**Laboratory 6 – SECIRITY, NAT, PAT**

Student name: Phan Phuong Duy. Student ID: ITITIU16010

***Lab 6.1: Access Control Lists***

**Topology Diagram**



**Addressing Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Device** | **Interface** | **IP Address** | **Subnet Mask** | **Default Gateway** |
| R1 | Fa0/0 | 192.168.10.1 | 255.255.255.0 | N/A |
|  | Fa0/1 | 192.168.11.1 | 255.255.255.0 | N/A |
|  | S0/0/0 | 10.1.1.1 | 255.255.255.252 | N/A |
| R2 | Fa0/0 | 192.168.30.1 | 255.255.255.0 | N/A |
|  | Fa0/1 | 192.168.20.1 | 255.255.255.0 | N/A |
|  | S0/0/0 | 10.1.1.2 | 255.255.255.252 | N/A |
|  | Lo0 | 209.165.200.225 | 255.255.255.224 | N/A |
| PC1 | NIC | 192.168.10.10 | 255.255.255.0 | 192.168.10.1 |
| PC2 | NIC | 192.168.11.10 | 255.255.255.0 | 192.168.11.1 |
| PC3 | NIC | 192.168.30.10 | 255.255.255.0 | 192.168.30.1 |
| Server | NIC | 192.168.20.254 | 255.255.255.0 | 192.168.20.1 |

**Learning Objectives**

Upon completion of this lab, you will be able to:

• Design named standard and named extended ACLs.

• Apply named standard and named extended ACLs.

• Test named standard and named extended ACLs.

• Troubleshoot named standard and named extended ACLs.

**Scenario**

In this lab, you will learn how to configure basic network security using Access Control Lists. You will apply both standard and extended ACLs.

**Task 1: Prepare the Network**

**Step 1: Cable a network that is similar to the one in the topology diagram.**

You can use any current router in your lab as long as it has the required interfaces shown in the topology diagram.

**Step 2: Clear any existing configurations on the routers.**

**Task 2: Perform Basic Router Configurations**

Configure the R1, R2 routers according to the following steps:

1. Configure the router hostname to match the topology diagram.
2. Disable DNS lookup.
3. Configure an EXEC mode password of class.
4. Configure a message-of-the-day banner.
5. Configure a password of cisco for console connections.
6. Configure a password for VTY connections.
7. Configure IP addresses and masks on all devices.
8. Enable OSPF area 0 with a process ID of 1 on all routers for all networks.
9. Configure a loopback interface on R2 to simulate the ISP.
10. Verify full IP connectivity using the ping command. If ping isn’t successful between PCs, troubleshooting the problem.

Paste the output of the **show running-config**, **show ip route** commands on both routers here.

**Router 1:**

Router#show running-config

Building configuration...

Current configuration : 969 bytes

!

version 15.1

no service timestamps log datetime msec

no service timestamps debug datetime msec

no service password-encryption

!

hostname Router

!

!

!

!

!

!

!

!

no ip cef

no ipv6 cef

!

!

!

!

license udi pid CISCO2901/K9 sn FTX1524AT8D

!

!

!

!

!

!

!

!

!

!

!

spanning-tree mode pvst

!

!

!

!

!

!

interface GigabitEthernet0/0

ip address 192.168.10.1 255.255.255.0

duplex auto

speed auto

!

interface GigabitEthernet0/1

ip address 192.168.11.1 255.255.255.0

duplex auto

speed auto

!

interface Serial0/3/0

ip address 10.1.1.1 255.255.255.252

clock rate 2000000

!

interface Serial0/3/1

no ip address

clock rate 2000000

shutdown

!

interface Vlan1

no ip address

shutdown

!

router ospf 1

log-adjacency-changes

network 192.168.10.0 0.0.0.255 area 0

network 192.168.11.0 0.0.0.255 area 0

network 10.1.1.0 0.0.0.3 area 0

default-information originate

!

ip classless

!

ip flow-export version 9

!

!

!

!

!

!

!

line con 0

!

line aux 0

!

line vty 0 4

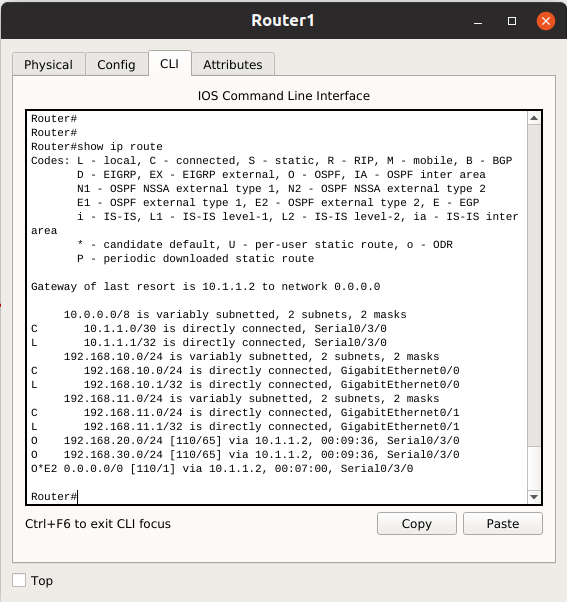
login

!

!

!

end



**Router 2:**

Router#show running-config

Building configuration...

Current configuration : 1051 bytes

!

version 15.1

no service timestamps log datetime msec

no service timestamps debug datetime msec

no service password-encryption

!

hostname Router

!

!

!

!

!

!

!

!

no ip cef

no ipv6 cef

!

!

!

!

license udi pid CISCO2901/K9 sn FTX1524P7JY

!

!

!

!

!

!

!

!

!

!

!

spanning-tree mode pvst

!

!

!

!

!

!

interface Loopback0

ip address 209.165.200.225 255.255.255.224

!

interface GigabitEthernet0/0

ip address 192.168.30.1 255.255.255.0

duplex auto

speed auto

!

interface GigabitEthernet0/1

ip address 192.168.20.1 255.255.255.0

duplex auto

speed auto

!

interface Serial0/3/0

ip address 10.1.1.2 255.255.255.252

!

interface Serial0/3/1

no ip address

clock rate 2000000

shutdown

!

interface Vlan1

no ip address

shutdown

!

router ospf 1

log-adjacency-changes

network 192.168.30.0 0.0.0.255 area 0

network 192.168.20.0 0.0.0.255 area 0

network 10.1.1.0 0.0.0.3 area 0

default-information originate

!

ip classless

ip route 0.0.0.0 0.0.0.0 Loopback0

!

ip flow-export version 9

!

!

!

!

!

!

!

line con 0

!

line aux 0

!

line vty 0 4

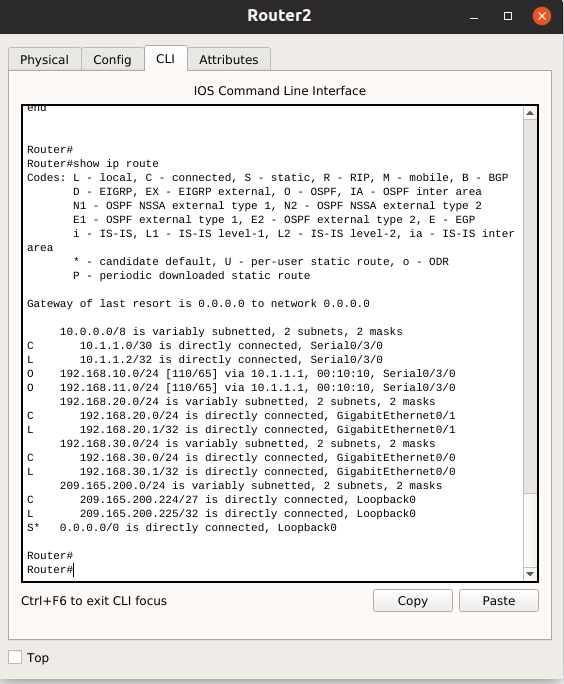
login

!

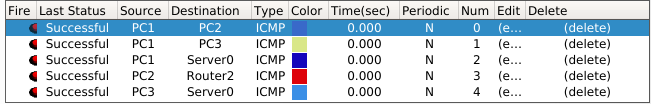
!

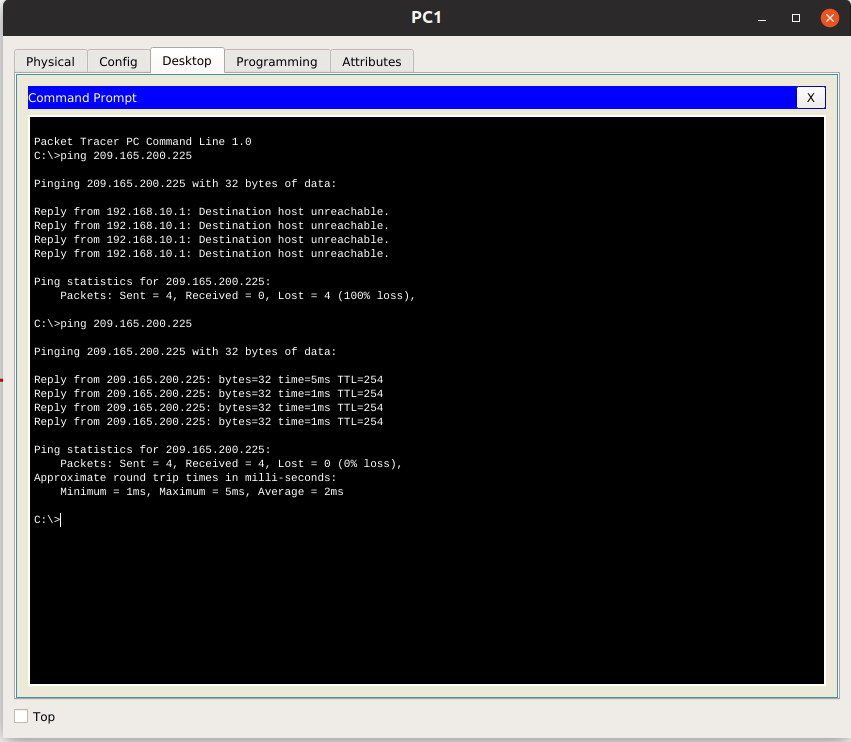
!

end



Test the connection between the PCs using ping command.





**Task 3: Configuring a Standard ACL**

Standard ACLs can filter traffic based on source IP address only. A typical best practice is to configure a standard ACL as close to the destination as possible. In this task, you are configuring a standard ACL. The ACL is designed to block traffic from the 192.168.11.0/24 network from accessing any local networks on R2.

This ACL will be applied outbound on the R2 FastEthernet 0/0 interface. Remember that every ACL has an implicit “deny all” that causes all traffic that has not matched a statement in the ACL to be blocked. For this reason, add the permit any statement to the end of the ACL.

Before configuring and applying this ACL, be sure to test connectivity from PC2 (or the Fa0/1 interface on R1) to PC3 (or the Fa0/0 interface on R2). Connectivity tests should be successful before applying the ACL.

**Step 1: Create the ACL on router R2.**

In global configuration mode, create a standard named ACL called **STND-1**.

R2(config)#ip access-list standard STND-1

In standard ACL configuration mode, add a statement that denies any packets with a source address of 192.168.11.0/24 and prints a message to the console for each matched packet.

**R2(config-std-nacl)#deny 192.168.11.0 0.0.0.255 log**

Permit all other traffic.

**R2(config-std-nacl)#permit any**

**Step 2: Apply the ACL.**

Apply the ACL STND-1 as a filter on packets entering R2 through FastEthernet interface 0/0.

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

**R2(config-if)#ip access-group STND-1 out**

**R2 config-if)#end**

**R2#copy run start**

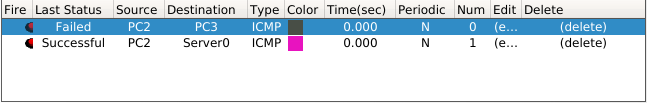
**Step 3: Test the ACL.**

Before testing the ACL, make sure that the console of R2 is visible. This will allow you to see the access list log messages when the packet is denied.

Test the ACL by pinging from PC2 to PC3. Since the ACL is designed to block traffic with source addresses from the 192.168.11.0/24 network, PC2 (192.168.11.10) should not be able to ping PC3.

You can also use an extended ping from the **Fa0/1 interface** on R1 to the Fa0/1 interface on R2.

Does the ping successful ?



From PC2, it is impossible to ping to PC3

From fa0/1 interface R1, it is possible to ping to fa0/1 interface on R2

What is the result of the ping ?

Why ?

From PC2, it is impossible to ping to PC3 but can ping to the Server because we only set ACL on interface fa0/0 on R2 denying the network of PC2

You should see the following message on the R2 console:

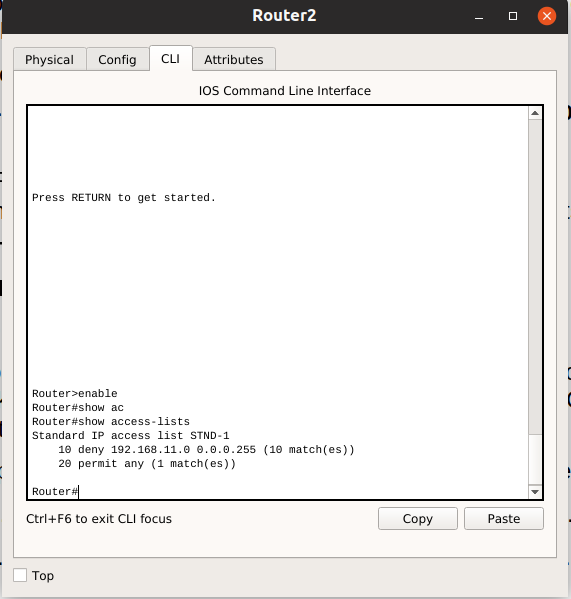
**\*Sep 4 03:22:58.935: %SEC-6-IPACCESSLOGNP: list STND-1 denied 0 0.0.0.0 -> 192.168.11.1, 1 packet**

In privileged EXEC mode on R2, issue the **show access-lists** command. You see output similar to the following. Each line of an ACL has an associated counter showing how many packets have matched the rule.

**Standard IP access list STND-1**

**10 deny 192.168.11.0, wildcard bits 0.0.0.255 log (5 matches)**

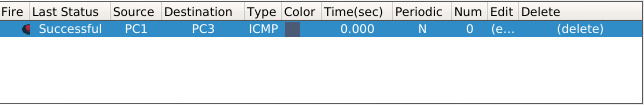
**20 permit any (25 matches)**

****

The purpose of this ACL was to block hosts from the 192.168.11.0/24 network. Any other hosts, such as those on the 192.168.10.0/24 network should be allowed access to the networks on R2. Conduct another test from PC1 to PC3 to ensure that this traffic is not blocked.

You can also use an extended ping from the **Fa0/0 interface** on R1 to the Fa0/0 interface on R2.

Does the ping successful ?

Yes

What is the result of the ping ?

Why ?

Because we only set ACL deny traffic from interface fa0/1 of R1 to interface fa0/0 of R2

**Task 4: Configuring an Extended ACL**

When greater granularity is required, you should use an extended ACL. Extended ACLs can filter traffic based on more than just source address. Extended ACLs can filter on protocol, source, and destination IP addresses, and source and destination port numbers.

An additional policy for this network states that devices from the 192.168.10.0/24 LAN are only permitted to reach internal networks. Computers on this LAN are not permitted to access the Internet. Therefore, these users must be blocked from reaching the IP address 209.165.200.225. Because this requirement needs to enforce both source and destination, an extended ACL is needed.

In this task, you are configuring an extended ACL on R1 that blocks traffic originating from any device on the 192.168.10.0/24 network to access the 209.165.200.255 host (the simulated ISP). This ACL will be applied outbound on the R1 Serial 0/0/0 interface. A typical best practice for applying extended ACLs is to place them as close to the source as possible.

Before beginning, verify that you can ping 209.165.200.225 from PC1.

**Step 1: Configure a named extended ACL.**

In global configuration mode, create a named extended ACL called EXTEND-1.

**R1(config)#ip access-list extended EXTEND-1**

Notice that the router prompt changes to indicate that you are now in extended ACL configuration mode. From this prompt, add the necessary statements to block traffic from the 192.168.10.0/24 network to the host. Use the host keyword when defining the destination.

**R1(config-ext-nacl)#deny ip 192.168.10.0 0.0.0.255 host 209.165.200.225**

Recall that the implicit “**deny all**” blocks all other traffic without the additional permit statement. Add the permit statement to ensure that other traffic is not blocked.

**R1(config-ext-nacl)#permit ip any any**

**Step 2: Apply the ACL.**

With standard ACLs, the best practice is to place the ACL as close to the destination as possible. Extended ACLs are typically placed close to the source. The EXTEND-1 ACL will be placed on the Serial interface, and will filter outbound traffic.

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

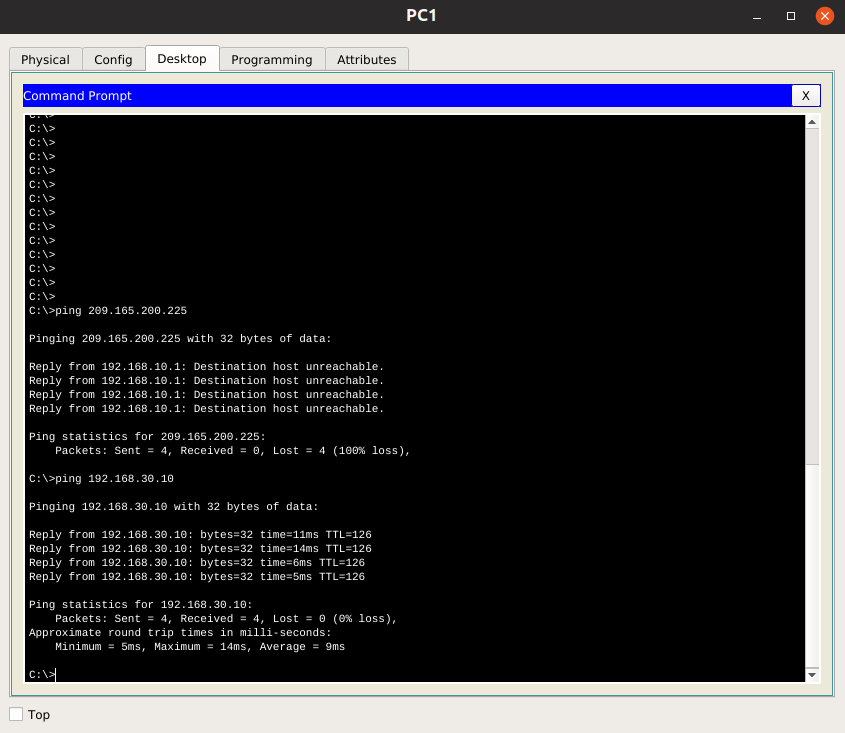
**R1(config-if)#ip access-group EXTEND-1 out**

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

**R1#copy run start**

**Step 3: Test the ACL.**

From PC1, ping the loopback interface on R2. These pings should fail, because all traffic from the 192.168.10.0/24 network is filtered when the destination is 209.165.200.225. If the destination is any other address, the pings should succeed. Confirm this by pinging PC3 from the 192.168.10.0/24 network device.



**Note**: The extended ping feature on R1 cannot be used to test this ACL, since the traffic will originate within R1 and will never be tested against the ACL applied to the R1 serial interface.

You can further verify this by issuing the **show ip access-list** on R1 after pinging.

**R1#show ip access-list**

**Extended IP access list EXTEND-1**

**10 deny ip 192.168.10.0 0.0.0.255 host 209.165.200.225 (4 matches)**

**20 permit ip any any**

**Task 5: Control Access to the VTY Lines with a Standard ACL**

It is good practice to restrict access to the router VTY lines for remote administration. An ACL can be applied to the VTY lines, allowing you to restrict access to specific hosts or networks. In this task, you will configure a standard ACL to permit hosts from two networks to access the VTY lines. All other hosts are denied.

Verify that you can telnet to R2 from R1.

**Step 1: Configure the ACL.**

Configure a named standard ACL on R2 that permits traffic from 10.2.2.0/30. Deny all other traffic. Call the ACL TASK-5.

**R2(config)#ip access-list standard TASK-5**

**R2(config-std-nacl)#permit 192.168.10.0 0.0.0.255**

**Step 2: Apply the ACL.**

Enter line configuration mode for VTY lines 0–4.

**R2(config)#line vty 0 4**

Use the **access-class** command to apply the ACL to the vty lines in the inbound direction. Note that this differs from the command used to apply ACLs to other interfaces.

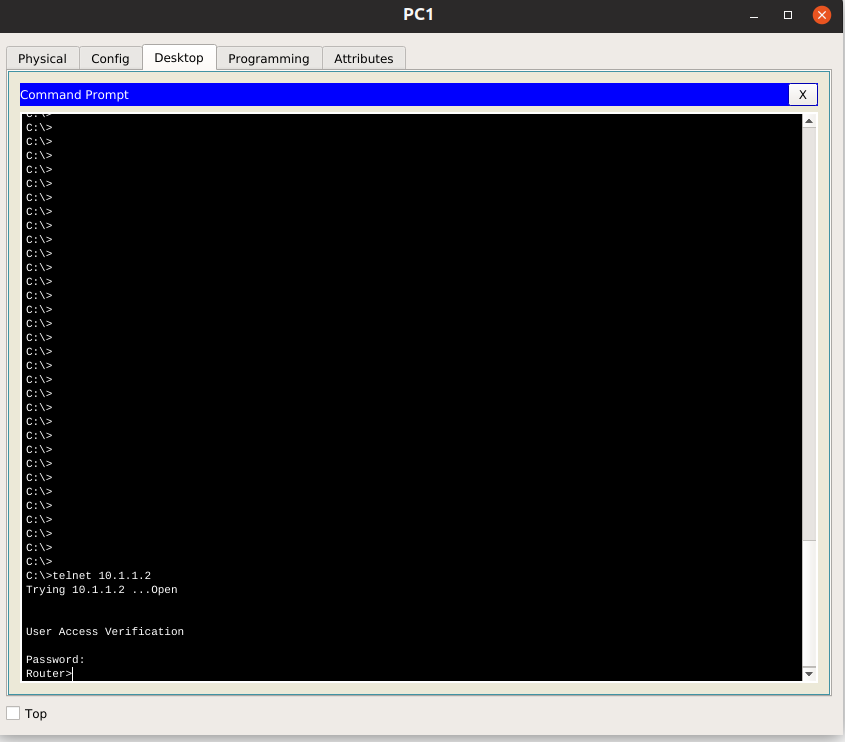
**R2(config-line)#access-class TASK-5 in**

**R2(config-line)#end**

**R2#copy run start**

**Step 3: Test the ACL**

From PC1, telnet to R2. What is the result ?

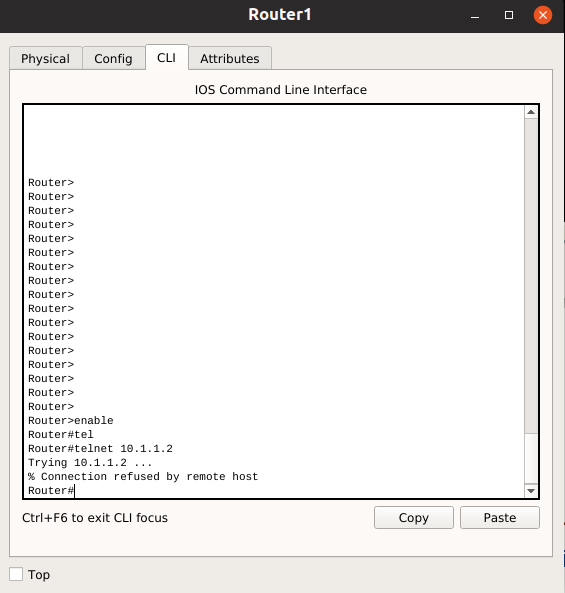


Why ?

Because we allow PC1 to access Router 2 vty line

From R1, telnet to R2 from R1. Does the connection attempt successfully ?

No



Why ? Why do connection attempts from other networks fail even though they are not specifically listed in the ACL?

Because we only allow network 192.168.10.0 to access the vty line of R2

**Task 6: Troubleshooting ACLs**

When an ACL is improperly configured or applied to the wrong interface or in the wrong direction, network traffic may be affected in an undesirable manner.

Step 1: Remove ACL STND-1 from Fa0/0 of R2.

In an earlier task, you created and applied a named standard ACL on R2. Use the **show running-config** command to view the ACL and its placement. You should see that an ACL named STND-1 was configured and applied outbound on FastEthernet 0/0. Recall that this ACL was designed to block all network traffic with a source address from the 192.168.11.0/24 network from accessing the LAN3.

To remove the ACL, go to interface configuration mode for FastEthernet 0/0 on R2. Use the **no ip access-group STND-1 out** command to remove the ACL from the interface.

Use the **show running-config** command to confirm that the ACL has been removed from FastEthernet 0/0.

**Step 2: Apply ACL STND-1 on Fa0/0 inbound.**

To test the importance of ACL filtering direction, reapply the STND-1 ACL to the Fa0/0interface. This time the ACL will be filtering inbound traffic, rather than outbound traffic. Remember to use the **in** keyword when applying the ACL.

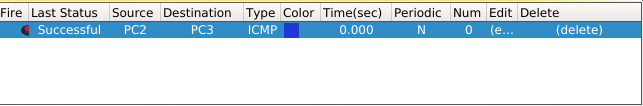
**R2(config)#interface Fa0/0**

**R2(config-if)#ip access-group STND-1 in**

**Step 3: Test the ACL.**

Test the ACL by pinging from PC2 to PC3. Does the pings success ?

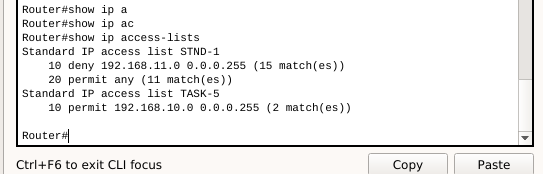
Yes



Why ?

Because ACL only filters inbound traffic, while PC2 ping to PC3 is outbound traffic

Issuing the **show ip access-list** command on R2. Do the ACL counters are incremented ?



Yes

**Step 4: Restore the ACL to its original configuration.**

Remove the ACL from the inbound direction and reapply it to the outbound direction.

**R3(config)#interface fa0/0**

**R3(config-if)#no ip access-group STND-1 out**

**R3(config-if)#ip access-group STND-1 in**

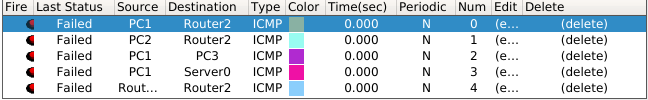
**Step 5: Apply TASK-5 to the R2 serial 0/0/0 interface inbound.**

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

**R2(config-if)#ip access-group TASK-5 in**

**Step 6: Test the ACL.**

Attempt to communicate to any device connected to R2 from R1 or LAN2. Notice that all communication is blocked; however, ACL counters are not incremented. This is because of the implicit “deny all” at the end of every ACL. This deny statement will prevent all inbound traffic to serial 0/0/0 from any source other than LAN1. Essentially, this will cause routes from LAN2 to be removed from the routing table.



Remove ACL TASK-5 from the interface, and save your configurations.

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

**R2(config-ino ip access-group TASK-5 in**

**f)#R2(config)#exit**

**R2#copy run start**

**Task 7: Document the Router Configurations**

Capture the configuration of routers to a text file and attach to your report.

**Task 8: Clean Up**

Erase the configurations and reload the routers. Disconnect and store the cabling. For PC hosts that are normally connected to other networks, such as the school LAN or the Internet, reconnect the appropriate cabling and restore the TCP/IP settings.

***Lab 6.2: DHCP & NAT Configuration***

**Topology Diagram**

**R1**

**R2**

**ISP**



**PC1**

**PC2**

Fa0/0

Fa0/1

S0/0/0

S0/0/0



**Server**

Fa0/0



S0/0/1

**ISP**

**Addressing Table**

**x = 0**

|  |  |  |  |
| --- | --- | --- | --- |
| **Device** | **Interface** | **IP address** | **Subnet-mask** |
| **R1** | S0/0/0 | 10.1.1.1 | 255.255.255.252 |
| Fa0/0 | 192.168.0.1 | 255.255.255.0 |
| Fa0/1 | 192.168.1.1 | 255.255.255.0 |
| **R2** | S0/0/0 | 10.1.1.2 | 255.255.255.252 |
| S0/0/1 | 209.165.200.225 | 255.255.255.252 |
| Fa0/0 | 192.168.2.254 | 255.255.255.0 |
| **ISP** | S0/0/1 | 209.165.200.226 | 255.255.255.252 |

**Learning Objectives**

Upon completion of this lab, you will be able to:

* Prepare the network.
* Perform basic router configurations.
* Configure a Cisco IOS DHCP server.
* Configure static and default routing.
* Configure static NAT.
* Configure dynamic NAT with a pool of addresses.
* Configure NAT overload.

**Scenario**

In this lab, you will configure the DHCP and NAT IP services. One router is the DHCP server. The other router forwards DHCP requests to the server. You will also configure both static and dynamic NAT configurations, including NAT overload. When you have completed the configurations, verify the connectivity between the inside and outside addresses.

**Task 1: Prepare the Network**

**Step 1: Cable a network that is similar to the one in the topology diagram.**

You can use any current router in your lab as long as it has the required interfaces shown in the topology.

**Step 2: Clear all existing configurations on the routers.**

**Task 2: Perform Basic Router Configurations**

Configure the R1, R2, and ISP routers according to the following guidelines:

* Configure the device hostname.
* Disable DNS lookup.
* Configure a privileged EXEC mode password.
* Configure a message-of-the-day banner.
* Configure a password for the console connections.
* Configure a password for all vty connections.
* Configure IP addresses on all routers. The PCs receive IP addressing from DHCP later in the lab.
* Enable OSPF with process ID 1 on R1 and R2. Do not advertise the 209.165.20x.224/27 network.

**Task 3: Configure PC1 and PC2 to receive an IP address through DHCP**

Make sure the button is selected that says Obtain an IP address automatically.



Once this has been done on both PC1 and PC2, they are ready to receive an IP address from a DHCP server.

**Task 4: Configure a Cisco IOS DHCP Server**

Cisco IOS software supports a DHCP server configuration called **Easy IP**. The goal for this lab is to have devices on the networks 192.168.10.0/24 and 192.168.11.0/24 request IP addresses via DHCP from R2.

**Step 1: Exclude statically assigned addresses.**

The DHCP server assumes that all IP addresses in a DHCP address pool subnet are available for assigning to DHCP clients. You must specify the IP addresses that the DHCP server should not assign to clients. These IP addresses are usually static addresses reserved for the router interface, switch management IP address, servers, and local network printer. The **ip dhcp excluded-address** command prevents the router from assigning IP addresses within the configured range. The following commands exclude the first 10 IP addresses from each pool for the LANs attached to R1. These addresses will not be assigned to any DHCP clients.

R2(config)#**ip dhcp excluded-address 192.168.x0.1 192.168.x0.10**

R2(config)#**ip dhcp excluded-address 192.168.x1.1 192.168.x1.10**

**Step 2: Configure the pool.**

Create the DHCP pool using the **ip dhcp pool** command and name it R1Fa0.

R2(config)#**ip dhcp pool R1Fa0**

Specify the subnet to use when assigning IP addresses. DHCP pools automatically associate with an interface based on the network statement. The router now acts as a DHCP server, handing out addresses in the 192.168.x0.0/24 subnet starting with 192.168.x0.1.

R2(dhcp-config)#network 192.168.x0.0 255.255.255.0

Configure the default router and domain name server for the network. Clients receive these settings via DHCP, along with an IP address.

R2(dhcp-config)#dns-server 192.168.x1.5

R2(dhcp-config)#default-router 192.168.x0.1

**Note:** There is not a DNS server at 192.168.x1.5. You are configuring the command for practice only.

Because devices from the network 192.168.x1.0/24 also request addresses from R2, a separate pool must be created to serve devices on that network. The commands are similar to the commands shown above:

R2(config)#ip dhcp pool R1Fa1

R2(dhcp-config)#network 192.168.x1.0 255.255.255.0

R2(dhcp-config)#dns-server 192.168.x1.5

R2(dhcp-config)#default-router 192.168.x1.1

**Step 3: Test DHCP**

On PC1 and PC2 test whether each has received an IP address automatically. On each PC go to Start -> Run -> cmd -> ipconfig

What are the results of your test?

The results are not correct. PC1 and PC2 ip addresses do not belongs to corresponding subnets.

Why are these the results?

Because DHCP server and DHCP clients are not on the same subnet, so R1 does not forward DHCP broadcasts to R2

**Step 4: Configure a helper address.**

Network services such as DHCP rely on Layer 2 broadcasts to function. When the devices providing these services exist on a different subnet than the clients, they cannot receive the broadcast packets. Because the DHCP server and the DHCP clients are not on the same subnet, configure R1 to forward DHCP broadcasts to R2, which is the DHCP server, using the **ip helper-address interface configuration** command.

Notice that ip helper-address must be configured on each interface involved.

R1(config)#interface fa0/0

R1(config-if)#ip helper-address 1x.1.1.2

R1(config)#interface fa0/1

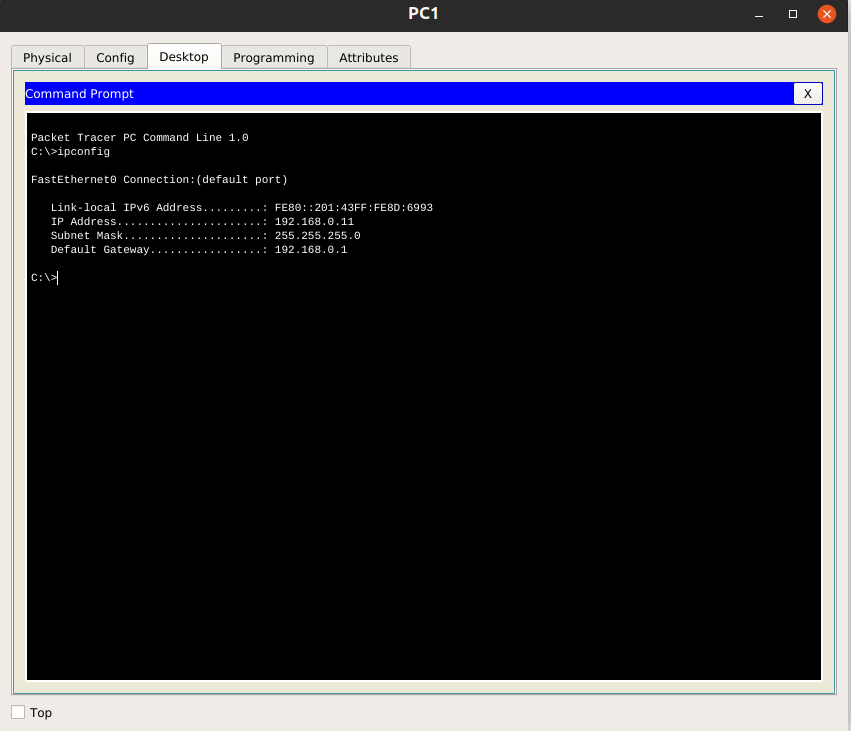
R1(config-if)#ip helper-address 1x.1.1.2

**Step 5: Release and Renew the IP addresses on PC1 and PC2**

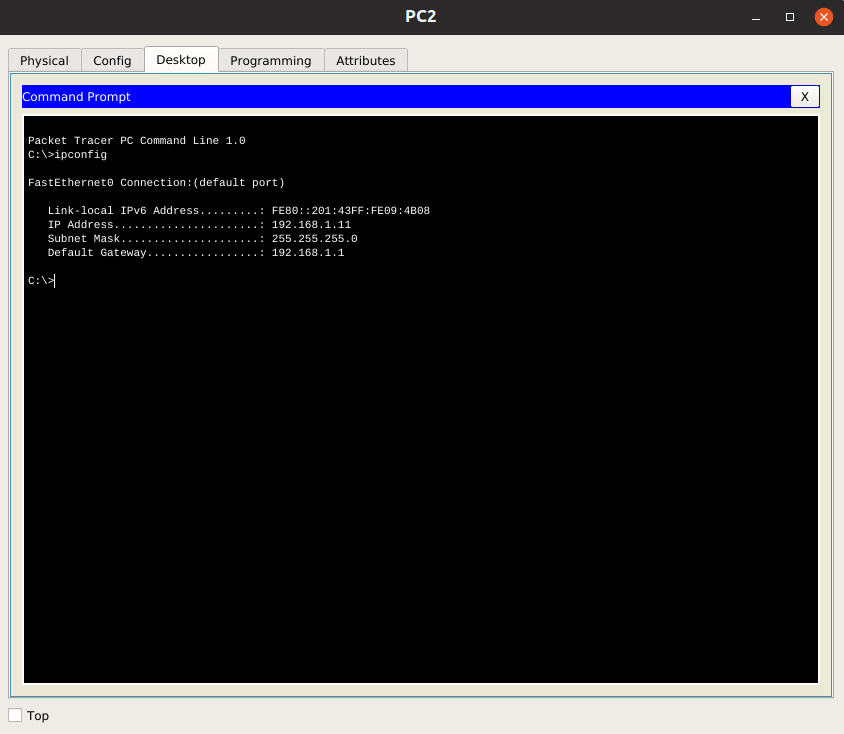
Depending upon whether your PCs have been used in a different lab, or connected to the internet, they may already have learned an IP address automatically from a different DHCP server. We need to clear this IP address using the **ipconfig /release** and **ipconfig /renew** commands.

What is the IP addresses of PC1 & PC2 ?

**PC1:**

****

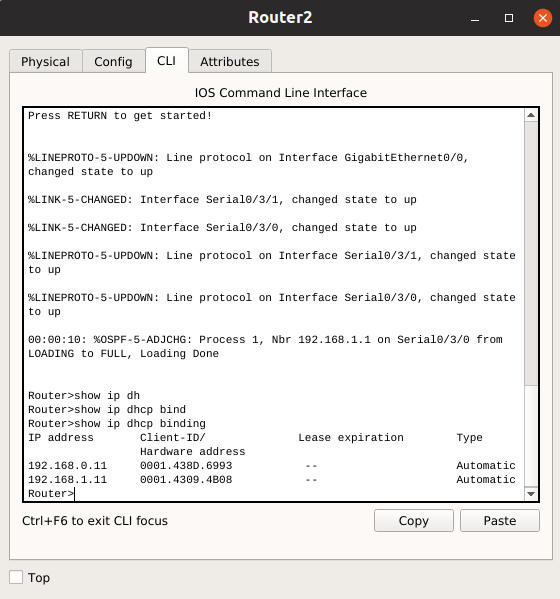
**PC2:**

****

**Step 6: Verify the DHCP configuration.**

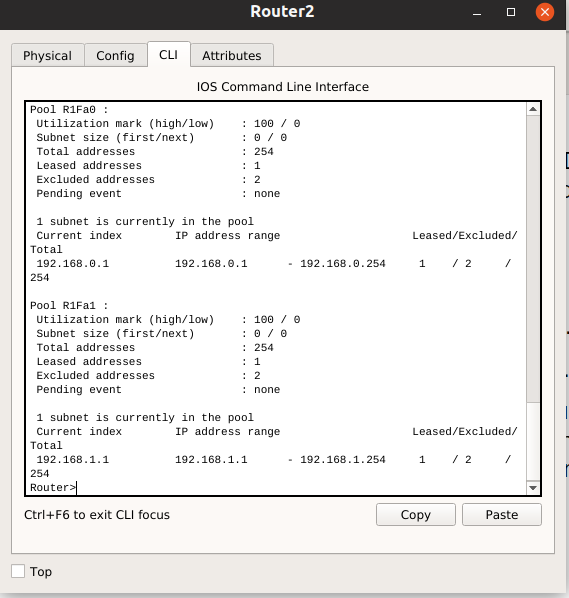
You can verify the DHCP server configuration in several different ways. Issue the command **ipconfig** on PC1 and PC2 to verify that they have now received an IP address dynamically. You can then issue commands on the router to get more information. The **show ip dhcp binding** command provides information on all currently assigned DHCP addresses. For instance, the following output shows that the IP address 192.168.10.11 has been assigned to MAC address 3031.632e.3537.6563. The IP lease expires on September 14, 2007 at 7:33 p.m.

R1#show ip dhcp binding



The **show ip dhcp pool** command displays information on all currently configured DHCP pools on the router. In this output, the pool R1Fa0 is configured on R1. One address has been leased from this pool. The next client to request an address will receive 192.168.10.12.

R2#show ip dhcp pool



How many leased addresses are there ?

There are two leased addresses

What is it ?

192.168.0.11

192.168.1.11

The **debug ip dhcp server events** command can be extremely useful when troubleshooting DHCP leases with a Cisco IOS DHCP server. The following is the debug output on R1 after connecting a host. Notice that the highlighted portion shows DHCP giving the client an address of 192.168.x0.12 and mask of 255.255.255.0

\*Sep 13 21:04:18.072: DHCPD: Sending notification of DISCOVER:

\*Sep 13 21:04:18.072: DHCPD: htype 1 chaddr 001c.57ec.0640

\*Sep 13 21:04:18.072: DHCPD: remote id 020a0000c0a80b01010000000000

\*Sep 13 21:04:18.072: DHCPD: circuit id 00000000

\*Sep 13 21:04:18.072: DHCPD: Seeing if there is an internally specified pool class:

\*Sep 13 21:04:18.072: DHCPD: htype 1 chaddr 001c.57ec.0640

\*Sep 13 21:04:18.072: DHCPD: remote id 020a0000c0a80b01010000000000

\*Sep 13 21:04:18.072: DHCPD: circuit id 00000000

\*Sep 13 21:04:18.072: DHCPD: there is no address pool for 192.168.11.1.

\*Sep 13 21:04:18.072: DHCPD: Sending notification of DISCOVER:

R1#

\*Sep 13 21:04:18.072: DHCPD: htype 1 chaddr 001c.57ec.0640

\*Sep 13 21:04:18.072: DHCPD: remote id 020a0000c0a80a01000000000000

\*Sep 13 21:04:18.072: DHCPD: circuit id 00000000

\*Sep 13 21:04:18.072: DHCPD: Seeing if there is an internally specified pool class:

\*Sep 13 21:04:18.072: DHCPD: htype 1 chaddr 001c.57ec.0640

\*Sep 13 21:04:18.072: DHCPD: remote id 020a0000c0a80a01000000000000

\*Sep 13 21:04:18.072: DHCPD: circuit id 00000000

R1#

\*Sep 13 21:04:20.072: DHCPD: Adding binding to radix tree (192.168.10.12)

\*Sep 13 21:04:20.072: DHCPD: Adding binding to hash tree

\*Sep 13 21:04:20.072: DHCPD: assigned IP address 192.168.10.12 to client 0063.6973.636f.2d30.3031.632e.3537.6563.2e30.3634.302d.566c.31.

\*Sep 13 21:04:20.072: DHCPD: Sending notification of ASSIGNMENT:

\*Sep 13 21:04:20.072: DHCPD: address 192.168.x0.12 mask 255.255.255.0

\*Sep 13 21:04:20.072: DHCPD: htype 1 chaddr 001c.57ec.0640

\*Sep 13 21:04:20.072: DHCPD: lease time remaining (secs) = 86400

\*Sep 13 21:04:20.076: DHCPD: Sending notification of ASSIGNMENT:

\*Sep 13 21:04:20.076: DHCPD: address 192.168.x0.12 mask 255.255.255.0

R1#

\*Sep 13 21:04:20.076: DHCPD: htype 1 chaddr 001c.57ec.0640

\*Sep 13 21:04:20.076: DHCPD: lease time remaining (secs) = 86400

**Task 5: Configure Static and Default Routing**

ISP uses static routing to reach all networks beyond R2. However, R2 translates private addresses into public addresses before sending traffic to ISP. Therefore, ISP must be configured with the public addresses that are part of the NAT configuration on R2. Enter the following static route on ISP:

ISP(config)#**ip route 209.165.200.240 255.255.255.240 serial 0/0/1**

This static route includes all addresses assigned to R2 for public use.

Configure a default route on R2 and propagate the route in OSPF.

R2(config)#ip route 0.0.0.0 0.0.0.0 209.165.20x.226

R2(config)#**router ospf 1**

R2(config-router)#**default-information originate**

Allow a few seconds for R1 to learn the default route from R2 and then check the R1 routing table. Alternatively, you can clear the routing table with the **clear ip route \*** command. A default route pointing to R2 should appear in the R1 routing table. Note that the static route that is configured on the ISP only routes to the public addresses that the R1 hosts will use after NAT is configured on R2. Until NAT is configured, the static route will lead to an unknown network, causing the pings from R1 to fail.

**Task 6: Configure Static NAT**

**Step 1: Statically map a public IP address to a private IP address.**

The inside server attached to R2 is accessible by outside hosts beyond ISP. Statically assign the public IP address 209.165.200.254 as the address for NAT to use to map packets to the private IP address of the inside server at 192.168.x2.254.

R2(config)#**ip nat inside source static 192.168.x2.254 209.165.20x.254**

**Step 2: Specify inside and outside NAT interfaces.**

Before NAT can work, you must specify which interfaces are inside and which interfaces are outside.

R2(config)#interface serial 0/0/1

R2(config-if)#**ip nat outside**

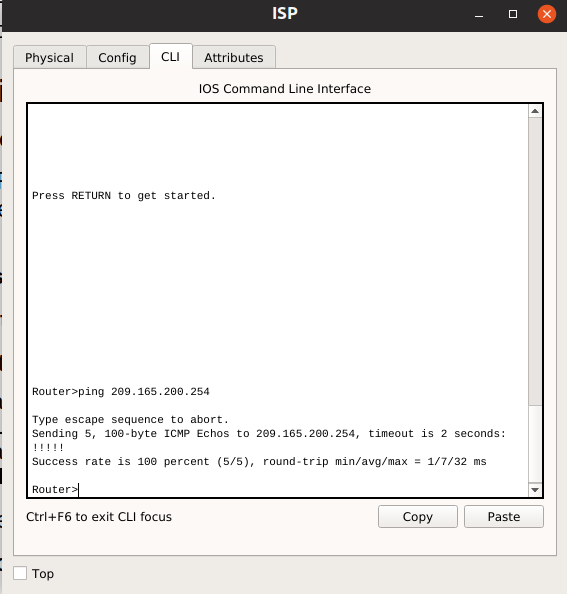
R2(config-if)#interface fa0/0

R2(config-if)#**ip nat inside**

**Note**: If using a simulated inside server, assign the ip nat inside command to the loopback interface.

**Step 3: Verify the static NAT configuration.**

From ISP, ping the public IP address 209.165.20x.254.



**Task 7: Configure Dynamic NAT with a Pool of Addresses**

While static NAT provides a permanent mapping between an internal address and a specific public address, dynamic NAT maps private IP addresses to public addresses. These public IP addresses come from a NAT pool.

**Step 1: Define a pool of global addresses.**

Create a pool of addresses to which matched source addresses are translated. The following command creates a pool named MY-NAT-POOL that translates matched addresses to an available IP address in the 209.165.20x.241–209.165.20x.246 range.

R2(config)#**ip nat pool MY-NAT-POOL 209.165.200.241 209.165.200.246 netmask 255.255.255.248**

**Step 2: Create an extended access control list to identify which inside addresses are translated.**

R2(config)#ip access-list extended NAT

R2(config-ext-nacl)#**permit ip 192.168.x0.0 0.0.0.255 any**

R2(config-ext-nacl)#**permit ip 192.168.x1.0 0.0.0.255 any**

**Step 3: Establish dynamic source translation by binding the pool with the access control list.**

A router can have more than one NAT pool and more than one ACL. The following command tells the router which address pool to use to translate hosts that are allowed by the ACL.

R2(config)#**ip nat inside source list NAT pool MY-NAT-POOL**

**Step 4: Specify inside and outside NAT interfaces.**

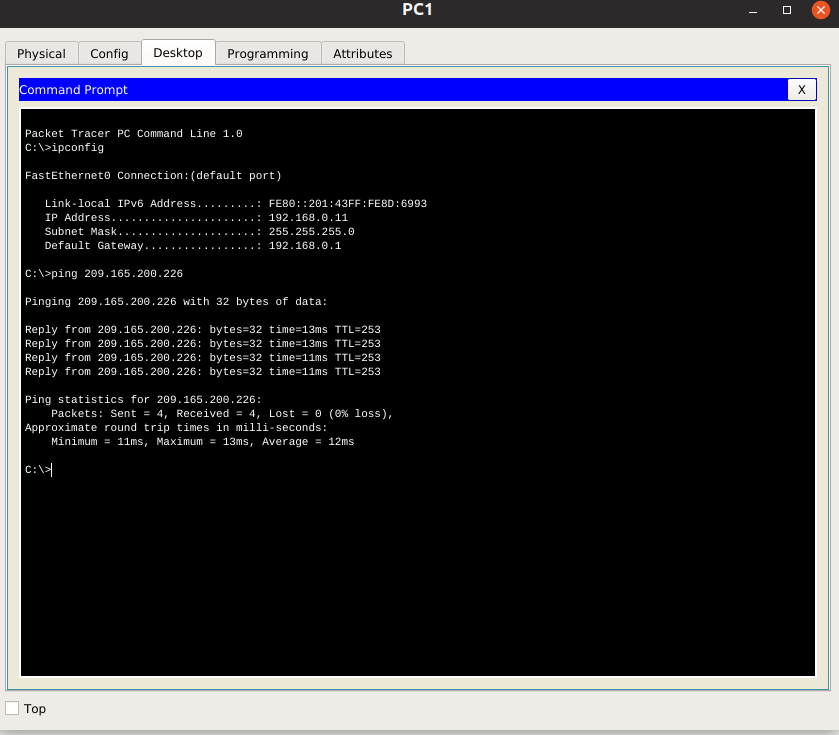
You have already specified the inside and outside interfaces for your static NAT configuration. Now add the serial interface linked to R1 as an inside interface.

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

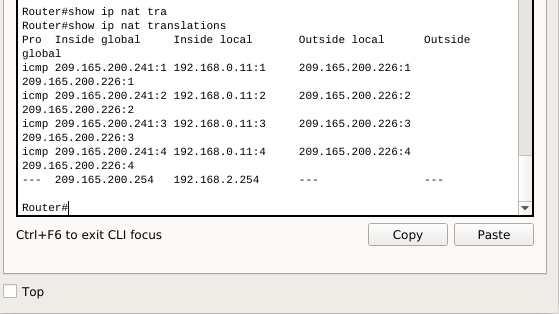
R2(config-if)#**ip nat inside**

**Step 5: Verify the configuration.**

Ping ISP from PC1 or the Fast Ethernet interface on R1 using extended ping. Then use the **show ip nat translations** and **show ip nat statistics** commands on R2 to verify NAT.



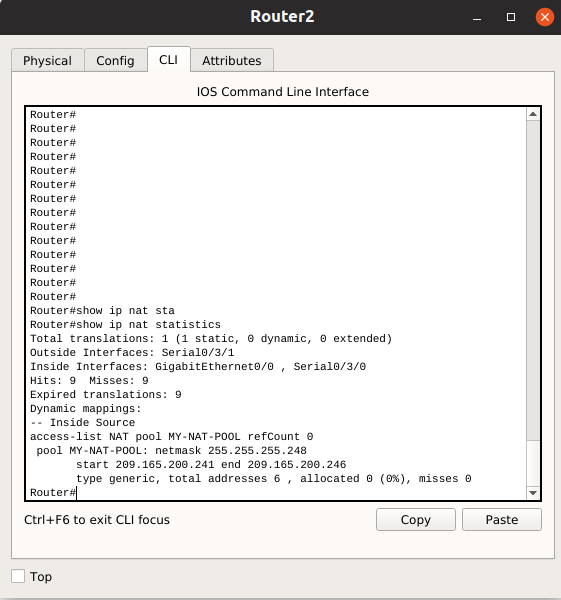
R2#**show ip nat translations**



What are the current translations on the nat table ?

Translate 192.168.0.11 to 209.165.200.241

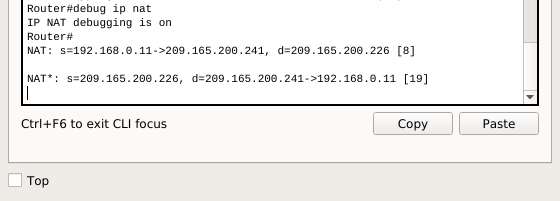
R2#show ip nat statistics



To troubleshoot issues with NAT, you can use the **debug ip nat** command. Turn on NAT debugging and repeat the ping from PC1.

R2#debug ip nat

IP NAT debugging is on



R2#

\*Sep 13 21:15:02.215: NAT\*: s=192.168.10.11->209.165.20x.241, d=209.165.20x.226 [25]

\*Sep 13 21:15:02.231: NAT\*: s=209.165.20x.226, d=209.165.20x.241->192.168.10.11 [25]

\*Sep 13 21:15:02.247: NAT\*: s=192.168.10.11->209.165.20x.241, d=209.165.20x.226 [26]

\*Sep 13 21:15:02.263: NAT\*: s=209.165.20x.226, d=209.165.20x.241->192.168.10.11 [26]

\*Sep 13 21:15:02.275: NAT\*: s=192.168.10.11->209.165.20x.241, d=209.165.20x.226 [27]

\*Sep 13 21:15:02.291: NAT\*: s=209.165.20x.226, d=209.165.20x.241->192.168.10.11 [27]

\*Sep 13 21:15:02.307: NAT\*: s=192.168.10.11->209.165.20x.241, d=209.165.20x.226 [28]

\*Sep 13 21:15:02.323: NAT\*: s=209.165.20x.226, d=209.165.20x.241->192.168.10.11 [28]

\*Sep 13 21:15:02.335: NAT\*: s=192.168.10.11->209.165.20x.241, d=209.165.20x.226 [29]

\*Sep 13 21:15:02.351: NAT\*: s=209.165.20x.226, d=209.165.20x.241->192.168.10.11 [29]

R2#

**Task 8: Configure NAT Overload**

In the previous example, what would happen if you needed more than the six public IP addresses that the pool allows?

Then any clients after could not access to the ISP

By tracking port numbers, NAT overloading allows multiple inside users to reuse a public IP address.

In this task, you will remove the pool and mapping statement configured in the previous task. Then you will configure NAT overload on R2 so that all internal IP addresses are translated to the R2 S0/0/1 address when connecting to any outside device.

**Step 1: Remove the NAT pool and mapping statement.**

Use the following commands to remove the NAT pool and the map to the NAT ACL.

R2(config)#no ip nat inside source list NAT pool MY-NAT-POOL

R2(config)#no ip nat pool MY-NAT-POOL 209.165.20x.241 209.165.20x.246 netmask 255.255.255.248

If you receive the following message, clear your NAT translations.

%Pool MY-NAT-POOL in use, cannot destroy

R2#**clear ip nat translation \***

**Step 2: Configure PAT on R2 using the serial 0/0/1 interface public IP address.**

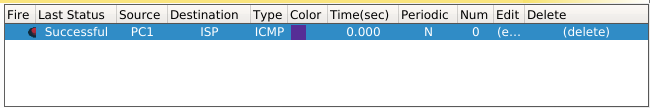
The configuration is similar to dynamic NAT, except that instead of a pool of addresses, the interface keyword is used to identify the outside IP address. Therefore, no NAT pool is defined. The overload keyword enables the addition of the port number to the translation.

Because you already configured an ACL to identify which inside IP addresses to translate as well as which interfaces are inside and outside, you only need to configure the following:

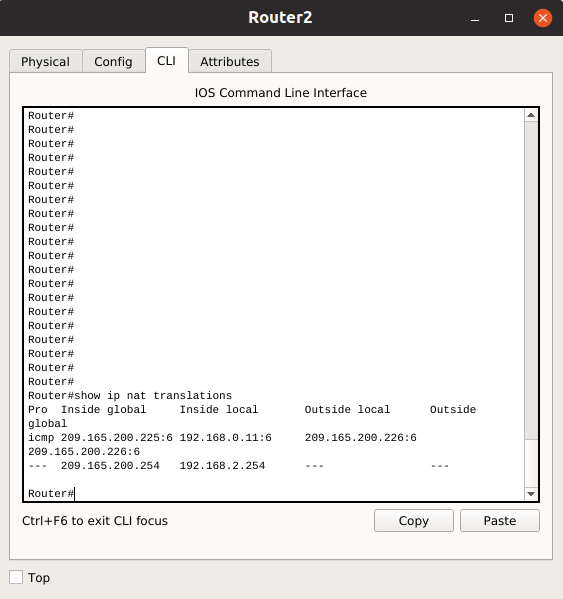
R2(config)#ip nat inside source list NAT interface S0/0/1 overload

**Step 3: Verify the configuration.**

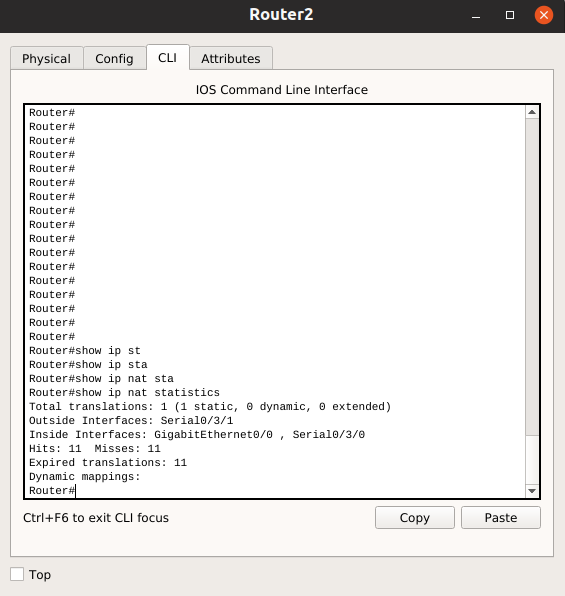
Ping ISP from PC1 or the Fast Ethernet interface on R1 using extended ping. Then use the show ip nat translations and show ip nat statistics commands on R2 to verify NAT.



R2#**show ip nat translations**



R2#**show ip nat statistics**



What are the current translations on the nat table ?

Translate from 192.168.0.11 to 209.165.200.225

List some important information in the output of the command **show ip nat statistics**

**Note**: In the previous task, you could have added the keyword overload to the **ip nat inside source list NAT pool MY-NAT-POOL** command to allow for more than six concurrent users.

**Task 9: Document the Network**

On each router, issue the show run command and capture the configurations. Attach your text file to your report.

**Task 10: Clean Up**

Erase the configurations and reload the routers. Disconnect and store the cabling. For PC hosts that are normally connected to other networks, such as the school LAN or the Internet, reconnect the appropriate cabling and restore the TCP/IP settings.