

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**KHOA AN TOÀN THÔNG TIN**

---



Bài báo cáo thực hành số 1

**Môn học : Phân tích mã độc**

**Tìm hiểu về Linux Capability**

**Tên sinh viên:** Ninh Chí Hường

**Mã sinh viên:** B20DCAT094

**Nhóm lớp :** 02

**Giảng viên hướng dẫn:** PGS.TS Đỗ Xuân Chợt

HÀ NỘI, THÁNG 10/2023

# Tìm hiểu về Linux Capability

## 1. 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.

## 2. Yêu cầu đối với sinh viên:

Đối với lab này, bạn cần làm quen với các lệnh sau đây đi kèm với libcap:

setcap: gán capabilities cho một tập tin.

getcap: hiển thị các capabilities được gán cho một tập tin.

getpcaps: hiển thị các capabilities được gán cho một tiến trình.

## 3. Nội dung thực hành

Chuẩn bị lab

- Khởi động lab:

```
lbtainer -r capabilities
```

*(chú ý: sinh viên sử dụng tên tài khoản của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu trong terminal, để sử dụng khi chấm điểm. Thông thường tên tài khoản của sinh viên chính là Mã sinh viên)*

```
student@ubuntu:~/labtainer/labtainer-student$ labtainer -r capabilities
non-network local connections being added to access control list

Please enter your e-mail address: [B20DCAT094]
Started 1 containers, 1 completed initialization. Done.

The lab manual is at
  file:///home/student/labtainer/trunk/labs/capabilities/docs/capabilities.pdf
A lab report template is at
  file:///home/student/labtainer/trunk/labs/capabilities/docs/capabilities-report.docx

You may open these by right clicking
and select "Open Link".

Press <enter> to start the lab

student@ubuntu:~/labtainer/labtainer-student$
```

## Nhiệm vụ 1: Trải nghiệm Capabilities

Trong các hệ điều hành như Linux, có nhiều hoạt động đặc quyền chỉ có thể được thực hiện bởi người dùng có đặc quyền. Ví dụ về các hoạt động đặc quyền bao gồm cấu hình các card giao diện mạng, sao lưu tất cả các tập tin người dùng, tắt máy tính, v.v. Mà không có capabilities, các hoạt động này chỉ có thể được thực hiện bởi siêu người dùng (superuser), người thường có nhiều đặc quyền hơn là cần thiết cho các nhiệm vụ dự định. Do đó, để cho phép siêu người dùng thực hiện các hoạt động đặc quyền này là vi phạm Nguyên tắc Quyền tối thiểu (Least-Privilege Principle).

Các hoạt động đặc quyền là cần thiết trong Linux và các hệ điều hành dựa trên Unix khác. Tất cả các chương trình Set-UID đều liên quan đến các hoạt động đặc quyền không thể được thực hiện bởi người dùng thông thường. Để cho phép người dùng thông thường chạy các chương trình này, các chương trình Set-UID biến người dùng thông thường thành người dùng mạnh (ví dụ: root) một cách tạm thời, mặc dù các hoạt động đặc quyền liên quan không cần tất cả các đặc quyền được cung cấp cho siêu người dùng. Điều này nguy hiểm: nếu chương trình bị xâm phạm, kẻ tấn công có thể có được đặc quyền root.

Capabilities chia quyền root thành một tập hợp các đặc quyền riêng biệt. Mỗi đặc quyền này được gọi là một capability. Với capabilities, chúng ta không cần phải là người dùng siêu người dùng để thực hiện các hoạt động đặc quyền. Chúng ta chỉ cần có những capabilities cần thiết cho các hoạt động đặc quyền. Do đó, ngay cả khi một chương trình đặc quyền bị xâm phạm, kẻ tấn công chỉ có thể có quyền hạn giới hạn. Như vậy, rủi ro của chương trình đặc quyền có thể được giảm thiểu.

Capabilities đã được triển khai trong Linux trong một thời gian khá dài, nhưng chỉ có thể gán cho các tiến trình. Kể từ phiên bản kernel 2.6.24, capabilities có thể được gán cho các tập tin (tức là các chương trình) và biến những chương trình này thành các chương trình đặc quyền. Khi một chương trình đặc quyền được thực thi, tiến trình chạy sẽ mang theo các capabilities đã được gán cho chương trình đó. Một số mặt, điều này tương tự như các tệp Set-UID, nhưng khác biệt chính là số lượng đặc quyền được mang bởi các tiến trình đang chạy.

Chúng ta sẽ sử dụng một ví dụ để cho thấy cách capabilities có thể được sử dụng để loại bỏ các đặc quyền không cần thiết được gán cho một số chương trình đặc quyền. Trước tiên, với tư cách là người dùng ubuntu không có đặc quyền, chạy lệnh sau:

```
% ping www.google.com
```

```
ubuntu@capabilities:~$ ping www.google.com
PING www.google.com (142.250.66.36) 56(84) bytes of data.
64 bytes from hkg12s26-in-f4.1e100.net (142.250.66.36): icmp_seq=1 ttl=127 time=27.5 ms
64 bytes from hkg12s26-in-f4.1e100.net (142.250.66.36): icmp_seq=2 ttl=127 time=29.3 ms
64 bytes from hkg12s26-in-f4.1e100.net (142.250.66.36): icmp_seq=3 ttl=127 time=28.8 ms
64 bytes from hkg12s26-in-f4.1e100.net (142.250.66.36): icmp_seq=4 ttl=127 time=29.5 ms
64 bytes from hkg12s26-in-f4.1e100.net (142.250.66.36): icmp_seq=5 ttl=127 time=29.5 ms
^C
--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 27.577/28.988/29.572/0.773 ms
ubuntu@capabilities:~$
```

```
labtainer: error: unrecognized arguments: capabilities
student@ubuntu:~/labtainer/labtainer-student$ labtainer -r capabilities
non-network local connections being added to access control list

Please enter your e-mail address: [B20DCAT094]
Started 1 containers, 1 completed initialization. Done.
```

Chương trình sẽ chạy thành công. Nếu bạn xem các thuộc tính tệp của chương trình /bin/ping, bạn sẽ thấy rằng ping là một chương trình Set-UID với chủ sở hữu là root, tức là khi bạn thực thi ping, ID người dùng hiệu lực của bạn trở thành root và tiến trình đang chạy do đó chạy với đặc quyền root. Nếu có lỗ hổng trong ping, toàn bộ hệ thống có thể bị xâm phạm. Bằng cách sử dụng capabilities, chúng ta có thể loại bỏ các đặc quyền không cần thiết từ ping.

Trước tiên, hãy biến /bin/ping thành một chương trình không phải Set-UID. Điều này có thể được thực hiện thông qua lệnh sau:

```
sudo chmod u-s /bin/ping
```

```
ubuntu@capabilities:~$ ping www.google.com
PING www.google.com (142.250.66.36) 56(84) bytes of data.
64 bytes from hkg12s26-in-f4.1e100.net (142.250.66.36): icmp_seq=1 ttl=127 time=27.5 ms
64 bytes from hkg12s26-in-f4.1e100.net (142.250.66.36): icmp_seq=2 ttl=127 time=29.3 ms
64 bytes from hkg12s26-in-f4.1e100.net (142.250.66.36): icmp_seq=3 ttl=127 time=28.8 ms
64 bytes from hkg12s26-in-f4.1e100.net (142.250.66.36): icmp_seq=4 ttl=127 time=29.5 ms
64 bytes from hkg12s26-in-f4.1e100.net (142.250.66.36): icmp_seq=5 ttl=127 time=29.5 ms
^C
--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 27.577/28.988/29.572/0.773 ms
ubuntu@capabilities:~$ sudo chmod u-s /bin/ping
ubuntu@capabilities:~$ ping www.google.com
ping: icmp open socket: Operation not permitted
ubuntu@capabilities:~$
```

```
labtainer: error: unrecognized arguments: capabilities
student@ubuntu:~/labtainer/labtainer-student$ labtainer -r capabilities
non-network local connections being added to access control list

Please enter your e-mail address: [B20DCAT094]
Started 1 containers, 1 completed initialization. Done.
```

Bây giờ, chạy 'ping www.google.com' và xem điều gì xảy ra. Lệnh sẽ không thành công. Điều này xảy ra vì ping cần mở một RAW socket, đó là một hoạt động đặc quyền chỉ có thể được thực hiện bởi root (trước khi capabilities được triển khai). Đó là lý do tại sao ping phải là một chương trình Set-UID. Hãy chỉ gán capability cap\_net\_raw cho ping và xem điều gì xảy ra:

```
sudo setcap cap_net_raw=ep /bin/ping
```

```
ping www.google.com
```

```
ubuntu@capabilities:~$ sudo setcap cap_net_raw=ep /bin/ping
ubuntu@capabilities:~$ ping www.google.com
PING www.google.com (142.251.220.36) 56(84) bytes of data.
64 bytes from hkg07s50-in-f4.1e100.net (142.251.220.36): icmp_seq=1 ttl=127 time=21.0 ms
64 bytes from hkg07s50-in-f4.1e100.net (142.251.220.36): icmp_seq=2 ttl=127 time=50.3 ms
64 bytes from hkg07s50-in-f4.1e100.net (142.251.220.36): icmp_seq=3 ttl=127 time=22.2 ms
64 bytes from hkg07s50-in-f4.1e100.net (142.251.220.36): icmp_seq=4 ttl=127 time=22.8 ms
64 bytes from hkg07s50-in-f4.1e100.net (142.251.220.36): icmp_seq=5 ttl=127 time=21.8 ms
^C
--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 21.050/27.676/50.335/11.345 ms
ubuntu@capabilities:~$
```

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

Ngoài việc giảm đặc quyền liên quan đến các chương trình setuid, capabilities có thể được sử dụng để cho phép người dùng không đặc quyền chạy một số chương trình đã chọn mà không cần cấp quyền sudo cho những người dùng đó. Một ví dụ phổ biến là chương trình /usr/bin/tcpdump. Trong các bài tập Labtainer khác, chúng ta sử dụng tcpdump để bắt lưu lượng mạng, tuy nhiên chúng ta làm điều đó bằng cách chạy lệnh:

```
sudo tcpdump
```

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

```
ubuntu@capabilities:~$ which tcpdump
/usr/bin/tcpdump
ubuntu@capabilities:~$ sudo setcap cap_net_raw,cap_net_admin=eip /usr/bin/tcpdump
ubuntu@capabilities:~$ tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
```

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

Một số chương trình setuid yêu cầu nhiều khả năng khác nhau để hoạt động. Tập: /usr/include/linux/capability.h mô tả các khả năng khác nhau có sẵn trong Linux. Chương trình /usr/bin/passwd yêu cầu các khả năng sau để hoạt động:

cap\_chown

cap\_dac\_override

cap\_fowner

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 (ban đầu là ubuntu).

```
ubuntu@capabilities:~$ sudo setcap cap_chown,cap_dac_override,cap_fowner+ep /usr/bin/passwd
ubuntu@capabilities:~$ passwd ubuntu
Changing password for ubuntu.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
You must choose a longer password
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
ubuntu@capabilities:~$
```

## Nhiệm vụ 2: Điều chỉnh đặc quyền

Với khả năng, ta có thể điều chỉnh mức độ đặc quyền một tiến trình có, điều này phù hợp với nguyên tắc đặc quyền tối thiểu. Ví dụ, khi một quyền đặc quyền không còn cần thiết trong một quy trình, ta nên cho phép tiến trình loại bỏ vĩnh viễn các khả năng liên quan đến quyền đặc quyền này. Do đó, ngay cả khi tiến trình bị xâm nhập, kẻ tấn công sẽ không thể sử dụng được các khả năng đã bị xóa này. Điều chỉnh đặc quyền có thể được thực hiện bằng cách sử dụng các thao tác quản lý khả năng sau đây.

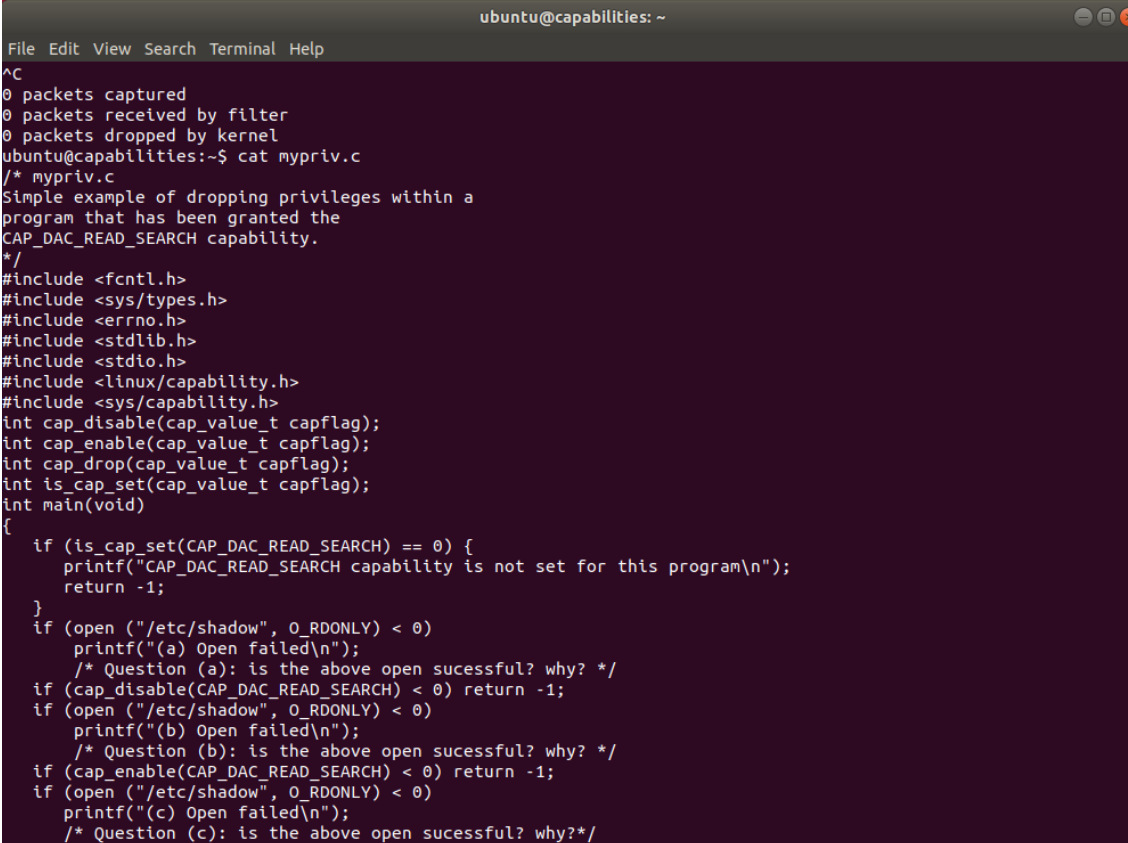
1. Xóa: Một tiến trình có thể xóa khả năng một cách vĩnh viễn.
2. Tạm ngừng: Một tiến trình có thể tạm thời vô hiệu hóa một khả năng. Không giống như việc xóa, việc tạm ngừng chỉ là tạm thời; tiến trình có thể kích hoạt lại sau này.
3. Kích hoạt: Một tiến trình có thể kích hoạt một khả năng đã bị tạm thời vô hiệu hóa. Một khả năng đã bị xóa không thể được kích hoạt.

Mà không có khả năng, một chương trình đặc quyền Set-UID cũng có thể xóa/vô hiệu

hóa/kích hoạt đặc quyền của chính nó. Điều này được thực hiện thông qua các cuộc gọi hệ thống `setuid()` và `seteuid()`; cụ thể, một tiến trình có thể thay đổi id người dùng hiệu lực của nó trong thời gian chạy. Tuy nhiên, cách sử dụng cuộc gọi hệ thống này khá thô sơ, vì bạn chỉ có thể là người dùng đặc quyền (ví dụ: root) hoặc là người dùng không đặc quyền. Với khả năng, đặc quyền có thể được điều chỉnh một cách tinh vi hơn nhiều, vì mỗi khả năng có thể được điều chỉnh một cách độc lập.

Để hỗ trợ việc điều chỉnh độ đặc quyền động, Linux sử dụng một cơ chế tương tự cơ chế Set-UID, tức là một tiến trình mang ba bộ sưu tập khả năng: được phép (P), kế thừa (I), và hiệu lực (E). Bộ sưu tập được phép bao gồm các khả năng mà quy trình được phép sử dụng; tuy nhiên, bộ sưu tập này có thể không hoạt động. Bộ sưu tập hiệu lực bao gồm các khả năng mà quy trình có thể sử dụng hiện tại (giống như uid người dùng hiệu lực trong cơ chế Set-UID). Bộ sưu tập hiệu lực phải luôn là một phần của bộ sưu tập được phép. Tiến trình có thể thay đổi nội dung của bộ sưu tập hiệu lực bất kỳ lúc nào miễn là bộ sưu tập hiệu lực không vượt quá bộ sưu tập được phép. Bộ sưu tập kế thừa chỉ được sử dụng để tính toán các bộ sưu tập khả năng mới sau `exec()`, tức là khả năng nào có thể được kế thừa bởi tiến trình con.

Xem lại chương trình `mypriv.c` trong thư mục `home`. Trả lời các câu hỏi được nhúng trong mã nguồn của chương trình. Sau đó, sử dụng kịch bản `build.sh` để biên dịch và liên kết chương trình. Sau đó, sử dụng `setcap` để đặt khả năng `CAP_DAC_READ_SEARCH` và chạy chương trình. So sánh kết quả với những gì bạn mong đợi.



```
ubuntu@capabilities: ~  
File Edit View Search Terminal Help  
^C  
0 packets captured  
0 packets received by filter  
0 packets dropped by kernel  
ubuntu@capabilities:~$ cat mypriv.c  
/* mypriv.c  
Simple example of dropping privileges within a  
program that has been granted the  
CAP_DAC_READ_SEARCH capability.  
*/  
#include <fcntl.h>  
#include <sys/types.h>  
#include <errno.h>  
#include <stdlib.h>  
#include <stdio.h>  
#include <linux/capability.h>  
#include <sys/capability.h>  
int cap_disable(cap_value_t capflag);  
int cap_enable(cap_value_t capflag);  
int cap_drop(cap_value_t capflag);  
int is_cap_set(cap_value_t capflag);  
int main(void)  
{  
    if (is_cap_set(CAP_DAC_READ_SEARCH) == 0) {  
        printf("CAP_DAC_READ_SEARCH capability is not set for this program\n");  
        return -1;  
    }  
    if (open("/etc/shadow", O_RDONLY) < 0)  
        printf("(a) Open failed\n");  
    /* Question (a): is the above open successful? why? */  
    if (cap_disable(CAP_DAC_READ_SEARCH) < 0) return -1;  
    if (open("/etc/shadow", O_RDONLY) < 0)  
        printf("(b) Open failed\n");  
    /* Question (b): is the above open successful? why? */  
    if (cap_enable(CAP_DAC_READ_SEARCH) < 0) return -1;  
    if (open("/etc/shadow", O_RDONLY) < 0)  
        printf("(c) Open failed\n");  
    /* Question (c): is the above open successful? why? */  
    if (cap_drop(CAP_DAC_READ_SEARCH) < 0) return -1;
```



```
ubuntu@capabilities: ~  
File Edit View Search Terminal Help  
/* Question (e): is the above open successful? why?*/  
}  
int is_cap_set(cap_value_t capflag)  
{  
    cap_t mycaps;  
    mycaps = cap_get_proc();  
    cap_flag_value_t cap_flags_value;  
    cap_get_flag(mycaps, capflag, CAP_EFFECTIVE, &cap_flags_value);  
    if(cap_flags_value == CAP_SET)  
        return 1;  
    else  
        return 0;  
}  
int cap_disable(cap_value_t capflag)  
{  
    cap_t mycaps;  
    mycaps = cap_get_proc();  
    if (mycaps == NULL)  
        return -1;  
    if (cap_set_flag(mycaps, CAP_EFFECTIVE, 1, &capflag, CAP_CLEAR) != 0)  
        return -1;  
    if (cap_set_proc(mycaps) != 0)  
        return -1;  
    return 0;  
}  
int cap_enable(cap_value_t capflag)  
{  
    cap_t mycaps;  
    mycaps = cap_get_proc();  
    if (mycaps == NULL)  
        return -1;  
    if (cap_set_flag(mycaps, CAP_EFFECTIVE, 1, &capflag, CAP_SET) != 0)  
        return -1;  
    if (cap_set_proc(mycaps) != 0)  
        return -1;  
    return 0;  
}  
int cap_drop(cap_value_t capflag)  
{  
    cap_t mycaps;  
    mycaps = cap_get_proc();  
    if (mycaps == NULL)  
        return -1;  
    if (cap_set_flag(mycaps, CAP_EFFECTIVE, 1, &capflag, CAP_CLEAR) != 0)  
        return -1;  
    if (cap_set_flag(mycaps, CAP_PERMITTED, 1, &capflag, CAP_CLEAR) != 0)  
        return -1;  
    if (cap_set_proc(mycaps) != 0)  
        return -1;  
    return 0;  
}  
ubuntu@capabilities:~$
```

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`

```
ubuntu@capabilities: ~
File Edit View Search Terminal Help
GNU nano 2.5.3      File: /home/ubuntu/build.sh      Modified

gcc -c mypriv.c
gcc -o mypriv mypriv.o -lcap
sudo setcap cap_dac_read_search=+ep ~/mypriv

^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos
^X Exit          ^R Read File     ^\ Replace       ^U Uncut Text    ^T To Linter    ^_ Go To Line
```

```
ubuntu@capabilities:~$ nano ~/build.sh
ubuntu@capabilities:~$ chmod +x ~/build.sh
ubuntu@capabilities:~$
```

```
student@ubuntu:~/labtainer/labtainer-student$ checkwork
Results stored in directory: /home/student/labtainer_xfer/capabilities
Labname capabilities

Student      | ping_use_cap | passwd_changed |
=====|=====|=====|
B20DCAT094   | Y            | Y              |
What is automatically assessed for this lab:

A subset of the lab goals.
ping_use_cap: Was setuid disabled on ping, and the net_raw capability set?
passwd_changed: Able to change password without suid on passwd?
```

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.