HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN I

BÀI BÁO CÁO THỰC HÀNH SỐ 4

Môn học: PHÂN TÍCH MÃ ĐỘC

SỬ DỤNG CÔNG CỤ IDA ĐỂ PHÂN TÍCH TĨNH

Tên sinh viên: Ninh Chí Hướng

Mã sinh viên: B20DCAT094

Nhóm lớp : 02

Gi.ảng viên hướng dẫn: PGS.TS Đỗ Xuân Chợ

HÀ NỘI, THÁNG 10/2023

I. Giới thiệu:

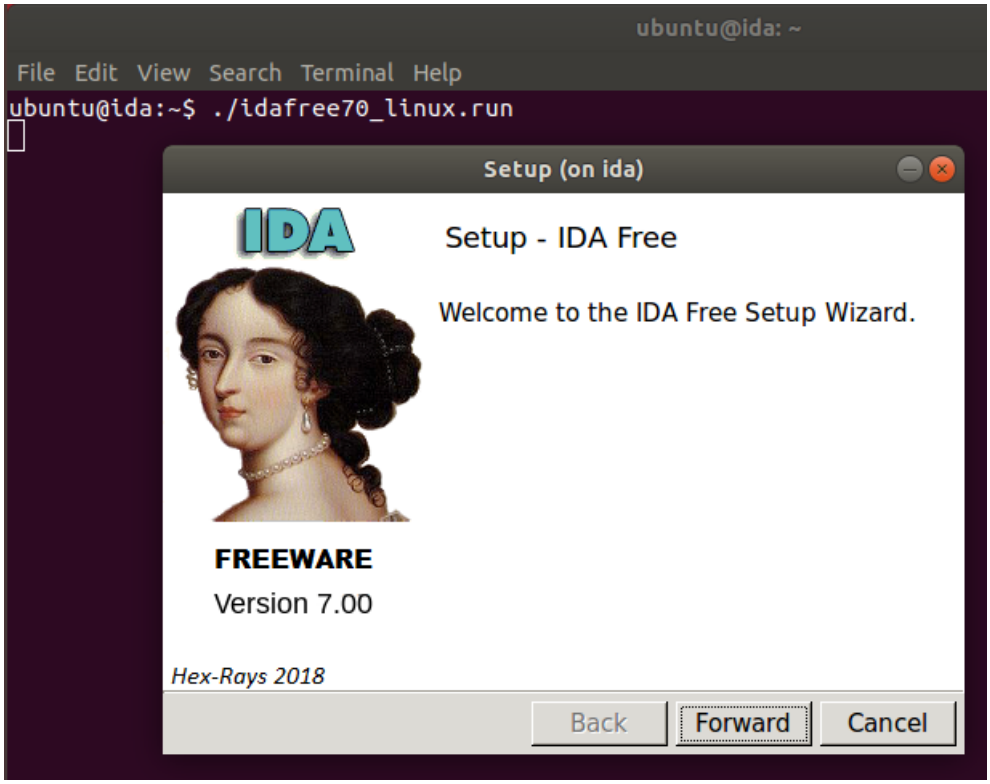
- IDA là phần mềm dịch dịch mã ngược (disassembler) và gỡ lỗi (debugger) giàu tính năng, hỗ trợ nhiều họ vi xử lý (multi-processor), nhiều nền tảng khác nhau (cross platform), đây là một công cụ tạo mã nguồn ngôn ngữ Assembly (assembly language source code) từ mã thực thi của máy (machine-executable code) . IDA là trình phân tách thông minh và đầy đủ tính năng nhất thế giới, được sử dụng bởi các chuyên gia bảo mật phần mềm trên toàn thế giới. Được viết hoàn toàn bằng C ++, IDA chạy trên ba hệ điều hành chính: Microsoft Windows, Mac OS X và Linux.
- Ưu điểm chính của công cụ này là cho phép bạn thay đổi tương tác bất kỳ yếu tố nào của dữ liệu được hiển thị:
    - Đặt tên cho các hàm (functions), biến (variables), cấu trúc dữ liệu (data structures), v.v.
    - Thay đổi cách biểu diễn dữ liệu (dưới dạng số - numbers, dưới dạng chuỗi trong một bảng mã khác nhau – strings, dưới dạng cấu trúc dữ liệu – data structures)
    - Xây dựng sơ đồ và mối liên hệ của dòng lệnh (code) để đơn giản hóa sự hiểu biết về mã nguồn được dịch ngược.
    - Sử dụng thông tin loại về đối số hàm (arguments) và định nghĩa cấu trúc từ C++, để các đối số (arguments) và biến (variables) được tự động đặt tên.
    - Tự động nhận dạng và đặt tên cho các hàm thư viện chuẩn (standard library functions) trong mã nguồn được biên dịch và nhiều hơn nữa..

II. Khái quát chung:

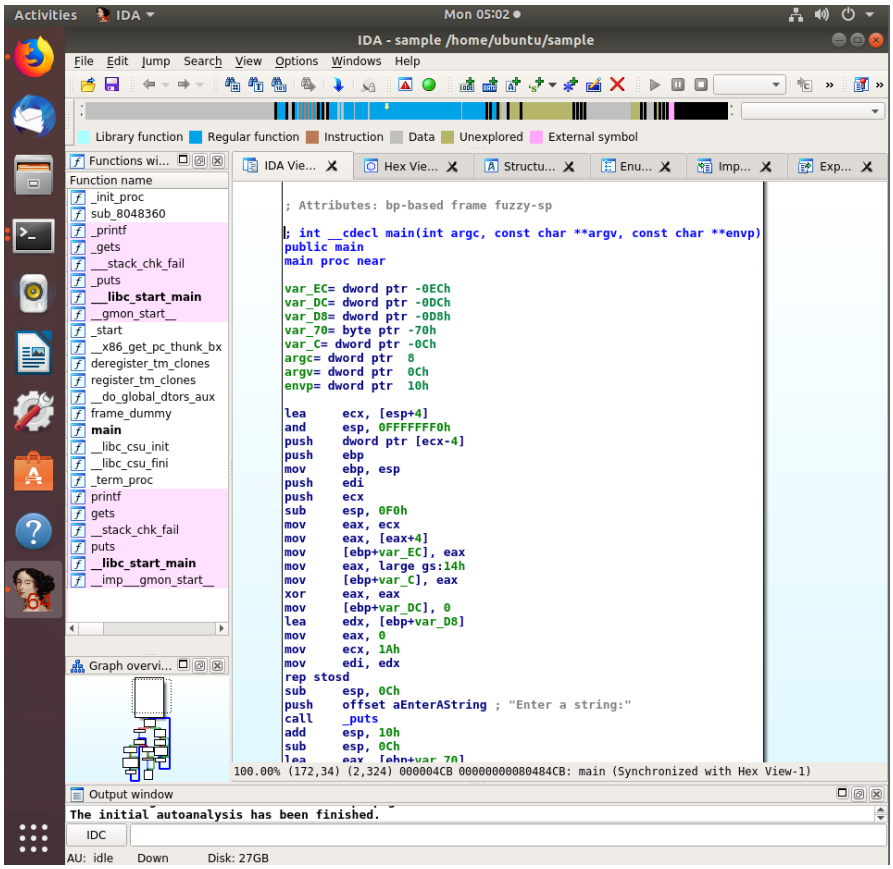
- Mục tiêu của bài thực hành là quan sát sự khác nhau giữa mã nguồn gốc và quá trình disassembly (phân tích ngược) từ dạng nhị phân của nó. Sinh viên sẽ sử dụng phần mềm IDA để phân tích ngược một chương trình C mẫu có tên là sample.c.
- Sau khi hoàn thành bài thực hành, sinh viên sẽ có thấy được những khó khăn khi phân tích ngược chương trình về mã nguồn gốc
- Yêu cầu đối với sinh viên:
    - Sinh viên sử dụng được ngôn ngữ lập trình C, Assembly cơ bản
    - Sinh viên biết và sử dụng được chương trình IDA

III. Các bước thực hiện:

- Cài đặt ida: ./idafree70_linux.run



- Bước 1: Chạy công cụ IDA để đọc chương trình mẫu: ./idafree-7.0/ida64 sample
- Bước 2: Trong giao diện của IDA, sinh viên sẽ thấy phiên bản đã được phân tích ngược của chương trình mẫu.



- Bước 3: so sánh với bản vừa phân tích ngược được với code nguồn sau:

```
#include <stdio.h>
int main(int argc, char * argv[]){
char string[100];
int c = 0, count[26] = {0};
printf("Enter a string:\n");
gets(string);
while ( string[c] != '\0' ){
        if ( string[c] >= 'a' && string[c] <= 'z' )
        count[string[c]-'a']++;
        c++;
}
for ( c = 0 ; c < 26 ; c++ ){
        if ( count[c] != 0 )
                printf("%d %d.\n",c+'a',count[c]);
        }
        return 0;
}
```

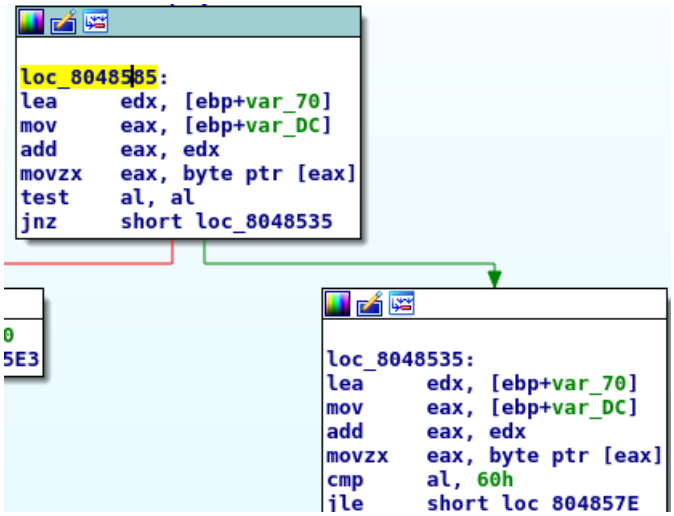
- Ta có thể thấy ngôn ngữ mã nguồn đã được chuyển đổi thành assembly thày vì C như ban đầu

```
main proc near

var_EC= dword ptr -0ECh
var_DC= dword ptr -0DCh
var_D8= dword ptr -0D8h
var_70= byte ptr -70h
var_C= dword ptr -0Ch
argc= dword ptr  8
argv= dword ptr  0Ch
envp= dword ptr  10h

lea     ecx, [esp+4]
and     esp, 0FFFFFFF0h
push    dword ptr [ecx-4]
push    ebp
mov     ebp, esp
push    edi
push    ecx
sub     esp, 0F0h
mov     eax, ecx
mov     eax, [eax+4]
mov     [ebp+var_EC], eax
mov     eax, large gs:14h
mov     [ebp+var_C], eax
xor     eax, eax
mov     [ebp+var_DC], 0
lea     edx, [ebp+var_D8]
mov     eax, 0
mov     ecx, 1Ah
mov     edi, edx
rep stosd
sub     esp, 0Ch
push    offset aEnterAString ; "Enter a string:"
call    _puts
add     esp, 10h
sub     esp, 0Ch
lea     eax, [ebp+var_70]
push    eax
call    _gets
add     esp, 10h
jmp     short loc_8048585
```

- Tên hàm đã bị thay đổi.



- Các dữ liệu trong IDA là byte thay thì các kiểu thông thường như string và int trong C

```
lea     ecx, [esp+4]
and     esp, 0FFFFFFF0h
```

- Bước 4: Đề xuất ít nhất hai sửa đổi cho mã nguồn gốc có thể được sử dụng để bảo vệ mã đó khỏi bị phân tích ngược khi sử dụng IDA
    - Mã hóa toàn bộ mã nguồn

- Sử dụng kỹ thuật obfuscation (mờ hóa)

- **Kết luận:**

  IDA là một công cụ dịch ngược phổ biến được sử dụng trong lĩnh vực phân tích và dịch ngược mã máy, ung cấp một giao diện đồ họa mạnh mẽ và linh hoạt để phân tích mã máy, giúp người dùng hiểu cấu trúc và hoạt động của một chương trình, hỗ trợ nhiều kiến trúc và hệ điều hành khác nhau, bao gồm cả Windows, Linux và macOS, có khả năng phân tích tĩnh và động, cho phép xem mã máy, xác định hàm, biến, cấu trúc dữ liệu, và các phần khác của chương trình, cung cấp các tính năng như tìm kiếm chuỗi, tìm kiếm mẫu, phân tích luồng điều khiển, phân tích đồ thị chương trình, và nhiều công cụ khác để hỗ trợ quá trình dịch ngược. Tuy IDA là một công cụ mạnh mẽ, nhưng việc sử dụng nó đòi hỏi kiến thức chuyên sâu về phân tích mã máy và cấu trúc chương trình.