**LẬP TRÌNH HỆ THỐNG– LỚP NT209.L21.ANTN**

**RE CHALLENGES 3: Recursive**

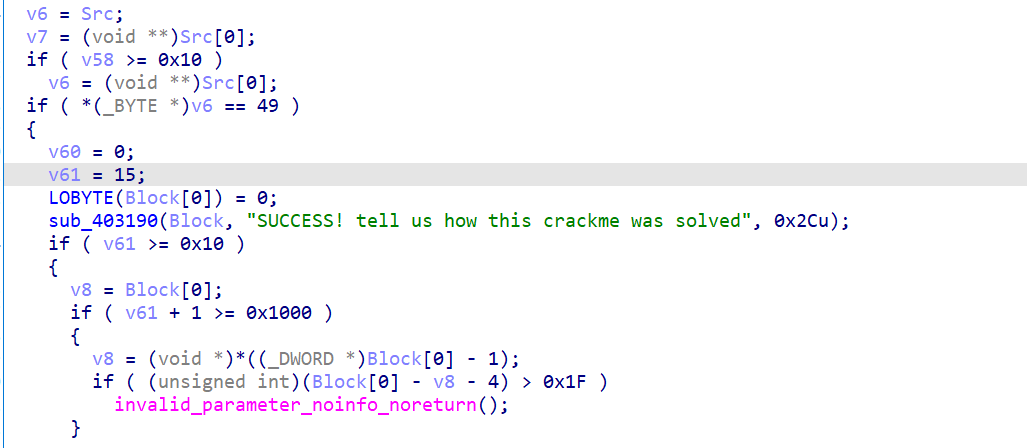
|  |  |  |  |
| --- | --- | --- | --- |
| **Giảng viên hướng dẫn** | **Phạm Văn Hậu** | | **ĐIỂM** |
| **Sinh viên thực hiện 1** | **Trần Đức Lương** | **19521815** |  |

Đây là 1 file thực thi PE 32-bit. Thực hiện chạy thử chương trình cho nhập password rồi tự thoát ra. Mở file bằng IDA phân tích hàm main.

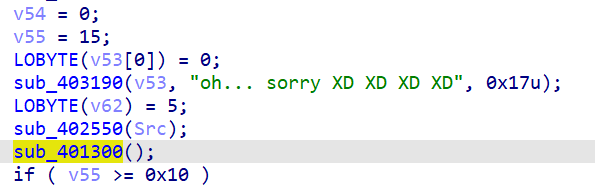


Có thể thấy sub\_4035A0 là hàm cout.

Chương trình lưu chuỗi password nhập vào Src. Ở đây có một loạt đoạn code điều kiện thực hiện những dòng tương tự nhau. Có chuỗi “SUCCESS!..” là chúng ta cần hướng đến, tuy nhiên ở những block code này dùng hàm sub\_403190 không phải hàm cout nên ta sẽ bỏ qua phần này.

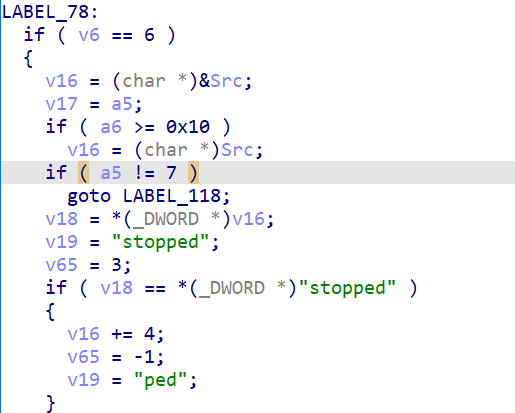


Thực hiện rà code xuống dưới thì thấy xuất hiện hàm sub\_401300 cần xem xét. Lí do vì nó xuất hiện hàm sub\_4035A0 là hàm cout với dòng chữ “SUCCESS! …”. Đây chính là nơi chúng ta cần hướng đến.

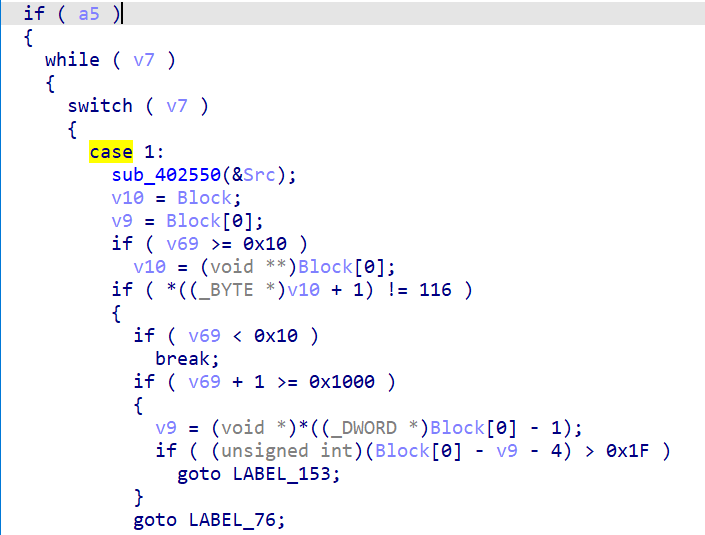




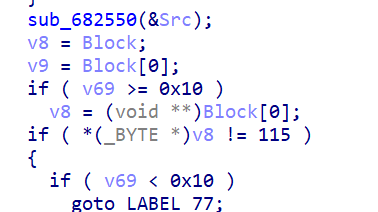
Tại hàm sub\_401300, điều kiện cần để đến với “SUCCESS” là v6 == 6. Ta sẽ đi kiểm tra cách hoạt động của v6.

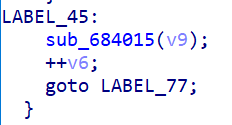


Password nhập vào được lưu ở Src và a5 chính là độ dài chuỗi của Src. Chương trình dùng switch case với v7 sau mỗi lần lặp. Ở đây có 5 case tác động lên v6 (tăng lên 1 ở mỗi case).

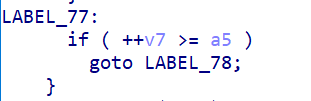


Ban đầu v7 = 0 nên chương trình nhảy xuống đoạn code này. v8 ở đây có lưu đoạn password Src chúng ta nhập vào. Vì ta cần v6 == 6 mà ở trên chỉ có 5 case có thể tăng giá trị v6 nên block code này cũng cần phải tăng v6 lên 1. Khi đó kí tự đầu tiên của password phải là ‘s’ (= 115) => Src[0] = ‘s’.

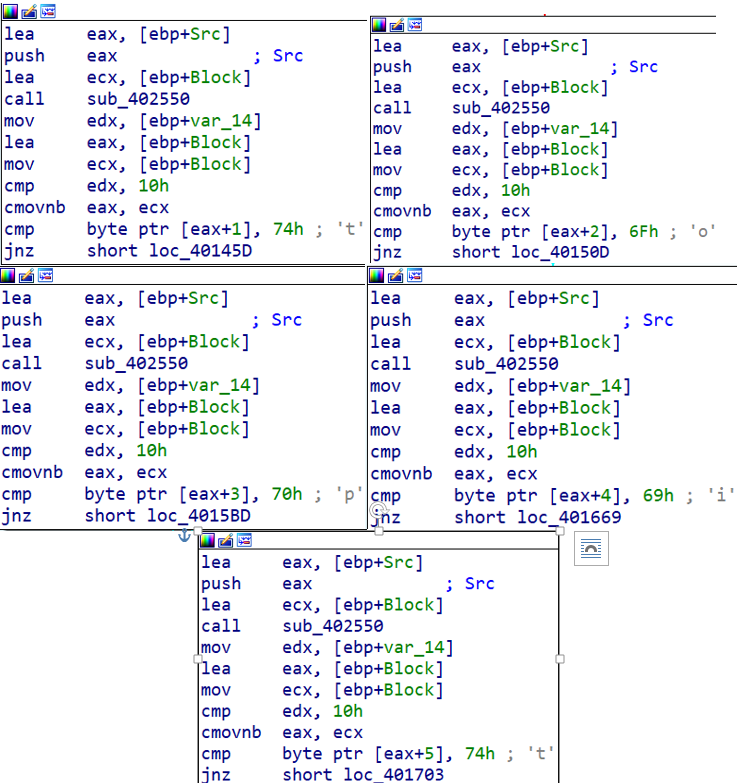




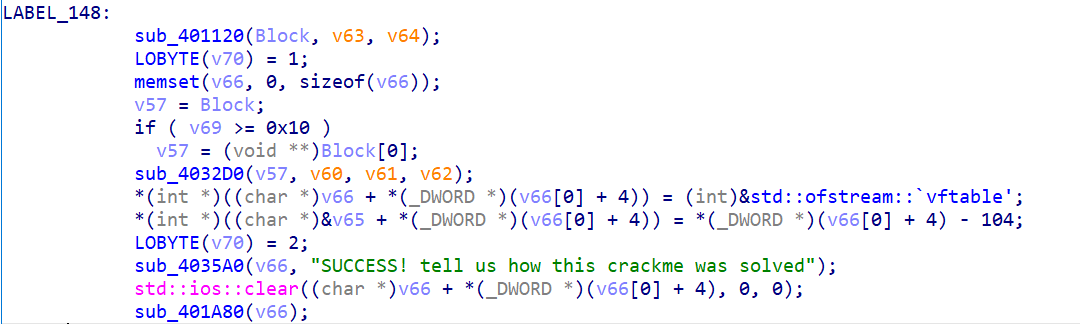
Sau đó, chương trình nhảy đến LABEL\_77 để tăng v7 lên 1, nếu v7 >= a5 tức là độ dài chuỗi nhập vào thì sẽ nhảy đến LABEL\_78 để kiểm tra điều kiện v6 == 6 đã nêu ở trên.



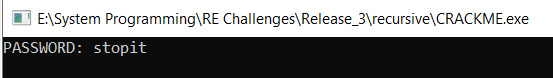
Chương trình tiếp tục với các case 1 đến case 5. Dưới đây là danh sách:

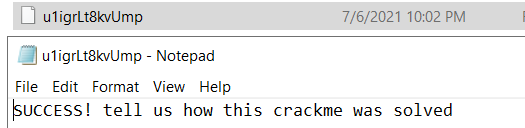


Vì mục tiêu của ta là tăng v6 thêm 5 nữa nên cả 5 case này cần nhảy đến LABEL\_45 để tăng v6. Khi đó dựa điều kiện kiểm tra ở các case, kết hợp với Src[0] = ‘s’ => Src[0:6] = “stopit”. Khi đó v6 == 6. Chương trình sẽ thoát vòng lặp trên khi v7 >= a5. Để tối ưu ta sẽ chỉ cần nhập password = “stopit”. Còn nếu nhập nhiều hơn thì vẫn được vì khi đó v7 sẽ tăng đến khi >= a5 mà không ảnh hưởng đến v6.



Chương trình đi vào block code if(v6 == 6) thì đến LABEL\_148 xuất hiện các hàm sub\_401120 là tạo file với tên Random ở cùng thư mục, hàm sub\_4032D0 thực hiện mở file đó và sub\_4035A0 chính là ghi chuỗi “SUCCESS!...” vào file đó. Thực hiện nhập password = “stopit” thì chương trình nhảy đến LABEL\_148 và tạo ra file “u1igrLt8kvUmp” có nội dùng “SUCCESS!”. Điều này chứng tỏ thành công.





Có thể nhập password = “stopit123” thì chương trình vẫn thành công, miễn là 6 kí tự đầu là chuỗi “stopit”.