**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**Architecture Computer - Assembly**

### **Đồ án 3:**

**CRACKING SOFTWARE**

By

Vũ Hoàng Dương - 1612127

Lê Quốc Dũng – 1612126

Nguyễn Hữu Điền – 1612104

Nguyễn Hoàng Giang – 1612151

**Teacher’s Name**

Lê Viết Long

**Class**: KTMTHN-2016/4

# Tổng quan

1. Yêu cầu:

* Cơ bản: với mỗi crackme phải chỉ ra các đoạn phát sinh key, ý nghĩa và đưa ra key tương

ứng minh họa

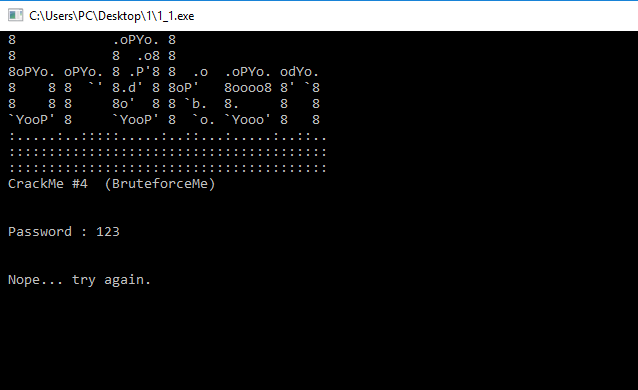
* Nâng cao: Viết chương trình keygen (bằng bất cứ ngôn ngữ nào) để phát sinh key từ username nhập vào.

2. Đề bài:

(7+6+4+1) mod 3 +1=1

# CRACKME PROGRAM 1

Chạy file 1\_1.exe, nhập vào 1 key bất kì ta được dòng thông báo



Dựa vào dòng thông báo ta dễ dàng tìm được Goodboy và Badboy

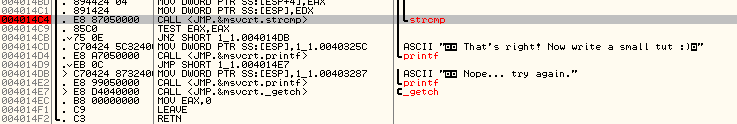
-Goodboy



-Badboy



Ta thấy trên Goodboy có lệnh Test EAX,EAX. Trên lệnh Test ta thấy có lời gọi hàm Call. Đặt BP ở đó:

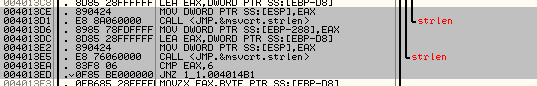


Tiến hành nhấn F9 và chạy bằng 1 key bất kì

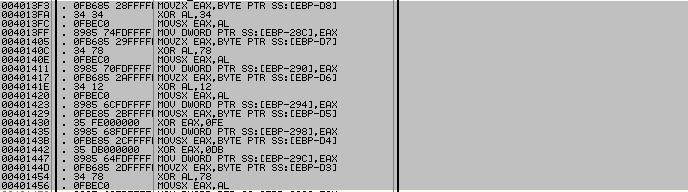
Nhấn F7 để mở các gói hàm



Thuật toán kiểm tra Key



Chiều dài Key do người dùng nhập vào sẽ được so sánh với 6. Nếu độ dài khác 6 sẽ đi đến Badboy và kết thúc. Nếu độ dài Key bằng 6 sẽ tiến hành kiểm tra Key



Quy luật của Key

Lấy lần lượt mỗi lần 2 kí tự của Key cứng ở dạng Hex

Key cứng dạng Hex sẽ có tổng cộng 12 kí tự

-Lần 1: XOR với 0x34

-Lần 2: XOR với 0x78

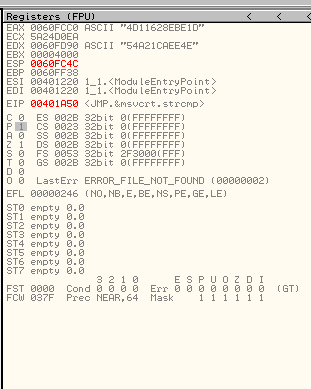
-Lần 3: XOR với 0x12

-Lần 4: XOR với 0x0FE

-Lần 5: XOR với 0x0DB

-Lần 6: XOR với 0x78

Sau khi nhập thử Key là 123456, quan sát ở khung Registers (FTU) ta tìm được Key cứng



Key cứng là: 4D11628EBE1D

Tiến hành XOR, các giá trị đổi được dưới dạng mã Hex sẽ quy về mã ASCII

0x4D XOR 0x34 =0x79 => ‘y’

0x11 XOR 0x78 = 0x69 => ‘i’

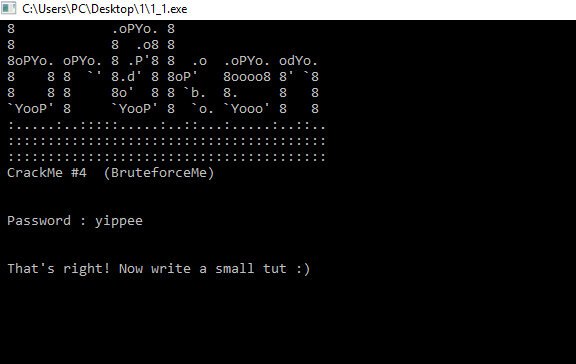
0x62 XOR 0x12 = 0x70 => ‘p’

0x8E XOR 0x0FE = 0x70 =>’p’

0xBE XOR 0x0DB = 0x65 =>’e’

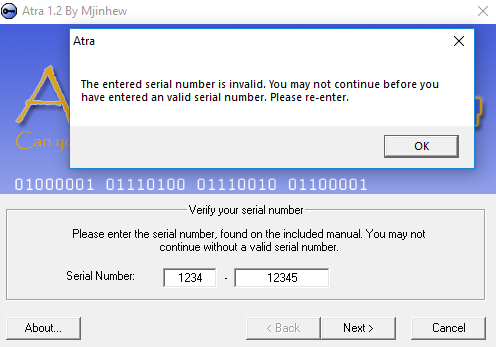
0x1D XOR 0x78 = 0x65 =>’e’

Key cần điền là ‘yippee’. Không có KeyGen



# CRACKME PROGRAM 2

Chạy file 1\_2.exe. Nhập vào 1 key bất kì. Ví dụ: 1234 12345



Dựa vào dòng thông báo, ta dễ dàng tìm được Goodboy và Badboy

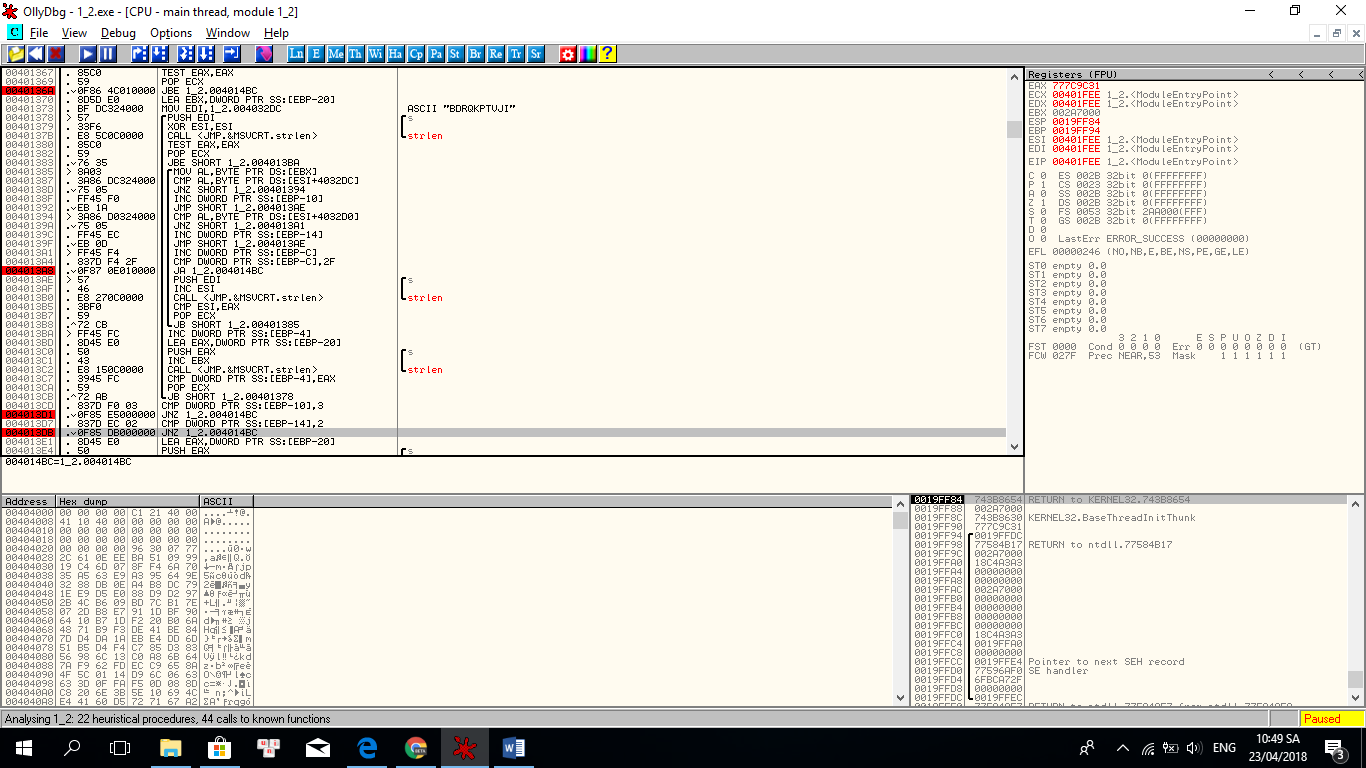
-Goodboy

-Badboy



Để mở cửa sổ thông báo nhập sai Serial Number (Badboy), lệnh 004014 BC 6A00 PUSH 0 cần được thực hiện. Trên toàn bộ chương trình có 5 lệnh nhảy đến lệnh đó: 00401325, 0040136A, 004013A8, 004013D1, 004013DB.

Tiến hành đặt BP tại các vị trí trên.



**Xét đoạn lệnh 0040136A:**

00401367 85C0 TEST EAX,EAX

0040136A 0F86 4C010000 JBE 1\_2.004014BC

Ta có JBE nhảy khi CF = 1 hoặc ZF = 1. Còn lệnh TEST đưa CF = 0, còn ZF = 1 nếu EAX AND EAX = 0 (có nghĩa là EAX = 0) và ZF = 0 khi ngược lại. Do đó, ZF = 1 khi EAX = 0, hay nói cách khác, JBE chỉ nhảy khi EAX = 0.

**Xét đoạn lệnh 004013D1:**

004013CD 837D F8 03 CMP DWORD PTR [EBP-10],3

004013D1 0F85 E5000000 JNZ 1\_2.004014BC

Theo các lệnh ở trên thì đoạn lệnh này so sánh [EBP-10] với 3, với [EBP-10] là số lượng ký tự trong chuỗi serial thứ nhát mà xuất hiện trong chuỗi "BDRQKPTVJI".

**Xét đoạn lệnh 004013DB:**

004013D7 837D EC 02 CMP DWORD PTR [EBP-14],2

004013DB 0F85 DB000000 JNZ 1\_2.004014BC

Theo các lệnh ở trên thì đoạn lệnh này so sánh [EBP-10] với 2, với [EBP-10] là số lượng ký tự trong chuỗi serial thứ nhất mà xuất hiện trong chuỗi "0123456789".

Suy ra: Chuỗi serial thứ nhất là hợp lệ khi có 3 ký tự trong chuỗi "BDRQKPTVJI" và 2 ký tự trong chuỗi "0123456789"

Giờ xét tới chuỗi serial thứ hai:

Qua các lệnh trong đoạn từ lệnh

004013E1 8D45 E0 LEA,DWORD PTR [EBP=20]

tới lệnh

004014BD 0068 C0 ADD BYTE PTR [EAX=40],CH

thì ta hiểu cách sinh serial number thứ hai như sau:

• Gán giá trị tại các địa chỉ 0012F9BC, 0012F9C0, 0012F9C4, 0012F9C8 thành 0x64752301, 0xEFCDAB89, 0x98BADCFE, 0x10325476

• Thực hiện một loạt các phép biển đổi giá trị các biến trên, kết hợp với các dữ liệu cố định trong vùng nhớ từ 0012F9D4 tới 0012FA13 (5 byte đầu của vùng nhớ này là chuỗi serial thứ nhất, byte kế tiếp mang giá trị 0x80, byte thứ 56 mang gía trị 0x28, các bayte còn lại của vùng nhớ mang giá trị 0x0)

• Sau khi thực hiện, chép kêt quả của 4 vùng nhớ trên lần lượt vào 0012FA5C, 0012FA60, 0012FA64, 0012FA68

• Lấy xor của 4 giá trị tại 4 vùng nhớ vừa được chép vào

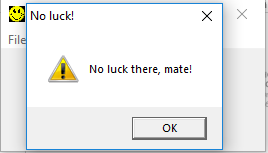
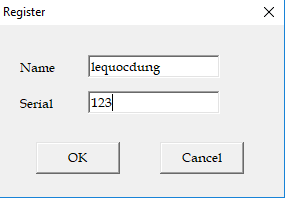
• Chuyển giá trị này sang dạng chuỗi (ở hệ hexa)

• Thực hiện tiếp các lệnh biến đổi từng byte của chuỗi (shift left, xor, ...) để tạo thành chuỗi kết quả (serial number thứ hai) (có sử dụng các dữ liệu trong vùng nhớ cố định).

**Chương trình KeyGen**: lấy dữ liệu thô và xử lí, sau đó kết quả là các chuỗi serial number được lưu thành từng dòng trong file output.txt

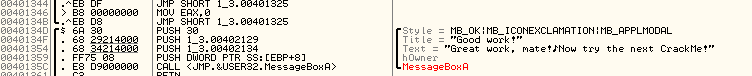
# CRACKME PROGRAM 3

Mở 1\_3.exe chạy thử với một username và password bất kì. Ví dụ: lequocdung – 123

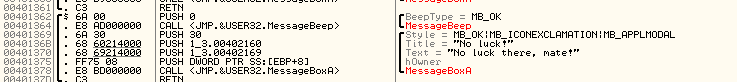


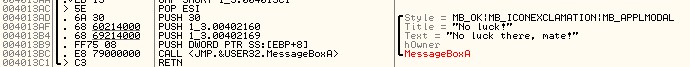
Ta dễ dàng tìm được Badboy và Goodboy

-Goodboy

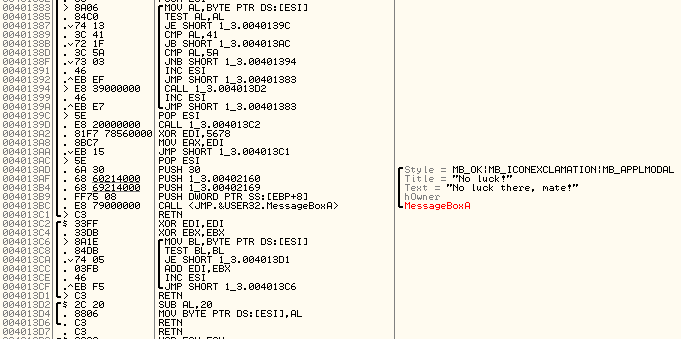


-Badboy





**-Quy luật của Username:**



-Với Username được nhập vào, sẽ được kiểm tra rằng tất cả các kí tự trong chuỗi Username phải có mã ASCII lớn hơn 65



Với các kí tự trong chuỗi Username nhập vào, nếu có mã ASCII lớn hơn 90 sẽ bị giảm đi 32 (mã ASCII)

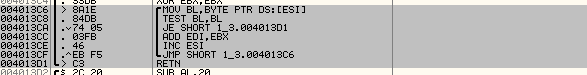
(So sánh mã ASCII của kí tự với 90)



(Trừ mã đi 32)



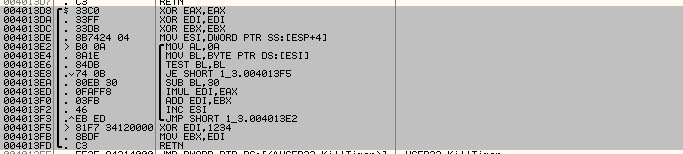
Tiến hành cộng mã ASCII của từng kí tự trong chuỗi tạo thành 1 số



Đem số tìm được xor với 0x5678 (22136)



**-Quy luật Password**:



-Với mỗi kí tự trong Password, trừ mã ASCII của chúng đi 48



-Cộng mã ASCII của từng kí tự trong chuỗi Password tạo thành 1 số



-Đem số tìm được xor với 0x1234 (4660)



**=> Cặp username và password hợp lệ khi hai kết quả tìm được khớp với nhau.**

**Chương trình KeyGen**: nhập vào 1 username bất kì, đưa ra password tương ứng.

Ví dụ:

Username: lequocdung

Password tìm được: 18107



