

LẬP TRÌNH HỆ THỐNG– LỚP NT209.L21.ANTN

RE CHALLENGES 3: Fence

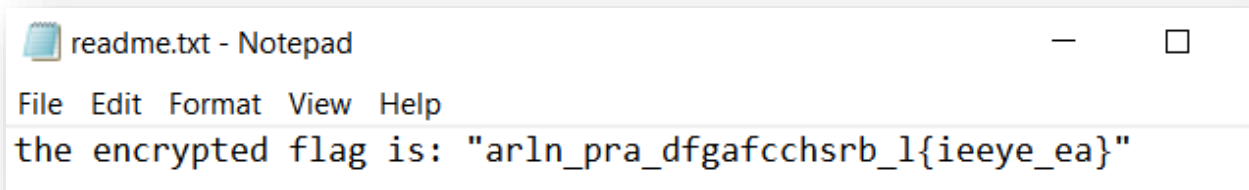
Giảng viên hướng dẫn	Phạm Văn Hậu		ĐIỂM
Sinh viên thực hiện 1	Trần Đức Lương	19521815	

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 đúng 1 tham số. Với tham số đầu vào **“test”** chương trình cho ra chuỗi **“stte”**. Như vậy nhiệm vụ của chương trình encryptor này là encrypt chuỗi tham số đầu vào. (Có thể nghe tên encryptor là đã đoán được.)

```
(janlele91@kali)-[~/Documents/RE Challenges/Release_3/fence]
$ ./encryptor
you must supply exactly one argument

(janlele91@kali)-[~/Documents/RE Challenges/Release_3/fence]
$ ./encryptor test
stte
```

Ta mở file **readme.txt** được cung cấp thì thấy flag cần tìm bị encrypt thành chuỗi **“arln_pra_dfgafcchsrbl{ieeye_ea}”**. Nhiệm vụ của chúng ta là dựa vào quy luật encrypt của chương trình encryptor để decrypt chuỗi này.



```
readme.txt - Notepad
File Edit Format View Help
the encrypted flag is: "arln_pra_dfgafcchsrbl{ieeye_ea}"
```

Mở file **encryptor** bằng IDA Pro để thực hiện quá trình dịch ngược. Mở hàm **main**, bắt đầu phân tích:

```

for ( i = 0LL; ; i += 3LL )
{
    v4 = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::length(v16);
    if ( i >= v4 )
        break;
    v5 = (char *)std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[](v16, i);
    std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator+=(v17, (unsigned int)*v5);
}

```

- Vòng for đầu tiên: Chương trình thực hiện nối những kí tự ở vị trí i, với $i \% 3 == 0$ thành một chuỗi và lưu vào chuỗi v17.

```

for ( j = 1LL; ; j += 3LL )
{
    v6 = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::length(v16);
    if ( j >= v6 )
        break;
    v7 = (char *)std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[](v16, j);
    std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator+=(v18, (unsigned int)*v7);
}

```

- Vòng for thứ 2: Chương trình thực hiện nối những kí tự ở vị trí i, với $i \% 3 == 1$ thành một chuỗi và lưu vào chuỗi v18.

```

for ( k = 2LL; ; k += 3LL )
{
    v8 = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::length(v16);
    if ( k >= v8 )
        break;
    v9 = (char *)std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[](v16, k);
    std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator+=(v19, (unsigned int)*v9);
}

```

- Vòng for thứ 3: Chương trình thực hiện nối những kí tự ở vị trí i, với $i \% 3 == 2$ thành một chuỗi và lưu vào chuỗi v19.

```

std::operator+<char>(v20, v19, v17);
std::operator+<char>(v21, v20, v18);
v10 = std::operator<<<char>(&std::cout, v21);
std::ostream::operator<<(v10, &std::endl<char,std::char_traits<char>>);

```

Khi đó với chuỗi nhập vào, chương trình sẽ in ra chuỗi được encrypt với dạng $v19+v17+v18$. Như vậy, với chuỗi flag *“arln_pra_dfgafchsrbl{ieeye_ea}”*

đã bị encrypt, ta sẽ decrypt dựa trên quy luật encrypt trên và được kết quả là *“flag{railfence_cyphers_are_bad_}”*. Đây chính là flag chúng ta cần tìm.

```
(janlele91@kali)-[~/Documents/RE Challenges/Release_3/fence]  
$ ./encryptor "flag{railfence_cyphers_are_bad_}"  
arl_n_pra_dfgafcchsrbl{ieeye_ea}
```