

# OFFSEC TOOLS

darodrig · dpuente- · juanrodr · acostal-

Cybersec · 42 Madrid

**Pequeño compendio de ejercicios relacionados con las redes y la ciberseguridad que podrían ser útiles más adelante. O no.**

## Contents

<b>Antes de empezar</b>	<b>3</b>
Requisitos . . . . .	3
Instrucciones . . . . .	3
Ex00 - Creando el entorno: . . . . .	4
<b>Ejercicios</b>	<b>5</b>
Ex01 . . . . .	5
Ex02 . . . . .	5
Ex03 . . . . .	5
Ex04 . . . . .	5
Ex05 . . . . .	6
Ex06 . . . . .	6
Ex07 . . . . .	6
Ex08 . . . . .	6
Pues ya estaría. . . . .	7
Bonus . . . . .	8
Ex09 . . . . .	8

## Antes de empezar

### Requisitos

- Docker (está en el *MSC*)
- Init\_docker.sh
- Suficiente espacio en disco
- Si Docker no arranca:

```
rm ~/Library/Group\ Containers/group.com.docker/settings.json  
rm ~/.docker
```

### Instrucciones

- Si un ejercicio contiene archivos, estos se encontrarán en la propia carpeta del ejercicio.
- Se sugieren ciertas herramientas, pero puedes utilizar cualquier otra.
- Si un ejercicio te resulta muy fácil, puedes ayudar a un compañero.
- Sentido común, Google, RTFM.

## Ex00 - Creando el entorno:

Iniciamos docker con `init__docker.sh`.

Nos situamos en la ruta donde está `docker-compose.yml` Lanzamos los servicios sobre los que trabajaremos:

```
docker-compose up -d
```

Creamos el Dockerfile con las herramientas necesarias:

---

```
cat << EOF > Dockerfile
FROM kalilinux/kali-rolling
RUN apt-get update --allow-insecure-repositories && \
    apt-get install -y --allow-unauthenticated \
        john \
        hashcat \
        file \
        wordlists \
        nmap \
        dirsearch \
        tcpdump \
        zip \
        gcc git make \
        curl whois \
        libreadline-dev
RUN mkdir -p /root/offsec
WORKDIR /root/offsec
EOF
```

---

Construimos la imagen y la ejecutamos, enlazando la carpeta actual como un volumen:

---

```
docker build -t kali-offsec .
docker run \
    -it \
    --mount type=bind,source="$(pwd)",target=/root/offsec/ \
    kali-offsec
```

---

Si necesitas abrir nuevas consolas puedes hacer (desde el propio mac):

```
docker exec -it $(docker ps | grep kali | cut -f1 -d ' ') bash
```

## Ejercicios

### Ex01

¿Están actualizados los servicios? ¿Cuál es la versión más reciente de cada uno de ellos? Averígualo usando `nmap` apuntando a tu **IP LOCAL**. (NO ES `LOCALHOST`, ya que cada contenedor tiene su propia `localhost`, si no algo como `10.11.XX.X`)

Guarda los resultados en un archivo `ex01/out.txt`.

### Ex02

Entra en `localhost:8080`. Utiliza `tcpdump` para capturar el tráfico al escuchar en ese puerto.

! Puedes lanzar la petición con `curl`.  
! ¿Desde dónde deberías capturar el tráfico?

Guarda el resultado en un archivo `ex02/out.pcap`.

¿Podrías capturar el tráfico de otro estudiante en el campus?

### Ex03

¿Es posible interpretar el tráfico del archivo `pcap`? Extrae en formato legible el contenido del `pcap`.

Guarda los resultados en `ex03/out.txt`.

### Ex04

Utiliza `dirsearch` para encontrar las urls disponibles en el puerto 8080.

! apunta a tu IP local != `localhost`

! no quieres ver todas las peticiones, solo las que resultan en una respuesta `http VÁLIDA`

! `dirsearch` incluye un diccionario por defecto, pero internet es amplio; en un futuro podrías armarte de diccionarios mejores.

Guarda los resultados en `ex04/out.txt`.

### Ex05

Intenta identificar qué algoritmo podría dar como resultado estos hashes. Puedes usar hashid, o mejor aún, **Houndsniff**.

```
a1d0c6e83f027327d8461063f4ac58a6
```

```
73475cb40a568e8da8a045ced110137e159f890ac4da883b6b17dc651b3a8049
```

¿Y qué diantres es esto?

```
YTFkMGM2ZTgzZjAyNzMyN2Q4NDYxMDYzZjRhYzU4YTY=
```

Guarda los resultados en `ex05/out.txt`.

### Ex06

Los archivos `data` y `data2` no tienen extensión. ¿Podrías recuperarla? Encuentra una herramienta para ello.

Guarda los resultados en `ex06/out.txt`.

### Ex07

- Lee sobre los archivos `/etc/passwd` y `/etc/shadow`.
- Crea un archivo que contenga 5 usuarios con sus respectivas contraseñas generadas con el algoritmo MD5. Las contraseñas serán lo suficientemente débiles como para poder hallarlas en un tiempo razonable, del orden de 6 caracteres.

```
user:passwd:uid:gid:gecos:home:shell
```

```
...
```

```
man mkpasswd
```

```
man john
```

- Detalla los pasos seguidos en `ex07/out.txt`

### Ex08

Con el comando `zip`, crea un archivo comprimido con contraseña.

```
man zip
```

Ahora busca la forma de hallar la contraseña del zip por fuerza bruta. Utiliza una contraseña breve y predecible por diccionario para tardar menos.

```
!! john2zip funciona con .zip > 4 KB
```

Guarda los resultados en `ex06/out.txt`.

**Pues ya estaría.**

Puedes detener todos los contenedores con

```
docker stop $(docker ps -q)
```

Puedes eliminar todos los contenedores con

```
docker rm $(docker ps -aq)
```

## **Bonus**

### **Ex09**

[ctf.hacku.org/register](https://ctf.hacku.org/register) ;)