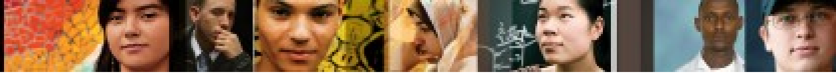
Instructor Materials Chapter 4: Access Control Lists

CCNA Routing and Switching

Connecting Networks

Cisco | Networking Academy®
Mind Wide Open™



Chapter 4: Best Practices

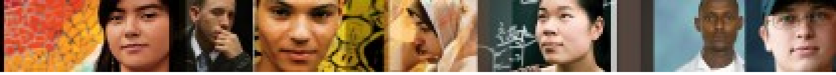
Prior to teaching Chapter 4, the instructor should:

- Complete Chapter 4 Assessment.
- Ensure all activities are completed. This is a very important concept and hands-on time is vital.
- Provide the students many ACL building activities.
- Encourage students to login with their cisco.com login and read http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_acl/configuration/15-sy/sec-data-acl-15-sy-book/sec-acl-ov-gdl.html

Chapter 4: Access Control Lists

Connecting Networks

Cisco | Networking Academy®
Mind Wide Open™

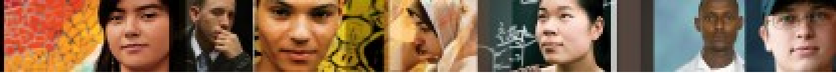


Chapter 4 - Sections & Objectives

- 4.1 Standard ACL Operation and Configuration
  - Configure standard IPv4 ACLs.
- 4.2 Extended IPv4 ACLs
  - Configure extended IPv4 ACLs.
- 4.3 IPv6 ACLs
  - Configure IPv6 ACLs.
- 4.4 Troubleshoot ACLs
  - Troubleshoot ACLs.

4.1 Standard ACL Operation and Configuration Review

Cisco | Networking Academy®
Mind Wide Open™



ACL Operation Overview

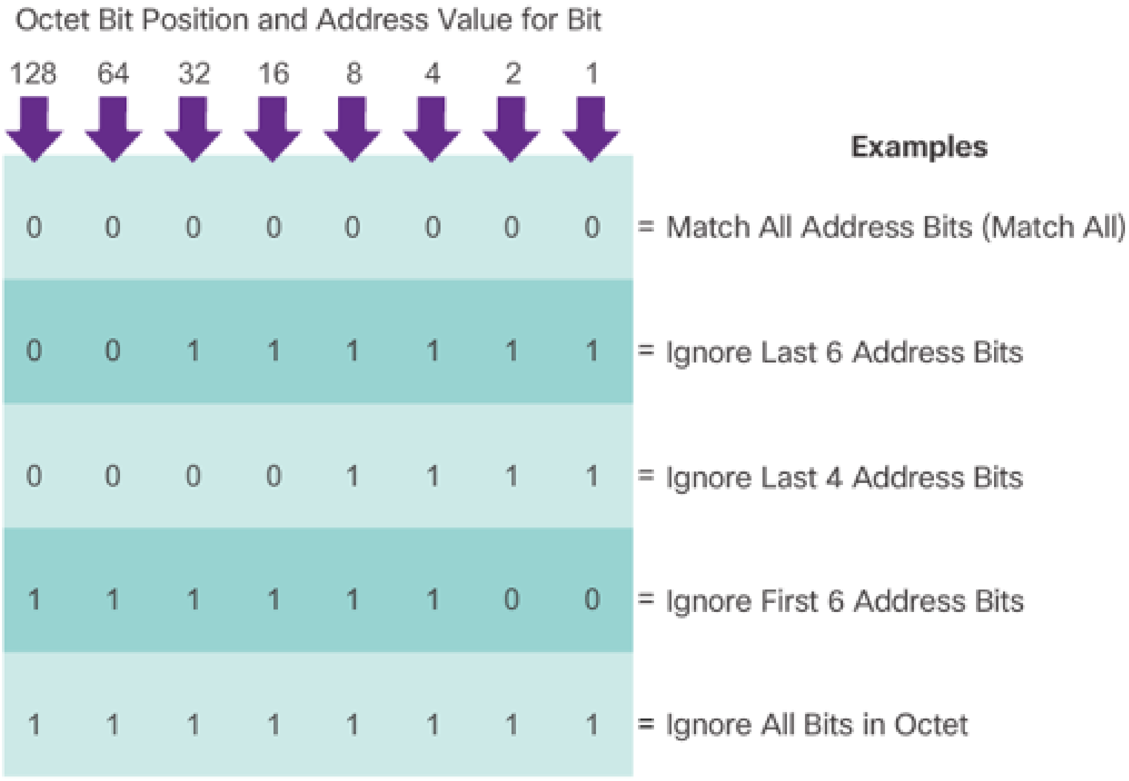
ACLs and the Wildcard Mask

- An ACL is a sequential list of permit or deny statements, known as access control entries (ACEs).
- As network traffic passes through an interface configured with an ACL, the router compares the information within the packet against each ACE.
- An IPv4 ACE includes the use of a wildcard mask to filter IPv4 addresses.

ACL Operation Overview

ACLs and the Wildcard Mask cont…

Wildcard Masking



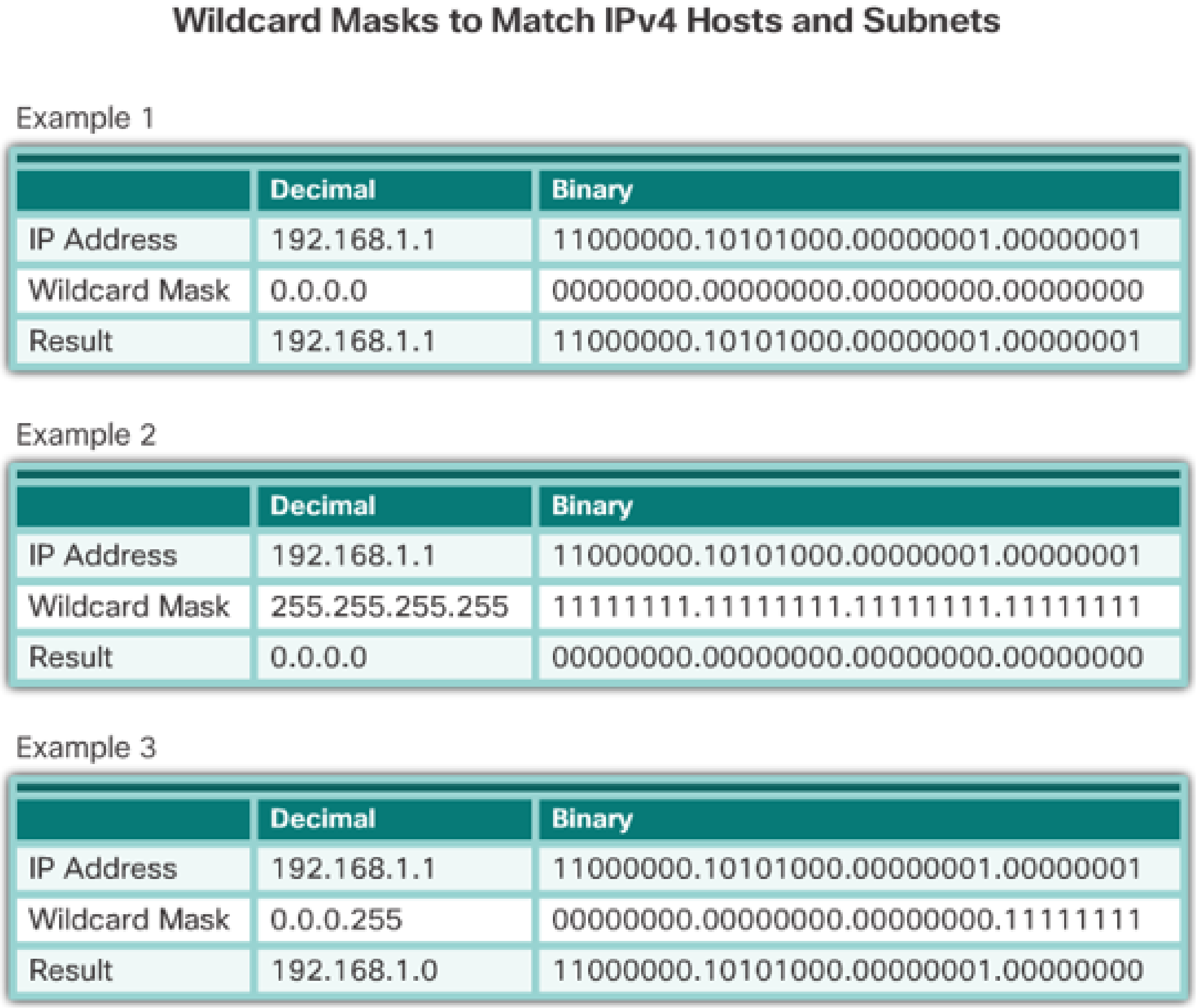
Octet Bit Position and Address Value for Bit

128	64	32	16	8	4	2	1	Examples
0	0	0	0	0	0	0	0	= Match All Address Bits (Match All)
0	0	1	1	1	1	1	1	= Ignore Last 6 Address Bits
0	0	0	0	1	1	1	1	= Ignore Last 4 Address Bits
1	1	1	1	1	1	0	0	= Ignore First 6 Address Bits
1	1	1	1	1	1	1	1	= Ignore All Bits in Octet

0 means to match the value of the corresponding address bit
1 means to ignore the value of the corresponding address bit

ACL Operation Overview

ACLs and the Wildcard Mask cont…



Wildcard Masks to Match IPv4 Hosts and Subnets

Example 1

	Decimal	Binary
IP Address	192.168.1.1	11000000.10101000.00000001.00000001
Wildcard Mask	0.0.0.0	00000000.00000000.00000000.00000000
Result	192.168.1.1	11000000.10101000.00000001.00000001

Example 2

	Decimal	Binary
IP Address	192.168.1.1	11000000.10101000.00000001.00000001
Wildcard Mask	255.255.255.255	11111111.11111111.11111111.11111111
Result	0.0.0.0	00000000.00000000.00000000.00000000

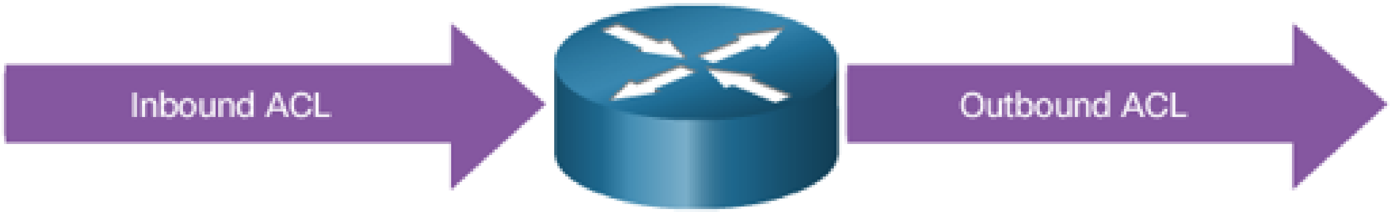
Example 3

	Decimal	Binary
IP Address	192.168.1.1	11000000.10101000.00000001.00000001
Wildcard Mask	0.0.0.255	00000000.00000000.00000000.11111111
Result	192.168.1.0	11000000.10101000.00000001.00000000

ACL Operation Overview

Applying ACLs to an Interface

Inbound and Outbound ACLs



Inbound ACL

Outbound ACL

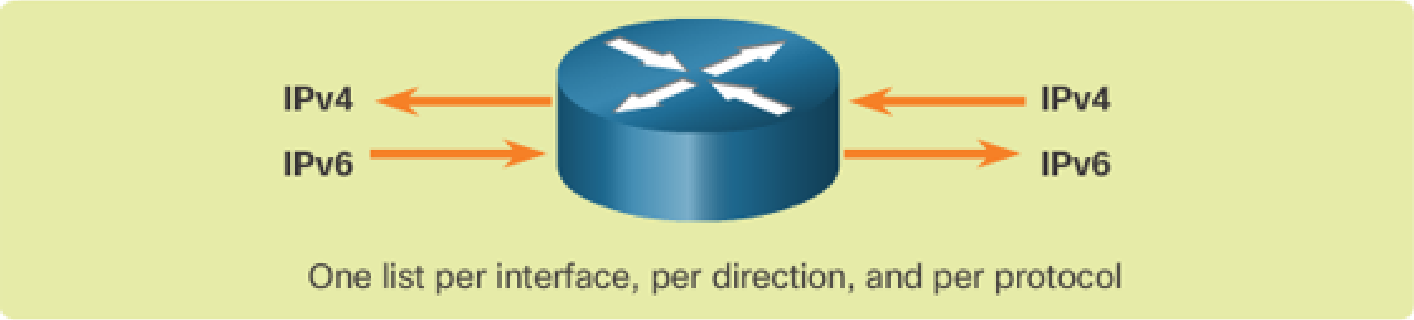
An inbound ACL filters packets coming into a specific interface and before they are routed to the outbound interface.

An outbound ACL filters packets after being routed, regardless of the inbound interface.

ACL Operation Overview

Applying ACLs to an Interface cont…

ACL Traffic Filtering on a Router



One list per interface, per direction, and per protocol

With two interfaces and two protocols running, this router could have a total of 8 separate ACLs applied.

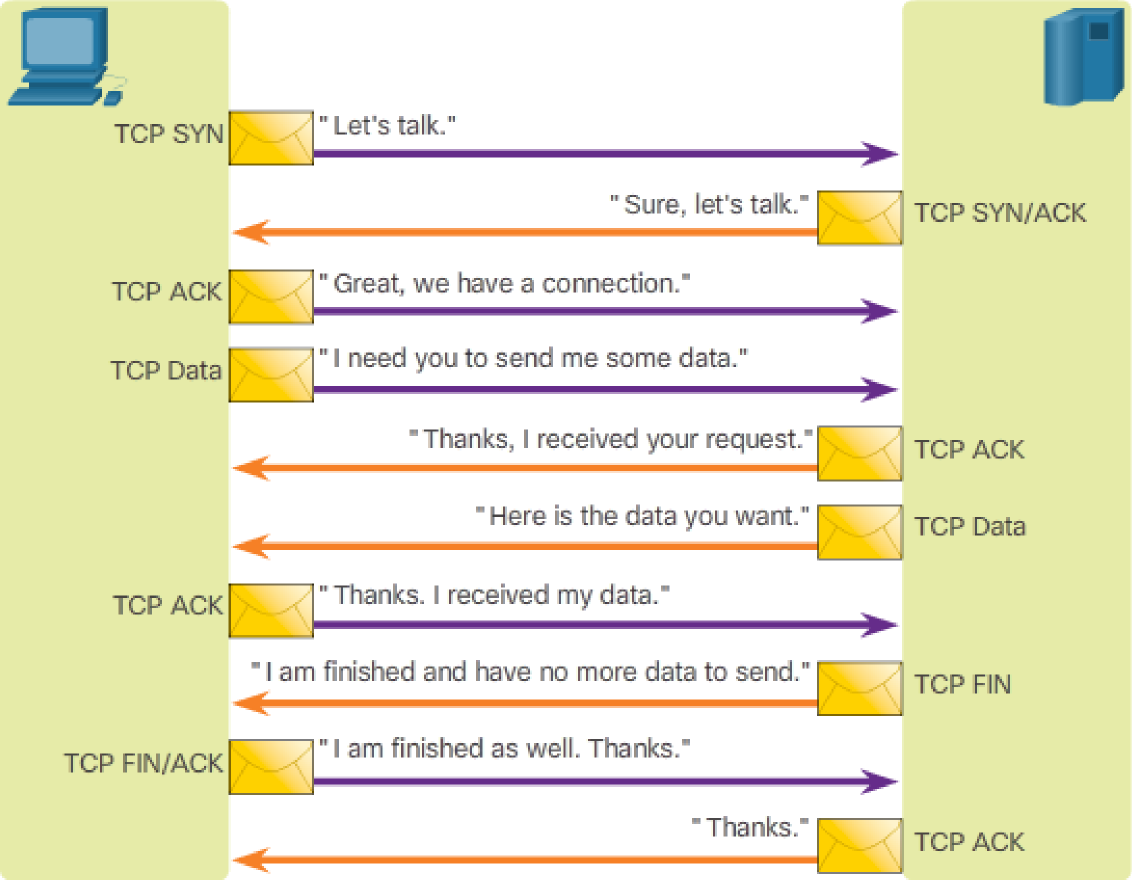
The Rules for Applying ACLs

You can only have one ACL per protocol, per interface, and per direction:

- One ACL per protocol (e.g., IPv4 or IPv6)
- One ACL per direction (i.e., IN or OUT)
- One ACL per interface (e.g., GigabitEthernet0/0)

ACL Operation Overview

A TCP Conversation


TCP SYN
"Let's talk."
"Sure, let's talk."
TCP SYN/ACK
TCP ACK
"Great, we have a connection."
TCP Data
"I need you to send me some data."
"Thanks, I received your request."
TCP ACK
"Here is the data you want."
TCP Data
TCP ACK
"Thanks. I received my data."
"I am finished and have no more data to send."
TCP FIN
TCP FIN/ACK
"I am finished as well. Thanks."
"Thanks."
TCP ACK

TCP segments are marked with flags that denote their purpose:

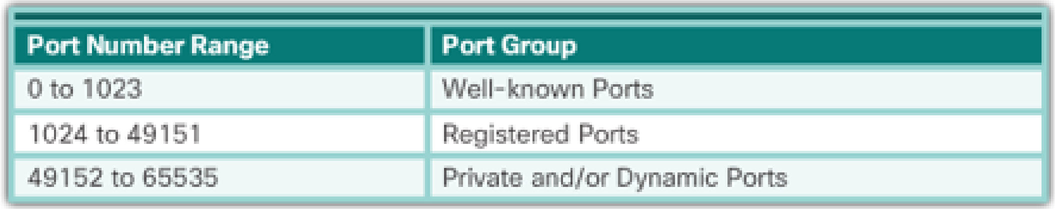
- a SYN starts (synchronizes) the session
- an ACK is an acknowledgment that an expected segment was received
- a FIN finishes the session.

ACL Operation Overview

A TCP Conversation cont…

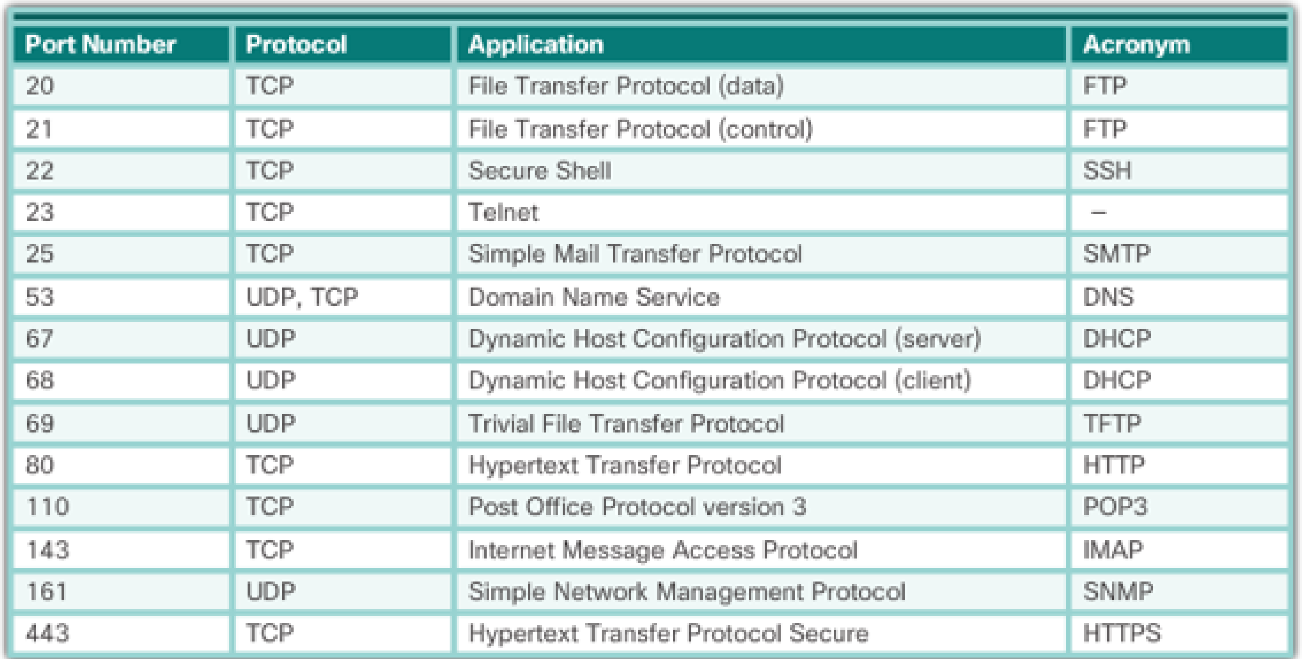
- The TCP data segment also identifies the port which matches the requested service.

Port Numbers



Port Number Range	Port Group
0 to 1023	Well-known Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Well-Known Port Numbers



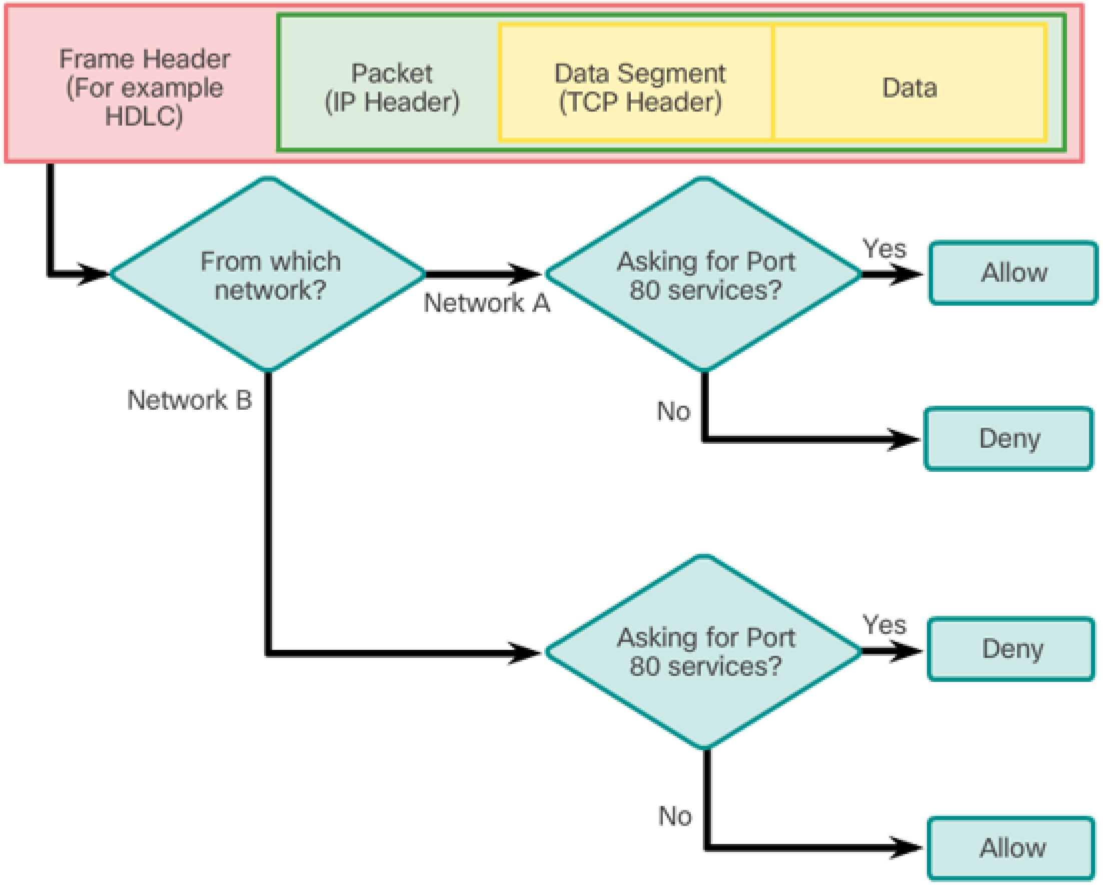
Port Number	Protocol	Application	Acronym
20	TCP	File Transfer Protocol (data)	FTP
21	TCP	File Transfer Protocol (control)	FTP
22	TCP	Secure Shell	SSH
23	TCP	Telnet	–
25	TCP	Simple Mail Transfer Protocol	SMTP
53	UDP, TCP	Domain Name Service	DNS
67	UDP	Dynamic Host Configuration Protocol (server)	DHCP
68	UDP	Dynamic Host Configuration Protocol (client)	DHCP
69	UDP	Trivial File Transfer Protocol	TFTP
80	TCP	Hypertext Transfer Protocol	HTTP
110	TCP	Post Office Protocol version 3	POP3
143	TCP	Internet Message Access Protocol	IMAP
161	UDP	Simple Network Management Protocol	SNMP
443	TCP	Hypertext Transfer Protocol Secure	HTTPS

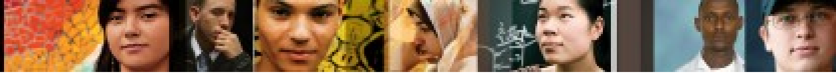
ACL Operation Overview

ACL Packet Filtering

- Packet filtering controls access to a network by analyzing the incoming and outgoing packets and forwarding them or discarding them based on given criteria.

Packet Filtering Example


Frame Header (For example HDLC)
Packet (IP Header)
Data Segment (TCP Header)
Data
From which network?
Network A
Asking for Port 80 services?
Yes
Allow
No
Deny
Network B
Asking for Port 80 services?
Yes
Deny
No
Allow



Types of IPv4 ACLs

Standard and Extended IPv4 ACLs

- Two types of Cisco IPv4 ACLS:
  - Standard
    - Standard ACLs can be used to permit or deny traffic only from source IPv4 addresses. The destination of the packet and the ports involved are not evaluated
  - Extended
    - Extended ACLs filter IPv4 packets based on several attributes:

      Protocol type

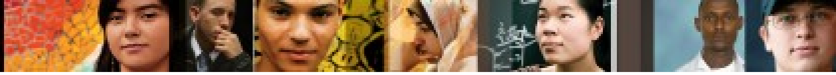
      Source IPv4 address

      Destination IPv4 address

      Source TCP or UDP ports

      Destination TCP or UDP ports

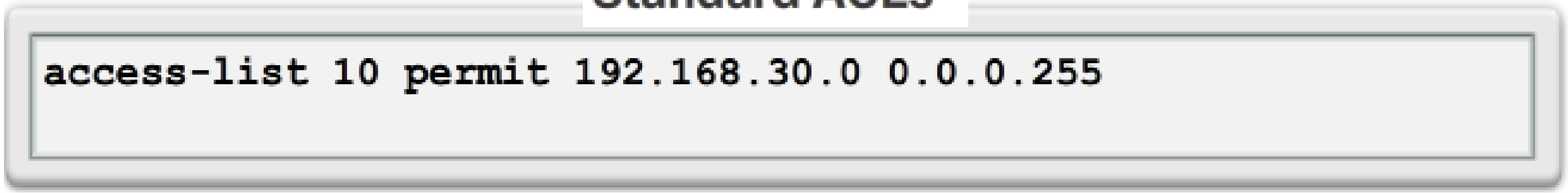
      Optional protocol type information for finer control



Types of IPv4 ACLs

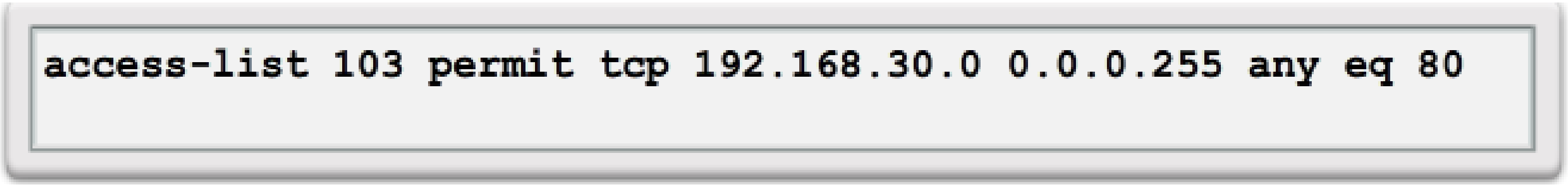
Standard and Extended IPv4 ACLs cont…

Standard ACLs



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

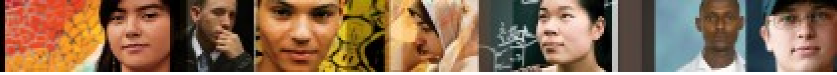
Extended ACLs



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

Extended ACLs filter IP packets based on several attributes, including the following:

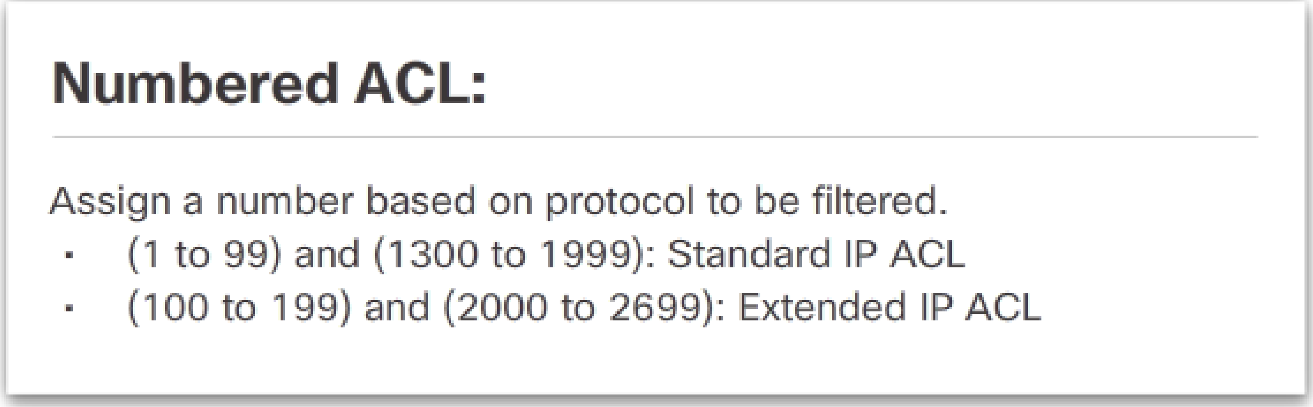
- Source and destination IP addresses
- Source and destination TCP and UDP ports
- Protocol type/Protocol number (example: IP, ICMP, UDP, TCP, etc.)



Types of IPv4 ACLs

Numbered and Named ACLs

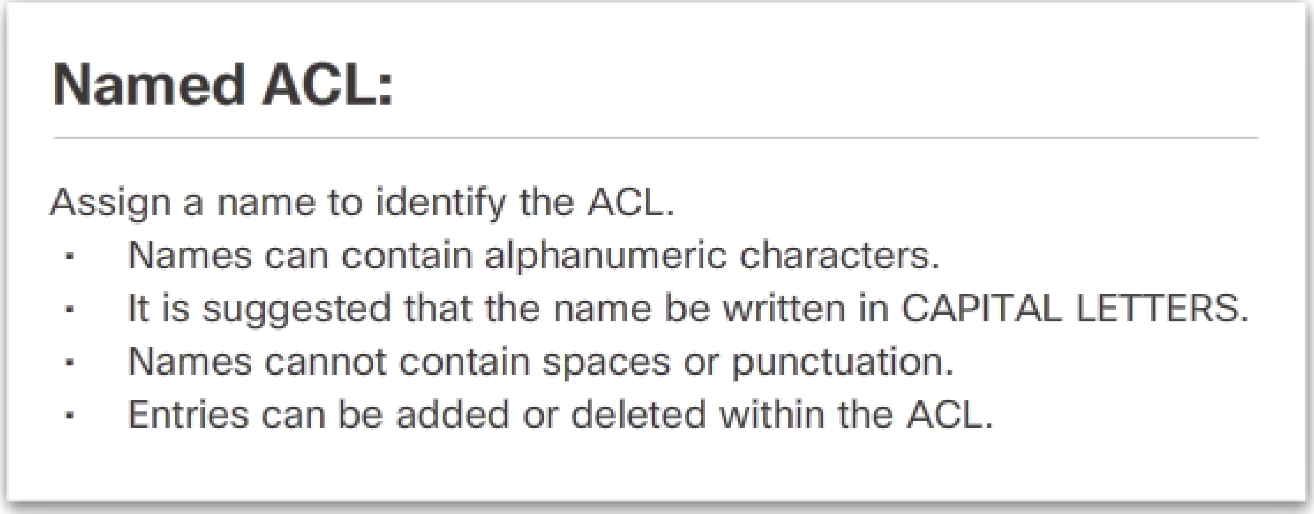
- Standard and extended ACLs can be created using either a number or a name to identify the ACL.



Numbered ACL:

Assign a number based on protocol to be filtered.

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

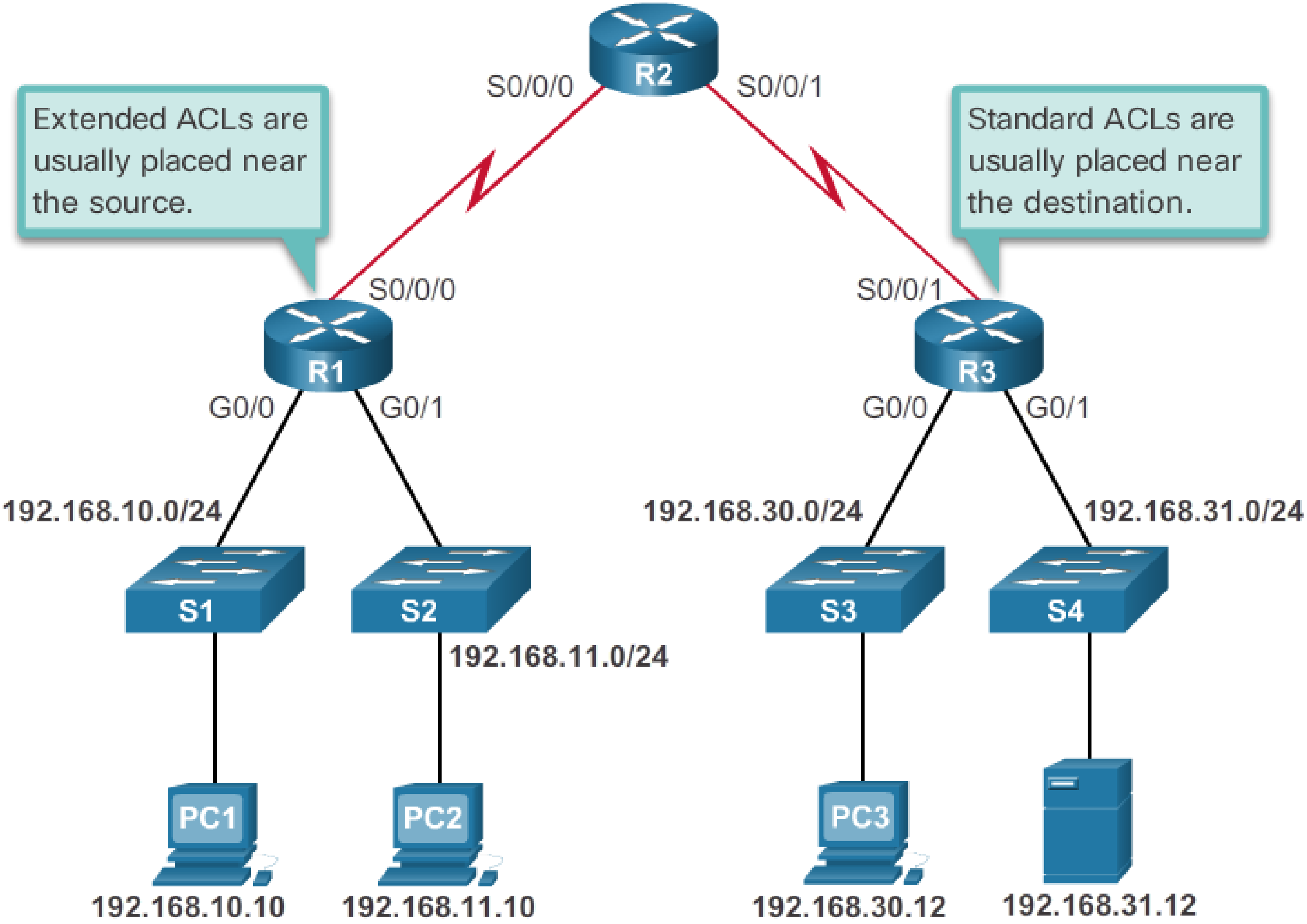


Named ACL:

Assign a name to identify the ACL.

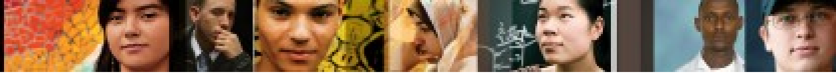
- Names can contain alphanumeric characters.
- It is suggested that the name be written in CAPITAL LETTERS.
- Names cannot contain spaces or punctuation.
- Entries can be added or deleted within the ACL.

Where to Place ACLs



Extended ACLs are usually placed near the source.

Standard ACLs are usually placed near the destination.



Types of IPv4 ACLs

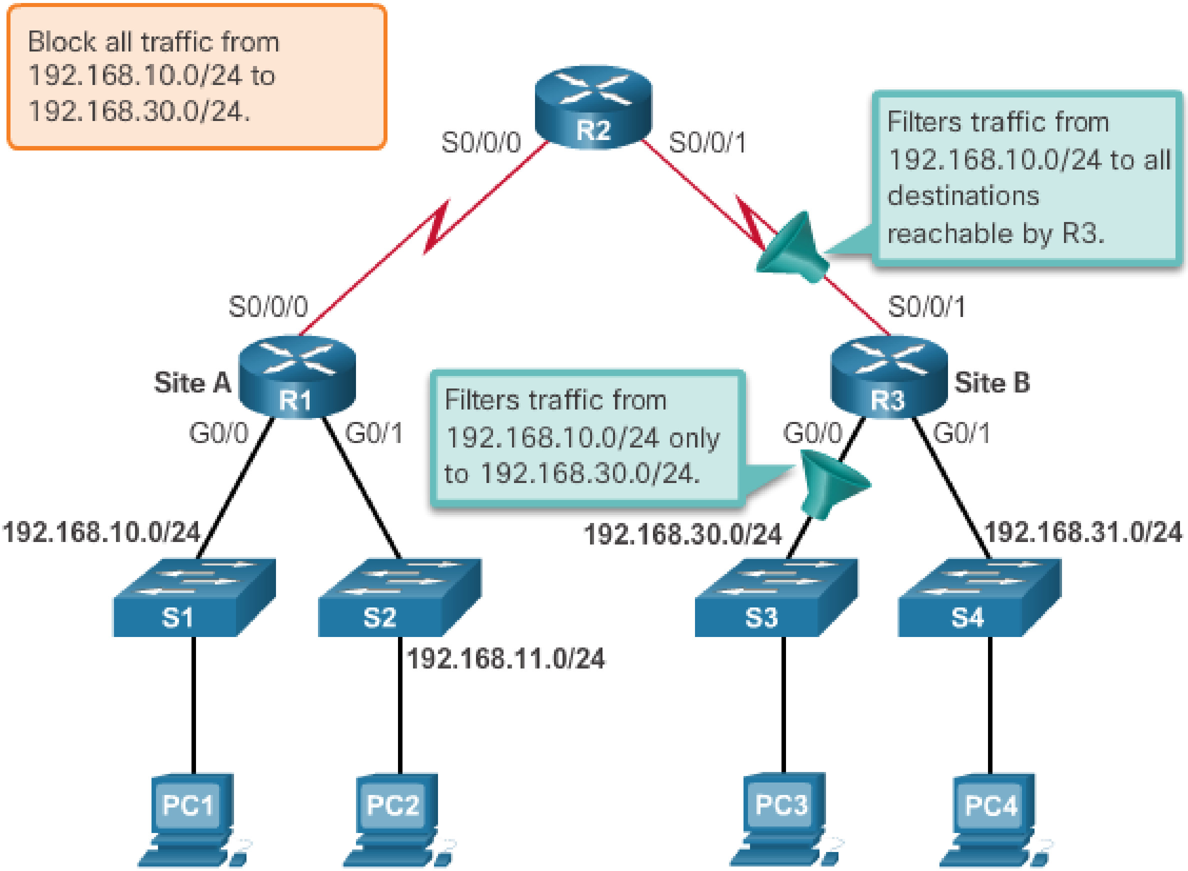
Where to Place ACLs cont…

- Every ACL should be placed where it has the greatest impact on efficiency. The basic rules are:
  - Extended ACLs - Locate extended ACLs as close as possible to the source of the traffic to be filtered.
  - Standard ACLs - Because standard ACLs do not specify destination addresses, place them as close to the destination as possible.
  - Placement of the ACL, and therefore the type of ACL used, may also depend on: the extent of the network administrator’s control, bandwidth of the networks involved, and ease of configuration.

Types of IPv4 ACLs

Standard ACL Placement Example

- The administrator wants to prevent traffic originating in the 192.168.10.0/24 network from reaching the 192.168.30.0/24 network.



Block all traffic from 192.168.10.0/24 to 192.168.30.0/24.

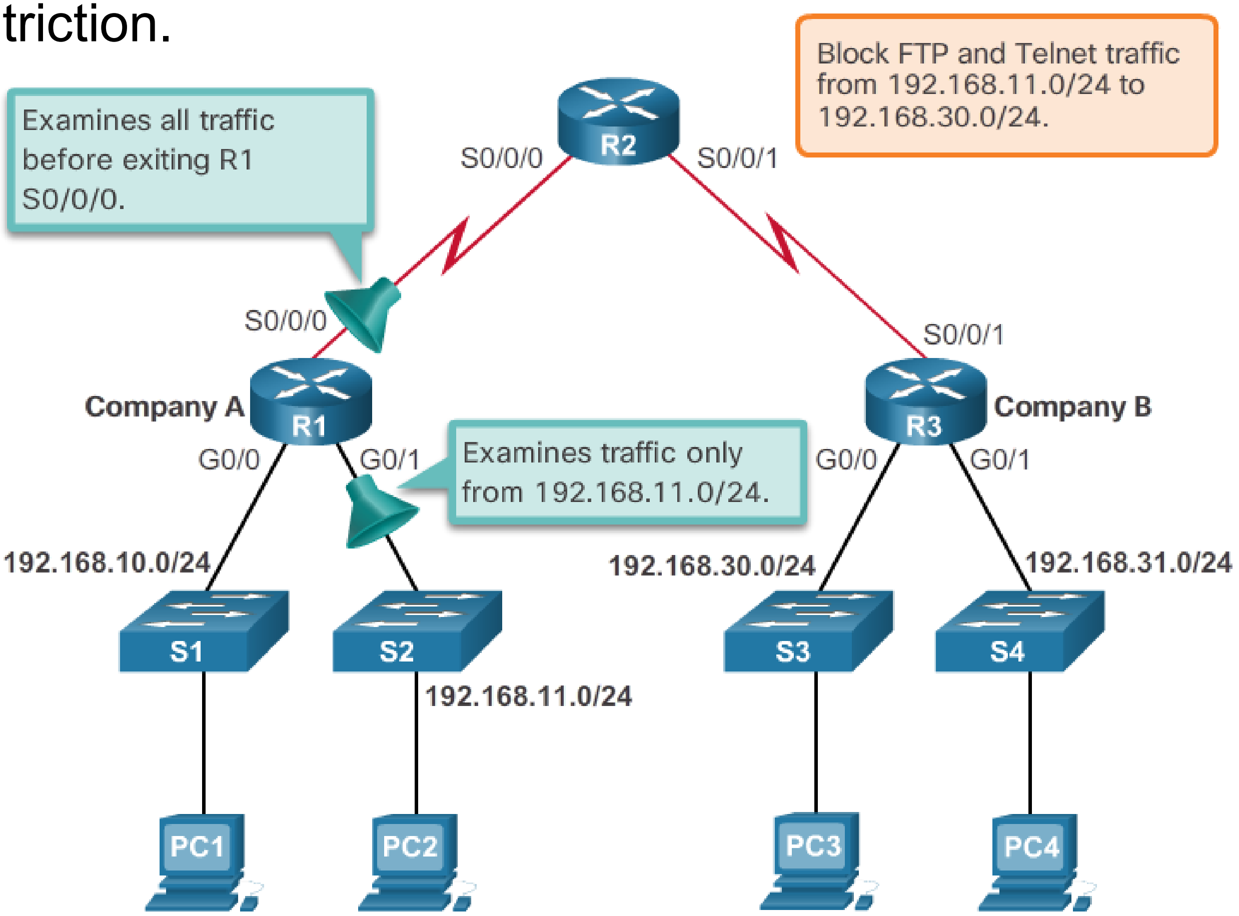
Filters traffic from 192.168.10.0/24 to all destinations reachable by R3.

Filters traffic from 192.168.10.0/24 only to 192.168.30.0/24.

Types of IPv4 ACLs

Extended ACL Placement Example

- The administrator wants to deny Telnet and FTP traffic from the 192.168.11.0/24 network to Company B’s 192.168.30.0/24 network. All other traffic from the .11 network must be permitted to leave Company A without restriction.



Block FTP and Telnet traffic from 192.168.11.0/24 to 192.168.30.0/24.

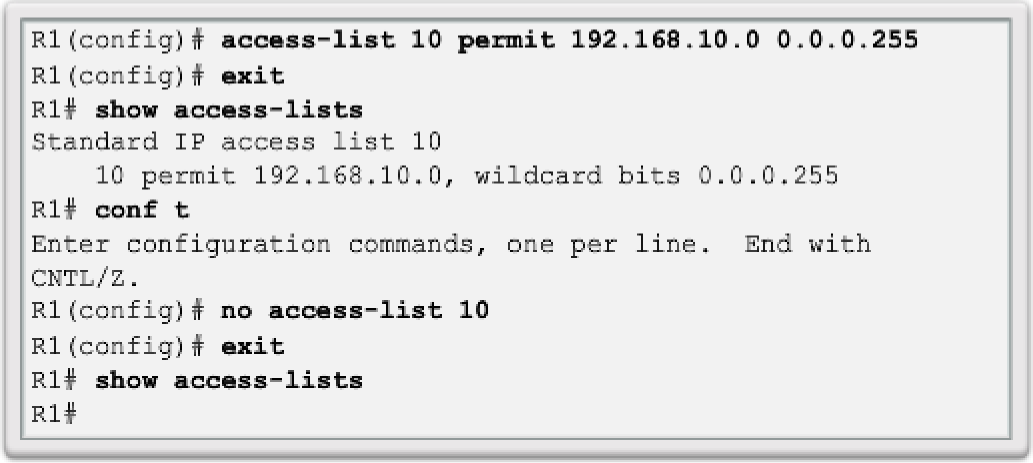
Examines all traffic before exiting R1 S0/0/0.

Examines traffic only from 192.168.11.0/24.

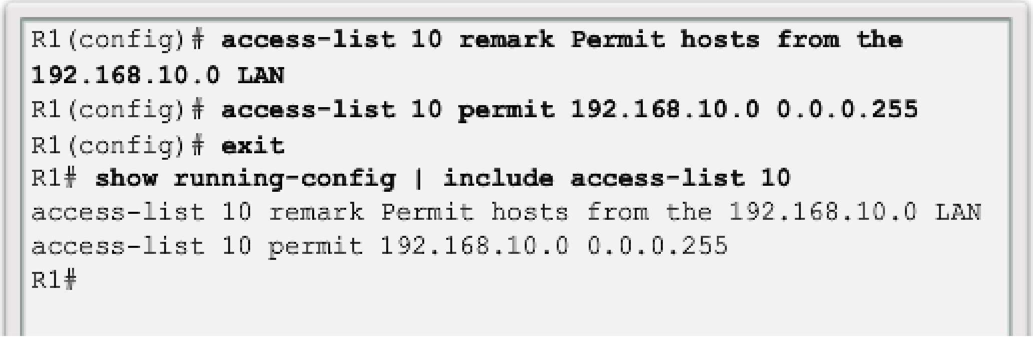
Standard IPv4 ACL Configuration

Configure a Standard IPv4 ACL

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



```
R1(config)# access-list 10 permit 192.168.10.0 0.0.0.255
R1(config)# exit
R1# show access-lists
Standard IP access list 10
    10 permit 192.168.10.0, wildcard bits 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# show access-lists
R1#
```

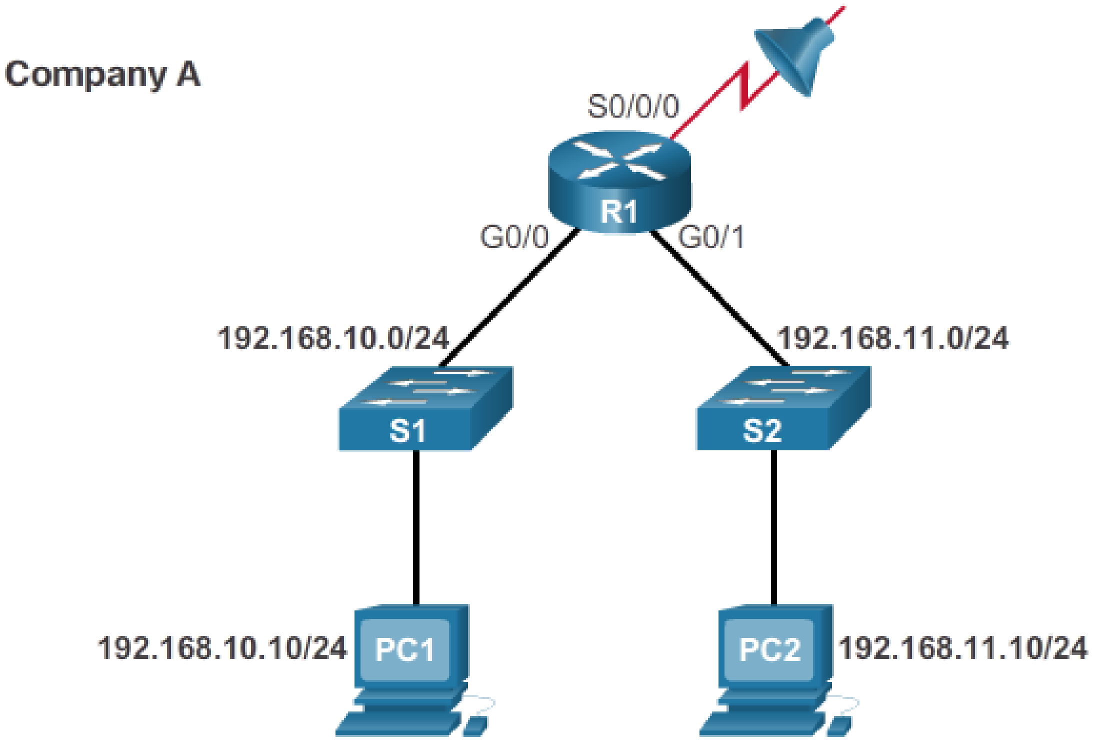


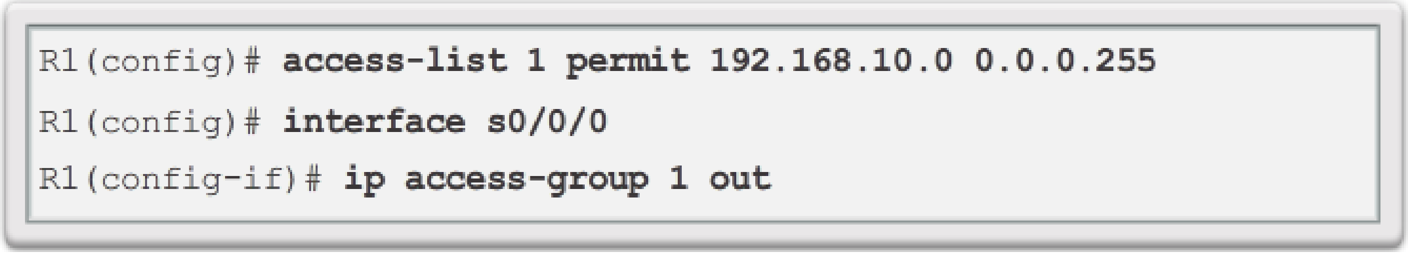
```
R1(config)# access-list 10 remark Permit hosts from the
192.168.10.0 LAN
R1(config)# access-list 10 permit 192.168.10.0 0.0.0.255
R1(config)# exit
R1# show running-config | include access-list 10
access-list 10 remark Permit hosts from the 192.168.10.0 LAN
access-list 10 permit 192.168.10.0 0.0.0.255
R1#
```

Standard IPv4 ACL Configuration

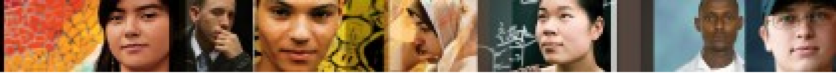
Apply a Standard IPv4 ACL

Permit a Specific Subnet



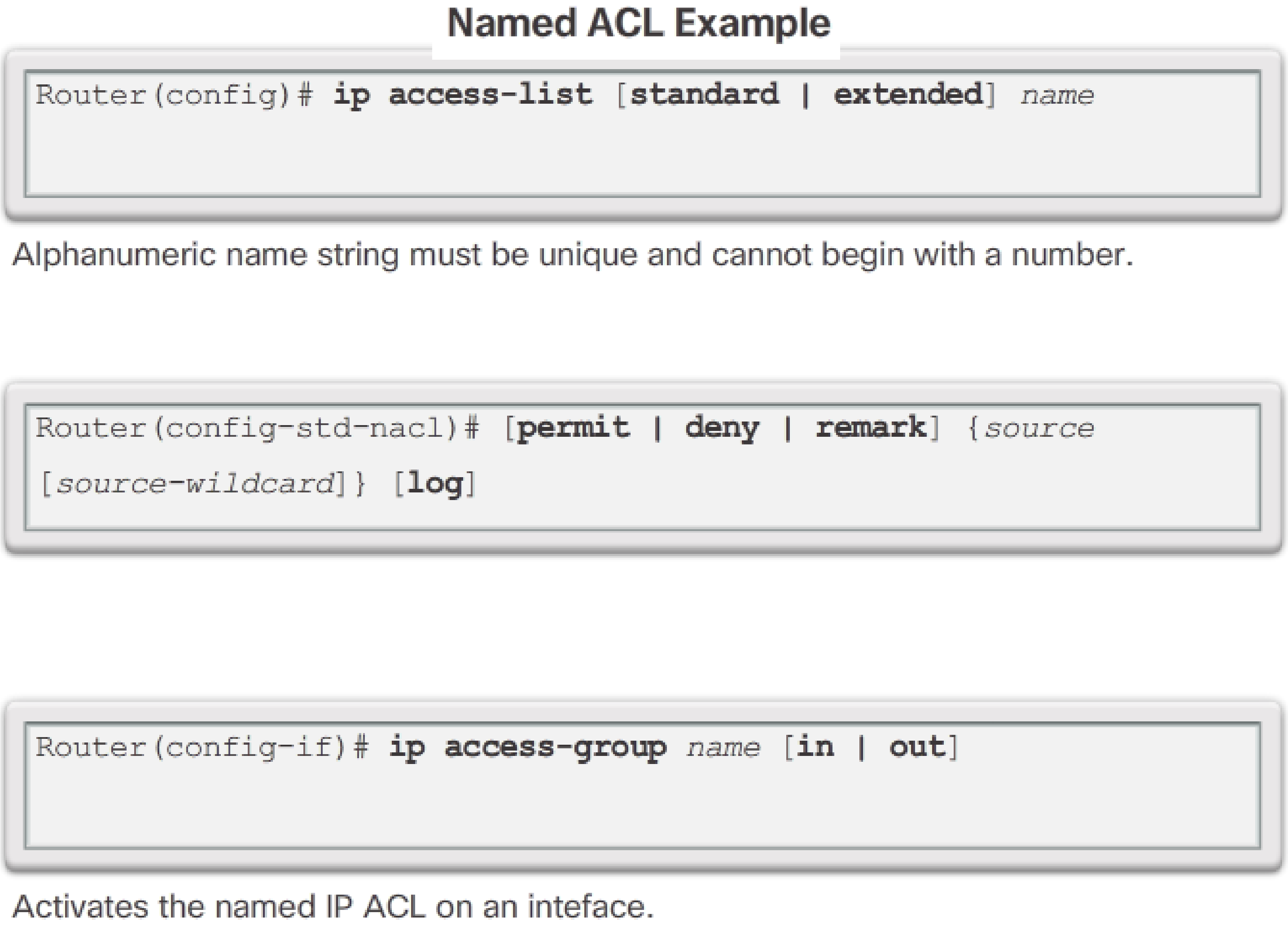


```
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)# interface s0/0/0
R1(config-if)# ip access-group 1 out
```



Standard IPv4 ACL Configuration

Named Standard IPv4 ACLs



Named ACL Example

```
Router(config)# ip access-list [standard | extended] name
```

Alphanumeric name string must be unique and cannot begin with a number.

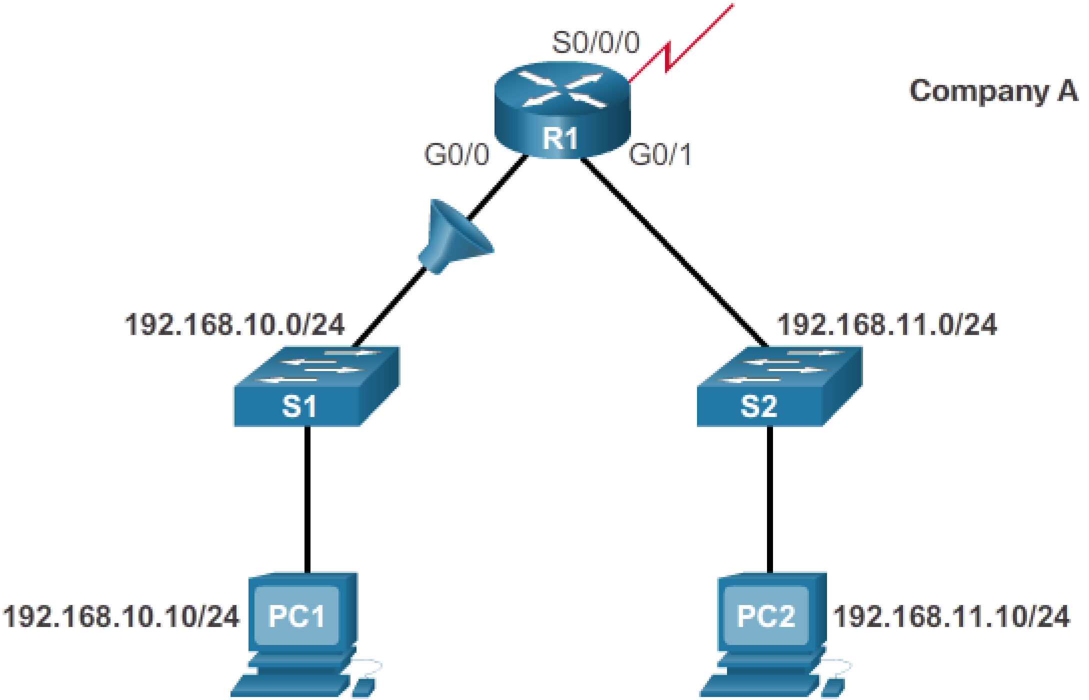
```
Router(config-std-nacl)# [permit | deny | remark] {source
[source-wildcard]} [log]
```

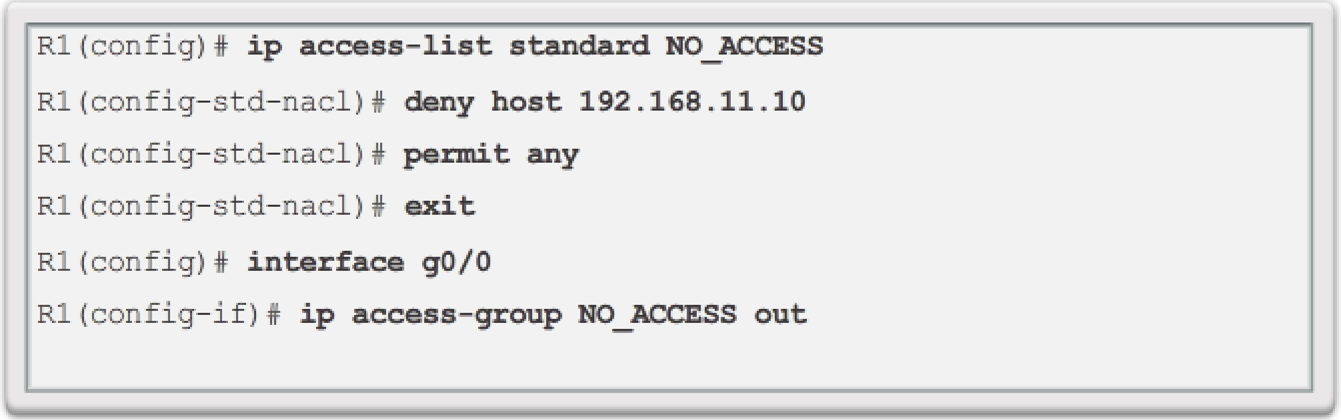
```
Router(config-if)# ip access-group name [in | out]
```

Activates the named IP ACL on an inteface.

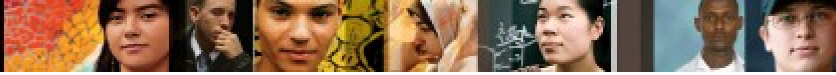
Standard IPv4 ACL Configuration

Named Standard IPv4 ACLs cont…



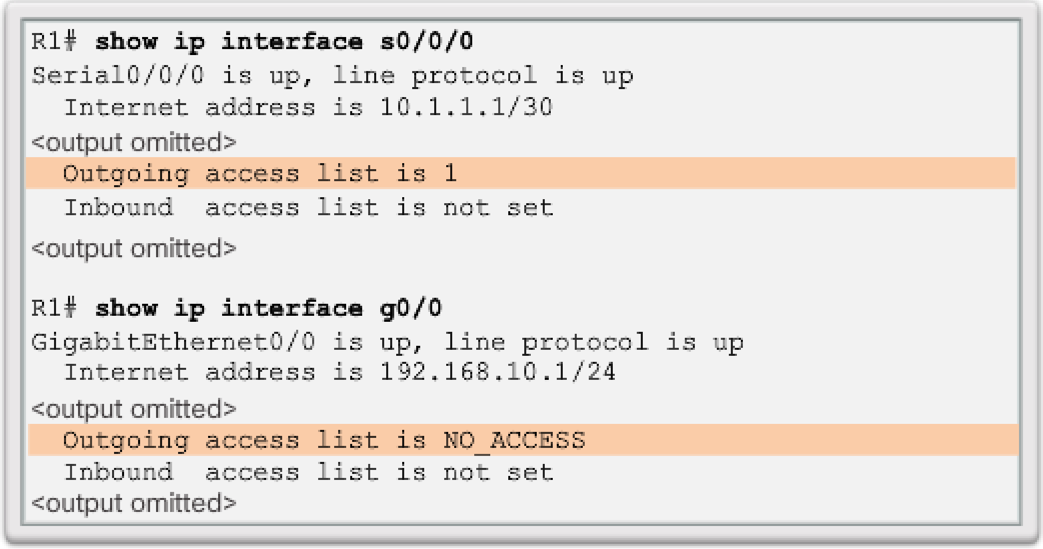


```
R1(config)# ip access-list standard NO_ACCESS
R1(config-std-nacl)# deny host 192.168.11.10
R1(config-std-nacl)# permit any
R1(config-std-nacl)# exit
R1(config)# interface g0/0
R1(config-if)# ip access-group NO_ACCESS out
```



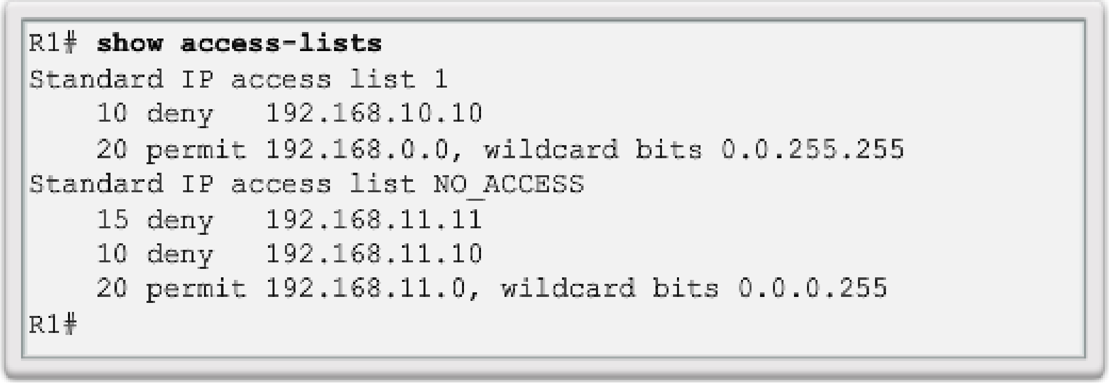
Standard IPv4 ACL Configuration

Verify ACLs



```
R1# show ip interface s0/0/0
Serial0/0/0 is up, line protocol is up
  Internet address is 10.1.1.1/30
<output omitted>
  Outgoing access list is 1
  Inbound  access list is not set
<output omitted>

R1# show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 192.168.10.1/24
<output omitted>
  Outgoing access list is NO_ACCESS
  Inbound  access list is not set
<output omitted>
```



```
R1# show access-lists
Standard IP access list 1
    10 deny   192.168.10.10
    20 permit 192.168.0.0, wildcard bits 0.0.255.255
Standard IP access list NO_ACCESS
    15 deny   192.168.11.11
    10 deny   192.168.11.10
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1#
```

4.2 Extended IPv4 ACLs

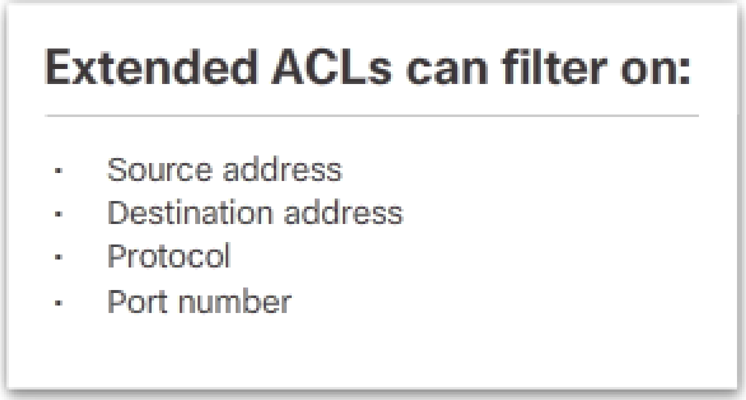
Cisco | Networking Academy®
Mind Wide Open™

Structure of an Extended IPv4 ACLs

Extended ACLs

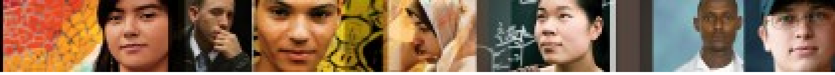
- Extended ACLs are used more often than standard ACLs because they provide a greater degree of control.

Extended ACL



Extended ACLs can filter on:

- Source address
- Destination address
- Protocol
- Port number

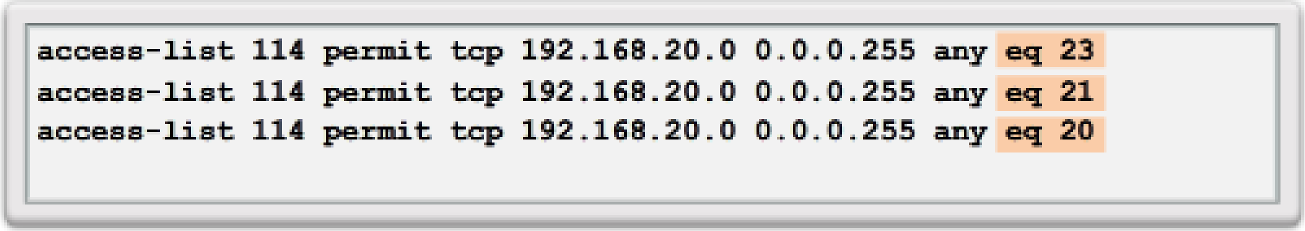


Structure of an Extended IPv4 ACLs

Filtering Ports and Services

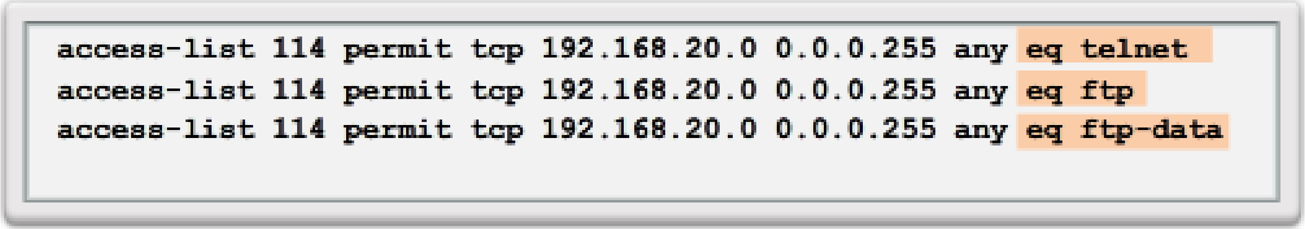
- The ability to filter on protocol and port number allows network administrators to build very specific extended ACLs.
- An application can be specified by configuring either the port number or the name of a well-known port.

Using Port Numbers

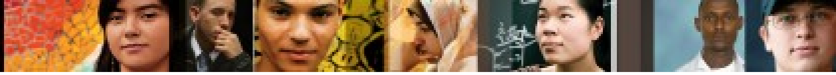


```
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 23
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 21
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 20
```

Using Keywords



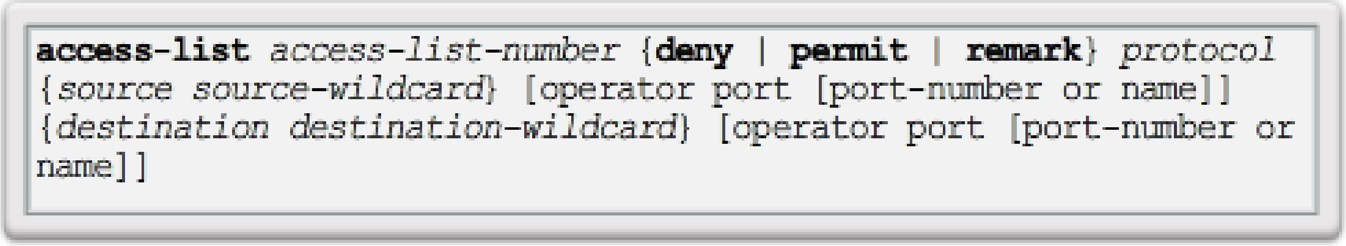
```
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq telnet
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp-data
```



Configure Extended IPv4 ACLs

Configuring Extended ACLs

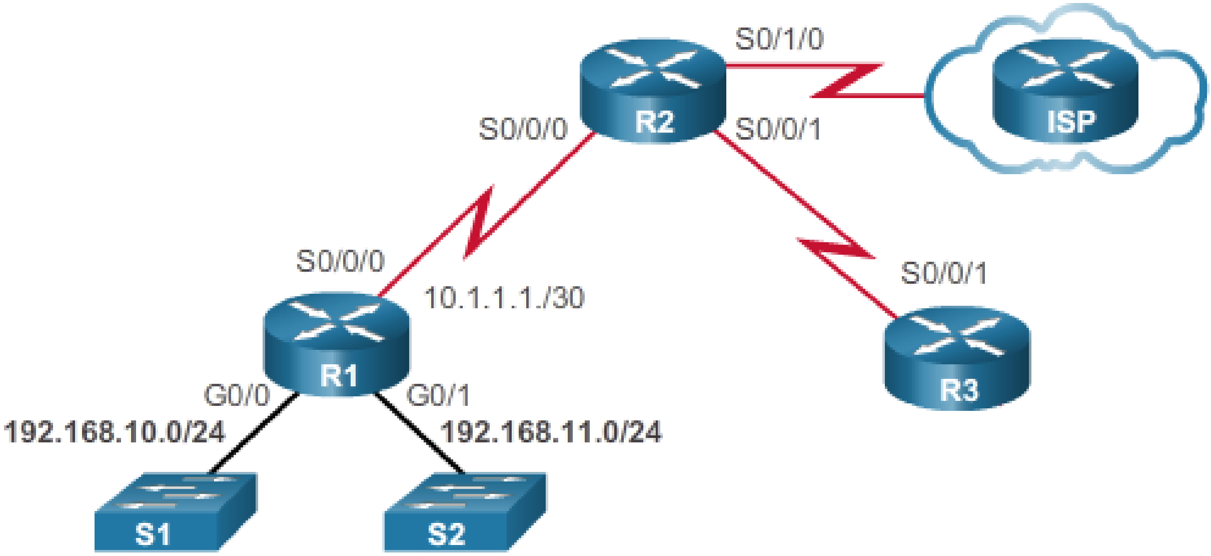
- The procedural steps for configuring extended ACLs are the same as for standard ACLs. The extended ACL is first configured, and then it is activated on an interface. However, the command syntax and parameters are more complex to support the additional features provided by extended ACLs.

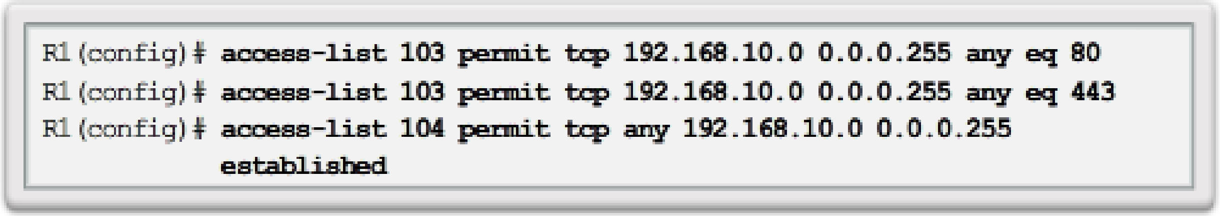


```
access-list access-list-number {deny | permit | remark} protocol
{source source-wildcard} [operator port [port-number or name]]
{destination destination-wildcard} [operator port [port-number or
name]]
```

Configure Extended IPv4 ACLs

Configuring Extended ACLs cont…



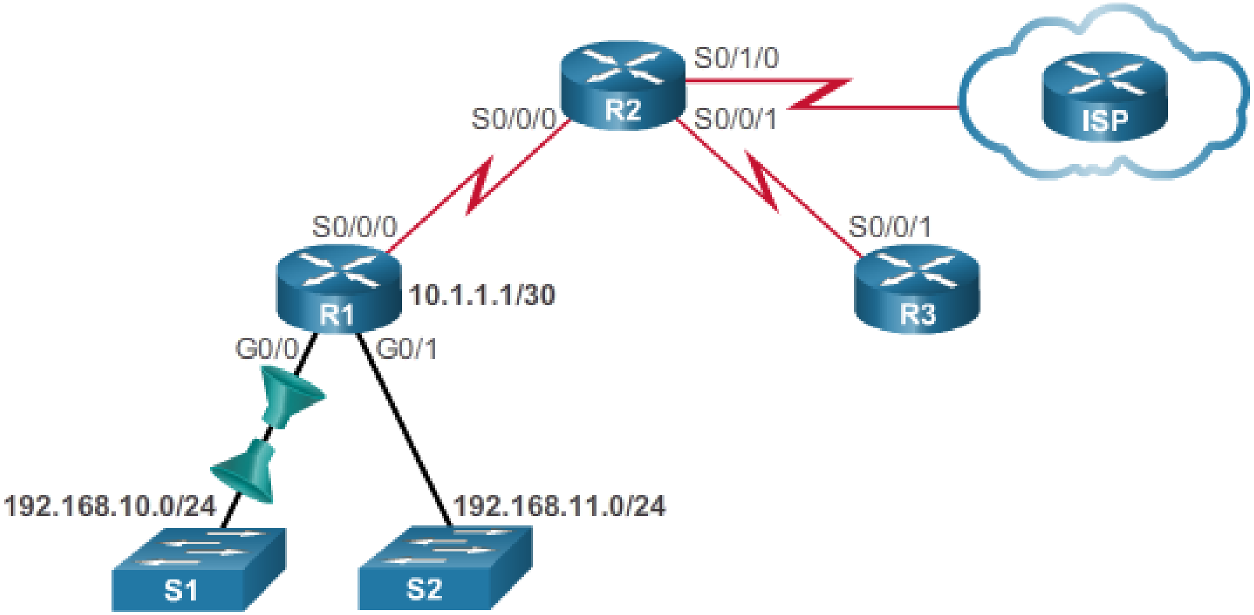


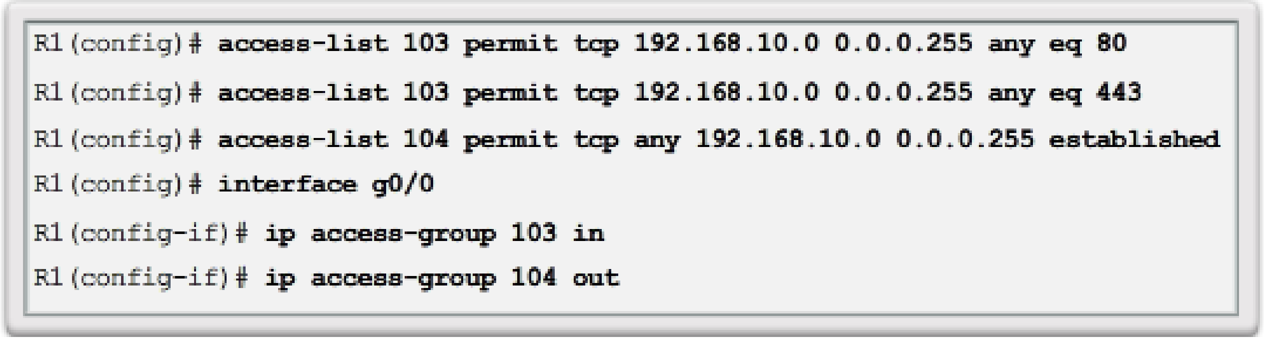
```
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)# access-list 104 permit tcp any 192.168.10.0 0.0.0.255
             established
```

- ACL 103 allows requests to ports 80 and 443.
- ACL 104 allows established HTTP and HTTPS replies.

Configure Extended IPv4 ACLs

Applying Extended ACLs to Interfaces

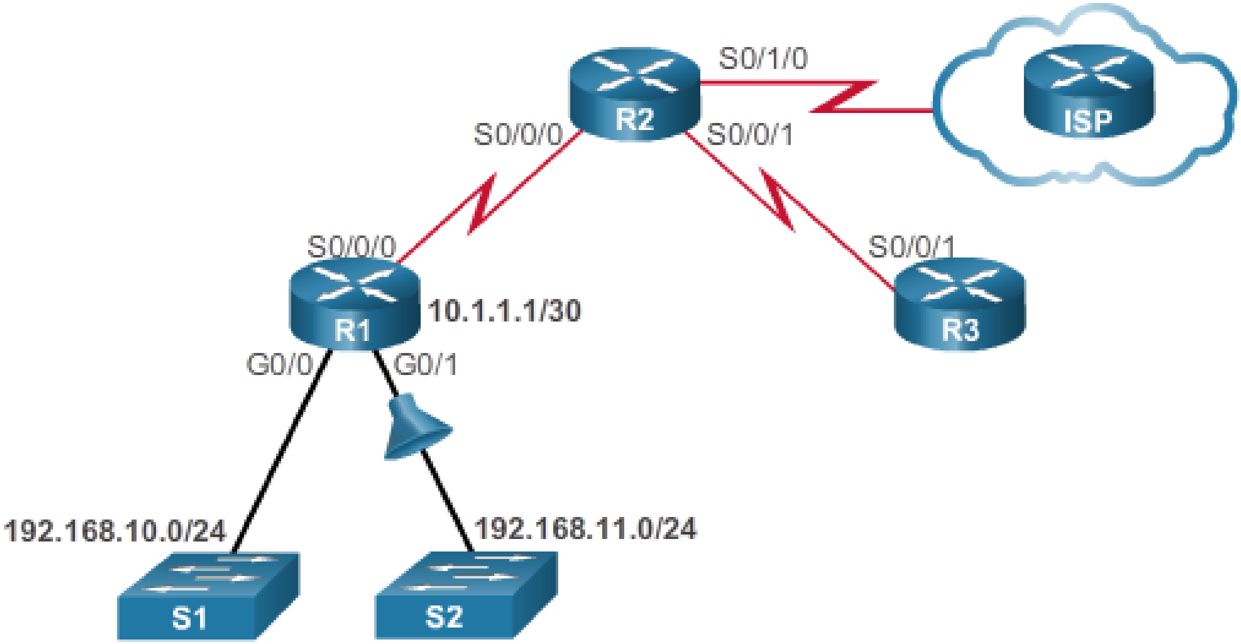


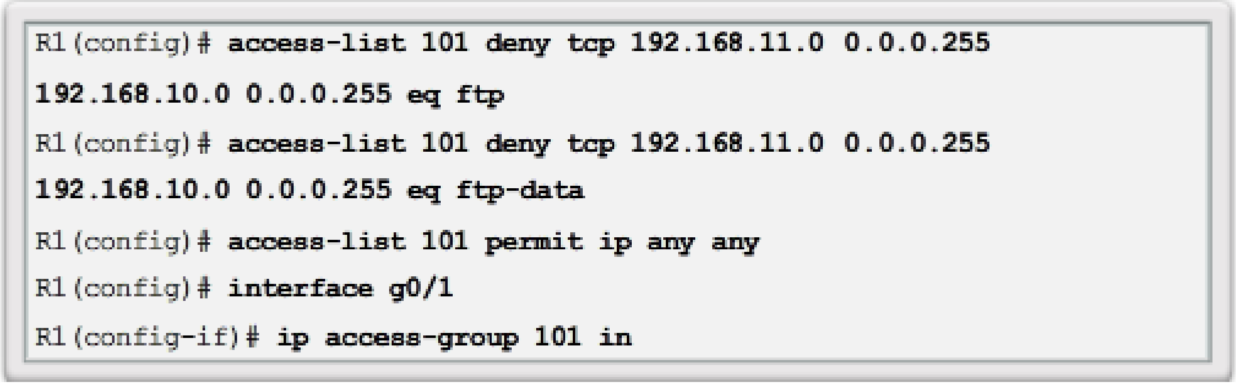


```
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)# access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
R1(config)# interface g0/0
R1(config-if)# ip access-group 103 in
R1(config-if)# ip access-group 104 out
```

Configure Extended IPv4 ACLs

Filtering Traffic with Extended ACLs

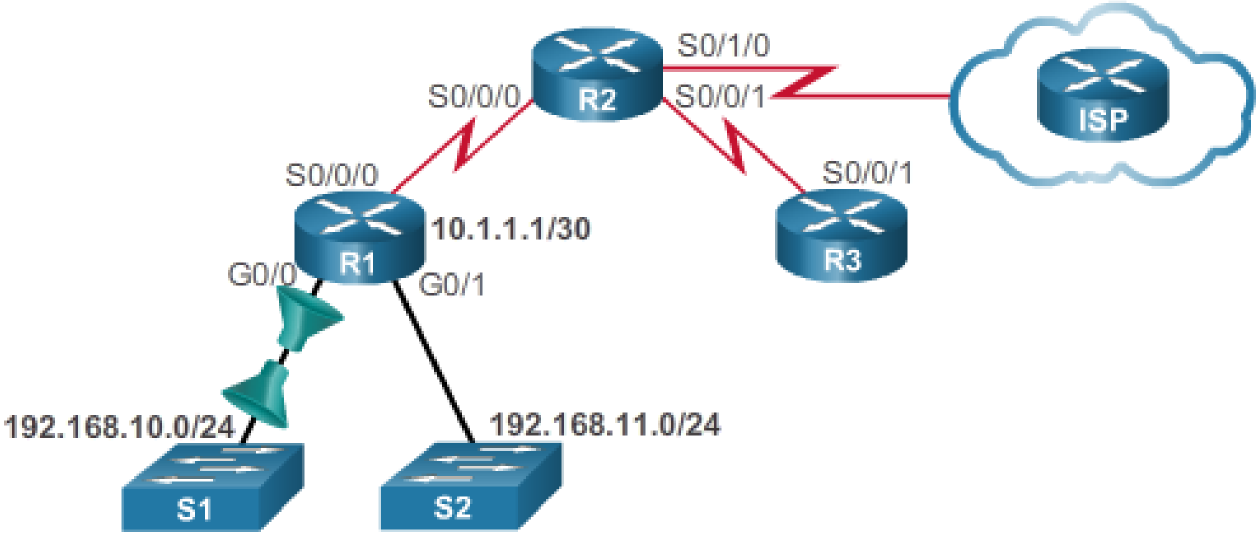


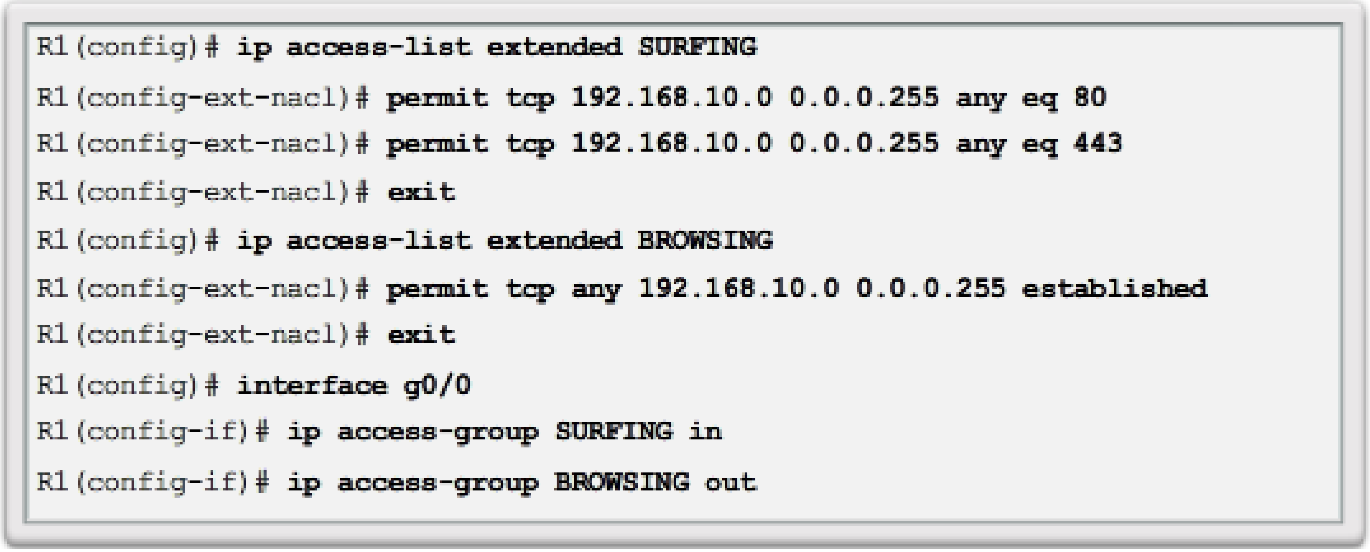


```
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255
192.168.10.0 0.0.0.255 eq ftp
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255
192.168.10.0 0.0.0.255 eq ftp-data
R1(config)# access-list 101 permit ip any any
R1(config)# interface g0/1
R1(config-if)# ip access-group 101 in
```

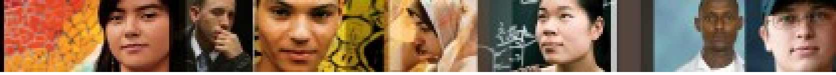
Configure Extended IPv4 ACLs

Creating Named Extended ACLs



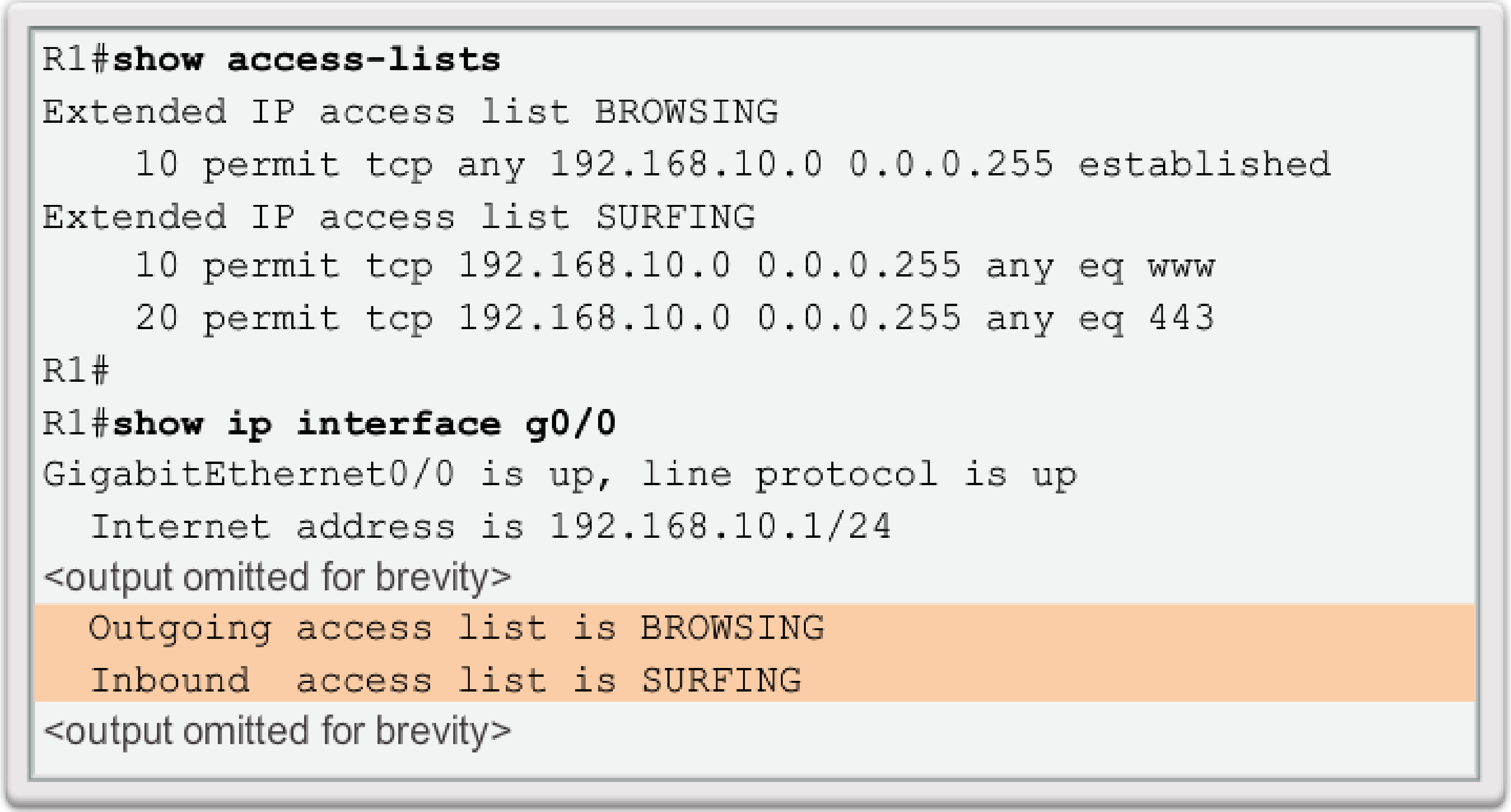


```
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config-ext-nacl)# exit
R1(config)# ip access-list extended BROWSING
R1(config-ext-nacl)# permit tcp any 192.168.10.0 0.0.0.255 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0
R1(config-if)# ip access-group SURFING in
R1(config-if)# ip access-group BROWSING out
```

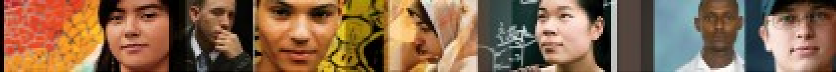


Configure Extended IPv4 ACLs

Verifying Extended ACLs



```
R1#show access-lists
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1#
R1#show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 192.168.10.1/24
<output omitted for brevity>
  Outgoing access list is BROWSING
  Inbound  access list is SURFING
<output omitted for brevity>
```



Configure Extended IPv4 ACLs

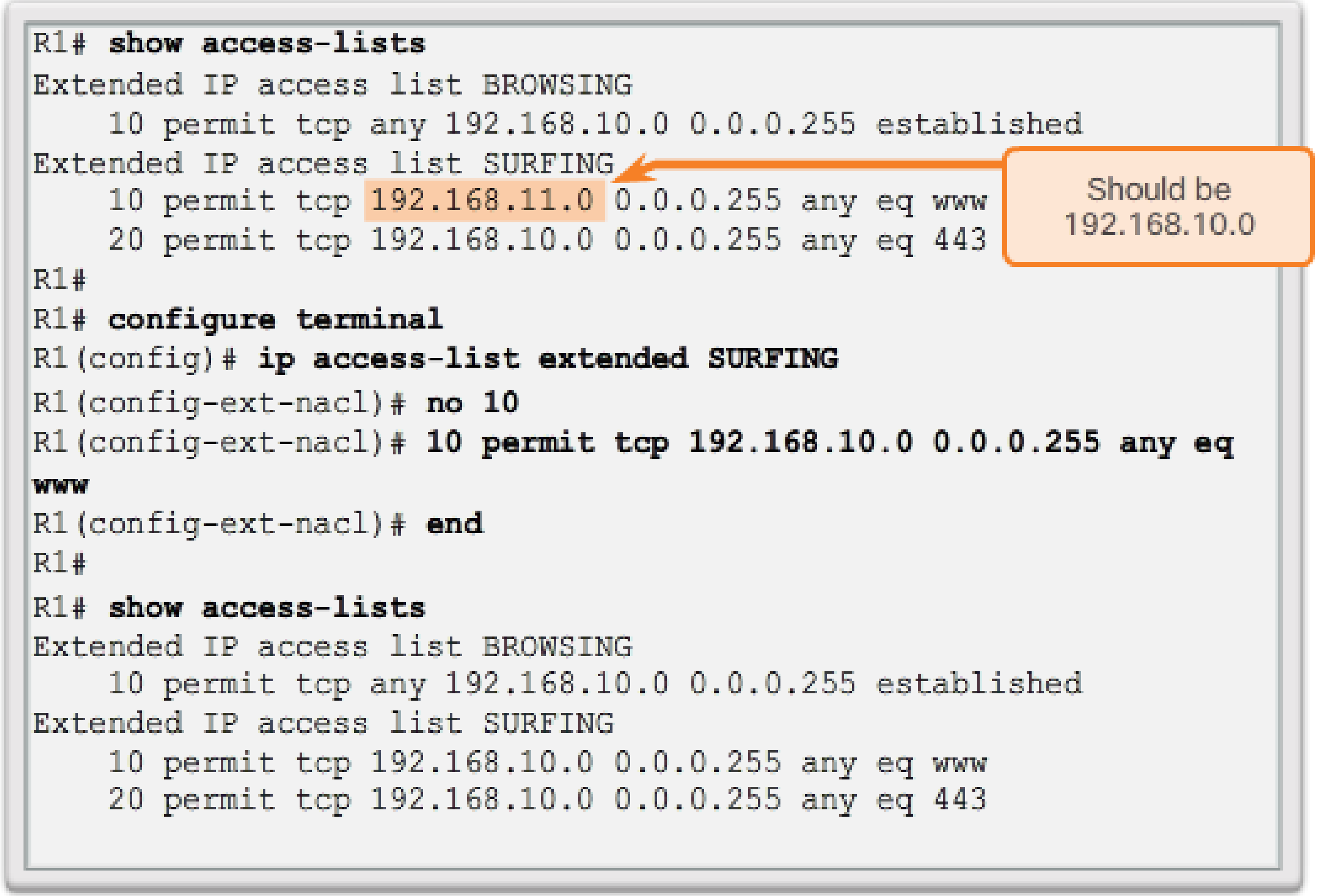
Editing Extended ACLs

- Editing an extended ACL can be accomplished using the same process as editing a standard. An extended ACL can be modified using:
  - Method 1 - Text editor
    - The ACL is copied and pasted into the text editor where the changes are made. The current access list is removed using the **no access-list** command. The modified ACL is then pasted back into the configuration.
  - Method 2 – Sequence numbers
    - Sequence numbers can be used to delete or insert an ACL statement.

Configure Extended IPv4 ACLs

Editing Extended ACLs cont…

- Editing an extended ACL via Sequence Numbers:



```
R1# show access-lists
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
    10 permit tcp 192.168.11.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1#
R1# configure terminal
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# no 10
R1(config-ext-nacl)# 10 permit tcp 192.168.10.0 0.0.0.255 any eq
www
R1(config-ext-nacl)# end
R1#
R1# show access-lists
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
```

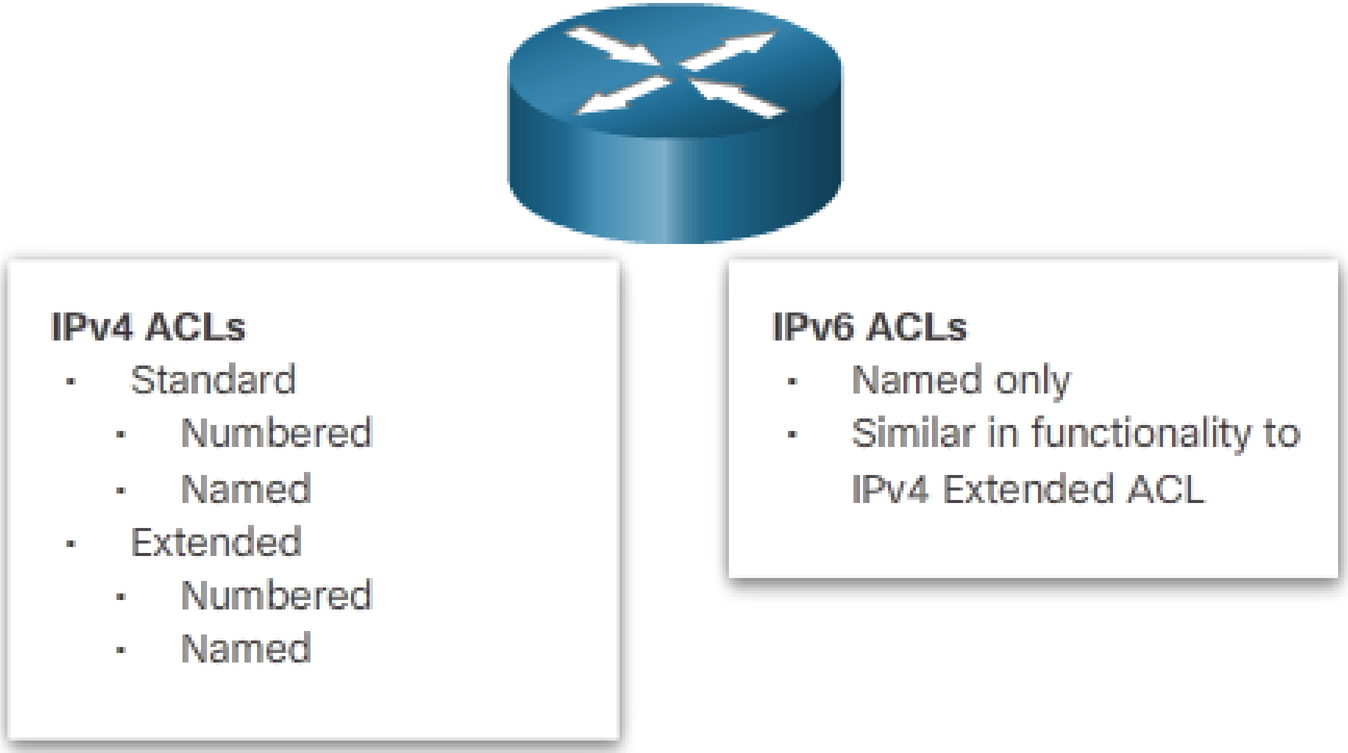
Should be 192.168.10.0

4.3 IPv6 ACLs

Cisco | Networking Academy®
Mind Wide Open™

IPv6 ACL Creation

Types of IPv6 ACLs

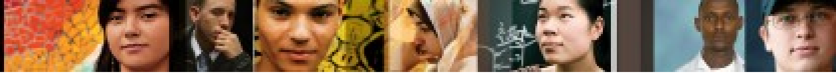


IPv4 ACLs

- Standard
  - Numbered
  - Named
- Extended
  - Numbered
  - Named

IPv6 ACLs

- Named only
- Similar in functionality to IPv4 Extended ACL



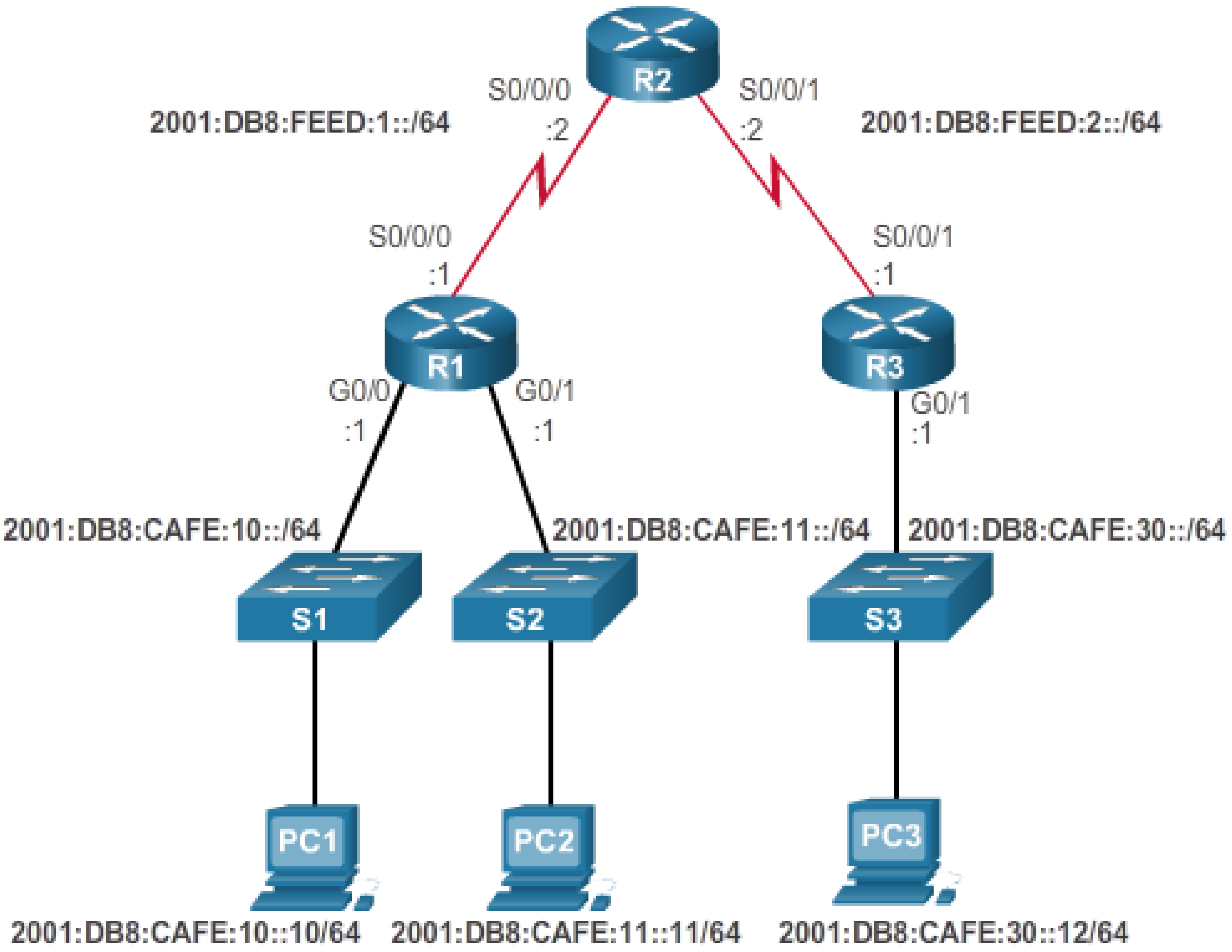
IPv6 ACL Creation

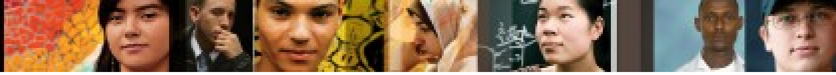
Comparing IPv4 and IPv6 ACLs

Although IPv4 and IPv6 ACLs are very similar, there are three significant differences between them.

- Applying an IPv6 ACL
  - IPv6 uses the **`ipv6 traffic-filter`** command to perform the same function for IPv6 interfaces.
- No Wildcard Masks
  - The prefix-length is used to indicate how much of an IPv6 source or destination address should be matched.
- Additional Default Statements
  - **`permit icmp any any nd-na`**
  - **`permit icmp any any nd-ns`**

Configuring IPv6 Topology

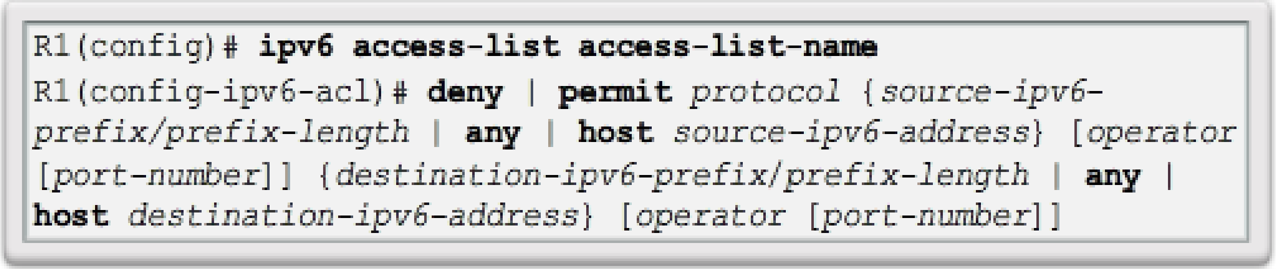




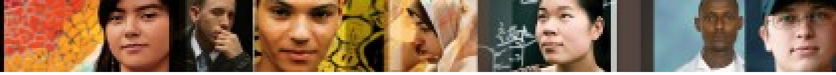
Configuring IPv6 ACLs

There are three basic steps to configure an IPv6 ACL:

1. From global configuration mode, use the **`ipv6 access-list`** *`name`* command to create an IPv6 ACL.
2. From the named ACL configuration mode, use **`permit`** or **`deny`** statements to specify one or more conditions to determine if a packet is forwarded or dropped.
3. Return to privileged EXEC mode



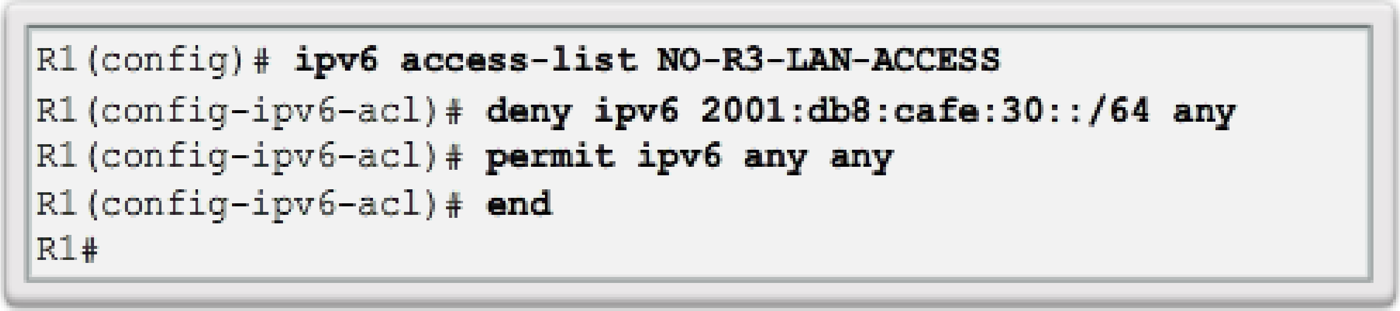
```
R1(config)# ipv6 access-list access-list-name
R1(config-ipv6-acl)# deny | permit protocol {source-ipv6-
prefix/prefix-length | any | host source-ipv6-address} [operator
[port-number]] {destination-ipv6-prefix/prefix-length | any |
host destination-ipv6-address} [operator [port-number]]
```



Configuring IPv6 ACLs

Configuring IPv6 ACLs cont…

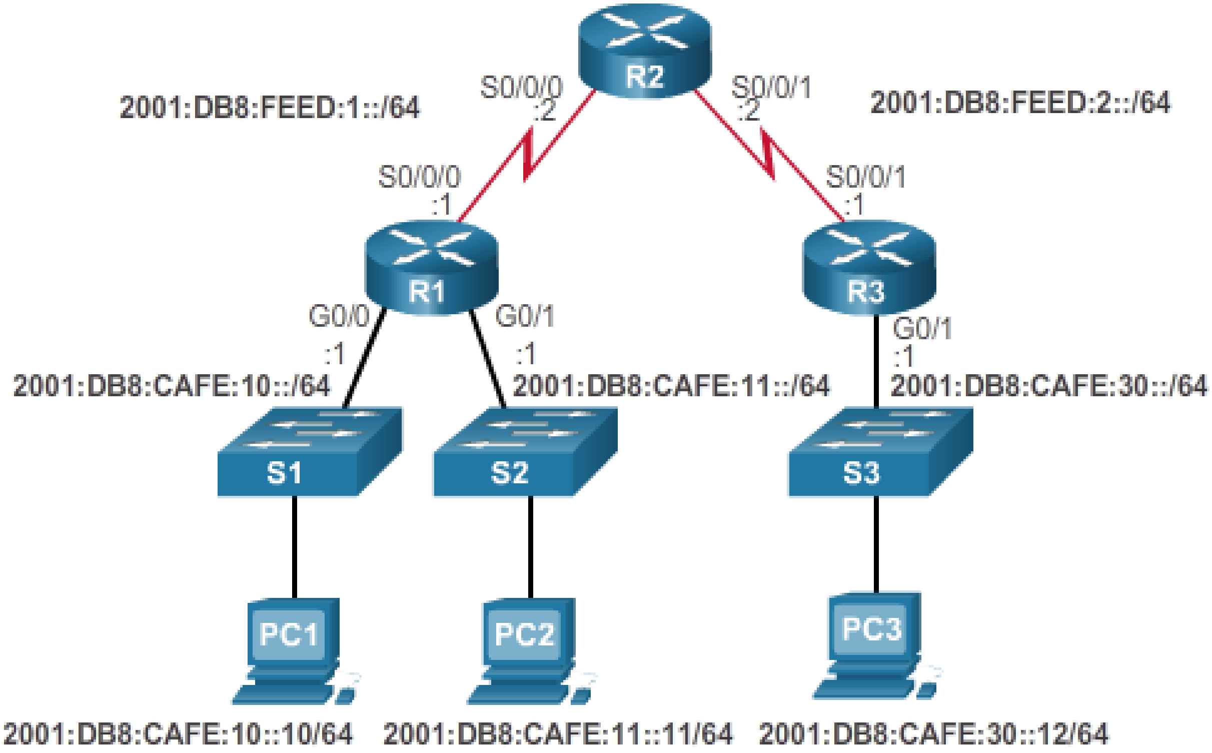
- This IPv6 ACL does the following:
  - The first statement names the IPv6 access list NO-R3-LAN-ACCESS.
  - The second statement denies all IPv6 packets from the 2001:DB8:CAFE:30::/64 destined for any IPv6 network.
  - The third statement allows all other IPv6 packets.

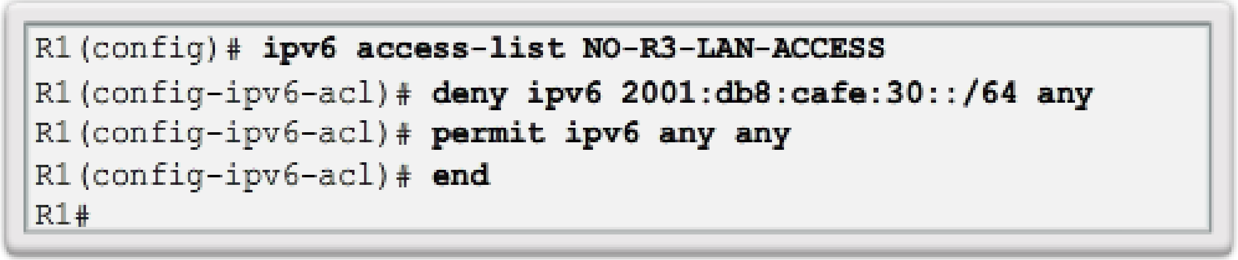


```
R1(config)# ipv6 access-list NO-R3-LAN-ACCESS
R1(config-ipv6-acl)# deny ipv6 2001:db8:cafe:30::/64 any
R1(config-ipv6-acl)# permit ipv6 any any
R1(config-ipv6-acl)# end
R1#
```

Configuring IPv6 ACLs

Configuring IPv6 ACLs cont…

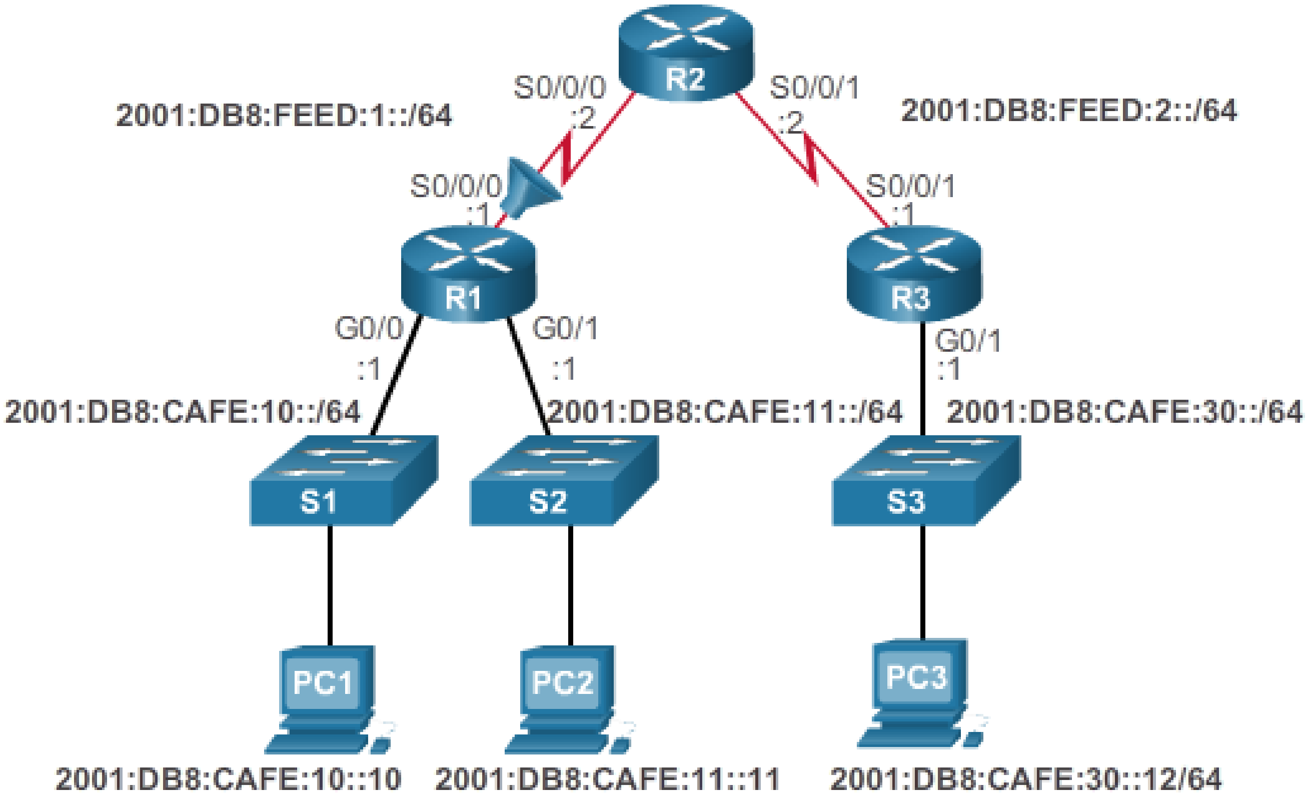




```
R1(config)# ipv6 access-list NO-R3-LAN-ACCESS
R1(config-ipv6-acl)# deny ipv6 2001:db8:cafe:30::/64 any
R1(config-ipv6-acl)# permit ipv6 any any
R1(config-ipv6-acl)# end
R1#
```

Configuring IPv6 ACLs

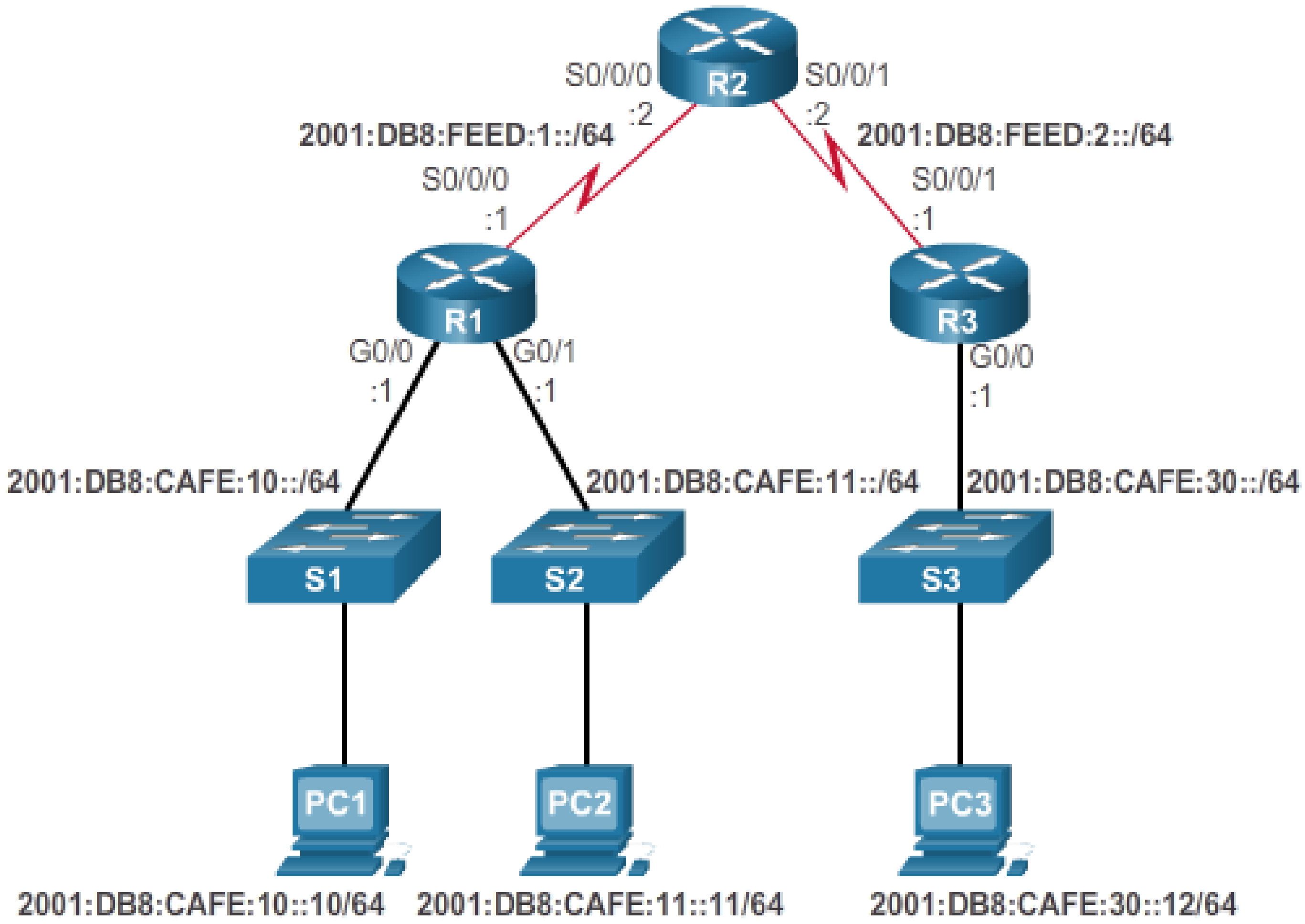
Applying an IPv6 ACL to an Interface



```
R1(config)# interface s0/0/0
R1(config-if)# ipv6 traffic-filter NO-R3-LAN-ACCESS in
```

Configuring IPv6 ACLs

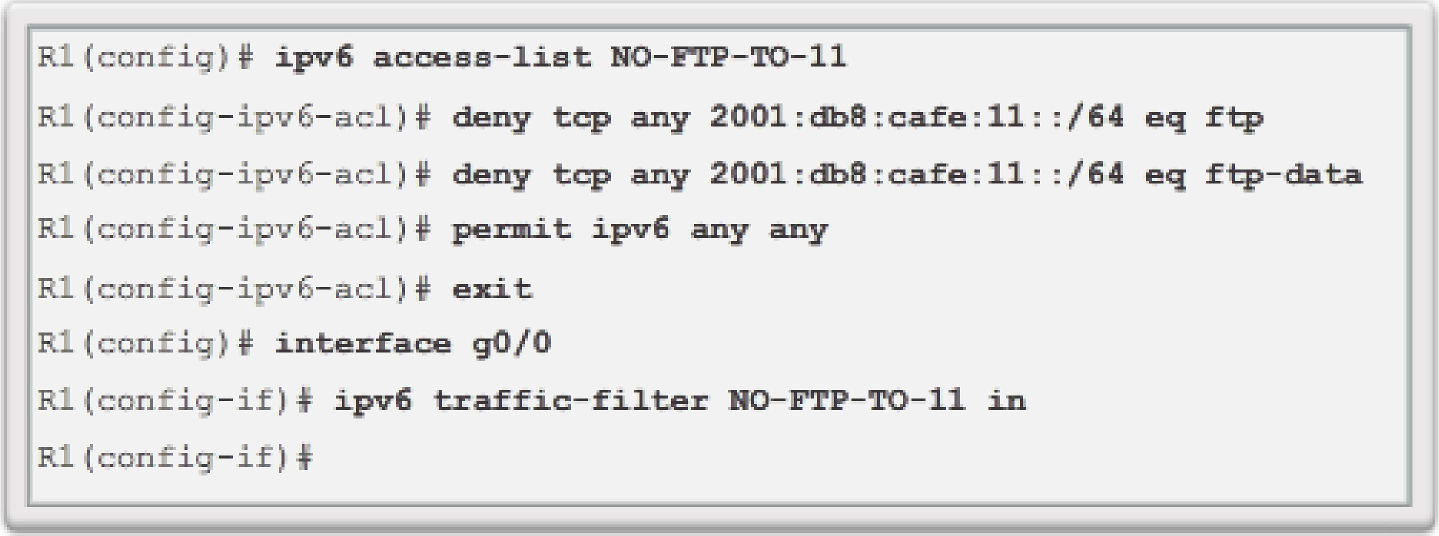
IPv6 ACL Examples



Configuring IPv6 ACLs

IPv6 ACL Examples cont…

- Router R1 is configured with an IPv6 access list to deny FTP traffic to 2001:DB8:CAFE:11::/64. Ports for both FTP data (port 20) and FTP control (port 21) need to be blocked.
- Because the filter is applied inbound on the G0/0 interface on R1, only traffic from the 2001:DB8:CAFE:10::/64 network will be denied.



```
R1(config)# ipv6 access-list NO-FTP-TO-11
R1(config-ipv6-acl)# deny tcp any 2001:db8:cafe:11::/64 eq ftp
R1(config-ipv6-acl)# deny tcp any 2001:db8:cafe:11::/64 eq ftp-data
R1(config-ipv6-acl)# permit ipv6 any any
R1(config-ipv6-acl)# exit
R1(config)# interface g0/0
R1(config-if)# ipv6 traffic-filter NO-FTP-TO-11 in
R1(config-if)#
```

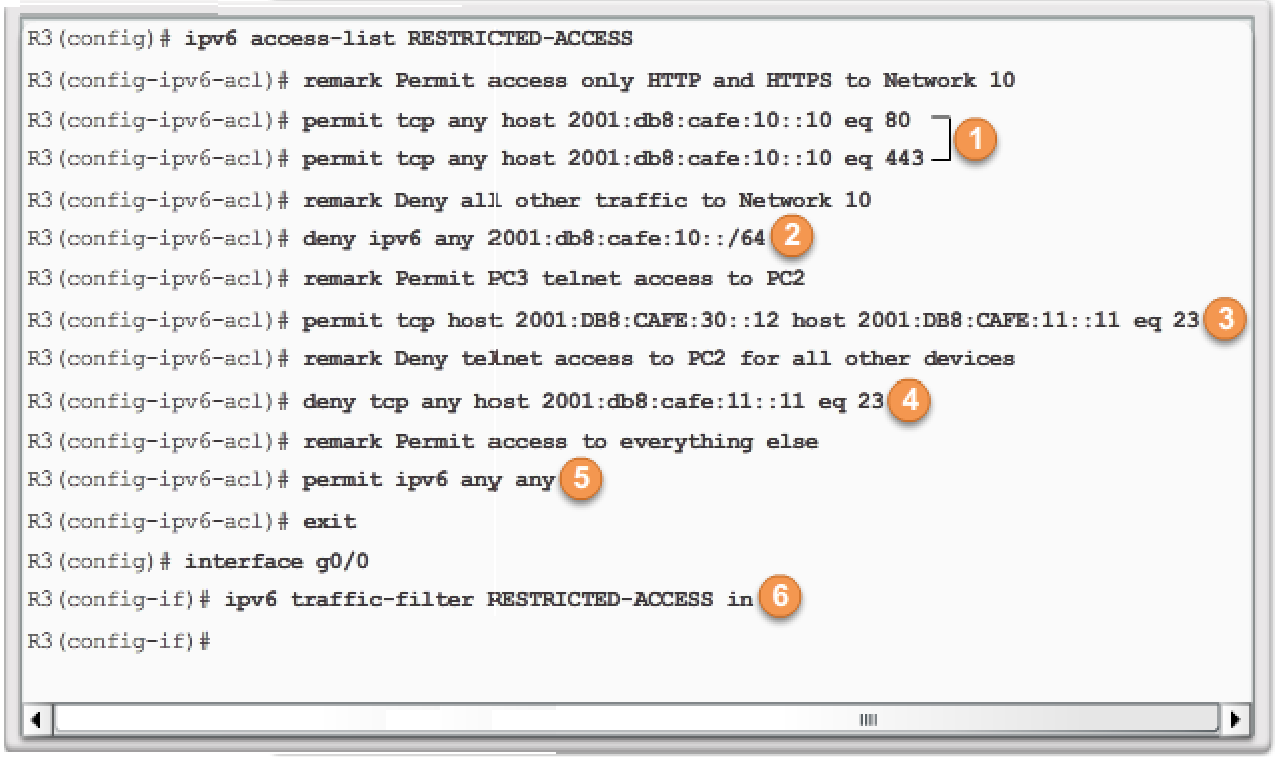
Configuring IPv6 ACLs

IPv6 ACL Examples cont…

1. The first two permit statements allow access from any device to the web server at 2001:DB8:CAFE:10::10.

2. All other devices are denied access to network 2001:DB8:CAFE:10::/64.

3. PC3 at 2001:DB8:CAFE:30::12 is permitted Telnet access to PC2 which has the IPv6 address 2001:DB8:CAFE:11::11.



```
R3(config)# ipv6 access-list RESTRICTED-ACCESS
R3(config-ipv6-acl)# remark Permit access only HTTP and HTTPS to Network 10
R3(config-ipv6-acl)# permit tcp any host 2001:db8:cafe:10::10 eq 80
R3(config-ipv6-acl)# permit tcp any host 2001:db8:cafe:10::10 eq 443
R3(config-ipv6-acl)# remark Deny all other traffic to Network 10
R3(config-ipv6-acl)# deny ipv6 any 2001:db8:cafe:10::/64
R3(config-ipv6-acl)# remark Permit PC3 telnet access to PC2
R3(config-ipv6-acl)# permit tcp host 2001:DB8:CAFE:30::12 host 2001:DB8:CAFE:11::11 eq 23
R3(config-ipv6-acl)# remark Deny telnet access to PC2 for all other devices
R3(config-ipv6-acl)# deny tcp any host 2001:db8:cafe:11::11 eq 23
R3(config-ipv6-acl)# remark Permit access to everything else
R3(config-ipv6-acl)# permit ipv6 any any
R3(config-ipv6-acl)# exit
R3(config)# interface g0/0
R3(config-if)# ipv6 traffic-filter RESTRICTED-ACCESS in
R3(config-if)#
```

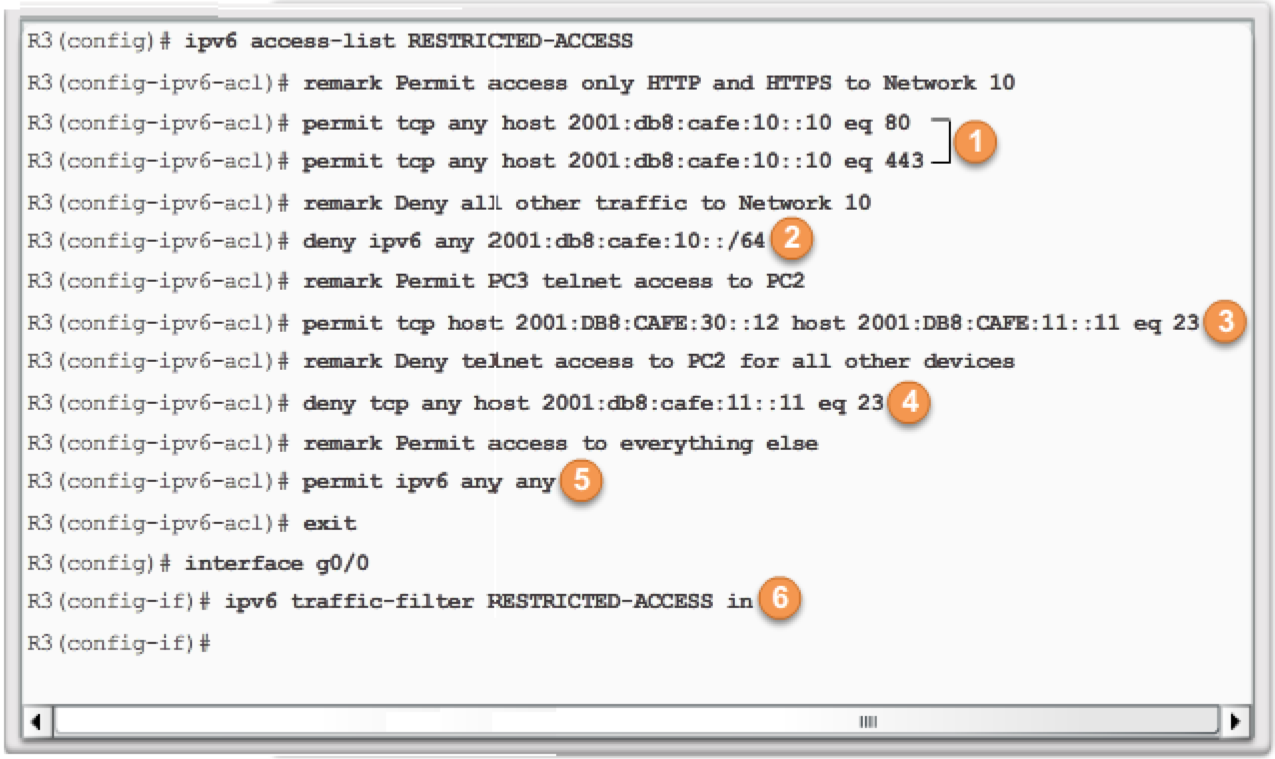
Configuring IPv6 ACLs

IPv6 ACL Examples cont…

4. All other devices are denied Telnet access to PC2.

5. All other IPv6 traffic is permitted to all other destinations.

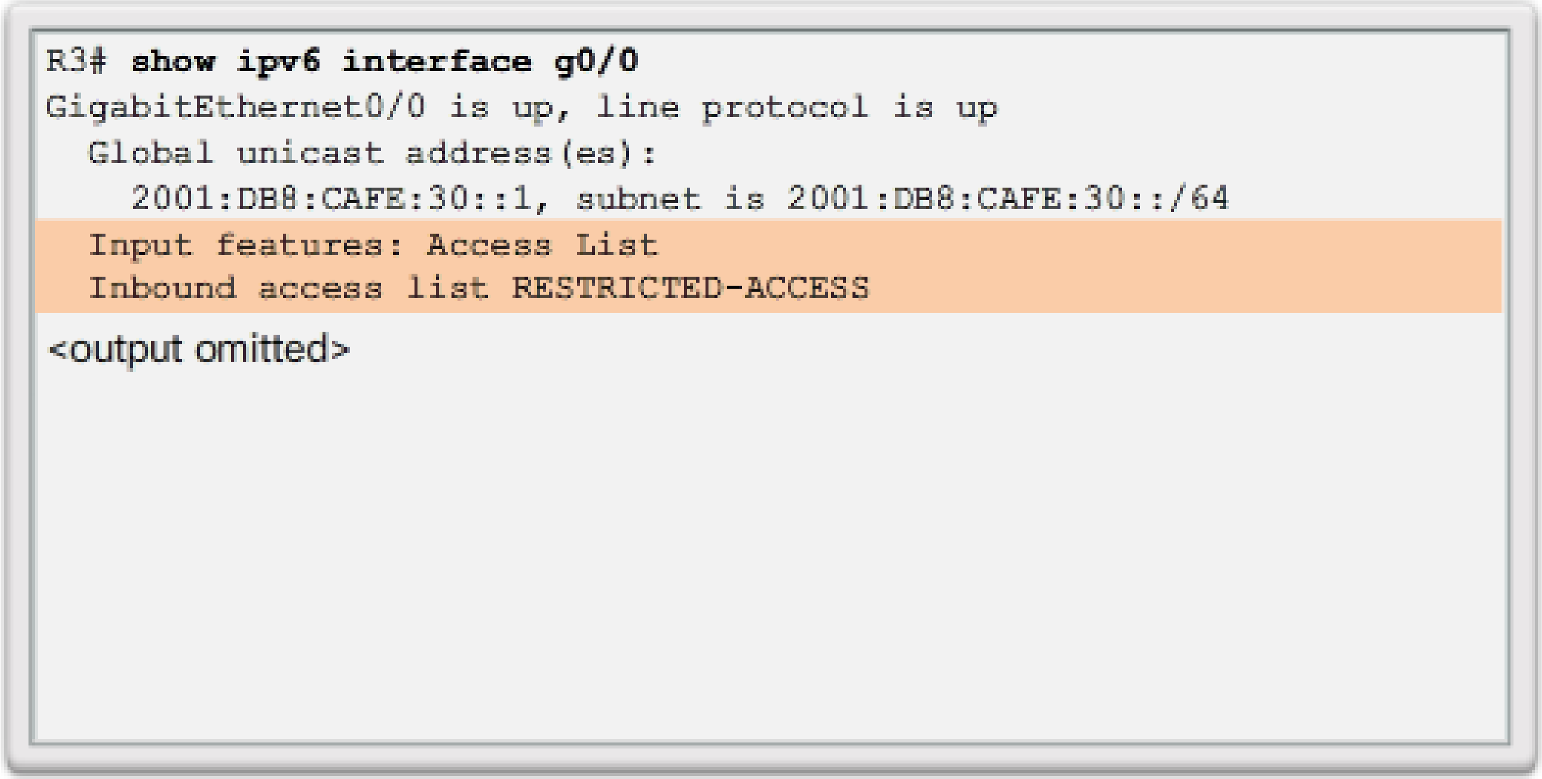
6. The IPv6 access list is applied to interface G0/0 in the inbound direction, so only the 2001:DB8:CAFE:30::/64 network is affected.



```
R3(config)# ipv6 access-list RESTRICTED-ACCESS
R3(config-ipv6-acl)# remark Permit access only HTTP and HTTPS to Network 10
R3(config-ipv6-acl)# permit tcp any host 2001:db8:cafe:10::10 eq 80
R3(config-ipv6-acl)# permit tcp any host 2001:db8:cafe:10::10 eq 443
R3(config-ipv6-acl)# remark Deny all other traffic to Network 10
R3(config-ipv6-acl)# deny ipv6 any 2001:db8:cafe:10::/64
R3(config-ipv6-acl)# remark Permit PC3 telnet access to PC2
R3(config-ipv6-acl)# permit tcp host 2001:DB8:CAFE:30::12 host 2001:DB8:CAFE:11::11 eq 23
R3(config-ipv6-acl)# remark Deny telnet access to PC2 for all other devices
R3(config-ipv6-acl)# deny tcp any host 2001:db8:cafe:11::11 eq 23
R3(config-ipv6-acl)# remark Permit access to everything else
R3(config-ipv6-acl)# permit ipv6 any any
R3(config-ipv6-acl)# exit
R3(config)# interface g0/0
R3(config-if)# ipv6 traffic-filter RESTRICTED-ACCESS in
R3(config-if)#
```

Configuring IPv6 ACLs

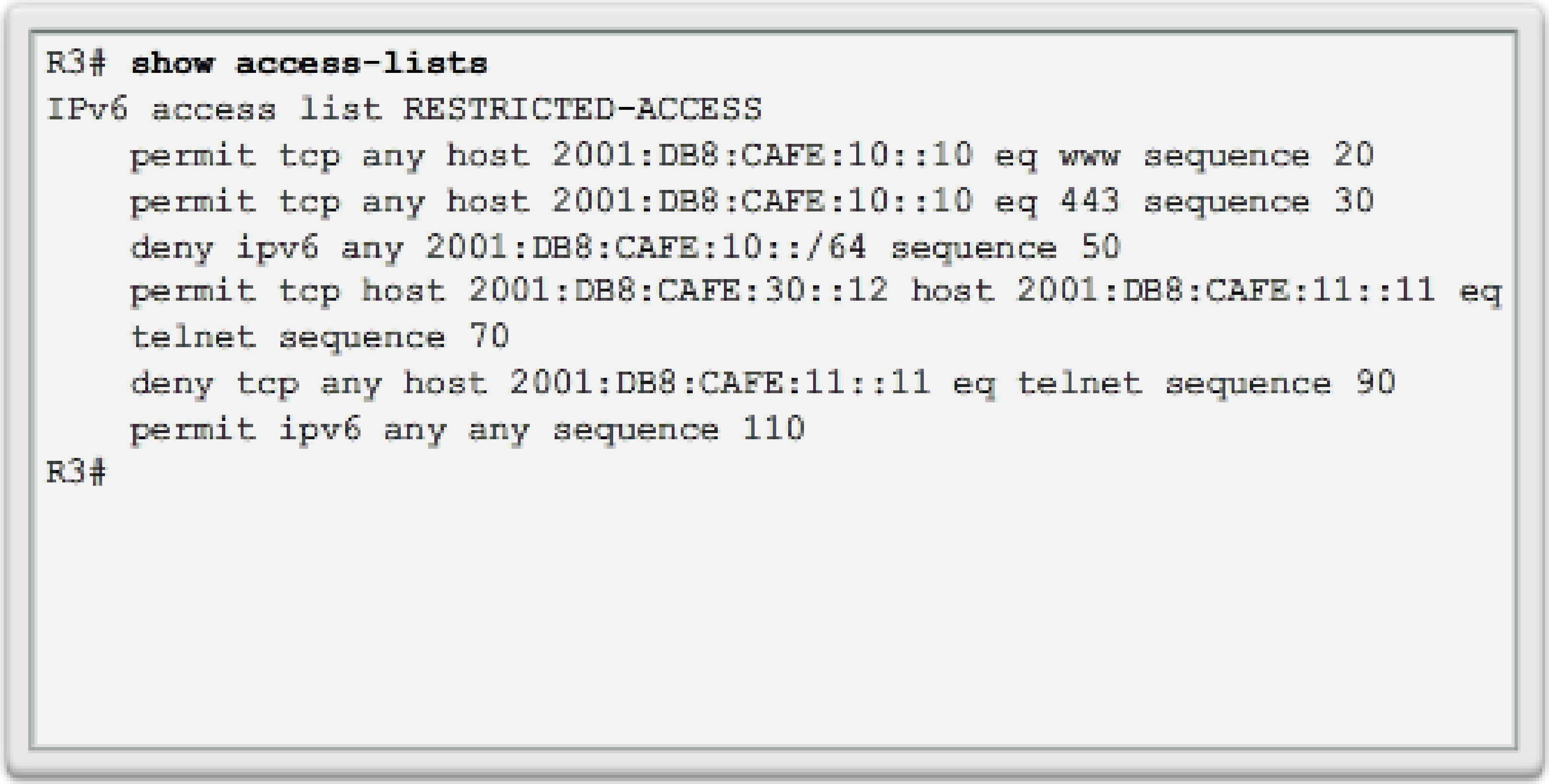
Verifying IPv6 ACLs



```
R3# show ipv6 interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Global unicast address(es):
    2001:DB8:CAFE:30::1, subnet is 2001:DB8:CAFE:30::/64
  Input features: Access List
  Inbound access list RESTRICTED-ACCESS
<output omitted>
```

Configuring IPv6 ACLs

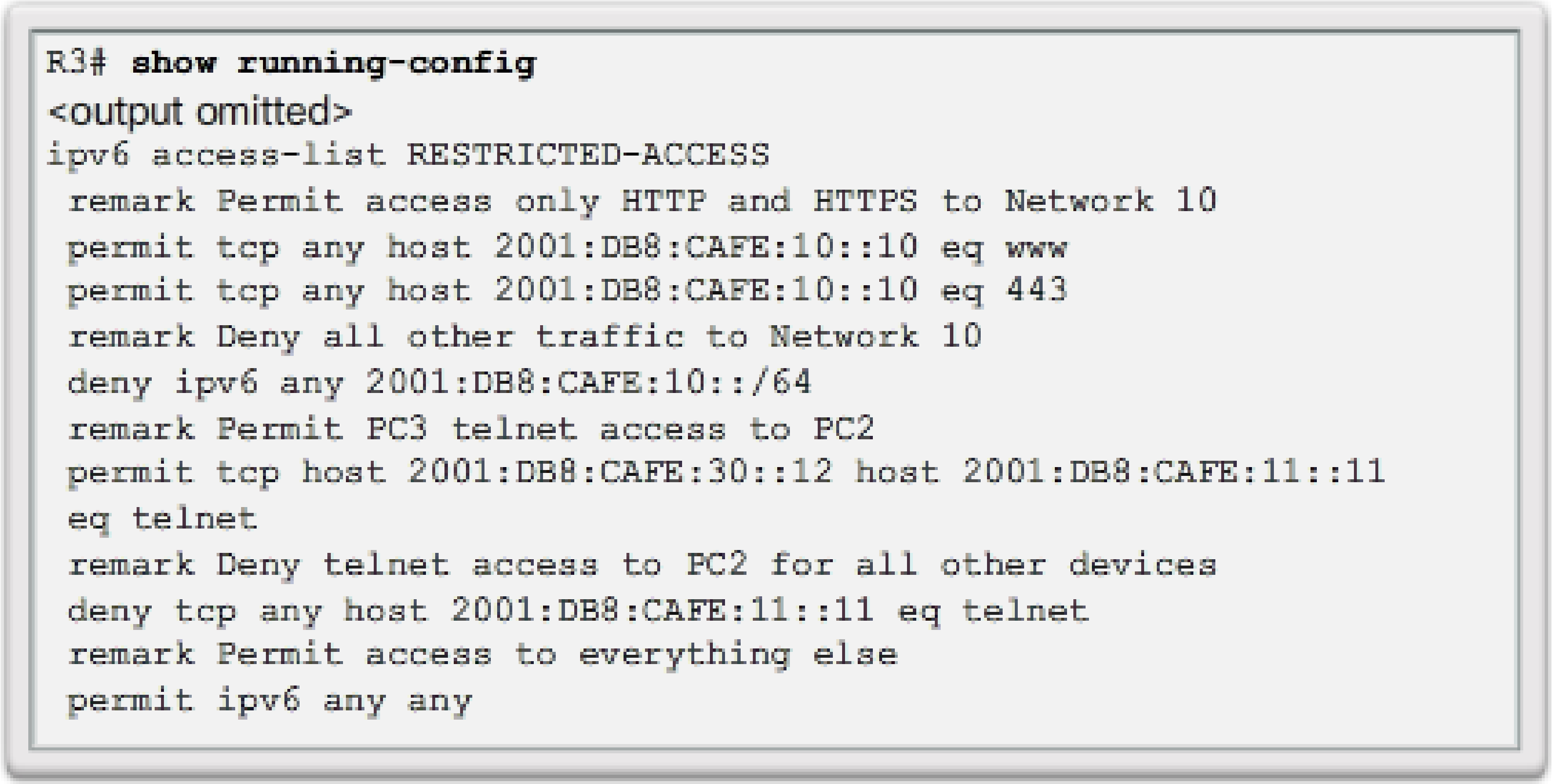
Verifying IPv6 ACLs cont…



```
R3# show access-lists
IPv6 access list RESTRICTED-ACCESS
    permit tcp any host 2001:DB8:CAFE:10::10 eq www sequence 20
    permit tcp any host 2001:DB8:CAFE:10::10 eq 443 sequence 30
    deny ipv6 any 2001:DB8:CAFE:10::/64 sequence 50
    permit tcp host 2001:DB8:CAFE:30::12 host 2001:DB8:CAFE:11::11 eq
    telnet sequence 70
    deny tcp any host 2001:DB8:CAFE:11::11 eq telnet sequence 90
    permit ipv6 any any sequence 110
R3#
```

Configuring IPv6 ACLs

Verifying IPv6 ACLs cont…



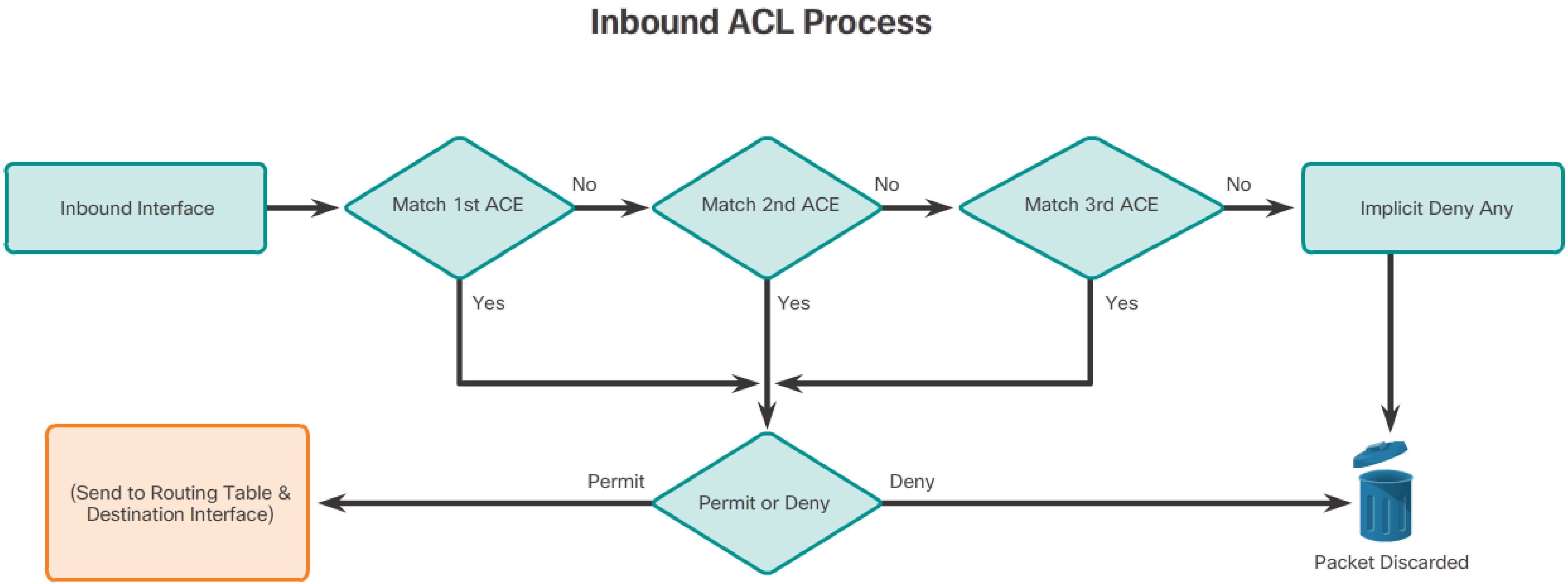
```
R3# show running-config
<output omitted>
ipv6 access-list RESTRICTED-ACCESS
 remark Permit access only HTTP and HTTPS to Network 10
 permit tcp any host 2001:DB8:CAFE:10::10 eq www
 permit tcp any host 2001:DB8:CAFE:10::10 eq 443
 remark Deny all other traffic to Network 10
 deny ipv6 any 2001:DB8:CAFE:10::/64
 remark Permit PC3 telnet access to PC2
 permit tcp host 2001:DB8:CAFE:30::12 host 2001:DB8:CAFE:11::11
 eq telnet
 remark Deny telnet access to PC2 for all other devices
 deny tcp any host 2001:DB8:CAFE:11::11 eq telnet
 remark Permit access to everything else
 permit ipv6 any any
```

4.4 Troubleshoot ACLs

Cisco | Networking Academy®
Mind Wide Open™

Processing Packets with ACLs

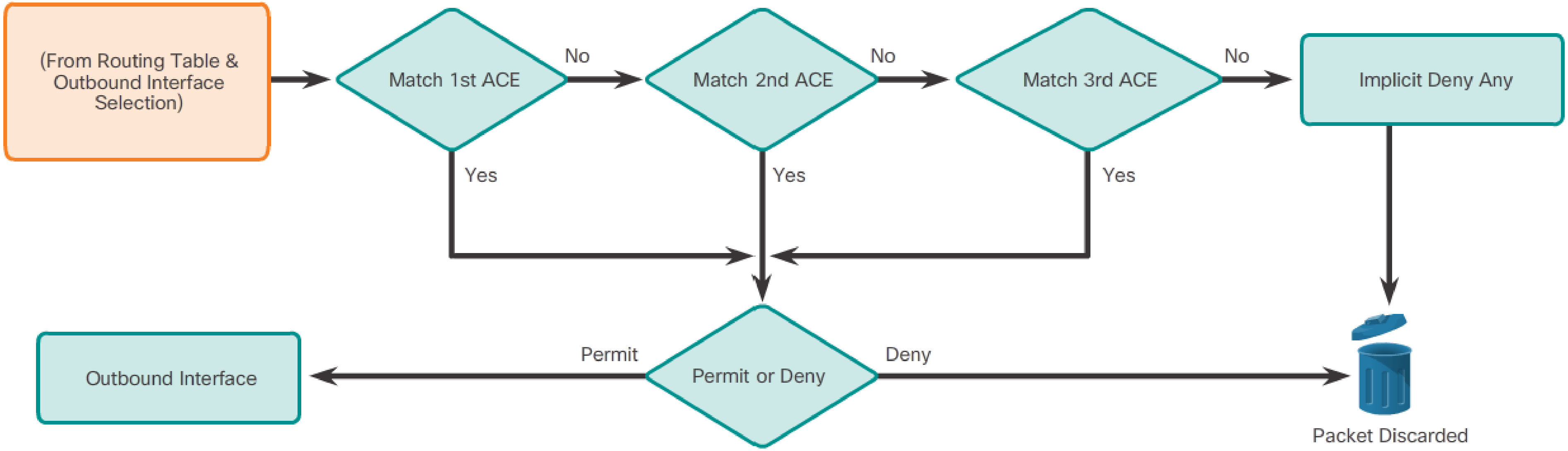
Inbound and Outbound ACL Logic



Processing Packets with ACLs

Inbound and Outbound ACL Logic

Outbound ACL Process



(From Routing Table & Outbound Interface Selection)

Match 1st ACE

Match 2nd ACE

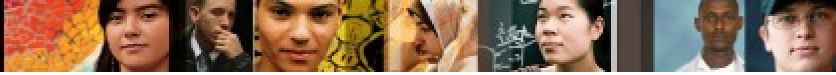
Match 3rd ACE

Implicit Deny Any

Outbound Interface

Permit or Deny

Packet Discarded



Processing Packets with ACLs

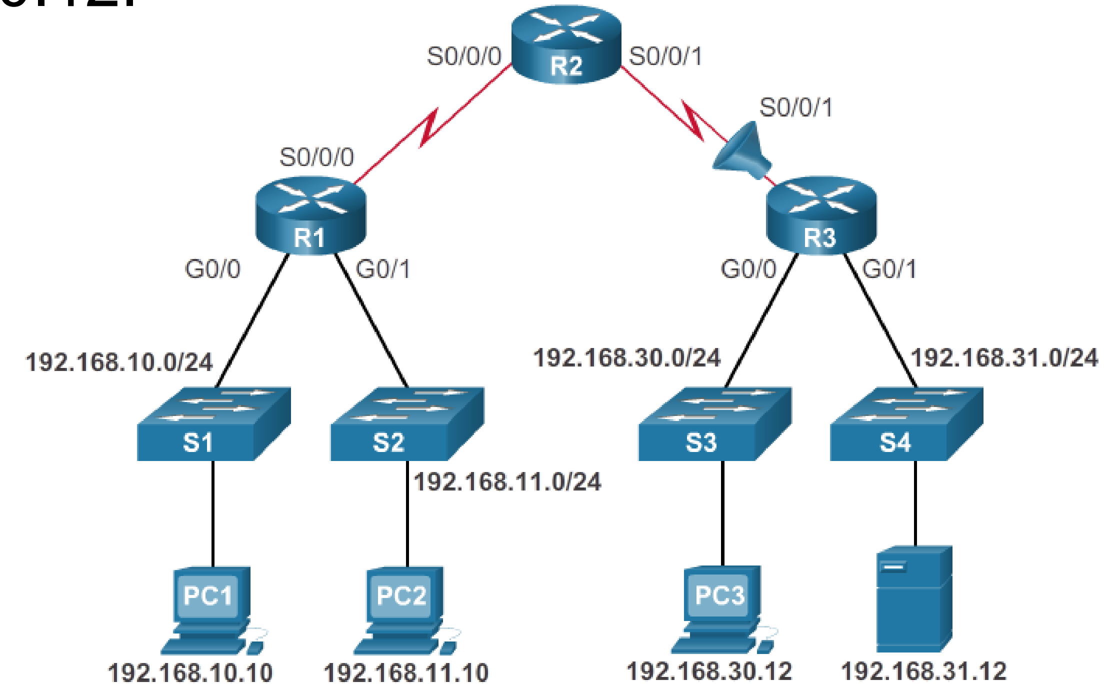
ACL Logic Operations

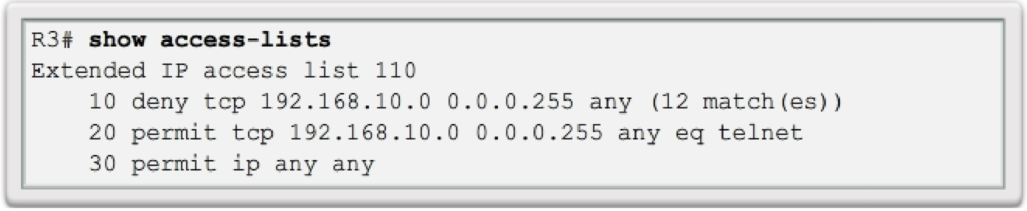
- As a frame enters an interface, the router checks to see whether the destination Layer 2 address matches its interface Layer 2 address, or whether the frame is a broadcast frame.
- If the frame address is accepted, the frame information is stripped off and the router checks for an ACL on the inbound interface.
- If an ACL exists, the packet is tested against the statements in the list.
- If the packet matches a statement, the packet is either permitted or denied.
- If the packet is accepted, it is then checked against routing table entries to determine the destination interface.
- If a routing table entry exists for the destination, the packet is then switched to the outgoing interface, otherwise the packet is dropped.
- Next, the router checks whether the outgoing interface has an ACL. If an ACL exists, the packet is tested against the statements in the list. If the packet matches a statement, it is either permitted or denied.
- If there is no ACL or the packet is permitted, the packet is encapsulated in the new Layer 2 protocol and forwarded out the interface to the next device.

Common ACL Errors

Troubleshooting IPv4 ACLs- Example 1

- Host 192.168.10.10 has no Telnet connectivity with 192.168.30.12.



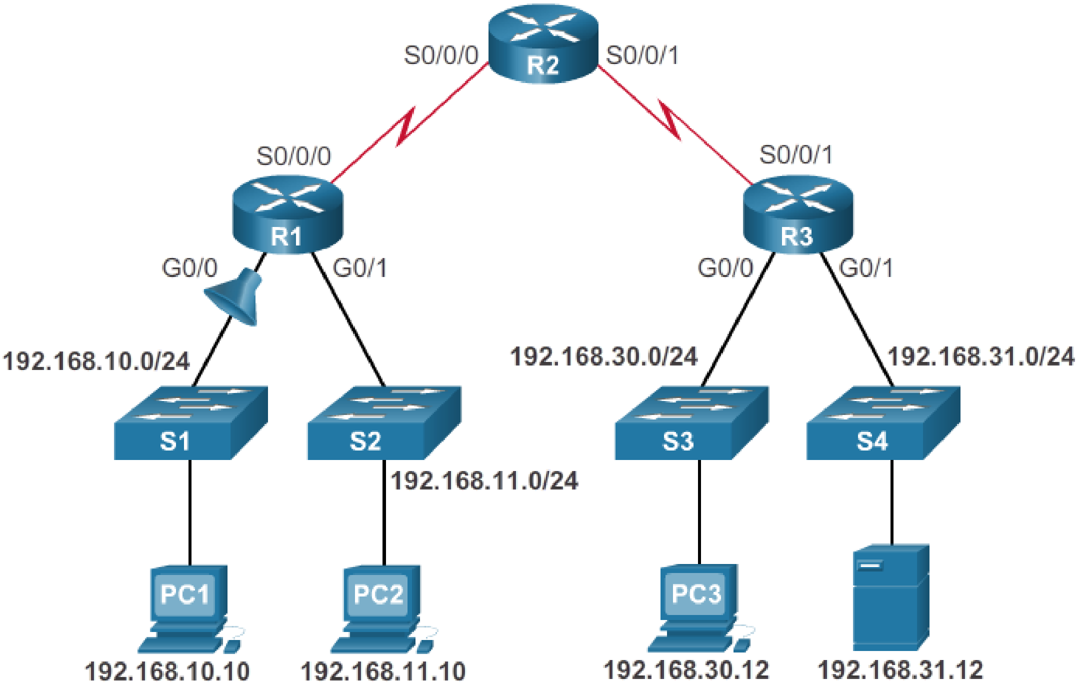


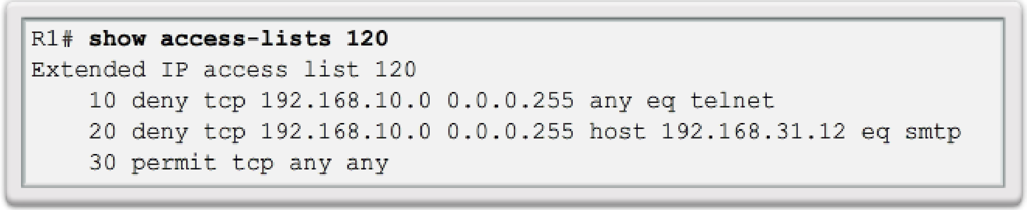
```
R3# show access-lists
Extended IP access list 110
    10 deny tcp 192.168.10.0 0.0.0.255 any (12 match(es))
    20 permit tcp 192.168.10.0 0.0.0.255 any eq telnet
    30 permit ip any any
```

Common ACL Errors

Troubleshooting IPv4 ACLs- Example 2

- The 192.168.10.0/24 network cannot use TFTP to connect to the 192.168.30.0/24 network.



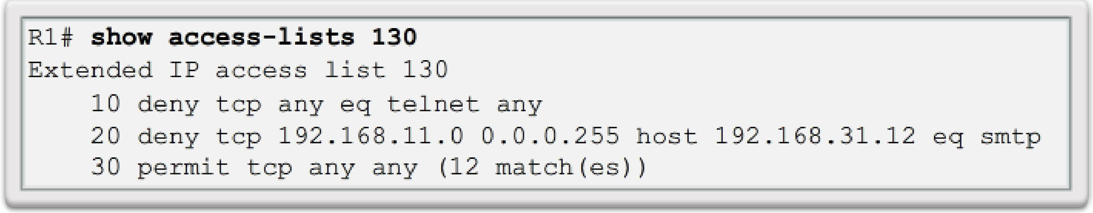


```
R1# show access-lists 120
Extended IP access list 120
    10 deny tcp 192.168.10.0 0.0.0.255 any eq telnet
    20 deny tcp 192.168.10.0 0.0.0.255 host 192.168.31.12 eq smtp
    30 permit tcp any any
```

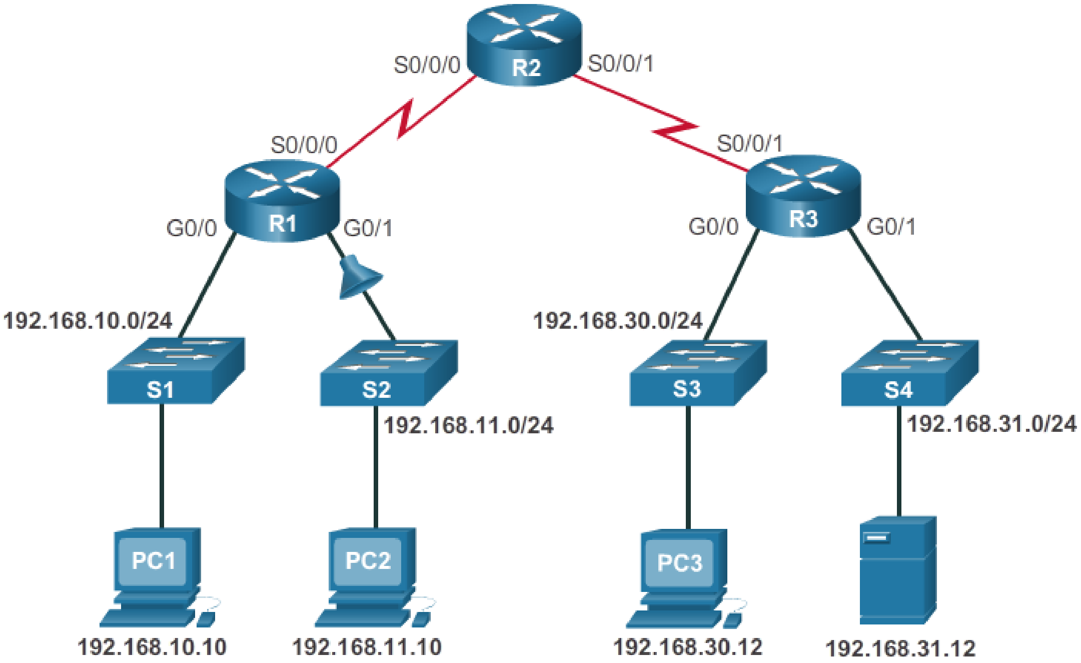
Common ACL Errors

Troubleshooting IPv4 ACLs- Example 3

- The 192.168.11.0/24 network can use Telnet to connect to 192.168.30.0/24, but this connection should not be allowed.



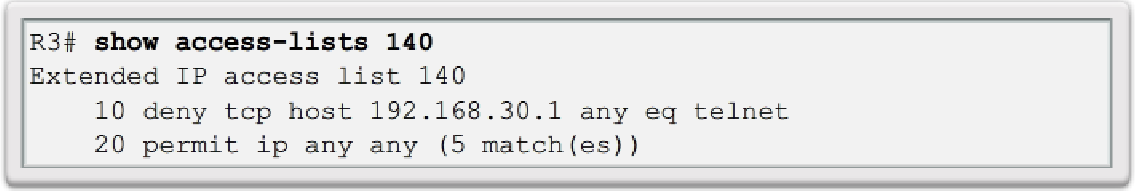
```
R1# show access-lists 130
Extended IP access list 130
    10 deny tcp any eq telnet any
    20 deny tcp 192.168.11.0 0.0.0.255 host 192.168.31.12 eq smtp
    30 permit tcp any any (12 match(es))
```



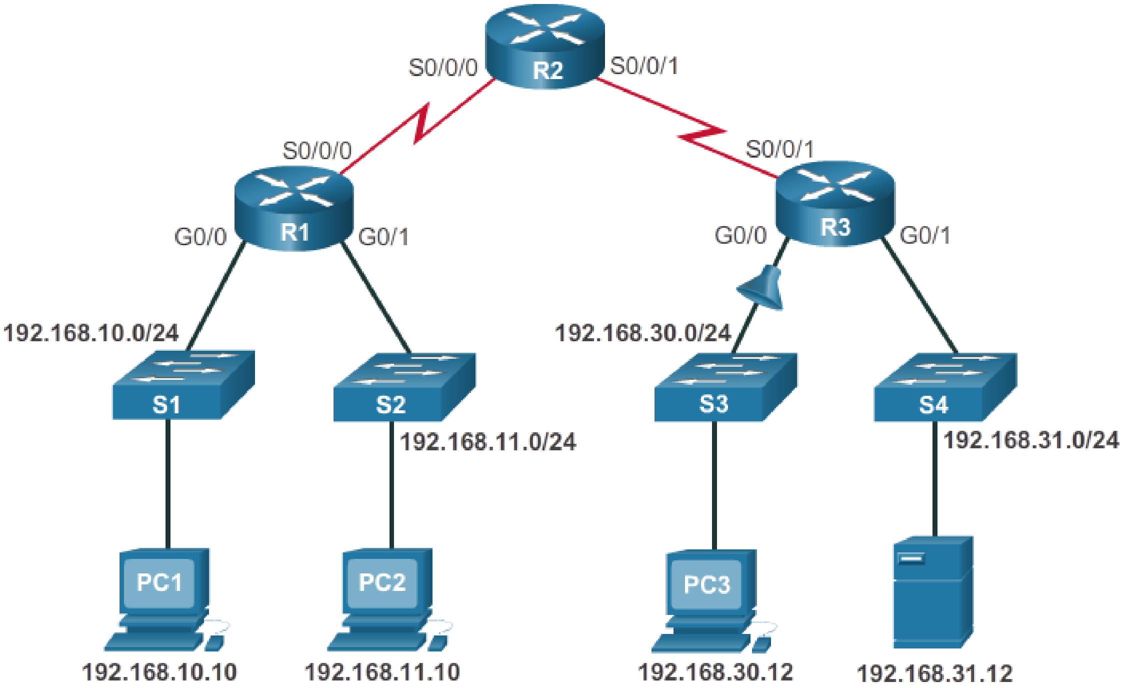
Common ACL Errors

Troubleshooting IPv4 ACLs- Example 4

- Host 192.168.30.12 is able to Telnet to connect to 192.168.31.12, but this connection should not be allowed.



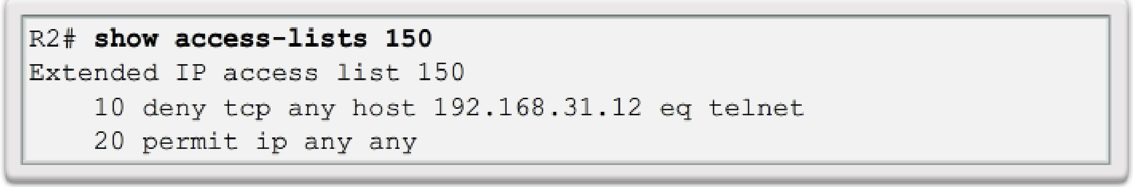
```
R3# show access-lists 140
Extended IP access list 140
    10 deny tcp host 192.168.30.1 any eq telnet
    20 permit ip any any (5 match(es))
```



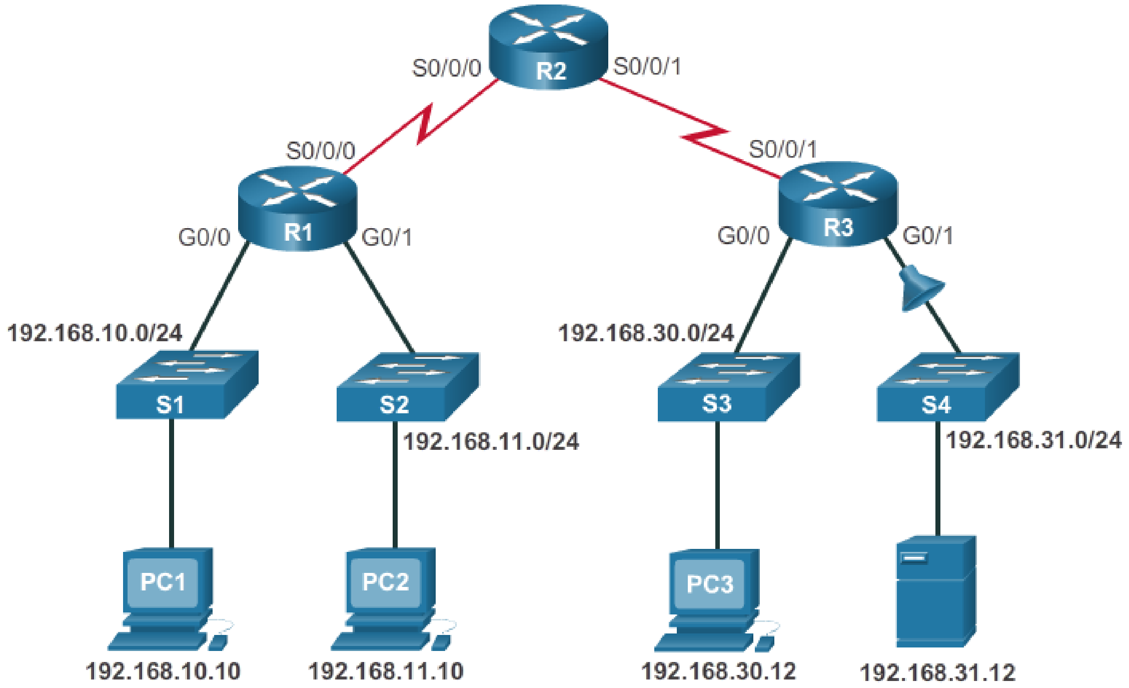
Common ACL Errors

Troubleshooting IPv4 ACLs- Example 5

- Host 192.168.30.12 can use Telnet to connect to 192.168.31.12, but this connection should not be allowed.



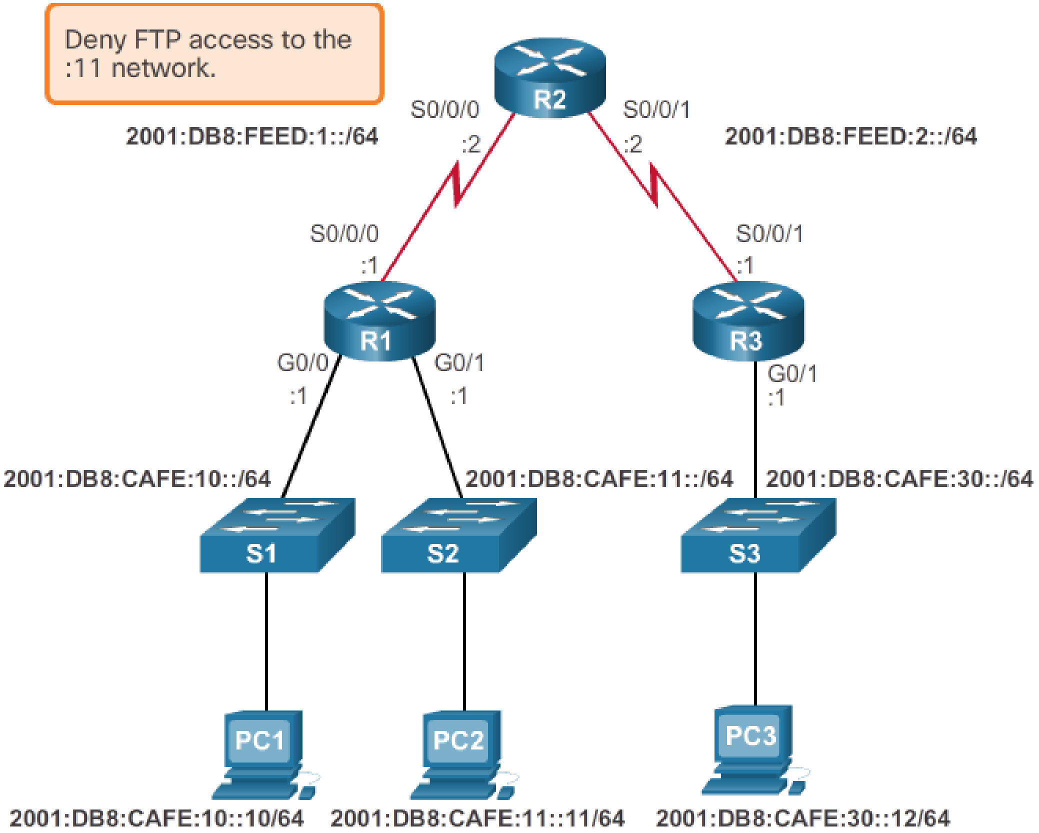
```
R2# show access-lists 150
Extended IP access list 150
    10 deny tcp any host 192.168.31.12 eq telnet
    20 permit ip any any
```



Common ACL Errors

Troubleshooting IPv6 ACLs- Example 1

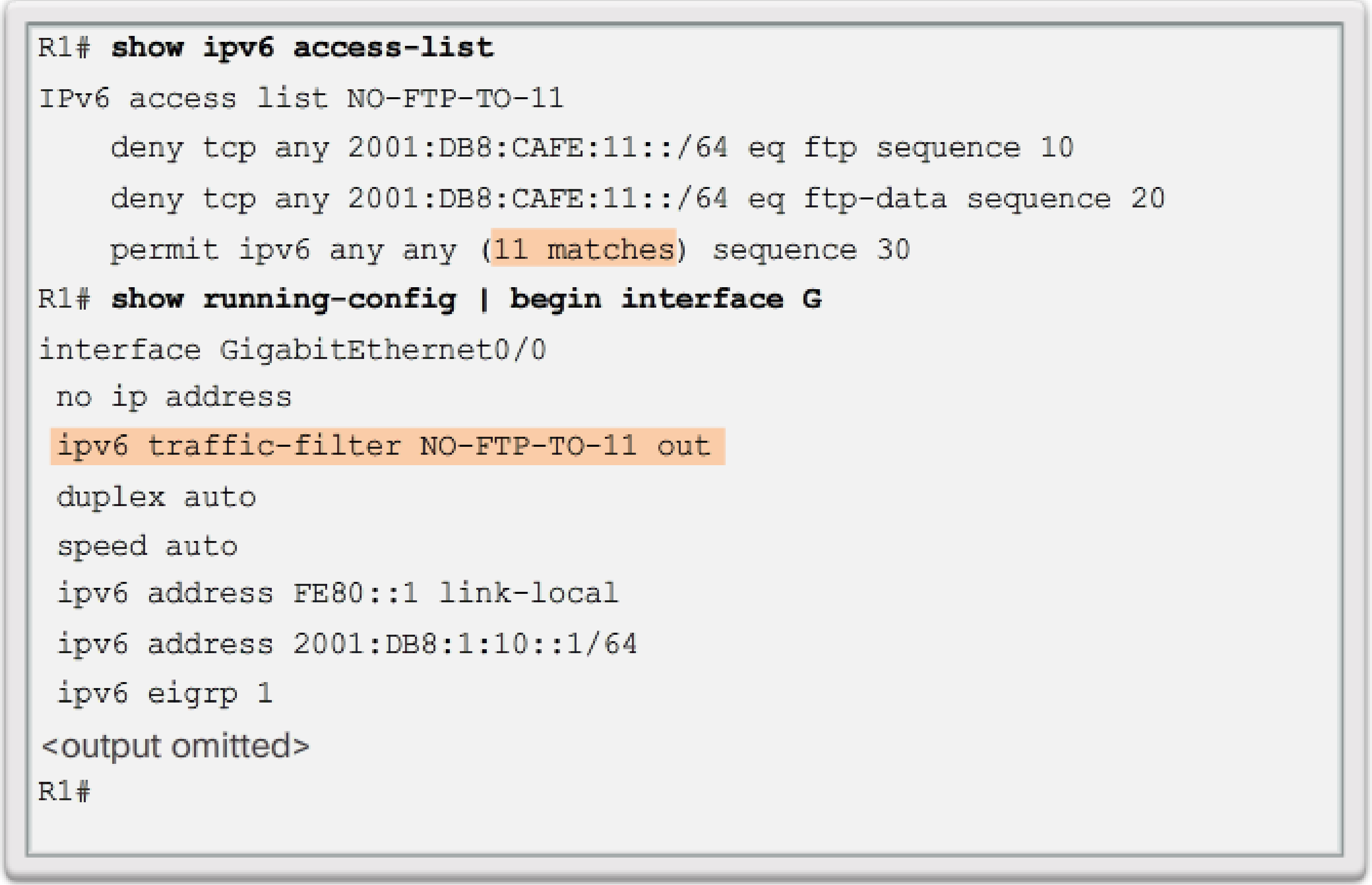
- R1 is configured with an IPv6 ACL to deny FTP access from the :10 network to the :11 network, but PC1 is still able to connect to the FTP server running on PC2.


Deny FTP access to the :11 network.

Common ACL Errors

Troubleshooting IPv6 ACLs- Example 1 cont…

Verify the IPv6 ACL Configuration and Application

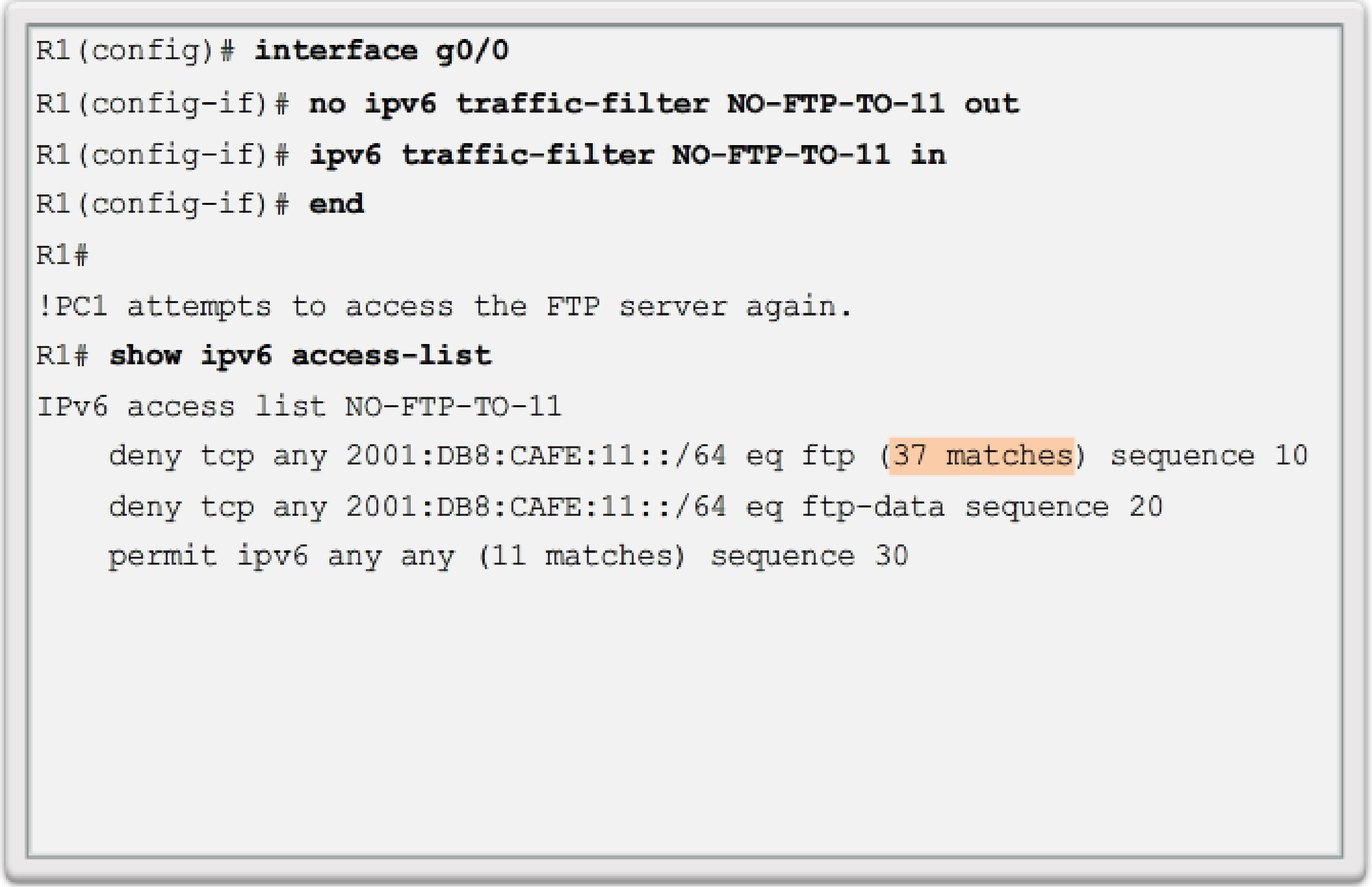


```
R1# show ipv6 access-list
IPv6 access list NO-FTP-TO-11
    deny tcp any 2001:DB8:CAFE:11::/64 eq ftp sequence 10
    deny tcp any 2001:DB8:CAFE:11::/64 eq ftp-data sequence 20
    permit ipv6 any any (11 matches) sequence 30
R1# show running-config | begin interface G
interface GigabitEthernet0/0
 no ip address
 ipv6 traffic-filter NO-FTP-TO-11 out
 duplex auto
 speed auto
 ipv6 address FE80::1 link-local
 ipv6 address 2001:DB8:1:10::1/64
 ipv6 eigrp 1
<output omitted>
R1#
```

Common ACL Errors

Troubleshooting IPv6 ACLs- Example 1 cont…

Correct and Verify the IPv6 ACL

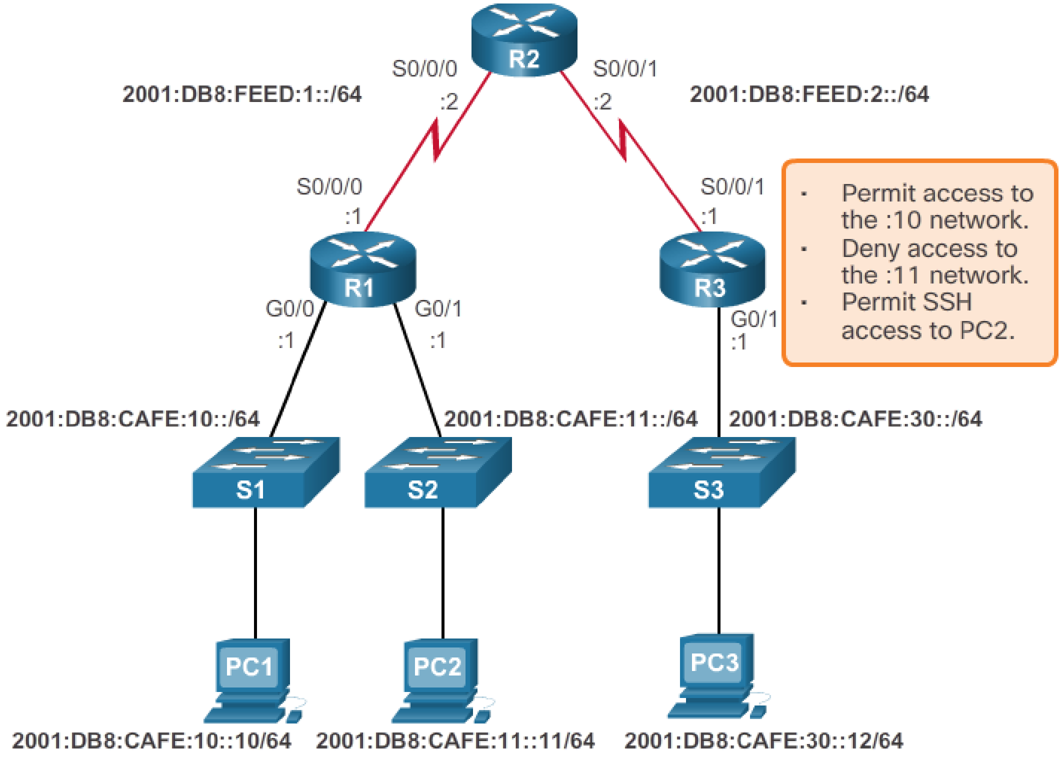


```
R1(config)# interface g0/0
R1(config-if)# no ipv6 traffic-filter NO-FTP-TO-11 out
R1(config-if)# ipv6 traffic-filter NO-FTP-TO-11 in
R1(config-if)# end
R1#
!PC1 attempts to access the FTP server again.
R1# show ipv6 access-list
IPv6 access list NO-FTP-TO-11
    deny tcp any 2001:DB8:CAFE:11::/64 eq ftp (37 matches) sequence 10
    deny tcp any 2001:DB8:CAFE:11::/64 eq ftp-data sequence 20
    permit ipv6 any any (11 matches) sequence 30
```

Common ACL Errors

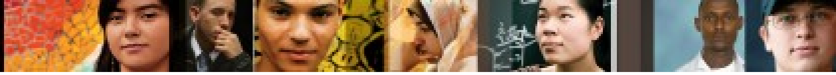
Troubleshooting IPv6 ACLs- Example 2

- R3 is configured with IPv6 ACL RESTRICTED-ACCESS that should enforce the following policy for the R3 LAN:



- Permit access to the :10 network.
- Deny access to the :11 network.
- Permit SSH access to PC2.

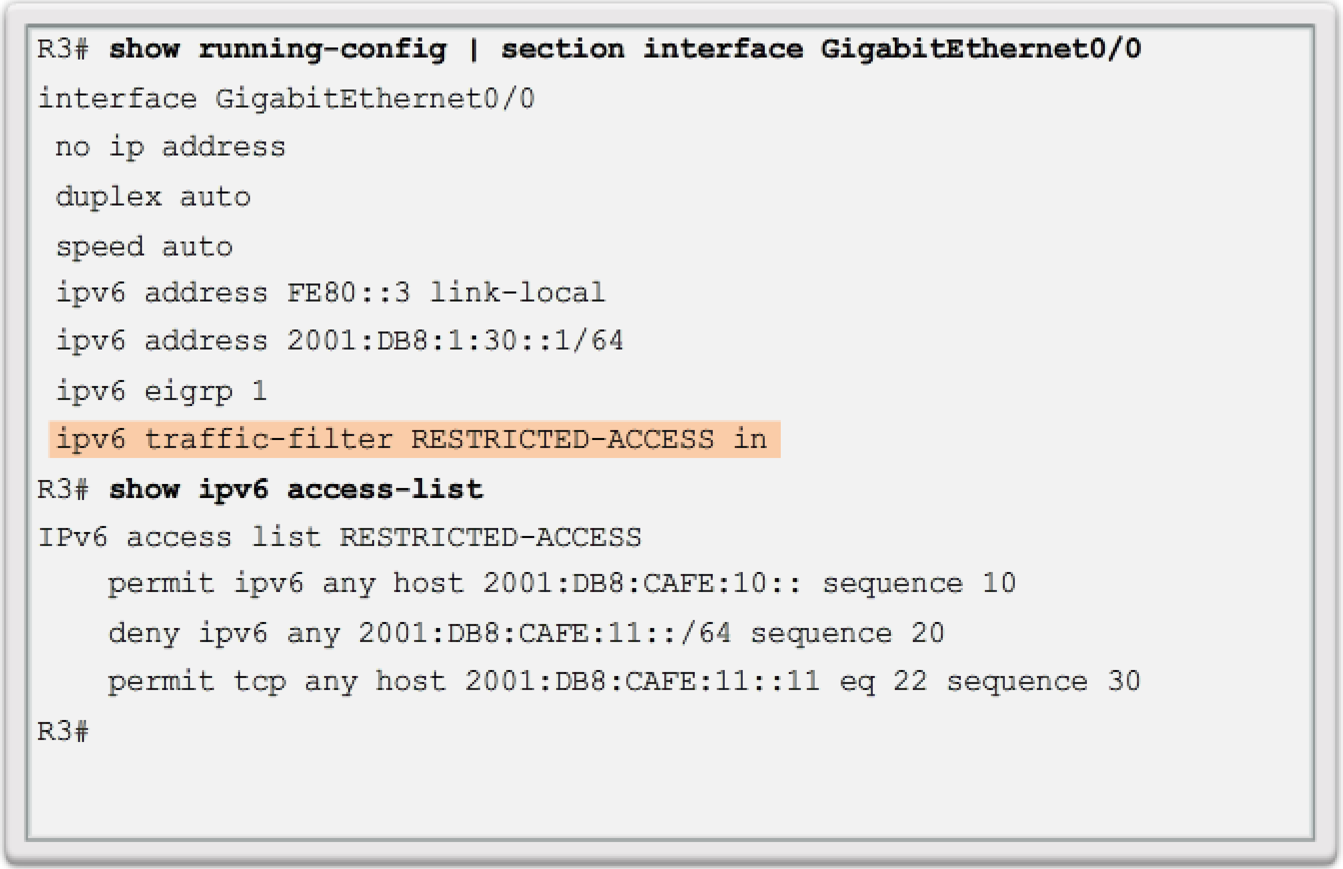
- However, after configuring the ACL, PC3 cannot reach the 10 network or the 11 network, and it cannot SSH into the host at 2001:DB8:CAFE:11::11.



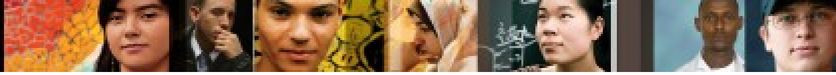
Common ACL Errors

Troubleshooting IPv6 ACLs- Example 2 cont…

Verify the IPv6 ACL Configuration and Application

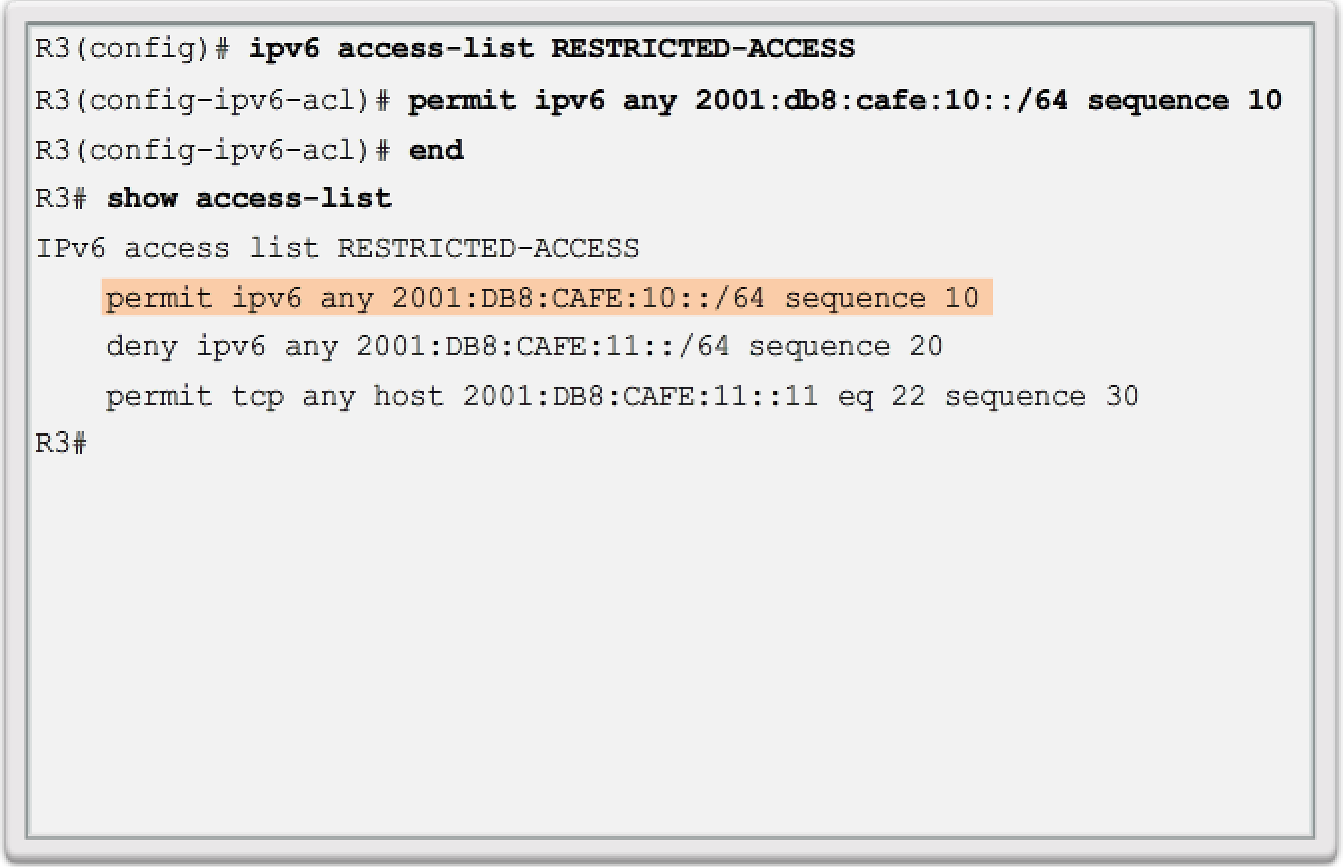


```
R3# show running-config | section interface GigabitEthernet0/0
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
 ipv6 address FE80::3 link-local
 ipv6 address 2001:DB8:1:30::1/64
 ipv6 eigrp 1
 ipv6 traffic-filter RESTRICTED-ACCESS in
R3# show ipv6 access-list
IPv6 access list RESTRICTED-ACCESS
    permit ipv6 any host 2001:DB8:CAFE:10:: sequence 10
    deny ipv6 any 2001:DB8:CAFE:11::/64 sequence 20
    permit tcp any host 2001:DB8:CAFE:11::11 eq 22 sequence 30
R3#
```

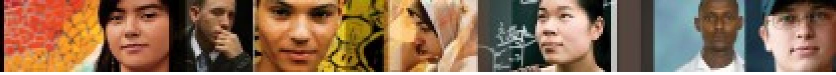


Common ACL Errors

Troubleshooting IPv6 ACLs- Example 2 cont…



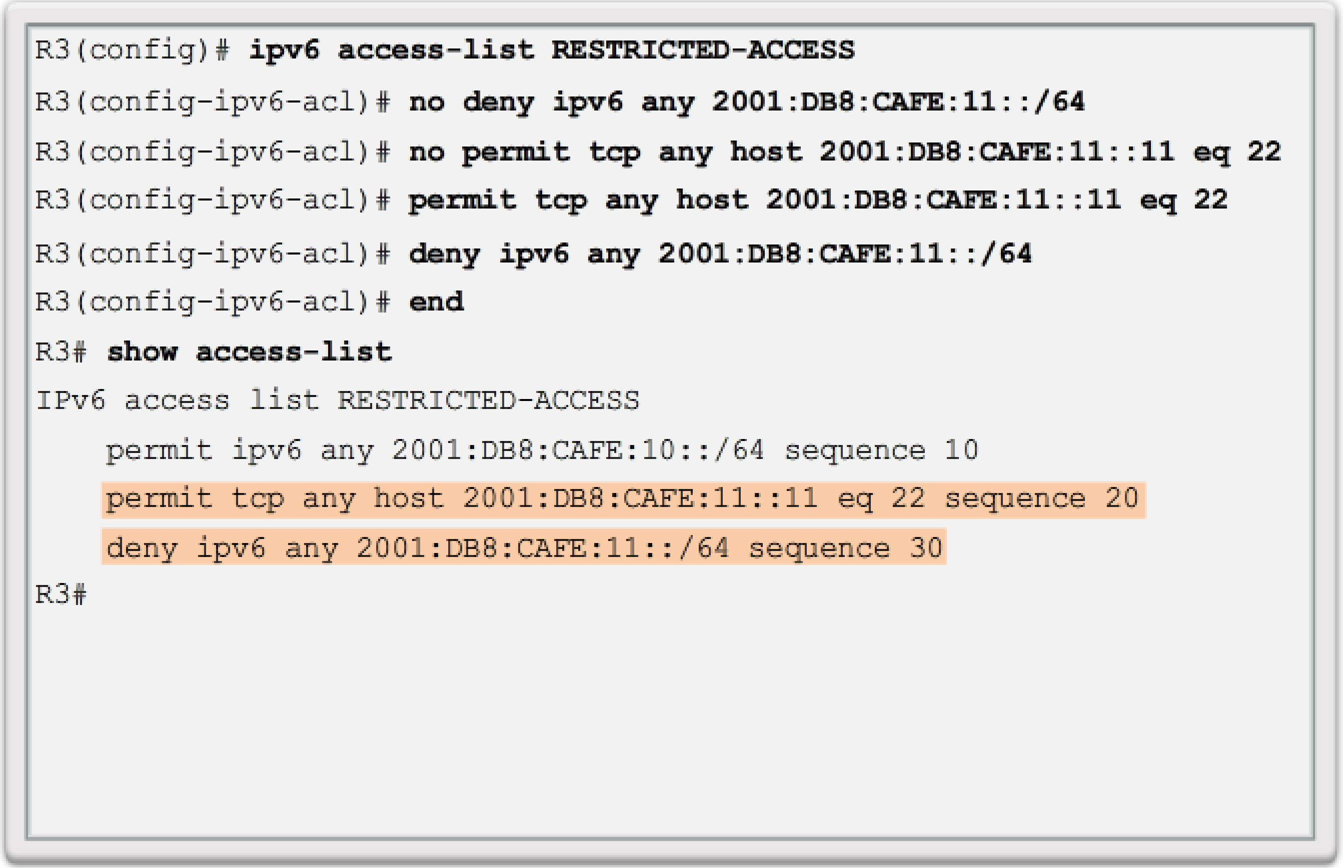
```
R3(config)# ipv6 access-list RESTRICTED-ACCESS
R3(config-ipv6-acl)# permit ipv6 any 2001:db8:cafe:10::/64 sequence 10
R3(config-ipv6-acl)# end
R3# show access-list
IPv6 access list RESTRICTED-ACCESS
    permit ipv6 any 2001:DB8:CAFE:10::/64 sequence 10
    deny ipv6 any 2001:DB8:CAFE:11::/64 sequence 20
    permit tcp any host 2001:DB8:CAFE:11::11 eq 22 sequence 30
R3#
```



Common ACL Errors

Troubleshooting IPv6 ACLs- Example 2 cont…

Replace the IPv6 ACL Host Statement

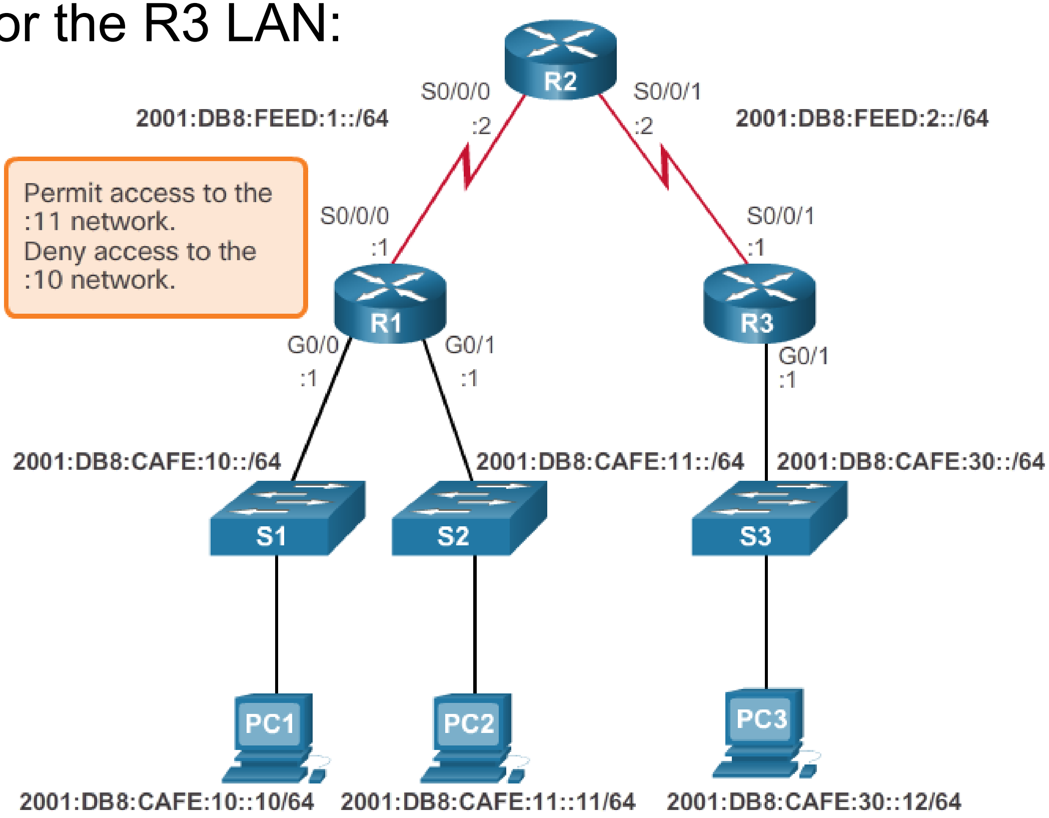


```
R3(config)# ipv6 access-list RESTRICTED-ACCESS
R3(config-ipv6-acl)# no deny ipv6 any 2001:DB8:CAFE:11::/64
R3(config-ipv6-acl)# no permit tcp any host 2001:DB8:CAFE:11::11 eq 22
R3(config-ipv6-acl)# permit tcp any host 2001:DB8:CAFE:11::11 eq 22
R3(config-ipv6-acl)# deny ipv6 any 2001:DB8:CAFE:11::/64
R3(config-ipv6-acl)# end
R3# show access-list
IPv6 access list RESTRICTED-ACCESS
    permit ipv6 any 2001:DB8:CAFE:10::/64 sequence 10
    permit tcp any host 2001:DB8:CAFE:11::11 eq 22 sequence 20
    deny ipv6 any 2001:DB8:CAFE:11::/64 sequence 30
R3#
```

Common ACL Errors

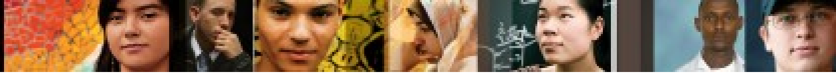
Troubleshooting IPv6 ACLs- Example 3

- R1 is configured with IPv6 ACL DENY-ACCESS that should enforce the following policy for the R3 LAN:



Permit access to the :11 network.
Deny access to the :10 network.

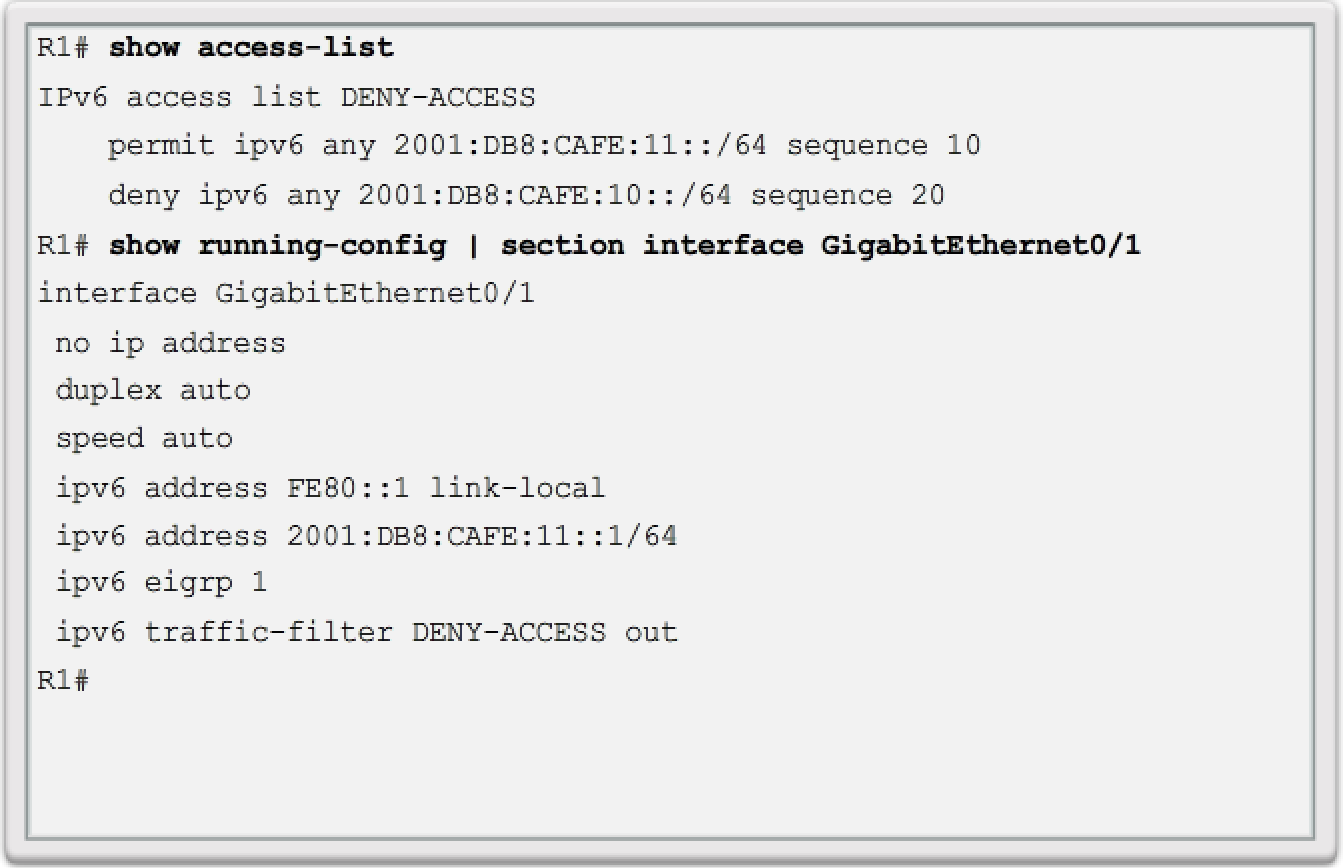
- However, after applying the ACL to the interface the :10 network is still reachable from the :30 network.



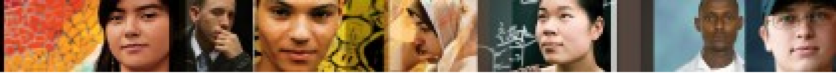
Common ACL Errors

Troubleshooting IPv6 ACLs- Example 3 cont…

Verify the IPv6 ACL Configuration and Application



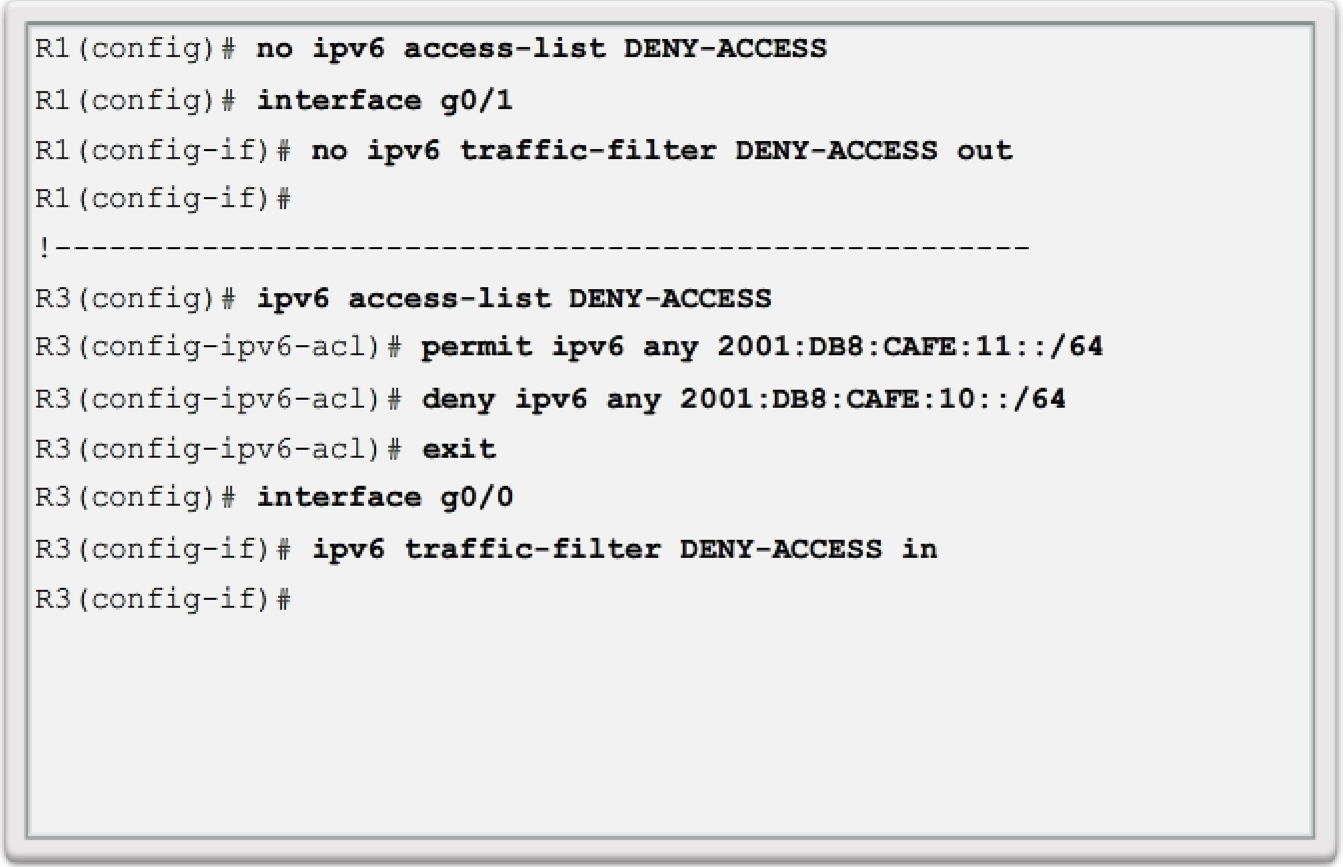
```
R1# show access-list
IPv6 access list DENY-ACCESS
    permit ipv6 any 2001:DB8:CAFE:11::/64 sequence 10
    deny ipv6 any 2001:DB8:CAFE:10::/64 sequence 20
R1# show running-config | section interface GigabitEthernet0/1
interface GigabitEthernet0/1
 no ip address
 duplex auto
 speed auto
 ipv6 address FE80::1 link-local
 ipv6 address 2001:DB8:CAFE:11::1/64
 ipv6 eigrp 1
 ipv6 traffic-filter DENY-ACCESS out
R1#
```



Common ACL Errors

Troubleshooting IPv6 ACLs- Example 3 cont…

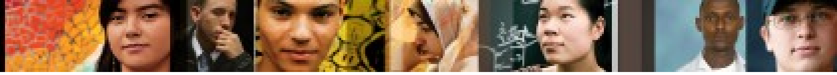
Remove ACL on R1, then Configure and Apply ACL on R2



```
R1(config)# no ipv6 access-list DENY-ACCESS
R1(config)# interface g0/1
R1(config-if)# no ipv6 traffic-filter DENY-ACCESS out
R1(config-if)#
!----------------------------------------------------
R3(config)# ipv6 access-list DENY-ACCESS
R3(config-ipv6-acl)# permit ipv6 any 2001:DB8:CAFE:11::/64
R3(config-ipv6-acl)# deny ipv6 any 2001:DB8:CAFE:10::/64
R3(config-ipv6-acl)# exit
R3(config)# interface g0/0
R3(config-if)# ipv6 traffic-filter DENY-ACCESS in
R3(config-if)#
```

4.5 Chapter Summary

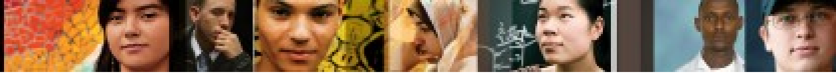
Cisco | Networking Academy®
Mind Wide Open™



Chapter Summary

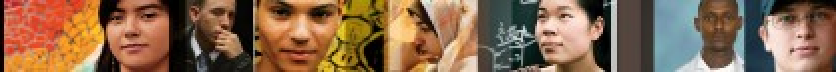
Summary

- By default a router does not filter traffic. Traffic that enters the router is routed solely based on information within the routing table.
- An ACL is a sequential list of permit or deny statements. The last statement of an ACL is always an implicit deny any statement which blocks all traffic. To prevent the implied deny any statement at the end of the ACL from blocking all traffic, the **permit ip any any** statement can be added.
- When network traffic passes through an interface configured with an ACL, the router compares the information within the packet against each entry, in sequential order, to determine if the packet matches one of the statements. If a match is found, the packet is processed accordingly.
- ACLs can be applied to inbound traffic or to outbound traffic.
- Standard ACLs can be used to permit or deny traffic only from a source IPv4 addresses. The basic rule for placing a standard ACL is to place it close to the destination.
- Extended ACLs filter packets based on several attributes: protocol type, source or destination IPv4 address, and source or destination ports. The basic rule for placing an extended ACL is to place it as close to the source as possible.



Summary Continued

- The **access-list** global configuration command defines a standard ACL with a number in the range of 1 through 99 or an extended ACL with numbers in the range of 100 to 199. The **ip access-list standard** *name* is used to create a standard named ACL, whereas the command **ip access-list extended** *name* is for an extended access list.
- After an ACL is configured, it is linked to an interface using the **ip access-group** command in interface configuration mode. A device an only have one ACL per protocol, per direction, per interface.
- To remove an ACL from an interface, first enter the **no ip access-group** command on the interface, and then enter the global **no access-list** command to remove the entire ACL.
- The **show running-config** and **show access-lists** commands are used to verify ACL configuration. The **show ip interface** command is used to verify the ACL on the interface and the direction in which it was applied.
- The **access-class** command configured in line configuration mode is used to link an ACL to a particular VTY line.
- Unlike IPv4, IPv6 ACLs e is no need for a standard or extended option.
- From global configuration mode, use the **ipv6 access-list** *name* command to create an IPv6 ACL. Unlike IPv4 ACLs, IPv6 ACLs do not use wildcard masks. Instead, the prefix-length is used to indicate how much of an IPv6 source or destination address should be matched.
- After an IPv6 ACL is configured, it is linked to an interface using the **ipv6 traffic-filter** command.



Summary Continued

- Unlike IPv4, IPv6 ACLs do not have support for a standard or extended option.
- From global configuration mode, use the **ipv6 access-list** *name* command to create an IPv6 ACL.
- Unlike IPv4 ACLs, IPv6 ACLs do not use wildcard masks. Instead, the prefix-length is used to indicate how much of an IPv6 source or destination address should be matched.
- After an IPv6 ACL is configured, it is linked to an interface using the **ipv6 traffic-filter** command.

Cisco | Networking Academy®

Mind Wide Open™