



TEMA 2: ACCESS CONTROL LIST

ACLs de Cisco

- Listas de control de acceso → controlar flujo info. dentro de la red.
- Motivo < Δ seguridad → filtrar / denegar entrada/salida tráfico no se asocian a los interfaces → al final usa regla predeterminada (denegar)

Tipos:

- ACL Estándar
- ACL Extendida
- ACL Nombre

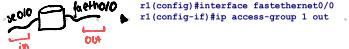
ACL Complejas

dinámicas referidas basadas en el @

Proceso de creación de un ACL

- 1) Definición ACL
- ```
> config # r1(config)# access-list 1 permit 192.168.1.0 0.0.0.255
```

- 2) Asociación a un interface



## ACLs Extendidas

flexibilidad  
control

- Basadas en:
- Dirección IP origen o destino
- Puertos TCP / UDP origen o destino
- establecimiento de conexión TCP
- tipo de protocolo
- campos de datagrama o fragmentos TCP / UDP

• Al incorporar IP destino se filtra de forma más sencilla posible del

```
access-list num acl permit/deny protocolo
IP_origen [wildmask_origen] [operacion
[puerto_origen]] IP_destino [wildmask_destino]
[operacion [puerto_destino]] [establecido]
[prioridad] [tos tos] [log] [log
[time-range nombre_rango]] [fragmentos]
```

Protocolo: ip, icmp, tcp, udp

Operación: ls (less than), gt (greater than), eq (equal), neq (not equal), y range (rango de valores incluidos).

Puerto: ftp, ftp-data, www, ssh, tftp, snmp, ... (o valores numéricos)

## ACLs Estándar

• Reglas basadas en dirección IP origen

• Numeradas: 1-99 100-199

• Aplicar a una interfaz

```
access-list numero-acl {permit | deny} IP_origen
[wildmask_origen]
interface _____
> ip access-group numero-acl {in | out}
```

## Ejemplos:

- Permitir tráfico desde 192.168.0.0/16:

```
access-list 10 permit 192.168.0.0 0.0.255.255
```
- Permitir tráfico desde el host 192.168.1.4:

```
access-list 10 permit 192.168.1.4 0.0.0.0
access-list 10 permit host 192.168.1.4
```
- Permitir tráfico desde cualquier sitio:

```
access-list 10 permit 0.0.0.0 255.255.255.255
access-list 10 permit any
```

## ACLs con Nombre

• Identificación a secuencia, alfanumérica.

- Se introducen a partir IOS 4.2
- MÁS fáciles de administrar [✓] menor relojero [✗] menor número en cualquier posición

```
ip access-list {extended | standard} nombre
permit | deny protocolo IP_origen [wildmask_origen]
[operacion puerto_origen] IP_destino
[wildmask_destino] [operacion puerto_destino]
[establecido] [precedencia prioridad] [tos tos] [log]
[time-range nombre_rango] [fragmentos]
```

## Comandos comprobación:

```
show access-list
Visualizar una ACL
show access-list numero-ACL | nombre-ACL
ACLs asociados a las interfaces:
show ip interface
Añade comentarios a las ACLs:
access-list numero-acl remark comentario
```

Permitir tráfico http y ftp a los hosts de la red 172.16.5.0/24, cualquiera que sea el destino (Se aplica en sentido saliente):

```
access-list 110 permit tcp 172.16.5.0 0.0.0.255 any eq www
access-list 110 permit tcp 172.16.5.0 0.0.0.255 any eq ftp
access-list 110 permit tcp 172.16.5.0 0.0.0.255 any eq ftp-data
```

Permitir sólo tráfico TCP saliente (se aplica al sentido entrante):

```
access-list 120 permit tcp any any established
```

Permitir tráfico ICMP entrante (también en sentido entrante):

```
access-list 120 permit icmp any any echo-reply
access-list 120 permit icmp any any unreachable
```

## El orden es IMP

```
access-list 1 deny 192.168.1.5 0.0.0.0
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 1 deny 192.168.1.5 0.0.0.0
```

→ 1 niegas  
→ 2 permit TODAS  
→ 3 permit  
→ 4 niegas → niegas = 5

## Restricción de acceso VTY

↓  
(puertos virtuales)  
restringir acceso remoto al no tráfico   
Vty → Pueden tener aplicadas ACLs

```
rl(config)#access-list 2 permit 172.16.5.50
rl(config)#line vty 0 4 *
rl(config-line)# login
rl(config-line)# password contraseña
rl(config-line)# access-class 2 in
```

0 → no controla  
4 → 4 líneas q pueden entrar obtienen a red  
comandos q pueden tener ...

## Planteamiento

¿Qué necesitamos filtrar?

¿Dónde debemos aplicar filtro? (interfaces y routers)

¿Qué reglas implementan este filtro?

¿En qué orden se debe procesar?

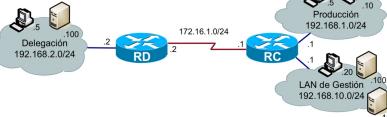
rl# reload in 30 → Q gana estar la regla activa

Ejercicio: reunir el tráfico de salida de la red

192.168.1.0/24



```
> config terminal
> access-list 1 permit ip 192.168.1.0 0.0.0.255
> interface serial0/0
> (...) ip access-group 1 out ("tráfico saliente")
```



- Configurar la red propuesta (equipos con sus interfaces, direcciones IP propuestas y encaminamiento). Verificar la conectividad entre los equipos.
- Los routers sólo pueden administrarse por el puerto de consola o remotamente (telnet) desde el red de gestión (192.168.10.0/24). Nota: es necesario asignar contraseñas de acceso y de administración.
- Los usuarios de la red de producción no pueden alcanzar la delegación, salvo desde el equipo 192.168.1.5
- En la red de gestión se cuenta con un servidor Web de incidencias (en la IP 192.168.10.100). Este servidor recibe conexiones desde toda la red. El resto de equipos de esta subred no pueden recibir conexiones TCP entrantes.

```
> access-list 1 permit 192.168.10.0 0.0.0.255
> line vty 0 4
> access-class 1 in
> password mm
> login
```

## ACLs Complejas

### ACLs Dinámicas

lock-and-key

- Tráfico bloqueado usuario autenticado
- Se config Un → **ACL extendida**  
Para bloquear tráfico IP a través

- Si un usuario autenticado → Nueva regla ACL reconfigurar → por un

```
R1(config)# username pruebas password 0 cisco
R1(config)# access-list 100 dynamic-router-telnet
timeout 15 permit ip 192.168.1.0 0.0.0.255
192.168.2.0 0.0.0.255
```

# TEMA 2: ACCESS Control List

## (ACLs complejas)

### ACLs Reflexivas

- permiten filtrar en (src) de una sesion ya establecida  
    (filtro con estado)
- uso habitual → limitar trafico a entorno
  - ↳ bloqueando todo lo que no sea respuesta al → saliente

✓ examina condiciones salientes  
genera reglas temporales  
para aceptar datagramas → respuesta

### Utilizan → ACLs extendidos con nombre

- equivalente a clausula "established"
- any puede utilizarse si con TCP o UDP

### EJEMPLO:

```
R1(config)# ip access-list extended SALIENTE
R1(config-ext-nacl)# permit tcp 192.168.1.0
0.0.0.255 any reflect TRAFICO_TCP

R1(config)# ip access-list extended ENTRANTE
R1(config-ext-nacl)# evaluate TRAFICO_TCP

R1(config)# interface fa 0/0
R1(config-if)# ip access-group SALIENTE out
R1(config-if)# ip access-group ENTRANTE in
```

### ACLs basadas en @

- permiten establecer control de acceso en (src) de franjas horarias y/o días de la semana

```
R1(config)# time-range MisHoras
R1(config-time-range)# periodic Monday Friday 10:00
to 18:00
```

```
R1(config)# access-list 105 permit tcp host
192.168.1.10 any eq www time-range MisHoras
```

```
R1(config)# interface serial 0/0
R1(config-if)# ip access-group 101 out
```

Denegar FTP 20 y 21 entre subredes y permitir el resto

```
> access-list 101 deny tcp 192.168.2.0 0.0.0.255
any 192.168.1.0 0.0.0.255 eq 21
```

↳ 11 11 eq 20

```
> access-list 101 permit ip any any
```

```
> int fa 0/0
```

```
> (...) ip access-group 101 in
```

Denegar telnet (23) a la subred 192.168.1.0

```
> access-list 102 deny tcp 192.168.1.0 0.0.0.255
any eq 23
```

```
> access-list 102 permit ip any any
```

```
> int fa 0/0
```

```
> (...) ip access-group 102 out
```