Task 1: Understanding Cyber Security Basics & Attack Surface

1. What is Cyber Security? (CIA Triad)

Cyber Security is the practice of protecting systems, networks, and data from digital attacks, unauthorized access, and damage.

It is based on the **CIA Triad**:

🔐 Confidentiality

Ensures that data is accessible **only to authorized users**.

Examples:

- **Banking apps**: Only you can see your account balance using login + OTP.
- **Social media**: Private messages are visible only to sender and receiver.

Threats: Data leaks, hacking, phishing
Controls: Passwords, encryption, access control

▯ Integrity

Ensures that data is **accurate and not altered** without permission.

Examples:

- Bank transaction amounts should not change.
- Social media posts should not be modified by others.

Threats: Data tampering, SQL injection
Controls: Hashing, checksums, validation

⏱ Availability

Ensures that systems and data are **available when needed**.

Examples:

- ATM servers must work 24/7.
- WhatsApp servers should be reachable anytime.

Threats: DDoS attacks, server crashes
Controls: Load balancers, backups, redundancy

---

2. Types of Attackers

👶 Script Kiddies

- Beginners using ready-made tools
- No deep technical knowledge
- Attack for fun or fame

Example: Running automated hacking tools

🧑‍💼 Insiders

- Employees or trusted users
- Have legitimate access
- Very dangerous because they know the system

Example: Employee leaking customer data

✊ Hacktivists

- Hack for political or social causes
- Want to spread a message

Example: Website defacement to protest

🏴 Nation-State Actors

- Government-backed hackers
- Highly skilled and well-funded

- Target critical infrastructure

Example: Attacks on power grids, defence systems

---

3. Common Attack Surfaces

An **attack surface** is any point where an attacker can try to enter.

🌐 Web Applications

- Login forms, search bars
- Vulnerable to SQL injection, XSS

📱 Mobile Applications

- Insecure storage
- Reverse engineering

🔌 APIs

- Weak authentication
- Data exposure

🖧 Networks

- Open ports
- Weak Wi-Fi security

☁ Cloud Infrastructure

- Misconfigured storage (public S3 buckets)
- Weak IAM policies

---

4. OWASP Top 10: 2025 — What They Are & Why They are Dangerous

1) A01:2025 — Broken Access Control

What it is: When an application doesn't correctly enforce *who can do what*. This includes weak permissions, insecure direct object references (IDOR), token

manipulation, or missing role checks.
Why it’s dangerous:

- **Attackers can access data or actions they shouldn’t — like viewing other users’ info or admin functions.**
- **It can lead to data breaches, privilege escalation, or unauthorized transactions.**
- **Even microservices and APIs often fail to check permissions consistently.**

---

2) A02:2025 — Security Misconfiguration

What it is: Incorrect or unsafe settings in the application or infrastructure — e.g., default credentials, open cloud storage, exposed developer consoles.
Why it’s dangerous:

- **Widely present in modern systems; almost every test shows misconfigurations.**
- **Exposed services or permissions give attackers easy entry points.**
- **Can lead to data exposure, system compromise, and unauthorized access.**

---

3) A03:2025 — Software Supply Chain Failures

What it is: Risks across software dependencies, build systems, libraries, CI/CD pipelines, and delivery tools — not just “outdated components.”
Why it’s dangerous:

- **A compromised third-party library or build tool can infect your app even if it’s secure.**
- **Attackers can inject malware into widely used packages or manipulate the software pipeline.**
- **These failures can be hard to detect and impact many systems simultaneously.**

---

4) A04:2025 — Cryptographic Failures

What it is: Weak, missing, or outdated encryption and poor key management.
Why it’s dangerous:

- **Sensitive data in transit or at rest can be intercepted or exposed.**

- **Weak randomization, insecure ciphers, or exposed keys make data easy to break.**
- **Attackers might steal credentials, financial details, or personal information.**

---

5) A05:2025 — Injection

What it is: Untrusted input being executed as code or commands (e.g., SQL injection or OS command injection).
Why it's dangerous:

- **Attackers can run malicious queries or commands on databases or servers.**
- **This often leads to data theft, data destruction, or full system takeover.**
- **Despite ongoing improvements, injection remains a common and impactful risk.**

---

6) A06:2025 — Insecure Design

What it is: Flaws in the application's architecture and logic rather than just implementation bugs.
Why it's dangerous:

- **These issues stem from poor threat modeling and unsafe workflows, so they are *built into* the system.**
- **They often persist through patches and require redesign.**
- **Attackers exploit business logic flaws, not just technical bugs.**

---

7) A07:2025 — Authentication Failures

What it is: Weak login controls, missing multi-factor authentication, unsafe session management.
Why it's dangerous:

- **Allows attackers to impersonate users or bypass login entirely.**
- **Poor policies (e.g., no MFA, weak passwords) make account takeover easier.**
- **Often leads to mass breaches when combined with credential theft.**

---

8) A08:2025 — Software or Data Integrity Failures

What it is: Systems failing to verify integrity of code, updates, or critical data (e.g., unsigned releases).
Why it's dangerous:

- **Attackers can inject malicious code via updates or tampered data.**
- **Compromised CI/CD pipelines or dependencies can quietly poison applications.**
- **This undermines trust and can lead to widespread malware.**

---

9) A09:2025 — Logging and Alerting Failures

What it is: Insufficient logging, monitoring, or alerting mechanisms.
Why it's dangerous:

- **Breaches can go undetected for long periods.**
- **Attackers operate stealthily, increasing damage before discovery.**
- **Delayed detection means higher costs and slower responses.**

---

10) A10:2025 — Mishandling of Exceptional Conditions

What it is: Improper handling of errors, edge cases, or unexpected conditions.
Why it's dangerous:

- **Poor error handling can reveal internal state or crash systems.**
- **Failing unsafe "open" can let attackers exploit logic gaps.**
- **Can cause denial of service, security bypasses, or information leaks.**

---

5. Mapping Daily-Use Apps to Attack Surfaces

Application	Possible Attack Surfaces
Email	Phishing, malware attachments

Application	Possible Attack Surfaces
WhatsApp	Account takeover, message interception
Banking App	API abuse, credential theft
Social Media	XSS, fake profiles, data scraping

---

6. Data Flow (User → App → Server → Database)

1. **User** enters data (login, message, payment)
2. **Application** processes the request
3. **Server** validates and handles logic
4. **Database** stores or retrieves data
5. Response goes back to user

---

7. Where Attacks Can Happen

- **User Side**: Phishing, keyloggers
- **Application**: XSS, insecure input validation
- **Server**: Misconfigurations, outdated software
- **Database**: SQL injection, unauthorized access
- **Network**: Man-in-the-middle attacks

---

8. Summary

Cyber security is about protecting data and systems using the principles of confidentiality, integrity, and availability. Different attackers have different motivations and skill levels, ranging from beginners to nation-state hackers. Modern applications expose many attack surfaces such as web apps, APIs, and cloud services. By understanding how data flows through systems and where vulnerabilities exist, we can identify points where attacks may occur. Awareness of common vulnerabilities like those listed in OWASP Top 10 helps in building secure systems and reducing risk.