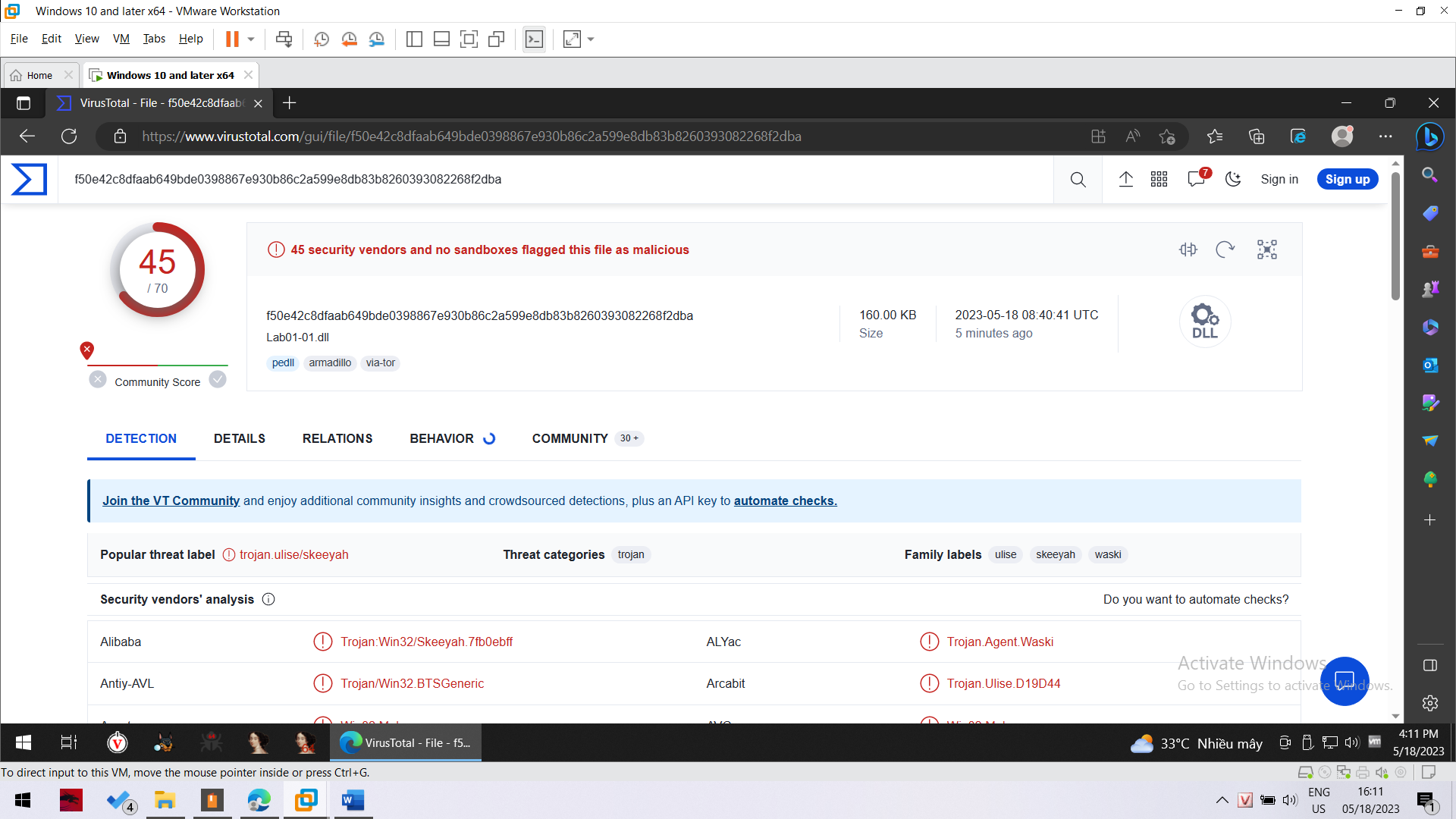
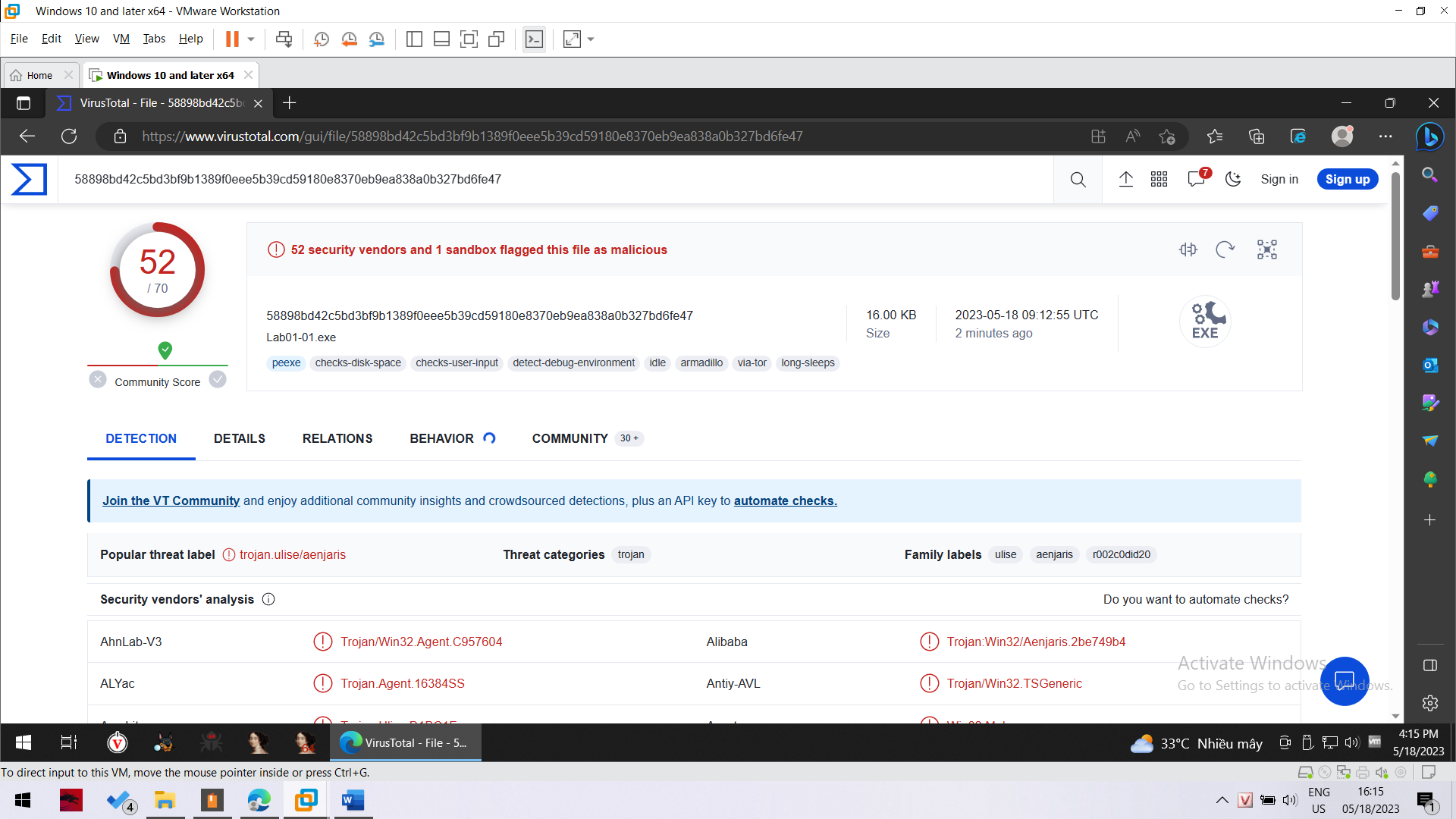
**Lab 2**

Trong bài lab này, chúng ta sẽ tiến hành phân tích trên các con exe và dll lab1.

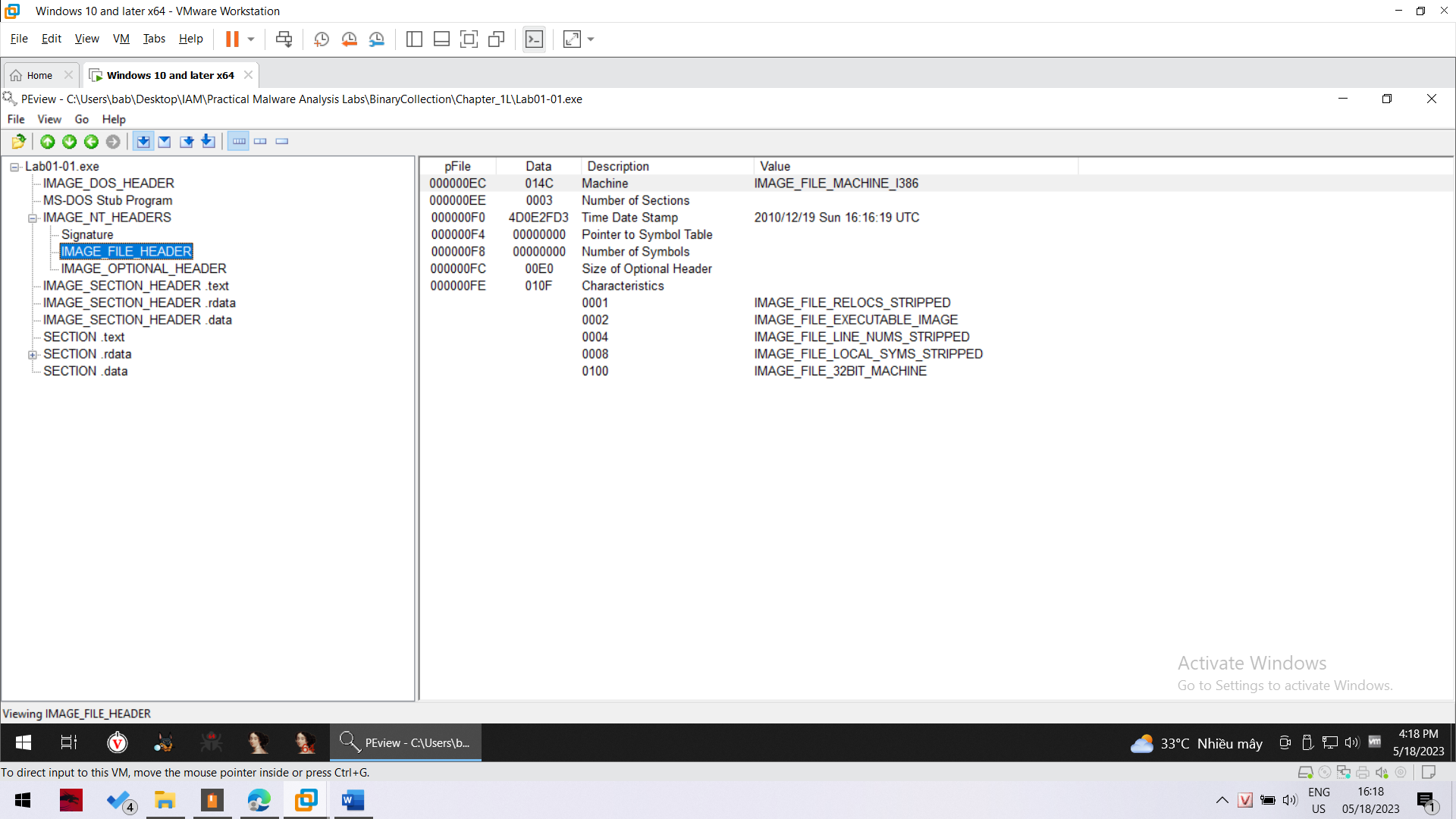


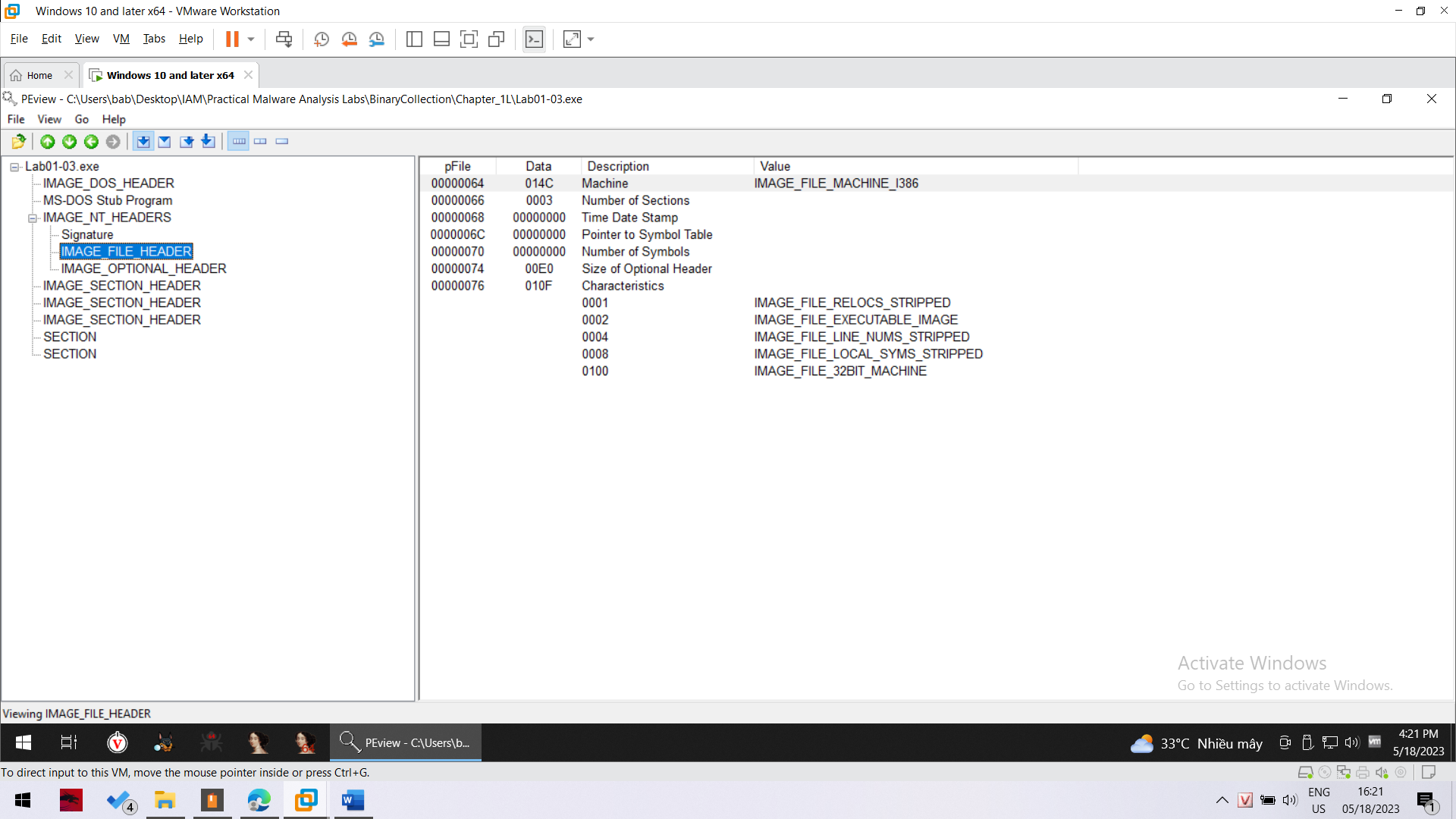
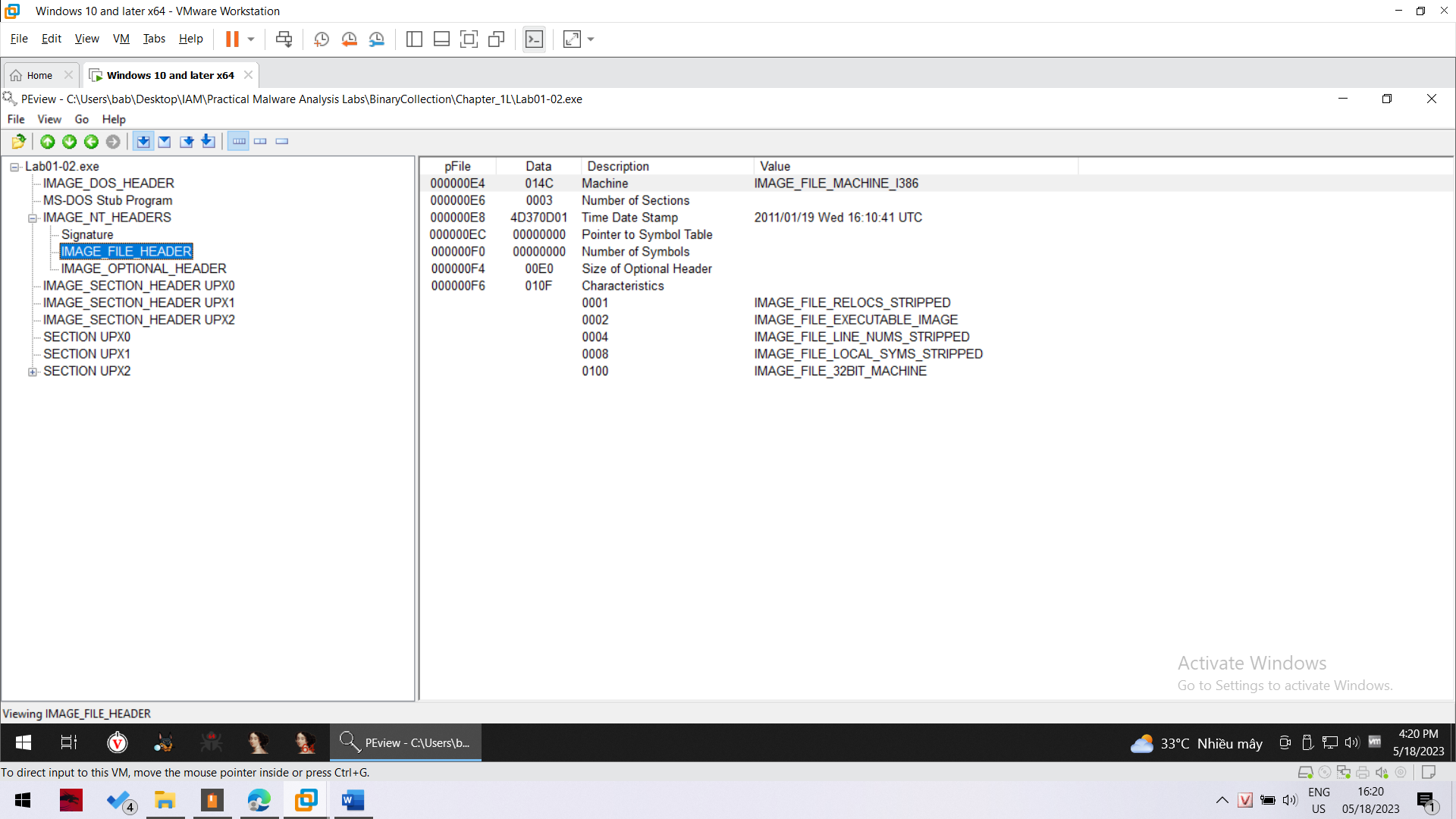
Ném file dll lên virus total, ta thấy sau quá trình phân tích đây là một con malware trojan.

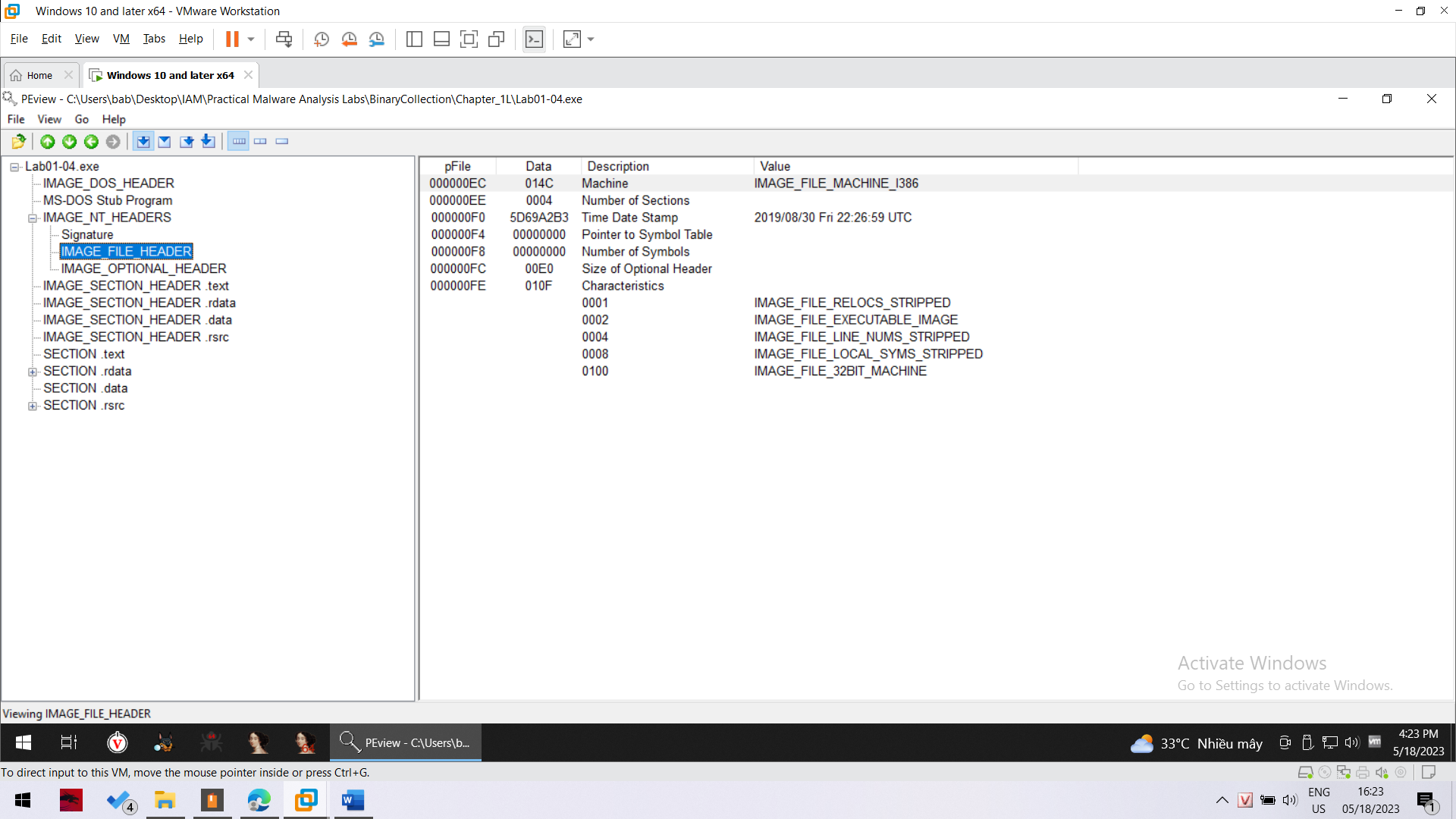


Tiếp tục up file lab01-01.exe lên ta nhận được kết quả cùng một loại malware với con malware phân tích từ file dll.

Tiếp theo ta sẽ xem “Time Date Stamp” từ mỗi file bằng PE view.

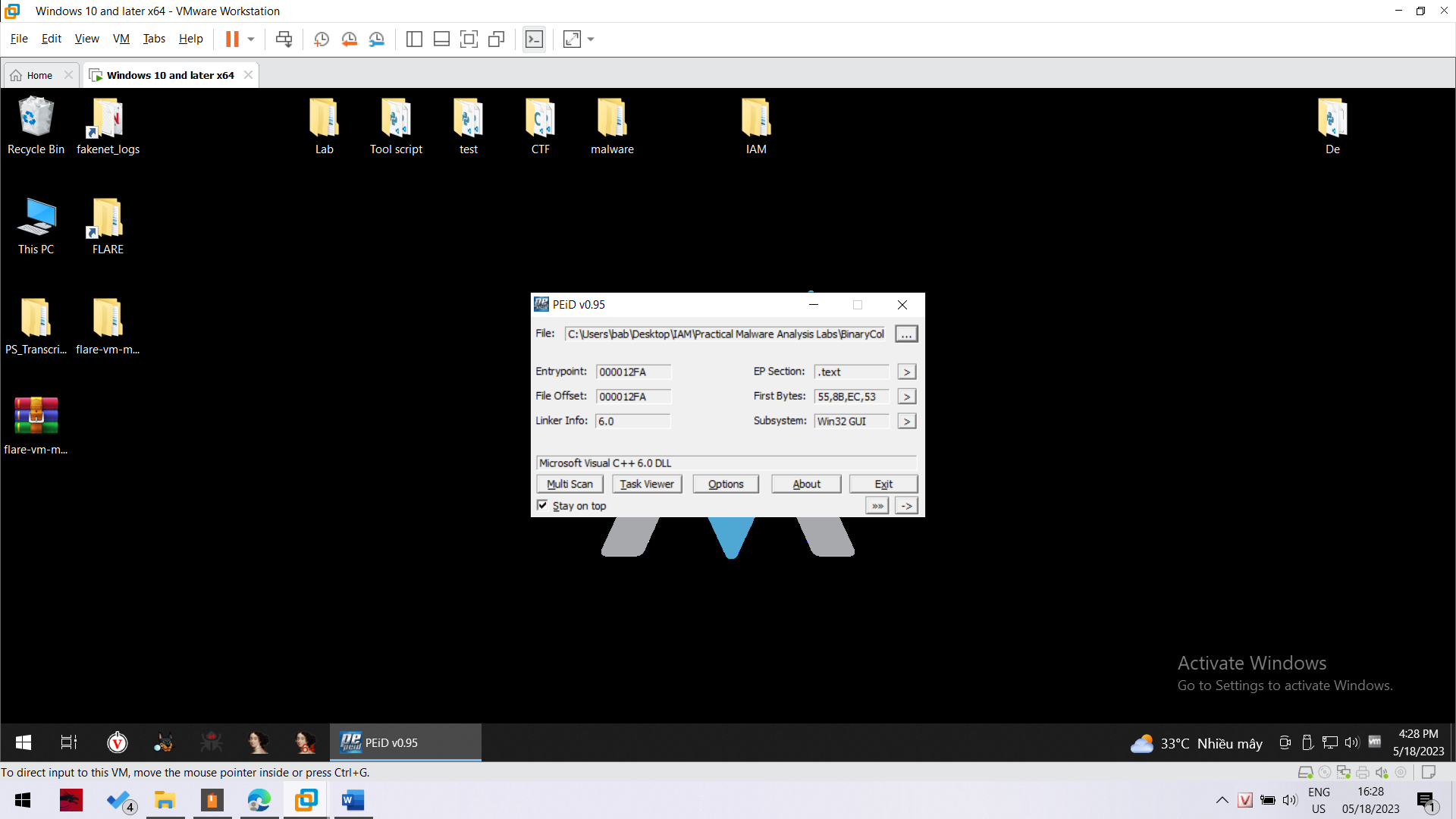


Ta có thể thấy một điểm lưu ý ở đây là ở file lab01-03.exe không có Time Date Stamp, rất có thể là tên hacker đã cố tình xóa đi metadata này trong section header.



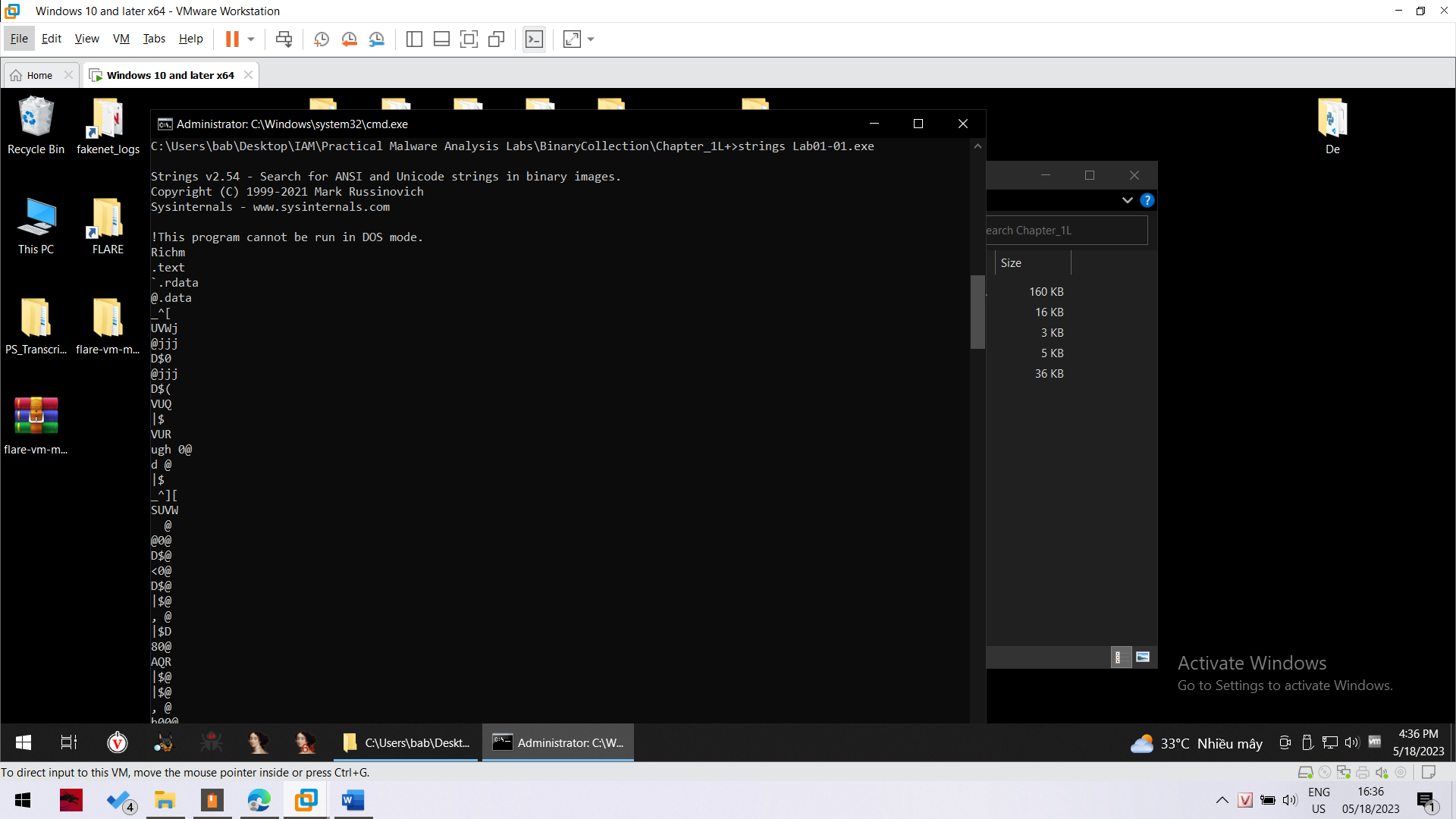
Tiếp theo, ta sẽ sử dụng PEiD để detect xem các file thực thi này được viết bằng ngôn ngữ gì, có bị pack hay không, và một số các thông tin hữu ích khác nữa.

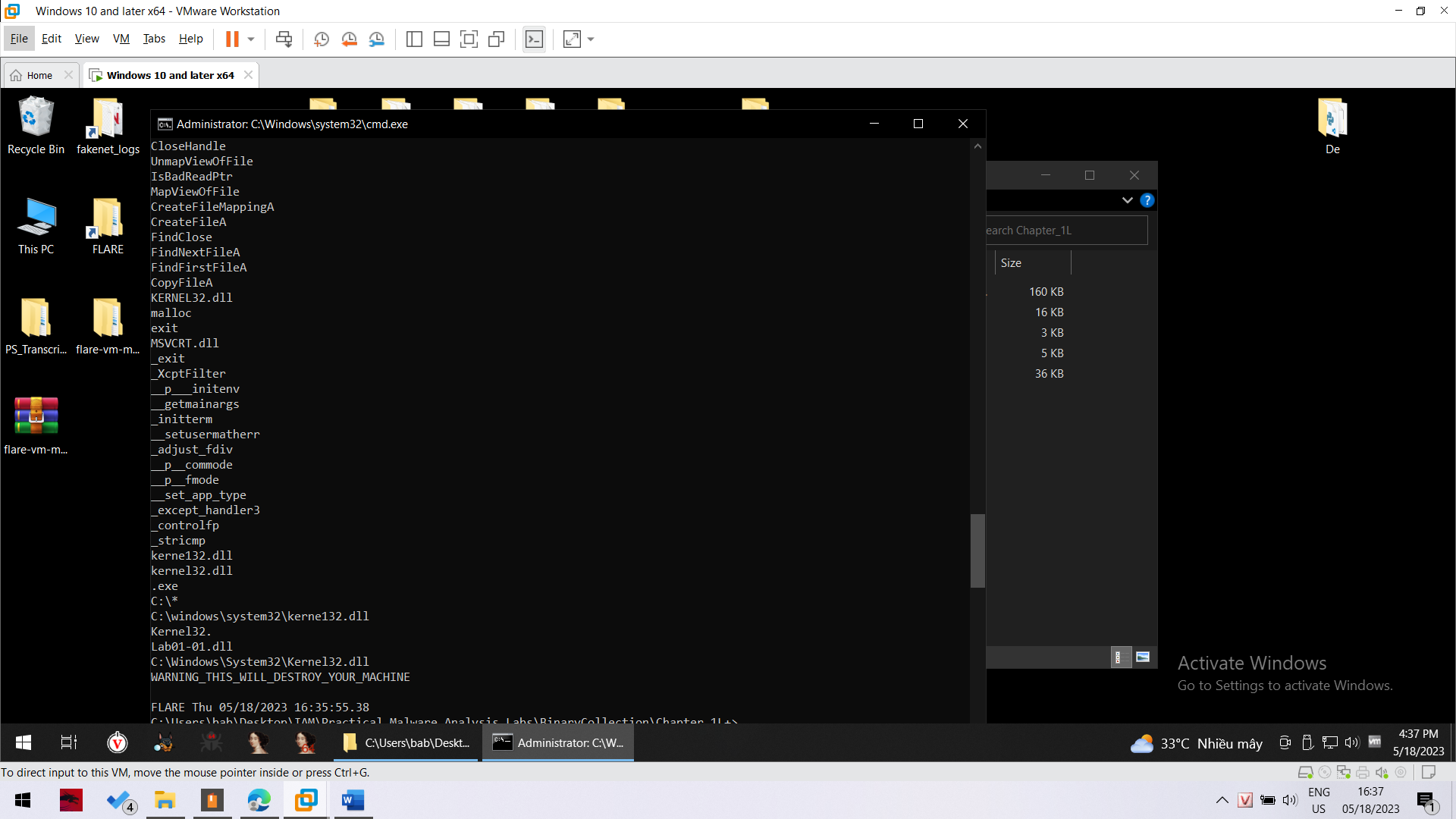
Load file dll vào PEiD



Ta thấy con malware này được viết bằng c++ có entry point là 0x12FA

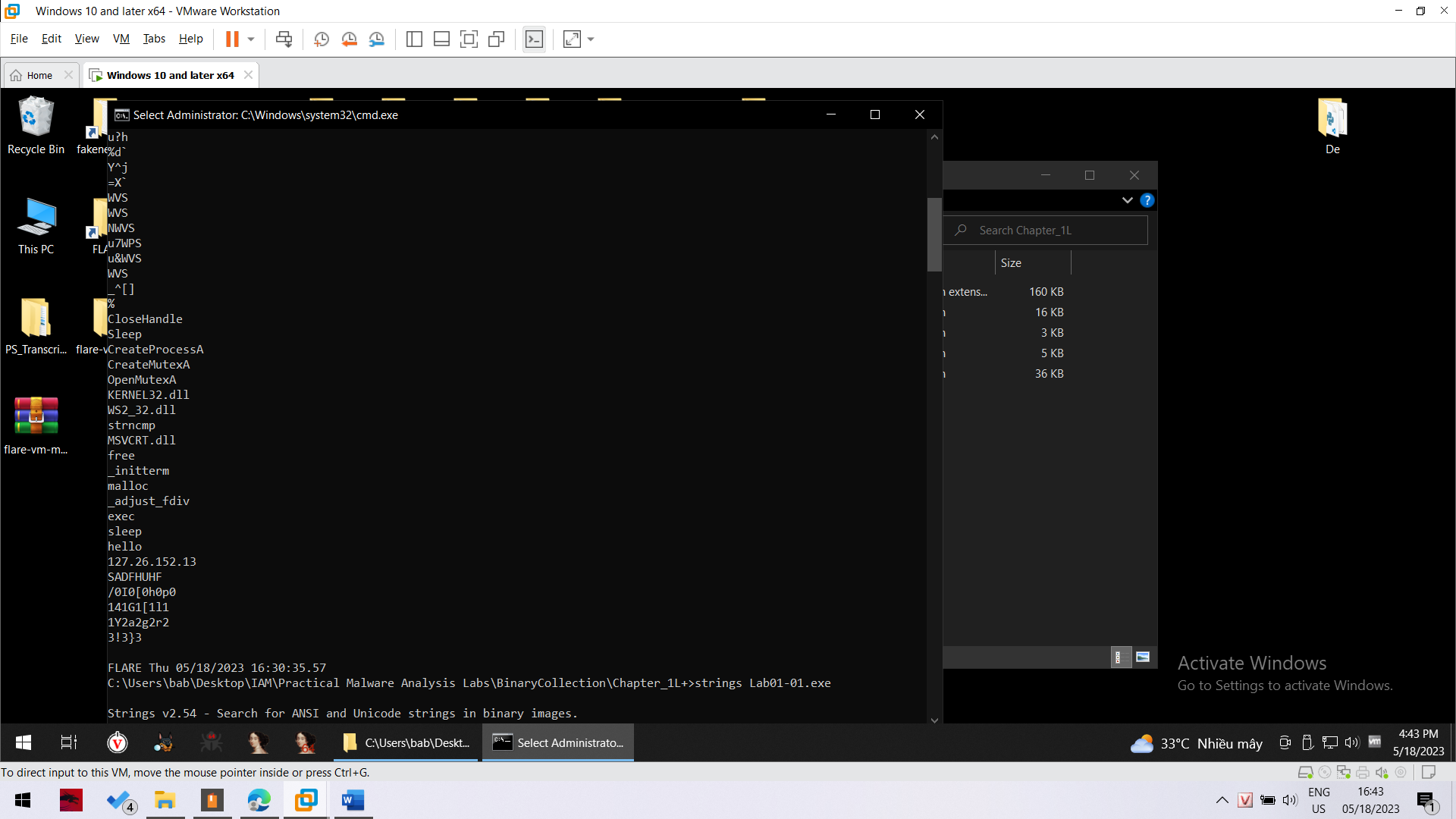
Ta thực hiện câu lệnh strings để xem một số strings bên trong file exe





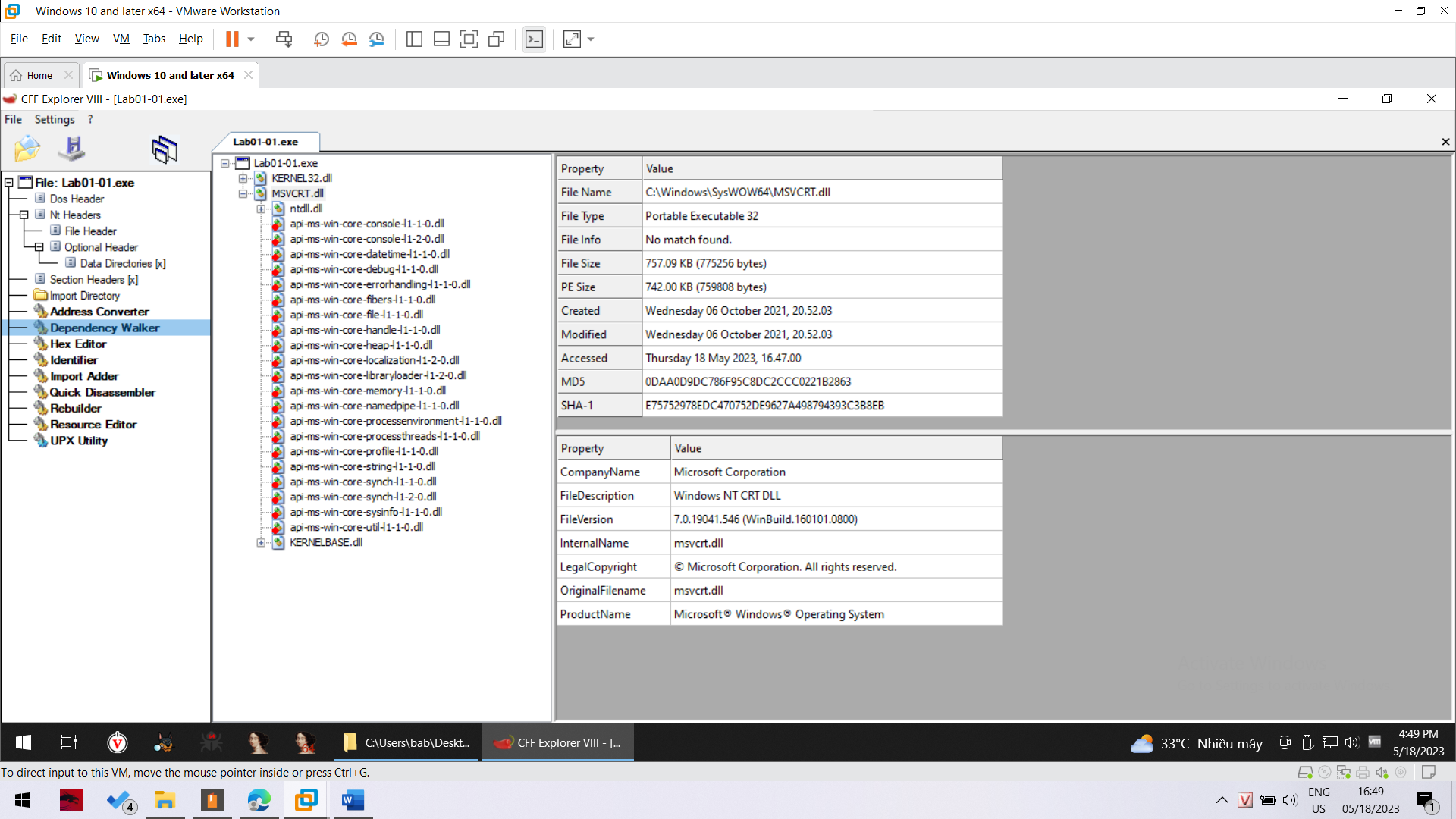
Ta thấy có một đoạn chuỗi trông khá là khả nghi WARNING\_THIS\_WILL\_DESTROY\_YOUR\_MACHINE. Và một số thư viện dll được con malware này load lên trong quá trình chạy như kernel32.dll, findFileA.

Ta tiếp tục dùng strings lên file dll.

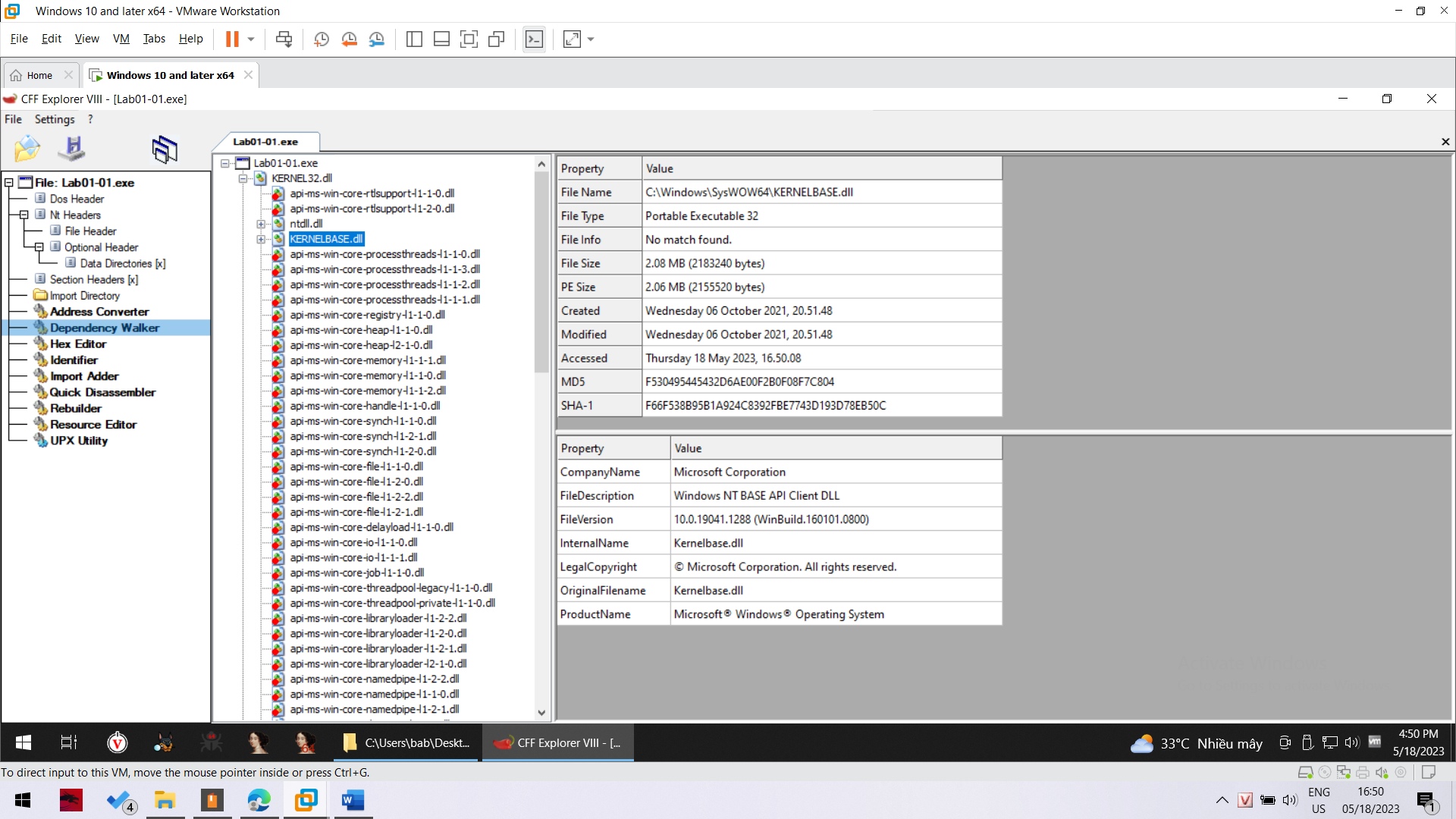


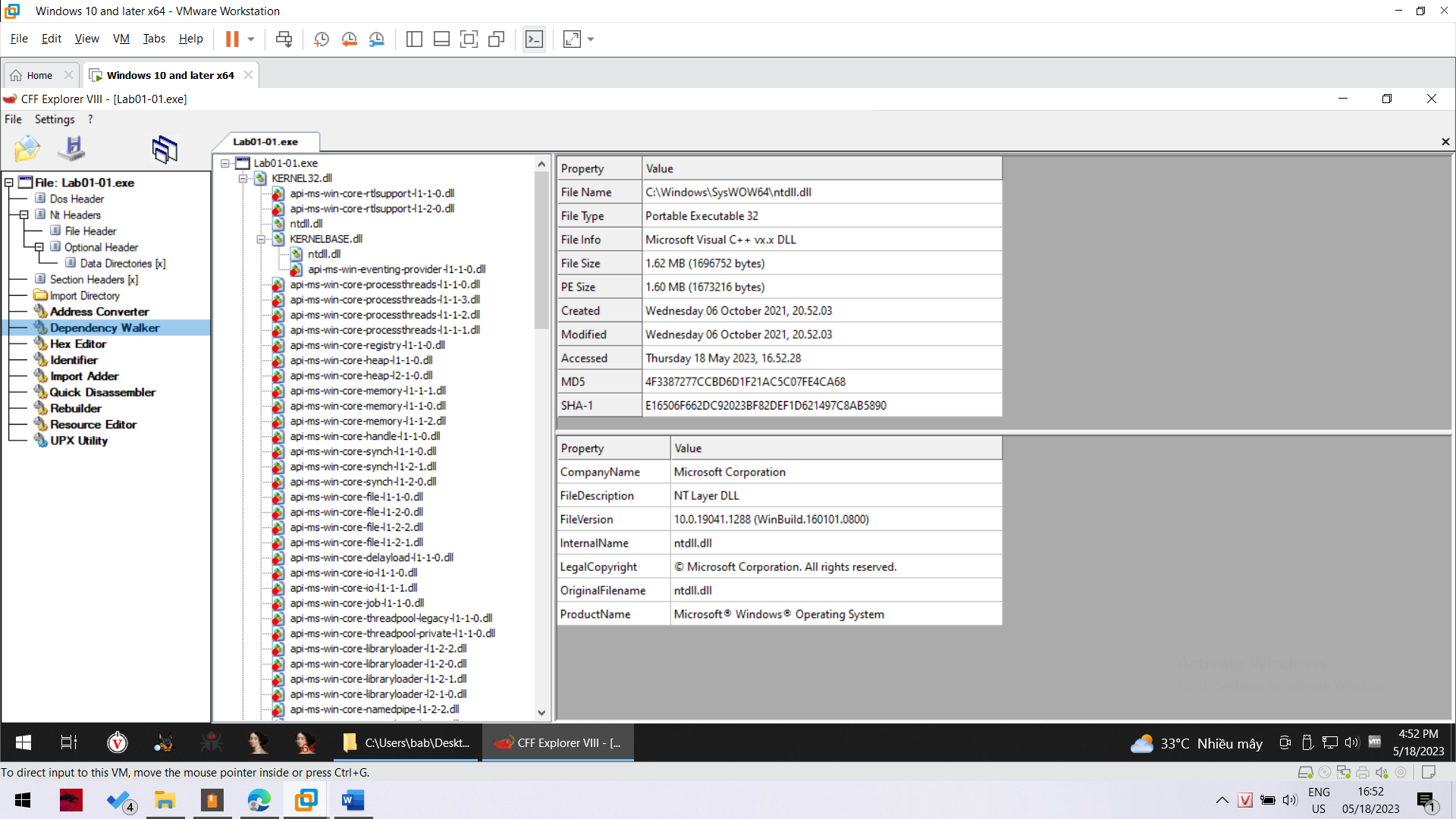
Ta phát hiện ra một địa chỉ IP khả nghi 127.26.152.13, rất có thể con malware này downfile mã độc khác từ ip này hoặc gửi file trên máy mình tới địa chỉ này, nhưng tất cả chỉ là suy đoán.

Mình sẽ sử dụng CFF explorer vì trong này có tích hợp cả dependency walker trong đây luôn.



Chọn xem kernel32.dll để xem coi con malware này xài những hàm gfi từ thư viện kernel32.dll.

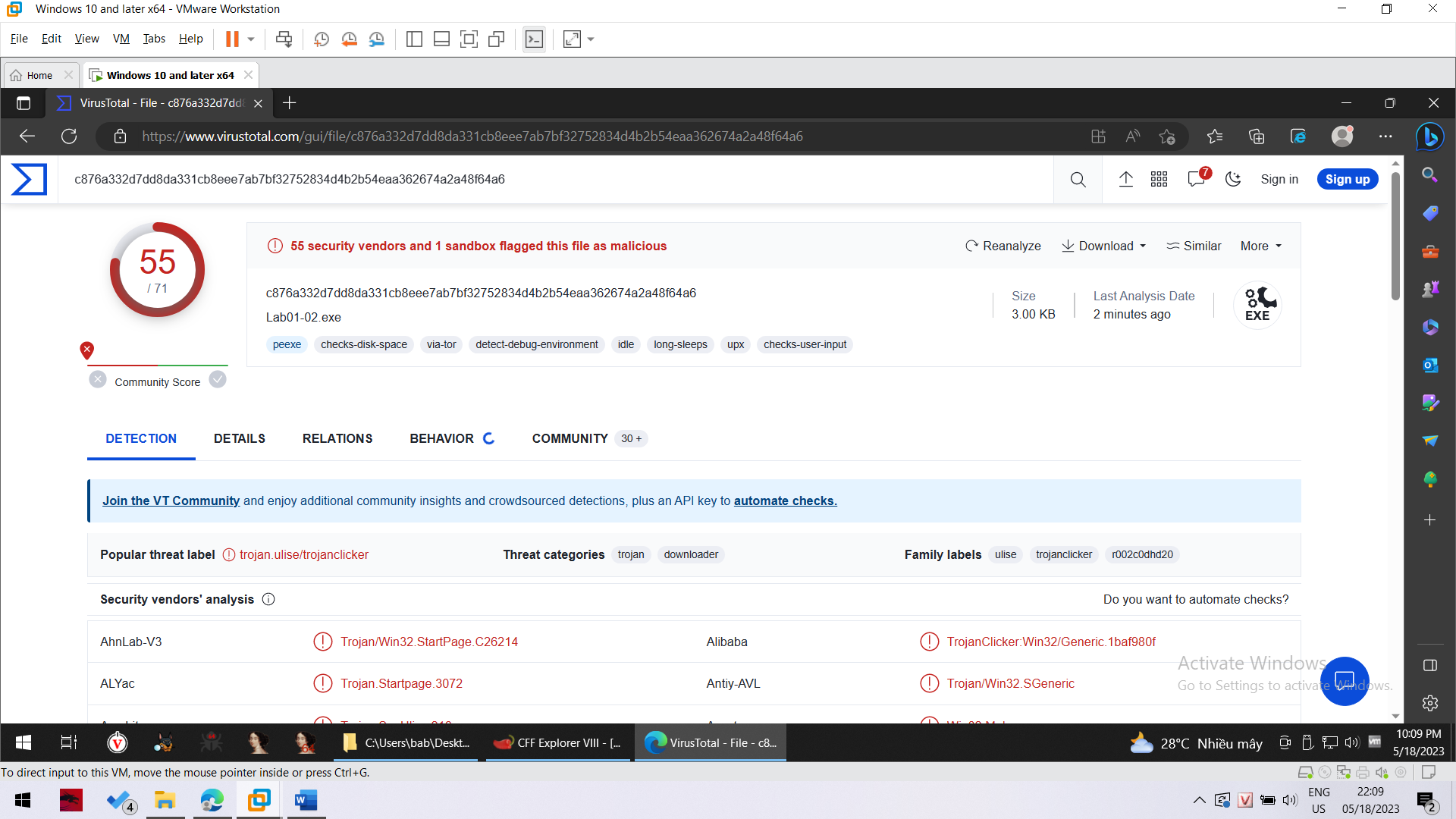




Ta có thể thấy những hàm API của windows được file malware này gọi thuộc kernel32.dll. Những hàm này nhìn là thấy đặc trưng của các con malware luôn vì nó có hàm tìm file, tạo file, hàm handle để handle error trong quá trình thực thi code.

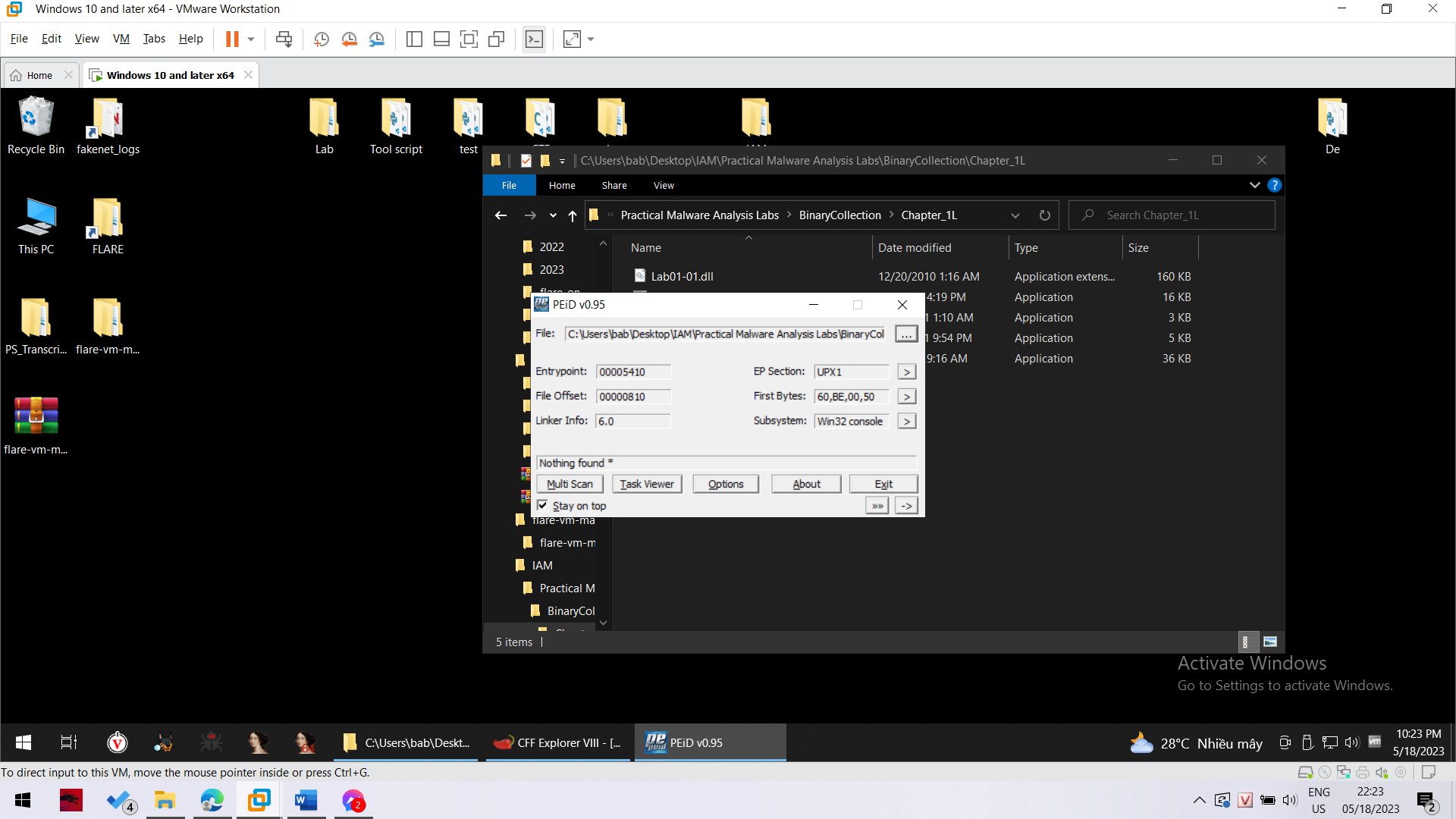
Bây giờ ta sẽ sang phần tiếp theo.

Ném con lab01-02.exe lên virus total.

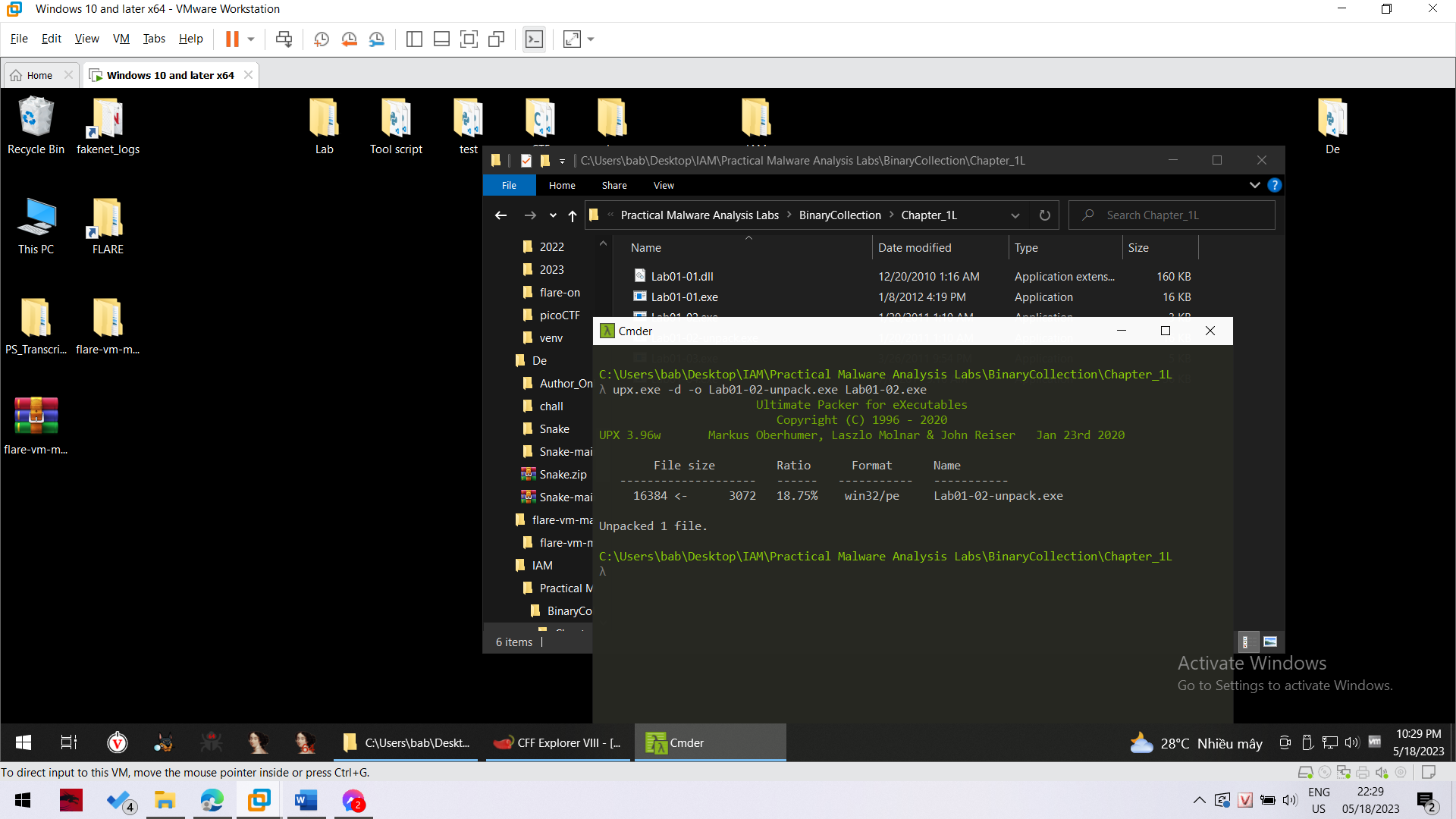


Kết quả ta nhận được vẫn là một con trojan nhưng là con trojan khác với con trước.

Sử dụng PEiD lên file exe này, ta thấy một thông tin khá thú vị.

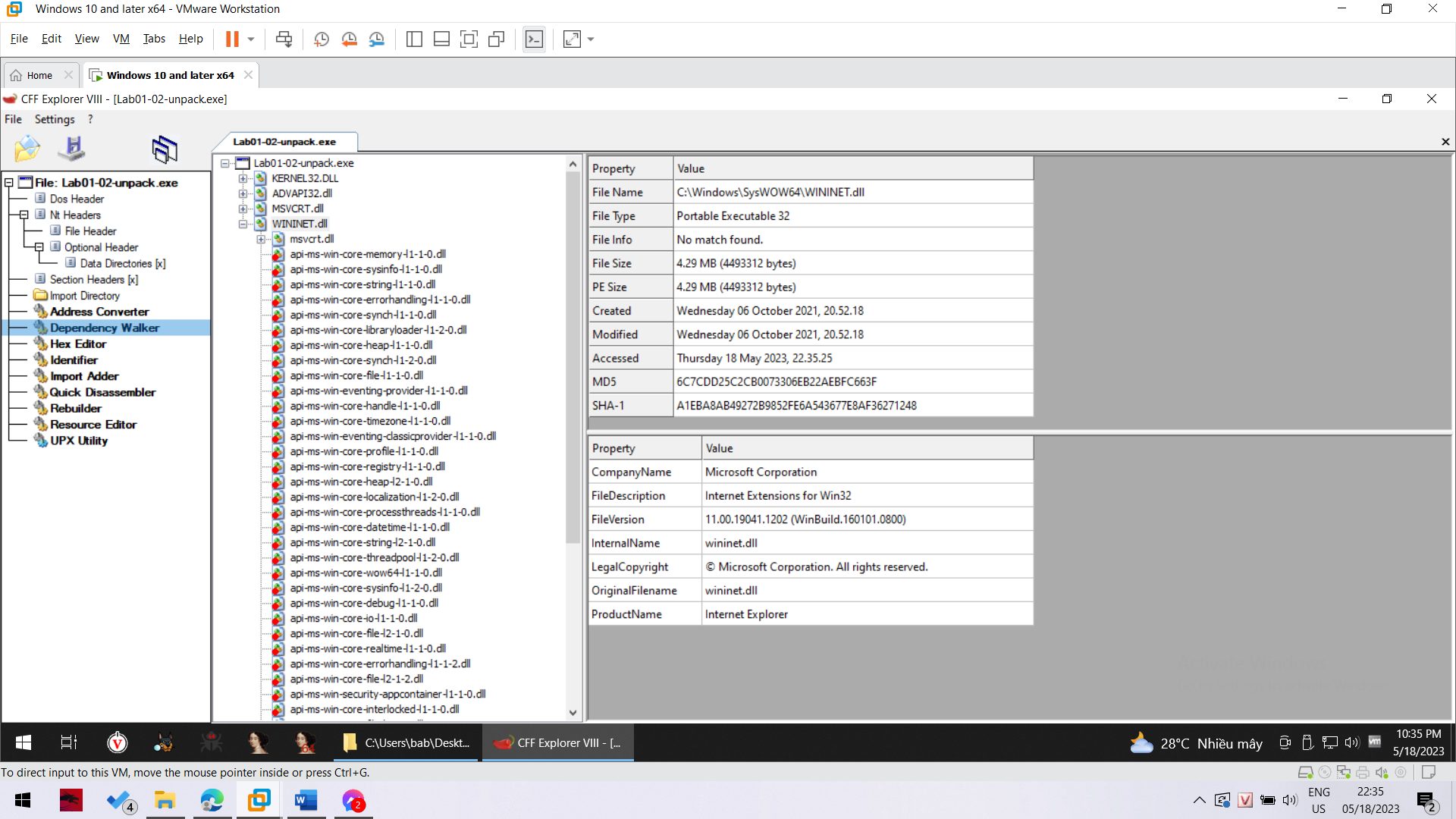
PEiD không tìm ra được file exe này được viết bằng ngôn ngữ gì nhưng nó đã tìm ra được file exe này đã bị pack bởi UPX. Một trình packer cơ bản rất được hay dùng bởi các mã độc để chống dịch ngược mã nguồn.

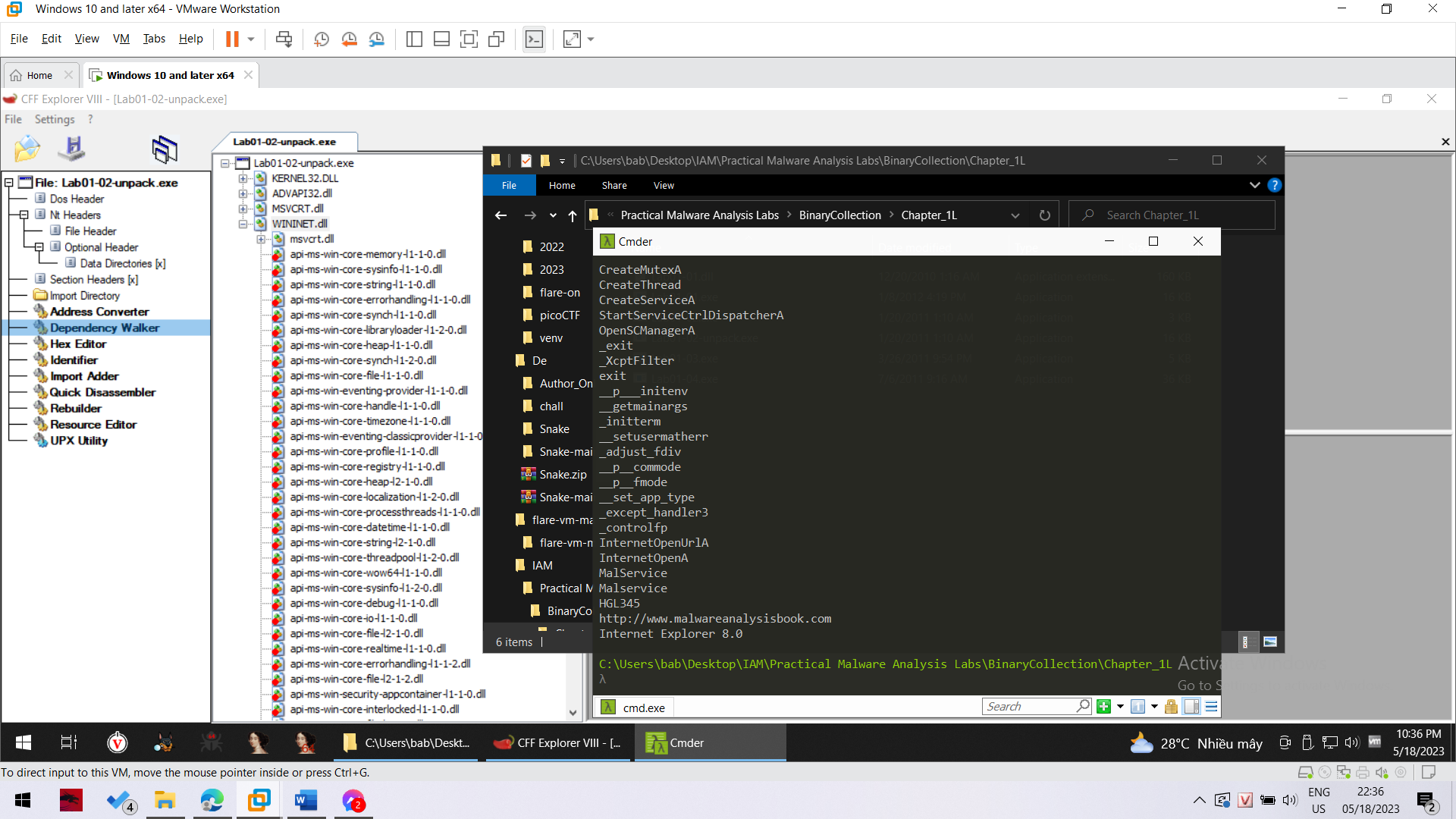
Vậy bây giờ làm thế nào để ta có thể unpack nó lại mã nguồn ban đầu. UPX cũng chính là lời giải vì công cụ này có command phục vụ luôn cho quá trình unpack.



Vậy là quá trình unpack đã hoàn tất. Ta dùng PEiD để kiểm tra lại.

PEiD đã nhận ra được file sau khi unpack được viết bằng C++

Load file vào cff explorer ta có các thông tin như trên ở phần dependency walker.

Dùng câu lệnh strings ta thấy được có một cái domain ở đây.