
CISCO SYSTEMS

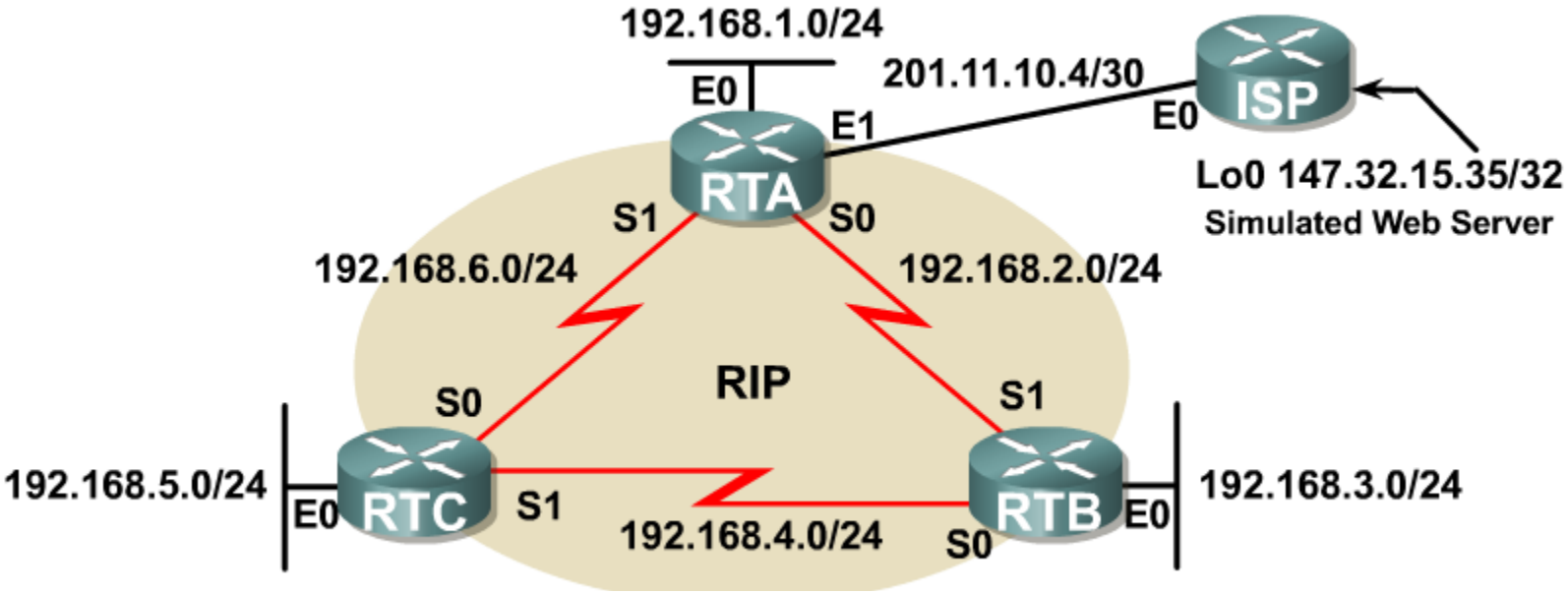
Access Control Lists (ACLs)

CCNA 2 v3.0

Conceptos Básicos de ACL

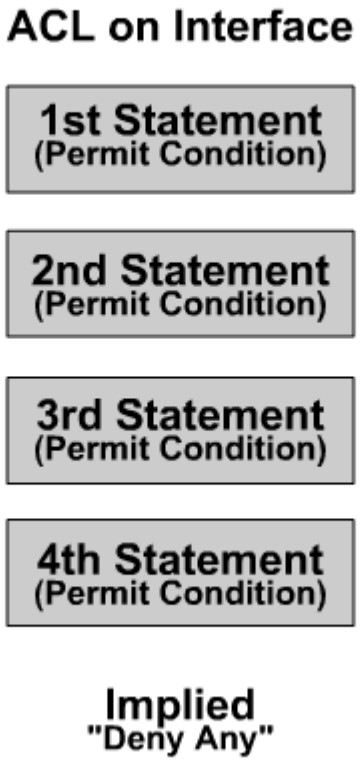
Configuración de Laboratorio: Topología y Scripts

- **La Topología mostrada abajo es usada en esta presentación.**
  - **RTA es un 2514; RTB, RTC, e ISP son 2501's**
  - **Todos los routers están corriendo imágenes IOS C2500-JS-L, Ver. 12.2(13b)**
  - **E0's y S0's tienen la 1ª dirección IP; E1 y S1's, tienen la 2ª**
  - **S0's son los DCE de los enlaces**



Conceptos Básicos de ACL

- **ACLs pueden ser configuradas en un router pare permitir o negar un paquete basado en una lista de condiciones.**
  - **Esta lista de condiciones es leída secuencialmente, de arriba hacia abajo, por el router hasta que una coincidencia suceda.**
  - **La última condición es siempre una negación implícita “deny any”**
  - **Usted puede permitir o negar paquetes basados en algo como:**
    - **Dirección Origen**
    - **Dirección Destino**
    - **Puertos TCP & UDP**



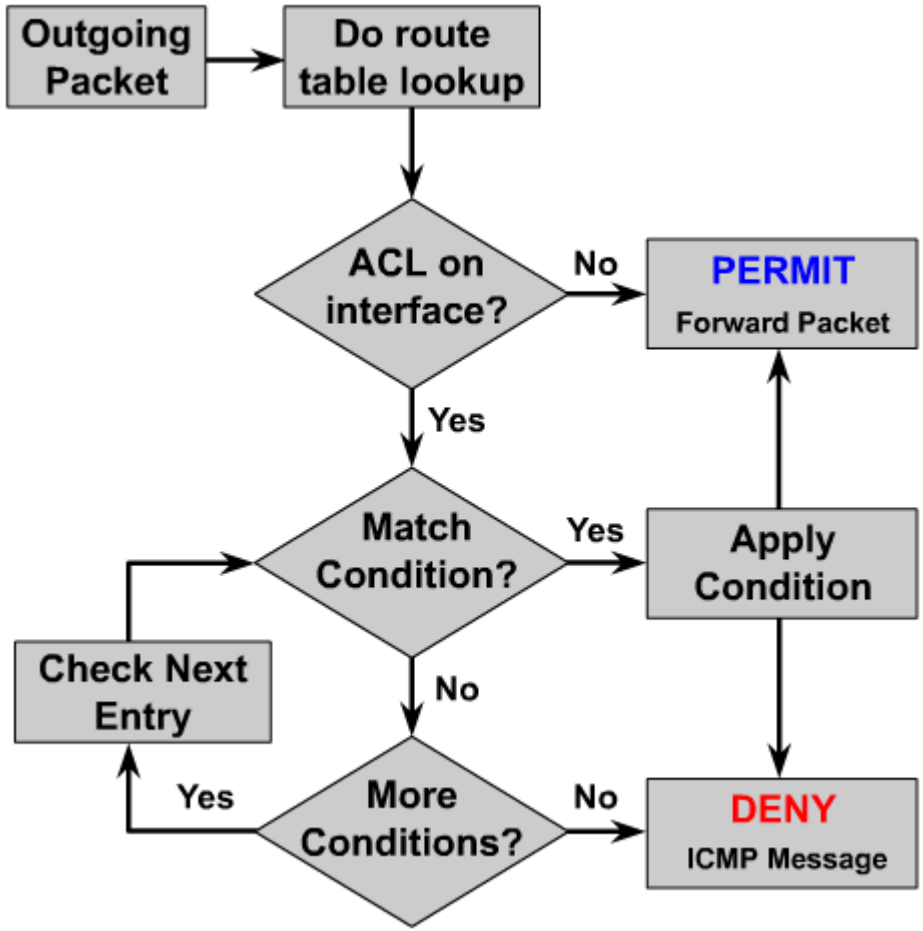
Cómo usa un Router una ACL (salida)

Verifica para ver si el paquete es enrutable. Si es así, busca una ruta en la tabla de enrutamiento.

Verifica si hay una ACL de salida para la interface

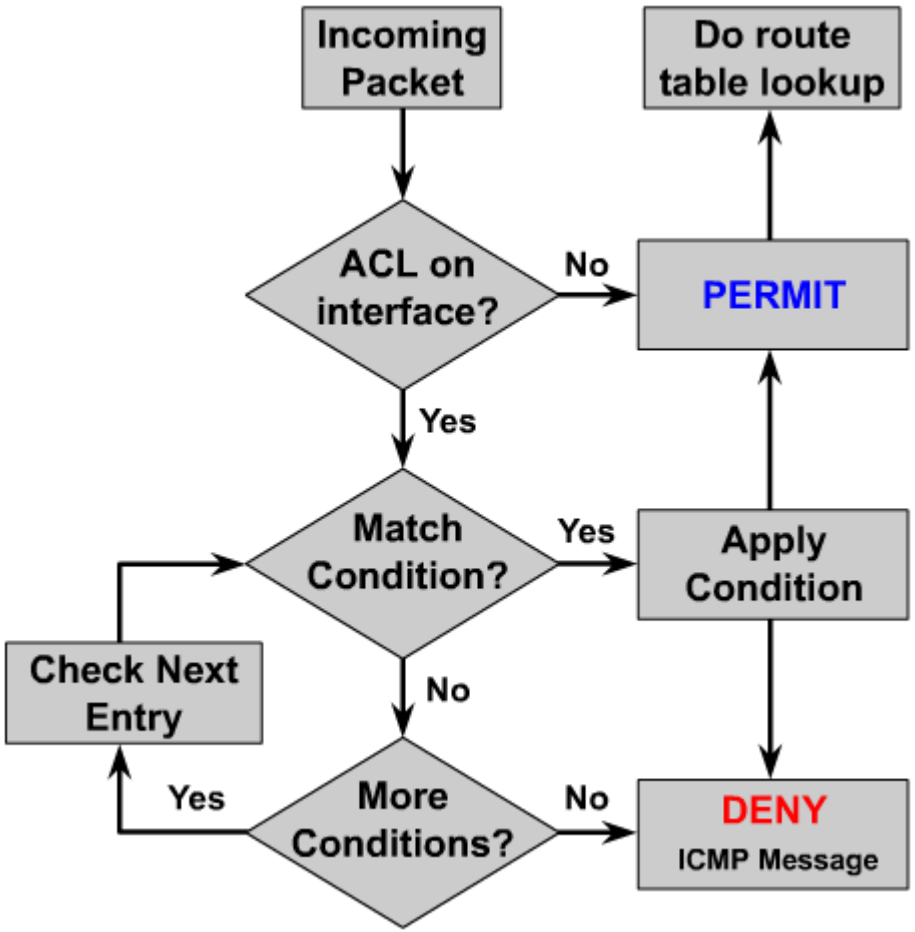
- **Si no hay ACL, conmuta el paquete a la interface destino para su salida**
- **Si existe una ACL, verifica el paquete contra las sentencias de la ACL en forma secuencial– negando o permitiendo basado en una condición coincidente.**

Si ninguna sentencia coincide, ¿qué sucede?



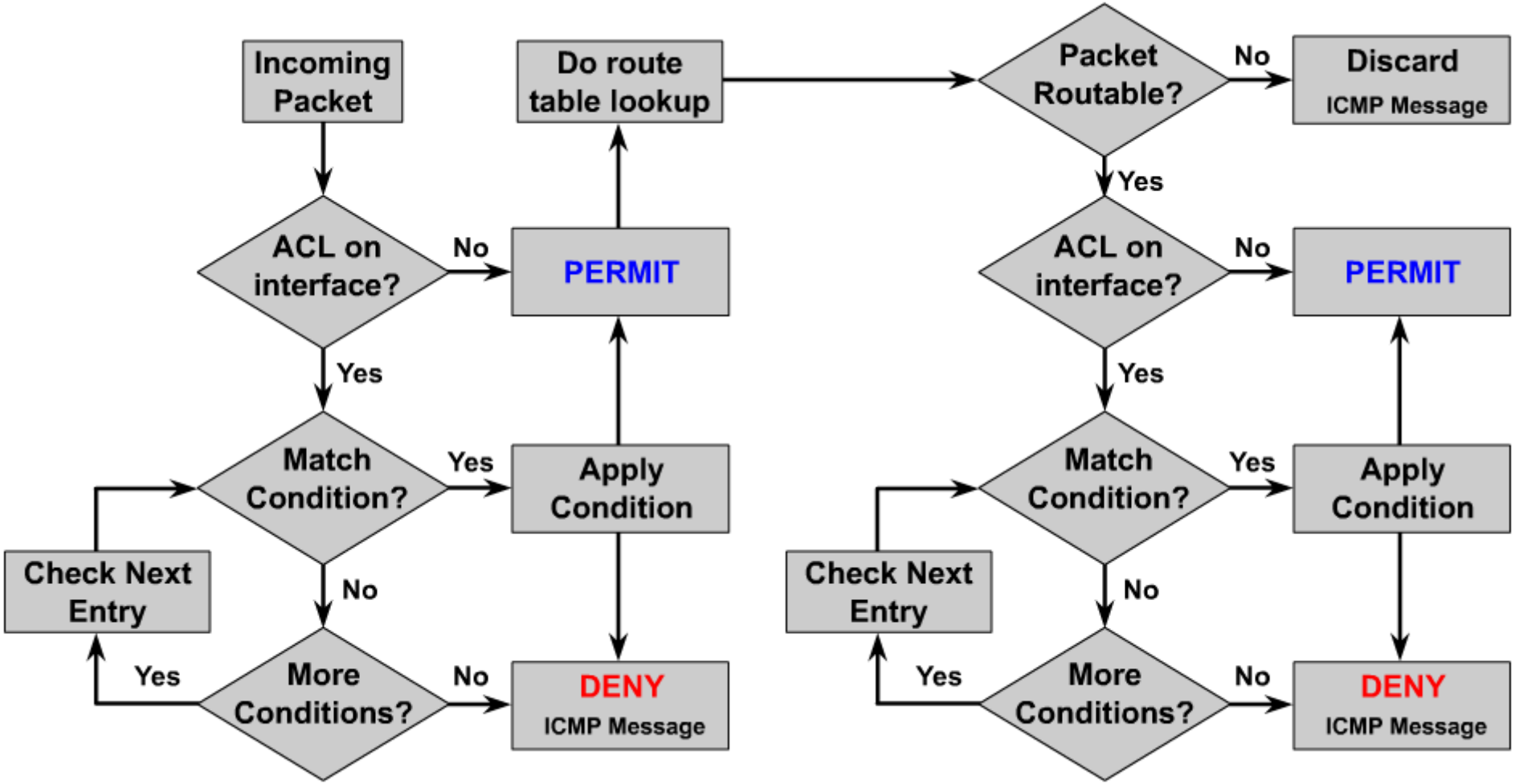
Cómo usa un Router una ACL (entrada)

- **Si una ACL es configurada para filtrar tráfico de entrada, la búsqueda en la tabla de enrutamiento es realizada solo si el paquete es permitido.**



Procesando una ACL de Entrada y Salida

Cisco.com



Configurando ACLs Estándar

Dos Tipos de IP ACLs

- **ACLs Estándar**

  Filtra el tráfico basado en la dirección origen solamente

- **ACLs Extendidas**

  Pueden filtrar tráfico basado en:

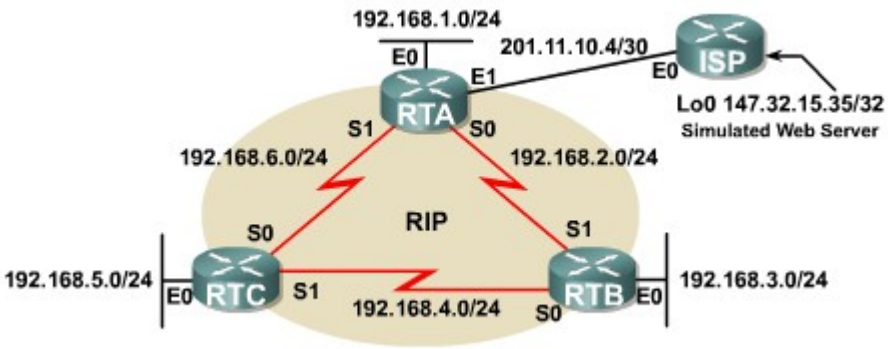
  Dirección Origen

  Dirección Destino

  Puertos TCP y UDP

Creando ACLs: Dos Tareas Básicas

- **Escriba las sentencias de la ACL en forma secuencial en modo de configuración global.**
- **Aplique la ACL a una o más interfaces en el modo de configuración de interface**




```
!Step 1: Write the ACL in global configuration mode

RTA#config t
RTA(config)#access-list 1 permit 192.168.1.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.3.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.5.0 0.0.0.255
!---Implied "Deny Any"

!Step 2: Apply the ACL to the desired interface

RTA(config)#interface e1
RTA(config-if)#ip access-group 1 out
RTA(config-if)#end
```

Sintaxis para Escribir una ACL Estándar


```
router(config)#access-list access-list-number
{permit|deny} source-prefix source-wildcard
```

- El argumento access-list-number especifica el tipo de ACL.
- El argumento {permit|deny} especifica la acción a tomar sobre un paquete
- El argumento source-prefix especifica la red, subred, rango de host, o simple dirección de host.
- El argumento source-wildcard especifica qué bits son “verificados” y qué bits son “ignorados” en el prefijo-origen (source-prefix).

El argumento access-list-number


```
router(config)#access-list access-list-number
{permit|deny} source-prefix source-wildcard
```

El access-list-number especifica qué protocolo se filtrará y si es una ACL estándar o extendida.

Este argumento puede ser reemplazado con el argumento nombre.

ACLs Nombradas serán vistas después.

Range	Protocol
1-99	IP standard access list
100-199	IP extended access list
200-299	Protocol type-code access list
300-399	DECnet access list
400-499	XNS standard access list
500-599	XNS extended access list
600-699	Appletalk access list
700-799	48-bit MAC address access list
800-899	IPX standard access list
900-999	IPX extended access list
1000-1099	IPX SAP access list
1100-1199	Extended 48-bit MAC address access list
1200-1299	IPX summary address access list
1300-1399	IP standard access list (expanded range)
2000-2699	IP extended access list (expanded range)

El argumento {permit|deny}


```
router(config)#access-list access-list-number
{permit|deny} source-prefix source-wildcard
```

Después que haya escrito access-list y seleccionado el número correcto de la lista de acceso, escriba permit o deny dependiendo de la acción que desea tomar.

La acción de permitir o negar son referidas como “filtrado” (“filtering”)

El argumento *source-prefix*


```
router(config)#access-list access-list-number
{permit|deny} source-prefix source-wildcard
```

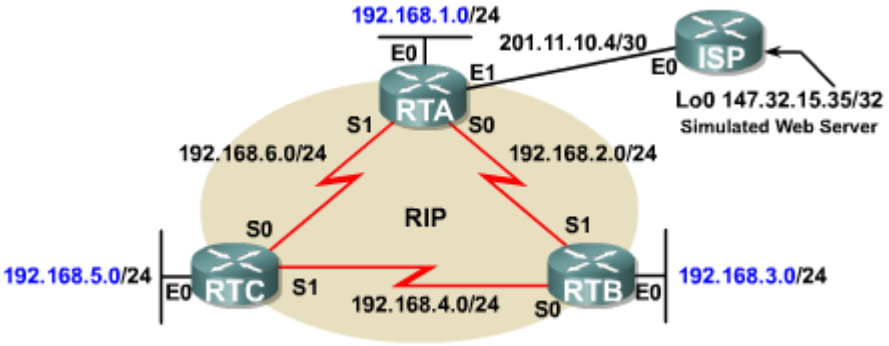
El argumento *source-prefix* puede ser una dirección de subred, un rango de direcciones, o una simple dirección de host.

Ejemplo de *source-prefix*

- **En nuestro ejemplo, queremos permitir a todos los hosts en las tres redes LAN el acceso a Internet.**

Por lo tanto tenemos escritas tres sentencias, una para cada subred.

El argumento *source-prefix* en este caso es cada dirección de subred de las LAN's




```
RTA#config t
RTA(config)#access-list 1 permit 192.168.1.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.3.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.5.0 0.0.0.255
```

El argumento source_wildcard


```
router(config)#access-list access-list-number
{permit|deny} source-prefix source-wildcard
```

- El argumento source-wildcard especifica cuáles bits deberán ser verificados en el prefijo-origen (source-prefix).

  Comúnmente llamada “wildcard mask”, es un número de 32-bit representado en formato decimal-punteado.

  Un “0” significa “verifica” esta posición de bit.

  Un “1” significa “ignora” esta posición de bit.

Explicación de la “Wildcard Mask”

La “máscara wildcard” (“Wildcard Mask”) no tiene relación funcional con la máscara de subred.

Sin embargo, en muchos casos la máscara de subred puede ser usada para derivar la máscara wildcard.

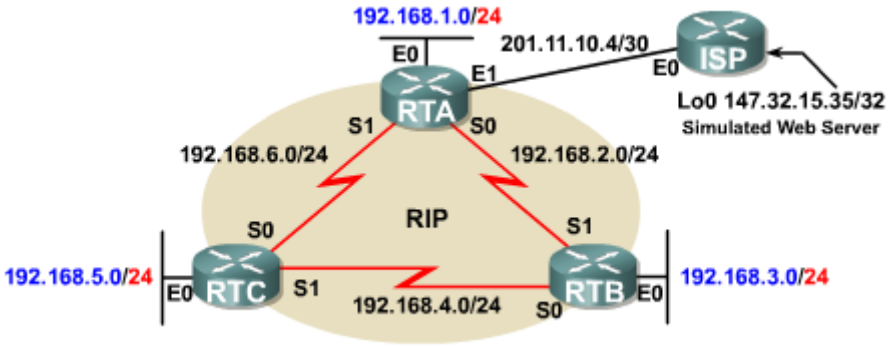
- Por ejemplo, usted quiere filtrar tráfico de todos los hosts en la subred 192.168.1.0/24.
  - La máscara de subred es 255.255.255.0
  - Para encontrar la máscara wildcard, tome lo inverso de la máscara de subred.
  - La máscara wildcard es 0.0.0.255.

Ejemplo de source-wildcard

- **En este ejemplo, usaremos todos los “1’s” en el último octeto de la máscara wildcard para cada prefijo-origen.**

  La máscara de subred para cada LAN tiene todos los “0’s” en el último octeto.

- **Un “1” significa ignorar esta posición de bit en la dirección IP origen del paquete.**




```
RTA#config t
RTA(config)#access-list 1 permit 192.168.1.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.3.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.5.0 0.0.0.255
```

La “Máscara Wildcard” a Nivel de Bit


```
RTA(config)#access-list 1 permit 192.168.1.0 0.0.0.255
```

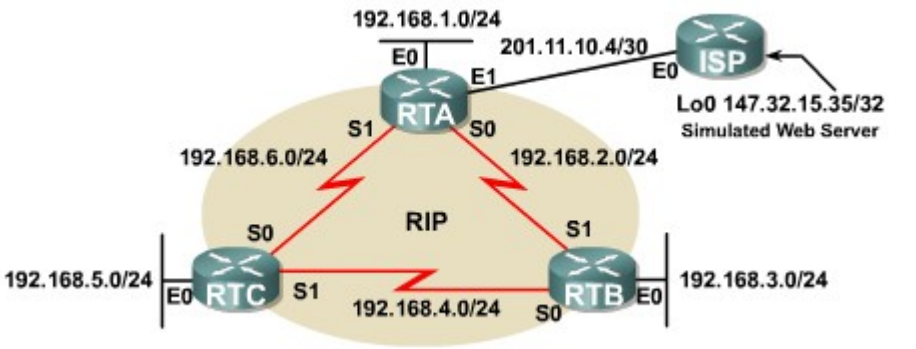
Source IP Address

Source Prefix	11000000.10101000.00000001.00000000
Wildcard Mask	00000000.00000000.00000000.11111111

- **El router lee la dirección IP origen de un paquete.**
  - **Para cada posición de bit, el router verifica la máscara wildcard.**
    - **Si hay un “0”, el router compara el valor del bit de la dirección IP Origen contra el valor del bit del Prefijo Origen para una “Coincidencia”.**
    - **Si hay un “1”, el router ignora aquella posición de bit.**

La Última Sentencia de la ACL: “Deny Any”

- **La última sentencia en todas las ACLs es un “deny any” implícito.**
- **Si un paquete no coincide con ninguna sentencia en la ACL, éste es negado.**




```
RTA#config t
RTA(config)#access-list 1 permit 192.168.1.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.3.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.5.0 0.0.0.255
!---Implied "Deny Any"
```

Aplicar la ACL

- **Una ACL no puede filtrar tráfico hasta que haya sido aplicada a una interface.**
  - **Esta es una característica de seguridad del IOS.**
    - **Usted puede escribir las sentencias de la ACL de forma segura sin que las sentencias afecten inmediatamente en el tráfico.**
- **La sintaxis del comando es la misma para las ACL de IP Estándar y Extendidas.**

Sintaxis para Aplicar ACLs de IP


```
router(config-if)#ip access-group {access-list-number|name} {in|out}
```

- Las ACLs son aplicadas a una interface
- El argumento *access-list-number* se refiere a la ACL escrita en configuración global.

  Use el argumento *name* para aplicar una ACL nombrada.
- El argumento `{in|out}` especifica en qué dirección la ACL deberá ser aplicada.

  Especificar `in` significa “filtra paquetes de entrada”.

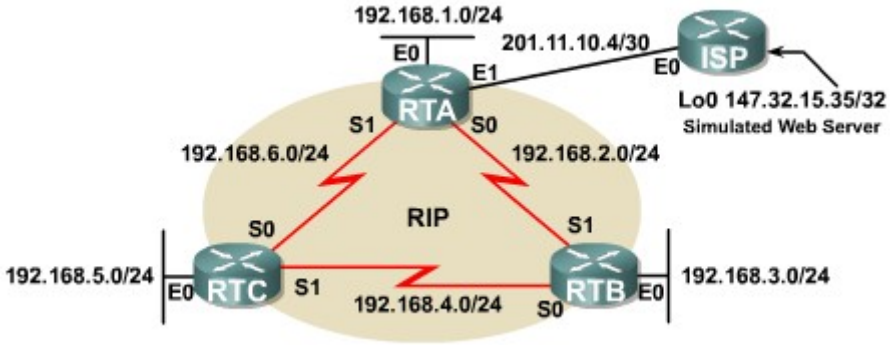
  Especificar `out` significa “filtra paquetes de salida”.

Ubicación Incorrecta de una ACL Estándar

- **ACLs Estándar no tienen un argumento destino.**

  Por lo tanto, ubique ACLs estándar tan cerca del destino como sea posible.

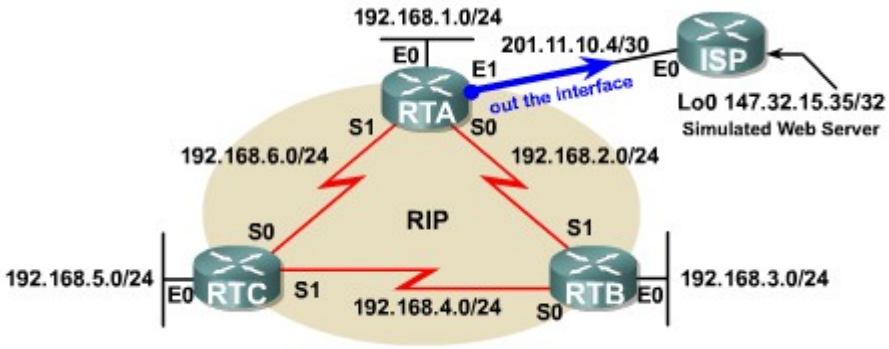
- **Para ver el porqué, puede preguntarse a sí mismo ¿qué le pasaría a todo el tráfico de IP de la LAN de RTA si la ACL fuera aplicada como se muestra?**




```
!--The following ACL is applied incorrectly
RTA#config t
RTA(config)#access-list 1 deny 192.168.1.0 0.0.0.255
RTA(config)#interface e0
RTA(config-if)#ip access-group 1 in
RTA(config-if)#end
RTA#
```

Ubicación Correcta de una ACL Estándar

- **En nuestro ejemplo, nosotros queremos permitir a todas las LANs acceso a Internet.**
- **Por lo tanto, aplicaremos la ACL a la interface E1 de RTA y se especificará “out” como la dirección.**

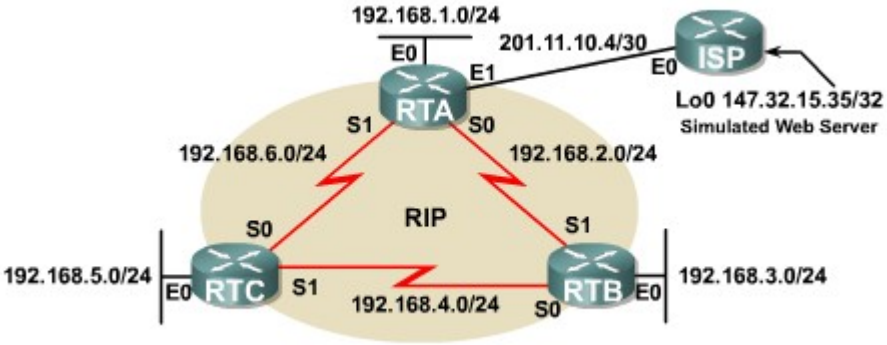



```
RTA#config t
RTA(config)#access-list 1 permit 192.168.0.0 0.0.7.255
RTA(config)#interface e1
RTA(config-if)#ip access-group 1 out
RTA(config-if)#end
RTA#
```

Primordial el “Deny Any” Implícito

- **Recuerde: La última sentencia que el router aplicará es un implícito “deny any”.**
- **¿Qué haría si usted quiere escribir una ACL que niegue específicos tipos de tráfico y permita todo lo demás?**
- **Ejemplo:**

  Niegue las LANs de RTB y RTC para acceder a la LAN de RTA, pero permita todo el otro tráfico.



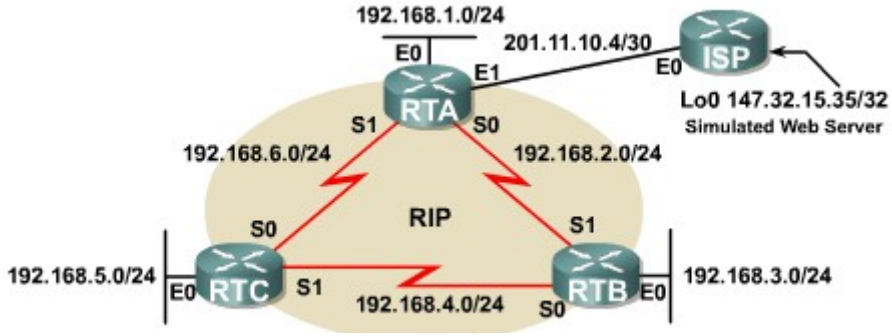
Escriba una Sentencia “permit any”

- **Primero, niegue tráfico de las dos LANs.**
- **Segundo, permita todo el otro tráfico.**

  `0.0.0.0` significa “cualquier dirección origen”.

  `255.255.255.255` significa “ignora todas las posiciones de bit”.

- **Tercero, aplique la ACL para filtar el tráfico de “salida” de E0.**




```
RTA#config t
RTA(config)#access-list 2 deny 192.168.3.0 0.0.0.255
RTA(config)#access-list 2 deny 192.168.5.0 0.0.0.255
!--Write a statement permitting all other traffic
RTA(config)#access-list 2 permit 0.0.0.0 255.255.255.255
RTA(config)#interface e0
RTA(config-if)#ip access-group 2 out
RTA(config-if)#end
RTA#
```

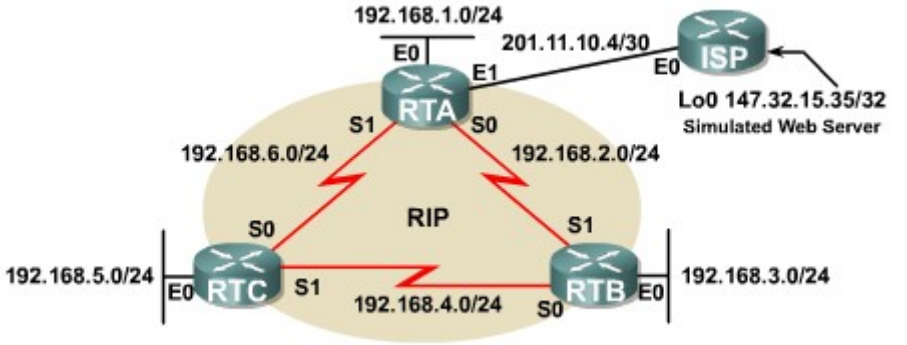
Sustituyendo 0.0.0.0 255.255.255.255

- **En lugar de escribir…**

```
permit 0.0.0.0
255.255.255.255
```

- **Escriba…**

```
permit any
```




```
RTA#config t
RTA(config)#access-list 2 deny 192.168.3.0 0.0.0.255
RTA(config)#access-list 2 deny 192.168.5.0 0.0.0.255
!--Substitute "0.0.0.0 255.255.255.255" with "any"
RTA(config)#access-list 2 permit any
RTA(config)#interface e0
RTA(config-if)#ip access-group 2 out
RTA(config-if)#end
RTA#
```

Filtrando Tráfico Desde un Simple Host

- **¿Qué haría si un usuario en particular está abusando de los privilegios de Internet?**

  ¿Cómo negaría aquella dirección IP del host, sin embargo, permitiendo a todos los demás?

  Recuerde: Una máscara wildcard le dice al router qué bits verifique.

Escribiendo una Sentencia “host”

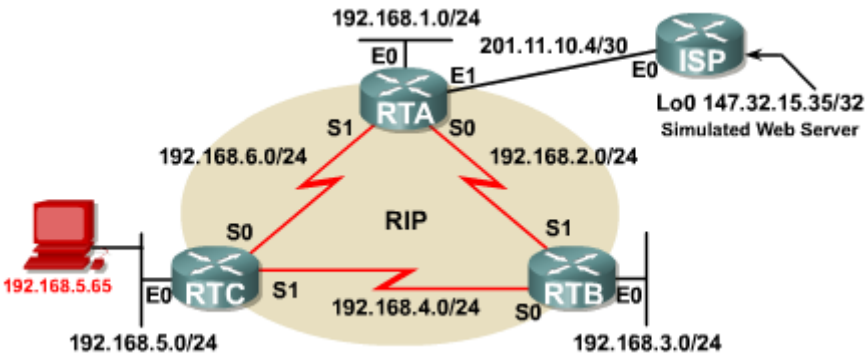
- **Ejemplo:**

  Deny 192.168.5.65.

  Source prefix es el origen de la dirección IP.

  La máscara Wildcard es “todos 0s”, significa que verifique cada posición de bit.

- **¿Porqué la sentencia deny es listada primero?**




```
RTA#config t
RTA(config)#access-list 1 deny 192.168.5.65 0.0.0.0
RTA(config)#access-list 1 permit 192.168.1.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.3.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.5.0 0.0.0.255
RTA(config)#interface e1
RTA(config-if)#ip access-group 1 out
RTA(config-if)#end
RTA#
```

Sustituyendo la Wildcard de Host, 0.0.0.0

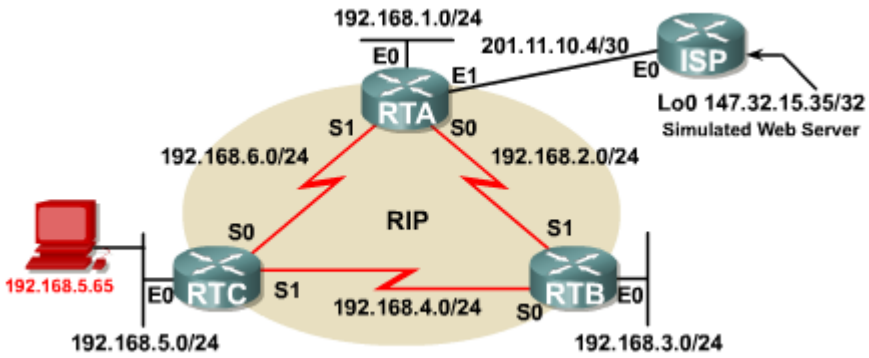
- En lugar de escribir…

```
deny
192.168.5.65
0.0.0.0
```

- Escriba…

```
deny host
192.168.5.65
```

- Note que la palabra clave `host` viene antes del prefijo origen.




```
RTA#config t
!--Substitute "0.0.0.0" with the keyword "host"
RTA(config)#access-list 1 deny host 192.168.5.65
RTA(config)#access-list 1 permit 192.168.1.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.3.0 0.0.0.255
RTA(config)#access-list 1 permit 192.168.5.0 0.0.0.255
RTA(config)#interface e1
RTA(config-if)#ip access-group 1 out
RTA(config-if)#end
RTA#
```

Configurando ACLs Extendidas

Resumen de ACL Extendidas

ACLs Extendidas controlan el tráfico comparando las direcciones origen y destino de los paquetes IP contra las direcciones configuradas en la ACL.

Usted puede controlar también tráfico basado en:

- **Protocolos IP de Capa 3**
- **Números de puerto TCP y UDP de Capa 4**

Sintaxis para Escribir una ACL Extendida


```
router(config)#access-list access-list-number
{permit|deny} protocol source-prefix source-
wildcard dest-prefix dest-wildcard [operator
[port]] [established]
```

- El argumento *access-list-number* puede ser un número entre 100 y 199
- El argumento *protocol* pueden ser varios como `ip`, `tcp`, `icmp`, y `rip`.
- Los argumentos destino tienen el mismo propósito que los argumentos origen.
- Opciones adicionales mostradas serán discutidas.

El argumento *protocol*

- El argumento ***protocol*** puede ser un número de opciones.
- Nos concentraremos en **ip**, **tcp**, **udp** e **icmp**.

```
RTA(config)#access-list 100 deny ?
  <0-255>  An IP protocol number
  ahp      Authentication Header Protocol
  eigrp    Cisco's EIGRP routing protocol
  esp      Encapsulation Security Payload
  gre      Cisco's GRE tunneling
  icmp     Internet Control Message Protocol
  igmp     Internet Gateway Message Protocol
  igrp     Cisco's IGRP routing protocol
  ip       Any Internet Protocol
  ipinip   IP in IP tunneling
  nos      KA9Q NOS compatible IP over IP tunneling
  ospf     OSPF routing protocol
  pcp      Payload Compression Protocol
```

Los Argumentos Destino

- **Los argumentos *dest-prefix* y *dest-wildcard* trabajan como los argumentos para el origen.**
- **ACLs Extendidas permiten filtrar basándose en la información de destino.**

  Dado que la información destino es incluida, usted puede ubicar la ACL tan cerca del origen como sea posible.

La opción *operator*

- La opción *operator* puede ser usada cuando el argumento *protocol* es tcp o udp.

```
!--"any" source and "any" destination
RTA(config)#access-list 100 permit tcp any any ?
  ack           Match on the ACK bit
  dscp          Match packets with given dscp value
  eq            Match only packets on a given port number
  established   Match established connections
  fin           Match on the FIN bit
  fragments     Check non-initial fragments
  gt            Match only packets with a greater port number
  log           Log matches against this entry
  log-input     Log matches against this entry, including input interface
  lt            Match only packets with a lower port number
  neq           Match only packets not on a given port number
  precedence    Match packets with given precedence value
```

La opción puerto

- La opción puerto es usada cuando la opción operador es una de lo siguiente:

  eq (equal to); gt (greater than); lt (less than); neg (not equal to); or range

- Usted puede filtrar tráfico de cualquier puerto TCP o UDP escribiendo el número de puerto o su correspondiente nombre.

```
RTA(config)#access-list 100 permit tcp any any eq ?
  <0-65535>    Port number
  bgp          Border Gateway Protocol (179)
  chargen      Character generator (19)
  cmd          Remote commands (rcmd, 514)
  daytime      Daytime (13)
  discard      Discard (9)
  domain       Domain Name Service (53)
  echo         Echo (7)
  exec         Exec (rsh, 512)
  finger       Finger (79)
  ftp          File Transfer Protocol (21)
  ftp-data     FTP data connections (20)
  gopher       Gopher (70)
```

La opción `established`

- **Agregando la palabra clave `established` a la sentencia de la ACL, usted está solicitando que la sesión TCP o UDP deberá ser establecida.**
- **Por ejemplo, permitir a los hosts detrás de su firewall establecer conexiones con host externos.**

  Host externos pueden enviar paquetes de regreso al origen solo si el origen inició la sesión.

Ubicación Correcta de las ACLs Extendidas

- Dado que una ACL extendida tiene información destino, deberá ubicarla tan cerca del origen como sea posible.
  - Esto reduce tráfico de red innecesario cuando un paquete será negado cuando alcance el destino.
- Ubicar una ACL extendida en la primera interface del router a la cual el paquete entre y especificarla como de entrada en el comando access-group.

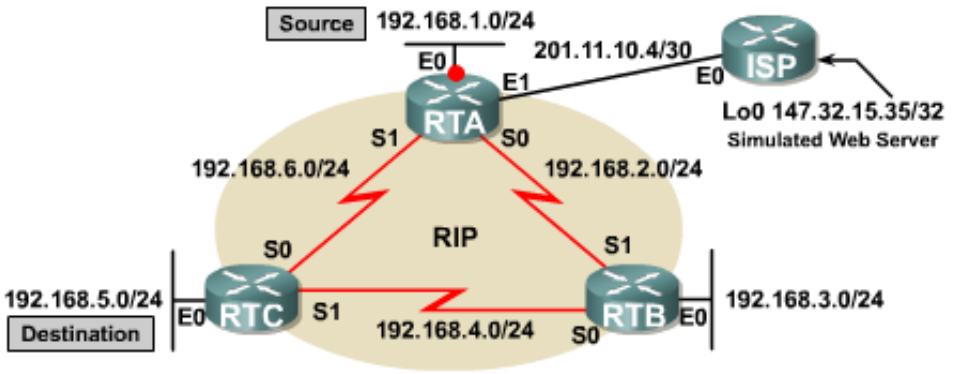
Ejemplo de Ubicación de una ACL Extendida

- **Negar acceso de la LAN de RTA a la LAN de RTB.**

  Podemos hacer esto con una ACL estándar, pero entonces la ACL tendría que ser ubicada en RTB.

  Una ACL estándar causaría que tráfico innecesario cruce un enlace WAN costoso.

- **Note que el "permit IP any any" permite todo el otro tráfico en la interface.**




```
RTA(config)#access-list 100 deny ip 192.168.1.0 0.0.0.255
            192.168.5.0 0.0.0.255
RTA(config)#access-list 100 permit ip any any
RTA(config)#interface e0
RTA(config-if)#ip access-group 100 in
```

ACLs Nombradas

- **Una característica importante en las versiones de IOS 11.2 y posteriores es la habilidad de nombrar a las ACLs. Las ventajas incluyen:**

  Nombres intuitivos que puedan posiblemente identificar el propósito de la ACL.

  ACLs no limitadas; las ACLs numeradas tienen un rango limitado.

  La habilidad de eliminar entradas sin tener que rescribir la ACL completa; nuevas líneas son agregadas al final de la lista, tal y como en las ACLs numeradas.

  Reduce la cantidad de escritura; no neceista escribir `access-list` y *`access-list-number`* para cada sentencia.

Sintaxis para Nombrar una ACL


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

- El prompt del Router cambia a modo de configuración de ACL.

  Ahora puede simplemente empezar cada sentencia con el argumento `permit` o `deny` .

```
RTB(config)#ip access-list standard MY_ACL1
RTB(config-std-nacl)#deny host 192.168.3.25
RTB(config-std-nacl)#permit 192.168.3.0 0.0.0.255
RTB(config-std-nacl)#exit
RTB(config)#interface e0
RTB(config-if)#ip access-group MY_ACL1 in
!
RTB(config)#ip access-list extended MY_ACL2
RTB(config-ext-nacl)#permit tcp any host 192.168.3.39 eq www
RTB(config-ext-nacl)#permit tcp any host 192.168.3.39 eq ftp
RTB(config-ext-nacl)#permit icmp any any
RTB(config-ext-nacl)#exit
RTB(config)#interface e0
RTB(config-if)#ip access-group MY_ACL2 out
```

Verificando las ACLs

- **`show access-lists`**

  muestra todas las access-lists configuradas en el router

- **`show [protocol] access-lists {name| number}`**

  muestra la lista de acceso identificada

- **`show ip interface`**

  muestra las access-lists aplicadas a la interface —entrada y salida.

- **`show running-config`**

  muestra todas las listas de acceso y en cuáles interfaces están aplicadas

Agregando Comentarios a las ACLs


```
Router(config)#access-list access-list number
                  remark remarks
Router(config-std-nacl)#remark remarks
Router(config-ext-nacl)#remark remarks
```

- El comando `remark` le permite hacer comentarios dentro de la configuración de la ACL para documentar la configuración activa.
- Los comentarios son desplegados cuando escriba el comando `show run`. Sin embargo, no verá los comentarios mostrados con el comando `show access-list`.

Ejemplo del Uso del Comando `remark`


```
RTA(config)#access-list 100 remark Deny Charlie web access
RTA(config)#access-list 100 deny tcp host 192.168.1.5 any eq 80
RTA(config)#access-list 100 permit ip any any

RTA#show access-list
Extended IP access list 100
    deny tcp host 192.168.1.35 any eq www
    permit ip any any

!--The "show access-list" command does NOT display remarks
!--Use "show run" to see remarks.

RTA#show run
<output omitted>
```

Reglas Básicas para una ACL

- **Una ACL por protocolo, por interface, por dirección. Escriba sus ACL's cuidadosamente, ya que tiene que incluir todo el tráfico que deberá ser filtrado de entrada o salida en una simple ACL!**
- **ACL's Estándar deberán ser aplicadas más cerca del destino. ACL's Extendidas deberán ser aplicadas más cerca del origen.**
- **Las sentencias son procesadas secuencialmente hasta que una coincidencia es encontrada, si ninguna coincidencia es encontrada entonces el paquete es negado ("deny any" implícito).**
- **Hosts específicos deberán ser filtrados primero, y grupos o filtros en general deberán ir al último.**

Reglas Básicas para una ACL

- **Nunca trabaje con una lista de acceso que esté activamente aplicada. Use un editor de texto primero.**
- **Líneas nuevas siempre son agregadas al final de la lista de acceso. No es posible seleccionar dónde agregar y remover líneas.**
- **Una lista de acceso de IP envía un mensaje ICMP de host inalcanzable al que envía sobre el paquete rechazado.**
- **Filtros de salida no afectan al tráfico originado por el router local.**


CISCO SYSTEMS
EMPOWERING THE
INTERNET GENERATION