Access Control Lists (ACLs)

Access control lists (ACLs) are set of rules which allows you to permit or deny packets based on source and destination IP address, IP protocol information, or TCP or UDP protocol information. You can configure the following types of ACLs:

- **Standard** – Permits or denies packets based on source IP address. Valid standard ACL IDs are 1 – 99
- **Extended** – Permits or denies packets based on source and destination IP address and also based on IP protocol information. Valid extended ACL IDs are a number from 100 – 199

Access-lists use wildcard masks to match traffic.Access control lists (ACLs) can be used for two purposes on Cisco devices:

- **To filter traffic**
- **To identify traffic**

When filtering traffic, access lists are applied on interfaces. As a packet passes through a router, the top line of the rule list is checked first, and the router continues to go down the list until a match is made. Once a match is made, the packet is either permitted or denied.

NOTE : ***There is an implicit 'deny all' at the end of all access lists. We cant delete it.So an access lists that contain only deny statements will prevent all traffic.If you want ACL to allow traffic there must be a permit statement***

Access lists are applied either inbound (packets received on an interface, before routing), or outbound (packets leaving an interface, after routing). **Only one access list per interface, per protocol, per direction is allowed.**

Even filtering traffic is the primary use of access lists, there are several instances when it is necessary to identify traffic using ACLs, including:

- Identifying interesting traffic to bring up an ISDN link or VPN tunnel
- Identifying routes to filter or allow in routing updates
- Identifying traffic for QoS purposes

Types of Access List

There are two categories of access lists:

Numbered ACL .it is the basic one.You cannot remove individual lines from a numbered access list. The entire access list must be deleted and recreated. All new entries to a numbered access list are added to the bottom. Best practice is to use a text editor to manage your access-lists. There are two common types of numbered access lists:

1. IP standard access lists
2. IP extended access lists

Named ACL provide more flexibility than Numbered access list.We can give names to identify your access-lists. individual lines can be removed from a named access-list. All new entries are added to the bottom of the access list like numbered ACL
There are two common types of named access lists:

1. IP standard named access lists
2. IP extended named access lists

How to permit or deny a specific host in Access list ?
we can use an example of 172.16.10.1 .As we want to block a specific address(host) in a network, we can use wildcard mask "**0.0.0.0**" .all octet in wildcard mask set to "0" means every octet must be matched.

There are actually two ways we can match a host:

- Using a wildcard mask "**0.0.0.0**" – 172.16.10.1 **0.0.0.0**
- Using the keyword "**host**" – host 172.16.10.1

Above method is use to match exactly a host.So how what we do to match the all address ?
There are actually two ways we can match all addresses:

- Using a wildcard mask "**255.255.255.255**" - 0.0.0.0 **255.255.255.255**
- Using the keyword "**any**" – any source or destination

Standard IP Access List
Syntax

*access-list [**1-99**] [**permit** | **deny**] [**source address**] [**wildcard mask**]*
Standard IP access-lists are based upon the source host or network IP address, and should be placed closest to the destination network. Range of standard access list is from 1-99

Example

Qn : Block network 172.20.0.0 from accessing the 172.19.0.0 network

- ***Router(config)# access-list 20 deny 172.20.0.0 0.0.255.255***
- ***Router(config)# access-list 20 permit any***

Note : Access list must be created on the router which is close to destination

- First line deny all hosts on the 172.20.x.x network.
- The second line uses a keyword of "any", which will match (permit) any other address.

Always remember that you must have at last one permit statement in your access list.otherwise all traffic will be blocked because of implicit deny at the end

Creating a access-list wont do anything it the network.It must be applied on an interface.To apply this access list, we would configure the following on Router:

- ***Router(config)# int s0***
- ***Router(config-if)# ip access-group 20 in***

To view all IP access lists configured on the router:

Router# show ip access-list

To view what interface an access-list is configured on:

- ***Router# show ip interface***
- ***Router# show running-config***

Extended IP Access List

Syntax

*access-list [**100-199**] [**permit | deny**] [**protocol**] [**source address**] [**wildcard mask**] [**destination address**] [**wildcard mask**] [**operator**] [**port**]*

Extended IP access-lists block based upon the source IP address, destination IP address, and TCP or UDP port number. Extended access-lists should be placed closest to the source network.

Example :

- ***access-list 100 deny tcp host 1.1.1.1 host 2.2.2.2 eq 23***
- ***access-list 100 deny tcp 3.3.3.0 0.0.0.255 any eq 80***
- ***access-list 100 permit ip any any***

1. The first line deny host 1.1.1.1 from accessing host 2.2.2.2 via telnet (port 23)

2. The second line deny http (eq port 80)access of 3.3.3.0 network
3. The third line allows all other traffic

Like our earlier example this ACL also be applied on interface to take effect.To apply this access list, we would configure the following command

- ***int fa 0/0***
- ***ip access-group 100 in***

In the above example we used eq port 80 to block http.Click here to view the list of common ports used

We can use several other operators for port numbers:

1. **eq** Matches a specific port
2. **gt** Matches all ports greater than the port specified
3. **lt** Matches all ports less than the port specified
4. **neq** Matches all ports except for the port specified
5. **range** Match a specific inclusive range of ports

The following will match all ports greater than 100:

Router(config)# access-list 101 permit tcp any host 172.16.10.10 gt 100

The following will match all ports less than 1024:

Router(config)# access-list 101 permit tcp any host 172.16.10.10 lt 1024

The following will match all ports that do not equal 443:

Router(config)# access-list 101 permit tcp any host 172.16.10.10 neq 443

The following will match all ports between 80 and 88:

Router(config)# access-list 101 permit tcp any host 172.16.10.10 range 80 88

Named Access Lists

Named access lists provide us with two advantages over numbered access lists. First, we can apply an identifiable name to an access list, for documentation purposes. Second, we can remove individual lines in a named access-list, which is not possible with numbered access lists.

Please note, though we can remove individual lines in a named access list, we cannot insert individual lines into that named access list. New entries are always placed at the bottom of a named access list

To create a standard named access list, the syntax would be as follows:

- *Router(config)# ip access-list standard NAME*
- *Router(config-std-nacl)# deny 172.18.0.0 0.0.255.255*
- *Router(config-std-nacl)# permit any*

To create an extended named access list, the syntax would be as follows:

- *Router(config)# ip access-list extended NAME*
- *Router(config-ext-nacl)# permit tcp 172.18.0.0 0.0.255.255 host 172.16.10.10 eq 80*
- *Router(config-ext-nacl)# deny ip 172.18.0.0 0.0.255.255 172.16.0.0 0.0.255.255*
- *Router(config-ext-nacl)# permit ip any any*

Troubleshooting

- *show access-lists [<number> | <name>]*
- *show ip access-lists [<number> | <name>]*
- *show ip access-lists interface <interface>*
- *show ip access-lists dynamic*
- *show ip interface [<interface>*