Sadiq S.

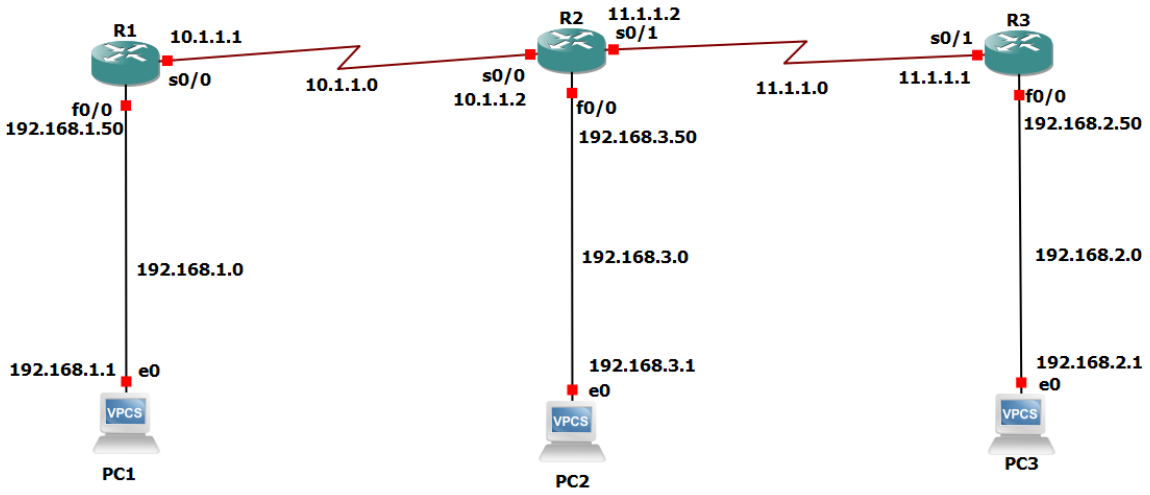
Practical 2

AIM: Implement IPv4 ACLs

1. Standard

2. Extended

Step 1: Build the network as follow:



Step 2: Explain what ACL is and how we apply it in the current system.

- Access-list (ACL) is a set of rules defined for controlling network traffic and reducing network attacks. ACLs are used to filter traffic based on the set of rules defined for the incoming or outgoing of the network.

- In a way, an ACL is like a guest list at an exclusive club.

- ACL is also called as packet filtering firewall

- There are two main different types of Access-list namely:
  1. **Standard Access-list** – These are the Access-list that are made using the source IP address only. These ACLs permit or deny the entire protocol suite. They don't distinguish between the IP traffic such as TCP, UDP, HTTPS, etc. These use range 1-99 or 1300-1999.
     Here we only give source IP Address.
  2. **Extended Access-list** – These are the ACL that uses source IP, Destination IP, source port, and Destination port. These types of ACL, we can also mention which IP traffic should be allowed or denied. These use range 100-199 and 2000-2699.
     Here we give both the source and destination IP Address.

- To enable ACL on our network we use Routing protocols because via connection is not reachable so we will be applying a RIP.
- After applying RIP all the Routers and PCs are able to communicate and ping each other.

Step 3: Configure IP to all the routers and PC's

R1:

```
R1#en
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int f0/0
R1(config-if)#ip add 192.168.1.50 255.255.255.0
R1(config-if)#no shut
R1(config-if)#ex
R1(config)#
*Mar  1 00:15:24.855: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:15:25.855: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config)#int s0/0
R1(config-if)#ip add 10.1.1.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#ex
R1(config)#
*Mar  1 00:17:27.827: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:17:28.827: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
R1(config)#do wr
Building configuration...
[OK]
```

R2:

```
R2#en
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#int f0/0
R2(config-if)#ip add 192.168.3.50 255.255.255.0
R2(config-if)#no shut
R2(config-if)#ex
R2(config)#
*Mar  1 00:22:31.619: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:22:32.619: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2(config)#int s0/0
R2(config-if)#ip ad  10.1.1.2 255.255.255.0
% Ambiguous command:  "ip ad  10.1.1.2 255.255.255.0"
R2(config-if)#ip add  10.1.1.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#ex
R2(config)#
*Mar  1 00:23:11.791: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:23:12.791: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
R2(config)#int s0/1
R2(config-if)#ip add 11.1.1.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#ex
```

R3:

```
R3#en
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#int f0/0
R3(config-if)#ip add 192.168.2.50 255.255.255.0
R3(config-if)#no shut
R3(config-if)#ex
R3(config)#int s
*Mar  1 00:27:04.679: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:27:05.679: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R3(config)#int s0/1
R3(config-if)#ip add 11.1.1.1 255.255.255.0
R3(config-if)#no shut
R3(config-if)#ex
R3(config)#
*Mar  1 00:27:30.699: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
*Mar  1 00:27:31.699: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to up
R3(config)#exit
```

PC1:

```
PC1> ip 192.168.1.1 255.255.255.0 192.168.1.50
Checking for duplicate address...
PC1 : 192.168.1.1 255.255.255.0 gateway 192.168.1.50

PC1> sh ip

NAME        : PC1[1]
IP/MASK     : 192.168.1.1/24
GATEWAY     : 192.168.1.50
DNS         :
MAC         : 00:50:79:66:68:02
LPORT       : 10026
RHOST:PORT  : 127.0.0.1:10027
MTU:        : 1500
```

PC2:

```
PC2> ip 192.168.3.1 255.255.255.0 192.168.3.50
Checking for duplicate address...
PC1 : 192.168.3.1 255.255.255.0 gateway 192.168.3.50

PC2> sh ip

NAME        : PC2[1]
IP/MASK     : 192.168.3.1/24
GATEWAY     : 192.168.3.50
DNS         :
MAC         : 00:50:79:66:68:01
LPORT       : 10024
RHOST:PORT  : 127.0.0.1:10025
MTU:        : 1500
```

PC3:

```
PC3> ip 192.168.2.1 255.255.255.0 192.168.2.50
Checking for duplicate address...
PC1 : 192.168.2.1 255.255.255.0 gateway 192.168.2.50

PC3> sh ip

NAME        : PC3[1]
IP/MASK     : 192.168.2.1/24
GATEWAY     : 192.168.2.50
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10028
RHOST:PORT  : 127.0.0.1:10029
MTU:        : 1500
```

Step 4: After assigning the IP Address to all the router and PC try ping the neighbor PC or router. It should work successfully.

R1:

```
R1#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/16/28 ms
R1#ping 10.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/26/32 ms
R1#router rip
```

R2:

```
R2#ping 192.168.3.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/18/32 ms
R2#ping 10.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/32 ms
R2#ping 11.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
R2#conf t
```

R3:

```
[OK]
R3#ping 192.168.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/16/28 ms
R3#ping 11.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/29/32 ms
R3#conf t
```

After that try ping to indirect connection. It won't be able to ping. Because we haven't configured any routing protocols for indirect connection to work.

So we will now configure RIP on all the router

Step 5: Follow below to configure RIP in routers.

R1:

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router rip
R1(config-router)#network 192.168.1.0
R1(config-router)#network 10.1.1.0
R1(config-router)#ex
R1(config)#do wr
Building configuration...
[OK]
```

R2:

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router rip
R2(config-router)#network 10.1.1.0
R2(config-router)#network 11.1.1.0
R2(config-router)#network 192.168.3.0
R2(config-router)#ex
R2(config)#do wr
Building configuration...
[OK]
R2(config)#do sh ip route
```

R3:

```
R3(config)#router rip
R3(config-router)#network 11.1.1.0
R3(config-router)#network 192.168.2.0
R3(config-router)#ex
R3(config)#do wr
Building configuration...
[OK]
```

Once RIP is configured ping indirect connection. Now ping will work as we have applied RIP protocol.

Step 6: Now we will apply ACL.

As we have discussed there are 2 types of ACL. We will apply Standard ACL now.

R1:

```
R1(config)#access-list 10 deny host 192.168.2.1
R1(config)#ex
% Ambiguous command:  "ex"
R1(config)#exit
R1#
*Mar  1 01:05:48.075: %SYS-5-CONFIG_I: Configured from console by console
R1#show access-list
Standard IP access list 10
    10 deny   192.168.2.1
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#access-list 10 permit any
R1(config)#int s0/0
R1(config-if)#ip access-group 10 in
R1(config-if)#exit
R1(config)#do wr
Building configuration...
[OK]
```

Here the scenario we are going to use is that whatever data that is being sent by PC 3 should not be received by PC 1, so based on the features of a standard ACL, we know that it needs to be place near the destination.

The number of the standard ACL should be in the range of 1-99, in our case we have chosen 10.

So by configuring the access list to 'deny', we deny all communication from PC 3 to PC 3.

Here we can observe that when PC 3 tries to send data to PC 1, its access is denied.

```
PC3> ping 192.168.1.1
*10.1.1.1 icmp_seq=1 ttl=253 time=76.089 ms (ICMP type:3, code:13, Communication administratively prohibited)
*10.1.1.1 icmp_seq=2 ttl=253 time=59.842 ms (ICMP type:3, code:13, Communication administratively prohibited)
*10.1.1.1 icmp_seq=3 ttl=253 time=60.578 ms (ICMP type:3, code:13, Communication administratively prohibited)
*10.1.1.1 icmp_seq=4 ttl=253 time=59.990 ms (ICMP type:3, code:13, Communication administratively prohibited)
*10.1.1.1 icmp_seq=5 ttl=253 time=60.313 ms (ICMP type:3, code:13, Communication administratively prohibited)
```

R3:

```
[OK]
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#access-list 40 deny host 192.168.3.1
R3(config)#do sh access-list
Standard IP access list 40
    10 deny   192.168.3.1
R3(config)#access-list 40 permit any
R3(config)#int s0/1
R3(config-if)#ip access-group 40 in
R3(config-if)#ex
R3(config)#exit
R3#
```

Here R3 have block PC2. So PC3 wont be receiving data from PC2.

```
PC2> ping 192.168.2.1
*11.1.1.1 icmp_seq=1 ttl=254 time=59.942 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.1 icmp_seq=2 ttl=254 time=60.443 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.1 icmp_seq=3 ttl=254 time=46.001 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.1 icmp_seq=4 ttl=254 time=61.334 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.1 icmp_seq=5 ttl=254 time=60.528 ms (ICMP type:3, code:13, Communication administratively prohibited)
```

R2:

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#access-list 10 deny 192.168.1.1
R2(config)#exit
R2#
*Mar  1 01:19:27.295: %SYS-5-CONFIG_I: Configured from console by console
R2#sh access-list
Standard IP access list 10
    10 deny   192.168.1.50 (15 matches)
    30 deny   192.168.1.1
    20 permit any (69 matches)
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#access-list 10 permit any
R2(config)#int s0/0
R2(config-if)#ip access-group 10 in
R2(config-if)#exit
R2(config)#access-list 20 deny icmp host 192.168.1.1 host 192.168.2.1
```

Here R2 have block PC1. So if PC1 send data PC3 it will deny.

```
PC1> ping 192.168.3.1
*192.168.1.50 icmp_seq=1 ttl=255 time=44.946 ms (ICMP type:3, code:1, Destination host unreachable)
*192.168.1.50 icmp_seq=2 ttl=255 time=28.260 ms (ICMP type:3, code:1, Destination host unreachable)
*192.168.1.50 icmp_seq=3 ttl=255 time=18.311 ms (ICMP type:3, code:1, Destination host unreachable)
*192.168.1.50 icmp_seq=4 ttl=255 time=16.307 ms (ICMP type:3, code:1, Destination host unreachable)
*192.168.1.50 icmp_seq=5 ttl=255 time=30.731 ms (ICMP type:3, code:1, Destination host unreachable)
```

Step 7: We will now apply Extended ACL

R3:

```
R3(config)#access-list 121 deny icmp host 192.168.3.1 host 192.168.2.1
R3(config)#do sh access-list 121
Extended IP access list 121
    10 deny icmp host 192.168.3.1 host 192.168.2.1
R3(config)#access-list 121 permit icmp any any
R3(config)#do sh access-list 121
Extended IP access list 121
    10 deny icmp host 192.168.3.1 host 192.168.2.1
    20 permit icmp any any
R3(config)#int s0/1
R3(config-if)#ip access-group 121 out
R3(config-if)#do sh access-list 121
Extended IP access list 121
    10 deny icmp host 192.168.3.1 host 192.168.2.1
    20 permit icmp any any
R3(config-if)#exit
R3(config)#exit
```

We will apply ACL on R3. It will deny PC2 to send data to PC3.

Here we can observe that when PC 2 tries to send data to PC 3, its access is denied.

```
PC2> ping 192.168.2.1
*11.1.1.1 icmp_seq=1 ttl=254 time=61.248 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.1 icmp_seq=2 ttl=254 time=30.668 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.1 icmp_seq=3 ttl=254 time=45.552 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.1 icmp_seq=4 ttl=254 time=47.201 ms (ICMP type:3, code:13, Communication administratively prohibited)
*11.1.1.1 icmp_seq=5 ttl=254 time=45.955 ms (ICMP type:3, code:13, Communication administratively prohibited)
```

But if any other device ping PC 3 it will permit it

```
Success rate is 80 percent (4/5), round-trip min/avg/max = 44/49/64 ms
R2#ping 192.168.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/36/60 ms
R2#
```

Here PC3 have accept R2 but deny PC2.

We can assume that we have successfully used an extended access list.