

Laboratorio para pentesting con Docker

IES SAN VICENTE

TRABAJO DE FINAL DE GRADO

Administración de Sistemas Informáticos y Redes

Laboratorio para pentesting con Docker

Autor: Franco Matías Oscar Larrea

Tutor: Raquel Esteve

Indice

1 Abstract.....	3
2 Introducción.....	3
1 Objetivo.....	3
3 Plan de empresa.....	4
1 DAFO.....	4
2 Plan de tesorería.....	5
3 Cuenta de resultados.....	6
4 Balance de previsión.....	7
4 Análisis y diseño.....	8
1 Qué es una VPN.....	8
2 Qué es la virtualización.....	8
3 Qué es la contenerización.....	8
4 Esquema de despliegue de contenedores.....	9
5 Manuales de despliegue.....	10
1 Instalación del servidor Docker.....	10
2 Desplegando la primera VPN.....	11
3 Desplegando la primera máquina.....	12
4 Probando la primera VPN y máquina.....	13
5 Borrando la primera VPN y la primera máquina.....	18
6 Automatizando el despliegue de las VPN y máquinas.....	19
7 Añadiendo la imagen <i>Gift</i>	22
8 Vulnerando el contenedor <i>gift</i>	24
9 Publicando las imágenes en Dockerhub.....	29
10 Entorno gráfico y conexión por rdp.....	30
debian-rdp.dockerfile.....	30
kali-rdp.dockerfile.....	31
11 Probando las maquinas con entorno gráfico.....	32
12 Publicando el proyecto en GitHub.....	36
6 Webgrafia.....	37
1 Proyecto.....	37
2 Recursos.....	37
3 Documentación.....	37

1 Abstract

It is intended to develop the infrastructure to deploy, in a simple and fast way, controlled environments for cybersecurity testing.

These environments will be vulnerable machines containerized for pentest testing and 'Capture The Flag' games.

Some examples of these platforms:

- <https://tryhackme.com/>
- <https://hackmyvm.eu/>
- <https://overthewire.org/wargames/>
- <https://www.hackthebox.com/>

Unlike the mentioned pages this project will be opensource.

The idea is that by minimally modifying the code and having a server with Docker installed and some open ports, anyone can have their own lab, with user and machine management.

2 Introducción

Se pretende desarrollar la infraestructura para desplegar, de una forma sencilla y rápida, entornos controlados para pruebas de ciberseguridad.

Estos entornos serán máquinas vulnerables contenerizadas para hacer pruebas de pentest y juegos del estilo 'Capture The Flag'.

Algunos ejemplo de estas plataformas:

- <https://tryhackme.com/>
- <https://hackmyvm.eu/>
- <https://overthewire.org/wargames/>
- <https://www.hackthebox.com/>

A diferencia de las paginas mencionadas este proyecto será opensource.

La idea es que modificando mínimamente el código y teniendo un servidor con Docker instalado y unos puertos abiertos, cualquiera pueda tener su propio laboratorio, con gestión de usuarios y máquinas.

1 Objetivo

Desplegar una máquina vulnerable a petición del usuario y una VPN para que este pueda conectarse.

Tanto la máquina vulnerable como la VPN se tienen que crear cuando el usuario lo solicite, de la misma forma se deben eliminar.

Si hay distintos usuarios y distintas máquinas cada usuario debe de estar aislado en su red.

3 Plan de empresa

1 DAFO

El análisis DAFO es una herramienta que nos permite analizar de forma detallada cuales son los puntos fuertes y débiles de la empresa. Este análisis nos permitirá saber en qué aspectos debemos incidir para mejorar nuestra competitividad como empresa.

<p style="text-align: center;">DEBILIDADES</p> <ul style="list-style-type: none"> • Sin experiencia en el sector. • Disponemos por ahora de una plantilla escasa debido a los costes iniciales. • Si la empresa no consigue tener ganancias en un plazo de uno o dos años, peligrará el negocio. 	<p style="text-align: center;">AMENAZAS</p> <ul style="list-style-type: none"> • Siempre será necesario actualizarse ya que la era de la información está en continuo desarrollo. • Facilidad de entrada de más competidores.
<p style="text-align: center;">FORTALEZAS</p> <ul style="list-style-type: none"> • Calidad en el servicio al cliente. • El diseño, montaje y mantenimiento de las infraestructuras se pueden hacer no sólo de manera física, sino que además se puede hacer en la nube o de forma virtual. 	<p style="text-align: center;">OPORTUNIDADES</p> <ul style="list-style-type: none"> • Hay gran escasez de empresas reales de soluciones informáticas en la zona con una mano de obra especializada y rápida. • Este sector está en continuo crecimiento, por lo tanto el trabajo siempre se va aumentando.

2 Plan de tesorería

El plan de tesorería es el documento donde se indica las salidas y entradas del dinero previstas con el objetivo de poder prever si faltará o sobrará dinero.

PLAN DE TESORERÍA													
Partidas	Enero	Febr.	Mar.	Abril	Mayo	Junio	Julio	Agosto	Sept.	Oct.	Nov.	Dic.	Total
Capital inicial	25000	-	-	-	-	-	-	-	-	-	-	-	25000
Ventas	2000	2000	3300	4200	3100	500	750	1300	1100	1700	2150	2900	25000
Prest. Servicios	7500	3700	4500	5000	3000	5050	3500	2700	5000	4400	6700	10000	61050
Intereses Banco	25	25	25	25	25	25	25	25	25	25	25	25	300
Total cobros	34525	5725	7825	9225	6125	5575	4275	4025	6125	6125	8875	12925	111350
Compras	600	1000	2200	200	1400	800	1000	500	300	2000	2000	3000	15000
Sueldos y salarios	1600	1600	1600	1600	1600	1600	1600	1600	1600	1600	1600	1600	19200
Seguridad social	480	480	480	480	480	480	480	480	480	480	480	480	5760
Cuota de autónomo	120	120	120	120	120	120	120	120	120	120	120	120	1440
Arrendamientos y cánones	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	24000
Publicidad	800	800	800	800	1000	1000	1000	1000	1000	1500	1500	1500	12700
Suministros	100	-	-	-	100	-	-	-	100	-	-	-	300
Prima de seguros	25	-	-	25	-	-	25	-	-	25	-	-	100
Servicios de profesionales independientes	200	300	450	50	200	50	50	400	500	500	500	1000	4200
Gastos transportes	50	50	100	50	50	50	50	50	50	50	100	100	750
Fianza	600	-	-	-	-	-	-	-	-	-	-	-	600
Mobiliario	2000												2000
Maquinaria	14000												14000
Eq. procesos de información	10000												10000
Licencias	100												100
Aplicaciones Informáticas	200												200
Total de pagos	32875	6350	7750	5325	6950	6100	6325	6150	6150	8275	8300	9800	110350
Diferencia	1650	-625	75	3900	-825	-525	-2050	-2125	-25	-2150	575	3125	1000
Saldo acumulado	1650	1025	1100	5000	4175	3650	1600	-525	-550	-2700	-2125	1000	1000

3 Cuenta de resultados

La cuenta de resultados es el documento en el que se indican los beneficios y pérdidas generadas por la empresa teniendo en cuenta los ingresos y gastos.

Con los resultados obtenidos, podremos tomar las decisiones oportunas para mejorar la empresa.

PÉRDIDAS Y GANANCIAS			
INGRESOS DE EXPLOTACIÓN	86050	GASTOS DE EXPLOTACIÓN	86600
Venta de mercaderías	25000	Alquiler	24000
Ingresos por prestación de servicios	61050	Sueldos y Salarios	19200
		Cuota de Autónomos	1440
		Seguridad social a cargo empresa	5760
		Publicidad	12700
		Compra de mercaderías	15000
		Seguros	100
		Trabajos realizados por otras empresas	4200
		Fianza	600
		Amortización maquinaria	50
		Amortización Eq. Informáticos	2500
		Transporte	750
		Suministros	300
INGRESOS FINANCIEROS		GASTOS FINANCIEROS	
-	-	-	-
RESULTADOS DE EXPLOTACIÓN			(550)
RESULTADOS FINANCIEROS			0
RESULTADOS ANTES DE IMPUESTOS			(550)
RESULTADO DEL EJERCICIO			(550)

4 Balance de previsión

El balance de previsión es el documento donde se indica la contabilidad de los activos, pasivos y su diferencia.

Los resultados de este documento también indicarán la viabilidad económica.

BALANCE			
ACTIVO		PN + PASIVO	
ACTIVO NO CORRIENTE	21950	PATRIMONIO NETO	24450
Inmovilizado material	21650	Resultado del ejercicio	-550
Mobiliario	2000	Capital social	25000
Equipos para procesos de información	10000	Reserva legal	0
Maquinaria	14000	Reserva voluntaria	0
Amortización Acumulada	(2550)		
Inmovilizado intangible	300		
Licencias	100		
Aplicaciones informáticas	200		
ACTIVO CORRIENTE	4700	PASIVO NO CORRIENTE	1100
Disponible	0	Proveedores a l/p	1100
		PASIVO CORRIENTE	1100
Realizable	3700	Proveedores a c/p	1100
Deudores	3000		
Clientes	700		
Existencias	1000		
Mercaderías	1000		
TOTAL:	26650	TOTAL:	26650

4 Análisis y diseño

En este punto empecé a explorar las tecnologías optima para realizar este proyecto. Pero antes me hice algunas preguntas.

1 Qué es una VPN

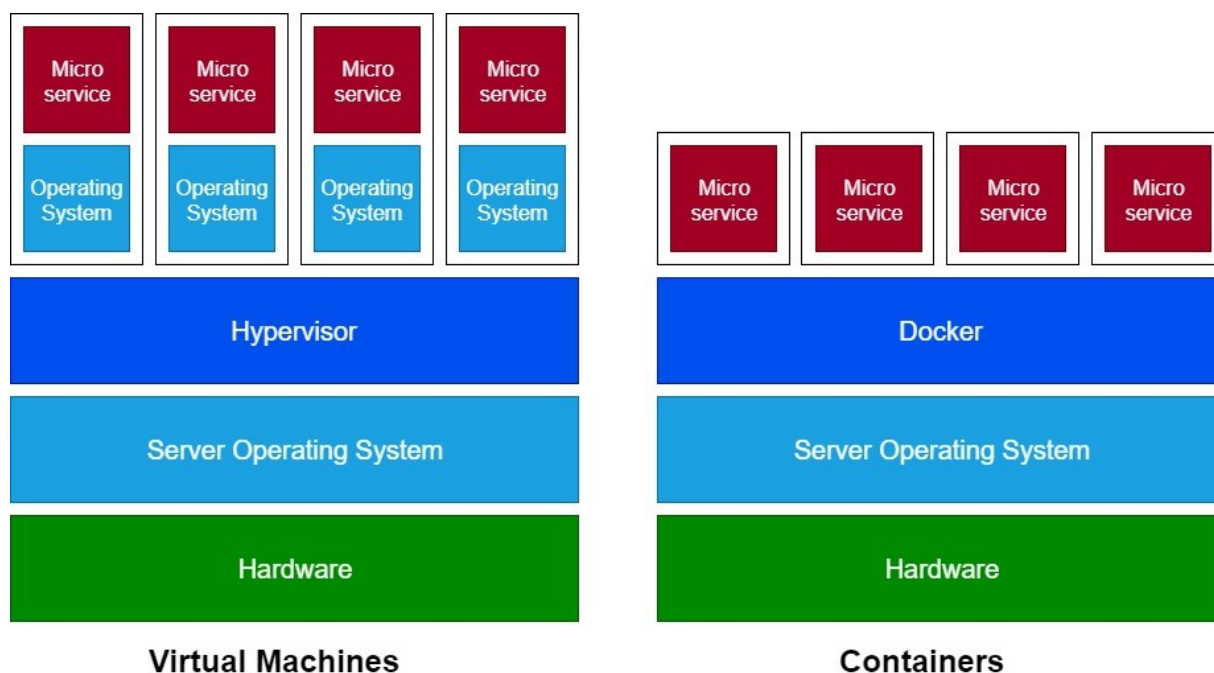
Una red privada virtual (RPV) (en inglés, Virtual Private Network, VPN) es una tecnología de red de ordenadores que permite una extensión segura de la red de área local (LAN) sobre una red pública o no controlada como Internet. Permite que el ordenador en la red envíe y reciba datos sobre redes compartidas o públicas como si fuera una red privada, con toda la funcionalidad, seguridad y políticas de gestión de una red privada. Esto se realiza estableciendo una conexión virtual punto a punto mediante el uso de conexiones dedicadas, cifrado o la combinación de ambos métodos.

2 Qué es la virtualización

En Informática, la virtualización utiliza el software para imitar las características del hardware y crear un sistema informático virtual. Esto permite a las organizaciones de TI ejecutar más de un sistema virtual, y múltiples sistemas operativos y aplicaciones, en un solo servidor.

3 Qué es la contenerización

La contenerización de aplicaciones es un método de virtualización de nivel de sistema operativo (nivel OS) para implementar y ejecutar aplicaciones distribuidas sin lanzar una máquina virtual completa (VM) para cada aplicación.



4 Esquema de despliegue de contenedores

Supongamos que tenemos las siguientes maquinas y los siguientes usuarios.

Usuarios:

- pepe
- pato
- fmol

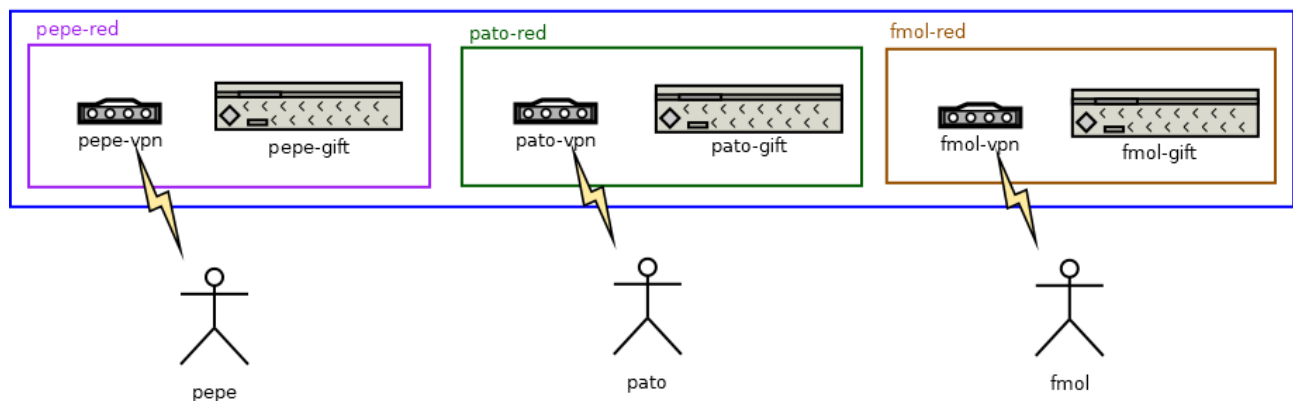
Maquinas:

- Gift
- Metasploitable2

Y para cada usuario tenemos que: crear una red, desplegar la VPN Wireguard y desplegar el contenedor que haya elegido.

Si los usuarios *pepe* y *pato* encendieran sus maquinas Gift. Y el usuario *fmol* encendiera su maquina Metasploitable2 se generaria una serie de redes, VPNs y contenedores como se muestra en el siguiente esquema.

Servidor Docker en Azure



5 Manuales de despliegue

1 Instalación del servidor Docker

Primero desplegamos una máquina virtual en Azure con la [CLI de Azure](#).

```
az vm create --resource-group hackf_group --name hackf-vmachine --image  
Debian --size Standard_B2s --admin-username fmo1 --generate-ssh-keys
```

Abrimos los puertos para la VPN.

```
az vm open-port --port 51820-51850 --resource-group hackf_group --name  
hackf-vmachine
```

La cantidad de puertos para la VPN abiertos son el numero de usuarios que vamos a poder tener conectados a la VPN simultaneamente.

Nos conectamos por SSH a la máquina virtual e instalamos Docker.

```
ssh fmo1@52.142.216.158  
sudo apt update && sudo apt install docker.io docker-compose -y  
sudo usermod -aG docker $(whoami)
```

2 Desplegando la primera VPN

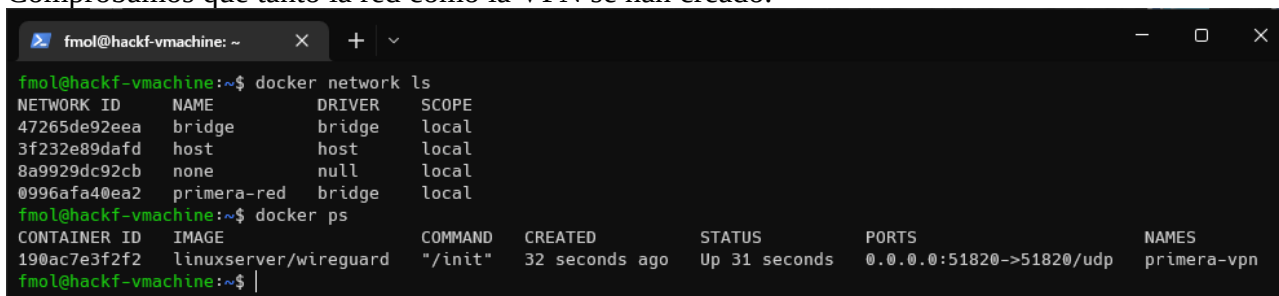
Antes de desplegar la VPN vamos a crear una red exclusiva para esta primera VPN y la primera máquina.

```
docker network create primera-red
```

Utilizaré [Wireguard](#) como VPN, es software libre y de código abierto.

```
docker run -d \
  --network=primera-red \
  --name=primera-VPN \
  --cap-add=NET_ADMIN \
  --cap-add=SYS_MODULE \
  -e PUID=1000 \
  -e PGID=1000 \
  -e TZ=Europe/Madrid \
  -e SERVERURL=52.142.216.158 \
  -e SERVERPORT=51820 \
  -e PEERS=1 \
  -e PEERDNS=auto \
  -e INTERNAL_SUBNET=10.10.10.0 \
  -p 51820:51820/udp \
  -v /wireguard/primera-VPN:/config \
  -v /lib/modules:/lib/modules \
  -v /usr/src:/usr/src \
  --sysctl="net.ipv4.conf.all.src_valid_mark=1" \
  --restart unless-stopped \
  linuxserver/wireguard
```

Comprobamos que tanto la red como la VPN se han creado.



```
fmol@hackf-vmachine: ~$ docker network ls
NETWORK ID      NAME      DRIVER  SCOPE
47265de92eea    bridge    bridge  local
3f232e89dafd    host      host    local
8a9929dc92cb    none      null    local
0996afa40ea2    primera-red  bridge  local
fmol@hackf-vmachine:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
190ac7e3f2f2   linuxserver/wireguard  "/init"                  32 seconds ago  Up 31 seconds  0.0.0.0:51820->51820/udp  primera-vpn
fmol@hackf-vmachine:~$
```

Los datos de configuración para conectarse a la VPN se guardan en `/wireguard/primera-VPN/peer1/`.

Esta ruta la especificamos en el comando que inicia la VPN.

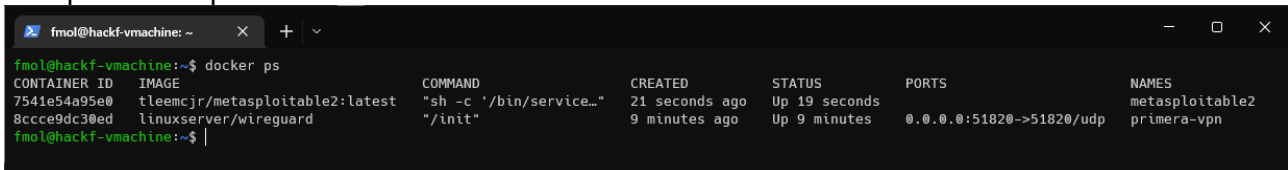
3 Desplegando la primera máquina

Una de las máquinas que usaremos en este proyecto es la conocida [Metasploitable2](#).

La desplegaremos en la misma red que la VPN.

```
docker run --rm -dit \
  --name metasploitable2 \
  --network=primera-red \
  tleemcjr/metasploitable2:latest
```

Comprobamos que se ha creado correctamente.

A terminal window titled 'fmol@hackf-vmachine: ~' showing the output of the 'docker ps' command. The output is a table with columns: CONTAINER ID, IMAGE, COMMAND, CREATED, STATUS, PORTS, and NAMES. Two containers are listed: 'metasploitable2' and 'primera-vpn'.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7541e54a95e0	tleemcjr/metasploitable2:latest	"sh -c '/bin/service..."	21 seconds ago	Up 19 seconds		metasploitable2
8ccce9dc30ed	linuxserver/wireguard	"/init"	9 minutes ago	Up 9 minutes	0.0.0.0:51820->51820/udp	primera-vpn

4 Probando la primera VPN y máquina

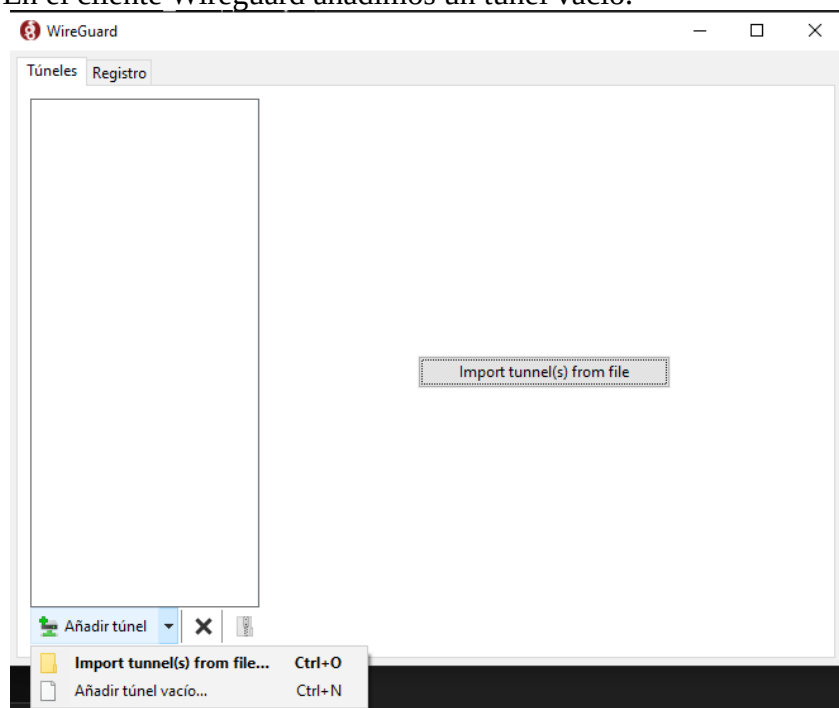
Vamos a conectarnos a la VPN. Para ello nos descargamos el cliente Wireguard para nuestro sistema operativo y nos conectamos con los datos de configuración.

```
cat /wireguard/primera-VPN/peer1/peer1.conf
```

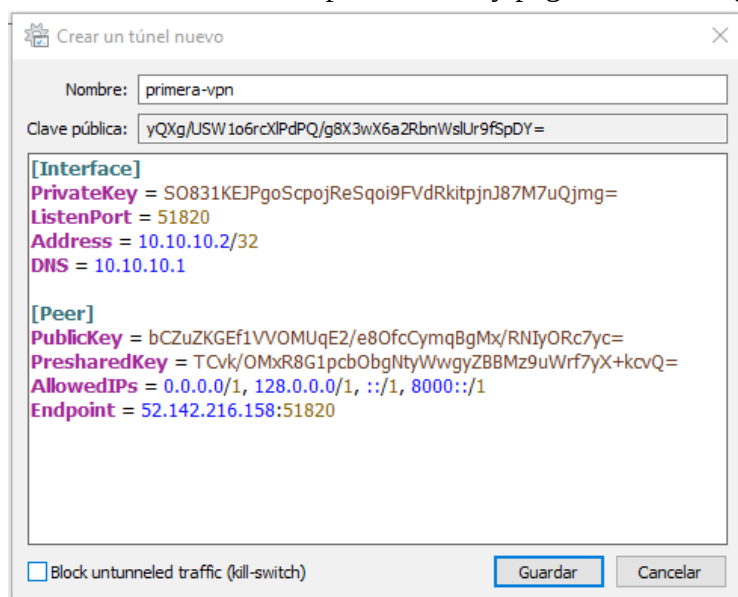
```
[Interface]
Address = 10.10.10.2
PrivateKey = SO831KEJPgoScpojReSqoi9FVdRkitpjnJ87M7uQjmg=
ListenPort = 51820
DNS = 10.10.10.1

[Peer]
PublicKey = bCZuZKGEf1VVOMUqE2/e8OfcCymqBgMx/RNIyORc7yc=
PresharedKey = TCvk/OMxR8G1pcbObgNtyWwgyZBBMz9uWrf7yX+kcvQ=
Endpoint = 52.142.216.158:51820
AllowedIPs = 0.0.0.0/0, ::/0
```

En el cliente Wireguard añadimos un tunel vacío.

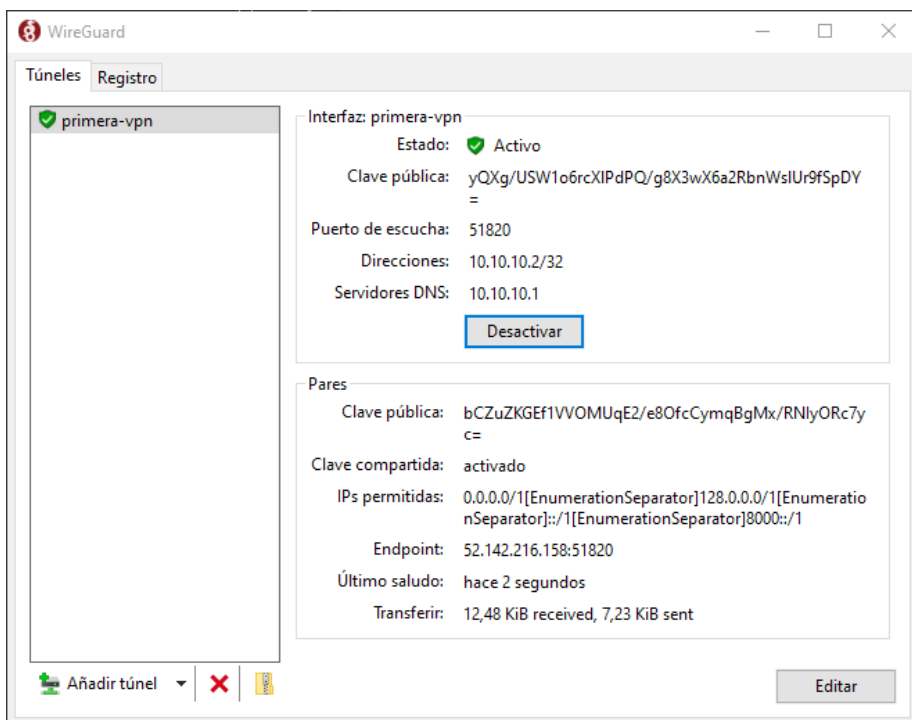


Introducimos un nombre para la VPN y pegamos la configuración.



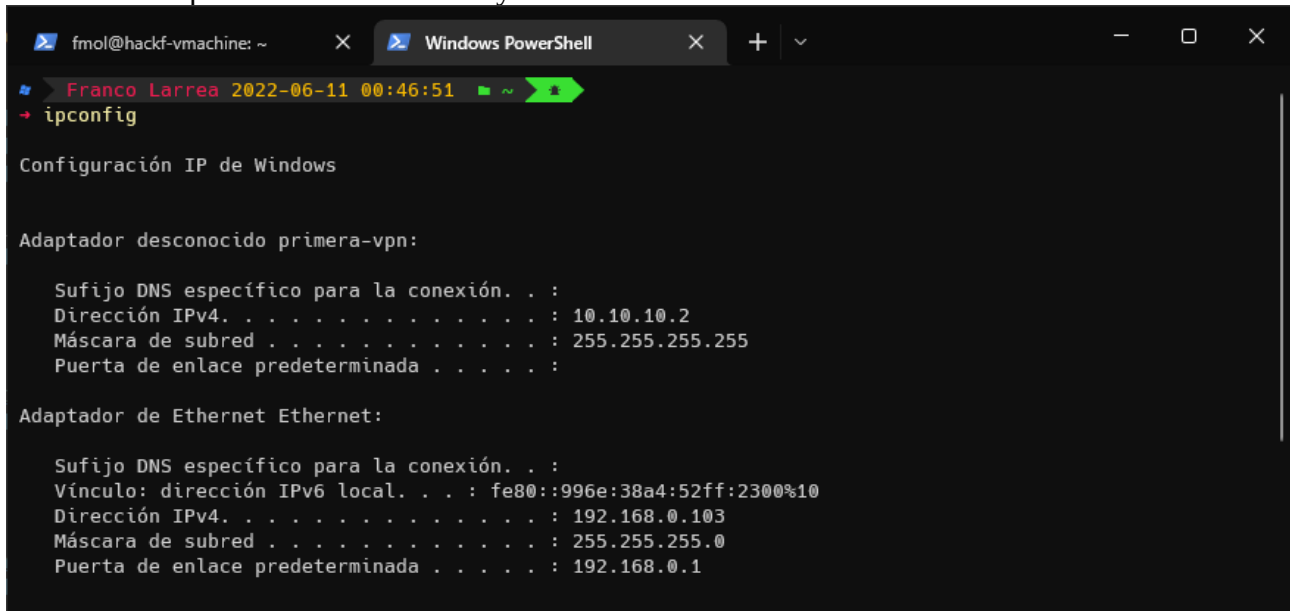
Es importante desmarcar la casilla “*Block untunneled traffic*”.
Guardamos la configuración.

Ahora activamos la VPN.



Vamos a comprobar que efectivamente estamos conectados y que tenemos visibilidad con la máquina Metasploitable2.

Podemos ver que estamos conectados y tenemos la IP 10.10.10.2.



```
fmol@hackf-vmachine: ~ X Windows PowerShell X + v
Franco Larrea 2022-06-11 00:46:51 ~
ipconfig

Configuración IP de Windows

Adaptador desconocido primera-vpn:

    Sufixo DNS específico para la conexión. . . :
    Dirección IPv4. . . . . : 10.10.10.2
    Máscara de subred . . . . . : 255.255.255.255
    Puerta de enlace predeterminada . . . . . :

Adaptador de Ethernet Ethernet:

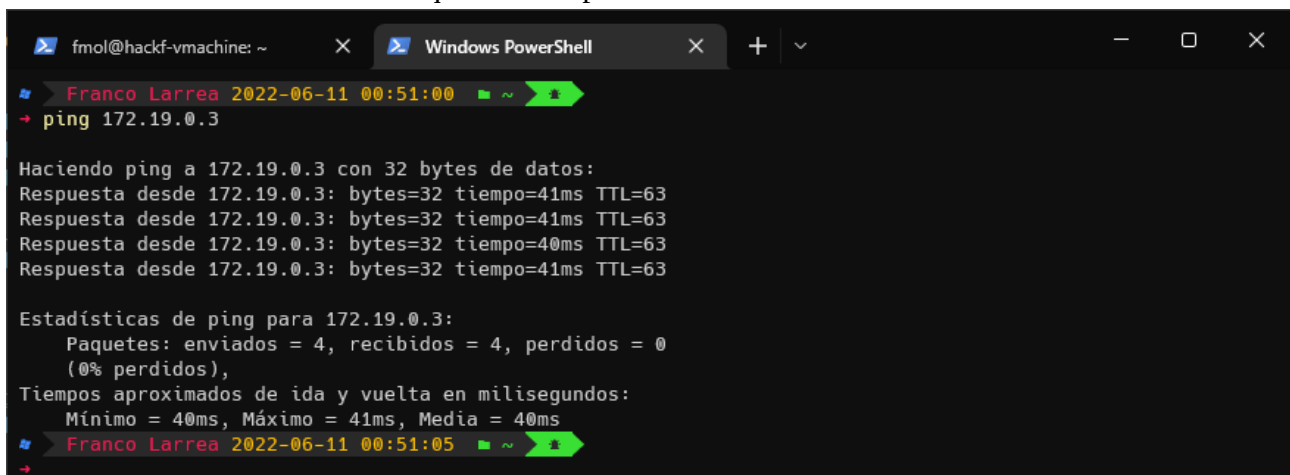
    Sufixo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . : fe80::996e:38a4:52ff:2300%10
    Dirección IPv4. . . . . : 192.168.0.103
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 192.168.0.1
```

`docker inspect metasploitable2 | grep IPAddress | tail -n1`

```
"IPAddress": "172.18.0.3",
```

La máquina Metasploitable2 tiene la IP 172.18.0.3.

Tenemos conectividad con la máquina Metasploitable2.

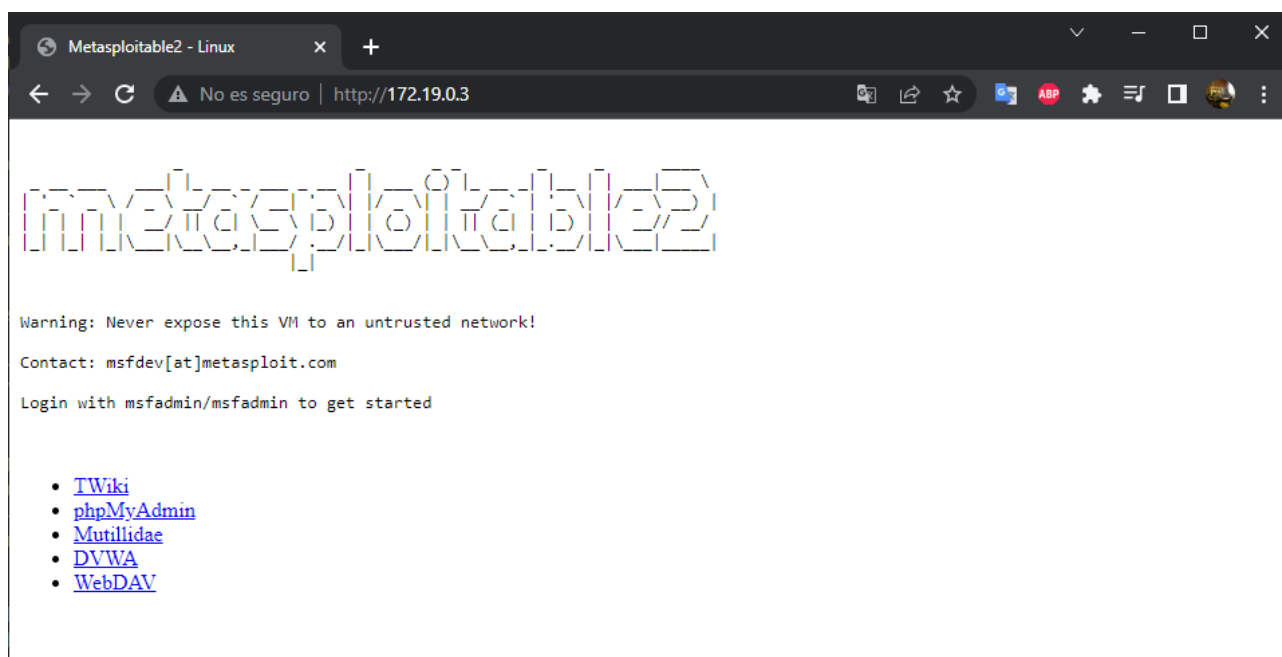


```
fmol@hackf-vmachine: ~ X Windows PowerShell X + v
Franco Larrea 2022-06-11 00:51:00 ~
ping 172.19.0.3

Haciendo ping a 172.19.0.3 con 32 bytes de datos:
Respuesta desde 172.19.0.3: bytes=32 tiempo=41ms TTL=63
Respuesta desde 172.19.0.3: bytes=32 tiempo=41ms TTL=63
Respuesta desde 172.19.0.3: bytes=32 tiempo=40ms TTL=63
Respuesta desde 172.19.0.3: bytes=32 tiempo=41ms TTL=63

Estadísticas de ping para 172.19.0.3:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 40ms, Máximo = 41ms, Media = 40ms
Franco Larrea 2022-06-11 00:51:05 ~
```

Laboratorio para pentesting con Docker



Desde un Parrot OS vamos a hacer un escaneo de puertos con nmap.

nmap -p0-65535 172.18.0.3

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-11 01:05 CEST
Nmap scan report for metasploitable2.primera-red (172.18.0.3)
Host is up (0.063s latency).
Not shown: 65511 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
```


Laboratorio para pentesting con Docker

```
8180/tcp open unknown
8787/tcp open msgsrvr
35835/tcp open unknown
```

```
Nmap done: 1 IP address (1 host up) scanned in 44.88 seconds
```

Vamos a conectarnos por SSH con las credenciales por defecto msfadmin:msfadmin.

ssh msfadmin@172.18.0.3

```
The authenticity of host '172.18.0.3 (172.18.0.3)' can't be established.
RSA key fingerprint is
SHA256:BQHm5EoHX9GCIOLuVscegPXLQOsuPs+E9d/rrJB84rk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.18.0.3' (RSA) to the list of known
hosts.
```

```
msfadmin@172.18.0.3's password:
```

```
Linux 32554753bfe5 4.13.0-21-generic #24-Ubuntu SMP Mon Dec 18 17:29:16
UTC 2017 x86_64
```

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

```
To access official Ubuntu documentation, please visit:
```

```
http://help.ubuntu.com/
```

```
No mail.
```

```
Last login: Sun Jul 16 21:04:01 2017
```

```
msfadmin@9392ea93dcdf:~$
```

msfadmin@9392ea93dcdf:~\$ id

```
uid=1000(msfadmin) gid=1000(msfadmin)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(vide
o),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(ms
fadmin)
```

```
Después de esto hemos comprobado que tenemos conectividad con la máquina
y que la máquina Metasploitable2 ofrece casi los mismos servicios que
una virtualizada.
```

```
Según el creador de la imagen faltan los siguientes servicios: Bind9,
NSF y Klogd.
```

5 Borrando la primera VPN y la primera máquina

Vamos a parar y eliminar los contenedores creados. También vamos a eliminar la red.

Eliminamos la máquina. (No hay que borrarla ya que se lanzó con el parámetro --rm)

```
docker stop metasploitable2
```

Eliminamos la VPN.

```
docker stop primera-VPN && docker rm primera-VPN
```

Eliminamos la red.

```
docker network rm primera-red
```

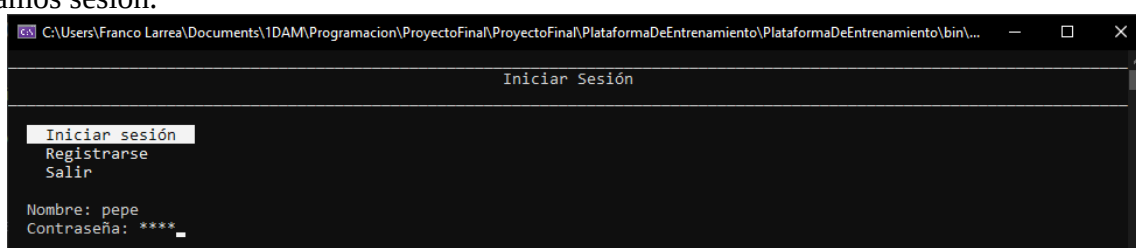
6 Automatizando el despliegue de las VPN y máquinas

Ahora que sabemos como desplegar y eliminar este conjunto de *red-vpn-máquina* podemos automatizar su despliegue.

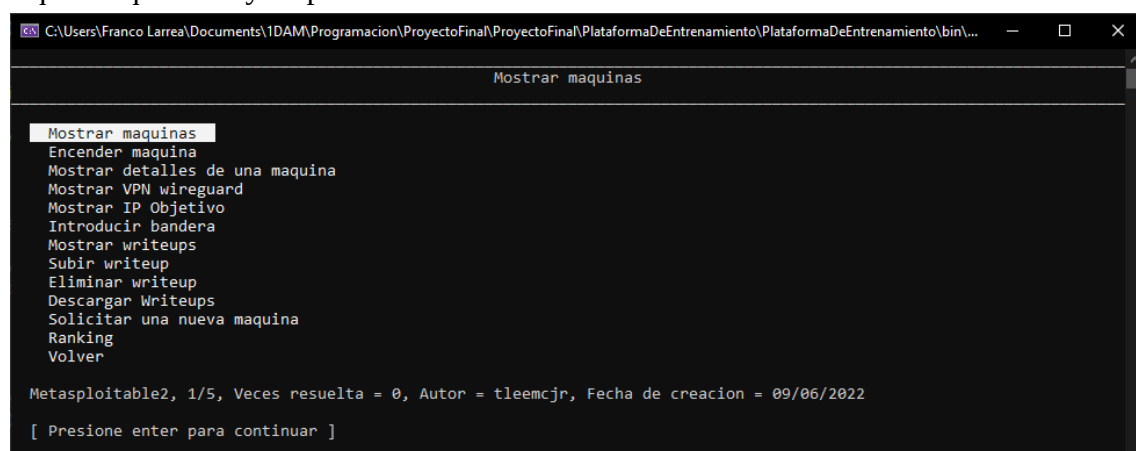
Aunque sabemos como lanzar los contenedores y es relativamente sencillo, también hay que decir que es algo tedioso, hay que añadirle algo de lógica para que un usuario pueda ejecutar todo esto de forma transparente y lo más importante, para que sea automático.

Para esto hemos desarrollado una aplicación de consola en C#. La aplicación nos permite administrar los usuarios, las máquinas y el despliegue de estas.

Iniciamos sesión.

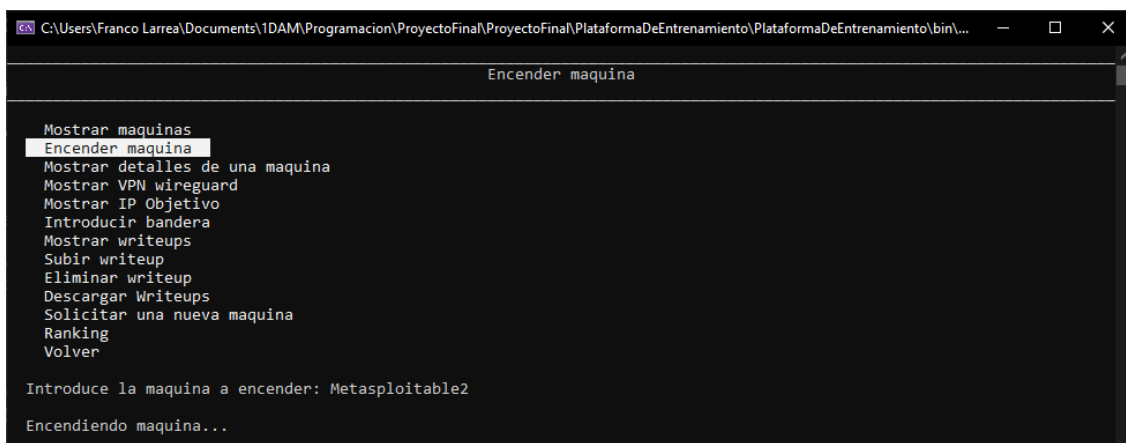


Vemos que máquinas hay disponibles.



Encendemos la máquina Metasploitable2.

Laboratorio para pentesting con Docker



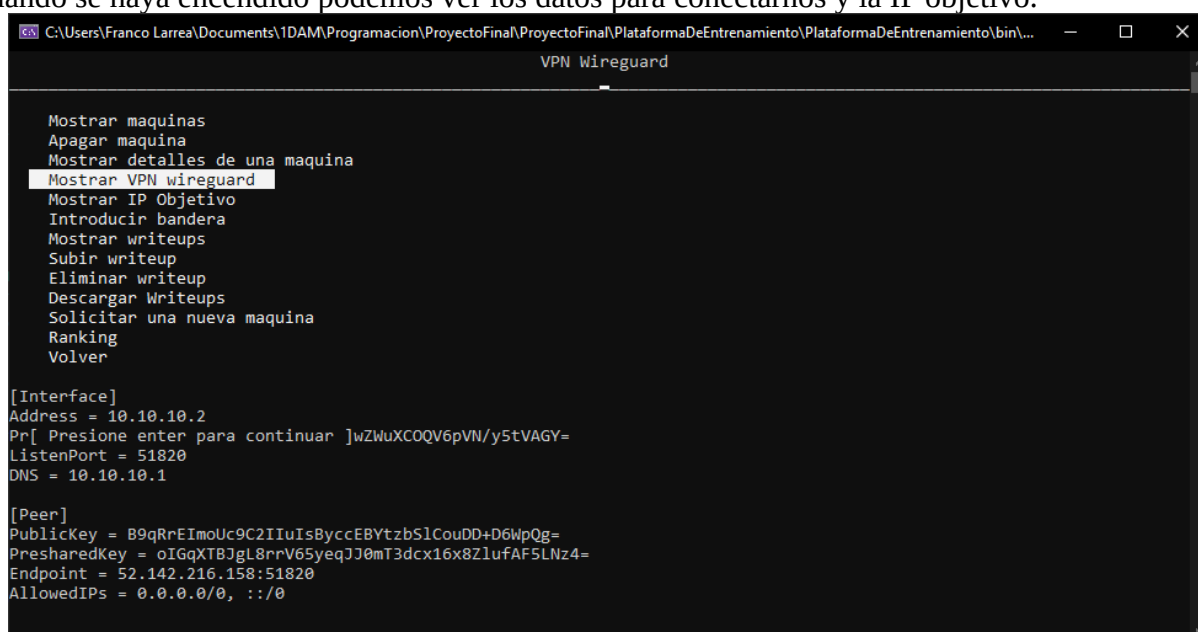
```
C:\Users\Franco Larrea\Documents\1DAM\Programacion\ProyectoFinal\ProyectoFinal\PlataformaDeEntrenamiento\PlataformaDeEntrenamiento\bin\...
Encender maquina

Mostrar maquinas
Encender maquina
Mostrar detalles de una maquina
Mostrar VPN wireguard
Mostrar IP Objetivo
Introducir bandera
Mostrar writeups
Subir writeup
Eliminar writeup
Descargar Writeups
Solicitar una nueva maquina
Ranking
Volver

Introduce la maquina a encender: Metasploitable2

Encendiendo maquina...
```

Cuando se haya encendido podemos ver los datos para conectarnos y la IP objetivo.

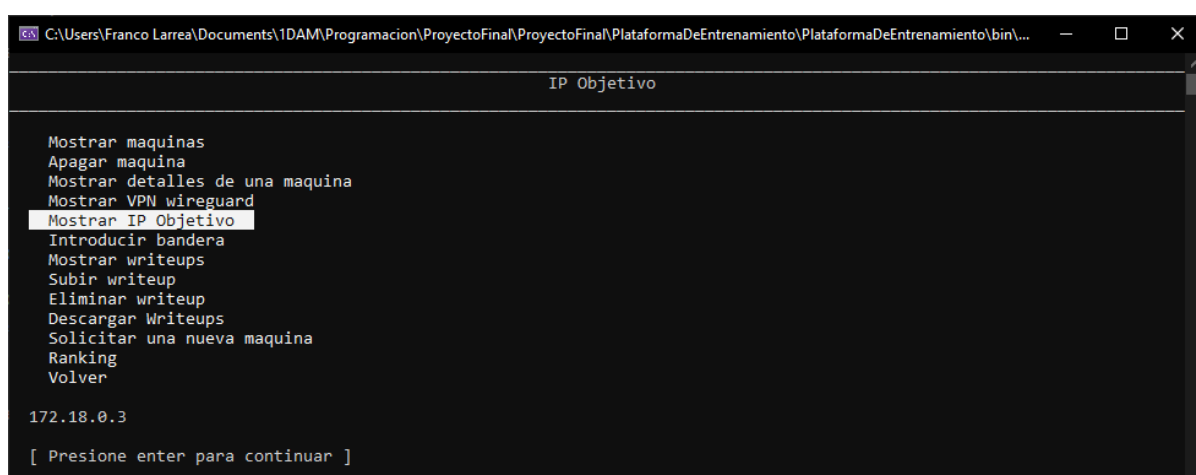


```
C:\Users\Franco Larrea\Documents\1DAM\Programacion\ProyectoFinal\ProyectoFinal\PlataformaDeEntrenamiento\PlataformaDeEntrenamiento\bin\...
VPN Wireguard

Mostrar maquinas
Apagar maquina
Mostrar detalles de una maquina
Mostrar VPN wireguard
Mostrar IP Objetivo
Introducir bandera
Mostrar writeups
Subir writeup
Eliminar writeup
Descargar Writeups
Solicitar una nueva maquina
Ranking
Volver

[Interface]
Address = 10.10.10.2
Pr[ Presione enter para continuar ]wZWuXCOQV6pVN/y5tVAGY=
ListenPort = 51820
DNS = 10.10.10.1

[Peer]
PublicKey = B9qRrEImoUc9C2IIuIsByccEBYtzbS1CouDD+D6WpQg=
PresharedKey = oIGqXTBJgl8rrV65yeqJJ0mT3dcx16x8ZlufAF5LNz4=
Endpoint = 52.142.216.158:51820
AllowedIPs = 0.0.0.0/0, ::/0
```



```
C:\Users\Franco Larrea\Documents\1DAM\Programacion\ProyectoFinal\ProyectoFinal\PlataformaDeEntrenamiento\PlataformaDeEntrenamiento\bin\...
IP Objetivo

Mostrar maquinas
Apagar maquina
Mostrar detalles de una maquina
Mostrar VPN wireguard
Mostrar IP Objetivo
Introducir bandera
Mostrar writeups
Subir writeup
Eliminar writeup
Descargar Writeups
Solicitar una nueva maquina
Ranking
Volver

172.18.0.3

[ Presione enter para continuar ]
```

Desde el servidor Docker comprobamos que la máquina se ha creado con su respectiva VPN y red.

Laboratorio para pentesting con Docker

```
fmol@hackf-vmachine: ~  
Every 1.0s: docker ps -a && echo "" && docker network ls  
hackf-vmachine: Sat Jun 11 01:29:48 2022  


| CONTAINER ID | IMAGE                           | COMMAND                  | CREATED       | STATUS       | PORTS                    | NAMES                |
|--------------|---------------------------------|--------------------------|---------------|--------------|--------------------------|----------------------|
| eb19335abfce | tleemcjr/metasploitable2:latest | "sh -c '/bin/service..." | 8 minutes ago | Up 8 minutes |                          | pepe-Metasploitable2 |
| 0dfa927248ea | linuxserver/wireguard           | "/init"                  | 8 minutes ago | Up 8 minutes | 0.0.0.0:51820->51820/udp | pepe-vpn             |


| NETWORK ID   | NAME     | DRIVER | SCOPE |
|--------------|----------|--------|-------|
| 873b36cd7ba7 | bridge   | bridge | local |
| 3f232e89dafd | host     | host   | local |
| 8a9929dc92cb | none     | null   | local |
| 40d53dc0b619 | pepe-red | bridge | local |


```

Apagamos la máquina.

```
C:\Users\Franco Larrea\Documents\1DAM\Programacion\ProyectoFinal\ProyectoFinal\PlataformaDeEntrenamiento\PlataformaDeEntrenamiento\bin\...  
Apagar maquina  
  
Mostrar maquinas  
Apagar maquina  
Mostrar detalles de una maquina  
Mostrar VPN wireguard  
Mostrar IP Objetivo  
Introducir bandera  
Mostrar writeups  
Subir writeup  
Eliminar writeup  
Descargar writeups  
Solicitar una nueva maquina  
Ranking  
Volver  
  
¿Parar la maquina Metasploitable2? [s/N]: s  
Apagando maquina...
```

Desde el servidor Docker comprobamos que se ha eliminado correctamente.

```
fmol@hackf-vmachine: ~  
Every 1.0s: docker ps -a && echo "" && docker network ls  
hackf-vmachine: Sat Jun 11 01:31:48 2022  


| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES |
|--------------|-------|---------|---------|--------|-------|-------|
|--------------|-------|---------|---------|--------|-------|-------|


| NETWORK ID   | NAME   | DRIVER | SCOPE |
|--------------|--------|--------|-------|
| 873b36cd7ba7 | bridge | bridge | local |
| 3f232e89dafd | host   | host   | local |
| 8a9929dc92cb | none   | null   | local |


```

7 Añadiendo la imagen *Gift*

Ahora que tenemos esta aplicación para que nos gestione el despliegue de las máquinas vamos a crear una imagen en docker.

Esta será una máquina vulnerable muy sencilla.

Consistirá en un servidor web que liste los directorios desde la raíz y un servidor ssh.

nano gift.dockerfile

```
FROM debian:stable

# Update system
RUN apt update && apt upgrade -y

# Install ssh & apache2
RUN apt install openssh-server apache2 -y

# Configure ssh
RUN mkdir /root/.ssh && ssh-keygen -t rsa -N "" -f /root/.ssh/id_rsa
RUN cat /root/.ssh/id_rsa.pub > /root/.ssh/authorized_keys
RUN chmod -R 755 /root/

# Configure apache2
RUN echo '<Directory />' >> /etc/apache2/conf-available/security.conf \
    && echo '  Options Indexes MultiViews' >> /etc/apache2/conf-
available/security.conf \
    && echo '  AllowOverride None' >>
/etc/apache2/conf-available/security.conf \
    && echo '  Require all granted' >>
/etc/apache2/conf-available/security.conf \
    && echo '</Directory>' >> /etc/apache2/conf-available/security.conf

RUN sed -i 's/DocumentRoot \/var\/www\/html/DocumentRoot \/g'
/etc/apache2/sites-available/000-default.conf

# Set user & root flags
RUN mkdir /root/.flags/

RUN echo 'this_is_a_gift' > /root/.flags/user.txt
RUN echo 'gg_well_played' > /root/.flags/root.txt
RUN echo 'Look for the "user.txt" and "root.txt" flags.' >
/root/info.txt

CMD service ssh start && service apache2 start && tail -f /dev/null
```

Creamos la imagen.

```
docker build . -f gift.dockerfile -t gift
```

Laboratorio para pentesting con Docker

Comprobamos que se haya creado.

```
fmol@hackf-vmachine: ~/gift x + v
fmol@hackf-vmachine:~/gift$ docker image ls gift
REPOSITORY TAG IMAGE ID CREATED SIZE
gift latest 292a769aa4f2 5 minutes ago 286MB
fmol@hackf-vmachine:~/gift$ |
```

Iniciamos sesión como administrador y añadimos la máquina al programa.

```
C:\Users\Franco Larrea\Documents\TDAM\Programacion\ProyectoFinal\ProyectoFinal\PlataformaDeEntrenamiento\PlataformaDeEntrenamiento\bin\... - □ X
Añadir maquina

Mostrar maquinas activas
Parar todas las maquinas
System Prune
Añadir maquina
Eliminar maquina
Mostrar solicitudes para nuevas maquinas
Volver

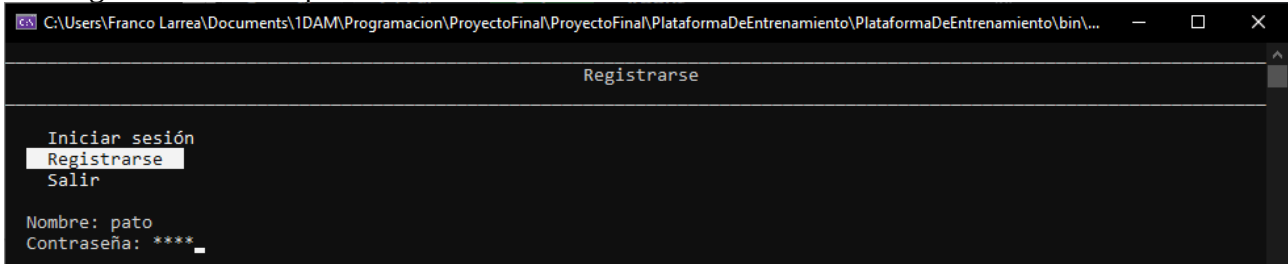
Nombre: Gift
Dificultad: 1
Autor: fmol
Bandera usuario: this_is_a_gift
Bandera root: gg_well_played

Hay que añadir lo siguiente al comando:
' --name <NOMBREMAQUINA> --network=<NETWORK> '
Comando docker:
docker run -dit --name <NOMBREMAQUINA> --network=<NETWORK> gift
```

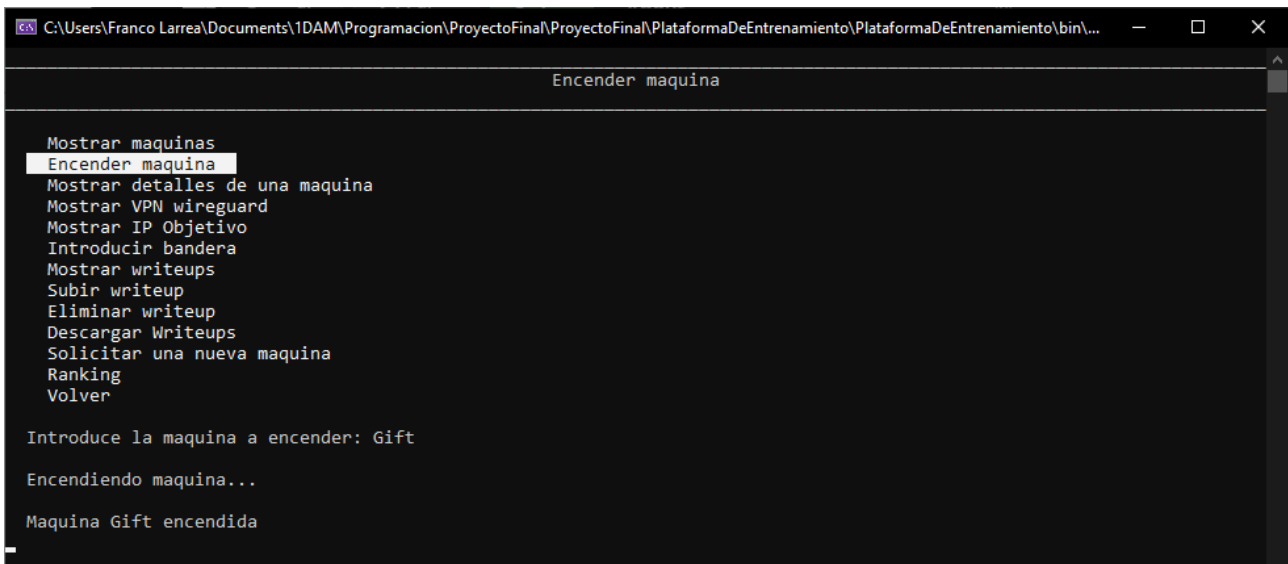
8 Vulnerando el contenedor *gift*

Vamos a intentar vulnerar el contenedor gift y comprobar que se ha configurado *incorrectamente* antes de subirlo a Dockerhub.

Nos registramos en la aplicación.



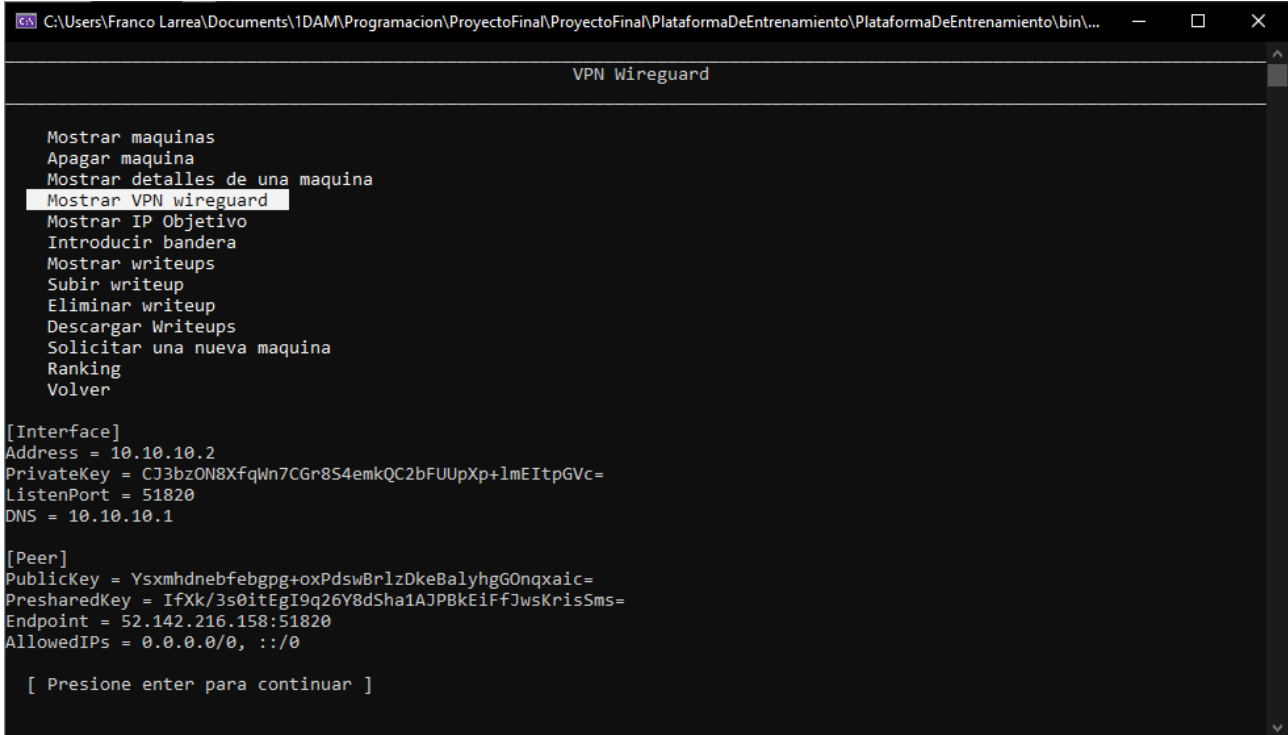
Encendemos la máquina Gift



Laboratorio para pentesting con Docker

Desde un Parrot OS nos conectamos a la VPN.

Primero guardamos la configuración en `/etc/wireguard/pato-vpn.conf`.



```
C:\Users\Franco Larrea\Documents\1DAM\Programacion\ProyectoFinal\ProyectoFinal\PlataformaDeEntrenamiento\PlataformaDeEntrenamiento\bin\...
VPN Wireguard

Mostrar maquinas
Apagar maquina
Mostrar detalles de una maquina
Mostrar VPN wireguard
Mostrar IP Objetivo
Introducir bandera
Mostrar writeups
Subir writeup
Eliminar writeup
Descargar Writeups
Solicitar una nueva maquina
Ranking
Volver

[Interface]
Address = 10.10.10.2
PrivateKey = Cj3bz0N8XfqWn7CGr8S4emkQC2bFUUpXp+lmEItpGVc=
ListenPort = 51820
DNS = 10.10.10.1

[Peer]
PublicKey = Ysxmhdnebfegbgp+oxPdswBrlzDkeBalyhgG0nqxaic=
PresharedKey = IfXk/3s0itEgI9q26Y8dSha1AJPBkEiffJwsKrisSms=
Endpoint = 52.142.216.158:51820
AllowedIPs = 0.0.0.0/0, ::/0

[ Presione enter para continuar ]
```

Y activamos la VPN.

wg-quick up pato-vpn

```
> wg-quick up pato-vpn
[#] ip link add pato-vpn type wireguard
[#] wg setconf pato-vpn /dev/fd/63
[#] ip -4 address add 10.10.10.2 dev pato-vpn
[#] ip link set mtu 1420 up dev pato-vpn
[#] resolvconf -a tun.pato-vpn -m 0 -x
[#] wg set pato-vpn fwmark 51820
[#] ip -6 route add ::/0 dev pato-vpn table 51820
[#] ip -6 rule add not fwmark 51820 table 51820
[#] ip -6 rule add table main suppress_prefixlength 0
[#] nft -f /dev/fd/63
[#] ip -4 route add 0.0.0.0/0 dev pato-vpn table 51820
[#] ip -4 rule add not fwmark 51820 table 51820
[#] ip -4 rule add table main suppress_prefixlength 0
[#] sysctl -q net.ipv4.conf.all.src_valid_mark=1
[#] nft -f /dev/fd/63
```

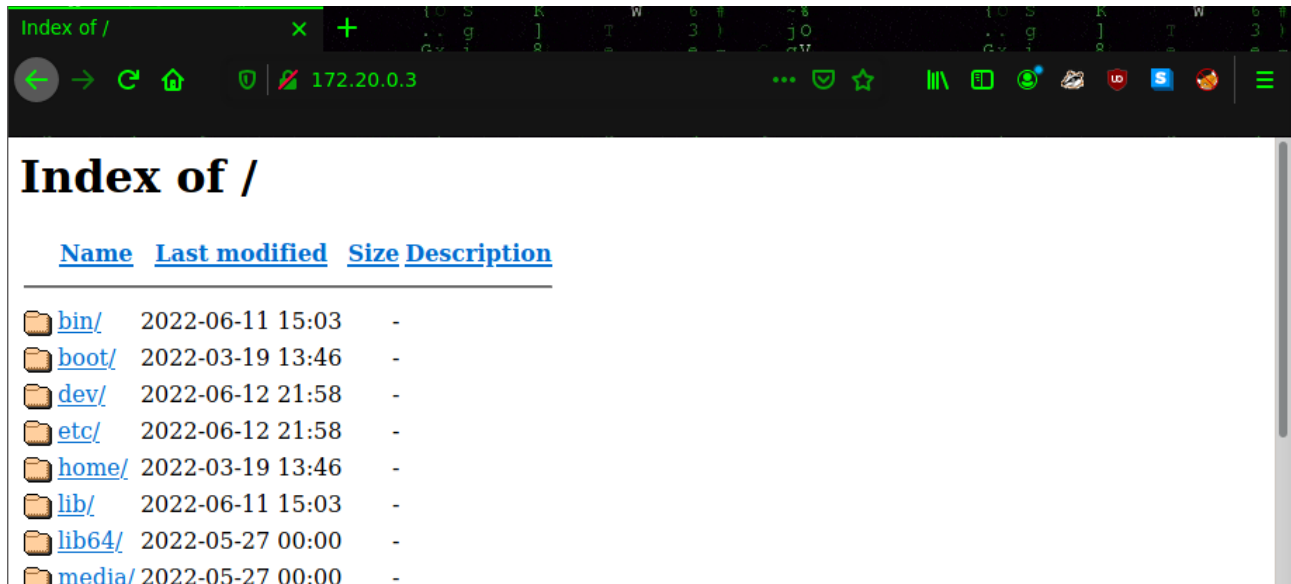
Escaneamos los puertos del contenedor con *nmap*.

```
nmap -p- 172.20.0.3
```

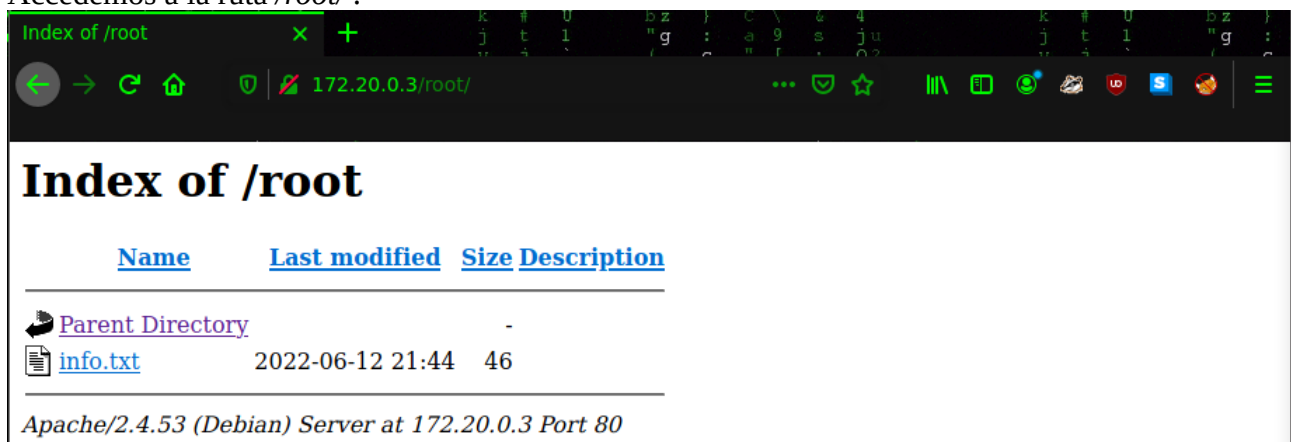
```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-13 00:02 CEST
Nmap scan report for pato-Gift.pato-red (172.20.0.3)
Host is up (0.064s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 40.97 seconds
```

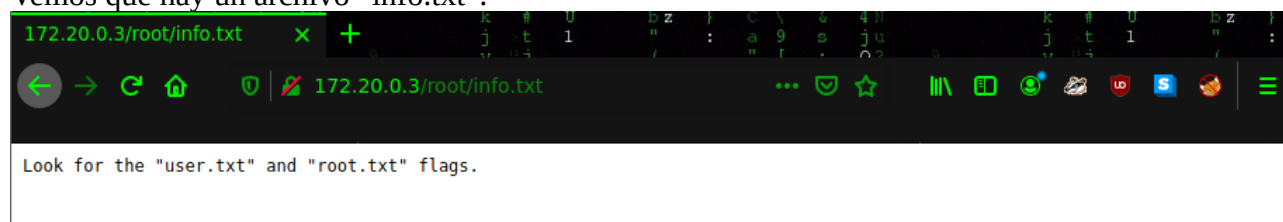
Si abrimos el navegador e introducimos la IP del objetivo podemos ver que se listan los archivos del sistema.



Accedemos a la ruta `/root/`.

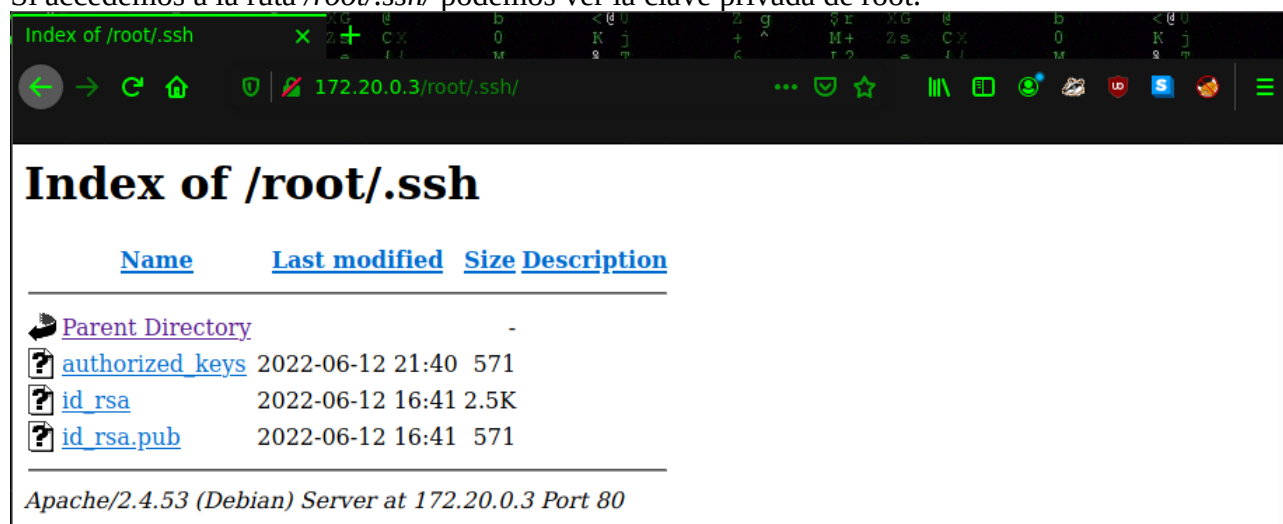


Vemos que hay un archivo "info.txt".



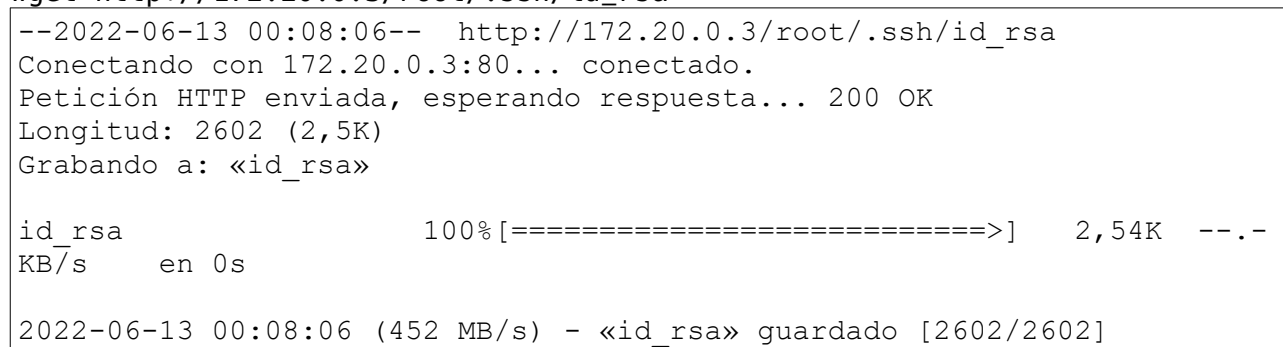
Nos dice que busquemos las banderas de usuario y root.

Si accedemos a la ruta /root/.ssh/ podemos ver la clave privada de root.



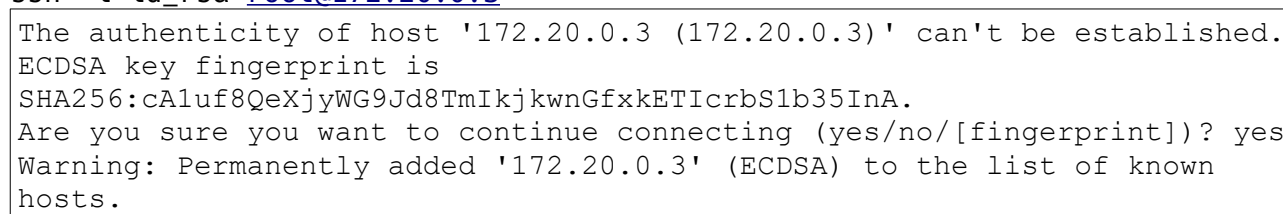
Nos descargamos el archivo, le damos permisos y nos conectamos al contenedor gift.

wget http://172.20.0.3/root/.ssh/id_rsa



chmod 600 id_rsa

ssh -i id_rsa root@172.20.0.3



Laboratorio para pentesting con Docker

```
Linux f8cab4dd7bf5 5.10.0-14-cloud-amd64 #1 SMP Debian 5.10.113-1 (2022-04-29) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
```

```
root@f8cab4dd7bf5:~# id
```

```
uid=0(root) gid=0(root) groups=0(root)
```

Vamos a buscar las banderas del usuario y root.

```
root@f8cab4dd7bf5:~# find / -name "user.txt" 2>/dev/null
```

```
/root/.flags/user.txt
```

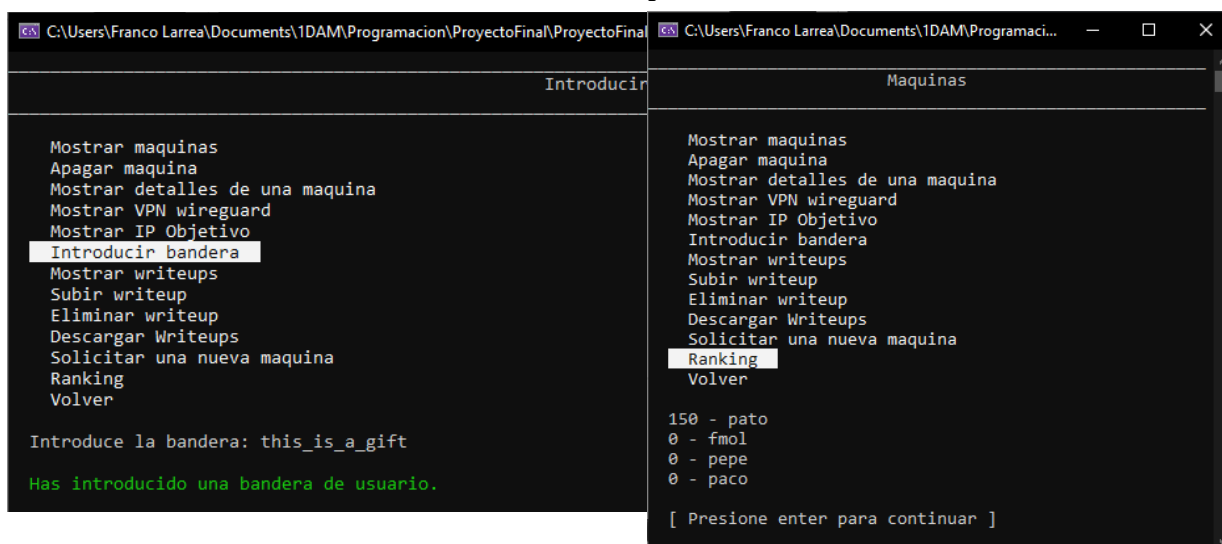
```
root@f8cab4dd7bf5:~# find / -name "root.txt" 2>/dev/null
```

```
/root/.flags/root.txt
```

```
root@f8cab4dd7bf5:~# cat /root/.flags/*
```

```
gg_well_played  
this_is_a_gift
```

Por último vamos a introducir las banderas en la aplicación.



Con esto podemos dar la máquina por finalizada. El usuario pato está primero en el ranking.

9 Publicando las imágenes en [Dockerhub](#)

Vamos a publicar la máquina a [Dockerhub](#), por un lado esto evitara que dependamos de tener la imagen en local y por otro lado compartiremos la imagen con otros usuarios.

Nos creamos una cuenta en [Dockerhub](#) y creamos un repositorio.

Create Repository

This is a vulnerable image for penetration testing.

Visibility

Using 0 of 1 private repositories. [Get more](#)

☒ **Public** Appears in Docker Hub search results

☐ **Private** Only visible to you

Iniciamos sesión en Dockerhub desde el servidor Docker.

```
fmol@hackf-vmachine:~/gift$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: fmol107
Password:
WARNING! Your password will be stored unencrypted in /home/fmol/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Ahora volvemos a crear la imagen pero con el siguiente tag *dockerhub_nick/tag* .

```
docker build . -f gift.dockerfile -t fmol107/gift
```

Y subimos la imagen a nuestro repositorio público.

```
docker push fmol107/gift
```

Ahora cualquier persona puede descargarse la imagen. → [docker pull fmol107/gift]

Y crear contenedores. → [docker run -d fmol107/gift]

<https://hub.docker.com/r/fmol107/gift>

10 Entorno gráfico y conexión por rdp

debian-rdp.dockerfile

debian-rdp.dockerfile

```
FROM debian:stable

ENV DEBIAN_FRONTEND="noninteractive"
RUN apt-get update && apt-get install -y \
    --no-install-recommends \
    xorg \
    dbus-x11 \
    firefox-esr \
    git \
    locales \
    pavucontrol \
    pulseaudio \
    pulseaudio-utils \
    sudo \
    x11-xserver-utils \
    xfce4 \
    xfce4-goodies \
    xfce4-pulseaudio-plugin \
    xorgxrdp \
    xrdp \
    iproute2 \
    nmap \
    wget \
    ssh \
    iputils-ping

RUN groupadd --gid 1020 debian
RUN useradd --shell /bin/bash --uid 1020 --gid 1020 --password $(openssl passwd debian) --
create-home --home-dir /home/debian debian
RUN usermod -aG sudo debian

# Start services
CMD /usr/sbin/xrdp-sesman && /usr/sbin/xrdp --nodaemon
```

Generar la imagen.

```
docker build . -f debian-rdp.dockerfile -t debian-rdp
```

Lanzar contenedor.

```
docker run -d -p 3389:3389 --name debian-rdp --hostname debian-rdp debian-rdp
```

kali-rdp.dockerfile

kali-rdp.dockerfile

```
FROM kalilinux/kali-rolling:amd64

ENV DEBIAN_FRONTEND="noninteractive"
RUN apt update \
    && apt dist-upgrade -y \
    && apt install -y kali-desktop-xfce xrdp \
        nmap fping masscan curl binutils iputils-ping

RUN echo "root:root" | chpasswd

CMD /etc/init.d/xrdp start && tail -f /dev/null
```

Generar la imagen.

```
docker build . -f kali-rdp.dockerfile -t kali-rdp
```

Lanzar contenedor.

```
docker run -d -p 3389:3389 --name kali-rdp --hostname kali-rdp kali-rdp
```

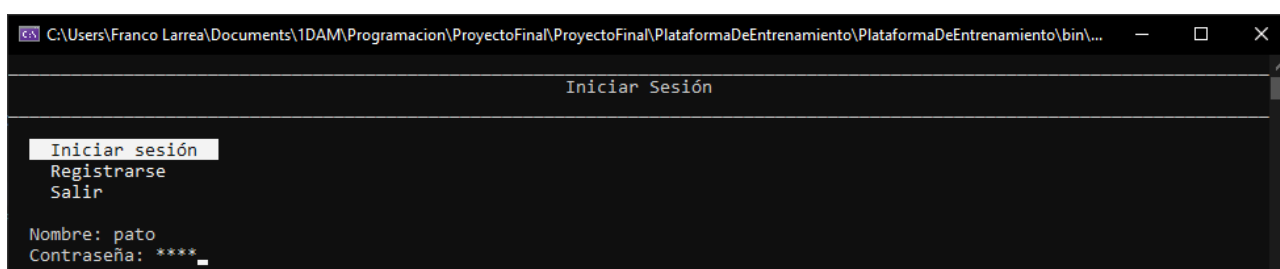
11 Probando las maquinas con entorno gráfico

Hemos añadido a la aplicación la opción de desplegar una maquina con entorno gráfico junto con la maquina objetivo.

La idea es que el usuario pueda utilizar la maquina con entorno gráfico como maquina atacante para vulnerar la maquina objetivo.

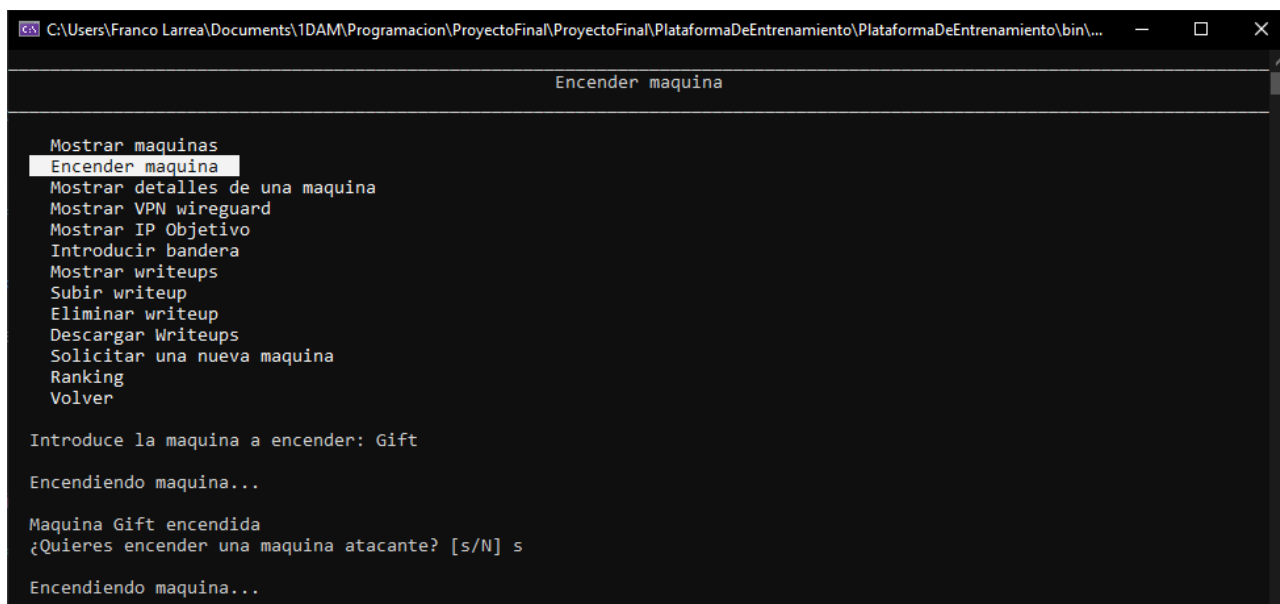
Una vez conectados a la VPN podemos conectarnos por RDP a la maquina atacante.

Iniciamos sesión.



Encendemos una máquina.

Nos preguntará si queremos encender también una máquina atacante.



Laboratorio para pentesting con Docker

Si nos conectamos al servidor Docker por SSH podemos monitorizar las redes y maquinas desplegadas.

```
fmol@hackf-vmachine: ~  
Every 1.0s: docker ps -a && echo "" && docker network ls  
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                NAMES  
a1bd626e65d9   fml107/debian-rdp  "/bin/sh -c '/usr/sb..."  About a minute ago  Up About a minute  pato-debian-rdp  
e6b4ec4638fc   fml107/gift       "/bin/sh -c 'service..."  3 minutes ago    Up 3 minutes      pato-gift  
9ac8f15c87ba   linuxserver/wireguard  "/init"                  3 minutes ago    Up 3 minutes      0.0.0.0:51820->51820/udp  pato-vpn  
  
NETWORK ID     NAME      DRIVER  SCOPE  
6ce557ef575a   bridge   bridge  local  
3f232e89dafd   host     host    local  
8a9929dc92cb   none     null    local  
04a14188f247   pato-red bridge  local
```

Podemos observar que se ha creado una red para *pato* y se han desplegado tres contenedores, la vpn Wireguard, la maquina objetivo y la maquina atacante con servidor rdp.

Nos conectamos a la VPN de pato.

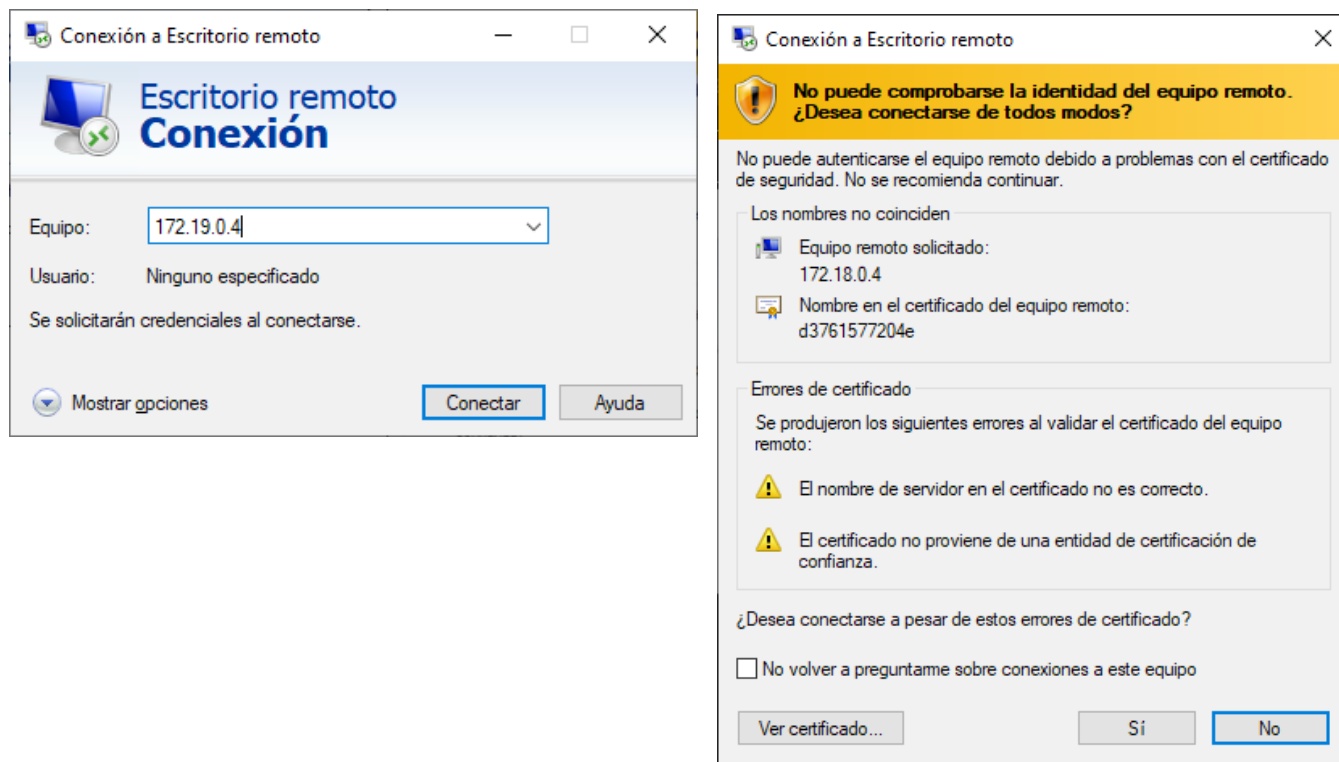
```
C:\Users\Franco Larrea\Documents\1DAM\Programacion\ProyectoFinal\ProyectoFinal\PlataformaDeEntrenamiento\PlataformaDeEntrenamiento\bin\...  
VPN Wireguard  
  
Mostrar maquinas  
Apagar maquina  
Mostrar detalles de una maquina  
Mostrar VPN wireguard  
Mostrar IP Objetivo  
Introducir bandera  
Mostrar writeups  
Subir writeup  
Eliminar writeup  
Descargar Writeups  
Solicitar una nueva maquina  
Ranking  
Volver  
  
[Interface]  
Address = 10.10.10.2  
PrivateKey = Cj3bz0N8XfqWn7CGr8S4emkQC2bFUUpXp+lmEitpGVC=  
ListenPort = 51820  
DNS = 10.10.10.1  
  
[Peer]  
PublicKey = Ysxmhdnebfegbpg+oxPdsWBrLzDkeBalyhgGOnqxaic=  
PresharedKey = IfXk/3s0itEgI9q26Y8dSha1AJPBkEiFfJwsKrisSms=  
Endpoint = 52.142.216.158:51820  
AllowedIPs = 0.0.0.0/0, ::/0  
  
[ Presione enter para continuar ]
```

Mostramos las Ips de los contenedores desplegados.

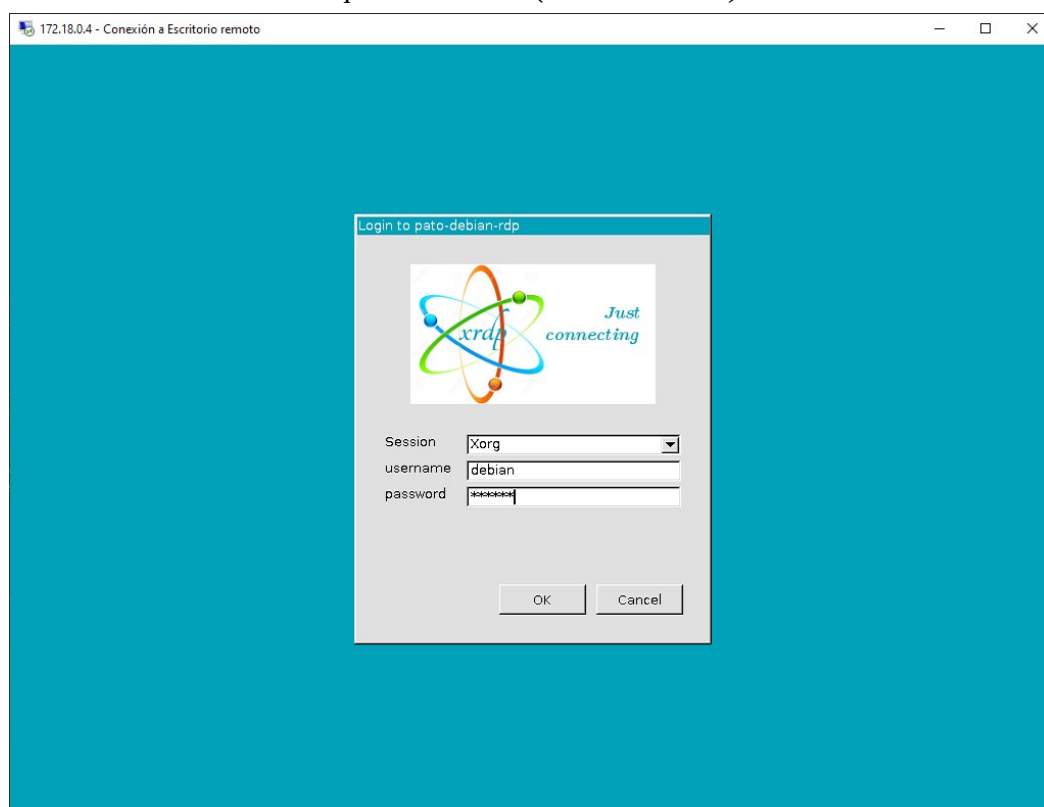
```
C:\Users\Franco Larrea\Documents\1DAM\Programacion\ProyectoFinal\ProyectoFinal\PlataformaDeEntrenamiento\PlataformaDeEntrenamiento\bin\...  
IP Objetivo  
  
Mostrar maquinas  
Apagar maquina  
Mostrar detalles de una maquina  
Mostrar VPN wireguard  
Mostrar IP Objetivo  
Introducir bandera  
Mostrar writeups  
Subir writeup  
Eliminar writeup  
Descargar Writeups  
Solicitar una nueva maquina  
Ranking  
Volver  
  
172.18.0.3  
172.18.0.4  
  
[ Presione enter para continuar ]
```

Laboratorio para pentesting con Docker

Nos conectamos por RDP y aceptamos el certificado.

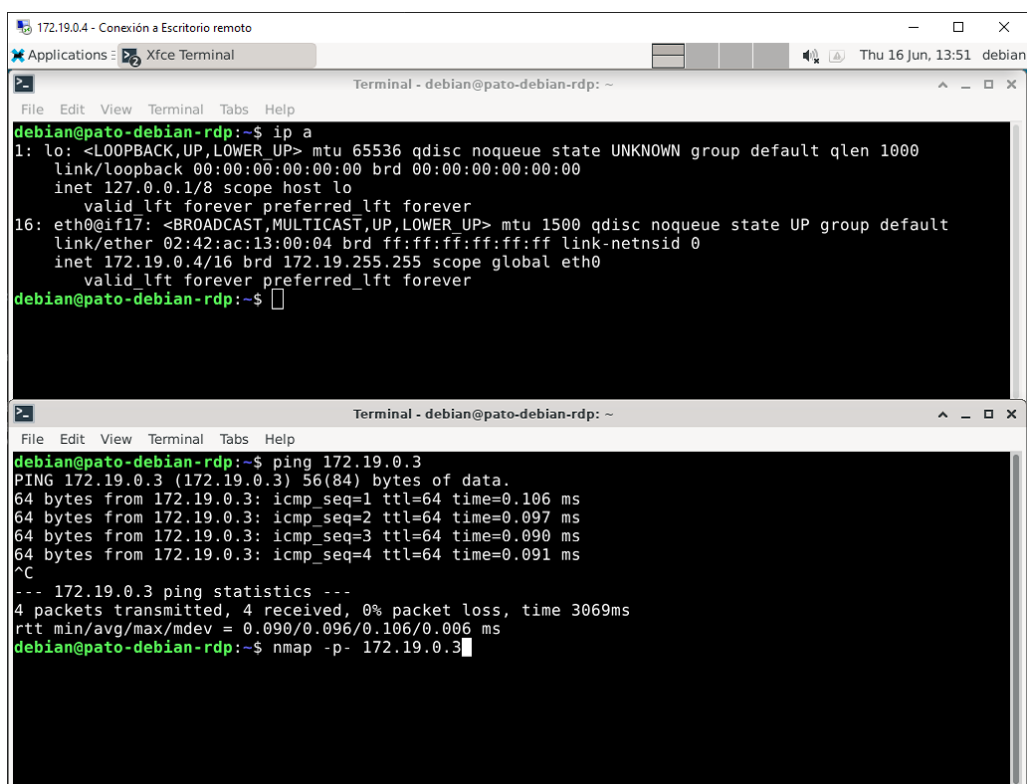
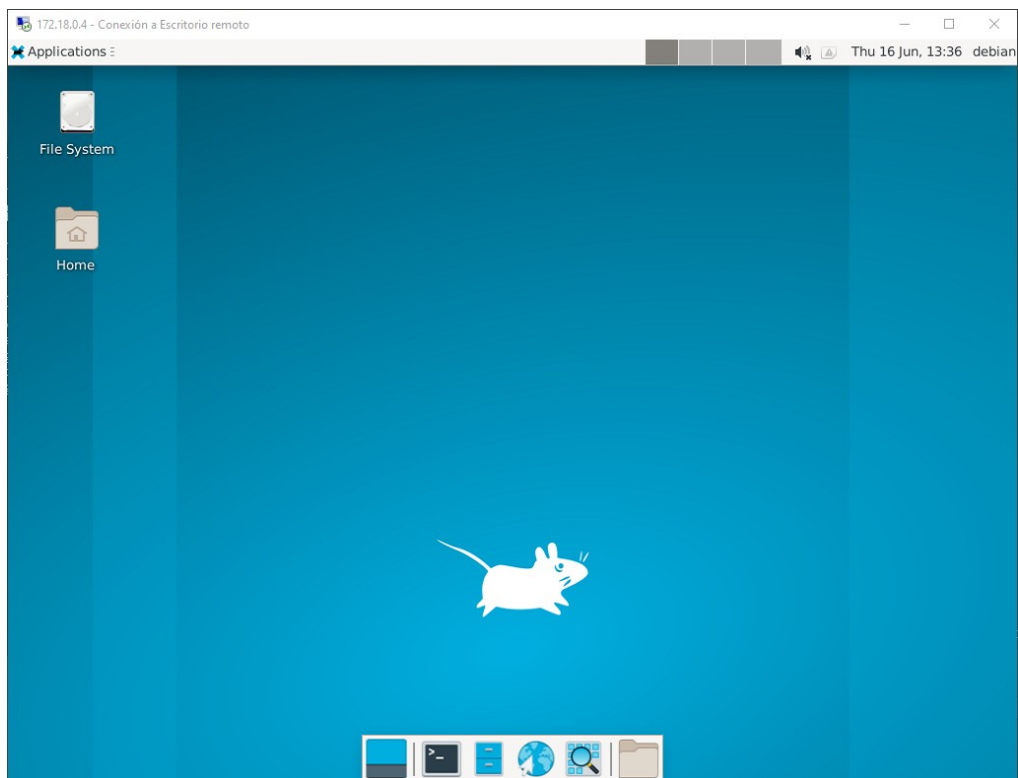


Iniciamos sesión en la maquina atacante. (*debian:debian*)



Laboratorio para pentesting con Docker

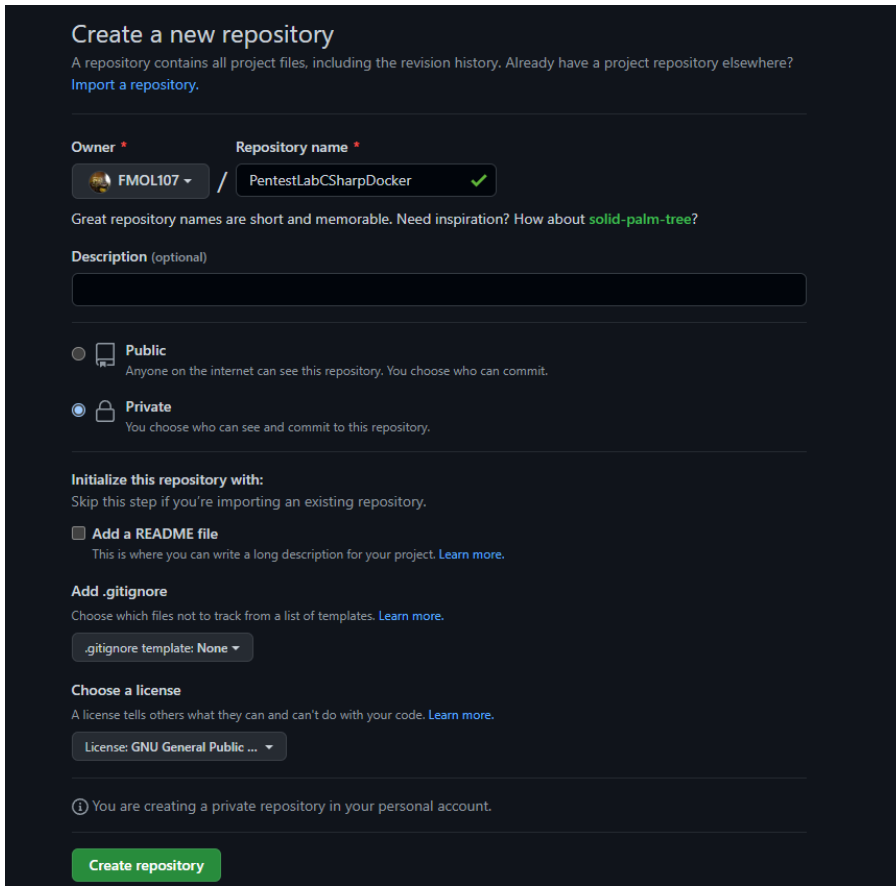
Con esto disponemos de un sistema Unix con entorno gráfico funcional donde podemos instalar lo que necesitemos para vulnerar las maquinas.



12 Publicando el proyecto en GitHub

Queremos que este proyecto sea opensource, para ello vamos a publicar todo el código y la documentación a [GitHub](#) con su respectiva licencia.

Creamos un repositorio en GitHub y especificamos que tenga la licencia GPLv3.



The screenshot shows the GitHub 'Create a new repository' page. The 'Owner' is 'FMOL107' and the 'Repository name' is 'PentestLabCSharpDocker'. The 'Description' field is empty. The 'Visibility' is set to 'Private'. The 'Initialize this repository with' section has 'Add a README file' checked. The 'Add .gitignore' section has '.gitignore template: None' selected. The 'Choose a license' section has 'License: GNU General Public ...' selected. A green 'Create repository' button is at the bottom.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner ^{*} Repository name ^{*}

FMOL107 / PentestLabCSharpDocker ✓

Great repository names are short and memorable. Need inspiration? How about [solid-palm-tree?](#)

Description (optional)

☐ Public
Anyone on the internet can see this repository. You choose who can commit.

☒ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None

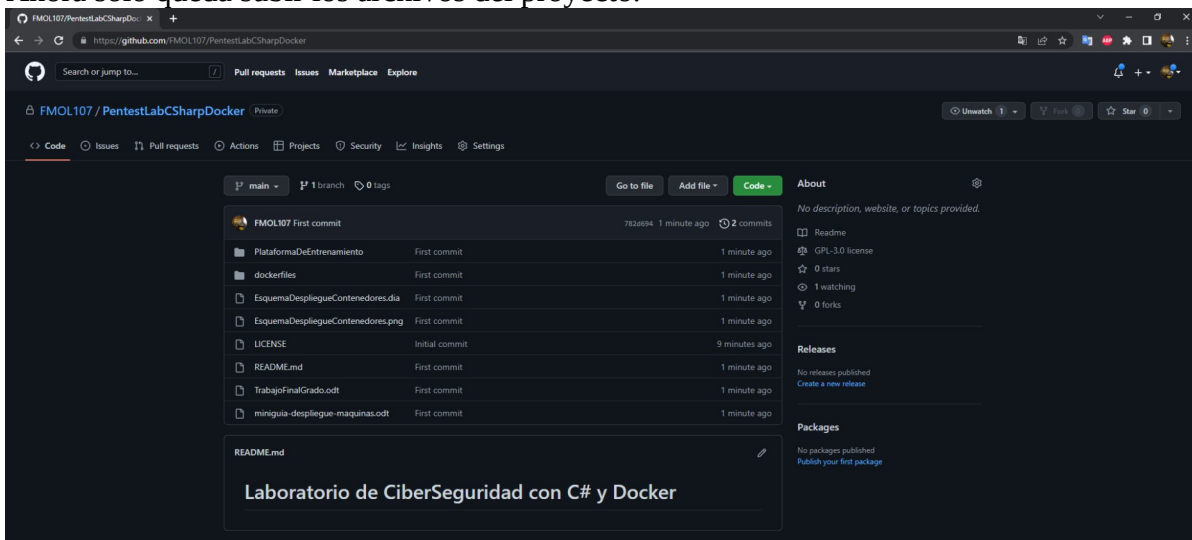
Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: GNU General Public ...

ⓘ You are creating a private repository in your personal account.

Create repository

Ahora solo queda subir los archivos del proyecto.



6 Webgrafia

1 Proyecto

- [PentestLabCSharpDocker](#)
- [fmol107/debian-rdp](#)
- [fmol107/kali-rdp](#)
- [fmol107/gift](#)

2 Recursos

- [linuxserver/wireguard](#)
- [tleemcjr/metasploitable2](#)
- [debian](#)
- [kalilinux/kali-rolling](#)

3 Documentación

- [Crea tu propia plataforma de Hacking con Docker \(D. Cr0hn\) T12 - CyberCamp 2017](#)
- [NETWORKING EN DOCKER!](#)
- [Instala tu PROPIA VPN GRATIS - Wireguard en Docker](#)
- [🔧 Instalación y configuración de WIREGUARD 🐳 en Linux con DOCKER 🇸🇦](#)
- [CÓMO escribir LOS MEJORES Dockerfiles](#)
- [KaliLinux en Docker \(Parte 3/3\) - Interfaz Gráfica](#)
- [How To: Setup Kali and Metasploitable2 on Docker Containers](#)
- [DIFERENCIA entre CMD, RUN, y ENTRYPOINT en DOCKER – V2M](#)
- [C# Interactive Menu for Console Application](#)
- [Introduction to Docker Hacking](#)
- [What is wireguard on Azure, I'll do you better why is wireguard on Azure ?](#)
- [PENTEST LAB: LABORATORIO DE PENTESTING LOCAL UTILIZANDO DOCKER-COMPOSE](#)
- [Building a Pentest lab with Docker](#)
- [Labainers](#)
- [Installing `lightdm` in Dockerfile raises interactive keyboard layout menu](#)
- [Composerize](#)
- [Función para encriptar en sha256 en C# .Net](#)
- [Implementación de aplicaciones de .NET Core con Visual Studio](#)
- [General Usage Instructions and Examples](#)
- [Preparación de un entorno con Docker para Hacking web](#)
- [How To Dockerize Your Pen-testing Lab \[feat. Kali Linux\]](#)
- [How to Get A Docker Container IP Address - Explained with Examples](#)
- <https://hackmyvm.eu/>
- [docker compose](#)
- [docker run](#)
- [docker exec](#)
- [Networking overview](#)
- [REDES EN DOCKER](#)
- [ubuntu/apache2](#)

Laboratorio para pentesting con Docker

- [gtfobins/docker](#)
- [How to setup an ssh server within a docker container](#)
- [Docker Expose Port: What It Means and What It Doesn't Mean](#)
- [Docker Hub Quickstart](#)