Segurança em Redes de Comunicações

Universidade de Aveiro

João Ferreira, Rui Campos

Segurança em Redes de Comunicações

Report 1

Universidade de Aveiro

João Ferreira, Rui Campos
(103625) joaop.ferreira@ua.pt, (103709) ruigabriel2@ua.pt

20 de abril de 2024

Índice

Chapter 1

Introduction

1.1 Initial Configuration

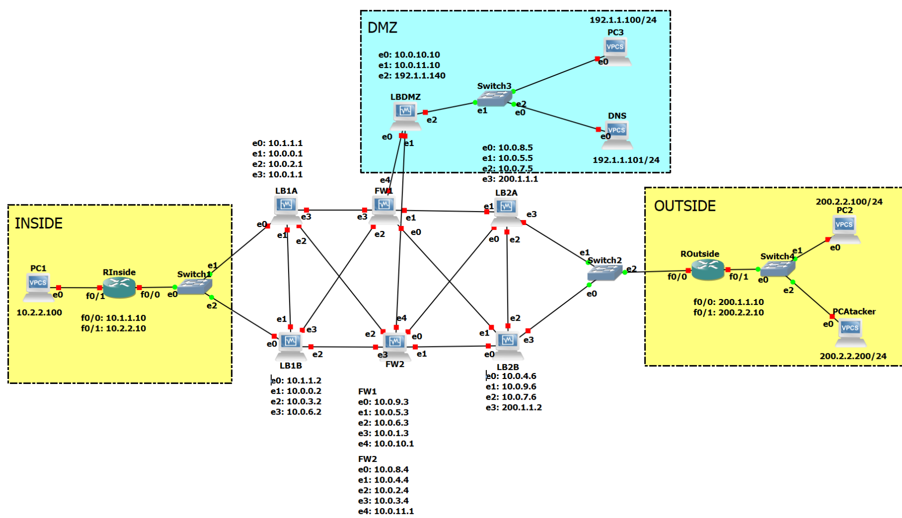


Fig.1: Network Configuration

In this project, we have designed and implemented a network infrastructure composed of three distinct zones: INSIDE, OUTSIDE, and DMZ, each serving specific purposes and requiring tailored security measures.

The INSIDE zone represents our internal network environment, housing a subnet with various terminals, including a VPC with the IP address 10.2.2.100/24.

On the other hand, the OUTSIDE zone consists of a separate subnet containing a VPC simulating an external device that we aim to access from our internal network (IP address 200.2.2.100).

To optimize the network's performance and reliability, we have implemented redundant load balancers (LB1A, LB1B, LB2A, LB2B and LBDMZ) responsible for distributing firewall load. These load balancers utilize conntrack-sync to synchronize their routing decisions, effectively distributing traffic load between the redundant firewalls (FW1 and FW2) without the need for constant firewall synchronization.

Besides that, we have a NAT mechanism implemented on those firewalls, in order to hide the internal topology of the network. The NAT pool comprises the subnet 192.1.0.0/24, which is subdivided into smaller subnets. Each of these smaller subnets is allocated to one of the "eth0"/"eth1" interfaces of FW1 and FW2.

The DMZ zone have a subnet dedicated to hosting server resources, including a DMZ-Server with two IP addresses. One IP is designated for a web service accessible both internally and externally. Additionally, a VPC has been instantiated to simulate a DNS server within the DMZ.

Finnaly, in our network setup, we introduced a PCAttacker node to simulate potential security threats and vulnerabilities. However, it's important to note that we did not develop any attack scripts or malicious activities for this node. Instead, we opted to keep it within the network topology to illustrate our awareness and consideration of potential attack scenarios. While the PCAttacker remains inactive, its presence underscores our commitment to evaluating and addressing security concerns within our network architecture. Through proactive measures and careful planning, we aim to fortify our defenses and ensure the integrity and resilience of our network infrastructure.

1.2 Load Balancing and State Synchronization

The network employs 5 VyOS instances labeled as LB1A, LB1B, LB2A, LB2B, and LBDMZ to manage load balancing tasks. These VyOS instances are configured to evenly distribute incoming traffic across two firewalls, FW1 and FW2. Additionally, by enabling sticky connections, the load balancer keeps track of connection states, ensuring that subsequent packets within the same connection are consistently routed through the same firewall. Consequently, the load balancer takes on the responsibility of maintaining connection state information, eliminating the need for synchronization between the firewalls.

```
vyos@LB1A:~$ show wan-load-balance status
Chain WANLOADBALANCE_PRE (1 references)
 pkts bytes target     prot opt in     out     source               destination

    2   168 ISP_eth2   all  --  eth0   *       0.0.0.0/0            0.0.0.0/0
         state NEW statistic mode random probability 0.50000000000
    2   168 ISP_eth3   all  --  eth0   *       0.0.0.0/0            0.0.0.0/0
         state NEW
    0     0 CONNMARK   all  --  eth0   *       0.0.0.0/0            0.0.0.0/0
         CONNMARK restore
vyos@LB1A:~$ show wan-load-balance connection
Type    State           Src                     Dst                     Packets
Bytes
udp                     10.2.2.100:17622        200.2.2.100:5125        2
168
vyos@LB1A:~$ show wan-load-balance connection
Type    State           Src                     Dst                     Packets
Bytes
udp                     10.2.2.100:17622        200.2.2.100:5125        4
336
```

Fig.2: Load Balancer 1A status during a ping from PC1 to PC2

```
vyos@LB1B:~$ show wan-load-balance status
Chain WANLOADBALANCE_PRE (1 references)
 pkts bytes target     prot opt in     out     source               destination

    0     0 ISP_eth2   all  --  eth0   *       0.0.0.0/0            0.0.0.0/0
         state NEW statistic mode random probability 0.50000000000
    0     0 ISP_eth3   all  --  eth0   *       0.0.0.0/0            0.0.0.0/0
         state NEW
    0     0 CONNMARK   all  --  eth0   *       0.0.0.0/0            0.0.0.0/0
         CONNMARK restore
vyos@LB1B:~$ show wan-load-balance connection
Type    State           Src                     Dst                     Packets
Bytes
udp                     10.2.2.100:19241        200.2.2.100:5125        0
0
udp                     10.2.2.100:17163        200.2.2.100:5125        0
0
```

Fig.3: Load Balancer 1B status during a ping from PC1 to PC2

```
vyos@LB2A:~$ show wan-load-balance status
Chain WANLOADBALANCE_PRE (1 references)
 pkts bytes target     prot opt in     out     source               destination

    0     0 ISP_eth0   all  --  eth3   *       0.0.0.0/0            0.0.0.0/0
         state NEW statistic mode random probability 0.50000000000
    0     0 ISP_eth1   all  --  eth3   *       0.0.0.0/0            0.0.0.0/0
         state NEW
   14   784 CONNMARK   all  --  eth3   *       0.0.0.0/0            0.0.0.0/0
         CONNMARK restore
vyos@LB2A:~$ show wan-load-balance connection
Type    State           Src                     Dst                     Packets
Bytes
udp                     192.1.0.15:14503        200.2.2.100:5125        2
168
udp                     192.1.0.143:14503       200.2.2.100:5125        0
0
vyos@LB2A:~$ show wan-load-balance connection
Type    State           Src                     Dst                     Packets
Bytes
udp                     192.1.0.15:14503        200.2.2.100:5125        3
252
udp                     192.1.0.143:14503       200.2.2.100:5125        0
0
```

Fig.4: Load Balancer 2A status during a ping from PC1 to PC2

```
vyos@LB2B:~$ show wan-load-balance status
Chain WANLOADBALANCE_PRE (1 references)
 pkts bytes target     prot opt in     out     source               destination

    0     0 ISP_eth0   all  --  eth3   *       0.0.0.0/0            0.0.0.0/0
         state NEW statistic mode random probability 0.50000000000
    0     0 ISP_eth1   all  --  eth3   *       0.0.0.0/0            0.0.0.0/0
         state NEW
   12   672 CONNMARK   all  --  eth3   *       0.0.0.0/0            0.0.0.0/0
         CONNMARK restore
vyos@LB2B:~$ show wan-load-balance connection
Type    State           Src                     Dst                     Packets
Bytes
udp                     192.1.0.143:28458       200.2.2.100:5125        1
84
udp                     192.1.0.15:28458        200.2.2.100:5125        0
0
vyos@LB2B:~$ show wan-load-balance connection
Type    State           Src                     Dst                     Packets
Bytes
udp                     192.1.0.143:28458       200.2.2.100:5125        1
84
udp                     192.1.0.15:28458        200.2.2.100:5125        0
0
```

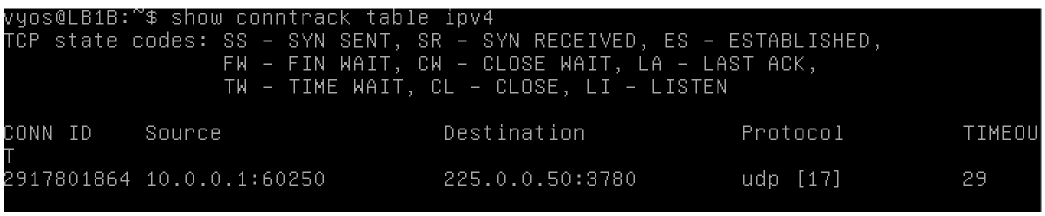
Fig.5: Load Balancer 2B status during a ping from PC1 to PC2

Moreover, conntrack-sync is enabled on the eth1 interface of LB1 and on the eth2 interface of LB2, while load balancing functionality is active on eth2 and eth3 interfaces of LB1A and LB1A and on eth0 and eth1 interfaces of LB2A and LB2B.

In the provided screenshots, upon running the command "show conntrack table ipv4" it becomes evident that both LB1A and LB1B, as well as LB2A and LB2B, are effectively tracking and synchronizing their current connections.

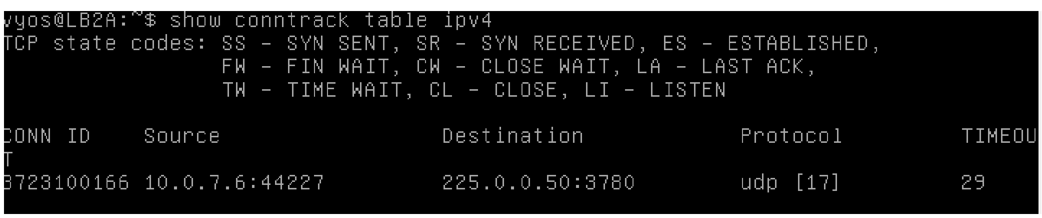
```
vyos@LB1A:~$ show conntrack table ipv4
TCP state codes: SS - SYN SENT, SR - SYN RECEIVED, ES - ESTABLISHED,
                 FW - FIN WAIT, CW - CLOSE WAIT, LA - LAST ACK,
                 TW - TIME WAIT, CL - CLOSE, LI - LISTEN

CONN ID    Source                Destination           Protocol        TIMEOU
T
323269264  10.0.0.2:39162        225.0.0.50:3780       udp [17]        29

2306411467 10.0.0.2              224.0.0.18            vrrp [112]      599
```

Fig.6: Current Load-Balancer 1A IPv4 Connection Tracking Synchronization



```
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        TIMEOU
T
2917801864 10.0.0.1:60250        225.0.0.50:3780       udp [17]        29
```

Fig.7: Current Load-Balancer 1B IPv4 Connection Tracking Synchronization



```
vyos@LB2A:~$ 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
3723100166 10.0.7.6:44227        225.0.0.50:3780       udp [17]        29
```

Fig.8: Current Load-Balancer 2A IPv4 Connection Tracking Synchronization

```
vyos@LB2B:~$ 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
637457362  10.0.7.5:53004        225.0.0.50:3780       udp [17]        29

1495041337 10.0.7.5              224.0.0.18            vrrp [112]      599
```

Fig.9: Current Load-Balancer 2B IPv4 Connection Tracking Synchronization

Chapter 2

Configuration

The network relies on static routes and Network Address Translation (NAT) to manage traffic flow in the absence of a routing protocol (like OSPF or BGP).

2.1 Routing with Static Routes

- **RInside**: Any traffic not matching specific routes on R1 gets sent to the "LB1A" and "LB1B" interfaces (part of the 10.1.1.0/24 network). This acts like a default route for RInside.
- **ROutside**: Traffic destined for the 192.1.0.0/24 network on ROutside is directed towards the "LB2A" and "LB2B" interfaces (part of the 200.1.1.0/24 network).
- **Load Balancers (LB1/LB2)**:

  They forward traffic like this:

  - LB1(A/B): Traffic going to the 10.2.2.0/24 network is sent to the R1 router interface with the IP address 10.1.1.10.
  - LB2(A/B): Traffic going to the 200.2.2.0/24 network is sent to the R2 router interface with the IP address 200.1.1.10.
- **Firewalls (FW1/FW2)**: Traffic destined for the 10.2.2.0/24 network is directed to one of the directly connected interfaces of the LB1(A/B) with the firewalls. Any other traffic is sent to one of the directly connected interfaces of the LB2(A/B) with the firewalls.

2.2 Policies and Rules

As depicted in the diagram, our network comprises three distinct zones: INSIDE, OUTSIDE, and DMZ. The INSIDE zone safeguards internal devices, while the OUTSIDE zone represents the broader internet. The DMZ serves as a semi-protected area where servers and services are exposed to external access.

To fortify our network's security, we've implemented stringent policies regulating communication between these zones. Two VyOS firewalls have been deployed to enforce these policies effectively.

At first, we imposed limitations on the connectivity between devices in the INSIDE and OUTSIDE zones using the FROM-INSIDE-TO-OUTSIDE firewall rule. This rule permitted traffic flow from the INSIDE zone to the OUTSIDE zone under specific conditions:

- Only communication utilizing the UDP protocol was permitted.
- Communication between zones was restricted to destination ports falling within the range of 5000 to 6000.

As we explained previously, one of the services implemented allows communication from the INSIDE network to the OUTSIDE network via UDP on ports 5000 to 6000. This service mirrors a scenario where devices within the INSIDE network need to engage in UDP-based communication with devices located in the OUTSIDE network. Notably, in this setup, communication initiation from the OUTSIDE network is restricted, as depicted in the next figure.

```
PC1> ping 200.2.2.100 -P 17 -p 5121 -s 5000
84 bytes from 200.2.2.100 udp_seq=1 ttl=59 time=40.901 ms
84 bytes from 200.2.2.100 udp_seq=2 ttl=59 time=37.869 ms
84 bytes from 200.2.2.100 udp_seq=3 ttl=59 time=35.103 ms
84 bytes from 200.2.2.100 udp_seq=4 ttl=59 time=34.885 ms
84 bytes from 200.2.2.100 udp_seq=5 ttl=59 time=36.026 ms

PC1> ping 200.2.2.100 -P 17 -p 8265 -s 5000
200.2.2.100 udp_seq=1 timeout
200.2.2.100 udp_seq=2 timeout
200.2.2.100 udp_seq=3 timeout
200.2.2.100 udp_seq=4 timeout
200.2.2.100 udp_seq=5 timeout
```

Fig.10: UDP Pings from PC1 to PC2

```
PC2> ping 10.2.2.100 -P 17 -p 5005
*200.2.2.10 udp_seq=1 ttl=255 time=9.145 ms (ICMP type:3, code:1, Destination host unreachable)
*200.2.2.10 udp_seq=2 ttl=255 time=7.967 ms (ICMP type:3, code:1, Destination host unreachable)
*200.2.2.10 udp_seq=3 ttl=255 time=7.926 ms (ICMP type:3, code:1, Destination host unreachable)
*200.2.2.10 udp_seq=4 ttl=255 time=8.927 ms (ICMP type:3, code:1, Destination host unreachable)
*200.2.2.10 udp_seq=5 ttl=255 time=7.904 ms (ICMP type:3, code:1, Destination host unreachable)
```

Fig.11: UDP Pings from PC2 to PC1

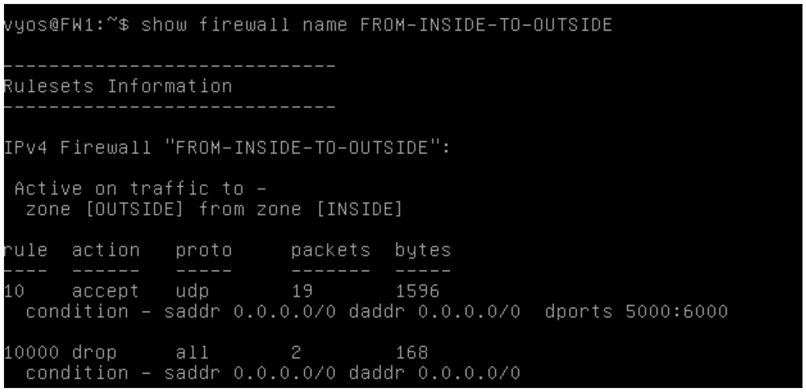


Fig.12: FW1 status during UDP Pings from PC1 to PC2

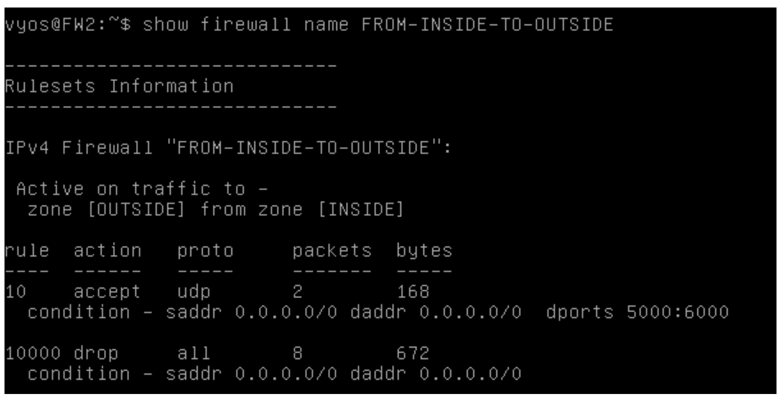


Fig.13: FW2 status during UDP Pings from PC1 to PC2

As illustrated in the image above, PC1 can establish connectivity with PC2 only if it adheres to the specified requirements. The initial ping, "ping 200.2.2.100 -P 17 -p 5121 -s 5000", traverses FW1 and is permitted to pass, while the subsequent ping, "ping 200.2.2.100 -P 17 -p 7777 -s 5555", traverses FW2 and is dropped due to failure to meet the requirements.

The DMZ zone hosts the following services:

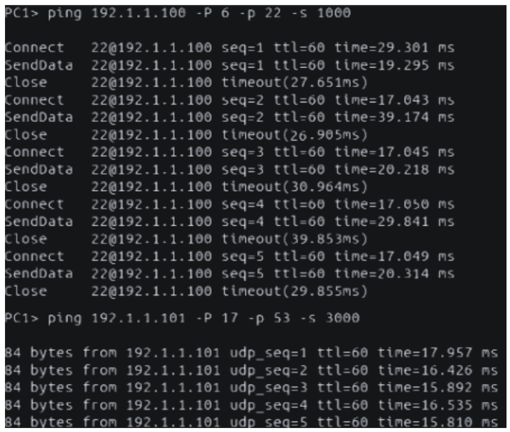
- Http Service on 192.1.1.100:80
- Https Service on 192.1.1.100:443
- SSH Service on 192.1.1.100:22
- DNS Service on 192.1.1.101:53

The criteria for accessing services in the DMZ are as follows:

- Destination port must be 22 (SSH), 80 (HTTP), or 443 (HTTPS), with TCP protocol, and destination address must be 192.1.1.100 (DMZ).

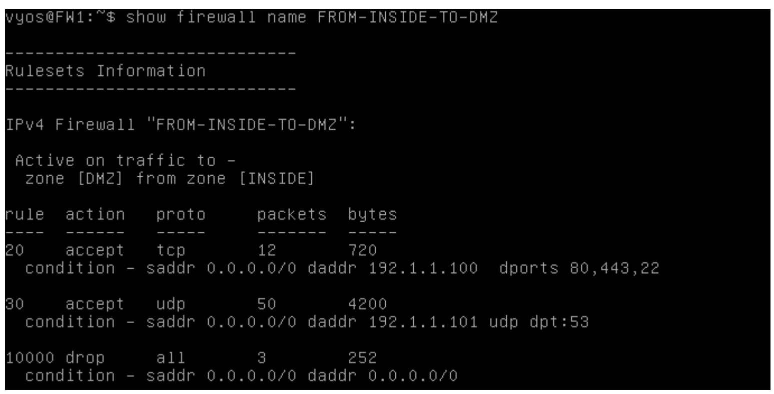
- For DNS Service, destination port must be 53, address must be 192.1.1.101 (DNS-Server), and UDP protocol must be used.

The screenshot below demonstrates the connectivity of the INSIDE zone (PC1) with some of the services available in the DMZ.



```
PC1> ping 192.1.1.100 -P 6 -p 22 -s 1000

Connect   22@192.1.1.100 seq=1 ttl=60 time=29.301 ms
SendData  22@192.1.1.100 seq=1 ttl=60 time=19.295 ms
Close     22@192.1.1.100 timeout(27.651ms)
Connect   22@192.1.1.100 seq=2 ttl=60 time=17.043 ms
SendData  22@192.1.1.100 seq=2 ttl=60 time=39.174 ms
Close     22@192.1.1.100 timeout(26.905ms)
Connect   22@192.1.1.100 seq=3 ttl=60 time=17.045 ms
SendData  22@192.1.1.100 seq=3 ttl=60 time=20.218 ms
Close     22@192.1.1.100 timeout(30.964ms)
Connect   22@192.1.1.100 seq=4 ttl=60 time=17.050 ms
SendData  22@192.1.1.100 seq=4 ttl=60 time=29.841 ms
Close     22@192.1.1.100 timeout(39.853ms)
Connect   22@192.1.1.100 seq=5 ttl=60 time=17.049 ms
SendData  22@192.1.1.100 seq=5 ttl=60 time=20.314 ms
Close     22@192.1.1.100 timeout(29.855ms)

PC1> ping 192.1.1.101 -P 17 -p 53 -s 3000

84 bytes from 192.1.1.101 udp_seq=1 ttl=60 time=17.957 ms
84 bytes from 192.1.1.101 udp_seq=2 ttl=60 time=16.426 ms
84 bytes from 192.1.1.101 udp_seq=3 ttl=60 time=15.892 ms
84 bytes from 192.1.1.101 udp_seq=4 ttl=60 time=16.535 ms
84 bytes from 192.1.1.101 udp_seq=5 ttl=60 time=15.810 ms
```

Fig.14: Ping from Inside to DMZ



```
vyos@FW1:~$ show firewall name FROM-INSIDE-TO-DMZ

-----------------------------
Rulesets Information
-----------------------------

IPv4 Firewall "FROM-INSIDE-TO-DMZ":

 Active on traffic to -
  zone [DMZ] from zone [INSIDE]

rule  action   proto     packets  bytes
----  ------   -----     -------  -----
20    accept   tcp       12       720
  condition - saddr 0.0.0.0/0 daddr 192.1.1.100  dports 80,443,22

30    accept   udp       50       4200
  condition - saddr 0.0.0.0/0 daddr 192.1.1.101 udp dpt:53

10000 drop     all       3        252
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0
```

Fig.15: FW1 status during a Ping from Inside to DMZ

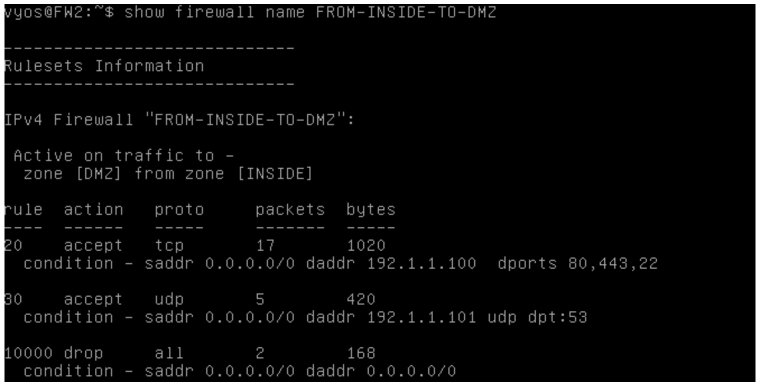


Fig.16: FW2 status during a Ping from Inside to DMZ

For external devices attempting to access services in the DMZ zone, all available services are accessible except for the SSH service. Consequently, the rules specified in the FROM-OUTSIDE-TO-DMZ firewall mirror those defined in the FROM-INSIDE-TO-DMZ firewall, with the sole exception of permitting access to port 22 for the SSH service.

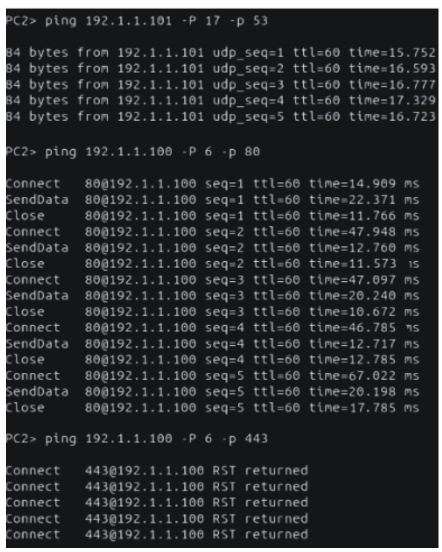
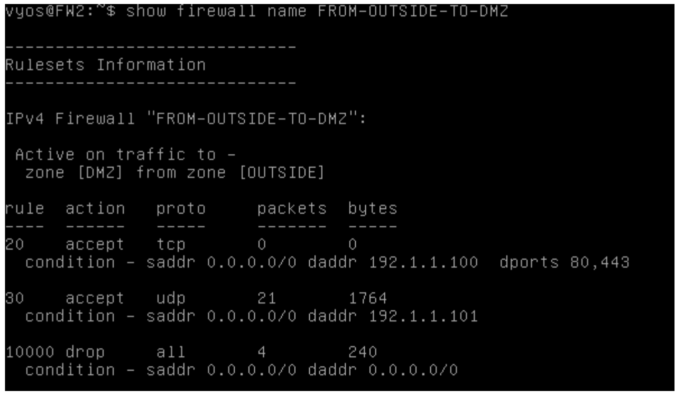
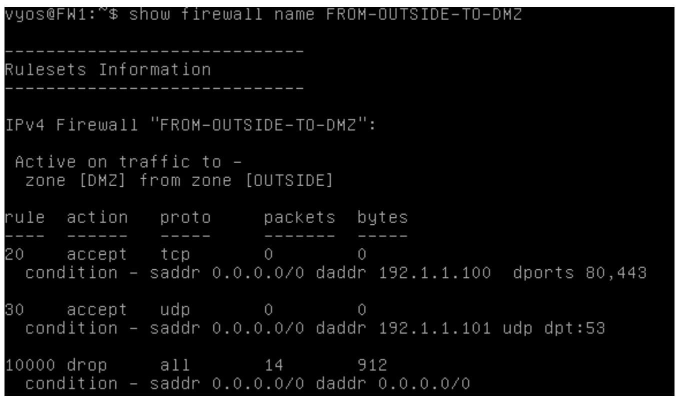


Fig.17: : Ping from Outside to DMZ



```
vyos@FW2:~$ show firewall name FROM-OUTSIDE-TO-DMZ

-----------------------------
Rulesets Information
-----------------------------

IPv4 Firewall "FROM-OUTSIDE-TO-DMZ":

 Active on traffic to -
  zone [DMZ] from zone [OUTSIDE]

rule  action   proto     packets  bytes
----  ------   -----     -------  -----
20    accept   tcp       0        0
  condition - saddr 0.0.0.0/0 daddr 192.1.1.100  dports 80,443

30    accept   udp       21       1764
  condition - saddr 0.0.0.0/0 daddr 192.1.1.101

10000 drop     all       4        240
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0
```

Fig.18: FW1 status during a Ping from Outside to DMZ



```
vyos@FW1:~$ show firewall name FROM-OUTSIDE-TO-DMZ

-----------------------------
Rulesets Information
-----------------------------

IPv4 Firewall "FROM-OUTSIDE-TO-DMZ":

 Active on traffic to -
  zone [DMZ] from zone [OUTSIDE]

rule  action   proto     packets  bytes
----  ------   -----     -------  -----
20    accept   tcp       0        0
  condition - saddr 0.0.0.0/0 daddr 192.1.1.100  dports 80,443

30    accept   udp       0        0
  condition - saddr 0.0.0.0/0 daddr 192.1.1.101 udp dpt:53

10000 drop     all       14       912
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0
```

Fig.19: FW2 status during a Ping from Outside to DMZ

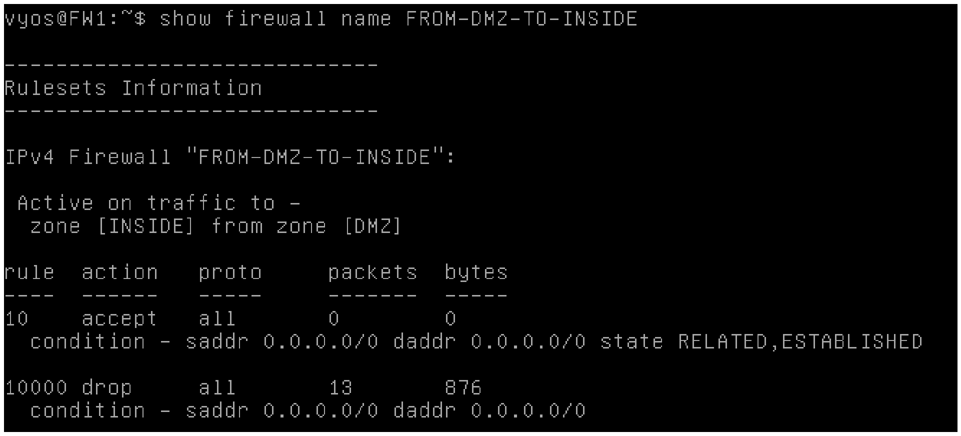
To mitigate unauthorized access from the DMZ zone to other areas, we implemented a preventive measure to restrict autonomous communication from the DMZ. Two new firewalls, namely FROM-DMZ-TO-INSIDE and FROM-DMZ-TO-OUTSIDE, were introduced. These firewalls exclusively permit traffic when it's a response to a request initiated by a device within the DMZ zone.

```
DNS> ping 10.2.2.100 -P 17
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

DNS> ping 10.2.2.100 -P 6
Connect   7@10.2.2.100 timeout
Connect   7@10.2.2.100 timeout
Connect   7@10.2.2.100 timeout
Connect   7@10.2.2.100 timeout
Connect   7@10.2.2.100 timeout
```

Fig.20: DNS ping to Inside unsuccessfully

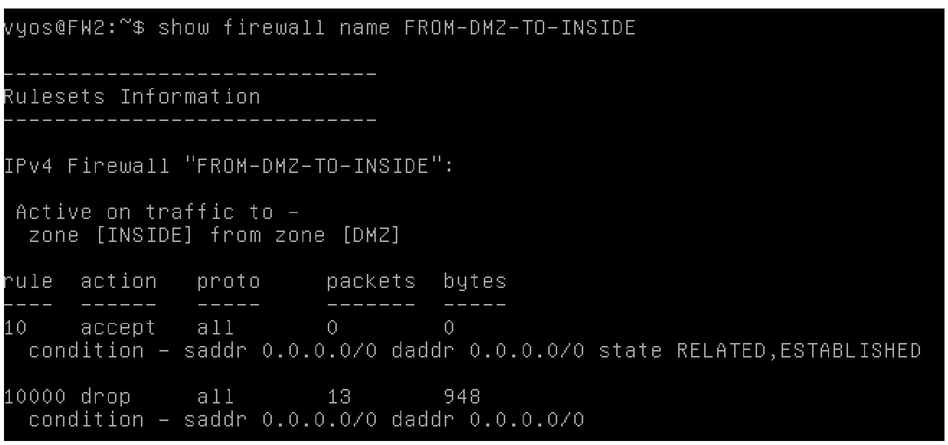
```
DNS> ping 200.2.2.100 -P 17
200.2.2.100 udp_seq=1 timeout
200.2.2.100 udp_seq=2 timeout
200.2.2.100 udp_seq=3 timeout
200.2.2.100 udp_seq=4 timeout
200.2.2.100 udp_seq=5 timeout

DNS> ping 200.2.2.100 -P 6
Connect   7@200.2.2.100 timeout
Connect   7@200.2.2.100 timeout
Connect   7@200.2.2.100 timeout
Connect   7@200.2.2.100 timeout
Connect   7@200.2.2.100 timeout
```

Fig.21: DNS ping to Outside unsuccessfully



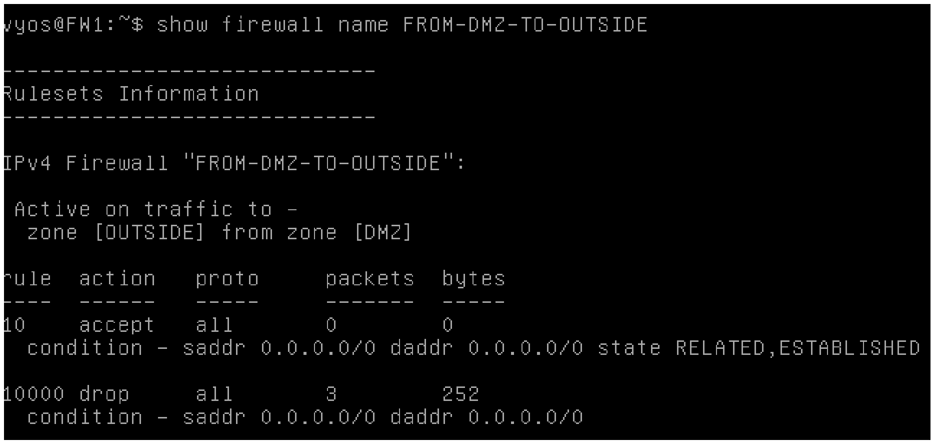
```
vyos@FW1:~$ show firewall name FROM-DMZ-TO-INSIDE

-----------------------------
Rulesets Information
-----------------------------

IPv4 Firewall "FROM-DMZ-TO-INSIDE":

 Active on traffic to -
  zone [INSIDE] from zone [DMZ]

rule  action   proto     packets  bytes
----  ------   -----     -------  -----
10    accept   all       0        0
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0 state RELATED,ESTABLISHED

10000 drop     all       13       876
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0
```

Fig.22: Firewall 1 status while DMZ is trying to iniciate communication with Inside zone



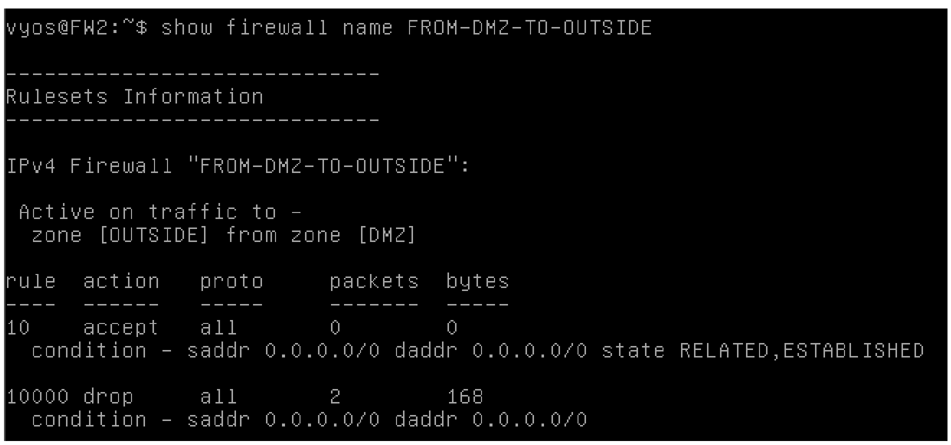
```
vyos@FW2:~$ show firewall name FROM-DMZ-TO-INSIDE

-----------------------------
Rulesets Information
-----------------------------

IPv4 Firewall "FROM-DMZ-TO-INSIDE":

 Active on traffic to -
  zone [INSIDE] from zone [DMZ]

rule  action   proto     packets  bytes
----  ------   -----     -------  -----
10    accept   all       0        0
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0 state RELATED,ESTABLISHED

10000 drop     all       13       948
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0
```

Fig.23: Firewall 2 status while DMZ is trying to iniciate communication with Inside zone



```
vyos@FW1:~$ show firewall name FROM-DMZ-TO-OUTSIDE

-----------------------------
Rulesets Information
-----------------------------

IPv4 Firewall "FROM-DMZ-TO-OUTSIDE":

 Active on traffic to -
  zone [OUTSIDE] from zone [DMZ]

rule  action   proto     packets  bytes
----  ------   -----     -------  -----
10    accept   all       0        0
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0 state RELATED,ESTABLISHED

10000 drop     all       3        252
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0
```

Fig.24: Firewall 1 status while DMZ is trying to iniciate communication with Outside zone



```
vyos@FW2:~$ show firewall name FROM-DMZ-TO-OUTSIDE

-----------------------------
Rulesets Information
-----------------------------

IPv4 Firewall "FROM-DMZ-TO-OUTSIDE":

 Active on traffic to -
  zone [OUTSIDE] from zone [DMZ]

rule  action   proto     packets  bytes
----  ------   -----     -------  -----
10    accept   all       0        0
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0 state RELATED,ESTABLISHED

10000 drop     all       2        168
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0
```

Fig.25: Firewall 2 status while DMZ is trying to iniciate communication with Outside zone

2.3 DDoS Attacks

During a DDoS attack, which is a malicious attempt to disrupt the normal traffic of a targeted server, service, or network by overwhelming it with a flood of internet traffic, synchronizing connection information between "Load Balancers" can overload them. This can cause both "Load Balancers" to fail, leaving the firewalls unprotected.

So, the solution is IP Hash for Load Balancing, which reduce the vulnerability and then we can disable synchronization between Load Balancers in a pair.

The algorithm IP Hash for Load Balncing uses the client's IP address (like a fingerprint) to decide which Load Balancer gets the traffic. Even without syncing, both Load Balancers will likely send traffic from the same client to the same firewall because they use the same "fingerprint" calculation. This keeps the load balanced and avoids overloading the Load Balancers during an attack.

Chapter 3

Conclusions

In our network setup, we have included a PCAttacker node to simulate potential security threats and test the resilience of our defenses. However, it's important to note that we have not developed any attack scripts or malicious activities for this node. Instead, its presence serves as a visual reminder of our commitment to thorough testing and preparation for potential security incidents.

By incorporating the PCAttacker node, we acknowledge the importance of proactively identifying and mitigating security vulnerabilities. While we have refrained from actively engaging in attacks, the inclusion of this node underscores our dedication to maintaining a vigilant and proactive approach to network security.

Moving forward, we remain focused on continually enhancing our network's defenses and refining our security protocols to ensure the utmost protection against potential threats.

In conclusion, the network configuration and security measures implemented in this project represent a comprehensive approach to safeguarding organizational assets and ensuring reliable network performance. By distributing firewall load through redundant load balancers and activating conntrack-sync, we enhance the resilience and availability of our network infrastructure.

The synchronization of load balancers eliminates the need for firewall synchronization by ensuring consistent routing decisions across redundant devices. Additionally, certain load balancing algorithms, such as round-robin or least connections, can further reduce the need for load balancer synchronization by evenly distributing traffic across available resources.

However, it's important to recognize that device/connection state synchronization, particularly during a DDoS attack, can introduce challenges and potential risks. In such scenarios, the rapid influx of malicious traffic can overwhelm synchronization mechanisms, leading to performance degradation or service disruptions.

Moving forward, our network defense policies emphasize the implementation of best practices to mitigate risks and protect against potential threats. By allowing limited access for internal users to internal, DMZ, and external services, while enforcing strict access controls for public services in the DMZ, we strike a balance between accessibility and security.

In the event of a DDoS attack, our network is equipped with dynamic blocking capabilities, leveraging an external monitoring system to identify and block malicious IP addresses in real-time. This proactive approach helps to mitigate the impact of DDoS attacks and maintain the integrity and availability of our network resources.

Overall, this project underscores our commitment to robust network defense strategies and proactive security measures, ensuring the resilience and reliability of our network infrastructure in the face of evolving threats and challenges.

Chapter 4

Contributions

In conclusion, this report has comprehensively documented the configuration and operational tests conducted on a network featuring redundant load-balancers and firewalls. Through meticulous deployment, routing, synchronization, and policy-based configurations, the network's high-availability and security measures were thoroughly evaluated. The successful completion of these tasks underscores the importance of robust infrastructure and policy adherence in ensuring the resilience and integrity of network operations.

Our Contributions are:

- João Ferreira (103625) - 50%
- Rui Campos (103709) - 50%

Appendix A

Load-Balancers Codes

A.1 Load-Balancer 1

LB1A

```
# set system host-name LB1A
# set interfaces ethernet eth0 address 10.1.1.1/24
# set interfaces ethernet eth1 address 10.0.0.1/24
# set interfaces ethernet eth2 address 10.0.2.1/24
# set interfaces ethernet eth3 address 10.0.1.1/24
# set protocols static route 10.2.2.0/24 next-hop 10.1.1.10
# set load-balancing wan interface-health eth3 nexthop 10.0.1.3
# set load-balancing wan interface-health eth2 nexthop 10.0.2.4
# set load-balancing wan rule 1 inbound-interface eth0
# set load-balancing wan rule 1 interface eth3 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
# set high-availability vrrp group LB1Cluster vrid 10
# set high-availability vrrp group LB1Cluster interface eth1
# set high-availability vrrp group LB1Cluster virtual-address 192.168.100.1/24
# set high-availability vrrp sync-group LB1Cluster member LB1Cluster
# set high-availability vrrp group LB1Cluster rfc3768-compatibility
# set service conntrack-sync accept-protocol 'tcp,udp,icmp'
# set service conntrack-sync failover-mechanism vrrp sync-group LB1Cluster
# set service conntrack-sync interface eth1
# set service conntrack-sync mcast-group 225.0.0.50
# set service conntrack-sync disable-external-cache
# commit
# save
```

LB1B

```
# set system host-name LB1B
# set interfaces ethernet eth0 address 10.1.1.2/24
# set interfaces ethernet eth1 address 10.0.0.2/24
# set interfaces ethernet eth2 address 10.0.3.2/24
# set interfaces ethernet eth3 address 10.0.6.2/24
# set protocols static route 10.2.2.0/24 next-hop 10.1.1.10
# set load-balancing wan interface-health eth2 nexthop 10.0.3.4
# set load-balancing wan interface-health eth3 nexthop 10.0.6.3
# 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 sticky-connections inbound
# set load-balancing wan disable-source-nat
# set high-availability vrrp group LB1Cluster vrid 10
# set high-availability vrrp group LB1Cluster interface eth1
# set high-availability vrrp group LB1Cluster virtual-address 192.168.100.1/24
# set high-availability vrrp sync-group LB1Cluster member LB1Cluster
# set high-availability vrrp group LB1Cluster rfc3768-compatibility
# set service conntrack-sync accept-protocol 'tcp,udp,icmp'
# set service conntrack-sync failover-mechanism vrrp sync-group LB1Cluster
# set service conntrack-sync interface eth1
# set service conntrack-sync mcast-group 225.0.0.50
# set service conntrack-sync disable-external-cache
# commit
# save
```

A.2 Load-Balancer 2

LB2A

```
# set system host-name LB2A
# set interfaces ethernet eth0 address 10.0.8.5/24
# set interfaces ethernet eth1 address 10.0.5.5/24
# set interfaces ethernet eth2 address 10.0.7.5/24
# set interfaces ethernet eth3 address 200.1.1.1/24
# set protocols static route 200.2.2.0/24 next-hop 200.1.1.10
# set load-balancing wan interface-health eth1 next-hop 10.0.5.3
# set load-balancing wan interface-health eth0 next-hop 10.0.8.4
# set load-balancing wan rule 1 inbound-interface eth3
# set load-balancing wan rule 1 interface eth1 weight 1
# set load-balancing wan rule 1 interface eth0 weight 1
# set load-balancing wan sticky-connections inbound
# set load-balancing wan disable-source-nat
# set high-availability vrrp group LB2Cluster vrid 20
# set high-availability vrrp group LB2Cluster interface eth2
```

```
# set high-availability vrrp group LB2Cluster virtual-address 192.168.200.1/24
# set high-availability vrrp sync-group LB2Cluster member LB2Cluster
# set high-availability vrrp group LB2Cluster rfc3768-compatibility
# set service conntrack-sync accept-protocol 'tcp,udp,icmp'
# set service conntrack-sync failover-mechanism vrrp sync-group LB2Cluster
# set service conntrack-sync interface eth2
# set service conntrack-sync mcast-group 225.0.0.100
# set service conntrack-sync disable-external-cache
# commit
# save
```

LB2B

```
# set system host-name LB2B
# set interfaces ethernet eth0 address 10.0.4.6/24
# set interfaces ethernet eth1 address 10.0.9.6/24
# set interfaces ethernet eth2 address 10.0.7.6/24
# set interfaces ethernet eth3 address 200.1.1.2/24
# set protocols static route 200.2.2.0/24 next-hop 200.1.1.10
# set load-balancing wan interface-health eth0 nexthop 10.0.4.4
# set load-balancing wan interface-health eth1 nexthop 10.0.9.3
# set load-balancing wan rule 1 inbound-interface eth3
# 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
# set high-availability vrrp group LB2Cluster vrid 20
# set high-availability vrrp group LB2Cluster interface eth2
# set high-availability vrrp group LB2Cluster virtual-address 192.168.200.1/24
# set high-availability vrrp sync-group LB2Cluster member LB2Cluster
# set high-availability vrrp group LB2Cluster rfc3768-compatibility
# set service conntrack-sync accept-protocol 'tcp,udp,icmp'
# set service conntrack-sync failover-mechanism vrrp sync-group LB2Cluster
# set service conntrack-sync interface eth2
# set service conntrack-sync mcast-group 225.0.0.100
# set service conntrack-sync disable-external-cache
# commit
# save
```

A.3 Load-Balancer DMZ

LBDMZ

```
# set system host-name LBDMZ
# set interfaces ethernet eth0 address 10.0.10.10/24
```

```
# set interfaces ethernet eth1 address 10.0.11.10/24
# set interfaces ethernet eth2 address 192.1.1.140/24
# set load-balancing wan interface-health eth0 nexthop 10.0.10.1
# set load-balancing wan interface-health eth1 nexthop 10.0.11.1
# 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
```

Appendix B

Firewalls Codes

B.1 Firewall 1

FW1

```
# set system host-name FW1
# set interfaces ethernet eth0 address 10.0.9.3/24
# set interfaces ethernet eth1 address 10.0.5.3/24
# set interfaces ethernet eth2 address 10.0.6.3/24
# set interfaces ethernet eth3 address 10.0.1.3/24
# set interfaces ethernet eth4 address 10.0.10.1/24
# set protocols static route 0.0.0.0/0 next-hop 10.0.5.5
# set protocols static route 0.0.0.0/0 next-hop 10.0.9.6
# set protocols static route 10.2.2.0/24 next-hop 10.0.1.1
# set protocols static route 10.2.2.0/24 next-hop 10.0.6.2
# set protocols static route 192.1.1.0/24 next-hop 10.0.10.10
# set nat source rule 10 outbound-interface eth01
# set nat source rule 10 source address 10.0.0.0/8
# set nat source rule 10 translation address 192.1.0.1-192.1.0.62
# set nat source rule 20 outbound-interface eth0
# set nat source rule 20 source address 10.0.0.0/8
# set nat source rule 20 translation address 192.1.0.65-192.1.0.126
# set zone-policy zone INSIDE description "Inside"
# set zone-policy zone INSIDE interface eth2
# set zone-policy zone INSIDE interface eth3
# set zone-policy zone OUTSIDE description "Outside"
# set zone-policy zone OUTSIDE interface eth0
# set zone-policy zone OUTSIDE interface eth1
# set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 action accept
# set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 protocol udp
# set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 destination port 5000-6000
# set zone-policy zone OUTSIDE from INSIDE firewall name FROM-INSIDE-TO-OUTSIDE
```

```
# set firewall name FROM-OUTSIDE-TO-INSIDE rule 10 action accept
# set firewall name FROM-OUTSIDE-TO-INSIDE rule 10 state established enable
# set firewall name FROM-OUTSIDE-TO-INSIDE rule 10 state related enable
# set zone-policy zone INSIDE from OUTSIDE firewall name FROM-OUTSIDE-TO-INSIDE
# set firewall name FROM-INSIDE-TO-DMZ rule 20 description "TCP"
# set firewall name FROM-INSIDE-TO-DMZ rule 20 action accept
# set firewall name FROM-INSIDE-TO-DMZ rule 20 protocol tcp
# set firewall name FROM-INSIDE-TO-DMZ rule 20 destination port 80,443,22
# set firewall name FROM-INSIDE-TO-DMZ rule 20 destination address 192.1.1.100
# set firewall name FROM-INSIDE-TO-DMZ rule 30 description "UDP"
# set firewall name FROM-INSIDE-TO-DMZ rule 30 action accept
# 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 destination address 192.1.1.101
# set zone-policy zone DMZ from INSIDE firewall name FROM-INSIDE-TO-DMZ
# set firewall name FROM-DMZ-TO-INSIDE rule 10 action accept
# set firewall name FROM-DMZ-TO-INSIDE rule 10 state established enable
# set firewall name FROM-DMZ-TO-INSIDE rule 10 state related enable
# set zone-policy zone INSIDE from DMZ firewall name FROM-DMZ-TO-INSIDE
# set firewall name FROM-OUTSIDE-TO-DMZ rule 20 description "TCP"
# set firewall name FROM-OUTSIDE-TO-DMZ rule 20 action accept
# set firewall name FROM-OUTSIDE-TO-DMZ rule 20 protocol tcp
# set firewall name FROM-OUTSIDE-TO-DMZ rule 20 destination port 80,443
# set firewall name FROM-OUTSIDE-TO-DMZ rule 20 destination address 192.1.1.100
# set firewall name FROM-OUTSIDE-TO-DMZ rule 30 description "UDP"
# set firewall name FROM-OUTSIDE-TO-DMZ rule 30 action accept
# set firewall name FROM-OUTSIDE-TO-DMZ rule 30 protocol udp
# set firewall name FROM-OUTSIDE-TO-DMZ rule 30 destination port 53
# set firewall name FROM-OUTSIDE-TO-DMZ rule 30 destination address 192.1.1.101
# set zone-policy zone DMZ from OUTSIDE firewall name FROM-OUTSIDE-TO-DMZ
# set firewall name FROM-DMZ-TO-OUTSIDE rule 10 action accept
# set firewall name FROM-DMZ-TO-OUTSIDE rule 10 state established enable
# set firewall name FROM-DMZ-TO-OUTSIDE rule 10 state related enable
# set zone-policy zone OUTSIDE from DMZ firewall name FROM-DMZ-TO-OUTSIDE
# set zone-policy zone DMZ description "DMZ Server"
# set zone-policy zone DMZ interface eth4
# commit
# save
```

B.2 Firewall 2

FW2

```
# set system host-name FW2
# set interfaces ethernet eth0 address 10.0.8.4/24
# set interfaces ethernet eth1 address 10.0.4.4/24
```

```
# set interfaces ethernet eth2 address 10.0.2.4/24
# set interfaces ethernet eth3 address 10.0.3.4/24
# set interfaces ethernet eth4 address 10.0.11.1/24
# set protocols static route 0.0.0.0/0 next-hop 10.0.4.6
# set protocols static route 0.0.0.0/0 next-hop 10.0.8.5
# set protocols static route 10.2.2.0/24 next-hop 10.0.3.2
# set protocols static route 10.2.2.0/24 next-hop 10.0.2.1
# set protocols static route 192.1.1.0/24 next-hop 10.0.11.10
# set nat source rule 10 outbound-interface eth1
# set nat source rule 10 source address 10.0.0.0/8
# set nat source rule 10 translation address 192.1.0.129-192.1.0.190
# set nat source rule 20 outbound-interface eth0
# set nat source rule 20 source address 10.0.0.0/8
# set nat source rule 20 translation address 192.1.0.129-192.1.0.254
# set zone-policy zone INSIDE description "Inside"
# set zone-policy zone INSIDE interface eth2
# set zone-policy zone INSIDE interface eth3
# set zone-policy zone OUTSIDE description "Outside"
# set zone-policy zone OUTSIDE interface eth0
# set zone-policy zone OUTSIDE interface eth1
# set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 action accept
# set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 protocol udp
# set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 destination port 5000-6000
# set zone-policy zone OUTSIDE from INSIDE firewall name FROM-INSIDE-TO-OUTSIDE
# set firewall name FROM-OUTSIDE-TO-INSIDE rule 10 action accept
# set firewall name FROM-OUTSIDE-TO-INSIDE rule 10 state established enable
# set firewall name FROM-OUTSIDE-TO-INSIDE rule 10 state related enable
# set zone-policy zone INSIDE from OUTSIDE firewall name FROM-OUTSIDE-TO-INSIDE
# set firewall name FROM-INSIDE-TO-DMZ rule 20 description "TCP"
# set firewall name FROM-INSIDE-TO-DMZ rule 20 action accept
# set firewall name FROM-INSIDE-TO-DMZ rule 20 protocol tcp
# set firewall name FROM-INSIDE-TO-DMZ rule 20 destination port 80,443,22
# set firewall name FROM-INSIDE-TO-DMZ rule 20 destination address 192.1.1.100
# set firewall name FROM-INSIDE-TO-DMZ rule 30 description "UDP"
# set firewall name FROM-INSIDE-TO-DMZ rule 30 action accept
# 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 destination address 192.1.1.101
# set zone-policy zone DMZ from INSIDE firewall name FROM-INSIDE-TO-DMZ
# set firewall name FROM-DMZ-TO-INSIDE rule 10 action accept
# set firewall name FROM-DMZ-TO-INSIDE rule 10 state established enable
# set firewall name FROM-DMZ-TO-INSIDE rule 10 state related enable
# set zone-policy zone INSIDE from DMZ firewall name FROM-DMZ-TO-INSIDE
# set firewall name FROM-OUTSIDE-TO-DMZ rule 20 description "TCP"
# set firewall name FROM-OUTSIDE-TO-DMZ rule 20 action accept
# set firewall name FROM-OUTSIDE-TO-DMZ rule 20 protocol tcp
```

```
# set firewall name FROM-OUTSIDE-TO-DMZ rule 20 destination port 80,443
# set firewall name FROM-OUTSIDE-TO-DMZ rule 20 destination address 192.1.1.100
# set firewall name FROM-OUTSIDE-TO-DMZ rule 30 description "UDP"
# set firewall name FROM-OUTSIDE-TO-DMZ rule 30 action accept
# set firewall name FROM-OUTSIDE-TO-DMZ rule 30 protocol udp
# set firewall name FROM-OUTSIDE-TO-DMZ rule 30 destination port 53
# set firewall name FROM-OUTSIDE-TO-DMZ rule 30 destination address 192.1.1.101
# set zone-policy zone DMZ from OUTSIDE firewall name FROM-OUTSIDE-TO-DMZ
# set firewall name FROM-DMZ-TO-OUTSIDE rule 10 action accept
# set firewall name FROM-DMZ-TO-OUTSIDE rule 10 state established enable
# set firewall name FROM-DMZ-TO-OUTSIDE rule 10 state related enable
# set zone-policy zone OUTSIDE from DMZ firewall name FROM-DMZ-TO-OUTSIDE
# set zone-policy zone DMZ description "DMZ Server"
# set zone-policy zone DMZ interface eth4
# commit
# save
```

Appendix C

Routers Codes

C.1 Router Inside

RInside

```
# conf t
# int f0/1
# ip address 10.2.2.10 255.255.255.0
# no shutdown
# int f0/0
# ip address 10.1.1.10 255.255.255.0
# no shutdown
# ip route 0.0.0.0 0.0.0.0 10.1.1.1
# ip route 0.0.0.0 0.0.0.0 10.1.1.2
# end
# write
```

C.2 Router Outside

ROutside

```
# conf t
# int f0/0
# ip address 200.1.1.10 255.255.255.0
# no shutdown
# int f0/1
# ip address 200.2.2.10 255.255.255.0
# no shutdown
# end
# write
# conf t
```

```
# ip route 192.1.0.0 255.255.254.0 200.1.1.1
# ip route 192.1.0.0 255.255.254.0 200.1.1.2
# end
# write
```