


|   |   |                           |   |
|---|---|---------------------------|---|
| Generalitat de Catalunya<br>Departament d'Educació<br><b>INS Provençana</b> | Mòdul 6:<br>Seguretat informàtica       | Curs<br>2023-2024         |  |
| Departament d'informàtica<br>Grup SMX2C                                     | UF5 - Tallafocs i monitoratge de xarxes | Nota                      |   |
|   | <b>NF1 - Monitoratge de xarxes</b>      | Data: 30 de Gener de 2024 |   |

**Professors:** Laura Montesinos

**Alumne:** Cristian Martínez

## Instruccions

### Lliurament:

1. Poseu el **nom i cognoms** dins del document que lliureu.
2. Heu de lliurar un **únic document** en format **pdf**.
3. Heu de fer captures de pantalla per demostrar que heu realitzar l'exercici. Les captures de pantalla han d'anar annexades en aquest document.

**Consell:** Cada apartat té exercicis de diferent dificultat (feu primer els que sapigueu).

4. Durada: **2** hores.

- Pots consultar tot el material que consideris adient (també Internet)
- Tots els exercicis valen el mateix.
- Cada apartat té exercicis de diferent dificultat (fes primer els fàcils)

Crea una màquina virtual amb el teu nom (enlloc de Alumne) i el codi **alumne-245**

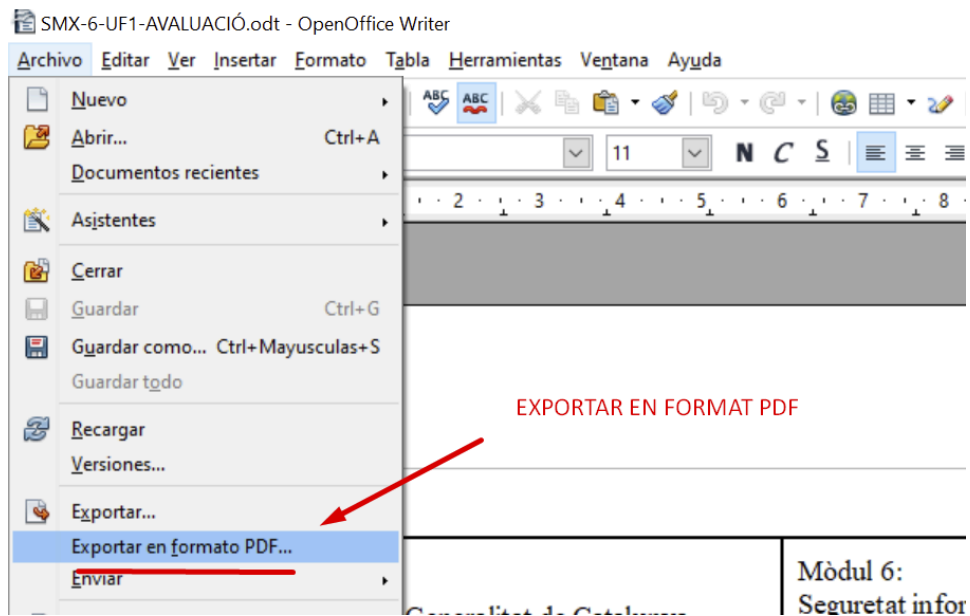
Arrenca la màquina configurant --cpu i --memory :

Inicieu la màquina.

Important. "Aneu al VirtualBox i poseu-hi adaptador pont".  
box.exe code laura-245 -c 4 -m 8000

|         |            |               |            |
|---------|------------|---------------|------------|
| C01-F25 | Versió 1.0 | Pàgina 1 de 2 | 10-09-2020 |
|---------|------------|---------------|------------|

El document l'has d'exportar a PDF:



## Exercicis

### 1. Docker-emmagatzematge- 25%

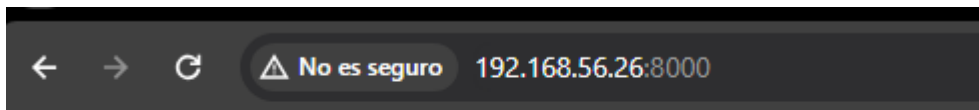
1.1. Borra tots els **contenidors** i verifica amb un docker ps.

```
box@cristian-245:~$ docker rm -f $(docker ps -a -q)
163481f21555
a1b53a90250f
6dd83b230188
box@cristian-245:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
```

Borramos con el comando docker rm -f (forzado) \$(docker ps -a -q)

1.2. Arrenca un contenidor apache amb l'opció **-p 8000:80** perquè poguem connectar-nos des de fora del host. Verifica la pàgina d'inici del teu contenidor apache

```
box@cristian-245:~$ docker run -d -p 8000:80 --name apache_contenedor httpd
1606e2d76739a0bbb44eafa1342e799b277fa43dd3709758181cfe3d19104af6
box@cristian-245:~$
```



## It works!

Funciona!

1.3. Crea una carpeta amb el nom web i crea la pàgina d'inici amb el missatge "**La meva WEB: Nom\_alumne**"

```
box@cristian-245:~$ mkdir web
box@cristian-245:~$ ls
web
```

```
box@cristian-245:~$ cd web
box@cristian-245:~/web$ sudo nano index.html
```

```
box@cristian-245: ~/web
GNU nano 6.2 index.html *
<h1>La meva WEB: Cristian</h1>_
```

- 1.4. Torna a crear un nou contenidor **apache\_web** montant la carpeta **web**

```
Create and Run a new container from an image
box@cristian-245:~$ docker run -d -p 8001:80 --name apache_miweb -v $(pwd)/web:/usr/local/apache2/htdocs httpd
3c7c92fbbf0fa4746ea0859f0a51efe084350815207b01432a4c1b05f7e25bae
box@cristian-245:~$
```

## Montamos la carpeta en el nuevo docker

- 1.5. Fes un **curl** a localhost pots veure que ara respon amb el teu fitxer **index.html** no amb l'index d'inici, que has fet perquè això passi?

```
box@cristian-245:~$ curl http://localhost:8001
<h1>La meva WEB: Cristian</h1>
box@cristian-245:~$ curl http://localhost:8000
<html><body><h1>It works!</h1></body></html>
box@cristian-245:~$
```

Ahora en el puerto 8001 está nuestro texto!

## 2. Docker-xarxes 25%

Per evitar problemes, primer elimina totes les xarxes que no s'estan fent servir:

```
docker network prune -f
```

- 2.1. Crea dues xarxes virtuals amb les següents ips:

net-1→10.0.10.0/24

net-2→ 10.0.20.0/24

```
See docker network create --help .
box@cristian-245:~$ docker network create --driver bridge --attachable \
> --subnet 10.0.10.0/24 net-1
1491e557ae18c881006fbd4bd51f759ac95d671d80243f5839577791e6b944f3
box@cristian-245:~$ docker network create --driver bridge --attachable \
> --subnet 10.0.20.0/24 net-2
6940d55dd350d1e0e1627140865843273480cfe637dde99c15948edb90ca9b42
```

|         |            |               |            |
|---------|------------|---------------|------------|
| C01-F25 | Versió 1.0 | Pàgina 1 de 2 | 10-09-2020 |
|---------|------------|---------------|------------|

Creamos las redes virtuales con los siguientes comandos:

Net-1 : `docker network create --driver bridge --attachable --subnet 10.0.10.0/24 net-1`

Net-2: `docker network create --driver bridge --attachable --subnet 10.0.20.0/24 net-2`

## 2.2. Mostreu el resultat amb **docker network ls**.

```
box@cris-tian-245:~$ docker network ls
NETWORK ID          NAME       DRIVER  SCOPE
453e08421e94        bridge    bridge  local
f399a4bd34c0        host      host    local
1491e557ae18        net-1     bridge  local
5948d55dd350        net-2     bridge  local
237a6718c67c        none      null    local
```

## 2.3. Arrenca un contenidor amb nginx connectat a la primera xarxa-net-1

```
box@cris-tian-245:~$ sudo docker run -d --name nginx-contenedor --network=net-1 nginx
a1b53a90250fc2aec2439fa3027f80c58fec72a9d56ba93557bf24a0ff3a9c0a
box@cris-tian-245:~$
```

Arrancamos el docker con el comando:

`docker run -d --name nginx-contenedor --network=net-1 nginx`

## 2.4. Arrenca un contenidor amb apache connectat a la segona xarxa-net-2.

```
box@cris-tian-245:~$ sudo docker run -d --name apache-contenedor --network=net-2 httpd
6dd83b230188baaa25bef3192d987e4a6ad4fd42c7b7f5d3d4c98c8f33270180
box@cris-tian-245:~$
```

Arrancamos el docker de apache con el siguiente comando:

`docker run -d --name apache-contenedor --network=net-2 httpd`

### 2.4.1. Verifica els contenidors

```
box@cris-tian-245:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
a1b53a90250f   nginx    "/docker-entrypoint..." About a minute ago Up About a minute 80/tcp       nginx-contenedor
6dd83b230188   httpd    "httpd-foreground"      About a minute ago Up About a minute 80/tcp       apache-contenedor
```

Verificamos los contenedores con el siguiente comando

## 2.5. Crear un contenidor amb nom **explorer** amb **alpine**

|         |            |               |            |
|---------|------------|---------------|------------|
| C01-F25 | Versió 1.0 | Pàgina 1 de 2 | 10-09-2020 |
|---------|------------|---------------|------------|

```
explorer
box@cristian-245:~$ sudo docker run -it -d --name explorer alpine
163481f21555c54a2f3fb8b5b6861230d99d33b79809304ac3797885aa5e1dfd
box@cristian-245:~$ sudo docker ps
```

- 2.5.1. Connecta el contenidor explorer a les dues xarxes amb **docker network connect** i arrenca els contenidors:

Para conectarlo a las subnets utilizamos el comando:

```
sudo docker network connect net-1 explorer
```

```
y sudo docker network connect net-2 explorer
```

Para iniciar los dockers utilizamos el comando sudo docker start "nombre del docker"

```
box@cristian-245:~$ sudo docker network connect net-1 explorer
box@cristian-245:~$ sudo docker network connect net-2 explorer
```

```
box@cristian-245:~$ sudo docker start nginx-contenedor
nginx-contenedor
box@cristian-245:~$ sudo docker start apache-contenedor
apache-contenedor
box@cristian-245:~$
```

- 2.6. Verifica les interfícies disponibles del docker

docker exec explorer ip -f inet -4 -o add

```
box@cristian-245:~$ sudo docker exec explorer ip -f inet -4 -o add
1: lo: inet 127.0.0.1/8 scope host lo\          valid_lft forever preferred_lft forever
55: eth0: inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0\          valid_lft forever preferred_lft forever
57: eth1: inet 10.0.10.3/24 brd 10.0.10.255 scope global eth1\          valid_lft forever preferred_lft forever
59: eth2: inet 10.0.20.3/24 brd 10.0.20.255 scope global eth2\          valid_lft forever preferred_lft forever
box@cristian-245:~$
```

Ahi vemos las subnets añadidas por nosotros mismos

### 3. Nmap 30%

- 3.1. Feu un escaneig del domini (escull un domini: [ibm.es](http://ibm.es), [movistar.es](http://movistar.es), [telefonica.es](http://telefonica.es)) i explica els resultats més rellevants:

Primero instalamos nmap

```
<html><body><h1>It works!</h1></body></html>
box@cristian-245:~$ sudo apt install nmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be inst
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host
box@cristian-245:~$ nmap movistar.es
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-30 19:12 UTC
Nmap scan report for movistar.es (81.47.192.13)
Host is up (0.014s latency).
rDNS record for 81.47.192.13: 13.red-81-47-192.staticip.rima-tde.net
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 4.87 seconds
box@cristian-245:~$
```

Podemos ver que el host está activo, podemos ver también los puertos que tienen y su estado además del tipo de protocolo. A parte también vemos la ip como es evidente.

- 3.2. Mostra las ip's activas cercanas al domini XXXX/28 seleccionat abans.

```
Nmap done: 1 IP address (1 host up) scanned in 4.87 seconds
box@cristian-245:~$ nmap -sL movistar.es/28
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-30 19:14 UTC
Nmap scan report for 0.red-81-47-192.staticip.rima-tde.net (81.47.192.0)
Nmap scan report for 1.red-81-47-192.staticip.rima-tde.net (81.47.192.1)
Nmap scan report for 2.red-81-47-192.staticip.rima-tde.net (81.47.192.2)
Nmap scan report for 3.red-81-47-192.staticip.rima-tde.net (81.47.192.3)
Nmap scan report for 4.red-81-47-192.staticip.rima-tde.net (81.47.192.4)
Nmap scan report for 5.red-81-47-192.staticip.rima-tde.net (81.47.192.5)
Nmap scan report for 6.red-81-47-192.staticip.rima-tde.net (81.47.192.6)
Nmap scan report for 7.red-81-47-192.staticip.rima-tde.net (81.47.192.7)
Nmap scan report for 8.red-81-47-192.staticip.rima-tde.net (81.47.192.8)
Nmap scan report for 9.red-81-47-192.staticip.rima-tde.net (81.47.192.9)
Nmap scan report for 10.red-81-47-192.staticip.rima-tde.net (81.47.192.10)
Nmap scan report for 11.red-81-47-192.staticip.rima-tde.net (81.47.192.11)
Nmap scan report for 12.red-81-47-192.staticip.rima-tde.net (81.47.192.12)
Nmap scan report for movistar.es (81.47.192.13)
rDNS record for 81.47.192.13: 13.red-81-47-192.staticip.rima-tde.net
Nmap scan report for 14.red-81-47-192.staticip.rima-tde.net (81.47.192.14)
Nmap scan report for 15.red-81-47-192.staticip.rima-tde.net (81.47.192.15)
Nmap done: 16 IP addresses (0 hosts up) scanned in 0.02 seconds
box@cristian-245:~$
```

Listo!

- 3.3. El domini **seleccionat** resuelve la IP XXXXXXXXX? . Indica a quien pertenece el servidor detrás de la IP?

```
sudo apt install whois
box@crístian-245:~$ sudo apt install whois -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Instalamos whois y ejecutamos el comando donde podemos ver que pertenece a Telefonica

```
box@crístian-245:~$ whois 81.47.192.13
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See https://apps.db.ripe.net/docs/HTML-Terms-And-Conditions
%
% Note: this output has been filtered.
% To receive output for a database update, use the "-B" flag.
%
% Information related to '81.47.192.0 - 81.47.192.255'
% Abuse contact for '81.47.192.0 - 81.47.192.255' is 'nemesys@telefonica.es'
inetnum:      81.47.192.0 - 81.47.192.255
netname:      RIMA
descr:        Telefonica de Espana SAU
descr:        Red de servicios IP
descr:        Spain
country:      ES
admin-c:      ATdE1-RIPE
tech-c:       TTdE1-RIPE
status:       ASSIGNED PA
remarks:      INFRA-AW
mnt-by:       MAINT-AS3352
created:      2003-04-28T10:42:14Z
last-modified: 2009-08-19T11:27:20Z
source:       RIPE # Filtered

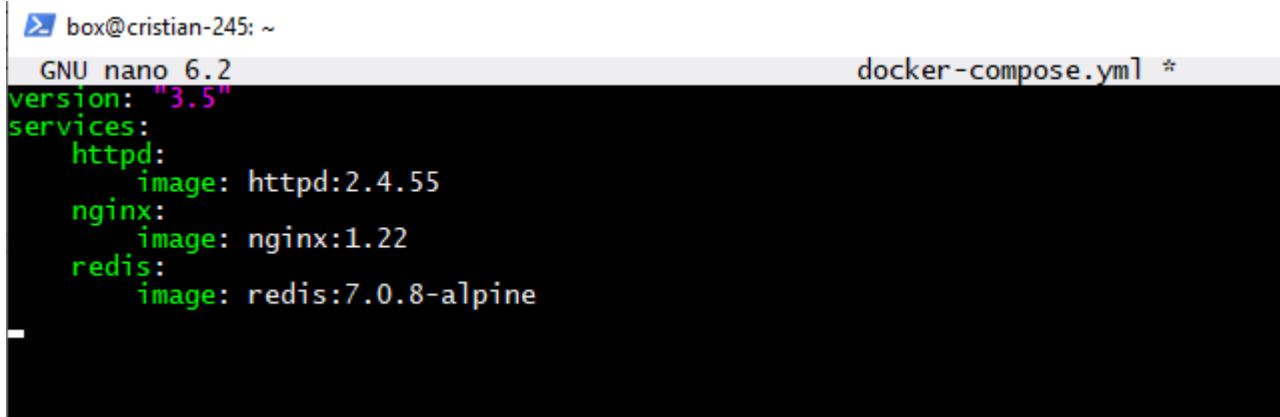
role:         Administradores Telefonica de Espana
address:      Ronda de la Comunicacion s/n
address:      Edificio Norte 1, planta 6
address:      28050 Madrid
address:      SPAIN
org:          ORG-TDE1-RIPE
admin-c:      KIX1-RIPE
tech-c:       TTdE1-RIPE
nic-hdl:      ATdE1-RIPE
mnt-by:       MAINT-AS3352
abuse-mailbox: nemesys@telefonica.es
created:      2006-01-18T12:24:41Z
last-modified: 2018-09-18T10:36:42Z
source:       RIPE # Filtered

role:         Tecnicos Telefonica de Espana
address:      Ronda de la Comunicacion S/N
address:      28050-MADRID
address:      SPAIN
org:          ORG-TDE1-RIPE
admin-c:      TTE2-RIPE
tech-c:       TTE2-RIPE
nic-hdl:      TTdE1-RIPE
```



### 3.4. - Crea aquest fitxer docker-compose.yml

```
version: "3.5"
services:
  httpd:
    image: httpd:2.4.55
  nginx:
    image: nginx:1.22
  redis:
    image: redis:7.0.8-alpine
```



The screenshot shows a terminal window with the prompt 'box@cristian-245: ~'. The nano text editor is open, editing 'docker-compose.yml'. The editor's status bar at the top shows 'GNU nano 6.2' and 'docker-compose.yml \*'. The content of the file is the same as the code block above, with syntax highlighting: 'version' is red, 'services' is green, and service names are green while their 'image' values are white.

Lo creamos con nano

- 3.5. Executa la comanda docker-compose up en el mateix directori per aixecar el serveis definits en la configuració.

```
box@cristian-245:~$ docker-compose up -d
Creating network "box_default" with the default driver
Pulling httpd (httpd:2.4.55)...
2.4.55: Pulling from library/httpd
3f9582a2cbe7: Pull complete
9423d69c3be7: Pull complete
d1f584c02b5d: Pull complete
8f73a485e312: Pull complete
b7697f8af320: Pull complete
Digest: sha256:83e99e7c437898cb564bbd3ceba7f1ea3f2d86e1cbd7a5324940086e59082f2b
Status: Downloaded newer image for httpd:2.4.55
Pulling nginx (nginx:1.22)...
1.22: Pulling from library/nginx
f1f26f570256: Pull complete
fd03b214f774: Pull complete
ef2fc869b944: Pull complete
ac713a9ef2cc: Pull complete
fd071922d543: Pull complete
2a9f38700bb5: Pull complete
Digest: sha256:fc5f5fb7574755c306aaf88456ebf0b006420a184d52b923d2f0197108f6b7
Status: Downloaded newer image for nginx:1.22
Pulling redis (redis:7.0.8-alpine)...
7.0.8-alpine: Pulling from library/redis
63b65145d645: Pull complete
6a83e1b979d3: Pull complete
33568fda55fd: Pull complete
92c907937b14: Pull complete
ae96d2ab3885: Pull complete
fbd5435e8d0e: Pull complete
Digest: sha256:28b21d9e1726c2ba89610897a9a765bcd83edcc55baf6b9e31bed80123670001
Status: Downloaded newer image for redis:7.0.8-alpine
Creating box_nginx_1 ... done
Creating box_httpd_1 ... done
Creating box_redis_1 ... done
box@cristian-245:~$
```

- 3.6. Mostra els dockers actius.

```
box@cristian-245:~$ docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED        STATUS        PORTS
9761d3e226b8   redis:7.0.8-alpine    "docker-entrypoint.s..." 28 seconds ago Up 24 seconds 6379/tcp
db0dbb6ae657   httpd:2.4.55          "httpd-foreground"       28 seconds ago Up 24 seconds 80/tcp
ed47e9ee8481   nginx:1.22            "/docker-entrypoint..." 28 seconds ago Up 24 seconds 80/tcp
3c7c92fbbf0f   httpd                  "httpd-foreground"       11 minutes ago Up 11 minutes 0.0.0.0:8001->80/tcp, :::8001->80/tcp
```

3.7. Obre un terminal interactiu en un contenidor i escaneja la xarxa privada

apt install -y nmap iproute2

```
Execute a Command in a Running Container
box@cristian-245:~$ docker exec -it box_nginx_1 bash

root@ed47e9ee8481:/# apt install -y nmap iproute2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dbus libapparmor1 libatm1 libblas3 libbpf0 libcap2 libcap2-bin libdbus-1-3 libelf1 liblinear4 liblua5.3-0 libmn10
  libpam-cap libpcap0.8 libxtables12 lua-lpeg nmap-common
Suggested packages:
  default-dbus-session-bus | dbus-session-bus iproute2-doc liblinear-tools liblinear-dev ncat ndiff zenmap
The following NEW packages will be installed:
  dbus iproute2 libapparmor1 libatm1 libblas3 libbpf0 libcap2 libcap2-bin libdbus-1-3 libelf1 liblinear4 liblua5.3-0
  libmn10 libpam-cap libpcap0.8 libxtables12 lua-lpeg nmap nmap-common
0 upgraded, 19 newly installed, 0 to remove and 29 not upgraded.
Need to get 8389 kB of archives.
After this operation, 33.4 MB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main amd64 libapparmor1 amd64 2.13.6-10 [99.3 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 libdbus-1-3 amd64 1.12.28-0+deb11u1 [223 kB]
0 upgraded, 0 newly installed, 0 to remove and 109 not upgraded.
box@cristian-245:~$ ip -brief addr | grep enp0s8
enp0s8 UP 192.168.56.26/24 fe80::a00:27ff:fe5f:ccf1/64
box@cristian-245:~$

root@ed47e9ee8481:/# nmap 192.168.56.26
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-30 19:26 UTC
Nmap scan report for cristian-245 (192.168.56.26)
Host is up (0.000018s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
8000/tcp   open  http-alt
8001/tcp   open  vcom-tunnel

Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
root@ed47e9ee8481:/#
```

3.8. Explica la topografia de la xarxa privada/24 (ip, port , servei). Omple la següent taula.

| Host Name    | Port     | IP            | Service     |  |
|--------------|----------|---------------|-------------|--|
| cristian-245 | 22/tcp   | 192.168.56.26 | ssh         |  |
| cristian-245 | 8000/tcp | 192.168.56.26 | http-alt    |  |
| cristian-245 | 8001/tcp | 192.168.56.26 | vcom-tunnel |  |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|--|--|--|--|--|

### 3.9. Elimina tots els contenidors

```
box@cristian-245:~$ sudo docker rm -f $(docker ps -aq)
9761d3e226b8
db0dbb6ae657
ed47e9ee8481
3c7c92fbbf0f
1606e2d76739
box@cristian-245:~$
```

## 4. TCPDUMP 20%

### 01 - tcpdump

#### 4.1. Mostra les interfícies que tens a la màquina.

```
box@cristian-245:~$ tcpdump -D
1.enp0s3 [Up, Running, Connected]
2.enp0s8 [Up, Running, Connected]
3.docker0 [Up, Running, Connected]
4.br-1491e557ae18 [Up, Running, Connected]
5.br-5948d55dd350 [Up, Running, Connected]
6.veth5b2d1e5 [Up, Running, Connected]
7.veth2e52da9 [Up, Running, Connected]
8.veth7df7d20 [Up, Running, Connected]
9.veth4d475da [Up, Running, Connected]
10.veth882b379 [Up, Running, Connected]
11.any (Pseudo-device that captures on all interfaces) [Up, Running]
12.lo [Up, Running, Loopback]
13.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
14.nflog (Linux netfilter log (NFLOG) interface) [none]
15.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
16.dbus-system (D-Bus system bus) [none]
17.dbus-session (D-Bus session bus) [none]
box@cristian-245:~$
```

#### 4.2. Obre un nou terminal i executa la comanda per veure tot el tràfic en una interfície

```
box@cristian-245:~$ sudo tcpdump -i enp0s3
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:54:02.824722 IP 192.168.124.182.51526 > 239.255.255.250.1900: UDP, length 175
18:54:02.904106 IP cristian-245.55360 > 192.168.124.5.domain: 54099+ [Iau] PTR? 250.255.255.239.in-addr.arpa. (57)
18:54:03.140800 Loopback, skipCount 0, invalid (256)
18:54:05.140986 Loopback, skipCount 0, invalid (256)
18:54:05.916188 ARP, Request who-has 192.168.124.128 tell 192.168.124.5, length 46
18:54:07.068717 ARP, Request who-has 192.168.124.5 tell cristian-245, length 28
18:54:07.069223 ARP, Reply 192.168.124.5 is-at 7a:11:5f:dd:9b:57 (oui Unknown), length 46
18:54:07.141068 Loopback, skipCount 0, invalid (256)
18:54:07.907929 ARP, Request who-has 192.168.124.6 tell cristian-245, length 28
18:54:07.908332 ARP, Reply 192.168.124.6 is-at d6:c5:e7:a3:d2:2d (oui Unknown), length 46
18:54:07.908338 IP cristian-245.47584 > 192.168.124.6.domain: 54099+ [Iau] PTR? 250.255.255.239.in-addr.arpa. (57)
18:54:07.943999 IP 192.168.124.6.domain > cristian-245.47584: 54099 NXDomain 0/1/1 (114)
18:54:07.944131 IP cristian-245.47584 > 192.168.124.6.domain: 54099+ PTR? 250.255.255.239.in-addr.arpa. (46)
18:54:07.944912 IP 192.168.124.6.domain > cristian-245.47584: 54099 NXDomain 0/1/0 (103)
18:54:07.945592 IP cristian-245.57390 > 192.168.124.6.domain: 19171+ [Iau] PTR? 182.124.168.192.in-addr.arpa. (57)
18:54:07.946397 IP 192.168.124.6.domain > cristian-245.57390: 19171 NXDomain* 0/1/1 (112)
18:54:07.946486 IP cristian-245.57390 > 192.168.124.6.domain: 19171+ PTR? 182.124.168.192.in-addr.arpa. (46)
18:54:07.947069 IP 192.168.124.6.domain > cristian-245.57390: 19171 NXDomain* 0/1/0 (101)
18:54:07.947771 IP cristian-245.59103 > 192.168.124.6.domain: 6545+ [Iau] PTR? 5.124.168.192.in-addr.arpa. (55)
18:54:07.948530 IP 192.168.124.6.domain > cristian-245.59103: 6545 NXDomain* 0/1/1 (110)
18:54:07.948692 IP cristian-245.59103 > 192.168.124.6.domain: 6545+ PTR? 5.124.168.192.in-addr.arpa. (44)
18:54:07.949251 IP 192.168.124.6.domain > cristian-245.59103: 6545 NXDomain* 0/1/0 (99)
18:54:07.950009 IP cristian-245.38061 > 192.168.124.6.domain: 43163+ [Iau] PTR? 69.124.168.192.in-addr.arpa. (56)
18:54:07.950625 IP 192.168.124.6.domain > cristian-245.38061: 43163 NXDomain* 0/1/1 (111)
18:54:07.950693 IP cristian-245.38061 > 192.168.124.6.domain: 43163+ PTR? 69.124.168.192.in-addr.arpa. (45)
18:54:07.951319 IP 192.168.124.6.domain > cristian-245.38061: 43163 NXDomain* 0/1/0 (100)
18:54:07.953150 IP cristian-245.59410 > 192.168.124.6.domain: 21664+ [Iau] PTR? 128.124.168.192.in-addr.arpa. (57)
18:54:07.953777 IP 192.168.124.6.domain > cristian-245.59410: 21664 NXDomain* 0/1/1 (112)
18:54:07.953899 IP cristian-245.59410 > 192.168.124.6.domain: 21664+ PTR? 128.124.168.192.in-addr.arpa. (46)
18:54:07.954513 IP 192.168.124.6.domain > cristian-245.59410: 21664 NXDomain* 0/1/0 (101)
18:54:07.956630 IP cristian-245.52666 > 192.168.124.6.domain: 58428+ [Iau] PTR? 6.124.168.192.in-addr.arpa. (55)
18:54:07.957257 IP 192.168.124.6.domain > cristian-245.52666: 58428 NXDomain* 0/1/1 (110)
18:54:07.957356 IP cristian-245.52666 > 192.168.124.6.domain: 58428+ PTR? 6.124.168.192.in-addr.arpa. (44)
18:54:07.958041 IP 192.168.124.6.domain > cristian-245.52666: 58428 NXDomain* 0/1/0 (99)
18:54:08.666654 IP 192.168.124.65.58069 > 239.255.255.250.1900: UDP, length 174
18:54:08.701744 IP cristian-245.38591 > 192.168.124.6.domain: 59037+ [Iau] PTR? 65.124.168.192.in-addr.arpa. (56)
18:54:08.702537 IP 192.168.124.6.domain > cristian-245.38591: 59037 NXDomain* 0/1/1 (111)
18:54:08.702642 IP cristian-245.38591 > 192.168.124.6.domain: 59037+ PTR? 65.124.168.192.in-addr.arpa. (45)
18:54:08.703551 IP 192.168.124.6.domain > cristian-245.38591: 59037 NXDomain* 0/1/0 (100)
18:54:09.141295 Loopback, skipCount 0, invalid (256)
18:54:09.163750 LLDP, length 46
18:54:09.667740 IP 192.168.124.65.58069 > 239.255.255.250.1900: UDP, length 174
18:54:10.668818 IP 192.168.124.65.58069 > 239.255.255.250.1900: UDP, length 174
18:54:11.141494 Loopback, skipCount 0, invalid (256)
18:54:11.670760 IP 192.168.124.65.58069 > 239.255.255.250.1900: UDP, length 174
```

Para ver el tráfico utilizamos “sudo tcpdump -i (interfaz)”

#### 4.3. Explica el resultat més importants del punt anterior

```
18:54:08.666654 IP 192.168.124.65.58069 > 239.255.255.250.1900: UDP, length 174
18:54:08.701744 IP cristian-245.38591 > 192.168.124.6.domain: 59037+ [Iau] PTR? 65.124.168.192.in-addr.arpa. (56)
18:54:08.702537 IP 192.168.124.6.domain > cristian-245.38591: 59037 NXDomain* 0/1/1 (111)
18:54:08.702642 IP cristian-245.38591 > 192.168.124.6.domain: 59037+ PTR? 65.124.168.192.in-addr.arpa. (45)
18:54:08.703551 IP 192.168.124.6.domain > cristian-245.38591: 59037 NXDomain* 0/1/0 (100)
18:54:09.141295 Loopback, skipCount 0, invalid (256)
18:54:09.163750 LLDP, length 46
18:54:09.667740 IP 192.168.124.65.58069 > 239.255.255.250.1900: UDP, length 174
18:54:10.668818 IP 192.168.124.65.58069 > 239.255.255.250.1900: UDP, length 174
18:54:11.141494 Loopback, skipCount 0, invalid (256)
18:54:11.670760 IP 192.168.124.65.58069 > 239.255.255.250.1900: UDP, length 174
```

Este comando nos deja ver información útil sobre el tráfico, como el protocolo, las ips de origen y destino y hora + fecha de salida...

- 4.4. En un altre terminal executa un nslookup a google.es:

```
Last login: Tue Jan 30 19:02:22 2024 from 192.168.56.
box@cristian-245:~$ nslookup google.es
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.es
Address: 142.250.184.163
Name:   google.es
Address: 2a00:1450:4003:80c::2003
box@cristian-245:~$
```

- 4.5. Executa la comanda que filtra el tràfic de totes les interfícies cuyo destí es dns  
excluyen el port ssh

```
box@cristian-245:~$ sudo tcpdump -i any not src port 22
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
18:58:35.333260 enp0s8 In IP 192.168.56.1.51528 > cristian-245.ssh: Flags [.], ack 2552362219, win 82
18:58:35.376256 enp0s8 In IP 192.168.56.1.51528 > cristian-245.ssh: Flags [.], ack 205, win 8209, len
18:58:35.434370 lo In IP localhost.53962 > localhost.domain: 53486+ [1au] PTR? 26.56.168.192.in-ad
18:58:35.434605 enp0s3 Out IP cristian-245.56135 > 192.168.124.6.domain: 19026+ [1au] PTR? 26.56.168.1
55)
```

Escribimos el comando tcpdump -i any not src port 22

diciendole que revise todo el trafico en el puerto 22, el puerto del SSH.