Lecture (07)
Standard Access list

By:

Dr. Ahmed ElShafee

Introduction

- Access Control Lists can inspect the traffic based on different protocols and criteria.
- Types:
  - IP-based ACLs.
  - MAC-based ACLs,
- Have you noticed how your hand luggage is being scanned at the airport?
- ACL is similar to such scanner only used on the router.
- It can look at the content of the packet traversing it and check the content of the packet up to the layer 4 (extended ACL).
- as an administrator, get to decide what the action is going to be if the packet matches your criteria

- A few applications of ACLs are as follows:
  - ACLs can filter the packets that traverse the router in order to drop the unwanted traffic.
  - ACLs can deny SSH or Telnet traffic to vty lines (router/switch remote management).
  - ACLs are used as to match an interesting traffic to trigger VPN tunnel establishment and encrypt data.
  - ACLs are commonly used in Quality of Service to prioritize certain applications or traffic flows over others or provide different treatment to a certain stream of packets.
  - ACLs are used to filter inbound or outbound dynamic protocol advertisements.

IP Based ACLs

- You can use two major IP-based ACLs
  - Standard ACLs (numbers: 1 through 99)
  - Extended ACLs (numbers: 100 through 199)
  - Named ACL (standard or extended). Named ACLs offer more flexibility in terms of modifying ACLs 'on the fly.
- The major differences between them are the amount and type of criteria we can use to inspect the packets as well as the syntax used to create the statement.

ACL Guidelines

- ACLs use the **top-down processing**. This means that statements are being processed from the one listed on the top of the list first. If the statement is a successful match (permit or deny), the remaining entries listed below this matching statement are NOT inspected anymore.
- The ACL number will determine whether it is **IP standard ACL** (numbers **1-99**) or **IP extended ACL** (numbers **100-199**).
- **Standard ACL** can only inspect the **source IP** of the packet.
- **Extended ACL** will inspect **both source and destination IP**. In addition to these it can match on **layer 4 protocols** (TCP, UDP, OSPF, EIGRP etc.) and even the **layer 4 port numbers** (either source or destination or both).

- The **standard ACLs** should be placed **close to the destination**, **extended ACLs** should be placed **close to the source** of the transmission.
- There is an **IMPLICIT DENY ALL at the end of all the statements** that you create. This means, that if your statements have been created to deny traffic, there must be at least a single permit statements. Otherwise all traffic crossing the interface where ACL has been configured will be denied (filtered out).

- The process of configuring ACLs consist of two steps:
  - Configuring the ACL statements in the global configuration mode.
  - Applying the ACLs on the interfaces to inbound or outbound traffic.

Standard Access Lists

- Standard ACL offers you only a single criterion to single packets out from the flows a router handles.
- It is a source IP address.
- Based on this criterion, a router determines if the packet should be forwarded or dropped.
- This type of ACL does not check if IP carries TCP, UDP, OSPF, EIGRP etc.
- Based on the source IP address the “whole” IP packet (irrespective of the layer 4-5 content) will be forwarded or dropped.

- Syntax

```
access-list access-list-number {permit|deny}
{host|source source-wildcard|any}
```

- **access-list** - the ACL keyword that is followed by a number of argument
- *access-list-number* - IP standard ACLs use the numbers in the range 1-99
- **permit|deny** - what is in braces '{}' are possible options; here permit or deny
- *host*|*source* - again in braces '{}' are possible options; here either a host address or other source such as network or subnet.
- *source-wildcard*|**any** - this is the inversed network mask ('1' becomes '0' and vice versa).

- The next step in configuring an ACL (standard or Extended) is to apply it on the interface in either inbound or outbound direction.

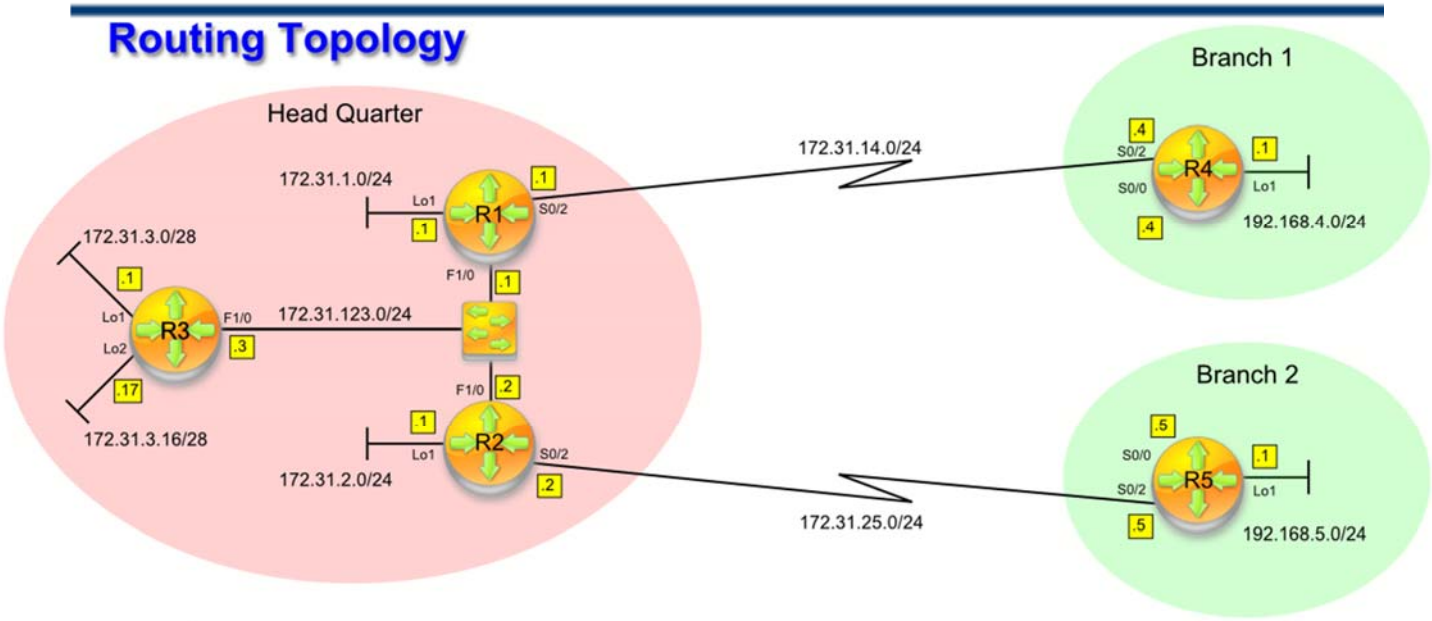
```
interface <interface>
ip access-group number {in|out}
```

- **interface** - the keyword to enter the interface context (must be in the 'config mode')
  <*interface*> - type/number of the interface (e.g. interface **Fa0/0**)
  ip access-group - the keyword that applies an ACL on the interface
  number - the number of access-list configured in the 'config' mode (standard ACL use range 1-99)

  in|**out** - the direction inbound or outbound (how packets are going to be processed)

- **INBOUND ACLs**
  This type of ACL analyzes the packets coming towards the router (the interface where packet was received on). Based on the criteria defined in the ACL, the packet will further be processed (layer 3 lookup performed trying to find the outbound interface), or dropped.

- **OUTBOUND ACLs**
  If you apply the ACL as 'out' the incoming interface does NOT compare the packet content with the ACL statements. It performs a layer 3 lookup immediately. Once the outbound interface is found, and the ACL is applied there as 'out' it analyzes the ACL statement one by one (top-down). Once the match is found (permit or deny) the packet is or is not sent out that interface.



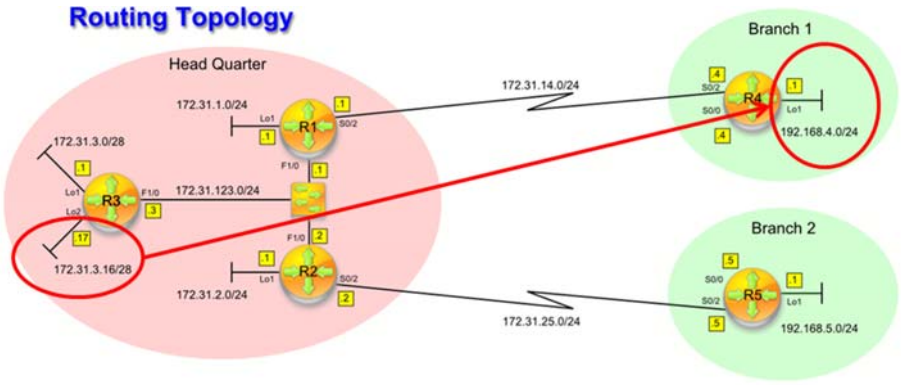
- **Task 1**
- Configure an IP standard ACL that denies packets coming from 172.31.3.16/28 going towards 192.168.4.0/24. All other traffic should be allowed.

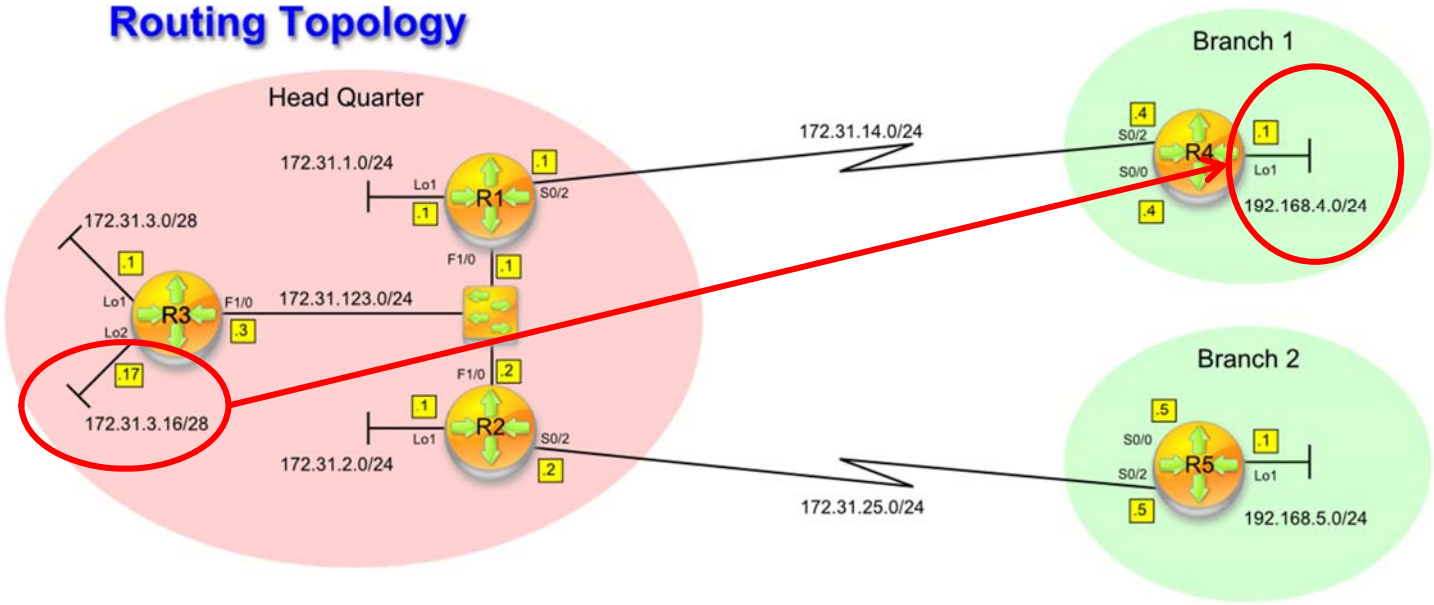
- **Task 2**
- Configure an IP standard ACL that denies packets coming from the host 172.31.123.3 going towards 192.168.5.0/24. Traffic from other sources should be allowed.

- When it comes to configuring ACLs my work flow goes like this:
- Which ACL is going to accomplish my goal: Standard or Extended?
- Which router and interface should I apply the ACL on?
- Which direction should I use: in or out?

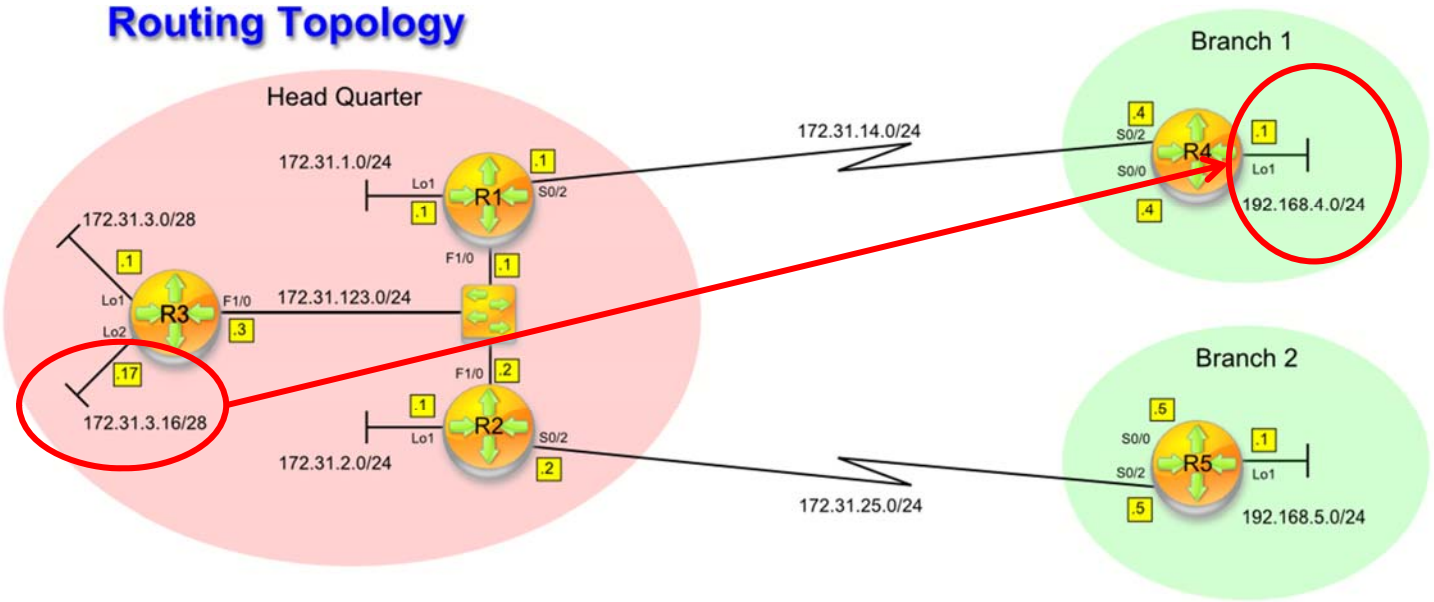
Task 1

- the answer to the first question is obvious: standard ACL must be used.
- In real life examples, the goals you try to accomplish will impose the criteria.
- If you must filter out some specific TCP traffic (e.g. going towards port 80), an extended ACL must be used as the standard one cannot filter on TCP (source IP only).





- The guidelines specify that standard ACL must be placed as close to the destination as possible.
- If I applied the ACL in Task 1 on **R3** F1/0 out, the packet with the source 172.31.3.32/28 could not go anywhere out that interface. We're supposed to filter this source going towards Branch 1 (R4) and not anywhere else. **R4** is going to be the router I'm going to apply the ACL on.



- As for the last question about the direction, I could use **R4**'s S0/2 inbound or F1/0 outbound. Since there are only two interfaces, I can apply this on s0/2 inbound and it won't make much difference except that the packet will be rejected on the inbound interface. This way, R4 won't have to do layer 3 lookup and waste it resources. If there were more interfaces I would use it on F1/0 interface outbound since the traffic should not be sent to the specific network (192.168.4.0).

- *Configure an IP standard ACL that denies packets coming from 172.31.3.16/28 going towards 192.168.4.0/24. All other traffic should be allowed.*

```
File  Edit  View  Terminal  Help

R3#ping 192.168.4.1 source 172.31.3.17

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
Packet sent with a source address of 172.31.3.17
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/12 ms
R3#
```


R4 Configuration:

```
!
! Step 1 - Create an ACL statement in the global config
R4(config)#access-list 1 deny 172.31.3.16 0.0.0.15
R4(config)#access-list 1 permit any
R4(config)#
!
! Step 2 - Apply the ACL on the interface
R4(config)#int s0/2
R4(config-if)#ip access-group 1 in
R4(config-if)#
!
```

- **access-list 1** - the number 1 implies the standard ACL (1-99)
- **deny** - the packets meeting the criterion that follows will be denied (dropped)
- **172.31.3.16** - the source IP criterion
- **0.0.0.15** - the wildcard mask which is the subnet's inversed network mask
- The subnet's network mask is: 255.255.255.240. In the binary it looks like:
- 11111111.11111111.11111111.11110000
- If we inverse this network mask we get:
- 00000000.00000000.00000000.00001111
- That, converted to decimal is: 0.0.0.15

- **permit any** - this permits all other source IP. The word '**any**' is the alias for:
- **0.0.0.0 255.255.255.255**

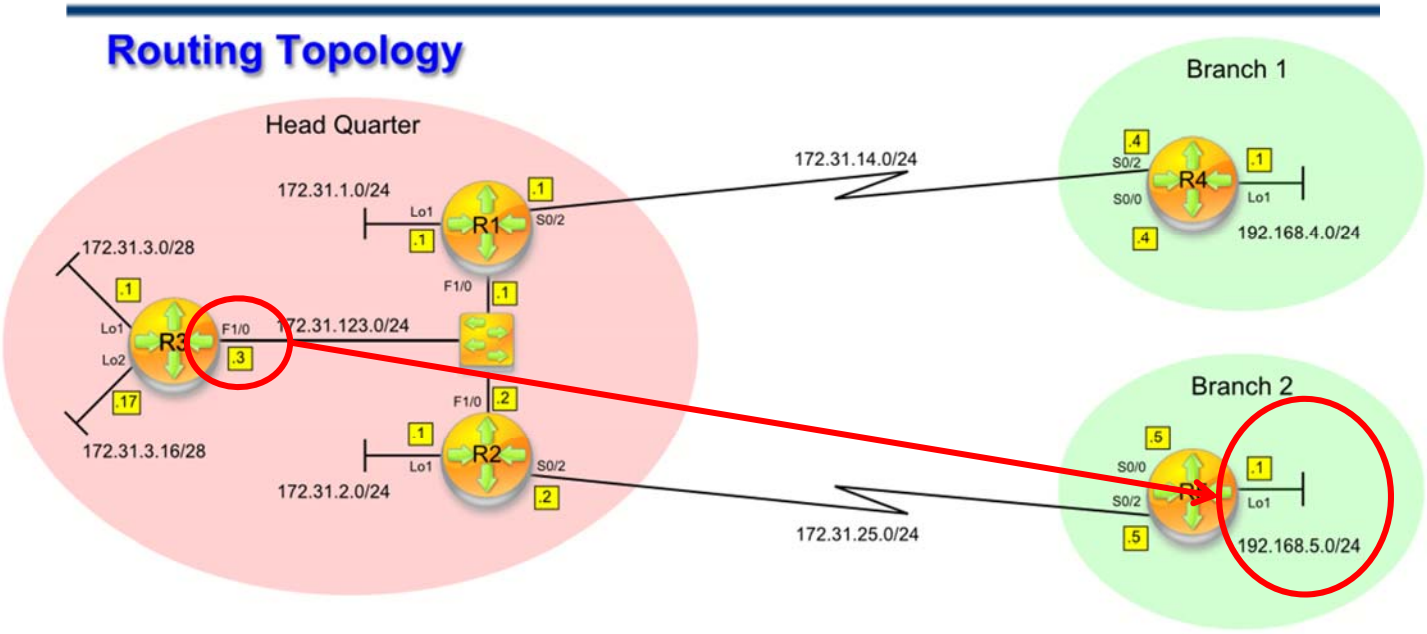
- **Verification**

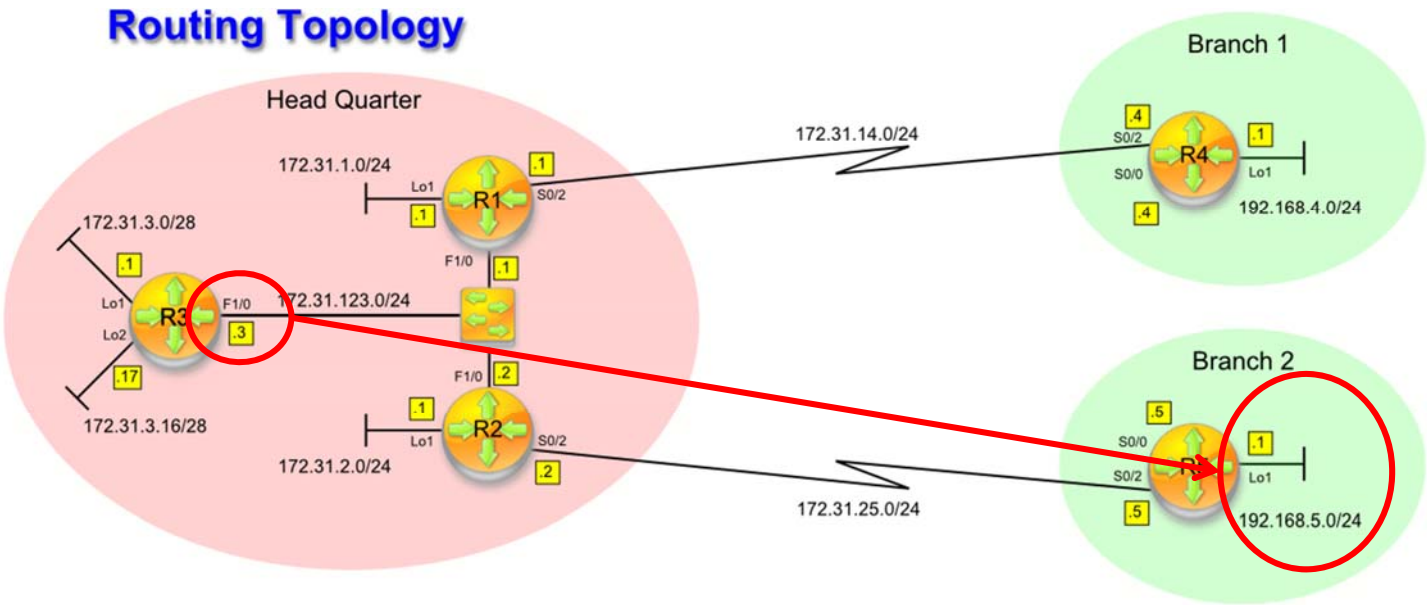
```
File  Edit  View  Terminal  Help

R3#ping 192.168.4.1 source 172.31.3.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
Packet sent with a source address of 172.31.3.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/8 ms
R3#
R3#ping 192.168.4.1 source 172.31.3.17

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
Packet sent with a source address of 172.31.3.17
U.U.U
Success rate is 0 percent (0/5)
R3#
```

- Only packets sourced from the 172.31.3.16/24 are being blocked on R4. The packets from 172.31.3.0/28 are getting through.

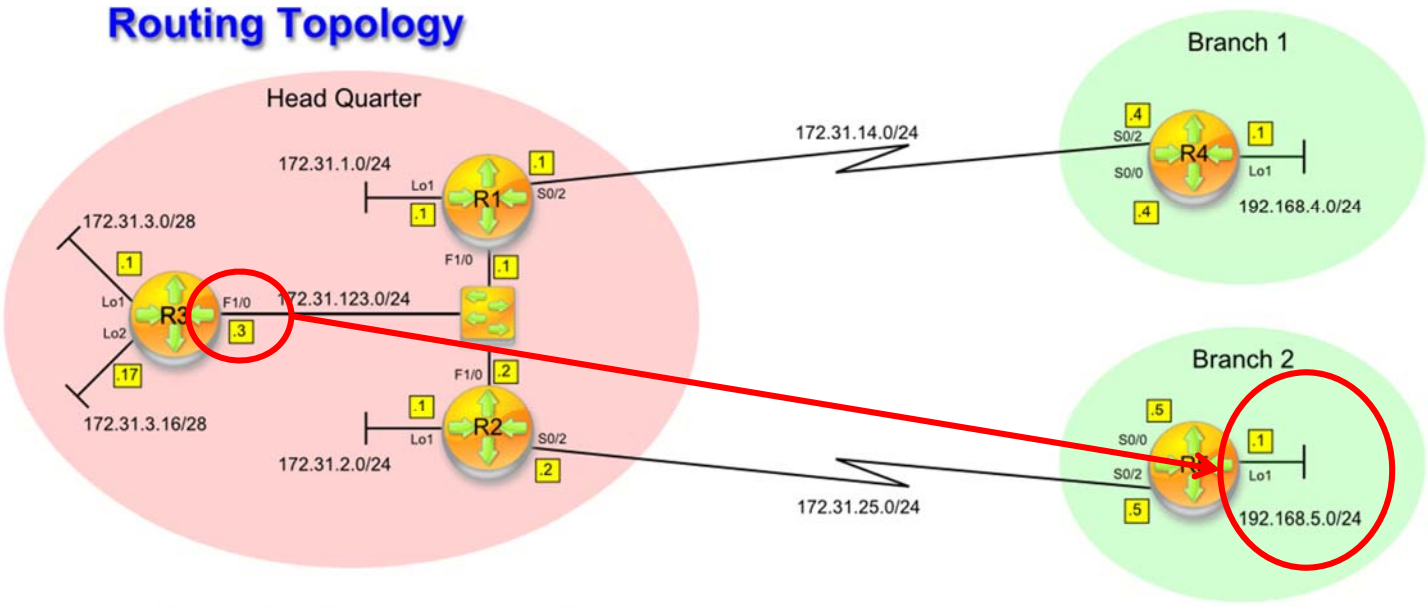
```
File  Edit  View  Terminal  Help

R4#show access-list
Standard IP access list 1
    10 deny   172.31.3.16, wildcard bits 0.0.0.15 (11 matches)
    20 permit any (240 matches)
R4#
```

Task 2

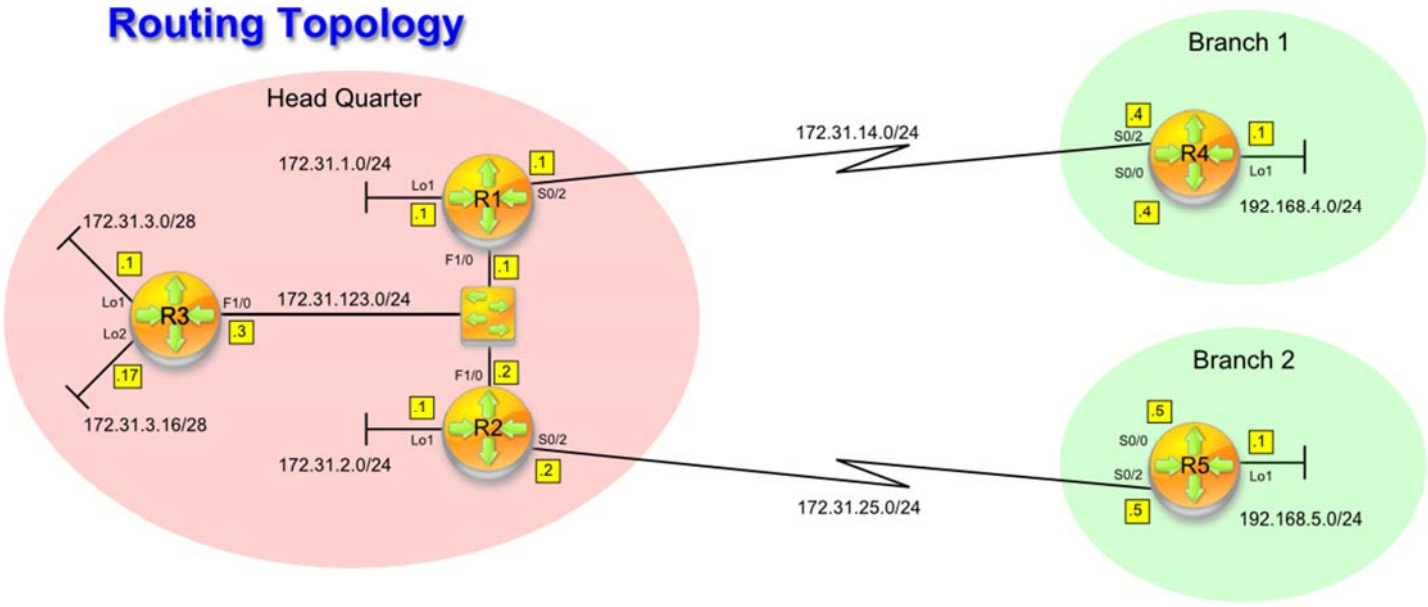


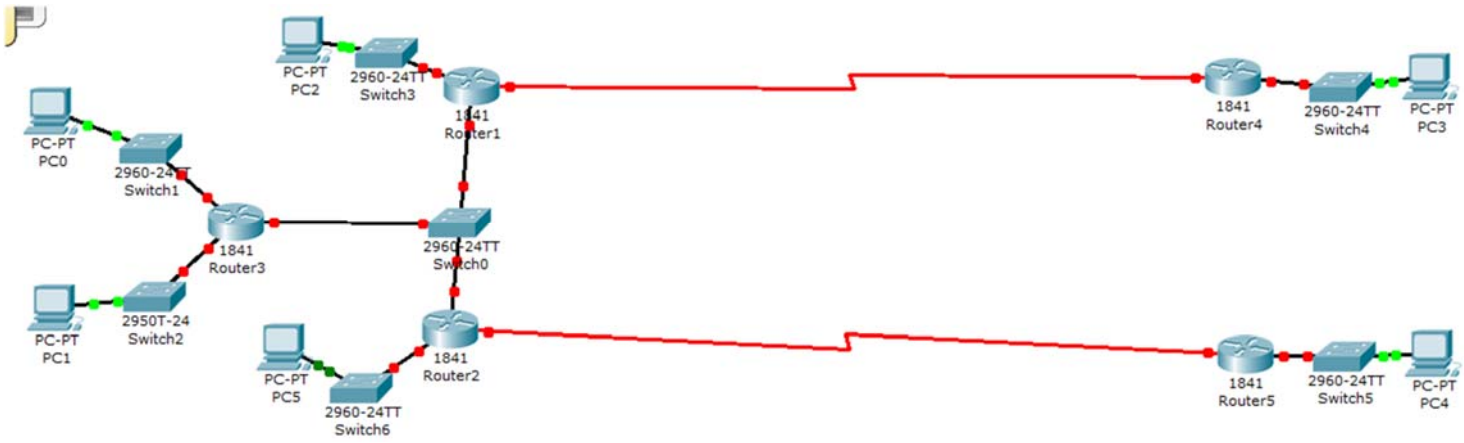


- !
- ! *Step 1 - Create an ACL statement in the global config*
- R4(config)#access-list 2 deny 172.31.123.3 0.0.0.7.
  R4(config)#access-list 2 permit any
  R4(config)#
- !



- ! *Step 2 - Apply the ACL on the interface*
- R4(config)#int S0/2
  R4(config-if)#ip access-group 2 in
  R4(config-if)#
- !
-





PCS configuration

****

PC0	**172.31.3.2**	**255.255.255.240**	**172.31.3.1**	**fa0/1**
PC1	**172.31.3.18**	**255.255.255.240**	**172.31.3.17**	**fa0/1**
PC2	**172.31.1.2**	**255.255.255.240**	**172.31.1.1**	**fa0/1**
PC3	**172.31.4.2**	**255.255.255.240**	**172.31.4.1**	**fa0/1**
PC1	**172.31.5.2**	**255.255.255.240**	**172.31.5.1**	**fa0/1**
PC5	**172.31.2.2**	**255.255.255.240**	**172.31.2.1**	**fa0/1**

```
Router 3
******
en
config t
interface fa0/0
ip address 172.31.123.3 255.255.255.0
no sh

interface fa0/0/0
ip address 172.31.3.1 255.255.255.240
no sh

interface fa0/0/1
ip address 172.31.3.17 255.255.255.240
no sh

ip route 0.0.0.0 0.0.0.0 fa0/0

end

copy running-config startup-config

exit
```



```
Router 1
******
en
config t
interface fa0/0
ip address 172.31.123.1 255.255.255.0
no sh

interface fa0/1
ip address 172.31.1.1 255.255.255.0
no sh

interface se0/0/0
ip address 172.31.14.1 255.255.255.0
clock rate 19200
no sh

ip route 172.31.3.0 255.255.255.0 fa0/0
ip route 172.31.2.0 255.255.255.0 fa0/0
ip route 172.31.25.0 255.255.255.0 fa0/0
ip route 192.168.5.0 255.255.255.0 fa0/0
ip route 192.168.4.0 255.255.255.0 se0/0/0

end

copy running-config startup-config

exit
```

```
Router 2
******
en
config t
interface fa0/0
ip address 172.31.123.2 255.255.255.0
no sh

interface fa0/1
ip address 172.31.2.1 255.255.255.0
no sh

interface se0/0/0
ip address 172.31.25.2 255.255.255.0
clock rate 19200
no sh

ip route 172.31.3.0 255.255.255.0 fa0/0
ip route 172.31.1.0 255.255.255.0 fa0/0
ip route 172.31.14.0 255.255.255.0 fa0/0
ip route 192.168.5.0 255.255.255.0 se0/0/0
ip route 192.168.4.0 255.255.255.0 fa0/0

end

copy running-config startup-config

exit
```



```
Router 4
******
en
config t
interface fa0/0
ip address 192.168.4.1 255.255.255.0
no sh

interface se0/0/0
ip address 172.31.14.4 255.255.255.0
no sh

ip route 0.0.0.0 0.0.0.0 se0/0/0

end

copy running-config startup-config

exit
```

```
Router 5
******
en
config t
interface fa0/0
ip address 192.168.5.1 255.255.255.0
no sh

interface se0/0/0
ip address 172.31.25.5 255.255.255.0
no sh

ip route 0.0.0.0 0.0.0.0 se0/0/0

end

copy running-config startup-config

exit
```


ASL configuration

```
Router 04
*******
enable
Config t
access-list 1 deny 172.31.3.16 0.0.0.15
access-list 1 permit any
end
int se0/0/0
ip access-group 1 in
copy running-config startup-config

exit
```

```
Router 05
*******
enable
Config t
access-list 1 deny 172.31.123.3 0.0.0.255
access-list 1 permit any
end
int fe0/0/0
ip access-group 1 in
copy running-config startup-config

exit
```


Thanks,..
See you next week (ISA),…