BRYANT CCNA SEC13 – ACCESS LISTS

SEC 13.1 Access List Rules

ACLs are likely the most commonly-used Cisco feature in use today. It is very important to master ACL fundamentals.

In this section we will use ACLs primarily for permitting and denying traffic based on packet source/destination IP addresses.

ACLs are also used to identify traffic to trigger other features. This usage is beyond the scope of CCNA, however.

1. Every ACL has an implicit deny at the end
2. The ACL searches for a match starting at the top line, and stops when a match is found. No remaining lines are processed after a match.

When a packet enters or exits an interface that has an ACL applied, at least one packet value is compared against the ACL on a line-by-line basis. Here's an example:

```
R1(config)# access-list 1 permit 10.1.1.1 0.0.0.0
            "            1 permit 10.2.1.1 0.0.0.0
            "            1 permit 10.3.1.1 0.0.0.0
            "            1 permit 10.4.1.1 0.0.0.0
            "            1 permit 10.5.1.1 0.0.0.0
```

Standard ACLs can only match on the source IP address of a packet. This access list would only allow packets from those exact source IP addresses.

SEC 13.2 Wildcard Masking Made Easy

Wildcard masks are binary masks that work like this:
- zeros mean the address bits must match
- ones mean the bit is ignored in matching.

For example to allow packets from 196.17.100.0/24:

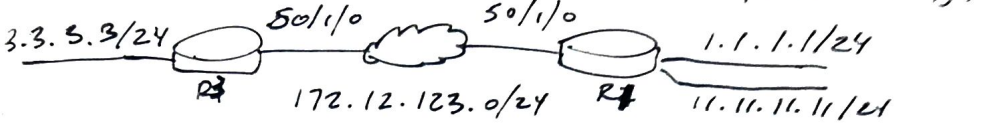
```
R1(config)# access-list 1 permit 196.17.100.0 0.0.0.255
```

SEC 13.3 Standard ACL Lab Begins

A Standard ACL is concerned only with the source IP address of the packet.

There are two numeric ranges to choose from when writing standard ACLs.

<1-99> IP Standard Access List
<1300-1999> IP Standard Access List (expanded range)



Requirements:

- Block traffic from 3.3.3.0/24 if it is destined for 11.11.11.0/24
- R1 should allow packets from 3.3.3.0/24 if intended for any other subnet, including those added in the future.
- The ACL must be applied on the serial interface of R1.

Let's try this with a standard ACL:

```
R1(config)# access-list 5 deny 3.3.3.0 0.0.0.255
          # access-list 5 allow any
```

and apply it to the interface:

```
R1# conf t
R1(config)# int serial 0/1/0
R1(config-if)# ip access-group 5 in
```

To test traffic from 3.3.3.0/24:

```
R3# ping 11.11.11.11 source 3.3.3.3
```

This is blocked! It should be noted, though, that we are not meeting all the requirements because ALL traffic from 3.3.3.0/24 is being blocked to ALL destinations on S0/1/0 on R1. We need to use extended ACLs

SEC 13.5 Extended ACL Lab

Let's try this again with an Extended ACL

```
R1(config)# access-list 100 deny ip 3.3.3.0 0.0.0.255 11.11.11.0 0.0.0.255
R1(config)#access-list 100 permit ip any any
          # int serial 0/1/0
R1(config-if)# ip access-group 100 in
```

: can test and everything works!

SEC 13.6 "Host", "any" and "the order of the lines"

- any is basically the same as entering any IP address with an all ones 32-bit mask (match any)
- host basically the same as an address with an all-zero wildcard mask.

Watch the order of your ACL lines!

Just one line out of place can ruin an ACL.

SEC 13.7 Named Standard ACL Lab

"Aren't there enough numbers?"

```
R1(config)# ip access-list BLOCK11
R1(config-std-nacl)# deny 3.3.3.0 0.0.0.255
                    # permit any
```

This creates a standard access list named BLOCK11

SEC 13.8 Named Extended ACL Lab

Quick note: delete the old ACL just like anything else:

```
R1(config)# no ip access-list standard BLOCK11
```

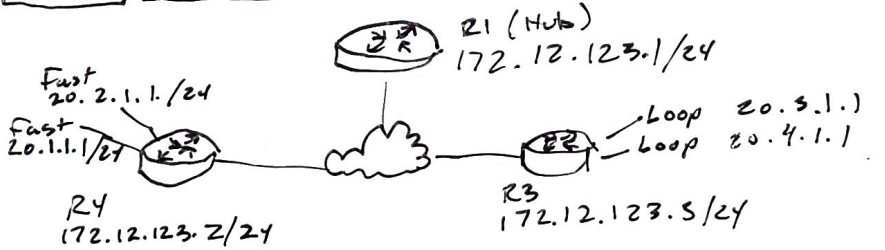
To set up a named extended ACL:

```
R1(config)# ip access-list extended BLOCK11
R1(config-ext-nacl)# deny ip 3.3.3.0 0.0.0.255 11.11.11.0 0.0.0.255
R1(config-ext-nacl)# permit ip any any
```

and apply it:

```
R1(config)# int serial 0/1/0
R1(config-if)# ip access-group BLOCK11
```

SEC 13.9 Telnet and Extended ACLs



On R1, setup telnet privileged login.

Setup an ACL that allows R2 to telnet to R1, and noone else.

We will use an Extended ACL:

```
R1(config)# access-list 101 permit tcp host 172.12.123.2 any eq 23
R1(config)# access-list 101 deny ip any any
```

The explicit deny is not strictly necessary but we will soon see some uses for that. NOTE: any traffic other than tcp port 23 from R2 will be dropped. Including pings and other traffic from R2.

Also, we can apply this rule on only the vty lines:

```
R1(config)# line vty 0 4
R1(config-line)# access-class 101 in
```

This applies the ACL to only the telnet interface and allows other traffic to the router.

ACL 13.10 telnet, ext. ACLs, and time ranges

We can also configure time ranges to use with ACL rules.

```
R1(config)# time-range TELNET_ALLOWED
R1(config-time-range)# periodic weekdays 9:00 to 17:00
```

This time-range runs Mon-Fri 9am-5pm.

Now, we will add this onto ACL 101 from before:

```
R1(config)# access-list 101 permit tcp host 172.12.123.3 any \
                time-range TELNET-ALLOWED
```

This rule would let R3 telnet to R1 from 9-5 weekdays. if there was not the explicit deny above it in the ACL.

SEC 13.11 Sequence Numbers, Ext. ACLs, and telnet

To remove line 30 from our ACL:

```
R1(config)# ip access-list extended 101
R1(config-ext-nacl)# no 30
R1(config-ext-nacl)# 15 permit tcp host 172.12.123.3 any eq 23 \
                time-range TELNET-ALLOWED
```

Now we have removed line 30 and re-entered the rule as line 15 in the ACL.

SEC 13.12 one more telnet test, plus absolute ranges

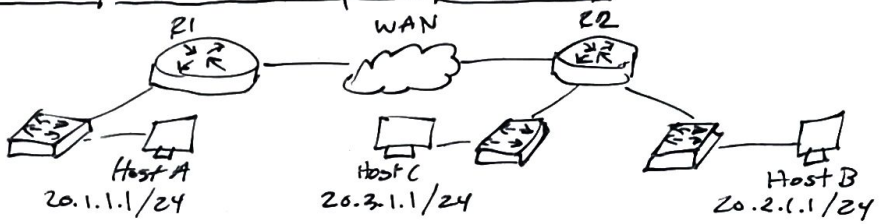
To change system time:

```
clock set HH:MM:SS Day Month Year
```

note that month is spelled out, not a number.

Absolute times have the same time format as clock setting. If no end time is given, the rule will stay active forever once triggered (or until it is removed).

SEC 13.13 ACL: where to put it, and why



- Apply Extended ACL To incoming packets at the interface closest to the packet source.
- Standard ACLs are generally placed as close to the destination as possible.