1 Network Troubleshooting Tools 🚨

These tools are essential for diagnosing and troubleshooting issues related to network connectivity, performance, and security.

- **Ping**

Ping is a basic network diagnostic tool that checks if a network device is reachable. It sends ICMP Echo Request packets to a specified host and waits for an ICMP Echo Reply. Ping also measures round-trip time (RTT) and packet loss, helping determine network health and latency.

Use Case: Testing if a server or device is online and reachable.

- **Traceroute**

Traceroute traces the route that packets take from your device to a destination host. It shows each hop along the path, helping identify where delays or packet losses occur. It uses ICMP Echo Requests (or UDP packets) to map the route.

Use Case: Diagnosing where delays or failures occur in a network path.

- **Netstat**

Netstat (Network Statistics) displays active network connections, routing tables, and network interface statistics. It helps you identify listening ports, active connections, and any network-related issues.

Use Case: Monitoring network connections and troubleshooting issues like open ports or unusual traffic.

- **ss**

ss (socket statistics) is similar to netstat but more efficient and provides detailed information on TCP, UDP, and UNIX socket connections. It can show connection states, listening sockets, and detailed stats on connection performance.

Use Case: Advanced socket statistics for performance and security analysis.

- **tcpdump**

Tcpdump is a command-line packet analyzer. It captures and displays the raw network traffic that passes through your network interface. It can filter traffic by protocols, ports, or IPs and is crucial for in-depth network troubleshooting and security analysis.

Use Case: Capturing and analyzing network traffic for security vulnerabilities or performance issues.

• **iftop**

Iftop shows real-time bandwidth usage for individual connections, helping you monitor which processes or IPs are consuming the most bandwidth. It's useful for troubleshooting congestion and optimizing network resources.

Use Case: Real-time monitoring of network bandwidth and diagnosing slow or heavy traffic.

• **nmap**

Nmap is a powerful network exploration tool used for discovering hosts and services on a network. It can be used for network scanning, security auditing, and vulnerability assessments. Nmap can identify open ports, services running on those ports, and the operating systems of remote devices.

Use Case: Security auditing, network mapping, and vulnerability scanning.

• **nslookup / dig**

Both nslookup and dig are DNS querying tools. **nslookup** (Name Server Lookup) is used to query DNS records and troubleshoot DNS-related issues. **dig** (Domain Information Groper) provides more detailed information about DNS queries, including query time and flags.

Use Case: Troubleshooting DNS issues, such as incorrect IP addresses or server misconfigurations.

---

2. DNS and DHCP Configuration

DNS (Domain Name System) and **DHCP (Dynamic Host Configuration Protocol)** are fundamental network services. DNS resolves domain names to IP addresses, and DHCP automatically assigns IP addresses to devices on a network.

• **DNS server setup (BIND or Unbound)**

DNS servers translate domain names (like example.com) into IP addresses. BIND (Berkeley Internet Name Domain) and Unbound are two popular DNS server software. BIND is widely used for authoritative DNS services, while Unbound is designed for recursive DNS queries.

Use Case: Setting up a local DNS server to resolve domain names internally or externally.

• Configuring DNS zones and records (A, MX, PTR, etc.)

DNS zones store information about domain names. Common DNS records include:

- **A** (Address) records: map domain names to IP addresses.
- **MX** (Mail Exchange) records: define mail servers for email routing.
- **PTR** (Pointer) records: reverse DNS lookup, mapping IPs to domain names.

Use Case: Configuring DNS for a domain, including email handling and reverse DNS.

• DHCP server setup (using dhcpd or dnsmasq)

A DHCP server automatically assigns IP addresses to devices on a network. **dhcpd** (DHCP Daemon) is the most common DHCP server software, while **dnsmasq** provides lightweight DHCP and DNS services. Configuring a DHCP server involves defining a range of IP addresses and setting lease times.

Use Case: Automatically assigning IP addresses and other network settings to devices.

• Configuring DHCP clients (dhclient, systemd-networkd)

DHCP clients like **dhclient** (on Linux) and **systemd-networkd** (on modern Linux distributions) are used by devices to request an IP address and other network configurations from a DHCP server.

Use Case: Ensuring devices can request and receive valid IP addresses.

• Static IP assignments via DHCP

DHCP can be configured to assign the same IP address to a device based on its MAC address. This is called a "static lease" and ensures that specific devices always receive the same IP.

Use Case: Assigning fixed IPs to servers or network devices that require consistent addressing.

• Troubleshooting DNS and DHCP issues

Common troubleshooting involves:

- Verifying server configurations and IP ranges.
- Using tools like nslookup, dig, or dhclient to check DNS and DHCP communication.
- Checking firewall settings, network routes, and logs.

Use Case: Resolving issues like IP conflicts, DNS resolution failures, or DHCP service disruptions.

---

3 Network File Systems

Network File Systems allow files to be accessed over a network, enabling file sharing between systems.

- **NFS (Network File System)**

NFS allows Linux and Unix systems to share files over a network. It enables a remote file system to appear as if it's part of the local file system.

Use Case: Sharing files between Linux systems in an enterprise environment.

- **Samba**

Samba is a software suite that enables file sharing between Linux and Windows systems. It implements the SMB (Server Message Block) protocol, allowing Linux systems to appear as Windows shares.

Use Case: Enabling file sharing between Linux and Windows systems.

- **SSHFS**

SSHFS allows mounting remote directories over an SSH connection. It is based on FUSE (Filesystem in Userspace), providing a secure and straightforward way to access remote files.

Use Case: Securely accessing files on remote systems over SSH.

- **iSCSI**

iSCSI allows block-level access to remote storage devices over a TCP/IP network. It is often used in storage area networks (SANs) to allow remote servers to access disk volumes.

Use Case: Connecting remote storage devices to servers for block-level access.

---

4 Routing and IP Forwarding

Routing determines the path that data takes across a network. IP forwarding allows a system to forward network packets from one interface to another.

- **Enabling IP forwarding**

IP forwarding is the process of forwarding network packets between network interfaces. On Linux, it is enabled with the command:

sysctl net.ipv4.ip_forward=1

This allows the system to act as a router, forwarding packets from one network to another.

Use Case: Configuring a Linux server to route packets between networks.

• Viewing routing tables

Routing tables define the routes that packets take to reach their destination. Use commands like route, ip route, or netstat -r to view and troubleshoot routing information.

Use Case: Checking which routes the system uses to reach different networks.

• Static Routing vs. Dynamic Routing

- **Static Routing**: Manually configured routes. These routes don't change unless explicitly modified.
- **Dynamic Routing**: Routes that are automatically updated based on network conditions, using protocols like RIP, OSPF, or BGP.

Use Case: Static routes are used for simple, predictable paths; dynamic routing is used in large networks for automatic path updates.

• Configuring a static route with ip route

To configure a static route, use the ip route command. For example:

ip route add 192.168.1.0/24 via 192.168.0.1

This command configures a static route to the 192.168.1.0 network through the 192.168.0.1 gateway.

Use Case: Directing traffic between specific networks through a defined gateway.

• Using ip rule for policy-based routing

Policy-based routing (PBR) uses rules to determine which route to take for a packet. ip rule allows you to set different routes for different types of traffic based on source address, protocol, etc.

Use Case: Routing traffic from specific IP ranges through different gateways for load balancing or security.

• Network Address Translation (NAT): SNAT, DNAT

NAT modifies the source or destination IP of a packet.

- **SNAT** (Source NAT) changes the source IP, typically used for outbound traffic from a local network to the internet.
- **DNAT** (Destination NAT) changes the destination IP, typically used to direct traffic from the internet to a local server.

Use Case: SNAT is used for internet access; DNAT is used for port forwarding.

• Configuring routing with iptables

iptables is a powerful firewall tool that can also be used to configure NAT and routing. You can use iptables rules to control how traffic is routed or blocked between interfaces.

Use Case: Setting up NAT, blocking unwanted traffic, or routing