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

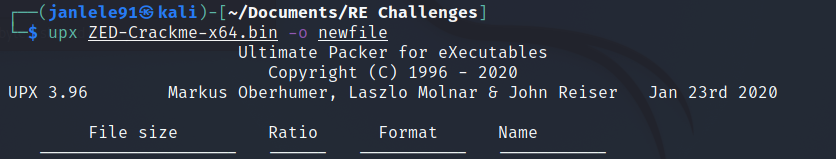
**RE CHALLENGES 2: ZED-Crackme**

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

Kiểm tra định dạng file đã cho thì thấy đây là một **file .so** ELF 64-bit. Mở IDA Pro phân tích thì không thấy có điều gì đặc biệt, ngoài string “**upx …**”. Điều này làm ta nghĩ đến việc giải nén file này bằng **upx**.

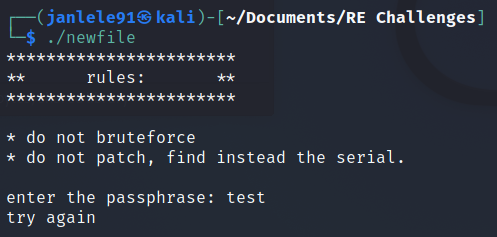


Thực hiện giải nén file **ZED-Crackme-x64.bin** bằng **upx** trên nền tảng linux ta được file thực thi, đặt tên là **newfile**.

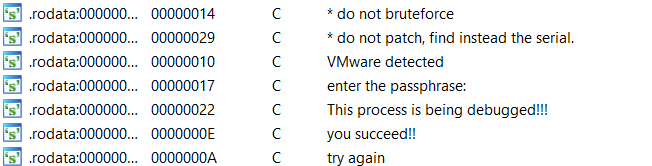




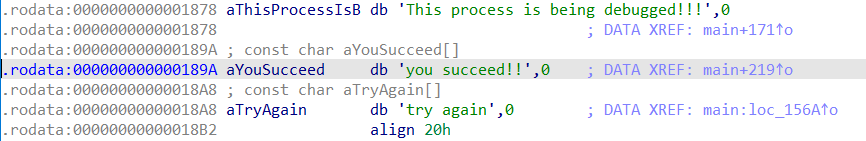
Chạy thử **./newfile** ta được giao diện như hình dưới. Thực hiện nhập passphrase là “**test**” thì chương trình báo không thành công. Nhiệm vụ của chúng ta là đi tìm passphrase của chương trình mà không vi phạm 2 rules chương trình đề ra.



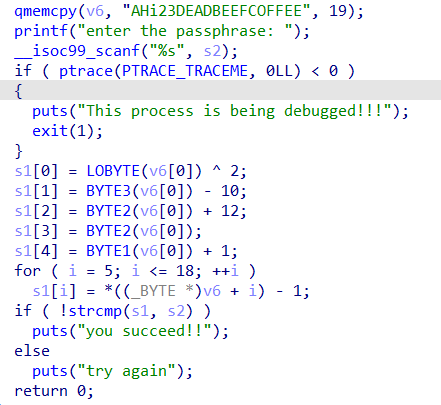
Mở **newfile** bằng IDA Pro, xem các strings có trong file thì thấy xuất hiện dòng “**you succeed!!**”. Đây chính là mục tiêu chúng ta cần hướng đến.



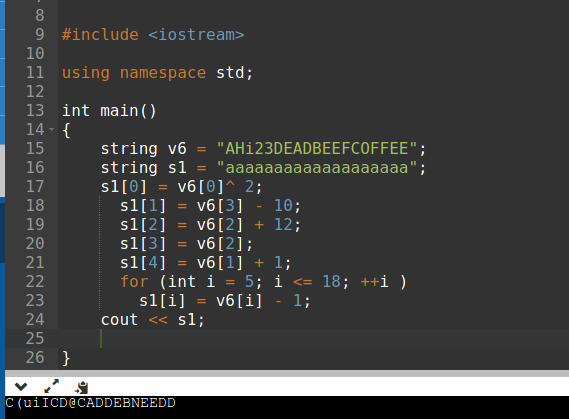
Truy vết string đó thì thấy nó nằm trong hàm **main**



Mở hàm main và ta có một số nhận xét:



Cụ thể, chương trình yêu cầu không cho debug bằng câu lệnh **if(ptrace(PTRACE\_TRACEME, 0) < 0)**, nếu không sẽ thoát chương trình. Điều này khiến ta phải tìm ra passphrase mà không ùng debug. Đọc code thì thấy passphrase ta nhập sẽ được lưu vào chuỗi **s2**, sau đó sẽ so sánh với chuỗi **s1**, nếu bằng nhau thì báo “**you succeed!!**”. Điều đó có nghĩa **s1** chính là passphrase chúng ta cần tìm. Đọc code thì thấy **s1** chính là kết quả biến đổi từng kí tự của chuỗi **v6** là “**AHi23DEADBEEFCOFFEE**”. Thực hiện chạy đoạn code đó bằng C++ ta có kết quả s1 là “**C(uiICD@CADDEBNEEDD**”. Đây chính là passphrase cần tìm.



Chạy chương trình với passphrase “C(uiICD@CADDEBNEEDD” thành công.

