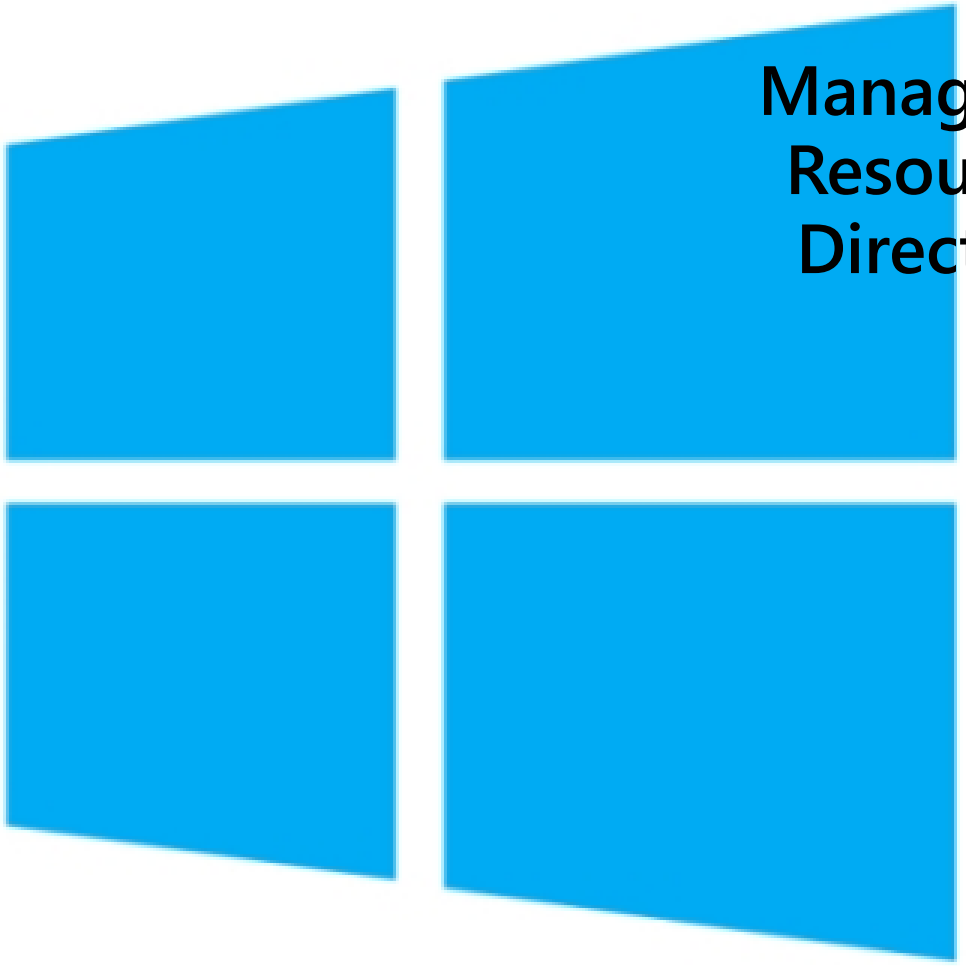
Module 5

Managing Access to Resources in Active Directory® Domain Services



Windows Server 2016

Module Overview

- Managing Access Overview
- Managing NTFS File and Folder Permissions
- Assigning Permissions to Shared Resources
- Determining Effective Permission

Lesson 1: Managing Access Overview

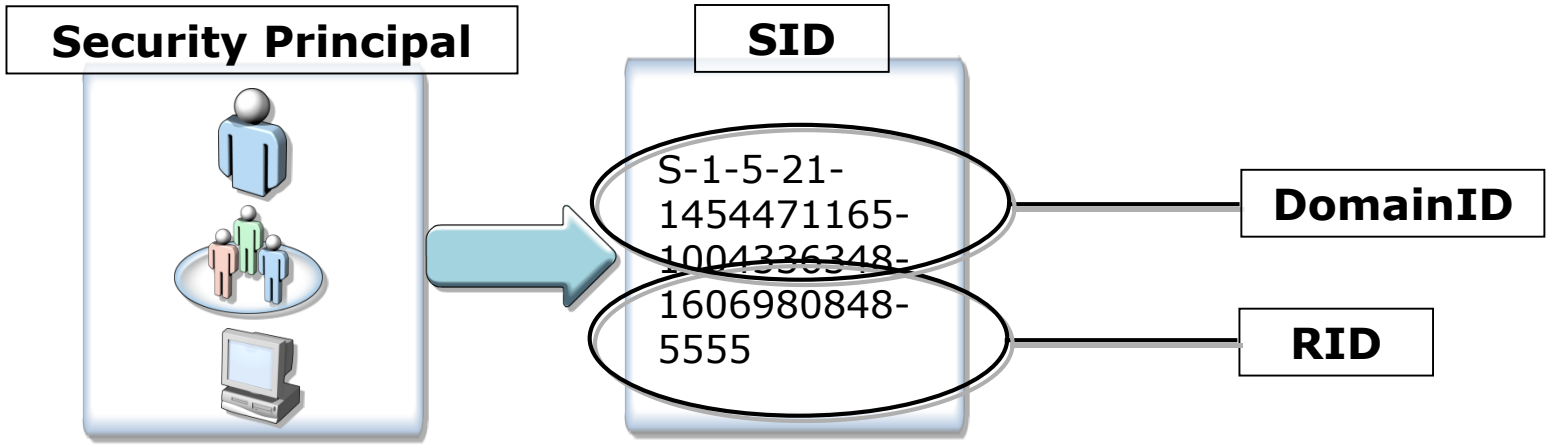
- What Are Security Principals?
- What Are Access Tokens?
- What Are Permissions?
- How Access Control Works

What Are Security Principals?

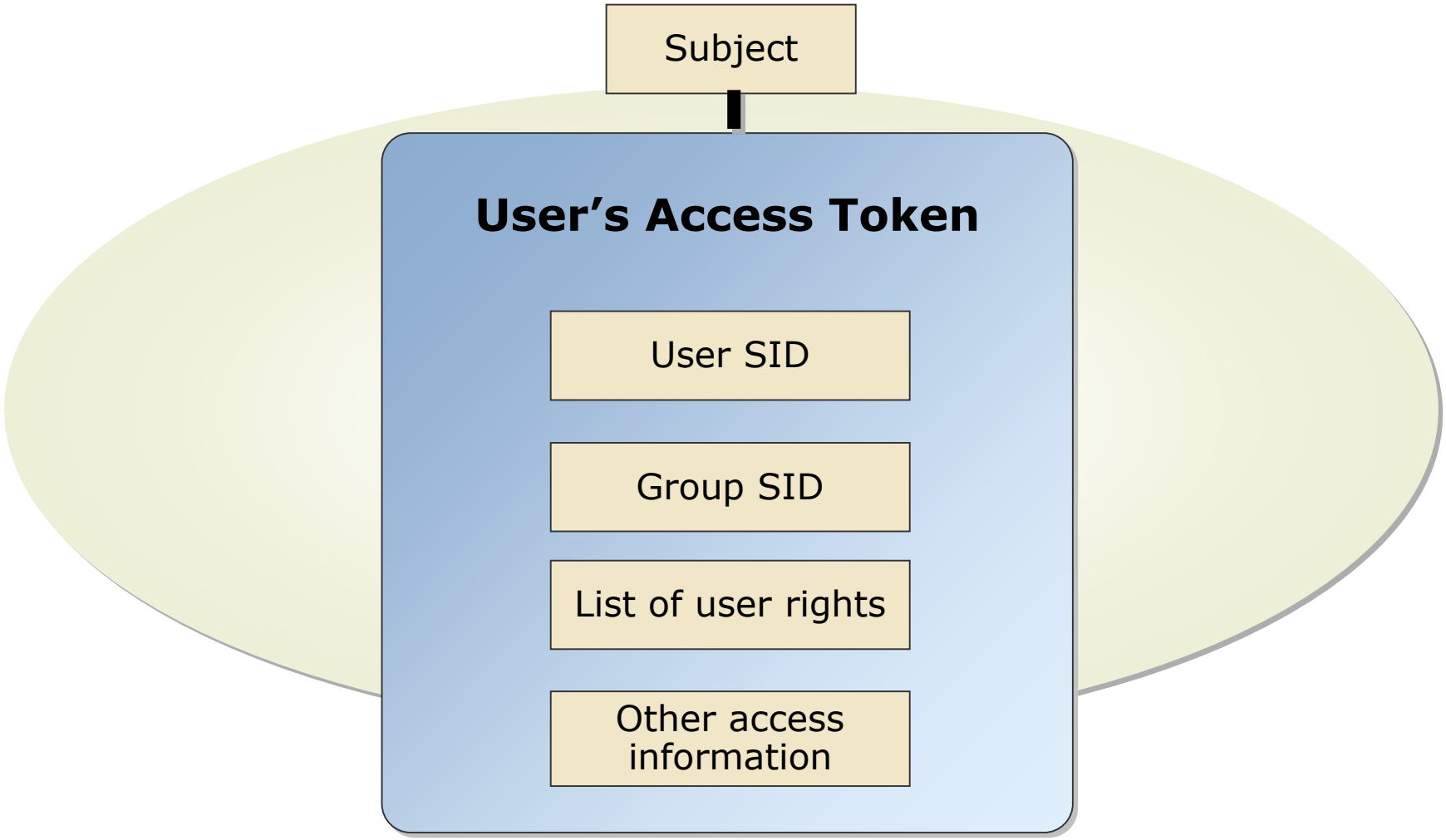
Security Principal - A user, group, or computer object that can be used for authentication and to assign access to resources.

Security ID (SID) - A unique value assigned when a user, computer or security group is created. Internal processes in Windows refer to an account's SID instead of the account's user or group name.

Relative ID (RID) - The part of a security ID (SID) that uniquely identifies an account or group within a domain.



What Are Access Tokens?



What Are Permissions?

Permissions:

- **Are rules to grant or deny access to an object**
- **Used to control access**

How are permissions assigned?

- **Allow or deny permissions can be assigned to a resource (folder, printer, file)**
- **Permissions can be assigned to accounts from the local computer or from AD DS**
- **Permissions can be explicitly applied, inherited, or implicitly applied**

How Access Control Works

Discretionary Access Control List (DACL)

- DACL contains a list of users and groups that can access or have been denied access to the resource
- Every file and folder on a NTFS volume has an associated DACL

System Access Control List (SACL)

- SACL controls auditing of access to the resource

Access Control Entry (ACE)

- Defines each entry in a DACL or SACL
- Specifies the set of SIDs that are to be allowed, denied or audited
- If no ACE is specified within a DACL, access to the resource is denied

Lesson 2: Managing NTFS File and Folder Permissions

- What Are NTFS Permissions?
- What Are Standard and Special Permissions?
- What Is NTFS Permissions Inheritance?
- Effects on NTFS Permissions When Copying and Moving Files and Folders

What Are NTFS Permissions?

File Permissions	Folder Permissions
Read	Read
Write	Write
Read & Execute	List Folder Contents
Modify	Read & Execute
Full Control	Modify
	Full Control

Deny Permissions take precedence over Allow Permissions

What Are Standard and Special Permissions?

Special Permissions		
Traverse Folder/ Execute File	Create Folders/Append Data	Read Permissions
List Folder/ Read Data	Write Attributes	Change Permissions
Read Attributes	Write Extended Attributes	Take Ownership
Read Extended Attributes	Delete Subfolders and Files	Synchronize
Create Files/Write Data	Delete	

Standard Permissions		
Read	List Folder Contents	Modify
Write	Read & Execute	Full Control

What Is NTFS Permissions Inheritance?

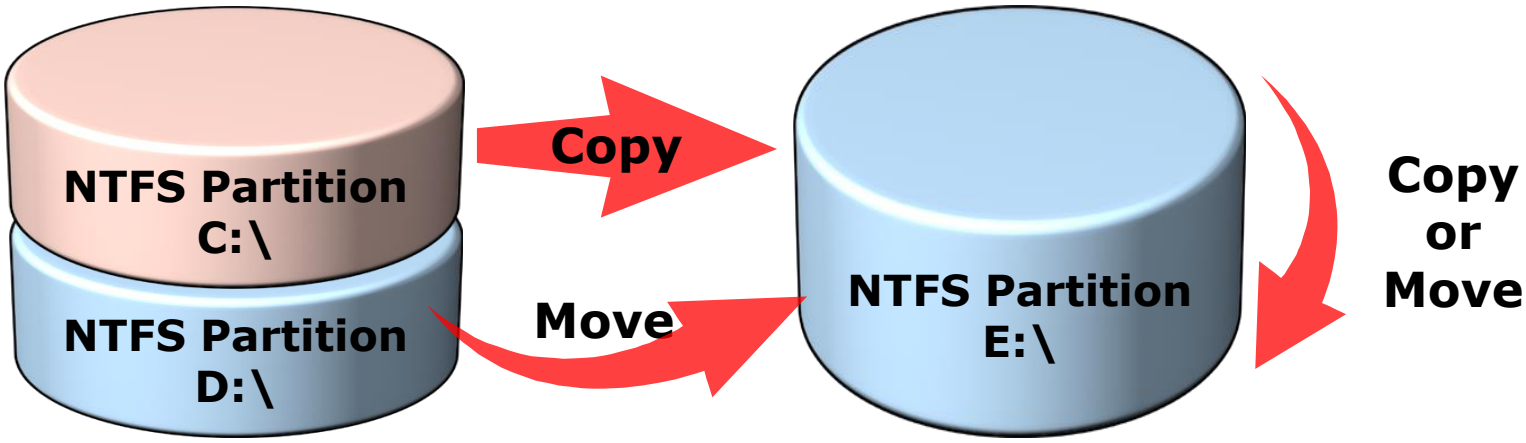
Inheritance is used to manage access to resources without assigning explicit permissions to each object

By default, NTFS permissions are inherited in a parent/child relationship

Blocking

- **Permission Inheritance can be blocked**
- **Blocking can be performed at the file or folder level**
- **Blocking on a folder can be set to propagate the new permissions to child objects**

Effects on NTFS Permissions When Copying and Moving Files and Folders



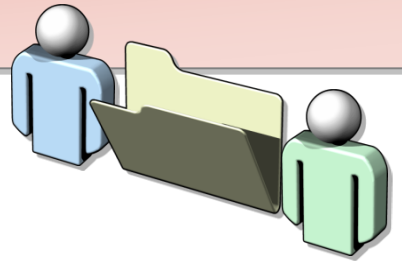
- When you copy files and folders, they inherit the permissions of the destination folder
- When you move files and folders within the same partition, they keep their permissions
- When you move files and folders to a different partition, they inherit the permissions of the destination folder

Lesson 3: Assigning Permissions to Shared Resources

- What Are Shared Folders?
- What Are Administrative Shared Folders?
- Shared Folder Permissions
- Connecting to Shared Folders
- Considerations for Using Shared Folders
- Offline File Configuration and Deployment

What Are Shared Folders?

Shared Folders are folders that allow network access to their contents



- **Folders can be shared, but individual files cannot**
- **By default the shared folders permission is Full Control for the user that shared the folder**
- **Shared folders can be identified:**
  - **Through the MMC Console Share and Storage Management**
  - **In Windows Explorer by the two user icon under the folder**
  - **Through the command line through Net Share**
  - **Through Computer Manager under Shared Files**

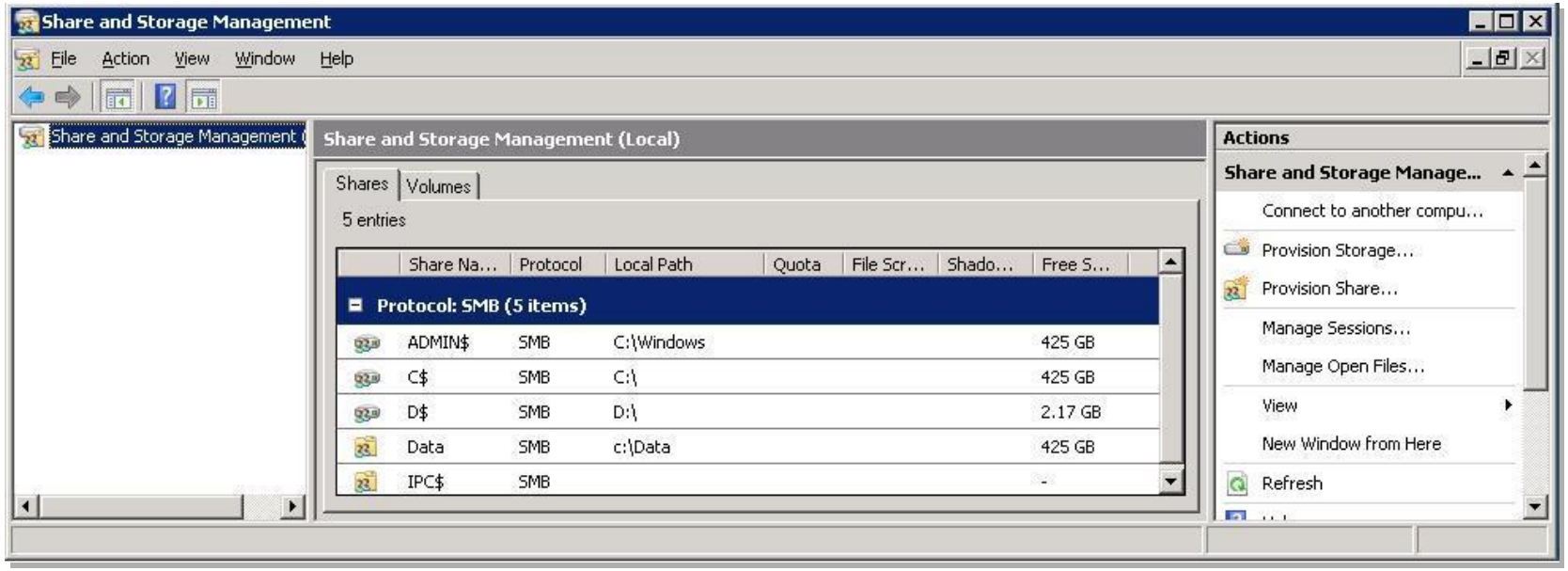
What Are Administrative Shared Folders?

Administrative Shares:

- **Are hidden shares**
- **Are not displayed when using Net View or in the Network view**

Administrators have full permissions

Share permissions cannot be changed



Shared Folder Permissions

Permission Level	Access
Read	• Allows for viewing of data in files • Allows for subfolder browsing • Programs in the shared folder can be executed • By default, applied to the Everyone group
Change	• All the permissions in the Read category • New files and subfolders can be created • Data in existing files can be modified or removed • Files and subfolders can be deleted
Full Control	• Full permissions included in the Read and Change categories plus permission to change security settings

Connecting to Shared Folders

Access through UNC:

- Naming convention is \\servername\share or \\servername\share\file
- Can be accessed through Windows Explorer, command line, or programmatically

Access through mapped drives:

- Use Windows Explorer or command line to map a drive to \\servername\share

Access through Network:

- Uses a graphical tool to browse the network for shares
- Works in domain or workgroup mode
- Does not show hidden or administrative shares

Considerations for Using Shared Folders

When creating shared folders:

- **Use the most restrictive permissions possible**
- **Avoid assigning permissions to individual users, use groups whenever possible**
- **Remember Full Control lets users modify NTFS permissions. Add groups to the Full Control permission group with caution**
- **Add the Authenticated Users group and remove the Everyone group from the share's permissions**

Offline File Configuration and Deployment

When creating offline files:

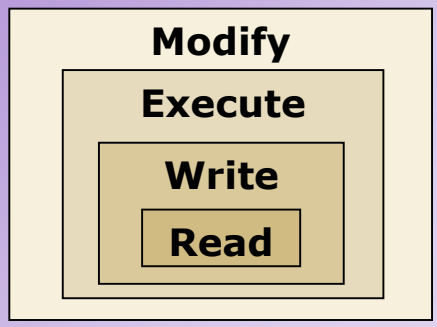
- **Select a folder at a networking place, synchronize and then disconnect computer**
- **Make edits to documents on disconnected computer**
- **Reconnect to the computer to the network again to update changes**
- **Files are synchronized automatically**

Lesson 4: Determining Effective Permission

- What Are Effective NTFS Permissions
- Discussion: Applying NTFS Permissions
- Effects of Combining Shared Folder and NTFS Permissions
- Discussion: Determining Effective NTFS and Shared Folder Permissions
- Considerations for Implementing NTFS and Shared Folder Permissions

What Are Effective NTFS Permissions?

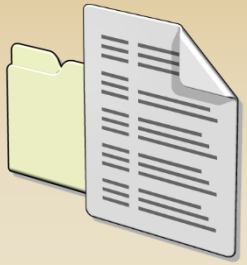
- **NTFS Permissions are cumulative**



- **Deny takes precedence**

- **Permissions can be applied to a user or a group**

- **File permissions override folder permissions**



- **Creators of file and folders are the owners**

Discussion: Applying NTFS Permissions

1

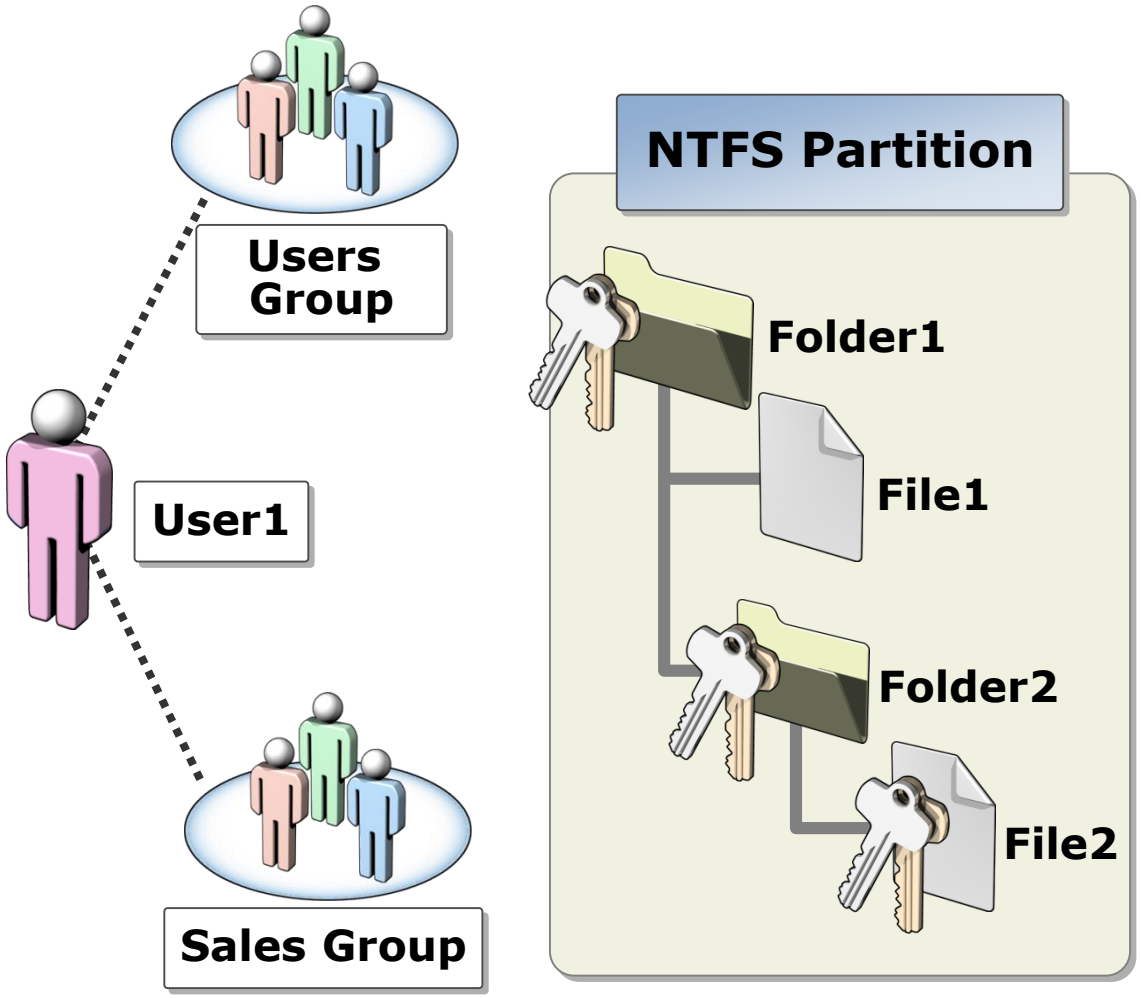
- Users group has **Write** for Folder1
- Sales group has **Read** for Folder1

2

- Users group has **Read** for Folder1
- Sales group has **Write** for Folder2

3

- Users group has **Modify** for Folder1
- File2 should only be available to Sales group with Read permission



Effects of Combining Shared Folder and NTFS Permissions

- **When combining shared folder and NTFS permissions, the most restrictive permission is applied**

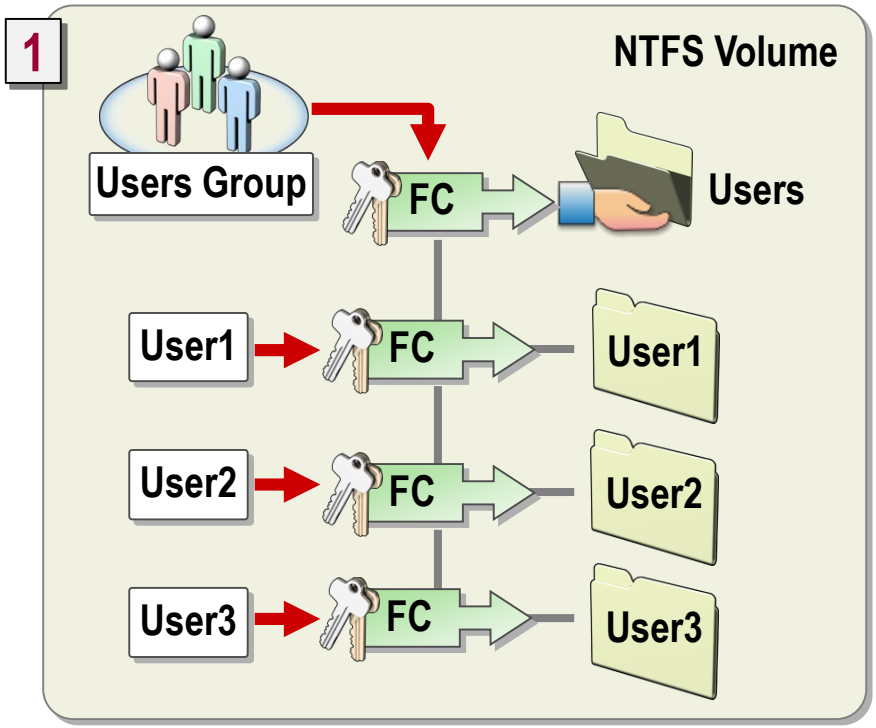
  Example: If a user or group is given the Share permission of Read and the NTFS permission of Write, the user or group will only be able to read the file because it is the more restrictive permission

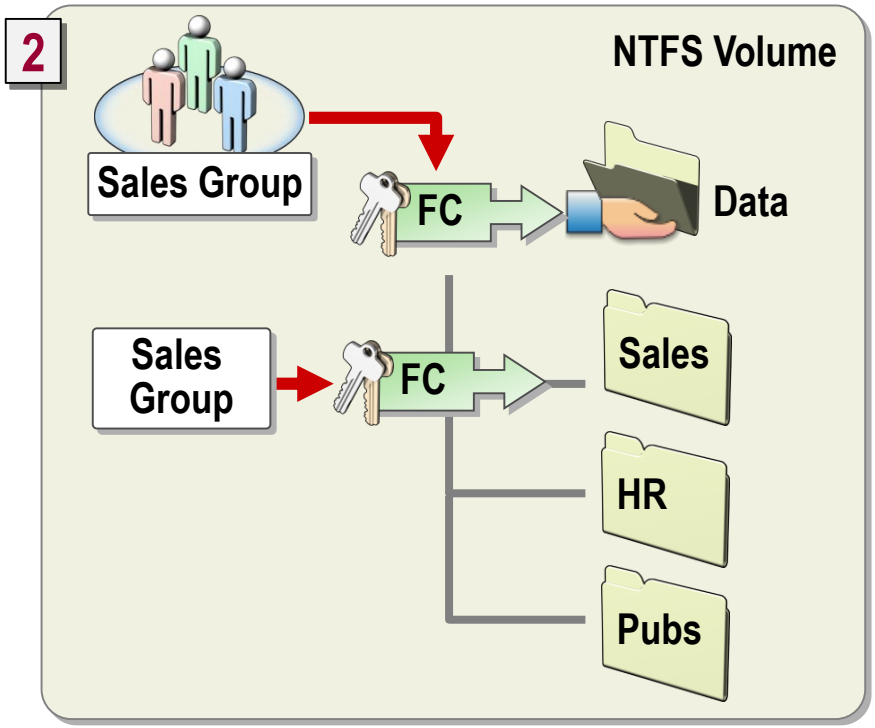
- **Both the share and the NTFS File and Folder permissions must have the correct permissions, otherwise the user or group will be implicitly denied access to the resource**

Discussion: Determining Effective NTFS and Shared Folder Permissions

Class discussion:

- Determine effective NTFS permissions
- Determine shared folder permissions





Considerations for Implementing NTFS and Shared Folder Permissions

- **Grant permissions to groups instead of users**
- **Use Deny permissions only when necessary**
- **Never deny the Everyone group access to an object**
- **Grant permissions as high in the folder structure as possible**
- **Use NTFS permissions instead of shared permissions for fine-grained access**

Module Review and Takeaways

- Review questions
- Considerations for managing shared folders and NTFS permissions