

[Redes] Comandos: tracert/traceroute

Función

- En Windows usaremos `tracert` y en Linux `traceroute`. Es posible que sea necesario instalarlo en Linux porque no venga instalado por defecto.
- Sirven para visualizar la ruta que siguen los paquetes, volumen de información, desde un punto de la red, o host, a otro.
- **Enrutamiento**
 - Para garantizar que la información (paquetes) llega a su destino desde el punto de emisión se tienen que “encaminar” es decir enrutarlos en una misma dirección y de ello se encargan los enrutadores que son dispositivos para las interconexiones de red.
 - Todo este camino que recorre la información crea una traza es decir, un recorrido en el que intervienen enrutadores para la transmisión de paquetes (volúmenes de información) y todo esto lo podemos observar con el comando `tracert`. Dependiendo si el recorrido es mas o menos largo, tendrá mas o menos saltos.
- **Utilidad.** Sabiendo el recorrido de información podemos usarlo para muchas cosas como por ejemplo solventar determinados problemas de red, cuanto tiempo se detiene la información en un determinado punto, fallos en routeadores... entre otros.

tracert

- Si queremos hacer `tracert` a una ip, introduciremos en la consola `tracert` y a continuación la IP:

```
tracert 192.168.1.23
```

- Si queremos hacer `tracert` a una Web, escribiremos `tracert` y la url encabezada por `www` (recordar no poner `http`):

```
tracert www.google.es
```

- La consola nos mostrará el trazado de la conexión indicada mostrándonos los saltos (cada una de las detenciones de los paquetes de transmisión) y el tiempo en milisegundos.

Parámetro y función

- `-d` . Especifica que no se resuelvan las direcciones en nombres de host.
- `-h` . `maximum_hops`. Especifica el número máximo de saltos para alcanzar el destino.
- `-j` . `host-list`. Especifica la ruta de origen a lo largo de la lista de hosts.
- `-w` . `timeout`. Espera el número de milisegundos especificado de tiempo.
- `-target_name` . Nombre o dirección IP del host de destino.

traceroute

- Comando básico

```
traceroute ejemplo.com
```

- Si queremos ver los saltos con IP en lugar de nombres:

```
traceroute -n ejemplo.com
```

- Pequeñas diferencias frente a `tracert`
 - `tracert` usa ICMP por defecto en Windows.
([pandorafms](https://pandorafms.com/es/it-topics/tracert-o-traceroute/))(<https://pandorafms.com/es/it-topics/tracert-o-traceroute/>)

- o traceroute en Linux suele usar UDP por defecto, pero podemos forzar ICMP con la opción `-I` si lo necesitamos:

```
traceroute -I ejemplo.com
```

tracepath

- tracepath es una herramienta de diagnóstico de red en Linux que rastrea la ruta desde nuestro equipo hasta un host remoto, similar a traceroute, pero integrando también información sobre la MTU (Unidad de Transmisión Máxima) y la latencia en cada salto.
- No requiere permisos especiales (a diferencia de algunas versiones de traceroute) y suele usarse para depurar problemas de red y de MTU.
[labex](<https://labex.io/es/tutorials/linux-linux-tracepath-command-with-practical-examples-422964>)

Función principal

1. Muestra la secuencia de enrutadores (saltos) por los que pasan los paquetes hacia un destino (IP o nombre de dominio).
[juandiegolizarazo.blogspot](<http://juandiegolizarazo.blogspot.com/2018/04/tracepath-y-traceroute-linux.html>)
2. Incluye el tiempo de ida-y-vuelta (RTT) en cada salto y la MTU efectiva en el camino, útil para detectar cuellos de botella o problemas de fragmentación.
[labex](<https://labex.io/es/tutorials/linux-linux-tracepath-command-with-practical-examples-422964>)

Sintaxis básica

- En la mayoría de distribuciones de Linux, la sintaxis es muy simple:
[juandiegolizarazo.blogspot](<http://juandiegolizarazo.blogspot.com/2018/04/tracepath-y-traceroute-linux.html>)

```
tracepath [opciones] <destino>
```

donde:

1. `<destino>` es una IP (`8.8.8.8`) o un nombre de host (`google.com`).
[labex](<https://labex.io/es/tutorials/linux-linux-tracepath-command-with-practical-examples-422964>)
2. Las opciones habituales son:
[juandiegolizarazo.blogspot](<http://juandiegolizarazo.blogspot.com/2018/04/tracepath-y-traceroute-linux.html>)

Opción	Ejemplo	Descripción resumida
<code>-n`</code>	<code>tracepath -n google.com`</code>	No resuelve nombres DNS; muestra solo IPs.
<code>-b`</code>	<code>tracepath -b google.com`</code>	Muestra tanto IPs como nombres de host.
<code>-I <MTU>`</code>	<code>tracepath -I 1400 google.com`</code>	Establece el tamaño inicial del paquete (MTU).
<code>-h <n>`</code>	<code>tracepath -h 30 google.com`</code>	Máximo número de saltos (hops).

Para ver todas las opciones, ejecutar:

```
tracepath --help
```

o

```
man tracepath
```

Ejemplos prácticos

- **1. Rastreo básico a un dominio**

```
tracert google.com
```

Muestra la ruta hacia Google incluyendo número de salto, IP, RTT y MTU en cada enlace.

[juandiegolizarazo.blogspot](<http://juandiegolizarazo.blogspot.com/2018/04/tracert-y-traceroute-linux.html>)

- **2. Sin resolución DNS (solo IPs)**

```
tracert -n google.com
```

Ahorra tiempo si hay problemas de DNS.

[baulderasec.wordpress](<https://baulderasec.wordpress.com/analisis-software/linux/8-configuracion-basica-de-redes/8-4-diagnosticar-conexiones-de-red/8-4-2-hacer-una-traza-de-una-ruta/>)

- **3. Con tamaño de paquete limitado (MTU)**

```
tracert -l 1400 google.com
```

Útil para detectar si algún enlace rompe la conexión por una MTU más baja.

[labex](<https://labex.io/es/tutoriales/linux-linux-tracert-command-with-practical-examples-422964>)

- **4. Limitar número máximo de saltos**

```
tracert -h 10 google.com
```

Detiene el rastreo a 10 saltos aunque no haya llegado al destino.

[juandiegolizarazo.blogspot](<http://juandiegolizarazo.blogspot.com/2018/04/tracert-y-traceroute-linux.html>)

Enlaces

- Material obtenido en parte del artículo escrito por S3L3N1TY para Hacker's Land.
- [¿Qué es el comando Tracert o Traceroute y cómo utilizarlo?](#)

From:
<https://euloxio.myds.me/dokuwiki/> - **Euloxio wiki**

Permanent link:
<https://euloxio.myds.me/dokuwiki/doku.php/doc:tec:net:cmd:tracert:inicio?rev=1775724411>

Last update: **2026/04/09 10:46**

