IPSec Technical Reference

Internet Protocol security (IPSec) in the Microsoft Windows Server 2003 operating system helps protect networks from active and passive attacks by securing IP packets through the use of packet filtering, cryptographic security services, and the enforcement of trusted communications. IPSec is integrated within the security framework of Windows Server 2003, which allows you to use Active Directory domains to provide identity and manage trust relationships between users and computers in different departments within an organization.

This subject provides a high-level description of IPSec that includes the scenarios for which it is intended and not intended.

In This Subject

What Is IPSec?

What Is IPSec?

In This Subject

Internet Protocol security (IPSec) is a framework of open standards for helping to ensure private, secure communications over Internet Protocol (IP) networks through the use of cryptographic security services. IPSec supports network-level data integrity, data confidentiality, data origin authentication, and replay protection. Because IPSec is integrated at the Internet layer (layer 3), it provides security for almost all protocols in the TCP/IP suite, and because IPSec is applied transparently to applications, there is no need to configure separate security for each application that uses TCP/IP.

IPSec helps provide defense-in-depth against:

- Network-based attacks from untrusted computers, attacks that can result in the denial-of-service of applications, services, or the network
- Data corruption
- Data theft
- User-credential theft
- Administrative control of servers, other computers, and the network.

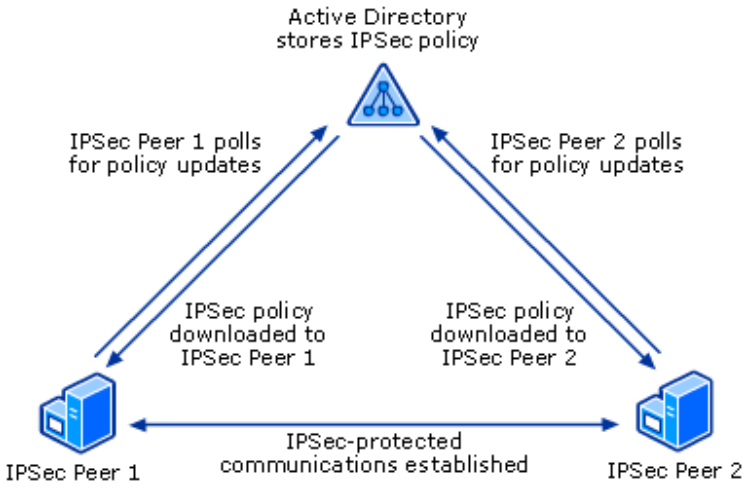
You can use IPSec to defend against network-based attacks through a combination of host-based IPSec packet filtering and the enforcement of trusted communications.

IPSec is integrated with the Windows Server 2003 operating system and it can use the Active Directory directory service as a trust model. You can use Group Policy to configure Active Directory domains, sites, and organizational units (OUs), and then assign IPSec policies as required to Group Policy objects (GPOs). In this way, IPSec policies can be implemented to meet the security requirements of many different types of organizations.

This section describes the solution that IPSec is intended to provide by providing information about core IPSec scenarios, IPSec dependencies, and related technologies.

The following figure shows an Active Directory-based IPSec policy being distributed to two IPSec peers and IPSec-protected communications being established between those two peers.

Two IPSec Peers Using Active Directory-based IPSec Policy



The MicrosoftWindows implementation of IPSec is based on standards developed by the Internet Engineering Task Force (IETF) IPSec working group. For a list of relevant IPSec RFCs, see the “Related Information” section later in this subject.

IPSec Scenarios

IPSec is a general-purpose security technology that can be used to help secure network traffic in many scenarios. However, you must balance the need for security with the complexity of configuring IPSec policies. Additionally, due to a lack of suitable standards, IPSec is not appropriate for some types of connectivity. This section describes IPSec scenarios that are recommended, IPSec scenarios that are not recommended, and IPSec scenarios that require special consideration.

Recommended Scenarios for IPSec

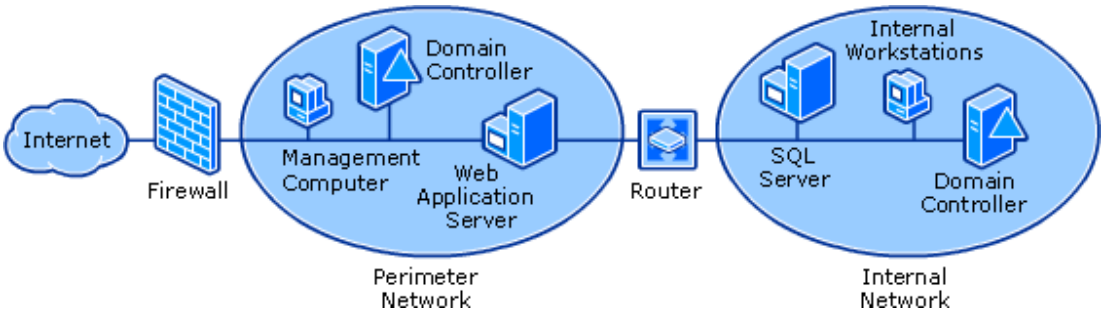
IPSec is recommended for the following scenarios:

- Packet filtering
- End-to-end security between specific hosts
- End-to-end traffic through a Microsoft Internet Security and Acceleration (ISA) Server-secured network address translator
- Secure server
- Layer Two Tunneling Protocol (L2TP) over IPSec (L2TP/IPSec) for remote access and site-to-site virtual private network (VPN) connections
- Site-to-site IPSec tunneling with non-Microsoft IPSec gateways

Packet Filtering

IPSec can perform host-based packet filtering to provide limited firewall capabilities for end systems. You can configure IPSec to permit or block specific types of unicast IP traffic based on source and destination address combinations and specific protocols and specific ports. For example, nearly all the systems illustrated in the following figure can benefit from packet filtering to restrict communication to only specific addresses and ports. You can strengthen security by using IPSec packet filtering to control exactly the type of communication that is allowed between systems.

Filtering Packets by Using IPSec



As illustrated in this figure:

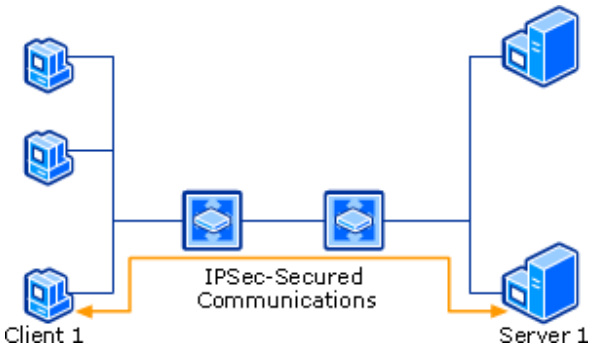
- The internal network domain administrator can assign an Active Directory-based IPSec policy (a collection of security settings that determines IPSec behavior) to block all traffic from the perimeter network (also known as a demilitarized zone [DMZ], demilitarized zone, or screened subnet).
- The perimeter network domain administrator can assign an Active Directory-based IPSec policy to block all traffic to the internal network.
- The administrator of the computer running Microsoft SQL Server on the internal network can create an exception in the Active Directory-based IPSec policy to permit structured query language (SQL) protocol traffic to the Web application server on the perimeter network.
- The administrator of the Web application server on the perimeter network can create an exception in the Active Directory-based policy to permit SQL traffic to the computer running SQL Server on the internal network.
- The administrator of the Web application server on the perimeter network can also block all traffic from the Internet, except requests to TCP port 80 for the HyperText Transfer Protocol (HTTP) and TCP port 443 for HTTPS (HTTP over Secure Sockets Layer/Transport Layer Protocol [SSL/TLS]), which are used by Web services. This provides additional security for traffic allowed from the Internet in case the firewall was misconfigured or compromised by an attacker.
- The domain administrator can block all traffic to the management computer, but allow traffic to the perimeter network.

You can also use IPSec with the IP packet-filtering capability or NAT/Basic Firewall component of the Routing and Remote Access service to permit or block inbound or outbound traffic, or you can use IPSec with the Internet Connection Firewall (ICF) component of Network Connections, which provides stateful packet filtering. However, to ensure proper Internet Key Exchange (IKE) management of IPSec security associations (SAs), you must configure ICF to permit UDP port 500 and port 4500 traffic needed for IKE messages.

End-to-End Security Between Specific Hosts

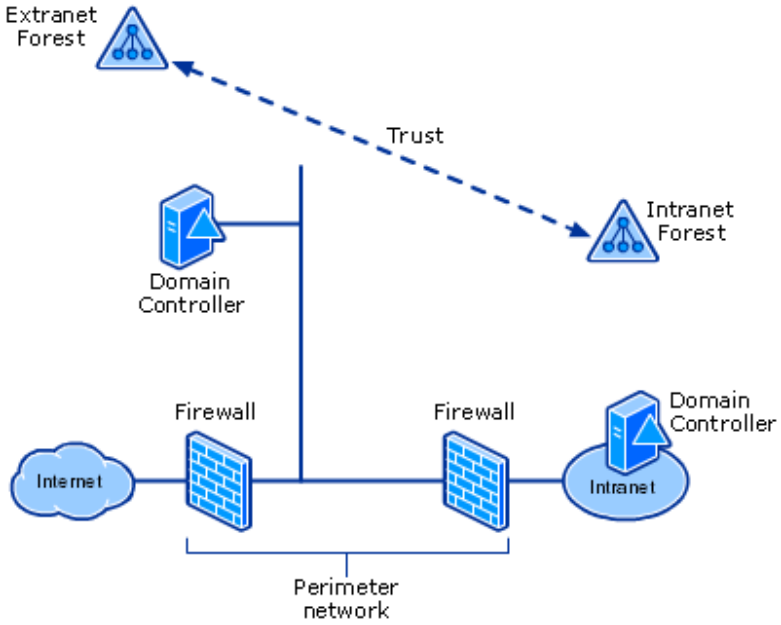
IPSec establishes trust and security from a unicast source IP address to a unicast destination IP address (end-to-end). For example, IPSec can help secure traffic between Web servers and database servers or domain controllers in different sites. As shown in the following figure, only the sending and receiving computers need to be aware of IPSec. Each computer handles security at its respective end and assumes that the medium over which the communication takes place is not secure. The two computers can be located near each other, as on a single network segment, or across the Internet. Computers or network elements that route data from source to destination are not required to support IPSec.

Securing Communications Between a Client and a Server by Using IPSec



The following figure shows domain controllers in two forests that are deployed on opposite sides of a firewall. In addition to using IPSec to help secure all traffic between domain controllers in separate forests, as shown in the figure, you can use IPSec to help secure all traffic between two domain controllers in the same domain and between domain controllers in parent and child domains.

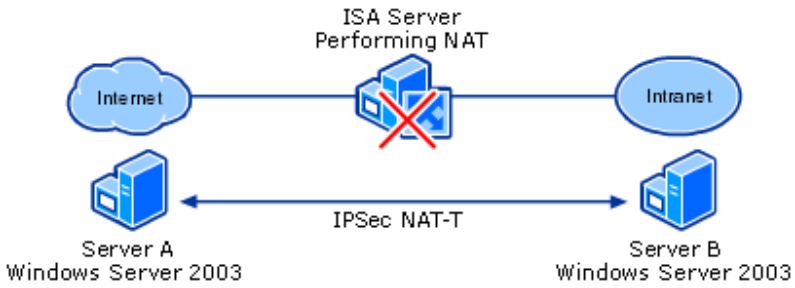
Securing Communications Between Two Domain Controllers in Different Forests by Using IPSec



End-to-End Traffic Through an ISA-Secured Network Address Translator

Windows Server 2003 supports IPSec NAT Traversal (NAT-T). IPSec NAT-T allows traffic to be secured by IPSec and also to be translated by a network address translator. For example, you can use IPSec transport mode to help secure host-to-host traffic through a computer that is running ISA Server and that is functioning as a network address translator if ISA (or any other NAT device) does not need to inspect the traffic between the two hosts. IPSec transport mode is used to protect traffic between hosts and it can provide security between computers that are on the same local area network (LAN) or connected by private wide area network (WAN) links. In the following figure, a computer running Windows Server 2003 and Microsoft Internet Security and Acceleration (ISA) Server is functioning as a network address translator. The IPSec policy on Server A is configured to secure traffic to the IP address of Server B, while the IPSec policy on Server B is configured to secure traffic to the external IP address of the computer running ISA Server.

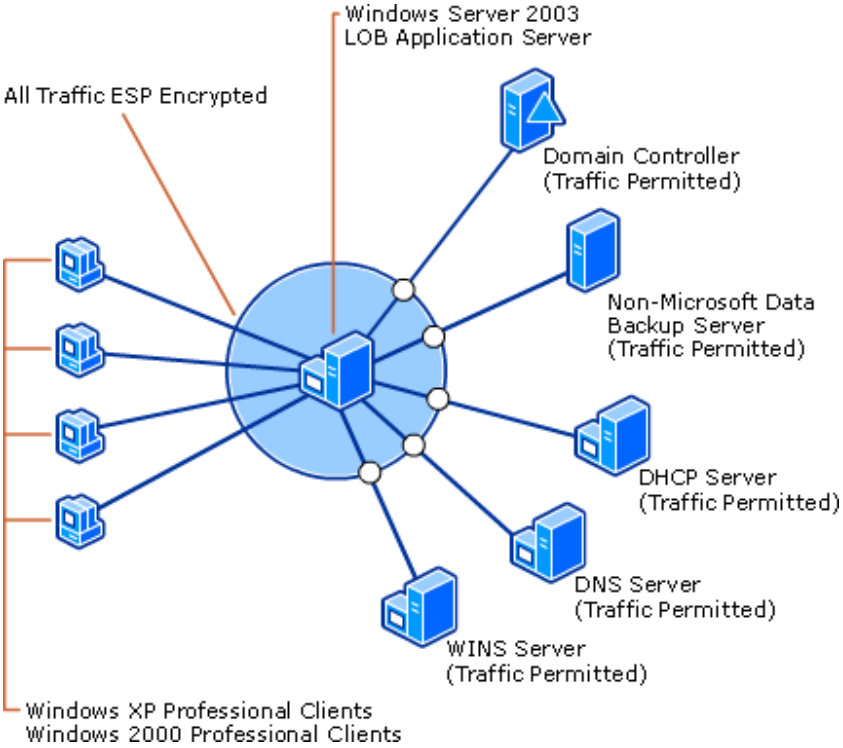
Securing Communications Through an ISA-Secured NAT by Using IPSec NAT-T



Secure Server

You can require IPSec protection for all client computers that access a server. In addition, you can set restrictions on which computers are allowed to connect to a server running Windows Server 2003. The following figure shows IPSec in transport mode securing a line of business (LOB) application server.

Securing an Application Server by Using IPSec



In this scenario, an application server in an internal corporate network must communicate with clients running Windows 2000 or Windows XP Professional; a Windows Internet Name Service (WINS) server, Domain Name System (DNS) server, and Dynamic Host Configuration Protocol (DHCP) server; Active Directory domain controllers; and a non-Microsoft data backup server. The users on the client computers are company employees who access the application server to view their personal payroll information and performance review scores.

Because the traffic between the clients and the application server involves highly sensitive data, and because the server should only communicate with other domain members, the network administrator uses an IPSec policy that requires ESP encryption and communication only with trusted computers in the Active Directory domain.

Other traffic is permitted as follows:

- Traffic between the WINS server, DNS server, DHCP server, and the application server is permitted because WINS servers, DNS servers, and DHCP servers must typically communicate with computers that run on a wide range of operating systems, some of which might not support IPSec.
- Traffic between Active Directory domain controllers and the application server is permitted, because using IPSec to secure communication between domain members and their domain controllers is not a recommended usage.
- Traffic between the non-Microsoft data backup server and the application server is permitted because the non-Microsoft backup server does not support IPSec.

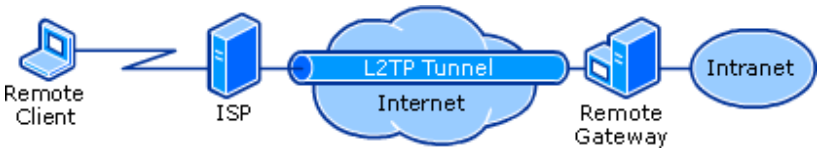
L2TP/IPSec for Remote Access and Site-to-Site VPN Connections
You can use L2TP/IPSec for all VPN scenarios. This does not require the configuration and deployment of IPSec policies. Two common scenarios for L2TP/IPSec are securing communications between remote access clients and the corporate network across the Internet and securing communications between branch offices.

Note

- Windows IPSec supports both IPSec transport mode and tunnel mode. Although VPN connections are commonly referred to as "tunnels," IPSec transport mode is used for L2TP/IPSec VPN connections. IPSec tunnel mode is most commonly used to help protect site-to-site traffic between networks, such as site-to-site networking through the Internet.

L2TP/IPSec for remote access connections
A common requirement for organizations is to secure communications between remote access clients and the corporate network across the Internet. Such a client might be a sales consultant who spends most of the time traveling, or an employee working from a home office. In the following figure, the remote gateway is a server that provides edge security for the corporate intranet. The remote client represents a roaming user who requires regular access to network resources and information. An ISP is used as an example to demonstrate the path of communication when the client uses an ISP to access the Internet. L2TP/IPSec provides a simple, efficient way to build a VPN tunnel and help protect the data across the Internet.

Securing Remote Access Clients by Using L2TP/IPSec



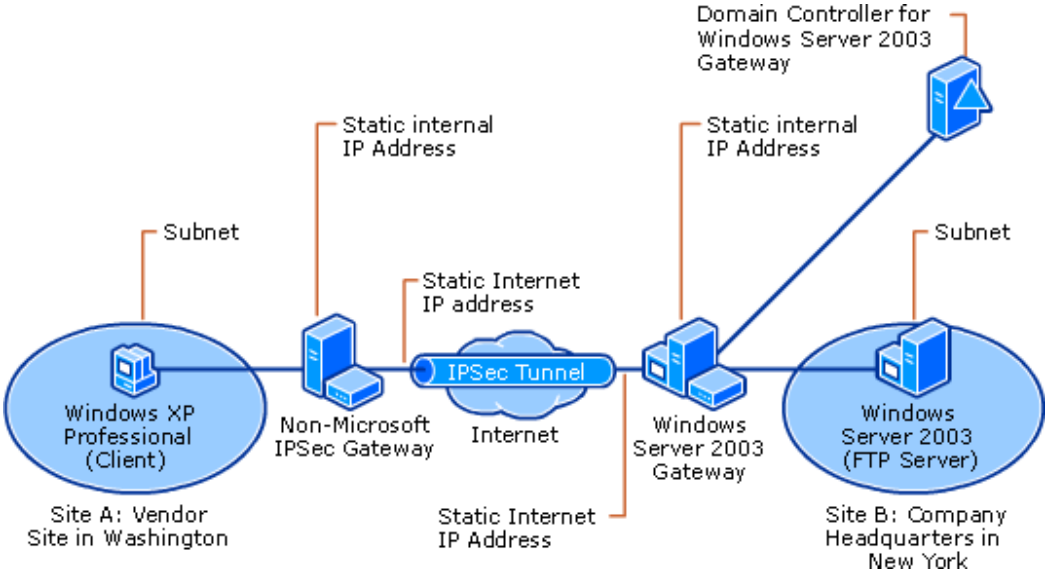
L2TP/IPSec for site-to-site VPN connections
A large corporation often has multiple sites that require communication — for example, a corporate office in New York and a sales office in Washington. In this case, L2TP/IPSec provides the VPN connection and helps protect the data between the sites. In the following figure, the router running Windows Server 2003 provides edge security. The routers might have a leased line, dial-up, or other type of Internet connection. The L2TP/IPSec VPN tunnel runs between the routers only and provides protected communication across the Internet.

Establishing an L2TP/IPSec VPN Tunnel Between Sites

Site-to-Site IPSec Tunneling with Non-Microsoft Gateways

For interoperability with gateways or end systems that do not support L2TP/IPSec or Point-to-Point Tunneling Protocol (PPTP) VPN site-to-site connections, you can use IPSec in tunnel mode. When IPSec tunnel mode is used, the sending gateway encapsulates the entire IP datagram by creating a new IP packet that is then protected by one of the IPSec protocols. The following figure illustrates site-to-site IPSec tunneling.

Establishing an IPSec Gateway-to-Gateway Tunnel Between Sites



In this figure, traffic is being sent between a client computer in a vendor site (Site A) and a File Transfer Protocol (FTP) server at the corporate headquarters site (Site B). Although an FTP server is used for this scenario, the traffic can be any unicast IP traffic. The vendor uses a non-Microsoft IPSec-enabled gateway, while corporate headquarters uses a gateway running Windows Server 2003. An IPSec tunnel is used to secure traffic between the non-Microsoft gateway and the gateway running Windows Server 2003.

Scenarios for Which IPSec Is Not Recommended

IPSec policies can be quite complex to configure and manage. Additionally, IPSec can incur performance overhead to establish and maintain secure connections, and it can incur network latency. In some deployment scenarios, the lack of standard methods for user authentication and address assignment make IPSec an unsuitable choice. Because IPSec depends on IP addresses for establishing secure connections, you cannot specify dynamic IP addresses. It is often necessary for a server to have a static IP address in IPSec policy filters. In large network deployments and in some mobile user cases, using dynamic IP addresses at both ends of the connection can increase the complexity of IPSec policy design. For these reasons, IPSec is not recommended for the following scenarios:

- Securing communication between domain members and their domain controllers
- Securing all traffic in a network
- Securing traffic for remote access VPN connections using IPSec tunnel mode

Securing Communication Between Domain Members and their Domain Controllers
Using IPSec to help secure traffic between domain members (either clients or servers) and their domain controllers is not recommended because:

- If domain members were to use IPSec-secured communication with domain controllers, increased latency might occur, causing authentication and the process of locating a domain controller to fail.
- Complex IPSec policy configuration and management is required.
- Increased load is placed on the domain controller CPU to maintain SAs with all domain members. Depending on the number of domain members in the domain controller's domain, such a load might overburden the domain controller.

Securing All Traffic in a Network

In addition to reduced network performance, using IPSec to help secure all traffic in a network is not recommended because:

- IPSec cannot secure multicast and broadcast traffic.
- Traffic from real-time communications, applications that require Internet Control Message Protocol (ICMP), and peer-to-peer applications might be incompatible with IPSec.
- Network management functions that must inspect the TCP, UDP, and other protocol headers are less effective, or cannot function at all, due to IPSec encapsulation or encryption of IP payloads.

Securing Traffic for Remote Access VPN Connections by Using IPSec Tunnel Mode
IPSec tunnel mode is not a recommended technology for remote access VPN connections, because there are no standard methods for user authentication, IP address assignment, and name server address assignment. Using IPSec tunnel mode for gateway-to-gateway VPN connections is possible using computers running Windows Server 2003. But because the IPSec tunnel is not represented as a logical interface over which packets can be forwarded and received, routes cannot be assigned to use the IPSec tunnel and routing protocols do not operate over IPSec tunnels. Therefore, the use of IPSec tunnel mode is only recommended as a VPN solution for site-to-site VPN connections in which one end of the tunnel is a non-Microsoft VPN server or security gateway that does not support L2TP/IPSec. Instead, use L2TP/IPSec or PPTP for remote access VPN connections.

IPSec Uses That Require Special Considerations
The following scenarios merit special consideration, because they introduce an additional level of complexity for IPSec policy configuration and management:

- Securing traffic over IEEE 802.11wireless networks
- Securing traffic in home networking scenarios
- Securing traffic in environments that use dynamic IP addresses

Securing Traffic Sent over 802.11 Networks
You can use IPSec transport mode to protect traffic sent over 802.11 wireless networks. However, IPSec is not the recommended solution for providing security for corporate 802.11 wireless LAN networks. Instead, it is recommended that you use either 802.11 Wired Equivalent Privacy (WEP) encryption or Wi-Fi Protected Access (WPA) and IEEE 802.1X authentication.

To use IPSec to help secure traffic sent over 802.11 networks, you must ensure that client computers and servers support IPSec. Configuration management and trust are also required on client computers and servers when IPSec is used. Because many computers on a network do not support IPSec or are not managed, it is not appropriate to use IPSec alone to protect all 802.11 corporate wireless LAN traffic.

Securing Traffic in Home Networking Scenarios
Although IPSec is not optimized for use in general home networking scenarios, when network security administrators deploy IPSec with appropriate scripts and support tools, it can be used effectively on home computers for specific scenarios.

IPSec can be used to connect home computers to a corporate intranet for remote access. Network security administrators can use scripts and support tools to deploy IPSec on the home computers of employees who require secure connectivity to the corporate network. For example, an administrator can use a Connection Manager profile to deploy an L2TP/IPSec-based VPN connection on home computers. Employees can then establish IPSec-secured connections across the Internet to the corporate network by using the VPN client built-in to Network Connections.

Note

- In some cases, non-Microsoft VPN or firewall clients might disable the IPSec service, which is required for IPSec to function. If you encounter this problem, it is recommended that you contact the VPN or firewall vendor.

IPSec is not recommended for end users in general home networking scenarios for the following reasons:

- The IPSec policy configuration user interface (IP Security Policy Management) is intended for professional network security administrators, rather than for end users. Improper policy configuration can result in blocked communications, and if problems occur, built-in support tools are not yet available to aid end users in troubleshooting.

- Some home networking applications use broadcast and multicast traffic, for which IPSec cannot negotiate security.
- Many home networking scenarios use a wide range of dynamic IP addresses.
- Many home networking scenarios involve the use of a network address translator. To use IPSec across a NAT, both IPSec peers must support IPSec NAT-T.

Securing Traffic in Environments That Use Dynamic IP Addresses
IPSec depends on IP addresses for establishing secure connections, and it is often necessary for a server to have a static IP address in IPSec policy filters. In large network deployments and in some mobile user cases, using dynamic IP addresses at both ends of the connection can increase the complexity of IPSec policy design.

IPSec Dependencies

There is no single optimal environment for IPSec. However, there are dependencies that are critical to the successful deployment of IPSec. This section describes how the following two IPSec dependencies affect the deployment of IPSec:

- Active Directory (if your deployment requires the use of Active Directory-based IPSec policies, rather than local IPSec policies)
- Successful mutual authentication

Active Directory
For organizations with large numbers of computers that must be managed in a consistent way, it is best to distribute IPSec policies by using Group Policy to configure Active Directory domains, sites, and organizational units (OUs), and then assigning IPSec policies as required to Group Policy objects (GPOs). Although you can assign local IPSec policies to computers that are not members of a trusted domain, distributing IPSec policies and managing IPSec policy configuration and trust relationships is much more time-consuming for computers that are not members of a trusted domain.

If you do use Active Directory-based IPSec policies, IPSec policy design and management must take into account the delays that result from the replication of Group Policy data from domain controllers to domain members. Often, the first step in troubleshooting a problem with IPSec connectivity is to determine whether the computer in question has the most current Group Policy assignment. To do this, you must be a member of the local Administrators group on the computer for which troubleshooting is being performed.

Successful Mutual Authentication
For IPSec-secured communications to be established, there must be successful mutual authentication between IPSec peers. IPSec requires the use of one of the following authentication methods: Kerberos version 5, an X.509 version 3 computer certificate issued by a public key infrastructure (PKI), or a preshared key. The two IPSec peers must use at least one common authentication method or communication will fail. Make sure that you choose an authentication method that is appropriate for your environment.

When you deploy IPSec to negotiate security for upper-layer protocols such as TCP connections, failures to communicate are often caused by the failure of IPSec to mutually authenticate the two communication endpoints. Authentication might succeed for some computers and fail for others due to issues within the authentication system itself (typically, Kerberos or public key certificates, rather than preshared keys, because preshared keys are not recommended). For these reasons, it is important to evaluate how the dependency of IPSec connectivity on authentication affects your environment and support practices. Additional training is recommended so that administrators can quickly determine whether a connectivity problem is caused by an IPSec authentication failure and understand how to investigate the authentication system.

IPSec and ICF

IPSec is similar to the ICF feature of Network Connections. However, there are important differences between these two technologies as well. It is important to understand the similarities and differences, so that you can deploy IPSec where it is truly needed and obtain the maximum benefits from the security that IPSec provides. This section describes similarities and differences between IPSec and ICF.

Microsoft ICF is a locally managed, stateful host firewall that, by default, discards all incoming packets except those sent in response to packets sent by the host. One primary difference between IPSec and ICF is that IPSec provides complex static filtering based on IP addresses, while ICF provides stateful filtering for all addresses on a network interface. For example, when you use IPSec, you can configure a filter to block all inbound traffic that is sent to a specific IP address over a specific protocol and port (for example, “Block all inbound traffic from the

Internet to TCP port 135"). You can also configure exemptions to permit specific types of traffic from specific source IP addresses. When you use ICF, you can configure exemptions to permit traffic based solely on the port, regardless of the source IP address.

It is recommended that you use ICF when you want a firewall for a network interface that can be accessed through the Internet. It is recommended that you use IPSec when you want to secure traffic over upper-layer protocols or when you need to allow access only to a group of trusted computers. Note that it is easier to configure ICF to permit traffic over a certain port than it is to configure an IPSec policy.

IPSec is not a full-featured host firewall. However, it does provide the ability to centrally manage policies that can permit, block, or secure unicast IP traffic based on specific addresses, protocols, and ports. Some of the functions found in standard firewalls that IPSec does not provide include stateful inspection, application protocol awareness, intrusion detection, and packet logging. Although IPSec lacks some features of firewalls, the packet blocking and filtering it provides can be effective in helping to limit the spread of viruses and thwart specific attacks known to use specific ports. You can also use IPSec to prevent specific applications and services from being used on the network.

Related Information

The following resources contain additional information that is relevant to this section.

- Active Directory Collection
- How IPSec Policy Extension Works
- How TCP/IP Works
- How VPN Works
- How 802.11 Wireless Works

The following RFCs and Internet Drafts are relevant to IPSec. To find the RFCs, type the appropriate RFC number in the IETF RFC Database. To find the Internet Drafts, type the appropriate keyword in the IETF Internet Drafts database.

- RFC 2401: Security Architecture for the Internet Protocol
- RFC 2402: IP Authentication Header
- RFC 2406: IP Encapsulating Security Payload (ESP)
- RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP)
- RFC 2409: The Internet Key Exchange (IKE)
- UDP Encapsulation of IPSec Packets (draft-ietf-ipsec-udp-encaps-02.txt)
- Negotiation of NAT-Traversal in the IKE (draft-ietf-ipsec-nat-t-ike-02.txt)

How IPSec Works

How IPSec Works

In this section

- IPSec Architecture
- IPSec Protocols
- IPSec Processes and Interactions
- Network Ports and Protocols Used by IPSec
- Related Information

In the Microsoft Windows Server 2003 operating system, Internet Protocol security (IPSec) helps provide defense-in-depth against network-based attacks from untrusted computers. IPSec provides protection from attack in host-to-host, virtual private network (VPN), site-to-site (also known as gateway-to-gateway or router-to-router), and secure server environments. You can configure IPSec policies to meet the security requirements of a computer, an organizational unit, a domain, a site, or a global organization.

IPSec uses packet filtering and cryptography. Cryptography provides user authentication, ensures data confidentiality and integrity, and enforces trusted communication. The strong cryptographic-based authentication and encryption support that IPSec provides is especially effective for securing traffic that must traverse untrusted

network paths, such as those on a large corporate intranet or the Internet. IPSec also is especially effective for securing traffic that uses protocols and applications that do not provide sufficient security for communications.

To successfully deploy IPSec for Windows Server 2003, you must ensure the following:

- If your scenario requires Active Directory-based IPSec policy (a collection of IPSec rules that determine IPSec behavior), the Active Directory directory service and Group Policy must be configured correctly on the corporate network, appropriate trusts must be defined, and appropriate permissions must be applied. Although Group Policy applies to both users and computers, IPSec policy is a computer configuration Group Policy setting that applies only to computers.
- Each computer that will establish IPSec-secured communications must have an IPSec policy assigned. This policy must be compatible with the IPSec policy that is assigned to other computers with which that computer must communicate.
- Authentication must be configured correctly and an appropriate authentication method must be specified in the IPSec policy so that mutual authentication can occur between IPSec peers.
- Routers, firewalls, or other filtering devices must be configured correctly to permit IPSec protocol traffic on all parts of the corporate network, if IPSec negotiation messages and IPSec-secured traffic must pass through these devices.
- Computers must run operating systems that automatically support IPSec or must have appropriate client software installed.
- If computers are running different versions of the Microsoft Windows operating system (for example, Windows Server 2003, the Microsoft Windows XP operating system, and the Microsoft Windows 2000 operating system), you must address the compatibility of the IPSec policies.
- If clients must establish IPSec-secured connections with servers, those servers must be adequately sized to support those connections. If necessary, you can use IPSec hardware offload network adapters.
- The number of IPSec policies are kept to a minimum, and the IPSec policies are made as simple as possible.
- Systems administrators who will configure and support IPSec must be properly trained and must be members of the appropriate administrative groups.

IPSec Architecture

Several Requests for Comments (RFCs) define the architecture and components of IPSec. These components and their interrelationship comprise the logical architecture of IPSec. This section briefly describes the fundamental components of the IPSec logical architecture and then explains how these components are implemented in Windows Server 2003.

For comprehensive descriptions of IPSec architecture and components, see the RFCs that are listed in "Related Information" later in this section.

Logical Architecture

The IPSec architecture can be categorized into four main areas:

- Security Associations
- SA and key management support
- IPSec protocols
- Algorithms and methods

Security Associations

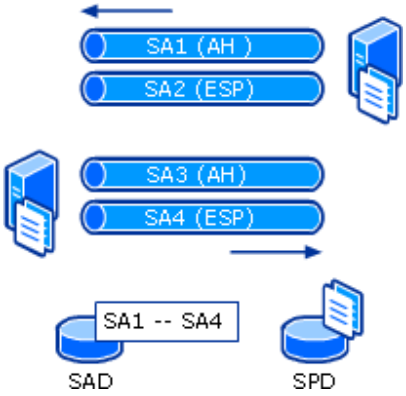
Security Associations (SAs) are a combination of a mutually agreeable policy and keys that defines the security services, mechanisms, and keys used to protect communications between IPSec peers. Each SA is a one-way or simplex connection that provides security services to the traffic that it carries.

Because SAs are defined only for one-way communication, each IPSec session requires two SAs. For example, if both IPSec protocols, Authentication Header (AH) and Encapsulating (ESP), are used for an IPSec session between two peers, then four SAs would be required.

SAs for IPSec-secured communications require two databases: a security policy database (SPD) and security association database (SAD). The SPD stores the security requirements or policy requisites for an SA to be established. It is used during both inbound and outbound packet processing. IPSec checks inbound packets to ensure that they have been secured according to policy. Outbound packets are secured according to policy.

The SAD contains the parameters of each active SA. The Internet Key Exchange (IKE) protocol automatically populates the SAD. After an SA is established, the information for each SA is stored in the SAD. The following figure shows the relationship between SAs, the SPD, and the SAD.

SA, SPD, and SAD Architecture

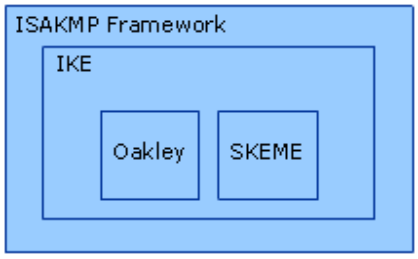


SA and Key Management

IPSec requires SA and key management support. The Internet Security Association and Key Management Protocol (ISAKMP) defines the framework for authentication and key exchange by providing procedures for negotiating, establishing, changing, and deleting SAs. It does not define the actual key exchange: it merely provides the framework.

IPSec requires support for both manual and automatic management of SAs and keys. IKE is the default automated key management protocol for IPSec. IKE is a hybrid protocol that incorporates parts of the Oakley key exchange protocol and the SKEME keying techniques protocol. The following figure shows the relationship between the ISAKMP, IKE, Oakley, and SKEME protocols.

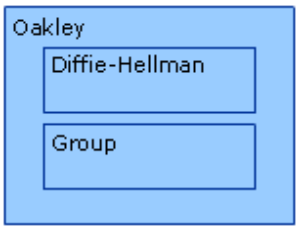
ISAKMP, IKE, Oakley, and SKEME Protocol Architecture



The Oakley protocol uses the Diffie-Hellman key exchange or key agreement algorithm to create a unique, shared, secret key, which is then used to generate keying material for authentication or encryption. For example, such a shared secret key could be used by the DES encryption algorithm for the required keying material. A Diffie-Hellman exchange can use one of a number of groups that define the length of the base prime numbers (key size) which are created for use during the key exchange process. The longer the number, the greater the key strength. Well-known groups include Groups 1, 2, and 14.

The following figure shows the relationship between the Oakley protocol, the Diffie-Hellman algorithm, and well-known Diffie-Hellman key exchange groups.

Oakley Protocol, Diffie-Hellman Key Exchange Algorithm, and Well-Known Diffie-Hellman Groups Architecture



The Oakley protocol defines several modes for the key exchange process. These modes correspond to the two negotiation phases defined in the ISAKMP protocol. For phase 1, the Oakley protocol defines two principle modes: main and aggressive. IPSec for Windows does not implement aggressive mode. For phase 2, the Oakley protocol defines a single mode, quick mode.

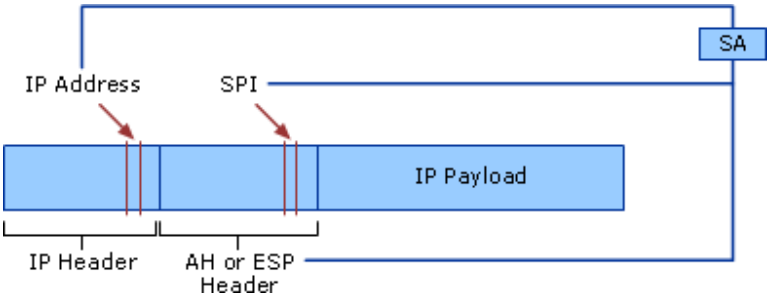
IPSec Protocols

To provide security for the IP layer, IPSec defines two protocols: Authentication Header (AH) and Encapsulating Security Payload (ESP). These protocols provide security services for the SA. Each SA is identified by the Security Parameters Index (SPI), IP destination address, and security protocol (AH or ESP) identifier.

The SPI is a unique, identifying value in an SA that is used to distinguish among multiple SAs on the receiving computer. For example, IPSec communication between two computers requires two SAs on each computer. One SA services inbound traffic and the other services outbound traffic. Because the addresses of the IPSec peers for the two SAs are the same, the SPI is used to distinguish between the inbound and outbound SA. Because the encryption keys differ for each SA, each SA must be uniquely identified.

The following figure shows the relationship between the SA, SPI, IP destination address, and security protocol.

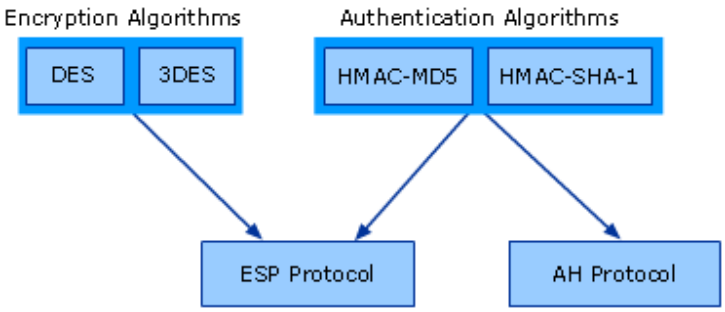
IPSec Protocols and SA Architecture



Algorithms and Methods

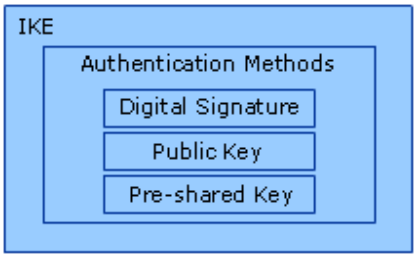
The IPSec protocols use authentication, encryption, and key exchange algorithms. Two authentication or keyed hash algorithms, HMAC-MD5 (Hash Message Authentication Code - MD5) and HMAC-SHA-1, are used with both the AH and ESP protocols, The DES and 3DES encryption algorithms are used with ESP. The following figure shows the relationship between the authentication and encryption algorithms and the AH and ESP protocols.

IPSec Protocols and Algorithms for Authentication and Encryption



The authentication methods for IPSec, as defined by the IKE protocol, are grouped into three categories: digital signature, public-key, and pre-shared key. The following figure shows the relationship between the IKE protocol and the authentication methods.

IKE Protocol and Authentication Methods Architecture

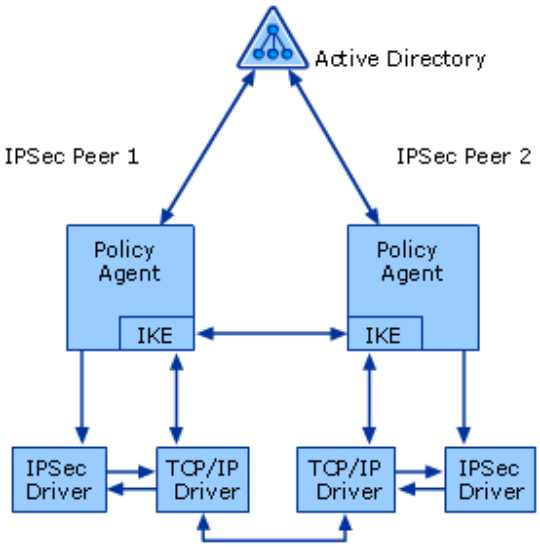


Windows Server 2003 IPSec Architecture and Components

The components and architecture of Windows Server 2003 IPSec are based on the IPSec RFCs. The basic IPSec architecture for Windows Server 2003 has the following components: Active Directory, a Policy Agent, the IKE protocol, an IPSec driver, and a TCP/IP driver.

The following figure illustrates how these components interact.

Windows Server 2003 IPSec Architecture



The following table describes each of these components.

IPSec Components

Component	Description
Active Directory	Windows Server 2003 Active Directory stores domain-wide IPSec policies for computers that are members of the domain. Active Directory-based IPSec policies are polled and retrieved by the Policy Agent.
Policy Agent	The Policy Agent retrieves IPSec policy from an Active Directory domain, a configured set of local policies, or a local cache. The Policy Agent then distributes authentication and security settings to the IKE component and the IP filters to the IPSec driver.

IKE	IKE receives authentication and security settings from the Policy Agent and waits for requests to negotiate IPSec SAs. When requested by the IPSec driver, IKE negotiates both kinds of SAs (main mode and quick mode) with the appropriate endpoint requested by the IPSec driver based on the policy settings obtained from the Policy Agent. After negotiating an IPSec SA, IKE sends the SA settings to the IPSec driver.
IPSec driver	The IPSec driver monitors and secures outbound unicast IP traffic and monitors, decrypts, and validates inbound unicast IP traffic. After the IPSec driver receives the filters from the Policy Agent, it determines which packets are permitted, which are blocked, or which are secured. For secure traffic, the IPSec driver either uses active SA settings to secure the traffic or requests that new SAs be created. The IPSec driver is bound to the TCP/IP driver to provide IPSec processing for IP packets that pass through the TCP/IP driver.
TCP/IP driver	The TCP/IP driver is the Windows Server 2003 implementation of the TCP/IP protocol. It is a kernel-mode component that is loaded from the tcpip.sys file during startup.

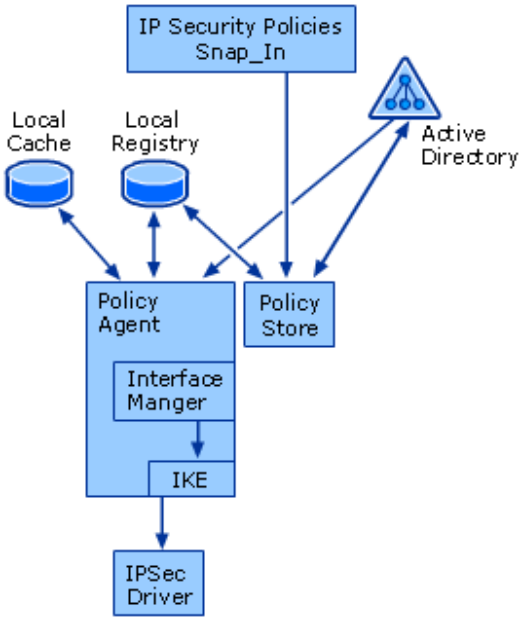
The architecture of the Policy Agent, IKE protocol, and IPSec driver are described in more detail in the following sections.

Policy Agent Architecture

The Policy Agent retrieves IPSec policy information, handles the internal interpretation and processing of the policy, and sends it to the other IPSec components that require the information to perform security services. The Policy Agent has the following components: policy store, Policy Agent service, local registry, local cache, and Interface Manager.

The following figure shows the architecture of the Policy Agent.

Policy Agent Architecture



The following table briefly describes the Policy Agent components.

Policy Agent Components

Component	Description
Policy store	The IPSec policy store maintains both IPSec policy descriptions and interfaces to applications and other tools that provide policy data management. The policy store accesses IPSec policy data that is stored in either the local registry or in Active Directory.

Policy Agent	The IPSec Policy Agent controls the retrieval and distribution of IPSec policy and maintains the data about the configured policy for the IPSec driver and IKE.
Local registry	The local registry stores the locally configured IPSec policies, the local cache, and other IPSec settings.
Local cache	The local cache stores IPSec policies after they are downloaded from an Active Directory domain controller by the Policy Agent.
Interface Manager	Interface Manager manages a list that contains items that correspond to each physical and logical network adapter on the system.

The following sections provide additional detail about each of these components.

Policy store

The policy store organizes IPSec policy data and stores it in a format that the Policy Agent can use. In Windows Server 2003, policy data can be stored in the following:

- Active Directory
- Local and remote registry
- A file (for exporting and importing only)

In addition to providing an interface that UI services can use to store policy in each of these media, the policy store does the following:

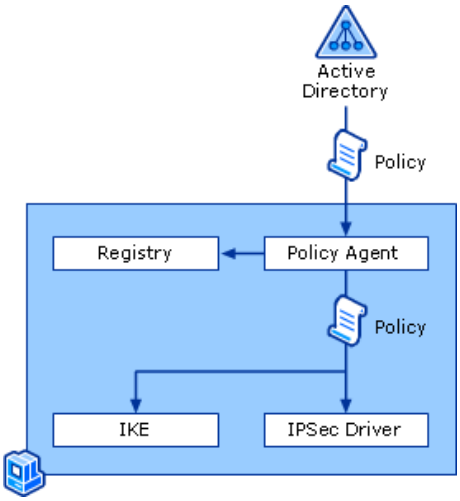
- Provides policy data for default IPSec policies
- Checks policy information for consistency
- Retrieves policy version information

The policy store reads and writes policy information both to and from persistent storage and is aware of shared policy-setting dependencies. This ensures that all policies using shared settings are marked as changed when they are modified and that Windows Server 2003 IPSec components download the modified policies.

Policy Agent

The Policy Agent retrieves IPSec policy information and delivers it to other IPSec components that require this information to perform security services, as shown in the following illustration.

Policy Agent Service Retrieving and Delivering IPSec Policy Information



The Policy Agent performs the following tasks:

- Retrieves the appropriate IPSec policy (if one has been assigned) from Active Directory if the computer is a domain member or from the local registry if the computer is not a member of a domain
- Determines filter list order
- Delivers the assigned IPSec policy information (IP filters) to the IPSec driver

- Delivers both main mode and quick mode settings to IKE
- Polls for changes in policy configuration. If the computer is a member of a domain, policy retrieval occurs when the computer starts, at the interval specified in the IPSec policy, and at the default Winlogon polling interval. You can also manually poll Active Directory for policy by using the **gpupdate /target:computer** command.

If there are no IPSec policies in Active Directory or the registry, or if the IPSec Policy Agent cannot connect to Active Directory, the IPSec Policy Agent waits for policy to be assigned or activated.

The Policy Agent appears in the list of computer services in the Services snap-in under the name IPSEC Services and starts automatically as part of the initialization of the Local Security Authority (LSA) service.

Local registry
Each computer running Windows XP or a Windows Server 2003 has only one local Group Policy object, often called the local computer policy. Using this local Group Policy object allows Group Policy settings to be stored on individual computers regardless of whether they are members of an Active Directory domain.

The local registry maintains the IPSec policy configuration in the following registry key and its subkeys:**HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\IPSec**.

If you assign local IPSec policies and you do not assign Active Directory-based IPSec policies, the local policies are stored in the following registry key:**HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\IPSec\Policy\Local**.

If you assign Active Directory-based IPSec policies, the policies are read from Active Directory and stored in the local cache.

Local cache
The local cache is part of the registry. If you assign an IPSec policy in Active Directory, the policy is stored in and read from Active Directory. A copy of the current policy in Active Directory is maintained in a cache in the local registry at: **HKEY_LOCAL_MACHINE\Software\Policies \Microsoft\Windows\IPSec\Policy\Cache**.

If a computer to which an IPSec policy in Active Directory is assigned cannot connect to the domain, the cached copy of the policy in Active Directory is applied. When the computer reconnects to the domain, new policy information for that computer replaces old, cached information. You cannot configure or manage the cached copy of an IPSec policy in Active Directory.

Interface Manager
Interface Manager maintains a list of physical and logical network adapters on the computer and notifies the Policy Agent when interface and address changes occur. Interface Manager also maintains a complete list of generic filters. Generic filters are filters that are configured to use My IP Address either as a source address or as a destination address. Generic filters are saved in the appropriate IPSec policy storage location with either a source address or a destination address of 0.0.0.0 and a corresponding subnet mask of 255.255.255.255.

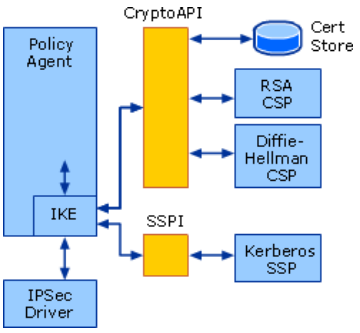
IKE Module Architecture
The IKE module receives authentication and security settings from the Policy Agent and waits for requests to negotiate SAs. When the IKE module receives a request to negotiate an SA from the IPSec driver, the IKE module negotiates both kinds of SAs (the main mode SA and the quick mode SA) with the appropriate endpoint based on the request of the IPSec driver that the policy settings obtained from the Policy Agent. After it has negotiated an SA, the IKE module sends the SA settings to the IPSec driver.

The IKE module has the following components:

- CryptoAPI
- Diffie-Hellman Cryptographic Service Provider (CSP)
- RSA CSP
- Certificate store
- Security Support Provider Interface (SSPI)
- Kerberos Security Support Provider (SSP)

The following figure shows the architecture of the IKE module.

IKE Module Architecture



- A cryptographic service provider (CSP) is an independent software module that provides implementations of cryptographic standards and algorithms. The CSP carries out the cryptographic functions of CrytoAPI, creating keys, destroying them, and using them to perform a variety of cryptographic operations.
- The following table briefly describes the IKE module components.

IKE Module Components

Component	Description
CryptoAPI	CryptoAPI provides a set of functions that allows applications based on Windows to encrypt or digitally sign data in a flexible manner while providing protection for the user's sensitive private key data. Actual cryptographic operations are performed by independent modules known as CSPs. The IKE negotiation must be encrypted. This encryption is limited by what can be configured in IPSec policy. The standard CryptoAPI functions for keyed hashing (using HMAC-MD5 and HMAC-SHA1) and data encryption (using DES and 3DES) are used
Diffie-Hellman CSP	The Diffie-Hellman CSP contains the implementation of the Diffie-Hellman key exchange and determination algorithm. IKE uses only the Microsoft Base or Enhanced CSP for Diffie-Hellman. However, the Diffie-Hellman calculation can be accelerated using the CryptoAPI exponentiation offload interface (OffloadModExpo), as documented in the CryptoAPI SDK.
RSA CSP	The RSA CSP contains the implementation of the Rivest-Shamir-Adleman (RSA) cryptographic algorithms. When certificate authentication is selected, IKE checks the CryptoAPI default provider to see if it is capable of performing RSA 512-bit digital signatures. If so, then IKE uses this default CSP. If not, IKE enumerates the RSA providers, selects hardware-based providers first and ensures that they can provide 512-bit signatures. IKE performs these actions to open the certificate store. The CSP that is used for signature operations during IKE negotiation is specified by the certificate selected during the IKE negotiation; the certificate's associated CSP for the private key signature is used. As long as the RSA CSP supports the NOHASHID flag for the **CryptSignHash( )** API call, IKE can use that CSP for private key signing of IKE payloads. Because the enumeration process does not permit IKE to know if the CSP supports the NOHASHID option, it is possible to choose a certificate that appears valid, but whose CSP does not allow IKE to construct the proper signature.
Certificate store	The certificate store is a permanent storage location where certificates, certificate revocation lists, and certificate trust lists are stored. The certificate store is a physical store on the Windows Server 2003 computer, but it is viewed logically as belonging to the user account of the currently logged-on user, a service account, or the computer account. IKE can use only the computer account, usually referred to as the computer store. You can view the computer store by using the Certificates snap-in.
SSPI	The SSPI enables network applications to access one of several security providers to establish authenticated connections and exchange data securely over those connections.
Kerberos SSP	The Kerberos SSP contains an implementation of the Kerberos security protocol. The Kerberos SSP is an SSPI provider

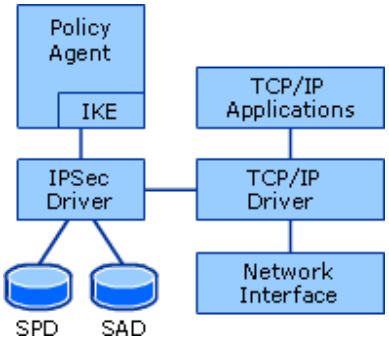
IPSec Driver Architecture

The IPSec driver is a kernel-mode component that monitors and secures IP packets. In addition to the Policy Agent and IKE, the IPSec driver uses the following components: the Security Association Database (SAD), the Security Policy Database (SPD), the TCP/IP driver, TCP/IP applications, and the network interface.

The IPSec driver matches IP packet information with the IP filters that are configured in the active SPD. If traffic must be secured, the IPSec driver either uses the appropriate SA to determine how to provide packet security or requests that the IKE module negotiate SAs to be used to provide packet security. After the IPSec driver determines which SA to use, it creates and validates encrypting, decrypting, and hashing to create or interpret the AH and ESP headers on an IPSec-protected packet.

The following figure shows the IPSec driver architecture and how the driver interacts with other components in Windows Server 2003.

IPSec Driver Architecture



The following table briefly describes the IPSec driver components.

IPSec Driver Components

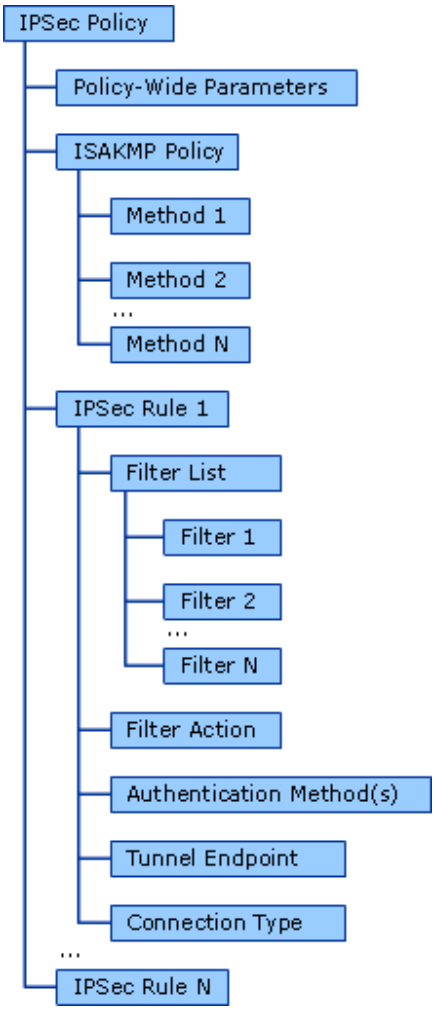
Component	Description
SAD	The SAD is a database in the IPSec driver that contains the parameters associated with each active SA. This database is populated automatically from the IKE module.
SPD	The SPD is a database in the IPSec driver that specifies the filter lists and associated settings that determine the status of all inbound or outbound IP traffic. Inbound packets are checked to ensure that they have been secured according to policy. Outbound packets are permitted, blocked, or secured, according to policy. For secured traffic, the security policy that is used is the negotiated SA, which is stored in the SAD.
TCP/IP driver	The TCP/IP driver is the Windows Server 2003 implementation of the TCP/IP protocol. It is a kernel-mode component that is loaded from the Tcpip.sys file during startup.
TCP/IP applications	TCP/IP applications use TCP/IP and access TCP/IP network services through an appropriate network API, such as Windows Sockets, NetBIOS, or Remote Procedure Call (RPC).
Network interface	The network interface is the logical or physical interface over which IP packets are sent and received. The details of the Network Driver Interface Specification (NDIS) interface, the network adapter driver, and the physical media over which the IP packets are sent and received are beyond the scope of this subject.

Policy Data Structure

The data in a policy indicates the desired protection for the traffic between computers on a network. The data is made up of various computer-related attributes (for example, IP address and port number), the communication methods allowed (for example, algorithms and key lengths), and the IKE key negotiation and management. The policy store updates and stores the policy data. The Policy Agent retrieves the stored policy data and makes it available to all IPSec components.

As shown in the following figure, an IPSec policy contains several subsets of information.

IPSec Policy Structure



The following table describes the components of an IPSec policy.

IPSec Policy Components

Component	Description
Policy-wide parameters	Policy-wide parameters specify the polling interval used to detect changes in policy. The policy-wide parameters are configured on the **General** tab in the properties of an IPSec policy.
ISAKMP policy	The ISAKMP policy contains IKE parameters, such as encryption key lifetimes, and other settings. The ISAKMP policy settings are configured in the **Key Exchange Settings** dialog box, which is available from the **Advanced** button on the **General** tab in the properties of an IPSec policy. The ISAKMP policy also contains a list of security methods for protecting the identity of IPSec peers during authentication. These methods are, listed in order of preference, and are configured in the **Key Exchange Security Methods** dialog box, which is available from the **Methods** button on the **Key Exchange Settings** dialog box.
IPSec rules	IPSec rules contain a statement that associates a filter list with a filter action, an authentication method, an IPSec mode, and other settings. Typically, an IPSec rule is configured for a specific purpose (for example, "Block all inbound traffic from the Internet to TCP port 135"). You can define many IPSec rules in a single IPSec policy. IPSec rules are configured on the **Rules** tab in the properties of an IPSec policy.

IPSec rules associate IKE negotiation parameters with one or more IP filters. The following table describes the components of an IPSec rule.

IPSec Rule Components

Component	Description
Filter list	The filter list contains one or more predefined filters that describe the types of traffic to which an action (permit, block, or secure) is applied. The filter list is configured on the **IP Filter List** tab in the properties of an IPSec rule within an IPSec policy.
Filter action	The filter action defines the security requirements for the data transmission. A filter action can be configured to permit traffic, block traffic, or negotiate secure communications with IPSec for packets matching the filter list. If security negotiation is selected, you must also configure security methods and their order: whether initial incoming unsecured traffic should be accepted, whether unsecured communication with computers that do not support IPSec should be allowed, and whether to use perfect forward secrecy (PFS). PFS is a mechanism that determines whether the existing keying material for a master key can be used to derive a new session key. Session key PFS performs a new Diffie-Hellman key exchange to generate new master key keying material instead of using master key keying material to derive more than one session key. The negotiation settings are configured on the **Filter Action** tab in the properties of an IPSec rule within an IPSec policy.
Authentication method(s)	An IPSec rule contains one or more authentication methods, listed in order of preference, that are used for protection during IKE negotiations. The available authentication methods are the Kerberos v5 protocol, the use of a certificate issued from a specified certification authority (CA), or a preshared key. The negotiation data is configured on the **Authentication Methods** tab in the properties of an IPSec rule within an IPSec policy.
Tunnel endpoint	A setting that specifies whether traffic is tunneled and, if it is, specifies the tunnel endpoint, which is the tunneling computer that is closest to the IP traffic destination, as specified by the associated IP filter list. Two rules are required to describe an IPSec tunnel. For the outbound traffic rule, the tunnel endpoint is the IP address or subnet of the IPSec peer on the other end of the tunnel. For the inbound traffic rule, the tunnel endpoint is an IP address or subnet configured on the local computer. The tunnel endpoint is configured on the **Tunnel Setting** tab in the properties of an IPSec rule within an IPSec policy.
Connection type	The connection type setting specifies whether the rule applies to only local area network (LAN) connections, to only dial-up connections, or to both types of connections. The interface applicability is configured on the **Connection Type** tab in the properties of an IPSec rule within an IPSec policy.

Note

- The Kerberos v5 authentication method is not supported on computers running the Microsoft Windows XP Home Edition operating system.

Default response rule

Each policy has a default response rule. The default response rule has the IP filter list of **<Dynamic>** and the filter action of **Default Response** when the list of rules is viewed with the IP Security Policies snap-in. The default response rule cannot be deleted, but it can be deactivated. It is activated for all of the default policies and in the IP Security Policy wizard.

IPSec uses the default response rule to ensure that the computer responds to requests for secure communication. If an active policy does not include a rule for a computer requesting secure communication, IPSec applies the default response rule and negotiates security. For example, if Host A intends to communicate securely with Host B, but Host B does not have an inbound filter defined for Host A, then IPSec uses the default response.

For the defense response rule, you can configure only the security methods for secure traffic and the authentication methods. The following table lists the default security methods for the default response rule.

Default Security Methods for the Default Response Rule

Type	AH Integrity	ESP Confidentiality	ESP Integrity
Encryption and Integrity	<None>	3DES	SHA1
Custom	<None>	3DES	MD5
Custom	<None>	DES	SHA1
Custom	<None>	DES	MD5
Custom	SHA1	<None>	<None>
Custom	MD5	<None>	<None>

You can configure the security methods and their preference order on the **Security Methods** tab in the properties of the default response rule in the IP Security Policies snap-in.

The default response rule works in the following way:

1. If the IKE module receives a request to negotiate security, it queries the Policy Agent for a matching filter for traffic to and from the source and destination address of the ISAKMP message. If a matching filter is explicitly configured, the IKE negotiation is based on the settings of the associated rule.
2. If no matching filter is found and the default response rule is not activated, IKE negotiation fails.
3. If no matching filter is found and the default response rule is activated, then IKE dynamically creates an IP filter within the Policy Agent that corresponds to the traffic specification of the incoming ISAKMP message. IKE authenticates and negotiates security based on the settings on the **Authentication Methods** and **Security Methods** tabs for the default response rule.

You configure the default authentication method for the default response rule by using the IP Security Policy wizard.

Example: Default response rule
Typically, the default response rule is used when a group of servers are configured with policy to secure communications between themselves and any IP address and to accept unsecured communication, but respond using secured communications. The client computers are configured with the default response rule. When the clients communicate with each other, the traffic is not secured. When the clients communicate with the server, the traffic is secured (with the exception of the initial packet sent by the client to the server).

For example, a client computer can reliably exchange data with a server by using Transmission Control Protocol (TCP). A TCP connection is established through the exchange of three TCP segments: SYN (synchronize), SYN-ACK (synchronize-acknowledgement), and ACK (acknowledgement).

Because the client computer does not have an explicit rule to secure traffic to the server, the client computer sends an unsecured SYN segment to the server. The server receives the SYN segment and checks its filters. It finds the filter that requires secure traffic to and from any IP address. However, the filter action settings also allow the server to accept unsecured communication, responding with secured communication. Consequently, the SYN segment sent by the client and received by the server is sent to the TCP/IP protocol on the server.

The TCP/IP protocol on the server responds by sending a SYN-ACK segment back to the client. This IP packet is sent to the IPSec driver on the server. Checking the filter settings to require secured communications, the IPSec driver notes that there are no active SAs between itself and the client. The IPSec driver requests that the IKE module negotiate SAs for secure communication.

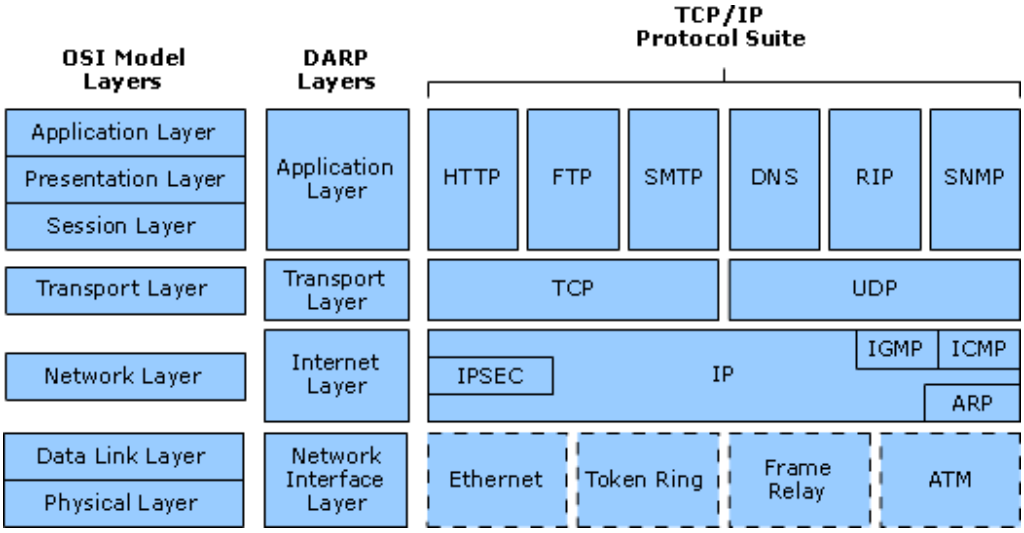
The IKE module on the server sends an ISAKMP message to the client to begin the process of negotiating SAs for secure communications. When the ISAKMP message is received on the client computer, the IKE module on the client computer queries the filter lists in the Policy Agent. Because it finds no explicit filter match and the default response rule is activated, the IKE module creates a dynamic filter for traffic to and from the client and server computer.

IKE negotiation continues based on the policy settings of the server and the client. After IKE negotiation is finished, the server sends the secured TCP SYN-ACK segment to the client. The client responds with a secured TCP ACK segment and secured TCP data can be exchanged between the client and the server.

IPSec Protocols

IPSec is integrated at the IP layer (layer 3) of the TCP/IP stack, so it provides security for almost all protocols in the TCP/IP suite. Because IPSec is applied to all applications, you do not need to configure separate security for each application that uses TCP/IP.

IPSec Protocol Architecture



IPSec AH and ESP Protocols

The IPSec protocols — Authentication Header (AH) and Encapsulating Security Payload (ESP) — provide data and identity protection for each IP packet. The AH protocol is an IPSec protocol that provides data origin authentication, data integrity, and anti-replay protection for the entire packet (the IP header and the data payload carried in the packet, except fields in the IP header that are allowed to change in transit). AH can be used alone, in combination with the ESP protocol, or in IPSec tunnel mode. IPSec tunnel mode is used to protect site-to-site (gateway-to-gateway) traffic between networks, such as site-to-site networking through the Internet. The sending gateway encapsulates the entire IP datagram by adding a new IP header and then protects the new packet using one of the IPSec protocols. Windows Server 2003 supports IPSec tunnel mode for configurations where Layer Two Tunneling Protocol (L2TP) cannot be used.

The ESP protocol is an IPSec protocol that provides data confidentiality, data origin authentication, data integrity, and anti-replay protection for the ESP payload. The ESP protocol can be used alone, in combination with the AH protocol, or in IPSec tunnel mode.

IPSec AH and ESP Protocols in IPSec Transport Mode

IPSec protocols provide data and identity protection for each IP packet by adding their own security protocol header to each packet. This section describes how AH and ESP protect IP packets when IPSec is used in transport mode.

You use IPSec in transport mode to protect traffic in end-to-end communications scenarios (for example, for communications between a client and a server). You can also use IPSec in transport mode for basic packet filtering (that is, to statically permit or block traffic based on source and destination address combinations and on the IP protocol and TCP and UDP ports).

IPSec transport mode encapsulates the original IP payload with an IPSec header (AH or ESP).

AH transport mode

The AH protocol provides data origin authentication, data integrity, and anti-replay protection *for the entire packet* (both the IP header and the data payload carried in the packet), except for the fields in the IP header that are allowed to change in transit. AH does not provide data confidentiality, which means that it does not encrypt the data. The data is readable, but protected from modification and spoofing.

AH Transport Mode Packet Structure

As shown in the figure, data integrity and authentication are provided by the placement of the AH header between the IP header and the IP packet payload. The AH protocol uses keyed hash algorithms to sign the packet for integrity. The AH protocol is identified in the IP header with an IP protocol ID of 51. This protocol can be used alone or with the ESP protocol.

The following table describes the AH header fields.

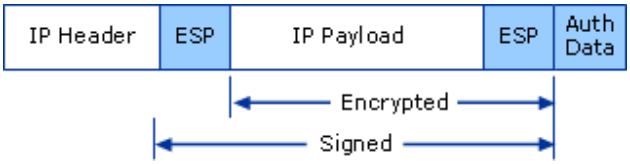
AH Header Fields

AH Header Field	Function
Next header	Identifies the IP payload by using the IP protocol ID. For example, a value of 6 represents TCP.
Length	Indicates the length of the AH header.
SPI	Used in combination with the destination address and the security protocol (AH or ESP) to identify the correct SA for the communication. The receiver uses this value to determine with which SA the packet is identified.
Sequence number	Provides anti-replay protection for the packet. The sequence number is a 32-bit, incrementally increasing number (starting from 1) that indicates the packet number sent over the SA for the communication. The sequence number cannot repeat for the life of the quick mode security association. The receiver checks this field to verify that a packet for a security association with this number has not already been received. If one has been received, the packet is rejected.
Authentication data	Contains the integrity check value (ICV), also known as the message authentication code, which is used to verify both data origin authentication and data integrity. The receiver calculates the ICV value and checks it against this value (which is calculated by the sender) to verify integrity. The ICV is calculated over the IP header, the AH header, and the IP payload.

ESP transport mode

The ESP protocol provides data origin authentication, data integrity, anti-replay protection, and the option of confidentiality *for the IP payload only*. ESP in transport mode does not protect the entire packet with a cryptographic checksum nor does it protect the IP header.

ESP Transport Mode Packet Structure



As shown in the figure, the ESP header is placed before the IP payload and an ESP trailer and ESP trailer and authentication data field are placed after the IP payload. The ESP protocol is identified in the IP header with the IP protocol ID of 50.

The following table describes the ESP header fields.

ESP Header Fields

ESP Header Field	Function
SPI	When used in combination with the destination address and the security protocol (AH or ESP), the SPI identifies the SA for the communication. The receiver uses this value to determine the SA with which this packet should be identified.
Sequence number	Provides anti-replay protection for the packet. The sequence number is a 32-bit, incrementally increasing number (starting from 1) that indicates the packet number sent over the quick mode SA for the communication. The sequence number cannot repeat for the life of the quick mode SA. The receiver checks this field to verify that a packet for an SA with this number has not already been received. If one has been received, the packet is rejected.

The following table describes the ESP trailer fields.

ESP Trailer Fields

ESP Trailer Field	Function
Padding	Padding of 0 to 255 bytes is used to ensure that the encrypted payload is on byte boundaries required by encryption algorithms.
Padding length	Indicates the length of the Padding field in bytes. The receiver uses this field to remove padding bytes after the encrypted payload with the padding bytes has been decrypted.
Next header	Identifies the type of data in the payload, such as TCP or UDP.

The following table describes the ESP authentication trailer field.

ESP Authentication Trailer Field

ESP Authentication Trailer Field	Function
Authentication data	Contains the ICV, also known as the message authentication code, which is used to verify both message authentication and integrity. The receiver calculates the ICV value and checks it against this value (which is calculated by the sender) to verify integrity. The ICV is calculated over the ESP header, the payload data, and the ESP trailer.

IPSec AH and ESP Protocols in IPSec Tunnel Mode

You use IPSec tunnel mode primarily to protect traffic between sites that must traverse an untrusted path. For example, you can use tunnel mode to do the following:

- Establish gateway-to-gateway tunnels between sites, when the gateways or end systems do not support L2TP/IPSec Virtual Private Network (VPN) connections.

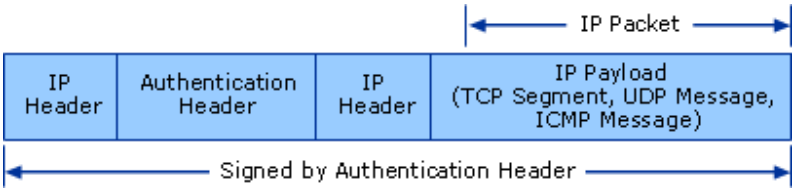
- Protect traffic end-to-end when one endpoint of the communication does not support IPSec. You can send protected traffic to a computer that supports tunnel mode and that is placed immediately in front of the computer that does not support IPSec.

With tunnel mode, an entire IP packet is encapsulated with an AH or ESP header and an additional IP header. The IP addresses of the outer IP header are the tunnel endpoints, and the IP addresses of the encapsulated IP header are the ultimate source and destination addresses.

You can use both the ESP and AH protocols in combination when tunneling to provide both confidentiality for the tunneled IP packet and integrity and authentication for the entire packet.

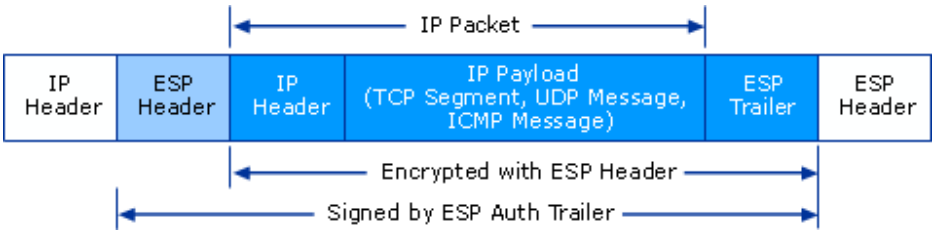
AH tunnel mode
AH tunnel mode encapsulates an IP packet with an AH and IP header and signs the entire packet for integrity and authentication.

AH Tunnel Mode Packet Structure



ESP tunnel mode
ESP tunnel mode encapsulates an IP packet with both an ESP and IP header and an ESP authentication trailer.

ESP Tunnel Mode Packet Structure



The signed portion of the packet indicates where the packet has been signed for integrity and authentication. The encrypted portion of the packet indicates what information is protected with confidentiality.

Because a new header for tunneling is added to the packet, everything that comes after the ESP header is signed (except for the ESP authentication trailer) because it is now encapsulated in the tunneled packet. The original header is placed after the ESP header. The entire packet is appended with an ESP trailer before encryption occurs. All data that follows the ESP header, except for the ESP authentication trailer, is encrypted. This includes the original header, which is now considered to be part of the data portion of the packet.

The entire ESP payload is then encapsulated within the new tunnel header. The tunnel header is not encrypted because it is used only to route the packet from origin to tunnel endpoint.

If the packet is being sent across a public network, it is routed to the IP address of the gateway for the receiving intranet. The gateway decrypts the packet, discards the ESP header, and uses the original IP header to route the packet to the intranet computer.

IPSec Processes and Interactions

This section describes the following IPSec processes and interactions:

- Policy Agent initialization
- Policy data retrieval and distribution
- IKE main mode and quick mode negotiation
- Key protection
- IPSec Driver Processes

Policy Agent Initialization

When the Policy Agent is started by the Local Security Authority (LSA) service, it performs a number of initialization steps and then begins a cycle during which it waits for a number of events to be signaled.

At startup, the Policy Agent checks the registry for the value that indicates the location of the computer's IPSec policy in Active Directory. To find the value, it checks the following registry key**HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\IPSec\GPTIPSECPolicy\DSIPSECPolicyPath**.

If the computer is a member of a domain and an IPSec policy has been configured in Active Directory, this value was set during computer initialization. If the computer is not a member of a domain or there is no IPSec policy in Active Directory, the Policy Agent finds no value.

After initializing interval values for event processing, the Policy Agent starts the Interface Manager (IM). The IM establishes the event processing that needs to be signaled whenever a network interface configuration or status changes. The Policy Agent then starts the IPSec driver.

Next, IKE starts. At this point, additional initialization occurs. This includes determining through the IPSec driver whether strong cryptography is available.

In preparation for loading the IPSec policy from either Active Directory or the local registry, the Policy Agent initializes the policy state and polling interval. It then loads the IPSec policy from the appropriate store, as described in the following section. After the policy is loaded, the Policy Agent begins a service loop.

If, during the service wait cycle, the Policy Agent encounters either an unexpected event or the Policy Agent service stop event, the Policy Agent shuts down IKE, stops the IPSec driver, cleans up local data, and then stops itself.

Policy Data Retrieval and Distribution
The Policy Agent loads the static IPSec policy from one of the stores if one of the following situations occurs:

- The Policy Agent has initialized and started IKE and the IPSec driver.
- An event has signaled the Policy Agent to reload the policy.
- A polling interval timeout occurs when the Policy Agent unsuccessfully accesses the current policy location to check for updates.

The Policy Agent first attempts to retrieve the policy from Active Directory. If it successfully does so, the Policy Agent supplies the retrieved policy data to IKE and the IPSec driver. The Policy Agent then copies the IPSec policy to the registry cache.

If the Policy Agent fails to retrieve the policy from Active Directory, it checks the local cache for the IPSec policy data. If the local cache contains policy data, the Policy Agent supplies the data to IKE and the IPSec driver.

If the Policy Agent fails to retrieve policy from Active Directory and the local cache, it checks the registry for local policy data. If the policy is successfully retrieved and the data is successfully provided to IKE and the IPSec driver, the Policy Agent is placed in the Local Downloaded policy state.

If the Policy Agent successfully loads a policy, the polling interval for policy checks is set from the value in the policy data. If the Policy Agent fails to load a policy, it goes into the Initial policy state and sets the polling interval to the default value.

While loading the static IPSec policy, the Policy Agent notes the state of the static policy data. If a service loop timeout occurs, the Policy Agent uses the state information to determine activity.

IKE Main Mode and Quick Mode Negotiation
The IKE component is a Policy Agent service. IKE is started by the Policy Agent, restarted by the Policy Agent as needed, and shut down when the Policy Agent shuts down. All policy data that IKE requires for operation is provided by the Policy Agent.

IKE establishes a combination of mutually agreeable policy and keys that defines the SA; the security services, protection mechanisms, and cryptographic keys between communicating peers.

To ensure successful and secure communication, IKE performs a two-phase negotiation operation. Phase 1 negotiation is known as main mode negotiation and Phase 2 is known as quick mode negotiation. The IKE main mode SA (also known as the ISAKMP SA) protects the IKE negotiation itself. The SAs created during the second IKE negotiation phase are known as the quick mode SAs (also known as IPSec SAs). The quick mode SAs protect application traffic. Two quick mode SAs are negotiated, one for inbound and one for outbound traffic.

Main Mode Negotiation
IKE performs main mode negotiation with an IPSec peer to establish protection mechanisms and keys for subsequent use in protecting quick mode IKE communications. IKE main mode negotiation occurs in three parts:

- Part one: Negotiation of protection mechanisms
- Part two: Diffie-Hellman exchange
- Part three: Authentication

Main mode negotiation consists of the exchange of a series of six ISAKMP messages. An ISAKMP message is the payload of a (User Datagram Protocol) UDP message with the source and destination UDP ports set to 500 (or 4500). An ISAKMP message has an ISAKMP header and one or more ISAKMP payloads as defined in RFC 2408.

Both the initiator and the responder in the exchange send three messages. The initiator is the IPSec peer that initiates secure communications by sending the first message. The responder, which sends the second message, is the IPSec peer with which the initiator is requesting secure communications.

The following table shows the first four main mode messages, which are not encrypted.

Main Mode Messages 1 Through 4

Main Mode Message	Sender	Payload
1	Initiator	ISAKMP header, Security Association (contains proposals)
2	Responder	ISAKMP header, Security Association (contains a selected proposal)
3	Initiator	ISAKMP header, Key Exchange (contains Diffie-Hellman key), Nonce, additional payloads (depending on authentication method)
4	Responder	ISAKMP header, Key Exchange (contains Diffie-Hellman key), Nonce, additional payloads (depending on authentication method)

The first four main mode messages contain the following ISAKMP payloads:

- **Security Association**. The Security Association payload sent in message 1 is a list of proposed protection mechanism for the main mode SA. The Security Association payload sent in message 2 is a specific protection suite for the main mode SA that is common to both IPSec peers. It is selected by the responder.
- **Key Exchange**. The Key Exchange payload is sent in message 3 by the initiator and in message 4 by the responder and contains Diffie-Hellman key determination information for the Diffie-Hellman key exchange process.
- **Nonce**. The Nonce payload is sent in messages 3 and 4 and contains a nonce, which is a pseudorandom number that is used only once. The initiator and responder each send their own unique nonces. Nonces are used to provide replay protection.

Depending on the authentication method that is selected in the IPSec policy, messages 3 and 4 might contain additional payloads. The payloads of all messages beyond the first four messages are encrypted and vary based on the authentication method selected.

Note

- It is important to understand the differences in negotiation behavior between initiating an IKE main mode negotiation and quick mode negotiation and responding to one, and rekeying an existing one. The IKE RFC 2409 requires that rekeys can be performed by either peer (in either direction) at any time, regardless of the security association lifetimes negotiated. Therefore, the computer that initiates a negotiation might become the responder and these roles might alternate many times. Some of these differences are due to behavior required for interoperability and some are caused by enforcement of policy settings.

Part One: Negotiation of protection mechanisms

When initiating an IKE exchange, IKE proposes protection mechanisms based on the applied security policy. Each proposed protection mechanism includes attributes for encryption algorithms, hash algorithms, authentication methods, and Diffie-Hellman groups. The first part of the main mode is contained in main mode messages 1 and 2.

The following table lists the protection mechanism attribute values that are supported by Windows Server 2003 IKE. These values are described in more detail in later sections.

Main Mode Attribute Values Supported by IKE

Main Mode Attribute	Value
Encryption algorithm	DES, 3DES
Integrity algorithm	HMAC-MD5, HMAC-SHA1
Authentication method	Kerberos v5, public key certificate, preshared key
Diffie-Hellman group	Group 1, Group 2, Group 14 (2048)

The encryption algorithm, integrity algorithm, and Diffie-Hellman group are configured as one of multiple key exchange security methods.

The initiating IKE peer proposes one or more protection suites in the same order as they appear in the applied security policy. If one of the protection suites is acceptable to the responding IKE peer, the responder selects it for use and responds to the initiator with its choice. Because the responding IKE peer might not be running Windows Server 2003 or Windows 2000 and is selecting the first proposed protection suite that is acceptable, the protection suites in the applied security policy should be configured in the order of most secure to least secure.

Part Two: Diffie-Hellman exchange
After a protection suite is negotiated, IKE queries a Diffie-Hellman CSP through CryptoAPI to generate a Diffie-Hellman public and private key pair based on the negotiated Diffie-Hellman group. The Diffie-Hellman public key is sent to the IKE peer in an ISAKMP Key Exchange payload. Main mode negotiation part 2 is contained in main mode messages 3 and 4.

The cryptographic strength of a Diffie-Hellman key pair is related to its prime number length (key size). Windows Server 2003 IKE supports the following Diffie-Hellman groups:

- Group 1 (768 bits)
- Group 2 (1024 bits)
- Group 14 (2048 bits)

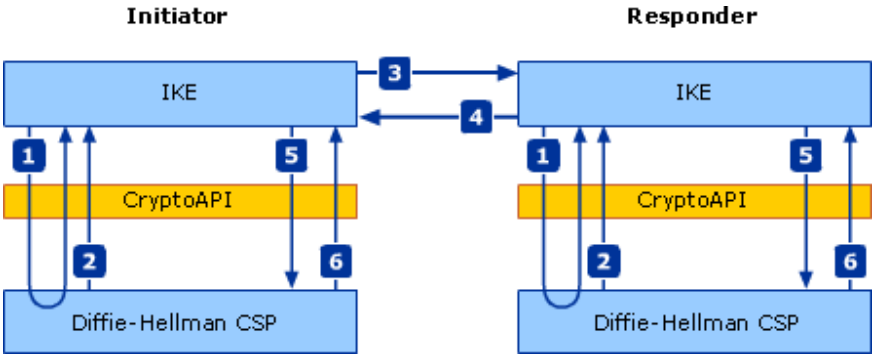
Note

- For enhanced security, Windows Server 2003 IPSec includes Diffie-Hellman Group 14, which provides 2048 bits of keying strength. However, Diffie-Hellman Group 14 is not currently supported in Windows 2000 or Windows XP for general IPSec policy use. For updated information about the availability of Diffie-Hellman Group 14 for L2TP/IPSec connections for Windows XP and Windows 2000, search for article 818043, L2TP/IPSec NAT-T Update for Windows XP and Windows 2000, in the Microsoft Knowledge Base.

After the Diffie-Hellman public keys are exchanged, IKE accesses CryptoAPI to compute the shared key based on the mutually agreeable authentication method.

The following figure shows the Diffie-Hellman exchange between the IPSec peers and the relationship between IKE, CryptoAPI, and the Diffie-Hellman CSP.

IKE Diffie-Hellman Key Exchange



The Diffie-Hellman exchange occurs in the following steps:

1. On each IPSec peer, IKE requests that the first Diffie-Hellman CSP generate a Diffie-Hellman public and private key pair based on the Diffie-Hellman group selected during main mode messages 1 and 2.
2. The public portion of the Diffie-Hellman public and private key pair is returned to IKE.
3. The initiator sends the Diffie-Hellman public key to the responder in a Key Exchange payload (main mode message 3).
4. The responder sends its Diffie-Hellman public key to the initiator in a Key Exchange payload (main mode message 4).
5. On each IPSec peer, the IKE component sends the other IPSec peer's Diffie-Hellman public key to the Diffie-Hellman CSP.
6. The Diffie-Hellman CSP computes the shared secret and returns the value to the IKE component.

Part Three: Authentication
IKE supports three methods of authentication:

- Kerberos v5
- Public key certificate
- Preshared key

The authentication that occurs is a computer-based authentication, also known as machine-based authentication. The authentication process verifies only the identity of the computers, not the individual using the computer when the authentication process occurs.

Kerberos v5 authentication
Kerberos v5 authentication is the default authentication standard in Windows Server 2003 and Windows 2000 domains. Any computer in the domain or a trusted domain can use this method of authentication.

Note

- In Windows Server 2003, the Kerberos protocol is no longer a default exemption. Therefore, if you want to enable Kerberos authentication, you must create filters in the IPSec policy that explicitly allow such traffic. For more information, see article 810207, "IPSec Default Exemptions Are Removed in Windows Server 2003" in the Microsoft Knowledge Base.

Windows IKE Kerberos authentication is based on the Generic Security Service (GSS) API IKE authentication method, which is described in the Internet draft entitled "A GSS-API Authentication Method for IKE."

The following table lists the ISAKMP messages exchanged during a Kerberos authentication main mode negotiation.

Kerberos Authentication Method Main Mode Messages

Main Mode Message	Sender	Payload
1	Initiator	ISAKMP header, Security Association (contains proposals)
2	Responder	ISAKMP header, Security Association (contains a selected proposal)
3	Initiator	ISAKMP header, Key Exchange (contains Diffie-Hellman key), Nonce, Initiator Kerberos Token
4	Responder	ISAKMP header, Key Exchange, Nonce, Responder Kerberos Token
5*	Initiator	ISAKMP header, Identification, Initiator Hash
6*	Responder	ISAKMP header, Identification, Responder Hash

* ISAKMP payloads of message are encrypted.

Authentication occurs when:

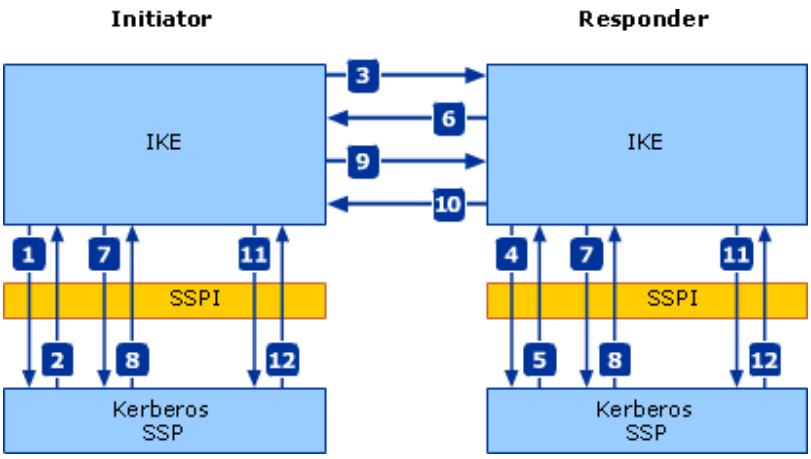
- Each peer authenticates the other peer's Kerberos token: The responder verifies the initiator's Kerberos token and the initiator verifies the responder's Kerberos token.
- Each peer's hash is calculated and verified: The responder verifies the initiator's hash and the initiator verifies the responder's hash.

The hash calculation is performed over the following:

- The Diffie-Hellman public values of the initiator and responder
- The initiator and responder cookies
- The ISAKMP payloads of message 2
- The identity name string for the IPSec peer
- The IPSec peer's Kerberos token

To perform Kerberos authentication, Windows Server 2003 uses a Kerberos SSP that is accessible through the SSPI. The following figure shows the exchange of Kerberos tokens between the IKE peers and the relationship between the IKE, SSPI, and the Kerberos SSP.

IKE Authentication with Kerberos



Kerberos authentication is performed in the following steps:

1. IKE on the initiator accesses the SSPI to initialize a security context.

2. The initiator's Kerberos SSP creates a Kerberos token and returns it to the IKE component.
3. The initiator sends the initiator's Kerberos token and computer identity to the responder in main mode message 3.
4. The responder IKE component accesses the SSPI to acquire a security context.
5. The responder's Kerberos SSP creates a Kerberos token and returns it to IKE.
6. The responder sends the responder Kerberos token and computer identity to the sender in main mode message 4.
7. On each IPSec peer, IKE creates the hash for the next main mode message to be sent and then requests that SSPI sign the hash with the Kerberos session key.
8. The Kerberos SSP returns the signature to the IKE component.
9. The initiator sends the signed initiator hash to the responder in main mode message 5.
10. The responder sends the signed responder hash to the initiator in main mode message 6.
11. On each IPSec peer, IKE accesses the SSPI to compute the hash for the other peer. The initiator accesses the SSPI to compute the responder hash. The responder accesses the SSPI to compute the initiator hash.
12. The Kerberos SSP returns the computed hash to the IKE component where the hash value is verified to complete IKE authentication. The initiator compares its calculated responder hash with the responder hash received in main mode message 6. The responder compares its calculated initiator hash with the initiator hash received in main mode message 5.

The following sections describe the IKE certificate selection and acceptance process. If you decide to use certificates for IKE authentication, understanding this process and its requirements is integral to ensuring proper deployment.

Public key certificate authentication

Windows IKE performs public key certificate authentication during main mode in compliance with RFC 2409. IKE uses CryptoAPI to retrieve the computer certificate, verify peer certificates and certificate chains, check certificate revocation, and create and verify digital signatures. All certificate, certificate chain, and signature information is exchanged in main mode messages, as shown in the following table.

Certificate-based IKE Authentication Main Mode Messages

Main Mode Message	Sender	Payload
1	Initiator	ISAKMP header, Security Association (contains proposals)
2	Responder	ISAKMP header, Security Association (contains a selected proposal)
3	Initiator	ISAKMP header, Key Exchange (contains Diffie-Hellman key), Nonce
4	Responder	ISAKMP header, Key Exchange (contains Diffie-Hellman key), Nonce, Certificate Request
5*	Initiator	ISAKMP header, Identification, Certificate, Certificate Request, Signature
6*	Responder	ISAKMP header, Identification, Certificate, Signature

* ISAKMP payloads of message are encrypted.

IKE certificate selection process

When IKE negotiates to use certificates for authentication, the following process is used to select a computer certificate:

1. The list of trusted roots is prepared. This is the list of the CA root names provided by the peer in the Certificate Request Payloads (CRPs), and it matches the CA root names configured in the list of trusted

roots in the appropriate authentication method of the IPSec policy. If there are no matching CA root names, all trusted CA root names from the appropriate authentication method are used.

2. IKE searches the computer store for an IPSec certificate that chains to any of the trusted CA roots identified in step 1. An IPSec certificate contains an Enhanced Key Usage (EKU) attribute with a value equal to the IP security IKE intermediate object identifier (OID) 1.3.6.1.5.5.8.2.2.
3. For each certificate chain found, checks are performed to verify the following:

    - The certificate chain does not have any trust errors.
    - The certificate chain is not a root-only chain.
    - The computer certificate has a private key.
    - The computer certificate has an RSA type public/private key pair.
    - The computer certificate has a public key length that is greater than 512 bits.
    - The computer certificate has a Digital Signature key usage.
    - The computer certificate is not a CA signing certificate that is used to issue certificates.
    - The certificate chain passes certificate revocation list (CRL) checking, which is performed by default or if the value of the **StrongCRLCheck** registry subkey is set to **1** or **2**. For more information about CRL checking, see "IPSec CRL checking" later in this section.

    If all of these checks succeed, IKE selects the certificate chain to be sent to the IPSec peer. If any of these checks fails, IKE continues to search for another IPSec type certificate, using the same list of root CA names.

4. If a valid computer certificate chain is not located, IKE retries the process, from step 2. Although it uses the same list of root CA names, IKE does not search for an IPSec type certificate.
5. If a valid computer certificate chain is still not found and if the list of root CA names in step 1 is a subset of the names allowed by the local IPSec policy, IKE retries, from step 2. This time, IKE uses the entire list of root CA names allowed by the local authentication method.
   This step is required for successful authentication when cross-certificates are used to establish trust relationships.
6. After IKE selects a computer certificate, it includes all intermediate certificates in the chain up to the root, except for the root CA certificate. A certificate chain in PKCS#7 format is then sent to the IPSec peer. If there are no intermediate CAs, only the computer certificate is sent.
   If a computer certificate cannot be selected, the authentication fails.
   Note

    - If IKE negotiates with another computer running Windows Server 2003, or with other Microsoft IKE implementations that use IPSec NAT-T (such as Microsoft L2TP/IPSec VPN Client), a special method to avoid fragmentation of ISAKMP UDP packets might be implemented. Otherwise, the ISAKMP message that contains the certificate chain will likely be fragmented as the packet is transmitted.

IKE certificate acceptance process

1. IKE receives the peer's certificates or certificate chains and verifies that the peer's certificates chain up to any of the root CAs in the appropriate authentication method of the local IPSec policy.
2. For each peer certificate chain, checks are performed to verify that:

    - The computer certificate Subject Name or Subject AltName is consistent with the peer's ID field passed in the IKE negotiation.

- The computer certificate chain does not have any trust errors. If there is a trust error, the peer authentication fails.

3. If the two checks in step 2 succeed, checks are performed for the peer certificate chain to verify that:

   - The certificate chain passes CRL checking, which is performed by default or if the value of the**StrongCRLCheck** registry subkey is set to **1** or **2**.
   - The computer certificate has an RSA type public/private key pair.
   - The computer certificate has a public key length that is greater than 512 bits.
   - The computer certificate has a Digital Signature key usage.

   If any of these checks fails, the peer authentication fails.

4. If certificate-to-account mapping is enabled in the IPSec policy for the certificate root CA of the peer, IKE calls the Windows secure channel (Schannel) APIs to perform the mapping. Schannel completes the mapping and builds an access token for the computer account. This access token is automatically evaluated against the **Access this computer from the network** or the **Deny this computer access from the network** logon right defined in Group Policy Security settings.
   If the logon right evaluation fails, the peer authentication fails.

IPSec CRL checking

If you use certificate-based authentication, you can also enable IPSec certificate revocation list (CRL) checking. By default, in Windows XP and Windows Server 2003, IPSec CRLs are automatically checked during IKE certificate authentication, but a fully successful CRL check is not required for the certificate to be accepted. However, if enhanced security is required, a fully successful CRL check is also required. CRL checking can cause delays in authentication or unnecessary failures, and some non-Microsoft PKI systems might not support it. You can disable IPSec CRL checking or specify a stronger level of IPSec CRL checking by using the Netsh IPSec context or by modifying the registry.

To disable IPSec CRL checking or specify a different level of IPSec CRL checking, use the following command:

netsh ipsec dynamic set config strongcrlcheck value={0 | 1 | 2}

To enable IPSec CRL checking through the registry
Note

- Incorrectly editing the registry might severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.

1. Under the **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\PolicyAgent**\ key, add a new Oakley subkey, with a DWORD entry named **StrongCRLCheck**.
2. Assign this entry any value from 0 through 2, where:

   - A value of **0** disables CRL checking (this is the default for Windows 2000).
   - A value of **1** causes CRL checking to be attempted and certificate validation to fail only if the certificate is revoked (this is the default for Windows XP and Windows Server 2003). Other failures that are encountered during CRL checking (such as the revocation URL not being reachable) do not cause certificate validation to fail.
   - A value of **2** enables strong CRL checking, which means that CRL checking is required and that certificate validation fails if any error is encountered during CRL processing. Set this registry value for enhanced security.

3. Do one of the following:

   - Restart the computer.
   - Stop and then restart the IPSec service by running the **net stop policyagent** and **net start policyagent** commands at the command prompt.

Note that IPSec CRL checking does not guarantee that certificate validation fails immediately when a certificate is revoked. There is a delay between the time that the revoked certificate is placed on an updated and published CRL and the time when the computer that performs the IPSec CRL checking retrieves this CRL. The computer does not retrieve a new CRL until the current CRL has expired or until the next time the CRL is published. By default, IKE requests that CryptoAPI wait 15 seconds to complete the CRL retrieval. If the CRL cannot be retrieved at that time, IKE either ignores the error (if the value of the **StrongCRLCheck** registry subkey is set to **1**, or it causes authentication to fail (if the value of **StrongCRLCheck** is set to **2**). CRLs are cached in memory and in **\Documents and Settings*UserName*\Local Settings\Temporary Internet Files** by CryptoAPI. Because CRLs persist across computer restarts, if a CRL cache problem occurs, restarting the computer does not resolve the problem.

Excluding the CA name from certificate requests

If you use certificate authentication to establish trust between IPSec peers, you can also use Windows Server 2003 to exclude CA names from certificate requests. Excluding the CA name prevents a malicious user from learning sensitive information about the trust relationships of a computer, such as the name of the company that owns the computer and the domain membership of the computer (if an internal PKI is being used). Although excluding the CA name from certificate requests enhances security, computers with multiple certificates from different roots might require the CA root names to select the correct certificate. Also, some non-Microsoft IKE implementations might not respond to a certificate request that does not include a CA name. For these reasons, excluding the CA name from certificate requests might cause IKE certificate authentication to fail in certain cases.

Certificate-to-account mapping

In Windows Server 2003, a specific group of computers can be authorized to use IPSec when either Kerberos v5 or certificates are used for IKE authentication. This capability enables much stronger peer authentication and allows IPSec to be used to restrict network access to a server. When you enable IPSec certificate-to-account mapping, the IKE protocol associates (that is, maps) a computer certificate to a computer account in an Active Directory domain or forest, and then retrieves an access token, which includes the list of computer security groups. This process ensures that the certificate offered by the IPSec peer corresponds to an active computer account in the domain, and that the certificate is one that should be used by that computer.

Certificate-to-account mapping can be used only for computer accounts that are in the same forest as the computer performing the mapping. This provides much stronger authentication than simply accepting any valid certificate chain. For example, you can use this capability to restrict access to computers that are within the same forest. Certificate-to-account mapping, however, does not ensure that a specific trusted computer is allowed IPSec access.

If the certificate-to-account mapping process is not completed properly, authentication will fail and IPSec-protected connections will be blocked.

Preshared key authentication
Preshared key authentication requires that each IKE peer use a predefined and shared key to authenticate the IKE exchange.

The following table describes the main mode messages for preshared key authentication.

Preshared Key Authentication Main Mode Messages

Main Mode Message	Sender	Payload
1	Initiator	ISAKMP header, Security Association (contains proposals)
2	Responder	ISAKMP header, Security Association (contains a selected

		proposal)
3	Initiator	ISAKMP header, Key Exchange (contains Diffie-Hellman key), Nonce
4	Responder	ISAKMP header, Key Exchange (contains Diffie-Hellman key), Nonce
5*	Initiator	ISAKMP header, Identification, Initiator Hash
6*	Responder	ISAKMP header, Identification, Responder Hash

* ISAKMP payloads of message are encrypted.

Messages 5 and 6 contain an initiator and responder hash calculated with the preshared key. Each IPSec peer authenticates the other peer's packet by decrypting and verifying the hash inside the packet (the hash inside the packet is a hash of the preshared key).

Note

- Preshared keys are easily implemented but can be compromised if they are not used correctly. Microsoft does not recommend the use of preshared key authentication because the key value is not securely stored, and it is, therefore, difficult to keep secret. Preshared key authentication is provided for interoperability purposes and compliance with RFC standards.

Quick Mode Negotiation
When main mode negotiation completes or an existing quick mode SA expires, IKE begins quick mode negotiation. During quick mode negotiation, IKE queries the Policy Agent for information required to perform the appropriate filter actions, including whether the IPSec mode is tunnel or transport, whether the protocol is ESP or AH or both, and which encryption and hashing algorithms are proposed or accepted.

Quick Mode SA negotiation
The quick mode negotiation process is implemented as defined in RFC 2409. All quick mode negotiation messages are protected with the main mode SA that was established during the main mode negotiation. Each successful quick mode negotiation establishes two quick mode SAs. One SA is inbound and the other SA is outbound.

The following table lists the quick mode messages exchanged by two IPSec peers running Windows IPSec.

Quick Mode Messages

Quick Mode Message	Sender	Payload
1*	Initiator	ISAKMP header, Security Association (contains proposals and secure traffic description)
2*	Responder	ISAKMP header, Security Association (contains a selected proposal)
3*	Initiator	ISAKMP header, Hash
4*	Responder	ISAKMP header, Notification

* ISAKMP payloads of message are encrypted.

The four quick mode messages contain the following payloads:

- **Security Association.** This payload contains a list of proposals and encryption and hashing algorithms (AH or ESP, DES or 3DES, and HMAC-MD5 or HMAC-SHA1) for securing the traffic and a description of the traffic that is protected. This description might include IP addresses, IP protocols, TCP ports, or UDP ports, and is based on the matching filter of the initiator. The Security Association in the second quick-

mode message includes a Security Association payload that contains the chosen method of securing the traffic.

- **Hash.** This payload provides verification and replay protection.
- **Notification.** This payload has a connected notify message. This message is requested and sent between two IPSec peers running Windows Server 2003. Quick mode message 4 with the Notification payload is not required by the IKE standard and is used to prevent the initiator from sending IPSec-protected packets to the responder before the responder is ready to receive them.
- For more information about the contents of ISAKMP payloads for quick mode exchanges, see RFC 2407 and RFC 2408 in the IETF RFC Database.

Windows Server 2003 IPSec supports the filter action choices listed in the following table.

Filter Action Choices

Filter Action Choices	ESP Encryption / Integrity Algorithm	AH
Encryption and integrity	3DES/HMAC-SHA1	None
Integrity only	None/HAMC-SHA1	None
Custom	DES, 3DES, or none/HMAC-MD5, HMAC-SHA1, or none	HMAC-MD5 or HMAC-SHA1

Note

- Computers running Windows Server 2003 and Windows XP support the 3DES and DES algorithms and do not require installation of additional components. However, computers running Windows 2000 must have the High Encryption Pack or Service Pack 2 (or later) installed in order to use 3DES. If a computer running Windows 2000 is assigned a policy that uses 3DES encryption, but does not have the High Encryption Pack or Service Pack 2 (or later) installed, the security method defaults to the weaker DES algorithm.

Generating and regenerating session key material
IKE generates session keys for both the inbound and outbound quick mode SAs based on the main mode shared master key and nonce material exchanged during the quick mode negotiation. Additionally, Diffie-Hellman key exchange material can also be exchanged and used to enhance the cryptographic strength of the IPSec session key.

Key Protection
The following features enhance the base prime numbers (keying material) and the strength of the keys for master and session keys.

Key Lifetimes
Key lifetimes control when a new key is generated. A key lifetime allows you to force automatic key regeneration after a specific interval of either kilobytes (KB) or seconds, whichever occurs first. This ensures that even if an attacker is able to decipher part of a communication protected by one key, new keys protect the remainder of the communication. Whenever a key lifetime is reached, the SA is also renegotiated and the key is refreshed or regenerated. For this reason, key lifetimes are also referred to as SA lifetimes. You can specify key lifetimes for the master key and for session keys.

The master key lifetime corresponds to the main mode SA lifetime created by the IKE main mode negotiation. It is configured in terms of time and number of quick mode negotiations and it applies to all security rules in the IPSec policy. The main mode SA and master key typically have a long lifetime (the default value is eight hours). Master keys are much more resource-intensive to generate than session keys because they require reauthentication and additional Diffie-Hellman exchanges.

The session keys correspond to the quick mode SAs that are used to protect program traffic. The session keys and quick mode SAs are quickly derived from the master key by IKE quick mode negotiation. Session keys are used to protect data and they have lifetimes based on the amount of data sent and the amount of time elapsed since the key started being used. Typically, session keys have shorter lifetimes than master keys. The default values are 100,000 KB (approximately 100 megabytes) or one hour.

When session keys are refreshed, new quick mode SAs replace the old ones. Quick mode SA session keys must be refreshed before either the data or time lifetime expires; otherwise, traffic is discarded. A session key will be deleted if the quick mode SAs become idle (by default, SAs become idle after five minutes). Because the master key lives longer than the session key, it allows new quick mode SAs and session keys to be established quickly. To prevent data loss, the quick mode SA session key is generated shortly before the main mode SA expires. The main mode SA and corresponding master key are not deleted when session keys and quick mode SAs are deleted, unless the specified number of quick mode SA negotiations has elapsed.

For example, if a communication takes one hour (3,600 seconds) and if you specify the minimum session key lifetime of five minutes (300 seconds), more than 12 keys are generated to complete the communication. If a 100 MB file is transferred over a fast corporate LAN using an IPSec security association with a 100 MB and one hour lifetime, at least one, if not two, session rekeys occur.

Session Key Refresh Limit
Repeated rekeying from a session key can compromise the Diffie-Hellman shared secret. Consequently, you might want to limit the number of quick mode session keys that can be derived from a main mode negotiation.

If you have enabled master key PFS, the session key limit is disregarded. Setting a session key limit to 1 is identical to enabling master key PFS. If both a master key lifetime and a session limit are specified, whichever limit is reached first causes a new main mode negotiation. By default, IPSec policy does not specify a session limit.

Diffie-Hellman Groups
Diffie-Hellman groups are used to determine the length of the base prime numbers (key material) for the Diffie-Hellman exchange. The cryptographic strength of any key derived from a Diffie-Hellman exchange depends, in part, on the strength of the Diffie-Hellman group on which the prime numbers are based. When a stronger group is used, the key that is derived from a Diffie-Hellman exchange is stronger and more difficult for an attacker to break.

IKE negotiates which group to use, ensuring that there are not any negotiation failures that result from a mismatched Diffie-Hellman group between the two peers.

If session key PFS is enabled, a new Diffie-Hellman key is negotiated during the first quick mode SA negotiation. This new key removes the dependency of the session key on the Diffie-Hellman exchange that is performed for the master key.

Both the initiator and responder must have session key PFS enabled, or negotiation fails.

The Diffie-Hellman group is the same for both the main mode and quick mode SA negotiations. When session key PFS is enabled, even though the Diffie-Hellman group is set as part of the main mode SA negotiation, it affects any rekeys during session key establishment.

Perfect Forward Secrecy
Unlike key lifetimes, PFS determines how a new key is generated, rather than when it is generated. Specifically, PFS ensures that the compromise of a single key permits access only to data that is protected by it, not necessarily to the entire communication. To achieve this, PFS ensures that a key used to protect a transmission cannot be used to generate additional keys. In addition, if the key that was used was derived from specific keying material, that material cannot be used to generate other keys.

Master key PFS
In Windows Server 2003 IPSec, you can configure the number of times quick mode SAs can be created based on a single main mode SA. If you enable master key PFS, the IKE allows only a single quick mode SA for each main mode SA. By default, master key PFS is disabled, so there is no limit to the number of quick mode SAs that can be created from one main mode SA. To derive a new quick mode SA, a new main mode negotiation is performed, which includes a new Diffie-Hellman exchange and a new authentication process.

Session key PFS
Whenever a quick mode SA requires renegotiation, IKE determines whether a session key PFS is specified in the corresponding filter rule. If it is, IKE additionally generates a new Diffie-Hellman key and exchanges it with the IKE peer during quick mode negotiation. By performing another Diffie-Hellman key exchange, IKE provides additional cryptographic strength to quick mode key generation beyond that already contributed by the main mode SA. Performing additional Diffie-Hellman exchanges requires additional computational resources and might affect IPSec performance.

IPSec Driver Processes
The IPSec driver does not participate in IP packet processing until the first time the Policy Agent informs the driver that there is an active IPSec policy. If no IPSec policy is active, the IPSec driver does not participate in inbound and outbound IP traffic processing.

IPSec Driver Responsibilities
The IPSec driver is responsible for the following:

- Maintaining the SAD and SPD.
- Checking each IP packet to determine whether it matches a policy filter. When a match is found and an SA must be created, the IPSec driver invokes the IKE module. After the IKE module completes its negotiations, an SA is returned to the IPSec driver and stored in the SAD.
- Implementing the IPSec policy as specified in the SA (for example, using a specific hashing method for outbound packets, verifying the integrity of inbound packets, and using a specific method for encrypting and decrypting).
- Tracking the length of time a specific key is in use and requesting a new key from IKE as necessary.
- Tracking the number of bytes that have been transformed (that is, hashed or encrypted) for each SA and requesting a new key from the IKE module if the byte count allowed by the SA is exceeded.
- For each secured inbound packet that contains an AH or ESP header, parsing the packet for the SPI to determine the SA.
- For each non-secured inbound packet, checking the filter list in the SPD to determine whether the packet is permitted or discarded:
    - The filter list can contain an inbound permit filter if the corresponding filter action is set to**Permit** or **Negotiate security** and either the **Accept unsecured communication, but always respond with IPSec** or **Allow unsecured communication with non IPSec-aware computer** options are enabled. The IPSec driver sends unmodified permitted packets to the TCP/IP driver for additional processing.
    - The packet is discarded either because the filter action is set to Block or the filter action is set to Negotiate security and unsecured communications are not allowed.
- Handling hardware offload of cryptographic functions by skipping cryptographic processing on packets processed by offload network adapters and managing offloaded SAs.
- Handling network layer issues such as path maximum transmission unit (PMTU) discovery.
- Creating the SPI that the responder uses to identify the appropriate SA for an inbound packet.
- Deleting expired SAs.
- Providing the implementation of the AH and ESP protocols.
  For outbound traffic that must be secured; the IPSec driver, based on the parameters of the SA, calculates and places the AH or ESP or both headers and trailer on the IP packet before sending it to the TCP/IP driver. For inbound traffic that contains an AH or ESP header, the IPSec driver processes the header and, if it is valid, sends the authenticated and decrypted packet without the AH or ESP headers and trailer back to the TCP/IP driver.

IPSec Driver Communication
The following types of communications occur between the Policy Agent, IPSec driver, and IKE module:

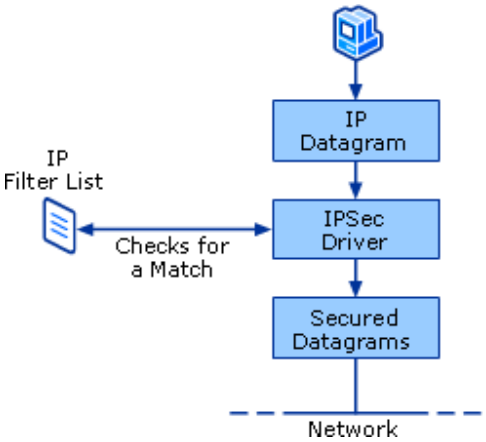
- The Policy Agent adds a set of filters to the IPSec driver. Each filter is accompanied by a policy identifier (a GUID) and an index. The index indicates to IPSec what weight to assign to the filter. A lower index indicates higher precedence and higher weight.
- When the Policy Agent deletes a set of filters, it deletes all associated outbound SAs first and then allows the inbound SAs to expire.
- The Policy Agent can receive a set of usage statistics from the IPSec driver. These statistics include the number of packets sent and received for both AH and ESP protocols, the number of SAs, the number of rekeys, and the number of bad packets.
- The IKE module adds SAs as the result of successful negotiation of keys.

- The IKE module can expire a specified SA.

Packet Processing

The IPSec driver receives the active IP filter list from the Policy Agent, as shown in the following illustration, and then attempts to match every inbound and outbound packet against the filters in the list.

IPSec Driver Matching an IP Filter List



When a packet matches a filter, the IPSec driver applies the filter action. When a packet does not match any filters, the IPSec driver passes the packet back without modification to the TCP/IP driver to be received or transmitted.

If the filter action permits transmission, the packet is received or sent with no modifications. If the action blocks transmission, the packet is discarded. If the action requires the negotiation of security, main mode and quick mode SAs are negotiated.

The negotiated quick mode SA and keys are used with both outbound and inbound processing. The IPSec driver stores all current quick mode SAs in a database. The IPSec driver uses the SPI field to match the correct SA with the correct packet.

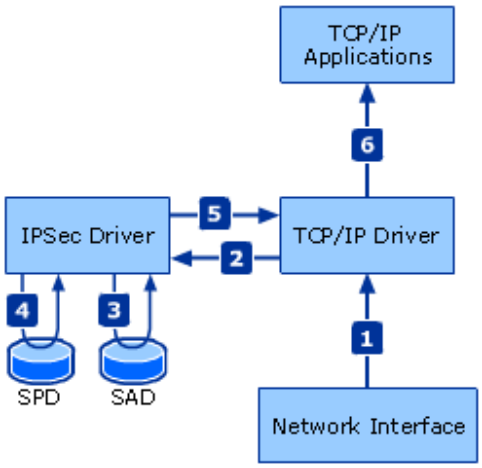
When an outbound IP packet matches the IP filter list with an action to negotiate security, the IPSec driver queues the packet and then notifies IKE, which begins security negotiations with the destination IP address of that packet. If several outbound packets are going to the same destination and match the same filter before IKE has finished the negotiation, then only the last packet sent is saved.

The following sections describe the basic inbound packet and outbound packet processing that the IPSec driver performs in transport mode.

Inbound packet processing

The following figure illustrates this process.

Basic Inbound Packet Process



Note

- The inbound packet process applies only to local host unicast traffic (traffic with the unicast destination address of the host) when there is an active IPSec policy.

Basic inbound packet processing for transport mode occurs in the following sequence:

1. IP packets are sent from the network interface to the TCP/IP driver.
2. The TCP/IP driver sends the IP packet to the IPSec driver.
3. If the inbound packet is IPSec-protected, the IPSec driver looks up the SA in the SAD.
4. If the inbound packet is not IPSec-protected, the IPSec driver checks the packet for a filter match by looking up the filters in the SPD.
5. After the IPSec-protected inbound packet is authenticated and decrypted, the AH or ESP or both headers are removed and the packet is sent to the TCP/IP driver. If a packet that is not IPSec-protected is permitted by policy, that packet is sent to the TCP/IP driver.
6. The TCP/IP driver performs IP packet processing as needed and sends the application data to the TCP/IP application.

Detailed inbound packet processing for transport mode occurs in the following sequence:

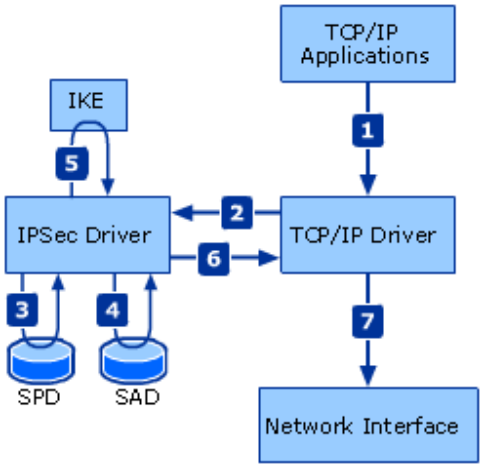
1. The TCP/IP driver sends the unicast packet to the IPSec driver.
2. If the packet is ISAKMP, the unmodified packet is sent back to the TCP/IP driver.
   Note

   - To modify the default filtering behavior for Windows Server 2003 IPSec, you can use the Netsh IPSec context or modify the registry. For more information, see “Default exemptions to IPSec filtering” later in this section.
3. If hardware offload processing was performed, the IPSec driver checks to determine whether the hardware processing was successful.
4. If the hardware processing was not successful, an event is logged and the packet is discarded.
5. The packet is parsed to determine whether an AH or ESP header or both are present.
6. If the packet does not contain an AH or ESP header, the packet is compared to the filter list for a match.
7. If a filter match is not found, the unmodified packet is sent to the TCP/IP driver.
8. If a filter match is found, the IPSec driver attempts to find an SA based on the packet contents.
9. If an SA is not found, the matching filter is checked to determine if it is an inbound permit filter.
10. If the matching filter is an inbound permit filter, the unmodified packet is sent to the TCP/IP driver.
11. If the matching filter is not an inbound permit filter, the packet is discarded.
12. If an SA is found, it is checked to determine whether it is a soft SA. A soft SA is one in which the**Negotiate security** filter action is enabled, but there is no authentication or encryption being performed because the computer with which communication occurs is not running IPSec. This process is also known as fallback to clear. Even though the packet is not being protected, an SA without an AH or ESP header is still maintained in the SAD. Soft SAs and fallback to clear are possible only when**Allow unsecured communication with non IPSec-aware computer** is selected on the **Security methods** tab in the properties of a filter action.
13. If the SA is a soft SA, the unmodified packet is sent to the TCP/IP driver.
14. If the SA is not a soft SA, the packet is discarded.
15. If the packet contains an AH or ESP header (or both), the header is parsed for the SPI.
16. The SPI is used to look up the SA in the SAD.
17. If the SA corresponding to the SPI is not found in the SAD, a Bad SPI event is logged and the packet is discarded.
18. If the SA corresponding to the SPI is found in the SAD, the current time is used to update the SA's last used time. The time is used for aging the SA.
19. The SA is checked to determine whether cryptographic processing for the SA was offloaded to hardware. For packets that have been processed by hardware offload, steps 20 and 21 are skipped.

20. The packet is authenticated or decrypted or both. This process involves verifying the HMAC in the AH or ESP header, processing the other fields in the AH and ESP headers and trailer, and decrypting the ESP payload.
21. If cryptographic processing is unsuccessful, an event is logged and the packet is discarded.
22. The AH or ESP headers and ESP trailer are removed.
23. If the SA for this packet is a tunnel SA (using either AH or ESP tunnel mode), the decapsulated packet is reinjected into the TCP/IP driver and the original packet is discarded. By reinjecting the decapsulated packet, the TCP/IP driver treats it as if it were received from the network adapter.
24. If the SA for this packet is not a tunnel SA, the IP packet, with the AH and ESP headers removed, is sent back to TCP/IP driver for additional processing.

Outbound packet processing
Basic outbound packet processing is shown in the following figure.

Basic Outbound Packet Process



Basic outbound packet processing for transport mode occurs in the following sequence:

1. Application data is sent to the TCP/IP driver from the TCP/IP application.
2. The TCP/IP driver sends an IP packet to the IPSec driver.
3. The IPSec driver checks the packet for a filter match by looking up the filters in the SPD.
4. The IPSec driver checks the packet for an active SA by looking up the SAs in the SAD. Based on the SA, the traffic is authenticated or encrypted or both.
5. If the traffic must be protected and there is not an SA, the IPSec driver requests that IKE create the appropriate SAs. The IP packet is then held until the SA is established and can be IPSec framed.
6. The IP packet is sent back to the TCP/IP driver.
7. The TCP/IP driver sends the IP packet to the network interface.

Detailed outbound packet processing for transport mode occurs in the following sequence:

1. The TCP/IP driver sends the unicast outbound packet to the IPSec driver.
2. If the packet is ISAKMP, the unmodified packet is sent back to the TCP/IP driver.
3. The IPSec driver attempts to find a filter that matches the packet. If a filter is not found, the unmodified packet is sent back to the TCP/IP driver.
4. If a filter match is found, the IPSec driver attempts to find an SA that matches the packet.
5. If an SA is not found, the filter action is checked. If the filter action is set to **Negotiate security**, the IPSec driver requests that the IKE module negotiate the appropriate SAs.
6. If the IKE negotiation is successful, the IKE module informs the IPSec driver of the new SA and the IPSec driver looks up the SA again.
7. If the IKE negotiation is not successful, the packet is discarded.

8. If the filter action is set to **Permit**, the unmodified packet is sent back to the TCP/IP driver. Otherwise, the packet is discarded.
9. If an SA is found in the SAD, the current time is used to update the SA's last used time. The time is used for aging the SA.
10. The SA is checked to determine whether it is about to expire. If the SA is about to expire, the IPSec driver informs the IKE module to initiate a quick mode or Phase 2 rekey of the quick mode SA.
11. The SA is checked to determine whether it has expired. If the SA has expired, the packet is discarded.
12. The Don't Fragment (DF) flag in the IP header of the packet is checked. If the DF flag is set to 1, the size of the IP packet with the proposed AH or ESP or both headers and trailer is calculated.
13. If the size of the IP packet with the proposed IPSec overhead is larger than the path maximum transmission unit (PMTU) for the destination IP address, the IPSec driver indicates a packet-too-large condition for the packet and the unmodified packet is sent back to the TCP/IP driver. The packet-too-large condition allows the TCP/IP driver to either adjust the PMTU for the destination or, in the case of transit traffic, inform the sending host with an Internet Control Message Protocol (ICMP) Destination Unreachable-Fragmentation Needed and DF Set message that includes the new PMTU. The packet is eventually discarded by the TCP/IP driver.
14. If the DF flag is not set to 1, or if it is set to 1 and the additional IPSec overhead is not greater than the current PMTU for the destination, blank AH or ESP both headers and trailer are constructed (based on the settings for the SA).
15. The IPSec driver checks to determine whether the hardware offload is capable of offloading the SA for this packet. If so, the IPSec driver checks to determine whether the SA for the packet was offloaded to the hardware.
16. If the SA was offloaded to the hardware, an offload status is set on the packet and the modified packet with blank AH or ESP or both headers and trailer is sent to the TCP/IP driver.
17. If the SA has not been offloaded to the hardware, the IPSec driver accesses NDIS with instructions to add the SA to the hardware offload network interface.
18. If hardware offload is not enabled or the SA has not been offloaded to the hardware, the IPSec driver performs the cryptographic processing and adds the appropriate values in the fields of the AH or ESP or both headers and trailer.
19. The IPSec driver sends the modified packet to the TCP/IP driver.

Default exemptions to IPSec filtering
In Windows Server 2003, the default filtering exemptions have been removed for Kerberos, Resource Reservation Setup Protocol (RSVP), and multicast and broadcast traffic, but remain for ISAKMP traffic, and inbound multicast and broadcast traffic.

To modify the default filtering behavior for Windows Server 2003 IPSec, you can use the Netsh IPSec context or modify the registry.

To modify the default filtering behavior using Netsh, use the following command:

netsh ipsec dynamic set config ipsecexempt value={ 0 | 1 | 2 | 3}

Depending on which exemptions you want, specify the appropriate values as follows:

- A value of **0** specifies that multicast, broadcast, RSVP, Kerberos, and ISAKMP traffic are exempt from IPSec filtering. This is the default filtering behavior for Windows 2000 (with Service Pack 3 and earlier service packs) and Windows XP.
  Note

  - Use this setting only if it is required for compatibility with Windows 2000 and Windows XP. If Kerberos traffic is exempted from filtering, an attacker can bypass other IPSec filters by using either UDP or TCP source port 88 to access any open port. Many port scan tools will not detect this because these tools do not allow setting the source port to 88 when checking for open ports.

- A value of **1** specifies that Kerberos and RSVP traffic are not exempt from IPSec filtering (multicast, broadcast, and ISAKMP traffic are exempt).
- A value of **2** specifies that multicast and broadcast traffic are not exempt from IPSec filtering (RSVP, Kerberos, and ISAKMP traffic are exempt).
- A value of **3** specifies that only ISAKMP traffic is exempt from IPSec filtering. This is the default filtering behavior for Windows Server 2003, Windows 2000 (with Service Pack 4 and later service packs) and Windows XP (with Service Pack 1 and later service packs).

If you change the value for this setting, you must restart the computer for the new value to take effect.

To modify the default filtering behavior by using the registry

1. In Regedit, under the **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IPSEC** key, add a new DWORD entry named **NoDefaultExempt**.
2. Assign this entry any value from **0** through **3**.
3. Restart the computer.

The filtering behaviors for each value are equivalent to those noted above for the **netsh ipsec dynamic set config ipsecexempt value=***x* command.

The following table summarizes the equivalent filters that are implemented if all default exemptions to IPSec filtering are enabled (that is, if **NoDefaultExempt** is **0**). When the IP address is specified, the subnet mask is 255.255.255.255. When the IP address is **Any**, the subnet mask is 0.0.0.0.

Equivalent Filters When NoDefaultExempt=0

Source Address	Destination Address	Protocol	Source Port	Destination Port	Filter Action
My IP Address	Any IP Address	UDP	Any	88	Permit
Any IP Address	My IP Address	UDP	88	Any	Permit
Any IP Address	My IP Address	UDP	Any	88	Permit
My IP Address	Any IP Address	UDP	88	Any	Permit
My IP Address	Any IP Address	TCP	Any	88	Permit
Any IP Address	My IP Address	TCP	88	Any	Permit
Any IP Address	My IP Address	TCP	Any	88	Permit
My IP Address	Any IP Address	TCP	88	Any	Permit
My IP Address	Any IP Address	UDP	500	500 [1]	Permit
Any IP Address	My IP Address	UDP	500	500	Permit
My IP Address	Peer IP Address	UDP	4500	4500 [2]	Permit
Peer IP Address	My IP Address	UDP	4500	4500	Permit
My IP Address	Any	46 (RSVP)			Permit
Any IP Address	My IP Address	46 (RSVP)			Permit
Any IP Address	<multicast> [3]				Permit
My IP Address	<multicast>				Permit
Any IP Address	<broadcast> [4]				Permit
My IP Address	<broadcast>				Permit

<All IPv6 protocol traffic> [5]					Permit

[1]In order for IPSec transport mode to be negotiated through an IPSec tunnel mode SA, ISAKMP traffic cannot be exempted if it needs to pass through the IPSec tunnel first.

[2] When IPSec NAT-T is performed, the filter exemption for UDP port 4500 is automatically generated based on the source and destination IP addresses used during the initial part of the IKE negotiation on UDP port 500. This dynamic permit filter for port 4500 is displayed in the IP Security Monitor snap-in, under **Quick Mode\Specific Filters**, and in the output for the **netsh ipsec dynamic show qmfilter** command.

[3]Multicast traffic is defined as the class D range, with a destination address range of 224.0.0.0 with a 240.0.0.0 subnet mask, which corresponds to the range of addresses from 224.0.0.0 to 239.255.255.255.

[4]Broadcast traffic is defined as a destination address of 255.255.255.255 (the limited broadcast address) or as having the host ID portion of the IP address set to all 1's (the subnet broadcast address).

[5]IPSec does not support filtering for IP version 6 (IPv6) packets, except when IPv6 packets are encapsulated with an IPv4 header.

Windows Server 2003 IPSec does not support specific filters for broadcast protocols or ports, nor does it support multicast groups, protocols, or ports. Because IPSec does not negotiate security for multicast and broadcast traffic, these types of traffic are dropped if they match a filter with a corresponding filter action to negotiate security. A filter with a source address of **Any IP Address** and a destination address of **Any IP Address** can block or permit all multicast and broadcast traffic. By default (and if **the NoDefaultExempt**registry key is set to a value of **2** or **3**), outbound multicast or broadcast traffic will be matched against a filter with a source address of **My IP Address** and a destination address of **Any IP Address**. More specific unicast IP address filters that block, permit, or negotiate security for unicast IP traffic should be configured in the same IPSec policy to achieve appropriate security.

Hardware acceleration (offloading)
Hardware acceleration is accomplished by offloading specific processing tasks that are normally completed by an operating system component to the network adapter. Some network adapters can perform IPSec cryptographic functions, such as encryption and decryption of data and the calculation and verification of message authentication codes.

When the NDIS interface binds, the offload capability of the network interface is queried. During outbound packet processing, after an SA is created a check is made to ensure that the network interface can offload cryptographic functions, support transport-over-tunnel functionality, and support IP header options. If not, the packet cannot be offloaded. A check is also made to determine whether the SA for the packet being offloaded is a soft SA. A soft SA is an SA in which no authentication or encryption is being performed because the computer with which communication occurs is not running IPSec. Because no AH or ESP headers need to be processed, hardware offloading is unnecessary.

If hardware offloading is enabled, a check is made per packet to determine whether the SA for an outbound packet has already been offloaded to the offload adapter. If so, the existing offloaded SA is used. If the SA is not yet offloaded and the offload has not previously failed for this SA, an attempt is made to offload the SA. However, the attempt to offload the SA is made asynchronously. The IPSec driver does not wait for the SA offload to be successful before continuing to process the packet. This causes the first packet to always be cryptographically processed by the IPSec driver, with the cryptographic processing of following packets occurring on the hardware offload network adapter.

Typically, IPSec network offload adapters do not accelerate the IKE negotiation. However, some SSL offload adapters might be capable of processing the IKE Diffie-Hellman calculation in hardware. To determine whether your SSL offload adapter can do so, see the manufacturer's documentation.

Windows 2000, Windows XP, and Windows Server 2003 provide hardware acceleration APIs in the Windows Driver Development Kit (DDK) as part of TCP/IP Task Offload. For more information about these APIs, see*Task Offload* on MSDN.

Network Ports and Protocols Used by IPSec

The following table lists the network ports and protocols used by IPSec.

IPSec Port and Protocol Assignments

Protocol	Protocol ID	UDP	TCP
ESP	50	N/A	N/A
AH	51	N/A	N/A
ISAKMP	N/A	500 (4500)	N/ A

The following sections describe how to configure routers, firewalls, or other filtering devices to ensure that traffic that is sent over IPSec protocols can pass through these devices. Additional considerations for IPSec NAT traversal are also described.

Firewall Filters

In order for IPSec-secured communications to take place through a firewall or other filtering device, you must configure the firewall to permit IPSec traffic on UDP source and destination port 500 (ISAKMP) and IP Protocol 50 (ESP). You might also need to configure the firewall to permit IPSec traffic on IP protocol 51 (AH) to permit troubleshooting by IPSec administrators and to allow the traffic to be inspected while it is still IPSec-encapsulated.

Ensure that the firewall filter can permit or track fragments for ISAKMP. IKE with certificate or Kerberos authentication requires ISAKMP packets to be fragmented because the ISAKMP protocol uses UDP. ISAKMP messages that are larger than the local interface MTU are automatically fragmented by IP. If only certificate authentication is used, Windows Server 2003 implements a method to avoid IKE message fragmentation. When certificate authentication is used for communication between computers running Windows Server 2003 IPSec and Windows XP IPSec or Windows 2000 IPSec, fragmentation is required.

You must also allow ISAKMP to be initiated from either a source or destination IP address. RFC 2408 specifies that the ISAKMP protocol must be able to negotiate security in either direction. Stateful filtering that allows only one computer to initiate IKE to a responder typically times out and deletes the stateful inbound filter in the firewall. As a result, IKE cannot rekey IPSec security associations, and IPSec connectivity is lost.

IPSec NAT-T

In Windows 2000 and Windows XP, if traffic between the client and a server must pass through a network address translator (NAT), then IPSec cannot secure the traffic (the IKE negotiation will fail when translated by a NAT). Windows Server 2003 provides support for version 2 of a new IETF Internet draft called IPSec NAT-T. IPSec NAT-T allows IPSec ESP packets in either transport mode or tunnel mode to pass through NATs that allow UDP traffic. In this design, IKE automatically detects NATs and uses UDP-ESP encapsulation on UDP port 4500 to enable traffic to pass through a network address translator. The Windows Server 2003 implementation of IPSec NAT-T also supports PMTU discovery for UDP-ESP encapsulation. This new functionality allows you to secure servers running Windows Server 2003, when clients are behind a network address translator. IPSec NAT-T does not support the use of AH across network address translators.

If you are using IPSec NAT-T to secure a server, it is recommended that you do not create UDP port 4500 filters in the IPSec policy that is assigned to the server. The IPSec driver recognizes UDP port 4500 traffic and detects the associated UDP-ESP quick mode SA. However, if you are using firewalls or filtering routers to filter traffic for the IPSec-secured server, then you must configure the firewalls or filtering routers to permit the UDP-ESP traffic.

To configure firewalls or filtering routers to permit traffic on UDP source and destination port 4500, use the following settings to create a filter called “Permit IPSec NAT-T ISAKMP traffic on UDP port 4500”:

- Source address = *SpecificIPAddress*
- Destination address = *SpecificIPAddress*
- Protocol = UDP
- Source port = **Any** or **4500** (The network address translator might translate source port 4500 to a different source port)
- Destination port = 4500

On computers that are running Windows 2000 or Windows XP, you can install an update package that allows L2TP/IPSec clients that are behind network address translators to use IPSec NAT-T. The NAT-T functionality provided in this update package meets the specifications of IIETF RFC 3193, “Securing L2TP using IPSec,” and version 2 of the “UDP Encapsulation of IPSec Packets” and “Negotiation of NAT-Traversal in the IKE” Internet drafts and is compatible with Windows Server 2003 IPSec NAT-T. However, using IPSec in general transport mode for NAT-T is not supported on computers running Windows XP or Windows 2000, even when this update is installed, because PMTU discovery is not provided by UDP-ESP traffic. For more information about the IPSec

update package for Windows 2000 and Windows XP, see article 818043, "L2TP/IPSec NAT-T Update for Windows XP and Windows 2000," in the Microsoft Knowledge Base.

Related Information

The following RFCs and Internet Drafts are relevant to IPSec. To find the RFCs, type the appropriate RFC number in the IETF RFC Database. To find the Internet Drafts, type the appropriate keyword in the IETF Internet Drafts Database.

- RFC 2401: Security Architecture for the Internet Protocol
- RFC 2402: IP Authentication Header
- RFC 2406: IP Encapsulating Security Payload (ESP)
- RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP)
- RFC 2409: The Internet Key Exchange (IKE)
- UDP Encapsulation of IPSec Packets (draft-ietf-ipsec-udp-encaps-02.txt)

Negotiation of NAT-Traversal in the IKE (draft-ietf-ipsec-nat-t-ike-02.txt)

IPSec Tools and Settings

IPSec Tools and Settings

In this section

Use the following Internet Protocol security (IPSec) tools and registry setting to enable, configure, and manage IPSec on a computer running the Microsoft Windows Server 2003 operating system.

IPSec Tools

To manage IPSec policy in Windows Server 2003, you use three tools: the IP Security Policy Management snap-in, the Netsh IPSec context, and the Resultant Set of Policy (RSoP) snap-in. To monitor IPSec performance, you use the IP Security Monitor snap-in. If you need additional troubleshooting functionality, you can use audit logging and detailed IKE logging in Event Viewer and Network Monitor.

As the following table shows, several of the tools that you used to create and manage IPSec policies in previous versions of the Microsoft Windows operating system have been changed, replaced, or are no longer available in Windows Server 2003.

IPSec Tool Changes in Windows Server 2003

Tool	Where It Was Previously Available	Changes in Windows Server 2003
IPSeccmd	Support Tools folder of the Microsoft Windows XP operating system CD	Not included in Windows Server 2003. Use the Netsh IPSec

		context instead.
IPSecmon.exe	Microsoft Windows 2000	Replaced by the new IP Security Monitor snap-in.
IPSecpol.exe	*Microsoft Windows 2000 Server Resource Kit*	Not included in Windows Server 2003.
Netdiag.exe	Support Tools folder of the Windows Server 2003 operating system CD	No longer includes IPSec functionality. Use the Netsh IPSec context instead.

Despite these changes, you can share IPSec policies created on computers running the Windows XP, Windows 2000, and Windows Server 2003 operating systems among any of those operating systems.

IP Security Policy Management Snap-In
Category
The IP Security Policy Management snap-in is included in Windows Server 2003.

Version compatibility
The IP Security Policy Management snap-in is available with the Microsoft Windows XP, Windows 2000 Server, and Windows Server 2003 operating systems.

To create and manage IPSec policy, you primarily use the IP Security Policy Management snap-in that is available in the Microsoft Management Console (MMC). You can use IP Security Policy Management to create, modify, and store local IPSec policies or IPSec policies based on the Active Directory directory service. Additionally, you can use IP Security Policy Management to modify IPSec policy on remote computers.

To manage local IPSec policy, you can use either IP Security Policy Management or the command line, depending on your deployment needs. It is recommended that you use IP Security Policy Management to manage IPSec policy for Active Directory-based IPSec policy,

You use different methods to access IP Security Policy Management, depending on whether the IPSec policy is Active Directory-based or local.

To access Active Directory-based IPSec policy, do either of the following on the computer from which you want to manage policy:

- Start IP Security Policy Management from the appropriate organizational unit (OU) in Active Directory (Group Policy).
- Add IP Security Policy Management for Active Directory-based IPSec policy to MMC.

To access local IPSec policy for a computer, do any of the following on the computer for which you want to manage policy:

- Start IP Security Policy Management from Local Security Policy.
- Add IP Security Policy Management for local IPSec policy to MMC.
- Add Group Policy Object Editor for local IPSec policy to MMC.

To create an IPSec policy, a user or process must be logged on to the computer as a member of the Domain Admins group or the local Administrators group or must be running with local system privileges.

For more information about Group Policy settings, see "Group Policy Settings Reference for Windows Server 2003" in the Tools and Settings Collection.

Resultant Set of Policy (RSoP) Snap-In
Category
RSoP is included as part of the operating system.

Version compatibility
Logging mode is available on Windows XP and later operating systems. Planning mode requires that you have a Windows Server 2003 computer as a domain controller.

To view IPSec policy assignments for a computer or for members of a Group Policy container, use the RSoP addition to Group Policy. You can use information about IPSec policy assignments to troubleshoot policy precedence issues and to plan your deployment.

To view IPSec policy assignments in RSoP, you must first open the RSoP MMC console, and then run a query. RSoP provides two types of queries: logging mode queries (for viewing IPSec policy assignments for a computer) and planning mode queries (for viewing IPSec policy assignments for members of a Group Policy container).

For more information about RSoP, see “How Core Group Policy Works” in Core Group Policy Technical Reference.

Logging mode queries
To view all of the IPSec policies that are assigned to an IPSec client, you run an RSoP logging mode query. When you run a logging mode query, RSoP retrieves policy information from the Windows Management Instrumentation (WMI) repository on the target computer, and then displays this information in the RSoP console. In this way, RSoP provides a view of the policy settings that are being applied to a computer at a given time.

The query results show the precedence of each IPSec policy assignment, so that you can quickly determine which IPSec policies are assigned but are not being applied and which IPSec policy is being applied. The RSoP console also displays detailed settings (the filter rules, filter actions, authentication methods, tunnel endpoints, and connection type) for the IPSec policy that is being applied.

Planning mode queries
To view all of the IPSec policies that are assigned to members of a Group Policy container, you run an RSoP planning mode query. A planning mode query can be useful if, for example, you are planning a company reorganization and you want to move computers from one OU to a new OU. By supplying the appropriate information and then running a planning mode query, you can determine which IPSec policies are assigned but are not being applied to the new OU and which IPSec policy is being applied. In this way, you can identify which policy would be applied if you were to move the computers to the new OU. As with logging mode queries, when you run a planning mode query, the RSoP console displays detailed policy settings for the IPSec policy that is being applied.

When you run a planning mode query, RSoP retrieves the names of the target user, computer, and domain controller from the WMI repository on the domain controller. WMI then uses the Group Policy Data Access Service (GPDAS) to create the policy settings that would be applied to the target computer, based on the RSoP query settings that you entered. RSoP reads the policy settings from the WMI repository on the domain controller, and then displays this information in the RSoP console user interface.

IP Security Monitor Snap-In
Category
IP Security Monitor is part of the operating system.

Version compatibility
In Windows 2000, IP Security Monitor was implemented as an executable program (IPSecmon.exe). In Windows XP and Windows Server 2003, IP Security Monitor is implemented as an MMC console. In Windows Server 2003, IP Security Monitor includes enhancements that allow you to:

- Monitor IPSec information for your local computer and for remote computers.
- View details —, including the policy name, description, date last modified, store, path, OU, and Group Policy object name — about active IPSec policies.
- View main mode and quick mode generic filters and specific filters.
- View main mode and quick mode statistics.
- Customize refresh rates and use Domain Name System (DNS) name resolution for filter and security association output.
- Search for specific main mode or quick mode filters that match any source or destination IP address, a source or destination IP address on your local computer, or a specific source or destination IP address.

Netsh.exe: Netsh
Category
Netsh is a command-line tool.

Version compatibility

Although you can use netsh commands with both Windows 2000 Server and Windows Server 2003, specific IPSec-related capabilities were added in Windows Server 2003.

The Netsh commands for IPSec provide a fully equivalent alternative to the console-based management and diagnostic capabilities provided by the IP Security Policy Management and IP Security Monitor consoles. You can use Netsh commands for IPSec to script IPSec policy creation, display details about IPSec policies, and change the IPSec configuration for troubleshooting. In addition, administering IPSec from the command line is useful when you want to extend the security and manageability of IPSec. For example, you can use Netsh commands for IPSec to enable IPSec driver event logging, set default traffic exemptions, and configure computer startup security.

For more information about using Netsh IPSec, see the "Command-Line Reference" in the Tools and Settings Collection.

Eventvwr.msc: Audit Logging in Event Viewer
Category
Event Viewer is part of the operating system.

Version compatibility
Event Viewer is available with Windows XP, Windows 2000 Server, and Windows Server 2003.

You can view the success or failure of Internet Key Exchange (IKE) negotiations in the Event Viewer security log. To view these events, enable success or failure auditing for the **Audit logon events** audit policy for your domain or local computer.

To disable audit logging, see "DisableIKEAudits" in IPSec Registry Entries.

Eventvwr.msc: Detailed IKE Logging in Event Viewer
Category
Event Viewer is part of the operating system.

Version compatibility
Event Viewer is available with Windows XP, Windows 2000 Server, and Windows Server 2003.

Enabling audit logging for IKE events and viewing the events in Event Viewer is the fastest and simplest way to troubleshoot failed main mode or quick mode negotiations. However, some scenarios might require a more detailed analysis of the IKE main mode negotiation and quick mode negotiations for troubleshooting. If the audit failure events do not provide enough information, you can enable tracing for IKE negotiations. The IKE tracing log is a very detailed log intended for troubleshooting IKE interoperability under controlled circumstances. Expert knowledge of the Internet Security Association and Key Management Protocol (ISAKMP) RFC 2408 and IKE RFC 2409 is required to interpret this log.

The IKE tracing log appears as the *systemroot*\Debug\Oakley.log file. The log has a fixed size of 50,000 lines and will overwrite as necessary. Each time the IPSec service is started, a new Oakley.log file is created and the previous version of the Oakley.log file is saved as Oakley.log.sav. When the Oakley.log file becomes full, it is saved as Oakley.log.bak, and a new Oakley.log file is created.

Many IKE negotiations can occur simultaneously. Therefore, to capture a more easily interpreted log, minimize the number of negotiations and log for as short a period of time as possible.

In Windows Server 2003, you can enable or disable the IKE tracing log dynamically while the IPSec service is running. You do this by using Netsh IPSec. For more information about how to enable or disable the IKE tracing log, see "Command-Line Reference" in the Tools and Settings Collection.

Netmon.exe: Network Monitor
Category
Network Monitor is available in Microsoft Systems Management Server or with Windows 2000 Server and Windows Server 2003.

Version compatibility
You can use Network Monitor to capture and view packets in Windows XP, Windows 2000, or Windows Server 2003.

To view IPSec and other network communication, you can install and use Network Monitor.

Note

- You can use the version of Network Monitor that is provided with Windows Server 2003 to view only the network traffic that is sent to or from the computer on which Network Monitor is installed. To view network traffic that is sent to or from another computer and is routed through your computer (using the Routing and Remote Access service), you must use the version of Network Monitor that is provided with Systems Management Server.

The version of Network Monitor that is provided with Windows Server 2003 includes parsers for the ISAKMP (IKE), Authentication Header (AH), and Encapsulating Security Payload (ESP) protocols. The Network Monitor parsers for ESP can parse inside the ESP packet only if null-encryption is being used and the full ESP packet is captured. Network Monitor cannot parse the encrypted portions of IPSec-secured ESP traffic when encryption is performed in software. However, if encryption is performed by an IPSec hardware offload network adapter, the ESP packets are decrypted when Network Monitor captures them and, as a result, they can be parsed and interpreted into the upper-layer protocols. If you need to diagnose ESP software-encrypted communication, you must disable ESP encryption and use ESP-null encryption by changing the IPSec policy on both computers.

IPSec Registry Entries

In Windows Server 2003, you can use the Netsh IPSec command-line tool to perform many of the tasks that you might have performed previously by modifying the registry.

When you enable success or failure auditing for the **Audit logon events** audit policy, IPSec records the success or failure of each main mode and quick mode negotiation and the establishment and termination of each negotiation as separate events. Keep in mind, however, that enabling this type of auditing can cause the security log to fill with IKE events. In Windows 2000, you cannot disable auditing of IKE events. In Windows Server 2003, however, you can disable auditing of IKE events by modifying the registry.

The following information is provided as a reference for use in troubleshooting or verifying that the required settings are applied. It is recommended that you do not directly edit the registry unless there is no other alternative. Modifications to the registry are not validated by the registry editor or by Windows before they are applied, and as a result, incorrect values can be stored. This can result in unrecoverable errors in the system. When possible, use Group Policy or other Windows Server 2003 tools, such as Microsoft Management Console (MMC), to accomplish tasks rather than editing the registry directly. If you must edit the registry, use extreme caution.

DisableIKEAudits
Registry path
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Audit\

Version compatibility
Windows Server 2003

When you enable success or failure auditing for the **Audit logon events** audit policy, IPSec records the success or failure of each main mode and quick mode negotiation and the establishment and termination of each negotiation as separate events. However, enabling this type of auditing can cause the security log to fill with IKE events. For example, for servers that are connected to the Internet, attacks on the IKE protocol can fill the security log with IKE events. IKE events can also fill the security log for servers that use IPSec to secure traffic to many clients. To avoid this, you can disable auditing for IKE events in the security log by modifying the registry.

To disable auditing of IKE events in the security log, you must first create the **DisableIKEAudits** key and set the registry setting to a value of **1**.After making this change to the registry, you must either restart the computer or stop and then restart the IPSec service by running the **net stop policyagent** and **net start policyagent** commands at the command prompt.

Note

- Stopping and restarting the IPSec service can disconnect all of the computers that are using IPSec from the computer on which the service is stopped and it can prevent further communication with that computer. If you restart the IPSec service immediately, the retransmit behavior of TCP might cause the TCP-based communication to resume after new IKE and IPSec Security Associations (SAs) are established.

For more information about this registry entry, see the "Registry Reference for Windows Server 2003" in the[Tools and Settings Collection](#).

Related Information

The following resources contain additional information that is relevant to this section.

- IPSec Policy Extension Technical Reference
- "Command-Line Reference for Windows Server 2003" in the Tools and Settings Collection
- "Registry Reference for Windows Server 2003" in the Tools and Settings Collection