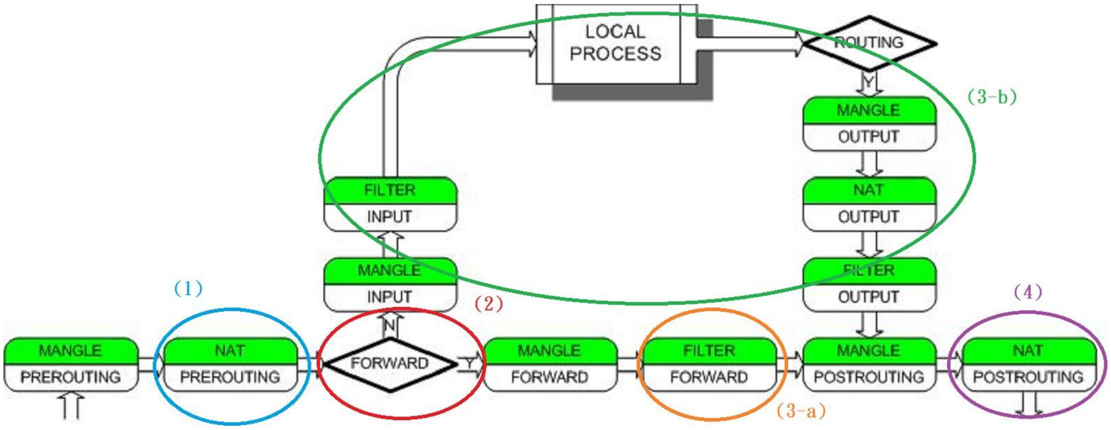
LabII Security/ Wireless LANs

Report:

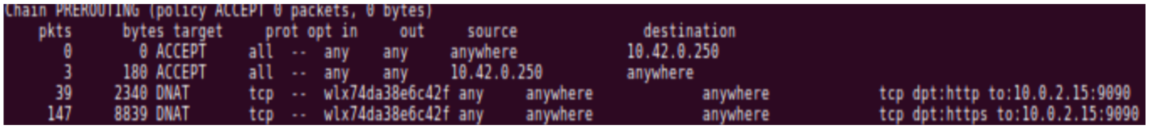
1. Team number: 13
   Member list: B07902066, B07902062, B07902068, B07902080
   Environment: OS: Ubuntu 18.04
   VM platform: VMWare
   Language: node.js(built upon express framework)+CSS+HTML

2. Briefly describe how packets go through the iptables.



(1) PREROUTING chain of NAT table

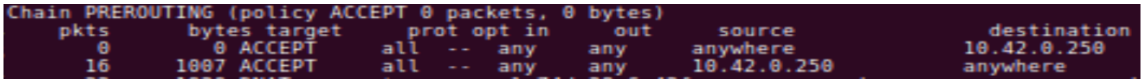
When the packets first enter the iptables, they will pass through the PREROUTING chain of NAT table to check if it has to be redirected to a certain destination address. In this lab, tcp traffic from the WLAN using ports of HTTP and HTTPS are redirected to port 9090 of the local machine's IP, which is the address of the login page. If there are any additional specific rules above, like shown in the picture, the packets that match the rule will follow the rule first. Others will follow the default rule, which can be defined using `iptables -P`, either DROP or ACCEPT. Dropping in the PREROUTING chain results in immediate loss of network connection.



```
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  any    any     anywhere             10.42.0.250
    3   180 ACCEPT     all  --  any    any     10.42.0.250          anywhere
   39  2340 DNAT       tcp  --  wlx74da38e6c42f any     anywhere             anywhere             tcp dpt:http to:10.0.2.15:9090
  147  8839 DNAT       tcp  --  wlx74da38e6c42f any     anywhere             anywhere             tcp dpt:https to:10.0.2.15:9090
```

(2) Routing decision

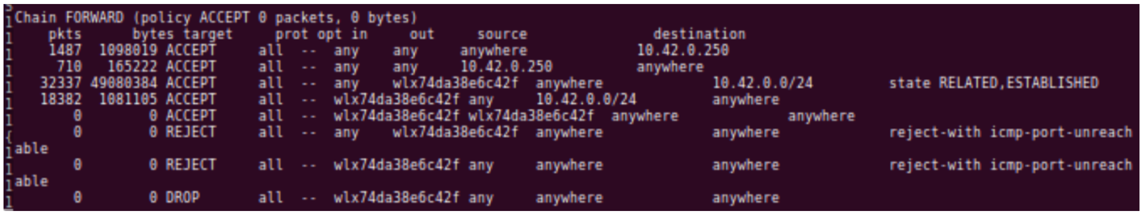
If the packet is accepted in the prerouting phase, like the packets from 10.42.0.250 shown in the picture below, the packet will then move on to the routing decision, which determines the nest chain to apply its rules, either INPUT(toward local machine) or FORWARD(packets forward by local machine)



```
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  any    any     anywhere             10.42.0.250
   16  1007 ACCEPT     all  --  any    any     10.42.0.250          anywhere
```

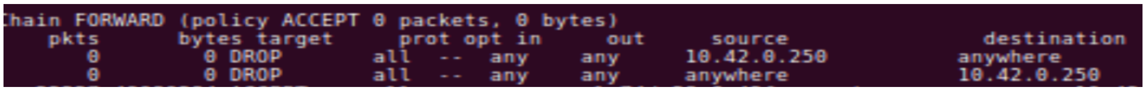
(3-a) Forward chain of filter table

If the packet goes through the FORWARD chain of the filter table, the rules of the FORWARD chain will then be applied to the packets. Since we included the rule "*`iptables -A FORWARD -i ${wlan0} -j DROP`*" before, all packets forwarded are dropped. However, since the chain checks the rules from top to bottom, if there's an ACCEPT rule for the specific IP, 10.42.0.250 in this case, above the DROP rule, the packets from or to the corresponding IP will then be accepted and forwarded, like the case shown below.



```
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
    1487  1098019 ACCEPT     all  --  any    any     anywhere             10.42.0.250
     710   165222 ACCEPT     all  --  any    any     10.42.0.250          anywhere
   32337 49080384 ACCEPT     all  --  any    wlx74da38e6c42f  anywhere             10.42.0.0/24         state RELATED,ESTABLISHED
   18382  1081105 ACCEPT     all  --  wlx74da38e6c42f any     10.42.0.0/24         anywhere
       0        0 ACCEPT     all  --  wlx74da38e6c42f wlx74da38e6c42f  anywhere             anywhere
       0        0 REJECT     all  --  any    wlx74da38e6c42f  anywhere             anywhere             reject-with icmp-port-unreach
able
       0        0 REJECT     all  --  wlx74da38e6c42f any     anywhere             anywhere             reject-with icmp-port-unreach
able
       0        0 DROP       all  --  wlx74da38e6c42f any     anywhere             anywhere
```

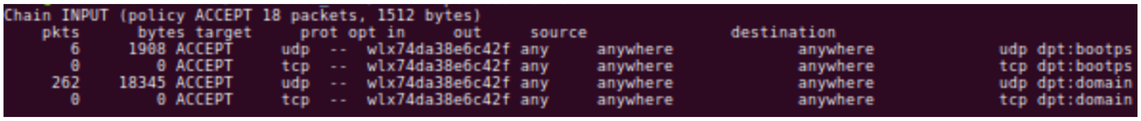
However, if the policy is DROP for the specific IP, then all traffic from or to the IP will be dropped, as shown, resulting in packets not able to be forwarded to or from the external network.



```
hain FORWARD (policy ACCEPT 0 packets, 0 bytes)
   pkts      bytes target     prot opt in     out     source               destination
      0        0 DROP       all  --  any    any     10.42.0.250          anywhere
      0        0 DROP       all  --  any    any     anywhere             10.42.0.250
```

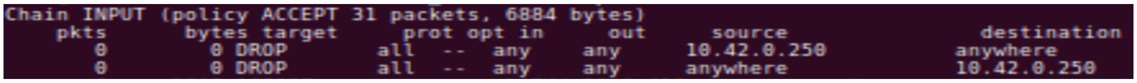
(3-b) INPUT chain of filter table, second routing decision and OUTPUT chain

If the packet goes through the INPUT chain of the filter table, the rules of the INPUT chain will then be applied to the packets. By default, we didn't set any rules for INPUT, so all traffic can pass through. After the packet is accepted and some processes are done by the local machine, the local machine may like to send some feedback back to the source IP. The feedback packet will then again go through a routing decision, then pass through the OUTPUT chain to decide whether it can be sent or dropped.



```
Chain INPUT (policy ACCEPT 18 packets, 1512 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       6     1908 ACCEPT     udp  --  wlx74da38e6c42f any     anywhere             anywhere             udp dpt:bootps
       0        0 ACCEPT     tcp  --  wlx74da38e6c42f any     anywhere             anywhere             tcp dpt:bootps
     262    18345 ACCEPT     udp  --  wlx74da38e6c42f any     anywhere             anywhere             udp dpt:domain
       0        0 ACCEPT     tcp  --  wlx74da38e6c42f any     anywhere             anywhere             tcp dpt:domain
```

However, if we set the policy to DROP for some IP, then the IP may not even access the login page(the packet is dropped by the INPUT chain), which is set on the localhost, resulting in the IP not able to access the internet.



```
Chain INPUT (policy ACCEPT 31 packets, 6884 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       all  --  any    any     10.42.0.250          anywhere
       0        0 DROP       all  --  any    any     anywhere             10.42.0.250
```

(4) POSTROUTING chain of NAT table

Packets that passed through the FORWARD chain or the OUTPUT chain will finally arrive at the POSTROUTING chain, which is used to transform the source address of the packet. After following the policies of POSTROUTING, the packet is ready to be forward to the next host.

3. Describe how your program (server & web page) interact with the iptables.

When a device successfully logs in, four new rules are added into the iptables according to the device's IP address, using function *spawnsync()*, which creates child processes to perform the iptables commands, as well as guarantee the order of commands. Two of these rules are for the FORWARD table:

```
iptables -I FORWARD -s(-d) IP -j ACCEPT
```

And the other two for nat table:

```
iptables -t nat -I PREROUTING 1 -s(-d) IP -j ACCEPT
```

The command allows the device to bypass the login page and directly access the internet, since the localhost will help forward its packets. The admin web page then fetches the information of the device using the command:

```
iptables -L -v -x,
```

which gives us the information of transferred packets and total bytes of corresponding IP addresses.

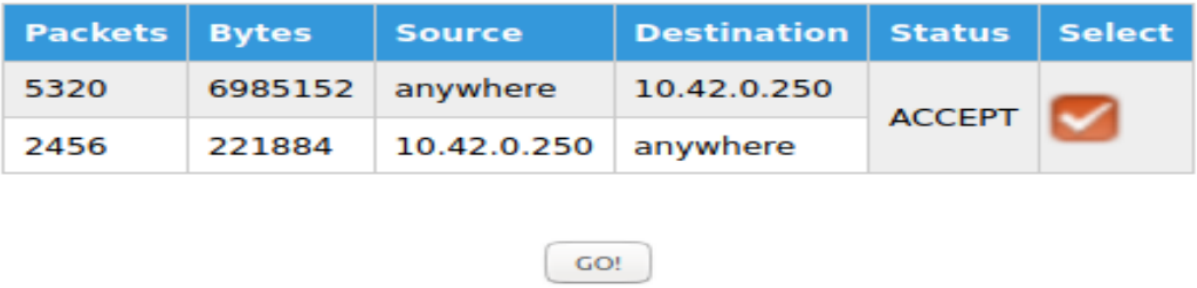
When the admin would like to block a device, he can simply check the toggle box on the right, and press the "GO!" button. The web page deletes the four rules above using

```
iptables -t nat -D PREROUTING -s(-d) IP -j ACCEPT
```

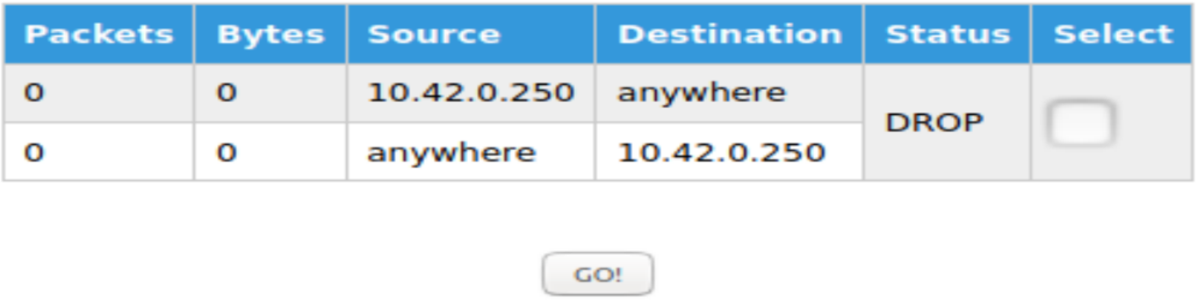
and

```
iptables -D FORWARD -d IP -j ACCEPT,
```

then add two DROP rules in both INPUT and FORWARD tables to prevent the blocked IP from accessing the network. To unblock the device, do the same process again to the DROP rule, then the web page will delete the rules that it added in the previous actions.



Packets	Bytes	Source	Destination	Status	Select
5320	6985152	anywhere	10.42.0.250	ACCEPT	☑
2456	221884	10.42.0.250	anywhere		



Packets	Bytes	Source	Destination	Status	Select
0	0	10.42.0.250	anywhere	DROP	☐
0	0	anywhere	10.42.0.250		

Situational questions:

1. Your website is working on port 8080 and can only be accessed by 140.112.0.0/16. How to modify your iptables to block unavailable users.

   To block other users, there are two methods:

   (1) Drop the traffics in PREROUTING chain:

   ```
   iptables -t nat -A PREROUTING -j DROP
   iptables -t nat -I PREROUTING 1 -s 140.112.0.0/16 -p tcp
   --dport 80(443) -j DNAT --to-destination `${local_IP}:8080`
   ```

   (2) Drop the traffics in INPUT chain:

   ```
   iptables -A INPUT --dport 8080 -j DROP
   iptables -I INPUT -s(-d) 140.112.0.0/16 -j ACCEPT
   ```

2. Behind the machine, there is a ssh server which locates at 192.168.10.2 in the eth1. People who want to connect to ssh server from eth0 need to connect the machine at port 2222 and the machine will redirect the flow to the ssh server at port 22. How to configure the iptable?(two commands for PREROUTING and POSTROUTING)

   ```
   iptables -t nat -I PREROUTING -i eth0 -p tcp --dport 2222
   -j DNAT --to-destination 192.168.10.2:22
   iptables -t nat -I POSTROUTING -s 192.168.10.2 -o eth0 -p
   tcp -j MASQUERADE
   ```