***Discretionary Access Control (DAC)***

In computer security, Discretionary Access Control (DAC) is a type of [access control](http://www.webopedia.com/TERM/A/access_control.html) in which a user has complete control over all the programs it owns and executes, and also determines the permissions other users have those those files and programs. Because DAC requires permissions to be assigned to those who need access, DAC is commonly called described as a "need-to-know" access model.

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.

**Techopedia explains *Discretionary Access Control (DAC)***

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.

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

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

++++++++++++++++++++++++++++++

# Mandatory, Discretionary, Role and Rule Based Access Control

One of the key foundations of a comprehensive IT security strategy involves implementing an appropriate level of access control to all computer systems in an organization or enterprise. This chapter of[Security+ Essentials](http://www.techotopia.com/index.php/Security%2B_Essentials) will provide an overview of four types of access control for which an understanding is required to achieve CompTIA Security+ certification:

* Mandatory Access Control
* Discretionary Access Control
* Rule-Based Access Control
* Role-Based Access Control

## *An Overview of Access Control*

The term *Access Control* is something of an ambiguous term. To some it could be interpreted as controlling the access to a system from an external source (for example controlling the login process via which users gain access to a server or desktop system). In fact, such access control is actually referred to as *Authentication* or *Identity Verification* and is not what is meant by *Access Control* in this context (authentication is covered in detail in the [Authentication and Identity Verification](http://www.techotopia.com/index.php/Security%2B_-_Authentication_and_Identity_Verification) chapter of this book).

The term *Access Control* actually refers to the control over access to system resources *after* a user's account credentials and identity have been authenticated and access to the system granted. For example, a particular user, or group of users, might only be permitted access to certain files after logging into a system, while simultaneously being denied access to all other resources.

## *Mandatory Access Control*

Mandatory Access Control (MAC) is the strictest of all levels of control. The design of MAC was defined, and is primarily used by the government.

MAC takes a hierarchical approach to controlling access to resources. Under a MAC enforced environment access to all resource objects (such as data files) is controlled by settings defined by the system administrator. As such, all access to resource objects is strictly controlled by the operating system based on system administrator configured settings. It is not possible under MAC enforcement for users to change the access control of a resource.

Mandatory Access Control begins with *security labels* assigned to all resource objects on the system. These security labels contain two pieces of information - a classification (top secret, confidential etc) and a category (which is essentially an indication of the management level, department or project to which the object is available).

Similarly, each user account on the system also has classification and category properties from the same set of properties applied to the resource objects. When a user attempts to access a resource under Mandatory Access Control the operating system checks the user's classification and categories and compares them to the properties of the object's security label. If the user's credentials match the MAC security label properties of the object access is allowed. It is important to note that *both* the classification and categories must match. A user with top secret classification, for example, cannot access a resource if they are not also a member of one of the required categories for that object.

Mandatory Access Control is by far the most secure access control environment but does not come without a price. Firstly, MAC requires a considerable amount of planning before it can be effectively implemented. Once implemented it also imposes a high system management overhead due to the need to constantly update object and account labels to accommodate new data, new users and changes in the categorization and classification of existing users.

## *Discretionary Access Control*

Unlike Mandatory Access Control (MAC) where access to system resources is controlled by the operating system (under the control of a system administrator), Discretionary Access Control (DAC) allows each user to control access to their own data. DAC is typically the default access control mechanism for most desktop operating systems.

Instead of a *security label* in the case of MAC, each resource object on a DAC based system has an *Access Control List* (ACL) associated with it. An ACL contains a list of users and groups to which the user has permitted access together with the level of access for each user or group. For example, *User A* may provide read-only access on one of her files to *User B*, read and write access on the same file to *User C* and full control to any user belonging to *Group 1*.

It is important to note that under DAC a user can only set access permissions for resources which they already own. A hypothetical *User A* cannot, therefore, change the access control for a file that is owned by *User B*. *User A* can, however, set access permissions on a file that she owns. Under some operating systems it is also possible for the system or network administrator to dictate which permissions users are allowed to set in the ACLs of their resources.

Discretionary Access Control provides a much more flexible environment than Mandatory Access Control but also increases the risk that data will be made accessible to users that should not necessarily be given access.

## *Role Based Access Control*

Role Based Access Control (RBAC), also known as *Non discretionary Access Control*, takes more of a real world approach to structuring access control. Access under RBAC is based on a user's job function within the organization to which the computer system belongs.

Essentially, RBAC assigns permissions to particular roles in an organization. Users are then assigned to that particular role. For example, an accountant in a company will be assigned to the *Accountant*role, gaining access to all the resources permitted for all accountants on the system. Similarly, a software engineer might be assigned to the *developer* role.

Roles differ from *groups* in that while users may belong to multiple groups, a user under RBAC may only be assigned a single role in an organization. Additionally, there is no way to provide individual users additional permissions over and above those available for their role. The accountant described above gets the same permissions as all other accountants, nothing more and nothing less.

## *Rule Based Access Control*

Rule Based Access Control (RBAC) introduces acronym ambiguity by using the same four letter abbreviation (RBAC) as Role Based Access Control.

Under Rules Based Access Control, access is allowed or denied to resource objects based on a set of rules defined by a system administrator. As with *Discretionary Access Control*, access properties are stored in Access Control Lists (ACL) associated with each resource object. When a particular account or group attempts to access a resource, the operating system checks the rules contained in the ACL for that object.

Examples of Rules Based Access Control include situations such as permitting access for an account or group to a network connection at certain hours of the day or days of the week.

As with MAC, access control cannot be changed by users. All access permissions are controlled solely by the system administrator.

--------------------

### Q. Describe the theory of Discretionary Access control based on granting and revoking privileges?

Discretionary access control is based on the idea of access rights, or privileges, and mechanisms for giving users such privileges. A privilege allows a user to access some data object in a certain manner (e.g. to read or modify). A user who creates data object such as a table or a view automatically gets all applicable privileges on that object and the user can also propagate privileges using "Grant Option". The DBMS subsequently keeps track of how these privileges are granted to other users, and possibly revoked, and ensures that at all times only users with the necessary privileges can access an object.

SQL Syntax

SQL supports discretionary access control through the GRANT and REVOKE commands.

The GRANT command gives users privileges to base tables and views.

The REVOKE command cancels uses' privileges.

**For example:**

GRANT privilege1, privilege2, ...                    REVOKE privilege1, privilege2, ...  
                         ON object\_name                                              ON object\_name  
                         TO user1, user2, ... ;                                         FROM user1, user2, ... ;  
                      
                         GRANT SELECT, ALTER                               ROVOKE SELECT, ATLER  
                         ON student                                                         ON student  
                         TO db2\_14                                                         FROM db2\_14