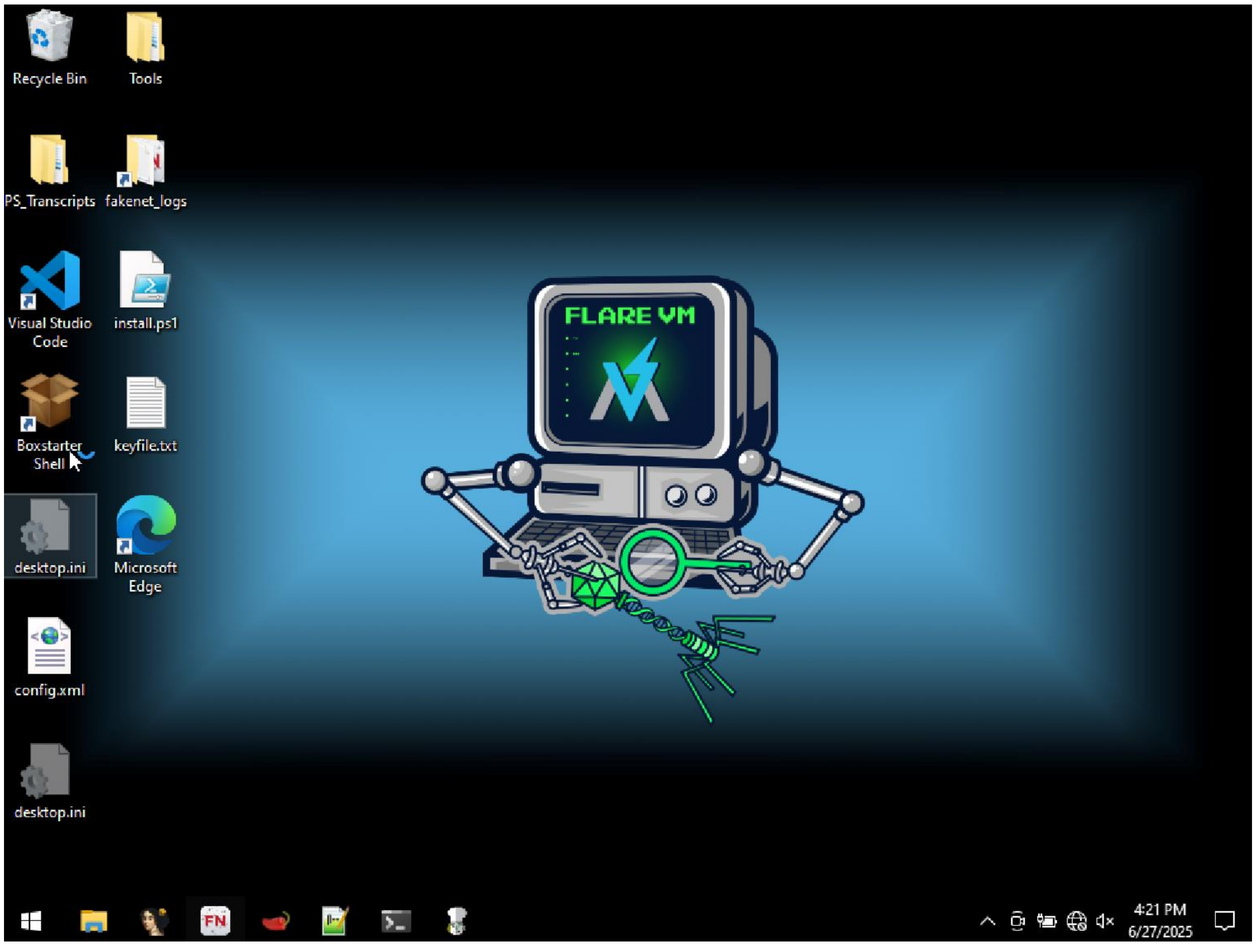
Create script to turn on Windows 10 virtual machine in Vmware.



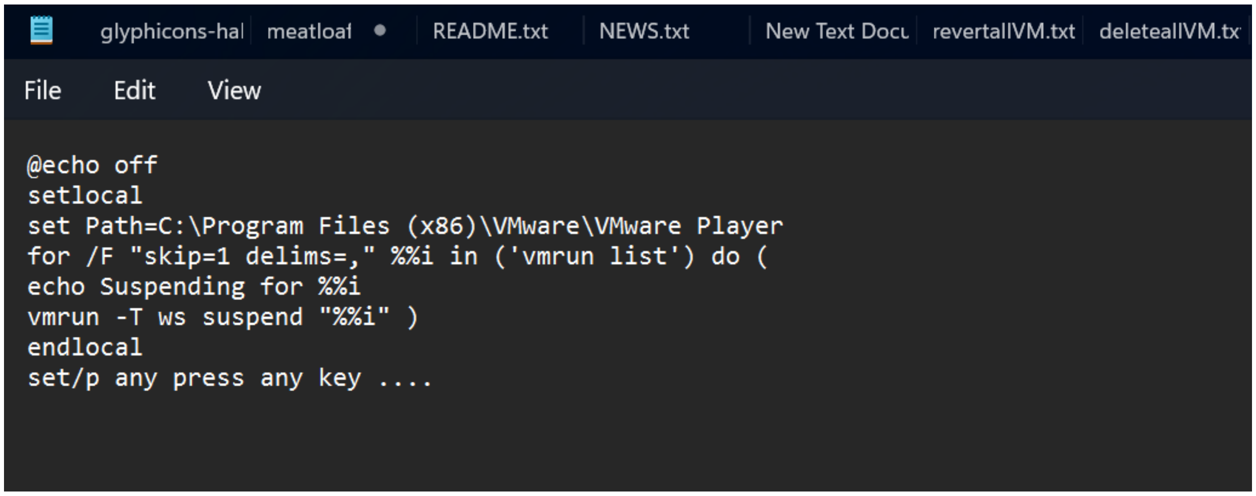
```
@cd C:\Program Files (x86)\VMware\VMware Player
@vmrun.exe -T ws start "E:\Lab1OSP\New group\Windows_10_Auto.vmx"
@echo Open successfully!
@pause
```



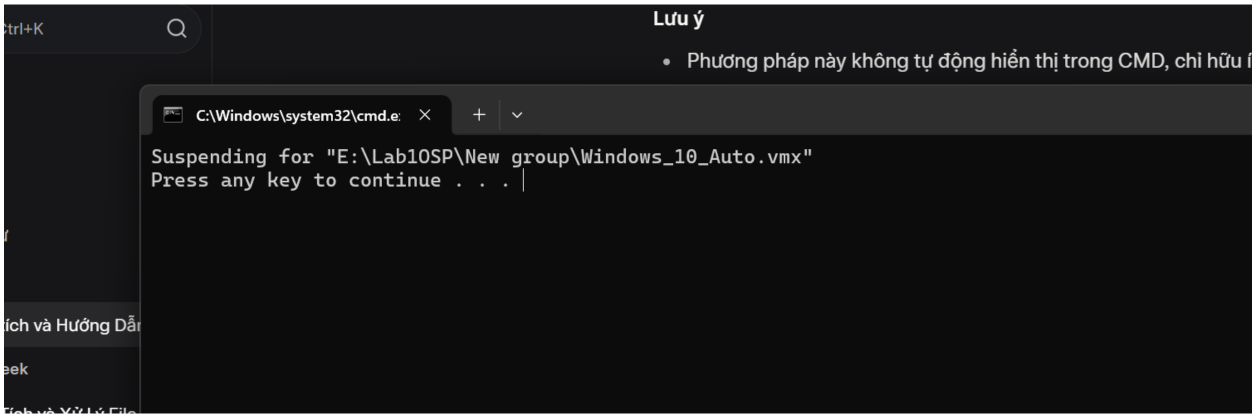
```
Open successfully!
Press any key to continue . . .
```


Recycle Bin
Tools
PS_Transcripts
fakenet_logs
Visual Studio Code
install.ps1
Boxstarter Shell
keyfile.txt
desktop.ini
Microsoft Edge
config.xml
desktop.ini
FLARE VM
4:21 PM
6/27/2025

Create a script to suspend all virtual machines running on VMware.

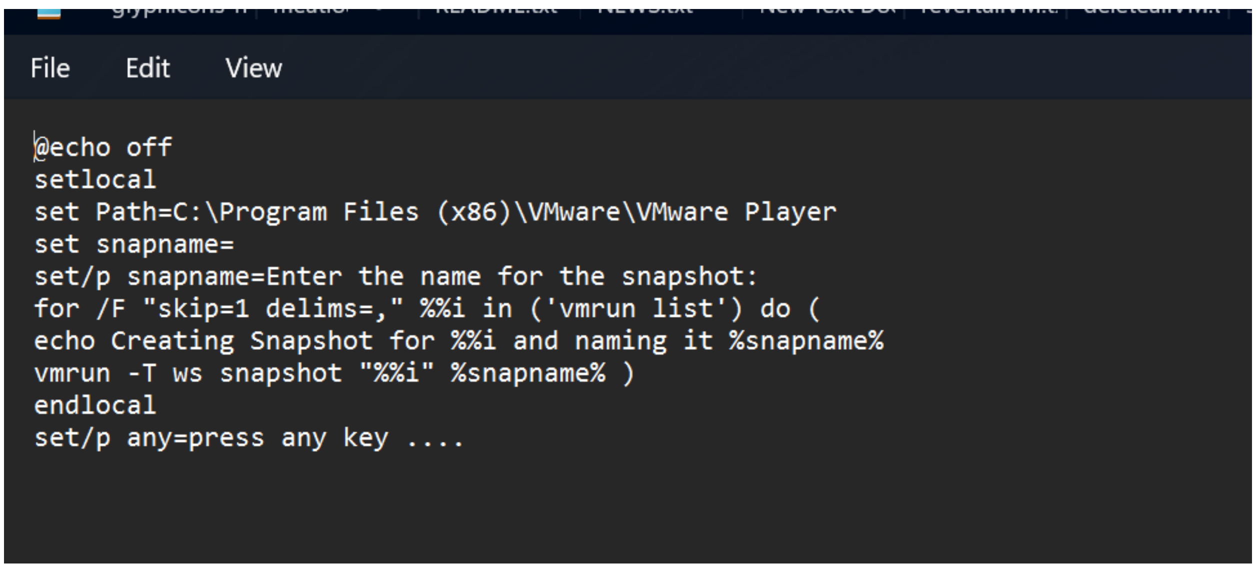


```
@echo off
setlocal
set Path=C:\Program Files (x86)\VMware\VMware Player
for /F "skip=1 delims=," %%i in ('vmrun list') do (
echo Suspending for %%i
vmrun -T ws suspend "%%i" )
endlocal
set/p any press any key ....
```

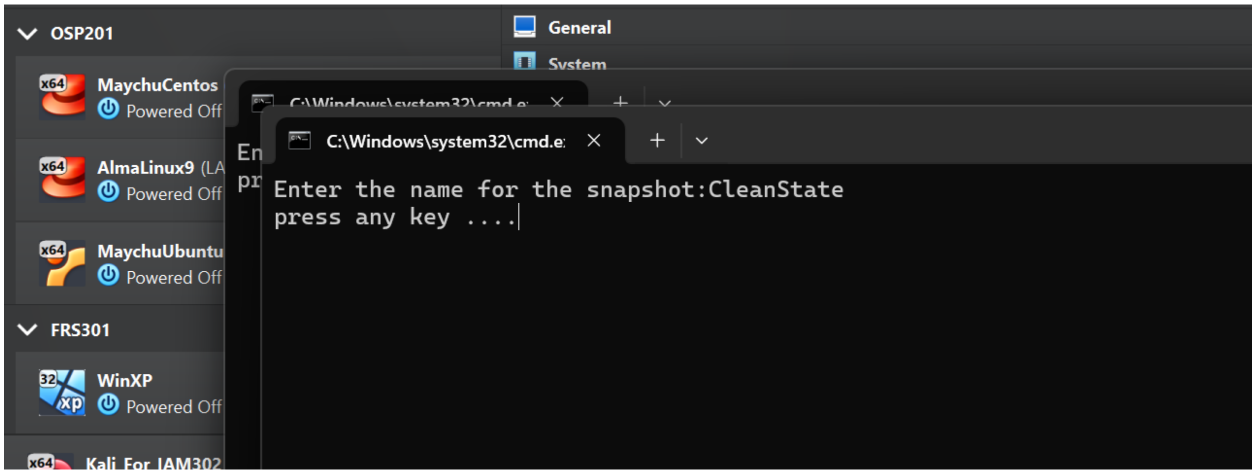

trl+K
Lưu ý
• Phương pháp này không tự động hiển thị trong CMD, chỉ hữu í
C:\Windows\system32\cmd.e
ích và Hướng Dẫn
eek

```
Suspending for "E:\Lab1OSP\New group\Windows_10_Auto.vmx"
Press any key to continue . . .
```

Create a script to create snapshots for virtual machines running on VMware.

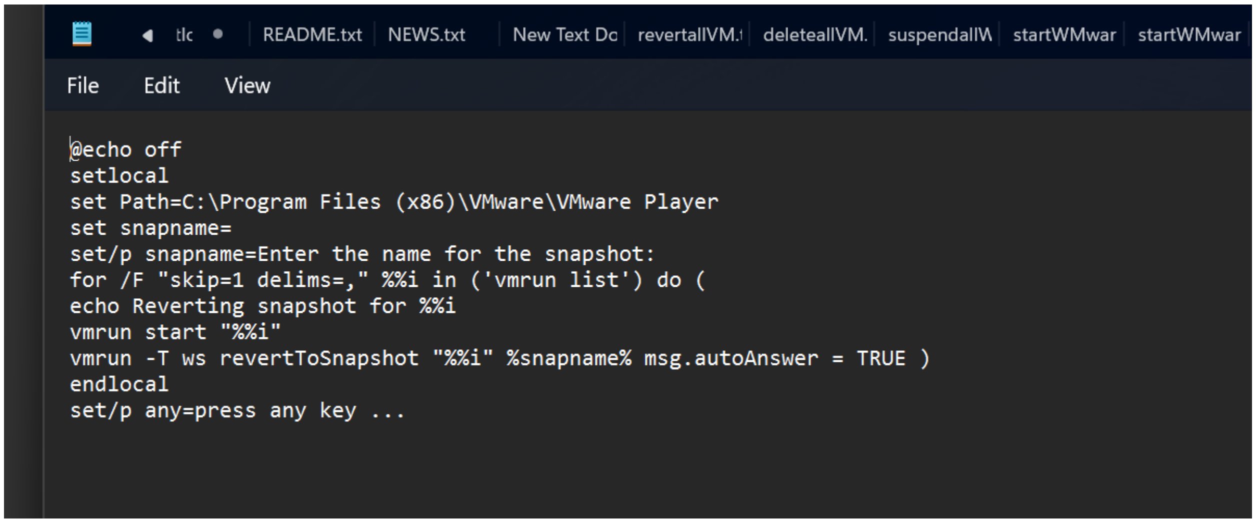


```
@echo off
setlocal
set Path=C:\Program Files (x86)\VMware\VMware Player
set snapname=
set/p snapname=Enter the name for the snapshot:
for /F "skip=1 delims=," %%i in ('vmrun list') do (
echo Creating Snapshot for %%i and naming it %snapname%
vmrun -T ws snapshot "%%i" %snapname% )
endlocal
set/p any=press any key ....
```

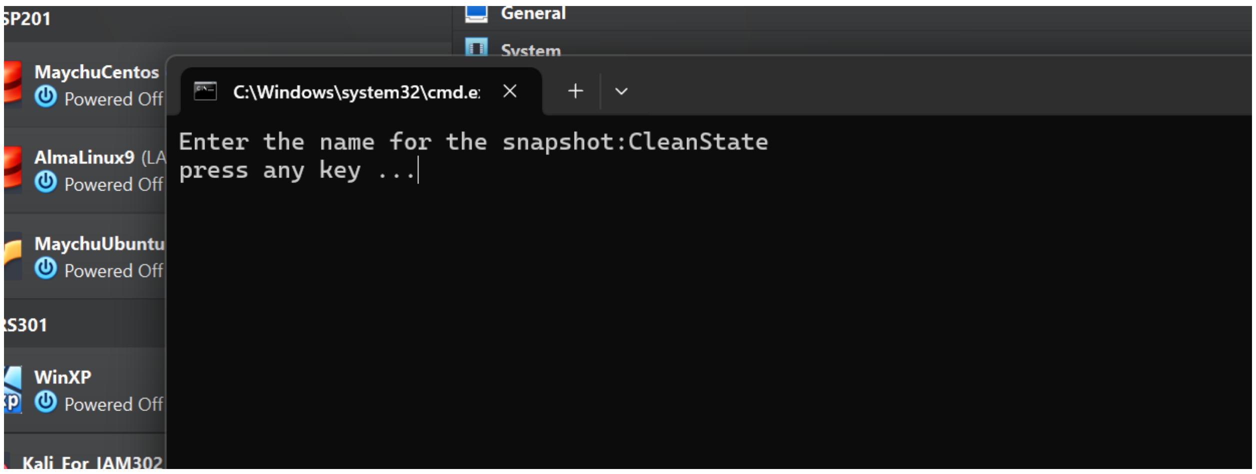


```
Enter the name for the snapshot:CleanState
press any key ....
```

Create a script to revert all snapshots for virtual machines running on VMware.

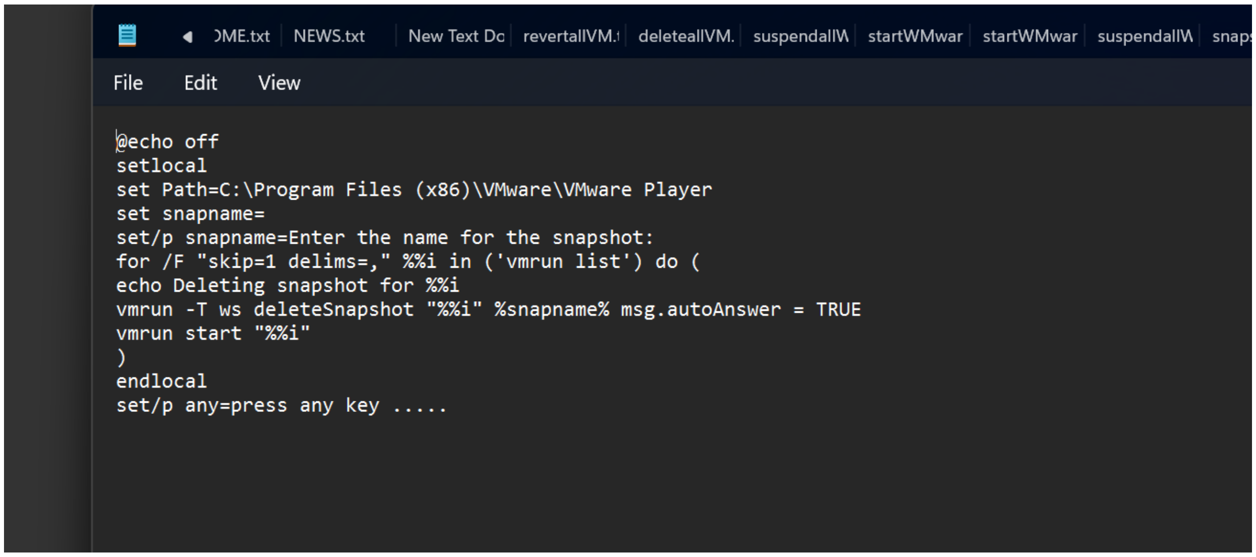


```
@echo off
setlocal
set Path=C:\Program Files (x86)\VMware\VMware Player
set snapname=
set/p snapname=Enter the name for the snapshot:
for /F "skip=1 delims=," %%i in ('vmrun list') do (
echo Reverting snapshot for %%i
vmrun start "%%i"
vmrun -T ws revertToSnapshot "%%i" %snapname% msg.autoAnswer = TRUE )
endlocal
set/p any=press any key ...
```

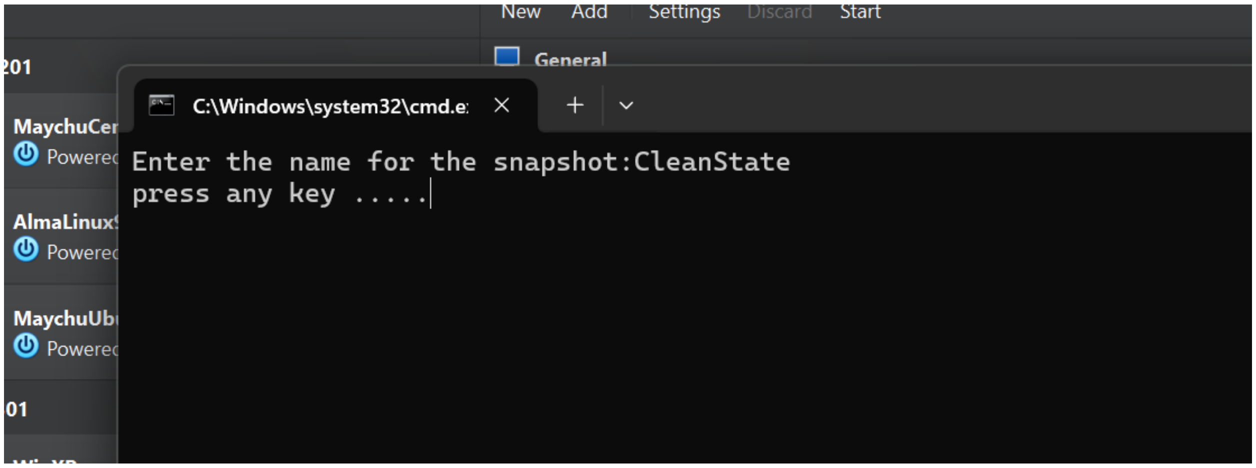


```
Enter the name for the snapshot:CleanState
press any key ...
```

Create a script to delete newly created snapshot files from the above script



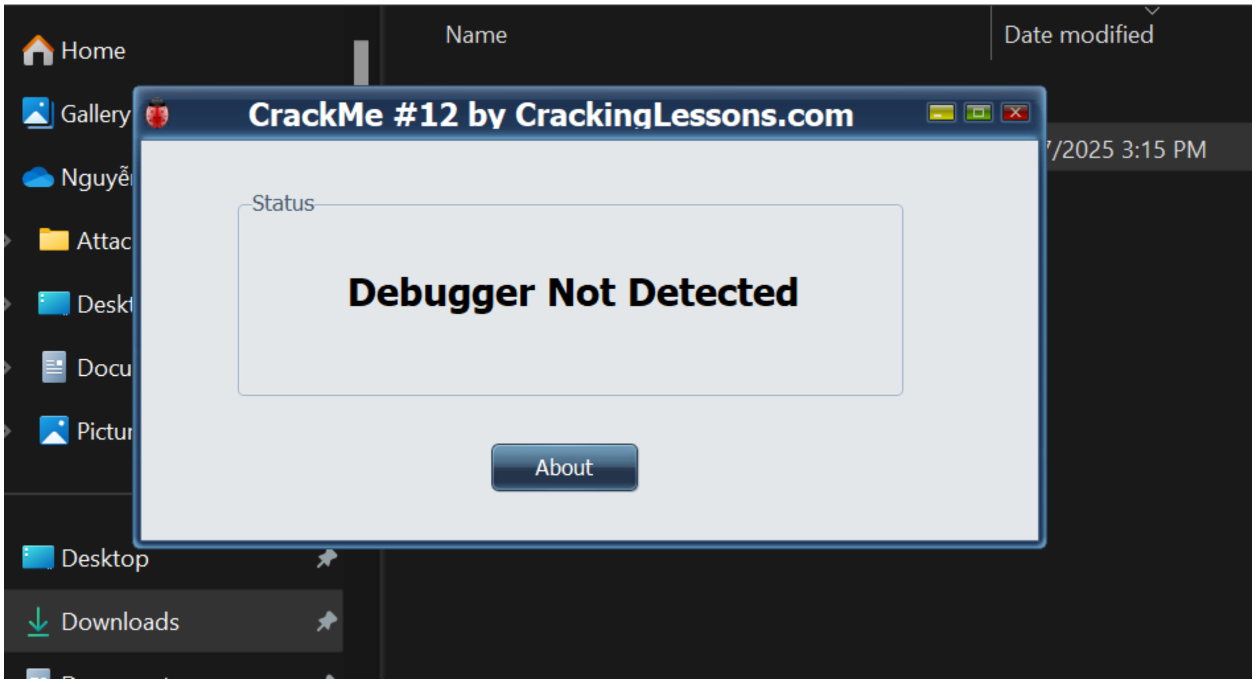
```
@echo off
setlocal
set Path=C:\Program Files (x86)\VMware\VMware Player
set snapname=
set/p snapname=Enter the name for the snapshot:
for /F "skip=1 delims=," %%i in ('vmrun list') do (
echo Deleting snapshot for %%i
vmrun -T ws deleteSnapshot "%%i" %snapname% msg.autoAnswer = TRUE
vmrun start "%%i"
)
endlocal
set/p any=press any key .....
```



```
Enter the name for the snapshot:CleanState
press any key .....
```

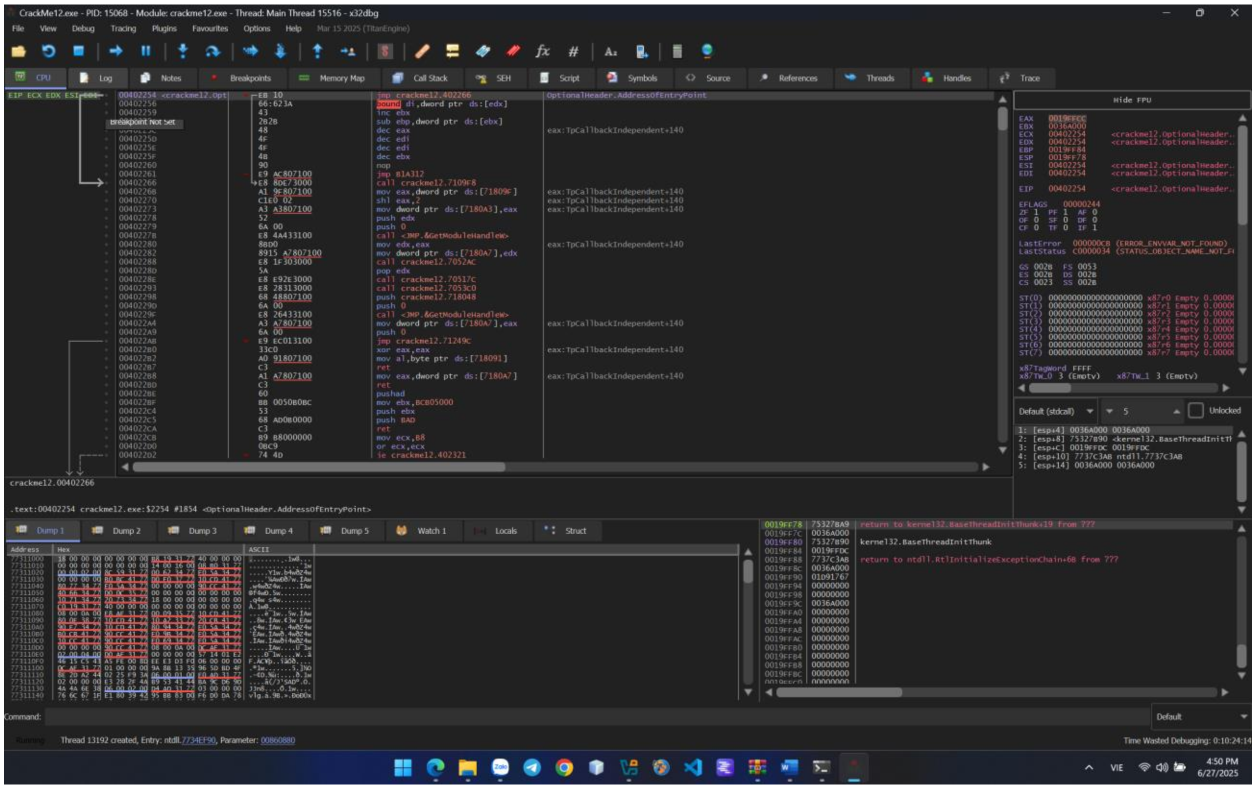
CrackMe #12

This CrackMe has Anti-Debugging features. If you open it with a debugger and then run, it will detect the debugger and exit. Your task is to bypass the anti-debugging feature, so that the program will continue to run and show the window below:



Debugger Not Detected

Open in x32dbg and press F9 to run, but nothing appeared. May be this program has anti debugger so it didn't run in x32dbg tools.



```
CrackMe12.exe - PID: 15068 - Module: crackme12.exe - Thread: Main Thread 15516 - x32dbg
```

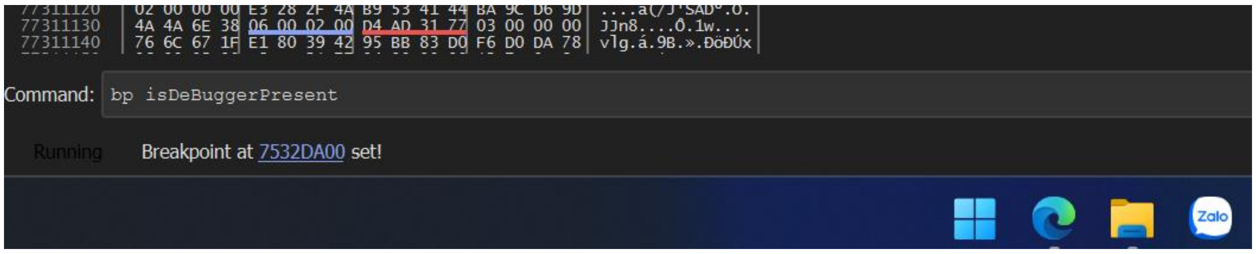
```
crackme12.00402266
.text:00402254 crackme12.exe:$2254 #1854 <OptionalHeader.AddressOfEntryPoint>
Command:
Running    Thread 13192 created, Entry: ntdll.7734EF90, Parameter: 00860880
Time Wasted Debugging: 0:10:24:14
```

To avoid anti-debugging, people came up with anti-anti-debugging, and there are 2 commonly used methods for it to work: setting breakpoints on anti-debugging APIs or use anti-anti-debugging plugins.

I will follow the way setting breakpoint API anti-debugging

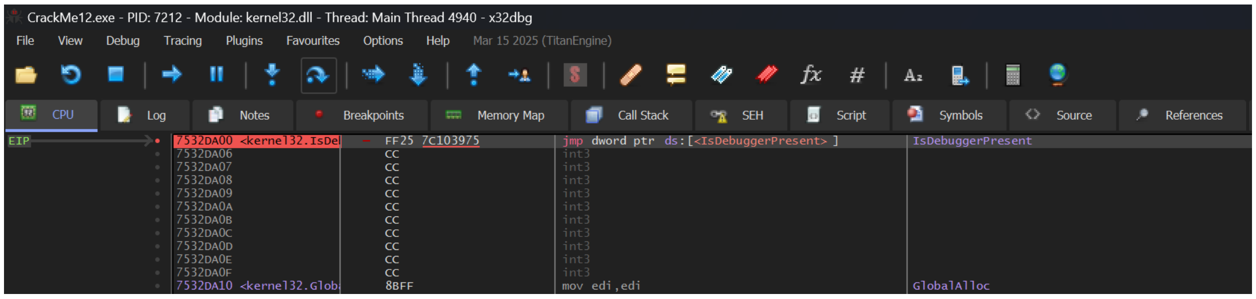
Set a breakpoint on the isDebuggerPresent() function to prevent the program from terminating immediately when clicking run.

Note: The isDebuggerPresent() function is the most commonly used API to check whether a program is running with a debugger or not.



```
77311130  4A 4A 6E 38 06 00 02 00 D4 AD 31 77 03 00 00 00  JJn8....Ô.1w....
77311140  76 6C 67 1F E1 80 39 42 95 BB 83 D0 F6 D0 DA 78  vlg.á.9B.».ÐöÐÚx

Command: bp isDeBuggerPresent

Running    Breakpoint at 7532DA00 set!
```

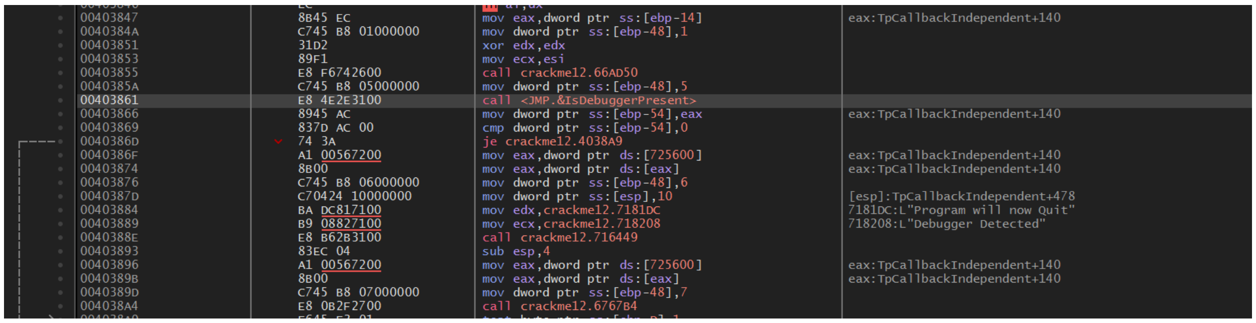
Click run to see where that function is located.



CrackMe12.exe - PID: 7212 - Module: kernel32.dll - Thread: Main Thread 4940 - x32dbg

```
EIP  7532DA00 <kernel32.IsDe   FF25 7C103975   jmp dword ptr ds:[<IsDebuggerPresent> ]   IsDebuggerPresent
     7532DA06                  CC              int3
     7532DA07                  CC              int3
     7532DA08                  CC              int3
     7532DA09                  CC              int3
     7532DA0A                  CC              int3
     7532DA0B                  CC              int3
     7532DA0C                  CC              int3
     7532DA0D                  CC              int3
     7532DA0E                  CC              int3
     7532DA0F                  CC              int3
     7532DA10 <kernel32.Glob   8BFF            mov edi,edi                               GlobalAlloc
```

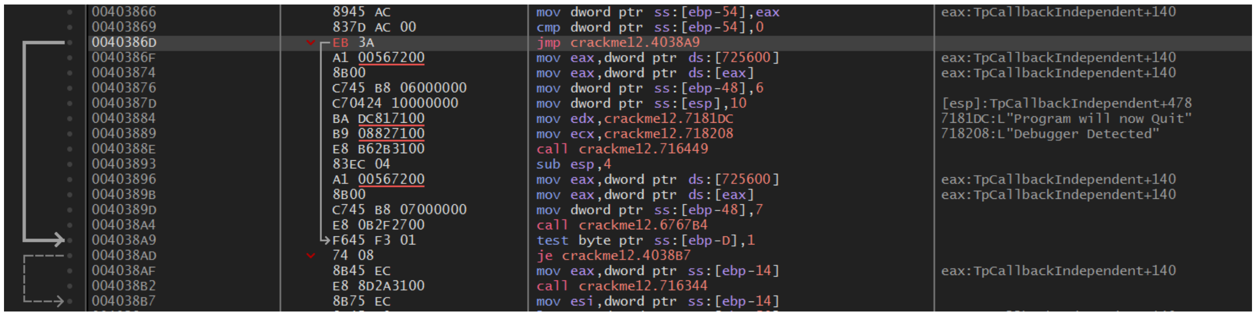
Click "run to user code" to view the overall code.



```
00403847   8B45 EC            mov eax,dword ptr ss:[ebp-14]       eax:TpCallbackIndependent+140
0040384A   C745 B8 01000000   mov dword ptr ss:[ebp-48],1
00403851   31D2               xor edx,edx
00403853   89F1               mov ecx,esi
00403855   E8 F6742600        call crackme12.66AD50
0040385A   C745 B8 05000000   mov dword ptr ss:[ebp-48],5
00403861   E8 4E2E3100        call <JMP.&IsDebuggerPresent>
00403866   8945 AC            mov dword ptr ss:[ebp-54],eax       eax:TpCallbackIndependent+140
00403869   837D AC 00         cmp dword ptr ss:[ebp-54],0
0040386D   74 3A              je crackme12.4038A9
0040386F   A1 00567200        mov eax,dword ptr ds:[725600]       eax:TpCallbackIndependent+140
00403874   8B00               mov eax,dword ptr ds:[eax]          eax:TpCallbackIndependent+140
00403876   C745 B8 06000000   mov dword ptr ss:[ebp-48],6
0040387D   C70424 10000000    mov dword ptr ss:[esp],10           [esp]:TpCallbackIndependent+478
00403884   BA DC817100        mov edx,crackme12.7181DC            7181DC:L"Program will now Quit"
00403889   B9 08827100        mov ecx,crackme12.718208            718208:L"Debugger Detected"
0040388E   E8 B62B3100        call crackme12.716449
00403893   83EC 04            sub esp,4
00403896   A1 00567200        mov eax,dword ptr ds:[725600]       eax:TpCallbackIndependent+140
0040389B   8B00               mov eax,dword ptr ds:[eax]          eax:TpCallbackIndependent+140
0040389D   C745 B8 07000000   mov dword ptr ss:[ebp-48],7
004038A4   E8 0B2F2700        call crackme12.6767B4
```

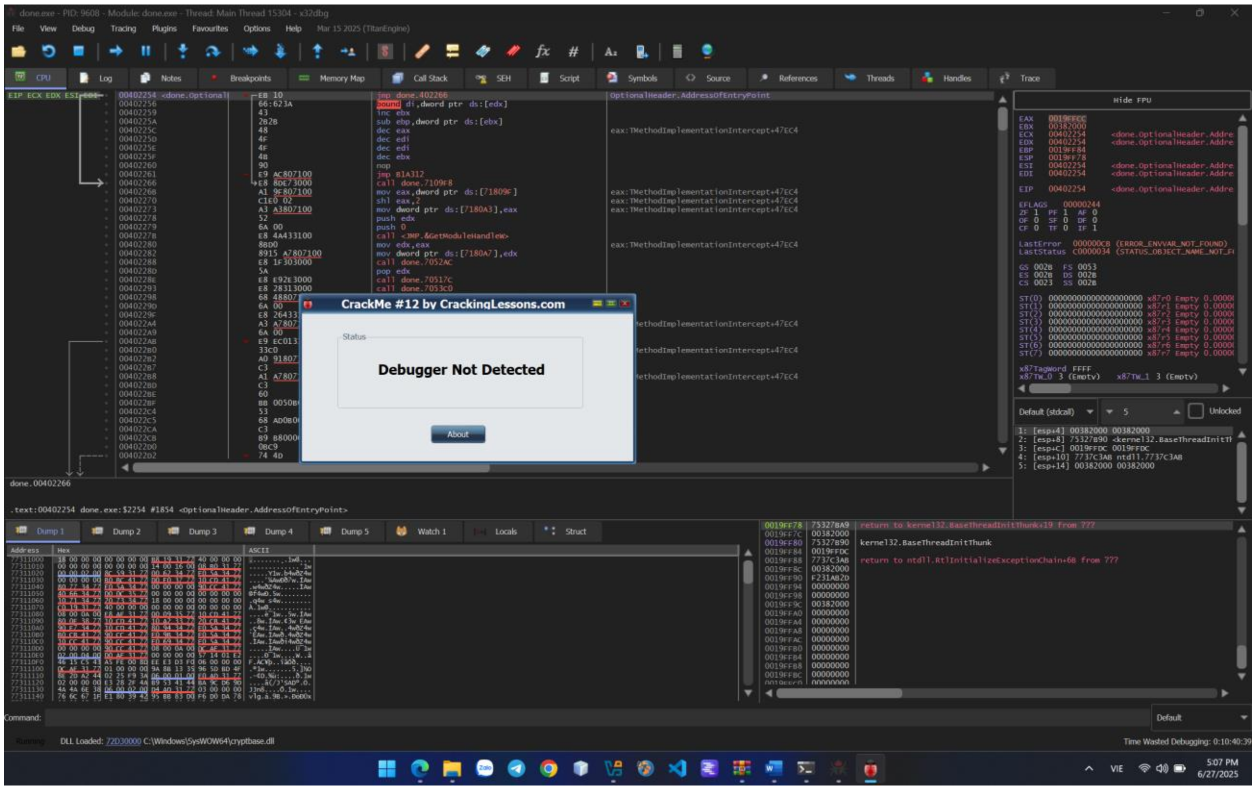
It appears to check what value the function returns, then stores it from eax, compares it with 0, and if it's not equal to 0, the je instruction won't be executed and will display the quit program window.

Change je to jmp to always jump and prevent the quit window from appearing. After making the modification, click run again and the program has reappeared, successfully cracking the anti-debugger.



```
00403866   8945 AC            mov dword ptr ss:[ebp-54],eax       eax:TpCallbackIndependent+140
00403869   837D AC 00         cmp dword ptr ss:[ebp-54],0
0040386D   EB 3A              jmp crackme12.4038A9
0040386F   A1 00567200        mov eax,dword ptr ds:[725600]       eax:TpCallbackIndependent+140
00403874   8B00               mov eax,dword ptr ds:[eax]          eax:TpCallbackIndependent+140
00403876   C745 B8 06000000   mov dword ptr ss:[ebp-48],6
0040387D   C70424 10000000    mov dword ptr ss:[esp],10           [esp]:TpCallbackIndependent+478
00403884   BA DC817100        mov edx,crackme12.7181DC            7181DC:L"Program will now Quit"
00403889   B9 08827100        mov ecx,crackme12.718208            718208:L"Debugger Detected"
0040388E   E8 B62B3100        call crackme12.716449
00403893   83EC 04            sub esp,4
00403896   A1 00567200        mov eax,dword ptr ds:[725600]       eax:TpCallbackIndependent+140
0040389B   8B00               mov eax,dword ptr ds:[eax]          eax:TpCallbackIndependent+140
0040389D   C745 B8 07000000   mov dword ptr ss:[ebp-48],7
004038A4   E8 0B2F2700        call crackme12.6767B4
004038A9   F645 F3 01         test byte ptr ss:[ebp-D],1
004038AD   74 08              je crackme12.4038B7
004038AF   8B45 EC            mov eax,dword ptr ss:[ebp-14]       eax:TpCallbackIndependent+140
004038B2   E8 8D2A3100        call crackme12.716344
004038B7   8B75 EC            mov esi,dword ptr ss:[ebp-14]
```

Debugging successfully!



done.exe - PID: 9608 - Module: done.exe - Thread: Main Thread 15304 - x32dbg

```
EIP ECX EDX ESI EDI
00402254 <done.OptionalH  EB 10            jmp done.402266              OptionalHeader.AddressOfEntryPoint
00402256                  66:623A          bound di,dword ptr ds:[edx]
00402259                  43               inc ebx
0040225A                  2B2B             sub ebp,dword ptr ds:[ebx]
0040225C                  48               dec eax                      eax:TMethodImplementationIntercept+47EC4
0040225D                  4F               dec edi
0040225E                  4F               dec edi
0040225F                  4B               dec ebx
00402260                  90               nop
00402261                  E9 AC807100      jmp 81A312
00402266                  E8 8DE73000      call done.7109F8
0040226B                  A1 9F807100      mov eax,dword ptr ds:[71809F]  eax:TMethodImplementationIntercept+47EC4
00402270                  C1E0 02          shl eax,2                    eax:TMethodImplementationIntercept+47EC4
00402273                  A3 A3807100      mov dword ptr ds:[7180A3],eax  eax:TMethodImplementationIntercept+47EC4
00402278                  52               push edx
00402279                  6A 00            push 0
0040227B                  E8 4A433100      call <JMP.&GetModuleHandleW>
00402280                  8BD0             mov edx,eax                  eax:TMethodImplementationIntercept+47EC4
00402282                  8915 A7807100    mov dword ptr ds:[7180A7],edx
00402288                  E8 1F303000      call done.7052AC
0040228D                  5A               pop edx
0040228E                  E8 E92E3000      call done.70517C
00402293                  E8 28313000      call done.7053C0
00402298                  68 48807
0040229D                  6A 00
0040229F                  E8 26433
004022A4                  A3 A7807                                      ethodImplementationIntercept+47EC4
004022A9                  6A 00
004022AB                  E9 EC013                                      ethodImplementationIntercept+47EC4
004022B0                  33C0
004022B2                  A0 91807
004022B7                  C3
004022B8                  A1 A7807                                      ethodImplementationIntercept+47EC4
004022BD                  C3
004022BE                  60
004022BF                  BB 0050B
004022C4                  53
004022C5                  68 AD0B0
004022CA                  C3
004022CB                  B9 B8000
004022D0                  0BC9
004022D2                  74 4D
```

CrackMe #12 by CrackingLessons.com

Status

Debugger Not Detected

```
Hide FPU

EAX     0019FFCC
EBX     00382000
ECX     00402254     <done.OptionalHeader.Addre
EDX     00402254     <done.OptionalHeader.Addre
EBP     0019FF84
ESP     0019FF78
ESI     00402254     <done.OptionalHeader.Addre
EDI     00402254     <done.OptionalHeader.Addre

EIP     00402254     <done.OptionalHeader.Addre

EFLAGS    00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError   000000CB (ERROR_ENVVAR_NOT_FOUND)
LastStatus  C0000034 (STATUS_OBJECT_NAME_NOT_F

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

ST(0) 0000000000000000000 x87r0 Empty 0.0000
ST(1) 0000000000000000000 x87r1 Empty 0.0000
ST(2) 0000000000000000000 x87r2 Empty 0.0000
ST(3) 0000000000000000000 x87r3 Empty 0.0000
ST(4) 0000000000000000000 x87r4 Empty 0.0000
ST(5) 0000000000000000000 x87r5 Empty 0.0000
ST(6) 0000000000000000000 x87r6 Empty 0.0000
ST(7) 0000000000000000000 x87r7 Empty 0.0000

x87TagWord  FFFF
x87TW_0 3 (Empty)    x87TW_1 3 (Empty)
```

Default (stdcall) 5 Unlocked

```
1: [esp+4] 00382000 00382000
2: [esp+8] 75327B90 <kernel32.BaseThreadInitTh
3: [esp+C] 0019FFDC 0019FFDC
4: [esp+10] 7737C3AB ntdll.7737C3AB
5: [esp+14] 00382000 00382000
```

```
done.00402266

.text:00402254 done.exe:$2254 #1854 <OptionalHeader.AddressOfEntryPoint>
```

```
0019FF78  75327BA9  return to kernel32.BaseThreadInitThunk+19 from ???
0019FF7C  00382000
0019FF80  75327B90  kernel32.BaseThreadInitThunk
0019FF84  0019FFDC
0019FF88  7737C3AB  return to ntdll.RtlInitializeExceptionChain+6B from ???
0019FF8C  00382000
0019FF90  F231AB2D
0019FF94  00000000
0019FF98  00000000
0019FF9C  00382000
0019FFA0  00000000
0019FFA4  00000000
0019FFA8  00000000
0019FFAC  00000000
0019FFB0  00000000
0019FFB4  00000000
0019FFB8  00000000
0019FFBC  00000000
```

Command: Default

Running DLL Loaded: 72D30000 C:\Windows\SysWOW64\cryptbase.dll Time Wasted Debugging: 0:10:40:39

5:07 PM 6/27/2025

DONE!!!