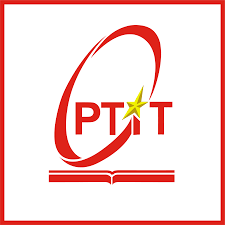
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**



**MÔN HỌC:** PHÂN TÍCH MÃ ĐỘC

**BÁO CÁO THỰC HÀNH BÀI 1**

**Giảng viên:** PGS.TS. Đỗ Xuân Chợ

**Sinh viên:** Hoàng Trung Kiên – B20DCAT098

Hà Nội – 5/2023

Mục lục

[1. Tìm hiểu về Linux Capability 3](#_Toc147478891)

[a, Mục đích 3](#_Toc147478892)

[b, Lý thuyết. 3](#_Toc147478893)

[2. Thực hành 3](#_Toc147478894)

[**Nhiệm vụ 1:** Trải nghiệm Capabilities 4](#_Toc147478895)

[***Nhiệm vụ 1.1: Cho phép người dùng không đặc quyền chạy tcpdump*** 6](#_Toc147478896)

[**Nhiệm vụ 1.2: Chuyển đổi passwd để sử dụng capabilities** 6](#_Toc147478897)

[**Nhiệm vụ 2:**Điều chỉnh đặc quyền 7](#_Toc147478898)

[3. Checkwork 10](#_Toc147478899)

[4. Trả lời câu hỏi 10](#_Toc147478900)

# 1. Tìm hiểu về **Linux Capability**

## a, Mục đích

Mục tiêu học tập của lab này là để sinh viên có được kinh nghiệm trực tiếp về việc sử dụng Linux capabilities để đạt được nguyên tắc của quyền tối thiểu. Lab này dựa trên POSIX 1.e capabilities, được thực hiện trong các phiên bản gần đây của kernel Linux.

Hệ thống dựa trên capabilities được đôi khi quảng cáo như là một chiến lược kiểm soát truy cập so với việc sử dụng Access Control Lists (ACLs) hoặc quyền tập tin Unix. Trong thực tế, hệ thống Linux thường sử dụng capabilities để giới hạn đặc quyền của chương trình thay vì để kiểm soát quyền truy cập vào các đối tượng có tên. Lab này tập trung vào việc sử dụng capabilities để giới hạn đặc quyền.

## b, Lý thuyết.

Linux Capability là một tập các quyền hạn đặc biệt có thể được cấp cho một tiến trình hoặc tệp. Chúng cung cấp khả năng kiểm soát chi tiết hơn đối với các quyền so với cách phân định root/non-root truyền thống.

Có hai loại Linux Capability:

* Permitted capabilities: Đây là những quyền mà một tiến trình hoặc tệp được phép sử dụng.
* Effective capabilities: Đây là những quyền mà một tiến trình hoặc tệp thực sự đang sử dụng.

Capabilities là một công cụ mạnh mẽ để kiểm soát quyền hạn trong Linux. Chúng có thể được sử dụng để cấp các quyền đặc biệt cho các tiến trình và tệp mà không cần phải biến chúng thành root.

Dưới đây là một số ví dụ về cách sử dụng Linux Capability:

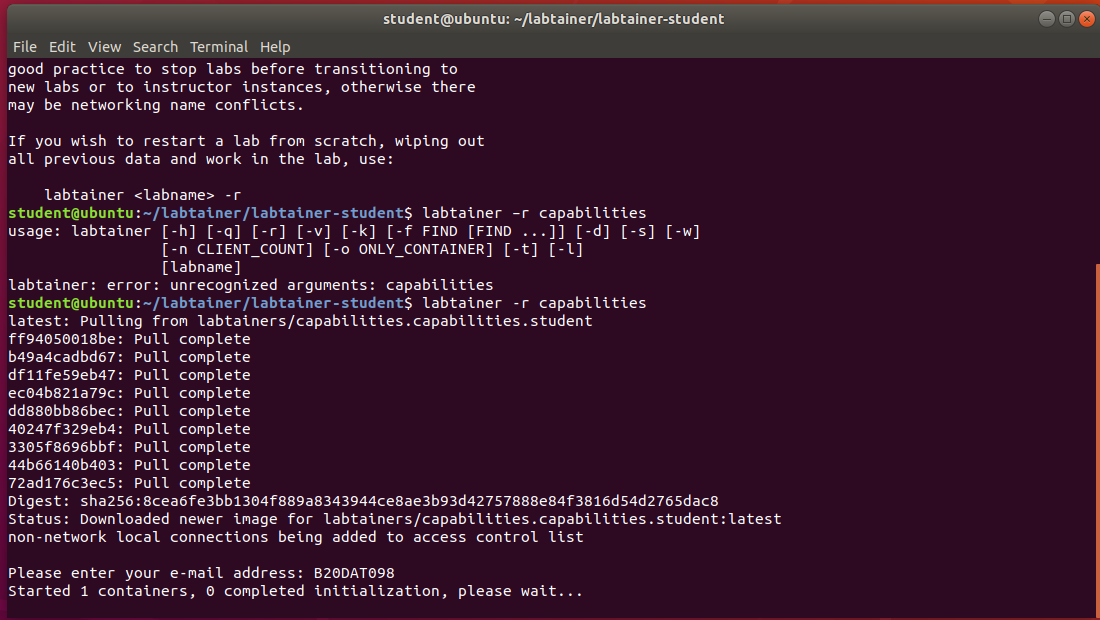
* Một máy chủ web có thể được cấp quyền CAP\_NET\_BIND\_SERVICE để nó có thể liên kết với các cổng được ưu tiên.
* Một máy chủ cơ sở dữ liệu có thể được cấp quyền CAP\_SYS\_RESOURCE để nó có thể khóa bộ nhớ.
* Một tệp có thể được cấp quyền CAP\_SETUID để nó có thể được thực thi dưới dạng người dùng khác.

Capabilities cũng có thể được sử dụng để tạo sandbox cho các tiến trình. Ví dụ, một runtime container có thể sử dụng capabilities để hạn chế quyền hạn của các tiến trình mà nó chạy.

Linux Capabilities là một chủ đề phức tạp, nhưng chúng là một công cụ mạnh mẽ để kiểm soát quyền hạn trong Linux.

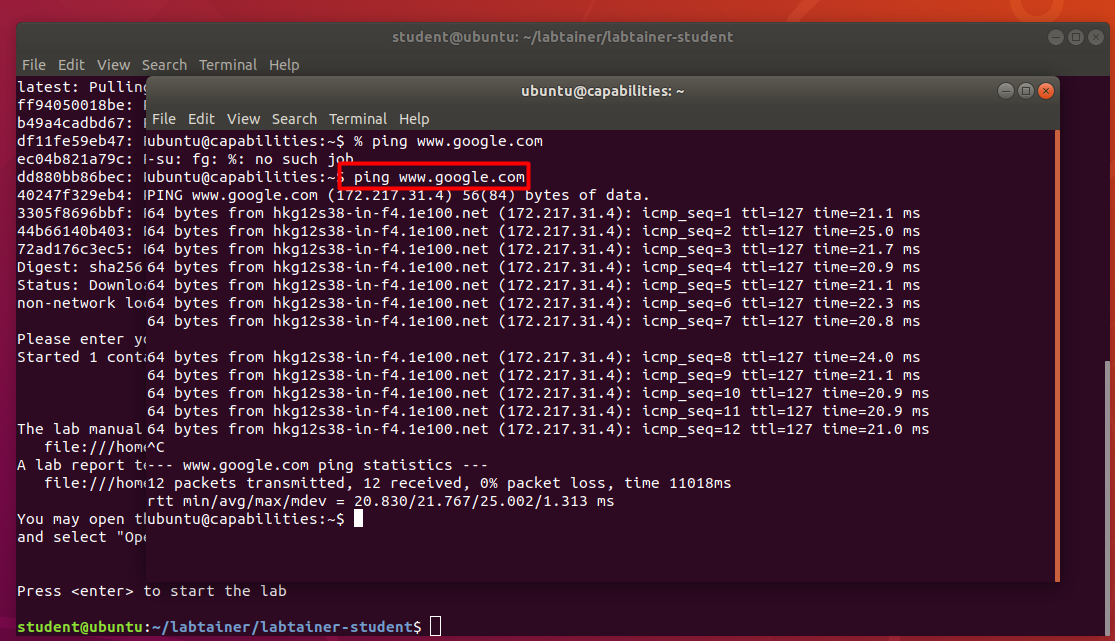
# 2. Thực hành

Khởi động bài lab và nhập mã sinh viên

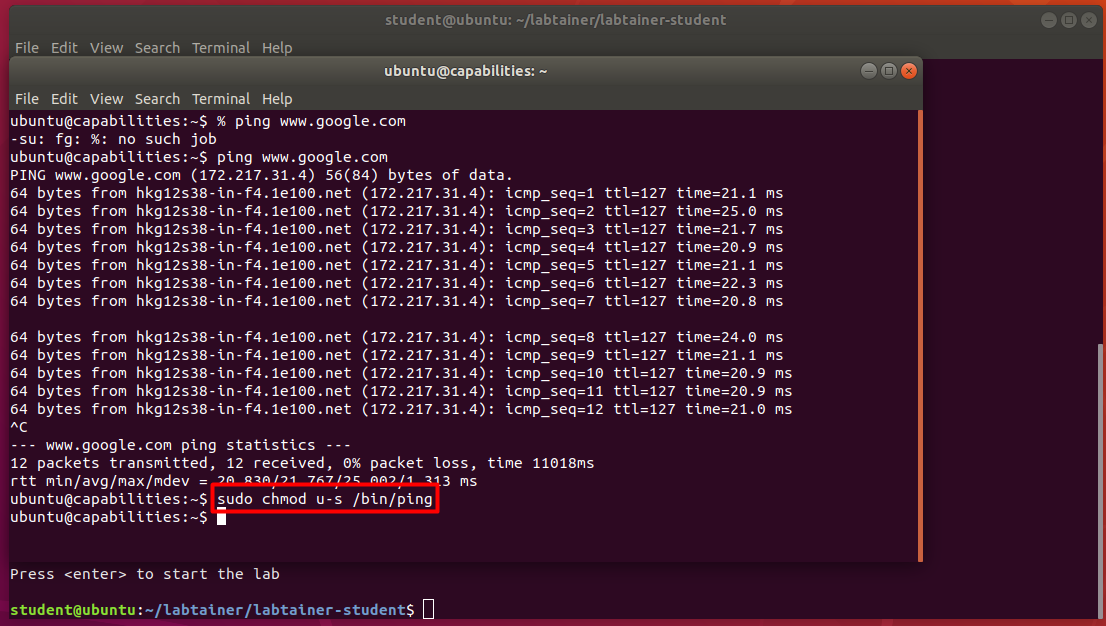


## **Nhiệm vụ 1:** Trải nghiệm Capabilities

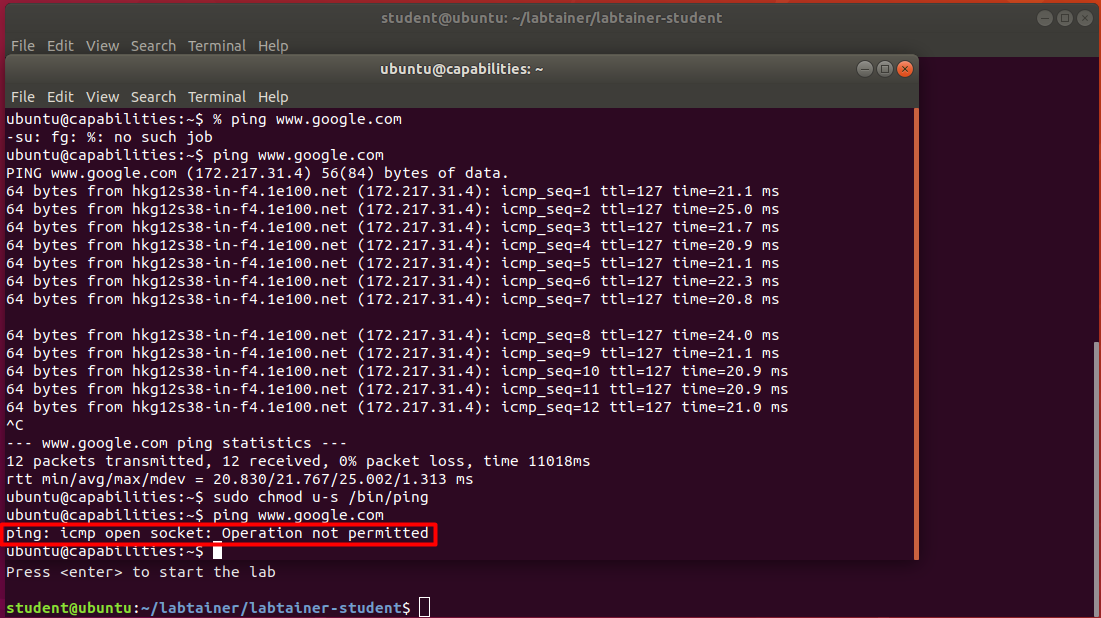
Với tư cách là người dùng ubuntu không có đặc quyền



Bằng cách sử dụng capabilities, chúng ta có thể loại bỏ các đặc quyền không cần thiết từ ping.



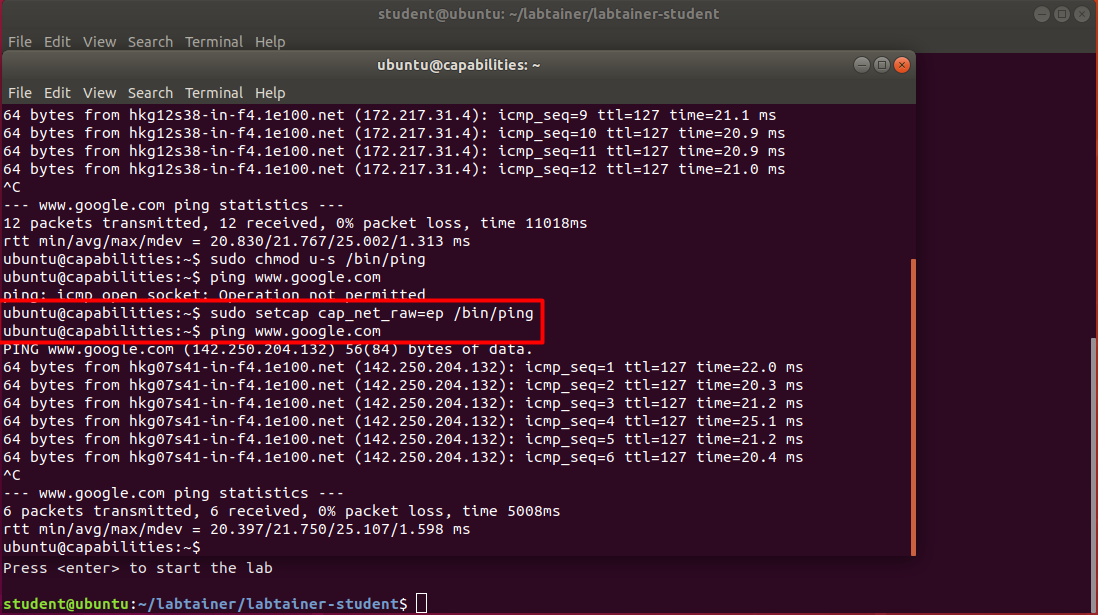
Chạy lại với lệnh “ping [www.google.com](http://www.google.com)” sau khi thực hiện lệnh sudo chmod u-s /bin/ping



Ta gán capability cap\_net\_raw cho ping:

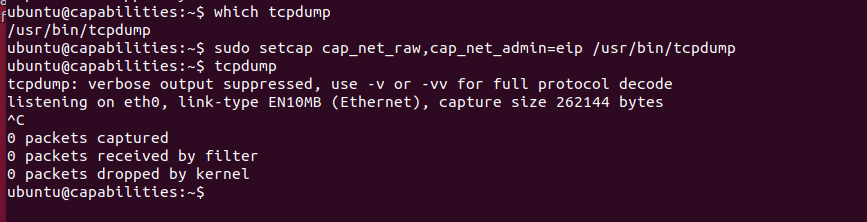
sudo setcap cap\_net\_raw=ep /bin/ping

sau đó chạy lệnh: ping www.google.com



### **Nhiệm vụ 1.1: Cho phép người dùng không đặc quyền chạy tcpdump**

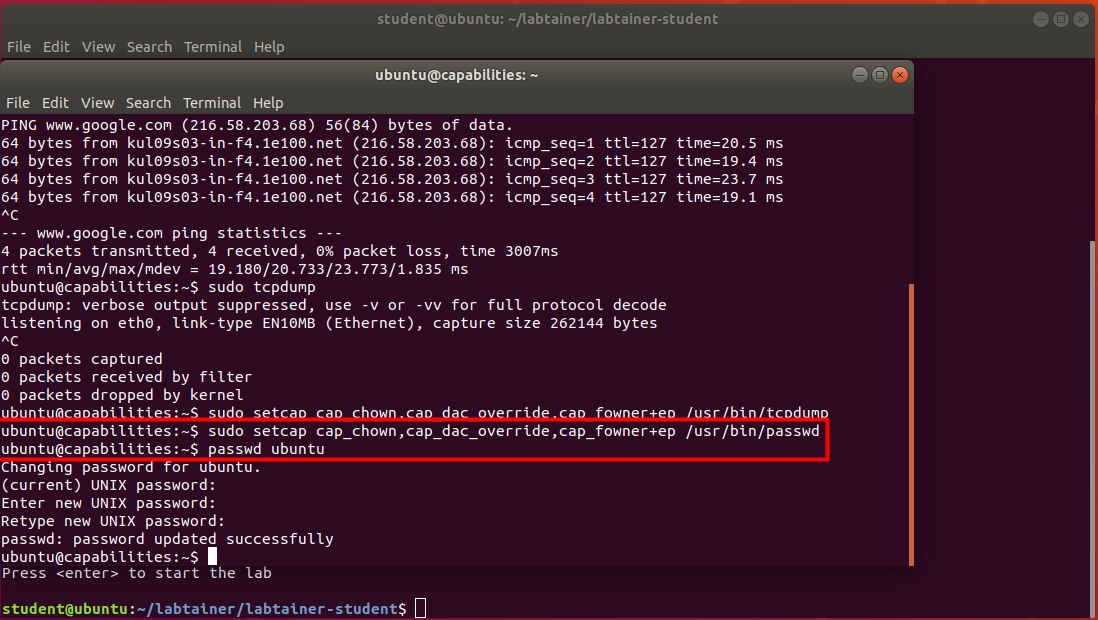
Chạy lệnh sudo tcpdump



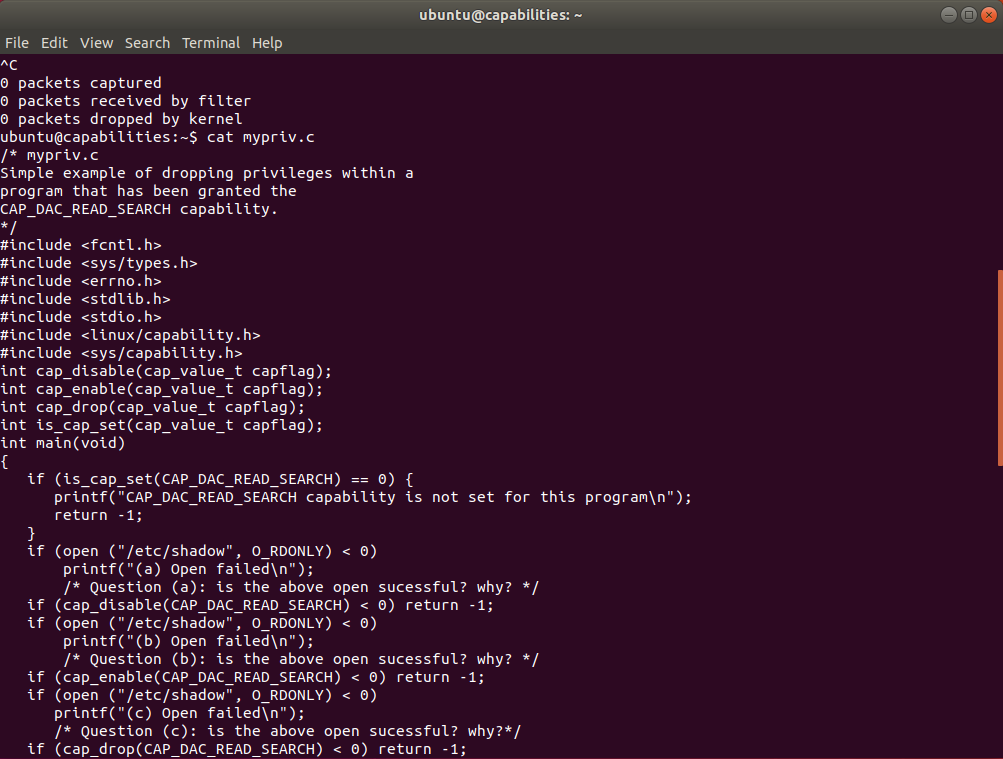
Sửa đổi chương trình tcpdump để cho phép người dùng không đặc quyền chạy nó.

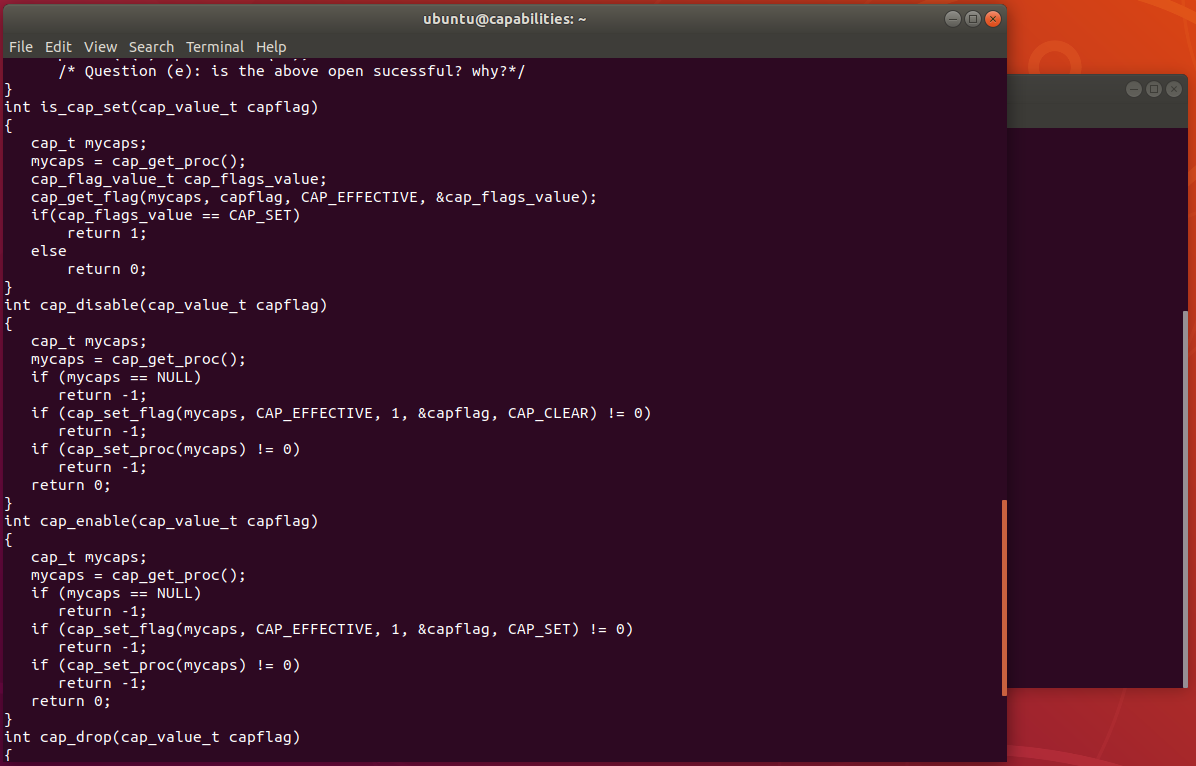
### **Nhiệm vụ 1.2: Chuyển đổi passwd để sử dụng capabilities**

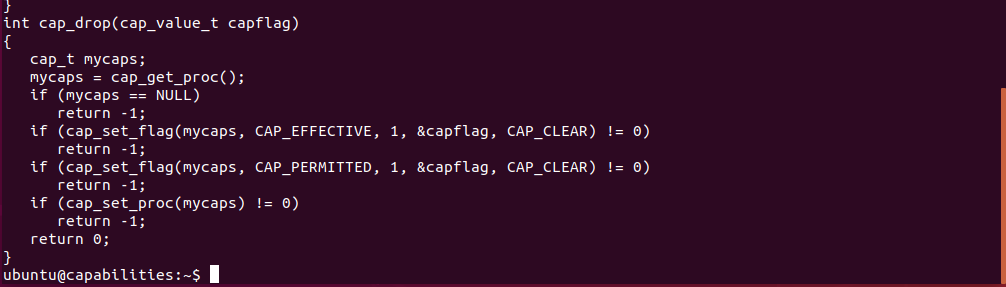
Sửa đổi chương trình passwd để sử dụng khả năng thay vì setuid, sau đó chứng minh rằng nó vẫn hoạt động bằng cách thay đổi mật khẩu của người dùng ubuntu



## **Nhiệm vụ 2:**Điều chỉnh đặc quyền







Câu hỏi (a): Câu lệnh open ("/etc/shadow", O\_RDONLY) trả về giá trị -1, điều này có nghĩa rằng việc mở tệp /etc/shadow không thành công.

Câu hỏi (b): Câu lệnh open ("/etc/shadow", O\_RDONLY) cũng trả về giá trị -1. Khi chúng ta gọi cap\_disable(CAP\_DAC\_READ\_SEARCH), chúng ta đã tắt khả năng CAP\_DAC\_READ\_SEARCH, dẫn đến việc không thể đọc tệp /etc/shadow nữa.

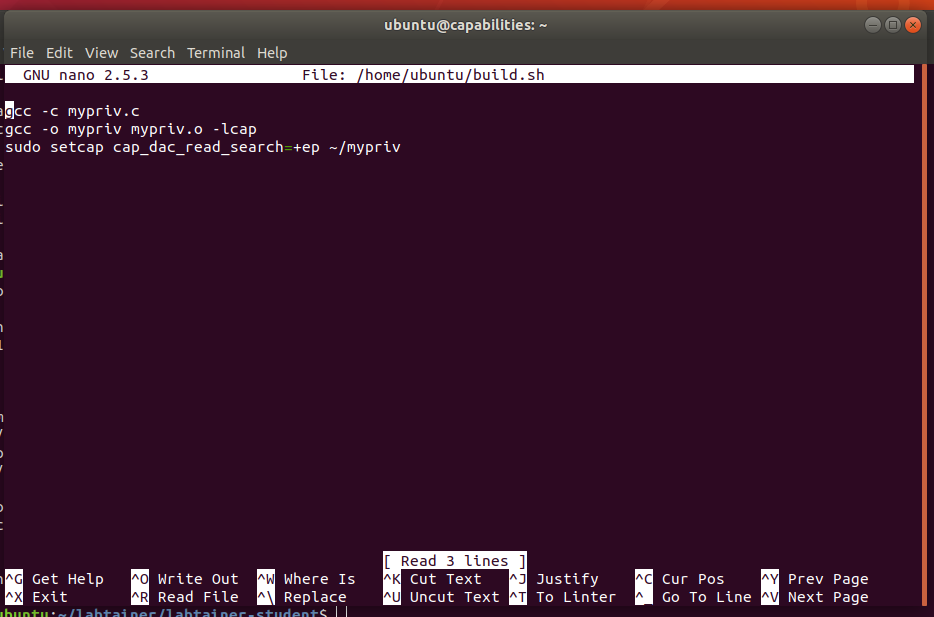
Câu hỏi (c): Câu lệnh open ("/etc/shadow", O\_RDONLY) trở thành thành công, không trả về giá trị -1. Sau khi chúng ta gọi cap\_enable(CAP\_DAC\_READ\_SEARCH), chúng ta đã bật lại khả năng CAP\_DAC\_READ\_SEARCH, cho phép đọc tệp /etc/shadow.

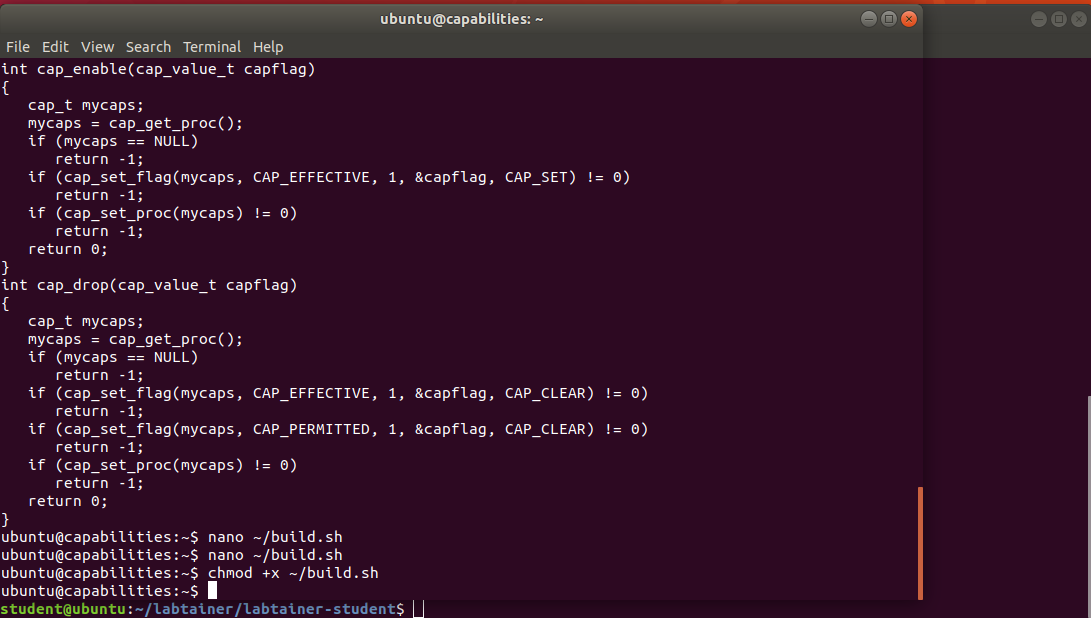
Câu hỏi (d): Câu lệnh open ("/etc/shadow", O\_RDONLY) trả về giá trị -1. Sau khi chúng ta gọi cap\_drop(CAP\_DAC\_READ\_SEARCH), chúng ta đã loại bỏ khả năng CAP\_DAC\_READ\_SEARCH, điều này dẫn đến việc không thể đọc tệp /etc/shadow nữa.

Câu hỏi (e): Câu lệnh open ("/etc/shadow", O\_RDONLY) trở thành thành công, không trả về giá trị -1. Điều này xảy ra sau khi chúng ta gọi cap\_enable(CAP\_DAC\_READ\_SEARCH) để bật lại khả năng CAP\_DAC\_READ\_SEARCH, cho phép đọc tệp /etc/shadow. Mã nguồn mypriv.c này giúp minh họa cách rơi bỏ quyền đặc biệt trong một chương trình đã được cấp quyền CAP\_DAC\_READ\_SEARCH. Chương trình sử dụng các hàm cap\_disable(), cap\_enable(), và cap\_drop() để điều chỉnh trạng thái của khả năng CAP\_DAC\_READ\_SEARCH. Việc rơi bỏ các quyền đặc biệt như vậy có thể giúp giảm tiềm năng rủi ro bảo mật trong các ứng dụng hệ thống.

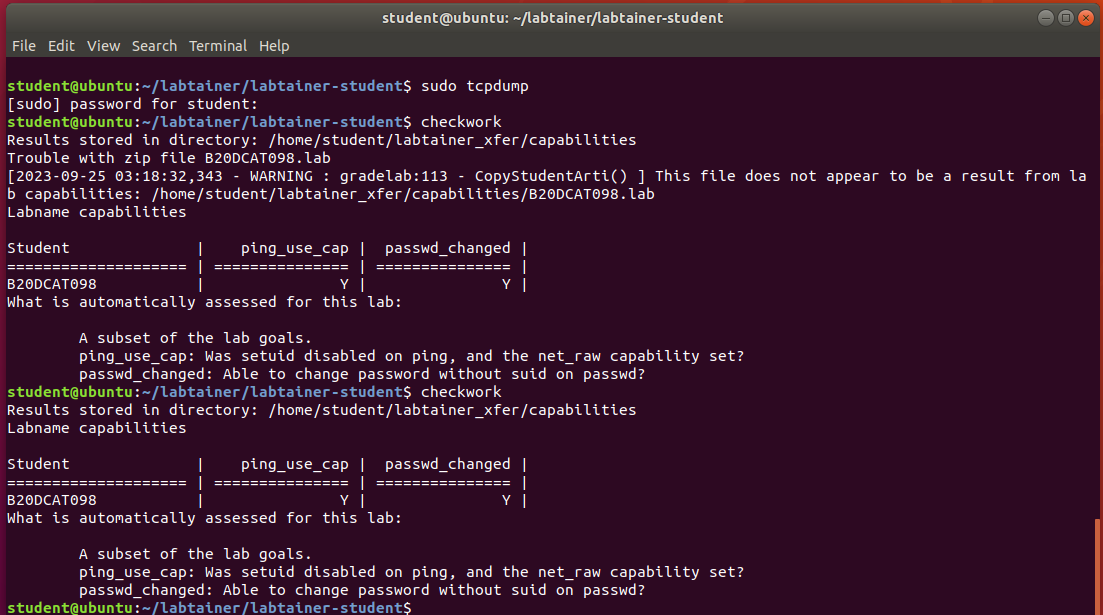
-Vào file build.sh, thêm câu lệnh: sudo setcap cap\_dac\_read\_search=+ep ~/mypriv







# 3. Checkwork



# 4. Trả lời câu hỏi

Câu 1:

Trong tình huống mô tả của bạn, sau khi một chương trình vô hiệu hóa khả năng A, nó bị tấn công bởi một cuộc tấn công tràn bộ đệm và kẻ tấn công chèn mã độc vào không gian ngăn xếp của chương trình:

-Kẻ tấn công không thể sử dụng khả năng A nếu chương trình đã vô hiệu hóa nó: Nếu chương trình đã được vô hiệu hóa khả năng A, điều này có nghĩa là khả năng A không còn khả dụng cho chương trình đó. Dù kẻ tấn công có chèn mã độc vào không gian ngăn xếp, nó không thể sử dụng khả năng A trong quá trình thực thi mã độc đó.

-Kẻ tấn công không thể sử dụng khả năng A nếu quy trình xóa khả năng: Nếu quy trình xóa khả năng A đã được thực hiện, điều đó có nghĩa là khả năng A đã bị gỡ bỏ hoàn toàn khỏi chương trình. Ngay cả khi kẻ tấn công chèn mã độc vào không gian ngăn xếp, nó cũng không thể sử dụng khả năng A, vì nó đã bị xóa và không có sẵn trong chương trình nữa.

Câu 2:

Trong trường hợp cuộc tấn công được thay thế bằng cuộc tấn công đua điều kiện:

-Kẻ tấn công có thể sử dụng khả năng A nếu khả năng này bị tạm ngừng: Nếu khả năng A trong chương trình đã bị tạm ngừng, điều này có nghĩa là chương trình không thể sử dụng khả năng A trong quá trình thực thi bình thường. Tuy nhiên, nếu kẻ tấn công khai thác cuộc đua điều kiện và chèn mã độc vào chương trình, nó có thể tận dụng các điều kiện đặc biệt để kích hoạt lại khả năng A trong quá trình thực thi mã độc. Điều này có thể cho phép kẻ tấn công sử dụng khả năng A mà không cần phải dựa vào quá trình thực thi bình thường của chương trình.

-Kẻ tấn công không thể sử dụng khả năng nếu khả năng đã bị xóa: Nếu khả năng A đã bị xóa hoàn toàn khỏi chương trình, thì dù kẻ tấn công có khai thác cuộc đua điều kiện hoặc chèn mã độc vào chương trình, nó không thể sử dụng khả năng A. Việc xóa khả năng A đồng nghĩa với việc loại bỏ khả năng đó khỏi chương trình và không cho phép bất kỳ thực thể nào sử dụng nó, bao gồm cả kẻ tấn công.