**Configure Extended Access Lists**

An extended ACL gives you much more power than just a standard ACL. Extended IP ACLs check both the source and destination packet addresses. They can also check for specific protocols, port numbers, and other parameters, which allow administrators more flexibility and control.

**access-list access-list-number {permit | deny}**

**protocol source source-wildcard [operator port]**

**destination destination-wildcard [operator port]**

**[established] [log]**

|  |  |
| --- | --- |
| **Command Parameters** | **Descriptions** |
| ***access-list*** | Main command |
| ***access-list-number*** | Identifies the list using a number in the ranges of 100–199 or 2000– 2699. |
| ***permit | deny*** | Indicates whether this entry allows or blocks the specified address. |
| ***protocol*** | IP, TCP, UDP, ICMP, GRE, or IGRP. |
| ***source and destination*** | Identifies source and destination IP addresses. |
| ***source-wildcard and destination-wildcard*** | The operator can be lt (less than), gt (greater than), eq (equal to), or neq (not equal to). The port number referenced can be either the source port or the destination port, depending on where in the ACL the port number is configured. As an alternative to the port number, well-known application names can be used, such as Telnet, FTP, and SMTP. |
| ***established*** | For inbound TCP only. Allows TCP traffic to pass if the packet is a response to an outbound-initiated session. This type of traffic has the acknowledgement (ACK) bits set. (See the Extended ACL with the Established Parameter example.) |
| ***log*** | Sends a logging message to the console. |

**Before we configure Extended Access list you should cram up some important port number**

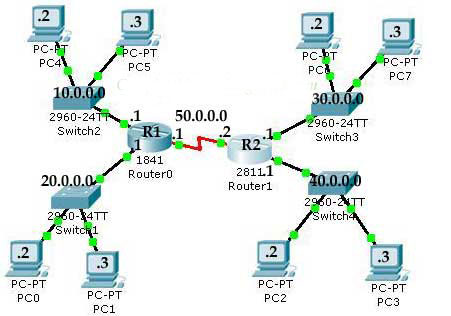
**Well-Known Port Numbers and IP Protocols**

|  |  |
| --- | --- |
| Port Number | IP Protocol |
| 20 (TCP) | FTP data |
| 21 (TCP) | FTP control |
| 23 (TCP) | Telnet |
| 25 (TCP) | Simple Mail Transfer Protocol (SMTP) |
| 53 (TCP/UDP) | Domain Name System (DNS) |
| 69 (UDP) | TFTP |
| 80 (TCP) | HTTP |

In this article we will configure Extended access list. If you want to read the feature and characteristic of access list reads this previous article.

[Access control list](http://computernetworkingnotes.com/ccna_certifications/access_control_list.htm)

In this article we will use a RIP running topology. Which we created in RIP routing practical.

[](http://computernetworkingnotes.com/ccna_certifications/rip_routing.htm)

**Three basic steps to configure Extended Access List**

* Use the access-list global configuration command to create an entry in a Extended ACL.
* Use the interface configuration command to select an interface to which to apply the ACL.
* Use the ip access-group interface configuration command to activate the existing ACL on an interface.

With Access Lists you will have a variety of uses for the wild card masks, but typically For CCNA exam prospective you should be able to do following:

1. ***Block host to host***
2. ***Block host to network***
3. ***Block Network to network***
4. ***Block telnet access for critical resources of company***
5. ***Limited ftp access for user***
6. ***Stop exploring of private network from ping***
7. ***Limited web access***
8. ***Configure established keyword***

**Block host to host**

**Task**

*You are the network administrator at* ***ComputerNetworkingNotes.com.*** *Your company hire a new employee and give him a pc 10.0.0.3. your company's critical record remain in 40.0.0.3. so you are asked to block the access of 40.0.0.3 from 10.0.0.3. while 10.0.0.3 must be able connect with other computers of network to perfom his task.*

**Decide where to apply ACL and in which directions.**

As we are configuring Extended access list. With extended access list we can filter the packed as soon as it genrate. So we will place our access list on F0/0 of Router1841 the nearest port of 10.0.0.3

To configure Router1841 (Hostname R1) double click on it and select CLI

***R1>enable***

***R1#configure terminal***

***Enter configuration commands, one per line. End with CNTL/Z.***

***R1(config)#access-list 101 deny ip host 10.0.0.3 40.0.0.3 0.0.0.0***

***R1(config)#access-list 101 permit ip any any***

***R1(config)#interface fastEthernet 0/0***

***R1(config-if)#ip access-group 101 out***

***R1(config-if)#exit***

***R1(config)#***

Verify by doing ping from 10.0.0.3 to 40.0.0.3. It should be reqest time out. Also ping other computers of network including 40.0.0.2. ping shuld be sucessfully.

**Block host to network**

**Task**

*Now we will block the 10.0.0.3 from gaining access on the network 40.0.0.0. ( if you are doing this practical after configuring pervious example don't forget to remove the last access list 101. With no access-list command. Or just close the packet tracer without saving and reopen it to be continue with this example.)*

***R1(config)#access-list 102 deny ip host 10.0.0.3 40.0.0.0 0.255.255.255***

***R1(config)#access-list 102 permit ip any any***

***R1(config)#interface fastEthernet 0/0***

***R1(config-if)#ip access-group 102 out***

***R1(config-if)#exit***

***R1(config)#***

Verify by doing ping from 10.0.0.3 to 40.0.0.3. and 40.0.0.2.It should be reqest time out. Also ping computers of other network. ping shuld be sucessfully.

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

***R2>enable***

***Enter configuration commands, one per line. End with CNTL/Z.***

***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 replay.

**Network to Network Access List**

**Task**

*Student’s lab is configured on the network of 10.0.0.0. While management's system remain in the network of 40.0.0.0. You are asked to stop the lab system from gaining access in management systems*

Now we will block the network of 10.0.0.0 from gaining access on the network 40.0.0.0. ( if you are doing this practical after configuring pervious example don't forget to remove the last access list 101. With no access-list command. Or just close the packet tracer without saving and reopen it to be continue with this example.)

***R1(config)#access-list 103 deny ip 10.0.0.0 0.255.255.255 40.0.0.0 0.255.255.255***

***R1(config)#access-list 103 permit ip any any***

***R1(config)#interface fastethernet 0/0***

***R1(config-if)#ip access-group 103 in***

***R1(config-if)#exit***

***R1(config)#***

Verify by doing ping from 10.0.0.3 and 10.0.0.2 to 40.0.0.3. and 40.0.0.2.It should be reqest time out. Also ping computers of other network. ping shuld be sucessfully.

**Network to host**

**Task**

For the final scenario you will block all traffic to 40.0.0.3 from the Network of 10.0.0.0 To accomplish this write an extended access list. The access list should look something like the following.

***R1(config)#interface fastethernet 0/0***

***R1(config-if)#no ip access-group 103 in***

***R1(config-if)#exit***

***R1(config)#no access-list 103 deny ip 10.0.0.0 0.255.255.255 40.0.0.0 0.255.255.255***

***R1(config)#access-list 104 deny ip 10.0.0.0 0.255.255.255 40.0.0.3 0.0.0.0***

***R1(config)#access-list 104 permit ip any any***

***R1(config)#interface fastethernet 0/0***

***R1(config-if)#ip access-group 104 in***

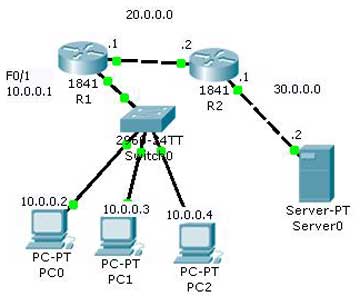
***R1(config-if)#exit***

***R1(config)#***

Verify by doing ping from 10.0.0.3 and 10.0.0.2 to 40.0.0.3.It should be reqest time out. Also ping computers of other network. ping shuld be sucessfully.

**Application based Extended Access list**

In pervoius example we filter ip base traffic. Now we will filter applicaion base traffic. To do this practical either create a topology as shown in figure and enable telnet and http and ftp service on server or download this pre configured topology and load it in packet tracer.

[Extended Access list  
[](http://computernetworkingnotes.com/ccna_certifications/download/extended_acl.pkt)](http://computernetworkingnotes.com/ccna_certifications/download/extended_acl.pkt)

**The established keyword**

The **established** keyword is a advanced feature that will allow traffic through only if it sees that a TCP session is already established. A TCP session is considered established if the three-way handshake is initiated first. This keyword is added only to the end of extended ACLs that are filtering TCP traffic.  
You can use TCP established to deny all traffic into your network except for incoming traffic that was first initiated from inside your network. This is commonly used to block all originating traffic from the Internet into a company's network except for Internet traffic that was first initiated from users inside the company. The following configuration would accomplish this for all TCP-based traffic coming in to interface serial 0/0/0 on the router:

***R1(config)#access-list 101 permit tcp any any established***

***R1(config)#interface serial 0/0/0***

***R1(config-if)#ip access-group 101 in***

***R1(config-if)#exit***

Although the access list is using a permit statement, all traffic is denied unless it is first established from the inside network. If the router sees that the three-way TCP handshake is successful, it will then begin to allow traffic through.

To test this access list double click on any pc from the network 10.0.0.0 and select web brower. Now give the ip of 30.0.0.2 web server. It should get sucessfully access the web page. Now go 30.0.0.2 and open command prompt. And do ping to 10.0.0.2 or any pc from the network the 10.0.0.0. it will request time out.

**Stop ping but can access web server**

We host our web server on 30.0.0.2. But we do not want to allow external user to ping our server as it could be used as denial of services. Create an access list that will filter all ping requests inbound on the serial 0/0/0 interface of router2.

***R2(config)#access-list 102 deny icmp any any echo***

***R2(config)#access-list 102 permit ip any any***

***R2(config)#interface serial 0/0/0***

***R2(config-if)#ip access-group 102 in***

To test this access list ping from 10.0.0.2 to 30.0.0.2 it should be request time out. Now open the web browser and access 30.0.0.2 it should be successfully retrieve

**Grant FTP access to limited user**

You want to grant ftp access only to 10.0.0.2. no other user need to provide ftp access on server. So you want to create a list to prevent FTP traffic that originates from the subnet 10.0.0.0/8, going to the 30.0.0.2 server, from traveling in on Ethernet interface E0/1 on R1.

***R1(config)#access-list 103 permit tcp host 10.0.0.2 30.0.0.2 0.0.0.0 eq 20***

***R1(config)#access-list 103 permit tcp host 10.0.0.2 30.0.0.2 0.0.0.0 eq 21***

***R1(config)#access-list 103 deny tcp any any eq 20***

***R1(config)#access-list 103 deny tcp any any eq 21***

***R1(config)#access-list 103 permit ip any any***

***R1(config)#interface fastethernet 0/1***

***R1(config-if)#ip access-group 103 in***

***R1(config-if)#exit***

**Grant Telnet access to limited user**

For security purpose you don’t want to provide telnet access on server despite your own system. Your system is 10.0.0.4. create a extended access list to prevent telnet traffic that originates from the subnet of 10.0.0.0 to server.

***R1(config)#access-list 104 permit tcp host 10.0.0.4 30.0.0.2 0.0.0.0 eq 23***

***R1(config)#access-list 104 deny tcp 10.0.0.0 0.255.255.255 30.0.0.2 0.0.0.0 eq 23***

***R1(config)#access-list 104 permit ip any any***

***R1(config)#interface fast 0/1***

***R1(config-if)#ip access-group 104 in***

***R1(config-if)#exit***