Swaraj Aditya Sahoo
2241019482
SL-62.

WEEK-END ASSIGNMENT-13

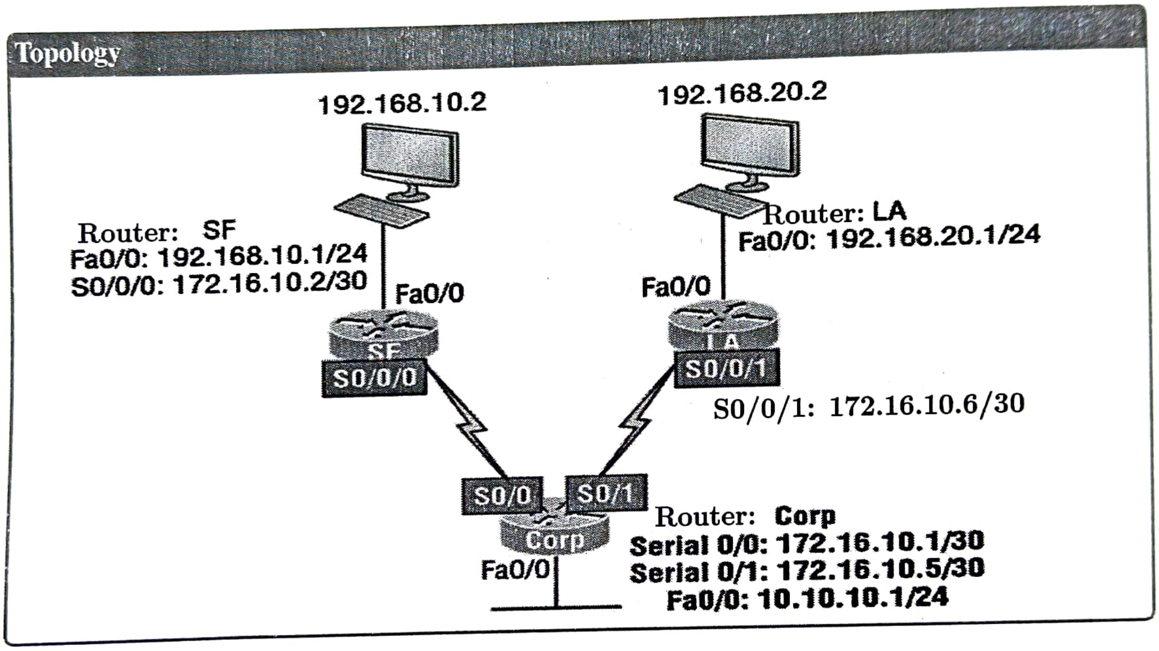
Computer Networking Workshop (CSE 4541)

Publish on: 21-05-2024
Course Outcome: CO_5
Program Outcome: PO_{4-5}
Submission on: 25-05-2024
Learning Level: L_{4-5}

Security: Access Control List

In this assignment, you will complete two exercises on access control lists, *Standard IP Access Lists* & *Extended IP Access Lists* to filter traffic as it either enters or leaves an interface using Cisco Packet Tracer (CPT).

Consider the following network topology and the IP distribution:



1. Configure the above network using CPT with the networks in different color frame.

Paste CPT configuration diagram and IOS commands for routing

2. Create and apply **standard access lists** to allow only packets from a single host, **192.168.10.2**, on the **SF** LAN to enter the **LA LAN**.

Standard ACL **Remark**

1. State IOS commands to create standard access list:

```
access-list 10 deny 192.168.10.0 0.0.0.255
access-list 10 permit any
```

2. Apply the access list at the interface:

```
interface fa 0/0
ip- access-group 10 out
```

3. Write the commands to verify the created access list:

```
Router(config)# show ip access-lists
```

4. Test the access list using ping command:

```
ping 192.168.10.2
```

5. If you have another host on the LA LAN, ping that address from SF LAN, which should fail if your ACL is working. (State Yes/No).

Yes

3. Write the wildcard mask to specify only host 192.168.20.2.

4. State the wildcard mask for the network address, **192.168.10.0**.

5. Lets say that you want to block access to the part of the network that ranges from 172.16.8.0 through 172.16.15.0. Write the standard access list at a router, **R1**, using standard access list 10.

Std acl Command: **Remark**

```
R1(config)# configure terminal
R1(config)# access-list 10 deny 172.168.8.0 0.0.7.255
access-list 10 permit any.
interface fa 0/0
R1(config-if)# access-group 10 in
```

6. Find the range of addresses the router, **Corp**, blocks as a result of the following access list.

Std acl Command: **Remark**

```
Corp(config)#access-list 10 deny 172.16.16.0 0.0.3.255
```

172.16.16.0 via 172.16.19.255

7. Find the range of addresses the router, **Corp**, permits as a result of the following access list.

Std acl Command: **Remark**

```
Corp(config)#access-list 10 permit 172.16.16.0 0.0.7.255
```

172.16.16.0 via 172.16.23.255

8. What do you think the range of this one is?

```
Corp(config)#access-list 10 deny 172.16.32.0 0.0.15.255.
```

Std acl Command: **Remark**

Range is 172.16.32.0 via 172.16.47.255

9. Determine the range of networks the router blocks for the given access list;

```
Router(config)#access-list 10 deny 172.16.64.0 0.0.63.255.
```

Std acl Command: **Remark**

Range :- 172.16.64.0 via 172.16.127.255

10. Write the IP and wildcard mask for the command **any**.

Std acl Command: **Remark**

It can use any ip.
wild card mask : 0.0.0.0 255.255.255.255

11. Determine the range of networks the router permits for the given access list;

```
R1(config)#access-list permit deny 192.168.160.0 0.0.31.255.
```

Std acl Command: **Remark**

192.168.160.0 via 192.168.191.255

12. In this exercise, you will use an **extended access list** to stop host 192.168.10.2 from creating a Telnet session to router **LA** (172.16.10.6). However, the host still should be able to ping the **LA** router. IP extended lists should be placed close to the source, so add the extended list on router **SF**.

Extended ACL **Remark**

1. State IOS commands to create extended access list:

```
SF(config)# ip access-list extended
            ACL-TELNET-STOP
SF(config-ext-nacl)# deny tcp host 192.168.10.2 any
SF(config-ext-nacl)# permit ip host 192.168.10.2 any
```

2. Apply the access list at the interface:

```
SF(config)# interface ethernet 0/0
SF(config-if)# ip access-group ACL-TELNET-STOP
                                          in
```

3. Write the commands to verify the created access list:

```
SF(config)# show ip access-lists
```

4. Test the access list using ping command:

```
ping 192.168.10.2
```

5. Test the access list using telnet command: *Try telnetting from host 192.168.10.2 to LA using the destination IP address of 172.16.10.6.*.This should fail, but the ping command should work. [Defend your answer]

```
telnet 172.16.10.6
ping 172.16.10.6
```

13. In this exercise, design the access list for the network shown at the first page using **named access list** to stop host 192.168.10.2 from creating a Telnet session to router **LA** (172.16.10.6). However, the host still should be able to ping the **LA** router. IP extended lists should be placed close to the source, so add the extended list on router **SF**.

Named ACL **Remark**

1. State IOS commands to create extended access list:

```
SF(config)# ip access-list extended no-telnet-to-LA
SF (config)# deny tcp host 192.168.10.2 any eq telnet
SF (config)# permit ip any any
```

2. Apply the access list at the interface:

```
SF(config) interface ethernet 0/0.
SF (config-if)# ip access-group no-telnet-to-LA in
```

3. Write the commands to verify the created access list:

```
SF(config)# show access-list extended-no-telnet-to-LA.
```

4. Test the access list using ping command:

```
Ping 172.16.10.6
```

5. Test the access list using telnet command: *Try telnetting from host 192.168.10.2 to LA using the destination IP address of 172.16.10.6.*.This should fail, but the ping command should work. [Defend your answer]

```
telnet 172.16.10.6
Ping 172.16.10.6
```