CSE 4512 [Computer Networks Lab]

Lab # 10

1. Objectives:

- Describe the concept of Access Control List (ACL)
- Implement standard numbered ACL

2. Theory:

Access Control Lists (ACL):

Defining ***who*** *can/can't access* ***what*** is basically the gist of ACL. In our day-to-day life, we are applying the concept of ACL in many areas. A simple example could be like you need to show your ID card to enter an office. There's a list of employees and your ID is checked against that list to grant access. Similar access controls are in effect in virtually everywhere, especially in places where security is critical. In digital world, this access control is more needed so that only allowed ones can access a certain digital resource. For example, only admins would be allowed access in the backend of a web server or only database admins would be allowed to access database server etc.

In networked devices, ACLs play a crucial role to allow only authorized person/devices to a certain resource. For example, you can define that only a certain host device would be able to access your webserver. You can also define ACLs so that hosts belonging to a particular network can't communicate with hosts of certain other network. There are more scenarios that can be defined depending on the needs of an administrator.

In this lab, we'll learn about Cisco IP ACL i.e., filtering network traffic based on IP address. There are several ACL types that can be configured on a Cisco device. But for the purpose of this lab, we'll only focus on *Numbered Standard IPv4 ACL*. There are two steps to implement an ACL. First, **define the rule**. Second, **apply the rule to an interface**.

The command format for defining a numbered standard IP ACL is:

```
Router(config)# access-list access-list-number
                {permit|deny}
               {source_address source_wildcard|any}
```

You can either permit or deny a packet based on the source IP of the packet in numbered standard IP ACL. As like the OSPF configuration, you need to specify a wildcard mask to permit/deny a range of source IP addresses based on the given pattern. One important thing you should keep in mind that whenever you apply an ACL to an interface, **all the traffic that doesn't match any ACL rule will be discarded by default**. So, for example, you have defined an ACL to deny a certain source IP. Whenever you apply that rule to an interface, all other packets other than the denied source will also be discarded

because there's no matching rule for those packets. So, you must allow other traffic explicitly by defining another ACL. The **any** keyword is handy in this case. To permit (or deny) any packet other than the previously specified rules, you can just add the keyword **any** in place of the source_address and source_wildcard like the following:

```
Router(config)# access-list 1 permit any
```

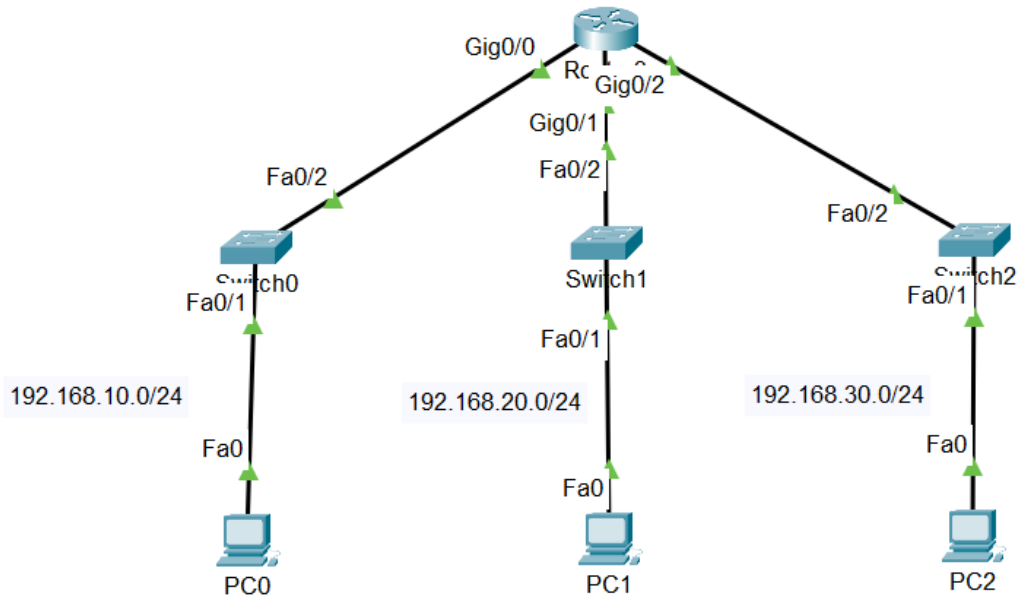
Another thing is you can only use numbers in the range *1 to 99* for specifying the *access-list-number*. Other numbers are used for **extended numbered ACL**. After defining the ACL rule, now we need to apply it to an interface. Remember that the ACL has no effect until you apply it. The command format for applying an ACL to an interface is like below:

```
Router(config-if)# ip access-group access-list-number {in|out}
```

The ACL is applied either for inbound traffic or outbound traffic of an interface and you need to specify the corresponding keyword i.e., **in** or **out** for that. One best practice before applying an ACL to an interface is to *verify* the rule by using the following command:

```
Router# show access-lists
```

3. Configure ACL:



I. **Configure Router Interfaces**
Router`(config)# int g0/0`

```
Router(config-if)# ip address 192.168.10.1 255.255.255.0
Router(config-if)# no shutdown
Router(config)# int g0/1
Router(config-if)# ip address 192.168.20.1 255.255.255.0
Router(config-if)# no shutdown
Router(config)# int g0/2
Router(config-if)# ip address 192.168.20.1 255.255.255.0
Router(config-if)# no shutdown

Router(config-if)# exit
Router# copy running-config startup-config
```

II. Configure PC0

```
IP: 192.168.10.5
Mask: 255.255.255.0
Gateway: 192.168.10.1
```

III. Configure PC1

```
IP: 192.168.20.5
Mask: 255.255.255.0
Gateway: 192.168.20.1
```

IV. Configure PC2

```
IP: 192.168.30.5
Mask: 255.255.255.0
Gateway: 192.168.30.1
```

V. Define ACL

```
Router(config)# access-list 1 deny 192.168.10.0 0.0.0.255
Router(config)# access-list 1 permit any
```

VI. Verify ACL

```
Router# show access-lists
```

VII. Apply ACL

```
Router(config)# interface gigabitEthernet 0/2
Router(config-if)# ip access-group 1 out
```

4. Tasks:

I. You will configure *numbered standard ACL* following the instructions given in the task. The task description for this task is provided in the pdf ***Task-1_configure-standard-ipv4-acls***. You're provided a .pka file for this task.

Packet Tracer - Configure Numbered Standard IPv4 ACLs

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	192.168.10.1	255.255.255.0	N/A
	G0/1	192.168.11.1	255.255.255.0	
	S0/0/0	10.1.1.1	255.255.255.252	
	S0/0/1	10.3.3.1	255.255.255.252	
R2	G0/0	192.168.20.1	255.255.255.0	N/A
	S0/0/0	10.1.1.2	255.255.255.252	
	S0/0/1	10.2.2.1	255.255.255.252	
R3	G0/0	192.168.30.1	255.255.255.0	N/A
	S0/0/0	10.3.3.2	255.255.255.252	
	S0/0/1	10.2.2.2	255.255.255.252	
PC1	NIC	192.168.10.10	255.255.255.0	192.168.10.1
PC2	NIC	192.168.11.10	255.255.255.0	192.168.11.1
PC3	NIC	192.168.30.10	255.255.255.0	192.168.30.1
WebServer	NIC	192.168.20.254	255.255.255.0	192.168.20.1

Objectives

Part 1: Plan an ACL Implementation

Part 2: Configure, Apply, and Verify a Standard ACL

Background / Scenario

Standard access control lists (ACLs) are router configuration scripts that control whether a router permits or denies packets based on the source address. This activity focuses on defining filtering criteria, configuring standard ACLs, applying ACLs to router interfaces, and verifying and testing the ACL implementation. The routers are already configured, including IP addresses and Enhanced Interior Gateway Routing Protocol (EIGRP) routing.

Instructions

Part 1: Plan an ACL Implementation

Step 1: Investigate the current network configuration.

Before applying any ACLs to a network, it is important to confirm that you have full connectivity. Verify that the network has full connectivity by choosing a PC and pinging other devices on the network. You should be able to successfully ping every device.

Step 2: Evaluate two network policies and plan ACL implementations.

a. The following network policies are implemented on **R2**:

- The 192.168.11.0/24 network is not allowed access to the **WebServer** on the 192.168.20.0/24 network.
- All other access is permitted.

To restrict access from the 192.168.11.0/24 network to the **WebServer** at 192.168.20.254 without interfering with other traffic, an ACL must be created on **R2**. The access list must be placed on the outbound interface to the **WebServer**. A second rule must be created on **R2** to permit all other traffic.

b. The following network policies are implemented on **R3**:

- The 192.168.10.0/24 network is not allowed to communicate with the 192.168.30.0/24 network.
- All other access is permitted.

To restrict access from the 192.168.10.0/24 network to the 192.168.30/24 network without interfering with other traffic, an access list will need to be created on **R3**. The ACL must be placed on the outbound interface to **PC3**. A second rule must be created on **R3** to permit all other traffic.

Part 2: Configure, Apply, and Verify a Standard ACL

Step 1: Configure and apply a numbered standard ACL on R2.

a. Create an ACL using the number **1** on **R2** with a statement that denies access to the 192.168.20.0/24 network from the 192.168.11.0/24 network.

b. By default, an access list denies all traffic that does not match any rules. To permit all other traffic, configure the following statement:

c. Before applying an access list to an interface to filter traffic, it is a best practice to review the contents of the access list, in order to verify that it will filter traffic as expected.

```
R2# show access-lists
Standard IP access list 1
        10 deny 192.168.11.0 0.0.0.255
        20 permit any
```

d. For the ACL to actually filter traffic, it must be applied to some router operation. Apply the ACL by placing it for outbound traffic on the GigabitEthernet 0/0 interface. Note: In an actual operational network, it is not a good practice to apply an untested access list to an active interface.

Step 2: Configure and apply a numbered standard ACL on R3.

a. Create an ACL using the number **1** on **R3** with a statement that denies access to the 192.168.30.0/24 network from the **PC1** (192.168.10.0/24) network.

b. By default, an ACL denies all traffic that does not match any rules. To permit all other traffic, create a second rule for ACL 1.

c. Verify that the access list is configured correctly.

```
R3# show access-lists
Standard IP access list 1
        10 deny 192.168.10.0 0.0.0.255
        20 permit any
```

d. Apply the ACL by placing it for outbound traffic on the GigabitEthernet 0/0 interface.

Step 3: Verify ACL configuration and functionality.

a. Enter the **show run** or **show ip interface gigabitethernet 0/0** command to verify the ACL placements.

b. With the two ACLs in place, network traffic is restricted according to the policies detailed in Part 1. Use the following tests to verify the ACL implementations:

- A ping from 192.168.10.10 to 192.168.11.10 succeeds.
- A ping from 192.168.10.10 to 192.168.20.254 succeeds.
- A ping from 192.168.11.10 to 192.168.20.254 fails.
- A ping from 192.168.10.10 to 192.168.30.10 fails.
- A ping from 192.168.11.10 to 192.168.30.10 succeeds.
- A ping from 192.168.30.10 to 192.168.20.254 succeeds.

c. Issue the **show access-lists** command again on routers **R2** and **R3**. You should see output that indicates the number of packets that have matched each line of the access list. Note: The number of matches shown for your routers may be different, due to the number of pings that are sent and received.

```
R2# show access-lists
Standard IP access list 1
        10 deny 192.168.11.0 0.0.0.255 (4 match(es))
        20 permit any (8 match(es))

R3# show access-lists
Standard IP access list 1
        10 deny 192.168.10.0 0.0.0.255 (4 match(es))
        20 permit any (8 match(es))
```