Internetworking LAB 14

Basic Access Control Lists

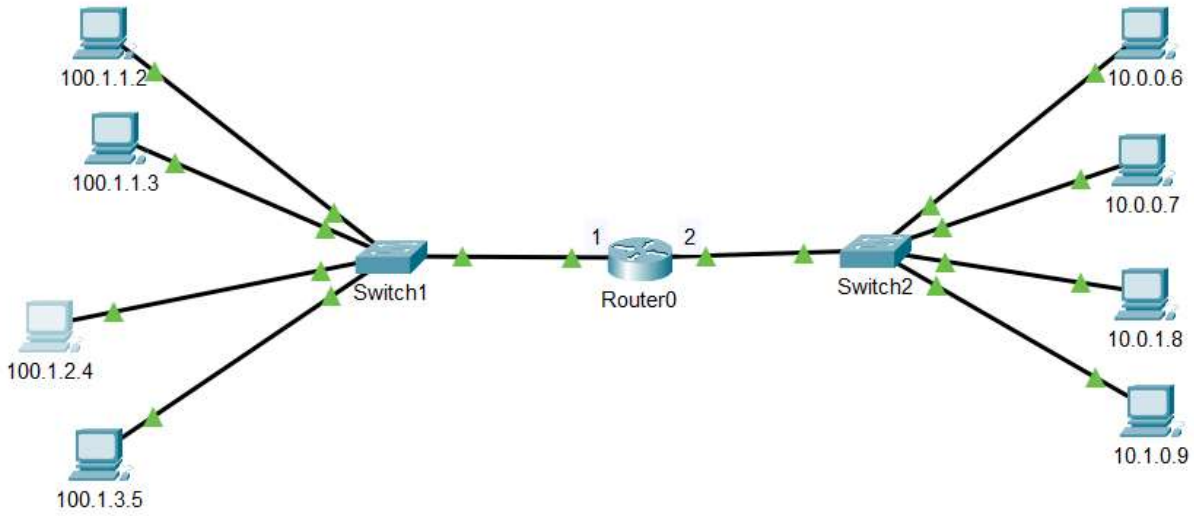
Objectives

Up to now a properly configured network allows all packets to pass through routers. Creating an Access Control List allows us to limit/control which packets are permitted or denied. Again, the theory behind this operation is explained in the text.

Step 1: Create and Configure the initial topology

a. In Packet Tracer turn simulation on – click on the SHOW NONE to clear the list, then click on EDIT FILTERS so you only filter ICMP frames

b. Create the below topology. Add a textbox with name, class id and "LAB 14"



c. IMPORTANT: **Update the router to add another G0 port for later use**.

d. IMPORTANT: Note the **exact** IP addresses of the PC's.

e. Note that we are using a CLASS A networks (10.0.0.0 and 100.0.0.0) – so our IP and subnet masks will be 255.0.0.0!

f. Configure the router's ports for IP address 100.0.0.1 and 10.0.0.1.

g. Configure all PC's – note that the default gateway will be 100.0.0.1 and 10.0.0.1

h. Important: At this point ping from each PC to PCs on the other subnet so that you have full connectivity. If a ping fails debug the wrong IP address or gateway address.

Part 2: Access Group Creation – Specific Address (no mask)

1. Since there are no ACL's all the pings work; all packets are allowed. But let's create our first access group.
    a. Enter global config mode on the Router.
    b. Enter **ip access-list standard 1** - we enter config-std nacl mode
    c. Enter **99 deny any** - a default row to deny all packets
    d. Enter **do show ip access-lists** - to verify that your ACL was created
    e. Examine the list – do you understand what you see?
2. While the access group is created, we need to enable it to a router port.
    a. Enter configuration mode for the port connected to Switch 1 – G0/0/1
    b. Enter **ip access-group 1 in**
3. Look at the running configuration and verify that the port has access group 1 assigned.

```
interface GigabitEthernet0/0/1
 ip address 100.0.0.1 255.0.0.0
 ip access-group 1 in
 duplex auto
 speed auto
```

4. Now, in simulation mode:
    a. Ping from PC2 thru PC5 to PC6. It should fail. Note where the packet fails. Why?
    b. Then ping from P^ to PC2 (the reverse order). Does it fails a different way?
5. Let's add a permit for one specific host:
    a. Enter **ip access-list standard 1** -- we enter config-std nacl mode
    b. Enter **10 permit 100.1.1.2** -- we want this before row 99 – why?
    c. Enter **do show ip access-lists** -- you should see rows 10 and 99
    d. Ping from PC2 to PC6 - It should pass FROM PC2! (*you may need 2 pings)*
    Ping from PC3 to PC6 - It should still fail.
6. Ping from PC6 to PC3 and watch what happens.
Why does the packet pass through going from right to left but fail returning left to right?
7. Let's add a few more commands:
    a. **15 permit 100.1.1.3** - then test – does the ping work?
    b. **no 15** - ping again, it should fail again
    What did the NO command do?
    c. 5 deny 100.1.1.2 - ping PC2 to PC6 – why does it now fail?
    d. do show ip access-lists - see that that the commands sequence themselves this is why we should use sequence numbers!
    e. no 5 - deletes that deny command

The "ANY" command should be the last command in the sequence. Let's see why.

f. Add the following sequences
5 deny 100.1.1.2
10 deny 100.1.1.3
20 deny 100.1.1.4
50 deny 100.1.15

g. All pings should fail.

h. **Do show ip access-lists**

i. Add: **2** permit any

j. All pings should work. Why?

Part 3: Access Group Creation – Using wildcards

1. Delete all sequences except 99 in access list #1.
Enter **do show ip access-lists** to verify**.**

2. Now add a permit for all addresses that have 100.1.1 as their first 3 bytes (addresses 100.1.1.2 and 100.1.1.3).

a. ip access-list standard 1

b. **10 permit 100.1.1.0 0.0.0.255** -- note the mask

c. **do show ip access-lists** -- do you see the mask?

d. Ping PC2 to PC6 and PC3 to PC6 -- these should work
Ping PC4 to PC6 and PC5 to PC6 -- these should fail

3. Now add a permit for all addresses that have 100.1. as their first 2 bytes (PC2-PC5).

a. ***20 permit 100.1.0.0 0.0.255.255*** -- note the mask

b. "*show ip access-lists"* -- do you see the mask?

c. Ping from PC2-PC5 to PC6 -- all pings should work

Part 4: ACL List logic

1. At this point pings from PC2-PC5 are all permitted.
2. Add a deny for PC3 at sequence 50.
   a. **50 deny 100.1.1.3**
   b. **Do show ip access-lists** *(*note: you may see different matches)

```
10 permit 100.1.1.0 0.0.0.255 (6 match(es))
20 permit 100.1.0.0 0.0.255.255 (3 match(es))
50 deny host 100.1.1.3
99 deny any (9 match(es))
```

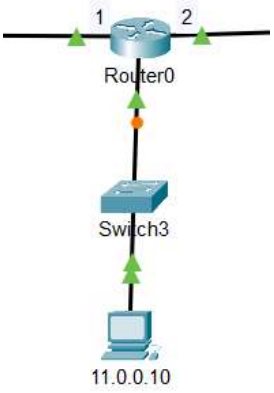
3. Ping PC3 to PC6 – Why didn't the deny work?

4. Change the sequence of the deny.
   a. **no 50** - delete that entry
   b. 5 deny 100.1.1.3
   c. do show ip access-lists

5. Ping PC3 to PC6 – Why does the deny work now?

Part 5: Create and Configure an Output ACL

1. Now add subnet – 11.0.0.0 /8 with a switch and PC. Configure IP addresses, subnet masks, default gateways, and router ports. When complete you should be able to ping from PC2, PC4 and PC5 to PC10. (Packets from PC3 should be denied by ACL 1.)



2. From Global Config Mode:
    a. **ip access-list standard 2**
    b. **99 permit any**
    c. **50 deny 100.1.2.4**
3. Show the access lists – you should now see 2, and note the sequence order of ACL 2
4. Move to specific configuration mode for the port g0/0/0
5. Enter **ip access-group 2 out** *- Note that we specify OUT (not IN)*
6. Ping from PC4 to PC10 fails, but from PC4 to PC6 works.
7. Examine this logic so you fully understand how important the the placement of the ACL is!

Part 6: Create a New ACL List

Create a standard ACL – number 10. You will create the list but don't configure to any port. Read through the requirements below carefully. Follow the logic and insure your sequence numbers are correct.

You can't necessarily sequence in the same order as the requirements below. You need to think this through logically!

a. Block all IP addresses not specified below.
b. Block all from network 9.x.x.x except 9.2.3.4
c. Always allow from 200.200.55.7
d. Block from 33.7.5.2
e. Block all from 192.168.x.x except 192.168.6.x
f. Allow any from 220.x.x.x except 220.100.x.x

Part 7: Lab Completion

1. Open a textbox and include all new commands introduced in module 14.
2. Save your packet file as "L14 *Lastname*.PKT"
3. Submit file in BrightSpace