



universidade
de aveiro

Segurança em Redes de Comunicações

Projeto 1

Nuno Ferreira (121758)
Miguel Ferreira (98345)

Docentes:

Prof. Paulo Salvador (salvador@ua.pt)

Prof. António Nogueira (nogueira@ua.pt)

Índice

Topologia da Rede	3
Política de segurança Implementada	4
Configuração das Rotas Estáticas e NAT:	5
Exercício 9.1	6
Exercício 9.2	7
Exercício 9.3	7
Exercício 10	8
Serviços	8
Demonstração dos Serviços a funcionar:	10
INSIDE para DMZ (Admin):	10
INSIDE para OUTSIDE:	13
OUTSIDE para DMZ :	14
OUTSIDE para DMZ (attackers) :	15
Blocking Rules Script	18
Anexos	20
Endereçamento IP:	20
LB1A	22
LB1B:	23
LB2A:	24
LB2B:	24
LBDMZ:	25
FW1:	26
FW2:	30

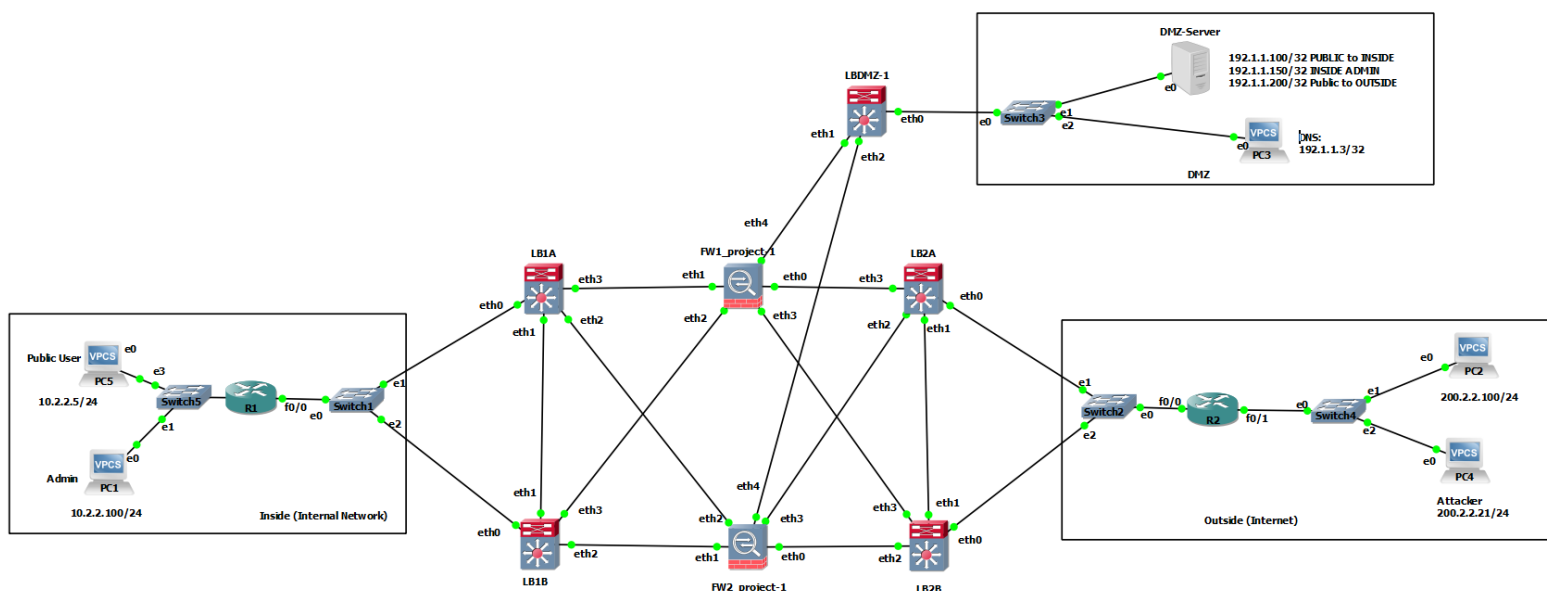
Topologia da Rede

Para o desenvolvimento do projeto, estruturámos a rede por três zonas, Inside, Outside e DMZ.

Na zona Inside (rede interna), existem duas sub-redes. A rede 10.2.2.0/24, que denominámos Rede Administrativa, contém um VPC com o endereço IP 10.2.2.100. Os restantes IPs da rede destinam-se aos restantes terminais da empresa.

A zona Outside (Internet) é constituída pela sub-rede 200.2.2.0/24 com um VPC para simular um dispositivo externo a que queremos aceder a partir de um terminal da empresa e outro VPC para simular um atacante.

Por último, a zona DMZ inclui a sub-rede 192.1.1.0/24 com três endereços IP. Os IPs 192.1.1.100 e 192.1.1.200 fazem parte do Servidor DMZ, o 192.1.1.200 simula um serviço web acessível interna e externamente, 192.1.1.100 simula outro serviço web acessível apenas internamente, o que pode representar um serviço web interno da empresa, como uma página administrativa. Adicionalmente, instanciámos um VPC com o endereço IP 192.1.1.150 para simular um servidor DNS.



Política de segurança Implementada

De uma forma rápida a nossa implementação foi a seguinte:

Na **DMZ**, todas as conexões de entrada para a rede interna (**INSIDE**) são permitidas. Isso inclui qualquer tipo de conexão, desde que já esteja estabelecida ou seja relacionada a uma conexão já existente. Além disso, todas as conexões da DMZ para o exterior (**OUTSIDE**) também são permitidas, garantindo flexibilidade para os serviços na DMZ se comunicarem com a internet.

No que diz respeito ao **INSIDE** é permitido ICMP's do tipo 8 para a **DMZ**, o que permite a realização de testes de ping e outras solicitações ICMP específicas, além disso, serviços como HTTP (porta 80), HTTPS (porta 443), SSH (porta 22) são permitidos para a **DMZ**, para o endereço IP 192.1.1.150/24, e isto apenas se aplica para o Admin, que é identificado pelo endereço IP 10.2.2.100/24z.

Já o DNS (porta 53), apenas é aceite no endereço IP 192.1.1.3/24. Dentro do **INSIDE** também tem um **utilizador público**, que tem acesso apenas aos serviços **HTTP**, **HTTPS**, ou seja, **não tem acesso ao SSH e DNS**, como o Admin tem. Quanto ao tráfego de saída, UDP para as portas 5000-6000 é permitido para o exterior. Conexões já estabelecidas ou relacionadas mais uma vez também são aceitas do exterior para a rede interna, garantindo que as respostas às solicitações originadas na rede interna sejam permitidas. E ainda permitimos o tráfego HTTP e HTTPS de usuários públicos na rede interna para a DMZ, sem ter acesso ao SSH como o Admin tem.

Para a zona **OUTSIDE** (Internet), as regras que implantamos permitem ICMP do tipo 8 e TCP na porta 443 para a DMZ. Essas permissões apenas são específicas para o endereço IP público 192.1.1.150/32, para uma maior segurança o **OUTSIDE** só consegue estabelecer conexão com 1 endereço IP. Para o tráfego de saída, qualquer conexão estabelecida ou relacionada é permitida do exterior tanto para a DMZ quanto para a rede interna. Ainda implementamos uma regra adicional para **bloquear** o tráfego originado dos endereços IP específicos entre 200.2.2.20 e 200.2.2.30 ao tentar **aceder** a **DMZ**, que vão ser simulados como endereços de IP de atacantes.

Configuração das Rotas Estáticas e NAT:

Começando com os Routers, definimos duas rotas estáticas básicas, e nos Load Balancers apenas definimos rotas estáticas para o exterior da rede, ou seja, ou para o Inside caso seja os LB1A e LB1B ou para o Outside caso seja no LB2A e LB2B, como se pode ver nos anexos.

Nas Firewalls, também definimos algumas rotas estáticas, tanto para o Inside como para o Outside, tudo o que for para os endereços 10.0.0.0/8, ou seja para o Inside vai ser redirecionado tanto pela interface eth1 ou eth2. Caso vá para o Outside, ou seja, para os endereços 0.0.0.0/0, vai pelas interfaces eth0 ou eth3. Tudo o que for destinado para a DMZ vai pela eth4.

Em relação ao NAT, implementámos regras de tradução de endereços de origem nas firewalls FW1 e FW2. Para a FW1, estabelecemos duas regras de NAT de origem. A primeira regra (Regra 10) traduz os endereços IP da rede interna (10.0.0.0/8) para um intervalo de endereços IP entre 192.1.0.1 e 192.1.0.10 quando o tráfego é roteado pela interface eth0. A segunda regra (Regra 20) realiza a mesma tradução, mas quando o tráfego é roteado pela interface eth3. Para o FW2 fizemos o mesmo, mas resolvemos o *range* dos endereços de IP para 192.1.0.11 e 192.1.0.20 para não haver conflito na tradução e para saber assim quando a capturar os pacotes, por onde essa tradução foi feita.

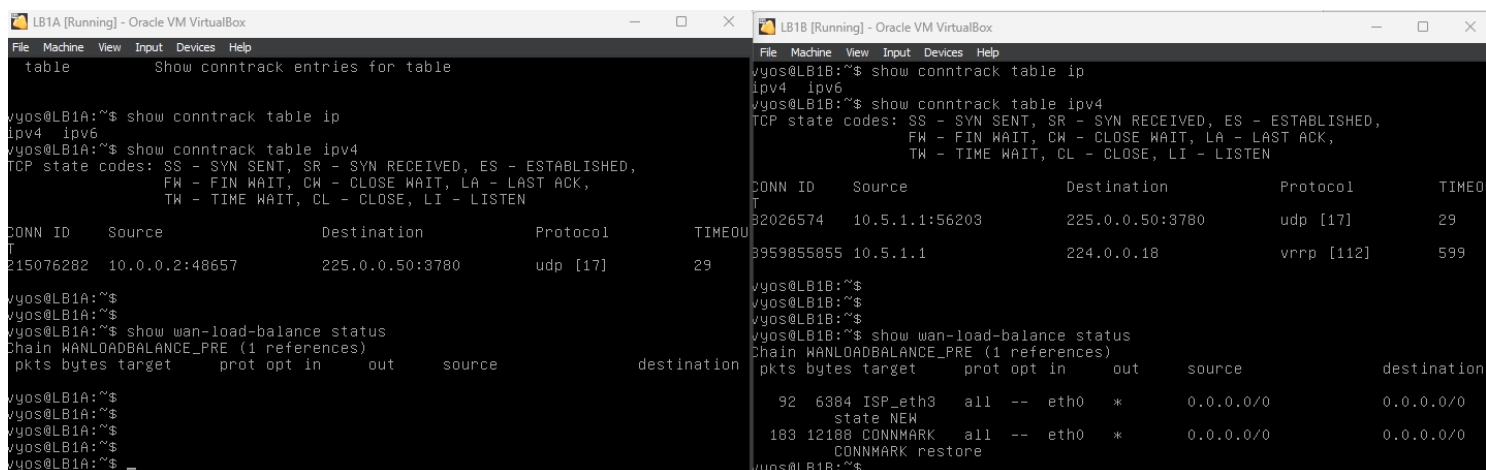
FW1:

```
vyos@FW1:~$ show nat source translations
Pre-NAT      Post-NAT      Prot  Timeout
10.5.3.1      10.5.3.1      icmp  29
10.3.3.1      10.3.3.1      icmp  23
10.5.3.1      10.5.3.1      icmp  23
10.6.1.1      10.6.1.1      icmp  26
vyos@FW1:~$
```

O mesmo no FW2.

Exercício 9.1

Através dos comandos *set load-balancing wan sticky-connections inbound* e *set load-balancing wan disable-source-nat* permite que não haja a necessidade de as Firewalls terem sincronização. Com o comando *set load-balancing wan sticky-connections* as interfaces vão memorizar de onde o tráfego foi recebido permitindo que o tráfego entre dispositivos seja sempre enviado pela mesma ligação e, com o comando *set load-balancing wan disable-source-nat*, o endereço ip de origem dos pacotes de saída será preservado quando forem enviados pela interface WAN de maneira a garantir que os pacotes sejam roteados corretamente de volta ao remetente. Por exemplo, se o request do ping do PC1 for por LB1A-FW1-LB2A até ao PC2, o mesmo, por causa dos comandos anteriores, vai saber por onde tem de enviar o reply para o PC1.



The image shows two terminal windows from Oracle VM VirtualBox, one for LB1A and one for LB1B, both running VyOS.

LB1A [Running] - Oracle VM VirtualBox

```
vyos@LB1A:~$ show conntrack table ip
table
Show conntrack entries for table ip
vyos@LB1A:~$ show conntrack table ipv4
ipv4 ipv6
vyos@LB1A:~$ show conntrack table ipv4
TCP state codes: SS - SYN SENT, SR - SYN RECEIVED, ES - ESTABLISHED,
FW - FIN WAIT, CW - CLOSE WAIT, LA - LAST ACK,
TW - TIME WAIT, CL - CLOSE, LI - LISTEN
CONN ID      Source      Destination      Protocol      TIMEO
215076282    10.0.0.2:48657  225.0.0.50:3780  udp [17]      29
vyos@LB1A:~$
vyos@LB1A:~$
vyos@LB1A:~$ show wan-load-balance status
Chain WANLOADBALANCE_PRE (1 references)
pkts bytes target  prot opt in  out  source      destination
vyos@LB1A:~$
vyos@LB1A:~$
vyos@LB1A:~$
vyos@LB1A:~$
```

LB1B [Running] - Oracle VM VirtualBox

```
vyos@LB1B:~$ show conntrack table ip
ipv4 ipv6
vyos@LB1B:~$ show conntrack table ipv4
TCP state codes: SS - SYN SENT, SR - SYN RECEIVED, ES - ESTABLISHED,
FW - FIN WAIT, CW - CLOSE WAIT, LA - LAST ACK,
TW - TIME WAIT, CL - CLOSE, LI - LISTEN
CONN ID      Source      Destination      Protocol      TIMEO
32026574     10.5.1.1:56203  225.0.0.50:3780  udp [17]      29
3959855855   10.5.1.1      224.0.0.18       vrrp [112]    599
vyos@LB1B:~$
vyos@LB1B:~$
vyos@LB1B:~$
vyos@LB1B:~$ show wan-load-balance status
Chain WANLOADBALANCE_PRE (1 references)
pkts bytes target  prot opt in  out  source      destination
92  6384  ISP_eth3  all  --  eth0  *    0.0.0.0/0   0.0.0.0/0
state NEW
183 12188 CONNMARK  all  --  eth0  *    0.0.0.0/0   0.0.0.0/0
CONNMARK restore
vyos@LB1B:~$
```

Exercício 9.2

O algoritmo IP Hash pode permitir a inexistência de sincronização nos Load Balancers porque, neste algoritmo existe uma hash function que é usada para mapear cada request para um servidor específico. A hash function assegura que o mesmo request é enviado para o mesmo servidor, assim, cada Load Balancer, através do endereço ip do cliente, pode calcular, independentemente, o servidor/firewall para qual o request dever ser enviado, eliminando a necessidade de sincronização.

Exercício 9.3

Os dispositivos que tenham a sincronização ativa estão constantemente a trocar informação para se atualizarem. Se houver um ataque DDOS nesses dispositivos de sincronização, a rede pode tornar-se muito mais lenta, pelo número elevado de pacotes a serem encaminhados nas interfaces de sincronização, por exemplo, se houver um ataque DDOS no LB1A, como o LB1A e o LB1B estão sincronizados, esse ataque ia se propagar para o LB1B, ou seja, ia afetar ainda mais a rede levando a um efeito em cascada. Posto isto, a sincronização pode ser prejudicial durante um ataque de DDOS.

Exercício 10

Serviços

Os primeiros serviço que consideramos foi o acesso do **INSIDE** para a **DMZ**, onde todos os dispositivos internos (que sejam admins, no nosso caso apenas o source address 10.2.2.100), podem fazer ping com o ICMP aos dispositivos na rede DMZ comunicar-se via UDP na porta **53** para uma comunicação **DNS** e ainda podem comunicar-se via TCP nas portas **80** e **443**, permitindo-lhes aceder serviços web através do **HTTP** e **HTTPS** para uma comunicação mais segura, e ainda (apenas a rede de administração 10.2.2.100/24) pode se comunicar com a DMZ via TCP na porta **22**, significando que apenas administradores podem estabelecer conexões **SSH** aos terminais dentro da DMZ. O resto dos utilizadores, que não sejam Admin, apenas podem comunicar vai TCP por **HTTP** e **HTTPS**.

Rule 10:

Protocols: ICMP

Services: ping

Destination address: 192.1.1.100/24

Source address: 10.2.2.100/24 (Admin)

Rule 20:

Protocols: TCP

Services: HTTP, HTTPS, SSH

Ports: 80, 443, 22

Destination address: 192.1.1. 100/24

Source address: 10.2.2.100/24 (Admin)

Rule 30:

Protocols: UDP

Services: DNS

Ports: 53

Destination address: 192.1.1.3/24. (VPC3)

Source address: 10.2.2.100/24 (Admin)

Rule 40:

Protocols: TCP

Services: HTTP, HTTPS

Ports: 80, 443

Destination address: 192.1.1.150/32

Demonstração dos Serviços a funcionar:

INSIDE para DMZ (Admin):

Rule 10

- PC1 a pingar 192.1.1.150 com ICMP:

```
PC1> ping 192.1.1.150
84 bytes from 192.1.1.150 icmp_seq=1 ttl=60 time=20.686 ms
84 bytes from 192.1.1.150 icmp_seq=2 ttl=60 time=14.008 ms
84 bytes from 192.1.1.150 icmp_seq=3 ttl=60 time=13.918 ms
84 bytes from 192.1.1.150 icmp_seq=4 ttl=60 time=14.882 ms
```

242	333.073444	10.2.2.100	192.1.1.150	ICMP	98 Echo (ping) request	id=0xea3b, seq=2/512, ttl=61 (reply in 243)
243	333.073974	192.1.1.150	10.2.2.100	ICMP	98 Echo (ping) reply	id=0xea3b, seq=2/512, ttl=63 (request in 242)
244	333.895597	10.6.1.1	10.6.1.2	ICMP	74 Echo (ping) request	id=0x065c, seq=256/1, ttl=64 (reply in 245)
245	333.895874	10.6.1.2	10.6.1.1	ICMP	74 Echo (ping) reply	id=0x065c, seq=256/1, ttl=64 (request in 244)

Rule 20

- PC1 a pingar 192.1.1.150 com TCP pela porta 80, HTTP :

```
PC1> ping 192.1.1.150 -P 6 -p 80
Connect 80@192.1.1.150 seq=1 ttl=60 time=15.096 ms
SendData 80@192.1.1.150 seq=1 ttl=60 time=20.915 ms
Close 80@192.1.1.150 seq=1 ttl=60 time=23.383 ms
Connect 80@192.1.1.150 seq=2 ttl=60 time=14.164 ms
SendData 80@192.1.1.150 seq=2 ttl=60 time=21.082 ms
Close 80@192.1.1.150 seq=2 ttl=60 time=13.333 ms
```

tcp.port == 80							
No.	Time	Source	Destination	Protocol	Length	Info	DSCP Val
16	3.989605	192.1.1.150	10.2.2.100	TCP	74	80 → 38126 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 TSval=437343631 ...	
17	4.009153	10.2.2.100	192.1.1.150	TCP	66	38126 → 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSval=1713715873 TSecr=0	
18	4.009261	10.2.2.100	192.1.1.150	TCP	122	38126 → 80 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=56 TSval=1713715873 TSecr=0	
19	4.010272	192.1.1.150	10.2.2.100	TCP	66	80 → 38126 [ACK] Seq=1 Ack=57 Win=65152 Len=0 TSval=437343652 TSecr=1713715...	
20	4.010380	192.1.1.150	10.2.2.100	HTTP	558	HTTP/1.1 400 Bad Request (text/html)	
21	4.010409	192.1.1.150	10.2.2.100	TCP	66	80 → 38126 [FIN, ACK] Seq=493 Ack=57 Win=65152 Len=0 TSval=437343652 TSecr=...	
22	4.030288	10.2.2.100	192.1.1.150	TCP	66	[TCP ACKed unseen segment] 38126 → 80 [ACK] Seq=57 Ack=545 Win=5840 Len=0 T...	
23	4.031011	192.1.1.150	10.2.2.100	TCP	66	[TCP Dup ACK 19#1] 80 → 38126 [ACK] Seq=494 Ack=57 Win=65152 Len=0 TSval=43...	

Rule 20

- PC1 a pingar 192.1.1.150 com TCP pela porta 443, HTTPS, (“RST returned” porque não tínhamos aberto a porta 443 no servidor) :

```
PC1> ping 192.1.1.150 -P 6 -p 443
Connect 443@192.1.1.150 RST returned
Connect 443@192.1.1.150 RST returned
Connect 443@192.1.1.150 RST returned
Connect 443@192.1.1.150 RST returned
Connect 443@192.1.1.150 timeout
```

10.2.2.100	192.1.1.150	TCP	74 [TCP Port numbers reused] 64040 → 443 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 T...
192.1.1.150	10.2.2.100	TCP	54 443 → 64040 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
10.2.2.100	192.1.1.150	TCP	74 [TCP Port numbers reused] 64040 → 443 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 T...
192.1.1.150	10.2.2.100	TCP	54 443 → 64040 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
10.2.2.100	192.1.1.150	TCP	74 [TCP Port numbers reused] 64040 → 443 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 T...
192.1.1.150	10.2.2.100	TCP	54 443 → 64040 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Rule 20

- PC1 a pingar 192.1.1.150 com TCP pela porta 22 SSH:

```
PC1> ping 192.1.1.150 -P 6 -p 22
Connect 22@192.1.1.150 seq=1 ttl=60 time=17.778 ms
SendData 22@192.1.1.150 seq=1 ttl=60 time=20.569 ms
Close 22@192.1.1.150 timeout
Connect 22@192.1.1.150 seq=2 ttl=60 time=12.782 ms
SendData 22@192.1.1.150 seq=2 ttl=60 time=13.839 ms
Close 22@192.1.1.150 timeout
Connect 22@192.1.1.150 seq=3 ttl=60 time=17.264 ms
SendData 22@192.1.1.150 seq=3 ttl=60 time=21.128 ms
Close 22@192.1.1.150 timeout
Connect 22@192.1.1.150 seq=4 ttl=60 time=19.245 ms
SendData 22@192.1.1.150 seq=4 ttl=60 time=20.335 ms
Close 22@192.1.1.150 timeout
Connect 22@192.1.1.150 seq=5 ttl=60 time=16.927 ms
SendData 22@192.1.1.150 seq=5 ttl=60 time=21.192 ms
Close 22@192.1.1.150 timeout
```

202 248.515750	192.1.1.150	10.2.2.100	TCP	74 22 → 32793 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 TSval=437588157 ...
203 248.536128	10.2.2.100	192.1.1.150	TCP	66 32793 → 22 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSval=1713716117 TSecr=0
204 248.541092	192.1.1.150	10.2.2.100	SSH	107 Server: Protocol (SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2)
205 248.546883	10.2.2.100	192.1.1.150	SSH	122 Client: Encrypted packet (len=56)
206 248.547337	192.1.1.150	10.2.2.100	TCP	66 22 → 32793 [ACK] Seq=42 Ack=57 Win=65280 Len=0 TSval=437588189 TSecr=171371...

Rule 30

- **PC1 a pingar 192.1.1.3 com UDP pela porta 53 DNS:**

Dá um “network unreachable “ mas deveria estar a funcionar e não encontramos a solução. Deixamos em baixo a rule 30 para verificar que estava tudo correto.

```
PC1> ping 192.1.1.3 -P 17 -p 53
*10.1.1.1 udp_seq=1 ttl=63 time=19.270 ms (ICMP type:3, code:0, Destination network unreachable)
*10.1.1.1 udp_seq=2 ttl=63 time=18.072 ms (ICMP type:3, code:0, Destination network unreachable)
*10.1.1.1 udp_seq=3 ttl=63 time=21.347 ms (ICMP type:3, code:0, Destination network unreachable)
*10.1.1.1 udp_seq=4 ttl=63 time=18.948 ms (ICMP type:3, code:0, Destination network unreachable)
192.1.1.3 udp_seq=5 timeout
```

```
rule 30 {
    action accept
    description "Accept DNS to DMZ"
    destination {
        address 192.1.1.3/32
        port 53
    }
    protocol udp
    source {
        address 10.2.2.100/24
    }
}
```

INSIDE para OUTSIDE:

Do INSIDE para o OUTSIDE definimos que apenas o INSIDE consegue iniciar uma comunicação via UDP nas portas 5000-6000, onde o OUTSIDE não consegue iniciar a comunicação.

```
PC1> ping 200.2.2.100 -P 17 -p 5001
84 bytes from 200.2.2.100 udp_seq=1 ttl=59 time=43.061 ms
84 bytes from 200.2.2.100 udp_seq=2 ttl=59 time=34.252 ms
84 bytes from 200.2.2.100 udp_seq=3 ttl=59 time=35.236 ms

PC1> ping 200.2.2.100 -P 17 -p 6001
200.2.2.100 udp_seq=1 timeout
200.2.2.100 udp_seq=2 timeout
200.2.2.100 udp_seq=3 timeout
```

Como conseguimos ver na imagem, o PC1 (Internal network) consegue comunicar com um endereço de ip do OUTSIDE (Internet) 200.2.2.100 usando o protocolo UDP (-P17) na porta 5001 (-p 5001). Se agora tentarmos comunicar pela porta 6001 que esta fora do range, como esperado já dá um timeout. No caso do PC2 (Outside) tentar pingar o PC1 do Inside, é de esperar que não consiga estabelecer uma comunicação.

```
PC2> ping 10.2.2.100 -P 17 -p 5005
10.2.2.100 udp_seq=1 timeout
10.2.2.100 udp_seq=2 timeout
10.2.2.100 udp_seq=3 timeout
10.2.2.100 udp_seq=4 timeout
10.2.2.100 udp_seq=5 timeout
```

OUTSIDE para DMZ :

Para o OUTSIDE com o DMZ, a política de segurança que implementamos foi que o OUTSIDE apenas pode comunicar com o DMZ pelo endereço de IP 192.1.1.200/32 e apenas comunicações **tcp's** pela porta **443** e **ICMP**, apenas para ser um serviço do tipo HTTPS onde é uma ligação mais segura possível, e ICMP para testar a conexão primeiramente.

```
PC2> ping 192.1.1.200
84 bytes from 192.1.1.200 icmp_seq=1 ttl=60 time=20.547 ms
84 bytes from 192.1.1.200 icmp_seq=2 ttl=60 time=19.175 ms
84 bytes from 192.1.1.200 icmp_seq=3 ttl=60 time=15.883 ms
84 bytes from 192.1.1.200 icmp_seq=4 ttl=60 time=11.702 ms
84 bytes from 192.1.1.200 icmp_seq=5 ttl=60 time=21.485 ms

PC2> ping 192.1.1.100
192.1.1.100 icmp_seq=1 timeout
192.1.1.100 icmp_seq=2 timeout
192.1.1.100 icmp_seq=3 timeout

PC2> ping 192.1.1.200 -P 6 -p 443
Connect 443@192.1.1.200 RST returned
Connect 443@192.1.1.200 RST returned
Connect 443@192.1.1.200 RST returned
Connect 443@192.1.1.200 RST returned
Connect 443@192.1.1.200 RST returned
```

Com os pings conseguimos ver que de facto ele consegue pingar o endereço 192.1.1.200 para testar a conexão. Depois no segundo ping verificamos que ele não consegue comunicar com outro endereço se não o imposto pelas regras da firewall. E no terceiro verificamos que ele consegue inicializar uma comunicação TCP pela porta 443, simulando uma conexão por HTTPS, neste caso a mais segura.

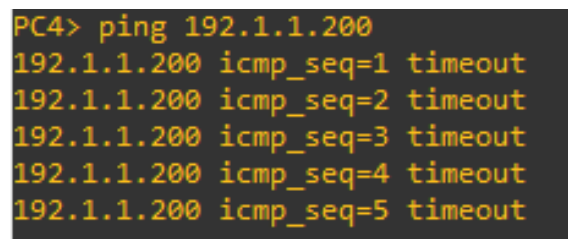
OUTSIDE para DMZ (attackers) :

Por fim, foram adicionadas configurações para dar drop/bloquear os pacotes com base no endereço de ip de origem(ip do atacante). De salientar que, de maneira a verificar sempre em primeiro lugar se o ip de origem é um ip identificado como sendo um atacante(dentro do range **200.2.2.20 – 200.2.2.30**), definimos um menor sequence number para esta regra (**rule 5**).

Na figura abaixo estão presentes as configurações onde mostramos essa regra:

```
set firewall name FROM-OUTSIDE-TO-DMZ rule 5 action drop
set firewall group address-group attackers address '200.2.2.20-200.2.2.30'
set firewall name FROM-OUTSIDE-TO-DMZ rule 5 source group address-group attackers
```

Como era previsto, o ping do atacante para a DMZ foi bloqueado pela firewall, porque o ip do atacante é 200.2.2.22 e como definimos como regra na firewall que os ip's de origem no range 200.2.2.20 – 200.2.2.30 iam ser bloqueados, então o ip do atacante foi de facto bloqueado.



```
PC4> ping 192.1.1.200
192.1.1.200 icmp_seq=1 timeout
192.1.1.200 icmp_seq=2 timeout
192.1.1.200 icmp_seq=3 timeout
192.1.1.200 icmp_seq=4 timeout
192.1.1.200 icmp_seq=5 timeout
```

Nas capturas do wireshark, dá para ver que nenhum pacote passou pela firewall:

No.	Time	Source	Destination	Protocol	Length	Info
39	88.679309	10.5.3.1	10.5.3.2	ICMP	74	Echo (ping) request id=0x06d6, seq=256/1, ttl=64 (reply in 40)
40	88.679521	10.5.3.2	10.5.3.1	ICMP	74	Echo (ping) reply id=0x06d6, seq=256/1, ttl=64 (request in 39)
41	94.746189	10.0.3.2	10.5.3.1	ICMP	74	Echo (ping) reply id=0x072c, seq=256/1, ttl=64
42	99.768054	10.5.3.1	10.5.3.2	ICMP	74	Echo (ping) request id=0x06d6, seq=256/1, ttl=64 (reply in 43)
43	99.768355	10.5.3.2	10.5.3.1	ICMP	74	Echo (ping) reply id=0x06d6, seq=256/1, ttl=64 (request in 42)
44	105.866128	10.0.3.2	10.5.3.1	ICMP	74	Echo (ping) reply id=0x072c, seq=256/1, ttl=64
45	110.854892	10.5.3.1	10.5.3.2	ICMP	74	Echo (ping) request id=0x06d6, seq=256/1, ttl=64 (reply in 46)
46	110.855117	10.5.3.2	10.5.3.1	ICMP	74	Echo (ping) reply id=0x06d6, seq=256/1, ttl=64 (request in 45)
47	111.079042	PcsCompu_22:4f:00	PcsCompu_56:35:f1	ARP	42	Who has 10.5.3.1? Tell 10.5.3.2
48	111.079303	PcsCompu_56:35:f1	PcsCompu_22:4f:00	ARP	42	10.5.3.1 is at 08:00:27:56:35:f1

Políticas e regras

Definimos 3 Zonas, INSIDE, OUTSIDE e DMZ:

Na zona INSIDE, definimos 2 *policies* com apenas 1 regra básica cada:

1- FROM-DMZ-TO-INSIDE:

Rule 10 - Permite que o tráfego que seja parte de uma conexão já estabelecida ou que seja relacionado a uma conexão estabelecida posteriormente, seja aceite pela firewall.

2- FROM-OUTSIDE-TO-INSIDE:

Rule 10 - Permite que o tráfego que seja parte de uma conexão já estabelecida ou que seja relacionado a uma conexão estabelecida posteriormente, seja aceite pela firewall.

Na zona OUTSIDE, definimos 3 *policies*:

1- FROM-INSIDE-TO-OUTSIDE:

Rule 10 - Permite que o tráfego que seja parte de uma conexão já estabelecida ou que seja relacionado a uma conexão estabelecida posteriormente, seja aceito pela firewall e ainda que os pacotes **UDP** com destino às portas entre **5000** e **6000** sejam **aceites** pelo firewall

2- FROM-DMZ-TO-OUTSIDE:

Rule 10 - Permite que o tráfego que seja parte de uma conexão já estabelecida ou que seja relacionado a uma conexão estabelecida posteriormente, seja aceite pela firewall.

Na zona DMZ, definimos 2 *policies*:

1- FROM-INSIDE-TO-DMZ:

Rule 10 - Permite pacotes ICMP do tipo Echo Request (tipo 8) sejam aceites pela firewall quando destinados ao DMZ (192.1.1.150/32).

Rule 20 - Permite que o tráfego **TCP** da rede administrativa (10.2.2.100/24) e com destino à Zona DMZ (192.1.1.150/32) seja aceite para os serviços **SSH**, **HTTP** e **HTTPS** nas portas 22, 80 e 443, respetivamente. Apenas quando o source address é igual a 10.2.2.100/24, pois este é o Admin.

Rule 30 - Permite que o tráfego **UDP** com destino à porta 53 (**DNS**) e destinado ao endereço IP **192.1.1.3**, seja aceite pelo firewall.

Rule 40- Igual á Rule 20, mas apenas destinado ao HTTP e HTTPS, pois é destinada a usuários públicos.

2- FROM-OUTSIDE-TO-DMZ:

Rule 10- Permite que o tráfego que seja parte de uma conexão já estabelecida ou que seja relacionado.

Blocking Rules Script

Este script tem como objetivo criar automaticamente as regras de bloqueio da firewall após a identificação do endereço de ip dos atacantes.

Foi definido um range de endereços para simular a lista com ip's dos atacantes. Em seguida, através do comando da linha do **tcpdump**, capturamos os pacotes na interface **enp0s3** (por exemplo), filtrando os pacotes que têm o endereço ip no range definido pelas variáveis **\$ip_start** e **\$ip_end** e parar após capturar 1000 pacotes. Ainda na mesma linha, o ***while read attacker_ip*** vai iterar sobre cada linha e vai atribuir o valor à variável **attacker_ip**. Posto isto, se pacotes forem capturados nesta interface através do comando **tcpdump** significa que houve pacotes cujo ip origem estava na lista dos endereços ip's dos atacantes e iniciamos uma conexão remota via **ssh** para as duas firewalls para bloquear pacotes com base no endereço ip de origem(**attacker_ip**).

Infelizmente tentámos testar este script, mas não conseguimos chegar ao resultado pretendido, mas pelo menos fica uma ideia daquilo que pretendíamos fazer no script.

```

#!/bin/bash

# Set the firewall IP addresses
firewall1_ip_address="firewall1_ip_address"
firewall2_ip_address="firewall2_ip_address"

# Set the IP range to block
ip_start="200.2.2.20"
ip_end="200.2.2.30"

# SSH password
password="vyos"

echo "Searching for DDOS attacks"

# Start capturing network traffic with tcpdump and filter it based on
source IP address
tcpdump -n -i enp0s3 src net $ip_start/$ip_end -c 1000 | grep -oE "\b([0-
9]{1,3}\.){3}[0-9]{1,3}\b" | sort -u | while read attacker_ip;
do
    # Connect to the firewall1 via SSH using sshpass
    sshpass -p $password ssh -o StrictHostKeyChecking=no
vyos@$firewall1_ip_address << EOF
    configure
    set firewall name BLOCK rule 10 action drop
    set firewall name BLOCK rule 10 source address $attacker_ip
    commit
    save
    exit
EOF

    # Connect to the firewall2 via SSH using sshpass
    sshpass -p $password ssh -o StrictHostKeyChecking=no
vyos@$firewall2_ip_address << EOF
    configure
    set firewall name BLOCK rule 10 action drop
    set firewall name BLOCK rule 10 source address $attacker_ip
    commit
    save
    exit
EOF

# Print a message to confirm that the blocking rule has been added
echo "Blocking rule added to the firewalls to block traffic from IP
address $attacker_ip"

done

```

Anexos

Endereçamento IP:

RouterInside

- PC1: 10.2.2.100/24
- PC5: 10.2.2.5/24
- f0/1: 10.2.2.10/24
- f0/0: 10.1.1.10/24

RouterOutside

- PC2: 200.2.2.100/24
- PC4: 200.2.2.21/24
- f0/1: 200.2.2.10/24
- f0/0: 200.1.1.10/24

DMZ

- Servidor Inside publico: 192.1.1.100/24
- Servidor privado:192.1.1.150/24
- Servidor Outisde Publico: 192.1.1.200/24
- Servidor DNS: 192.1.1.3/24

LB1A

- eth0: 10.1.1.1/24
- eth1: 10.0.0.1/24
- eth2: 10.0.2.1/24
- eth3: 10.0.3.1/24

FW1

- eth0: 10.5.3.2/24
- eth1: 10.0.3.2/24
- eth2: 10.3.3.2/24
- eth3: 10.4.3.2/24
- eth4: 10.6.1.2/24

LB2A

- eth0: 200.1.1.1/24
- eth1: 10.5.1.1/24
- eth2: 10.5.2.1/24

- **eth3:** 10.5.3.1/24

FW2

- **eth0:** 10.4.2.2/24
- **eth1:** 10.3.2.2/24
- **eth2:** 10.0.2.2/24
- **eth3:** 10.5.2.2/24
- **eth4:** 10.6.2.2/24

LB1B

- **eth0:** 10.1.1.2/24
- **eth1:** 10.0.0.2/24
- **eth2:** 10.3.2.1/24
- **eth3:** 10.3.3.1/24

LB2B

- **eth0:** 200.1.1.2/24
- **eth1:** 10.5.1.2/24
- **eth2:** 10.4.2.1/24
- **eth3:** 10.4.3.1/24

LBDMZ

- **eth0:** 192.1.1.1/24
- **eth1:** 10.6.1.1/24
- **eth2:** 10.6.2.1/24

Router 1 Inside:

conf term

ip route 0.0.0.0 0.0.0.0 10.1.1.1 (tudo o que for para fora enviar pelas duas interfaces)

ip route 0.0.0.0 0.0.0.0 10.1.1.2

Router 2 Outside:

conf term

ip route 10.0.0.0 255.0.0.0 200.1.1.1 (tudo o que for para a 10.0.0.0 envia pelo LB2A)

ip route 192.1.0.0 255.255.254.0 200.1.1.1 (tudo o que for para o DMZ envia pelo LB2A)

LB1A:

set protocols static route 10.2.2.0/24 next-hop 10.1.1.10 (Tudo o for para o 10.2.2.0/24 vai pelo 10.1.1.10)

```
set high-availability vrrp group LB1AB_Cluster interface eth1
set high-availability vrrp group LB1AB_Cluster rfc3768-compatibility
set high-availability vrrp group LB1AB_Cluster virtual-address 10.0.0.1/24
set high-availability vrrp group LB1AB_Cluster vrid 1
set high-availability vrrp sync-group LB1AB_Cluster member LB1AB_Cluster
```

```
set service conntrack-sync accept-protocol tcp,udp,icmp
set service conntrack-sync failover-mechanism vrrp sync-group LB1AB_Cluster
set service conntrack-sync interface eth1
set service conntrack-sync mcast-group 225.0.0.50 (Que valor meto aqui??)
set service conntrack-sync disable-external-cache
```

```
set load-balancing wan interface-health eth2 nexthop 10.0.2.2
set load-balancing wan interface-health eth3 nexthop 10.0.3.2
set load-balancing wan rule 1 inbound-interface eth0
set load-balancing wan rule 1 interface eth2 weight 1
set load-balancing wan rule 1 interface eth3 weight 1
set load-balancing wan rule 1 protocol all
set load-balancing wan sticky-connections inbound
set load-balancing wan disable-source-nat
```

LB1B:

set protocols static route 10.2.2.0/24 next-hop 10.1.1.10 (Tudo o for para o 10.2.2.0/24 vai pelo 10.1.1.10)

```
set high-availability vrrp group LB1AB_Cluster interface eth1
set high-availability vrrp group LB1AB_Cluster rfc3768-compatibility
set high-availability vrrp group LB1AB_Cluster virtual-address 10.0.0.1/24
set high-availability vrrp group LB1AB_Cluster vrid 1
set high-availability vrrp sync-group LB1AB_Cluster member LB1AB_Cluster
```

```
set service conntrack-sync accept-protocol tcp,udp,icmp
set service conntrack-sync failover-mechanism vrrp sync-group LB1AB_Cluster
set service conntrack-sync interface eth1
set service conntrack-sync mcast-group 225.0.0.50 (Que valor meto aqui??)
set service conntrack-sync disable-external-cache
```

```
set load-balancing wan interface-health eth2 nexthop 10.3.2.2
set load-balancing wan interface-health eth3 nexthop 10.3.3.2
set load-balancing wan rule 1 inbound-interface eth0
set load-balancing wan rule 1 interface eth2 weight 1
set load-balancing wan rule 1 interface eth3 weight 1
set load-balancing wan rule 1 protocol all
set load-balancing wan sticky-connections inbound
set load-balancing wan disable-source-nat
```

LB2A:

```
set protocols static route 200.2.2.0/24 next-hop 200.1.1.10
```

```
set high-availability vrrp group LB2AB_Cluster interface eth1
```

```
set high-availability vrrp group LB2AB_Cluster rfc3768-compatibility
```

```
set high-availability vrrp group LB2AB_Cluster virtual-address 192.168.100.1/24 (Ip correto  
mas podia ser o 192.168.100.1/24 ?)
```

```
set high-availability vrrp group LB2AB_Cluster vrid 2
```

```
set high-availability vrrp sync-group LB2AB_Cluster member LB2AB_Cluster
```

```
set load-balancing wan interface-health eth2 nexthop 10.5.2.2
```

```
set load-balancing wan interface-health eth3 nexthop 10.5.3.2
```

```
set load-balancing wan rule 1 inbound-interface eth0
```

```
set load-balancing wan rule 1 interface eth2 weight 1
```

```
set load-balancing wan rule 1 interface eth3 weight 1
```

```
set load-balancing wan rule 1 protocol all
```

```
set load-balancing wan sticky-connections inbound
```

```
set load-balancing wan disable-source-nat
```

```
set service conntrack-sync accept-protocol tcp,udp,icmp
```

```
set service conntrack-sync failover-mechanism vrrp sync-group LB2AB_Cluster
```

```
set service conntrack-sync interface eth1
```

```
set service conntrack-sync mcast-group 225.0.0.50 (Que valor meto aqui??)
```

```
set service conntrack-sync disable-external-cache
```

LB2B:

```
set protocols static route 200.2.2.0/24 next-hop 200.1.1.10
```

```
set high-availability vrrp group LB2AB_Cluster interface eth1
```

```
set high-availability vrrp group LB2AB_Cluster rfc3768-compatibility
```

set high-availability vrrp group LB2AB_Cluster virtual-address 192.168.100.1/24 (Ip correto mas podia ser o 192.168.100.1/24 ?)

set high-availability vrrp group LB2AB_Cluster vrid 2

set high-availability vrrp sync-group LB2AB_Cluster member LB2AB_Cluster

set load-balancing wan interface-health eth2 nexthop 10.4.2.2

set load-balancing wan interface-health eth3 nexthop 10.4.3.2

set load-balancing wan rule 1 inbound-interface eth0

set load-balancing wan rule 1 interface eth2 weight 1

set load-balancing wan rule 1 interface eth3 weight 1

set load-balancing wan rule 1 protocol all

set load-balancing wan sticky-connections inbound

set load-balancing wan disable-source-nat

set service conntrack-sync accept-protocol tcp,udp,icmp

set service conntrack-sync failover-mechanism vrrp sync-group LB2AB_Cluster

set service conntrack-sync interface eth1

set service conntrack-sync mcast-group 225.0.0.50 (Que valor meto aqui??)

set service conntrack-sync disable-external-cache

LBDMZ:

set protocols static route 192.1.1.0/24 next-hop 192.1.1.1

set load-balancing wan interface-health eth1 nexthop 10.6.1.2

set load-balancing wan interface-health eth2 nexthop 10.6.2.2

set load-balancing wan rule 1 inbound-interface eth0

set load-balancing wan rule 1 interface eth1 weight 1

set load-balancing wan rule 1 interface eth2 weight 1

set load-balancing wan rule 1 protocol all

set load-balancing wan sticky-connections inbound

set load-balancing wan disable-source-nat

FW1:

""Aplicar NAT:

Regra de NAT número 10: Aplicada ao tráfego que sai pela interface eth0 com endereços IP originais da faixa 10.0.0.0/8, que serão traduzidos para a faixa 192.1.0.1-192.1.0.10.

Regra de NAT número 20: Aplicada ao tráfego que sai pela interface eth3 com endereços IP originais da faixa 10.0.0.0/8, que também serão traduzidos para a faixa 192.1.0.1-192.1.0.10.

Ambas as regras estão traduzindo os endereços IP da faixa 10.0.0.0/8 para a faixa 192.1.0.1-192.1.0.10 quando o tráfego sai pelas interfaces eth0 e eth3.""

```
set nat source rule 10 outbound-interface eth0
```

```
set nat source rule 10 source address 10.0.0.0/8
```

```
set nat source rule 10 translation address 192.1.0.1-192.1.0.10          RULE 10 E
PELA eth0!
```

```
set nat source rule 20 outbound-interface eth3
```

```
set nat source rule 20 source address 10.0.0.0/8
```

```
set nat source rule 20 translation address 192.1.0.1-192.1.0.10        RULE 20 E
PELA eth3!
```

```
set protocols static route 0.0.0.0/0 next-hop 10.4.3.1 (Tudo o que for para 0.0.0.0 vai pelo
10.5.3.2 , ou seja tudo o que for para o exterior) eth0
```

```
set protocols static route 0.0.0.0/0 next-hop 10.5.3.1 "" "" eth3
```

```
set protocols static route 10.0.0.0/8 next-hop 10.0.3.1 (Tudo o que for para 10.0.0.0/8 vai pelo
10....., ou seja tudo o que for para o interior) eth1
```

```
set protocols static route 10.0.0.0/8 next-hop 10.3.3.1 "" "" eth2
```

```
set protocols static route 192.1.1.0/24 next-hop 10.6.1.1
```

FIREWALL RULES:

Connection entre DMZ para INSIDE aceitar "todas":

```
set firewall name FROM-DMZ-TO-INSIDE rule 10 action accept
```

```
set firewall name FROM-DMZ-TO-INSIDE rule 10 description "Accept connection DMZ-
INSIDE"
```

set firewall name FROM-DMZ-TO-INSIDE rule 10 state established enable

set firewall name FROM-DMZ-TO-INSIDE rule 10 state related enable

Connection entre INSIDE para o servicos de DMZ:

set firewall name FROM-INSIDE-TO-DMZ rule 10 action accept

set firewall name FROM-INSIDE-TO-DMZ rule 10 description "Accept ICMP only to DMZ"

set firewall name FROM-INSIDE-TO-DMZ rule 10 destination address 192.1.1.150/32

set firewall name FROM-INSIDE-TO-DMZ rule 10 icmp type 8

set firewall name FROM-INSIDE-TO-DMZ rule 10 protocol icmp

set firewall name FROM-INSIDE-TO-DMZ rule 10 source address 10.2.2.100/24 (ADMIN)

set firewall name FROM-INSIDE-TO-DMZ rule 20 action accept

set firewall name FROM-INSIDE-TO-DMZ rule 20 description "Accept multiple services for admin use: HTTP, HTTPS, SSH"

set firewall name FROM-INSIDE-TO-DMZ rule 20 protocol tcp

set firewall name FROM-INSIDE-TO-DMZ rule 20 destination port 22,80,443

set firewall name FROM-INSIDE-TO-DMZ rule 20 destination address 192.1.1.150/32

set firewall name FROM-INSIDE-TO-DMZ rule 20 source address 10.2.2.100/32 (ADMIN)

set firewall name FROM-INSIDE-TO-DMZ rule 30 action accept

set firewall name FROM-INSIDE-TO-DMZ rule 30 description "Accept DNS to DMZ"

set firewall name FROM-INSIDE-TO-DMZ rule 30 destination address 192.1.1.3/32

set firewall name FROM-INSIDE-TO-DMZ rule 30 protocol udp

set firewall name FROM-INSIDE-TO-DMZ rule 30 destination port 53

set firewall name FROM-INSIDE-TO-DMZ rule 30 source address 10.2.2.100/32 (ADMIN)

set firewall name FROM-INSIDE-TO-DMZ rule 40 action accept

set firewall name FROM-INSIDE-TO-DMZ rule 40 description "Accept normal use of HTTP and HTTPS for public users"

set firewall name FROM-INSIDE-TO-DMZ rule 40 destination address 192.1.1.100/32

set firewall name FROM-INSIDE-TO-DMZ rule 40 destination port 80,443

set firewall name FROM-INSIDE-TO-DMZ rule 40 protocol tcp

Connection entre INSIDE para o Outside:

```
set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 action accept
set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 description "Accept UDP from
ports 5000-6000"
set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 protocol udp
set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 destination port 5000-6000
```

Connection entre OUTSIDE para os servicos DMZ:

```
set firewall name FROM-OUTSIDE-TO-DMZ rule 10 action accept
set firewall name FROM-OUTSIDE-TO-DMZ rule 10 description "Accept ICMP Requests
for public use OUTSIDE-DMZ"
set firewall name FROM-OUTSIDE-TO-DMZ rule 10 destination address 192.1.1.150/32
("public ip address for public use icmp")
set firewall name FROM-OUTSIDE-TO-DMZ rule 10 icmp type 8
set firewall name FROM-OUTSIDE-TO-DMZ rule 10 protocol icmp

set firewall name FROM-OUTSIDE-TO-DMZ rule 20 action accept
set firewall name FROM-OUTSIDE-TO-DMZ rule 20 description "Accept Tcp connection
for public use"
set firewall name FROM-OUTSIDE-TO-DMZ rule 20 destination address 192.1.1.150/32
("public ip address for public use tcp")
set firewall name FROM-OUTSIDE-TO-DMZ rule 20 destination port 443
set firewall name FROM-OUTSIDE-TO-DMZ rule 20 protocol tcp

set firewall name FROM-OUTSIDE-TO-DMZ rule 5 action drop
set firewall group address-group attackers address '200.2.2.20-200.2.2.30'
set firewall name FROM-OUTSIDE-TO-DMZ rule 5 source group address-group
attackers
```

Connection entre o OUTSIDE com o INSIDE:

```
set firewall name FROM-OUTSIDE-TO-INSIDE rule 10 action accept
```

set firewall name FROM-OUTSIDE-TO-INSIDE rule 10 description "Accept Established-related connections"

set firewall name FROM-OUTSIDE-TO-INSIDE rule 10 state established enable

set firewall name FROM-OUTSIDE-TO-INSIDE rule 10 state related enable

Connection entre DMZ para o OUTSIDE, aceitar "todas"

set firewall name FROM-DMZ-TO-OUTSIDE rule 10 action accept

set firewall name FROM-DMZ-TO-OUTSIDE rule 10 description "Accept connections DMZ-OUTSIDE"

set firewall name FROM-DMZ-TO-OUTSIDE rule 10 state established enable

set firewall name FROM-DMZ-TO-OUTSIDE rule 10 state related enable

Connection entre Outside para o servicos de DMZ :

set zone-policy zone INSIDE description "Inside (Internal Network)"

set zone-policy zone INSIDE from DMZ firewall name FROM-DMZ-TO-INSIDE
aplicar as regras na firewall

set zone-policy zone INSIDE from OUTSIDE firewall name FROM-OUTSIDE-TO-INSIDE
aplicar as regras definidas na firewall

set zone-policy zone INSIDE interface eth1

set zone-policy zone INSIDE interface eth2

set zone-policy zone OUTSIDE description "Outside (Internet)"

set zone-policy zone OUTSIDE from DMZ firewall name FROM-DMZ-TO-OUTSIDE

set zone-policy zone OUTSIDE from INSIDE firewall name FROM-INSIDE-TO-OUTSIDE

set zone-policy zone OUTSIDE interface eth0

set zone-policy zone OUTSIDE interface eth3

set zone-policy zone DMZ description "DMZ services"

set zone-policy zone DMZ from INSIDE firewall name FROM-INSIDE-TO-DMZ

set zone-policy zone DMZ from OUTSIDE firewall name FROM-OUTSIDE-TO-DMZ

set zone-policy zone DMZ interface eth4

set service ssh access-control allow user vyos

```
set service ssh listen-address 10.6.1.2
```

```
set service ssh port 22
```

FW2:

"" --- mesma coisa que em cima -- ""

```
set nat source rule 10 outbound-interface eth0
```

```
set nat source rule 10 source address 10.0.0.0/8
```

```
set nat source rule 10 translation address 192.1.0.11-192.1.0.20      RULE 10 E  
PELA eth0!
```

```
set nat source rule 20 outbound-interface eth3
```

```
set nat source rule 20 source address 10.0.0.0/8
```

```
set nat source rule 20 translation address 192.1.0.11-192.1.0.20  
RULE 20 E PELA eth3!
```

```
set protocols static route 0.0.0.0/0 next-hop 10.4.2.1 (Tudo o que for para 0.0.0.0 vai pelo  
10.0.1.1, ou seja tudo o que for para o exterior) eth0
```

```
set protocols static route 0.0.0.0/0 next-hop 10.5.2.1      "" "" eth3
```

```
set protocols static route 10.0.0.0/8 next-hop 10.3.2.2 (Tudo o que for para 10.0.0.0/8 vai  
pelo 10....., ou seja tudo o que for para o interior) eth1
```

```
set protocols static route 10.0.0.0/8 next-hop 10.0.2.2      "" "" eth2
```

```
set protocols static route 192.1.1.0/24 next-hop 10.6.2.1
```

```
set service shh access-control allow user vyos
```

```
set service ssh listen-address 10.6.2.2
```

```
set service ssh port 22
```

