

3

HỆ THỐNG NHÚNG
MẠNG KHÔNG DÂY

Giới thiệu Netfilter và Iptables

Biên soạn: Phan Trung Phát
Liên hệ: phatpt@uit.edu.vn

Lưu hành nội bộ

A. TỔNG QUAN

1. Mục tiêu

- Giới thiệu về **Netfilter** và **Iptables**.
- Tìm hiểu cách vận hành và thực hiện viết một Netfilter module.
- Hiểu về iptables và cách viết các rules trong iptables.
- Sử dụng được các công cụ kiểm tra mạng **Iperf**, **Ping**.

2. Môi trường thực hành

- Máy ảo Ubuntu 20.04 và 1 máy client bất kỳ (Windows, Linux...).
- Phần mềm Wireshark.

B. Giới thiệu

1. Netfilter

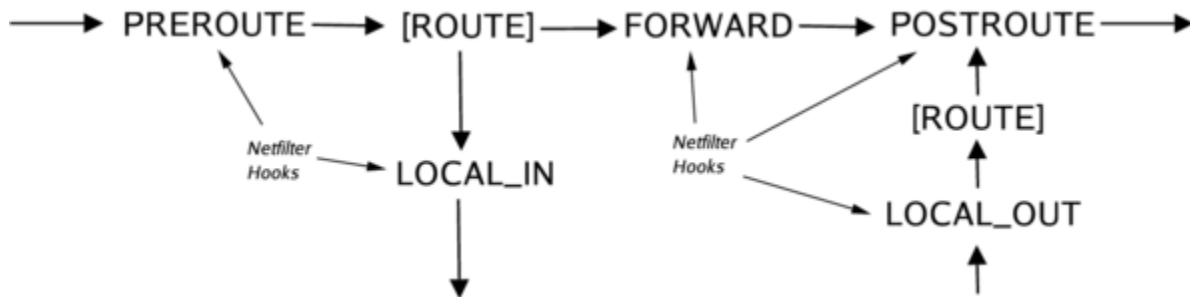
NetFilter là thành phần packet filtering trong Linux kernel. Netfilter chủ yếu cung cấp một tập các hooks trong Linux kernel cho phép các kernel modules có thể đăng ký các hàm trả về (callback) với network stack. Một callback function sẽ được gọi khi mỗi gói tin truyền qua đúng hook mà lúc đầu đã đăng ký trong network stack. Trong bài thực hành này chúng ta sẽ làm việc với gói tin IPv4.

2. Các Netfilter hooks cho IPv4

Có 5 Netfilter hooks mà các chương trình có thể đăng ký. Chúng được định nghĩa trong file *linux/netfilter_ip4.h*. Khi các gói tin đi qua stack, chúng sẽ kích hoạt các kernel modules đã đăng ký với các hook này. Các hooks mà một gói sẽ kích hoạt phụ thuộc vào việc gói đến hay đi, điểm đến của gói và liệu gói có bị loại bỏ hoặc bị từ chối tại điểm trước đó hay không.

Bảng 1. Các sự kiện mà các hooks có thể đăng ký

Hook	Được gọi lúc
NF_INET_PRE_ROUTING	Được xử lý trước khi có bất kỳ quyết định định tuyến nào liên quan đến nơi gửi gói tin.
NF_INET_LOCAL_IN	Sau các quyết định định tuyến nếu các gói tin đi đến host này.
NF_INET_FORWARD	Được kích hoạt sau khi một gói tin đến và được chuyển tiếp đến một host khác.
NF_INET_LOCAL_OUT	Khi gói tin đi ra ngoài.
NF_INET_POST_ROUTING	Được kích hoạt bởi bất kỳ lưu lượng gửi đi hoặc chuyển tiếp, sau khi quá trình định tuyến đã diễn ra và ngay trước khi được gửi đi trên đường truyền.



Hình 1. Thứ tự các sự kiện mà các hooks có thể đăng ký

Sau khi các hàm trả về đã hoàn thành việc xử lý của chúng với gói tin, chúng phải trả về (đưa ra) các quyết định DROP hay ACCEPT gói tin, các quyết định này được định nghĩa sẵn trong Netfilter dưới các code như sau:

Bảng 2. Các mã trả về của các hàm Netfilter callback

Return code	Ý nghĩa
NF_DROP	Hủy gói tin.
NF_ACCEPT	Giữ gói tin, và thực thi tiếp các hooks còn lại.

NF_STOLEN	Tạm quên gói tin, nhưng vẫn giữ gói tin trong resource của kernel.
NF_QUEUE	Đưa gói tin vào hàng đợi.
NF_REPEAT	Gọi hàm trả về này thêm lần nữa.
NF_STOP	Giữ gói tin, bỏ qua các hooks còn lại.

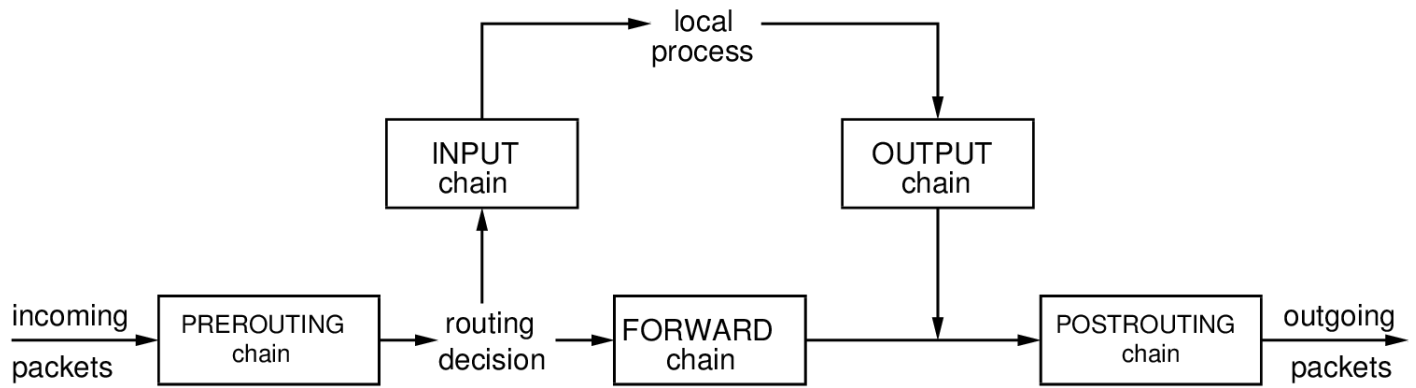
3. Iptables

Iptables là một công cụ tường lửa được sử dụng rộng rãi để giao tiếp với Netfilter packet của Linux kernel. Chúng hoạt động bằng cách tương tác với các hooks trong network stack của Linux kernel. Có thể nói rằng, iptables chịu trách nhiệm giao tiếp với người dùng và sau đó đẩy các luật của người dùng vào cho Netfilter xử lý. Về cơ bản, iptables được xây dựng dựa trên 3 thành phần chính đó là: table, chain và target.

Iptables firewall sử dụng các **tables** để sắp xếp các quy tắc của nó. Các tables này phân loại các quy tắc theo loại quyết định mà chúng được sử dụng để thực hiện. Ví dụ, nếu quy tắc được sử dụng để quyết định có cho phép gói tin tiếp tục đến đích hay không, nó có thể sẽ được thêm vào filter table.

Mỗi iptables table sẽ được gắn thêm các **chain**. Việc gắn thêm chain vào table cho phép xử lý gói tin ở những giai đoạn khác nhau, quyết định việc xử lý gói tin khi nào.

Các chains trong iptables: PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING. Mỗi chain trong iptables tương ứng với các Netfilter hook mà chúng liên kết đến.



Hình 2. Các chains trong iptables

Target hiểu đơn giản là các hành động áp dụng cho các gói tin. Đối với những gói tin đúng theo rule mà chúng ta đặt ra thì các hành động có thể thực hiện được đó là:

Bảng 3. Các loại target khác nhau của iptables

Target	Ý nghĩa
ACCEPT	Chấp nhận gói tin, cho phép gói tin đi vào hệ thống.
DROP	Loại bỏ gói tin, xem như chúng chưa tồn tại.
REJECT	Loại bỏ gói tin nhưng có trả lời table gói tin khác.
LOG	Chấp nhận gói tin nhưng có ghi lại log.

C. THỰC HÀNH

1. Cài đặt Netfilter và chạy thử một module cơ bản

Tiến hành cài đặt các gói thư viện trên Ubuntu.

```

$ sudo apt-get install build-essential module-assistant iperf flex bison
$ sudo apt install linux-headers-generic
$ sudo reboot
  
```

Khởi động lại giữa phím **Shift** nếu không hiện vào Bootmenu GRUB.

```
GNU GRUB version 2.04

Ubuntu
*Advanced options for Ubuntu
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.
```

```
GNU GRUB version 2.04

*Ubuntu, with Linux 5.15.0-52-generic
Ubuntu, with Linux 5.15.0-52-generic (recovery mode)
Ubuntu, with Linux 5.15.0-48-generic
Ubuntu, with Linux 5.15.0-48-generic (recovery mode)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line. ESC to return previous
menu.
```

Di chuyển file **Makefile** và **nkmod.c** được cung cấp tại website môn học vào chung 1 thư mục, thực thi lệnh:

```
$ sudo make
```

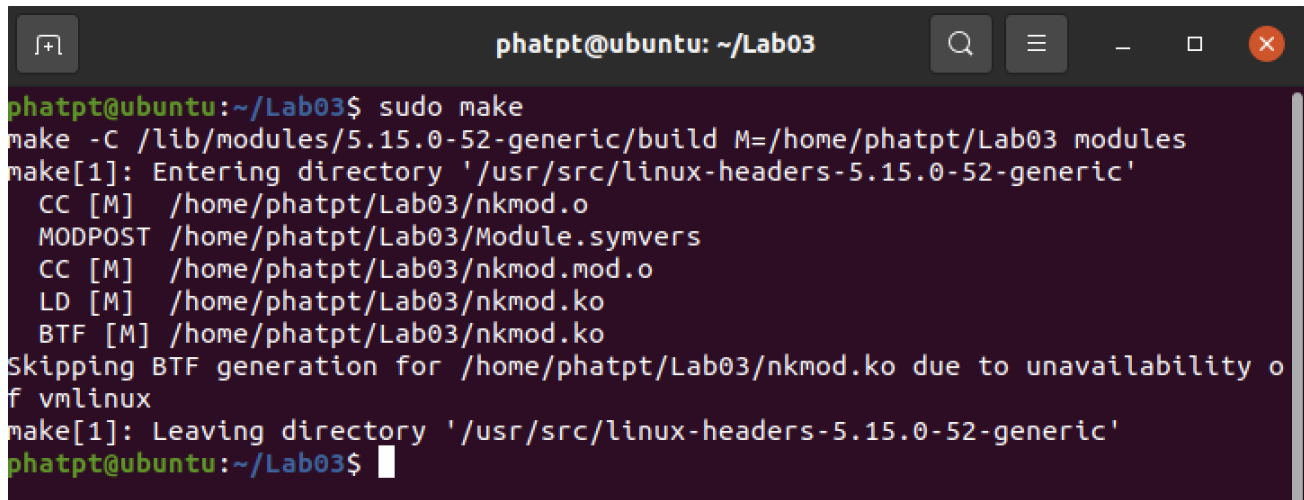
Lưu ý: các khoảng trắng và tab trong **Makefile**:

```
obj-m += nkmod.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

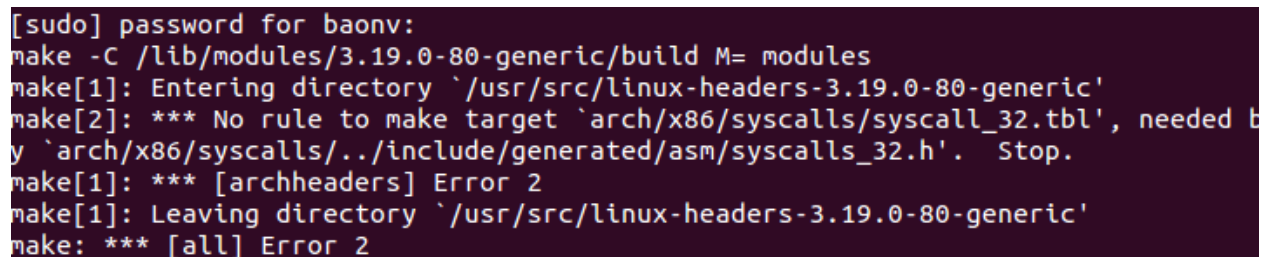
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

Kết quả:



```
phatpt@ubuntu: ~/Lab03
phatpt@ubuntu:~/Lab03$ sudo make
make -C /lib/modules/5.15.0-52-generic/build M=/home/phatpt/Lab03 modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-52-generic'
CC [M] /home/phatpt/Lab03/nkmod.o
MODPOST /home/phatpt/Lab03/Module.symvers
CC [M] /home/phatpt/Lab03/nkmod.mod.o
LD [M] /home/phatpt/Lab03/nkmod.ko
BTF [M] /home/phatpt/Lab03/nkmod.ko
Skipping BTF generation for /home/phatpt/Lab03/nkmod.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-52-generic'
phatpt@ubuntu:~/Lab03$
```

Lưu ý: trong trường hợp bị lỗi như hình dưới thì sửa lại **Makefile**:



```
[sudo] password for baonv:
make -C /lib/modules/3.19.0-80-generic/build M= modules
make[1]: Entering directory '/usr/src/linux-headers-3.19.0-80-generic'
make[2]: *** No rule to make target 'arch/x86/syscalls/syscall_32.tbl', needed by 'arch/x86/syscalls/./include/generated/asm/syscalls_32.h'. Stop.
make[1]: *** [archheaders] Error 2
make[1]: Leaving directory '/usr/src/linux-headers-3.19.0-80-generic'
make: *** [all] Error 2
```

```

Makefile x
obj-m += nkmod.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(shell pwd) modules

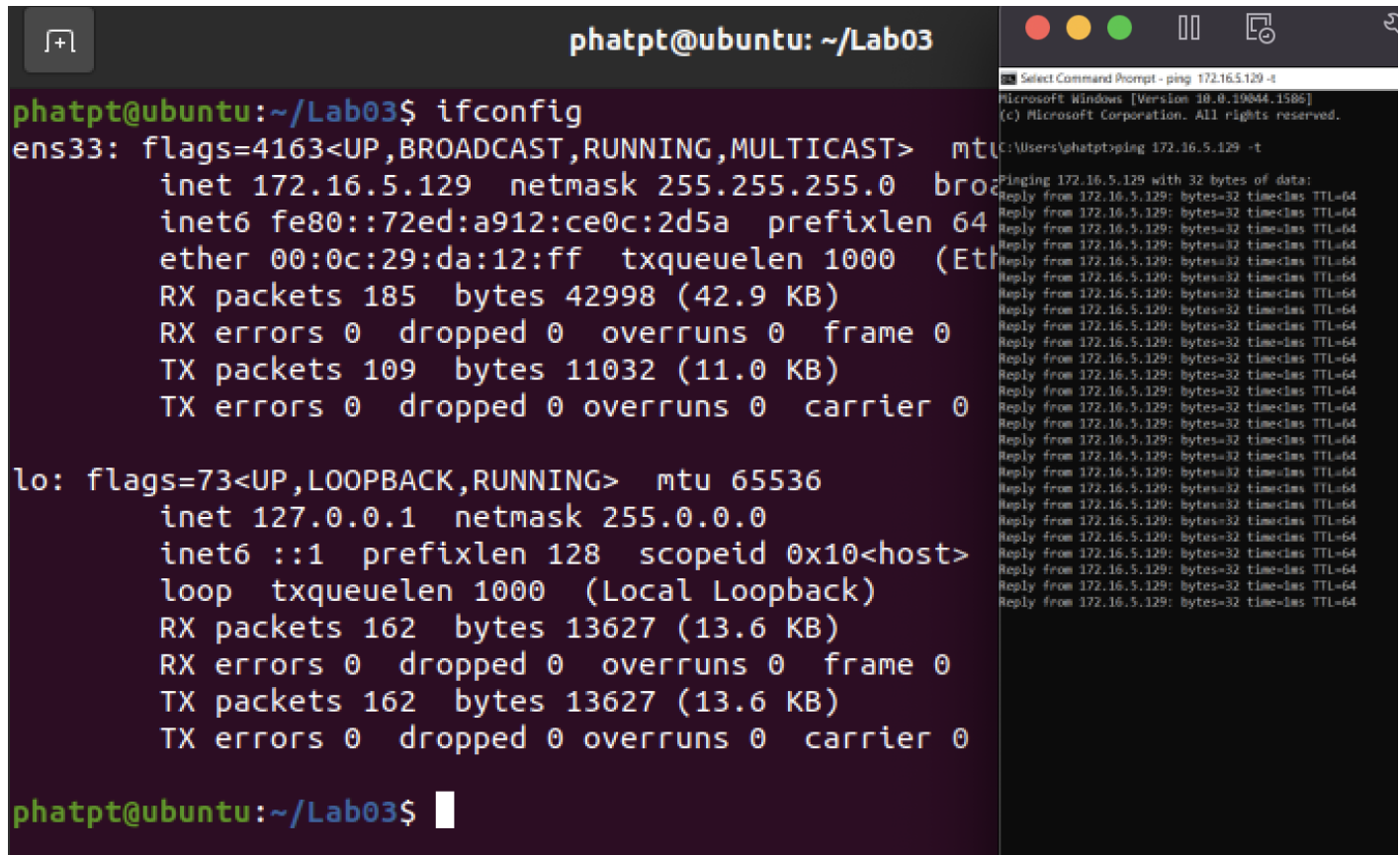
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean

```

Lưu ý: Trong trường hợp build bị lỗi nếu thiếu linux-header, thì cài đặt gói thư viện sau:

```
$ sudo apt-get install linux-headers-`uname -r`
```

Sau khi build xong dùng máy còn lại ping đến máy Ubuntu:



The screenshot shows two windows. The main window is a terminal on a Ubuntu machine with IP 172.16.5.129. The user runs the command `ifconfig`, displaying details for the `ens33` and `lo` interfaces. The `ens33` interface is configured with IP 172.16.5.129 and netmask 255.255.255.0. The `lo` interface is the loopback address 127.0.0.1. A second window, titled 'Select Command Prompt - ping 172.16.5.129 -t', shows a continuous stream of ping replies from 172.16.5.129, indicating successful connectivity.

Dùng lệnh **insmod** để cài đặt Netfilter module này vào kernel:

```
$ sudo insmod nkmod.ko
```


Kiểm tra log, sử dụng lệnh:

```
$ dmesg | tail
```

Dùng máy còn lại ping đến máy Ubuntu một lần nữa.

→ Q1: Quan sát kết quả và trình bày kết quả thu được?

→ Q2: Giải thích hiện tượng thu được ở trên?

Dùng lệnh **rmmod** để unregister module:

```
$ sudo rmmod nkmod.ko
```

2. Kiểm tra cài đặt và thử nghiệm iptables

Thông thường, iptables được cài đặt sẵn trong các bản phân phối của Linux. Để kiểm tra iptables đã được cài đặt hay chưa, sử dụng lệnh:

```
$ iptables --version
```

Nếu chưa được cài đặt, sử dụng lệnh sau để cài đặt:

```
$ sudo apt-get update
```

```
$ sudo apt-get install iptables
```

Liệt kê tất cả các rules (tham số -L) và thể hiện thêm các danh sách bổ trợ:

```
$ sudo iptables -L -v
```

Cấu trúc của lệnh iptables ¹:

```
$ sudo iptables <action with chains/rules> -t <table> -p <protocol> -i <input interface> -  
o <output interface> -s <IP source> -d <IP destination> --sport <source port no.> --  
dport <destination port no.> -j <target>
```

¹ Tham khảo thêm tại: <https://linux.die.net/man/8/iptables>

Dùng máy còn lại ping đến máy Ubuntu.

Sử dụng lệnh sau để chặn không cho máy còn lại ping đến máy Ubuntu:

```
$ sudo iptables -A INPUT -p icmp -s 192.168.20.22 -j DROP
```

Dùng máy còn lại ping một lần nữa đến máy Ubuntu để kiểm tra.

Thông thường, các iptables rules chúng ta tạo ra sẽ được lưu lại trong bộ nhớ, tuy nhiên nếu máy được reboot thì chúng ta cần phải thực hiện lại. Vì thế, cần phải lưu vào cấu hình hệ thống, sử dụng lệnh:

```
$ sudo /sbin/iptables-save
```

Để tắt iptables firewall:

```
$ service iptable stop
```

Để xóa tất cả các rules, sử dụng lệnh:

```
$ sudo iptables -F
```

Để xóa từng rules khác nhau, đầu tiên cần sử dụng lệnh để hiển thị số thứ tự và chain của chúng:

```
$ sudo iptables -L --line-numbers
```

```
phatpt@ubuntu:~/Desktop$ sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num target      prot opt source                destination
1    DROP          icmp -- 172.16.5.128           anywhere

Chain FORWARD (policy ACCEPT)
num target      prot opt source                destination

Chain OUTPUT (policy ACCEPT)
num target      prot opt source                destination
phatpt@ubuntu:~/Desktop$
```

Để xoá được rule số 1 ở chain INPUT, sử dụng lệnh:

```
$ sudo iptables -D INPUT 1
```

D. YÊU CẦU & ĐÁNH GIÁ

1. Yêu cầu

Đối với nội dung làm quen với viết Netfilter module, sinh viên thực hiện các yêu cầu sau:

1. Sửa lại file **nkmod.c** để hook chỉ DROP các gói tin UDP.

Gợi ý:

Tham khảo Protocol code: <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>

Test connection UDP:

Trên máy Ubuntu khởi động UDP server bằng lệnh:

```
$ iperf -s -u -i 1
```

Máy còn lại bắt đầu UDP client để kết nối tới máy Ubuntu, sử dụng lệnh:

```
$ iperf -c <ip_máy_ubuntu> -u
```

Test TCP connection ở port 80:

Trên máy Ubuntu, khởi động TCP server bằng lệnh:

```
$ iperf -s -i 1 -p 80
```

Máy còn lại bắt đầu UDP client để kết nối tới máy Ubuntu bằng lệnh:

```
$ iperf -c <ip_máy_ubuntu> -p 80
```

2. DROP các packet có source IP đến từ địa chỉ máy client, chẳng hạn như 192.168.20.22, ACCEPT tất cả các packet còn lại.

Gợi ý:

Sử dụng cách sau để parse kiểu uint32_t của iphdr→saddr qua char array:

```
char target[16]="192.168.20.22";
snprintf(sipaddr, 16, "%pI4", &ip_header->saddr);
```

Sử dụng memcmp để so sánh:

```
memcmp(&sipaddr, &target, sizeof(sipaddr))
```

3. DROP các gói tin TCP và UDP đi đến với địa chỉ port đích là 80.

4. Tìm cách để in ra địa chỉ nguồn và địa chỉ đích của các gói tin khi thực hiện một tác vụ bất kỳ.

```
phatpt@ubuntu: ~/Desktop
phatpt@ubuntu:~/Desktop$ ping google.com
PING google.com (142.251.220.46) 56(84) bytes of data.
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=1 ttl=128 tim
e=34.1 ms
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=2 ttl=128 tim
e=44.3 ms
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=3 ttl=128 tim
e=91.9 ms
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=4 ttl=128 tim
e=106 ms
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=5 ttl=128 tim
e=33.6 ms
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=6 ttl=128 tim
e=35.6 ms
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=7 ttl=128 tim
e=34.2 ms
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=8 ttl=128 tim
e=34.7 ms
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=9 ttl=128 tim
e=34.7 ms
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=10 ttl=128 ti
me=34.5 ms
64 bytes from hkg07s50-in-f14.1e100.net (142.251.220.46): icmp_seq=11 ttl=128 ti
me=76.0 ms

phatpt@ubuntu: ~/Lab03
phatpt@ubuntu:~/Lab03$ netstat -tlnp | tail
79.518472] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
70.502726] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
71.504604] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
72.506973] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
73.507968] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
74.509879] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
75.510550] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
76.512810] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
77.514255] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
78.515860] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
79.518472] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
71.504604] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
72.506973] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
73.507968] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
74.509879] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
75.510550] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
76.512810] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
77.514255] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
78.515860] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
79.518472] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
30.520233] My IP address = 172.16.5.129 AND Your IP address = 142.251.220.46
```

5. Chỉ cho phép các gói tin đi đến **uit.edu.vn** (*biết trước địa chỉ IP*) đi ra ngoài, DROP tất cả còn lại.

Đối với nội dung làm quen với viết iptables, sinh viên thực hiện các yêu cầu sau:

6. Chặn các gói tin Echo Request đến từ máy client.
7. Cấm tất cả các gói tin UDP và TCP có port đích là 80 đến từ bất kỳ máy tính nào.
8. Xoá tất cả rules, thiết đặt chặn các traffic từ máy Ubuntu đến địa chỉ IP máy client và ngược lại, các hoạt động khác bình thường.
9. Xoá rule câu số 8, Cấm các gói tin đến từ dãy IP chứa máy client, chẳng hạn như 192.168.20.0/24 với IP máy client là 192.168.20.22. Điều gì sẽ xảy ra khi thiết đặt lệnh này?
10. Để giải quyết điều đặt biệt từ câu số 9, chèn thêm 1 rule để địa chỉ IP Default Gateway có thể giao tiếp bình thường với máy Ubuntu. Kiểm tra lại kết quả thu được.
11. Xoá rule câu số 10, thiết lập lệnh không cho phép client SSH vào máy Ubuntu. Hãy chứng minh điều đó?
12. Tìm hiểu và trình bày về cuộc tấn công Deny of Service (DoS), thiết lập rule để phòng chống cuộc tấn công này.

Lưu ý: Khi thực hiện xong các ý, phải có bước kiểm tra lại kết quả bằng **iperf** hoặc lệnh **ping** hoặc các phương thức khác dựa trên đặt trưng của câu hỏi. Trình bày cách thức thực hiện và giải thích kết quả thu được.

2. Đánh giá

- Sinh viên tìm hiểu và thực hành theo hướng dẫn. Thực hiện theo **nhóm**.
- Sinh viên báo cáo kết quả thực hiện và nộp bài bằng file. Trong đó:
 - Trình bày chi tiết quá trình thực hành và trả lời các câu hỏi nếu có (kèm theo các ảnh chụp màn hình tương ứng).
 - Giải thích, tìm hiểu, lý giải các kết quả đạt được.
 - Tải mẫu báo cáo thực hành và trình bày theo mẫu được cung cấp.

Nén **.ZIP** tất cả các file và đặt tên file theo định dạng theo mẫu:

NhomY-LabX_MSSV1_MSSV2

Ví dụ: Nhom1-Lab03_20520001_20520002

- Nộp báo cáo trên theo thời gian đã thống nhất tại website môn học.
- Các bài nộp không tuân theo yêu cầu sẽ **KHÔNG** được chấm điểm.

HẾT

Chúc các bạn hoàn thành tốt bài thực hành!