Segurança em Redes de Comunicações

High-Availability Firewall Scenarios

Universidade de Aveiro

Bernardo Falé,Lara Rodrigues

Segurança em Redes de Comunicações

High-Availability Firewall Scenarios

DETI

Universidade de Aveiro

Bernardo Falé,Lara Rodrigues
(93331) mbfale@ua.pt,(93427) laravieirarodrigues@ua.pt

Abril 2023

Conteúdo

Capítulo 1

Introdução

Neste trabalho foi proposto a realização de um cenário de load-balancing redundante, que permitisse um fluxo de tráfego de acordo com boas práticas de segurança em redes. O processo de inclusão/exclusão de comandos está presente num repositório de GitHub em `https://github.com/bernardofale/SRC`

1.1 Load-Balancing Scenario (with redundancy and state synchronization)

Neste primeiro ponto do relatório foi proposto que os alunos configurassem uma rede num cenário que implementasse uma ligação stateful (entre load-balancers) e que promovesse redundância, sem pontos de falha.

Inicialmente não conseguimos observar o comportamento correto da rede, e por isso, decidimos ter uma approach iterativa, para conseguirmos corrigir os IP's, as rotas, e principalmente as interfaces, visto que se não forem bem configuradas os load-balancers não conseguirão rotear os pacotes de forma correta (Sticky connections). Depois de testarmos a ligação entre os VPC's com rotas estáticas configuradas previamente (De/para a rede 10.2.2.0/24 / 200.2.2.0/24) configurámos o VRRP, o conntrack, e os serviços de load-balancing; Assim conseguimos desde já, retirar as rotas estáticas presentes nos load-balancers, e "trocar"pelo routing next-hop desses mesmos serviços. O mecanismo de sincronização foi feito, como dito anteriormente, de acordo com o VRRP e através de uma criação de dois clusters, um para cada dois load-balancers. De seguida foi feita a verificação destas configurações; Foi efetuada uma captura no wireshark nas ligações entre load-balancers e entre FW's e load-balancers, e conseguímos verificar que existia uma troca de pacotes VRRP e ICMP, respetivamente. Esta troca era feita periodicamente, no segundo caso seriam "Hello's"para que a rota entre os load-balancers e as firewalls fosse mantida de forma correta. Depois desta verificação avançámos para a configuração das zonas internas e externas da rede; De notar que os border load-balancers apenas têm rota estática para o R1/R2.

Antes da criação das zonas, qualquer tráfego era permitido na rede, isto é, o VyOs por default permite a passagem de pacotes até a primeira customização ser configurada; Os devices que tinham conectividade à priori (VPC's), deixariam, na teoria, de ter essa conectividade. Esta hipótese foi corroborada com algumas capturas entre a rede interna e a rede externa. Depois, então, de configurar as firewalls para permitir apenas tráfego UDP, entre os portos 5000 e 6000, foi feito um teste nas ligações FW1-LB1A e FW1-LB1B, e apurámos que o funcionamento estava correto, ou seja, tínhamos um request e um reply. Com mais uma iteração completa, decídimos configurar o NAT nas firewalls, e remover as rotas estáticas explícitas para a rede interna no R2 para permitir o encaminhamento de tráfego através da NAT pool. Como temos duas firewalls usámos duas pools diferentes, com a mesma rede de origem. Facilmente comprovámos o funcionamento desta configuração, isto é, fizemos uma captura entre FW's e border load-balancers da rede externa para verificar o uso destes IP's. Finalmente, verificámos que as firewalls estavam bem configuradas fazendo um ping UDP simples entre os VPC's num porto fora da pool de destination ports.

Conseguímos concluir que os load-balancers permitem ausência de mecanismos de sincronização entre firewalls, no sentido em que a sua ligação stateful com o conntrack/sticky connections são o suficiente para que o fluxo de tráfego seja feito de forma correto sem comprometer a "privacidade"das firewalls. Este cenário de redundância permite também que não haja um point of failure, e permite que a rede escale de forma muito mais ativa, sem drawbacks gigantes.

1.2 Policies Definition and Integrated Deployment

Neste ponto do trabalho, após ser testada a ligação, foi criada a zona de DMZ, e as suas regras de acordo com o enunciado; Assim foram implementados os serviços de HTTP, HTTPS, DNS e SSH através da criação de regras com o protocolo TCP e os respetivos portos. Como os serviços na DMZ são públicos ambas as redes (interna/externa) têm acesso, com regras iguais; As regras para fora foram feitas de acordo com o estado da conexão, isto é, se um pacote que está associado a uma conexão gerou pacotes em ambas as direções, ou se esse pacote está a começar uma conexão mas está associado a uma já existente.

Os testes de implementação foram feitos através de ping's TCP nos diferentes portos. Com a realização dos testes foi possível observar que em alguns dos casos os pings devolvem timeout devido à ausência de um load balancer, que seria necessário para distribuir o tráfego da DMZ para as firewalls.

Para que seja possível o acesso do servidor via SSH às firewalls foi também configurado o serviço SSH. Este serviço nas firewalls está apenas à "escuta"nas interfaces ligada diretamente ao servidor para tornar a comunicação mais segura, e para que clientes de fora não consigam aceder.

O próximo ponto é a configuração das políticas de controlo de acesso; Para isso, criámos uma política com ação de deny que, ignorando o destino, exclui um dado host IP. Após a configuração foram desenvolvidos testes e para isto mudamos o IP do PC2 para verificar o tráfego nas firewalls. De acordo com a

teoria e com a política configuradas este VPC não deveria conseguir aceder aos serviços da DMZ, no entanto, estes não foram os resultados obtidos nos testes realizados. Apesar do IP do VPC coincidir com o IP excluído nas firewalls. Para solucionar este problema poderiamos criar uma nova chain rule na firewall, com a ação deny, associado à zona de DMZ, de forma a não deixar passar o tráfego vindo desse IP.

Por fim, era pedido que fosse feito um script que criasse automaticamente as regras de bloqueio nas firewalls durante um ataque DDoS mediante a identificação do endereço IP dos atacantes. Esta tarefa não foi conseguida com sucesso, apesar da ligação SSH estar funcional, uma vez que na prática os comandos não são executados remotamente, apesar da blueprint lá estar. Este script infere que recebe um argumento na linha de comandos com o número da regra pretendida, para criação. Este script está em anexo.

Capítulo 2

Anexo

2.1 Configurações

Configure ip address of PC1 and PC2
PC2> ip 200.2.2.100/24 200.2.2.10
PC1> ip 10.2.2.100/24 10.2.2.10

Configure ip addresses and routes in Router 2 (Outside)
R2(config)#interface f0/0
R2(config-if)#ip address 200.1.1.10 255.255.255.0
R2(config-if)#interface f0/1
R2(config-if)#ip address 200.2.2.10 255.255.255.0
R2(config)#ip route 192.1.0.0 255.255.255.0 200.1.1.1
R2(config)#ip route 192.1.0.0 255.255.255.0 200.1.1.2

Configure ip addresses and routes in Router 1(Inside)
R1(config)#interface f0/0
R1(config-if)#ip address 10.1.1.10 255.255.255.0
R1(config-if)#interface f0/1
R1(config-if)#ip address 10.2.2.10 255.255.255.0
R1(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1
R1(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.2

Configure names, addresses, and routes in FW1 and FW2

FW1
#set system host-name FW1
#set interfaces ethernet eth2 address 10.0.1.1/24
#set interfaces ethernet eth0 address 10.0.3.1/24
#set interfaces ethernet eth1 address 10.0.2.1/24
#set interfaces ethernet eth3 address 10.0.4.1/24

#set protocols static route 200.2.2.0/24 next-hop 10.0.3.10
#set protocols static route 200.2.2.0/24 next-hop 10.0.4.11
#set protocols static route 10.2.2.0/24 next-hop 10.0.1.10
#set protocols static route 10.2.2.0/24 next-hop 10.0.2.11
#commit
#save
#exit

FW2

#set system host-name FW2
#set interfaces ethernet eth2 address 10.0.2.2/24
#set interfaces ethernet eth0 address 10.0.4.2/24
#set interfaces ethernet eth1 address 10.0.1.2/24
#set interfaces ethernet eth3 address 10.0.3.2/24
#set protocols static route 200.2.2.0/24 next-hop 10.0.4.10
#set protocols static route 200.2.2.0/24 next-hop 10.0.3.11
#set protocols static route 10.2.2.0/24 next-hop 10.0.2.10
#set protocols static route 10.2.2.0/24 next-hop 10.0.1.11
#commit
#save
#exit

Configure VRRP and conntrack on Load-Balancers

#set high-availability vrrp group FWCluster vrid 10
#set high-availability vrrp group FWCluster interface eth1
#set high-availability vrrp group FWCluster virtual-address 192.168.100.1/24
#set high-availability vrrp sync-group FWCluster member FWCluster
#set high-availability vrrp group FWCluster rfc3768-compatibility

#set service conntrack-sync accept-protocol 'tcp,udp,icmp'
#set service conntrack-sync failover-mechanism vrrp sync-group FWCluster
#set service conntrack-sync interface eth1
#set service conntrack-sync mcast-group 225.0.0.50
#set service conntrack-sync disable-external-cache

Configure names, adresses, and routes in load balancers (LB1A, LB1B, LB2A, LB1A)

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 10.0.1.10/24
#set interfaces ethernet eth3 address 10.0.2.10/24
#set protocols static route 10.2.2.0/24 next-hop 10.1.1.10

#commit
#save
#exit

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 10.0.2.11/24
#set interfaces ethernet eth3 address 10.0.1.11/24
#set protocols static route 10.2.2.0/24 next-hop 10.1.1.10
#commit
#save
#exit

LB2A

#set system host-name LB2A
#set interfaces ethernet eth0 address 200.1.1.1/24
#set interfaces ethernet eth1 address 10.0.0.3/24
#set interfaces ethernet eth2 address 10.0.3.10/24
#set interfaces ethernet eth3 address 10.0.4.10/24
#set protocols static route 200.2.2.0/24 next-hop 200.1.1.10
#commit
#save
#exit

LB2B

#set system host-name LB2B
#set interfaces ethernet eth0 address 200.1.1.2/24
#set interfaces ethernet eth1 address 10.0.0.4/24
#set interfaces ethernet eth2 address 10.0.4.11/24
#set interfaces ethernet eth3 address 10.0.3.11/24
#set protocols static route 200.2.2.0/24 next-hop 200.1.1.10
#commit
#save
#exit

Configure zone-policy and firewall

FW1

#set zone-policy zone INSIDE description "Internal Network"
#set zone-policy zone INSIDE interface eth2
#set zone-policy zone INSIDE interface eth1
#set zone-policy zone OUTSIDE description "External Network"
#set zone-policy zone OUTSIDE interface eth0
#set zone-policy zone OUTSIDE interface eth3

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

Configure NAT/PAT mechanisms in FW1 and FW2

#set nat source rule 100 outbound-interface eth0
#set nat source rule 100 source address 10.0.0.0/8
#set nat source rule 100 translation address 192.1.0.1-192.1.0.10 (/192.1.0.11-192.1.0.20)
#set nat source rule 200 outbound-interface eth3
#set nat source rule 200 source address 10.0.0.0/8
#set nat source rule 200 translation address 192.1.0.1-192.1.0.10 (/192.1.0.11-192.1.0.20)
#commit
#save
#exit

Configure load-balancing services

LB2A

#set load-balancing wan interface-health eth2 nexthop 10.0.3.1
#set load-balancing wan interface-health eth3 nexthop 10.0.4.2
#set load-balancing wan rule 1 inbound-interface eth0
#set load-balancing wan rule 1 interface eth2 weight 1
#set load-balancing wan rule 1 interface eth3 weight 1
#set load-balancing wan sticky-connections inbound
#set load-balancing wan disable-source-nat

LB2B

#set load-balancing wan interface-health eth2 nexthop 10.0.4.1
#set load-balancing wan interface-health eth3 nexthop 10.0.3.2
#set load-balancing wan rule 1 inbound-interface eth0
#set load-balancing wan rule 1 interface eth2 weight 1
#set load-balancing wan rule 1 interface eth3 weight 1
#set load-balancing wan sticky-connections inbound
#set load-balancing wan disable-source-nat

LB1A

#sset load-balancing wan interface-health eth2 nexthop 10.0.1.1
#sset load-balancing wan interface-health eth3 nexthop 10.0.2.2
#sset load-balancing wan rule 1 inbound-interface eth0
#sset load-balancing wan rule 1 interface eth2 weight 1
#sset load-balancing wan rule 1 interface eth3 weight 1
#sset load-balancing wan sticky-connections inbound
#sset load-balancing wan disable-source-nat

LB1B

#set load-balancing wan interface-health eth2 nexthop 10.0.2.1
#set load-balancing wan interface-health eth3 nexthop 10.0.1.2
#set load-balancing wan rule 1 inbound-interface eth0
#set load-balancing wan rule 1 interface eth2 weight 1
#set load-balancing wan rule 1 interface eth3 weight 1
#set load-balancing wan sticky-connections inbound
#set load-balancing wan disable-source-nat

Configure zone-policy and firewall

FW1

#set zone-policy zone INSIDE description "Internal Network"
#set zone-policy zone INSIDE interface eth2
#set zone-policy zone INSIDE interface eth1
#set zone-policy zone OUTSIDE description "External Network"
#set zone-policy zone OUTSIDE interface eth0
#set zone-policy zone OUTSIDE interface eth
#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 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 zone-policy zone OUTSIDE from INSIDE firewall name FROM-INSIDE-TO-OUTSIDE

Cofigure VRRP and conntrack on Load-Balancers

#set high-availability vrrp group FWCluster vrid 10
#set high-availability vrrp group FWCluster interface eth1
#set high-availability vrrp group FWCluster virtual-address 192.168.100.1/24
#set high-availability vrrp sync-group FWCluster member FWCluster

#set high-availability vrrp group FWCluster rfc3768-compatibility

#set service conntrack-sync accept-protocol 'tcp,udp,icmp'
#set service conntrack-sync failover-mechanism vrrp sync-group FWCluster
#set service conntrack-sync interface eth1
#set service conntrack-sync mcast-group 225.0.0.50
#set service conntrack-sync disable-external-cache

Add server and create DMZ
FW1> set interfaces ethernet eth4 address 192.1.1.1/24
FW2> set interfaces ethernet eth4 address 192.1.1.2/24

#set zone-policy zone DMZ description "DMZ (Server Farm)"
#set zone-policy zone DMZ interface eth4

#set firewall name FROM-INSIDE-TO-DMZ rule 10 action accept
#set firewall name FROM-INSIDE-TO-DMZ rule 10 protocol udp
#set firewall name FROM-INSIDE-TO-DMZ rule 10 destination port 5000-6000
#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 zone-policy zone DMZ from INSIDE firewall name FROM-INSIDE-TO-DMZ

Update R2 to support routes to the DMZ zone R2(config)no ip route 192.1.0.0 255.255.255.0 200.1.1.2
R2(config)no ip route 192.1.0.0 255.255.255.0 200.1.1.1
R2(config)ip route 192.1.0.0 255.255.0.0 200.1.1.2
R2(config)ip route 192.1.0.0 255.255.0.0 200.1.1.1

Enable SSH, HTTP, HTTPS and DNS services in the firewalls
#set firewall name TO-DMZ rule 12 action accept
#set firewall name TO-DMZ rule 12 description 'HTTP'
#set firewall name TO-DMZ rule 12 protocol tcp
#set firewall name TO-DMZ rule 12 destination address 192.1.1.100
#set firewall name TO-DMZ rule 12 destination port 80

#set firewall name TO-DMZ rule 14 action accept
#set firewall name TO-DMZ rule 14 description 'HTTPS'
#set firewall name TO-DMZ rule 14 protocol tcp
#set firewall name TO-DMZ rule 14 destination address 192.1.1.100
#set firewall name TO-DMZ rule 14 destination port 443

#set firewall name TO-DMZ rule 16 action accept
#set firewall name TO-DMZ rule 16 description 'DNS'

#set firewall name TO-DMZ rule 16 protocol tcp
#set firewall name TO-DMZ rule 16 destination address 192.1.1.100
#set firewall name TO-DMZ rule 16 destination port 53

#set firewall name TO-DMZ rule 20 aiction accept
#set firewall name TO-DMZ rule 20 description 'SSH'
#set firewall name TO-DMZ rule 20 protocol tcp
#set firewall name TO-DMZ rule 20 destination address 192.1.1.100
#set firewall name TO-DMZ rule 20 destination port 22

#set zone-policy zone DMZ from INSIDE firewall name TO-DMZ
#set zone-policy zone DMZ from OUTSIDE firewall name TO-DMZ

Turn on SSH service for remote connection
#set service ssh

On the server:
ssh vyos@FirewallIP

Creating ACL policies in the firewalls
#set policy access-list 100
#set policy access-list 100 description 'Banned IPs'
#set policy access-list 100 rule 10 action deny
#set policy access-list 100 rule 10 destination any
#set policy access-list 100 rule 10 source host 200.2.2.105 (for testing)

2.2 Script

```
import paramiko
import sys

try:
    # Connect to the device using SSH
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh.connect("192.1.1.1", username="vyos", password="vyos")

    with open("ip_list.txt", "r") as f:
        ips = [line.strip() for line in f]

    for ip in ips:
        ssh.exec_command("configure")
        ssh.exec_command("set policy access-list 100 rule "+ sys.argv[1] +
        " action deny")
```

```
            ssh.exec_command("set policy access-list 100 rule "+ sys.argv[1] +
            " destionation any")
            ssh.exec_command("set policy access-list 100 rule "+ sys.argv[1] +
            " source host " +ip)
            ssh.exec_command("commit")
            ssh.exec_command("save")
            ssh.exec_command("exit")
        # Close the SSH connection
        ssh.close()

        # Empty the file
        with open("ip_list.txt", "w") as f:
            f.write("")

    except paramiko.AuthenticationException:
            print(f"Failed to connect to {ip}: authentication failed")
    except paramiko.SSHException as e:
            print(f"Failed to connect to {ip}: {e}")
```