Authorization and Access Control Technologies

Updated: March 28, 2003

Applies To: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

Authorization and Access Control Technologies

Security in the Microsoft Windows Server 2003 operating system controls the use of system and network resources through the interrelated mechanisms of authentication and authorization. After a user is authenticated, Windows Server 2003 uses the authorization and access control technologies to implement the second phase of protecting resources: determining if an authenticated user has the correct authorization to access a resource. Shared resources — resources available to users and groups other than the resource's owner — need to be protected from unauthorized use. In the Windows Server 2003 access control model, users and groups are assigned rights that inform the operating system what each user and group can and cannot do. Shared resources are assigned permissions that enable resource managers to enforce access control in two ways: by denying access to unauthorized users, and by limiting the extent of access provided to authorized users.

Authorization and Access Control Technologies Architecture

The authorization and access control model used in Windows Server 2003 is based on the following concepts:

User-based authorization
Every application that a user starts runs in that user's security context, not in the application's security context. Applications can also run in a restricted security context, with fewer privileges and more limited access than their user's security context.

Discretionary access to securable objects
The user who owns a securable object can control who has permission to use it and in what way. An object's owner can give permission for different kinds of access to particular users or groups of users. Owners can also allow or deny other users access to individual properties of certain types of objects, as well as to the entire object.

Inheritance of permissions
You can control permissions for new objects created in a container object by setting inheritable permissions on the container. The permissions that you set on a container are also inherited by existing objects in the container as well as newly created objects.

Administrative privileges
You can control which users or groups have the right to perform various administrative functions or to take any action that affects systemwide resources. Domain administrators can use Group Policy to manage privileges on several computers at once or even on all computers joined to a domain.

Auditing of system events
The auditing feature detects attempts to circumvent protections on resources and creates an audit trail of administrative actions on the system. If another administrator changes the auditing policy so that failed logon attempts are no longer audited, the log shows this event too. You can also use Group Policy to centrally control who is allowed to manage security logs on computers joined to a domain, as well as to control such configuration options as log size and retention method.

How Access Control Works

The access control model used by the Windows Server 2003 operating system ensures authorized use of its objects by security principals. Security principals include users and groups. Security principals perform actions on objects, which include files, folders, printers, registry keys, Active Directory directory service objects, and other types of objects.

Each object has an owner that grants permissions to security principals. During the access control check, these permissions are examined to determine which security principals can access the object and how they can access it.

Object owners generally grant permissions to security groups rather than to individual users. Users added to existing groups adopt the permissions of that group. Object owners also often define permissions for container objects, rather than individual objects, to ease access control management. If an object (such as a folder) can hold other objects (such as subfolders and files), it is called a container. In a hierarchy of objects, the relationship between a container and its content is expressed by referring to the container as the parent and to an object in the container as the child.

Although users are the ones who attempt to access a shared resource, programs actually perform the operations. More specifically, the executable program sequence within a process, called a thread, runs program instructions. A process can have several threads, all executing at the same time. In describing the access control model, the subject taking action is always a thread, never a user, program, or process.

Although only threads can act on system objects, threads do not carry their own security identifiers (SIDs). A thread assumes the rights and privileges of the security principal that initiated the process. When a user logs on, the system creates an access token for that user. The access token contains the user's SID, the SIDs for any groups the user belongs to, and the user's privileges. This token provides the security context for whatever actions the user executes on that computer.

Threads do not access files in the same way that users do. Threads interact with objects through one of several application programming interfaces (APIs) that are provided by the operating system. For example, the thread that opens a file is probably executing code with the following instruction:

```
hfile=CreateFile(pszFile,GENERIC_WRITE,0,NULL,OPEN_EXISTING,0,NULL);
```

The second argument in the call to CreateFile() specifies a desired set of access rights, GENERIC_WRITE, which indicates to the operating system that the thread wants to open the file and modify it. Other APIs work in a similar fashion. The calling process must signal its intentions for an object by specifying a desired level of access.

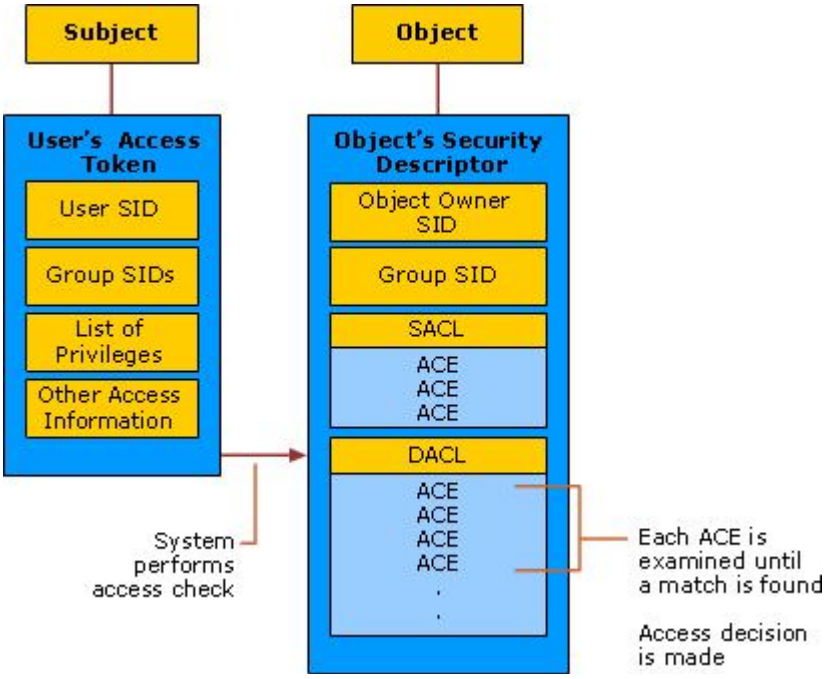
Before allowing a subject to proceed with the action it intends to carry out on an object, the operating system's security subsystem performs an access check to determine whether the subject is authorized to perform the action. The access check systematically compares information in two key data structures:

- The subject's access token, which contains a SID for the user and additional SIDs for groups that the user belongs to.
- The object's security descriptor, which contains a discretionary access control list (DACL) with a list of access control entries (ACEs) that specify the access rights that are allowed or denied to particular users or groups, each of whom is identified by a SID.

The security subsystem checks the object's DACL, looking for ACEs that apply to the user and group SIDs in the subject's access token. The system examines each ACE in order until it finds one that either allows or denies access to the user or one of the user's groups, or until there are no more ACEs to check. It is possible for a DACL to have several ACEs that apply to the token's SIDs. If this occurs, the access rights granted by each ACE accumulate until the total matches or exceeds the level of access that the subject has requested. For example, if one ACE grants read access to a group and another ACE grants write access to a user who is a member of the group, the user can have both read/write access to the object. If the access check reaches the end of the DACL and the desired access is still not explicitly allowed or denied, the security subsystem denies access.

The following figure shows the authorization and access control process.

Authorization and Access Control Process



Authorization and Access Control Technologies Components

The following table provides a summary of the components that comprise the authorization and access control model.

Components of Authorization and Access Control

Components	Description
Security Principal	A user, group, or computer. Security principals have accounts. Local accounts are managed by the Security Accounts Manager (SAM) on the computer. Domain accounts are managed by Active Directory.
Security Identifier (SID)	The Windows security model identifies account objects — such as users, groups, computers, and domain trusts — by SIDs. SIDs are domain-unique values, built when the user or group is created or when the computer or trust is registered with the domain. The components of a SID follow a hierarchical convention: A SID contains parts that identify the revision number, the authority — such as the domain — that issued the SID, and a variable number of sub authority or relative identifier (RID) values that uniquely identify the security principal relative to the issuing authority.
Access Tokens	After a user logs on and is authenticated, the system creates an access token for the user containing the SID of the user (the primary SID), and the SIDs of all the domain groups the user is a member of. The system uses this access token to determine whether to grant the user access to system resources.
Permissions	Authority to perform an operation or a set of operations on an object or object property. Permissions are granted or denied by an object's owner. Because access to an object is at the owner's discretion, the type of access control used in Windows Server 2003 is called discretionary access control. Permissions are applied to any secured objects such as files, Active Directory objects, or registry objects. Permissions can be granted to any account — a user, group, or computer. You can assign permissions for objects to: • Groups, users, and special identities in the domain. • Groups and users in that domain and any trusted domains. • Local groups and users on the computer where the object resides.

	The permissions that can be attached to an object depend on the type of object. For example, the permissions that can be attached to a file are different from those that can be attached to a registry key.
User Rights	Authority to perform an operation that affects an entire computer rather than a particular object. User rights (also known as privileges) are assigned by administrators to individual users or groups as part of the security settings for the computer. Although user rights can be managed centrally through Group Policy, they are applied locally. Users can (and usually do) have different user rights on different computers. User rights grant specific privileges and logon rights to users and groups in your computing environment.
Security Descriptors	A data structure containing the security information associated with a securable object, such as a file or a printer.
Access Control Lists (ACLs)	If permissions are configured for an object, its security descriptor contains a discretionary access control list (DACL) with SIDs for the users and groups who are allowed or denied access. If auditing is configured for the object, its security descriptor also contains a system access control list (SACL) that controls how the security subsystem audits attempts to access the object.

How authorization and access control functions also depends upon related technologies in the Windows Server 2003 security architecture. The following table provides a summary of these related technologies.

Technologies Related to Authorization and Access Control

Related Components	**Description**
Authentication	During the authentication process, a user is identified to the local or trusted domain by presentation of credentials, usually in the form of a user name and password. Assuming these credentials are acceptable, the system creates an access token for the user.
Auditing	Establishing an audit policy allows you to track potential security problems, help to ensure user accountability, and provide evidence in the event of a security breach. The most common types of events to be audited are: • Access to objects, such as files and folders. • Management of user accounts and group accounts. • Users logging on to and logging off from the system. You can then view these security-related events in the security log with the Event Viewer.
Trusts	By default, all users in a specific Microsoft Windows NT 4.0 or Active Directory domain can connect to resources contained within that domain. This means that a single domain can, by default, provide network connectivity, for users that are members of that domain, to all resources that reside within that domain. To expand resource sharing beyond the boundaries of a single domain, you will need to use trust relationships. Trust relationships provide a security channel for data to flow across domains or forests.

SIDHistory	In Active Directory, domain migration or restructuring across trusts is made considerably easier as a result of an attribute of Active Directory security principals called SIDHistory. SIDHistory is used to store the former SIDs of moved objects such as users and security groups. When a user is moved, the SIDHistory attribute of the user object is updated with the former SID. When the user then logs onto the system, the system retrieves the entries in the SIDHistory and adds them to the user access token. In this way SIDHistory ensures that migrated users can continue to access resources even when the resources are located in a trusting domain.
SID Filtering	SID filtering prevents domains from accepting SIDs with domain SIDs from outside the sender's domain. Applying SID filtering to trusts can prevent malicious users who have domain administrator level access in the trusted domain from granting, to themselves or other user accounts in their domain, elevated user rights to the trusting domain.

Authorization and Access Control Technologies Deployment Scenarios

Common deployment scenarios for controlling access to resources include the User/ACL method, the Account group/ACL method, the Account group/Resource group method, and the role-based authorization method.

These methods are all based on a planning and deployment process that involves first identifying your users by workgroup, job function, or a combination of workgroup and job function. You can then identify the different types of resources that users might access, such as departmental or job-specific data. This allows you to plan for different levels of access. These access control methods rely on the following concepts:

- **Account group** An account group is a security group whose members are user or computer accounts, all of which require the same permissions for a resource
- **Resource group** A resource group is a security group that has been added to the access control list (ACL) of a resource and granted a specific set of access permissions
- **Principle of Least Access** The principle of least access states that users must have access to the software, data, and devices required to perform their daily duties, but must not have access to local or network resources that are not required for their job tasks. This minimizes potential security risks

In Windows Server 2003, there are four basic strategies for controlling access to resources:

User/ACL Method

Security principal accounts are added directly to the ACL for a resource. The ACL is then configured to allow the appropriate permissions for each security principal. This option is appropriate only for managing basic access to a limited number of resources for a small number of users. Otherwise, the complexity and administrator time needed to set and maintain the appropriate permissions becomes burdensome.

Account Group/ACL Method

Security groups, rather than individual security principal accounts, are added to the resource ACL, and the group is given a set of access permissions. The Account group/ACL method is scalable because security groups can be nested if your domain is configured for Microsoft Windows 2000 or Windows Server 2003 native mode functional levels. If groups from multiple domains or from multiple forests require identical access permissions, they can be grouped together into one all-encompassing security group, and you can add this security group to the resource ACL. However, if different groups require different access permissions, this method requires more administrative effort from the resource owner. In this case, each group must be added to the ACL separately, and access permissions must be granted to each group.

Account Group/Resource Group Method

Users with similar access requirements are grouped into account groups, and the account groups are added to a resource group that is granted specific resource access permissions. The Account group/resource group method is often most appropriate for large organizations with many shared resources because it is scalable and can be maintained in all environments and at all domain functional levels. This method can be used in any environment, but it is most useful when

you cannot nest groups because, for example, your domains are at Windows 2000 mixed functional levels or you need to share resources with Windows NT version 4.0 operating system domains. The Account group/resource group method is also preferable if your resources require a stable set of common permissions.

Role-based Authorization Method

Users with similar roles are authorized to perform predefined sets of tasks based on scripts called authorization rules. This allows you to apply fine-grained control over the mapping between access control and the structure and tasks performed in your organization. To use role-based authorization, however, programs need to be written to take advantage of this capability. Additionally, your environment must be operating at the Windows Server 2003 domain functional level.

Community Additions
