Changes to Functionality in Microsoft Windows XP Service Pack 2

Part 2: Network Protection Technologies

Published: August 9, 2004 | Updated: November 4, 2004
By Starr Andersen, Technical Writer; Vincent Abella, Technical Editor

This document is Part 2 of "Changes to Functionality in Windows XP Service Pack 2" and provides detailed information about the network protection technologies included in Microsoft® Windows XP Service Pack 2. You can obtain the other parts of the paper in the Microsoft Download Center, at http://go.microsoft.com/fwlink/?LinkId=28022.

This document applies to Microsoft Windows® XP Service Pack 2 (SP2) for the 32-bit versions of Windows XP Professional and Windows XP Home Edition. It does not describe all of the changes that are included in the service pack, but instead highlights those changes that will have the most impact on your use of Windows XP SP2 and provides references to additional information that may be available.

On This Page

In This Article

Alerter and Messenger Services

What do the Alerter and Messenger Services do?

The Alerter and Messenger services are components of Windows that allow simple messages to be communicated between computers on a network. The Messenger service relays messages from different applications and services, while the Alerter service is intended specifically for administrative alerts.

Who does this feature apply to?

Administrators that communicate with their users should be aware of the changes to these services. In addition, developers that use these services to notify users about events or broadcast messages on the network should be aware of these changes. Although these changes apply to all computers running Microsoft Windows XP Service Pack 2, only computers connected to a network are affected.

What existing functionality is changing in Windows XP Service Pack 2?

Alerter and Messenger Services Disabled
Detailed description

In previous versions of Windows, the Messenger service is set to start automatically and the Alerter service is set to manual start. In Windows XP Service Pack 2, both of these services are set to Disabled. No other changes are made to these services.

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

When the services are started, they allow incoming network connections and present an attack surface. This elevates their security risk. Also, these services are used infrequently in current computing environments. Because of the additional attack surface that the services present, and their limited general use, they are now disabled by default.

What works differently? Are there any dependencies?

Any applications or services that use the Alerter or Messenger services to communicate with the user will not be successful by default.

How do I resolve these issues?

There are two possible avenues to resolve the issue. The recommended resolution is to revise the software to use another method to communicate with the user. This allows you to communicate with the user with enhanced security, without having to use the Alerter or Messenger services.

The second way is to have the application start the Alerter or Messenger service before making use of its services. Information on starting services can be found in online Help and in MSDN. For an example, see "Using the Services Administrative Tool to Configure Services" on the Microsoft Web site at **http://go.microsoft.com/fwlink/?LinkId=25974**.

Do I need to change my code to work with Windows XP Service Pack 2?

If your code makes use of the Messenger or Alerter services, you may need to change your code. For more information, see "How do I resolve these issues?" above.

Bluetooth

What does Bluetooth do?

Bluetooth® wireless technology is a low cost, short-range wireless specification for connecting mobile devices and is available in a wide variety of devices. Support for Bluetooth wireless technology is included in Windows XP Service Pack 2. This support was not previously available directly from Microsoft. It is included now because customers requested that this technology be added to the core Windows operating system.

With this release, you can:

- Connect a Bluetooth device to a computer.
- Create a wireless desktop with a Bluetooth keyboard and mouse.
- Transfer files to or from a Bluetooth device.
- Print to a Bluetooth printer.
- Connect to a computer network or the Internet through a Bluetooth mobile phone.
- Set up an Internet Protocol (IP) connection to the Internet through a Bluetooth mobile phone.

If the appropriate Microsoft or non-Microsoft software programs are installed on Windows XP, you can also perform other operations with Bluetooth devices, such as:

- Synchronizing contacts and calendars with a Bluetooth mobile phone or personal digital assistant (PDA).
- Reading coordinates from a GPS receiver.

This release also has support for these Bluetooth profiles:

- **Personal Area Networking (PAN)**. Enables IP connections over Bluetooth wireless technology.
- **Hard Copy Replacement Profile (HCRP)**. Enables printing.
- **Host Interface Device (HID)**. Enables Bluetooth keyboards, mice, and joysticks.
- **Dial-Up Networking.** Enables Bluetooth mobile phones to work as modems.
- **Object Push Profile (OPP)**. Enables file transfers.
- **Virtual COM ports (SPP)**. Enables legacy programs to communicate with Bluetooth devices.

In addition, these Bluetooth features are included:

- **Selective suspend**. Reduces the power consumption of Bluetooth transceivers connected to the computer by means of a Universal Serial Bus (USB) connection.
- **Boot-mode keyboards**. Enables specifically-configured Bluetooth keyboards to work with the BIOS.

If no Bluetooth transceiver is present on the system, there is no change to the system's behavior. When a Bluetooth device that is approved by the Windows Hardware Quality Labs (WHQL) is present, Bluetooth support is enabled.

When Bluetooth support is enabled, you can find changes in Network Connections in Control Panel. In addition, a new Control Panel item called **Bluetooth Devices** has also been added. You will also find a **Bluetooth** icon in the taskbar notification area. When you click this icon, you will see a menu of Bluetooth tasks you can perform. You can also start the new Bluetooth File Transfer Wizard. To do this, click **Start**, point to **Accessories**, point to **Communications**, and then select **Bluetooth File Transfer Wizard**.

If an existing non-Microsoft Bluetooth driver is installed, upgrading to Windows XP Service Pack 2 does not cause the existing driver to be replaced. It can be replaced later, either manually or programmatically.

For complete documentation on Bluetooth in Windows XP Service Pack 2, see online Help.

Top of page

Client Administrative Tools

What do the client administrative tools do?

The client administrative tools are a set of Microsoft Management Console (MMC) snap-ins that allow you to administer users, computers, services, and other system components on local and remote computers. Two system-generated dialog boxes that these snap-ins use for management are **Select Users, Computers, or Groups** and **Find Users, Contacts, and Groups**. **Select Users, Computers, or Groups** is used when setting access control lists (ACLs) on a shared folder, specifying a remote computer for retargeting a snap-in, or managing local users and groups. **Find Users, Contacts, and Groups** is used to search Active Directory in My Network Places, find a printer in the Add a Printer Wizard, and find objects in the directory within the Active Directory Users and Computers snap-in.

Both dialog boxes are used to find and select objects such as users, computers, printers, and other security principals from the local computer or Active Directory. Although other applications can use these dialog boxes, only the changes to the client administrative tools that are listed here are covered by this section.

Who do these features apply to?

These features apply to administrators who need to manage Windows XP from a remote location using the affected administrative tools, which are listed below. Administrators and users who are using these tools to manage the local computer are not affected.

What existing functionality is changing in Windows XP Service Pack 2?

Remote connectivity
Detailed description

For the administrative tools that are listed here to connect to a remote computer, that remote computer must allow incoming network traffic on TCP port 445. However, the default configuration of Windows Firewall in Windows XP Service Pack 2 blocks incoming network traffic on TCP port 445. As a result, you might receive one or more of the following error messages. When you receive one of these messages, the text that is italicized in the example messages below will be replaced with the system variable appropriate to the error condition:

- **Unable to access the computer** *Computer_Name*. The error was **Access is denied**.
- **Unable to access the computer** *Computer_Name*. The error message previously said **The network path was not found**.
- **Failed to open Group Policy object on***Computer_Name*. **You might not have appropriate rights.**
- **Details: The network path was not found.**
- **An object (Computer) with the following name cannot be found:** "*Computer_Name*." **Check the selected object types and location for accuracy and ensure that you have typed the object name correctly, or remove this object from the selection.**
- **Computer** *Computer_Name* **cannot be managed. The network path was not found. To manage a different computer, on the** Action **menu, click** Connect to another computer.
- **System error 53 has occurred. The network path was not found.**

These errors can occur when one of the following MMC snap-ins is used for remote administration:

- Certificates
- Computer Management
- Device Manager
- Disk Management
- Event Viewer
- Group Policy
- Indexing Service
- IP Security Monitor
- IP Security Policy
- Local Users & Groups
- Removable Storage Management
- Resultant Set of Policy
- Services
- Shared Folders
- WMI Control

In addition to the MMC snap-ins, these dialog boxes and administrative tools are affected:

- Select Users, Computers, or Groups
- Find Users, Contacts, and Groups
- Net.exe

How do I resolve these issues?

To use these tools to remotely connect a computer running Windows XP with Windows Firewall enabled, you need to open TCP port 445 in the firewall on the remote computer. To do this, use the following procedure:

1. Click Start, point to All Programs, point to Accessories, and click Command Prompt.
2. At the command prompt, type **netsh firewall set portopening TCP 445 ENABLE** and then press ENTER.

   Note Open firewall ports can be a security vulnerability. You should carefully plan and test any such configuration change before it is implemented.

Top of page

DCOM Security Enhancements

What does DCOM do?

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. For more information, see "Component Object Model" on the Microsoft Web site at http://go.microsoft.com/fwlink/?LinkId=20922

Who does this feature apply to?

If you only use COM for in-process COM components, this section 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 permission that is necessary to run it.
- The application is usually activated on a computer running Microsoft Windows XP by a remote COM client without using an administrative account.
- By default, the application uses unauthenticated remote callbacks on a computer running Windows XP.
- The application is only meant to be used 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?

Computerwide restrictions
Detailed description

A change has been made in COM to provide computerwide 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 computerwide 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 will be 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 will be a computerwide 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 computerwide 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 who has 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.

By default, Windows XP computer restriction settings are:

Permission	Administrator	Everyone	Anonymous
Launch	Local (Launch) Local Activate Remote (Launch) Remote Activate	Local (Launch) Local Activate	
Access		Local (Call) Remote (Call)	Local (Call)

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 (as explained in the previous section), these interfaces can be called by anyone, including unauthenticated users. As a result, RpcSs can be attacked by malicious users using 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 could get 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 XP, 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 computerwide restrictions mitigates 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 call permissions. This should enable all local scenarios to work without modification to the software or the operating system.

By default, the Everyone group is granted remote call 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. This might disable scenarios that require unauthenticated remote 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 or remote unauthenticated calls, 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 will need to change the default configuration for this feature.

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. Security Settings provides the ability to 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

This is a named-value that is set to a REG_BINARY type that contains data describing 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 that COM_RIGHTS_EXECUTE rights must always be present, because absence of this right will generate an invalid security descriptor.

Only users with Administrator rights can modify these settings.

What existing functionality is changing in Windows XP Service Pack 2?

RPCSS runs as a network service
Detailed description

In Windows XP SP2, 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 since an elevation of privilege in the RPCSS service will now be limited to the Network Service privilege.

What works differently?

This change should be transparent to users since the combination of the RPCSS and DCOMLaunch services in Windows XP Service Pack 2 is equivalent to the previous RPCSS service provided in prior versions of Windows.

More specific COM permissions
Detailed description

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.

Another 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 satisfy 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.

Another change is that the call and activation rights are being separated to reflect to two distinct operations and to move the activation right from the access permission ACL to the launch permission ACL. Since 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 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 Calls (LC)
- Remote Calls (RC)

This COM security allows the administrator to apply very specific security configurations. For example, you can configure a COM server so that it accepts local calls from everyone, while only accepting remote 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 want to 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, because malicious users could use that scenario to gain unauthorized access to that server.

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

There are no backwards-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 needed for client activation scenarios, then the Launch Permissions ACL must be modified to give the authorized clients the appropriate 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 will have no compatibility issues, since the launch permissions must already include anyone who is able to activate an object. If these permissions are not configured correctly, there might be random activation failures when callers without

launch permission try to activate an object when the COM server is not already running.

The applications of most concern for compatibility issues are COM applications that are already started by way of some other mechanism, such as Windows Explorer, or Service Control Manager. You can also start these applications by way of 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 on addressing this compatibility issue, see "How do I resolve these issues?" in the next section.

If a system that was upgraded to Windows XP Service Pack 2 is rolled back to an earlier service pack, any access control entry that was edited to allow local access, remote access, or both, will be interpreted to allow both local and remote access. Any ACE that was edited to deny local access, remote access, or both, will be interpreted to deny both local and remote access. Whenever you uninstall a service pack, you should ensure that no newly-set ACEs will 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 will need to 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. For more information about launch permissions, see "LaunchPermission" on the MSDN Web site at **http://go.microsoft.com/fwlink/?LinkId=20924**.

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 Control

RC - Remote Control

Setting name	Location	Previous default value	Default value	Possible values
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**	**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**	**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	**1** - Always log event log failures during a call in the COM Server process. **2** - Never log event log failures during a call in the call server process.

Setting name	Location	Previous default value	Default value	Possible values
			value to **1** to start logging this information to help you troubleshoot an issue, be sure to monitor the size of your event log, as this is an event that can generate a large number of entries.	
InvalidSecurity Descriptor LoggingLevel	**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 log event log failures when COM infrastructure finds an invalid security descriptor. **2** - Never log event log failures when COM infrastructure finds an invalid security descriptor.
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 Control List in SDDL Format. Existence of this policy, overrides, values in **MachineLaunch Restriction**, above.
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 Control List in SDDL Format. Existence of this policy, overrides, values in **MachineAccess Restriction**, above.

Top of page

TCP/IP

What does TCP/IP do?

Transmission Control Protocol/Internet Protocol (TCP/IP) is a suite of standard protocols for connecting computers across networks. TCP/IP enables Windows-based computers to connect and share information with other Microsoft and non-Microsoft systems.

Who does this feature apply to?

All users who use TCP/IP to connect and communicate information over a network should be aware of the changes incorporated in Windows XP Service Pack 2.

What new functionality is added to this feature in Windows XP Service Pack 2?

Restricted traffic over raw sockets
Detailed description

A very small number of Windows applications make use of raw IP sockets, which provide an industry-standard way for applications to create TCP/IP packets with fewer integrity and security checks by the TCP/IP stack. The Windows implementation of TCP/IP still supports receiving traffic on raw IP sockets. However, the ability to send traffic over raw sockets has been restricted in two ways:

- TCP data cannot be sent over raw sockets.
- UDP datagrams with invalid source addresses cannot be sent over raw sockets. The IP source address for any outgoing UDP datagram must exist on a network interface or the datagram is dropped.

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

This change limits the ability of malicious code to create distributed denial-of-service attacks and limits the ability to send spoofed packets, which are TCP/IP packets with a forged source IP address.

Limited number of simultaneous incomplete outbound TCP connection attempts
Detailed description

The TCP/IP stack now limits the number of simultaneous incomplete outbound TCP connection attempts. After the limit has been reached, subsequent connection attempts are put in a queue and will be resolved at a fixed rate. Under normal operation, when applications are connecting to available hosts at valid IP addresses, no connection rate-limiting will occur. When it does occur, a new event, with ID 4226, appears in the system's event log.

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

This change helps to limit the speed at which malicious programs, such as viruses and worms, spread to uninfected computers. Malicious programs often attempt to reach uninfected computers by opening simultaneous connections to random IP addresses. Most of these random addresses result in a failed connection, so a burst of such activity on a computer is a signal that it may have been infected by a malicious program.

What works differently?

This change may cause certain security tools, such as port scanners, to run more slowly.

How do I resolve these issues?

Stop the application that is responsible for the failing connection attempts.

Winsock self-healing
Detailed description

Winsock, Windows' network socket facility for applications, is extensible by a mechanism known as a Layered Service Provider (LSP). Winsock LSPs are available for a wide range of useful purposes, including internet parental controls, and web content filtering. In previous versions of Windows XP, removing a malformed (also known as "buggy") LSP could result in corruption of the Winsock catalog in the registry, potentially resulting in a loss of all network connectivity. Winsock now has the ability to self-heal after a user uninstalls such an LSP.

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

Customers need to be able to safely remove from their systems poorly implemented LSPs.

New Winsock Netsh commands
Detailed description

Two new Netsh commands are available in Windows XP Service Pack 2.

- **netsh winsock reset catalog**

  This command resets the Winsock catalog to the default configuration. This can be useful if a malformed LSP is installed that results in loss of network connectivity. While use of this command can restore network connectivity, it should be used with care because any previously-installed LSPs will need to be re-installed.

- **netsh winsock show catalog**

  This command displays the list of Winsock LSPs that are installed on the computer.

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

These commands provide additional management capabilities for maintaining and troubleshooting Winsock LSPs and can be used in a script to aid in recovering from a widespread installation of malformed LSPs.

Top of page

RPC Interface Restriction

What does RPC Interface Restriction do?

A number of changes have been made in the Remote Procedure Call (RPC) service for Windows XP Service Pack 2 that help make RPC interfaces secure by default and reduce the attack surface of Windows XP. The most significant change is the addition of the **RestrictRemoteClients** registry key. This key modifies the behavior of all RPC interfaces on the system and will, by default, eliminate remote anonymous access to RPC interfaces on the system, with some exceptions. Additional changes include the **EnableAuthEpResolution** registry key and three new interface registration flags.

Who does this feature apply to?

This feature applies to RPC application developers. System administrators should also be familiar with this change to RPC.

What new functionality is added to this feature in Windows XP Service Pack 2?

RestrictRemoteClients Registry Key
Detailed description

When an interface is registered using **RpcServerRegisterIf**, RPC allows the server application to restrict access to the interface, typically through a security callback. The **RestrictRemoteClients** registry key forces RPC to perform additional security checks for all interfaces, even if the interface has no registered security callback.

RPC clients that use the named pipe protocol sequence (ncacn_np) are exempt from all restrictions discussed in this section. The named pipe protocol sequence cannot be restricted by default, due to several significant backwards compatibility issues.

The **RestrictRemoteClients** registry key can have one of three DWORD values that can also be controlled programmatically in rpcdce.h. If the key is not present, it is equivalent to setting the DWORD=1 value (**RPC_RESTRICT_REMOTE_CLIENT_DEFAULT**).

Key reference

Key name: **RestrictRemoteClients**

Type: DWORD

Configurable through User Interface: Yes. This key can be configured using the Group Policy Object Editor.

Default value: 1

Meaning: This value is the default value in Windows XP Service Pack 2. Restricts access to all RPC interfaces. All remote anonymous calls are rejected by the RPC runtime. This corresponds to the value **RPC_RESTRICT_REMOTE_CLIENT_DEFAULT** in rpcdce.h. If an interface registers a security callback and provides the **RPC_IF_ALLOW_CALLBACKS_WITH_NO_AUTH** flag, then this restriction does not apply to that interface.

Value: **0**

Meaning: Causes the system to bypass the RPC interface restriction. This corresponds to the value **RPC_RESTRICT_REMOTE_CLIENT_NONE** in rpcdce.h. It is entirely the responsibility of the server application to impose appropriate RPC restrictions. This setting is equivalent to the behavior in previous versions of Windows.

Value: **2**

Meaning: All remote anonymous calls are rejected by the RPC runtime with no exemptions. This corresponds to the value **RPC_RESTRICT_REMOTE_CLIENT_HIGH** in rpcdce.h. When this value is set, a system cannot receive remote anonymous calls using RPC.

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

It is much more difficult to attack an interface if you require calls to perform authentication, even a relatively low level of authentication. This is a particularly useful mitigation against worms which rely on exploitable buffer overruns that can be invoked remotely through anonymous connections.

What works differently?

If your RPC application expects to receive calls from remote anonymous RPC clients, this change might not allow your application to run correctly. As a result, applications that use DCOM might not work correctly if this value is set.

Because secure RPC calls over connectionless protocols such as UDP and IPX (ncadg_ip_udp and ncadg_ipx) use a lower level of security than calls over connection-oriented protocols these calls are always considered non-secure for the purposes of this policy. As a result, RPC calls over connectionless protocols will fail by default in Windows XP SP2.

To allow RPC client calls using connectionless protocols set the RestrictRemoteClients value to 0 (RPC_RESTRICT_REMOTE_CLIENT_NONE).

How do I resolve these issues?

There are three options to resolve these issues. These options are listed in order of preference.

- Require your RPC clients to use RPC security when contacting your server application. This is the best method to mitigate security threats.
- Exempt your interface from requiring authentication by setting the **RPC_IF_ALLOW_CALLBACKS_WITH_NO_AUTH** flag during interface registration. This configures RPC to allow anonymous connections to only your application's interface.
- Force RPC to exhibit the same behavior as earlier versions of Windows by setting the registry key to **RPC_RESTRICT_REMOTE_CLIENT_NONE (0)**. RPC will then accept anonymous connections to all interfaces. This option should be avoided if possible, as it reduces the overall security of the computer.

EnableAuthEpResolution Registry Key
Detailed description

An RPC interface that is remotely and anonymously accessible and is registered by default on Windows XP presents a significant attack surface. RPC itself must register such an interface to provide endpoint resolution for calls using dynamic endpoints.

With the addition of the **RestrictRemoteClients** flag, by default, the RPC Endpoint Mapper interface is not accessible anonymously. This is a significant security improvement, but it changes the task of resolving an endpoint. Currently, an RPC client that attempts to make a call using a dynamic endpoint will first query the RPC Endpoint Mapper on the server to determine what endpoint it should connect to. This query is performed anonymously, even if the RPC client call itself is performed using RPC security.

Anonymous calls to the RPC Endpoint Mapper interface will fail by default on Windows XP Service Pack 2 because of the default value for the new **RestrictRemoteClients** key. This makes it necessary to modify

the RPC client runtime to perform an authenticated query to the Endpoint Mapper. If the **EnableAuthEpResolution** key is set, the RPC client runtime will use NTLM to authenticate to the endpoint mapper. This authenticated query will only take place if the actual RPC client call uses RPC authentication.

Why is this change important?

This change is required to enable an RPC client to make a call to an RPC server which has registered a dynamic endpoint on a system running Windows XP Service Pack 2. The client computer must set this registry key so that it will perform an authenticated query to the RPC Endpoint Mapper.

What works differently?

This registry key is used to enable the specific scenario described in the previous section. When this key is turned on, all RPC Endpoint Mapper queries that are performed on behalf of authenticated calls are performed using NTLM authentication.

This setting can also be specified using the Group Policy Object Editor to configure the Group Policy object located in Computer Configuration \Administrative Templates \System\Remote Procedure Call\RPC Endpoint Mapper Client Authentication.

New RPC Interface Registration Flags
Detailed description

Three new interface registration flags have been created which make it easier for an application developer to secure an RPC interface.

- **RPC_IF_ALLOW_CALLBACKS_WITH_NO_AUTH**

  When this flag is registered, the RPC runtime invokes the registered security callback for all calls, regardless of the call security settings. Without this flag, RPC rejects all unauthenticated calls before they reach the security callback. This flag works only when a security callback is registered.

- **RPC_IF_SEC_NO_CACHE**

  A security callback is registered for an interface in order to restrict access to that interface. The typical security callback impersonates the client to determine if the client has sufficient rights to make a call to the interface. If a particular client identity passes a security callback once, it usually passes the same security callback every time.

  The RPC runtime takes advantage of this pattern by remembering when an individual client identity passes a security callback and skips the security callback for subsequent calls by that client to the same interface. This feature is called *security callback caching* and has existed since Windows 2000. For Windows XP Service Pack 2, you can use the **RPC_IF_SEC_NO_CACHE** flag to disable security callback caching for a given interface. This is useful if the security check might change, possibly rejecting a client identity which was previously permitted.

- **RPC_IF_LOCAL_ONLY**

  When an interface is registered with this flag, RPC rejects calls made by remote RPC clients. In addition, local calls over all **ncadg_*** protocol sequences and all **ncacn_*** protocol sequences (except for named pipes, using **ncacn_np**) are also rejected. If a call is made on **ncacn_np**, RPC only allows the call if it does not come from SVR, which filters out all remote calls. **Ncalrpc** calls are always allowed through.

Why is this change important?

This change provides RPC application developers with additional security tools to help secure their RPC interface.

What works differently?

These flags will not change any existing Windows XP application or cause it not to run correctly. The use of these new flags is at the discretion of the application developer.

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

Setting name	Location	Default value	Possible values
Restrict RemoteClients	**HKEY_LOCAL_MACHINE \ SOFTWARE\Policies\ Microsoft\Windows NT\RPC** -or- (Group Policy object) **Computer Configuration \Administrative Templates \System\Remote Procedure Call\Restrictions for Unauthenticated RPC Clients**	**1** - Default	**0** – None **1** – Default **2** – High
EnableAuthEp Resolution	**HKEY_LOCAL_MACHINE \SOFTWARE\Policies \Microsoft\Windows NT\RPC**	**0** - Disabled	**0** – Disabled **1** – Enabled

Setting name	Location	Default value	Possible values
	-or- (Group Policy Object) **Computer Configuration \Administrative Templates \System\Remote Procedure Call\Restrictions for Unauthenticated RPC Clients**		

Do I need to change my code to work with Windows XP Service Pack 2?

You may need to change your code to work with Windows XP Service Pack 2. For more information about application changes which might be required, see the previous sections on **RestrictRemoteClients** and **EnableAuthEpResolution**.

Top of page

WebDAV Redirector

What does WebDAV Redirector do?

The WebDAV Redirector (DAVRdr) allows computers running Windows XP to use WebDAV (Web-based Distributed Authoring and Versioning) servers, such as Windows SharePoint Services and MSN Communities, as if they were standard file servers. It consists of a kernel component that connects to a Windows NT remote file system stack, and a user-level component (Web client service) that translates file system requests into WebDAV requests.

Who does this feature apply to?

This feature is used by people who access WebDAV servers through the remote file system. WebDAV Redirector is implemented in the remote file system stack. Client administrators, and users who are concerned with the security of their computer credentials, need to be aware that every access to remote files on a WebDAV server by Universal Naming Convention (UNC) (for example, *ServerName**ShareName**File.txt*) will be processed by WebDAV Redirector.

What new functionality is added to this feature in Windows XP Service Pack 2?

Disabling Basic Authentication over a clear channel
Detailed description

WebDAV is an extension of Hypertext Transfer Protocol (HTTP), and as such includes the use of *Basic Authentication* (BasicAuth). BasicAuth is one form of user authentication, or means by which a user is securely identified to the server. With BasicAuth, the client transmits the user's credentials (user name and password) to the server. If the channel is unencrypted, such as with normal HTTP traffic, any computer on the network can see the user's user name and password and therefore steal their identity. The DAVRdr does not support encrypted HTTP (HTTPS or SSL), and will transmit the user's credentials in the clear (or, without encryption) if the server supports basic authentication. Although a server most likely would not be configured to use basic authentication, it would be possible to set up the server expressly to obtain users' credentials.

Because of this possibility, Windows XP Service Pack 2 (SP) for adds the ability to enable or disable the use of BasicAuth by the DAVRdr. By default, use of BasicAuth is disabled with SP2. When BasicAuth is disabled, the client will either use a different authentication method (if the server supports one) or fail the request.

Why is this change important?

Users can log on to WebDAV servers for remote file access without fear of transmitting their password in the clear.

What threats does it help mitigate?

Imagine a corporate user at Contoso Corporation who routinely accesses the file share \\Contoso_Server\Sales outside the corporation on a public network, and uses an application which attempts to access that share as part of normal background activity. Since the user's portable computer is outside the corporate network, the request should fail. However, the DAVRdr will transmit a request to see if there is a DAV server named Contoso_Server, even though the actual server that the portable computer is attempting to access is an SMB server.

An attacker can be operating on that same public network with a computer that spoofs WINS requests, returning a pointer to itself in response to any WINS request. The portable computer will then try to access a DAV share on that rogue server. If the rogue server responds with BasicAuth as the authentication method, a dialog box appears that asks for the user's credentials. The dialog box identifies the server as Contoso_Server, leading the user to believe the request is legitimate. If the user enters their user name and password, the client transmits that information in the clear and the attacker thus gains access to that user's login information. There is no indication to the user that the channel is not secure, that the request is being handled by the DAVRdr, or that the portable computer will transmit the user name and password in the clear. Note that the current default Windows authentication methods never transmit a user's password in the clear.

What works differently?

Since the change to default behavior only affects the DAVRdr, the only scenarios that fail to work are those that require basic authentication, and that use the DAVRdr. An example is using Notepad.exe to

access a Web site that only allows BasicAuth. This scenario will no longer work. Also, even if the server was configured to only use basic authentication, other applications such as Office will continue to work since they use a different DAV client.

How do I resolve these issues?

You can enable BasicAuth by adding the following registry key and setting it to a non-zero value:

HKEY_LOCAL_MACHINE\SYSTEM
\CurrentControlSet\Services \WebClient\Parameters\UseBasicAuth (DWORD)

If you delete the registry key or set it to 0, the behavior reverts to the default, or disabling the use of BasicAuth.

WININET: Disabling Basic Authentication over a clear channel
Detailed description

Because the DAVRdr is part of the remote file system stack, a computer is open to attack whenever an attempt is made to remotely access files. Although the threat to other applications that use the Internet APIs is less severe than it is for the DAVRdr, a similar attack is possible whenever an application (or the user) attempts to access a URL. For this reason, WinInet is exposing the mechanism by which the DAVRdr disables BasicAuth to other users of the Internet APIs.

With Windows XP Service Pack 2, there are two ways to block the use of Basic Authentication over clear (or unencrypted) channels:

- Create the following registry key and set it to a non-zero value.

  HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows \CurrentVersion\ Internet Settings\DisableBasicOverClearChannel (DWORD)

  This prevents WININET from attempting to use BasicAuth unless the channel is secured (HTTPS or SSL).

- The application can disable the use of BasicAuth for its connections by setting the AUTH_FLAG_DISABLE_BASIC_CLEARCHANNEL **flag (0x4) in the value supplied in the call to InternetSetOption using** INTERNET_OPTION_AUTH_FLAGS.

Why is this change important?

Users can log on to WebDAV servers for remote file access without fear of transmitting their password in the clear.

What threats does it help mitigate?

Imagine a corporate user who routinely accesses the Web site http://www.contoso.com/sales. While outside the corporation on a public network, the user attempts to access that site using Internet Explorer. Since the laptop is outside the corporation, the request should fail with a "Server not found" message. An attacker can run on that same public network with a computer that spoofs WINS requests, returning a pointer to itself in response to any WINS lookup. The laptop will then try to send the HTTP request to load the page from the rogue server. If the rogue server responds with BasicAuth as the authentication method, the laptop responds to the user, asking for his or her credentials. It identifies the site http://www.contoso.com/sales, leading the user to believe the request is legitimate. If the user enters his or her user name and password, the client will transmit that information in the clear, and the attacker thus gains access to that user's login information. In particular, there is no indication to the user that the channel is insecure, or that the laptop will transmit the user name and password in the clear.

What works differently?

By default, there is no change in behavior for WININET applications (except for the DAVRdr as noted above). If this setting is disabled, the user is unable to connect to HTTP servers that only support Basic Authentication.

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

Setting name	Location	Previous default value (if applicable)	Default value	Possible values
UseBasicAuth	**HKEY_LOCAL_MACHINE \System \CurrentControlSet \Services \WebClient \Parameters \UseBasicAuth**	Not applicable.	Key doesn't exist (BasicAuth disabled for DAVRdr)	**0**, non-zero
DisableBasicOver ClearChannel	**HKCU\SOFTWARE \Microsoft \Windows \CurrentVersion \Internet Settings \DisableBasicOver ClearChannel**	Not applicable.	Key doesn't exist (BasicAuth enabled for everything else)	**0**, non-zero

Do I need to change my code to work with Windows XP Service Pack 2?

No changes are necessary. Developers who write applications that use the Internet APIs and want to disable BasicAuth, such as the DAVRdr, can add a call to **InternetSetOptions()**.

Top of page

Windows Firewall

What does Windows Firewall do?

Windows Firewall (previously called Internet Connection Firewall or ICF) is a software-based, stateful filtering firewall for Microsoft Windows XP. Windows Firewall provides protection for computers that are connected to a network by preventing unsolicited incoming traffic through TCP/IP version 4 (IPv4) and TCP/IP version 6 (IPv6). Configuration options include:

- Configuring and enabling port-based exceptions
- Configuring and enabling program-based exceptions
- Configuring basic ICMP options
- Logging dropped packets and successful connections

Who does this feature apply to?

This feature applies to

- All computers that are connected to a network, including the Internet
- All programs (applications and services) that listen on the network
- All programs that do not work with stateful filtering

What new functionality is added to this feature in Windows XP Service Pack 2?

On-by-Default
Detailed description

Windows Firewall is turned on by default for all network interfaces. This provides more network protection by default for Windows XP on new installations and upgrades. On-by-Default also protects new network connections as they are added to the system. This applies to both IPv4 and IPv6 traffic, and is enabled even if there is another firewall already present on the system.

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

Prior to Windows XP Service Pack 2, Windows XP shipped with Internet Connection Firewall disabled by default. The user either needed to run a wizard or navigate through the Network Connections folder to manually enable Windows Firewall. This experience proved too difficult for many users, and resulted in many computers not having any firewall protection.

By enabling Windows Firewall by default, the computer has more protection from many network-based attacks. For example, if Windows Firewall had been enabled by default, the recent MSBlaster attack would have been greatly reduced in impact, regardless of whether users were up-to-date with patches.

What works differently?

After installing Windows XP Service Pack 2, Windows Firewall is enabled by default. This might create application incompatibility if the application does not work with stateful filtering by default.

How do I resolve these issues?

Modifying an application to work with Windows Firewall is described in detail later in this document, in "Do I need to change my code to work with Windows XP Service Pack 2?"

Boot time security
Detailed description

In earlier versions of Windows, there is a window of time between when the network stack comes up and when Internet Connection Firewall provides protection. This results in the ability for a packet to be received and delivered to a service without Internet Connection Firewall providing filtering and potentially exposes the computer to vulnerabilities. This was due to the firewall driver not starting to filter until the firewall user-mode service was loaded and had applied appropriate policy. The firewall service has a number of dependencies that causes the service to wait until those dependencies are cleared before it pushes the policy down to the driver. This time period is based upon the speed of the computer.

In Windows XP Service Pack 2, the IPv4 and IPv6 firewall drivers have a static rule to perform stateful filtering. This static rule is called a *boot-time policy*. This allows the computer to perform basic networking tasks such as DNS and DHCP and communicate with a domain controller to obtain policy. Once the Windows Firewall service is running, it loads and applies the run-time policy and removes the boot-time filters. The boot-time policy cannot be configured.

There is no boot-time security if the Windows Firewall service (which is listed as **Windows Firewall/Internet Connection Sharing (ICS)** in the Service Control Manager) is stopped and set to either Manual or Disabled.

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

With this change, the computer is open to fewer attacks during startup and shutdown.

What works differently?

If the Windows Firewall service fails to start, boot-time security remains in effect. This means that all incoming connections are blocked. In this case, an administrator will not be able to remotely troubleshoot the issue because all the ports will be closed, including the port used by Remote Desktop.

How do I resolve these issues?

To turn off boot-time security, stop the Windows Firewall/Internet Connection Sharing (ICS) service and set its startup type to either Manual or Disabled.

If the computer is in boot-time security mode because the firewall service has not started, an administrator must log onto the console, resolve the cause of the failure, and then manually start the firewall service.

Global configuration
Detailed description

In earlier versions of Windows, Windows Firewall was configured on a per-interface basis. This meant that each network connection had its own set of firewall settings, for example, one set of settings for wireless, another set of settings for Ethernet. This made it difficult to synchronize firewall settings between connections. Additionally, new connections would not have any of the configuration changes that had been applied to the existing connections. Non-standard network connections, such as those created by proprietary dialers (for instance, ISP configured dial-up networking connections) could not be protected.

With global configuration, whenever a configuration change occurs, it automatically applies to all network connections in the **Network Connections** folder, including any non-Microsoft dialers. When new connections are created, the configuration is applied to them as well. Configuration can still be performed on a per-interface basis. Non-standard network connections will only have global configuration. Configuration changes also apply to both IPv4 and IPv6.

Why is this change important?

Having global policy makes it easier for the user to manage their firewall policy across all network connections and enables configuration through Group Policy. It also allows you to enable applications to work on any interface with a single configuration option.

What works differently?

In earlier versions of Windows, firewall configuration was on a per-interface basis. In Windows XP Service Pack 2, the configuration is global and applies to both IPv4 and IPv6.

How do I resolve these issues?

If your application or service requires static openings to work, you should open the ports globally, as described later, in "Do I need to change my code to work with Windows XP Service Pack 2?"

Traffic scoping for exceptions
Detailed description

ICF allowed excepted traffic to come from any IPv4 address. With Windows Firewall in Windows XP Service Pack 2, you can also configure an exception to only allow incoming traffic from addresses that are directly reachable (the My network (subnet) only scope option) (based on entries in the IPv4 and IPv6 routing tables), or from specific IPv4 addresses and specific IPv4 address ranges (the Custom list scope option).

When the File and Printer Sharing built-in exception is enabled with the NetShare application programming interface (API), the Network Setup Wizard, or through the Windows Firewall user interface, incoming file and printer sharing connection requests by default can only come from directly reachable addresses.

For machines in a workgroup, some exceptions are restricted to locally reachable addresses by default. These exceptions are those needed for file and printer sharing and the Universal Plug and Play (UPnP™) framework. Additionally, when these exceptions are opened for locally reachable addresses on an Internet Connection Sharing host, the exceptions will not be opened on the ICS public interface. If you enable these exceptions for all possible addresses, they will be opened on the ICS public interface, which is not recommended.

It is recommended that you apply the locally reachable addresses restriction to any exception that is used for communicating on the local network. It can be done programmatically, through the Windows Firewall Netsh Helper, or the Windows Firewall user interface.

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

Some applications only need to talk to other hosts on the local network and not hosts on the Internet. Configuring Windows Firewall to only allow traffic from locally reachable addresses or from specific address ranges corresponding to locally attached subnets restricts the set of addresses from which unsolicited incoming can be accepted. This mitigates, but does not eliminate, attacks that can occur for enabled exceptions.

What works differently?

When the File and Printer Sharing built-in exception is enabled on a computer that is a member of a workgroup, four ports are specifically affected by the locally reachable addresses restriction. The following ports will only receive traffic from locally reachable addresses:

- UDP port 137
- UDP port 138
- TCP port 139
- TCP port 445

If an application or service also uses these ports, it will only be able to communicate with other nodes that are assigned locally reachable addresses.

When the UPnP framework is enabled, two ports are specifically affected the locally reachable addresses

restriction and only receive traffic from the locally reachable addresses:

- UDP port 1900
- TCP port 2869

How do I resolve these issues?

If your application or service does not work with this type of restriction, you should open the port for global connections, as described in "Do I need to change my code to work with Windows XP Service Pack 2?" later in this document.

Command-line support
Detailed description

The Windows Firewall Netsh Helper was added to Windows XP in the Advanced Networking Pack. This helper only applied to IPv6 Windows Firewall. With Windows XP Service Pack 2, the structure and syntax of the helper changes and expands to include support for configuring IPv4 as well. With the Netsh Helper, you can fully configure Windows Firewall, including:

- Configure the default state of Windows Firewall (Off, On, On with no exceptions)
- Configure and enable port-based exceptions
- Configure the logging options
- Configure the Internet Control Message Protocol (ICMP) handling options
- Configure and enable program-based exceptions

Why is this change important?

Providing a command-line interface provides administrators with a method to configure Windows Firewall without going through the graphic user interface. The command-line interface can be used in logon scripts and remote management.

What works differently?

Any script that was created with the Netsh Helper that was made available with the Advanced Networking Pack for Windows XP, no longer works and needs to be updated.

"On with no exceptions" operational mode
Detailed description

Windows Firewall can be configured for exceptions to allow specific unsolicited incoming traffic during normal use. Typically, this is because key scenarios, like file and printer sharing, must be enabled. If a security issue is discovered in one or more of the listening services or applications that are running on the computer, it may be necessary for the computer to switch into a client-only mode, which is called "On with no exceptions." Switching into this client-only mode configures Windows Firewall to prevent all unsolicited incoming traffic without having to reconfigure the firewall.

When in this mode, all exceptions are temporarily disabled and any existing connections are dropped. Any API call to Windows Firewall to create an exception is allowed and the requested firewall configuration is stored, but it is not enabled until the operational mode switches back to normal operation. All listen requests by applications are also ignored.

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

Viruses, worms, and attackers look for services to exploit. When in this operational mode, Windows Firewall helps to prevent these types of attacks from succeeding.

What works differently?

When in this operational mode, the computer cannot listen for requests that originate from the network. Any existing incoming connections are terminated. Outgoing connections are the only connections that succeed.

How do I resolve these issues?

When in this operational mode, it is expected that some functionality will fail because of the strict network security in place. You can restore functionality by returning the operational mode to On, which is its default state. This action should only be performed by the user after the threat has been identified and mitigated, because the security of the computer is reduced by performing this action.

Program-based exceptions
Detailed description

Some programs (applications or services) act as both network clients and servers. When they act as servers, they need to allow unsolicited incoming traffic to come in, because they do not know who the peer will be ahead of time.

In earlier versions of Windows, a program needed to call the firewall APIs to enable the necessary listening ports to be open. This proved difficult in peer-to-peer situations when the port was not known in advance. It was up to the program to close the port again once communication was completed. Without this, there would be unnecessary holes in the firewall if the program terminated unexpectedly.

Additionally, these ports could only be opened if programs were running in the security context of a local administrator. This violated the principle of least privilege by requiring programs to run in an administrative context, rather than only with the minimum necessary privileges.

In Windows XP Service Pack 2, a program that needs to listen to the network can be added to the Windows Firewall exceptions list. If a program is on the Windows Firewall exceptions list, Windows opens and closes the necessary listening ports automatically, regardless of the program's security context. For more information on adding programs to the Windows Firewall exceptions list, see "How do I resolve these issues?" later in this document.

Note Programs that work with stateful filtering do not need to be placed on the Windows Firewall exceptions list. Only administrators can add a program to the Windows Firewall exceptions list.

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

When a program is on the Windows Firewall exceptions list, only the necessary ports are opened, and they are only opened for the duration that the program is listening on those ports. A program cannot open a port that it is not using, which might deliberately or inadvertently expose another program or service to network traffic from that port.

This also allows programs that are listening to the network to run using an account with lesser privileges. In previous versions of Windows, the user had to run these programs with Administrator rights.

What works differently?

If a program needs to listen on the network, it must be on the Windows Firewall exceptions list. If it is not, then the necessary ports in Windows Firewall are not opened and the program will not be able to receive unsolicited incoming traffic.

How do I resolve these issues?

A program can be placed on the Windows Firewall exceptions list in five ways:

1. **Programmatically**. It is recommended that independent software vendors (ISVs) place their program on the Windows Firewall exceptions list during installation. For more information on how to programmatically add an exception, see "Do I need to change my code to work with Windows XP Service Pack 2?" later in this document.

2. **Command-line interface.** This method can be used by IT administrators who manage Windows XP systems using scripts or other command-line tools.

3. **Group Policy settings**. This method can be used by IT administrators to add the program to the exceptions list through Group Policy.

4. **Windows Security Alert notification message**. A user with Administrator rights can interact with the Windows Security Alert notification message and add the application to the exceptions list.

   When a application performs a TCP listen or UDP bind to a non-wildcard port, the network stack passes the application name and port to Windows Firewall. Windows Firewall looks up the application name on the exceptions list. If the application is on the exceptions list and enabled, then the corresponding port is opened in the firewall. If the application is on the exceptions list and disabled, then the corresponding ports are not opened. If the application is not on the exceptions list, then users are asked to make a choice. If the users have administrative rights, they can:

   - Unblock the application to allow it to listen on the network. It is added to the exceptions list as Enabled and the ports are opened.

   - Block the application from listening on the network. It is added to the exceptions list as Disabled and the ports are not opened. Unsolicited incoming traffic for the application is blocked unless the local administrator specifically enables the exception on the **Exceptions** tab. By adding the application to the exceptions list, Windows Firewall does not prompt the user every time the application is run.

   - Choose to be asked again later. The application is not added to the exceptions list and the ports are not opened.

   If the user is does not have administrative rights, they are notified that the application is not allowed to listen on the network and that an Administrator must enable the application. At this point, the application is listed in the exceptions list as Disabled.

5. **Manual configuration**. The user can decide to enable a program manually by selecting it from a list that is populated from the list of applications in the Start menu, or by browsing for the program.

Multiple Profiles
Detailed description

Multiple profile support in Windows Firewall allows you to create two sets of firewall policy: one for when the computer is connected to a managed network and one for when the computer is not. You can specify settings that are less strict when the computer is connected to the corporate network to enable line of business applications to work. You can also have a more aggressive security policy that will be enforced when the computer leaves the managed network, which helps to protect mobile users.

Note Multiple profiles for Windows Firewall only applies to computers that are joined to a domain. Computers that are in a workgroup only have one profile.

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

For a mobile computer, it is desirable to have more than one firewall configuration. Often, a configuration that is safe on a corporate network is likely to be susceptible to attack on the Internet. Therefore, being able to have ports opened on the corporate network and not on other networks is critical to ensuring that

only the necessary ports are exposed at any given time.

What works differently?

If an application needs to be listed in the Windows Firewall exceptions list in order to work correctly, it might not work on both networks, as the two profiles might not have the same set of policy. For an application to work on all networks, it must be listed in both profiles. (For more information about the Windows Firewall exceptions list, see the earlier section.

How do I resolve these issues?

If the computer is joined to a domain, you must ensure that the application is listed in both firewall configurations.

RPC support for System Services
Detailed description

In earlier versions of Windows, Internet Connection Firewall blocked remote procedure call (RPC) communication. While Internet Connection Firewall could be configured to allow network traffic to the RPC Endpoint Mapper, the port that RPC used was unknown and the application would still fail.

Many enterprise applications and components fail if RPC is not allowed to communicate over the network. Some examples include, but are not limited to, the following:

- Remote administration, such as the Computer Management feature and the Select User, Computers, and Groups dialog box which is used by many applications
- Remote Windows Management Instrumentation (WMI) configuration
- Scripts that manage remote clients and servers

RPC opens several ports and then exposes many different servers on those ports. Since so many RPC servers are included with Windows XP, Windows Firewall takes a different approach for system services using RPC. Windows Firewall will only accept this claim if the caller is running in the Local System, Network Service, or Local Service security contexts.

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

In order to enable scenarios around remote administration, many enterprise-wide deployments require that the system services that use RPC work with Windows Firewall by default. By using more precision, you can control which RPC services are exposed to the network.

What works differently?

By default, RPC does not function through Windows Firewall. All system services that use RPC are affected. However, Windows Firewall can be configured to allow RPC to work for these services.

How do I resolve these issues?

See “Do I need to change my code to work with Windows XP Service Pack 2?” later in this document.

Restore Defaults
Detailed description

Previously, there was no way for a user to reset the configuration of Windows Firewall. Over time, Windows Firewall might be configured to allow unsolicited incoming traffic, either through adding applications or ports to the Windows Firewall exception list. This may make it difficult for the user to easily and quickly go back to a default configuration.

This option enables the user to restore Windows Firewall settings to their original defaults. In addition, the Windows Firewall defaults can be modified by OEMs and businesses to provide custom default configuration options.

Why is this change important?

This option allows end-users to restore their Windows Firewall settings to the out-of-the-box defaults.

What works differently?

No functional changes in Windows Firewall result from this addition. However, use of this feature disables Internet Connection Sharing and Network Bridge.

Unattended Setup support
Detailed description

In earlier versions of Windows, it was not possible to configure Internet Connection Firewall during installation. This made it difficult for OEMs and businesses to preconfigure Internet Connection Firewall before distributing the computer to their end users. In Windows XP Service Pack 2, you can configure the following options of Windows Firewall through unattended setup:

- Operational mode
- Applications on the Windows Firewall exception list
- Static ports on the exception list
- ICMP options
- Logging options

Why is this change important?

A method to preconfigure Windows Firewall allows Windows resellers and large enterprises more flexibility and customization options for Windows Firewall.

What works differently?

This feature adds configuration flexibility to Windows Firewall. No functional changes in Windows Firewall result from this addition.

What existing functionality is changing in Windows XP Service Pack 2?

Enhanced multicast and broadcast support
Detailed description

Multicast and broadcast network traffic differs from unicast traffic because the response comes from an unknown host. As such, stateful filtering prevents the response from being accepted. This stops a number of scenarios from working, ranging from streaming media to discovery.

To enable these scenarios, Windows Firewall will allow a unicast response for 3 seconds from any source address on the same port from which the multicast or broadcast traffic originated.

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

This allows applications and services which use multicast and broadcast for communicating to work without either the user or application/service to alter the firewall policy. This is important for things like NETBIOS over TCP/IP, so that sensitive ports such as port 135 are not exposed.

What works differently? Are there any dependencies?

In previous versions of Windows, Internet Connection Firewall did not perform any multicast or broadcast filtering. In Windows XP Service Pack 1, Internet Connection Firewall statefully filtered multicast and broadcast traffic, which required the user to manually open the port to receive the response. In Service Pack 2, Windows Firewall accepts the response to the multicast or broadcast traffic without additional configuration.

Integration of Internet Connection Firewall and IPv6 Windows Firewall
Detailed description

The version of Internet Connection Firewall that was introduced with Windows XP only filtered IPv4 traffic. IPv6 Internet Connection Firewall was introduced with the Advanced Networking Pack for Windows XP. At the time these two firewalls were separate, and each used its own configuration options. With Windows XP Service Pack 2, Internet Connection Firewall and IPv6 Internet Connection Firewall are integrated into a single component called Windows Firewall.

With this change, any configuration change applies to both IPv4 and IPv6 traffic. For example, when a static port is opened, it is opened for both IPv4 and IPv6 traffic.

Why is this change important?

This allows for easier configuration management and application compatibility.

What works differently?

The separate IPv6 firewall service is removed from the system and replaced with the Windows Firewall service, which filters both IPv4 and IPv6 traffic. All APIs that were introduced with the Advanced Networking Pack for Windows XP are superseded by new APIs introduced with Windows XP Service Pack 2.

How do I resolve these issues?

For more information, see “Do I need to change my code to work with Windows XP Service Pack 2?” later in this document.

Updated Netsh Helper
Detailed description

The firewall context of Netsh Helper was first introduced with the Advanced Networking Pack for Windows XP. This only applied to IPv6 Windows Firewall. With the integration of Windows Firewall and IPv6 Windows Firewall, the firewall context of Netsh Helper no longer has an IPv6 context.

Why is this change important?

This change accommodates the changes to Windows Firewall and integration of IPv4 filtering configuration options in the existing firewall context of Netsh Helper.

What works differently?

Any existing scripts that use the firewall context that appears with the addition of the Advanced Networking Pack will no longer work.

How do I resolve these issues?

Update any scripts you might have so that they include the new firewall context and syntax.

Updated user interface
Detailed description

The Windows Firewall user interface is updated in Windows XP Service Pack 2 to accommodate the new configuration options and the integration of IPv6 Internet Connection Firewall. It provides the user with the ability to change the operational states, the global configuration, logging options, and ICMP options.

The primary entry to the user interface has been moved from the Properties dialog box of the connection to a Control Panel icon. A link from the old location is still provided. Additionally, Windows XP Service Pack 2 creates a link from the Network Connections folder.

Why is this change important?

The functionality that is added in Windows XP Service Pack 2 required updates to the user interface.

What works differently?

The user interface is moved from the **Advanced** tab of the network connection's **Properties** dialog box to a specific Windows Firewall icon in Control Panel.

New Group Policy support
Detailed description

In earlier versions of Windows, Internet Connection Firewall had a single Group Policy object (GPO): **Prohibit Use of Internet Connection Firewall on your DNS domain network**. With Windows XP Service Pack 2, every configuration option can be set through Group Policy. Examples of the new configuration options available include:

- Define program exceptions
- Allow local program exceptions
- Allow ICMP exceptions
- Prohibit notifications
- Allow file and printer sharing exception
- Allow logging

Each of these objects can be set for both the corporate and standard profile. For a complete list of Group Policy options, see "Deploying Internet Connection Firewall Settings for Microsoft Windows XP with Service Pack 2" in the Microsoft Download Center at http://go.microsoft.com/fwlink/?linkid=23277.

Why is this change important?

It is important for administrators to manage Windows Firewall policy to enable applications and scenarios to work in the corporate environment.

What works differently?

The IT administrator can now decide the default Windows Firewall policy set. This can either enable or disable applications and scenarios. This allows more control, but the policies do not change the underlying functionality of Windows Firewall.

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

Setting name	Location	Previous default value	Default value	Possible values
Protect all network connections	(Group Policy object) **Computer Configuration \Administrative Templates \Network\Network Connections\ \Windows Firewall**	Not applicable.	Not configured	**Enabled** **Disabled**
Do not allow Exceptions	(Group Policy object) **Computer Configuration \Administrative Templates \Network\Network Connections \Windows Firewall**	Not applicable.	Not configured	**Enabled** **Disabled**
Define program exceptions	(Group Policy object) **Computer Configuration \Administrative Templates \Network\Network Connections \Windows Firewall**	Not applicable.	Not configured	**Enabled** **Disabled** **Program path** **Scope**
Allow local program exceptions	(Group Policy object) **Computer Configuration \Administrative Templates \Network\Network Connections \Windows Firewall**	Not applicable.	Not configured	**Enabled** **Disabled**

Setting name	Location	Previous default value	Default value	Possible values
Allow remote administration exception	(Group Policy object) **Computer Configuration \Administrative Templates \Network\Network Connections \Windows Firewall**	Not applicable.	Not configured	**Enabled** **Disabled**
Allow file and printer sharing exception	(Group Policy object) **Computer Configuration \Administrative Templates \Network\Network Connections \Windows Firewall**	Not applicable	Not configured	**Enabled** **Disabled**
Allow ICMP Settings	(Group Policy object) **Computer Configuration \Administrative Templates \Network\Network Connections \Windows Firewall**	Not applicable.	Not Configured	Echo Request: **On, Off** **Source Quench: On, Off** **Redirect: On, Off** **Destination Unreachable: On, Off** **Router Request: On, Off** **Time Exceeded: On, Off** **Parameter Problem: On, Off** **Mask Request: On, Off** **Timestamp Request: On, Off**
Allow remote desktop exception	**(Group Policy object) Computer Configuration \Administrative Templates \Network\Network Connections \Windows Firewall**	Not applicable	Not configured	**Enabled** **Disabled**
Allow UPnP framework exception	**(Group Policy object) Computer Configuration \Administrative Templates \Network\Network Connections \Windows Firewall**	Not applicable	Not configured	**Enabled** **Disabled**
Prohibit notifications	**(Group Policy object) Computer Configuration \Administrative Templates\Network \Network Connections \Windows Firewall**	Not applicable	Not configured	**Enabled** **Disabled**
Allow logging	**(Group Policy object) Computer Configuration \Administrative Templates \Network\Network Connections \Windows Firewall**	Not applicable	Not configured	**Enabled** **Disabled**

Setting name	Location	Previous default value	Default value	Possible values
Prohibit unicast response to multicast or broadcast requests	**(Group Policy object) Computer Configuration \Administrative Templates \Network\Network Connections \Windows Firewall**	Not applicable	Not configured	**Enabled** **Disabled**

Do I need to change my code to work with Windows XP Service Pack 2?

Outbound connections
Description

For typical consumer and office computers, the computer is a client on the network. Software on the computer connects out to a server (an outbound connection) and gets responses back from the server. Windows Firewall allows all outbound connections, but applies rules to the types of communication that are allowed back into the computer. For more information about what network traffic Windows Firewall allows as part of Transmission Control Protocol (TCP) and User Data Protocol (UDP) outbound connections, see Notes, below.

Action Required

None. Windows Firewall will automatically allow all outbound connections, regardless of the program and the user context.

Notes

- When a computer initiates a TCP session request to a target computer, it will only accept a response from that target computer.
- When the computer sends UDP packets, Windows Firewall allows UDP responses to the port from which the UDP packets were sent from any IP address for 90 seconds.
- Unicast responses to multicast and broadcast traffic are allowed through Windows Firewall for 3 seconds if the responses are to the port from which the multicast traffic was sent and are from IP addresses on the same subnet as the computer. A setting in the firewall controls this behavior, which is enabled by default.

Examples

- Surfing the Web using Microsoft Internet Explorer
- Checking e-mail in Outlook Express
- Chatting in MSN Messenger or Windows Messenger

Unsolicited inbound connections for applications
Description

This scenario covers an application that completes a listen operation on a TCP socket or successfully binds to a specific UDP socket through Winsock. For this scenario, Windows Firewall can automatically open and close ports as needed by the application.

Action Required

When an application that needs to listen on a port or ports is being installed by an administrator, the users must indicate whether if they want to allow the application to open ports in the firewall.

- If the user consents to this, then the application should use the *INetFwAuthorizedApplication* API to add itself to the AuthorizedApplications collection as Enabled.
- If the user does not consent, then the application should use the *INetFwAuthorizedApplication* API to add itself to the AuthorizedApplications collection as Disabled.

When using the *INetFwAuthorizedApplication* API to add an application to the AuthorizedApplications collection, the following values are required:

- **Image File Name**. This is the file that calls Winsock to listen for network traffic. This must be a fully-qualified path, but it might contain environment variables.
- **Friendly Name**. This is the description for the application that will be shown to users in the Windows Firewall user interface.

For more information on the *INetFwAuthorizedApplication* API, see "INetFwAuthorizedApplication" in the Microsoft Platform Software Development Kit (SDK) at http://go.microsoft.com/fwlink/?LinkId=32000.

Windows Firewall monitors Winsock to see when applications start and stop listening on ports. As a result, ports are automatically opened and closed for applications once their entries have been enabled in the Windows Firewall exceptions list. This means that no action is required by Winsock applications to actually open and close ports.

Notes

- An application must be running in the context of a user with Administrator rights to add itself to the

Windows Firewall exceptions list.

- Ports are automatically opened and closed for allowed Winsock applications, regardless of the user context in which the applications are running.
- Applications should get user consent before adding themselves to the AuthorizedApplications collection.
- Svchost.exe cannot be added to the AuthorizedApplications collection.

Examples

Some examples of tasks involving Microsoft applications that might work differently include:

- Using audio and video in MSN Messenger or Windows Messenger
- Transferring files in MSN Messenger or Windows Messenger
- Hosting a multiplayer game

Inbound connections for services
Description

While developers are advised to use the *AuthorizedApplication* APIs for all other scenarios, the use of global port APIs in Windows Firewall is recommended for services that listen on fixed ports. Since these ports are always open, there is minimal benefit to dynamically opening the ports. Instead, users gain the ability to customize the firewall settings for these fixed ports when the global port APIs are used.

Action Required

When a service needs to listen on a fixed port, it must ask the user whether it should allow the service to open ports in the firewall. Ideally this should be done at installation time of the service.

If the user consents to this, then the service should use the *INetFwOpenPort* API to add rules to Windows Firewall to open the fixed port or ports needed by the service. These rules should be enabled.

If the user does not consent, then the service should still use the *INetFwOpenPort* API to add rules to Windows Firewall to open the fixed port or ports needed by the service. These rules, however, should not be enabled.

When using the *INetFwOpenPort* API to add a port opening to Windows Firewall, the following values are required:

- **Port**. This is the number of the port to be opened. It must be between 0 and 65,535, inclusive.
- **Friendly Name**. This is the description for the port opening that will be shown to users in the Windows Firewall user interface.

For more information on the *INetFwAuthorizedApplication* API, see "InetFwAuthorizedApplication" on the Platform Software Development Kit on the MSDN Web site at http://go.microsoft.com/fwlink/?linkid=32000.

When a service is disabled, it should again use the *INetFwOpenPort* API to close the static ports that it opened whenever possible. This can be easily done if it is the only service that uses the ports. If the service potentially shares the ports with other services, however, it should not close the ports unless it can verify that none of the other services are using the ports.

An application must be running in the context of a user with Administrator rights to statically open ports in Windows Firewall.

Notes

- When statically opening ports through the *INetFw* API, a service should limit itself to traffic from the local subnet whenever possible.
- Services must get user consent before statically opening ports in Windows Firewall. A service should never just automatically open ports without first warning the user.

Examples

Some examples of services which require inbound connections are:

- File and printer sharing
- UPnP architecture
- Remote Desktop

Inbound connections on RPC and DCOM ports for System Services
Description

Some system services require the use of RPC ports, either through DCOM or RPC directly, for inbound connections. Because of the significant security implications when opening RPC ports, these ports are handled as a special case, and developers should only try to enable RPC for system services through Windows Firewall when absolutely necessary.

Action Required

Windows Firewall can be configured to enable the automatic opening and closing RPC and DCOM ports for system services. By default, however, RPC will be blocked by Windows Firewall. This means that

applications that use the RPC ports to transfer data to system services will need to configure Windows Firewall appropriately. When an application needs to enable this feature, it must ask the user whether it should allow the services to open ports in the firewall. Ideally, this should be done at installation time.

If the user consents to allowing the RPC ports to be opened, then the service should use the *INetFwRemoteAdminSettings* API to open the ports that are needed by the service.

If the user does not consent to allowing the RPC ports to be opened, then the application or service should not configure Windows Firewall to allow the RPC ports.

For more information on the *INetFwRemoteAdminSettings* API, see "INetFwAuthorizedApplication" on the MSDN Web site at http://go.microsoft.com/fwlink/?linkid=32000 and, in the table of contents, click "RemoteAddresses Property of InetFwAuthorizedApplication."

Notes

- To enable or disable the automatic opening of RPC ports in Windows Firewall, an application or service must be running in the context of a user with Administrator rights.
- Before allowing RPC ports through Windows Firewall, an application or service should get user consent.
- An application or service should try to allow the RPC ports through Windows Firewall only when absolutely necessary.
- If the RPC ports are already allowed, then the application or service does not need to do anything in order to function correctly.
- The RPC ports setting only works for RPC servers which run in the context of Local System, Network Service, or Local Service. Ports opened by RPC servers running in other user contexts will not be enabled through this setting. Instead, those RPC servers should use the Windows Firewall exceptions list.

Top of page

Windows Media Player

What does Windows Media Player do?

Windows Media Player is an application that provides media content, plays media files, and helps organize your stored media files.

Who does this feature apply to?

All users who use Windows Media Player should be aware of the changes incorporated in Windows XP Service Pack 2.

What new functionality is added to this feature in Windows XP Service Pack 2?

Windows Media Player 9 Series
Detailed description

Windows Media Player 9 Series is installed as part of Windows XP Service Pack 2. This version of Windows Media Player includes security fixes and new functionality.

During Windows XP Service Pack 2 installation, if you select the option to archive files, you can remove Windows Media Player 9 Series later. To do so, remove the service pack through Add or Remove Programs. Windows Media Player 9 Series is removed along with the service pack, and both the previous version of Windows Media Player and the operating system are restored to their previous version.

If you perform a new installation of Windows XP that includes Service Pack 2 on a computer that is running a previous version of Windows, the operating system is replaced, and Windows Media Player 9 Series cannot be removed. For more information, see the Windows Media 9 Series Knowledge Center at http://go.microsoft.com/fwlink/?LinkId=32062.

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

Earlier versions of Windows Media Player contained security vulnerabilities. Although these vulnerabilities have been fixed with software updates, a more thorough solution is to upgrade earlier versions to Windows Media Player 9 Series. Windows Media Player 9 Series has also been thoroughly tested and updated to work with the other security enhancements contained in Windows XP Service Pack 2.

What works differently?

If you uninstall Windows XP Service Pack 2, you might need to reacquire some licenses in order to play the content that you have previously licensed. This applies if you have upgraded your computer from Windows 2000 to or Windows XP to Windows XP Service Pack 2, because Windows Media Player 9 Series handles digital content licenses differently than earlier versions.

How do I resolve these issues?

To ensure that your existing licensed content remains available to Windows Media Player after you remove Windows XP Service Pack 2, do one of the following:

- Before you upgrade a computer to Windows XP Service Pack 2, back up the licenses for your digital media files. (You can do this through License Management in Windows Media Player.) Then, before removing the service pack, back up any additional licenses you have acquired. After you remove the service pack, restore all of the licenses.

- After you remove Windows XP Service Pack 2, you can install Windows Media Player 9 Series from the Microsoft Windows Media Download Center.

Top of page

Windows Messenger

What does Windows Messenger do?

Windows Messenger is an instant messaging program that lets you communicate in real time with other people who use Windows Messenger or MSN Messenger.

With Windows Messenger you can:

- Create a contact list of your friends, family, and coworkers who also use Messenger.
- See when your contacts are signed into Windows Messenger and available.
- Send text messages back and forth with your contacts.
- Call a phone almost anywhere in the world for a very low rate—and use your microphone or headset to talk.
- Call a contact's computer for free and see each other while you talk.
- Send pictures, music, or documents to your contacts.
- Link directly to the e-mail inbox of your default e-mail program.
- Invite someone to play a game, look at a program on your computer, or use a whiteboard together.
- Use Remote Assistance to allow someone to help you with your computer.
- Receive up-to-the-minute information using Microsoft .NET Alerts.

Who does this feature apply to?

All Windows Messenger users should be aware of the changes made to this feature in Windows XP Service Pack 2.

What new functionality is added to this feature in Windows XP Service Pack 2?

The following features have been added to Windows Messenger in Windows XP Service Pack 2:

- Block unsafe file transfers
- Require user display name
- Windows Messenger and Windows Firewall

Block Unsafe File Transfers
Detailed description

File transfers are blocked when they contain content that could be used to damage your computer. When someone tries to send a file to you, Windows Messenger first checks to see if the sender is on your list of contacts. Next, the file is put through a security check.

Files are blocked when both of the following occur:

- The sender is not on your Contacts list.
- Someone tries to send you a file that is considered unsafe.

Files with the extensions .jpg, .txt, and .gif are generally considered safe and you can receive them from someone not on your Contacts list.

Certain file types should be opened with caution, even if you know the sender, because they may include executable code. When you receive a certain type of file from someone on your Contacts list, you are asked what you want to do with the file. You should only accept the file if you can save it to your computer's hard disk and then scan it with an antivirus program before you open it.

You are prompted before opening the following file types:

- Microsoft Office files, such as .doc, .ppt, .xls.
- Files from other applications, such as .zip, .wpd, and .pdf.
- Computer applications, programs, or any file that contains software code or script including macros, executables, and JavaScript.
- Files with these extensions: .exe, .cmd, .wsh, .bat, .vb, .vbs; .pif, .scr, .scf.
- Other file types.

For a list of files that are generally considered unsafe, see "Information About the Unsafe File List in Internet Explorer 6" on the Microsoft Web site at http://go.microsoft.com/fwlink/?LinkId=25999.

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

File transfers are blocked when they contain content that could be used to damage your computer.

What works differently?

Some file transfers will be blocked.

How do I resolve these issues?

Files are only blocked when both of the following occur:

- The sender is not on your contact list.
- The file someone tries to send you is considered unsafe.

If you know the file is safe, you can add the user to your contact list and all file transfers you receive from them will succeed.

Required User Display Name
Detailed description

Windows Messenger now requires a user display name that is different from the user's e-mail address.

This requirement is enforced when the user signs in to Windows Messenger and when they update their display name in the **Options** dialog box.

When users sign in to Windows Messenger, they are prompted to provide a user display name prior to logon. If a display name is not provided, they cannot log on.

When users update their display name in the **Options** dialog box, Windows Messenger verifies whether the display names match their e-mail address.

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

When instant messaging conversations are saved for future reference using **Save As** on the **File** menu, the user's display name is stored in a text file, along with any associated text messages. Virus applications can discover the e-mail address in the text file and use it to propagate to the user's contacts through these same instant messaging conversations.

What works differently?

Users cannot sign in until they provide a user display name.

How do I resolve these issues?

The user must provide a user display name when prompted during sign-in.

Windows Messenger and Windows Firewall
Detailed description

Windows Messenger needs permission to connect to the Internet through the Windows Firewall.

To give Windows Messenger permission to connect to the Internet, click **Start**, click **Control Panel**, click **Security Options**, click **Windows Firewall**, click **Exceptions**, and then, in **Programs and Services**, click **Windows Messenger**.

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

The Windows Firewall has been turned on by default to protect the user's computer.

What works differently?

Windows Messenger will not run.

How do I resolve these issues?

Add Windows Messenger to the Windows Firewall Exception list.

Top of page

Wireless Provisioning Services

What does Wireless Provisioning Services do?

An increasing number of users are accessing the Internet through a growing number of public wireless networks, or wireless fidelity (Wi-Fi®) hotspots. Using Wireless Provisioning Services (WPS) in Service Pack 2 for Windows XP (SP2) with Service Pack 1 for Windows Server 2003 provides wireless users with a consistent experience and seamless connectivity to public Wi-Fi hotspots through automatic provisioning of the client and seamless roaming. WPS enables Wireless Internet Service Providers (WISPs) to use a standards-based and integrated platform to provide Wi-Fi hotspots with enhanced security that are easy to use and manage. In addition, WPS enables enterprises to easily provide guest access with enhanced security to private wireless networks.

With WPS, WISPs and enterprises can send provisioning and configuration information to mobile clients as they connect to the Internet or the corporate network. This in turn allows seamless, automatic and secure configuration of mobile clients, enabling a uniform sign-up experience in the enterprise and across different public network providers and hotspot locations.

Note This feature requires Service Pack 1 for Windows Server 2003 to enable the user scenarios described in this section. Windows Server 2003 Service Pack 1 has not yet been released.

Who does this feature apply to?

Wireless Provisioning Service is designed for three types of organizations:

- Hotspot Service Providers (HSP)

  HSPs deploy wireless access points in public places, such as shopping malls and airports, but HSPs are not Internet Service Providers (ISPs). Instead, the HSP contracts with one or more ISPs, and offers users one or more service plans to choose from when they want to establish an account for Internet access.

- Wireless Internet Service Provider (WISP)

  WISPS are ISPs that either deploy Wi-Fi hotspots in public places or outsource Wi-Fi hotspot services to an HSP.

- Enterprises

  Enterprises can use WPS technology to provide managed guest access on their networks.

What new functionality is added to this feature in Windows XP Service Pack 2?

Wireless Provisioning Services
Detailed description

Wireless Provisioning Services is an extension to the existing wireless services and user interfaces within Windows XP and Windows Server 2003. It builds on the wireless features already in Windows, such as the Wireless AutoConfiguration, and the wireless security features such as Protected Extensible Authentication (PEAP) and Wi-Fi Protected Access (WPA). WPS also includes modifications to Windows Server 2003. The Windows 2003 IAS (Internet Authentication Service) component was modified to include guest authentication of the clients in the provisioning process.

WPS includes a provisioning service component which allows for Wireless Internet Service Providers (WISPs) and enterprises to send provisioning and configuration information to a mobile client that is trying to connect to the Internet or the corporate network. By using Wireless Provisioning Services, WISPs can offer services at multiple network locations and use multiple network names (service set identifiers, or SSIDs). Once users have signed up to a WISP in one location or are pre-provisioned and have downloaded the provisioning information, they can automatically connect to the Internet on subsequent occasions using the network provided by the WISP in their different hotspot locations. The wireless auto configuration service will automatically choose the correct network belonging to the WISP based on the provisioning files supplied. WSP also enables automatic and seamless roaming between different providers.

Further, when WPS is used the client computer automatically keeps the provisioning information stored on the client computer up to date. This allows the provider to change the network settings, add new locations, and so on, without disrupting the service or having users reconfigure their systems.

When a user connects his computer to a WISP and establishes an account for the first time, the following four stages occur:

1. The computer discovers the WISP network at a Wi-Fi hotspot.
2. The user is authenticated using a guest account and the computer is connected to the Wi-Fi network.
3. The mobile client is provisioned and the user establishes an account with the WISP.
4. The user is authenticated on the Wi-Fi network using the new user account credentials.

Each of these stages is discussed in detail in the following scenario :

A user arrives at a Wi-Fi hotspot with a portable computer running Windows XP SP2 and Wireless Provisioning Services, when the computer comes within range of the WISP access point beacon the following occurs:

1. The Wireless Auto Configuration (WAC) service on the client computer detects the beacon information from the access point, which is enabled with broadcast Service Set Identifier (SSID). The SSID is equivalent to the network name.
2. The user is informed by Windows XP that a wireless network is available. The user views information in Windows XP, including the network's friendly name. In this example, the user possesses a promotion code to use for account establishment, and proceeds by clicking Connect. This causes the WPS client to connect the user's computer to the wireless network using a guest account with limited privileges.

When the guest account is authenticated by the Wi-Fi network, the following occurs:

1. WAC uses 802.1x and Protected Extensible Authentication Protocol (PEAP) to connect and authenticate as guest to the WISP network through the access point, automatically passing a blank user name and password to the WISP Internet Authentication Service (IAS) server (IAS is also known as the Microsoft RADIUS server). The access point is connected to a gateway device that allows traffic from the client to pass to the provisioning services in the network to complete the sign up process, but blocks the client from accessing the Internet.
2. The IAS server (or RADIUS server) is the PEAP authenticator and Transport Layer Security (TLS) endpoint for users who connect as guest. The TLS tunnel is created between the client and the IAS server. All subsequent messages between client and server pass through this tunnel, which traverses the access point and the gateway device.

3. Server authentication is performed when the IAS server verifies its identity to the client computer using a certificate that contains the Server Authentication purpose in Enhanced Key Usage (EKU) extensions. This certificate is issued by a public trusted root Certification Authority (CA) that the client computer trusts.

4. The IAS server authenticates and authorizes the user as Guest. In the Access-Accept message that the IAS server sends to the client is a container with a URL to the provisioning information. This URL provides the Wireless Provisioning Services engine running on the client, with the location of the XML master file.

When the client is provisioned and the user creates an account, the following occurs:

1. On the client computer, the Wireless Provisioning Services downloads the XML master file and sub-files from the provisioning server. The master file contains pointers to XML sub files that control the client's progress through the process. When the XML sign-up schema is downloaded, the sign-up wizard is launched on the client to allow the user to create and pay for an account with the WISP.

2. Using the sign-up wizard on the client computer, the user steps through the process of signing up for an account. The user enters the promotion code as well as personal data such as name, address, and credit card number. The data entered by the user is converted by the Wireless Provisioning Services client into an XML document.

3. The XML document containing the user's sign-up data is sent to the Web application on the WISP provisioning server.

4. The Web application checks the promotion code entered by the user against the promotion code database (e.g. SQL Server database). If the promotion code is valid, the Web application continues processing the user's data.

5. The Web application processes the user's payment information. Once payment is verified and sign-up information is completed successfully, the Web application reads the domain and security group information from the promotion code database and creates a user account in identity services (such as Active Directory) and adds the account to the security group. The Web application also enters the new user name in the promotion code database.

6. An XML document containing the new account credentials is sent from the WISP provisioning server to the Wireless Provisioning Services client on the client computer. The client computer uses the credentials to configure WAC and 802.1x under the name of the WISP. The connection is re-initiated with the new user account password-based credentials (user name and password).

Finally, when the user is authenticated using the new account credentials and gains Internet access, the following happens:

1. The Wireless Auto Configuration (WAC) service on the client computer restarts the association to the SSID for the WISP.

2. WAC finds the correct 802.11 profile which was downloaded with the other WISP information in the XML master file. WAC re-associates with the access point using the correct profile.

3. WAC uses 802.1x to start the authentication process using a combination of the Protected Extensible Authentication Protocol and the Microsoft Challenge Handshake Authentication Protocol version 2 (PEAP-MSCHAPv2) using the new account credentials passed to 802.1x by the Wireless Provisioning Services client.

4. As the client starts the authentication process with PEAP-MSCHAPv2 authentication, a TLS channel is created between the user's client computer and the WISP IAS server.

5. In the second stage of PEAP-MSCHAPv2 authentication, the WISP IAS server authenticates and authorizes the connection request against the new account in the user accounts database (e.g. Active Directory). The IAS server sends an Access-Accept message to the access point. Included in the Access-Accept message are attributes that specify the user can now get access to the Internet.

6. The access point instructs the gateway device to assign the client to the logical segment network with access to the Internet.

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

Wireless Provisioning Services makes it easier to use wireless hotspots without compromising security. WPS, with Windows Server 2003 Service Pack 1, and Microsoft IAS (also known as a RADIUS server), allows users' computers to more easily discover, connect and roam between wireless hotspots with enhanced security.

- The current connection model for WISP signup and use is not secured. Most Wi-Fi hotspots are configured for open authentication and without data encryption. Users are generally required to launch a Web browser to initially sign up to the WISP service and for subsequent logins. WSP mitigates this threat by adding encryption and authentication to the communications between the client and the wireless network.

- Browser redirection-based deployment has many usability issues. Users may not even know they have to launch their browser to get connected. Another example of what can happen is when the browser is set to use proxy settings to access the Internet and the user is connected directly to the corporate network. In this case, browser redirection does not work and the user would have to know to disable the proxy settings to connect to the hotspot. This can cause costly support calls to the WISP, or the

enterprise helpdesk.

- Browser based deployment is vulnerable to man-in-the-middle attacks, for example, by a malicious front-end server using a rogue access point. Users queried by this access point might unknowingly be giving away personal identification and credit card information. By eliminating the need for a Web login WSP reduces the vulnerability of WISP users to this type of attack.
- Without additional hotspot client software users can not easily detect hotspots and do not have a unified mechanism to sign up to them. It is not easy for users to find out information about the WISP or search for the hotspot locations for that WISP. If users sign up at one hotspot, they are not necessarily configured to automatically use the other hotspots. In addition, there is no standard mechanism to keep their provisioning and configuration information up-to-date.
- Add-on hotspot client software can help the user access that specific WISP's network. However, add-on software can also conflict with the wireless services native to the operating system, or client software from other providers potentially causing interoperability problems, even instability of the system as they all attempt to control the wireless settings of entire system. Updates to the WISP configuration usually require updates to the client software. For these reasons, many corporate IT departments are reluctant to deploy 3rd party hotspot client software to their users.
- There is no standardized mechanism across WISPs to process user sign ups and update their configurations. As a result, the user experience is fragmented and automatic and seamless roaming across providers can be difficult.

Wireless Network Registration Wizard
Detailed description

The Wireless Network Registration Wizard provides the user interface to sign-up for a wireless hotspot and guides the user through the provisioning process. The wizard builds content from provisioning information (XML files) provided by the WISP. The provisioning information can be dynamically downloaded or pre-installed on the client system. Pre-installation can be provided by an OEM for new systems, by the IT department within an organization, or from a WISP Web site. The WISP owns and creates the provisioning information and drives the users' sign up and provisioning experience. The following example presents a simple Wireless Network Registration Wizard experience with pre-paid code. The XML schema and wizard are flexible and can enable more complex sign-up experiences.

First, the user can either right-click the wireless network icon in the notification area and then click **View Available Wireless Networks**, or the user can respond to the notification message in the notification area that indicates availability of a new wireless network in range. When **Choose a wireless network** appears, the user selects a new wireless network and places that network on the preferred networks list.

The user then selects a network name (an SSID) and clicks **Connect** to connect to the wireless network. With a WPS-based Wi-Fi hotspot, the client detects that there is more provisioning information in form of XML files that is available about the network and the provider. It then confirms with the user whether the provisioning information should be downloaded. With a non-WPS network, the experience would be the same as with Windows XP today: either the user is prompted for a security key when connecting to a secure network or the user is warned that the network they are trying to connect to is not secure, and they are asked if they still want to connect to it.

After the download is complete, the Wireless Network Registration Wizard automatically launches and guides the user through the sign in process. The first screen, displays a customized logo (or banner) and content from the provider.

The subsequent screens may include selecting a subscription plan, entering credit card information, personal information and so on. In this example there is just one plan and the user is asked to enter their pre-paid or promotional code to get access to the network. Next, Wi-Fi Hotspot Deployment displays information about the selected plan, such the terms of the service agreement and privacy statement.

On the last screen, the wizard asks the user for their connectivity preferences for this connection. These default preferences can be set by the provider but can be overridden by the user. For example, if the user selects a monthly subscription with unlimited data, they probably want to always connect to the network automatically whenever in range. If the user chooses a "pay-as-you-go" plan, they probably want to control when to connect and choose a manual connection option as their preference.

The second option determines whether the client keeps the provisioning information automatically up to date. For example, if the provider adds new network names, new locations, or changes the network or security settings, the client can automatically update the information without any user interaction required while connected to the network.

On subsequent visits to hotspots in the same or different locations by the provider, or their roaming partners, and in case automatic connection is selected, all the user has to do is to turn their mobile computer back on or resume operations from standby, and they are automatically connected. When connected, instead of showing a cryptic network name or SSID in the **Choose wireless network** dialog box (which opens from the **View Available Wireless Networks** notification window), a friendly name of the provider will be shown, along with a logo of the provider.

From this dialog box, users can also search for available hotspot locations or view the help and support information provided by the WISP. Both the help and hotspot location information is downloaded as part of the provisioning information. The location information can be searched and viewed online or offline.

What existing functionality is changing in Windows XP Service Pack 2?

The wireless user interface has changed – a new **View Available Wireless Networks** dialog box will replace the existing dialog box.

Do I need to change my code to work with Windows XP Service Pack 2?

Wireless Provisioning Service does not require any changes to existing applications. There are two new APIs with WPS. One of the new APIs provides for adding and queries through the XML data on the machine. This API can be used to pre-provision the client from the WISP Web site by the user (using a stand-alone application), by OEMs, or IT departments.

Top of page

Wireless Network Setup Wizard

What does Wireless Network Setup Wizard do?

The Wireless Network Setup Wizard helps users configure a wireless network with enhanced security and simplifies the process of configuring wireless network connections on computers and other devices. The wizard stores the wireless settings including the configuration and security key onto removable media such as a USB Flash Drive that can be transported to other devices and computers that will comprise your wireless network. The wizard also provides an easy way to print your configuration information, so that it can be entered manually on devices that are not able to read information from removable media.

Who does this feature apply to?

Users that have computers with the following hardware components

- USB Host ports
- Wireless Networks Installed

Note The wizard can be run from a computer that does not have a wireless connection.

What new functionality is added to this feature in Windows XP Service Pack 2?

Wireless Network Setup Wizard
Detailed description

With the increasing popularity of wireless networking, more users are creating wireless networks in their home and networking computers and other devices that support wireless networking. Prior to Windows XP Service Pack 2 creating a secure wireless network and propagating the wireless settings to wireless clients and additional access points was very difficult. The Wireless Network Setup Wizard was designed to address the growing need for a simple yet secure method of configuring and bootstrapping wireless networking hardware (also known as, Wireless Access Points or WAPs) and wireless clients (including computer's and other devices).

The Wireless Network Setup Wizard provides a means for a Windows user to easily create and propagate network settings using an Extensible Markup Language (XML) schema and removable media. In the future this XML schema may also be used to transfer settings for wide area networks (WANs), local area networks LANs, as well as wireless LANs (WLANs). However, the XML files created by the Wireless Network Setup Wizard for Windows XP SP2 will only be used to transfer configuration settings for WLANs.

Why is this change important?

Many home wireless networks are deployed without security options, due to the complexity of adding security identifiers to all of the networked components. This wizard makes it simple for a user to create a wireless network with enhanced security. As more computers and devices are being manufactured with wireless hardware support standard, the demand for an easy method of configuring and implementing the feature has increased. In many cases, the devices do not have any method for inputting wireless network settings that have enhanced security. This wizard simplifies the configuration of such devices, so that deploying a wireless network with enhanced security is not a burdensome task. The following list provides examples of the devices that can be configured using the Wireless Network Setup Wizard:

- Wireless access points
- Desktop computers
- Network printers
- Photo stations
- Digital media receivers
- Electronic picture frames
- Set top boxes
- Portable computers and devices

What threats does it help mitigate?

Wireless networks that are deployed without security options. Wireless networks that are not secured provide an entry point to malicious users to access digital information and other assets of the network.

What works differently?

This is a new wizard that provides a new mechanism for configuring a wireless network. This wizard can be useful to configure computers and devices, even if they do not directly support this architecture. (You can create a wireless network and print the settings so that you can manually enter the settings into wireless devices and computers.)

Top of page 2 of 8