Segurança em Redes de Comunicações

Report 1

Universidade de Aveiro

Bruno Silva (97931)brunosilva16@ua.pt
Marta Oliveira (97613) marta.alex@ua.pt

VERSAO 1

Segurança em Redes de Comunicações

DETI

Universidade de Aveiro

Bruno Silva (97931)brunosilva16@ua.pt
Marta Oliveira (97613) marta.alex@ua.pt

16 de Abril de 2023

Índice

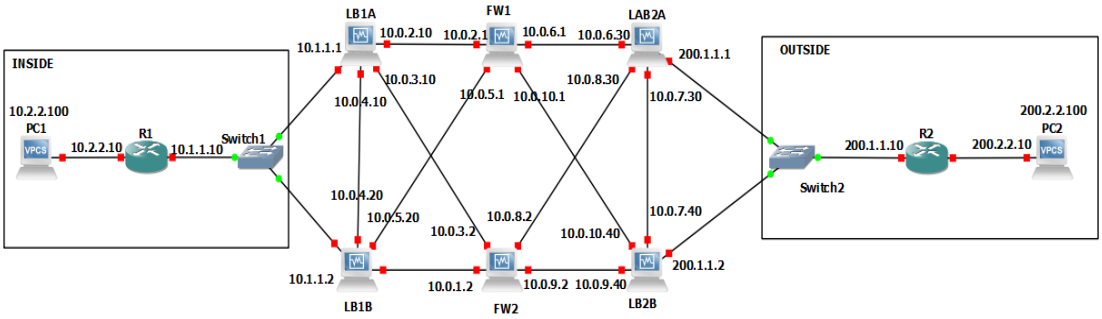
Capítulo 1

Introdução

O presente relatório tem como objetivo descrever a resolução do guião *HIGH-AVAILABILITY FIREWALLS SCENARIOS* desenvolvido no âmbito da unidade curricular de Segurança em Redes de Comunicações.

Capítulo 2

Exercício 9)



Load-balancers synchronization

Os Load balancers são responsáveis na distribuição de tráfego de rede de forma equilibrada entre servidores.

Em casos como neste exerícico, múltiplos load balancers são utilizados para criar uma configuração redundante, sendo portanto essencial que estes estejam sincronizados (configurações e políticas de encaminhamentos iguais).

Após esta sincronização torna-se desnecessário configurar firewalls individualmente em cada servidor pois os load balancers, como já mencionado, controlam o acesso a servidores. Isto simplifica a configuração da rede e reduz o risco de má configuração e inconsistência entre servidores.

Neste cenário, como estamos a utilizar duas firewals, sem sincronismo, criámos duas pools de endereços públicos distintas. Os Load balancers reencaminham o

tráfego para a firewall correspondente que traduziu o IP e desta forma, não existe necessidade dos firewalls sincronizarem as suas tabelas de tradução NAT/PAT.

Portanto, esta configuração permite que a carga de trabalho seja distribuída de forma eficiente.

Load-balancing algorithms

O algoritmo *Ip hash* distribui tráfego com base no endereço IP de origem do pedido recebido. Consequentemente, os pedidos provenientes do mesmo endereço IP de origem serão sempre encaminhados para o mesmo servidor.

Se um Load Balancer não estiver sincronizado e a direcionar o tráfego para um conjunto diferente de servidores, este algoritmo consegue mesmo assim distribuir tráfego uniformemente entre os servidores que estão a ser acessados pelos Load Balancers que se encontram sincronizados.

Isso ocorre pois os pedidos provenientes do mesmo endereço IP de origem serão sempre direcionados para o mesmo servidor, independentemente de qual Load Balancer esteja a direcionar o tráfego.

State Synchronization during a DDoS Attack

Durante um ataque de negação de serviço distribuído (DDoS), um atacante tenta sobrecarregar um site ou rede com um número elevado de tráfego.

Um dos problemas causados com a sincronização dos dispositivos durante um ataque DDoS é que esta pode causar a sobrecarga de todos os dispositivos e impedir que funcionem adequadamente. Se os dispositivos estiverem sincronizados, todos podem tentar bloquear o mesmo tráfego de entrada ao mesmo tempo. Consequentemente, a largura de banda da rede e a capacidade de resposta dos serviços serão afetadas.

A sincronização dos estados de conexão pode também causar falsos positivos, o que leva ao bloqueio de tráfego legítimo. Durante um ataque DDoS, muitos pedidos de entrada podem parecer maliciosos, mas alguns podem ser tráfego benigno de usuários que estão a tentar acessar o site "alvo".

Além disso, se o dispositivo sincronizado falhar ou for comprometido, pode fazer com que todos os dispositivos falhem e deixem o site ou rede de destino vulnerável a ataques.

Configuração

De seguida, iremos mostrar partes da nossa configuração de forma a demonstrar o funcionamento do que foi pedido para este exercício:

Na captura seguinte, observamos a comunicação entre ambos os pc's.

```
Checking for duplicate address...
PC1 : 10.2.2.100 255.255.255.0 gateway 10.2.2.10

PC1>  ping 200.2.2.100 -P 17 -p 5001
84 bytes from 200.2.2.100 udp_seq=1 ttl=59 time=95.118 ms
84 bytes from 200.2.2.100 udp_seq=2 ttl=59 time=37.105 ms
84 bytes from 200.2.2.100 udp_seq=3 ttl=59 time=36.411 ms
84 bytes from 200.2.2.100 udp_seq=4 ttl=59 time=38.651 ms
84 bytes from 200.2.2.100 udp_seq=5 ttl=59 time=37.266 ms
```

Como é necessário haver sincronização entre os load balancers, em cada um deles foi configurado o seguinte:

```
high-availability {
    vrrp {
        group FWCluster {
            interface eth1
            rfc3768-compatibility
            virtual-address 192.168.100.1/24
            vrid 10
        }
        sync-group FWCluster {
            member FWCluster
        }
    }
}
```

```
service {
    conntrack-sync {
        accept-protocol tcp,udp,icmp
        disable-external-cache
        event-listen-queue-size 8
        failover-mechanism {
            vrrp {
                sync-group FWCluster
            }
        }
        interface eth1 {
        }
        mcast-group 225.0.0.50
        sync-queue-size 1
    }
```

De maneira a controlar o fluxo na rede foram criadas duas regras no FW1 e FW2 para permitir apenas tráfego UDP com origem em INSIDE e com destino a OUTSIDE que usam portos de destino entre 5000 e 6000.

Desta forma, criámos:

1. INSIDE-TO-OUTSIDE
2. OUTSIDE-TO-INSIDE

Para isso, foi necessário em primeiro, definir as zonas:

```
}
zone-policy {
    zone INSIDE {
        default-action drop
        description "Inside (Internal Network)"
        from OUTSIDE {
            firewall {
                name FROM-OUTSIDE-TO-INSIDE
            }
        }
        interface eth0
        interface eth1
    }
    zone OUTSIDE {
        default-action drop
        description "Outside (Internet)"
        from INSIDE {
            firewall {
                name FROM-INSIDE-TO-OUTSIDE
            }
        }
        interface eth2
        interface eth3
    }
```

E de seguida, foram também criadas as regras mencionadas anteriormente como é possível observar nas capturas seguintes:

```
    name FROM-INSIDE-TO-OUTSIDE {
        default-action drop
        rule 10 {
            action accept
            destination {
                port 5000-6000
            }
            protocol udp
        }
    }
```

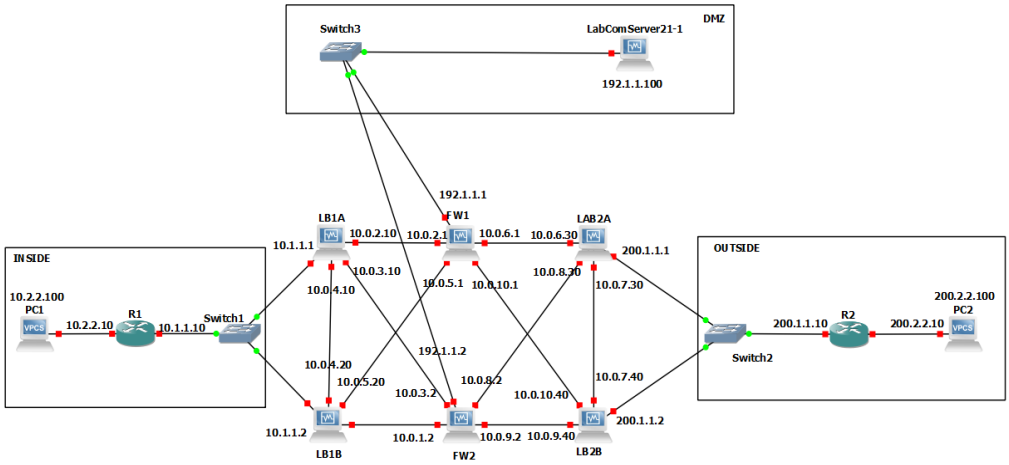
```
    name FROM-OUTSIDE-TO-INSIDE {
        default-action drop
        rule 10 {
            action accept
            state {
                established enable
            }
        }
    }
```

Por último, a configuração dos load balancers:

```
load-balancing {
    wan {
        disable-source-nat
        interface-health eth2 {
            failure-count 1
            nexthop 10.0.3.2
            success-count 1
        }
        interface-health eth3 {
            failure-count 1
            nexthop 10.0.2.1
            success-count 1
        }
        rule 1 {
            inbound-interface eth0
            interface eth2 {
                weight 1
            }
            interface eth3 {
                weight 1
            }
            protocol all
        }
```

Capítulo 3

Exercício 10



Configuração

Firewalls

Para este cenário, o servidor na zona DMZ suporta múltiplos serviços. Consequentemente, por questões de segurança, no firewall, foram criadas quatro novas regras:

1. DMZ-TO-INSIDE
2. DMZ-TO-OUTSIDE
3. OUTSIDE-TO-DMZ

4. INSIDE-TO-DMZ

A configuração atualizada das firewalls é apresentada nas capturas seguintes:

```
    name FROM-DMZ-TO-INSIDE {
        default-action drop
        rule 10 {
            action accept
            state {
                established enable
            }
        }
    }
    name FROM-DMZ-TO-OUTSIDE {
        default-action drop
        rule 10 {
            action accept
            state {
                established enable
            }
        }
    }
```

```
    name FROM-INSIDE-TO-DMZ {
        default-action drop
        rule 10 {
            action accept
            destination {
                port 22,443,21,53
            }
            protocol udp
        }
        rule 11 {
            action accept
            destination {
                port 22,443,21,53
            }
            protocol tcp
        }
    }
```

INSIDE-TO-DMZ são permitidas conexões para os portos 21,22,53 e 443 que são, usalmente, usadas para FTP,SSH,DNS e HTTPS respetivamente. Estas conexões tanto podem ser TCP ou UPD.

OUTSIDE-TO-DMZ são permitidos HTTPS e portos de serviços de domínio e igualmente estas conexões tanto podem ser TCP ou UDP.

```
name FROM-OUTSIDE-TO-DMZ {
    default-action drop
    rule 10 {
        action accept
        destination {
            port https,domain
        }
        protocol udp
    }
    rule 11 {
        action accept
        destination {
            port https,domain
        }
        protocol tcp
    }
}
```

Nas firewalls DMZ-TO-INSIDE ou DMZ-TO-OUTSIDE como o tráfego que chega já está filtrado e como o que server irá fazer será apenas responder a requests então as regras definidas são apenas respostas a sessões já definidas anteriormente.

De forma a comprovar a conexão com o servidor DMZ de seguida encontra-se uma captura realizada com origem em INSIDE:

```
PC1> ping 192.1.1.100 -P 17 -p 443
84 bytes from 192.1.1.100 udp_seq=1 ttl=61 time=14.022 ms
84 bytes from 192.1.1.100 udp_seq=2 ttl=61 time=13.833 ms
84 bytes from 192.1.1.100 udp_seq=3 ttl=61 time=16.678 ms
84 bytes from 192.1.1.100 udp_seq=4 ttl=61 time=17.306 ms
84 bytes from 192.1.1.100 udp_seq=5 ttl=61 time=17.733 ms
```

Adicionalmente, criámos um nova regra na firewall de forma a bloquear um IP específico, neste caso foi o ip 200.2.2.200. Este simula um ip que seria obtido através da monitorização da rede e que estaria a efetuar um ataque DDos.