International Institute of Information Technology
Hyderabad
System and Network Security (CS5470)

Assignment 5: Configuring Firewalls using iptables
Total Marks: 100 [Report: 60, Quiz: 40]
Deadline: 9th April, Monday, 11:55 PM

Instructions:

The final submission will be a pdf report named **lab<no>_rollno**.
This assignment involves using **iptables** in Linux to manipulate firewall configuration. The Linux firewall is called netfilter, while the fundamental command to configure it is iptables. Use two virtual machines - a server and a client. All iptables commands will be executed on the server.

References:

1. Ubuntu Linux iptables HOWTO, https://help.ubuntu.com/community/IptablesHowTo
2. iptables man page, http://ipset.netfilter.org/iptables.man.html(or use man iptables)

By default, there are no rules configured as can be seen from the output of the command (iptables -L) below.

```
$sudo iptables -L
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
```

Packets are filtered as per their origin and the chains of rule are created accordingly. In the output above, three default chains are displayed, each for a different origin. The INPUT chain is for packets that are sent to the host on which the firewall is configured. The OUTPUT chain is for packets that are sent from the host on which the firewall is configured. The FORWARD chain determines which packets are routed to the next hop on hosts configured as routers or firewalls between two networks.

Since this assignment does not require routing, the FORWARD chain can be ignored.

The **iptables -F** command removes all configured firewall rules.

Setup

Configure two virtual machines using VirtualBox(https://www.virtualbox.org/) - one for server and one for client. By default the network adapter configured is NAT. Add another **host-only or internal network** adapter on both VMs. Configure the **added** adapter interfaces appropriately with static IP addresses. The configuration should survive reboot.

For example, Client IP address can be 10.1.1.19, Server IP address can be 10.1.1.20 and the netmask is 255.255.255.0.

On the server, verify the **iptables rules** as follows. The –n option lists machine addresses and ports as numbers rather than names.

$sudo iptables –n -L

For the below questions, take screenshots of the rules added (**using iptables -n -L command**) and the effect of adding the rules. Ping the server from client VM to verify. Also show all open ports at each instance on the server. Explain the result achieved briefly.

1. Create rules to allow only ssh and port 80 traffic

2. Create rules to accept traffic only on a range of ports

3. Add rules to deny packets to a specific domain (eg-www.google.com)

4. Create rules to prevent IP spoofing and ping of death attacks on local network.
Also explain the attacks briefly with ports involved and how to block them.

.