

Modify IPv4 ACLs

Two Methods to Modify an ACL

After an ACL is configured, it may need to be modified. ACLs with multiple ACEs can be complex to configure. Sometimes the configured ACE does not yield the expected behaviors. For these reasons, ACLs may initially require a bit of trial and error to achieve the desired filtering result.

This section will discuss two methods to use when modifying an ACL:

- Use a Text Editor
- Use Sequence Numbers

Text Editor Method

ACLs with multiple ACEs should be created in a text editor. This allows you to plan the required ACEs, create the ACL, and then paste it into the router interface. It also simplifies the tasks to edit and fix an ACL.

For example, assume ACL 1 was entered incorrectly using 19 instead of 192 for the first octet, as shown in the running configuration.

```
R1# show run | section access-list
access-list 1 deny 19.168.10.10
access-list 1 permit 192.168.10.0 0.0.0.255
R1#
```

In the example, the first ACE should have been to deny the host at 192.168.10.10. However, the ACE was incorrectly entered.

To correct the error:

- Copy the ACL from the running configuration and paste it into the text editor.
- Make the necessary edits changes.
- Remove the previously configured ACL on the router otherwise, pasting the edited ACL commands will only append (i.e., add) to the existing ACL ACEs on the router.
- Copy and paste the edited ACL back to the router.

Assume that ACL 1 has now been corrected. Therefore, the incorrect ACL must be deleted, and the corrected ACL 1 statements must be pasted in global configuration mode, as shown in the output.

```
R1(config)# no access-list 1
R1(config)#
R1(config)# access-list 1 deny 192.168.10.10
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)#
R1#
```

Sequence Numbers Method

An ACL ACE can also be deleted or added using the ACL sequence numbers. Sequence numbers are automatically assigned when an ACE is entered. These numbers are listed in the **show access-lists** command. The **show running-config** command does not display sequence numbers.

In the previous example, the incorrect ACE for ACL 1 is using sequence number 10, as shown in the example.

```
R1# show access-lists
Standard IP access list 1
  10 deny 19.168.10.10
  20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

Use the **ip access-list standard** command to edit an ACL. Statements cannot be overwritten using the same sequence number as an existing statement. Therefore, the current statement must be deleted first with the **no 10** command. Then the correct ACE can be added using sequence number 10 is configured. Verify the changes using the **show access-lists** command, as shown in the example.

```
R1# conf t
R1(config)# ip access-list standard 1
R1(config-std-nacl)# no 10
R1(config-std-nacl)# 10 deny host 192.168.10.10
R1(config-std-nacl)# end
R1# show access-lists
Standard IP access list 1
  10 deny 192.168.10.10
  20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

Modify a Named ACL Example

Named ACLs can also use sequence numbers to delete and add ACEs. Refer to the example for ACL NO-ACCESS.

```
R1# show access-lists
Standard IP access list NO-ACCESS
  10 deny 192.168.10.10
  20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

Assume that host 192.168.10.5 from the 192.168.10.0/24 network should also have been denied. If you entered a new ACE, it would be appended to the end of the ACL. Therefore, the host would never be denied because ACE 20 permits all hosts from that network.

The solution is to add an ACE denying host 192.168.10.5 in between ACE 10 and ACE 20, such as ACE 15, as shown in the example. Also notice that the new ACE was entered without using the **host** keyword. The keyword is optional when specifying a destination host.

Use the **show access-lists** command to verify the ACL now has a new ACE 15 inserted appropriately before the permit statement.

Notice that sequence number 15 is displayed prior to sequence number 10. We might expect the order of the statements in the output to reflect the order in which they were entered. However, the IOS puts host statements in an order using a special hashing function. The resulting order optimizes the ACL to search by host entries first, and then by network entries.

Note: The hashing function is only applied to host statements in an IPv4 standard access list. The details of the hashing function are beyond the scope of this course.

```
R1# configure terminal
R1(config)# ip access-list standard NO-ACCESS
R1(config-std-nacl)# 15 deny 192.168.10.5
R1(config-std-nacl)# end
R1#
R1# show access-lists
Standard IP access list NO-ACCESS
  10 deny 192.168.10.1
  15 deny 192.168.10.5
  20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

ACL Statistics

Notice that the **show access-lists** command in the example shows statistics for each statement that has been matched. The deny ACE in the NO-ACCESS ACL has been matched 20 times and the permit ACE has been matched 64 times.

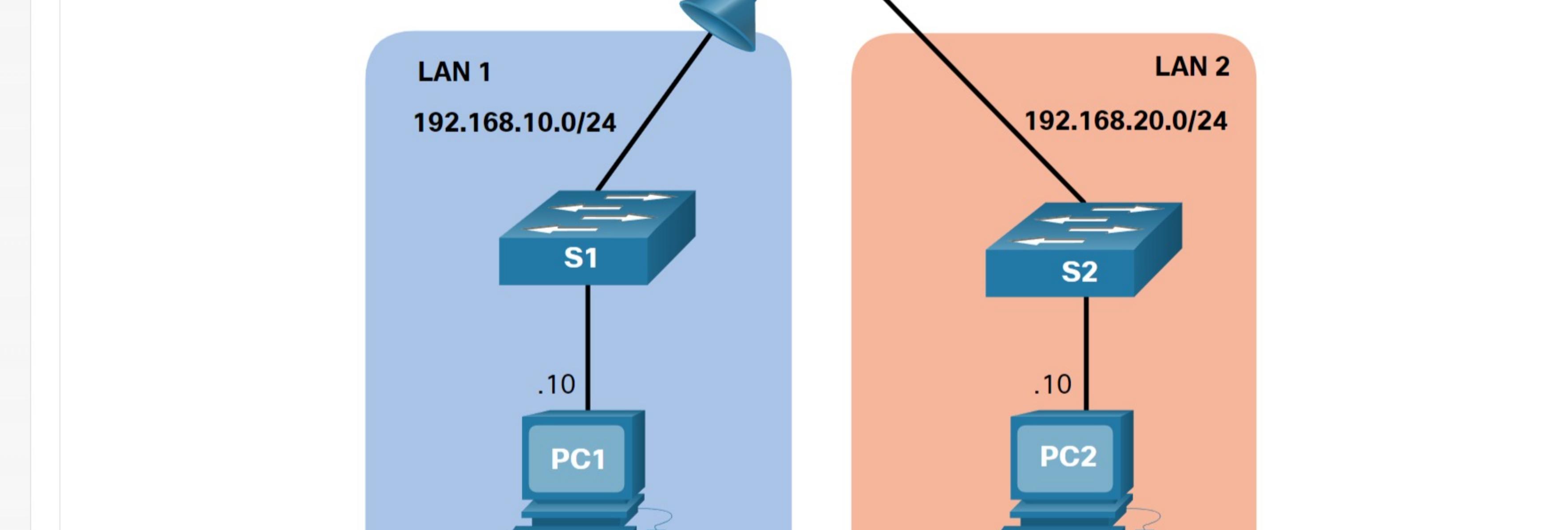
Note that the implied deny any the last statement does not display any statistics. To track how many implicit denied packets have been matched, you must manually configure the **deny any** command at the end of the ACL.

Use the **clear access-list counters** command to clear the ACL statistics. This command can be used alone or with the number or name of a specific ACL.

```
R1# show access-lists
Standard IP access list NO-ACCESS
  10 deny 192.168.10.10 (20 matches)
  20 permit 192.168.10.0, wildcard bits 0.0.0.255 (64 matches)
R1# clear access-list counters NO-ACCESS
R1# show access-lists
Standard IP access list NO-ACCESS
  10 deny 192.168.10.10
  20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

Syntax Checker – Modify IPv4 ACLs

Modify an ACL using sequence numbers.



Use the **show access-lists** command to verify the configured ACLs.

```
R1# show access-lists
Standard IP access list 1
  10 deny 19.168.10.10
  20 permit 192.168.10.0, wildcard bits 0.0.0.255
```

You notice that ACE 10 is incorrect and needs to be edited. Enter global configuration mode and use the **ip access-list standard** command for ACL 1.

```
R1#
```

Reset **Show Me** **Show All**

Packet Tracer – Configure and Modify Standard IPv4 ACLs

In this Packet Tracer activity, you will complete the following objectives:

- Part 1: Configure Devices and Verify Connectivity
- Part 2: Configure and Verify Standard Numbered and Named ACLs
- Part 3: Modify a Standard ACL

Configure and Modify Standard IPv4 ACLs

+ Configure and Modify Standard IPv4 ACLs