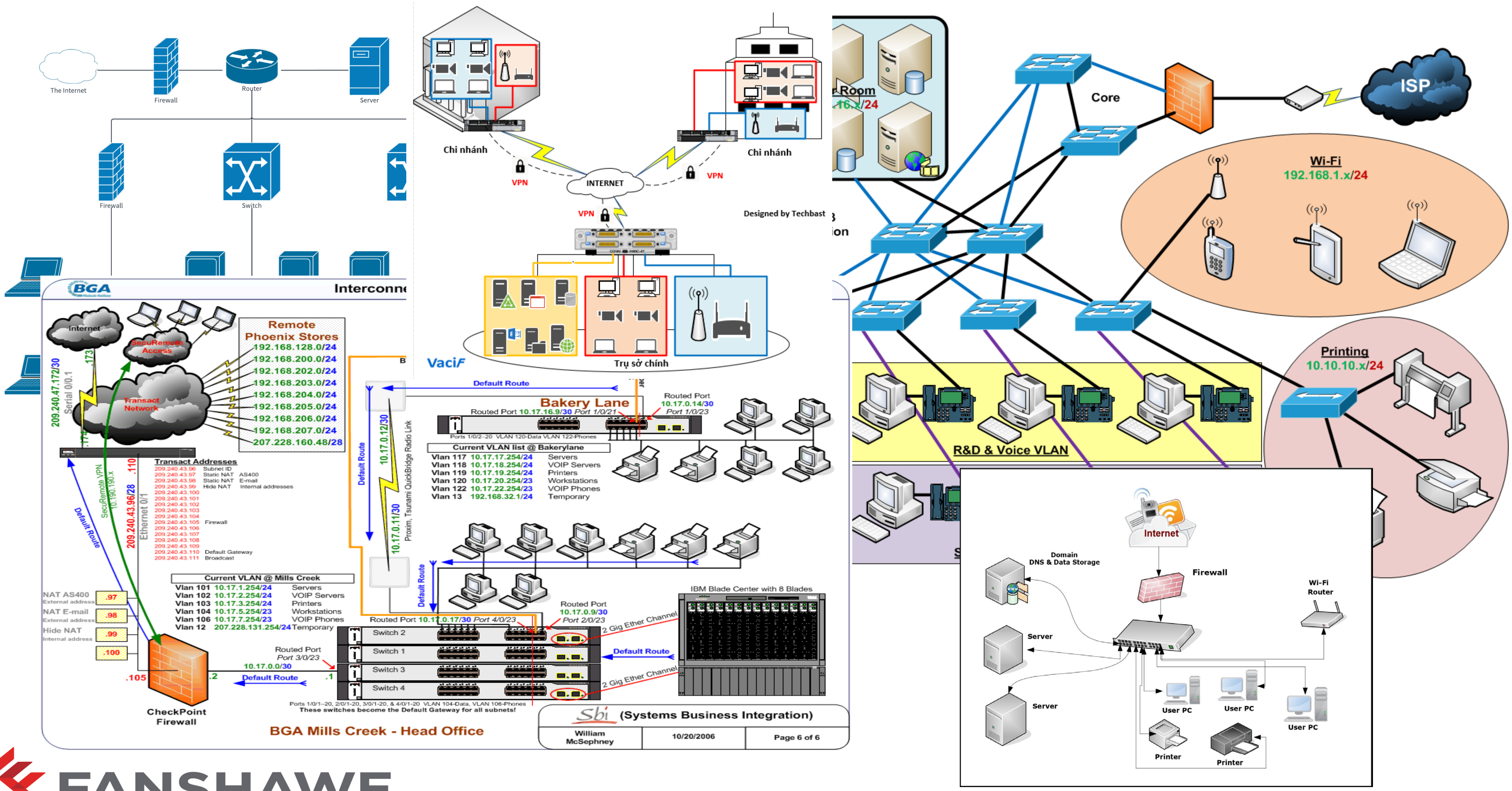
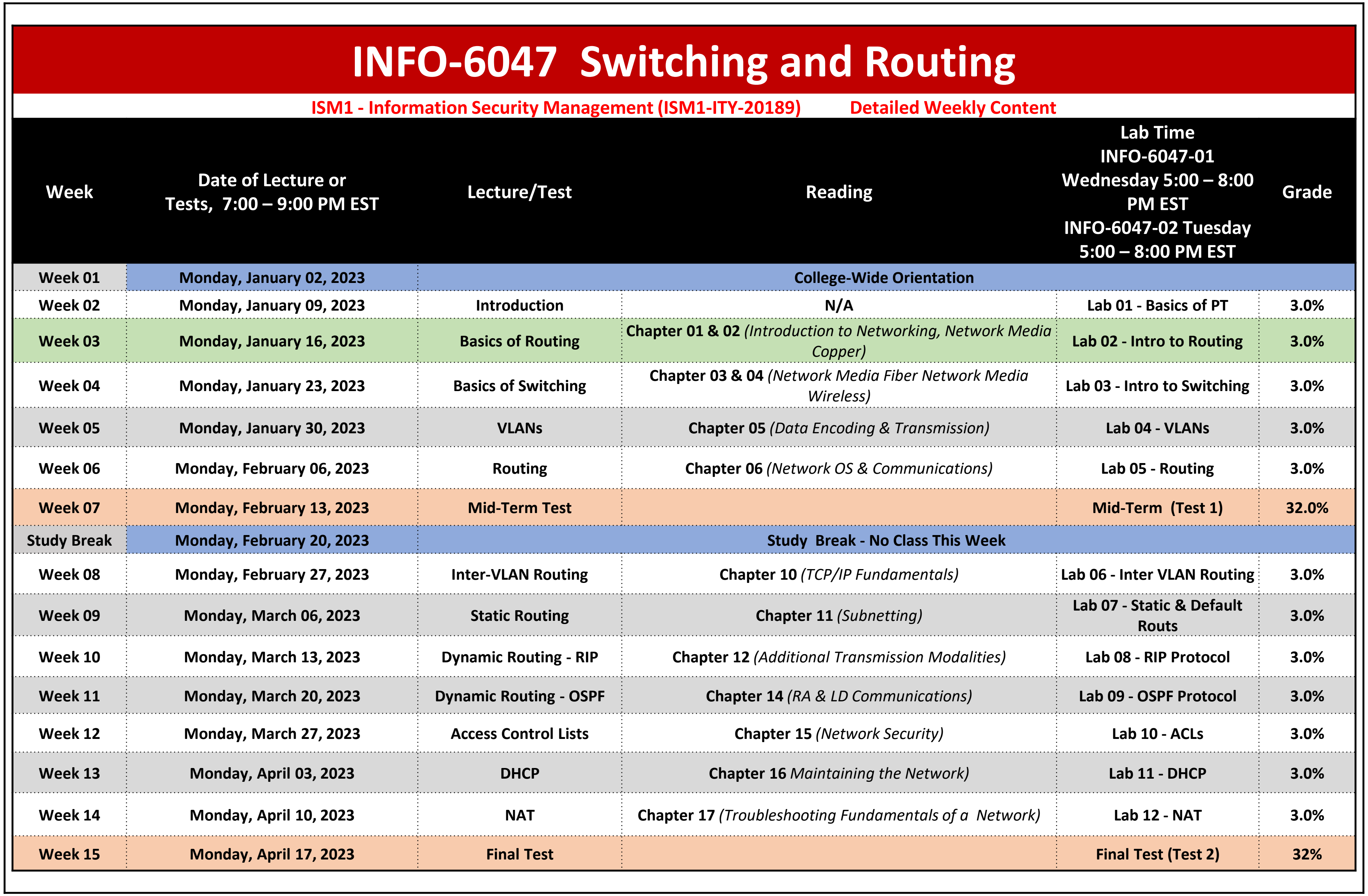
# ACLs


The Internet
Firewall
Router
Server
Firewall
Switch
Chi nhánh
Chi nhánh
VPN
INTERNET
VPN
VPN
Designed by Techbast
Trụ sở chính
VaciF
Core
ISP
Room
.16.x/24
Wi-Fi
192.168.1.x/24
Printing
10.10.10.x/24
R&D & Voice VLAN
BGA
Interconne
Internet
SecuRemote Access
Transact Network
Remote Phoenix Stores
192.168.128.0/24
192.168.200.0/24
192.168.202.0/24
192.168.203.0/24
192.168.204.0/24
192.168.205.0/24
192.168.206.0/24
192.168.207.0/24
207.228.160.48/28
209.240.47.172/30
Serial 0/0.1
.173
.174
Transact Addresses
SecuRemote VPN 10.190.190.x
209.240.43.96/28
.110
Ethernet 0/1
Default Route
NAT AS400
External address
NAT E-mail
External address
Hide NAT
Internal address
.97
.98
.99
.100
.105
.2
CheckPoint Firewall
Current VLAN @ Mills Creek
Vlan 101 10.17.1.254/24 Servers
Vlan 102 10.17.2.254/24 VOIP Servers
Vlan 103 10.17.3.254/24 Printers
Vlan 104 10.17.5.254/23 Workstations
Vlan 106 10.17.7.254/23 VOIP Phones
Vlan 12 207.228.131.254/24 Temporary
Routed Port Port 3/0/23
10.17.0.0/30
Default Route
.1
Switch 2
Switch 1
Switch 3
Switch 4
Routed Port 10.17.0.17/30 Port 4/0/23
Routed Port 10.17.0.9/30 Port 2/0/23
2 Gig Ether Channel
2 Gig Ether Channel
Default Route
IBM Blade Center with 8 Blades
These switches become the Default Gateway for all subnets!
BGA Mills Creek - Head Office
Default Route
Default Route
10.17.0.12/30
10.17.0.11/30
Proxim, Tsunami QuickBridge Radio Link
Default Route
Bakery Lane
Routed Port 10.17.16.9/30 Port 1/0/21
Routed Port 10.17.0.14/30 Port 1/0/23
Ports 1/0/2–20 VLAN 120-Data VLAN 122-Phones
Current VLAN list @ Bakerylane
Vlan 117 10.17.17.254/24 Servers
Vlan 118 10.17.18.254/24 VOIP Servers
Vlan 119 10.17.19.254/24 Printers
Vlan 120 10.17.20.254/23 Workstations
Vlan 122 10.17.22.254/23 VOIP Phones
Vlan 13 192.168.32.1/24 Temporary
Sbi (Systems Business Integration)
William McSephney
10/20/2006
Page 6 of 6
Internet
Domain DNS & Data Storage
Firewall
Wi-Fi Router
Server
Server
User PC
User PC
User PC
Printer
Printer



# INFO-6047  Switching and Routing

ISM1 - Information Security Management (ISM1-ITY-20189)        Detailed Weekly Content

| Week | Date of Lecture or Tests,  7:00 – 9:00 PM EST | Lecture/Test | Reading | Lab Time<br>INFO-6047-01<br>Wednesday 5:00 – 8:00 PM EST<br>INFO-6047-02 Tuesday 5:00 – 8:00 PM EST | Grade |
|---|---|---|---|---|---|
| Week 01 | Monday, January 02, 2023 | College-Wide Orientation | | | |
| Week 02 | Monday, January 09, 2023 | Introduction | N/A | Lab 01 - Basics of PT | 3.0% |
| Week 03 | Monday, January 16, 2023 | Basics of Routing | Chapter 01 & 02 *(Introduction to Networking, Network Media Copper)* | Lab 02 - Intro to Routing | 3.0% |
| Week 04 | Monday, January 23, 2023 | Basics of Switching | Chapter 03 & 04 *(Network Media Fiber Network Media Wireless)* | Lab 03 - Intro to Switching | 3.0% |
| Week 05 | Monday, January 30, 2023 | VLANs | Chapter 05 *(Data Encoding & Transmission)* | Lab 04 - VLANs | 3.0% |
| Week 06 | Monday, February 06, 2023 | Routing | Chapter 06 *(Network OS & Communications)* | Lab 05 - Routing | 3.0% |
| Week 07 | Monday, February 13, 2023 | Mid-Term Test | | Mid-Term  (Test 1) | 32.0% |
| Study Break | Monday, February 20, 2023 | Study  Break - No Class This Week | | | |
| Week 08 | Monday, February 27, 2023 | Inter-VLAN Routing | Chapter 10 *(TCP/IP Fundamentals)* | Lab 06 - Inter VLAN Routing | 3.0% |
| Week 09 | Monday, March 06, 2023 | Static Routing | Chapter 11 *(Subnetting)* | Lab 07 - Static & Default Routs | 3.0% |
| Week 10 | Monday, March 13, 2023 | Dynamic Routing - RIP | Chapter 12 *(Additional Transmission Modalities)* | Lab 08 - RIP Protocol | 3.0% |
| Week 11 | Monday, March 20, 2023 | Dynamic Routing - OSPF | Chapter 14 *(RA & LD Communications)* | Lab 09 - OSPF Protocol | 3.0% |
| Week 12 | Monday, March 27, 2023 | Access Control Lists | Chapter 15 *(Network Security)* | Lab 10 - ACLs | 3.0% |
| Week 13 | Monday, April 03, 2023 | DHCP | Chapter 16 *Maintaining the Network)* | Lab 11 - DHCP | 3.0% |
| Week 14 | Monday, April 10, 2023 | NAT | Chapter 17 *(Troubleshooting Fundamentals of a  Network)* | Lab 12 - NAT | 3.0% |
| Week 15 | Monday, April 17, 2023 | Final Test | | Final Test (Test 2) | 32% |

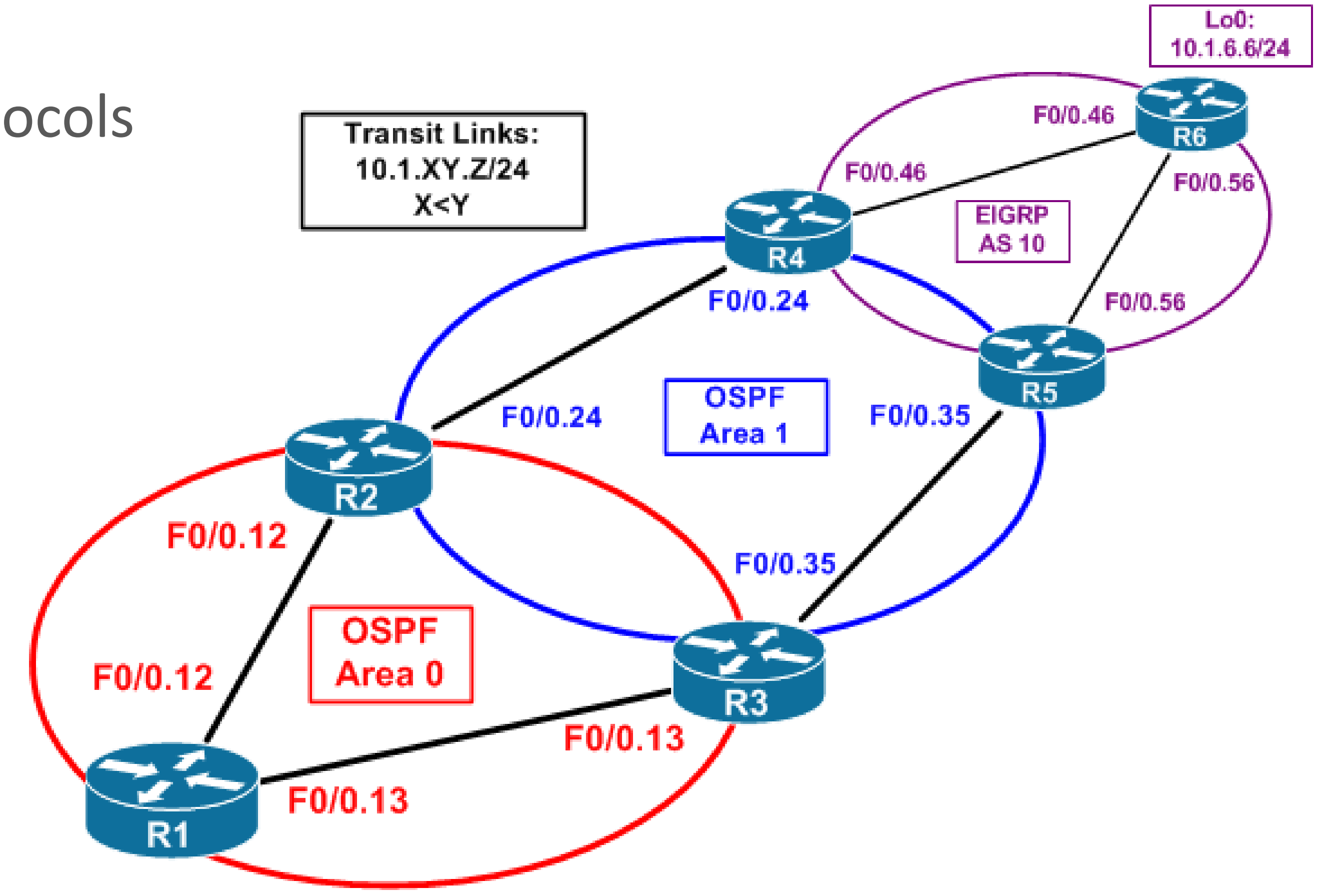
# Final Exam

- Section 1 and 2
- When: Monday April 17th at 7:00 PM
- Where: B1071
- How: Open book (120 minutes for 120 points)
  - Allowed resources: Lecture slides, labs, and textbook
  - Not allowed: Phones, Google, and instant messaging.
- Note: Bring your laptop charger as you will be asked to keep your screen brightness up and to avoid having your laptop battery dying.

- Online and part-time
- When: Tuesday April 18th 12:00 AM to 11:59 PM (24 hours to begin exam)
- Where: Online (120 minutes for 120 points)
- How: Open book
  - Allowed resources: Lecture slides, labs, and textbook
  - Not allowed: Phones, Google, and instant messaging.

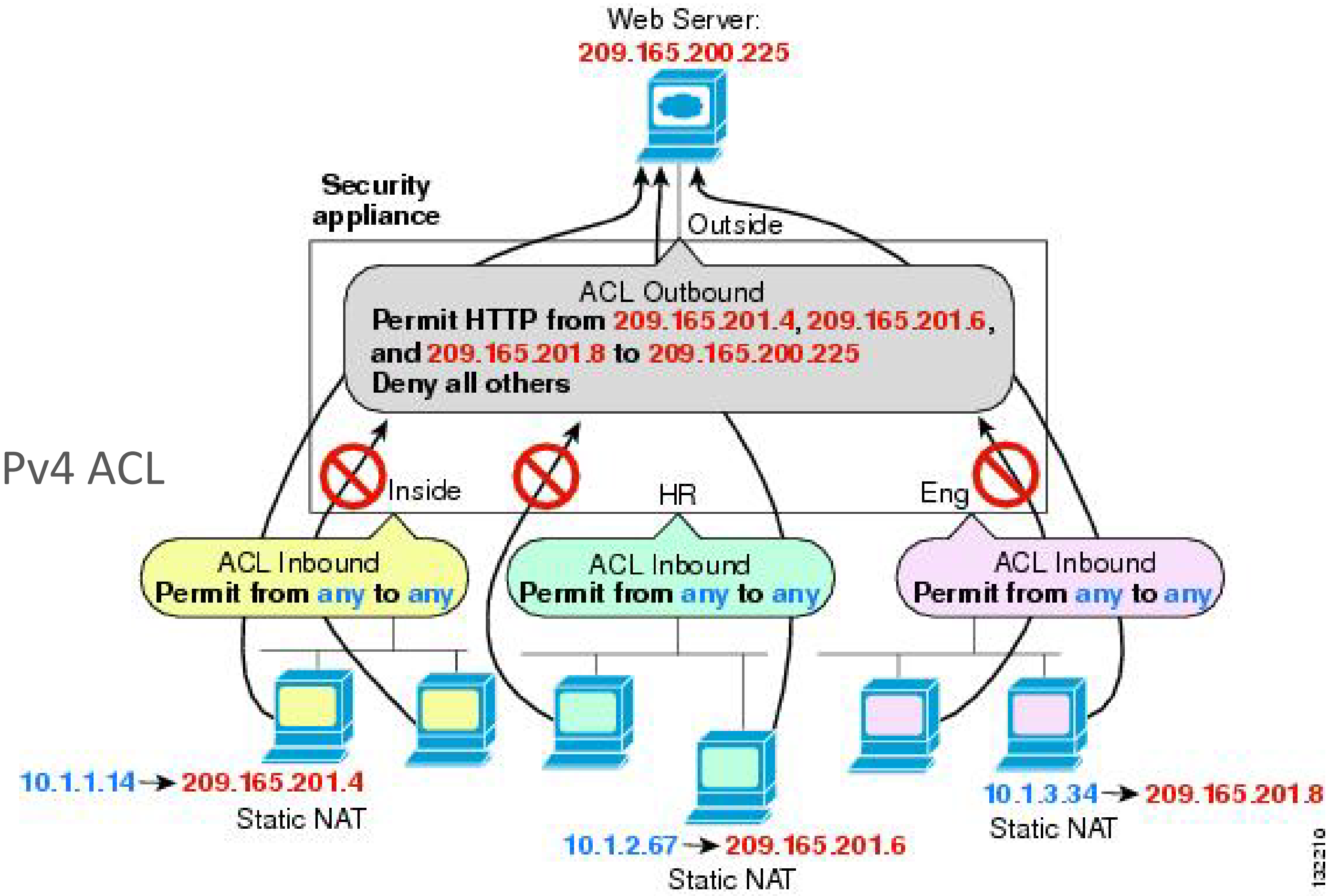
# Review - Lecture 09 – Dynamic Routing - OSPF

- Link-State Routing Protocol Operation
- Link-State Updates
- Why Use Link-State Routing Protocols
- Open Shortest Path First
- OSPF Messages
- OSPF Operation
- OSPF Router ID
- Configure Single-area OSPFv2
- OSPF Cost
- Verify OSPF
- OSPFv2 vs. OSPFv3
- Configuring OSFPv3



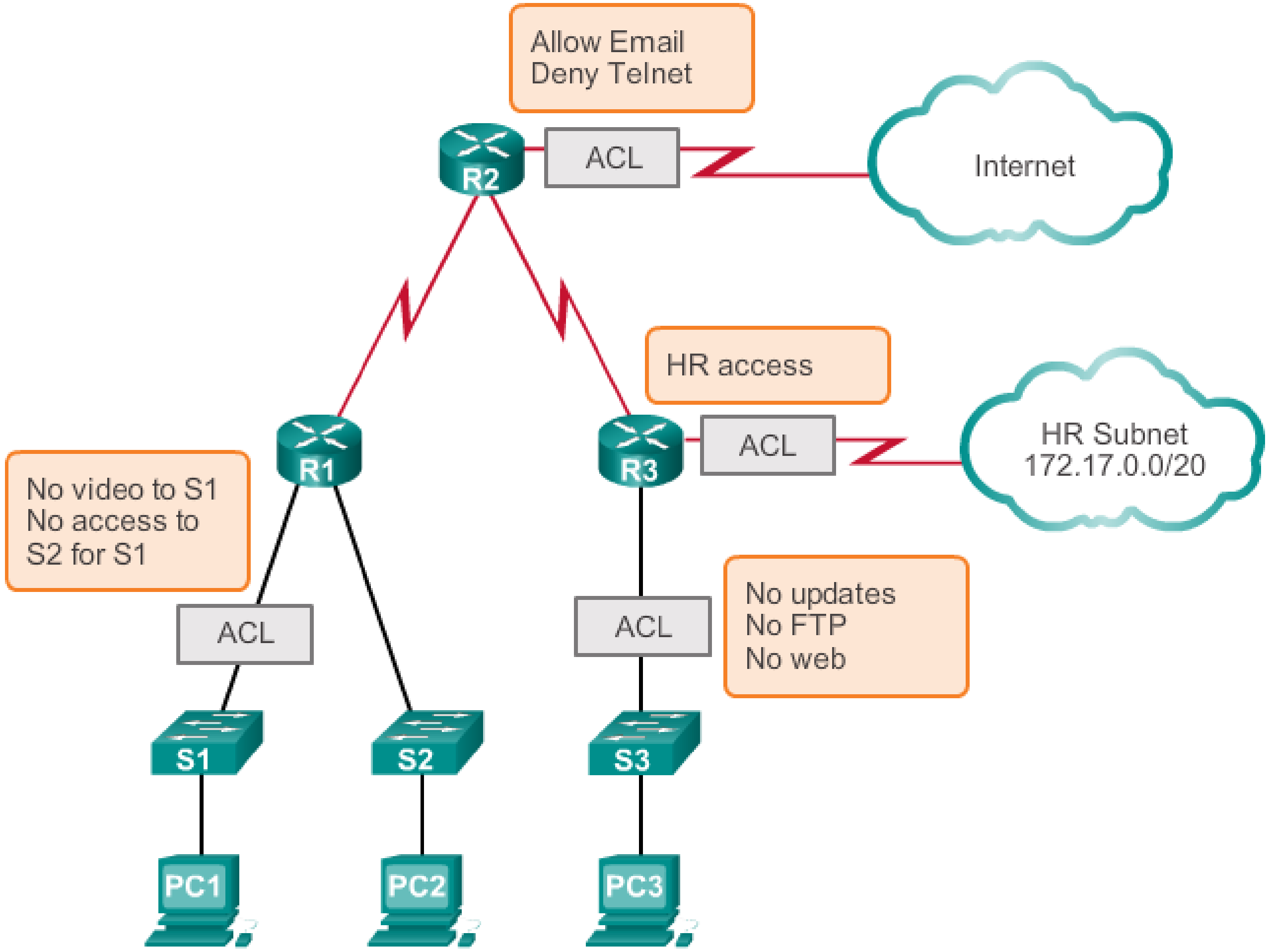
# Summary - ACLs

- Purpose of ACLs
- Standard versus Extended IPv4 ACLs
- Wildcard Masks in ACLs
- Guidelines for ACL creation
- Guidelines for ACL Placement
- Configure Standard IPv4 ACLs
- Modify IPv4 ACLs
- Securing VTY ports with a Standard IPv4 ACL
- Configure Extended IPv4 ACLs
- Processing Packets with ACLs
- Common ACLs Errors
- IPv6 ACL Creation
- Configuring IPv6 ACLs
- Summary
- Lab



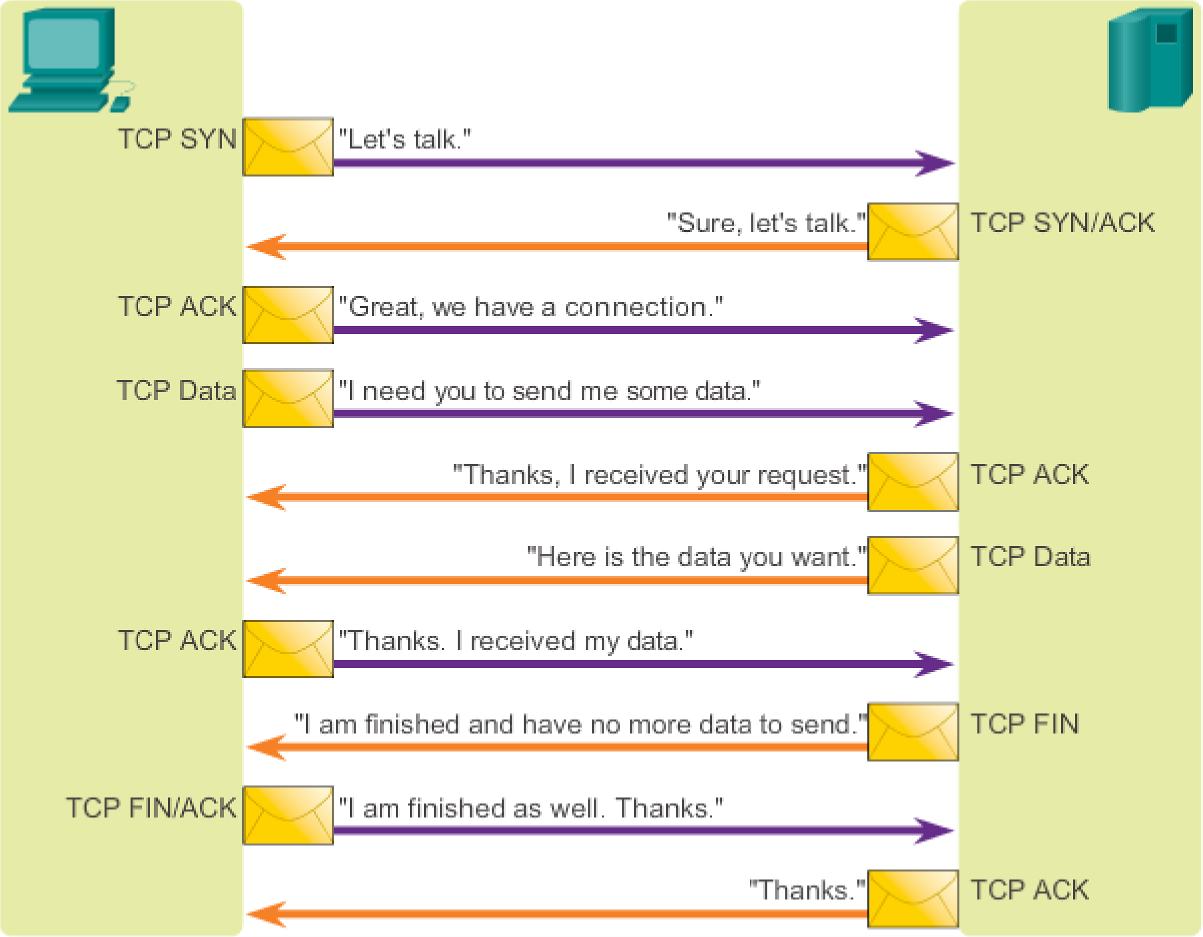
# Purpose of ACLs

- What is an ACL?
  - **Filters traffic** based on traffic type
  - Permit or deny hosts access to network services
  - Limits network traffic to increase network performance
  - Provide basic level of security for network access



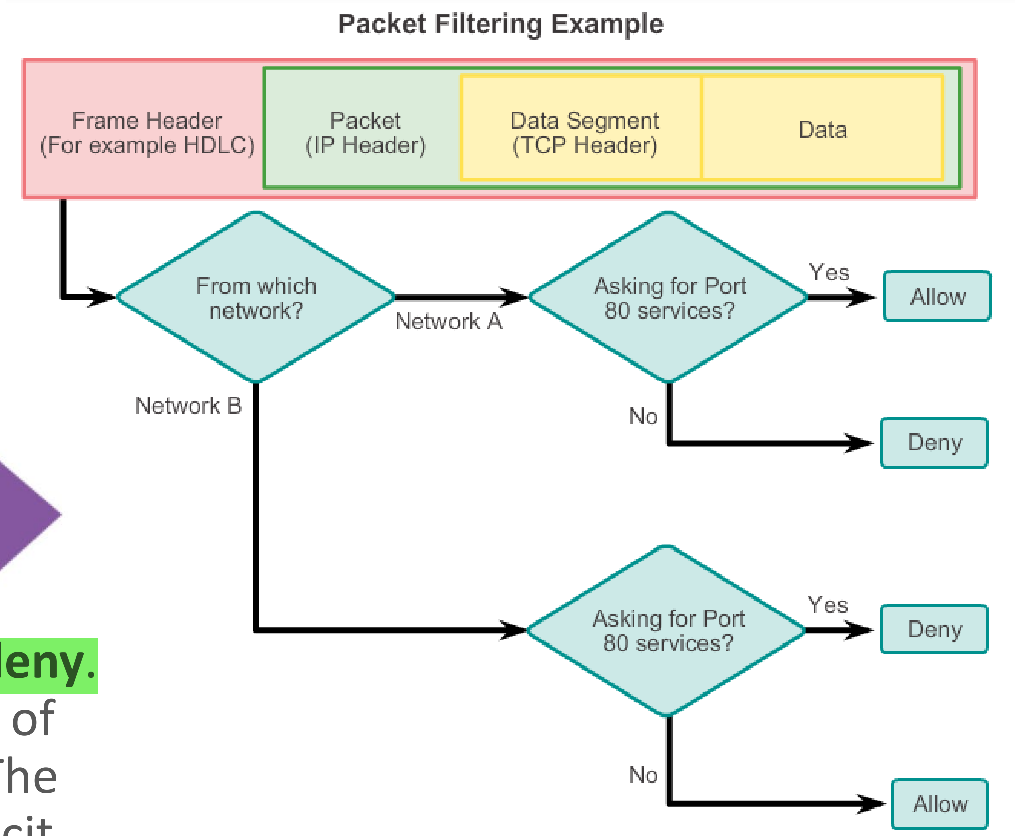
# Purpose of ACLs (continued)

- A Typical TCP Conversation
- Packet Filtering (ACLs)
  - Packet filtering, sometimes called static packet filtering, controls access to a network by analyzing the incoming and outgoing packets and passing or dropping them based on given criteria, such as the source IP address, destination IP addresses, and the protocol carried within the packet.
  - A router acts as a packet filter when it forwards or denies packets according to filtering rules.
  - An ACL is a sequential list of permit or deny statements, known as access control entries (ACEs).


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

# Purpose of ACLs (continued)

- Packet Filtering (continued)

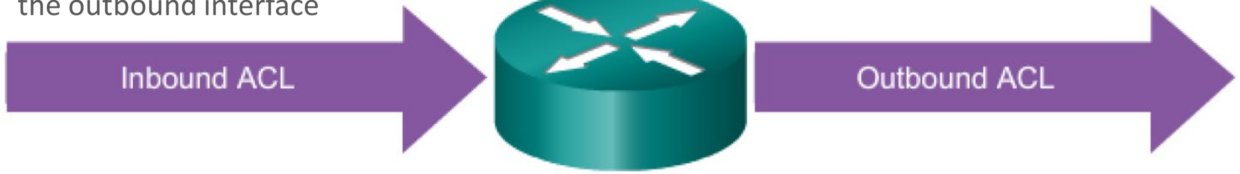


Packet Filtering Example

- ACL Operation

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

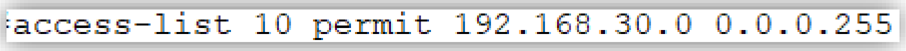
An outbound ACL filters packets after being routed, regardless of the inbound interface



- The last statement of an ACL is always an **implicit deny**. This statement is automatically inserted at the end of each ACL even though it is not physically present. The implicit deny blocks all traffic. Because of this implicit deny, an **ACL that does not have at least one permit statement will block all traffic.**

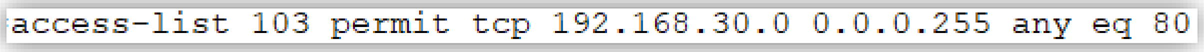
# Standard versus Extended IPv4 ACLs

- Types of Cisco IPv4 ACLs
  - Standard ACLs
    - **access-list 1-99 {permit | deny} address mask**



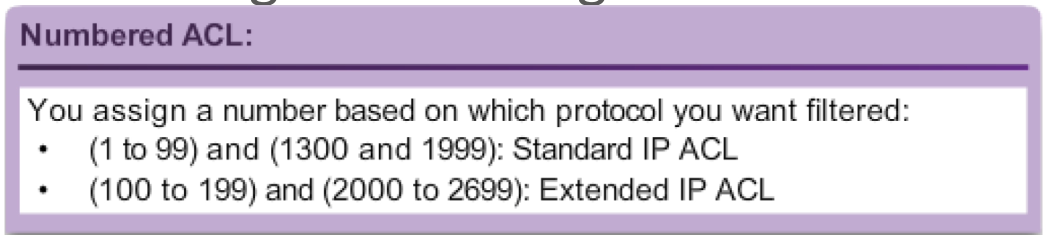
```
access-list 10 permit 192.168.30.0 0.0.0.255
```

    - Standard ACLs filter packets based on the source address only
  - Extended ACLs
    - **access-list 100-199 {permit | deny} {ip | tcp | udp | icmp} source source-mask [lt | gt | eq | neq] [source-port] destination dest-mask [lt|gt|eq|neq] [dest-port] [log]**



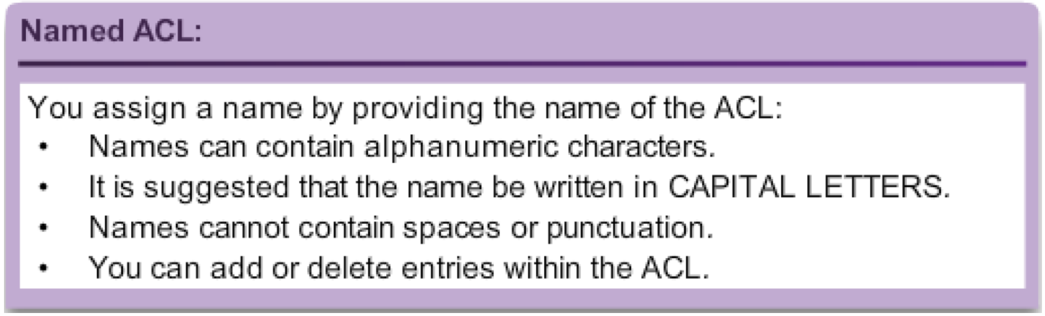
```
access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80
```

    - Extended ACLs filter packets based on serval attributes, including the following:
      - Source and destination IPaddresses
      - Source and destination TCP and UDP ports
      - Protocol type / Protocol number (example: IP, ICP, UDP, TCP, etc…)
- Numbering and Naming ACLs



**Numbered ACL:**

You assign a number based on which protocol you want filtered:
- (1 to 99) and (1300 and 1999): Standard IP ACL
- (100 to 199) and (2000 to 2699): Extended IP ACL



**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.
- You can add or delete entries within the ACL.

# Wildcard Masks in ACLs

- Introducing ACL Wildcard Masking
  - Wildcard masks and subnet masks differ in the way they match binary 1s and 0s. Wildcard masks use the following rules to match binary 1s and 0s:
    - Wildcard mask bit 0 -Match the corresponding bit value in the address.
    - Wildcard mask bit 1 -Ignore the corresponding bit value in the address.
  - Wildcard masks are often referred to as an inverse mask. The reason is that, unlike a subnet mask in which binary 1 is equal to a match and binary 0 is not a match, in a wildcard mask the reverse is true.
- Calculating the Wildcard Mask
  - Calculating the Wildcard Mask from a known Mask or CIDR

$$\begin{array}{rl} & 255\ .\ 255\ .\ 255\ .\ 255 \\ - & \underline{255\ .\ 255\ .\ 255\ .\ \ \ 0\ \ } \ /\ 24 \\ & 0\ \ .\ \ 0\ \ .\ \ 0\ \ .\ 255 \end{array}$$

$$\begin{array}{rl} & 255\ .\ 255\ .\ 255\ .\ 255 \\ - & \underline{255\ .\ 255\ .\ 255\ .\ 252} \ /\ 30 \\ & 0\ \ .\ \ 0\ \ .\ \ 0\ \ .\ \ 3 \end{array}$$

$$\begin{array}{rl} & 255\ .\ 255\ .\ 255\ .\ 255 \\ - & \underline{255\ .\ 255\ .\ 255\ .\ 192} \ /\ 26 \\ & 0\ \ .\ \ 0\ \ .\ \ 0\ \ .\ 63 \end{array}$$

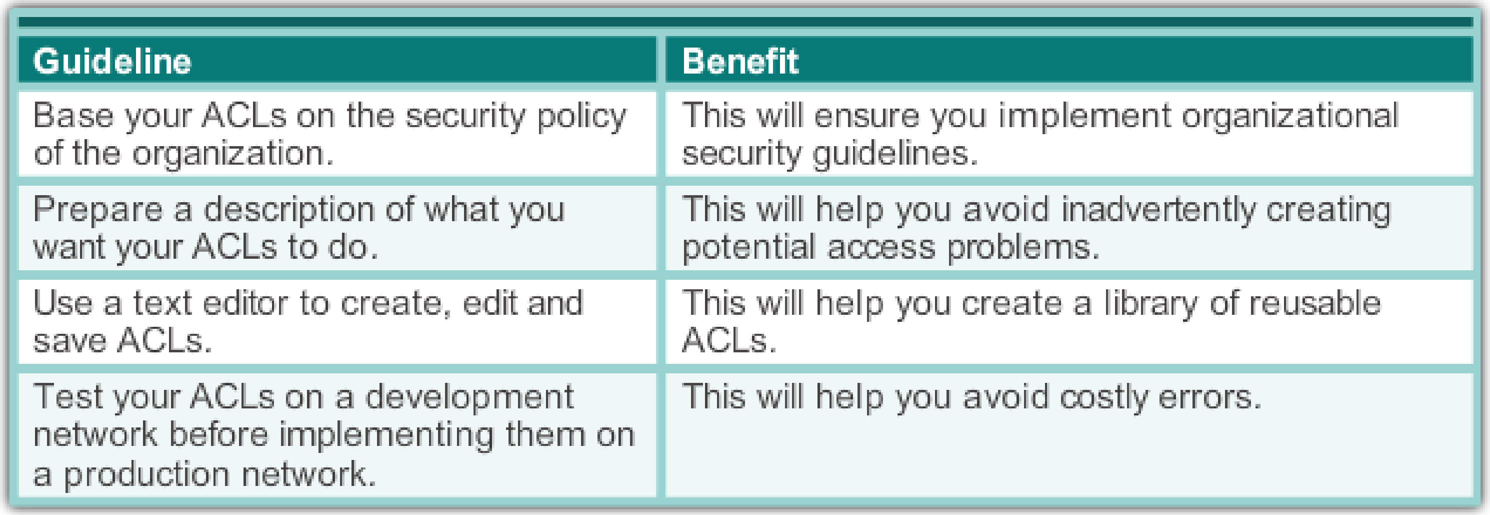
# Guidelines for ACL creation

- General Guidelines for Creating ACLs
  - Use ACLs in firewall routers positioned between your internal network and an external network such as the Internet.
  - Use ACLs on a router positioned between two parts of your network to control traffic entering or exiting a specific part of your internal network.
  - Configure ACLs on border routers, that is routers situated at the edges of your networks.
  - Configure ACLs for each network protocol configured on the border router interfaces.
  - The Three **P**s
    - One ACL **per protocol** -To control traffic flow on an interface, an ACL must be defined for each protocol enabled on the interface. (eg, IPv4 or IPv6)
    - 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 and outbound traffic. (ie, IN or OUT)
    - One ACL **per interface** -ACLs control traffic for an interface. (eg, FastEthernet0/0)

FANSHAWE

# Guidelines for ACL creation (continued)

- ACL Best Practices



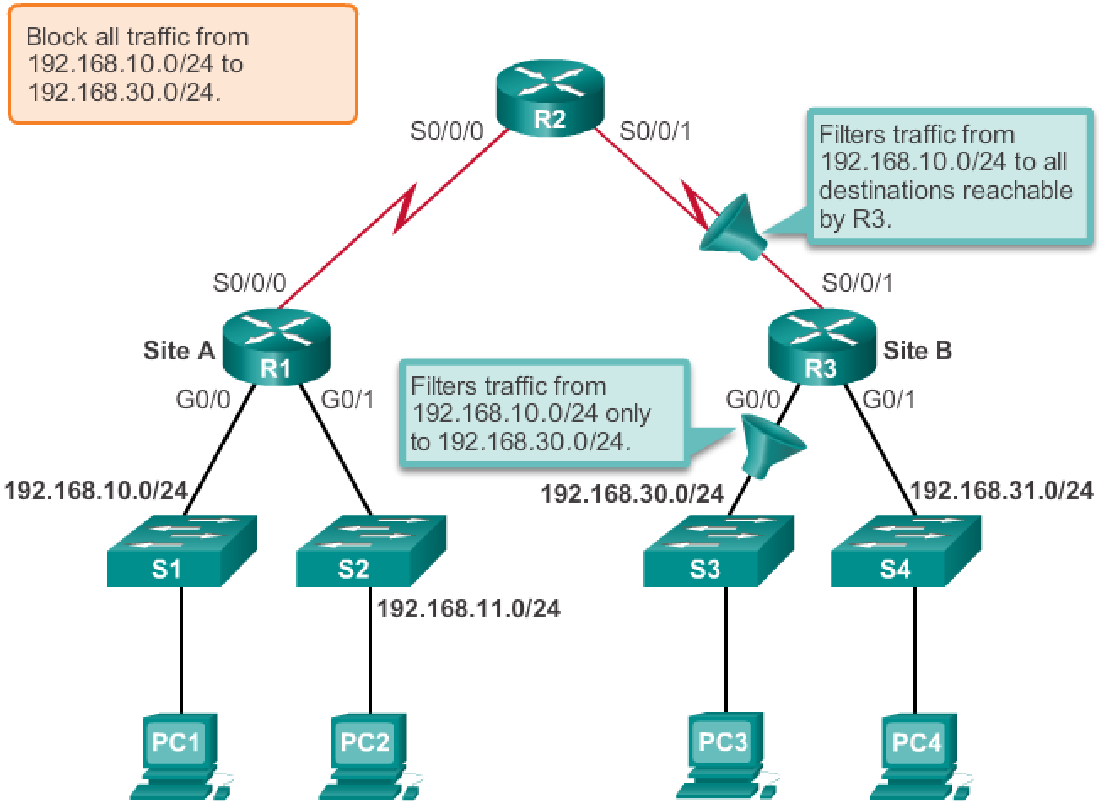
| Guideline | Benefit |
|---|---|
| Base your ACLs on the security policy of the organization. | This will ensure you implement organizational security guidelines. |
| Prepare a description of what you want your ACLs to do. | This will help you avoid inadvertently creating potential access problems. |
| Use a text editor to create, edit and save ACLs. | This will help you create a library of reusable ACLs. |
| Test your ACLs on a development network before implementing them on a production network. | This will help you avoid costly errors. |

# Guidelines for ACL Placement

- Where to Place ACLs
  - Every ACL should be placed where it has the greatest impact on efficiency. The basic rules are:
    - Standard ACLs - Because standard ACLs **do not specify destination addresses**, place them **as close to the destination as possible**.
    - Extended ACLs - Locate extended ACLs **as close as possible to the source of the traffic** to be filtered.
  - 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.

# Guidelines for ACL Placement (continued)

- Standard ACL Placement

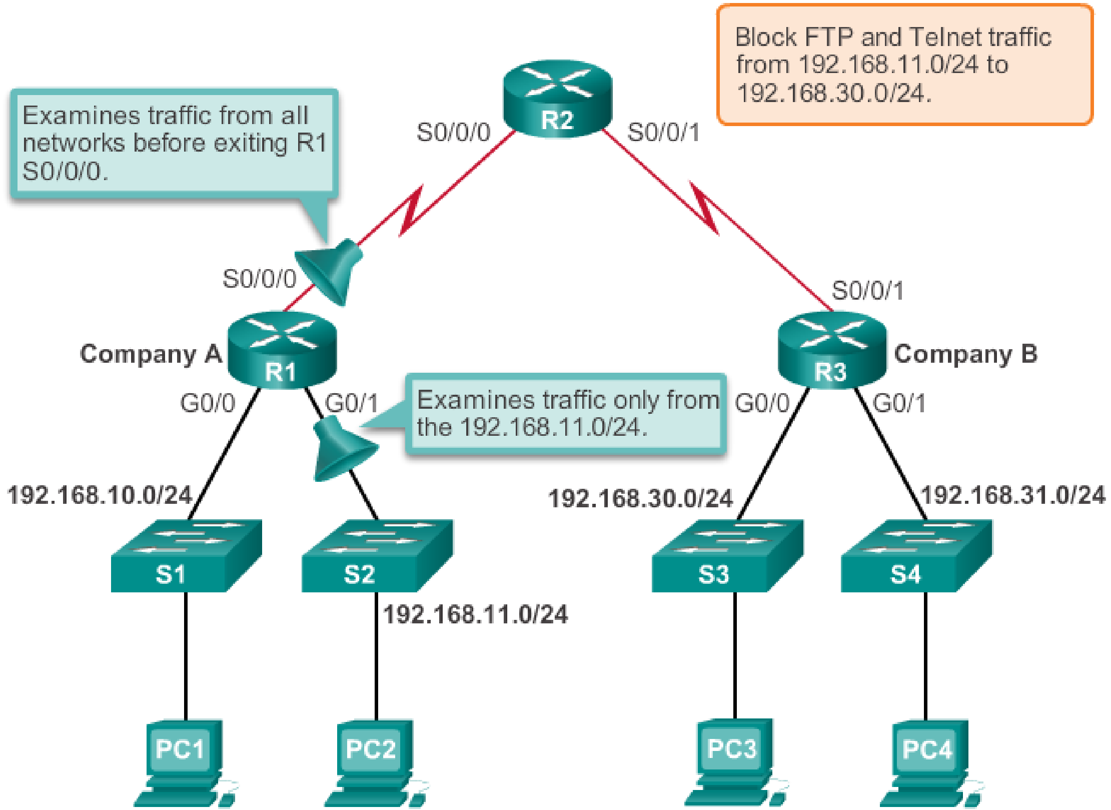


Block all traffic from 192.168.10.0/24 to 192.168.30.0/24.

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.

## Extended ACL Placement



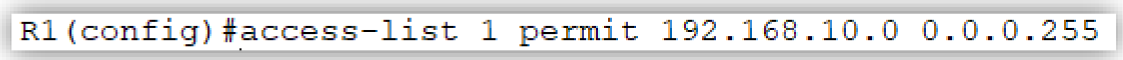
Block FTP and Telnet traffic from 192.168.11.0/24 to 192.168.30.0/24.

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

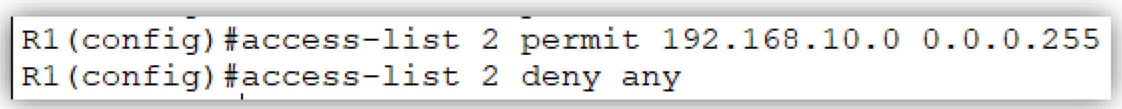
Examines traffic only from the 192.168.11.0/24.

# Configure Standard IPv4 ACLs

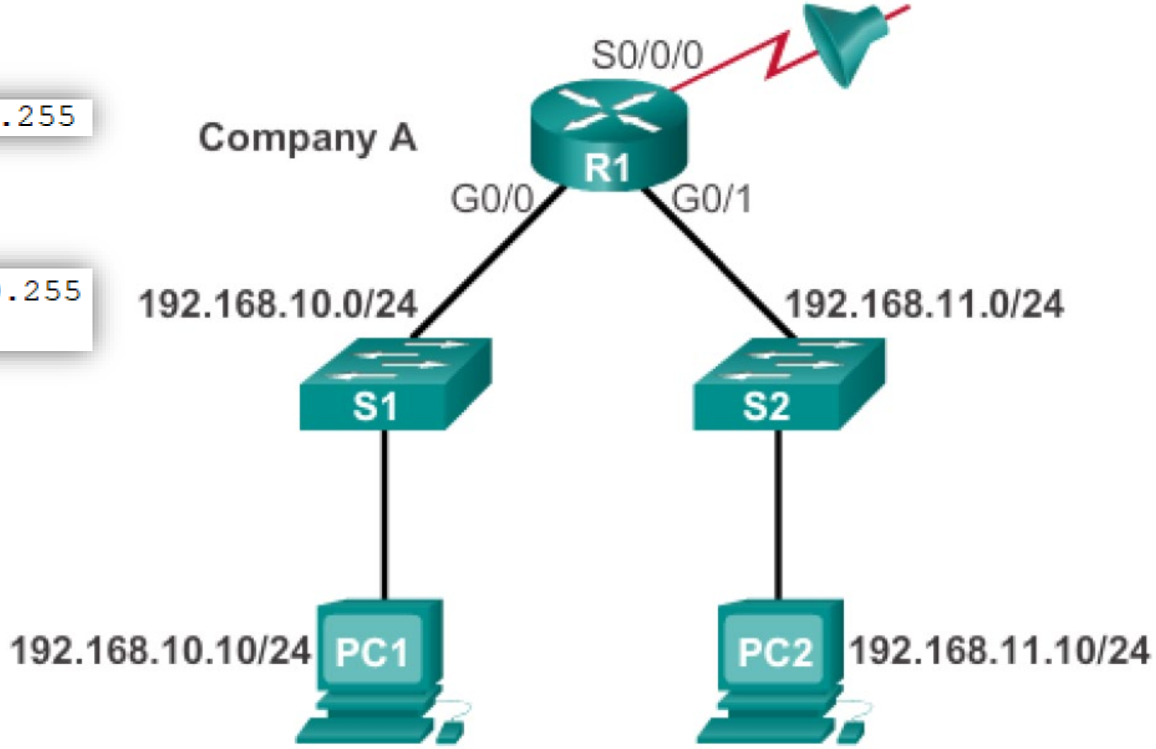
- Entering Criteria Statements



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

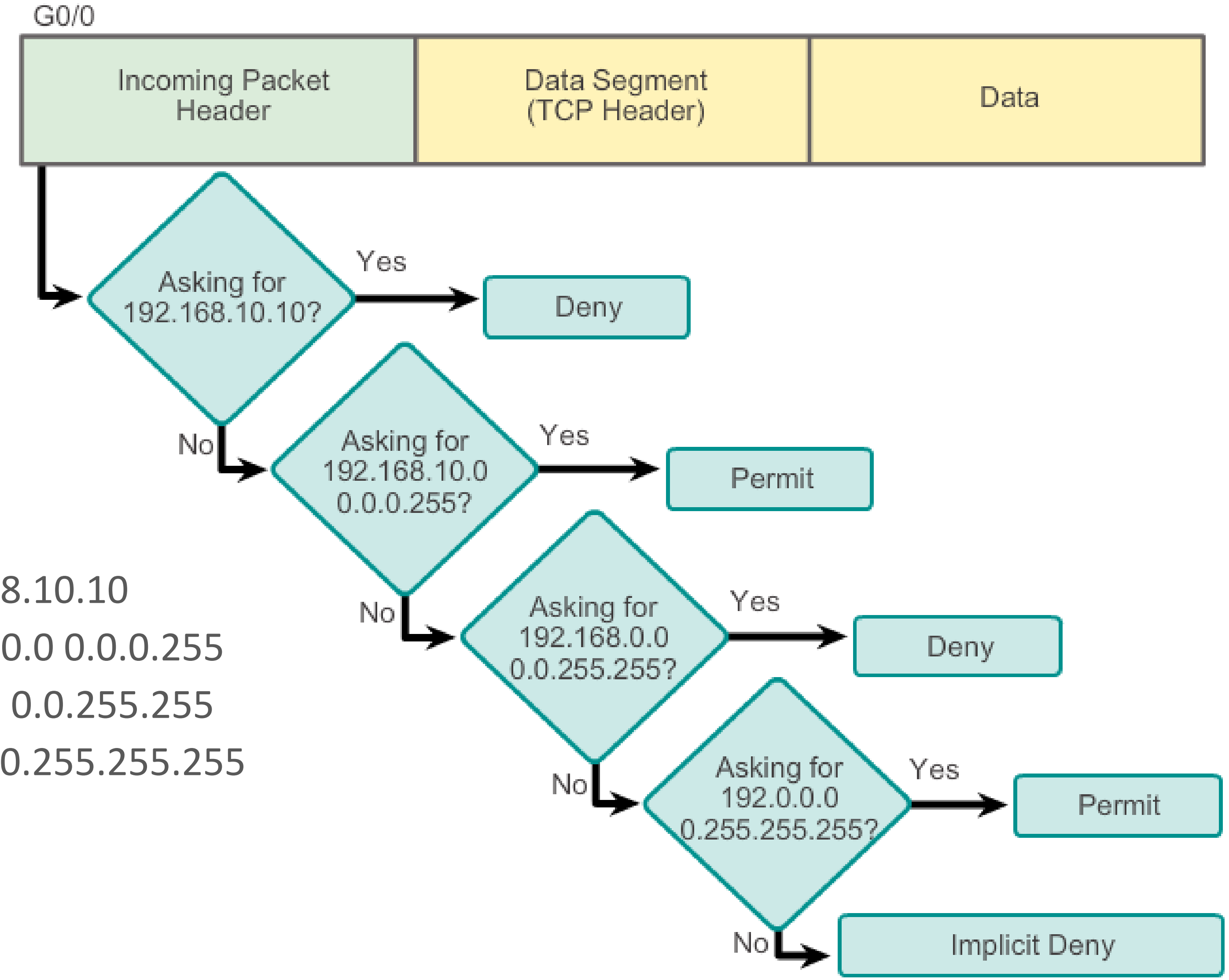


```
R1(config)#access-list 2 permit 192.168.10.0 0.0.0.255
R1(config)#access-list 2 deny any
```



# Configure Standard IPv4 ACLs (continued)

- Configuring a Standard ACL

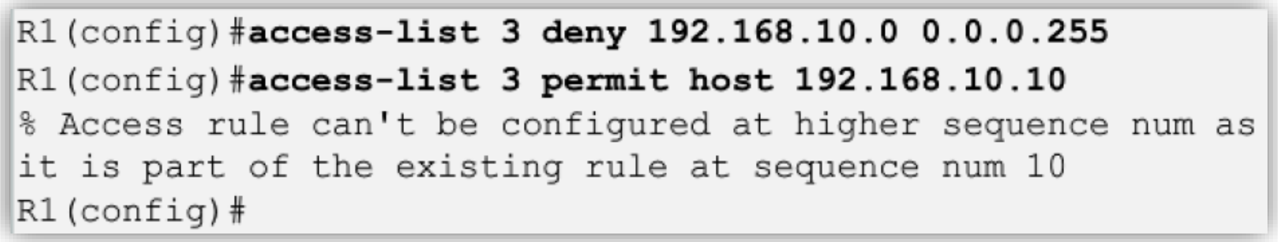

G0/0
Incoming Packet Header
Data Segment (TCP Header)
Data
Asking for 192.168.10.10?
Yes
Deny
No
Asking for 192.168.10.0 0.0.0.255?
Yes
Permit
No
Asking for 192.168.0.0 0.0.255.255?
Yes
Deny
No
Asking for 192.0.0.0 0.255.255.255?
Yes
Permit
No
Implicit Deny

- Example ACL
  - access-list 2 deny host 192.168.10.10
  - access-list 2 permit 192.168.10.0 0.0.0.255
  - access-list 2 deny 192.168.0.0 0.0.255.255
  - access-list 2 permit 192.0.0.0 0.255.255.255

# Configure Standard IPv4 ACLs (continued)

- Configuring a Standard ACL (continued)
  - The full syntax of the standard ACL command is as follows:
  - Router(config)# **access-list** access-list-number {**deny** | **permit** | **remark**} source-ipaddress wildcard [log]
  - To remove the ACL, the global configuration no access-list (#) command is used.

    (no access-list will remove all access lists, no access-list #/name will remove that specific access-list)
  - The remark keyword is used for documentation and makes access lists a great deal easier to understand.
- Internal Logic
  - Cisco IOS applies an internal logic when accepting and processing standard access list statements. As discussed previously, access list statements are processed sequentially. Therefore, the order in which statements are entered is important.

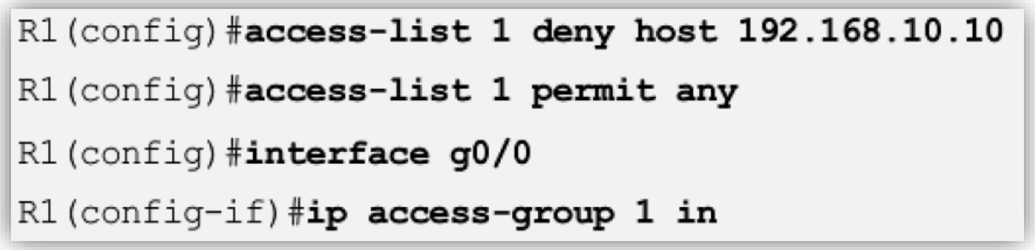


```
R1(config)#access-list 3 deny 192.168.10.0 0.0.0.255
R1(config)#access-list 3 permit host 192.168.10.10
% Access rule can't be configured at higher sequence num as
it is part of the existing rule at sequence num 10
R1(config)#
```

ACL 3: Host statement conflicts with the previous range statement.

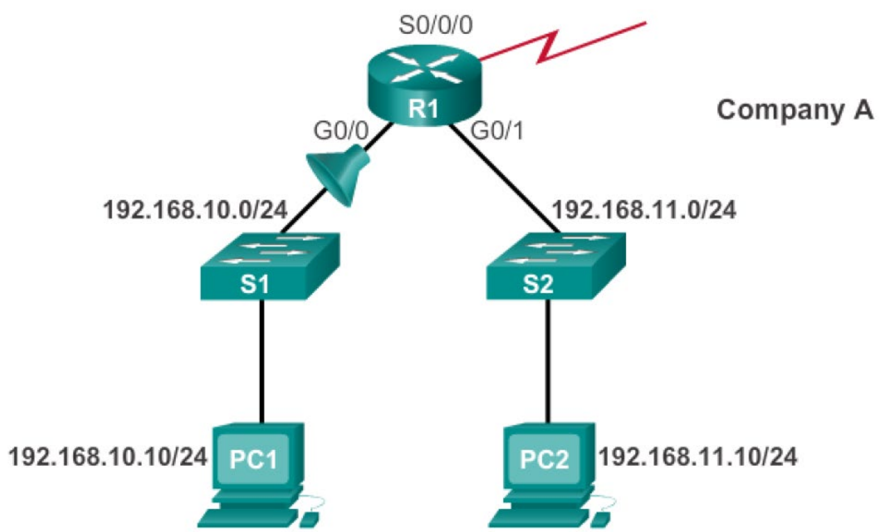
# Configure Standard IPv4 ACLs (continued)

- Applying Standard ACLs to Interfaces
  - After a standard ACL is configured, it is linked to an interface using the ip access-group command in interface configuration mode:
  - `Router(config-if)# ip access-group {access-list-number | access-list-name} {in | out}`
  - 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.



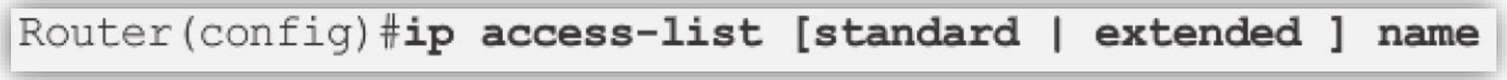
```
R1(config)#access-list 1 deny host 192.168.10.10
R1(config)#access-list 1 permit any
R1(config)#interface g0/0
R1(config-if)#ip access-group 1 in
```

Deny a Specific Host



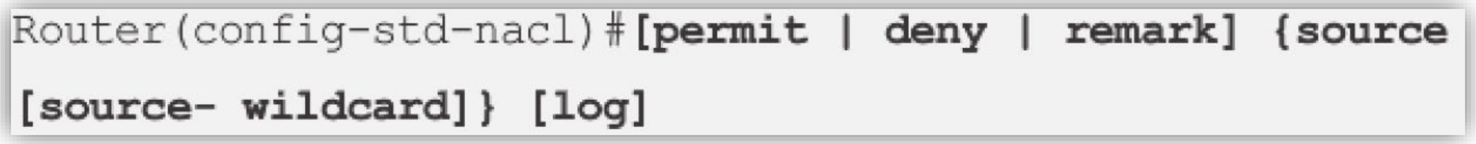
# Configure Standard IPv4 ACLs (continued)

- Creating Named Standard ACLs

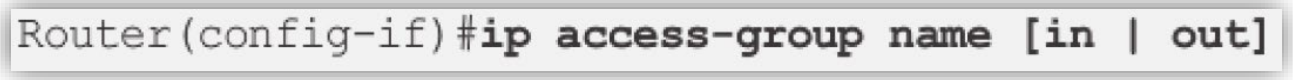


```
Router(config)#ip access-list [standard | extended ] name
```

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



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

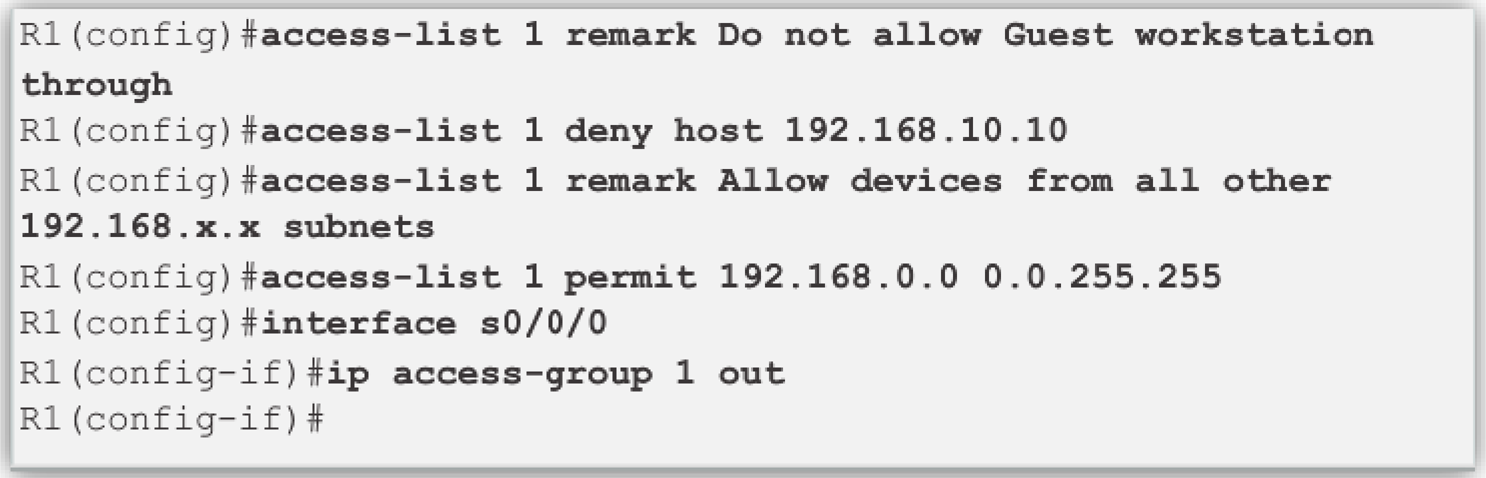


```
Router(config-if)#ip access-group name [in | out]
```

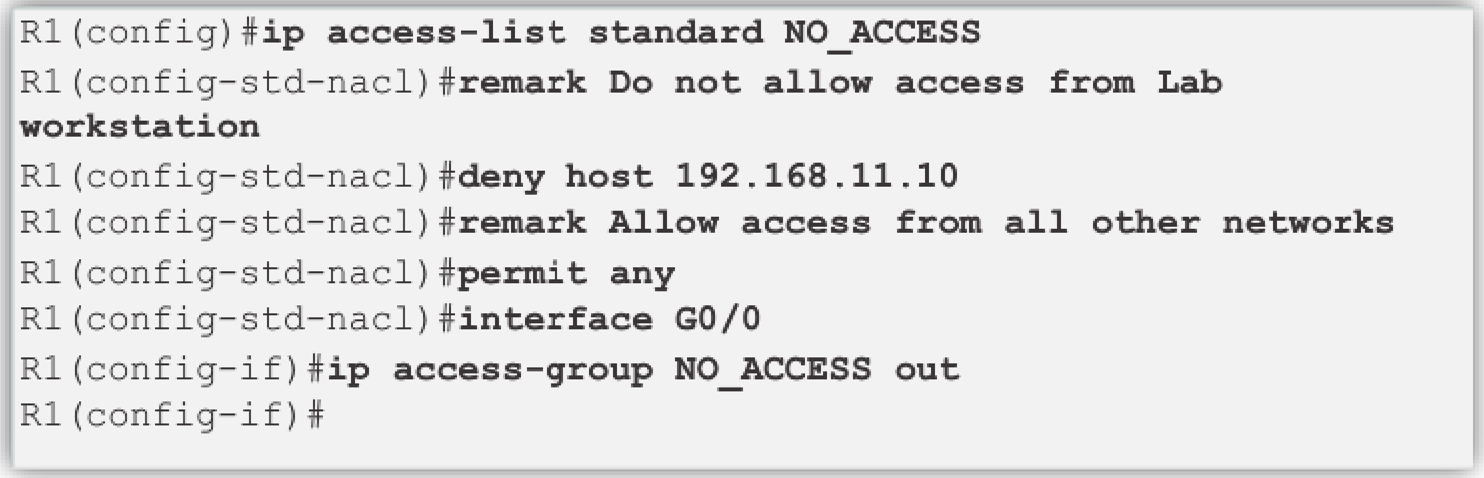
Activates the named IP ACL on an interface

# Configure Standard IPv4 ACLs (continued)

- Commenting ACLs



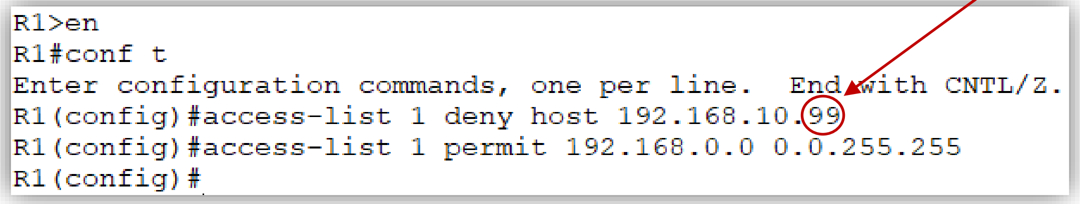
```
R1(config)#access-list 1 remark Do not allow Guest workstation
through
R1(config)#access-list 1 deny host 192.168.10.10
R1(config)#access-list 1 remark Allow devices from all other
192.168.x.x subnets
R1(config)#access-list 1 permit 192.168.0.0 0.0.255.255
R1(config)#interface s0/0/0
R1(config-if)#ip access-group 1 out
R1(config-if)#
```



```
R1(config)#ip access-list standard NO_ACCESS
R1(config-std-nacl)#remark Do not allow access from Lab
workstation
R1(config-std-nacl)#deny host 192.168.11.10
R1(config-std-nacl)#remark Allow access from all other networks
R1(config-std-nacl)#permit any
R1(config-std-nacl)#interface G0/0
R1(config-if)#ip access-group NO_ACCESS out
R1(config-if)#
```

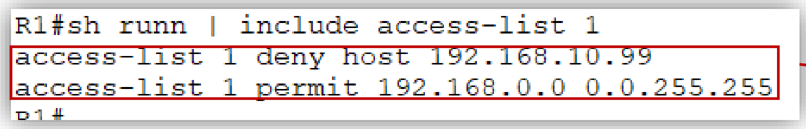
# Modify IPv4 ACLs

- Editing Standard Numbered ACLs



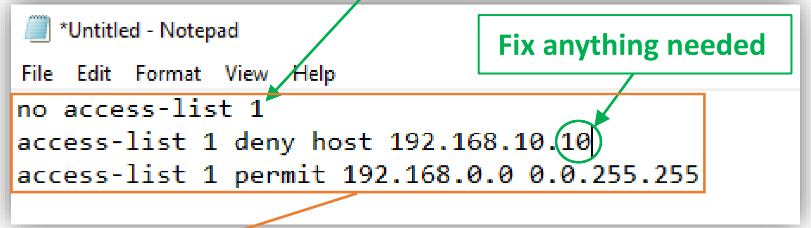
```
R1>en
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#access-list 1 deny host 192.168.10.99
R1(config)#access-list 1 permit 192.168.0.0 0.0.255.255
R1(config)#
```

Mistake 99 should be a 10



```
R1#sh runn | include access-list 1
access-list 1 deny host 192.168.10.99
access-list 1 permit 192.168.0.0 0.0.255.255
R1#
```

Copy/Paste

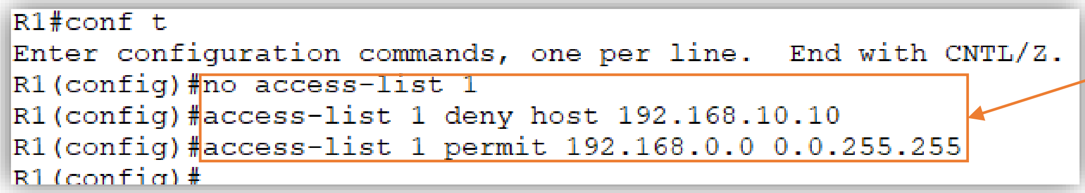


```
no access-list 1
access-list 1 deny host 192.168.10.10
access-list 1 permit 192.168.0.0 0.0.255.255
```

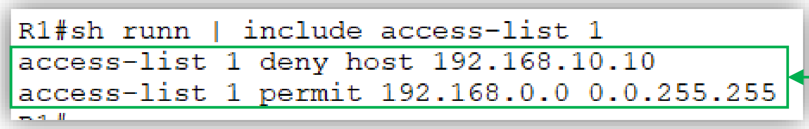
Insert anything needed

Fix anything needed

Copy/Paste



```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#no access-list 1
R1(config)#access-list 1 deny host 192.168.10.10
R1(config)#access-list 1 permit 192.168.0.0 0.0.255.255
R1(config)#
```

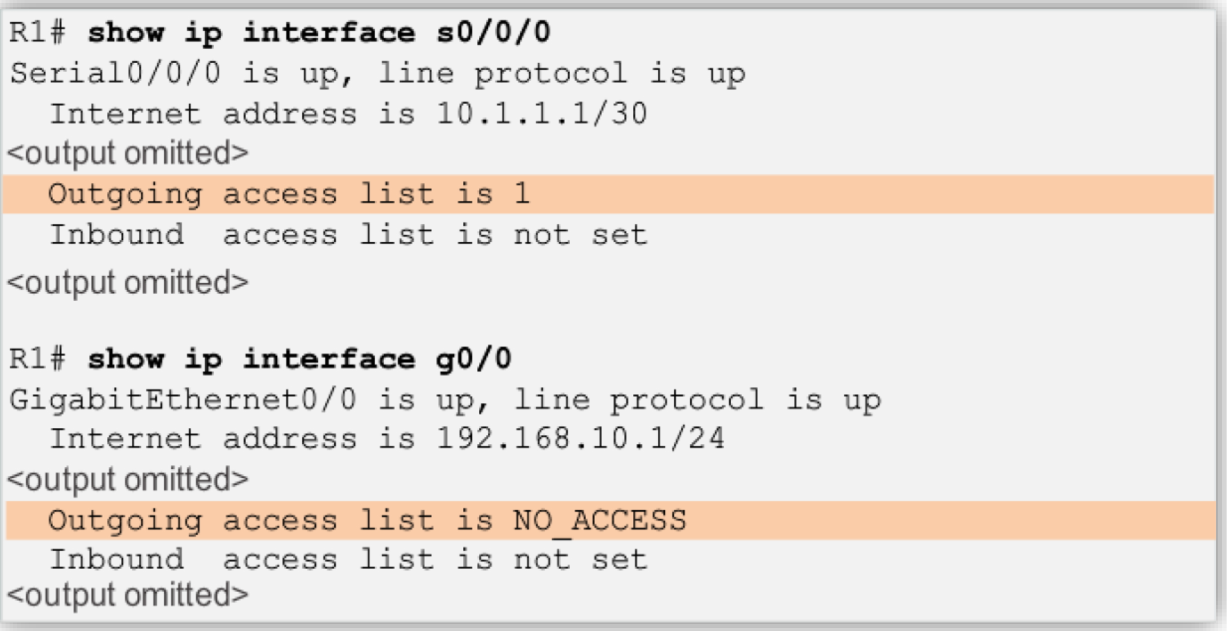


```
R1#sh runn | include access-list 1
access-list 1 deny host 192.168.10.10
access-list 1 permit 192.168.0.0 0.0.255.255
R1#
```

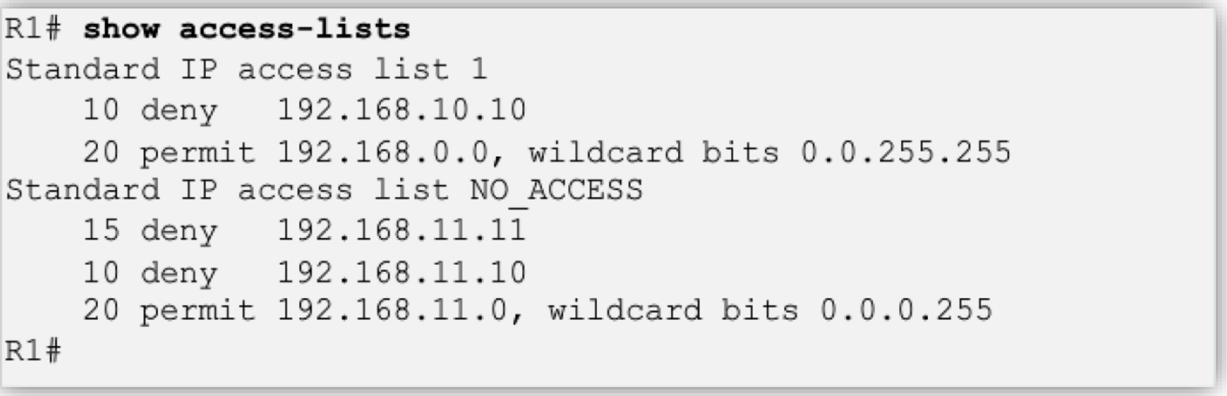
Access-list Fixed

# Modify IPv4 ACLs (continued)

- Verifying 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>
```



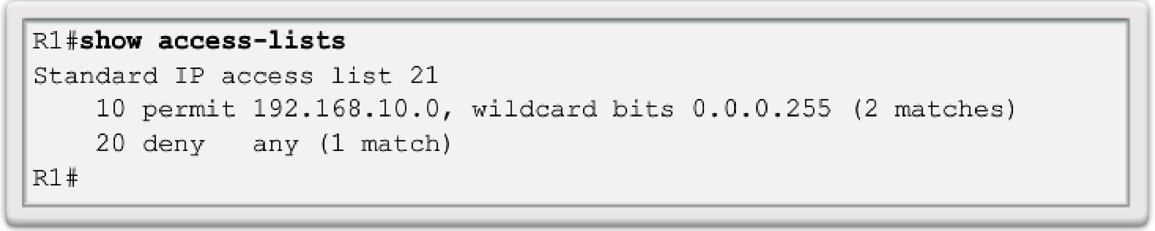
```
R1# show access-lists
Standard IP access list 1
    10 deny   192.168.10.10
    20 permit 192.168.0.0, wildcard bits 0.0.255.255
Standard IP access list NO_ACCESS
    15 deny   192.168.11.11
    10 deny   192.168.11.10
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1#
```

# Modify IPv4 ACLs (continued)

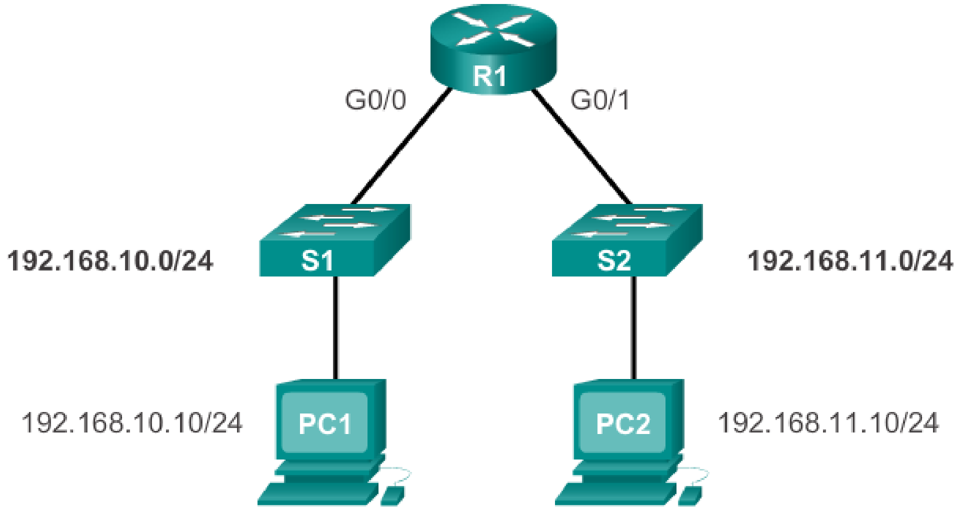
- Standard ACL Sequence Numbers
  - Another part of the IOS internal logic involves the internal sequencing of standard ACL statements. Range statements that deny three networks are configured first followed by five host statements. The host statements are all valid statements because their host IP addresses are not part of the previously entered range statements.
  - The host statements are listed first by the show command, but not necessarily in the order that they were entered. The IOS puts host statements in an order using a special hashing function. The resulting order optimizes the search for a host ACL entry.

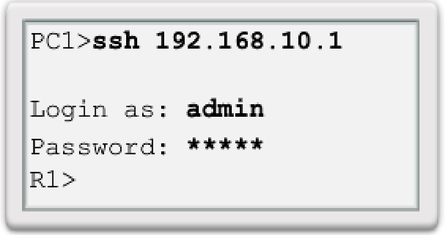
# Securing VTY ports with a Standard IPv4 ACL

- Configuring a Standard ACL to Secure a VTY Port
  - Filtering Telnet or SSH traffic is typically considered an extended IP ACL function because it filters a higher level protocol. However, because the **access-class** command is used to filter incoming or outgoing Telnet/SSH sessions by source address, a standard ACL can be used.
  - `Router(config-line)# access-class access-list-number{in[vrf-also]|out}`

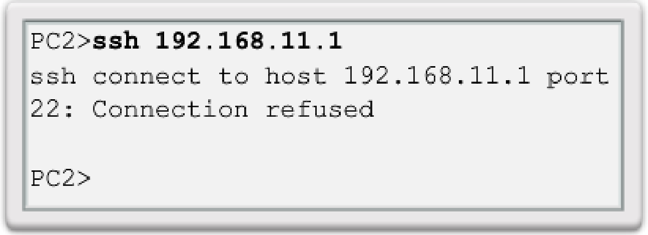


```
R1#show access-lists
Standard IP access list 21
    10 permit 192.168.10.0, wildcard bits 0.0.0.255 (2 matches)
    20 deny   any (1 match)
R1#
```





```
PC1>ssh 192.168.10.1

Login as: admin
Password: *****
R1>
```



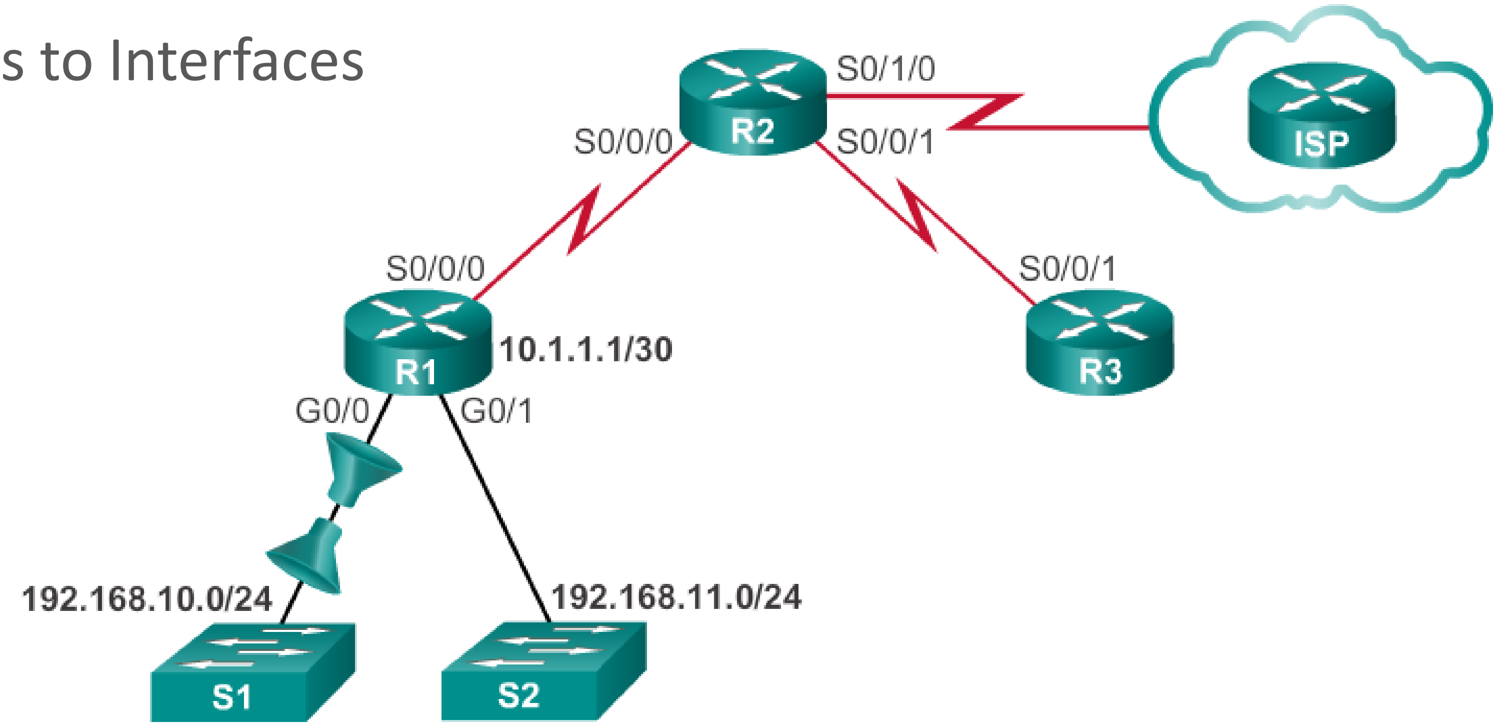
```
PC2>ssh 192.168.11.1
ssh connect to host 192.168.11.1 port
22: Connection refused

PC2>
```

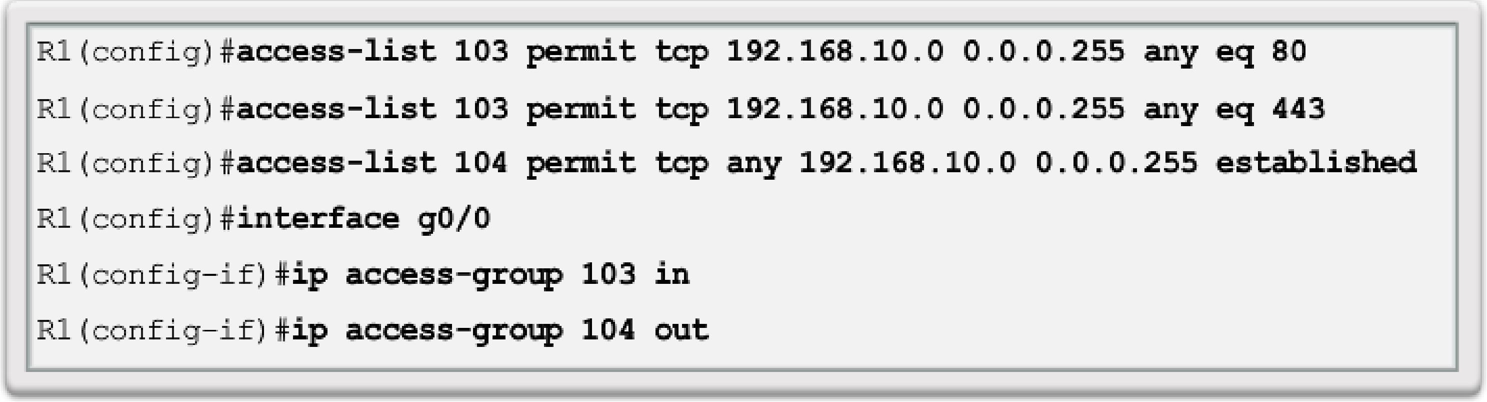
# Configure Extended IPv4 ACLs

- 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 100-199 {permit|deny|remark} {ip|tcp|udp|icmp} source-ip source-wildcard [lt|gt|eq|neq] [source-port or name] destination-ip dest-wildcard [lt|gt|eq|neq] [dest-port or name] [established]**
  - The "**established**" keyword is used to indicate an established connection for TCP protocol. This means that the packets belong to an existing connection if the Transmission Control Protocol (TCP) segment has the Acknowledgment (ACK) or Reset (RST) bit set. ...

# Configure Extended IPv4 ACLs (continued)

- Applying Extended ACLs to Interfaces



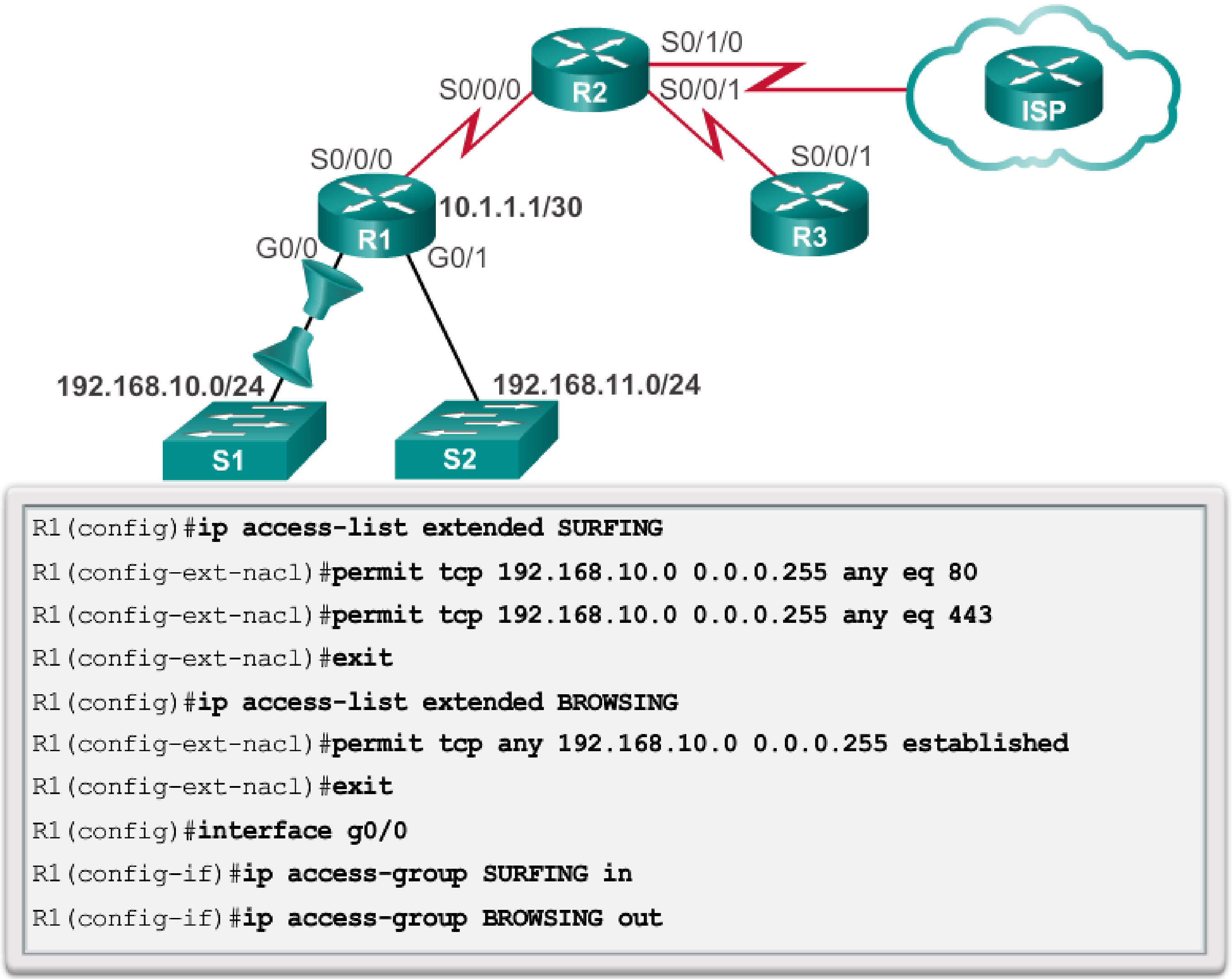


```
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)#access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
R1(config)#interface g0/0
R1(config-if)#ip access-group 103 in
R1(config-if)#ip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

# Configure Extended IPv4 ACLs (continued)

- Creating Named Extended 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
```

# Processing Packets with ACLs

- Inbound ACL Logic
  - Packets are tested against an inbound ACL, if one exists, before being routed.
  - If an inbound packet matches an ACL statement with a **permit**, it is sent to be routed.
  - If an inbound packet matches an ACL statement with a **deny**, it is dropped and not routed.
  - If an inbound packet does not meet any ACL statements, then it is “**implicitly denied**” and dropped without being routed.

- Outbound ACL Logic
  - Packets are first checked for a route before being sent to an outbound interface. If there is no route, the packets are dropped.
  - If an outbound interface has no ACL, then the packets are sent directly to that interface.
  - If there is an ACL on the outbound interface, it is tested before being sent to that interface.
  - If an outbound packet matches an ACL statement with a **permit**, it is sent to the interface.
  - If an outbound packet matches an ACL statement with a **deny**, it is dropped.
  - If an outbound packet does not meet any ACL statements, then it is “**implicitly denied**” and dropped.

# Processing Packets with ACLs (continued)

- ACL Logic Operations
  - When a packet arrives at a router interface, the router process is the same, whether ACLs are used or not. As a frame enters an interface, the router checks to see whether the destination Layer 2 address matches its the interface Layer 2 address or if 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 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 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.

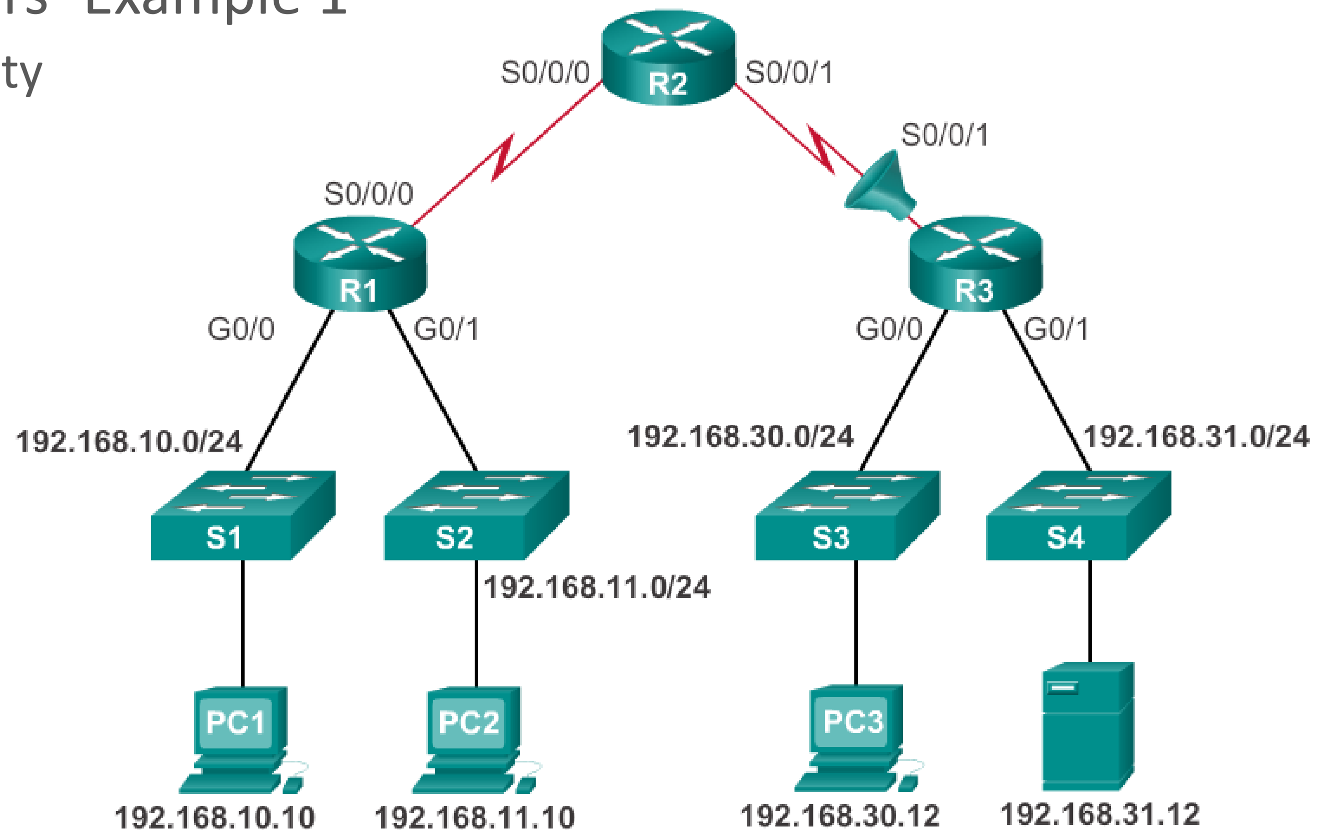
# Processing Packets with ACLs (continued)

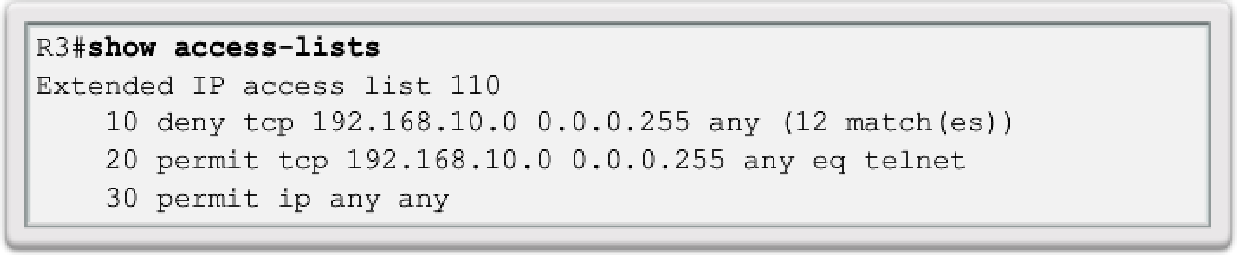
- Standard ACL Decision Process
  - Standard ACLs only examine the **source** IPv4 address. The destination of the packet and the ports involved are not considered.
  - Cisco IOS software tests addresses against the conditions in the ACL. The first match determines whether the software accepts or rejects the address. Because the software stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the address is rejected.

- Extended ACL Decision Process
  - The ACL first filters on the **source** address, then on the **port** and protocol of the source. It then filters on the **destination** address, then on the **port** and **protocol** of the destination, and makes a final permit or deny decision.

# Common ACLs Errors

- Troubleshooting Common ACL Errors -Example 1
  - Host 192.168.10.10 has no connectivity with 192.168.30.12.

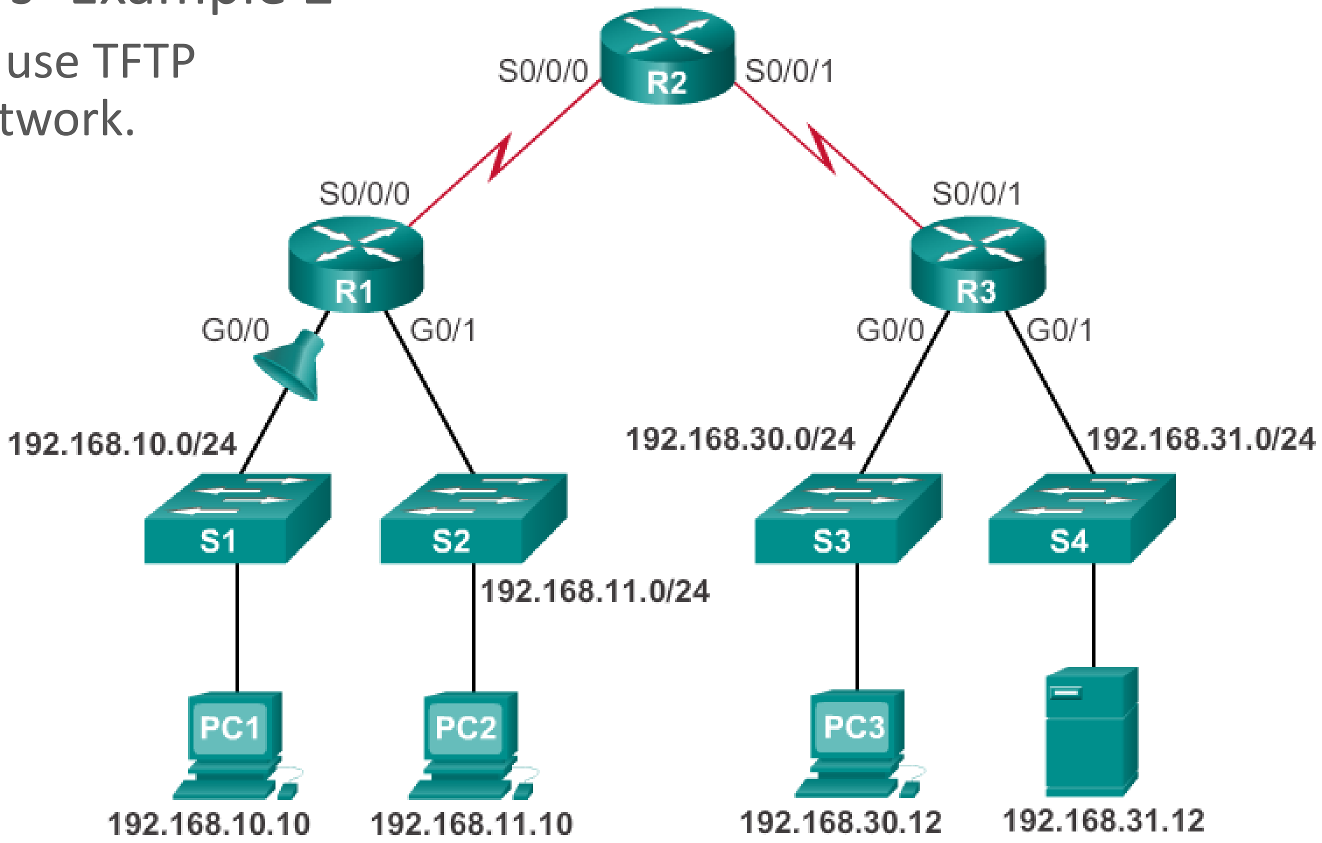


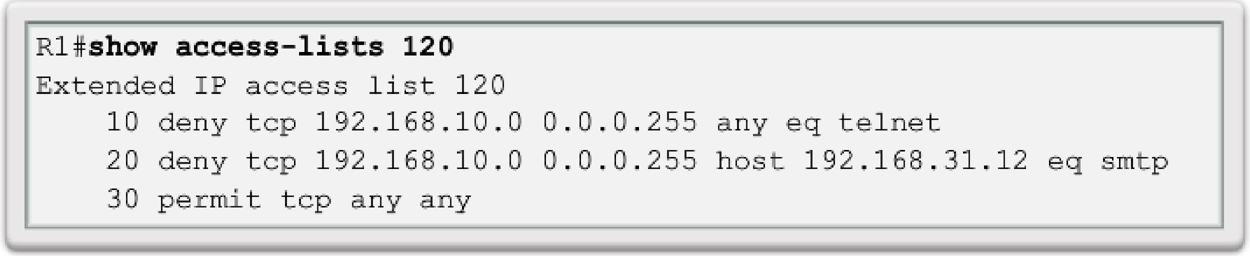


```
R3#show access-lists
Extended IP access list 110
    10 deny tcp 192.168.10.0 0.0.0.255 any (12 match(es))
    20 permit tcp 192.168.10.0 0.0.0.255 any eq telnet
    30 permit ip any any
```

# Common ACLs Errors (continued)

- Troubleshooting Common ACL Errors -Example 2
  - The 192.168.10.0 /24 network cannot use TFTP to connect to the 192.168.30.0 /24 network.

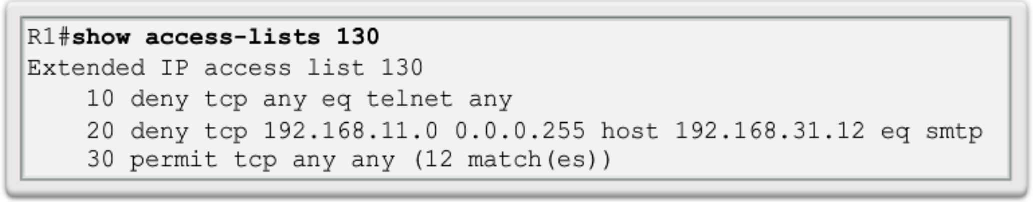




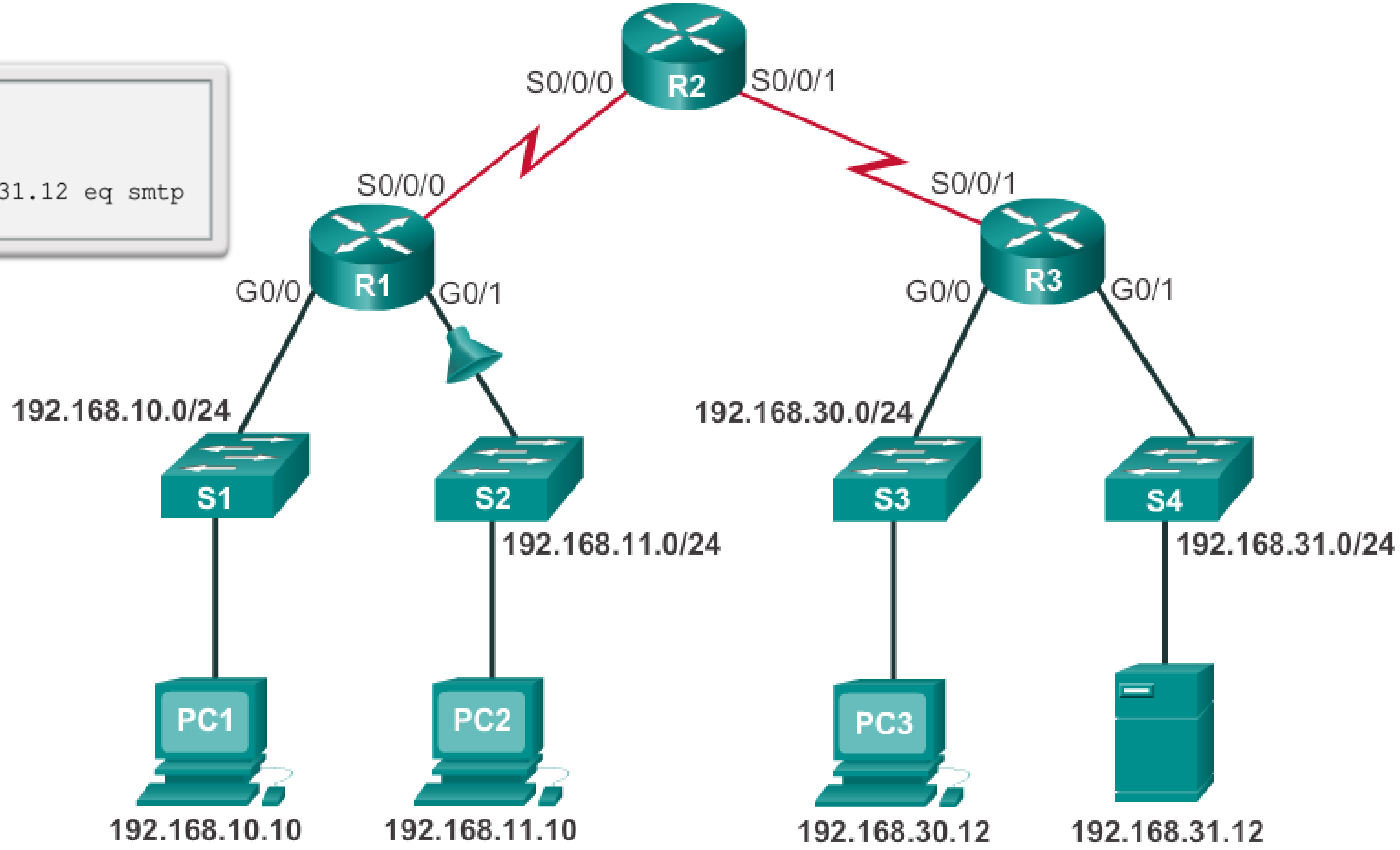
```
R1#show access-lists 120
Extended IP access list 120
    10 deny tcp 192.168.10.0 0.0.0.255 any eq telnet
    20 deny tcp 192.168.10.0 0.0.0.255 host 192.168.31.12 eq smtp
    30 permit tcp any any
```

# Common ACLs Errors (continued)

- Troubleshooting Common ACL Errors -Example 3
  - The 192.168.11.0 /24 network can use Telnet to connect to 192.168.30.0 /24, but according to company policy, this connection should not be allowed.

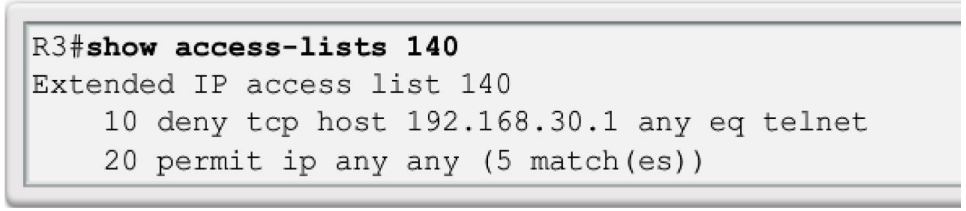


```
R1#show access-lists 130
Extended IP access list 130
    10 deny tcp any eq telnet any
    20 deny tcp 192.168.11.0 0.0.0.255 host 192.168.31.12 eq smtp
    30 permit tcp any any (12 match(es))
```

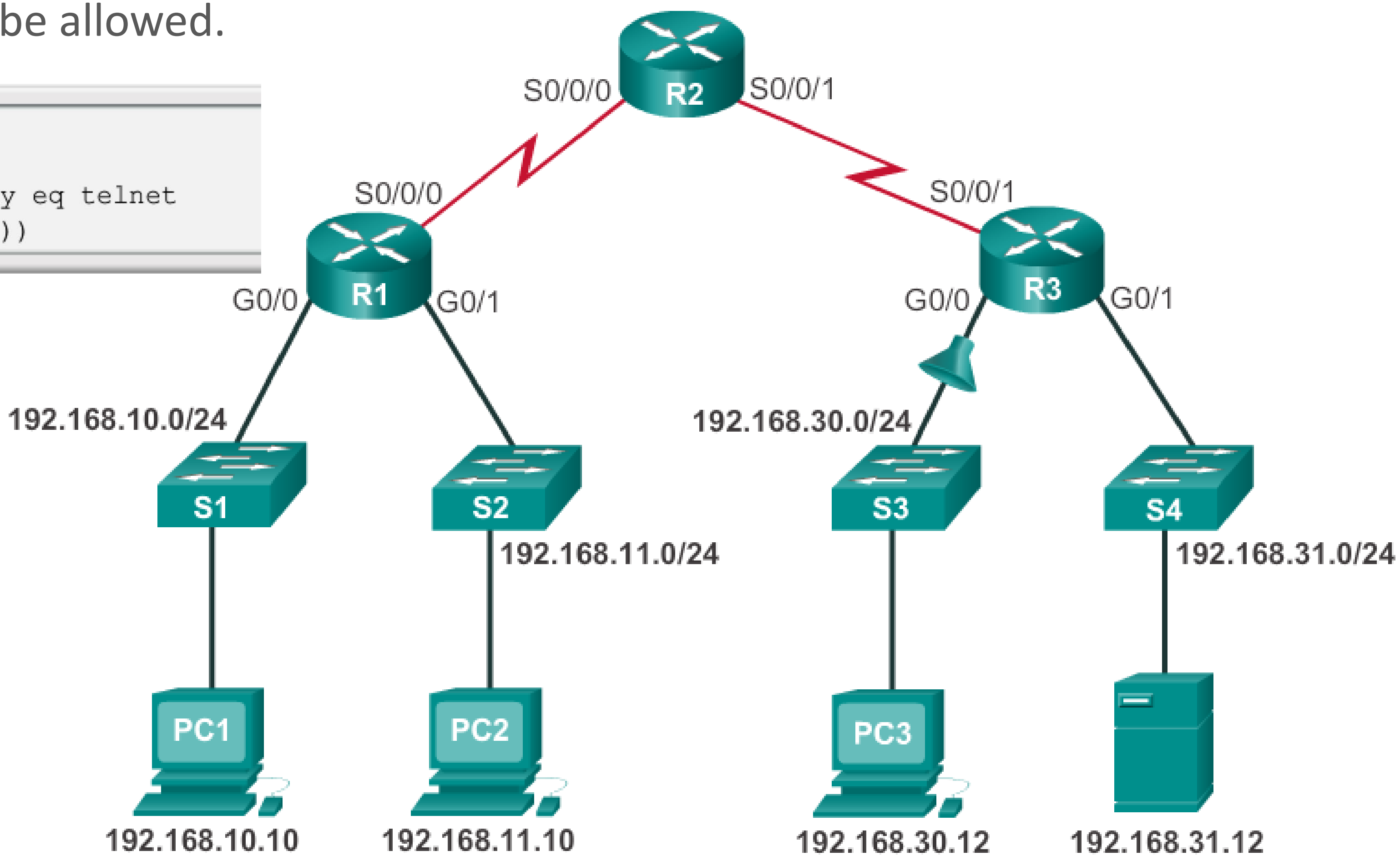


# Common ACLs Errors (continued)

- Troubleshooting Common ACL Errors -Example 4
  - Host 192.168.30.12 is able to Telnet to connect to 192.168.31.12, but company policy states that this connection should not be allowed.

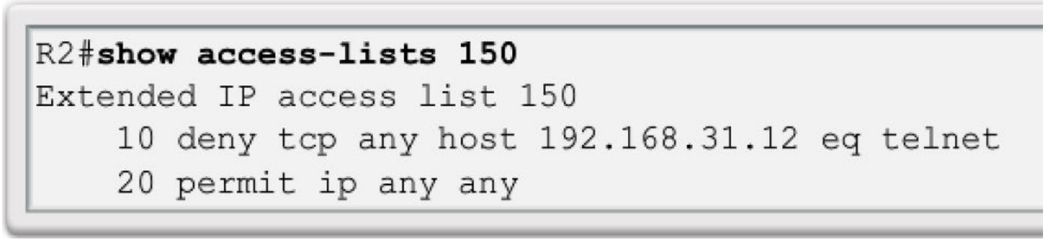


```
R3#show access-lists 140
Extended IP access list 140
    10 deny tcp host 192.168.30.1 any eq telnet
    20 permit ip any any (5 match(es))
```

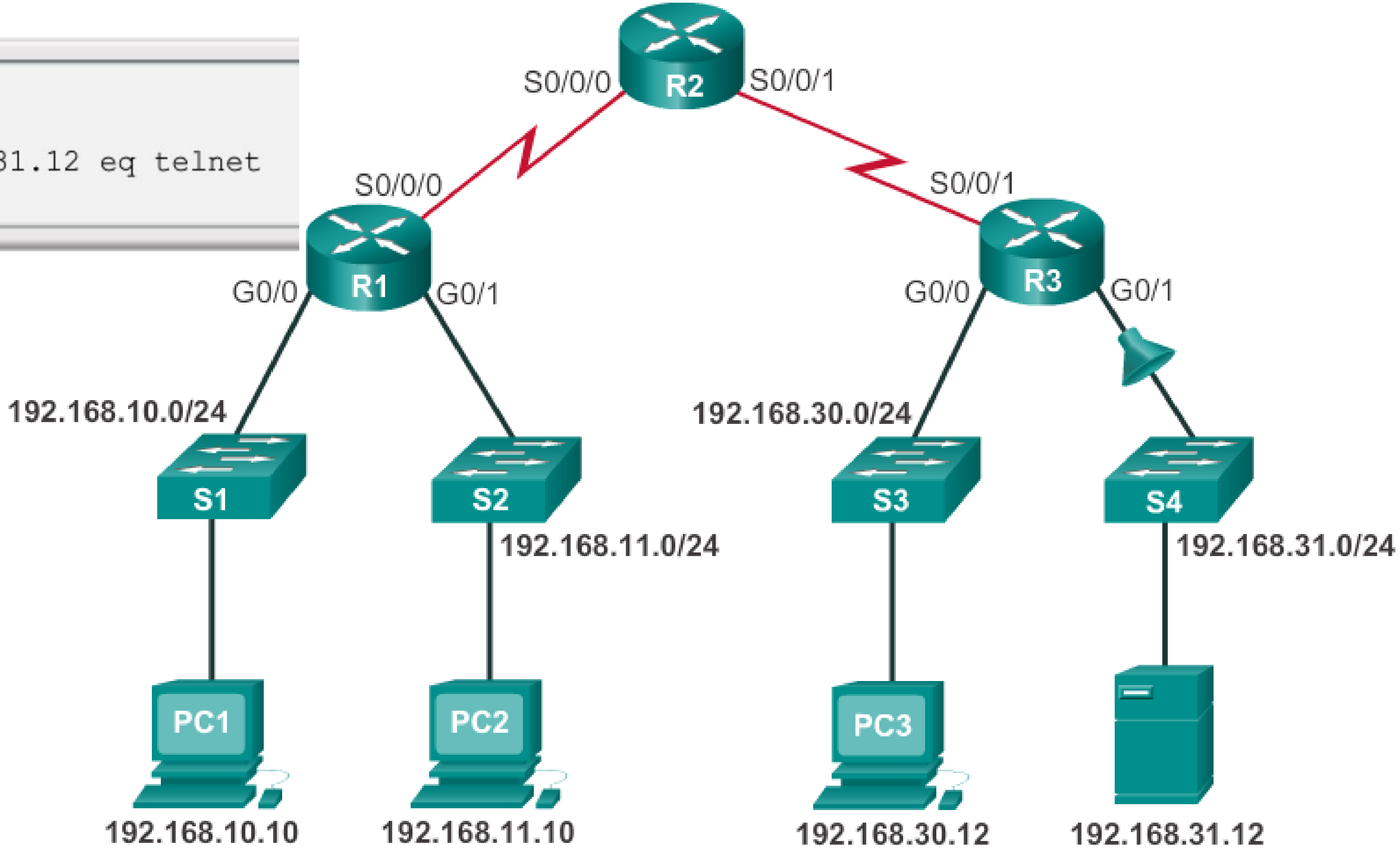


# Common ACLs Errors (continued)

- Troubleshooting Common ACL Errors -Example 5
  - Host 192.168.30.12 can use Telnet to connect to 192.168.31.12, but according to the security policy, this connection should not be allowed.

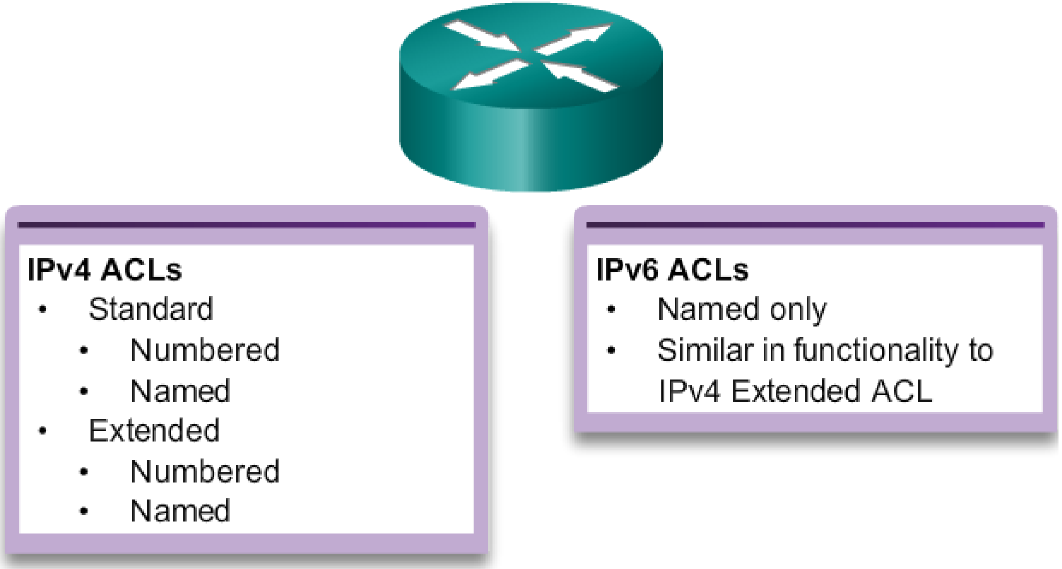


```
R2#show access-lists 150
Extended IP access list 150
    10 deny tcp any host 192.168.31.12 eq telnet
    20 permit ip any any
```



# IPv6 ACL Creation

- Type of IPv6 ACLs



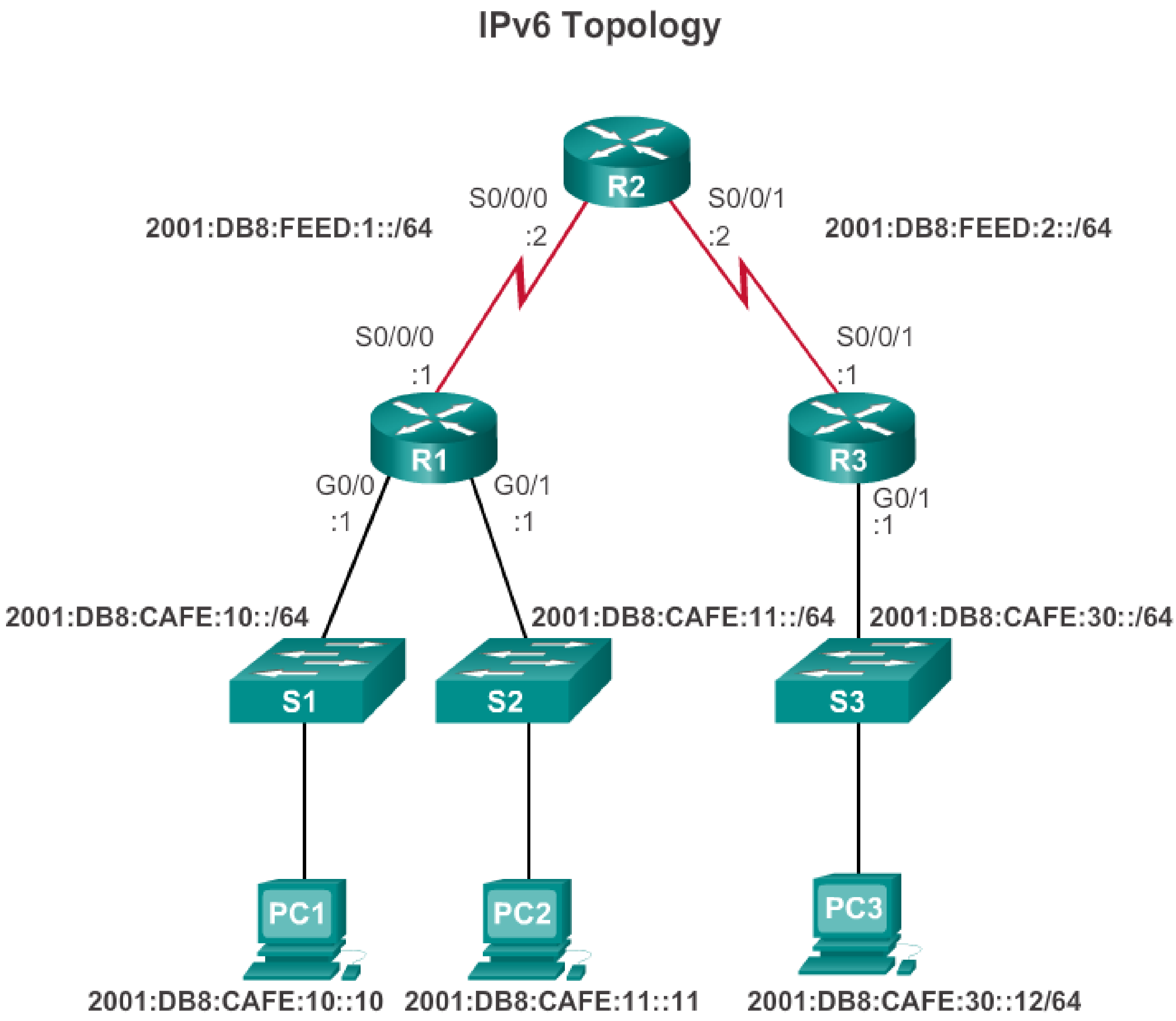
**IPv4 ACLs**
- Standard
  - Numbered
  - Named
- Extended
  - Numbered
  - Named

**IPv6 ACLs**
- Named only
- Similar in functionality to IPv4 Extended ACL

- 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

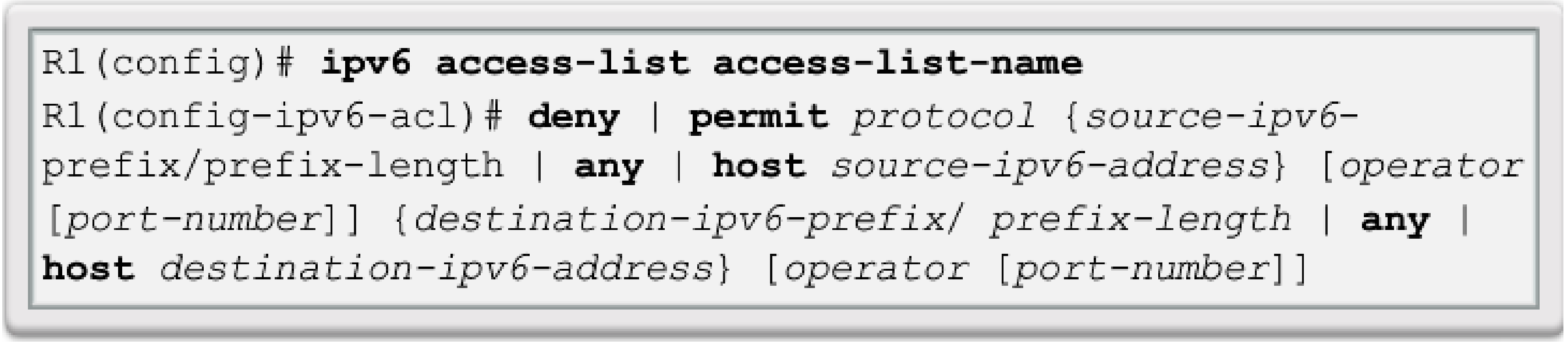
# Configuring IPv6 ACLs

- Configuring IPv6 Topology



# Configuring IPv6 ACLs (continued)

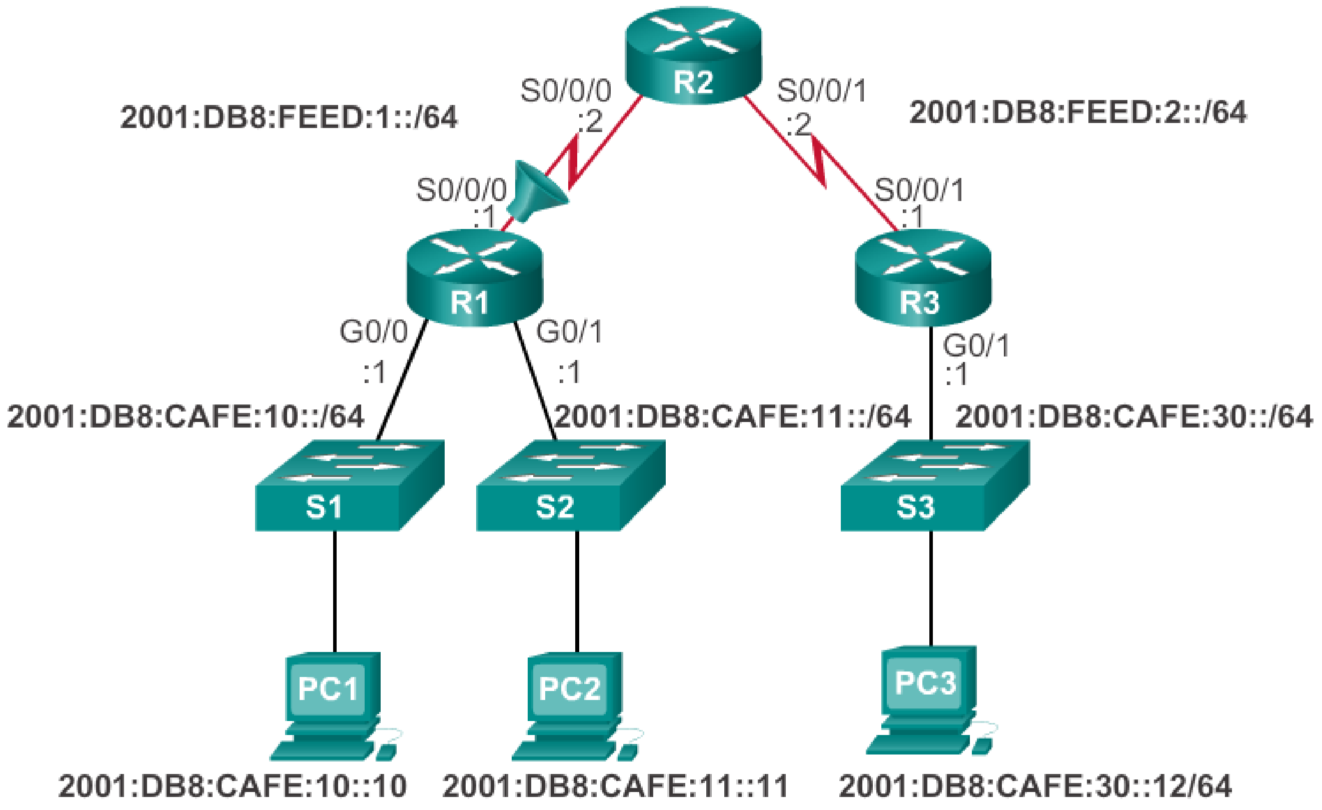
- Configuring IPv6 ACLs
  - There are three basic steps to configure an IPv6 ACL:
    1. From global configuration mode, use the **ipv6 access-list name** command to create an IPv6 ACL.
    2. From the named ACL configuration mode, use the **permit** or **deny** statements to specify one or more conditions to determine if a packet is forwarded or dropped.
    3. Return to privileged EXEC mode with the end command.

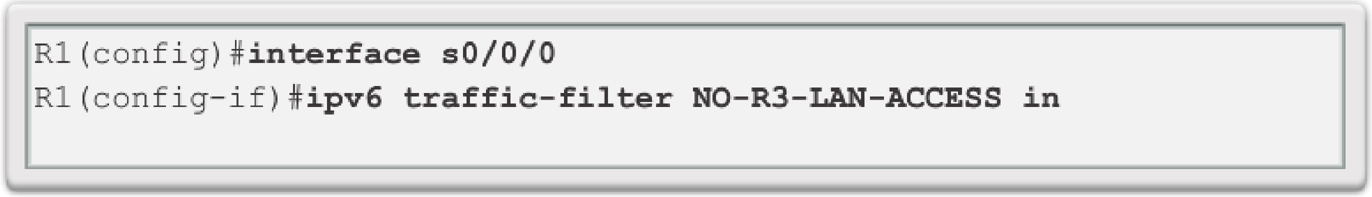


```
R1(config)# ipv6 access-list access-list-name
R1(config-ipv6-acl)# deny | permit protocol {source-ipv6-
prefix/prefix-length | any | host source-ipv6-address} [operator
[port-number]] {destination-ipv6-prefix/ prefix-length | any |
host destination-ipv6-address} [operator [port-number]]
```

# Configuring IPv6 ACLs (continued)

- Applying an IPv6 ACL to an Interface



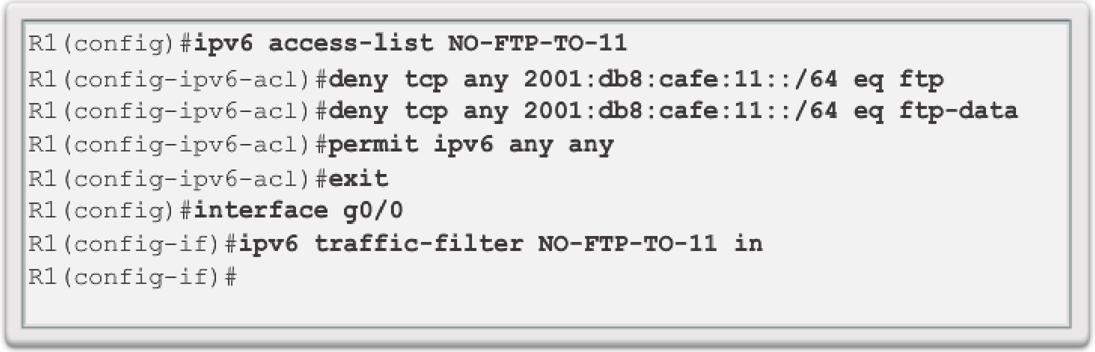


```
R1(config)#interface s0/0/0
R1(config-if)#ipv6 traffic-filter NO-R3-LAN-ACCESS in
```

# Configuring IPv6 ACLs (continued)

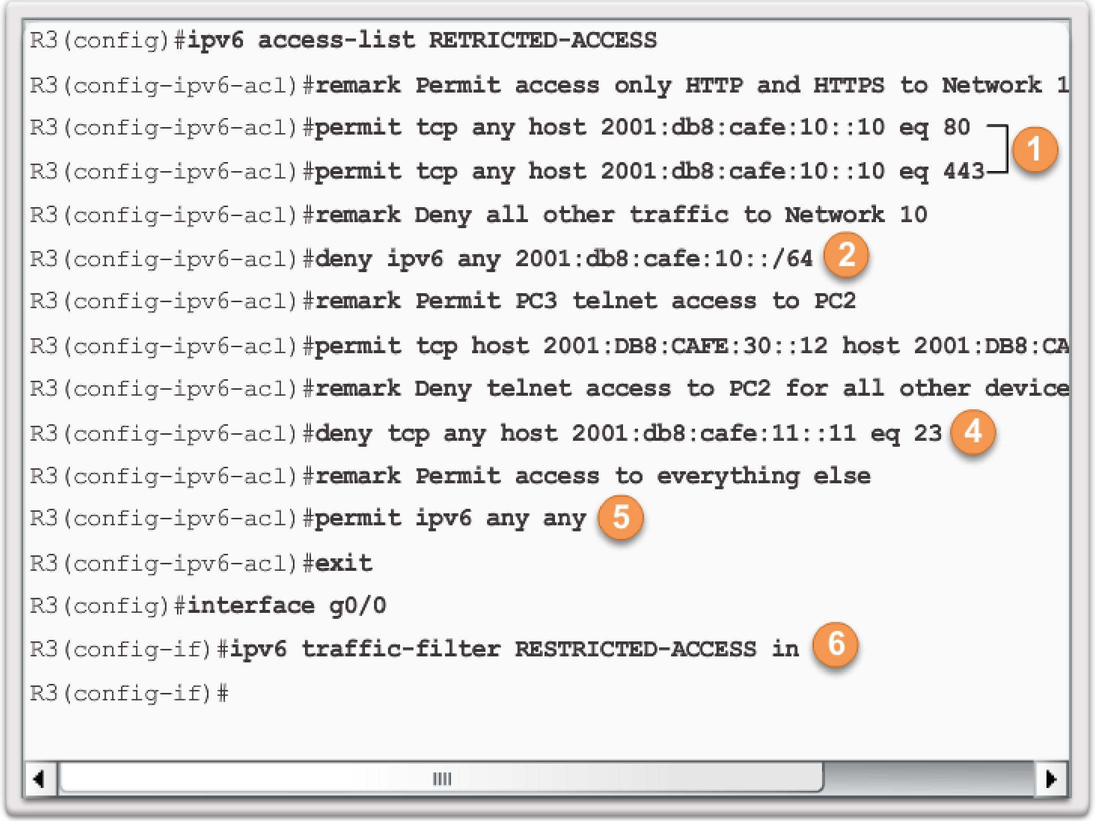
- IPv6 ACL Examples

Deny FTP



```
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)#
```

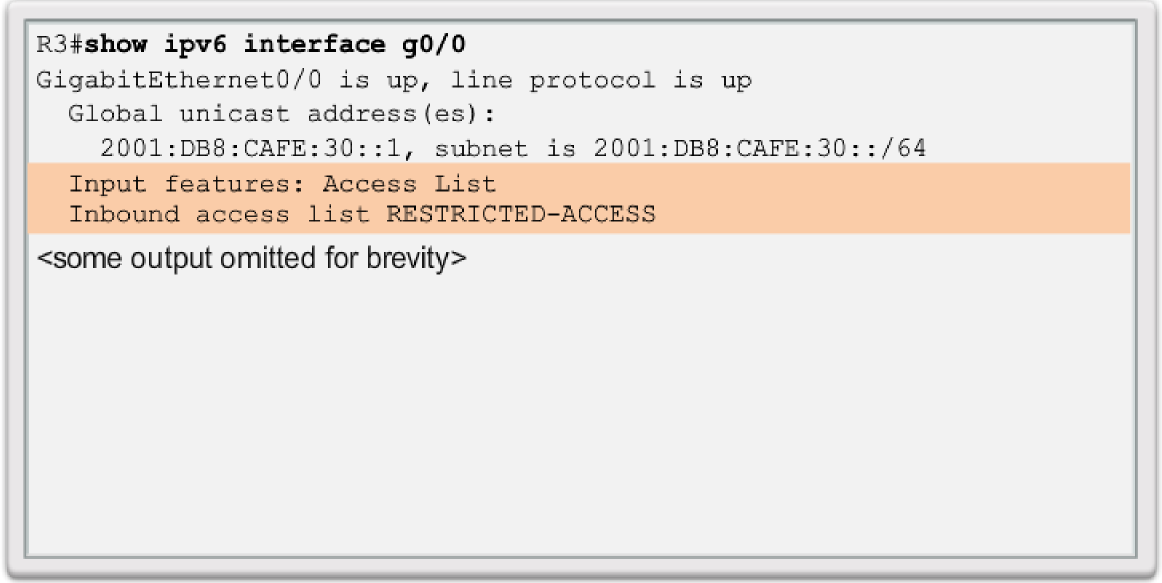
Restrict Access



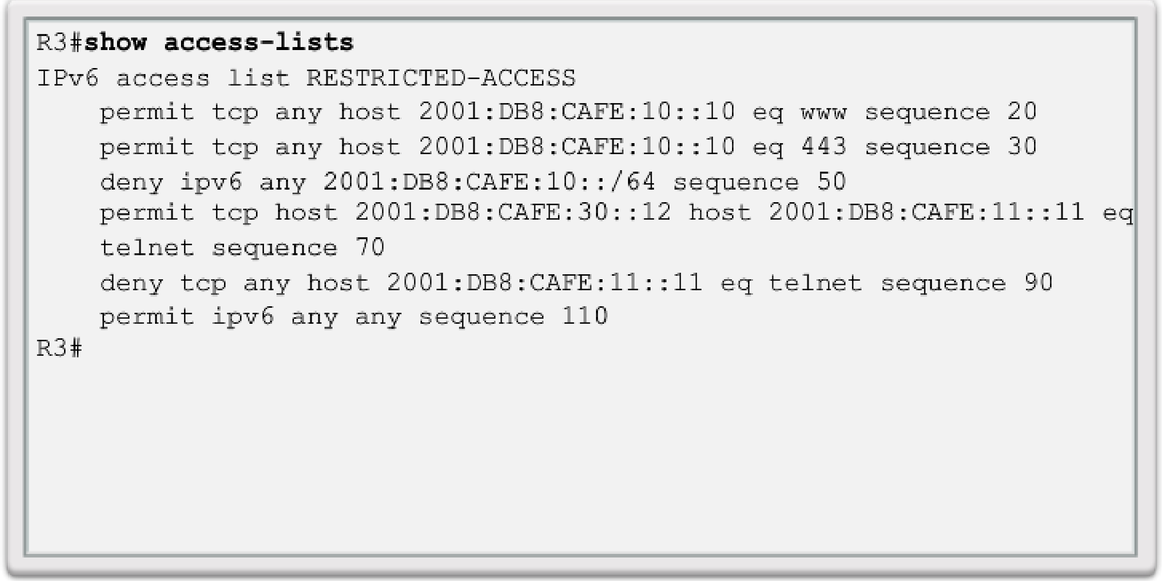
```
R3(config)#ipv6 access-list RETRICTED-ACCESS
R3(config-ipv6-acl)#remark Permit access only HTTP and HTTPS to Network 1
R3(config-ipv6-acl)#permit tcp any host 2001:db8:cafe:10::10 eq 80
R3(config-ipv6-acl)#permit tcp any host 2001:db8:cafe:10::10 eq 443
R3(config-ipv6-acl)#remark Deny all other traffic 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:CA
R3(config-ipv6-acl)#remark Deny telnet access to PC2 for all other device
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 (continued)

- 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
<some output omitted for brevity>
```



```
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#
```

# Summary

- By default, a router does not filter traffic. Traffic that enters the router is routed solely based on information within the routing table.
- **Packet filtering**, controls access to a network by analyzing the incoming and outgoing packets and passing or dropping them based on criteria such as the source IP address, destination IP addresses, and the protocol carried within the packet.
- A packet-filtering router uses rules to determine whether to permit or deny traffic. A router can also perform packet filtering at Layer 4, the transport layer.
- **An ACL is a sequential list of permit or deny statements.**
- The last statement of an ACL is always an implicit deny 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 are configured to apply to inbound traffic or to apply to outbound traffic.

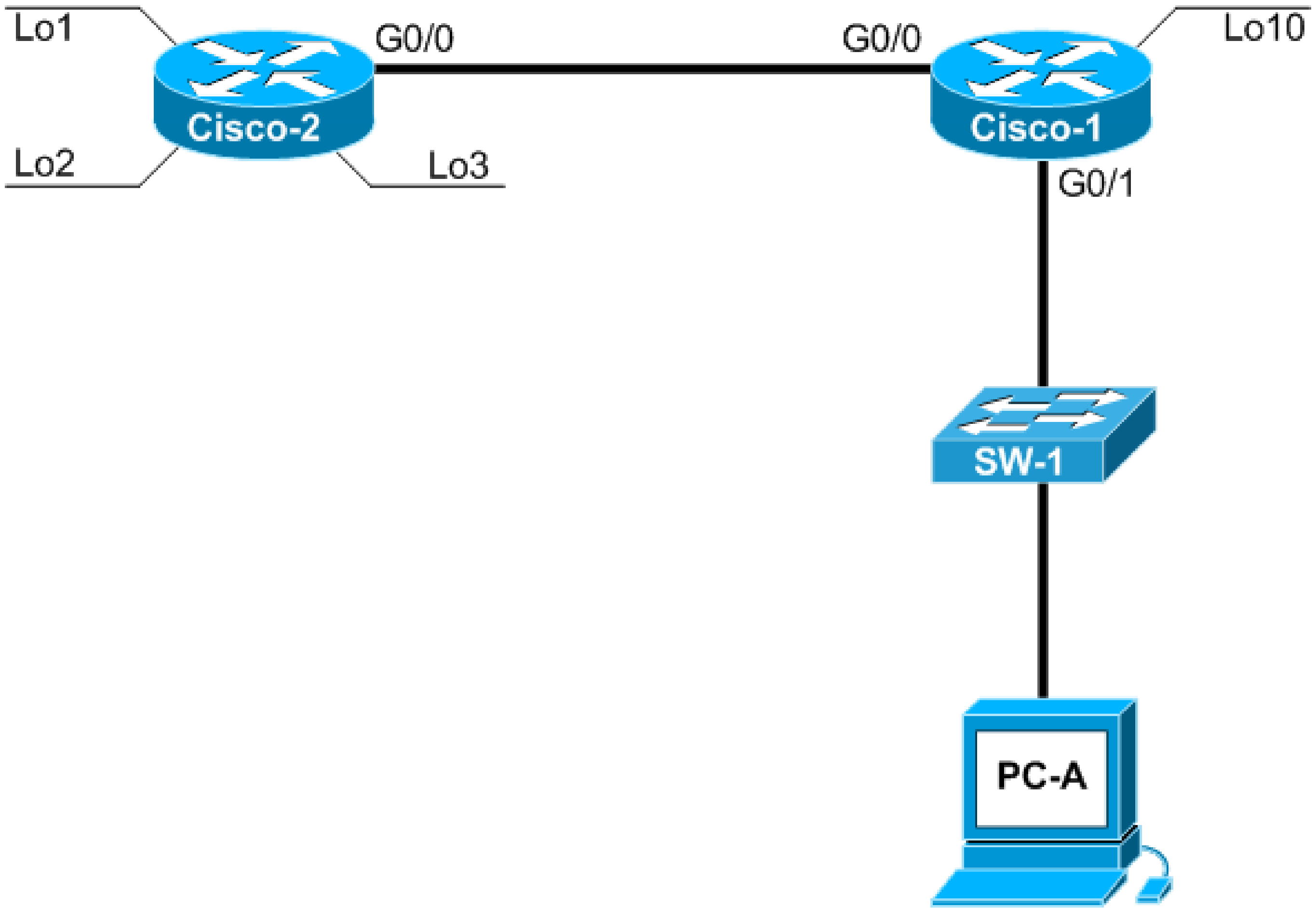
# Summary (continued)

- **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. 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.
- The access-list global configuration command defines a standard ACL with a number in the range of 1 to 99 or an extended ACL with numbers in the range of 100 to 199 and 2000 to 2699. Both standard and extended ACLs can be named.
- 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. IPv4 ACL statements include the use of wildcard masks.
- After an ACL is configured, it is linked to an interface using the **ip access-group** command in interface configuration mode.

# Summary (continued)

- Remember the three **P**s, 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 restricts incoming and outgoing connections between a particular VTY and the addresses in an access list.
- Like IPv4 named ACLs, IPv6 names are alphanumeric, case sensitive and **must be unique**. Unlike IPv4, there 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. 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.

# Lab



# Lab (continued)

| Device | interface | IP Address | Subnet Mask | Default Gateway |
|---|---|---|---|---|
| Cisco-1 | G0/0 | 172.16.1.22 | /30 | |
| | G0/1 | 172.16.1.62 | /27 | |
| | Lo10 | 10.10.10.1 | /24 | |
| Cisco-2 | G0/0 | 172.16.1.21 | /30 | |
| | Lo1 | 192.168.1.1 | /24 | |
| | Lo2 | 192.168.2.1 | /24 | |
| | Lo3 | 192.168.3.1 | /30 | |
| SW-1 | | | | |
| PC-A | | 172.16.1.33 | /27 | 172.16.1.62 |

| Device | interface | IP Address | Subnet Mask | Default Gateway |
|---|---|---|---|---|
| Cisco-1 | G0/0 | 2001:db8:acad:2::2 | /64 | N/A |
| | G0/1 | 2001:db8:acad:1::254 | /64 | N/A |
| | Lo10 | 2001:db8:acad:100::1 | /64 | N/A |
| Cisco-2 | G0/0 | 2001:db8:acad:2::1 | /64 | N/A |
| | Lo1 | 2001:db8:acad:10::1 | /64 | N/A |
| | Lo2 | 2001:db8:acad:20::1 | /64 | N/A |
| | Lo3 | 2001:db8:acad:30::1 | /64 | N/A |
| SW-1 | N/A | N/A | N/A | N/A |
| PC-A | | 2001:db8:acad:1::10 | /64 | 2001:db8:acad:1::254 |

FANSHAWE

# QUESTIONS

?