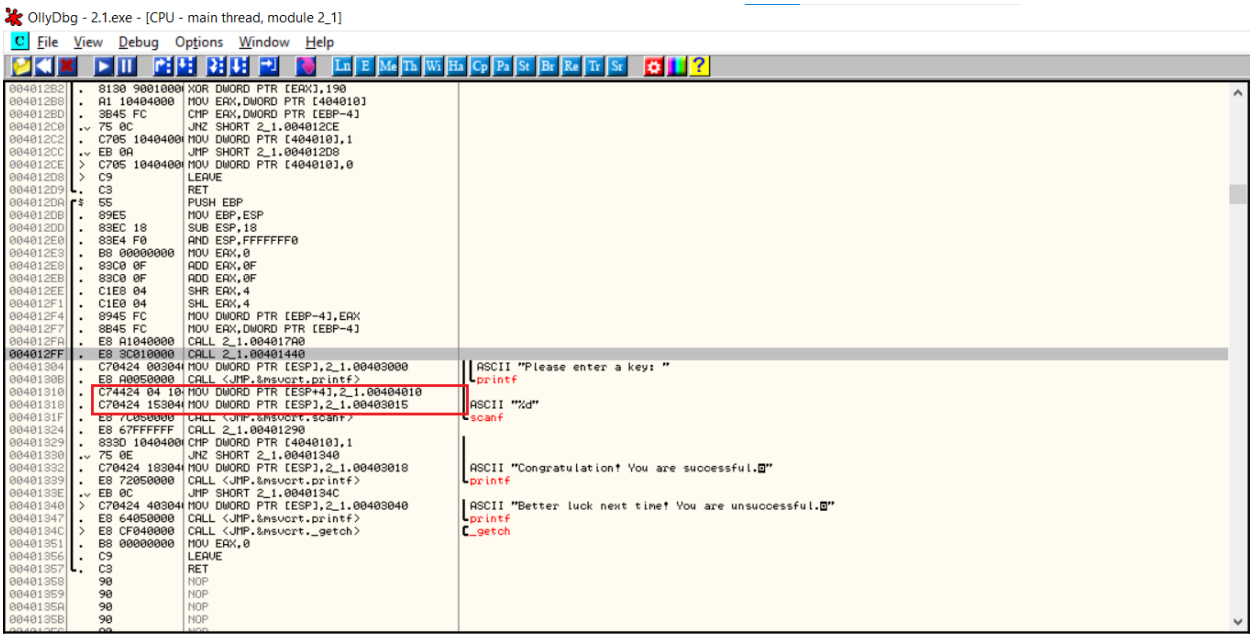
Bài 2.1:

B1: Mở file 2.1.exe bằng phần mềm OllyDbg

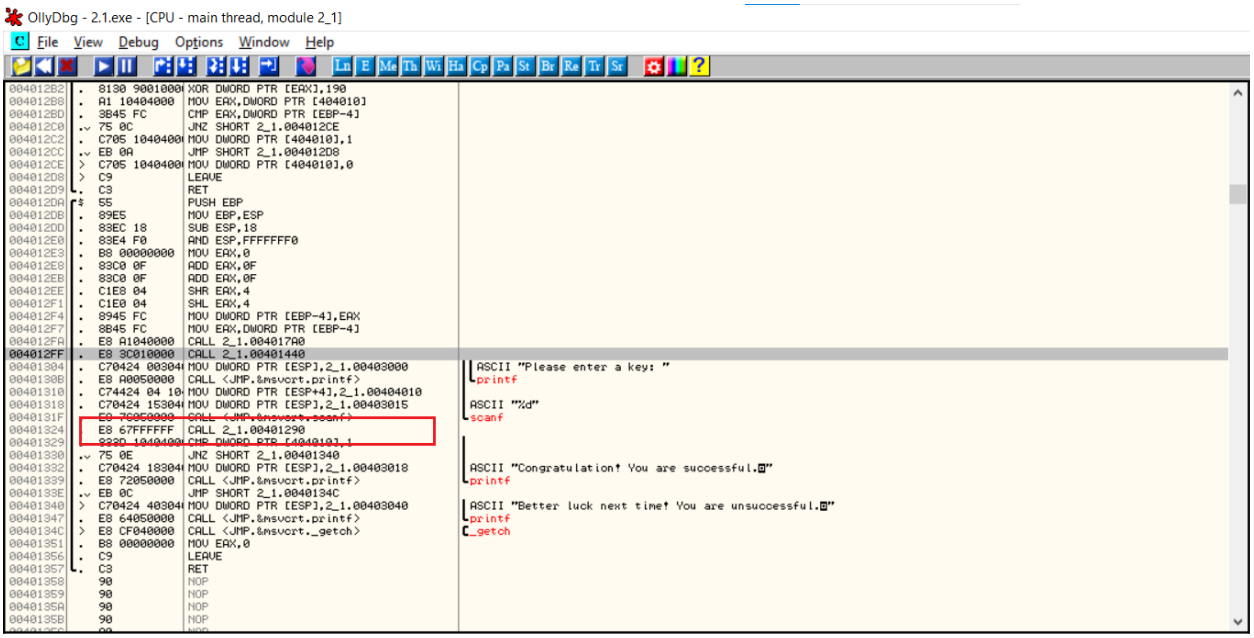


B2: Đọc code và tìm ra BadBoy và GoodBoy:

-Ta sẽ tìm được dữ liệu đầu vào của hàm scanf ở dòng 401310 và 401318 -> 404010 sẽ là nơi lưu trữ dữ liệu nhập vào.

Tìm ra được: BadBoy 401340 Goodboy 401332

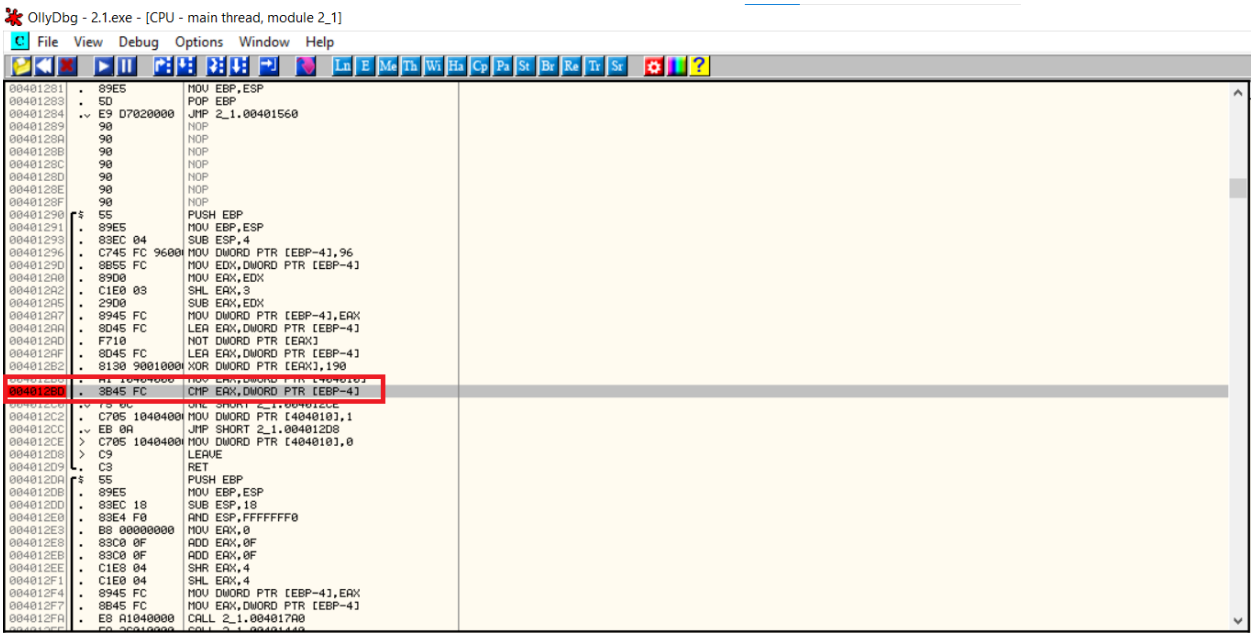
B3: T thấy sau khi nhập liệu xong chương trình gọi lệnh nhảy tới vị trí 401290 (*1)



-Sau lệnh nhảy xuất hiện lệnh CMP so sánh kết quả của ô nhớ 404010 với 1 để nhảy tới Goodboy và badBoy nên chương trình sẽ thực thi

So sánh Key với kết quả trong lệnh Call (*1)

B4: Trong khối lệnh ta phát hiện lệnh CMP EAX,DWORD OTR [EBP - 4] khả nghi nên ta sẽ đặt breakpoint tai câu lệnh này

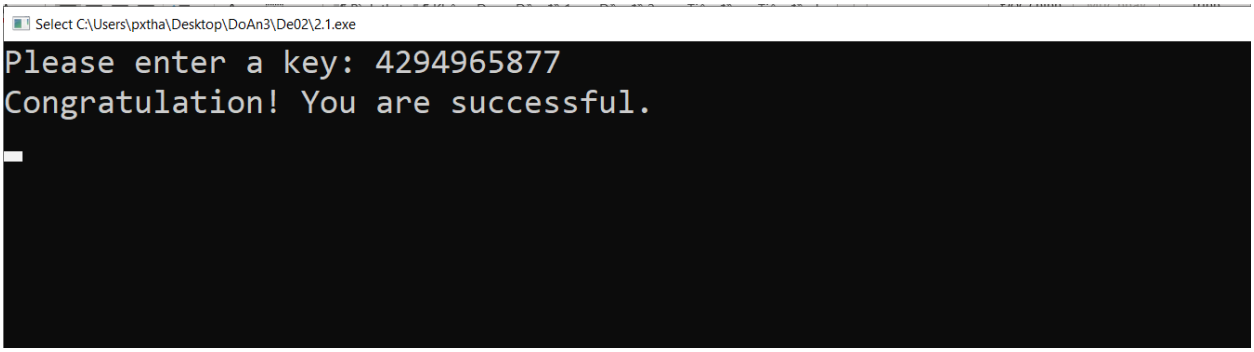


B5: Thực thi chương trình và Nhập Key = 15

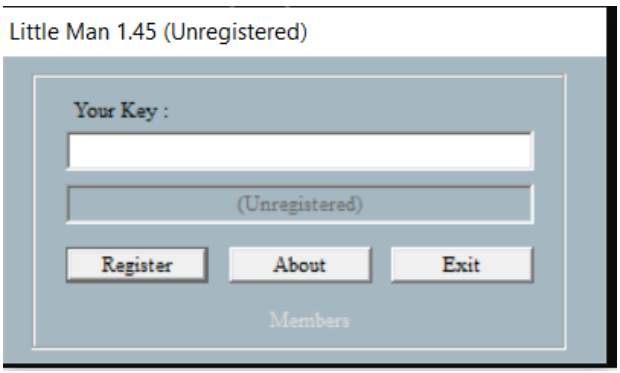
Ta thấy thanh ghi EAX = F_{16} =15_{10} là Key ta nhập vào
Vậy EBP – 4=60FEF4 sẽ lưu password

B6:Vào ô nhớ ta lấy kết quả: $FFFFFA75_{16}$ = 4294965877_{10} là kết quả

Kết luận: Key là 4294965877



Bài 2.2:

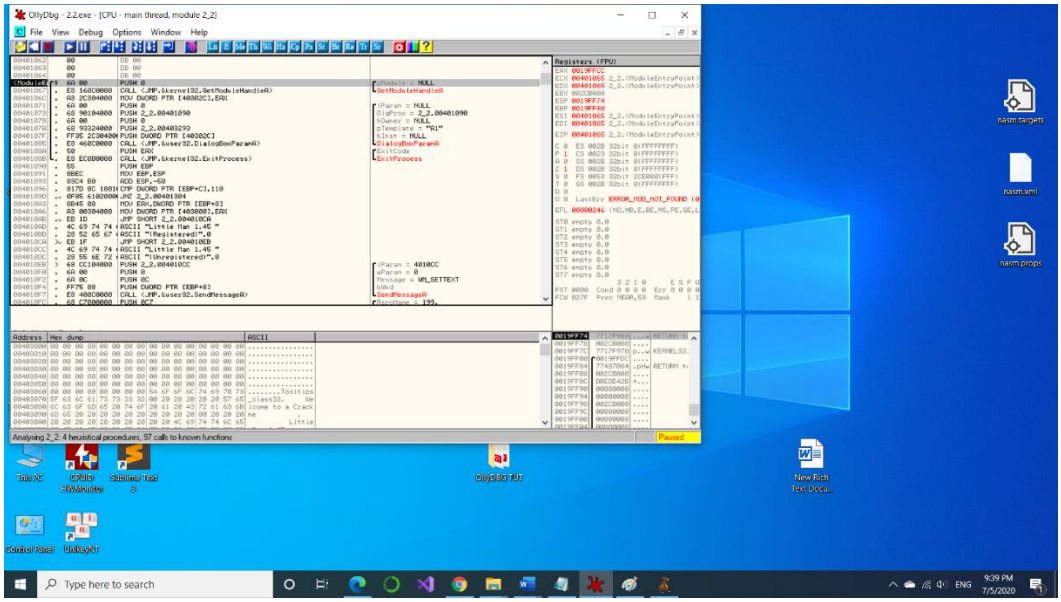


Phân tích:

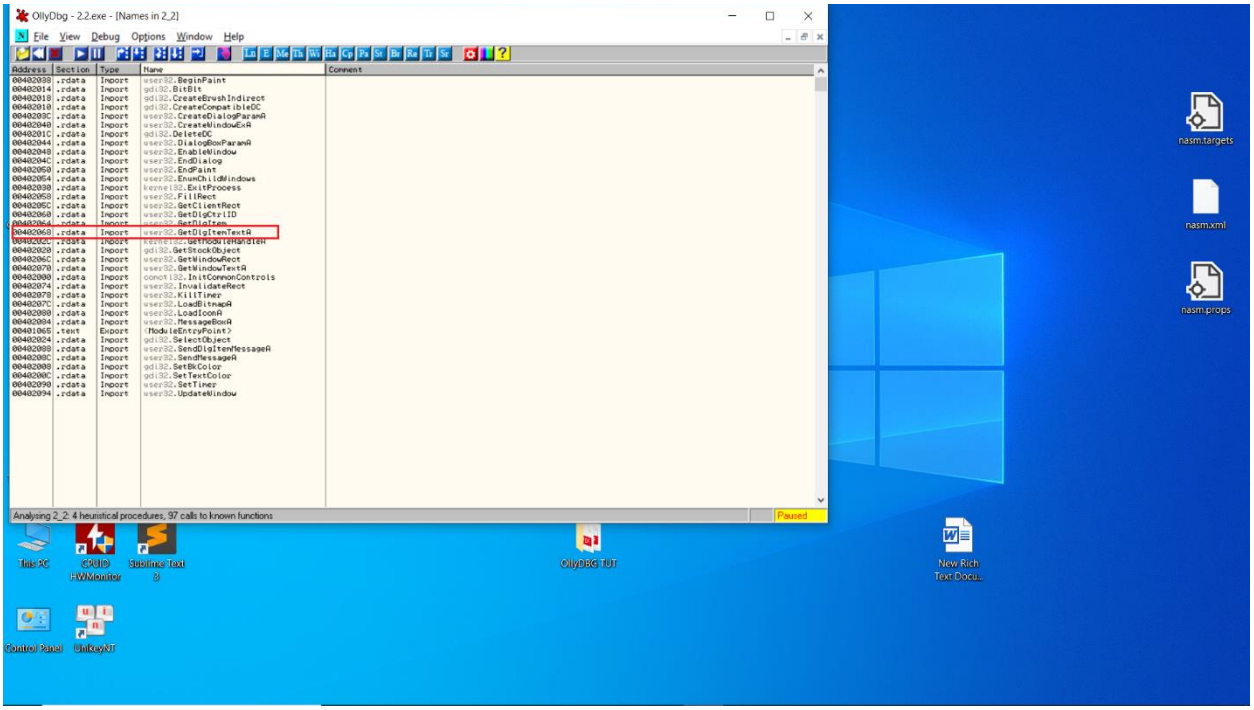
Sử dụng các hàm hỗ trợ Win32 của Microsoft.

Badboy được viết sẵn trong box.

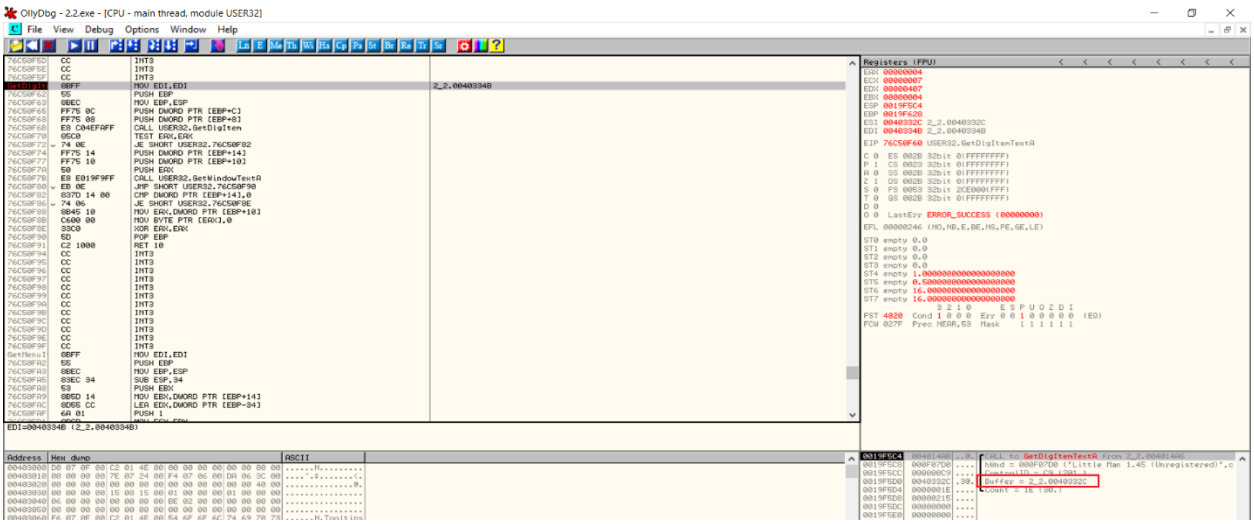
Chạy OllDBG:



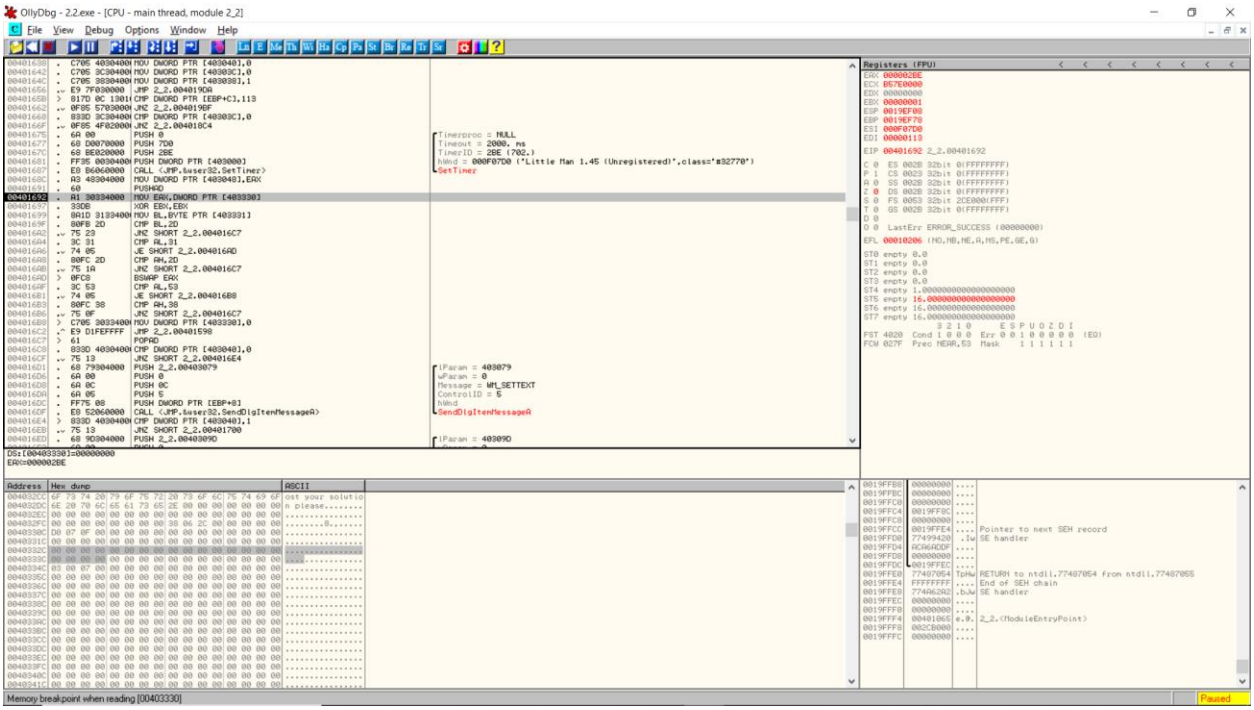
Bấm tổ hợp phím Ctrl+N để hiện ra bảng các hàm APIs được sử dụng.



- Ta chú ý hàm GetDlgItemTextA là hàm sẽ lấy dữ liệu dạng text (mô tả)
- Sử dụng Breakpoint vào hàm này: sau khi nhập liệu thì chương trình sẽ dừng lai tại BP mà chúng ta đã đặt.
  Nhìn vào cửa sổ stack thì ta thấy các tham số truyền vào của hàm.và buffer của nó 40332C



- Ta đặt MemoryBreakpoint vào ô nhớ này để xem những lệnh nào sẽ tác động đến:
- Sau khi chạy F9 thì chương trình dừng lại ở câu lệnh 401692:

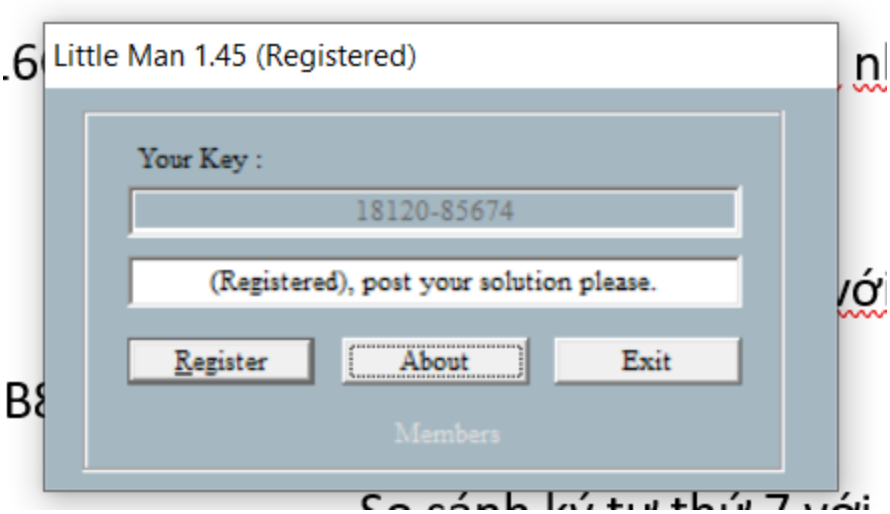


- Ta xem xét ý nghĩa của đoạn lệnh này:

00401692 MOV EAX,DWORD PTR [403330]	EAX = 4 ký tự cuối theo thứ tự từ phải qua trái
00401697 XOR EBX, EBX	Xóa dữ liệu trong EBX
00401699 MOV BL,BYTE PTR [403331]	BL = ký tự thứ 6 từ đầu đến cuối
0040169F CMP BL,2D	So sánh BL với '-'
004016A2 JNZ SHORT 2_2.004016C7	Nếu không bằng thì nhảy
004016AD BSWAP EAX	Đảo ngược EAX
004016AF CMP AL, 53h	So sánh ký tự thứ 8 với 'S'
004016B1 JE SHORT 2_2.004016B8	Đúng thì nhảy
004016B3 CMP AH,38	So sánh ký tự thứ 7 với '8'

➢ Vậy ta sẽ có 2 dạng Key(với * là ký tự tùy ý):

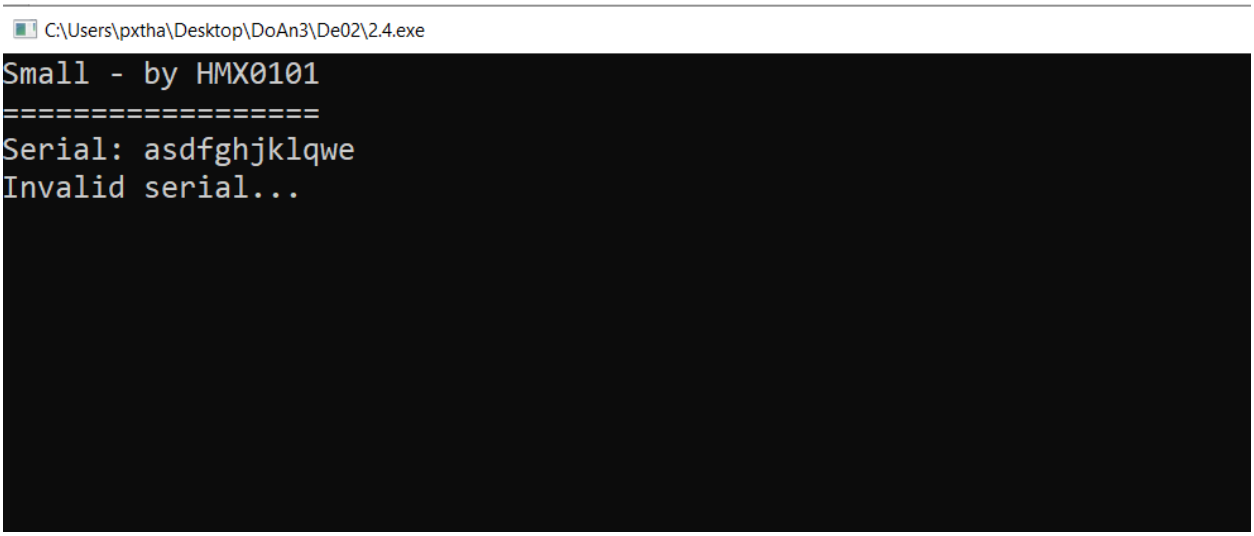
- *****-*S***
- *****-8****



Bài 2.4:

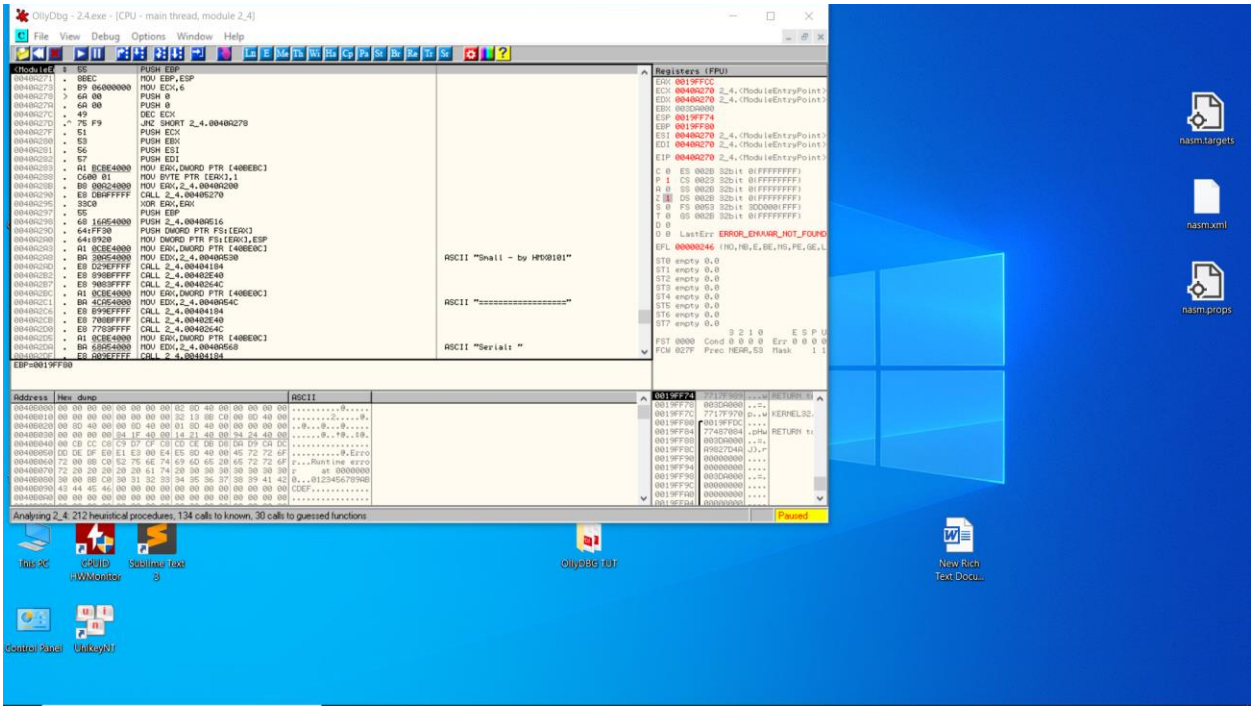
Chạy thử phân tích:

Yêu cầu bắt buộc serial phải có 12 ký tự.

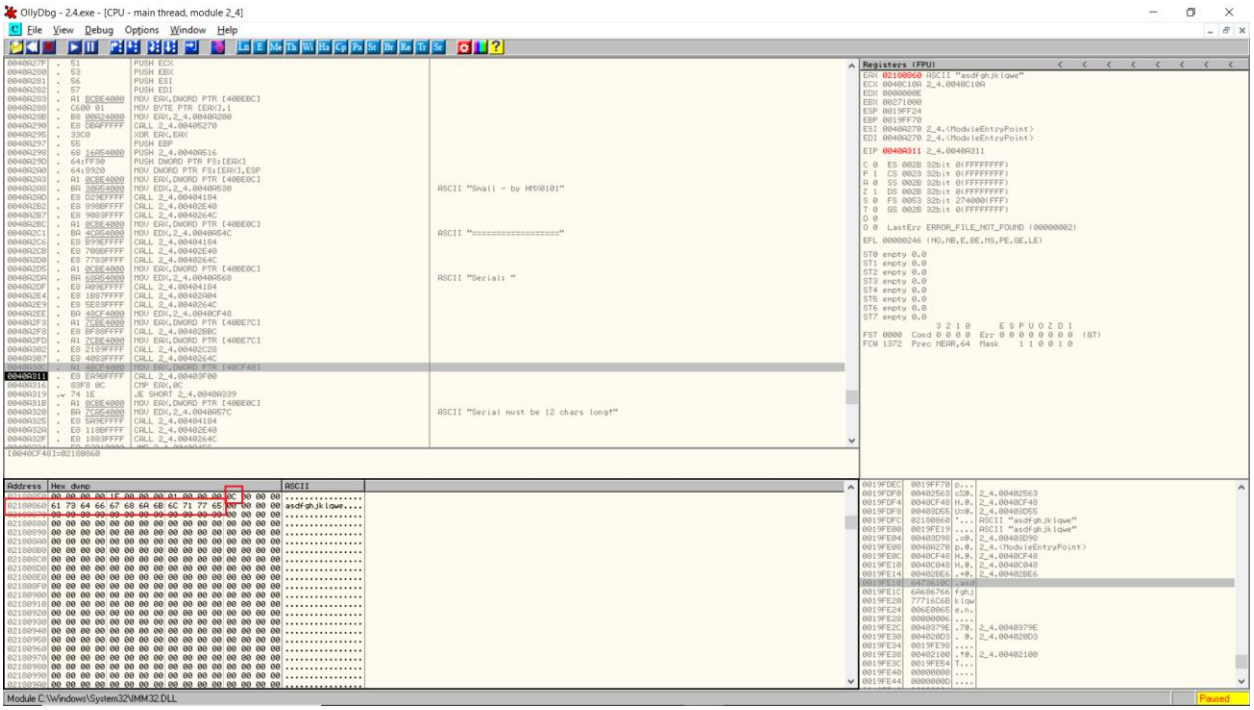


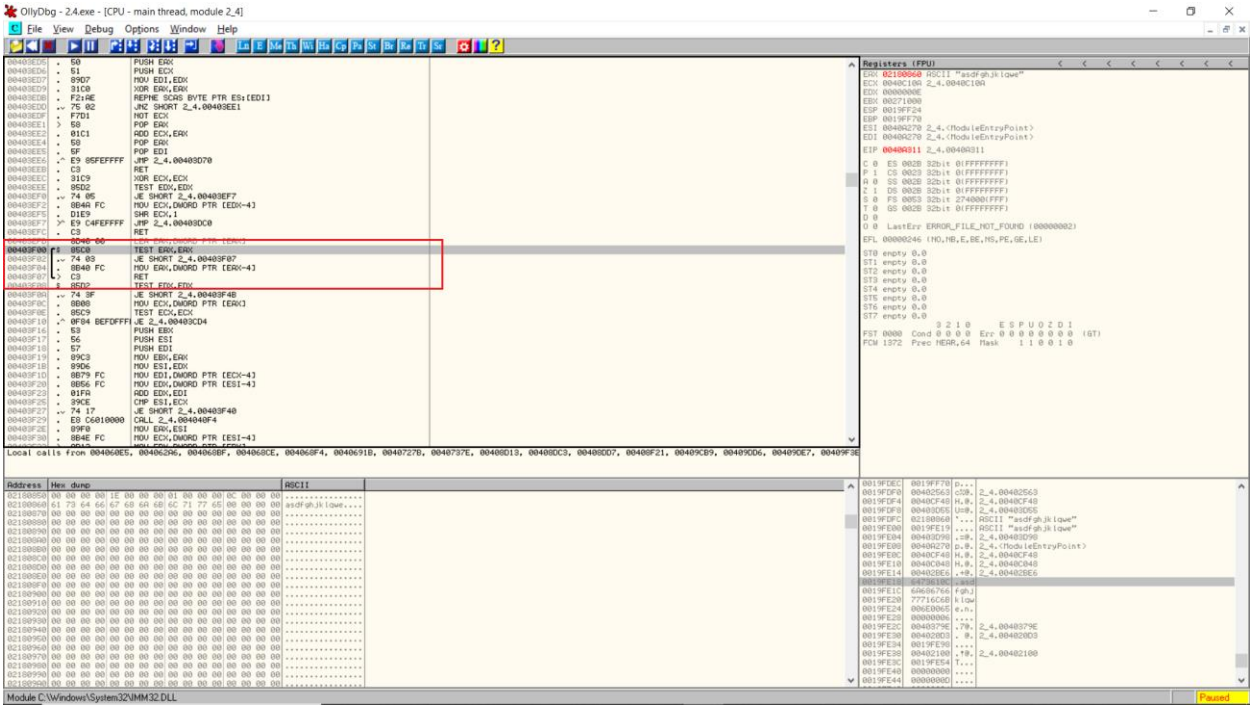
Badboy: “Invalid….”

Chạy OllyDBG phân tích chương trình:

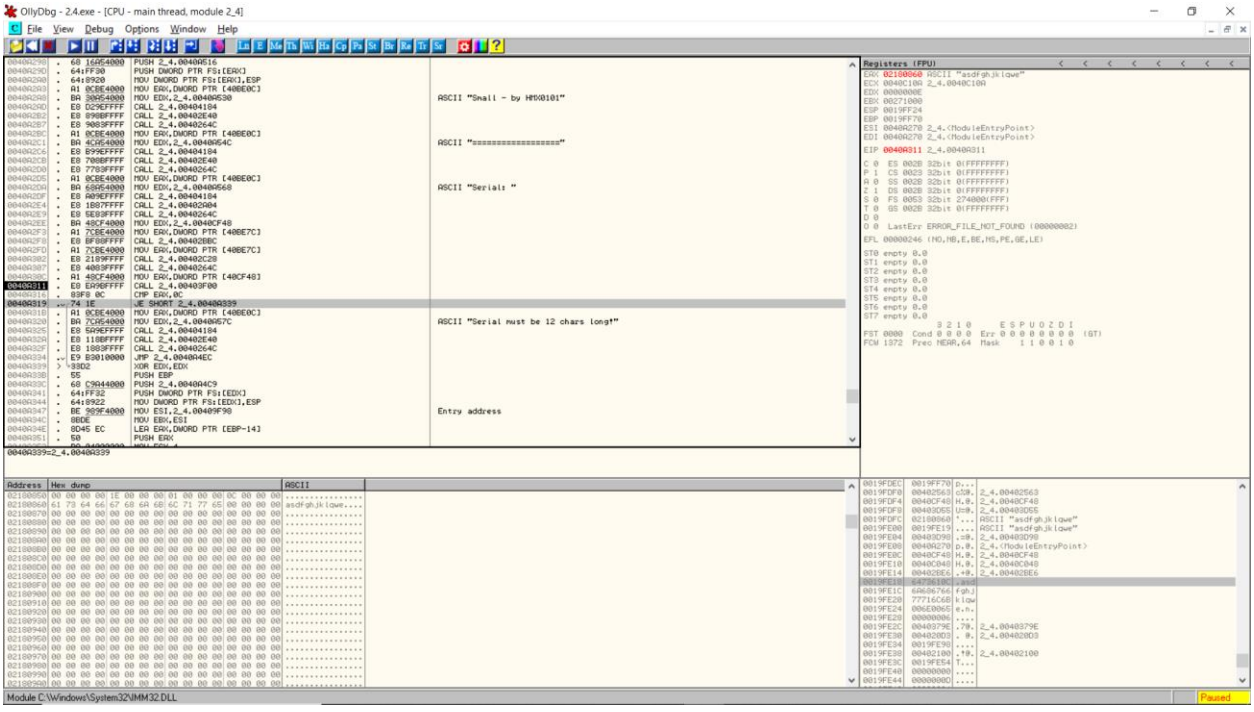


Chương trình sẽ xử lý nhập chuỗi tại 402BBC: lưu chuỗi vào 3 vị trí 40C194 , STACK 19FE19, trong ES 2XXXXXX (Extra Segment do hệ điều hành cung cấp. VD: ở đây là 2180860)
Ở vị trí 2XXXXXX-4 sẽ lưu chiều dài của chuỗi:





Thực ra hàm 403F00 chỉ là lấy giá trị ở vị trí EAX-4: sau đó đem so sánh với 0xC(12) để nhảy qua nhãn kế tiếp



Mức độ hoàn thiện 40%

Tài liệu tham khảo:

https://docs.microsoft.com/en-us/windows/win32/api/winuser

https://drive.google.com/file/d/0B_5KHFI8OfKCQ1BlT0JGcmRZOUU/view