LAB#8 Access Control Lists (ACLs)

Theory

The Cisco Access Control List (ACL) is are used for filtering traffic based on a given filtering criteria on a router or switch interface. Based on the conditions supplied by the ACL, a packet is allowed or blocked from further movement. Cisco ACLs are available for several types of routed protocols including IP, IPX, AppleTalk, XNS, DECnet, and others. However, we will be discussing ACLs pertaining to TCP/IP protocol only. ACLs for TCP/IP traffic filtering are primarily divided into two types:

- *Standard Access Lists*
- *Extended Access Lists*

Standard Access Control Lists:

Standard IP ACLs range from 1 to 99. A Standard Access List allows you to permit or deny traffic FROM specific IP addresses. The destination of the packet and the ports involved can be anything.

This is the command syntax format of a standard ACL:

access-list access-list-number {permit|deny} {host|source source-wildcard|any}

Standard ACL example:

access-list 10 permit 192.168.2.0 0.0.0.255

access-list 10 deny any

This list allows traffic from all addresses in the range 192.168.2.0 to 192.168.2.255.

Note that when configuring access lists on a router, you must identify each access list uniquely by assigning either a name or a number to the protocol's access list. There is an implicit deny added to every access list.

Extended Access Control Lists:

Extended IP ACLs allow you to permit or deny traffic from specific IP addresses to a specific destination IP address and port. It also allows you to have granular control by specifying controls for different types of protocols such as ICMP, TCP, UDP, etc within the ACL statements. Extended IP ACLs range from 100 to 199. In Cisco IOS Software Release 12.0.1, extended ACLs began to use additional numbers (2000 to 2699)

The syntax for IP Extended ACL is given below:

access-list access-list-number {deny | permit} protocol source source-wildcard destination destination-wildcard [precedence precedence]

Note that the above syntax is simplified, and given for general understanding only.

Extended ACL example:

access-list 110 - Applied to traffic leaving the office (outgoing) access-list 110 permit tcp 92.128.2.0 0.0.0.255 any eq 80 ACL 110 permits traffic originating from any address on the 92.128.2.0 network. The 'any' statement means that the traffic is allowed to have any destination address with the limitation of going to port 80. The value of 0.0.0.0/255.255.255.255 can be specified as 'any'.

Applying an ACL to a router interface:

After the ACL is defined, it must be applied to the interface (inbound or outbound). The syntax for applying an ACL to a router interface is given below:

interface ip access-group {number|name} {in|out}

An Access List may be specified by a name or a number. "in" applies the ACL to the inbound traffic, and "out" applies the ACL on the outbound traffic.

Example: To apply the standard ACL created in the previous example, use the following commands:

Router(config)#interface serial 0 Router(config-if)#ip access-group 10 out

WILD-CARD MASK With ACLs:

You will have a variety of uses for the wild card masks. A wildcard mask is a mask of bits that indicates which parts of an IP address are available for examination. In the Cisco IOS, they are used in several places, for example:

• To indicate the size of a network or subnet for some routing protocols, such as OSPF.

• To indicate what IP addresses should be permitted or denied in access control lists (ACLs). Wildcard mask is an inverse of subnet mask. It is a matching rule where:

• 0 means that the equivalent bit must match

• 1 means that the equivalent bit does not matter – ignore In providing a wild card mask, it should be able to do the following:

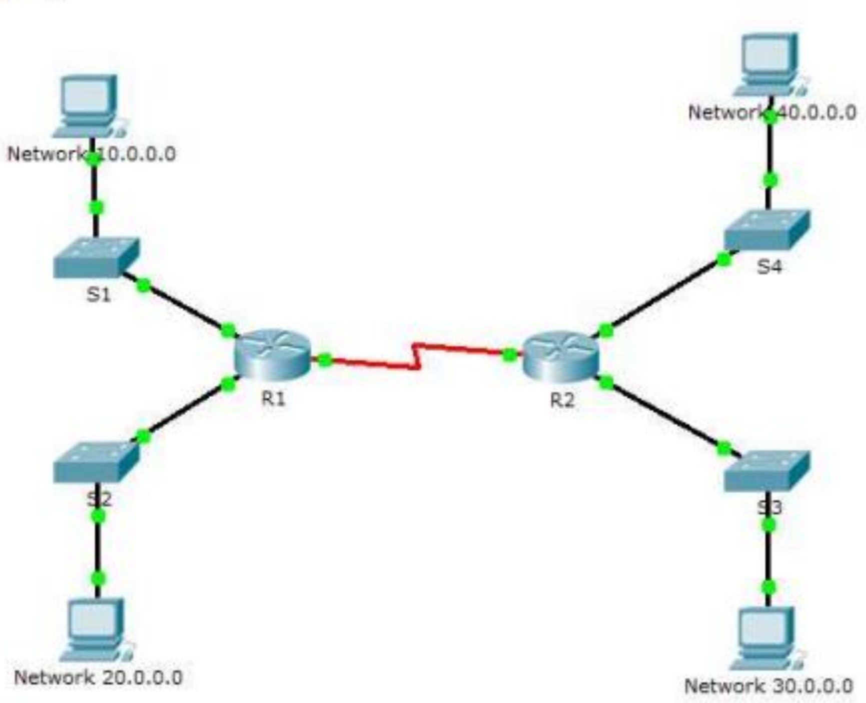
1. Match a specific host

2. Match the entire subnet

3. Match an IP range, or

4. Match everyone and anyone

Topology:



Match a specific host Task:

You have given a task to block 10.0.0.3 from gaining access on 40.0.0.0. While 10.0.0.3 must be able to communicate with networks. Other computer from the network of 10.0.0.0 must be able to connect with the network of 40.0.0.0.

Decide where to apply ACL and in which directions. Our host must be able to communicate with other host except 40.0.0.0 so we will place this access list on FastEthernet 0/1 of R2 connected to the network of 40.0.0.0. Direction will be outside as packet will be filter while it's leaving the interface. If you place this list on R1 then host 10.0.0.3 will not be able to communicate with any other hosts including 40.0.0.0.

To configure R2 double click on it and select CLI (Choose only one method result will be same) The configuration will be:

```
 R2>enable R2#configure terminal

R2(config)#access-list 1 deny 10.0.0.3 0.0.0.0

R2(config)#access-list 1 permit any

R2(config)#interface fastEthernet 0/1

 R2(config-if)#ip access-group 1 out .
```

To test, first do ping from 10.0.0.3 to 40.0.0.3 it should be request time out as this packet will filter by ACL. Then ping 30.0.0.3 it should be successfully reply. As we applied access list only on specific host so other computer from the network of 10.0.0.0 must be able to connect with the network of 40.0.0.0. To test, do ping from 10.0.0.2 to 40.0.0.3. It should be successfully reply. Match an entire subnet Task: You have given a task to the network of 10.0.0.0 from gaining access on 40.0.0.0. While 10.0.0.0 must be able to communicate with networks. Wildcards are used with access lists to specify an individual host, a network, or a certain range of a network or networks. The key to matching an entire subnet is to use the following formula for the wildcard mask. It goes as follows: Wildcard mask = 255.255.255.255 – subnet.

So for example if my current subnet was 255.0.0.0, the mask would be 0.255.255.255.

Once you have calculated the wild card mask, the rest is same as we did in pervious example:

```
 R2>enable

R2(config)#access-list 2 deny 10.0.0.0 0.255.255.255

R2(config)#access-list 2 permit any

R2(config)#interface fastethernet 0/1

R2(config-if)#ip access-group 2 out

R2(config-if)#
```

To test, first do ping from 10.0.0.3 to 40.0.0.3 it should be request time out as this packet will filter by ACL. Then ping 30.0.0.3 it should be successfully reply. Now do ping from 10.0.0.2 to 40.0.0.3 and further 30.0.0.2 result should be same as the packet is filtering on network based. Match an IP range Task: You are a network administrator at ComputerNetworkingNotes.com. You task is to block an ip range of 10.3.16.0 – 10.3.31.255 from gaining access to the network of 40.0.0.0 Solutions: Our range is 10.3.16.0 – 10.3.31.255. In order to find the mask, take the higher IP and subtract from it the lower IP. 10.3.31.255 10.3.16.0 .

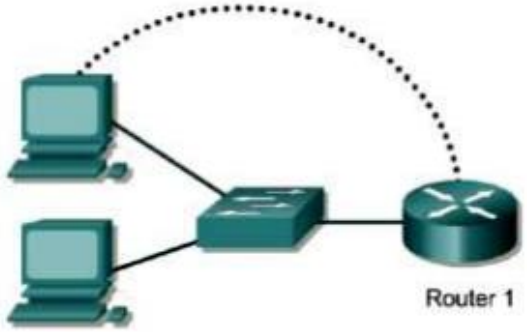
In this case the wildcard mask for this range is 0.0.15.255. To permit access to this range, you would use the following:

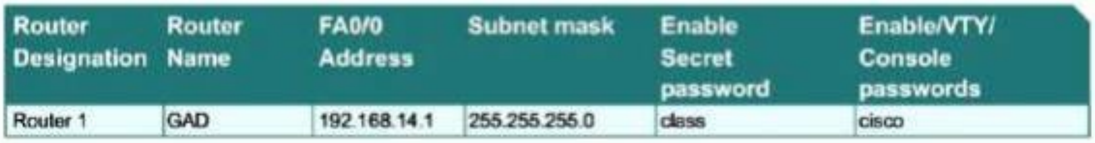
```
R2>enable

R2(config)#access-list 2 deny 10.3.16.0 0.0.15.255
```

```
R2(config)#access-list 2 permit any
R2(config)#interface fastethernet 0/1
R2(config-if)#ip access-group 2 out
R2(config-if)#
```

LAB: Configure router with ACL





Router Designation	Router Name	FA0/0 Address	Subnet mask	Enable Secret password	Enable/VTY/ Console passwords
Router 1	GAD	192.168.14.1	255.255.255.0	class	cisco

Objective

- *Configure, and apply a standard ACL to permit or deny specific traffic.*
- *Test the ACL to determine if the desired results were achieved.*

Draw a network as illustrated above by using any router from 1900 family with appropriate configuration details in the table and follow all the steps given below:

Step 1 : *Configure the hostname and passwords on the Router1*

a. On the Router1, enter the global configuration mode and configure the hostname as shown in the table. Then configure the console, virtual terminal and enable passwords. Configure the FastEthernet interface on the router according to the table.

Step 2 *: Configure the hosts on the Ethernet segment:*

a. Host 1 IP address 192.168.14.2 Subnet mask 255.255.255.0 Default gateway 192.168.14.1

b. Host 2 IP address 192.168.14.3 Subnet mask 255.255.255.0 Default gateway 192.168.14.1

***Step 3**: Save the configuration information from the privileged EXEC command mode GAD#copy running-config startup-config*

***Step 4:** Confirm connectivity by pinging the default gateway from both hosts*

a. If the pings are not successful, correct the configuration and repeat until they are successful.

***Step 5:** Prevent access to the Ethernet interface from the hosts*

a. Create an access list that will prevent access to FastEthernet 0 from the 192.168.14.0 network.

b. At the router configuration prompt type the following command:

```
 GAD(config)#access-list 1 deny 192.168.14.0 0.0.0.255
GAD(config)#access-list 1 permit any
```

c. Why is the second statement needed?

***Step 6** : Ping the router from the hosts :*

a. Were these pings successful?

b. Why or why not?

***Step 7:** Apply the Access list to the interface :*

a. At the FastEthernet 0 interface mode prompt type the following:

```
GAD(config-if)#ip access-group 1 in.
```

***Step 8** : Ping the router from the hosts :*

a. Were these pings successful?

b. Why or why not?

***Step 9**: Create a new access list :*

a. Now create an access list that will not allow the even numbered hosts to ping but permit the odd numbered hosts to ping.

b. What will that access list look like?

Finish this command with an appropriate comparison IP address (aaa.aaa.aaa.aaa) and wildcard mask (www.www.www.www):

```
access-list 2 permit aaa.aaa.aaa.aaa www.www.www.www
```

c. Why was it not necessary to have the permit any statement at the end this time?

***Step 10:** Apply access list to the proper router interface:*

a. First remove the old access list application by typing no ip access-group 1 in at the interface configuration mode.

b. Apply the new access list by typing ip access-group 2 in

Step 11*: Ping the router from each hosts :*

a. Was the ping from host 1 successful?

b. Why or why not?

c. Was the ping from host 2 successful?

d. Why or why not?

Upon completion of the previous steps, logoff by typing exit. Please erase the configuration file