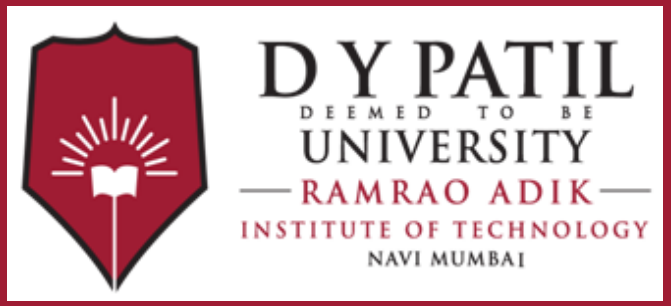


Subject Name: Information System

Unit No:02 Unit Name: Access Control Models

Faculty: Mrs. Bhavana Alte

Mr. Prathmesh Gunjgur

Discretionary Access Control (DAC), Mandatory Access Control (MAC)

What is Discretionary Access Control (DAC)?

•DAC is an access control model where the **owner** of a resource (e.g., file, system, database) has full discretion over who can access it and what actions they can perform on it (read, write, execute).

•**The owner has the ability to grant or revoke access permissions** to other users, making DAC more flexible but potentially less secure.

•***Definition***:*In DAC, the owner of the object (such as a file or folder) determines who can access the object and what operations they can perform (e.g., read, write, delete).*

Key Characteristics of DAC

•**Owner Control**:
•The **owner** has control over the resource and can decide who is allowed to access it.

•**Flexible Permissions**:
•Permissions can be set for individuals or groups, and these permissions can be dynamically changed by the owner.

•**Access Control List (ACL)**:
• It is a list of users (or groups of users) associated with an object (e.g., a file, folder, database) and the permissions granted to them. ACLs define who can access the resource and what actions they can perform on it.
•Folder: Project Files
•----------
•User: Alice - Read, Write, Execute
•User: Bob - Read
•User: Carol - Read, Write
•Group: Managers - Read, Write
•User: Dave - No Access

How DAC Works…

- **Resource Owner**:

•The owner is the person who creates or controls the resource.

•Example: A file created by a user in a system.

- **Permissions**:

•The owner defines permissions for other users, which could include **read**, **write**, or **execute** access.

•**Permissions can be set as follows**:

- •**Read (R)**: Allows viewing the content.
- •**Write (W)**: Allows modifying the content.
- •**Execute (X)**: Allows running the file if it's an executable or script.

How DAC Works…

- **Access Control List (ACL)**:Each object (e.g., file) has an associated ACL that specifies which users or groups have access to it and what actions they can perform.
- Example of an ACL for a file
  - File: report.txt
  - User: Alice - Read, Write
  - User: Bob - Read
  - Group: Managers - Read, Write

- **Permission Propagation**: In some cases, **permissions are inherited** from parent directories or objects.
- For example, if you set permissions on a folder, the same permissions can be propagated to all files within that folder.

Example of DAC in Action

- **Scenario: Personal File System on a Computer**
- **Owner**: John creates a text document named "Budget Plan" on his personal computer.
- **Owner Control**: John decides that he wants to share the document with his colleague, Alice, but not with anyone else.
- **Action**: John gives Alice read-only access, while he retains full control (read, write, execute) over the document.
- **ACL for "Budget Plan"**:
  - File: Budget Plan.txt
  - User: John - Read, Write, Execute
  - User: Alice – Read
- **Dynamic Changes**: Later, John may decide to give Alice write access if they decide to collaborate on the document.

Advantages of DAC

•**Flexibility**:

•Users have control over their resources and can modify access permissions dynamically.

•Useful for small systems or personal use where strict access control is not required.

•**Ease of Implementation**:

•Simple to implement in environments where users are responsible for managing their own access rights.

•**User-Friendly**:

•Users with access to resources can easily manage and modify permissions for others, which simplifies user management.

Disadvantages of DAC

•**Security Risks**:

•Since the owner has control over permissions, there is a risk that users might grant excessive permissions, potentially compromising security.

•Example: A user might accidentally share a sensitive file with others or give write access to a file they shouldn't.

•**Lack of Centralized Control**:

•There is no centralized authority enforcing access control. This can be problematic in larger organizations or systems with many resources.

•**Scalability Issues**:

•In large systems with thousands of users and resources, manually managing permissions can become cumbersome and error-prone.

Disadvantages of DAC

- **Inconsistent Enforcement**:
- Since each resource is controlled individually, enforcing consistent security policies across the system can be challenging

DAC in Real-World Applications

- **1. File System Permissions (Windows/Linux)**
- **Windows OS**: DAC is used in NTFS file systems where the file owner determines who has access to the files and directories. Permissions such as **Full Control**, **Modify**, **Read & Execute**, and **Write** are commonly used.

- **2. Shared Network Folders**
- In a company network, DAC is used to grant access to shared network folders. A user can decide who in their department or team has access to their shared files.

- **3. Personal Cloud Storage**
- Many personal cloud storage services (e.g., Google Drive, Dropbox) implement DAC. The owner of a file can decide who has **view** or **edit** access.

CONCLUSION

•**Summary of DAC**:

•DAC allows owners to control access to their resources, providing flexibility and ease of use.

•Ideal for small systems or environments where users need control over their own data.

•**Risks**: It may introduce security vulnerabilities if permissions are not carefully managed.

•**Use Cases**:

•Personal file sharing.

•Small business file management.

•Cloud-based systems where users control sharing.

Introduction to Mandatory Access Control (MAC)

- **What is MAC?**
- **Mandatory Access Control (MAC)** is a type of access control model in which access to resources is governed by policies set by the system administrator, not the resource owner.
- **Key Feature**: Access decisions are based on **predefined security policies** and **security labels** rather than user discretion.
- **Primary Focus**: Security and confidentiality of resources.

- **Why MAC?**
- Used in environments where **high security** is required, such as **military**, **government**, and **financial institutions**, where unauthorized access to sensitive data must be strictly controlled.

Core Principles of MAC

1. System-Enforced Policies:

- Access control is enforced by the system based on predefined rules.
- **Example**: In a military system, a document labeled as **Top Secret** can only be accessed by users with **Top Secret** clearance.

2. No User Discretion:

- Users cannot modify or alter access permissions. Permissions are strictly controlled by the system.
- **Example**: A user with **Top Secret** clearance cannot share or alter access to a **Top Secret** file without system authorization.
- 3. **Levels of Security**:
- Security levels are **hierarchical** (e.g., **Top Secret**, **Secret**, **Confidential**, **Unclassified**), and users are granted access based on these levels.
- **Example**: A **Secret** user can read **Secret** and **Confidential** documents but cannot access **Top Secret** documents.

How MAC Works

•**Subjects**: Entities (users or processes) requesting access to resources.

•**Objects**: Resources (files, documents, or other data) being requested.

•**Security Labels**: These are assigned to both **subjects** and **objects** and dictate the level of access.

- **Example of MAC in Action:**
- **Subject**: A user with **Top Secret** clearance.
- **Object**: A file labeled **Secret**.
- **Policy**:
  - The user with **Top Secret** clearance can access the **Secret** file, but a user with **Secret** clearance cannot access the **Top Secret** file

Advantages of MAC

•**High Security**:

•Provides tight control over access to sensitive data, ensuring that only authorized individuals can access critical resources.

•**Centralized Control**:

•Centralized control by system administrators reduces the risks of unauthorized access or errors from resource owners.

•**Prevention of Data Leaks**:

•Policies like **No Read Up** and **No Write Down** prevent sensitive data from leaking to unauthorized users.

Disadvantages of MAC

•**Lack of Flexibility**:

•Users cannot modify access policies themselves, making the system rigid and difficult to adapt to changes in organizational needs.

•**Complex Management**:

•Setting up and managing security labels and access policies can be complex and time-consuming.

•**Scalability Issues**:

•As the organization grows, managing numerous security labels and access control policies becomes increasingly difficult.

When to Use MAC

- **Military**: MAC is ideal for protecting highly sensitive information, such as military intelligence, classified documents, and secure communications.
- **Government**: Ensuring compliance with regulations that require strict control over sensitive government data.
- **Healthcare**: Protecting patient health records and complying with privacy regulations like HIPAA.
- **Financial Institutions**: Safeguarding sensitive financial data and ensuring that only authorized personnel can access certain financial records.

MAC in Practice – Case Study Example

•**Scenario**:

•A **government** organization uses MAC to secure its **classified documents**.

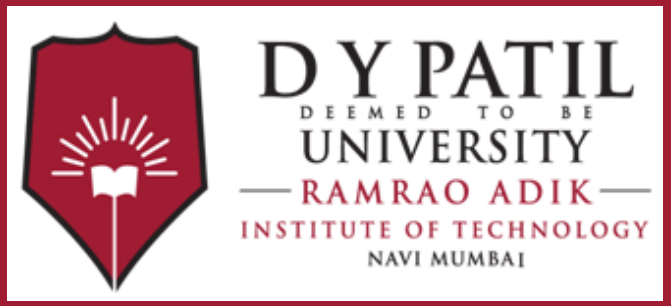
•**Top Secret** files are accessible only by senior officials with **Top Secret** clearance.

•**Secret** files are accessible by mid-level officials with **Secret** clearance.

•Each user's ability to access certain files is strictly based on their clearance level.

•**Result**:

•The system prevents accidental data leaks and ensures that only authorized individuals access classified materials, preserving national security.


D Y PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Thank You