Module 6- Linux server - Manage basic networking & Security

1. Use ifconfig or ip to view and configure network interfaces.
   ➔ To view and configure network interfaces on a computer, we can use two main commands: **`ifconfig`** and **`ip`**. These commands are commonly used in Linux systems to manage and troubleshoot network settings.

1. `ifconfig` command:

- **View network interfaces:**
    - Type `ifconfig` in the terminal to see a list of all network interfaces on our computer, such as Ethernet (eth0), Wi-Fi (wlan0), and others.
    - Example:
      `ifconfig`

Configure network interfaces:

- We can also use ifconfig to set IP addresses and other network settings (though ip is preferred for newer systems).
- Example (set a static IP):
  sudo ifconfig eth0 192.168.1.100 netmask 255.255.255.0

2. ip command (more modern and powerful):

- **View network interfaces:**
    - Type ip a to see all network interfaces and their settings.

Example:
ip a

- **Configure network interfaces:**
  The ip command is often used for more complex configurations. For instance, to change the IP address:
  sudo ip addr add 192.168.1.100/24 dev eth0
- **Bring an interface up or down:**
    - To enable or disable a network interface,we can use the following:

- Bring interface up (enable):
  sudo ip link set eth0 up
- Bring interface down (disable):
  sudo ip link set eth0 down

2.Use ping to test network connectivity

➔ The **ping** command is used to test if our computer can reach another device on the network, like a website, another computer, or a server.

How to use ping:

1. **Test if our computer can reach another device (e.g., Google):**
   Open a terminal and type:
   ping google.com
   - This sends small "packets" of data to **google.com** and waits for a response. If us see responses like "Reply from..." it means our computer can reach Google, and the network is working.

2. **Test if our computer can reach a specific IP address:**
   - If you know the IP address of another device, you can test it directly. For example:
     ping 192.168.1.1
   - This tests if our computer can reach the device with the IP address **192.168.1.1** (like our router).

3. **Stop the test:**

   - By default, **ping** keeps sending requests until we stop it. To stop it, press **Ctrl + C**.

What you'll see:

If the test is successful, we'll see lines like:
Reply from 8.8.8.8: bytes=32 time=14ms TTL=53

- This means the network is working, and the time is how long it took for the data to go there and back.
- If the test fails, we might see something like:
  Request Timed Out

- This means the computer couldn’t reach the target.

3. Understand basic firewall configuration using FIREWALL-CMD.

➔ **firewall-cmd** is a command-line tool used to manage the firewall in Linux, specifically for systems using **firewalld** (a firewall management tool). The firewall controls which connections are allowed or blocked on our system, helping protect it from unauthorized access.

Basic Concepts:

- **Firewall Zones:** These define different levels of trust for network interfaces (like a home network or public network). The firewall applies rules based on the zone of the network connection.

- **Services:** These are predefined rules for common applications, like HTTP (web traffic), SSH (remote login), etc.

- **Ports:** Specific network ports that services listen to for connections.

Basic Commands with firewall-cmd:

1. **Check the firewall status:**
   To see if the firewall is active:
   sudo firewall-cmd --state or systemctl status firewalld.service
- If it says "running," the firewall is active and working.

2. **View current rules:**

- To see the rules for the current active zone (like which services are allowed):
  sudo firewall-cmd --list-all
3. **Allow a service (e.g., SSH or HTTP):**

- To allow a service (e.g., SSH for remote login):
  sudo firewall-cmd --zone=public --add-service=ssh --permanent
    - --zone=public: Specifies which network zone to apply the rule to.

        - --add-service=ssh: Adds the SSH service to be allowed.

        - --permanent: Makes the change permanent (it survives a reboot).

4. **Reload the firewall to apply permanent changes:**

- After making permanent changes (like allowing SSH), reload the firewall:
  sudo firewall-cmd --reload
5. **Allow a specific port (e.g., port 8080 for a web server):**

- If we want to allow a specific port (e.g., port 8080):
  sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
6. **Remove a rule (e.g., block SSH):**

- To remove a service from the allowed list (e.g., block SSH):
  sudo firewall-cmd --zone=public --remove-service=ssh --permanent
7. **Check available zones:**

    - To see different zones and what they do (like trusted or home):
      sudo firewall-cmd --get-zones

4.Add ssh services in firewall

➔ To allow **SSH** (which is used for remote login) through our firewall, we need to add it as an allowed service using **firewall-cmd**.

Steps to add SSH service to the firewall:

1. **Allow SSH service temporarily:**

➔ If we want to allow SSH through the firewall right away (without making it permanent), can use this command:
sudo firewall-cmd --add-service=ssh

2. **Allow SSH service permanently:**

➔ To make this change permanent (so it stays even after a reboot), use the --permanent option:
sudo firewall-cmd --add-service=ssh --permanent

3. **Reload the firewall to apply permanent changes:**

➔ After making the change permanent, reload the firewall for the rule to take effect:
sudo firewall-cmd --reload

4. **Verify that SSH is allowed:**

➔ To check if SSH has been added to the allowed services, use:
sudo firewall-cmd --list-all

- we should see ssh listed under "services" in the output.