SSI

Práctica 7: Integridad Y Control De Acceso

Realizado por:

- Marcos RIvas Kyoguro (70962760D)
- Pablo Moreno Barrios (70908442V)

MORENO BARRIOS PABLO - 70908442V Firmado digitalmente por MORENO BARRIOS PABLO - 70908442V Fecha: 2024.12.08 19:54:40 +01'00'

MARCOS|RIVAS|KYOGURO Firmado digitalmente por MARCOS|RIVAS|KYOGURO Fecha: 2024.12.08 19:56:10 +01'00'

Objetivos

- **Configurar un entorno de red seguro**
- **Implementar servicios seguros en un servidor**
- **Configurar Tcp Wrappers**
- **Seguridad del sistema en redes**
- **Herramientas de análisis de puertos**

El Entorno de trabajo en el que se llevará a cabo esta práctica está compuesto por un equipo windows con WSL desde el que haremos pruebas, una máquina virtual con ubuntu en la que se encontrará el servidor y por último una máquina virtual con debian en la que haremos algunas pruebas también.

Configuraciones que ayudan a la seguridad

En primer lugar instalaremos las herramientas que utilizaremos a lo largo de la práctica:

- **Openssh-server para el ssh**

```
pablo@abduscan:~/Escritorio/Práctica_Seguridad$ sudo apt install openssh-server
[sudo] contraseña para pablo:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

- **Vsftpd para el ftp**

```
pablo@abduscan:~/Escritorio/Práctica_Seguridad$ sudo apt install vsftpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

- **Apache2 para el servidor web**

```
pablo@abduscan:~/Escritorio/Práctica_Seguridad$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Una vez instaladas todas las herramientas que vamos a utilizar, empezaremos con los TCPWrappers para filtrar las conexiones a estos servicios.

Para esto modificaremos el fichero /etc/vsftpd.conf

```
pablo@abduscan:/etc$ nano vsftpd.conf
```

Donde añadiremos la línea “tcp_wrappers=yes” que nos permitirá hacer uso de ellos

```
tcp_wrappers=YES
```

Reiniciamos el servicio vsftpd para que los cambios sean efectivos con el comando

```
sudo systemctl restart vsftpd
```

Ahora trataremos de acceder a todos los servicios desde el WSL que tenemos instalado en nuestro equipo windows y desde el buscador de nuestro equipo windows:

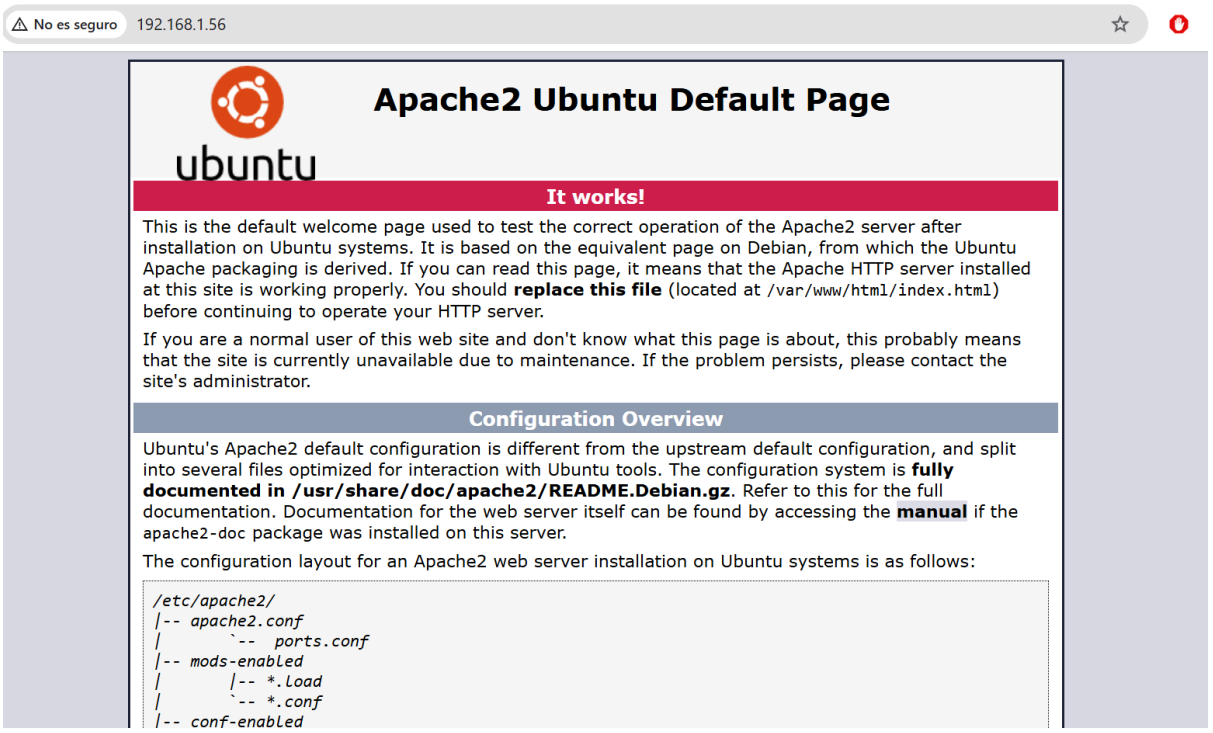
SSH:

```
pablo@LAPTOP-2CN68HQK:~$ ssh pablo@192.168.1.56
The authenticity of host '192.168.1.56 (192.168.1.56)' can't be established.
ED25519 key fingerprint is SHA256:Rlz2+VWSMat+e9kMAH3Kut+F50m/oXx+Vsn9Cb141YA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.56' (ED25519) to the list of known hosts.
pablo@192.168.1.56's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.15.0-57-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Se pueden aplicar 419 actualizaciones de forma inmediata.
314 de estas son actualizaciones de seguridad estándares.
Para ver estas actualizaciones adicionales ejecute: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.

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.

pablo@abduscan:~$
```

FTP:

```
pablo@LAPTOP-2CN68HQK:~$ ftp pablo@192.168.1.56
Connected to 192.168.1.56.
220 (vsFTPd 3.0.5)
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

APACHE:



Los 3 servicios funcionan correctamente y sin problemas aparentes.

Ahora modificaremos los TCPWrappers para filtrar las conexiones

Para ello tendremos que cambiar los archivos /etc/hosts.allow y /etc/hosts.deny para controlar el acceso a nuestros servicios anteriores.

Modificaremos primero el archivo /etc/hosts.deny

```
pablo@abduscan:/etc$ sudo nano hosts.deny
```

Donde añadiremos la línea siguiente:

```
ALL:ALL
```

Esta línea hará que se bloqueen todas las conexiones a excepción de las que dejemos indicadas en /etc/hosts.allow.

```
pablo@abduscan:/etc$ sudo nano hosts.allow
```

De la misma forma en /etc/hosts.allow añadiremos la siguiente línea:

```
sshd: 192.168.1.0/255.255.255.0
```

Añadiendo esta línea permitimos el acceso a ssh desde la subred indicada

Ahora volveremos a probar el acceso a los servicios con los cambios que hemos hecho, deberíamos ahora observar que no podemos acceder a ftp pero sí a ssh

SSH:

```
pablo@LAPTOP-2CN68HQK:~$ ssh pablo@192.168.1.56
pablo@192.168.1.56's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.15.0-57-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Se pueden aplicar 419 actualizaciones de forma inmediata.
314 de estas son actualizaciones de seguridad estándares.
Para ver estas actualizaciones adicionales ejecute: apt list --upgradable

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Dec  8 15:46:55 2024 from 192.168.1.43
pablo@abduscan:~$
```

FTP:

```
pablo@LAPTOP-2CN68HQK:~$ ftp pablo@192.168.1.56
Connected to 192.168.1.56.
421 Service not available.
ftp>
```

Podemos observar que como pensábamos podemos acceder al servicio ssh pero no a ftp tal y como hemos establecido en nuestros archivos etc/hosts.allow y etc/hosts.deny

Como apunte destacar que a pesar de la orden ALL:ALL que añadimos en hosts.deny el servicio apache continúa funcionando por lo que hemos podido deducir que apache no está bajo el control de TCPWrappers.

Probaremos ahora al revés, cambiando la línea de etc/hosts.allow por la siguiente:

```
vsftpd: 192.168.1.0/255.255.255.0
```

Ahora el resultado debería ser al contrario deberíamos poder acceder a ftp pero no a ssh.

SSH:

```
pablo@LAPTOP-2CN68HQK:~$ ssh pablo@192.168.1.56
kex_exchange_identification: read: Connection reset by peer
Connection reset by 192.168.1.56 port 22
pablo@LAPTOP-2CN68HQK:~$
```

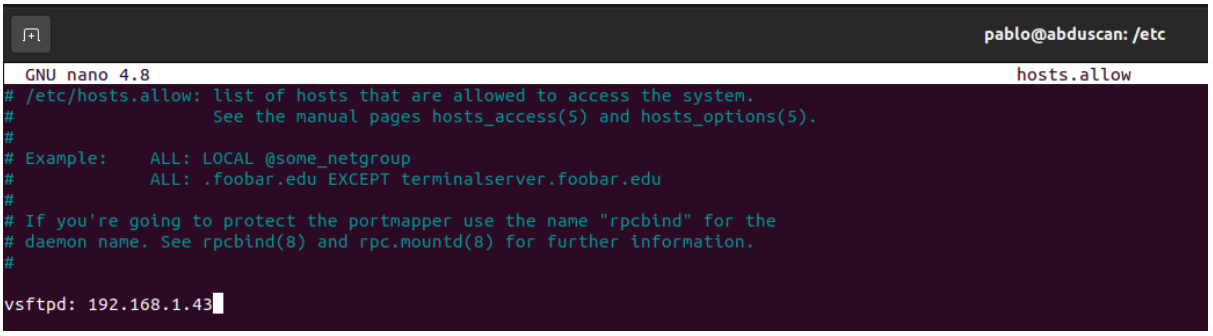
FTP:

```
pablo@LAPTOP-2CN68HQK:~$ ftp pablo@192.168.1.56
Connected to 192.168.1.56.
220 (vsFTPd 3.0.5)
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Efectivamente obtenemos el resultado esperado.

Ahora probaremos a permitir únicamente el acceso a nuestro equipo Windows, desde el que estamos accediendo con la terminal de WSL. Para ello modificamos el archivo etc/hosts.allow de la siguiente forma:

```
Adaptador de LAN inalámbrica Wi-Fi:

   Sufijo DNS específico para la conexión. . :
   Vínculo: dirección IPv6 local. . . : fe80::1bbd:a92b:b12a:9d65%17
   Dirección IPv4. . . . . . . . . . . . : 192.168.1.43
   Máscara de subred . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.1.1
```

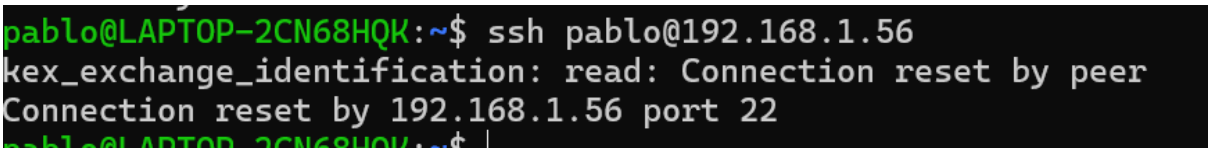


De esta forma solo tendrá acceso a ftp nuestro equipo Windows.

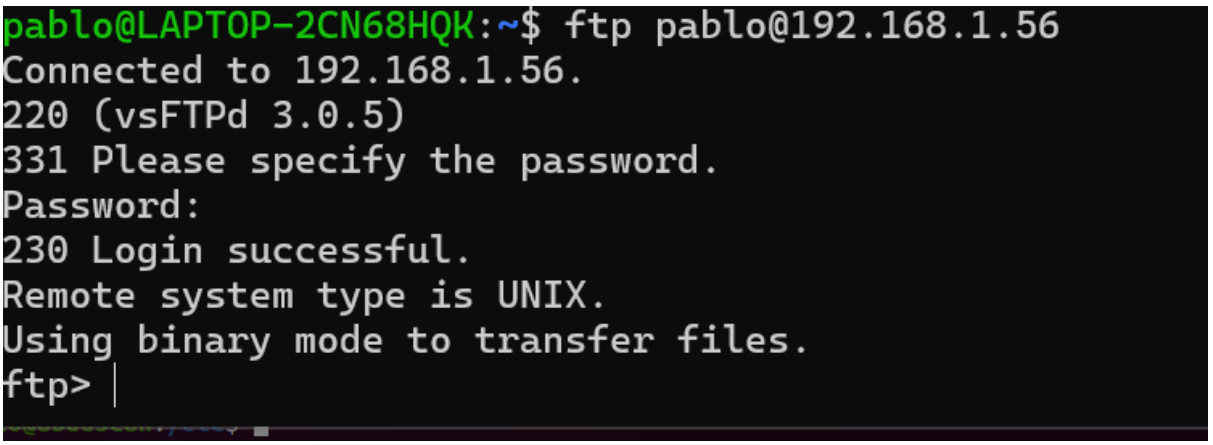
Probamos de nuevo:

En nuestro equipo Windows:

SSH:



FTP:



En otro equipo debian de la misma subred:

Vemos que el resultado es el esperado. La única conexión exitosa ha sido la de ftp desde nuestro equipo windows ya que es lo que tenemos configurado en nuestro etc/hosts.allow

Ahora modificaremos el archivo /etc/sysctl.conf para la seguridad en redes, descomentaremos o añadiremos los siguientes parámetros:

Deshabilitar el reenvío de paquetes IP.

```
net.ipv4.ip_forward=0
```

Para evitar el uso de rutas de origen maliciosas (IP Spoofing).

```
net.ipv4.conf.default.rp_filter=1
net.ipv4.conf.all.rp_filter=1
```

Habilita la protección contra SYN floods mediante el uso de "SYN cookies".

```
net.ipv4.tcp_syncookies=1
```

A continuación ejecutaremos el comando sysctl -p para que los cambios se hagan efectivos.

```
pablo@abduscan:/etc$ sudo sysctl -p
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.ip_forward = 0
pablo@abduscan:/etc$
```

Para probar alguna de estas configuraciones,

Desde nuestro equipo windows con WSL enviaremos paquetes ICMP a nuestro servidor para comprobar si hay IP Forwarding, lo cual impedíamos con el parámetro net.ipv4.ip_forward = 0.

```
pablo@LAPTOP-2CN68HQK:~$ traceroute 192.168.1.56
traceroute to 192.168.1.56 (192.168.1.56), 30 hops max, 60 byte packets
 1  LAPTOP-2CN68HQK.mshome.net (172.29.208.1)  0.366 ms  0.348 ms  0.367 ms
 2  192.168.1.56 (192.168.1.56)  0.979 ms  1.149 ms *
```

Como vemos llega directamente de un equipo a otro sin que haya IP forwarding.

Trabajando con iptables

iptables es una herramienta fundamental en la seguridad de sistemas Linux. Permite filtrar el tráfico de red entrante y saliente basándose en reglas que definen qué paquetes se permiten o bloquean. Estas reglas se organizan en tablas y cadenas.

Tablas: Las principales tablas son filter (para filtrar paquetes), nat (para traducción de direcciones de red) y mangle (para modificar paquetes). La tabla filter es la que se usa con más frecuencia para configurar un firewall básico.

Cadenas: Dentro de cada tabla hay cadenas predefinidas como INPUT (para tráfico entrante), OUTPUT (para tráfico saliente) y FORWARD (para tráfico que pasa a través del servidor).

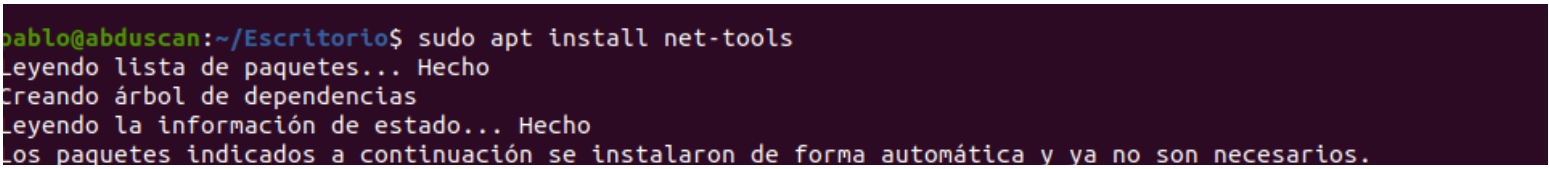
En comparación con TCPWrappers

Aunque ambos mecanismos, iptables y TCP Wrappers, se utilizan para controlar el acceso a los servicios de red, existen diferencias clave:

Característica	iptables	TCP Wrappers
Nivel de operación	Kernel	Espacio de usuario
Granularidad	Muy granular (filtra por puertos, protocolos, direcciones IP, etc.)	Menos granular (filtra principalmente por direcciones IP y nombres de dominio)
Flexibilidad	Mayor flexibilidad, permite reglas complejas	Más simple, fácil de configurar para filtrado básico
Rendimiento	Mayor rendimiento al operar en el kernel	Menor rendimiento al operar en el espacio de usuario
Complejidad	Más complejo de configurar	Más fácil de aprender y usar

Análisis de servicios activos

Ahora vamos a comprobar con la orden netstat todos los servicios activos, primero necesitaremos instalarla



La orden netstat permite monitorizar las conexiones de red de un dispositivo.
Nos da una visión general de lo que está sucediendo en tu red a nivel de conexiones.

Sus funciones principales son:

- **Mostrar conexiones activas:** Podemos ver qué conexiones TCP están activas en tu equipo, incluyendo la dirección IP local y remota, los puertos utilizados y el estado de la conexión. Esto es útil para identificar qué aplicaciones están usando la red y con quién se están comunicando.

- **Identificar puertos en escucha:** netstat muestra qué puertos están "escuchando" en tu equipo, es decir, qué servicios están esperando conexiones entrantes. Esto ayuda a identificar qué servicios están corriendo en tu equipo y si hay algún puerto abierto que no debería estarlo.

- **Ver estadísticas de red:** Obtenemos información sobre las interfaces de red, como la cantidad de paquetes enviados y recibidos, errores, etc. Esto puede ser útil para diagnosticar problemas de rendimiento en la red.

- **Mostrar la tabla de enrutamiento:** netstat permite ver la tabla de enrutamiento IP, que indica cómo se enrutan los paquetes en tu red.

- **Solucionar problemas de red:** En general, netstat es una herramienta muy útil para diagnosticar problemas de red, como conexiones lentas, errores de conexión, etc.

Para ver todos los puertos en escucha utilizaremos el comando netstat con las siguientes opciones:

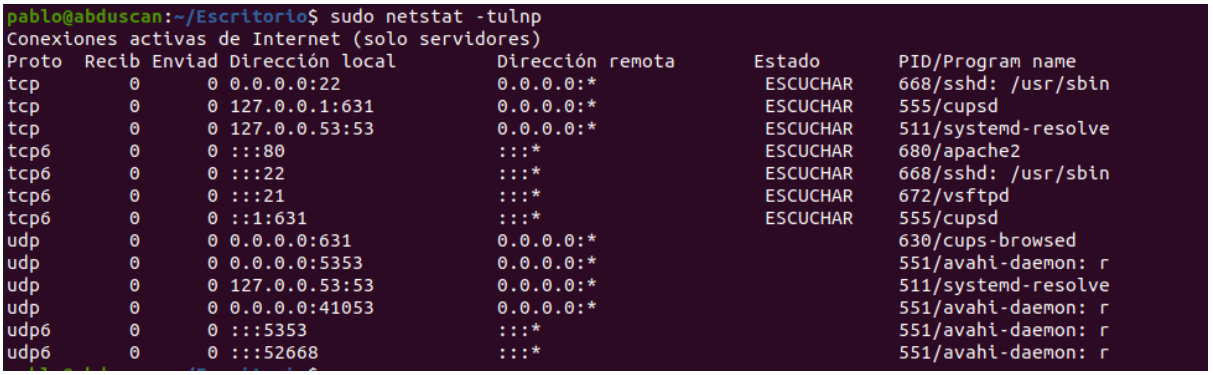
-t: Muestra conexiones TCP.

-u: Muestra conexiones UDP.

-l: Muestra solo los servicios en escucha.

-n: Muestra direcciones y puertos en formato numérico.

-p: Muestra el ID del proceso (PID) y el nombre del programa asociado a cada socket. Esto permite saber qué aplicación está usando cada puerto.



Aquí observamos los distintos servicios que hemos utilizado, ssh, apache, vsftpd, etc.

Para ver todos los servicios activos utilizaremos el comando netstat con las siguientes opciones:

-a:Mostrar todas las conexiones y puertos en escucha.

-n: Muestra direcciones y puertos en formato numérico.

-p: Muestra el ID del proceso (PID) y el nombre del programa asociado a cada socket. Esto permite saber qué aplicación está usando cada puerto.

```
pablo@abduscan:~/Escritorio$ sudo netstat -anp
Conexiones activas de Internet (servidores y establecidos)
Proto  Recib Enviad Dirección local         Dirección remota       Estado       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*              ESCUCHAR     668/sshd: /usr/sbin
tcp        0      0 127.0.0.1:631           0.0.0.0:*              ESCUCHAR     555/cupsd
tcp        0      0 127.0.0.53:53           0.0.0.0:*              ESCUCHAR     511/systemd-resolve
tcp6       0      0 :::80                   :::*                   ESCUCHAR     680/apache2
tcp6       0      0 :::22                   :::*                   ESCUCHAR     668/sshd: /usr/sbin
tcp6       0      0 :::21                   :::*                   ESCUCHAR     672/vsftpd
tcp6       0      0 ::1:631                 :::*                   ESCUCHAR     555/cupsd
udp        0      0 0.0.0.0:631             0.0.0.0:*                           630/cups-browsed
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           551/avahi-daemon: r
udp        0      0 127.0.0.53:53           0.0.0.0:*                           511/systemd-resolve
udp        0      0 192.168.1.56:68         192.168.1.1:67         ESTABLECIDO  557/NetworkManager
udp        0      0 0.0.0.0:41053           0.0.0.0:*                           551/avahi-daemon: r
udp6       0      0 :::5353                 :::*                                551/avahi-daemon: r
udp6       0      0 :::52668                :::*                                551/avahi-daemon: r
raw6       0      0 :::58                   :::*                   7            557/NetworkManager
Sockets activos de dominio UNIX (servidores y establecidos)
Proto RefCnt Flags       Type       State         I-Node   PID/Program name     Ruta
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    33320    989/gdm3             @/tmp/dbus-ovoKmKTJ
unix  2      [ ]         DGRAM                    32728    1363/systemd         /run/user/1000/systemd/notify
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    32731    1363/systemd         /run/user/1000/systemd/private
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    32777    1363/systemd         /run/user/1000/bus
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    32778    1363/systemd         /run/user/1000/gnupg/S.dirmngr
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    32779    1363/systemd         /run/user/1000/gnupg/S.gpg-agent.browser
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    32780    1363/systemd         /run/user/1000/gnupg/S.gpg-agent.extra
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    32781    1363/systemd         /run/user/1000/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    32782    1363/systemd         /run/user/1000/gnupg/S.gpg-agent
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    32783    1363/systemd         /run/user/1000/pk-debconf-socket
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    32784    1363/systemd         /run/user/1000/pulse/native
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    32785    1363/systemd         /run/user/1000/snapd-session-agent.socket
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    35269    1612/gnome-session-  @/tmp/.ICE-unix/1612
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    32932    1375/gnome-keyring-  /run/user/1000/keyring/control
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    33491    1424/Xorg            @/tmp/.X11-unix/X0
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    35141    1375/gnome-keyring-  /run/user/1000/keyring/pkcs11
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    35380    1375/gnome-keyring-  /run/user/1000/keyring/ssh
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    33492    1424/Xorg            /tmp/.X11-unix/X0
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    34735    1576/ssh-agent       /tmp/ssh-amhhswe5tCfv/agent.1470
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    35270    1612/gnome-session-  /tmp/.ICE-unix/1612
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    50973    1727/gvfsd-trash     @/dbus-vfs-daemon/socket-V7lD1apu
unix  3      [ ]         DGRAM      CONECTADO     15558    1/init               /run/systemd/notify
unix  2      [ ACC ]     FLUJO      ESCUCHANDO    15561    1/init               /run/systemd/private
```

Ahora utilizaremos nmap para comprobar los puertos abiertos en el equipo donde se configuraron los servicios de consola remota, transferencia de ficheros y WEB.

```
pablo@LAPTOP-2CN68HQK:~$ nmap -p 21,22,80 192.168.1.56
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-08 19:32 CET
Nmap scan report for 192.168.1.56
Host is up (0.0011s latency).

PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

Con esta orden escaneamos los puertos indicados de la Ip introducida.

Otra técnica de sondeo que podemos utilizar con nmap es:

Sondeo SYN (-sS)

- **Descripción:**
  - También conocido como "half-open scan".
  - Solo envía un paquete SYN al puerto objetivo y espera una respuesta:
    - Si el puerto está abierto, recibe un SYN/ACK.

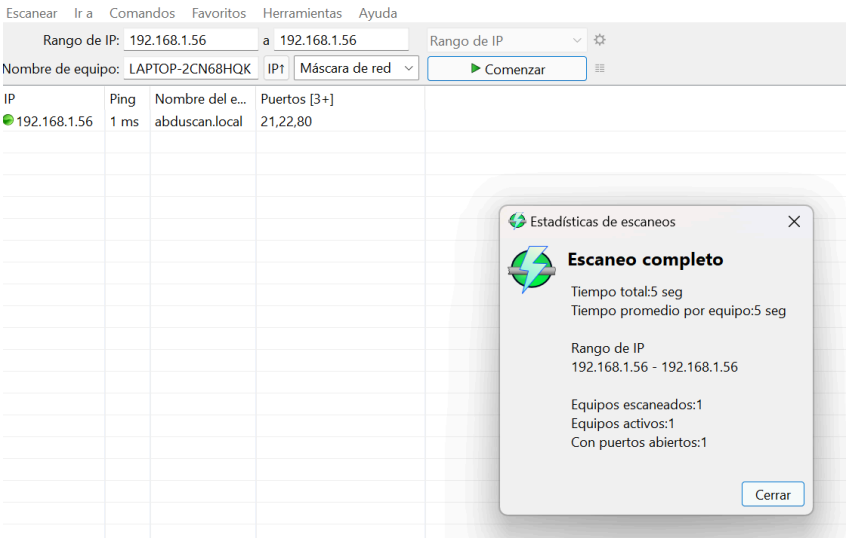
        - Si está cerrado, recibe un RST.
    - No completa la conexión (no envía el paquete ACK), por lo que es más rápido y sigiloso.
- **Ventajas:**
    - Más rápido y menos detectable.
    - Evita registrar conexiones completas en los logs del sistema.

```
pablo@LAPTOP-2CN68HQK:~$ sudo nmap -sS 192.168.1.56
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-08 19:36 CET
Nmap scan report for 192.168.1.56
Host is up (0.0019s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
pablo@LAPTOP-2CN68HQK:~$
```

Herramientas alternativas a nmap

Una alternativa interesante a Nmap para el análisis de puertos es **Angry IP Scanner**, una herramienta gratuita y multiplataforma que permite realizar escaneos de red de manera sencilla. Entre sus características, incluye:

- Interfaz gráfica
- Resultados personalizados
- Rapidez
- Amigable



Ejemplo de uso