Chapter 7: Access Control Lists

The password used in the Packet Tracer activities in this chapter is: **PT_ccna5**

Access router configuration

```
router ospf 100
 redistribute connected metric 100 metric-type 1 subnets
 redistribute static metric 100 metric-type 1 subnets
 passive-interface Ethernet0
 network 10.0.144.0 0.0.0.255 area 9
 area 9 nssa
!
no ip classless
logging buffered alerts
logging console informational
logging 10.192.17.3
access-list 1 permit 10.132.36.16
access-list 1 permit 10.132.37.11
access-list 1 permit 10.132.37.3
access-list 2 permit 10.0.0.0 0.255.255.255
snmp-server community public RW 1
snmp-server trap-source Loopback1
snmp-server packetsize 8192
snmp-server trap-authentication
```

Access router configuration

```
router ospf 100
 redistribute connected metric 100 metric-type 1 subnets
 redistribute static metric 100 metric-type 1 subnets
 passive-interface Ethernet0
 network 10.0.144.0 0.0.0.255 area 9
 area 9 nssa
!
no ip classless
logging buffered alerts
logging console informational
logging 10.192.17.3
access-list 1 permit 10.132.36.16
access-list 1 permit 10.132.37.11
access-list 1 permit 10.132.37.3
access-list 2 permit 10.0.0.0 0.255.255.255
snmp-server community public RW 1
snmp-server trap-source Loopback1
snmp-server packetsize 8192
snmp-server trap-authentication
```

útok pomocí snmpbrute

Backbone router configuration

```
version 11.2
no service finger
service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname Prague_General_Staff
!
boot system flash slot0:rsp-isv-mz_112-17_P.bin
aaa new-model
aaa authentication login default tacacs+ local
aaa authorization exec tacacs+ local
aaa authorization commands 1 tacacs+ local
aaa authorization commands 15 tacacs+ local
aaa accounting exec start-stop tacacs+
aaa accounting commands 0 start-stop tacacs+
:
aaa accounting commands 15 start-stop tacacs+
aaa accounting system start-stop tacacs+
enable password 7 121A0C041B04
!
```

Backbone router configuration

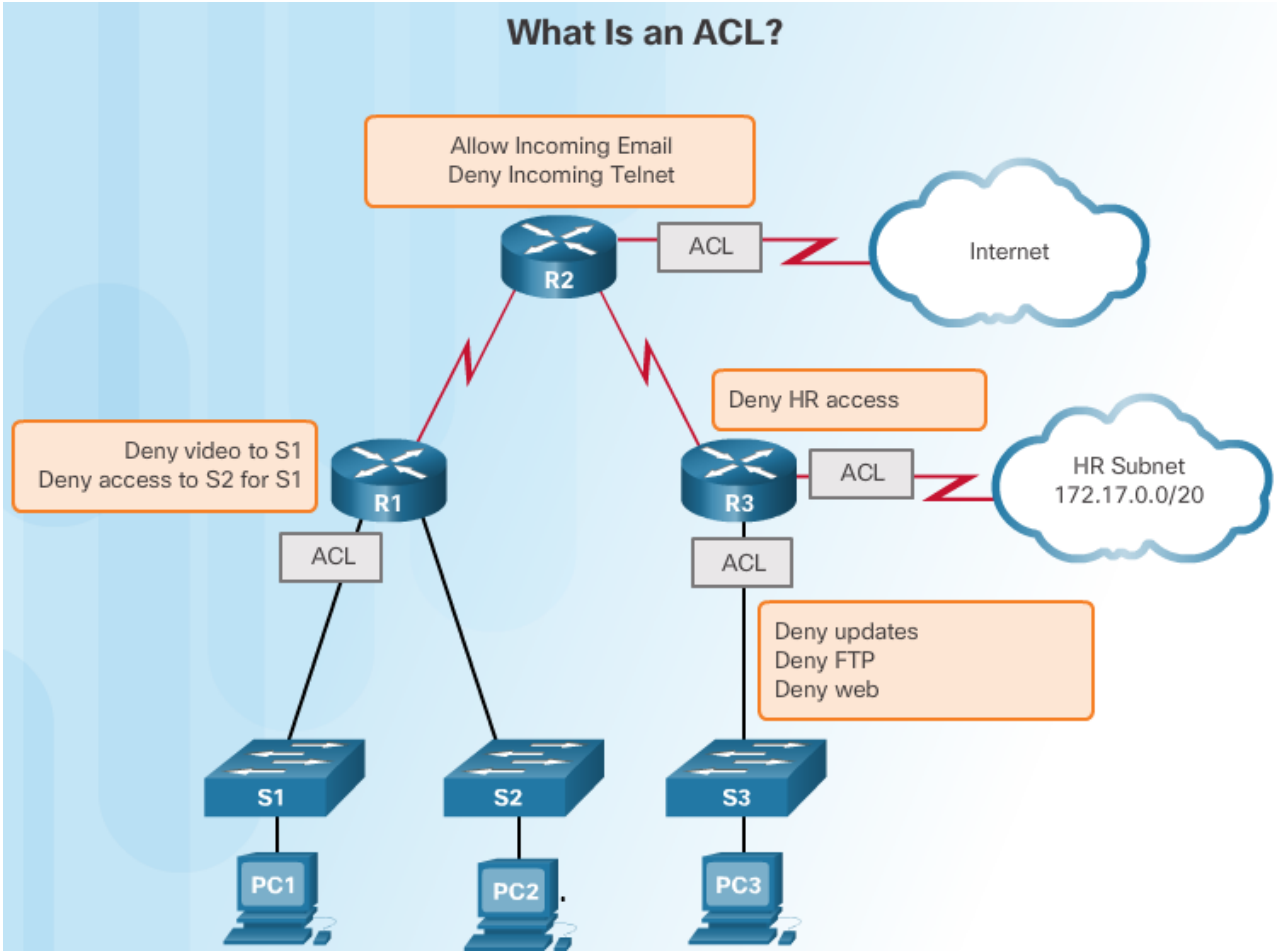
```
version 11.2
no service finger
service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname Prague_Generalni_stab
!
boot system flash slot0:rsp-isv-mz_112-17_P.bin
aaa new-model
aaa authentication login default tacacs+ local
aaa authorization exec tacacs+ local
aaa authorization commands 1 tacacs+ local
aaa authorization commands 15 tacacs+ local
aaa accounting exec start-stop tacacs+
aaa accounting commands 0 start-stop tacacs+
:
aaa accounting commands 15 start-stop tacacs+
aaa accounting system start-stop tacacs+
enable password 7 121A0C041B04
!
```

7.1 ACL Operation

7.1 ACL Operation

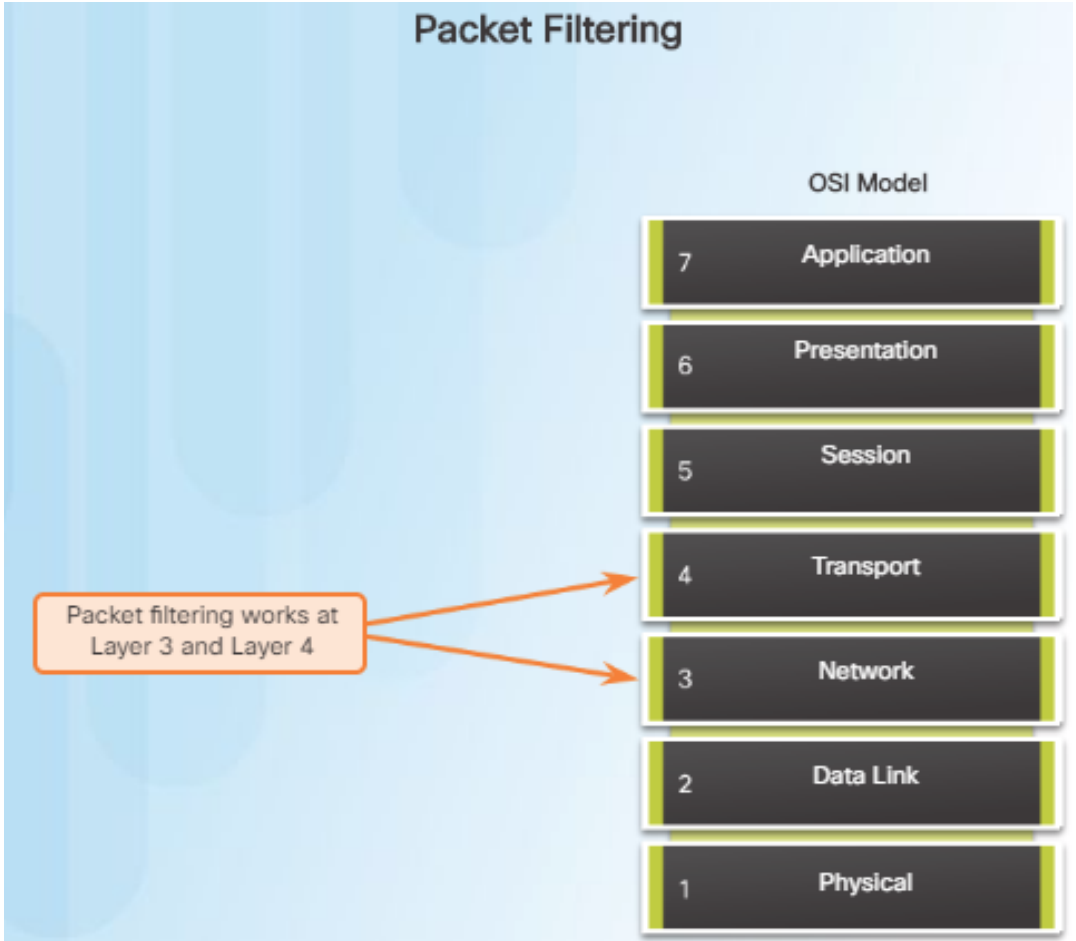
- how ACLs filter traffic
- how ACLs use wildcard masks
- how to place ACLs

What is an ACL?



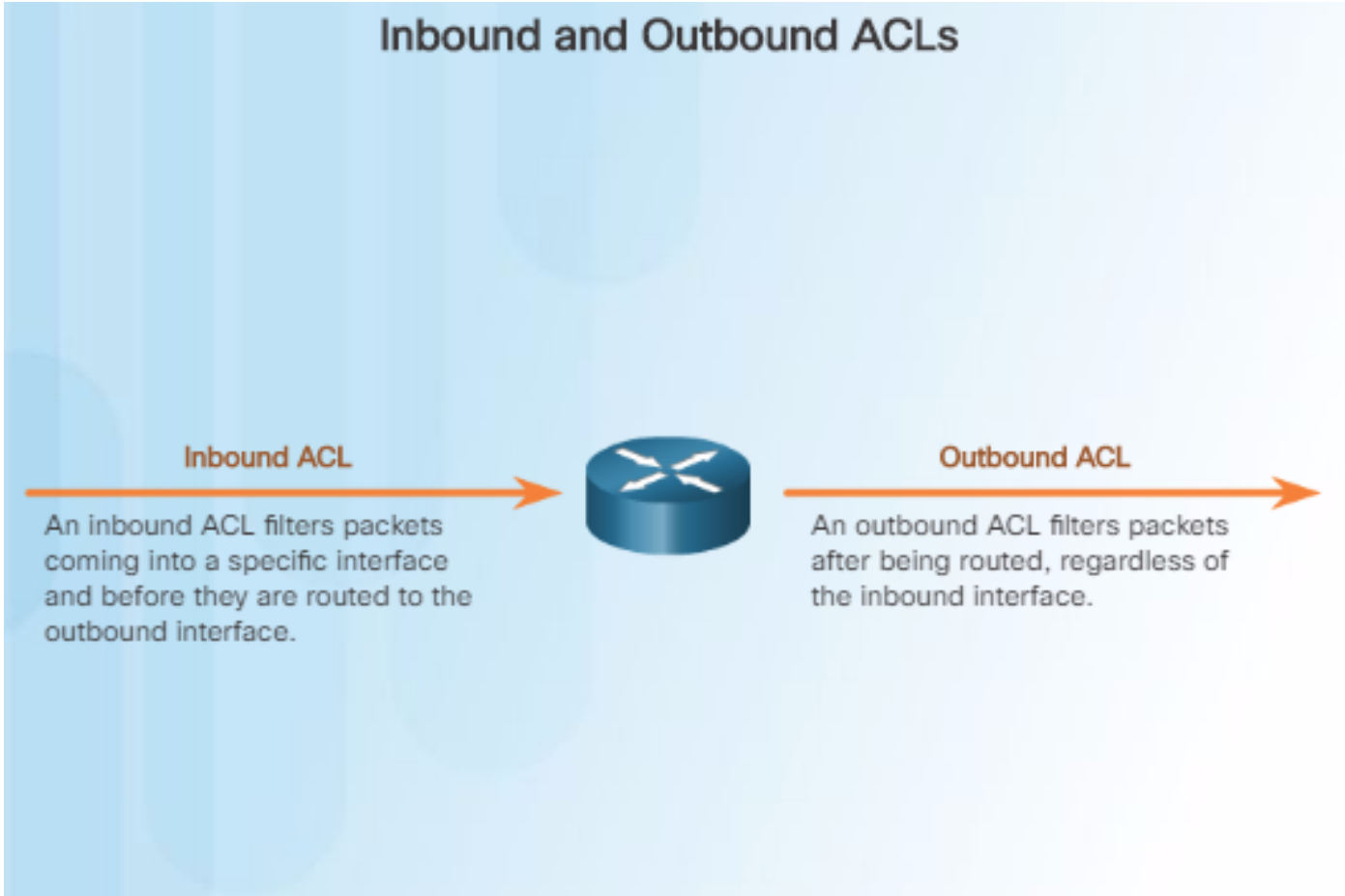
- An ACL is a series of IOS commands that control whether a router forwards or drops packets.
- ACLs are not configured by default.

Purpose of ACLs: Packet Filtering



- An ACL is a sequential list of permit or deny statements (ACEs).
- Packet Filtering:
  - incoming/outgoing packets.
  - Layer 3 or Layer 4.
- The last statement: implicit deny => at least one permit statement.

ACL Operation



- ACLs do not act on packets that originate from the router itself.
- ACLs:
  - Inbound ACLs.
  - Outbound ACLs.

Wildcard = inverse mask

	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

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

- Example 1: all.
- Example2: nothing.
- Example 3: all from 192.168.1.0/24.

Wildcard Mask Calculation

Example 1

```
  255.255.255.255
- 255.255.255.000
-----------------
              255
```

Example 2

```
  255.255.255.255
- 255.255.255.240
-----------------
               15
```

Example 3

```
  255.255.255.255
- 255.255.254.000
-----------------
            1.255
```

Wildcard Mask Keywords

192.168.10.10 0.0.0.0 = host 192.168.10.10

0.0.0.0 255.255.255.255 = any

The any and host Keywords

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.

ACL Traffic Filtering on a Router

One list per interface, per direction, and per protocol

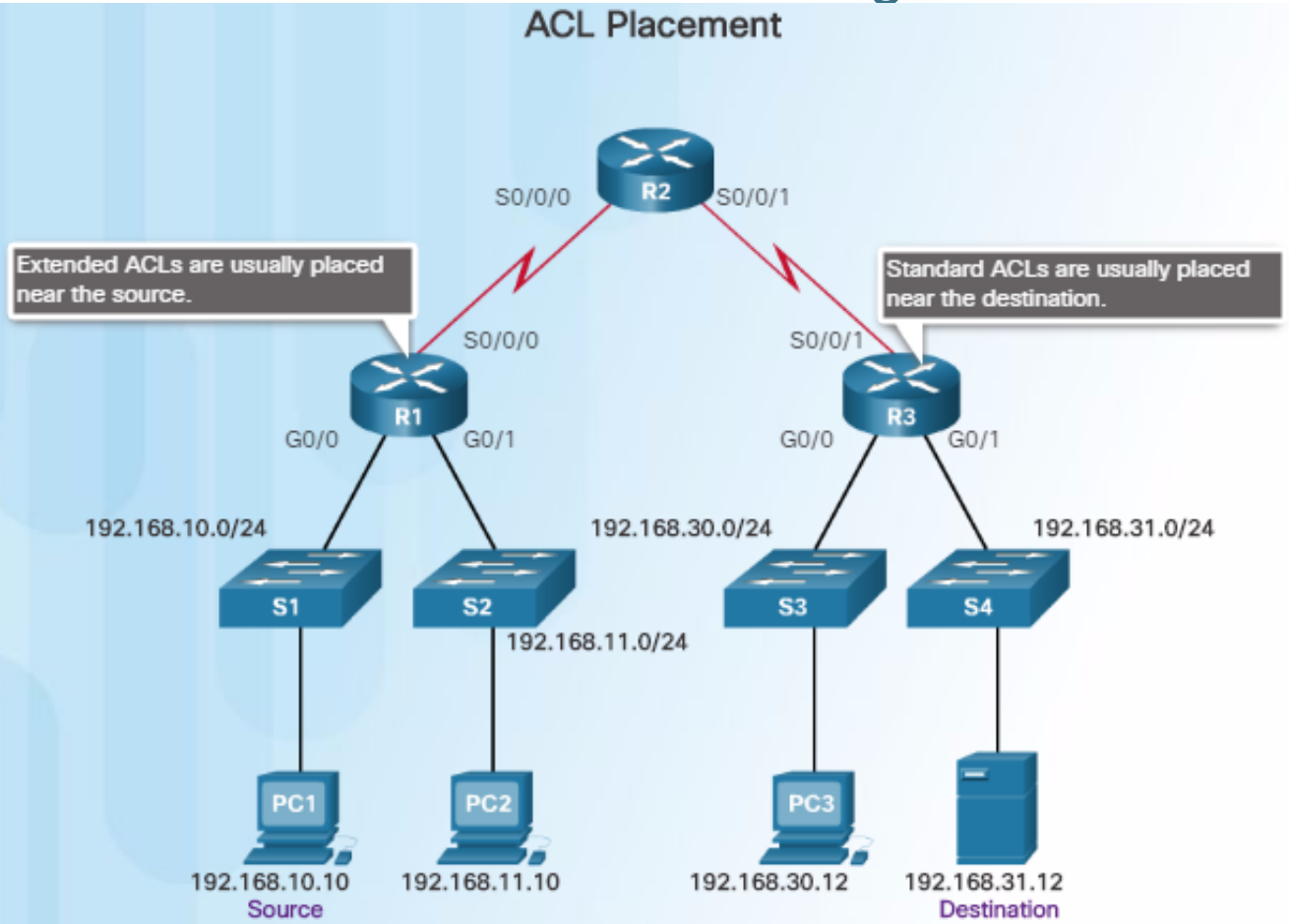
With two interfaces and two protocols running, this router could have a total of 8 separate ACLs applied.

The Rules for Applying ACLs

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

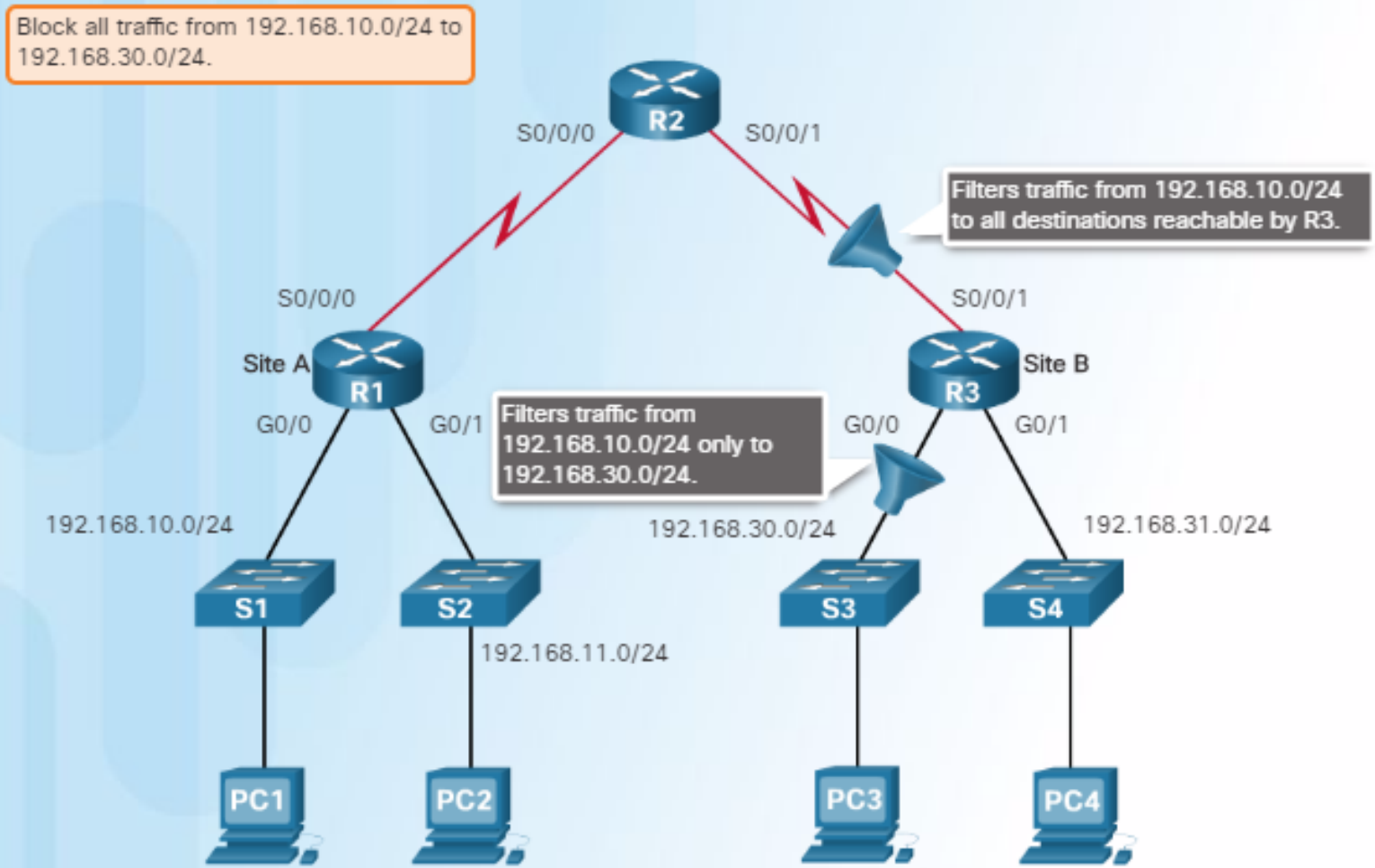
- One ACL per protocol (e.g., IPv4 or IPv6)
- One ACL per direction (i.e., IN or OUT)
- One ACL per interface (e.g., GigabitEthernet0/0)

General Guidelines for Creating ACLs



Standard ACLs – Since standard ACLs do not specify destination addresses, they should be configured as close to the destination as possible.

Standard ACL Placement



7.2 Standard IPv4 ACLs

Chapter 7.2 - ACL Operation

- Configure standard IPv4 ACLs to filter traffic in a SMB network.

- Configure a standard ACL to secure VTY access.

Numbered Standard IPv4 ACL Syntax

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

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.

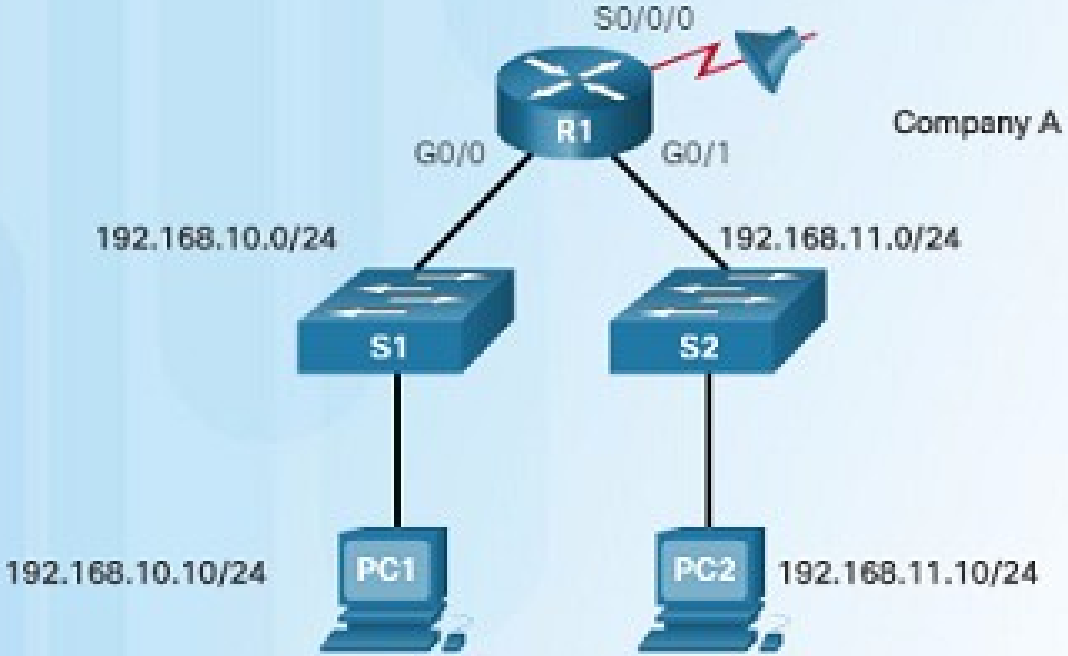
```
R1(config)# interface serial 0/0/0
```

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
```

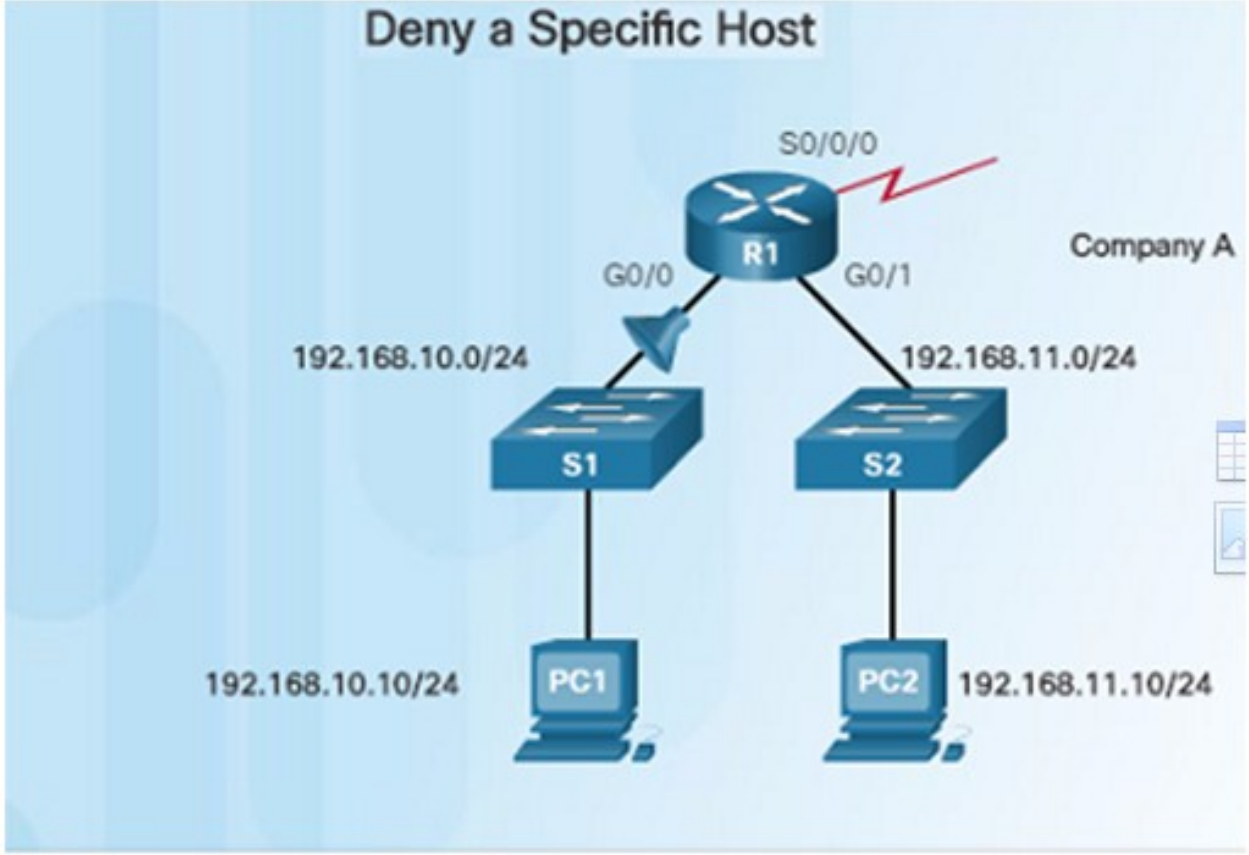
This example activates the standard IPv4 ACL 1 on the interface as an outbound filter.

Deny a Specific Host and Permit a Specific Subnet



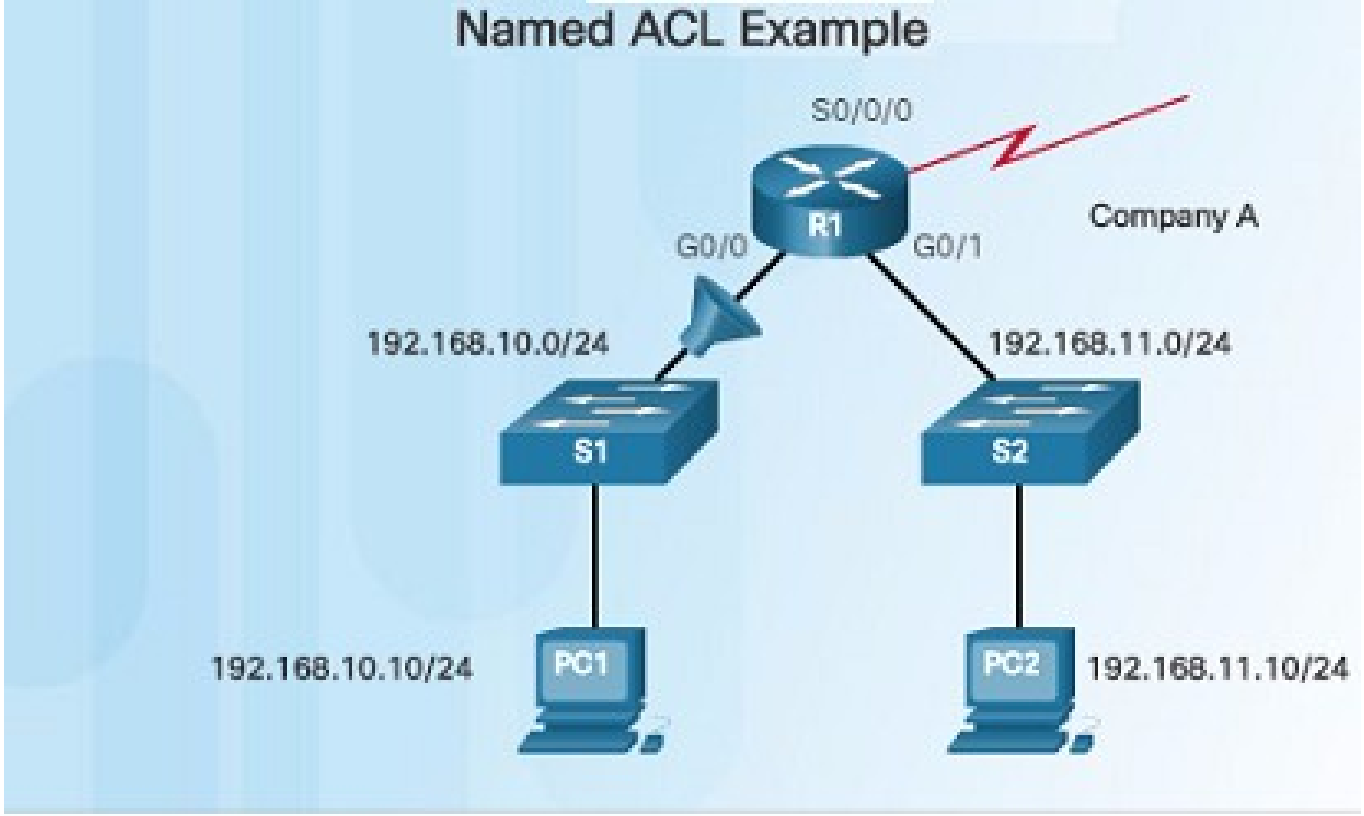

```
R1(config)# no access-list 1
R1(config)# access-list 1 deny host 192.168.10.10
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)# interface s0/0/0
R1(config-if)# ip access-group 1 out
```

Deny a Specific Host



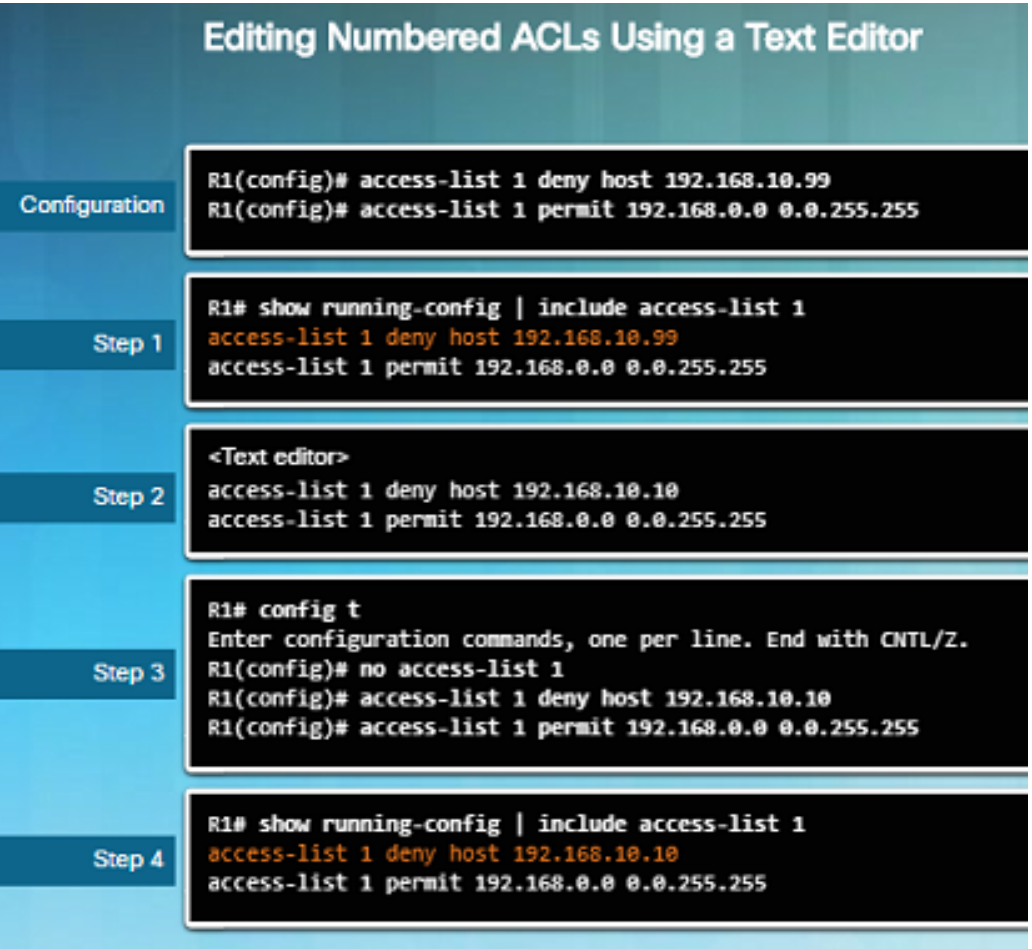

```
R1(config)# no access-list 1
R1(config)# access-list 1 deny host 192.168.10.10
R1(config)# access-list 1 permit any
R1(config)# interface g0/0
R1(config-if)# ip access-group 1 in
```

Named ACL Example



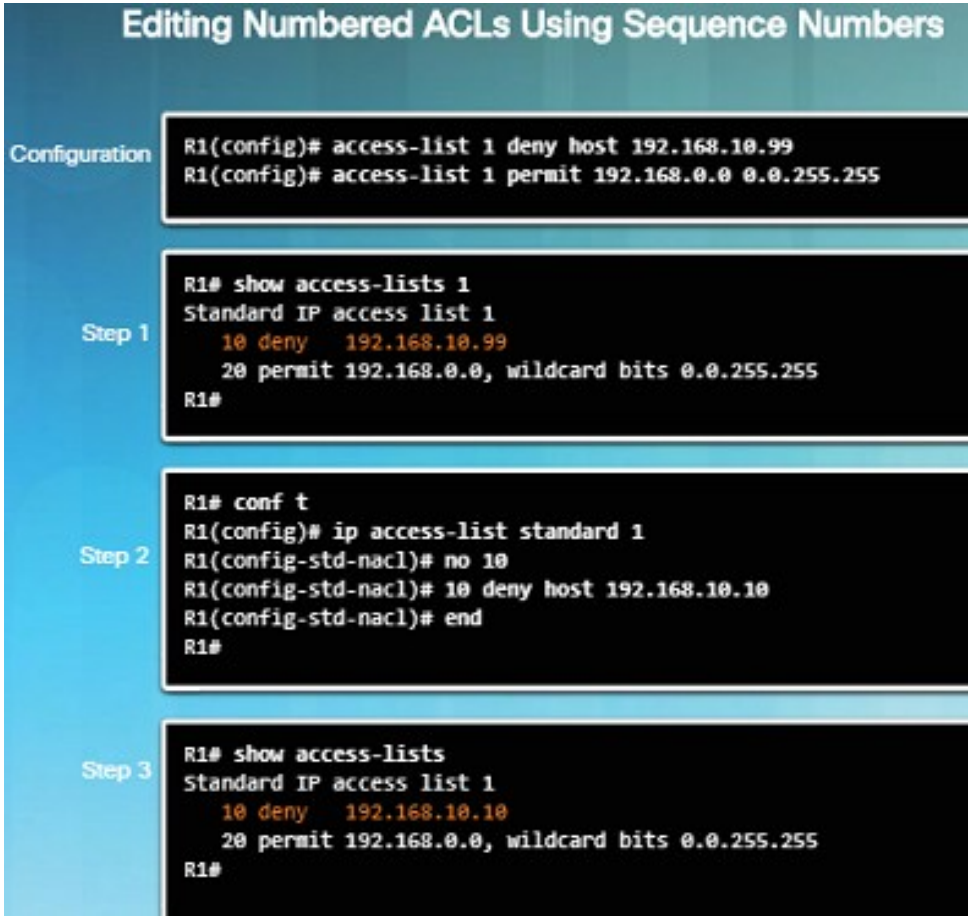

```
R1(config)# ip access-list standard NO_ACCESS
R1(config-std-nacl)# deny host 192.168.11.10
R1(config-std-nacl)# permit any
R1(config-std-nacl)# exit
R1(config)# interface g0/0
R1(config-if)# ip access-group NO_ACCESS out
```

Method 1 – Use a Text Editor



use the **show running-config** command to display the ACL

Method 2 – Use Sequence Numbers



- The **deny 192.168.10.99** statement is incorrect. The host to deny should be 192.168.10.10
- The misconfigured statement had to be deleted with the no command: **no 10**
- new statement with the correct host was added: **10 deny host 192.168.10.10**

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#
```

- Use the **show ip interface** command to verify that the ACL is applied to the correct interface.
- The output will display the name of the access list and the direction in which it was applied to the interface.
- Use the **show access-lists** command to display the access-lists configured on the router.
- Notice how the sequence is displayed out of order for the NO_ACCESS access list. This will be discussed later in this section.

ACL Statistics

```
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.

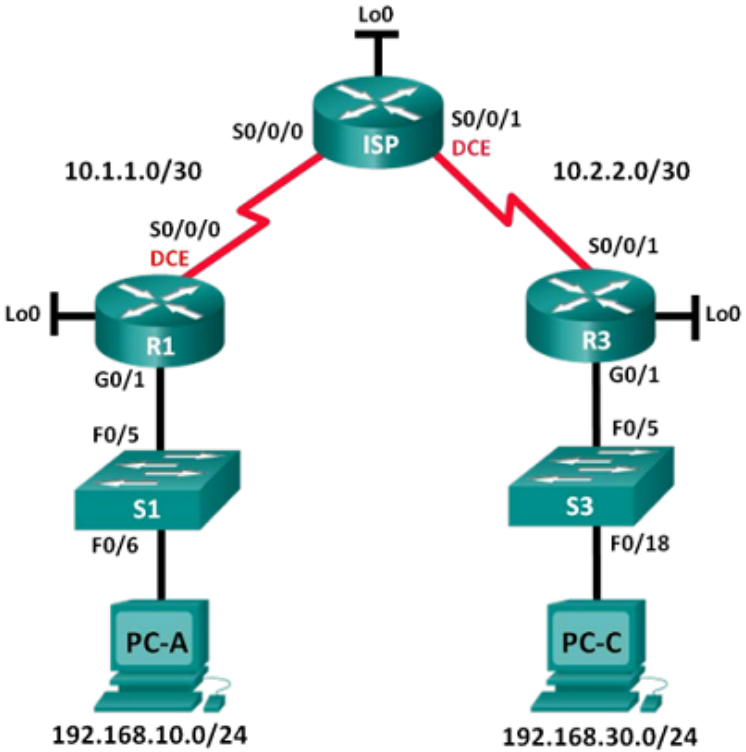
- **show access-lists**
- **clear access-list counters**

Recall that every ACL has an implicit **deny any** as the last statement. The statistics for this implicit command will not be displayed. However, if this command is configured manually, the results will be displayed.

Lab – Configuring and Modifying Standard IPv4 ACLs

Lab – Configuring and Verifying Standard IPv4 ACLs

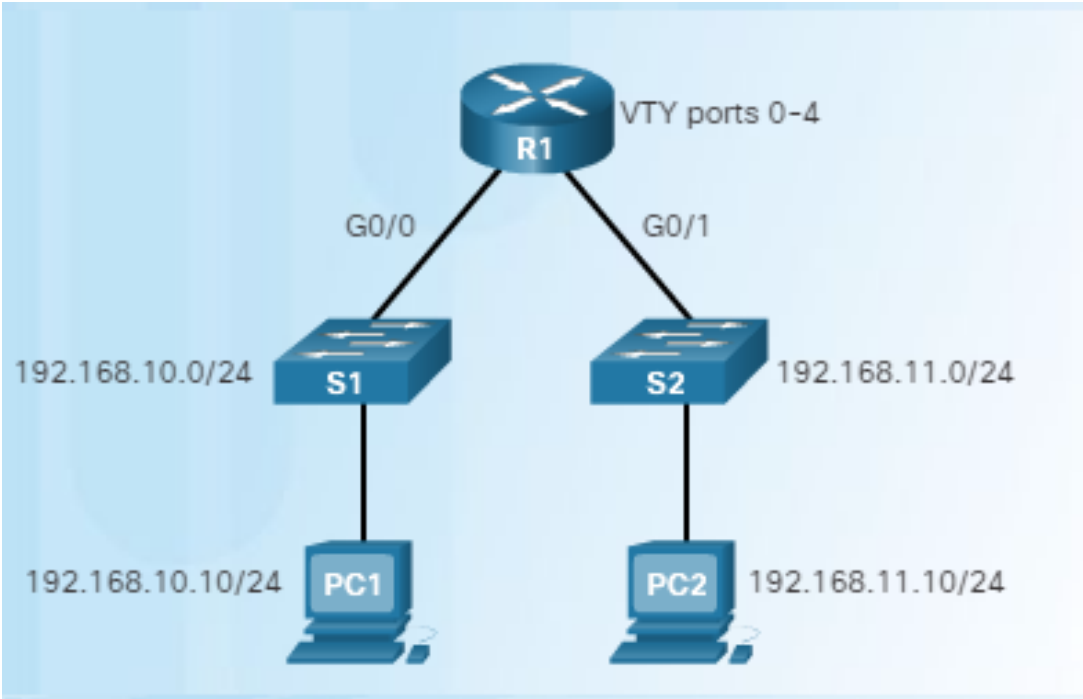
Topology



- ACL that allows traffic from all hosts on the 192.168.10.0/24 network and all hosts on the 192.168.20.0/24 network to access all hosts on the 192.168.30.0/24 network.

- Napište na tabuli!

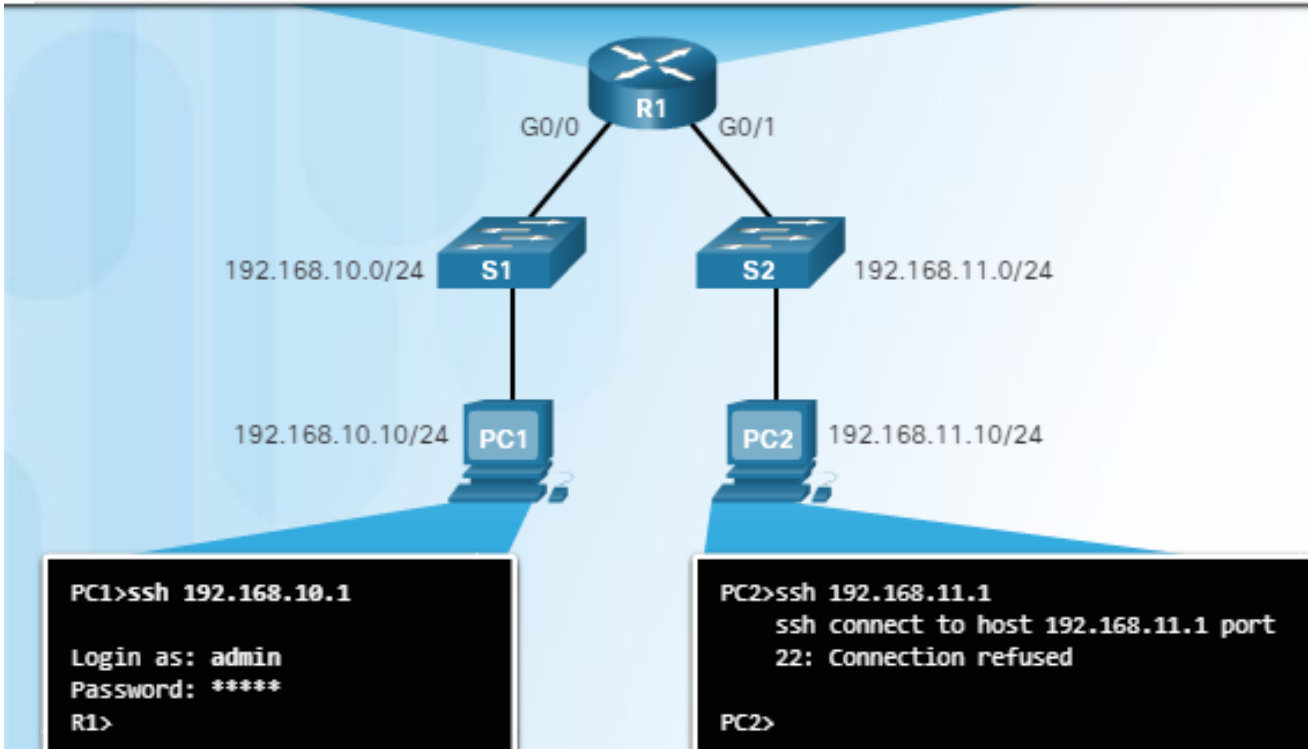
The access-class Command




```
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
```

Verifying the VTY Port is Secured

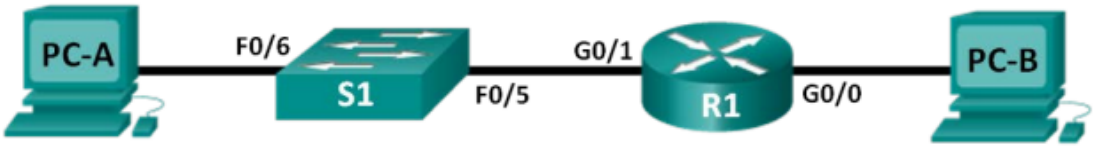
```
R1# show access-lists
Standard IP access list 21
    10 permit 192.168.10.0, wildcard bits 0.0.0.255 (2 matches)
    20 deny any (1 match)
R1#
```



Lab – Configuring and Verifying VTY Restrictions

Lab – Configuring and Verifying VTY Restrictions

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	192.168.0.1	255.255.255.0	N/A
	G0/1	192.168.1.1	255.255.255.0	N/A
S1	VLAN 1	192.168.1.2	255.255.255.0	192.168.1.1
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-B	NIC	192.168.0.3	255.255.255.0	192.168.0.1

Objectives

Part 1: Configure Basic Device Settings

Part 2: Configure and Apply the Access Control List on R1

Part 3: Verify the Access Control List Using Telnet

Part 4: Challenge - Configure and Apply the Access Control List on S1

only administrator PCs have permission to telnet or SSH into the router.

Příklad reálného filtru

```
access-list 100 deny   ip     15.2.6.0  0.0.0.255           any log
access-list 100 deny   ip     host 15.1.1.20  host 15.1.1.20    log
access-list 100 deny   ip     127.0.0.0 0.255.255.255       any log
access-list 100 deny   ip     10.0.0.0  0.255.255.255       any log
access-list 100 deny   ip     0.0.0.0   0.255.255.255       any log
access-list 100 deny   ip     172.16.0.0 0.15.255.255       any log
access-list 100 deny   ip     192.168.0.0 0.0.255.255       any log
access-list 100 deny   ip     192.0.2.0 0.0.0.255           any log
access-list 100 deny   ip     169.254.0.0 0.0.255.255       any log
access-list 100 deny   ip     224.0.0.0 15.255.255.255      any log
access-list 100 deny   ip     any host 15.2.6.255               log
access-list 100 deny   ip     any host 15.2.6.0  log
access-list 100 permit tcp    any 15.2.6.0 0.0.0.255 established
access-list 100 deny   icmp   any any echo log
access-list 100 deny   icmp   any any redirect log
access-list 100 deny   icmp   any any mask-request log
access-list 100 permit icmp   any 15.2.6.0  0.0.0.255
access-list 100 permit ospf   15.1.0.0  0.0.255.255 host 14.1.1.20
access-list 100 deny   tcp    any any range 6000 6063 log
access-list 100 deny   tcp    any any eq 6667 log
access-list 100 deny   tcp    any any range 12345 12346 log
access-list 100 deny   tcp    any any eq 31337 log
access-list 100 permit tcp    any eq 20 15.2.6.0  0.0.0.255 gt 1023
access-list 100 deny   udp    any any eq 2049 log
access-list 100 deny   udp    any any eq 31337 log
access-list 100 deny   udp    any any range 33400 34400 log
access-list 100 permit udp    any eq 53 15.2.6.0  0.0.0.255 gt 1023
access-list 100 deny   tcp    any range 0 65525 any range 0 65535 log
access-list 100 deny   udp    any range 0 65525 any range 0 65535 log
access-list 100 deny   ip     any any log
```

Příklady na závěr

R1(config)#**access-list 10 deny 172.16.16.0 0.0.3.255**

R2(config)#**access-list 10 deny 172.16.16.0 0.0.7.255**

R3(config)#**access-list 10 deny 172.16.32.0 0.0.15.255**

R4(config)#**access-list 10 deny 172.16.64.0 0.0.63.255**

R5(config)#**access-list 10 deny 192.168.160.0 0.0.31.255**


CISCO