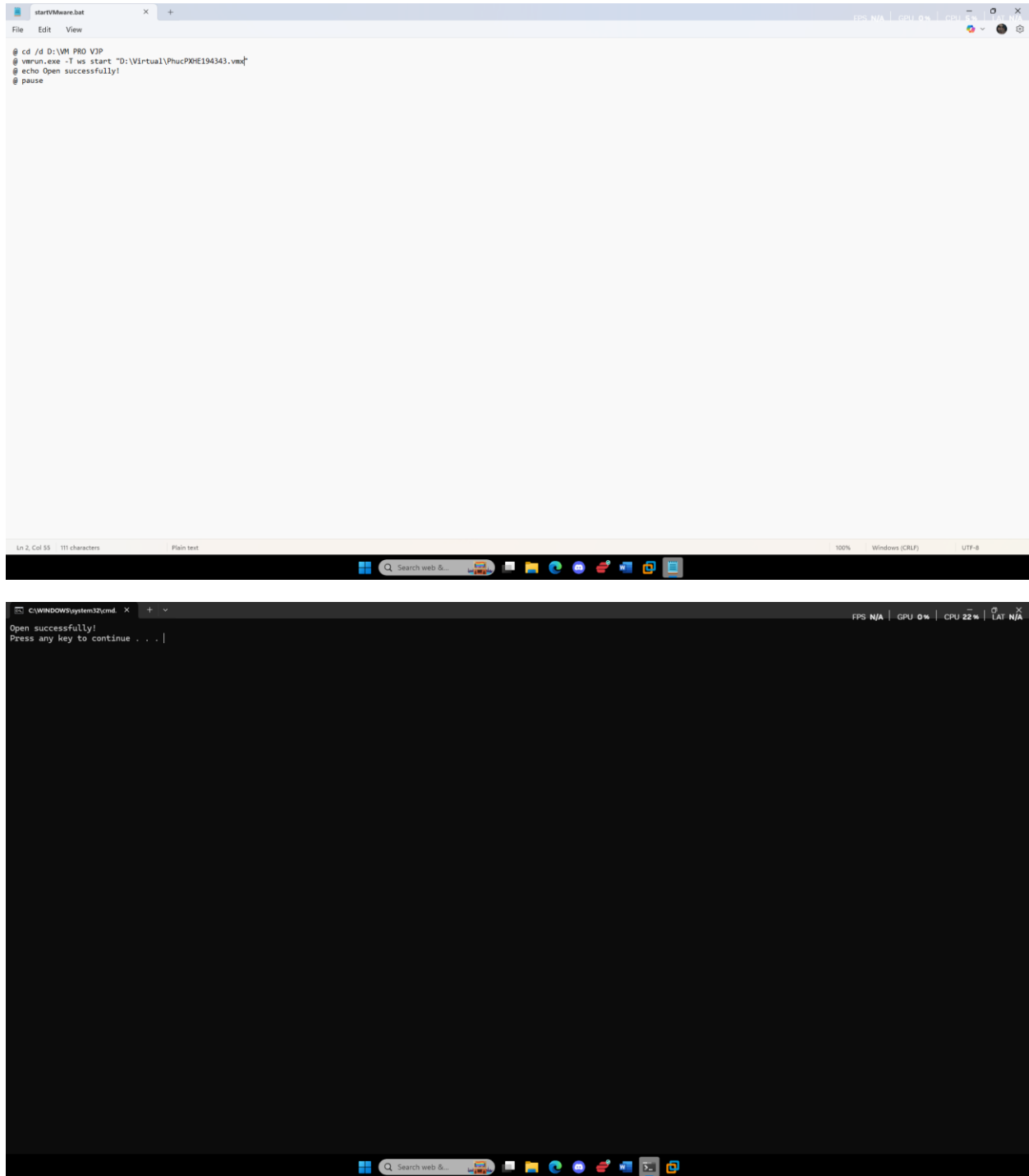


Lab 13



```
startVMware.bat suspendallVMware.bat
File Edit View
@echo off
setlocal
set path=D:\VM PRO VXP
for /f "skip=1 delims=" %i in ('verun list') do (
echo Suspending for %i
verun -T us suspend "%i"
)
endlocal
set /p any=press any key ...|
```

```
c:\WINDOWS\system32\cmd. X +
Suspending for D:\Virtual\PhucPXHE194343.vmx
press any key ...|
```

FPS N/A | GPU 0% | CPU 7% | LAT N/A

```
startVMware.bat suspendallVMware.bat snapshottallVMware.bat X +
File Edit View

@echo off
setlocal
set Path=D:\VM PRO VJP
set snapname=
set /p snapname=Enter the name for the snapshot:
for /F "skip=1 delims=" %X1 in ('vmlist') do (
    echo Creating Snapshot for %X1 and naming it %snapname%
    vmls -T vs snapshot "%X1" %snapname%
)
endlocal
set /p any=press any key ....

Ln 3, Col 23 293 characters Plain text 100% Windows (CRLF) UTF-8
```

```
C:\WINDOWS\system32\cmd. X + v FPS N/A GPU 0% CPU 0% LAT N/A
Enter the name for the snapshot: SYBAU
press any key ....


C:\WINDOWS\system32\cmd. X + v FPS N/A GPU 0% CPU 0% LAT N/A
```

startVMware.batsuspendVMware.batsnapshotVMware.batrevertVMware.bat

FileEditView

```
@echo off
setlocal
set Path=D:\VM PRO VJP
set snapname=
set /p snapname=Enter the name for the snapshot:
for /F "skip=1 delims=" %i in ('verun list') do (
echo Reverting snapshot for %i
verun -I vs revertToSnapshot "%i" %snapname% msg.autoAnswer = TRUE
verun start "%i"
)
endlocal
set /p any=press any key ...
```

Ln 3, Col 23 316 charactersPlain text100%Windows (CRLF)UTF-8





startVMware.batsuspendallVMware.batsnapshotallVMware.batrevertallVMware.batdeleteallVMware.bat

FileEditView

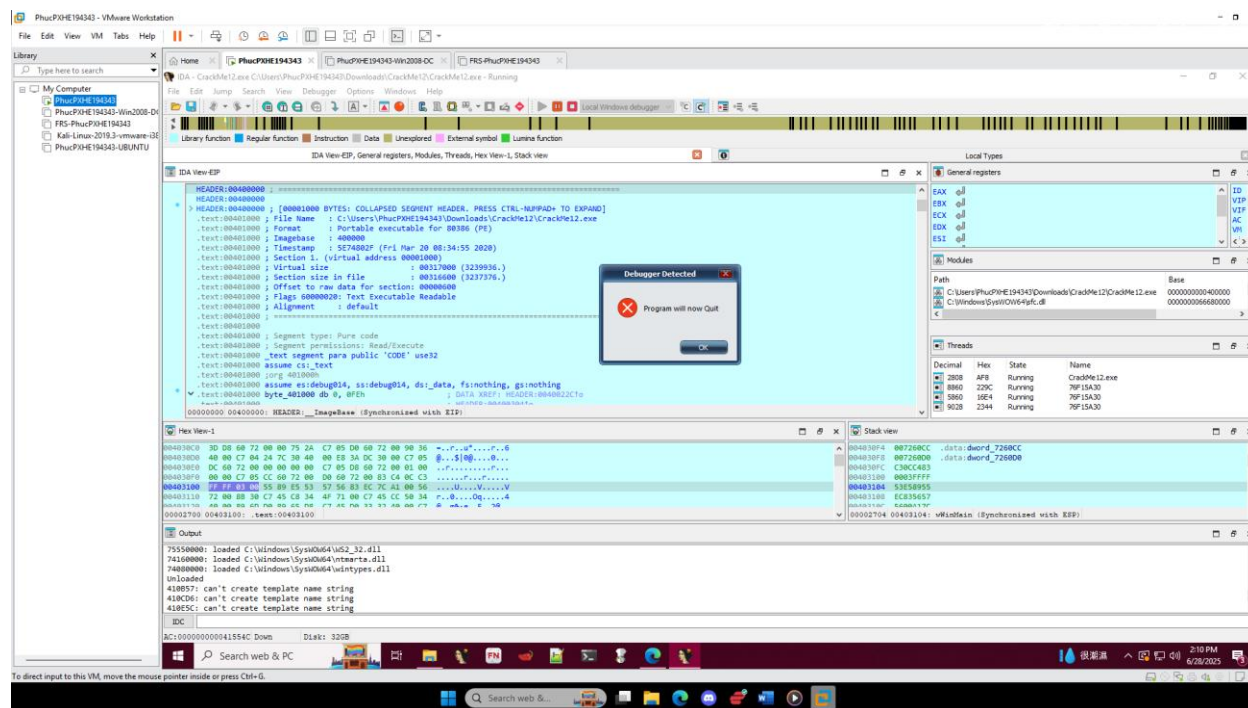
```
@echo off
setlocal
set Path=D:\VM PRO VJ\
set snapname=
set /p snapname=Enter the name for the snapshot:
for /f "skip=1 delims=" %i in ('varun list') do (
echo Deleting snapshot for %i
varun -T vs deleteSnapshot "%i" %snapname% msg.autoAnswer = TRUE
varun start "%i"
)
endlocal
set /p any=press any key ....
```

Ln 3, Col 23 314 characters Plain text 100% Windows (CRLF) UTF-8

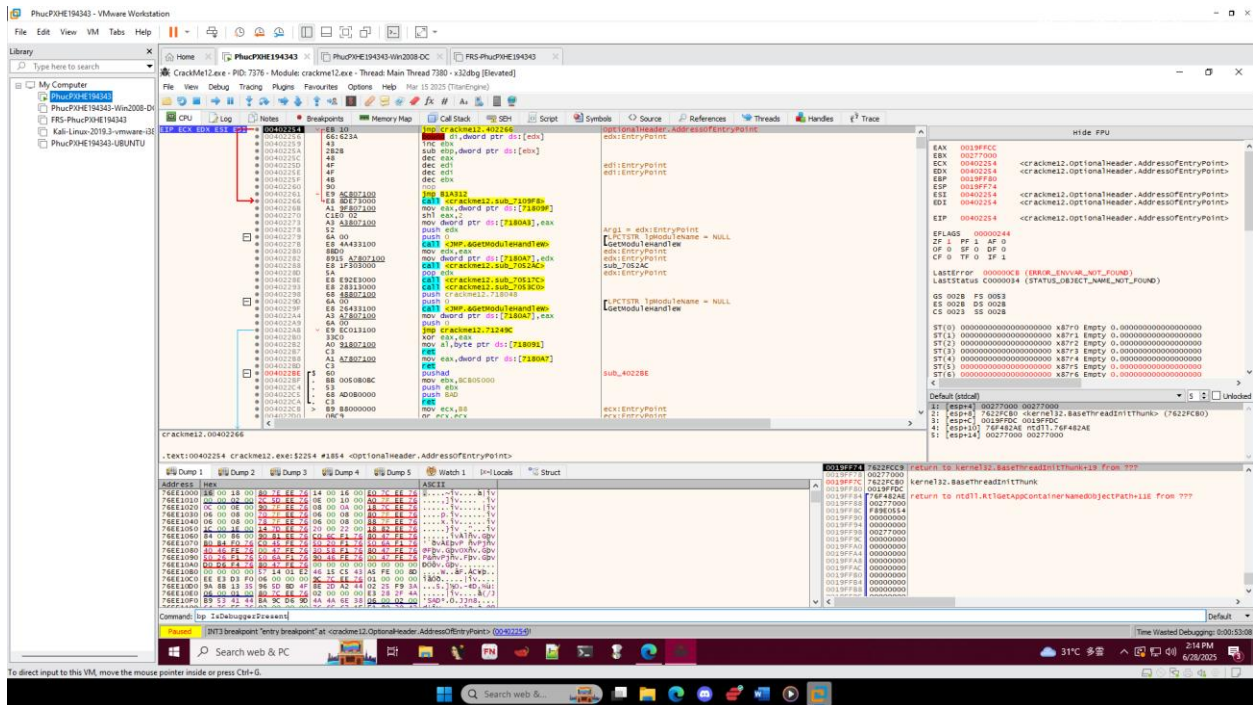


Crack Me 12

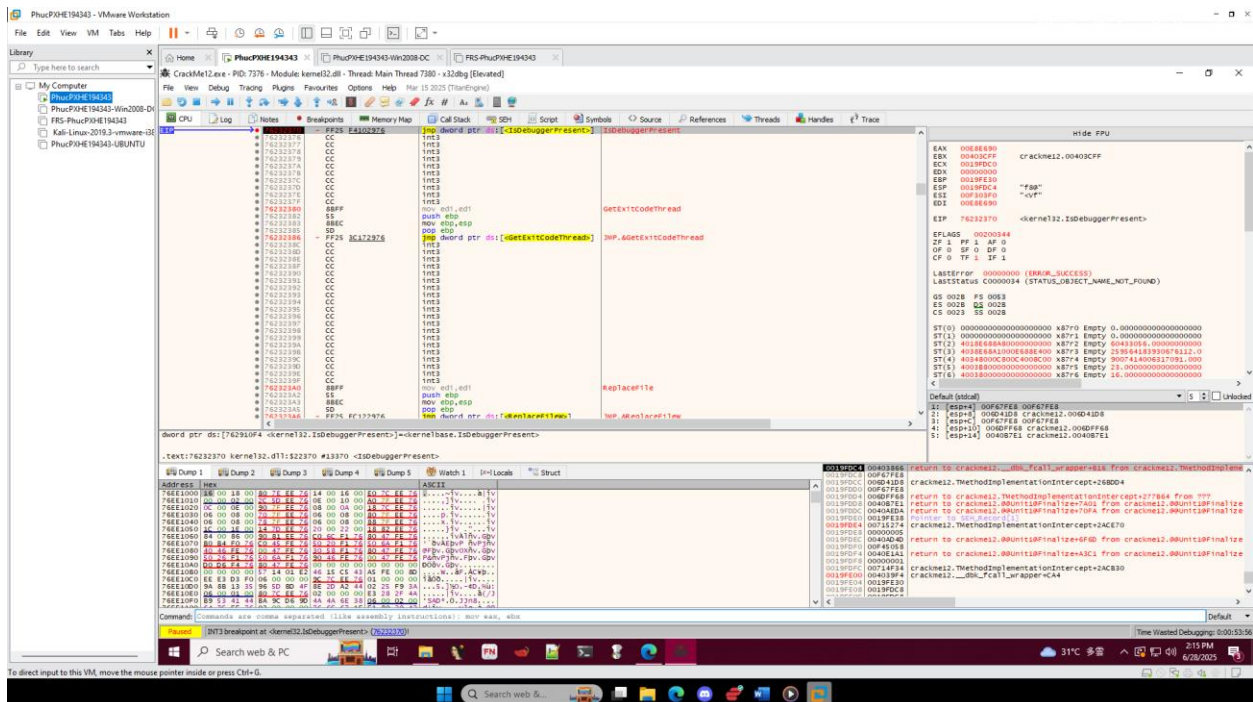
Step 1: Em bật phần mềm lên và kiểm thử xem là load phần mềm crack me vào 1 phần mềm có tính năng debugger và ngay lập tức phần mềm crack me phát hiện được debugger và dừng chương trình



Step 2: Em load crack me vào trong x32dbg để tiến hành patched chương trình, em dùng command “bp IsDebuggerPresent” để đặt 1 breakpoint ở 1 cái Windows API function gọi là IsDebuggerPresent.

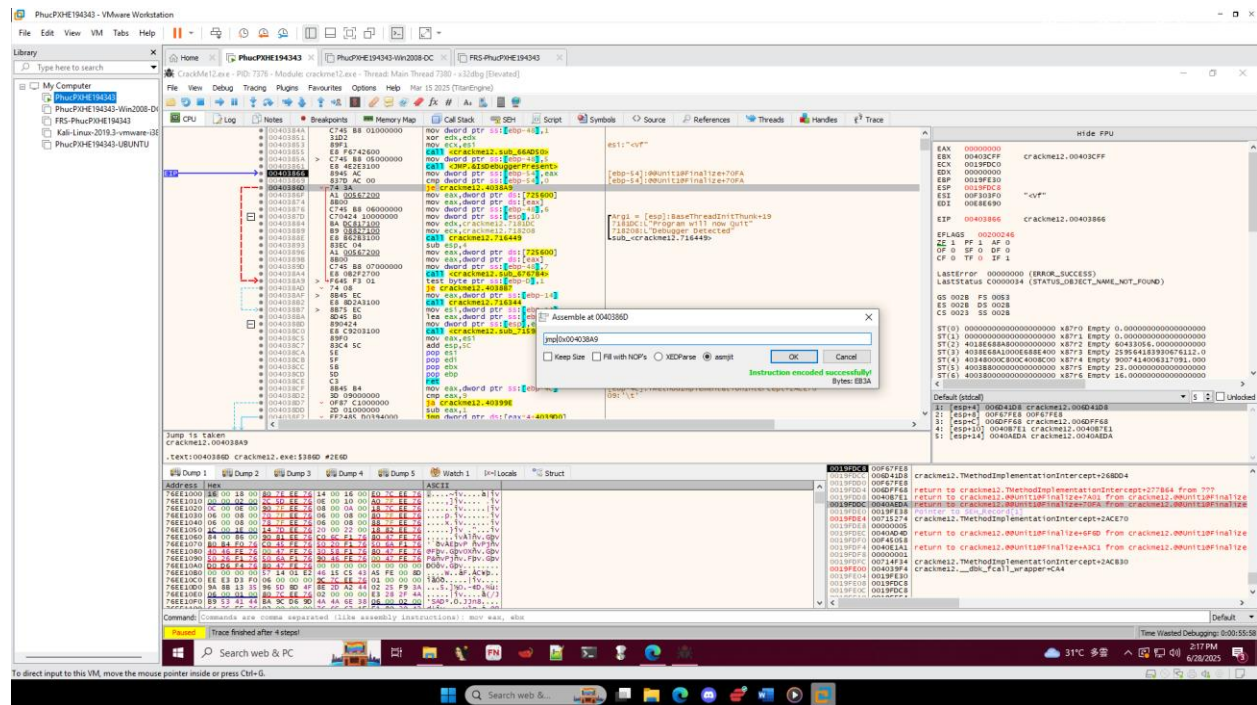


Step 3: Thì sau đấy chương trình nhảy đến 1 breakpoint nằm trong system module



Step 4: Quay trở về user module thì ta đã thấy function IsDebuggerPresent, ta đổi je về jmp ở câu lệnh ở dưới thì bằng cách đó chúng ta bypass được cái bad

message bằng cách vượt qua vòng loop, chúng ta tiến hành patched chương trình sau chỉnh sửa



Step 5: Chạy lại chương trình đã patched vào 1 phần mềm có tính năng debugger và chúng ta nhận được kết quả mà đề bài đã yêu cầu

