

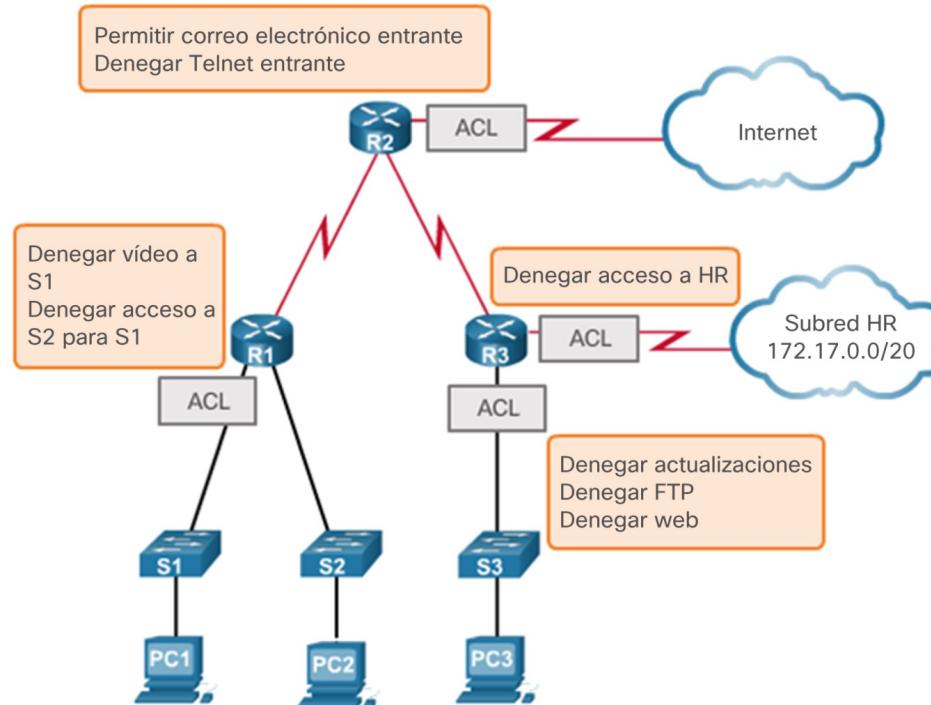
# Listas de control de acceso



# Funcionamiento de una ACL

# ¿Qué es una ACL?

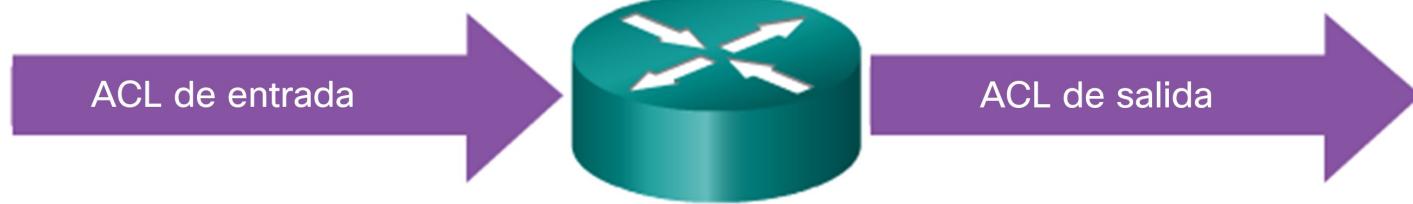
- Los routers no tienen listas ACL configuradas de manera predeterminada, por lo que no filtran el tráfico de manera predeterminada.



# Filtrado de paquetes

- El filtrado de paquetes, a veces denominado “filtrado de paquetes estático”, controla el acceso a una red mediante el análisis de los paquetes entrantes y salientes y la transferencia o el descarte de estos según determinados criterios, como la dirección IP de origen, la dirección IP de destino y el protocolo incluido en el paquete.
- Cuando reenvía o deniega los paquetes según las reglas de filtrado, un router funciona como filtro de paquetes.
- Una ACL es una lista secuencial de instrucciones permit (permitir) o deny (denegar), conocidas como “entradas de control de acceso” (ACE).

# Funcionamiento de una ACL



Las ACL de entrada filtran los paquetes que ingresan a una interfaz específica y lo hacen antes de que se enrutan a la interfaz de salida.

Las ACL de salida filtran los paquetes después de que se enrutan, independientemente de la interfaz de entrada.

## Máscaras de comodín en listas ACL

# Introducción a las máscaras de comodín en listas ACL

### Máscaras de wildcard

Posición del bit de octeto y valor de dirección para el bit

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |   |
|-----|----|----|----|---|---|---|---|---|
| 0   | 0  | 0  | 0  | 0 | 0 | 0 | 0 | = Hacer coincidir todos los bits de dirección (coincidir todos) |
| 0   | 0  | 1  | 1  | 1 | 1 | 1 | 1 | = Ignorar los últimos 6 bits de dirección                       |
| 0   | 0  | 0  | 0  | 1 | 1 | 1 | 1 | = Ignorar los últimos 4 bits de dirección                       |
| 1   | 1  | 1  | 1  | 1 | 1 | 1 | 0 | = Ignorar los primeros 6 bits de dirección                      |
| 1   | 1  | 1  | 1  | 1 | 1 | 1 | 1 | = Omitir todos los bits del octeto                              |

#### Ejemplos

0 significa hacer coincidir el valor del bit de dirección correspondiente  
1 significa ignorar el valor del bit de dirección correspondiente

## Máscaras de comodín en listas ACL

# Introducción a las máscaras de comodín en listas ACL (continuación)

### Ejemplo

|                            | <b>Dirección decimal</b> | <b>Dirección binaria</b>            |
|----------------------------|--------------------------|-------------------------------------|
| Dirección IP para procesar | 192.168.10.0             | 11000000.10101000.00001010.00000000 |
| Máscara de comodín         | 0.0.255.255              | 00000000.00000000.11111111.11111111 |
| Dirección IP resultante    | 192.168.0.0              | 11000000.10101000.00000000.00000000 |

# Ejemplos de máscaras de comodín

### Máscaras de comodín para establecer coincidencias con hosts y subredes IPv4

Ejemplo 1

|                    | Decimal     | Binario                             |
|--------------------|-------------|-------------------------------------|
| Dirección IP       | 192.168.1.1 | 11000000.10101000.00000001.00000001 |
| Máscara de comodín | 0.0.0.0     | 00000000.00000000.00000000.00000000 |
| Resultado          | 192.168.1.1 | 11000000.10101000.00000001.00000001 |

Ejemplo 2

|                    | Decimal         | Binario                             |
|--------------------|-----------------|-------------------------------------|
| Dirección IP       | 192.168.1.1     | 11000000.10101000.00000001.00000001 |
| Máscara de comodín | 255.255.255.255 | 11111111.11111111.11111111.11111111 |
| Resultado          | 0.0.0.0         | 00000000.00000000.00000000.00000000 |

Ejemplo 3

|                    | Decimal     | Binario                             |
|--------------------|-------------|-------------------------------------|
| Dirección IP       | 192.168.1.1 | 11000000.10101000.00000001.00000001 |
| Máscara de comodín | 0.0.0.255   | 00000000.00000000.00000000.11111111 |
| Resultado          | 192.168.1.0 | 11000000.10101000.00000001.00000000 |

## Máscaras de comodín en listas ACL

# Ejemplos de máscaras de comodín (continuación)

### Máscaras de comodín para establecer coincidencias con rangos

Ejemplo 1

|                     | Decimal                             | Binario   |
|---------------------|-------------------------------------|---|
| Dirección IP        | 192.168.16.0                        | 11000000.10101000.00010000.00000000   |
| Máscara de comodín  | 0.0.15.255                          | 00000000.00000000.00001111.11111111   |
| Rango de resultados | 192.168.16.0<br>a<br>192.168.31.255 | 11000000.10101000.00010000.00000000<br>a<br>11000000.10101000.00011111.11111111 |

Ejemplo 2

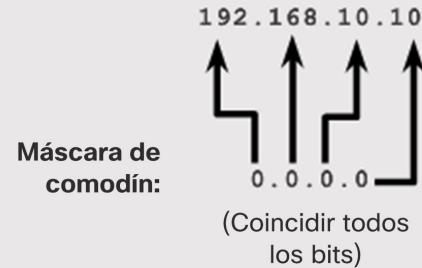
|                     | Decimal   | Binario   |
|---------------------|---|---|
| Dirección IP        | 192.168.1.0   | 11000000.10101000.00000001.00000000   |
| Máscara de comodín  | 0.0.254.255   | 00000000.00000000.11111110.11111111   |
| Rango de resultados | 192.168.1.0<br><br>Todas las subredes con número impar en la red principal<br>192.168.0.0 | 11000000.10101000.00000001.00000000<br><br>Todas las subredes con número impar en la red principal<br>192.168.0.0 |

# Palabras clave de una máscara de comodín

## Abreviaturas de la máscara de bits de comodín

### Ejemplo 1

- 192.168.10.10 0.0.0.0 coincide con todos los bits de la dirección.
- Abrevie esta máscara de comodín utilizando la dirección IP precedida por la palabra clave host (host 192.168.10.10).



### Ejemplo 2

- 0.0.0.0 255.255.255.255 omite todos los bits de la dirección.
- Abrevie la expresión con la palabra clave any.



## Ejemplos de palabras clave de una máscara de comodín

### Ejemplo 1:

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

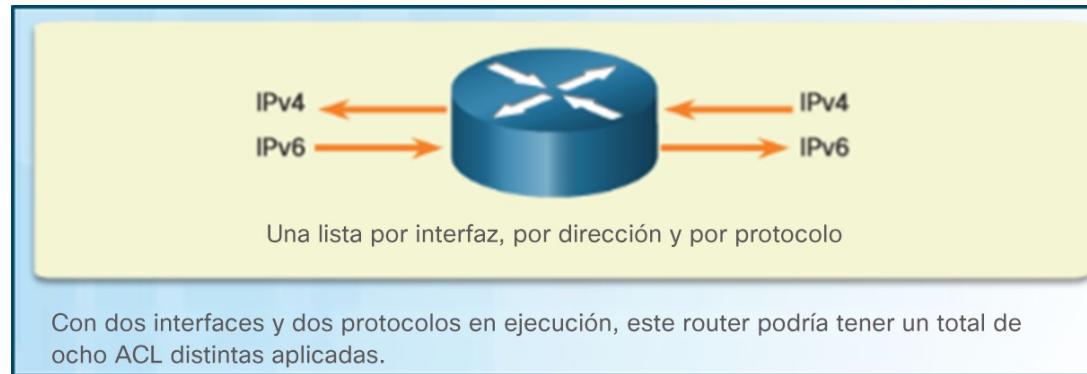
### Ejemplo 2:

```
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
```

Este es el formato de las palabras clave opcionales any y host en una sentencia ACL.

# Descripción general de la operación de la ACL

- Puede configurar lo siguiente:
  - **Una ACL por protocolo:** para controlar el flujo de tráfico en una interfaz, se debe definir una ACL para cada protocolo habilitado en la interfaz.
  - **Una ACL por sentido:** las ACL controlan el tráfico en una interfaz de a un sentido por vez. Se deben crear dos ACL diferentes para controlar el tráfico entrante y saliente.
  - **Una ACL por interfaz:** las ACL controlan el tráfico para una interfaz, por ejemplo, GigabitEthernet 0/0.



### Tipos de ACL IPv4

- Las **ACL estándar** filtran paquetes solamente según la dirección de origen.

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

- Las **ACL extendidas** filtran paquetes según lo siguiente:

- El tipo y número de protocolo (p. ej., IP, ICMP, UDP, TCP...)
- Las direcciones IP de origen y destino
- Los puertos TCP y UDP de origen y destino

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

# Tipos de ACL IPv4

- Las ACL estándar y extendidas se pueden crear con un número o un nombre para identificar la ACL y su lista de instrucciones.

## ACL numeradas

Asignar un número según el protocolo que se debe filtrar.

- (1 a 99) y (1300 a 1999): ACL de IP estándar
- (100 a 199) y (2000 a 2699): ACL de IP extendida

## ACL denominada

Asignar un nombre para identificar la ACL.

- Los nombres pueden contener caracteres alfanuméricos.
- Se sugiere escribir el nombre en MAYÚSCULAS.
- Los nombres no pueden contener espacios ni signos de puntuación.
- Se pueden agregar o eliminar entradas dentro de la ACL.

# ACL IPv4 estándar

# Sintaxis de una ACL de IPv4 estándar numerada

- Router(config)# **access-list** *número-de-lista-de-acceso* { **deny** | **permit** | **remark** } *origen* [ *comodín-de-origen* ] [ **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
CTRL/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#
```

## Configurar listas ACL de IPv4 estándares

# Aplicar listas ACL de IPv4 estándares a las interfaces

### Procedimiento para la configuración de ACL estándar

Paso 1: Utilice el comando de configuración global **access-list** para crear una entrada en una ACL de IPv4 estándar.

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

La instrucción del ejemplo coincide con cualquier dirección que comience con 192.168.10.x. Utilice la opción **remark** (comentario) para agregar una descripción a la ACL.

Paso 2: Utilice el comando de configuración **interface** para seleccionar una interfaz a la cual aplicarle la ACL.

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

Paso 3: Utilice el comando de configuración de interfaz **ip access-group** para activar la ACL actual en una interfaz.

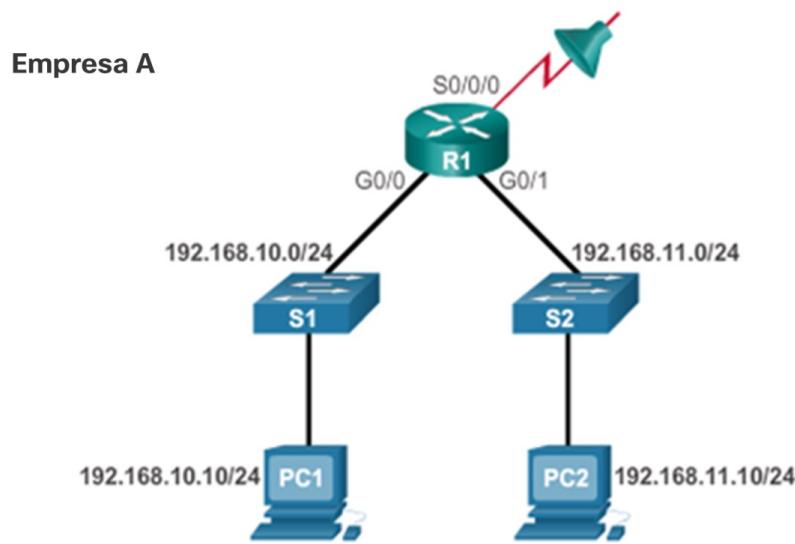
```
R1(config-if)# ip access-group 1 out
```

Este ejemplo activa la ACL estándar IPv4 1 en la interfaz como filtro de salida.

## Configurar listas ACL de IPv4 estándares

# Aplicar listas ACL de IPv4 estándares a las interfaces

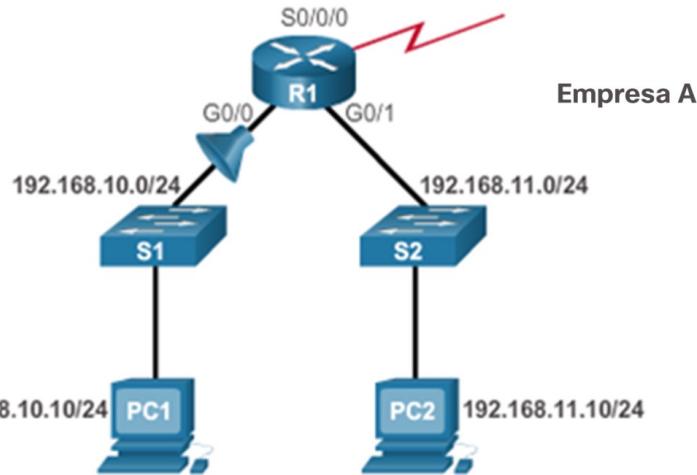
### Admisión de una subred específica



```
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
```

## Ejemplos de listas ACL de IPv4 estándares numeradas (continuación)

### Denegación de un host específico



```
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
```

# Sintaxis de una ACL de IPv4 estándar con nombre

### Ejemplo de ACL denominada

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

La cadena de nombres alfanuméricos debe ser única y no puede comenzar con un número.

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

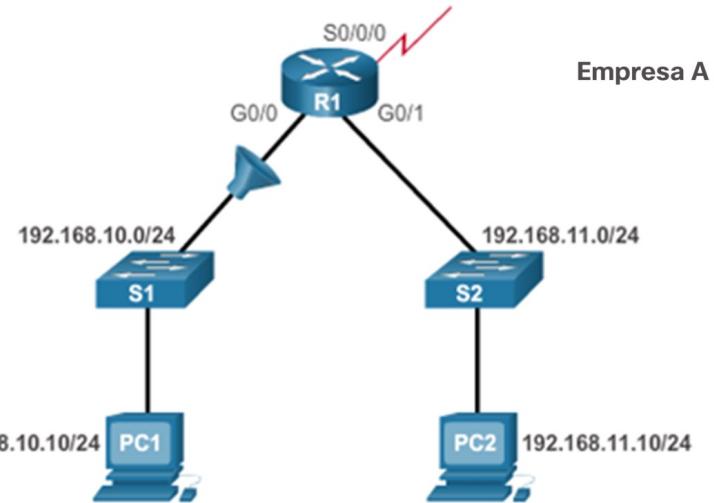
```
Router(config-if)# ip access-group name [in | out]
```

Activa la ACL IP denominada en una interfaz.

## Configurar listas ACL de IPv4 estándares

### Sintaxis de una ACL de IPv4 estándar con nombre (continuación)

Ejemplo de ACL denominada



```
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
```

# Método 1: Utilizar un editor de texto

## Edición de ACL numeradas mediante un editor de texto

Configuración

```
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
```

Paso 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
```

Paso 2

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

Paso 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
```

Paso 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
```

## Método 2: Utilizar números de secuencia

### Edición de ACL numeradas mediante números de secuencia

Configuración

```
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
```

Paso 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#
```

Paso 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#
```

Paso 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#
```

# Editar listas ACL estándares con nombre

## Cómo agregar una línea a la ACL denominada

```
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#
```

**Nota:** El comando `no sequence-number` named-ACL se usa para eliminar instrucciones individuales.

## Verificar listas ACL

```
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#
```

# Estadísticas de una ACL

```
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#
```

Resultado después de hacer ping  
a la PC3 desde la PC1

Aumentaron  
las  
coincidencias.

```
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#
```

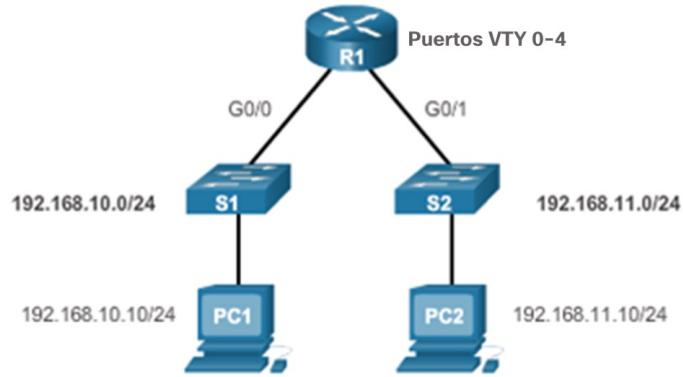
```
R1# show access-lists
Standard IP access list 1
    10 deny  192.168.10.10 (8 match(es))
    20 permit 192.168.0.0, wildcard bits 0.0.255.255
Standard IP access list NO_ACCESS
    15 deny  192.168.11.11
    10 deny  192.168.11.10 (4 match(es))
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
R1#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

Se borraron las coincidencias.

## El comando access-class

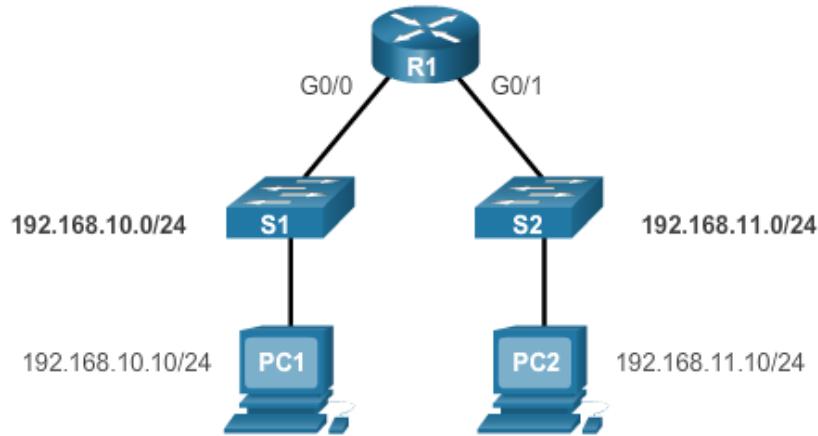
- El comando **access-class** configurado en el modo de configuración de línea restringe las conexiones entrantes y salientes entre una VTY determinada (en un dispositivo de Cisco) y las direcciones incluidas en una lista de acceso.



```
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
```

## Verificar que el puerto VTY esté asegurado

```
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#
```



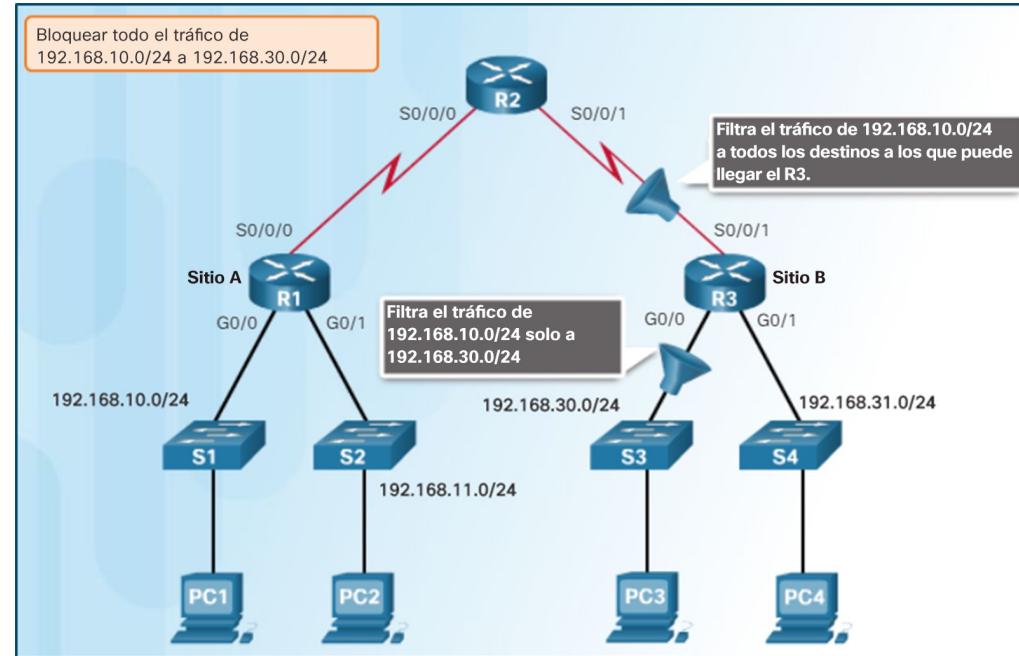
```
PC1>ssh 192.168.10.1
Login as: admin
Password: *****
R1>
```

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

## Revisión de la operación y configuración de ACL estándar

### Ubicación de ACL IPv4 estándar

- Se configurará una ACL estándar para que bloquee todo el tráfico que va de 192.168.10.0/24 a 192.168.30.0/24.
- Las ACL estándar se deben aplicar lo más cerca posible del destino y, por lo tanto, podrían aplicarse en la interfaz de salida R3 G0/0.
  - La aplicación en la interfaz de entrada R3 S0/0/1 impedirá que el tráfico llegue a 192.168.31.0/24 y, por lo tanto, no se debe aplicar a esta interfaz.



# Implementación de ACL IPv4 estándar

- La sintaxis completa del comando de ACL estándar es la siguiente:
  - **access-list** *ACL-#* {**deny** | **permit** | **remark**} *source* [*source-wildcard*] [**log**]
- Por ejemplo:
  - Permitir todas las direcciones IP en la red 192.168.10.0/24.
  - Usar el comando **no access-list 10** para quitar una ACL.
  - Utilizar la palabra clave **remark** para documentar una ACL para que sea más fácil comprender.

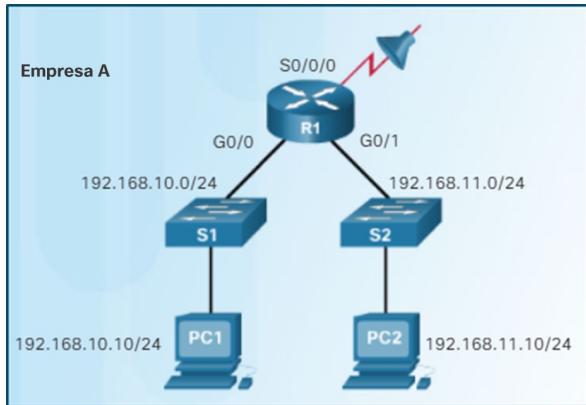
```
R1(config)# access-list 10 permit 192.168.10.0 0.0.0.255
R1(config)# exit
R1# show access-lists
Standard IP access list 10
    10 permit 192.168.10.0, wildcard bits 0.0.0.255
R1# conf t
Enter configuration commands, one per line. End with
CTRL/Z.
R1(config)# no access-list 10
R1(config)# exit
R1# 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#
```

## Revisión de la operación y configuración de ACL estándar

# Implementación de ACL IPv4 estándar

- Una ACL IPv4 se la vincula a una interfaz con el siguiente comando del modo de configuración de interfaz:
  - **ip access-group {ACL-# | access-list-name} {in | out}**



```
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
```

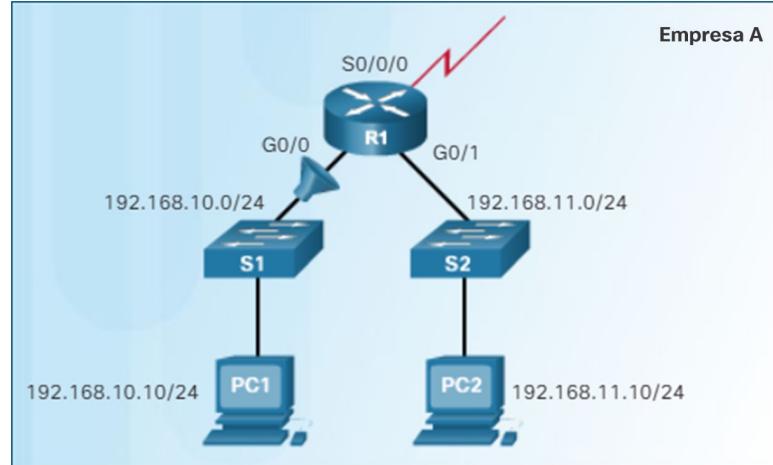
- **Nota:**

- Para eliminar una ACL de una interfaz, primero introduzca el comando **no ip access-group** en la interfaz; luego, introduzca el comando global **no access-list** para eliminar toda la ACL.

# Revisión de la operación y configuración de ACL estándar

## Implementación de ACL IPv4 estándar

- Crear una ACL con nombre estándar.
  - Utilice el comando de configuración global **ip access-list standard name**.
  - Los nombres son alfanuméricos, distinguen mayúsculas de minúsculas y deben ser únicos.
  - El comando ingresa en el modo de configuración de ACL con nombre estándar.
  - Use las instrucciones **permit**, **deny** o **remark**.
  - Aplique la ACL en una interfaz con el comando **ip access-group name**.



```
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
```

## Revisión de la operación y configuración de ACL estándar

# Implementación de ACL IPv4 estándar

- Utilice el comando **show ip interface** para verificar la ACL en la interfaz.
  - El resultado incluye el número o el nombre de la lista de acceso y el sentido en el que se aplicó la ACL.
- Utilice el comando **show access-lists [ACL-# | access-list-name]** para ver el contenido de una ACL estándar.

```
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#
```

# ACL IPv4 extendidas

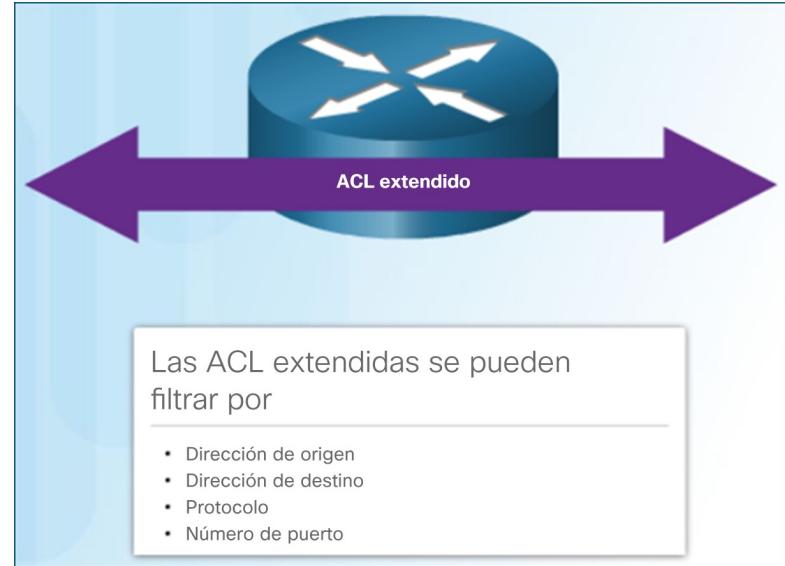
# Descripción general de la operación de la ACL

- Las ACL extendidas pueden filtrar el tráfico mediante el análisis de los números de puerto TCP.
- Los números de puerto TCP y UDP comunes incluyen los siguientes:

| Número de puerto | Protocolo | Aplicación   | Acrónimo |
|------------------|-----------|--|----------|
| 20               | TCP       | Protocolo de transferencia de archivos (datos)         | FTP      |
| 21               | TCP       | Protocolo de transferencia de archivos (Control)       | FTP      |
| 22               | TCP       | Secure Shell   | SSH      |
| 23               | TCP       | Telnet   | –        |
| 25               | TCP       | Protocolo simple de transferencia de correo            | SMTP     |
| 53               | UDP, TCP  | Servicio de nombres de dominios                        | DNS      |
| 67               | UDP       | Protocolo de configuración dinámica de host (servidor) | DHCP     |
| 68               | UDP       | Protocolo de configuración dinámica de host (Cliente)  | DHCP     |
| 69               | UDP       | Protocolo trivial de transferencia de archivos         | TFTP     |
| 80               | TCP       | Protocolo de transferencia de hipertexto               | HTTP     |
| 110              | TCP       | Protocolo de oficina de correos versión 3              | POP3     |
| 143              | TCP       | Protocolo de acceso a mensajes de Internet             | IMAP     |
| 161              | UDP       | Protocolo simple de administración de redes            | SNMP     |
| 443              | TCP       | Protocolo seguro de transferencia de hipertexto        | HTTPS    |

# Estructura de las ACL IPv4 extendidas

- Las ACL IPv4 extendidas proporcionan un filtrado más preciso.
  - Las ACL extendidas se numeran del 100 al 199 y del 2000 a 2699, lo que da un total de 799 ACL extendidas numeradas posibles.
  - Las ACL extendidas también pueden tener nombre.
  - Las ACL extendidas se utilizan con más frecuencia que las ACL estándar, porque proporcionan un mayor grado de control.



# Estructura de las ACL IPv4 extendidas

- Las ACL extendidas se pueden filtrar por protocolo y número de puerto.
- Se puede especificar una aplicación mediante la configuración de alguna de las siguientes opciones:
  - El número de puerto

```
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
```

- El nombre de un puerto conocido

```
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
```

- Nota:

- Utilice el signo de interrogación (?) para ver los nombres de puerto conocidos disponibles.
- P. ej.: **access-list 101 permit tcp any any eq ?**

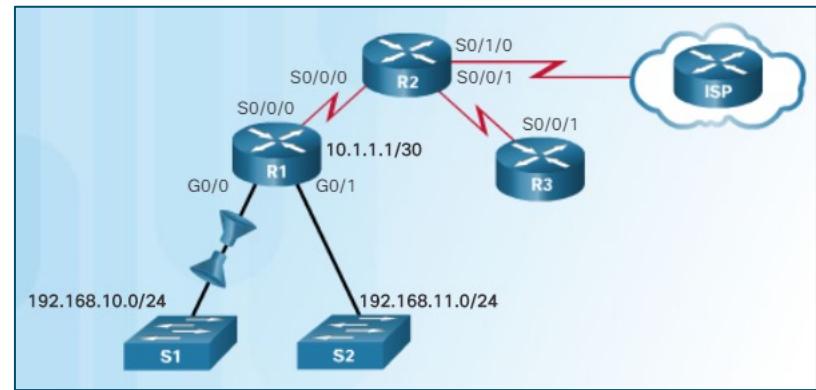
# Configuración de las ACL IPv4 extendidas

- La sintaxis completa del comando de ACL extendida es el siguiente:

- **access-list** *ACL-#* {**deny** | **permit** | **remark**} *protocol* {*source source-wildcard*} [*operator* [*port-number* | *port-name*]] {*destination destination-wildcard*} [*operator* [*port-number* | *port-name*]]]

- Por ejemplo:

- La ACL 103 permite enviar solicitudes a los puertos 80 y 443.
- La ACL 104 permite recibir respuestas de HTTP y HTTPS establecidos.
- El parámetro **established** permite que solo las respuestas al tráfico procedente de la red 192.168.10.0/24 vuelvan a esa red.

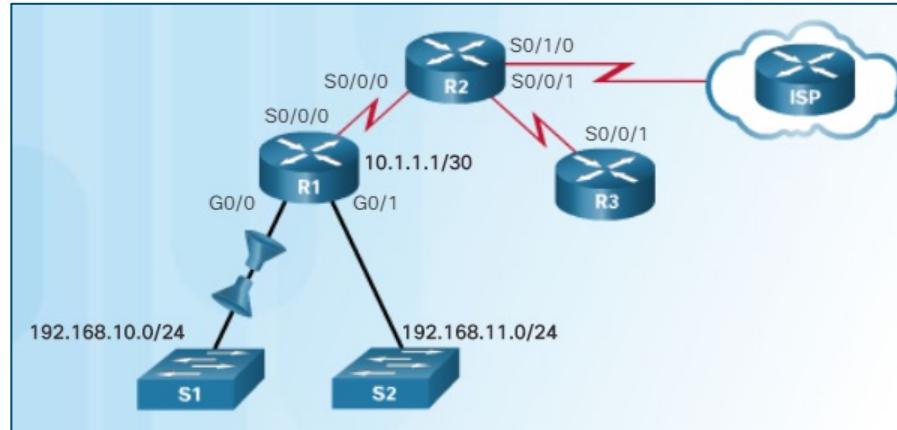


```
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)# access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
```

# Configuración de las ACL IPv4 extendidas

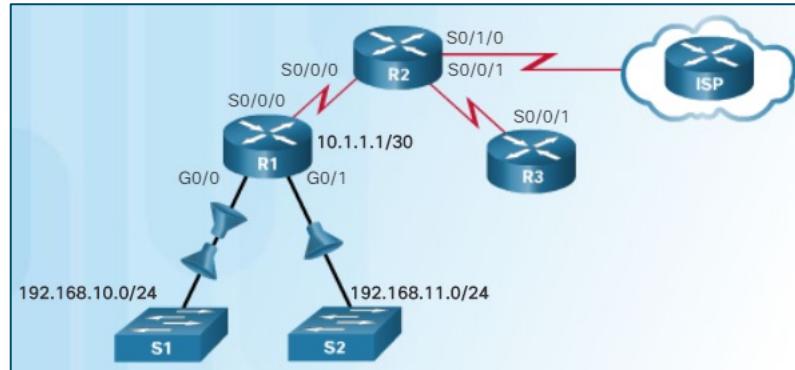
- La aplicación de ACL extendidas es similar a la de ACL estándar, con la diferencia de que debe aplicarse lo más cerca posible del origen.

```
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 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
```



# Configuración de las ACL IPv4 extendidas

- En este ejemplo, se deniega el tráfico FTP de la subred 192.168.11.0 que se dirige a la subred 192.168.10.0, pero se permite todo el tráfico restante.
- FTP utiliza dos números de puerto (puertos TCP 20 y 21); por lo tanto, se requieren dos ACE.
- En el ejemplo se utilizan los nombres de puerto conocidos **ftp** y **ftp-data**.
- Si no hay por lo menos una instrucción permit en una ACL, todo el tráfico en la interfaz donde se aplicó esa ACL se descarta.
- La ACL se aplica en sentido de entrada a la interfaz G0/1 del R1.

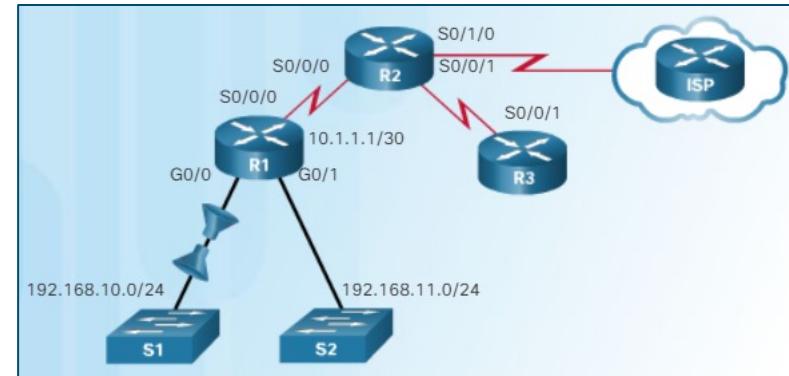


```
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255  
192.168.10.0 0.0.0.255 eq ftp  
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255  
192.168.10.0 0.0.0.255 eq ftp-data  
R1(config)# access-list 101 permit ip any any  
R1(config)# interface g0/1  
R1(config-if)#ip access-group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

# Configuración de las ACL IPv4 extendidas

- Las ACL extendidas con nombre se crean de la misma forma que las ACL estándar con nombre.
- En este ejemplo, se crean dos ACL con nombre.
  - SURFING permite que los usuarios en la red 192.168.10.0/24 salgan y vayan a los puertos 80 y 443.
  - BROWSING permite el regreso del tráfico HTTP y HTTPS.

```
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
```



# Configuración de las ACL IPv4 extendidas

- Los comandos **show ip interface** y **show access-lists** se pueden utilizar para verificar el contenido de las ACL extendidas.

- El resultado y los números de secuencia que se muestran en el resultado del comando **show access-lists** están en el orden en que se introdujeron las instrucciones.

- A diferencia de las ACL estándar, las ACL extendidas no implementan la misma lógica interna ni la misma función de hash.
- Las entradas de host no se enumeran automáticamente antes de las entradas de rango.

- El comando **show ip interface** se utiliza para verificar la ACL en la interfaz y el sentido en el que se aplicó.
- El resultado de este comando incluye el número o el nombre de la lista de acceso y el sentido en el que se aplicó la ACL.

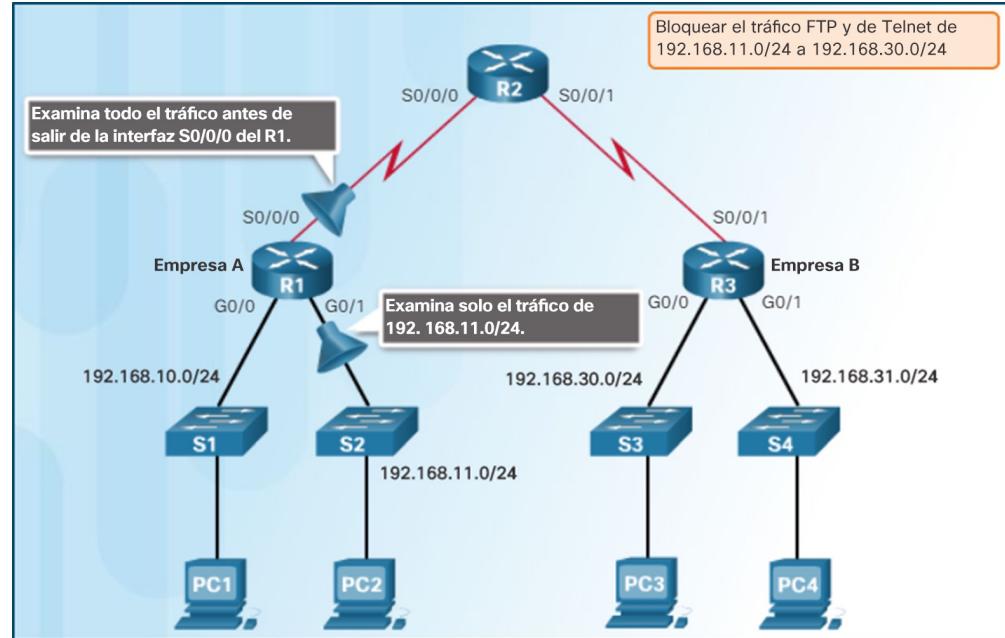
```
R1# show access-lists
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1#
```

```
R1# show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 192.168.10.1/24
<output omitted>
  Outgoing access list is BROWSING
  Inbound access list is SURFING
<output omitted>
```

## Revisión de la operación y configuración de ACL estándar

### Ubicación de ACL IPv4 extendida

- Se configurará una ACL extendida para que bloquee todo el tráfico FTP y Telnet que va de 192.168.11.0/24 a 192.168.30.0/24.
- La ACL extendida se debe aplicar lo más cerca posible del origen y, por lo tanto, podría aplicarse en la interfaz de entrada R1 G0/1.
  - La aplicación en la interfaz de salida R1 S0/0/1 impediría que el tráfico llegue 192.168.30.0/24, pero también procesaría innecesariamente paquetes de 192.168.10.0/24.

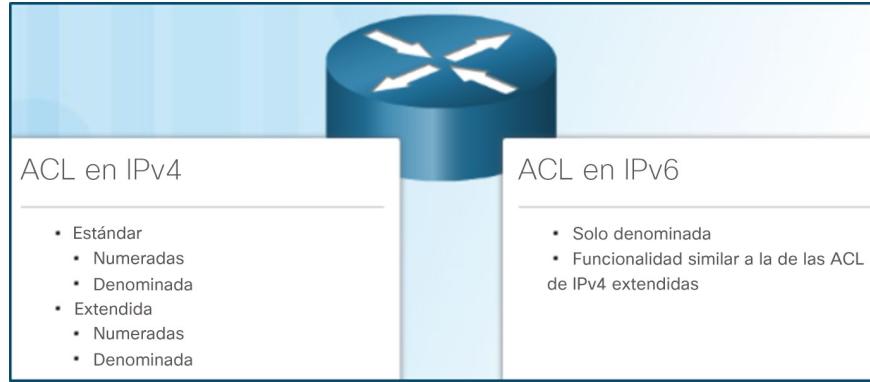


# ACL IPv6

# Creación de una ACL IPv6

- Las ACL IPv6 son similares a las ACL IPv4 tanto en la configuración como en el funcionamiento.

Existen dos tipos de ACL en IPv4: las estándar y las extendidas. Ambos tipos de ACL pueden tener un número o nombre.



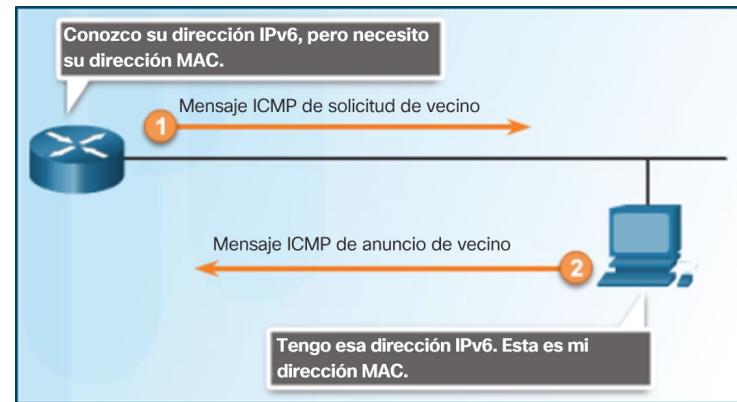
En cuanto a IPv6, hay solamente un tipo de ACL, que equivale a la ACL IPv4 extendida con nombre. No hay ninguna ACL con número en IPv6.

- Nota:**

- Una ACL IPv4 y una ACL IPv6 no pueden tener el mismo nombre.

# Creación de una ACL IPv6

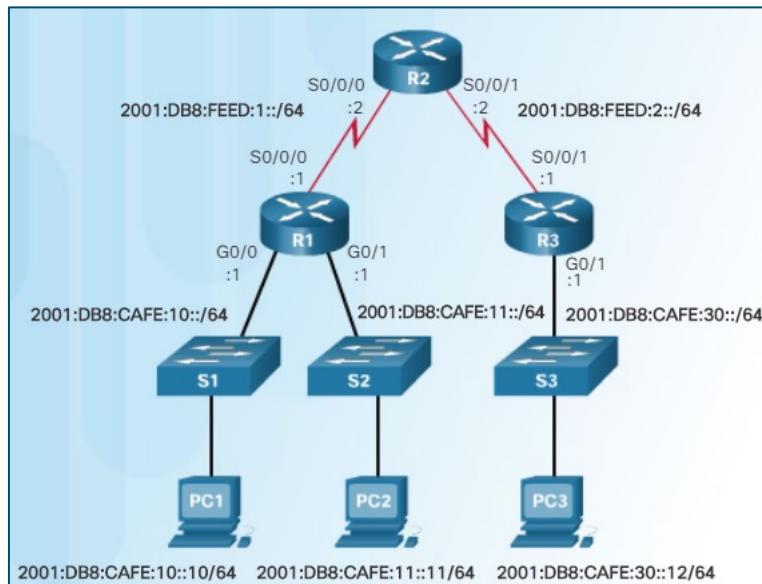
- Hay tres diferencias fundamentales entre las ACL IPv4 y las ACL IPv6:
  - El comando que se utiliza para aplicar una ACL IPv6 a una interfaz es **ipv6 traffic-filter**.
  - Las ACL IPv6 no usan máscaras de comodín, sino que, en cambio, especifican la longitud del prefijo para indicar el grado de coincidencia de una dirección IPv6 de origen o destino.
  - Una ACL IPv6 agrega dos instrucciones permit implícitas al final de cada lista de acceso IPv6.
    - **permit icmp any any nd-na**
    - **permit icmp any any nd-ns**
    - **deny ipv6 any any statement**
- Estas dos instrucciones adicionales permiten que los mensajes IPv6 ICMP Neighbor Discovery (ND) y Neighbor Solicitation (NS) logren lo mismo que ARP IPv4.



# Configuración de ACL IPv6

- Esta es la topología de ejemplo que se utilizará para demostrar ACL IPv6.
  - Todas las interfaces están configuradas y activas.

```
R1# show ipv6 interface brief
GigabitEthernet0/0      [up/up]
  FE80::FE99:47FF:FE75:C3E0
  2001:DB8:CAFE:10::1
GigabitEthernet0/1      [up/up]
  FE80::FE99:47FF:FE75:C3E1
  2001:DB8:CAFE:11::1
Serial0/0/0             [up/up]
  FE80::FE99:47FF:FE75:C3E0
  2001:DB8:FEED:1::1
<output omitted>
R1#
```



```
R2# show ipv6 interface brief
Serial0/0/0      [up/up]
  FE80::FE99:47FF:FE71:78A0
  2001:DB8:FEED:1::2
Serial0/0/1      [up/up]
  FE80::FE99:47FF:FE71:78A0
  2001:DB8:FEED:2::2
<output omitted>
R2#
```

```
R3# show ipv6 interface brief
GigabitEthernet0/0      [up/up]
  FE80::FE99:47FF:FE71:7A20
  2001:DB8:CAFE:30::1
Serial0/0/1             [up/up]
  FE80::FE99:47FF:FE71:7A20
  2001:DB8:FEED:2::1
R3#
```

# Configuración de ACL IPv6

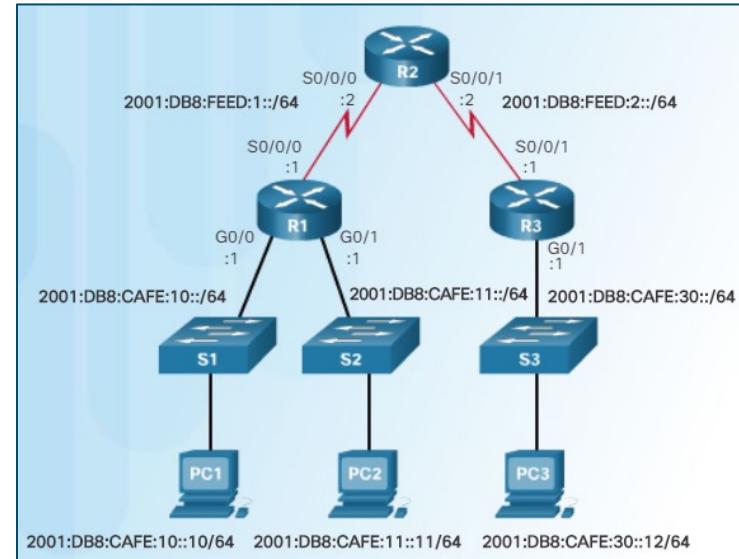
- En IPv6 solo hay ACL con nombre y su configuración es similar a la de ACL IPv4 extendidas con nombre.

```
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]]}
```

```
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#
```

- En este ejemplo:

- La primera instrucción da el nombre IPv6 ACL **NO-R3-LAN-ACCESS**.
- La segunda instrucción deniega todos los paquetes IPv6 de 2001:DB8:CAFE:30::/64 con destino a a cualquier red IPv6.
- La tercera instrucción permite el resto de los paquetes IPv6.



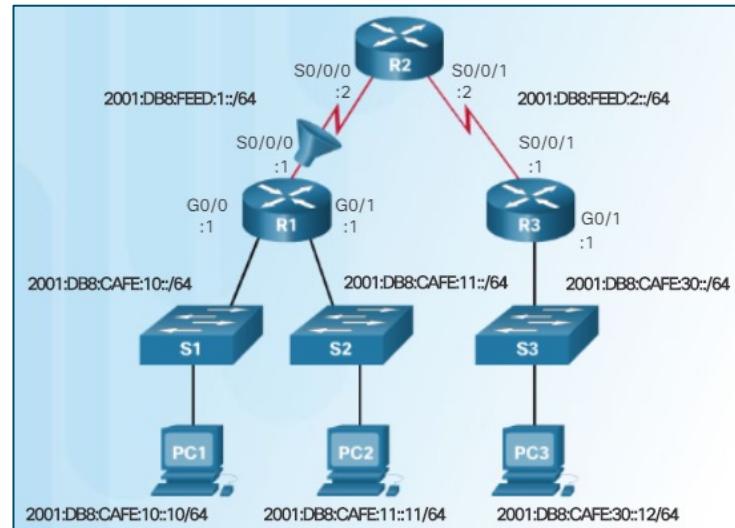
# Configuración de ACL IPv6

- Una vez configurada, una ACL IPv6 se vincula a una interfaz con el siguiente comando de interfaz:
  - **ipv6 traffic-filter access-list-name {in | out}**

El comando aplica la ACL IPv6 NO-R3-LAN-ACCESS en sentido de entrada a la interfaz S0/0/0 de R1.

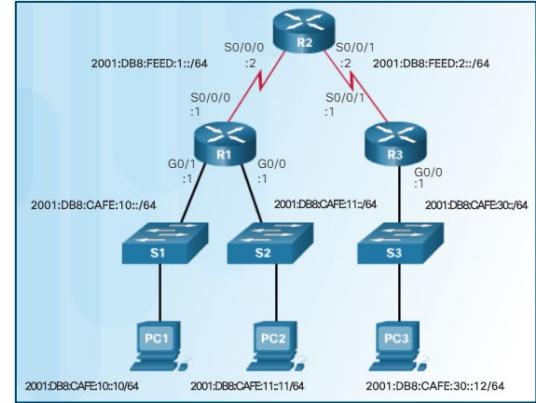
```
R1(config)# interface s0/0/0
R1(config-if)# ipv6 traffic-filter NO-R3-LAN-ACCESS in
```

- Para quitar una ACL IPv6, introduzca el comando **no ipv6 traffic-filter** en la interfaz y, luego, introduzca el comando global **no ipv6 access-list** para quitar la lista de acceso.
- Nota: Tanto en IPv4 como en IPv6 se utiliza el comando **access-class** para aplicar una lista de acceso a los puertos VTY.



# Configuración de ACL IPv6

- En este ejemplo, una ACL IPv6 permite el acceso limitado de usuarios de LAN del R3 a las LAN en el R1.
- Estas ACE permiten el acceso desde cualquier dispositivo hasta el servidor web (2001:DB8:CAFE:10::10).
  - El resto de los dispositivos tienen denegado el acceso a la red 2001:DB8:CAFE:10::/64.
  - A la PC3 (2001:DB8:CAFE:30::12) se le permite el acceso por Telnet a la PC2 (2001:DB8:CAFE:11::11).
  - El resto de los dispositivos tiene denegado el acceso por Telnet a la PC2.
  - El resto del tráfico IPv6 se permite al resto de los destinos.
  - La lista de acceso IPv6 se aplica en sentido de entrada a la interfaz G0/0 , por lo que solo la red 2001:DB8:CAFE:30::/64 se ve afectada.



```
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)#

```

# Configuración de ACL IPv6

- Los comandos que se utilizan para verificar una lista de acceso de IPv6 son similares a los que se utilizan para las ACL IPv4.
- Utilice el comando **show ipv6 interface** para ver qué ACL y dirección están configuradas en una interfaz.

```
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>
```

- Utilice el comando **show access-lists** para ver todas las listas de acceso IPv4 e IPv6 configuradas.
  - Observe que los números de secuencia de ACL IPv6 se muestran al final de la ACE.
- El comando **show running-config** muestra todas las ACE y las instrucciones **remark**.

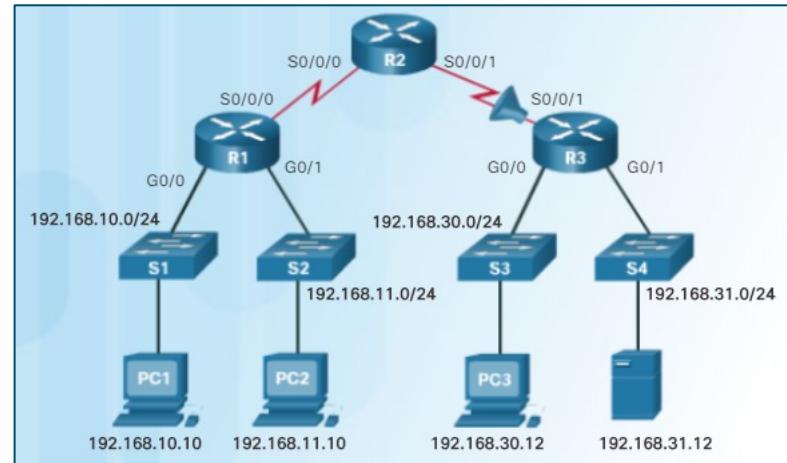
```
R3# show access-lists
IPv6 access list RESTRICTED-ACCESS
  permit tcp any host 2001:DB8:CAFE:10::10 eq www sequence 20
  permit tcp any host 2001:DB8:CAFE:10::10 eq 443 sequence 30
  deny ipv6 any 2001:DB8:CAFE:10::/64 sequence 50
  permit tcp host 2001:DB8:CAFE:30::12 host 2001:DB8:CAFE:11::11 eq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

# Solución de problemas de ACL

## Solución de problemas de las ACL

# Errores comunes de las ACL

- Los errores más comunes de las ACL incluyen introducir las ACE en el orden incorrecto y no aplicar los criterios adecuados a las reglas de ACL.
- En este ejemplo, el host 192.168.10.10 no tiene conectividad de Telnet con 192.168.30.12.
  - El comando **show access-lists** muestra coincidencias para la primera instrucción “deny”, lo que indica que el tráfico coincidió con esta ACE.
- Solución:**
  - El host 192.168.10.10 no tiene conectividad con 192.168.30.12 porque la instrucción 10 deniega el host 192.168.10.10; por lo tanto, nunca se puede establecer la coincidencia con la instrucción 20.
  - Las instrucciones 10 y 20 deben invertirse.



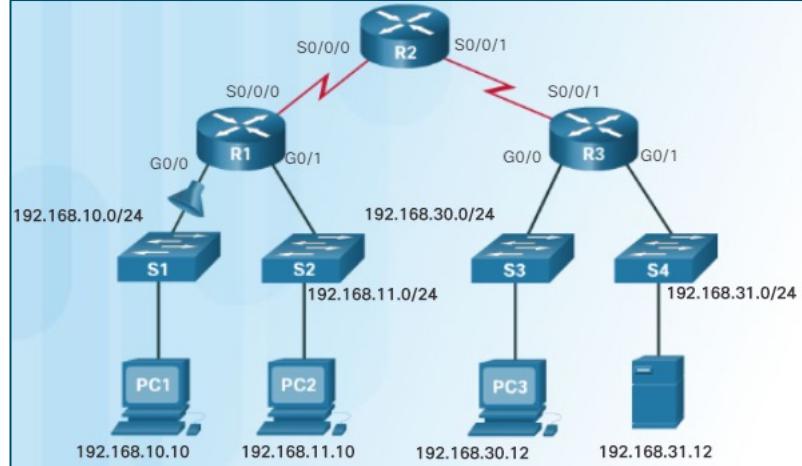
```
R3# show access-lists
Extended IP access list 110
 10 deny tcp 192.168.10.0 0.0.0.255 any (12 match(es))
 20 permit tcp 192.168.10.0 0.0.0.255 any eq telnet
 30 permit ip any any
```

# Errores comunes de las ACL

- En este ejemplo, la red 192.168.10.0/24 no puede utilizar TFTP para conectarse a la red 192.168.30.0/24.

### Solución:

- La instrucción 30 en la lista de acceso 120 permite todo el tráfico TCP.
- Sin embargo, como TFTP utiliza UDP en lugar de TCP, se deniega implícitamente.
- La instrucción 30 debería ser **permit ip any any**.

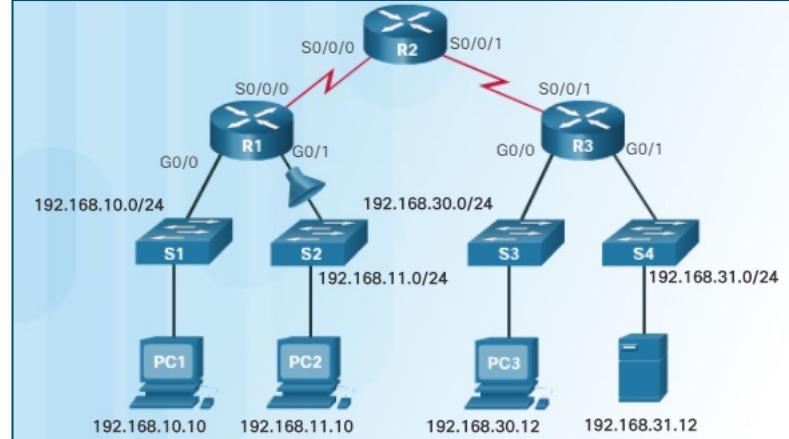


```
R3# show access-lists 120
Extended IP access list 120
 10 deny tcp 192.168.10.0 0.0.0.255 any eq telnet
 20 deny tcp 192.168.10.0 0.0.0.255 host 192.168.31.12 eq smtp
 30 permit tcp any any
```

## Solución de problemas de las ACL

# Errores comunes de las ACL

- En este ejemplo, la red 192.168.11.0/24 puede utilizar Telnet para conectarse a 192.168.30.0/24, pero según la política de la empresa, esta conexión no debería permitirse.
- Los resultados del comando **show access-lists 130** indican que se encontró una coincidencia para la instrucción “permit”.



### Solución:

- El número de puerto de Telnet en la instrucción 10 de la ACL 130 figura en el orden incorrecto, ya que actualmente deniega cualquier paquete de origen con un número de puerto equivalente a Telnet.
- Configure **10 deny tcp 192.168.11.0 0.0.0.255 192.168.30.0 0.0.0.255 eq telnet**.

```
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))
```

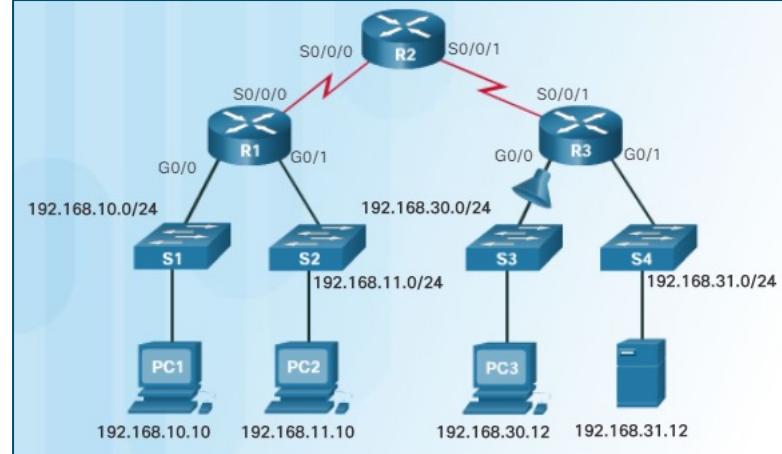
## Solución de problemas de las ACL

# Errores comunes de las ACL

- En este ejemplo, el host 192.168.30.12 puede conectarse a 192.168.31.12 mediante Telnet, pero la política de la empresa establece que esta conexión no debe permitirse.
- Los resultados del comando **show access-lists 140** indican que se encontró una coincidencia para la instrucción “permit”.

### Solución:

- El host 192.168.30.12 puede utilizar Telnet para conectarse a 192.168.31.12 porque no hay reglas que denieguen el host 192.168.30.12 o su red como origen.
- La instrucción 10 de la lista de acceso 140 deniega la interfaz del router por la que el tráfico ingresa a este.
- La dirección host IPv4 en la instrucción 10 debería ser 192.168.30.12.



```
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))
```

