Domain Name & DNS -

1. Domain Name

- **Definition**: Human-readable address of a website → replaces hard-to-remember IP addresses.
- Example: www.google.com instead of 142.250.182.46.
- Structure (from right to left):
    - **Root (.)** → Hidden dot at end of domain.
    - **TLD (Top-Level Domain)** → .com, .org, .in.
    - **Second-Level Domain** → google, wikipedia.
    - **Subdomain** → mail.google.com, blog.example.com.

👉 Acts like a nickname for IP addresses.

2. DNS (Domain Name System)

- **Definition**: A distributed system that **translates domain names into IP addresses**.
- Without DNS → you must type raw IP addresses.

3. How DNS Works (Resolution Process)

When you type www.example.com in a browser:

1. **Browser Cache** → Checks if it already knows IP.
2. **OS Cache (Local DNS Resolver)** → Checks system DNS cache.
3. **Recursive Resolver (ISP's DNS)** → Contacts DNS hierarchy if not cached.
4. **Root Server** → Points to correct TLD server (.com, .org).
5. **TLD Server** → Points to authoritative name server of domain.
6. **Authoritative Name Server** → Gives final IP (e.g., 93.184.216.34).
7. Browser connects to server using that IP.

4. DNS Record Types

- **A Record** → Maps domain → IPv4 address.
- **AAAA Record** → Maps domain → IPv6 address.
- **CNAME (Canonical Name)** → Alias to another domain.
- **MX (Mail Exchange)** → For emails.
- **NS (Name Server)** → Authoritative servers for domain.
- **TXT** → Misc info (SPF, DKIM for email security).

5. Ethical Hacking Relevance

- **Reconnaissance phase** → DNS is goldmine for hackers.
- Attacks include:

  - **DNS Spoofing / Cache Poisoning** → Redirect user to malicious site.
  - **DNS Hijacking** → Modify DNS settings to control traffic.
  - **Subdomain Enumeration** → Discover hidden services (admin panels, APIs).
  - **Zone Transfer Attack** → Misconfigured DNS server leaks entire domain info.
- Tools: **dig, nslookup, dnsenum, Fierce, Sublist3r**.

📝 Short Notes: Domain Name & DNS

Domain Name

- Human-friendly website name.
- Parts: Root → TLD → SLD → Subdomain.
- Example: blog.example.com.

DNS

- System that converts **domain → IP**.
- Resolution steps: Cache → Recursive → Root → TLD → Authoritative.
- Records: A, AAAA, CNAME, MX, NS, TXT.
- Hacking use: Recon, spoofing, hijacking, subdomain discovery.

Domain Name -

1. Definition

- A **Domain Name** is a **human-readable address** used to access websites on the internet.
- It acts as a **nickname** for IP addresses (which are hard to remember).
- Example:
  - Domain Name → www.google.com
  - IP Address → 142.250.182.46

2. Structure of Domain Name

Domain names are hierarchical (read from right to left):

1. **Root (.)** → Invisible dot at the end of every domain.
2. **TLD (Top-Level Domain)** → .com, .org, .net, .in.
3. **Second-Level Domain (SLD)** → The main name (e.g., google, wikipedia).
4. **Subdomain** → Part before SLD (e.g., mail.google.com, blog.example.com).

👉 Example Breakdown:

- mail.google.com
  - Subdomain = mail
  - Second-Level Domain = google
  - Top-Level Domain = .com

3. Types of Domain Names

- **Generic TLDs (gTLDs)** → .com, .org, .net.
- **Country-Code TLDs (ccTLDs)** → .in (India), .uk (UK).
- **Sponsored TLDs** → .gov, .edu, .mil.
- **Subdomains** → Custom extensions like shop.example.com.

4. Importance of Domain Names

- Easy for humans to remember (vs numeric IPs).
- Brand identity (e.g., amazon.com).
- Used in email addresses.
- Can host multiple services using subdomains.

5. Ethical Hacking Relevance

- Domain names are first step in **reconnaissance**.
- Hackers analyze:
  - Registered domain details (WHOIS lookup).
  - Subdomains for hidden services (admin.example.com).
  - Expired domains (domain hijacking).
- Tools: **whois, dig, nslookup, Sublist3r, Amass**.

📝 Short Notes: Domain Name

- **Definition** → Human-readable website address.
- **Structure** → Root → TLD → SLD → Subdomain.
- **Types** → gTLD, ccTLD, sponsored TLD, subdomain.
- **Importance** → Easy access, branding, emails, services.
- **Hacking use** → WHOIS lookup, subdomain enumeration, domain hijacking.

DNS (Domain Name System) -

1. Definition

- **DNS (Domain Name System)** is like the **phonebook of the internet**.

- It translates **domain names → IP addresses** so that computers can communicate.
- Without DNS, we'd have to remember IPs like 142.250.182.46 instead of www.google.com.

2. How DNS Works (Resolution Process)

When you type www.example.com in a browser:

1. **Browser Cache** → Checks if the IP is already stored.
2. **OS Resolver Cache** → Checks system-level cache.
3. **DNS Resolver (ISP's server)** → If not cached, it queries further.
4. **Root Server** → Directs query to correct TLD server (.com, .org, .in).
5. **TLD Server** → Points to authoritative name server of the domain.
6. **Authoritative Name Server** → Returns the final IP (e.g., 93.184.216.34).
7. Browser → Connects to that IP and loads website.

3. DNS Record Types

- **A Record** → Domain → IPv4 address.
- **AAAA Record** → Domain → IPv6 address.
- **CNAME** → Alias (maps one domain to another).
- **MX** → Mail servers for email.
- **NS** → Points to authoritative name servers.
- **TXT** → Extra info (used for SPF, DKIM for email security).

4. Importance of DNS

- Makes internet **user-friendly**.
- Enables **scalability** (multiple domains, subdomains).
- Used in **email, websites, cloud, apps**.

5. Ethical Hacking Relevance

- DNS is a **reconnaissance goldmine**.
- Attacks include:
  - **DNS Spoofing / Cache Poisoning** → Fake IP mapping to redirect users.
  - **DNS Hijacking** → Changing DNS records to control traffic.
  - **Zone Transfer Attack** → Misconfigured servers leak entire domain data.
  - **Subdomain Enumeration** → Finding hidden services (e.g., admin.example.com).
- Tools: **nslookup, dig, dnsenum, Fierce, Sublist3r, Amass**.

📝 Short Notes: DNS

- **Definition** → Phonebook of internet (maps domain → IP).
- **How it works** → Cache → Resolver → Root → TLD → Authoritative → IP.
- **Records** → A, AAAA, CNAME, MX, NS, TXT.
- **Hacking relevance** → Spoofing, hijacking, subdomain discovery, zone transfer.

DNS Records and Their Uses -

1. A Record (Address Record)

- Maps a **domain name → IPv4 address**.
- Example: example.com → 93.184.216.34
- **Use** → Main record for browsing websites.
- **Hacking use** → Reveals server IP (target for scanning/attacks).

2. AAAA Record (IPv6 Address Record)

- Maps a **domain name → IPv6 address**.
- Example: example.com → 2606:2800:220:1:248:1893:25c8:1946
- **Use** → Supports modern IPv6 networks.

3. CNAME (Canonical Name Record)

- Creates an **alias** of a domain.
- Example: www.example.com → example.com
- **Use** → Multiple services point to the same domain.
- **Hacking use** → Sometimes leaks third-party services (like AWS, Azure).

4. MX Record (Mail Exchange Record)

- Defines **mail servers** responsible for a domain.
- Example: example.com → mail.example.com
- **Use** → Email delivery.
- **Hacking use** → Attackers target mail servers for phishing/spam.

5. NS Record (Name Server Record)

- Points to the **authoritative name servers** for a domain.
- Example: example.com → ns1.hosting.com, ns2.hosting.com
- **Use** → Delegates DNS responsibility.
- **Hacking use** → Misconfigured NS can allow **zone transfer**.

6. TXT Record (Text Record)

- Holds arbitrary text info.
- Examples:
  - **SPF** → Email sender validation.

    - **DKIM** → Email authentication.
    - **Verification** → Google site verification.
- **Use** → Security + verification.
- **Hacking use** → TXT records may expose sensitive details (emails, configs).

7. PTR Record (Pointer Record)

- **Reverse DNS lookup** (IP → domain).
- Example: 93.184.216.34 → example.com
- **Use** → Email spam filtering, server validation.
- **Hacking use** → Helps attackers find domains hosted on an IP.

8. SOA Record (Start of Authority)

- Stores **domain admin details, refresh time, version info**.
- Example: Primary server, admin email, TTL values.
- **Use** → Controls DNS zone settings.
- **Hacking use** → Reveals admin email & update cycles.

9. SRV Record (Service Record)

- Defines **location of specific services** (host + port).
- Example: _sip._tcp.example.com → server1.example.com:5060
- **Use** → VOIP, chat servers, MS Exchange, etc.
- **Hacking use** → Can reveal hidden services.

10. Other Rare Records

- **CAA** → Defines which certificate authorities can issue SSL certs.
- **NAPTR** → Used in telephony (VoIP).
- **DNSKEY** → Used in DNSSEC for security.

📝 Short Notes: DNS Records

- **A** → Domain → IPv4 (main website IP).
- **AAAA** → Domain → IPv6.
- **CNAME** → Alias of another domain.
- **MX** → Mail server details.
- **NS** → Authoritative name servers.
- **TXT** → Extra info (SPF, DKIM, verification).
- **PTR** → Reverse DNS (IP → Domain).
- **SOA** → Domain authority & admin details.
- **SRV** → Defines services + ports.
- **CAA** → Restricts SSL certificate issuers.

Zone File -

1. Definition

- A **Zone File** is a **text file** on a **DNS server** that contains all the DNS records for a particular domain (or DNS zone).
- It tells the DNS server how to resolve requests for that domain.
- Stored on **authoritative name servers**.

2. Structure of a Zone File

- Written in **BIND format** (Berkeley Internet Name Domain).
- Contains different **resource records (RRs)**:
  - **SOA (Start of Authority)** – Information about the zone (admin, refresh, expiry).
  - **NS (Name Server)** – Authoritative DNS servers.
  - **A / AAAA** – Maps domain → IP (IPv4/IPv6).
  - **MX** – Mail servers.
  - **CNAME** – Aliases.
  - **TXT** – Text records (SPF, DKIM, etc.).

🔹 Example Zone File (for example.com)

```
$TTL 86400

@   IN  SOA ns1.example.com. admin.example.com. (

    2025081601 ; Serial

    3600      ; Refresh

    1800      ; Retry

    1209600   ; Expire

    86400 )   ; Minimum TTL


@     IN  NS     ns1.example.com.

@     IN  NS     ns2.example.com.

@     IN  A     93.184.216.34

www    IN  CNAME   example.com.
```

```
mail    IN  MX 10   mail.example.com.
```

👉 Here you see:

- **SOA** → Zone authority info.
- **NS** → Authoritative name servers.
- **A** → Main website IP.
- **CNAME** → Alias for www.
- **MX** → Mail server.

3. Importance

- Zone file = **blueprint of a domain's DNS setup**.
- Without it, DNS resolution won't work.

4. Ethical Hacking Relevance

- Attackers try **Zone Transfer Attack** to steal the full zone file.
- If misconfigured, the DNS server may allow unauthorized transfers, leaking:
  - All subdomains
  - Mail servers
  - Internal networks
  - Admin emails (from SOA record)
- Tools: dig axfr example.com @ns1.example.com

📝 Short Notes: Zone File

- **Definition** → Text file on DNS server with domain's DNS records.
- **Contains** → SOA, NS, A, AAAA, MX, CNAME, TXT, etc.
- **Format** → BIND style (resource records).
- **Use** → Guides DNS resolution for a domain.
- **Hacking relevance** → Zone transfer attack leaks full domain info.

OSI vs TCP/IP Model -

1. OSI Model (Open Systems Interconnection)

- Conceptual model by **ISO (International Organization for Standardization)**.
- Divides networking into **7 layers**.
- Each layer has specific functions, helps in standardization.

Layers of OSI (7 → 1)

1. **Application** → User interaction (HTTP, FTP, DNS).
2. **Presentation** → Data format, encryption, compression (SSL/TLS, JPEG).
3. **Session** → Manages sessions, connections (NetBIOS, RPC).
4. **Transport** → Reliable delivery (TCP/UDP, port numbers).
5. **Network** → Logical addressing, routing (IP, ICMP).
6. **Data Link** → Physical addressing (MAC, Ethernet, ARP).
7. **Physical** → Hardware signals (Cables, NIC, Wi-Fi).

2. TCP/IP Model

- Practical model used in the internet.
- Has **4 layers** (sometimes shown as 5).
- Developed by **DARPA (Defense Advanced Research Projects Agency)**.

Layers of TCP/IP (4 → 1)

1. **Application** → All higher-level functions (HTTP, FTP, DNS, SMTP).
2. **Transport** → Process-to-process delivery (TCP/UDP).
3. **Internet** → Logical addressing & routing (IP, ICMP).
4. **Network Access / Link** → Physical + Data Link (Ethernet, Wi-Fi, ARP).

3. Key Differences: OSI vs TCP/IP

Feature	OSI Model	TCP/IP Model
Developed by	ISO	DARPA
Layers	7 layers	4 layers
Concept	Theoretical, for standardization	Practical, real-world
Transport Layer	Connection-oriented & connectionless	Supports both (TCP = reliable, UDP = fast)
Application Layer	Separate layers (App, Presentation, Session)	Single Application Layer
Usage	Used for teaching, reference	Used in real networking
Protocol Dependency	Protocol-independent	Protocol-specific (TCP, IP)
Example Protocols	HTTP, FTP, TCP, IP, ARP, etc.	HTTP, FTP, TCP, UDP, IP, etc.

4. Ethical Hacking Relevance

- Hackers must understand **which layer is being attacked**.
  - **Physical Layer** → Wiretapping, jamming Wi-Fi.
  - **Data Link Layer** → MAC spoofing, ARP poisoning.
  - **Network Layer** → IP spoofing, ICMP flooding.
  - **Transport Layer** → TCP SYN flood, UDP flood.
  - **Application Layer** → SQL injection, XSS, CSRF.
- OSI helps in **theory**, TCP/IP helps in **real-world attacks & defense**.

📝 Short Notes: OSI vs TCP/IP

- **OSI Model** → 7 layers, theoretical, good for learning.
- **TCP/IP Model** → 4 layers, practical, real-world internet use.
- **Main Difference** → OSI separates **App/Presentation/Session**, TCP/IP combines them.
- **Ethical Hacking** → Different attacks target different layers (ARP poisoning = Data Link, SQL injection = Application).

OSI Model -

1. Definition

- **OSI (Open Systems Interconnection) Model** is a **theoretical framework** created by **ISO (International Organization for Standardization)**.
- It defines **how computers communicate over a network** by dividing communication into **7 layers**.
- Each layer has its own functions and interacts only with the layer above and below.

2. Purpose of OSI Model

- Standardizes networking for different devices and systems.
- Helps understand **data flow** in a network.
- Provides a reference for **troubleshooting, designing, and securing** networks.

3. 7 Layers of OSI Model (Top → Bottom)

1. **Application Layer**
   - Closest to users.
   - Provides network services to applications.
   - Examples: **HTTP, FTP, DNS, SMTP**.
2. **Presentation Layer**
   - Data **formatting, encryption, compression**.
   - Ensures data is understandable.
   - Examples: **SSL/TLS, JPEG, GIF**.
3. **Session Layer**
   - Manages **sessions (connections)** between devices.
   - Establish, maintain, terminate communication.
   - Examples: **NetBIOS, RPC**.
4. **Transport Layer**
   - Ensures **reliable data delivery** (error checking, sequencing).
   - Provides **ports** for communication.
   - Protocols: **TCP, UDP**.
5. **Network Layer**
   - Logical addressing & routing.
   - Decides path for data.
   - Protocols: **IP, ICMP, OSPF**.
6. **Data Link Layer**
   - Physical addressing (**MAC address**).
   - Error detection in frames.
   - Protocols: **Ethernet, ARP, PPP**.
7. **Physical Layer**

    - Deals with **hardware signals**.
    - Media: **Cables, Wi-Fi, Hubs, NICs**.

4. How Data Travels in OSI (Encapsulation & Decapsulation)

- Sender: Data → App → Presentation → Session → Transport → Network → Data Link → Physical → Transmission.
- Receiver: Reverse process.
- Each layer **adds/removes headers** to data.

5. Ethical Hacking Relevance

- Attackers target different OSI layers:
    - **Layer 1 (Physical):** Cutting cables, jamming Wi-Fi.
    - **Layer 2 (Data Link):** ARP spoofing, MAC flooding.
    - **Layer 3 (Network):** IP spoofing, ICMP flooding.
    - **Layer 4 (Transport):** SYN flood, UDP flood.
    - **Layer 7 (Application):** SQL Injection, XSS, CSRF.

📝 Short Notes: OSI Model

- **Definition** → Standard 7-layer model by ISO for network communication.
- **Layers (7 → 1):** Application, Presentation, Session, Transport, Network, Data Link, Physical.
- **Purpose** → Standardization, troubleshooting, security.
- **Ethical Hacking** → Attacks exist on every layer (Layer 2 = ARP spoofing, Layer 7 = SQL Injection).

How OSI Model Works -

1. Concept of Working

- The OSI model is a **conceptual framework**, meaning it doesn't physically exist but explains how communication happens.
- Data passes **down the 7 layers** on the sender's side, and **up the 7 layers** on the receiver's side.
- Each layer adds or removes information (called **headers**) to make communication possible.
- This process is known as **Encapsulation (sender)** and **Decapsulation (receiver)**.

2. Example of Data Transfer (Sending an Email)

🔹 **Sender (Encapsulation)**

1. **Application (Layer 7):** User writes email → Protocol used = SMTP.
2. **Presentation (Layer 6):** Email content is encrypted (TLS) & converted into a standard format.
3. **Session (Layer 5):** A communication session is established with the mail server.
4. **Transport (Layer 4):** TCP adds **port numbers** (e.g., Port 25 for SMTP).
5. **Network (Layer 3):** IP address of sender & receiver is attached.
6. **Data Link (Layer 2):** MAC address of devices is added for local delivery.
7. **Physical (Layer 1):** Data is sent as **bits (0s & 1s)** over cables/wireless.

🔹 **Receiver (Decapsulation)**

1. Bits arrive at Physical Layer → move upward.
2. Data Link verifies MAC → passes frame.
3. Network verifies IP address → forwards packet.
4. Transport checks port number → passes to SMTP.
5. Session re-establishes connection.
6. Presentation decrypts & converts format.
7. Application shows the email in the inbox.

3. Key Points of Working

- Each layer **only talks to its adjacent layers**.
- Every layer adds its own **header/trailer** (extra info).
- Final goal → Ensure sender and receiver **understand each other** regardless of hardware/software differences.

4. Ethical Hacking Angle

- Hackers target specific layers depending on attack:
  - **Layer 2:** ARP spoofing → trick Data Link.
  - **Layer 3:** IP spoofing → fake identity at Network.
  - **Layer 4:** SYN Flood → exploit TCP handshake.
  - **Layer 7:** DDoS / SQL Injection → exploit Application.

📝 Short Notes: How OSI Works

- Data goes **down 7 layers (sender)**, then **up 7 layers (receiver)**.
- **Encapsulation:** Each layer adds its header (extra info).
- **Decapsulation:** Receiver removes headers to get actual data.
- Example: Sending email → SMTP (App) → TCP (Port) → IP (Address) → MAC → Bits → Receiver reverses process.
- Used to **standardize communication** & **identify attack points**.

TCP/IP Model -

1. Definition

- **TCP/IP Model** = A practical networking model that defines how data is transmitted over the internet.
- Developed by **DoD (Department of Defense)** in the 1970s.
- Unlike the **OSI model (theoretical)**, TCP/IP is **real-world implementation** used in all networks today.

2. Layers of TCP/IP Model (4 Layers)

1. **Application Layer**
   - Combines OSI’s Application, Presentation, and Session layers.
   - Provides services for user applications.
   - Examples: **HTTP, HTTPS, FTP, DNS, SMTP, DHCP**.
2. **Transport Layer**
   - Responsible for end-to-end communication.
   - Provides error checking, reliability, and ports.
   - Protocols: **TCP (connection-oriented), UDP (connectionless)**.
3. **Internet Layer**
   - Handles logical addressing & routing.
   - Ensures data finds the best path across networks.
   - Protocols: **IP, ICMP, ARP, IGMP**.
4. **Network Access Layer (Link Layer)**
   - Deals with actual hardware transmission.
   - Includes physical devices, drivers, and protocols.
   - Examples: **Ethernet, Wi-Fi, MAC, PPP**.

3. How TCP/IP Works (Example: Opening a Website)

1. **Application Layer:** Browser sends HTTP request to web server.
2. **Transport Layer:** TCP breaks data into segments, assigns port numbers (e.g., Port 80/443).
3. **Internet Layer:** IP adds source & destination IP addresses.
4. **Network Access Layer:** Data is converted to frames → bits → transmitted via cable/Wi-Fi.
5. **Receiver:** Process reversed (decapsulation) → webpage displayed.

4. Difference from OSI Model

- **OSI Model** = 7 layers (theoretical).
- **TCP/IP Model** = 4 layers (practical, used in real networks).
- OSI is a **reference framework**, TCP/IP is **implementation standard**.

5. Ethical Hacking Relevance

- Hackers analyze TCP/IP to find vulnerabilities:
  - **Transport Layer:** SYN flood (TCP handshake attack).
  - **Internet Layer:** IP spoofing, ICMP flood.
  - **Application Layer:** HTTP attacks (XSS, SQL Injection).

📝 Short Notes: TCP/IP Model

- **Definition:** Practical model for internet communication, made by DoD.
- **Layers (4):** Application, Transport, Internet, Network Access.
- **Working:** Data moves down sender layers (encapsulation) → across network → up receiver layers (decapsulation).
- **Use:** Basis of modern networking & internet.
- **Vs OSI:** OSI = 7 layers (theoretical), TCP/IP = 4 layers (practical).

Linux -

1. Definition

- **Linux** is an **open-source, Unix-like operating system kernel** first created by **Linus Torvalds in 1991**.
- It's widely used in servers, security systems, and hacking because it's **free, customizable, and secure**.
- Popular in **ethical hacking & cybersecurity** (Kali Linux, Parrot OS, Ubuntu).

2. Key Features of Linux

- **Open Source** → Anyone can modify and use it.
- **Secure** → Better permissions and user control than Windows.
- **Multi-User & Multitasking** → Many users can log in & run tasks at the same time.
- **Command-Line Interface (CLI)** → Powerful terminal for hacking tools & automation.
- **Lightweight & Fast** → Runs on old hardware.
- **Community Support** → Huge global support.

3. Linux Distributions (Distros)

- Different versions of Linux made for different purposes:
  - **Kali Linux** → Ethical hacking & penetration testing.
  - **Ubuntu** → General-purpose, user-friendly.
  - **Parrot Security OS** → Advanced penetration testing.

  - **CentOS / Debian / Fedora** → Servers & enterprise.

4. File System Structure

- Linux organizes files in a **hierarchical structure**:
  - / → Root directory (base of everything).
  - /home → User files.
  - /etc → Configuration files.
  - /bin → Basic commands (ls, cp, mv).
  - /var → Logs, temporary files.

5. Why Linux in Ethical Hacking?

- Most hacking tools are **built for Linux**.
- Greater **control over network & system resources**.
- Provides **powerful scripting (Bash, Python)**.
- Used for **servers, exploits, reverse engineering**.

6. Example in Hacking

- A hacker uses **Kali Linux** to run nmap (network scanner) from terminal to find open ports in a target system.

📝 Short Notes: Linux

- **Linux** = Open-source, Unix-like OS (1991 by Linus Torvalds).
- **Features:** Secure, free, customizable, multitasking, CLI support.
- **Distros:** Kali (hacking), Ubuntu (general), Parrot (security).
- **File System:** / root, /home, /etc, /bin, /var.
- **Use in Hacking:** Supports tools (Nmap, Metasploit), scripting, penetration testing.

Features of Linux -

1. Open Source

- Linux is **free and open-source** → source code available for anyone to use, modify, and distribute.
- Encourages customization and innovation.

2. Security

- Strong **user permissions** and **firewall tools**.
- Harder for malware/viruses compared to Windows.
- Widely used in **cybersecurity & servers**.

3. Multi-User

- Multiple users can work on the same system **simultaneously** without interfering.

4. Multitasking

- Linux handles **several processes at once** (running programs, background tasks, services).

5. Portability

- Can run on almost any hardware → PCs, servers, mobiles (Android is Linux-based), IoT devices.

6. Stability & Performance

- Rarely crashes, can run for **years without reboot** (important for servers).
- Efficient use of system resources.

7. Shell/Command Line Interface (CLI)

- Powerful terminal for executing commands.
- Essential for **hacking, scripting, automation**.

8. File System Structure

- Organized **hierarchical structure** (/, /home, /etc, /var).
- Makes file management simple.

9. Community Support

- Large global community, thousands of forums & tutorials.

10. Distribution Variety

- Many **distros (distributions)** for different needs:
  - **Kali Linux** → Ethical hacking.
  - **Ubuntu** → Beginners/general users.
  - **CentOS/Debian** → Servers.

📝 Short Notes: Features of Linux

- Open-source & free.

- Highly secure (permissions, firewall).
- Multi-user & multitasking.
- Portable (runs on any hardware).
- Stable & fast performance.
- Powerful CLI (shell).
- Organized file system.
- Huge community support.
- Many distributions (Kali, Ubuntu, Fedora, etc.).

Basic Linux File System -

Linux uses a hierarchical file system (tree-like structure).
Everything starts from / (root directory) → the top of the hierarchy.

🔑 Important Directories in Linux

1. **/ (Root Directory)**
   - **The base of the Linux file system.**
   - **Every file & directory starts from here.**
2. **/home**
   - **Stores user files & folders.**
   - **Example: /home/alice → Alice's personal directory.**
3. **/bin (Binary)**
   - **Contains basic user commands (ls, cp, mv, cat).**
   - **Needed even in single-user mode.**
4. **/sbin (System Binaries)**
   - **Commands for system administration (shutdown, reboot, ifconfig).**
   - **Used mostly by root user.**
5. **/etc (Configuration)**
   - **Stores system-wide configuration files.**
   - **Example: /etc/passwd (user accounts), /etc/hosts.**
6. **/var (Variable Data)**
   - **Holds log files, mail, temporary data.**
   - **Example: /var/log/syslog.**
7. **/usr (User Programs)**
   - **Applications & libraries installed by users.**
   - **/usr/bin → extra commands.**
   - **/usr/lib → libraries.**
8. **/tmp (Temporary Files)**
   - **Stores temporary files (auto-deleted on reboot).**
9. **/dev (Devices)**
   - **Represents hardware devices as files.**

  - Example: /dev/sda (hard disk).

10. **/proc (Processes)**
    - Virtual directory that shows system processes & kernel info.
    - Example: /proc/cpuinfo.
11. **/boot**
    - Files needed to boot Linux (kernel, GRUB bootloader).
12. **/lib**
    - Libraries required by /bin and /sbin.
13. **/opt**
    - Optional software packages (extra apps).
14. **/mnt and /media**
    - Used for mounting external devices (USB, DVD, external HDD).

📝 Short Notes: Basic Linux File System

- / → Root (base directory).
- /home → User files.
- /bin → Basic commands.
- /sbin → System admin commands.
- /etc → Config files.
- /var → Logs & variable data.
- /usr → User programs.
- /tmp → Temporary files.
- /dev → Device files.
- /proc → Process/kernel info.
- /boot → Boot files.
- /lib → Libraries.
- /opt → Optional software.
- /mnt, /media → Mounted devices.