DAC: Discretionary access control (identity)

MAC: Mandatory access control (policy)

RBAC: Role-based access control (role)

***DAC: DAC: Discretionary access control (identity)***

**<What>**

Discretionary access control (DAC) is a type of security access control that grants or restricts object access via an access policy determined by an object's owner group and/or subjects. DAC mechanism controls are defined by user identification with supplied credentials during authentication, such as username and password. DACs are discretionary because the subject (owner) can transfer authenticated objects or information access to other users. In other words, the owner determines object access privileges.

**<How>**

In DAC, each system object (file or data object) has an owner, and each initial object owner is the subject that causes its creation. Thus, an object's access policy is determined by its owner.

A typical example of DAC is Unix file mode, which defines the read, write and execute permissions in each of the three bits for each user, group and others.

**<DAC attributes include:>**

* User may transfer object ownership to another user(s).
* User may determine the access type of other users.
* After several attempts, authorization failures restrict user access.
* Unauthorized users are blind to object characteristics, such as file size, file name and directory path.
* Object access is determined during access control list (ACL) authorization and based on user identification and/or group membership.

**<Disadvantages>**

DAC is easy to implement and intuitive but has certain disadvantages, including:

* Inherent vulnerabilities (Trojan horse)
* ACL(Access Control List) maintenance or capability
* Grant and revoke permissions maintenance
* Limited negative authorization power

<https://www.techopedia.com/definition/229/discretionary-access-control-dac>

***MAC: Mandatory access control (policy)***

**<What>**

In computer security, mandatory access control (MAC) refers to a type of access control by which the operating system or database constrains the ability of a subject or initiator to access or generally perform some sort of operation on an object or target.

**<How>**

In the case of operating systems, a subject is usually a process or thread; objects are constructs such as files, directories, TCP/UDP ports, shared memory segments, IO devices, etc. Subjects and objects each have a set of security attributes. Whenever a subject attempts to access an object, an authorization rule enforced by the operating system kernel examines these security attributes and decides whether the access can take place. Any operation by any subject on any object is tested against the set of authorization rules (aka policy) to determine if the operation is allowed. A database management system, in its access control mechanism, can also apply mandatory access control; in this case, the objects are tables, views, procedures, etc.

MAC over DAC

With mandatory access control, this security policy is centrally controlled by a security policy administrator; users do not have the ability to override the policy and, for example, grant access to files that would otherwise be restricted. By contrast, discretionary access control (DAC), which also governs the ability of subjects to access objects, allows users the ability to make policy decisions and/or assign security attributes. (The traditional Unix system of users, groups, and read-write-execute permissions is an example of DAC.) MAC-enabled systems allow policy administrators to implement organization-wide security policies. Under MAC (and unlike DAC), users cannot override or modify this policy, either accidentally or intentionally. This allows security administrators to define a central policy that is guaranteed (in principle) to be enforced for all users.

***RBAC: Role-based access control (role)***

**<What>**

Role-based access control (RBAC) restricts network access based on a person's role within an organization and has become one of the main methods for advanced access control. The roles in RBAC refer to the levels of access that employees have to the network.

<**How>**

Employees are only allowed to access the information necessary to effectively perform their job duties. Access can be based on several factors, such as authority, responsibility, and job competency. In addition, access to computer resources can be limited to specific tasks such as the ability to view, create, or modify a file.

As a result, lower-level employees usually do not have access to sensitive data if they do not need it to fulfill their responsibilities. This is especially helpful if you have many employees and use third-parties and contractors that make it difficult to closely monitor network access. Using RBAC will help in securing your company’s sensitive data and important applications.

<**Examples>**

Through RBAC, you can control what end-users can do at both broad and granular levels. You can designate whether the user is an administrator, a specialist user, or an end-user, and align roles and access permissions with your employees’ positions in the organization. Permissions are allocated only with enough access as needed for employees to do their jobs.

What if an end-user's job changes? You may need to manually assign their role to another user, or you can also assign roles to a role group or use a role assignment policy to add or remove members of a role group.

Some of the designations in an RBAC tool can include:

* Management role scope – it limits what objects the role group is allowed to manage.
* Management role group – you can add and remove members.
* Management role – these are the types of tasks that can be performed by a specific role group.
* Management role assignment – this links a role to a role group.
* By adding a user to a role group, the user has access to all the roles in that group. If they are removed, access becomes restricted. Users may also be assigned to multiple groups in the event they need temporary access to certain data or programs and then removed once the project is complete.

Other options for user access may include:

* Primary – the primary contact for a specific account or role.
* Billing – access for one end-user to the billing account.
* Technical – assigned to users that perform technical tasks.
* Administrative – access for users that perform administrative tasks.

<**Advantages>**

Managing and auditing network access is essential to information security. Access can and should be granted on a need-to-know basis. With hundreds or thousands of employees, security is more easily maintained by limiting unnecessary access to sensitive information based on each user’s established role within the organization. Other advantages include:

* **Reducing administrative work and IT support.** With RBAC, you can reduce the need for paperwork and password changes when an employee is hired or changes their role. Instead, you can use RBAC to add and switch roles quickly and implement them globally across operating systems, platforms and applications. It also reduces the potential for error when assigning user permissions. This reduction in time spent on administrative tasks is just one of several economic benefits of RBAC. RBAC also helps to more easily integrate third-party users into your network by giving them pre-defined roles.
* **Maximizing operational efficiency.** RBAC offers a streamlined approach that is logical in definition. Instead of trying to administer lower-level access control, all the roles can be aligned with the organizational structure of the business and users can do their jobs more efficiently and autonomously.
* **Improving compliance.** All organizations are subject to federal, state and local regulations. With an RBAC system in place, companies can more easily meet statutory and regulatory requirements for privacy and confidentiality as IT departments and executives have the ability to manage how data is being accessed and used. This is especially significant for health care and financial institutions, which manage lots of sensitive data such as PHI and PCI data.