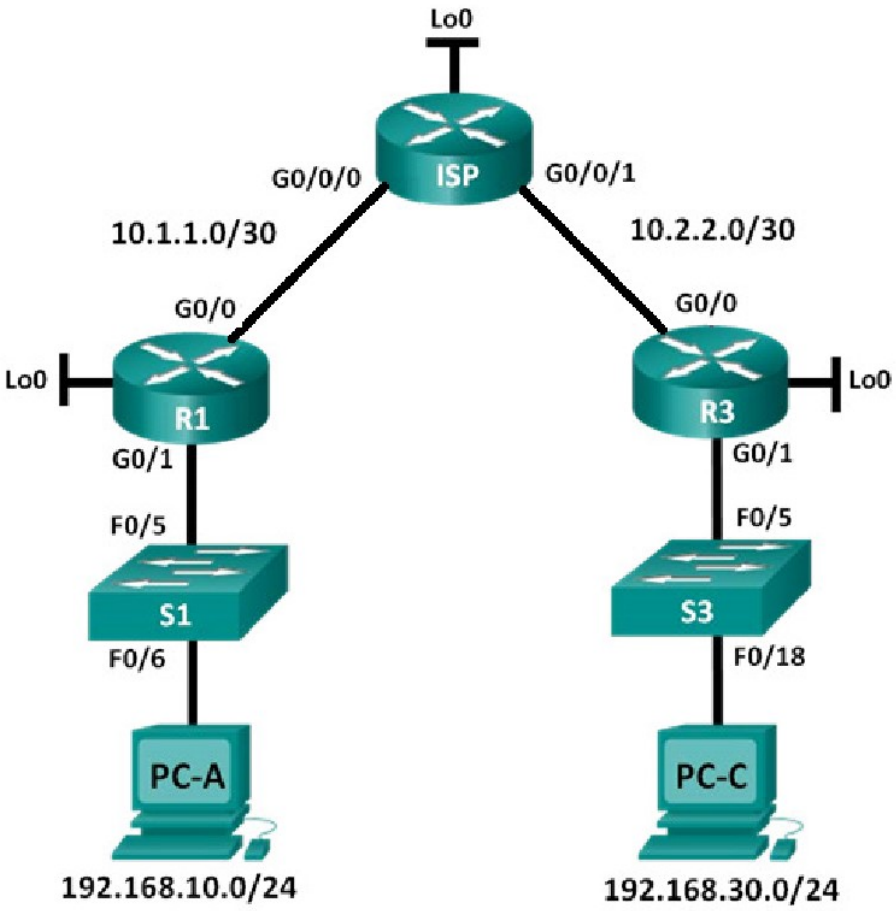
Lab – Configuring and Verifying Standard IPv4 ACLs

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/1	192.168.10.1	255.255.255.0	N/A
	Lo0	192.168.20.1	255.255.255.0	N/A
	G0/0	10.1.1.1	255.255.255.252	N/A
ISP	G0/0/0	10.1.1.2	255.255.255.252	N/A
	Lo0	209.165.200.225	255.255.255.224	N/A
	G0/0/1	10.2.2.2	255.255.255.252	N/A
R3	G0/0	10.2.2.1	255.255.255.252	N/A
	G0/1	192.168.30.1	255.255.255.0	N/A
	Lo0	192.168.40.1	255.255.255.0	N/A
PC-A	NIC	192.168.10.3	255.255.255.0	192.168.10.1
PC-C	NIC	192.168.30.3	255.255.255.0	192.168.30.1

Objectives

Part 1: Set Up the Topology and Initialize Devices

- Set up equipment to match the network topology.
- Initialize and reload the routers and switches.

Part 2: Configure Devices and Verify Connectivity

- Assign a static IP address to PCs.
- Configure basic settings on routers.
- Configure basic settings on switches.
- Configure static and default routes on R1, ISP, and R3.
- Verify connectivity between devices.

Part 3: Configure and Verify Standard Numbered and Named ACLs

- Configure, apply, and verify a numbered standard ACL.
- Configure, apply, and verify a named ACL.

Part 4: Modify a Standard ACL

- Modify and verify a named standard ACL.
- Test the ACL.

Background / Scenario

Network security is an important issue when designing and managing IP networks. The ability to configure proper rules to filter packets, based on established security policies, is a valuable skill.

In this lab, you will set up filtering rules for two offices represented by R1 and R3. Management has established some access policies between the LANs located at R1 and R3, which you must

implement. The ISP router sitting between R1 and R3 will not have any ACLs placed on it. You would not be allowed any administrative access to an ISP router because you can only control and manage your own equipment.

Note: The routers used with CCNA hands-on labs are Cisco 2900s Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). The switches used are Cisco Catalyst 2960s with Cisco IOS Release 15.0(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs.

Note: Make sure that the routers and switches have been erased and have no startup configurations. If you are unsure, contact your instructor.

Required Resources

- 3 Routers (Cisco 2900 with Cisco IOS Release 15.2(4)M3 universal image or comparable)
- 2 Switches (Cisco 2960 with Cisco IOS Release 15.0(2) lanbasek9 image or comparable)
- 2 PCs (Windows 10 with terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Part 1: Set Up the Topology and Initialize Devices

In Part 1, you set up the network topology and clear any configurations, if necessary.

Step 1: Cable the network as shown in the topology.

Step 2: Initialize and reload the routers and switches.

Part 2: Configure Devices and Verify Connectivity

In Part 2, you configure basic settings on the routers, switches, and PCs. Refer to the Topology and Addressing Table for device names and address information.

Step 1: Configure IP addresses on PC-A and PC-C.

Step 2: Configure basic settings for the routers.

a. Console into the router and enter global configuration mode.

b. Configure the device name as shown in the topology.

c. Create loopback interfaces on each router as shown in the Addressing Table.

```
R1(config)# interface loopback number
      or
R1(config)# interface lo number
R1(config-if)# ip address ip-address subnet-mask
R1(config-if)# exit
```

where *number* can be any value between 0 – 2147483647.

d. Configure interface IP addresses as shown in the Topology and Addressing Table.

Step 3: (Optional) Configure basic settings on the switches.

a. Console into the switch and enter global configuration mode.

b. Configure the device name as shown in the topology.

Step 4: Configure static and default routes on R1, ISP, and R3.

a. Configure necessary static and default routes on R1, ISP, and R3.

b. After configuring, verify that all routers have complete routing tables listing all networks. Troubleshoot if this is not the case.

Step 5: Verify connectivity between devices.

Note: It is very important to test whether connectivity is working **before** you configure and apply access lists! You want to ensure that your network is properly functioning before you start to filter traffic.

a. From PC-A, ping PC-C and the loopback interface on R3. Were your pings successful?

b. From R1, ping PC-C and the loopback interface on R3. Were your pings successful?

c. From PC-C, ping PC-A and the loopback interface on R1. Were your pings successful?

d. From R3, ping PC-A and the loopback interface on R1. Were your pings successful?

Part 3: Configure and Verify Standard Numbered and Named ACLs

Step 1: Configure a numbered standard ACL.

Standard ACLs filter traffic based on the source IP address only. A typical best practice for standard ACLs is to configure and apply it as close to the destination as possible. For the first access list, create a standard numbered ACL that allows traffic from all hosts on the 192.168.10.0/24 network and all hosts on the 192.168.20.0/24 network to access all hosts on the 192.168.30.0/24 network and all hosts on the 192.168.40.0/24 network. The security policy also states that a **deny any** access control entry (ACE), also referred to as an ACL statement, should be present at the end of all ACLs.

What wildcard mask would you use to allow all hosts on the 192.168.10.0/24 network to access the 192.168.30.0/24 network?

Following Cisco's recommended best practices, on which router would you place this ACL?

On which interface would you place this ACL? In what direction would you apply it?

a. Configure the ACL on R3. Use 1 for the access list number.

```
R3(config)# access-list 1 remark Allow R1 LANs Access
R3(config)# access-list 1 permit 192.168.10.0 0.0.0.255
R3(config)# access-list 1 permit 192.168.20.0 0.0.0.255
R3(config)# access-list 1 deny any
```

b. Apply the ACL to the appropriate interface in the proper direction.

```
R3(config)# interface g0/0
R3(config-if)# ip access-group 1 in
```

c. Verify a numbered ACL.

The use of various **show** commands can aid you in verifying both the syntax and placement of your ACLs in your router.

To see access list 1 in its entirety with all ACEs, which command would you use?

What command would you use to see where the access list was applied and in what direction?

1) On R3, issue the **show access-lists 1** command.

```
R3# show access-list 1
Standard IP access list 1
    10 permit 192.168.10.0, wildcard bits 0.0.0.255
    20 permit 192.168.20.0, wildcard bits 0.0.0.255
    30 deny   any
```

2) On R3, issue the **show ip interface g0/0** command.

```
R3# show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 10.2.2.1/30
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.10
  Outgoing access list is not set
  Inbound access list is 1
  Output omitted
```

3) Test the ACL to see if it allows traffic from the 192.168.10.0/24 network access to the 192.168.30.0/24 network. From the PC-A command prompt, ping the PC-C IP address. Were the pings successful?

4) Test the ACL to see if it allows traffic from the 192.168.20.0/24 network access to the 192.168.30.0/24 network. You must do an extended ping and use the loopback 0 address on R1 as your source. Ping PC-C's IP address. Were the pings successful?

```
R1# ping
Protocol [ip]:
Target IP address: 192.168.30.3
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.20.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp,
Verbose[none]: Sweep range of sizes [n]:
Type escape sequence to abort. Sending 5, 100-byte ICMP Echos
to 192.168.30.3, timeout is 2 seconds:
Packet sent with a source address of 192.168.20.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
```

d. From the R1 prompt, ping PC-C's IP address again.

```
R1# ping 192.168.30.3
```

Was the ping successful? Why or why not?

Step 2: Configure a named standard ACL.

Create a named standard ACL that conforms to the following policy: allow traffic from all hosts on the 192.168.40.0/24 network access to all hosts on the 192.168.10.0/24 network. Also, only allow host PC-C access to the 192.168.10.0/24 network. The name of this access list should be called BRANCH-OFFICE-POLICY.

Following Cisco's recommended best practices, on which router would you place this ACL?

On which interface would you place this ACL? In what direction would you apply it?

a. Create the standard named ACL BRANCH-OFFICE-POLICY on R1.

```
R1(config)# ip access-list standard BRANCH-OFFICE-POLICY
R1(config-std-nacl)# permit host 192.168.30.3
R1(config-std-nacl)# permit 192.168.40.0 0.0.0.255
R1(config-std-nacl)# end
R1#
*Feb 15 15:56:55.707: %SYS-5-CONFIG_I: Configured from console by console
```

Looking at the first permit ACE in the access list, what is another way to write this?

b. Apply the ACL to the appropriate interface in the proper direction.

```
R1# config t
R1(config)# interface g0/1
R1(config-if)# ip access-group BRANCH-OFFICE-POLICY out
```

c. Verify a named ACL.

1) On R1, issue the **show access-lists** command.

```
R1# show access-lists
Standard IP access list BRANCH-OFFICE-POLICY
    10 permit 192.168.30.3
    20 permit 192.168.40.0, wildcard bits 0.0.0.255
```

2) On R1, issue the **show ip interface g0/1** command.

```
R1# show ip interface g0/1
GigabitEthernet0/1 is up, line protocol is up
  Internet address is 192.168.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.10
  Outgoing access list is BRANCH-OFFICE-POLICY
  Inbound access list is not set
<Output omitted>
```

3) Test the ACL. From the command prompt on PC-C, ping PC-A's IP address. Were the pings successful?

4) Test the ACL to ensure that only the PC-C host is allowed access to the 192.168.10.0/24 network. You must do an extended ping and use the G0/1 address on R3 as your source. Ping PC-A's IP address. Were the pings successful?

5) Test the ACL to see if it allows traffic from the 192.168.40.0/24 network access to the 192.168.10.0/24 network. You must perform an extended ping and use the loopback 0 address on R3 as your source. Ping PC-A's IP address. Were the pings successful?

Part 4: Modify a Standard ACL

It is common in business for security policies to change. For this reason, ACLs may need to be modified. In Part 4, you will change one of the previous ACLs you configured to match a new management policy being put in place.

Management has decided that users from the 209.165.200.224/27 network should be allowed full access to the 192.168.10.0/24 network. Management also wants ACLs on all of their routers to follow consistent rules. A **deny any** ACE should be placed at the end of all ACLs. You must modify the BRANCH-OFFICE-POLICY ACL.

You will add two additional lines to this ACL. There are two ways you could do this:

OPTION 1: Issue a **no ip access-list standard BRANCH-OFFICE-POLICY** command in global configuration mode. This would effectively take the whole ACL out of the router. Depending upon the router IOS, one of the following scenarios would occur: all filtering of packets would be cancelled and all packets would be allowed through the router; or, because you did not take off the **ip access-group** command on the G0/1 interface, filtering is still in place. Regardless, when the ACL is gone, you could retype the whole ACL, or cut and paste it in from a text editor.

OPTION 2: You can modify ACLs in place by adding or deleting specific lines within the ACL itself. This can come in handy, especially with ACLs that have many lines of code. The retyping of the whole ACL or cutting and pasting can easily lead to errors. Modifying specific lines within the ACL is easily accomplished.

Note: For this lab, use Option 2.

Step 1: Modify a named standard ACL.

a. From R1 privileged EXEC mode, issue a **show access-lists** command.

```
R1# show access-lists
Standard IP access list BRANCH-OFFICE-POLICY
    10 permit 192.168.30.3 (8 matches)
    20 permit 192.168.40.0, wildcard bits 0.0.0.255 (5 matches)
```

b. Add two additional lines at the end of the ACL. From global config mode, modify the ACL BRANCH-OFFICE-POLICY.

```
R1#(config)# ip access-list standard BRANCH-OFFICE-POLICY
R1(config-std-nacl)# 30 permit 209.165.200.224 0.0.0.31
R1(config-std-nacl)# 40 deny any
R1(config-std-nacl)# end
```

c. Verify the ACL.

1) On R1, issue the **show access-lists** command.

```
R1# show access-lists
Standard IP access list BRANCH-OFFICE-POLICY
    10 permit 192.168.30.3 (8 matches)
    20 permit 192.168.40.0, wildcard bits 0.0.0.255 (5 matches)
    30 permit 209.165.200.224, wildcard bits 0.0.0.31
    40 deny   any
```

Do you have to apply the BRANCH-OFFICE-POLICY to the G0/1 interface on R1?

2) From the ISP command prompt, issue an extended ping. Test the ACL to see if it allows traffic from the 209.165.200.224/27 network access to the 192.168.10.0/24 network. You must do an extended ping and use the loopback 0 address on ISP as your source. Ping PC-A's IP address. Were the pings successful?

Reflection

1. As you can see, standard ACLs work quite well. Why would you ever have the need for using extended ACLs?

2. Typically, more typing is required when using a named ACL as opposed to a numbered ACL. Why would you choose named ACLs over numbered?