Access Control Lists
and
DHCP

Introduction to Networks v6.0

Chapter 7: Access Control Lists

Pertemuan ke 21

Kompetensi Khusus

- Mahasiswa mampu melakukan konfigurasi DHCP pada router untuk melakukan pembagian IP Address secara otomatis pada perangkat yang terkoneksi dalam jaringan (C3)

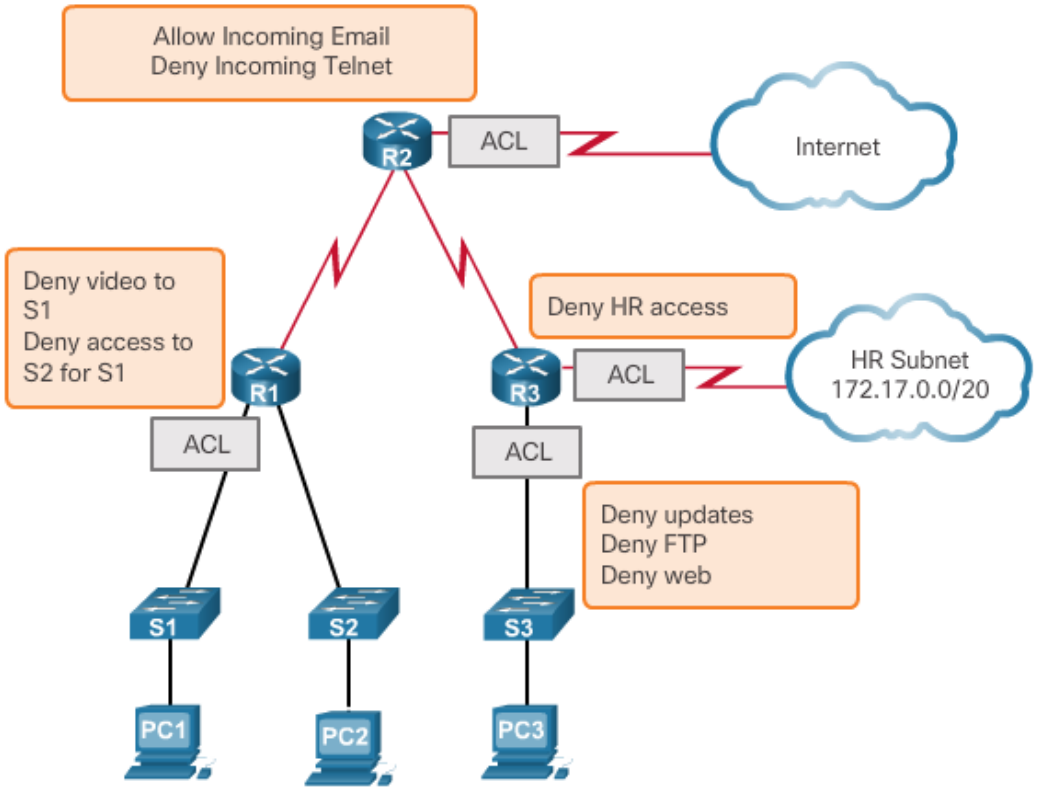
Materi:

1. ACL Operation
2. Standard IPv4 ACLs
3. Troubleshoot ACLs
4. DHCPv4
5. DHCPv6

1. ACL Operation

1.1 What is an ACL?

- By default, a router does not have ACLs configured; therefore, by default a router does not filter traffic.



1.2 Packet Filtering

- 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).

1.3 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.

1.4 Introducing ACL Wildcard Masking

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

1.4 Introducing ACL Wildcard Masking

Example

	Decimal Address	Binary Address
IP Address to be Processed	192.168.10.0	11000000.10101000.00001010.00000000
Wildcard Mask	0.0.255.255	00000000.00000000.11111111.11111111
Resulting IP Address	192.168.0.0	11000000.10101000.00000000.00000000

1.5 Wildcard Mask Examples

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

1.5 Wildcard Mask Examples

Wildcard Masks to Match Ranges

Example 1

	Decimal	Binary
IP Address	192.168.16.0	11000000.10101000.00010000.00000000
Wildcard Mask	0.0.15.255	00000000.00000000.00001111.11111111
Result Range	192.168.16.0 to 192.168.31.255	11000000.10101000.00010000.00000000 to 11000000.10101000.00011111.11111111

Example 2

	Decimal	Binary
IP Address	192.168.1.0	11000000.10101000.00000001.00000000
Wildcard Mask	0.0.254.255	00000000.00000000.11111110.11111111
Result	192.168.1.0	11000000.10101000.00000001.00000000
	All odd numbered subnets in the 192.168.0.0 major network	

1.6 Calculating the Wildcard Mask

- Calculating wildcard masks can be challenging. One shortcut method is to subtract the subnet mask from 255.255.255.255.

Example 1

	255.255.255.255
–	255.255.255.000
	000.000.000.255

Example 2

	255.255.255.255
–	255.255.255.240
	000.000.000.015

Example 3

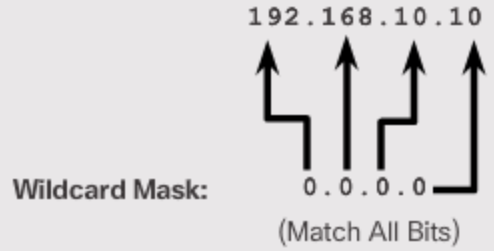
	255.255.255.255
–	255.255.252.000
	000.000.003.255

1.7 Wildcard Mask Keywords

Wildcard Bit Mask Abbreviations

Example 1

- 192.168.10.10 0.0.0.0 matches all of the address bits
- Abbreviate this wildcard mask using the IP address preceded by the keyword `host` (`host 192.168.10.10`)



Example 2

- 0.0.0.0 255.255.255.255 ignores all address bits
- Abbreviate expression with the keyword `any`

1.8 Wildcard Mask Keyword Examples

Example 1:

```
R1(config)# access-list 1 permit 0.0.0.0 255.255.255.255
!OR
R1(config)# access-list 1 permit any
```

Example 2:

```
R1(config)# access-list 1 permit 192.168.10.10 0.0.0.0
!OR
R1(config)# access-list 1 permit host 192.168.10.10
```

This is the format of the `host` and `any` optional keywords in an ACL statement.

1.9 General Guidelines for Creating ACLS

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

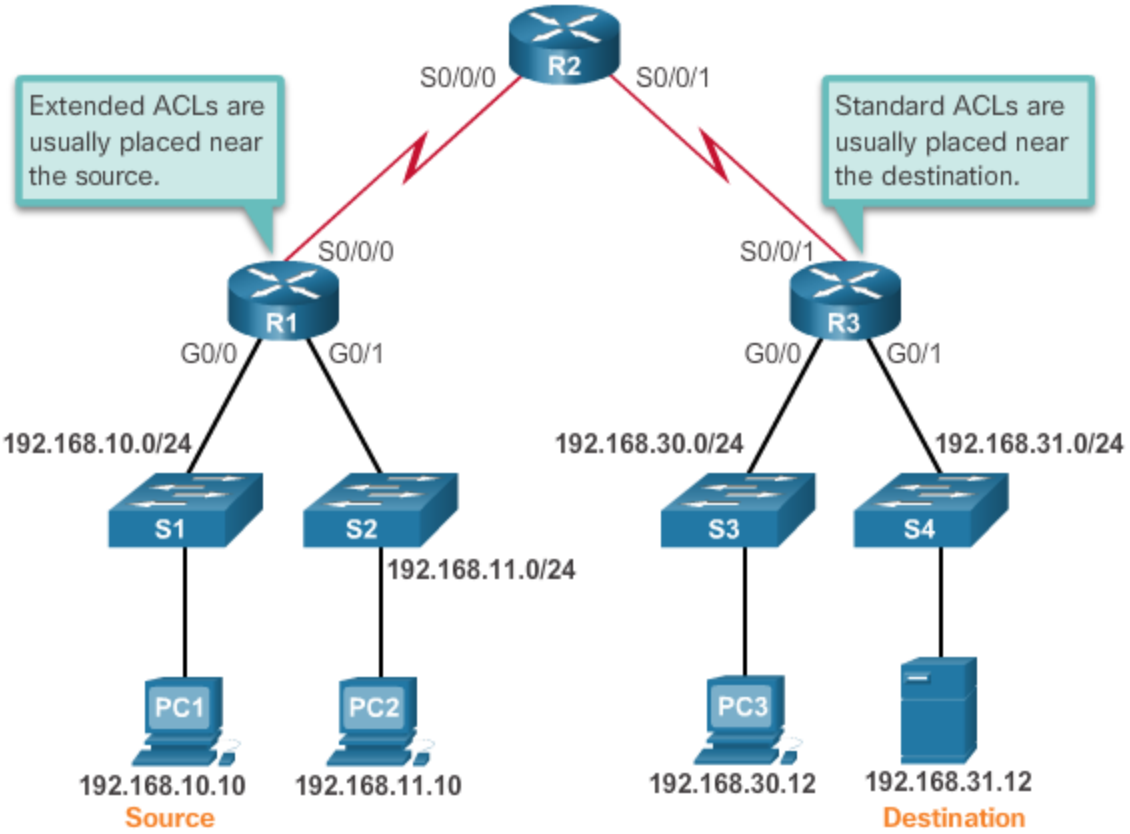
You can only have one ACL per protocol, per interface, and per direction:

- 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)

1.10 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.

1.11 Where to Place ACLs

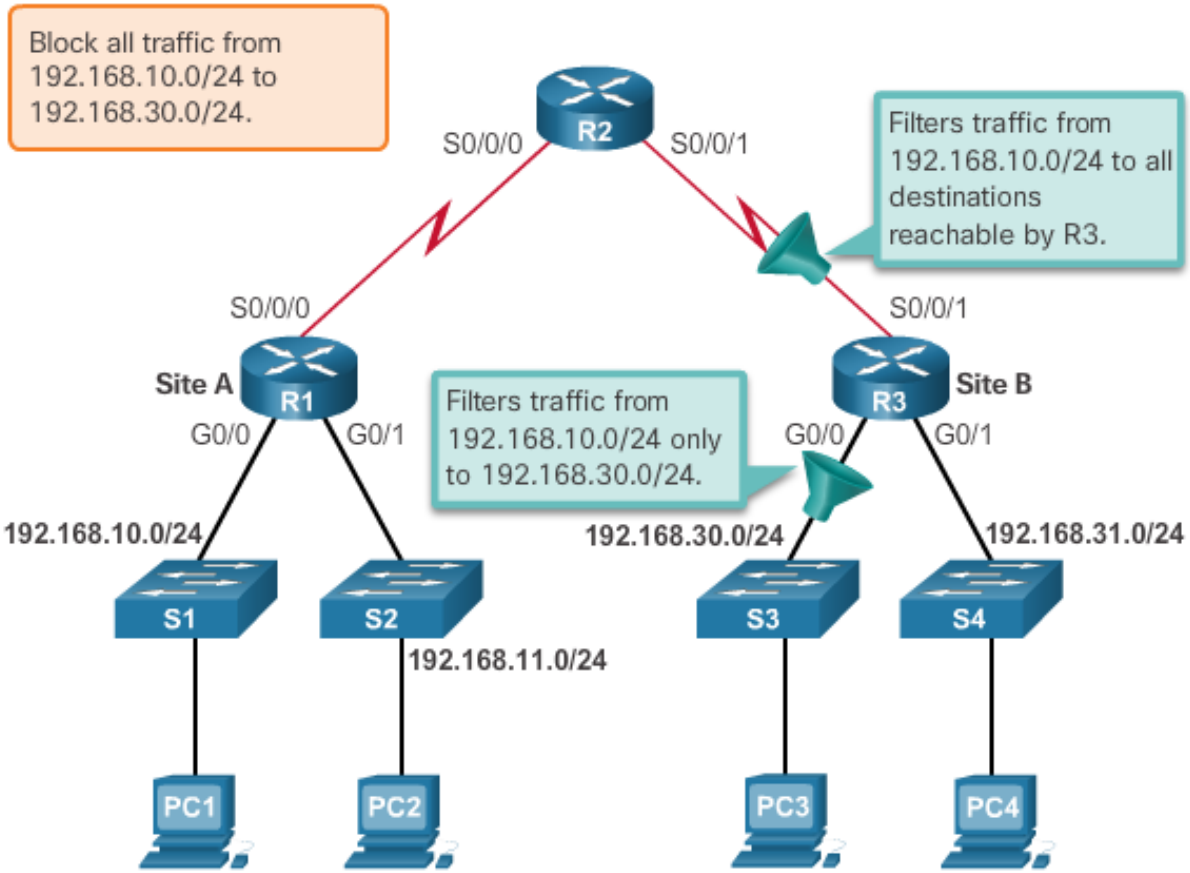


1.11 Where to Place ACLs

- 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.

1.12 Standard ACL Placement

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



2. Standard IPv4 ACLs

2.1 Numbered Standard IPv4 ACL Syntax

- 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#
```

2.2 Applying Standard IPv4 ACLs to Interfaces

Procedure for Configuring Standard ACLs

Step 1: Use the **access-list** global configuration command to create an entry in a standard IPv4 ACL.

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

The example statement matches any address that starts with 192.168.10.x. Use the **remark** option to add a description to your ACL.

Step 2: Use the **interface** configuration command to select an inteface to which to apply the ACL.

```
R1(config)# interface serial 0/0/0
```

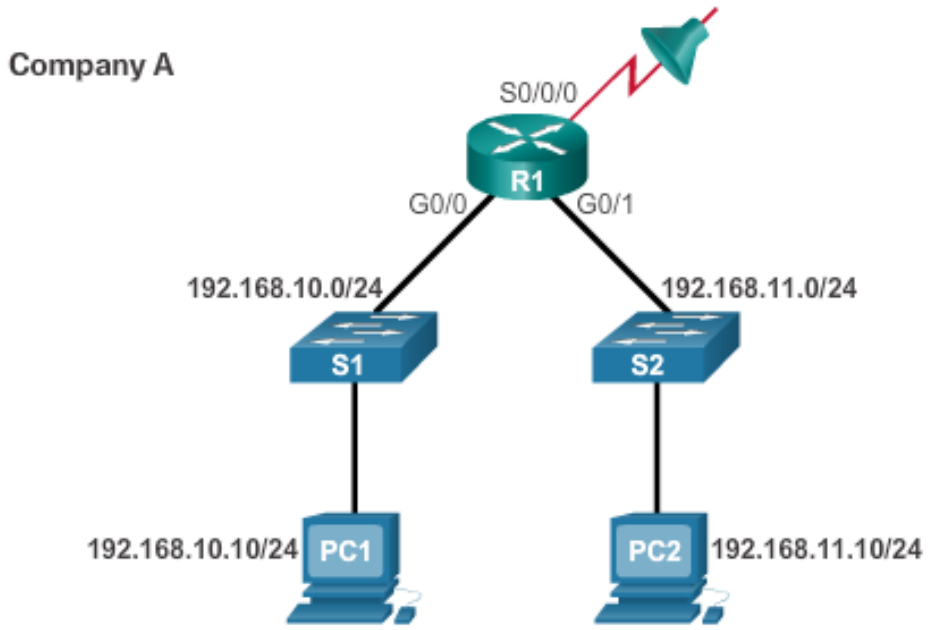
Step 3: Use the **ip access-group** interface configuration command to activate the existing ACL on an interface.

```
R1(config-if)# ip access-group 1 out
```

This example activates the standard IPv4 ACL 1 on the interface as an outbound filter.

2.2 Applying Standard IPv4 ACLs to Interfaces

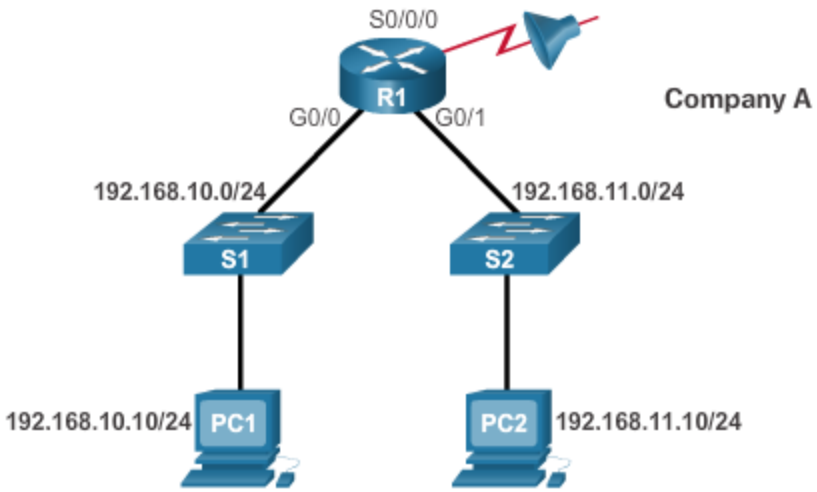
Permit a Specific Subnet




```
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
```

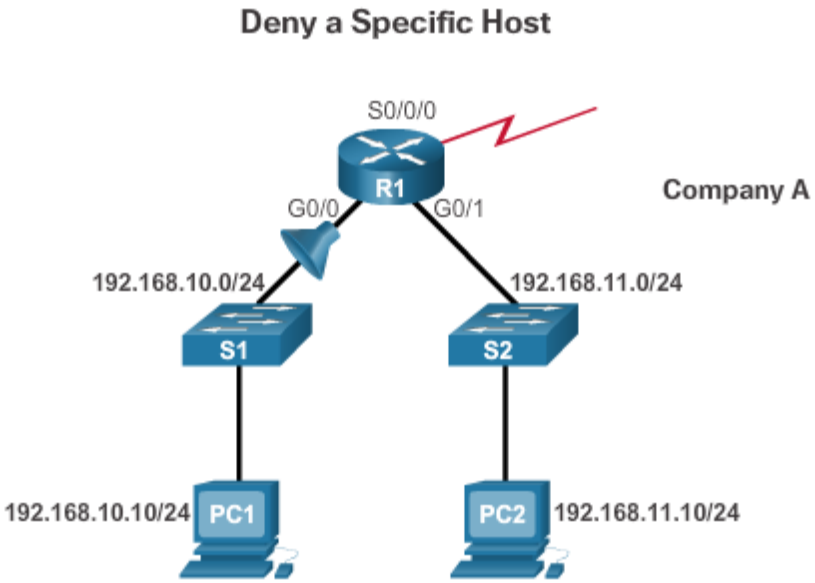
2.3 Numbered Standard IPv4 ACL Examples

Deny a Specific Host and Permit a Specific Subnet




```
R1(config)# no access-list 1
R1(config)# access-list 1 deny host 192.168.10.10
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)# interface s0/0/0
R1(config-if)# ip access-group 1 out
```

2.3 Numbered Standard IPv4 ACL Examples




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

2.4 Named Standard IPv4 ACL Syntax

Named ACL Example

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

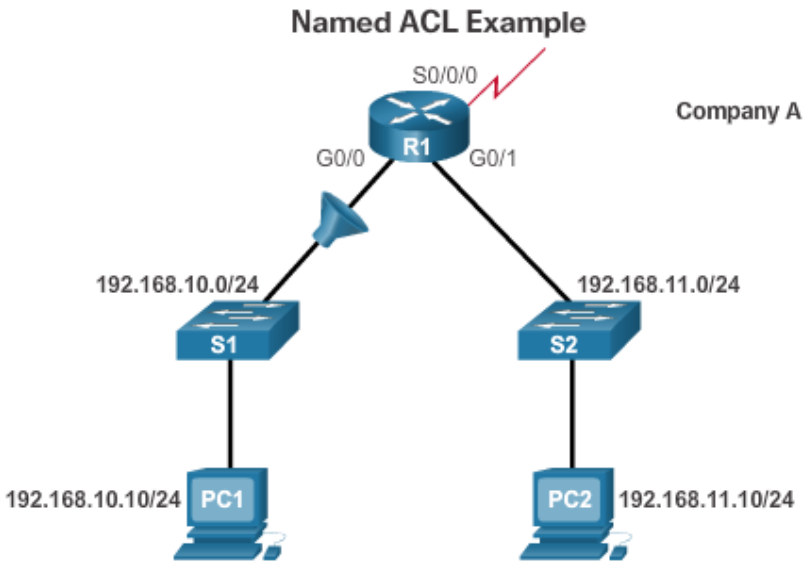
Alphanumeric name string must be unique and cannot begin with a number.

```
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.

2.4 Named Standard IPv4 ACL Syntax




```
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
```

2.5 Method 1 – Use a Text Editor

Editing Numbered ACLs Using a Text Editor

Configuration

```
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
```

Step 1

```
R1# show running-config | 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
```

Step 2

```
<Text editor>
access-list 1 deny host 192.168.10.10
access-list 1 permit 192.168.0.0 0.0.255.255
```

Step 3

```
R1# config 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
```

Step 4

```
R1# show running-config | 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
```

2.6 Method 2 – Use Sequence Numbers

Editing Numbered ACLs Using Sequence Numbers

Configuration

```
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
```

Step 1

```
R1# show access-lists 1
Standard IP access list 1
    10 deny   192.168.10.99
    20 permit 192.168.0.0, wildcard bits 0.0.255.255
R1#
```

Step 2

```
R1# conf t
R1(config)# ip access-list standard 1
R1(config-std-nacl)# no 10
R1(config-std-nacl)# 10 deny host 192.168.10.10
R1(config-std-nacl)# end
R1#
```

Step 3

```
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
R1#
```

2.7 Editing Standard Named ACLs

Adding a Line to a Named ACL

```
R1# show access-lists
Standard IP access list NO_ACCESS
    10 deny   192.168.11.10
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1# conf t
Enter configuration commands, one per line.  End with
CNTL/Z.
R1(config)# ip access-list standard NO_ACCESS
R1(config-std-nacl)# 15 deny host 192.168.11.11
R1(config-std-nacl)# end
R1# show access-lists
Standard IP access list NO_ACCESS
    10 deny   192.168.11.10
    15 deny   192.168.11.11
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1#
```

Note: The `no` *sequence-number* named-ACL command is used to delete individual statements.

2.8 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#
```

2.9 ACL Statistics

```
R1# show access-lists
Standard IP access list 1
    10 deny   192.168.10.10 (4 match(es))
    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 (4 match(es))
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1#
```

Output after pinging PC3 from PC1

Matches have been incremented.

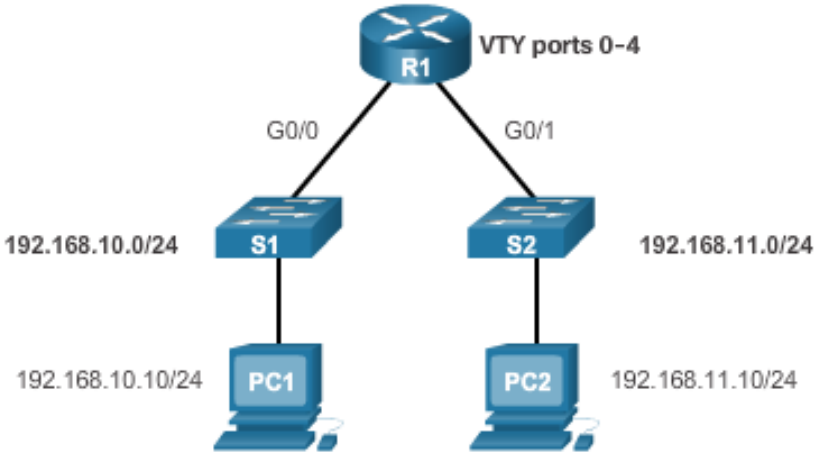
```
R1# show access-lists
Standard IP access list 1
    10 deny   192.168.10.10 (8 match(es))
    20 permit 192.168.0.0, wildcard bits 0.0.255.255
Standard IP access list NO_ACCESS
    15 deny   192.168.11.11
    10 deny   192.168.11.10 (4 match(es))
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1#
```

```
R1# show access-lists
Standard IP access list 1
    10 deny   192.168.10.10 (8 match(es))
    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 (4 match(es))
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1# clear access-list counters 1
R1#
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 (4 match(es))
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
```

Matches have been cleared.

2.10 The access-class Command

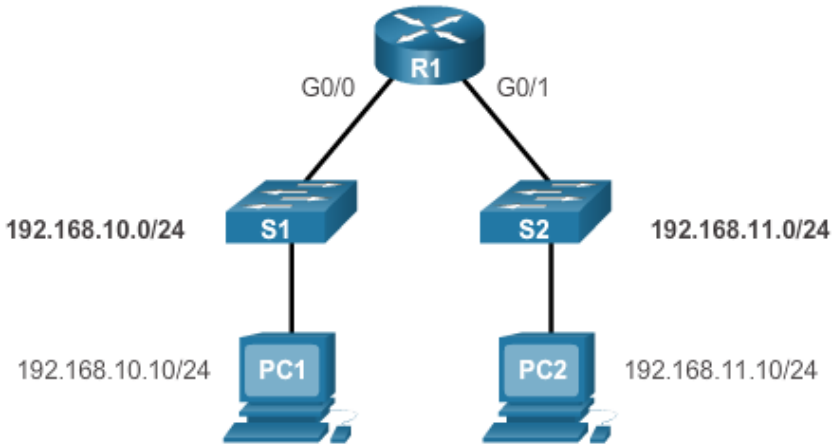
- The **access-class** command configured in line configuration mode restricts incoming and outgoing connections between a particular VTY (into a Cisco device) and the addresses in an access list.




```
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# access-class 21 in
R1(config-line)# exit
R1(config)# access-list 21 permit 192.168.10.0 0.0.0.255
R1(config)# access-list 21 deny any
```

2.11 Verifying the VTY Port is Secured

```
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>
```

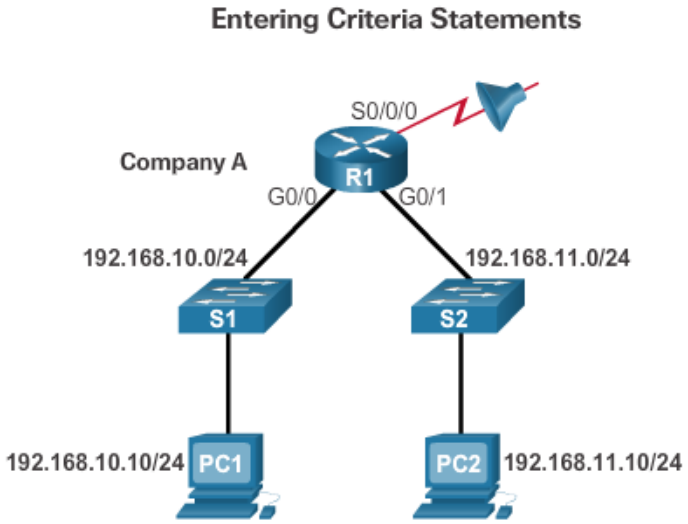
```
PC2>ssh 192.168.11.1
ssh connect to host 192.168.11.1 port
22: Connection refused

PC2>
```

3. Troubleshoot ACLs

3.1 The Implicit Deny Any

- At least one permit ACE must be configured in an ACL or all traffic is blocked.
- For the network in the figure, applying either ACL 1 or ACL 2 to the S0/0/0 interface of R1 in the outbound direction will have the same effect.



ACL 1

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

ACL 2

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

3.2 The Order of ACEs in an ACL

```
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 previous range statement.

```
R1(config)# access-list 4 permit host 192.168.10.10
R1(config)# access-list 4 deny 192.168.10.0 0.0.0.255
R1(config)#
```

ACL 4: Host statement can always be configured before range statements.

```
R1(config)# access-list 5 deny 192.168.10.0 0.0.0.255
R1(config)# access-list 5 permit host 192.168.11.10
R1(config)#
```

ACL 5: Host statement can be configured after range statement if there is no conflict.

3.3 Cisco IOS Reorders Standard ACLs

- Notice that the statements are listed in a different order than they were entered.

Sequencing Considerations During Configuration

```
R1(config)# access-list 1 deny 192.168.10.0 0.0.0.255
R1(config)# access-list 1 deny 192.168.20.0 0.0.0.255
R1(config)# access-list 1 deny 192.168.30.0 0.0.0.255
R1(config)# access-list 1 permit 10.0.0.1
R1(config)# access-list 1 permit 10.0.0.2
R1(config)# access-list 1 permit 10.0.0.3
R1(config)# access-list 1 permit 10.0.0.4
R1(config)# access-list 1 permit 10.0.0.5
R1(config)# end
R1# show running-config | include access-list 1
access-list 1 permit 10.0.0.2
access-list 1 permit 10.0.0.3
access-list 1 permit 10.0.0.1
access-list 1 permit 10.0.0.4
access-list 1 permit 10.0.0.5
access-list 1 deny   192.168.10.0 0.0.0.255
access-list 1 deny   192.168.20.0 0.0.0.255
access-list 1 deny   192.168.30.0 0.0.0.255
R1#
```

Range (network) statements

Host statements

3.3 Cisco IOS Reorders Standard ACLs

- The order in which the standard ACEs are listed is the sequence used by the IOS to process the list.

```
R1# show access-lists 1
Standard IP access list 1
 50 permit 10.0.0.2
 60 permit 10.0.0.3
 40 permit 10.0.0.1
 70 permit 10.0.0.4
 80 permit 10.0.0.5
 10 deny   192.168.10.0, wildcard bits 0.0.0.255
 20 deny   192.168.20.0, wildcard bits 0.0.0.255
 30 deny   192.168.30.0, wildcard bits 0.0.0.255
R1# copy running-config startup-config
R1# reload
R1# show access-lists 1
Standard IP access list 1
 10 permit 10.0.0.2
 20 permit 10.0.0.3
 30 permit 10.0.0.1
 40 permit 10.0.0.4
 50 permit 10.0.0.5
 60 deny   192.168.10.0, wildcard bits 0.0.0.255
 70 deny   192.168.20.0, wildcard bits 0.0.0.255
 80 deny   192.168.30.0, wildcard bits 0.0.0.255
R1#
```

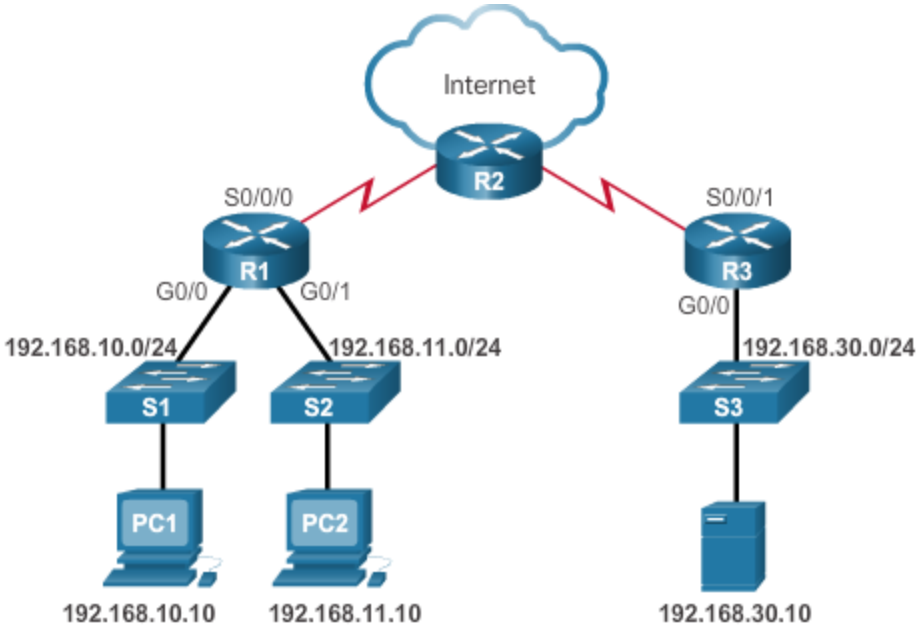
Host statements are listed first, in an order to be efficiently processed by the IOS.

Range statements are listed after host statements, in the order they were entered.

3.4 Routing Processes and ACLs

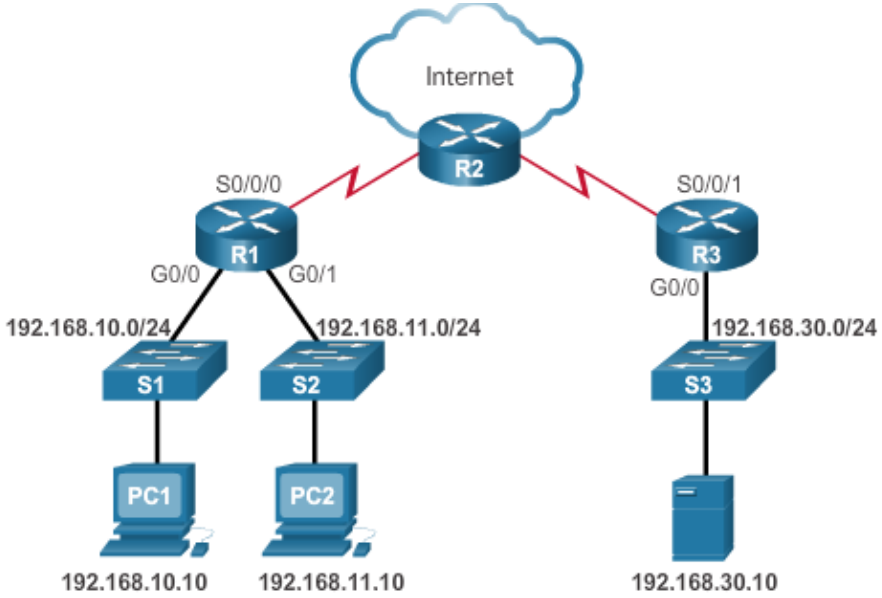
- As a frame enters an interface, the router checks to see whether the destination Layer 2 address matches its interface Layer 2 address, or whether the frame is a broadcast frame.
- If the frame address is accepted, the frame information is stripped off and the router checks for an ACL on the inbound 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.

3.5 Troubleshooting Standard IPv4 ACLs – Example 1




```
R3# show access-list
Standard IP access list 10
    10 deny   192.168.11.10
R3#
```

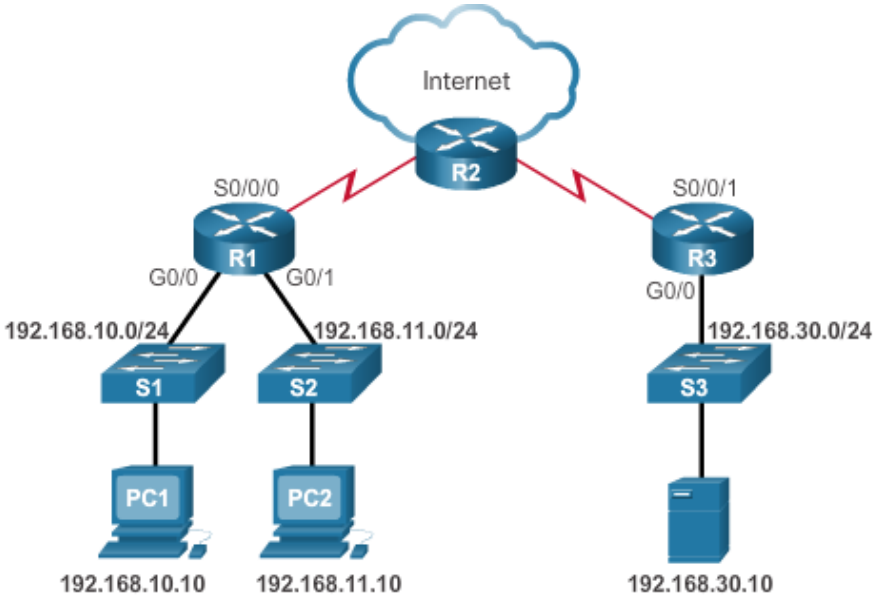
3.5 Troubleshooting Standard IPv4 ACLs – Example 1




```
R3(config)# access-list 10 permit any
R3(config)# end
R3# show access-list
Standard IP access list 10
    10 deny   192.168.11.10
    20 permit any (4 match(es))
R3#
```

3.6 Troubleshooting Standard IPv4 ACLs – Example 2

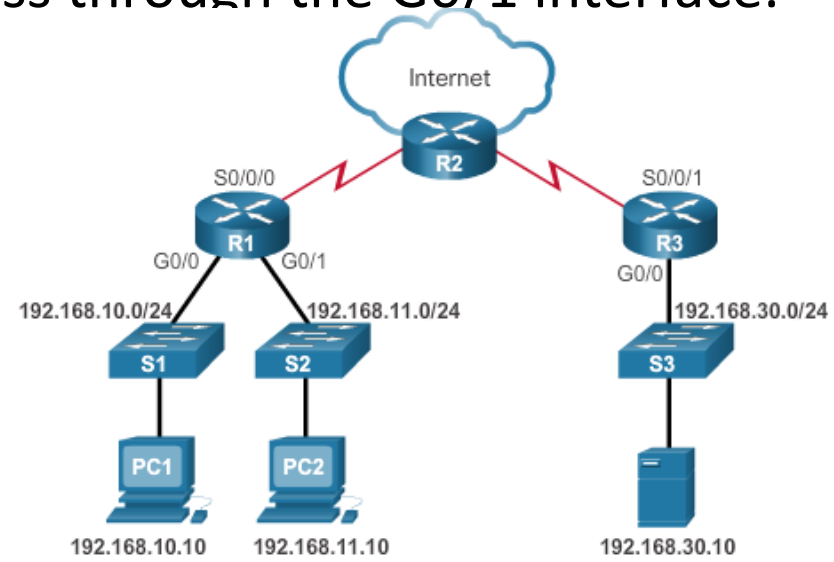
- Security Policy: The 192.168.11.0/24 network should not be able to access the 192.168.10.0/24 network.




```
R1# show access-list
Standard IP access list 20
    10 deny   192.168.11.0, wildcard bits 0.0.0.255 (8 match(es))
    20 permit any
```

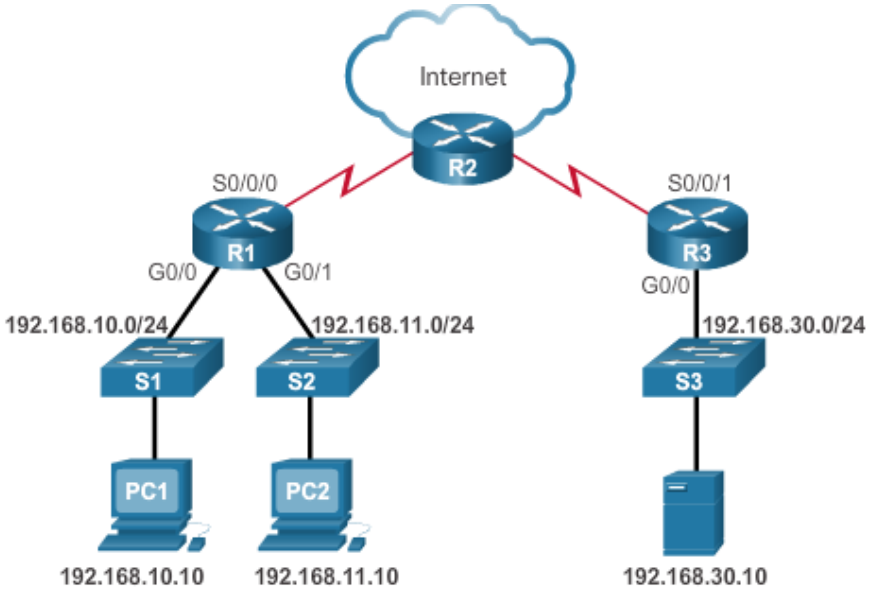
3.6 Troubleshooting Standard IPv4 ACLs – Example 2

- ACL 20 was applied to the wrong interface and in the wrong direction. All traffic from the 192.168.11.0/24 is denied inbound access through the G0/1 interface.




```
speed auto
interface GigabitEthernet0/1
ip address 192.168.11.1 255.255.255.0
ip access-group 20 in
duplex auto
speed auto
```

3.6 Troubleshooting Standard IPv4 ACLs – Example 2

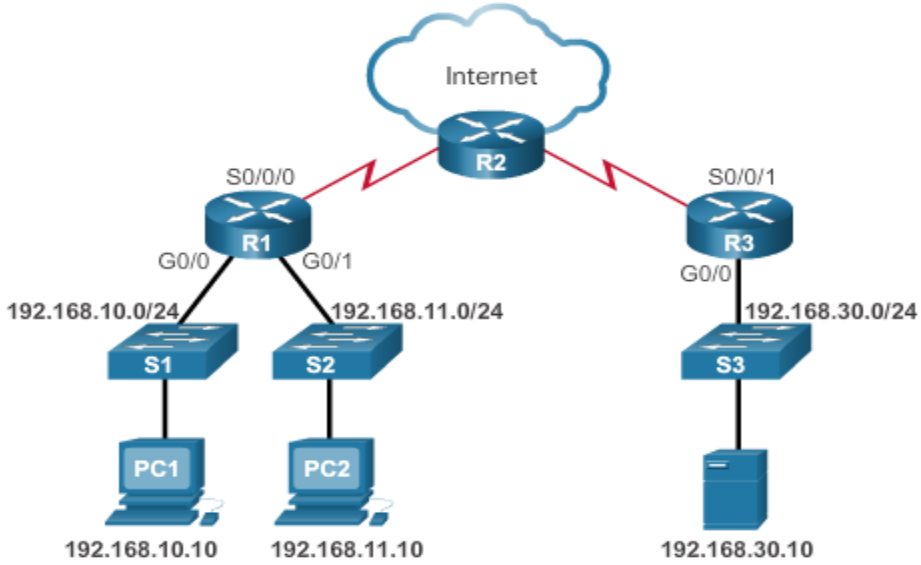



```
R1# config t
R1(config)# interface g0/1
R1(config-if)# no ip access-group 20 in
R1(config-if)# interface g0/0
R1(config-if)# ip access-group 20 out
```

3.7 Troubleshooting Standard IPv4 ACLs – Example 3

Problem

- **Security Policy**: Only PC1 is allowed SSH remote access to R1.

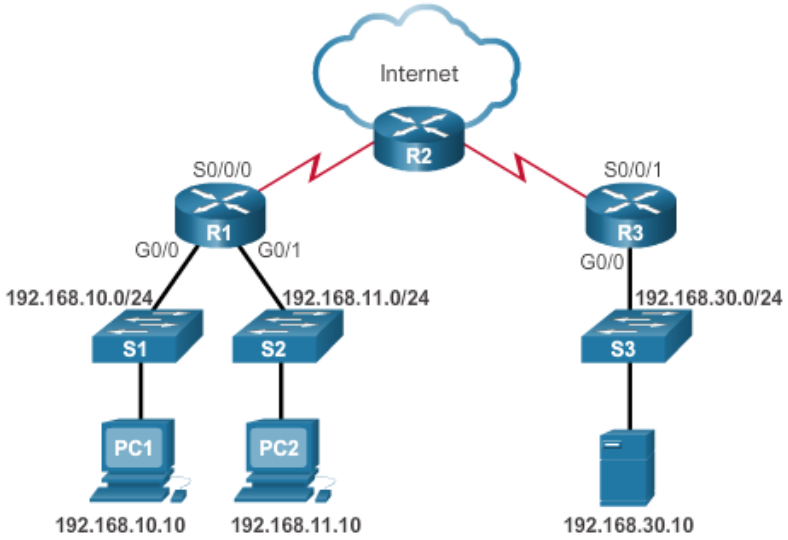



```
R1# show run | section line vty
line vty 0 4
 access-class PC1-SSH in
 login
 transport input ssh
R1# show access-list
Standard IP access list PC1-SSH
    10 permit 192.168.10.1
    20 deny   any (5 match(es))
R1#
```

3.7 Troubleshooting Standard IPv4 ACLs – Example 3

Solution!

- **Security Policy**: Only PC1 is allowed SSH remote access to R1.




```
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ip access-list standard PC1-SSH
R1(config-std-nacl)# no 10
R1(config-std-nacl)# 10 permit host 192.168.10.10
R1(config-std-nacl)# end
R1# clear access-list counters
```

```
R1# show access-list
Standard IP access list PC1-SSH
    10 permit 192.168.10.10 (2 match(es))
    20 deny   any
R1#
```

Chapter Summary

Summary

- Explain how ACLs filter traffic.
- Explain how ACLs use wildcard masks.
- Explain how to create ACLs.
- Explain how to place ACLs.
- Configure standard IPv4 ACLs to filter traffic to meet networking requirements.
- Use sequence numbers to edit existing standard IPv4 ACLs.
- Configure a standard ACL to secure vty access.
- Explain how a router processes packets when an ACL is applied.
- Troubleshoot common standard IPv4 ACL errors using CLI commands.

TERIMA KASIH