1.Cybersecurity Basics

1. CIA Triad: Core of Cybersecurity

The CIA Triad is the foundation of information security. Every security measure ties back to one or more of these principles.

a) Confidentiality

- **Definition:** Ensuring that sensitive information is only accessible to authorized users.
- **Goal:** Protect data from unauthorized access and leaks.
- **Techniques/Tools:**
    - Encryption (AES, RSA)
    - Access Control (Role-Based Access Control – RBAC)
    - Password policies
    - Multi-Factor Authentication (MFA)
- **Example:** Your bank account info should only be visible to you and the bank. If someone steals your login credentials, confidentiality is breached.

b) Integrity

- **Definition:** Ensuring that data remains accurate, consistent, and unaltered unless authorized.
- **Goal:** Prevent unauthorized modification of data.

- **Techniques/Tools:**
    - Hashing (SHA-256)
    - Digital Signatures
    - Version control systems
    - Checksums
- **Example:** If someone changes transaction data in a banking system, integrity is compromised. A hash value can verify the original data hasn't been tampered with.

c) Availability

- **Definition:** Ensuring that systems, applications, and data are accessible when needed by authorized users.
- **Goal:** Minimize downtime and disruption.
- **Techniques/Tools:**
    - Redundant systems and backups
    - Load balancers
    - DDoS mitigation tools
    - Cloud failover strategies
- **Example:** A hospital's patient database must be available to doctors 24/7. A server outage could endanger lives.

---

2. Threat Types: What Can Attack You?

a) Phishing

- **Definition:** Deceptive attempts (usually via email, SMS, or websites) to trick users into giving sensitive information.
- **How it works:** Attackers send emails or links that look legitimate but lead to credential theft.
- **Defense:** Email filters, awareness training, MFA.

---

b) Malware

- **Definition:** Malicious software designed to damage or exploit systems.
- **Types:**
    - **Virus:** Attaches to files, spreads when executed.
    - **Worm:** Self-replicates across networks.
    - **Trojan:** Disguised as legitimate software.
    - **Spyware/Adware:** Monitors activity or delivers ads.
- **Defense:** Antivirus, firewalls, software updates.

---

c) DDoS (Distributed Denial of Service)

- **Definition:** Overwhelming a server or network with excessive traffic to make it unavailable.

- **Example:** A website receives millions of fake requests and crashes.
- **Defense:** DDoS protection services (Cloudflare, AWS Shield), traffic filtering.

---

d) SQL Injection

- **Definition:** Attackers insert malicious SQL queries into input fields to manipulate databases.
- **Impact:** Can reveal sensitive info, modify or delete data.
- **Defense:** Parameterized queries, input validation, web application firewalls.

---

e) Brute Force Attack

- **Definition:** Attackers try all possible password combinations to gain unauthorized access.
- **Defense:** Strong passwords, account lockouts, rate limiting, MFA.

---

f) Ransomware

- **Definition:** Malware that encrypts files and demands a ransom for the decryption key.
- **Impact:** Can cripple organizations (hospitals, corporations).
- **Defense:** Regular backups, patching systems, endpoint protection.

3. Attack Vectors: How Attackers Get In

a) Social Engineering

- **Definition:** Manipulating people into revealing confidential information.

- **Techniques:**
    - Pretexting: Pretending to be someone trustworthy.
    - Baiting: Offering something attractive to get info.
    - Tailgating: Physically following someone into restricted areas.
- **Defense:** Security awareness training, strict verification processes.

---

b) Wireless Attacks

- **Definition:** Exploiting vulnerabilities in Wi-Fi or Bluetooth networks.
- **Examples:**
    - Rogue Access Points
    - Wi-Fi eavesdropping (sniffing)
    - Man-in-the-middle (MITM) attacks on unencrypted networks
- **Defense:** WPA3, VPNs, disabling unused wireless interfaces.

c) Insider Threats

- **Definition:** Threats from employees, contractors, or anyone with authorized access who abuses it.
- **Types:**
    - Malicious insiders: Steal or sabotage data.
    - Negligent insiders: Accidental leaks due to carelessness.
- **Defense:** Monitoring, access controls, employee training, least-privilege principle.

Summary

- **CIA Triad** → Protect **data** (Confidentiality), keep it **accurate** (Integrity), and **available** (Availability).
- **Threat Types** → Know **what attacks exist**: phishing, malware, DDoS, SQL injection, brute force, ransomware.
- **Attack Vectors** → Understand **how attackers get in**: social engineering, wireless attacks, insider threats.

2.Linux Fundamentals

1. File System Navigation

cd (Change Directory)

Used to move between directories in the Linux file system.
Purpose: Navigate to required project folders, logs, scripts, configurations.
Examples:

- `cd /home/user/Documents` → Go to a specific directory.
- `cd ..` → Move one level up.
- `cd ~` → Go to the home directory.

ls (List Files)

Displays files and folders in the current directory.
Purpose: Check available files, permissions, and directory structure.
Examples:

- `ls` → Basic listing.
- `ls -l` → Detailed view with permissions and file sizes.

- `ls -a` → Shows hidden files.

pwd (Print Working Directory)

Shows the full path of the current working directory.
Purpose: Helps avoid working in the wrong folder during configuration or scripting tasks.
Example:

- `pwd` → Outputs `/home/user/project`.

---

2. File & Directory Permissions

chmod (Change Permissions)

Used to modify read, write, execute permissions for files and directories.
Purpose: Secure files, restrict access, allow script execution.
Modes:

- Numeric: `chmod 755 script.sh`
- Symbolic: `chmod u+x script.sh`

Meaning of digits (example 755):

- **7** → Owner: read + write + execute
- **5** → Group: read + execute
- **5** → Others: read + execute

chown (Change Ownership)

Changes the owner or group of a file/directory.
Purpose: Assign correct ownership for system processes, services, and user access.
Examples:

- `chown user file.txt` → Change owner.
- `chown user:group file.txt` → Change owner and group.
- `sudo chown -R www-data:/var/www` → Apply recursively.

---

3. Package Management (Debian/Ubuntu)

apt (Advanced Package Tool)

Used to install, update, and remove software packages.
Purpose: Manage software required for development, networking, and security tasks.
Examples:

- `sudo apt update` → Refresh package lists.
- `sudo apt upgrade` → Update installed packages.
- `sudo apt install nmap` → Install a package.
- `sudo apt remove package` → Remove a package.

dpkg (Debian Package)

Low-level tool to install `.deb` files manually.
Purpose: Install packages not available through apt, and troubleshoot installation errors.
Examples:

- `sudo dpkg -i package.deb` → Install `.deb` file.

- `sudo dpkg -l` → List installed packages.
- `sudo dpkg -r package` → Remove package.

---

4. Networking Commands

ifconfig

Shows and configures network interfaces.
Purpose: Check IP address, network status, troubleshoot interface issues.
Examples:

- `ifconfig` → View interfaces.
- `sudo ifconfig eth0 down` → Disable interface.
- `sudo ifconfig eth0 up` → Enable interface.

(Note: On modern systems, `ip a` is preferred.)

ping

Tests connectivity between your system and a target (IP/domain).
Purpose: Diagnose network availability, latency, and packet loss.
Examples:

- `ping google.com`

- `ping -c 4 8.8.8.8` → Send 4 packets only.

netstat

Displays active network connections, routing tables, and ports.
Purpose: Audit open ports, monitor connections, check listening services.
Examples:

- `netstat -tuln` → Show listening TCP/UDP ports.
- `netstat -anp` → Show processes linked to network activity.

(Note: Modern alternative: `ss` command.)

traceroute

Shows the exact path packets take to reach a destination.
Purpose: Identify network hops, delay points, and route failures.
Example:

- `traceroute google.com`

3.Networking Basics

1. OSI MODEL – LAYERS & FUNCTIONS

The **OSI (Open Systems Interconnection)** Model is a conceptual framework that standardizes how data moves across a network.
It consists of **7 layers**, each with a specific function and responsibility.

Layer 7 – Application Layer

- Closest to the user.
- Provides network services like email, web browsing, and file transfer.
- Protocols: **HTTP, HTTPS, DNS, FTP, SMTP, SSH**.

Function: Enables applications to communicate over the network.

Layer 6 – Presentation Layer

- Translates data formats (encryption, compression).
- Converts data into a common format for communication.

Function: Makes sure data is readable regardless of system differences.
Example: JPEG, MP4, TLS encryption.

Layer 5 – Session Layer

- Manages communication sessions between two systems.
- Handles opening, maintaining, and closing connections.

Function: Ensures continuous communication between devices.

Layer 4 – Transport Layer

- Controls data flow using segmentation and reassembly.
- Provides reliable or unreliable transport.

Protocols:

- **TCP** – reliable, connection-oriented
- **UDP** – fast, connectionless

Function: End-to-end communication and error handling.

Layer 3 – Network Layer

- Handles routing of data packets.
- Assigns logical addressing (IP address).

Protocols: IP, ICMP, ARP
Devices: Routers

Function: Determines the best path for data across networks.

Layer 2 – Data Link Layer

- Deals with frames, MAC addresses, and switching.
- Ensures error-free delivery over physical hardware.

Devices: Switches
Protocols: Ethernet, PPP

Function: Node-to-node communication.

Layer 1 – Physical Layer

- Transmits raw bits over cables, radio, or fiber.
- Defines voltage levels, connectors, frequencies.

Devices: Cables, NICs, hubs
Function: Physical transmission of signals.

---

2. TCP/IP PROTOCOL SUITE

TCP/IP is the real-world networking model used in the internet.
It has only **4 layers**, combining OSI concepts.

Layer 4 – Application Layer

Equivalent to OSI Layers 5, 6, and 7.
Protocols:

- **HTTP/HTTPS**
- **DNS**
- **FTP**
- **SSH**
- **SMTP/POP3/IMAP**

Layer 3 – Transport Layer

Manages end-to-end data communication.

TCP (Transmission Control Protocol)

- Reliable
- Connection-oriented
- Guarantees delivery
  Used for: HTTPS, email, file transfers.

UDP (User Datagram Protocol)

- Faster
- Unreliable
- No connection
  Used for: DNS, gaming, streaming.

Layer 2 – Internet Layer

Routes packets across networks.

Protocols:

- **IP (IPv4/IPv6)**
- **ICMP**
- **ARP**

Layer 1 – Network Access Layer

Handles the physical and data link operations:

- MAC addresses
- Frames
- Network hardware (switches, Wi-Fi, Ethernet)

---

3. DNS & HTTP/HTTPS DEEP DIVE

DNS (Domain Name System)

DNS converts human-readable domain names into IP addresses.

Why DNS matters

- Computers understand IP addresses, not names.
- Makes the internet usable for humans.
- Critical for almost every online service.

How DNS Works (Step-by-Step)

1. User enters `www.example.com`.
2. Browser checks cache.

3. DNS resolver asks root servers.
4. Root points to TLD (Top-Level Domain).
5. TLD points to domain's authoritative server.
6. Authoritative server returns the IP.
7. Browser connects to that IP.

DNS record types

- **A** → Maps domain to IPv4
- **AAAA** → IPv6
- **CNAME** → Alias
- **MX** → Mail server
- **NS** → Nameserver
- **TXT** → Security/verification (SPF, DKIM)

HTTP & HTTPS

HTTP (HyperText Transfer Protocol)

- A stateless protocol used for web communication.
- Sends information as plain text.
- Not secure.

HTTPS

- HTTP + SSL/TLS encryption
- Protects data from attackers
- Uses certificates issued by CAs (Certificate Authorities)

How HTTPS Works

1. Client requests website.
2. Server sends SSL certificate.
3. Browser verifies the certificate.
4. A secure session key is created.
5. Encrypted communication begins.

Importance

- Essential for login pages, banking, payments, personal data.
- Prevents Man-in-the-Middle (MITM) attacks.

---

4. IP ADDRESSING, SUBNETTING, AND NAT

IP Addressing

An IP address uniquely identifies a device on a network.

IPv4 Format

Example: `192.168.1.10`
Consists of 4 octets (32-bit).

IPv6 Format

Example: `2001:0db8:85a3::7334`
Uses 128-bit addressing for the modern internet.

Types of IPs

- **Public IP** → Internet-facing
- **Private IP** → Internal networks
  Examples:
  - 192.168.x.x
  - 10.x.x.x
  - 172.16.x.x – 172.31.x.x
- **Static IP** (fixed)
- **Dynamic IP** (changes, assigned by DHCP)

Subnetting

Subnetting divides a large network into smaller, efficient sub-networks.

Why Subnetting is used

- Reduces network congestion
- Enhances security
- Efficient IP allocation
- Helps in routing

CIDR Notation

Example: `192.168.1.0/24`

Meaning:

- `/24` → first 24 bits are network part
- Remaining 8 bits are host part
- Total hosts = $2^8 - 2$ = **254 hosts**

Quick Reference

CIDR	Host Count
/24	254 hosts
/25	126 hosts
/26	62 hosts
/27	30 hosts
/30	2 hosts

NAT (Network Address Translation)

NAT allows multiple private devices to access the internet using a single public IP.

Why NAT is important

- Conserves public IP addresses.
- Provides security by hiding internal IPs.
- Used in routers and firewalls.

Types of NAT

1. **SNAT (Source NAT)**
   Used for private-to-public translation.
2. **DNAT (Destination NAT)**
   Used for port forwarding (ex: exposing a local server).
3. **PAT (Port Address Translation)**
   Many private IPs share one public IP using different ports.
   Example: Home Wi-Fi networks.

4.Cryptography Basics

1. Symmetric vs Asymmetric Encryption

Cryptography uses two major encryption techniques to secure data:

A. Symmetric Encryption

Symmetric encryption uses **one single key** for both encryption and decryption.

Key Points

- One key = shared between sender and receiver
- Fast, efficient, used for large data
- Requires secure key sharing
- Used in: **AES, DES, 3DES, Blowfish**

Example Process

1. Sender encrypts message using a secret key.
2. Receiver decrypts the message with the same key.

Advantages

- High speed
- Low computational cost
- Suitable for bulk data encryption

Disadvantages

- Key distribution problem
- If the key leaks, security is broken

Common Algorithms

- **AES-128/192/256** (industry standard)
- **DES** (outdated)
- **3DES** (deprecated but still seen in legacy systems)

B. Asymmetric Encryption

Asymmetric encryption uses **two keys**:

- **Public Key** (shared with everyone)
- **Private Key** (kept secret)

Key Points

- Data encrypted with the public key can only be decrypted with the private key
- Used for authentication, digital signatures, secure key exchange
- Slower than symmetric encryption
- Used in: **RSA, ECC, Diffie–Hellman**

Example Uses

- Secure login
- SSL/TLS certificates
- SSH authentication
- Digital signatures

Advantages

- No need to share private keys
- More secure key exchange
- Supports authentication

Disadvantages

- Slower
- Not suitable for large data encryption

Comparison

Feature	Symmetric	Asymmetric
Keys	Single shared key	Public + Private keys
Speed	Fast	Slower
Use Cases	Bulk data encryption	Authentication, key exchange
Security	Risky key sharing	More secure

---

2. Hashing (MD5, SHA256)

Hashing converts data into a fixed-length string called a **hash value**.
It is a **one-way process** — hashes cannot be reversed.

Key Properties of Hash Functions

- One-way (non-reversible)
- Produces fixed-length output
- Small input changes produce huge output changes
- Collision-resistant (hard to find two inputs with same hash)

MD5 (Message Digest Algorithm 5)

- Produces a **128-bit hash**
- Fast but **cryptographically broken**
- Vulnerable to collision attacks
- Still used for file checksums, but not for security

Example Hash Output:

```
d41d8cd98f00b204e9800998ecf8427e
```

SHA-256 (Secure Hash Algorithm 256-bit)

- Part of the SHA-2 family
- Produces a **256-bit hash**
- Highly secure, widely used in modern systems
- Used in:

- Blockchain
- Password storage
- SSL certificates
- Digital signatures

Example Hash Output:

```
e3b0c44298fc1c149afbf4c8996fb924...
```

---

3. Digital Certificates & SSL/TLS

What is a Digital Certificate?

A digital certificate is an electronic document that verifies the identity of a website, user, or device.
It binds a **public key** to an organization.

Contents of a Certificate

- Owner name
- Public key
- Signature of Certificate Authority (CA)
- Expiry date
- Serial number

Issued by

- Certificate Authorities (CA) like:
    - DigiCert
    - Let's Encrypt
    - GlobalSign
    - GoDaddy

SSL/TLS (Secure Socket Layer / Transport Layer Security)

Purpose

- Encrypt communication between client and server
- Prevent eavesdropping and tampering
- Authenticate server identity

How SSL/TLS Works (Simplified TLS Handshake)

1. Client connects to website
2. Server sends its digital certificate
3. Browser verifies certificate with CA
4. A session key is generated (symmetric key)
5. Encrypted communication begins

Key Insight

- TLS uses **asymmetric encryption** to exchange keys
- Then uses **symmetric encryption** for actual data transfer (for speed)

5.Tool Familiarization

Wireshark – Packet Capture & Network Traffic Analysis

Wireshark is an open-source tool for capturing and analyzing network packets. It helps cybersecurity professionals inspect network protocols, troubleshoot issues, and detect malicious activities.

Key Capabilities:

- Real-time packet capture from network interfaces.
- Detailed analysis of protocols such as TCP, UDP, ARP, DNS, HTTP, DHCP, TLS.
- Ability to filter traffic using display filters (e.g., `http`, `ip.addr == 192.168.1.1`).
- Reconstruction of TCP streams to view full conversations.
- Detection of anomalies like packet drops, retransmissions, or suspicious flows.

Use Cases:

- Diagnose network latency issues.

- Identify unauthorized communication.
- Analyze malware communication patterns.
- Inspect HTTP requests and responses.

Nmap – Network Scanning and Enumeration

Nmap (Network Mapper) is a powerful scanning tool used to discover hosts, services, and vulnerabilities in a network.

Key Capabilities:

- Host discovery to find active devices.
- Port scanning including TCP SYN scan and UDP scan.
- Service version detection.
- OS fingerprinting.
- Running Nmap Scripting Engine (NSE) scripts for vulnerability detection.

Common Commands:

- `nmap <IP>` – Basic scan.
- `nmap -sV <IP>` – Detect services and versions.
- `nmap -O <IP>` – OS detection.

- `nmap -A <IP>` – Aggressive scan (OS + version + scripts).
- `nmap --script vuln <IP>` – Quick vulnerability scan.

Use Cases:

- Initial reconnaissance in penetration testing.
- Checking open or misconfigured ports.
- Mapping a network's attack surface.
- Identifying outdated or risky services.

Burp Suite – Web Application Security Testing

Burp Suite is an industry-standard tool for analyzing and testing web applications. It acts as a proxy to intercept HTTP/HTTPS traffic between the browser and server.

Key Components:

- **Proxy** – intercept and modify requests.
- **Intruder** – automated attack tool for brute-force or fuzzing.
- **Repeater** – manually modify and resend requests.
- **Scanner (Pro Version)** – automatically detects vulnerabilities.

What Burp Helps Test:

- SQL Injection
- Cross-Site Scripting (XSS)
- Broken authentication

- Session hijacking
- Insecure cookies
- Request tampering

Hands-on Tasks:

- Captured and analyzed live HTTP requests.
- Modified requests to test server responses.
- Inspected input fields for injection vulnerabilities.

Netcat – Network Debugging & Testing

Netcat, often called the “Swiss Army knife” of networking, is highly versatile for creating TCP/UDP connections, debugging networks, and transferring data.

Key Capabilities:

- Listen on specific ports or connect to remote hosts.
- Transfer files over the network.
- Perform banner grabbing.
- Simple port scanning.

Common Commands:

- `nc -lvp 4444` – Listen on port 4444.
- `nc <IP> 4444` – Connect to a remote listener.

- `nc -zv <IP> 1-1000` – Quick port scan.
- `nc <IP> <PORT> > file.txt` – Receive file.

Use Cases:

- Testing if a port is open.
- Debugging network connectivity.
- Sending and receiving logs.
- Simple client-server testing.
- Checking service banners for identification.

Table for Tools

Tool	Purpose	Typical Use
Wireshark	Packet capture & analysis	Network troubleshooting, protocol inspection
Nmap	Host, port & vulnerability scan	Reconnaissance, network mapping
Burp Suite	Web app security testing	HTTP/HTTPS interception, injection testing
Netcat	Debugging & port communication	File transfer, connectivity testing, banner grabbing