đại học quốc gia tphcm  
trường đại học khoa học tự nhiên  
khoa công nghệ thông tin  


**Báo cáo đồ án**  
**Đề tài: Reverse Engineering**

**Môn học: Hệ thống máy tính**

*Sinh viên thực hiện:*

|  |  |
| --- | --- |
| Nguyễn Minh Hiếu | 23120124 |
| Huỳnh Mạnh Tường | 23120119 |
| Nguyễn Thanh Phong | 23120111 |

*Giáo viên hướng dẫn:*  
Lê Viết Long

# Giới thiệu

Đồ án Reverse Engineering giúp rèn luyện kỹ năng phân tích mã máy và tìm hiểu cách thức hoạt động nội tại của các phần mềm thông qua các bài tập dạng *KeygenMe*. Mỗi chương trình KeygenMe yêu cầu người dùng nhập một cặp giá trị gồm *username* và *key*. Nếu key hợp lệ, chương trình cho phép tiếp tục; ngược lại sẽ báo lỗi. Việc phân tích ngược nhằm xác định quy tắc kiểm tra key và xây dựng lại thuật toán sinh key tương ứng.

Quá trình thực hiện bao gồm việc sử dụng các công cụ hỗ trợ như **OllyDbg**, **IDA Pro** để dịch ngược mã thực thi, đọc hiểu mã hợp ngữ, xác định thuật toán kiểm tra key, sau đó xây dựng lại chương trình sinh key (keygen) bằng một ngôn ngữ lập trình bất kỳ.

Đồ án không chỉ củng cố kiến thức về kiến trúc máy tính và hợp ngữ, mà còn tạo cơ hội áp dụng kỹ năng phân tích, tư duy logic và hiểu sâu hơn về bảo mật phần mềm ở mức thấp.

# Phân tích thuật toán và phát triển chương trình keygen

## File 1

### Mô tả thuật toán phát sinh khóa

### Chương trình phát sinh khóa

## File 2

### Mô tả thuật toán phát sinh khóa

### Chương trình phát sinh khóa

## File 3

### Mô tả thuật toán phát sinh khóa

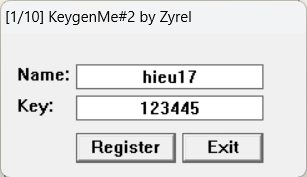
Tải tệp 3.exe vào công cụ OllyDbg, sau đó sử dụng chức năng *Search for  All referenced text strings* để lọc các chuỗi ký tự đáng chú ý. Một trong những chuỗi có thể thu hút sự quan tâm được minh họa như sau:

image

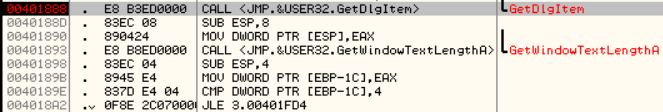
Di chuyển đến dòng mã chứa chuỗi trên, có thể quan sát thấy các lệnh gọi đến hàm GetDlgItem. Đây là một API của Windows, thường được sử dụng trong các ứng dụng giao diện đồ họa (GUI) để lấy *handle* của một control như ô nhập liệu, nút nhấn, v.v.

Vì đây là vị trí có khả năng cao chứa logic xử lý dữ liệu đầu vào, cần đặt các breakpoint tại các lệnh gọi hàm GetDlgItem và GetDlgItemTextA để theo dõi quá trình hoạt động của chương trình. Cụ thể, các địa chỉ đặt breakpoint gồm: 0x401888, 0x4018DC, 0x401CBB, và 0x401CF7.

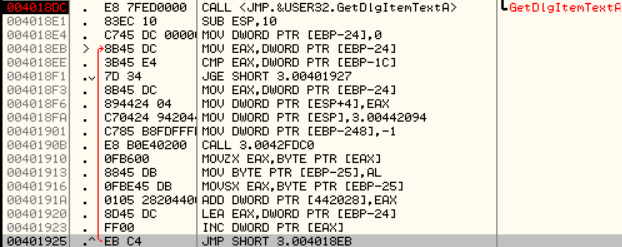
Sau khi đặt breakpoint, tiến hành debug và nhập thông tin:



Tại breakpoint đầu tiên, quan sát tại địa chỉ 0x401898 thấy chương trình gọi API GetWindowTextLengthA để lấy độ dài chuỗi Name và lưu kết quả vào thanh ghi EAX. Ngay sau đó, chương trình so sánh EAX với 4. Nếu EAX 4, luồng thực thi sẽ nhảy xuống địa chỉ 0x00401FD4. Điều này cho thấy điều kiện ràng buộc đầu tiên là **độ dài chuỗi Name phải lớn hơn 4**.



Tại breakpoint thứ hai, quan sát thấy một vòng lặp đang được thực hiện.



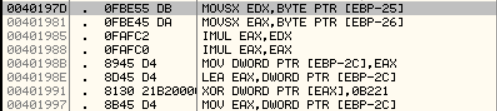
Vòng lặp này tính tổng mã ASCII của các ký tự trong chuỗi Name. Với chuỗi ví dụ là hieu17, tổng sẽ là:

Quan sát đoạn mã tiếp theo:

image

Thực hiện phép toán lấy độ dài chuỗi Name mũ 3 rồi XOR với tổng ASCII vừa tính. Với độ dài là 6, phép toán thực hiện là:

Tiếp tục trace đến địa chỉ 0x40197D:



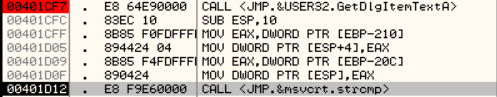
Chương trình lấy mã ASCII của ký tự đầu tiên và ký tự cuối cùng trong chuỗi Name, nhân với nhau, sau đó bình phương kết quả vừa tính. Cuối cùng, kết quả đó được XOR với hằng số 0xB221. Cụ thể:

Tại địa chỉ 0x4019A4:

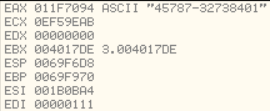
image

Kết quả từ bước trên sau đó được chia cho giá trị tại vùng nhớ 0x442028. Cụ thể:

Tại breakpoint thứ ba, không có thao tác đặc biệt đáng chú ý. Di chuyển đến breakpoint thứ tư:



Tại địa chỉ 0x401D12, chương trình gọi hàm strcmp để so sánh chuỗi. Quan sát trong cửa sổ thanh ghi, thấy xuất hiện cặp giá trị: 45787-32738401.



Tiến hành nhập lại trong chương trình 3.exe với Name là "hieu17" và Serial là "45787-32738401" thì xuất hiện thông báo chúc mừng. Như vậy, đây chính là serial hợp lệ tương ứng với chuỗi Name = "hieu17".

Có thể nhận thấy, phần đầu của serial là kết quả từ phép chia:

Và phần sau là kết quả từ phép XOR:

Từ đây suy ra, serial có dạng:

### Chương trình phát sinh khóa

def keygen3(name: str) -> str:  
 if len(name) <= 4:  
 return "Name too short"  
  
 ascii\_sum = sum(ord(char) for char in name)  
 suffix = ((ord(name[0]) \* ord(name[-1])) \*\* 2) ^ 0xB221  
 prefix = suffix // ((len(name)\*\*3) ^ ascii\_sum)  
  
 return f"{prefix}-{suffix}"

# Đánh giá mức độ hoàn thành