

BÁO CÁO BÀI TẬP

Môn học: An toàn mạng máy tính

Kỳ báo cáo: Buổi 04 (Session 04)

Tên chủ đề: Firewall

GV: Nghi Hoàng Khoa

Ngày báo cáo: xx/xx/20xx

Nhóm: XX (nếu không có xoá phần này)

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT101.N11.ANTN

STT	Họ và tên	MSSV	Email
1	Võ Anh Kiệt	20520605	20520605@gm.uit.edu.vn
2	Nguyễn Bảo Phương	20520704	20520704@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Task 01	100%
2	Task 02	100%
3	Task 03	100%
4	Task 04	100%
5	Task 05	100%

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

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

Set up

Đầu tiên ta sẽ thực hiện cài đặt file yml docker để tạo các máy host, router để thực hiện bài lab

```
[12/02/22]seed@VM:~/.../Labsetup$ docker-compose build
HostA uses an image, skipping
Host1 uses an image, skipping
Host2 uses an image, skipping
Host3 uses an image, skipping
Building Router
Step 1/2 : FROM handsonsecurity/seed-ubuntu:large
--> cecb04fbf1dd
Step 2/2 : RUN apt-get update      && apt-get install -y kmod      && apt-get clean
--> Using cache
--> 46d2677e7083

Successfully built 46d2677e7083
Successfully tagged seed-router-image:latest
[12/02/22]seed@VM:~/.../Labsetup$ docker-compose up
Starting seed-router    ... done
Starting host1-192.168.60.5 ... done
Starting host3-192.168.60.7 ... done
Starting host2-192.168.60.6 ... done
Starting hostA-10.9.0.5   ... done
Attaching to host1-192.168.60.5, hostA-10.9.0.5, seed-router, host3-192.168.60.7, host2-192.168.60.6
host1-192.168.60.5 | * Starting internet superserver inetd          [ OK ]
hostA-10.9.0.5 | * Starting internet superserver inetd          [ OK ]
seed-router | * Starting internet superserver inetd          [ OK ]
host2-192.168.60.6 | * Starting internet superserver inetd          [ OK ]
host3-192.168.60.7 | * Starting internet superserver inetd          [ OK ]
```

Task 1: Implementing a simple firewall

1.A. Implementing a simple kernel module

Đầu tiên ta sẽ có được file hello.c sẵn trong chương trình. Chương trình này sẽ thực hiện in ra Hello World khi thực hiện việc initialize và in ra Bye-bye World khi thực hiện việc clean up.

```
C hello.c 5 X
files > kernel_module > C hello.c > ...
1 #include <linux/module.h>
2 #include <linux/kernel.h>
3
4 int initialization(void)
5 {
6     printk(KERN_INFO "Hello World!\n");
7     return 0;
8 }
9
10 void cleanup(void)
11 {
12     printk(KERN_INFO "Bye-bye World!.\n");
13 }
14
15 module_init(initialization);
16 module_exit(cleanup);
17
18 MODULE_LICENSE("GPL");
19
```

Tiếp tục thì ta sẽ có sẵn file Makefile bài lab cung cấp với nội dung như sau chúng ta sẽ thực hiện để build file

```
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
```

Tiếp tục thực hiện, ở terminal bên trái ta sẽ thực hiện makefile bằng lệnh make kết hợp với makefile được cung cấp sẵn. Sau đó ở ta sẽ thực hiện initialize với lệnh insmod và lsmod để kiểm tra, khi đó ta thấy được terminal bên trái in ra dòng chữ Hello World. Và khi thực hiện việc remove đi thì terminal bên trái sẽ in ra dòng Bye-bye world.

```

seed@VM:~/.../kernel_module
make -C /lib/modules/5.15.0-50-generic/build M=/home/seed/Downloads/lab42/Labse
tup/Files/kernel_module modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-50-generic'
  CC [M]  /home/seed/Downloads/lab42/Labsetup/Files/kernel_module/hello.o
  MODPOST /home/seed/Downloads/lab42/Labsetup/Files/kernel_module/Module.symver
  LD [M]  /home/seed/Downloads/lab42/Labsetup/Files/kernel_module/hello.ko
  BTF [M]  /home/seed/Downloads/lab42/Labsetup/Files/kernel_module/hello.ko
Skipping BTF generation for /home/seed/Downloads/lab42/Labsetup/Files/kernel_mo
dule/hello.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-50-generic'
[12/02/22]seed@VM:~/.../kernel_modules$ sudo insmod hello.ko
[12/02/22]seed@VM:~/.../kernel_modules$ lsmod | grep hello
hello           16384  0
[12/02/22]seed@VM:~/.../kernel_modules$ sudo rmmod hello
[12/02/22]seed@VM:~/.../kernel_modules$ 

```

1.B. Implement a Simple Firewall Using Netfilter

1.B.1. Ở nhiệm vụ này ta sẽ thực hiện implement 1 module filter và add vào trong kernel, module này sẽ giúp cho việc lọc các gói tin dựa theo các quy định mà mình cấu hình trong file.

Đầu tiên, theo hướng dẫn ta sẽ thực hiện việc kiểm tra kết nối đến ip 8.8.8.8 thì thấy được rằng kết nối hoàn toàn được xác lập

```

[12/02/22]seed@VM:~/.../Files$ 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: 3973
;; 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.      17905    IN      A      93.184.216.34

;; Query time: 24 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Fri Dec 02 11:23:15 EST 2022
;; MSG SIZE  rcvd: 60

[12/02/22]seed@VM:~/.../Files$ 

```

Makefile đã được cung cấp sẵn trong phần lab này ta sẽ thực hiện sử dụng để build file

```
M Makefile ×

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

Ở terminal bên trái ta sẽ thực hiện việc make file và insert mode và trong kernel. Sau đó ta sẽ thử kiểm tra lại việc kết nối đến ip 8.8.8.8 thì ở bên terminal bên phải ta có thể thấy được việc các gói tin đến ip 8.8.8.8 ở port 53 đều bị drop.

The screenshot shows two terminal windows side-by-side. The left window (seed@VM: ~/.../packet_filter) displays the command-line interface for building a kernel module named 'seedFilter'. It includes commands like 'make', 'cd', and 'insmod' to compile the module and load it into the kernel. The right window (seed@VM: ~-/Labsetup) shows the system's network traffic using 'sudo dmesg -n -w'. It lists numerous UDP packets originating from 'LOCAL_OUT' (IP 10.0.2.15) and destined for various external IP addresses (e.g., 117.18.232.200, 203.113.131.3, 185.125.190.49). A specific entry at the bottom indicates that UDP packets to port 53 from 8.8.8.8 are being dropped by the kernel module.

```
[12/02/22]seed@VM:~/.../packet_filter
make -C /lib/modules/5.15.0-50-generic/build M=/home/seed/Downloads/lab42/Labsetup/Files/packet_filter/modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-50-generic'
  CC [M]  /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedFilter.o
  MODPOST /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/Module.symvers
  CC [M]  /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedFilter.mod.o
  LD [M]  /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedFilter.ko
  BTF [M] /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedFilter.ko
Skipping BTF generation for /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedFilter.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-50-generic'
[12/02/22]seed@VM:~/.../packet_filter$ sudo insmod seedFilter.ko
[12/02/22]seed@VM:~/.../packet_filter$ lsmod | grep seedFilter
seedFilter           16384  0
[12/02/22]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
[12/02/22]seed@VM:~/.../packet_filter$ 
```

```
[12/02/22]seed@VM:~/.../Labsetup
seed@VM: ~/Labsetup$ sudo dmesg -n -w
[ 6924.733224] Registering filters.
[ 6958.793884] *** LOCAL_OUT
[ 6958.794169] 10.0.2.15 --> 117.18.232.200 (TCP)
[ 6971.029500] *** LOCAL_OUT
[ 6971.029504] 10.0.2.15 --> 203.113.131.3 (UDP)
[ 6971.034465] *** LOCAL_OUT
[ 6971.034469] 10.0.2.15 --> 185.125.190.49 (TCP)
[ 6971.307931] *** LOCAL_OUT
[ 6971.307935] 10.0.2.15 --> 185.125.190.49 (TCP)
[ 6971.308384] *** LOCAL_OUT
[ 6971.308386] 10.0.2.15 --> 185.125.190.49 (TCP)
[ 6971.580905] *** LOCAL_OUT
[ 6971.581152] 10.0.2.15 --> 185.125.190.49 (TCP)
[ 6971.582053] *** LOCAL_OUT
[ 6971.582055] 10.0.2.15 --> 185.125.190.49 (TCP)
[ 6982.710166] *** LOCAL_OUT
[ 6982.710196] 127.0.0.1 --> 127.0.0.53 (UDP)
[ 6982.710354] *** LOCAL_OUT
[ 6982.710355] 10.0.2.15 --> 203.113.131.3 (UDP)
[ 6982.714403] *** LOCAL_OUT
[ 6982.714406] 127.0.0.53 --> 127.0.0.1 (UDP)
[ 6982.936486] *** LOCAL_OUT
[ 6982.936490] 127.0.0.1 --> 127.0.0.53 (UDP)
[ 6982.936992] *** LOCAL_OUT
[ 6982.936994] 127.0.0.53 --> 127.0.0.1 (UDP)
[ 7002.105532] *** LOCAL_OUT
[ 7002.105535] 127.0.0.1 --> 127.0.0.1 (UDP)
[ 7002.106479] *** LOCAL_OUT
[ 7002.106480] 10.0.2.15 --> 8.8.8.8 (UDP)
[ 7002.106487] *** Dropping 8.8.8.8 (UDP), port 53
[ 7003.852026] *** LOCAL_OUT
[ 7003.852369] 10.0.2.15 --> 117.18.232.200 (TCP)
[ 7007.101795] *** LOCAL_OUT
[ 7007.101799] 10.0.2.15 --> 8.8.8.8 (UDP)
[ 7007.101813] *** Dropping 8.8.8.8 (UDP), port 53
[ 7012.101564] *** LOCAL_OUT
[ 7012.101568] 10.0.2.15 --> 8.8.8.8 (UDP)
[ 7012.101582] *** Dropping 8.8.8.8 (UDP), port 53
```

Sau khi hoàn thành xong ta sẽ thực hiện gõ mode Filter để chuẩn bị cho bài tiếp theo.

[12/02/22]seed@VM:~/.../packet_filter\$ sudo rmmod seedFilter

1.B.2. seedPrint

Đầu tiên ta sẽ thực hiện code chương trình c seedPrint với nội dung chặn gói tin UDP đi ra ngoài và drop các kết nối UDP trước khi ra bên ngoài. Code được thực hiện ở phía bên dưới;

```
C seedPrint.c 2 ×
Files > packet_filter > C seedPrint.c > blockUDP(void *, sk_buff *, const nf_hook_state *)
1  #include <linux/kernel.h>
2  #include <linux/module.h>
3  #include <linux/netfilter.h>
4  #include <linux/netfilter_ipv4.h>
5  #include <linux/ip.h>
6  #include <linux/tcp.h>
7  #include <linux/udp.h>
8  #include <linux/icmp.h>
9  #include <linux/if_ether.h>
10 #include <linux/inet.h>
11
12 static struct nf_hook_ops hook1, hook2, hook3, hook4, hook5;
13
14 unsigned int blockUDP(void *priv, struct sk_buff *skb,
15                      const struct nf_hook_state *state)
16 {
17     struct iphdr *iph;
18     struct udphdr *udph;
19
20     u16 port = 53; // DNS
21     char ip[16] = "8.8.8.8";
22     u32 ip_addr;
23
24     if (!skb)
25         return NF_ACCEPT;
26
27     iph = ip_hdr(skb);
28     // Convert the IPv4 address from dotted decimal to 32-bit binary
29     in4_pton(ip, -1, (u8 *)&ip_addr, '\0', NULL);
30
31     if (iph->protocol == IPPROTO_UDP)
32     {
33         udph = udp_hdr(skb);
34         if (iph->daddr == ip_addr && ntohs(udph->dest) == port)
35         {
36             printk(KERN_WARNING "*** Dropping %pI4 (UDP), port %d\n", &(iph->daddr), port);
37             return NF_DROP;
38         }
39     }
40     return NF_ACCEPT;
41 }
42
```

```
43     unsigned int printInfo(void *priv, struct sk_buff *skb,
44                           const struct nf_hook_state *state)
45 {
46     struct iphdr *iph;
47     char *hook;
48     char *protocol;
49
50     switch (state->hook)
51     {
52     case NF_INET_LOCAL_IN:
53         hook = "LOCAL_IN";
54         break;
55     case NF_INET_LOCAL_OUT:
56         hook = "LOCAL_OUT";
57         break;
58     case NF_INET_PRE_ROUTING:
59         hook = "PRE_ROUTING";
60         break;
61     case NF_INET_POST_ROUTING:
62         hook = "POST_ROUTING";
63         break;
64     case NF_INET_FORWARD:
65         hook = "FORWARD";
66         break;
67     default:
68         hook = "IMPOSSIBLE";
69         break;
70     }
71     printk(KERN_INFO "*** %s\n", hook); // Print out the hook info
72
73     iph = ip_hdr(skb);
74     switch (iph->protocol)
75     {
76     case IPPROTO_UDP:
77         protocol = "UDP";
78         break;
79     case IPPROTO_TCP:
80         protocol = "TCP";
81         break;
82     case IPPROTO_ICMP:
83         protocol = "ICMP";
84         break;
85     default:
86         protocol = "OTHER";
87         break;
88     }
89     // Print out the IP addresses and protocol
90     printk(KERN_INFO "%pI4 -> %pI4 (%s)\n",
91           &(iph->saddr), &(iph->daddr), protocol);
92
93     return NF_ACCEPT;
94 }
```

```

96  int registerFilter(void)
97  {
98      printk(KERN_INFO "Registering filters.\n");
99
100     // NF_INET_PRE_ROUTING
101     hook1.hook = printInfo;
102     hook1.hooknum = NF_INET_PRE_ROUTING;
103     hook1.pf = PF_INET;
104     hook1.priority = NF_IP_PRI_FIRST;
105     nf_register_net_hook(&init_net, &hook1);
106
107     // NF_INET_LOCAL_IN
108     hook2.hook = printInfo;
109     hook2.hooknum = NF_INET_LOCAL_IN;
110     hook2.pf = PF_INET;
111     hook2.priority = NF_IP_PRI_FIRST;
112     nf_register_net_hook(&init_net, &hook2);
113
114     // NF_INET_FORWARD
115     hook3.hook = printInfo;
116     hook3.hooknum = NF_INET_FORWARD;
117     hook3.pf = PF_INET;
118     hook3.priority = NF_IP_PRI_FIRST;
119     nf_register_net_hook(&init_net, &hook3);
120
121     // NF_INET_LOCAL_OUT
122     hook4.hook = printInfo;
123     hook4.hooknum = NF_INET_LOCAL_OUT;
124     hook4.pf = PF_INET;
125     hook4.priority = NF_IP_PRI_FIRST;
126     nf_register_net_hook(&init_net, &hook4);
127
128     // NF_INET_POST_ROUTING
129     hook5.hook = printInfo;
130     hook5.hooknum = NF_INET_POST_ROUTING;
131     hook5.pf = PF_INET;
132     hook5.priority = NF_IP_PRI_FIRST;
133     nf_register_net_hook(&init_net, &hook5);
134
135     return 0;
136 }

138 void removeFilter(void)
139 {
140     printk(KERN_INFO "The filters are being removed.\n");
141     nf_unregister_net_hook(&init_net, &hook1);
142     nf_unregister_net_hook(&init_net, &hook2);
143     nf_unregister_net_hook(&init_net, &hook3);
144     nf_unregister_net_hook(&init_net, &hook4);
145     nf_unregister_net_hook(&init_net, &hook5);
146 }
147
148 module_init(registerFilter);
149 module_exit(removeFilter);
150
151 MODULE_LICENSE("GPL");
152

```

Makefile: Ở file này ta sẽ thực hiện việc tái sử dụng lại file makefile mà lab cung cấp và đổi tên lại obj-m thành seedPrint.o

```
C seedPrint.c 2      M Makefile  X
Files > packet_filter > M Makefile
1  #obj-m += seedFilter.o
2  obj-m += seedPrint.o
3
4
5  all:
6      make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
7
8  clean:
9      make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
10
11 ins:
12     sudo dmesg -C
13     sudo insmod seedFilter.ko
14
15 rm:
16     sudo rmmod seedFilter
17
18
```

Bên terminal bên trái ta sẽ thực hiện makefile và insert mode vào, sau đó ta sẽ thực hiện việc kiểm tra đến kết nối với ip 8.8.8.8 thì ta thấy được bên terminal bên phải in ra các thông tin kết nối khi thực hiện việc kết nối.

The screenshot shows two terminal windows side-by-side. The left terminal window displays the execution of a Makefile named 'seedPrint' in a directory 'packet_filter'. It shows the compilation of source files into object files ('CC', 'LD'), the creation of a module ('make modules'), and the insertion of the module into the kernel ('sudo insmod seedPrint.ko'). The right terminal window shows the output of the 'tcpdump' command capturing network traffic. It lists several network packets, primarily showing TCP connections between the local host (10.0.2.15) and an external host (54.148.213.75). The traffic includes pre-routing, forward, and post-routing stages, along with local input and output stages. The traffic is mostly associated with DNS queries and responses, such as 'dig' requests to 'www.example.com' and responses from an external server at '8.8.8.8'.

```
seed@VM: ~/packet_filter
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-50-generic'
  CC [M] /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedPrint.o
  MODPOST /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/Module.symvers
  CC [M] /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedPrint.mod.o
  LD [M] /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedPrint.ko
  BTF [M] /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedPrint.ko
Skipping BTF generation for /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedPrint.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-50-generic'
[12/02/22]seed@VM:~/packet_filter$ sudo insmod seedPrint.ko
[12/02/22]seed@VM:~/packet_filter$ lsmod | grep seedPrint
seedPrint    16384  0
[12/02/22]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: 4019
;; 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. 19813  IN      A      93.184.216.34

;; Query time: 32 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Fri Dec 02 12:01:04 EST 2022
;; MSG SIZE rcvd: 60

[12/02/22]seed@VM:~/Labsetup$ sudo dmesg -C
[12/02/22]seed@VM:~/Labsetup$ sudo dmesg -k -w
[8884.144275] Registering filters.
[8941.247486] *** PRE_ROUTING
[8941.247744] 54.148.213.75 --> 10.0.2.15 (TCP)
[8941.247923] *** LOCAL_IN
[8941.248090] 54.148.213.75 --> 10.0.2.15 (TCP)
[8941.248444] *** LOCAL_OUT
[8941.248946] 10.0.2.15 --> 54.148.213.75 (TCP)
[8941.248950] *** POST_ROUTING
[8941.248975] 10.0.2.15 --> 54.148.213.75 (TCP)
[8941.249513] *** PRE_ROUTING
[8941.249515] 54.148.213.75 --> 10.0.2.15 (TCP)
[8941.249519] *** LOCAL_IN
[8941.249519] 54.148.213.75 --> 10.0.2.15 (TCP)
[8960.311298] *** LOCAL_OUT
[8960.311302] 10.0.2.15 --> 203.113.131.3 (UDP)
[8960.311314] *** POST_ROUTING
[8960.311315] 10.0.2.15 --> 203.113.131.3 (UDP)
[8961.346127] *** PRE_ROUTING
[8961.346374] 203.113.131.3 --> 10.0.2.15 (UDP)
[8961.346558] *** LOCAL_IN
[8961.346725] 203.113.131.3 --> 10.0.2.15 (UDP)
[8962.174049] *** LOCAL_OUT
[8962.174053] 127.0.0.1 --> 127.0.0.1 (UDP)
[8962.174063] *** POST_ROUTING
[8962.174063] 127.0.0.1 --> 127.0.0.1 (UDP)
[8962.174119] *** PRE_ROUTING
[8962.174121] 127.0.0.1 --> 127.0.0.1 (UDP)
[8962.174123] *** LOCAL_IN
[8962.174123] 127.0.0.1 --> 127.0.0.1 (UDP)
[8962.174815] *** LOCAL_OUT
[8962.174816] 10.0.2.15 --> 8.8.8.8 (UDP)
[8962.174823] *** POST_ROUTING
[8962.174824] 10.0.2.15 --> 8.8.8.8 (UDP)
[8962.203861] *** PRE_ROUTING
[8962.204125] 8.8.8.8 --> 10.0.2.15 (UDP)
[8962.204310] *** LOCAL_IN
[8962.204540] 8.8.8.8 --> 10.0.2.15 (UDP)
```

Phân tích: sẽ có 2 đường gói tin có thể đi

Đường 1: NIC -> preRouting -> forward -> postRouting -> NIC

Đường 2: NIC -> preRouting -> Input -> localProcess -> Output -> postRouting -> NIC

Ở trong kết quả ở terminal 2 ta có thể thấy được gói tin đang đi theo đường 2 do đã thiết lập localProcess trong file seedPrint nên các gói tin phải đi theo đường 2 và qua kiểm tra mới có thể được đi ra ngoài. Với quy trình lọc đầu ra đầu vào này các gói tin có thể được lọc và thông qua firewall để chặn các gói tin bị lọc

Sau khi thực hiện xong ra cần phải remove mode để chuẩn bị cho bài tiếp theo

```
[12/02/22] seed@VM:~/....packet_filter$ sudo rmmod seedPrint  
[12/02/22] seed@VM:~/....packet_filter$
```

1.B.3 seedBlock

Đầu tiên ta sẽ thực hiện code seedBlock với nội dung: block các máy muốn ping và telnet đến 10.9.0.1

```

1  #include <linux/kernel.h>
2  #include <linux/module.h>
3  #include <linux/netfilter.h>
4  #include <linux/netfilter_ipv4.h>
5  #include <linux/ip.h>
6  #include <linux/tcp.h>
7  #include <linux/udp.h>
8  #include <linux/icmp.h>
9  #include <linux/if_ether.h>
10 #include <linux/inet.h>
11
12 static struct nf_hook_ops hook1, hook2, hook3, hook4;
13
14 // blocking ping to vm - 10.9.0.1
15 unsigned int blockICMP(void *priv, struct sk_buff *skb,
16                         const struct nf_hook_state *state)
17 {
18     struct iphdr *iph;
19     struct icmphdr *icmph;
20
21     // u16 port = 53; // DNS
22     char ip[16] = "10.9.0.1";
23     u32 ip_addr;
24
25     if (!skb)
26         return NF_ACCEPT;
27
28     iph = ip_hdr(skb);
29     // Convert the IPv4 address from dotted decimal to 32-bit binary
30     in4_pton(ip, -1, (u8 *)&ip_addr, '\0', NULL);
31
32     if (iph->protocol == IPPROTO_ICMP)
33     {
34         icmph = icmp_hdr(skb);
35         if (iph->daddr == ip_addr && icmph->type == ICMP_ECHO)
36         {
37             printk(KERN_WARNING "*** Dropping %pI4 (ICMP) \n", &(iph->daddr));
38             return NF_DROP;
39         }
40     }
41     return NF_ACCEPT;
42 }
```

```
45 unsigned int blockUDP(void *priv, struct sk_buff *skb,
46 | | | | | const struct nf_hook_state *state)
47 {
48     struct iphdr *iph;
49     struct udphdr *udph;
50
51     u16 port = 53; // DNS
52     char ip[16] = "8.8.8.8";
53     u32 ip_addr;
54
55     if (!skb)
56         return NF_ACCEPT;
57
58     iph = ip_hdr(skb);
59     // Convert the IPv4 address from dotted decimal to 32-bit binary
60     in4_pton(ip, -1, (u8 *)&ip_addr, '\0', NULL);
61
62     if (iph->protocol == IPPROTO_UDP)
63     {
64         udph = udp_hdr(skb);
65         if (iph->daddr == ip_addr && ntohs(udph->dest) == port)
66         {
67             printk(KERN_WARNING "*** Dropping %pI4 (UDP), port %d\n", &(iph->daddr), port);
68             return NF_DROP;
69         }
70     }
71     return NF_ACCEPT;
72 }
73
74 // blocking telnet to 10.9.0.1 : 23
75 unsigned int blockTelnet(void *priv, struct sk_buff *skb,
76 | | | | | | | | | const struct nf_hook_state *state)
77 {
78     struct iphdr *iph;
79     struct tcphdr *tcph;
80
81     u16 port = 23; // DNS
82     char ip[16] = "10.9.0.1";
83     u32 ip_addr;
84
85     if (!skb)
86         return NF_ACCEPT;
87
88     iph = ip_hdr(skb);
89     // Convert the IPv4 address from dotted decimal to 32-bit binary
90     in4_pton(ip, -1, (u8 *)&ip_addr, '\0', NULL);
91
92     if (iph->protocol == IPPROTO_TCP)
93     {
94         tcph = tcp_hdr(skb);
95         if (iph->daddr == ip_addr && ntohs(tcph->dest) == port)
96         {
97             printk(KERN_WARNING "*** Dropping %pI4 (TCP), port %d\n", &(iph->daddr), port);
98             return NF_DROP;
99         }
100    }
101    return NF_ACCEPT;
102 }
```

```
104     unsigned int printInfo(void *priv, struct sk_buff *skb,
105                           const struct nf_hook_state *state)
106 {
107     struct iphdr *iph;
108     char *hook;
109     char *protocol;
110
111     switch (state->hook)
112     {
113         case NF_INET_LOCAL_IN:
114             hook = "LOCAL_IN";
115             break;
116         case NF_INET_LOCAL_OUT:
117             hook = "LOCAL_OUT";
118             break;
119         case NF_INET_PRE_ROUTING:
120             hook = "PRE_ROUTING";
121             break;
122         case NF_INET_POST_ROUTING:
123             hook = "POST_ROUTING";
124             break;
125         case NF_INET_FORWARD:
126             hook = "FORWARD";
127             break;
128         default:
129             hook = "IMPOSSIBLE";
130             break;
131     }
132     printk(KERN_INFO "**** %s\n", hook); // Print out the hook info
133
134     iph = ip_hdr(skb);
135     switch (iph->protocol)
136     {
137         case IPPROTO_UDP:
138             protocol = "UDP";
139             break;
140         case IPPROTO_TCP:
141             protocol = "TCP";
142             break;
143         case IPPROTO_ICMP:
144             protocol = "ICMP";
145             break;
146         default:
147             protocol = "OTHER";
148             break;
149     }
150     // Print out the IP addresses and protocol
151     printk(KERN_INFO "%pI4 -> %pI4 (%s)\n",
152           &(iph->saddr), &(iph->daddr), protocol);
153
154     return NF_ACCEPT;
155 }
```

```

157 int registerFilter(void)
158 {
159     printk(KERN_INFO "Registering filters.\n");
160
161     hook1.hook = printInfo;
162     hook1.hooknum = NF_INET_LOCAL_OUT;
163     hook1.pf = PF_INET;
164     hook1.priority = NF_IP_PRI_FIRST;
165     nf_register_net_hook(&init_net, &hook1);
166
167     hook2.hook = blockUDP;
168     hook2.hooknum = NF_INET_POST_ROUTING;
169     hook2.pf = PF_INET;
170     hook2.priority = NF_IP_PRI_FIRST;
171     nf_register_net_hook(&init_net, &hook2);
172
173     hook3.hook = blockICMP;
174     hook3.hooknum = NF_INET_PRE_ROUTING;
175     hook3.pf = PF_INET;
176     hook3.priority = NF_IP_PRI_FIRST;
177     nf_register_net_hook(&init_net, &hook3);
178
179     hook4.hook = blockTelnet;
180     hook4.hooknum = NF_INET_PRE_ROUTING;
181     hook4.pf = PF_INET;
182     hook4.priority = NF_IP_PRI_FIRST;
183     nf_register_net_hook(&init_net, &hook4);
184
185     return 0;
186 }
187
188 void removeFilter(void)
189 {
190     printk(KERN_INFO "The filters are being removed.\n");
191     nf_unregister_net_hook(&init_net, &hook1);
192     nf_unregister_net_hook(&init_net, &hook2);
193     nf_unregister_net_hook(&init_net, &hook3);
194     nf_unregister_net_hook(&init_net, &hook4);
195 }
196
197 module_init(registerFilter);
198 module_exit(removeFilter);
199
200 MODULE_LICENSE("GPL");
201

```

Tiếp theo ta sẽ chạy lệnh make để thực hiện makefile được kết hợp cài đặt. Sau đó ta thực hiện insert mode và trong kernel và ls để kiểm tra

```
[12/03/22] seed@VM:~/.../packet_filter$ make
make -C /lib/modules/5.15.0-50-generic/build M=/home/seed/Downloads/lab42/Labsetup/Files/packet_filter modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-50-generic'
  CC [M]  /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedBlock.o
  MODPOST /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/Module.symvers
  CC [M]  /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedBlock.mod.o
  LD [M]  /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedBlock.ko
  BTF [M] /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedBlock.ko
Skipping BTF generation for /home/seed/Downloads/lab42/Labsetup/Files/packet_filter/seedBlock.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-50-generic'
[12/03/22] seed@VM:~/.../packet_filter$ sudo insmod seedBlock.ko
[12/03/22] seed@VM:~/.../packet_filter$ lsmod | grep seedBlock
seedBlock           16384  0
```

Sau khi đã thực hiện xong ta sẽ thử ping tới ip 10.9.0.1 thì ta thấy bị mất 100% và bên terminal bên phải ta thấy đều bị drop

<pre>root@abcd2220d1e6:/# 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 --- 4 packets transmitted, 0 received, 100% packet loss, time 3059ms root@abcd2220d1e6:/#</pre>	<pre>[12/03/22] seed@VM:~/.../Labsetup\$ sudo dmesg -k -w [4662.431497] *** Dropping 10.9.0.1 (ICMP) [4663.442450] *** Dropping 10.9.0.1 (ICMP) [4664.462682] *** Dropping 10.9.0.1 (ICMP) [4665.490261] *** Dropping 10.9.0.1 (ICMP) ^C [12/03/22] seed@VM:~/.../Labsetup\$</pre>
--	--

Tương tự với telnet cũng vậy ta thấy được việc kết nối đã bị chặn và bên terminal bên phải đều bị drop

<pre>root@abcd2220d1e6:/# telnet 10.9.0.1 Trying 10.9.0.1... ^C root@abcd2220d1e6:/#</pre>	<pre>[12/03/22] seed@VM:~/.../Labsetup\$ sudo dmesg -C [12/03/22] seed@VM:~/.../Labsetup\$ sudo dmesg -k -w [4607.546666] *** Dropping 10.9.0.1 (TCP), port 23 [4608.558463] *** Dropping 10.9.0.1 (TCP), port 23 [4610.574237] *** Dropping 10.9.0.1 (TCP), port 23 [4614.606715] *** Dropping 10.9.0.1 (TCP), port 23 ^C [12/03/22] seed@VM:~/.../Labsetup\$</pre>
--	--

Sau khi thực hiện xong thì ta sẽ thực hiện remove để chuẩn bị cho bài tiếp theo

```
[12/03/22] seed@VM:~/.../Labsetup$ sudo rmmod seedBlock
[12/03/22] seed@VM:~/.../Labsetup$
```

Task2 Experimenting with Stateless Firewall Rules

2A Protecting the Router

Đầu tiên ta sẽ sử dụng router và show thử ip table với filter đã cài đặt, có thể thấy là khi ta chưa cài đặt gì thì filter chưa có gì

```
[12/04/22] seed@VM:~/.../Labsetup$ docksh dc
root@dcf38c46ecbb:/# iptables -L -n -t filter
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
Chain FORWARD (policy ACCEPT)
target     prot opt source          destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination
```

Ta sẽ thực hiện cấu hình cho câu 2A, với yêu cầu thì ta sẽ thực hiện từ chối mọi kết nối nhưng chỉ giữ lại ping

```
root@dcf38c46ecbb:/# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
root@dcf38c46ecbb:/# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
root@dcf38c46ecbb:/# iptables -P OUTPUT DROP
root@dcf38c46ecbb:/# iptables -P INPUT DROP
root@dcf38c46ecbb:/# iptables -L -n -t filter
Chain INPUT (policy DROP)
target     prot opt source          destination
ACCEPT    icmp  --  0.0.0.0/0      0.0.0.0/0          icmp type 8
Chain FORWARD (policy ACCEPT)
target     prot opt source          destination
Chain OUTPUT (policy DROP)
target     prot opt source          destination
ACCEPT    icmp  --  0.0.0.0/0      0.0.0.0/0          icmp type 0
root@dcf38c46ecbb:/# █
```

Ta sẽ vào hostA để ping thử thì vẫn được, nhưng khi ta thực hiện telnet đến thì hoàn toàn bị chặn

```
root@abcd2220d1e6:/# ping seed-router
PING seed-router (10.9.0.11) 56(84) bytes of data.
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=1 ttl=64 time=0.439 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=2 ttl=64 time=0.064 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=3 ttl=64 time=0.069 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=5 ttl=64 time=0.062 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=6 ttl=64 time=0.059 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=7 ttl=64 time=0.064 ms
^C
--- seed-router ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6104ms
rtt min/avg/max/mdev = 0.059/0.117/0.439/0.131 ms
root@abcd2220d1e6:/# telnet seed-router
Trying 10.9.0.11...
telnet: Unable to connect to remote host: Connection timed out
root@abcd2220d1e6:/# █
```

Sau khi thực hiện xong thì ta sẽ clear các trường thông tin để chuẩn bị cho bài sau

```
root@dcf38c46ecbb:/# iptables -F
root@dcf38c46ecbb:/# iptables -P OUTPUT ACCEPT
root@dcf38c46ecbb:/# iptables -P INPUT ACCEPT
root@dcf38c46ecbb:/# █
```

2B Protecting the Internal Network

Ở bài này ta sẽ thực hiện cấu hình cho host ngoài không ping được vào bên trong (dòng 1), host bên ngoài ping được router (dòng 2), host bên trong ping được host bên ngoài (dòng 3), những trường hợp khác ngoài ping đều bị drop

```
root@dcf38c46ecbb:/# iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-request -j DROP
root@dcf38c46ecbb:/# iptables -A FORWARD -i eth1 -p icmp --icmp-type echo-request -j ACCEPT
root@dcf38c46ecbb:/# iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-reply -j ACCEPT
root@dcf38c46ecbb:/# iptables -P FORWARD DROP
root@dcf38c46ecbb:/# iptables -L -n -t filter
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
Chain FORWARD (policy DROP)
target     prot opt source          destination
DROP      icmp --  0.0.0.0/0          0.0.0.0/0          icmp type 8
ACCEPT    icmp --  0.0.0.0/0          0.0.0.0/0          icmp type 8
ACCEPT    icmp --  0.0.0.0/0          0.0.0.0/0          icmp type 0
Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination
root@dcf38c46ecbb:/# █
```

Thử ping từ hostA tới host1 và hostA tới router, kết quả ping được

```
root@abcd2220d1e6:/# ping -c 4 192.168.60.5
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 3065ms

root@abcd2220d1e6:/# ping -c 4 seed-router
PING seed-router (10.9.0.11) 56(84) bytes of data.
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=1 ttl=64 time=0.169 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=2 ttl=64 time=0.088 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=3 ttl=64 time=0.066 ms
64 bytes from seed-router.net-10.9.0.0 (10.9.0.11): icmp_seq=4 ttl=64 time=0.066 ms

--- seed-router ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3056ms
rtt min/avg/max/mdev = 0.066/0.097/0.169/0.042 ms
root@abcd2220d1e6:/# █
```

Ping từ host1 tới hostA, kết quả ping được

```
root@492231d1191e:/# 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=0.454 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.079 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.079 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=0.078 ms
^C
--- 10.9.0.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3070ms
rtt min/avg/max/mdev = 0.078/0.172/0.454/0.162 ms
root@492231d1191e:/#
```

Telnet

Telnet từ hostA tới host1 và host1 tới hostA để bị từ chối

```
root@492231d1191e:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
root@492231d1191e:/#
```

```
root@abcd2220d1e6:/# telnet 192.168.60.5
Trying 192.168.60.5...
telnet: Unable to connect to remote host: Connection timed out
root@abcd2220d1e6:/#
```

2C Protect Internal Servers

Ở task này ta sẽ thực hiện cấu hình sao tất cả các host123 đều chạy telnet, nhưng chỉ có host 1 mới truy cập được, các host nội bộ truy cập lẫn nhau, bên ngoài không thể truy cập vào và bên trong không thể truy cập ra.

```
root@dcf38c46ecbb:/# iptables -A FORWARD -i eth0 -d 192.168.60.5 -p tcp --dport 23 -j ACCEPT
root@dcf38c46ecbb:/# iptables -A FORWARD -i eth1 -s 192.168.60.5 -p tcp --sport 23 -j ACCEPT
root@dcf38c46ecbb:/# iptables -P FORWARD DROP
root@dcf38c46ecbb:/# iptables -L -n -t filter
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
          prot opt source          destination

Chain FORWARD (policy DROP)
target     prot opt source          destination
ACCEPT    tcp   --  0.0.0.0/0      192.168.60.5      tcp dpt:23
ACCEPT    tcp   --  192.168.60.5   0.0.0.0/0        tcp spt:23
Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination
root@dcf38c46ecbb:/#
```

Thực hiện telnet hostA tới host1

```
root@abcd2220d1e6:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
492231d1191e login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.15.0-50-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.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@492231d1191e:~$ █
```

Thực hiện telnet hostA tới host2

```
root@abcd2220d1e6:/# telnet 192.168.60.6
Trying 192.168.60.6...
telnet: Unable to connect to remote host: Connection timed out
root@abcd2220d1e6:/# █
```

Thực hiện telnet từ host2 tới hostA

```
root@656c3b54525a:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
root@656c3b54525a:/# █
```

Thực hiện telnet từ host2 tới host1

```

root@656c3b54525a:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
492231d1191e login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.15.0-50-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.
Last login: Sun Dec  4 16:10:54 UTC 2022 from www.SeedLabSQLInjection.com on pts
/2
seed@492231d1191e:~$
```

Sau khi thực hiện ta thấy thỏa các yêu cầu đề bài

Task3 Connection Tracking and Stateful Firewall

3A Experiment with the connection tracking

Để setup stateful firewall ta sẽ xem các kết nối bằng conntrack của kernel bằng lệnh bên dưới

```

root@dcf38c46ecbb:/# conntrack -L
conntrack v1.4.5 (conntrack-tools): 0 flow entries have been shown.
```

Sau đó ta sẽ thực hiện ping từ hostA tới 192.168.60.5

```

root@abcd2220d1e6:/# 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=1.23 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.072 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.071 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.070 ms
^C
--- 192.168.60.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3029ms
rtt min/avg/max/mdev = 0.070/0.361/1.234/0.503 ms
root@abcd2220d1e6:/#
```

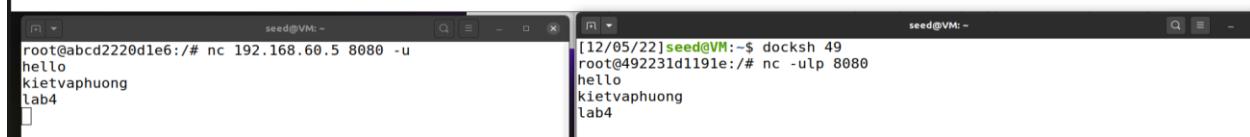
Sau khi ping xong ta sẽ thực hiện lệnh xem các kết nối thì ta thấy được 4 gói tin icmp, quan sát thử thêm 1 vài lần thì ta có thể thấy được trạng thái kết nối của gói tin icmp đang ở khoảng 30s

```
root@dcf38c46ecbb:/# conntrack -L
icmp      1 4 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=32 src=192.1
68.60.5 dst=10.9.0.5 type=0 code=0 id=32 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@dcf38c46ecbb:/# █
```

Kiểm tra kết nối với udp

Thì ta có thể thấy được rằng là ở hostA và host 1 bên dưới thực hiện netcat để giao tiếp với các lệnh tương ứng ở hai terminal phía dưới và thực hiện giao tiếp. Sau đó ta sẽ kiểm tra kết nối thì thấy được là thời gian giữ kết nối này cũng dự kiến khoảng 30s

```
root@dcf38c46ecbb:/# conntrack -L
conntrack v1.4.5 (conntrack-tools): 0 flow entries have been shown.
root@dcf38c46ecbb:/# conntrack -L
icmp      1 4 src=10.9.0.5 dst=192.168.60.5 type=8 code=0 id=32 src=192.168.60.5 dst=10.9.0.5 type=0 code=0 id=32 mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@dcf38c46ecbb:/# conntrack -L
udp      17 26 src=10.9.0.5 dst=192.168.60.5 sport=38686 dport=8080 [UNREPLIED] src=192.168.60.5 dst=10.9.0.5 sport=8080 dport=38686 mark=0
use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@dcf38c46ecbb:/# conntrack -L
udp      17 23 src=10.9.0.5 dst=192.168.60.5 sport=38686 dport=8080 [UNREPLIED] src=192.168.60.5 dst=10.9.0.5 sport=8080 dport=38686 mark=0
use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@dcf38c46ecbb:/# conntrack -L
udp      17 16 src=10.9.0.5 dst=192.168.60.5 sport=38686 dport=8080 [UNREPLIED] src=192.168.60.5 dst=10.9.0.5 sport=8080 dport=38686 mark=0
use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@dcf38c46ecbb:/# conntrack -L
udp      17 13 src=10.9.0.5 dst=192.168.60.5 sport=38686 dport=8080 [UNREPLIED] src=192.168.60.5 dst=10.9.0.5 sport=8080 dport=38686 mark=0
use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@dcf38c46ecbb:/# █
```



Kiểm tra kết nối với tcp

Ta sẽ tiếp tục thực hiện netcat như 2 lệnh dưới và ta có thể thấy thời gian kết nối giữa 2 này rất lớn có thể lên đến 432000s dự kiến lên đến 5 ngày.

```

seed@VM: ~
root@dcf38c46ecbb:/# conntrack -L
tcp      6 431995 ESTABLISHED src=10.9.0.5 dst=192.168.60.5 sport=49744 dport=8080 src=192.168.60.5 dst=10.9.0.5 sport=8080 dport=49744 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@dcf38c46ecbb:/# conntrack -L
tcp      6 431995 ESTABLISHED src=10.9.0.5 dst=192.168.60.5 sport=49744 dport=8080 src=192.168.60.5 dst=10.9.0.5 sport=8080 dport=49744 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@dcf38c46ecbb:/# conntrack -L
tcp      6 431987 ESTABLISHED src=10.9.0.5 dst=192.168.60.5 sport=49744 dport=8080 src=192.168.60.5 dst=10.9.0.5 sport=8080 dport=49744 [ASSURED] mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@dcf38c46ecbb:/# 

seed@VM: ~
root@abcd2220d1e6:/# nc 192.168.60.5 8080
hello
kietvaphuong
lab
[]

seed@VM: ~
root@492231d1191e:/# nc -l 8080
hello
kietvaphuong
lab
[]

```

Nhưng nếu thực hiện việc ngắt kết nối đi thì thời gian này giảm còn 60s so với ban đầu

```

root@dcf38c46ecbb:/# conntrack -L
tcp      6 56 CLOSE_WAIT src=10.9.0.5 dst=192.168.60.5 sport=49744 dport=8080 src=192.168.60.5 dst=10.9.0.5 sport=8080 dport=49744 [ASSURED]
mark=0 use=1
conntrack v1.4.5 (conntrack-tools): 1 flow entries have been shown.
root@dcf38c46ecbb:/# []

```

3B Setting Up a Stateful firewall

Ở phần này ta sẽ tiếp tục thực hiện cấu hình sao cho:

Cho phép trao đổi qua tcp

Chặn các kết nối mới được thực hiện bằng việc chặn gói syn

Drop các trường hợp còn lại

```

root@dcf38c46ecbb:/# iptables -A FORWARD -p tcp -i eth0 -d 192.168.60.5 --dport 23 --syn -m conntrack --ctstate NEW -j ACCEPT
root@dcf38c46ecbb:/# iptables -A FORWARD -i eth1 -p tcp --syn -m conntrack --ctstate NEW -j ACCEPT
root@dcf38c46ecbb:/# iptables -A FORWARD -p tcp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
root@dcf38c46ecbb:/# iptables -A FORWARD -p tcp -j DROP
root@dcf38c46ecbb:/# iptables -P FORWARD ACCEPT
root@dcf38c46ecbb:/# []

```

Ta thực hiện kiểm tra thì thấy việc gửi gói tin đã bị drop và không thể gửi bình thường

```

root@abcd2220d1e6:/# nc 192.168.60.5 8080
hello
kietvaphuong
[...]

```

```

root@492231d1191e:/# nc -l 8080
day la thong tin may 1
khong the ket noi
[...]

```

Ta sẽ thực hiện cấu hình theo cấu hình theo 2C trước đó

```
root@dcf38c46ecbb:/# iptables -A FORWARD -p tcp -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
root@dcf38c46ecbb:/# iptables -A FORWARD -p tcp -j DROP
root@dcf38c46ecbb:/# iptables -P FORWARD ACCEPT
root@dcf38c46ecbb:/# █
```

Thực hiện hostA telnet host1

```
root@abcd2220d1e6:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
492231d1191e login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.15.0-50-generic x8
6_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.
Last login: Sun Dec 4 16:33:50 UTC 2022 from host2-192.168.60.6.net-192.168.60.0 on pts/2
seed@492231d1191e:~\$ telnet 192.168.60.6
Trying 192.168.60.6...
Connected to 192.168.60.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
656c3b54525a login: Connection closed by foreign host.

hostA telnet máy host2

```
root@abcd2220d1e6:/# telnet 192.168.60.6
Trying 192.168.60.6...
telnet: Unable to connect to remote host: Connection timed out
root@abcd2220d1e6:/# █
```

Thực hiện telnet từ host2 tới host1

```
[12/05/22]seed@VM:~$ docksh 65
root@656c3b54525a:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
492231d1191e login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.15.0-50-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.
Last login: Mon Dec  5 16:14:00 UTC 2022 from www.SeedLabSQLInjection.com on pts
/2
seed@492231d1191e:~$
```

Thực hiện telnet từ host1 tới hostA

```
root@492231d1191e:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
abcd2220d1e6 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.15.0-50-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.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@abcd2220d1e6:~\$

stateless firewall:

Với mục đích là để xem lưu lượng và hạn chế hay chặn các gói dựa trên địa chỉ
nguồn và đích hoặc các giá trị tĩnh khác với tốc độ nhanh hơn và tốt hơn so với
việc việc tải lưu lượng do nặng hơn

Stateful yêu cầu các thông tin truy cập (như IP)

Stateless không yêu cầu các thông tin truy cập (như IP)

Task4 Limiting Network traffic

Ở bài này ta sẽ thực hiện cấu hình giới hạn gói tin với yêu cầu không quá 10 gói 1 phút và không quá 5 gói mỗi lần gửi. Ta sẽ cấu hình như hình bên dưới

```
root@dcf38c46ecbb:/# iptables -A FORWARD -s 10.9.0.5 -m limit --limit 10/minute --limit-burst 5 -j ACCEPT
root@dcf38c46ecbb:/# iptables -A FORWARD -s 10.9.0.5 -j DROP
root@dcf38c46ecbb:/# pttables -t filter -L -n
bash: pttables: command not found
root@dcf38c46ecbb:/# iptables -t filter -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
Chain FORWARD (policy ACCEPT)
target     prot opt source          destination
ACCEPT    all  --  10.9.0.5        0.0.0.0/0           limit: avg 10/min burst 5
DROP      all  --  10.9.0.5        0.0.0.0/0
Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination
root@dcf38c46ecbb:/#
```

Sau khi cấu hình xong ta sẽ thực hiện kiểm tra

Ping từ hostA tới host1

```
root@abcd2220d1e6:/# 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=0.244 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.077 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.073 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.072 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.075 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.097 ms
^C
--- 192.168.60.5 ping statistics ---
12 packets transmitted, 6 received, 50% packet loss, time 112
75ms
rtt min/avg/max/mdev = 0.072/0.106/0.244/0.062 ms
root@abcd2220d1e6:/#
```

Ta có thấy được rằng là ping này bị mất đi 6 gói tin do bị giới hạn lưu lượng

Ping từ host2 tới host1

```
root@656c3b54525a:/# 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=64 time=0.308 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.065 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=64 time=0.061 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=64 time=0.061 ms
^C
--- 192.168.60.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3068ms
rtt min/avg/max/mdev = 0.061/0.123/0.308/0.106 ms
root@656c3b54525a:/#
```

Thì ta thấy được không bị mất đi gói nào do không bị yêu cầu kiểm soát lưu lượng truy cập

Cấu hình và kiểm thử lại thông tin kiểm soát truy cập

```
root@dcf38c46ecbb:/# iptables -A FORWARD -s 10.9.0.5 -m limit --limit 10/minute --limit-burst 5 -j ACCEPT
root@dcf38c46ecbb:/# iptables -t filter -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
Chain FORWARD (policy ACCEPT)
target     prot opt source          destination
ACCEPT    all   --  10.9.0.5        0.0.0.0/0           limit: avg 10/min burst 5
Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination
root@dcf38c46ecbb:/# █
```

Ping lại từ hostA tới host1

```
root@abcd2220d1e6:/# 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=0.296 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.077 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.074 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.074 ms
^C
--- 192.168.60.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3204ms
rtt min/avg/max/mdev = 0.074/0.130/0.296/0.095 ms
root@abcd2220d1e6:/# █
```

Có thể thấy không một gói nào bị chặn nếu không thực hiện lệnh thứ 2 nhưng vẫn sẽ bị xử lý nếu vi phạm việc kiểm soát lưu lượng ở câu lệnh 1

Lợi ích có thể chặn được các cuộc tấn công Dos thông qua việc kiểm soát lưu lượng

Tác hại sẽ bị mất gói tin và các thông tin khi trao đổi có thể bị chậm trễ

Task 5 Load balancing

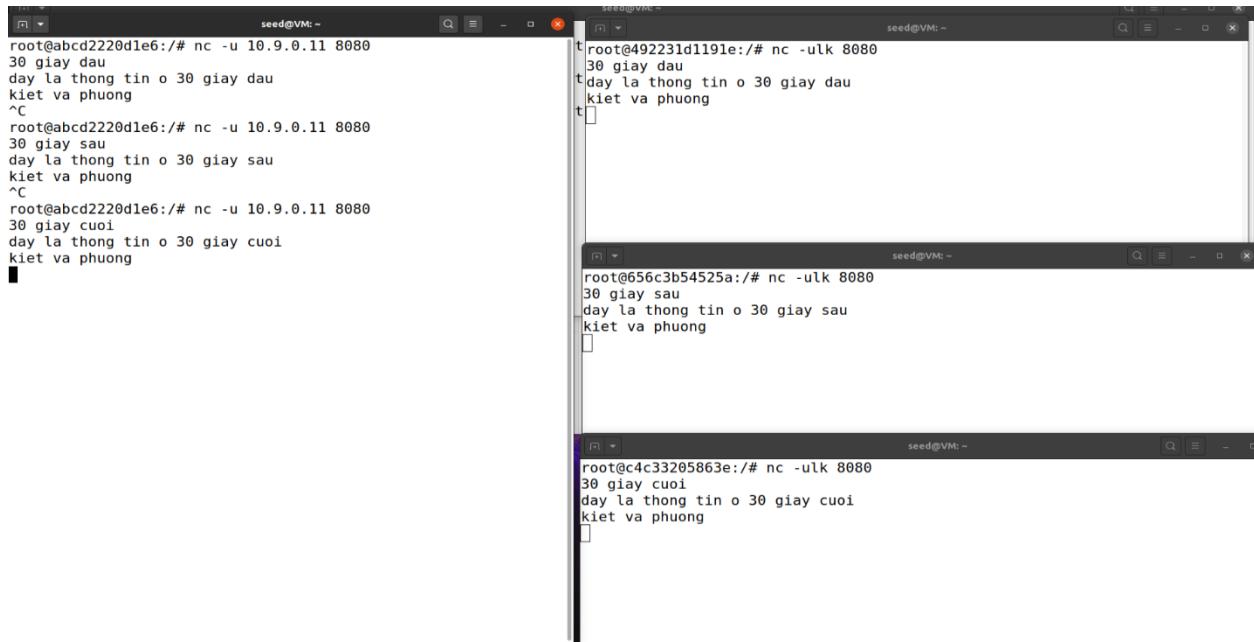
Việc thực hiện cấu hình sẽ giúp điều phối lưu lượng truy cập vào, giúp điều phối tốt lưu lượng

5A Using the nth mode

Đầu tiên ta sẽ thực hiện cấu hình để truy cập theo cơ chế nth mode

```
root@dcf38c46ecbb:/# 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@dcf38c46ecbb:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode nth --every 2 --packet 0 -j DNAT --to-destination 192.168.60.6:8080
root@dcf38c46ecbb:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode nth --every 1 --packet 0 -j DNAT --to-destination 192.168.60.7:8080
root@dcf38c46ecbb:/# iptables -t filter -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@dcf38c46ecbb:/#
```

Sau đó ta sẽ thực hiện lệnh netcat để thực hiện kết nối và kiểm tra thử, thì ta thấy được là cơ chế này sẽ đổi máy truy cập vào cứ mỗi 30 theo như hình bên dưới đã thực hiện



5B using random mode

Ở lần này ta sẽ thực hiện phân bổ theo xác suất đổi với mỗi host khi truy cập tùy vào tỉ lệ phần trăm ta cấu hình bên dưới với tỉ lệ 0.3333, 0.5 và 1

```
[12/05/22]seed@VM:~$ docker restart dc
dc
[12/05/22]seed@VM:~$ docksh
root@dcf38c46ecbb:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random --probability 0.3333 -j DNAT --to-destination 192.168.60.5:8080
root@dcf38c46ecbb:/# 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@dcf38c46ecbb:/# iptables -t nat -A PREROUTING -p udp --dport 8080 -m statistic --mode random --probability 1 -j DNAT --to-destination 192.168.60.7:8080
root@dcf38c46ecbb:/# iptables -t filter -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Ta sẽ thực hiện thử nghiệm như bên dưới thì sau mỗi 30 giây thì sẽ thay đổi host dựa trên xác suất được cấu hình và việc thay đổi này dựa trên thông tin mà chúng ta cung cấp trong phần cấu hình router bên trên

The image shows four terminal windows from a Linux environment. Each window displays a command-line session where a user is sending text to a port (8080) using the nc (netcat) or ulk (UserLand) tools. The text sent includes Vietnamese text ('day la 30 giay dau', 'kiet va phuong', 'ffff', '^C') and some random characters ('ajndfjasdjf', 'asjdfnjasndfj'). The hosts involved are abcd2220d1e6, 492231d1191e, 656c3b54525a, and c4c33205863e.

```
seed@VM: ~
root@abcd2220d1e6:/# nc -u 10.9.0.11 8080
day la 30 giay dau
kiet va phuong
ffff
^C
root@abcd2220d1e6:/# nc -u 10.9.0.11 8080
day la 30 giay sau
kietvaphuong
ffff
^C
root@abcd2220d1e6:/# nc -u 10.9.0.11 8080
day la 30 giay cuoi
ajndfjasdjf
asjdfnjasndfj

seed@VM: ~
root@492231d1191e:/# nc -ulk 8080
day la 30 giay sau
kietvaphuong
ffff
day la 30 giay cuoi
ajndfjasdjf
asjdfnjasndfj

seed@VM: ~
root@656c3b54525a:/# nc -ulk 8080
day la 30 giay sau
kietvaphuong
ffff
day la 30 giay cuoi
ajndfjasdjf
asjdfnjasndfj

seed@VM: ~
root@c4c33205863e:/# nc -ulk 8080
day la 30 giay dau
kiet va phuong
ffff
```

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File **.PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng **Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)** – cỡ **chữ 13**. **Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.**
- Đặt tên theo định dạng: [Mã lớp]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành/Tên Cá nhân đã đăng ký với GV).

Ví dụ: [NT101.K11.ANTT]-Session1_Group3.

- Nếu báo cáo có nhiều file, nén tất cả file vào file **.ZIP** với cùng tên file báo cáo.
- **Không đặt tên đúng định dạng – yêu cầu, sẽ KHÔNG chấm điểm.**
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện. Khuyến khích:

- Chuẩn bị tốt.
- Có nội dung mở rộng, ứng dụng trong kịch bản/câu hỏi phức tạp hơn, có đóng góp xây dựng.

Bài sao chép, trẽ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT