Frank Ned Santa Cruz de Oliveira
Responsável pelo Website Verdade
@Absolutahttp://www.absoluta.org
frank@absoluta.org

ARTIGO

Segurança em Roteadores

Palavras Chave: roteador, ACL, lista de controle de acesso, cisco

Observação:
O assunto tratado neste texto tem por objetivo principal ajudar os administradores de rede no processo de configuração de roteadores,com sua modesta abrangência, este trabalho pretende servir como motivação ao leitor para busca de novos conhecimentos no campo da segurança da informação. Não houve aqui a pretensão de esgotar o assunto, mas sim fornecer ao leitor um texto condensado, reunindo conceitos fundamentais ao entendimento dos termos relacionados.

Boa leitura!

Índice

---

1 - Introdução

Este artigo tem por objetivo abordar a configuração de listas de controle de acesso (ACL) em roteadores CISCO. Inicialmente faremos algumas considerações sobre o hardware e software dos roteadores CISCO. Em seguida abordaremos os conceitos das listas de controle de acesso e sua implementação no roteador.

A seguranca possui muitas faces e uma das mais importantes é a capacidade de controlar o fluxo de pacotes em uma rede, com o objetivo de proteger nossas redes de falhas, degradação dos servicos, roubo ou comprometimento dos dados resultantes de uma ação intencional ou de um erro provocado por usuários.

Mas uma solução efetiva de segurança não deve ser baseada somente em recursos técnicos, deve-se elaborar uma política de segurança de forma a definir-se as diretrizes de seguranca da instituição. Para tanto, existem padrões internacionais com as melhores práticas de seguranca, que podem auxiliar no processo de elaboração da política de segurança.

Antes de prosseguir a leitura deste artigo analise o item 9 - Exemplos, mesmo que agora você não entenda tudo, pois com a visualisação da ACL o entendimento dos conceitos torna-se mais rápido.

2 - Os Três As (AAA)

O controle de acesso é a forma pela qual pode-se controlar quem tem acesso aos servidores da rede e a quais serviços pode-se utilisar uma vez possuindo acesso aos mesmos.

- Autenticar - A autenticação é o metodo de identificação dos usuários que podem utilizar os recursos da rede;
- Autorizar - A autorização é o método de controle de acesso remoto;
- Auditar - A auditoria é o metodo de coletar as informações sobre os acessos, utilização dos recursos, tentativas de acesso falhas, horário de início de término de determinadas transações, número de pacotes enviados por protocolo entre outras;

Neste texto estaremos focados na AUTORIZAÇÃO. Para tanto, iremos tratar dos recursos de controle de acesso que podem ser implementados em roteadores CISCO, embora os conceitos possam ser aplicados a outros elementos de controle de acesso.

Deve-se ter em mente que um sistema efetivo de segurança não deve ser baseado somente em regras nos roteadores, deve-se utilizar outros elementos como:

- firewall
- ferramentas de IDS

- honey pots
- segurança dos hosts
- preservação e análise dos logs
- a política de segurança

O sistema operacional da Cisco (IOS - Internetwork Operational System) fornece várias funcionalidades que podem ser utilizadas para elevar o nível de segurança de uma rede. Entre estas funcionalidades está o filtro de pacotes que serão estudados nos próximos tópicos.

3 - Componentes Básico do Hardware

A CISCO produz vários tipos de roteadores. Embora estes produtos possuam diferenças consideráveis quanto ao seu poder de processamento e quanto ao número de interfaces que suportam, eles utilizam um conjunto básico de hardware. A figura-1 mostra um esquema genérico que destaca os componentes básicos de um roteador CISCO. Embora a CPU ou micro-processador, quantidade de RAM e ROM, quantidade e tipos de porta de I/O possam ser diferentes de produto para produto, cada roteador possui os componentes referenciados na figura-1.

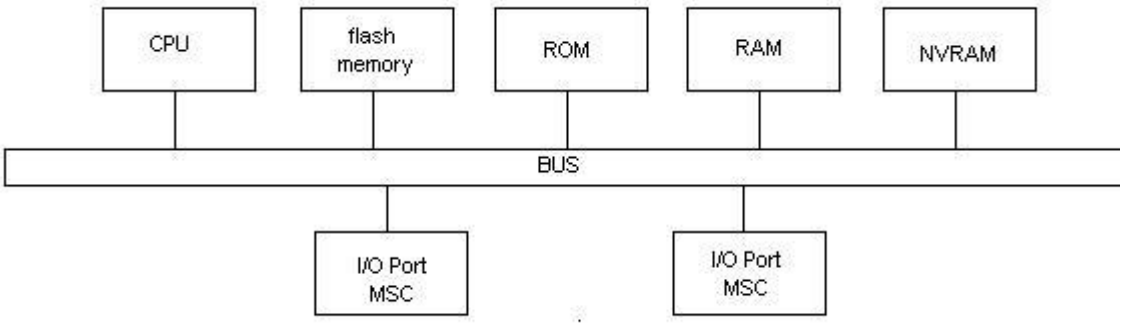

Figura 1 - componentes básicos do hardware

CPU:

A CPU ou micro-processador é responsável pela execução das instruções que ativam o roteador. O poder de processamento da CPU está relacionado de forma direta com a capacidade de processamento do roteador.

Flash Memory:

A flash memory é um tipo de ROM re-programavel. Esta memória pode ser utilizada para armazenar várias imagens de OS e micro-códigos do roteador. Esta fun"cão é útil para testar novas imagens. A flash memory também pode ser utilizada para trival file transfer protocol (TFTP) uma imagem de OS para outro roteador.

ROM:

A ROM contém códigos que realizam diagnósticos de inicialização do

roteador semelhante ao POST, (power on self-test) realizado por muitos PCs. Além disso, um progama bootstrap é utilizado para carregar o OS.

RAM:

A RAM é utilizada para armazenar as tabelas de roteamento, buffer de pacotes, fornecer uma área para enfileirar pacotes quando os mesmos não podem ser enviados para a saída devido ao grande volume de tráfego roteado para uma interface em comum. Além disto, prover espaço para armazenamento de informações sobre endereços ARP de forma a reduzir o tráfego de ARP e melhorar a capacidade de transmissão para LANs conectadas ao roteador. Quando o roteador é desligado, perde-se todas as informações armazenadas na RAM.

NVRAM:

A NVRAM (Nonvolatile RAM) ao contrário da RAM, não perde seu conteúdo quando o roteador é desligado. A NVRAM possui um backup da configuração do roteador. Desta forma, o roteador pode retornar à operação sem a necessidade de ser re-configurado. O uso da NVRAM elemina a necessidade de ter HD ou unidade de disquete em um roteador.

Portas de I/O e MSC (Media-Specific Converters):

As portas de entrada/saída (I/O) representam as conexões pelas quais os pacotes entram e saem do roteador. Cada porta de entrada/saida (I/O) é conectada a uma media-specific converter (MSC), que fornece a interface física para um tipo específico de meio como uma LAN ethernet ou token ring ou a uma WAN RS-232 ou V.35. Os dados são recebidos através de uma LAN; os cabeçalhos da camada 2 são removidos e os pacotes são enviados para a RAM. Quando estas ações acontecem, a CPU examina as tabelas de rotas para determinar a porta de saida dos pacotes e o formato no qual os mesmos devem ser encapsulados.

Este processo é chamado de process switching, no qual cada pacote deve ser processado pela CPU que consulta as tabelas de rota e determina para onde enviar os pacotes. Os roteadores CISCO possuem outro processo chamado de fast switching, nesta forma de processo o roteador mantém um cache na memória com informações sobre o destino dos pacotes IP e a próxima interface.

O roteador constrói este cache salvando as informações previamente obtidas da tabela de roteamento. O primeiro pacote para um destino específico executa um processamento da CPU para consultar as tabelas de rota. Uma vez que esta informação é obtida a mesma é inserida no cache do fast switching. Desta forma as tabelas de roteamento não são consultadas quando um novo pacote é enviado para o mesmo destino. Desta forma o roteador pode enviar os pacotes de forma mais rápida e consequentemente reduzir a carga de processamento da CPU. Vale resaltar que existem algumas variações quanto à forma de processamento em alguns equipamentos.

Existe outra forma de cache chamada de netflow switching, onde além de armazenar o IP de destino armazena-se o IP de origem e as portas TCP e UDP. Este recurso está disponível somente em roteadorees de maior capacidade como os da familia 7000.

4 - O Processo de Inicialização do Roteador

Quando você liga o roteador algumas rotinas de inicialização sao executadas (FIGURA-2):

- POST - power-on self-test, durante este processo o roteador executa diagnósticos a partir da ROM, esses diagnósticos verificam as operações básicas da CPU, a memória e as interfaces. Após a verificação das funções do hardware o roteador realiza a inicialização do software.

- Localização e carga da imagem do IOS - Após o POST o roteador procura o registro de configuração para determinar onde está localizado a imagem do IOS. Se o roteador não encontrar uma imagem válida do sistema ou se a sequência de boot for interrompida o sistema entra no modo ROM monitor; caso contrário o mesmo procura na NVRAM o indicador da localização da imagem que pode estar:
  - na ROM;
  - em um servidor TFTP;
  - na flesh memory;
- Uma vez que a imagem do IOS seja encontrada e carregada passa-se para a próxima fase.

- Localizar e carregar o arquivo de configuração - Este arquivo possui todas as informações de configuração especificadas para o roteador em questão. O arquivo de configuração é armazenado na NVRAM, mas você pode configurar o roteador para carregá-lo a partir de um servidor TFTP. Caso não seja encontrado um arquivo de configuração o roteador entra no modo de setup.

Após completar o processo de inicialização o roteador começa a operar. A partir deste ponto você pode construir novos parâmetros de configuração ou alterar os existentes.

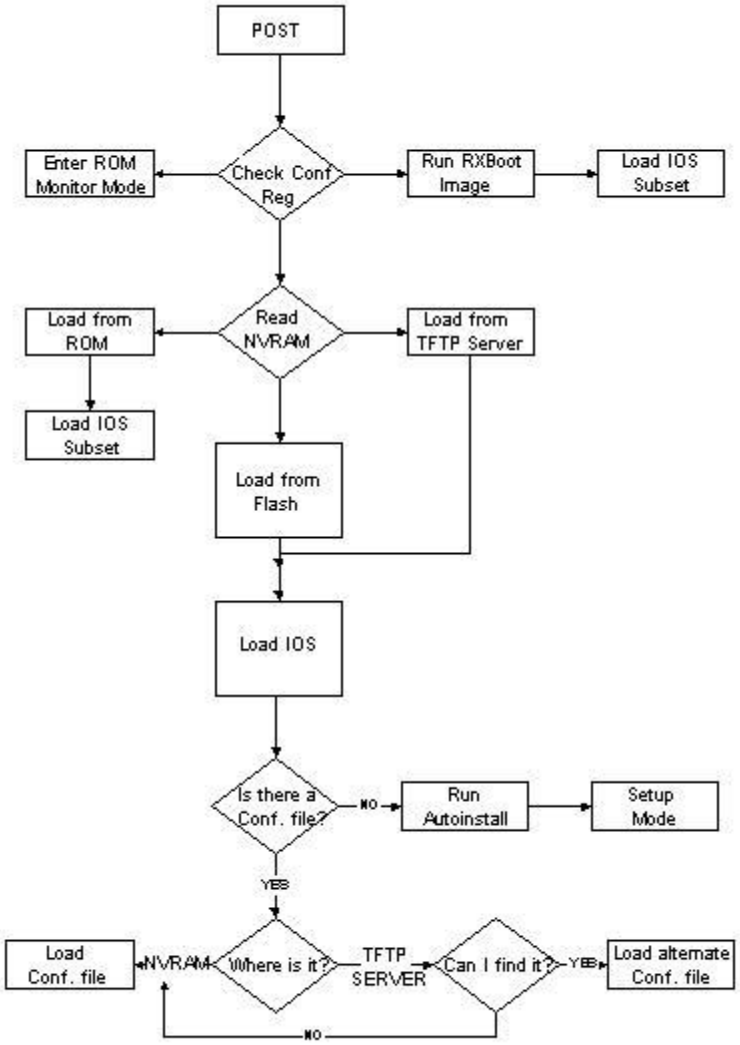

Figura 2 - fluxo do processo de inicialização do roteador

Após a inicialização tanto a imagem do IOS quanto o arquivo de configuração são armazemados na RAM, sendo que a imagem do IOS é armazenado nos endereços baixos e o arquivo de inicialização no endereço alto, conforme ilustrado na fugura-3.

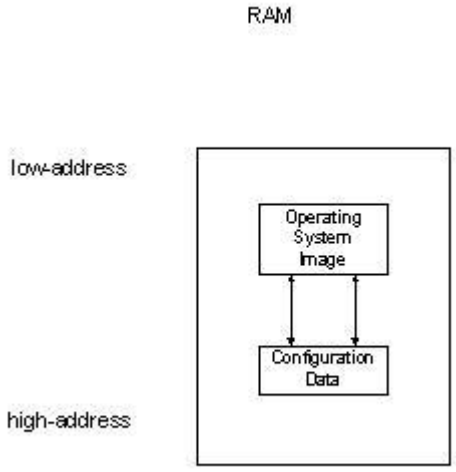

Figura 3 - IOS e arquivo de configuração na RAM

5 - Fluxo dos Dados

Uma vez que o roteador sabe qual é ou quais são os tipos de interfaces (Ethernet, Token Ring, FDDI, X.25, Frame Relay, ...), o mesmo pode verificar o formato dos frames que chegam e montar os frames de saída, além disso o roteador pode verificar a integridade dos dados que chegam, pois como o mesmo conhece o tipo de interface, pode calcular o cyclic redundancy check (CRC), da mesma forma o roteador pode calcular o CRC dos frames de saída.

Caso as tabelas de roteamento possuam apenas rotas estáticas, estas tabelas não serão trocadas com outros roteadores.

O cache ARP representa uma área da memória onde são armazenadas as relações entre o endereco IP e seu endereço físico (o endereço MAC da camada 2).

Os dados que são recebidos ou preparados para transmissão podem entrar em filas de prioridades, onde o tráfego de baixa prioridade é atrasado em favor do processamento do tráfego de alta prioridade. Se o modelo do roteador suportar priorização de tráfego, certos parâmetros de configuração podem ser informados ao roteador para indicar como realizar esta priorização.

As informações sobre o fluxo dos dados como localização e status dos pacotes são armazenadas na hold queue.

As entradas das tabelas de roteamento informam a interface de destino para o qual determinados pacotes devem ser roteado. Se o destino for uma LAN e for necessária a resolução de endereço, o roteador procúra o endereço MAC inicialmente no cache ARP. Caso o endereço não seja encontrado no cache ARP, o roteador monta um pacote ARP para descobrir o endereço MAC.

Uma vez que o endereço de destino e o método de encapsulação estão determinados os pacotes são enviados para a porta da interface. Dependendo do volume de tráfego novamente o pacote pode entrar em uma fila de prioridade, hardware buffer, até que possa ser enviado.

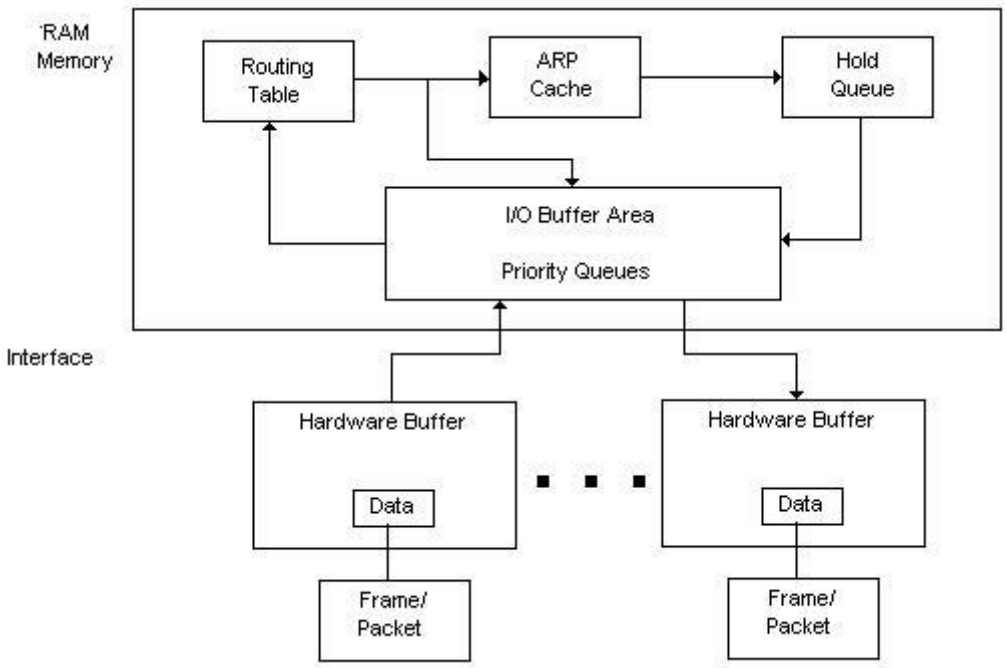

Figura 4 - fluxo dos dados

6 - Controle do Tráfego com ACL

Uma lista de controle de acesso (ACL) é uma lista de regras ordenadas que permitem ou bloqueiam o tráfego de pacotes baseado em certas informações presentes nestes. Vamos analisar esta frase de forma mais detalhada:

- Uma lista de controle de acesso (ACL) é uma lista de regras ordenadas - isso significa que a ordem de criação das regras na lista de acesso é muito importante. Um dos erros mais comuns durante a criação de listas de acesso é a configuração das regras em ordem incorreta.

- que permitem ou bloqueiam o tráfego de pacotes - antes de mais nada é importante sabermos que no final da lista de acesso existe uma regra implícita que bloqueia tudo. Um pacote que não é explicitamente permitido será bloqueado pela regra que bloqueia tudo. Outro erro comum durante a criação de uma lista de controle de acesso é o esquecimento deste fato.

- baseado em certas informações presentes nos pacotes - geralmente são informações presentes nos cabeçalhos dos pacotes da camada 3 ou da camada 4. Fora algumas exceções listas de controle de acesso não podem usar informações de camadas superior a 4 com o objetivo de filtragem. Por exemplo: uma lista de controle de acesso não possui a capacidade de filtrar comandos FTP. Na verdade existe uma forma de fazer isso... :)) usando o Context-Based Access Control (CBAC) que possui a capacidade de filtrar pacotes baseado nas informações das aplicações conhecidas, mas este assunto será tratado em outro artigo.

Além do controle de acesso as ACLs podem ser utilizadas para outras funções como:

- Dial on Demand - listas de controle de acesso são utilizadas para definir que pacotes são permitidos para a ocorrencia de uma conexão dial (DDR - Dial-on-Demand Routing);
- Queuing Features - listas de acesso podem controlar que tipo de pacotes serão alocados em certos tipos de filas, por exemplo, para o controle de prioridade;
- Routing Update Filters - listas de acesso podem controlar a troca de informação entre roteadores;
- Router Access - lista de acesso pode controlar o acesso telnet ou SNMP ao roteador.

Note que estas funções são diferentes do controle de fluxo de pacotes através do roteador.

Listas de acesso podem ser configuradas para todos os tipos de protocolos roteáveis como: IP, AppleTalk entre outros. Neste texto nosso foco é TCP/IP.

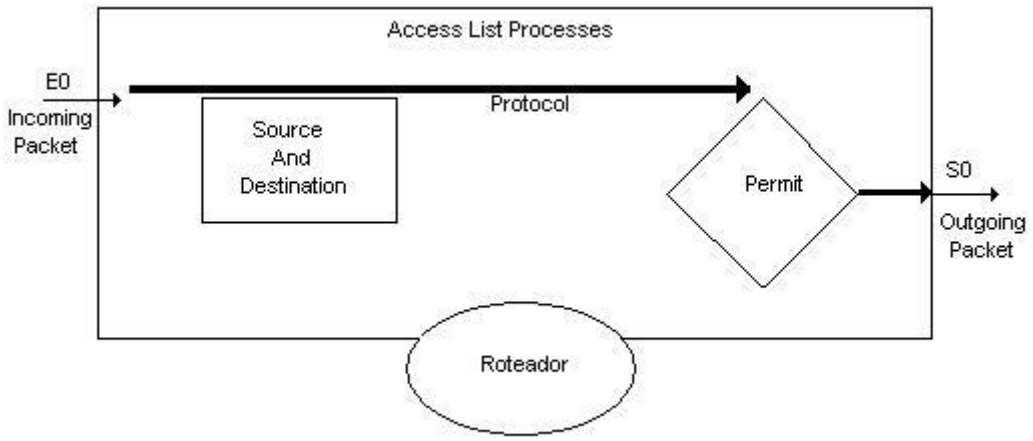

Figura 5 - Visão geral do fluxos de uma lista de acesso

7 - Como Funciona a ACL

Fluxo dos Pacotes Através das Listas de Acesso

O processo tem início quando uma interface recebe um pacote. O roteador verifica a tabela de roteamento à procura de uma rota para o pacote. Caso não tenha uma rota este pacote será descartado e sera enviada para a origem uma mensagem de ICMP (unreachable destination). Caso contrário verifica-se se existe uma lista de controle de acesso aplicada à interface, não existindo o pacote é enviado para o buffer da porta de saída, caso contrário o pacote é analisado pela lista de controle de acesso da interface em questão. Uma vez que o fluxo de dados atraves da uma determinada interface é bidirecional, uma ACL pode ser aplicada em uma direção especifica da interface:

- inbound - verifica se o processamento do pacote deve continuar após o seu recebimento em uma determinada interface;
- outbound - verifica se o pacote deve ser enviado para uma interface de saída ou bloqueado.

Vale notar que os pacotes gerados pelo roteador como troca de tabelas de roteamento não são afetados pelas regras aplicadas a uma interface no sentido outbound, a única forma de controlar os pacotes gerados pelo roteador como atualização de tabela é atraves de ACL de inbound.

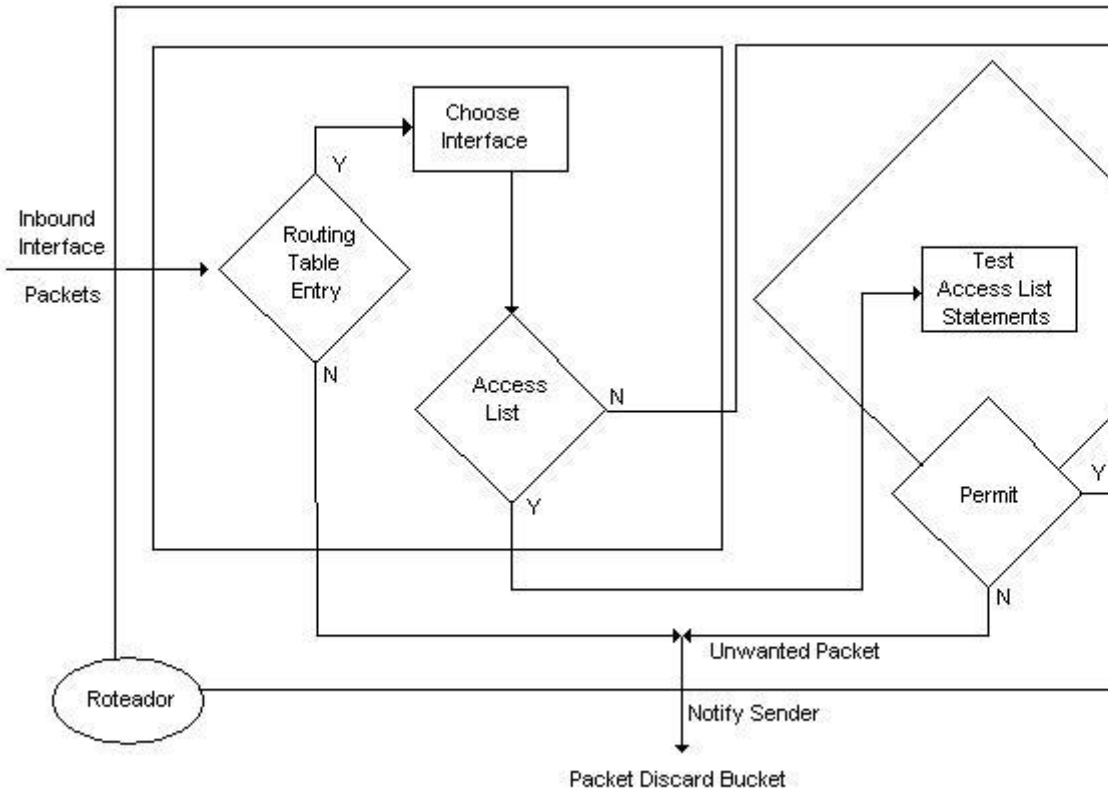

Figura 6 - fluxo do pacote através da ACL

A lista de acesso é conferida em ordem seqüencial, ou seja, o pacote é testado a partir da primeira regra. Assim, se o pacote enquadrar-se em alguma regra e verificando a condição do mesmo - se permitido ou bloqueado. Caso o pacote não se enquadre em nenhuma das regras, o mesmo será bloqueado pela última regra, a qual é implícita e bloqueia tudo que não está explicitamente permitido, conforme já dito anteriormente.

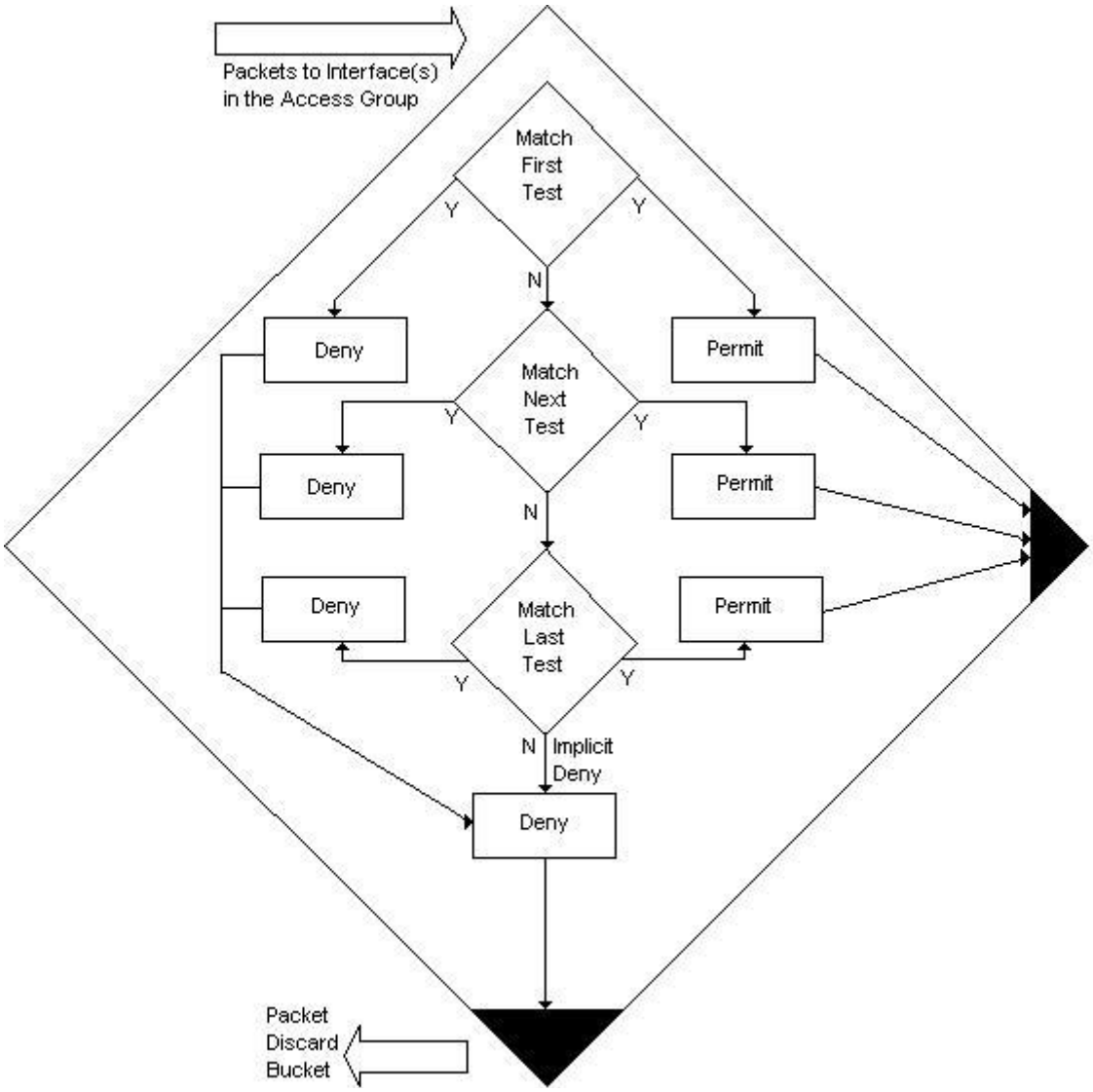

Figura 7 - fluxo da desicao de analise do pacote

Tipos de Listas de Acesso

Os dois principais tipos de listas de acesso são:

- **Standard** - A lista de acesso standard verifica o IP de origem de um pacote que pode ser roteado. Baseada na rede/sub-rede/ endereço do hosts é permitido ou bloqueado o envio do pacote, ou seja, que o mesmo saia por outra interface.

- **Extended** - A lista de acesso extended possui maior recursos de verificação pode analisar IP origem, IP destino, porta origem, porta destino, protocolos e alguns outros parâmetros, de forma a permitir ao administrador de segurança maior flexibilidade na elaboração das regras.

Identificando as Listas de Acesso

Ao se configurar listas de acesso em um roteador deve-se identificar cada

lista de forma única.

Em alguns protocolos as listas de acesso devem ser identificadas por nome, já em outros devem ser identificadas por número e alguns protocolos permitem a identificação por nome ou número. Quando utilizamos números para identificar as listas de acesso estes devem pertencer a um conjunto de números que "identificam" o protocolo.
A partir do IOS 11.2 é permitida a identificação da lista de acesso utilizando-se nomes definidos pelo administrador, isso para listas de acesso standard e/ou extended.

Baseado no identificador o roteador decide qual software de controle de acesso deve ser utilizado.

Observe as tabelas abaixo o agrupamento de nomenclatura para listas de acesso:

Tipo de Lista de Acesso	Número/Identificador
IP Standard Extended	1 - 99 100 - 199 por nome (IOS >= 11.2)
IPX Standard Extended Filtro SAP	800 - 899 900 - 999 por nome (IOS >= 11.2F)
AppleTalk	600 - 699

Figura 8 - Agrupamento por tipo de lista de acesso

Protocolo
Apollo Domain
IP
IPX
ISO CLNS
NetBIOS IPX
Source-router bridging Netbios

Figura 9 - Protocolos identificados por nome

Protocolo	Faixa
IP	1 - 99
Extended IP	100 - 199
Ethernet type code	200 - 299
Ethernet address	700 - 799
Transparent bridging (protocol type)	200 - 299
Transparent bridging (vendor code)	700 - 799
Extended transparent bridging	1100 - 1199
DECnet and extended DECnet	300 - 399
XNS	400 - 499
Extended XNS	500 - 599
AppleTalk	600 - 699
Sourc-route bridging (protocol type)	200 - 299
Source-router bridging (vendor code)	700 - 799
IPX	800 - 899
Extended IPX	900 - 999
IPX SAP	1000 - 1099
Standard VINES	1 - 100
Extended Vines	101 - 200
Simple VINES	201 - 300

Figura 10 - Protocolos identificados por número

8 - Implementando ACL

Finalmente é chegada a hora da implementação.

Inicialmente, será abordado um tópico que com freqüência causa confusão ao se tratar de listas de controle de acesso: as máscaras de wildcards.

Funcionamento do *wildcar* em roteadores Cisco [1]

A filtragem de endereço ocorre com a utilização de máscaras wildcard para identificar o que é permitido ou bloqueado nos bits do IP. As máscaras wildcard para os bits de endereço IP utilizam o número 1 e o número 0 para a identificação do que deve ser filtrado nos bits do IP.

- Uma máscara com valor 0 significa que o bit deve ser checado;
- Uma máscara com valor 1 significa que o bit deve ser ignorado;

```
128  64  32  16  8  4  2  1
-----------------------------------------------------------------
0    0   0   0   0  0  0  0   = verifica todos os bits
0    0   0   0   0  1  1  1   = ignora os ultimos 3 bits
1    1   1   1   0  0  0  0   = verifica os ultimos 4 bits
1    1   1   1   1  1  1  1   = ignora todos os bits do octeto
```

Figura 11 - exemplo de wildcard

É importante observar que estes bits NÃO possuem relação com as máscaras de IP. A máscara de sub-rede é utilizada para determinar quantos bits de um IP representam uma porção da sub-rede, ou seja, a máscara de

sub-rede determina quais bits são importantes para definir uma sub-rede. Um binário setado em 1 indica que o bit do endereço IP é parte de uma sub-rede, já o binário setado com 0 indica que o bit do endereço IP faz parte da porção host.

No exemplo a seguir vamos verificar como as máscaras de bits (0 | 1) bloqueiam ou permitem o tráfego de pacotes baseado no endereco IP.

Teste de condição de uma lista de controle de acesso com protocolo IP: Um administrador deseja testar um endereço IP por sub-rede (172.30.16.0 até 172.30.31.0), os dois primeiros octetos correspondem a parte rede (172.30) o terceiro octeto corresponte a sub-rede (16 até 31), o quarto octeto corresponde ao host.

O administrador deseja usar as máscaras wildcard para bits IP para verificar as sub-redes 172.30.16.0 até 172.30.31.0. Isso é feito da seguinte forma:

- Inicialmente a máscara verifica os dois primeiros octetos 172.30, para isso usa 0s nos bits do wildcard;
- Como não existe interesse em filtar a parte host o quarto octeto será ignorado, para isso os bits devem ser setados para 1;
- No terceiro octeto, onde está localizada a sub-rede, a máscara irá verificar a posição correspondente ao binário 16, ou seja este bit deve estar com 0 bem como os bits superiores, já os bits abaixo do binário 16 devem ser ignorados, para que isso ocorra devem ser setados com 1s, assim temos:

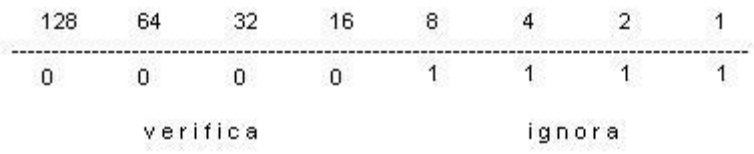

Figura 12 - exemplo de wildcard para IP 172.30.16.0 a 172.30.31.0

O resultado final é:
Para o endereço: 172.30.16.0
Com a máscara: 0.0.15.255
Serão verificadas as sub-redes: 172.30.16.0 a 172.30.31.0

Existem algumas abreviações que facilitam a utilização dos wildcard:

- Vamos considerar uma rede onde o administrador permite a entrada de qualquer endereço IP, ou seja, qualquer destino para sua rede é permitido. Para indicar qualquer endereço IP o administrador deve informar 0.0.0.0, agora para indicar que a lista de controle de acesso deve ignorar a verificação de qualquer bit a máscara deve ser setada com 1s, ignorar a verificação de qualquer bit significa aceitar todos, o resultado final será:

  Endereço IP: 0.0.0.0
  Máscara: 255.255.255.255
  Resultado: permite / aceita qualquer endereço

Neste caso a máscara 255.255.255.255 pode ser substituída pela palavra (abreviação) **any**.

- Outra possibilidade é quando o administrador deseja criar uma regra que verifique um endereço IP específico; dado o IP 172.30.1.29 deseja-se que a regra criada verifique todo o endereço, para que isso ocorra todo os bits devem ser setados com 0s, desta forma teremos:

Endereço IP: 172.30.1.29
Máscara: 0.0.0.0
Resultado: verificação do endereço específico.

Neste caso a máscara 0.0.0.0 pode ser substituída pela palavra (abreviação) host.

A utilização de máscara incorreta pode levar à implementação de listas de acesso com falhas, como no exemplo a seguir:

Vamos supor que desejamos permitir todos os pacotes IP originados na sub-rede 10.10.0.0 255.255.0.0 com destino ao host 160.10.2.100, todos os demais pacotes devem ser bloqueados. (caso não entenda os comandos, eles serão explicados no próximo tópico).

```
absoluta(config)# access-list 101 permit ip 10.10.0.0 0.0.0.0 160.10.2.100 0.0.0.0
absoluta(config)# exit
absoluta(config)# show access-list 101
Extended IP access list 101
    permit ip host 10.10.0.0 host 160.10.2.100
```

Pode-se notar que foi criada uma lista de acesso usando a máscara 0.0.0.0. Quando usamos o comando "show access list" o roteador exibe uma entrada com "host". Isso significa que o endereço de origem deve ser exatamente 10.10.0.0, ou seja, somente será permitido o tráfego de pacotes com endereço IP de origem 10.10.0.0 e destino 160.10.2.100. Todos os demais endereços serão bloqueados, inclusive 10.10.1.1, 10.10.1.2, etc. Não é isso que queremos. O nosso objetivo é permitir o tráfego de todos os hosts da sub-rede 10.10.0.0/16.

Tendo por base o exemplo acima, deve-se criar uma máscara que verifique os dois primeiros octetos "10.10" e ignore os dois últimos "0.0". Lembre-se que binário 0 significa verificar e binário 1 significa ignorar, baseado nesta informação, vamos novamente criar a lista de controle de acesso:

```
absoluta(config)# access-list 101 permit ip 10.10.0.0 0.0.255.255 160.10.2.100 0.0.0.0
```

```
absoluta(config)#                                    exit
absoluta(config)#          show          access-list          101
Extended          IP          access          list          101
   permit    ip    10.10.0.0    0.0.255.255    host    160.10.2.100
```

Agora o comando "show access list" mostra a nova máscara, desta vez de forma correta. Os dois últimos octetos contém todos os bits setados para 1 (o que equivale ao decimal 255).

Existe, contudo, uma forma mais prática de determinar a máscara a ser utilizada, não falei isso no início por que sempre uso a lógica militar "se pode complicar...para que simplificar" ...:).

Tudo que você tem a fazer é subtrair a máscara de sub-rede em formato decimal de 255, isso para cada um dos octetos, vejamos um exemplo:

```
   máscara          de          rede:          255.255.224.0
máscara              wildcard:                ???.???.???.???
pimeiro octeto = 255 - mascara de rede = 255 - 255 = 0
segundo octeto = 255 - mascara de rede = 255 -
                 255                   =          0
terceiro octeto = 255 - mascara de rede = 255 -
                 224                   =          31
quarto   octeto = 255 - mascara de rede = 255 -
                 0                     =          255
máscara wildcard: 0.0.31.255
```

Esta é a forma rápida e fácil de determinar as máscaras wildcard. Mas o importante é entender **por que** isso funciona e não simplesmente **como** funciona.

Criando Listas de Acesso

Como dito anteriormente neste artigo tratamos das listas standard e extended, desta forma vamos passar para a sintaxe deste tipo de listas.

STANDARD

```
access-list   número da lista   {permit | deny}   origem   [mascara wildcard]
```

Figura 13 - Sintaxe de uma lista de acesso standard

Lembre-se que para listas de controle de acesso standard a faixa de numeração utilizada inicia-se em 1 e vai até 99.

A partir do modelo apresentado na figura-15, vamos montar alguns exemplos de lista de controle de acesso:

Exemplo 1: permitir somente a entrada de pacotes com origem na rede 172.16.0.0:

```
access-list    1    permit    172.16.0.0    0.0.255
```

Exemplo 2: bloquear um hosts específico com origem na rede 172.16.0.0:

```
access-list    1    deny    172.16.1.30    255.255.255
```

Exemplo 3: bloquear uma determinada sub-rede :

```
access-list    1    deny    172.16.1.0    0.0.0.255
```

EXTENDED

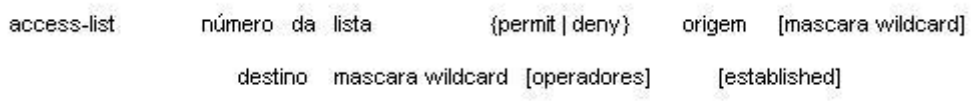

Figura 14 - Sintaxe de uma lista de acesso extended

Para as listas de controle de acesso extended a faixa de numeração utilizada para referência inicia-se em 100 e vai até 199.

Os protocolos passíveis de filtragem no caso de TCP/IP, são:

- IP;
- TCP;
- UDP;
- ICMP;
- GRE;
- IGRP;

Os operadores são:

- lt (less than) - menor que
- gt (greater than) - maior que
- eq (equal) - igual
- neq (not equal) - não igual

Exemplo 1: bloquear FTP para a interface ETH0:

```
access-list 101 deny tcp
172.16.1.0  0.0.0.255  192.168.0.0  0.0.255.255  eq  21
access-list 101 deny tcp
172.16.1.0  0.0.0.255  192.168.0.0  0.0.255.255  eq  20
```

Exemplo 2: bloquear tentativas de telnet para fora da rede 192.168.1.0 e permite os demais tráfegos:

```
access-
list   101   deny   tcp   192.168.1.0   0.0.0.255   any   eq    23
access-list   101   permit ip    any    any
```

Implementando as Listas de Acesso

Modo de Configuração

Para implementar uma lista de controle de acesso devemos:

- Entrar em modo de configuração:

```
absoluta# configure
absoluta(config)#
```

- Remover a lista de controle de acesso vigente:

```
absoluta(config)# no access-list numero da lista
```

- Aplicar a lista com as novras regras:

```
absoluta(config)# access-list numero da lista
absoluta(config)# access-list numero da lista
absoluta(config)# access-list numero da lista
```

- Selecionar uma interface para aplicar a lista de controle de acesso:

```
absoluta(config)# interface ethernet 0
```

- Aplicar a lista de controle de acesso de acordo com o sentido de verificação:

```
absoluta(config)# ip access-group  numero da lista  {in | out}
```

- Para sair do modo de configuração digite: exit ou end ou CTRL-Z.
- Gravar a nova configuração na NVRAM:

```
absoluta(config)# write mem
```

Performance

Geralmente a performance do roteador é uma das principais preocupações dos administradores de rede quando se fala em implementação de listas de controle de acesso em roteadores. sabemos que as regras da lista de controle de acesso são analisadas seqüencialmente, ou seja, regra-1, regra-2, regra-3 e assim sucessivamente até que seja encontrada uma regra, que coincida com o pacote analisado ou encontrar a última regra que bloqueia tudo que não está permitido. Desta forma devemos observar alguns procedimentos que devem ser adotados no sentido de minimizar o impacto que as listas de controle de acesso possam causar:

- Mensurar os recursos do roteador (memória, processador, outros);
- Avaliar os serviços habilitados no roteador (criptografia, outros);
- Entender o tráfego da rede;
- Mensurar o volume de pacotes;
- Classificar o volume de tráfego por servidor, protocolo e sentido do tráfego;

- A análise destas informações é de vital importância para a implementação das regras de forma a minimizar os possíveis impactos que possam vir a ocorrer no roteador.

Uma sugestão de estratégia de implementação com atenção ao impacto é:

- Sempre que possível aplicar as listas de controle de acesso no sentido de entrada (in), pois desta forma os pacotes serão descartados antes de serem roteados para uma das interfaces de saída, consequentemente minimizando o processamento de roteamento de pacotes;
- Implementar inicialmente as regras que contemplam o maior volume de transações da sua rede, agrupada por servidor/serviços;
- Como a pilha IP inclui ICMP,TCP e UDP. Sempre insira primeiro as regras mais específicas, para depois colocar as mais genêricas;
- Após a implementação da lista por grupo de servidor/serviço insira uma regra que bloqueia todos os demais pacotes do grupo, isso evita que o pacote passe pelo crivo de outros grupos ao qual não pertence;

Mantendo *Backup* dos Arquivos de Configuração

A manutenção de cópias de segurança dos arquivos de configuracao e do IOS é de fundamental importância, pois eventualmente devido a falhas de corrente elétrica estes arquivos pode ser danificados ou apagados da memória flash. Além disso a manutenção de backup dos arquivos facilita a administração de redes com vários roteadores. O admistrador de rede possui várias possibilidades de realização de cópias de segurança, uma delas é o TFTP.

O ideal é existir uma rede segregada dedicada para a função de gerência de rede, onde inclui-se a gerência da segurança, esta rede deve estar isolada da rede de dados e prover canais de comunicação out-of-band, no caso de necessidade de utilização dos canais de dados para funções de gerência é recomendável a utilização de canais seguros.

De qualuqer forma, como o serviço TFTP não requer altenticação é extremamente recomendável a implementação de algum mecanismo que controle a origem das conexões ao servidor TFTP.

Os comandos TFTP para realizar cópia de segurança e atualização de arquivos são:

- **copy tftp running-config** - Configura o roteador de forma direta, copiando os arquivos do servidor TFTP para a DRAM do roteador.

- **copy startup-config tftp** - Realiza o back-up dos arquivos de inicialização da NVRAM para o servidor TFTP.

- **copy tftp startup-config** - Atualiza o arquivo de inicialização, copiando do servidor TFTP e gravando na NVRAM.

Veja alguns exemplos abaixo:

Para caregar uma nova versão da imagem do IOS para a memória flash do roteador, use o comando **copy tftp flash**, como mostrado abaixo:

```
absoluta# copy tftp flash

System flash directory:
File Length Name/status
 1 4171336 c4500-j-mz_112-15a.bin
[4171400 bytes used, 22904 available, 4194304 total]
Address or name of remote host [10.1.10.40]? 10.1.10.40
Source file name? c4500-j-mz_112-15a.bin
Destination file name [c4500-j-mz_112-15a.bin]? yes
Accessing file 'c4500-j-mz_112-15a.bin' on 10.1.10.40...
Loading c4500-j-mz_112-15a.bin from 10.1.1.12 (via TokenRing1): [OK]

Erase flash device before writing? [confirm] yes
Flash contains files. Are you sure you want to erase? [confirm] yes

Copy 'c4500-j-mz_112-15a.bin' from server
 as 'yes' into Flash WITH erase? [yes/no] yes
Erasing device... eeeeeeeeeeeeeee ...erased
Loading c4500-j-mz_112-15a.bin from 10.1.1.12 (via TokenRing1): !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 4171336/4194304 bytes]

verifying checksum... OK (0x29D5)
Flash copy took 00:00:30 [hh:mm:ss]
absoluta#
```

Para realizar um backup do arquivo de configuração, use o comando copy running-config tftp, conforme o exemplo abaixo:

```
absoluta# copy running-config tftp
Remote host [ ]? 10.1.10.40
Name of configuration file to write [router-confg]?
config_router_principal_2000-05-20.bak
Write file config_router_principal_2000-05-20.bak on host 10.1.10.40?
[confirm] yes
Building configuration...

Write config_router_principal_2000-05-20.bak !!!!!! [OK]
absoluta#
```

9. Exemplos

Para nossos exemplos vamos assumir uma topologia de rede conforme a

figura 15:

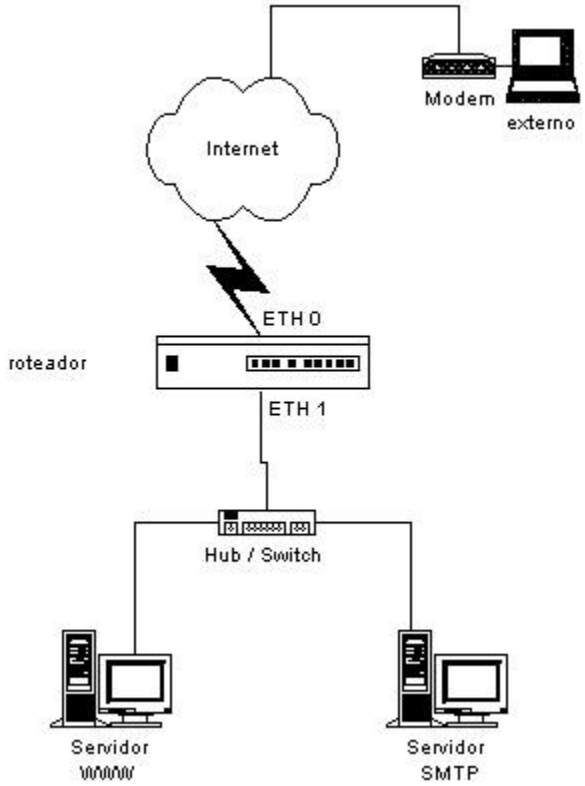

Figura 15 - visão da topologia de rede para os exemplos

O objetivo desta seção é demonstrar o processo de criação de regras. Não existe a intenção de demonstrar um grupo completo de regras, cada rede possui suas caracteristicas próprias e os filtros devem ser criados em conformidade com a realidade da rede.

Caso você não comprenda o que esta fazendo poderá ter problemas devido a uma má implementação de filtros, desta forma, evite simplemente copiar as regras que encontar nos livros e artigos, mas sim entende-las antes da implementação. De qualquer forma consulte os artigos listados na seção link, pois os mesmos possuem um conjunto de regras interessantes.

Vamos usar como exemplo para montagem de regras a conexão de SMTP, de forma a permitir o envio e recebimento de emails.

regra	direção	ip_orig.	ip_dest.	protocolo	porta_dest.	ação
A	in	externo	interno	TCP	25	permite
B	out	interno	externo	TCP	>1023	permite
C	out	interno	externo	TCP	25	permite
D	in	externo	interno	TCP	>1023	permite

E	ambas	qualquer	qualquer	qualquer	qualquer	bloqueia

- As regras A e B permitem a entrada de email
- As regras C e D permitem a saida de email
- A regra E é a regra default que bloquei tudo

Vamos considerar alguns exemplos de pacotes para facilitar o entendimento das regras. Nosso servidor SMTP possui o IP 10.0.0.1 e alguem de uma rede externa com IP 172.16.2.3 tenta enviar um email para nós. A porta de origem utilizada pelo cliente externo é 1234 com destino para 25, analisanos esta situação de acordo com as regras implementadas temos o seguinte:

pacote	direção	ip_orig.	ip_dest.	protocolo	porta_dest.	ação
1	in	172.16.2.3	10.0.0.1	TCP	25	permite (A)
2	out	10.0.0.1	172.16.2.3	TCP	>1023	permite (B)

Neste caso as regras do nosso roteador permitem a entrado dos pacotes de email:

- A regra A permite a entrada do pacote com origem em 172.16.2.3 e destino 10.0.0.1.
- A regra B permite que o servidor 10.0.0.1 responda ao cliente 172.16.2.3.

Agora vamos considerar o caso de alguem respondendo este email. O cliente localizado na rede interna possui o IP 10.0.0.4 usando a porta 1245 e vai responder para um usuário que possui conta no servidor 172.16.2.1 porta 25:

pacote	direção	ip_orig.	ip_dest.	protocolo	porta_dest.	ação
3	out	10.0.0.4	172.16.2.1	TCP	25	permite (C)
4	in	172.16.2.1	10.0.0.4	TCP	>1245	permite (D)

Neste caso as regras do nosso roteador permitem a saida dos pacotes de email:

- A regra C permite que o cliente 10.0.0.4 envie o email para o servidor 172.16.2.1.
- A regra D permite que o servidor 172.16.2.1 responda ao cliente 10.0.0.4.

Agora vamos supor que algum localizado em uma rede externa, 172.16.2.3, usando a porta 4321 tente abrir uma conexão no servidor 10.0.0.1 na porta de x-windows, 6000:

pacote	direção	ip_orig.	ip_dest.	protocolo	porta_dest.	ação
5	in	172.16.2.3	10.0.0.1	TCP	6000	permite (D)

6	out	10.0.0.1	172.16.2.3	TCP	4321	permite (B)

Neste caso as regras do nosso roteador comportam-se da seguinte forma:

- As regras A e B permitem a entrada de pacotes SMTP.
- As regras C e D permitem a saida de pacotes SMTP.
- Já as regras B e D permitem qualquer conexão que utilizem portas >1023.

Certamente isso não é o que queremos. Para contornar esta situação devemos agregar mais um elemento as nossas regras, a porta de origem. Vejamos como fica:

regra	direção	ip_orig.	ip_dest.	protocolo	porta_orig.	porta_dest.	ação
A	in	externo	interno	TCP	>1023	25	permite
B	out	interno	externo	TCP	25	>1023	permite
C	out	interno	externo	TCP	>1023	25	permite
D	in	externo	interno	TCP	25	>1023	permite
E	ambas	qualquer	qualquer	qualquer	qualquer	qualquer	bloqueia

Agora vejamos o comportamento das regras com este novo elemento:

pacote	direção	ip_orig.	ip_dest.	protocolo	porta_origem	porta_dest.	ação
1	in	172.16.2.3	10.0.0.1	TCP	1234	25	permite (A)
2	out	10.0.0.1	172.16.2.3	TCP	25	1234	permite (B)
3	out	10.0.0.4	172.16.2.1	TCP	1245	25	permite (C)
4	in	172.16.2.1	10.0.0.4	TCP	25	1245	permite (D)
5	in	172.16.2.3	10.0.0.1	TCP	4321	6000	bloqueia (E)
6	out	10.0.0.1	172.16.2.3	TCP	6000	4321	bloqueia (E)

Como podemos observar, após a inclusão deste novo elemento de filtragem foi possivel bloquear o atque a porta x-windows.

Mas o que impede de alguem tentar abrir uma conexão na porta x-windows, 6000, usando como origem a porta 25?

Vamos analisar o que acontece nesta situação:

pacote	direção	ip_orig.	ip_dest.	protocolo	porta_orig.	porta_dest.	ação
7	in	172.16.1.2	10.0.0.1	TCP	25	6000	permite (D)
8	out	10.0.0.1	172.16.1.2	TCP	6000	25	permite (C)

Como podemos notar este pacote será permitido e a tentativa de abertura de conexão terá sucesso.

Para resolvermos este problemas temos que incluir mais um elemento em nossos filtros, desta vez vamos incluir a analise das flags dos pacotes TCP, expecificamente a flag ACK

regra	direção	ip_orig.	ip_dest.	protocolo	porta_orig.	porta_dest.	flag	ação
A	in	externo	interno	TCP	>1023	25	qualquer	permite
B	out	interno	externo	TCP	25	>1023	somente ACK	permite
C	out	interno	externo	TCP	>1023	25	qualquer	permite
D	in	externo	interno	TCP	25	>1023	somente ACK	permite
E	ambas	qualquer	qualquer	qualquer	qualquer	qualquer	qualquer	bloqueia

Como você sabe, no processo de estabelecimento de conexão TCP, sempre o primeiro pacote possui a flag ACK setada como 0. Já os demais pacotes da conexão possuem a flag ACK setada em 1.

Agora vamos analisar o que ocorre com a inclusão deste novo elemento em nosso filtro:

pacote	direção	ip_orig.	ip_dest.	protocolo	porta_orig.	porta_dest.	flag	ação
7	in	172.16.1.2	10.0.0.1	TCP	25	6000	ACK=0	bloqueia(E)

Como você pode notar agora as tentativas de abertura de conexão com origem em redes externas e destinadas a portas >1023 serão bloqueadas.

Vale lembra que é recomendável logar todas as tentativas de violação das regras, pois desta forma você poderá "saber" as tentivas de violação da sua política de segurança.

No caso de roteadores CISCO as regras A e B são traduzidas para uma única regras, pois o mesmo possui recuros de manter tabelas internas com status da conexão, este recurso está disponível em roteadores de outros fabricantes. Esta regra seria traduzida para o seguinte:

```
access-list 101 permit tcp any host <endereco_IP_serv_SMTP> eq smtp
```

Agora irei apresentar uma série de regras reais. Analise cuidadosamente cada uma de forma a entender sua aplicação. Mas lembre-se, estas regras podem não representar as necessidades da sua rede, como dito anteriormente, cada rede possui uma realidade própria.

```
!Lista 101 - Regras de Entrada
!Limpar a lista para permitir atualizacao
no access-list 101
!Restricoes aos enderecos de origem dos pacotes
!proibe enderecos iguais ao IP interno (spoofing)
!access-list 101 deny ip <Classe_C_Interno> 0.0.0.255 any log
!proibe enderecos das interfaces do router (land attack)
!access-list 101 deny ip <Endereço_IP_da_S0> 0.0.0.0<Endereço_IP_da_S0>
0.0.0.0 log
!access-list 101 deny ip <Endereço_IP_da_S1> 0.0.0.0 <Endereço_IP_da_S1>
0.0.0.0 log
!access-list 101 deny ip <Endereço_IP_da_Eth0> 0.0.0.0
<Endereço_IP_da_Eth0> 0.0.0.0 log
!proibe enderecos reservados a redes privativas (RFC-1918)
!access-list 101 deny ip 10.0.0.0 0.255.255.255 any log
!access-list 101 deny ip 172.16.0.0 0.15.255.255 any log
!access-list 101 deny ip 192.168.0.0 0.0.255.255 any log
!proibe o endereco de loopback
!access-list 101 deny ip 127.0.0.0 0.255.255.255 any log
!proibe o bradcasting (evita ping amplifying)
!access-list 101 deny ip host 255.255.255.255 any log
!Permite conexoes iniciadas internamente (TCP ACK=1)
!access-list 101 permit tcp any any established
!Restricoes e redireconamento de servicos
!proibe acesso ao TFTP
!access-list 101 deny udp any any eq 69 log
!proibe acesso ao X-Windows
!access-list 101 deny tcp any any range 6000 6005 log
!access-list 101 deny udp any any range 6000 6005 log
!proibe o acesso ao SNMP
!access-list 101 deny udp any any eq snmp log
!access-list 101 deny udp any any eq snmptrap log
!permite o acesso a porta 113/tcp e 113/udp (identd e auth), mas faz log
!essas portas sao usadas em winoob/winnuke attacks
!access-list 101 permit tcp any any eq 113 log
!access-list 101 permit udp any any eq 113 log
!HTTP apenas para o servidor HTTP
!access-list 101 permit tcp any host <Endereco_IP_Serv_WWW> eq www
!access-list 101 permit udp any host <Endereco_IP_Serv_WWW> eq 80
!SMTP apenas para o servidor de mail
!access-list 101 permit tcp any host <Endereco_IP_Serv_SMTP> eq smtp
!POP3 apenas para o servidor de POP3 (se for permitir externo)
access-list 101 permit tcp any host <Endereco_IP_Serv_POP3> eq pop3
!DNS x-fer apenas para secundarios
access-list 101 permit udp any host <Endereco_IP_Serv_DNS1> eq domain
access-list 101 permit udp any host <Endereco_IP_Serv_DNS2> eq domain
access-list 101 permit tcp host <Endereco_IP_Secund_DNS1> host
<Endereco_IP_Serv_DNS1> eq
domain
access-list 101 permit tcp host <Endereco_IP_Secund_DNS1> host
<Endereco_IP_Serv_DNS2> eq
```

```
domain
access-list 101 permit tcp host <Endereco_IP_Secund_DNS2> host
<Endereco_IP_Serv_DNS1> eq
domain
access-list 101 permit tcp host <Endereco_IP_Secund_DNS2> host
<Endereco_IP_Serv_DNS2> eq
domain
!permite pacotes TCP e UDP apenas para portas nao privativas do SO
access-list 101 permit tcp any any gt 1023
access-list 101 permit udp any any gt 1023
!proibe todo o resto, mas faz log
access-list 101 deny ip any any log

!Lista 102 - Regras de Saida
!Limpar a lista para permitir atualizacao
no access-list 102
!permite apenas enderecos internos
access-list 102 permit ip <Endereco_Classe_C_Interno> 0.0.0.255 any
!proibe qualquer outra coisa e faz log
access-list 102 deny ip any any log
```

10 - Conclusão

Como dito no inicio o assunto tratado neste texto tem por objetivo principal ajudar os administradores de rede no processo de configuração de roteadores,com sua modesta abrangência, este trabalho pretende servir como motivação ao leitor para busca de novos conhecimentos no campo da segurança da informação. Não houve aqui a pretensão de esgotar o assunto, mas sim fornecer ao leitor um texto condensado, reunindo conceitos fundamentais ao entendimento dos termos relacionados.

Desta forma procurei descrever o processo de tomada de decisão interno de um roteador de forma a permitir ao leitor um melhor entendimento dos recuros presentes em suas redes. Alem disso vale notar que a simples implementação de listas de contorle de acesso em roteadore não é o suficiente para garantir a segurança de uma rede, trata-se mais um recurso que deve ser utilizado de acordo com a política de segurança e em conjunto com os demais elementos que componhem o perímetro de segurança da rede.

Firewall é uma tecnologia que permite várias abordagens, onde podemos chamar um simples elemento com filtros de pacote de firewall até estruturas de várias camadas com zonas desmilitarizadas.

Antes de realizar qualquer implementação tenha certeza de que entendeu o funcionamento de todas as regras criadas.

O autor não dá nenhuma garantia quanto a danos que possam ser causados devido a implementação das regras discutidas neste artigo.

12 - Referencias

- Livros / Revistas

- Cisco Security Architectures, Gil Held & Kent Hundley - (ISBN: 0-07-134708-9).
- Cisco IOS Network Security - (ISBN: 1-57870-057-4).
- Cisco A Beginner's Guide - (ISBN: 0-07-212115-7).
- Cisco Access Lists - (ISBN: 0-07-212335-4).
- Introduction to Cisco Router Configuration: Student Guide.
- Building Bastion Routers Using Cisco IOS - Phrack Magazine (P55-10).
- Building Internet Firewalls (ISBN - 1-56592-124-0)

- Links
  - Melhorando a Segurança com Filtros de Pacote (http://www.absoluta.org)
  - Aumentando a segurança de redes TCP/IP através de filtros de pacote (http://www.xxx.com.br)
  - Filtros de Pacote em Roteadores Cisco (http://www.rnp.br - Março de 1998)
  - Cisco Systems (http://www.cisco.com)

- Consulta
  - [1] - Introduction to Cisco Router Configuration: Student Guide / capitulo 13.

O que você achou deste Artigo ?

Qualidade: Excelente | Medio | Fraco

Abordagem do Assunto: Objetiva | Extensa | Reduzida

Comentário:

Enviar | Limpar