1. Cybersecurity

Cybersecurity means the practice of protecting **computers, networks, systems, and data** from digital attacks, unauthorized access, damage, or theft.

CIA triad

CIA triad (Confidentiality, Integrity, Availability) is its core model for thinking about what needs to be protected.
The CIA triad is a framework that defines three essential goals of information security:

1. Confidentiality **(keeping data secret)** : Ensuring that information is accessible only to those authorized to see it.
    - A breach of confidentiality here could mean leaked private chats, impersonation, or doxxing (publishing someone's personal info without consent)
2. Integrity **(ensuring data is accurate)** : Ensuring that data is accurate, complete, and has not been tampered with.
    - A loss of integrity could mean fake news, impersonation, or manipulated images/videos being spread as if they were real.
3. Availability **(keeping systems accessible)**: Ensuring that systems and data are accessible and usable by authorized users when needed.
    - A major outage or attack can make the platform unusable for hours, leading to user frustration, loss of trust, and business impact.

Principle	Meaning	Real-World Example
Confidentiality	Ensuring only authorized users can access sensitive data.	**Banking**: Customers' account details are encrypted and protected with multi-factor authentication. **Social media**: Private messages are restricted to sender and recipient.

Principle	Meaning	Real-World Example
Integrity	Guaranteeing that data is accurate, consistent, and not tampered with.	**Banking**: Transaction records must remain unchanged; even a small alteration could cause fraud. **Social media**: Posts and profiles must not be maliciously altered to spread misinformation.
Availability	Ensuring systems and data are accessible when needed.	**Banking**: Online banking services must be available 24/7; downtime could block critical payments. **Social media**: Platforms must remain online to allow communication, even during high traffic or cyberattacks.

2. Different types of attackers

Attacker Type	Characteristics	Motivation	Example Scenario
Script kiddies	Amateur hackers with limited skills, relying on pre-made tools or malware kits.	Thrill, curiosity, or reputation among peers.	A teenager launching a DDoS attack on a gaming server using downloaded software.
Insiders	Employees or contractors with legitimate access who misuse it.	Revenge, financial gain, coercion.	A disgruntled staff member leaking sensitive customer data to competitors.
Hacktivists	Ideologically motivated attackers targeting organizations to promote political or social causes.	Activism, protest, exposure.	Anonymous defacing government websites to protest policies.

Attacker Type	Characteristics	Motivation	Example Scenario
Nation-state actors	Highly skilled, government-sponsored groups with vast resources.	Espionage, sabotage, geopolitical goals.	APT29 (Russia) or APT41 (China) conducting cyber espionage against critical infrastructure

Attack Surface	Key Risks	Real-World Example
Web applications	SQL injection, cross-site scripting (XSS), session hijacking.	An e-commerce site leaking credit card data due to poor input validation.
Mobile apps	Insecure storage, weak authentication, malicious third-party libraries.	A banking app exposing user PINs because of unencrypted local storage.
APIs	Broken authentication, excessive data exposure, lack of rate limiting.	Social media APIs leaking private user data through poorly secured endpoints.
Networks	Man-in-the-middle attacks, unpatched routers, insecure Wi-Fi.	Attackers intercepting traffic on public Wi-Fi to steal login credentials.
Cloud infrastructure	Misconfigured storage buckets, weak IAM policies, shared responsibility gaps.	Sensitive files exposed in public AWS S3 buckets due to misconfiguration.

Introduction to the OWASP Top Ten

- A1: Injection
- A2: Broken Authentication
- A3: Sensitive Data Exposure
- A4: XML External Entities (XXE)
- A5: Broken Access Control
- A6: Security Misconfiguration

- A7: Cross-Site Scripting (XSS)
- A8: Insecure Deserialization
- A9: Using Components with Known Vulnerabilities
- A10: Insufficient Logging and Monitoring

Email (Gmail, Outlook, etc.)

Email is one of the richest attack surfaces because it holds login links, passwords, financial info, and is used for account recovery.

1. User interface / inbox

- Phishing emails that look like banks, social media, or colleagues, tricking the user into clicking a malicious link or opening an infected attachment.
- Fake “urgent” messages (e.g., “Your account will be closed”) that pressure the user into giving credentials or downloading malware.

2. Login & authentication

- Weak or reused passwords that can be cracked or used in credential-stuffing attacks.
- Lack of multi-factor authentication (MFA), allowing attackers to log in with just a stolen password.
- Session hijacking if the user logs in over an unsecured Wi-Fi network and the attacker steals the session cookie.

3. Network communication

- Unencrypted connections (no TLS/SSL) that let attackers eavesdrop on emails and passwords.
- Man-in-the-middle (MitM) attacks on public Wi-Fi, where the attacker intercepts and modifies email traffic.

4. Device & app

- Malware on the device that reads emails, steals saved passwords, or logs keystrokes.
- Outdated email clients or OS with known vulnerabilities that can be exploited to gain access.

5. Account recovery

- Weak security questions (e.g., mother's maiden name) that can be guessed or found on social media.
- Recovery email or phone number that is compromised, allowing an attacker to reset the password.

WhatsApp / Messaging apps

Messaging apps like WhatsApp are attack surfaces for social engineering, malware, and device compromise, especially because they are personal and trusted.

1. User interface / chat

- Social engineering: fake messages from "friends," "HR," or "customer support" asking for money, passwords, or personal details.
- Impersonation of executives or family members to trick the user into transferring money or sharing sensitive info.
- Fake job offers or "verified" accounts that collect personal data or credentials.

2. Links & media

- Malicious links that open in the app's built-in browser and steal login sessions or install malware.
- Exploitable media files (images, videos, documents) that trigger vulnerabilities in the app or OS when opened.
- Fake websites that look like banks or social media, used to harvest credentials after the user clicks a link.

3. Authentication & sessions

- SIM-swap attacks where the attacker takes over the phone number and registers it on WhatsApp, gaining access to all messages.
- Session hijacking if the user logs in on a public or shared device and forgets to log out.
- Weak or no screen lock on the device, allowing anyone with physical access to read messages.

4. Device & app

- Outdated WhatsApp or OS with known vulnerabilities (e.g., memory corruption bugs) that allow remote code execution.
- Malware that reads messages, steals 2FA codes, or sends messages on behalf of the user.
- Jailbroken/rooted devices that bypass app sandboxing and allow deeper access to messages and data.

5. Cloud & backups

- Unencrypted or weakly protected cloud backups (e.g., Google Drive, iCloud) that can be accessed if the cloud account is compromised.
- Backup files that contain chat history, contacts, and media, which can be valuable to attackers.

Mobile banking apps

Banking apps are high-value targets because they give direct access to money and sensitive financial data.

1. User interface / login

- Fake login screens (overlay apps or phishing sites) that capture the user's username, password, and OTP.
- Social engineering via SMS, WhatsApp, or calls pretending to be the bank, asking for credentials or OTPs.
- Weak or reused passwords that are vulnerable to credential stuffing.

2. Authentication & sessions

- Lack of strong MFA (e.g., only SMS OTP) that can be intercepted via SIM-swap or malware.
- Session tokens stored insecurely on the device, which can be stolen by malware.
- Session fixation or hijacking on public Wi-Fi if the app doesn't use proper session management.

3. Network communication

- Man-in-the-middle attacks on unsecured Wi-Fi that intercept login credentials or transaction details.
- Weak or missing TLS/SSL, or certificate pinning not enforced, allowing attackers to decrypt traffic.

4. Device & app

- Malware (banking trojans) that overlays the banking app, captures keystrokes, or steals 2FA codes.
- Fake banking apps downloaded from third-party stores that look identical to the real app but steal credentials.
- Jailbroken/rooted devices that allow attackers to bypass app protections and read stored data.

5. Data storage

- Sensitive data (account numbers, transaction history, passwords) stored in plain text or weakly encrypted on the device.
- Logs or cache files that accidentally contain sensitive information and can be accessed by other apps or malware.

6. Transactions & business logic

- Logic flaws that allow unlimited transactions, incorrect amounts, or bypassing approval steps.
- Transaction manipulation where an attacker changes the payee or amount during transfer (if integrity checks are weak).

7. Physical device

- Stolen or lost phones with weak or no screen lock, allowing direct access to the banking app.
- Physical tampering (e.g., installing spyware) on a device that is left unattended.

🔄 Data Flow Overview

1. **User**

- The journey begins when a user interacts with a system through a device (browser, mobile app, desktop client).
- Example: A customer enters login credentials on a banking app.

2. **Application**
    - The application (front-end) captures user input and processes it locally (e.g., form validation, encryption).
    - Example: The banking app encrypts the password before sending it.
3. **Server**
    - The application sends the request to a backend server via secure protocols (HTTPS).
    - The server acts as the middle layer, handling business logic, authentication, and authorization.
    - Example: The server checks if the user's credentials match stored records.
4. **Database**
    - The server queries the database to fetch or update information.
    - The database stores structured data (e.g., customer accounts, transactions).
    - Example: The server retrieves account balance from the database and sends it back to the application.
5. **Response Flow Back**
    - The database returns results to the server.
    - The server processes and formats the response.
    - The application displays the result to the user in a readable format.
    - Example: The banking app shows "Your balance is $5,000."

Visual Representation (Conceptual)

Code

User → Application → Server → Database

↑ ↓

←———————— Response ————————

Real-World Example

- **Banking app login**:
    - User enters credentials → Application encrypts and sends → Server validates → Database confirms → Response shows account dashboard.
- **Social media post**:
    - User writes a post → Application sends content → Server applies rules (e.g., moderation, formatting) → Database stores post → Response confirms post is live.

Attack Points in the Data Flow

1. **User Layer**
    - **Phishing**: Trick users into revealing credentials.
    - **Social engineering**: Manipulate users into bypassing security.
    - **Malware on device**: Keyloggers or spyware capturing input before it's encrypted.
2. **Application Layer**
    - **Injection attacks**: SQL injection, XSS, command injection via poorly validated input.
    - **Session hijacking**: Stealing cookies or tokens to impersonate users.
    - **Weak authentication**: Exploiting weak passwords or broken login mechanisms.
3. **Server Layer**
    - **Denial-of-Service (DoS/DDoS)**: Overloading servers to make them unavailable.
    - **Privilege escalation**: Exploiting flaws to gain higher-level access.

- **Unpatched vulnerabilities**: Exploiting outdated software or misconfigurations.

4. **Database Layer**
    - **SQL injection**: Directly manipulating queries to extract or alter data.
    - **Data exfiltration**: Stealing sensitive records (e.g., credit card numbers).
    - **Ransomware**: Encrypting or corrupting stored data to demand payment.
5. **Response Flow Back**
    - **Man-in-the-middle (MITM)**: Intercepting data between server and user if encryption is weak.
    - **Replay attacks**: Reusing intercepted requests to gain unauthorized access.

Cybersecurity is about protecting digital systems and data by ensuring confidentiality, integrity, and availability—the CIA triad. Confidentiality keeps information private, integrity ensures it remains accurate and unaltered, and availability guarantees access when needed. Different attackers pose threats in unique ways: script kiddies use pre-made tools for mischief, insiders exploit legitimate access, hacktivists attack for political or social causes, and nation-state actors conduct sophisticated, government-backed operations. Common attack surfaces include web applications, mobile apps, APIs, networks, and cloud infrastructure, each vulnerable to exploits like injection attacks, insecure storage, or misconfigurations. Data typically flows from the user to the application, then to the server, and finally to the database before returning as a response, and attacks can occur at every stage—phishing at the user level, injection at the application, denial-of-service at the server, SQL injection at the database, and man-in-the-middle during response. Defenses require layered strategies such as encryption, strong authentication, secure coding, patching, monitoring, and strict access controls to protect the entire journey of data.