

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**BÁO CÁO BÀI THỰC HÀNH**

**Kiểm thử xâm nhập**

**Lỗ hổng Buffer Overflow**

**Giảng viên: Đinh Trường Duy**

**Nhóm lớp: 02**

**Sinh viên: Hoàng Trung Kiên**

**Mã sinh viên: B20DCAT098**

**Hà Nội – 2024**

**Mục lục**

[1. **Mục đích.** 3](#_Toc164409415)

[2. **Yêu cầu đối với sinh viên.** 3](#_Toc164409416)

[3. **Nội dung thực hành.** 3](#_Toc164409417)

[Nhiệm vụ 1: Khai thác lỗ hổng bảo mật. 4](#_Toc164409418)

[Nhiệm vụ 2: Ngẫu nhiên hóa địa chỉ. 5](#_Toc164409419)

[Nhiệm vụ 3: Stack Guard. 6](#_Toc164409420)

[Nhiệm vụ 4: Ngăn xếp không thực thi. 6](#_Toc164409421)

[4. **Checkwork**. 7](#_Toc164409422)

1. **Mục đích.**

Giúp sinh viên hiểu rõ nguyên nhân, cơ chế cũng như cách thức khắc phục, xử lý lỗ hổng tràn bộ đệm.

1. **Yêu cầu đối với sinh viên.**

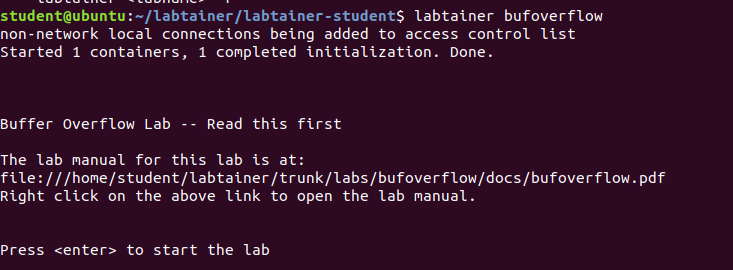
Có kiến thức về ngôn ngữ lập trình C.

Tìm hiểu về Stack, Function, Buffer Overflow và các cơ chế bảo vệ của hệ thống.

1. **Nội dung thực hành.**

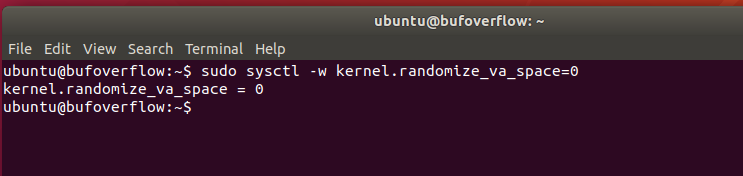
Khởi động lab

Chạy lệnh: *labtainer -r**bufoverflow*trong terminal của Labtainer



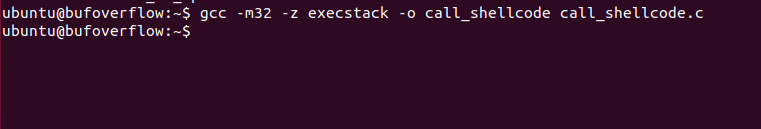
- Tắt các cơ chế bảo vệ có liên quan

sudo sysctl -w kernel.randomize\_va\_space=0



biên dịch call\_shellcode.c

gcc -m32 -z execstack -o call\_shellcode call\_shellcode.c



Biên dịch chương trình stack.c và chuyển quyền thành set-root-uid

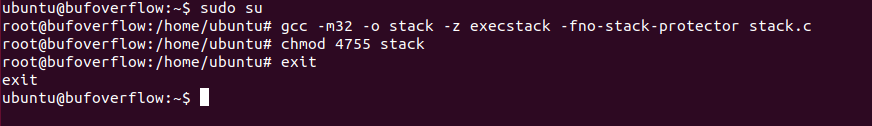
sudo su gcc -m32 -o stack -z execstack -fno-stack-protector stack.c

chmod

4755

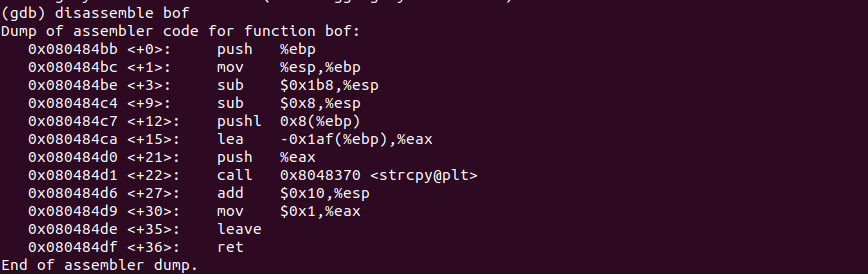
stack

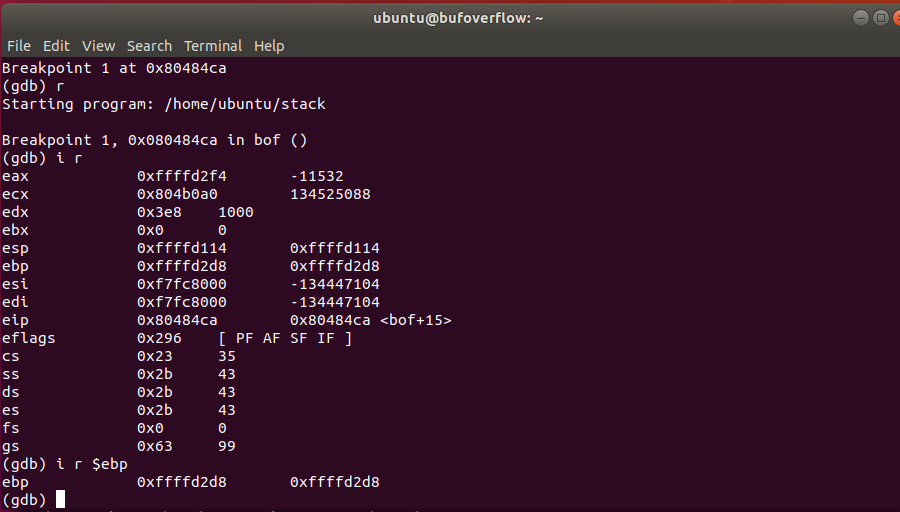
exit

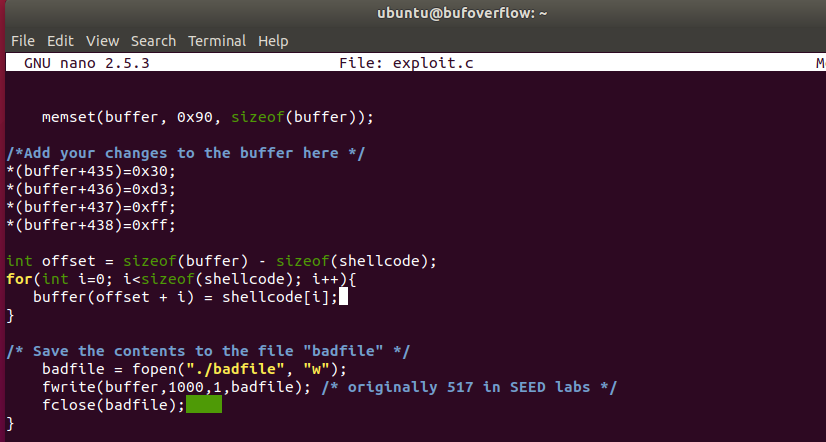


## Nhiệm vụ 1: Khai thác lỗ hổng bảo mật.

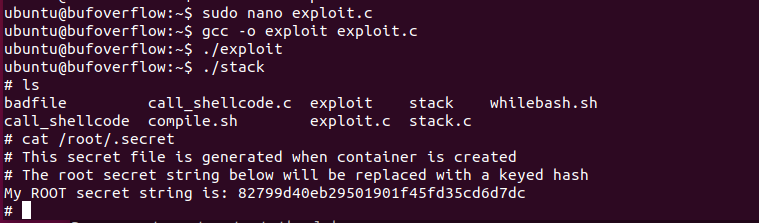
Tìm offset của buffer







Biên dịch và chạy



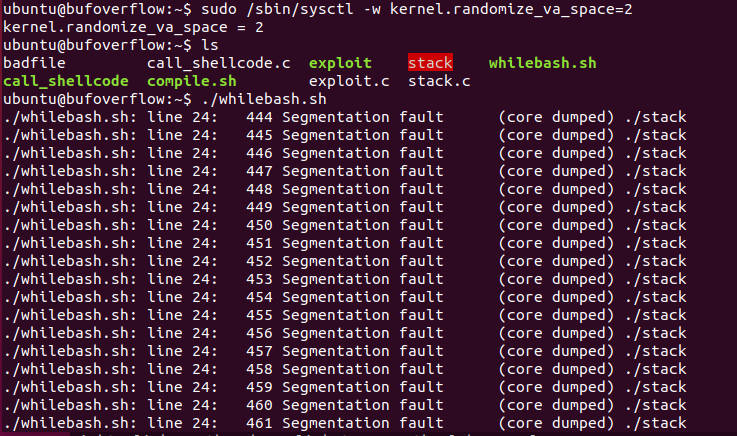
Ta đã có quyền root và đọc được file secret

## Nhiệm vụ 2: Ngẫu nhiên hóa địa chỉ.

Bật ngẫu nhiên hóa địa chỉ cho file

sudo /sbin/sysctl -w kernel.randomize\_va\_space=2

gcc -m32 -o stack -z execstack stack.c

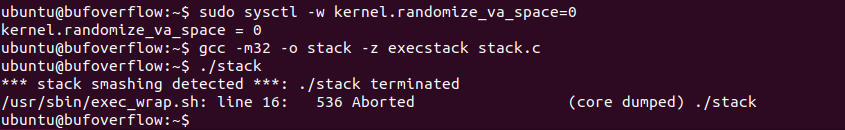


**Trả lời câu hỏi:** Ta có thể nhận được shell không? Nếu không, vấn đề là gì? Làm thế nào để ngẫu nhiên hóa địa chỉ làm cho các cuộc tấn công trở nên khó khăn?  
Ta không nhận được shell vì địa chỉ chúng ta set cho return trong hàm bof() của stack mặc định, nên khi nó ngẫu nhiên địa chỉ thì địa chỉ trả về của chúng ta không đúng trong stack nên nó không thành công. Điều đó làm các cuộc tấn công khó khăn thực hiện thành công thì không thể biết được giá trị nào hợp lệ

## Nhiệm vụ 3: Stack Guard.

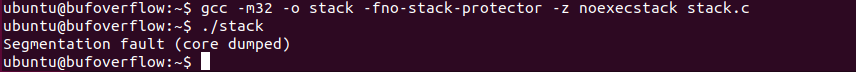
sudo sysctl -w kernel.randomize\_va\_space=0

gcc -m32 -o stack -z execstack stack.c



## Nhiệm vụ 4: Ngăn xếp không thực thi.

gcc -m32 -o stack -fno-stack-protector -z noexecstack stack.c



Tùy chọn -z noexecstack khi biên dịch chương trình sẽ ngăn chặn khả năng thực thi trên stack. Điều này thường được sử dụng như một biện pháp bảo mật để ngăn chặn việc thực thi mã được đặt vào stack, giảm thiểu khả năng bị tấn công bởi các cuộc tấn công như tràn bộ đệm. Sau khi biên dịch chương trình với noexecstack, việc tìm và thực thi shellcode từ stack sẽ không còn khả thi.

1. **Checkwork**.

