Trabalho prático 1 de Segurança em Redes de Computadores

Autores

- Ana Vidal (118408)
- Simão Andrade (118345)

Estrutura do Relatório

1. Introdução;
2. Objetivos;
3. Política de segurança;
4. Ponto 9:
    1. Rotas de Rede e conectividade;
    2. Load-Balancers;
    3. Sincronização de Estados de Dispositivos;
    4. Testes de Funcionamento;
    5. Questões Finais;
5. Ponto 10:
    1. Configuração do Servidor DMZ;
    2. Configuração da Firewall;
    3. Zonas e Regras;
    4. Testes de Funcionamento;
6. Conclusão;

Introdução

Nos dias de hoje, a continuidade operacional e a segurança das redes desempenham um papel crítico no ambiente empresarial. No âmbito da segurança cibernética, os *Firewalls* assumem uma importância inegável na proteção dos ativos e na defesa contra ameaças digitais. Este trabalho tem como objetivo explorar os cenários de *Firewalls* de alta disponibilidade utilizando o VyOS, seguindo um conjunto de politicas previamente definidas assegurando a confidencialidade e integridade da rede. Focar-nos-emos também na configuração de *Firewalls* e *Load Balancers* redundantes e na distribuição de carga de tráfego, de modo a garantir a disponibilidade contínua dos serviços de rede, avaliando os riscos associados a esta redundância. Adicionalmente, iremos implementar funcionalidades como a sincronização de estados, que permite a conexão entre os dispositivos de *Load Balancers*.

Objetivos

Apresentar um relatório dos **testes de configuração** e de **funcionamento** dos cenários descritos nos pontos `9` e `10` do guia laboratorial “High-Availability Firewall Scenarios”.

Temos as seguintes tarefas a serem realizadas:

- [x] Firewall and load-balancers deployment (2 valores).
- [x] Network routing and connectivity (2 valores).
- [x] Devices state synchronization (3 valores).
- [x] Zones definition (3 valores).
- [x] Inter-zone rules (6 valores).
- [x] Report (4 valores).

Política de Segurança

A política de segurança deve priorizar a proteção dos recursos da rede, garantindo que apenas o tráfego necessário, seguro e autorizado seja permitido e implementando medidas para mitigar possíveis ataques.

Com base nisso, as foram definidas as seguintes diretrizes de segurança a serem implementadas:

Código	Descrição
D_01	A rede interna irá receber apenas o tráfego estabelecido por esta, bloqueando todo o tráfego não solicitado.
D_02	As comunicações para a rede exterior serão apenas permitidas para continuidade de serviços (e.g. acesso remoto a serviços, internet).
D_03	A rede interna e a DMZ deverão ser capazes de manter a sua disponibilidade e conseguir gerir todos os pedidos recebidos.
D_04	O acesso ao servidor DMZ será permitido apenas durante o horário laboral.
D_05	Os serviços da empresa só podem ser acedidos com dispositivos ligados á rede da empresa.
D_06	O acesso remoto ao servidor DMZ é restrito apenas ao administrador.

Ponto 9

Topologia

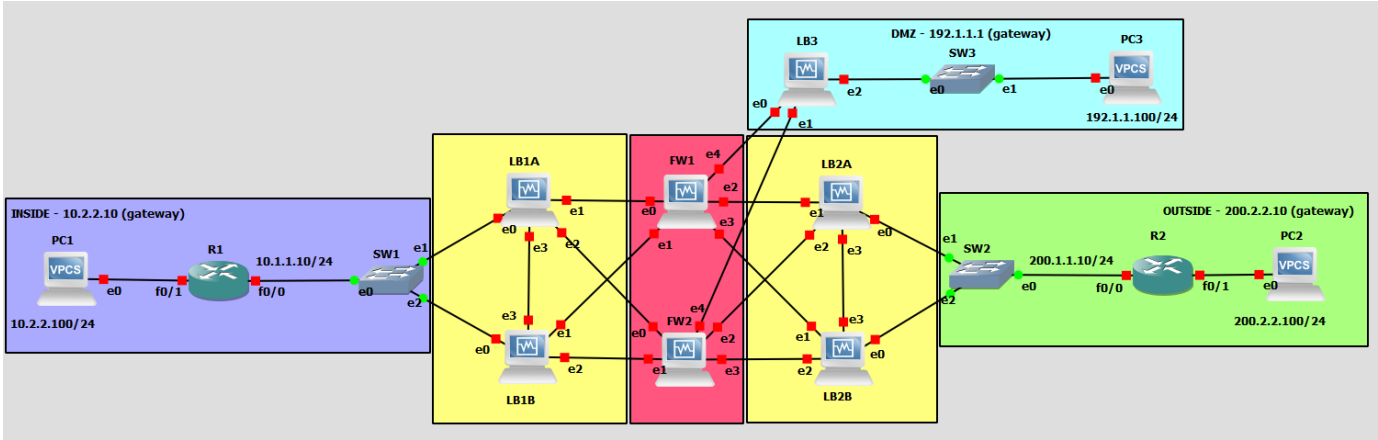


Fig. 1 - Topologia da rede

Configuração

Vamos começar por atribuir os endereços IP às interfaces dos routers e aos computadores de acordo com o enunciado.

PC1 (computador interno):

```
ip 10.2.2.100/24 10.2.2.10
save
```

PC2 (computador externo):

```
ip 200.2.2.100/24 200.2.2.10
save
```

R1 (*router* interno):

```
conf t
ip route 0.0.0.0 0.0.0.0 10.1.1.11 # LB1A
int f0/1
ip add 10.2.2.10 255.255.255.0
no shut
int f0/0
ip add 10.1.1.10 255.255.255.0
no shut
end
write
```

> Na rota estática apenas é necessário definir o *next-hop* para **um dos *load balancers***, uma vez que estes **estão sincronizados** (no entanto, por questões de redundância, é possível definir rotas para ambos os *load balancers*).

R2 (*router* externo):

```
conf t
ip route 0.0.0.0 0.0.0.0 200.1.1.12 # LB2B
int f0/1
ip add 200.2.2.10 255.255.255.0
no shut
int f0/0
ip add 200.1.1.10 255.255.255.0
no shut

end
write
```

LB1A (*load balancer* superior interno):

```
configure
set system host-name LB1A

# Interfaces
set interfaces ethernet eth0 address 10.1.1.11/24
set interfaces ethernet eth1 address 10.0.1.11/24
set interfaces ethernet eth2 address 10.0.6.1/24
set interfaces ethernet eth3 address 10.3.1.1/24

# Rotas Estáticas
set protocols static route 10.2.2.0/24 next-hop 10.1.1.10 # R1

# Load-Balancing
set load-balancing wan interface-health eth1 nexthop 10.0.1.12 # FW1
set load-balancing wan interface-health eth2 nexthop 10.0.6.2 # FW2
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 sticky-connections inbound
set load-balancing wan disable-source-nat

# VRRP
set high-availability vrrp group LBCluster1 vrid 10
set high-availability vrrp group LBCluster1 interface eth3
set high-availability vrrp group LBCluster1 virtual-address 192.168.100.1/24
set high-availability vrrp sync-group LBCluster1 member LBCluster1
set high-availability vrrp group LBCluster1 rfc3768-compatibility
```

```
# Conntrack-sync
set service conntrack-sync accept-protocol 'tcp,udp,icmp'
set service conntrack-sync failover-mechanism vrrp sync-group LBCluster1
set service conntrack-sync interface eth3
set service conntrack-sync mcast-group 225.0.0.50
set service conntrack-sync disable-external-cache

commit
save
```

> Por norma, em ataques DDoS, o objetivo do atacante é sobrecarregar o servidor com um grande volume de tráfego malicioso, tornando-o inacessível para os utilizadores legítimos. Uma das técnicas para mitigar este tipo de ataques é a distribuição de carga de tráfego, de modo a que o servidor não fique sobrecarregado.
>
> No entanto, a sincronização de estados entre os *load balancers* pode ser **prejudicial durante um ataque DDoS**, uma vez que aumenta o overhead de processamento, atrasa a deteção e mitigação do ataque, esgota os recursos e aumenta a complexidade da rede. Por isso, seria preferível o uso de algoritmos de *load balancing* que não requerem esta sincronização.
>
> Alguns exemplos de **algoritmos de *load balancing*** que não requerem a sincronização de estados entre os *load balancers* são:
>
> - **Round Robin**: Os *requests* são distribuídos sequencialmente pelos servidores, voltando ao primeiro servidor quando o final da lista é atingido.
> - **IP Hash**: Os *requests* são encaminhados para servidores com base num hash do endereço IP do cliente.
> - **Least Connections**: Os *requests* são encaminhados para o servidor com o menor número de conexões ativas.
> - **Random**: Os *requests* são encaminhados para um servidor aleatório, logo nenhum estado é mantido.

> O *virtual address* é partilhado entre os *load balancers* e tem como objetivo garantir a disponibilidade dos serviços, permitindo que os *load balancers* partilhem um único endereço IP virtual.

LB1B (*load balancer* inferior interno):

```
configure
set system host-name LB1B

# Interfaces
set interfaces ethernet eth0 address 10.1.1.12/24
set interfaces ethernet eth1 address 10.0.5.1/24
set interfaces ethernet eth2 address 10.0.2.12/24
set interfaces ethernet eth3 address 10.3.1.2/24

# Rotas Estáticas
set protocols static route 10.2.2.0/24 next-hop 10.1.1.10 # R1

# Load-Balancing
set load-balancing wan interface-health eth1 nexthop 10.0.5.2 # FW1
set load-balancing wan interface-health eth2 nexthop 10.0.2.13 # FW2
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 sticky-connections inbound
set load-balancing wan disable-source-nat

# VRRP
set high-availability vrrp group LBCluster1 vrid 10
set high-availability vrrp group LBCluster1 interface eth3
set high-availability vrrp group LBCluster1 virtual-address 192.168.100.1/24
set high-availability vrrp sync-group LBCluster1 member LBCluster1
set high-availability vrrp group LBCluster1 rfc3768-compatibility

# Conntrack-sync
```

```
set service conntrack-sync accept-protocol 'tcp,udp,icmp'
set service conntrack-sync failover-mechanism vrrp sync-group LBCluster1
set service conntrack-sync interface eth3
set service conntrack-sync mcast-group 225.0.0.50
set service conntrack-sync disable-external-cache

commit
save
```

> A sincronização de estados é feita através do `conntrack-sync`, que permite a sincronização de estados de conexão entre os dispositivos de *firewall*.

> O mecanismo utilizado é o *fail-over*, onde um dos *load balancers* é definido como o principal e o outro como secundário. O *load balancer* principal é responsável por encaminhar o tráfego para os servidores, enquanto o secundário fica em *standby*, pronto para assumir o controlo em caso de falha do principal, criando redundância.

LB2A (*load balancer* superior externo):

```
configure
set system host-name LB2A

# Interfaces
set interfaces ethernet eth0 address 200.1.1.11/24
set interfaces ethernet eth1 address 10.0.4.2/24
set interfaces ethernet eth2 address 10.0.8.2/24
set interfaces ethernet eth3 address 10.4.1.1/24

# Rotas Estáticas
set protocols static route 200.2.2.0/24 next-hop 200.1.1.10 # R2

# Load-Balancing
set load-balancing wan interface-health eth1 nexthop 10.0.4.1 # FW1
set load-balancing wan interface-health eth2 nexthop 10.0.8.1 # FW2
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 sticky-connections inbound
set load-balancing wan disable-source-nat

# VRRP
set high-availability vrrp group LBCluster2 vrid 10
set high-availability vrrp group LBCluster2 interface eth3
set high-availability vrrp group LBCluster2 virtual-address 192.168.100.2/24
set high-availability vrrp sync-group LBCluster2 member LBCluster2
set high-availability vrrp group LBCluster2 rfc3768-compatibility

# Conntrack-sync
set service conntrack-sync accept-protocol 'tcp,udp,icmp'
set service conntrack-sync failover-mechanism vrrp sync-group LBCluster2
set service conntrack-sync interface eth3
set service conntrack-sync mcast-group 225.0.0.50
set service conntrack-sync disable-external-cache

commit
save
```

> Ativando o *sticky sessions*, permite que os pedidos do cliente sejam sempre encaminhados pelo mesmo *load balancer*.

> Evitando que o *firewall* tenha de sincronizar estados entre os servidores.

LB2B (*load balancer* inferior externo):

```
configure
set system host-name LB2B

# Interfaces
set interfaces ethernet eth0 address 200.1.1.12/24
set interfaces ethernet eth1 address 10.0.7.2/24
set interfaces ethernet eth2 address 10.0.3.2/24
set interfaces ethernet eth3 address 10.4.1.2/24

# Rotas Estáticas
set protocols static route 200.2.2.0/24 next-hop 200.1.1.10 # R2

# Load-Balancing
set load-balancing wan interface-health eth1 nexthop 10.0.7.1 # FW1
set load-balancing wan interface-health eth2 nexthop 10.0.3.1 # FW2
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 sticky-connections inbound
set load-balancing wan disable-source-nat

# VRRP
set high-availability vrrp group LBCluster2 vrid 10
set high-availability vrrp group LBCluster2 interface eth3
set high-availability vrrp group LBCluster2 virtual-address 192.168.100.2/24
set high-availability vrrp sync-group LBCluster2 member LBCluster2
set high-availability vrrp group LBCluster2 rfc3768-compatibility

# Conntrack-sync
set service conntrack-sync accept-protocol 'tcp,udp,icmp'
set service conntrack-sync failover-mechanism vrrp sync-group LBCluster2
set service conntrack-sync interface eth3
set service conntrack-sync mcast-group 225.0.0.50
set service conntrack-sync disable-external-cache

commit
save
```

> Não é necessário a definição de rotas estáticas, pois o *next-hop* já está definido nas configurações de *load balancing*.

FW1 (*firewall* superior):

```
configure
set system host-name FW1

# Interfaces
set interfaces ethernet eth0 address 10.0.1.12/24
set interfaces ethernet eth1 address 10.0.5.2/24
set interfaces ethernet eth2 address 10.0.4.1/24
set interfaces ethernet eth3 address 10.0.7.1/24
set interfaces ethernet eth4 address 10.0.9.1/24

# Rotas Estáticas
set protocols static route 10.2.2.0/24 next-hop 10.0.1.11 # LB1A
set protocols static route 10.2.2.0/24 next-hop 10.0.5.1 # LB1B
set protocols static route 0.0.0.0/0 next-hop 10.0.4.2 # LB2A
set protocols static route 0.0.0.0/0 next-hop 10.0.7.2 # LB2B
set protocols static route 192.1.1.0/24 next-hop 10.0.9.2 # LB3
```

```
# NAT Translation
set nat source rule 10 outbound-interface eth2 # LB2A
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
set nat source rule 20 outbound-interface eth3 # LB2B
set nat source rule 20 source address 10.0.0.0/8
set nat source rule 20 translation address 192.1.0.11-192.1.0.21

# Zone Definition
(No Ponto 10)

# Zone Policy
(No Ponto 10)

commit
save
```

> Temos uma rota estática para cada *load balancer*, de forma a que caso um deles falhe, o tráfego seja encaminhado para o outro.

> O NAT (*Network Address Translation*) é uma técnica utilizada para traduzir endereços IP e portas de um endereço para outro, permitindo que os dispositivos de uma rede privada comuniquem com dispositivos de uma rede pública. O NAT é utilizado para proteger a rede interna de ataques externos, ocultando os endereços IP privados dos dispositivos internos e permitindo que estes comuniquem com a rede externa através de um único endereço IP público.
>
> É necessário definir uma regra por interface de saída para o NAT (*outbound-interface*), especificando o intervalo de endereços IP a serem traduzidos (*translation address*).

FW2 (*firewall* inferior):

```
configure
set system host-name FW2

# Interfaces
set interfaces ethernet eth0 address 10.0.6.2/24
set interfaces ethernet eth1 address 10.0.2.13/24
set interfaces ethernet eth2 address 10.0.8.1/24
set interfaces ethernet eth3 address 10.0.3.1/24
set interfaces ethernet eth4 address 10.0.10.1/24

# Rotas Estáticas
set protocols static route 10.2.2.0/24 next-hop 10.0.2.12 # LB1B
set protocols static route 10.2.2.0/24 next-hop 10.0.6.1 # LB1A
set protocols static route 0.0.0.0/0 next-hop 10.0.3.2 # LB2B
set protocols static route 0.0.0.0/0 next-hop 10.0.8.2 # LB2A
set protocols static route 192.1.1.0/24 next-hop 10.0.10.2 # LB3

# NAT Translation
set nat source rule 10 outbound-interface eth3 # LB2B
set nat source rule 10 source address 10.0.0.0/8
set nat source rule 10 translation address 192.1.0.22-192.1.0.32
set nat source rule 20 outbound-interface eth2 # LB2A
set nat source rule 20 source address 10.0.0.0/8
set nat source rule 20 translation address 192.1.0.33-192.1.0.43


# Zone Definition
(No Ponto 10)

# Zone Policy
(No Ponto 10)
```

```
commit
save
```

> É necessário definir uma *pool* de endereços IP para a tradução NAT diferentes para cada interface pois as *firewalls* não estão sincronizadas.

Testes de Funcionamento

Conectividade na rede

Para testar a conectividade entre os computadores internos e externos, foi utilizado o comando `ping 200.2.2.100 -P 17 -p 5001` que envia pacotes UDP para o computador externo.

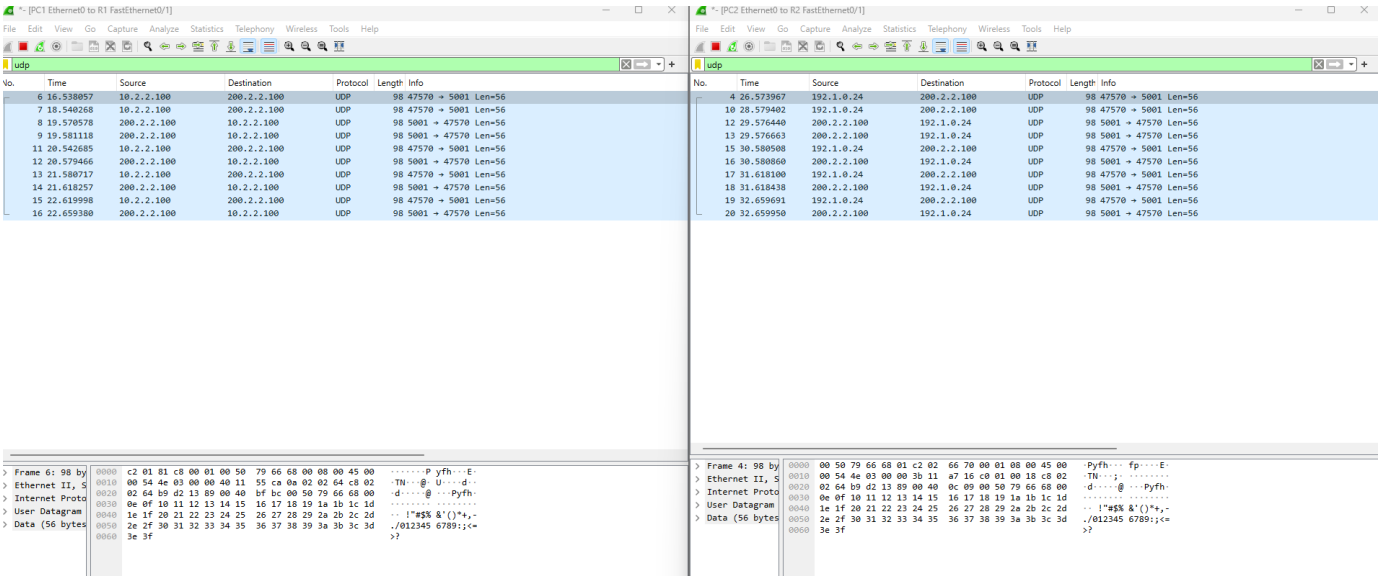


Fig. 2 - Captura Wireshark entre a rede interna e a rede externa com filtragem para o protocolo UDP.

Também foi feito o mesmo teste, mas com o protocolo ICMP, para verificar se a conexão é estabelecida corretamente (`ping 200.2.2.100`).

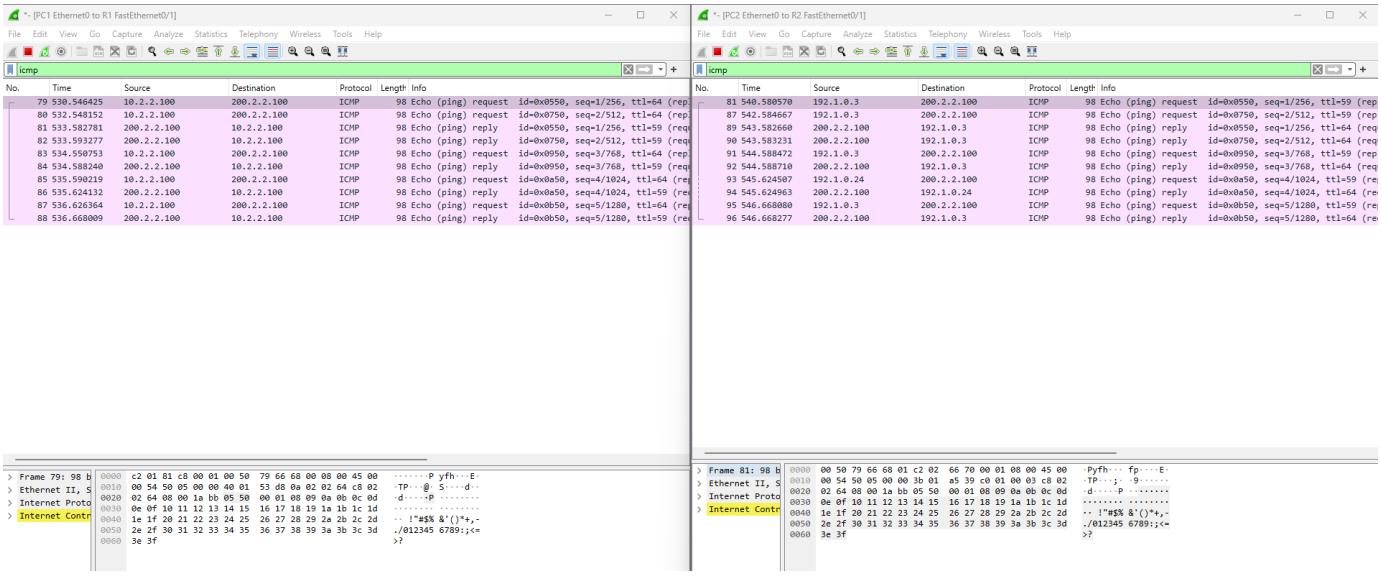


Fig. 3 - Captura Wireshark entre a rede interna e a rede externa com filtragem para o protocolo ICMP.

Verificação de rotas

As rotas de rede foram verificadas nos *routers* e nos *load balancers* para garantir que o tráfego é encaminhado corretamente para os destinos pretendidos.

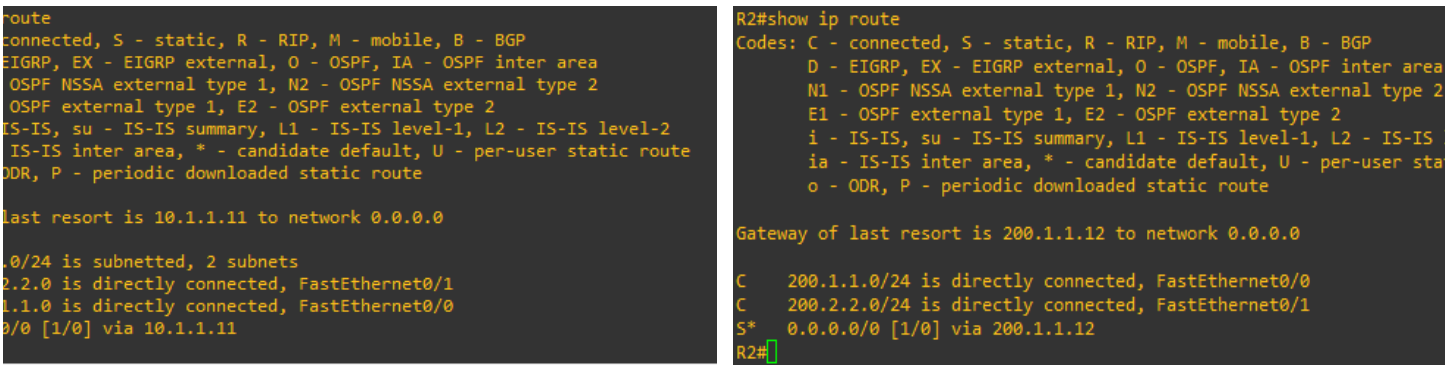


Fig. 4 - Tabela de rotas do R1 e R2

Fig. 5 - Tabela de rotas do LB1A e LB1B

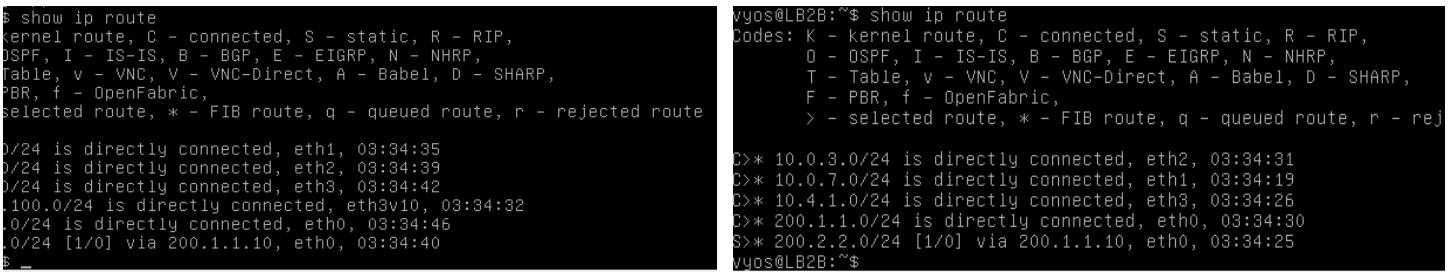


Fig. 6 - Tabela de rotas do LB2A e LB2B

Teste de tradução NAT

Como podemos verificar pelas tabelas de tradução NAT, os endereços IP dos computadores internos foram traduzidos para endereços IP públicos.

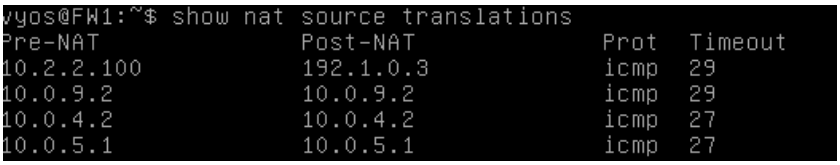


Fig. 7 - VyOS NAT Translation

Testes de sincronização dos *load balancers*

Usando o Wireshark na interface `eth3` entre os *load balancers* superior e inferior, foi possível verificar a troca de pacotes VRRP (Virtual Router Redundancy Protocol) que permite que os *load balancers* compartilhem um único endereço IP virtual (`192.168.100.1`).

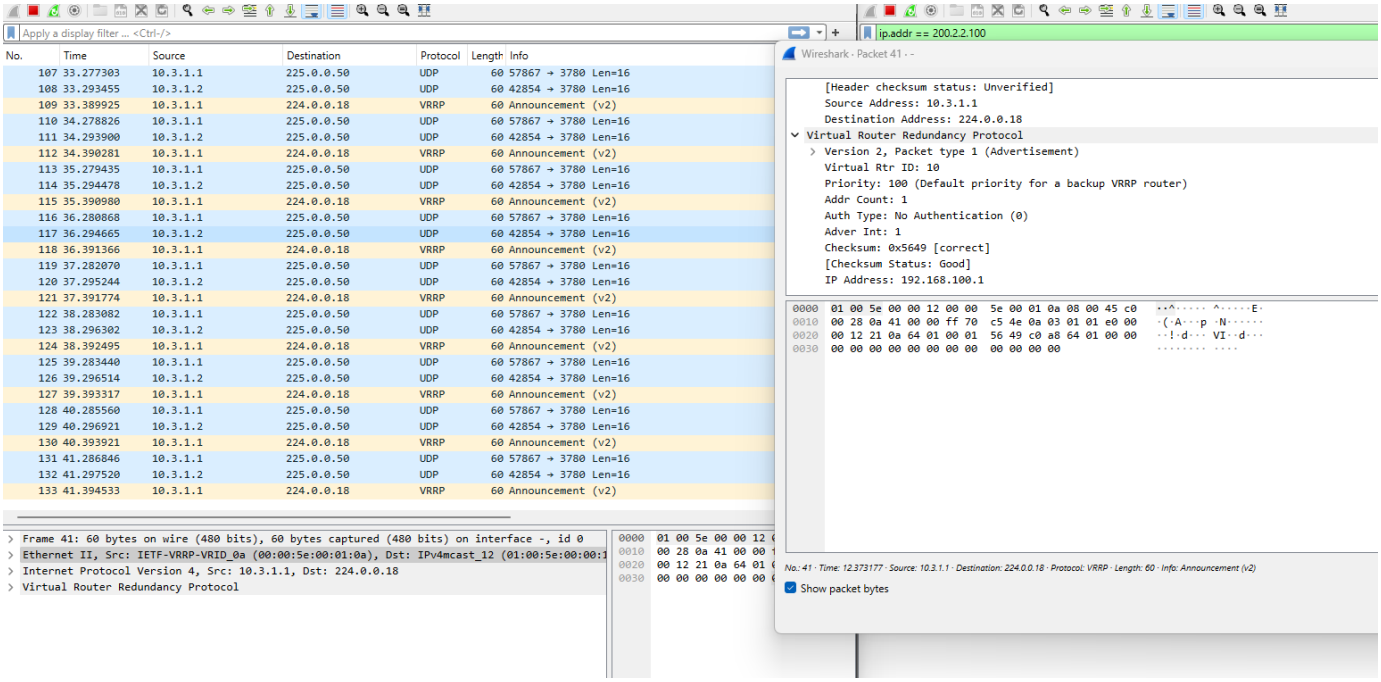


Fig. 8 - Captura Wireshark entre os Load Balancers 1A e 1B

De modo a analisar o seguimento das conexões sincronizadas entre os *load balancers*, utilizamos o comando `show conntrack table ipv4` para visualizar as conexões ativas.

```
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         TIMEOU
T
2560247854 10.3.1.2:42854        225.0.0.50:3780       udp [17]         29

vyos@LB1A:~$
```

Fig. 8 - Tabela de conexões ativas

> Este comando também serve para confirmar que o conntrack encontra-se habilitado e a funcionar corretamente.

Além disso, foi utilizado o comando `show conntrack-sync statistics` que mostra quais as conexões *conntrack* que encontram-se ativas entre os *load balancers*.

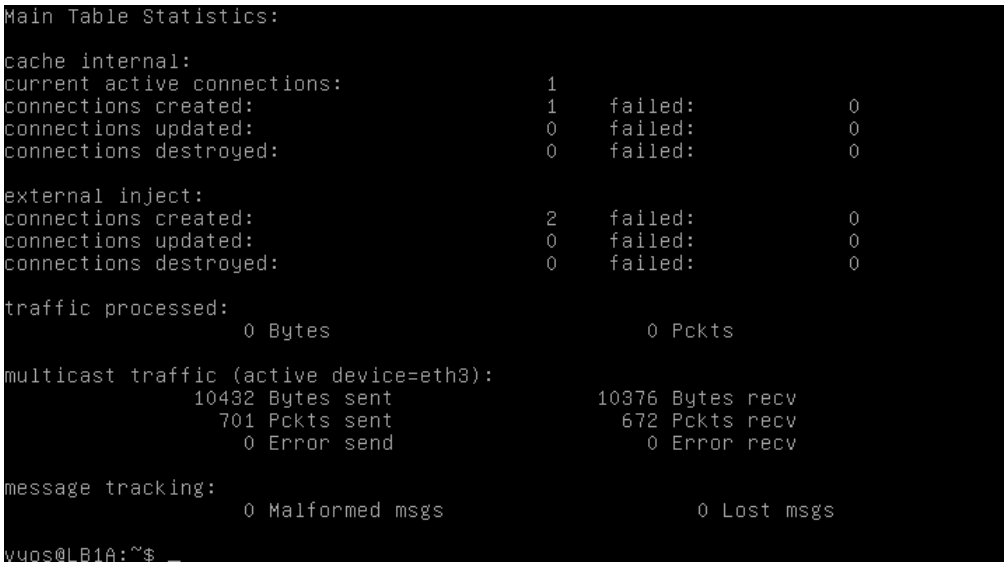


Fig. 9 - Conexões conntrack ativas do Load Balancer 1A

Podemos também verificar através do comando `ip route` que existe uma rota virtual configurada para o *load balancer* superior e inferior (`eth3v0`). Esta rota é utilizada para encaminhar o tráfego para o *load balancer* principal, que por sua vez encaminha o tráfego para os servidores.

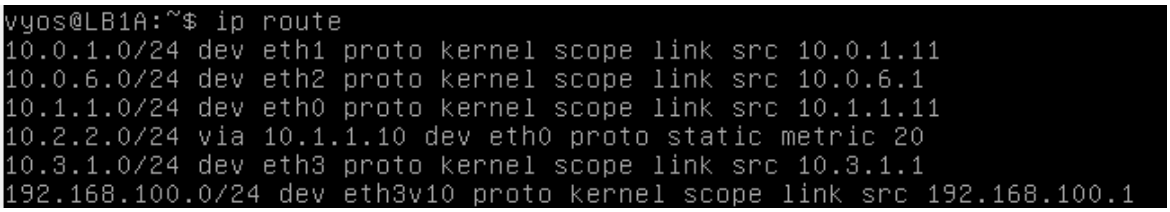


Fig. 10 - Rotas do Load Balancer 1A

E por último foi utilizado o comando `show conntrack-sync internal-cache` que mostra as conexões *conntrack* que encontram-se armazenas na cache do *load balancer*.

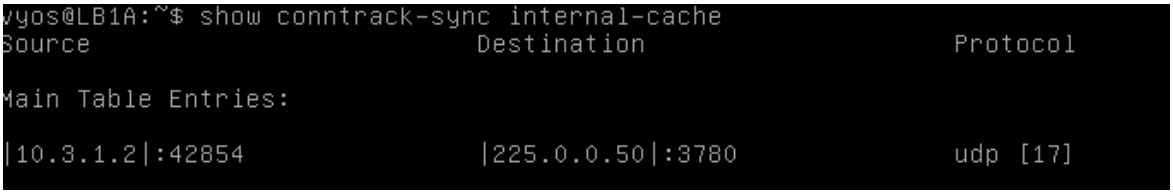


Fig. 11 - Cache de conexões conntrack do Load Balancer 1A

> Estas conexões são armazenadas na cache por motivos de performance, de modo a que seja reduzido o acesso a informação por dispositivos externos.

Testes de distribuição de carga

Como podemos verificar através do comando `sudo iptables -vL -t mangle`, o tráfego é distribuído de forma equitativa (mais ou menos) entre as portas do *load balancer*.

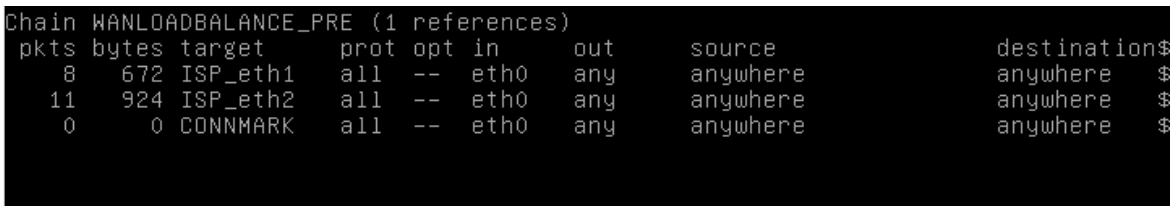


Fig. 12 - Distribuição de carga entre os Load Balancers 1A

Questões finais

1. Explain why the synchronization of the load-balancers allows the nonexistence of firewall synchronization.

R: A sincronização feita nos load balancers permite que os pedidos do cliente atinjam sempre o mesmo servidor, evitando que o firewall tenha de sincronizar estados entre os servidores. Isto é feito através do conceito de *sticky sessions*, que permite que os pedidos do cliente sejam sempre encaminhados para o mesmo servidor, evitando que o firewall tenha de sincronizar estados entre os servidores.

2. Which load balancing algorithm may also allow the nonexistence of load-balancers synchronization?

R: Um algoritmo load balancing que não requer a sincronização de estados entre os load balancers é o algoritmo *IP Hash*. Neste algoritmo, os pedidos são encaminhados para servidores com base num hash do endereço IP do cliente. Deste modo, o pedido do cliente é sempre encaminhado para o mesmo servidor, evitando que o firewall tenha de sincronizar estados entre os servidores.

3. Explain why device/connection states synchronization may be detrimental during a DDoS attack.

R: Durante um ataque DDoS, a sincronização de estados nos load balancers pode ser prejudicial devido ao aumento do overhead de processamento, atrasos na deteção e mitigação do ataque, esgotamento de recursos e aumento da complexidade da rede. Isso pode comprometer a capacidade dos load balancers de lidar eficazmente com o grande volume de tráfego malicioso, colocando em risco a disponibilidade dos serviços.

Ponto 10

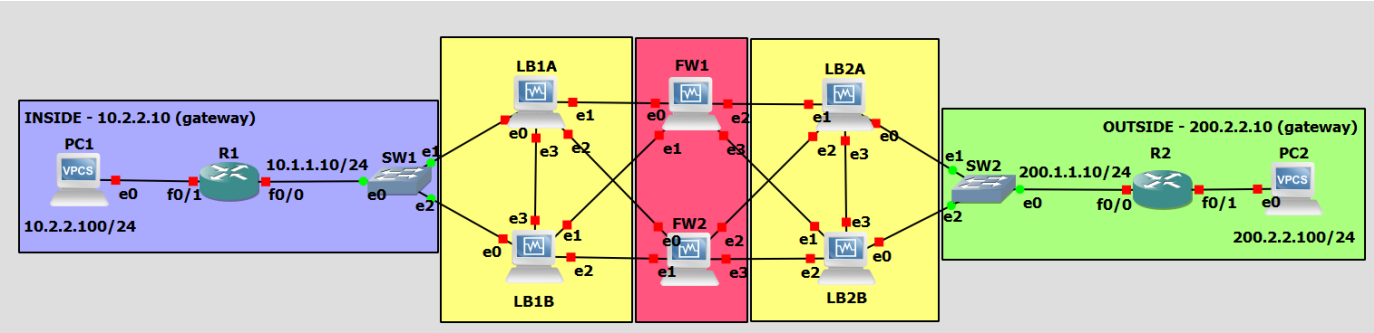


Fig. 13 - Topologia da rede com a DMZ

Servidor DMZ:

```
ip 192.1.1.100/24 192.1.1.1
save
```

> A DMZ (*Demilitarized Zone*) é uma sub-rede isolada que separa a rede interna da rede externa, permitindo que os servidores públicos sejam acessíveis a partir da Internet, mas não diretamente da rede interna.

LB3 (*load balancer* DMZ):

```
configure
set system host-name LB3

# Interfaces
set interfaces ethernet eth0 address 10.0.9.2/24
set interfaces ethernet eth1 address 10.0.10.2/24
set interfaces ethernet eth2 address 192.1.1.1/24

# Load-Balancing
set load-balancing wan interface-health eth0 nexthop 10.0.9.1
set load-balancing wan interface-health eth1 nexthop 10.0.10.1
set load-balancing wan rule 1 inbound-interface eth2
```

```
set load-balancing wan rule 1 interface eth0 weight 1
set load-balancing wan rule 1 interface eth1 weight 1
set load-balancing wan sticky-connections inbound
set load-balancing wan disable-source-nat

commit
save
```

> Neste caso foi utilizado apenas um *load balancer* para a DMZ, uma vez que a redundância já é garantida pelos *load balancers* superiores e inferiores.

Definição de Zonas

Para definir as zonas de segurança, foram criadas as seguintes zonas nas *firewalls* `FW1` e `FW2`:

```
set zone-policy zone INSIDE description "Inside (Internal Network)"
set zone-policy zone INSIDE interface eth0
set zone-policy zone INSIDE interface eth1
set zone-policy zone OUTSIDE description "Outside (External Network)"
set zone-policy zone OUTSIDE interface eth2
set zone-policy zone OUTSIDE interface eth3
set zone-policy zone DMZ description "DMZ (Server Farm)"
set zone-policy zone DMZ interface eth4
```

> Após inserir estas configurações, a conectividade é afetada, uma vez que o tráfego entre as zonas é bloqueado por padrão. Para permitir o tráfego entre as zonas, é necessário definir regras de controlo de tráfego.

Descrição da Configuração

Para limitar o acesso à rede, as seguintes ACLs foram implementadas nas *firewalls*:

```
set firewall name CONTROLLED default-action drop
set firewall name ESTABLISHED default-action drop
```

> A lista de acessos `CONTROLLED` é utilizada para definir as regras de controlo de tráfego (o que pode ou não passar), enquanto a lista `ESTABLISHED` é utilizada para definir as regras de tráfego já estabelecido.
>
> **Por padrão**, todo o tráfego é bloqueado, sendo necessário definir regras para permitir o tráfego entre as zonas.

Regras entre Zonas

Foram escolhidas um conjunto de regras para implementar nas firewalls, de forma a garantir a segurança e a integridade da rede mitigando ataques comuns à rede (e.g SYN Flood na regra `6`, DDoS na regra `4`).

Lista de regras entre zonas:

1. Permitir tráfego de saída do **INSIDE** dos seguintes protocolos: TCP, UDP, ICMP, HTTP, DNS e HTTPS;
2. Permitir tráfego já estabelecido pelo **INSIDE**;
3. Bloquear qualquer tráfego de saída do **OUTSIDE** para os endereços IP privados (ip privado: 10.2.2.0/24);
4. Limitar o tráfego de rede para o servidor **DMZ** (porta 4 das FWs) para 25 Mbps;
5. Permitir acesso ao servidor **DMZ** apenas em horário laboral (9h-18h);
6. Limitar o envio de pacotes SYN para 100 por segundo;
7. Limitar o acesso ao **DMZ** por protocolo SSH apenas ao administrador da rede (ip 10.2.2.200).

Nas *firewalls* `FW1` e `FW2`, as regras assim definidas:

```
configure

# Regra 1 - Definição de conexões permitidas
set firewall name CONTROLLED rule 11 description "Accept HTTP" # HTTP traffic
set firewall name CONTROLLED rule 11 action accept
set firewall name CONTROLLED rule 11 protocol tcp
set firewall name CONTROLLED rule 11 destination address 0.0.0.0/0
set firewall name CONTROLLED rule 11 destination port 80

set firewall name CONTROLLED rule 12 description "Accept HTTPS" # HTTPS traffic
set firewall name CONTROLLED rule 12 action accept
set firewall name CONTROLLED rule 12 protocol tcp
set firewall name CONTROLLED rule 12 destination address 0.0.0.0/0
set firewall name CONTROLLED rule 12 destination port 443

set firewall name CONTROLLED rule 13 description "Accept UDP" # UDP traffic
set firewall name CONTROLLED rule 13 action accept
set firewall name CONTROLLED rule 13 protocol udp
set firewall name CONTROLLED rule 13 destination address 0.0.0.0/0

set firewall name CONTROLLED rule 14 description "Accept ICMP" # ICMP traffic
set firewall name CONTROLLED rule 14 action accept
set firewall name CONTROLLED rule 14 protocol icmp
set firewall name CONTROLLED rule 14 destination address 0.0.0.0/0

set firewall name CONTROLLED rule 15 description "Accept DNS TCP/UDP" # DNS traffic
set firewall name CONTROLLED rule 15 action accept
set firewall name CONTROLLED rule 15 protocol tcp_udp
set firewall name CONTROLLED rule 15 destination address 0.0.0.0/0
set firewall name CONTROLLED rule 15 destination port 53

# Regra 2 - Conexões já estabelecidas
set firewall name ESTABLISHED rule 20 description "Accept Established-Related Connections"
set firewall name ESTABLISHED rule 20 action accept
set firewall name ESTABLISHED rule 20 state established enable
set firewall name ESTABLISHED rule 20 state related enable

# Regra 3 - Bloqueio p/ endereços privados
set firewall name CONTROLLED rule 30 description "Block Private IP Addresses"
set firewall name CONTROLLED rule 30 action drop
set firewall name CONTROLLED rule 30 source address 0.0.0.0/0
set firewall name CONTROLLED rule 30 destination address 10.2.2.0/24 # Private IP Address

# Regra 4 - Rate limiting
set traffic-policy shaper RATE-LIMIT bandwidth 25mbit
set traffic-policy shaper RATE-LIMIT default bandwidth 100%
set interfaces ethernet eth4 traffic-policy out RATE-LIMIT

# Regra 5 - Horário laboral p/ servidor DMZ
set firewall name CONTROLLED rule 50 description "Allow DMZ Access During Business Hours"
set firewall name CONTROLLED rule 50 action accept
set firewall name CONTROLLED rule 50 state new enable
set firewall name CONTROLLED rule 50 time starttime 09:00:00
set firewall name CONTROLLED rule 50 time stoptime 18:00:00
set firewall name CONTROLLED rule 50 time utc
set firewall name CONTROLLED rule 50 destination address 192.1.1.0/24

# Regra 6 - Mitigação de SYN Flood
set firewall name CONTROLLED rule 60 description "SYN Flood Protection"
set firewall name CONTROLLED rule 60 action drop
set firewall name CONTROLLED rule 60 protocol tcp
```

```
set firewall name CONTROLLED rule 60 state new enable
set firewall name CONTROLLED rule 60 tcp flags 'SYN'
set firewall name CONTROLLED rule 60 limit rate 100/second

# Regra 7 - Acesso SSH ao servidor DMZ
set firewall name CONTROLLED rule 70 description "Allow SSH Access to DMZ from Admin"
set firewall name CONTROLLED rule 70 action accept
set firewall name CONTROLLED rule 70 protocol tcp
set firewall name CONTROLLED rule 70 source address 10.2.2.200 # Admin IP
set firewall name CONTROLLED rule 70 destination address 192.1.1.0/24
set firewall name CONTROLLED rule 70 destination port 22

commit
save
```

> Ataques SYN Flood (inundação de *SYN requests* sem receção do *ACK request*) ainda são comuns e uma abordagem eficaz para mitigar este tipo de ataque é limitar o número de pacotes SYN que um host pode enviar por segundo, referido na regra `6`.

Aplicação das Regras

Nas *firewalls* `FW1` e `FW2`, as regras foram aplicadas da seguinte forma:

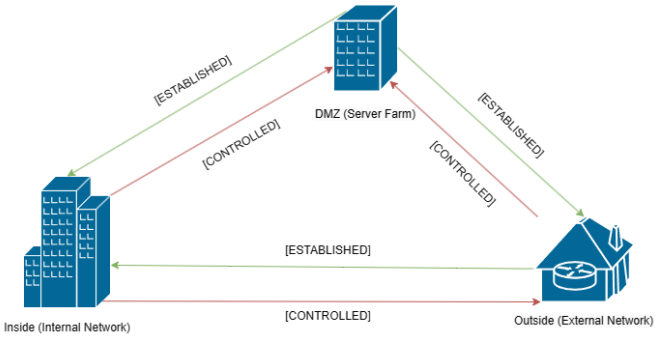


Fig. 14 - Diagrama de aplicação das ACLs

```
# p/ DMZ
set zone-policy zone DMZ from INSIDE firewall name CONTROLLED
set zone-policy zone DMZ from OUTSIDE firewall name CONTROLLED

# p/ OUTSIDE
set zone-policy zone OUTSIDE from INSIDE firewall name CONTROLLED
set zone-policy zone OUTSIDE from DMZ firewall name ESTABLISHED

# p/ INSIDE
set zone-policy zone INSIDE from OUTSIDE firewall name ESTABLISHED
set zone-policy zone INSIDE from DMZ firewall name ESTABLISHED
```

> Ou seja, todo o tráfego para o exterior e para o DMZ (por conta de ser uma zona isolada) é filtrado, enquanto o tráfego restante é estabelecido por regras já existentes.

Testes de Funcionamento

Teste de Conectividade entre Zonas

Para testar a conectividade entre as zonas, foram enviados pacotes ICMP do computador interno para o externo e vice-versa.

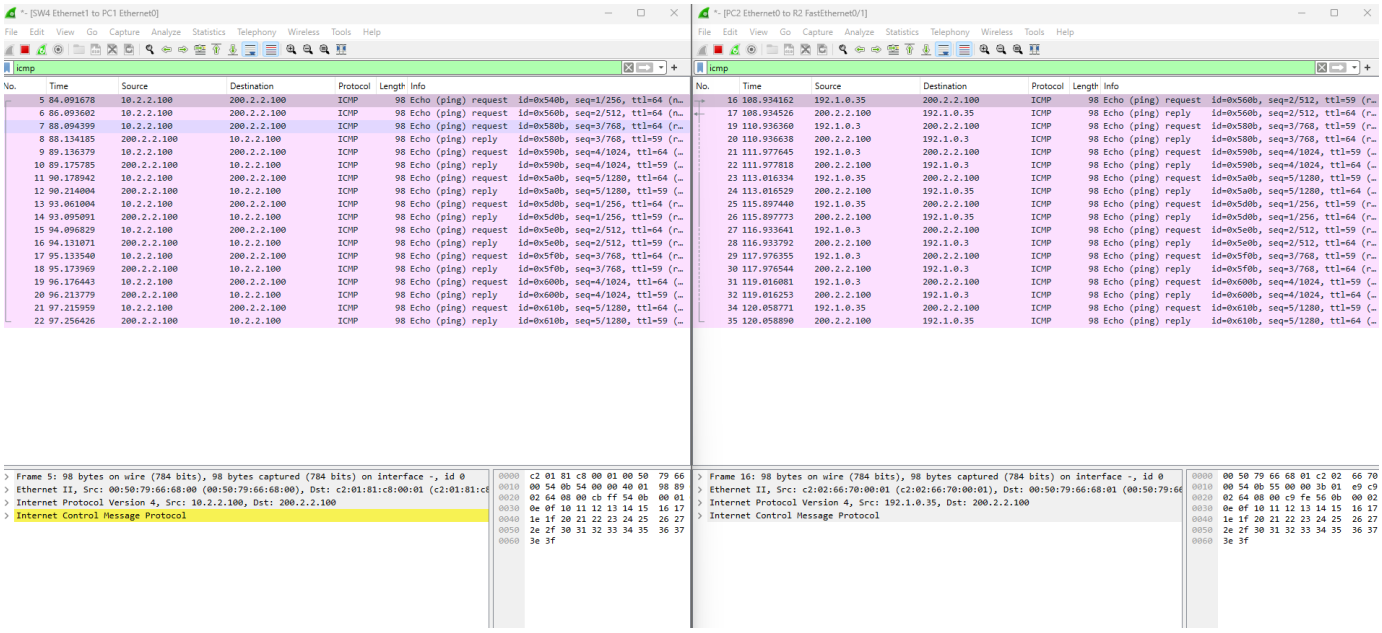


Fig. 15 - Captura de ecrã do teste de conectividade entre INSIDE-OUTSIDE

> Deste modo, podemos verificar que os pedidos provenientes do INSIDE não são bloqueados pela firewall, uma vez que a regra 1 permite o tráfego de saída do INSIDE para o OUTSIDE através do protocolo ICMP.

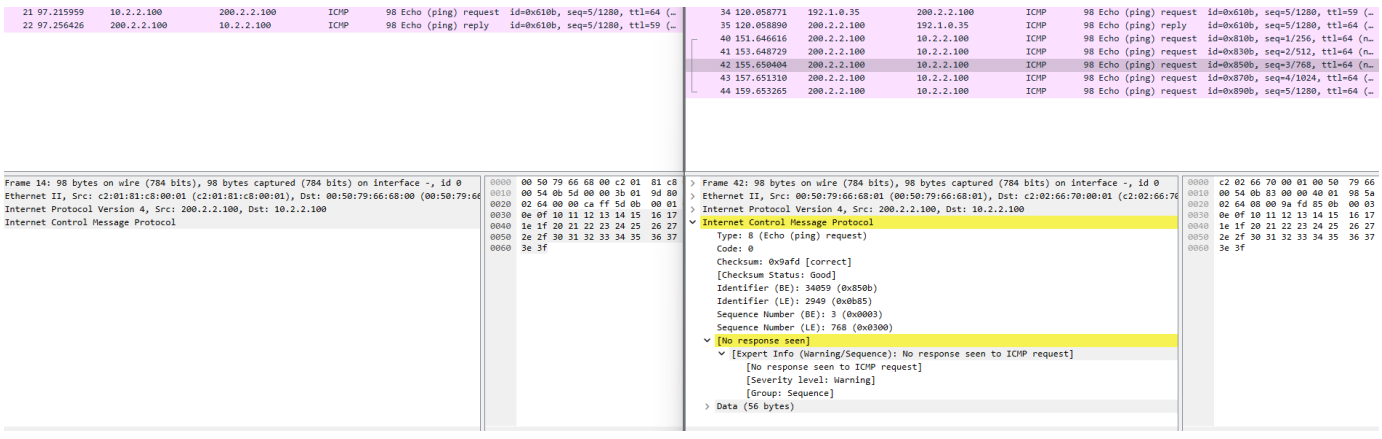


Fig. 16 - Captura de ecrã do teste de conectividade entre OUTSIDE-INSIDE

> Como podemos verificar, os pedidos provenientes do OUTSIDE são bloqueados pela firewall, uma vez que a regra 3 bloqueia o tráfego de saída do OUTSIDE para os endereços IP privados.

```
IPv4 Firewall "ESTABLISHED":

 Active on traffic to -
  zone [INSIDE] from zones [DMZ, OUTSIDE]
  zone [OUTSIDE] from zone [DMZ]

rule  packets   bytes     action  source          destination
----  -------   -----     ------  ------          -----------
20    16        1.34K     ACCEPT  0.0.0.0/0       0.0.0.0/0
10000 2         168       DROP    0.0.0.0/0       0.0.0.0/0

vuos@FW2:~$
```

Fig. 17 - Log da firewall

> Através do comando `show log firewall name ESTABLISHED`, podemos verificar que a regra de tráfego `ESTABLISHED` está a ser aplicada corretamente e todos os pacotes que não foram estabelecidos pela rede interna são bloqueados.

Verificação de regra de tráfego (`SYN Flood Protection`)

De modo a verificar se a regra de tráfego `SYN Flood Protection` está a ser aplicada corretamente, foi adicionado uma máquina virtual com a distribuição Ubuntu ao INSIDE da rede (`10.2.2.200`), onde a mesma irá enviar pacotes SYN para o servidor DMZ.

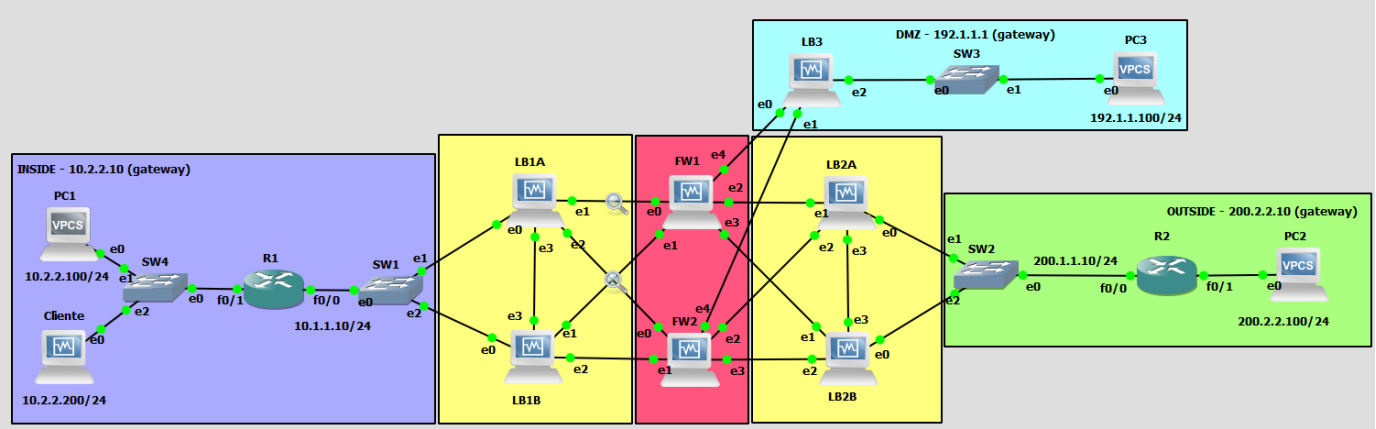


Fig. 18 - Topologia com máquina virtual linux

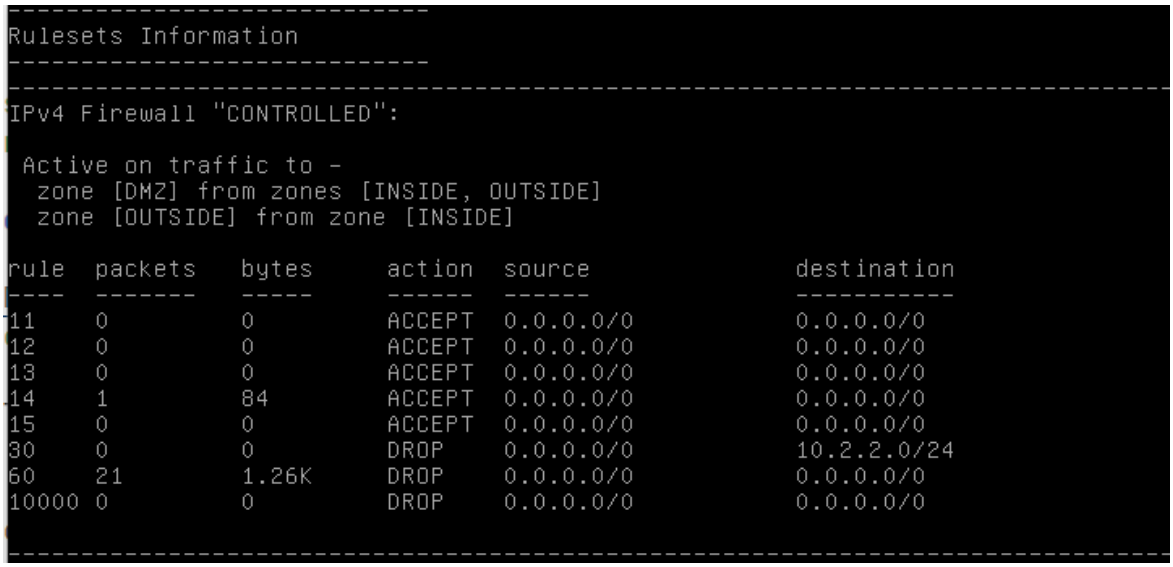
Através do comando `iperf -c 192.1.1.100 -b 26m`, foi possível enviar pacotes SYN para o servidor DMZ.

```
simao@simao-vm:~$ iperf -c 192.1.1.100 -b 26m -o test.txt
Output from stdout and stderr will be redirected to file test.txt
simao@simao-vm:~$ cat test.txt
tcp connect failed: Connection timed out
------------------
Client connecting to 192.1.1.100, TCP port 5001
TCP window size: -1.00 Byte (default)
------------------------------------------------------------
[  1] local 0.0.0.0 port 0 connected with 192.1.1.100 port 5001
```

Fig. 19 - Envio de pedidos TCP

> Os resultados foram guardados num ficheiro e exibidos através do comando `cat <ficheiro>`. É de observar que a conexão foi bloqueada após atingir o limite de pacotes SYN por segundo.

Para verificar se a regra de tráfego `SYN Flood Protection` está a ser aplicada corretamente, foi utilizado o comando `show log firewall name CONTROLLED` para exibir os logs de tráfego da firewall.



```
----------------------------
Rulesets Information
----------------------------
-------------------------------------------------------------------------------
IPv4 Firewall "CONTROLLED":

 Active on traffic to -
  zone [DMZ] from zones [INSIDE, OUTSIDE]
  zone [OUTSIDE] from zone [INSIDE]

rule  packets   bytes     action  source            destination
----  -------   -----     ------  ------            -----------
11    0         0         ACCEPT  0.0.0.0/0         0.0.0.0/0
12    0         0         ACCEPT  0.0.0.0/0         0.0.0.0/0
13    0         0         ACCEPT  0.0.0.0/0         0.0.0.0/0
14    1         84        ACCEPT  0.0.0.0/0         0.0.0.0/0
15    0         0         ACCEPT  0.0.0.0/0         0.0.0.0/0
30    0         0         DROP    0.0.0.0/0         10.2.2.0/24
60    21        1.26K     DROP    0.0.0.0/0         0.0.0.0/0
10000 0         0         DROP    0.0.0.0/0         0.0.0.0/0

-------------------------------------------------------------------------------
```

Fig. 20 - Logs da firewall VyOS

> Como podemos verificar, a regra de tráfego `SYN Flood Protection` está a ser aplicada corretamente.

Extra

Como medida para mitigar os ataques DDoS, foi criado um script que permite adicionar uma lista de endereços IP de atacantes a uma lista de bloqueio nas firewalls, automatizando o processo. O script utiliza a biblioteca `paramiko` para estabelecer uma conexão SSH com as firewalls e adicionar as regras de bloqueio.

```python
import paramiko

import re

def validate_ipv4(ip_address):
    # Regular expression pattern for IPv4 validation
    ipv4_pattern = r"^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$"

    # Check if the IP address matches the pattern
    if re.match(ipv4_pattern, ip_address):
        return True
    else:
        return False

def main(firewall : str, blocklist : str):
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

    username = "vyos"
    password = "vyos"
    hostname = firewall_ip_address
    port = 22 # SSH port

    try:
      with open(blocklist, "r") as file:
          ssh.connect(hostname, port, username, password)
          command = ""
          attackers = file.readlines()

          for attacker in attackers:
            if validate_ipv4(attacker) == False:
              print(f"Invalid IPv4 address: {attacker}")
              continue
```

```
            command = f"sudo iptables -A INPUT -s {attacker} -j DROP"

            stdin, stdout, stderr = ssh.exec_command(command)

            if stderr.channel.recv_exit_status() != 0: # Error
              print(f"Error occurred: {stderr.read().decode()}")
            else:
              print(f"Blocking rule added for {ip_address}")

          ssh.close()
      print("Blocking list added successfully to the " + firewall + " firewall.")
    except Exception as e:
      print(f"Error reading blocklist file: {e}")

if __name__ == "__main__":

    if len(sys.argv) < 3:
        print("Usage: python3 block_attackers.py <firewall_ip_address> <attackers_file>")
        sys.exit(1)

    firewall_ip = sys.argv[1]
    blocklist_filepath = sys.argv[2]

    if validate_ipv4(firewall_ip) == False:
      print("Invalid IPv4 address for the firewall.")

    main(firewall_ip, blocklist_filepath)
```

Exemplo de utilização:

```
$ python3 block_attackers.py 10.0.10.1 /etc/ddos_blocklist.txt
```

Conclusão

Em suma, a implementação de firewalls de alta disponibilidade desempenha um papel fundamental na salvaguarda da continuidade operacional e da segurança de uma infraestrutura de rede. A exploração do VyOS permitiu a análise de diversos cenários de configuração, visando maximizar a disponibilidade e a resiliência dos sistemas de segurança de rede. Ao configurar os quatro *Load Balancers*, com sincronização de estados, e distribuindo assim o tráfego de forma equilibrada, alcançámos a mitigação de vulnerabilidades e garantimos uma proteção contínua contra ameaças cibernéticas. Isto contribuiu para uma resposta mais eficaz e robusta da infraestrutura de segurança. Ao conectar as *Firewalls* aos *Load Balancers*, estabelecemos um ambiente de alta disponibilidade com redundância, onde cada firewall atua como um gateway seguro entre as zonas outside e inside da rede. Como aprendizagem adicional, destacamos a importância de testes de funcionamento abrangentes para validar a eficácia das configurações implementadas.