Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Práctica 1: Entorno de trabajo auditoría de redes inalámbricas

Seguridad de las Comunicaciones Inalámbricas

Autor: Cheuk Kelly Ng Pante (alu0101364544@ull.edu.es)

Fecha: 8 de diciembre de 2025

Índice general

1. Análisis y Configuración de Interfaces de Red

Ejecutar el comando para listar todas las interfaces de red:

```
ip addr show
```

Forma abreviada: `ip a`

1.1. Enumeración de Interfaces

- **Dirección MAC:** Valor `link/ether`
- **Direcciones IP:** `inet` (IPv4) e `inet6` (IPv6)
- **Dirección Broadcast:** Valor `brd`
- **Estado:** `STATE UP` o `STATE DOWN`
- **MTU:** Valor numérico `mtu`

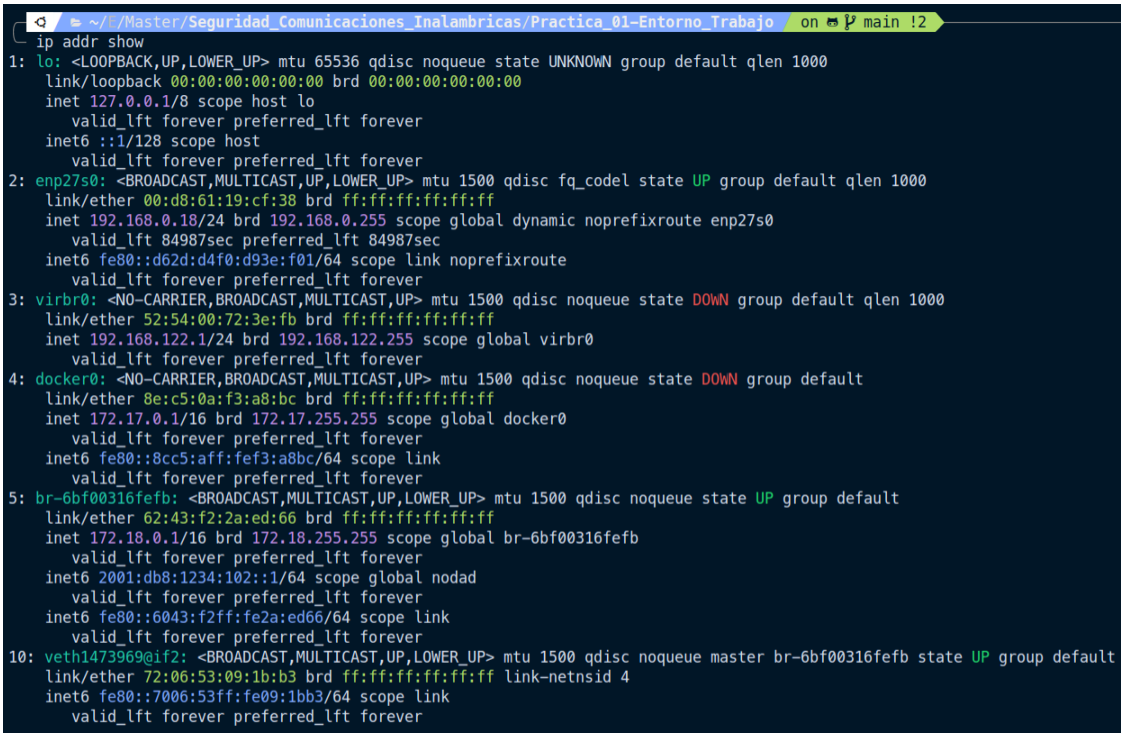


Figura 1.1: Salida del comando `ip addr show`

2. Gesión del Estado de la Interfaz y Escalado de Privilegios

2.1. Intento sin privilegios

Intentar deshabilitar la interfaz eth0:

```
ip link set eth0 down
```

Resultado esperado: Error de permiso denegado.

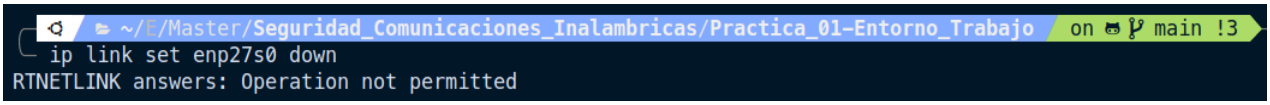


Figura 2.1: Error al intentar deshabilitar eth0 sin privilegios

2.2. Ejecución con privilegios

Ejecutar con `sudo`:

```
sudo ip link set eth0 down
```

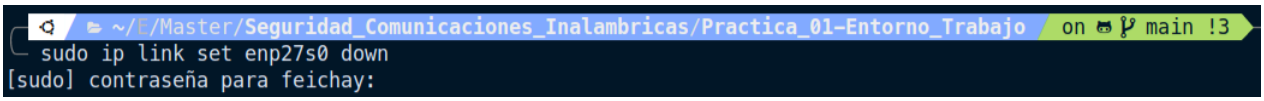


Figura 2.2: Deshabilitación exitosa de eth0 con privilegios

2.3. Verificación de conectividad

Confirmar que la interfaz está inactiva:

```
ip a
```

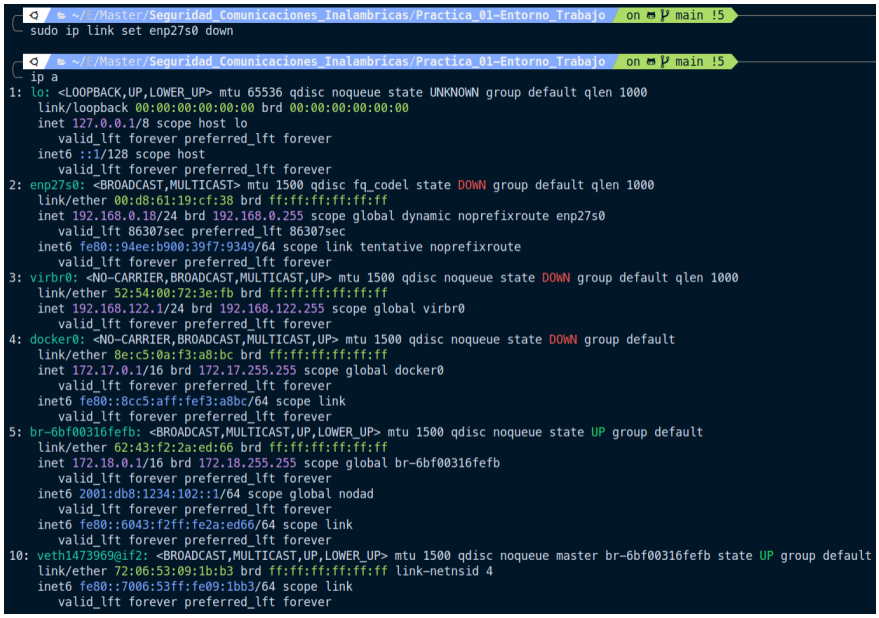


Figura 2.3: Verificación de estado inactivo de eth0

Al acceder a alguna página web podemos ver que se ha perdido la conexion:

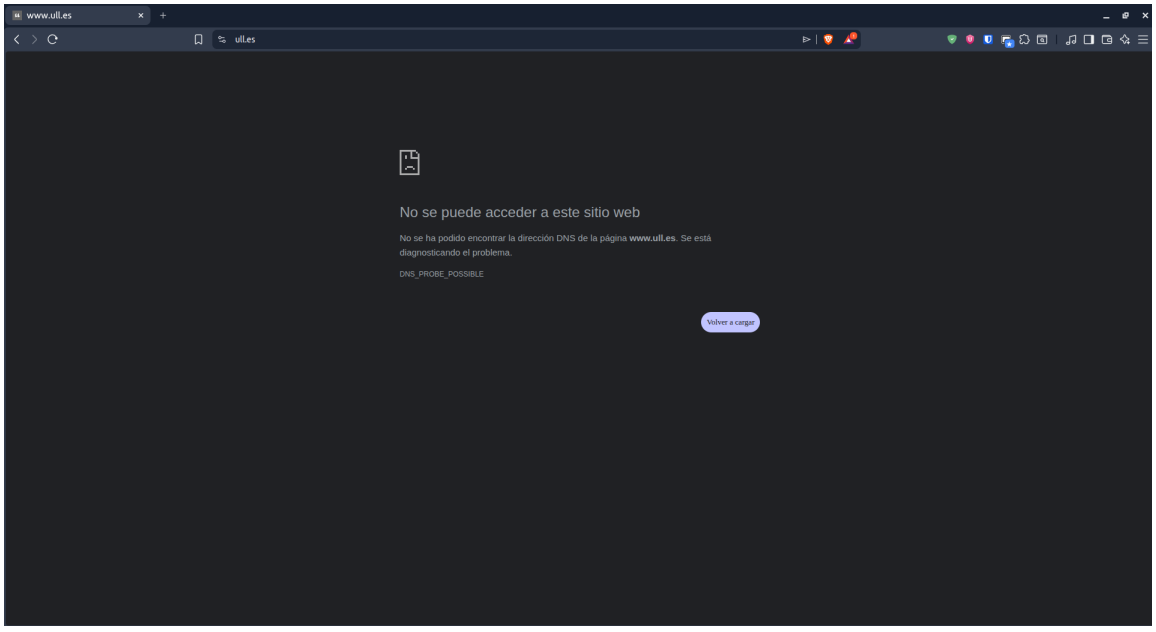


Figura 2.4: Perdida de conexion

2.4. Reactivación

Restaurar la interfaz:

```
sudo ip link set eth0 up
```

Validar la restauración con `ip a` y verificar conectividad a Internet.

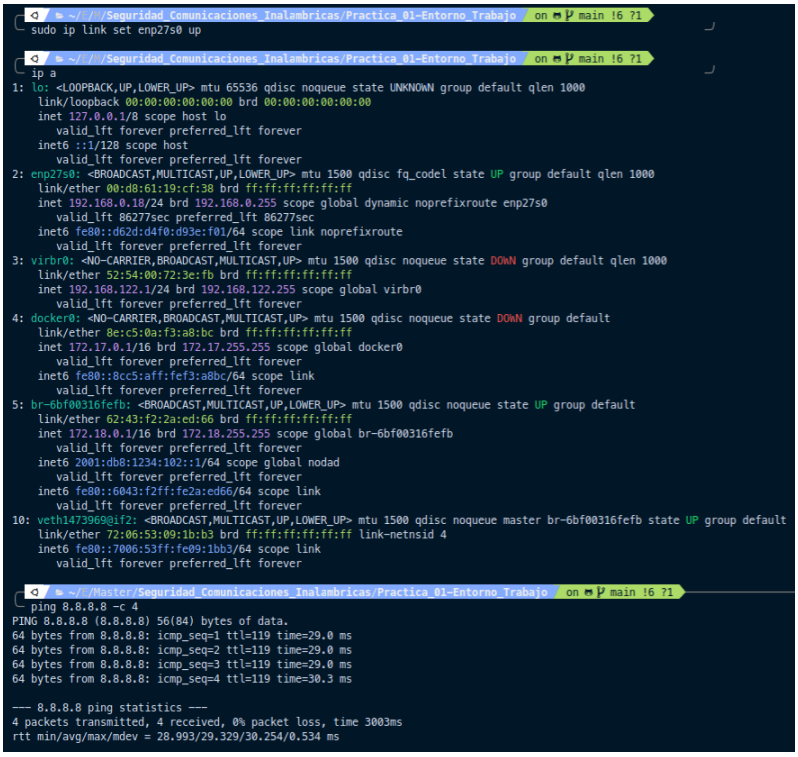


Figura 2.5: Reactivación exitosa de eth0

2.4.1. Comandos adicionales

```
sudo ip addr add 192.168.1.200/24 dev eth0
```

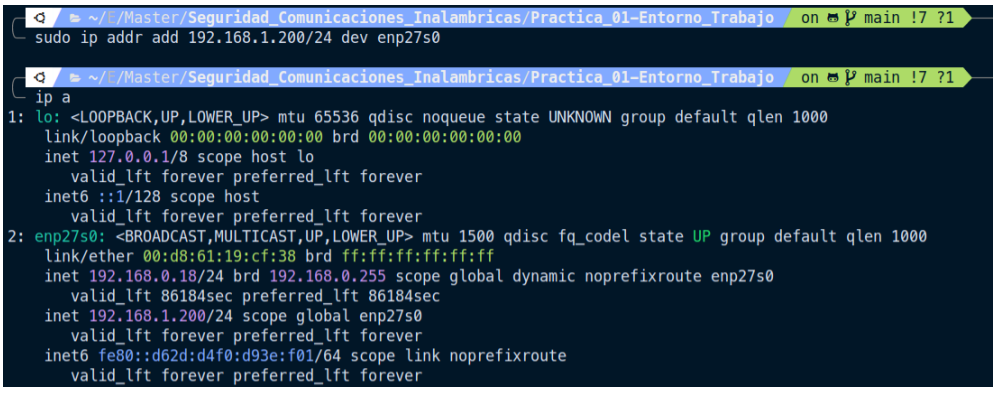


Figura 2.6: Asignación temporal de nueva IP a eth0

Al hacer esto el cambio es temporal y se pierde al reiniciar. Para hacerlo persistente hay que editar el fichero `/etc/netplan/01-network-manager-all.yaml`.

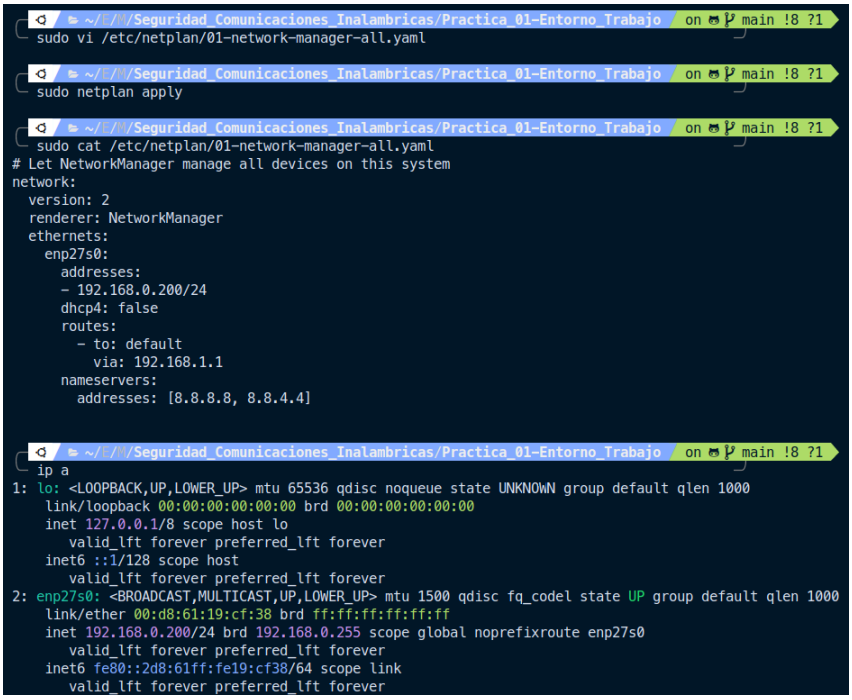


Figura 2.7: Verificación de nueva IP asignada a eth0

3. Verificación de la Pila TCP/IP y Conectividad Externa

3.1. Prueba de Loopback

Ping a la interfaz de loopback:

```
ping -c 4 127.0.0.1
```

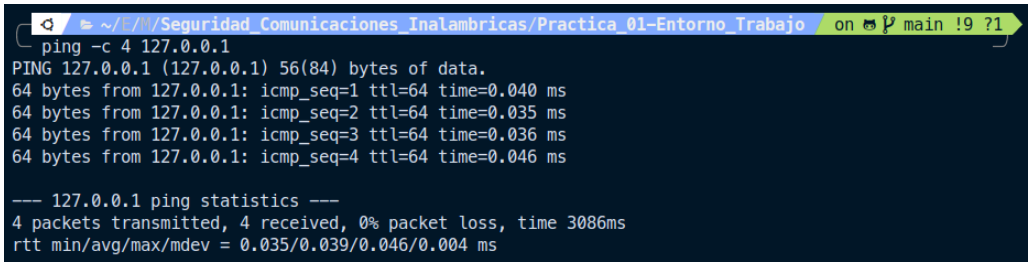


Figura 3.1: Prueba de conectividad a loopback

3.2. Prueba de Conectividad Externa

Verificar resolución DNS y conectividad WAN:

```
ping -c 4 www.ull.es
```

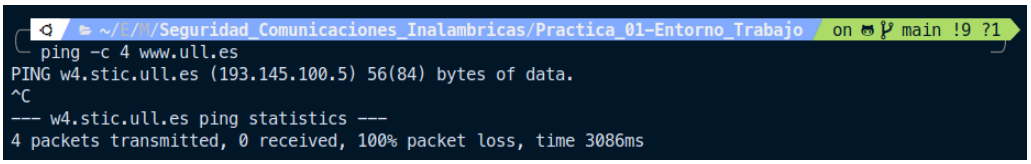


Figura 3.2: Prueba de conectividad a www.ull.es

A continuación se documentan los resultados obtenidos tras ejecutar el comando de diagnóstico de red hacia el dominio de la Universidad de La Laguna.

a. Estadísticas de Paquetes Basado en la línea final: *4 packets transmitted, 0 received, 100 % packet loss.*

- **Transmitidos:** 4
- **Recibidos:** 0
- **Pérdida (%):** 100 %

b. Estadísticas de RTT (Round-Trip Time) Debido a que la pérdida de paquetes fue total (ningún paquete retornó), el sistema no pudo calcular los tiempos de viaje.

- **Mínimo:** N/A (No disponible)
- **Promedio (Avg):** N/A (No disponible)
- **Máximo:** N/A (No disponible)

Observación: El fallo en la recepción de paquetes (100 % de pérdida) sugiere que el host destino (`193.145.100.5`) está inactivo o, lo más probable, que existe un firewall bloqueando las solicitudes ICMP.

4. Análisis de la Caché del Protocolo ARP

Al realizar la práctica en Ubuntu 22.04, se observó que el comando clásico arp -a no está disponible por defecto, ya que pertenece al paquete obsoleto net-tools. Por este motivo, se utiliza la suite moderna iproute2, donde el comando ip neigh (abreviatura de ip neighbor) permite consultar y gestionar la tabla de vecinos, que para IPv4 corresponde a la caché ARP.

En consecuencia, en este informe se emplea el comando ip neigh show como alternativa directa a arp -a para inspeccionar el mapeo entre direcciones IP y direcciones MAC de los dispositivos de la red, incluida la puerta de enlace.

El comando utilizado es:

```
ip neigh
```

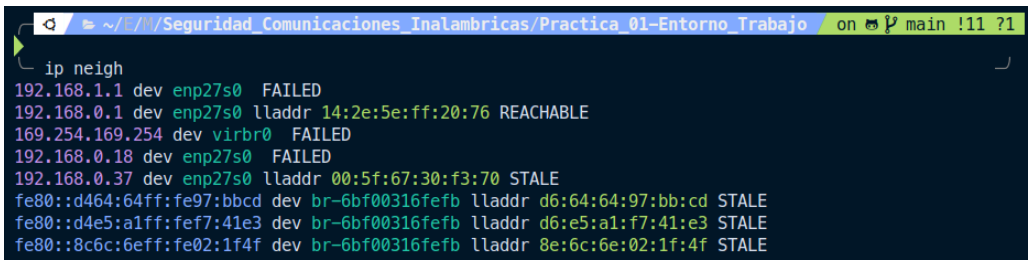


Figura 4.1: Salida del comando `ip neigh`

5. Inspección de la Tabla de Enrutamiento

Analizar cómo el sistema decide dónde enviar el tráfico. Para mostrar la tabla de enrutamiento:

```
ip route show
```

Forma abreviada: `ip r`

En la salida del comando ip route show se observan dos rutas por defecto configuradas en la interfaz `enp27s0`. Estas rutas apuntan a dos puertas de enlace distintas, `192.168.1.1` (configurada de forma estática) y `192.168.0.1` (obtenida por DHCP), ambas con la misma métrica (20100), por lo que el kernel puede utilizar cualquiera de ellas como *default gateway* en función de su disponibilidad.

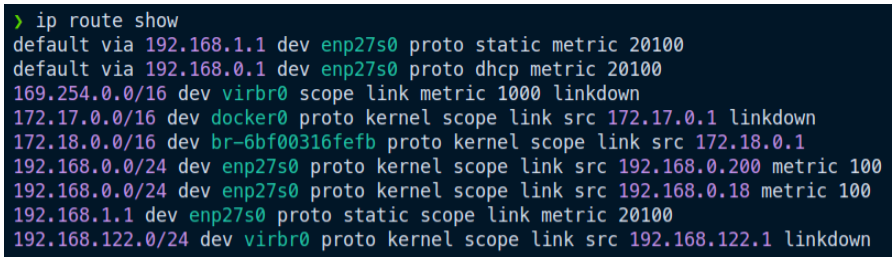
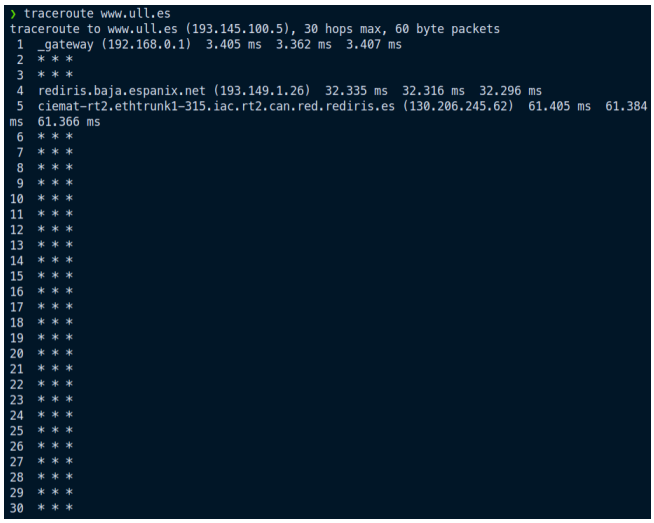


Figura 5.1: Salida del comando `ip route show`

6. Trazado de Ruta de Red (Traceroute)

Los trazados de ruta realizados son:

```
traceroute www.ull.es
traceroute www.net.berkeley.edu
```



(a) Trazado de ruta a www.ull.es

(b) Trazado de ruta a www.net.berkeley.edu

Figura 6.1: Trazados de ruta a dos destinos diferentes

Al realizar la práctica en Ubuntu 22.04 se ejecutó el comando `traceroute` hacia dos destinos: `www.net.berkeley.edu` y `www.ull.es`, con el objetivo de identificar los routers intermedios y la latencia asociada a cada salto de Capa 3. En el caso de `www.net.berkeley.edu`, el trazado muestra un total de 14 saltos desde la puerta de enlace local (`192.168.0.1`), con un tiempo de ida y vuelta inicial de aproximadamente 3 ms y latencias que aumentan progresivamente hasta valores cercanos a 170–180 ms en el destino final, lo que refleja la distancia geográfica y el número de dominios de red atravesados.

Para `www.ull.es`, el traceroute alcanza la puerta de enlace local y varios routers pertenecientes a la red académica (por ejemplo, nodos de RedIRIS), con latencias en torno a 30–60 ms. A partir de cierto punto, numerosos saltos aparecen como `* * *`, lo que indica que esos routers no responden a las sondas de `traceroute` (por filtrado o limitación de tráfico ICMP), aun cuando la conectividad con el destino sigue siendo funcional.

7. Escaneo de Puertos Locales con Nmap

Para realizar el auto-descubrimiento de servicios se ejecutó un escaneo con `nmap` sobre la propia máquina (`nmap -sV localhost`). Este comando permite identificar qué puertos TCP se encuentran en estado *LISTEN* y, además, intenta determinar la versión de los servicios asociados mediante el flag `-sV`, lo que simula un reconocimiento inicial desde la perspectiva de un atacante situado en la misma red local.

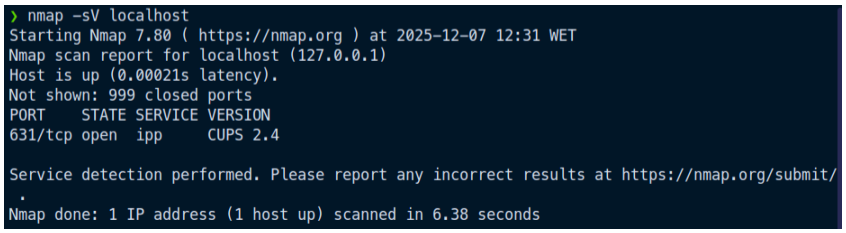


Figura 7.1: Salida del comando `nmap -sV localhost`

Al ejecutar el comando `nmap -sV localhost` sobre la propia máquina, se comprobó que el host `localhost` (`127.0.0.1`) está activo y responde con una latencia muy baja. El escaneo analiza los 1000 puertos TCP más habituales y muestra que 999 de ellos se encuentran cerrados, detectándose únicamente un puerto abierto: el `631/tcp`, asociado al servicio `ipp` (Internet Printing Protocol), que en este caso corresponde a `CUPS 2.4`, es decir, el subsistema de impresión del sistema operativo escuchando en la interfaz de *loopback*.

8. Análisis de Sockets y Conexiones de Red

Para inspeccionar los sockets y conexiones de red activas en el sistema, se utilizó el comando `ss` (socket statistics), que forma parte de la suite `iproute2` y ofrece una visión detallada de las conexiones TCP, UDP y otros tipos de sockets.

```
1 ss -tuln
```

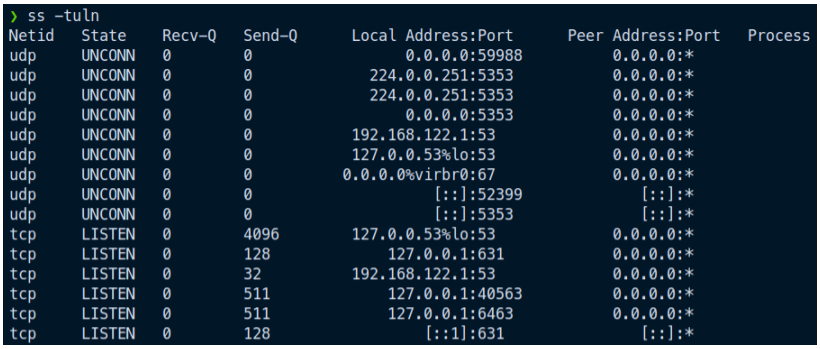


Figura 8.1: Salida del comando `ss -tuln`

Al ejecutar el comando `ss -tuln` se obtuvo un listado de los sockets UDP y TCP abiertos en el sistema, incluyendo su estado y la dirección/puerto local asociados. En la parte UDP, los sockets aparecen en estado `UNCONN`, con varios puertos `53` y `5353` escuchando en direcciones como `127.0.0.53%lo`, `192.168.122.1` y `0.0.0.0`, lo que indica la presencia de servicios de resolución de nombres (DNS) y de descubrimiento en la red local, así como un servidor DHCP vinculado a la interfaz virtual `virbr0`.

En la sección TCP, los sockets se muestran en estado `LISTEN`, destacando que la mayoría de servicios están restringidos a la interfaz de *loopback* (por ejemplo, `127.0.0.53%lo:53`, `127.0.0.1:631` y `[::1]:631`, correspondientes a servicios de DNS local y CUPS), mientras que otros puertos como `192.168.122.1:53` sólo son accesibles desde la red virtual interna. En conjunto, la salida evidencia que los servicios expuestos

hacia el exterior son mínimos y que los puertos más sensibles se limitan a la propia máquina o a redes virtuales controladas.

9. Resolución de Nombres DNS (NSLookup)

9.1. Objetivo

Realizar consultas DNS para resolver nombres de dominio a `www.ull.es` y `www.w3c.org` se utilizaron consultas DNS mediante `nslookup`. En ambos casos se identificó el servidor DNS que resolvió la consulta (línea `Server:`) y se registraron las direcciones IPv4 devueltas como registros de tipo A para cada dominio.

```
❯ nslookup www.ull.es
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
www.ull.es      canonical name = w4.stic.ull.es.
Name:   w4.stic.ull.es
Address: 193.145.100.5

❯ nslookup www.w3c.org
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
www.w3c.org     canonical name = webredir.vip.gandi.net.
Name:   webredir.vip.gandi.net
Address: 217.70.184.50
```

Figura 9.1: Salida del comando `nslookup <dominio>`

En las consultas DNS realizadas con `nslookup` para `www.ull.es` y `www.w3c.org`, el servidor que resuelve ambas peticiones es `127.0.0.53`, es decir, el resolvedor local de Ubuntu que actúa como caché y reenvía las consultas a los DNS configurados en el sistema.

Para `www.ull.es`, la respuesta incluye un nombre canónico `w4.stic.ull.es` y un registro A con la dirección IPv4 `193.145.100.5`. Para `www.w3c.org`, el nombre canónico resuelto es `webredir.vip.gandi.net` y el registro A obtenido es la dirección IPv4 `217.70.184.50`

10. Interacción de Red con Netcat (nc)

Se utilizó Netcat (`nc`) como herramienta de depuración de red. En modo escucha se empleó la sintaxis `nc -l -p 1234` para iniciar un servicio que acepta conexiones entrantes en el puerto 1234. En modo cliente se utilizó `nc <host>1234` para establecer una conexión TCP hacia dicho puerto remoto y verificar el intercambio de datos extremo a extremo.

10.1. Sintaxis clave

Modo Escucha (Listener):

```
nc -l -p 1234
```

Modo Cliente:

```
nc [IP_remota] [puerto]
```

En la Figura 10.1 se muestra la configuración de Netcat en modo escucha y cliente en dos terminales diferentes.

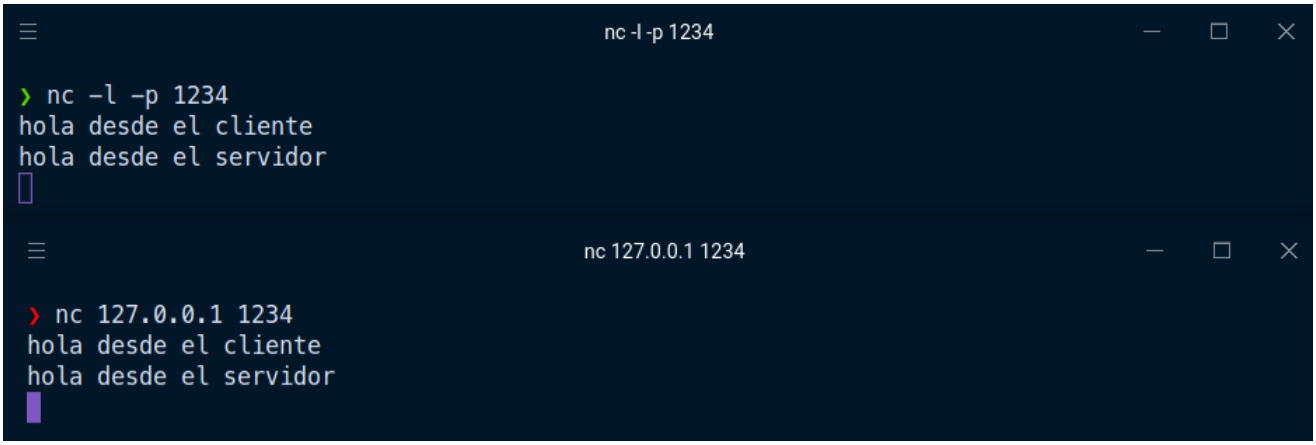


Figura 10.1: Uso de Netcat en modo escucha y cliente

Para probar la interacción de red con Netcat se lanzó un listener en la máquina local mediante `nc -l -p 1234` y, en una segunda terminal, un cliente con `nc 127.0.0.1 1234`. Una vez establecida la conexión, cualquier texto introducido en una de las terminales aparecía en la otra, demostrando que se había creado un canal TCP bidireccional entre cliente y servidor sobre el puerto 1234 de `localhost`, lo que simula el funcionamiento básico de un servicio y su cliente en la misma red.

11. Reconocimiento DNS Avanzado con dnsenum

Se ejecutó `dnsenum tecnomobile.com` para realizar enumeración DNS automatizada del dominio, obteniendo direcciones de host (registros A/AAAA), servidores de nombres (NS), servidores de correo (MX) y, adicionalmente, posibles subdominios y rangos de red asociados descubiertos por fuerza bruta y consultas complementarias.

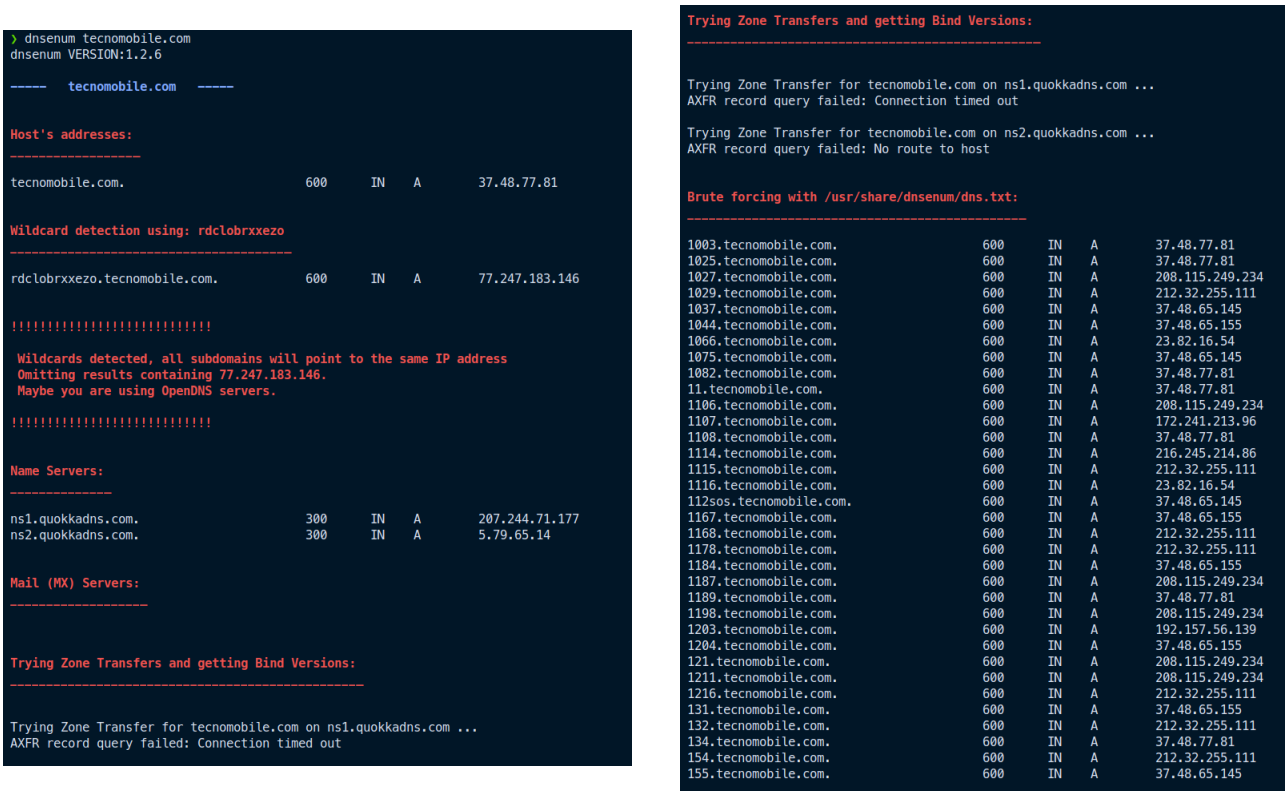


Figura 11.1: Salida del comando `dnsenum tecnomobile.com`

Mediante la ejecución de `dnsenum tecnomobile.com` se identificó que el dominio principal `tecnomobile.com` dispone de un registro A asociado a la dirección IPv4 `37.48.77.81`. Asimismo, se detectaron como servidores de nombres (registros NS) los hosts `ns1.quokkadns.com` (`207.244.71.177`) y `ns2.quokkadns.com` (`5.79.65.14`). La herramienta no devolvió registros MX para el dominio, pero sí enumeró varios subdominios numéricos (por ejemplo, `1003.tecnomobile.com`, `1025.tecnomobile.com`, etc.) con distintas direcciones IPv4, lo que aporta información adicional sobre la infraestructura asociada al dominio.

12. Análisis de Tráfico en Terminal con tcpdump

Se realizó una captura básica con `tcpdump` mediante el comando `sudo tcpdump -i enp27s0 -n -c 20`. En las primeras líneas se observa tráfico de resolución ARP dentro de la red local (petición para averiguar la MAC de `192.168.0.1` desde `192.168.0.37`), así como algún frame Ethernet de broadcast de tipo no identificado por la herramienta. A continuación aparecen múltiples paquetes IP entre la máquina local `192.168.0.200` y distintos servidores externos en el puerto `443`, tanto en UDP (probablemente tráfico QUIC) como en TCP, donde pueden verse flags de *push* y *finish* que indican el envío de datos de aplicaciones web y el cierre ordenado de las conexiones TLS.

```
❯ sudo tcpdump -i enp27s0 -n -c 20
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp27s0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
13:38:24.880293 ARP, Request who-has 192.168.0.1 tell 192.168.0.37, length 46
13:38:25.780269 00:5f:67:30:f3:70 > ff:ff:ff:ff:ff:ff, ethertype Unknown (0x8f83), length 60:
        0x0000:  0802 0002 0000 33ae 0000 0000 0003 7374  ......3.......st
        0x0010:  0000 6439 0000 0000 0000 0000 0000 0000  ..d9............
        0x0020:  0000 0000 0000 0000 0000 0000 0000       ..............
13:38:26.436881 IP 192.168.0.200.56712 > 142.250.200.110.443: UDP, length 1250
13:38:26.484666 IP 142.250.200.110.443 > 192.168.0.200.56712: UDP, length 1250
13:38:26.485128 IP 192.168.0.200.56712 > 142.250.200.110.443: UDP, length 774
13:38:26.503207 IP 142.250.200.110.443 > 192.168.0.200.56712: UDP, length 24
13:38:26.510503 IP 192.168.0.200.56712 > 142.250.200.110.443: UDP, length 32
13:38:26.542296 IP 142.250.200.110.443 > 192.168.0.200.56712: UDP, length 28
13:38:26.553710 IP 142.250.200.110.443 > 192.168.0.200.56712: UDP, length 66
13:38:26.553935 IP 192.168.0.200.56712 > 142.250.200.110.443: UDP, length 35
13:38:26.618406 IP 142.250.200.110.443 > 192.168.0.200.56712: UDP, length 24
13:38:26.918768 IP 192.168.0.200.53016 > 141.95.47.140.443: Flags [P.], seq 2107674895:2107674919, ack
 3016217431, win 615, options [nop,nop,TS val 1204892893 ecr 1447076448], length 24
13:38:26.918794 IP 192.168.0.200.53016 > 141.95.47.140.443: Flags [F.], seq 24, ack 1, win 615, option
s [nop,nop,TS val 1204892893 ecr 1447076448], length 0
13:38:26.931665 IP 192.168.0.200.53022 > 141.95.47.140.443: Flags [P.], seq 3921099354:3921099378, ack
 163463223, win 665, options [nop,nop,TS val 1204892905 ecr 1447076788], length 24
13:38:26.931696 IP 192.168.0.200.53022 > 141.95.47.140.443: Flags [F.], seq 24, ack 1, win 665, option
s [nop,nop,TS val 1204892905 ecr 1447076788], length 0
13:38:26.970649 IP 141.95.47.140.443 > 192.168.0.200.53016: Flags [F.], seq 1, ack 24, win 504, option
s [nop,nop,TS val 1447085984 ecr 1204892893], length 0
13:38:26.970691 IP 192.168.0.200.53016 > 141.95.47.140.443: Flags [.], ack 2, win 615, options [nop,no
p,TS val 1204892944 ecr 1447085984], length 0
13:38:26.974291 IP 141.95.47.140.443 > 192.168.0.200.53016: Flags [.], ack 25, win 504, options [nop,n
op,TS val 1447085985 ecr 1204892893], length 0
13:38:26.983716 IP 141.95.47.140.443 > 192.168.0.200.53022: Flags [F.], seq 1, ack 25, win 504, option
s [nop,nop,TS val 1447085997 ecr 1204892905], length 0
13:38:26.983754 IP 192.168.0.200.53022 > 141.95.47.140.443: Flags [.], ack 2, win 665, options [nop,no
p,TS val 1204892958 ecr 1447085997], length 0
20 packets captured
20 packets received by filter
0 packets dropped by kernel
```

Figura 12.1: Salida del comando `tcpdump`