

Redes de Computadores 2

Projeto

Elaborado por:

Bruno Palma

Diogo Patusca

Docente:

Armando Ventura

Pedro Moreira

João Tavanez

15-05-2024

Índice

1. Introdução	3
2. Topologia no GNS3.....	4
2.1. Configuração dos routers	5
3. Interface de Loopback.....	7
4. Configurar o OSPF	8
5. Pingar o IP da Google	9
6. Bloquear comunicações entre os Clientes	10
7. Aceder ao Telnet pelo Cliente 1	11
8. Port Knocking	12
9. Aceder às finanças e segurança social	13
10. Bloquear acesso á Internet no Cliente 1	14
11. Bloquear tráfego entre as Vlans.....	15
12. API	16
13. Conclusões.....	21

Lista de Figuras

Figura 1-Topologia no GNS3.....	4
Figura 2- Configuração das interfaces do router Cisco	5
Figura 3- Switch	6
Figura 4- Interface de Loopback1.....	7
Figura 5- Interface de Loopback no router Cisco	7
Figura 6- Rede OSPF no R1 da Mikrotik	8
Figura 7- OSPF no router Cisco.....	8
Figura 8- Ping 8.8.8.8 do Cliente1	9
Figura 9- Regras firewall para bloquear comunicações entre clientes	10
Figura 10- Telnet no router Cisco	11
Figura 11- Acesso ao Telnet	11
Figura 12- Regras firewall para o Port Knocking	12
Figura 13- Regras firewall para aceder aos sites permitidos	13
Figura 14- Firewall e Queue para aceder á Internet e limite de velocidade	14
Figura 15- Queue para limitar velocidade.....	14
Figura 16- Configuração ACL entre Vlans.....	15
Figura 17- Função show_interfaces	16
Figura 18- Função show_OSPF	17
Figura 19- Função show_firewall	18
Figura 20- Login no Router através do nosso programa	18
Figura 21- Ver as interfaces e os IPs.....	19

Figura 22- Redes da OSPF.....	19
Figura 23- Regras de Firewall	20

1. Introdução

No projeto da unidade curricular de Redes de Computadores 2, foi pedido para criar uma topologia no GNS3 com 2 routers da Mikrotik, 1 router da Cisco, 2 switches e 2 Clientes virtuais. Para implementar a topologia pedida foram estabelecidas várias configurações, desde a instalação dos aparelhos no GNS3 até à configuração de cada interface, OSPF e DHCP servers. Permitimos ainda que a máquina cliente aceda ao telnet do router R3, com a password rc2 no acesso ao telnet e no modo privilegiado.

Foram também criadas ACLs, que bloquearam a comunicação entre as máquinas cliente, a máquina do cliente2 só aceder á internet ao site do portal das finanças e da segurança social, entre as horas pretendidas, a máquina cliente 1 só aceder á internet no horário pretendido com limite de 1Mbits de velocidade e bloqueámos o tráfego entre Vlans, permitindo apenas através dos portos 80 e 443.

Implementámos também o mecanismo de Port Knocking no R1 de forma que para permitir acesso ao seu login necessita de receber um pacote TCP na porta 3000, depois receber um pedido TCP na porta 4000, depois por fim um pedido TCP na porta 5000, onde cada nível terá de ficar em lista 15 segundos e no final ficar disponível meia hora.

2. Topologia no GNS3

Para realizar a topologia na aplicação GNS3 na versão 2.2.46, começámos por criar as máquinas virtuais para os routers da Mikrotik, através do Oracle VM Virtual Box. Começámos por instalar o “cd image” do router MikrotikX86, disponível no site da Mikrotik. De seguida, criámos no Virtual Box uma máquina virtual com o sistema Linux e Outro Linux (32 bit), nas definições metemos o ficheiro “.iso” do router no disco, e nas definições de rede metemos 1 breach adapter e 2 controladores genéricos. Quando inicializámos a VM escolhemos a opção para instalar tudo, premindo a tecla “a”, e depois a tecla “i” para a instalação. Quando estava tudo instalado removemos o ficheiro do router do disco e inicializámos a VM. Para ser mais fácil identificar os routers, alteramos-lhes os nomes com o seguinte código: “system identity set name=NOME”, onde NOME deve ser substituído pelo nome a ser atribuído ao router, para o mesmo nome da figura da topologia fornecida, alterámos também a palavra-passe para “mikrotik” e repetimos o processo para o outro router com a alteração de nas definições de rede não ter o breach adapter, mas sim 2 controladores genéricos, foi também alterado o nome do router e alterada a palavra-passe para “mikrotik”.

No passo seguinte, decidimos instalar o router do cisco no GNS3, utilizando o ficheiro já fornecido pelo docente através do Moodle. No GNS3, fomos às preferências e criámos um IOS router template com a imagem descarregada anteriormente e com um slot NM-1FE-TX. As máquinas virtuais dos clientes, foram criadas com o Windows 10, de forma padrão, para serem os clientes da topologia. Com todas as máquinas virtuais criadas e o router cisco adicionado procedemos á criação da topologia como na figura seguinte.

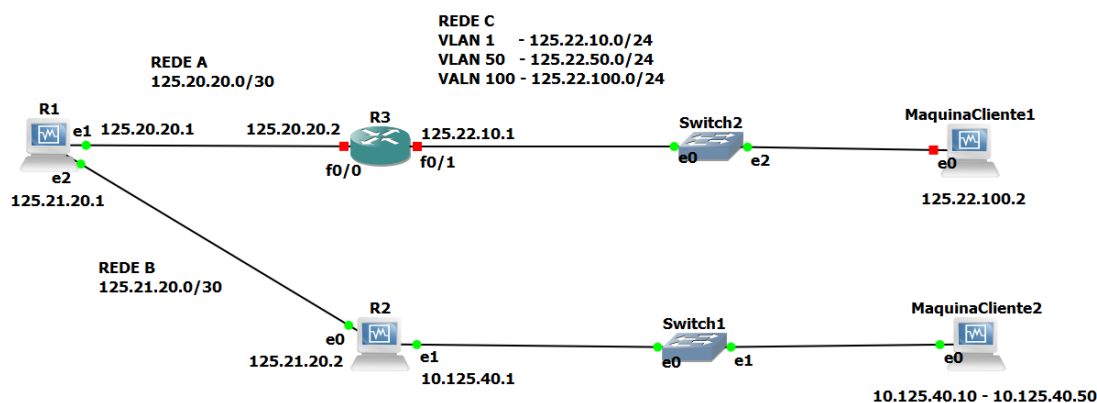


Figura 1-Topologia no GNS3

2.1. Configuração dos routers

O IP de cada rede para cada aluno foi atribuído através da soma do resto da divisão inteira do número de aluno por 200, sendo o resultado a atribuição de F, que no nosso caso, foi 125. Ficando assim com as seguintes redes:

- Rede A- 125.20.20.0/30
- Rede B- 125.21.20.0/30
- Rede C- VLAN1- 125.22.10.0/24
- VLAN50- 125.22.50.0/24
- VLAN100- 125.22.100.0/24

Para configurar o router1 da Mikrotik, começamos por aceder ao router através da aplicação WinBox versão 3.40, através do MAC Address do router. De seguida, acedemos á barra lateral IP -> DHCP Client e adicionámos um Client na interface ether1 com uma default route. Com isto, ao router foi atribuído um IP da mesma gama do portátil pessoal e já é possível pingar o IP 8.8.8.8(Google) através deste router porque já tem os dados necessários para aceder á internet (pode ser conferido em IP -> Routes, no WinBox). Como este router faz a ponte de toda a topologia para acesso á rede adicionámos o NAT, ou seja, a tradução dos IPs internos para o exterior, para realizarmos isto, acedemos a IP -> Firewall -> NAT, em Action seleccionámos a opção masquerade.

De seguida, adicionámos um IP da REDE A, á interface “ether2”, acedendo a IP -> Addresses, e adicionámos o IP 125.20.20.1/30, escolhendo a interface “ether2”. Na “ether3”, acedendo a IP -> Addresses, adicionámos o IP 125.21.20.1/30, pertencente á REDE B. De seguida configurámos o router2 do cisco, onde começámos por aceder a este router pelo console e utilizamos o seguinte código após enable -> configure terminal:

```
!
interface FastEthernet0/0
 ip address 125.20.20.2 255.255.255.252
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 125.22.10.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1.50
 encapsulation dot1Q 50
 ip address 125.22.50.1 255.255.255.0
!
interface FastEthernet0/1.100
 encapsulation dot1Q 100
 ip address 125.22.100.1 255.255.255.0
!
```

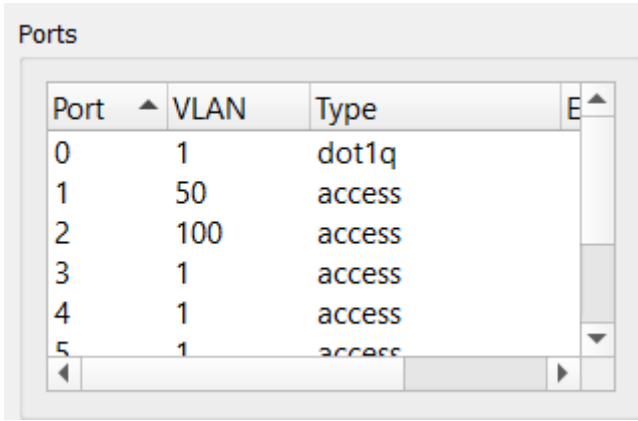
Figura 2- Configuração das interfaces do router Cisco

Após cada “ip address IP SUBNET_MASK”, escrevemos “no shutdown” para ligar a interface do router. No código apresentado em cima, adicionámos a cada interface o seu IP e a máscara de rede respetiva.

Para configurar o Router3 teve de ser pelo WinBox instalado na VM do cliente2, porque este cliente está ligado diretamente ao router, possibilitando assim aceder-lhe por MAC. No WinBox, acedemos a IP -> Addresses, adicionámos o IP 125.21.20.1/30 á interface “ether1” e um DHCP server que vai adicionar um IP no range 10.125.40.10/24-10.125.40.50/24 á interface “ether2”.

Para configurar os IPs nas máquinas virtuais dos clientes, neste momento foi feito manualmente nas definições de rede na máquina cliente 1 e automaticamente no cliente 2.

Após a configuração das interfaces principais, passámos às configurações do switch 2, onde metemos a porta 0 da Vlan 1 em dot1q, para que todas as Vlans comuniquem com o router através dessa porta, por fim alterámos as Vlans correspondentes nas portas 1 e 2 do switch.

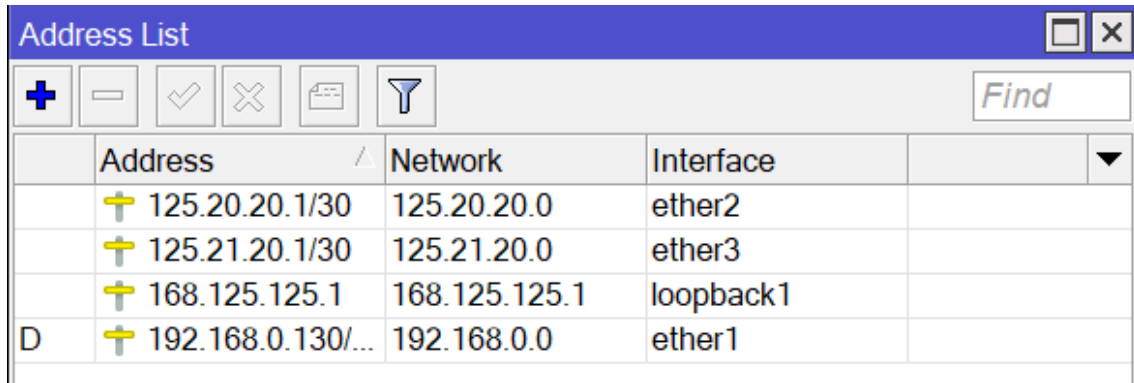


Port	VLAN	Type
0	1	dot1q
1	50	access
2	100	access
3	1	access
4	1	access
5	1	access

Figura 3- Switch

3. Interface de Loopback

Neste tópico do projeto, foi pedido para adicionar as interfaces de loopback para cada router, com gamas de IP á escolha. Nós escolhemos os IPs 168.125.125.N/32, onde N representa o número do router. Para adicionar a interface de loopback nos routers da Mikrotik, no WinBox fomos a Bridge e adicionámos uma interface com o nome LoopbackN, sendo N o número do router, e depois fomos a IP -> Addresses e adicionámos a interface loopback com o respetivo IP.



	Address	Network	Interface	
	125.20.20.1/30	125.20.20.0	ether2	
	125.21.20.1/30	125.21.20.0	ether3	
	168.125.125.1	168.125.125.1	loopback1	
D	192.168.0.130/...	192.168.0.0	ether1	

Figura 4- Interface de Loopback1

No router do cisco, entrámos no modo privilegiado e depois no modo de configuração global, onde escrevemos o seguinte código: interface Loopback3 -> "ip address 168.125.125.3 255.255.255.255" -> "no shutdown". Neste caso, não é obrigatório dizer "no shutdown", porque por defeito a interface fica automaticamente ligada.

```
interface Loopback3
ip address 168.125.125.3 255.255.255.255
!
```

Figura 5- Interface de Loopback no router Cisco

4. Configurar o OSPF

Nos routers da Mikrotik para adicionar as rotas OSPF, escolhemos ser através do WinBox, no R1 fomos a Routing -> OSPF -> Instances, adicionamos no router ID o mesmo IP da porta ether2, porque assim é garantido que este não vai ser repetido em mais lado nenhum e alteramos também o Redistribute Default Route para always (as type 2), para que ele distribua a rota por defeito para os seus “vizinhos”. Depois fomos a Networks e adicionamos as redes necessárias (125.20.20.0/30; 125.21.20.1/30; 168.125.125.1/32).

Para a configuração no R2, refizemos os passos do Winbox até aceder às “Instances”, nessa “Instance” adicionamos o IP 125.21.20.2 ao Router ID e mantivemos o Redistribute Default Route em never, pois o R1 já faz essa redistribuição. Depois em Networks adicionamos as redes necessárias (125.21.20.0/30; 10.125.40.0/24; 168.125.125.2/32).

Por fim, verificamos a área, de forma a ser igual em todas as configurações.

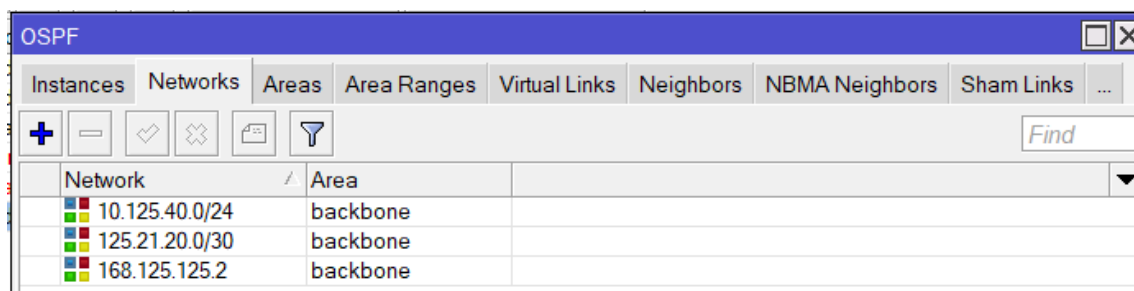


Figura 6- Rede OSPF no R1 da Mikrotik

No router do cisco, para adicionar as rotas do OSPF deve-se seguir os seguintes passos:

- enable
- configure terminal
- router ospf 10
- network IP WILDCARD_MASK area 0.0.0.0

A wildcard mask é calculada fazendo 255.255.255.255 – a máscara da rede do IP. Após realizar esta sequência de comandos para todas as redes, inclusive as redes das VLAN, ficamos com o OSPF configurado da seguinte maneira:

```
router ospf 10
log-adjacency-changes
network 125.20.20.0 0.0.0.3 area 0.0.0.0
network 125.22.0.0 0.0.255.255 area 0.0.0.0
network 168.125.125.3 0.0.0.0 area 0.0.0.0
```

Figura 7- OSPF no router Cisco

5. Pingar o IP da Google

De forma às máquinas virtuais pingarem o IP 8.8.8.8 neste momento, alterámos manualmente o IP atribuído ao cliente 1, a gateway e o DNS server nas definições de rede

De forma a entrar na página para alterar manualmente o IP, em Windows, deverá seguir os seguintes passos:

1. Carregar com o botão direito do rato no símbolo do Windows e ir a “Network Connections”
2. Aceder a “Change adapter options”
3. Carregar com o botão direito do rato na sua rede e carregar em “Properties”
4. Ir à opção “Internet Protocol Version 4(TCP/IPv4)”
5. Escolher a opção “Use the following IP address”
6. Configurar de acordo com o pretendido

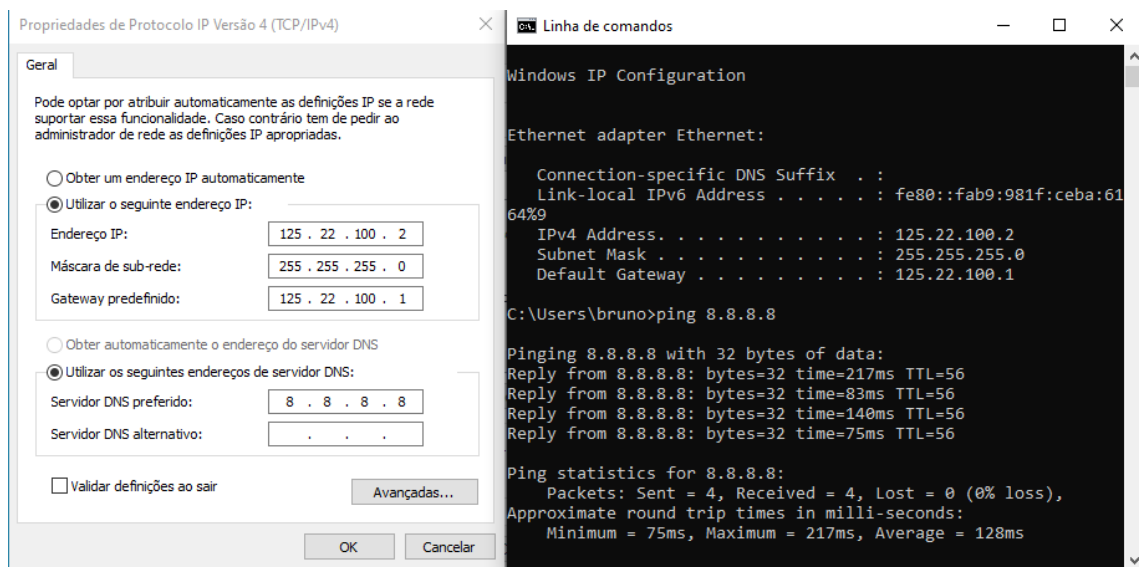


Figura 8- Ping 8.8.8.8 do Cliente1

Na máquina virtual do cliente 2, não é necessário realizar estes passos, uma vez que foi configurado um DHCP Server na ether2 do R2, então essas informações são obtidas automaticamente pelo computador.

6. Bloquear comunicações entre os Clientes

Nesta secção do relatório, iremos abordar a forma como bloqueámos as comunicações entre as máquinas virtuais “Cliente1” e “Cliente2”.

Esta regra da firewall, foi aplicada no R2 Mikrotik, uma vez que o IP da máquina 2 pode mudar devido ao DHCP Server, esta alteração do IP resultaria em que o IP da regra da firewall poderia não corresponder ao IP do Cliente, fazendo com que deixasse de bloquear as comunicações entre eles.

Para configurar esta regra de firewall, acedemos através do Winbox pela máquina do Cliente 2 e fomos a IP -> Firewall, e adicionámos as 2 regras pretendidas. A regra 0, dá drop de tudo o que tem destino á máquina do Cliente 1, já a regra 1 bloqueia todo o tráfego com origem da máquina do Cliente 1.

#	Action	Chain	Src. Address	Dst. Address	Proto...	Src. Port	Dst. Port	In. Inter...	Out. ▼
0	✖ drop	forward		125.22.100.2					
1	✖ drop	forward	125.22.100.2						

Figura 9- Regras firewall para bloquear comunicações entre clientes

7. Aceder ao Telnet pelo Cliente 1

Para realizar o pretendido nesta parte do trabalho, as regras foram aplicadas no R3 da cisco.

Para criar a ACL, foram introduzidos os seguintes comandos no router:

- enable
- configure terminal
- access-list 10 permit 125.22.100.2
- access-list 10 deny any

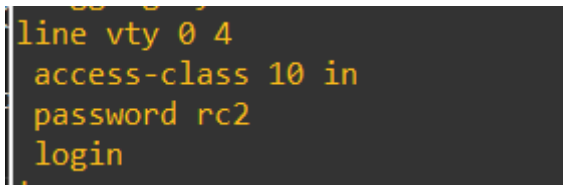
Esta ACL, permite que apenas o IP do Cliente 1, comunique com a interface em que a ACL está aplicada, bloqueando tudo o resto.

Para introduzir a password no modo privilegiado do router, foram aplicados os seguintes comandos:

- enable
- password rc2

Para configurar o acesso ao Telnet, foram aplicados os seguintes comandos:

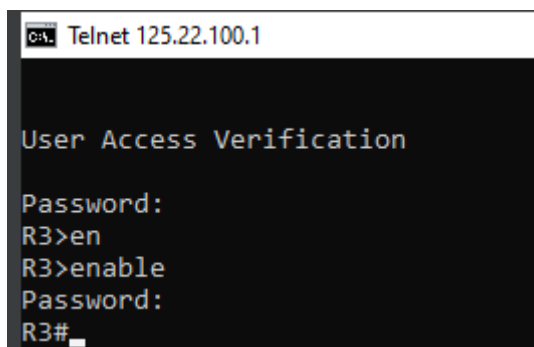
- enable
- configure terminal
- line vty 0 4
- password rc2
- login
- transport input telnet
- access-class 10 in



```
line vty 0 4
access-class 10 in
password rc2
login
```

Figura 10- Telnet no router Cisco

O comando “transport input telnet”, faz com que seja apenas permitido ligações ao router através do protocolo telnet



```
Telnet 125.22.100.1

User Access Verification

Password:
R3>en
R3>enable
Password:
R3#
```

Figura 11- Acesso ao Telnet

8. Port Knocking

Nesta parte do relatório, será explicado como foi implementado o mecanismo de Port Knocking no R1, de forma que para permitir acesso ao seu login necessita de receber um pacote TCP na porta 3000, depois receber um pedido TCP na porta 4000, depois por fim um pedido TCP na porta 5000, em cada nível de knock o source IP fica em lista durante 15 segundos e por fim o source IP tem permissão de efetuar login durante 30 minutos.

Ao inicio foram criadas 3 address lists, com os seguintes nomes:

- nivel1
- nivel2
- safe

Ao tentar aceder á porta 3000, o source IP é adicionada á lista Nivel1, depois se o IP já constar na lista Nivel1 e tentar aceder na porta 4000 é adicionado o IP á lista Nivel2, por fim se o IP estiver na lista Nivel2 e tentar aceder ao router pela porta 5000 é adicionado o IP á lista safe, se o IP estiver presente na lista safe é permitida a entrada no router.

Após a criação das address lists, procedemos á criação das regras da firewall, em IP -> Firewall adicionámos uma nova regra, regra 0, no menu “General” em “Chain” colocámos a opção “input”, na opção “Protocol” colocámos “6 (tcp)” e na “Dst. Port” colocámos “3000”, por fim em “Action” colocámos “add src to address list”, e na “Address List” colocámos “nivel1” com timeout 00:00:15.

Na regra 1, repetimos o processo realizado anteriormente, porém no menu “General” colocámos a “Dst. Port” como “4000”, no menu “Advanced” na opção “Src. Address List” colocámos “nivel1” e no menu “Action” colocámos “add src to address list”, e na “Address List” colocámos “nivel2” com timeout 00:00:15.

Para a regra 2, voltámos a repetir o processo feito anteriormente, onde no menu “General” colocámos a “Dst. Port” como “5000”, no menu “Advanced” na opção “Src. Address List” colocámos “nivel2” e no menu “Action” colocámos “add src to address list”, e na “Address List” colocámos “safe” com timeout 00:30:00.

A regra de firewall 3, serve para permitir o acesso ao router, para isso, no menu “General” na opção “Chain” colocámos “input” e no “Protocol” colocámos “6 (tcp)”, no menu “Advanced” na opção “Src. Address List” colocámos “safe” e no menu “Action” colocámos “accept”.

A última regra consiste na negação de tudo o resto, para isso, no menu “General” na opção “Chain” colocámos “input” e no “Protocol” colocámos “6 (tcp)” e no menu “Action” colocámos “drop”.

#	Action	Chain	Src. Address	Dst. Address	Proto...	Src. Port	Dst. Port	In. Interf...	Out. Inte...	In. Interf...	Out. Inte...	Src. Ad...	Dst. Ad...	Bytes	Packets	
0	add ...	input			6 (tcp)		3000							208 B	4	
1	add ...	input			6 (tcp)		4000					nivel1		208 B	4	
2	add ...	input			6 (tcp)		5000					nivel2		260 B	5	
3	acc...	input			6 (tcp)							safe		21.1 KiB	123	
4	drop	input			6 (tcp)									416 B	8	

Figura 12- Regras firewall para o Port Knocking

9. Aceder às finanças e segurança social

Nesta parte do relatório, vamos abordar como foram criadas as regras da firewall para a máquina virtual do Cliente 2 só poder aceder ao exterior à Internet apenas ao site do portal das finanças e da segurança social entre as 9:00 e as 12:00 e das 14:00 às 17:00, onde tudo o resto é negado a qualquer hora.

Como era necessário ter em consideração a hora do dia para a realização de exercício, começámos por conferir se o horário do router coincidia com o horário da máquina virtual, em System -> Clock.

Depois de verificar o horário do router, começámos a criar a address list, com o nome “sitespermitidos”, onde adicionámos os sites das finanças e da segurança social. Já com o horário correto e a address list criada, fomos adicionar as regras á firewall. Em IP -> Firewall, adicionámos uma nova regra, no menu “General” na opção “Chain” colocámos a opção “forward”, no menu “Advanced” em “Dst. Address List” colocámos o nome da lista(“sitespermitidos”), e por fim no menu “Extra” na opção “Time” seleccionámos todos os dias e adicionámos o horário pretendido, neste caso, das 09:00:00 às 12:00:00. Na regra seguinte, repetimos todo o processo, mas alterámos o horário para das 14:00:00 às 17:00:00. Por fim, na última regra bloqueámos o acesso a tudo o resto que não fosse permitido pelas regras acima mencionadas.

--- inactive time									
2		acc...	forward						sitesper...
--- inactive time									
3		acc...	forward						sitesper... 254.1
4		drop	forward		6 (tcp)	80			5.3
5		drop	forward		6 (tcp)	443			125.7

Figura 13- Regras firewall para aceder aos sites permitidos

10. Bloquear acesso á Internet no Cliente 1

Nesta parte do relatório, iremos abordar a forma de como fizemos para que a máquina cliente 1 só possa aceder à Internet a qualquer site entre as 17 horas até às 23 horas de segunda a sexta-feira, com velocidade permitida até 1 Mbits down e up.

Como era necessário ter em consideração a hora do dia para a realização de exercício, começámos por conferir se o horário do router coincidia com o horário da máquina virtual, em System -> Clock.

Depois de verificar o horário do router, começámos a criar as regras de firewall.

Em IP -> Firewall, adicionámos uma nova regra, no menu “General” na opção “Chain” colocámos a opção “forward”, na opção “Src. Address” colocámos o IP do Cliente 1(125.22.100.2), na opção “Protocol” colocámos “6 (tcp)” e na “Dst. Port” colocámos “80”, no menu “Extra” na opção “Time” seleccionámos os dias de segunda a sexta e adicionámos o horário pretendido, neste caso, das 17:00:00 às 23:00:00, por fim no menu “Action” colocámos a “Action” como “accept”. Na regra seguinte, repetimos todo o processo, mas alterámos a “Dst. Port” onde colocámos “443”.

Para bloquear as comunicações, o processo foi igual ao referido anteriormente, com a diferença que no menu “Action”, colocámos a opção “Action” como “drop” e também foram removidas as horas e os dias, uma vez que, vai bloquear sempre.

5	acc...	forward	125.22.100.2	6 (tcp)	80					0 B	0
6	drop	forward	125.22.100.2	6 (tcp)	80					73.4 KiB	1 446
7	acc...	forward	125.22.100.2	6 (tcp)	443					317.4 KiB	1 675
8	drop	forward	125.22.100.2	6 (tcp)	443					258.0 KiB	5 098

Queue List

Simple Queues

Interface Queues

Queue Tree

Queue Types

Reset Counters

Reset All Counters

Find

#	Name	Target	Upload Max Limit	Download Max Limit	Packet Marks	Total Max Limit (bit...
0	Cliente1	125.22.100.2	1M	1M		

Figura 14- Firewall e Queue para aceder á Internet e limite de velocidade

Para limitar a velocidade de download e upload para 1Mbit acedemos a Queues e adicionámos uma “Simple Queue”, colocámos o IP da máquina do cliente 2(125.22.100.2) em “Target”, e alterámos o limite do Target Upload e do Target Download para “1M”. Por fim, na opção “Time” colocámos as horas e os dias em que esta fila vai ser aplicada.

Simple Queue <Cliente1>

General

Advanced

Statistics

Traffic

Total

Total Statistics

Name: Cliente1

Target: 125.22.100.2

Dst:

Target Upload

Target Download

Max Limit: 1M

1M

bits/s

Burst

Burst Limit: unlimited

unlimited

bits/s

Burst Threshold: unlimited

unlimited

bits/s

Burst Time: 0

0

s

Time

Time: 17:00:00 - 23:00:00

Days: ☐ sun ☒ mon ☒ tue ☒ wed ☒ thu ☒ fri ☐ sat

enabled

OK

Cancel

Apply

Disable

Comment

Copy

Remove

Reset Counters

Reset All Counters

Torch

Figura 15- Queue para limitar velocidade

11. Bloquear tráfego entre as Vlans

Para bloquear o tráfego entre as Vlans da rede C, começamos por criar as ACLs, no R3 do cisco:

```
access-list 101 permit tcp any any eq www
access-list 101 permit tcp any any eq 443
access-list 101 permit ip 125.22.10.0 0.0.0.255 125.22.10.0 0.0.0.255
access-list 101 deny ip 125.22.10.0 0.0.0.255 125.22.0.0 0.0.255.255
access-list 101 permit ip any any
access-list 102 permit tcp any any eq www
access-list 102 permit tcp any any eq 443
access-list 102 permit ip 125.22.50.0 0.0.0.255 125.22.50.0 0.0.0.255
access-list 102 deny ip 125.22.50.0 0.0.0.255 125.22.0.0 0.0.255.255
access-list 102 permit ip any any
access-list 103 permit tcp any any eq www
access-list 103 permit tcp any any eq 443
access-list 103 permit ip 125.22.100.0 0.0.0.255 125.22.100.0 0.0.0.255
access-list 103 deny ip 125.22.100.0 0.0.0.255 125.22.0.0 0.0.255.255
access-list 103 permit ip any any
```

Figura 16- Configuração ACL entre Vlans

A ACL 101 começa por permitir qualquer tráfego pelas portas 80 e 443, na regra 3 permite o tráfego entre os Pcs dentro da mesma Vlan, na regra 4 bloqueamos o tráfego para as outras Vlans, na última regra permitimos o tráfego para o exterior.

A ACL 102 começa por permitir qualquer tráfego pelas portas 80 e 443, na regra 3 permite o tráfego entre os Pcs dentro da mesma Vlan, na regra 4 bloqueamos o tráfego para as outras Vlans, na última regra permitimos o tráfego para o exterior.

A ACL 103 começa por permitir qualquer tráfego pelas portas 80 e 443, na regra 3 permite o tráfego entre os Pcs dentro da mesma Vlan, na regra 4 bloqueamos o tráfego para as outras Vlans, na última regra permitimos o tráfego para o exterior.

12. API

Na parte final do projeto, foi pedida a conexão aos routers da Mikrotik da topologia através da sua API utilizando uma linguagem de programação à escolha do aluno, no nosso caso escolhemos utilizar Python.

Utilizando várias bibliotecas disponibilizadas no python, desenvolvemos um programa que permite conectar ao router através do IP, username e password, é possível também visualizar as interfaces e os respetivos IPs, as redes do routing OSPF e as regras da firewall aplicadas.

No código python criámos funções que passamos a explicar:

Essa função mostra as interfaces do router, depois de conectar ao router essa conexão é guardada na variável API e podemos passar comandos para essa variável.

Os comandos são o caminho que é necessário fazer no winbox, nesse caso o comando que utilizamos foi `/ip/route/print`, que devolve tudo que aparece no winbox nessa seção, como só queremos retirar algumas informações como o nome da interface, IP e network, o que fizemos foi um loop for que percorre a informação devolvida pelo comando e selecionamos o que queremos em cada interface.

```
def show_interfaces():
    try:
        command = '/ip/route/print'
        results = list(api(cmd=command))
        filtered_results = []

        count = 0
        for result in results:
            if count != 0:
                filtered_result = {
                    'Interface': result.get('gateway'),
                    'IP da Interface': result.get('pref-src'),
                    'Network': result.get('dst-address')
                }
                filtered_results.append(filtered_result)

            count += 1

        # Formatar os resultados para exibição
        display_text = "Interface \t\t IP da Interface \t\t Network \n"
        for res in filtered_results:
            display_text += f"{res['Interface']} \t\t {res['IP da Interface']} \t\t {res['Network']}\n"

        # Exibir os resultados na caixa de texto
        result_text.config(state=tk.NORMAL) # Habilita a edição temporariamente para inserir texto
        result_text.delete(1.0, tk.END) # Limpa o texto anterior
        result_text.insert(tk.END, display_text)
        result_text.config(state=tk.DISABLED) # Define a caixa de texto como somente leitura novamente

    except Exception as e:
        messagebox.showerror("Error", f"Failed to retrieve interfaces: {e}")
```

Figura 17- Função show_interfaces

De maneira semelhante, criamos uma função que vai mostrar as opções pretendidas sobre o OSPF do router. Os comandos são o caminho que é necessário fazer no winbox, nesse caso o comando que utilizamos foi `/routing/ospf/instance/print`, que devolve tudo que aparece no winbox nessa seção, como só queremos retirar algumas informações como as redes do OSPF, o que fizemos foi um loop for que percorre a informação devolvida pelo comando e selecionamos o que mostrar como o nome da instância, router id e o tipo de distribuição do router, seguido das redes ligadas.

```
def show_ospf():
    try:
        command_instance = '/routing/ospf/instance/print'
        results_instance = list(api(cmd=command_instance))

        display_text = "OSPF Configuration:\n"

        for instance in results_instance:
            instance_name = instance.get('name')
            router_id = instance.get('router-id')
            distribution = instance.get('distribute-default')

            display_text += f"\nInstance: {instance_name}, Router ID: {router_id}, Distribution: {distribution}\n"

            # Comando para obter as redes associadas à instância OSPF
            command_network = f'/routing/ospf/network/print'
            results_network = list(api(cmd=command_network))

            # Adiciona as redes à exibição
            for network in results_network:
                network_address = network.get('network')
                display_text += f"\tNetwork: {network_address}\n"

        # Exibir os resultados na caixa de texto
        result_text.config(state=tk.NORMAL) # Habilita a edição temporariamente para inserir texto
        result_text.delete(1.0, tk.END) # Limpa o texto anterior
        result_text.insert(tk.END, display_text)
        result_text.config(state=tk.DISABLED) # Define a caixa de texto como somente leitura novamente

    except Exception as e:
        messagebox.showerror("Error", f"Failed to retrieve OSPF configuration: {e}")
```

Figura 18- Função show_OSPF

Na seguinte função, vamos conseguir obter as informações relativas às regras de firewall do router, neste caso o comando que utilizamos foi `/ip/firewall/filter/print`. A recolha de informação através da API foi feita de forma semelhante, através do loop `for` que percorre a informação devolvida pelo comando e selecionamos o que mostrar como o “chain”, “action”, “src address” e “dst address”.

```
def show_firewall():
    try:
        command = '/ip/firewall/filter/print'
        results = list(api(cmd=command))
        filtered_results = []

        for result in results:
            filtered_result = {
                'Chain': result.get('chain'),
                'Action': result.get('action'),
                'Src Address': result.get('src-address'),
                'Dst Address': result.get('dst-address')
            }
            filtered_results.append(filtered_result)

        # Formatar os resultados para exibição
        display_text = "Firewall Configuration:\n"
        for res in filtered_results:
            display_text += f"\n Chain: {res['Chain']}, Action: {res['Action']}, Src Address: {res['Src Address']}, Dst Address: {res['Dst Address']}\n"

        # Exibir os resultados na caixa de texto
        result_text.config(state=tk.NORMAL) # Habilita a edição temporariamente para inserir texto
        result_text.delete(1.0, tk.END) # Limpa o texto anterior
        result_text.insert(tk.END, display_text)
        result_text.config(state=tk.DISABLED) # Define a caixa de texto como somente leitura novamente

    except Exception as e:
        messagebox.showerror("Error", f"Failed to retrieve firewall configuration: {e}")
```

Figura 19- Função `show_firewall`

De seguida, vamos mostrar imagens da interface:

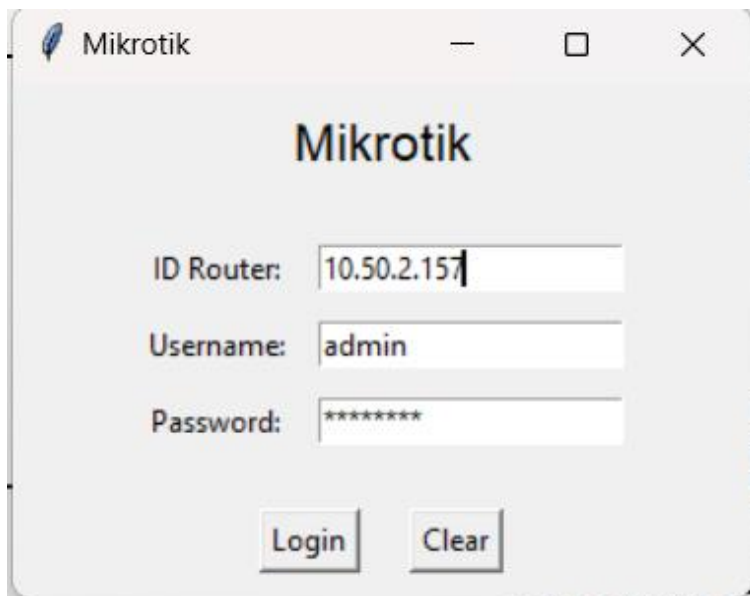


Figura 20- Login no Router através do nosso programa

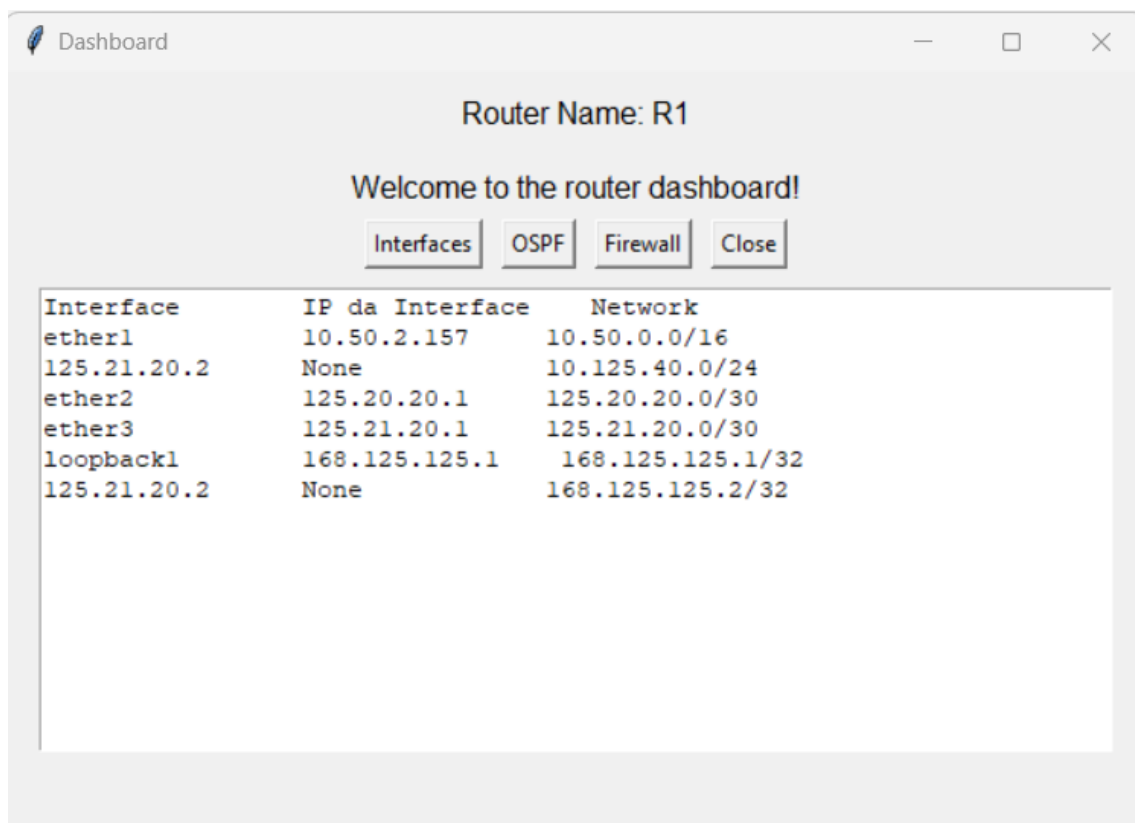


Figura 21- Ver as interfaces e os IPs

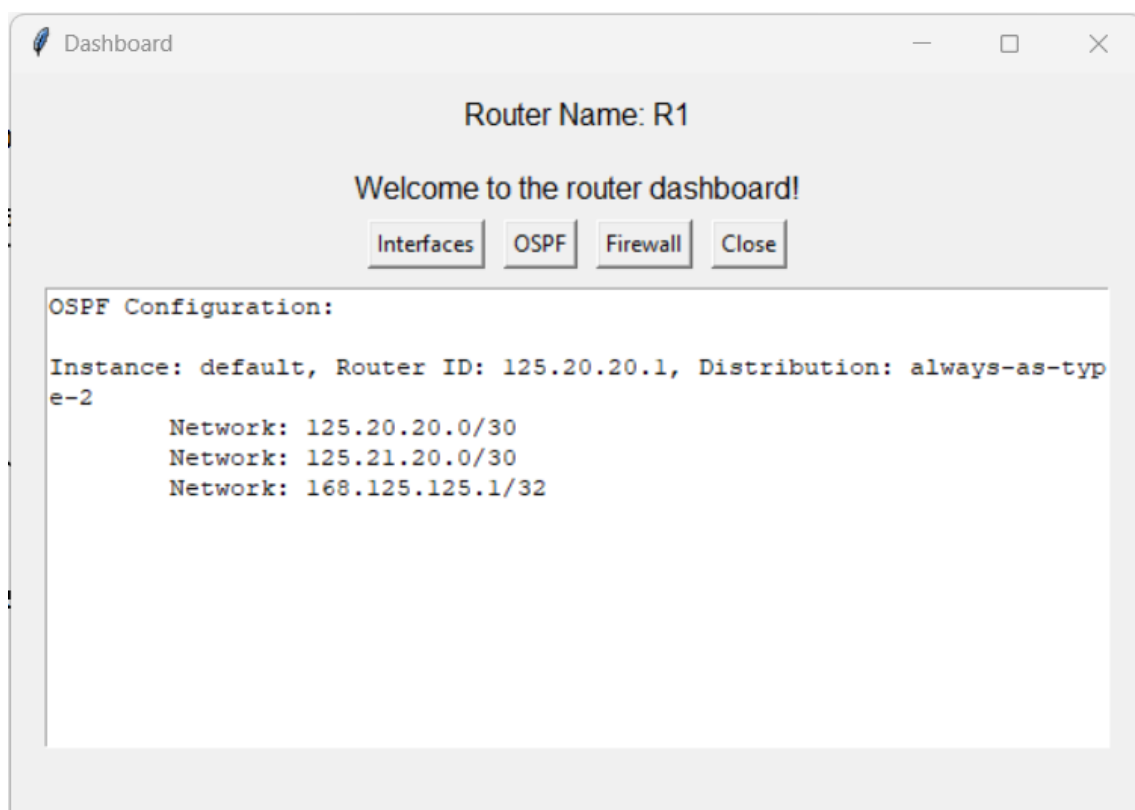


Figura 22- Redes da OSPF

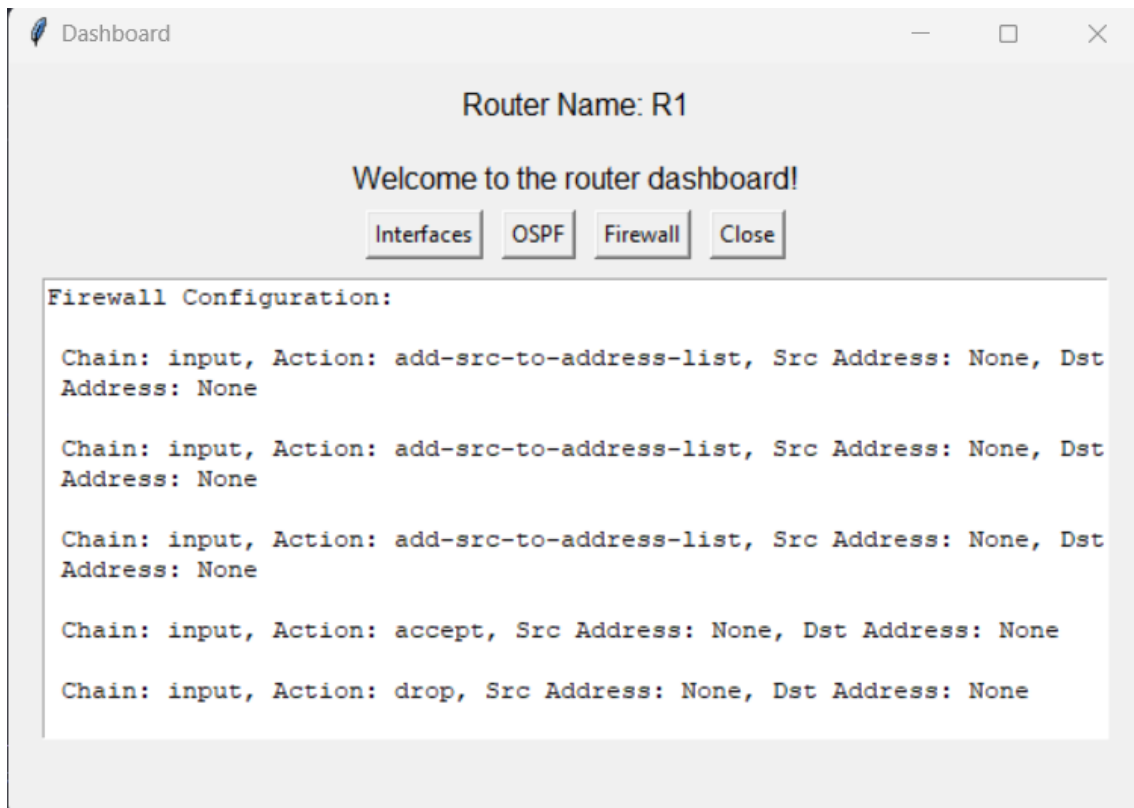


Figura 23- Regras de Firewall

13. Conclusões

Neste trabalho foi possível adquirir melhor os conhecimentos sobre a parte teórica abordada em contexto de sala de aula.

As principais dificuldades encontradas ao longo do trabalho, foram na utilização do GNS e o facto de ao realizar os exercícios na escola o DNS não funcionar corretamente.

Todos os desafios extra que foram aparecendo ao longo da realização do trabalho foram superados com sucesso, através de alguma pesquisa e de debugging.

Todos os requisitos pedidos no trabalho foram realizados com sucesso e devidamente testados.

Conseguimos configurar a topologia de rede solicitada, implementando regras de firewall, ACLs, port knocking e limite de velocidade na internet, utilizando os routers Mikrotik e Cisco, também desenvolvemos uma aplicação em python, enviada junto com o relatório, para a interação com os routers.

Bibliografia

https://cms.ipbeja.pt/pluginfile.php/364778/mod_resource/content/8/Ficha%20PortKnocking%20-%20ProfArmandoVentura.pdf

https://cms.ipbeja.pt/pluginfile.php/241366/mod_resource/content/5/2%20-%20Mikrotik%20Firewall.pdf

https://cms.ipbeja.pt/pluginfile.php/342048/mod_resource/content/2/Apresenta%C3%A7%C3%A3o_ACL.pdf

<https://mikrotik.com/download>

<https://www.youtube.com/watch?v=ENV--A0khNE>