Lab 8

(Due your class day of July 21/26)

In this lab, you will be able to practice the firewall commands using iptables and sense its effect. The iptables rule is specified using the following format

iptables -t TABLE_NAME -A Chain_name several_conditions -j action

Its means that in **chain** *Chain_name* of **table** *TABLE_NAME*, the rule below will be appended:

if *several_conditions* is satisfied, then *action* will be taken.

Here **adding** the rule is specified by –A. You can also change to -D for **deletion** or –I for **insertion** (at somewhere of the chain). If -t does not occur, the default table **filter** is assumed.

Important: for each problem, provide your screen shot on the whole experiment record (if any) of each VM (involved). Also briefly describe the command and why the event happens (even if I have described in the problem statement, your description will show your understanding).

1. Use the following commands on a VM **(10.0.2.4)** to set the default policies for a table**.**

Sudo iptables –P INPUT DROP

Sudo iptables –P OUTPUT DROP

Sudo iptables –P FORWARD DROP

Recall, INPUT is to check incoming packet; OUTPUT is to check outgoing packet; FORWARDING is to check the passing packet (at router). Further, the commands assume the default table **filter (-t filter)**.

- Run **$ ping 10.0.2.4** from another VM and run **$ping 10.0.2.5 from 10.0.2.4**. You will see the operation will fail.
- Change **DROP** to **ACCEPT** in all the three commands. Try the pings in the above step again. You will see now it succeeds.

2. [**blocking an IP**]

- if we want to block packets **from** an ip address 10.0.2.5, use command
  sudo iptables -A INPUT -s 10.0.2.5 -j DROP
  /*this uses INPUT chain as it is incoming packet***/**
  ping the current VM from 10.0.2.5 and what can be observed on 10.0.2.5?
- if we want to block packets **to** an ip address 10.0.2.5, use command
  sudo iptables -A OUTPUT -d 10.0.2.5 -j DROP
  /*this uses OUTPUT chain as it is outgoing packet***/**

ping to 10.0.2.5 and what can be observed by the current VM.

3. [List all rules]

- You can see all the firewall rules by the following command
  $ sudo iptables -L
  /* again, this assume filter table (i.e., **-t filter**) by default*/
- You can see all the fire rules in each chain with index number. The index will be used for other operation such as deletion later.
  $ sudo iptables -L --line-number

4. [Delete a rule] To delete a rule in a chain (such as INPUT), we can first list with index:

$ sudo iptables –L INPUT --line-number

```
[11/11/20]seed@VM:~$ sudo iptables -L INPUT --line-number
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       all  --  10.0.2.5             anywhere
[11/11/20]seed@VM:~$
```

Then, delete the rule using the index:

$sudo iptables -D INPUT 1

Run **$ sudo iptables -L INPUT** to verify whether rule 1 is deleted or not.

5.[Delete all rules in a TABLE] This can be done by flushing the rules in a table (e.g., **filter**):

$sudo iptables -t filter -F

/*again,**-t filter** can be omitted*/

Then, run **$sudo iptables -L** and you will not see any rule.

6 [**Drop all incoming connections, except SSH**] To block connections to any server on the current VM (10.0.2.4) except for SSH server, we can set default policy for INPUT chain of filter Table to be DROP and then specify a rule to accept incoming SSH connection.

$ sudo iptables -P INPUT DROP

$ sudo iptables -A INPUT -p tcp - -dport 22 -j ACCEPT

/* we remind that the default policy is applied only if all the rules in the chain have been executed without a decision (either ACCEPT or DROP or REJECT). Here **-p** stands for protocol. */

Then, ping and ssh to this VM 10.0.2.4 (from other VM); you can see that only SSH succeeds.

/*after this problem, run **$ sudo iptables -F** to flush all rules in **filter** table and recover the default policy: **$ sudo iptables -P INPUT ACCEPT** */

7 [**drop outgoing DNS request to 8.8.8.8**] In this case, since it is outgoing packet, we add rule to OUTPUT chain. Since it is DNS request, the destination should be the DNS server, which has a port number 53. Finally, since DNS is implemented using UDP, we use protocol UDP. Hence, we add the following rule:

$ sudo iptables -A OUTPUT -p udp - - dport 53 -d 8.8.8.8 -j DROP

Then, try **$ dig** www.uwindsor.ca and **dig @8.8.8.8** www.uwindsor.ca to see what happens.
/* delete the rule in order not to affect the following experiment */

8 [**block incoming ping request**] You can not ping uwindsor webserver. Most l/ikely, this is blocked by firewall of uwindsor. Here is the way to block an incoming icmp request.

$ sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP

Then, ping the current VM from other VM to see what can be observed.

9 if you want to block all incoming connections (to your servers) but you do not want this to affect your access to external servers. But a problem arises. If you sends a request to a server, the latter should reply to you while this packet will be blocked by your firewall. To resolve this issue, you should regard the response packet (to your request) as related to your outgoing request packet and should be allowed to come in. This is achieved using the conntrack module.

$ sudo iptables -P INPUT DROP
$ sudo iptables -A INPUT -p tcp -m conntrack --ctstate RELATED, ESTABLISHED -j ACCEPT

Then, telnet to another VM. You will see it is still successful.
Next, telnet from the other VM to the firewall VM. You will see that it fails.

10 (optional) [**save your firewall rules and restore it**] After you have done firewall, you want to save your rules to a file you can run

$ sudo iptables-save >myiptables.rules

Later, you can restore your rules by running

$ sudo iptables-restore <myiptables.rules

/* to see the effect, you can flush your firewall after running iptables-save command and then run iptables-restore command to see if you have restored your firewall */