ESTUDO SNAT E DNAT

## . [Controlando o que será submetido a NAT](https://netfilter.org/documentation/HOWTO/pt/NAT-HOWTO.html#toc5)

Você precisa criar regras NAT as quais dirão ao kernel quais conexões serão alteradas, e como alterá-las. Para fazer isso, utiliza-se o versátil iptables e dizemos a ele para alterar a tabela NAT ao especificar a opção `-t nat'.

As regras da tabela NAT contêm três listas chamadas `chains': cada regra é examinada em ordem, até que alguma delas corresponda-se com o pacote analizado. As três 'chains' são chamadas PREROUTING (para Destination NAT, quando os pacotes estão prestes a entrar), POSTROUTING (para Source NAT, assim que os pacotes saem), e OUTPUT (para Destination NAT de pacotes gerados localmente.

O diagrama abaixo ilustraria isso perfeitamente se eu tivesse talento artístico: :)

\_\_\_\_\_ \_\_\_\_\_

/ \ / \

PREROUTING -->[Decisão ]----------------->POSTROUTING----->

\D-NAT/ [de Roteamento] \S-NAT/

| ^

| \_\_|\_\_

| / \

| | OUTPUT|

| \D-NAT/

| ^

| |

----> Processamento Local ----

Em cada um dos pontos acima, quando um pacote passa, é verificado a conexão com a qual ele está associado. Se é uma nova conexão, é verificada a chain correspondente na tabela NAT para ver o que fazer com a mesma. A resposta dada será idêntica para todos os outros pacotes relacionados com tal conexão.

## 5.1 [Seleção simples utilizando iptables](https://netfilter.org/documentation/HOWTO/pt/NAT-HOWTO.html#toc5.1)

iptables tem um número de opções padrão conforme a lista abaixo. Todas as opções com dois hífens (--) podem ser abreviadas, desde que o iptables ainda possa diferenciá-las das demais opções disponíveis. Se seu kernel tem suporte a iptables via módulos, você precisará carregar o módulo ip\_tables.o antes com o seguinte comando: `insmod ip\_tables'.

A opção mais importante é a seleção da tabela com `-t'. Para todas as operações NAT, será necessária a opção `-t nat' para a tabela NAT. A segunda mais importante é a opção '-A', que adiciona uma nova regra no fim da chain (`-A POSTROUTING'), ou `-I' para adiconá-la no início (eg. `-I PREROUTING').

Você pode especificar a origem (`-s' or `--source') e o destino (`-d' or `--destination') dos pacotes que sofrerão NAT. Tais opções podem ser seguidas de um simples endereço IP ( 192.168.1.1), um nome (www.gnumonks.org), ou um endereço de rede (192.168.1.0/24 ou 192.168.1.0/255.255.255.0).

Também podem ser especificadas as interfaces de entrada (`-i' ou `--in-interface') ou de saída (`-o' ou `--out-interface'). Qual das interfaces que poderá ser especificada dependerá da chain que receberá a regra: em PREROUTING você pode selecionar apenas a interface de entrada, e em POSTROUTING (e OUTPUT) apenas a interface de saída é selecionada. Se você fizer a opção errada, o iptables restornará um erro.

## 5.2 [Pontos mais específicos sobre o tratamento dos pacotes](https://netfilter.org/documentation/HOWTO/pt/NAT-HOWTO.html#toc5.2)

Foi dito acima que podem ser especificados endreços de origem e destino. Se o endereço de origem foi omitido, a regra servirá para qualquer origem. Se o endereço de destino for omitido, a regra serviá para qualquer destino.

Um protocolo específico também pode ser indicado (`-p' or `--protocol'), como o TCP ou o UDP. Ao selecionar um protocolo, apenas pacotes do mesmo se encaixarão na regra. A principal razão para selecionar um protocolo, é que isso permite algumas opções extras. Especificamente as opções `--source-port' e `--destination-port' (abreviadas como `--sport' and `--dport').

Essas opções permitem especificar que apenas pacotes com certa porta de origem e destino se encaixarão na regra. Isso é útil para redirecionamento de requisições web (porta TCP 80 ou 8080).

Todas essas opções devem seguir a `-p' (a qual carrega uma biblioteca compartilhada para aquele protocolo). Podem ser usados os números das portas ou um nome do arquivo /etc/services.

Todas as outras formas de se selecionar um pacote estão descritas em detalhes na página de manual do iptables (man iptables).

Agora já sabemos como escolher os pacotes a serem tratados. Para completar, precisamos dizer ao kernel o que fazer com estes pacotes.

**6.1**[**Source NAT**](https://netfilter.org/documentation/HOWTO/pt/NAT-HOWTO.html#toc6.1)

Você quer fazer Source NAT; mudar o endereço de origem das conexões para algo diferente. Isso é feito na chain POSTROUTING, logo antes de que o pacote saia da máquina. Isso é um detalhe muito importante, pois significa que qualquer tarefa da sua máquina Linux (routing, packet filtering) verá o pacote inalterado. Isso também significa que a opção `-o'(interface de saída) pode ser utilizada.

Source NAT é especificado com `-j SNAT', e a opção `--to-source' demonstra um endereço IP, um range de endereços IP, e uma porta opcional ou um range de portas (apenas para os protocolos TCP e UDP).

## Mudando o endereço de origem para 1.2.3.4.

# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4

## Mudando o endereço de origem para 1.2.3.4, 1.2.3.5 ou 1.2.3.6

# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4-1.2.3.6

## Mudando o endereço de origem para 1.2.3.4, portas 1-1023

# iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT --to 1.2.3.4:1-1023

**Masquerading**

Há um caso especial de Source NAT chamado masquerading, que só deve ser utilizado para endereços IP dinâmicos, como as conexões dialup (para endereços IP estáticos, utilize SNAT descrito acima).

No masquerading, não é necessário explicitar o endereço de origem: será utilizado o endereço de origem da interface de saída do pacote. Mas o mais importante é que se o link cair, o endereço de origem que estava sendo utilizado é descartado, dando lugar ao novo endereço de origem da interface quando o link for restabelecido.

## Mascarando tudo que sai pela interface ppp0.

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

**6.2**[**Destination NAT**](https://netfilter.org/documentation/HOWTO/pt/NAT-HOWTO.html#toc6.2)

DNAT é feito na chain PREROUTING, assim que o pacote chega; isso significa que todas as outras tarefas da sua máquina Linux (routing, packet filtering) verão o pacote já indo para seu destino verdadeiro. A opção `-i' (interface de entrada) pode ser utilizada.

Para alterar o destino de pacotes gerados localmente, a chain OUTPUT deve ser utilizada, mas isso não é muito utilizado (além disso, mudar o destino para outro local que não a própria máquina Linux ainda não funciona -- Abril 2001).

Destination NAT é especificada utilizando `-j DNAT', e a opção `--to-destination' especifica um endereço IP, um range de IPs, e uma porta opcional ou um range de portas (apenas para protocolos TCP e UDP).

## Mudando destino para 5.6.7.8

# iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 5.6.7.8

## Mudando destino para 5.6.7.8, 5.6.7.9 ou 5.6.7.10.

# iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 5.6.7.8-5.6.7.10

## Mudando destino do tráfego web para 5.6.7.8, porta 8080.

# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth0 \

-j DNAT --to 5.6.7.8:8080

## Redirecionar pacotes locais com destino a 1.2.3.4 para loopback.

# iptables -t nat -A OUTPUT -d 1.2.3.4 -j DNAT --to 127.0.0.1

**Redirecionamento**

Há um caso especializado de Destination NAT chamado redirecionamento: é uma simples conveniência, a qual equivale a fazer DNAT para o endereço da própria interface de entrada.

## Mandando tráfego web da porta-80 para nosso squid proxy (transparente)

# iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 \

-j REDIRECT --to-port 3128

Lembre-se que o squid precisa ser configurado para saber que está sendo um proxy transparente!

**6.3**[**Algumas curiosidades menos utilizadas...**](https://netfilter.org/documentation/HOWTO/pt/NAT-HOWTO.html#toc6.3)

Há alguns refinamentos do NAT os quais a maioria das pessoas nunca utilizará. Eles estão aqui documentados apenas para os curiosos.

**Seleçao de múltiplos endereços em um range**

Se um range de endereços IP é dado, o IP escolhido para uso é o que está sendo menos utilizado pelas conexões conhecidas. Isso porvê um balanceamento de carga (load-balancing) bem primitivo.

**Criando mapeamentos NAT nulos**

Você pode utilizar `-j ACCEPT' para aceitar uma conexão sem que haja nenhuma espécie de NAT.

**Comportamento padrão do NAT**

O comportamento padrão é alterar a conexão o mínimo possível, modificando apenas o definido pelo usuário nas suas regras. Isso significa que as portas não serão remapeadas a não ser que as regras mandem que elas sejam.

**Mapeamento de portas de origem implícito**

Mesmo quando nenhum tipo de NAT é requisitado para uma conexão, a alteração da porta de origem pode ocorrer de forma implícita, quando uma conexão requisita uma porta que está em uso. Considere um caso de masquerading, que gera esse tipo de situação:

1. Uma conexão web estabelecida pela máquina 192.1.1.1 vinda da porta 1024 para www.netscape.com porta 80.
2. Tal conexão é mascarada pela máquina de masquerading a fim de usar o endreço de IP de origem 1.2.3.4, que é o endereço de origem da máquina que realiza o masquerading.
3. A máquina que faz o masquerading tenta realizar a conexão web com www.netscape.com porta 80 vinda de 1.2.3.4 (endereço de origem da interface de saída) porta 1024.
4. O código de NAT alterará a porta de origem da segunda conexão para a porta 1025, assim as duas conexão não conflitarão.

Quando esse mapeamento de origem implícito ocorre, as portas são divididas em três classes:

* Portas menores que 512
* Ports entre 512 e 1023
* Portas acima de 1024.

Uma porta NUNCA será mapeada implicitamente para uma classe diferente.

**O que ocorre quando NAT falha**

Se não há formas de mapear uma conexão conforme requisitado pelo usuário, a mesma será descartada (DROP). Isso também se aplica a pacotes que não foram classificados como parte de qualquer conexão, devido a malformação, ou falta de memória da máquina, etc.

**Mapeamentos múltiplos, overlap e conflitos**

Você pode ter regras NAT que mapeiam pacotes para o mesmo range; o código NAT suficientemente inteligente para evitar conflitos. Não há nenhum problema em mapear os endereços de origem 192.168.1.1 e 192.168.1.2 para 1.2.3.4.

Além disso, você pode mapear para endereços IP reais e utilizados, desde que tais endereços passem pela máquina responsável pelo mapeamento. Então, se você tem uma rede válida (1.2.3.0/24), mas possui uma rede interna que utilize um IP privado (192.168.1.0/24), você pode realizar SNAT do endereço 192.168.1.0/24 para a rede 1.2.3.0, sem medo de quaisquer conflitos:

# iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth1 \

-j SNAT --to 1.2.3.0/24

A mesma lógica se aplica aos endereços utilizados pela própria máquina que realiza NAT: essa é a forma que o masquerading funciona (compartilhando o endereço da interface de saída entre os pacotes mascarados e os gerados localmente).

Você também pode mapear os mesmos pacotes para alvos distintos, e estes serao compartilhados. Por exemplo, se você não quer mapear NADA para o endereço 1.2.3.5, você faria:

# iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth1 \

-j SNAT --to 1.2.3.0-1.2.3.4 --to 1.2.3.6-1.2.3.254

**Alterando destino de pacotes gerados localmente**

Se o destino de um pacote gerado localmente é alterado (pela chain OUTPUT), e isso leva o pacote a passar por uma interface diferente, e então o endereço de origem também é alterado para o endereço da interface de saída. Por exemplo, mudando o destino de um pacote loopback para sair por eth0 irá resultar na mudança do endereço de origem de 127.0.0.1 para o endereço de eth0; ao contrário dos outros mapeamentos de origem (que são feitos depois de tudo, na chain POSTROUNTING) esse é realizado imediatamente. Naturalmente, todos esses mapeamentos são revertidos no momento que os pacotes de resposta chegam.