8.3.3 Authorization Facts

Authorization is the process of controlling access to resources, such as computers, files, or printers. When managing access to resources, be aware of the following:

- A *group* is an object that identifies a set of users with similar access needs. Microsoft systems have two kinds of groups, distribution and security. Only security groups can be used for controlling access to objects.
- Assigning permissions to a group grants those same permissions to all members of the group.
- On a Microsoft system, a *user right* is a privilege or action that can be taken on the system, such as logging on, shutting down the system, backing up the system, or modifying the system date and time.
- Permissions apply to objects (files, folders, printers, etc.), while user rights apply to the entire system (computer).

An *access control list* (ACL) identifies users or groups who have specific security assignments to an object. The term *permission* identifies the type of access that is allowed or denied for the object. For example, permissions for a file include read and write and can either allow or deny the specified access. The table below describes two types of NTFS access lists:

Type of Access List	Characteristics
Discretionary Access Control List	A discretionary access control list (DACL) is an implementation of discretionary access control (DAC). Owners add users or groups to the DACL for an object and identify the permissions allowed for that object.
System Access Control List	A system access control list (SACL) is used by Microsoft for auditing to identify past actions performed by users on an object.

A *security principal* is an object that can be given permissions to an object. Security principals include user accounts, computer accounts, and security group accounts.

- Each security principal is given a unique identification number called a SID (security ID).
- When a security principal logs on, an access token is generated. The access token is used for controlling access to resources and contains the following information:
  - The SID for the user or computer
  - The SID for all groups the user or computer is a member of
  - User rights granted to the security principal
- When the security principal tries to access a resource or take an action, information in the access token is checked. For example, when a user tries to access a file, the access token is checked for the SID of the user and all groups. The SIDs are then compared to the SIDs in the object's DACL to identify permissions that apply.
- On a Microsoft system, the access token is only generated during authentication. Changes made to group memberships or user rights do not take effect until the user logs on again and a new access token is created.

A key part of the security administrator's job is to control access to resources. Access to resources is controlled using permissions, privileges, and roles.

> Permissions, privileges, and roles are usually cumulative, making it possible for one user account to receive access to more than one entity.

Types of permission are described in the table below.

Permission Type	Description
Effective Permissions	Access rights (permissions) are cumulative. If you are a member of two groups, both with different permissions, you will have the combined permissions of both groups (known as *effective permissions*). The effective permissions are the combined inherited permissions and explicit permissions.
Deny Permissions	Deny permissions always override Allow permissions. For example, if a user belongs to two groups and a specific permission is allowed for one group and denied for the other, the permission is denied. However, the exception to this rule comes with inherited permissions. If an object has an explicit Allow permission entry, inherited Deny permissions do not prevent access to the object. Explicit permissions override inherited permissions, even Deny permissions.
Cumulative Permissions	Use the following suggestions to help plan permissions and mitigate issues related to cumulative permissions: ▪ Identify the users and their access needs (the actions each user needs to be able to perform). ▪ Based on the types of users you identify, create groups for multiple users with similar needs and then make users members of groups. ▪ Assign each group (not user) the permissions appropriate to the group's data access needs. Grant only the permissions that are necessary.

	▪ As you assign permissions, take inheritance into account. *Inheritance* means that permissions granted to a parent container object flow down to child objects within the container. Set permissions as high as possible on the parent container and allow each child container to inherit the permissions. ▪ When necessary, you can override inheritance on a case-by-case basis.