National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member:__ Umar Farooq __ **Dated: ____13/04/2022___**

Semester:___6th ______ **Section: _____D___**

EE-357 Computer and Communication Networks

Experiment - 9

Access Control Lists (ACL's)

		PLO5/ CLO3		PLO5/ CLO3	PLO5/ CLO3	PLO5/ CLO3
Name	**Reg. No**	**Viva / Quiz / Lab Performance 5 Marks**	**Analysis of data in Lab Report 5 Marks**	**Modern Tool Usage 5 Marks**	**Ethics and Safety 5 Marks**	**Individual and Team Work 5 Marks**
Myesha Khalil	**305093**					
Noor Ansar	**284825**					

National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

EXPERIMENT NO 9

Access Control Lists (ACL's)

1. Objective

- This lab exercise is designed to understand creating and applying ACLs on Cisco routers.

2. Resources Required

- Computer
- Packet Tracer (version 5 or higher)
- Visit following link for a detailed software demo *(use "ctrl+left click" to open following links)*:
  StandardL's
  Extended ACL's

3. Introduction

Cisco routers can be used as part of a good overall security strategy. One of the most important tools in Cisco IOS software used as a part of that strategy are Access Control Lists (ACLs—also called Access Lists). ACLs define rules that can be used to prevent some packets from flowing through the network. The access list is a group of statements. Each statement defines a pattern that would be found in an IP packet. As each packet comes through an interface with an associated access list, the list is scanned from top to bottom--in the exact order that it was entered--for a pattern that matches the incoming packet. A permit or deny rule associated with the pattern determines that packet's fate. You also can use a mask, which is like a wild card, to determine how much of an IP source or destination address to apply to the pattern match. The pattern statement also can include a TCP or UDP (User Datagram Protocol) port number.

3.1 Types of Access Lists

Access lists are generally broken into 2 major groups:

a) Standard ACLs only operate on the Network layer of the OSI model. These are used to block or permit networks from reaching other networks.

b) Extended ACLs Extended access lists function on both Network and Transport layers of the OSI model. That is, they allow you to filter not only by network address but also by the type of traffic that is being sent or received. Extended access lists are much more flexible and allow for much greater control of traffic into and out of your network than standard access lists.

3.2 Named Access Control Lists

The ACLs introduced in the beginning by Cisco IOS were distinguished by a special number e.g. 1-99 for Standard ACLs and 100-199 for Extended ACLs. The named ACLs (introduced with IOS version 11.2) do the same as standard and extended ACLs but they are not represented by a number but by a name that is given by the user to easily remember. In addition to using more memorable names, the other major advantage of named ACLs over numbered ACLs, at the time they were introduced into IOS, was that you could delete individual lines in named IP access list.

3.3 In, Out, Inbound, Outbound, Source, and Destination

The router uses the terms in, out, source, and destination as references. Traffic on the router can be compared to traffic on the highway. If you were a law enforcement officer in Pennsylvania and wanted to stop a truck going from Maryland to New York, the source of the truck is Maryland and the destination of the truck is New York. The roadblock could be applied at the Pennsylvania– New York border (out) or the Maryland–Pennsylvania border (in). When you refer to a router, these terms have these meanings.

a) Out—Traffic that has already been through the router and leaves the interface. The source is where it has been, on the other side of the router, and the destination is where it goes.

b) In—Traffic that arrives on the interface and then goes through the router. The source is where it has been and the destination is where it goes, on the other side of the router.

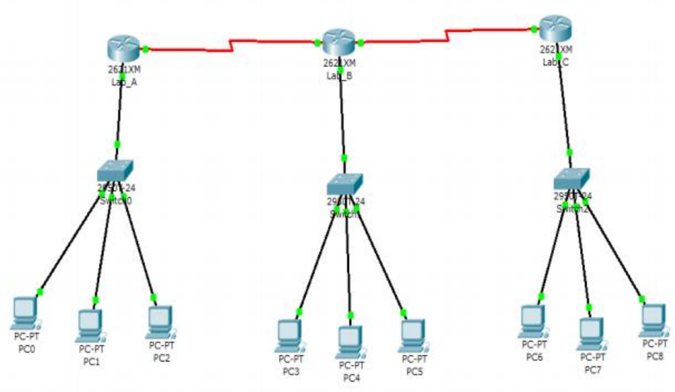
c) Inbound —If the access list is inbound, when the router receives a packet, the Cisco IOS software checks the criteria statements of the access list for a match. If the packet is permitted, the software continues to process the packet. If the packet is denied, the software discards the packet.

d) Outbound—If the access list is outbound, after the software receives and routes a packet to the outbound interface, the software checks the criteria statements of the access list for a match. If the packet is permitted, the software transmits the packet. If the packet is denied, the software discards the packet.

Thus in ACL has a source on a segment of the interface to which it is applied and a destination off of any other interface. The out ACL has a source on a segment of any interface other than the interface to which it is applied and a destination off of the interface to which it is applied.

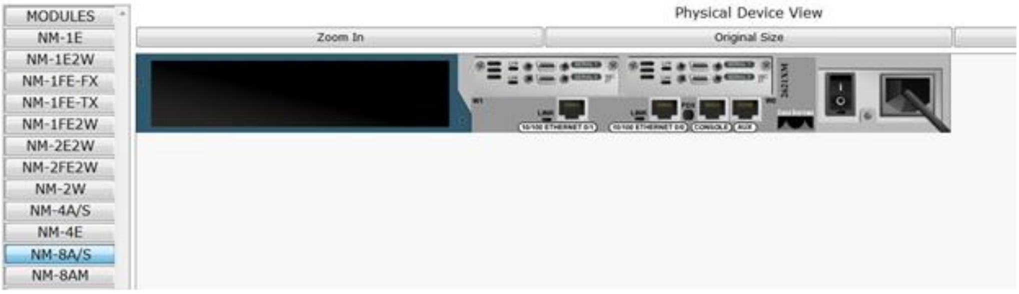
4. Procedure

1. Open Packet Tracer 5 and setup a network similar to the following network. Use Cisco 2950T switch & Cisco 2621XM router.



2. The different thing is the red link which is serial link (used for WAN). By default, it is not available so we have to add the modules to the router. Double click on any router. Turn it off by using power button on the router figure in **Physical** tab. On left side modules bar is present. Drag two **WIC-2T** to smaller blank space and one **NM-8A/S** to larger blank space. Now, turn on the router using power switch. Do the same on second router. Then use **Serial DTE** or **Serial DCE** link from **Connections**. The router interface that is chosen first becomes that of that type while the second one becomes the other e.g if you choose DTE and click first router, it becomes DTE while the second one

becomes DCE and vice versa. Just remember that by default all serial interfaces are DTE so we have to provide clocking on the DCE one!



3. Use the following values to setup IP addresses on respective interfaces.

Router	Network Address	Interface	Address
Lab_A	192.168.10.0	fa0/0	192.168.10.1
Lab_A	192.168.20.0	s1/0	192.168.20.1
Lab_A	192.168.50.0	s1/1	192.168.50.1
Lab_B	192.168.30.0	fa0/0	192.168.30.1
Lab_B	192.168.20.0	s1/0	192.168.20.2
Lab_B	192.168.40.0	s1/1	192.168.40.1
Lab_C	192.168.60.0	fa0/0	192.168.60.1
Lab_C	192.168.40.0	s1/0	192.168.40.2
Lab_C	192.168.70.0	s1/1	192.168.70.1

A sample configuration is given as under

```
Router>en
Router#config t
Router(config)#hostname Lab_A
Lab_A(config)#interface fa0/0
Lab_A(config-if)#ip address 192.168.10.1 255.255.255.0
Lab_A(config-if)#description Lab_A LAN Connection
Lab_A(config-if)#no shut
Lab_A(config-if)#interface serial 1/0
Lab_A(config-if)#ip address 192.168.20.1 255.255.255.0
Lab_A(config-if)#description WAN Connection to Lab_B
Lab_A(config-if)#no shut
Lab_A(config-if)#interface serial 1/1
Lab_A(config-if)#ip address 192.168.50.1 255.255.255.0
Lab_A(config-if)#no shut
Lab_A(config-if)#exit
Lab_A(config)#banner motd #
This is the Lab_A router
#
Lab_A(config)#^z

Lab_A#copy running-config startup-config

Destination filename [startup-config]? [Enter]

Lab_A#
```

Before you jump in and configure a serial interface, there are a couple of things you need to know. First, the interface will usually be attached to a CSU/DSU type of device that provides clocking for the line to the router. But if you have a back-to-back configuration (for example, one that's used in a lab environment), one end—the data communication equipment (DCE) end of the cable—must provide clocking. By default, Cisco routers are all data terminal equipment (DTE) devices, so you must tell an interface to provide clocking if you need it to act like a DCE device.

To check the DCE interface, just bring your mouse over serial link, the interface with whose name

you see a 🕒 (clock symbol) is the DCE one. You configure a DCE serial interface with the clock rate command:

```
Lab_B(config)#interface serial 1/0
Lab_B(config-if)#clock rate ?
<300-4000000> Choose clockrate from list above
Router(config-if)#clock rate 64000
```

Notice that the clock rate command is in bits per second.

Configure the PCs and Switches too. Make sure all devices are communicating with each other (use **ping** to verify).

Now you must have noticed that routers can communicate with devices directly connected to them. PC0-PC2 and Switch0 can communicate with Lab_A router & in between themselves but can’t with Lab_B router and Switch1 & PC2-PC5 and vice versa.

Configure Static routing (see Lab 9) or Dynamic routing via RIP or OSPF (See Lab 10 or 11) so that all the nodes are communicating with each other.

4.1 Configuring Standard ACLs

Standard IP access lists filter network traffic by examining the source IP address in a packet. You create a standard IP access list by using the access-list numbers 1–99 Access-list types are generally differentiated using a number. Based on the number used when the access list is created, the router knows which type of syntax to expect as the list is entered. By using numbers 1–99, you’re telling the router that you want to create a standard IP access list, so the router will expect syntax specifying only the source IP address in the test lines.

Below is an example of the many access-list number ranges that you can use to filter traffic on your network (the protocols for which you can specify access lists depend on your IOS version):

```
Lab_A(config)#access-list ?
<1-99> IP standard access list
<100-199> IP extended access list
```

Let’s take a look at the syntax used when creating a standard access list:

```
Lab_A(config)#access-list 10 ?
deny Specify packets to reject
permit Specify packets to forward
remark Access list entry comment
```

By using the access-list numbers between 1–99, you’re telling the router that you want to create a standard IP access list. After you choose the access-list number, you need to decide whether you’re creating a permit or deny statement. For this example, you will create a deny statement:

```
Lab_A(config)#access-list 10 deny ?
A.B.C.D Address to match
any Any source host
host A single host address
```

The next step requires a more detailed explanation. There are three options available. You can use the any parameter to permit or deny any host or network; you can use an IP address to specify either a single host or a range of them; or you can use the host command to specify a specific host only. The any command is pretty obvious—any source address matches the statement, so every packet compared against this line will match. The host command is relatively simple. Here's an example using it:

```
Lab_A(config)#access-list 10 deny host 192.168.30.3
```

This tells the list to deny any packets from host 192.168.30.3. The default parameter is host. In other words, if you type **access-list 10 deny 192.168.30.3**, the router assumes you mean host 192.168.30.3. But there's another way to specify either a particular host or a range of hosts—you can use wildcard masking. In fact, to specify any range of hosts, you have to use wildcard masking in the access list.

4.1.1 Wildcard Masking

Wildcards are used with access lists to specify an individual host, a network, or a certain range of a network or networks. To understand a wildcard, you need to understand what a block size is; it's used to specify a range of addresses. Some of the different block sizes available are 64, 32, 16, 8, and 4.

When you need to specify a range of addresses, you choose the next-largest block size for your needs. For example, if you need to specify 34 networks, you need a block size of 64. If you want to specify 18 hosts, you need a block size of 32. If you only specify two networks, then a block size of 4 would work.

Wildcards are used with the host or network address to tell the router a range of available addresses to filter. To specify a host, the address would look like this:

192.168.30.3 0.0.0.0

The four zeros represent each octet of the address. Whenever a zero is present, it means that octet in the address must match exactly. To specify that an octet can be any value, the value of 255 is used. As an example, here's how a /24 subnet is specified with a wildcard:

192.168.30.0 0.0.0.255

This tells the router to match up the first three octets exactly, but the fourth octet can be any value.

4.1.2 A simple Example (Standard ACLs)

Consider you want to deny the whole **192.168.60.0** network and the host **192.168.30.3** to communicate with 192.168.10.0 network. You only need to use the following statements on

Lab_A router.

```
Lab_A(config)#access-list 10 deny 192.168.60.0 0.0.0.255
Lab_A(config)#access-list 10 deny 192.168.30.3
Lab_A(config)#access-list 10 permit any
```

The last statement is used to prevent any other packet drop as default action of access lists is **deny**.Now you have to choose the outbound interface (i.e. **FastEthernet 0/0**) which is directly connected to **192.168.10.0** network or inbound interface (i.e. **Serial 1/0**) from which the packet has to enter the router **Lab_A**. (Any one would suffice)

```
Lab_A(config)#interface serial 1/0
Lab_A(config-if)# ip access-group 10 in
```

Or

```
Lab_A(config)#interface fastEthernet 0/0
Lab_A(config-if)# ip access-group 10 out
```

Now check using Simulation mode to see the result of your access list in a better way!

4.2 Configuring Named ACLs

The named ACLs are not represented by a number but by a name that is given by the user. They behave the same way as standard or extended ACLs but have a name instead of a number.

```
Lab_A(config)#ip access-list ?
extended Extended Access List
standard Standard Access List
```

The difference is obvious; we have used **ip access-list** instead of **access-list.**

To do the task done in **4.1 (Standard ACLs)** using named ACLs, we use the following commands:

```
Lab_A(config)#ip access-list standard Part_4.1
Lab_A(config-std-nacl)#deny 192.168.60.0 0.0.0.255
Lab_A(config-std-nacl)#deny 192.168.30.3
Lab_A(config-std-nacl)#permit any
Lab_A(config)#interface serial 1/0
Lab_A(config-if)# ip access-group Part_4.1 in
```

Or

```
Lab_A(config)#interface fastEthernet 0/0
Lab_A(config-if)# ip access-group Part_4.1 out
```

To do the task done in 4.2 (Extended ACLs) using named ACLs, we use the following commands:

```
Lab_A(config)#ip access-list extended Part_4.2
Lab_A(config-ext-nacl)#permit ip 192.168.60.0 0.0.0.255 host 192.168.10.3
Lab_A(config-ext-nacl)#permit ip host 192.168.30.3 host 192.168.10.2
Lab_A(config-ext-nacl)#deny ip 192.168.60.0 0.0.0.255 any
Lab_A(config-ext-nacl)#deny ip host 192.168.30.3 any
Lab_A(config-ext-nacl)#permit ip any any
Lab_A(config)#interface serial 1/0
Lab_A(config-if)# ip access-group Part_4.2 in
```

Or

```
Lab_A(config)#interface fastEthernet 0/0
Lab_A(config-if)# ip access-group Part_4.2 out
```

Now check using Simulation mode to see the result of your access list in a better way!

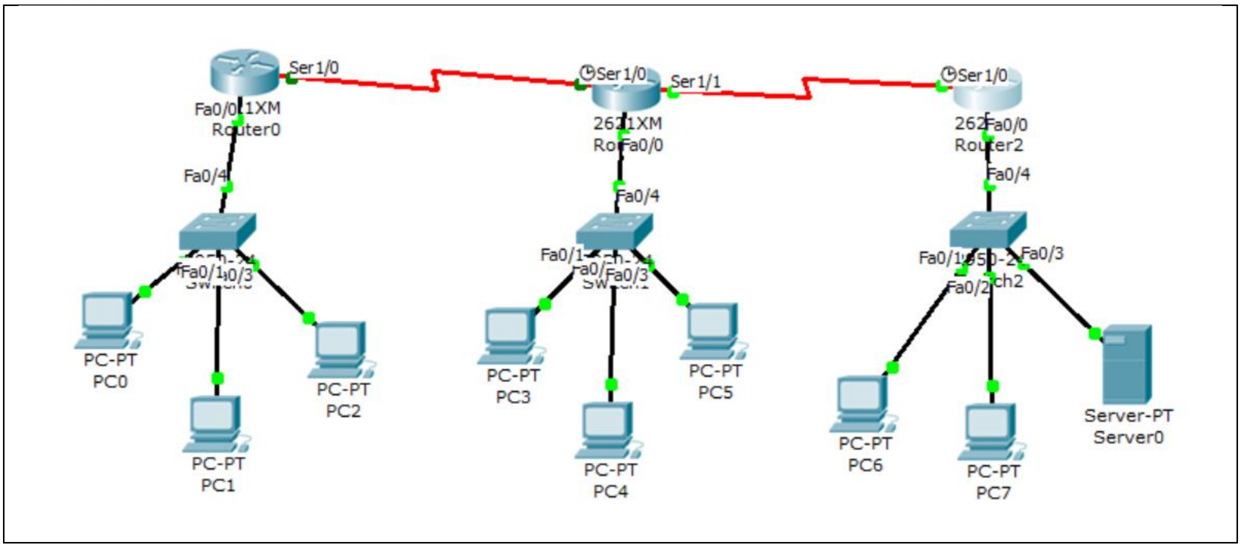
4.3 Verifying the Access List Configurations

Use the command show access-lists to see the access list configurations.

```
Lab_A#show access-lists
Standard IP access list 10
deny 192.168.60.0 0.0.0.255
deny host 192.168.30.3
permit any (12 match(es))
```

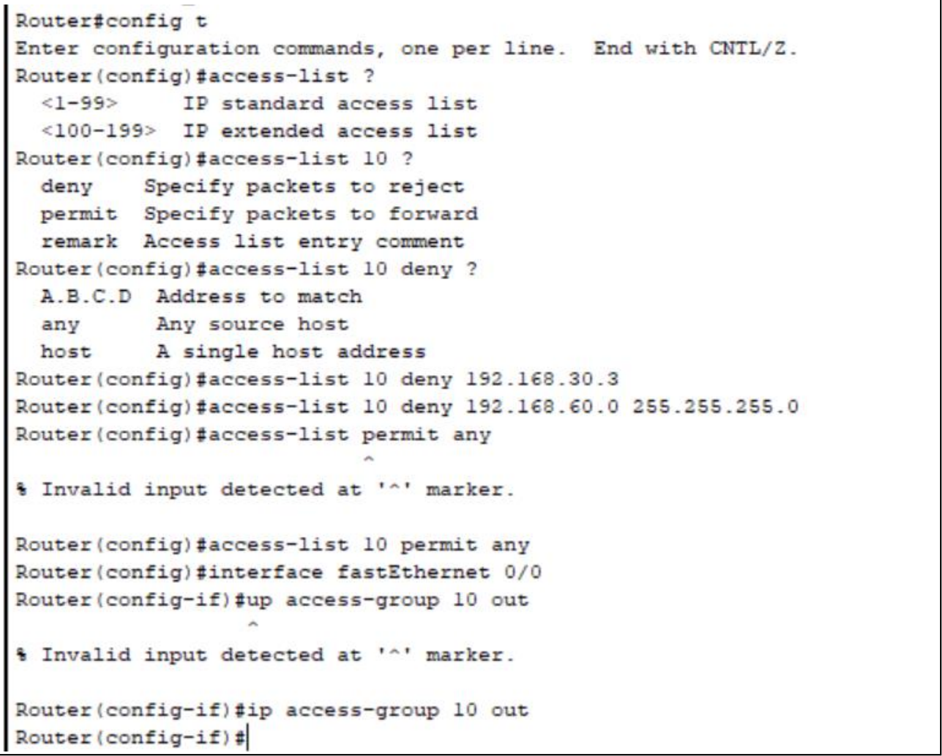
4.4 Student activity:

Use the link provided at the start of the manual to watch a detailed demo for how to implement Extended ACL’s and attach screenshots of your work below.

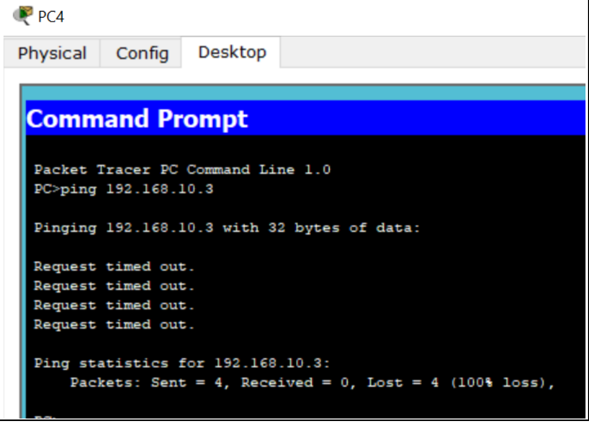


1. **Standard ACLs**

Router 0

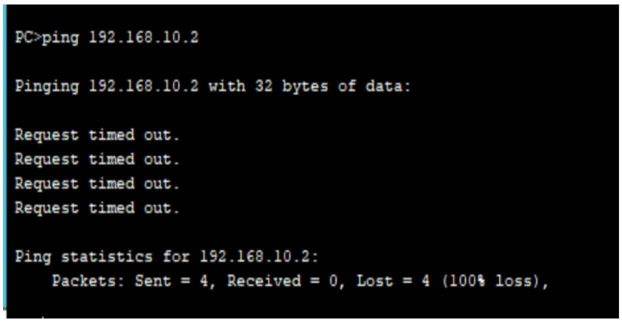

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#access-list ?
  <1-99>      IP standard access list
  <100-199>   IP extended access list
Router(config)#access-list 10 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
  remark  Access list entry comment
Router(config)#access-list 10 deny ?
  A.B.C.D  Address to match
  any      Any source host
  host     A single host address
Router(config)#access-list 10 deny 192.168.30.3
Router(config)#access-list 10 deny 192.168.60.0 255.255.255.0
Router(config)#access-list permit any
                           ^
% Invalid input detected at '^' marker.

Router(config)#access-list 10 permit any
Router(config)#interface fastEthernet 0/0
Router(config-if)#up access-group 10 out
                  ^
% Invalid input detected at '^' marker.

Router(config-if)#ip access-group 10 out
Router(config-if)#
```

Pinging network 10.0 by 192.168.30.3



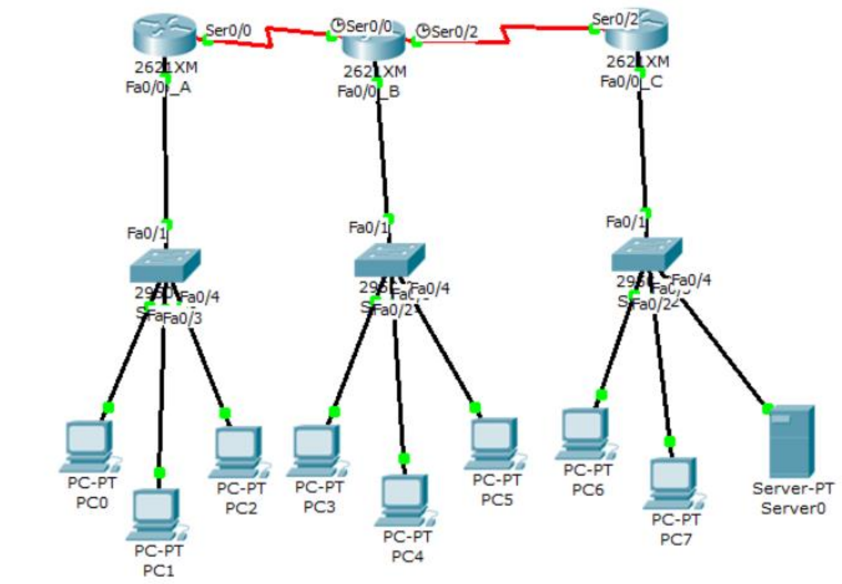
```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.10.3

Pinging 192.168.10.3 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.10.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Pinging network 10.0 by network 60.0



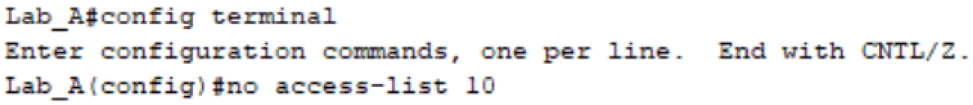
```
PC>ping 192.168.10.2

Pinging 192.168.10.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.10.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

2. Extended ACLs



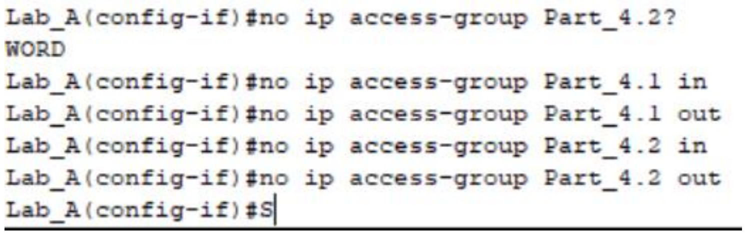
Existing from all the ACL's

By Number



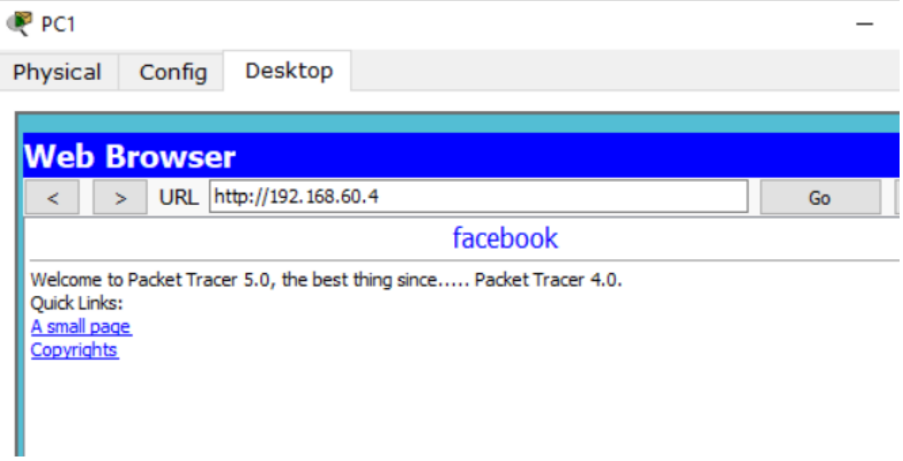
```
Lab_A#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Lab_A(config)#no access-list 10
```

By Name



```
Lab_A(config-if)#no ip access-group Part_4.2?
WORD
Lab_A(config-if)#no ip access-group Part_4.1 in
Lab_A(config-if)#no ip access-group Part_4.1 out
Lab_A(config-if)#no ip access-group Part_4.2 in
Lab_A(config-if)#no ip access-group Part_4.2 out
Lab_A(config-if)#S
```

Open Web Browser form PC

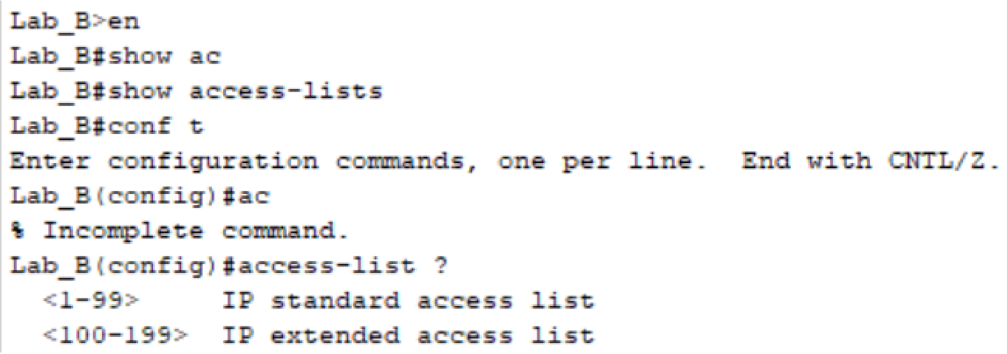


http://192.168.60.4

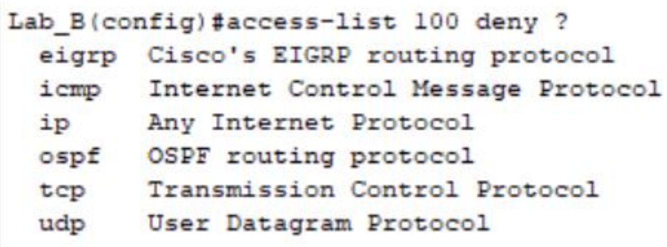
facebook

Welcome to Packet Tracer 5.0, the best thing since..... Packet Tracer 4.0.
Quick Links:
A small page
Copyrights

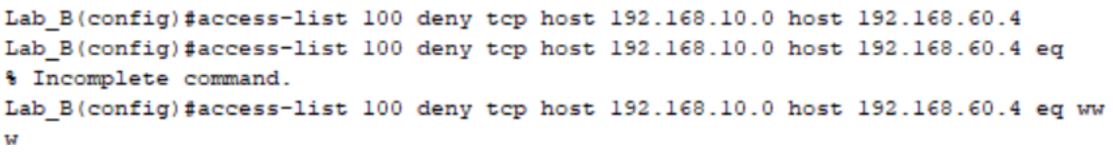
Using Extended ACL Denying the TCP from Server to NETWORK 10.0



```
Lab_B>en
Lab_B#show ac
Lab_B#show access-lists
Lab_B#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Lab_B(config)#ac
% Incomplete command.
Lab_B(config)#access-list ?
  <1-99>      IP standard access list
  <100-199>   IP extended access list
```

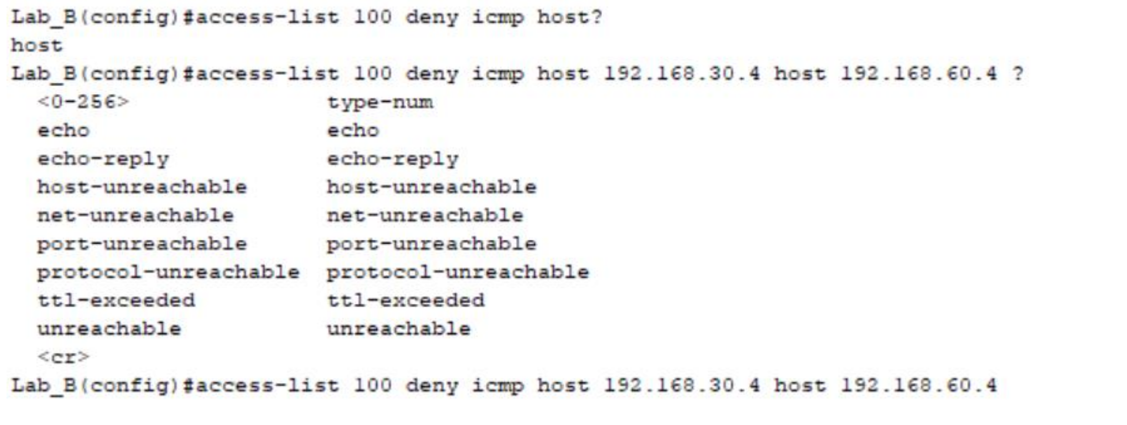


```
Lab_B(config)#access-list 100 deny ?
  eigrp   Cisco's EIGRP routing protocol
  icmp    Internet Control Message Protocol
  ip      Any Internet Protocol
  ospf    OSPF routing protocol
  tcp     Transmission Control Protocol
  udp     User Datagram Protocol
```

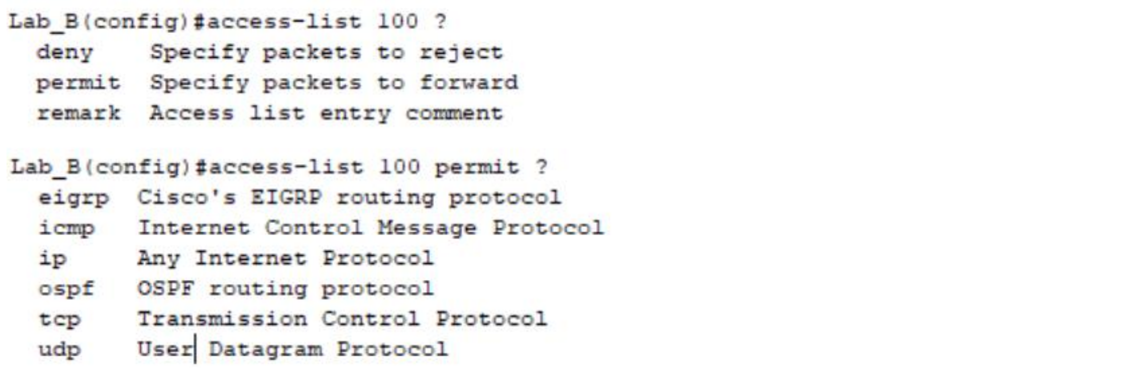


```
Lab_B(config)#access-list 100 deny tcp host 192.168.10.0 host 192.168.60.4
Lab_B(config)#access-list 100 deny tcp host 192.168.10.0 host 192.168.60.4 eq
% Incomplete command.
Lab_B(config)#access-list 100 deny tcp host 192.168.10.0 host 192.168.60.4 eq ww
w
```

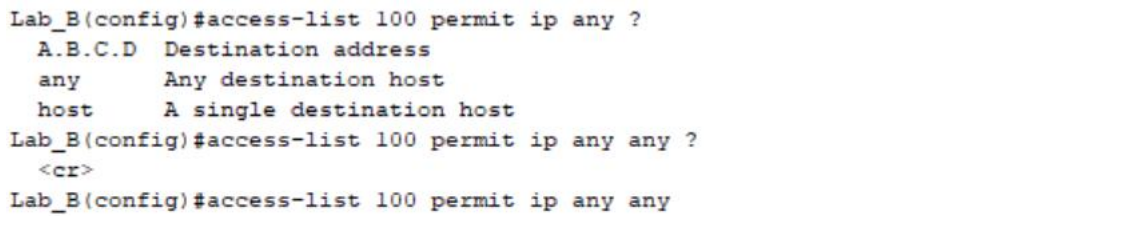
Using Extended ACL Denying the ICMP from Server to PC5



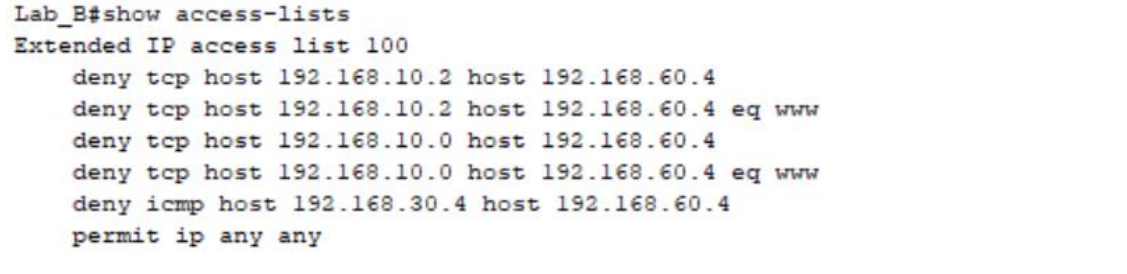
```
Lab_B(config)#access-list 100 deny icmp host?
host
Lab_B(config)#access-list 100 deny icmp host 192.168.30.4 host 192.168.60.4 ?
  <0-256>                type-num
  echo                   echo
  echo-reply             echo-reply
  host-unreachable       host-unreachable
  net-unreachable        net-unreachable
  port-unreachable       port-unreachable
  protocol-unreachable   protocol-unreachable
  ttl-exceeded           ttl-exceeded
  unreachable            unreachable
  <cr>
Lab_B(config)#access-list 100 deny icmp host 192.168.30.4 host 192.168.60.4
```



```
Lab_B(config)#access-list 100 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
  remark  Access list entry comment

Lab_B(config)#access-list 100 permit ?
  eigrp  Cisco's EIGRP routing protocol
  icmp   Internet Control Message Protocol
  ip     Any Internet Protocol
  ospf   OSPF routing protocol
  tcp    Transmission Control Protocol
  udp    User Datagram Protocol
```

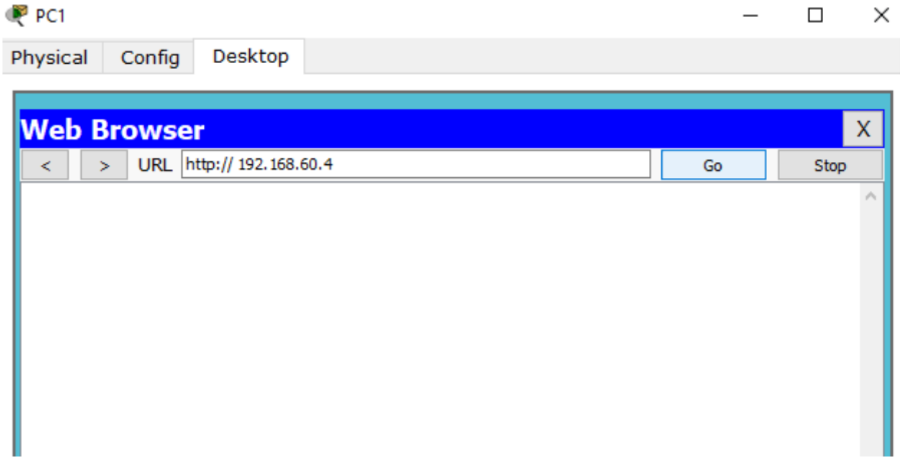


```
Lab_B(config)#access-list 100 permit ip any ?
  A.B.C.D  Destination address
  any      Any destination host
  host     A single destination host
Lab_B(config)#access-list 100 permit ip any any ?
  <cr>
Lab_B(config)#access-list 100 permit ip any any
```



```
Lab_B#show access-lists
Extended IP access list 100
    deny tcp host 192.168.10.2 host 192.168.60.4
    deny tcp host 192.168.10.2 host 192.168.60.4 eq www
    deny tcp host 192.168.10.0 host 192.168.60.4
    deny tcp host 192.168.10.0 host 192.168.60.4 eq www
    deny icmp host 192.168.30.4 host 192.168.60.4
    permit ip any any
```

Web is blocked, so unreachable


PC1
Physical
Config
Desktop
Web Browser
X
<
>
URL
http:// 192.168.60.4
Go
Stop