

Universidade de Aveiro

Projeto 1
High-Availability Firewall Scenarios
Segurança em Redes de Comunicações



João Fernandes 93460, Pedro Pereira 93196

Departamento de Electrónica, Telecomunicações e Informática

16 de abril de 2023

Conteúdo

1	Introdução	1
2	Configurações	2
2.1	Topologia de Rede	2
2.2	VPCs	3
2.2.1	VPC Inside	3
2.2.2	VPC Outside	3
2.2.3	Servidor DMZ	3
2.3	Routers	3
2.3.1	Router Inside	4
2.3.2	Router Outside	4
2.4	Firewalls	5
2.4.1	Firewall 1	5
2.4.2	Firewall 2	6
2.4.3	Zones Definition	7
2.4.4	Inter-zone rules	7
2.5	Load Balancers	10
2.5.1	Load Balancer Inside Cluster	10
2.5.2	Load Balancer Outside Cluster	13
2.5.3	Load Balancer DMZ	16
3	Resultados	18
3.1	INSIDE -> OUTSIDE	18
3.1.1	TCP	18
3.1.2	UDP	19

3.1.3	ICMP	19
3.2	INSIDE -> DMZ	19
3.2.1	TCP	19
3.2.2	UDP	20
3.2.3	ICMP	20
3.3	OUTSIDE -> DMZ	20
3.3.1	TCP	20
3.3.2	UDP	21
3.3.3	ICMP	21
4	Análise	22
4.1	Load Balancer na zona DMZ	22
4.2	VM na zona DMZ	22
4.3	Firewalls sem sincronização	23
4.4	Load Balancers sem sincronização	23
4.5	Sincronização de dispositivos durante um ataque DDoS	23
4.6	Script para bloquear DDoS	24

Capítulo 1

Introdução

Este relatório tem como objetivo descrever a resolução do Projeto 1 - "High-Availability Firewall Scenarios" realizado no âmbito da unidade curricular de Segurança em Redes de Comunicações. Este projeto tem como objetivo o desenvolvimento de uma rede com load balancers e firewalls redundantes

Este relatório está dividido em 3 capítulos.

O primeiro capítulo tem como foco a apresentação e descrição do processo de configuração da rede desenvolvida.

No segundo capítulo são mostrados os resultados dos testes operacionais executados na rede.

Finalmente, no terceiro capítulo é assumido um ponto de vista crítico e é feita uma análise da rede e dos seus aspetos mais relevantes.

Capítulo 2

Configurações

Este capítulo tem como objetivo a apresentação e descrição do processo de desenvolvimento e configuração da rede.

2.1 Topologia de Rede

De modo a realizar a correta configuração da rede que nos foi sugerida, o primeiro foco seria a atribuição correta e eficiente dos endereços de ip das interfaces dos dispositivos que a compõem.

Assim, após uma análise cuidadosa, foi elaborada uma solução ilustrada pela seguinte imagem:

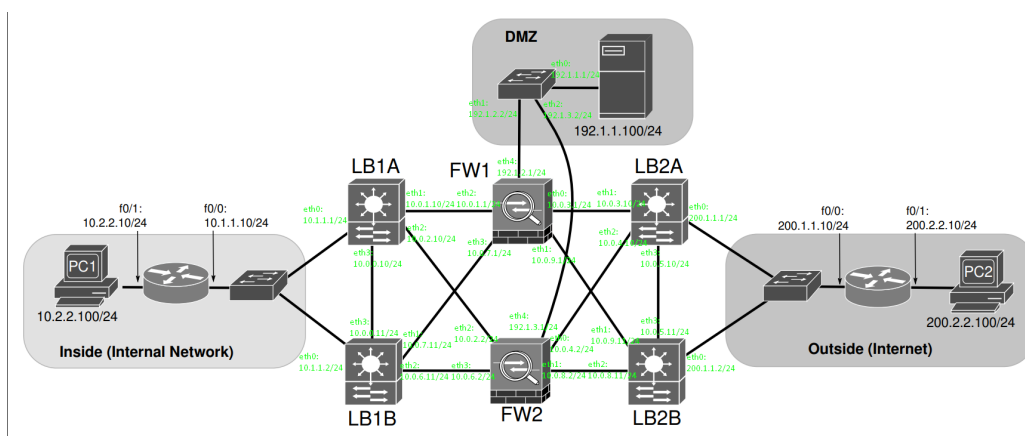


Figure 2.1: Topologia da Rede

É necessário realçar a utilização de um Load Balancer ao invés de um switch na zona DMZ e de uma VM de Linux(Debian LXDE, fornecida no moddle da disciplina) para além de um VPC de GNS3 para simular o servidor DMZ.

2.2 VPCs

Para a configuração dos VPCs, apenas definimos os seus endereços ip e as suas default gateways.

2.2.1 VPC Inside

```
1 set pname PC1
2 ip 10.2.2.100 10.2.2.10 24
```

2.2.2 VPC Outside

```
1 set pname PC2
2 ip 200.2.2.100 200.2.2.10 24
```

2.2.3 Servidor DMZ

A configuração deste dispositivo depende de se utilizamos um VPC do GNS3 ou uma VM de Linux.

Esta primeira secção de código refere-se à configuração do VPC.

```
1 set pname PCDMZ
2 ip 192.1.1.100 192.1.1.1 24
```

Caso utilizemos uma VM de Linux, a configuração é feita da seguinte forma:

```
1 sudo ip link set up dev enp0s3
2 sudo ip addr add 192.1.1.100 dev enp0s3
3 sudo ip route add 192.1.1.1 dev enp0s3
4 sudo ip route add default via 192.1.1.1
```

2.3 Routers

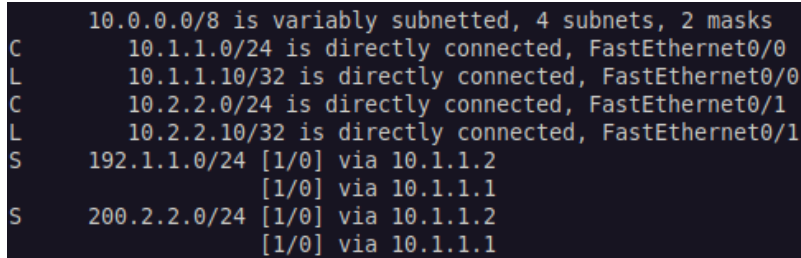
A configuração dos routers, consistiu na definição dos endereços ip das suas interfaces e das suas rotas estáticas.

2.3.1 Router Inside

O router da zona interna possui duas rotas estáticas para a zona Outside (uma para LB1A, uma para LB1B) e duas rotas para a zona DMZ (uma para LB1A, uma para LB1B).

```
1 configure terminal
2 hostname R1
3 interface FastEthernet 0/0
4 ip addr 10.1.1.10 255.255.255.0
5 no shutdown
6 interface FastEthernet 0/1
7 ip addr 10.2.2.10 255.255.255.0
8 no shutdown
9 ip route 200.2.2.0 255.255.255.0 10.1.1.1
10 ip route 200.2.2.0 255.255.255.0 10.1.1.2
11 ip route 192.1.1.0 255.255.255.0 10.1.1.1
12 ip route 192.1.1.0 255.255.255.0 10.1.1.2
13 write
```

A partir destas configurações, obtivemos a seguinte routing table:



```
10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C    10.1.1.0/24 is directly connected, FastEthernet0/0
L    10.1.1.10/32 is directly connected, FastEthernet0/0
C    10.2.2.0/24 is directly connected, FastEthernet0/1
L    10.2.2.10/32 is directly connected, FastEthernet0/1
S    192.1.1.0/24 [1/0] via 10.1.1.2
                  [1/0] via 10.1.1.1
S    200.2.2.0/24 [1/0] via 10.1.1.2
                  [1/0] via 10.1.1.1
```

Figure 2.2: Routing Table do Router Inside

2.3.2 Router Outside

O router da zona Outside possui duas rotas estáticas para acesso à zona interna e à zona DMZ (uma para LB1A, uma para LB1B).

```
1 configure terminal
2 hostname R2
3 interface FastEthernet 0/0
4 ip addr 200.1.1.10 255.255.255.0
5 no shutdown
6 interface FastEthernet 0/1
7 ip addr 200.2.2.10 255.255.255.0
8 no shutdown
9 ip route 192.1.0.0 255.255.254.0 200.1.1.1
10 ip route 192.1.0.0 255.255.254.0 200.1.1.2
11 write
```

Destas configurações resultou a seguinte routing table:

```

S    192.1.0.0/23 [1/0] via 200.1.1.2
      [1/0] via 200.1.1.1
S    192.1.1.0/24 [1/0] via 200.1.1.2
      [1/0] via 200.1.1.1
      200.1.1.0/24 is variably subnetted, 2 subnets, 2 masks
C    200.1.1.0/24 is directly connected, FastEthernet0/0
L    200.1.1.10/32 is directly connected, FastEthernet0/0
      200.2.2.0/24 is variably subnetted, 2 subnets, 2 masks
C    200.2.2.0/24 is directly connected, FastEthernet0/1
L    200.2.2.10/32 is directly connected, FastEthernet0/1

```

Figure 2.3: Routing Table do Router Outside

2.4 Firewalls

Há duas firewalls na rede que controlam o tráfego. Utilizam protocolo nat para enviar pacotes para o exterior e para o DMZ. Para a zona exterior, os pacotes TCP são enviados pelos portos 80 e 443 (HTTP e HTTPS) e os pacotes UDP são enviados pelo porto 53 (DNS). Para o DMZ os pacotes TCP são envias pelos portos 80,443 e 22(HTTP, HTTPS e SSH) e os pacotes UDP são enviados pelo porto 53 (DNS).

2.4.1 Firewall 1

Nesta secção é apresentada a configuração dos endereços ip das interfaces, das rotas estáticas e do protocolo NAT da Firewall 1.

```

1  sudo cp /opt/vyatta/etc/config.boot.default /config/config.boot
2  reboot
3
4  configure
5
6  set system host-name FW1
7
8  set interfaces ethernet eth0 address 10.0.3.1/24
9  set interfaces ethernet eth1 address 10.0.9.1/24
10 set interfaces ethernet eth2 address 10.0.1.1/24
11 set interfaces ethernet eth3 address 10.0.7.1/24
12 set interfaces ethernet eth4 address 192.1.2.1/24
13
14 set protocols static route 200.2.2.0/24 next-hop 10.0.3.10
15 set protocols static route 200.2.2.0/24 next-hop 10.0.9.11
16 set protocols static route 10.2.2.0/24 next-hop 10.0.1.10
17 set protocols static route 10.2.2.0/24 next-hop 10.0.7.11
18
19 set nat source rule 10 outbound-interface eth0
20 set nat source rule 10 source address 10.0.0.0/8
21 set nat source rule 10 translation address 192.1.0.1-192.1.0.10
22
23 set nat source rule 20 outbound-interface eth1
24 set nat source rule 20 source address 10.0.0.0/8

```



```

25 set nat source rule 20 translation address 192.1.0.1-192.1.0.10
26
27 set nat source rule 30 outbound-interface eth4
28 set nat source rule 30 source address 10.0.0.0/8
29 set nat source rule 30 translation address 192.1.0.1-192.1.0.10

```

A partir destas configurações obtivemos a seguinte routing table:

```

C>* 10.0.1.0/24 is directly connected, eth2, 00:01:57
C>* 10.0.3.0/24 is directly connected, eth0, 00:01:37
C>* 10.0.7.0/24 is directly connected, eth3, 00:01:02
C>* 10.0.9.0/24 is directly connected, eth1, 00:01:20
S>* 10.2.2.0/24 [1/0] via 10.0.1.10, eth2, 00:00:34
    *                via 10.0.7.11, eth3, 00:00:34
S>* 200.2.2.0/24 [1/0] via 10.0.3.10, eth0, 00:00:30
    *                via 10.0.9.11, eth1, 00:00:30

```

Figure 2.4: Routing Table da Firewall 1

2.4.2 Firewall 2

Nesta seção é apresentada a configuração dos endereços ip das interfaces, das rotas estáticas e do protocolo NAT da Firewall 2.

```

1 sudo cp /opt/vyatta/etc/config.boot.default /config/config.boot
2 reboot
3
4 configure
5
6 set system host-name FW2
7
8 set interfaces ethernet eth0 address 10.0.4.2/24
9 set interfaces ethernet eth1 address 10.0.8.2/24
10 set interfaces ethernet eth2 address 10.0.2.2/24
11 set interfaces ethernet eth3 address 10.0.6.2/24
12 set interfaces ethernet eth4 address 192.1.3.1/24
13
14 set protocols static route 200.2.2.0/24 next-hop 10.0.4.10
15 set protocols static route 200.2.2.0/24 next-hop 10.0.8.11
16 set protocols static route 10.2.2.0/24 next-hop 10.0.2.10
17 set protocols static route 10.2.2.0/24 next-hop 10.0.6.11
18
19 set nat source rule 10 outbound-interface eth0
20 set nat source rule 10 source address 10.0.0.0/8
21 set nat source rule 10 translation address 192.1.0.11-192.1.0.20
22
23 set nat source rule 20 outbound-interface eth1
24 set nat source rule 20 source address 10.0.0.0/8
25 set nat source rule 20 translation address 192.1.0.11-192.1.0.20
26
27 set nat source rule 30 outbound-interface eth4
28 set nat source rule 30 source address 10.0.0.0/8
29 set nat source rule 30 translation address 192.1.0.11-192.1.0.20
30

```

```
31 commit
32 save
33 exit
```

Destas configurações resultou a seguinte routing table:

```
C>* 10.0.2.0/24 is directly connected, eth2, 00:05:26
C>* 10.0.4.0/24 is directly connected, eth0, 00:05:07
C>* 10.0.6.0/24 is directly connected, eth3, 00:04:31
C>* 10.0.8.0/24 is directly connected, eth1, 00:04:47
S>* 10.2.2.0/24 [1/0] via 10.0.2.10, eth2, 00:04:04
*                      via 10.0.6.11, eth3, 00:04:04
S>* 200.2.2.0/24 [1/0] via 10.0.4.10, eth0, 00:04:00
*                      via 10.0.8.11, eth1, 00:04:00
```

Figure 2.5: Routing Table da Firewall 2

2.4.3 Zones Definition

Nesta secção, configuramos as zonas de acordo com a topologia de rede apresentada anteriormente.

```
1 set zone-policy zone INSIDE description "Inside Network"
2 set zone-policy zone INSIDE interface eth2
3 set zone-policy zone INSIDE interface eth3
4
5 set zone-policy zone OUTSIDE description "Outside Network"
6 set zone-policy zone OUTSIDE interface eth0
7 set zone-policy zone OUTSIDE interface eth1
8
9 set zone-policy zone DMZ description "DMZ"
10 set zone-policy zone DMZ interface eth4
11
12 commit
13 save
14 exit
```

2.4.4 Inter-zone rules

Esta secção dedica-se à descrição das políticas para interações entre as zonas da nossa rede.

Inside to Outside

```
1 set firewall name INSIDE-TO-OUTSIDE rule 10 description "TCP-80"
2 set firewall name INSIDE-TO-OUTSIDE rule 10 action accept
3 set firewall name INSIDE-TO-OUTSIDE rule 10 protocol tcp
4 set firewall name INSIDE-TO-OUTSIDE rule 10 destination port 80
5
```

```

6 set firewall name INSIDE-TO-OUTSIDE rule 11 description "UDP-53"
7 set firewall name INSIDE-TO-OUTSIDE rule 11 action accept
8 set firewall name INSIDE-TO-OUTSIDE rule 11 protocol udp
9 set firewall name INSIDE-TO-OUTSIDE rule 11 destination port 53
10
11 set firewall name INSIDE-TO-OUTSIDE rule 12 description "Inside to Outside ...
    ICMP"
12 set firewall name INSIDE-TO-OUTSIDE rule 12 action accept
13 set firewall name INSIDE-TO-OUTSIDE rule 12 protocol icmp
14 set firewall name INSIDE-TO-OUTSIDE rule 12 icmp type 8
15
16 set firewall name INSIDE-TO-OUTSIDE rule 13 description "TCP-443"
17 set firewall name INSIDE-TO-OUTSIDE rule 13 action accept
18 set firewall name INSIDE-TO-OUTSIDE rule 13 protocol tcp
19 set firewall name INSIDE-TO-OUTSIDE rule 13 destination port 443
20
21 set firewall name TO-INSIDE rule 10 description "Accept ...
    Established-Related Connections"
22 set firewall name TO-INSIDE rule 10 action accept
23 set firewall name TO-INSIDE rule 10 state established enable
24 set firewall name TO-INSIDE rule 10 state related enable
25
26 set zone-policy zone INSIDE from OUTSIDE firewall name TO-INSIDE
27 set zone-policy zone OUTSIDE from INSIDE firewall name INSIDE-TO-OUTSIDE
28
29 commit
30 save
31 exit

```

Inside to DMZ

```

1 set firewall name INSIDE-TO-DMZ rule 10 description "TCP-80"
2 set firewall name INSIDE-TO-DMZ rule 10 action accept
3 set firewall name INSIDE-TO-DMZ rule 10 protocol tcp
4 set firewall name INSIDE-TO-DMZ rule 10 destination address 192.1.1.0/24
5 set firewall name INSIDE-TO-DMZ rule 10 destination port 80
6
7 set firewall name INSIDE-TO-DMZ rule 11 description "UDP-53"
8 set firewall name INSIDE-TO-DMZ rule 11 action accept
9 set firewall name INSIDE-TO-DMZ rule 11 protocol udp
10 set firewall name INSIDE-TO-DMZ rule 11 destination address 192.1.1.0/24
11 set firewall name INSIDE-TO-DMZ rule 11 destination port 53
12
13 set firewall name INSIDE-TO-DMZ rule 12 description "ICMP"
14 set firewall name INSIDE-TO-DMZ rule 12 action accept
15 set firewall name INSIDE-TO-DMZ rule 12 protocol icmp
16 set firewall name INSIDE-TO-DMZ rule 12 icmp type 8
17 set firewall name INSIDE-TO-DMZ rule 12 destination address 192.1.1.0/24
18
19 set firewall name INSIDE-TO-DMZ rule 13 description "TCP-443"
20 set firewall name INSIDE-TO-DMZ rule 13 action accept
21 set firewall name INSIDE-TO-DMZ rule 13 protocol tcp
22 set firewall name INSIDE-TO-DMZ rule 13 destination address 192.1.1.0/24

```

```

23 set firewall name INSIDE-TO-DMZ rule 13 destination port 443
24
25 set firewall name INSIDE-TO-DMZ rule 15 description "TCP-22"
26 set firewall name INSIDE-TO-DMZ rule 15 action accept
27 set firewall name INSIDE-TO-DMZ rule 15 protocol tcp
28 set firewall name INSIDE-TO-DMZ rule 15 destination address 192.1.1.0/24
29 set firewall name INSIDE-TO-DMZ rule 15 destination port 22
30
31 set zone-policy zone INSIDE from DMZ firewall name TO-INSIDE
32 set zone-policy zone DMZ from INSIDE firewall name INSIDE-TO-DMZ
33
34 commit
35 save
36 exit

```

Outside to DMZ

```

1 set firewall name OUTSIDE-TO-DMZ rule 10 description "TCP-80"
2 set firewall name OUTSIDE-TO-DMZ rule 10 action accept
3 set firewall name OUTSIDE-TO-DMZ rule 10 protocol tcp
4 set firewall name OUTSIDE-TO-DMZ rule 10 destination address 192.1.1.100
5 set firewall name OUTSIDE-TO-DMZ rule 10 destination port 80
6
7 set firewall name OUTSIDE-TO-DMZ rule 11 description "UDP-53"
8 set firewall name OUTSIDE-TO-DMZ rule 11 action accept
9 set firewall name OUTSIDE-TO-DMZ rule 11 protocol udp
10 set firewall name OUTSIDE-TO-DMZ rule 11 destination address 192.1.1.100
11 set firewall name OUTSIDE-TO-DMZ rule 11 destination port 53
12
13 set firewall name OUTSIDE-TO-DMZ rule 12 description "ICMP"
14 set firewall name OUTSIDE-TO-DMZ rule 12 action accept
15 set firewall name OUTSIDE-TO-DMZ rule 12 protocol icmp
16 set firewall name OUTSIDE-TO-DMZ rule 12 icmp type 8
17 set firewall name OUTSIDE-TO-DMZ rule 12 destination address 192.1.1.100
18
19 set firewall name OUTSIDE-TO-DMZ rule 13 description "TCP-443"
20 set firewall name OUTSIDE-TO-DMZ rule 13 action accept
21 set firewall name OUTSIDE-TO-DMZ rule 13 protocol tcp
22 set firewall name OUTSIDE-TO-DMZ rule 13 destination address 192.1.1.100
23 set firewall name OUTSIDE-TO-DMZ rule 13 destination port 443
24
25 set firewall name OUTSIDE-TO-DMZ rule 15 description "TCP-22"
26 set firewall name OUTSIDE-TO-DMZ rule 15 action accept
27 set firewall name OUTSIDE-TO-DMZ rule 15 protocol tcp
28 set firewall name OUTSIDE-TO-DMZ rule 15 destination address 192.1.1.100
29 set firewall name OUTSIDE-TO-DMZ rule 15 destination port 22
30
31 set firewall name OUTSIDE-TO-DMZ rule 17 description "Block private addresses"
32 set firewall name OUTSIDE-TO-DMZ rule 17 action drop
33 set firewall name OUTSIDE-TO-DMZ rule 17 destination address ...
    10.0.0.0-10.255.255.255
34

```

```
35 set firewall name DMZ-TO-OUTSIDE rule 10 description "Accept ...  
    Established-Related Connections"  
36 set firewall name DMZ-TO-OUTSIDE rule 10 action accept  
37 set firewall name DMZ-TO-OUTSIDE rule 10 state established enable  
38 set firewall name DMZ-TO-OUTSIDE rule 10 state related enable  
39  
40 set zone-policy zone DMZ from OUTSIDE firewall name OUTSIDE-TO-DMZ  
41 set zone-policy zone OUTSIDE from DMZ firewall name DMZ-TO-OUTSIDE  
42  
43  
44 commit  
45 save  
46 exit
```

2.5 Load Balancers

Os Load Balancers da rede desenvolvida estão organizados em dois clusters, um comunica diretamente com o router de acesso à zona interna, outra com o router de acesso à zona Outside. Para além destes, foi configurado também o Load Balancer que foi colocado na zona DMZ.

2.5.1 Load Balancer Inside Cluster

Load Balancer 1A

Nesta secção é apresentada a configuração dos endereços ip das interfaces, das rotas estáticas e do serviço de load balancing do Load Balancer 1A.

```
1 sudo cp /opt/vyatta/etc/config.boot.default /config/config.boot  
2 reboot  
3  
4 configure  
5  
6 set system host-name LB1A  
7  
8 set interfaces ethernet eth0 address 10.1.1.1/24  
9 set interfaces ethernet eth1 address 10.0.1.10/24  
10 set interfaces ethernet eth2 address 10.0.2.10/24  
11 set interfaces ethernet eth3 address 10.0.0.10/24  
12  
13 set protocols static route 10.2.2.0/24 next-hop 10.1.1.10  
14  
15 set load-balancing wan interface-health eth1 nexthop 10.0.1.1  
16 set load-balancing wan interface-health eth2 nexthop 10.0.2.2  
17 set load-balancing wan rule 1 inbound-interface eth0  
18 set load-balancing wan rule 1 interface eth1 weight 1  
19 set load-balancing wan rule 1 interface eth2 weight 1  
20 set load-balancing wan sticky-connections inbound  
21 set load-balancing wan disable-source-nat
```

```
22
23 commit
24 save
25 exit
```

Apresenta-se a seguir a routing table que obtivemos:

```
C>* 10.0.0.0/24 is directly connected, eth3, 00:10:27
C>* 10.0.1.0/24 is directly connected, eth1, 00:10:38
C>* 10.0.2.0/24 is directly connected, eth2, 00:10:49
C>* 10.1.1.0/24 is directly connected, eth0, 00:10:55
S>* 10.2.2.0/24 [1/0] via 10.1.1.10, eth0, 00:10:14
C>* 192.168.100.0/24 is directly connected, eth3v10, 00:10:02
```

Figure 2.6: Routing Table do Load Balancer 1A

Os detalhes sobre a configuração do serviço de load balancing apresenta-se a seguidamente:

```
wan {
  disable-source-nat
  interface-health eth1 {
    failure-count 1
    nexthop 10.0.1.1
    success-count 1
  }
  interface-health eth2 {
    failure-count 1
    nexthop 10.0.2.2
    success-count 1
  }
  rule 1 {
    inbound-interface eth0
    interface eth1 {
      weight 1
    }
    interface eth2 {
      weight 1
    }
    protocol all
  }
  sticky-connections {
    inbound
  }
}
```

Figure 2.7: Load Balancing do Load Balancer 1A

Load Balancer 1B

Nesta secção é apresentada a configuração dos endereços ip das interfaces, das rotas estáticas e do serviço de load balancing do Load Balancer 1B.

```
1 sudo cp /opt/vyatta/etc/config.boot.default /config/config.boot
2 reboot
3
4 configure
5
6 set system host-name LB1B
```

```

7
8 set interfaces ethernet eth0 address 10.1.1.2/24
9 set interfaces ethernet eth1 address 10.0.7.11/24
10 set interfaces ethernet eth2 address 10.0.6.11/24
11 set interfaces ethernet eth3 address 10.0.0.11/24
12
13 set protocols static route 10.2.2.0/24 next-hop 10.1.1.10
14
15 set load-balancing wan interface-health eth1 nexthop 10.0.7.1
16 set load-balancing wan interface-health eth2 nexthop 10.0.6.2
17 set load-balancing wan rule 1 inbound-interface eth0
18 set load-balancing wan rule 1 interface eth1 weight 1
19 set load-balancing wan rule 1 interface eth2 weight 1
20 set load-balancing wan sticky-connections inbound
21 set load-balancing wan disable-source-nat
22
23 commit
24 save
25 exit

```

A routing table obtida é a seguinte:

```

C>* 10.0.0.0/24 is directly connected, eth3, 00:23:21
C * 10.0.6.0/24 is directly connected, eth1, 00:23:36
C>* 10.0.6.0/24 is directly connected, eth2, 00:23:49
C * 10.0.7.0/24 is directly connected, eth1, 00:23:36
C>* 10.0.7.0/24 is directly connected, eth2, 00:23:49
C>* 10.1.1.0/24 is directly connected, eth0, 00:24:02
S>* 10.2.2.0/24 [1/0] via 10.1.1.10, eth0, 00:23:04

```

Figure 2.8: Routing Table do Load Balancer 1B

Seguem-se os detalhes sobre a configuração do serviço de load balancing:

```

wan {
  disable-source-nat
  interface-health eth1 {
    failure-count 1
    nexthop 10.0.7.1
    success-count 1
  }
  interface-health eth2 {
    failure-count 1
    nexthop 10.0.6.2
    success-count 1
  }
  rule 1 {
    inbound-interface eth0
    interface eth1 {
      weight 1
    }
    interface eth2 {
      weight 1
    }
    protocol all
  }
  sticky-connections {
    inbound
  }
}

```

Figure 2.9: Load Balancing do Load Balancer 1B

VRRP and Conntrack Sync

Nesta secção é apresentada a configuração dos mecanismos de high-availability (Virtual Router Redundancy Protocol - VRRP) e connection state synchronization (conntrack-sync) para o cluster de Load Balancers que comunica diretamente com o router de acesso à zona interna, ao qual pertencem os Load Balancers 1A e 1B.

```
1 set high-availability vrrp group LB1Cluster vrid 10
2 set high-availability vrrp group LB1Cluster interface eth3
3 set high-availability vrrp group LB1Cluster virtual-address 192.168.100.1/24
4 set high-availability vrrp sync-group LB1Cluster member LB1Cluster
5 set high-availability vrrp group LB1Cluster rfc3768-compatibility
6
7 set service conntrack-sync accept-protocol 'tcp,udp,icmp'
8 set service conntrack-sync failover-mechanism vrrp sync-group LB1Cluster
9 set service conntrack-sync interface eth3
10 set service conntrack-sync mcast-group 225.0.0.50
11 set service conntrack-sync disable-external-cache
12
13 commit
14 save
15 exit
```

2.5.2 Load Balancer Outside Cluster

Load Balancer 2A

Nesta secção é apresentada a configuração dos endereços ip das interfaces, das rotas estáticas e do serviço de load balancing do Load Balancer 2A.

```
1 sudo cp /opt/vyatta/etc/config.boot.default /config/config.boot
2 reboot
3
4 configure
5
6 set system host-name LB2A
7
8 set interfaces ethernet eth0 address 200.1.1.1/24
9 set interfaces ethernet eth1 address 10.0.3.10/24
10 set interfaces ethernet eth2 address 10.0.4.10/24
11 set interfaces ethernet eth3 address 10.0.5.10/24
12
13 set protocols static route 200.2.2.0/24 next-hop 200.1.1.10
14
15 set load-balancing wan interface-health eth1 nexthop 10.0.3.1
16 set load-balancing wan interface-health eth2 nexthop 10.0.4.2
17 set load-balancing wan rule 1 inbound-interface eth0
18 set load-balancing wan rule 1 interface eth1 weight 1
19 set load-balancing wan rule 1 interface eth2 weight 1
20 set load-balancing wan sticky-connections inbound
```



```
21 set load-balancing wan disable-source-nat
22
23 commit
24 save
25 exit
```

Obtivemos a seguinte routing table:

```
C>* 10.0.3.0/24 is directly connected, eth1, 00:30:41
C>* 10.0.4.0/24 is directly connected, eth2, 00:30:57
C>* 10.0.5.0/24 is directly connected, eth3, 00:30:26
C>* 192.168.101.0/24 is directly connected, eth3v11, 00:30:04
C>* 200.1.1.0/24 is directly connected, eth0, 00:31:14
S>* 200.2.2.0/24 [1/0] via 200.1.1.10, eth0, 00:30:15
```

Figure 2.10: Routing Table do Load Balancer 2A

O serviço de load balancing ficou configurado da seguinte maneira:

```
wan {
  disable-source-nat
  interface-health eth1 {
    failure-count 1
    nexthop 10.0.3.1
    success-count 1
  }
  interface-health eth2 {
    failure-count 1
    nexthop 10.0.4.2
    success-count 1
  }
  rule 1 {
    inbound-interface eth0
    interface eth1 {
      weight 1
    }
    interface eth2 {
      weight 1
    }
    protocol all
  }
  sticky-connections {
    inbound
  }
}
```

Figure 2.11: Load Balancing do Load Balancer 2A

Load Balancer 2B

Nesta secção é apresentada a configuração dos endereços ip das interfaces, das rotas estáticas e do serviço de load balancing do Load Balancer 2B.

```
1 sudo cp /opt/vyatta/etc/config.boot.default /config/config.boot
2 reboot
3
4 configure
5
6 set system host-name LB2B
```

```

7
8 set interfaces ethernet eth0 address 200.1.1.2/24
9 set interfaces ethernet eth1 address 10.0.9.11/24
10 set interfaces ethernet eth2 address 10.0.8.11/24
11 set interfaces ethernet eth3 address 10.0.5.11/24
12
13 set protocols static route 200.2.2.0/24 next-hop 200.1.1.10
14
15 set load-balancing wan interface-health eth1 nexthop 10.0.9.1
16 set load-balancing wan interface-health eth2 nexthop 10.0.8.2
17 set load-balancing wan rule 1 inbound-interface eth0
18 set load-balancing wan rule 1 interface eth1 weight 1
19 set load-balancing wan rule 1 interface eth2 weight 1
20 set load-balancing wan sticky-connections inbound
21 set load-balancing wan disable-source-nat
22
23 commit
24 save
25 exit

```

A partir destas configurações, obtivemos a seguinte routing table:

```

C>* 10.0.5.0/24 is directly connected, eth3, 00:39:59
C * 10.0.8.0/24 is directly connected, eth1, 00:40:15
C>* 10.0.8.0/24 is directly connected, eth2, 00:40:29
C * 10.0.9.0/24 is directly connected, eth1, 00:40:15
C>* 10.0.9.0/24 is directly connected, eth2, 00:40:29
C>* 200.1.1.0/24 is directly connected, eth0, 00:40:42
S>* 200.2.2.0/24 [1/0] via 200.1.1.10, eth0, 00:39:42

```

Figure 2.12: Routing Table do Load Balancer 2B

Seguidamente, apresentamos os detalhes sobre a configuração do serviço de load balancing:

```

wan {
  disable-source-nat
  interface-health eth1 {
    failure-count 1
    nexthop 10.0.9.1
    success-count 1
  }
  interface-health eth2 {
    failure-count 1
    nexthop 10.0.8.2
    success-count 1
  }
  rule 1 {
    inbound-interface eth0
    interface eth1 {
      weight 1
    }
    interface eth2 {
      weight 1
    }
    protocol all
  }
  sticky-connections {
    inbound
  }
}

```

Figure 2.13: Load Balancing do Load Balancer 2B

VRRP and Conntrack Sync

Nesta secção é apresentada a configuração dos mecanismos de high-availability (Virtual Router Redundancy Protocol - VRRP) e connection state synchronization (conntrack-sync) para o cluster de Load Balancers que comunica diretamente com o router de acesso à zona Outside, ao qual pertencem os Load Balancers 2A e 2B.

```
1 set high-availability vrrp group LB2Cluster vrid 11
2 set high-availability vrrp group LB2Cluster interface eth3
3 set high-availability vrrp group LB2Cluster virtual-address 192.168.101.1/24
4 set high-availability vrrp sync-group LB2Cluster member LB2Cluster
5 set high-availability vrrp group LB2Cluster rfc3768-compatibility
6
7 set service conntrack-sync accept-protocol 'tcp,udp,icmp'
8 set service conntrack-sync failover-mechanism vrrp sync-group LB2Cluster
9 set service conntrack-sync interface eth3
10 set service conntrack-sync mcast-group 225.0.0.50
11 set service conntrack-sync disable-external-cache
12
13 commit
14 save
15 exit
```

2.5.3 Load Balancer DMZ

Nesta secção é apresentada a configuração dos endereços ip das interfaces, das rotas estáticas e do serviço de load balancing do Load Balancer DMZ.

```
1 sudo cp /opt/vyatta/etc/config.boot.default /config/config.boot
2 reboot
3
4 configure
5
6 set interfaces ethernet eth0 address 192.1.1.1/24
7 set interfaces ethernet eth1 address 192.1.2.2/24
8 set interfaces ethernet eth2 address 192.1.3.2/24
9
10 set load-balancing wan interface-health eth1 nexthop 192.1.2.1
11 set load-balancing wan interface-health eth2 nexthop 192.1.3.1
12 set load-balancing wan rule 1 inbound-interface eth0
13 set load-balancing wan rule 1 interface eth1 weight 1
14 set load-balancing wan rule 1 interface eth2 weight 1
15 set load-balancing wan sticky-connections inbound
16 set load-balancing wan disable-source-nat
17
18 commit
19 save
20 exit
```

A partir destas configurações, obtivemos a seguinte routing table:

```
C>* 192.1.1.0/24 is directly connected, eth0, 00:48:18
C>* 192.1.2.0/24 is directly connected, eth1, 01:32:26
C>* 192.1.3.0/24 is directly connected, eth2, 01:32:29
```

Figure 2.14: Routing Table do Load Balancer DMZ

Seguidamente, apresentamos os detalhes sobre a configuração do serviço de load balancing:

```
wan {
    disable-source-nat
    interface-health eth1 {
        failure-count 1
        nexthop 192.1.2.1
        success-count 1
    }
    interface-health eth2 {
        failure-count 1
        nexthop 192.1.3.1
        success-count 1
    }
    rule 1 {
        inbound-interface eth0
        interface eth1 {
            weight 1
        }
        interface eth2 {
            weight 1
        }
        protocol all
    }
    sticky-connections {
        inbound
    }
}
```

Figure 2.15: Load Balancing do Load Balancer DMZ

Capítulo 3

Resultados

Neste capítulo, apresentamos os resultados de teste operacionais efetuados na rede desenvolvida. Para além do que será seguidamente apresentado, foram também executadas capturas de pacotes efetuadas utilizando a ferramenta de análise de pacotes Wireshark. Estas capturas foram enviadas em anexo na entrega do projeto.

3.1 INSIDE -> OUTSIDE

3.1.1 TCP

Ao usar um VPC normal não era possível obter a ligação TCP. Colocamos então uma VM Debian e conseguimos ter conexão com sucesso

```
PC1> ping 200.2.2.100 -P 6 -p 80
Connect 80@200.2.2.100 seq=1 ttl=59 time=25.879 ms
SendData 80@200.2.2.100 seq=1 ttl=59 time=28.241 ms
Close 80@200.2.2.100 seq=1 ttl=59 time=41.336 ms
Connect 80@200.2.2.100 seq=2 ttl=59 time=61.784 ms
SendData 80@200.2.2.100 seq=2 ttl=59 time=26.073 ms
Close 80@200.2.2.100 seq=2 ttl=59 time=27.233 ms
Connect 80@200.2.2.100 seq=3 ttl=59 time=29.319 ms
SendData 80@200.2.2.100 seq=3 ttl=59 time=25.371 ms
Close 80@200.2.2.100 seq=3 ttl=59 time=44.927 ms
Connect 80@200.2.2.100 seq=4 ttl=59 time=56.553 ms
SendData 80@200.2.2.100 seq=4 ttl=59 time=31.222 ms
Close 80@200.2.2.100 seq=4 ttl=59 time=30.212 ms
Connect 80@200.2.2.100 seq=5 ttl=59 time=38.015 ms
SendData 80@200.2.2.100 seq=5 ttl=59 time=31.318 ms
Close 80@200.2.2.100 seq=5 ttl=59 time=31.294 ms
```

Figure 3.1: Ping TCP no porto 80 para o Outside

3.1.2 UDP

```
PC1> ping 200.2.2.100 -P 17 -p 53
84 bytes from 200.2.2.100 udp_seq=1 ttl=59 time=38.507 ms
84 bytes from 200.2.2.100 udp_seq=2 ttl=59 time=38.286 ms
84 bytes from 200.2.2.100 udp_seq=3 ttl=59 time=43.141 ms
84 bytes from 200.2.2.100 udp_seq=4 ttl=59 time=25.518 ms
84 bytes from 200.2.2.100 udp_seq=5 ttl=59 time=28.498 ms
```

Figure 3.2: Ping UDP no porto 53 para o Outside

3.1.3 ICMP

```
PC1> ping 200.2.2.100
84 bytes from 200.2.2.100 icmp_seq=1 ttl=59 time=23.552 ms
84 bytes from 200.2.2.100 icmp_seq=2 ttl=59 time=25.507 ms
84 bytes from 200.2.2.100 icmp_seq=3 ttl=59 time=27.485 ms
84 bytes from 200.2.2.100 icmp_seq=4 ttl=59 time=28.432 ms
84 bytes from 200.2.2.100 icmp_seq=5 ttl=59 time=29.476 ms
```

Figure 3.3: Ping ICMP para o Outside

3.2 INSIDE -> DMZ

3.2.1 TCP

Tal como se verificou na situação Inside-Outside, também tivemos aqui de implementar uma VM Debian para conseguirmos obter ligação através do protocolo TCP

```
PC1> ping 192.1.1.100 -P 6 -p 22
Connect 22@192.1.1.100 seq=1 ttl=60 time=15.656 ms
SendData 22@192.1.1.100 seq=1 ttl=60 time=22.469 ms
Close 22@192.1.1.100 timeout
Connect 22@192.1.1.100 seq=2 ttl=60 time=19.498 ms
SendData 22@192.1.1.100 seq=2 ttl=60 time=19.494 ms
Close 22@192.1.1.100 timeout
Connect 22@192.1.1.100 seq=3 ttl=60 time=18.532 ms
SendData 22@192.1.1.100 seq=3 ttl=60 time=19.537 ms
Close 22@192.1.1.100 timeout
Connect 22@192.1.1.100 seq=4 ttl=60 time=19.532 ms
SendData 22@192.1.1.100 seq=4 ttl=60 time=20.442 ms
Close 22@192.1.1.100 timeout
Connect 22@192.1.1.100 seq=5 ttl=60 time=19.555 ms
SendData 22@192.1.1.100 seq=5 ttl=60 time=21.469 ms
Close 22@192.1.1.100 timeout
PC1> █
```

Figure 3.4: Ping TCP no porto 22 para o DMZ

3.2.2 UDP

Não conseguimos pingar o DMZ, quando este se tratava de uma VM, pois o pacote gerado pelo PC1 é considerado "Packet malformed" e o VyOS não envia resposta. Então foi trocado para um VPC normal e assim conseguimos obter resposta.

```
PC1> ping 192.1.1.100 -P 17 -p 53
84 bytes from 192.1.1.100 udp_seq=1 ttl=60 time=18.701 ms
84 bytes from 192.1.1.100 udp_seq=2 ttl=60 time=15.792 ms
84 bytes from 192.1.1.100 udp_seq=3 ttl=60 time=16.747 ms
84 bytes from 192.1.1.100 udp_seq=4 ttl=60 time=17.700 ms
84 bytes from 192.1.1.100 udp_seq=5 ttl=60 time=16.742 ms
```

Figure 3.5: Ping UDP no porto 53 para o DMZ

3.2.3 ICMP

```
PC1> ping 192.1.1.100
84 bytes from 192.1.1.100 icmp_seq=1 ttl=60 time=19.731 ms
84 bytes from 192.1.1.100 icmp_seq=2 ttl=60 time=53.410 ms
84 bytes from 192.1.1.100 icmp_seq=3 ttl=60 time=17.780 ms
84 bytes from 192.1.1.100 icmp_seq=4 ttl=60 time=17.803 ms
84 bytes from 192.1.1.100 icmp_seq=5 ttl=60 time=16.736 ms
```

Figure 3.6: Ping ICMP para o DMZ

3.3 OUTSIDE -> DMZ

3.3.1 TCP

```
PC2> ping 192.1.1.100 -P 6 -p 80
Connect 80@192.1.1.100 seq=1 ttl=60 time=21.503 ms
SendData 80@192.1.1.100 seq=1 ttl=60 time=19.520 ms
Close 80@192.1.1.100 seq=1 ttl=60 time=19.567 ms
Connect 80@192.1.1.100 seq=2 ttl=60 time=18.696 ms
SendData 80@192.1.1.100 seq=2 ttl=60 time=19.529 ms
Close 80@192.1.1.100 seq=2 ttl=60 time=21.494 ms
Connect 80@192.1.1.100 seq=3 ttl=60 time=17.565 ms
SendData 80@192.1.1.100 seq=3 ttl=60 time=20.460 ms
Close 80@192.1.1.100 seq=3 ttl=60 time=21.491 ms
Connect 80@192.1.1.100 seq=4 ttl=60 time=19.576 ms
SendData 80@192.1.1.100 seq=4 ttl=60 time=19.549 ms
Close 80@192.1.1.100 seq=4 ttl=60 time=21.461 ms
Connect 80@192.1.1.100 seq=5 ttl=60 time=19.497 ms
SendData 80@192.1.1.100 seq=5 ttl=60 time=19.636 ms
Close 80@192.1.1.100 seq=5 ttl=60 time=21.483 ms
```

Figure 3.7: Ping TCP no porto 80 para o DMZ

3.3.2 UDP

```
PC2> ping 192.1.1.100 -P 17 -p 53
84 bytes from 192.1.1.100 udp_seq=1 ttl=60 time=21.607 ms
84 bytes from 192.1.1.100 udp_seq=2 ttl=60 time=18.700 ms
84 bytes from 192.1.1.100 udp_seq=3 ttl=60 time=14.819 ms
84 bytes from 192.1.1.100 udp_seq=4 ttl=60 time=18.702 ms
84 bytes from 192.1.1.100 udp_seq=5 ttl=60 time=18.748 ms
```

Figure 3.8: Ping UDP no porto 53 para o DMZ

3.3.3 ICMP

```
PC2> ping 192.1.1.100
84 bytes from 192.1.1.100 icmp_seq=1 ttl=60 time=18.684 ms
84 bytes from 192.1.1.100 icmp_seq=2 ttl=60 time=12.918 ms
84 bytes from 192.1.1.100 icmp_seq=3 ttl=60 time=18.719 ms
84 bytes from 192.1.1.100 icmp_seq=4 ttl=60 time=18.743 ms
84 bytes from 192.1.1.100 icmp_seq=5 ttl=60 time=17.827 ms
```

Figure 3.9: Ping ICMP para o DMZ

Capítulo 4

Análise

Neste capítulo iremos abordar temas importantes relativos à rede desenvolvida e aos protocolos e serviços utilizados pela mesma.

4.1 Load Balancer na zona DMZ

Durante a configuração da rede, enfrentámos dificuldades com o routing de pacotes à saída desta mesma zona. Possuindo apenas uma default route, o servidor DMZ cria problemas na comunicação entre a sua zona e a zona Inside.

Tomando por exemplo um fluxo de tráfego que origina no PC1, com destino ao servidor DMZ, atravessando a firewall 2, se a resposta originada pelo servidor DMZ for enviada pela firewall 1, é impossível este efetuar a tradução utilizando o protocolo NAT (visto que este não observou a saída do pacote original e as firewalls não se encontram sincronizadas). Como consequência o tráfego entre as zonas Inside e DMZ era inconsistente e propício a falhas.

A utilização de um Load Balancer na zona DMZ solucionou o nosso problema devido ao modo como este é capaz de memorizar a interface por onde o pacote original é recebido e envia a resposta pela mesma interface, garantindo assim o correto funcionamento da rede.

4.2 VM na zona DMZ

A utilização de uma VM de Linux (Debian LXDE, fornecida no moodle da disciplina), serviu para solucionar os problemas que surgiram na comunicação através do protocolo TCP quando utilizámos um VPC do GNS3.

Deve-se realçar que, apesar de a VM conseguir comunicar através de protocolo TCP, esta não é capaz de comunicar utilizando protocolo UDP (enquanto que o VPC consegue).

4.3 Firewalls sem sincronização

A necessidade de sincronizar as firewalls surgiu inicialmente da existencia dos mecanismos NAT/PAT.

Sem a sincronização das firewalls, os routers poderiam criar um fluxo de tráfego onde uma firewall iria receber pacotes de resposta da rede "Outside" em que os IPs não iriam estar na tabela de tradução NAT pois estes teriam sido enviados por outra firewall.

A sincronização dos load-balancers permite que não seja necessária a sincronização das firewalls pois garantem que os pacotes de respostas serão reencaminhados pela mesma firewall. Fazem isto memorizando a interface do load balancer por onde passaram os pacotes pedidos.

4.4 Load Balancers sem sincronização

A hipótese da implementação desta rede não requerir sincronização entre os load balancers só seria possível através da utilização de um algoritmo de load balancing que garantisse que os pacotes de respostas fossem sempre enviados para a mesma firewall por onde o pacote de original foi recebido, independentemente do load balancer que está a direccionar o tráfego.

O algoritmo de load balancing com base em IP hashing pode servir para tornar esta opção possível. Este algoritmo utiliza uma função de hashing que aceita o endereço IP como parâmetro e produz um valor hash. Tendo certas gamas de valores associadas a cada uma das rotas possíveis, a escolha do destino do pacote dependerá assim da gama à qual pertence o valor obtido. Deste modo, é possível prever que pacotes provenientes de um mesmo endereço IP específico serão sempre enviados para o mesmo destino, independentemente do load balancer.

Assim, desde que os load balancers partilhem a mesma função de hashing e a mesma associação entre gamas de valores e dispositivos destinatários, podemos concluir que este algoritmo possibilita a utilização de load balancers não sincronizados.

4.5 Sincronização de dispositivos durante um ataque DDoS

Durante um ataque DDoS, o atacante envia um grande volume de tráfego para o sistema alvo sobrecarregando-o e fazendo com que este fique inacessível para os utilizadores.

A sincronização dos estados dos dispositivos/conexões é o processo de manutenção e a partilha de informações sobre o estado dos dispositivos de rede e as suas conexões entre vários dispositivos. A informação pode incluir ligações que estão abertas, os dispositivos disponíveis e os recursos que estão a ser usados.

Com este sistema, grupos de dispositivos na rede ficam sincronizados e partilham a mesma informação sobre os seus estados e ligações. Desta maneira se um atacante ganhar acesso a esta informação, poderá usá-la para identificar os dispositivos e ligações mais vulneráveis, sendo mais fácil explorar vulnerabilidades na rede.

4.6 Script para bloquear DDoS

Foi criado um script python para detetar DDoS(Distributed Denial of Service).

Para detecção de potenciais ips malignos usamos a biblioteca Scapy para fazer sniff dos pacotes que são enviados para o DMZ. São lidos os IPs dos pacotes e são adicionados a uma lista onde são contadas as ocorrências de cada IP. Para efeitos de teste, sempre que forem detetados 5 pacotes com o mesmo IP em menos de 10 segundos, este é considerado um potencial DDoS e será bloqueado.

Para bloquear os IPs temos de estabelecer uma ligação com as Firewalls. Para tal usamos a biblioteca netmiko que serve para interagir com os dispositivos da rede. Assim conseguimos enviar os comandos VyOS para as Firewalls.