**SCB3491-Cryptography and Cyber Security**

**PART-C (15 Marks)**

**UNIT-1 Introduction to Security**

**1.Explain in detail about various types of attacks.**

Cyberattacks can be classified into **active and passive attacks**.

**1. Passive Attacks (Monitoring & Eavesdropping)**

* **Objective:** Stealthily gather information without altering data.
* **Examples:**
  1. **Eavesdropping/Sniffing:** Capturing network traffic.
  2. **Traffic Analysis:** Monitoring patterns to infer sensitive details.

**Prevention:**  
✔ **Use encryption (TLS, VPNs)**.  
✔ Implement **secure authentication**.

**2. Active Attacks (Modification & Disruption)**

* **Objective:** Alter or disrupt systems actively.
* **Examples:**
  1. **Masquerade Attack:** Impersonating a legitimate user.
  2. **Denial-of-Service (DoS):** Flooding a system to crash it.
  3. **Man-in-the-Middle (MITM):** Intercepting & altering communications.
  4. **SQL Injection:** Injecting malicious SQL queries.
  5. **Ransomware Attack:** Encrypting data & demanding ransom.

**Prevention:**  
✔ Use **firewalls & intrusion detection systems (IDS)**.  
✔ **Patch vulnerabilities** regularly.  
✔ Implement **multi-factor authentication (MFA)**.

**3. Insider Attacks (Internal Threats)**

* **Caused by employees or trusted entities.**
* **Examples:**
  1. **Data Theft:** Employees stealing confidential files.
  2. **Privilege Misuse:** Using admin access for malicious actions.

**Prevention:**  
✔ **Role-based access control (RBAC)**.  
✔ **Audit logs & monitoring**.

**4. Web-Based Attacks**

1. **Cross-Site Scripting (XSS):** Injecting scripts into websites.
2. **Cross-Site Request Forgery (CSRF):** Forcing users to execute unintended actions.
3. **Phishing:** Fake emails/websites to steal credentials.

**Prevention:**  
✔ Use **input validation** and **sanitization**.  
✔ Enable **email filtering & anti-phishing tools**.

**2. Explain in detail about various services provided by X.800.**

X.800 defines **security services** to protect networks and communications.

**1. Authentication Services**

✔ Ensures identity verification of users/devices.  
✔ Types:

* **Peer Entity Authentication:** Validates ongoing communication sessions.
* **Data-Origin Authentication:** Confirms the sender's identity.

**2. Access Control**

✔ Prevents unauthorized access to systems.  
✔ Uses **passwords, biometrics, and access lists**.

**3. Data Confidentiality**

✔ Protects information from unauthorized access.  
✔ Uses **encryption techniques like AES & RSA**.  
✔ Types:

* **Connection Confidentiality:** Protects data over a communication channel.
* **Selective-Field Confidentiality:** Encrypts specific parts of a message.

**4. Data Integrity**

✔ Ensures data is not altered during transmission.  
✔ Uses **Hash Functions (SHA, MD5, HMAC)**.

**5. Non-Repudiation**

✔ Prevents sender/receiver from denying message transmission.  
✔ Uses **digital signatures & timestamps**.  
✔ Types:

* **Non-repudiation of Origin:** Proves sender sent the message.
* **Non-repudiation of Receipt:** Proves receiver got the message.

**6. Availability Services**

✔ Ensures data/system accessibility despite attacks.  
✔ Protection against **DoS & DDoS attacks**.

**3.Explain in detail about various Mechanisms provided by X.800.**

Security mechanisms enforce **X.800 security services**.

**1. Encipherment (Encryption)**

✔ Protects data confidentiality using encryption.  
✔ Types:

* **Symmetric Encryption (AES, DES):** Faster, uses a single key.
* **Asymmetric Encryption (RSA, ECC):** Secure, uses public/private keys.

**2. Digital Signatures**

✔ Ensures **authentication, integrity & non-repudiation**.  
✔ Uses **RSA, DSA, ECC** for signing messages.

**3. Access Control Mechanisms**

✔ Restricts unauthorized access via:

* **User authentication (passwords, biometrics)**.
* **Role-Based Access Control (RBAC)**.

**4. Data Integrity Mechanisms**

✔ Prevents unauthorized data modification.  
✔ Uses **hash functions (SHA-256, HMAC, MD5)**.

**5. Authentication Exchange**

✔ Uses **challenge-response authentication** (e.g., Kerberos, OTPs).

**6. Traffic Padding**

✔ Adds dummy data to hide real transmission patterns.

**7. Routing Control**

✔ Changes network paths dynamically to avoid attacks.

**8. Notarization**

✔ Uses a **trusted third party (TTP)** for authentication & verification.

**UNIT-II SYMMETRIC CIPHERS**

**1.Explain the block cipher modes of operation.**

Block ciphers encrypt fixed-size blocks of data (e.g., **AES** uses **128-bit blocks**). Different modes define how plaintext is processed.

**1. Electronic Codebook (ECB) Mode**

✔ **Each block is encrypted independently**.  
✔ **Fast but insecure** for large data.

**Example:**

makefile

CopyEdit

Plaintext: A B A B

Ciphertext: X Y X Y (Same plaintext → Same ciphertext)

🔴 **Weakness:** Identical blocks result in identical ciphertext → vulnerable to pattern analysis.

**2. Cipher Block Chaining (CBC) Mode**

✔ **Each plaintext block is XORed with the previous ciphertext block** before encryption.  
✔ **Uses an Initialization Vector (IV)** to prevent identical encryption for the same input.

**Example:**

ini

CopyEdit

C1 = Encrypt(P1 ⊕ IV)

C2 = Encrypt(P2 ⊕ C1)

🟢 **Advantage:** Hides plaintext patterns.  
🔴 **Weakness:** Requires IV; errors propagate.

**3. Cipher Feedback (CFB) Mode**

✔ **Converts a block cipher into a stream cipher**.  
✔ **Uses IV and previous ciphertext to generate keystream**.

**Example:**

scss

CopyEdit

Encrypt(IV) → Output XOR P1 → C1

Encrypt(C1) → Output XOR P2 → C2

🟢 **Advantage:** Works for any data size.  
🔴 **Weakness:** Sensitive to bit errors.

**4. Output Feedback (OFB) Mode**

✔ **Generates a keystream using encryption but without feedback loops**.  
✔ **More resistant to transmission errors**.

🟢 **Advantage:** No error propagation.  
🔴 **Weakness:** Vulnerable to replay attacks if IV is reused.

**5. Counter (CTR) Mode**

✔ Uses a **counter** as input for encryption.  
✔ Generates a keystream that **XORs with plaintext**.

🟢 **Advantages:**

* Supports **parallel encryption**.
* No error propagation.  
  🔴 **Weakness:** Reusing counter values leaks plaintext.

**Summary of Modes:**

| **Mode** | **Advantages** | **Disadvantages** |
| --- | --- | --- |
| **ECB** | Fast, simple | No randomness, insecure |
| **CBC** | Good randomness | Slow, error propagation |
| **CFB** | Stream cipher support | Error-sensitive |
| **OFB** | No error propagation | Reusing IV is risky |
| **CTR** | Fast, parallelizable | Must use unique counters |

2.**Explain the key management of public key encryption in detail.**

Key management is **essential for secure cryptography**. Public-key encryption (e.g., RSA, ECC) requires:

1. **Key Generation**
2. **Key Distribution**
3. **Key Storage & Usage**
4. **Key Revocation**

**1. Key Generation**

✔ Public/private key pairs are generated using **RSA, ECC, or Diffie-Hellman**.  
✔ Example (RSA):

* **Select two large prime numbers**: p,qp, qp,q
* Compute n=p×qn = p \times qn=p×q, **public key (e,n)(e, n)(e,n), private key (d,n)(d, n)(d,n)**.

**2. Key Distribution**

🔹 **Methods to share public keys:**

1. **Public Key Infrastructure (PKI)** – Uses a **Certificate Authority (CA)**.
2. **Web of Trust** – Users manually verify identities.
3. **Key Exchange Protocols** – **Diffie-Hellman (DH)** for secure key sharing.

**3. Key Storage & Usage**

✔ Private keys should be **stored securely** (e.g., Hardware Security Modules).  
✔ Public keys are **shared openly but must be authenticated**.

**4. Key Revocation & Expiry**

✔ Keys must be **revoked if compromised**.  
✔ Certificate Revocation Lists (CRLs) & Online Certificate Status Protocol (OCSP) are used.

**3.** **Explain ECC - Diffie Hellman key Exchange with both keys in detail with an example.**

**1. What is ECC-Diffie Hellman?**

✔ **Elliptic Curve Cryptography (ECC)** is used for **key exchange**.  
✔ Uses **elliptic curve math** for security with **smaller key sizes** than RSA.

**2. How ECC-DH Works**

1. **Public Curve Parameters**:
   * Choose an elliptic curve equation: y2=x3+ax+by^2 = x^3 + ax + by2=x3+ax+b over a finite field.
   * Select a **base point (G)** on the curve.
2. **Key Pair Generation:**
   * Alice chooses **private key**: aaa
   * Computes **public key**: PA=a×GP\_A = a \times GPA​=a×G
   * Bob chooses **private key**: bbb
   * Computes **public key**: PB=b×GP\_B = b \times GPB​=b×G
3. **Key Exchange:**
   * Alice computes **shared key**: S=a×PBS = a \times P\_BS=a×PB​
   * Bob computes **shared key**: S=b×PAS = b \times P\_AS=b×PA​
   * Both obtain the same key SSS, since: S=a×(b×G)=b×(a×G)S = a \times (b \times G) = b \times (a \times G)S=a×(b×G)=b×(a×G)

**3. Example Calculation**

✔ **Elliptic Curve**: y2=x3+2x+3y^2 = x^3 + 2x + 3y2=x3+2x+3 over GF(23)GF(23)GF(23)  
✔ **Base Point (G)** = (5, 19)  
✔ **Alice’s Private Key**: a=7a = 7a=7  
✔ **Bob’s Private Key**: b=11b = 11b=11

**Public Keys:**

* Alice: PA=7×G=(XA,YA)P\_A = 7 \times G = (X\_A, Y\_A)PA​=7×G=(XA​,YA​)
* Bob: PB=11×G=(XB,YB)P\_B = 11 \times G = (X\_B, Y\_B)PB​=11×G=(XB​,YB​)

**Shared Secret:**

* Alice computes: S=7×PBS = 7 \times P\_BS=7×PB​
* Bob computes: S=11×PAS = 11 \times P\_AS=11×PA​
* Both get the same shared secret!

**4. Advantages of ECC-DH**

✔ **Smaller key sizes** (256-bit ECC ~ 3072-bit RSA)  
✔ **More secure** than traditional DH  
✔ **Faster & efficient for mobile devices**

**UNIT-III ASYMMETRIC CIPHERS**

**1.Explain RSA algorithm in detail with an example?**

The **RSA algorithm** is an **asymmetric encryption** technique used for secure data transmission. It uses **two keys**:

* **Public Key** (for encryption)
* **Private Key** (for decryption)

**1. Steps of RSA Algorithm**

✔ **Step 1: Key Generation**

1. **Select two prime numbers:** ppp and qqq
2. Compute **modulus nnn:** n=p×qn = p \times qn=p×q
3. Compute **Euler's totient function ϕ(n)\phi(n)ϕ(n):** ϕ(n)=(p−1)×(q−1)\phi(n) = (p - 1) \times (q - 1)ϕ(n)=(p−1)×(q−1)
4. Choose an **encryption key eee** such that: 1<e<ϕ(n),gcd⁡(e,ϕ(n))=11 < e < \phi(n), \quad \gcd(e, \phi(n)) = 11<e<ϕ(n),gcd(e,ϕ(n))=1
5. Compute **decryption key ddd** using modular inverse: d=e−1mod  ϕ(n)d = e^{-1} \mod \phi(n)d=e−1modϕ(n)

✔ **Step 2: Encryption**

* Convert **plaintext (M)** into a numeric value.
* Compute ciphertext: C=Memod  nC = M^e \mod nC=Memodn

✔ **Step 3: Decryption**

* Compute plaintext from ciphertext: M=Cdmod  nM = C^d \mod nM=Cdmodn

**2. Example of RSA Algorithm**

✔ **Step 1: Key Generation**

* Select **prime numbers**: p=7,q=17p = 7, q = 17p=7,q=17
* Compute **n**: n=7×17=119n = 7 \times 17 = 119n=7×17=119
* Compute **ϕ(n)\phi(n)ϕ(n)**: ϕ(119)=(7−1)×(17−1)=96\phi(119) = (7-1) \times (17-1) = 96ϕ(119)=(7−1)×(17−1)=96
* Choose e=5e = 5e=5 (since **gcd(5, 96) = 1**)
* Compute **ddd**: d=5−1mod  96=77d = 5^{-1} \mod 96 = 77d=5−1mod96=77

✔ **Step 2: Encryption**

* Assume plaintext **M = 10**
* Compute **C**: C=105mod  119=100000mod  119=82C = 10^5 \mod 119 = 100000 \mod 119 = 82C=105mod119=100000mod119=82

✔ **Step 3: Decryption**

* Compute **M**: M=8277mod  119=10M = 82^{77} \mod 119 = 10M=8277mod119=10

✅ **Plaintext recovered successfully!**

**3. Advantages of RSA**

✔ **Strong security** (based on prime factorization).  
✔ **Used for digital signatures & secure communication.**  
✔ **No need to share a secret key (unlike symmetric encryption).**

🔴 **Weakness:** Slow for large data encryption.

**2.Briefly explain the idea behind Elliptic Curve Cryptosystem?**

**Elliptic Curve Cryptography (ECC)** is a **public-key encryption system** based on **elliptic curves over finite fields**.

**1. What is an Elliptic Curve?**

An **elliptic curve** is defined by the equation:

y2=x3+ax+b(mod p)y^2 = x^3 + ax + b \quad \text{(mod p)}y2=x3+ax+b(mod p)

where:

* a,ba, ba,b are constants.
* ppp is a prime number (finite field).
* The curve must satisfy: 4a3+27b2≠04a^3 + 27b^2 \neq 04a3+27b2=0.

**2. ECC Key Pair Generation**

1. Choose an elliptic curve y2=x3+ax+by^2 = x^3 + ax + by2=x3+ax+b.
2. Select a **base point GGG**.
3. Choose **private key kkk** (random number).
4. Compute **public key P=kGP = kGP=kG** using **point multiplication**.

**3. ECC Encryption & Decryption**

✔ **Encryption**

1. Sender picks a random number rrr.
2. Computes two points:
   * C1=rGC1 = rGC1=rG
   * C2=PM+rPC2 = P\_M + rPC2=PM​+rP (message point + shared secret).
3. Sends **(C1, C2)** to receiver.

✔ **Decryption**

1. Receiver computes: PM=C2−kC1P\_M = C2 - kC1PM​=C2−kC1 (where kC1=k(rG)=rPkC1 = k(rG) = rPkC1=k(rG)=rP, canceling out).

✅ Message is recovered!

**4. Example of ECC Key Exchange**

✔ Choose curve: **y2=x3+2x+3y^2 = x^3 + 2x + 3y2=x3+2x+3 (mod 23)**  
✔ Base point G=(5,19)G = (5,19)G=(5,19)  
✔ **Alice’s Private Key**: kA=7k\_A = 7kA​=7  
✔ **Bob’s Private Key**: kB=11k\_B = 11kB​=11  
✔ Compute public keys:

PA=7G,PB=11GP\_A = 7G, \quad P\_B = 11GPA​=7G,PB​=11G

✔ Shared secret:

S=kAPB=kBPAS = k\_A P\_B = k\_B P\_AS=kA​PB​=kB​PA​

✅ **ECC provides same security as RSA with smaller key sizes!**

**5. Advantages of ECC**

✔ **Smaller key sizes** (256-bit ECC ≈ 3072-bit RSA).  
✔ **Faster computation** than RSA.  
✔ **Ideal for mobile & IoT security**.

**3.Explain Data Encryption Standard (DES) in detail.**

DES is a **symmetric key block cipher** that encrypts **64-bit blocks** using a **56-bit key**.

**1. Steps of DES Algorithm**

✔ **Step 1: Key Generation**

* **Initial Permutation (IP):** Rearranges plaintext bits.
* **16 Rounds of Encryption:** Uses **Feistel structure** with substitution & permutation.

✔ **Step 2: Feistel Structure (Rounds 1-16)**

1. **Divide 64-bit input** into two **32-bit halves (L, R)**.
2. For **each round** (1-16):
   * Compute: Ri=Li−1⊕f(Ri−1,Ki)R\_i = L\_{i-1} \oplus f(R\_{i-1}, K\_i)Ri​=Li−1​⊕f(Ri−1​,Ki​)
   * Swap L,RL, RL,R.
3. **Final Permutation (FP)** produces ciphertext.

✔ **Step 3: Decryption**

* DES **decryption is identical** to encryption but with **keys in reverse order**.

**2. Example of DES Encryption**

✔ Assume **plaintext = "11001100 10101010"**  
✔ Use **56-bit key = "101110..."**  
✔ Perform **initial permutation**  
✔ **16 rounds of Feistel operations**  
✔ Apply **final permutation**  
✅ **Ciphertext produced!**

**3. Strengths & Weaknesses of DES**

✅ **Advantages:**  
✔ Simple & efficient for hardware implementation.  
✔ Used in ATM PIN encryption.

🔴 **Weaknesses:**  
✖ **56-bit key is too short** (vulnerable to brute-force attacks).  
✖ **Replaced by AES (Advanced Encryption Standard).**

**UNIT-IV INTEGRATION AND AUTHENTICATION ALGORITHMS**

**1.How AES is used for encryption/decryption? Discuss with example.**

**Advanced Encryption Standard (AES)** is a **symmetric key block cipher** used to encrypt **128-bit blocks** with **key sizes of 128, 192, or 256 bits**.

**1. Steps of AES Algorithm**

AES consists of **four operations** repeated in **multiple rounds** depending on the key size:

* **128-bit key → 10 rounds**
* **192-bit key → 12 rounds**
* **256-bit key → 14 rounds**

✔ **Step 1: Key Expansion**

* Generates multiple round keys from the original key.

✔ **Step 2: Initial Round**

1. **AddRoundKey** → XOR plaintext with the first round key.

✔ **Step 3: Main Rounds (9, 11, or 13 rounds based on key size)**  
Each round includes:

1. **SubBytes** → Byte substitution using S-Box.
2. **ShiftRows** → Rows of state matrix are shifted left.
3. **MixColumns** → Column mixing for diffusion.
4. **AddRoundKey** → XOR with a round key.

✔ **Step 4: Final Round**

1. **SubBytes**
2. **ShiftRows**
3. **AddRoundKey** (No MixColumns in the last round).

**2. AES Decryption**

* **Same steps as encryption but reversed** (inverse S-Box, inverse MixColumns, etc.).

**3. Example of AES Encryption**

✔ Assume **plaintext**: "HELLO123" (converted into a 128-bit binary).  
✔ **Key (128-bit)**: "AESSECRETKEY!".  
✔ **Apply 10 rounds of AES** encryption.  
✔ **Ciphertext generated**: "A8D4F56B23...".

✅ **Decryption follows the reverse process!**

**4. Advantages of AES**

✔ **Stronger than DES** (larger key size).  
✔ **Fast & efficient** for software/hardware implementation.  
✔ **Used in SSL, VPNs, and secure communication.**

🔴 **Weakness:** Vulnerable to **side-channel attacks** (if key management is weak).

**2. What is Kerberos? (15 Marks)**

**Kerberos** is a **network authentication protocol** that provides **secure authentication** in a client-server environment using a **trusted third party (TTP)**.

**1. How Kerberos Works?**

✔ Uses **symmetric encryption** (AES/DES) for authentication.  
✔ Prevents **password theft** by avoiding direct password transmission.  
✔ Uses a **Ticket-Granting System (TGS)**.

**2. Steps of Kerberos Authentication**

✔ **Step 1: Login & Request for Ticket**

* User sends **username** to **Authentication Server (AS)**.
* AS generates a **Ticket-Granting Ticket (TGT)** encrypted with a secret key.

✔ **Step 2: Request for Service Ticket**

* User sends the TGT to **Ticket-Granting Server (TGS)** to request access to a service.
* TGS generates a **Service Ticket (ST)** encrypted with the service’s key.

✔ **Step 3: Access to Service**

* User presents the **Service Ticket (ST)** to the service.
* If valid, access is granted!

**3. Example of Kerberos in Action**

✔ **Scenario:** User logs into a corporate network.  
✔ Kerberos verifies the identity and grants access without sending the **password**.  
✔ **Encryption ensures security** against password sniffing.

**4. Advantages of Kerberos**

✔ **Prevents password theft** (password never sent over the network).  
✔ **Supports mutual authentication** (client & server verify each other).  
✔ **Used in Windows Active Directory & secure enterprise authentication.**

🔴 **Weakness:**  
✖ If the **Kerberos server is compromised**, security is lost.

**9. Explain the format of the X.509 certificate.**

**X.509** is the standard format for **digital certificates** used in **SSL/TLS authentication**.

**1. Structure of an X.509 Certificate**

An **X.509 certificate** contains the following fields:

✔ **1. Version** → Specifies the X.509 version (V1, V2, or V3).  
✔ **2. Serial Number** → Unique number assigned by the certificate authority (CA).  
✔ **3. Signature Algorithm** → Algorithm used to sign the certificate (e.g., RSA, ECDSA).  
✔ **4. Issuer** → Name of the **Certificate Authority (CA)** that issued the certificate.  
✔ **5. Validity** → Contains:

* **Not Before Date** (start date).
* **Not After Date** (expiration date).  
  ✔ **6. Subject** → Name of the entity the certificate is issued to (e.g., a website domain).  
  ✔ **7. Public Key** → The entity’s public key used for encryption.  
  ✔ **8. Extensions (X.509 v3)** → Additional information like:
* **Key Usage** (e.g., encryption, digital signatures).
* **Subject Alternative Name (SAN)** (e.g., multiple domains).  
  ✔ **9. Digital Signature** → Signature of the issuing CA (verifies authenticity).

**2. Example of an X.509 Certificate**

yaml

CopyEdit

Certificate:

Data:

Version: 3

Serial Number: 123456789

Signature Algorithm: sha256WithRSAEncryption

Issuer: CN=Let's Encrypt, O=Internet Security Research Group

Validity:

Not Before: Mar 1 12:00:00 2024 GMT

Not After: Mar 1 12:00:00 2025 GMT

Subject: CN=example.com

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public-Key: (2048 bit)

Extensions:

Key Usage: Digital Signature, Key Encipherment

Subject Alternative Name: DNS:example.com, DNS:www.example.com

Signature Algorithm: sha256WithRSAEncryption

**3. Use of X.509 Certificates in Web Security**

✔ **Used in HTTPS for SSL/TLS encryption**.  
✔ Ensures that a website is **authentic & secure**.  
✔ Helps in **email encryption & digital signatures**.

🔴 **Weakness:**  
✖ If the **CA (Certificate Authority) is compromised**, certificates can be forged.

**UNIT-V CYBER CRIMES AND CYBER SECURITY**

**1.Assume when an attacker tries to modify the database content by inserting an UPDATE statement. Identify this SQL injection attack method and modify and justify. Detail the methods used to prevent SQL injection attack.**

SQL injection occurs when an attacker **injects malicious SQL queries** to manipulate a database.

**Example Attack:**

sql

CopyEdit

Input: ' OR '1'='1'; --

Query: SELECT \* FROM users WHERE username = '' OR '1'='1';

**Impact:** Returns all records, bypassing authentication.

**Types of SQL Injection:**

1. **Error-Based Injection:** Uses error messages to extract data.
2. **Union-Based Injection:** Merges malicious results using UNION.
3. **Blind SQL Injection:** No visible output, uses true/false conditions.

**Prevention Methods:**

✔ **Use Prepared Statements (Parameterized Queries):**

python

CopyEdit

cursor.execute("SELECT \* FROM users WHERE username = ? AND password = ?", (user, pwd))

✔ **Use ORM (e.g., SQLAlchemy, Hibernate).**  
✔ **Validate User Input & Escape Special Characters.**  
✔ **Use Web Application Firewalls (WAFs).**  
✔ **Limit Database Privileges & Disable Error Messages.**

**2. What is keylogger? Explain types of keyloggers with its advantages and disadvantages.**

A **keylogger** is malware that records keystrokes to steal sensitive data.

**Types of Keyloggers:**

1. **Hardware Keylogger:**
   * **USB or keyboard circuit-based.**
   * **Advantage:** Hard to detect.
   * **Disadvantage:** Physical access required.
2. **Software Keylogger:**
   * Installed on a system to record keystrokes.
   * **Advantage:** Can be remotely installed.
   * **Disadvantage:** Detectable by antivirus.
3. **Rootkit Keylogger:**
   * Embedded in the OS to evade detection.
   * **Advantage:** Stealthy and hard to remove.
   * **Disadvantage:** Requires advanced knowledge to develop.
4. **Clipboard & Screen Recorders:**
   * Captures copied text and screenshots.

**Prevention:**

✔ Use **antivirus & anti-malware tools**.  
✔ Enable **two-factor authentication (2FA)**.  
✔ Use **virtual keyboards** for sensitive input.

**3. Write short notes on network access control.**

NAC is a **security framework** that regulates device access to a network.

**Key Features:**

1. **Authentication & Authorization:**
   * Uses **passwords, biometrics, or certificates**.
2. **Endpoint Security Checks:**
   * Ensures devices comply with security policies.
3. **Role-Based Access Control:**
   * Limits network access based on user roles.
4. **Quarantine Non-Compliant Devices:**
   * Prevents infected devices from joining the network.

**Types of NAC:**

1. **Agent-Based NAC:** Installed software checks security status.
2. **Agentless NAC:** Uses network-based authentication.
3. **Inline vs. Out-of-Band NAC:** Direct control vs. monitoring mode.

✔ **Example:** Used in **corporate networks, banking systems**.

|  |  |
| --- | --- |
|  |  |