EA080 O - Atividade 4

Gustavo Ciotto Pinton RA 117136

Exercício 0

A primeira fase involve a instalação da topologia de rede especificada na figura 1 do enunciado. As configurações do roteador R_{21} estão representadas nas figuras 1 e 2. A primeira ilustra os endereços das interfaces habilitadas e a segunda, a tabela de roteamento com as rotas estáticas ajustadas. Destaca-se que o *gateway default* é a interface *eth2* de R_{22}.

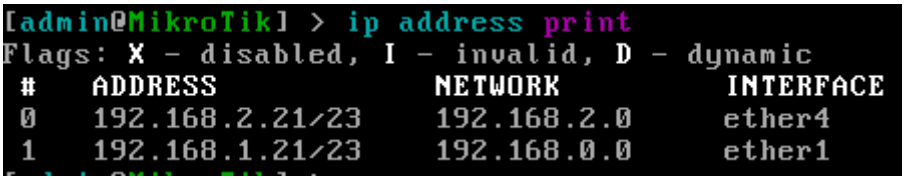

Figura 1: Endereços das interfaces habilitadas em R_{21}

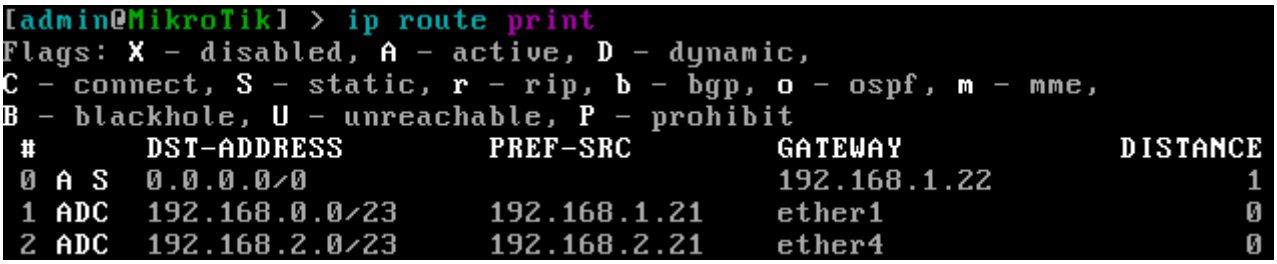

Figura 2: Tabela de roteamento de R_{21}

A configuração de R_{22}, por sua vez, está representada nas figuras 3 e 4. Duas rotas estáticas foram particularmente configuradas: uma para a sub-rede *10.0.0.0/23* e a outra para *192.168.2.0/23*.

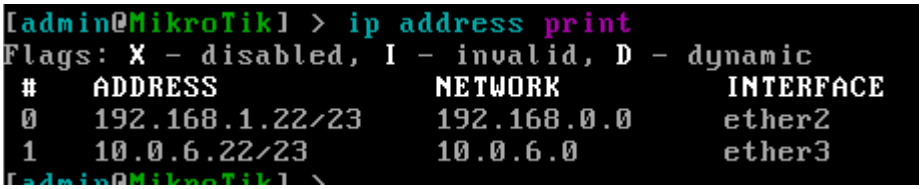

Figura 3: Endereços das interfaces habilitadas em R_{22}

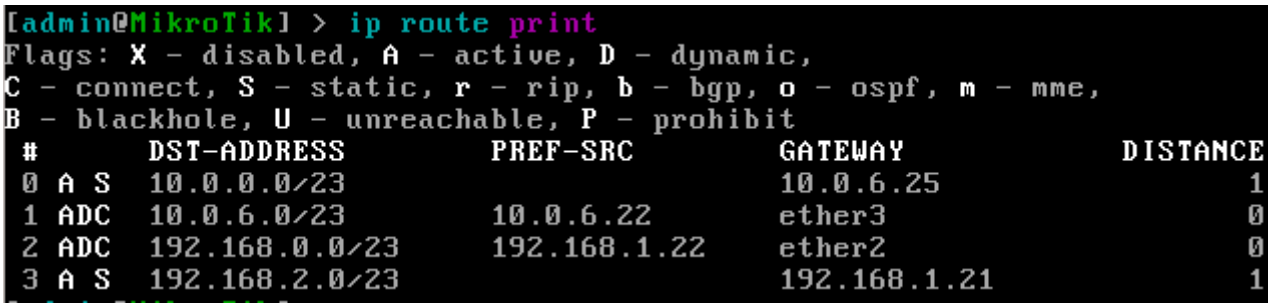

Figura 4: Tabela de roteamento de R_{22}

A configuração de R_{23} está presente nas figuras 5 e 6. A rota *default* é a interface *eth2* de R_{24}. Este roteador foi configurado de maneira semelhante ao R_{21}.

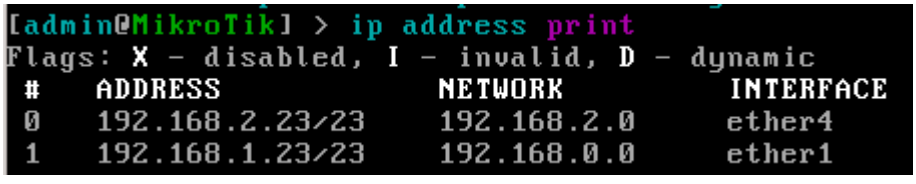

Figura 5: Endereços das interfaces habilitadas em R_{23}

```
[admin@MikroTik] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 A S  0.0.0.0/0                          192.168.1.24              1
 1 ADC  192.168.0.0/23     192.168.1.23    ether1                    0
 2 ADC  192.168.2.0/23     192.168.2.23    ether4                    0
```

Figura 6: Tabela de roteamento de R_{23}

R_{24}, análogo de R_{22}, foi configurado conforme figuras 7 e 8.

```
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   10.0.1.24/23       10.0.0.0        ether5
 1   192.168.1.24/23    192.168.0.0     ether2
```

Figura 7: Endereços das interfaces habilitadas em R_{24}

```
[admin@MikroTik] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 ADC  10.0.0.0/23        10.0.1.24       ether5                    0
 1 A S  10.0.6.0/23                        10.0.1.25                 1
 2 ADC  192.168.0.0/23     192.168.1.24    ether2                    0
 3 A S  192.168.2.0/23                     192.168.1.23              1
```

Figura 8: Tabela de roteamento de R_{24}

O último roteador, R_{25}, é configurado conforme figuras 9 e 10.

```
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   10.0.1.25/23       10.0.0.0        ether5
 1   10.0.6.25/23       10.0.6.0        ether2
```

Figura 9: Endereços das interfaces habilitadas em R_{25}

```
[admin@MikroTik] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 A S  0.0.0.0/0                          10.0.6.22                 1
 1 ADC  10.0.0.0/23        10.0.1.25       ether5                    0
 2 ADC  10.0.6.0/23        10.0.6.25       ether2                    0
```

Figura 10: Tabela de roteamento de R_{25}

Enfim, os *hosts PCx*, de endereço *192.168.2.190*, e *PCy,* , de endereço *192.168.2.101*, estão configurados conforme figuras 11 e 12, respectivamente.

```
root@maumagal-VirtualBox:/home/ea080# ifconfig
eth2      Link encap:Ethernet  HWaddr 08:00:27:96:07:e0
          inet addr:192.168.2.190  Bcast:192.168.3.255  Mask:255.255.254.0
          inet6 addr: fe80::a00:27ff:fe96:7e0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:958 errors:0 dropped:0 overruns:0 frame:0
          TX packets:275 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:271167 (271.1 KB)  TX bytes:45157 (45.1 KB)
```

Figura 11: Configuração do *host* PC_x.

```
root@slitaz:/home/tux# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:3D:AF:9D
          inet addr:192.168.2.101  Bcast:192.168.3.255  Mask:255.255.254.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:123 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1205 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16993 (16.5 KiB)  TX bytes:379666 (370.7 KiB)
```

Figura 12: Configuração do *host* PC_y.

Exercício 1

A fim de permitir a tradução do endereço privado do PC_x (*192.168.2.190*), para o endereço publico *10.0.6.22/23*, adicionou-se ao roteador presente na borda da rede, isto é, R_{22}, uma regra NAT do tipo *srcnat*. Este tipo de regra altera o endereço dos pacotes cuja origem é a rede *privada*. Logo, tais pacotes terão seus endereços de origem substituídos por um endereço público à medida que passam por R_{22}. A regra criada pode ser visualizada na figura 13. Seu número de identificação é 0. Desconsiderar momentaneamente a regra cujo número é 1.

```
[admin@Router-22] > ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=srcnat action=src-nat to-addresses=10.0.6.22
      src-address=192.168.2.190 log=no log-prefix=""

 1    chain=dstnat action=dst-nat to-addresses=192.168.2.190
      dst-address=10.0.6.22 log=no log-prefix=""
```

Figura 13: Regras NAT adicionadas ao roteador R_{22}.

Uma vez adicionada essa regra, o comando *ping* entre as máquinas PC_x e PC_z, de endereço *10.0.0.200/23*, torna-se possível. As figuras 14 e 15, mostram, respectivamente os resultados dos *pings* de PC_x para PC_z e de PC_z para PC_x.

```
ea080@maumagal-VirtualBox:~$ ping 10.0.0.200 -c 5
PING 10.0.0.200 (10.0.0.200) 56(84) bytes of data.
64 bytes from 10.0.0.200: icmp_req=1 ttl=125 time=1.60 ms
64 bytes from 10.0.0.200: icmp_req=2 ttl=125 time=1.81 ms
64 bytes from 10.0.0.200: icmp_req=3 ttl=125 time=1.74 ms
64 bytes from 10.0.0.200: icmp_req=4 ttl=125 time=1.51 ms
64 bytes from 10.0.0.200: icmp_req=5 ttl=125 time=1.06 ms

--- 10.0.0.200 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 1.062/1.547/1.813/0.264 ms
```

Figura 14: Comando *ping* de PC_x a PC_z.

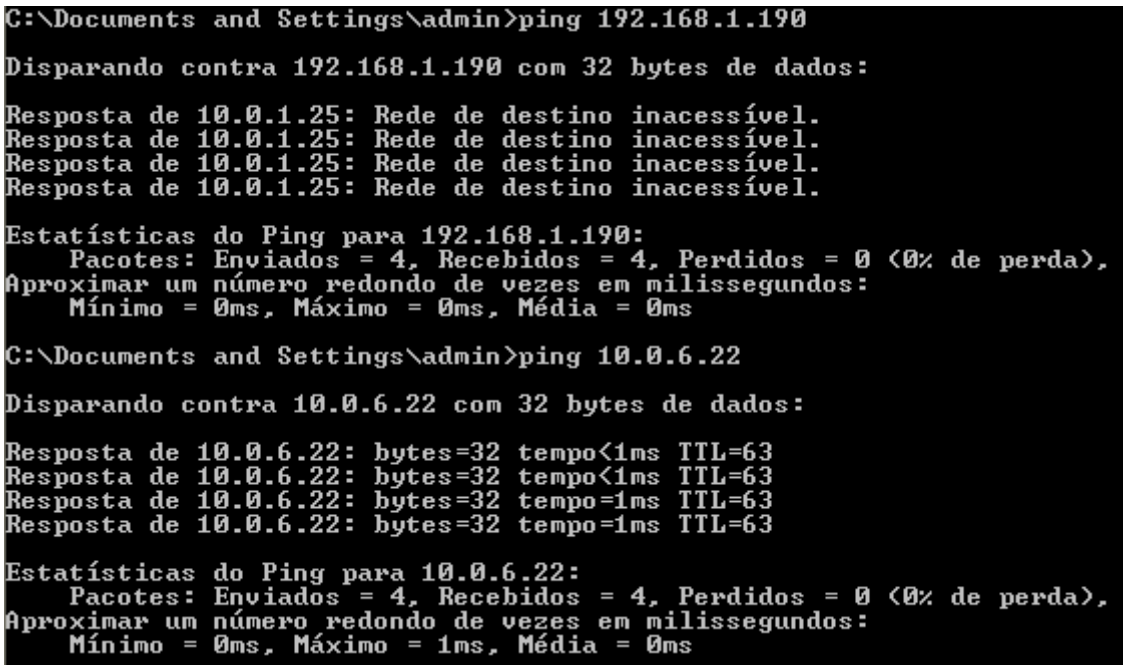

Figura 15: Comando *ping* de PC_z a PC_x.

Na figura 14, observa-se que o valor de TTL dos pacotes enviados vale 125, equivalendo ao fato de que os pacotes passaram por três roteadores no seu percurso. Este fato é confirmado pela topologia de rede proposta na figura 1 do enunciado: os pacotes passam por R_{21}, R_{22} e R_{25}.

Na figura 15, nota-se, pelo comando *ping* na parte superior da imagem, que o *host* PC_x é inacessível através do seu endereço privado, isto é, *192.168.2.190*, conforme esperado. O *ping* na parte inferior da imagem, apesar de bem-sucedido, não corresponde a um *ping* entre os dois *hosts*, já que o TTL vale 63, equivalendo ao fato de que somente um roteador pertence ao percurso. Quando executamos o *ping* para 10.0.6.22 do *host* PC_z, estamos efetivamente acessando a interface *eth3* de R_{22} e não Pc_x, uma vez que não há regras NAT configuradas no que se diz respeito a pacotes destinados à rede privada. Sendo assim, quando adicionamos a regra, cuja identificação 1 da figura 13, de tipo *dstnat,* certificamo-nos que o endereço de destino dos pacotes serão modificados de 10.0.6.22 para 192.168.2.190 ao passar por R_{22}.

A figura 16 mostra o resultado do *ping* de Pc_z para Pc_x com as *duas* regras ativadas. Destaca-se que o valor de TTL, que valia 63 anteriormente, muda para 61 após a ativação desta segunda regra. Isto indica que mais 2 roteadores participam do percurso, equivalendo, assim, a R_{22} e R_{21}.

Figura 16: Comando *ping* de PC_z a Pc_x antes e depois da ativação da segunda regra.

Exercício 2

A fim de acessarmos também o endereço privado 192.168.2.21 sem utilizarmos *masquerade*, atribui-se um novo endereço à interface *eth3* de R_{22} e adiciona-se uma nova regra do tipo *dstnat* em R_{22}, utilizando este novo endereço como *dst-address*. A figura 17 apresenta os endereços das interfaces de R_{22}. Destaca-se que *eth3* pode ser acessado também através de *10.0.6.222*.

```
[admin@Router-22] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   192.168.1.22/23    192.168.0.0     ether2
 1   10.0.6.22/23       10.0.6.0        ether3
 2   10.0.6.222/23      10.0.6.0        ether3
```

Figura 17: Endereços das interface do roteador R_{22}.

A figura 18 apresenta todas as regras ativas em R_{22}. Além das 2 do exercício anterior, há uma outra regra, de tipo *dstnat*, responsável por modificar os endereços de destino de *10.0.6.222* para *192.168.2.21* dos pacotes que passam por R_{22}.

```
[admin@Router-22] > ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=srcnat action=src-nat to-addresses=10.0.6.22
      src-address=192.168.2.190 log=no log-prefix=""

 1    chain=dstnat action=dst-nat to-addresses=192.168.2.190
      dst-address=10.0.6.22 log=no log-prefix=""

 2    chain=dstnat action=dst-nat to-addresses=192.168.1.21
      dst-address=10.0.6.222 log=no log-prefix=""
```

Figura 18: Regras NAT adicionadas ao roteador R_{22}.
Destaque à regra n.2, responsável por permitir a tradução dos endereços de destino de 10.0.6.222 para 192.168.1.21.

Enfim, o comando *ping* de Pc_z para *192.168.1.21* está representado na figura 19. O valor de TTL vale 62, indicando que dois roteadores participaram do percurso (R_{22} e R_{25}) e, consequentemente, que os endereços foram corretamente traduzidos.

```
C:\Documents and Settings\admin>ping 10.0.6.222 -n 5

Disparando contra 10.0.6.222 com 32 bytes de dados:

Resposta de 10.0.6.222: bytes=32 tempo=1ms TTL=62
Resposta de 10.0.6.222: bytes=32 tempo=1ms TTL=62
Resposta de 10.0.6.222: bytes=32 tempo=1ms TTL=62
Resposta de 10.0.6.222: bytes=32 tempo<1ms TTL=62
Resposta de 10.0.6.222: bytes=32 tempo=1ms TTL=62

Estatísticas do Ping para 10.0.6.222:
    Pacotes: Enviados = 5, Recebidos = 5, Perdidos = 0 (0% de perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 0ms, Máximo = 1ms, Média = 0ms
```

Figura 19: Comando *ping* de PC_z a *10.0.6.222*.

Exercício 3

Nesta etapa, deseja-se que todos os endereços da rede privada sejam traduzidos para um único endereço público. Para tal, utiliza-se uma regra do tipo *masquerade* em R_{22}, conforme figura 20. Observa-se que este tipo de regra é semelhante às regras *srcnat*, à medida que pacotes originados da rede privada têm seu endereço de origem modicado.

```
[admin@Router-22] > ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=srcnat action=masquerade out-interface=ether3 log=no log-prefix=""
```

Figura 20: Regra *masquerade* adicionada ao roteador R_{22}.

(a) Os pings de PC_z para os endereços *192.168.0.190* e *192.168.1.21* não são bem-sucedidos visto que não há nenhuma regra do tipo *destination NAT.* Dessa forma, o roteador R_{22} não sabe como traduzir os endereços dos pacotes de destino à rede privada e, portanto, Pc_z não consegue acessar nem R_{21} e nem PC_x. O resultado do *ping* de Pc_z à 10.0.6.22 é semelhante a aquele apresentado na parte superior da figura 16. O TTL dos pacotes vale 63, indicando que, na verdade, acessamos a *eth3* de R_{22}.

(b) Contrariamente ao item anterior, o roteador R_{22} sabe como traduzir endereços dos pacotes *de origem* da rede privada, através da regra *masquerade*. Para este item, ativa-se o *Wireshark* em ambos os *hosts* e executa-se *pings* entre eles.

O comando *ping* de PC_x para PC_z produz os seguintes resultados no tráfego de rede, capturados pelo *Wireshark* em PC_z.

```
10.0.6.22     10.0.0.200    ICMP    98 Echo (ping) request  id=0x41d3, seq=3/768, ttl=61 (reply in 13)
10.0.0.200    10.0.6.22     ICMP    98 Echo (ping) reply    id=0x41d3, seq=3/768, ttl=128 (request in 12)
```

Figura 21: Tráfego gerado pelo comando *ping* de PC_x para PC_z e capturado em PC_z.

Observa-se que o endereço de origem dos pacotes, *10.0.6.22*, é a interface *eth3* de R_{22} configurada na regra *masquerade*. Confirma-se, portanto, que este roteador altera o endereço de origem dos pacotes, substituindo o endereço privado *192.168.1.190* para um público, conforme esperado. Além disso, observa-se que o valor de TTL dos pacote vale 61, isto é, três roteadores participaram do percurso, sendo eles, R_{21}, R_{22} e R_{25}.

O mesmo comando *ping* produz o seguintes resultados, desta vez capturados em PC_x.

```
192.168.2.190 10.0.0.200    ICMP    98 Echo (ping) request  id=0x41d3, seq=1/256, ttl=64 (reply in 4)
10.0.0.200    192.168.2.190 ICMP    98 Echo (ping) reply    id=0x41d3, seq=1/256, ttl=125 (request in 3)
```

Figura 22: Tráfego gerado pelo comando *ping* de PC_x para PC_z e capturado em PC_x.

Nota-se, neste caso, que os endereço de destino do segundo pacote, enviado por PC_z, é um endereço privado (192.168.2.190) e, portanto, desconhecido por tal *host*. Isto indica que R_{22} traduziu o endereço *10.0.6.22* por *192.1.168.190* quando os pacotes enviados por PC_z passaram por ele. Além disso, constata-se também que o TTL dos pacotes de retorno sofreram um descréscimo de 3, indicando que 3 roteadores participaram do percurso, igualmente ao parágrafo anterior.

Exercício 4

Para permitir o comando *telnet* entre os roteadores R_{21} e R_{23}, é necessário defirnirmos duas regras em cada um dos roteadores R_{22} e R_{24}. A primeira deve ser responsável por traduzir os endereços de origem dos pacotes vindos da rede privada e a segunda deve ser capaz de traduzir os endereços de destino públicos dos pacotes originados na rede pública para um endereço privado. De acordo com os exercícios anteriores, vimos que regras do tipo *masquerade* são capazes de satisfazer a primeira condição e regras do tipo *dstnat*, a segunda. Sendo assim, as regras ativadas em cada um dos roteadores R_{22} e R_{24} estão representadas nas figuras 23 e 24, respectivamente.

```
[admin@Router-22] > ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=srcnat action=masquerade out-interface=ether3 log=no log-prefix=""

 1    chain=dstnat action=dst-nat to-addresses=192.168.1.21
      dst-address=10.0.6.22 log=no log-prefix=""
```

Figura 23: Regras NAT adicionadas ao roteador R_{22}.

```
[admin@Router-24] > ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=srcnat action=masquerade out-interface=ether5 log=no log-prefix=""

 1    chain=dstnat action=dst-nat to-addresses=192.168.1.23
      dst-address=10.0.1.24 log=no log-prefix=""
```

Figura 24: Regras NAT adicionadas ao roteador R_{24}.

As primeiras regras “escondem” as respectivas redes privadas atrás de um endereço público. As segundas regras traduzem endereços públicos em privados. Para o R_{22}, o endereço público será traduzido no endereço da interface *eth1* de R_{21}, isto é, *192.168.1.21* e, para R_{24}, o endereço público será traduzido para a *eth1* de R_{23}, cujo endereço é *192.168.1.23*.

Dessa maneira, foi possível realizar os comandos *telnet* em ambos os roteadores, conforme figuras 25 e 26.

```
[admin@Router-23] > quitConnection closed by foreign host.

Welcome back!
[admin@Router-21] > system telnet 10.0.1.24
```

Figura 25: *Telnet* de R_{21} a R_{23}.

```
[admin@Router-21] > quitConnection closed by foreign host.

Welcome back!
[admin@Router-23] > system telnet 10.0.6.22
```

Figura 26: *Telnet* de R_{23} a R_{21}.

Nota-se que os endereços utilizados para comunicação foram aqueles definidos nas respectivas regras *dstnat*.

Em seguida, definem-se regras para não permitir os protocolos *ssh* e *ftp* nas redes privadas. Adiciona-se a seguinte regra em R_{22} e R_{24}. Para tal, usamos os fatos que ambos protocolos são implementados sobre o protocolo TCP e que utilizam as portas 20 e 21, para o *ftp*, e 22, para o *ssh*.

```
5    ;;; deny FTP and SSH
     chain=forward action=drop protocol=tcp port=20,21,22 log=no
     log-prefix=""
```

Figura 26: Regra para negar protocolos *ssh* e *ftp*.

Exercício 5

Para permitir a comunicação *ftp* entre PC_x e R_{24}, é necessário inicialmente retirar as regras definidas no exercício 4, cujo objetivo era justamente impedir esse tipo de troca de dados. Em seguida, para R_{22}, definimos a regras *masquerade* (tradução de endereços de origem privados) e *dstnat* (tradução do endereço de destino público para o endereço do *host* PC_x, isto é *192.168.1.190*), conforme figura 27.

```
[admin@Router-22] > ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=srcnat action=masquerade out-interface=ether3 log=no log-prefix=""

 1    chain=dstnat action=dst-nat to-addresses=192.168.2.190
      dst-address=10.0.6.22 log=no log-prefix=""
```

Figura 27: Regras NAT adicionadas ao roteador R_{22}.

Para R_{24}, basta mantermos a regra *masquerade*, conforme figura 28.

```
[admin@Router-24] > ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=srcnat action=masquerade out-interface=ether5 log=no log-prefix=""
```

Figura 28: Regras NAT adicionadas ao roteador R_{24}.

Em seguida, executamos o *Wireshark* nos *hosts* PC_x e PC_z e iniciamos uma transferência de dados *ftp.* No host PC_z, presente na rede pública, obtém-se a figura 29 para um dos pacotes capturados.

```
150 388.779575 10.0.6.22          10.0.1.24          FTP      90 Request: PORT 10,0,6,22,178,218
⊞ Internet Protocol Version 4, Src: 10.0.6.22 (10.0.6.22), Dst: 10.0.1.24 (10.0.1.24)
⊞ Transmission Control Protocol, Src Port: 52842 (52842), Dst Port: 21 (21), Seq: 34, Ack: 171, Len:
⊟ File Transfer Protocol (FTP)
  ⊟ PORT 10,0,6,22,178,218\r\n
      Request command: PORT
      Request arg: 10,0,6,22,178,218
      Active IP address: 10.0.6.22 (10.0.6.22)
      Active port: 45786
```

Figura 29: Tráfego *ftp* capturado em PC_z.

Conclui-se, analisando o conteúdo do pacote *ftp* na figura 29, que o protocolo NAT também altera o endereço IP privado presente no *payload* desse pacote, modificando-o para um endereço público. Dessa maneira, a comunicação entre *hosts* de redes públicas e privadas é possível.

No host PC_x, captura-se o seguinte pacote:

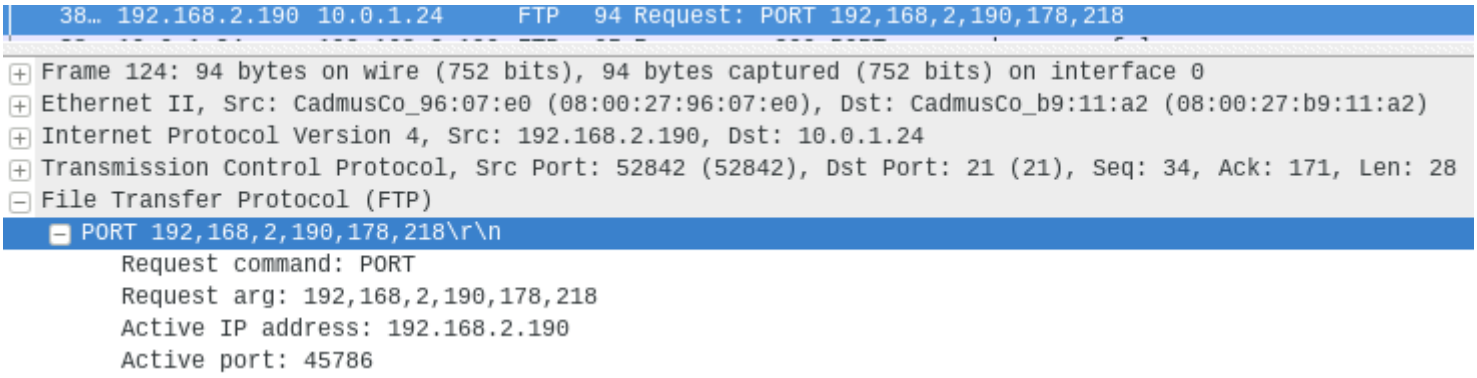

Figura 30: Tráfego *ftp* capturado em PC_x.

Nota-se que o endereço IP nos dados neste caso é privado, isto é, *192.168.1.190*, o que indica que realmente o protocolo NAT altera tal endereço no *payload* dos pacotes. Destaca-se ainda que a *Active Port* é a mesma, isto é, 45786.

Exercício 6

O processo de obtenção de um endereço IP de um servidor DHCP possui essencialmente 4 fases: descoberta do servidor DHCP, oferta de um endereço IP, requisição do endereço IP e reconhecimento do servidor.

I. **Descoberta do servidor DHCP:** um cliente envia mensagens via endereço de *broadcast 255.255.255.255* buscando um servidor DHCP. As mensagens são do tipo *DHCP Discover.*
II. **Oferta de um endereço IP:** quando um servidor DHCP recebe uma mensagem de descoberta de um cliente, o servidor reserva um dos endereços dentro do *pool* e envia uma mensagem do tipo *DHCP Offer* ao cliente. Esta mensagem contém o endereço MAC do cliente, permitindo assim a identificação do *host* correto, o endereço IP oferecido, a máscara de rede, a duração da concessão e o endereço IP do servidor.
III. **Requisição do endereço IP:** o cliente, então, envia novamente uma mensagem via *broadcast* ao servidor, requirindo o endereço oferecido previamente.
IV. **Reconhecimento do servidor:** quando o servidor recebe um *DHCP Request* de um cliente, ele envia uma mensagem de *Acknowlegment,* que conclui a negociação dos parâmetros.

(a) As quatro fases acima podem ser observados para os dois *hosts* nas figuras 31 e 32 a seguir.

1	0.000000	0.0.0.0	255.255.255.…	DHCP	342 DHCP Discover - Transaction ID 0x67d9811d
2	0.003121	10.0.2.21	10.0.2.75	DHCP	342 DHCP Offer - Transaction ID 0x67d9811d
3	0.003436	0.0.0.0	255.255.255.…	DHCP	342 DHCP Request - Transaction ID 0x67d9811d
4	0.003771	10.0.2.21	10.0.2.75	DHCP	342 DHCP ACK - Transaction ID 0x67d9811d
5	0.018297	CadmusCo_96:07:e0	Broadcast	ARP	42 Who has 10.0.2.21? Tell 10.0.2.75
6	0.018528	CadmusCo_b9:11:a2	CadmusCo_96:…	ARP	60 10.0.2.21 is at 08:00:27:b9:11:a2

Figura 31: Tráfego gerado entre servidor DHCP e PC_x. (UBUNTU)

1	0.000000	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0xbdaeee53
2	0.000058	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0xbdaeee53
3	0.001535	10.0.6.25	10.0.0.55	DHCP	342 DHCP Offer - Transaction ID 0xbdaeee53
4	0.001755	0.0.0.0	255.255.255.255	DHCP	347 DHCP Request - Transaction ID 0xbdaeee53
5	0.001798	0.0.0.0	255.255.255.255	DHCP	347 DHCP Request - Transaction ID 0xbdaeee53
6	0.002954	10.0.6.25	10.0.0.55	DHCP	342 DHCP ACK - Transaction ID 0xbdaeee53
7	0.004422	CadmusCo_f7:…	Broadcast	ARP	42 Gratuitous ARP for 10.0.0.55 (Request)
8	0.004447	CadmusCo_f7:…	Broadcast	ARP	42 Gratuitous ARP for 10.0.0.55 (Request)
9	0.090051	CadmusCo_f7:…	Broadcast	ARP	42 Gratuitous ARP for 10.0.0.55 (Request)
10	0.090107	CadmusCo_f7:…	Broadcast	ARP	42 Gratuitous ARP for 10.0.0.55 (Request)

Figura 32: Tráfego gerado entre servidor DHCP e PC_y. (WINDOWS)

Foram atribuídos, portanto, os endereços *10.0.2.75/23* para o PC_x e *10.0.0.55/23* para o PC_y. O atributo *subnet mask* (*255.255.254.0* para ambos) também pode ser consultado no conteúdo dessas mensagens, assim como a identificação do servidor DHCP, como endereço IP, domínio, tipo de mensagem etc.

(b) Os clientes, que não possuem endereços IP, comunicam-se com o servidor via endereço *broadcast,* isto é, *255.255.255.255*. O servidor, cujo endereço é *10.0.2.21*, comunica-se diretamente aos *hosts* ou via um roteador que atua como *relay.* Neste caso, tal *relay* é o rotedor R_{25} e a sua interface ligada ao *host* apresenta endereço *10.0.6.25*.

(c) O *host* é capaz de se comunicar com o servidor via *broadcast.* O servidor é capaz de se comunicar com o cliente graças ao endereço MAC, que também é transmitido nas mensagens DHCP.

(d) Sim, muitos pacotes ARP são enviados, conforme figuras anteriores. A finalidade desses pacotes é verificar o endereço IP recentemente adquirido na rede a fim de evitar conflitos de endereços causados por possíveis sobreposições dos *pools* nos diferentes servidores DHCP que podem compor a rede.

(e) As diferentes mensagens DHCP estão explicadas previamente.

(f) Para as mensagens do tipo DHCP possuem, dentre outros campos:

- *Client IP address, Your (client) IP address, Client MAC Address, Relay agent IP Address:* informações sobre o cliente e o *relay* (caso exista);
- *DHCP Message Type*: o tipo de mensagem (*discover, offer, request* ou *acknowledgment*);
- *Client identifier*: possui o endereço MAC da máquina cliente;
- *Requested IP Address:* o endereço IP requirido pelo cliente;
- *DHCP Server Identifier:* endereço IP do servidor/relay DHCP.
- *Parameter Request List:* lista com diversos parâmetros de configuração de rede. Tais parâmetros incluem submáscara de rede, rotas estáticas, nome do domínio etc.