🌐 Examen 109 - Fundamentos de Redes

109.3 Resolución de problemas de red básica

Teoría

Cuando experimentas problemas de red (no puedes acceder a un sitio web, un servicio remoto no responde, no puedes hacer SSH a otro servidor), necesitas una metodología para identificar la causa. La resolución de problemas de red a menudo implica verificar la conectividad y la configuración capa por capa.

Metodología de Resolución de Problemas Básica:

1. **Verificar la Conexión Física/Capa de Enlace:** ¿Está el cable de red conectado? ¿Está la interfaz habilitada? ¿Hay errores en la interfaz? (Luces en el switch/router/tarjeta de red).
2. **Verificar el Estado y la Configuración de la Interfaz:** ¿Tiene la interfaz una dirección IP y una máscara de subred correctas? ¿Está la interfaz activa?
3. **Verificar la Puerta de Enlace (Gateway):** ¿Está configurada la puerta de enlace predeterminada correcta? ¿Puedes alcanzar la puerta de enlace?
4. **Verificar la Resolución de Nombres (DNS):** ¿Puedes resolver nombres de host a direcciones IP? (Si puedes hacer `ping` a una IP pero no a un nombre, el problema puede ser DNS).
5. **Verificar el Firewall:** ¿Hay un firewall local (en tu máquina) o en la red que esté bloqueando el tráfico?
6. **Verificar el Host/Servicio Remoto:** ¿Está el host remoto encendido? ¿Está el servicio remoto (ej: servidor web, SSH) corriendo y escuchando en el puerto correcto?
7. **Rastrear la Ruta:** ¿Hay algún punto de fallo o alta latencia en el camino entre tu máquina y el destino?

Herramientas Clave de Resolución de Problemas:

1. **`ip` y `ifconfig`:** (Revisitado de 109.1/109.2)
   - Uso: Verificar el estado de la interfaz, dirección IP, máscara, estado (UP/DOWN), errores (`RX errors`, `TX errors`).
   - `ip addr show <interfaz>` o `ifconfig <interfaz>`
   - `ip link show <interfaz>`: Muestra el estado de la Capa de Enlace.
2. **`ip route` y `netstat -r`:** (Revisitado de 109.1/109.2)
   - Uso: Verificar la tabla de enrutamiento y la puerta de enlace predeterminada.
   - `ip route show` o `netstat -r`
3. **`ping`:**
   - Uso: Probar si un host es alcanzable a nivel IP (Capa de Internet) enviando paquetes ICMP "echo request" y esperando "echo reply". Mide el tiempo de ida y vuelta (latencia).

- `ping <direccion_ip_o_hostname>`
- `ping -c <numero>`: Envía un número específico de paquetes.
- `ping -I <interfaz>`: Especifica la interfaz de origen.
- Salida: Muestra el tiempo de respuesta, el TTL (Time To Live - indica cuántos saltos le quedan al paquete), y estadísticas de pérdida de paquetes.

4. **`traceroute`:**
   - Uso: Rastrea la ruta que toman los paquetes IP desde tu host hasta un destino, mostrando cada "salto" (router) en el camino. Útil para identificar dónde falla la conexión o dónde hay latencia.
   - `traceroute <direccion_ip_o_hostname>`
   - **Nota:** Utiliza paquetes UDP o ICMP con TTL incremental. Algunos firewalls pueden bloquear estos paquetes.
5. **`mtr` (My Traceroute):**
   - Uso: Combina la funcionalidad de `ping` y `traceroute` en una sola herramienta interactiva. Envía paquetes continuamente y muestra estadísticas en tiempo real para cada salto (pérdida de paquetes, latencia). A menudo es más útil que `traceroute` para diagnosticar problemas intermitentes o de latencia.
   - `mtr <direccion_ip_o_hostname>`
   - Presiona `q` para salir.
6. **`host`, `dig`, `nslookup`:** (Introducidos aquí, cubiertos en 109.4)
   - Uso: Probar la resolución de nombres DNS.
   - `host <hostname>`: Resuelve nombre a IP y viceversa.
   - `dig <hostname>`: Herramienta más flexible y detallada para consultas DNS.
   - `nslookup <hostname>`: Herramienta más antigua para consultas DNS (en desuso frente a `dig` y `host`).
7. **`ss` y `netstat`:** (Revisitado de 109.1/109.2)
   - Uso: Verificar si un servicio local está a la escucha en un puerto (`ss -tulnp`) o si hay conexiones activas a/desde un puerto (`ss -antup`). Útil para confirmar que el servicio está corriendo y que el firewall local no lo bloquea (si el puerto está LISTEN).
8. **`nc` (netcat):**
   - Uso: Una herramienta versátil ("la navaja suiza de redes") para leer y escribir datos a través de conexiones de red. Muy útil para probar si un puerto remoto está abierto y aceptando conexiones.
   - `nc -zv <direccion_ip_o_hostname> <puerto>`: Escanea si un puerto está abierto (`-z` para cero I/O, `-v` verbose).
   - `nc <direccion_ip_o_hostname> <puerto>`: Intenta establecer una conexión TCP. Si tiene éxito, puedes escribir en la terminal y los datos se enviarán al puerto remoto (útil para interactuar con servicios simples como HTTP o probar si aceptan conexiones).

    - `nc -l -p <puerto>`: Pone `nc` a la escucha en un puerto local (`-l` listen, `-p` port).
9. **`telnet`:**
    - Uso: Históricamente para acceder a terminales remotos (inseguro). Hoy en día se usa a menudo simplemente para probar si un puerto TCP está abierto, ya que intenta conectarse y a menudo muestra un banner si el servicio responde.
    - `telnet <direccion_ip_o_hostname> <puerto>`
    - Si la conexión se establece, verás una pantalla en blanco o un banner. Presiona `Ctrl+]` y luego `quit` para salir. Si la conexión falla inmediatamente, el puerto está cerrado o inaccesible.

Problemas Comunes de Red y Diagnóstico:

- **`ping` falla a todo excepto `127.0.0.1`:** Problema en la interfaz local o configuración IP/máscara.
- **`ping` falla a hosts en la red local pero funciona a `127.0.0.1`:** Problema de conectividad en la red local (cable, switch) o firewall local bloqueando ICMP.
- **`ping` falla a hosts fuera de la red local pero funciona a hosts locales:** Problema con la puerta de enlace predeterminada o enrutamiento, o firewall en el router/red remota.
- **`ping` a IP funciona, pero `ping` a hostname falla:** Problema de resolución DNS.
- **Puedes `ping` a un servidor remoto, pero no puedes acceder al servicio (ej: SSH, web):** El host remoto está encendido, pero el servicio no está corriendo, está escuchando en el puerto incorrecto, o un firewall (local o remoto) está bloqueando el puerto. Usa `nc` o `telnet` para probar el puerto.

Paquetes Comunes para Herramientas:

- `ping`: Suele venir en el paquete `iputils-ping` (Debian/Ubuntu) o `iputils` (Red Hat/Fedora).
- `traceroute`: Paquete `traceroute`.
- `mtr`: Paquete `mtr`.
- `nc` (netcat): Paquete `netcat` (o variantes como `netcat-openbsd`, `nmap-ncat`).
- `telnet`: Paquete `telnet`.
- `host`, `dig`: Paquete `dnsutils` (Debian/Ubuntu) o `bind-utils` (Red Hat/Fedora).
- `ss`, `ip`: Paquete `iproute2` (estándar en ambos).
- `ifconfig`, `netstat`: Paquete `net-tools` (disponible en ambos, a menudo no instalado por defecto en nuevas VMs).