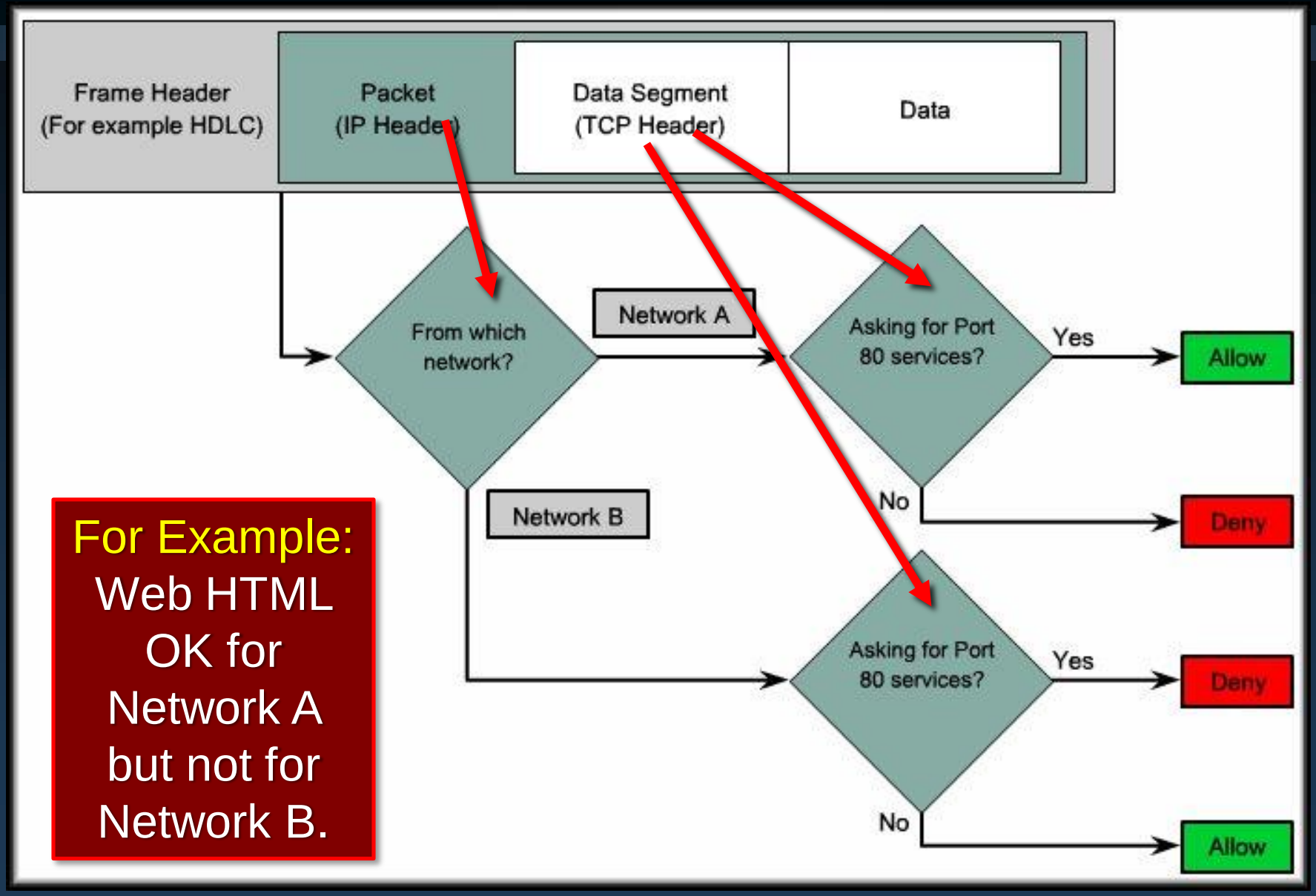
Access Control Lists
(ACLs)

Using ACLs to Secure Networks

- ACLs enable you to control traffic into and out of your network.
  - Can be as simple as permitting or denying network hosts or addresses.
  - Or to control network traffic based on the TCP port being used.

Using ACLs to Secure Networks



For Example:
Web HTML OK for Network A but not for Network B.

What is an ACL?

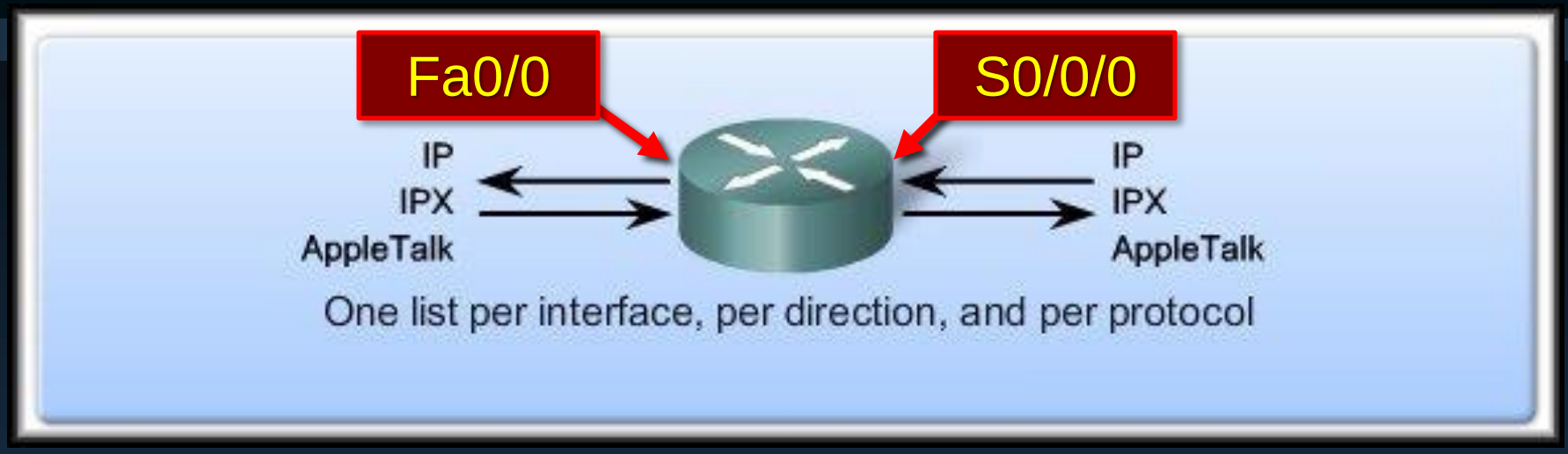
- An Access Control List (ACL) is:
  - A sequential list of permit or deny statements.
    - Apply to IP addresses (Layer 3 header)
    - Apply to upper-layer protocols (Layer 4 header).
  - Controls whether a router permits or denies packets to pass through the router.

- By default, a router does not have any ACLs.

The Three P's

- One ACL per protocol:
  - An ACL must be defined for each protocol enabled on the interface.
- One ACL per direction:
  - ACLs control traffic in one direction at a time on an interface.
    - Two separate ACLs must be created to control:
      - Inbound Traffic: Traffic coming into the interface.
      - Outbound Traffic: Traffic leaving an interface.
- One ACL per interface:
  - ACLs control traffic for an interface (Fa0/0, s0/0/0).

The Three P's

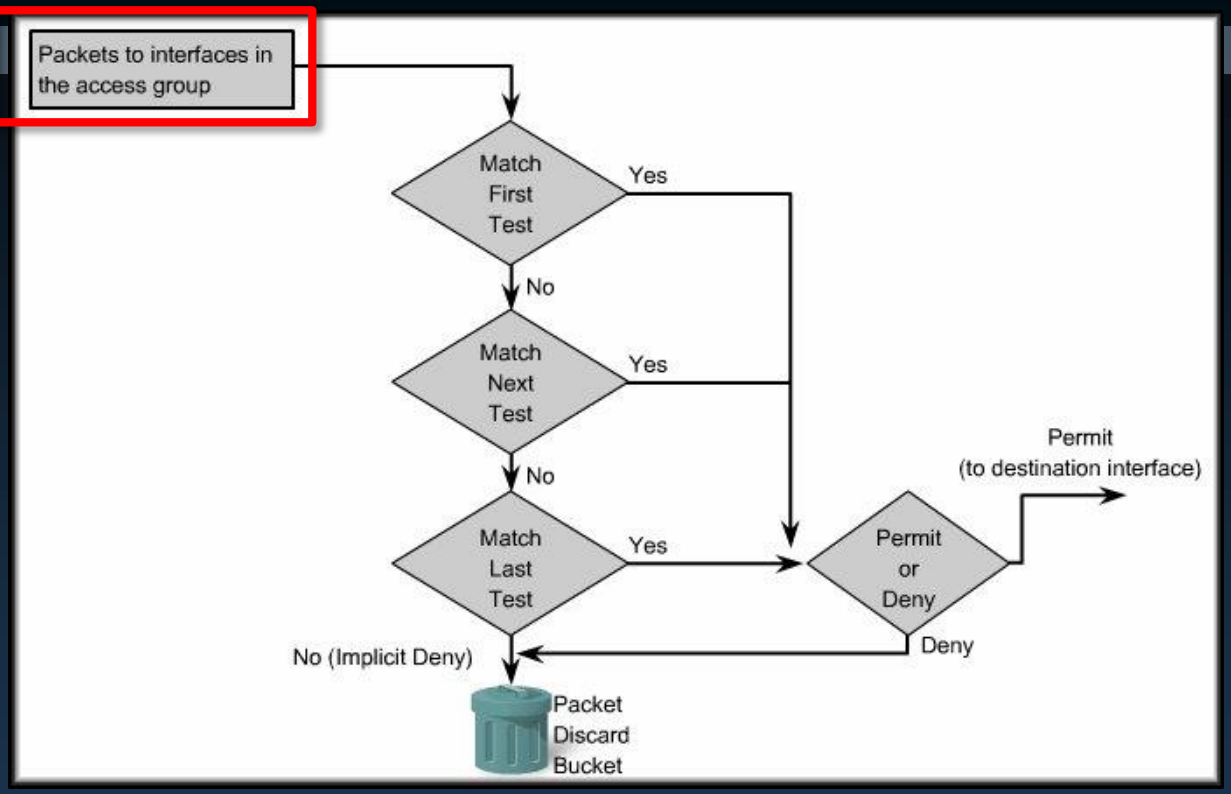


- One Access Control List per protocol.
- One Access Control List per direction.
- One Access Control List per interface.

- How many possible ACLs?
  - 3 protocols X 2 directions X 2 ports
  - Possibility of 12 separate lists.
    - *Note that the same list can be used on multiple interfaces.*

How ACLs Work

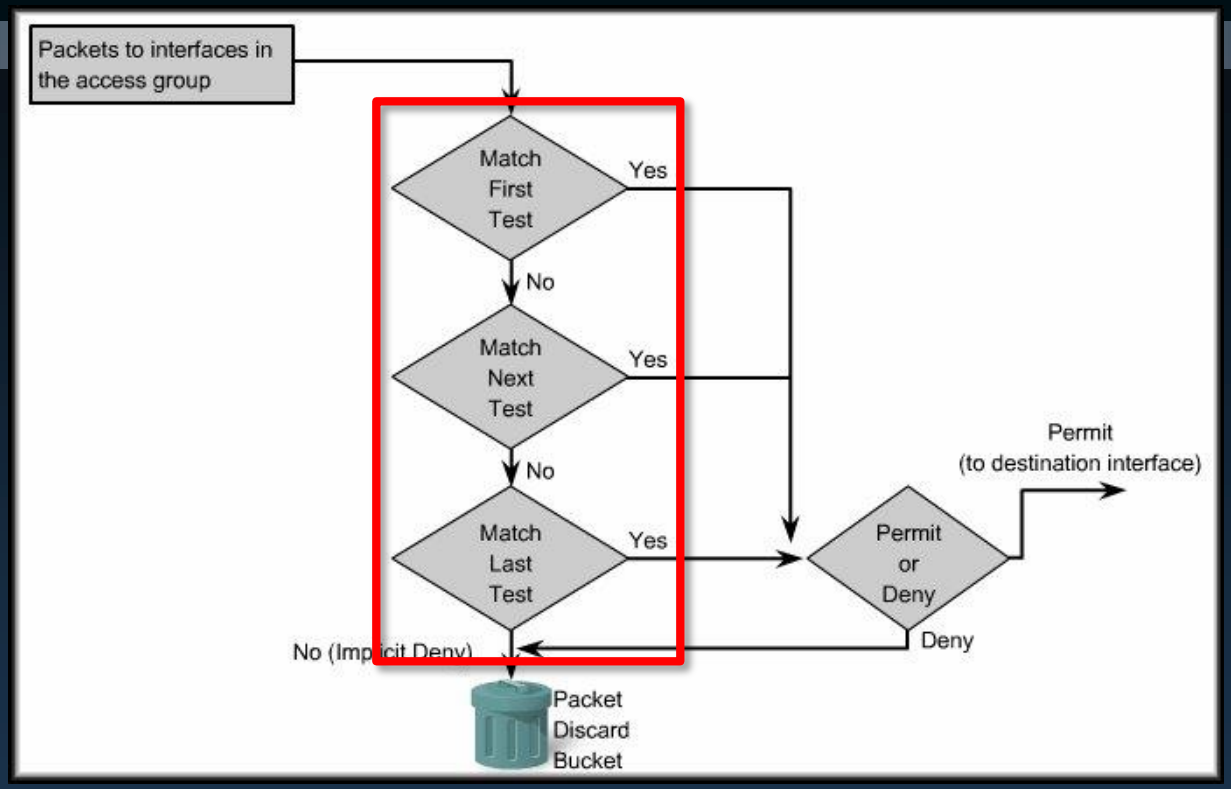
Inbound ACL



- We have to assign the list of ACLs to the interface and specify the direction of the traffic to be checked.

How ACLs Work

Inbound ACL

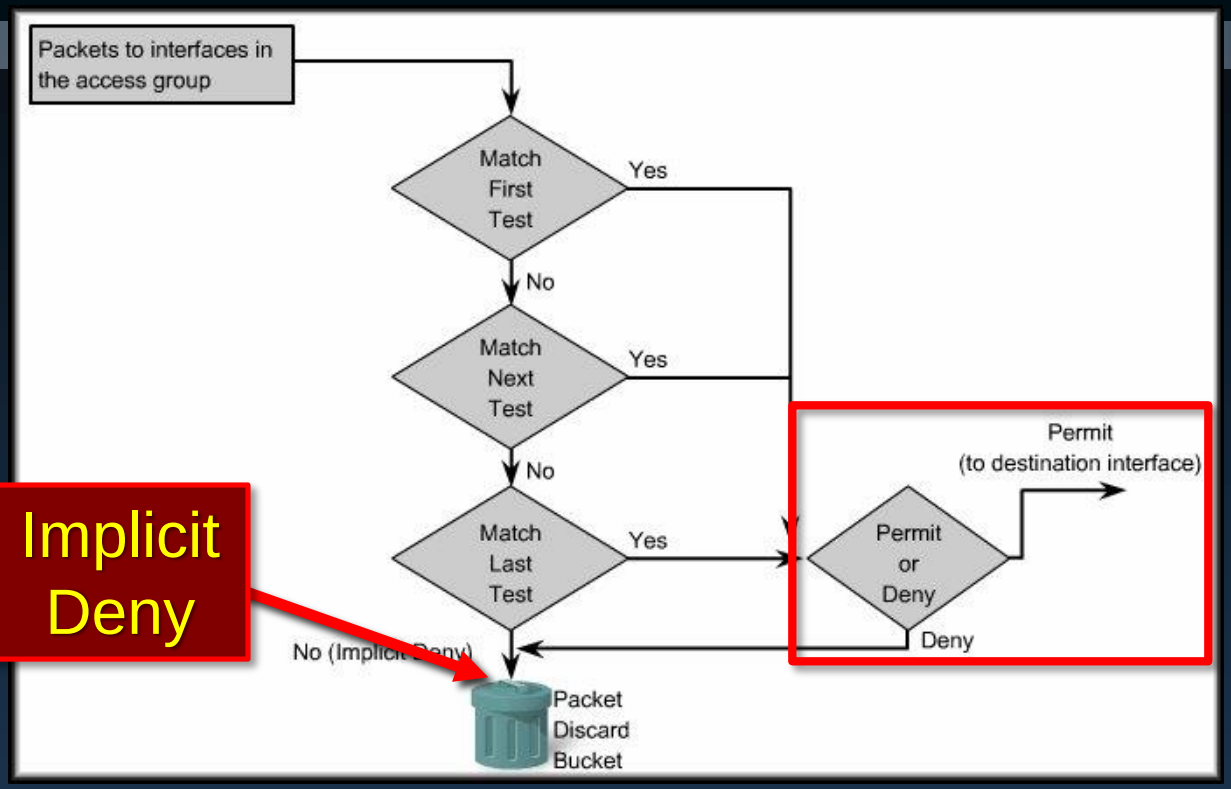


- ACL statements are processed in a sequential, logical order.
- The logic used to create the list and the order of the list items is very important.

How ACLs Work

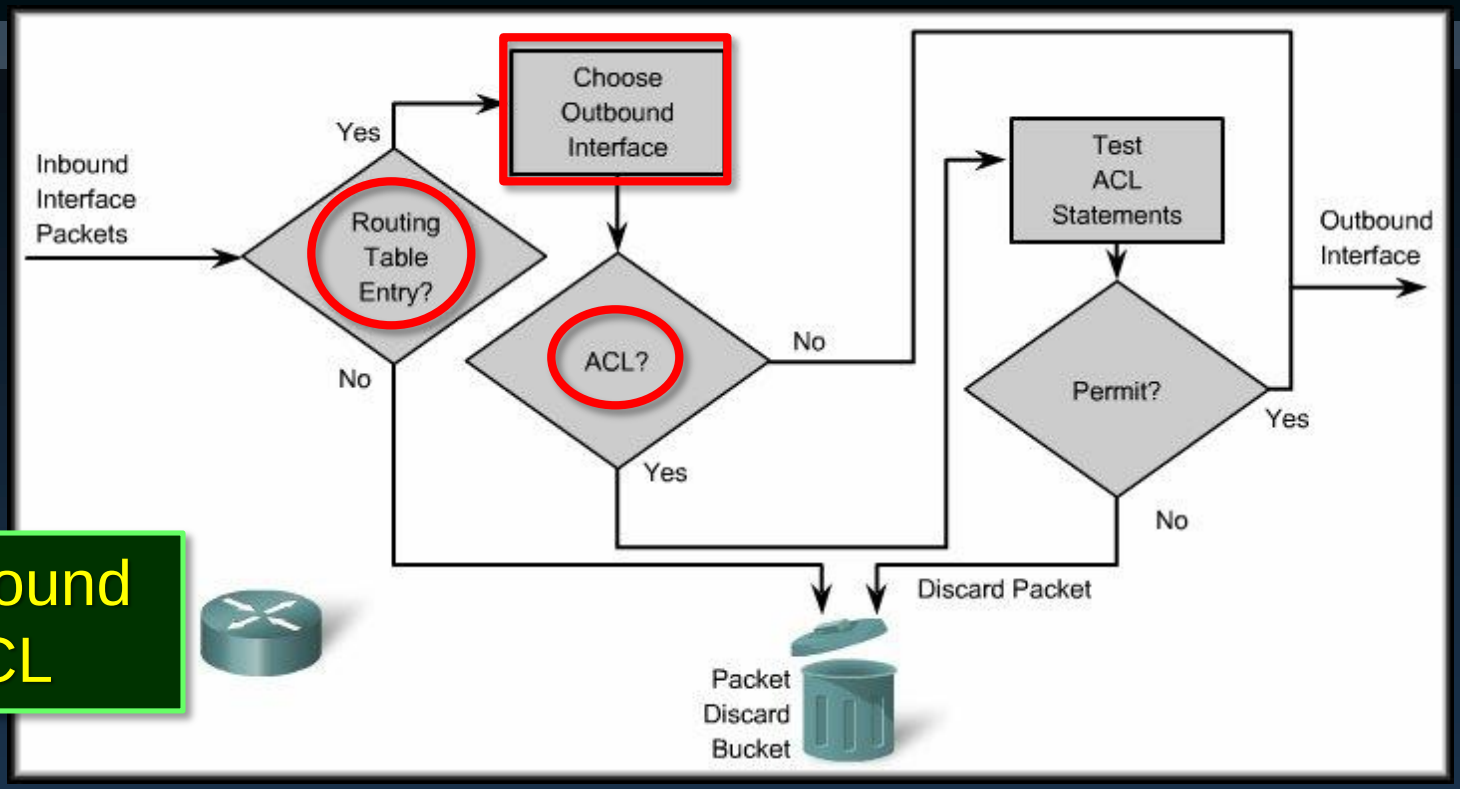
Inbound ACL

Implicit Deny



- If a condition match is true, the packet is permitted or denied and the rest of the ACL statements are not checked.
- If all the ACL statements are unmatched, an implicit deny any statement is placed at the end of the list by default.

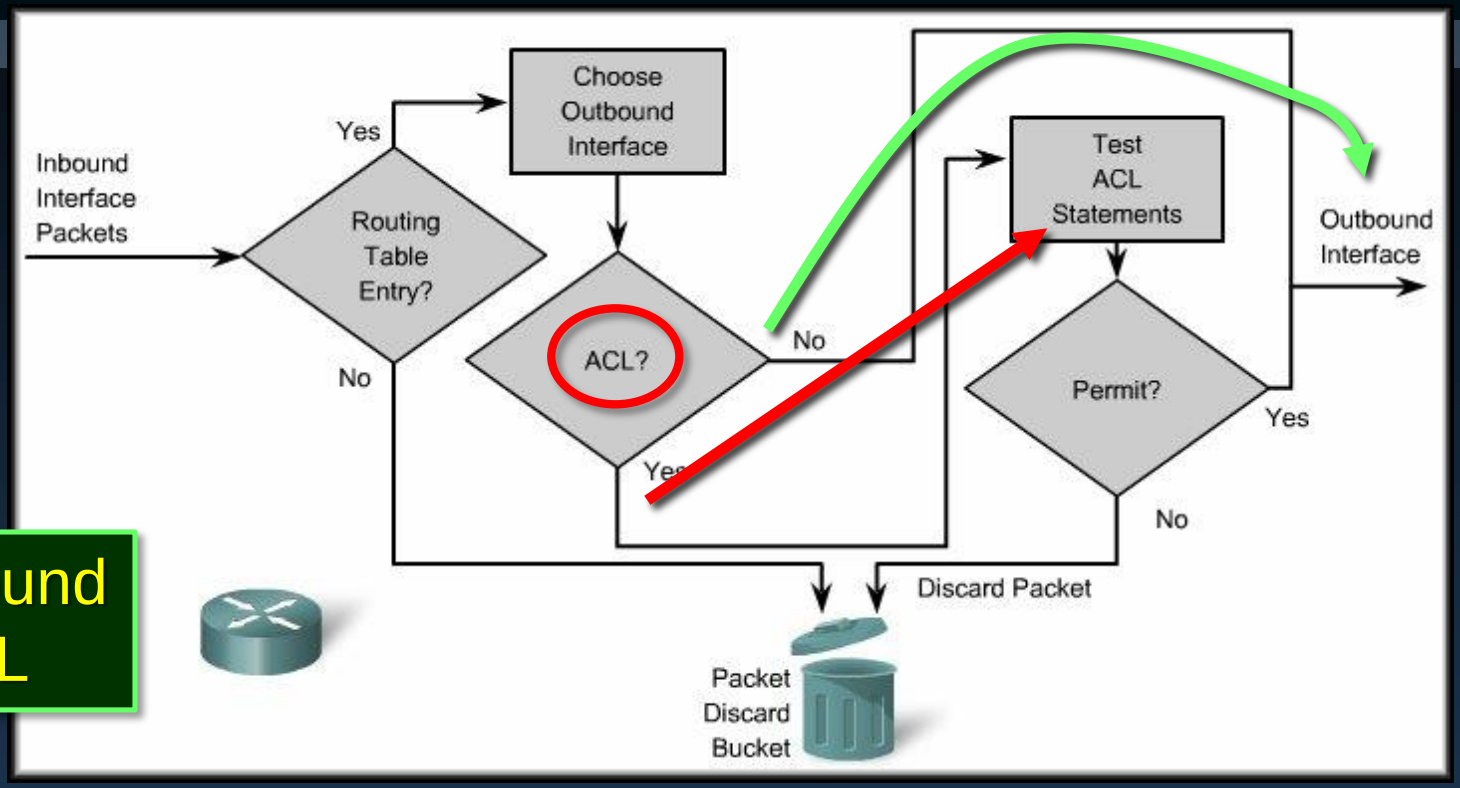
How ACLs Work



Outbound ACL

- Before a packet is forwarded to an outbound interface, the router checks the routing table.
- Next, the router checks to see whether the outbound interface is assigned to an ACL List.

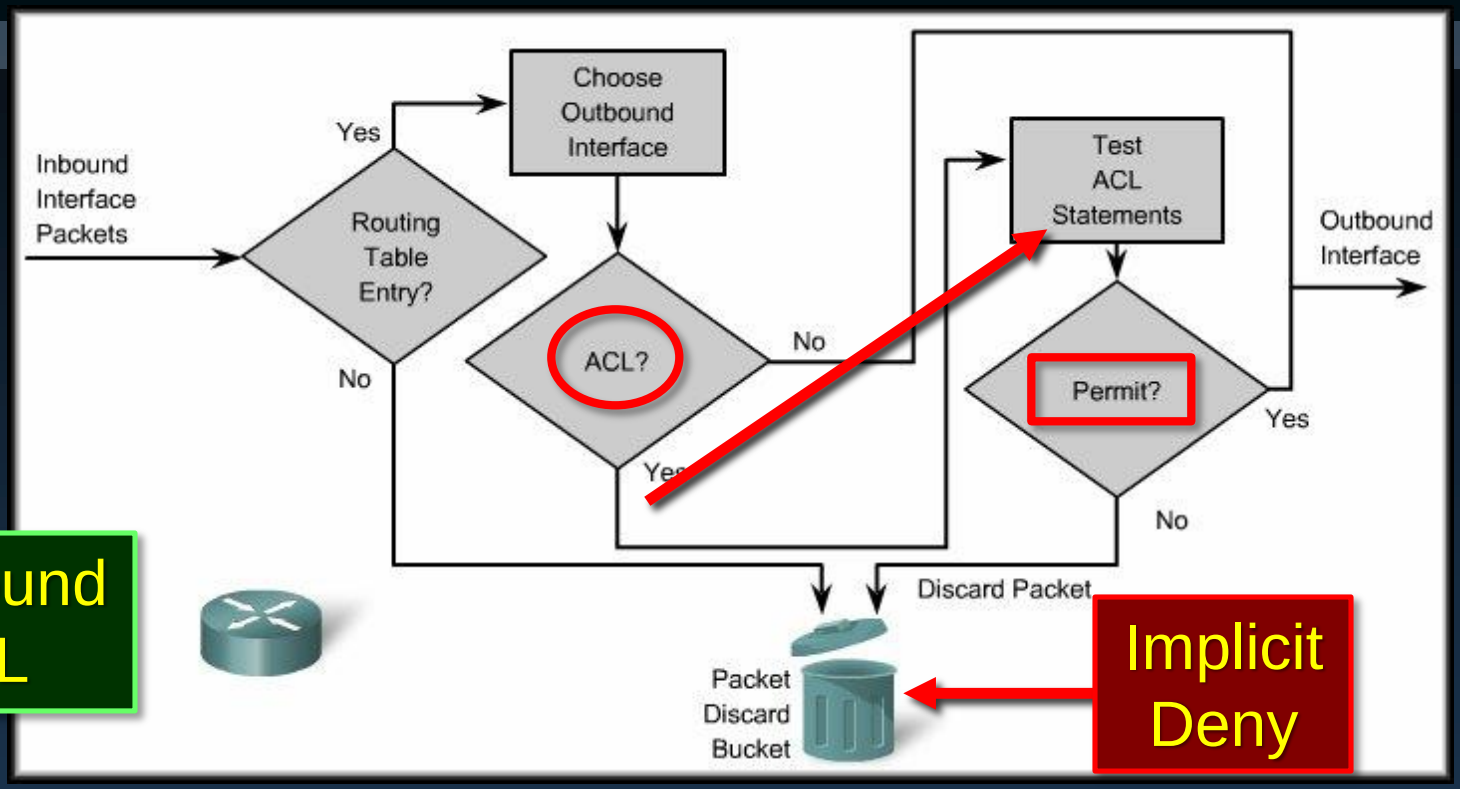
How ACLs Work



Outbound ACL

- If no ACL is present, the packet is forwarded out the interface.
- If an ACL is present, the packet is tested by the combination of ACL statements that are associated with that interface.

How ACLs Work



- The packet is either permitted (sent to the outbound interface) or denied (dropped).
- If the packet does not meet any of the criteria, it is dropped (Implicit Deny).

How ACLs Work

- Access list statements operate in **sequential, logical order**.
- They evaluate packets from the **top - down**.
- Once there is an **access list statement match**, the router skips the rest of the statements.
- If a condition **match is true**, the packet is permitted or denied.
- There can be **only one** access list **per protocol, per interface**.
- There is an **implicit deny any** at the end of every access list.

  - *ACLs do not block packets that originate within the router.* (i.e. pings, telnets, ssh, etc.)

Types of Cisco ACLs

- Two types:
  - Standard ACLs:
    - Standard ACLs allow you to permit or deny traffic based on the source IP addresses.
    - The destination of the packet and the ports involved do not matter.

```
access-list 10 permit 192.168.30.0 0.0.0.255
```

    - Permit all traffic from network 192.168.30.0/24 network.
    - Because of the implied "deny any" at the end, all other traffic is blocked with this ACL.

Types of Cisco ACLs

- Two types:
  - Extended ACLs:
    - Extended ACLs filter IP packets based on several attributes;
      - Protocol type, source and/or destination IP address, source and/or destination TCP or UDP ports.

```
access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80
```

      - Permits traffic originating from any address on the 192.168.30.0/24 network to any destination host port 80 (HTTP).

Numbering and Naming ACLs

Numbered ACL:

You assign a number based on which protocol you want filtered:

- (1 to 99) and (1300 to 1999): Standard IP ACL
- (100 to 199) and (2000 to 2699): Extended IP ACL

- Using numbered ACLs is an effective method for determining the ACL type on smaller networks with more homogeneously defined traffic.

Numbering and Naming ACLs

Numbered ACL:

You assign a number based on which protocol you want filtered:

- (1 to 99) and (1300 to 1999): Standard IP ACL
- (100 to 199) and (2000 to 2699): Extended IP ACL

- When configuring ACLs on a router, each ACL must be uniquely identified by assigning a number.

One group numbered 8

```
access list 8 permit…
access list 8 permit…
access list 8 permit…
access list 8 permit…
```

Multiple groups

```
access list 1 permit…
access list 2 permit…
access list 3 permit…
access list 4 permit…
```

Numbering and Naming ACLs

Named ACL:

You assign a name by providing the name of the ACL:

- Names can contain alphanumeric characters.
- It is suggested that the name be written in CAPITAL LETTERS.
- Names cannot contain spaces or punctuation and must begin with a letter.
- You can add or delete entries within the ACL.

- Using named ACLs:
  - A numbered ACL does not tell you the purpose of the list.
  - Starting with Cisco IOS Release 11.2, you can use a name to identify a Cisco ACL.

Where to Place ACLs

- ACLs can act as firewalls to filter packets and eliminate unwanted traffic.
  - Every ACL should be placed where it has the greatest impact on efficiency.
  - The basic rules are:
    - Standard ACLs do not specify a destination address. Place them as close to the destination as possible.
    - Extended ACLs are located as close as possible to the source of the traffic denied.
      - Undesirable traffic is filtered without crossing the network infrastructure.

Access Control Lists

Configuring Standard ACLS

Configuring a Standard ACL

- To configure a standard ACL you must:
  - Create the standard ACL
  - Activate the ACL on an interface.
    - The **`access-list`** global configuration command defines a standard ACL with a number in the range of 1 to 99 or 1300 to 1399.

```
Router(config)#access-list access-list-number
                           [deny | permit | remark]
                           source [source-wildcard]
                           [log]
```

Configuring a Standard ACL

- For Example:
  - To create a numbered ACL designated 10 that would permit network 192.168.10.0 /24, you would enter:

```
R1(config)#access-list 10 permit 192.168.10.0 0.0.0.255
```

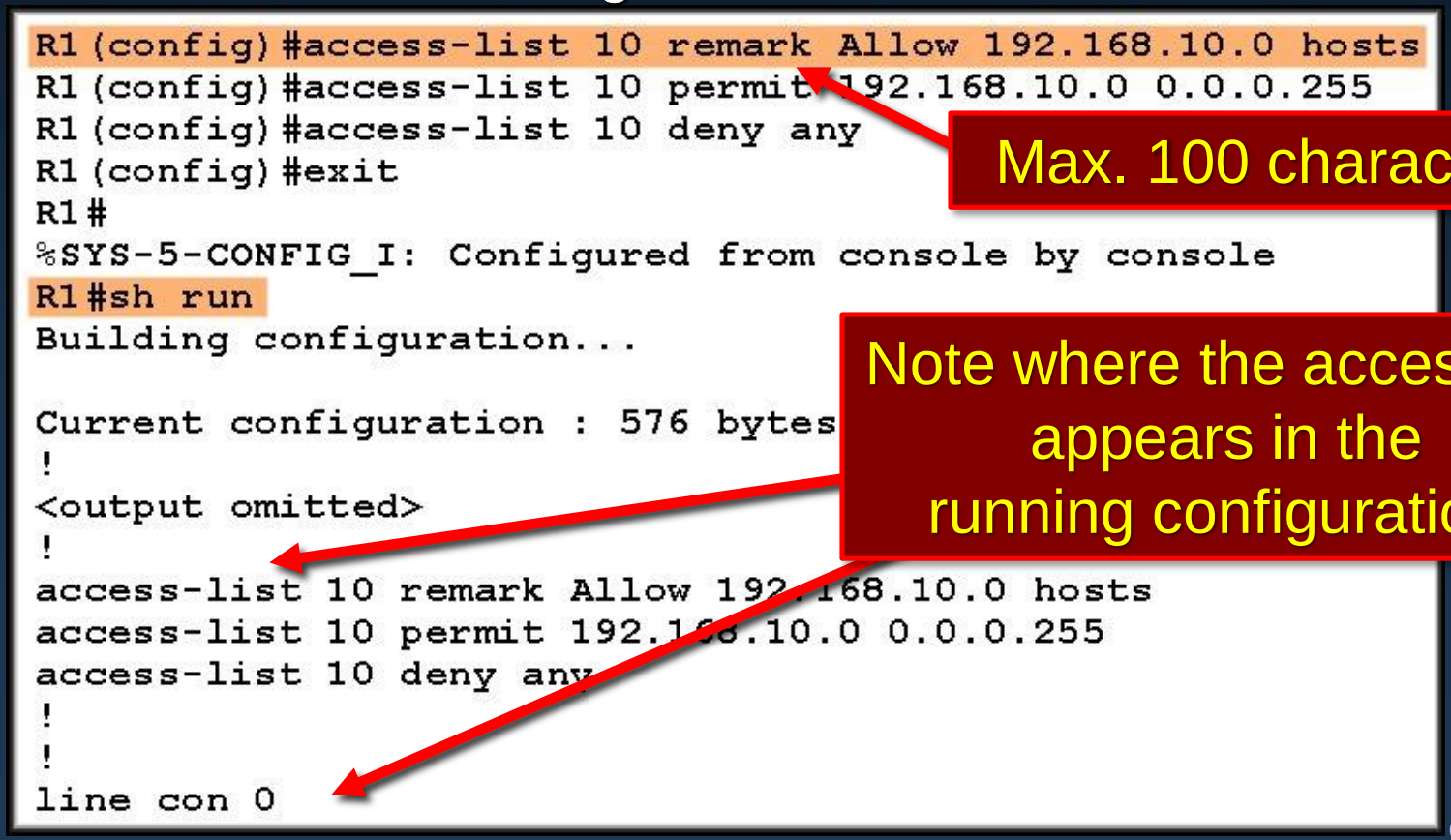
  - To remove an access list, use the no form of the command.

```
R1#show access-list
Standard IP access list 10
    permit 192.168.10.0 0.0.0.255
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#no access-list 10
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console
R1#show access-list
R1#
```



Configuring a Standard ACL

- For Example:
  - The **`remark`** keyword is used for documentation and makes access lists a great deal easier to understand.



ACL Wildcard Masking

- Wildcard Masking:
  - ACLs statements include wildcard masks.
    - (Remember OSPF network entries?)
  - A wildcard mask is a string of binary digits telling the router to check specific parts of the subnet number.
    - The numbers 1 and 0 in the mask identify how to treat the corresponding IP address bits.
  - Wildcard masks are referred to as an inverse mask.
    - Unlike a subnet mask in which binary 1 is equal to a match (network) and binary 0 is not a match (host), the reverse is true.
    - It also does not have to be contiguous 1's and 0's.

ACL Wildcard Masking

- Wildcard Masking:
  - Wildcard masks use the following rules to match binary 1s and 0s:
    - Wildcard mask bit 0:
      - The corresponding bit value in the IP Address to be tested must match the bit value in the address specified in the ACL.
    - Wildcard mask bit 1:
      - Ignore the corresponding bit value.

ACL Wildcard Masking

Checking/Calculating the Wildcard Mask

Network 172.16.32.0 Subnet Mask 255.255.240.0

Subnet Mask	255	.	255	.	240	.	0
plus Wildcard Mask	0	.	0	.	15	.	255
	255	.	255	.	255	.	255

We can calculate the Wildcard Mask using the Subnet Mask.

	255	.	255	.	255	.	255
minus Subnet Mask	255	.	255	.	240	.	0
Wildcard Mask	0	.	0	.	15	.	255

Time for some Practice!

```
RouterB(config)#access-list 10 permit  ?   ?
```

Permit the following networks:		Address / Wildcard Mask
A	172.16.0.0 255.255.0.0	172.16.0.0 0.0.255.255
B	172.16.1.0 255.255.255.0	172.16.1.0 0.0.0.255
C	192.168.1.0 255.255.255.0	192.168.1.0 0.0.0.255
D	172.16.16.0 255.255.240.0	172.16..16.0 0.0.15.255
E	172.16.128.0 255.255.192.0	172.16.128.0 0.0.63.255

Permit the following hosts:		
A	172.16.10.100	172.16.10.100 0.0.0.0
B	192.168.1.100	192.168.1.100 0.0.0.0
C	All hosts	0.0.0.0 255.255.255.255

ACL Wildcard Masking

- Wildcard Masking:

Just this host

	Decimal	Binary
IP Address	192.168.1.1	11000000.10101000.00000001 .00000001
Wildcard Mask	0.0.0.0.	00000000.00000000.00000000.00000000

Any Host

	Decimal	Binary
IP Address	192.168.1.1	11000000.10101000.00000001 .00000001
Wildcard Mask	255.255.255.255	11111111.11111111.11111111.11111111

Subnet Hosts

	Decimal	Binary
IP Address	192.168.1.1	11000000.10101000.00000001 .00000001
Wildcard Mask	0.0.0.255	00000000.00000000.00000000.11111111

ACL Wildcard Masking

- Wildcard Masking:

	Decimal	Binary
IP Address	192.168.16.0	11000000.10101000.00010000.00000000
Wildcard Mask	0.0.15.255	00000000.00000000.00001111.11111111

All IP addresses that have a match in the first 20 bits of the address.

All Subnets 192.168.16.0 to 192.168.31.0

ACL Wildcard Masking

- Wildcard Masking:

	Decimal	Binary
IP Address	192.168.1.0	11000000.10101000.00000001 .00000000
Wildcard Mask	0.0.254.255	**00000000.00000000.11111110.11111111**

All IP addresses that have a match in the first 16 bits of the address and the last bit of the second octet.

All Odd numbered subnets in 192.168.0.0

ACL Wildcard Masking

- Wildcard Bit Mask Keywords:
  - The keywords host and any help identify the most common uses of wildcard masking.
    - host:
      - Used instead of 0.0.0.0 for the wildcard mask (all IP address bits must match).
    - any:
      - Used instead of 255.255.255.255 for the wildcard mask (accept any addresses).

ACL Wildcard Masking

- Wildcard Bit Mask Keywords:

```
access-list 10 permit 172.16.30.2 0.0.0.0
```

OR

```
access-list 10 permit host 172.16.30.2
```

OR

```
access-list 10 permit 172.16.30.2
```

```
access-list 10 deny 0.0.0.0 255.255.255.255
```

OR

```
access-list 10 deny any
```

```
access-list 10 permit 0.0.0.0 255.255.255.255
```

OR

```
access-list 10 permit any
```

Applying Standard ACLs to Interfaces

- You can define ACLs without applying them but they will have no effect until they are applied to the router's interface.
  - Remember……It is a good practice to:
    - Apply the Standard ACLs on the interface *closest to the destination of the traffic*.
    - Apply Extended ACLs on the interface *closest to the source of the traffic*.

Applying Standard ACLs to Interfaces

- Apply the standard ACL to an interface using the following command:

The number or name assigned during the **access-list** configuration.

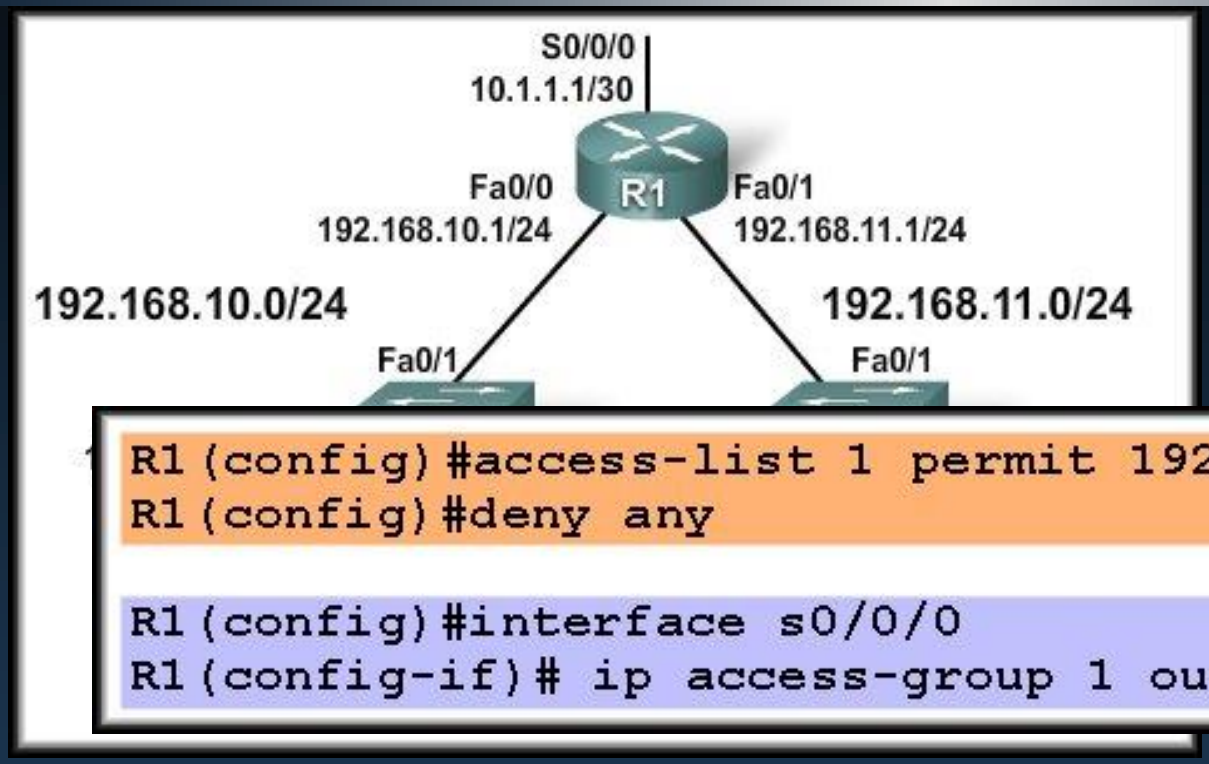
```
R1(config-if)#access-group
              {access-list number | access-list-name}
              {in | out}
```

Consider the traffic from the router's viewpoint.

in: Traffic that is arriving on the interface.

out: Traffic that has already been routed by the router and is leaving the interface.

Applying Standard ACLs to Interfaces




```
R1(config)#access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)#deny any

R1(config)#interface s0/0/0
R1(config-if)# ip access-group 1 out
```

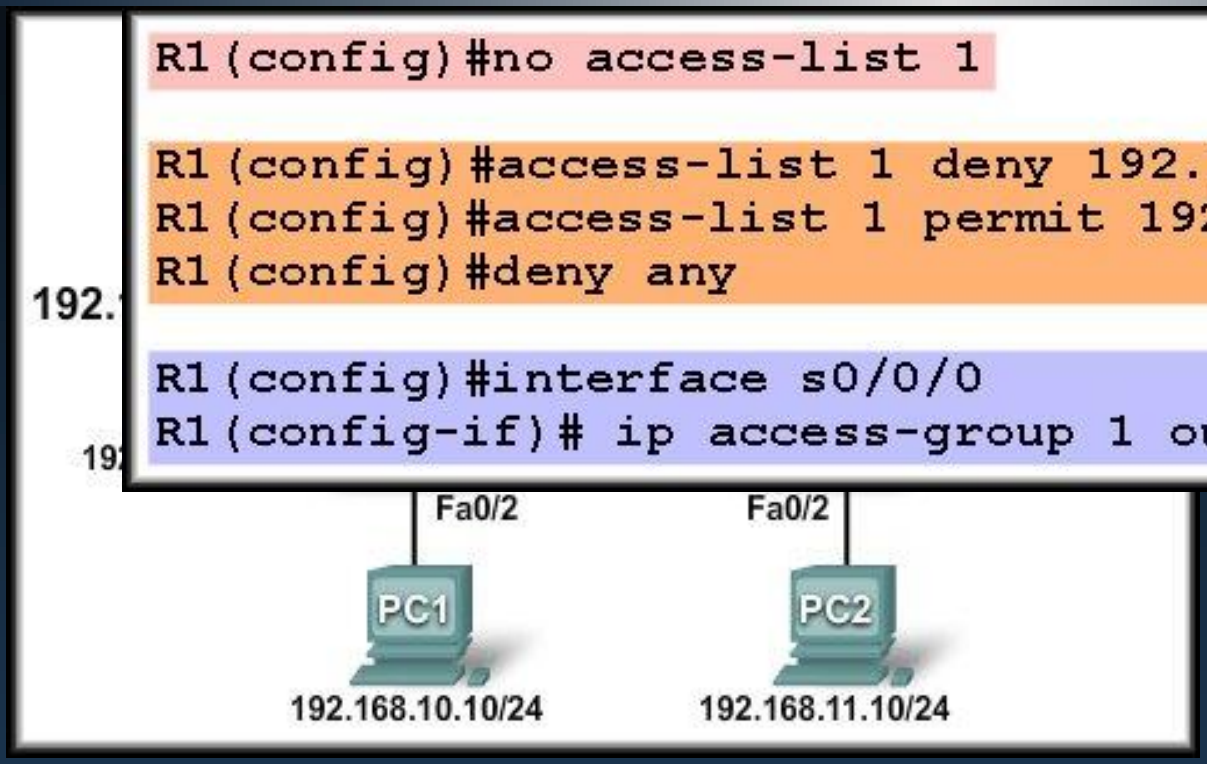
- Example 1:
  - Allow only traffic from network 192.168.10.0 to exit the network on S0/0/0. Block any traffic from any other network.

Applying Standard ACLs to Interfaces

```
R1(config)#no access-list 1

R1(config)#access-list 1 deny 192.168.10.10 0.0.0.0
R1(config)#access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)#deny any

R1(config)#interface s0/0/0
R1(config-if)# ip access-group 1 out
```



- Example 2:
  - Deny any traffic from host 192.168.10.10 and allow any other 192.160.10.0 traffic to exit the network on S0/0/0. Block any traffic from any other network.

Applying Standard ACLs to Interfaces

- Example 3:
  - Deny any traffic from host 192.168.10.10 and allow any other subnet traffic to exit the network on S0/0/0.

Applying Standard ACLs to Interfaces

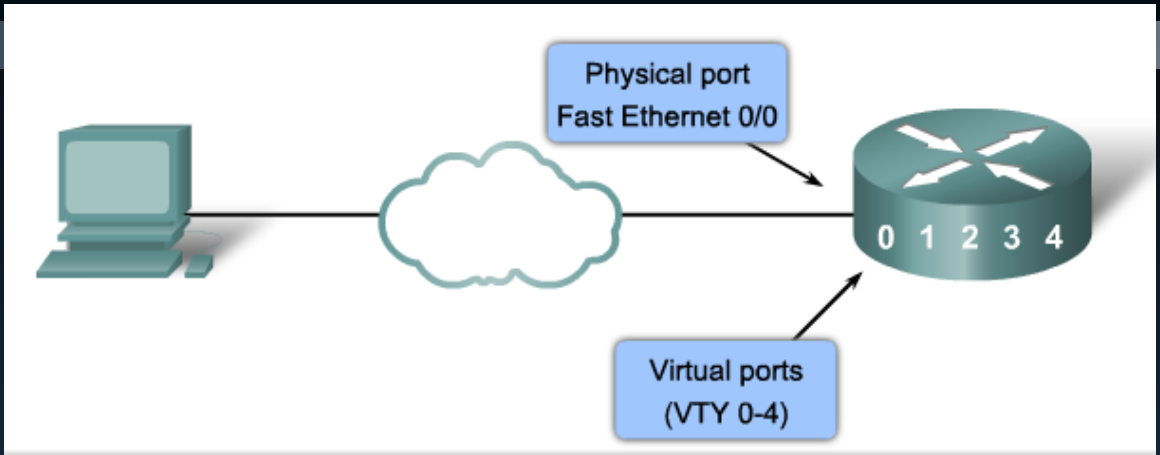
- Using an ACL to Control VTY Access:
  - If your router does not support SSH, this technique allows you to define which IP addresses are allowed Telnet access to the router EXEC process.

```
access-class access-list-number {in [vrf-also] | out}
```

List number

in – restricts incoming connections
out – restricts outgoing connections

Using an ACL to Control VTY Access



- Creating the ACLs:

```
R1(config)#access-list 21 permit 192.168.10.0 0.0.0.255
R1(config)#access-list 21 permit 192.168.11.0 0.0.0.255
R1(config)#access-list 21 deny any
```

- Applying the ACLs:

```
R1(config)#line vty 0 4
R1(config-line)#login
R1(config-line)#password secret
R1(config-line)#access-class 21 in
```

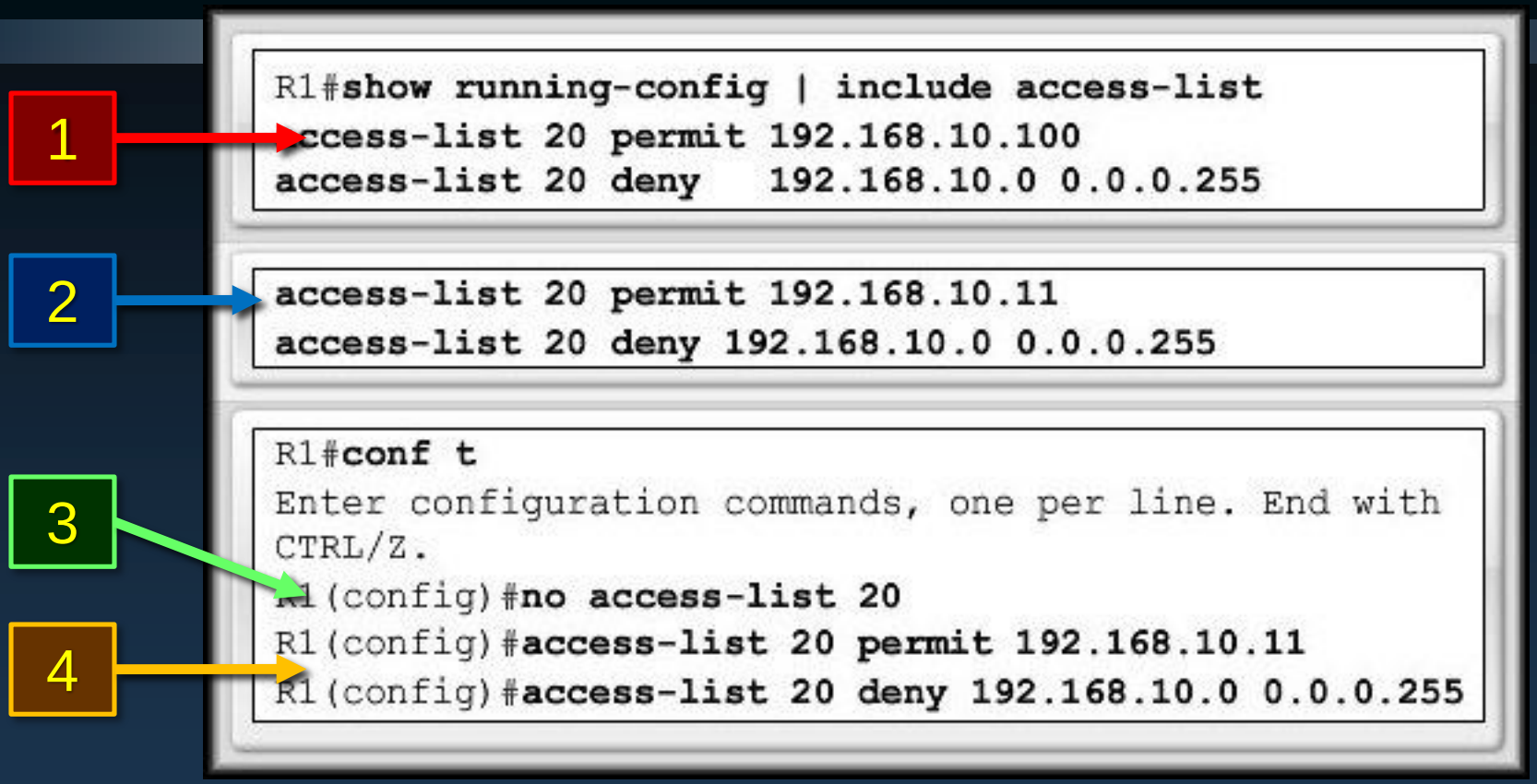
Editing Numbered ACLs

- When configuring an ACL, the statements are added in the order that they are entered at the end of the ACL.
  - *There is no built-in editing feature that allows you to edit a change in an ACL.*
  - You cannot selectively insert or delete lines.
  - It is strongly recommended that any ACL be constructed in a text editor such as Notepad.

Editing Numbered ACLs

- When configuring an ACL, the statements are added in the order that they are entered at the end of the ACL.
  - Four Steps:
    - Display the ACL using the **`show running-config`** command.
    - Highlight the ACL, copy it, and then paste it into Notepad.
    - Make your changes.
    - Disable the access list using the **`no access-list`** command. Otherwise, the new statements would be appended to the existing ACL.
    - Paste the new ACL into the configuration of the router.

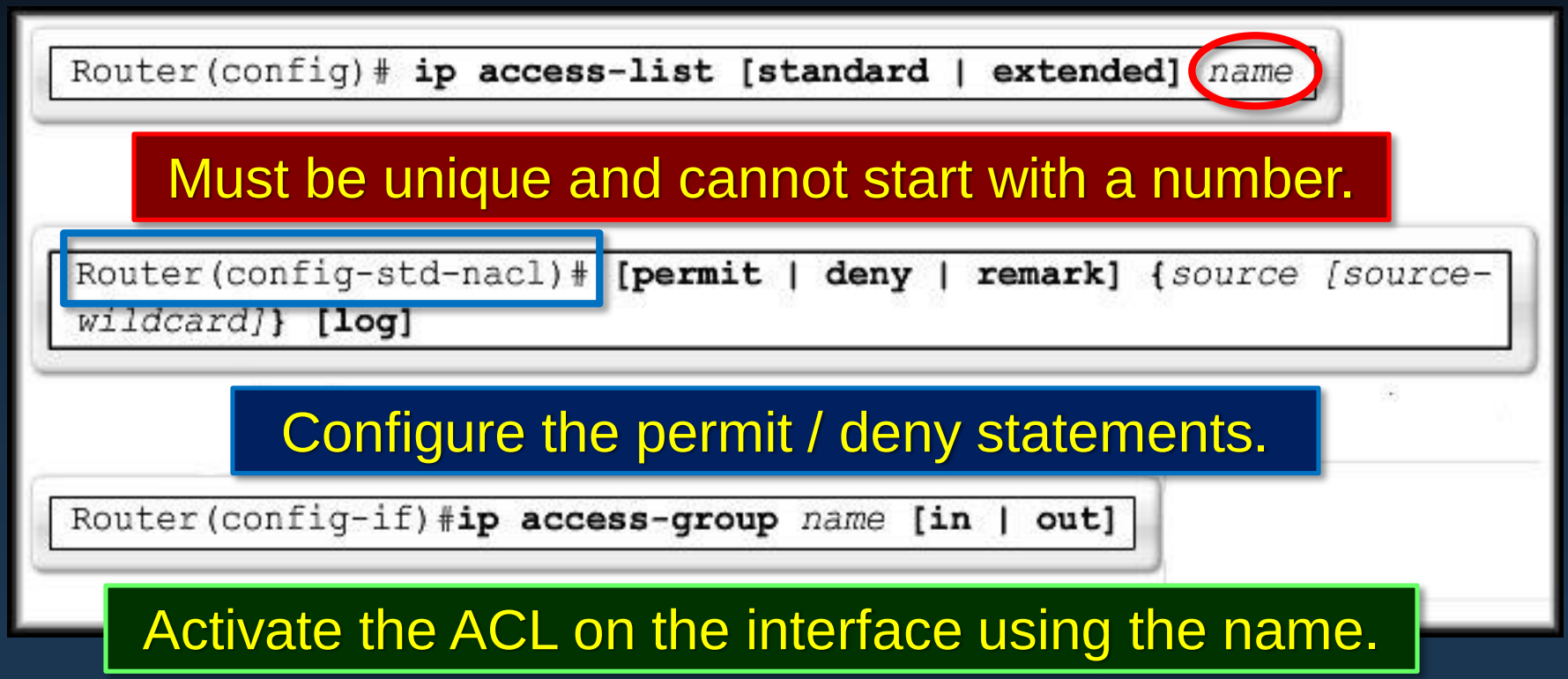
Editing Numbered ACLs



- Be aware that when you use the **`no access-list`** command, no ACL is protecting your network.
  - If you make an error in the new list, you have to disable it and troubleshoot the problem.

Creating Standard Named ACLs

- Naming an ACL makes it easier to understand.



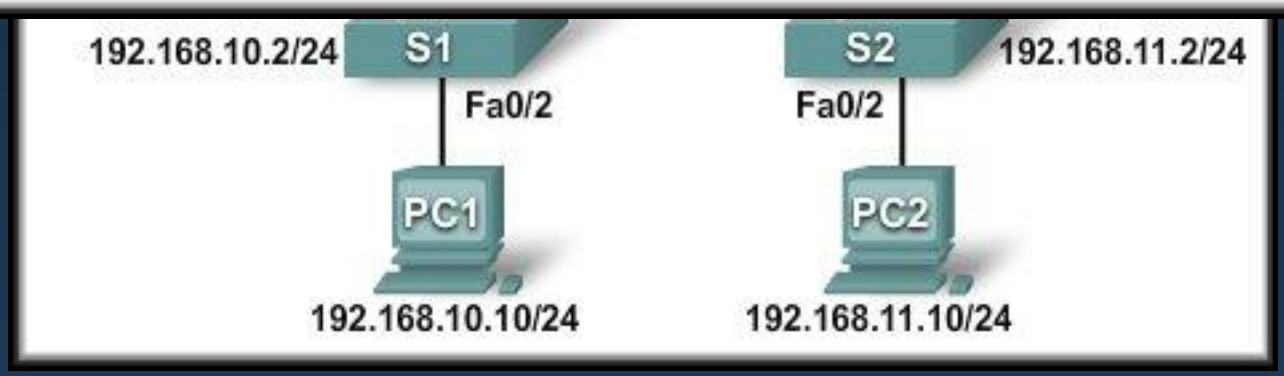
Creating Standard Named ACLs

- Naming an ACL makes it easier to understand.

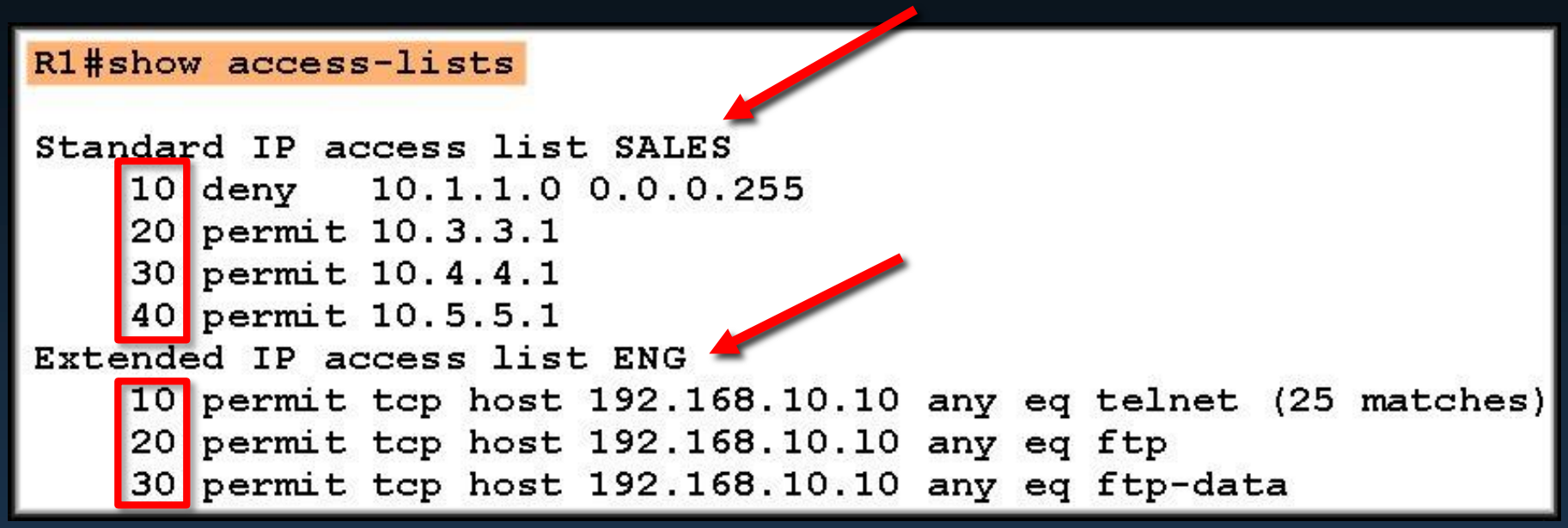
```
R1(config)#ip access-list standard NO_ACCESS

R1(config-std-nacl)#deny host 192.168.11.10
R1(config-std-nacl)#permit 192.168.11.0 0.0.0.255
R1(config-std-nacl)#deny any
R1(config-std-nacl)#end

R1(config)#interface Fa0/1
R1(config-if)# ip access-group NO_ACCESS out
```



Monitoring and Verifying ACLs



Remember that there is an implied **deny any** at the end of each access control list.

Editing Named ACLs

- Named ACLs have a big advantage over numbered ACLs in that they are easier to edit.

```
R1#show access-lists
Standard IP access list WEBSERVER
    10 permit host 192.168.10.10
    20 deny 192.168.10.0 0.0.0.255
    30 deny 192.168.11.0 0.0.0.255
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip access-list standard WEBSERVER
R1(config-std-nacl)#15 permit host 192.168.11.0
R1(config-std-nacl)#end
R1#
%SYS-5-CONFIG_I: Configured from console by console
R1#show access-lists
Standard IP access list WEBSERVER
    10 permit host 192.168.10.10
    15 permit host 192.168.11.0
    20 deny 192.168.10.0 0.0.0.255
    30 deny 192.168.11.0 0.0.0.255
R1#
```