FIREWALL COM IPTABLES

www.eriberto.pro.br/iptables

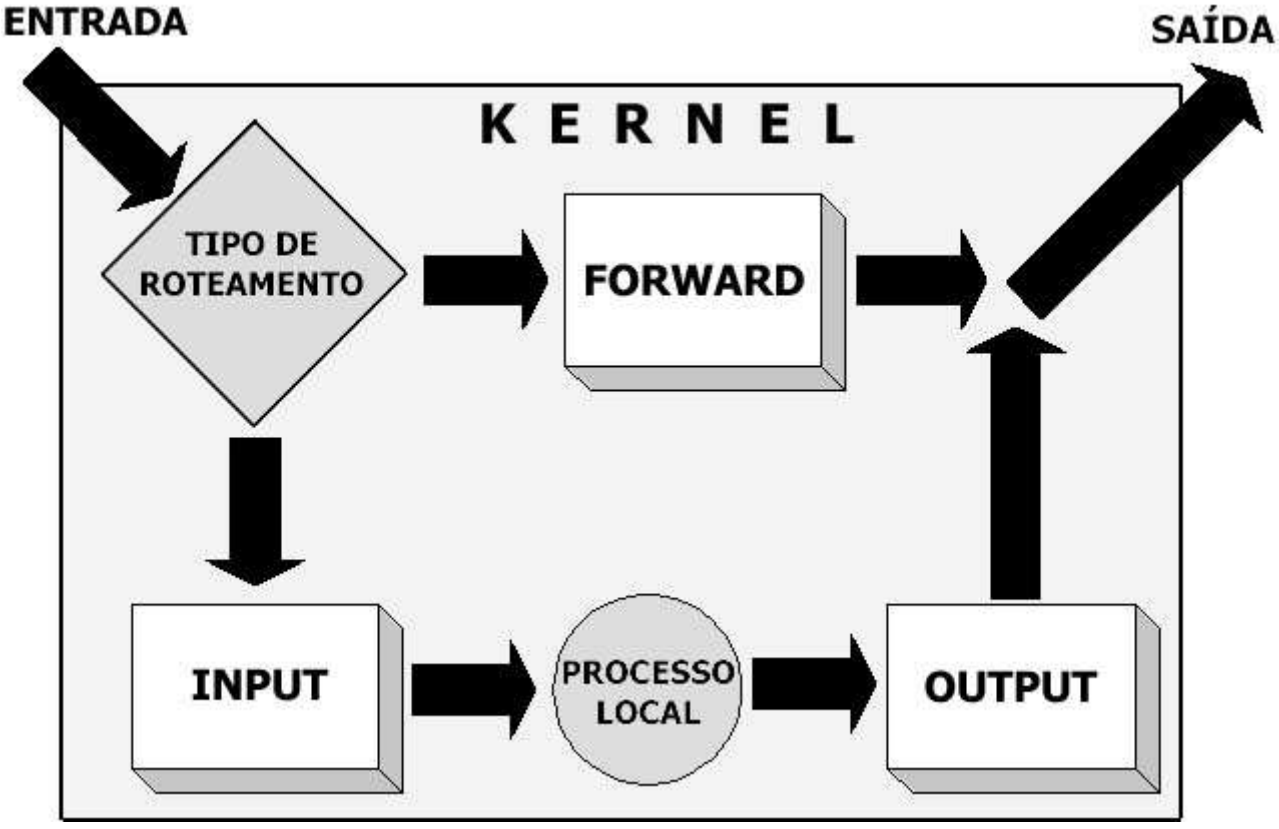
by João Eriberto Mota Filho

3. TABELAS

Tabela Filter

Vejamos o funcionamento da tabela filter (default) e as suas respectivas chains:

ESQUEMA DA TABELA FILTER



São três, as possíveis chains:

--> INPUT: utilizada quando o destino final é a própria máquina filtro;

--> OUTPUT: qualquer pacote gerado na máquina fltro e que deva sair para a rede será tratado pela chain OUTPUT;

--> FORWARD: qualquer pacote que atravessa o fltro, oriundo de uma máquina e direcionado a outra, será tratado pela chain FORWARD.

Regras de filtragem

As regras (rules) de fitragem, geralmente, são compostas assim:

#iptables [-t tabela] [opção] [chain] [dados] -j [ação]

Exemplo:

#iptables -A FORWARD -d 192.168.1.1 -j DROP

A linha acima determina que todos os pacotes destinados à máquina 192.168.1.1 devem ser descartados. No caso:

tabela: filter (é a default)

opção: -A

chain: FORWARD

dados: -d 192.168.1.1

ação: DROP

Existem outras possibilidades que fogem à sintaxe mostrada anteriormente. É o caso do comando #iptables -L, que mostra as regras em vigor.

Análise de regras com a tabela filter

Opções

As principais opções são:

-P --> Policy (política). Altera a política da chain. A política inicial de cada chain é ACCEPT. Isso faz com que o filtro, inicialmente, aceite qualquer INPUT, OUTPUT ou FORWARD. A política

pode ser alterada para DROP, que irá negar o serviço da chain, até que uma opção -A entre em vigor. O -P não aceita REJECT ou LOG. Exemplos:

#iptables -P FORWARD DROP

#iptables -P INPUT ACCEPT

-A --> Append (anexar). Acresce uma nova regra à chain. Tem prioridade sobre o -P. Geralmente, como buscamos segurança máxima, colocamos todas as chains em política DROP, com o -P e, depois, abrimos o que é necessário com o -A. Exemplos:

#iptables -A OUTPUT -d 172.20.5.10 -j ACCEPT

#iptables -A FORWARD -s 10.0.0.1 -j DROP

#iptables -A FORWARD -d www.chat.com.br -j DROP

-D --> Delete (apagar). Apaga uma regra. A regra deve ser escrita novamente, trocando-se a opção para -D. Exemplos:

Para apagar as regras anteriores, usa-se:

#iptables -D OUTPUT -d 172.20.5.10 -j ACCEPT

#iptables -D FORWARD -s 10.0.0.1 -j DROP

#iptables -D FORWARD -d www.chat.com.br -j DROP

Também é possível apagar a regra pelo seu número de ordem. Pode-se utilizar o -L para verificar o número de ordem. Verificado esse número, basta citar a chain e o número de ordem. Exemplo:

#iptables -D FORWARD 4

Isso deleta a regra número 4 de forward.

-L --> List (listar). Lista as regras existentes. Exemplos:

#iptables -L

#iptables -L FORWARD

-F --> Flush (esvaziar). Remove todas as regras existentes. No entanto, não altera a política (-P). Exemplos:

#iptables -F

```
#iptables -F FORWARD
```

Chains

As chains já são conhecidas:

INPUT --> Refere-se a todos os pacotes destinados à máquina filtro.

OUTPUT --> Refere-se a todos os pacotes gerados na máquina filtro.

FORWARD --> Refere-se a todos os pacotes oriundos de uma máquina e destinados a outra. São pacotes que atravessam a máquina filtro, mas não são destinados a ela.

Dados

Os elementos mais comuns para se gerar dados são os seguintes:

-s --> Source (origem). Estabelece a origem do pacote. Geralmente é uma combinação do endereço IP com a máscara de sub-rede, separados por uma barra. Exemplo:

```
-s 172.20.0.0/255.255.0.0
```

No caso, vimos a sub-rede 172.20.0.0. Para hosts, a máscara sempre será 255.255.255.255. Exemplo:

```
-s 172.20.5.10/255.255.255.255
```

Agora vimos o host 172.20.5.10. Ainda no caso de hosts, a máscara pode ser omitida. Caso isso ocorra, o iptables considera a máscara como 255.255.255.255. Exemplo:

```
-s 172.20.5.10
```

Isso corresponde ao host 172.20.5.10. Há um recurso para simplificar a utilização da máscara de sub-rede. Basta utilizar a quantidade de bits 1 existentes na máscara. Assim, a máscara 255.255.0.0 vira 16. A utilização fica assim:

```
-s 172.20.0.0/16
```

Outra possibilidade é a designação de hosts pelo nome. Exemplo:

```
-s www.chat.com.br
```

Para especificar qualquer origem, utilize a rota default, ou seja, 0.0.0.0/0.0.0.0, também admitindo 0/0.

-d --> Destination (destino). Estabelece o destino do pacote. Funciona exatamente como o -s, incluindo a sintaxe.

-p --> Protocol (protocolo). Especifica o protocolo a ser filtrado. O protocolo IP pode ser especificado pelo seu número (vide /etc/protocols) ou pelo nome. Os protocolos mais utilizados são udp, tcp e icmp. Exemplo:

```
-p icmp
```

-i --> In-Interface (interface de entrada). Especifica a interface de entrada. As interfaces existentes podem ser vistas com o comando #ifconfig. O -i não pode ser utilizado com a chain OUTPUT. Exemplo:

```
-i ppp0
```

O sinal + pode ser utilizado para simbolizar várias interfaces. Exemplo:

```
-i eth+
```

eth+ refere-se à eth0, eth1, eth2 etc.

-o --> Out-Interface (interface de saída). Especifica a interface de saída. Similar a -i, inclusive nas flexibilidades. O -o não pode ser utilizado com a chain INPUT.

! --> Exclusão. Utilizado com -s, -d, -p, -i, -o e outros, para excluir o argumento. Exemplo:

```
-s ! 10.0.0.1
```

Isso refere-se a qualquer endereço de entrada, exceto o 10.0.0.1.

```
-p ! tcp
```

Todos os protocolos, exceto o TCP.

--sport --> Source Port. Porta de origem. Só funciona com as opções -p udp e -p tcp. Exemplo:

```
-p tcp --sport 80
```

Refere-se à porta 80 sobre protocolo TCP.

--dport --> Destination Port. Porta de destino. Só funciona com as opções -p udp e -p tcp. Similar a --sport.

Ações

As principais ações são:

ACCEPT --> Aceitar. Permite a passagem do pacote.

DROP --> Abandonar. Não permite a passagem do pacote, descartando-o. Não avisa a origem sobre o ocorrido.

REJECT --> Igual ao DROP, mas avisa a origem sobre o ocorrido (envia pacote icmp unreachable).

LOG --> Cria um log referente à regra, em /var/log/messages. Usar antes de outras ações.

Exemplos comentados de regras de filtragem (tabela filter)

#iptables -L

Lista todas as regras existentes.

#iptables -F

Apaga todas as regras sem alterar a política.

#iptables -P FORWARD DROP

Estabelece uma política de proibição inicial de passagem de pacotes entre sub-redes.

#iptables -A FORWARD -j DROP

Todos os pacotes oriundos de qualquer sub-rede e destinados a qualquer sub-rede deverão ser descartados.

#iptables -A FORWARD -j ACCEPT

Todos os pacotes oriundos de qualquer sub-rede e destinados a qualquer sub-rede deverão ser aceitos.

#iptables -A FORWARD -s 10.0.0.0/8 -d www.chat.com.br -j DROP

Os pacotes oriundos da sub-rede 10.0.0.0 (máscara 255.0.0.0) e destinados aos hosts cujos endereços IP respondem pelo nome www.chat.com.br deverão ser descartados. Note que se a máquina possuir domínios virtuais, todos esses serão bloqueados.

#iptables -A FORWARD -s 10.0.0.0/8 -d www.chat.com.br -j REJECT

Os pacotes oriundos da sub-rede 10.0.0.0 (máscara 255.0.0.0) e destinados aos hosts cujos endereços IP respondem pelo nome www.chat.com.br deverão ser descartados. Deverá ser enviado um ICMP avisando à origem.

#iptables -A FORWARD -d www.chat.com.br -j DROP

Os pacotes oriundos de qualquer lugar e destinados aos hosts cujos endereços IP respondem pelo nome www.chat.com.br deverão ser descartados.

#iptables -A FORWARD -d 10.0.0.0/8 -s www.chat.com.br -j DROP

Os pacotes destinados à sub-rede 10.0.0.0 (máscara 255.0.0.0) e oriundos aos hosts cujos endereços IP respondem pelo nome www.chat.com.br deverão ser descartados.

#iptables -A FORWARD -s www.chat.com.br -j DROP

Os pacotes oriundos aos hosts cujos endereços IP respondem pelo nome www.chat.com.br e destinados a qualquer lugar deverão ser descartados.

#iptables -A FORWARD -s 200.221.20.0/24 -j DROP

Os pacotes oriundos da sub-rede 200.221.20.0 (máscara 255.255.255.0) e destinados a qualquer lugar deverão ser descartados.

#iptables -A FORWARD -s 10.0.0.5 -p icmp -j DROP

Os pacotes icmp oriundos do host 10.0.0.5 e destinados a qualquer lugar deverão ser descartados.

#iptables -A FORWARD -i eth0 -j ACCEPT

Os pacotes que entrarem pela interface eth0 serão aceitos.

#iptables -A FORWARD -i ! eth0 -j ACCEPT

Os pacotes que entrarem por qualquer interface, exceto a eth0, serão aceitos.

#iptables -A FORWARD -s 10.0.0.5 -p tcp --sport 80 -j LOG

O tráfego de pacotes TCP oriundos da porta 80 do host 10.0.0.5 e destinados a qualquer lugar deverá ser gravado em log. No caso, /var/log/messages.

#iptables -A FORWARD -p tcp --dport 25 -j ACCEPT

Os pacotes TCP destinados à porta 25 de qualquer host deverão ser aceitos.

Observações importantes

Impasses e ordem de processamento

As regras serão interpretadas na ordem em que aparecerem. Sempre que um pacote se adequar a uma regra, tal regra processará o pacote e a sequência iptables será finalizada naquele instante, sem que as regras seguintes atuem. Isso não se aplicará às regras terminadas com -j LOG.

Nesse caso, a regra com -j LOG irá atuar, se for o caso, e permitirá o prosseguimento da seqüencia.

Conclusão: se houver impasse entre regras, sempre valerá a primeira. Assim, entre as regras:

#iptables -A FORWARD -p icmp -j DROP

#iptables -A FORWARD -p icmp -j ACCEPT

Valerá:

#iptables -A FORWARD -p icmp -j DROP

Já entre as regras:

#iptables -A FORWARD -p icmp -j ACCEPT

#iptables -A FORWARD -p icmp -j DROP

Valerá:

#iptables -A FORWARD -p icmp -j ACCEPT

Em resumo:

ACCEPT --> Pára de processar regras para o pacote atual;

DROP --> Pára de processar regras para o pacote atual;

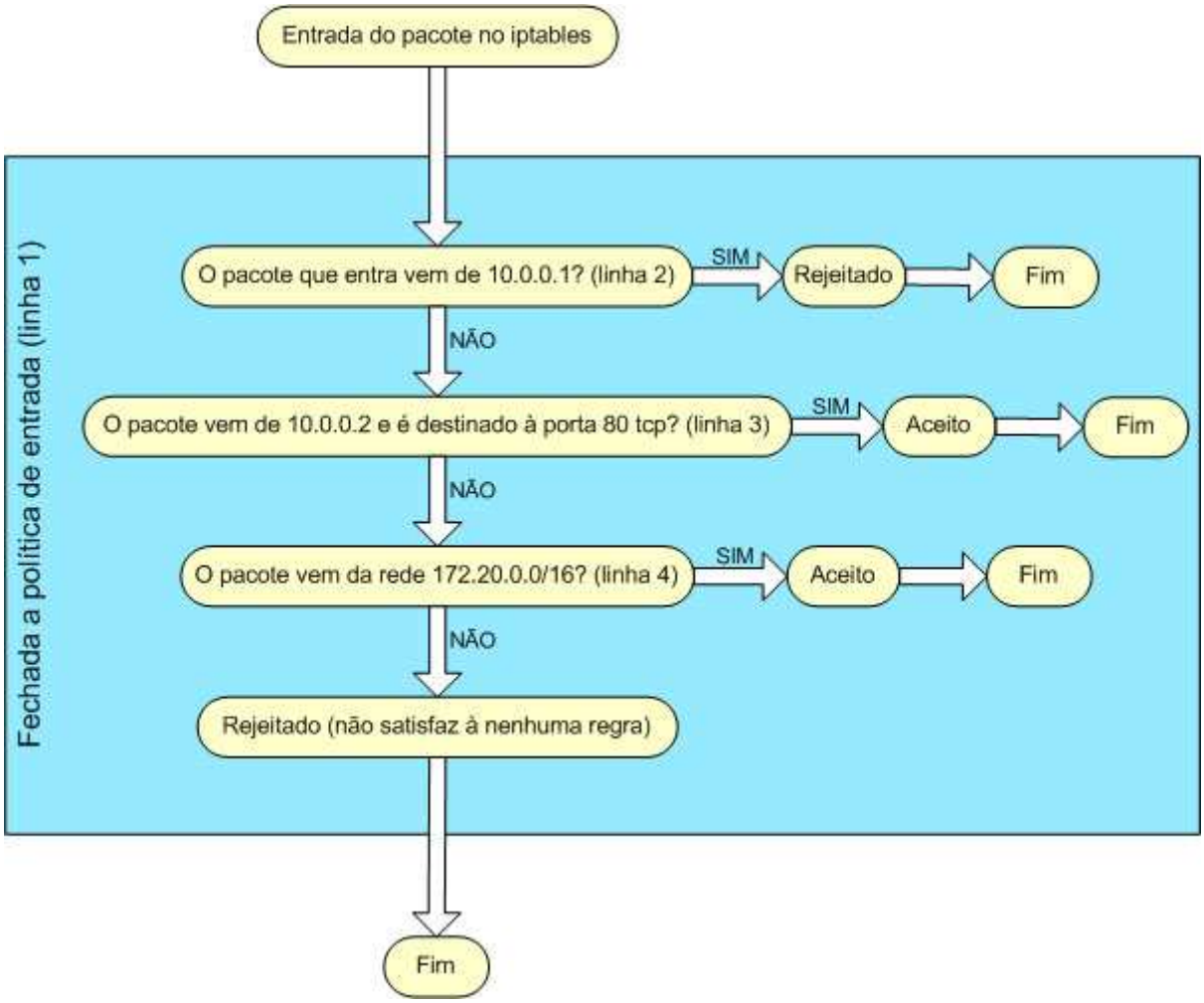
REJECT --> Pára de processar regras para o pacote atual;

LOG --> Continua a processar regras para o pacote atual;

Vamos ver um exemplo. As regras serão as seguintes:

```
iptables -P INPUT DROP
iptables -A INPUT -s 10.0.0.1 -j DROP
iptables -A INPUT -s 10.0.0.2 -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -s 172.20.0.0/16 -j ACCEPT
```

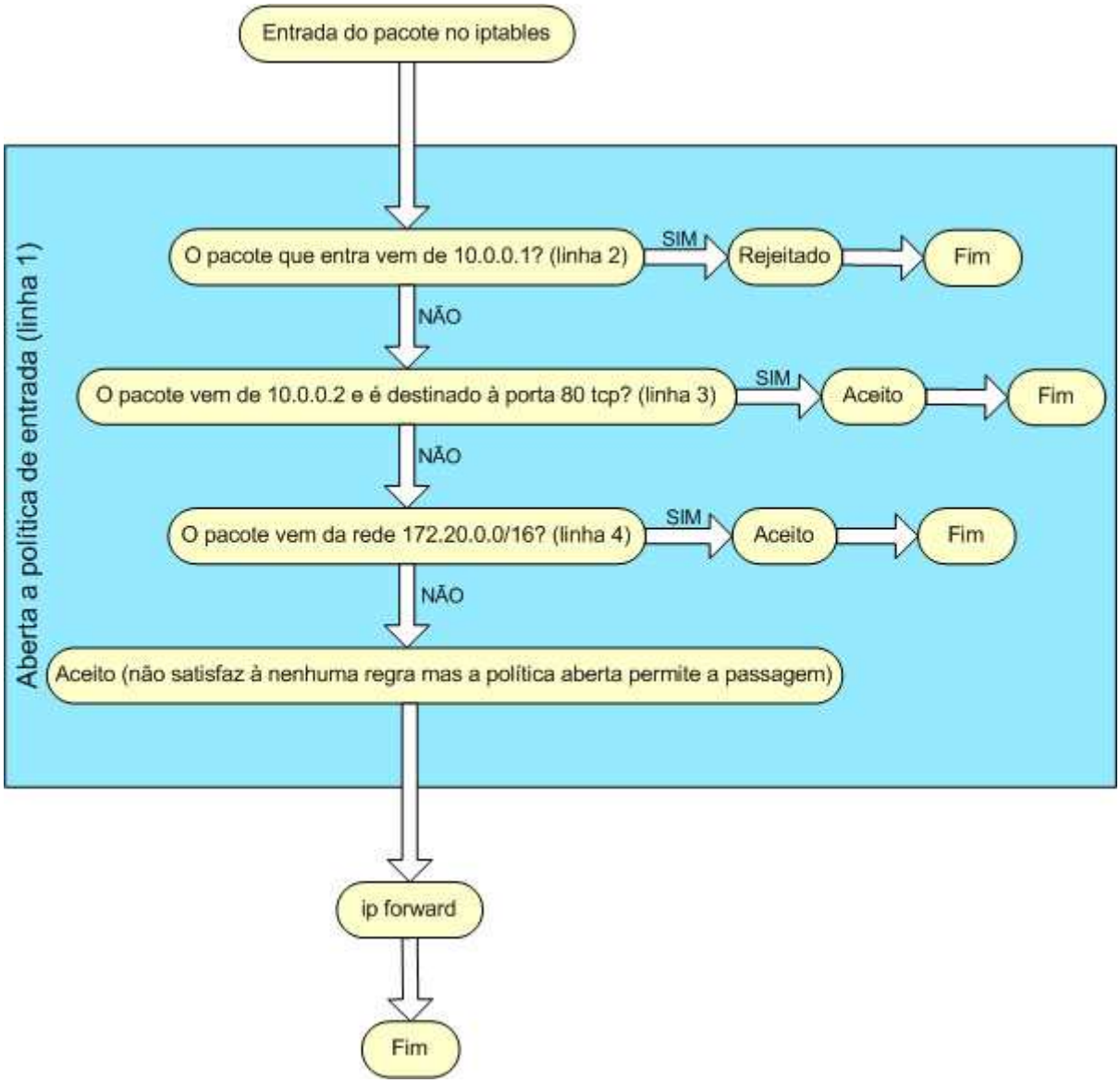
Analisando-se o fluxo de um pacote, chegamos ao diagrama:



Agora, vamos abrir a política:

```
iptables -P INPUT ACCEPT
iptables -A INPUT -s 10.0.0.1 -j DROP
iptables -A INPUT -s 10.0.0.2 -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -s 172.20.0.0/16 -j ACCEPT
```

O fluxo também mudará:



O retorno

Ao se fazer determinadas regras, devemos prever o retorno. Assim, digamos que exista a seguinte situação:

```
#iptables -P FORWARD DROP

#iptables -A FORWARD -s 10.0.0.0/8 -d 172.20.0.0/16 -j ACCEPT
```

Com as regras anteriores, fechamos todo o FORWARD e depois abrimos da sub-rede 10.0.0.0 para a sub-rede 172.20.0.0. No entanto, não tornamos possível a resposta da sub-rede 172.20.0.0 para a sub-rede 10.0.0.0. O correto, então, seria:

```
#iptables -P FORWARD DROP

#iptables -A FORWARD -s 10.0.0.0/8 -d 172.20.0.0/16 -j ACCEPT
```

```
#iptables -A FORWARD -d 10.0.0.0/8 -s 172.20.0.0/16 -j ACCEPT
```

IP FORWARD

Caso haja o envolvimento de mais de uma sub-rede, será necessário que o IP FORWARD seja ativado para que o iptables funcione corretamente. O IP FORWARD, via kernel, pode ser ativado pelo comando:

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
```

Cabe lembrar que a reinicialização do daemon de rede fará com que o roteamento seja perdido. Uma forma de deixar a regra de roteamento permanentemente ativada, resistindo a qualquer tipo de reinicialização, seria a alteração do arquivo /etc/sysctl.conf:

```
net.ipv4.ip_forward = 1
```

O carregamento

Na primeira vez em que o iptables for utilizado, poderá surgir a mensagem:

```
ip_tables: (c)2000 Netfilter core team
```

Não se preocupe. Isso é normal. É o carregamento dos módulos necessários. Caso surjam mensagens de erro, certifique-se de que o ipchains não esteja carregado. Caso esteja, descarregue-o:

```
#rmmod ipchains
```

Salvando e recuperando tudo

As regras iptables poderão ser salvas com o comando:

```
#iptables-save > arquivo
```

A recuperação poderá ser feita pelo comando:

```
#iptables-restore < arquivo
```

Um típico exemplo de carregamento de regras de iptables, após a inicialização do sistema, seria:

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
#iptables-restore < /etc/iptables.rules
```

Isso pode ser inserido no fim do arquivo /etc/rc.d/rc.local.

Nada impede que as regras sejam colocadas diretamente dentro de um shell script.

Extensões

As extensões permitem filtragens especiais, principalmente contra ataques de hackers. Os exemplos abaixo mostram como controlar os pings que atravessam o filtro:

Contra Ping

```
#iptables -A FORWARD -p icmp --icmp-type echo-request -j DROP
```

Contra Ping of Death

```
#iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT
#iptables -A FORWARD -p icmp --icmp-type echo-request -j DROP
```

É lógico que as regras anteriores podem ser utilizadas com INPUT.

Não faz parte dos meus objetivos o aprofundamento no assunto extensões. Procure por documentos especializados. Consulte a seção de links.

Mais proteção

Existe, ainda, uma regra muito importante que pode ser utilizada como segurança. É a proteção contra pacotes danificados, suspeitos ou mal formados.

```
#iptables -A FORWARD -m unclean -j DROP
```

Também pode ser utilizado com INPUT.

Network Address Translator - NAT (tabela nat)

Existem vários recursos que utilizam NAT. Os mais conhecidos são:

--> Mascaramento (masquerading)

--> Redirecionamento de portas (port forwarding ou PAT)

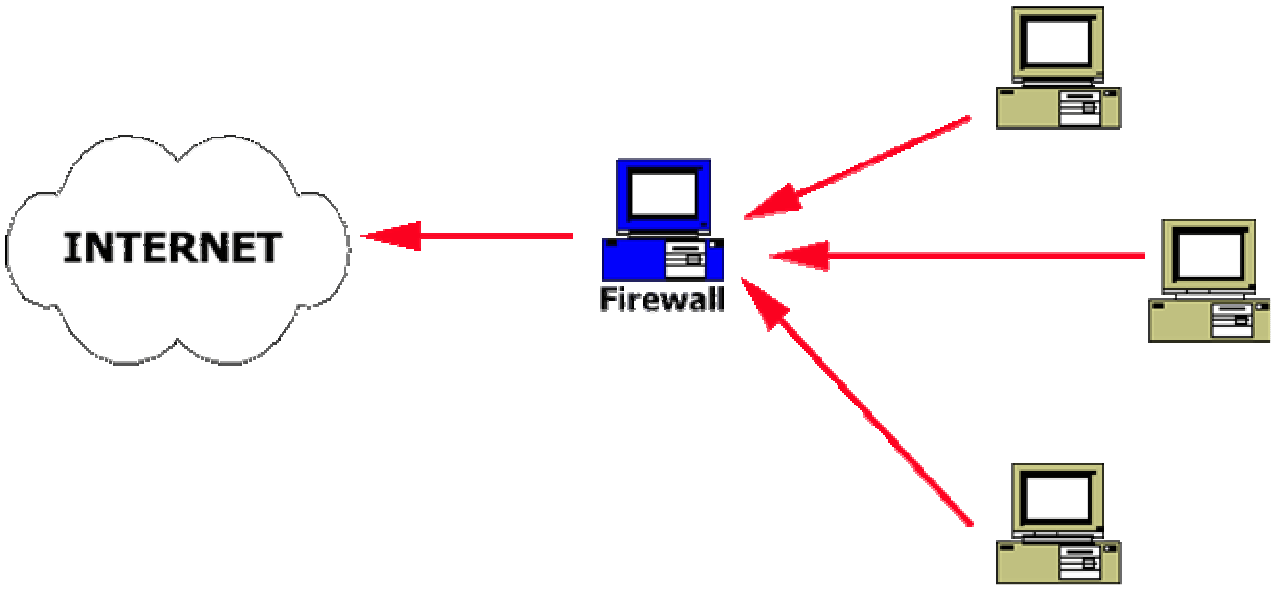
--> Redirecionamento de servidores (forwarding)

--> Proxy transparente (transparent proxy)

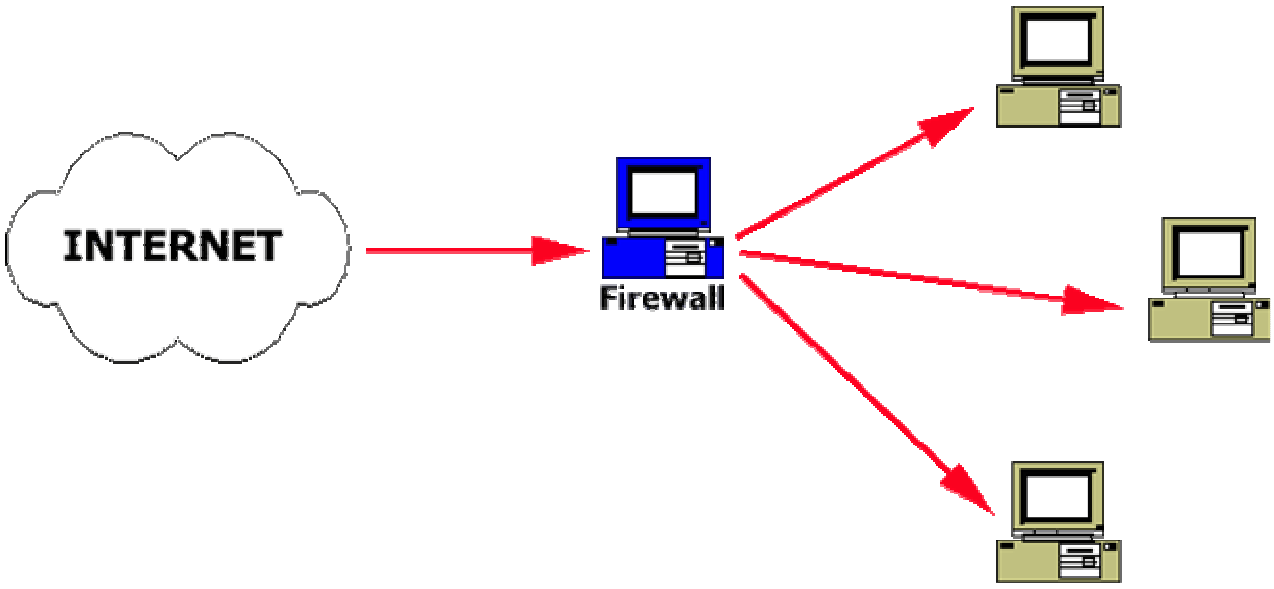
--> Balanceamento de carga (load balance)

Mascaramento

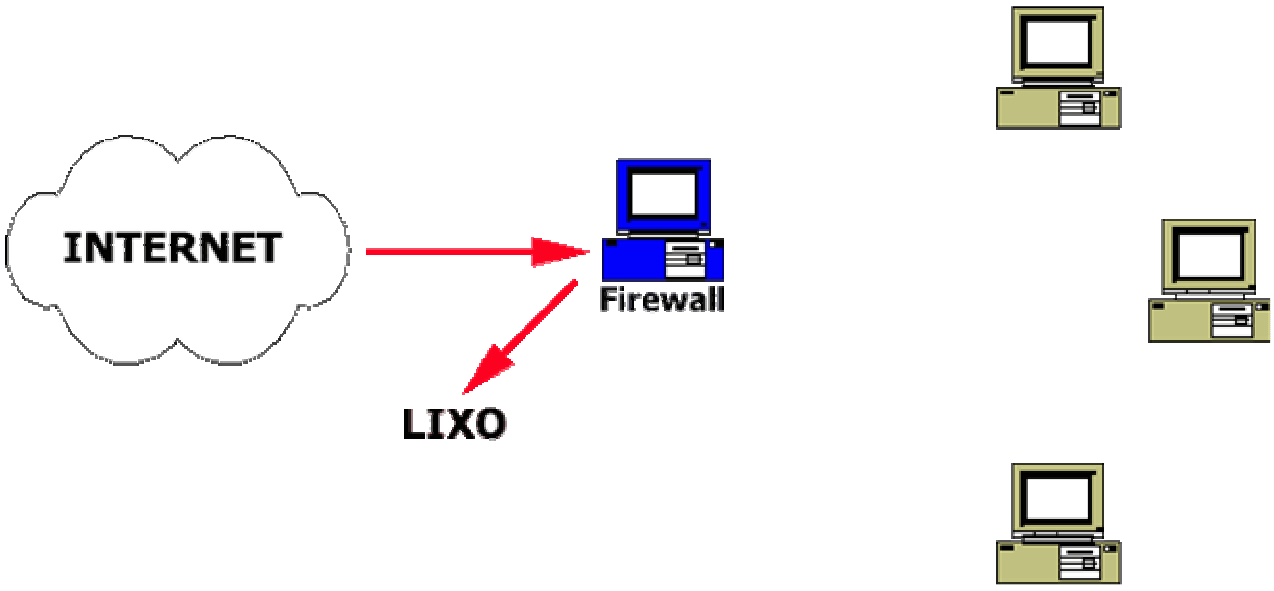
O mascaramento é uma forma de fazer NAT (Network Address Translation). Com isso, é possível fazer uma rede privada navegar na Internet. A rede solicita os dados para a máquina que faz o mascaramento. Essa busca tais dados na Internet...



...e os entrega aos solicitantes:



No entanto, um host da Internet, por vontade própria, não consegue ultrapassar o filtro que faz mascaramento, em direção à rede:



O único endereço IP que irá circular na Internet será o do filtro.

O mascaramento também possui um esquema de funcionamento. Como haverá trocas de endereços, deveremos utilizar a tabela NAT para fazer isso.

Redirecionamento de portas

O redirecionamento de portas ocorre quando desejamos alterar a porta de destino de uma requisição. Exemplo: tudo que for destinado à porta 23 de qualquer máquina, quando passar pela máquina filtro, será redirecionado para a porta 10000 de outro servidor.

Redirecionamento de servidores

Todos os pacotes destinados a um servidor serão redirecionados para outro servidor.

Proxy transparente

É a técnica que força o uso de um servidor proxy na rede.

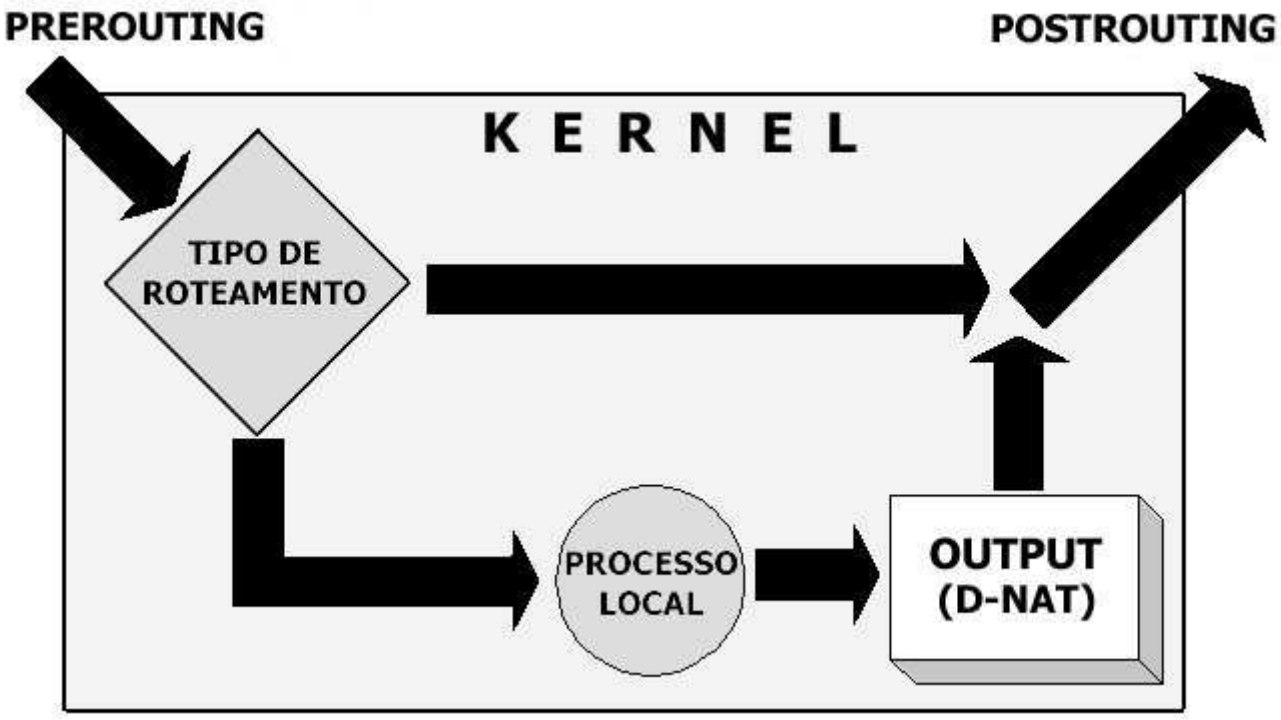
Balanceamento de carga

O balanceamento de carga (load balance) é uma técnica utilizada para distribuir carga entre servidores sincronizados. O load balance é o ato de distribuir os clientes aos servidores mais desocupados. Esse trabalho também pode ser feito por servidores DNS.

A tabela NAT

A tabela NAT funciona da seguinte forma:

ESQUEMA DA TABELA NAT



O NAT é dividido em:

--> SNAT: aplica-se quando desejamos alterar o endereço de origem do pacote. Somente a chain POSTROUTING faz SNAT. O mascaramento é um exemplo de SNAT.

--> DNAT: aplica-se quando desejamos alterar o endereço de destino do pacote. As chains PREROUTING e OUTPUT fazem DNAT. O redirecionamento de porta, o redirecionamento de servidor, o load balance e o proxy transparente são exemplos de DNAT.

Regras de NAT

Para fazer o mascaramento, deveremos, antes, carregar o módulo de NAT:

#modprobe iptable_nat

As regras mais utilizadas, além da maioria dos recursos descritos para uso com a tabela filter, contêm o seguinte:

Chains

Existem as seguintes chains:

--> **PREROUTING**: utilizada para analisar pacotes que estão entrando no kernel para sofrerem NAT. O PREROUTING pode fazer ações de NAT com o endereço de destino do pacote. Isso é conhecido como DNAT (Destination NAT);

--> **POSTROUTING**: utilizada para analisar pacotes que estão saindo do kernel, após sofrerem NAT. O POSTROUTING pode fazer ações de NAT com o endereço de origem do pacote. Isso é conhecido como SNAT (Source NAT);

--> **OUTPUT**: utilizada para analisar pacotes que são gerados na própria máquina e que irão sofrer NAT. O OUTPUT pode fazer ações de NAT com o endereço de destino do pacote. Também é DNAT.

Opções

-A --> Append (anexar).

-D --> Deletar.

Dados

-t --> Table (tabela). Estabelece a tabela a ser utilizada. A tabela default, por omissão, é filter. Para o mascaramento ou NAT será nat. Exemplo:

```
#iptables -t nat -A ...
```

--to --> utilizado para definir IP e porta de destino, após um DNAT, ou de origem, após um SNAT. Deve ser utilizado após uma ação (-j ação). Assim:

```
-j DNAT --to 10.0.0.2

-j DNAT --to 10.0.0.2:80

-j SNAT --to 172.20.0.2
```

--dport --> assim como -d define um host de destino, --dport define uma porta de destino. Deve ser utilizado antes de uma ação (-j ação). Antes de --dport, deve ser especificado um protocolo (-p). Exemplo:

```
-d 172.20.0.1 -p tcp --dport 80 -j DNAT --to 10.0.0.2
```

--sport --> assim como -s define um host de origem, --sport define uma porta de origem. Deve ser utilizado antes de uma ação (-j ação).

--to-port --> define uma porta de destino, após um REDIRECT.

Obs: A maioria dos dados básicos apresentados para a tabela filter continuam valendo. Exemplo: -p servirá para definir um protocolo de rede; -d define um host de destino.

Ações

SNAT --> Utilizado com POSTROUTING para fazer ações de mascaramento da origem.

DNAT --> Utilizado com PREROUTING e OUTPUT para fazer ações de redirecionamento de portas e servidores, balanceamento de carga e proxy transparente. Caso a porta de destino não seja especificada, valerá a porta de origem. No filtro, a porta que será redirecionada não pode existir ou estar ocupada por um daemon.

MASQUERADE --> Faz mascaramento na saída de dados.

REDIRECT --> Redireciona uma requisição para uma porta local do filtro.

Exemplos comentados de regras de filtragem (tabela nat)

#iptables -t nat -L

Mostra as regras de NAT ativas.

#iptables -t nat -F

Apaga todas as regras de NAT existentes.

#iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE

Todos os pacotes que saírem pela interface ppp0 (modem) serão mascarados. Isso dá um nível de segurança elevado à rede que está

atrás da ppp0. É uma boa regra para navegação na Internet. Note que esse tipo de mascaramento não usa SNAT.

#iptables -t nat -A POSTROUTING -d 0/0 -j MASQUERADE

Tem o mesmo efeito da regra anterior. No entanto, parece ser menos segura, pois estabelece que qualquer pacote destinado a qualquer outra rede, diferente da interna, será mascarado. A regra anterior refere-se aos pacotes que saem por determinada interface. A opção -d 0/0 poderia ser -d 0.0.0.0/0 também. É uma outra regra para navegação na Internet.

#iptables -t nat -A PREROUTING -p tcp -d 10.0.0.2 --dport 80 -j DNAT --to 172.20.0.1

Redireciona todos os pacotes destinados à porta 80 da máquina 10.0.0.2 para a máquina 172.20.0.1. Esse tipo de regra exige a especificação do protocolo. Como não foi especificada uma porta de destino, a porta 80 será mantida como destino.

#iptables -t nat -A OUTPUT -p tcp -d 10.0.0.10 -j DNAT --to 10.0.0.1

Qualquer pacote TCP, originado na máquina filtro, destinado a qualquer porta da máquina 10.0.0.10, será desviado para a máquina 10.0.0.1 .

#iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 200.20.0.1

Essa regra faz com que todos os pacotes que irão sair pela interface eth0 tenham o seu endereço de origem alterado para 200.20.0.1 .

#iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 172.20.0.1

Todos os pacotes que entrarem pela eth0 serão enviados para a máquina 172.20.0.1

#iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 172.20.0.1-172.20.0.3

Aqui haverá o load balance. Todos os pacotes que entrarem pela eth0 serão distribuídos entre as máquinas 172.20.0.1 , 172.20.0.2 e 172.20.0.3

#iptables -t nat -A PREROUTING -s 10.0.0.0/8 -p tcp --dport 80 -j REDIRECT --to-port 3128

Todos os pacotes TCP que vierem da rede 10.0.0.0, com máscara 255.0.0.0, destinados à porta 80 de qualquer host, não sairão; serão redirecionados para a porta 3128 do filtro. Isso é o passo necessário para fazer um proxy transparente. O proxy utilizado deverá aceitar esse tipo de recurso. No caso, o Squid, que aceita transparência, deverá estar instalado na máquina filtro, servindo na porta 3128.

```
#iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth1 -j SNAT
200.20.5.0/24
```

Uma situação interessante: todos os pacotes que saírem da rede 192.168.1.0 serão transformados em 200.20.5.0 .

Apesar de estarmos lidando com um filtro de pacotes, que é um roteador controlado, há a possibilidade de fazermos algumas operações dentro da mesma sub-rede. No entanto, isso tem que ser bem estudado. Muitas vezes irá exigir regras especiais de roteamento estático (comando #route).

Execução do mascaramento destinado à Internet

Por ser uma atividade perigosa, o acesso à Internet deve ser feito com um máximo grau de segurança. Assim, vejamos as regras básicas para permitir que uma rede privada navegue com um IP válido.

Primeiro exemplo: uma rede na Internet

Vamos permitir que a rede 10.0.0.0 navegue na Internet. A máquina filtro (gateway) será a 10.0.0.1. Regras:

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
#modprobe iptable_nat
#iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o ppp0 -j MASQUERADE
```

O procedimento é totalmente seguro, pois discrimina uma origem, que só poderá sair pela ppp0, de forma mascarada. Hoje em dia, o carregamento do módulo iptable_nat, na maioria das vezes, se dá automaticamente, dispensando a segunda linha.

Segundo exemplo: alguns hosts na Internet

Vamos permitir que alguns hosts, no caso, o 10.0.0.10, o 10.0.0.20 e o 10.5.2.41, naveguem na Internet. A máquina filtro (gateway) será a 10.0.0.1. Regras:

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
#iptables -t nat -A POSTROUTING -s 10.0.0.10 -o ppp0 -j MASQUERADE
#iptables -t nat -A POSTROUTING -s 10.0.0.20 -o ppp0 -j MASQUERADE
#iptables -t nat -A POSTROUTING -s 10.5.2.41 -o ppp0 -j MASQUERADE
```

Execução de FTP

Para executar sessões de FTP, será necessário o carregamento de dois módulos:

```
#insmod ip_conntrack_ftp
#insmod ip_nat_ftp
```

Salvando e recuperando regras

As regras de iptables devem ser salvas com o comando iptables-save e carregadas com iptables-restore. O roteamento estático e o carregamento dos módulos FTP devem ser feitos separadamente. Apenas para ilustrar, vamos executar o salvamento e a recuperação das regras.

Salvamento de regras

```
--> Criando as regras

#iptables -t nat -A POSTROUTING -s 10.0.0.10 -o ppp0 -j MASQUERADE
#iptables -t nat -A POSTROUTING -s 10.0.0.20 -o ppp0 -j MASQUERADE
#iptables -t nat -A POSTROUTING -s 10.5.2.41 -o ppp0 -j MASQUERADE

--> Salvando

#iptables-save > /etc/iptables.rules

--> Recuperação
```

Um script de recuperação, inicializado pelo /etc/rc.d/rc.local, poderia ter a seguinte estrutura:

```
#!/bin/bash
echo 1 > /proc/sys/net/ipv4/ip_forward
modprobe iptable_nat
iptables-restore < /etc/iptables.rules
insmod ip_conntrack_ftp
insmod ip_nat_ftp
```

Tabelas Filter e NAT atuando em conjunto

As tabelas filter e nat podem atuar em conjunto, funcionando em paralelo. Há de se ter cuidado pois, como já disse, elas atuam em paralelo, como duas pilhas que serão executadas ao mesmo tempo. Assim sendo, se tivermos as regras:

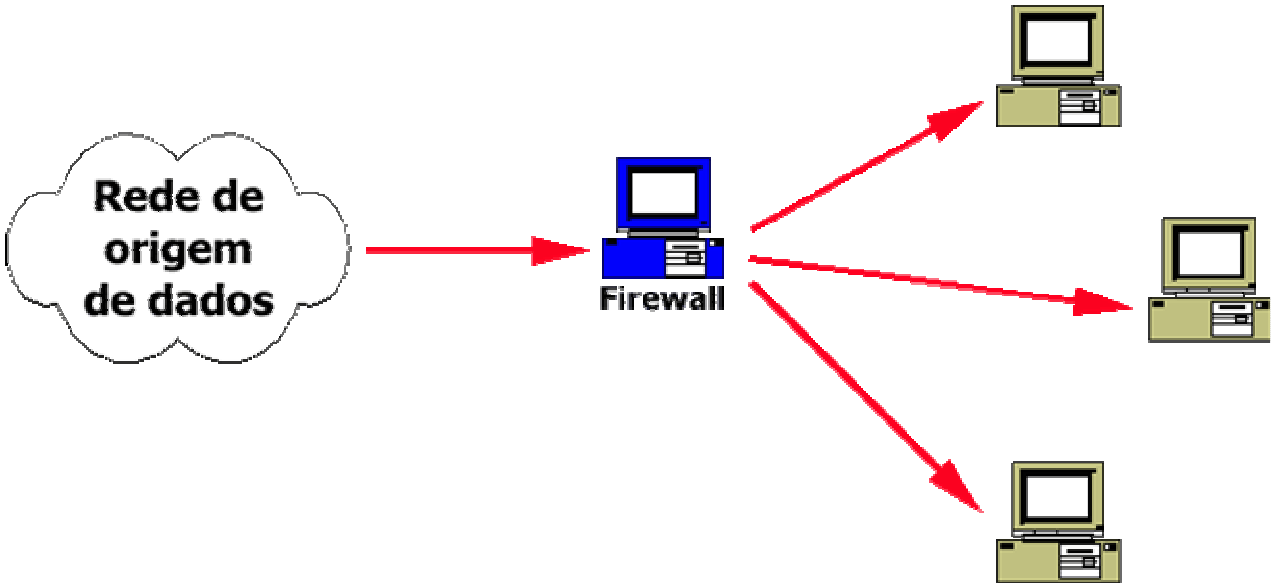
```
#iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o eth0 -j MASQUERADE
#iptables -A FORWARD -j DROP
```

Apesar da primeira (tabela nat) possibilitar a navegação mascarada da rede 10.0.0.0 na Internet, essa navegação não ocorrerá, pois a segunda regra (tabela filter) irá barrar o forward entre as redes.

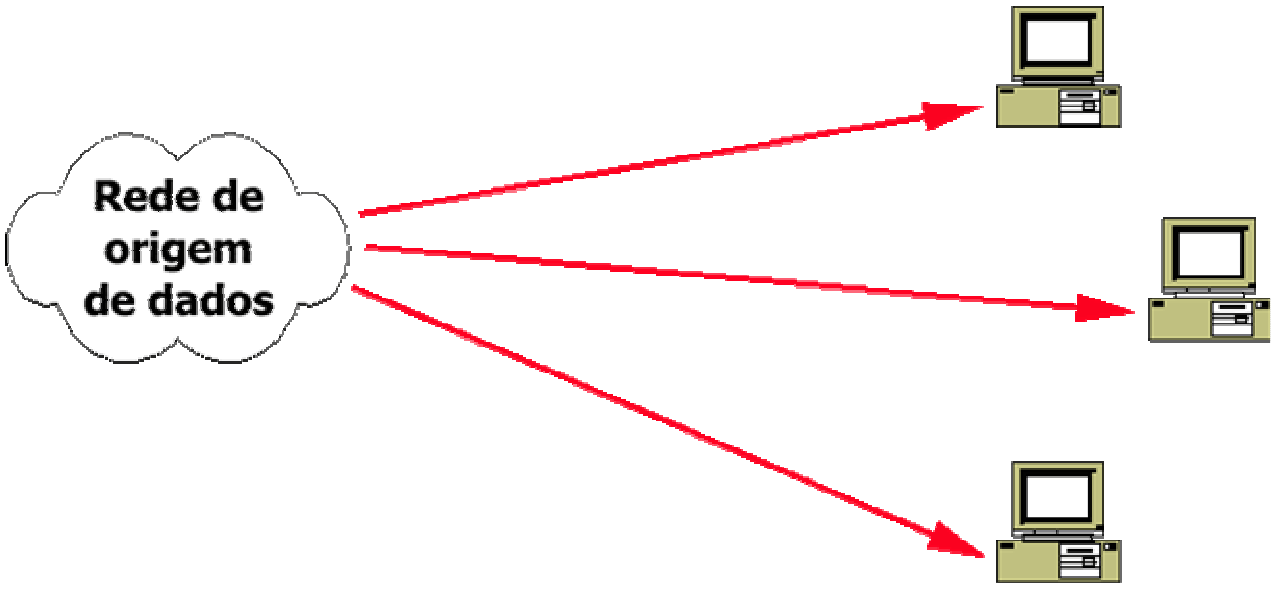
Topologias de filtros de pacotes

O posicionamento de um filtros de pacotes dentro da rede é de extrema importância para ela. Há duas possibilidades básicas para a utilização de um filtros de pacotes: filtro isolado e filtro incorporado, sendo este último o mais inseguro e menos desejável.

O filtro será isolado quando estiver entre máquinas, com função exclusiva de filtro:



O filtro será incorporado quando não houver uma máquina isolada como filtro. Nesse caso, as máquinas da rede deverão estabelecer, individualmente, as suas próprias regras de filtragem. É o sistema mais inseguro:



Nada impede que os dois sistemas sejam utilizados em parceria, oferendo-se assim um alto grau de segurança à rede. Cabe ressaltar que no filtro incorporado há maior nível de insegurança, uma vez que outros processos rodam junto com o filtro.