[Redes] Comandos: tracert/traceroute/tracepath

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` . Específica 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.
  - `traceroute` en Linux suele usar UDP por defecto, pero podemoss 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.

Función principal

1. Muestra la secuencia de enrutadores (saltos) por los que pasan los paquetes hacia un destino (IP o nombre de dominio).
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.

Sintaxis básica

- En la mayoría de distribuciones de Linux, la sintaxis es muy simple:

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

  donde:

  1. `<destino>` es una IP (`8.8.8.8`) o un nombre de host (`google.com`).
  2. Las opciones habituales son:

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

Para ver todas las opciones, ejecutar:

```
tracepath --help
```

o

```
man tracepath
```

Ejemplos prácticos

- 1. **Rastreo básico a un dominio**

  ```
  tracepath google.com
  ```

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

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

```
tracepath -n google.com
```

Ahorra tiempo si hay problemas de DNS.

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

```
tracepath -l 1400 google.com
```

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

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

```
tracepath -h 10 google.com
```

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

Ejemplo típico explicado

Un ejemplo típico de salida de `tracepath` en Linux es algo como:

```
tracepath google.com
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              1.234ms
 1:  _gateway                                              0.987ms
 2:  10.236.32.1                                           3.456ms
 3:  no reply
 4:  172.16.100.1                                          5.678ms
     pmtu 1492
 ...
10:  142.250.179.174                                      24.789ms
     reached
     Resume: pmtu 1492 hops 10 back 10
```

Vamos a desglosar campo por campo para poder interpretarlo en nuestro entorno de red.

1. Línea inicial y PMTU local

```
 1?: [LOCALHOST]                      pmtu 1500
```

1. `1?:` → primer “salto” (hop 1), marcado como `?` porque aún no se ha visto respuesta ICMP válida.
2. `[LOCALHOST]` → indica que el primer salto es desde tu propia máquina; no se ha identificado todavía el primer enrutador.
3. `pmtu 1500` → Path MTU detectado hasta este punto es 1500 bytes (MTU típico de Ethernet).

- **Utilidad**: Nos dice el tamaño de paquete máximo que podemos usar en nuestro segmento local sin fragmentación; si vemos valores bajos (1492, 1480, etc.) suele indicar PPPoE, VLAN, o túneles.

2. Salto con IP y nombre (`_gateway`)

```
 1:  _gateway                                              1.352ms
 1:  _gateway                                              1.426ms
```

1. `1:` → número de salto.
2. `_gateway` → nombre del primer enrutador (normalmente tu router de acceso a la LAN; puede ser `192.168.1.1`, `10.0.0.1`, etc.).
3. `1.352ms`, `1.426ms` → RTT (round-trip time) para ese salto; `tracepath` suele hacer varias pruebas por hop.

- Interpretación en nuestra red:

1. Un salto `1:` con tiempos bajos (< 2 ms) indica que nuestra LAN está sana.
2. Si son muy altos o aparecen `no reply` en el primer salto, revisar switch, router o incluso ARP.

3. Saltos intermedios

```
2:  10.236.32.1                                           3.764ms
3:  no reply
4:  172.16.100.1                                          5.678ms
```

1. `2:`, `3:`, `4:` → segundos, terceros, cuartos enrutadores en la ruta.
2. `10.236.32.1 / 172.160.100.1` → IP del enrutador o dispositivo de la red.
3. `3: no reply` → el enrutador en el salto 3 no responde a ICMP (firewall bloqueando, política de seguridad, etc.).
4. `5.678ms` → RTT acumulado hasta ese nodo.

- Para un entorno como el nuestro:
    1. Si saltos intermedios aparecen como `no reply` pero el destino final se alcanza, suele ser normal (ISP/proveedor censura ICMP).
    2. Si desde cierto salto todo es `no reply` y luego se corta, probablemente ahí estamos tocando un enclave donde no hay retorno ICMP o hay micro-cortes.

4. Cambio de MTU (`pmtu`)

```
     pmtu 1492
```

1. Línea extra que aparece cuando el camino detecta un cambio de MTU.

- En nuestro entorno:
    1. Si vemos `pmtu 1492` en un salto, es muy típico de PPPoE (MTU 1500 − 8 bytes de cabecera PPPoE).
    2. Si más adelante vemos `pmtu` menor aún (1480, 1400, etc.), puede haber túneles GRE/IPsec, VPNs o enlaces más restringidos.

5. Último salto y “reached”

```
10:  142.250.179.174                                      24.789ms
     reached
```

1. `10:` → décimo salto en la ruta.
2. `142.250.179.174` → IP del host destino (por ejemplo, servidor de Google).
3. `24.789ms` → RTT total hasta el destino.
4. `reached` → indica que el destino responde correctamente.

- En un uso diario:
    1. Si vemos `reached` pero con RTT muy alto (cientos de ms), buscar enlace congestionado o ruta larga.
    2. Si no vemos `reached`, sino que al final se corta con `no reply` o `TOO LONG`, el destino o algún tramo intermedio no está respondiendo o hay bloqueo de tráfico.

6. Línea de resumen final

```
     Resume: pmtu 1492 hops 10 back 10
```

1. `pmtu 1492` → MTU efectivo del camino completo.
2. `hops 10` → número total de saltos ida hasta el destino.
3. `back 10` → número de saltos de vuelta (en general igual a `hops` en rutas simétricas).

- Para nosotros como administrador:
    1. Si `hops` es muy alto en una red interna, revisar diseño de ruteo (circuitos, políticas BGP/IGP, etc.).
    2. Si vemos `pmtu` muy bajo en nuestra red interna, investigar si hay túneles innecesarios, PPPoE mal configurado o VLANs que estén “truncando” MTU.

Cómo utilizarlo en nuestro entorno

- Usa `tracepath` para:
    1. Verificar MTU real en la ruta hacia un servidor remoto (por ejemplo, VPN, túnel GRE, o ISP).
    2. Asegurarnos de que no hay cuellos de botella de MTU que provoquen fragmentación o drops silenciosos.
    3. Combinarlo con `ping -M do -s 1472` (Ethernet) para confirmar el tamaño máximo de payload sin fragmentar.

Enlaces

- Material obtenido en parte del artículo escrito por S3L3N1TY para Hacker's Land.
- ¿Qué es el comando Tracert o Traceroute y cómo utilizarlo?
- Tracert o Traceroute: qué es, cómo funciona o cómo se utiliza
- Comando tracepath de Linux con Ejemplos Prácticos
- http://juandiegolizarazo.blogspot.com/2018/04/tracepath-y-traceroute-linux.html
- 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/