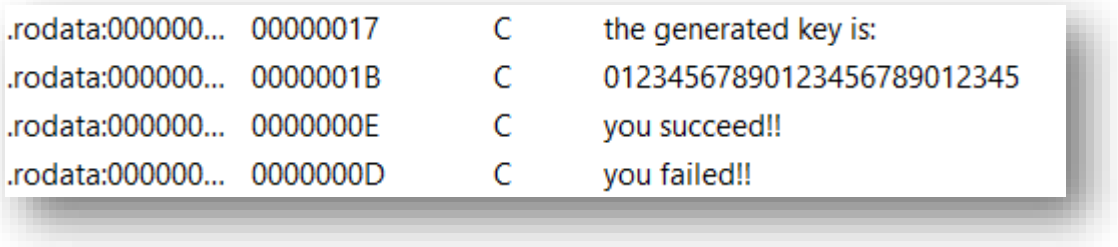
# LẬP TRÌNH HỆ THỐNG– LỚP NT209.L21.ANTN

## RE CHALLENGES 3: ZED - Frequency

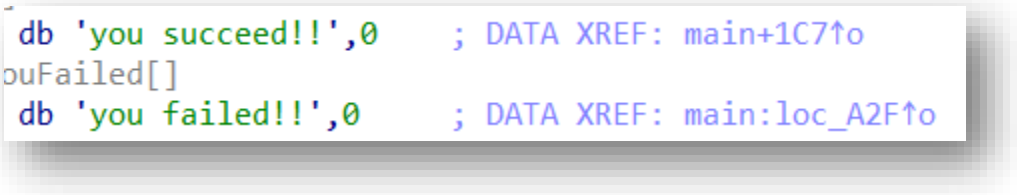
| Giảng viên hướng dẫn | Phạm Văn Hậu | | ĐIỂM |
|---|---|---|---|
| Sinh viên thực hiện 1 | Trần Đức Lương | 19521815 | |

Đây là file thực thi ELF 64-bit. Thực hiện chạy thử chương trình với không tham số đầu vào thì thấy chương trình báo cần thêm tham số là 1 file chứa flag. Ý tưởng của bài này chỉ là tìm flag thích hợp rồi lưu vào file flag.txt để truyền vào làm tham số khi thực thi.

Mở file **ZED-Frequency.bin** bằng IDA Pro để thực hiện quá trình dịch ngược. Xem subview strings thì thấy có dòng "**you succeed!!**" chính là mục tiêu chúng ta cần hướng đến.




Truy vết string trên thì thấy nó nằm trong hàm **main**.




Mở hàm **main**, bắt đầu phân tích:

```
stream = fopen(argv[1], "rt");
for ( i = 0; i <= 25; ++i )
  v8[i] = 0;
while ( 1 )
{
  v5 = fgetc(stream);
  if ( v5 == -1 )
    break;
  if ( v5 <= 96 || v5 > 122 )
  {
    if ( v5 > 64 && v5 <= 90 )
      ++v8[v5 - 65];
  }
  else
  {
    ++v8[v5 - 97];
  }
}
```

Ban đầu chương trình khởi tạo một mảng v8 có 26 phần tử mang giá trị 0. Trong vòng while, chương trình lần lượt lấy từng kí tự của chuỗi lưu trong file đầu vào thông qua stream lưu vào biến v5. Nếu v5 là chữ cái thì thực hiện tăng v8[i]++, trong đó i+1 chính là vị trí của v5 trong bảng chữ cái alphabet. Ví dụ v5 == ‘a’ hoặc ‘A’ thì v8[0]++ ; v5 == ‘b’ hoặc ‘B’ thì v8[1]++ . Như vậy mảng v8 lưu lại số lần xuất hiện của các chữ cái có trong chuỗi đầu vào.

```
for ( j = 0; j <= 25; ++j )
{
  printf("%d", (unsigned int)v8[j]);
  s1[j] = LOBYTE(v8[j]) + 0x30;
}
s1[26] = 0;
putchar(10);
if ( !strcmp(s1, "01234567890123456789012345") )
  puts("you succeed!!");
else
  puts("you failed!!");
```

Chương trình tạo string s1 lưu lại mảng v8 dưới dạng chuỗi rồi đem đi so sánh với chuỗi "***012345678901234567890123456789012345***". Nếu bằng nhau thì ta thành công. Như vậy dựa vào phân tích ở trên, chuỗi đầu vào phải chứa 0 chữ 'a' hoặc 'A', 1 chữ 'b' hoặc 'B', 2 chữ 'c' hoặc 'C', …. Tương tự như thế cho đến hết bảng chữ cái alphabet. Khi đó, sẽ có nhiều chuỗi đầu vào thảo mãn điều kiện trên, ở đây em sẽ chọn chuỗi:
"***bccdddeeeeffffffgggggghhhhhhhhiiiiiiiijjjjjjjjjlmmnnnoooopppppqqqqqqrrrrrrrsssssssssttttttttvwwxxxyyyyzzzzz***".

```
┌──(janlele91㉿kali)-[~/Documents/RE Challenges/Release_3/ZED-frequency]
└─$ echo -n "bccdddeeeeffffffgggggghhhhhhhhiiiiiiiijjjjjjjjjlmmnnnoooopppppqqqqqqrrrrrrr
xxyyyyzzzzz" > text.txt

┌──(janlele91㉿kali)-[~/Documents/RE Challenges/Release_3/ZED-frequency]
└─$ ls
'2021-06-23 02:47:54'   text.txt   ZED-Frequency.bin

┌──(janlele91㉿kali)-[~/Documents/RE Challenges/Release_3/ZED-frequency]
└─$ ./ZED-Frequency.bin text.txt
the generated key is: 012345678901234567890123456789012345
you succeed!!
```

Thực hiện lưu chuỗi trên vào file text.txt làm tham số đầu vào và nhận được kết quả thành công.