



Segurança e Confiabilidade

Relatório Projeto 3

IPTables

Engenharia Informática
2018/2019

Grupo 17

Diogo Nogueira, Nº 49435

Filipe Capela, Nº 50296

Filipe Silveira, Nº 49506

Índice

Regras do comando IPTables.....	2
Método de teste e observações	4
Script.....	4
Computadores utilizados	4
Testes realizados	4

Regras do comando IPTables

```
#!/bin/sh
```

```
#<-----Limpar as tables----->
```

```
sudo /sbin/iptables -F INPUT ACCEPT
sudo /sbin/iptables -F FORWARD ACCEPT
sudo /sbin/iptables -F OUTPUT ACCEPT
sudo /sbin/iptables -F
sudo /sbin/iptables -X
```

```
#<-----Aceitar apenas pings vindos do gcc----->
```

```
sudo /sbin/iptables -A INPUT -s 10.101.151.5 -p ICMP --icmp-type 8 -j ACCEPT
sudo /sbin/iptables -A INPUT -p ICMP --icmp-type 8 -j REJECT
```

```
#<-----Aceitar todas as ligações de serviços cruciais----->
```

```
sudo /sbin/iptables -A INPUT -p tcp -s 10.121.52.14 -j ACCEPT
sudo /sbin/iptables -A INPUT -p tcp -s 10.121.52.15 -j ACCEPT
sudo /sbin/iptables -A INPUT -p tcp -s 10.101.52.16 -j ACCEPT
sudo /sbin/iptables -A INPUT -p tcp -s 10.121.72.23 -j ACCEPT
sudo /sbin/iptables -A INPUT -p tcp -s 10.101.85.6 -j ACCEPT
sudo /sbin/iptables -A INPUT -p tcp -s 10.101.85.138 -j ACCEPT
sudo /sbin/iptables -A INPUT -p tcp -s 10.101.85.24 -j ACCEPT
sudo /sbin/iptables -A INPUT -p tcp -s 10.101.148.1 -j ACCEPT
sudo /sbin/iptables -A INPUT -p tcp -s 10.101.85.134 -j ACCEPT
```

```
sudo /sbin/iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
```

```
#<-----Aceitar apenas ligações de máquinas na sub-rede----->
```

```
sudo /sbin/iptables -A INPUT -p tcp ! -s 10.101.148.0/23 -j DROP
```

#<-----Permitir que apenas seja possível fazer pings para a sub-rede----->

```
sudo /sbin/iptables -A OUTPUT -p icmp -m icmp -d 10.101.149.0/23 --icmp-type 8 -m limit --
limit 2/second --limit-burst 2 -j ACCEPT
sudo /sbin/iptables -A OUTPUT -p icmp -j DROP
```

#<-----Pontos ii) ii) e iv) do enunciado----->

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

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

#Ver lista final

```
sudo /sbin/iptables -L
```

Nota: Abaixo podemos ver o estado final das tabelas após ser executado o script que contém todas as linhas mencionadas nesta secção.

```
fc50296@linux:~/Desktop$ sudo /sbin/iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     icmp -- gcc.alunos.di.fc.ul.pt anywhere        icmp echo-request
REJECT     icmp -- anywhere              anywhere        icmp echo-request reject-with icmp-port-unreachable
ACCEPT     tcp -- fc-dc04.fc.ul.pt      anywhere
ACCEPT     tcp -- fc-dc05.fc.ul.pt      anywhere
ACCEPT     tcp -- 10.101.52.16          anywhere
ACCEPT     tcp -- 10.121.72.23          anywhere
ACCEPT     tcp -- iate.di.fc.ul.pt     anywhere
ACCEPT     tcp -- falua.di.fc.ul.pt    anywhere
ACCEPT     tcp -- luna.alunos.di.fc.ul.pt anywhere
ACCEPT     tcp -- submarino.alunos.di.fc.ul.pt anywhere
ACCEPT     tcp -- proxy.alunos.di.fc.ul.pt anywhere
ACCEPT     tcp -- anywhere             anywhere        tcp dpt:ssh ctstate NEW,ESTABLISHED
DROP       tcp -- !10.101.148.0/23      anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     icmp -- anywhere            10.101.148.0/23    icmp echo-request limit: avg 2/sec burst 2
DROP       icmp -- anywhere            anywhere
```

Figura 1 - Estado final das tabelas.

Método de teste e observações

Script

De forma a agilizar o processo de criação das regras, foi criado o script *iptables.sh* que, quando executado, percorre uma lista de comandos que alteram as iptables do computador onde é executado.

Este pode ser corrido escrevendo no terminal ***“sh iptables.sh”***.

Computadores utilizados

Como está demonstrado na secção acima, todos os comandos estão divididos tendo em conta a função que desempenham, de modo a facilitar a sua leitura e execução.

Para testar o grupo teve ao seu dispor dois computadores, localizados na sala de aula 1.3.12, ou seja, computadores que fazem parte da sub-rede requerida, 10.101.149.0/23.

Visando simplificar a compreensão, estes denominam-se Computador A (10.101.149.63) e Computador B (10.101.149.64).

Testes realizados

De modo a testar todas as restrições necessárias e disponibilizar apenas os serviços pedidos no enunciado, fizeram-se vários testes. Estes foram separados tendo em conta o seu propósito.

Para começar, apenas é permitido ao servidor receber *pings* da máquina *gcc* (10.101.151.5), para tal, criou-se uma regra para apenas aceitar *pings* vindos dessa máquina e foram rejeitados todos os pacotes que viessem de outros *IP's*, deste modo seria apresentada a mensagem de erro na consola do utilizador que fizesse o pedido de *ping*. Esta funcionalidade verificou-se pois, ao testar enviar *pings* da máquina *gcc*, os pedidos eram recebidos no computador A (utilizou-se o utilitário *snort* para verificar), mas, quando se fez *ping* do computador B para o computador A, aparecia a mensagem de porto impossível de contactar.

Relativamente ao segundo ponto do enunciado, o servidor alojado no computador A apenas pode aceitar conexões de clientes e ligações *ssh* vindas de computadores que estejam na mesma sub-rede do mesmo, enquanto são permitidas conexões com serviços cruciais ao funcionamento do servidor.

Para tal, pretendeu-se testar visando perceber se eram apenas aceites estas ligações, enquanto que todas as outras eram rejeitadas. Foi possível observar o que se queria pois, assim que se introduziam as regras já mencionadas acima na tabela de INPUT, o computador deixava de poder receber conexões no browser de internet, ou seja, essas mesmas ligações estavam a ser rejeitadas, ao passo que, quando o computador B se tentou ligar ao computador A, a ligação era aceite e estabelecida. O mesmo se sucedeu aquando uma ligação *ssh* era feita para o computador A vinda do computador B ou ainda de um computador C, também este na sub-rede local.

Por último, o servidor deverá ter alguns serviços mínimos, sendo estes a possibilidade de fazer apenas *pings* para máquinas da sub-rede local. Contudo a frequência dos *pings* deve ser limitada a um máximo de 2 *pings/segundo*.

Para tal, a partir do Computador A fez-se *ping* para o Computador B, utilizando o comando ***“ping -n -i 0.5 <IP do computador B>”***, deste modo, foram enviados *pings* a cada 0.5 segundos. Como era espectável não houve quaisquer problemas pois está dentro do limite. Para demonstrar que, ao serem enviados *pings* a uma taxa superior a 0.5 segundos, o servidor tem de lançar uma mensagem de erro utilizou-se o comando ***“ping -n -i 0.2 <IP do computador B>”***. Desta maneira são enviados *pings* a uma taxa de 5 *pings/segundo*, logo, a partir do momento em que se ultrapassa os 2 *pings/segundo*, é apresentada a mensagem de erro ***“ping: sendmsg: Operation not permitted”***. Este fenómeno pode ser observado na imagem a seguir, onde se vê que são bloqueados os *pings* que ultrapassam a taxa estipulada pela regra indicada na secção anterior.

```

IPTables - Google Drive - M... Desktop - File Manager Terminal
File Edit View Terminal Tabs Help
64 bytes from 10.101.149.64: icmp_seq=1 ttl=64 time=0.400 ms
64 bytes from 10.101.149.64: icmp_seq=2 ttl=64 time=0.377 ms
64 bytes from 10.101.149.64: icmp_seq=3 ttl=64 time=0.205 ms
64 bytes from 10.101.149.64: icmp_seq=4 ttl=64 time=0.188 ms
64 bytes from 10.101.149.64: icmp_seq=5 ttl=64 time=0.209 ms
64 bytes from 10.101.149.64: icmp_seq=6 ttl=64 time=0.424 ms
^C
--- 10.101.149.64 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 2497ms
rtt min/avg/max/mdev = 0.188/0.300/0.424/0.102 ms
fc50296@linux:~$ ping -n -i 0.2 10.101.149.64
PING 10.101.149.64 (10.101.149.64) 56(84) bytes of data.
64 bytes from 10.101.149.64: icmp_seq=1 ttl=64 time=0.212 ms
64 bytes from 10.101.149.64: icmp_seq=2 ttl=64 time=0.320 ms
ping: sendmsg: Operation not permitted
64 bytes from 10.101.149.64: icmp_seq=4 ttl=64 time=0.414 ms
ping: sendmsg: Operation not permitted
64 bytes from 10.101.149.64: icmp_seq=6 ttl=64 time=0.455 ms
ping: sendmsg: Operation not permitted
64 bytes from 10.101.149.64: icmp_seq=9 ttl=64 time=0.410 ms
ping: sendmsg: Operation not permitted
64 bytes from 10.101.149.64: icmp_seq=11 ttl=64 time=0.350 ms
ping: sendmsg: Operation not permitted
64 bytes from 10.101.149.64: icmp_seq=14 ttl=64 time=0.339 ms
ping: sendmsg: Operation not permitted
64 bytes from 10.101.149.64: icmp_seq=16 ttl=64 time=0.407 ms
ping: sendmsg: Operation not permitted
64 bytes from 10.101.149.64: icmp_seq=19 ttl=64 time=0.522 ms
ping: sendmsg: Operation not permitted
64 bytes from 10.101.149.64: icmp_seq=21 ttl=64 time=0.311 ms
ping: sendmsg: Operation not permitted
64 bytes from 10.101.149.64: icmp_seq=24 ttl=64 time=0.475 ms
ping: sendmsg: Operation not permitted
64 bytes from 10.101.149.64: icmp_seq=26 ttl=64 time=0.404 ms
ping: sendmsg: Operation not permitted
^C
--- 10.101.149.64 ping statistics ---
27 packets transmitted, 12 received, 55% packet loss, time 5297ms
rtt min/avg/max/mdev = 0.212/0.384/0.522/0.084 ms
fc50296@linux:~$ ^C

```

Figura 2 - Teste de pings enviados