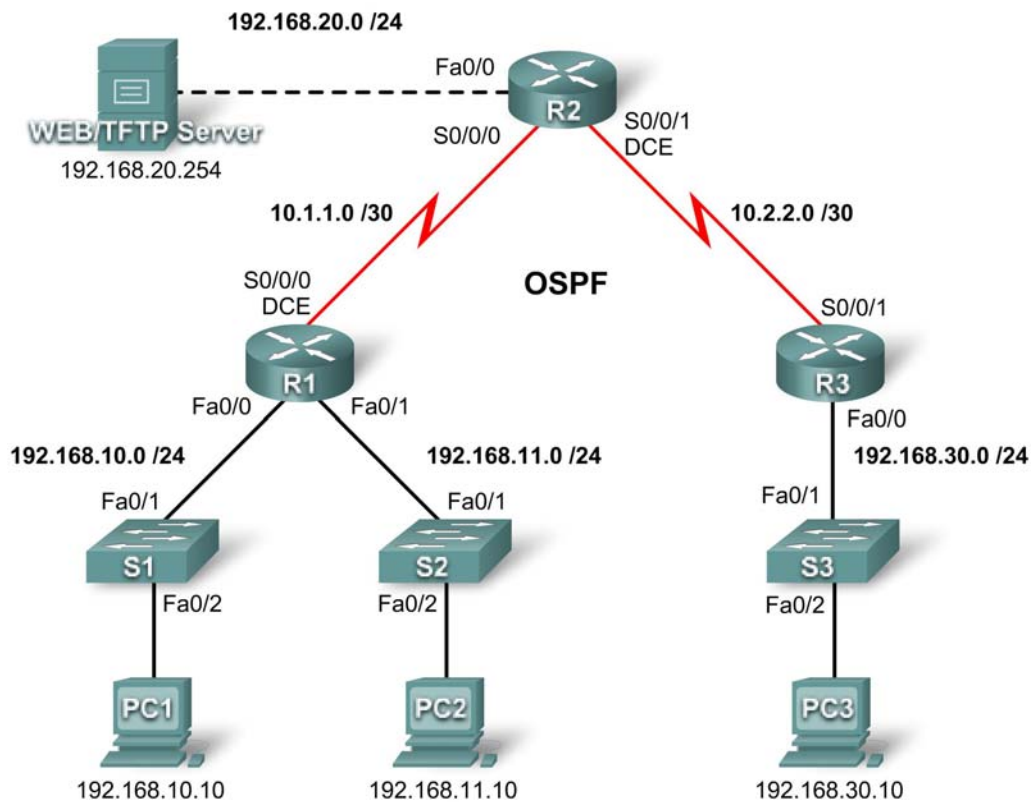# PT Activity 5.5.1: Basic Access Control Lists

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	192.168.20.1	255.255.255.0	N/A
	S0/0/0	10.1.1.2	255.255.255.252	N/A
	S0/0/1	10.2.2.1	255.255.255.252	N/A
	Lo0	209.165.200.225	255.255.255.224	N/A
R3	Fa0/0	192.168.30.1	255.255.255.0	N/A
	S0/0/1	10.2.2.2	255.255.255.252	N/A

Addressing Table on the next page

Addressing Table continued

S1	VLAN 1	192.168.10.2	255.255.255.0	192.168.10.1
S2	VLAN 1	192.168.11.2	255.255.255.0	192.168.11.1
S3	VLAN 1	192.168.30.2	255.255.255.0	192.168.30.1
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
Web Server	NIC	192.168.20.254	255.255.255.0	192.168.20.1

Learning Objectives

- Perform basic router and switch configurations.
- Configure a standard ACL.
- Configure an extended ACL.
- Control access to the vty lines with a standard ACL.
- Troubleshoot ACLs.

Introduction

In this activity, you will design, apply, test and troubleshoot access list configurations.

Task 1: Perform Basic Router and Switch Configurations

Step 1. Configure the routers and switches.

Configure the R1, R2, R3, S1, S2, and S3 routers and switches according to the following guidelines:

- Configure hostnames to match the topology diagram.
- Disable DNS lookup.
- Configure an encrypted privileged EXEC password of **class**.
- Configure a **message-of-the-day** banner.
- Configure a password of **cisco** for console connections.
- Configure a password of **cisco** for vty connections.
- Configure IP addresses and masks on all devices. Set the clock rate to **64000**.
- Enable OSPF using process ID 1 on all routers for all networks.
- Configure a loopback interface on R2.
- Configure IP addresses for the VLAN 1 interface on each switch.
- Configure each switch with the appropriate default gateway.
- Verify full IP connectivity using the **ping** command.

Step 2. Configure the PCs and WEB/TFTP Server.

- Using the addressing table and diagram, configure IP addresses, subnet masks, and default gateways for each PC via the **Desktop** > **IP Configuration** tab.
- Configure the IP address, subnet mask and default gateway for the WEB/TFTP Server.

Step 3. Check results.

Your completion percentage should be 93%. If not, click **Check Results** to see which required components are not yet completed.

Task 2: Configuring a Standard ACL

Standard ACLs can filter traffic based on source IP address only. In this task, you are configuring a standard ACL that blocks traffic from the 192.168.11.0 /24 network. This ACL will be applied inbound on the R3 serial 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.

Step 1. Create the ACL.

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

```
R3(config)#ip access-list standard std-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.

```
R3(config-std-nacl)#deny 192.168.11.0 0.0.0.255
```

Permit all other traffic.

```
R3(config-std-nacl)#permit any
```

Step 2. Apply the ACL.

Apply the ACL std-1 as a filter on packets entering R3 through serial interface 0/0/1.

```
R3(config)#interface serial 0/0/1
R3(config-if)#ip access-group std-1 in
```

Step 3. Test the ACL.

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.

In privileged EXEC mode on R3, 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 std-1
    deny 192.168.11.0 0.0.0.255 (3 match(es))
    permit any
```

Step 4. Check results.

Your completion percentage should be 95%. If not, click **Check Results** to see which required components are not yet completed.

Task 3: 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, 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. This ACL will be applied outbound on the R1 Serial 0/0/0 interface.

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
```

Step 3. Test the ACL.

From PC1 or any other device on the 192.168.10.0 /24 network, 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 R3 from the 192.168.10.0 /24 network device.

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

You should have matches for both rules of the ACL. This is because the ping from PC1 to R2s loopback interface was denied while the ping to R3 was permitted.

```
R1#show ip access-list
Extended IP access list extend-1
    deny ip 192.168.10.0 0.0.0.255 host 209.165.200.225 (4 match(es))
    permit ip any any (4 match(es))
```

Step 4. Check results.

Your completion percentage should be 99%. If not, click **Check Results** to see which required components are not yet completed.

Task 4: 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 both R1 and R3.

Step 1. Configure the ACL.

Configure a named standard ACL on R2 that permits traffic from 10.2.2.0 /30 and 192.168.30.0 /24. Deny all other traffic. Call the ACL **Task-4**.

```
R2(config)#ip access-list standard Task-4
R2(config-std-nacl)#permit 10.2.2.0 0.0.0.3
R2(config-std-nacl)#permit 192.168.30.0 0.0.0.255
```

Step 2. Apply the ACL.

Enter line configuration mode for vty access.

```
R2(config)#line vty 0 15
```

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-4 in
```

Step 3. Test the ACL.

Telnet to R2 from R1. Note that R1 does not have IP addresses in the address range listed in the ACL Task-4 permit statements. Connection attempts should fail.

From R3, telnet to R2 or any device on the 192.168.30.0 /24 network. You will be presented with a prompt for the vty line password.

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

_____

_____

Step 4. Check results.

Your completion percentage should be 100%. If not, click **Check Results** to see which required components are not yet completed.

Task 5: 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. Test the ACL.

In an earlier task, you created and applied a named standard ACL on R3. Use the **show running-config** command to view the ACL and its placement. You should see that an ACL named **std-1** was configured and applied inbound on Serial 0/0/1. 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 LAN on R3.

To remove the ACL, go to interface configuration mode for Serial 0/0/1 on R3.

```
R3(config)#interface serial 0/0/1
```

Use the **no ip access-group std-1 in** command to remove the ACL from the interface.

```
R3(config-if)#no ip access-group std-1 in
```

Use the **show running-config** command to confirm that the ACL has been removed from Serial 0/0/1

Step 2. Apply ACL std-1 on S0/0/1 outbound.

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

```
R3(config-if)#ip access-group std-1 out
```

Step 3. Test the ACL.

Test the ACL by pinging from PC2 to PC3. As an alternative, use an extended ping from R1. Notice that this time pings succeed, and the ACL counters are not incremented. Confirm this by issuing the **show ip access-list** command on R3.

Step 4. Restore the ACL to its original configuration.

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

```
R3(config)#interface serial 0/0/1
R3(config-if)#no ip access-group std-1 out
R3(config-if)#ip access-group std-1 in
```

Step 5. Apply Task-4 to the R2 serial 0/0/0 interface inbound.

```
R2(config)#interface serial 0/0/0
R2(config-if)#ip access-group Task-4 in
```

Step 6. Test the ACL.

Attempt to communicate to any device connected to R2 or R3 from R1 or its attached networks. 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.

You should see messages similar to the following printed on the consoles of R1 and R2 after the OSPF dead timers expires:

```
*Sep  4 09:51:21.757: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.11.1 on
Serial0/0/0 from FULL to DOWN, Neighbor Down: Dead timer expired
```

Remove ACL Task-4 from the interface.