Laboratório com Cisco Packet Tracer - Configurando ACL

Uma **ACL - Access Control List (Lista de Controle de Acesso)**, no contexto dos produtos **Cisco**, é um recurso do **IOS** que permite a você filtrar determinados pacotes, exatamente como um **firewall** faria, porém de uma maneira muito mais simplificada e com menos recursos.

ACLs são conjuntos de regras para filtrar pacotes, permitindo ou negando que eles sigam em frente. Utilizando ACLs no seu equipamento você pode filtrar tentativas de conexões indo/vindo de/para hosts específicos; pode bloquear completamente um determinado protocolo antes de tal requisição entrar na sua rede (claro que isso depende do posicionamento da ACL); controlar atualizações enviadas por protocolos de roteamento, etc.

Além disso, ACLs também podem ajudar a mitigar alguns tipos de ataques:

- IP spoofing;
- DoS (Denial of Server) através de TCP SYN;
- Além de atuar como um filtro de mensagens ICMP.

Bem básico, porém bastante útil de se ter na borda da sua rede, diminuindo o tráfego a ser analisado pelos firewalls internos.

Quais os tipos de ACLs?

Existem três "tipos" diferentes de ACLs no IOS:

- **ACLs standard (padrão)**: Esta é a lista mais básica, com menos funcionalidades. Filtram apenas através do endereço IP de origem.
    - Os números de identificação deste tipo de ACL vão de 1 a 99 e de 1300 a 1999.
    - Este tipo de ACL deve ser posicionado o mais próximo possível do destino do tráfego.
- **ACLs extended**: Além de filtrarem através dos endereços de IP de origem e destino, também permitem que se filtre o tráfego através de portas e protocolos.
    - Os números de identificação deste tipo de ACL vão de 100 a 199 e de 2000 a 2699.
    - Este tipo de ACL deve ser posicionada o mais próximo possível da origem.

- **ACLs named**: ACLs named (ou nomeadas) têm as mesmas características que ACLs extended, e permitem que você use um nome mais intuitivo para a ACL como allow_rh_out, deny_dmz_out, etc. Além disso, este tipo de ACL também oferece modos de edição mais avançados facilitando a vida do administrador. Os nomes utilizados para identificar este tipo de ACL não podem conter espaços ou pontuação e devem começar com uma letra.

ACL IP Padrão é o primeiro e mais simples tipo de bloqueio de pacotes em uma rede. O funcionamento básico da ACL consiste em pegar o IP de origem do pacote e fazer uma avaliação com as regras existentes em sua tabela montada pelo administrador de rede. A primeira regra combinada encerra a busca sequencial na lista de acesso e uma ação é executada.

Em todos os tipos de ACL, o método de análise é: cada pacote é comparado com cada uma das regras presentes na ACL *EM ORDEM* e a ação descrita na primeira regra que "casar" com o pacote será tomada.

Outra coisa importantíssima que é verdade em todos os tipos de ACL: sempre há uma regra "*deny all*" implícita no final de todas as ACLs. Ou seja, uma ACL sem nenhuma regra irá bloquear absolutamente todos os pacotes que são comparados com ela, pois caso não se combine com nenhuma regra o pacote é descartado em função da regra *deny all* implícita no final da lista de acesso. Portanto muito cuidado quando for aplicar uma ACL em uma interface.

A sintaxe do comando a seguir explora a **ACL padrão:**

access-list *<número da lista>* {deny | permit} [*endereço da origem*]

Exemplo:

```
                      (a)     (b)   (c)     (d)          (e)
RouterA(config)#access-list 10 deny 192.168.1.10 0.0.0.0
```

Sintaxe ACL IP Padrão

(a) comando
(b) identificação da ACL – pode ser de 1 a 99
(c) ação a ser executada quando a regra combina com o pacote - permit|deny
(d) endereço de origem
(e) máscara curinga (*wildcards*) ou **any**

OBSERVAÇÕES :

- Em (b) identificação da ACL : Em todas as versões de software, o *número da lista* pode ser qualquer número entre 1 e 99. No Cisco IOS Software Release 12.0.1, as ACLs padrão começam a usar números adicionais (1300 a 1999). Esses números adicionais são chamados de ACLs de IP expandidas. O Cisco IOS Software Release 11.2 incluiu a capacidade de usar *nome* de lista em ACLs padrão.
- O *endereço da origem* pode ser o endereço de um único host (item (d)), ou de um conjunto de nós ou sub-rede. Neste caso, deve-se usar a máscara curinga (item (e)).

Sobre a direção dos fluxos de dados

Não basta apenas criar uma ACL para que o tráfego passe a ser corretamente filtrado: você precisa definir a direção e a interface onde esta ACL será aplicada.

Mas como assim "direção"? Existem duas direções de tráfego em roteadores:

- **inbound**: Todo o tráfego vindo da rede e entrando no roteador através de uma de suas interfaces de roteamento.
- **outbound**: Todo o tráfego saindo de alguma interface do roteador e indo para a rede.

Depois de definida, a ACL deve ser aplicada à interface (entrada ou saída). Sintaxe:

```
router(config)#  interface <interface>
router (config-if)# ip access-group <number> {in|out}
```

Uma coisa que deve ser observada e que vai influenciar diretamente no seu design de ACL: é permitida apenas uma ACL por protocolo, por direção e por interface. Ou seja, você não pode ter duas ACLs que trabalhem com TCP, na direção inbound na mesma interface. É uma ACL ou a outra, nunca ambas.

Wildcard masks

Para definir quais sub-redes serão afetadas por uma determinada regra de ACL, você vai precisar usar a wildcard mask. Neste tipo de máscara 0 e 255 tem sentidos exatamente **opostos** à máscara de sub-rede que você já está acostumado.

Neste tipo de máscara, um bit 0 quer dizer que o octeto tem que "casar" exatamente e um bit 1 é ignorado (pode ser qualquer coisa). Ou seja, exatamente o contrário do que se faz quando se usa uma máscara de sub-rede normal. Porém, o valor dos bits continua o mesmo: da esquerda para a direita eles valerão 128, 64, 32, 16, 8, 4, 2, 1, respectivamente.

NOTA: Se você estiver configurando uma ACL em um ASA ou PIX, *não* use *wildcard masks*. Use a máscara normal.

Dois exemplos simples para você pegar o jeito:

Digamos que você precisa de uma *wildcard mask* que reconheça todos os IPs da rede 192.168.0.0/24. A máscara seria 0.0.0.255:

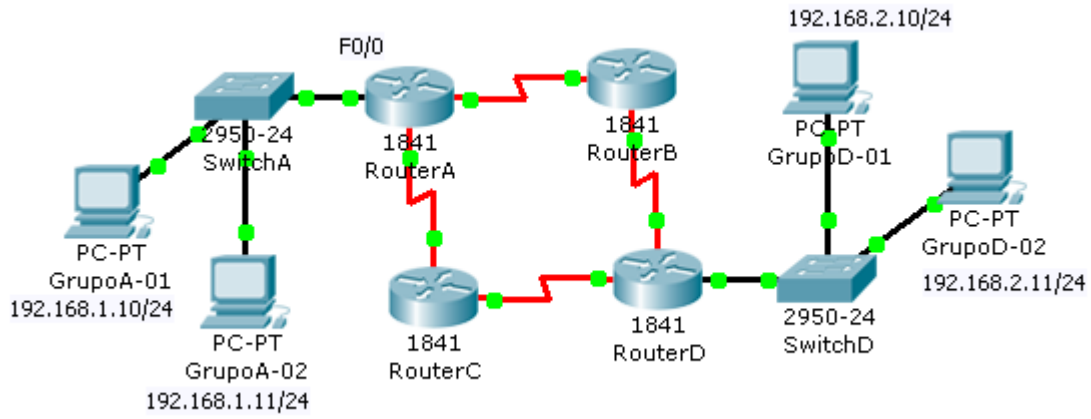
- Os três primeiros 0s indicam que é obrigatório que 192.168.0 esteja presente no endereço IP do pacote sendo analisado.
- O .255 indica que naquele octeto qualquer número é válido.

Para identificar um conjunto de sub-redes é a mesma coisa. Suponhamos que você quer que todas as redes entre 192.168.16.0/24 a 192.168.31.0/24 sejam afetadas por uma regra. A wildcard seria 0.0.15.255:

- Os dois primeiros 0s indicam que é obrigatório que 192.168 esteja presente;
- 15 é a soma dos bits para que as sub-redes desejadas casem com a regra; e
- 255 indica que o último octeto não importa e vai casar qualquer que seja o número lá.

EXEMPLO:

Para a rede apresentada na figura abaixo vamos simular um bloqueio de acesso do host GrupoA-02 ao GrupoD.



Topologia ACL IP Padrão - CCNA

Para resolver o problema vamos acessar o RouterD e executar a seguinte configuração:

```
RouterD>en
RouterD#conf t
```

```
Enter configuration commands, one per line. End with CNTL/Z.
RouterD(config)#access-list 10 deny host 192.168.1.11
RouterD(config)#access-list 10 permit any
RouterD(config)#int f0/0
RouterD(config-if)#ip access-group 10 out
```

As ACLs não têm efeito até que sejam aplicadas à interface do roteador. No exemplo, isso é feito nas últimas 2 linhas.

É importante lembrar sempre de acrescentar uma regra que não filtre os pacotes ao final, caso contrário todos os pacotes serão negados em função da regra *deny all* implícita em toda ACL.

A partir de agora, o host 192.168.1.11 não tem acesso ao GrupoD.

Regra de ouro: ACL IP Padrão deve sempre ser aplicada mais próxima do destino do pacote.

ACLs Estendidas

As ACLs estendidas foram introduzidas no Cisco IOS Software Release 8.3. As ACLs estendidas controlam o tráfego por meio da comparação dos endereços de origem e de destino dos pacotes IP com os endereços configurados na ACL.

O formato geral da sintaxe do comando de ACLs estendidas é

access-list *<número da lista>* {deny | permit} [*protocolo da camada de rede ou transporte:* ip|udp|tcp|...] [*endereço da origem*] [*endereço de destino*] [*nº da porta* ou opções]

Os comandos detalhados estão listados a seguir. As linhas estão distribuídas aqui considerando o espaço.

IP

access-list *access-list-number* {deny | permit} *protocol*

*{***host** *<IP>* **|** *source source-wildcard}*

destination destination-wildcard

[parâmetros adicionais]

OBS: As opções de *protocolo* são *ahp, eigrp, esp, gre, icmp, igmp, ip, ipinip, nos, ospf, pcp, pim, tcp e udp*

ICMP

access-list *access-list-number* {deny | permit} **icmp**

source source-wildcard

destination destination-wildcard

[*icmp-type* | [[*icmp-type icmp-code*] | [*icmp-message*]]

[parâmetros adicionais]

TCP

access-list *access-list-number* {deny | permit} **tcp**

source source-wildcard [*operator* [*port*]]

destination destination-wildcard [*operator* [*port*]] [**established**]

[parâmetros adicionais]

OBS: Operadores podem ser **eq** (diz que esta regra só se aplica quando a porta for igual ao numero indicado), **gt** (a regra só casa com portas acima do número especificado), **lt** (a regra só casa com portas abaixo do número especificado), **neq** (a regra só casa quando a porta *NÃO* for a que foi especificada), **range** (a regra casa apenas quando a porta especificada no pacote está no intervalo de portas especificado na linha de comando). Além destes também existem alguns outros com finalidades diferentes. Por exemplo, **established** (conexão estabelecida -- tráfego relacionado com uma conexão já estabelecida é automaticamente permitido, sem que você tenha que criar uma outra entrada na ACL para que o tráfego de volta possa passar pelo roteador corretamente), **rst** (casa com pacotes que tem o flag TCP RST ativado), **fin** (casa com pacotes que tem o flag TCP FIN ativado), etc. Opções aqui abundam :)

UDP

access-list *access-list-number* {deny | permit} **udp**

source source-wildcard [*operator* [*port*]]

destination destination-wildcard [*operator* [*port*]]

[parâmetros adicionais]

Em todas as versões de software, o *access-list-number* pode ser um número entre 101 e 199. No Cisco IOS Software Release 12.0.1, as ACLs estendidas começam a usar números adicionais (2000 a 2699). Esses números adicionais são referidos como ACLs de IP expandidas. O Cisco IOS Software Release 11.2 incluiu a capacidade de usar *nome* de lista em ACLs estendidas.

O valor 0.0.0.0/255.255.255.255 pode ser especificado como **qualquer**. Depois de definidas as ACL, devem ser aplicadas à interface (entrada ou saída). A direção deverá ser especificada.

interface <interface>

ip access-group {*number*|*name*} {*in*|*out*}

EXEMPLO

Essa ACL estendida é utilizada para permitir tráfego na rede 198.162.1.x (interna) e para receber respostas de pings externos ao mesmo tempo em que pings não solicitados são impedidos, sem que todos os outros tipos de tráfego sejam impedidos.

```
RouterA(config)# access-list 101 deny icmp any 192.168.1.0 0.0.0.255 echo
RouterA(config)# access-list 101 permit ip any 192.168.1.0 0.0.0.255

RouterA(config)# interface f 0/1
RouterA(config-if)#ip access-group 101 in
```

Observação: Alguns aplicativos, como gerenciamento de redes, exigem pings para obter uma função de manutenção de atividade. Se esse for o caso, convém limitar o bloqueio de pings de entrada ou ser mais granular em IPs permitidos/negados.

Para ver quais são as ACLs existentes no dispositivo (aplicadas a interfaces ou não), execute o seguinte comando:

router# show access-lists [*access-list-number*]

Como remover uma ACL de uma interface?

Depois de aplicar uma ACL em uma interface, obviamente você não precisa deixar ela lá para o resto da vida! Remover uma ACL de uma interface é bem simples, basta adicionar um “no” na frente do comando que você usou para colocá-la lá:

router(config-if)# no access-list 102

Considerando que a ACL 102 estivesse aplicada na interface acima, ela teria sido removida com o comando mostrado.