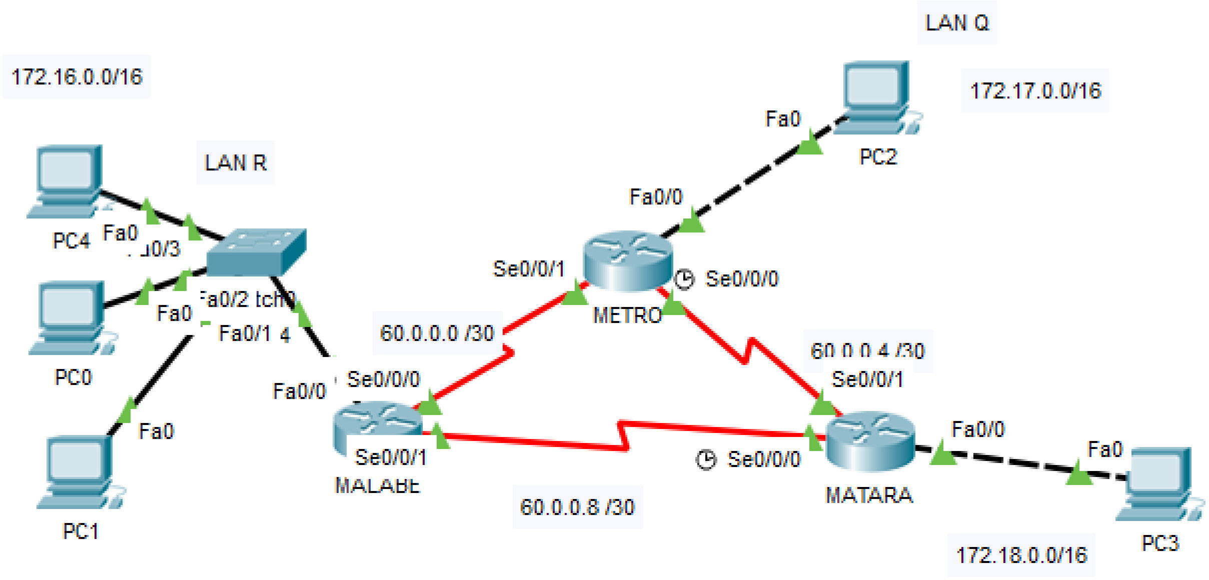
Access Control Lists (ACL) - Self Study Guide

Standard ACL



If all the routers are correctly configured in the given network, all the PCs should be able to communicate with each other. ACLs will be implemented inside the routers and can be used to restrict the communication between PCs. Standard ACLs can be used to simply block or allow a PC and Extended ACLs can be used to block communication destination wise or protocol wise.

Assigning IP addresses for PCs

Make sure all the PCs are configured with suitable IP addresses, subnet masks and default gateways.

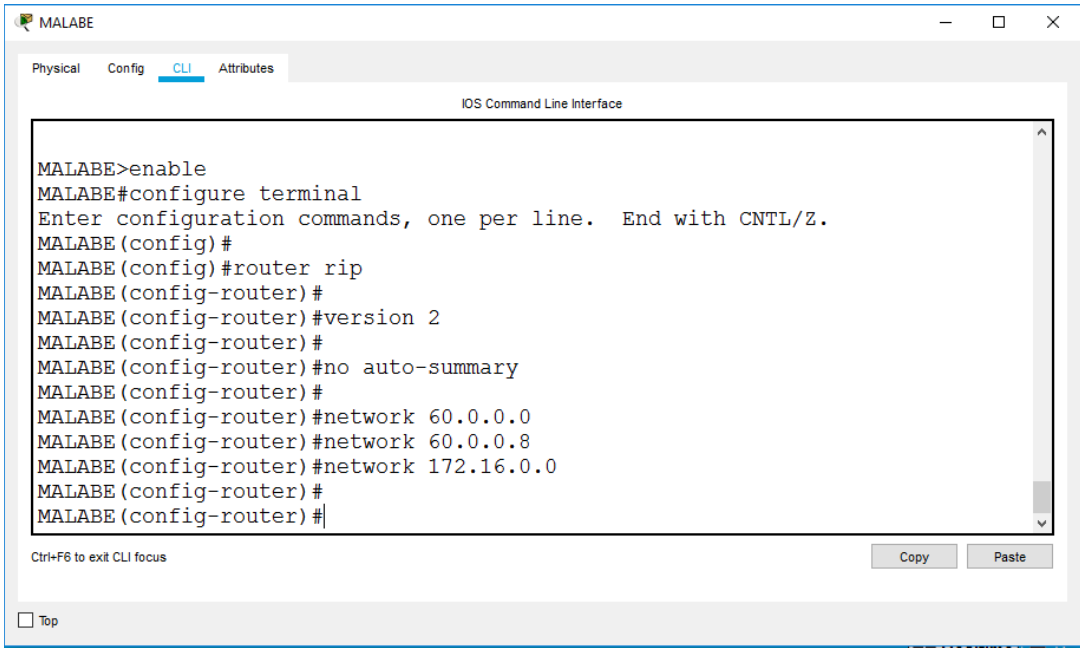
To assign an IP address to a PC,

Click on the PC -> Desktop Tab -> IP Configuration

Device	IP address	Subnet mask	Default gateway
PC0	172.16.0.10	255.255.0.0	172.16.0.1
PC1	172.16.0.20	255.255.0.0	172.16.0.1
PC2	172.17.0.10	255.255.0.0	172.17.0.1
PC3	172.18.0.10	255.255.0.0	172.18.0.1
PC4	172.16.0.30	255.255.0.0	172.16.0.1

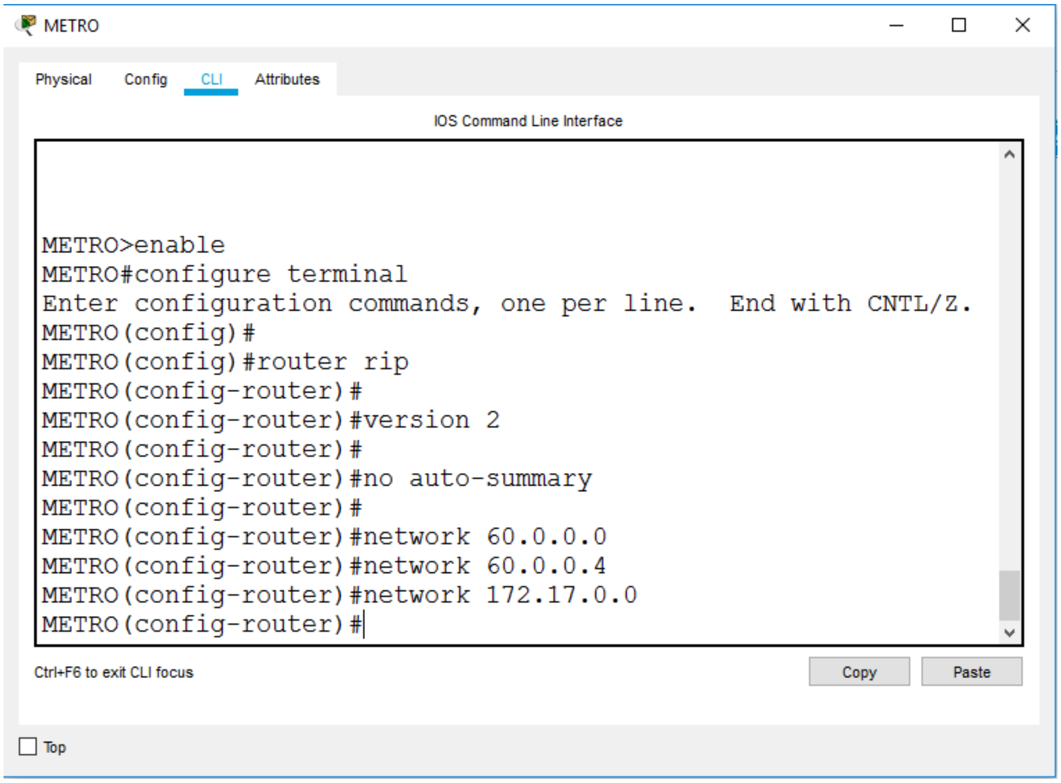
Implementing RIP Protocol

Malabe Router



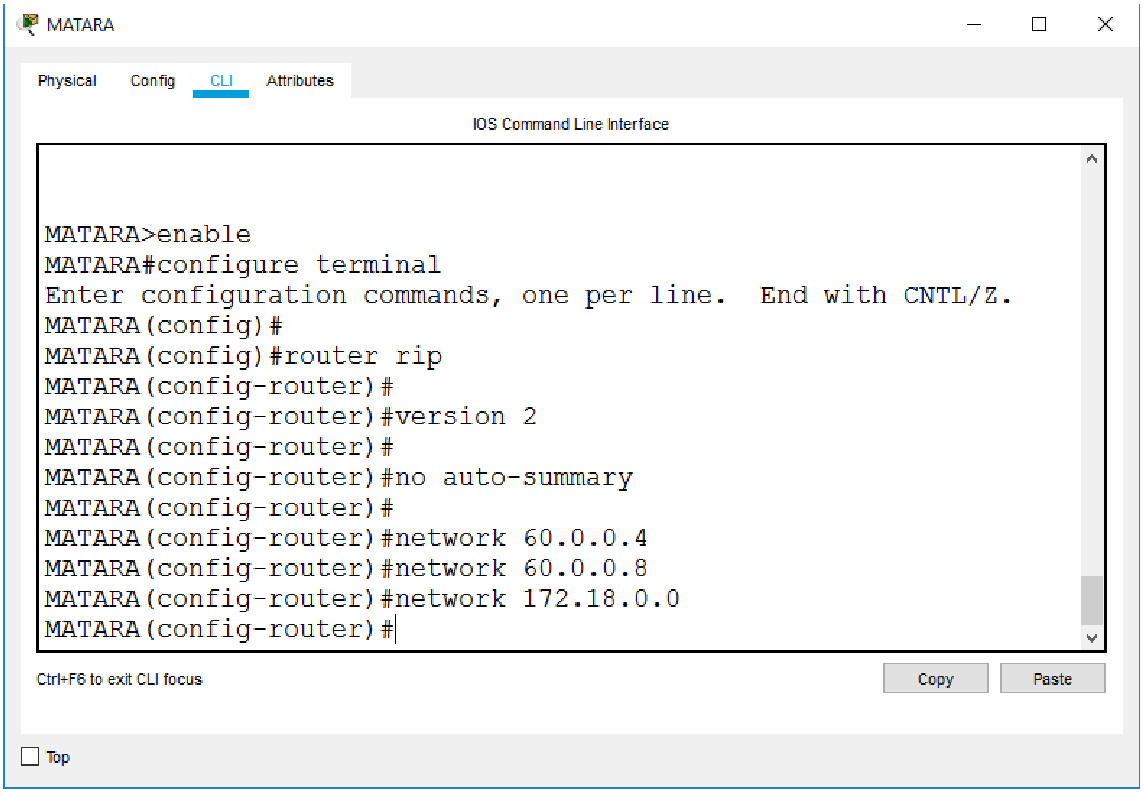
```
MALABE>enable
MALABE#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
MALABE(config)#
MALABE(config)#router rip
MALABE(config-router)#
MALABE(config-router)#version 2
MALABE(config-router)#
MALABE(config-router)#no auto-summary
MALABE(config-router)#
MALABE(config-router)#network 60.0.0.0
MALABE(config-router)#network 60.0.0.8
MALABE(config-router)#network 172.16.0.0
MALABE(config-router)#
MALABE(config-router)#
```

Metro Router



```
METRO>enable
METRO#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
METRO(config)#
METRO(config)#router rip
METRO(config-router)#
METRO(config-router)#version 2
METRO(config-router)#
METRO(config-router)#no auto-summary
METRO(config-router)#
METRO(config-router)#network 60.0.0.0
METRO(config-router)#network 60.0.0.4
METRO(config-router)#network 172.17.0.0
METRO(config-router)#
```

Matara Router



```
MATARA>enable
MATARA#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
MATARA(config)#
MATARA(config)#router rip
MATARA(config-router)#
MATARA(config-router)#version 2
MATARA(config-router)#
MATARA(config-router)#no auto-summary
MATARA(config-router)#
MATARA(config-router)#network 60.0.0.4
MATARA(config-router)#network 60.0.0.8
MATARA(config-router)#network 172.18.0.0
MATARA(config-router)#
```

After implementing RIP protocol in all 3 routers, PCs should be able to communicate with each other.

Applying Standard ACL

Standard ACLs can be used to allow or deny network traffic from a device which is going to pass through the specific router. When creating an ACL, a number between 1 and 99 have to be chosen for the ACL. Standard ACLs are applied to the router which is nearest to the destination. According to the lab sheet following rules should be applied,

- PC2 is allowed to access network 172.16.0.0
- PC3 is not allowed to access network 172.16.0.0
- Only PC4 of LAN_R can transmit data to LAN_Q

In first scenario, PC2 acts as the source and LAN_R acts as the destination. This ACL rule must be implemented in the MALABE router since it is the closest router to the destination.

```
MALABE >enable
MALABE #configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
MALABE(config)#access-list 1 permit 172.17.0.10
```

Above ACL rule simply allow data traffic from PC2(172.17.0.10) to pass through. This action is already possible in the network since RIP protocol is implemented but when ACLs are going to be implemented and assigned for an interface, all the conditions should be specified explicitly.

Note: An ACL implicitly denies all the traffic unless they are allowed explicitly.

Note: After creating an ACL, it should be applied for an interface and should be specified whether it for incoming or outgoing traffic.

Since PC2 can reach LAN_R via Fa0/0 interface of MALABE router, this created ACL will be applied to outgoing traffic from Fa0/0 interface. (Code available in next activity)

In second scenario PC3 should not be able to access LAN_R. Since destination is LAN_R, this rule also should be implemented in the Malabe router and should be applied to the outgoing traffic of Fa0/0. So ACL 1 will be added with another rule denying PC3 to communicate.

```
MALABE(config)#access-list 1 deny 172.18.0.10
```

To assign this ACL to Fa0/0 interface, following code segment have to be executed in Malabe router.

```
MALABE(config)#interface fa0/0
MALABE(config-if)#ip access-group 1 out
```

At the end try to send a PDU between PC2 and PC3. Will it work? Why?

(Hint: Implicit deny, ‘access-list 1 permit any’)

In the third scenario only PC4 from LAN_R should be able to communicate with LAN_Q. In this scenario since LAN_Q is the destination, ACL should be implemented in the Metro router. After allowing traffic from PC4 to pass through router, to deny traffic from other PCs, individual entries can be added to the ACL. But in case a new PC joins the network ACL also should be updated. To avoid such hassle simply the whole network can be blocked using an ACL rule.

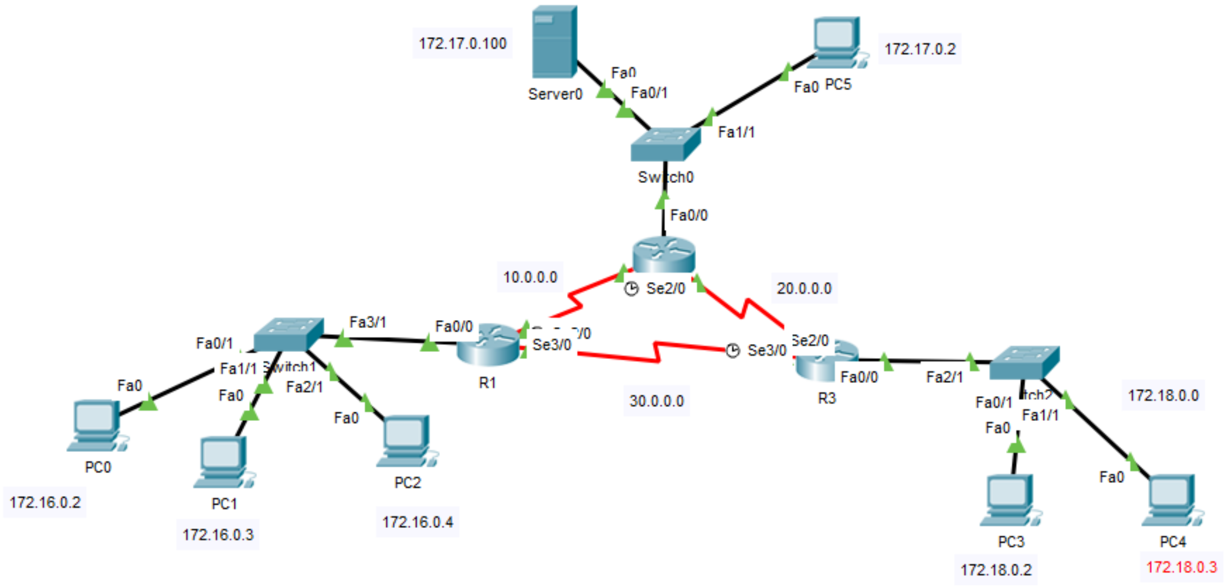
```
METRO(config)#access-list 1 permit 172.16.0.30
METRO (config)#access-list 1 deny 172.16.0.0 0.0.255.255
METRO (config)#interface fa0/0
METRO (config-if)#ip access-group 1 in
```

Note: Using same ACL number on the both routers is not going to make confusions because simply they reside in two separate routers. But in a single router each ACL should have a unique number.

To block a whole network a **wild card mask** is used along with the network address. A wild card mask is simply the outcome of subnet mask of the network (255.255.0.0) deducted by 255.255.255.255. As a result, any IP address which starts with 172.16 will be denied regardless the latter part of the IP address.

Note: The order of the ACL rules matters. In the above example if the 2 rules are swapped, traffic from PC4 also will be denied because router will check the ACL rules one by one and find a match in the first rule stating to block all the traffic from IP addresses which starts with 172.16.

<u>Extended ACL</u>



Testing services on the server

In the given network the server serves a webpage which is accessible via any PC in the given network. To access the webpage through a PC,

Click on PC -> Desktop Tab -> Web Browser -> Enter the Server IP address as URL

Accessing this webpage from any PC is possible since there are no ACLs currently in the network and this ensures that every PC have access to the HTTP service in the webserver.

To check the accessibility of FTP service by logging in to FTP server,

Click on PC -> Desktop Tab -> Command Prompt

Then enter the command ‘ftp’ followed by the IP address of the server which is 172.17.0.100

```
C:\> ftp 172.17.0.100
```

use the user name and password as **cisco** when asked

to list down all the files available in the ftp server used the command: **dir**

to download a file, use the command get followed by the file name: **get myFile.txt**

Applying Extended ACL

Extended ACLs should be implemented in the router which is closest to the source. The main advantage of extended ACLs over the standard ACLs is that they can be used to restrict communication based on the protocol and the destination. Any number between 100-199 can be chosen as extended ACL number. In the lab sheet following rules have to implemented using extended ACLs.

1. **PC1 is not allowed to access the web service in Server1.**
2. **PC2 is not allowed to access the ftp service in Server1.**
3. **PC3 is not allowed to access the ftp service in Server1.**
4. **Any other traffic will be allowed.**

Since PC1 and PC2 are available in the same network both first and second rules should be implemented in the router R1 since it is the closest to the source. In the router R1 following commands should be executed,

```
R1(config)#access-list 110 deny tcp host 172.16.0.3 host 172.17.0.100 eq 80
R1(config)#access-list 110 deny tcp host 172.16.0.4 host 172.17.0.100 eq ftp
R1(config)#access-list 110 permit ip any any
R1(config)#interface FastEthernet0/0
R1(config-if)#ip access-group 110 in
```

In the first line the **first IP address** after the keyword host is the **IP address of the source**. The **second IP address** is the **IP address of destination** which is the IP address of the server. At the very end of the line port number is specified. Port number 80 represents HTTP. In here both the name of the port or the port number can be used.

Since PC3 is available in the R3 router, to implement third rule, following commands should be executed in R3 to create the extended ACL with the rules and to apply the ACL to the relevant interface.

```
R3(config)#access-list 110 deny tcp host 172.18.0.2 host 172.17.0.100 eq ftp
R3(config)#access-list 110 permit ip any any
R3(config)#interface FastEthernet0/0
R3(config-if)#ip access-group 110 in
```

Since ACLs have implicit deny by default, after implementing the given rules, all other traffic should be allowed. As the answer for the fourth scenario, following entries were added to the ACLs along with other rules.

```
R1(config)#access-list 110 permit ip any any
R3(config)#access-list 110 permit ip any any
```

For more information:

Standard ACLs: https://www.cbtnuggets.com/blog/certifications/cisco/networking-basics-how-to-configure-standard-acls-on-cisco-routers

Extended ACLs: https://www.cbtnuggets.com/blog/certifications/cisco/networking-basics-configuring-extended-access-lists-on-cisco-routers

Wildcard Mask: https://www.cbtnuggets.com/blog/technology/networking/networking-basics-what-are-wildcard-masks-and-how-do-they-work