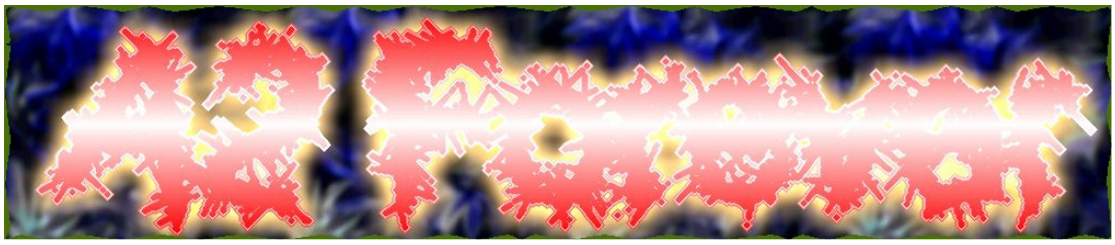


Tut2

Target	fwdv2, fwdv3.
Độ khó	Cơ bản
Packed	N/A
Công cụ	Olly Dbg 1.10, PEiD 09.3, Máy tính ☺
Mục tiêu	Tìm Serial

Tiếp theo tut1, hôm nay tôi sẽ cùng các bạn thực hành với 2 Crackme tiếp theo, trước khi tiếp tục tôi muốn bạn nhớ lại:

-Các bước cần thiết khi Crack một Target.
-Cách sử dụng Olly Dbg mức căn bản.
-Kiến thức về BreakPoint.
-Ý nghĩa của hai câu lệnh JE và CMP.
...

fwdv2 Crackme

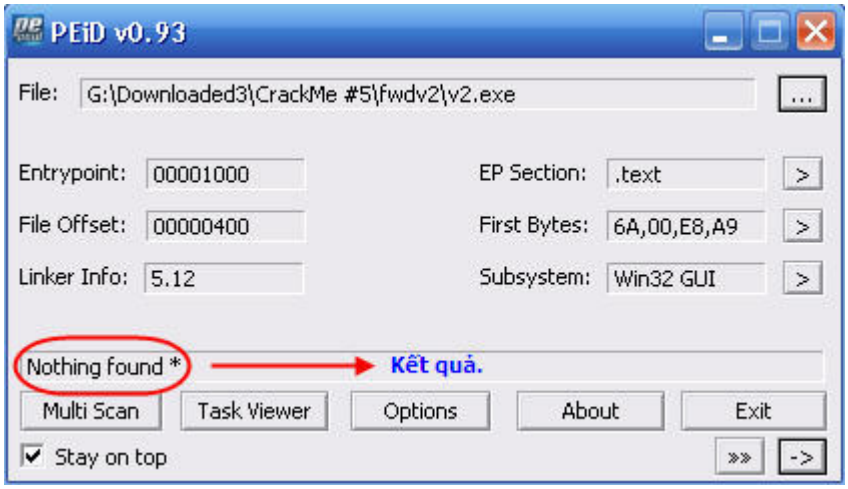
Lấy thông tin:
Thử chạy Crackme ta thấy như sau:

Giao diện tương tự như Crackme1 ☺. Nhập thử một số bất kỳ vào ô nhập Serial – tôi nhập là 30041991, sau đó nhấn OK, một cái Nag (Nag là cái gì ?) hiện ra:

Đây là Goodboy hay Badboy ?
Crackme này không có nhiều thông tin để ta tìm hiểu, với một ít thông tin có được, chúng ta bắt tay vào Crack...

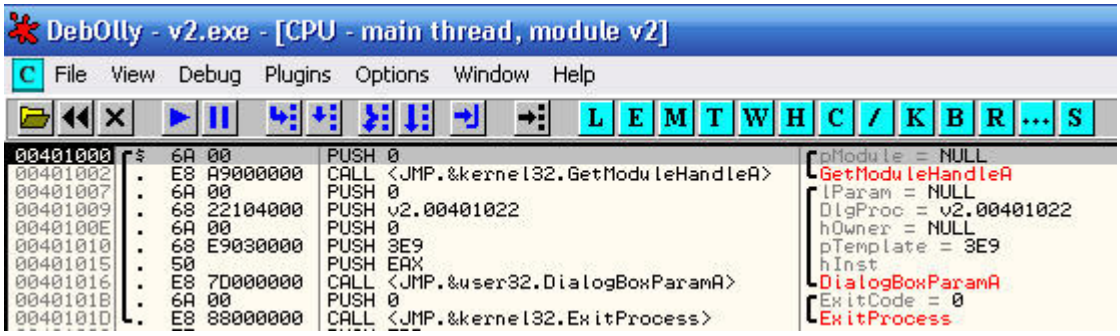
Crack:

Dùng PEiD (đọc lại tut1 để biết cách kiểm tra bằng PEiD) để kiểm tra ta Target có kết quả:

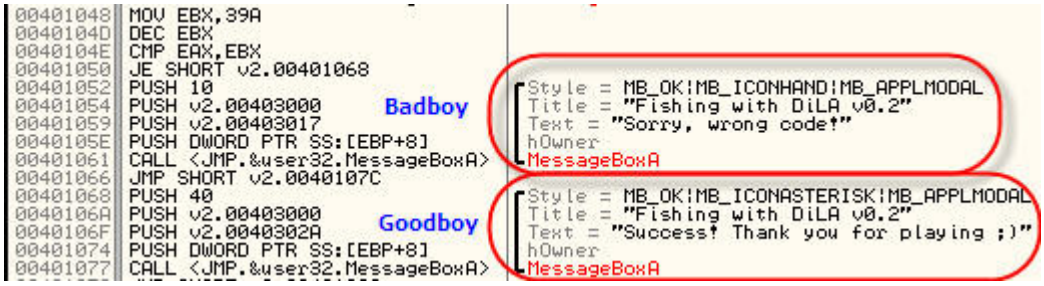


Lại là Nothing found*, tuy nhiên với Crackme trước chúng ta vẫn có thể crack được bằng Olly nên hôm nay hãy thử làm như vậy...

Mở Olly lên, sau đó mở Crackme trong Olly (phím tắt là gì ?) – Thao tác mở một Target trong Olly ta thường gọi là "Load" (có thể bạn sẽ gặp một số câu đại loại như "Load Target trong Olly ta thấy...").
Chờ Olly load xong chúng ta ở đây:



Để tìm chuỗi (String) "Sorry, wrong code!" chúng ta có thể dùng chức năng tìm kiếm String trong Olly hoặc có thể thấy ngay chuỗi này vì code của Crackme này rất ngắn (bạn có nhớ phải làm thế nào để tìm kiếm String trong Olly không ?)
Dù làm theo cách nào thì cái mà tôi muốn các bạn thấy là đoạn code này:



Vận dụng kiến thức ở tut1, bạn hiểu gì về đoạn code này...
Thực ra, chúng ta không cần hiểu nhiều lắm về đoạn code này nhưng tôi muốn bạn nắm được những điều sau:

-Câu lệnh nhảy JE trên Badboy có ý nghĩa gì ?.
-Câu lệnh CMP EAX, EBX để làm gì ?.
-Từ các điều đó hãy xem chúng ta cần gì để đạt được Goodboy ?...

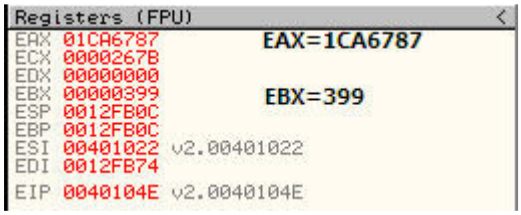
Hãy nhớ lại tut1 và vận dụng vào Crackme này và ta thấy:

-Lệnh nhảy JE nếu thực thi sẽ đưa ta đến Goodboy.
-Câu lệnh CMP EAX, EBX để so sánh hai giá trị EAX và EBX, đây chính là thứ quyết định xem câu lệnh JE có thực thi hay không
→Để đạt được Goodboy thì ta cần EAX = EBX ☺

Cái chúng ta cần xem bây giờ là giá trị của EAX và EBX tại câu lệnh CMP EAX, EBX → bạn sẽ làm điều này như thế nào ?
Để xem giá trị của EAX và EBX tại câu lệnh CMP EAX, EBX thì ta cần set BP tại câu lệnh này, sau đó khi Olly dừng lại ta sẽ thấy cái cần tìm ☺.
Chọn câu lệnh CMP EAX, EBX sau đó nhấn F2 (để làm gì ?):

Bây giờ nhấn F9 (để làm gì ?), chuyển sang Crackme, đăng ký lại – tôi vẫn điền là 30041991, sau đó nhấn nút Register → Olly dừng tại BP ☺.
Trở lại Olly ta thấy như sau:

Nhìn sang bên phải tại cửa sổ Registers (FPU) ta thấy giá trị của EBX và EAX tại phép so sánh này:



Tôi hy vọng bạn hiểu cả hai giá trị này đều là những giá trị ở hệ HEX, bạn còn nhớ 1CA6787 là gì không ☺.
Nếu bạn đã đọc qua tut1 thì tôi xin nói ngay 1CA6787 là giá trị ở hệ HEX của số 30041991 – Còn nếu bạn chưa đọc tut1→tôi khuyên bạn hãy tìm và đọc nó nếu thấy cần.
Như vậy với những kiến thức có được từ tut1 ta có thể thấy được "Serial thực" là số nào rồi phải không – Hãy tự tìm nó trước khi đọc tiếp...

Chúng ta có các thông tin sau:

-Trên Badboy có một lệnh nhảy sẽ nhảy qua Badboy để đến Goodboy → Nhiệm vụ của chúng ta là làm cho lệnh nhảy này thực thi.
-Trên lệnh nhảy JE có một phép so sánh CMP EAX, EBX dùng để so sánh hai gí trị EAX và EBX (ở hệ HEX) → Chúng ta cần có EAX = EBX để thực thi lệnh nhảy JE đồng nghĩa ta sẽ đến được Goodboy.

Chúng ta thấy chỉ đơn giản là Serial ta nhập vào sau khi chuyển từ DEC→HEX phải có giá trị là 399 (đọc lại tut1 để xem chi tiết), vì vậy ta chỉ cần đổi giá trị 399 từ HEX→DEC là có "Serial thực": Tôi dùng máy tính Windows để thực hiện (xem lại tut1 để biết cách thực hiện):

Vậy "Serial thực" là 921, hãy thử xem:

Chúng ta đã xong Target này, nhưng tôi muốn bạn biết thêm một chút về ngôn ngữ ASM → Tại sao giá trị của EBX tại phép so sánh CMP EAX, EBX lại = 399. Bạn có muốn biết không ☺

Trở lại đoạn code có câu lệnh CMP EAX, EBX → nhìn lên trên ta thấy có các câu lệnh sau:

Bạn cần biết một số câu lệnh sau:

-MOV X, Y → đặt giá trị X = Y, điều này có nghĩa là dù trước đó X có bằng bao nhiêu đi nữa thì sau câu lệnh này X sẽ phải bằng Y, ví dụ như X đang bằng 2, sau câu lệnh MOV X, 1 thì X sẽ = 1 (cũng không quá khó nhỉ ☺).
-DEC X → Giảm giá trị của X đi 1, ví dụ như X đang = 2 sau câu lệnh này X sẽ =1 (2 – 1 = 1 ☺).

Với các kiến thức đó (hãy luôn nhớ chữ H chỉ là thể hiện giá trị đó đang ở hệ HEX thôi), bạn hãy thử vận dụng xem tại sao đến câu lệnh CMP EAX, EBX giá trị của EBX lại = 399 (mã HEX) nhé → Nếu bạn không trả lời được thì hãy hỏi trên Forum, tôi sẽ trả lời cho bạn.
Trước khi chuyển sang Target tiếp theo, tôi có một bài tập nhỏ cho bạn, hãy xét đoạn code sau và xem chúng ta cần gì để đạt Goodboy với EAX là Serial ta nhập vào đã chuyển sang HEX (đây chỉ là một ví dụ nên hãy khoan bàn đến tính chính xác của đoạn code):

BT1:

1111	MOV EBX, 1991
2222	MOV EBX, 3004
3333	DEC EBX
4444	CMP EAX, EBX
5555	JE 8888
6666	Badboy

7777 ...
8888 Goodboy

fwdv3 Crackme

Lấy thông tin:

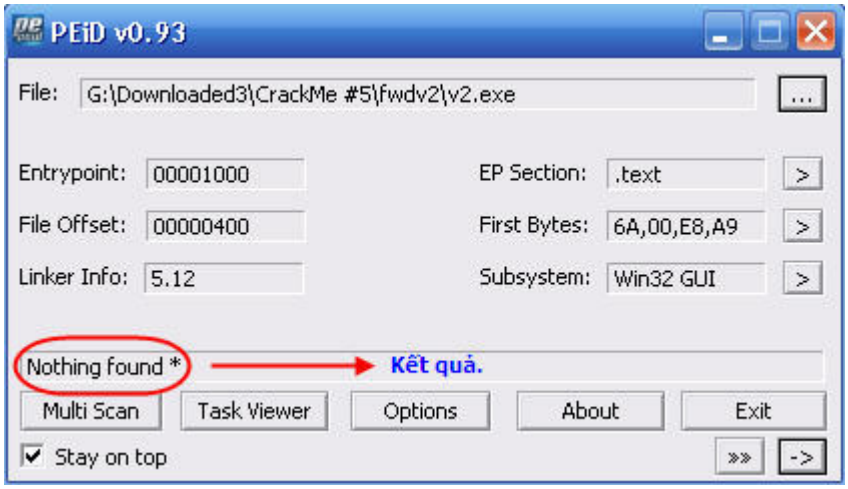
Thử chạy Crackme ta thấy như sau:

Giao diện tương tự như Crackme1 ☺. Nhập thử một số bất kỳ vào ô nhập Serial – tôi nhập là 30041991, sau đó nhấn OK, một cái Nag (Nag là cái gì ?) hiện ra:

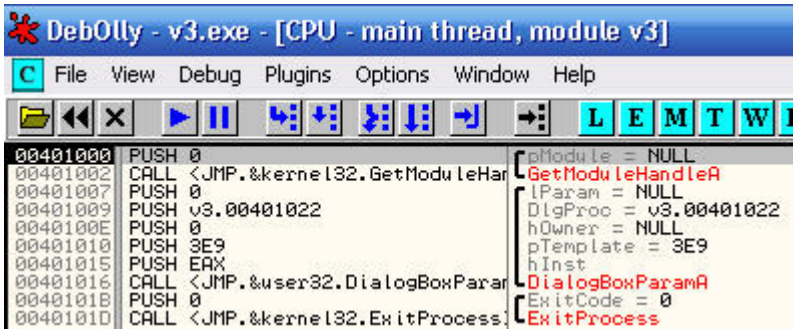
Tôi sẽ không nói lại những điều đã nói ở trên, vì vậy hãy bắt tay vào Crack thôi ☺

Crack:

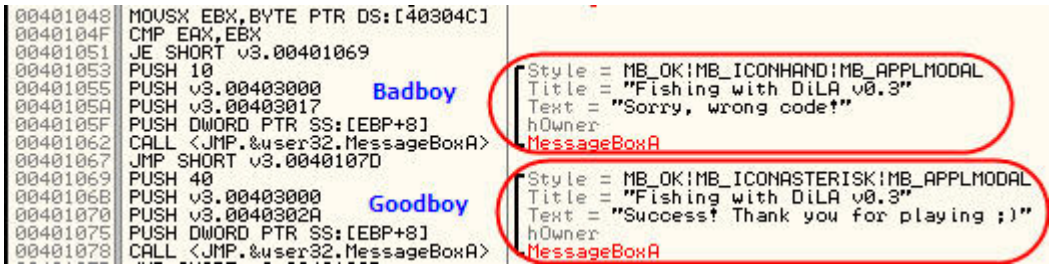
Dùng PEiD để kiểm tra ta Target có kết quả:



Bạn nghĩ sao ? Chẳng cần phải nghĩ gì cả, hãy mở Olly lên ☺. Load Target trong Olly, chờ Olly load xong ta ở đây:



Nhìn xuống dưới một chút ta thấy:



Với các kiến thức bạn có được cho đến lúc này, hãy thử tìm "Serial thực" trước khi đọc tiếp ☺.

Chúng ta hãy đặt BP tại câu lệnh CMP EAX, EBX:

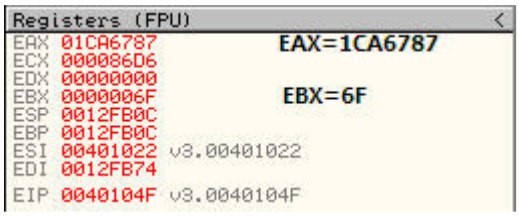
Nhấn F9 để chạy Target, đăng ký sau đó nhấn nút Register → Olly dừng tại BP:

```
0040104F  CMP EAX,EBX
```

Nhấn F2 bỏ BP tại đây:

```
0040104F  CMP EAX,EBX
```

Nhìn sang cửa sổ Registers (FPU) để thấy giá trị của EAX và EBX tại phép so sánh:



Từ một số kiến thức ở trên có thể thấy “Serial thực” là:

Yep ! Vậy “Serial thực” là 111, thử xem sao:

Vậy là xong...

Tuy nhiên tôi muốn bạn biết tại sao giá trị của EBX tại phép so sánh lại = 6F (mã HEX).
Trở lại đoạn code và chú ý câu lệnh sau:

Bạn cần biết câu lệnh sau:
MOVSX X, BYTE PTR [Y] → Gán X là ký tự đầu tiên trong chuỗi Y. Ví dụ: ta có chuỗi Y=”ABC” thì sau câu lệnh MOVSX X, BYTE PTR [Y] giá trị của X sẽ là 41 (41 là mã HEX của ký tự A trong mã ASCII). Ở đây bạn cần biết thêm một chút về mã ASCII:

-ASCII: viết tắt của American Standard Code for Information Interchange.
Là một cách định nghĩa những ký tự mà có thể hiển thị bởi một máy tính trên một màn hình, cũng như một số ký tự điều khiển mà có những chức năng đặc biệt.

Việc chuyển từ mã ASCII sang các mã khác và ngược lại khá phức tạp, với mã ASCII bạn không thể sử dụng máy tính Windows tuy nhiên vẫn có thể sử dụng các công cụ chuyển đổi khác ☺. (Có kèm theo trong tut)

00401048 l.>MOVSX EBX, BYTE PTR [40304CH]
Câu lệnh này bạn không thể hiểu Y là 40304CH được, không cần quá phức tạp về vấn đề này, tôi chỉ muốn bạn hiểu 40304CH là một địa chỉ mà tại câu lệnh MOVSX EBX, BYTE PTR [40304CH] nó đang chứa một chuỗi nào đó, có nhiều cách để xem chuỗi mà địa chỉ 40304CH đang chứa là gì, hãy thử làm như sau:

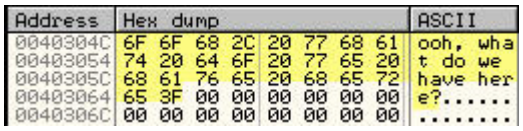
-Đầu tiên nhấn Ctrl+F2 để Restart lại Target.

-Đặt BP tại câu lệnh MOVSX EBX, BYTE PTR [40304CH].

-Nhấn F9 để chạy Target, đăng ký lại như bình thường, sau đó nhấn nút Register, Olly dừng lại:

Nhấn chuột phải và chọn như sau:

Sau đó nhìn xuống cửa sổ HEX Dump ở dưới và thấy:

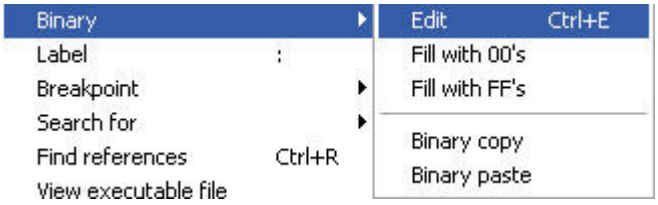


Bạn không cần thấy quá phức tạp, chỉ cần nhìn cột ASCII và bạn có thể thấy ký tự mà địa chỉ 0040304C đang chứa là o trong cả một chuỗi mặc định là "ooh, what do we have here?", dùng một công cụ chuyển đổi từ mã ASCII sang mã HEX ta được kết quả :

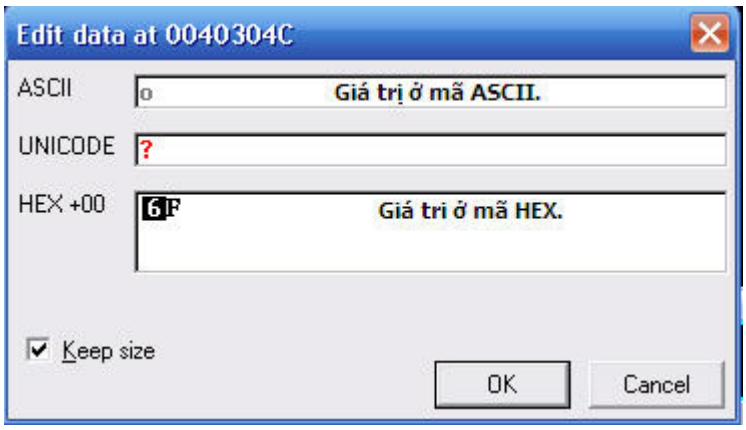
Như vậy là ta đã biết tại sao giá trị của EBX tại câu lệnh CMP EAX, EBX là 6F rồi phải không nào ☺...
Tuy nhiên, không chỉ muốn tut này chỉ dừng lại tại đây, chúng ta hãy thử tìm hiểu xem:
-Liệu có thể thay chuỗi mặc định này bằng chuỗi mặc định của ta không nhỉ?

→ Câu trả lời là có : Việc này thực ra chỉ là thay đổi giá trị đang lưu trong bộ nhớ của Target sau đó lưu lại thành một file mới.

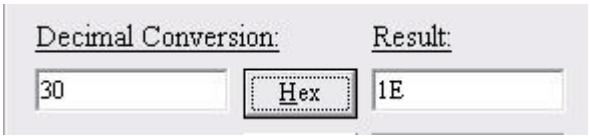
Bây giờ giả sử tôi muốn "Serial thực" phải là 30 thì phải làm sao ?
Hãy thử như sau:
Nhấn đúp chuột vào chữ o đầu tiên trong chuỗi "ooh, what do we have here?", sau đó nhấn Ctrl+E (Edit) hoặc chọn như sau:



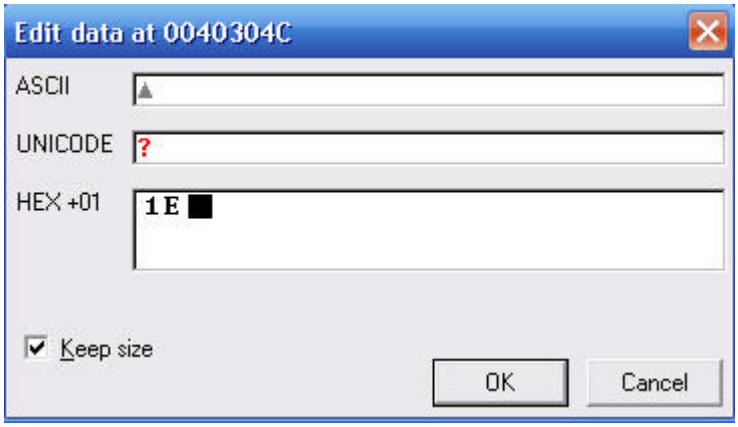
Một cửa sổ nhỏ hiện ra:



Như vậy chúng ta chỉ cần sửa ở mã HEX hay mã ASCII thì sẽ đạt được điều mong muốn, như đã nói tôi muốn "Serial thực" là 30, dùng một công cụ chuyển đổi ta được:

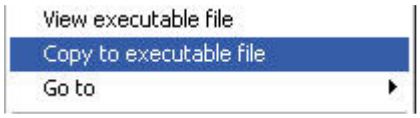


Như vậy chỉ cần thay giá trị 6F thành 1E là được, sửa lại như vậy:

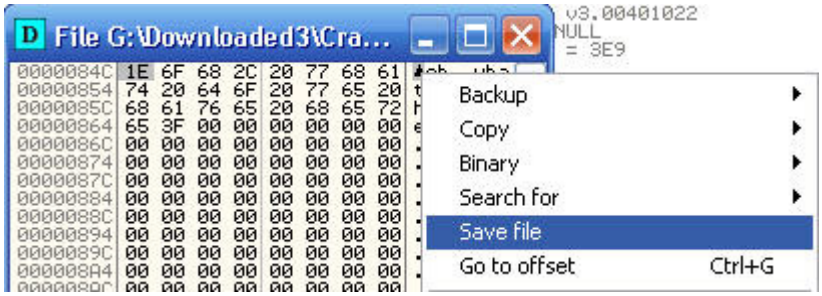


Nhấn OK, ở đây tôi muốn bạn biết rằng:

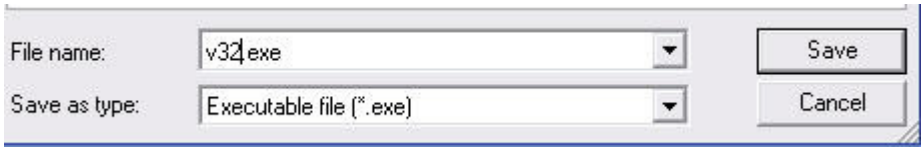
- Khi một Target thực thi, nó dùng một phần bộ nhớ máy tính để làm nơi lưu trữ các file tạm, vì vậy những thứ bạn thay đổi chỉ là thay đổi trong bộ nhớ và điều đó có nghĩa là những thay đổi đó chỉ có giá trị trong lần thực thi đó của Target, để có thể thay đổi trong những lần chạy sau thì ta cần lưu lại thành một file mới, để lưu lại những thay đổi trong Olly ta làm như sau:
-Bôi đen và nhấn chuột phải tại địa chỉ vừa thay đổi giá trị và chọn:



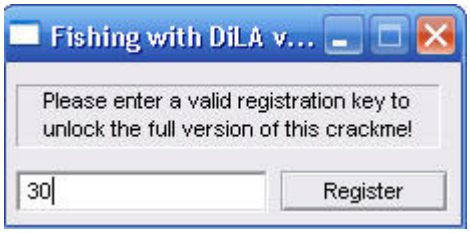
Tại cửa sổ mới hiện ra chọn:



Sau đó điền một tên để lưu lại (nên điền tên khác với Target để không ghi đè):



Nhấn nút Save...
Chạy thử file vừa lưu, điền Serial là 30:



Kiến thức cần nắm:

-Thành thục hơn về kỹ năng set BP
-Ý nghĩa và tác dụng của cấu lệnh CMP, MOV, MOVSX.
-Cách thay đổi một giá trị trong bộ nhớ.
-Cách lưu file trong Olly (một trường hợp)
....

Còn tiếp...

Còn tiếp...

Xin gửi lời Cảm ơn đến:
NhatPhuongLe, 3rr0r, HTS, moth, Benina, hacnho, Unregistered !, Merc, Rongchaua, TQN, Why not Bar, TrickyBoy, **ZOMBIE** nhc1987, ::Xcross87::..., lena151, Registered, www.VnCeRt.info, www.REAOnline.net,
Và những ai đọc tut này...

====P.E Onimusha===
08/09/2007