Course › Unit 5: Access Cont... › 5.3 Activity and Dis... › Activity: Configurin...

Activity: Configuring, Applying, and Testing ACLs

ACTIVITY: CONFIGURING, APPLYING, AND TESTING ACLS

This activity is ungraded.

Getting help: If you have trouble with these activities, please ask questions in the “Troubleshooting” forum in General Class Discussions.

Review: This simulation mimics what I demonstrated in these videos:

- Configuring and Testing a Standard ACL
- Configuring and Testing a Second Standard ACL
- Configuring and Testing an Extended ACL

System: For these activities, you’ll use Packet Tracer, a network simulation program from Cisco. See the first activity in this unit for download and installation instructions.

Time: This activity should take you 30 to 60 minutes to complete.

Goal

- To configure, apply, and test ACLs

Instructions

Note: Hit **Enter** after each command.

Configuring, Applying, and Testing a Standard ACL

1. On R2 (the router closer to 10.3.0.1), go into Global Configuration mode by clicking *enable, configure terminal*.
2. Enter **`access-list 1 deny 10.1.0.0 0.0.255.255`** to deny all traffic originating from the 10.1.0.0/16 subnet.
3. Enter **`access-list 1 permit any`** to permit any other traffic that doesn’t meet the above statement.
4. Enter **`int f0/1`** to go into interface configuration mode.
5. Enter **`ip access-group 1 out`** to apply the ACL as an outbound list on interface f0/1. Standard ACLs should be placed as close as possible to the destination to prevent unnecessary filtering. This means not only the router closest to the destination, but the interface as well.
6. Enter **`end`** to go back down to enable mode.
7. Enter **`show access-lists`** to see our ACL.
8. Enter **`show ip interface f0/1`** to see that the list is applied to the f0/1 interface as an outbound ACL.

You'll notice a line close to the top: *Outgoing access list is 1*. Advance line by line with the **Enter** key or page by page with the **spacebar**. Break out with `ctrl+c`.

9. Enter `show run` to see the entire configuration of this router.
10. From the PC 10.1.0.1, try once again to ping 10.3.0.1. This time the ping will fail, because of our ACL.
11. Go back to R2, and in Privileged Exec Mode, enter `show access-lists` as before.

This time, you'll notice that the output shows four packets met the first ACL statement and were blocked. These were, of course, the four ICMP Echo Requests sent by 10.1.0.1.

Modifying and Testing our Standard ACL

1. From Global Configuration Mode on R2, enter `no access-list 1` to remove our current ACL.
2. Enter `access-list 1 deny 10.1.0.2` to block just this host from 10.3.0.0/16.
3. Enter `access-list 1 permit any` to permit any other traffic that doesn't meet the above statement.
4. Enter `end` to go back down to Privileged Exec Mode.
   The new access-list 1 is automatically applied as an outbound ACL to int f0/1, since we never removed the application of the previous access-list 1.
5. Go back to the PC 10.1.0.1, and once again send a ping to 10.3.0.1. The ping should succeed once again, because 10.1.0.2 is being blocked now, not the entire 10.1.0.0/16 subnet.
6. Change the IP address of this PC to 10.1.0.2, and once again try to ping 10.3.0.1. Since the new IP address is 10.1.0.2, this set of pings will be blocked, as it matches the first statement in our new ACL.
7. Go back to R2, and enter `show access-lists`.
   You'll notice that each statement has four matches. The first statement was matched from the four ICMP Echo Requests sent from 10.1.0.2 after we changed the PC's IP address, while the second statement was matched from the four ICMP Echo Requests sent from 10.1.0.1 before we changed the PC's IP address.
8. Click *enable, configure terminal* to go to Global Configuration Mode.
9. Enter `no access-list 1` to remove the ACL.
10. Change the PC's IP address back from 10.1.0.2 to 10.1.0.1.
11. In Interface Configuration Mode (int f0/1), enter `no ip access-group 1 out` to remove the application of the ACL to interface f0/1.

Configuring, Applying, and Testing an Extended ACL

1. Go to Global Configuration Mode on R1.
2. Enter `access-list 101 deny tcp 10.1.0.1 0.0.0.0 10.3.0.1 0.0.0.0 eq 80` to block just HTTP traffic from 10.1.0.1 to 10.3.0.1 over port 80.
3. Enter `access-list 101 permit ip any any` to allow any traffic not meeting the above statement through.
4. Enter `int f0/1` to enter Interface Configuration Mode.
5. Enter `ip access-group 101 in` to apply this ACL as an inbound list on interface f0/1.
6. Go to the PC 10.1.0.1, and click the *Desktop* tab, and then *Web Browser*.
7. In the URL bar, enter `10.3.0.1`, and click the *Go* button. You'll see the message "Request Timeout."
8. Go back to R1, and in Privileged Exec Mode, enter `show access-lists`.
   You'll notice that the first statement has matches. These are the attempts to connect over the

default port of 80 from 10.1.0.1.

9. Go back to the PC 10.1.0.1, and in the Web Browser URL bar, enter `10.3.0.1:99`, to specify a non-default port of 99.
10. Now you'll notice a different message, *Server Reset Connection*, since there is no service on 10.3.0.1 listening on port 99.
11. Go back to R1, and in Privileged Exec Mode, enter **`show access-lists`**.

You'll notice that the second statement now has a match. This is the attempt to connect over the non-default port of 99 from 10.1.0.1, which wasn't blocked from the first ACL statement, and Activity: Constructing a Topology.

After you've finished, answer the **Check Your Work** questions.

Check Your Work

2/2 points (ungraded)

If an ACL has a number of 82, we can say with 100% certainty that it is a(n) _____ ACL.

○ Extended

○ Inbound

● Standard ✓

○ Outbound

An Extended ACL that blocks certain traffic leaving the network applied as close as possible to the source would be a(n) __________ ACL.

○ Outbound

○ Standard

○ Bidirectional

● Inbound ✓

Submit