Linux Networking Documentation

---

1. Introduction to Networks:

A network serves as a means for devices to connect and share data. The two primary types of networks are:

- LAN (Local Area Network): For close-range connections.
- WAN (Wide Area Network): For long-distance connections like the internet.

Technologies:

- WLAN (Wi-Fi): Wireless connection.
- NIC (Network Interface Card): Hardware component.
- Ethernet: Wired connection.
- Asterisk: An open-source platform for real-time communications applications.
- Ping with Domain Name of Google: `ping www.google.com`

2. IP Addresses and DNS:

- IPv4: Internet Protocol Version 4 assigns unique numerical addresses to devices (e.g., 142.250.192.36).

  ```
  nslookup www.google.com
  ```

To Convert IP Address to One Octet:

- Example: Google's IP Address 142.250.192.36

  142*2^24 + 250*2^16 + 192*2^8 + 36*2^0 = 2398797860

Traceroute in CLI:

```
traceroute google.com
```

Continuous Ping for Monitoring Connectivity:

```
ping -t [hostname]
```

- DNS (Domain Name System): Facilitates user interaction with devices on the Internet without remembering numeric addresses.

3. Minimum Requirements for Network Connection:

- Physical medium (wired or wireless).
- Booted up operating system on both sides.
- NIC or Ethernet card and drivers.
- Assigned IP address to the network card.

4. Networking in Linux:

- Involves setting up and managing network connections, assigning IP addresses, etc.
- 16 versions of IP addresses available, with IPv4 being the most widely used.
- Network name decided by netmask.

5. Public and Private IP Addresses:

- Public IPs are paid and used for internet connectivity.
- Private IPs are freely available for local connectivity (LAN).
- NAT (Network Address Translation) required for connecting private and public addresses.

6. Networking Devices:

- Routers used in OSI layer 3, switches in layer 2.
- L3 switches work as both switches and routers.

Linux Server Configuration and User Management

1. Introduction to Servers

Servers play a crucial role in providing various services such as web hosting, email, databases, and remote login. Remote login allows users to access a system virtually, enabling them to work on a different machine.

2. Remote Login Protocols

There are several protocols for remote login, including RSH, Telnet, and SSH. SSH (Secure Shell) is the most secure protocol and is commonly used for secure remote access.

3. SSH Server Configuration

3.1 Install OpenSSH Server

To install the OpenSSH server, use the following commands:

```
sudo yum install openssh-server  # Install the OpenSSH server package

sudo rpm -q openssh-server      # Check if the package is available
```

3.2 Configure SSH Server

Edit the SSH server configuration file using a text editor such as Vim:

Set configurations such as port number, security options, etc.

3.3 Start SSH Services

Start the SSH services with:

```
sudo systemctl start sshd  # Start the SSH services

sudo systemctl status sshd  # Check the status of the SSH server
```

4. Client Configuration

4.1 Install SSH Client Software

Install an SSH client like Putty on the Windows system:

Download PuTTY from putty.org.
Install PuTTY by following the installation instructions.

4.2 Connecting to SSH Server

Open PuTTY and enter the server IP address, port, and select the SSH protocol for a secure connection.

5. Changing SSH Port

To change the SSH port, modify the `sshd_config` file:

```
sudo vim /etc/ssh/sshd_config
```

Change the "Port" setting to your desired port number, then restart the SSH services.

6. User Management

6.1 Root User and General Users

Linux has two main user types: root (superuser) and general users with limited permissions. Use the `su` command to switch between users.

6.2 Restricting Root Login

To restrict root login, modify the `sshd_config` file:

```
sudo vim /etc/ssh/sshd_config
```

Set `PermitRootLogin prohibit-password` and `PasswordAuthentication no`.

6.3 Banners for Login Messages

Create a banner file:

```
sudo vim /etc/mybanner
```

Add a welcome message. Uncomment "Banner none" in `sshd_config` and restart the SSH services.

6.4 Running Single Commands at Login

Execute a command at login:

```
ssh -l root ip-address date
```

This logs in, runs the command, and exits.

7. Key-Based Authentication

7.1 Create Key Pair

Generate a public-private key pair using:

ssh-keygen.exe:-This command will generate a new SSH key pair by default in the `~/.ssh/` directory. The private key will be named `id_rsa`, and the public key will be named `id_rsa.pub`.

If you are using a Windows system and have installed the OpenSSH client, you can also run this command in the Windows Command Prompt or PowerShell to generate SSH keys.

Remember to protect your private key and only share your public key when necessary. The public key is typically added to the `~/.ssh/authorized_keys` file on the server you want to connect to.

7.2 Enable Public Key Authentication

Modify `sshd_config`:

```
sudo vim /etc/ssh/sshd_config
```

Set `PubKeyAuthentication yes` and `AuthorizedKeysFile /path/tofile`.

7.3 Copying Public Key to Server

Use `ssh-copy-id` to copy the public key:

```
ssh-copy-id [username]@[ip_address]
```

After copying, log in without a password:

```
ssh [username]@[ip_address]
```

Keys are stored in the `.ssh/` folder on the client side. Check keys using:

```
cd .ssh/

ls
```

On the server, authorized keys are stored in `.ssh/`:

```
cd .ssh/

ls

cat authorized_keys
```