Vamos a presentar la práctica de escaneo de Docker con Nmap

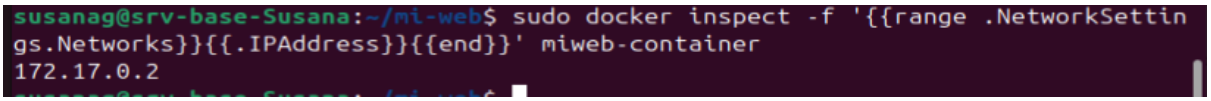
Partimos de la comprobación de que el contenedor está iniciado. Se visualiza la imagen del **localhost** siguiente:

Acto seguido listamos los contenedores que podamos tener para saber el ID del contenedor que nos interesa. Para ello usamos el comando:

sudo docker ps (confirmamos que está ejecutándose)

Posteriormente verificamos la IP interna del contenedor para posteriormente pasar a usar Nmap. Usamos el comando:

sudo docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' miweb-container



En nuestro caso la IP es **172.17.0.2** y vamos a escanear dicha IP con **Nmap**. Primero hacemos el paso previo de instalar Nmap si no lo tenemos instalado con los comandos:

sudo apt update

sudo apt install nmap

Para el escaneo usamos el comando:

sudo nmap 172.17.0.2

```
susanag@srv-base-Susana:~/mi-web$ sudo nmap 172.17.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-16 17:24 CEST
Nmap scan report for 172.17.0.2
Host is up (0.000011s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 02:42:AC:11:00:02 (Unknown)
```

Cuál es el significado de lo que se ve en la imagen:

Host is up: El host (la IP 172.17.0.2) está activo y responde rápido (11 microsegundos).

Not shown: 999 closed tcp ports: De los 1000 puertos que escanea por defecto, 999 están cerrados.

Puerto 80/tcp open http: El puerto 80 está abierto y corriendo un servicio HTTP (web server).

MAC Address: Es la dirección física del adaptador virtual de red del contenedor (Docker usa direcciones MAC asignadas virtualmente).

El contenedor Docker en esa IP tiene activo un servidor web (NGINX) escuchando en el puerto 80, que es justamente lo que se esperaba para servir el sitio.

El escaneado para detección de servicios lo ejecutamos (para el contenedor) con el comando:

sudo nmap -sV 172.17.0.2

```
susanag@srv-base-Susana:~/mi-web$ sudo nmap -sV 172.17.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-16 17:32 CEST
Nmap scan report for 172.17.0.2
Host is up (0.000012s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    nginx 1.29.0
MAC Address: 02:42:AC:11:00:02 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.64 seconds
```

Arroja este resultado:

PORT   STATE SERVICE VERSION

80/tcp open http   nginx 1.29.0

El puerto 80 está **abierto**.

El servicio es **nginx versión 1.29.0**.

Esto confirma que el servidor web en el contenedor está **activo** y **sirviendo contenido**.

Para el host con el puerto **8080** (en el host **192.168.1.69**, que es la interfaz de red física o virtual de la máquina host)

PORT STATE SERVICE VERSION

8080/tcp filtered http-proxy

El puerto 8080 aparece como **filtered** (filtrado).

Esto significa que algún firewall o regla de red está bloqueando o filtrando las respuestas en ese puerto.

Por eso nmap no pudo determinar exactamente qué servicio hay, aunque sepamos que se ejecuta nuestro nginx vía Docker.

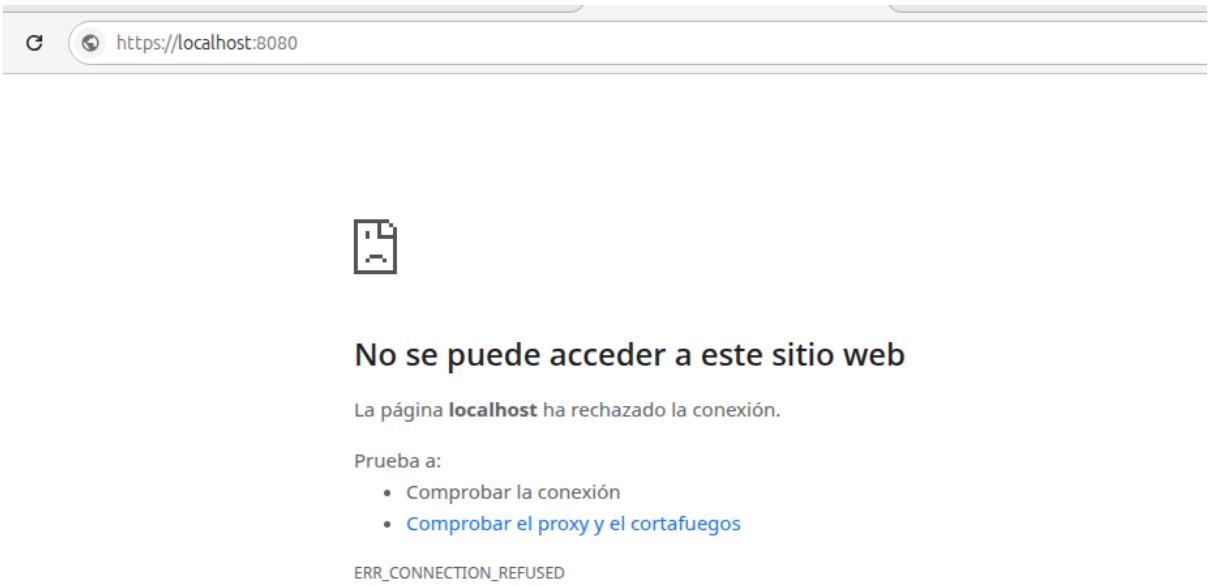
sudo nmap -sV -p 8080 192.168.1.69

```
susanag@srv-base-Susana:~/mi-web$ sudo nmap -sV -p 8080 192.168.1.69
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-16 17:33 CEST
Nmap scan report for mail.codearts.local (192.168.1.69)
Host is up.

PORT     STATE    SERVICE    VERSION
8080/tcp filtered http-proxy

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.32 seconds
```

Verificamos que no podemos acceder al localhost desde otra máquina (usamos el navegador de la máquina real), algo que sería una brecha de seguridad del contenedor



A continuación detenemos el contenedor con el comando:

sudo docker stop miweb-container

```
susanag@srv-base-Susana:~/mi-web$ sudo docker stop miweb-container
miweb-container
```

Y procedemos al escaneo, donde se observa la diferencia de información al tener el contenedor ahora detenido.

Escaneamos con:

sudo nmap -sV 172.17.0.2 y obtenemos:

```
susanag@srv-base-Susana:~/mi-web$ sudo nmap -sV 172.17.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-16 17:40 CEST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 1.68 seconds
```

o con el comando:

sudo nmap -sV -p 8080 192.168.1.69

donde aparece el puerto 8080 con **state CLOSED**

```
susanag@srv-base-Susana:~/mi-web$ sudo nmap -sV -p 8080 192.168.1.69
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-16 17:42 CEST
Nmap scan report for mail.codearts.local (192.168.1.69)
Host is up (0.000082s latency).

PORT     STATE  SERVICE    VERSION
8080/tcp closed http-proxy

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
```

La reflexión final sobre la visibilidad de servicios en una red y su importancia en seguridad sería la siguiente:

La **visibilidad de servicios** en una red es fundamental para comprender qué sistemas están activos, qué aplicaciones se están ejecutando y cómo están configurados los accesos.

Las herramientas como **nmap** permiten detectar puertos abiertos, servicios y versiones, lo que es clave tanto para la administración como para la seguridad.

- **Desde la perspectiva de seguridad**, una buena visibilidad ayuda a:
    - Identificar servicios no autorizados o vulnerables.
    - Detectar posibles puertas traseras o servicios expuestos que no deberían estar accesibles.
    - Auditar configuraciones y confirmar que solo los servicios necesarios están disponibles.
    - Preparar defensas proactivas: saber qué atacar potencialmente un atacante.
- **Sin visibilidad adecuada**, las organizaciones están a ciegas, lo que aumenta el riesgo de brechas, ataques y explotación de servicios desconocidos o mal configurados.
- **Sin embargo, la visibilidad también es un arma de doble filo**, ya que un atacante con acceso a la red también puede usar estas técnicas para mapear y atacar.

Por eso, es vital equilibrar la **transparencia interna** para administración y monitoreo, con **controles estrictos y segmentación de red** para minimizar la exposición externa.

En resumen: **Conocer qué servicios existen y cómo están expuestos es la base para proteger cualquier infraestructura. La visibilidad es poder, y ese poder debe usarse para reforzar, no para debilitar la seguridad.**