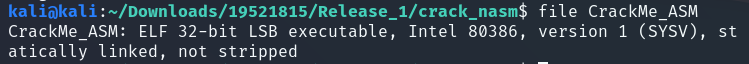
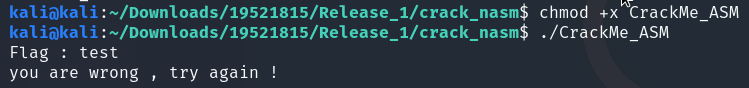
**LẬP TRÌNH HỆ THỐNG– LỚP NT209.L21.ANTN**

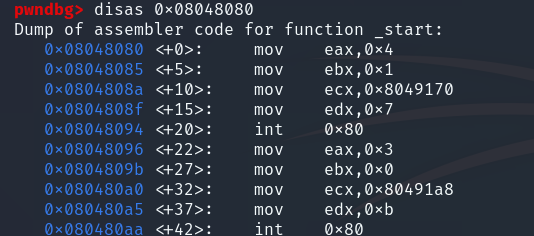
**RE CHALLENGES: CRACK\_NASM**

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

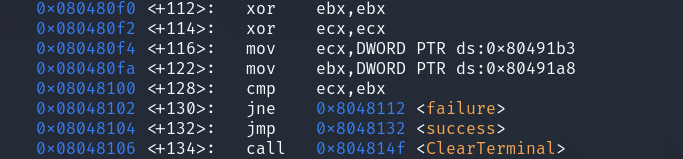
Đây là file ELF 32-bit chạy trên nền Linux.

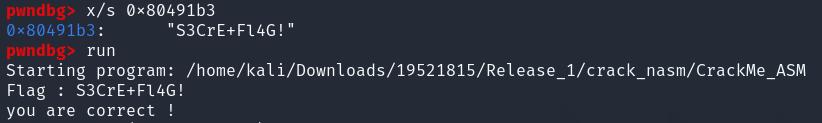
Thực hiện cấp quyền và chạy thử chương trình trên terminal. Chương trình yêu cầu nhập flag để kiểm tra xem có thỏa mãn với flag gốc mà chương trình đưa ra hay không. Thử nhập “test” thì chương trình trả về “you are wrong, try again”. Ý tưởng của bài này cũng chỉ là nhập vào flag. Nếu đúng thì kết quả trả về thành công, ngược lại trả về chuỗi như bên dưới.

Thực hiện quá trình debg trên GDB thì xuất hiện danh sách các hàm. Ta sẽ đi vào disassembly hàm \_start tại offset 0x08048080. Hình bên dưới chính là bước xuất ra màn hình chữ “Flag : ” và nhập flag vào từ bàn phím. Giá trị nhập đó được lưu ở địa chỉ 0x80491a8 và có độ dài 11 kí tự (0xb).



Sau đó chương trình thực hiện so sánh hai giá trị tại hai địa chỉ 0x80491b3 và 0x80491a8, trong đó 0x80491a8 chính là địa chỉ lưu chuỗi mình vừa nhập vào. Nếu chúng không bằng nhau thì nhảy đến hàm <failure>, và hiển thị chuỗi “you are wrong, try again”; ngược lại, hàm <success> sẽ được gọi, tức là chuỗi nhập vào chính là flag cần tìm. Khi đó có thể suy ra là giá trị tại 0x80491b3 chính là flag cần tìm.

Sử dụng câu lệnh x/s để kiểm tra giá trị tại 0x80491b3, kết quả là chuỗi “S3Cr3+Fl4G!”. Thực thi lại chương trình và nhập chuỗi trên thì thành công.

Flag: S3Cr3+Fl4G!