



OZonE
CIBERSECURITY

Red Team - Enumeración Script para descubrir hosts y escaneo con nmap

all port

```
nmap -p- -sV 192.168.1.1
```

Ruben Apablaza Muñoz

-OZonE-

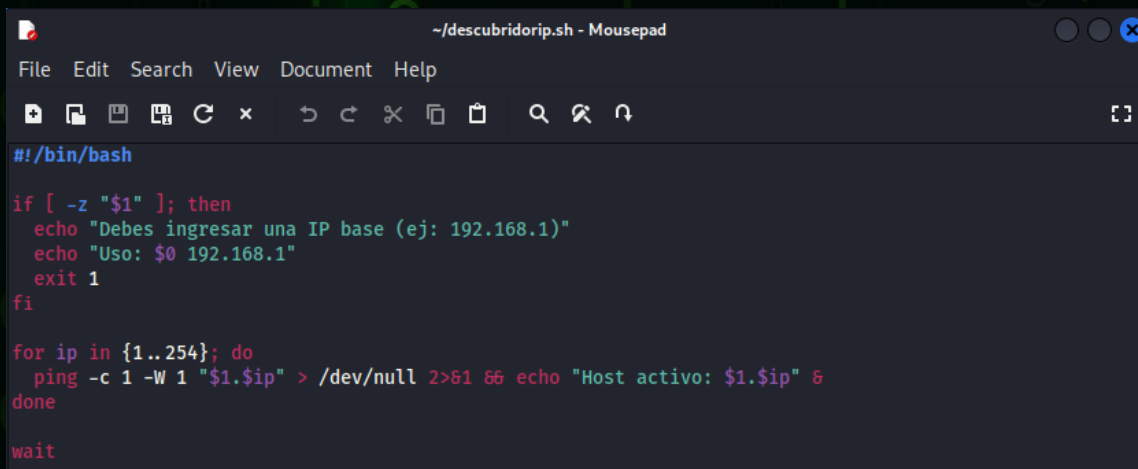
SCRIPT PARA DESCUBRIR IP's Y ESCANEO DE PUERTOS

1. Lo primero será realizar un pequeño script que escanea los hosts activos en una red, se puede hacer en nano, vim, gedit. En este caso utilizare mousepad y se le puede colocar el nombre que cada uno prefiera, en mi caso, "descubridorip.sh"

```
(kali㉿kali)-[~]  
$ mousepad descubridorip.sh
```

2. En su interior crearemos lo siguiente:

```
#!/bin/bash  
  
if [ -z "$1" ]; then  
    echo "Debes ingresar una IP base (ej: 192.168.1)"  
    echo "Uso: $0 192.168.1"  
    exit 1  
fi  
  
for ip in {1..254}; do  
    ping -c 1 -W 1 "$1.$ip" > /dev/null 2>&1 && echo "Host activo:  
$1.$ip" &  
done  
  
wait
```



```
~/descubridorip.sh - Mousepad  
File Edit Search View Document Help  
[Icons]  
#!/bin/bash  
  
if [ -z "$1" ]; then  
    echo "Debes ingresar una IP base (ej: 192.168.1)"  
    echo "Uso: $0 192.168.1"  
    exit 1  
fi  
  
for ip in {1..254}; do  
    ping -c 1 -W 1 "$1.$ip" > /dev/null 2>&1 && echo "Host activo: $1.$ip" &  
done  
  
wait
```



```
#!/bin/bash
```

Define que el script se ejecutará con Bash.

```
if [ -z "$1" ]; then
```

Verifica si no se ingresó un argumento (IP base vacía).

```
    echo "Debes ingresar una IP base (ej: 192.168.1)"
```

```
    echo "Uso: $0 192.168.1"
```

```
    exit 1
```

Muestra mensajes de ayuda y termina el script si no se ingresó IP.

```
for ip in {1..254}; do
```

Recorre los valores del 1 al 254 para escanear cada host de la red.

```
    ping -c 1 -W 1 "$1.$ip" > /dev/null 2>&1 && echo "Host  
activo: $1.$ip" &
```

→ Hace ping 1 vez con un timeout de 1 segundo al host.

→ Si responde (&&), muestra "Host activo: IP".

→ > /dev/null 2>&1 oculta la salida del ping.

→ & lo ejecuta en segundo plano (rápido).

```
done
```

Termina el bucle for.

```
wait
```

Espera a que todos los pings en segundo plano terminen antes de cerrar el script.

3. Ejecutamos nuestro script con `./descubridorip.sh` y como se puede ver envía mensaje de error debido a que no se ingresó la red a revisar.

```
(kali㉿kali)-[~]  
$ ./descubridorip.sh  
Debes ingresar una IP base (ej: 192.168.1)  
Uso: ./descubridorip.sh 192.168.1
```

Por lo tanto, ahora verificamos con `./descubridor.sh 192.168.0` y se puede ver como si funciona

```
(kali㉿kali)-[~]  
$ ./descubridorip.sh 192.168.0  
Host activo: 192.168.0.1  
Host activo: 192.168.0.41  
Host activo: 192.168.0.30  
Host activo: 192.168.0.32  
Host activo: 192.168.0.252  
Host activo: 192.168.0.31
```

4. Ahora lo que haremos será guardar la salida de este script en un archivo de texto con

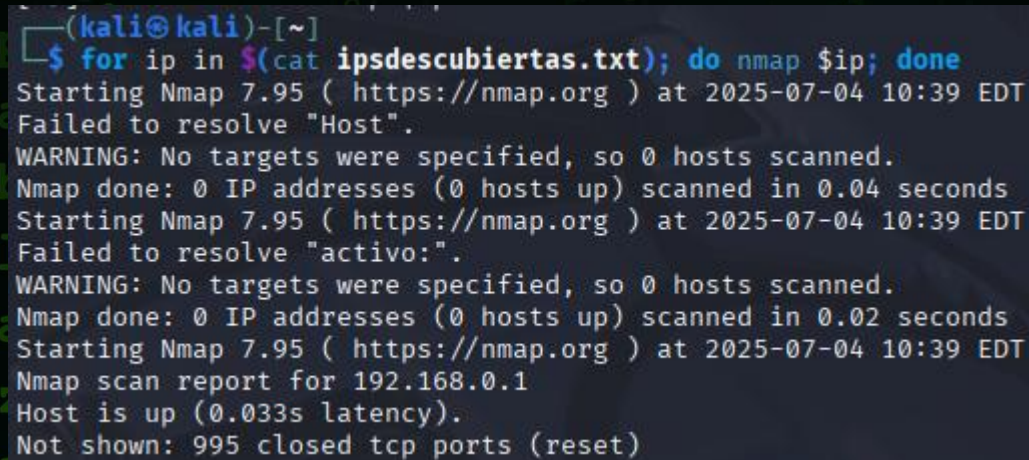
`./descubridorip.sh 192.168.0 > ipsdescubiertas.txt`

Y con `cat` verificamos el archivo que ya contiene las ip.

```
(kali㉿kali)-[~]  
$ ./descubridorip.sh 192.168.0 > ipsdescubiertas.txt  
  
(kali㉿kali)-[~]  
$ cat ipsdescubiertas.txt  
Host activo: 192.168.0.1  
Host activo: 192.168.0.32  
Host activo: 192.168.0.41  
Host activo: 192.168.0.30  
Host activo: 192.168.0.252  
Host activo: 192.168.0.31
```

5. Con esta línea lo que haremos será llamar un bucle for para que por cada ip contenida en el archivo `ipdescubiertas.txt` realice un nmap de una en una con “;” en vez de todas en paralelo con “&” después de la “ip” y antes del “done” en la parte final de la línea de comando

```
for ip in $(cat ipdescubiertas.txt); do nmap $ip; done
```



A terminal window with a dark background and light-colored text. The prompt is `(kali@kali)-[~]`. The user enters the command `$ for ip in $(cat ipdescubiertas.txt); do nmap $ip; done`. The terminal output shows three iterations of the loop. The first two iterations fail to resolve the host names "Host" and "activo:", respectively, and show a warning that no targets were specified. The third iteration successfully scans the IP address 192.168.0.1, showing that the host is up and that 995 closed TCP ports were reset.

```
(kali@kali)-[~]  
$ for ip in $(cat ipdescubiertas.txt); do nmap $ip; done  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-04 10:39 EDT  
Failed to resolve "Host".  
WARNING: No targets were specified, so 0 hosts scanned.  
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.04 seconds  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-04 10:39 EDT  
Failed to resolve "activo:".  
WARNING: No targets were specified, so 0 hosts scanned.  
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.02 seconds  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-04 10:39 EDT  
Nmap scan report for 192.168.0.1  
Host is up (0.033s latency).  
Not shown: 995 closed tcp ports (reset)
```

- Escanea una ip a la vez de forma secuencial (no en paralelo &)
- Utiliza nmap por cada línea(ip) dentro del archivo `ipdescubiertas.txt`

CONCLUSIONES.

El objetivo del script mostrado es descubrir IPs activas en una red y escanear sus puertos de forma organizada. Es un ejercicio que busca mostrar la creación y ejecución de scripts en Kali Linux y ayuda a entender la lógica detrás de estas herramientas.

Se trata de un script básico que incluso se puede mejorar, como por ejemplo incluir la creación del archivo con las IPs descubiertas desde el mismo script en vez de hacerlo después como se muestra en el paso 4.

Se debe dejar en claro que existen herramientas que realizan esta misma función como nmap o para redes mas grandes incluso se puede utilizar masscan + nmap. Pero el enfoque siempre debe ser, entender lo básico primero. Uno de los pocos escenarios posibles en que se podría utilizar este script por sobre otras herramientas, seria en el caso de estar en una red local aislada sin acceso a internet y no se tenga acceso a Kali Linux; en este caso el script como se presenta hasta el paso 3 solo serviría para descubrir hosts, para que pueda escanear puertos habría que modificarlo, pero este es un punto que escapa a este documento.

A continuación se deja un cuadro explicativo respecto de estas herramientas.

Método	Velocidad	Detalle	Escalable	Recomendado para
<code>nmap -sn + nmap -p</code>	Media	Alto	Media	Redes medianas o pequeñas
<code>nmap -sn + xargs + -P</code>	Alta	Alto	Alta	Uso profesional por pasos
<code>masscan + nmap</code>	Muy alta	Alto	Muy alta	Redes grandes (más de /24)
Script bash manual + ping	Lento	Bajo	No	Enseñanza básica

Algunas opciones de nmap

```
nmap -sn 192.168.0.0/24
```

Descubre hosts activos (sin puertos)

```
nmap -p 22,80,443 192.168.0.0/24
```

- Escanea los puertos 22 (SSH), 80 (HTTP), 443 (HTTPS) de cada IP.
- No necesitas -sn, ya que nmap detecta automáticamente si el host está activo antes de escanear.

```
nmap -p- 192.168.0.0/24
```

MUCHO MAS LENTO!!! Pero escanea todos los puertos

```
nmap -p- 192.168.0.0/24 --stats-every 10s
```

mostrará el progreso cada 10 segundos. Se puede configurar para cambiar esos segundos, ideal para escaneos largos.

```
nmap -F 192.168.0.0/24
```

- Escaneo rápido de todos los puertos comunes.

