

Bài 2.1:

B1: Mở file 2.1.exe bằng phần mềm OllyDbg

OllyDbg - 2.1.exe - [CPU - main thread, module 2_1]

File View Debug Options Window Help

```
004012B2 . 8130 90010000 XOR DWORD PTR [EAX],190
004012B8 . R1 10404000 MOV ERX,DWORD PTR [4040101]
004012BD . 3B45 FC CMP ERX,DWORD PTR [EBP-4]
004012C0 . 75 0C JNZ SHORT 2_1.004012CE
004012C2 . C705 10404000 MOV DWORD PTR [4040101],1
004012CC . EB 0A JMP SHORT 2_1.004012D8
004012CE . C705 10404000 MOV DWORD PTR [4040101],0
004012D8 . C9 LEAVE
004012D9 . C3 RET
004012DA . 55 PUSH EBP
004012DB . 89E5 MOV EBP,ESP
004012DD . 83EC 18 SUB ESP,18
004012E0 . 83E4 F0 AND ESP,FFFFFFF0
004012E3 . B8 00000000 MOV ERX,0
004012E8 . 83C0 0F ADD ERX,0F
004012EB . 83C0 0F ADD ERX,0F
004012EE . C1E8 04 SHR ERX,4
004012F1 . C1E8 04 SHL ERX,4
004012F4 . 8945 FC MOV DWORD PTR [EBP-4],ERX
004012F7 . 8B45 FC MOV ERX,DWORD PTR [EBP-4]
004012FA . E8 A1040000 CALL 2_1.004017A0
004012FF . E8 3C010000 CALL 2_1.00401440
00401304 . C70424 00304 MOV DWORD PTR [ESP],2_1.00403000
00401306 . E8 A0050000 CALL <JMP.0nsvcrt.printf>
00401310 . C74424 04 10 MOV DWORD PTR [ESP+4],2_1.00404010
00401318 . C70424 15304 MOV DWORD PTR [ESP],2_1.00403015
0040131F . E8 7C050000 CALL <JMP.0nsvcrt scanf>
00401324 . E8 67FFFFFF CALL 2_1.00401290
00401329 . 8330 10404000 CMP DWORD PTR [4040101],1
00401330 . 75 0E JNZ SHORT 2_1.00401340
00401332 . C70424 18304 MOV DWORD PTR [ESP],2_1.00403018
00401339 . E8 72050000 CALL <JMP.0nsvcrt.printf>
0040133E . EB 0C JMP SHORT 2_1.0040134C
00401340 . C70424 40304 MOV DWORD PTR [ESP],2_1.00403040
00401347 . E8 64050000 CALL <JMP.0nsvcrt.printf>
0040134C . E8 CF040000 CALL <JMP.0nsvcrt._getch>
00401351 . B8 00000000 MOV ERX,0
00401356 . C9 LEAVE
00401357 . C3 RET
00401358 . 90 NOP
00401359 . 90 NOP
0040135A . 90 NOP
0040135B . 90 NOP
0040135C . 90 NOP
```

ASCII "Please enter a key: "
printf
scanf
ASCII "%d"
ASCII "Congratulations! You are successful."
printf
ASCII "Better luck next time! You are unsuccessful."
printf
_getch

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)

OllyDbg - 2.1.exe - [CPU - main thread, module 2_1]

File View Debug Options Window Help

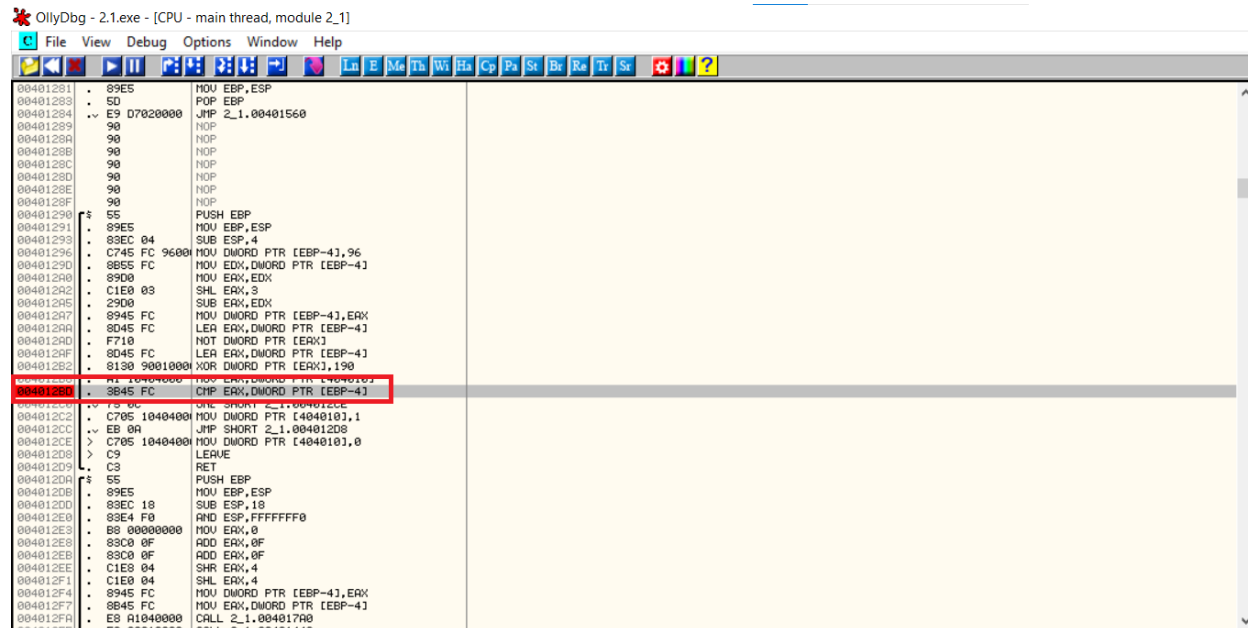
```
004012B2 . 8130 90010000 XOR DWORD PTR [EAX],190
004012B8 . R1 10404000 MOV ERX,DWORD PTR [4040101]
004012BD . 3B45 FC CMP ERX,DWORD PTR [EBP-4]
004012C0 . 75 0C JNZ SHORT 2_1.004012CE
004012C2 . C705 10404000 MOV DWORD PTR [4040101],1
004012CC . EB 0A JMP SHORT 2_1.004012D8
004012CE . C705 10404000 MOV DWORD PTR [4040101],0
004012D8 . C9 LEAVE
004012D9 . C3 RET
004012DA . 55 PUSH EBP
004012DB . 89E5 MOV EBP,ESP
004012DD . 83EC 18 SUB ESP,18
004012E0 . 83E4 F0 AND ESP,FFFFFFF0
004012E3 . B8 00000000 MOV ERX,0
004012E8 . 83C0 0F ADD ERX,0F
004012EB . 83C0 0F ADD ERX,0F
004012EE . C1E8 04 SHR ERX,4
004012F1 . C1E8 04 SHL ERX,4
004012F4 . 8945 FC MOV DWORD PTR [EBP-4],ERX
004012F7 . 8B45 FC MOV ERX,DWORD PTR [EBP-4]
004012FA . E8 A1040000 CALL 2_1.004017A0
004012FF . E8 3C010000 CALL 2_1.00401440
00401304 . C70424 00304 MOV DWORD PTR [ESP],2_1.00403000
00401306 . E8 A0050000 CALL <JMP.0nsvcrt.printf>
00401310 . C74424 04 10 MOV DWORD PTR [ESP+4],2_1.00404010
00401318 . C70424 15304 MOV DWORD PTR [ESP],2_1.00403015
0040131F . E8 7C050000 CALL <JMP.0nsvcrt scanf>
00401324 . E8 67FFFFFF CALL 2_1.00401290
00401329 . 8330 10404000 CMP DWORD PTR [4040101],1
00401330 . 75 0E JNZ SHORT 2_1.00401340
00401332 . C70424 18304 MOV DWORD PTR [ESP],2_1.00403018
00401339 . E8 72050000 CALL <JMP.0nsvcrt.printf>
0040133E . EB 0C JMP SHORT 2_1.0040134C
00401340 . C70424 40304 MOV DWORD PTR [ESP],2_1.00403040
00401347 . E8 64050000 CALL <JMP.0nsvcrt.printf>
0040134C . E8 CF040000 CALL <JMP.0nsvcrt._getch>
00401351 . B8 00000000 MOV ERX,0
00401356 . C9 LEAVE
00401357 . C3 RET
00401358 . 90 NOP
00401359 . 90 NOP
0040135A . 90 NOP
0040135B . 90 NOP
0040135C . 90 NOP
```

ASCII "Please enter a key: "
printf
scanf
ASCII "%d"
ASCII "Congratulations! You are successful."
printf
ASCII "Better luck next time! You are unsuccessful."
printf
_getch

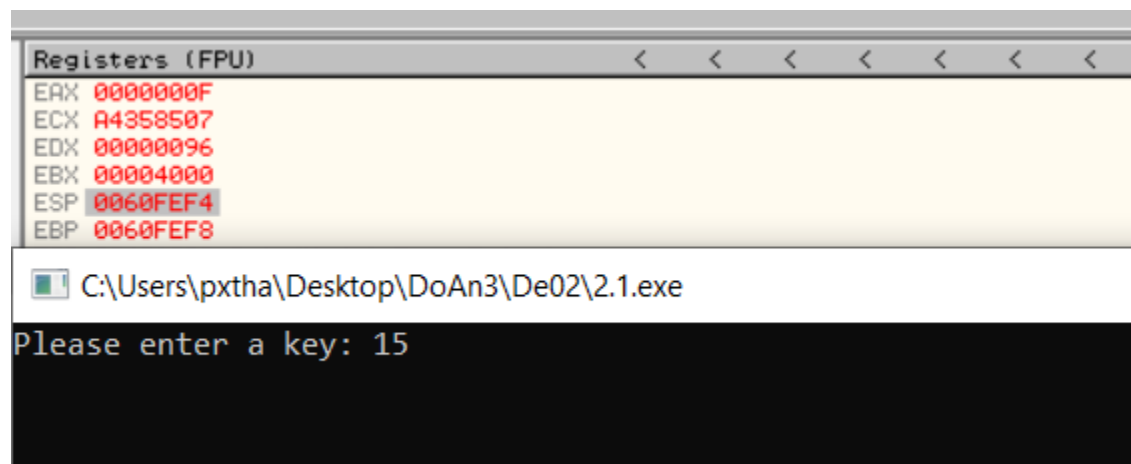
-Sau lệnh nhảy xuất hiện lệnh CMP so sánh kết quả của ô nhớ 404010 với 1 để nhảy tới Goodboy và badBoy nên chương trình sẽ thực thi

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

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



B5: Thực thi chương trình và Nhập Key = 15

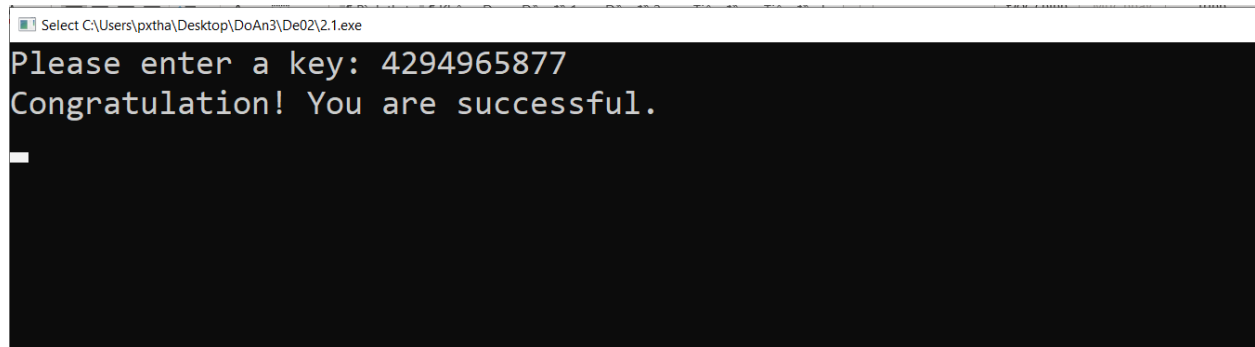


Ta thấy thanh ghi EAX = $F_{16} = 15_{10}$ là Key ta nhập vào

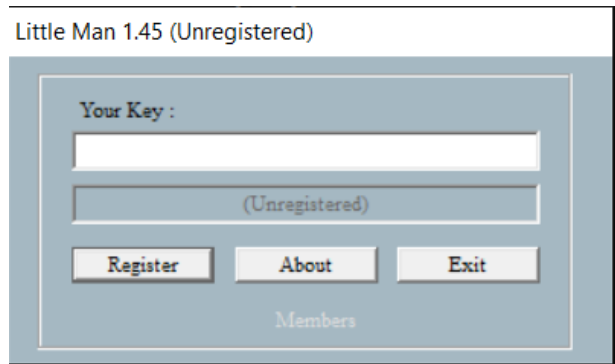
Vậy **EBP - 4 = 60FEF4** sẽ lưu password

B6: Vào ô nhớ ta lấy kết quả: **FFFFFA75₁₆ = 4294965877₁₀** là kết quả

Kết luận: Key là 4294965877



Bài 2.2:

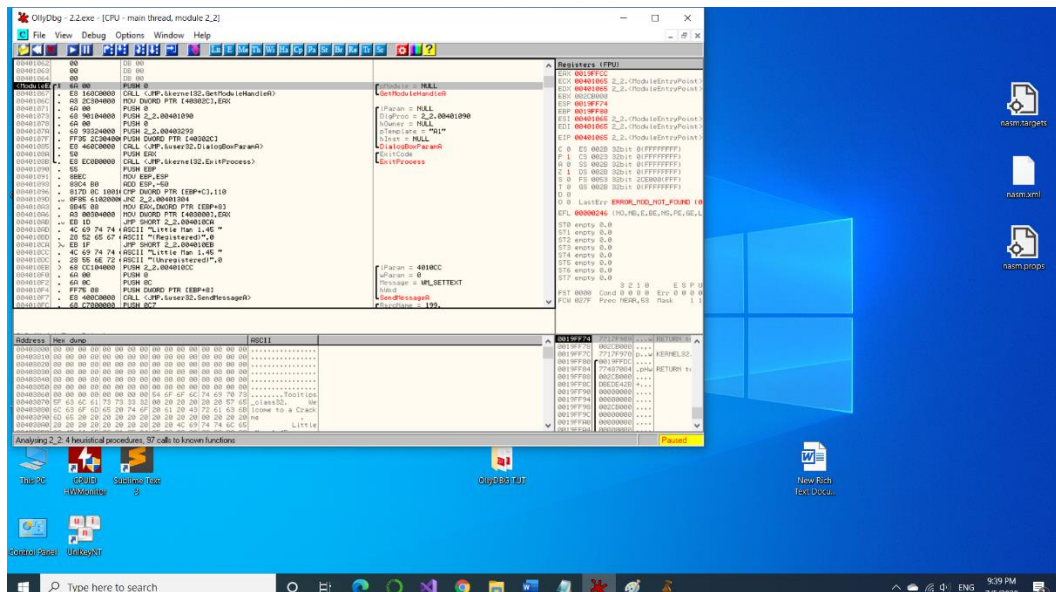


Phân tích:

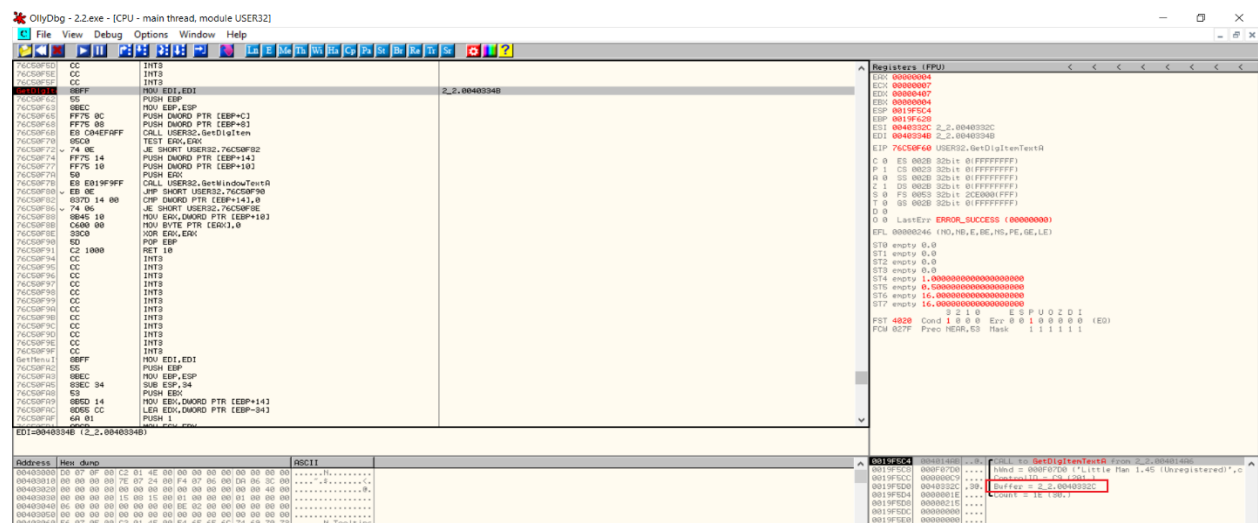
Sử dụng các hàm hỗ trợ Win32 của Microsoft.

Badboy được viết sẵn trong box.

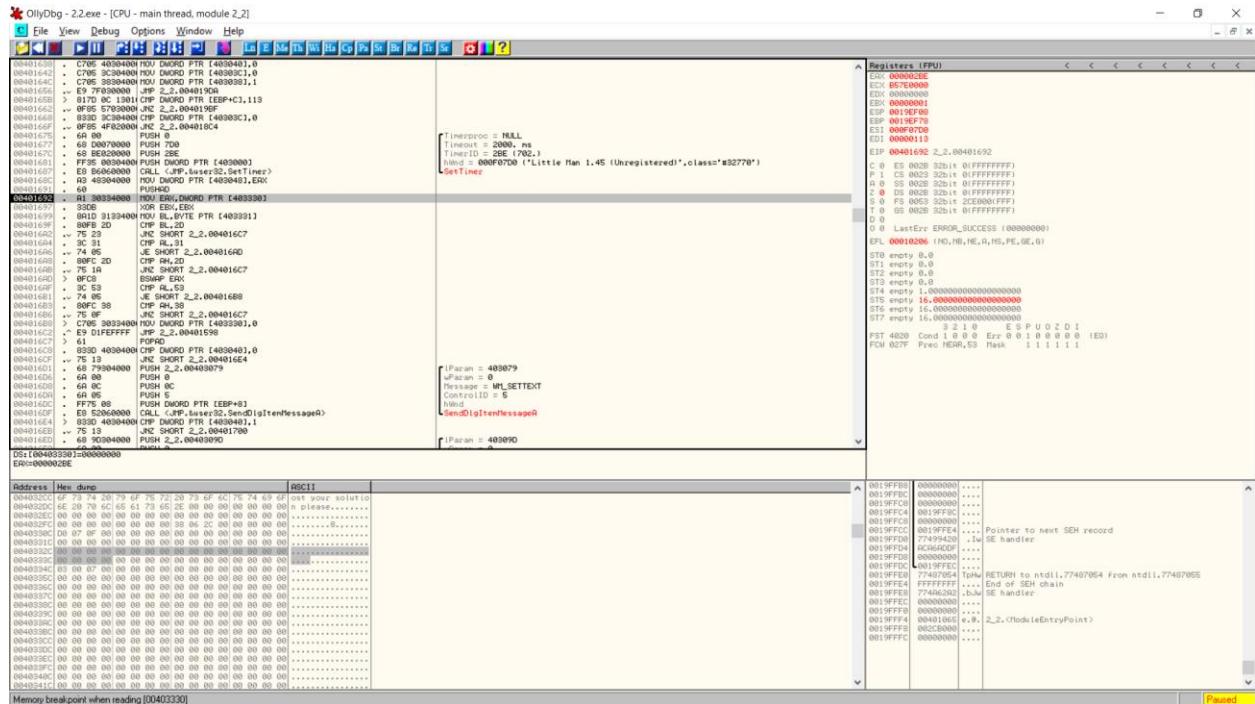
Chạy OIILDBG:



- 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 nhẩy

004016B3 CMP AH,38

So sánh ký tự thứ 7 với '8'

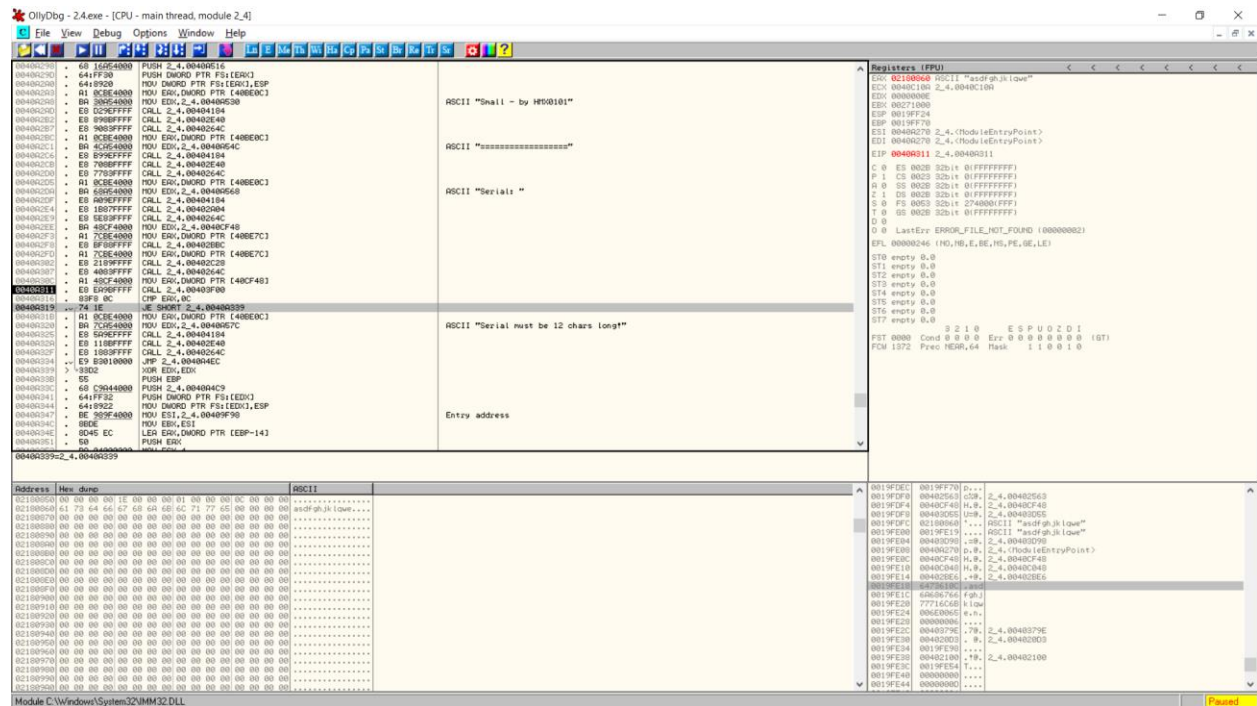
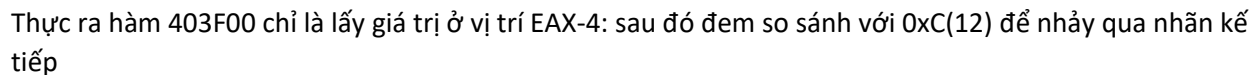
- Vậy ta sẽ có 2 dạng Key (với * là ký tự tùy ý):

- *****_S*****

- *****-8*****

[illegible]

OnlyDag - 24.exe - [CPU - main thread, module 24.exe]
File View Debug Options Window Help



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_5KHFI8OfKCQ1BIT0JGcmRZOUU/view