|  |  |
| --- | --- |
| Đại học Quốc gia TP Hồ Chí Minh  Đại học Khoa học tự nhiên TP Hồ Chí Minh  Khoa Công nghệ thông tin  Bộ môn MMT & VT | Kiến trúc máy tính & Hợp ngữ  Năm học: 2018-2019  Học kì: 2  GVLT: Lê Viết Long |

**BÁO CÁO ĐỒ ÁN**

**“Đồ án 03”**

# **Thông tin sinh viên thực hiện**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ và tên** | **Email** | **Điện thoại** |
| 1 | 1712863 | Trương Văn Tú | 1712863@student.hcmus.edu.vn | 0978966563 |
| 2 | 1712875 | Nguyễn Mạnh Tuấn | 1712875@student.hcmus.edu.vn |  |
| 3 | 1712874 | Nguyễn Chánh Anh Tuấn | 1712874@student.hcmus.edu.vn |  |
| 4 | 1712881 | Hà Sơn Tùng | 1712881@student.hcmus.edu.vn |  |
| 5 | 1712913 | Nguyễn Trí Vinh | 1712913@student.hcmus.edu.vn |  |

# **Phân chia công việc**

|  |  |
| --- | --- |
| **Tên công việc** | **Người thực hiện** |
| Crack 2\_1 | Trương Văn Tú, Hà Sơn Tùng. |
| Crack 2\_2 | Nguyễn Chánh Anh Tuấn, Nguyễn Trí Vinh. |
| Crack 2\_3 | Nguyễn Mạnh Tuấn. |

# **Đánh giá tổng quan đồ án**

## Chức năng làm được:

## + Crack thành công cả 3 crackme. Chỉ ra đoạn code phát sinh key của 2\_1 và 2\_2

## + Cung cấp được các username và pasword mẫu.

## + Viết được keygen cho 2 bản 2\_1 và 2\_2

## + Có file thực thi việc sinh keygen cho crackme 2\_3 (Sưu tầm)

## + Mở đóng gói cho bản crackme 2\_3

## Chức năng chưa hoàn thành được

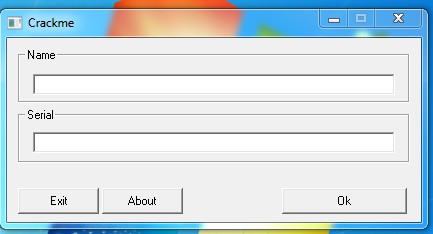
## + Chưa chỉ ra được đoạn code phát sinh key của bản 2\_3

## + Chưa viết được keygen cho bản crackme 2\_3

* Mức độ hoàn thành các yêu cầu: 95%.
* Điểm đề nghị: 9/10.

1. **Chi tiết đồ án**
2. **Crackme 2\_1**

* **Giao diện crackme 2\_1**



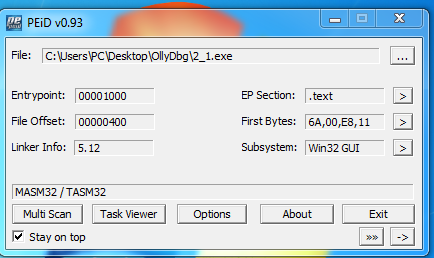
### Username và password mẫu

### + Username: username

+ Pasword:00005F89-00000EFE-0105B2E3-03B48205

### Giao diện khi nhập đúng pasword.

* Kiểm tra bản crackme này với PEid

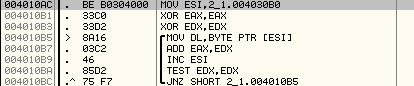


+ Theo thông báo từ PEid thì bản crackme này không bị pack và có khả năng cao là viết bằng MASM32. Nên ta thực hiện load bằng OllyDbg và tiến hành crack nó.

* **Quá Trình phát sinh pasword và kiểm tra pasword nhập vào.**

#### B1: Xử lí Username.

B1.1: Tính tổng các chứ cái của username nhập vào dựa vào mã ascii. Và lưu tổng đó tại vị trí đầu tiên của 1 mảng.



B1.2: Lấy tổng vừa tính được giảm đi 1 (ở hệ hecxa) và sau đó nhân cho 3. Lưu kết quả vừa tính được và phân tử thứ 2 của mảng.

B1.3: Lưu 1 biến có giá trị 12345678 (ở thể hecxa).

B1.3.1: Lần lượt lấy số đó xem xor với các phân tử của Username được người dùng nhập vào. Kết quả sẽ được lưu trong biến ban đầu.

B1.3.2: Lấy kết quả đó xoay trái 5.

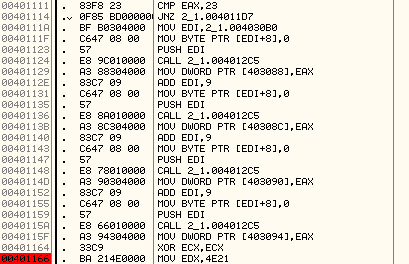
(Cứ liên tục thực hiện như vậy cho đến khi duyệt qua hết các phần tử của username.)

B1.3.3: Lấy kết quả thu được sau tất cả các lần XOR và ROL như trên thì lấy kết quả đó chia cho 7A69 (hệ Hecxa).

B1.3.4: Lưu kết quả phần dư vào mãng ở vị trí thứ 3.

B1.3.5: Lấy kết quả phần nguyễn sau khi thực hiện phép chia đó đem đi cộng với FFF (hệ Hexca) và lưu kết quả vào phần tử thứ 5 của mảng.

#### B2: Xử lí Pasword.

B2.1: kiểm tra xem pasword nhập vào có đử 23 (hệ Hecxa) kí tự hay không nếu không đủ thì không thực hiện xử lí pasword.

B2.2: Thực hiện mỗi lần lấy 8 phần tử của pasword nhập vào (các vị trí 9 18 27 sẽ không được sử dụng). Có 1 biến EAX = 0;

B2.2.1: Lấy lần lượt các phần tử của 8 phần tử password trừ đi cho 30 (hệ Hecxa).

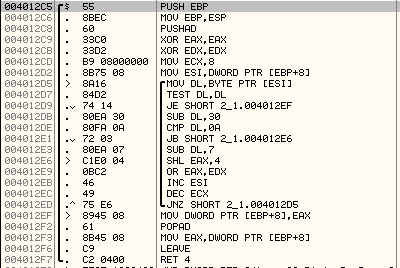
B2.2.2: Kiểm tra xem số đó có lớn hơn hay bé hơn A (hệ Hecxa). Nếu lớn hơn lấy nó trừ đi 7(hệ Hecxa)

B2.2.3: Dịch trái 4 bit của EAX.

B2.2.4: EAX = EAX OR phần tử user đã được xử lí ở trên.

B2.2.5: Làm như vậy cho đến khi hết 8 phần tử của pasword.

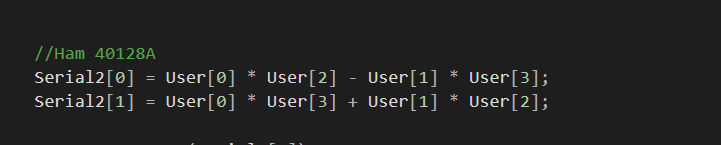
B2.2.6: Lưu kết quả của EAX vào trong phần tử của mảng. Tiếp tục như vậy cho đến khi đủ 4 lần và mỗi lần thì bỏ đi các vị trí đã nếu trên.



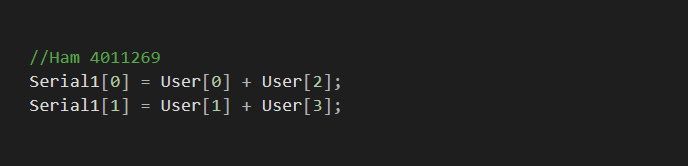
#### B3: Thực hiện việc kiểm tra.

B3.1: Thực hiện hàm 4011F1. Ở tại hàm này thì sẽ xử lí pasword. Bản chất của hàm này là lấy 4 phần tử được xử lí từ username ở trên xử lí trên 2 hàm nhỏ là:

+ 40128A: Hàm này có 4 tham số truyền vào là các phần tử của 4 phần tử được xử lí từ username và lần lượt thực hiện các công việc.



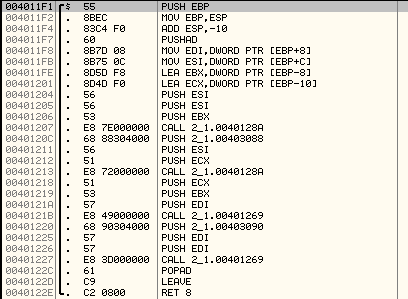
+ 401269: Hàm này có 4 tham số truyền vào là các phần tử của pasword đã được xử lí ở trên và lần lượt thực hiện các công việc.



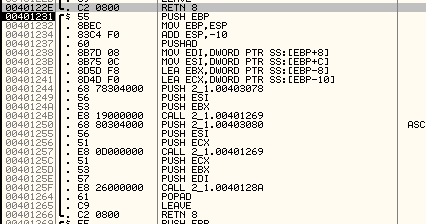
Sau khi kết thúc hàm 4011F1 thì sẽ có được 2 giá trị là

Giá trị 1: 3 \* 3 - 4E1F \* 4E1F + Pasword1 \* 3 - Pasword2 \* 4E1F + Pasword3

Giá trị 2: 3 \* 2 \* 4E1F + Pasword1 \* 4E1F + 3 \* Pasword2 + Pasword4



B3.2: Thực hàm 401231. Ở hàm này thì xử lí 4 phần tử của username đã được xử lí ở bên. Trong hàm nãy bản chất cũng sử dụng 2 hàm nhỏ đã được trình bày ở trên.



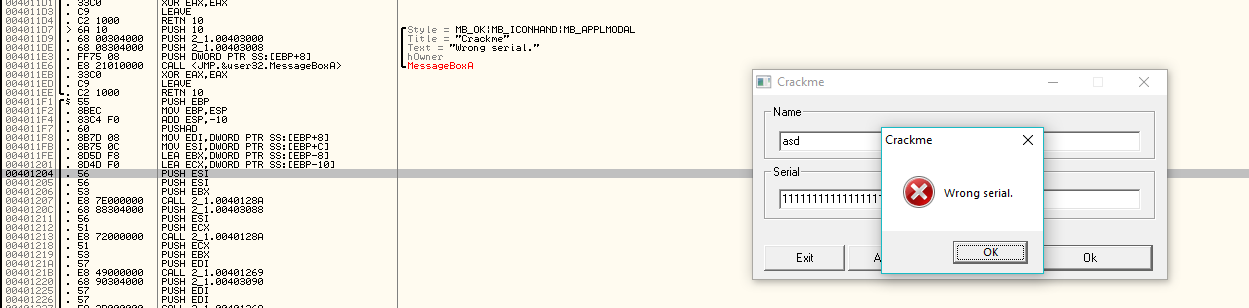
+ Kết quả thu được sau khi thực hiện hàm 401231 cũng là 2 giá trị:

Giá trị 3: (username3+3) \* (username1 +3) – (username4 + 4E1F) \* (username2 + 4E1F)

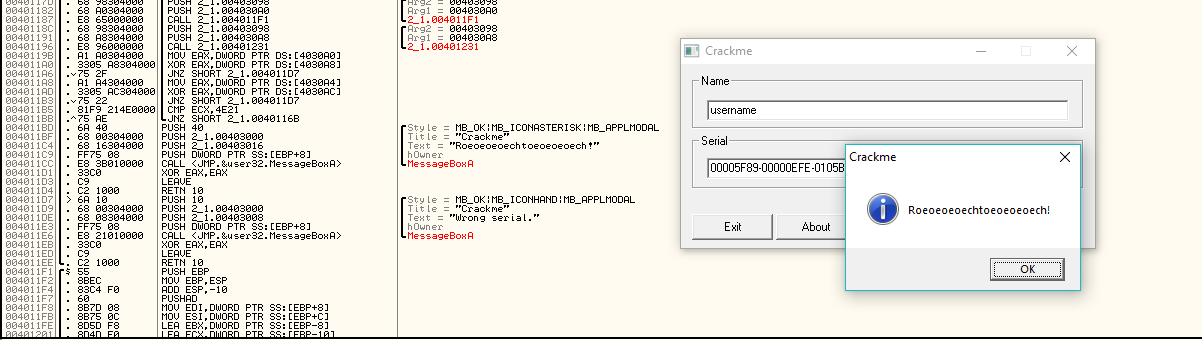
Giá Trị 4: (username3 +3) \* (4E1F + username2) + (username1 + 3) \* (username4 + 4E1F)

+ Sau đó lần lượt kiểm tra các giá trị đã nhận được (Giá trị 1 = Giá trị 3 và Giá trị 2 = Giá trị 4) nếu chúng bằng nhau thì thực hiện tiếp quá trình bắt đầu từ địa chỉ 40116B.

+ Trong quá trình nếu có 1 cặp giá trị nào không bằng nhau thì sẽ nhảy tới Bad boy tại địa chỉ 4011D7.



+ Ngược lại kết thúc quá trình sẽ nhảy tới Goob boy tại địa chỉ 4011BD.



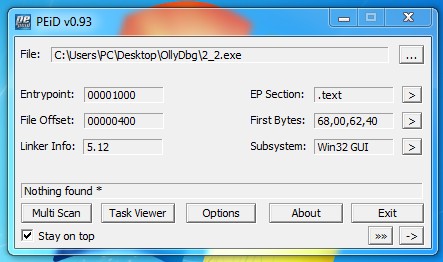
1. **Crackme 2\_2**

*  **Giao diện của Crackme 2\_2**
* **Username và password mẫu.**

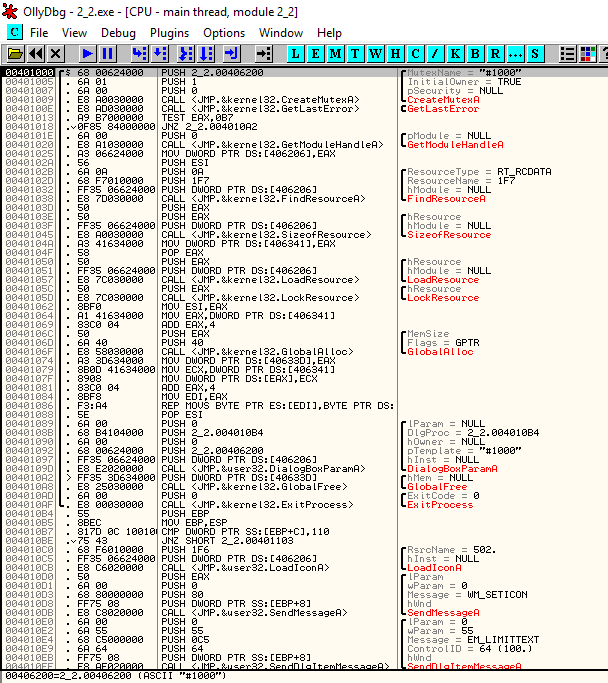
+ Username : username

+ Password : 1577438470



* **Kiểm tra bản crackme này với Pied**

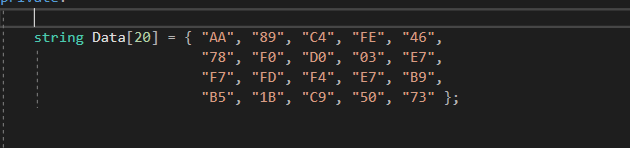
+ Pied thông báo Nothing found ta biết rằng file có thể chưa bị pack. Vậy ta dùng OllyDbg để chạy thử kiểm tra. Kết quả thu được là crackme không bị pack.



* **Quá Trình phát sinh pasword và kiểm tra pasword nhập vào.**

**+** Quá trình này chỉ thao tác từ giá trị username[1] đến username[length] (với user[length] = 0).

Có 1 mảng các giá trị mặc định cho trước luôn không đổi với mọi các username nhập vào giá trị đó là.



+ Quá trình 1: Lần lượt lấy các giá trị của username nhập vào từ trị trí 1 đến length. Quá trình này sẽ được thực hiện dựa theo 5 phần tử đầu tiên của Data[].

I = 0, j=0, n = length.

B1: Lấy giá trị của user[i+1] đem đi xor với Data[j]. giá trị được lưu trong BL

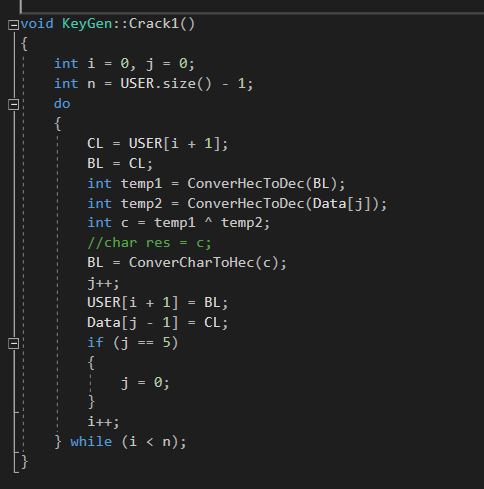
B2: Gán giá trị của user[i+1] = BL

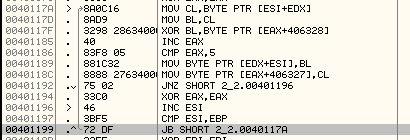
B3: Data[j] = giá trị của username[i+1] ban đầu.

B4: j++, nếu j = 5 thì gán j = 0.

B5: i++;

B6: nếu I < n thì quay lại bước 1.





+ Quá trình 2: Quá trình này sẽ thực hiện trên 5 phần tử tiếp theo của Data[] (từ 5 đến 10) và sẽ thực hiện từ username[length] đến username[1].

I = 0, j = 0, n = length.

B1: BL = Data[5+j], AL = username[n].

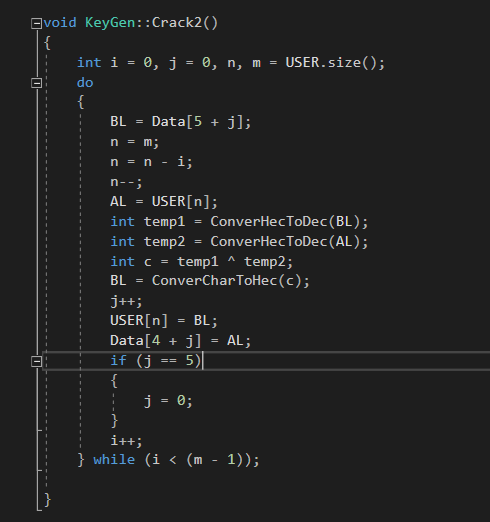
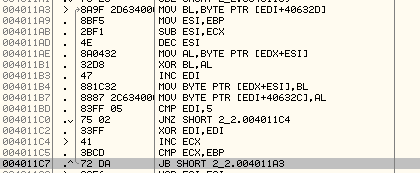
B2: BL = BL^AL

B3: user[n] = BL, Data[5+j] = AL

B4: j++, nếu j=5 thì gán j = 0

B5: i++

B6: nếu i < n quay lại bước 1.



+ Quá trình 3: Quá trình này sẽ thực hiện trên 5 phần tử tiếp theo của Data[] (từ 11 đến 15) và sẽ thực hiện từ username[1] đến username[length].

I = 0, j = 0, n = length

B1: AL = username[i+1], CL=Data[10+j]

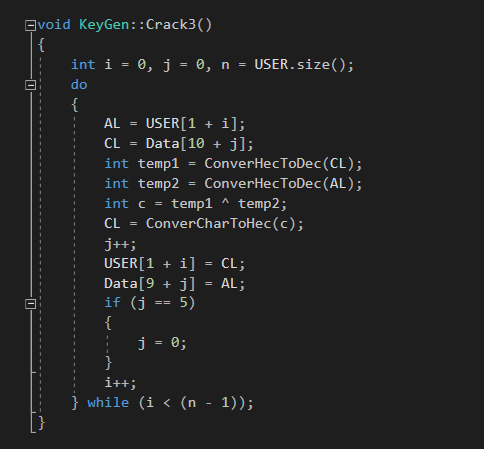
B2: CL= Cl^AL

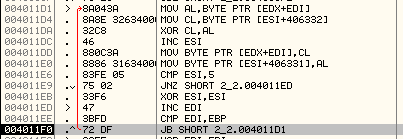
B3 user[1+i] = CL, Data[AL]

B4: j++, nếu j = 5 thì j = 0

B5: i++

B6 nếu i < n thì quay lại bước 1.





+ Quá trình 4: Quá trình này sẽ thực hiện trên 5 phần tử tiếp theo của Data[] (từ 16 đến 20) và sẽ thực hiện từ username[length] đến username[1].

I=0, j=0, n=length.

B1: BL = Data[15+j], AL = username[n].

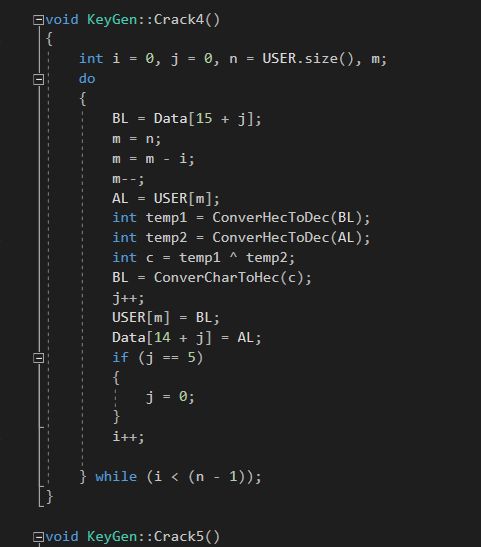
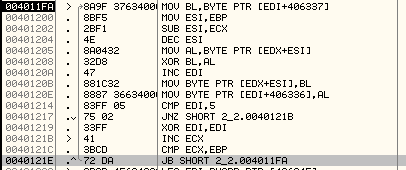
B2: Bl = BL^AL.

B3: User[n] = BL, Data[15+j] = AL

B4: j++, nếu j = 6 thì j=0.

B5: i++

B6: Nếu i < n thì quay lại bước 1.



+ Quá trình 5: quá trình tạo pasword.

Có 1 mảng có 4 phần tử để lưu giá trị nhận được và giá trị ban đầu của mảng là 0.

I = 0, j = 0, n = length.

B1: I = j;

B2: I = i & 3

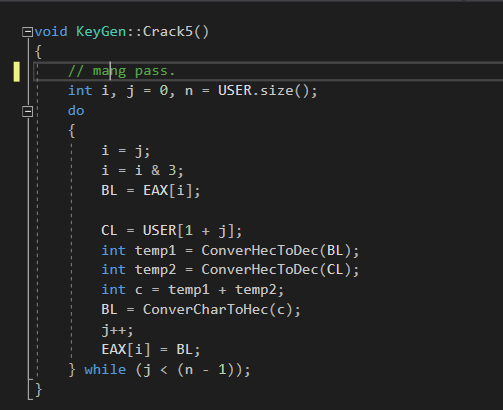
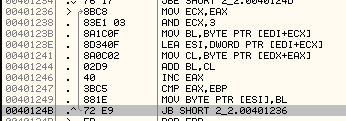
B3: CL = username[1+j], BL = EAX[i]

B4: BL= BL+CL.

B5: EAX[i] = BL

B6: j++

B7: Nếu j < n thì quay lại bước 1.



+ Quá trình 6: Lấy key.

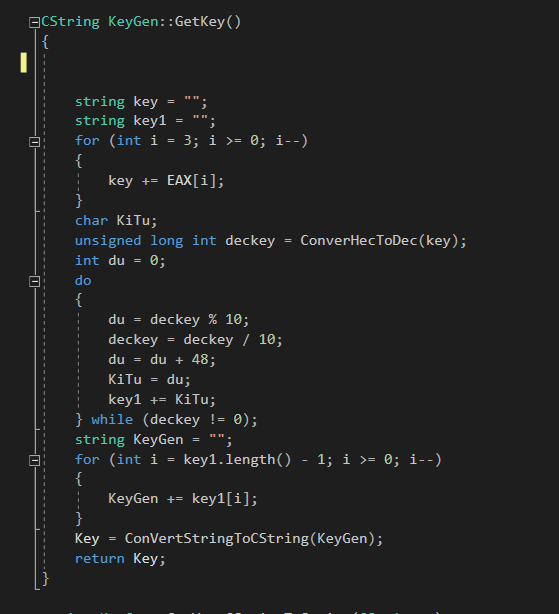
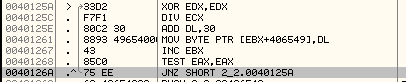
Gán EAX = giá trị của mảng EAX đã tạo ra ở quá trình 5.

B1: Lấy EAX= EAX/10 số dư sẽ lưu trong Dl.

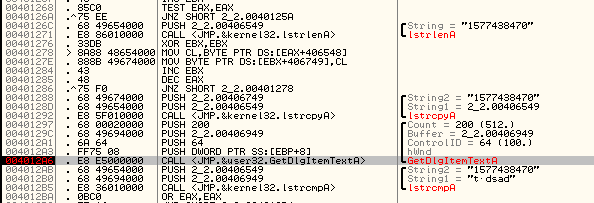
B2: Xử lí DL để DL ra mã ascii và lưu nó lại.

B3: EAX chưa = 0 thì quay lại bước 1.

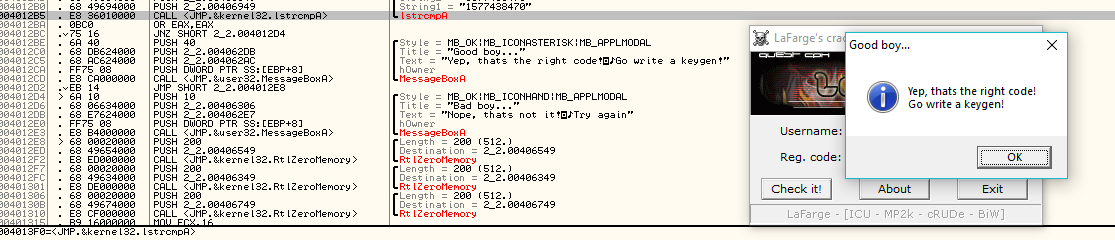
B4: Key sẽ là dãy số được ghi ngược lại từ B2.



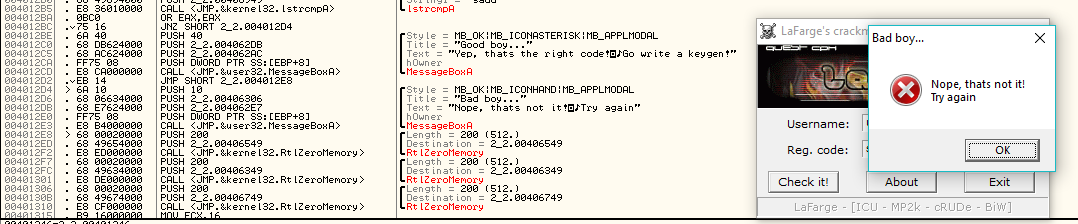
+ Sau khi chạy xong 6 quá trình ra sẽ được Key và so sánh Key nhận được vởi key mà user nhập vào.



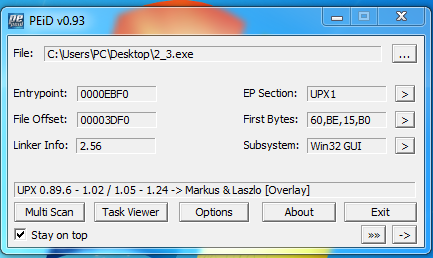
**+** Nếu Key user nhập đúng với Key được xử lí từ user thì chương trình sẽ nhảy tới Good boy tại địa chỉ 4012BE.



+ Nếu Key user nhập sai với Key được xử lí từ user thì chương trình sẽ nhảy tới Bad boy tại địa chỉ 4012D4.



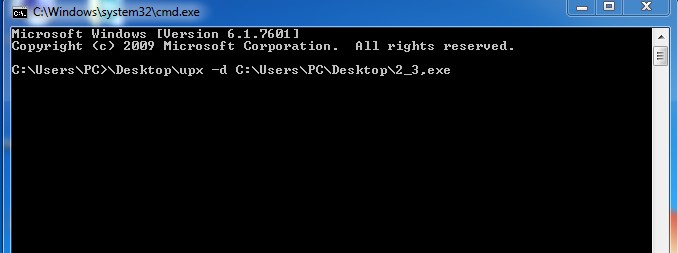
1. **Crackme 2\_3**

* **Kiểm tra crackme bằng pied.**

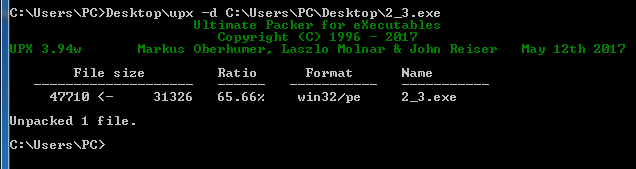
+ File 2\_3 này đã bị đóng gói và cần được unpack và quá trình unpack như sau :

B1: Tải upx394w tại link này: <https://drive.google.com/open?id=1Em2mo3r50wY26QiXYzticaA47kqchkzb>

B2 : Phần mềm giải mã đóng gói này sẽ đươc thực hiện bằng cmd. Gõ lệnh như mình theo cú pháp: Upx -d <tên file exe muốn unpack>



Nếu câu lệnh bạn đúng và file exe của bạn bị upx thì sẽ xuất hiện như thế này là thành công



+ Sau khi unpack được crackme ta dùng OllyDbg để tiến hành tìm key.

* **Giao Diện của Crackme 2\_3**

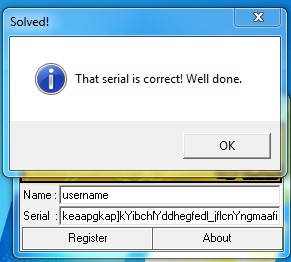


### + Username và pasword mẫu.

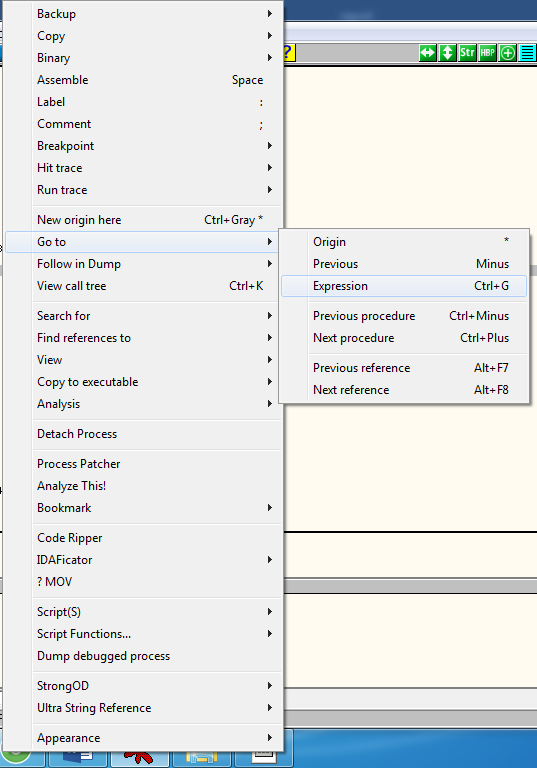
### Username: username

#### Pasword:Zk`gZaaabegghi`o\_f[m]k^keaapgkap]kYibchlYddhegfedl\_jflcnYngmaafi

* **Giao diện khi nhập đúng pasword**

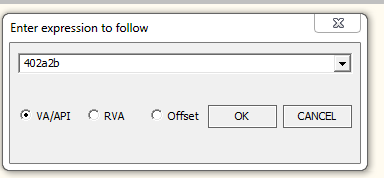


* Hướng dẫn lấy key ở bản crack 2\_3 bằng OllyDbg. (thuật toán quá phức tạp. Nên nhóm em không nói được quá trình phát sinh khóa.)
* Sau quá trình debug và tìm hiểu thì biết được giá trị pasword của crackme 2\_3 được lưu trong vùng nhớ 4063D0 và vùng thực hiện lệnh đó là tại câu lệnh 40253C

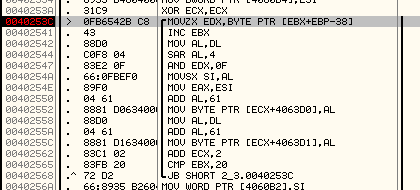
+ Ta thực hiện như sau để có thể lấy được key.

B1: Nháy phải -> goto ->Expreession (hoặc nhấn Ctrl + G)

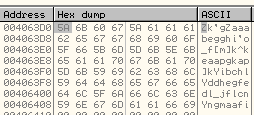
B2: Hộp thoại xuất hiện và nhập 40253C và tiên hành đặt 1 debug tại vị trí đó.



B3: Sau khi đặt 1 debug tại đó thì tiến hành đến địa chỉ 4063D0 như trên để quan sát password được sinh ra.



+ Sau mỗi lần F9 thì sẽ xuất hiện 1 kí tự 4063D0 nên chúng ta chỉ cần ghi lại và cuối cùng sẽ được password được sinh ra từ username. 😊



+ Sau khi chạy hết vòng lặp đó thì ta được kết quả như thế này với dữ liệu username nhập vào là “username”.

+ Bản crackme này quá phức tạp và mình chưa hiểu hết được các câu lệnh trong đó nên không thể hướng dẫ cụ thể phần tìm ra key của bản crackme này. Nên không thể viết keygen.

1. **Tài liệu tham khảo**
   * Tài liệu crackme 2\_3 : <http://crackmes.cf/users/j/cerebus/>