Capítulo 9: Listas de Controle de Acesso

Roteamento e switching

Cisco | Networking Academy®
Mind Wide Open™

Capítulo 9

9.1 Fundamentos de operação de uma ACL

9.2 ACLs IPv4 do tipo padrão

9.3 ACLs IPv4 do tipo estendidas

9.4 Unidade de contexto: Depurar com ACLs

9.5 Identificar e Solucionar Problemas de ACL

9.6 Unidade de contexto: ACLs IPv6

9.7 Resumo do Capítulo

Capítulo 9: Objetivos

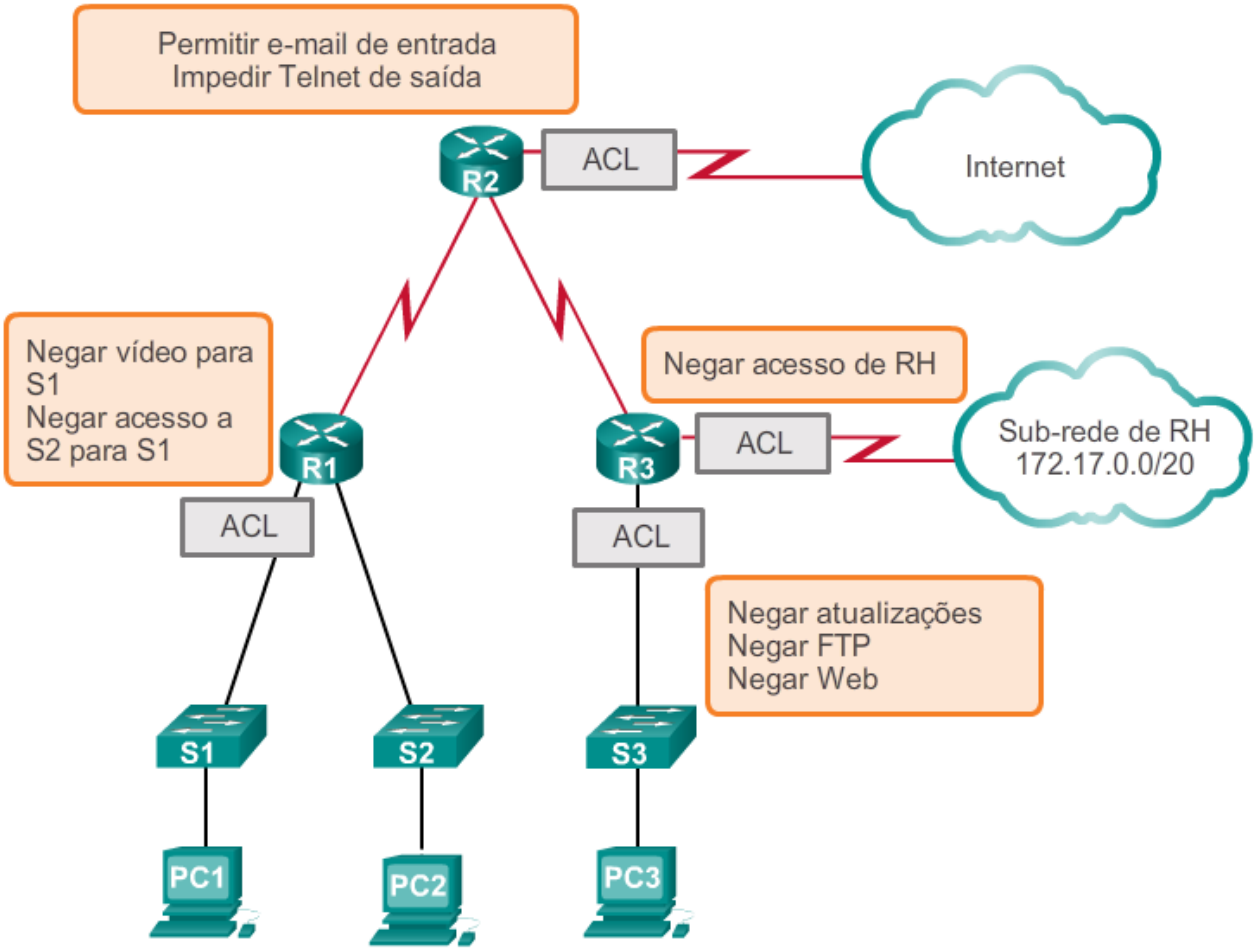
- Explique como as ACLs são usadas para filtrar o tráfego.
- Compare ACLs IPv4 padrão e estendidas.
- Explique como as ACLs usam máscaras curinga.
- Explique as diretrizes para criar ACLs.
- Explique as diretrizes de posicionamento das ACLs
- Configure ACLs IPv4 padrão para filtrar o tráfego de acordo com os requisitos de rede.
- Modifique uma ACL IPv4 padrão usando os números de sequência.
- Configure uma ACL padrão para proteger o acesso vty.

Capítulo 9: Objetivos (continuação)

- Explique a estrutura de uma entrada de controle de acesso estendida (ACE).
- Configure ACLs IPv4 estendidas para filtrar o tráfego de acordo com os requisitos de rede.
- Configure uma ACL para limitar a saída do debug.
- Explique como um roteador processa pacotes quando uma ACL é aplicada.
- Solucione erros comuns de ACLs usando comandos CLI.
- Compare a criação de ACL IPv4 e IPv6.
- Configure ACLs IPv6 para filtrar o tráfego de acordo com os requisitos de rede.

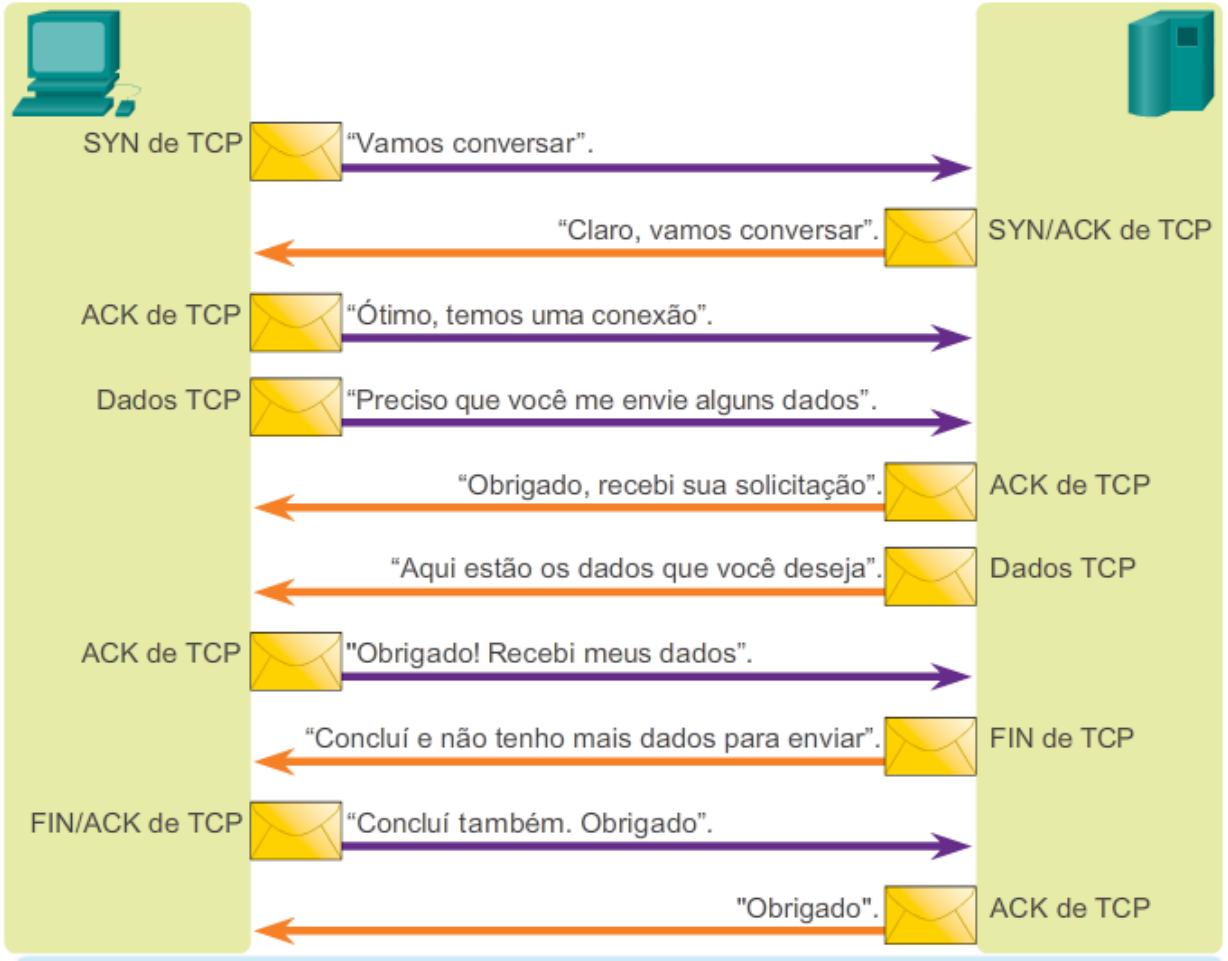
Finalidade das ACLs

O que é uma ACL?



Finalidade das ACLs

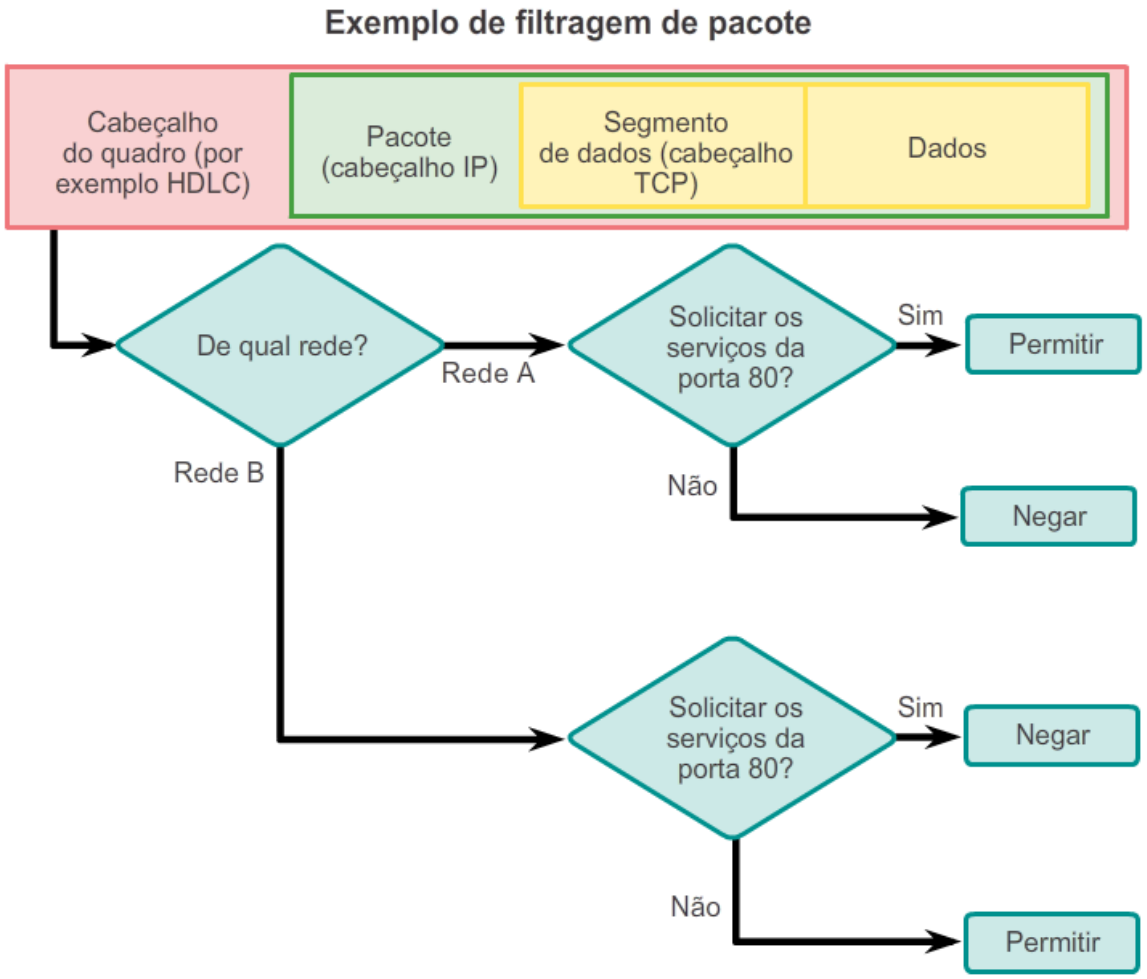
Uma conversa de TCP



Filtragem de pacotes

- A filtragem de pacote, às vezes chamada de filtragem de pacote estática, controla acesso a uma rede analisando os pacotes de entrada e saída e transmitindo-os ou eliminando-os com base em critérios, como o endereço IP de origem, o Endereço IP de destino e o protocolo transportado no pacote.
- Um Roteador atua como um filtro de pacote ao encaminhar ou recusar pacotes de acordo com as regras de filtragem.
- Uma ACL é uma lista sequencial de instruções de permissão ou de negação, conhecidas como entradas de controle de acesso (ACEs).

Filtragem de pacotes (continuação)



Operação de ACL

Uma ACL de entrada filtra pacotes que entram em uma interface específica, antes de eles serem roteados para a interface de saída.

Uma ACL de saída filtra pacotes após seu roteamento, independentemente da interface de entrada.

A última instrução de uma ACL é sempre uma deny implícito. Essa afirmativa é automaticamente inserida no final de cada ACL, mesmo que não esteja fisicamente presente. O deny implícito bloqueia todo o tráfego. Devido a esta negação implícita, uma ACL que não tenha pelo menos uma instrução de permissão bloqueará todo o tráfego.

Tipos de ACLs IPv4 da Cisco

ACLs padrão

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

As ACLs padrão filtram os pacotes IP com base apenas no endereço origem.

ACLs estendidas

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

As ACLs estendidas filtram os pacotes IP com base em vários recursos, incluindo:

- Endereços IP origem e destino
- Portas TCP e UDP origem e destino
- Tipo de protocolo/número do protocolo (exemplo: IP, ICMP, UDP, TCP etc.)

ACLs IPv4 padrão versus estendidas

Numerando e nomeando ACLs

ACL Numerada:

Atribua um número com base no protocolo a ser filtrado.

- (1 a 99) e (1300 e 1999): ACL de IP padrão
- (100 a 199) e (2000 a 2699): ACL de IP estendido

ACL Nomeada:

Atribua um nome para identificar a ACL.

- Os nomes podem conter caracteres alfanuméricos.
- Sugerimos que o nome seja escrito em LETRAS MAIÚSCULAS.
- Os nomes não podem conter espaços ou pontuação.
- É possível adicionar ou excluir entradas na ACL.

Introdução às máscaras curinga de ACL

As máscaras curinga e as máscaras de sub-rede diferem na maneira de corresponder ao binário 1s e 0s. As máscaras curinga utilizam as seguintes regras para corresponder ao binário 1s e 0s:

- Máscara curinga 0 - associada ao valor de bit correspondente no endereço.
- Máscara curinga 1 - ignora o valor de bit correspondente no endereço.

As máscaras curinga são normalmente chamadas de máscaras inversas. A razão é que, diferentemente de uma máscara de sub-rede em que binário 1 é igual a uma correspondência e o binário 0 não é uma correspondência, em uma máscara curinga ocorre o contrário isso.

Exemplos de máscara curinga: Hosts / sub-redes

Exemplo 1

	Decimal	Binário
Endereço IP	192.168.1.1	11000000.10101000.00000001.00000001
Máscara curinga	0.0.0.0	00000000.00000000.00000000.00000000
Resultado	192.168.1.1	11000000.10101000.00000001.00000001

Exemplo 2

	Decimal	Binário
Endereço IP	192.168.1.1	11000000.10101000.00000001.00000001
Máscara curinga	255.255.255.255	11111111.11111111.11111111.11111111
Resultado	0.0.0.0	00000000.00000000.00000000.00000000

Exemplo 3

	Decimal	Binário
Endereço IP	192.168.1.1	11000000.10101000.00000001.00000001
Máscara curinga	0.0.0.255	00000000.00000000.00000000.11111111
Resultado	192.168.1.0	11000000.10101000.00000001.00000000

Exemplos de máscara curinga: Intervalos de associação

Exemplo 1

	Decimal	Binário
Endereço IP	192.168.16.0	11000000.10101000.00010000.00000000
Máscara curinga	0.0.15.255	00000000.00000000.00001111.11111111
Intervalo de resultados	192.168.16.0 a 192.168.31.255	11000000.10101000.00010000.00000000 a 11000000.10101000.00011111.11111111

Exemplo 2

	Decimal	Binário
Endereço IP	192.168.1.0	11000000.10101000.00000001.00000000
Máscara curinga	0.0.254.255	00000000.00000000.11111110.11111111
Resultado	192.168.1.0	11000000.10101000.00000001.00000000
	Todas as sub-redes ímpares numeradas na rede principal de 192.168.0.0	

Calculando a máscara curinga

Calcular as máscaras curinga pode ser um desafio. Um método de atalho é subtrair a máscara de sub-rede de 255.255.255.255.

Exemplo 1

	255.255.255.255
–	255.255.255.000
	000.000.000.255

Exemplo 2

	255.255.255.255
–	255.255.255.240
	000.000.000.015

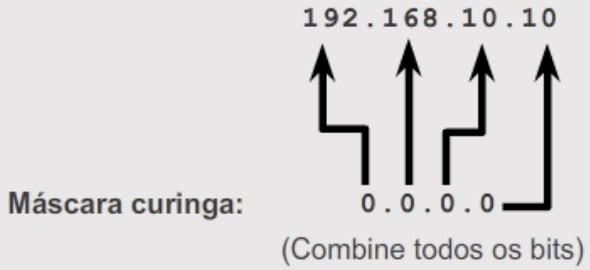
Exemplo 3

	255.255.255.255
–	255.255.254.000
	000.000.001.255

Palavras-chave de máscara curinga

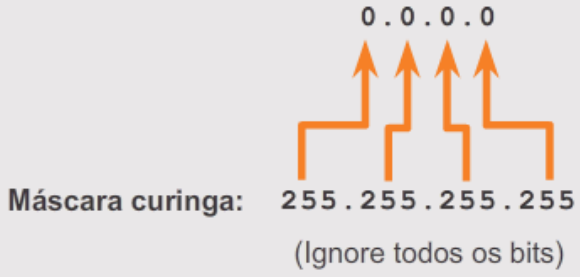
Exemplo 1

- 192.168.10.10 0.0.0.0 combina todos os bits do endereço
- Abrevie essa máscara curinga usando o endereço IP precedido pela palavra-chave **`host`** (**`host 192.168.10.10`**)



Exemplo 2

- 0.0.0.0 255.255.255.255 ignora todos os bits do endereço
- Abrevie a expressão com a palavra-chave **`any`**



Exemplos de palavras-chave de máscara curinga

Exemplo 1:

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

Exemplo 2:

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

Diretrizes gerais para criação de ACLs

- Use as ACLs em roteadores com firewall posicionados entre a rede interna e uma rede externa como a Internet.
- Use as ACLs em um roteador posicionado entre duas partes da rede para controlar o tráfego que chega ou sai de uma determinada parte da rede interna.
- Configure as ACLs em roteadores de borda, que são roteadores situados nas bordas de suas redes.
- Configure ACLs para cada protocolo de rede configurado nas interfaces do roteador de borda.

Diretrizes gerais para criação de ACLs

Os três Ps

- Uma ACL por protocolo - Para controlar o fluxo de tráfego em uma interface, deve-se definir uma ACL para cada protocolo ativado na interface.
- Uma ACL por direção - As ACLs controlam o tráfego em uma direção de cada vez em uma interface. Duas ACLs separadas devem ser criadas para controlar o tráfego de entrada e de saída.
- Uma ACL por interface - as ACLs controlam o tráfego de uma interface, por exemplo, GigabitEthernet 0/0.

Diretrizes para a criação de ACLs

Práticas Recomendadas de ACL

Diretiva	Benefício
Baseie suas ACLs na política de segurança da empresa.	Isso garantirá que você implemente as diretrizes de segurança da organização.
Prepare uma descrição do que você deseja que suas ACLs façam.	Isso o ajudará a evitar a criação inadvertida de potenciais problemas de acesso.
Use um editor de texto para criar, editar e salvar ACLs.	Isso o ajudará a criar uma biblioteca de ACLs reutilizáveis.
Teste suas ACLs em uma rede de desenvolvimento antes de sua implementação em uma rede de produção.	Isso o ajudará a evitar erros que podem custar caro.

Onde colocar ACLs

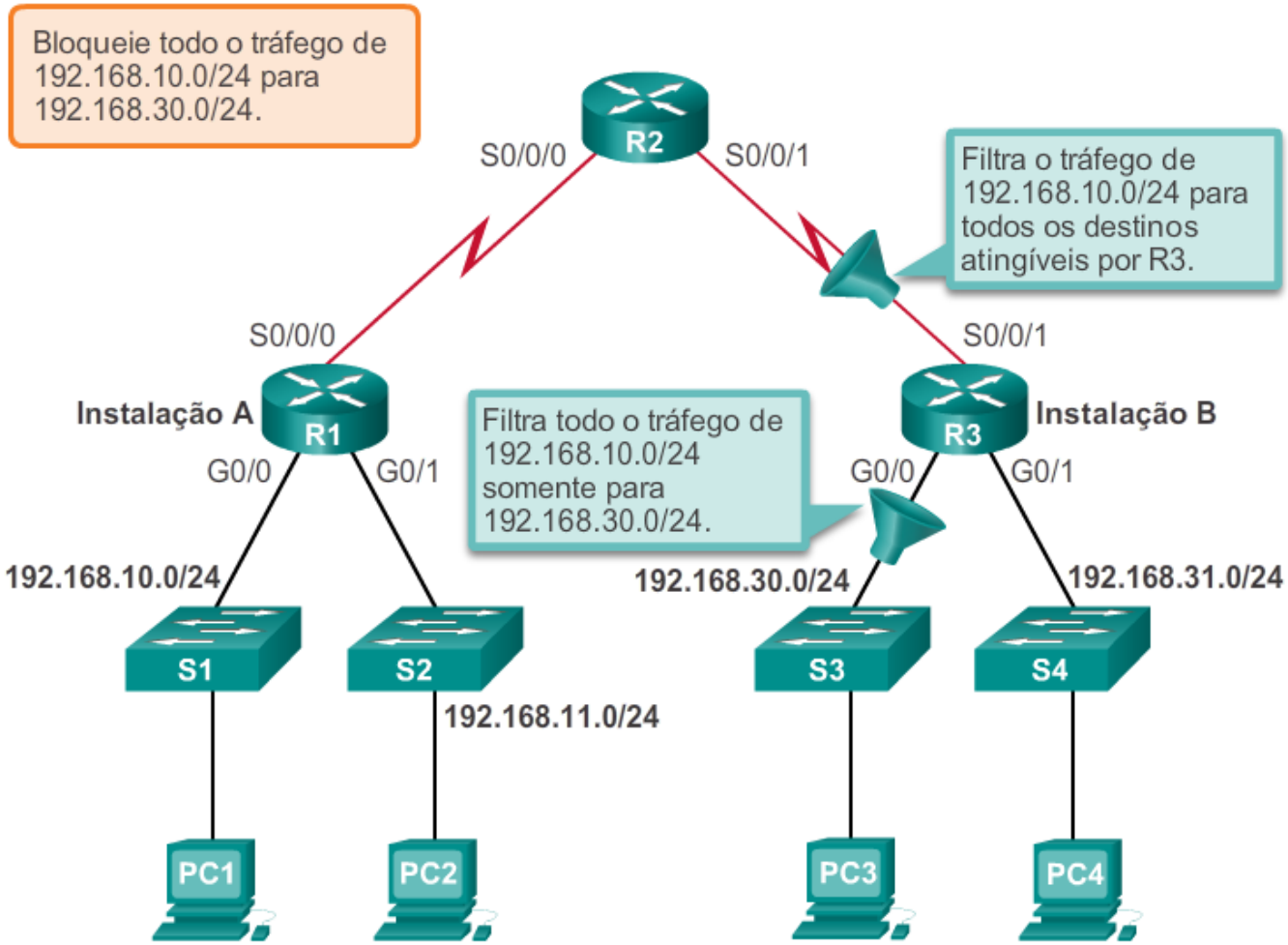
Cada ACL deve ser posicionada onde há maior impacto sobre o aumento da eficiência. As regras básicas são:

- ACLs estendidas: coloque ACLs estendidas o mais perto possível da origem de tráfego a ser filtrada.
- ACLs padrão: como as ACLs padrão não especificam endereços de destino, coloque-as o mais perto possível do destino.

O posicionamento das ACLs e, portanto, o tipo de ACL usado também pode depender do seguinte: extensão do controle do administrador de rede, largura de banda de redes envolvidas e facilidade de configuração.

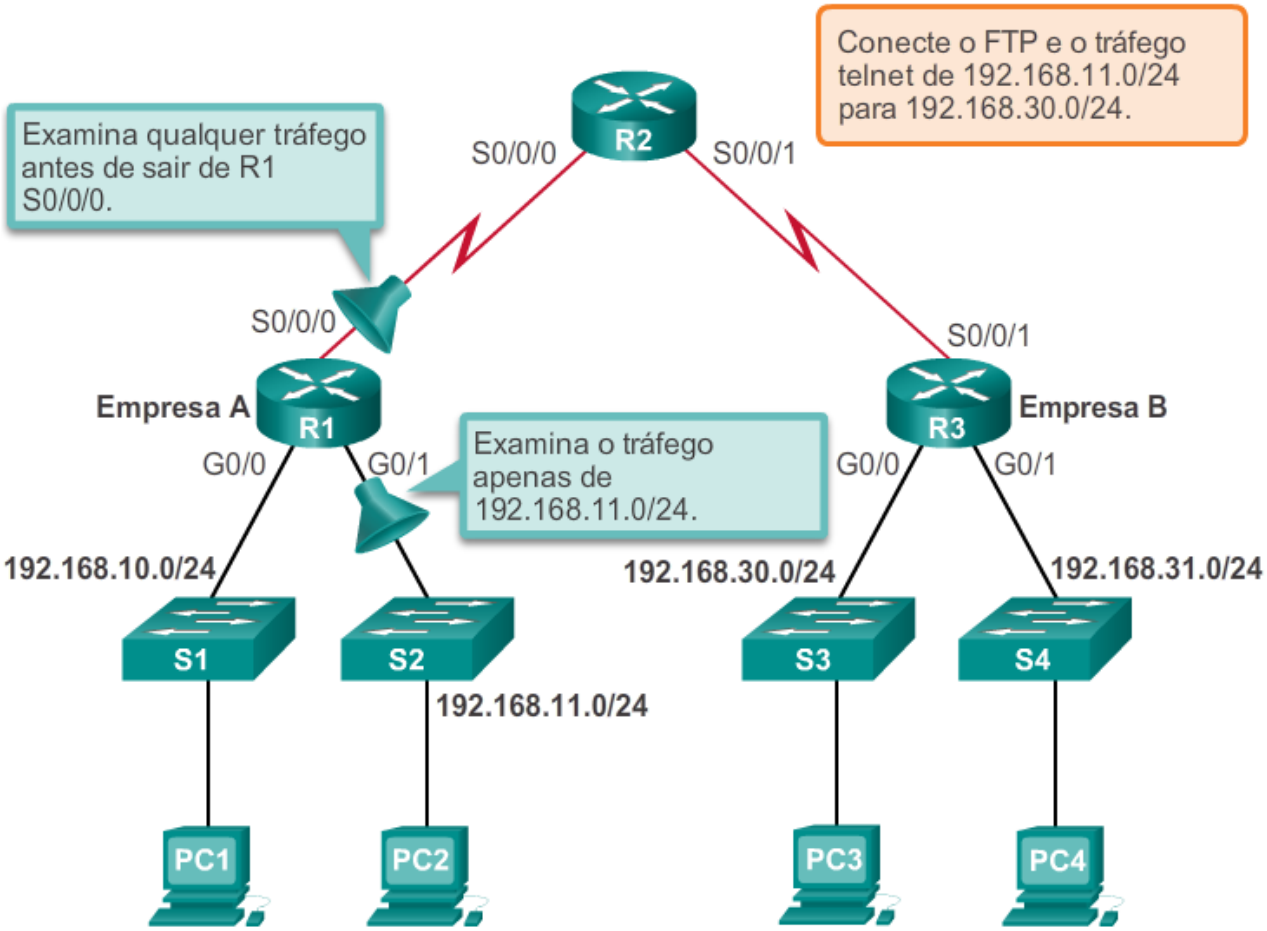
Diretrizes para a colocação da ACL

Posicionamento da ACL padrão



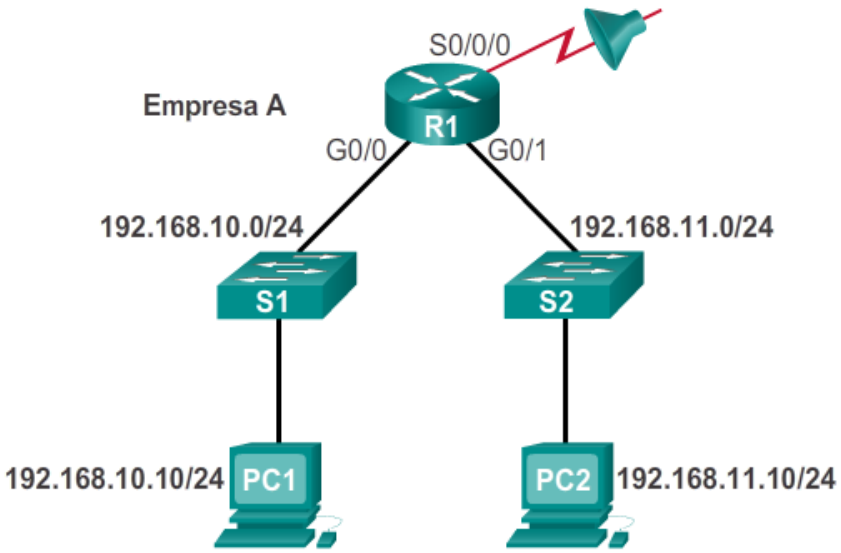
Diretrizes para a colocação da ACL

Posicionamento da ACL estendida



Configurar ACLs IPv4 padrão

Inserindo instruções de critérios



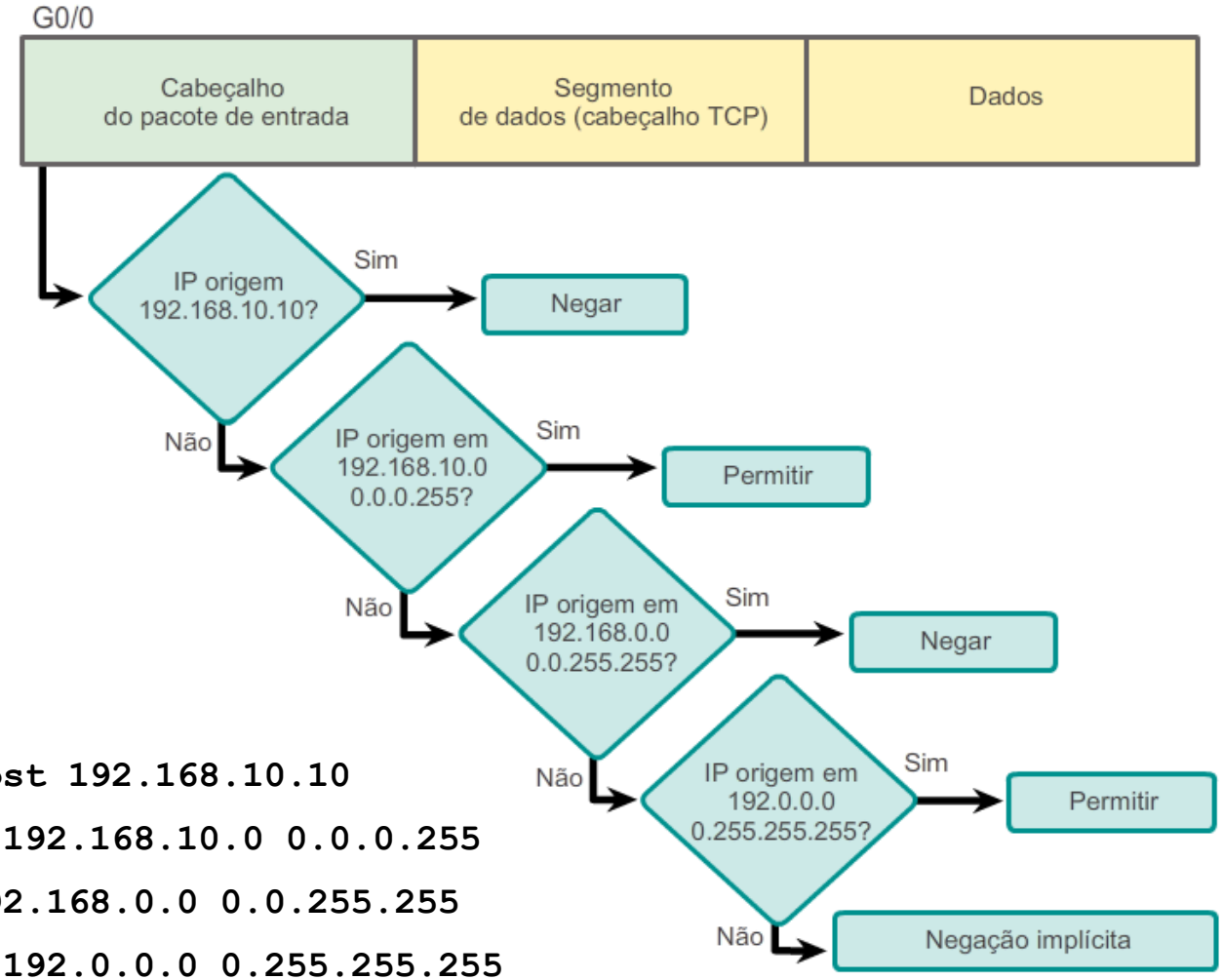
ACL 1

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

ACL 2

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

Configurando uma ACL padrão



Exemplo de ACL

- `access-list 2 deny host 192.168.10.10`
- `access-list 2 permit 192.168.10.0 0.0.0.255`
- `access-list 2 deny 192.168.0.0 0.0.255.255`
- `access-list 2 permit 192.0.0.0 0.255.255.255`

Configurando uma ACL padrão (continuação)

A sintaxe completa do comando para criar uma ACL padrão é a seguinte:

```
Router(config)# access-list access-list-number
deny permit remark source [ source-wildcard ] [
log ]
```

Para remover a ACL, o comando global configuration **`no access-list`** é usado.

A palavra-chave **`remark`** é usada para documentação e faz das listas de acesso um ótimo negócio e mais fácil de entender.

Lógica interna

- O IOS Cisco aplica uma lógica interna ao aceitar e ao processar instruções de lista de acesso padrão. Conforme discutido anteriormente, as instruções de lista de acesso são processadas sequencialmente. Portanto, a ordem em que as instruções são inseridas é importante.

```
R1(config)# access-list 3 deny 192.168.10.0 0.0.0.255
R1(config)# access-list 3 permit host 192.168.10.10
% Access rule can't be configured at higher sequence num as
it is part of the existing rule at sequence num 10
R1(config)#
```

ACL 3: A instrução de host está em conflito com a instrução de intervalo anterior.

Aplicando ACLs padrão às interfaces

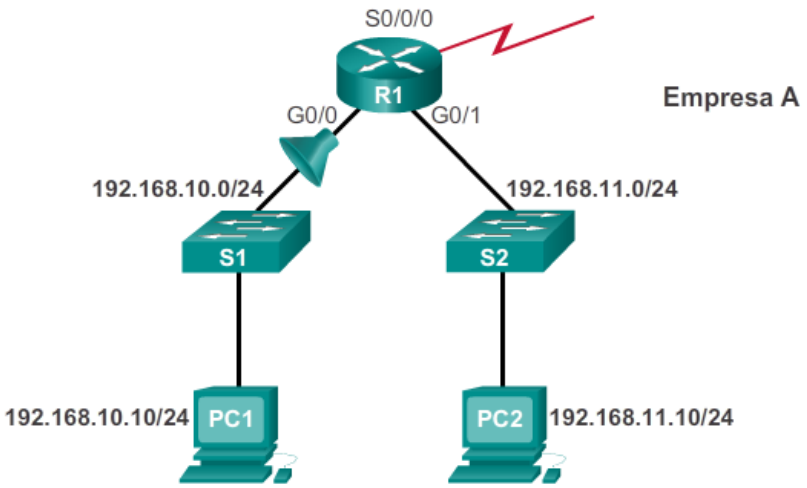
Após a configuração de uma ACL padrão, ela é vinculada a uma interface com o uso do comando **ip access-group** no modo de configuração de interface:

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

Para remover uma ACL de uma interface, primeiro insira o comando **no ip access-group** na interface e, em seguida, insira o comando global **no access-list** para remover toda a ACL.

Aplicando ACLs padrão às interfaces (continuação)

Negar um host específico




```
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
```

Configurar ACLs padrão do IPv4

Criando ACLs padrão nomeadas

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

A sequência alfanumérica do nome deve ser exclusiva e não pode começar com um número.

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

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

Ativa a ACL de IP nomeada em uma interface.

Configurar ACLs padrão do IPv4

Comentando ACLs

Exemplo 1: Comentários sobre uma ACL numerada

```
R1(config)# access-list 1 remark Do not allow Guest workstation
through
R1(config)# access-list 1 deny host 192.168.10.10
R1(config)# access-list 1 remark Allow devices from all other
192.168.x.x subnets
R1(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R1(config)# interface s0/0/0
R1(config-if)# ip access-group 1 out
R1(config-if)#
```

Exemplo 2: Comentários sobre uma ACL nomeada

```
R1(config)# ip access-list standard NO_ACCESS
R1(config-std-nacl)# remark Do not allow access from Lab
workstation
R1(config-std-nacl)# deny host 192.168.11.10
R1(config-std-nacl)# remark Allow access from all other networks
R1(config-std-nacl)# permit any
R1(config-std-nacl)# exit
R1(config-std-nacl)# interface G0/0
R1(config-if)# ip access-group NO_ACCESS out
R1(config-if)#
```

Modificar ACLs IPv4

Editando ACLs numeradas padrão

Edição de ACL numeradas usando um editor de texto

Configuração

```
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
```

Etapa 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
```

Etapa 2

```
<Editor de texto>
access-list 1 deny host 192.168.10.10
access-list 1 permit 192.168.0.0 0.0.255.255
```

Etapa 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
```

Etapa 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
```

Modificar ACLs IPv4

Editando ACLs numeradas padrão (continuação)

Edição de ACLs numeradas usando os números de sequência

Configuração

```
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
```

Etapa 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#
```

Etapa 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#
```

Etapa 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#
```

Modificar ACLs IPv4

Editando ACLs nomeadas padrão

Adição de uma linha a uma ACL nomeada

```
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#
```

Observação: o comando **no** *sequence-number* da ACL nomeada é usado para excluir instruções individuais.

Modificar ACLs IPv4

Verificando 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#
```

Estatísticas da 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#
```

Output after pinging PC3 from PC1.

As correspondências foram incrementadas.

```
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#
```

Números de sequência da ACL padrão

- Outra parte da lógica interna do IOS envolve o sequenciamento interno das instruções de ACL padrão. As instruções de intervalo que não têm três redes são configuradas inicialmente seguidas por cinco instruções do host. As instruções são todas válidas, porque o endereço IP do host não é parte das declarações inseridas anteriormente no intervalo.
- As instruções do host são listadas primeiro pelo comando show, mas não necessariamente na ordem em que foram inseridas. O IOS insere instruções do host em uma ordem usando uma função de hashing especial. A ordem resultante otimiza a busca por uma entrada da ACL de host.

Protegendo portas VTY com uma ACL padrão IPv4

Configurando uma ACL padrão para proteger uma porta VTY

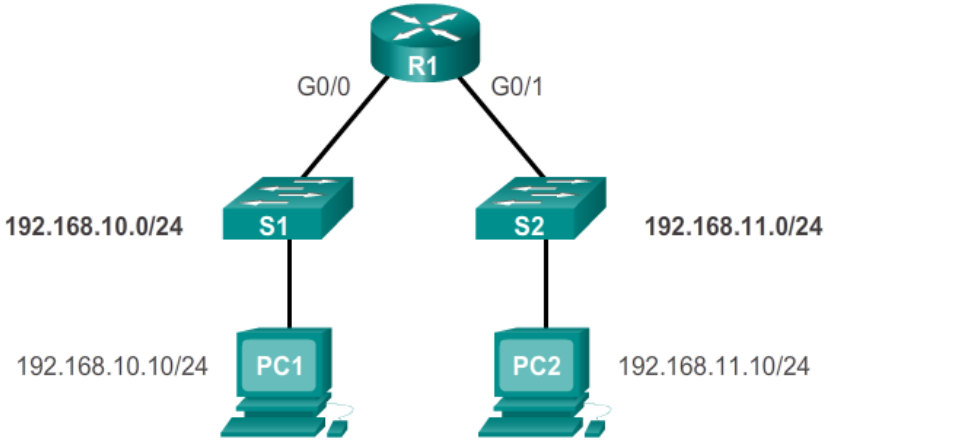
A filtragem do tráfego de Telnet ou SSH geralmente é considerada uma função de ACL IP estendida, pois ela filtra um protocolo de nível mais alto. No entanto, como o comando `the` **`access-class`** é usado para filtrar sessões de Telnet/SSH de entrada ou saída por endereço de origem, uma ACL padrão pode ser usada.

- `Router(config-line)#` **`access-class`** *`access-list-number`* `{` **`in`** `[` **`vrf-also`** `]` `|` **`out`** `}`

Protegendo portas VTY com uma ACL padrão IPv4

Verificando uma ACL padrão usada para proteger uma porta VTY

```
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>
```

Estrutura de uma ACL de IPv4 estendida

ACLs estendidas

ACLs estendidas podem filtrar baseadas em:

- Endereço origem
- Endereço destino
- Protocolo
- Números de portas

Estrutura de uma ACL de IPv4 estendida

ACLs estendidas (continuação)

Utilização de números de porta

```
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
```

Utilização de palavras-chave

```
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
```

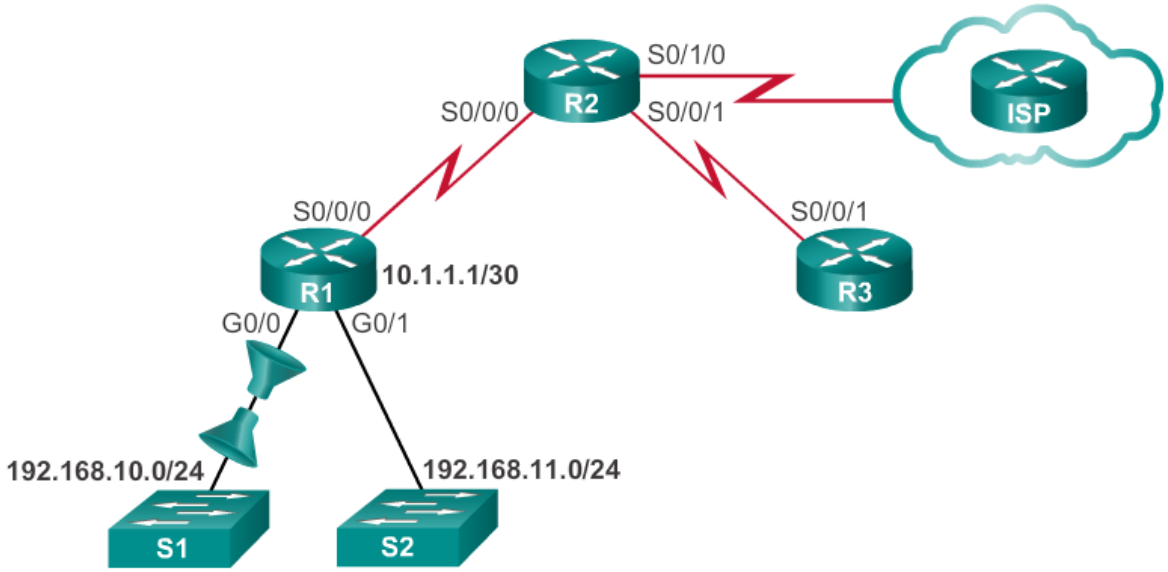
Configurando ACLs estendidas

As etapas de procedimentos para configurar ACLs estendidas são as mesmas usadas para ACL padrão. A ACL estendida é configurado em primeiro lugar, e é ativada em uma interface. No entanto, a sintaxe de comandos e os parâmetros são mais complexos para suportar recursos adicionais fornecidos pelas ACLs estendidas.

```
access-list access-list-number {deny | permit | remark}
protocol source [source-wildcard] [operator operand]
[port port-number or name] destination [destination-wildcard]
[operator operand] [port port-number or name][established]
```

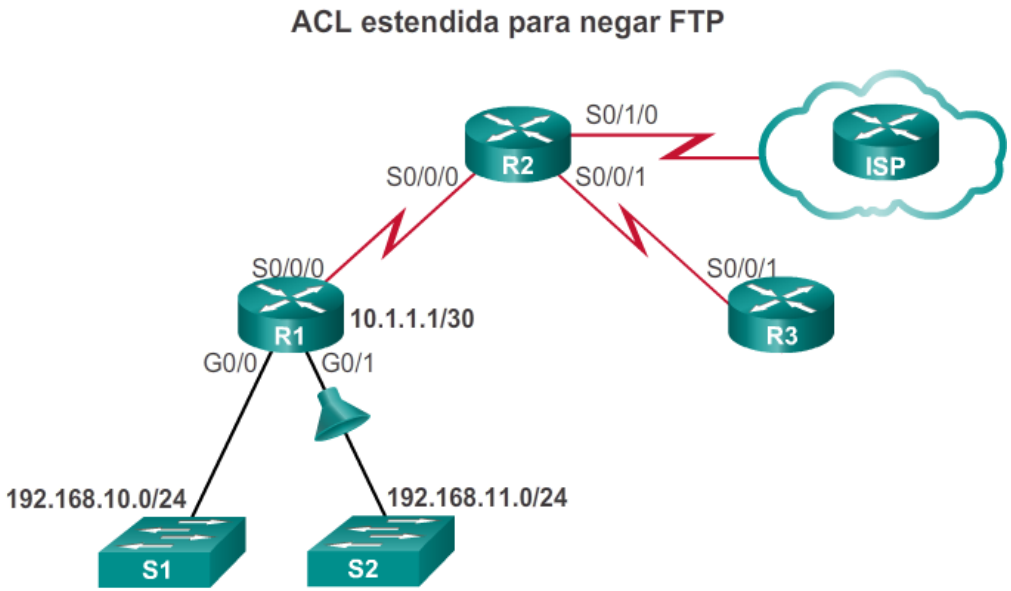
Configurar ACLs IPv4 estendidas

Aplicando ACLs estendidas às interfaces




```
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
R1(config)#interface g0/0
R1(config-if)#ip access-group 103 in
R1(config-if)#ip access-group 104 out
```

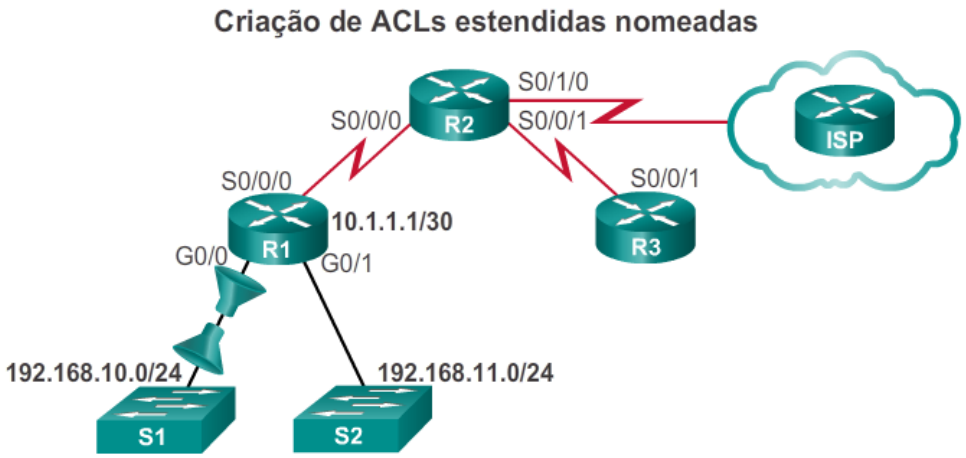
Filtrando tráfego com ACLs estendidas




```
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
```

Configurar ACLs IPv4 estendidas

Criando ACLs nomeadas estendidas




```
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
```

Verificando ACLs estendidas

```
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
<saída omitida>
  Outgoing access list is BROWSING
  Inbound  access list is SURFING
<saída omitida>
```

Editando ACLs estendidas

É possível editar uma ACL estendida usando o mesmo processo para editar uma ACL padrão. Uma ACL estendida pode ser modificada com o uso de:

- Método 1 - Editor de texto
- Método 2 – Números de sequência

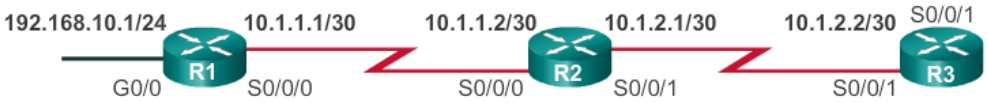
Limitando a saída de depuração

Objetivo da limitação da depuração da saída com ACLs

- Os comandos de depuração são as ferramentas usadas para ajudar a verificar, identificar e solucionar problemas de operações de rede.
- Quando algumas opções de depuração são usadas, a saída pode exibir muito mais informações do que o necessário ou pode ser facilmente visualizada.
- Em uma rede de produção, a quantidade de informações fornecida por comandos de depuração pode ser confusa e pode causar interrupções de rede.
- Alguns comandos de depuração podem ser combinados com uma lista de acesso para limitar a saída de modo que somente as informações necessárias para verificação ou resolução de um problema específico sejam exibidas.

Configurando ACLs para limitar a saída de depuração

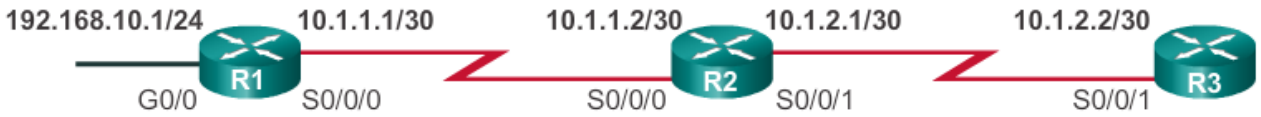
O administrador de R2 deseja verificar se o tráfego está sendo devidamente roteada usando **`debug ip packet`**. Para limitar a saída da depuração para incluir somente tráfego ICMP entre R1 e de R3, a ACL 101 será aplicada.




```
R2(config)#ip access-list extended 101
R2(config-ext-nacl)#permit icmp host 10.1.1.1 host 10.1.2.2
R2(config-ext-nacl)#permit icmp host 10.1.2.2 host 10.1.1.1
R2(config-ext-nacl)#exit
R2(config)#interface s0/0/0
R2(config-if)#no ip route-cache
R2(config-if)#exit
R2(config)#interface s0/0/1
R2(config-if)#no ip route-cache
R2(config-if)#end
R2#
R2#debug ip packet 101
IP packet debugging is on for access list 101
R2#
```

Limitando a saída de depuração

Verificando ACLs que limitam a saída de depuração




```
R2# debug ip packet 101
IP packet debugging is on for access list 101
R2#

<ping 10.1.2.2 command entered on R1>

*Jan 25 20:49:26.910: IP: s=10.1.1.1 (Serial0/0/0), d=10.1.2.2
(Serial0/0/1), g=10.1.2.2, len 100, forward
*Jan 25 20:49:26.910: IP: s=10.1.1.1 (Serial0/0/0), d=10.1.2.2
(Serial0/0/1), len 100, sending full packet
*Jan 25 20:49:26.938: IP: s=10.1.2.2 (Serial0/0/1), d=10.1.1.1
(Serial0/0/0), g=10.1.1.1, len 100, forward
*Jan 25 20:49:26.938: IP: s=10.1.2.2 (Serial0/0/1), d=10.1.1.1
(Serial0/0/0), len 100, sending full packet

<output omitted>
```

Lógica da ACL de entrada

- Os pacotes são testados em relação a uma ACL de entrada, se houver, antes de serem roteados.
- Se um pacote de entrada corresponder a uma instrução da ACL com uma permissão, ele será enviado para ser roteado.
- Se um pacote de entrada corresponder a uma instrução da ACL com um deny, ele será descartado e não roteado.
- Se um pacote de entrada não encontrar nenhuma instrução da ACL, ele será “negado implicitamente” e descartado sem ser roteado.

Lógica da ACL de saída

- Os pacotes são verificados primeiro para uma rota antes de serem enviados a uma interface de saída. Se não houver uma rota, os pacotes serão descartados.
- Se uma interface de saída não tiver ACLs, os pacotes serão encaminhados diretamente a essa interface.
- Se houver uma ACL na interface de saída, ela será testada antes de ser enviada para essa interface.
- Se um pacote de saída corresponder a uma instrução da ACL com uma permissão, ele será enviado para a interface.

Lógica da ACL de saída (continuação)

- Se um pacote de saída corresponder a uma instrução da ACL com um deny, ele será descartado.
- Se um pacote de saída não encontrar nenhuma instrução da ACL, ele será “negado implicitamente” e descartado.

Operações de lógica da ACL

- Quando um pacote chega a uma interface de roteador, o processo de Roteador é o mesmo, sendo as ACLs usadas ou não. À medida que um quadro entra em uma interface, o roteador verifica se o endereço da Camada 2 de destino corresponde ao endereço da Camada 2 da interface ou se o quadro é um quadro de broadcast.
- Se o endereço do quadro for aceito, as informações do quadro são removidas, e o roteador verifica se há uma ACL na interface de entrada. Se existir uma ACL, o pacote é testado em relação às instruções da lista.

Operações de lógica da ACL(continuação)

- Se o pacote for aceito, ele será testado em relação às entradas da tabela de roteamento para determinar a interface de destino. Se existir uma entrada de tabela de roteamento para o destino, o pacote será encaminhado para a interface de saída, se não, o pacote será descartado.
- Depois, o roteador verifica se a interface de saída tem uma ACL. Se existir uma ACL, o pacote é testado em relação às instruções da lista.
- Se não houver uma ACL ou se o pacote for permitido, o pacote será encapsulado no novo protocolo da camada 2 e encaminhado através da interface para o próximo dispositivo.

Processo de decisão de ACL padrão

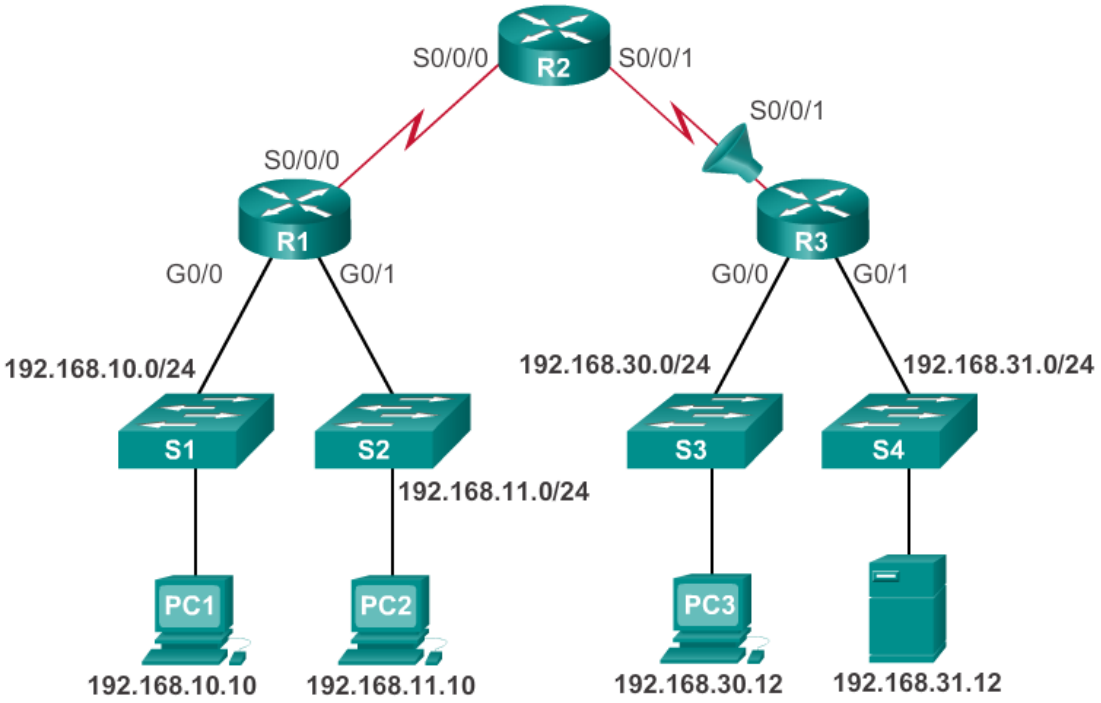
- As ACLs padrão examinam apenas o endereço de origem de IPv4. O destino do pacote e as portas envolvidas não são considerados.
- O software CISCO IOS testa endereços com relação às condições da ACL, uma a uma. A primeira combinação determina se o software aceita ou rejeita o endereço. Como o software interrompe o teste das condições depois da primeira correspondência, a ordem das condições é crítica. Se nenhuma condição for correspondente, o endereço será descartado.

Processo de decisão de ACL estendida

- A ACL primeiro filtra no endereço de origem, depois na porta e no protocolo de origem. Ele, então, filtra no endereço destino, na porta e no protocolo de destino e toma uma decisão final de permissão ou recusa.

Solucionando erros comuns de ACLs - Exemplo 1

O host 192.168.10.10 não tem nenhuma conectividade com 192.168.30.12.

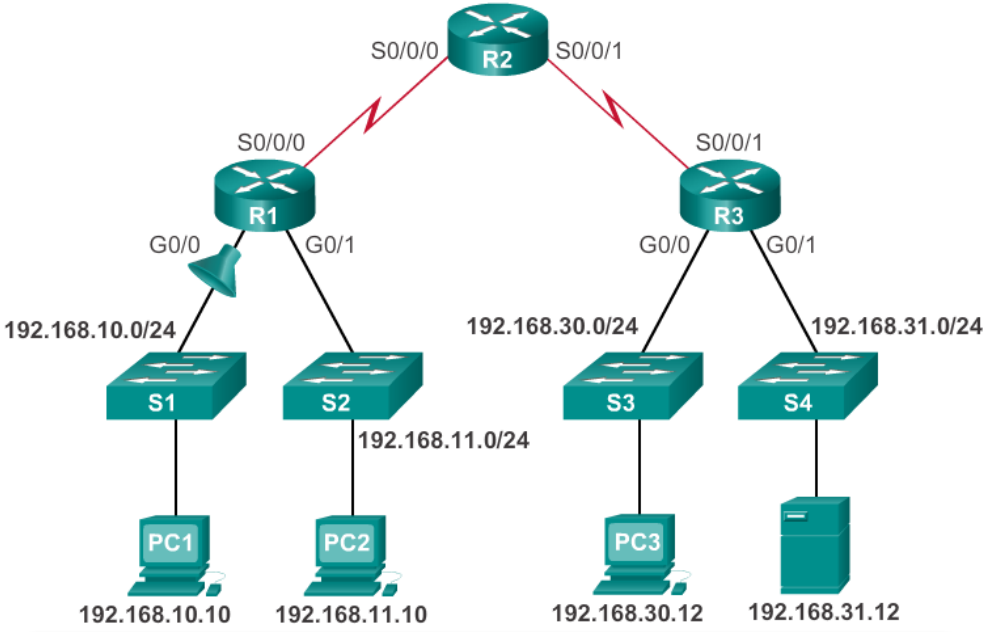



```
R3#show access-lists
Extended IP access list 110
    10 deny tcp 192.168.10.0 0.0.0.255 any (12 match(es))
    20 permit tcp 192.168.10.0 0.0.0.255 any eq telnet
    30 permit ip any any
```

Erros comuns de ACLs

Solucionando erros comuns de ACLs – Exemplo 2

A rede 192.168.10.0 /24 não pode usar TFTP para se conectar com a rede 192.168.30.0 /24.

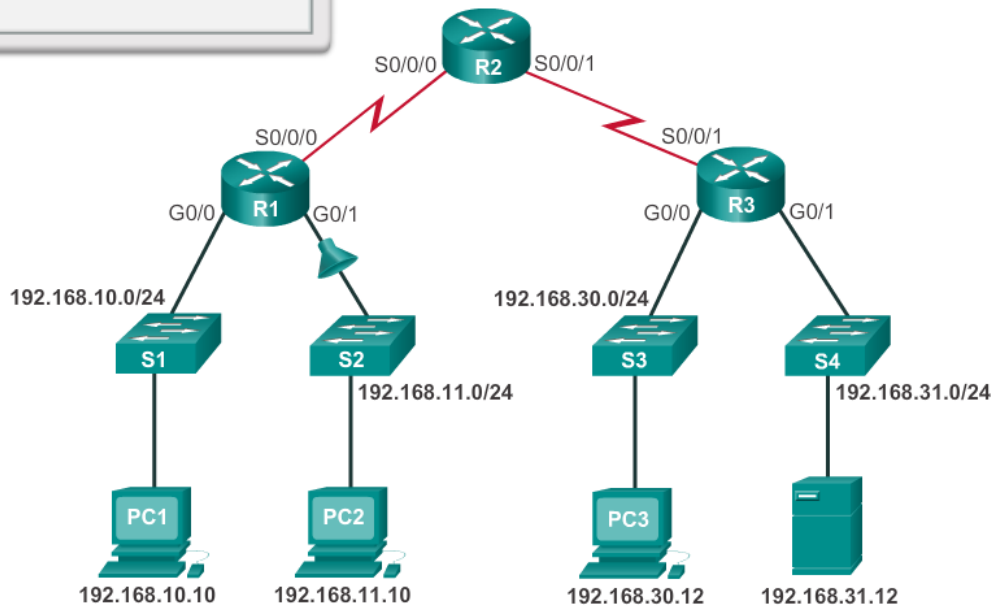



```
R1#show access-lists 120
Extended IP access list 120
    10 deny tcp 192.168.10.0 0.0.0.255 any eq telnet
    20 deny tcp 192.168.10.0 0.0.0.255 host 192.168.31.12 eq smtp
    30 permit tcp any any
```

Solucionando erros comuns de ACLs – Exemplo 3

A rede 192.168.11.0 /24 pode usar Telnet para se conectar a 192.168.30.0 /24, mas de acordo com a política da empresa, essa conexão não deve ser negada.

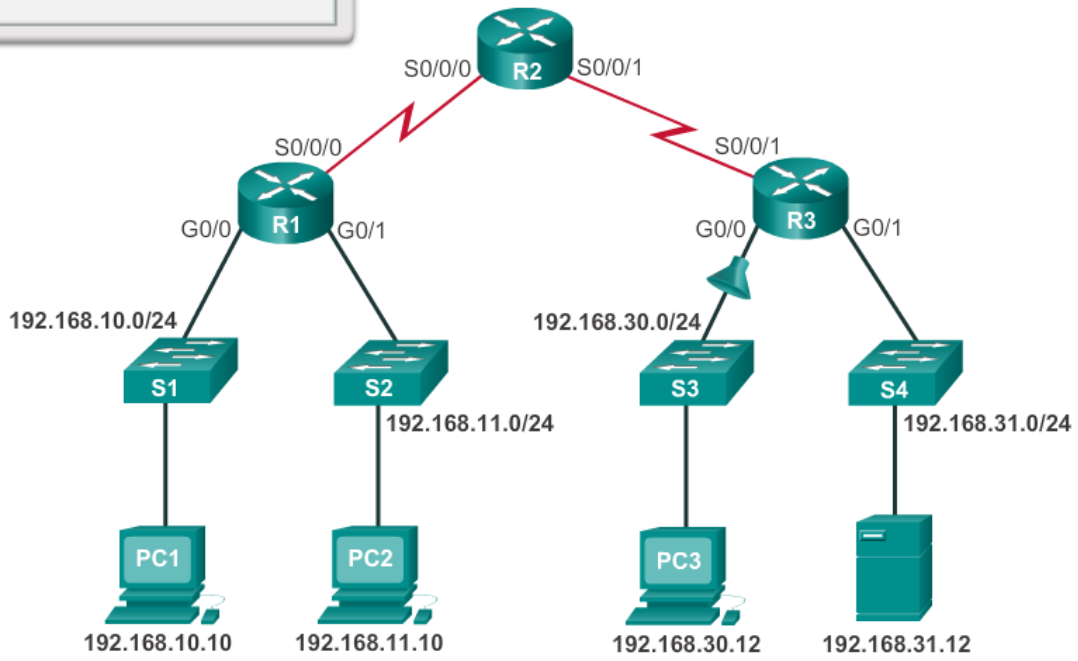
```
R1#show access-lists 130
Extended IP access list 130
    10 deny tcp any eq telnet any
    20 deny tcp 192.168.11.0 0.0.0.255 host 192.168.31.12 eq smtp
    30 permit tcp any any (12 match(es))
```



Solucionando erros comuns de ACLs – Exemplo 4

O host 192.168.30.12 pode executar Telnet para se conectar a 192.168.31.12, mas a política da empresa declara que essa conexão não deve ser permitida.

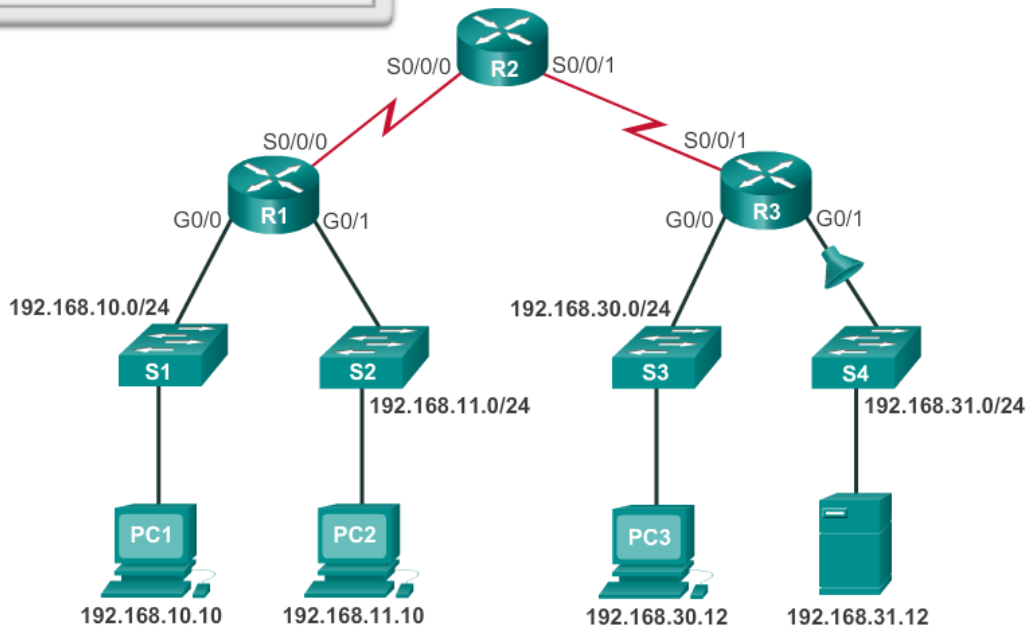
```
R3#show access-lists 140
Extended IP access list 140
    10 deny tcp host 192.168.30.1 any eq telnet
    20 permit ip any any (5 match(es))
```



Solucionando erros comuns de ACLs – Exemplo 5

O host 192.168.30.12 pode usar o Telnet para se conectar a 192.168.31.12, mas de acordo com a política de segurança, essa conexão não deve ser permitida.

```
R2#show access-lists 150
Extended IP access list 150
    10 deny tcp any host 192.168.31.12 eq telnet
    20 permit ip any any
```



Tipo de ACLs IPv6

IPv4 ACLs

- Padrão
  - Numerada
  - Nominal
- Estendida
  - Numerada
  - Nominal

ACLs IPv6

- Somente nomeadas
- Semelhante em funcionalidade à ACL estendida IPv4

Comparando ACLs IPv4 e IPv6

Embora as ACLs IPv4 e IPv6 sejam muito semelhantes, há três diferenças importantes entre elas.

- Aplicando ACLs IPv6

  O IPv6 usa o comando **`ipv6 traffic-filter`** para executar a mesma função para interfaces IPv6.

- Nenhuma máscara curinga

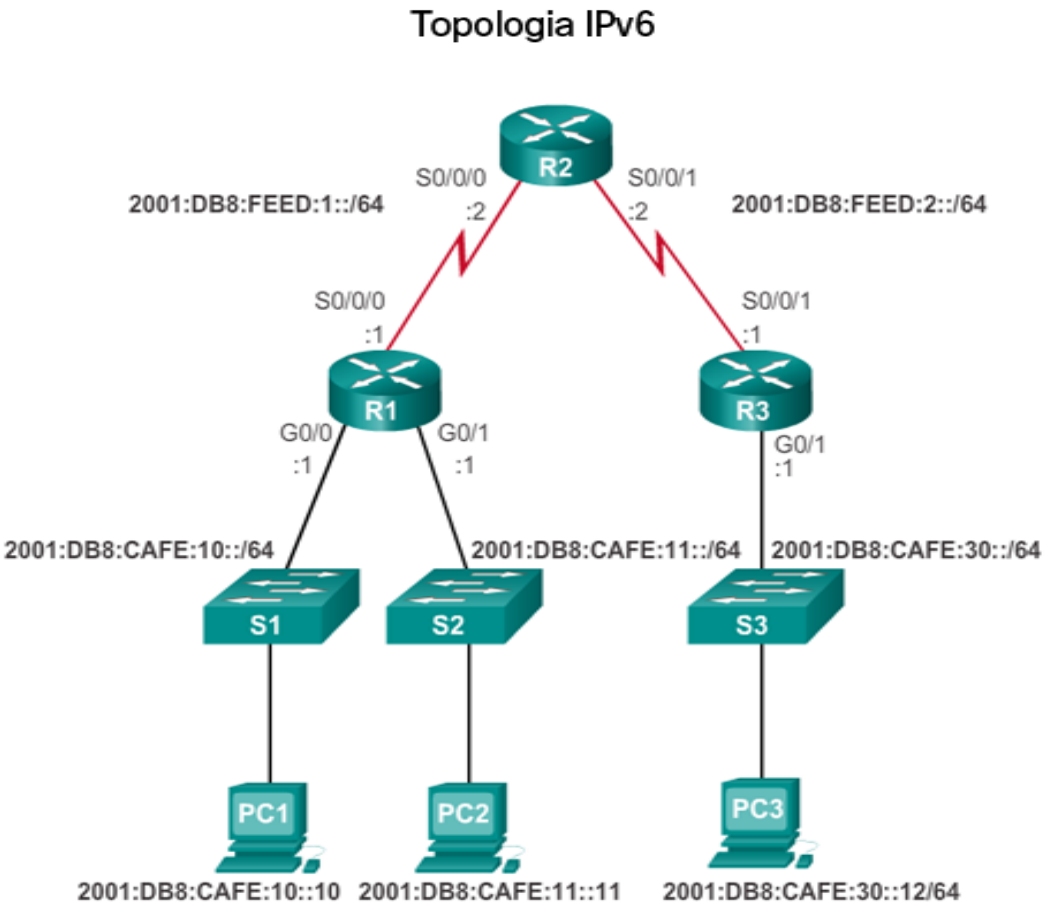
  O prefix-length é usado para indicar quanto de uma fonte ou de um endereço destino de IPv6 deve ser combinado.

- Instruções adicionais padrão

```
permit icmp any any nd-na
permit icmp any any nd-ns
```

Configurando ACLs IPv6

Configurando a topologia IPv6



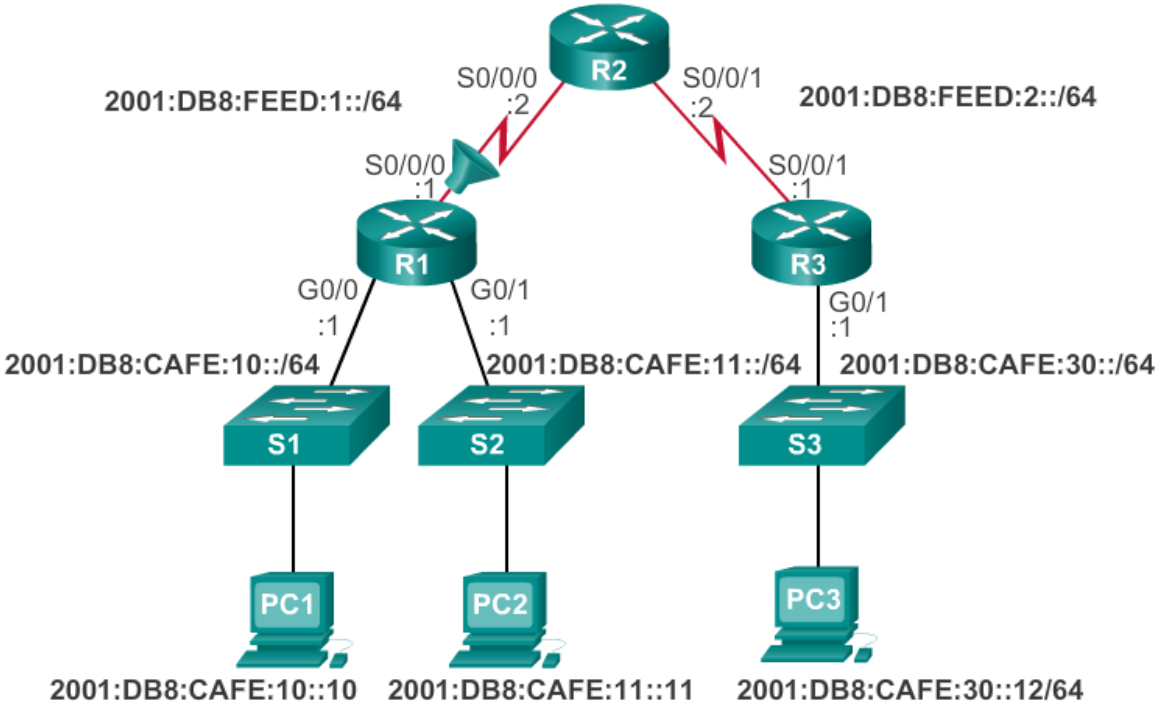
Configurando ACLs IPv6

Há três etapas básicas para configurar ACLs IPv6:

- No modo de configuração global, use o comando **ipv6 access-list** *name* para criar uma ACL IPv6.
- No modo de configuração ACL com nome, use as instruções **permit** ou **deny** para especificar uma ou mais condições para determinar se um pacote é encaminhado ou descartado.
- Retorne ao modo EXEC privilegiado com o comando **end**.

```
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]]
```

Aplicando uma ACL IPv6 ACL a uma interface




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

Exemplos de ACL IPv6

Negar FTP

```
R1(config)#ipv6 access-list NO-FTP-TO-11
R1(config-ipv6-acl)#deny tcp any 2001:db8:cafe:11::/64 eq ftp
R1(config-ipv6-acl)#deny tcp any 2001:db8:cafe:11::/64 eq ftp-data
R1(config-ipv6-acl)#permit ipv6 any any
R1(config-ipv6-acl)#exit
R1(config)#interface g0/0
R1(config-if)#ipv6 traffic-filter NO-FTP-TO-11 in
R1(config-if)#
```

Restringir acesso

```
R3(config)#ipv6 access-list RETRICTED-ACCESS
R3(config-ipv6-acl)#remark Permit access only HTTP and HTTPS to Network 1
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   1
R3(config-ipv6-acl)#remark Deny all other traffic to Network 10
R3(config-ipv6-acl)#deny ipv6 any 2001:db8:cafe:10::/64   2
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:CA
R3(config-ipv6-acl)#remark Deny telnet access to PC2 for all other device
R3(config-ipv6-acl)#deny tcp any host 2001:db8:cafe:11::11 eq 23   4
R3(config-ipv6-acl)#remark Permit access to everything else
R3(config-ipv6-acl)#permit ipv6 any any   5
R3(config-ipv6-acl)#exit
R3(config)#interface g0/0
R3(config-if)#ipv6 traffic-filter RESTRICTED-ACCESS in   6
R3(config-if)#
```

Configurando ACLs IPv6

Verificando ACLs IPv6

```
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
```

<some output omitted for brevity>

```
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#
```

Capítulo 9: Resumo

- Por padrão um Roteador não filtra o tráfego. O tráfego que entra no Roteador é instalado somente com base nas informações na tabela de roteamento.
- A filtragem de pacote controla acesso a uma rede analisando os pacotes de entrada e saída e transmitindo-os ou eliminando-os com base em critérios, como o endereço IP de origem, o Endereço IP de destino e o protocolo transportado no pacote.
- Um Roteador de filtragem de pacote utiliza regras para determinar se permite ou nega tráfego. Um roteador também pode realizar a filtragem de pacotes na camada 4, a camada de transporte
- Uma ACL é uma lista sequencial de instruções permit ou deny.

Capítulo 9: Resumo (continuação)

- A última instrução de uma ACL é sempre um implicit deny que bloqueia todo o tráfego. Para evitar que as instruções implicit deny no fim da ACL bloqueiem todo o tráfego, é possível adicionar a instrução **`permit ip any any`**.
- Quando o tráfego da rede passa por meio de uma interface configurada com uma ACL, o Roteador compara as informações no pacote com cada entrada, em ordem sequencial, para determinar se o pacote corresponde a uma das instruções. Se uma correspondência for encontrada, o pacote será processado em conformidade.
- As ACLs são configuradas para aplicação no tráfego de entrada ou no tráfego de saída.

Capítulo 9: Resumo (continuação)

- As ACLs padrão podem ser usadas para permitir ou negar tráfego somente dos endereços de IPv4 de uma origem. O destino do pacote e as portas envolvidas não são avaliados. A regra para fazer uma ACL padrão é colocá-lo próxima do destino.
- As ACLs estendidas filtram pacotes com base em vários atributos: tipo de protocolo, endereço de IPv4 de destino e origem e portas de origem ou destino. A regra para fazer uma ACL estendida é colocá-la o mais perto possível da origem.

Capítulo 9: Resumo (continuação)

- O comando de configuração global **`access-list`** define uma ACL padrão com um número no intervalo de 1 a 99 ou uma ACL estendida com números no intervalo de 100 a 199 e 2000 a 2699. As ACLs padrão e estendidas também podem ser nomeadas.
- O **`ip access-list standard`** *`name`* é usado para criar uma ACL nomeada padrão, enquanto o comando **`ip access-list extended`** *`name`* é para uma lista de acesso estendida. As instruções de ACL de IPv4 incluem o uso de máscaras curinga.
- Após uma ACL ser configurada, ela é vinculada a uma interface usando **o comando** **`ip access-group`** no modo configuração de interface.

Capítulo 9: Resumo (continuação)

- Lembre-se dos três Ps: uma ACL por protocolo, por direção, por interface.
- Para remover uma ACL de uma interface, primeiro insira **`o comando no ip access-group`** na interface e, em seguida, insira o comando global **`no access-list`** para remover toda a ACL.
- Os **comandos** **`show running-config`** e **`show access-lists`** são usados para remover toda a ACL. O **`comando show ip interface`** é usado para verificar a ACL na interface e a direção em que for aplicada.

Capítulo 9: Resumo (continuação)

- O **comando** access-class configurado no modo configuração de linha restringe as conexões de entrada e saída entre um determinado VTY e os endereços de uma lista de acesso.
- Como ocorre com as ACLs nomeadas do IPv4, os nomes do IPv6 são alfanuméricos, diferenciam maiúsculas de minúsculas e devem ser exclusivos. Ao contrário de IPv4, não há necessidade de uma opção entre padrão ou estendida.
- No modo de configuração global, use **o comando ipv6 access-list** *name* para criar uma ACL IPv6. O prefix-length é usado para indicar quanto de uma fonte ou de um endereço destino de IPv6 deve ser combinado.
- Depois que a ACL do IPv6 é configurada, ela é vinculada a uma interface usando **o ipv6** traffic-filter command.