ELEVATE LABS -CYBERSECURITY INTERNSHIP

TASK -1

Submitted By:-Sai Nanda Pavanan

- **Introduction to Cyber Security**

Cyber security refers to the strategic practice of protecting systems, networks, applications, and data from unauthorized access, misuse, disruption, or destruction. In today's digital-first ecosystem, cyber security is a business-critical enabler that ensures trust, continuity, and resilience across digital operations. At its core, cyber security is built on three foundational principles known as the CIA Triad: Confidentiality, Integrity, and Availability.

1. **Confidentiality**

Confidentiality ensures that sensitive information is accessed only by authorized individuals or systems. It focuses on preventing data breaches, leaks, and unauthorized disclosures.

- Real-world examples:

Banking: Customer data such as account numbers, PINs, and transaction details are protected using encryption, authentication, and access controls.

Social Media: Private messages, personal photos, and profile details are restricted so only intended users can view them.

Key controls: Encryption, multi-factor authentication (MFA), access control lists, and data classification.

2. **Integrity**

Integrity ensures that data remains accurate, complete, and unaltered throughout its lifecycle. It prevents unauthorized modification or manipulation of information.

- Real-world examples:

Banking: Transaction records must not be altered during or after processing to avoid fraud or financial loss.

Social Media: User posts, likes, and comments should not be modified by attackers to spread misinformation or damage reputations.

Key controls: Hashing, digital signatures, checksums, and audit logs.

3. **Availability**

Availability ensures that systems, services, and data are accessible to authorized users whenever required. It focuses on minimizing downtime and service disruptions.

- Real-world examples:

Banking: Online banking platforms must remain operational 24/7 to support transactions and customer access.

Social Media: Platforms must handle high traffic and defend against attacks like Distributed Denial of Service (DDoS) to remain accessible.

Key controls: Redundancy, backups, load balancing, disaster recovery, and incident response plans.

- **Types Of Attackers**

1. **Script Kiddies**

- Profile: Low-skill, high-noise attackers

Capability Level: Entry-level

Motivation: Curiosity, clout, bragging rights, boredom

- Key Characteristics:

Rely on pre-built tools, exploit kits, and public GitHub scripts

Minimal understanding of how attacks actually work

Commonly target poorly secured websites, game servers, or Wi-Fi networks

- Typical Attacks:

Website defacement

Basic DdoS attacks

Exploiting known vulnerabilities without customization

- Real-World Impact:

Low strategic impact but can still cause service disruptions and reputational damage, especially for small organizations.

2. **Insider Threats**

- Profile: Trusted users turned risk vectors

Capability Level: Medium to high (context-dependent)

Motivation: Financial gain, revenge, coercion, negligence

- Key Characteristics:

Employees, contractors, or partners with legitimate access

Deep knowledge of internal systems and workflows

Hard to detect due to valid credentials

- Types of Insiders:

Malicious insiders: Intentionally cause harm

Negligent insiders: Cause breaches accidentally (phishing, weak passwords)

- Typical Attacks:

Data theft and exfiltration

Privilege abuse

Sabotage of systems or logs

Real-World Impact:

High. Insiders are responsible for some of the most damaging breaches due to trusted access.

3. **Hacktivists**

- Profile: Ideology-driven attackers

Capability Level: Medium

Motivation: Political, social, or ideological causes

- Key Characteristics:

Target governments, corporations, or institutions

Aim to expose, embarrass, or disrupt rather than profit

Often operate under loosely organized groups

- Typical Attacks:

Website defacement with political messages

Data leaks (doxxing)

Distributed Denial-of-Service (DdoS) campaigns

- Real-World Impact:

Moderate to high reputational damage; operational disruption during campaigns.

4. **Nation-State Actors**

- Profile: Government-backed cyber operators

Capability Level: Advanced / Elite

Motivation: Espionage, national security, economic advantage, cyber warfare

- Key Characteristics:

Highly skilled teams with significant funding

Use zero-day exploits and custom malware

Long-term, stealthy operations (APT – Advanced Persistent Threats)

- Typical Attacks:

Cyber espionage

Critical infrastructure attacks

Supply chain compromises

- Real-World Impact:

Severe. These attacks can impact national security, economies, and critical services.

- **Exploring Common Attack Surfaces**

1. Web Applications 🌐

Attack Surface Overview:

Web applications are publicly exposed and continuously interact with untrusted users, making them a prime target.

- Common Exposure Points:

Input fields, forms, and search boxes

Authentication and session management

File uploads and downloads

Third-party libraries and plugins

- Typical Attacks:

SQL Injection

Cross-Site Scripting (XSS)

Cross-Site Request Forgery (CSRF)

Broken authentication and access control

- Business Impact:

Data breaches, account takeovers, reputational damage, and compliance violations.

2. Mobile Applications 📱

- Attack Surface Overview:

Mobile apps operate in hostile environments where attackers control the device and network.

- Common Exposure Points:

Insecure local storage (SharedPreferences, SQLite)

Hardcoded API keys or credentials

Weak authentication logic

Unprotected inter-process communication

- Typical Attacks:

Reverse engineering and tampering

Man-in-the-Middle (MITM) attacks

Insecure data storage exploitation

Abuse of excessive permissions

- Business Impact:

Credential leakage, unauthorized API access, and loss of user trust.

3. APIs 🔌

- Attack Surface Overview:

APIs are the backbone of modern applications and often expose business logic directly.

- Common Exposure Points:

Public or poorly authenticated endpoints

Excessive data exposure in responses

Inadequate rate limiting

Improper object-level authorization

- Typical Attacks:

Broken Object Level Authorization (BOLA)

API parameter tampering

Credential stuffing

Denial of Service (DoS)

- Business Impact:

Mass data extraction, service disruption, and systemic abuse at scale.

5. **Networks 🌐**

- Attack Surface Overview:

Networks connect systems internally and externally, making misconfigurations extremely costly.

- Common Exposure Points:

Open ports and exposed services

Weak firewall and router configurations

Unsecured Wi-Fi networks

Legacy protocols

- Typical Attacks:

Network sniffing and eavesdropping

Man-in-the-Middle (MITM)

Lateral movement

Distributed Denial of Service (DdoS)

- Business Impact:

Infrastructure compromise, service downtime, and pivot attacks across the environment.

6. **Cloud Infrastructure ☁**

- Attack Surface Overview:

Cloud environments introduce shared responsibility and configuration-driven risk.

- Common Exposure Points:

Misconfigured storage buckets

Over-privileged IAM roles

Insecure virtual machines and containers

Exposed management interfaces

- Typical Attacks:

Cloud misconfiguration abuse

Credential theft and privilege escalation

Container escape

Supply chain attacks

- Business Impact:

Large-scale data exposure, financial loss, and regulatory penalties.

- **OWASP TOP 10 ATTACKS**

1. **Broken Access Control**

Why it’s dangerous:

Attackers can bypass authorization and perform actions they should never be allowed to do—like accessing other users’ data or admin functions.

Risk Reality:

Privilege escalation, data leaks, full system compromise.

Bottom line: If access control fails, trust boundaries collapse.

2. **Cryptographic Failures**

Why it’s dangerous:

Sensitive data (passwords, financial info, personal data) is exposed due to weak or missing encryption.

Risk Reality:

Plaintext data interception, regulatory violations, identity theft.

Bottom line: Encryption mistakes turn private data into public assets.

3. **Injection**

Why it’s dangerous:

Untrusted input is interpreted as commands by the backend (SQL, OS, LDAP).

Risk Reality:

Database dumps, remote command execution, total application takeover.

Bottom line: The attacker speaks directly to your system—no filter.

4. **Insecure Design**

Why it’s dangerous:

Security was never embedded at the architecture level, so flaws are systemic and hard to patch.

Risk Reality:

Even “patched” apps remain exploitable due to flawed logic.

Bottom line: You can’t fix broken design with code-level patches.

5. **Security Misconfiguration**

Why it’s dangerous:

Default settings, open services, verbose error messages expose internal workings.

Risk Reality:

Easy wins for attackers—low effort, high impact.

Bottom line: Misconfigurations are silent enablers of breaches.

6. **Vulnerable and Outdated Components**

Why it’s dangerous:

Third-party libraries may contain known exploits attackers actively scan for.

Risk Reality:

Supply chain attacks, mass exploitation, zero-day reuse.

Bottom line: Your security is only as strong as your weakest dependency.

7. **Identification and Authentication Failures**

Why it’s dangerous:

Weak login systems allow attackers to impersonate users.

Risk Reality:

Account takeovers, credential stuffing, session hijacking.

Bottom line: If identity breaks, everything else becomes irrelevant.

8. **Software and Data Integrity Failures**

Why it’s dangerous:

Applications trust unverified updates, plugins, or CI/CD pipelines.

Risk Reality:

Malicious code execution, backdoored updates, large-scale compromise.

Bottom line: Trust without verification equals instant breach.

9. **Security Logging and Monitoring Failures**

Why it’s dangerous:

Attacks happen—but no one notices in time.

Risk Reality:

Long dwell time, undetected breaches, delayed incident response.

Bottom line: If you can’t see it, you can’t stop it.

10. **Server-Side Request Forgery (SSRF)**

Why it’s dangerous:

Attackers force the server to make requests on their behalf, often to internal systems.

Risk Reality:

Cloud metadata theft, internal network access, credential leakage.

Bottom line: The server becomes the attacker’s proxy.

Strategic Takeaway 💡

OWASP Top 10 vulnerabilities are dangerous because they:

Target core trust assumptions

Scale easily across systems

Combine technical impact with direct business risk

- **Mapping the attack surfaces of daily used applications.**

1. **Email Applications (Gmail, Outlook, Yahoo)**

- Web Interface

Description: Accessed via browser, often targeted by phishing or malicious links.

Threat Vectors: Phishing, credential harvesting, session hijacking.

- Mobile Application

Description: Outdated or compromised apps can leak credentials or session tokens.

Threat Vectors: Malware, unauthorized access, keylogging.

- Email Attachments

Description: Files may carry malware or exploit application vulnerabilities.

Threat Vectors: Ransomware, trojans, macro-based attacks.

- Embedded Links

Description: Redirect users to fake login portals or malicious websites.

Threat Vectors: Social engineering, drive-by downloads, phishing.

- Authentication Mechanisms

Description: Weak passwords or lack of multi-factor authentication.

Threat Vectors: Account takeover, brute-force attacks, credential stuffing.

2. **WhatsApp / Messaging Applications**

- Mobile Application

Description: App vulnerabilities can be exploited via malicious messages.

Threat Vectors: Exploitation of zero-day bugs, remote code execution.

- Media File Handling

Description: Images, videos, or documents can carry malicious payloads.

Threat Vectors: Malware injection, buffer overflow attacks.

- Contact Synchronization

Description: Access to contacts may expose personal information.

Threat Vectors: Privacy leakage, targeted phishing, social engineering.

- Cloud Backups

Description: Unprotected backups expose chat history and media.

Threat Vectors: Unauthorized data access, cloud account compromise.

- Account Verification (SIM/OTP)

Description: SIM-swap or OTP interception can hijack accounts.

Threat Vectors: Account takeover, identity impersonation.

3. **Banking Applications**

- Mobile Application Interface

Description: Fake or tampered apps can steal credentials or manipulate transactions.

Threat Vectors: Phishing apps, overlay attacks, malware injection.

- Backend APIs

Description: Improperly secured APIs can expose sensitive financial operations.

Threat Vectors: API abuse, broken object-level authorization (BOLA), data exfiltration.

- Authentication (PIN, OTP, Biometrics)

Description: Weak or bypassed authentication enables unauthorized account access.

Threat Vectors: Credential theft, brute-force, SIM swap attacks.

- Network Communication

Description: Insecure networks can expose sensitive data in transit.

Threat Vectors: Man-in-the-Middle (MITM), packet sniffing, session hijacking.

- Local Storage

Description: Sensitive data stored on devices may be accessed by malware.

Threat Vectors: Device compromise, keylogging, malware injection.

- **Documenting the Data Flow**

1. **User → Application (Client Layer)**

The user initiates interaction by entering data (login credentials, messages, transaction details) through a web browser or mobile app.

This data is captured via UI components such as forms, input fields, or buttons.

- Key Activities:

Data input and validation (basic checks)

- Session initiation

Data packaging (JSON/XML)

- Security Relevance:

This is the primary trust boundary. Poor input validation here enables injection, XSS, and client-side manipulation.

2. **Application → Server (Network & API Layer)**

The application sends user data to the backend server over the network, typically using HTTPS via REST or GraphQL APIs.

- Key Activities:

Encryption in transit (TLS)

API request handling

Authentication token validation (JWT, session ID)

- Security Relevance:

If transport security or API authentication is weak, attackers can intercept, replay, or manipulate requests.

3. **Server → Application Logic (Backend Processing)**

The server processes incoming requests using business logic and access control rules.

Key Activities:

Authorization checks

Business rule enforcement

Data sanitization

Query construction

Security Relevance:

Flaws here lead to broken access control, logic abuse, and injection vulnerabilities.

4. **Server → Database (Data Persistence Layer)**

The backend communicates with the database to read or write data using queries.

Key Activities:

Query execution (SELECT, INSERT, UPDATE)

Transaction handling

Data storage and retrieval

Security Relevance:

Unsafe queries or excessive privileges can result in SQL injection, data leakage, or full database compromise.

5. **Database → Server → Application → User (Response Flow)**

The database returns results to the server, which formats the response and sends it back to the application.

The application then renders the data to the user.

Key Activities:

Data filtering

Response formatting

UI rendering

Security Relevance:

Improper output encoding can expose sensitive data or enable reflected attacks like XSS.

- **Identifying the Attack Surfaces during the flow**

1. **User → Application (Client Layer)**

Where attacks happen:

Input fields (login, forms, search boxes)

File uploads or attachments

User interface components

Potential Attacks:

Phishing / social engineering: Trick users into entering sensitive data

Client-side script attacks (XSS): Inject scripts via inputs

Malware on device: Keylogging, credential theft

Input manipulation: Tampering with form data before submission

2. **Application → Server (Network & API Layer)**

Where attacks happen:

API endpoints

Network transport (HTTP/HTTPS)

Tokens and session identifiers

Potential Attacks:

Man-in-the-Middle (MITM): Intercept or modify data in transit

Replay attacks: Resend intercepted requests

API abuse / parameter tampering: Access unauthorized resources

Broken authentication: Stolen session tokens or weak token management

3. **Server → Application Logic (Backend Processing)**

Where attacks happen:

Server-side input handling and business logic

Authorization checks

Data sanitization and validation routines

Potential Attacks:

SQL / command injection: Improper input passed to queries or system commands

Broken access control: Users perform actions they shouldn’t

Logic flaws: Abuse of application rules (e.g., bypassing transaction limits)

Remote code execution (RCE): Exploiting server vulnerabilities

4. **Server → Database (Data Persistence Layer)**

Where attacks happen:

Database queries

Stored procedures

Privilege escalation points

Potential Attacks:

SQL Injection: Manipulate queries to extract or delete data

Privilege escalation: Exploit overly-permissive accounts

Data exfiltration: Unauthorized reading of sensitive information

Denial of Service (DoS): Heavy queries or malformed requests crashing DB

5. **Database → Server → Application → User (Response Flow)**

Where attacks happen:

Output formatting and rendering

Response headers

Data returned to UI

Potential Attacks:

Reflected or stored XSS: Malicious database content executed in user browser

Sensitive data exposure: Sending unencrypted or unnecessary data

Mass assignment / overexposure: Returning fields users shouldn’t see

Executive Insight

Every stage in the flow is a potential attack vector. Attackers exploit weak trust boundaries, improper validation, or misconfigured access. A holistic security strategy must secure:

Client input

Network transport

Server logic and APIs

Database access controls

Output encoding

This approach ensures attackers can’t compromise the system at any single point in the data flow.

- **Summary**

Cybersecurity focuses on protecting systems, networks, and data from digital threats. Its foundation is the CIA Triad: Confidentiality, Integrity, and Availability. Confidentiality ensures only authorized users can access sensitive data, such as encrypted banking credentials. Integrity ensures data remains accurate and unaltered, like preventing unauthorized changes to transactions or messages. Availability ensures systems and services remain accessible, which is critical for platforms such as email, social media, and online banking.

Cyber threats originate from different attacker types. Script kiddies use ready-made tools with limited skills to exploit common vulnerabilities. Insiders misuse legitimate access, either intentionally or accidentally. Hacktivists attack systems to promote political or social causes, often targeting public-facing services. Nation-state actors are highly skilled and well-funded, conducting advanced attacks for espionage or strategic advantage.

Attack surfaces are the points where attackers can interact with a system. Common attack surfaces include web applications, mobile apps, APIs, networks, and cloud infrastructure. The OWASP Top 10 highlights critical risks such as injection attacks, broken authentication, insecure design, and misconfigurations, showing how weak security controls can lead to serious breaches.

Daily-used applications clearly map to these attack surfaces. Email systems are commonly targeted through phishing and malware. Messaging apps like WhatsApp face risks from social engineering and insecure APIs. Banking applications are targeted through credential theft, session hijacking, and transaction manipulation.

Data typically flows from the user to the application, then to the server and database. Attacks can occur at each stage, from phishing at the user level to injections and data breaches at the server and database layers. Understanding this flow helps organizations identify risk points and implement effective security controls.