

# BÁO CÁO BÀI TẬP

Môn học: An toàn mạng

Tên chủ đề: Firewall Exploration Lab

GVHD: Nghi Hoàng Khoa

a. **THÔNG TIN CHUNG:**

Lớp: NT140.P11.ANTT

STT	Họ và tên	MSSV	Email
1	Từ Chí Kiên	22520713	22520713@gm.uit.edu.vn
2	Nguyễn Thanh Hưng	22520519	22520519@gm.uit.edu.vn
3	Hà Minh Quân	22521177	22521177@gm.uit.edu.vn

b. **NỘI DUNG THỰC HIỆN:**

STT	Công việc	Kết quả tự đánh giá
1	Task 1	100%
2	Task 2	100%
3	Task 3	100%
4	Task 4	100%
5	Task 5	100%

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết.

# BÁO CÁO CHI TIẾT

## Mục lục:

A.	Container Setup and Commands .....	2
B.	Task 1: Implementing a Simple Firewall .....	3
1.	Task 1.A: Implement a Simple Kernel Module .....	3
2.	Task 1.B: Implement a Simple Firewall UsingNetfilter .....	4
a)	Task 1 .....	5
b)	Task 2 .....	6
c)	Task 3 .....	9
C.	Task 2: Experimenting with Stateless Firewall Rules .....	12
3.	Task 2.A: Protecting the Router .....	12
4.	Task 2.B: Protecting the Internal Network .....	13
5.	Task 2.C: Protecting Internal Servers .....	14
D.	Task 3: Connection Tracking and Stateful Firewall .....	16
6.	Task 3.A: Experiment with the Connection Tracking .....	16
7.	Task 3.B: Setting Up a Stateful Firewall .....	17
E.	Task 4: Limiting Network Traffic .....	19
F.	Task 5: Load Balancing .....	20

### A. Container Setup and Commands

Đầu tiên tải tệp zip trên trang xuống:

iplabfiles.

#### Tasks (PDF)

- **VM version:** This lab has been tested on our [SEED Ubuntu-20.04 VM](#)
- **Lab setup files**
  - [Labsetup.zip](#)
  - [Labsetup-arm.zip](#) (for Apple Silicon machines)
- **Manual:** [Docker manual](#)

#### Time (Suggested)

Giải nén thì thấy có ba thư mục volumes, router, Files và một tệp yml

```
[12/14/24] seed@VM:~/.../Labsetup$ ls
docker-compose.yml  Files  router  volumes
```

## B. Task 1: Implementing a Simple Firewall

### 1. Task 1.A: Implement a Simple Kernel Module

Trong lab đã cung cấp cho tệp hello.c và tệp make để tạo

```
[12/14/24]seed@VM:~/.../Labsetup$ ls
docker-compose.yml  Files  router  volumes
[12/14/24]seed@VM:~/.../Labsetup$ cd Files
[12/14/24]seed@VM:~/.../Files$ ls
kernel_module  packet_filter
[12/14/24]seed@VM:~/.../Files$ cd kernel_module
[12/14/24]seed@VM:~/.../kernel_module$ ls
hello.c  Makefile
[12/14/24]seed@VM:~/.../kernel_module$ █
```

Trong đó tệp hello.c sẽ thực hiện ghi vào kernel khi tạo và xóa:

```
[12/14/24]seed@VM:~/.../kernel_module$ cat hello.c
#include <linux/module.h>
#include <linux/kernel.h>

int initialization(void)
{
    printk(KERN_INFO "Hello World!\n");
    return 0;
}

void cleanup(void)
{
    printk(KERN_INFO "Bye-bye World!.\n");
}

module_init(initialization);
module_exit(cleanup);
```

Trong tệp Makefile sẽ thực hiện biên dịch tệp trên

```
[12/14/24]seed@VM:~/.../kernel_module$ cat Makefile
obj-m += hello.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
```

Sử dụng lệnh make để biên dịch mã

```
[12/14/24]seed@VM:~/.../kernel_module$ make
make -C /lib/modules/5.4.0-54-generic/build M=/home/seed/Documents/Lab/firewall/
Labsetup/Files/kernel_module modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-54-generic'
  CC [M]  /home/seed/Documents/Lab/firewall/Labsetup/Files/kernel_module/hello.o
Building modules, stage 2.
MODPOST 1 modules
  CC [M]  /home/seed/Documents/Lab/firewall/Labsetup/Files/kernel_module/hello.m
od.o
  LD [M]  /home/seed/Documents/Lab/firewall/Labsetup/Files/kernel_module/hello.k
o
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-54-generic'
[12/14/24]seed@VM:~/.../kernel_module$ ll
total 32
-rw-rw-r-- 1 seed seed 279 Mar 14 2022 hello.c
-rw-rw-r-- 1 seed seed 3968 Dec 14 07:53 hello.ko
-rw-rw-r-- 1 seed seed 72 Dec 14 07:53 hello.mod
-rw-rw-r-- 1 seed seed 560 Dec 14 07:53 hello.mod.c
-rw-rw-r-- 1 seed seed 2800 Dec 14 07:53 hello.mod.o
-rw-rw-r-- 1 seed seed 2048 Dec 14 07:53 hello.o
-rw-rw-r-- 1 seed seed 156 Jan 13 2021 Makefile
-rw-rw-r-- 1 seed seed 72 Dec 14 07:53 modules.order
-rw-rw-r-- 1 seed seed 0 Dec 14 07:53 Module.symvers
[12/14/24]seed@VM:~/.../kernel_module$ █
```

Tiến hành thử nghiệm bằng cách thêm module hello qua tệp hello.ko, khi kiểm tra bằng lệnh lsmod thì thấy module hello được thêm vào.

rmmmod hello cho biết xóa module trên.

Cuối cùng sử dụng dmesg để xuất thông tin trong kernel

```
[12/14/24]seed@VM:~/.../kernel_module$ sudo insmod hello.ko
[12/14/24]seed@VM:~/.../kernel_module$ lsmod | grep hello
hello                  16384  0
[12/14/24]seed@VM:~/.../kernel_module$ sudo rmmmod hello
[12/14/24]seed@VM:~/.../kernel_module$ dmesg
```

Ở đây sẽ thấy Hello World! Và Bye-bye World!

```
default. Update your scripts to load br_netfilter if you need this.
[ 15.302678] Bridge firewalls registered
[ 15.373621] bpfILTER: Loaded bpfILTER_umh pid 1175
[ 15.375078] Started bpfILTER
[ 16.182570] Initializing XFRM netlink socket
[ 17.959851] rfkill: input handler disabled
[ 56.460614] rfkill: input handler enabled
[ 97.873998] rfkill: input handler disabled
[ 4759.953601] hello: module verification failed: signature and/or required key
missing - tainting kernel
[ 4759.970108] Hello World!
[ 4776.992019] Bye-bye World!.
```

## 2. Task 1.B: Implement a Simple Firewall UsingNetfilter

Trong lab đã cung cấp cho tệp seedFilter.c và tệp make để tạo

```
[12/14/24] seed@VM:~/.../Files$ cd packet_filter
[12/14/24] seed@VM:~/.../packet_filter$ ls
Makefile  seedFilter.c
[12/14/24] seed@VM:~/.../packet_filter$
```

Tương tự như task trên sử dụng make để biên dịch chương trình

```
[12/14/24] seed@VM:~/.../packet_filter$ ll
total 40
-rw-rw-r-- 1 seed seed 236 Jan 13 2021 Makefile
-rw-rw-r-- 1 seed seed 77 Dec 14 10:35 modules.order
-rw-rw-r-- 1 seed seed 0 Dec 14 10:35 Module.symvers
-rw-rw-r-- 1 seed seed 2746 Jan 13 2021 seedFilter.c
-rw-rw-r-- 1 seed seed 7088 Dec 14 10:35 seedFilter.ko
-rw-rw-r-- 1 seed seed 77 Dec 14 10:35 seedFilter.mod
-rw-rw-r-- 1 seed seed 560 Dec 14 10:35 seedFilter.mod.c
-rw-rw-r-- 1 seed seed 2808 Dec 14 10:35 seedFilter.mod.o
-rw-rw-r-- 1 seed seed 5160 Dec 14 10:35 seedFilter.o
[12/14/24] seed@VM:~/.../packet_filter$
```

### a) Task 1

Trong bài lab có cung cấp một chương trình seedFilter.c:

```
unsigned int blockUDP(void *priv, struct sk_buff *skb,
                      const struct nf_hook_state *state)
{
    struct iphdr *iph;
    struct udphdr *udph;

    u16 port = 53;
    char ip[16] = "8.8.8.8";
    u32 ip_addr;

    if (!skb) return NF_ACCEPT;

    iph = ip_hdr(skb);
    // Convert the IPv4 address from dotted decimal to 32-bit binary
    in4_pton(ip, -1, (u8 *)&ip_addr, '\0', NULL);

    if (iph->protocol == IPPROTO_UDP) {
        udph = udp_hdr(skb);
        if (iph->daddr == ip_addr && ntohs(udph->dest) == port){
            printk(KERN_WARNING "*** Dropping %pI4 (UDP), port %d\n",
                  &(iph->daddr), port);
            return NF_DROP;
        }
    }
}
```

Chương trình trên sẽ chặn truy cập đến ip 8.8.8.8 qua cổng 53, tương ứng với truy xuất dns của một trang qua ip 8.8.8.8 của google.

Trước khi mở tường lửa thì có thể truy xuất dns của example.com

---

```
[12/14/24] seed@VM:~/.../packet_filter$ dig @8.8.8.8 www.example.com

; <>> DiG 9.16.1-Ubuntu <>> @8.8.8.8 www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63579
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.example.com.           IN      A

;; ANSWER SECTION:
www.example.com.      3472    IN      A      93.184.215.14

;; Query time: 23 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Sun Dec 15 00:13:56 EST 2024
;; MSG SIZE rcvd: 60
```

Thêm module trên vào kernel

```
[12/15/24] seed@VM:~/.../packet_filter$ sudo insmod seedFilter.ko
[12/15/24] seed@VM:~/.../packet_filter$ lsmod | grep seedFilter
seedFilter            16384  0
```

Sau khi thêm thì không truy xuất dns của example.com nữa:

```
[12/15/24] seed@VM:~/.../packet_filter$ dig @8.8.8.8 www.example.com

; <>> DiG 9.16.1-Ubuntu <>> @8.8.8.8 www.example.com
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached
```

### b) Task 2

Sử dụng lại chương trình seedFilter.c trên với một số chỉnh sửa:

Thêm biến hook3, hook4, hook5.

```
static struct nf_hook_ops hook1, hook2, hook3, hook4, hook5;
```

Trong hàm registerFilter, thay phần .hook của từng hook thành hàm printInfo

Còn ở phần .hooknum sẽ thay bằng từng macros được cung cấp



```

int registerFilter(void) {
    printk(KERN_INFO "Registering filters.\n");

    //NF_INET_PRE_ROUTING
    hook1.hook = printInfo;
    hook1.hooknum = NF_INET_PRE_ROUTING;
    hook1(pf = PF_INET;
    hook1.priority = NF_IP_PRI_FIRST;
    nf_register_net_hook(&init_net, &hook1);

    //NF_INET_LOCAL_IN
    hook2.hook = printInfo;
    hook2.hooknum = NF_INET_LOCAL_IN;
    hook2(pf = PF_INET;
    hook2.priority = NF_IP_PRI_FIRST;
    nf_register_net_hook(&init_net, &hook2);

    //NF_INET_FORWARD
    hook3.hook = printInfo;
    hook3.hooknum = NF_INET_FORWARD;
    hook3(pf = PF_INET;
    hook3.priority = NF_IP_PRI_FIRST;
    nf_register_net_hook(&init_net, &hook3);
    ——————
}

//NF_INET_LOCAL_OUT
hook4.hook = printInfo;
hook4.hooknum = NF_INET_LOCAL_OUT;
hook4(pf = PF_INET;
hook4.priority = NF_IP_PRI_FIRST;
nf_register_net_hook(&init_net, &hook4);

//NF_INET_POST_ROUTING
hook5.hook = printInfo;
hook5.hooknum = NF_INET_POST_ROUTING;
hook5(pf = PF_INET;
hook5.priority = NF_IP_PRI_FIRST;
nf_register_net_hook(&init_net, &hook5);

```

Ở hàm removeFilter thì thêm các hook tạo thêm

```

void removeFilter(void) {
    printk(KERN_INFO "The filters are being removed.\n");
    nf_unregister_net_hook(&init_net, &hook1);
    nf_unregister_net_hook(&init_net, &hook2);
    nf_unregister_net_hook(&init_net, &hook3);
    nf_unregister_net_hook(&init_net, &hook4);
    nf_unregister_net_hook(&init_net, &hook5);
}

```

Sử lại tệp make



```

1 #obj-m += seedFilter.o
2 obj-m += seedPrinter.o
3 all:
4     make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
5
6 clean:
7     make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
8
9 ins:
10    sudo dmesg -C
11    sudo insmod seedPrinter.ko
12
13 rm:
14    sudo rmmod seedPrinter
15

```

Biên dịch dùng make:

```
[12/15/24]seed@VM:~/.../packet_printer$ ll
total 40
-rw-rw-r-- 1 seed seed 262 Dec 15 09:38 Makefile
-rw-rw-r-- 1 seed seed 79 Dec 15 09:44 modules.order
-rw-rw-r-- 1 seed seed 0 Dec 15 09:44 Module.symvers
-rw-rw-r-- 1 seed seed 3547 Dec 15 09:24 seedPrinter.c
-rw-rw-r-- 1 seed seed 8096 Dec 15 09:44 seedPrinter.ko
-rw-rw-r-- 1 seed seed 79 Dec 15 09:44 seedPrinter.mod
-rw-rw-r-- 1 seed seed 560 Dec 15 09:44 seedPrinter.mod.c
-rw-rw-r-- 1 seed seed 2808 Dec 15 09:44 seedPrinter.mod.o
-rw-rw-r-- 1 seed seed 6200 Dec 15 09:44 seedPrinter.o
```

Sử dụng dmesg -k -w để theo dõi kernel

```
[12/15/24]seed@VM:~/.../packet_printer$ dmesg -k -w
[    0.000000] Linux version 5.4.0-54-generic (buildd@lcy01-amd64-024) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)) #60-Ubuntu SMP Fri Nov 6 10:37:59 UTC 2020 (Ubuntu 5.4.0-54.60-generic 5.4.65)
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.0-54-generic root=UUID=a91f1a43-2770-4684-9fc3-b7abfd786c1d ro quiet splash
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
```

Thêm tệp ko vào

```
[12/15/24]seed@VM:~/.../packet_printer$ sudo insmod seedPrinter.ko
[12/15/24]seed@VM:~/.../packet_printer$ lsmod | grep seedPrinter
seedPrinter      16384  0
[12/15/24]seed@VM:~/.../packet_printer$
```

Truy cập đến dns của example.com

```
[12/15/24] seed@VM:~/.../packet_printer$ dig @8.8.8.8 www.example.com

; <>> DiG 9.16.1-Ubuntu <>> @8.8.8.8 www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15718
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.example.com.           IN      A

;; ANSWER SECTION:
www.example.com.      708      IN      A      93.184.215.14

;; Query time: 28 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Sun Dec 15 10:00:01 EST 2024
;; MSG SIZE rcvd: 60
```

Sẽ có kết quả sau:

```
[13702.318428] *** PRE_ROUTING
[13702.318429]   8.8.8.8 --> 10.0.2.15 (UDP)
[13702.318439] *** LOCAL_IN
[13702.318439]   8.8.8.8 --> 10.0.2.15 (UDP)
[13742.132321] *** LOCAL_OUT
[13742.132323]   10.0.2.15 --> 10.0.2.3 (UDP)
[13742.132337] *** POST_ROUTING
[13742.132383]   10.0.2.15 --> 10.0.2.3 (UDP)
[13742.148027] *** PRE_ROUTING
[13742.148029]   10.0.2.3 --> 10.0.2.15 (UDP)
[13742.148041] *** LOCAL_IN
[13742.148041]   10.0.2.3 --> 10.0.2.15 (UDP)
```

### c) Task 3

Trước khi chặn, ping và telnet từ máy 10.9.0.5 đến 10.9.0.1:

```

root@1d304830ddf7:/# ping 10.9.0.1
PING 10.9.0.1 (10.9.0.1) 56(84) bytes of data.
64 bytes from 10.9.0.1: icmp_seq=1 ttl=64 time=0.222 ms
^C
--- 10.9.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.222/0.222/0.222/0.000 ms
root@1d304830ddf7:/# telnet 10.9.0.1
Trying 10.9.0.1...
Connected to 10.9.0.1.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
VM login: seed
Password:
/usr/lib/update-notifier/update-motd-fsck-at-reboot[:64: integer expression expected:          0
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.

```

Sử dụng lại tệp chương trình trên xóa các hàm print và blockUDP, tạo 2 hàm mới để chặn  
Hàm chặn ping đến 10.9.0.1:

```

unsigned int blockPing(void *priv, struct sk_buff *skb,
                      const struct nf_hook_state *state)
{
    struct iphdr *iph;
    char vm_ip[16] = "10.9.0.1";
    u32 vm_ip_addr;

    if (!skb)
        return NF_ACCEPT;

    iph = ip_hdr(skb);
    // Convert the IPv4 address from dotted decimal to 32-bit binary
    in4_pton(vm_ip, -1, (u8 *)&vm_ip_addr, '\0', NULL);

    // Check if the protocol is ICMP and the destination is the VM's IP
    if (iph->protocol == IPPROTO_ICMP && iph->daddr == vm_ip_addr) {
        printk(KERN_WARNING "*** Dropping ICMP Echo Request to %pI4\n", &(iph->daddr));
        return NF_DROP;
    }

    return NF_ACCEPT;
}

```

Hàm chặn telnet 10.9.0.1:

```

unsigned int blockTelnet(void *priv, struct sk_buff *skb,
                        const struct nf_hook_state *state)
{
    struct iphdr *iph;
    struct tcphdr *tcp;
    char vm_ip[16] = "10.9.0.1";
    u32 vm_ip_addr;

    if (!skb)
        return NF_ACCEPT;

    iph = ip_hdr(skb);
    // Convert the IPv4 address from dotted decimal to 32-bit binary
    in4_pton(vm_ip, -1, (u8 *)&vm_ip_addr, '\0', NULL);

    if (iph->protocol == IPPROTO_TCP) {
        tcp = tcp_hdr(skb);
        // Check if the destination is the VM's IP and the port is Telnet (23)
        if (iph->daddr == vm_ip_addr && ntohs(tcp->dest) == 23) {
            printk(KERN_WARNING "*** Dropping Telnet connection to %pI4 (port 23)\n", &(iph->daddr));
            return NF_DROP;
        }
    }

    return NF_ACCEPT;
}

```

Tạo 2 hook:

```
static struct nf_hook_ops hook_ping, hook_telnet;
```

Register 2 hook đó:

```

int registerFilter(void)
{
    printk(KERN_INFO "Registering filters.\n");

    // NF_INET_PRE_ROUTING for general packet filtering
    hook_ping.hook = blockPing;
    hook_ping.hooknum = NF_INET_PRE_ROUTING;
    hook_ping.pf = PF_INET;
    hook_ping.priority = NF_IP_PRI_FIRST;
    nf_register_net_hook(&init_net, &hook_ping);

    hook_telnet.hook = blockTelnet;
    hook_telnet.hooknum = NF_INET_PRE_ROUTING;
    hook_telnet.pf = PF_INET;
    hook_telnet.priority = NF_IP_PRI_FIRST;
    nf_register_net_hook(&init_net, &hook_telnet);

    return 0;
}

```

Cũng như xóa 2 hook đó nếu loại bỏ

```

void removeFilter(void)
{
    printk(KERN_INFO "The filters are being removed.\n");
    nf_unregister_net_hook(&init_net, &hook_ping);
    nf_unregister_net_hook(&init_net, &hook_telnet);
}

```

Nạp module đó lên kernel

```
[12/15/24] seed@VM:~/.../packet_block$ sudo insmod seedBlock.ko
[12/15/24] seed@VM:~/.../packet_block$ lsmod | grep seedBlock
seedBlock           16384   0
```

Sử dụng dmesg -k -w để xem log:

Vào máy 10.9.0.5 ping và telnet đến 10.9.0.1 thì thấy không được nữa

```
root@bd804180ad42:/# ping 10.9.0.1
PING 10.9.0.1 (10.9.0.1) 56(84) bytes of data.
^C
--- 10.9.0.1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2028ms

root@bd804180ad42:/# telent 10.9.0.1
bash: telent: command not found
root@bd804180ad42:/# telnet 10.9.0.1
Trying 10.9.0.1...
^C
```

Vào bên log thì thấy chặn:

```
[ 4586.578545] Registering filters.
[ 4624.331342] *** Dropping ICMP Echo Request to 10.9.0.1
[ 4625.336990] *** Dropping ICMP Echo Request to 10.9.0.1
[ 4626.359573] *** Dropping ICMP Echo Request to 10.9.0.1
[ 4669.037341] *** Dropping Telnet connection to 10.9.0.1 (port 23)
[ 4670.038384] *** Dropping Telnet connection to 10.9.0.1 (port 23)
```

## C. Task 2: Experimenting with Stateless Firewall Rules

### 3. Task 2.A: Protecting the Router

Cấu hình trên tường lửa để có thể ping đến:

```
root@1e8a5ad520b3:/# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
root@1e8a5ad520b3:/# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
root@1e8a5ad520b3:/# iptables -P OUTPUT DROP
root@1e8a5ad520b3:/# iptables -P INPUT DROP
root@1e8a5ad520b3:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 10.9.0.11 netmask 255.255.255.0 broadcast 10.9.0.255
        ether 02:42:0a:09:00:0b txqueuelen 0 (Ethernet)
          RX packets 76 bytes 8826 (8.8 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 0 bytes 0 (0.0 B)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2 dòng lệnh đầu sẽ nhận gói tin icmp đến và đi

2 dòng lệnh sau sẽ cấu hình tường lửa mặc định là chặn các loại gói khác ở đầu vào và ra

Ở máy A thử ping đến router thì được:

```
[12/16/24] seed@VM:~/.../Labsetup$ docksh hostA-10.9.0.5
root@cb29c9cc6961:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.762 ms
^C
--- 10.9.0.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.762/0.762/0.762/0.000 ms
```

Nhung telnet đến thì không được:

```
root@cb29c9cc6961:/# telnet 10.9.0.11
Trying 10.9.0.11...
^C
root@cb29c9cc6961:/#
```

#### 4. Task 2.B: Protecting the Internal Network

Khi tìm hiểu về các interface của router thì thấy eth0 là mạng bên ngoài, eth1 là mạng bên trong

```
root@1e8a5ad520b3:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.9.0.11 netmask 255.255.255.0 broadcast 10.9.0.255
          ether 02:42:0a:09:00:0b txqueuelen 0 (Ethernet)
            RX packets 76 bytes 8826 (8.8 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.60.11 netmask 255.255.255.0 broadcast 192.168.60.255
          ether 02:42:c0:a8:3c:0b txqueuelen 0 (Ethernet)
            RX packets 51 bytes 5309 (5.3 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Tiến hành tạo các rule sau:

```
root@1e8a5ad520b3:/# iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-request -j DROP
root@1e8a5ad520b3:/# iptables -A FORWARD -i eth1 -p icmp --icmp-type echo-request -j ACCEPT
root@1e8a5ad520b3:/# iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-reply -j ACCEPT
root@1e8a5ad520b3:/# iptables -P FORWARD DROP
```

Rule 1 sẽ không có phép gói tin icmp request từ bên ngoài vào bên trong

Rule 2 sẽ cho phép gửi gói tin từ bên trong ra bên ngoài

Rule 3 sẽ cho phép gói tin từ bên ngoài đáp bên trong

Rule 4 sẽ là mặc định chặn các gói tin khác

Đầu tiên máy A không ping được máy 1:

```
root@cb29c9cc6961:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.

^C--- 192.168.60.5 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3071ms
```

Máy A có thể ping router:

```
root@cb29c9cc6961:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=1.08 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.065 ms
^C
--- 10.9.0.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.065/0.574/1.084/0.509 ms
```

Máy 1 sẽ ping được máy A:

```
root@7d8062114ce2:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
64 bytes from 10.9.0.5: icmp_seq=1 ttl=63 time=1.32 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.092 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.097 ms
^C
--- 10.9.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2012ms
rtt min/avg/max/mdev = 0.092/0.502/1.317/0.576 ms
```

Cuối cùng là máy A telnet không được đến máy 1:

```
root@cb29c9cc6961:/# telnet 192.168.60.5
Trying 192.168.60.5...
^C
```

Máy 1 cũng không telnet đến máy A:

```
root@7d8062114ce2:/# telnet 10.9.0.5
Trying 10.9.0.5...
^C
root@7d8062114ce2:/#
```

## 5. Task 2.C: Protecting Internal Servers

Trên máy router cấu hình các rule sau:

```
root@1e8a5ad520b3:/# iptables -A FORWARD -i eth0 -p tcp -d 192.168.60.5 --dport 23 -j ACCEPT
root@1e8a5ad520b3:/# iptables -A FORWARD -i eth1 -p tcp -s 192.168.60.5 --sport 23 -j ACCEPT
root@1e8a5ad520b3:/# iptables -P FORWARD DROP
```

Rule đầu sẽ cho phép gói tin từ cổng 23 đến máy 1 của máy bên ngoài

Rule thứ 2 sẽ cho phép gói tin đáp lại khi yêu cầu kết nối từ bên ngoài

Rule cuối sẽ chặn các loại gói tin khác

Máy bên ngoài có thể telnet đến máy 1:

```
root@cb29c9cc6961:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
seedUbuntu 20.04.1 LTS
seed7d8062114ce2 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

This system has been minimized by removing packages and content that are not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Nhưng không telnet đến các máy còn lại:

```
root@cb29c9cc6961:/# telnet 192.168.60.6
Trying 192.168.60.6...
^C
root@cb29c9cc6961:/# telnet 192.168.60.7
Trying 192.168.60.7...
^C
```

Máy 1 có thể telnet đến máy 2

```
root@7d8062114ce2:/# telnet 192.168.60.6
Trying 192.168.60.6...
Connected to 192.168.60.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
0090f43bebfd login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

Máy 1 không telnet đến máy A

```
seed@0090f43bebfd:~$ exit
logout
Connection closed by foreign host.
root@7d8062114ce2:/# telnet 10.9.0.5
Trying 10.9.0.5...
^C
root@7d8062114ce2:/# █
```

## D. Task 3: Connection Tracking and Stateful Firewall

### 6. Task 3.A: Experiment with the Connection Tracking

Khi ping từ máy A đến máy 1

```
root@cb29c9cc6961:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=0.087 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.122 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.127 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.105 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.094 ms
^C
--- 192.168.60.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4093ms
rtt min/avg/max/mdev = 0.087/0.107/0.127/0.015 ms
```

Khi kiểm tra conntrack trong máy router thì thấy sau khoảng 30 giây không có gói tin ping đến thì không còn entries nữa

```
root@1e8a5ad520b3:/# conntrack -L
icmp      1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=44 src=192.168.60.5 dst=10.9.0.5 type=0 code=0 i
d=44 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1e8a5ad520b3:/# conntrack -L
icmp      1 29 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=44 src=192.168.60.5 dst=10.9.0.5 type=0 code=0 i
d=44 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1e8a5ad520b3:/# conntrack -L
conntrack v1.4.5 (conntrack-tools): 0 flow entries have been shown.
```

Mở cổng lắng nghe netcat ở máy 1 nhận gói tin UDP:

```
root@7d8062114ce2:/# nc -lu 9090
hekk
j
^C
```

Ở máy A kết nối đến máy 1:

```
root@cb29c9cc6961:/# nc -u 192.168.60.5 9090
hekk
j
^C
```

Sau khi gửi tin nhắn thì ở máy router sẽ thấy:

```
root@1e8a5ad520b3:/# conntrack -L
udp      17 19 src=10.9.0.5 dst=192.168.60.5 sport=48660 dport=9090 [UNREPLIED] src=192.168.60.5 dst=10.9.0.5
sport=9090 dport=48660 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1e8a5ad520b3:/# conntrack -L
udp      17 27 src=10.9.0.5 dst=192.168.60.5 sport=48660 dport=9090 src=192.168.60.5 dst=10.9.0.5 sport=9090
dport=48660 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1e8a5ad520b3:/# conntrack -L
udp      17 18 src=10.9.0.5 dst=192.168.60.5 sport=48660 dport=9090 src=192.168.60.5 dst=10.9.0.5 sport=9090
dport=48660 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1e8a5ad520b3:/# conntrack -L
conntrack v1.4.5 (conntrack-tools): 0 flow entries have been shown.
```

Ở dòng có UNREPLIED là khi máy A gửi tin nhắn mà máy 1 chưa gửi thì conctrack cho biết là máy 1 không trả lời

Các dòng tiếp theo là khi máy 1 đã trả lời

Cuối cùng thì cũng sau một thời gian thì cũng không còn ghi lại

Ở máy 1 cũng mở cổng lắng nghe nhưng là gói tin TCP:

```
|root@7d8062114ce2:/# nc -l 9090  
|j  
|kl
```

Ở máy A kết nối cổng đó:

```
[`root@cb29c9cc6961:/# nc 192.168.60.5 9090
j
kl
^C
```

Trên máy router sẽ thấy khác với UDP sẽ có thêm [ASSURED]

Đầu tiên khi kết nối sẽ thấy có ESTABLISHED

Sau một thời gian không gửi tin nhắn, sẽ trở thành TIME\_WAIT

Cuối cùng sau một thời gian nữa thì sẽ không còn entry

```
root@1e8a5ad520b3:/# conntrack -L
tcp      6 431979 ESTABLISHED src=10.9.0.5 dst=192.168.60.5 sport=38730 dport=9090 src=192.168.60.5 dst=10.9.0.5
sport=9090 dport=38730 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1e8a5ad520b3:/# conntrack -L
tcp      6 84 TIME_WAIT src=10.9.0.5 dst=192.168.60.5 sport=38730 dport=9090 src=192.168.60.5 dst=10.9.0.5
sport=9090 dport=38730 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1e8a5ad520b3:/# conntrack -L
tcp      6 67 TIME_WAIT src=10.9.0.5 dst=192.168.60.5 sport=38730 dport=9090 src=192.168.60.5 dst=10.9.0.5
sport=9090 dport=38730 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@1e8a5ad520b3:/# conntrack -L
conntrack v1.4.5 (conntrack-tools): 0 flow entries have been shown.
```

## 7. Task 3.B: Setting Up a Stateful Firewall

Để tái tạo lại bài tập 2.C và thêm điều kiện là máy trong có thể giao tiếp đến bên ngoài thì tao các rule sau:

```
root@1e8a5ad520b3:/# iptables -A FORWARD -i eth0 -p tcp -d 192.168.60.5 --dport 23 --syn -m conntrack --ctstate NEW -j ACCEPT
root@1e8a5ad520b3:/# iptables -A FORWARD -i eth1 -p tcp --syn -m conntrack --ctstate NEW -j ACCEPT
root@1e8a5ad520b3:/# iptables -A FORWARD -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
root@1e8a5ad520b3:/# iptables -A FORWARD -p tcp -j DROP
root@1e8a5ad520b3:/# iptables -P FORWARD ACCEPT
```

Rule đầu sẽ cho phép kết nối bên ngoài đến máy 1

Rule 2 sẽ cho phép kết nối máy bên trong đến máy bên ngoài

Rule 3 cho phép những gói tin giao tiếp giữa các máy đã kết nối được qua tường lửa

Rule 4 không cho phép các loại gói tin TCP khác

Rule cuối sẽ cho phép các gói tin được qua lại nếu không phạm phải những rule trên.

Máy A kết nối đến máy 1:

```
root@cb29c9cc6961:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
7d8062114ce2 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

Nhưng không kết nối được đến các máy còn lại

```
root@cb29c9cc6961:/# telnet 192.168.60.6
Trying 192.168.60.6...
^C
root@cb29c9cc6961:/# telnet 192.168.60.7
Trying 192.168.60.7...
^C
```

Máy 1 kết nối đến máy 2:

```
root@7d8062114ce2:/# telnet 192.168.60.6
Trying 192.168.60.6...
Connected to 192.168.60.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
0090f43bebfd login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

Khác với câu 2.c thì máy 1 có thể kết nối đến máy A

```
root@7d8062114ce2:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
cb29c9cc6961 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

## E. Task 4: Limiting Network Traffic

Khi chỉ áp dụng câu lệnh đầu tiên

```
root@1e8a5ad520b3:/# iptables -A FORWARD -s 10.9.0.5 -m limit --limit 10/minute --limit-burst 5 -j ACCEPT
```

Ở bên máy A ping đến máy 1 thì thấy rằng vẫn hoạt động bình thường

```
root@cb29c9cc6961:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=0.277 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.108 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.120 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.113 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.142 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.119 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.101 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.116 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.121 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.126 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.068 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.117 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.117 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.081 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.068 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=0.085 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=0.204 ms
64 bytes from 192.168.60.5: icmp_seq=18 ttl=63 time=0.093 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.066 ms
64 bytes from 192.168.60.5: icmp_seq=20 ttl=63 time=0.113 ms
```

Điều này do câu lệnh trên cho biết 5 gói tin đầu sẽ được nhận ngay lập tức, sau đó nhận thêm 10 gói tin mỗi phút, các gói tin sau 10 gói đó sẽ theo rule tiếp theo nhưng do không có nên sẽ theo rule mặc định, rule mặc định là cho phép forward nên không có bỏ gói tin.

Vậy nếu thêm câu lệnh thứ 2 có tác dụng bỏ các gói tin không được rule 1 nhận

```
root@1e8a5ad520b3:/# iptables -A FORWARD -s 10.9.0.5 -j DROP
```

Khi ping thì thấy các gói tin

```

root@cb29c9cc6961:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=0.131 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.119 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.116 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.104 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.099 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.167 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.114 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.087 ms
64 bytes from 192.168.60.5: icmp_seq=25 ttl=63 time=0.070 ms
64 bytes from 192.168.60.5: icmp_seq=31 ttl=63 time=0.086 ms
64 bytes from 192.168.60.5: icmp_seq=37 ttl=63 time=0.073 ms
64 bytes from 192.168.60.5: icmp_seq=43 ttl=63 time=0.108 ms
^C
--- 192.168.60.5 ping statistics ---
46 packets transmitted, 12 received, 73.913% packet loss, time 46083ms
rtt min/avg/max/mdev = 0.070/0.106/0.167/0.025 ms

```

Thấy được một số gói tin bị bỏ

## F. Task 5: Load Balancing

Mở cổng lắng nghe trong 3 máy

Máy 1

```

[root@7d8062114ce2: ~]# nc -luk 8080
Connection closed by foreign host.

```

Máy 2

```

[12/16/24] seed@VM:~/.../Labsetup$ docksh 0090f43bebfd
root@0090f43bebfd:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.60.6 netmask 255.255.255.0 broadcast 192.168.60.255
        ether 02:42:c0:a8:3c:06 txqueuelen 0 (Ethernet)
        RX packets 226 bytes 17780 (17.7 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 122 bytes 10150 (10.1 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        loop txqueuelen 1000 (Local Loopback)
        RX packets 48 bytes 4970 (4.9 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 48 bytes 4970 (4.9 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@0090f43bebfd:/# nc -luk 8080

```

Máy 3

```
[12/16/24] seed@VM:~/.../Labsetup$ docksh 88404364d738
root@88404364d738:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.60.7 netmask 255.255.255.0 broadcast 192.168.60.255
        ether 02:42:c0:a8:3c:07 txqueuelen 0 (Ethernet)
        RX packets 87 bytes 8360 (8.3 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        loop txqueuelen 1000 (Local Loopback)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@88404364d738:/# nc -luk 8080
```

Sử dụng nthmode (round-robin) bằng cách cấu hình các câu lệnh sau:

```
root@1e8a5ad520b3:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode nth --every 3 --packet 0 -j DNAT --to-destination 192.168.60.5:8080
root@1e8a5ad520b3:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode nth --every 3 --packet 1 -j DNAT --to-destination 192.168.60.6:8080
root@1e8a5ad520b3:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode nth --every 3 --packet 2 -i DNAT --to-destination 192.168.60.7:8080
```

Mỗi 3 packet thì gửi packet lần lượt cho máy 1, máy 2, máy 3

Khi gửi tin nhắn có nội dung lần lượt là hello1, hello2, hello3

```
root@cb29c9cc6961:/# echo hello1 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo hello2 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo hello3 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/#
```

Máy 1

```
root@7d8062114ce2:/# nc -luk 8080
hello1
```

Máy 2

```
root@0090f43bebfd:/# nc -luk 8080
hello2
```

Máy 3

```
root@88404364d738:/# nc -luk 8080
hello3
```

Sử dụng randommode

Tạo các câu lệnh sau để cấu hình xác suất nhận gói tin

```
root@le8a5ad520b3:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random --probability 0.33 -j DNAT --to-destination 192.168.60.5:8080
root@le8a5ad520b3:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random --probability 0.5 -j DNAT --to-destination 192.168.60.6:8080
root@le8a5ad520b3:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -j DNAT --to-destination 192.168.60.7:8080
```

Câu lệnh 1 sẽ có 33% nhận được gói tin ở máy 1

Câu lệnh 2 sẽ có 50% nhận được gói tin ở máy 2

Câu lệnh 3 sẽ có 34% nhận được gói tin ở máy 3

Sau khi gửi 11 gói tin

```
root@cb29c9cc6961:/# echo pc1 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo pc2 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo pc3 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo pc4 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo pc5 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo pc6 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo pc7 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo pc8 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo pc8 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo pc9 | nc -u 10.9.0.11 8080
^C
root@cb29c9cc6961:/# echo pc10 | nc -u 10.9.0.11 8080
^C
```

Ở máy 1 nhận được 1 gói

```
root@7d8062114ce2:/# nc -luk 8080
pc3
```

Ở máy 2 nhận được 5 gói

```
root@0090f43bebfd:/# nc -luk 8080
pc2
pc5
pc7
pc8
pc10
```

Ở máy 3 nhận được 5 gói tin

```
root@88404364d738:/# nc -luk 8080
pc1
pc4
pc6
pc8
pc9
```

---

HẾT