DCOM Security Enhancements in Windows XP Service Pack 2 and Windows Server 2003 Service Pack 1 - Win32 apps

By stevewhims

Archived: 2026-04-05 16:05:07 UTC

Windows Server XP Service Pack 2 (SP2) and Windows Server 2003 Service Pack 1 (SP1) introduce enhanced default security settings for the Distributed Component Object Model (DCOM). Specifically, they introduce more granular rights that enable an administrator to have independent control over local and remote permissions for launching, activating, and accessing COM servers.

The Microsoft Component Object Model (COM) is a platform-independent, distributed, object-oriented system for creating binary software components that can interact. The Distributed Component Object Model (DCOM) allows applications to be distributed across locations that make the most sense to you and to the application. The DCOM wire protocol transparently provides support for reliable, secure, and efficient communication between COM components.

Who does this feature apply to?

If you use COM only for in-process COM components, this feature does not apply to you.

This feature applies to you if you have a COM server application that meets one of the following criteria:

- The access permission for the application is less stringent than the launch permission, which is necessary to run the application.
- The application is usually activated by a remote COM client without using an administrative account.
- The application is meant to be used only locally. This means you can restrict your COM server application so it is not remotely accessible.

What new functionality is added to this feature in Windows XP Service Pack 2 and Windows Server 2003 Service Pack 1?

Computer-wide restrictions

A change has been made in COM to provide computer-wide access controls that govern access to all call, activation, or launch requests on the computer. The simplest way to think about these access controls is as an additional **AccessCheck** call that is done against a computer-wide access control list (ACL) on each call, activation, or launch of any COM server on the computer. If the **AccessCheck** fails, the call, activation, or launch request is denied. This is in addition to any **AccessCheck** that is run against the server-specific ACLs. In effect, it provides a minimum authorization standard that must be passed to access any COM server on the computer. There is a computer-wide ACL for launch permissions to cover activate and launch rights, and a computer-wide ACL for access permissions to cover call rights. These can be configured through the Component Services Microsoft Management Console (MMC).

These computer-wide ACLs provide a way to override weak security settings specified by a specific application through **CoInitializeSecurity** or application-specific security settings. This provides a minimum security standard that must be passed, regardless of the settings of the specific server.

These ACLs are checked when the interfaces exposed by RPCSS are accessed. This provides a method to control access to this system service.

These ACLs provide a centralized location where an administrator can set general authorization policy that applies to all COM servers on the computer.

Note

Changing the computer-wide security settings will affect all COM server applications, and might prevent them from working properly. If there are COM server applications that have restrictions that are less stringent than the computer-wide restrictions, reducing the computer-wide restrictions might expose these applications to unwanted access. Conversely, if you increase the computer-wide restrictions, some COM server applications might no longer be accessible by calling applications.

By default, Windows XP SP2 computer restriction settings are:

Permission	Administrator	Everyone	Anonymous
Launch	Local Launch Local Activation Remote Launch Remote Activation	Local Launch Local Activation	
Access		Local Access Remote Access	Local Access

By default, Windows Server 2003 SP 1 computer restriction settings are as follows.

Permission	Administrator	Distributed COM Users (Built in Group)	Everyone	Anonymous
Launch	Local Launch Local Activation Remote Launch Remote Activation	Local Launch Local Activation Remote Launch Remote Activation	Local Launch Local Activation	N/A
Access	N/A	Local Access Remote Access	Local Access Remote Access	Local Access Remote Access

Note

Distributed COM Users is a new built-in group included with Windows Server 2003 SP1 to expedite the process of adding users to the DCOM computer restriction settings. This group is part of the ACL used by the MachineAccessRestriction and MachineLaunchRestriction settings, so removing users from this group will affect those settings.

Why is this change important? What threats does it help mitigate?

Many COM applications include some security-specific code (for example, calling **CoInitializeSecurity**), but use weak settings, often allowing unauthenticated access to the process. There is currently no way for an administrator to override these settings to force stronger security in earlier versions of Windows.

COM infrastructure includes the RPCSS, a system service that runs during system startup and always runs after that. It manages activation of COM objects and the running object table and provides helper services to DCOM remoting. It exposes RPC interfaces that can be called remotely. Because some COM servers allow unauthenticated remote access, these interfaces can be called by anyone, including unauthenticated users. As a result, RPCSS can be attacked by malicious users on remote, unauthenticated computers.

In earlier versions of Windows, there was no way for an administrator to understand the exposure level of the COM servers on a computer. An administrator got an idea of the exposure level by systematically checking the configured security settings for all the registered COM applications on the computer, but, given that there are about 150 COM servers in a default installation of Windows, that task was daunting. There was no way to view the settings for a server that incorporates security in the software, short of reviewing the source code for that software.

DCOM computer-wide restrictions mitigate these three problems. It also gives an administrator the capability to disable incoming DCOM activation, launch, and calls.

What works differently?

By default, the Everyone group is granted local launch, local activation, and local access call permissions. This enables all local scenarios to work without modification to the software or the operating system.

By default, in Windows XP SP2, the Everyone group is granted remote access call permissions. In Windows Server 2003 SP1, the Everyone and Anonymous groups are granted remote access permissions. This enables most COM client scenarios, including the common case where a COM client passes a local reference to a remote server, in effect turning the client into a server. In Windows XP SP2, this might disable scenarios that require unauthenticated remote access calls.

Also by default, only members of the Administrators group are granted remote activation and launch permissions. This disables remote activations by non-administrators to installed COM servers.

How do I resolve these issues?

If you implement a COM server and expect to support remote activation by a non-administrative COM client, then you should consider whether the risk associated with enabling this process is acceptable or if you should modify your implementation to not require remote activation by a non-administrative COM client or remote unauthenticated calls.

If the risk is acceptable and you want to enable remote activation by a non-administrative COM client or remote unauthenticated calls, you must change the default configuration for this feature.

Note

Changing the computer-wide security settings will affect all COM server applications, and might prevent them from working properly. If there are COM server applications that have restrictions that are less stringent than the computer-wide restrictions, reducing the computer-wide restrictions may expose these applications to unwanted access. Conversely, if you increase the computer-wide restrictions, some COM server applications might no longer be accessible by calling applications.

You can change the configuration settings using either the Component Services Microsoft Management Console (MMC) or the Windows registry.

If you use the Component Services MMC snap-in, these settings can be configured on the **COM Security** tab of the **Properties** dialog box for the computer you are managing. The **Access Permissions** area has been modified to enable you to set computer-wide limits in addition to the standard default settings for COM servers. Additionally, you can provide separate ACL settings for local and remote access under both limits and defaults.

In the **Launch and Activation Permissions** area, you can control the local and remote permissions as well as the computer-wide limits and the defaults. You can specify both local and remote activation and launch permissions independently.

Alternatively, you can configure these ACL settings using the registry.

These ACLs are stored in the registry at the following locations:

```
 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole\MachineAccessRestriction=ACL
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole\MachineLaunchRestriction=ACL
```

These are named values of type REG_BINARY that contain data that describe the ACL of the principals that can access any COM class or COM object on the computer. The access rights in the ACL are:

```
 COM_RIGHTS_EXECUTE 1
COM_RIGHTS_EXECUTE_LOCAL 2
COM_RIGHTS_EXECUTE_REMOTE 4
COM_RIGHTS_ACTIVATE_LOCAL 8
COM_RIGHTS_ACTIVATE_REMOTE 16
```

These ACLs can be created using normal security functions.

Note

To provide backward compatibility, an ACL can exist in the format used before Windows XP SP2 and Windows Server 2003 SP1, which uses only the access right COM_RIGHTS_EXECUTE, or it can exist in the new format used in Windows XP SP2 and Windows Server 2003 SP1, which uses COM_RIGHTS_EXECUTE together with a combination of COM_RIGHTS_EXECUTE_LOCAL, COM_RIGHTS_EXECUTE_REMOTE, COM_RIGHTS_ACTIVATE_LOCAL, and COM_RIGHTS_ACTIVATE_REMOTE. Note that COM_RIGHTS_EXECUTE> must always be present; the absence of this right generates an invalid security descriptor. Also note that you must not mix the old format and the new format within a single ACL; either all access control entries (ACEs) must grant only the COM_RIGHTS_EXECUTE access right, or they all must grant COM_RIGHTS_EXECUTE together with a combination of COM_RIGHTS_EXECUTE_LOCAL, COM_RIGHTS_EXECUTE_REMOTE, COM_RIGHTS_ACTIVATE_LOCAL, and COM_RIGHTS_ACTIVATE_REMOTE.

Note

Only users with Administrator rights can modify these settings.

What existing functionality is changing in Windows XP Service Pack 2 and Windows Server 2003 Service Pack 1?

RPCSS runs as a network service

RPCSS is a key service for the RPC Endpoint Mapper and DCOM infrastructure. This service ran as Local System in previous versions of Windows. To reduce the attack surface of Windows and provide defense in depth, the RPCSS service functionality was split into two services. The RPCSS service with all the original functionality that did not require Local System privileges now runs under the Network Service account. A new DCOMLaunch service that includes functionality that requires Local System privileges runs under the Local System account.

Why is this change important?

This change reduces the attack surface and provides defense in depth for the RPCSS service because an elevation of privilege in the RPCSS service is now limited to the Network Service privilege.

What works differently?

This change should be transparent to users because the combination of the RPCSS and DCOMLaunch services is equivalent to the previous RPCSS service provided in prior versions of Windows.

More specific COM permissions

COM server applications have two types of permissions: launch permissions and access permissions. Launch permissions control authorization to start a COM server during COM activation if the server is not already running. These permissions are defined as security descriptors that are specified in registry settings. Access permissions control authorization to call a running COM server. These permissions are defined as security descriptors provided to the COM infrastructure through the **CoInitializeSecurity** API, or using registry settings. Both launch and access permissions allow or deny access based on principals, and make no distinction as to whether the caller is local to the server or remote.

The first change distinguishes the COM access rights, based on distance. The two distances that are defined are Local and Remote. A Local COM message arrives by way of the Local Remote Procedure Call (LRPC) protocol, while a Remote COM message arrives by way of a remote procedure call (RPC) host protocol like transmission control protocol (TCP).

COM activation is the act of getting a COM interface proxy on a client by calling **CoCreateInstance** or one of its variants. As a side effect of this activation process, sometimes a COM server must be started to fulfill the client's request. A launch permissions ACL asserts who is allowed to start a COM server. An access permissions ACL asserts who is allowed to activate a COM object or call that object once the COM server is already running.

The second change is that the call and activation rights are separated to reflect to two distinct operations and to move the activation right from the access permission ACL to the launch permission ACL. Because activation and launching are both related to acquiring an interface pointer, activation and launch access rights logically belong together in one ACL. And because you always specify launch permissions through configuration (as compared to access permissions, which are often specified programmatically), putting the activation rights in the launch permission ACL provides the administrator with control over activation.

The launch permission access control entries (ACEs) are broken into four access rights:

- Local Launch (LL)
- Remote Launch (RL)
- Local Activate (LA)
- Remote Activate (RA)

The access permission security descriptor is split into two access rights:

- Local Access Calls (LC)
- Remote Access Calls (RC)

This allows the administrator to apply very specific security configurations. For example, you can configure a COM server so that it accepts local access calls from everyone, while only accepting remote access calls from Administrators. These distinctions can be specified through changes to the COM permissions security descriptors.

Why is this change important? What threats does it help mitigate?

Earlier versions of the COM server application have no way to restrict an application so that it can only be used locally without exposing the application on the network by way of DCOM. When a user has access to a COM server application, they have access for both local and remote use.

A COM server application might expose itself to unauthenticated users to implement a COM callback scenario. In this scenario, the application must also expose its activation to unauthenticated users, which might not be desirable.

Precise COM permissions give flexibility to the administrator to control a computer's COM permission policy. These permissions enable security for the described scenarios.

What works differently? Are there any dependencies?

To provide backward compatibility, existing COM security descriptors are interpreted to allow or deny both local and remote access simultaneously. That is, an access control entry (ACE) either allows both local and remote, or denies both local and remote.

There are no backward-compatibility issues for call or launch rights. There is, however, an activation rights compatibility issue. If, in the existing security descriptors for a COM server, the configured launch permissions are more restrictive than the access permissions and are more restrictive than what is minimally required for client activation scenarios, then the launch permissions ACL must be modified to give the authorized clients the appropriate activation permissions.

For COM applications that use the default security settings, there are no compatibility issues. For applications that are dynamically started using COM activation, most have no compatibility issues, because the launch permissions must already include anyone who is able to activate an object. Otherwise, such applications generate activation failures even before applying Windows XP SP2 or Windows Server 2003 SP1, when callers without launch permission try to activate an object and the COM server is not already running.

The applications of most concern for compatibility issues are COM applications that are already started by some other mechanism, such as Windows Explorer, or Service Control Manager. You can also start these applications by a previous COM activation, which overrides the default access and launch permissions and specifies launch permissions that are more restrictive than the call permissions. For more details about addressing this compatibility issue, see "How do I resolve these issues?" in the next section.

If a system that is upgraded to Windows XP SP2 or Windows Server 2003 SP1 is rolled back to an earlier state, any ACE that was edited to allow local access, remote access, or both, is interpreted to allow both local and remote access. Any ACE that was edited to deny local access, remote access, or both, is interpreted to deny both local and remote access. Whenever you uninstall a service pack, you should ensure that no newly set ACEs cause applications to stop working.

How do I resolve these issues?

If you implement a COM server and you override the default security settings, confirm that the application-specific launch permissions ACL grants activation permission to appropriate users. If it does not, you must change your application-specific launch permission ACL to give appropriate users activation rights so applications and Windows components that use DCOM do not fail. These application-specific launch permissions are stored in the registry.

The COM ACLs can be created or modified using normal security functions.

What settings are added or changed in Windows XP Service Pack 2?

Caution

Improper use of these settings can cause applications and Windows components that use DCOM to fail.

In the following table, these abbreviations are used:

LL - Local Launch

LA - Local Activation

RL - Remote Launch

RA - Remote Activation

LC - Local Access Calls

RC - Remote Access Calls

ACL - Access Control List

Setting name	**Location**	**Previous default value**	**Default value**	**Possible valu**
MachineLaunchRestriction	**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole**	Everyone - LL, LA, RL, RA Anonymous - LL, LA, RL, RA (This is a new registry key. Based on existing behavior, these are the effective values.)	Administrator: LL, LA, RL, RA	ACL
MachineAccessRestriction	**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole**	Everyone - LC, RC Anonymous - LC, RC (This is a new registry key. Based on existing behavior, these are the effective values.)	Everyone: LC, RC Anonymous: LC	ACL
CallFailureLoggingLevel	**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole**	Not applicable.	This registry key is not present; however, a missing key or value is interpreted as 2. This event is not logged by default. If you change this value to 1 to start logging this information to help you	1 - Always lo failures durin COM Server 2 - Never log failures durin call server pr

Setting name	Location	Previous default value	Default value	Possible valu
			troubleshoot an issue, be sure to monitor the size of your event log, because this is an event that can generate a large number of entries.	
InvalidSecurityDescriptorLoggingLevel	**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole**	Not applicable.	This registry key is not present; however, a missing key or value is interpreted as 1. This event is logged by default. It should rarely occur.	1 - Always lo failures wher infrastructure invalid securi 2 - Never log failures wher infrastructure invalid securi
DCOM:Machine Launch Restrictions in Security Descriptor Definition Language (SDDL) Syntax	(Group Policy object) Computer Configuration \Windows Settings\Local Policies \Security Options	Not applicable.	Not defined	Access contr SDDL forma this policy ov in MachineLau previously.
DCOM:Machine Access Restrictions in Security Descriptor Definition Language (SDDL) Syntax	(Group Policy object) Computer Configuration \Windows Settings \Local Policies \Security Options	Not applicable.	Not defined	Access contr SDDL forma this policy ov in MachineAcc previously.

What settings are added or changed in Windows Server 2003 Service Pack 1?

Note

Improper use of these settings can cause applications and Windows components that use DCOM to fail.

In the following table, these abbreviations are used:

LL - Local Launch

LA - Local Activation

RL - Remote Launch

RA - Remote Activation

LC - Local Access Calls

RC - Remote Access Calls

ACL - Access Control List

Setting name	Location	Previous default value	Default value	Possible val
MachineLaunchRestriction	**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole**	Everyone: LL, LA, RL, RA Anonymous: LL, LA, RL, RA (This is a new registry key. Based on existing behavior, these would be the effective values.)	Administrator: LL, LA, RL, RA Everyone: LL, LA Distributed COM users: LL, LA, RL, RA	ACL
MachineAccessRestriction	**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole**	Everyone: LC, RC Anonymous: LC, RC (This is a new registry key. Based on existing behavior, these would be the effective values.)	Everyone: LC, RC Anonymous: LC, RC	ACL
CallFailureLoggingLevel	**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole**	Not applicable.	This registry key is not present; however, a missing key or value is interpreted as 2. This event is not logged by default. If you change this value to 1 to start logging this information to help you troubleshoot an issue, be sure to monitor the size of your event log,	1 - Always l failures whe infrastructur invalid secu 2 - Never lo failures whe infrastructur invalid secu

Setting name	Location	Previous default value	Default value	Possible val
			because this is an event that can generate a large number of entries.	
InvalidSecurityDescriptorLoggingLevel	**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole**	Not applicable.	This registry key is not present; however, a missing key or value is interpreted as 1. This event is logged by default. It should rarely occur.	1 - Always l failures whe infrastructur invalid secu 2 - Never log failures whe infrastructur invalid secu
DCOM:Machine Launch Restrictions in Security Descriptor Definition Language (SDDL) Syntax	(Group Policy object) Computer Configuration \Windows Settings \Local Policies \Security Options	Not applicable.	Not defined.	Access contr SDDL forma this policy o in MachineLau previously.
DCOM:Machine Access Restrictions in Security Descriptor Definition Language (SDDL) Syntax	(Group Policy object) Computer Configuration \Windows Settings \Local Policies \Security Options	Not applicable.	Not defined.	Access contr SDDL forma this policy o in MachineAcc previously.

Security in COM
