

# Implementing TCP/IP Security

**In this chapter:**

<b>Securing TCP/IP</b> .....	<b>197</b>
<b>Using IPSec</b> .....	<b>218</b>
<b>Best Practices</b> .....	<b>233</b>
<b>Additional Information</b> .....	<b>234</b>

TCP/IP is an industry-standard suite of protocols designed to facilitate communication between computers on large networks. TCP/IP was developed in 1969 by the U.S. Department of Defense Advanced Research Projects Agency (DARPA) as the result of a resource-sharing experiment called ARPANET (Advanced Research Projects Agency Network). Since 1969, ARPANET has grown into a worldwide community of networks known as the Internet, and TCP/IP has become the primary protocol used on all networks. Unfortunately, TCP/IP was not designed with security in mind and thus has very few security components by default. Consequently, it is often a source of network vulnerabilities. The Microsoft Windows operating system provides several methods that you can use to add security to TCP/IP, including securing the TCP/IP stack and using IP Security (IPSec). We will examine both techniques in this chapter.

## Securing TCP/IP

You cannot successfully secure computer networks without knowing how TCP/IP works. Nearly all computers today use TCP/IP as their primary network communication protocol. Thus, without physical access to a computer, an attacker must use TCP/IP to attack it. Consequently, TCP/IP security is often your first line of defense against attackers attempting to compromise your organization's network and therefore should be part of any defense-in-depth strategy for securing networks. You can configure additional security for the TCP/IP protocol stack in Microsoft Windows Server 2003, Windows 2000, and Windows XP to protect a computer against common attacks, such as denial-of-service attacks, and to help prevent attacks on applications that use the TCP/IP protocol.

## Understanding Internet Layer Protocols

TCP/IP primarily operates at two levels in the OSI model: the Internet layer and the transport layer. The Internet layer is responsible for addressing, packaging, and routing functions. The core protocols of the Internet layer include the Internet Protocol (IP), Address Resolution Protocol (ARP), and Internet Control Message Protocol (ICMP):

- **IP** A routable protocol responsible for logical addressing, routing, and the fragmentation and reassembly of packets
- **ARP** Resolves IP addresses to Media Access Control (MAC) addresses and vice versa
- **ICMP** Provides diagnostic functions and reporting errors for unsuccessful delivery of IP packets

The TCP/IP protocol suite includes a series of interconnected protocols called the *core protocols*. All other applications and protocols in the TCP/IP protocol suite rely on the basic services provided by several protocols, including IP, ARP, and ICMP.

### IP

IP is a connectionless, unreliable datagram protocol primarily responsible for addressing and routing packets between hosts. *Connectionless* means that a session is not established to manage the exchange of data. *Unreliable* means that delivery is not guaranteed. IP always makes a best-effort attempt to deliver a packet. An IP packet might be lost, delivered out of sequence, duplicated, or delayed. IP does not attempt to recover from these types of errors. The acknowledgment of packets delivered and the recovery of lost packets is the responsibility of a higher-layer protocol, such as Transmission Control Protocol (TCP). IP is defined in RFC 791.

An IP packet consists of an IP header and an IP payload. The IP header contains information about the IP packet, and the IP payload is the data being encapsulated by the IP protocol to be transmitted to the receiving host. The following list describes the key fields in the IP header:

- **Source IP Address** The IP address of the source of the IP datagram.
- **Destination IP Address** The IP address of the destination of the IP datagram.
- **Identification** Used to identify a specific IP datagram and all fragments of a specific IP datagram if fragmentation occurs.
- **Protocol** Informs IP at the destination host whether to pass the packet up to TCP, User Datagram Protocol (UDP), ICMP, or other protocols.

- **Checksum** A simple mathematical computation used to verify the integrity of the IP header. If the IP header does not match the checksum, the receiving host will disregard the packet. This checksum does not include any information outside the IP header.
- **Time To Live (TTL)** Designates the number of networks on which the datagram is allowed to travel before being discarded by a router. The TTL is set by the sending host and is used to prevent packets from endlessly circulating on an IP network. When forwarding an IP packet, routers decrease the TTL by at least one.
- **Fragmentation And Reassembly** If a router receives an IP packet that is too large for the network to which the packet is being forwarded, IP fragments the original packet into smaller packets that fit on the downstream network. When the packets arrive at their final destination, IP on the destination host reassembles the fragments into the original payload. This process is referred to as fragmentation and reassembly. Fragmentation can occur in environments that have a mix of networking technologies, such as Ethernet and Token Ring. The fragmentation and reassembly process works as follows:
  1. When an IP packet is sent, the sending host places a unique value in the Identification field.
  2. The IP packet is received at the router. If the router determines that the Maximum Transmission Unit (MTU) of the network onto which the packet is to be forwarded is smaller than the size of the IP packet, the router fragments the original IP payload into multiple packets, each of which is smaller than the receiving network's MTU size. Each fragment is sent with its own IP header that contains the following:
    - ❑ The original Identification field, which identifies all fragments that belong together.
    - ❑ The More Fragments flag, which indicates that other fragments follow. The More Fragments flag is not set on the last fragment because no other fragments follow it.
    - ❑ The Fragment Offset field, which indicates the position of the fragment relative to the original IP payload.
  3. When the fragments are received by the destination host, they are identified by the Identification field as belonging together. The Fragment Offset field is then used to reassemble the fragments into the original IP payload.

## ARP

Address Resolution Protocol performs IP address-to-MAC address resolution for outgoing packets. As each outgoing addressed IP datagram is encapsulated in a frame, source and destination MAC addresses must be added. Determining the destination MAC address for each frame is the responsibility of ARP. ARP is defined in RFC 826.

## ICMP

Internet Control Message Protocol provides troubleshooting facilities and error reporting for packets that are undeliverable. For example, if IP is unable to deliver a packet to the destination host, ICMP sends a Destination Unreachable message to the source host. Table 10-1 shows the most common ICMP messages.

**Table 10-1 Common ICMP Messages**

Message	Description
Echo Request	Troubleshooting message used to check IP connectivity to a desired host. The Ping utility sends ICMP Echo Request messages.
Echo Reply	Response to an ICMP Echo Request.
Redirect	Sent by a router to inform a sending host of a better route to a destination IP address.
Source Quench	Sent by a router to inform a sending host that its IP datagrams are being dropped because of congestion at the router. The sending host then lowers its transmission rate.
Destination Unreachable	Sent by a router or the destination host to inform the sending host that the datagram cannot be delivered.

When the result of an ICMP request is a Destination Unreachable message, a specific message is returned to the requestor detailing why the Destination Unreachable ICMP message was sent. Table 10-2 describes the most common of these messages.

**Table 10-2 Common ICMP Destination Unreachable Messages**

Unreachable Message	Description
Host Unreachable	Sent by an IP router when a route to the destination IP address cannot be found
Protocol Unreachable	Sent by the destination IP node when the Protocol field in the IP header cannot be matched with an IP client protocol currently loaded
Port Unreachable	Sent by the destination IP node when the destination port in the UDP header cannot be matched with a process using that port
Fragmentation Needed and DF Set	Sent by an IP router when fragmentation must occur but is not allowed because of the source node setting the Don't Fragment (DF) flag in the IP header

ICMP does not make IP a reliable protocol. ICMP attempts to report errors and provide feedback on specific conditions. ICMP messages are carried as unacknowledged IP datagrams and are themselves unreliable. ICMP is defined in RFC 792.

## Understanding Transport Layer Protocols

The transport layer is responsible for providing session and datagram communication services over the IP protocol. The two core protocols of the transport layer are the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP):

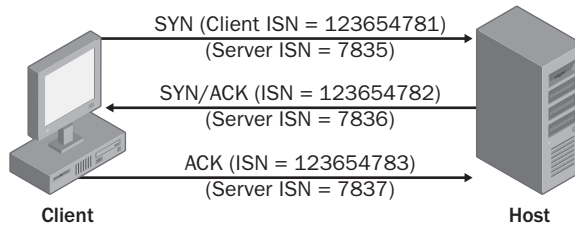
- **TCP** Provides a one-to-one, connection-oriented, reliable communications service. TCP is responsible for the establishment of a TCP connection, the sequencing and acknowledgment of packets sent, and the recovery of packets lost during transmission.
- **UDP** Provides a one-to-one or one-to-many, connectionless, unreliable communications service. UDP is used when the amount of data to be transferred is small (such as data that fits into a single packet), when the overhead of establishing a TCP connection is not desired, or when the applications or upper-layer protocols provide reliable delivery.

### How TCP Communication Works

When two computers communicate using TCP, the computer that initiates the communication is known as the client, regardless of whether it is running a client or server operating system, and the responding computer is known as the host. If the client and host are on the same network segment, the client computer first uses ARP to resolve the host's MAC address by sending a broadcast for the IP address of the host. Once the client has the MAC address of the host, it can commence communication to the port on the host by using the transport layer protocol specified by the application. There are 65,535 TCP and UDP ports, beginning with 0. Ports 1023 and below are regarded as well-known ports for legacy reasons, and ports above 1023 are known as high ports. Functionally, no difference exists between the well-known ports and the high ports. On the host, an application is bound to a certain port it specifies and is initialized in a listening state, where it waits for requests from a client. When the client initiates a connection to a TCP port, a defined series of packets, known as a three-way handshake and illustrated in Figure 10-1, constructs a session for reliable packet transmission. The steps for establishing connections follow:

1. The client sends the host a synchronization (SYN) message that contains the host's port and the client's Initial Sequence Number (ISN). TCP sequence numbers are 32 bits in length and are used to ensure session reliability by facilitating out-of-order packet reconstruction.

2. The host receives the message and sends back its own SYN message and an acknowledgment (ACK) message, which includes the host's ISN and the client's ISN incremented by 1.
3. The client receives the host's response and sends an ACK, which includes the ISN from the host incremented by 1. After the host receives the packet, the TCP session is established.



**Figure 10-1** Three-way TCP handshake

When the communication between the client and host is complete, the session is closed once the following steps occur:

1. The client sends a finalization (FIN) message to the host. The session is now *half closed*. The client no longer sends data but can still receive data from the host. Upon receiving this FIN message, the host enters a passive closed state.
2. The host sends an ACK message, which includes the client's sequence number augmented by 1.
3. The host ends its own FIN message. The client receives the FIN message and returns an ACK message that includes the host's sequence number augmented by 1.
4. Upon receiving this ACK message, the host closes the connection and releases the memory the connection was using.

## The Netstat.exe Command

To see port activity on your computers that run Windows Server 2003, Windows 2000, or Windows XP, you can use the Netstat.exe command. Netstat.exe will also show the status of TCP ports. The syntax for using Netstat.exe follows, and Table 10-3 describes the options available when using this command.

```
NETSTAT [-a] [-e] [-n] [-o] [-s] [-p proto] [-r] [interval]
```

Table 10-3 Netstat.exe Options

Option	Description
-a	Displays all connections and listening ports.
-e	Displays Ethernet statistics. This can be combined with the -s option.
-n	Displays addresses and port numbers in numerical form.
-o	Displays the owning process ID (PID) associated with each connection. This option does not exist in Windows 2000.
-p <i>protocol</i>	Shows connections for the protocol specified by <i>protocol</i> , which can be TCP, UDP, TCPv6, or UDPv6. If used with the -s option to display per-protocol statistics, the value for <i>protocol</i> can be IP, ICMP, TCP, or UDP.
-r	Displays the routing table.
-s	Displays per-protocol statistics. By default, statistics are shown for IP, ICMP, TCP, and UDP.
<i>interval</i>	Determines the refresh interval for the data displayed by Netstat.



**Tip** To find the process associated with a given active port in Windows Server 2003 and Windows XP, you can locate the PID associated with the port by typing `netstat-ano`. You can then find the process associated with the PID by typing `tasklist /FI "PID eq XX"`, where *XX* is the PID of the process.

As mentioned in Table 10-3, the -o option of Netstat.exe is not available in Windows 2000; however, you can download utilities from the Internet that have similar functionality that run in Windows 2000.

## Common Threats to TCP/IP

Several types of threats to TCP/IP can either compromise network security or lead to information disclosure. Although these attacks are more prevalent on the Internet, you should be concerned about them on internal computers as well. These common threats include the following:

- Port scanning
- Spoofing and hijacking
- Denial of service

### Port Scanning

To communicate with TCP/IP, applications running on host computers must listen for incoming TCP or UDP connections, and host operating systems must listen for broadcast and other network maintenance traffic. By scanning a computer to see which ports a host is listening for and which protocols it uses, an attacker might be able to

locate weaknesses in the host that he can later use to attack the computer. Attackers often perform port scans to reveal this information. Several types of port scans exist:

- **Ping sweeps** An attacker might use an automated tool to send ICMP Echo Request packets to entire networks or subnets. By default, all active hosts will respond unless they have firewalls enabled that filter ICMP traffic. This lets the attacker know that the host exists and is active. An attacker can also analyze the structure of the ICMP packet to determine the operating system running on the host.
- **Port enumeration** An attacker might want to enumerate all the services running on a host computer. Because hosts must respond to client computers to carry out legitimate operations, attackers can exploit this behavior to obtain critical information.



**Tip** You can download a command-line port-scanning tool from Microsoft called Port Query (Portqry.exe). This tool, found at <http://support.microsoft.com/kb/832919>, tests the security of hosts and performs network diagnostics. Port Query 2.0 also includes enhancements that enable it to retrieve basic information from services that communicate through session and application layer protocols, such as Lightweight Directory Access Protocol (LDAP) and remote procedure call (RPC). In addition, many free utilities that can perform port scans are available on the Internet.

Additionally, you can download a tool from Microsoft called Port Reporter (PortRptr.exe). Port Reporter runs as a system service that logs packets that are sent and received and the processes that sent or received them. You can download Port Reporter from the Microsoft Web site at <http://support.microsoft.com/kb/837243>. Conveniently, Port Reporter also has a companion parsing utility that you can download to help analyze the log files generated by Port Reporter.

- **Banner grabbing** Many common services respond with banners when sessions are initiated or requested. These banners contain basic information on the service or server. For example, by using Telnet to connect to port 25 of a Windows 2000 server running the default Simple Mail Transfer Protocol (SMTP) service, you can retrieve this banner:

```
220 SF0FS001.finance.woodgrovebank.com Microsoft ESMTp MAIL Service, Version: 5.0.2195.5329 ready at Sat, 12 Oct 2002 16:18:44 -0800
```

From interpreting this banner, you can determine that the target server is named SF0FS001. Given the version number, 5.0.2195.5329, the server SF0FS001 is probably a file server running Windows 2000 with Service Pack 3 installed and is physically located in the Pacific Time zone—most likely in San Francisco. The server is running a built-in instance of the SMTP service, which is

installed as part of Microsoft Internet Information Services (IIS) 5.0. Knowing that IIS is installed by default in Windows 2000 and that this server does not appear to be a Web server, it is likely that the server has a default installation of Windows 2000.



**Important** Changing service banners can also break applications that rely on them for information about the server they are communicating with. Furthermore, changing banners can break an application running on the computer that uses the information from service banners from other services running on the computer.

- **Half scan** This type of port scanning does not follow the precise TCP three-way handshake protocol and leaves TCP connections half open. Because most host System logs do not log packets until the host receives the final ACK, half scans can enable an attacker to gain information about a host without being detected.

## Spoofing and Hijacking

Attackers might want to spoof or mimic legitimate TCP/IP packets to attack a computer or network. Usually, spoofing a packet requires that the attacker handcraft a TCP/IP packet and send it to either the host she wants to attack or a third-party host that she has previously compromised to attack the targeted host or network. Many types of spoofing attacks exist. The following three are among the most well known:

- **Land attack** Takes advantage of security flaws in the many implementations of TCP/IP. To carry out a land attack, an attacker opens a legitimate TCP session by sending a SYN packet but spoofs the packet so that the source address and port and the destination address and port match the host IP address and the port to which the packet is being sent.

For example, to carry out a land attack on an e-mail server with the IP address 192.168.183.200, an attacker can create a packet with the source address of 192.168.183.200 and the source port of 25, rather than using the source address and port of his own computer. Now the source and destination addresses will be the same, as will the source and destination ports. If not patched to protect against the land attack, the packet will continually attempt to make a connection with itself on its own port 25, resulting in a denial-of-service situation.

- **Smurf attack** Uses a third-party network to carry out a denial-of-service attack on a host system by spoofing an ICMP Echo Request packet. The attacker obtains the host IP address and creates a forged ICMP Echo Request packet that looks like it came from the host IP address. The attacker sends thousands of copies of the spoofed packet to the broadcast address on an improperly secured

third-party network. This results in every computer in the third-party network responding to each spoofed packet by sending an ICMP Echo Reply packet to the host system. The amount of ICMP traffic that is generated by this attack will deny legitimate traffic from reaching the target host.

- **Session hijacking** Takes advantage of flaws in many implementations of the TCP/IP protocol by anticipating TCP sequence numbers to hijack a session with a host. To hijack a TCP/IP session, the attacker creates a legitimate TCP session to the targeted host, records the TCP sequence numbers used by the host, and then computes the round-trip time (RTT). This step often takes many exchanges in sequence. Using the stored sequence numbers and the RTT, the attacker can potentially predict future TCP sequence numbers. The attacker can then send a spoofed packet to another host, using the targeted host IP address as the source address and the next sequence number. If successful, the second host system will believe the packet originated from the targeted system and accept packets from the attacker. This type of attack is particularly effective when the second host trusts the targeted host.



**More Info** IP spoofing by predicting TCP/IP sequence numbers was the basis for the famous Christmas 1994 attack on Tsutomu Shimomura by Kevin Mitnick. The attack is chronicled in the book *Takedown: The Pursuit and Capture of Kevin Mitnick, America's Most Wanted Computer Outlaw—By the Man Who Did It* (Hyperion, 1996).

## Denial of Service

Denial-of-service attackers attempt to exploit the way the TCP/IP protocol works to prevent legitimate traffic from reaching the host system. One of the most common types of denial-of-service attacks is a SYN flood. A SYN flood attempts to create a situation in which the host system's maximum TCP connection pool is locked in a half-open state, thus denying legitimate traffic to and from the host. To carry out a SYN flood, the attacker creates a spoofed IP packet with an unreachable IP address for a source address, or she clips the receive wire on the Ethernet cable she is using. When the host receives the packet, it responds by sending a SYN/ACK response and waits for the final ACK in the TCP three-way handshake, which never comes. The session will remain in the half-open state until the predefined time-out is reached. This process is repeated until no more TCP sessions are allowed by the host system, which then cannot create any new sessions.



**More Info** See *Assessing Network Security* (Microsoft Press, 2004) by Kevin Lam, David LeBlanc, and Ben Smith for more information on common attacks on TCP/IP.

## Configuring TCP/IP Security in Windows

The remainder of this section presents several ways you can secure your computers that run Windows Server 2003, Windows 2000, and Windows XP against attacks on TCP/IP, including basic TCP/IP binding configurations, custom registry settings, and TCP/IP filtering.

### Implementing Basic TCP/IP Security

Three basic settings, outlined in the following list, will increase the security of TCP/IP for each network adapter in Windows Server 2003, Windows 2000, and Windows XP. You will need to ensure that each of these settings is compatible with your network and the applications that either run on the computer or must be accessible from the computer.

- **File And Printer Sharing For Microsoft Networks** By default, File And Printer Sharing For Microsoft Networks is bound on all network interfaces. The File And Printer Sharing For Microsoft Networks component enables other computers on a network to access resources on your computer. By removing the binding to File And Printer Sharing For Microsoft Networks from a network interface, you can prevent other computers from enumerating or connecting to files and printers that have been shared through that network interface. Stopping the Server service will also prevent a computer from hosting file or print shares. After removing this binding from a network interface, the File and Print Services computer will no longer listen for Server Message Block (SMB) connections on TCP port 139 of that interface but will still listen on port 445 for other SMB packets. Direct hosted “NetBIOS-less” SMB traffic uses port 445 (TCP and UDP). If NetBIOS is still used for other services on the computer, port 139 will still be listening, just not for File and Print Services. Removing this setting will not interfere with the computer’s ability to connect to other shared files or printers. You can unbind File And Printer Sharing For Microsoft Networks in the Network And Dial-Up Connections Control Panel or on the Properties page of the network interface. You can prevent the Windows operating system from listening for direct SMB traffic by deleting the default value from the registry entry `HKLM\SYSTEM\CurrentControlSet\Services\NetBT\Parameters\TransportBindName`. A host-based firewall, such as Windows Firewall in Windows XP and Windows Server 2003, is also an effective option for preventing other computers from connecting to SMB ports.

- **NetBIOS Over TCP/IP** Windows Server 2003, Windows 2000, and Windows XP support file and printer sharing traffic by using the SMB protocol directly hosted on TCP. This differs from earlier operating systems in which SMB traffic requires the NetBIOS over TCP/IP (NetBT) protocol to work on a TCP/IP transport. If both the direct-hosted and NetBT interfaces are enabled, both methods are tried at the same time and the first to respond is used. This enables the Windows operating system to function properly with operating systems that do not support direct hosting of SMB traffic. NetBIOS over TCP/IP traditionally uses the following ports:

NetBIOS name	137/UDP
NetBIOS name	137/TCP
NetBIOS datagram	138/UDP
NetBIOS session	139/TCP



**Note** Direct-hosted “NetBIOS-less” SMB traffic uses port 445 (TCP and UDP). If you disable NetBIOS over TCP/IP (NetBT) and unbind File And Printer Sharing For Microsoft Networks, the computer will no longer respond to any NetBIOS requests. Applications and services that depend on NetBT will no longer function once NetBT is disabled. Therefore, verify that your clients and applications no longer need NetBT support before you disable it.

- **DNS Registration** By default, computers running Windows Server 2003, Windows 2000, and Windows XP attempt to register their host names and IP address mappings automatically in the Domain Name System (DNS) for each adapter. If your computer is using a public DNS server or cannot reach the DNS server, as is often the case when the computer resides in a perimeter network, you should remove this behavior on each adapter.

## Configuring Registry Settings

Denial-of-service attacks are network attacks aimed at making a computer or a particular service on a computer unavailable to network users. Denial-of-service attacks can be difficult to defend against. To help prevent denial-of-service attacks, you can harden the TCP/IP protocol stack on computers that run Windows Server 2003, Windows 2000, and Windows XP. You should harden the TCP/IP stack against denial-of-service attacks, even on internal networks, to prevent denial-of-service attacks that originate from inside the network as well as on computers attached to public networks. You can harden the TCP/IP stack by customizing these registry values, which are stored in the registry key `HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\`:

- **EnableICMPRedirect** When ICMP redirects are disabled (by setting the value to 0), attackers cannot carry out attacks that require a host to redirect the ICMP-based attack to a third party.
- **SynAttackProtect** Enables SYN flood protection in Windows Server 2003, Windows 2000, and Windows XP. You can set this value to 0, 1, or 2. The default setting, 0, provides no protection. Setting the value to 1 will activate SYN/ACK protection contained in the TCPMaxPortsExhausted, TCPMaxHalfOpen, and TCPMaxHalfOpenRetried values. Setting the value to 2 will protect against SYN/ACK attacks by more aggressively timing out open and half-open connections and preventing scalable windows. In Windows Server 2003, you can set this to be either on (1) or off (0). Turning it on is effectively the same as setting it to 2 in Windows 2000 and Windows XP. Windows Server 2003 Service Pack 1 enables SynAttackProtect.
- **TCPMaxConnectResponseRetransmissions** Determines how many times TCP retransmits an unanswered SYN/ACK message. TCP retransmits acknowledgments until the number of retransmissions specified by this value is reached.
- **TCPMaxHalfOpen** Determines how many connections the server can maintain in the half-open state before TCP/IP initiates SYN flooding attack protection. This entry is used only when SYN flooding attack protection is enabled on this server—that is, when the value of the SynAttackProtect entry is 1 or 2 and the value of the TCPMaxConnectResponseRetransmissions entry is at least 2.
- **TCPMaxHalfOpenRetired** Determines how many connections the server can maintain in the half-open state even after a connection request has been retransmitted. If the number of connections exceeds the value of this entry, TCP/IP initiates SYN flooding attack protection. This entry is used only when SYN flooding attack protection is enabled on this server—that is, when the value of the SynAttackProtect entry is 1 and the value of the TCPMaxConnectResponseRetransmissions entry is at least 2.
- **TCPMaxPortsExhausted** Determines how many connection requests the system can refuse before TCP/IP initiates SYN flooding attack protection. The system must refuse all connection requests when its reserve of open connection ports runs out. This entry is used only when SYN flooding attack protection is enabled on this server—that is, when the value of the SynAttackProtect entry is 1, and the value of the TCPMaxConnectResponseRetransmissions entry is at least 2.
- **TCPMaxDataRetransmissions** Determines how many times TCP retransmits an unacknowledged data segment on an existing connection. TCP retransmits data segments until they are acknowledged or until the number of retransmissions specified by this value is reached.

- **EnableDeadGWDetect** Determines whether the computer will attempt to detect dead gateways. When dead gateway detection is enabled (by setting this value to 1), TCP might ask IP to change to a backup gateway if a number of connections are experiencing difficulty. Backup gateways are defined in the TCP/IP Configuration dialog box in Network Control Panel for each adapter. When you leave this setting enabled, it is possible for an attacker to redirect the server to a gateway of his choosing.
- **EnablePMTUDiscovery** Determines whether path MTU discovery is enabled (1), for which TCP attempts to discover the largest packet size over the path to a remote host. When path MTU discovery is disabled (0), the path MTU for all TCP connections is fixed at 576 bytes.
- **DisableIPSourceRouting** Determines whether a computer allows clients to pre-determine the route that packets take to their destination. When this value is set to 2, the computer will disable source routing for IP packets.
- **KeepAliveTime** Determines how often TCP attempts to verify that an idle connection is still intact by sending a keep-alive packet. If the remote computer is still active, it will respond and the session will remain open. Keep-alive packets are not automatically sent by the TCP/IP stack in the Windows operating system. The default value is set to 2 hours (7,200,000) when keep-alive transmissions are enabled.
- **NoNameReleaseOnDemand** Determines whether the computer will release its NetBIOS name if requested by another computer or a malicious packet attempting to hijack the computer's NetBIOS name.
- **PerformRouterDiscovery** Determines whether the computer performs router discovery on this interface. Router discovery solicits router information from the network and adds the information retrieved to the route table. Setting this value to 0 prevents the interface from performing router discovery.

Table 10-4 lists the registry entries that you can make to harden the TCP/IP stack on your computers that run Windows Server 2003, Windows 2000, and Windows XP. These settings must be added to the registry and configured appropriately.

**Table 10-4 Registry Settings to Harden TCP/IP**

Value	Data (DWORD)
EnableSecurityFilters	0
SynAttackProtect	2 (1 in Windows Server 2003)
TCPMaxConnectResponseRetransmissions	2
TCPMaxHalfOpen	500
TCPMaxHalfOpenRetired	400
TCPMaxPortsExhausted	5
TCPMaxDataRetransmissions	3

Table 10-4 Registry Settings to Harden TCP/IP

Value	Data (DWORD)
EnableDeadGWDetect	0
EnablePMTUDiscovery	0
DisableIPSourceRouting	2
KeepAliveTime	300,000
NoNameReleaseOnDemand	1
PerformRouterDiscovery	0

Additionally, you can secure the TCP/IP stack for Windows Sockets (Winsock) applications such as FTP servers and Web servers. The driver Afd.sys is responsible for connection attempts to Winsock applications. Afd.sys in Windows Server 2003, Windows 2000, and Windows XP supports large numbers of connections in the half-open state without denying access to legitimate clients. Afd.sys can use a dynamic backlog, which is configurable, rather than a static backlog. You can configure four parameters for the dynamic backlog:

- **EnableDynamicBacklog** Switches between using a static backlog and a dynamic backlog. By default, this parameter is set to 0, which enables the static backlog. You should enable the dynamic backlog for better security on Winsock.
- **MinimumDynamicBacklog** Controls the minimum number of free connections allowed on a listening Winsock endpoint. If the number of free connections drops below this value, a thread is queued to create additional free connections. Making this value too large (setting it to a number greater than 100) will degrade the performance of the computer.
- **MaximumDynamicBacklog** Controls the maximum number of half-open and free connections to Winsock endpoints. If this value is reached, no additional free connections will be made.
- **DynamicBacklogGrowthDelta** Controls the number of Winsock endpoints in each allocation pool requested by the computer. Setting this value too high can cause system resources to be occupied unnecessarily.

Each of these values must be added to the registry key HKLM\SYSTEM\CurrentControlSet\Services\AFD\Parameters. Table 10-5 lists the parameters and the recommended levels of protection.

Table 10-5 Registry Settings to Harden Winsock

Value	Data (DWORD)
EnableDynamicBacklog	1
MinimumDynamicBacklog	20
MaximumDynamicBacklog	20,000
DynamicBacklogGrowthDelta	10

## Using TCP/IP Filtering

Windows Server 2003, Windows 2000, and Windows XP include support for TCP/IP filtering, a feature known as TCP/IP Security in Microsoft Windows NT 4.0. TCP/IP filtering enables you to specify which types of inbound local host IP traffic are processed for all interfaces. This feature prevents traffic from being processed by the computer in the absence of other TCP/IP filtering, such as that provided by Routing and Remote Access Service (RRAS), Windows Firewall (in Windows XP and Windows Server 2003 SP1), and other TCP/IP applications or services. TCP/IP filtering is disabled by default.

When configuring TCP/IP filtering, you can permit either all or only specific ports or protocols listed for TCP ports, UDP ports, or IP protocols. Packets destined for the host are accepted for processing if they meet one of the following criteria:

- The destination TCP port matches the list of TCP ports.
- The destination UDP port matches the list of UDP ports.
- The IP protocol matches the list of IP protocols.
- The packet is an ICMP packet.



**Note** TCP/IP port filtering applies to all interfaces on the computer and cannot be applied on a per-adapter basis. However, you can configure allowed ports and protocols on a per-adapter basis.

In addition to being able to configure TCP/IP filtering in the Options tab of the TCP/IP advanced properties in the user interface, you can apply the settings directly to the registry. Table 10-6 lists the registry values to configure TCP/IP filtering. TCP/IP filtering is set in the key HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters, whereas the specific settings for each interface are configured in the key HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\*Interface\_GUID*.

**Table 10-6 Registry Values for TCP/IP Filtering**

Setting	Type	Description
EnableSecurityFilters	DWORD	1 enables TCP/IP filtering; 0 disables TCP/IP filtering.
UdpAllowedPorts	MULTI_SZ	0 allows all UDP ports; an empty (null) value blocks all UDP ports; otherwise, the specific allowed UDP ports are listed.
TCPAllowedPorts	MULTI_SZ	0 allows all TCP ports; an empty (null) value blocks all TCP ports; otherwise, the specific allowed TCP ports are listed.

**Table 10-6 Registry Values for TCP/IP Filtering**

Setting	Type	Description
RawIpAllowedProtocols	MULTI_SZ	0 allows all IP protocols; an empty (null) value blocks all IP protocols; otherwise, the specific allowed IP protocols are listed.

## Using Windows Firewall in Windows Server 2003 and Windows XP

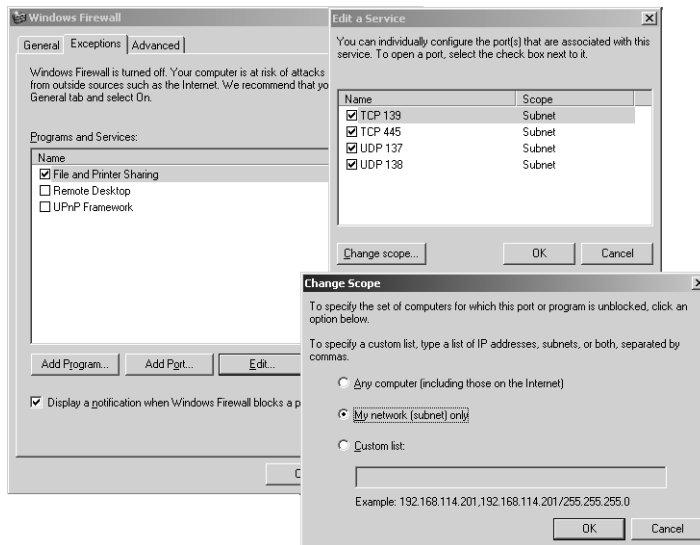
Windows Server 2003 and Windows XP both include a personal firewall called Internet Connection Firewall (ICF) in their initial release. Its much improved successor, called Windows Firewall, was released in Windows Server 2003 Service Pack 1 and Windows XP Service Pack 2. Windows Firewall is a stateful firewall—it monitors incoming traffic received by the network adapter configured to use Windows Firewall and it inspects the source and destination addresses of each message that it handles. To prevent unsolicited traffic from the public side of the connection from entering the private side, Windows Firewall keeps a table of all communications that have originated from the Windows Firewall computer. When used in conjunction with Internet Connection Sharing (ICS), Windows Firewall creates a table for tracking all traffic originated from the Windows Firewall/ICS computer and all traffic originated from private network computers. Inbound Internet traffic is allowed to reach the computers in your network only when a matching entry in the table shows that the communication exchange originated within your computer or private network or is permitted by rule to an application or port.

Windows Firewall improves on ICF, which operated on a binary basis, either off or on, by adding an option to enable the firewall but not allow exemptions to the packet filtering. This mode blocks all unsolicited inbound traffic. As an administrator, you can determine whether users, even if they are not local administrators, can enable Windows Firewall through the Group Policy option Prohibit Use Of Internet Connection Firewall On Your DNS Domain Network, which is listed in the computer portion of Group Policy under Administrative Templates, Network, Network Connections.

Additionally, through the use of profiles, network administrators can configure Windows Firewall settings to be different (presumably more relaxed) when connected to the corporate network. The two profiles that are added to Group Policy in Windows XP SP2 and Windows Server 2003 SP1 are the Domain profile and the Standard profile. The client computer determines whether it is connected to the corporate network by using the Network Location Awareness (NLA) system service. When the network adapter is initialized when connecting to a network, the NLA service compares the connection-specific DNS suffix of the connection it received from Group Policy to the domain suffix of the domain to which the computer belongs. If the two match, NLA judges that the computer is connected to the corporate network, and the domain

profile is applied. Because the DNS suffix is generally provided by the Dynamic Host Configuration Protocol (DHCP) server from which the computer gets its IP address, the use of profiles should not be used on networks that require a great degree of security. An attacker with ample resources could potentially force a client computer into the domain profile if she has sufficient knowledge of the target's network. Although this scenario is remote at best, it is worth considering the security conditions under which your network operates.

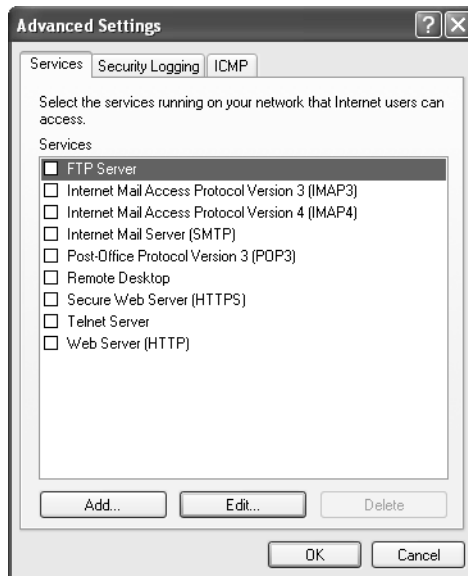
Using Windows Firewall, you can create exceptions by defining which programs and services are allowed to receive unsolicited traffic from other computers. Exceptions configured through Programs and Services are global on the system, whereas exceptions configured through Services are interface specific. For programs and services, which can be configured locally or through Group Policy, Windows Firewall enables you to create an exception when traffic originates on the local subnet or from within a predefined range of IP addresses, but it blocks the traffic when the traffic originates from remote networks. Figures 10-2 and 10-3 show the configuration of the predefined File And Printer Sharing exception, which restricts the scope of the exception to the local subnet.



**Figure 10-2** Restricting a service to accept packets from only the local subnet

If an application needs to receive unsolicited traffic when Windows Firewall is enabled and is blocked by Windows Firewall, for administrators, the operating system will prompt the user to grant the application executable a specific exception to the rule while non-administrators will receive an error message. You can also add applications to the Programs and Services list by the name of the executable rather than having to create a static, always-open port such as was required in the Internet Connection Firewall.

You can configure services to allow unsolicited traffic from the Internet to be forwarded by the Windows Firewall computer to the private network. For example, if you are hosting an HTTP Web server service and have enabled the HTTP service on your Windows Firewall computer, unsolicited HTTP traffic will be forwarded by the Windows Firewall computer to the HTTP Web server. A set of operational information, known as a *service definition*, is required by Windows Firewall to allow the unsolicited Internet traffic to be forwarded to the Web server on your private network. The Services tab of Windows Firewall Advanced Settings dialog box is shown in Figure 10-3.



**Figure 10-3** Services tab of Windows Firewall Advanced Settings dialog box

If there is no service definition for the service that you would like to allow to be connected, you can add custom services in the Services tab of the Advanced Settings dialog box. Windows Firewall can also perform port translation for incoming connections. When you create a custom service, you will need to specify the following:

- **Description of service** Determines how the service is displayed in the Services tab
- **Name or IP address** Determines the host name or IP address of the computer offering the service if the service is not hosted on the local computer
- **External port** Defines the TCP or UDP port on the Windows Firewall computer that will listen to inbound traffic to the service
- **Internal port** Defines the TCP or UDP port to which the Windows Firewall computer will forward the inbound traffic to the computer defined in the Name Or IP Address field

Communications that originate from a source outside the Windows Firewall computer, such as the Internet, are dropped by the firewall unless an entry in the Services tab is made to allow passage. Windows Firewall silently discards unsolicited communications, preventing common attacks, such as port scanning and NetBIOS enumeration. Windows Firewall does not block outgoing network traffic because this provides little extra protection and is completely unusable by average users.

Windows Firewall can create a Security log so you can view the activity that is tracked by the firewall. You can choose whether to log dropped, successful, or dropped and successful packets. By default, packets are logged to %systemroot%\pfirewall.log. The log file has a default maximum size of 4098 KB. Table 10-7 describes the fields in the packet log file.

**Table 10-7 Description of Information Logged by Windows Firewall**

Field	Description
Date	Specifies the date that the recorded transaction occurred in the format <i>YY-MM-DD</i> .
Time	Specifies the time that the recorded transaction occurred in the format <i>HH:MM:SS</i> .
Action	Specifies which operation was observed by the firewall. The options available to the firewall are OPEN, CLOSE, DROP, and INFO-EVENTS-LOST. An INFO-EVENTS-LOST action indicates the number of events that happened but were not placed in the log.
Protocol	Specifies which IP protocol was used for the communication.
Src-ip	Specifies the source IP address of the computer attempting to establish communications.
Dst-ip	Specifies the destination IP address of the communication attempt.
Src-port	Specifies the source port number of the sending computer. Only TCP and UDP will return a valid src-port entry.
Dst-port	Specifies the port of the destination computer. Only TCP and UDP will return a valid dst-port entry.
Size	Specifies the packet size in bytes.
Tcpflags	Specifies the TCP control flags found in the TCP header of an IP packet: <ul style="list-style-type: none"> <li>■ <b>ACK</b> Acknowledgment field significant</li> <li>■ <b>FIN</b> No more data from sender</li> <li>■ <b>PSH</b> Push function</li> <li>■ <b>RST</b> Reset the connection</li> <li>■ <b>SYN</b> Synchronize sequence numbers</li> <li>■ <b>URG</b> Urgent Pointer field</li> </ul>
Tcpsyn	Specifies the TCP synchronization number in the packet.
Tcpack	Specifies the TCP acknowledgment number in the packet.
Tcpwin	Specifies the TCP window size in bytes in the packet.

**Table 10-7 Description of Information Logged by Windows Firewall**

Field	Description
lcmptype	Specifies a number that represents the Type field of the ICMP message.
lcmpcode	Specifies a number that represents the Code field of the ICMP message.
Info	Specifies an information entry that depends on the type of action that occurred. For example, an INFO-EVENTS-LOST action will create an entry of the number of events that happened but were not placed in the log since the last occurrence of this event type.

In addition to the Group Policy objects that were added to manage Windows Firewall configuration, Windows Server 2003 SP1 and Windows XP SP2 add the ability to configure the firewall during scripted installations and from the command line.

For scripted installations, you can create a configuration file called `Netfw.inf` to replace the version on the installation media for Windows XP SP2:

1. Extract `Netfw.inf` from the installation media for Windows XP SP2, or copy it from an unaltered installation.
2. Make the desired changes to the configuration by editing the INF file.
3. Save the modified INF file as `Netfw.inf`.
4. Replace the default `Netfw.inf` with the modified `Netfw.inf` on the installation share for Windows XP SP2 or run the command `Netsh firewall reset` on the computer running Windows XP with SP2 for computers that have already had the operating system installed. If you are completing scripted installs from CD-based media, you can copy the modified `Netfw.inf` to the computer during installation and run the `Netsh firewall reset` command from a script called in the run-once registry key.
5. For information on customizing the `Netfw.inf` file, see the white paper: “Using the Windows Firewall INF File in Microsoft Windows XP Service Pack 2” on the Microsoft Web site at  
<http://www.microsoft.com/downloads/details.aspx?FamilyID=cb307a1d-2f97-4e63-a581-bf25685b4c43&displaylang=en>
6. From the command line or programmatically from batch file or scripts, you can use `Netsh` to manage the Windows Firewall configuration. For example, to see the current settings of Windows Firewall, you can type “**netsh firewall show config**” at the command prompt. You can also add and delete settings as well as revert to the settings in the `Netfw.inf` file. Because the network shell uses IntelliSense context-specific technology, you only need type the first couple unique letters from each string, for example, “`netsh fi sh co`” is the same as “`netsh firewall show config`.” This is nice for those of us who are constantly challenged by typing skills (or patience) that leave something to be desired.

## Using IPsec

By its design, TCP/IP is an open protocol created to connect heterogeneous computing environments with the least amount of overhead possible. As is often the case, interoperability and performance design goals do not generally result in security—and TCP/IP is no exception to this. TCP/IP provides no native mechanism for the confidentiality or integrity of packets. To secure TCP/IP, you can implement IP Security. IPsec implements encryption and authenticity at a lower level in the TCP/IP stack than application-layer protocols such as Secure Sockets Layer (SSL) and Transport Layer Security (TLS). Because the protection process takes place lower in the TCP/IP stack, IPsec protection is transparent to applications. IPsec is a well-defined, standards-driven technology.

The IPsec process encrypts the payload after it leaves the application at the client and then decrypts the payload before it reaches the application at the server. An application does not have to be IPsec aware because the data transferred between the client and the server is normally transmitted in plaintext.

IPsec is composed of two protocols that operate in two modes with three different authentication methods. IPsec is policy driven and can be deployed centrally by using Group Policy. To deploy IPsec, you must determine the following:

- Protocol
- Mode
- Authentication methods
- Policies

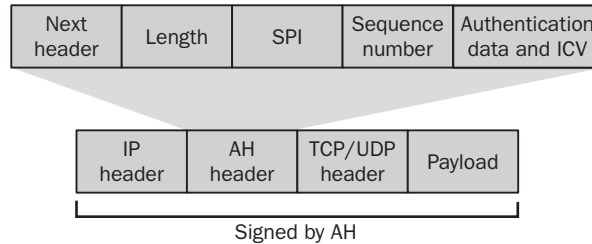
## Securing Data Transmission with IPsec Protocols

As mentioned, IPsec is composed of two protocols: IPsec Authentication Header (AH) and IPsec Encapsulating Security Payload (ESP). Each protocol provides different services; AH primarily provides packet integrity services, whereas ESP provides packet confidentiality services. IPsec provides mutual authentication services between clients and hosts, regardless of whether AH or ESP is being used.

### Using AH

IPsec AH provides authentication, integrity, and anti-replay protection for the entire packet, including the IP header and the payload. AH does not provide confidentiality. When packets are secured with AH, the IPsec driver computes an Integrity Check Value (ICV) after the packet has been constructed but before it is sent to the

computer. In the Windows operating system, you can use either the Hash-Based Message Authentication Code (HMAC) SHA1 or HMAC MD5 algorithm to compute the ICV. Figure 10-4 shows how AH modifies an IP packet.



**Figure 10-4** AH modifications to an IP packet

The fields in an AH packet include these:

- **Next Header** Indicates the protocol ID for the header that follows the AH header. For example, if the encrypted data is transmitted using TCP, the next header value would be 6, which is the protocol ID for TCP.
- **Length** Contains the total length of the AH.
- **Security Parameters Index (SPI)** Identifies the security association (the IPSec agreement between two computers) that was negotiated in the Internet Key Exchange (IKE) protocol exchange between the source computer and the destination computer.
- **Sequence Number** Protects the AH-protected packet from replay attacks in which an attacker attempts to resend a packet that he has previously intercepted, such as an authentication packet, to another computer. For each packet issued for a specific security association (SA), the sequence number is incremented by 1 to ensure that each packet is assigned a unique sequence number. The recipient computer verifies each packet to ensure that a sequence number has not been reused. The sequence number prevents an attacker from capturing packets, modifying them, and then retransmitting them later.
- **Authentication Data** Contains the ICV created against the signed portion of the AH packet by using either HMAC SHA1 or HMAC MD5. The recipient performs the same integrity algorithm and compares the result of the hash algorithm with the result stored within the Authentication Data field to ensure that the signed portion of the AH packet has not been altered in transit. Because the TTL, Type of Service (TOS), Flags, Fragment Offset, and Header Checksum fields are not used in the ICV, packets secured with IPSec AH can cross routers, which can change these fields.

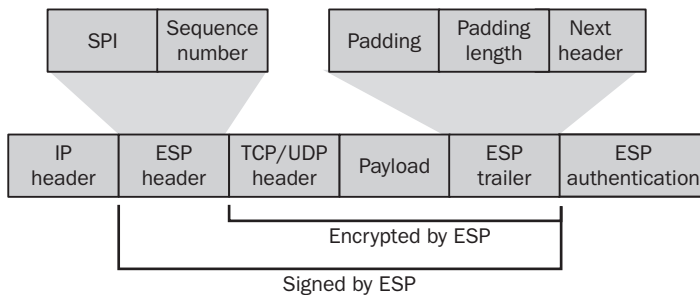
## Using ESP

ESP packets are used to provide encryption services to transmitted data. In addition, ESP provides authentication, integrity, and anti-replay services. When packets are sent using ESP, the payload of the packet is encrypted and authenticated. The encryption is done with either Data Encryption Standard (DES) or 3DES, and the ICV calculation is done with either HMAC SHA1 or HMAC MD5.



**Tip** When designing an IPSec solution, you can combine AH and ESP protocols in a single IPSec SA. Although both AH and ESP provide integrity protection for transmitted data, AH protects the entire packet from modification, whereas ESP protects only the IP payload from modification.

ESP encrypts the TCP or UDP header and the application data included within an IP packet. It does not include the original IP header unless IPSec tunnel mode is used. Figure 10-5 shows how ESP modifies an IP packet.



**Figure 10-5** ESP modifications to an IP packet

The ESP header has two fields that are inserted between the original IP header and the TCP or UDP header from the original packet:

- **Security Parameters Index (SPI)** Identifies the SA that was negotiated between the source computer and the destination computer for IPSec communication. The combination of the SPI, the IPSec protocol (AH or ESP), and the source and destination IP addresses identifies the SA used for the IPSec transmission within the ESP packet.
- **Sequence Number** Protects the ESP-protected packet from replay attacks. This field is incremented by 1 to ensure that packets are never received more than once. If a packet is received with a sequence number that's already been used, that packet is dropped.

The ESP trailer is inserted after the application data from the original packet and includes the following fields:

- **Padding** A variable length from 0 to 255 bytes that brings the length of the application data and ESP trailer to a length divisible by 32 bits so that they match the required size for the cipher algorithm.
- **Padding Length** Indicates the length of the Padding field. After the packet is decrypted, this field is used to determine the length of the Padding field.
- **Next Header** Identifies the protocol used for the transmission of the data, such as TCP or UDP.

Following the ESP trailer, the ESP protocol adds an ESP authentication trailer to the end of the packet. The ESP authentication trailer contains a single field:

- **Authentication Data** Contains the ICV, which verifies the originating host that sent the message and ensures that the packet was not modified in transit. The ICV uses the defined integrity algorithm to calculate the ICV. The integrity algorithm is applied to the ESP header, the TCP/UDP header, the application data, and the ESP trailer. Because the ICV does not include the IP header, ESP packets can cross routers.

ESP provides integrity protection for the ESP header, the TCP/UDP header, the application data, and the ESP trailer. ESP also provides inspection protection by encrypting the TCP/UDP header, the application data, and the ESP trailer.

## Choosing Between IPSec Modes

IPSec operates in two modes: transport mode and tunnel mode. IPSec transport mode is used for host-to-host connections, and IPSec tunnel mode is used for network-to-network or host-to-network connections.

### Using IPSec Transport Mode

IPSec transport mode is fully routable, as long as the connection does not cross a network address translation (NAT) interface, which would invalidate the ICV. Used this way, IPSec must be supported on both hosts, and each host must support the same authentication protocols and have compatible IPSec filters configured and assigned. IPSec transport mode is used to secure traffic from clients to hosts for connections where sensitive data is passed.

## Using IPSec Tunnel Mode

IPSec tunnel mode is used for network-to-network connections (IPSec tunnels between routers) or host-to-network connections (IPSec tunnels between a host and a router). Used this way, IPSec must be supported on both endpoints, and each endpoint must support the same authentication protocols and have compatible IPSec filters configured and assigned. IPSec tunnel mode is commonly used for site-to-site connections that cross public networks, such as the Internet.

## Selecting an IPSec Authentication Method

During the initial construction of the IPSec session—also known as the Internet Key Exchange, or IKE—each host or endpoint authenticates the other host or endpoint. When configuring IPSec, you must ensure that each host or endpoint supports the same authentication methods. IPSec supports three authentication methods:

- Kerberos
- X.509 certificates
- Preshared key



**Tip** Because IPSec requires mutual authentication, it also can be used to control network access to computers on your network that store high-value assets. For more information on how you can use IPSec this way, see the white paper “Using Microsoft Windows IPSec to Help Secure an Internal Corporate Network Server” on the Microsoft Web Site at <http://www.microsoft.com/downloads/details.aspx?FamilyID=a774012a-ac25-4a1d-8851-b7a09e3f1dc9&displaylang=en>.

## Authenticating with Kerberos

Kerberos is used for IPSec mutual authentication by default. For Kerberos to be used as the authentication protocol, both hosts in transport mode or both endpoints in tunnel mode must receive Kerberos tickets from the same Active Directory forest. Thus, you should choose Kerberos for IPSec authentication only when both hosts in transport mode or both endpoints in tunnel mode are within your own organization. Kerberos is an excellent authentication method for IPSec because it requires no additional configuration or network infrastructure.



**Important** Some types of traffic are exempted by default from being secured by IPSec, even when the IPSec policy specifies that all IP traffic should be secured. The IPSec exemptions apply to Broadcast, Multicast, Resource Reservation Setup Protocol (RSVP), IKE, and Kerberos traffic. Kerberos, a security protocol itself, can be used by IPSec for IKE authentication.



**Important** To remove the exemption for Kerberos and RSVP, set the value *NoDefaultExempt* to 1 in the registry key HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\IPSEC.

## Authenticating with X.509 Certificates

You can use X.509 certificates for IPsec mutual authentication of hosts or endpoints. Certificates enable you to create IPsec-secured sessions with hosts or endpoints outside your Active Directory forests, such as with business partners in extranet scenarios. You also must use certificates when using IPsec to secure virtual private network (VPN) connections made by using Layer Two Tunneling Protocol (L2TP). To use certificates, the hosts must be able to check that the other's certificate is valid.

## Authenticating with Preshared Key

You can use a preshared key, which is a simple, case-sensitive text string, to authenticate hosts or endpoints. Preshared key authentication should be used only when testing or troubleshooting IPsec connectivity because the preshared key is not stored in a secure fashion by hosts or endpoints.

## Creating IPsec Policies

IPsec is a policy-driven technology. In Windows Server 2003, Windows 2000, and Windows XP, you can have only one IPsec policy assigned at a time. IPsec policies are dynamic, meaning you do not have to stop and start the IPsec service or restart the computer when assigning or unassigning IPsec policies. You can also use Group Policy to deploy IPsec policies to clients running Windows Server 2003, Windows 2000, and Windows XP. The Windows operating system includes three precreated IPsec policies:

- **Client (Respond Only)** A computer configured with the Client policy will use IPsec if the host it is communicating with requests using IPsec and supports Kerberos authentication.
- **Server (Request Security)** A computer configured with the Server policy will always attempt to negotiate IPsec but will permit unsecured communication with hosts that do not support IPsec. The Server policy permits unsecured ICMP traffic.
- **Secure Server (Require Security)** A computer configured with the Secure Server policy will request that IPsec be used for all inbound and outbound connections. The computer will accept unencrypted packets but will always respond by using IPsec-secured packets. The Secure Server policy permits unsecured ICMP traffic.

In addition to the precreated policies, you can create custom IPSec policies. When creating your own IPSec policies, you must configure rules that include the following settings:

- IP filter list
- Tunnel settings
- Filter actions
- Authentication methods
- Connection types

IPSec rules determine which types of network traffic will initiate IPSec between the computer and the host or endpoint it is communicating with. A computer can have any number of IPSec rules. You should ensure that only one rule is created for each type of traffic. If multiple rules apply to a given type of traffic, the most specific rule will be processed first.

## IP Filter List

The IP filter list defines the types of network traffic to which the IPSec rule applies. You must define the following details for each entry in the filter list:

- **Source address** Can be a specific IP address, a specific IP subnet address, or any address. Windows Server 2003 adds the ability to use logical addresses in filters for greater flexibility.
- **Destination address** Can be a specific IP address, a specific IP subnet address, or any address.
- **Protocol** The protocol ID or transport protocol used by the protocol. For example, Point-to-Point Tunneling Protocol (PPTP) uses Generic Routing Encapsulation (GRE) packets. GRE packets are identified by their protocol ID, which is protocol ID 47. Telnet, on the other hand, uses TCP as its transport protocol, so an IPSec filter for Telnet would define the protocol type only as TCP.
- **Source port** If the protocol were to use TCP or UDP, the source port could be defined for the protected connection. The source port is set to a specific port or to a random port, depending on the protocol being defined. Most protocols use a random port for the source port.
- **Destination port** If the protocol uses TCP or UDP, the protocol uses a specific port at the server to accept transmissions. For example, Telnet configures the server to listen for connections on TCP port 23.

When configuring IP filter lists for transport mode connections, you should always choose to have the IPSec rule mirrored to secure the return communication defined in the rule. For tunnel mode connections, you must manually specify both the inbound and outbound filter list.

## Tunnel Settings

The tunnel setting determines whether IPSec operates in transport or tunnel mode. If you want IPSec to operate in transport mode, select This Rule Does Not Specify A Tunnel when creating an IPSec rule using the Security Rule Wizard. If you want the filter to operate in tunnel mode, you must specify the IP address of the endpoint of the tunnel.

## Filter Actions

For each filter rule, you must choose a filter action. The filter action defines how the traffic defined in the IP filter will be handled by the filter rule. The three filter actions are listed here and are shown in Figure 10-6.

- **Permit** Allows packets to be transmitted without IPSec protection. For example, Simple Network Management Protocol (SNMP) includes support for devices that might not be IPSec aware. Enabling IPSec for SNMP would cause a loss of network management capabilities for these devices. In a highly secure network, you could create an IPSec filter for SNMP and set the IPSec action to Permit to allow SNMP packets to be transmitted without IPSec protection. Packets that are permitted are not subject to IPSec authentication.
- **Block** Discards packets. If the associated IPSec filter is matched, all packets with the block action defined are discarded. Packets that are blocked are not subject to IPSec authentication.
- **Negotiate Security** Allows an administrator to define the desired encryption and integrity algorithms to secure data transmissions if an IPSec filter is matched.

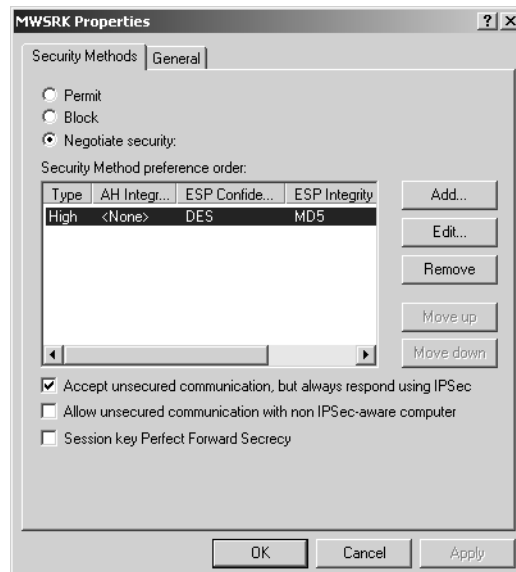


Figure 10-6 IPSec filter actions

In addition to these three basic actions, you can define settings that indicate how the computer will react if non-IPSec-protected data is received and how frequently new session keys are defined to protect the IPSec data. Options include the following:

- **Accept Unsecured Communication, But Always Respond Using IPSec** You use this option when the IPSec protection is enforced only at the servers, not at the clients. In a typical IPSec deployment, clients are configured to use IPSec if requested by the server but to never initiate an IPSec SA. This setting allows the initial packet to be received by the server, which then starts the IKE process to negotiate an SA between the client and the server. Although it is riskier to have the initial packet of a data transmission accepted by using plaintext, the response packet sent from the server will not be transmitted until an SA is established.
- **Allow Unsecured Communication With Non IPSec-Aware Computers** In a mixed network, this option allows non-IPSec-aware clients to connect to the server. Clients running Windows Server 2003, Windows 2000, and Windows XP, if configured to do so, will connect to the server and negotiate IPSec protection. Non-IPSec-aware clients will still be allowed to connect by using unprotected data streams.
- **Session Key Perfect Forward Secrecy** Using Perfect Forward Secrecy ensures that an existing key is never used as the foundation of a new key. When you use Perfect Forward Secrecy, all keys are generated without using existing keys. This reduces the risk of continual data exposure should a key be compromised because previous keys cannot be used to determine future keys.

## Authentication Methods

For each filter rule, you must choose an authentication method. You can enable multiple authentication methods for each rule and determine their order of precedence by editing the filter rule after it has been created.

## Connection Types

You must specify to which type of interfaces each filter rule applies. In Windows 2000 and Windows XP, you can choose to have the rule apply to the following:

- All network connections
- Local area network (LAN) connections
- Remote access connections



**Note** You can create IPsec policies by using the command line or from batch files and scripts in addition to using the user interface. Each operating system has introduced a new tool for doing this: for Windows 2000 you can use IPsecpol.exe, for Windows XP you can use IPseccmd.exe, and for Windows Server 2003 you can use IPseccmd.exe or Netsh.

## How IPsec Works

IPsec can be initiated by either the sending host or the receiving host. The two hosts or endpoints enter into a negotiation that will determine how the communication will be protected. The negotiation is completed in the IKE, and the resulting agreement is a set of security associations, or SAs.

IKE has two modes of operation, main mode and quick mode. We will examine each mode momentarily. IKE also serves two functions:

- Centralizes SA management, reducing connection time
- Generates and manages the authenticated keys used to secure the information

The SA is used until the two hosts or endpoints cease communication, even though the keys used might change. A computer can have many SAs. The SA for each packet is tracked using the Security Parameters Index (SPI).

### Main Mode

During the main mode negotiation, the two computers establish a secure, authenticated channel—the main mode SA. IKE automatically provides the necessary identity protection during this exchange. This ensures no identity information is sent without encryption between the communicating computers, thus enabling total privacy. Following are the steps in a main mode negotiation:

1. **Policy negotiation** These four mandatory parameters are negotiated as part of the main mode SA:
  - The encryption algorithm (DES or 3DES)
  - The hash algorithm (MD5 or SHA1)
  - The authentication method (certificate, preshared key, or Kerberos v5 authentication)
  - The Diffie-Hellman (DH) group to be used for the base keying material

If certificates or preshared keys are used for authentication, the computer identity is protected. However, if Kerberos v5 authentication is used, the computer identity is unencrypted until encryption of the entire identity payload takes place during authentication.

2. **DH exchange (of public values)** At no time are actual keys exchanged; only the base information needed by DH to generate the shared, secret key is exchanged. After this exchange, the IKE service on each computer generates the master key used to protect the final step: authentication.
3. **Authentication** The computers attempt to authenticate the DH exchange. Without successful authentication, communication cannot proceed. The master key is used, in conjunction with the negotiation algorithms and methods, to authenticate identities. The entire identity payload—including the identity type, port, and protocol—is hashed and encrypted by using the keys generated from the DH exchange in the second step. The identity payload, regardless of which authentication method is used, is protected from both modification and interpretation.

After the hosts have mutually authenticated each other, the host that initiated the negotiation presents an offer for a potential SA to the receiving host. The responder cannot modify the offer. Should the offer be modified, the initiator rejects the responder's message. The responder sends either a reply accepting the offer or a reply with alternatives. After the hosts agree on an SA, quick mode negotiation begins.

## Quick Mode

In this mode, SAs are negotiated on behalf of the IPSec service. The following are the steps in quick mode negotiation:

1. **Policy negotiation** The IPSec computers exchange their requirements for securing the data transfer:
  - ❑ The hash algorithm for integrity and authentication (MD5 or SHA1)
  - ❑ The algorithm for encryption, if requested (3DES or DES)
  - ❑ A description of the traffic to protect
2. **Session key material refresh or exchange** IKE refreshes the keying material, and new, shared, or secret keys are generated for authentication and encryption (if negotiated) of the packets. If a rekey is required, a second DH exchange takes place or a refresh of the original DH exchange occurs.
3. **SA exchange** The SAs and keys are passed to the IPSec driver, along with the SPI. A common agreement is reached, and two SAs are established: one for inbound communication, and one for outbound communication.

During the quick mode negotiation of shared policy and keying material, the information is protected by the SA negotiated during main mode. As mentioned in step 3, quick mode results in a pair of SAs: one for inbound communication and one for

outbound communication, each having its own SPI and key. Figure 10-7 shows a summary of what is negotiated during main mode and quick mode.

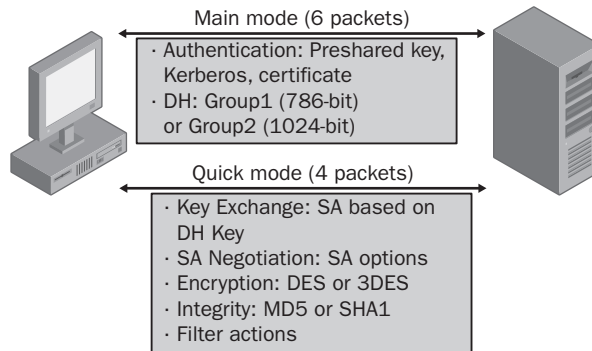


Figure 10-7 Main mode and quick mode negotiation

## IPSec, Routers, and NAT

IPSec creates a new IP header for a packet that can be routed as normal IP traffic. Routers and switches in the data path between the communicating hosts simply forward the packets to their destination. However, when a firewall or gateway lies in the data path, you must create firewall rules that allow traffic on the following IP protocols and UDP ports:

- **IP protocol ID 50** Create inbound and outbound filters to allow ESP traffic to be forwarded.
- **IP protocol ID 51** Create inbound and outbound filters to allow AH traffic to be forwarded.
- **UDP port 500** Create inbound and outbound filters to allow IKE traffic to be forwarded.

Because of the nature of the NAT and port address translation (PAT) technologies, which require that packets be altered to change IP address and port information, IPSec is not compatible with NAT. IPSec does not allow manipulation of packets during transfer. The IPSec endpoint will discard packets that have been altered by NAT because the ICVs will not match. Windows Server 2003 does allow IPSec ESP to pass NAT routers but encapsulates the IPSec-protected packet inside a UDP packet through a technology called NAT-T. NAT-T was added to Windows XP in Service Pack 2. Additionally, an L2TP/IPSec client that supports NAT-T for Windows 2000 can be downloaded from the Microsoft Web site at <http://support.microsoft.com/kb/818043>.

## Monitoring IPSec

You can monitor IPSec in Windows 2000 with IPsecmon.exe and in Windows Server 2003 and Windows XP using the IP Security Monitor Microsoft Management Console (MMC) snap-in. In addition, you can create log files in Windows Server 2003, Windows 2000, and Windows XP to view IPSec negotiations.

### Using IPsecmon in Windows 2000

In Windows 2000, you can view the status of IPSec SAs and basic information on IPSec sessions by running IPsecmon from the Run prompt. IPsecmon displays information about each SA and the overall statistics of IPSec and IKE sessions. Figure 10-8 shows IPsecmon in Windows 2000. The built-in Server IPSec policy is applied to the computer running Windows 2000 named SFOFS001. The SFOFS001 computer has attempted to negotiate IPSec with three other computers: SEADC001, SFODC001, and SFOXP001. However, SFOFS001 has successfully negotiated an SA with SFOXP001 only. The IPSec session with SFOXP001 uses the IPSec protocol ESP with 3DES as the encrypting algorithm and HMAC SHA1 as the authentication algorithm.

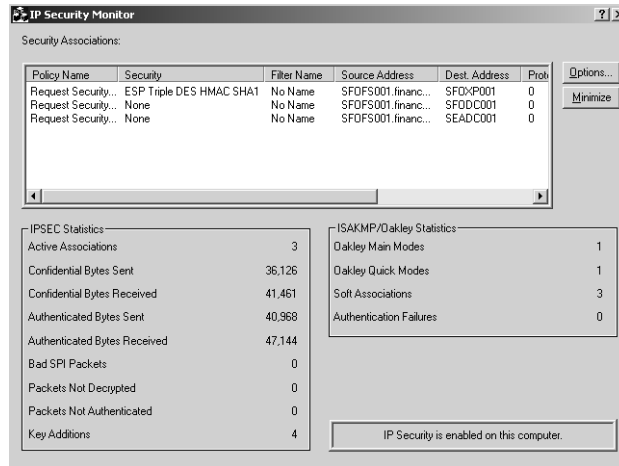


Figure 10-8 Using IPsecmon in Windows 2000

### Using the IP Security Monitor MMC Snap-In

In Windows Server 2003 and Windows XP, IPsecmon has been replaced with an MMC snap-in that provides all the information that IPsecmon did in Windows 2000, only in much greater detail. You can use the IP Security Monitor MMC snap-in to view

details of each SA, whereas in Windows 2000 you could view only the basic details of an SA. Figure 10-9 shows the IP Security Monitor MMC snap-in in Windows XP, which enables you to view the exact SA details negotiated during both main mode and quick mode.

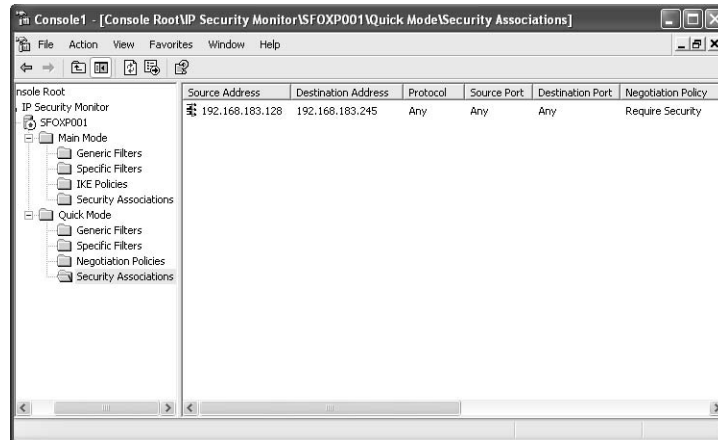


Figure 10-9 Using the IP Security Monitor MMC snap-in in Windows XP

## Using IPSec Logs

You can have IPSec log the IKE exchanges to a log file on the hard drive for troubleshooting or monitoring needs. To have your computer log IKE exchanges, you must create a registry value named *EnableLogging* in the registry key `HKLM\SYSTEM\CurrentControlSet\Services\PolicyAgent\Oakley`. To enable logging, set the value to 1 and restart the IPSec services. The log file will be written to the file `%systemroot%\debug\oakley.log`. If you use preshared keys for authentication, the key will appear in the log file in plaintext.



**Note** Although the IPSec log file will contain more detailed information than a network capture made with Network Monitor, you can also use Network Monitor to determine how IPSec SA negotiations function in relation to the other traffic on the network.

## Additional Changes to IPSec in Windows Server 2003

Several important changes were made to IPSec in Windows Server 2003. The changes can be grouped into three categories: management, security, and interoperability.

For improved management, the IP Security Monitor MMC snap-in that was first shipped with Windows XP has been added to the server platform to replace the IPsecmon.exe tool in Windows 2000. Windows Server 2003 also adds the ability to use logical addresses in addition to IP addresses. This can be useful when the IP addresses for computers change somewhat frequently, as can happen if computers get their addresses from a DHCP server. The most important manageability change to IPSec in Windows Server 2003 is the ability to configure IPSec through the network shell, Netsh. You can enter the network shell by typing **netsh** at the command line and can execute Netsh commands by placing them in batch files. In fact, several settings for IPSec can be set only through Netsh:

- Default exemption handling
- Strong certificate revocation list (CRL) checking
- IKE logging
- IPSec driver logging
- Persistent policies
- Startup exemptions

For example, you can set the default exemption level to allow multicast and broadcast traffic by typing the following string at the command prompt:

```
Netsh ipsec dynamic set config ipsecexempt value=2
```

You can set it back to its default value by typing:

```
Netsh ipsec dynamic set config ipsecexempt value=3
```

To improve the security of IPSec, several enhancements were made. The DH key length was increased to 2048 bits to better protect the secret key. To prevent an attacker from exploiting a system between the time the system is powered on and when the IPSec driver is loaded and running, Windows Server 2003 adds a computer startup policy. The computer startup policy, which can be configured through Netsh, is by default configured to allow DHCP traffic and the return of initiated outbound sessions through stateful packet filtering. Another feature

that is only manageable through Netsh is persistent policy. Persistent policy is always applied before and remains in effect regardless of whether IPsec policies are applied locally or by the Active Directory directory service. The other area that was greatly changed to improve security is the default exemption handling. In Windows 2000, five types of network traffic were exempted from IPsec. Table 10-8 contains the default exemptions in Windows 2000 and Windows Server 2003. By default, Windows Server 2003 IPsec default exemptions are set to 3. You can set default exemptions by setting the value for the registry key HKLM\SYSTEM\CurrentControlSet\Services\IPsec\NoDefaultExempt.

**Table 10-8 IPsec Default Exemptions in Windows Server 2003 and Windows 2000**

Value	0	1	2	3
Windows Server 2003	RSVP	IKE	RSVP	IKE
	IKE	Multicast	IKE	
	Kerberos	Broadcast	Kerberos	
	Multicast			
	Broadcast			
Windows 2000 and Windows XP	RSVP	IKE	Not available	Not available
	IKE	Multicast		
	Kerberos	Broadcast		
	Multicast			
	Broadcast			

For interoperability, Windows Server 2003 improves integration between IPsec and Network Load Balancing (NLB) and allows IPsec to be used across NAT and PAT routers through NAT traversal.

## Best Practices

- **Create a TCP/IP hardening policy.** Ensure that the TCP/IP stack on your computers that run Windows Server 2003, Windows 2000, and Windows XP is appropriately secure in regard to the threats to it. This is especially true of any computer directly connected to the Internet or in perimeter networks.
- **Use Windows Firewall for mobile and home computers running Windows XP.** Windows Firewall provides an excellent degree of protection for mobile clients and home computers. Be certain to provide training for users on how to enable and disable Windows Firewall.

- **Use IPSec to secure communications on corporate networks.** By using IPSec, you can increase the security for data transmission on your network as well as control network access to high-value servers.
- **Use IPSec hardware accelerators when possible.** By using IPSec hardware accelerators on computers that will have many IPSec sessions at a time, such as servers, you can prevent the computers' CPU performance from being overly taxed.

## Additional Information

- Internet Assigned Numbers Authority (IANA) TCP and UDP port number assignment list (<http://www.iana.org/assignments/port-numbers>)
- IANA IP protocol ID number list (<http://www.iana.org/assignments/protocol-numbers>)
- “5-Minute Security Advisor—Essential Security Tools for Home Office Users” (<http://www.microsoft.com/technet/columns/security/5min/5min-105.asp>)
- “Deploying Windows Firewall Settings for Microsoft Windows XP with Service Pack 2” white paper (<http://www.microsoft.com/downloads/details.aspx?FamilyID=4454e0e1-61fa-447a-bdcd-499f73a637d1&displaylang=en>)
- “IPSec Architecture” white paper (<http://www.microsoft.com/technet/itsolutions/network/security/ipsecarc.mspx>)
- “IPSec Implementation” white paper (<http://www.microsoft.com/technet/itsolutions/network/security/ipsecimp.mspx>)
- “Using Microsoft Windows IPSec to Help Secure an Internal Corporate Network Server” white paper (<http://www.microsoft.com/downloads/details.aspx?FamilyID=a774012a-ac25-4a1d-8851-b7a09e3f1dc9&displaylang=en>)
- “Improving Security with Domain Isolation: Microsoft IT Implements IP Security (IPSec)” white paper (<http://www.microsoft.com/technet/itsolutions/msit/security/ipsecdomisolwp.mspx>)
- “Security Considerations for Network Attacks” white paper (<http://www.microsoft.com/technet/security/topics/network/secdeny.mspx>)
- “Best Practices for Preventing DoS/Denial of Service Attacks” white paper (<http://www.microsoft.com/technet/security/bestprac/dosatack.asp>)
- Knowledge Base article 309798: “How to Configure TCP/IP Filtering in Windows 2000” (<http://support.microsoft.com/kb/309798>)
- Knowledge Base article 816792: “How to Configure TCP/IP Filtering in Windows Server 2003” (<http://support.microsoft.com/kb/816792>)