



**Trabajo práctico**  
**Laboratorio de redes**  
**y**  
**sistemas operativos**

Tema: Masscan

Integrantes:

Pereira, Matias

Vázquez, D. Julián

# Índice

<b>Objetivo del trabajo práctico</b>	<b>3</b>
<b>Masscan: ¿Qué es?</b>	<b>4</b>
<b>Masscan: Compilación</b>	<b>4</b>
<b>Masscan: Instalación</b>	<b>5</b>
<b>Requisitos tecnicos</b>	<b>5</b>
<b>Proceso de instalación</b>	<b>6</b>
<b>Masscan: Uso</b>	<b>6</b>
<b>Masscan: Contribución</b>	<b>8</b>
<b>Laboratorio</b>	<b>8</b>
Objetivo	8
Requisitos	8
Funcionamiento	8
Ejecución	8

## Objetivo del trabajo práctico

El presente trabajo práctico tiene como objetivo transmitir cuales son las principales características de la herramienta Masscan, proceso y requisitos de instalación y ejecución de la misma. Así también se busca identificar cómo puede ser ejecutado Masscan. Finalmente se presenta un repositorio mediante el cual puede ser ejecutado un laboratorio básico para entender su funcionamiento

# Masscan: ¿Qué es?

Es una herramienta de código abierto multiplataforma que mediante su CLI (Command Line interface) permite realizar escaneo de puertos en redes o equipos dentro de una. Está construido usando el lenguaje de programación C.

Mediante los parámetros que recibe su CLI es posible realizar escaneos de red modificando cabeceras en diferentes capas del modelo OSI.

Una de sus principales características es que mediante el parámetro “---rate” se puede especificar la cantidad de paquetes por segundo que serán enviados, pudiendo ser este valor desde 100 (por defecto) hasta 25 millones. También permite continuar un escaneo si fue abortado. Distribuir el trabajo entre otros equipos entre muchas otras de sus características

## Masscan: Compilación

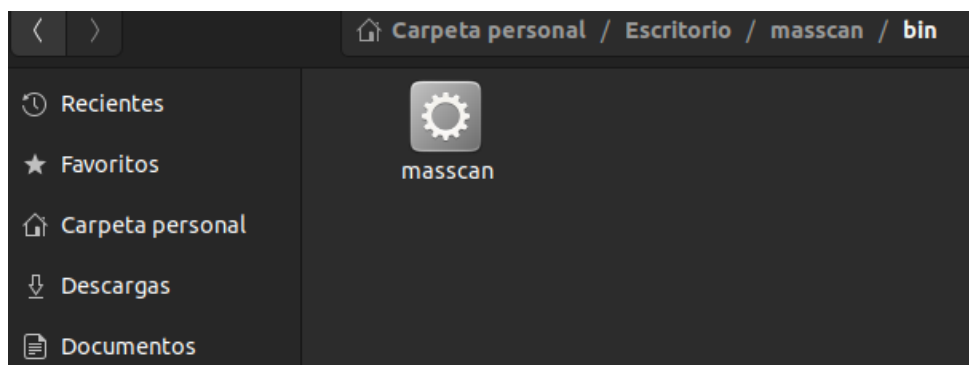
Una posible opción para ejecutar la herramienta es compilar su código fuente, el cual está ubicado en el repositorio <https://github.com/robertdavidgraham/masscan>.

Debido a que el código fuente consta de una gran cantidad de pequeños archivos, para simplificar el proceso de compilación el proyecto ofrece un Makefile que puede ser utilizado por la herramienta **Make**.

A continuación, se enumeran los pasos para compilar Masscan, en este caso, en un sistema operativo Ubuntu.

- Instalar dependencias  
`sudo apt install git make gcc -y`
- Clonar el repositorio  
`git clone https://github.com/robertdavidgraham/masscan`  
`cd masscan`
- Ejecutar compilación  
`make`

Una vez realizados estos pasos se generará un archivo ejecutable en la carpeta bin llamado masscan



Si se desea validar la compilación

```
apt install -y libpcap-dev  
./bin/masscan -help
```

Si la compilación fue exitosa se deberá observar una salida por consola similar a:

```
usage:  
masscan -p80,8000-8100 10.0.0.0/8 --rate=10000  
  scan some web ports on 10.x.x.x at 10kpps  
masscan --nmap  
  list those options that are compatible with nmap  
masscan -p80 10.0.0.0/8 --banners -oB <filename>  
  save results of scan in binary format to <filename>  
masscan --open --banners --readscan <filename> -oX <savefile>  
  read binary scan results in <filename> and save them as xml in <savefile>
```

## Masscan: Instalación

A continuación se detalla información acerca de la instalación

### Requisitos tecnicos

Sistemas operativo:

- Windows 7 en adelante
- Linux
- MacOS

Librerías:

Debido a como está diseñado su proceso de compilación la herramienta requiere que en el sistema operativo esté instalada la librería [libpcap](#).

Hardware:

- CPU de un núcleo
- 256 MB de memoria RAM
- Tarjeta grafica no requerida

## Proceso de instalación

El primer paso es copiar el compilado que se hizo en pasos anteriores al directorio /opt, destinado a la instalación de paquetes de software de aplicaciones complementarias que no forman parte de la instalación predeterminada o son de terceros.

```
sudo mkdir /opt/masscan  
sudo cp $PWD/bin/masscan /opt/masscan
```

Por último, ejecutamos el archivo compilado. Podemos en este punto decidir si queremos compartirlo con todos los usuarios o con el actual, para ello, utilizamos una de estas dos alternativas:

- A. Si se quiere compartir con todos los user  

```
sudo ln -s /opt/masscan/masscan /usr/bin/share/masscan
```
- B. Si solo será utilizado por el usuario actual  

```
sudo ln -s /opt/masscan/masscan /usr/local/bin/masscan
```

## Masscan: Uso

Partiendo de un ejemplo básico, podemos pedirle a masscan que detecte el protocolo/servicio que se encuentra en un puerto determinado dentro de nuestra red con el siguiente comando:

```
sudo masscan 192.168.100.0/24 -p80
```

```
juli@juli-VirtualBox:~/Escritorio$ sudo masscan 192.168.100.0/24 -p80  
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2022-11-27 18:33:19 GMT  
Initiating SYN Stealth Scan  
Scanning 256 hosts [1 port/host]  
Discovered open port 80/tcp on 192.168.100.1
```

El resultado arroja el host, el puerto y el protocolo que se encuentran abiertos. En este caso TCP en el puerto 80.

Si lo que se quiere es escanear un rango de puertos consecutivos, se puede utilizar el siguiente comando:

```
sudo masscan 192.168.100.0/24 -p0-79
```

```
juli@juli-VirtualBox:~/Escritorio$ sudo masscan 192.168.100.0/24 -p0-79  
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2022-11-27 18:29:06 GMT  
Initiating SYN Stealth Scan  
Scanning 256 hosts [80 ports/host]  
Discovered open port 53/tcp on 192.168.100.1
```

Si se quiere escanear un conjunto determinado de puertos, se los puede pasar como parámetro separados por “,”

```
sudo masscan 192.168.100.0/24 -p22,25,80,443,53
```

```
juli@juli-VirtualBox:~/Escritorio$ sudo masscan 192.168.100.0/24 -p22,25,80,443,53
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2022-11-27 18:36:00 GMT
Initiating SYN Stealth Scan
Scanning 256 hosts [5 ports/host]
Discovered open port 80/tcp on 192.168.100.1
Discovered open port 53/tcp on 192.168.100.1
```

Una funcionalidad importante de masscan es el ritmo o velocidad de transmisión. Por defecto este valor es de 100 paquetes por segundo. Es factible incrementar o disminuir dicha tasa de transferencia de la siguiente manera.

```
sudo masscan -p22,25,80,443 192.168.100.0/24 --banners --rate 1000
```

```
juli@juli-VirtualBox:~/Escritorio$ sudo masscan 192.168.100.0/24 -p0-80 --banners --rate 1000
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2022-11-27 18:43:55 GMT
Initiating SYN Stealth Scan
Scanning 256 hosts [81 ports/host]
Discovered open port 80/tcp on 192.168.100.1
rate: 1.00-kpps, 26.09% done, 0:00:17 remaining, found=1
```

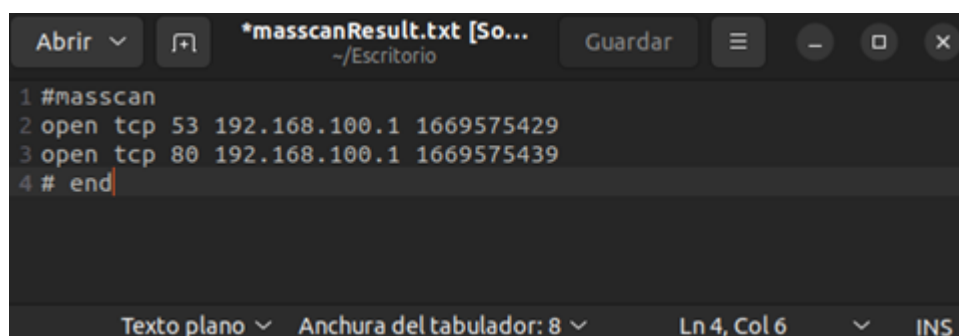
Mientras que en un escaneo normal vemos que la tasa promedio de transferencia es 0.10kpps.

```
juli@juli-VirtualBox:~/Escritorio$ sudo masscan 192.168.100.0/24 -p0-80
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2022-11-27 18:42:06 GMT
Initiating SYN Stealth Scan
Scanning 256 hosts [81 ports/host]
rate: 0.10-kpps, 1.12% done, 0:08:52 remaining, found=0
```

También es factible guardar los resultados del escaneo en cinco diferentes formatos; xml, binario (formato incorporado de masscan), greapeable, json y list (listado de un host y puerto por línea).

```
sudo masscan -p1-443 192.168.100.0/24 --banners --rate 10000 -oL ./masscanResult.txt
```

```
juli@juli-VirtualBox:~/Escritorio$ sudo masscan -p1-443 192.168.100.0/24 --banners --rate 10000 -oL ./masscanResult.txt
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2022-11-27 19:00:52 GMT
Initiating SYN Stealth Scan
Scanning 256 hosts [443 ports/host]
```



The screenshot shows a terminal window titled '\*masscanResult.txt [So...]' with a file icon and a 'Guardar' button. The terminal content is as follows:

```
1 #masscan
2 open tcp 53 192.168.100.1 1669575429
3 open tcp 80 192.168.100.1 1669575439
4 # end
```

At the bottom of the window, there is a status bar with the following information: 'Texto plano', 'Anchura del tabulador: 8', 'Ln 4, Col 6', and 'INS'.

# Masscan: Contribución

Dada su naturaleza open source Masscan permite contribuciones en su repositorio de código <https://github.com/robertdavidgraham/masscan>. Mediante este repositorio de git alojado en Github también es posible reportar issues

## Laboratorio

Se deja disponible en el repositorio <https://github.com/immatp/unq-labo-so> un laboratorio de ejemplo.

## Objetivo

El presente repositorio contiene tres laboratorios de ejemplos listos para ser ejecutados, mediante lo cuales se puede probar el comportamiento de la herramienta Masscan

## Requisitos

Para ejecutar este laboratorio es necesario tener Docker y Docker compose instalado

- Instalacion Docker: <https://docs.docker.com/engine/install/>
- Instalacion Docker Compose: <https://docs.docker.com/compose/install/>

## Funcionamiento

Aunque sería posible ejecutar Masscan en un equipo por si solo, los resultados de la búsqueda realizada por la herramienta puedan variar entre diferentes redes y equipos. Para evitar diferencias se organizó el laboratorio para ser ejecutado mediante Docker, Docker Compose y Bash

## Ejecución

En la carpeta scripts/runners se encuentran los scripts de bash:

- run\_basic\_nginx.sh
- run\_small\_network.sh
- run\_big\_network.sh



Cada uno de los scripts plantea un escenario diferente para realizar un escaneo con Masscan.

Script	Servicios que ejecuta	Puertos expuestos en cada contenedor
run_basic_nginx.sh	1 Nginx	80 (Nginx)
run_small_network.sh	1 Nginx - 1 FTP - 1 Telnet	80 (Nginx) - 21 (FTP) - 23 (Telnet)
run_big_network.sh	1 Nginx - 3 FTP - 3 Telnet	80 (Nginx) - 21 (FTP) - 23 (Telnet)

Para iniciar cualquiera de los escenarios se debe posicionar en la raíz del repositorio y ejecutar

**sh scripts/runners/NOMBRE\_DEL\_SCRIPT\_A\_EJECUTAR**

Como STDOUT del servicio "masscan" de cada laboratorio se obtendrá el resultado del escaneo realizado por Masscan dentro de la red Docker asignada al laboratorio