Protection in Operating Systems

1. Goals of Protection

Protection in an operating system ensures controlled access to system resources, preventing unauthorized access, modification, or misuse. The primary goals of protection include:

- **Confidentiality**: Ensure that information is only accessible to authorized users.
- **Integrity**: Prevent unauthorized modifications to data.
- **Availability**: Ensure that resources are available to authorized users when needed.
- **Fairness**: Prevent any single process from monopolizing system resources.
- **Isolation**: Protect processes from interfering with one another.
- **Security**: Defend against malicious activities like hacking or malware attacks.

2. Domain of Protection

A protection domain defines the **set of access rights** a process or user has within a system.

- **Each domain specifies**:
  - The **objects** it can access (e.g., files, devices).
  - The **rights** it has (e.g., read, write, execute).
- A domain can be:
  - **User-based** (rights assigned per user).
  - **Process-based** (each process has a separate domain).
  - **Procedure-based** (rights depend on the executing code segment).

Domain Switching

A process may switch between domains under controlled conditions, providing flexibility while maintaining security.

3. Access Matrix

The **Access Matrix** is a conceptual model used to represent the rights of subjects (users, processes) over objects (files, devices).

	File1	File2	Printer
User1	R/W	R	-
User2	-	W	Print

- **Rows** represent subjects (users or processes).
- **Columns** represent objects (files, devices, etc.).
- **Cells** specify allowed actions (Read, Write, Execute, etc.).

4. Access Control

Access control mechanisms enforce the access matrix by ensuring users and processes only access authorized resources.

Techniques for Access Control:

1. **Access Control Lists (ACLs):**
    - Stored **per object**, listing users with corresponding permissions.
    - Example: A file might have an ACL specifying User1 can read/write, while User2 can on1ly read.
2. **Capability Lists:**
    - Stored **per subject**, listing objects the subject can access.
    - Example: A process holds capabilities allowing it to access certain files.
3. **Role-Based Access Control (RBAC):**
    - Users are assigned roles, and permissions are granted based on roles.
    - Example: An admin role has access to all files, while a guest role has restricted access.

5. Implementation of Access Matrix

Since a full access matrix is impractical to store, it is implemented in one of the following ways:

1. Access Control List (ACL)

- Each object maintains a list of permitted users and their access rights.
- Efficient for systems with **few users per object**.

2. Capability List

- Each subject maintains a list of objects and allowed operations.
- Used in distributed and decentralized systems.

3. Lock-Key Mechanism

- Each object has **locks** and each domain has **keys**.
- Access is granted if the key matches the lock.

6. Revocation of Access Rights

Revocation of access rights is necessary when users or processes should no longer have access to certain resources.

Types of Revocation:

1. **Immediate vs Delayed**: Rights are revoked instantly or after a delay.
2. **Selective vs General**: Specific users vs. all users lose access.
3. **Partial vs Total**: Only some rights or all rights are revoked.
4. **Temporary vs Permanent**: Access is removed for a limited time or permanently.

Mechanisms for Revocation:

- Updating ACLs (removing users from the list).
- Updating Capability Lists (revoking rights from users).
- Using tokens or timestamps for temporary revocation.

Protection – Class Notes

1. Goals of Protection

Protection refers to mechanisms for **controlling access** to computer system resources.

Primary Goals:

- **Confidentiality**: Ensure information is not accessed by unauthorized users.
- **Integrity**: Ensure information is not altered by unauthorized users.
- **Availability**: Ensure that resources are available to authorized users.
- **Fairness**: Prevent one user/process from monopolizing resources.
- **Isolation**: Ensure one process does not interfere with another.
- **Security**: Protect against malicious access or attacks.

2. Domain of Protection

A **domain** defines a **set of access rights** that a process or user has.

- A domain can be:
    - **User-based** (per user)
    - **Process-based** (per process)
    - **Procedure-based** (per code segment)

Each domain specifies:

- **Objects** (resources like files, devices)
- **Rights** (read, write, execute, etc.)

Domain Switching:

- Allows a process to switch from one domain to another.
- Provides flexibility but requires careful control.

3. Access Matrix

A **conceptual model** for defining and enforcing access control.

Structure:

- **Rows = Subjects (users, processes)**
- **Columns = Objects (files, printers, etc.)**
- **Cells = Set of rights** (e.g., read, write, execute)

	File1	File2	Printer
User1	R/W	R	-
User2	-	W	Print

Common Rights:

- **Read (R)**
- **Write (W)**
- **Execute (X)**
- **Delete**
- **Print**
- **Transfer (change rights)**

4. Access Control

Access control is the **mechanism** by which the **access matrix is enforced**.

Techniques:

- **Access Control Lists (ACLs)** – For each object, list who has what rights.
- **Capability Lists** – For each subject, list what rights it has on which objects.

- **Role-Based Access Control (RBAC)** – Permissions are associated with roles, and users are assigned roles.

5. Implementation of Access Matrix

Since the full matrix is large and sparse, it's implemented in practical systems using:

1. Access Control List (ACL):

- Stored **per object**.
- Lists all subjects with permitted access.

2. Capability List:

- Stored **per subject**.
- Lists all objects the subject can access and how.

3. Lock-Key Mechanism:

- Each object has a list of **locks**.
- Each domain has a list of **keys**.
- Access is granted if a key matches a lock.

6. Revocation of Access Rights

Sometimes it's necessary to **remove or revoke access** from users/domains.

Types of Revocation:

1. **Immediate vs Delayed**: Rights are removed instantly or after a delay.
2. **Selective vs General**: Revoke rights from a specific subject or all.
3. **Partial vs Total**: Remove some rights or all.
4. **Temporary vs Permanent**: Revoke for a period or indefinitely.

Mechanisms:

- Updating ACLs or capability lists.
- Removing keys in lock-key systems.
- Using timestamps or revocation tokens in distributed systems.

🔐 1. Security Problems in Computing

Security problems arise due to vulnerabilities in software, hardware, and user behavior.

Common Security Problems:

- **Unauthorized Access** – Gaining access to systems or data without permission.
- **Data Breaches** – Exposure of confidential data to unauthorized parties.
- **Data Modification** – Alteration of data by unauthorized users.
- **Denial of Service (DoS)** – Preventing legitimate users from accessing services.
- **Malware** – Malicious software like viruses, worms, and Trojans.
- **Insider Threats** – Security threats originating from within the organization.

2. Authentication

Authentication verifies the identity of a user, device, or entity.

Types of Authentication:

- **Password-Based Authentication** – Common but vulnerable to attacks.
- **Two-Factor Authentication (2FA)** – Combines password + another factor (OTP, biometric).
- **Biometric Authentication** – Uses fingerprints, retina scans, etc.
- **Token-Based Authentication** – Digital tokens or physical devices.
- **Certificate-Based Authentication** – Digital certificates issued by Certificate Authorities.

3. Program Threats

Program threats are malicious code inserted into software.

Types of Program Threats:

- **Trojan Horse** – Malicious code disguised as legitimate software.
- **Trapdoor (Backdoor)** – Secret entry points into programs.
- **Logic Bomb** – Malicious code activated by specific conditions.
- **Virus** – Self-replicating code that attaches to files/programs.
- **Worm** – A standalone malware that replicates itself to spread to other systems.

4. System and Network Threats

Threats targeting entire systems or networks.

System Threats:

- **Rootkits** – Software that hides the presence of malware.
- **Privilege Escalation** – Exploiting bugs to gain higher access rights.

Network Threats:

- **Sniffing** – Capturing data packets from a network.
- **Spoofing** – Pretending to be another user or device.
- **Man-in-the-Middle (MitM)** – Intercepting communication between two parties.
- **Denial of Service (DoS)** – Overloading a system to crash or slow it.
- **Distributed DoS (DDoS)** – DoS attack from multiple sources.

5. Encryption

Encryption is converting data into a form that only authorized parties can understand.

Types of Encryption:

- **Symmetric Encryption** – Same key for encryption and decryption (e.g., AES, DES).
- **Asymmetric Encryption** – Public and private key pairs (e.g., RSA, ECC).
- **Hashing** – One-way transformation used for integrity checks (e.g., SHA-256).
- **Digital Signatures** – Ensure authenticity and integrity of messages.

6. Computer Security Classification

Classifies security based on level of protection.

Classification Models:

- **Bell-LaPadula Model** – Focuses on data confidentiality.
  - *Simple Security Property* – "No read up" (user cannot read data at a higher level).
  - *Star Property* – "No write down" (user cannot write data to a lower level).
- **Biba Model** – Focuses on data integrity.
  - *Simple Integrity Property* – "No read down"
  - *Star Integrity Property* – "No write up"
- **Clark-Wilson Model** – Ensures well-formed transactions and separation of duties.