

**CAPSTONE PROJECT**

**PROJECT TITLE**

Implementation of access control management

In operating system

**REPORT SUBMITTED BY**

POTHAMSETTI REDDY RENUKA(192311062)

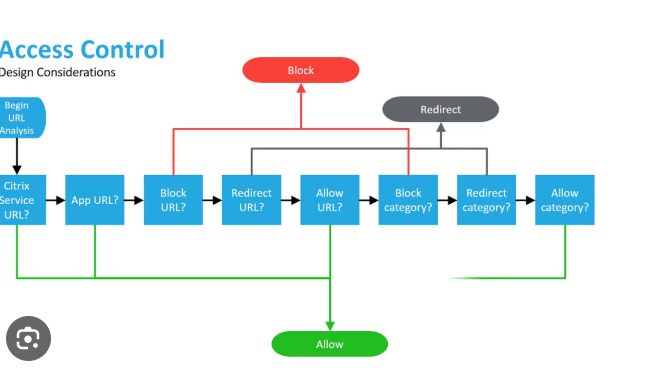
**COURSE CODE:** CSA0460

**COURSE NAME:** operating system for resource allocation

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Topic** | **Page no** |
| **1.0** | **Introduction** | **1** |
| **2.0** | **objectives** | **2** |
| **3.0** | **Access control models** | **3** |
| **4.0** | **Components of access control** | **4** |
| **5.0** | **Implementation in operating system** | **5** |
| **6.0** | **Tools and technologies** | **5** |
| **7.0** | **Challenges** | **6** |
| **8.0** | **Emerging trends in access control** | **6** |
| **9.0** | **Real world examples** | **7** |
| **10.0** | **Conclusion** | **7** |

**Introduction**

* **Define Access Control**
  + Access control is a security mechanism to regulate who or what can view or use resources in a computing environment



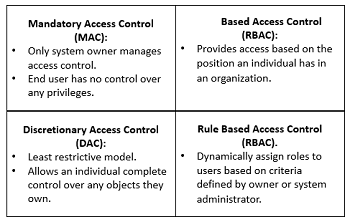
* **Importance in Operating Systems**
  + Ensures data integrity, confidentiality, and availability.

**Objectives**

* **What this presentation covers:**
  + Basics of access control
  + Types of access control
  + Implementation in operating systems
  + Real-world examples and challenges

**Access Control Models**

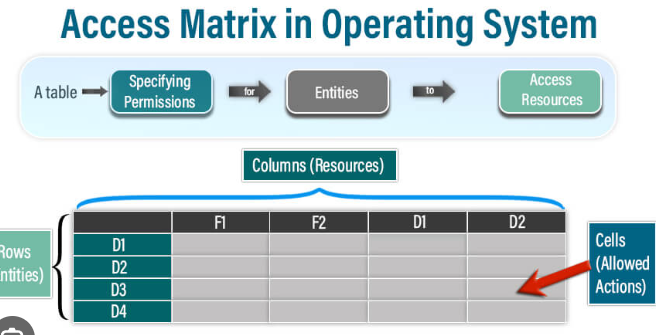
* **Discretionary Access Control (DAC)**:
  + Owner controls access to resources.
  + Example: File permissions in Windows/Linux.
* **Mandatory Access Control (MAC)**:
  + Access is determined by system policies.
  + Example: Military systems with security levels.
* **Role-Based Access Control (RBAC)**:
  + Access based on user roles.
  + Example: Admin, user, and guest privileges.
  + Add relevant diagrams or flowcharts



**Components of Access Control**

* **Subjects**: Users or processes requesting access.
* **Objects**: Files, databases, or resources being accessed.
* **Access Control Matrix**: Defines permissions between subjects and objects.

Include a table or diagram illustrating the Access Control Matrix.



**Implementation in Operating Systems**

* + **Windows**:
    - File system permissions (NTFS).
    - Access Control Lists (ACLs).
  + **Linux/Unix**:
    - File permissions (read, write, execute).
    - User and group policies.
  + **MacOS**:
    - Hybrid of Unix-based and proprietary models.
    - Add a comparison table for visual clarity

**Tools and Technologies**

* + **Authentication Methods**:
    - Passwords, biometrics, multi-factor authentication.
  + **Authorization Techniques**:
    - ACLs, capability lists, and security tokens.
    - Flow diagram showing authentication and authorization processes

**Challenges**

* Managing user permissions in large systems.
* Balancing usability and security.
* Handling insider threats.
* Adapting to new security standards.

**Emerging Trends in Access Control**

1. **Zero Trust Architecture**:
   * Assumes no implicit trust, even inside a network.
   * Verifies identity and access continuously.
2. **AI and Machine Learning**:
   * Automates monitoring and adjustment of access controls.
   * Identifies unusual access patterns or risks.
3. **Access Control in IoT**:
   * Lightweight mechanisms for resource-constrained devices.
4. **Blockchain for Decentralized Access Control**:
   * Ensures immutable logs and trustless verification.

**Real-World Examples**

* Highlight examples of access control in:
  + Enterprise environments (e.g., Active Directory).
  + Cloud platforms (e.g., AWS IAM policies).
  + Mobile operating systems (e.g., Android app permissions).

**Conclusion**

* Recap:
  + Importance of access control in maintaining security.
  + Different models and their applications.
* Future of access control:
  + AI and machine learning integration.
  + Zero Trust security model.
  + Thank the audience and provide Q&A time.