Iptables

Introdução

Começamos por limpar todas as regras anteriores e configurar política padrão:

```
sudo iptables -F
sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT ACCEPT
```

Na política padrão partimos do principio que todo o tráfego de entrada deve ser descartado e o de saída deve ser aceite.

Regras para restrições

Definimos as seguintes regras de forma a permitir *ping* apenas da máquina *gcc* (10.101.151.5) e sub-rede 10.101.85.0/24:

```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -s
    10.101.151.5 -j ACCEPT
sudo iptables -A INPUT -p icmp --icmp-type echo-request -s
    10.101.85.0/24 -j ACCEPT
```

Estas regras consistem em adicionar exceções á política de entrada para pacotes *icmp* dirigidos aos IP relevantes. Para testar o seu funcionamento testamos a utilização do comando *ping* em várias máquinas, permitidas e não permitidas, e verificamos que apenas a máquina *gcc* e as máquinas na sub-rede 10.101.85.0/24 conseguiram efetuar o *ping* com sucesso.

Permitir conexões do cliente $Tintolmarket$ de qualquer origem para o servidor $TintolmarketServer$ no porto default (12345):

```
sudo iptables -A INPUT -p tcp --dport 12345 -j ACCEPT
```

Tal como nas regras anteriores adicionamos uma exceção á regra de entrada, desta vez aceitando pacotes *tcp* que tenham como destino o porto 12345. Como teste, verificamos todas as máquinas na rede e concluímos que conseguiam ligar-se ao servidor através do cliente $Tintolmarket$.

Permitir ligações *ssh* apenas da máquina *gcc* e sub-rede $DC1$, $DC2$ e $DC3$ (máscara 255.255.255.224):

```
sudo iptables -A INPUT -p tcp --dport 22 -s 10.101.151.5 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 22 -s 10.121.52.0/27 -j
    ACCEPT
sudo iptables -A INPUT -p tcp --dport 22 -s 10.101.52.0/27 -j
    ACCEPT
```

Adicionamos mais duas exceções na política de entrada, desta vez para ligações no porto 22 (porto default *ssh*). Para testar o seu funcionamento iniciamos o servidor *ssh* na máquina de teste, com o comando:

```
sudo /usr/sbin/sshd -D
```

E utilizamos o em várias máquinas:

```
ssh -X ip-destino
```

Garantindo que apenas as máquinas permitidas conseguiram efetuar a ligação com sucesso.

Regras para serviços utilizados

Limitar *ping* à frequência máxima de 3 *pings* por segundo:

```
sudo iptables -A OUTPUT -p icmp --icmp-type echo-request -m limit
    --limit 3/second --limit-burst 3 -d 10.0.0.0/8 -j ACCEPT
sudo iptables -A OUTPUT -p icmp --icmp-type echo-request -j DROP
sudo iptables -A INPUT -p icmp --icmp-type echo-reply -s 10.0.0.0/8
     -j ACCEPT
```

O conceito destas regras consiste em aceitar tráfego *icmp* a um limite máximo de 3 pacotes por segundo para endereços na mesma sub-rede local (10.0.0.0/8) e de seguida fazer *drop* de todos os pacotes *icmp* que não se enquadrem nesta regra. É também necessário aceitar as respostas, pelo que se deve adicionar também uma nova regra de *input*. O teste consistiu em fazer *ping* a outras máquinas e, através da $flag\ -i$ variar o intervalo entre *pings*. Verificamos que para intervalos inferiores a 0.33 segundos o envio de alguns pacotes era negado.

Permitir *ssh* apenas para a máquina *gcc*:

```
sudo iptables -A OUTPUT -p tcp --dport 22 -d 10.101.151.5 -j ACCEPT
sudo iptables -A OUTPUT -p tcp --dport 22 -j DROP
```

Novamente, adicionamos uma regra que permite aceitar ligações no porto 22 do endereço da máquina *gcc* e outra para descartar ligações *ssh* em qualquer outro endereço. Para testar utilizamos novamente o servidor *ssh* em várias máquinas, incluindo *gcc*, e cliente na máquina de teste ,variando o campo $ip - destino$, verificamos que apenas era possível efetuar ligação com a máquina *gcc*.

Regras default

Por fim, executamos as regras mencionadas no enunciado, começando por permitir tráfego *loopback*:

```
sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A OUTPUT -o lo -j ACCEPT
```

E permitir tráfego relacionado com uma ligação já estabelecida:

```
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j
ACCEPT
sudo iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j
ACCEPT
```