INTRODUCTION TO THE CRACKING WITH OLLYDBG

FROM CRACKLATINOS

(_kienmanowar_)



Một cái đầu lanh để vững vàng, một trái tim đỏ lửa để yêu và làm việc hết mình!

I. Giới thiệu chung

Khà khà, Giáng sinh rồi ... có được chút thời gian rảnh rỗi tôi lại bắt tay vào viết tiếp bộ tutor này. Hi vọng các bạn vẫn còn hứng thú để đọc những gì tôi viết ⑤. Ở phần 13 của loạt tuts về Ollydbg, tôi đã hướng dẫn các bạn cách kiểm tra xem file có bị pack hay không, cách tìm các điểm quan trọng để tiếp cận mục tiêu, phân tích chi tiết hoạt động của Crackme CrueHead thông qua việc trace và analyze code để từ đó tìm ra một real serial cho chuỗi Name nhập vào. Như vậy, qua bài viết đó tôi đã truyền tải tới các bạn những kinh nghiệm thực tế khi làm việc với một crackme đơn giản nhưng cũng sẽ là tiền đề cho các bạn khi gặp các crackme hoặc các chương trình khác sau này....Ở phần 14 này chúng ta sẽ quay lại làm việc với các target do lão Rincardo đưa ra, cụ thể hơn là tập trung vào chủ đề Fishing Serial ⑤. N0w...L3t's G0 !!!!

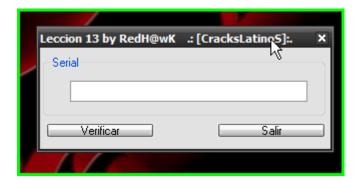
II. Fishing Serial ©

Chà Fishing Serial tức là gì nhỉ? Nghe như kiểu chúng ta đang đi câu cá, giữa một hồ cá rộng và sâu, làm sao ta câu được một con cá ưng ý \odot ... trong ngữ cảnh của Cracking thì ý nghĩa cũng gần như vậy. Fishing Serial ở đây có nghĩa là chúng ta đi câu Serial, mà phải là valid Serial nhé chứ câu lung tung là mệt và dễ stress. Đối với những bạn mới vào nghề thì việc tìm được một valid Serial luôn mang lại một cảm giác lâng lâng khó tả như tìm được một "kho báu" giữa lòng đại dương \odot . Hồi tôi chập chững lọ mọ đọc tươn và cặm cụi mò theo, cho đến khi tìm được serial hợp lệ tự nhiên cảm thấy sướng khó tả, lúc đó nếu có ai ở bên cạnh chắc tôi sẽ kéo vào và chỉ trỏ để khoe những gì mình đã làm, dù chắc gì người đó đã hiểu mình đang làm gì, có khi lại cho mình đang bị chập lolz.

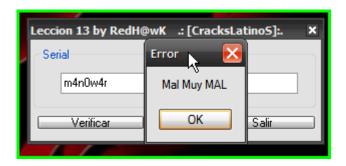
Trong phần 14 này tôi sẽ hướng dẫn các bạn làm việc với dạng Hardcoded Serial (đây là một dạng cơ bản và đơn giản), có nghĩa là dạng Serial cố định không được tính toán dựa trên Name nhập vào, cũng không thay đổi khi bạn chạy trên bất kì máy nào (tức là Serial đó valid trên mọi máy). Có bạn sẽ cho rằng vậy thì dễ quá, viết làm gì? Nhưng xin thưa, phải có dễ thì mới có khó, phải đi từ basic rồi mới tới advanced. Quan điểm của tôi là cứ từ từ mà tiến, không đi đâu mà phải vội vàng. Ok giờ chúng ta vào thực hành luôn nhé, trong bài này ta sẽ làm việc với hai Crackme của lão **RedH@wk** (lão này cũng là một thành viên trụ cột trong CrackLatinos).

1. LESSON 13 HARDCODED 1

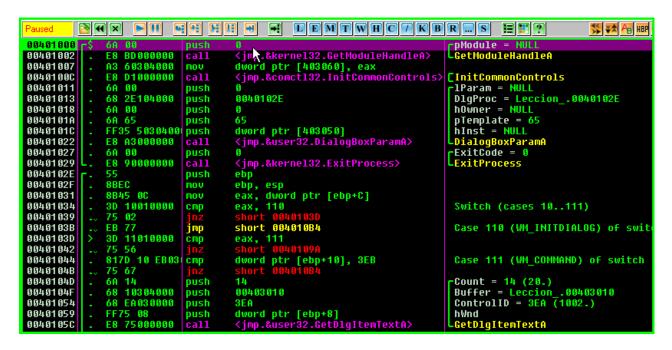
Trước khi load vào Olly, ta chạy thử xem mặt mũi cái crackme này thế nào :



Chà thấy có mỗi cái Textbox cho phép nhập Serial vào, còn mấy cái nút ở dưới toàn tiếng TBN ☺. Nhưng chắc 1 cái dùng để kiểm tra tính hợp lệ của Serial nhập vào, cái còn lại chắc dùng để thoát crackme.Ok, ta nhập đại một serial vào và nhấn thử nút Verificar :



Ăc thông báo bằng tiếng TBN, nhưng may cái Caption là Error, đủ để cho chúng ta biết rằng cái Serial mà ta nhập vào là không hợp lệ. Vậy thì ta phải **"đi câu"** để túm được valid Serial nào ☺. Load cái crackme vào trong Olly :



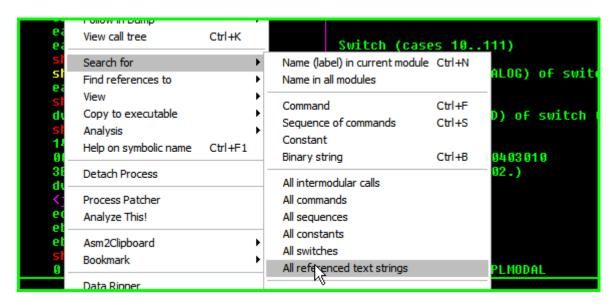
Sau khi load vào Olly, ta dừng lại tại EP của crackme như các bạn thấy ở hình minh họa trên. Dòm qua cái code thấy quá trong sáng, các bạn sẽ bắt gặp cái thông báo Error :

```
00401078 |. 68 35204000 push 00402035 ; |Text = "Mal Muy MAL"
```

và một cái thông báo khác, tôi đoán chắc là thông báo Serial hợp lệ:

```
0040108B |. 68 28204000 push 00402028 ; |Text = "Muy BIEN", A1, "", A1, "", A1, ""
```

Vậy Serial này ở đâu? Vì nó là dạng HardedCode, tức là người code đã include nó vào trong chương trình rồi, ta thử mò bằng cách tìm toàn bộ các String xem thế nào. Chuột phải và chọn như sau :



Ta có được kết quả:

Address	Disassembly	Text string	
00401000 00401061 00401078	mov edx, 00403008	(Initial CPU selection) ASCII "FIACA" nghi qúa ASCII "Mal Muy MAL"	

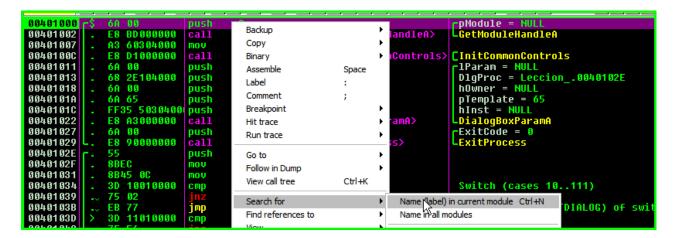
Ò, kết quả có được cũng không nhiều, dòng thứ 3 thì ta đã biết được nó là thông báo lỗi rồi, còn dòng thứ hai là gì nhỉ, thử đưa lên anh google nhờ anh ấy dịch hộ xem nó có nghĩa gì không ☺ :

```
Translation: Spanish » English
FIACA
```

Khà khả kết quả là nó vẫn là chính nó lolz, vậy có thể nghi ngờ nó là valid Serial. Tuy nhiên đây mới chỉ là nghi ngờ thôi, tôi không khuyến cáo các bạn áp dụng cách này để tìm Serial đơn giản vì lý do sau:

"Do đây là crackme đơn giản cho nên tác giả đã code hết sức gọn nhẹ để chúng ta thực hành, cho nên kết quả trả về cho việc tìm kiềm chỉ là hai dòng text duy nhất. Nhưng với các target phức tạp khác, kết quả trả về có thể lên đến hàng trăm hoặc hàng nghìn dòng, việc dò và kiểm thử xem có đúng hay không là điều khó thực hiện và mất thời gian."

Giờ chúng ta đã có điểm nghi ngờ là dòng text thứ hai, nếu nhanh gọn các bạn có thể chạy crackme và nhập vào để kiểm tra tính hợp lệ của nó. Tuy nhiên để chắc chắn hơn chúng ta cần chứng minh đúng là nó chứ không phải dựa vào phán đoán nhất thời. Ta tiến hành tìm kiếm các hàm API mà crackme này sử dụng:



Olly trả về kết quả cho chúng ta như sau :

Paused	≅ ₩ ×		E MTWHC/
Address	Section	Туре	Name
00402018		Import	user32.DialogBoxParamA
00402020		Import	user32.EndDialog
0040200C		Import	kernel32.ExitProcess
0040201C	.rdata	Import	user32.GetDlgItemTextA
00402008	.rdata	Import	kernel32.GetModuleHandleA
00402000	.rdata	Import	comet132.InitCommonControls
00402014		Import	user32. MessageBoxA - パ
00401000	.text	Export	<pre><moduleentrypoint></moduleentrypoint></pre>

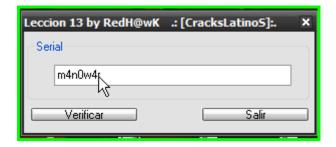
Chà trong danh sách này có hai API mà chúng ta đã khá quen thuộc từ các bài viết trước rồi, ta quan tâm nhất tới hàm user32. GetDlgItemTextA. Đặt BP tại hàm này như sau:

Address	Section	Туре	Name		Comment	
00402018 00402020 0040200C 0040201C 00402008 00402000 00402014	.rdata .rdata .rdata .rdata .rdata .rdata .rdata	Import	user32. DialogBoxParan user32. EndDialog kernel32. ExitProcess user32. CetDlgItemTe- kernel32. CetModuleF comctl32. In tCommor user32. Messag-BoxA			
00401000	.text	Export	<pre><moduleentrypoint></moduleentrypoint></pre>	View call tree Help on symbolic name Toggle breakpoort on impor Conditional breakpoint on in	Ctrl+F1	

Kiểm tra xem BP đã được thiết lập chưa tại cửa số quản lý BP:



Ok, sau khi đặt BP xong ta run chương trình và tiến hành nhập Serial :

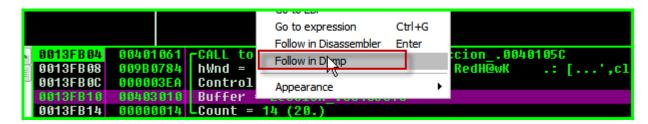


Sau khi nhập xong các bạn nhấn Verify, ngay lập tức Olly sẽ break:

```
LEMTWHC7KB
           44 ×
7E46AE36
           8BFF
                                   edi,
                                        edi
                           MOV
7E46AE38
                           push
           55
                                   ebp
7E46AE39
           8BEC
                           mov
                                   ebp, esp
                                   dword ptr
7E46AE3B
           FF75 0C
                           push
                                              [ebp+C]
           FF75 08
7E46AE3E
                                   dword ptr [ebp+8]
                           push
           E8 888FFBFF
7E46AE41
                                   GetDlgItem
                           call
           8500
74 0E
7E46AE46
                           test
                                   eax, eax
7E46AE48
                           push
7E46AE4A
                                   dword ptr [ebp+14]
           FF75 14
7E46AE4D
           FF75 10
                                             [ebp+10]
                           push
                                   dword ptr
                           push
7E46AE50
           50
                                   eax
 E46AE51
           E8
              D572FCFF
                           call
                                   GetWindowTextA
7E46AE56
           EΒ
              ØE
                                   short
```

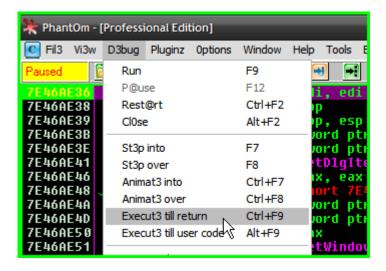
Ta đang dừng lại tại GetDlgItemTextA, dòm qua cửa sổ Stack ta có thông tin như sau :

Ở đây ta cần quan tâm tới Buffer, đây sẽ là nơi nhận lưu Serial mà chúng ta đã nhập vào. Chuột phải vào Buffer và chọn Follow in Dump:



Do hàm API của chúng ta chưa được thực hiện nên vùng buffer hiện thời vẫn còn trống :

Giờ chúng ta cho thực thi hàm API GetDlgItemTextA như sau :



Olly sẽ dừng lại tại lệnh RET, lúc này quan sát vùng Buffer ta có được kết quả như sau :

```
00403010
                бE
                   30
                          34
                                 00
                                    00
                                           00
                                              00
                                                        00
                                                            00 m4n 0w4r
00403020 00
            00
                99
                   99
                       00
                          00 00
                                 00 00
                                        00
                                          99
                                              00 00
                                                     99
                                                        99
                                                           99
00403030 00
            99   99
                   99
                       99
                          00 00
                                 99
                                    00 00 00
                                              00 00 00
                                                        00 00
00403040 00
                   99
                       99
                          00 00
                                    00 00 00
                                              00
                                                 00 00
                                                        00 00
            99   99
                                99
                          aa
                             aa
                                        aa
                                           aa
```

Không cần giải thích chắc các bạn cũng biết được đó là gì rồi! Giờ nhấn F8 để trace qua lệnh Ret để trở về code chính của chương trình :

```
LEMTWHC/KBR...S
                                                                                                           ☆ ★★ AB HBP
               8B1D 1030400 mov
00401066
                                            dword ptr
0040106C
               3B1D 0830400
                                            dword ptr
00401072
00401074
00401076
00401078
               74 13
                              push
               6A 00
                                                                                 ·Style = MB_OK|MB_APPLMODAL
                                                                                 Title = NULL
               6A 00
                              push
                                                                                 Text = "Mal Muy MAL"
               68 35204000
                                       00402035
                              push
              FF75 08
E8 57000000
                                       dword ptr [ebp+8]
<jmp.&user32.MessageBoxA>
short 00401098
0040107D
                              push
                                                                                 hNwner
                              call
00401080
00401085
               EB 11
                              jmp
push
00401087
                                                                                 Style = MB_OK|MB_APPLMODAL
00401089
                              .
push
                                                                                 Text = "Muy BIEN", A1, "", A1, "", A1
               68 28204000
0040108B
                              push
                                       00402028
                              push
call
                                       dword ptr [ebp+8]
<jmp.&user32.MessageBoxA>
00401090
                                                                                 h0wner
               E8 44000000
                                                                                 MessageBoxA
00401093
```

Ok, ta dừng lại quan sát và phân tích một chút. Để ý các bạn sẽ thấy có lệnh so sánh và lệnh nhảy, phụ thuộc vào kết quả so sánh mà lệnh nhảy sẽ đưa chúng ta đến một trong hai thông báo :

- Nếu sai chúng ta sẽ được nhận thông báo "Mal Muy MAL" («bad, very bad» approx. Ed.)
- 2. Nếu đùng chúng ta sẽ nhận thông báo "Muy BIEN" («very good »- Prim.per.).

Rõ ràng trong trường hợp của chúng ta chúng ta muốn nhận được thông báo "**Very good**". Vậy ta trace code để xem điều gì sẽ xảy ra tiếp theo. Đầu tiên ta sẽ thấy chương trình chuyển vào thanh ghi edx một hardcoded string là FIACA:

```
00401061 |. BA 08304000 mov edx, 00403008 ; ASCII "FIACA"
```

Tiếp theo tại 0x00401066, ta thấy thanh ghi ebx sẽ nhận được giá trị từ vùng nhớ [403010], mà theo phân tích ở trên thì đây chính là vùng Buffer nhân vào chuỗi Serial của chúng ta :

```
edx, 00403008
                  BA 08304000 mov
                                                                                                    ASCII "FIACA"
00401061
                   8B1D 10304001
0040106C
00401072
00401074
00401076
                                                                               Backup
                  3B1D 0830400 cmp
                                                 ebx, dword ptr [40
                                                                               Copy
                  74 13
                  6A 00
                                     push
                                                                                                                    DK|MB APPLMODAL
                                                 ß
                                                                               Binary
                  6A 00
                                     push
                                                                               Assemble
                                                                                                      Space
00401078
0040107D
                  68 35204000
                                                                                                                      Muy MAL"
                                                 00402035
                                     push
                                                                               Label
                                                                                                      ÷
                                                 dword ptr [ebp+8]
<jmp.&user32.Messa
short 00401098
                                     push
                                                                               Comment
                  E8 57000000
00401080
00401085
                  ΕB
                                     jmp
push
                                                                               Breakpoint
                  6A 00
                                                                                                                     K|MB_APPLMODAL
00401087
                                                                               Hit trace
00401089
                  6A
                       99
                                     push
                                                                               Run trace
                                                                                                                     BIEN", A1, "", A1
0040108B
                      28204000
                                                 00402028
                                     push
00401090
00401093
                                                dword ptr [ebp+8]
<jmp.&user32.Messa
short 004010B4
                  FF75 08
E8 44000000
                                     push
                                                                               New origin here
                                                                                                      Ctrl+Gray 3
                                     call
00401098
                  EB 1A
                                                                               Follow in Dump
                                                                                                                       Selection
0040109A
                  83F8 10
                                                 eax, 10
                                                                                                                     Memory address
Case 10 (WM CL
0040109D
0040109F
                                                                               View call tree
                                                                                                      Ctrl+K
                       ac
```

Trace qua lệnh này và quan sát thanh ghi EBX, kết quả như sau:

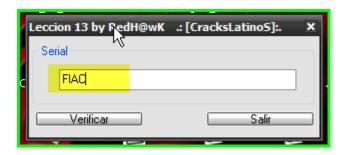
Sau khi trace qua lệnh 00401066 |. 8B1D 10304000 mov ebx, dword ptr [403010], ta sẽ đến lệnh CMP. Ta thấy nội dung của thanh ghi ebx sẽ được đem đi so sánh với vùng chứa chuỗi hardcoded. Ta follow in dump để quan sát giá trị:

```
aanalacu 🚣 lisə idsanaalliih — amola hel [zaaselas.ntatadnovialaiii asel
ds:[00403008]=43414946
ebx=306E346D
       k
                   43 41
                                                          00 FIACA...m4n0w4r
00403008
         46
             49
                41
                          99
                             00
                                00 6D
                                       34
                                          бE
                                             30
                                                77
                                                    34
                                                       72
00403018
         99
             00
               99
                   99
                      99
                          00
                             99
                                00
                                   00
                                       00
                                          00
                                             00
                                                00
                                                    99
                                                       00
                                                          99
00403028
                                00
                                          00
         99
             99
                99
                   99
                      99
                          99
                             99
                                   00
                                       00
                                             99
                                                99
                                                    99
                                                       99
                                                          99
                00
                                       00
00403038
         99
             99
                   99
                      99
                          00
                             00
                                00 00
                                          00
                                             99
                                                99
                                                    99
                                                       00
                                                          00
                             00 00 00 00
00403048
         00
             00
                99
                   99
                      00 00
                                          00 00 00
                                                   99
                                                       00 00
```

Theo như hình trên thì các bạn có thể thấy rằng, không phải toàn bộ chuỗi Hardcoded sẽ được đem đi so sánh với toàn bộ chuỗi serial mà ta nhập vào,mà ở đây chương trình chỉ lấy hai dword (tức là 4 chữ cái đầu tiên) đem so sánh với nhau. Nếu giống nhau thì quá tốt, còn không thì ⊕.Giờ tôi trace qua lệnh CMP :

```
LEMTWHC/KBR...S
         10401061
               BA 08304000
                                               00403008
               8B1D 1030400 mov
3B1D 0830400 cmp
                                         ebx, dword ptr [403010]
ebx, dword ptr [403008]
10401066
1949196C
               74 13
6A 00
10401072
                                                                                             = MB_OK|MB_APPLMODAL
10401074
               6A 00
68 35204000
FF75 08
E8 57000000
10401076
                                                                                      Title = NULL
                               push
                                         00402035
0401078
                                                                                      Text = "Mal Muy MAL"
                               push
                                         dword ptr [ebp+8]
<jmp.&user32.MessageBoxA>
short 00401098
1040107D
                               push
                                                                                      h0wner
                               call
jmp
push
                                                                                      MessageBoxA
10401080
10401085
               EB 11
10401087
               6A
                  99
                                                                                     rStyle = MB_OK|MB_APPLMODAL
```

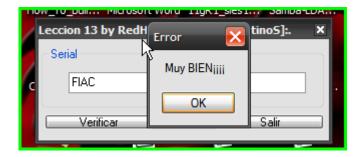
Đương nhiên là lệnh nhảy sẽ không thực hiện ví lý do Serial của tôi nhập vào không trùng khớp với Serial mà chương trình đưa ra. Bây giờ tôi đã biết được Serial của chương trình là gì rồi, đó chính là chuỗi "FIAC", không cần phải đầy đủ hết cả chuỗi vì như tôi đã phân tích ở trên. Chúng ta bỏ BP tại GetDlgItemTextA đi, sau đó đặt BP tại lệnh nhảy ở trên. Tiếp theo Restart Olly và cho thực thi chương trình:



Nhập valid Serial vào, sau đó nhấn Verify. Olly sẽ break:

```
0040106C
               3B1D 0830400 cmp
                                        ebx, dword ptr [403008
00401074
                                                                                             MB_OK|MB_APPLMODAL
                6A
00401076
               68 88
                                                                                    Title =
                               nush
00401078
                   35204000
               68
                                         00402035
                                                                                            "Mal Muy MAL"
                               push
                                                                                    Text =
                                         dword ptr [ebp+8]
<jmp.kuser32.MessageBoxA>
short 00401098
0040107D
                               push
                                                                                    h0wner
                               call
jmp
push
00401080
                   57000000
00401085
00401087
                   99
                                                                                    Style = MB_OK|MB_APPLMODAL
00401089
                6A 00
                               push
                                                                                    Title =
                                                                                              HULL
                   28204000
                                         00402028
0040108B
                68
                               push
                                                                                    Text = "Muy BIEN", A1, "", A1, "", A
               FF75 08
00401090
                               push
                                         dword ptr
                                                     [ebp+8]
                                                                                    hOwner
                   44000000
                                         <jmp.&user32.MessageBoxA>
short 004010B4
                                                                                     lessageBoxA
00401093
```

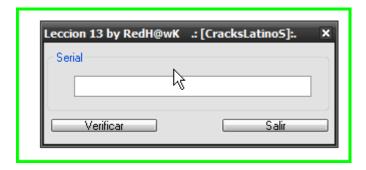
Khà khà Serial đã hợp lệnh, câu lệnh so sánh đã thành công và lệnh nhảy sẽ được thực hiện.Nhấn **F9** phát nữa nào :



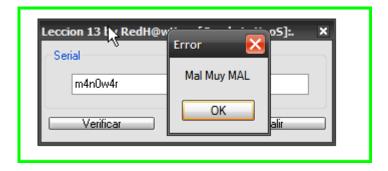
Hola, thành công rồi. Crackme 1 đến đây là kết thúc.

2. LESSON 13 HARDCODED 2

Xử xong crackmel rồi, giờ chúng ta chuyển qua crackme2. Chạy thử nó xem thế nào đã, đây là một bước cần thiết để thu thập thông tin:



Nhập thử Serial và nhấn Verify:



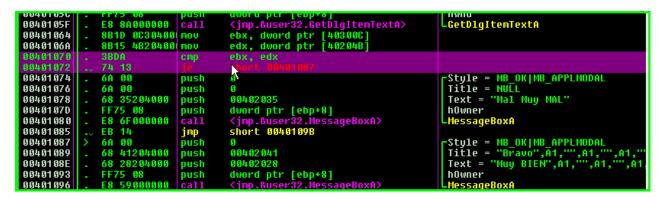
Chà thông báo trên thấy quen quá ©. Giờ ta load Crackme vào Olly để tìm hướng giải quyết bài toán. Trong Olly ta có được như sau :

```
-pModule = NULL
-GetModuleHandleA
                                                  <jmp.&kerne132.GetModuleHandleA>
dword ptr [403008], eax
<jmp.&comct132.InitCommonControls>
00401002
00401007
0040100C
00401011
00401013
00401018
                  A3 08304000
E8 E9000000
                                                                                                    CInitCommonControls
| Param = NULL |
| DlgProc = Leccion_.0940102E |
| hOwner = NULL |
                  6A 00
68 2E104000
6A 00
6A 65
                                      push
                                                  0040102E
                                      push
                                      bush
                                                                                                      pTemplate = 65
hInst = NULL
DialogBoxParamA
0040101A
                                      push
                                                  65
                                      push
                                                 dword ptr [403008]
<jmp.&usersv.DialogBoxParamA>
0040101C
                   FF35 0830400
00401022
                  6A 80
E8 A8000000
                   E8 BB000000
                                      call
                                                                                                      ExitCode =
00401027
                                      push
00401029
0040102E
0040102F
                                                                                                     ExitProcess
                                                  <jmp.&kernel32.ExitProcess>
                                      push
                   8BEC
                                                 ebp, esp
eax, dword ptr [ebp+C]
                                      mnu
00401031
                   8B45 0C
                                      mov
00401034
                   3D 10010000
                                                 eax, 110
                                      cmp
                                                                                                      Switch (cases 10..111)
00401039
                                                  904919CC
eax, 111
                                      jmp
cmp
                   E9 8C000000
                                                                                                      Case 110 (WM_INITDIALOG) of swite
0040103B
00401040
00401045
00401047
0040104E
                   3D 11010000
                   75 6B
                  817D 10 EB03
75 4D
                                                  dword ptr [ebp+10], 3EB
                                                                                                      Case 111 (WM_COMMAND) of switch
                                      cmp
00401050
                   6A 14
                                      push
                                                                                                      Count = 14 (20.)
00401052
                       00304000
                                                  0040300C
                                                                                                      Buffer = Leccion_
                                      push
00401057
                   68 EA030000
                                                  3EA
                                                                                                      ControlID = 3EA (1002.)
                                      push
                                                  dword ptr [ebp+8]
<jmp.&user32.GetDlgItemTextA>
0040105C
                                      push
                                                                                                      hWnd
                   E8 8A000000
                                                                                                      GetDlgItemTextA
                                      call
0040105F
```

Như các bạn thấy code trông giống hệt crackme1, ta thử search string xem có tìm được vàng không nhé:



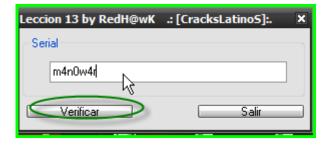
Ái chà, crackme2 này có vẻ khó hơn crackme1 rồi đây. Thông tin ta có được chỉ vẻn vẹn có mỗi một dòng thông báo lỗi, vậy là đầu mối tìm kiếm này đã không có tác dụng. Quay trở vể cửa sổ CPU của Olly, cuộn chuột xuống ta sẽ thấy đoạn code so sánh:



Tại đây ta cũng không thấy có dòng nào tương tự như crackme1, chẳng thấy có thông tin nào tương tư như 'FIACA' mà ta có được với crackme1 cả lolz ②. Do code của crackme2 tương tự crackme1 cho nên phần phân tích tôi bỏ qua, ta đi thẳng vào vấn đề chính. Tôi đặt BP tại lệnh mov bên dưới lời gọi tới hàm GetDlgItemTextA.



Nhấn **F9** để thực thi crackme, sau đó nhập đại một cái Serial vào rồi nhấn Verify:



Olly sẽ break tại chỗ mà ta đặt BP, ta Follow in Dump để biết nội dung tại [40300C] là gì:

```
ds:[0040300C]=306E346D
ebx=000000000
0040300C <mark>6D</mark>
                                                    99
                    30
                                                        99
                                                           99
                                                               99
                               72 00
                                      00
                                          00
                                             99
                                                 00
                                                                  m4n Ow4r
                        00 00
                               00 00
                                      00 00 00 00
                                                    00 00 00 00
0040301C
          99
             99
                 00 00
             99
                               00 00 00 00 00
```

Ta thấy 4 bytes sẽ được đưa vào thanh ghi ebx. Quan sát thanh ghi :

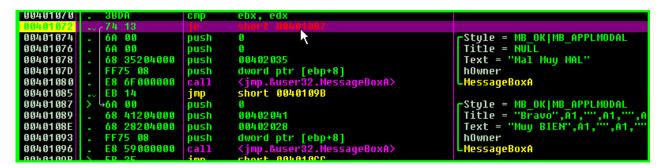
```
Registers (FPU)
EAX 00000007
ECX 7E43218C user32.7E43218C
EDX 7C90EB94 ntd11.KiFastSystemCallRet
EBX 306E346D
ESP 0013FB18
EBP 0013FB18
ESI 0040102E Leccion_.0040102E
EDI 0013FB80
EIP 0040106A Leccion_.0040106A
```

Vậy là thanh ghi ebx đã giữ thông tin về 4 kí tự đầu tiên trong chuỗi Serial mà tôi nhập vào. Bây giờ chúng ta đang dừng tại câu lệnh :

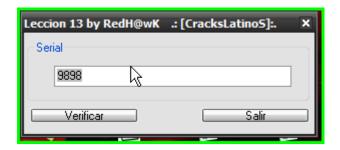
```
0040106A |. 8B15 4B204000 mov edx, dword ptr [40204B]
```

Không hiểu thanh ghi edx được mov thông tin gì vào nhỉ? Điểm nghi ngờ là ở đây, vì ở bên dưới tôi thấy lệnh cmp đem hai thanh ghi ebx và edx so sánh với nhau. Vậy ta có thể khẳng định chắc chắn rằng, vùng nhớ [40204B] đang nắm giữ một thông tin rất quan trọng, đó chính là valid Serial. Ta Follow in dump để biết thông tin đó là gì:

Khà khà, vậy là Serial mà chúng ta cần tìm là : 9898. Quay lại cửa số code, bỏ BP đã đặt trước đó đi, sau đó bạn đặt BP tại lệnh nhảy bên dưới lệnh cmp :

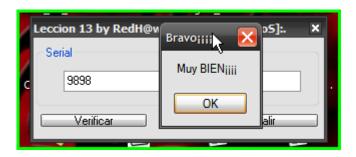


Restart lại Olly, sau đó nhấn F9 để run chương trình. Nhập valid Serial là 9898 vào và nhấn Verify:



```
00401074
00401076
00401078
                  6A 00
68 35204000
FF75 08
                                      push
                                                                                                      Title = NULL
                                                                                                      Text = "Mal Muy MAL"
                                                  00402035
                                      push
0040107D
00401080
                                                 dword ptr [ebp+8]
<jmp.&user32.MessageBoxA>
short 0040109B
                                      push
                                                                                                      h0wner
                  E8 6F000000
                                     call
jmp
push
00401085
                      14
00
                  EB
00401087
                                                                                                      Style =
                                                                                                                 MB OK | MB APPLMODAL
00401089
                  68 41204000
                                                                                                      Title = "Bravo",A1,"",A1,"",A1,
Text = "Muy BIEN",A1,"",A1,"",A
                                      .
push
0040108E
                  68 28204000
                                     push
                                                  00402028
                                                  dword ptr [ebp+8]
<jmp.&user32.MessageBoxA>
00401093
                                      .
push
                                                                                                      hOwner
                       59000000
00401096
```

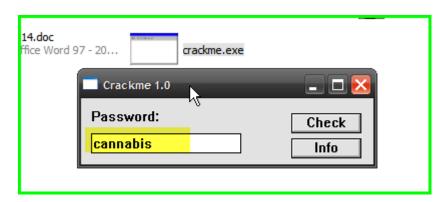
Hehe, như các bạn thấy lệnh nhảy tới Good boy sẽ được thực hiện sau khi so sánh. Vậy là việc Fishing của chúng ta đã thành công, nhấn F9 để kiểm chứng kết quả cuối cùng :



Vậy là xong, crackme2 đã được xử lý nhẹ nhàng!!

Ok vậy là phần 14 của loạt tuts về Ollydbg đến đây là kết thúc, qua bài viết này tôi đã giới thiệu tới các bạn thế nào là Fishing Serial và hướng dẫn các bạn thực hành với hai Crackme rất đơn giản, ngoài ra trong phần 14 này cũng có thêm một target nữa của lão **Detten** để các bạn tự thực hành để kiểm chứng kiến thức và khả năng của mình. Hi vọng qua bài viết này của tôi đã cung cấp thêm một số kiến thức cơ bản giúp các bạn làm việc hiệu quả với Olly. Hẹn gặp lại các bạn trong các phần tiếp theo của loạt tutor này, By3 By3!! ©

Solution for Detten's crackme:



Best Regards

[Kienmanowar]



--++--==[Greatz Thanks To]==--++--

My family, Computer_Angel, Moonbaby, Zombie_Deathman, Littleboy, Benina, QHQCrker, the_Lighthouse, Merc, Hoadongnoi, Nini ... all REA's members, TQN, HacNho, RongChauA, Deux, tlandn, light.phoenix, dump, dqtln, ARTEAM all my friend, and YOU.

--++--==[Thanks To]==--++--

iamidiot, WhyNotBar, trickyboy, dzungltvn, takada, hurt_heart, haule_nth, hytkl, moth, XIANUA, nhc1987, 0xdie, Unregistered!, akira, mranglex v..v.. các bạn đã đóng góp rất nhiều cho REA. Hi vọng các bạn sẽ tiếp tục phát huy ©

I want to thank **Teddy Roggers** for his great site, Reversing.be folks(especially **haggar**), Arteam folks(**Shub-Nigurrath, MaDMAn_H3rCuL3s**) and all folks on crackmes.de, thank to all members of **unpack.cn** (especially **fly** and **linhanshi**). Great thanks to **lena151**(I like your tutorials).Thanx to Orthodox, kanxue, TiGa and finally, thanks to **RICARDO NARVAJA** and all members on **CRACKSLATINOS**.

>>>> If you have any suggestions, comments or corrections email me: kienmanowar[at]reaonline.net

