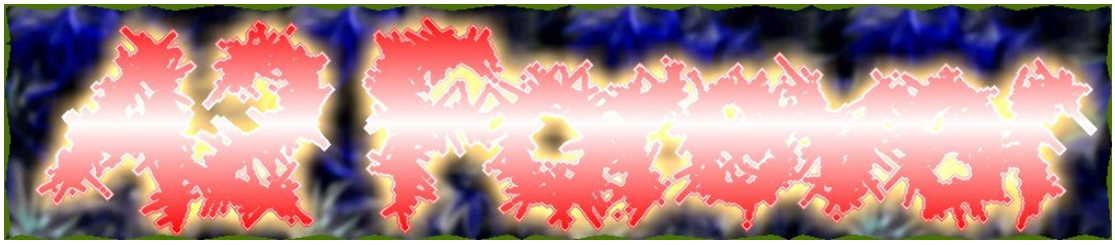


Tut1

Target	fwdv1 Crackme.
Độ khó	Rất cơ bản
Packed	N/A
Công cụ	Olly Dbg 1.10, PeiD 0.93
Mục tiêu	Tìm Serial

Lời nói đầu:

Đâu là một loạt tut nhỏ của tôi, tôi không khẳng định đây là kiến thức tự nhiên tôi có, là bởi đây là những kiến thức cơ bản mà tôi thu lượm được từ các tut khác – nhưng tôi khẳng định tất cả những câu từ trong tut này đều do tôi trình bày lại theo sự hiểu của mình chứ không lạm dụng việc copy – paste những cái có sẵn (nếu có tôi sẽ ghi rõ nguồn), loạt tut này không có mục đích gì khác là giúp cho những ai chưa biết gì về Crack có thể hiểu qua về Crack → từ đó có thể tự tay Crack một phần mềm cho riêng mình, vì kiến thức hạn hẹp nên loạt tut này không có gì là mới mẻ cả, tuy nhiên tôi hy vọng nó có ích cho các bạn, sau khi đọc xong loạt tut này các bạn có thể tìm đọc các tut khác để tự nâng cao kiến thức của mình, thân...

Trong từng tut tôi sẽ luôn cố gắng giải thích những vấn đề thật cơ bản nhưng cần thiết để mọi người có thể hiểu rõ việc mình đang làm và tại sao phải làm như vậy, tôi cũng xin nói rõ với mỗi Target có nhiều cách để Crack tùy thuộc vào khả năng của mỗi người nên mỗi tut này chỉ mang tính tham khảo và tôi không bao giờ đảm

bảo rằng cách mà tôi hướng dẫn bạn là cách crack hay nhất, việc đó còn tùy khả năng hiểu của bạn qua mỗi tut...

Việc thực hành là không thể thiếu với một lính mới, vì điều đó nên vơi mỗi tut tôi luôn kèm theo Target để mọi người không mất công tìm kiếm, bạn không nên quá áp đặt một cách Crack duy nhất, hãy luôn thử đổi mới nếu thấy có thể...

Những từ trong tut này là do tôi tự viết nhưng kiến thức thì do tôi góp nhặt lại từ các tut về Crack khác, việc học Crack theo hướng nắm vững lý thuyết rồi thực hành luôn luôn là hướng tốt nhất nhưng tôi đã và đang học như vậy, thực sự thấy nhiều lúc khá chán nản, tôi mạnh dạn trình bày theo một hướng khác, đó là những lý thuyết sẽ được nói tới trong từng trường hợp cần thiết, việc đó sẽ khiến bạn có hứng thú hơn với những kiến thức mình đang học...

Giới thiệu

Đây là một Target rất cơ bản trong việc Fishing serial: Tức là serial có thể nhìn thấy trong cửa sổ Stack hay một vị trí nào đó trong Olly (một trình Debugger – Disassembler... bạn sẽ tìm hiểu sau), việc fishing serial có thể thực hiện mà không cần biết nhiều ngôn ngữ Assemble (ASM-ngôn ngữ máy, bạn hãy tạm hiểu rằng mọi chương trình đều được dịch ra ngôn ngữ máy khi thực thi), và đó cũng là một lý do khiến Crackme này rất phù hợp cho sự khởi đầu với một người mới biết đến Crack...

Với đa số các target thì trước khi Crack ta thường kiểm tra xem nó có bị Pack bởi một Packer hay được bảo vệ bởi một Protector nào hay không: Việc này thực sự có ích bởi phần nhiều ta rất khó và có khi là không thể Crack được một Target đã bị Pack hay được bảo vệ, nếu việc Pack một Target (không phải dạng nén như WinRAR hay WinZIP) thường nhằm mục đích làm giảm dung lượng nhưng target đó vẫn có thể thực thi thì việc bảo vệ bằng một Protector hiện đại lại thiên về chức năng bảo vệ code của Target, điều này khiến nhiều khi làm tăng đáng kể dung lượng của một Target, như

một Cracker (cụ thể là lena151) có từng nói đã gặp trường hợp một Protector làm tăng dung lượng của một Target lên 6 lần...
Việc kiểm tra và biết được một Target có bị Pack/Protect hay không sẽ giúp chúng ta có những định hướng quan trọng cho quá trình Crack, vì có những ngôn ngữ đặc biệt không thực sự phù hợp với Olly (ví dụ như Visual Basic...), ngoài ra thì còn có nhiều lý do khác mà nếu có thể tôi sẽ nói sau...

Chạy thử:
Muốn Crack được một Target bạn phải tìm hiểu một chút về hoạt động của Target đó (ví dụ như những biểu hiện khi Target khởi động-như một yêu cầu đăng ký chẳng hạn), bởi vì việc đó sẽ mang lại cho chúng ta một số thông tin quý giá (có nhiều khi cần sự tinh ý của bạn) mà bạn sẽ cần để đề ra một hướng Crack hợp lý nhất, những cái tôi khuyên bạn nên chú ý là:

1. Tiêu đề của Target (ví dụ như một soft ABC có dòng chữ trên thanh tiêu đề là "ABC (Unregistered Version)"
2. Những thông báo, những dòng chữ thông báo khi ta đăng ký (ví dụ như "Invalid Serial !") hay những thứ đại loại như vậy, chúng đặc biết có ích với bạn.
3. Những giới hạn của soft khi ta chưa đăng ký (điều này thường cần thiết khi ta Patch một soft), bạn khó có thể biết hết được giới hạn của một soft nếu bạn không thử, nhưng theo tôi bạn chỉ cần tìm những giới hạn liên quan đến nhu cầu sử dụng soft đó của bạn (ví dụ như một soft có hai chức năng là nối và cắt file nhạc, nếu bạn chỉ cần dùng tính năng nối thì bạn có thể không cần tìm các giới hạn về việc cắt file ☺)
4. Tìm trong thư mục cài đặt của soft có những file nào "khả nghi" không (ví dụ như reg.ini), có thể bạn không hiểu rõ về từ "khả nghi", nhưng bạn cứ yên tâm vì cái này bạn sẽ dễ dàng biết được sau khi thực hành với một số Target.

...

Áp dụng vào Target hôm nay:
Thử chạy Target ta thấy như sau:

Đây là một Crackme – vậy Crackme là gì:

Theo tôi, Crackme tương tự như phần đăng ký trong các phần mềm thương mại, tức là một Crackme được viết ra thường không có mục đích gì khác ngoài việc để trở thành một bài thực hành với các Cracker, Crackme thể hiện các dạng khác nhau của quá trình đăng ký một phần mềm thương mại, ví dụ như : tìm serial, loại bỏ Nag (bạn sẽ biết Nag là gì ngay sau đây), tạo keyfile, Patch các giới hạn... đa số các Crackme thường có dung lượng rất nhỏ ☺...

Ta thấy Target này chỉ có một nơi để nhập mã đăng ký→nhiều khả năng Crackme này sẽ chỉ đơn giản là so sánh mã đăng ký ta nhập vào với một hay một vài số mặc định nào đó, ở đây bạn cần biết thêm một số thông tin về các dạng đăng ký:

1. Có hai nơi để nhập: một nơi nhập tên (Name) và một nơi nhập Serial (hoặc gọi là Key), với trường hợp này có các trường hợp xảy ra:
    1.1 Bắt kết nối internet khi đăng ký→ Chúng ta chưa bàn đến cái này vội.
    1.2 So sánh tên và Serial ta nhập vào với một hay nhiều cặp tên và Serial mặc định.
    1.3 Từ tên ta nhập vào soft tính toán theo một công thức nào đó để ra "Serial thực", sau đó so sánh "Serial thực" này với Serial ta nhập vào, nếu đúng thì ta đăng ký thành công – đây là một dạng khá phổ biến ngày nay, (ví dụ có một thuật toán là chuyển chuỗi tên ta nhập sang tất cả ký tự thường để làm Serial → khi đó với tên :P.E Onimusha thì Serial sẽ là p.e onimusha)
2. Có một nơi để nhập: Thường thì như tôi đã nói ở trên, ngoài ra thì có thể là mã nhập vào phải thỏa mãn một số điều kiện nào đó....

3. Dùng KeyFile (yêu cầu một file có dạng ABC.xyz để đăng ký): Dạng này ít dùng nhưng có thể gặp, cái này khá nâng cao nên tôi sẽ chưa bàn tới vội.
...
Chú ý: Ngày nay một số Target hiện đại có thêm một "Danh sách đen" chứa tên hay thông tin của một số tên tuổi hay nhóm Cracker nổi tiếng trên thế giới, nếu tên ta nhập vào xuất hiện trong danh sách này thì kể cả ta nhập đúng mã đăng ký (trong trường hợp 1) soft cũng sẽ không chấp nhận – Với tôi điều này không quá quan trọng vì tôi luôn dùng Nick của mình để đăng ký mà tôi lại là một Newbie→không bao giờ có tên trong "Danh sách đen" đó ☺.

Có thể thấy ngay Target hôm nay thuộc trường hợp 2. → ta nghĩ ngay đến việc Fishing Serial ☺. Hãy thử xem sao:

Thử nhập một số bất kỳ vào ô nhập trống – tôi nhập là 30041991, sau đó nhấn nút Register →một thông báo hiện ra:

Chúng ta gọi những thông báo kiểu như thế này là Nag (nôm na là một hộp thoại) ☺, tôi hy vọng bạn biết đó là một thông báo rằng ta đã nhập sai →chúng ta gọi những thông báo hay những câu có nghĩa tương tự như vậy là Badboy → ngược lại, những thông báo rằng ta đã nhập đúng được gọi là Goodboy...
Thường thì những chuỗi trong các thông báo như vậy có thể tìm được trong danh sách String của một số trình Disassemble, và chúng ta hay lợi dụng điều đó để tìm các chuỗi Badboy hoặc Goodboy, không có nhiều khác biệt giữa Badboy và Goodboy dưới cái nhìn của một newbie (như tôi) nhưng thực sự thì tôi luôn muốn tìm Goodboy hơn là Badboy, vì nhiều khi Badboy có rất nhiều, hơn nữa có trường hợp dù đã nhảy qua Badboy nhưng ta vẫn không đến được Goodboy – Có thể bạn sẽ hỏi nếu chúng ta đã thấy được Goodboy (tức là đã đăng ký đúng) thì cần gì phải Crack nữa...
Nhưng tôi muốn bạn hiểu rằng chúng ta hoàn toàn có thể đoán

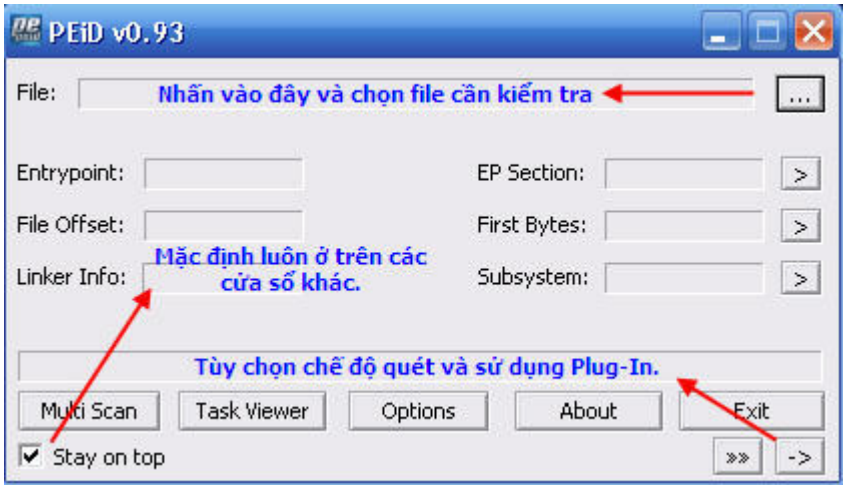
được phần nào của Goodboy mà không cần thấy nó – chỉ cần bạn Crack qua một số Target là cũng có một số kinh nghiệm trong chuyện này rồi (một số từ hay được dùng như: "Thank", "Correct", "Valid", "Registration", "Register" có thể giúp chúng ta thấy được Goodboy ☺).

Chúng ta sẽ cùng thực hành với Crackme hôm nay để nắm vững một số kiến thức cơ bản...

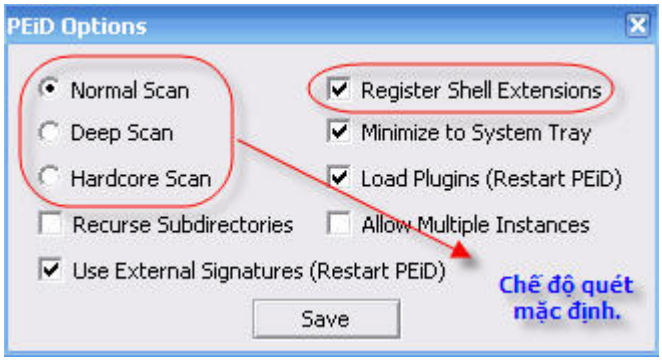
Kiểm tra:

Có khá nhiều công cụ cho phép ta kiểm tra xem một Target có bị Pack/Protect hay không và nếu không thì sẽ cho ta biết ngôn ngữ viết Target đó, một số công cụ phổ biến như PEiD, RDG Packer Detector...Ở đây tôi dùng PEiD.

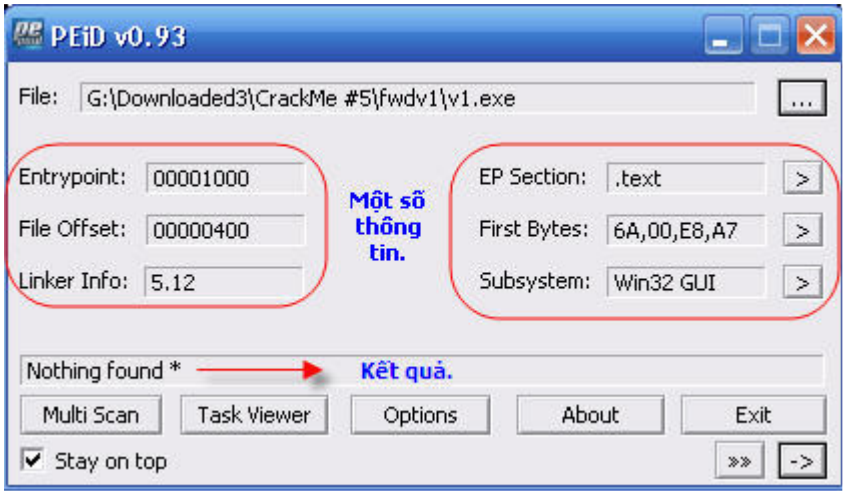
Mở PEiD lên, ta thấy như sau:



Nếu muốn thì ta có thể nhấn nút "..." để mở file cần kiểm tra nhưng để thuận tiện hơn thì ta có thể thiết đặt cho thao tác này xuất hiện trong Menu chuột phải như sau: Chọn mục Options và đánh dấu như sau:



Như trong hình trên ta có thể chọn chế độ quét mặc định cho PEiD, các chế độ đó được sắp xếp theo thứ tự sâu dần, và thường thì ta chỉ cần chọn chế độ Normal.
Sau khi chọn như trên nhấn Save và tắt PEiD đi, nhấn chuột phải vào Target chọn "Scan with PEiD..." ta được kết quả:



Nothing found * → Như vậy có nghĩa là đây là một dạng Packer mà PEiD chưa nhận dạng được, tuy nhiên hãy khoan nói đến điều này và thử dùng tới Olly...

Fishing Serial:

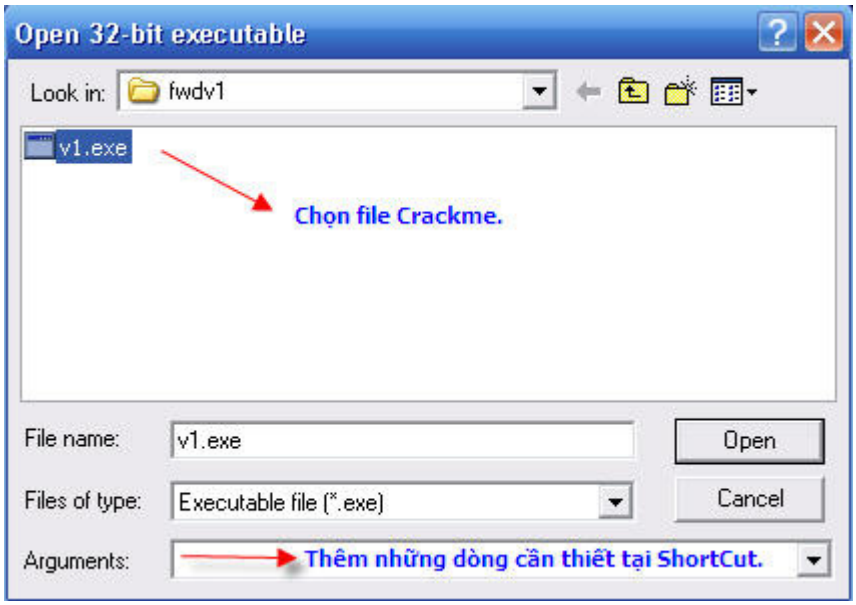
Giới thiệu về Olly:

- **OllDBG** : là 1 chương trình dịch hợp ngữ 32-bit với mức là phân tích gỡ rối (Debugger) trên Windows. Nó phân tích mọi chương trình dưới dạng mã Assembler, với việc phân tích này khiến OllyDbg đặc biệt hữu ích trong các trường hợp chương trình không có tệp tin nguồn . Nó còn cho ta thấy được giá trị của các thanh ghi, các thủ tục, lệnh gọi hàm API, các bảng, hằng số, chuỗi ký tự v.v... Ngoài ra ta còn có

thể ghi chú thích tại các dòng lệnh . Nói chung đây là một công cụ phổ biến được các Crackers ưa dùng nhất...

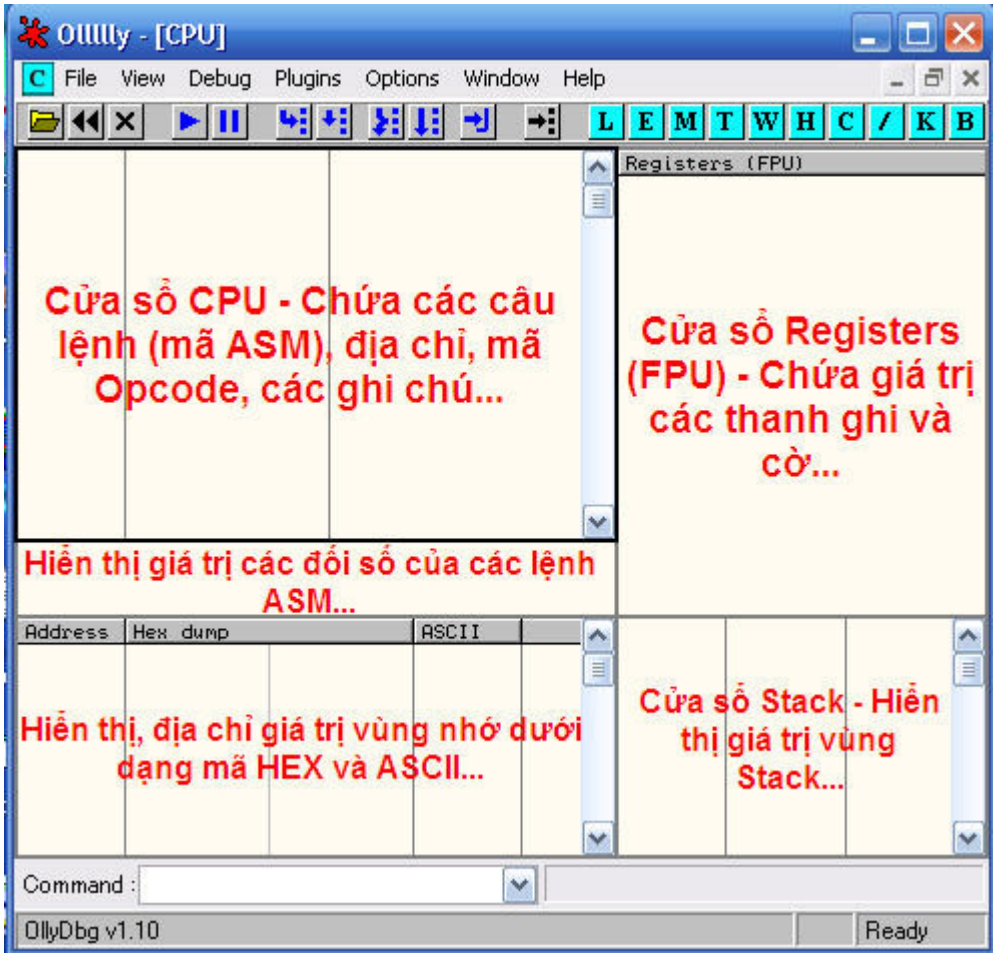
-(Trích Cracking Dictionary)-

Mở Olly lên, nhấn F3 (Open) (hoặc chọn qua Menu, nút bấm), sau đó chọn đến Crackme:

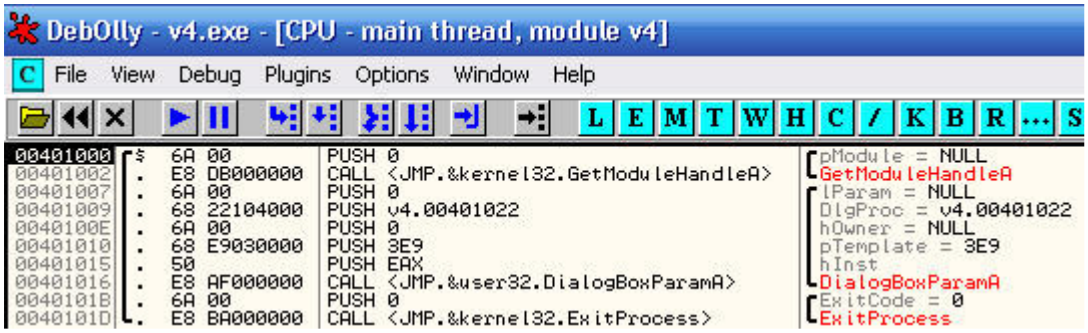


Ở mục Arguments ta có thể thêm những dòng cần thiết nếu cần: Điều này là do có một số Target chỉ có thể chạy thông qua một số câu lệnh hay qua các thông số cần thiết, khi chạy file chính (dạng .exe) trong thư mục cài đặt Target nhiều khi không thể được, vì vậy nếu muốn chạy được những Target kiểu này trong Olly thì ta cần xem ShortCut của Target có những tham số gì đặc biệt trong phần Target hoặc một số dấu hiệu đặc biệt trong quá trình thực thi thì hãy chép vào mục Arguments để có thể Run Target trong Olly bình thường.

Các cửa sổ trong giao diện mặc định của Olly như sau:



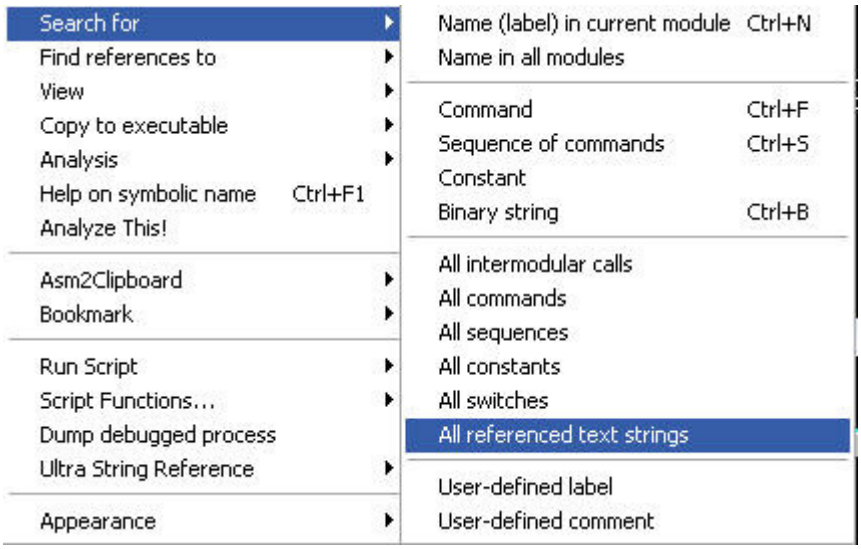
Sau khi nhấn Open và chờ Olly load xong chúng ta ở đây:



Bạn cần biết một số kiến thức:

- Các giá trị dạng 00401000, 00401002... ở cột thứ nhất là địa chỉ các code trong Olly.
- Cột thứ 3 là câu lệnh của Target trong ngôn ngữ ASM.
- Cột thứ 2 là mã của các câu lệnh tại cột thứ 3.
- Cột thứ 4 cho chúng ta một số thông tin như : Các String (chuỗi), các hàm, các sự kiện...

Với Target này thì nó có vẻ dễ dàng vì đoạn code rất ngắn, tuy nhiên tôi muốn bạn làm quen dần với một soft – mà thông thường code sẽ rất dài → sẽ không thể thấy dễ dàng cái ta cần tìm (Goodboy hay đơn giản hơn là Badboy), Olly có một chức năng tìm kiếm String khá hiệu quả, chúng ta làm như sau:
Nhấn chuột phải tại cửa sổ CPU của Olly (tức là cửa sổ chính to nhất theo mặc định) và chọn:



Ở cửa sổ hiện ra sau đó ta thấy như sau:

Đúng là Crackme này rất nhẹ, có tất cả 4 String mà Olly tìm được → Chúng ta không cần dùng đến chức năng tìm kiếm String của Olly nên tôi sẽ không nói trong tut này.
Nhìn vào String tôi hy vọng bạn biết được đâu là Goodboy và đâu là Badboy (bạn hãy xem như đây là bài tập về nhà nhé), ngoài ra còn có String là dòng tiêu đề của Crackme – đó là dòng nào ?
Để đến địa chỉ chứa String ta muốn thì chỉ cần nhấn đúp chuột vào dòng String đó...

Nhưng tại sao chúng ta phải tìm Badboy (hay có trường hợp là tìm Goodboy)?

- Là bởi vì bất cứ một hành động nào của Target cũng được thực thi dựa vào một hay các câu lệnh nhất định, để hiện một Badboy, thì thường sẽ cần Tiêu đề (Title), Kiểu (Style),

và Chuỗi cần hiển thị (Text), và như vậy khi hiển thị Badboy thì Target sẽ thực thi những câu lệnh cần thiết, nếu ta tìm được các câu lệnh thì ta sẽ ở rất gần quá trình kiểm tra – so sánh – tạo Serial...

Trong ví dụ trong Visual Basic ta có một câu lệnh như sau:

MsgBox "Welcome to VnCeRt Forum!", vbInformation, "Hello!"

Trong đó MsgBox là câu lệnh hiển thị một MessageBox (bạn chưa cần quan tâm), Welcome to VnCeRt Forum! Là chuỗi sẽ hiển thị trong MessageBox đó, vbInformation sẽ tạo một dấu chấm than "thân thiện" trong MessageBox, còn Hello! Là tiêu đề của MessageBox. Thực thi câu lệnh này chúng ta có kết quả:

Tôi hy vọng bạn có thể hiểu phần nào những gì tôi đang nói ☺.

Như vậy việc tìm Badboy sẽ giúp chúng ta biết làm thế nào để qua Badboy đó (bằng lệnh nhảy – sẽ tìm hiểu sau) → ít nhất muốn đến được Goodboy (tức là thành công) thì ta cũng phải qua tất cả các Badboy, từ đó có thể có hướng Crack hợp lý. Thường thì có nhiều cách để tìm Badboy (Goodboy) và khi ta đã tìm được Badboy thì việc Crack sẽ dễ dàng hơn.

Nhấn vào dòng "Sorry, wrong code!" ta trở lại cửa sổ CPU ở vị trí này:

Một chút kiến thức về lệnh nhảy:

-Lệnh nhảy chia ra làm 2 nhóm là lệnh nhảy có điều kiện và lệnh nhảy không có điều kiện:

- Lệnh nhảy có điều kiện (JE, JNE, JNZ, JL...) sẽ thực thi khi thỏa mãn hay không thỏa mãn một điều kiện nào đó.
- Lệnh nhảy không có điều kiện (JMP) là lệnh nhảy luôn thực thi, chỉ cần Target chạy đến câu lệnh nahyr đó thì nó sẽ nhảy.

- Vậy nhảy là gì – hãy xét đoạn code ví dụ sau:

```
1111    JMP 3333
2222    ...
3333    ...
```

Trong đoạn code trên thì 1111, 2222, 3333 là địa chỉ các câu lệnh. Nếu bình thường thì khi chạy một Target sẽ đi qua lần lượt các câu lệnh, tức là sẽ đi từ 1111→2222→3333, nhưng ở đây có một lệnh nhảy không có điều kiện ở địa chỉ 1111 và sẽ nhảy đến 3333 – Vì đây là lệnh nhảy không có điều kiện nên đoạn code trên sẽ được thực thi theo thứ tự 1111→3333.

Trở lại với Target hôm nay:

Có thể thấy ngay trên Badboy có một lệnh nhảy JE (Jump if Equal - nghĩa là nhảy nếu bằng - đây là một lệnh nhảy có điều kiện-tức là sẽ nhảy nếu thỏa mãn hay không thỏa mãn một điều kiện nào đó - tôi sẽ nói về vấn đề này sau nếu có thể) sẽ nhảy qua Badboy (ta biết dựa vào địa chỉ của Badboy và địa chỉ mà câu lệnh nhảy đến), cái chúng ta cần tìm là điều kiện để lệnh nhảy này thực thi→Đồng nghĩa với việc ta đến được Goodboy (với target này thì là như vậy).

Chúng ta lại thấy rằng trên lệnh nhảy JE có một câu lệnh CMP EAX, 29AH (CMP X, Y là một câu lệnh để so sánh hai giá trị X và Y có bằng nhau không (CMP = Compare) → nếu bằng nhau (tức là giá trị của EAX = 29A – đây là giá trị ở hệ HEX, bạn có thể thấy chữ H ở sau giá trị này) thì ta sẽ đạt được Goodboy ☺...
Nếu bình thường thì có lẽ tôi sẽ lên đầu function (bạn không cần quan tâm từ này vội ☺) và đặt một BP tại đó hay chí ít cũng đặt BP tại hàm Call ngay trên câu lệnh CMP ở trên... tuy nhiên với Crackme hôm nay thì việc đó là không cần thiết, chúng ta sẽ cùng đặt một BP tại câu lệnh CMP EAX, 29AH (nhấn F2), sau khi làm vậy ta thấy như sau:

```
00401048 |.  3D 9A020000 |CMP EAX,29A
```

Kiến thức về BP (viết tắt của BreakPoint):

BreakPoint: Có nghĩa là điểm dừng (hay là điểm ngắt), khi một Target thực thi, nó lần lượt thực thi các câu lệnh theo sự lập trình trước, việc set BP tại một địa chỉ trong Olly (hay một số trình Debug khác) nhằm dừng quá trình thực thi tại một địa chỉ mong muốn (chính là địa chỉ Set BP), thông thường nếu không có bất kì lỗi nào (hay sự kiện đặc biệt nào) xảy ra trong quá trình thực thi của Target thì Target sẽ dừng tại BP mà ta đã Set. Khi ta muốn xem xét kỹ hơn tác dụng, vai trò của một code hay một đoạn code của một Target, ta sẽ set BP tại Code hay đầu đoạn code đó
→Nếu Olly dừng lại ta sẽ làm một số thao tác mà tôi sẽ nói sau ☺.

Bây giờ hãy thử Run Target và xem xét...
Nhấn F9 (Run) Target, sau đó đăng ký-tôi vẫn dùng "30041991", điền xong nhấn nút Register và...Ta thấy Target có dừng lại (giống như bị treo ☺), đó là vì Olly đã dừng tại BP ta vừa đặt, quay trở lại Olly và thấy:

```
00401048 |.  3D 9A020000 |CMP EAX,29A
```

Nhìn sang bên phải tại cửa sổ Registers (FPU) ta thấy như sau:

```
Registers (FPU)                    <
EAX 01CA6787          EAX = 1CA6787
ECX 0000D49B
EDX 00000000
EBX 00000000
ESP 0012FB0C
EBP 0012FB0C
ESI 00401022 v1.00401022
EDI 0012FB74

EIP 00401048 v1.00401048
```

Kiến thức về cửa sổ Register:

- Các thanh ghi là nơi lưu giữ thông tin của bộ xử lý. Chúng được phân loại theo chức năng tương ứng (tổng cộng 13 thanh ghi)

  I. Thanh ghi dữ liệu: AX, BX, CX, DX (Với Windows được mở rộng thành EAX, EBX, ECX, EDX với 32 bits):
  - Như tên gọi chúng sử dụng vào việc thao tác dữ liệu (thực hiện nhanh hơn so với thao tác trực tiếp

trên bộ xử lý).
- Các thanh ghi được chia thành 2 byte cao/thấp tương ứng H/L (như AL là byte thấp, AH byte cao).
-(Trích **LÝ THUYẾT ASSEMBLY**)-

Như vậy là Target so sánh EAX lúc này = 1CA6787 với giá trị 29A, nếu bằng thì ta sẽ đến được Goodboy...

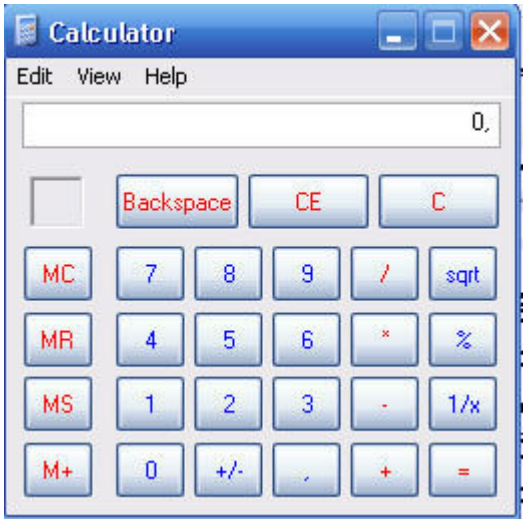
Kiến thức cơ bản:

DEC: Là hệ đếm cơ số 10 (thập phân), gồm các chữ số {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, đây là hệ số thường được dùng trong chương trình học cũng như trong cuộc sống.
HEX: Là hệ đếm cơ số 16 (thập lục phân), gồm các chữ số và 6 chữ cái {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}, nếu xét các ký tự này theo hệ HEX thì:
A=10
B=11
...
F=15

Có nhiều cách để chuyển một giá trị từ DEC→HEX và ngược lại...

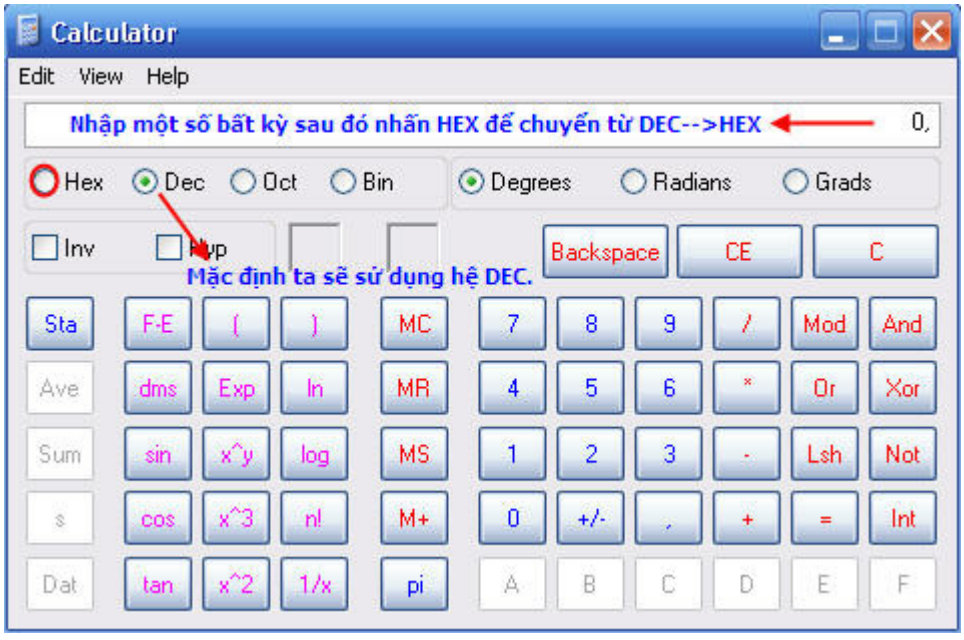
Vậy bạn có biết 1CA6787 là gì không ? Có thể bạn không biết nhưng tôi thì chắc chắn biết vì đó là số 30041991 (mã DEC) chuyển sang mã HEX, bạn đừng nghĩ cái này quá cao siêu gì cả, đơn giản là vì có rất nhiều Target hiện nay thường chuyển chuỗi Key ta nhập vào sang mã HEX rồi mới bắt đầu một quá trình tính toán hay so sánh, cái này chỉ cần làm qua một số Target bạn sẽ biết thôi, nên nếu bạn chưa hiểu thì cứ yên tâm, nó không quá khó đâu ☺.
Vì vậy tôi cũng khuyên bạn nên chọn một con số nào đó làm Key mặc định của mình khi crack một Target – nó có thể theo bạn trong suốt sự nghiệp Crack của mình, ngoài việc nhớ số này thì bạn cũng cần nhớ cả giá trị của số đó khi chuyển sang HEX nữa, điều đó sẽ giúp bạn rất nhiều sau này bởi bạn có thể dễ dàng nhận ra khi nào Target đụng tới Key của ta...
Để chuyển đổi một số từ mã DEC sang mã HEX thì chúng ta có thể dùng ngay máy tính Windows để thực hiện, cách làm như sau:

Sau khi mở máy tính lên (vào Start → Run → Gõ vào calc), mặc định ta thấy như sau:



Nhưng đây chỉ là những tính năng cơ bản, nếu muốn nâng cao hơn thì bạn chọn như sau:

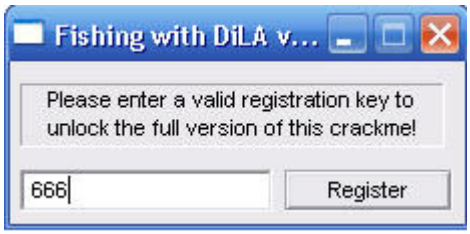
Sau khi chọn, ta được:



Thử nhập 30041991 sau đó nhấn mục HEX ta được kết quả là 1CA6787 ☺

Trở lại với việc Crack, ta thấy rằng nếu hiểu một cách đơn giản thì Target chỉ lấy mã HEX của số ta nhập vào và so sánh vơi số 29A cũng ở hệ HEX-Nếu bằng là xong, như vậy thì chỉ cần đổi ngược lại 29A từ HEX→DEC là ta sẽ có Key đúng, thực hiện với máy tính Windows ta được kết quả:

Vậy thì Key là 666 ☺, mở crackme lên và thử đăng ký lại với key = 666:



Vậy là xong...

Kiến thức cần nắm:

Nếu bạn thực sự là lính mới (như tôi) thì qua tut này tôi muốn bạn nắm được một kiến thức cơ bản sau:

- Thứ tự các thao tác khi Crack một soft.
- Định nghĩa, vai trò của việc Pack hay Protect một Target.
- Các khái niệm về Nag, Badboy, Goodboy…
- Cách sử dụng PEiD mức cơ bản.

- Về Olly bạn cần biết :Cách tìm String, cách Set BP trong Olly, một số câu lệnh cơ bản (JE, CMP)…
- Định nghĩa và vai trò của BreakPoint.
- Sử dụng máy tính Windows để chuyển đổi qua lại giữa các hệ cơ số.

…

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====

07/09/2007