Entrega: 16/11/2023

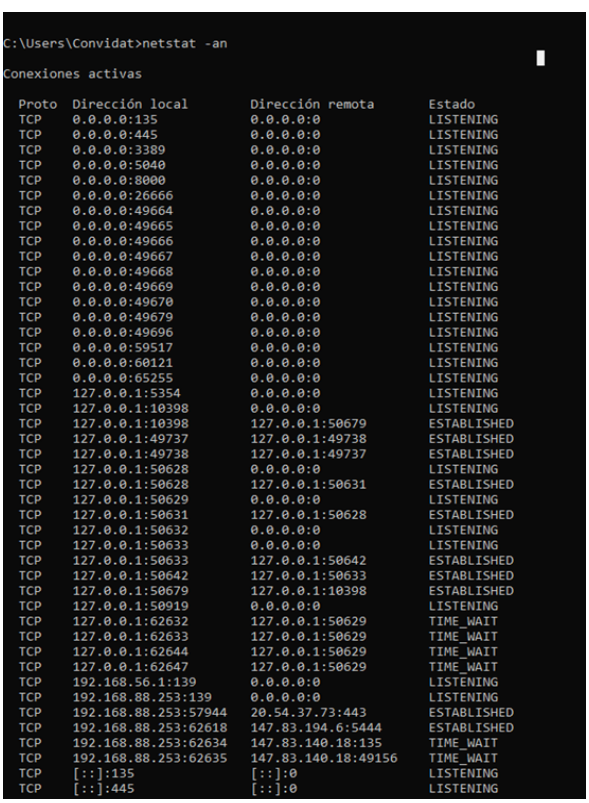
Alumnes: Francesco Oncins Spedo
Mariona Farré Tapias
Pau Alcázar Perdomo

INTERNET:

P2-Qüestionari sessió 1 -Serveis, ports i connexions: anàlisi en local i escaneig de ports

1.

Segons el tipus de sistema operatiu que es faci servir utilitzarem una comanda o una altra:
Per a **windows** podem utilitzar *netstat -an*: que obté un llistat de totes les connexions actives del dispositiu, per cada connexió veurem: el tipus de protocol (TCP o UDP) , la direcció inicial (local), la direcció final (remota) i l'estat d'aquesta:



```
C:\Users\Convidat>netstat -an

Conexiones activas

  Proto  Dirección local        Dirección remota       Estado
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:8000           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:26666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49669          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49670          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49679          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49696          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:59517          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:60121          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:65255          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:5354         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:10398        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:10398        127.0.0.1:50679        ESTABLISHED
  TCP    127.0.0.1:49737        127.0.0.1:49738        ESTABLISHED
  TCP    127.0.0.1:49738        127.0.0.1:49737        ESTABLISHED
  TCP    127.0.0.1:50628        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:50628        127.0.0.1:50631        ESTABLISHED
  TCP    127.0.0.1:50629        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:50631        127.0.0.1:50628        ESTABLISHED
  TCP    127.0.0.1:50632        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:50633        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:50633        127.0.0.1:50642        ESTABLISHED
  TCP    127.0.0.1:50642        127.0.0.1:50633        ESTABLISHED
  TCP    127.0.0.1:50679        127.0.0.1:10398        ESTABLISHED
  TCP    127.0.0.1:50919        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:62632        127.0.0.1:50629        TIME_WAIT
  TCP    127.0.0.1:62633        127.0.0.1:50629        TIME_WAIT
  TCP    127.0.0.1:62644        127.0.0.1:50629        TIME_WAIT
  TCP    127.0.0.1:62647        127.0.0.1:50629        TIME_WAIT
  TCP    192.168.56.1:139       0.0.0.0:0              LISTENING
  TCP    192.168.88.253:139     0.0.0.0:0              LISTENING
  TCP    192.168.88.253:57944   20.54.37.73:443        ESTABLISHED
  TCP    192.168.88.253:62618   147.83.194.6:5444      ESTABLISHED
  TCP    192.168.88.253:62634   147.83.140.18:135      TIME_WAIT
  TCP    192.168.88.253:62635   147.83.140.18:49156    TIME_WAIT
  TCP    [::]:135               [::]:0                 LISTENING
  TCP    [::]:445               [::]:0                 LISTENING
  TCP    [::]:3389              [::]:0                 LISTENING
  TCP    [::]:49664             [::]:0                 LISTENING
  TCP    [::]:49665             [::]:0                 LISTENING
  TCP    [::]:49666             [::]:0                 LISTENING
  TCP    [::]:49667             [::]:0                 LISTENING
  TCP    [::]:49668             [::]:0                 LISTENING
  TCP    [::]:49669             [::]:0                 LISTENING
  TCP    [::]:49670             [::]:0                 LISTENING
  TCP    [::]:49679             [::]:0                 LISTENING
  TCP    [::]:49696             [::]:0                 LISTENING
  TCP    [::]:59517             [::]:0                 LISTENING
  TCP    [::]:60121             [::]:0                 LISTENING
  TCP    [::]:65255             [::]:0                 LISTENING
  TCP    [::1]:49671            [::]:0                 LISTENING
  UDP    0.0.0.0:123            *:*
  UDP    0.0.0.0:500            *:*
  UDP    0.0.0.0:1434           *:*
  UDP    0.0.0.0:3389           *:*
  UDP    0.0.0.0:4500           *:*
  UDP    0.0.0.0:5050           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5355           *:*
  UDP    0.0.0.0:57951          *:*
  UDP    127.0.0.1:1900         *:*
  UDP    127.0.0.1:57492        *:*
  UDP    127.0.0.1:58714        *:*
  UDP    127.0.0.1:61516        *:*
  UDP    192.168.56.1:137       *:*
  UDP    192.168.56.1:138       *:*
  UDP    192.168.56.1:1900      *:*
  UDP    192.168.56.1:5353      *:*
  UDP    192.168.56.1:61514     *:*
  UDP    192.168.88.253:137     *:*
  UDP    192.168.88.253:138     *:*
  UDP    192.168.88.253:1900    *:*
  UDP    192.168.88.253:5353    *:*
  UDP    192.168.88.253:61515   *:*
  UDP    [::]:123               *:*
  UDP    [::]:500               *:*
  UDP    [::]:1434              *:*
  UDP    [::]:3389              *:*
  UDP    [::]:4500              *:*
  UDP    [::]:5353              *:*
  UDP    [::]:5355              *:*
  UDP    [::]:57952             *:*
  UDP    [::1]:1900             *:*
  UDP    [::1]:5353             *:*
  UDP    [::1]:61513            *:*
  UDP    [fe80::46c8:4f72:25c3:e672%14]:1900  *:*
  UDP    [fe80::46c8:4f72:25c3:e672%14]:61511  *:*
  UDP    [fe80::aee2:245c:bd4a:7c1d%4]:1900  *:*
  UDP    [fe80::aee2:245c:bd4a:7c1d%4]:61512  *:*
```

Els ports TCP que estiguin escoltant tindran l'estat de LISTENING, els sockets que tinguin una connexió establerta tindran com a estat ESTABLISHED i els sockets que estiguin esperant més paquets i que segueixin a la xarxa després de tancar-se, apareixeran amb l'estat de TIME_WAIT.

Si utilitzem la comanda *netstat -s* podem veure les estadístiques de cada protocol, versions IP, etc:

```
Estadísticas de TCP para IPv4

  Activos abiertos                    = 613
  Pasivos abiertos                    = 141
  Intentos de conexión erróneos       = 386
  Conexiones restablecidas            = 80
  Conexiones actuales                 = 10
  Segmentos recibidos                 = 27513
  Segmentos enviados                  = 24007
  Segmentos retransmitidos            = 97

Estadísticas de TCP para IPv6

  Activos abiertos                    = 6
  Pasivos abiertos                    = 0
  Intentos de conexión erróneos       = 102
  Conexiones restablecidas            = 0
  Conexiones actuales                 = 0
  Segmentos recibidos                 = 20
  Segmentos enviados                  = 16
  Segmentos retransmitidos            = 4

Estadísticas UDP para IPv4

  Datagramas recibidos    = 105769
  Sin puerto              = 498
  Errores de recepción    = 24
  Datagramas enviados     = 41795

Estadísticas UDP para IPv6

  Datagramas recibidos    = 7058
  Sin puerto              = 11
  Errores de recepción    = 14
  Datagramas enviados     = 418

C:\Users\Convidat>
```

Si encanvi utilitzem **linux**, amb l'execució de *ss -a:* obtindrem una llistas de totes les connexions, nosaltres ens hem fixat només amb els tipus UDP i TCP. Per a cada connexió veurem el tipus de protocol (Netid), l'estat d'aquesta connexió, la quantitat de dades que s'esperen rebre de la cua del socket (recv-Q), la quantitat de dades que estan esperant a ser enviades pel socket (Send-Q), l'adreça local IP de la màquina i el port del socket on es vol establir la connexió (Local Address: Port) i l'adreça local IP amb el número del port on es vol que arribi la connexió des del socket local (Peer Address:Port)

```
Netid      State       Recv-Q      Send-Q                Local Address:Port                Peer Address:Port      Process

udp        UNCONN      0           0                           0.0.0.0:mdns                      0.0.0.0:*
udp        UNCONN      0           0                     127.0.0.53%lo:domain                    0.0.0.0:*
udp        ESTAB       0           0          192.168.88.254%enp3s0:bootpc               192.168.88.1:bootps
udp        UNCONN      0           0                           0.0.0.0:631                       0.0.0.0:*
udp        UNCONN      0           0                           0.0.0.0:57995                     0.0.0.0:*
udp        UNCONN      0           0                              [::]:mdns                         [::]:*
udp        UNCONN      0           0                              [::]:50760                        [::]:*
tcp        LISTEN      0           4096                  127.0.0.53%lo:domain                    0.0.0.0:*
tcp        LISTEN      0           128                      127.0.0.1:ipp                        0.0.0.0:*
tcp        LISTEN      0           128                         [::1]:ipp                            [::]:*
francesco@francesco-GL75-Leopard-10SEK: $
```

Les connexions UDP i TCP tenen 3 possibles estats, l'estat UNCONN(Unconnected), l'estat ESTAB(established) i el de LISTEN.
Si la connexió té com a estat UNCONN, voldrà dir que el socket UDP no està activament connectat a cap socket, si l'estat és ESTAB, significarà que els dos hosts ja s'han enviat els paquets SYN i SYN-ACK i es tenen una connexió establerta entre ells, si en canvi l'estat és de LISTEN ens indicarà que el port TCP ja està escoltant un socket.

A la **màquina Virtual**, es pot fer servir la mateixa instrucció de Linux (*ss -an*): retorna el mateix tipus de llistat de connexions que a Linux, però ara amb el paràmetre *-an, on* la diferència és que tenim l'adreça completa sense tenir el *hostname* en l'adreça (en comptes de posar *domain* posarà el número concret de port, per exemple el 53)

```
udp    UNCONN     0      0                *:631                      *:*
udp    UNCONN     0      0                *:47063                    *:*
udp    UNCONN     0      0                *:5353                     *:*
udp    UNCONN     0      0     192.168.88.253:53                     *:*
udp    UNCONN     0      0          127.0.0.1:53                     *:*
udp    UNCONN     0      0                *:68                       *:*
udp    UNCONN     0      0               :::5353                    :::*
udp    UNCONN     0      0               :::53                      :::*
udp    UNCONN     0      0               :::40792                   :::*
tcp    LISTEN     0      10    192.168.88.253:53                     *:*
tcp    LISTEN     0      10         127.0.0.1:53                     *:*
tcp    LISTEN     0      128              *:22                       *:*
tcp    LISTEN     0      5          127.0.0.1:631                    *:*
tcp    LISTEN     0      128              *:23                       *:*
tcp    LISTEN     0      100              *:25                       *:*
tcp    LISTEN     0      128        127.0.0.1:953                    *:*
tcp    LISTEN     0      128             :::80                      :::*
tcp    LISTEN     0      10              :::53                      :::*
tcp    LISTEN     0      32              :::21                      :::*
tcp    LISTEN     0      128             :::22                      :::*
tcp    LISTEN     0      5              ::1:631                     :::*
tcp    LISTEN     0      100             :::25                      :::*
tcp    LISTEN     0      128            ::1:953                     :::*
root@debian95-INTE-server:~# _
```

Alguns exemples amb l'execució de ss -a:

```
udp    UNCONN     0      0        127.0.0.1:domain                   *:*
udp    UNCONN     0      0                *:bootpc                   *:*
udp    UNCONN     0      0               :::mdns                    :::*
udp    UNCONN     0      0               :::domain                  :::*
udp    UNCONN     0      0               :::40792                   :::*
```

Per poder veure els ports que està escoltant amb el **windows** s'executaria la instrucció: netstat -an | find "LISTENING"

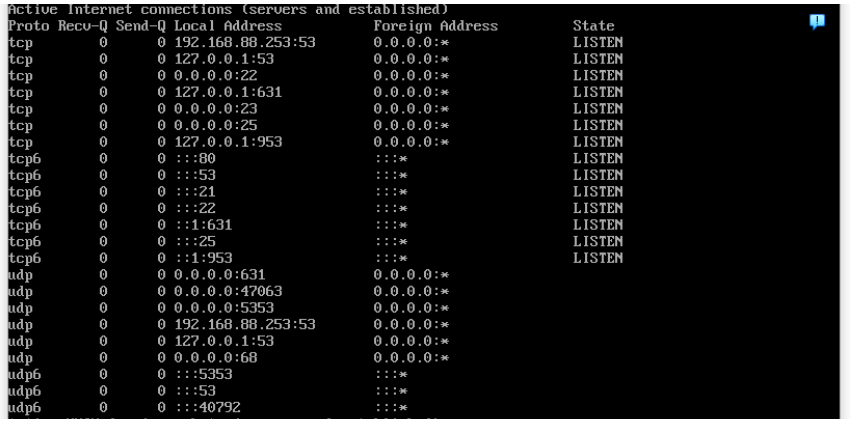
```
C:\Users\mft19>netstat -an | find "LISTENING"
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:7680           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49669          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49670          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49674          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:50572          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:57621          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:2015         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:57725        0.0.0.0:0              LISTENING
  TCP    192.168.1.139:139      0.0.0.0:0              LISTENING
  TCP    192.168.56.1:139       0.0.0.0:0              LISTENING
  TCP    [::]:135               [::]:0                 LISTENING
  TCP    [::]:445               [::]:0                 LISTENING
  TCP    [::]:7680              [::]:0                 LISTENING
  TCP    [::]:49664             [::]:0                 LISTENING
  TCP    [::]:49665             [::]:0                 LISTENING
  TCP    [::]:49666             [::]:0                 LISTENING
  TCP    [::]:49669             [::]:0                 LISTENING
  TCP    [::]:49670             [::]:0                 LISTENING
  TCP    [::]:49674             [::]:0                 LISTENING
  TCP    [::1]:49673            [::]:0                 LISTENING
```

Perquè només retornes el llistat de ports que realment està escoltant (només estat LISTENING)

Per **linux i màquina virtual** es pot executar la instrucció, per només veure els ports que s'escolten:

netstat -an | *more* o

netstat -tu/n (-t protocol TCP, -u UDP -n en format numeric)



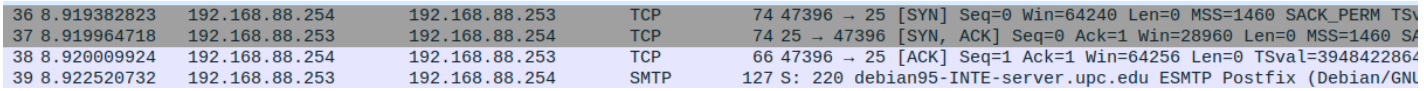
```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.88.253:53       0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
tcp6       0      0 :::25                   :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:47063           0.0.0.0:*
udp        0      0 0.0.0.0:5353            0.0.0.0:*
udp        0      0 192.168.88.253:53       0.0.0.0:*
udp        0      0 127.0.0.1:53            0.0.0.0:*
udp        0      0 0.0.0.0:68              0.0.0.0:*
udp6       0      0 :::5353                 :::*
udp6       0      0 ::1:53                  :::*
udp6       0      0 :::40792                :::*
```

2.

El protocol de telnet és un network protocol per accedir virtualment a un ordinador i establir una connexió bilateral entre les dos màquines, també es pot fer servir per revisar manualment si un port esta obert i connectat a un servei o aplicació.

El protocol de transport que es fa servir en un escaneig de ports amb telnet és TCP(Transmission Control Protocol), perquè opera dins de la capa de transport del model OSI i pel que ofereix TCP: és orientat a connexió, fiable, no permet duplicació, permet ordenació i missatges urgents.

Utilitza TCP per establir connexions a ports específics d'un dispositiu, permetent així determinar si aquests ports estan oberts o tancats.



```
36 8.919382823   192.168.88.254   192.168.88.253   TCP      74 47396 → 25 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSv
37 8.919964718   192.168.88.253   192.168.88.254   TCP      74 25 → 47396 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SA
38 8.920009924   192.168.88.254   192.168.88.253   TCP      66 47396 → 25 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3948422864
39 8.922520732   192.168.88.253   192.168.88.254   SMTP    127 S: 220 debian95-INTE-server.upc.edu ESMTP Postfix (Debian/GNU
```

En aquesta foto podem veure com telnet utilitza el protocol TCP.

3.

Telnet és un protocol que només pot escoltar connexions TCP, de manera que altres ports o serveis no poden ser escanejats eficaçment. Aquest ports/serveis que no poden ser escanejats amb telnet són:

- Ports UDP: utitlizen un protocol diferent de TCP, no estan orientats a connexió.
- Ports TCP Filtrats: els ports poden estar protegits per un firewall fent que no puguin ser escanejats per telnet.
- Serveis SSL/TLS: serveis que necessiten una connexió segura amb SSL o ports amb HTTPS, no poden ser escanejats perquè no deixarà establir connexió.
- Ports dins de la capa d'aplicació: si un servei necessita una autenticació per qualsevol interacció, per exemple amb servei de base de dades, telnet tampoc podrà connectar-se.

4.

Per fer un escaneig de ports (un telnet) des de Linux a la màquina virtual VM, i intentar tenir una connexió correcta, primer hem de comprovar els ports oberts de la llista de ports de VM que des de Linux es podrà connectar:

```
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
```

Una vegada triat un port obert, podem fer un telnet en aquest:

```
francesco@francesco-GL75-Leopard-10SEK:~$ telnet 192.168.88.253 80
Trying 192.168.88.253...
Connected to 192.168.88.253.
Escape character is '^]'.
```

Si entrem al wireshark podem fer el seguiment dels paquets intercanviats al fer el handshake de 3 passos entre els hosts i poder veure que la connexió s'ha fet correctament:

```
2 1.784736039   192.168.88.254   192.168.88.253   TCP   74 60458 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=146
3 1.785361343   192.168.88.253   192.168.88.254   TCP   74 80 → 60458 [SYN, ACK] Seq=0 Ack=1 Win=28960 Le
4 1.785408486   192.168.88.254   192.168.88.253   TCP   66 60458 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 T
```

Host Linux→ SYN → Host VM
Host Linux ← SYN-ACK ← Host VM
Host Linux → ACK → Host VM

SYN: El host Linux comença la connexió envia un paquet TCP amb la flag de SYN marcada al host VM.
SYN-ACK: La resposta del paquet SYN, VM reconeix el la petició de Linux, vol establir una connexió i respon amb el SYN i el ACK.
ACK: l'últim pas en l'execució, al rebre el paquet SYN-ACK, VM reconeix l'establiment de la connexió amb Linux.
A partir d'aquí la connexió està establerta des del port triat.

5.
Amb la comanda *netstat -an* | *more* podem veure els ports de la màquina virtual que estan escoltant (en estat de LISTEN) i també podem veure els ports que estan oberts

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.88.253:53       0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
tcp6       0      0 ::1:25                  :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:47063           0.0.0.0:*
udp        0      0 0.0.0.0:5353            0.0.0.0:*
udp        0      0 192.168.88.253:53       0.0.0.0:*
udp        0      0 127.0.0.1:53            0.0.0.0:*
udp        0      0 0.0.0.0:68              0.0.0.0:*
udp6       0      0 :::5353                 :::*
udp6       0      0 :::53                   :::*
udp6       0      0 :::40792                :::*
```

Ports TCP en estat de LISTEN:

- Port 21: FTP (File Transfer Protocol) Control (per a la transferència d'arxius).
- Port 22: És comú per SSH (Secure Shell) per a connexions segures.
- Port 23: Telnet per a la comunicació de text sense xifrar.
- Port 25: SMTP (Simple Mail Transfer Protocol) per a l'enviament de correus electrònics.
- Port 53: Utilitzat per DNS (Domain Name System) i pot escoltar tant en TCP com en UDP.
- Port 80: Servei web HTTP (Hypertext Transfer Protocol).
- Port 631: IPP (Internet Printing Protocol) per a impressió en xarxa.
- Port 953: Associat amb el servei rndc per a la gestió remota de servidors DNS BIND.

Ports UDP (aquests no estan en estat de LISTEN ja que UDP és un protocol no orientat a la connexió):

- Port 53: També per DNS, utilitzant UDP.
- Port 5353: mDNS (Multicast DNS).
- Port 40792: No és un port estàndard conegut i podria ser utilitzat per un servei específic.

6.

Si pel contrari, intentem fer un escaneig de ports a un port no obert, des de Linux ens retornarà un error de connexió, ja que li serà impossible connectar-se a un port no obert de la màquina virtual VM:

```
francesco@francesco-GL75-Leopard-10SEK:~$ telnet 192.168.88.253 20
Trying 192.168.88.253...
telnet: Unable to connect to remote host: Connection refused
```

Des de wireshark es pot veure l'intercanvi de paquets i com es rebutja la connexió:

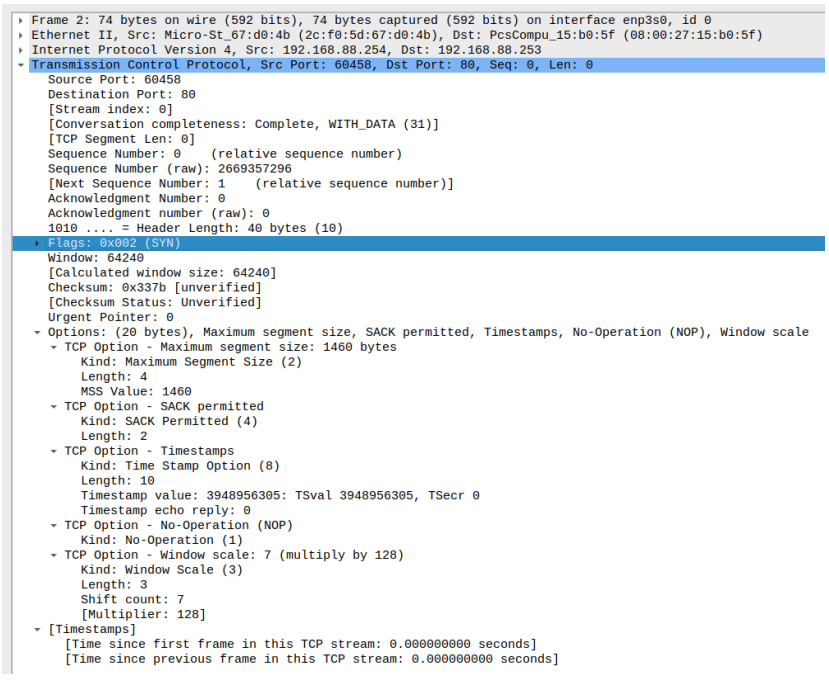
```
     5 2.425533177   192.168.88.254   192.168.88.253   TCP   74 42188 → 20 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PER
     6 2.426115087   192.168.88.253   192.168.88.254   TCP   60 20 → 42188 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

Host Linux→ SYN → Host VM
Host Linux ← RST-ACK ← Host VM

SYN: El host Linux comença la connexió enviant un paquet TCP amb la flag de SYN marcada al host VM.
RST-ACK: La resposta del paquet SYN, la VM no reconeix el la petició de Linux, i rebutja l'establiment de connexió responent amb el RST i el ACK.
El camp RST vol dir "reset", reestabliment de la connexió.

7.

Dins del paquet SYN que envia el host A al host B, ja que és el paquet que envia les característiques de la connexió que el host B pot acceptar o rebutjar:



Es pot veure que el paquet és de sincronisme SYN perquè té la Flag de SYN
Al entrar a les dades del paquet, podem veure les opcions TCP que ofereix el host A, com podria ser:

- la màxima llargada de segment: 1460 bytes
- Si el SACK està permès
- Els timestamps pel paquet
- No-operation
- El número màxim de la finestra (7 x 128)

8.

El Retransmission TimeOut (RTO) inicial en TCP es determina durant la fase de three-way handshake inicial de TCP basant-se en les mesures del Round-Trip Time (RTT) si estan disponibles. Si no hi ha prou dades per mesurar el RTT, TCP utilitzarà un valor predeterminat, que pot variar entre els diferents sistemes operatius. Sovint, aquest valor està al voltant de casi 0 a 3 segons.

```
▸ Flags: 0x002 (SYN)
  Window: 64240
  [Calculated window size: 64240]
  Checksum: 0x337b [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
▾ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps,
  ▾ TCP Option - Maximum segment size: 1460 bytes
      Kind: Maximum Segment Size (2)
      Length: 4
      MSS Value: 1460
  ▾ TCP Option - SACK permitted
      Kind: SACK Permitted (4)
      Length: 2
  ▾ TCP Option - Timestamps
      Kind: Time Stamp Option (8)
      Length: 10
      Timestamp value: 3948956305: TSval 3948956305, TSecr 0
      Timestamp echo reply: 0
  ▾ TCP Option - No-Operation (NOP)
      Kind: No-Operation (1)
  ▾ TCP Option - Window scale: 7 (multiply by 128)
      Kind: Window Scale (3)
      Length: 3
      Shift count: 7
      [Multiplier: 128]
▾ [Timestamps]
    [Time since first frame in this TCP stream: 0.000000000 seconds]
    [Time since previous frame in this TCP stream: 0.000000000 seconds]
```

```
▸ Flags: 0x012 (SYN, ACK)
  Window: 28960
  [Calculated window size: 28960]
  Checksum: 0x0f97 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
▾ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps,
  ▾ TCP Option - Maximum segment size: 1460 bytes
      Kind: Maximum Segment Size (2)
      Length: 4
      MSS Value: 1460
  ▾ TCP Option - SACK permitted
      Kind: SACK Permitted (4)
      Length: 2
  ▾ TCP Option - Timestamps
      Kind: Time Stamp Option (8)
      Length: 10
      Timestamp value: 1056069: TSval 1056069, TSecr 3948956305
      Timestamp echo reply: 3948956305
  ▾ TCP Option - No-Operation (NOP)
      Kind: No-Operation (1)
  ▾ TCP Option - Window scale: 5 (multiply by 32)
      Kind: Window Scale (3)
      Length: 3
      Shift count: 5
      [Multiplier: 32]
▾ [Timestamps]
    [Time since first frame in this TCP stream: 0.000625304 seconds]
    [Time since previous frame in this TCP stream: 0.000625304 seconds]
▾ [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 2]
    [The RTT to ACK the segment was: 0.000625304 seconds]
    [iRTT: 0.000672447 seconds]
```

Podem veure com en el primer paquet (SYN) que és el que envia el host de linux per establir la connexió no hi ha cap RTT i diu que el timestamp és de 0 segons, mentre que en el segon paquet que s'envia (SYN, ACK) i que és la resposta de la VM ja apareix un RTT de 0.000625304 segons del ACK i RTT en total de 0.000672447 segons.

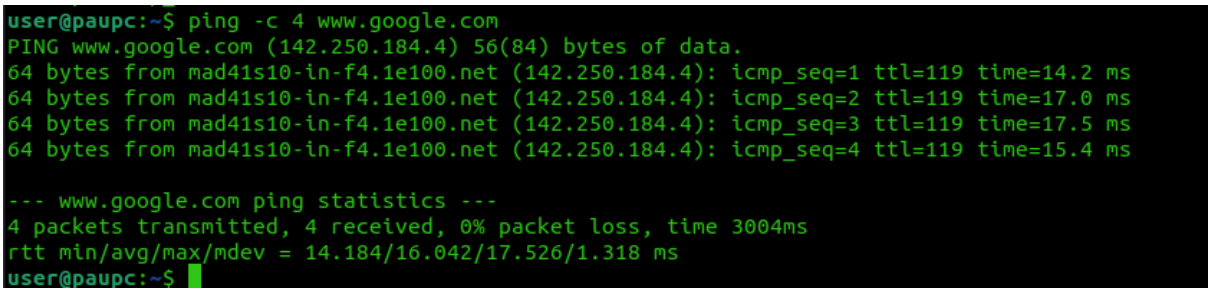
9.

La Per trobar el màxim de retransmissions d'un sistema operatiu, podem executar les següents instruccions:

```
user@paupc:~$ sysctl net.ipv4.tcp_retries2
net.ipv4.tcp_retries2 = 15
```

La imatge mostra una terminal d'un sistema operatiu de Unix/Linux on s'ha executat la comanda sysctl amb el paràmetre net.ipv4.tcp_retries2. Aquest paràmetre controla el nombre màxim de retransmissions TCP que es realitzaran en un intent d'establir una connexió abans de decidir que la contrapart no està disponible.

El valor 15 indica que el kernel està configurat per intentar retransmetre paquets TCP fins a 15 vegades abans de considerar fallida la connexió. Aquesta configuració és un dels mecanismes que utilitza TCP per gestionar la fiabilitat i el control de congestió en la xarxa. Un valor alt per a tcp_retries2 podria permetre més temps per a recuperar-se en condicions de xarxa inestables, però també podria significar un temps d'espera més llarg abans que una aplicació sigui notificada de la fallada de la connexió.



Aquesta captura mostra la sortida d'una comanda ping que ha estat executada en una terminal. La comanda s'ha utilitzat per enviar paquets de xarxa a www.google.com, i el resultat mostra la informació del temps que cada paquet ha trigat a anar i tornar des de l'host destinatari, conegut com a Round-Trip Time (RTT) es de 14.184, 16.02, 17.526 9 1.318 msegons per cada paquet enviat.

10.

En el sistema operatiu serà el que tindrà els valors per defecte pels paràmetres de TCP, com la mida de la finestra, els temporitzadors, el màxim segment size (MSS), i altres configuracions. Aquests valors són optimitzats generalment per oferir un rendiment adequat en la majoria de situacions d'ús comunes per aquest sistema operatiu.

A part d'això però les aplicacions poden sol·licitar o suggerir canvis en certs paràmetres de TCP per optimitzar la comunicació segons les seves necessitats específiques. Per exemple, una aplicació de streaming de vídeo podria requerir una mida de finestra major per permetre una transferència de dades més ràpida i eficient.