



*Operating System*

## Using Microsoft Windows IPsec to Help Secure an Internal Corporate Network Server

**William Dixon, Program Manager, Windows IPsec, Microsoft Corporation**

**David Wong, Principal Consultant, Foundstone Strategic Security**

**Joel Scambray, Senior Director of Security, MSN, Microsoft Corporation**

---

### **Abstract**

This paper describes how to configure Microsoft® Windows® 2000 IPsec and Windows XP IPsec to help secure an internal corporate network server against network-based attacks from untrusted computers. You can significantly enhance the ability of a server to defend against such attacks by requiring IPsec-authenticated, signed, and encrypted communication between computers. This paper describes the security threats to, and the benefits of using IPsec on, an internal corporate network server and uses a scenario to describe the process of IPsec policy design for an internal corporate network. Although the focus of this paper is Windows 2000 and Windows XP IPsec, it also provides information about IPsec functionality enhancements in Windows 2000 service packs and in the Microsoft® Windows Server™ 2003 family.

Microsoft Corporation and Foundstone Strategic Strategy coauthored this paper. The Windows product development team and the Microsoft IT security group customized the IPsec policy design described in this paper for use on an operational basis, during the development of Windows 2000, and they continue to use these policies today. Foundstone evaluated the security that the IPsec policies provide against a sophisticated, untrusted attacker, in laboratory environments.

---

*The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.*

*This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.*

*Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.*

*Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.*

*The example companies, organizations, products, people and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred.*

*© 2003 Microsoft Corporation. All rights reserved.*

*Microsoft Windows NT, Windows 2000, Windows XP, and Windows Server 2003 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.*

*The names of actual companies and products mentioned herein may be the trademarks of their respective owners.*

*Additional Contributors: Jon Barrow (Senior Technical Writer, Foundstone), Jon Billow (Artist, Windows Server IT Content) Chris Black (Program Manager, Windows IPSec), David Beder (Software Test Engineer, Windows IPSec), Alexandra Coury (Technical Editor, Windows IT Content), Clifton Hall (Project Manager, Windows IT Content), Jesper Johansson (Program Manager, Windows Security Engineering), Connie La Chasse (Technical Writer, Windows IT Content), Rosanne Newland (Copy Editor, Windows IT Content), Lee Walker (Security Group Manager, Corporate Technology Group), Chris Weber (Consultant, Foundstone)*

---

# Contents

<b>Contents .....</b>	<b>i</b>
<b>Introduction.....</b>	<b>1</b>
<b>IPSec and Network-Based Attacks .....</b>	<b>3</b>
Network-Based Attacks against an Internal Corporate Network Server.....	3
Business Costs Resulting from Network-Based Attacks against an Internal Corporate Network Server .....	4
Cost Incurred by Loss of Service .....	4
Cost Incurred by Theft of Information.....	4
Cost Incurred by Compromise of Administrative Credentials on the Server.....	5
Cost Incurred by Subsequent Legal Action.....	5
Benefits of Using IPSec as a Defense against Network-Based Attacks.....	5
<b>Scenario Introduction: The Internal Corporate Network Architecture.....</b>	<b>7</b>
Threats in this Scenario .....	8
Anonymous Client Network-Based Attacks.....	9
Remote Network-Based Denial-of-Service Attacks.....	10
Authenticated Client Network-Based Attacks.....	11
Physical Intrusion .....	11
Application-Level Network-Based Attacks .....	12
How IPSec Mitigates Network Threats in this Scenario .....	12
Restricting Network Access to Trusted Computers.....	14
Providing Transparent Cryptographic Protection for Network Traffic.....	14
Mitigating Trusted Attacks.....	15
<b>Preparing to Use IPSec to Secure the Internal Corporate Network in this Scenario .....</b>	<b>17</b>
Identifying Ports to Block or Secure with IPSec Filters.....	17
Table 1 Sample Netstat Output for a Server .....	18
Considerations for NetBT and SMB .....	18
Considerations for RPC.....	19
Overview of IPSec Policy Concepts .....	20
IKE Negotiation Process .....	21
Main Mode Negotiation .....	22
Quick Mode Negotiation .....	22

---

IKE Main Mode SAs and IPsec SAs .....	22
IPsec Policy Filters .....	23
Security Methods.....	24
IPsec Encapsulation Modes and Protocol Wire Formats .....	24
IPsec Encapsulation Modes .....	24
IPsec Protocol Wire Formats.....	24
IKE Authentication.....	26
IKE Authentication Process.....	26
IKE Authentication Methods.....	26
IPsec CRL Checking.....	27
IKE Authentication Methods and Security Method Preference Order.....	29
Security Negotiation Options.....	29
Fall Back to Clear .....	29
Inbound Passthrough .....	30
Session Key and Master Key PFS .....	31
<b>Scenario Details: IPsec Policy Design.....</b>	<b>32</b>
Tasks for Defining IPsec Policy in this Scenario.....	34
Configuring Firewalls and Filtering Routers to Permit IPsec-Secured Traffic .....	34
Removing the Default Filtering Exemption for Kerberos and RSVP Traffic.....	34
Configuring Additional Filtering Exemptions .....	35
Designing an IPsec Policy for the Server (CORPSRV).....	36
CORPSRV Policy General Settings .....	36
CORPSRV Policy Security Settings.....	37
CORPSRV Policy Rules.....	38
Table 2 Summary of CORPSRV Policy Rules .....	39
Designing IPsec Policies for the Client (CORPCLI) .....	52
CORPCLI Policy 1 Overview.....	52
CORPCLI Policy 1 General Settings.....	52
CORPCLI Policy 1 Rules.....	54
Table 3 Summary of CORPCLI Policy 1 Rules .....	54
CORPCLI Policy 2 Overview.....	57
CORPCLI Policy 2 General Settings.....	57

CORPCLI Policy 2 (Default Response Policy) Rules.....	58
Table 4 Summary of CORPCLI Policy 2 Rules .....	58
Modifying IPsec Policy Design in this Scenario .....	62
Modifying CORPSRV and CORPCLI Policy Design to Exempt All ICMP Traffic from IPsec Filtering .....	62
Modifying Rule 7 for CORPSRV Policy .....	63
Creating a New Rule (Rule 4) for CORPCLI Policy 1 .....	63
Confirming ICMP Permit Operation for Updated CORPSRV and CORPCLI Policies .....	64
Modifying CORPSRV and WEBSRV Policy Design to Defend against Perimeter Network Server (WEBSRV) Compromise .....	65
Modifying CORPSRV Policy Rule 1 .....	66
Table 5 Modifications to CORPSRV Policy Rule 1 .....	66
Modifying the WEBSRV Policy Rule .....	67
Table 6 Modifications to the WEBSRV Policy Rule.....	68
Additional IPsec Policy Modifications .....	70
Customized Rules for IPsec Policies Assigned to Microsoft Servers and Clients.....	70
Fall Back to Clear and Inbound Passthrough Configuration for Initial Deployment.....	71
Fall Back to Clear and Inbound Passthrough Configuration for Final Deployment.....	72
<b>Managing IPsec Policies: Operational Considerations.....</b>	<b>73</b>
Understanding the Impact of IPsec on CPU Performance and Network Utilization.....	73
Understanding Specific Factors that Contribute to IPsec Performance Overhead .....	73
Using IPsec Hardware Offload Network Adapters.....	74
Understanding the Impact of IPsec on Users and Applications .....	75
Administrative Experience and Security Group Membership for Managing IPsec Policy.....	76
Managing Active Directory-Based and Local IPsec Policy .....	76
Storing Active Directory-Based and Local IPsec Policy .....	77
Using IP Security Policy Management to Create and Modify IPsec Policy .....	77
Using Command-Line Tools to Script IPsec Policy Creation .....	78
Assigning and Distributing IPsec Policy .....	79
Delegating Permissions to Modify Active Directory-Based IPsec Policy.....	80
Preventing Unwanted Active Directory-Based IPsec Policy from Being Assigned to Computers.....	81
Customizing Active Directory-Based IPsec Policy for a Specific Server .....	82

---

Providing Backup Security During the Loss of Active Directory-Based IPsec Policy.....	82
IPsec Policy Compatibility Considerations .....	83
Remotely Managing IPsec Policy .....	83
Considerations for Backing Up and Restoring IPsec Policy .....	84
Exporting and Importing IPsec Policies to Back Up and Restore IPsec Policies .....	85
Testing and Monitoring Successful IPsec Operation .....	87
Gathering IPsec Data for Troubleshooting .....	87
Verifying That IPsec Policy Does Not Block Traffic Required by a Server.....	88
Verifying IPsec-Secured Connectivity .....	89
Viewing IPsec Events .....	89
Enabling Security Log Audit Events for IKE Negotiation.....	90
Enabling Logging for the IPsec Driver .....	91
Evaluating Bad SPI Events .....	93
Viewing IPsec and Other Network Communication.....	93
Table 7 Sample Network Monitor Capture for Packets Sent between CORPSRV and PNDC .....	94
Managing and Monitoring IPsec on Computers Running Windows XP .....	94
Viewing Network and IPsec-Related Information .....	95
Enabling Detailed Tracing for IKE Negotiations .....	96
Monitoring IPsec Status Information.....	97
<b>Configuring IPsec Policies: Security Considerations .....</b>	<b>98</b>
Security During Computer Startup .....	98
Impact of Group Policy Security Settings on IPsec.....	98
Considerations for Crossing Security Boundaries with IPsec-Secured Traffic .....	100
Creating Firewall Filters to Permit ISAKMP, AH, and ESP Traffic .....	100
Considerations for NAT Traversal .....	101
Considerations for Securing Traffic Required for Remote Management and Monitoring of IPsec Policy .....	102
Security Considerations for RPC and SMB Remote Management Traffic.....	102
Security Considerations for RDP Remote Management and Monitoring Traffic.....	103
Authentication Considerations for Remotely Managing IPsec Policy.....	104
Considerations for Different Trust Environments .....	104
Potential Issues for Perimeter Network Security .....	105

---

Security Risks of Enabling the Default Response Rule .....	105
Security Risks of Receiving Unsecured ICMP Protocol Traffic.....	107
Considerations for IKE and IPSec.....	109
<b>Resources .....</b>	<b>110</b>
Network Threats and Attacks .....	110
Windows 2000 IPSec (General).....	110
Windows Server 2003 IPSec (General) .....	111
Security for Windows Active Directory (includes information about IPSec).....	111
Security for Windows (General) .....	111
Networking for Windows 2000.....	112
Microsoft Knowledge Base Articles .....	112
Microsoft Downloads .....	113
IPSec Hardware Offload Adapters and Hardware Compatibility.....	113



---

## Introduction

The "2002 Computer Crime and Security Survey" conducted by the Computer Security Institute (CSI) and the Federal Bureau of Investigation (FBI) reported a total of \$456 million in financial losses due to computer crime and other security breaches, in contrast to only \$100 million in losses reported in 1997, despite only a five percent increase in survey respondents. In addition, although 89 percent of the respondents had firewalls and 60 percent of the respondents used intrusion detection systems (IDS), 40 percent of the respondents reported penetration from the Internet. Average losses to computer crime were highest in the following areas: theft of proprietary information (\$6.6 million), financial fraud (\$4.6 million), and telecom eavesdropping (\$1.2 million).

The CSI survey results provide many examples in which malicious users and attackers used compromised user credentials to access internal databases over the network. To provide general defensive measures against such attacks, you can ensure that all computers have the latest security patches installed, implement communications filtering functionality on all intervening devices (routers and firewalls) and on the computers themselves, disable unused services on computers, implement mandatory authentication and auditing of access to all resources, and implement proper authorization controls. For more information about the CSI survey, see the [CSI Web site](http://go.microsoft.com/fwlink/?LinkId=18767), at <http://go.microsoft.com/fwlink/?LinkId=18767>.

To provide defense-in-depth, you must take defensive measures at the network level (that is, on firewalls and routers) and configure individual computers with strong security measures. When you take defensive measures at the network level and configure individual computers with strong security measures, you can provide security when external measures fail or are misconfigured, perimeter security is ineffective against internal attacks, and applications are not as secure as necessary.

This paper provides example IPSec policies to show how you can use Windows 2000 IPSec and Windows XP IPSec to help provide defense-in-depth to an internal corporate network server in your organization through a combination of host-based IPSec packet filtering and the enforcement of trusted access (that is, allowing only trusted computers to access a server). You can use the same example policy design on computers running Windows Server 2003. Windows Server 2003 provides additional IPSec features to enhance security and additional Group Policy and auditing features to help you simplify the deployment of IPSec.

The example IPSec policies provided in this paper were verified in a laboratory network and in production on internal Microsoft network servers. These policies were customized as required for Microsoft internal networks. (For more information, see "[Customized Rules for IPSec Policies Assigned to Microsoft Servers and Clients](#)," later in this paper.) Additionally, Foundstone conducted network security tests from several attack points in a laboratory environment to assess the protection that these IPSec policies provided for a server running Windows 2000 Server. During their penetration testing, Foundstone did not find a method to compromise the security that the IPSec policy provided for the server.

---

**Important** The example IPsec policies provided in this paper do not guarantee the security of a server. The security testing that Foundstone conducted focused only on network-based attacks from untrusted computers. In addition, although this paper notes the risk of attacks that are possible when IPsec is not used to help secure traffic, Foundstone did not test such attacks.

---

Windows IPsec is not a full-featured firewall. It is designed to classify and help secure TCP/IP traffic based on the settings that you configure in an IPsec policy. This paper describes how you can customize the settings of the example policies to meet the security requirements of your network environment. For more information, see "[Modifying IPsec Policy Designs in this Scenario](#)" and "[Additional IPsec Policy Modifications](#)," later in this paper.

Before you deploy IPsec policies in a production environment, you should ensure that administrators with a strong foundation in IPsec evaluate the policies for security, operational stability, and supportability. Additionally, it is highly recommended that you perform additional procedures to enhance the security of servers in your network (procedures such as installing the latest security patches, verifying that the patches are properly installed and functioning, and disabling services that you can verify are not required for the servers to function properly). You can use such procedures to help defend your network against attacks that are mounted from trusted computers that use IPsec-secured traffic and against attacks that use unsecured traffic. References to additional sources of information about securing servers are provided in "[Resources](#)," later in this paper.

---

**Note** Portions of IPsec and related services for Windows 2000, Windows XP, and Windows Server 2003 were jointly developed by Microsoft and Cisco Systems, Inc.

---

---

## IPSec and Network-Based Attacks

Network-based attacks, such as denial-of-service attacks, data corruption, and data theft are made through network-aware services. Clients and servers running Windows provide a number of services that are network-aware. Although it is a good security practice to disable unused services, many network-aware services cannot be completely disabled, for example, services that use the network to maintain domain membership; allow remote management; authenticate domain users; or provide network-based applications, such as file sharing, Web hosting, and database storage.

TCP and UDP do not provide security against network-based attacks. Security properties such as data authentication, authorization, data confidentiality, and data integrity are typically provided by applications at the security services layer. For example, an Internet e-mail program might use Secure Sockets Layer (SSL) to help protect data that is transmitted during user account logons and during the upload and download of e-mail messages from a mail server. Alternatively, an Internet e-mail program might not include any protection, or it might not include strong enough protection against certain network-based attacks. If the protection is not strong enough, any attacker with the ability to eavesdrop over a network path that is used by an Internet e-mail program can steal passwords, e-mail, and e-mail attachments. To address weaknesses in application design or configuration, you can use IPSec. For example, you can configure IPSec policies to help secure network communications between clients and their mail servers or between mail servers. This paper describes how you can use IPSec to help secure all network access to a server, not just to the ports and protocols that e-mail communications with the server use. However, if other network services are attacked successfully, the security of the user account and e-mail program is lost.

### Network-Based Attacks against an Internal Corporate Network Server

An attacker can mount either a passive attack or an active attack against an internal corporate network server. A passive attack involves capturing communications between the internal corporate network server and other computers. An active attack involves connecting directly to the internal corporate network server or modifying communications.

Network-based attacks against the internal corporate network server might originate from any of the following sources:

- Within the internal corporate network (for example, an attacker might use a physical port in a corporate building to make a connection to the internal corporate network, or malicious users might use computers on the internal corporate network).
- Buildings outside of the corporation, but within eavesdropping range of the corporate wireless LAN (if the wireless LAN lacks sufficient encryption or authentication mechanisms to secure traffic).
- The Internet (if filtering for inbound traffic to the perimeter network is misconfigured or the perimeter network is otherwise compromised).

Finally, the attacker might have or gain knowledge of legitimate, internal user account passwords before attempting to access an internal corporate network server.

---

**Note** When you develop a network security strategy, create a threat model that you can use to estimate the type and the severity of threats that your IT infrastructure faces. Microsoft uses the STRIDE threat model, which categorizes threats according to the following types: spoofing, tampering, repudiation, information disclosure, denial-of-service, and elevation-of-privilege. The STRIDE threat model is often used to estimate the type and severity of threats to a server that is running a specific program (for example, Microsoft Exchange Server or Microsoft Commerce Server), that is performing a specific role (for example, a file and print server), or that is running a specific program and performing a specific role. This paper describes types of threats that are specific to communication between clients and servers and that you can use IPSec to mitigate. For information about STRIDE, see Chapter 2, “Defining the Security Landscape,” in [Microsoft Solution for Securing Windows 2000 Server](http://go.microsoft.com/fwlink/?LinkId=18758), at <http://go.microsoft.com/fwlink/?LinkId=18758>.

---

A successful network-based attack on any application or service running on an internal corporate network server can result in any of the following threats:

- Denial of service of the application, the service, or the network
- Data corruption
- Data theft
- User credential theft
- Administrative control of the server
- Administrative control of other computers and the network

### **Business Costs Resulting from Network-Based Attacks against an Internal Corporate Network Server**

The cost of a successful attack on an internal corporate network server can vary greatly, depending on the type of data that is stored on the server and the role that the server performs in your organization. When determining the total cost, consider the costs incurred by loss of service, theft of information, compromise of administrative credentials on the server, and subsequent legal action that might be required. The total costs of an internal server compromise very likely justify many security measures.

#### **Cost Incurred by Loss of Service**

To determine the total cost incurred by the loss of service on a network server, add the individual cost of each of the following:

- Incident response time required by support personnel
- Lost revenue due to application service interruption
- Lost internal productivity

#### **Cost Incurred by Theft of Information**

To determine the total cost incurred by the theft of information from an internal network server, add the individual cost of each of the following:

- Loss of intellectual property required to develop information

- 
- Loss of future revenue across all products due to customer mistrust, if the theft is publicized
  - Loss in market value due to investor mistrust, if the theft is publicized
  - Internal response time required by marketing and development
  - Loss of revenue opportunity due to internal response effort
  - Time required to mitigate the malicious use of information against business, employees, or customers by outsiders

### **Cost Incurred by Compromise of Administrative Credentials on the Server**

To determine the total cost incurred by the compromise of administrative credentials on an internal network server, add the individual cost of each of the following:

- Internal effort required to respond to the attack and replace the server
- Internal mitigation of attacks on other computers that were made possible by the compromise of administrative credentials on the server

### **Cost Incurred by Subsequent Legal Action**

To determine the total cost incurred by the requirement for subsequent legal action, add the individual cost of each of the following:

- Cost of legal action if the attacker can be identified, but your company loses the court decision
- Cost of legal action if the attacker can be identified and your company wins the court decision, but the defendant cannot pay the court-awarded damages

### **Benefits of Using IPSec as a Defense against Network-Based Attacks**

You can use IPSec as one measure to defend against network-based attacks from untrusted computers. IPSec is intended for use in environments where untrusted network access and attacks on network traffic are a realistic threat. IPSec is also useful as a means of auditing communication, to assist in network security investigations.

IPSec is a mature, state-of-the-art, Internet Engineering Task Force (IETF)-designed security protocol that provides defense-in-depth against network-based attacks from untrusted computers. IPSec provides data confidentiality, data integrity, data origin authentication, and anti-replay for unicast IP packets sent between trusted hosts. The strong, cryptographic-based authentication and encryption that IPSec provides is especially useful for securing traffic that must traverse untrusted network paths, such as on a large corporate intranet or the Internet. IPSec is also especially useful for securing traffic that uses protocols and applications that do not provide sufficient security for communications.

The following section introduces an example network scenario, five types of network threats that the internal corporate network server in the scenario faces, and how you can use IPSec to mitigate many of these threats. Later, this paper uses the scenario details to recommend IPSec policy designs that you can use to help secure communication paths between the internal corporate network server and clients in the same network and computers in a perimeter network (also known as DMZ, demilitarized zone, and screened subnet).

---

When you use IPSec as documented in this paper to help secure an internal corporate network server, the risk of a network attack against that server is reduced in the following ways:

- Attack opportunity is reduced to trusted computers. By granting inbound network access only to a group of trusted computers that you explicitly specify, you can substantially reduce the risk of attack from compromised user credentials and other unmanaged or untrusted computers on the network.
- Attack opportunity is reduced to only the communication paths, protocols, and ports that you explicitly specify in an IPSec policy.
- Sophisticated attacks based on capturing or manipulating network traffic are greatly reduced through the cryptographic protection that IPSec provides.

By providing these benefits, Windows IPSec can help you achieve a high level of security for the traffic and the servers in your network.

---

**Note** IPSec allows you to maintain a high level of control over communications security, but at the cost of additional administrative configuration. As an administrative tool, IPSec requires substantial understanding of TCP/IP networking and of IPSec itself. To effectively configure, manage, and troubleshoot IPSec policies, you should have an advanced knowledge of IP networking, experience administering firewalls and filtering routers, and experience using network tools.

---

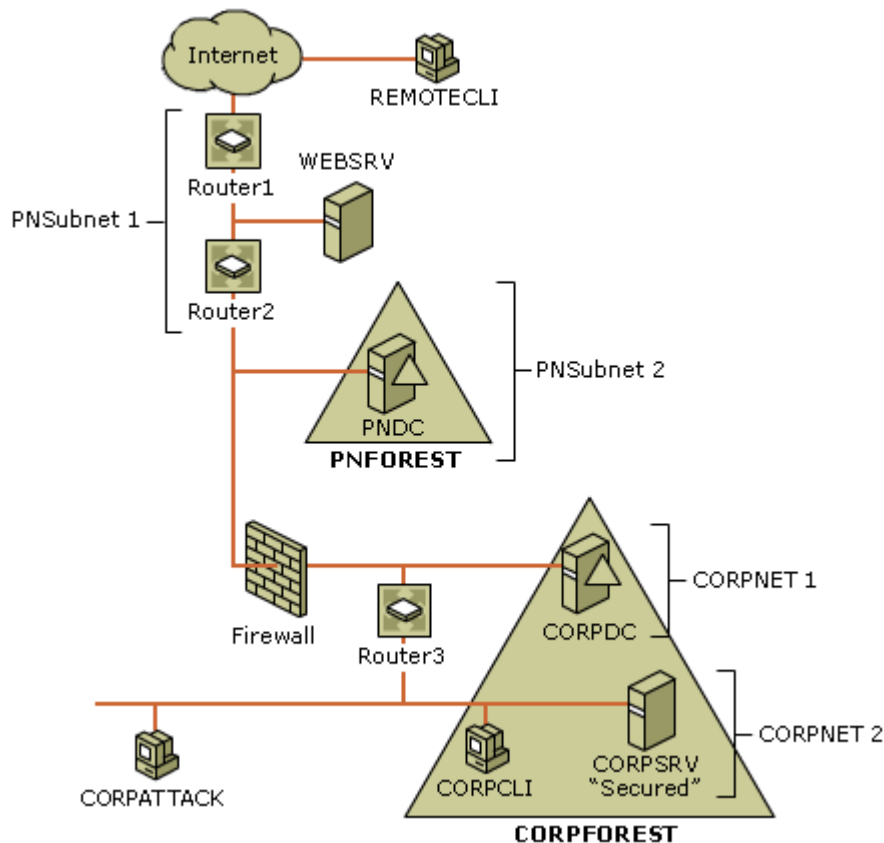
---

## Scenario Introduction: The Internal Corporate Network Architecture

The figure below shows the network architecture of an organization that can use IPSec to help secure a server on an internal corporate network that is not directly connected to the Internet. The organization might be a corporation, government organization, or university. The server is a member of an Active Directory security domain. Clients that are running Windows 2000 Professional or Windows XP Professional and that are also members of the domain or a trusted domain can access the server by using IPSec-secured communication.

You can use servers running Windows Server 2003 in this scenario (either as domain controllers or as IPSec-secured servers) but computers running Windows Server 2003 were not included in the testing conducted by Foundstone. Because Microsoft does not provide IPSec in Windows 95, Windows 98, Windows 98 SE, Microsoft® Windows NT® 4.0, or Windows Millennium Edition, computers running these operating systems cannot be used in this scenario. IPSec is provided in Windows XP Home Edition, but because this operating system does not support domain membership, computers running Windows XP Home Edition are not used in this scenario.

In the following figure, the IPSec-secured server is designated as CORPSRV.



As this figure shows, CORPSRV is a member of a domain in an internal Windows 2000 forest called CORPFOREST. CORPDC is a representative domain controller in that forest. CORPCL, a client, is also a member of a domain in the forest, but CORPATTACK is not. If there are other

---

Active Directory domains that are not mutually trusted, members of the untrusted domains are denied access to CORPSRV.

CORPATTACK is a computer that is connected to an internal network access point and that is being controlled by an attacker. The network access point might be located on the same LAN as CORPCLI, in a branch office, or in a network-connected subsidiary office in a distant geographical area. The attacker who uses CORPATTACK is assumed to have sophisticated network capture software, reconnaissance tools, and attack tools.

REMOTECLI is a remote access client that is a member of a trusted internal domain in CORPFOREST and that uses a virtual private network (VPN) connection to access the internal corporate network. Because IPsec policies are best distributed by using Group Policy to configure Active Directory domains and organizational units (OUs), REMOTECLI and all other clients are members of trusted internal domains. If they were not members of trusted internal domains, it would be difficult to manage IPsec policy and trust relationships for these clients.

---

**Note** Windows-based VPN clients can use Point-to-Point Tunneling Protocol (PPTP) or Layer 2 Tunneling Protocol (L2TP) with IPsec (L2TP/IPsec) to make IPsec-secured, end-to-end connections through VPN tunnels. If remote access clients in your organization require access to an internal corporate network server, verify that your VPN client software supports the use of IPsec-secured, end-to-end connections through VPN tunnels.

---

There are many reasons to use IPsec to help secure CORPSRV in this scenario. IPsec can provide authenticated access control for CORPSRV in the event that an internal corporate network (CORPNET 1 or CORPNET 2) cannot be physically secured. For example, a corporate network with wireless LAN connectivity might not be able to authenticate all computers that connect to it. If CORPSRV is located in an Internet-facing university network, you can use IPsec to restrict server access to faculty and staff members only and to prevent students and other Internet users from accessing the server through any open TCP or UDP ports.

In this scenario, CORPSRV faces several types of network threats. The following sections describe the network threats faced by CORPSRV and how you can use IPsec to mitigate these threats.

### Threats in this Scenario

In this scenario, CORPSRV faces the following five types of threats:

- Network-based anonymous client attacks on any network-aware service
- Remote network-based denial-of-service attacks on any network-aware service
- Network-based authenticated client attacks on any network-aware service
- Physical intrusion
- Network-based analytical attacks due to inadequate application-level security for network traffic

The risks that are presented by each of these threats vary, depending on the approach taken by the attacker and on the defensive measures that you take. For example, poor perimeter network defenses increase the risk of an anonymous client network-based attack. However, if the server

---

is protected by perimeter filtering but is stored in an unlocked, unmonitored room near a busy hallway, then the risk of physical intrusion is greater than the risk of an anonymous client network-based attack.

### **Anonymous Client Network-Based Attacks**

Anonymous client network-based attacks pose the greatest threat for those tasked with protecting IT resources. In the worst case, an attacker can penetrate the outer routers and firewall of an internal corporate network and gain access to the relatively unprotected internal computers, including an internal corporate network server such as CORPSRV.

Attackers can use many publicly available tools to exploit known vulnerabilities or misconfigurations of applications and operating systems that are running on computers in an internal corporate network. By doing so, the attackers can obtain sensitive and potentially valuable information, such as legitimate user credentials, customer credit card information, corporate user names and passwords, or even intellectual property that can be sold to competitors. After initially penetrating the network, attackers might try to obtain multiple, legitimate, trusted user credentials to mask their presence in the activity of other trusted users.

One anonymous client network-based attack is an attack that can occur if an attacker can collect and analyze data offline; the attacker can then effectively use password-guessing programs. Any network protocol that uses a password-based authentication method is at risk of passive interception. Password information for user accounts in Windows is stored in the Security Accounts Manager (SAM) database of the registry on workstations and member servers. Therefore, the primary passive attack methods that attackers use to obtain user passwords are network sniffing and attacking the SAM database.

Another type of anonymous client network-based attack is the type of attack that CORPATTACK can initiate if CORPATTACK is an unauthorized DHCP server. For example, an attacker might configure the DHCP service so that CORPATTACK is the default gateway for CORPSRV. As a result, an attacker can use CORPATTACK to perform man-in-the-middle attacks to inspect or modify all unsecured CORPSRV traffic. An unauthorized DHCP server presents an even greater risk to CORPSRV than external attacks. Internal, unauthorized DHCP servers are already behind external screening routers and firewalls, and therefore, an attacker can use an internal, unauthorized DHCP server to approach an internal corporate network server directly. In the case of large corporations or universities with less restrictive Internet access policies, it is possible for unauthorized DHCP servers to exist on internal networks and to be controlled by unauthorized external attackers.

To help protect against anonymous client network-based attacks, defense-in-depth security is critical. As mentioned earlier, not only should you take defensive measures at the network level (that is, through appropriate configuration of firewalls and routers), but you should also configure individual computers with strong security measures such as IPSec, in case external measures fail or become misconfigured during regular network maintenance or upgrades.

To defend against anonymous client network-based attacks, take the following countermeasures:

- Install the latest security patches on all computers, and then verify that the patches are properly installed and functioning.
- Implement communications filtering functionality on all intervening devices (routers and

---

firewalls) and on computers.

- Disable services that you can verify are not being used on computers (for example, Internet Information Services, or IIS).
- Require mandatory authentication and authorization for access to resources from the network. To do so, you might need to configure security settings in many different services and applications and use IPSec. You can configure these settings as part of your procedures to enhance security and then use a security template to apply them quickly. For more information, see [Hardening Systems and Servers Checklists and Guides](http://go.microsoft.com/fwlink/?LinkId=19321), at <http://go.microsoft.com/fwlink/?LinkId=19321>.

### **Remote Network-Based Denial-of-Service Attacks**

Network-based denial-of-service attacks are one of the easiest types of attacks. It often requires little effort to fully consume resources on the target computer, to starve the target computer of resources, or to cause critical services to fail or malfunction. Internal corporate networks typically do not have internal filtering defenses against common denial-of-service attacks, such as flooding.

In the scenario described in this paper, a network-based denial-of-service attack can occur when an attacker on the Internet has control over the internal corporate network computer CORPATTACK. An attacker who has control over CORPATTACK can cause considerable damage to CORPSRV and other internal computers. The attacker might control CORPATTACK by using a Trojan, a worm, or a virus to make an Internet connection from CORPATTACK. The traffic that is sent over the Internet connection will appear to be typical Internet traffic. The attacker can then use CORPATTACK to conduct reconnaissance and gather information about the internal network, services, and user accounts, and to download attack tools and receive control instructions from the attacker.

To defend against remote network-based denial-of-service attacks, take the following countermeasures:

- Enforce strict guidelines for quickly updating computers with operating system and application hotfixes and service pack updates. This countermeasure is a highly effective way to prevent new denial-of-service attacks.
- Implement configuration procedures that are recommended by Microsoft to enhance the security of both clients and servers. For more information, see [Hardening Systems and Servers Checklists and Guides](http://go.microsoft.com/fwlink/?LinkId=19321), at <http://go.microsoft.com/fwlink/?LinkId=19321>.
- Implement communications filtering functionality (if available) on all network devices and appropriate computers.
- Consider using IPSec or other techniques for controlling or auditing internal communication patterns to detect or block unusual network usage.
- Configure the TCP/IP protocol stack on potentially vulnerable Windows-based computers to resist common resource-starvation attacks. This is particularly important if a computer can be accessed through the Internet. For more information, see article 315669, "HOW TO: Harden the TCP/IP Stack Against Denial of Service Attacks in Windows 2000," in the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

---

**Note** IPsec can help prevent remote network-based denial-of-service attacks against all network-aware services. However, IPsec must also defend itself against many types of denial-of-service attacks. If the Internet Key Exchange (IKE) protocol is targeted by a denial-of-service attack, it responds by attempting to preserve existing IPsec security associations (SAs) but dropping requests for new SAs until the denial-of-service attack lessens or stops.

---

### **Authenticated Client Network-Based Attacks**

Most computer crime is perpetrated by insiders (malicious users on authenticated clients). Although this situation is changing due to greater Internet connectivity, malicious users are still in the closest proximity to valuable, sensitive data. For example, corporate employees who have administrative credentials possess the credentials to access highly sensitive elements of the company's intellectual capital.

Authenticated client network-based attacks typically result from security misconfigurations. Often, organizations that implement strong authentication mechanisms do not consider the corresponding need to require authorization to access resources after a user's identity is established. As a result, after users successfully authenticate to the network, they have unrestricted access to too many resources on that network. Such misconfiguration is common in large environments where IT cost savings have reduced the number of IT staff, thereby placing more responsibility for security administration on end users. Often, a single Windows 2000 Active Directory forest is used for all users and computers. In this environment, a single user or computer can authenticate to a domain within the forest and potentially gain access to any resources in the forest that do not have appropriate permissions set.

Both malicious users (users who have access to a system and pose a security threat to it) and attackers (users who do not necessarily have access to the system yet pose a security threat to it) can exploit these situations. For example, authenticated client network-based attacks can originate from a client such as CORPCLI that is a member of the same forest (in the scenario described in this paper, CORPFOREST) as CORPSRV. As an authenticated member of the forest, CORPCLI can use the shared Kerberos version 5 authentication infrastructure to access resources on the server. In addition, an attacker on CORPATTACK can obtain the credentials of a domain user from one of the domains in the forest and use the compromised identity to access resources on CORPSRV.

It is difficult to defend against attacks from malicious users. Doing so requires close analysis and an understanding of the specific resources that each user should be permitted to access. It also requires the careful configuration of authorization through the use of access control lists (ACLs) to explicitly implement permissions. However, auditing network communication is often an effective method for ensuring that system administrators and other users are not abusing their credentials. Audit logs also assist in investigations after an attack is discovered.

### **Physical Intrusion**

No network-layer security measure can defend a computer from an intruder with physical access to that computer. In fact, it is extremely unlikely that any software-based security measures employed on the computer can prevent a physical intrusion. Therefore, an intruder who can physically turn off CORPSRV and remove the server's hard disk might eventually gain complete

---

access to any data stored on the computer (the length of time required to access the data depends on the tools used). Additionally, an intruder might be able to use the identity of a legitimate local or domain computer or user on the network on which CORPSRV resides.

To defend against physical intrusion, take the following measures:

- Store servers in a locked room with controlled access.
- Implement hardware-level security features, such as cable locks and BIOS passwords.
- Disable removable media drives.
- Lock storage areas used for backups.
- Consider using the Windows 2000 Encrypting File System (EFS) to help protect user data.

For more information about the threat of physical intrusion and other threats to security, see [The Ten Immutable Laws of Security](http://go.microsoft.com/fwlink/?LinkId=18751), at <http://go.microsoft.com/fwlink/?LinkId=18751>.

### **Application-Level Network-Based Attacks**

Network-layer security controls access to applications. If an attacker accesses an application running on CORPSRV in a way that is consistent with network-layer security features (that is, if an attacker meets network-layer security requirements and is therefore able to pass that layer), then all of the data security on CORPSRV depends on application-level security features, the integrity of the application design, and the implementation of the application itself. Application-level network-based attacks are becoming the dominant type of attack. The continued prevalence of Web server compromises on the Internet testifies to the severity of application-level attacks today and the difficulty of preventing them in the future.

For example, the server message block (SMB) protocol and Remote Procedure Call (RPC) protocol are used extensively for Windows file sharing, printing, and Active Directory replication. Although SMB provides digital signing, it does not provide encryption. As a result, if data that is sent over the SMB protocol is captured, an attacker can also interpret the data.

An SMB and an RPC security issue was found and fixed in Windows 2000 and in Windows XP. For more information see “MS02-070: Security Update for Microsoft Windows: Flaw in SMB Signing May Permit Group Policy to Be Modified (329170)” and “MS03-026: Security Update for Microsoft Windows: Buffer Overrun In RPC Interface Could Allow Code Execution (823890),” in [Security Bulletins](http://go.microsoft.com/fwlink/?LinkId=19322), at <http://go.microsoft.com/fwlink/?LinkId=19322>.

If the servers are secured by IPSec, an attacker cannot use an untrusted computer to exploit SMB or RPC vulnerabilities because an untrusted computer cannot access the servers. To exploit these vulnerabilities, an attacker would instead have to mount attacks from a trusted computer that is allowed IPSec-secured access to the servers.

### **How IPSec Mitigates Network Threats in this Scenario**

There are a variety of circumstances that make internal computers difficult to secure without IPSec. For example, administrators typically deploy firewalls to separate networks with greater risk of attack from networks with less risk. However, when internal corporate network servers need to be secured, it is often impractical to place all of the servers behind a firewall because internal clients, servers, and their domain controllers are typically all considered to be subject to

the same level of risk from network-based attacks. In many cases, this risk is at the same level or only slightly less than the risk of attack from the Internet. Corporate offices often have many physical network connection points that can be used by anyone who is visiting or working in the building. Many internal networks are in fact open to the Internet, and administrators of large enterprise networks might not be able to monitor and control all points of access or attack from the Internet. Therefore, the threat of attacks on internal computers must be accepted and addressed when these attacks can be detected.

To mitigate the threat of network-based attacks against internal corporate network computers, you can use IPsec to help secure traffic in internal corporate networks. The full strengths of Windows IPsec are realized when you combine host-based, permit and block packet filters with the ability to enforce trusted access for network connections. Through host-based IPsec packet filtering, you can permit or block specific types of unicast IP traffic based on source and destination address combinations, specific protocols, and specific ports. Through the enforcement of trusted access, you can ensure that only trusted computers that have specific IP addresses or that are within specific IP address ranges can access an internal corporate network server. In addition, you can use IPsec to audit which computers are connecting to the server and when.

---

**Note** Although Windows IPsec can perform limited filtering, external firewalls and host-based firewalls can provide more advanced filtering capabilities.

---

IPsec is integrated within the security framework of Windows 2000 and Windows Server 2003. This allows you to use Active Directory domains to provide identity and manage trust relationships between users and computers in different departments within your organization. You can centrally manage IPsec policies through Group Policy configuration of Active Directory domains and OUs. The integration of IPsec within the security framework of Windows 2000 and Windows Server 2003, and the ability to use Active Directory domains to provide identity and manage trust relationships, allow you to use IPsec to ensure that network communication is allowed only among trusted and managed members of domains. Additionally, you do not need a separate trust infrastructure, such as a public key infrastructure (PKI) that issues X.509 v3 computer certificates (however, you can use a PKI, if this solution is appropriate for your network). You can also use IPsec to negotiate security without an Active Directory domain because IPsec supports two authentication methods (public key certificate and preshared key) that do not require an Active Directory environment.

IPsec allows you to configure customized security rules to ensure that network connections originate from a trusted source and are highly protected. By securing and authorizing host-to-host communication with an IPsec policy, you can deploy strong internal defenses against attacks within your internal corporate network.

IPsec can mitigate most of the network threats that CORPSRV faces in three ways:

- IPsec can restrict network access to trusted computers to help protect against remote network-based attacks. IPsec restricts access to trusted computers by requiring host-based authentication and authorization. With IPsec, access control is not based on user identity, but on the ability of two computers to mutually authenticate. IPsec access is granted implicitly when both computers successfully authenticate each other. You can specify which authentication method to use (public key certificate, Kerberos version 5, or preshared key) by configuring the appropriate setting in an IPsec policy. You can use these authentication

---

methods to restrict access to computers and applications.

- IPSec can provide transparent cryptographic protection for network traffic. Because IPSec is integrated at the IP layer (layer 3), you can specify which type of IPSec protection to use for application traffic (authentication only or authentication and encryption), and you can apply this protection without changing upper-layer protocols or applications.
- IPSec can partially mitigate trusted attacks by authorized computers through host-based packet filtering. For example, you can configure specific IPSec filters to limit remote computers' access to ports and protocols on a server.
- IPSec can provide an audit record of communication. If the attacker did not delete the audit log, then you can analyze IPSec audit events to determine which trusted computers were used to access the server and at what time. However, Windows IPSec does not provide audit log analysis tools. To generate reports of IPSec audit events, you must develop customized tools. For more information about IPSec audit events, see "[Enabling Security Log Audit Events for IKE Negotiation](#)," later in this paper.

The following sections describe in more detail each of the ways in which IPSec mitigates most of the network threats that CORPSRV faces.

### **Restricting Network Access to Trusted Computers**

For IPSec communication to succeed, two computers must successfully complete a mutual authentication process. Mutual computer authentication is important for the following reasons:

- It establishes trust between computers before upper-layer protocol connections can be made and therefore blocks the untrusted attacker (as long as the attacker is at an untrusted computer) from exploiting vulnerabilities (such as applications that might be vulnerable to harm from malicious data received in a network connection). Also, some protocols are not designed to defend against an attacker who spoofs a server IP address or captures network traffic in order to collect user identities and passwords.
- It limits the ability of an attacker to conduct reconnaissance by port scanning, fingerprinting application versions, and probing for application configurations (because it is difficult to use an untrusted computer to perform these attacks).
- It establishes initial trust before users attempt to authenticate. This is needed because weak protocol or application security design might allow anonymous access to information over the network. Because computers are often located within physically secured offices and buildings, you can enforce an effective network security policy by using computer trust groups in combination with their network IP addresses. Doing so helps provide a defense against unknown theft or the malicious use of legitimate user credentials.
- It provides the strong audit trail that is needed to detect unusual communication patterns and to track abuse of internal user access and network access.

### **Providing Transparent Cryptographic Protection for Network Traffic**

Many security services provided with upper-layer protocols depend on applications to use them and to use them correctly. However, because IPSec is integrated at the IP 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. Instead, IPSec helps secure data at the IP protocol layer to maintain data confidentiality and data integrity.

Many physical network connections allow computers to view and capture the network communication of other users on the same network. IPSec-signed communication defends against spoofing and modification of network traffic. IPSec-signed and IPSec-encrypted communication helps defend against all attacks that use information in the authentication or data messages of upper-layer protocols. IPSec data confidentiality ensures that data owners retain control of that data as it traverses a potentially nonsecure network or a network that is known to be nonsecure.

### **Mitigating Trusted Attacks**

As a computer-level network traffic protection service, IPSec does not replace the requirement of an application to authenticate users and to perform authorization based on user identities. Do not use IPSec as a replacement for other user and application security controls because IPSec cannot protect against attacks from established and trusted communication paths. IPSec cannot prevent attacks if a trusted computer has been infected or compromised by a malicious program, such as a virus or Trojan horse, or if it is being used by a malicious user. To help mitigate the risks from trusted attacks, you should perform additional procedures to enhance the security of internal corporate network servers, services, applications, and the authentication system used in your network. For example, because Windows 2000 IPSec and Windows XP IPSec do not provide intrusion detection or packet content inspection, consider implementing host-based intrusion detection systems (HIDS) to inspect inbound and outbound network traffic above the IPSec layer.

Because IPSec provides a new encapsulation for IP traffic, network-based intrusion detection systems (NIDS) might not be able to inspect IPSec-protected traffic. You can enhance NIDS to inspect IPSec Authentication Header (AH) or Encapsulating Security Payload (ESP) traffic without encryption, but these systems cannot inspect IPSec ESP traffic with encryption. It is recommended that you contact your hardware vendor for information about recent software updates that allow IPSec-signed communications to be inspected. Also, advances in virus writing toolkits have enabled the development of increasingly sophisticated viruses. For example, polymorphic and metamorphic viruses have engines that allow these viruses to randomly transform themselves to appear in many different patterns, drastically reducing the effectiveness of IP packet inspection for intrusion detection. Likewise, scanning IP packet content against a signature database is not effective against new attacks, and it is of limited use in detecting attacks within upper-layer protocols that span multiple IP packets.

---

NIDS that use traffic analysis for anomaly detection (that is, NIDS that are designed to detect abnormal patterns in traffic volume, the mix of protocols over which traffic is sent, and source and destination address distributions) might be effective when you adapt them to analyze IPSec-protected packets. However, this technique has limited effectiveness because IPSec encapsulation provides partial traffic flow analysis protection as a security feature of the IETF IPSec design. Accordingly, it is recommended that you use HIDS when traffic is protected by IPSec. In many cases, combining limited network access to IPSec-authenticated peers and address-based permit and block filtering is sufficient to help protect servers from a wide range of attacks. Using IPSec with non-Microsoft HIDS can be an extremely effective countermeasure against most current forms of attack from trusted computers.

---

## Preparing to Use IPsec to Secure the Internal Corporate Network in this Scenario

Before you design IPsec policies to protect a secured server, as described in this scenario, it is important to identify which ports on the server are open. It is also important to understand IPsec policy concepts so that you can design appropriate policies that function as intended.

### Identifying Ports to Block or Secure with IPsec Filters

The more ports that are open on a server, the wider the attack surface on that server. To reduce the attack surface, you must first verify which ports are open, which ports must be open, and which ports might have vulnerabilities. With this information, you can configure an IPsec policy with the filters that are required to block or secure access to vulnerable ports.

Use the **netstat** command to display all active TCP connections and the TCP and UDP ports on which a server is listening. If you run the **netstat -an** command on a server running Windows 2000 Server, it produces output similar to that which is shown in the following example.

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:21	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1057	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1114	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1147	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1180	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1782	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3372	0.0.0.0:0	LISTENING
TCP	0.0.0.0:42510	0.0.0.0:0	LISTENING
TCP	192.168.79.233:139	0.0.0.0:0	LISTENING
UDP	0.0.0.0:445	*:*	
UDP	0.0.0.0:1029	*:*	
UDP	0.0.0.0:1407	*:*	
UDP	0.0.0.0:3456	*:*	
UDP	192.168.79.233:137	*:*	
UDP	192.168.79.233:138	*:*	
UDP	192.168.79.233:500	*:*	
UDP	192.168.79.233:43508	*:*	

The output in this example indicates that the TCP ports are in a listening state but that the UDP ports are not. This is because the design of TCP/IP does not allow **netstat** to determine whether the UDP ports are used by a service for receiving traffic. A safe assumption is that these ports can be used to receive traffic.

---

Table 1 summarizes the key ports that appear in the **netstat -an** output.

**Table 1 Sample Netstat Output for a Server**

Service	Port/Protocol	Notes
File Transfer Protocol (FTP)	21/tcp	Well-known port
RPC endpoint mapper	135/tcp	Well-known port
SMB	445/tcp, 445/udp	Well-known port for file sharing and other SMB protocol functions
Unknown	1114/tcp	Dynamic port, used by an application
Unknown	1147/tcp	Dynamic port, used by an application
NetBIOS over TCP/IP (NetBT) name service	137/udp	Well-known port
NetBT datagram service	138/udp	Well-known port
NetBT session service	139/tcp	Well-known port
Internet Security Association and Key Management Protocol (ISAKMP)	500/udp	Well-known port required by the IKE protocol (the IKE protocol might also use UDP port 4500)

Note that the **netstat** output reflects only the applications and services that are running on the computer at a given time. Therefore, the output changes as soon as different applications and services are run. If you do not use IPsec or firewalls to filter traffic, an attacker can send packets to the ports that appear in the output to discover whether these ports are in a listening state (a technique known as port scanning). In less than one second, an attacker can determine which ports are available to attack.

In addition to the protocols that use TCP and UDP ports, the TCP/IP stack supports several IP protocols that do not use ports. For example, Internet Control Message Protocol (ICMP) does not use ports. ICMP provides network diagnostics and forwards communication control messages that are required in some cases for proper TCP and UDP network traffic flow. Also, the IPsec AH and ESP protocols do not use ports. When you configure an IPsec policy to block or help secure access to a server, make sure that you understand which additional protocols and ports your application requires. Do not rely solely on **netstat** output.

### Considerations for NetBT and SMB

NetBT was designed for LANs and does not provide strong security for its TCP or UDP packets. For information about how to disable NetBT, see article 299977, "Direct Hosting of SMB over TCP/IP," in the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

In addition, anonymous attackers might misuse NetBT ports 137, 138 and 139 and SMB port 445 to establish null sessions. A null session is a session with a server in which no user authentication is performed, and, therefore, anonymous access is allowed. Null sessions can present a security risk by providing an attacker with information that can be used to break user passwords. However, some applications might require null sessions. If the ports that are used by NetBT are open (UDP ports 137 and 138 and TCP port 139), an attacker might gain access to the computer. For more information about the risks of null sessions, see Chapter 3, "Enumeration," and Appendix B in *Hacking Exposed: Network Security Secrets & Solutions* (a featured title in

---

[Authored Books](http://go.microsoft.com/fwlink/?LinkId=19317), at <http://go.microsoft.com/fwlink/?LinkId=19317>). For best practices to defend against attacks based on the misuse of null sessions, see Chapter 6, “Hardening the Base Windows 2000 Server,” in [Microsoft Solution for Securing Windows 2000 Server](http://go.microsoft.com/fwlink/?LinkId=18758), at <http://go.microsoft.com/fwlink/?LinkId=18758>. If you cannot disable NetBT because it is required for applications in your network, consider using IPSec to secure traffic for those applications.

### Considerations for RPC

RPC is a message-passing facility that is used by many services and management tools. RPC allows a distributed application to call services that are available on various computers on a network. The RPC endpoint mapper or locator service is typically the initial point of contact for locating services that listen to RPC. To determine which services are registered to respond to network requests, you can use Rpcdump.exe, a command-line tool that is provided with the Windows 2000 Server Resource Kit. To download the latest version of Rpcdump.exe, see [Rpcdump.exe: RPC Dump](http://go.microsoft.com/fwlink/?LinkId=19324), at <http://go.microsoft.com/fwlink/?LinkId=19324>.

The following is an excerpt from the output that might appear if you run the **rpcdump /v /i** command:

```
Querying Endpoint Mapper Database...
16 registered endpoints found.
```

```
ProtSeq:ncacn_ip_tcp
Endpoint:1147
NetOpt:
Annotation:
IsListening:YES
StringBinding:ncacn_ip_tcp:192.168.79.233[1147]
UUID:82ad4280-036b-11cf-972c-00aa006887b0
ComTimeOutValue:RPC_C_BINDING_DEFAULT_TIMEOUT
VersMajor 2 VersMinor 0
```

```
ProtSeq:ncacn_ip_tcp
Endpoint:1114
NetOpt:
Annotation:
IsListening:YES
StringBinding:ncacn_ip_tcp:192.168.79.233[1114]
UUID:378e52b0-c0a9-11cf-822d-00aa0051e40f
ComTimeOutValue:RPC_C_BINDING_DEFAULT_TIMEOUT
VersMajor 1 VersMinor 0
```

```
ProtSeq:ncacn_ip_tcp
Endpoint:1114
NetOpt:
Annotation:
IsListening:YES
StringBinding:ncacn_ip_tcp:192.168.79.233[1114]
UUID:1ff70682-0a51-30e8-076d-740be8cee98b
ComTimeOutValue:RPC_C_BINDING_DEFAULT_TIMEOUT
VersMajor 1 VersMinor 0
```

The first block of the Rpcdump.exe output shows that an RPC program with a universal unique identifier (UUID) of 82ad4280-036b-11cf-972c-00aa006887b0 is using TCP port 1147. However, the output does not indicate which RPC program it is or what the program does. Additionally, the output does not indicate whether the RPC program can be accessed anonymously or whether the program requires authentication. The output can only confirm that an RPC program is using TCP port 1147.

---

The second and third blocks of the Rpcdump.exe output show that two RPC programs are using TCP port 1114 and that each program has its own UUID. Note also that this output does not show TCP and UDP port 135. The RPC locator service uses these ports to determine which port a specific RPC program is using on the remote computer.

The advantage of the RPC model is that the network is almost completely transparent to a distributed application. Client programs call what appear to be local procedures, and these procedures are automatically turned into remote calls. The code that translates data, accesses the network, and retrieves results is generated by Microsoft Interface Definition Language (MIDL) compiler and is invisible to applications. For more information, see [RPC Components](http://go.microsoft.com/fwlink/?LinkId=19222), at <http://go.microsoft.com/fwlink/?LinkId=19222>.

Although some RPC programs allow the use of a specific (static) port, RPC typically allocates ports for the RPC listener each time the RPC program runs. The design of RPC makes it difficult to associate a network port with a particular application. Communication often results in multiple TCP connections, sometimes in both directions, between computers. In some cases, critical operating system services or applications that use RPC cause network ports to open, even if the services or applications do not require network communication. As a result, you might not be able to stop or disable a service or application to prevent the RPC port from being opened.

From a network security perspective, it is important to know that static, port-based packet filtering is not an effective way to limit RPC communication because RPC port usage is dynamic. Many RPC-based applications use the locator service port. However, some RPC-based applications use a fixed port for certain RPC functions. For other RPC-based applications, you can configure a static port. Additionally, some RPC-based applications might use both fixed and dynamic ports for RPC communication, particularly remote management applications. For these reasons, blocking access to the RPC locator port, or blocking access to a fixed RPC port, is not always an effective way to block RPC attacks.

RPC provides many security services, including user authentication; negotiation of security parameters; and the ability to use integrity, encryption, or integrity and encryption to help protect the RPC network traffic. However, the security services are optional. You must design the RPC program to request these security services through the application programming interfaces (APIs). In some cases, for cross-platform compatibility, RPC applications do not use or require these security services. As the Rpcdump.exe output shows, it is not possible to determine which RPC programs require authentication and which do not. Also, without detailed traffic analysis, it is not possible to determine which RPC programs use digital signing and encryption to help protect their traffic or which algorithms and strengths are used.

To enhance security without modifying applications, you can use IPSec to help protect RPC traffic and all other types of network traffic.

For more information about RPC, see [The RPC Model](http://go.microsoft.com/fwlink/?LinkId=18762) at <http://go.microsoft.com/fwlink/?LinkId=18762>.

## **Overview of IPSec Policy Concepts**

When you create an IPSec policy, you configure IPSec rules, which determine IPSec behavior, and settings, which apply regardless of the rules that are configured. After you configure an IPSec policy, you must assign it to a computer for the policy to be enforced. Although multiple

---

IPSec policies can exist on a computer, only one IPSec policy can be assigned to a computer at a time.

An IPSec rule determines which types of traffic IPSec must examine; whether traffic is permitted, blocked, or security is negotiated; how to authenticate an IPSec peer; and other settings. When you configure an IPSec rule, you configure a filter list that includes one or more filters, a filter action, authentication methods, a connection type, and an IPSec encapsulation mode (transport mode or tunnel mode). An IPSec rule is typically configured for a specific purpose (for example, "Block all inbound traffic from the Internet to TCP port 135").

Filters define the traffic that you want to inspect, similar to a firewall rule, with source and destination IP addresses, protocols, and port numbers, if applicable. A filter action defines the security requirements for the network traffic. You can configure a filter action to permit, block, or negotiate security (negotiate IPSec). If you configure a filter action to negotiate security, you must also configure key exchange security methods (and their preference order), whether to accept initial incoming unsecured traffic, whether to allow unsecured communication with computers that do not support IPSec, and whether to use perfect forward secrecy (PFS).

Key exchange settings and key exchange security methods determine the IPSec protocol wire formats (AH or ESP), encryption and hashing algorithms, key lifetimes, and other settings required to configure both the IKE main mode and IPSec SAs. An SA is the agreement of security settings associated with keying material. The SA created during the first IKE negotiation phase is known as the IKE main mode SA (also known as the ISAKMP main mode SA). The IKE main mode SA protects the IKE negotiation itself. The SAs created during the second IKE negotiation phase are known as the IPSec SAs (also known as IKE quick mode SAs because each IKE quick mode negotiation negotiates the IPSec SA for each direction). The IPSec SAs protect application traffic.

This section provides information about the following important IPSec policy concepts:

- IKE negotiation process
- IPSec policy filters
- Security methods
- IPSec protocol wire formats
- IKE authentication
- IKE authentication method and security method preference order
- Security negotiation options

For more information about IPSec policy concepts, see Help and Support Center for Windows Server 2003.

### **IKE Negotiation Process**

The IKE protocol is designed to help securely establish a trust relationship between each computer, negotiate security options, and dynamically generate shared, secret cryptographic keying material. In order to ensure successful and secure communication, IKE performs a two-phase operation: Phase 1 (main mode) negotiation and Phase 2 (quick mode) negotiation.

---

Confidentiality and authentication may be ensured during each phase by the use of encryption and authentication algorithms that are agreed upon by the two computers during security negotiations.

### **Main Mode Negotiation**

During main mode negotiation, the two computers establish a secure, authenticated channel. First, the following IPSec policy parameters are negotiated: the encryption algorithm (DES or 3DES), the integrity algorithm (MD5 or SHA1), the Diffie-Hellman group to be used for the base keying material (Group 1, Group 2, or, in Windows Server 2003, Group 2048), and the authentication method (Kerberos version 5, public key certificate, or preshared key). After IPSec policy parameters are negotiated, the Diffie-Hellman exchange of public values is completed. The Diffie-Hellman algorithm is used to generate shared, symmetric, secret keys between computers. After the Diffie-Hellman exchange is complete, the IKE service on each computer generates the master key that is used to help protect authentication. The master key is used, with the negotiation algorithms and methods, to authenticate identities. The initiator of the communication then presents an offer for a potential SA to the responder. The responder sends either a reply accepting the offer or a reply with alternatives. The result of a successful IKE main mode negotiation is a main mode SA.

### **Quick Mode Negotiation**

During quick mode negotiation, a pair of IPSec SAs is established to help protect application traffic, which can include packets sent over TCP, UDP, and other protocols. First, the following policy parameters are negotiated: the IPSec protocol wire format (AH or ESP), the hash algorithm for integrity and authentication (MD5 or SHA1), and the algorithm for encryption (DES or 3DES), if encryption is requested. During this time, a common agreement is reached regarding the type of IP packets to be carried in the IPSec SA pair that is established. After IPSec policy parameters are negotiated, session key material (cryptographic keys and key lifetimes, in seconds and KBs, for each algorithm) is refreshed or exchanged.

Each IPSec SA is identified by a Security Parameter Index (SPI), which is inserted into the IPSec header of each packet sent. One SPI identifies the inbound IPSec SA; the other SPI identifies the outbound IPSec SA.

### **IKE Main Mode SAs and IPSec SAs**

Each time IPSec is used to help secure traffic, one IKE main mode SA and two IPSec SAs are established. In the example scenario, for IPSec-secured communications to occur between CORPCLI and CORPSRV, the following SAs are established:

CORPCLI [IP1] <----- IKE main mode SA [IP1, IP2] -----> [IP2] CORPSRV

CORPCLI [IP1] ----- IPSec SA [SPI=x]-----> [IP2] CORPSRV

CORPCLI [IP1] <----- IPSec SA [SPI=y] ----- [IP2] CORPSRV

Where:

- IP1 is the IP address of CORPCLI.
- IP2 is the IP address of CORPSRV.

- x is the SPI that identifies the inbound IPsec SA for CORPSRV from CORPCLI.
- y is the SPI that identifies the outbound IPsec SA for CORPSRV to CORPCLI.

As this summary indicates, the IKE main mode SA between CORPCLI and CORPSRV is bidirectional. Either computer can initiate a quick mode negotiation by using the protection provided by the IKE main mode SA. IPsec SAs are not dependent on the state of upper-layer protocols. For example, TCP connections can be established and ended while IPsec SAs continue, and IPsec SAs can expire before a TCP connection ends. IKE attempts to renegotiate, by using the quick mode negotiation to establish two new IPsec SA pairs before the lifetime of the existing IPsec SA pair expires, to help prevent a connection from being disrupted. Although this process is commonly referred to as rekeying the IPsec SA, two new IPsec SAs are actually established. The life of the IKE main mode SA is measured only by time and the number of IPsec SAs that have been attempted (not by the number of bytes of data that is transferred in the IKE protocol). The IKE main mode SA expires independently of the IPsec SA pair. If a new IPsec SA pair is needed, an IKE main mode SA is automatically renegotiated as required (when a main mode SA has expired). By IETF design, IKE must be able to rekey the main mode SA and negotiate IKE quick mode in either direction. Therefore, the authentication method that is configured in the IPsec policy on both computers for the IKE main mode SA should allow authentication to succeed in the direction from which the IKE main mode negotiation is initiated. Likewise, the IPsec policy settings in the filter action for quick mode should allow successful bidirectional quick mode negotiation.

### IPsec Policy Filters

Filters are the most important part of an IPsec policy. If you do not specify the proper filters in either client or server policies, or if the IP addresses change before the policy's filters are updated, security might not be provided. IPsec filters are inserted into the IP layer of the TCP/IP networking protocol stack on the computer so that they can examine (filter) all inbound or outbound IP packets. Except for a brief delay, which is required to negotiate a security relationship between two computers, IPsec is transparent to end-user applications and operating system services. Filters are associated with a corresponding filter action by the security rule in an IPsec policy. Windows IPsec supports both IPsec tunnel mode and IPsec transport mode as an option in the rule. IPsec tunnel mode rule configuration is very different from IPsec transport mode rule configuration. (For more information, see "IPsec Encapsulation Modes," in "[IPsec Encapsulation Modes and Protocol Wire Formats](#)," later in this paper.) Because the scenario in this paper describes only IPsec transport mode, the filters in this paper are referred to as IPsec transport mode filters.

The filtering rules associated with an IPsec policy are similar to firewall rules. By using the graphical user interface (GUI) provided by the IP Security Policy Management snap-in, you can configure IPsec to permit or block specific types of traffic based on source and destination address combinations and specific protocols and ports.

---

**Note** Windows IPsec is not a full-featured, host-based firewall, and it does not support dynamic or stateful filtering features, such as tracking the established bit during the TCP handshake to control the direction in which communication can flow.

---

---

## Security Methods

Security methods are used during the IKE main mode negotiation to define the encryption and hashing algorithms and the Diffie-Hellman group that is used to create the main mode SA and to help secure the IKE negotiation channel. Security methods are also used during the quick mode negotiation to define the encapsulation mode (transport or tunnel), IPSec protocol wire format (AH or ESP), encryption and hashing algorithms, and key lifetimes that are used to create the quick mode inbound and outbound SAs.

## IPSec Encapsulation Modes and Protocol Wire Formats

IPSec helps protect data in an IP packet by providing cryptographic protection of an IP payload. The protection that is provided depends on the mode in which IPSec is used and the protocol wire format. You can use IPSec in transport mode or tunnel mode.

### IPSec Encapsulation Modes

IPSec tunnel mode is most commonly used to help protect site-to-site (also known as gateway-to-gateway or router-to-router) traffic between networks, such as site-to-site networking through the Internet. When IPSec tunnel mode is used, the sending gateway encapsulates the entire, original IP packet by creating a new IP packet that is then protected by one of the IPSec protocol wire formats (AH or ESP). For information about IPSec in tunnel mode, see Chapter 6, “Deploying IPSec,” in *Deploying Network Services*, in the [Windows Server 2003 Deployment Kit](http://go.microsoft.com/fwlink/?LinkId=8195), on the Web at <http://go.microsoft.com/fwlink/?LinkId=8195>.

IPSec transport mode is used to help protect host-to-host communications, and it is the default mode for Windows IPSec. When IPSec transport mode is used, IPSec encrypts only the IP payload; the IP header is not encrypted. Windows IPSec is used in transport mode primarily to help protect end-to-end communication (such as communications between clients and servers).

---

**Note** The example scenario in this paper describes the use of IPSec in transport mode only. The use of tunnel mode for this scenario is not recommended because tunnel mode increases the complexity of the IPSec policy configuration, while providing no additional benefit. Although the use of tunnel mode in the scenario would protect the original IP header because the IP header of the tunnel packet and the original packet contain the same information, there is no benefit to using IPSec in tunnel mode.

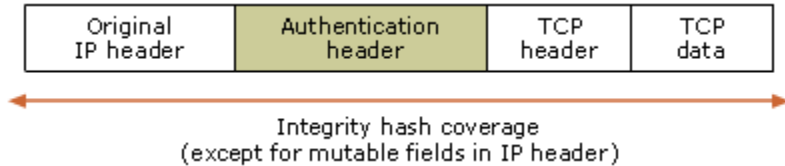
---

### IPSec Protocol Wire Formats

IPSec supports two protocol wire formats: AH or ESP. IPSec transport mode encapsulates the original IP payload with an IPSec header (AH or ESP).

## AH

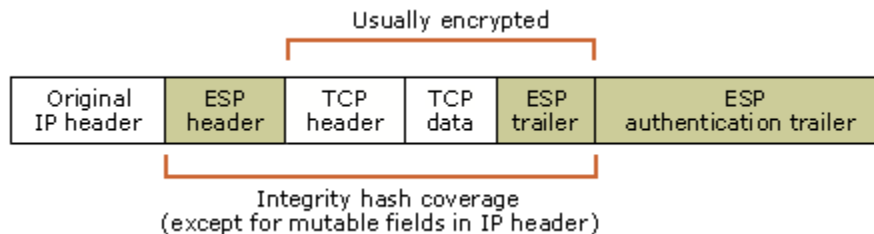
AH provides data origin authentication, data integrity, and anti-replay protection for the entire packet (both the IP header and the data payload carried in the packet), except for the fields in the IP header that are allowed to change in transit. AH does not provide data confidentiality, which means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



To use AH, in the properties for the appropriate rule, select the **Data and address integrity without encryption (AH)** check box, in the **Custom Security Method Settings** dialog box, and then specify the integrity algorithm to use.

## ESP

ESP 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. The IP header is not protected. As shown in the following figure, the ESP header is placed before the TCP data, and an ESP trailer and ESP authentication trailer are placed after the TCP data.



To use ESP, in the properties for the appropriate rule, select the **Data integrity and encryption (ESP)** check box, in the **Custom Security Method Settings** dialog box, and then specify the integrity and encryption algorithms to use.

---

**Note** In the example used in this paper, the rules for the IPSec policies that are assigned to CORPSRV and CORPCLI specify that ESP encryption is required for communication with the IPSec-secured server, CORPSRV. For more information, see [“Designing an IPSec Policy for the Server \(CORPSRV\),”](#) and “Configuring CORPCLI Policy 1 Rules,” in [“Designing IPSec Policies for the Client \(CORPCLI\),”](#) later in this paper.

---

---

## **IKE Authentication**

IKE uses mutual authentication between computers to establish trusted communications and requires the use of one of the following authentication methods: Kerberos version 5, a computer X.509 version 3 PKI certificate, or a preshared key. The two communication endpoints must have at least one common authentication method, or communication fails.

### **IKE Authentication Process**

During IKE negotiation, the IKE initiator proposes a list of authentication methods to the IKE responder. The responder uses the source IP address of the initiator to identify which filter controls the IKE negotiation. The authentication method list that corresponds to the filter in the responder's IPSec policy is used to select one authentication method from the initiator's list. The responder then replies to inform the initiator of the agreed-upon authentication method. If the selected authentication method fails, IKE does not provide a method for trying a different authentication method. If authentication is successful and the main mode negotiation is successfully completed, the main mode SA lasts for eight hours. If data is still being transmitted at the end of eight hours, then the main mode SA is renegotiated automatically.

### **IKE Authentication Methods**

It is important to choose the authentication method that is appropriate for your IPSec policy. An IPSec policy rule associates each IP address in a filter with an authentication method list so that IKE can determine which authentication method list to use with each IP address.

#### ***Kerberos version 5 authentication***

Kerberos version 5 is the default authentication standard in Windows 2000 and Windows Server 2003 Active Directory domains. Any computer in the domain or in a trusted domain can use this method of authentication.

When Kerberos authentication is used, during main mode negotiation, each IPSec peer sends its computer identity in unencrypted format to the other peer. The computer identity is unencrypted until encryption of the entire identity payload takes place during the authentication phase of the main mode negotiation. An attacker can send an IKE packet that causes the responding IPSec peer to expose its computer identity and domain membership. For this reason, to help secure computers that are connected to the Internet, certificate authentication is recommended.

By default, in Windows 2000 through Service Pack 3 and in Windows XP, Kerberos traffic is exempt from IPSec filtering. In order to remove the exemption for Kerberos traffic, you must modify the registry and then add an appropriate IPSec filter to help secure this traffic. For more information, see "[Removing the Default Filtering Exemption for Kerberos and RSVP Traffic](#)," later in this paper.

In the scenario used in this paper, Kerberos is used for authentication between CORPSRV and CORPCLI.

#### ***Public key certificate authentication***

In Windows 2000 Server, you can use Certificate Services to automatically manage computer certificates for IPSec throughout the certificate lifecycle. Certificate Services is integrated with Active Directory and Group Policy, and it simplifies certificate deployment by enabling certificate

---

auto-enrollment and renewal and by providing several default certificate templates that are compatible with IPSec. To use certificates for IKE authentication, you define an ordered list of acceptable root certification authorities (CAs) to use, not which specific certificate to use. Both computers must have a common root CA in their IPSec policy configuration, and clients must have an associated computer certificate.

During the certificate selection process, IKE performs a series of checks to help ensure that specific requirements are met for the computer certificate. For example, the computer certificate must have a public key length that is greater than 512 bits and use a Digital Signature key usage.

---

**Note** Certificates obtained from Certificate Services with the advanced option set for **Enable strong private key protection** do not work for IKE authentication because you cannot enter the required personal identification number (PIN) to access the private key for a computer certificate, during IKE negotiation.

---

#### *Preshared keys*

If you are not using Kerberos authentication and do not have access to a CA, a preshared key can be used. For example, a stand-alone computer on a network might need to use a preshared key because neither Kerberos authentication, through the computer's domain account, nor certificates from a CA can enable successful IKE authentication in some scenarios.

---

**Important** 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. The preshared key value is stored in plaintext in an IPSec policy. Any member of the local Administrators group can view a local IPSec policy, and a local IPSec policy can be read by any system service with Local System user rights. By default, any authenticated user in the domain can view a preshared key if it is stored in an Active Directory-based IPSec policy. Additionally, if attackers can capture IKE negotiation packets, published methods can enable the attackers to discover preshared key values. For more information, see [Authentication vulnerabilities in IKE and Xauth with Weak Pre-Shared Secrets](#), at <http://go.microsoft.com/fwlink/?LinkId=18769>.

---

Preshared key authentication is provided for interoperability purposes and compliance with RFC standards. If you must use preshared key authentication, use a 25-character or longer random key value and a different preshared key for each IP address pair. These practices result in different security rules for each destination and help ensure that a compromised preshared key compromises only those computers that share the key.

The scenario in this paper describes how to use a preshared key correctly. In the scenario used in this paper, preshared keys are used for authentication between CORPSRV and the perimeter network server (WEBSRV) and between CORPSRV and the perimeter network domain controller (PNDC) for specific reasons, as described in "CORPSRV Policy Rules," in "[Designing an IPSec Policy for the Server \(CORPSRV\)](#)," later in this paper.

#### **IPSec CRL Checking**

If you use certificate-based authentication, you can also enable IPSec certificate revocation list (CRL) checking. By default in Windows 2000, IPSec CRLs are not automatically checked during IKE certificate authentication.

#### **To enable IPSec CRL checking**

---

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

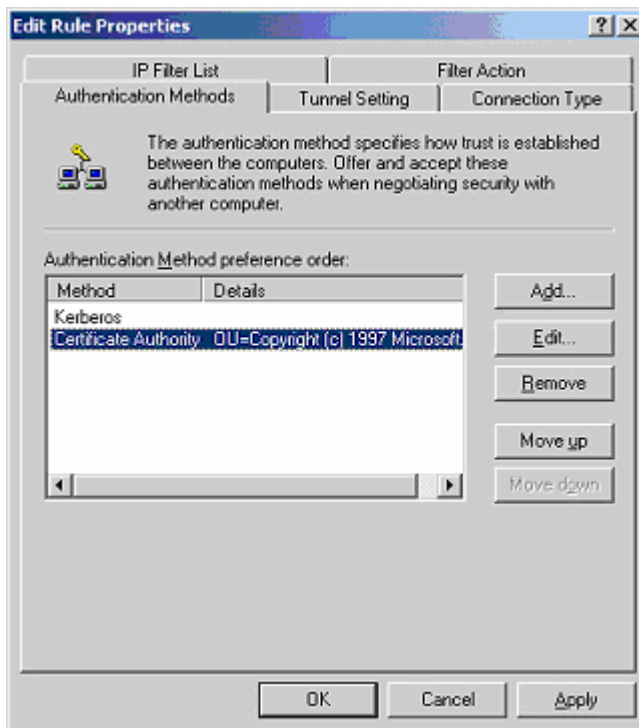
---

1. Under **HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\PolicyAgent\**, add a new **Oakley** key, with a DWORD entry named **StrongCrlCheck**.
2. Assign this entry any value from **0** through **2**, where:
  - A value of **0** disables CRL checking (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 (default for Windows XP and Windows Server 2003). Other failures that are encountered during CRL checking (such as the revocation URL being unreachable) 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. 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. For more information about CRLs, see [Troubleshooting Certificate Status and Revocation](http://go.microsoft.com/fwlink/?LinkId=18753), at <http://go.microsoft.com/fwlink/?LinkId=18753>.

## IKE Authentication Methods and Security Method Preference Order

You can configure an IPSec rule to specify only one authentication method or one security method. Or, you can specify a preferred list of authentication and security methods. Preference order applies to authentication methods and security methods so that you can specify each method from most preferred to least preferred. For example, you can specify that both Kerberos version 5 and public key certificate authentication are offered as authentication methods but give Kerberos the higher preference, as shown in the following figure.



If a client attempts to connect to CORPSRV but only accepts public key certificates for authentication, then CORPSRV uses this authentication method and continues establishing communication. IKE must succeed using the selected authentication method, or the communication is blocked. IKE does not attempt to use a different authentication method if the negotiation fails. The same principle applies to security methods, where, for example, ESP might be preferred over AH.

### Security Negotiation Options

You can configure whether an IPSec policy allows fall back to clear (fall back to unsecured communication), inbound passthrough, and session key PFS on the **Security Methods** tab, in the properties for a filter action. You can configure master key PFS in the **Key Exchange Settings** dialog box, in the general properties for a rule.

#### Fall Back to Clear

When fall back to clear is allowed, traffic is secured by IPSec when possible (if the computer at the other end of the connection supports IPSec with a complementary filter action and filter in its

---

policy), but traffic can be sent unsecured, if the peer does not have an IPSec policy to respond to the request for security negotiation. If the peer does not respond to the request for security negotiation within three seconds, an SA for plaintext traffic (a soft SA) is created. Soft SAs allow normal TCP/IP communication with no IPSec encapsulation to occur. Keep in mind that although IPSec might not secure such traffic, another application might help secure the traffic (for example, traffic might be secured by Lightweight Directory Access Protocol (LDAP) encryption or RPC authentication mechanisms). If the peer does respond within three seconds and the security negotiation fails, the communication that matches the corresponding filter is blocked.

Fall back to clear is a setting that allows interoperability with:

- Computers running operating systems earlier than Windows 2000.
- Computers running Windows 2000 or later systems that do not have IPSec policy configured.
- Computers running non-Microsoft operating systems that do not support IPSec.

To enable or disable fall back to clear, on the **Security Methods** tab, in the properties for a filter action, select or clear the **Allow unsecured communication with non-IPSec-aware computers** check box.

For client policy, you can either enable or disable this option. If you enable this option, and if the server does not respond to the client's request to negotiate security, you can allow the client to fall back to clear. If you clear this check box, and if the server does not respond to the client's request to negotiate security, communication is blocked. In some cases, it is useful to allow fall back to clear. However, IKE allows fall back to clear only if there is no reply. For security reasons, Windows IPSec does not allow unsecured communication if the IKE negotiation fails or if an error is experienced during an IKE negotiation (after the reply), such as failure to authenticate or to reach agreement on security parameters. For more information, see "[Customized Rules Added for IPSec Policies Assigned to Microsoft Servers and Clients](#)," later in this paper.

For initial deployments, it is recommended that you select this check box so that the client can fall back to clear and initial connectivity can be established when IPSec is disabled on the server.

### **Inbound Passthrough**

When inbound passthrough is allowed, normal inbound TCP/IP traffic (traffic that is not secured by IPSec, for example a TCP SYN packet) is accepted if it matches the inbound filter associated with the filter action. The upper-layer protocol response packet (for example, a TCP SYN ACK packet) matches the corresponding outbound filter and triggers a security negotiation. Two IPSec SAs are then negotiated, and the traffic is IPSec-secured in both directions. The inbound passthrough option allows a server to use the default response rule to initiate the security negotiation to clients. When you enable the default response rule in the client IPSec policy, clients do not need to maintain a filter that contains the IP address of the server. If you do not enable the default response rule in the client IPSec policy, then you do not need to enable the inbound passthrough option in the server IPSec policy. Additionally, you should never enable this option on computers connected to the Internet. To enable or disable inbound passthrough, on the **Security Methods** tab, in the properties for a filter action, select or clear the **Allow unsecured communication, but always respond using IPSec** check box.

---

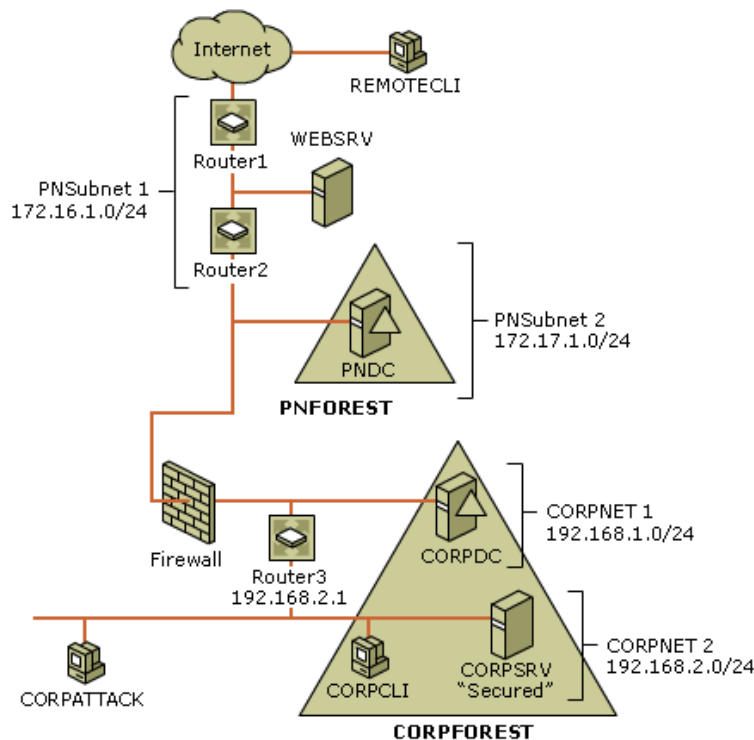
### Session Key and Master Key PFS

PFS is a mechanism that determines whether the existing keying material for a master key can be used to derive a new session key. PFS helps ensure that the compromise of a single key permits access only to data that is protected by PFS, not necessarily to the entire communication. To achieve this, PFS helps ensure that a key used to protect a transmission cannot be used to generate additional keys. Session key PFS can be used without a reauthentication and is less resource-intensive than master key PFS. When session key PFS is enabled, a new Diffie-Hellman key exchange is performed to generate new master key keying information. If you enable session key PFS in a server policy, you must also enable it in the client policy. You can enable session key PFS by selecting the **Use session key perfect forward secrecy (PFS)** check box, in the **Key Exchange Settings** dialog box, in the general properties for a rule. Master key PFS requires a reauthentication and is resource-intensive. It requires a new main mode negotiation for every quick mode negotiation that occurs. You can configure master key PFS by selecting the **Master key perfect forward secrecy (PFS)** check box. If you enable master key PFS in a server policy, you do not need to enable it in the client policy. It is recommended that you enable session key PFS or master key PFS only in hostile environments where IPSec traffic might be exposed to sophisticated attackers who will try to compromise the strong cryptographic protection provided by IPSec.

## Scenario Details: IPSec Policy Design

In Windows 2000, Windows XP, and Windows Server 2003, IPSec is implemented primarily as an administrative tool that you can use to enforce security policies on IP network traffic. The ability of IPSec to provide the appropriate level of security depends on the proper and secure configuration of the computer identity and the IPSec policy, on both sides of the communication. For example, you can configure an IPSec policy to help secure traffic over a specific TCP port or to help secure all traffic between two IP addresses, or you can configure an IPSec policy to respond to requests for secured communication. Although the ideal solution might be to use IPSec to help secure all communication, almost all servers need to communicate with other computers that cannot use IPSec or that are not configured to use IPSec. You can configure an IPSec policy with static filters that permit outbound traffic to these computers and that allow corresponding inbound traffic to be received from these computers. In this case, the degree of protection that a server has against untrusted attacks depends on the destination IP addresses from which inbound traffic is permitted.

When designing an IPSec policy, it is useful to construct an IP network topology diagram. The following figure shows the IP network topology of the example network introduced in "[Scenario Introduction: The Internal Corporate Network Architecture](#)," earlier in this paper, and includes example IP addresses and subnet ranges. These addresses and subnets are used in the example IPSec policies described in "[Designing an IPSec Policy for the Server \(CORPSRV\)](#)," "[Designing IPSec Policies for the Client \(CORPCLI\)](#)," and "[Modifying IPSec Policy Design in this Scenario](#)," later in this paper.



---

In this scenario, the security goal is to protect all network ports on CORPSRV from being accessed by untrusted computers. Trust for CORPSRV is defined as the ability of a computer to successfully authenticate as a member of the CORPDC domain or CORPFOREST. When access to CORPSRV is limited only to trusted computers, you can audit the names or IP addresses of computers that establish or fail to establish IPsec-secured connections.

To establish trusted communications for IPsec, IKE uses mutual authentication between computers. You can use a PKI by issuing digital certificates to each computer in CORPFOREST. In this scenario, Kerberos version 5 is used for IKE authentication between domain clients and CORPSRV. The Windows implementation of IKE builds on the existing trust in Active Directory domains by using Kerberos for authentication. In this scenario, you can use Kerberos because both CORPCLI and CORPSRV meet at least one of the following conditions:

- Both computers are members of the Windows 2000 CORPDC domain.
- Both computers are members of domains in the same forest.
- Both computers are members of domains for which you have configured two one-way, external trusts.

However, you cannot use Windows 2000 IPsec to configure IPsec-specific permissions or ACLs to form secure groups within a domain in which only certain computers can establish IPsec-secured communications with each other. For example, in Windows 2000, you cannot use Kerberos as an IKE authentication method to create a group that includes three computers (CORPA, CORPB, and CORPC) that can communicate with each other, but not with a fourth computer (CORPD), when all four computers are members of a trusted domain. You can add filters that allow IPsec to be negotiated only between the specific static IP addresses of the domain members that you want to include in such a group. However, this approach is only as secure as the control that you maintain over the IP address that each member of the group can use. If you use this approach, an attacker on another domain member might spoof a trusted IP address that is used by a member of the group, or another domain member might be configured to use the same trusted IP address.

Because forest membership defines the security boundary for computers that can be trusted, if you use Kerberos for IKE authentication, then plan your Windows 2000 IPsec security model to grant access to any domain computer. Because domain members might use dynamic IP addresses or be able to obtain different IP addresses, you should not design IPsec to provide access control based only on the IP address of any given domain member. Windows Server 2003 IPsec provides limited support for creating ACLs that include a specific set of computer accounts in a domain. For example, you can authorize a specific group of clients that are domain members to use IPsec to gain network access to CORPSRV, when you use either Kerberos or certificates for IKE authentication. To do this, configure Group Policy security settings, and assign either the **Access this computer from the network** or the **Deny access to this computer from the network** logon right to individual computer accounts or security groups in the domain, as needed.

---

**Note** Although IKE Kerberos authentication is affected by this setting in Windows 2000 and Windows XP, you can use this setting to configure IPsec-authenticated access control only on a computer running Windows Server 2003.

---

---

For more information, see [“Impact of Group Policy Security Settings on IPSec,”](#) later in this paper and Chapter 6, “Deploying IPSec,” in *Deploying Network Services*, in the [Windows Server 2003 Deployment Kit](#), on the Web at <http://go.microsoft.com/fwlink/?LinkId=8195>.

IPSec does not support the use of Kerberos to establish trusted communications across non-trusted domains, where explicit credential authentication is often used in upper-layer protocols. For example, in this scenario, you cannot use Kerberos to establish trusted communications between WEBSRV, the stand-alone server, and CORPSRV. A computer must be a domain member to use Kerberos, so Kerberos is not available to WEBSRV as an authentication method. In general, for large enterprise scenarios, Kerberos might not be appropriate because Kerberos requires two-way trusts across all forests and all child domains in a forest with other forest domains. For more information, see [“IKE Authentication,”](#) earlier in this paper.

In cases where clients cannot establish trusted communications with CORPSRV because they do not have access to CORPFOREST, you can use public key certificate authentication. You can also use public key certificate authentication with computers that are not running Windows. Preshared keys are also provided for interoperability but are not recommended.

---

**Important** To avoid the security risks associated with preshared key authentication, do not use preshared keys if you can use public key certificate or Kerberos authentication instead. Preshared keys are stored in plaintext in an IPSec policy, the system registry, and Active Directory. For more information, see [“IKE Authentication,”](#) earlier in this paper.

---

## Tasks for Defining IPSec Policy in this Scenario

To define an IPSec policy to help secure an internal corporate network server, perform the following tasks:

1. Configure firewalls and filtering routers to permit IPSec-secured traffic, if necessary.
2. Remove the default filtering exemption for Kerberos and Resource Reservation Protocol (RSVP) traffic.
3. Configure additional filtering exemptions, if necessary.
4. Create an IPSec policy for the server.
5. Create IPSec policies for clients.

The following sections describe how to define an IPSec policy to help secure CORPSRV.

### Configuring Firewalls and Filtering Routers to Permit IPSec-Secured Traffic

When a firewall or filtering router exists between IPSec peers, you must configure the firewall or filtering router to allow IPSec traffic on UDP source and destination port 500 (ISAKMP), IP protocol 51 (AH), and IP protocol 50 (ESP). For information about how to create these filters, see [“Creating Firewall Filters to Permit ISAKMP, AH, and ESP Traffic,”](#) later in this paper.

### Removing the Default Filtering Exemption for Kerberos and RSVP Traffic

By default, in Windows 2000 Service Packs 1 or later and in Windows XP, broadcast, multicast, Kerberos, RSVP, and ISAKMP traffic is exempt from IPSec filtering, even if you define a filter to match all IP traffic between the IP addresses of two computers. It is strongly recommended that you remove the default exemptions for Kerberos and RSVP traffic to ensure that IPSec can help

---

secure these traffic types. If you do not remove these default exemptions, then an attacker can use a technique known as source porting to bypass IPSec filtering. When this technique is used, the attacker constructs a unicast UDP or TCP packet that uses a source port of 88, which is used for Kerberos and therefore is exempt from IPSec filtering. RSVP is a protocol and therefore does not use TCP or UDP ports.

### To remove the default filtering exemption for Kerberos and RSVP traffic

---

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

---

1. Under **HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\IPSEC**, add a new DWORD entry named **NoDefaultExempt**.
2. Assign this entry a value of **1**. This specifies that Kerberos and RSVP traffic are not exempt from IPSec filtering. (Multicast, broadcast, and ISAKMP traffic are exempt.)
3. Restart the computer.

For more information about this registry setting, see [Microsoft Windows 2000 Security Hardening Guide](http://go.microsoft.com/fwlink/?LinkId=18759), at <http://go.microsoft.com/fwlink/?LinkId=18759>, and [Microsoft Solution for Securing Windows 2000 Server](http://go.microsoft.com/fwlink/?LinkId=18758), at <http://go.microsoft.com/fwlink/?LinkId=18758>.

---

**Important** To understand the traffic types that can be filtered by IPSec, it is highly recommended that you read article 811832, "IPSec Default Exemptions Can Be Used to Bypass IPSec protection in Some Scenarios," and article 810207, "IPSec Default Exemptions Are Removed in Windows Server 2003," in the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

---

### Configuring Additional Filtering Exemptions

In typical IPSec deployments, additional IPSec filters are configured to permit Domain Name System (DNS) and ICMP traffic. DNS traffic (TCP port 53 and UDP port 53) is typically permitted for name resolution. ICMP traffic is typically permitted to the default gateway to allow network discovery to function correctly. Additional IPSec filters might be required to enable allowed applications and services access to specific ports. However, if you configure IPSec filters to permit access to an allowed service, an attacker can use reconnaissance techniques or mount attacks as follows:

- **Operating system fingerprinting.** When this widely practiced reconnaissance technique is used, an attacker sends one or more TCP packets to a listening service. Based on the response that the packets receive, an attacker can determine the operating system that is running on the target computer. Operating system fingerprinting is not a direct attack, but it is often performed to determine which types of attacks are possible based on the operating system version.
- **Port scanning.** When this reconnaissance technique is used, an attacker sends one packet to a port to determine whether the computer replies. As with operating system fingerprinting, port scanning is not a direct attack, but it is often performed to determine which services can be attacked based on which ports are open.
- **Connection flooding.** When this attack is used, if a TCP service is available, the attacker produces a connection flood, which causes a denial of service.

- 
- **Exploitation of a known vulnerability in a service.** When this attack is used, an attacker targets a known vulnerability in a service to mount a specific attack. For example, if the port on which the RPC endpoint mapper listens (tcp/135) is not secured by IPSec, and that port is accessed, attackers on anonymous or untrusted computers can mount an attack on the RPC locator service. This specific security issue was found and fixed as described in [Microsoft Security Bulletin MS03-10: Flaw in RPC Endpoint Mapper Could Allow Denial of Service Attacks \(331953\)](http://go.microsoft.com/fwlink/?LinkId=18754), at <http://go.microsoft.com/fwlink/?LinkId=18754>.

## Designing an IPSec Policy for the Server (CORPSRV)

To help secure CORPSRV, you must restrict access to this server and design an IPSec policy for the server that helps secure communications with the server. To design an IPSec policy, you must configure general settings, security settings, and rules.

### CORPSRV Policy General Settings

This section describes the general settings that are applied with the IPSec policy that is assigned to CORPSRV. Although each rule that is associated with the CORPSRV IPSec policy is different, the general settings are always applied with this policy.

#### *Configuring CORPSRV policy name, description, and policy refresh interval*

The following general settings are configured on the **General** tab of the properties for the IPSec policy that is assigned to CORPSRV:

- **Name:** CORPSRV Server IPSec Policy *VersionOrDate-TimeLastModified*.
- **Description:** Secure all internal traffic, block perimeter network traffic, except to perimeter network domain controller and WEBSRV computers. Kerberos preferred.
- Policy refresh interval (**Check for policy changes every *n* minutes**): 480 minutes (eight hours).

The policy refresh interval determines how often the IPSec Policy Agent queries Active Directory for changes in the assigned IPSec policy. For the initial deployment of IPSec policy, consider setting this value to a short amount of time, such as five minutes, to ensure that you can change policy settings quickly in the event of an error or unforeseen operational impact on communications. However, if there are many servers in the domain, a short polling interval might increase the load on domain controllers.

#### *Configuring CORPSRV policy key exchange settings*

The following general settings are configured in the **Key Exchange Settings** dialog box of the properties for the IPSec policy that is assigned to CORPSRV:

- Key lifetime (**Authenticate and generate a new key after every *n* minutes**): 480 minutes (eight hours)
- Number of quick mode negotiations per main mode (**Authenticate and generate a new key after every *n* sessions**): 0 (An unlimited number of session keys can be created from the master key keying material.)

A key lifetime of eight hours ensures that the master key keying material (the Diffie-Hellman key) is regenerated after eight hours. Diffie-Hellman keys remain in memory during their lifetime, so, if

---

many clients (several thousand) are connecting to the server for short periods of time, consider reducing their lifetime to reclaim memory. You might also consider reducing the key lifetime in hostile environments where a sophisticated attacker might attempt to intercept the communication. One disadvantage to reducing the key lifetime is that if clients must perform an additional main mode negotiation, this operation can be time-consuming and memory-intensive, and frequent Diffie-Hellman calculations increase the computational load placed on the server.

#### **Configuring CORPSRV policy key exchange security methods**

The following general settings are configured in the **Key Exchange Security Methods** dialog box of the properties for the IPsec policy that is assigned to CORPSRV. These security methods are listed in order of preference, by encryption algorithm, integrity algorithm, and Diffie-Hellman group.

- 3DES/SHA1/Medium (Diffie-Hellman Group 2, 1024 bits)
- 3DES/MD5/Medium (Diffie-Hellman Group 2, 1024 bits)

Key exchange security method settings determine which security services, key settings, and algorithms are used to help protect identities during authentication and key exchange. As a best practice for security, it is recommended that you consider not using Diffie-Hellman Group 1, which provides 768 bits of keying strength. Group 1 does not provide a strong level of security (it is provided for interoperability). 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 Windows XP and Windows 2000, see article 818043, "L2TP/IPsec NAT-T Update for Windows XP and Windows 2000," in the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

---

**Note** To use 3DES on a computer running Windows 2000, you must install the High Encryption Pack or Service Pack 2 (or later). 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. For more information, see [Windows 2000 High Encryption Pack](http://go.microsoft.com/fwlink/?LinkId=7272), at <http://go.microsoft.com/fwlink/?LinkId=7272>.

---

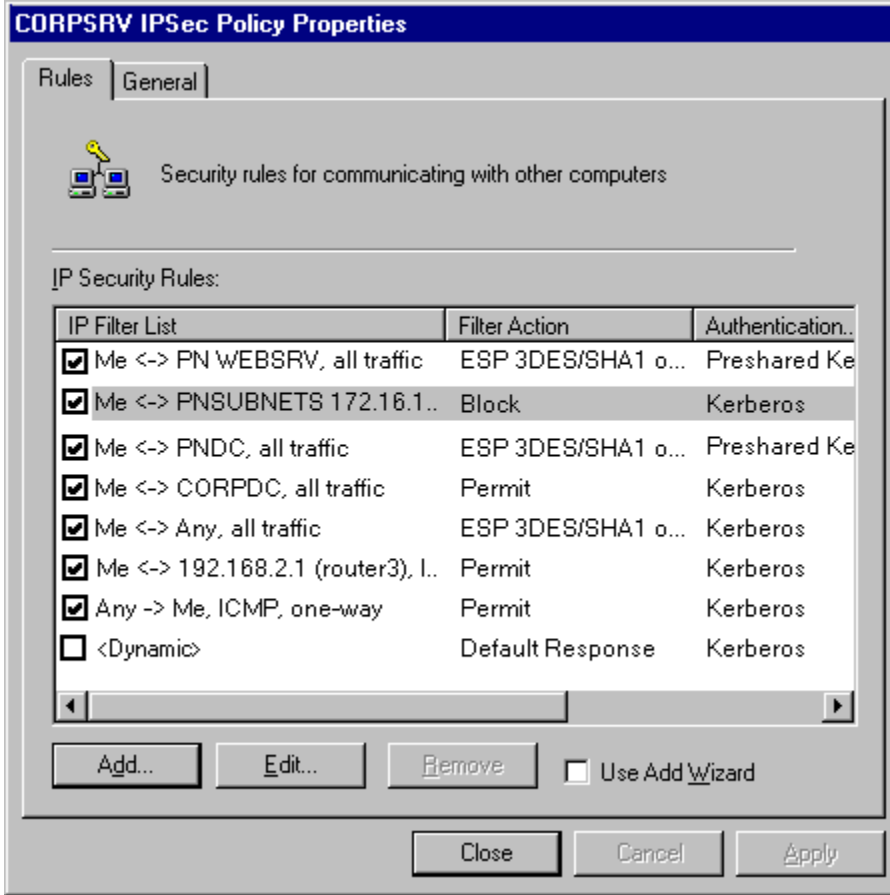
#### **CORPSRV Policy Security Settings**

The security settings associated with an IPsec policy can be as strict or as flexible as you require. They might be strict in the sense that you only want CORPSRV to use a specific authentication method and a specific security method when communicating with clients. For example, you might only allow the use of ESP and not allow fall back to clear. To provide more flexibility, you can configure an IPsec policy to allow fall back to clear or unsecured communications, in cases where the client does not support IPsec. You can also provide more flexibility by specifying multiple authentication and security methods, in order of preference.

For example, consider the key exchange security method preference order for the IPsec policy that is assigned to CORPSRV. By default, 3DES encryption, SHA1 integrity hashing, and Diffie-Hellman Group 2 are preferred for key exchange. However, if the computer on the other end of the communication cannot use 3DES encryption, CORPSRV can offer the third or fourth key exchange security method in the preference list, both of which use DES.

## CORPSRV Policy Rules

The following figure shows the eight IPsec rules that are required for the IPsec policy that is assigned to CORPSRV.



To access any TCP or UDP port on an IPsec-secured server such as CORPSRV, a computer must successfully use IKE to authenticate with CORPSRV at the IP level and then negotiate which traffic is to be secured by IPsec. The eight rules for the example IPsec policy include filters to help secure all unicast IP traffic between two computers. For enhanced security, you can create a more specific filter to help secure a subset of this traffic. For example, you can create a rule that allows and helps secure only TCP port 80 traffic, as described in “[Modifying CORPSRV and WEBSRV Policy Design to Defend against Perimeter Network Server \(WEBSRV\) Compromise](#),” later in this paper.

Table 2 provides additional details about the settings for each CORPSRV policy rule. This policy blocks perimeter network traffic, except for traffic between the perimeter network domain controller and WEBSRV. IPSec is negotiated with ESP 3DES/SHA1. Kerberos authentication is primarily used, but preshared key authentication is also used as necessary.

**Table 2 Summary of CORPSRV Policy Rules**

Rule	Filter Settings	Filter Action Settings	Authentication Method
<a href="#">1: Secure communication on path between CORPSRV and WEBSRV</a>	Me <-> Perimeter network WEBSRV, all traffic, mirrored	Negotiate ESP 3DES/SHA1 only, no clear, no fall back to clear	Preshared key
<a href="#">2: Block communication on path between CORPSRV and perimeter network subnets</a>	Me <-> Perimeter network subnets 172.16.1.0/24, 172.17.1.0/24, all traffic, mirrored	Block	N/A
<a href="#">3: Secure communication on path between CORPSRV and perimeter network domain controller</a>	Me <-> Perimeter network DC, all traffic, mirrored	Negotiate ESP 3DES/SHA1 only, no clear, no fall back to clear	Preshared key
<a href="#">4: Permit all communication on path between CORPSRV and CORPDC</a>	Me <-> CORPDC, all traffic, mirrored	Permit	N/A
<a href="#">5: Secure communication on path between CORPSRV and any computer</a>	Me <-> Any, all traffic, mirrored	Negotiate ESP 3DES/SHA1 only, inbound passthrough, no fall back to clear	Kerberos
<a href="#">6: Permit unsecured ICMP communication between CORPSRV and default gateway</a>	Me <-> 192.168.2.1 (default gateway/router 3), ICMP, mirrored	Permit	N/A
<a href="#">7: Permit CORPSRV to receive but not to send unsecured ICMP communications</a>	Any -> Me, ICMP, one-way	Permit	N/A
<a href="#">8: Default response rule disabled</a>	Default response rule	Disabled	N/A

***CORPSRV Policy Rule 1: Secure communication on path between CORPSRV and WEBSRV***

This rule is recommended when WEBSRV must communicate across trust boundaries with internal corporate network servers in the CORPDC domain. In some cases, WEBSRV might be a stand-alone server. In other cases, WEBSRV might be a member of a domain that is not fully

---

trusted by the CORPDC domain, due to the way in which external trusts are configured between the perimeter network domain and the CORPDC domain.

The following settings are configured for the first rule that is associated with the IPsec policy assigned to CORPSRV:

#### **IP filter list settings**

- **Name:** Me <-> Perimeter Network WEBSRV, all traffic

#### **Filter settings**

- **Source address:** **My IP Address**
- **Destination address:** **A specific IP address** (where the IP address is the IP address of WEBSRV)
- **Mirrored:** Yes (This check box is selected.)
- **Protocol:** **Any**
- **Description:** Me <-> WEBSRV, all traffic

#### **Filter action settings**

- **Name:** ESP 3DES-SHA1 required
- **Action:** **Negotiate security**
- **Accept unsecured traffic, but always respond using IPsec:** No (This check box is cleared, therefore inbound passthrough is not allowed.)
- **Allow unsecured communication with non-IPsec aware computers:** No (This check box is cleared, therefore fall back to clear is not allowed.)
- **Perfect forward secrecy (PFS):** No (This check box is cleared.)
- **Security method:** **Custom**
  - **Data and address integrity without encryption (AH):** No (This check box is cleared.)
  - **Data integrity and encryption (ESP):** Yes (This check box is selected.)
    - **Integrity algorithm:** **SHA1**
    - **Encryption algorithm:** **3DES**
  - **Session key settings** (key lifetimes): 100,000 KB (100 MB) / 3,600 seconds (one hour)

#### **Authentication method, IPsec mode, and connection type**

- **Authentication method:** **Preshared key**
- **Tunnel setting (IPsec mode):** **This rule does not specify an IPsec tunnel** (This option is selected, so transport mode is used.)
- **Connection type:** **All network connections**

---

This rule specifies that all traffic between CORPSRV and WEBSRV be secured with ESP encryption, SHA1 integrity hashing, and a preshared key for authentication. This rule is strict because it does not allow fall back to clear or any unsecured inbound communications. Keep in mind that WEBSRV must be able to negotiate the same IPSec parameters (ESP, SHA1) with the same preshared key as CORPSRV, or the two computers cannot set up SAs and communicate.

The filter for this rule specifies that IPSec-secured communications be permitted for all traffic between the IP addresses of the two servers. You might be able to enhance the security of this configuration by using a protocol-specific or a port-specific filter to provide much more restricted access to CORPSRV. For more information, see [“Modifying CORPSRV and WEBSRV Policy Design to Defend against Perimeter Network Server \(WEBSRV\) Compromise,”](#) later in this paper.

This rule uses preshared key for IKE authentication between CORPSRV and WEBSRV because it is the most appropriate authentication method for this scenario. The preshared key value that is configured for this rule applies only to the specific communication path between CORPSRV and WEBSRV. If the preshared key is compromised on one computer, it does not allow IKE authentication with any other computer.

Although the use of preshared keys is generally not recommended for IKE authentication, a preshared key is recommended for this particular rule, for the following reasons:

- Typically, you cannot use Kerberos authentication in this scenario because WEBSRV might be a stand-alone server (or a member of the perimeter network domain that is not fully trusted by CORPDC). Even if WEBSRV is a member of CORPDC or an internal domain that is fully trusted by CORPDC, it might be difficult to manage the firewall configuration that is required if WEBSRV uses Kerberos. The difficulty of firewall configuration in this scenario is due to Kerberos Key Distribution Center (KDC) location and referral behavior. Additionally, if you use Kerberos for authentication and if WEBSRV were compromised, an attacker might exploit the Kerberos trust relationship to gain IPSec-secured access to CORPSRV and additional internal corporate network computers, not just to CORPSRV.
- Although you can use certificate-based authentication in this scenario, doing so requires that you manually establish trust between the issuing root CA for the CORPSRV certificate and the issuing root CA for the WEBSRV certificate. To do this, you need to create a cross-certification trust relationship that is much broader than the trust relationship that you need, which creates a security risk if an attacker compromises WEBSRV because the attacker can then use the cross-certification trust on another computer. Because the filter for this rule uses the IP address of WEBSRV, any other computer must use the same IP address and authentication method as WEBSRV to gain IPSec-secured access to CORPSRV. But if an attacker compromises WEBSRV, it is important to use other security strategies to prevent the attacker from accessing other computers on the internal corporate network, rather than depending only on the fact that other computers do not have the same IP address as WEBSRV.

---

**Important** Make sure to use a properly generated preshared key that is unique to the IP address pair and filters that are specific to the IP addresses of WEBSRV and CORPSRV. To enhance the security that this rule provides, configure firewall filters to permit only ISAKMP, AH, and ESP traffic between the IP addresses of these two computers. For information about how to create these filters, see ["Creating Firewall Filters to Permit ISAKMP, AH, and ESP Traffic,"](#) later in this paper.

---

***CORPSRV Policy Rule 2: Block communication on path between CORPSRV and perimeter network subnets***

The following settings are configured for the second rule that is associated with the IPSec policy that is assigned to CORPSRV:

**IP filter list settings**

- **Name:** Me <-> Perimeter network subnets 172.16.1.0/24, 172.17.1.0/24, all traffic

**Filter 1 settings**

- **Source address:** My IP Address
- **Destination address:** A specific subnet (the first subnet)
  - **IP address:** 172.16.1.0
  - **Subnet mask:** 255.255.255.0
- **Mirrored:** Yes (This check box is selected.)
- **Protocol:** Any
- **Description:** Me <-> perimeter network subnet 172.16.1.0/24, all traffic

**Filter 2 settings**

- **Source address:** My IP Address
- **Destination address:** A specific subnet (the second subnet)
  - **IP address:** 172.17.1.0
  - **Subnet mask:** 255.255.255.0
- **Mirrored:** Yes (This check box is selected.)
- **Protocol type:** Any
- **Description:** Me <-> perimeter network subnet 172.17.1.0/24, all traffic

**Filter action settings**

- **Name:** Block all traffic
- **Action:** Block

**Authentication method, IPSec mode, and connection type**

- **Authentication method:** Kerberos (By default, Kerberos is specified, but it is not used because the filter action is to block traffic.)
- **Tunnel setting (IPSec mode):** This rule does not specify an IPSec tunnel (This option is selected, so transport mode is used.)
- **Connection type:** All network connections

---

This rule blocks traffic between CORPSRV and the perimeter network subnets 172.16.1.0/24 and 172.17.1.0/24. This rule is straightforward; it states that the IPSec driver on CORPSRV does not permit any traffic that is sent to or received from these subnets (with the exception of multicast, broadcast, and ISAKMP traffic, if the network allows ISAKMP traffic).

***CORPSRV Policy Rule 3: Secure communication on path between CORPSRV and perimeter network domain controller***

The following settings are configured for the third rule that is associated with the IPSec policy that is assigned to CORPSRV:

**IP filter list settings**

- **Name:** Me <-> perimeter network DC, all traffic

**Filter settings**

- **Source address:** My IP Address
- **Destination address:** A specific IP Address (where the IP address is the IP address of the perimeter network domain controller)
- **Mirrored:** Yes (This check box is selected.)
- **Protocol:** Any
- **Description:** Me <-> perimeter network DC, all traffic

The **NoDefaultExempt** registry key is set on CORPSRV to ensure that Kerberos traffic is not exempt from IPSec filtering. Because this filter matches all unicast traffic between CORPSRV and the perimeter network domain controller, it also includes any IPSec-secured Kerberos traffic. Remember to use the same **NoDefaultExempt** registry key setting on the perimeter network domain controller. This rule specifies that all traffic to the perimeter network domain controller be secured by IPSec for the following reasons: Kerberos uses TCP and UDP port 88, the domain controller locator service uses an unauthenticated LDAP query on UDP port 389, and you might need to secure ICMP traffic for testing or diagnostics.

**Filter action settings**

- **Name:** ESP 3DES-SHA1 required
- **Action:** Negotiate security
- **Accept unsecured traffic, but always respond using IPSec:** No (This check box is cleared, therefore inbound passthrough is not allowed.)
- **Allow unsecured communication with non-IPSec aware computers:** No (This check box is cleared, therefore fall back to clear is not allowed.)
- **Perfect forward secrecy (PFS):** No (This check box is cleared.)
- **Security method:** Custom
  - **Data and address integrity without encryption (AH):** No (This check box is cleared.)
  - **Data integrity and encryption (ESP):** Yes (This check box is selected.)

- 
- **Integrity algorithm: SHA1**
  - **Encryption algorithm: 3DES**
  - **Session key settings (key lifetimes):** 100,000 KB (100 MB) / 3,600 seconds (one hour)

Note that the filter action settings for this rule are the same settings described in Rule 1. A new filter action is not specified for this rule.

**Authentication method, IPSec mode, and connection type**

- **Authentication method: Preshared key**

---

**Important** As with Rule 1, this rule specifies that a preshared key be used for IKE authentication. Using preshared key authentication in this case provides the most narrow trust relationship possible that allows you to use IPSec-secured communication. Using a preshared key for authentication also ensures that IPSec can help secure the Kerberos protocol to the perimeter network domain controller, if a trust relationship is configured between the internal network domain and the perimeter network domain.

---

- Tunnel setting (IPSec mode): **This rule does not specify an IPSec tunnel** (This option is selected, so transport mode is used.)
- **Connection type: All network connections**

This rule helps secure all of the traffic between CORPSRV and the perimeter network domain controller with ESP encryption, SHA1 integrity hashing, and a preshared key for authentication. This rule does not allow fall back to clear or unsecured communication. This rule might seem contradictory with Rule 2, which blocks all traffic between the perimeter network subnets, however, both rules are valid because the Microsoft implementation of IPSec uses a weighting system to determine which filter is applied to traffic that is processed by the IPSec driver. More specific filters have a greater weight than less specific filters. The IPSec driver searches for the filter with the greatest weight when determining which filter to apply. Because filters that specify a specific IP address have a greater weight than filters that specify an entire subnet, the filters that match a specific IP address are applied.

This rule is included in the example CORPSRV policy to illustrate how to configure IPSec policy correctly if WEBSRV is a member of the CORPDC domain or a mutually trusted domain in the perimeter network and if CORPSRV is required to communicate with WEBSRV. Although it is strongly recommended that you do not use an internal domain member in a perimeter network, some scenarios require this architecture. For example, applications that run on servers in a perimeter network domain might need to initiate connections to servers in the internal corporate network domain to retrieve information.

Another example of when you might use an internal domain in a perimeter network is if users on computers in an internal corporate network domain need to post information that can be accessed by users on computers in the perimeter network domain. In this case, explicit, external one-way trusts must be configured so that the perimeter network domain trusts the internal corporate network domain. When this type of trust relationship is configured, most upper-layer protocols on CORPSRV will negotiate to use NTLM passthrough authentication rather than Kerberos authentication so that CORPSRV does not require a direct communication path to the domain controllers in the perimeter network. Using NTLM passthrough authentication, CORPSRV

---

passes an authentication request to a domain controller in the internal corporate network, which then contacts a domain controller in the perimeter network. Therefore, this rule might be necessary only if Kerberos is required to authenticate an upper-layer protocol. When Kerberos is required to authenticate an upper-layer protocol, CORPSRV might have to obtain a ticket from the perimeter domain controller to authenticate to WEBSRV, regardless of whether IPsec is used. If a ticket is required, you can use IPsec in transport mode to encapsulate communications between CORPSRV and the perimeter network domain controller to enhance security and to simplify firewall traversal.

This rule is not required if CORPSRV does not need to communicate with domain controllers in the perimeter network. In many network designs, the internal corporate network domain does not trust the perimeter network domain. In such cases, there is no reason for CORPSRV to communicate with domain controllers in the perimeter network. Domain controllers in a perimeter network might not be as well protected against network-based attacks as other computers in the perimeter network. Accordingly, there is a substantial risk associated with allowing the perimeter domain controller to establish IPsec-secured communications with computers in the internal corporate network. If you do configure this rule for the CORPSRV IPsec policy, make sure that you configure a complementary rule for the IPsec policy that is assigned to the perimeter network domain controller so that the domain controller can accept the security negotiation with CORPSRV.

---

**Note** The IPsec policy design for the perimeter network domain controller is not included in this paper.

---

***CORPSRV Policy Rule 4: Permit all communication on path between CORPSRV and CORPDC***

The following settings are configured for the fourth rule that is associated with the IPsec policy that is assigned to CORPSRV:

**IP filter list settings**

- **Name:** Me <-> CORPDC, all traffic

**Filter settings**

- **Source address:** My IP Address
- **Destination address:** A specific IP Address (the IP address of CORPDC)
- **Mirrored:** Yes (This check box is selected.)
- **Protocol:** Any
- **Description:** Me <-> CORPDC, all traffic

**Filter action settings**

- **Name:** Permit unsecured IP packets to pass through
- **Action:** Permit

**Authentication method, IPsec mode, and connection type**

- **Authentication method:** Kerberos (By default, Kerberos is specified, but it is not used because the filter action is to permit traffic.)

- 
- Tunnel setting (IPSec mode): **This rule does not specify an IPSec tunnel** (This option is selected, so transport mode is used.)
  - **Connection type: All network connections**

This rule allows all traffic between CORPSRV and CORPDC to be sent unsecured. Although this rule might seem unusual, using IPSec to help secure traffic to the domain controllers of the domain in which CORPSRV is a member can result in a number of problems. If domain members were to use IPSec-secured communication with domain controllers, increased latency might occur, causing domain controller location and authentication to fail. Also, complex IPSec policy configuration and management is required, and 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. For these reasons, it is currently not recommended that you use IPSec to help secure traffic between domain members (either clients or servers) and their domain controllers. For more information, see, "[Special IPSec Considerations](http://go.microsoft.com/fwlink/?LinkId=19223)" at <http://go.microsoft.com/fwlink/?LinkId=19223> and article 254949, "Client-to-Domain Controller and Domain Controller-to-Domain Controller IPSec Support," in the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

An attacker who spoofs the source IP address of the domain controller might use the static permit filters defined in this rule for the IP address of each domain controller to inject packets into the server (the IP addresses of the domain controllers are easily found using DNS). However, because bidirectional traffic is usually not possible from a spoofed source IP address, any reply traffic from CORPSRV will be routed by the network to CORPDC, not back to CORPATTACK. You can provide a partial mitigation for inbound spoofing attacks on CORPSRV by configuring inbound, blocking filters to well-known ports from the CORPDC IP addresses. However, you cannot configure port-specific inbound blocking filters for dynamic ports.

To help mitigate inbound spoofing, you might consider configuring an IPSec policy to allow fall back to clear when attempting to negotiate security to the IP address of CORPDC. However, this is not recommended because the three-second delay can cause operational problems with domain authentication. For example, if you configure a filter action that controls traffic to the domain controller to allow fall back to clear, CORPSRV might experience intermittent errors when authenticating domain accounts. Such errors might occur because the client domain controller attempts to contact the IP addresses of all corporate domain controllers if CORPDC does not respond within one-tenth of a second, and there is a three-second interval for the security negotiation attempt to fall back to clear. Although CORPSRV might not experience authentication failures in a domain with only one or two domain controllers, for a larger domain with more domain controllers, the time that it takes to contact a domain controller increases, and, therefore, authentication failures are more likely. For these reasons, it is not recommended that you enable the option to fall back to clear for domain controller communications.

In an environment where there are multiple domain controllers in the same domain (or when a single domain controller has multiple IP addresses), you must limit the filter list for this rule to include only the IP addresses of the domain controllers in the domain. Accordingly, the domain controllers must have static IP addresses, and you must update the filter list if the IP address of any domain controller changes. Typically, domain members running Windows 2000 or Windows XP attempt to contact each domain controller in the list until they locate a functioning domain

controller. As a result, some lack of synchronization between the domain controller IP addresses in the filter list and those used in DNS or WINS is acceptable.

***CORPSRV Policy Rule 5: Secure communication on path between CORPSRV and any computer***

The following settings are configured for the fifth rule that is associated with the IPSec policy that is assigned to CORPSRV:

---

**Important** This rule is the most critical rule for securing the server from untrusted attacks.

---

**IP filter list settings**

- **Name:** Me <-> any, all traffic

**Filter settings**

- **Source address:** My IP Address
- **Destination address:** Any IP Address
- **Mirrored:** Yes (This check box is selected.)
- **Protocol:** Any
- **Description:** Me <-> any, all traffic

This mirrored filter matches all unicast inbound and outbound IP packets, except for traffic that matches filters from other rules and traffic that matches any default exemption to filtering. The use of **My IP Address** as the source address in this filter ensures that the IPSec service creates specific filters for each of the IP addresses used by the CORPSRV network adapters. Additionally, if you add, change, or remove IP addresses or network adapters, the IPSec service automatically creates new filters or modifies or deletes filters, as needed, to update the policy. If you specify **A specific subnet** as the source address and if an IP address is not already configured when the IPSec policy is applied, then the correct filters might not be applied (note that although this behavior is a known issue in Windows 2000, it is not an issue in Windows XP or in Windows Server 2003).

For this example policy, the **NoDefaultExempt** registry key value is set to **1**, as described in [“Removing the Default Filtering Exemption for Kerberos and RSVP Traffic,”](#) earlier in this paper. Remember that when this registry key value is used, only ISAKMP, multicast, and broadcast traffic are exempt from filtering. If you specify **Any IP Address** as the destination address, then all IP addresses are still subject to these exemptions. For information about the risks that are associated with these exemptions, see article 811832, “IPSec Default Exemptions Can Be Used to Bypass IPSec Protection in Some Scenarios,” in the [Microsoft Knowledge Base](#), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

Windows Server 2003 IPSec is currently the only version of IPSec that provides the ability to block broadcast and multicast traffic. Windows 2000 IPSec and Windows XP IPSec do not support the use of a filter with a source address of **Any IP Address** and a destination address of **Any IP Address**. Windows Server 2003 IPSec provides limited support for the use of this filter. In Windows Server 2003 IPSec, a filter with a source address of **Any IP Address** and a destination address of **Any IP Address** can block multicast and broadcast traffic. For more information, see article 810207, “IPSec Default Exemptions Are Removed in Windows Server 2003,” in the [Microsoft Knowledge Base](#), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

---

## Filter action settings

- **Name:** ESP 3DES-SHA1 required.
- **Action:** Negotiate security.
- **Accept unsecured traffic, but always respond using IPSec:** If this check box is selected, inbound passthrough is allowed. You can select this check box based on whether CORPCLI Policy 1 or CORPCLI Policy 2 is assigned to the client. Because this rule in the CORPSRV policy affects all CORPCLI communication, there are two designs for the CORPCLI policy, depending on whether you allow inbound passthrough in the CORPSRV policy. For information about the CORPCLI policies, see [“Designing IPSec Policies for the Client \(CORPCLI\),”](#) later in this paper.

If you use CORPCLI Policy 1, you should not select the **Accept unsecured traffic, but always respond using IPSec** check box. If you use CORPCLI Policy 2, you can select this check box. When you select this check box, this rule is compatible with CORPCLI Policy 2 Rule 2 (for more information, see “CORPCLI Policy 2 Rule 2: Default response rule,” in [“Designing IPSec Policies for the Client \(CORPCLI\),”](#) later in this paper). When this check box is selected, CORPSRV can receive unsecured traffic, but security is negotiated for outbound traffic. If an attacker sends traffic to CORPSRV, security is negotiated for any attempted response. The security of the response is determined by whether the **Allow unsecured communication with non-IPSec-aware computers** check box is also selected. If the client responds and the security negotiation fails, the response is blocked, and bidirectional communication between the client and server fails.

When you select the **Accept unsecured traffic, but always respond using IPSec** check box, CORPSRV can receive unsecured traffic that matches the associated filters in the rule. Accordingly, selecting this check box opens all ports on CORPSRV and leaves this server vulnerable to anonymous denial-of-service attacks from any IP address. The most secure configuration, therefore, requires this check box to be cleared.

If you clear the **Accept unsecured traffic, but always respond using IPSec** check box, then you must assign CORPCLI Policy 1 to the client because CORPCLI must successfully complete the security negotiation to gain IPSec-secured access to CORPSRV. When CORPCLI Policy 1 is assigned to the client, any unsecured traffic that the client receives is dropped, except traffic that matches other permit filters or traffic that matches the default exemptions to filtering.

Before you configure this setting, evaluate the tradeoff between the ease of client configuration and the security of the server from anonymous inbound attacks. When you clear this check box, you must assign a client policy that requires additional configuration, but you also enhance the security of CORPSRV against anonymous inbound attacks.

- **Allow unsecured communication with non-IPSec aware computers:** No (This check box is cleared, therefore fall back to clear is not allowed.) In this example policy, fall back to clear is not allowed, so IPSec must help secure TCP and UDP unicast IP traffic between CORPSRV and any destination. Therefore, server administrators and applications on CORPSRV can only initiate TCP or UDP unicast connections to destinations that can successfully negotiate IKE to establish IPSec SAs.

- 
- **Perfect forward secrecy (PFS):** No (This check box is cleared.).
  - **Security method:** Custom.
    - **Data and address integrity without encryption (AH):** No (This check box is cleared.).
    - **Data integrity and encryption (ESP):** Yes (This check box is selected.).
      - **Integrity algorithm:** SHA1
      - **Encryption algorithm:** 3DES
    - **Session key settings (key lifetimes):** 100,000 KB (100 MB) / 3,600 seconds (one hour).

#### **Authentication method, IPSec mode, and connection type**

- **Authentication method:** Kerberos
- Tunnel setting (IPSec mode): **This rule does not specify an IPSec tunnel** (This option is selected, therefore transport mode is used.)
- **Connection type:** All network connections

This rule accounts for all traffic between CORPSRV and any other computer that the filters in the IPSec policy do not address. The rule specifies that all communication with CORPSRV be secured with ESP (using SHA1 integrity hashing and 3DES encryption). Accordingly, the computer on the other end of the communication must support the same data integrity hashing and encryption options (SHA1 and 3DES), or the security negotiation between CORPSRV and that computer fails.

#### ***CORPSRV Policy Rule 6: Permit unsecured ICMP communication between CORPSRV and default gateway***

The following settings are configured for the sixth rule that is associated with the IPSec policy that is assigned to CORPSRV:

##### **IP filter list settings**

- **Name:** Me <-> 192.168.2.1 (ROUTER3), ICMP only

##### **Filter settings**

- **Source address:** My IP Address
- **Destination address:** A specific IP Address (the IP address of ROUTER3)
- **Mirrored:** Yes (This check box is selected.)
- **Protocol:** ICMP
- **Description:** Me <-> default gateway, ICMP only

##### **Filter action settings**

- **Name:** Permit unsecured ICMP packets
- **Action:** Permit

#### **Authentication method, IPSec mode, and connection type**

- 
- **Authentication method:** Kerberos (By default, Kerberos is specified, but it is not used because the filter action is to permit traffic.)
  - Tunnel setting (IPSec mode): **This rule does not specify an IPSec tunnel** (This option is selected, so transport mode is used.)
  - **Connection type: All network connections**

This rule permits all unsecured ICMP traffic between CORPSRV and ROUTER3. This rule is necessary for the example scenario because the router does not use IPSec with computers in transport mode. ICMP traffic to the default gateway is allowed because Windows typically requires ICMP traffic for network discovery

***CORPSRV Policy Rule 7: Permit CORPSRV to receive but not to send unsecured ICMP communications***

The following settings are configured for the seventh rule that is associated with the IPSec policy that is assigned to CORPSRV:

**IP filter list settings**

- **Name:** Any -> me, ICMP one-way, for PMTU discovery

**Filter settings**

- **Source address: Any IP Address**
- **Destination address: My IP Address**
- **Mirrored:** No (This check box is cleared.)
- **Protocol: ICMP**
- **Description:** Any -> me, ICMP one-way, PMTU discovery

**Filter action settings**

- **Name:** Permit unsecured inbound ICMP
- Action: **Permit**

**Authentication method, IPSec mode, and connection type**

- **Authentication method:** Kerberos (By default, Kerberos is specified, but it is not used because the filter action is to permit traffic.)
- Tunnel setting (IPSec mode): **This rule does not specify an IPSec tunnel** (This option is selected, so transport mode is used.)
- **Connection type: All network connections**

This rule allows CORPSRV to receive ICMP Destination Unreachable messages so that TCP path maximum transmission unit (PMTU) discovery functions correctly. If IPSec-secured packets must traverse a link that is smaller than 1,500 bytes (for example, a VPN tunnel), PMTU discovery must function correctly for IPSec-secured TCP connectivity to be maintained. Although UDP and other protocols also use ICMP Destination Unreachable messages for PMTU discovery, the application that uses the protocol must be written with the ability to process these messages. TCP uses these ICMP messages, by default, to perform PMTU discovery.

---

In the IPSec policy configuration used in the example, CORPSRV can receive ICMP Echo Request messages in plaintext from any IP address on the internal corporate network. The corresponding ICMP Echo Reply message triggers a security negotiation with the initiator of the ICMP Echo Request message. If the initiator is assigned an IPSec policy that allows it to respond (for example, if the initiator is assigned an IPSec policy with the default response rule enabled or with a filter that is similar to the filter used in CORPSRV Rule 5), then IPSec connectivity is established and verified with an ICMP **ping** command. It is often useful to use the **ping** command to troubleshoot IPSec-secured connections. For more information, see [“Verifying IPSec-Secured Connectivity,”](#) later in this paper.

When you design an IPSec policy to help secure ICMP traffic, keep in mind that such a policy might cause services and tools that rely on ICMP to measure network response times to produce misleading results. The time that is required by IKE to negotiate security or to fall back to clear causes a delay in ICMP responses. Because applications and tools that measure network response times might wait only a very short time (a fraction of a second is typical) to receive an ICMP reply and cannot detect the delayed ICMP response, these applications and tools will report that the server has failed.

Similarly, the Tracert.exe command-line tool, and similar tools that depend on the processing of ICMP traffic by routers, might not function correctly. Tracert.exe sends ICMP Echo Request messages with incrementally increasing Time to Live (TTL) values to determine the path taken to a destination. If ICMP traffic is secured by IPSec, the traffic is encapsulated; the router, therefore, detects an IPSec packet, not the ICMP traffic that is within the IPSec encapsulation, and replies with an ICMP Time Exceeded message. This message includes only the first part of the IPSec-secured ICMP packet. This partial packet does not contain the information that the computer needs to properly interpret the ICMP Time Exceeded message. As a result, in this case, the Tracert.exe output shows only the destination computer, not the path of intermediate routers.

For these reasons, before you deploy IPSec in a production environment, conduct testing to verify whether PMTU discovery is required and whether ICMP can function correctly when it is secured by IPSec. If you decide to permit outbound ICMP traffic (that is, you allow ICMP traffic to be unsecured), then you must either mirror this filter or create another one-way filter to permit outbound ICMP traffic. Keep in mind that if you permit ICMP traffic, certain attacks are possible. For more information, see [“Configuring IPSec Policies: Security Considerations,”](#) later in this paper.

#### ***CORPSRV Policy Rule 8: Default response rule disabled***

To disable the default response rule on CORPSRV, on the **Rules** tab, in the properties for the CORPSRV policy, clear the **<Dynamic>** check box.

Because an explicitly defined filter to negotiate security takes precedence over the default response rule, the default response rule is not needed in this scenario, and therefore, it is disabled. If the source IP address of the IKE request in the perimeter network is the IP address of WEBSRV, then the IKE policy lookup matches Rule 1 because the filter for Rule 1 contains the IP address of WEBSRV. Rule 2 does not block any other perimeter network source IP address in the IKE request because the filter for Rule 2 blocks unsecured traffic. Instead, the IKE policy lookup matches Rule 5 because the filter for Rule 5 is defined to negotiate security with **Any IP Address**. There are other operational and security issues to consider, as well, before using the

---

default response rule. For more information, see [“Security Risks of Enabling the Default Response Rule,”](#) later in this paper.

### **Designing IPSec Policies for the Client (CORPCLI)**

You can use either of the following approaches when designing the IPSec policy for CORPCLI:

- Design an IPSec policy that allows CORPCLI to negotiate security with CORPSRV. In the IPSec policy examples that follow, this policy is referred to as CORPCLI Policy 1.
- Design an IPSec policy that enables the default response rule on CORPCLI so that CORPSRV security negotiation is accepted. In the IPSec policy examples that follow, this policy is referred to as CORPCLI Policy 2.

The following sections provide overviews of CORPCLI Policy 1 and CORPCLI Policy 2 and describe the policy general settings and policy rules that are associated with these two IPSec policies.

#### **CORPCLI Policy 1 Overview**

CORPCLI Policy 1 is compatible with the filter action for Rule 5 of the CORPSRV IPSec policy when you disable inbound passthrough (that is, when the **Accept unsecured traffic, but always respond using IPSec** check box is cleared). CORPCLI Policy 1 is different than CORPCLI Policy 2 because CORPCLI Policy 1 specifies that security is initiated and negotiated only with the static IP address of CORPSRV. Security is not negotiated with any other computer. Additionally, when security is negotiated, trust must be established with CORPSRV through Kerberos authentication, and only ESP 3DES/SHA1 is accepted. Finally, to enhance security, CORPCLI Policy 1 contains a rule to block unsecured traffic with the perimeter network subnets.

CORPCLI Policy 1 is more secure than CORPCLI Policy 2, which enables the default response rule to establish IPSec-secured communications with CORPSRV. However, it might not be possible to use CORPCLI Policy 1 because you might not be able to create a client IPSec policy that specifies a static IP address for a server. For more information about factors to consider when determining which CORPCLI policy to use, see “CORPSRV Policy Rule 5: Secure communication on path between CORPSRV and any computer” in [“Designing an IPSec Policy for the Server \(CORPSRV\),”](#) earlier in this paper.

#### **CORPCLI Policy 1 General Settings**

This section describes the general settings that are applied with CORPCLI Policy 1.

##### ***Configuring CORPCLI Policy 1 name, description, and policy refresh interval***

The following general settings are configured on the **General** tab of the properties for CORPCLI Policy 1:

- **Name:** CORPCLI Client IPSec Policy, Negotiate to CORPSRV, *VersionOrDate-TimeLastModified*.
- **Description:** Block perimeter network traffic. Require IPSec with ESP 3DES/SHA1 when communicating with CORPSRV. Kerberos authentication is preferred.

- 
- Policy refresh interval (**Check for policy changes every  $n$  minutes**): 90 minutes (one and a half hours).

The policy refresh interval for this IPsec policy is much shorter than it is for the IPsec policy that is assigned to CORPSRV because the CORPCLI policy must contain the IP address of CORPSRV. If the IP address that is assigned to CORPSRV changes, you must update the CORPCLI policy.

Typically, a client policy update adds the new IP address of a server one day before the IP address of the server actually changes. This delay allows for Active Directory replication of the new IP address and retrieval by all domain clients. In a single domain in which all clients are connected, the update occurs within one and a half hours. For large domains and forests, the update occurs within one day. A short policy refresh interval also allows you to quickly change a policy (for example, to block unwanted traffic caused by worm attacks or application errors). For information about how to configure an IPsec policy to block ports on Active Directory domain members, see article 813878 "How to Block Specific Network Protocols and Ports by Using IPsec," in the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

#### ***Configuring CORPCLI Policy 1 key exchange settings***

The following general settings are configured in the **Key Exchange Settings** dialog box of the properties for CORPCLI Policy 1:

- Key lifetime (**Authenticate and generate a new key after every  $n$  minutes**): 480 minutes (eight hours)
- Number of quick mode negotiations per main mode (**Authenticate and generate a new key after every  $n$  sessions**): 0 (An unlimited number of session keys can be created from the master key keying material.)

#### ***Configuring CORPCLI Policy 1 key exchange security methods***

The following general settings are configured in the **Key Exchange Security Methods** dialog box of the properties for CORPCLI Policy 1. These security methods are listed in order of preference, by encryption algorithm, integrity algorithm, and Diffie-Hellman group.

- 3DES/SHA1/Medium (Group 2, 1024 bits)
- 3DES/MD5/Medium (Group 2, 1024 bits)

---

**Note** The key exchange settings and key exchange security methods for the IPsec policy that is assigned to CORPCLI must be compatible with the settings in the IPsec policy that is assigned to CORPSRV. If the settings are not compatible, the security negotiation between these computers fails.

---

## CORPCLI Policy 1 Rules

Table 3 lists the three rules that are required for CORPCLI Policy 1. This policy blocks perimeter network traffic, and it allows CORPCLI to negotiate security with CORPSRV. Security negotiation is initiated with ESP 3DES/SHA1 when CORPCLI communicates with CORPSRV. Kerberos authentication is used.

Each of the CORPCLI Policy 1 rules is described in more detail in the following sections.

**Table 3 Summary of CORPCLI Policy 1 Rules**

Rule	Filter Settings	Filter Action Settings	Authentication Method
1: <a href="#">Block communications on path between Me and perimeter network subnets</a>	Me <-> Perimeter network subnets 172.16.1.0/24, 172.17.1.0/24, all traffic, mirrored	Block	N/A
2: <a href="#">Secure communication to CORPSRV</a>	Me <-> CORPSRV, all traffic, mirrored	Require ESP 3DES/SHA1, no clear, no fall back to clear	Kerberos
3: <a href="#">Default response rule disabled</a>	Default response rule	Disabled	N/A

### *CORPCLI Policy 1 Rule 1: Block communications on path between Me and perimeter network subnets*

The following settings are configured for the first rule that is associated with CORPCLI 1 Policy 1:

#### **IP filter list settings**

- **Name:** Me <-> Perimeter network subnets 172.16.1.0, 172.17.1.0, all traffic

#### **Filter 1 settings**

- **Source address:** My IP Address
- **Destination address:** A specific subnet (the first perimeter network subnet)
  - **IP address:** 172.16.1.0
  - **Subnet mask:** 255.255.255.0
- **Mirrored:** Yes (This check box is selected.)
- **Protocol type:** Any
- **Description:** Me <-> perimeter network subnet 172.16.1.0/24, all traffic

#### **Filter 2 settings**

- **Source address:** My IP Address

- 
- **Destination address: A specific subnet** (the second perimeter network subnet)
    - **IP address:** 172.17.1.0
    - **Subnet mask:** 255.255.255.0
  - **Mirrored:** Yes (This check box is selected.)
  - **Protocol:** Any
  - **Description:** Me <-> perimeter network subnet 172.17.1.0/24, all traffic

#### Filter action settings

- **Name:** Block all traffic
- Action: **Block**

#### Authentication method, IPSec mode, and connection type

- **Authentication method:** Kerberos (By default, Kerberos is specified, but it is not used because the filter action is to block traffic.)
- Tunnel setting (IPSec mode): **This rule does not specify an IPSec tunnel** (This option is selected, so transport mode is used.)
- **Connection type:** All network connections

This rule blocks traffic between CORPCLI and the perimeter network subnets 172.16.1.0/24 and 172.17.1.0/24. As in Rule 2 of the IPSec policy that is assigned to CORPSRV, if the client receives traffic from computers in the perimeter network subnet address ranges, the client discards the packets. Likewise, the IPSec filters discard traffic that the client attempts to send to computers in the perimeter network subnet address ranges.

Unlike a firewall, when an IPSec policy is assigned, Windows IPSec does not block traffic by default. Instead, you must create a filter action to determine whether traffic should be permitted, blocked, or IPSec-secured. This rule provides defense-in-depth against misconfiguration or compromise of the firewall that would otherwise allow computers in the perimeter network to send unsecured traffic to the clients in the internal corporate network.

#### *CORPCLI Policy 1 Rule 2: Secure communication to CORPSRV*

The following settings are configured for the second rule that is associated with CORPCLI Policy 1:

#### IP filter list settings

- **Name:** Me <-> CORPSRV IP address, all traffic

#### Filter settings

- **Source address:** My IP Address
- **Destination address:** A specific IP Address (the IP address of CORPSRV)
- **Mirrored:** Yes (This check box is selected.)
- **Protocol type:** Any

- 
- **Description:** Me <-> CORPSRV, all traffic

---

**Note** If you configure a filter that contains the IP address of a specific server in the IPsec policy that is assigned to the client, the server must have a static IP address. If you plan to change the IP address of the server, then first add the new IP address (but do not delete the existing address) to the IPsec policy that is assigned to the client. You should update the client policy first to allow time for propagation of the new server IP address to all domain members. Alternatively, you can use IPsec to help secure every server on a subnet, rather than a single server. For more information, see [“Modifying IPsec Policy Designs in this Scenario,”](#) later in this paper.

---

#### Filter action settings

- **Name:** ESP 3DES-SHA1 required
- Action: **Negotiate security**
- **Accept unsecured traffic, but always respond using IPsec:** No (This check box is cleared, therefore inbound passthrough is not allowed.)
- **Allow unsecured communication with non-IPsec aware computers:** No (This check box is cleared, therefore fall back to clear is not allowed.)
- **Perfect forward secrecy (PFS):** No (This check box is cleared.)
- **Security method: Custom**
  - **Data and address integrity without encryption (AH):** No (This check box is cleared.)
  - **Data integrity and encryption (ESP):** Yes (This check box is selected.)
    - **Integrity algorithm: SHA1**
    - **Encryption algorithm: 3DES**
  - **Session key settings (key lifetimes):** 100,000 KB (100 MB) / 3,600 seconds (one hour)

#### Authentication method, IPsec mode, and connection type

- **Authentication method: Kerberos**
- Tunnel setting (IPsec mode): **This rule does not specify an IPsec tunnel** (This option is selected, so transport mode is used.)
- **Connection type: All network connections**

This rule is required for clients to initiate a request to negotiate security with CORPSRV. The rule requires that an active IPsec policy be assigned to CORPSRV and that CORPSRV can respond successfully to the CORPCLI request to negotiate security. Before clients can establish any other TCP/IP communications with CORPSRV, successful IPsec SAs must first be established.

---

### ***CORPCLI Policy 1 Rule 3: Default response rule disabled***

Finally, you must disable the default response rule on CORPCLI by clearing the **<Dynamic>** check box, on the **Rules** tab, in the policy properties. It is recommended that you disable the default response rule when any of the following is true:

- The server uses a static IP address.
- All traffic, including the first packet of the upper-layer protocol, is secured by IPsec.
- The **Accept unsecured traffic, but always respond using IPsec** check box is not selected for the IPsec policy that is assigned to the server.
- Outbound ICMP or other traffic is not secured by IPsec.
- Security considerations require no IKE response to IKE probes.

### **CORPCLI Policy 2 Overview**

CORPCLI Policy 2 is compatible with the filter action for Rule 5 of the CORPSRV IPsec policy when you enable inbound passthrough (that is, when you select the **Accept unsecured traffic, but always respond using IPsec** check box). CORPCLI Policy 2 is different than CORPCLI Policy 1 because CORPCLI Policy 2 specifies that you enable the default response rule to allow IPsec-secured communications with CORPSRV. When CORPCLI Policy 2 is used, clients do not initiate a request to negotiate security with the IP address of CORPSRV. Instead, the default response rule is enabled so that clients can respond to any request to negotiate security, including requests from CORPSRV. When security is negotiated, trust must be established through Kerberos authentication, and only ESP 3DES/SHA1 is accepted.

As with CORPCLI Policy 1, to enhance security, CORPCLI Policy 2 contains a rule to block unsecured traffic between CORPCLI and the perimeter network subnets. For more information about factors to consider when determining which CORPCLI policy to use, see “CORPSRV Policy Rule 5: Secure communication on path between CORPSRV and any computer,” in [“Designing an IPsec Policy for the Server \(CORPSRV\),”](#) earlier in this paper.

### **CORPCLI Policy 2 General Settings**

This section describes the general settings that are applied with CORPCLI Policy 2. These settings are different than the CORPCLI Policy 1 general settings only in name, description, and refresh interval.

#### ***Configuring CORPCLI Policy 2 name, description, and policy refresh interval***

The following general settings are configured on the **General** tab of the properties for CORPCLI Policy 2:

- **Name:** CORPCLI Client IPsec Policy 2, default response, *VersionOrDate-TimeLastModified*.
- **Description:** Block perimeter network traffic. Respond to any request to negotiate security. Accept ESP 3DES/SHA1 only if trust is established with computer through Kerberos authentication.
- Policy refresh interval (**Check for policy changes every *n* minutes**): 90 minutes (one and a half hours).

---

### Configuring CORPCLI Policy 2 key exchange settings

The following general settings are configured in the **Key Exchange Settings** dialog box of the properties for CORPCLI Policy 2:

- Key lifetime (**Authenticate and generate a new key after every  $n$  minutes**): 480 minutes (eight hours)
- Number of quick mode negotiations per main mode (**Authenticate and generate a new key after every  $n$  sessions**): 0 (An unlimited number of session keys can be created from the master key keying material.)

### Configuring CORPCLI Policy 2 key exchange security methods

The following general settings are configured in the **Key Exchange Security Methods** dialog box of the properties for CORPCLI Policy 2. These security methods are listed in order of preference, by encryption algorithm, integrity algorithm, and Diffie-Hellman group.

- 3DES/SHA1/Medium (Group 2, 1024 bits)
- 3DES/MD5/Medium (Group 2, 1024 bits)

---

**Note** The key exchange settings and key exchange security methods for the policy that is assigned to CORPCLI must be compatible with the settings in the IPsec policy that is assigned to CORPSRV, or the security negotiation between these computers fails.

---

### CORPCLI Policy 2 (Default Response Policy) Rules

Table 4 lists the two rules that are required for CORPCLI Policy 2. This policy blocks perimeter network traffic. CORPCLI responds to any request to negotiate security and accepts ESP 3DES/SHA1 only if trust is established with the other computer through Kerberos authentication.

When this policy is used, CORPCLI does not initiate a request to negotiate security with CORPSRV to help secure the initial communications between the two computers.

Each of the CORPCLI Policy 2 rules is described in more detail in the following sections.

**Table 4 Summary of CORPCLI Policy 2 Rules**

Rule	Filter Settings	Filter Action Settings	Authentication Method
1: <a href="#">Block communications on path between Me and perimeter network subnets</a>	Me< -> Perimeter network subnets 172.16.1.0/24, 172.17.1.0/24, all traffic, mirrored	Block	N/A
2: <a href="#">Default response rule enabled</a>	Default response rule	Enabled, accept only ESP 3DES/SH A1	Kerberos

---

***CORPCLI Policy 2 Rule 1: Block communications on path between Me and perimeter network subnets***

The following settings are configured for the first rule that is associated with CORPCLI Policy 2:

**IP filter list settings**

- **Name:** Me <-> Perimeter network subnets 172.16.1.0, 172.17.1.0, all traffic

**Filter 1 settings**

- **Source address: My IP Address**
- **Destination address: A specific subnet** (the first perimeter network subnet)
  - **IP address:** 172.16.1.0
  - **Subnet mask:** 255.255.255.0
- **Mirrored:** Yes (This check box is selected.)
- **Protocol type:** Any
- **Description:** Me <-> perimeter network subnet 172.16.1.0/24, all traffic

**Filter 2 settings**

- **Source address: My IP Address**
- **Destination address: A specific subnet** (the second perimeter network subnet)
  - **IP address:** 172.17.1.0
  - **Subnet mask:** 255.255.255.0
- **Mirrored:** Yes (This check box is selected.)
- **Protocol:** Any
- **Description:** Me <-> perimeter network subnet 172.17.1.0/24, all traffic

**Filter action settings**

- **Name:** Block all traffic
- **Action:** **Block**

**Authentication method, IPSec mode, and connection type**

- **Authentication method:** Kerberos (By default, Kerberos is specified, but it is not used because the filter action is to block traffic.)
- **Tunnel setting (IPSec mode):** **This rule does not specify an IPSec tunnel** (This option is selected, so transport mode is used)
- **Connection type:** **All network connections**

This rule blocks communications between CORPCLI and perimeter network subnets. As in Rule 2 of the IPSec policy that is assigned to CORPSRV, if the client receives traffic from computers in the perimeter network subnet address ranges, the client discards the packets. Likewise, the

---

IPSec filters discard traffic that the client attempts to send to computers in the perimeter network subnet address ranges.

***CORPCLI Policy 2 Rule 2: Default response rule enabled***

The following settings are configured for the second rule that is associated with CORPCLI Policy 2. Although you cannot configure settings for filters and filter actions for the default response rule, you can configure security methods and their preference order and authentication methods and their preference order.

**IP filter list settings**

- <Dynamic> (cannot be configured)

**Filter settings**

- <Dynamic> (cannot be configured)

**Filter action settings**

- **Name:** Default Response
- **Action:** <Default Response> (This setting cannot be configured; the **Negotiate security** filter action is automatically used.)
- **Security method:** Custom
  - **Data and address integrity without encryption (AH):** No (This check box is cleared.)
  - **Data integrity and encryption (ESP):** Yes (This check box is selected.)
    - **Integrity algorithm:** SHA1
    - **Encryption algorithm:** 3DES
  - **Session key settings (key lifetimes):** 100,000 KB (100 MB) / 3,600 seconds (one hour)

**Authentication method, IPSec mode, and connection type**

- **Authentication method:** Kerberos
- **Tunnel setting (IPSec mode):** **This rule does not specify an IPSec tunnel** (This option is selected, so transport mode is used.)
- **Connection type:** **All network connections**

This rule specifies that CORPCLI responds to requests for secure communication and that CORPCLI helps secure traffic that is sent from or received only by computers that use ESP encryption, SHA1 integrity hashing, and Kerberos for authentication. Because this IPSec policy does not have other rules with filters that require IPSec to help secure traffic, outbound traffic from CORPCLI does not trigger a security negotiation. As a result, CORPCLI can send TCP, UDP, ICMP and other traffic to computers on the internal corporate network subnet address ranges just as it would if it were not assigned an IPSec policy. However, when CORPCLI sends a packet to CORPSRV, and when the **Allow unsecured communication, but always respond using IPSec** check box is selected for Rule 5 of the CORPSRV IPSec policy, then CORPSRV

---

attempts to negotiate security with CORPCLI. Because the CORPCLI IPsec policy does not include more specific filters to control CORPCLI's response to CORPSRV, CORPCLI uses the default response rule settings to negotiate security with CORPSRV. These settings indicate that CORPCLI can trust only domain members and can allow IPsec only if the computer on the other end of the communication (CORPSRV) uses ESP 3DES and SHA1.

Authentication, in this case, occurs in the opposite direction than if other client-server protocols are used. In this case, the IKE protocol on CORPSRV uses Kerberos authentication to request a Kerberos ticket for the client. Typically, one-way Kerberos trust environments are configured so that the servers can trust the client domains. In such environments, the IKE protocol on the server cannot obtain a Kerberos ticket to authenticate to the client, and security negotiation, therefore, fails. Also, the network paths must be open so that the Kerberos protocol on the server can contact all of the DNS servers and domain controllers that are required to process the Kerberos ticket request. As noted earlier in this paper, it is recommended that you use Kerberos authentication in a mutual domain trust environment. In a one-way trust environment, use certificate authentication instead, or preshared keys, if necessary.

If CORPSRV and CORPCLI establish mutual trust during the main mode negotiation, then CORPSRV proposes a filter to CORPCLI during the quick mode negotiation. The filter proposed by CORPSRV in Rule 5 of the CORPSRV policy helps secure all traffic between the IP address of CORPCLI and the IP address of CORPSRV. The security methods specified for the default response filter action used by CORPCLI must match the security methods specified for the filter action in Rule 5 of the CORPSRV IPsec policy (ESP 3DES and SHA1 required). If these security settings match, then CORPCLI dynamically creates and installs an All Traffic filter to use IPsec to help secure all traffic to the IP address of CORPSRV. IPsec helps secure traffic in both directions, between CORPSRV and CORPCLI, for all protocols.

Therefore, if CORPSRV sends traffic that is not secured by IPsec, CORPCLI discards these packets. For this reason, the IPsec policy that is assigned to CORPSRV must not exempt the types of outbound traffic that CORPCLI should receive. For example, the CORPSRV policy might include an additional rule to permit inbound access to a DNS service and would, therefore, include a filter to permit UDP and TCP traffic on port 53. When CORPCLI responds using the default response rule, the All Traffic filter, which is created dynamically, blocks unsecured DNS replies that are sent from CORPSRV.

The lifetime of the default response filter for CORPCLI does not correspond to the lifetimes of the IKE main mode SA and IPsec SAs. By default, IPsec SAs are deleted if they are idle for five minutes. Because TCP connections can live for two hours without any traffic, the default response filter remains in place for about two hours. This time is implemented in the IPsec driver and therefore cannot be modified. If you manually stop the IPsec service on CORPSRV, IKE sends delete messages to CORPCLI to ensure that all SAs are removed. CORPCLI, however, maintains the default response filter and continues to attempt to negotiate security with CORPSRV on demand, if upper-layer protocol traffic is sent to CORPSRV. When CORPSRV does not reply, the security negotiation attempts fail, which prevents upper-layer protocol communication. Communication between CORPCLI and CORPSRV can be cut off for two hours if the IPsec service is stopped and remains stopped on CORPSRV. If the IPsec service is restarted, all clients in CORPCLI that still have default response filters renegotiate IPsec SAs, and connectivity is restored. To remove default response filters and to restore unsecured

---

connectivity with CORPSRV, you must stop the IPSec service on CORPCLI as well. Therefore, you should not stop the IPSec service on the server when clients are using the default response rule to negotiate security with the server.

For these reasons, the default response policy is typically used for a client when several of the following are true:

- The server uses a dynamic IP address.
- The server uses a static IP address, but you do not want to manage client policy changes for the server IP address or changes in the IP address of the server.
- There are hundreds or thousands of servers that are secured by IPSec, making it administratively impractical to manage a client policy that contains the IP addresses of all of the servers.
- The IPSec service will not be stopped administratively.
- There is not a need to exempt specific traffic (for example, ICMP or DNS) from IPSec filtering on the client or the server.
- The first upper-layer protocol packets sent from a client to the server do not contain sensitive information that must be secured.

### **Modifying IPSec Policy Design in this Scenario**

The example IPSec policies for CORPSRV and CORPCLI can include filters that help secure all unicast IP traffic between two computers, as already described. To broaden the scope of these policies, you can create filters that help secure multiple servers, rather than just one server. Or, you can narrow the scope of these policies by creating filters that help secure only specific types of traffic between a client and a server.

For example, you can use IPSec to help secure every server on a subnet, rather than a single server. In CORPCLI Policy 1, the specific IP address of the server is used because only CORPSRV is secured by IPSec in this scenario. If you need to use IPSec to help secure every server on a subnet, then you can configure one filter for the entire subnet rather than one filter for the IP address of each server.

You can create specific filters for the client IPSec policy to help secure just one type of connection with the server, or you can create filters on the client for all traffic. For example, the IPSec policy for the client and the server might include a filter to help secure TCP traffic over port 3389, which is required for Terminal Services client on computers running Windows 2000 and Remote Desktop Connection on computers running Windows XP or Windows Server 2003. Because the IPSec policies in the example scenario are intended to help secure CORPSRV from almost all untrusted network access, filters for all traffic are used. In either case, filters for a client IPSec policy must match filters for the server IPSec policy.

### **Modifying CORPSRV and CORPCLI Policy Design to Exempt All ICMP Traffic from IPSec Filtering**

Rule 5 of the CORPSRV IPSec policy filters and helps secure all traffic. ICMP traffic is included because Rule 7 of the COPRSRV IPSec policy permits only unsecured, inbound ICMP traffic. You can modify this policy to exempt all ICMP traffic from IPSec filtering. An IPSec policy that

---

exempts all ICMP traffic allows you to run diagnostic tools on clients. For example, you can run the **ping** command on a client to locate a server. You can also run Tracert.exe on a client to determine the network path from the client to the server. Although a policy that exempts all ICMP traffic from IPsec filtering presents security risks, these risks might be considered acceptable in certain network environments.

Windows IPsec does not provide a tool to manually initiate and diagnose errors in a security negotiation. Additionally, some server monitoring applications can use ICMP only to determine if the server is still responding to network requests.

If you assign an IPsec policy to CORPSRV and the policy includes filters that help secure ICMP traffic, then you can run the **ping** command on that server to initiate a security negotiation with another computer (if an SA is not already established with that computer). If you modify the CORPSRV policy design to exempt ICMP traffic from IPsec filtering, make sure that you update diagnostic procedures accordingly. You must modify diagnostic procedures to reflect the fact that you cannot use **ping** to confirm that IPsec SAs can successfully be established from clients to CORPSRV. However, you might be able to use the **net view \\CORPSRV** command or another command that uses the TCP or UDP protocol to verify successful IPsec-secured connectivity between clients and CORPSRV.

The following sections describe how to modify CORPSRV and CORPCLI policies to exempt ICMP traffic. All other unicast IP traffic is secured by IPsec.

---

**Important** Before you modify CORPSRV policy design as described in this section, make sure that you fully understand the risks of doing so. For more information, see "[Security Risks of Receiving Unsecured ICMP Protocol Traffic](#)," later in this paper.

---

#### **Modifying Rule 7 for CORPSRV Policy**

CORPSRV Policy Rule 7 permits unsecured, inbound ICMP traffic. To permit unsecured, outbound ICMP traffic on CORPSRV, modify the filter settings for this rule so that the filter is mirrored. Mirroring creates two filters, one inbound and one outbound, to match the unsecured ICMP traffic that this rule permits. You do not need to change any other CORPSRV policy rules. CORPSRV Policy Rule 5 continues to match all inbound and outbound traffic between **My IP Address** and **Any IP Address** (the IP addresses of CORPSRV and all other computers). The modified Rule 7 uses a more specific filter to match all ICMP traffic for the same source and destination address combination.

To mirror Rule 7, in the properties for this rule, select the **Mirrored** check box, and then confirm that this change is activated in the IPsec policy that is assigned to CORPSRV. To do so, run the **netdiag /test:ipsec /debug** command on CORPSRV.

#### **Creating a New Rule (Rule 4) for CORPCLI Policy 1**

Rule 2 of CORPCLI Policy 1 specifies that all traffic between CORPCLI and CORPSRV be secured by IPsec. To permit unsecured ICMP traffic between CORPCLI and CORPSRV, you must add a new rule to CORPCLI Policy 1 to permit ICMP traffic to the IP address of CORPSRV. Note that you must use CORPCLI Policy 1 to permit unsecured ICMP traffic between CORPCLI and CORPSRV because the default response rule that CORPCLI Policy 2 uses generates a filter to help secure all traffic.

---

To create Rule 4 for CORPCLI Policy 1, select the **Mirrored** check box, and then confirm that this change is activated in the IPSec policy that is assigned to CORPCLI. To do so, you need to confirm the presence of the two permit filters for ICMP by running either the **ipseccmd show all** command (on clients running Windows XP) or the **netdiag /test:ipsec /debug** command (on clients running Windows 2000). For a large domain, it will take time for the update to CORPCLI Policy 1 to propagate to all clients.

The following settings are configured for the fourth rule that is associated with CORPCLI Policy 1:

**IP filter list settings**

- **Name:** Me <-> CORPSRV IP address, ICMP only

**Filter settings**

- **Source address:** My IP Address
- **Destination address:** A specific IP Address (the IP address of CORPSRV)
- **Mirrored:** Yes (This check box is selected.)
- **Protocol:** ICMP
- **Description:** Me <-> CORPSRV, ICMP only

**Filter action settings**

- **Name:** Permit
- **Action:** Permit

**Authentication method, IPSec mode, and connection type**

- **Authentication method:** Kerberos (By default, Kerberos is specified, but it is not used because the filter action is to permit traffic.)
- **Tunnel setting (IPSec mode):** This rule does not specify an IPSec tunnel (This option is selected, so transport mode is used.)
- **Connection type:** All network connections

Note that Rule 2 of CORPCLI Policy 1 (which helps secure all traffic between CORPCLI and CORPSRV) is still in effect. The filters for Rule 4 are more specific than the filter for Rule 2, however, so the Rule 4 filters are applied.

**Confirming ICMP Permit Operation for Updated CORPSRV and CORPCLI Policies**

After you confirm that the updates to CORPCLI Policy 1 have been propagated to appropriate clients and that the updates to the CORPSRV policy have been propagated to CORPSRV, you must confirm that unsecured ICMP traffic is permitted between CORPSRV and CORPCLI. To do so, type the following at the command prompt on CORPCLI:

**ping** CORPSRVIPAddress

**tracert** CORPSRVIPAddress

After you run these commands on CORPCLI, run the same commands on untrusted computers to ensure that these computers can also send ICMP traffic to CORPSRV.

---

Note that the changes to CORPCLI Policy 1 and COPRSRV might affect IPSec policies that are assigned to other computers. The modified CORPSRV Policy Rule 7, which permits inbound and outbound ICMP traffic, uses the same source and destination address combination as Rule 5, a more general rule that requires all traffic (other than ICMP) to be IPSec-secured. For example, the IPSec policy that is assigned to WEBSRV includes a filter to help secure all traffic, including ICMP. If CORPSRV permits unsecured, inbound ICMP traffic, WEBSRV sends ICMP traffic that is secured by IPSec, but WEBSRV receives unsecured ICMP traffic. As a result, WEBSRV drops the unsecured ICMP traffic because the WEBSRV policy requires that all traffic be IPSec-secured. To WEBSRV, it seems that CORPSRV is not responding to a **ping** command.

### **Modifying CORPSRV and WEBSRV Policy Design to Defend against Perimeter Network Server (WEBSRV) Compromise**

When designing an IPSec policy to help secure CORPSRV, you should consider the possibility that an attacker might compromise the perimeter network server, WEBSRV. If WEBSRV is compromised, the attacker might have IPSec-secured access to all ports on CORPSRV through the firewall, unless you take appropriate defensive measures. To defend CORPSRV in this case, you can modify the CORPSRV IPSec policy to restrict as much traffic as possible. Note that this attack is considered a trusted attack, from an IPSec perspective, because the attacker has gained control of a computer that is trusted to have IPSec-secured communications with CORPSRV. In this case, IPSec can provide only limited defense (that is, you can use IPSec filtering to allow IPSec-secured traffic over only specific protocols and ports). Because IPSec can provide only a limited defense against trusted attacks, you should take the following additional defense measures:

- Perform additional procedures to enhance the security of CORPSRV.
- If you do not need to encrypt the traffic that is sent to WEBSRV, change the filter action in the IPSec policy to use AH or ESP without encryption so that a firewall or other NIDS can analyze the IPSec-protected traffic.
- Use HIDS to detect applications that function abnormally because they have been compromised by application-level attacks that cannot be prevented by network-layer defenses.

The following sections describe how to modify CORPSRV and WEBSRV IPSec policy to defend against WEBSRV compromise.

## Modifying CORPSRV Policy Rule 1

Table 5 lists the original CORPSRV Policy Rule 1 and the modified rule. For the modified rule, the filter settings are changed to restrict the traffic that is allowed between CORPSRV and WEBSRV. In this example, WEBSRV might be an Outlook Web Access server, so it can use HTTP to retrieve data from CORPSRV, which might be running Exchange Server. Therefore, CORPSRV Policy Rule 1 is modified to allow only inbound HTTP communication to TCP port 80 between CORPSRV and WEBSRV.

**Table 5 Modifications to CORPSRV Policy Rule 1**

Rule	Filter Settings	Filter Action Settings	Authentication Method
Original Rule 1: Secure communication on path between CORPSRV and WEBSRV	Me <-> Perimeter network WEBSRV, all traffic, mirrored	Negotiate ESP 3DES/SHA1 only, no inbound passthrough, no fall back to clear	Preshared key
Modified Rule 1: Allow inbound HTTP (TCP port 80) traffic only between CORPSRV and WEBSRV	Me <-> Perimeter network WEBSRV, inbound HTTP (TCP port 80) only, mirrored	Negotiate ESP 3DES/SHA1 only, no inbound passthrough, no fall back to clear	Preshared key

The following settings are used for the modified Rule 1 that is associated with the CORPSRV IPsec policy:

### IP filter list settings

- **Name:** Perimeter network WEBSRV <->Me, allow inbound HTTP only

### Filter settings

- **Source address:** A specific IP Address (where the IP address is the IP address of WEBSRV)
- **Destination address:** My IP Address
- **Mirrored:** Yes (This check box is selected.)
- **Protocol:** TCP
- **Source port:** Any
- **Destination port:** 80
- **Description:** WEBSRV <-> Me, TCP src \*, dst 80, mirrored

### Filter action settings

- 
- **Name:** ESP 3DES-SHA1 required
  - Action: **Negotiate security**
  - **Accept unsecured traffic, but always respond using IPSec:** No (This check box is cleared.)
  - **Allow unsecured communication with non-IPSec aware computers:** No (This check box is cleared, therefore fall back to clear is not allowed.)
  - **Perfect forward secrecy (PFS):** No (This check box is cleared.)
  - **Security method: Custom**
    - **Data and address integrity without encryption (AH):** No (This check box is cleared.)
    - **Data integrity and encryption (ESP):** Yes (This check box is selected.)
      - **Integrity algorithm: SHA1**
      - **Encryption algorithm: 3DES**
    - **Session key settings (key lifetimes):** 100,000 KB (100 MB) / 3,600 seconds (one hour)

#### **Authentication method, IPSec mode, and connection type**

- **Authentication method: Preshared key**
- Tunnel setting (IPSec mode): **This rule does not specify an IPSec tunnel** (This option is selected, so transport mode is used.)
- **Connection type: All network connections**

This modified rule enhances the security provided by the original CORPSRV Policy Rule 1 because it allows and helps secure only inbound HTTP traffic to TCP port 80, rather than all traffic. For some applications (for example, an RPC application that cannot limit ports), an All TCP filter might be the most specific filter that you can create. For other applications, you can create a filter that is specific to a single protocol, as is done in this example.

#### **Modifying the WEBSRV Policy Rule**

Whether you use the original rule to negotiate security for all traffic between CORPSRV and WEBSRV or the modified rule to allow only inbound HTTP traffic between the two computers, you must add the same rule or a compatible rule to the IPSec policy that you assign to WEBSRV.

Table 6 lists the original WEBSRV rule and the modified rule. The modified rule is compatible with the modified CORPSRV Policy Rule 1.

---

**Note** This paper does not provide detailed settings for the IPSec policy that is assigned to WEBSRV because such a policy might require many customized rules that are appropriate for a perimeter network environment.

---

**Table 6 Modifications to the WEBSRV Policy Rule**

Rule	Filter Settings	Filter Action Settings	Authentication Method
Original rule: Negotiate traffic between WEBSRV and CORPSRV	Me <->CORPSRV, all traffic, mirrored	Negotiate ESP 3DES/SHA1 only, no inbound passthrough, no fall back to clear	Preshared key
Modified rule: Allow outbound HTTP (TCP port 80) traffic only between WEBSRV and CORPSRV	Me <->CORPSRV, outbound HTTP (TCP port 80) only, mirrored	Negotiate ESP 3DES/SHA1 only, no inbound passthrough, no fall back to clear	Preshared key

Note that the original rule that allows and helps secure all traffic between CORPSRV and WEBSRV is identical in the IPSec policies assigned to both computers. The modified rule, however, is different for the policy that is assigned to each computer. The modified rule for the CORPSRV IPSec policy allows only inbound traffic on TCP port 80, whereas the modified rule for the WEBSRV IPSec policy allows only outbound traffic on TCP port 80.

The following settings are used for the modified rule that is associated with the WEBSRV IPSec policy:

**IP filter list settings**

- **Name:** Me <-> CORPSRV, allow outbound HTTP only

**Filter settings**

- **Source address:** My IP Address
- **Destination address:** A specific IP Address (the IP address of CORPSRV)
- **Mirrored:** Yes (This check box is selected.)
- **Protocol:** TCP
- **Source port:** Any
- **Destination port:** 80
- **Description:** Me <-> CORPSRV, TCP src \*, dst 80, mirrored

**Filter action settings**

- **Name:** ESP 3DES-SHA1 required
- **Action:** Negotiate security
- **Accept unsecured traffic, but always respond using IPSec:** No (This check box is cleared, therefore inbound passthrough is not allowed.)

- 
- **Allow unsecured communication with non-IPSec aware computers:** No (This check box is cleared, therefore fall back to clear is not allowed.)
  - **Perfect forward secrecy (PFS):** No (This check box is cleared.)
  - **Security method: Custom**
    - **Data and address integrity without encryption (AH):** No (This check box is cleared.)
    - **Data integrity and encryption (ESP):** Yes (This check box is selected.)
      - **Integrity algorithm: SHA1**
      - **Encryption algorithm: 3DES**
    - **Session key settings (key lifetimes):** 100,000 KB (100 MB) / 3,600 seconds (one hour)

#### **Authentication method, IPSec mode, and connection type**

- **Authentication method: Preshared key**
- Tunnel setting (IPSec mode): **This rule does not specify an IPSec tunnel** (This option is selected, so transport mode is used.)
- **Connection type: All network connections**

This modified rule enhances the security provided by the original WEBSRV rule because it allows and helps secure only outbound HTTP traffic to TCP port 80, rather than all traffic.

If WEBSRV is compromised, keep in mind that the attacker might try to modify the IPSec policy to increase the amount of traffic that is included in the IPSec SA with CORPSRV. Because CORPSRV allows only TCP port 80 traffic, if WEBSRV proposes All Traffic, then CORPSRV denies the request, because the All Traffic proposal does not match the TCP port 80 filter. As a result, the IKE quick mode negotiation fails and, if auditing for IKE events is enabled (as described in [“Enabling Security Log Audit Events for IKE Negotiation,”](#) later in this paper), a failure audit is generated on CORPSRV.

If you configure a new IPSec policy for WEBSRV with a filter that matches only TCP port 80 traffic, but you do not modify the original All Traffic filter in Rule 1 of the CORPSRV IPSec policy to make the filter more specific, the filters for the two policies will not match. When this occurs, the security negotiation behavior might appear to be inconsistent, depending on which computer initiates the negotiation. This configuration is not recommended because CORPSRV will accept any traffic from WEBSRV. If you use such a configuration, and if WEBSRV initiates the first IKE quick mode security negotiation, then WEBSRV proposes a filter that is more specific than the filter that CORPSRV expects. In this case, CORPSRV accepts the proposal, and an SA pair is established from WEBSRV to CORPSRV for TCP 80 traffic.

CORPSRV creates the following two dynamic filters to manage the IPSec SAs. One filter is for inbound traffic. The other filter is the corresponding mirrored filter for outbound traffic.

---

#### **Inbound port-specific dynamic filter:**

- **Source address: A specific IP address** (the IP address of WEBSRV)
- **Destination address: My IP address** (This makes it an inbound filter.)
- **Protocol: TCP**
- **Source port: Any**
- **Destination port: 80**

#### **Outbound port-specific dynamic filter:**

- **Source address: My IP address** (This makes it an outbound filter.)
- **Destination address: A specific IP address** (the IP address of WEBSRV)
- **Protocol: TCP**
- **Source port: 80**
- **Destination port: Any**

As with default response filters, these dynamic filters are deleted in approximately two hours. IKE uses the dynamic filter to retain the port-specific settings for the SA (from WEBSRV to CORPSRV, protocol **TCP**, source port **Any**, destination port **80**) so that the IPsec SA pair can be refreshed with a quick mode negotiation (rekey), regardless of which computer initiates the quick mode request. If CORPSRV initiates the first IKE quick mode security negotiation, then CORPSRV proposes an All Traffic filter to WEBSRV. In this case, the security negotiation fails because the All Traffic filter is a more general filter than WEBSRV can accept. This does not mean that CORPSRV and WEBSRV cannot communicate at all. However, communication can succeed only when WEBSRV initiates the security negotiation and HTTP requests and responses (TCP destination port 80) from WEBSRV to CORPSRV are allowed.

The most secure IPsec filter configuration is one that allows traffic over only required ports and protocols and restricts the traffic to the specific direction that is required.

### **Additional IPsec Policy Modifications**

The example IPsec policy rules for CORPSRV, CORPCLI, and WEBSRV that are provided in this paper reflect best practices that were developed during the network penetration testing conducted by Foundstone and the production use of IPsec on internal Microsoft networks. The successful deployment of this policy was in large part due to initial small-scale lab testing and initial pilot testing conducted by the IPsec administrative team on less utilized servers. Before you deploy IPsec in a production environment, make sure that you conduct similar testing and perform additional security and performance assessments as required for your environment.

The following sections describe policy modifications that were implemented in the Microsoft network. You might want to make similar modifications in your own production environment.

#### **Customized Rules for IPsec Policies Assigned to Microsoft Servers and Clients**

When the example policies were placed into production use at Microsoft, the following additional rules were required to customize these policies for the Microsoft environment:

- 
- A rule to control the use of IPSec in subnets used by remote access clients. This rule allowed IPSec to be disabled on remote access clients when troubleshooting was required. This rule is similar to Rule 5 of the CORPSRV IPSec policy, but, rather than specifying **Any IP Address**, this rule specified the subnets that remote access clients used (these clients typically used IPSec when communicating with internal corporate network servers).
  - A rule to control the use of IPSec on each of the subnets in the internal corporate network, after several production servers were connected to a separate multihomed administrative network.
  - A rule to exempt traffic that was required by a computer that ran server-monitoring software.
  - A rule to exempt traffic that was required for a network backup system to run on some servers, to accommodate systems that did not support IPSec. For example, one backup system was a non-Microsoft device that did not support IPSec. In another case, the backup system was in the same protected room as the server using Gigabit Ethernet, and an IPSec hardware offload network adapter was not available.
  - Several rules to allow clients running the Macintosh operating system to access servers. The filters for these rules allowed traffic from either the IP addresses of specific clients or from all clients on a specific subnet.

You might want to add similar rules to permit other traffic types that are required in your network environment. When you create such rules, however, make sure to evaluate whether they allow your servers to operate safely within the security requirements of your organization.

### **Fall Back to Clear and Inbound Passthrough Configuration for Initial Deployment**

During the initial deployment of IPSec, the **Allow unsecured communication with non-IPSec-aware computers** (fall back to clear) and **Allow unsecured communication, but always respond using IPSec** (inbound passthrough) check boxes were both selected on the IPSec policy assigned to the servers, so that traffic could be sent unsecured to the servers. This configuration allowed server administrators to identify and troubleshoot authentication problems, while providing an option to restore connectivity to the servers. To restore connectivity, the administrators stopped the IPSec service on the client, not on the server.

To determine which domain members had not received Active Directory-based IPSec policy, which clients were not domain members, and on which clients the IPSec service was stopped for other reasons, administrators used the IP Security Monitor snap-in that is available in Microsoft Management Console (MMC) to identify clients that had established soft SAs (non-IPSec-secured communication) with servers.

---

## Fall Back to Clear and Inbound Passthrough Configuration for Final Deployment

On some servers, the IPSec policy used for final deployment was configured so that the **Accept unsecured communication, but always respond using IPSec** check box was cleared. Accordingly, those servers would not accept unsecured traffic. However, the **Allow unsecured communication with non-IPSec aware computers** check box was left selected so that these servers could initiate connections to computers that were not configured to use IPSec (for example, Internet proxy servers).

---

**Note** For this behavior to be implemented, computers must be running Windows 2000 Service Pack 3 or later, Windows XP Service Pack 1, or Windows Server 2003. On computers running Windows 2000 or Windows XP without an appropriate service pack installed, when you clear the **Accept unsecured communication, but always respond using IPSec** check box and select the **Allow unsecured communication with non-IPSec-aware computers** check box, then unsecured traffic is still accepted. This behavior occurs because, when the **Allow unsecured communication with non-IPSec-aware computers** check box is selected, IPSec processes the associated inbound filter as an inbound passthrough filter (the same behavior that occurs when the **Accept unsecured communication, but always respond using IPSec** check box is selected).

On computers running Windows 2000 or Windows XP with an appropriate service pack installed, or on computers running Windows Server 2003, when you select the **Allow unsecured communication with non-IPSec-aware computers** check box, IPSec does not process the associated inbound filter as an inbound passthrough filter. Therefore, these two settings result in independent actions, and they behave in a consistent way on computers running Windows 2000, Windows XP with an appropriate service pack installed, and Windows Server 2003.

---

To communicate with the servers, clients in the Microsoft corporate network were required to have an IPSec policy assigned and to have successfully authenticated and negotiated security with the server. On these clients, the IPSec policy used for final deployment was configured so that the **Allow unsecured communication with non-IPSec aware computers** check box was selected and associated with the filter for the servers. This policy allowed clients to request to negotiate security with IPSec-secured servers but to fall back to clear if the servers did not respond to the clients' requests. As a result, clients could establish connectivity with servers even when IPSec was disabled on those servers.

Foundstone performed penetration testing for servers that were configured to accept unsecured inbound traffic but to not fall back to clear and on servers that were configured to not accept unsecured inbound and to not fall back to clear. In both cases, Foundstone was unable to compromise the security that IPSec provided. Keep in mind, though, that if you select the **Accept unsecured communication, but always respond using IPSec** check box on a server, then IPSec does not secure inbound traffic. When this configuration is used, an unauthenticated attacker might be able to mount successful inbound denial-of-service and buffer overflow attacks against ports that are open on the server and against other parts of the TCP/IP stack. Likewise, if you select the **Allow unsecured communication with non-IPSec-aware computers** check box on a server, an attacker might be able to mount an attack from the computer that is not secured by IPSec when an active, soft SA between that computer and the server is established.

---

## Managing IPSec Policies: Operational Considerations

IPSec provides greater security at the expense of network and processing performance and compatibility with other services, tools, and features. The following sections describe these and other important operational considerations for managing IPSec policies.

### Understanding the Impact of IPSec on CPU Performance and Network Utilization

IPSec does incur performance overhead to establish and maintain secure connections. For example, IPSec adds overhead to IP packets, which increases packet-processing requirements and incurs network latency. Depending on the load placed on the computer CPU, it might be necessary to increase CPU memory or to install IPSec offload adapters to compensate for the increased overhead of IPSec. The effects of IPSec on CPU performance are typically negligible on clients because clients often have ample, spare CPU processing power and memory to accommodate IPSec. However, servers that have highly loaded CPUs might incur a significant increase in load, depending on the amount of traffic that must be secured by IPSec, the number of SAs that must be processed, and the cryptographic algorithms that are selected.

### Understanding Specific Factors that Contribute to IPSec Performance Overhead

The following are specific factors that contribute to IPSec performance overhead. Understanding these factors can help you plan more effectively for IPSec deployment. You can minimize the performance impact of IPSec on a server by understanding how specific IPSec policy settings affect performance and by configuring settings accordingly.

- **IKE SA establishment results in slower connection establishment.** When an SA is initially established between two computers, a latency of about one to three seconds occurs while security is negotiated. This time might vary, depending on policy design, the authentication method that is selected, network roundtrip time, and the load placed on the servers required to establish the connection. Accordingly, end-users might notice brief, initial delays when IPSec-secured connections are established. Typically, after SAs are established, no further delays occur.
- **SA maintenance affects scalability and results in increased CPU and memory overhead.** Performance degradation and lower throughput might be experienced by servers that must establish IPSec-secured communications with many clients or send IPSec ESP-encrypted traffic at high data rates without an IPSec offload network adapter. Typically, for each active IPSec client, 5 to 20 KBs of memory is used on a server. The IKE authentication method has the greatest impact on memory usage for each active IPSec client. For example, Kerberos authentication typically requires less memory per SA than certificate authentication. The size of a Kerberos ticket might vary, however, depending on Active Directory and security group configurations. Certificate and certificate chain sizes vary with PKI designs.
- **Per-packet processing overhead results in increased CPU and memory overhead.** Each IPSec-secured packet creates additional performance overhead for encryption, authentication, and other packet processing functions. Each packet that is not secured by IPSec must still pass through all IPSec filters. Therefore, when you configure an IPSec policy, make sure that

---

you create the minimum number of filters required for your environment. The impact of IPSec on throughput varies in proportion to the rate at which traffic is sent or received. Typically, active IPSec policies with 500 or fewer filters result in only a minor impact on throughput. In Windows XP IPSec and Windows Server 2003 IPSec, the per-packet processing performance is significantly improved, compared to the per-packet processing performance in Windows 2000 IPSec.

- **Per-packet volume overhead results in reduced throughput and increased network utilization.** For each IPSec-secured packet, the AH and ESP headers add an additional 24 to 36 bytes of data, depending on which method of encapsulation is used. This decreases the effective MTU for TCP and other upper-layer protocols. As a result, for the same amount of application data, traffic that is secured by IPSec uses more packets than traffic that is not secured by IPSec. However, TCP and IPSec are integrated into the TCP/IP stack so that packet fragmentation is avoided whenever possible.

To properly evaluate the effects of IPSec on CPU performance, establish performance benchmarks and goals for each computer. For each server, complete an inventory to record its hardware configuration, Ethernet interface capability, number of unique clients, average and peak CPU and memory utilization, and average and peak traffic flow characteristics. Keep in mind that the following factors determine the practical scalability limit for each server in your environment: traffic flow characteristics, whether you use IPSec network offload adapters, IPSec policy settings, and the load placed on servers by other services and applications. For example, servers that experience a high traffic volume might support only a few clients, if you require IPSec ESP encryption, and you do not use IPSec network offload adapters. The deployment of IPSec in the Microsoft internal corporate network has shown that in production, low-volume servers can support up to 20,000 active IPSec clients that use AH. Keep in mind that this number can vary across different environments, depending on factors such as the hardware configuration of the server.

In Windows Server 2003 IPSec, filtering and memory utilization is significantly more efficient than in Windows XP IPSec and Windows 2000 IPSec. However, none of the Windows IPSec implementations support IP packet compression in combination with IPSec processing.

### **Using IPSec Hardware Offload Network Adapters**

Network offload adapters can accelerate IPSec processing by performing hardware offload of IPSec cryptographic functions. By default, IPSec network offload adapters typically can process only a certain number of IPSec SAs in hardware simultaneously. If the number of IPSec SAs exceeds the processing limit of the network offload adapter, the IPSec driver processes the traffic for the excess SAs in software. This might result in sudden increases in CPU load for IPSec-secured traffic. If this problem occurs on a server, you might need to add several IPSec offload adapters to the same server to ensure that all IPSec-secured traffic can be processed in hardware. If you do this, keep in mind that the server will either have several DNS names or use DNS round-robin to alternate client connections to the IP addresses of different adapters. Alternatively, you might be able to change the default limit for the number of IPSec SAs that can be processed in hardware by modifying driver settings or registry key values. To determine whether your IPSec network offload adapter supports this change, see the manufacturer's documentation.

---

IPSec network offload adapters typically 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.

For information about the 10/100 MB Ethernet hardware offload network adapters that are available for Windows 2000 IPSec, see the following:

- The [Intel Web site](http://go.microsoft.com/fwlink/?LinkId=16474), at <http://go.microsoft.com/fwlink/?LinkId=16474>.

You can use the Pro 100 S series of hardware offload adapters for Windows 2000 IPSec hardware offload.

- The [3Com Web site](http://go.microsoft.com/fwlink/?LinkId=16475), at <http://go.microsoft.com/fwlink/?LinkId=16475>.

You can use 3Com network adapters with 3XP processors for Windows 2000 IPSec hardware offload.

For information about additional network adapters that are compatible with Windows 2000, see [Search for Compatible Hardware Devices](http://go.microsoft.com/fwlink/?LinkId=3787), at <http://go.microsoft.com/fwlink/?LinkId=3787>. For information about network adapters that are compatible with Windows XP, see [Browse Hardware, Networking and Modems, and LAN Cards](http://go.microsoft.com/fwlink/?LinkId=19934) at <http://go.microsoft.com/fwlink/?LinkId=19934>. For information about network adapters that are compatible with Windows Server 2003, see [Networking and Modems, LAN Cards](http://go.microsoft.com/fwlink/?LinkId=19935), at <http://go.microsoft.com/fwlink/?LinkId=19935>.

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, see [Task Offload](http://go.microsoft.com/fwlink/?LinkId=18763) at <http://go.microsoft.com/fwlink/?LinkId=18763>.

## Understanding the Impact of IPSec on Users and Applications

Administrators typically configure IPSec policies to use IPSec transport mode at the IP layer (layer 3), and such policies are, for the most part, transparent to applications. Accordingly, you can use IPSec to enhance the security of existing application TCP/IP traffic against network-based attacks from untrusted computers without modifying applications to provide stronger security. However, there are currently no APIs in Windows 2000, Windows XP, or Windows Server 2003 that enable an application to directly use IPSec transport mode to help secure its traffic. To use IPSec to help secure remote access to a network, you must configure an IPSec policy or use specific communication tools, such as L2TP/IPSec VPN tunnels.

Do not use IPSec policies as a replacement for application-level security configuration. Application-level security configuration controls user and data authentication and data confidentiality and enables proper authorization. These functions are critical to help defend against attacks within normal application communications, even if the traffic is IPSec-secured. IPSec transport mode provides a foundation layer to help secure TCP/IP traffic between computer IP addresses, thereby providing a defense against untrusted attacks. For example, although you can use IPSec to help secure SMTP communication, IPSec does not provide authentication and confidentiality for e-mail messages during transit. Likewise, although you can use IPSec to help secure HTTP traffic, IPSec does not provide a security model to control user access to certain content on a Web server. To use IPSec to help secure HTTP traffic when the

---

HTTP traffic passes through HTTP proxy servers, you need to perform several administrative steps.

Finally, IPSec might not be completely transparent to applications and end-users. The impact of IPSec on applications results from the behavior of IPSec itself and the effects of specific IPSec policy configurations. For example, IPSec static traffic filtering cannot provide the level of filtering flexibility that certain applications might require. To design an IPSec policy that helps secure only one application, you must have a detailed understanding of the way in which the application uses network traffic. For this reason, the example policies provided in this paper describe how to use IPSec to help secure almost all TCP/IP traffic that is sent to and from a server.

## **Administrative Experience and Security Group Membership for Managing IPSec Policy**

To properly deploy IPSec, you should have a detailed knowledge of IP networking, including a thorough understanding of how the Windows operating system and applications use the network. You should also be comfortable with analyzing network traffic traces. For example, you should be comfortable with using Network Monitor or similar network analysis tools. For information about Network Monitor, see "[Viewing IPSec and Other Network Communication](#)," later in this paper.

To manage Windows IPSec policy, you must be a member of the local Administrators group or the Domain Admins group on the computers that you want to manage. Because IPSec policy requires proper configuration on both sides of the communication to help provide security, you must configure IPSec policy on both clients and servers. Doing so might require you to cross organizational ownership boundaries (that is, if different administrators in your organization are responsible for client configuration and server configuration). You must have administrative credentials on the computers that receive the IPSec policy because the tools for monitoring and troubleshooting IPSec require these credentials.

In small organizations, the person most knowledgeable in networking should create and assign IPSec policy. In larger organizations, firewall administrators, DHCP administrators, router administrators, and network security professionals generally have the requisite background knowledge. Also, in larger organizations, the person responsible for configuring IPSec policy does not need to be a member of the Domain Admins group. However, that person should have local administrative credentials on servers or clients to enforce Active Directory-based IPSec policy. Such individuals might be corporate network administrators or IT security administrators, and they should have specialized knowledge in IPSec. Domain administrators must grant these individuals explicit permissions to create or modify Active Directory-based IPSec policy.

If you deploy IPSec in a larger organization, consider creating an IPSec Administrators group for the domain. Doing so makes it easier for local server administrators to add the appropriate permissions once or to include this group in the local Administrators group when a computer joins the domain.

## **Managing Active Directory-Based and Local IPSec Policy**

IPSec policies can be applied to local computers, domains, and sites, or they can be applied to OUs in Active Directory. The following sections provide considerations for administering Active Directory-based and local IPSec policies.

---

## Storing Active Directory-Based and Local IPSec Policy

After you create an IPSec policy, you must assign it for the policy to take effect. You can use a Group Policy configuration of Active Directory to assign IPSec policies to the Group Policy object (GPO) of a site, domain, or OU. In addition, each computer has one local GPO, which is also known as the local computer policy.

If you assign a local IPSec policy, the policy is stored in the local system registry. The local system 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 an Active Directory-based IPSec policy, a copy of the current policy is maintained in a cache in the local registry, and the policy is stored in Active Directory. For the example used in this paper, the policy for the domain controller CORPDC.CORPFOREST.COM is stored in the following Active Directory location:

**LDAP://CORPDC.CORPFOREST.COM/CN=IP Security,CN=System,DC=CORPFOREST,DC=COM**

If the computer to which an Active Directory-based policy is assigned cannot connect to the domain, the cached copy of the policy is applied.

Note that IPSec policy configurations are stored in a semi-readable form. Any names, comments, and descriptions associated with an IPSec policy are stored in plaintext, while the actual configurations are stored in non-ASCII format. Configurations include the source and destination IP addresses, source and destination ports, authentication method, and protocol filters.

By default, in Windows 2000, all domain computers and all authenticated users can read Active Directory-based IPSec policy. However, only members of the Domain Admins group can create or modify Active Directory-based IPSec policy.

---

**Important** Active Directory-based IPSec policies are stored in the IP Security Policies container in the System container. You can limit access to the IP Security Policies container by denying Read access to this container to members of the Domain Users group and granting Read access to members of the Domain Computers group and other administrative users. However, keep in mind that local administrators will have Read access to an assigned IPSec policy after it is cached in the local registry (in the **HKEY\_LOCAL\_MACHINE\Software\Policies\Microsoft\Windows\IPSec\** registry key and its subkeys). Accordingly, there is no effective way to provide highly restricted Read access to an Active Directory-based IPSec policy.

---

Depending on the security requirements of your organization, you might make changes to limit Read access to the registry location of the local IPSec policy. By default, users who are not members of the local Administrators group might have Read access to this location. For more information, see article 329194, "IPSec Policy Permissions in Windows 2000 and Windows Server 2003," in the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

For more information about assigning IPSec policy, see "[Assigning and Distributing IPSec Policy](#)," later in this paper.

## Using IP Security Policy Management to Create and Modify IPSec Policy

Although you can script IPSec policy creation, the primary method of creating and managing IPSec policy is by using the IP Security Policy Management snap-in that is available in MMC. You can use IP Security Policy Management to create, modify, and store local IPSec policies or Active

---

Directory-based IPsec policies. Additionally, you can use IP Security Policy Management to modify IPsec policy on remote computers.

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

When multiple IPsec policies are deployed in an organization, it might be difficult to quickly determine which IPsec policy is assigned to a specific computer. Within Group Policy Object Editor, you can use IP Security Policy Management to troubleshoot policy precedence issues and determine the exact set of policies that clients and servers are using.

You can 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 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 (Secpol.msc).
- 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 running with local system privileges.

To simplify IPsec policy management, you might want to dedicate one computer to configure local IPsec policy. You can then use the IP Security Policy Management **Export Policies** and **Import Policies** menu commands to back up and restore IPsec policy. For more information, see [“Considerations for Backing Up and Restoring IPsec Policy,”](#) later in this paper.

After you create an IPsec policy, use a version control system to track changes to the policy during the development, testing, and deployment phases.

### Using Command-Line Tools to Script IPsec Policy Creation

Windows 2000, Windows XP, and Windows Server 2003 IPsec each provide different command-line tools for scripting IPsec policy creation. You can use these tools to create, modify, and assign IPsec policies without immediately affecting the configuration of the active IPsec policy, or you can script IPsec dynamic policy to temporarily enhance the design of an assigned Active Directory-based or local IPsec policy.

The settings of a dynamic (temporary) IPsec policy are merged with the settings of the assigned Active Directory-based or local IPsec policy. These settings are considered temporary because they take effect immediately (when the IPsec service is running) and are not stored. If the IPsec

---

service is stopped and restarted, these settings are lost. Creating a dynamic IPsec policy is useful if you want to selectively enforce security rules that are required only for specific computers or network devices. For example, you can use a dynamic IPsec policy to create filters that permit unsecured traffic to a network backup device or a local IPsec policy management computer (the dedicated computer on which you manage IPsec policy). One drawback to this approach is that IPsec dynamic policy settings might override Active Directory-based policy, thus altering the behavior of the merged Active Directory and dynamic IPsec policy settings.

To script the creation of local or Active Directory-based IPsec policy on computers running Windows 2000, you can use Ipsecpol.exe, a command-line tool that is provided with the Windows 2000 Server Resource Kit. Ipsecpol.exe is not a full-featured command-line or scripting tool (for example, you cannot use Ipsecpol.exe to delete or rename filter lists or filter actions), nor is it supported under any Microsoft standard support program or service. You can use Ipsecpol.exe only to create an IPsec policy. To download the latest version of Ipsecpol.exe, see [Ipsecpol.exe: Internet Protocol Security Policies Tool](http://go.microsoft.com/fwlink/?LinkId=16466), at <http://go.microsoft.com/fwlink/?LinkId=16466>. For an example of how to use Ipsecpol.exe, see article 813878, "How to Block Specific Network Protocols and Ports by Using IPsec," in the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

To script the creation of local or Active Directory-based IPsec policy on computers running Windows XP, you can use Ipseccmd.exe, a Windows Support Tool that is included in the Support Tools folder of the Windows XP operating system disc. You can also use Ipseccmd.exe to monitor IPsec. For more information about Ipseccmd.exe, see "[Managing and Monitoring IPsec on Computers Running Windows XP](#)," later in this paper.

To script the creation of IPsec policy on computers running Windows Server 2003, you can use the Netsh commands for IPsec. For more information, see [Netsh commands for Internet Protocol security \(IPsec\)](http://go.microsoft.com/fwlink/?LinkId=19315), at <http://go.microsoft.com/fwlink/?LinkId=19315>.

---

**Note** You can use Ipsecpol.exe only on computers running Windows 2000, Ipseccmd.exe only on computers running Windows XP, and the Netsh commands for IPsec only on computers running Windows Server 2003. However, IPsec policies that are created with each of these tools are compatible across Windows 2000, Windows XP, and Windows Server 2003. For more information, see "[IPsec Policy Compatibility Considerations](#)," later in this paper.

---

## Assigning and Distributing IPsec Policy

To assign a local IPsec policy for a local computer or a remote computer, you can use IP Security Policy Management. To assign an Active Directory-based IPsec policy, however, you must use Group Policy Object Editor to assign the IPsec policy to the OU in Active Directory that contains the appropriate computer. Active Directory-based policy is accessed by using LDAP, so the IPsec policy configuration data can be authenticated and encrypted.

If a computer is not a member of a Windows 2000 domain or a Windows Server 2003 domain, it cannot retrieve IPsec policy from Active Directory. In such cases, you can distribute IPsec policy in two ways that do not use Active Directory:

- As a self-installing compressed executable file that contains a command script and the IPsec command-line tools (Ipsecpol.exe, Ipseccmd.exe, or Netsh commands for IPsec)

- 
- As an .ipsec file that you can import from another computer by using IP Security Policy Management

Neither of these distribution methods provides data origin authentication or encryption of the IPsec policy configuration data. Therefore, if you use either of these methods, make sure that you use strict version and change control processes to ensure that the policy file cannot be altered after it is created.

To understand how to apply IPsec policy in an Active Directory environment, it is important to understand IPsec policy precedence. When assigning an Active Directory-based IPsec policy, keep the following in mind:

- Like Group Policy settings, the assignment precedence for IPsec policies, from lowest to highest, is: local, site, domain, and OU. Active Directory-based IPsec policies override the settings of any local IPsec policy that is assigned on the computer. If no Active Directory-based IPsec policies are assigned, the local IPsec policy assignment is used. For example, if IPsec policies are assigned to the domain, and IPsec policies are assigned to an OU within the domain, then the computers that are members of the OU within the domain use the IPsec policies that are for the OU. Domain computers that are not members of the OU use the IPsec policies that are for the domain.
- Active Directory-based IPsec policies are not merged with local IPsec policies. Additionally, IPsec policies from different OUs are not merged.
- The settings of dynamic IPsec policies and persistent IPsec policies (which you create by using IPsec command-line tools) are merged with the settings of the active IPsec policy.

---

**Important** Keep in mind that members of the local Administrators group can view and modify the IPsec policy settings on their computers, and, therefore, it is important to restrict the use of administrative credentials in your organization. For example, even when an Active Directory-based IPsec policy is assigned to a computer, members of the Administrators group on that computer can use Ipsecpol.exe, Ipseccmd.exe, or the Netsh commands for IPsec to immediately affect the configuration of the active IPsec policy. In this way, local administrators might change the level of security that the Active Directory-based IPsec policy provides or completely defeat that security. This is not an inherent security vulnerability of IPsec because members of the local Administrators group can disable the IPsec service, have full control over the operating system, and, most likely, have physical control over the computer. Application users who are logged on with local administrative credentials also make their computers vulnerable to Trojans and viruses that might modify IPsec policy settings (if Ipsecpol.exe or Ipseccmd.exe are installed on the computer, or if the Trojans and viruses reverse-engineer the unpublished policy storage formats of the IPsec policy).

---

### Delegating Permissions to Modify Active Directory-Based IPsec Policy

When you use Active Directory-based IPsec policy, you can delegate permissions on the IP Security Policies container to specify those who are allowed to modify IPsec policy. The IPsec policies are a collection of related directory objects, some of which can be shared between policies. Therefore, an IPsec policy administrator typically requires Write access to all IPsec policy objects, and you should not assign specific permissions for specific IPsec policies. If many administrators in your organization want to manage Active Directory-based IPsec policy, it is recommended that these individuals dedicate one computer to configure local IPsec policy. They can then use IP Security Policy Management to export an IPsec policy file for the domain administrator or another delegated administrator. The domain administrator (or other delegated administrator) can then use IP Security Policy Management to import the IPsec policy file into the

---

IP Security Policies container in Active Directory. Doing so minimizes the number of administrators who can modify the IP Security Policies container in the domain.

To delegate permissions on the IP Security Policies container, you must use an Active Directory editing tool, such as ADSI Edit. ADSI Edit is a Windows Support Tool that uses the Active Directory Service Interfaces (ADSI) and that is provided on the Windows 2000 and Windows Server 2003 operating system disc.

### **Preventing Unwanted Active Directory-Based IPsec Policy from Being Assigned to Computers**

If you assign a local IPsec policy to a server before Active Directory-based IPsec policy is assigned, when the domain administrator assigns an Active Directory-based client IPsec policy to all domain members, the Active Directory-based IPsec policy propagates to the server, overrides the local IPsec policy, and might cause IPsec-secured communications to fail. Even if you group the server computer accounts into a separate OU, without additional measures, the Active Directory-based IPsec policy at the domain level applies to all of the servers, as well as to all of the clients. Keep in mind that only one IPsec policy can be active on a computer, even if multiple policies are assigned at different OU levels in the domain. To prevent an unwanted Active Directory-based IPsec policy from being assigned to a server, do one of the following:

- Create (or modify) a GPO at the domain level that is assigned to the servers' OU, assign the Active Directory-based client IPsec policy for the domain to the GPO, and then set permissions on the GPO to deny the server computer accounts Read access to the GPO. When you do this, the server does not detect the GPO, and, therefore, the server is not assigned the Active Directory-based client policy. For example, you can deny a group of internal file servers Read access to the GPO. Doing this allows server administrators to manage IPsec policy settings locally on each internal file server. This approach is recommended when you want to exempt servers from being assigned a specific Active Directory-based IPsec policy.
- Have a domain administrator group the servers into their own OU, create a GPO that allows an IPsec policy to be assigned to that OU, and then have local server administrators create an IPsec policy to import to the OU. Note that this approach does not involve local IPsec policy assignment. Rather, it allows server administrators to create and manage IPsec policies for servers locally and then to apply the policy from Active Directory. After the domain administrator creates the GPO for the servers' OU, he or she gives the server administrator permissions to modify only that GPO. The server administrator must provide the domain administrator with an appropriate set of IPsec policies to import. The server administrator can change the IPsec policy that is assigned to the GPO and other settings in the GPO, but the server administrator might not be given permissions to modify Active Directory-based IPsec policies. This approach is recommended when you want to delegate the details of designing IPsec policies to a server administrator, but you also want to use Active Directory to provide a consistent, centralized policy way to assign IPsec policy.
- Group the servers into their own OU and the clients into their own OU, create a separate GPO for each OU, and then assign a separate IPsec policy for each GPO. This approach might not be feasible for large organizations that have many servers and clients.

- 
- Assign an Active Directory-based client policy for the domain to the domain-level GPO (default domain policy), move server accounts into their own OU, and then create a new GPO for that OU. Do not assign any IPsec policies to this GPO. Configure the OU that is assigned to the servers' OU to block policy inheritance, but keep in mind that this configuration prevents the OU from inheriting the settings of all other policies assigned at higher levels of the Active Directory hierarchy, not just IPsec policies. Accordingly, the disadvantage of this approach is that it requires you to copy the settings of all other higher-level policies to the GPOs that are assigned to the server OU, if you want those settings to be applied. For example, if you assign a certificate auto-enrollment policy at the domain level, members of the server OU do not receive the GPO that delivers the certificate auto-enrollment policy, so you must copy those settings to the GPO that is assigned to the server OU for those settings to be applied. This approach is recommended if you want to apply custom local IPsec policies to individual servers in the OU.
  - If you have servers that should not be assigned any IPsec policy, then either configure these server accounts to be unable to read GPOs that assign IPsec policy from the domain, or stop and disable the IPsec service on these servers.

### **Customizing Active Directory-Based IPsec Policy for a Specific Server**

Although an Active Directory-based IPsec policy might be suitable to help secure most communications for a group of servers, you might need to customize an IPsec policy for a specific server. You can do this by using a Windows 2000, Windows XP, or Windows Server 2003 IPsec command-line tool to create a dynamic IPsec policy. For more information, see [“Using Command-Line Tools to Script IPsec Policy Creation,”](#) earlier in this paper.

To display filter details about the IPsec policy that is assigned to a computer, you can use any of the following command-line tools: Netdiag.exe (for computers running Windows 2000), Ipseccmd.exe (for computers running Windows XP), or Netsh commands for IPsec (for computers running Windows Server 2003). Or, for computers running Windows XP or Windows Server 2003, you can use IP Security Monitor. For information about Netdiag.exe, see [“Viewing Network and IPsec-Related Information,”](#) later in this paper. For more information about IP Security Monitor, see [“Monitoring IPsec Status Information,”](#) later in this paper.

### **Providing Backup Security During the Loss of Active Directory-Based IPsec Policy**

An Active Directory-based IPsec policy that is assigned to a server might be inadvertently unassigned or corrupted. As a safeguard against this, create and assign a local IPsec policy for the server. When you do this, if the Active Directory-based IPsec policy is unassigned or otherwise lost, the server remains IPsec-secured, and communication can be maintained if IPsec-secured communication is required.

The local policy that you create for this purpose can help secure the server so that all traffic to the server is blocked until an appropriate Active Directory-based policy can be reassigned, or the policy can allow IPsec-secured communications between the server and trusted computers. If you create and assign a local policy that allows the server to establish IPsec-secured communications with trusted computers, keep in mind that you must keep this policy up to date, so that the policy remains compatible with the Active Directory-based IPsec policy that is

---

assigned to the clients. For more information, see [“Considerations for Backing Up and Restoring IPsec Policy,”](#) later in this paper.

### **IPSec Policy Compatibility Considerations**

As mentioned earlier, IPSec policies are compatible across Windows 2000, Windows XP, and Windows Server 2003. For example, you can export the IPSec policies that you create by using Ipsecpol.exe from computers running Windows 2000 and then import these policies into Windows Server 2003 Active Directory or local IPSec policy stores on computers running Windows XP or Windows Server 2003. Additionally, you can assign Windows 2000 Active Directory-based policy to computers running Windows 2000, Windows XP, or Windows Server 2003. Any IPSec policies that you create on computers running Windows XP and Windows Server 2003 can be stored in Windows 2000 Active Directory or exported and then imported for use on computers running Windows 2000, if the policies use the basic IPSec features supported by Windows 2000.

---

**Important** If you plan to apply IPSec policies that use any of the new features that are available only in the Windows Server 2003 implementation of IPSec, do not assign these policies to computers running Windows 2000 or Windows XP. Also, do not use IP Security Policy Management, on computers running Windows 2000 or Windows XP, to manage policies that use features that are new for Windows Server 2003. If you use Windows 2000 or Windows XP to manage Windows Server 2003 policies, any new Windows Server 2003 features are lost.

---

If you use command-line tools to create or modify IPSec policy or to view IPSec information, note that the IPSec internal infrastructure components were significantly modified in Windows XP and again in Windows Server 2003, such that Ipsecpol.exe does not work on computers running Windows XP, Ipseccmd.exe does not work on computers running Windows 2000, and neither Ipsecpol.exe nor Ipseccmd.exe work on computers running Windows Server 2003. In Windows Server 2003, the Netsh commands for IPSec provides a full featured and Microsoft-supported interface. It is recommended that you use a computer running Windows Server 2003 as an IPSec policy management computer, due to the enhanced scripting support provided by the Netsh IPSec command-line tool and the availability of IP Security Monitor, which allows you to view filter details and other information about the assigned IPSec policy. You can then import the IPSec policy from the IPSec policy management computer into Active Directory and use Group Policy to assign the IPSec policy to an OU or domain. Or, you can use IP Security Policy Management to import the IPSec policy into local IPSec policy stores.

---

**Note** Windows 2000, Windows XP, and Windows Server 2003 do not have published APIs in the Platform Software Development Kit (SDK) for IPSec policy management.

---

### **Remotely Managing IPSec Policy**

After you create, assign, and test an IPSec policy, you can distribute it as a command-line script that is included as a startup script for a remote server. You can also remotely manage IPSec policy by using IP Security Policy Management to directly connect to a remote computer over authenticated RPC. Whether you use a command-line script or Active Directory-based IPSec policy for remote management, it is important to keep the following limitations in mind:

- You can remotely update the IPSec policy for a server by using a command-line tool only if the policy is stored and assigned as a local IPSec policy.
- When you create and assign an Active Directory-based IPSec policy, you must modify the

---

policy within Active Directory.

- You must use a command-line tool to create a dynamic IPsec policy and to view or change the settings of such a policy. You cannot use IP Security Policy Management to view or change the settings of a dynamic IPsec policy.

---

**Note** IP Security Policy Management requires user-authenticated and encrypted RPC connections when it is used to directly connect to a remote computer. Only users who are members of the local Administrators group on a computer can perform remote management. You cannot specify explicit credentials when you use IP Security Policy Management to remotely manage and monitor of IPsec policy.

---

## Considerations for Backing Up and Restoring IPsec Policy

Building a good IPsec policy can be complicated and time consuming. Therefore, it is important to back up IPsec policies in the event that you need to restore them. If you assign an Active Directory-based IPsec policy to a computer, then you do not need to back up the policy configuration locally on that computer. If you restore a computer or add a new computer to the domain and place it in the appropriate OU, the Active Directory-based policy is applied to the computer.

If you assign a local IPsec policy, and if you use backup and restore tools that preserve the **HKEY\_LOCAL\_MACHINE\Software** registry key path, then the IPsec policy local store and the cached copy of the Active Directory-based IPsec policy registry keys are included in the backup and restoration.

On computers for which IPsec policy has been restored from a backup, keep in mind that the IPsec policy that is applied might be a cached copy of the Active Directory-based IPsec policy, or it might be a local IPsec policy. If the computer is assigned Active Directory-based IPsec policy, then the IPsec service attempts to retrieve the latest copy of the assigned IPsec policy from Active Directory before applying the cached copy of the Active Directory-based policy. When doing so, the IPsec service first queries DNS for the current list of the IP addresses of all of the domain controllers. If the IPsec policy objects have been deleted from Active Directory, the cached copy of the Active Directory-based policy is applied instead. The list of domain controller IP addresses in the cached copy of the Active Directory-based IPsec policy might have changed substantially since the IPsec policy backup was created (for example, if new domain controllers were added). If this is the case, communication might be blocked with current domain controllers and therefore Kerberos authentication will fail when attempts are made to establish IPsec-secured connections remotely. In addition, the computer might not be able to receive Group Policy updates. To resolve this problem, do one of the following:

- Access the computer locally, and stop the IPsec service on that computer.
- Restart the computer in Safe Mode with Networking, and either configure the IPsec service to start manually, or disable the IPsec service to allow IPsec-secured communication with the IP addresses of the new domain controllers.

You can also use the IP Security Policy Management **Export Policies** and **Import Policies** menu commands to back up and restore IPsec policies. If you are using Active Directory-based IPsec policy, you must use these commands to back up and restore the IPsec policies themselves.

---

Because the IPSec policies themselves are not stored in Group Policy, you can use Group Policy backup and restore capabilities only to store information about which IPSec policies are assigned to specific Group Policy objects, not to back up and restore the IPSec policies.

---

**Important** It is important to secure your IPSec policy backups. The backup is a file that inherits the NTFS file system permissions of the directory in which it is stored. The data in the file is not encrypted or signed. You should protect the IPSec configuration information in these files with appropriate permissions. Only authorized IPSec administrators should have access to these backup files.

---

### Exporting and Importing IPSec Policies to Back Up and Restore IPSec Policies

If you dedicate one computer to create local IPSec policies, you can export the IPSec policy from this computer to a file, and then import the file into the local IPSec policy store on another computer or into Active Directory.

The **Export Policies** menu command exports all IPSec policy objects from the policy store into one .ipsec file.

Each policy object is internally identified by a globally unique identifier (GUID). Importing an IPSec policy file by using the **Import Policies** menu command either overwrites existing IPSec policy objects that have the same GUID or creates new IPSec policy objects if they do not exist. If you change only the name of an IPSec policy, filter list, or filter action and then reimport the policy file, then these IPSec policy objects are overwritten with the new names. However, in Windows 2000, if you select the **Delete all existing policy information** check box when you select the policy file to import, then all policies, filter actions and filter lists in the target store are deleted before the objects in the export file are written into the target policy store. Note that you cannot export or import a single IPSec policy, filter list, or filter action.

---

**Caution** If you reimport changes to an existing IPSec policy that is already assigned to a GPO, the IPSec policy is unassigned. You must edit the GPO and reassign the IPSec policy after the import is complete. (However, if you use a server running Windows Server 2003 as an IPSec policy management computer, this problem does not occur.)

Additionally, when importing or editing IPSec policy in Active Directory, do not close IP Security Policy Management before all of the IPSec policy data is written to Active Directory. If IP Security Policy Management cannot finish writing all of the policy data into Active Directory, then IPSec policy corruption might result.

**Note** If you detect IPSec policy corruption, you can try to reimport the IPSec policy file. In some cases, the IPSec policy objects must be deleted so that a new IPSec policy import operation can be successfully completed. You must use either LDP.exe or ADSI Edit to delete the IPSec policy objects. LDP.exe is a Windows Support Tool that is included in the Support Tools folder of the Windows 2000 and Windows Server 2003 operating system discs. ADSI Edit is a Windows Support Tool that uses the ADSI and that is also provided on the Windows 2000 and the Windows Server 2003 operating system discs. If you are managing IPSec policy remotely over slow links, then use a file copy technique to transfer the IPSec policies in .ipsec export files before you delete the IPSec policy objects. After you transfer the IPSec policies, use the Terminal Services client (on computers running Windows 2000) to connect to the remote server and perform the import operation quickly.

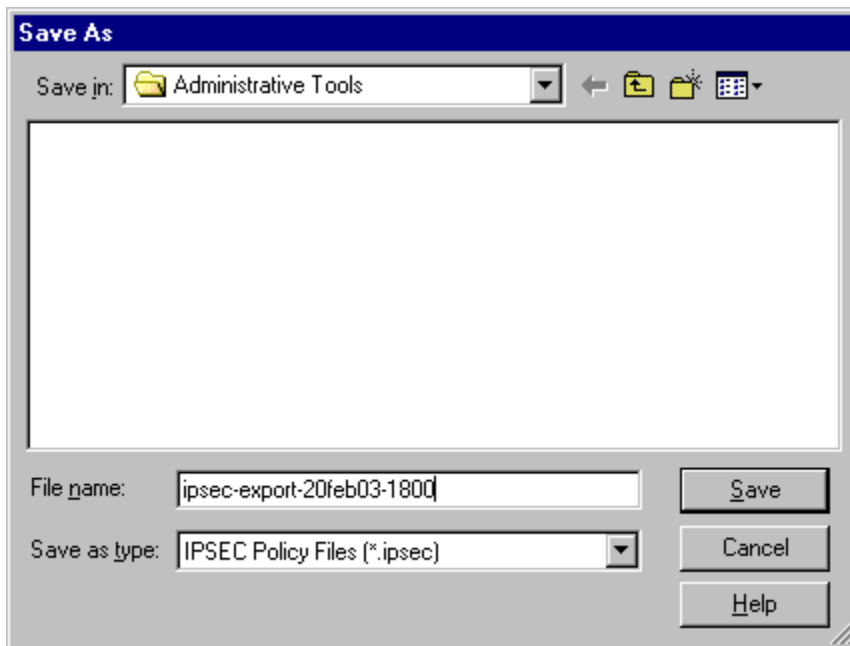
---

---

To export and import Active Directory-based IPsec policies, start IP Security Policy Management from MMC or access IP Security Policy Management from Group Policy (Active Directory), select the appropriate policy, and then follow the procedures in this section.

### To export local IPsec policies

1. Create a console containing IP Security Policies. Or, open a saved console file containing IP Security Policies.
2. In the console tree, click **IP Security Policies on Local Machine**, click **Action**, point to **All Tasks**, and then click **Export Policies**.
3. In **Save As**, specify where to save the .ipsec policy file, and then click **Save**.



---

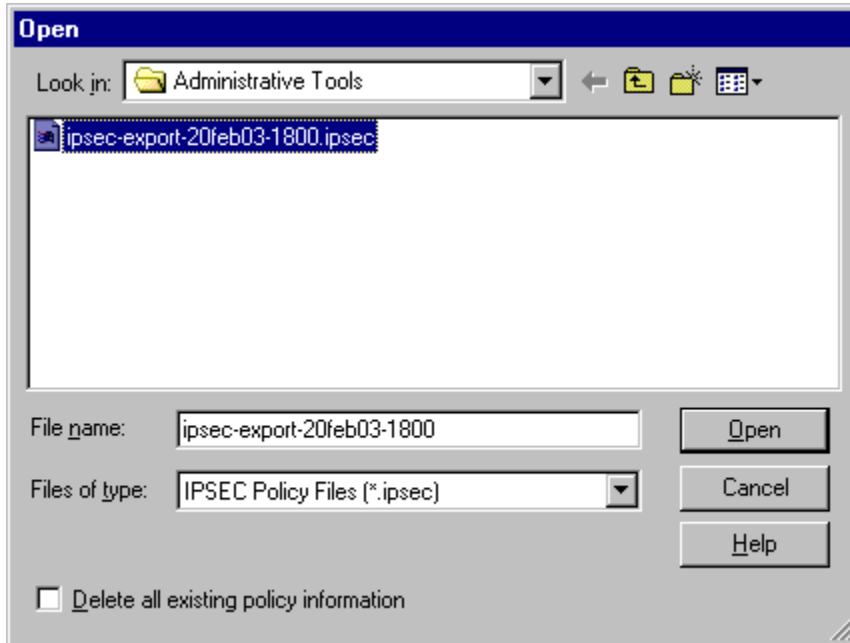
**Note** To ensure proper management of .an ipsec file, use a name that is specific enough to clearly identify the file (for example, you might want to include the date and version number in the file name).

---

### To import local IPsec policies from a file

1. Create a console containing IP Security Policies. Or, open a saved console file containing IP Security Policies.
2. In the console tree, click **IP Security Policies on Local Machine**, click **Action**, point to **All Tasks**, and then click **Import Policies**.

3. In **Open**, specify the .ipsec policy file, and then click **Open**.



## Testing and Monitoring Successful IPsec Operation

Verifying successful IPsec operation is an important, although often neglected, step in deploying IPsec. Incorrectly configuring IPsec and related Windows networking components can cause network traffic to be unintentionally blocked or sent without protection.

Verifying successful IPsec operations is uniquely different than testing other networking components. You can often verify successful operation of a system by performing tests to ensure that the applications work. However, applications in the overall system can work without IPsec. Testing IPsec involves ensuring that the applications work and that IPsec is properly restricting access and performing encryption.

Although the specifics of how to verify successful IPsec operation depend on your individual deployment environment, you can use the information provided in the following sections to identify both correct and incorrect IPsec behavior. To verify the security parameters of SAs in Windows 2000, you can use IPsecmon.exe or **netdiag /test:ipsec /v**.

---

**Note** To test and monitor Windows IPsec policy, you must be a member of the local Administrators group on the computers that you want to manage, or a member of the Domain Admins group.

---

## Gathering IPsec Data for Troubleshooting

Problems with network communications can be difficult to diagnose. When a communication error occurs between two IPsec peers, study the state of all components in the system. Synchronize the precise time between the two peers to assist in the correlation of events. Take specific steps to isolate each potential problem during your testing, and carefully document the results.

When you troubleshoot IPsec communications between a client and a server, it is recommended that you perform the following tasks:

- 
1. Verify that the IPSec service is running on both computers.
  2. Note the operating system that is running on each computer and any installed service packs, hotfixes, security patches, or other updates. Also note security template configurations.
  3. Save the Event Viewer event logs for the application, security, and system logs on each computer.
  4. Run an IPSec command-line tool or other available tools to determine the IPSec policy version and detailed configuration for each computer, and save the output. For example, you can use the **netdiag /test:ipsec /debug** command (for computers running Windows 2000), the **ipseccmd show all** command (for computers running Windows XP), or the **netsh ipsec static show** or **netsh ipsec dynamic show** command (for computers running Windows Server 2003).
  5. Enable tracing for IKE negotiation events, and save the log file (Oakley.log) from both computers.
  6. Conduct testing to reproduce the problem. During your testing, use a tool such as Network Monitor to collect a network trace from at least one computer, and save the trace. You can use network traces from both computers to help identify whether packets are being lost or modified along the communication path between the two computers.
  7. Verify that the proper computer is initiating the IKE negotiation.
  8. Verify that the IKE negotiation packets are being sent to the responder (the destination computer).
  9. Review the network trace and Oakley.log file on the responder to verify that the responder is receiving the IKE negotiation packets.
  10. Review the network trace and Oakley.log file to verify that the responder can properly reply.
  11. Verify that the initiator receives the reply from the responder.

### **Verifying That IPSec Policy Does Not Block Traffic Required by a Server**

The IPSec filtering rules that are provided as an example in this paper cause some Windows networking services to fail. These services fail because they cannot communicate with other computers due to the restrictions imposed by the filters. When communication failures occur, the services generate Event Log errors so that you can locate the cause of the failures.

Before you design an IPSec policy to require that traffic to all destinations be IPSec-secured, study the communications pattern of the server to ensure that such a policy will not block required traffic. Using the example in this paper, you can allow unsecured communication with the server so that you can study the communication pattern of the server. To do this, change the filter action in Rule 2 of the CORPSRV IPSec policy by selecting the **Allow unsecured communication with non-IPSec aware computers** check box. When you select this check box, IKE falls back to clear as needed. If you enable auditing for IKE events, security log success audit event number 541 is generated. This event includes the IP address of the remote access clients. If the attempt to fall back to clear fails, security log failure audit event number 547 is generated. By developing a tool to analyze both success (541) and failure (547) events, you can identify which clients are communicating and how many unique clients are communicating with the server at any given

time. For more information about how to enable auditing for these events, see [“Enabling Security Log Audit Events for IKE Negotiation,”](#) later in this paper.

### Verifying IPSec-Secured Connectivity

If an IPSec policy that is assigned to a server includes filters to help secure ICMP traffic, then run the **ping** command on that server to initiate a security negotiation with another computer (if an SA is not already established with that computer). As shown in the following output, the first **ping** command that is run from CORPSRV sends four ICMP Echo Request messages to the domain controller in the perimeter network. Because Rule 3 of the COPRSRV IPSec policy requires that traffic with the perimeter network domain controller be IPSec-secured, each Echo Request message elicits a Negotiating IP Security response. Meanwhile, IKE is establishing an SA between the two computers. If the **ping** command is run again, the SA is already established, and the remote computer acknowledges the ICMP Echo Request messages with a reply.

```
C:\>ping PNDC

Pinging PNDC [172.17.1.2] with 32 bytes of data:

Negotiating IP Security.
Negotiating IP Security.
Negotiating IP Security.
Negotiating IP Security.

Ping statistics for 172.17.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping PNDC

Pinging PNDC [172.17.1.2] with 32 bytes of data:

Reply from 172.17.1.2: bytes=32 time<10ms TTL=126
Reply from 172.17.1.2: bytes=32 time<10ms TTL=126
Reply from 172.17.1.2: bytes=32 time<10ms TTL=126
Reply from 172.17.1.2: bytes=32 time<10ms TTL=126

Ping statistics for 172.17.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

### Viewing IPSec Events

You can use Event Viewer to view the following IPSec-related events:

- IPSec Policy Agent and IPSec driver events in the system log
- IKE events in the application log
- IKE events in the security log

When you enable auditing for the following Audit Policy security settings, Event Viewer records useful IPSec information as events:

- **Audit logon events.** Determines whether to audit each instance of a user logging on to or logging off from a computer. The creation and deletion of IPSec SAs are audited as network logon events. You must enable this setting to view the success or failure of IKE negotiations in the Event Viewer security log.

- 
- **Audit process tracking.** Determines whether to audit detailed tracking information for events such as program activation, process exit, handle duplication, and indirect object access. In Windows XP and Windows Server 2003, you must enable this setting and the **Audit policy change** setting to view the success or failure of IPSec policy change events. In Windows XP and Windows Server 2003, you need to enable only the **Audit policy change** setting to view the success or failure of IPSec policy change events.
  - **Audit policy change.** Determines whether to audit every incident of a change to user rights assignment policies, audit policies, IPSec policies, or trust policies. When IPSec policy changes are detected, a policy change audit is generated. You can enable this setting to verify which IPSec policies are applied and when they are applied. In Windows 2000, you must enable this setting and the **Audit process tracking** setting to view the success or failure of IPSec policy change events. In Windows XP and Windows Server 2003, you only need to enable this setting to view the success or failure of IPSec policy change events.

To enable auditing for these Audit Policy categories, use Local Security Policy settings (for a local computer) or Group Policy Object Editor (for a domain). You can access Local Security Policy settings or Group Policy Object Editor by opening the appropriate security policy and expanding the console tree as follows: **Computer Configuration\Windows Settings\Security Settings\Local Policies\Audit Policy**.

Before you assign an IPSec policy to a server, it is recommended that you save the event logs that are generated on the server during typical operations so that you can use these logs as a baseline reference for future troubleshooting. Additionally, you should investigate and resolve all failure events in the application, security, and system logs. Microsoft does not currently provide reporting tools to analyze and report events that are generated by IPSec. If you plan to deploy IPSec in a large organization, you might need to develop specific tools to collect, parse, analyze, and generate reports for events that are generated by IPSec.

### **Enabling Security Log Audit Events for IKE Negotiation**

To view the success or failure of IKE negotiation events in the Event Viewer security log, enable success or failure auditing for the **Audit logon events** and **Audit process tracking** audit policies for your domain or local computer. When these two settings are enabled, you can view the success or failure of IKE negotiations in the Event Viewer security log. Relevant events include:

- **Event 541 (success).** Recorded when IKE successfully negotiates either a main mode SA or an IPSec SA. The SA parameters are noted in the text of the event.
- **Event 542 (success).** Recorded when IKE successfully deletes an IPSec SA. An IPSec SA might be deleted because the SA lifetime expired, a new SA was generated during quick mode rekey, the IPSec peer sent a delete message, the IPSec policy changed, or the IPSec service was stopped.
- **Event 543 (success).** Recorded when IKE successfully deletes a main mode SA. An IKE main mode SA might be deleted because the SA lifetime expired, the IPSec peer sent a delete message, the IPSec policy changed, or the IPSec service was stopped.
- **Event 544 (failure).** Recorded when the IKE negotiation is terminated due to a certificate trust failure and subsequent authentication failure. This failure might occur because a valid certificate chain could not be found on the IPSec peer, or the certificate chain that was found

could not be sent to a trusted root CA.

- **Event 545 (failure).** Recorded when the IKE negotiation is terminated due to the validation failure of a computer certificate signature. This event is rare because it indicates that the computer certificate on the IPSec peer has a mismatched RSA type public/private key pair.
- **Event 546 (failure).** Recorded when an SA cannot be established due to an invalid IKE proposal from the IPSec peer. This error typically occurs when an IPSec policy is incorrectly configured.
- **Event 547 (failure).** Recorded when the IKE negotiation fails. The causes of the failure are noted in the text of the event.

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.

#### To disable auditing of IKE events in the security log

---

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

---

1. On a computer running Windows Server 2003, set the **HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\Lsa\Audit\DisableIKEAudits** registry setting to a value of 1.

The **DisableIKEAudits** key does not exist by default and must be created.

2. 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.

#### Enabling Logging for the IPSec Driver

In Windows XP and Windows Server 2003, you can enable logging for the IPSec driver by modifying the registry. In Windows Server 2003, you can enable logging for the IPSec driver by using the Netsh commands for IPSec. When you enable logging for the IPSec driver, bad SPI events (the total number of packets for which the SPI was incorrect), IKE negotiation failures, IPSec processing failures, packets received with invalid packet syntax, and other errors are recorded in the System log. Unauthenticated hashes (with the exception of "Clear text received when should have been secured" events) are also logged.

#### To enable logging for the IPSec driver by modifying the registry

---

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

---

- 
1. On a computer running Windows XP or Windows Server 2003, set the **HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\IPSec\EnableDiagnostics** DWORD registry setting to a value of **1**.

The **EnableDiagnostics** key does not exist by default and must be created.

2. Restart the computer.

#### To enable logging for the IPsec driver by using the Netsh commands for IPsec

1. On a computer running Windows Server 2003, at the command prompt, type:

```
netsh ipsec dynamic set config ipsecdiagnostics 1
```

2. Restart the computer.

---

**Note** In Windows Server 2003, you can enable different levels of logging for the IPsec driver. For example, you can specify a value of **7** to enable logging of dropped inbound and outbound packets. For more information about the different levels of IPsec driver logging available in Windows Server 2003, see [Netsh commands for Internet Protocol security \(IPsec\)](http://go.microsoft.com/fwlink/?LinkId=19315) at <http://go.microsoft.com/fwlink/?LinkId=19315>.

---

In Windows 2000, Windows XP, and Windows Server 2003, you can also change the interval for logging IPsec driver events to the System log. By default, the IPsec driver logs events to the System log once an hour or after a threshold for the number of events has been reached. For troubleshooting, you should set this interval to the minimum value, 60 seconds. Valid decimal values, specified in seconds, range from 60 through 86400.

#### To change the interval for logging IPsec driver events by modifying the registry

---

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

---

1. On a computer running Windows 2000, Windows XP, or Windows Server 2003, set the **HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\IPSec\LogInterval** DWORD registry setting to **60 decimal**.

The **LogInterval** key does not exist by default and must be created. Also, **60 decimal** is specified as an example for troubleshooting, but you can specify any other decimal value within the valid range, as required.

2. Restart the computer.

The IPsec driver reads the registry during computer startup.

#### To change the interval for logging IPsec driver by using the Netsh commands for IPsec

1. On a computer running Windows Server 2003, at the command prompt, type:

```
netsh ipsec dynamic set config ipsecloginterval 60
```

**60** is specified as an example for troubleshooting, but you can specify any other value within the valid range, as required.

2. Restart the computer.

---

The IPsec driver reads the registry during computer startup.

### Evaluating Bad SPI Events

Bad SPI events indicate the total number of packets for which the SPI was incorrect. The IPsec driver records bad SPI events in the Event Viewer system log when it receives an IPsec-formatted packet that it cannot interpret. These events, which appear in the system log as “Event 4283: *Number* packets discarded due to bad SPI,” are usually benign.

You can expect a small number of bad SPI events per each IPsec peer IP address, due to the way in which IKE processes the transition of IPsec SAs during rekeys. This event is usually logged when an SA has been deleted on one IPsec peer, and the other IPsec peer still sends secure traffic on the deleted SA. This event is also logged when one IPsec peer begins sending IPsec-protected traffic using a new SA that the other peer is not yet ready to receive. These brief intervals during rekeys typically do not cause problems for traffic over upper-layer protocols, which can accommodate the loss of a few packets. This number is likely to increase if rekey intervals are short, if communication is running extremely fast, or if there is a large number of SAs. On computers running Windows 2000, you can use the **netdiag /test:ipsec /v** command to display the number of IKE rekeys. If the number of rekeys is very large compared to the amount of time that the connections have been active, consider setting longer key lifetimes in the IPsec policy.

---

**Note** You cannot disable logging for bad SPI events in Windows 2000. Windows XP and Windows Server 2003 do not log these events by default.

---

### Viewing IPsec and Other Network Communication

You can install and use Network Monitor to verify that IPsec packets are encrypted and that only allowed traffic is accepted. Network Monitor is a software-based protocol tracing and analysis tool that is available on the Microsoft Systems Management Server (SMS) operating system disc and as a component installation in Windows 2000 Server and Windows Server 2003. Network Monitor version 2.0 includes parsers for the ISAKMP (IKE), AH, and ESP protocols.

Network Monitor parses AH traffic into upper-layer protocols in the Network Monitor capture. For this reason, the AH traffic does not appear as AH packets, but rather as TCP, UDP, or other upper-layer transport protocol packets to facilitate viewing. When the Frame Viewer window displays the packet headers, the AH header is visible.

The following table shows an example of a list of frames, as they might appear in the Summary pane of the Network Monitor Frame Viewer window, if you monitored traffic between CORPSRV and PNDC, the perimeter network domain controller. The list of frames shows only ISAKMP and IPsec ESP-encrypted traffic between CORPSRV and PNDC.

**Table 7 Sample Network Monitor Capture for Packets Sent between CORPSRV and PNDC**

Frame	Time	Src MAC Addr	Dst MAC Addr	Protocol	Description	Src Other	Dst Other	Type
58	5.518155	CORPSRV	PNDC	ISAKMP	Major Version: 1 Minor Version: 0 Length: ...	CORPSRV	172.17.1.2	IP
59	5.518155	PNDC	CORPSRV	ISAKMP	Major Version: 1 Minor Version: 0 Length ...	172.17.1.2	CORPSRV	IP
60	5.518155	CORPSRV	PNDC	ISAKMP	Major Version: 1 Minor Version: 0 Length: 52	CORPSRV	172.17.1.2	IP
61	5.558214	PNDC	CORPSRV	ISAKMP	Major Version: 1 Minor Version: 0 Length: 84	172.17.1.2	CORPSRV	IP
62	5.558214	CORPSRV	PNDC	ESP	SPI = 0x818152DA, Seq = 0x1	CORPSRV	172.17.1.2	IP
63	5.558214	PNDC	CORPSRV	ESP	SPI = 0x1F432CE3, Seq = 0x1	172.17.1.2	CORPSRV	IP
64	6.549827	CORPSRV	PNDC	ESP	SPI = 0x818152DA, Seq = 0x2	CORPSRV	172.17.1.2	IP
65	6.559842	PNDC	CORPSRV	ESP	SPI = 0x17432CM3, Seq = 0x2	172.17.1.2	CORPSRV	IP
66	7.551307	CORPSRV	PNDC	ESP	SPI = 0x813F52DA, Seq = 0x3	CORPSRV	172.17.1.2	IP
67	7.551307	PNDC	CORPSRV	ESP	SPI = 0x1F432CED, Seq = 0x3	172.17.1.2	CORPSRV	IP

The Network Monitor parser in Windows 2000 cannot interpret ESP traffic. In Windows Server 2003, the parser can interpret ESP traffic, if an IPSec hardware acceleration adapter performs encryption or decryption of this traffic, or if you use ESP without encryption.

For information about how to install the version of Network Monitor that is available in Windows 2000, see [To Install Network Monitor](http://go.microsoft.com/fwlink/?LinkId=18749), at <http://go.microsoft.com/fwlink/?LinkId=18749>. For information about SMS, see the [Systems Management Server Web site](http://go.microsoft.com/fwlink/?LinkId=299), at <http://go.microsoft.com/fwlink/?LinkId=299>.

### Managing and Monitoring IPSec on Computers Running Windows XP

For computers running Windows XP, you can use Ipseccmd.exe to script the creation of local or Active Directory-based IPSec policy and to display IPSec policy assignments, active SAs, and detailed IPSec policy settings. To display active SAs, filters, and other IPSec statistics, type **ipseccmd show all** at the command prompt. If you assign Active Directory-based IPSec policy to

---

computers running Windows XP, you can also use the Netdiag.exe command-line tool to display information about which IPsec policy is being applied.

As mentioned earlier, Ipseccmd.exe is a Windows Support Tool that is included in the Support Tools folder of the Windows XP operating system disc. For more information about Ipseccmd.exe, see [ipseccmd](http://go.microsoft.com/fwlink/?LinkId=18755), at <http://go.microsoft.com/fwlink/?LinkId=18755>.

### Viewing Network and IPsec-Related Information

The Netdiag.exe command-line tool provides detailed information about the state of networking and distributed system components such as Kerberos. You can use this tool on computers running Windows 2000 to test basic networking capabilities, diagnose connectivity problems, and display IPsec policy details and statistics.

When you type **netdiag** at the command prompt on a computer running Windows 2000, a series of networking-related tests is performed. To verify whether an IPsec policy is assigned to a computer and to display IPsec policy details, use the **netdiag /test:ipsec** command. To display IPsec statistics and SAs, use the **netdiag /test:ipsec /v** command. To view all filter settings for the assigned IPsec policy, use the **netdiag /debug** command.

To run all Netdiag.exe commands, you must be a member of the local Administrators group. To use the **/debug** parameter to view when an Active Directory-based IPsec policy is assigned to a computer, you must be a member of the Domain Admins group.

Netdiag.exe is available in Windows 2000, Windows XP, and Windows Server 2003 as follows:

- For Windows 2000, Netdiag.exe is a Windows Support Tool that is included in the Support Tools folder of the Windows 2000 operating system disc. An updated version of Netdiag.exe for Windows 2000 is available for download from the Web. For more information, see [Netdiag.exe: Network Connectivity Tester](http://go.microsoft.com/fwlink/?LinkId=19316), at <http://go.microsoft.com/fwlink/?LinkId=19316>. Netdiag.exe was enhanced in Windows 2000 Service Pack 1 to display the number of bytes offloaded by the IPsec driver to a network adapter that is capable of IPsec hardware offload. The hardware offload statistics are Offloaded Bytes Sent and Offloaded Bytes Received.
- For Windows XP, Netdiag.exe is a Windows Support Tool that is included in the Support Tools folder of the Windows XP operating system disc (select the **Complete** setup option). The display of detailed IPsec policy information and IPsec statistics (provided by the **/debug** and **/v** options in Windows 2000) is not supported in Windows XP. Instead, to view details about IPsec policies, use the **ipseccmd show** command.
- For Windows Server 2003, Netdiag.exe is also available as a Windows Support Tool that is included in the Support Tools folder of the Windows Server 2003 operating system disc (select the **Complete** setup option). Although you can still use this version of Netdiag.exe to obtain basic networking information that is not IPsec-specific, the Netsh commands for IPsec replace all IPsec-specific functionality. To view details about IPsec policies, use either the **netsh ipsec static show** command or the **netsh ipsec dynamic show** command.

The design of an IPsec policy might determine the type of traffic that can be sent by Netdiag.exe, and therefore the ability of Netdiag.exe to verify network and domain connectivity. After you verify that IPsec is functioning correctly, it is recommended that you save Netdiag.exe diagnostic output

---

to serve as a baseline and that you train administrators accordingly, in case troubleshooting is required.

### Enabling Detailed Tracing for IKE Negotiations

Enabling audit logging for IKE events and viewing the events in Event Viewer are the fastest and simplest ways to troubleshoot failed main mode or quick mode negotiations. However, some scenarios might require a more detailed analysis. The IKE tracing log is a very detailed log intended for troubleshooting IKE interoperability under controlled circumstances. Keep in mind that the details of this tracing log are not documented, and advanced knowledge of ISAKMP RFC 2408 and IKE RFC 2409 is required to interpret this log. However, experienced IPSec administrators may find it useful. You can enable tracing for IKE negotiations if the audit failure events do not provide enough information.

#### To enable the IKE tracing log by modifying the registry

---

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

---

1. On a computer running Windows 2000 or Windows XP, set the **HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\PolicyAgent\Oakley\EnableLogging** registry setting to **1**.

The Oakley key does not exist by default and must be created.

2. 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.

In Windows Server 2003, you can enable or disable the IKE tracing log dynamically while the IPSec service is running by using the Netsh commands for IPSec.

#### To enable the IKE tracing log by using the Netsh commands for IPSec

- On a computer running Windows Server 2003, at the command prompt, type **netsh ipsec dynamic set config ikeLogging 1**.

This command creates the IKE tracing log file if it does not exist. If the file does exist, it appends logging information to the existing file.

---

**Note** To disable the IKE tracing log, on a computer running Windows Server 2003, at the command prompt, type **netsh ipsec dynamic set config ikeLogging 0**.

---

The IKE tracing log appears as the *systemroot\Debug\Oakley.log* file. A new Oakley.log file is created each time the IPSec service is started and the previous version of the Oakley.log file is saved as Oakley.log.sav. The log is limited to 50,000 lines. When the Oakley.log file becomes full, the current file is saved as Oakley.log.bak, and a new Oakley.log file is created.

Because many IKE negotiations can occur simultaneously, you should minimize the number of negotiations and enable the IKE tracing log for as briefly as possible to capture a more easily

---

interpreted log. Use the ISAKMP cookie pair, IP addresses, SPI, timestamps, and SA identifiers to identify messages related to one security negotiation or IPsec SA processing.

### **Monitoring IPsec Status Information**

After you assign an IPsec policy to a computer and IPsec-secured communication is established, you can use `Ipsecmon.exe` for monitoring of active SAs on computers running Windows 2000. To use this tool, run **ipsecmon** at the command prompt on the computer that is being monitored. Keep in mind, however, that `Ipsecmon` provides limited capabilities.

In Windows XP and Windows Server 2003, you can use the IP Security Monitor snap-in to monitor IPsec information for your local computer and for remote computers. Although IP Security Monitor provides significantly improved monitoring capabilities, you cannot use it for remote monitoring of computers running Windows 2000. For remote monitoring, you can use IP Security Monitor but only to monitor computers that are running the same version of the Windows operating system, unless you apply the update to IP Security Monitor as described in article 818043, "L2TP/IPsec NAT-T Update for Windows XP and Windows 2000," in the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>. If you apply this update to a computer running Windows XP, you can use IP Security Monitor on that computer to monitor computers running Windows Server 2003. Likewise, you can use IP Security Monitor on computers running Windows Server 2003 to view computers running Windows XP, if the update has been applied to the computers running Windows XP.

If you do not have this Windows XP update installed, or if you are running Windows 2000, to remotely monitor IPsec on a computer that is running a different version of Windows than your computer is running, you must use Terminal Services or Remote Desktop Connection.

---

## Configuring IPsec Policies: Security Considerations

When you configure an IPsec policy to help secure a server, keep the following security considerations in mind.

### Security During Computer Startup

In Windows 2000 and Windows XP, the IPsec service does not provide security during computer startup. As a result, IPsec does not protect traffic that is sent or received during this time. After the IPsec service starts and applies the assigned IPsec policy, the traffic will match IPsec filters and can trigger security negotiations. Windows Server 2003 IPsec provides new features used for protection during computer startup. For more information about these features, see Chapter 6, “Deploying IPsec,” in *Deploying Network Services*, in the [Windows Server 2003 Deployment Kit](http://go.microsoft.com/fwlink/?LinkId=8195), on the Web at <http://go.microsoft.com/fwlink/?LinkId=8195>.

In the Windows implementation of IPsec, the IPsec service starts automatically, by default. A delay in any service that starts automatically can delay all other dependent services that also start automatically. Therefore, to minimize the time required for a computer to start and to allow for IPsec policy to be applied as quickly as possible during this time, the IPsec service is configured by default to not have other dependent services. If the IPsec service is configured to have a dependent service, the IPsec service might be unable to apply the assigned IPsec policy before the dependent service is notified that the IPsec service is ready.

Additionally, in Windows 2000 and in Windows Server 2003, the IPsec service does not start when the computer is started in Safe Mode with Networking or when Directory Services Restore Mode is used. As a result, IPsec does not protect traffic that is sent or received during these times. However, in Windows Server 2003, if you have already assigned an IPsec policy to a computer, by default, the IPsec driver provides stateful filtering of inbound traffic to that computer for both Safe Mode and Directory Services Restore Mode.

---

**Note** In Windows 2000 and Windows XP, if the IPsec service fails or does not start for any reason, inbound and outbound traffic is permitted (that is, it is not filtered by IPsec or IPsec-secured). When you stop the IPsec service manually, it unassigns all IPsec policy and deletes all IPsec and IKE SAs.

In Windows Server 2003, if the IPsec service fails, the IPsec driver is set to block mode, and all inbound and outbound traffic is blocked, except for that which matches any exemptions that you configure (by using the **netsh ipsec dynamic set config bootexemptions** command). If the IPsec service does not start, the IPsec driver continues to provide stateful filtering of inbound traffic. When you stop the IPsec service manually on a computer running Windows Server 2003, the IPsec driver unassigns all IPsec policy and is set to permit mode, so it does not filter any traffic. Additionally, all IKE and IPsec SAs are deleted.

---

### Impact of Group Policy Security Settings on IPsec

If you use Kerberos as an authentication method, and you use Group Policy security settings to restrict access to a server (by assigning either the **Access this computer from the network** or **Deny access to this computer from the network** rights), the Group Policy settings might inadvertently influence IKE negotiation.

---

In this case, IKE verifies access controls, but only when it is a responder. IKE does not verify access controls when it initiates the negotiation. Keep in mind that regardless of which side of the communication initiates the IKE negotiation, IKE rekeying might reverse the roles of initiator and responder. Because IKE uses the domain computer account for authentication, Kerberos authentication might fail if you do not configure Group Policy settings to allow computer accounts access from the network. For a domain controller, these settings are typically not changed. However, some administrators might modify these settings to restrict user logons, not realizing the potential impact to IKE computer authentication.

---

**Note** Other Group Policy security settings, such as **Digitally sign client communication**, **Additional Restrictions for Anonymous Connections**, **Amount of idle time required before disconnecting session**, and **Secure Channel: Digitally encrypt or sign secure channel data**, do not influence how IPSec SAs are negotiated or managed.

---

---

## Considerations for Crossing Security Boundaries with IPSec-Secured Traffic

When you design an IPSec policy, it is important to consider security boundaries, such as firewalls, and different domains forest trust environments, that IPSec-secured traffic must cross.

### Creating Firewall Filters to Permit ISAKMP, AH, and ESP Traffic

In some cases, IPSec-secured traffic might need to pass through a router, firewall, or other filtering device. In the case of a router, unless the router filters TCP and UDP traffic or other upper-level protocol headers, no special configuration is required to permit the IPSec-secured traffic to be forwarded. In the case of a filtering router or a firewall, you must configure these devices to allow IPSec-secured traffic to be forwarded.

In the scenario used in this paper, IPSec-secured traffic from CORPSRV can travel on the local Ethernet, across ROUTER3, to CORPDC. Because ROUTER3 is not a filtering router, it does not need to be configured to forward the IPSec-secured traffic. However, IPSec-secured traffic from CORPSRV can also travel through the firewall to the perimeter network domain controller. In this case, you must configure the firewall to forward this traffic.

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.

To permit IPSec traffic on UDP source and destination port 500 (ISAKMP), use the following settings to create a firewall filter called "Permit ISAKMP traffic on UDP port 500":

- Source address = *SpecificIPAddress*
- Destination address = *SpecificIPAddress*
- Protocol = UDP
- Source port = 500
- Destination port = 500

To permit IPSec traffic on IP protocol 50 or IP protocol 51 (AH), use the following settings to create a firewall filter called "Permit IPSec traffic on ESP or AH protocol (50 or 51)":

- Source address = *SpecificIPAddress*
- Destination address = *SpecificIPAddress*
- Protocol = 50 or 51

In addition, when you configure the firewall, do the following:

- Configure the firewall to permit traffic between only the specific IP addresses of the IPSec peers.
- Configure the firewall filter to permit or track fragments for ISAKMP, AH, and ESP traffic. In Windows 2000 Service Pack 1 or later, Windows XP, and Windows XP Service Pack 1, IKE message fragmentation is required when certificate authentication is used. Also, many UDP

---

applications do not attempt to avoid fragmentation, and, therefore, the UDP traffic is fragmented when IPSec protects it.

- Do not configure the firewall to perform stateful filtering on UDP source and destination port 500 (ISAKMP), IP protocol 51 (AH), or IP protocol 50 (ESP).
- If communication requires TCP PMTU discovery, configure the firewall to permit ICMP Destination Unreachable messages.

For more information, see article 233256, "How to Enable IPSec Traffic Through a Firewall," in the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

---

**Note** Windows IPSec in transport mode is designed to work with Point-to-Point Protocol (PPP) dial-up connections and PPTP or L2TP/IPSec VPN connections. 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.

---

### Considerations for NAT Traversal

In Windows 2000 and Windows XP, if traffic between the client and a server must pass through a network address translator, then IPSec cannot secure the traffic (the IKE negotiation will fail when translated by a network address translator). Windows Server 2003 provides support for version 2 of a new IETF design called IPSec network address translation (NAT) traversal (IPSec NAT-T). IPSec NAT-T allows IPSec ESP packets in either transport mode or tunnel mode to pass through network address translators that allow UDP traffic. In this design, IKE automatically detects network address translators 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 use the example policies in this paper 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, as described in this paper, 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 IPSec 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 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 IETF 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 on 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](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

### **Considerations for Securing Traffic Required for Remote Management and Monitoring of IPSec Policy**

Because you might need to remotely manage and monitor IPSec policy, it is critical to properly secure the traffic that is required to do so. A successful attack on traffic that is required for the remote management of IPSec policy can compromise all IPSec-secured traffic. Therefore, it is recommended that the security for remote management and monitoring traffic be as strong as the security IPSec provides for all other communications. It is recommended that you use IPSec to help secure the traffic that is required for the remote management and monitoring of IPSec policy. Using IPSec to help protect remote management and monitoring traffic provides defense-in-depth against vulnerabilities in the management protocols. The following sections describe the security considerations for protocols that are used when you remotely manage and monitor IPSec policy.

#### **Security Considerations for RPC and SMB Remote Management Traffic**

When you use the IP Security Policy Management snap-in to manage a locally assigned IPSec policy on a remote computer, it uses remote registry APIs to connect to the remote computer and when you edit the local IPSec policy for that computer. These APIs use the RPC and SMB protocols. The security for this traffic is provided by the negotiation of RPC and SMB security parameters that are controlled by the remote registry APIs. You can configure security for RPC and SMB either through Local Security Policy settings (for a local computer) or Group Policy Object Editor (for a domain). To access these settings, open the appropriate security policy and expand the console tree as follows: **Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options**.

---

RPC and SMB security parameters affect the use of SMB and RPC not just by the remote registry APIs, but by the operating system and all services and applications. To preserve compatibility across different Windows operating systems and applications and to support scenarios that involve traffic across different domains, you might not be able to configure these settings for maximum security. However, you can use IPSec itself to help secure traffic on a specific path that is required for IPSec policy management. For information about security settings that affect the use of RPC and SMB, see [Threats and Countermeasures Guide](#), at <http://go.microsoft.com/fwlink/?LinkId=19618>.

---

**Note** The SMB protocol does not support the encryption of traffic. It is therefore recommended that you use IPSec ESP encryption to help secure remote management traffic when you use IP Security Policy Management to remotely manage IPSec policy.

---

### **Security Considerations for RDP Remote Management and Monitoring Traffic**

Using IP Security Policy Management for remote management of local IPSec policy and using IP Security Monitor for remote monitoring of local IPSec policy is only supported between computers running the same version of the Windows operating system. Therefore, it is recommended that you use the Terminal Services client (for Windows 2000) or Remote Desktop Connection (for Windows XP and Windows Server 2003), to connect to the computer that you want to manage or monitor, and then use IP Security Policy Management or IP Security Monitor on that computer, as needed. Both the Terminal Services client and Remote Desktop Connection require the remote desktop protocol (RDP), which uses TCP port 3389. However, the security methods that are used in RDP might not be as strong as those provided by the IPSec protocol. It is therefore recommended that you use IPSec ESP encryption to provide defense-in-depth security for Terminal Services client or Remote Desktop Connection traffic that is required for remote management of IPSec policy. To do this, assign an Active Directory-based policy at the domain level or a default local IPSec policy to both the IPSec management computer and the remote computer (the computer to which the IPSec policy management computer will connect). Add a rule to the policy to help secure traffic that is sent over the RDP protocol. In the example policies in this paper, all traffic to CORPSRV is secured by IPSec, including RDP traffic.

When you use IPSec, you can help protect remote management traffic against specific vulnerabilities in the protocols that are required for that traffic. For example, an identified security vulnerability was found in the implementation of session encryption in certain versions of the Microsoft implementation of RDP. In these versions of RDP, which are included in Windows 2000 and Windows XP, the checksums for the plaintext session data are not encrypted. Because IPSec ESP encrypts the entire IP payload, this encapsulation method encrypts the RDP protocol checksum of the session data, thus providing protection against the vulnerability. Although a supported fix is now available from Microsoft, as described in article 324380, "MS02-051: Cryptographic Flaw in RDP Protocol Can Cause Information Disclosure," in the [Microsoft Knowledge Base](#), at <http://go.microsoft.com/fwlink/?LinkId=16462>, using IPSec can provide protection against this vulnerability. Also, the example CORPSRV IPSec policy filters can help protect the Remote Desktop service and Remote Services client against untrusted computers performing attacks over RDP port 3389.

---

For information about how to use IPSec to help protect RDP, see article 816521, "HOW TO: Use IPSec Policy to Secure Terminal Services Communications in Windows Server 2003," in the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

### **Authentication Considerations for Remotely Managing IPSec Policy**

To use IP Security Policy Management to manage locally assigned policy for a remote computer, by default you must be logged on to the IPSec management computer as a member of the local Administrators group on the remote computer. If you are not logged on as a member of the local Administrators group on the remote computer, on the management computer, you must add the IP Security Policy Management snap-in, save it as an MMC console file, and then, open the saved MMC console file by using the **runas** command. To do so, at the command prompt, on the management computer (the computer from which you initiate the connection), type **runas**, and specify the credentials required to log on as a member of the local Administrators group on the remote computer.

If your IPSec policy specifies that Kerberos be used for IKE authentication between the IPSec management computer and the remote computer, the two computers must be in the same domain or in a trusted domain. Additionally, you must allow traffic over the ports required by RPC, Terminal Services, or Remote Desktop Services.

### **Considerations for Different Trust Environments**

When you design an IPSec policy, consider any logical boundaries that might affect IPSec-secured communications. For example, your domain and trust environment is critical in determining an appropriate IKE authentication method.

Kerberos is recommended for use in a two-way (mutual) domain and forest trust environment. In the scenario used in this paper, Kerberos is used for IKE authentication between CORPCLI and CORPSRV because both computers share a mutually trusted domain. You can use Kerberos for IKE authentication across trusted domains, if the domains are in the same forest or different forests. If the two domains are in different forests, you must configure two external trusts, one for each direction, between the domains. The external trusts must use the fully qualified domain name (FQDN) of the domains, and you must configure the IPSec policy on both computers to allow an IKE initiator to communicate to any domain controller in the forest domain hierarchy, so the initiator can obtain a Kerberos ticket from a domain controller in the responder's domain. If the domains are on opposite sides of a firewall, then you must configure the firewall to allow Kerberos traffic over UDP and TCP destination port 88, as well as UDP destination port 389.

For information about how use IPSec to help secure Windows 2000 Active Directory communication on opposite sides of a firewall, see "Active Directory in Networks Segmented by Firewalls," available at the [Download Center](http://go.microsoft.com/fwlink/?LinkId=3457), at <http://go.microsoft.com/fwlink/?LinkId=3457>.

Trusts between a perimeter network and internal corporate network are generally considered a security risk. For information about security considerations for using Active Directory in perimeter networks, see Chapter 5, "Security Design," in the [Reference Architecture Guide: Internet Data Center](http://go.microsoft.com/fwlink/?LinkId=16468), at <http://go.microsoft.com/fwlink/?LinkId=16468>, and Chapter 8, "Directory Services," in the [Reference Architecture Guide: Enterprise Data Center](http://go.microsoft.com/fwlink/?LinkId=16463), at <http://go.microsoft.com/fwlink/?LinkId=16463>.

---

## Potential Issues for Perimeter Network Security

As a security best practice, it is generally recommended that you do not place internal domain members in a perimeter network. If WEBSRV were a member of the internal domain, this computer would likely have full access to CORPDC for Group Policy updates, the Net Logon service, Kerberos authentication, and other functions. Although it might be a problem for WEBSRV to have access to CORPCLI, a malicious user with full control over a trusted domain member presents an even greater threat. As noted already, IPsec (and network-layer security) can do very little against the trusted attacker, particularly when attacks are mounted against a domain controller from a domain member.

The section "[Modifying CORPSRV and WEBSRV Policy Design To Defend against Perimeter Network Server \(WEBSRV\) Compromise](#)," earlier in this paper described how to modify the IPsec policy assigned to WEBSRV and CORPSRV to allow only HTTP traffic over TCP port 80. If you misconfigure the IPsec filtering for ports, an attacker might be able to use IPsec to gain greater access than expected to CORPSRV. Also, if an attacker compromises the administrator account on WEBSRV, the attacker can create an IPsec policy to specify that IPsec ESP be used to negotiate security for all traffic between WEBSRV and CORPSRV or other computers. As a countermeasure, configure your firewall to limit the internal corporate network IP addresses with which WEBSRV can negotiate security. Also, do not assign an IPsec policy to internal corporate network clients and servers that specifies an IKE authentication method that can be successfully used to communicate with WEBSRV.

## Security Risks of Enabling the Default Response Rule

When you enable the default response rule in an IPsec policy that is assigned to a computer, IKE responds to any request to negotiate security with that computer. IKE also responds to any request to negotiate security if you assign an IPsec policy with a **My IP Address to Any IP Address** filter to a computer. The difference between the default response rule and the **My IP Address to Any IP Address** filter is that the **My IP Address to Any IP Address** filter triggers a security negotiation when outbound traffic is sent. The default response rule does not trigger a security negotiation until it responds and successfully establishes dynamic filters for the IPsec SA pair. To avoid the security risks related to unwanted security negotiations, you might need to disable the default response rule.

For example, if you assign CORPCLI Policy 2 to an Internet-facing client laptop computer and enable the default response rule (as is done in Rule 2 of CORPCLI Policy 2), then an attacker might be able to obtain information about that computer through the security negotiation. A skilled Internet attacker can construct specific security negotiation requests to query and obtain the name of the client, trust relationships, and other settings that are configured in the default response rule. For example, if you use Kerberos as the authentication method for the default response rule, then the attacker can query the Kerberos identity of the client. The query results will provide the attacker with the computer name and domain hierarchy, such as corpcli@corpdc.corpforest.com. The query does not reveal user information, unless the computer name includes information about the user. For example, a user account is often in the same domain as the user's computer. In addition, if the domain names are descriptive, then the attacker might be able to gain information about the type of user that owns the computer.

---

If you use certificate-based authentication as the authentication method for the default response rule, then the attacker can obtain the names of the PKI trusted root CAs that are configured for the default response rule. If a company uses an internal corporate network root CA, the CA root certificate probably contains the name of the corporation, and therefore the attacker can determine that the client belongs to that corporation. Although less information is revealed in this case than is revealed by the Kerberos identity of the computer, the unwanted disclosure of the name of a corporation still presents a potential security risk.

Additionally, when you enable the default response rule, IKE might respond to requests to negotiate security, even when you configure the IPSec policy to prevent communication with the computer initiating the request. In Windows XP, Windows XP Service Pack 1, and Windows Server 2003, a trusted remote computer can successfully negotiate security and establish authenticated IPSec communications with an IPSec-secured computer, even when a filter exists to block traffic with the IP address of the remote computer. Although traffic that is not IPSec-secured is blocked, a policy with blocking filters does not prevent trusted remote computers from using IPSec to gain access to another IPSec-secured computer.

For example, if you assign CORPCLI Policy 2 to a client, and the default response rule is enabled for that policy, then SAs might be negotiated between CORPCLI and WEBSRV, or between CORPSRV and another computer in the perimeter network subnet, if other security measures fail. This situation might occur if all of the following conditions are true:

- The trust relationship for Kerberos, or for the PKI that is specified as the authentication method for the default response rule, exists between CORPCLI and a perimeter network computer (not WEBSRV).
- The firewall is compromised or misconfigured to permit any other perimeter network computer (not WEBSRV) to negotiate security with CORPCLI.
- A perimeter network computer is compromised or misconfigured to use IPSec policy to negotiate security with computers on the internal corporate network subnet.

The filter for CORPCLI Policy 2 Rule 1 that blocks all traffic with a source address of the perimeter network subnets matches unsecured traffic from the perimeter network subnets and block this unsecured traffic. However, this rule does not block security negotiations or IPSec-secured traffic from the perimeter network subnets. Therefore, if a perimeter network computer initiates a security negotiation with CORPCLI, CORPCLI will reply and attempt to negotiate security by using the security settings that are specified for the default response rule.

When the default response rule successfully negotiates IPSec SAs, a dynamic filter is created in response to the filter that the perimeter network computer proposes. The dynamic filter helps secure all traffic from the IP address of the client to the IP address of the perimeter network computer, using ESP 3DES/SHA1. This filter is more specific than the filter for Rule 1 of CORPCLI Policy 2, which blocks all traffic from the perimeter network subnets. As a result, inbound and outbound traffic matches only the dynamic filter and is secured by IPSec. If all of the conditions noted earlier in this section are true, then the perimeter network computer will gain IPSec-secured access to CORPCLI and potentially to all computers in the same domain as CORPCLI. However, due to the many other security mechanisms in place, there is minimal risk of computers in the perimeter network subnet gaining trusted access to CORPSRV, CORPCLI, or other computers that are in the same domains as these computers.

---

In other cases, the security risks introduced by enabling the default response rule are greater. For example, consider a network environment in which two competitive business groups use the same company network and are members of the same Active Directory domain (or are members of separate domains between which one-way or two-way trusts are established). If the computers in the two business groups belong to well-defined subnets, an IPsec administrator might create filters in the IPsec policy that is assigned to computers in each business group to block traffic from the subnets or specific IP addresses of the computers in the other business group. In this case, enabling the default response rule might introduce a greater security risk because remote computers are trusted using Kerberos authentication, based on their domain trust relationships.

If you must enable the default response rule for IPsec deployment in this scenario, to mitigate the security risks of doing so, consider using specific IPsec configuration strategies or use a firewall to filter traffic or router-based filtering.

Because Active Directory domains are trusted to enable a user authentication method or a directory service, consider configuring your IPsec policy to use certificate authentication rather than to use Kerberos for IKE authentication. If you decide to use certificate authentication, then do the following:

1. Move the computer accounts for each business group into separate Active Directory OUs.
2. Create two internal root CAs as enterprise CAs in Active Directory, one for each business group.
3. Assign a different certificate auto-enrollment policy to each business group OU so that the computers in each business group receive computer certificates from different root CAs.
4. Create a different IPsec policy for each business group and assign the appropriate policy to each business group OU. Make sure that the default response rule and other rules for each policy use the appropriate root CA for IKE authentication.

Alternatively, consider using firewall filters or router-based filtering. If traffic in your network is routed through a single point, at which it is reasonable to place a firewall, consider using the firewall to filter traffic, rather than configuring IPsec policies with filters to block traffic. If a firewall is not the best solution, you might also consider router-based filtering.

Finally, before implementing IPsec policy to help secure a server, make sure that you thoroughly understand how IPsec functions and the defense that you expect IPsec to provide against attacks from other computers. Testing should be completed to verify that IPsec secures traffic as expected and that IPsec provides the expected defense against attacks.

### **Security Risks of Receiving Unsecured ICMP Protocol Traffic**

Before configuring an IPsec policy rule to either block or help secure ICMP traffic, it is important to understand ICMP message types and the potential impact of blocking or securing these messages. An IPsec policy rule that either blocks or helps secure ICMP messages might disrupt TCP/IP communications, particularly if TCP PMTU discovery is disabled. At the same time, allowing unsecured ICMP traffic can expose your network to potential security risks. For more information, see [ICMP Attacks Illustrated](http://go.microsoft.com/fwlink/?LinkId=18765), at <http://go.microsoft.com/fwlink/?LinkId=18765>.

---

For more information about ICMP message types, see “Internet Control Message Protocol (ICMP)” in [Microsoft Windows 2000 TCP/IP Implementation Details](http://go.microsoft.com/fwlink/?LinkId=16467), at <http://go.microsoft.com/fwlink/?LinkId=16467>.

Following are the risks of allowing the receipt of unsecured ICMP traffic:

- **Denial-of-service attack or the ability of an attacker to gain control over a computer through the use of malformed ICMP messages.** If an error occurs in the processing of Windows TCP/IP ICMP messages, IPsec does not defend against an attacker exploiting that error with a malformed ICMP message. An attacker can use a malformed ICMP message to mount a denial-of-service attack on a computer or to exploit a buffer overflow condition allowing the attack to gain control over a server.
- **Denial-of-service attack through the use of MTU reduction.** An attacker on an unauthenticated computer can reduce the MTU for IPsec traffic (and unsecured TCP or UDP traffic) to as low as 68 bytes. This attack is sophisticated because the attacker must capture an IPsec packet or another packet type that is sent by a computer (such as CORPSRV), use that packet to construct an ICMP Destination Unreachable message so that it contains an excessively low PMTU value, and then send the ICMP message back to CORPSRV. If the attack is successful, each packet in a TCP connection between a client IP address and CORPSRV can contain only a very small amount of data, which might result in the loss of connectivity. The connectivity would be affected only with the IP address of the client that was communicating with CORPSRV.
- **Man-in-the-middle attack through the use of an ICMP Redirect message.** Attackers can use ICMP Redirect messages to actively interfere with communications between CORPSRV and specific clients. This attack is sophisticated because the attacker must capture an IPsec packet or another packet type that is sent by CORPSRV, use that packet to construct an ICMP Redirect message, and then send the ICMP message back to CORPSRV. If the attack is successful, it enables more sophisticated man-in-the-middle attacks against traffic that is not secured by IPsec. By IETF design, IKE and IPsec traffic is resistant to most man-in-the-middle attacks. However, an attacker can easily use a man-in-the-middle attack to block traffic or to modify any IKE or IPsec packet to cause a denial of service and subsequent loss of connectivity. Based on the way in which routing tables are maintained in Windows, an ICMP redirect attack would be effective for 10 minutes before the route is deleted. After the route is deleted, the attacker must mount another ICMP Redirect attack to regain a man-in-the-middle position and actively interfere with communications between CORPSRV and specific clients.

In Rule 7 of the example CORPSRV policy, a one-way inbound filter allows unsecured ICMP traffic to be received from any IP address. This filter allows TCP PMTU discovery to be performed when the more secure form of CORPSRV Policy Rule 5 is used (that is, when you clear the **Accept unsecured traffic, but always respond using IPsec** check box, in the filter action for Rule 5, so that inbound passthrough is not allowed). Because IPsec does not support the filtering of different types of ICMP protocol messages, Rule 7 either permits all ICMP messages or none.

Note that the only ICMP-based attacks made possible by Rule 7 involve the receipt of unsecured ICMP traffic. Outbound ICMP traffic matches the more general filter in Rule 5 and therefore is IPsec-secured. If your network environment permits, it is recommended that you disable Rule 7 to prevent the receipt of unsecured ICMP traffic. Because the filter that permits inbound

---

unsecured ICMP traffic is not applied, inbound ICMP traffic matches the filter that is associated with Rule 5 instead. The filter for Rule 5 matches traffic over any protocol, so if you disable Rule 7, make sure that you use the more secure form of Rule 5 so that inbound passthrough (unsecured traffic) is not allowed.

### Considerations for IKE and IPSec

When an IPSec policy is assigned to help secure communication between two computers, IKE sends and receives traffic on UDP port 500. If both computers support the specifications outlined in IETF IPSec NAT-T Specification Draft, version 2, and network address translation is taking place, IKE also sends and receives traffic on UDP port 4500. IKE opens these ports even if your IPSec policy uses only permit and block filters. If you configure an IPSec policy to permit outbound ICMP messages (for example, for TCP PMTU discovery), or if IKE replies to an attacker's request to negotiate security, then the attacker might be able to determine whether the computer can receive inbound IKE traffic. As a security protocol, IKE can resist many types of attacks. However, denial-of-service attacks against the IKE protocol impacts computer performance by generating high CPU utilization. When the denial-of-service attack is stopped, CPU utilization typically returns to normal quickly.

When IKE receives a request to negotiate security, it determines whether the request is properly formatted, whether the IPSec policy filters allow a response to be sent to the source IP address of the computer that initiated the IKE request, and whether the request parameters match the IPSec policy settings. If the request parameters do not match the IPSec policy settings, IKE ignores the request and does not send a response. If you create a mirrored filter with a source address of **My IP Address** and a destination address of **Any IP Address**, IKE replies to traffic that is sent from any source IP address. Also, if you enable the default response rule, IKE replies to properly formatted requests.

The core of the security that IPSec provides for communications depends on successful mutual authentication. However, as mentioned earlier, a skilled Internet attacker might be able to construct specific security negotiation requests to obtain identity information or other information about a computer. To assess whether identity information or other information about a computer is exposed in the IKE negotiation before successful mutual authentication is completed, you can use the version of Network Monitor that is provided in Windows Server 2003. By using Network Monitor, you can analyze the information that is included in the IKE negotiation. For information about Network Monitor, see "[Viewing IPSec and Other Network Communication](#)," earlier in this paper.

AH and ESP are also security protocols that can resist many types of attacks. For example, these protocols count and quickly discard invalid packets. However, if you configure an IPSec policy to permit outbound ICMP messages, an attacker might be able to determine whether the computer supports receiving IPSec protocol traffic and mount a denial-of-service attack. To assess whether attacks are being mounted against IPSec protocols, you can enable auditing for specific Audit Policy security settings so that you can view IPSec events in the System log. For more information about enabling auditing for IPSec events and viewing IPSec events, see "[Viewing IPSec Events](#)," earlier in this paper.

---

## Resources

This section provides a summary of the links that are referenced in this paper and additional links to relevant resources for Windows IPSec.

### Network Threats and Attacks

- For information about risks, vulnerabilities, and attack techniques for specific protocols, operating systems, and services, see the featured title “Hacking Exposed: Network Security Secrets & Solutions,” in [Authored Books](#), at <http://go.microsoft.com/fwlink/?LinkId=19317>.
- For information about the risks of network access (without IPSec) to applications using TCP and UDP ports, see [Sockets and Services from a Security Point of View](#), at <http://go.microsoft.com/fwlink/?LinkId=18756>.
- For information about the risks to network communications from network devices and other computers on the network, see [Topology Security](#), at <http://go.microsoft.com/fwlink/?LinkId=18760>.
- For information about the threat of physical intrusion and other threats to security, see [The Ten Immutable Laws of Security](#), at <http://go.microsoft.com/fwlink/?LinkId=18751>.
- For information about authentication vulnerabilities with preshared keys, see [Authentication vulnerabilities in IKE and Xauth with Weak Pre-Shared Secrets](#), at <http://go.microsoft.com/fwlink/?LinkId=18769>.
- For more information about the RPC endpoint mapper security issue that was found and fixed, see [Microsoft Security Bulletin MS03-010: Flaw in RPC Endpoint Mapper Could Allow Denial of Service Attacks \(331953\)](#), at <http://go.microsoft.com/fwlink/?LinkId=18754>.
- For information about ICMP attacks, see [ICMP Attacks Illustrated](#), at <http://go.microsoft.com/fwlink/?LinkId=18765>

### Windows 2000 IPSec (General)

- For information about Windows 2000 IPSec and other networking and communications features of Windows 2000, see [IPSec](#), at <http://go.microsoft.com/fwlink/?LinkId=13283>.
- For information about Windows 2000 IPSec, see [Internet Protocol Security](#), at <http://go.microsoft.com/fwlink/?LinkId=16465>.
- For information about diagnostics and troubleshooting procedures for Windows 2000 IPSec, see [Step-by-Step Guide to Internet Protocol Security \(IPSec\)](#), at <http://go.microsoft.com/fwlink/?LinkId=269>.
- For information about how to configure Windows 2000 IPSec blocking filters to secure servers, see “Chapter 7: Hardening Specific Server Roles,” in [Microsoft Solution for Securing Windows 2000 Server](#), at <http://go.microsoft.com/fwlink/?LinkId=15394>.

---

## Windows Server 2003 IPSec (General)

- For Help topics about Windows Server 2003 IPSec, see Help and Support Center for Windows Server 2003.
- For deployment information about Windows Server 2003 IPSec, see Chapter 6, “Deploying IPSec,” in *Deploying Network Services*, in the [Windows Server 2003 Deployment Kit](http://go.microsoft.com/fwlink/?LinkId=8195), on the Web at <http://go.microsoft.com/fwlink/?LinkId=8195>.
- For information about using Windows Server 2003 IPSec in specific scenarios, see, “[Special IPSec Considerations](http://go.microsoft.com/fwlink/?LinkId=19223)” at <http://go.microsoft.com/fwlink/?LinkId=19223>.
- For information about the Netsh commands for IPSec, see [Netsh commands for Internet Protocol security \(IPSec\)](http://go.microsoft.com/fwlink/?LinkId=19315), at <http://go.microsoft.com/fwlink/?LinkId=19315>.
- For information about Ipseccmd.exe, see [Ipseccmd](http://go.microsoft.com/fwlink/?LinkId=18755), at <http://go.microsoft.com/fwlink/?LinkId=18755>.

## Security for Windows Active Directory (includes information about IPSec)

- For information about how to use IPSec to help secure Windows 2000 Active Directory communication on opposite sides of a firewall, see “Active Directory in Networks Segmented by Firewalls,” available at the [Download Center](http://go.microsoft.com/fwlink/?LinkId=3457), at <http://go.microsoft.com/fwlink/?LinkId=3457>.
- For information about security considerations for using Active Directory in perimeter networks, see:
  - Chapter 5, “Security Design,” in the [Reference Architecture Guide: Internet Data Center](http://go.microsoft.com/fwlink/?LinkId=16468), at <http://go.microsoft.com/fwlink/?LinkId=16468>.
  - Chapter 8, “Directory Services,” in the [Reference Architecture Guide: Enterprise Data Center](http://go.microsoft.com/fwlink/?LinkId=16463), at <http://go.microsoft.com/fwlink/?LinkId=16463>.

## Security for Windows (General)

- For information about how to enhance security for Windows 2000, see [Microsoft Windows 2000 Security Hardening Guide](http://go.microsoft.com/fwlink/?LinkId=18759), at <http://go.microsoft.com/fwlink/?LinkId=18759>.
- For information about how to enhance security for Windows 2000 Server, see [Microsoft Solution for Securing Windows 2000 Server](http://go.microsoft.com/fwlink/?LinkId=18758), at <http://go.microsoft.com/fwlink/?LinkId=18758>.
- For information about procedures that enhance server security, see [Hardening Systems and Servers Checklists and Guides](http://go.microsoft.com/fwlink/?LinkId=19321), at <http://go.microsoft.com/fwlink/?LinkId=19321>.
- For information about security bulletins, see [Security Bulletins](http://go.microsoft.com/fwlink/?LinkId=19322), at <http://go.microsoft.com/fwlink/?LinkId=19322>.
- For information about how to formulate security plans, see [Best Practices for Enterprise Security](http://go.microsoft.com/fwlink/?LinkId=19318) at <http://go.microsoft.com/fwlink/?LinkId=19318>.
- For information about how to configure Windows security settings, see [Threats and Countermeasures Guide](http://go.microsoft.com/fwlink/?LinkId=19618), at <http://go.microsoft.com/fwlink/?LinkId=19618>.

- 
- For information about Windows security technologies, see:
    - [Windows 2000 Security Services](http://go.microsoft.com/fwlink/?LinkId=19319), at <http://go.microsoft.com/fwlink/?LinkId=19319>
    - Windows Server 2003 [Security Services](http://go.microsoft.com/fwlink/?LinkId=19619), at <http://go.microsoft.com/fwlink/?LinkId=19619>
    - [Deploying and Supporting a PKI at Microsoft](http://go.microsoft.com/fwlink/?LinkId=19620), at <http://go.microsoft.com/fwlink/?LinkId=19620>
    - [Step-by-Step Guide to Administering Certificate Services](http://go.microsoft.com/fwlink/?LinkId=326), at <http://go.microsoft.com/fwlink/?LinkId=326>.
    - [Troubleshooting Certificate Status and Revocation](http://go.microsoft.com/fwlink/?LinkId=18753), at <http://go.microsoft.com/fwlink/?LinkId=18753>.

### **Networking for Windows 2000**

- For information about TCP/IP, see [Microsoft Windows 2000 TCP/IP Implementation Details](http://go.microsoft.com/fwlink/?LinkId=16467), at <http://go.microsoft.com/fwlink/?LinkId=16467>.
- For information about how to enhance the security of TCP/IP, see [Security Considerations for Network Attacks](http://go.microsoft.com/fwlink/?LinkId=19320), at <http://go.microsoft.com/fwlink/?LinkId=19320>.
- For information about the network services provided with Windows 2000, see the [TCP/IP Core Networking Guide](http://go.microsoft.com/fwlink/?LinkId=18747), in the *Windows 2000 Resource Kits*, at <http://go.microsoft.com/fwlink/?LinkId=18747>.
- For information about how to install the version of Network Monitor that is available in Windows 2000, see [To Install Network Monitor](http://go.microsoft.com/fwlink/?LinkId=18749), at <http://go.microsoft.com/fwlink/?LinkId=18749>.
- For information about Netdiag.exe, see [Netdiag.exe: Network Connectivity Tester](http://go.microsoft.com/fwlink/?LinkId=19316), at <http://go.microsoft.com/fwlink/?LinkId=19316>.
- For information about SMS, see the [Systems Management Server Web site](http://go.microsoft.com/fwlink/?LinkId=299), at <http://go.microsoft.com/fwlink/?LinkId=299>.

### **Microsoft Knowledge Base Articles**

The following Knowledge Base articles are referenced in this paper, relevant to the subject of this paper, or both. To find Knowledge Base articles, see the [Microsoft Knowledge Base](http://go.microsoft.com/fwlink/?LinkId=16462), at <http://go.microsoft.com/fwlink/?LinkId=16462>.

- Knowledge Base article 150543, “Windows NT, Windows 2000, Terminal Server, and Microsoft Exchange Services Use TCP/IP Ports”
- Knowledge Base article 233256, “How to Enable IPSec Traffic Through a Firewall”
- Knowledge Base article 254728, “IPSec Does Not Secure Kerberos Traffic Between Domain Controllers”
- Knowledge Base article 254949, “Client-to-Domain Controller and Domain Controller-to-Domain Controller IPSec Support”

- 
- Knowledge Base article 289241, “A List of the Windows Server Domain Controller Default Ports”
  - Knowledge Base article 299977, “Direct Hosting of SMB over TCP/IP”
  - Knowledge Base article 315669, “HOW TO: Harden the TCP/IP Stack Against Denial of Service Attacks in Windows 2000”
  - Knowledge Base article 324380, “MS02-051: Cryptographic Flaw in RDP Protocol Can Cause Information Disclosure,”
  - Knowledge Base article 329194, “IPSec Policy Permissions in Windows 2000 and Windows Server 2003”
  - Knowledge Base article 810207, “IPSec Default Exemptions Are Removed in Windows Server 2003”
  - Knowledge Base article 811832, “IPSec Default Exemptions Can Be Used to Bypass IPSec Protection in Some Scenarios”
  - Knowledge Base article 813878, “How to Block Specific Network Protocols and Ports by Using IPSec”
  - Knowledge Base article 816521, “HOW TO: Use IPSec Policy to Secure Terminal Services Communications in Windows Server 2003”
  - Knowledge Base article 818043, “L2TP/IPSec NAT-T Update for Windows XP and Windows 2000”

### Microsoft Downloads

- To download the Windows 2000 High Encryption Pack, see [Windows 2000 High Encryption Pack](http://go.microsoft.com/fwlink/?LinkId=7272), at <http://go.microsoft.com/fwlink/?LinkId=7272>.  

If you are running Windows 2000 or Windows 2000 Service Pack 1, you must install the Windows 2000 High Encryption pack to use 3DES. If you are using Windows Service Pack 2 or later, you do not need to install the Windows 2000 High Encryption Pack to use 3DES.
- To download the latest version of Ipsecpol.exe, see [Ipsecpol.exe: Internet Protocol Security Policies Tool](http://go.microsoft.com/fwlink/?LinkId=16466), at <http://go.microsoft.com/fwlink/?LinkId=16466>.
- To download the latest version of Rpcdump.exe, see [Rpcdump.exe: RPC Dump](http://go.microsoft.com/fwlink/?LinkId=19324), at <http://go.microsoft.com/fwlink/?LinkId=19324>.

For more information about RPC, see [RPC Components](http://go.microsoft.com/fwlink/?LinkId=19222), at <http://go.microsoft.com/fwlink/?LinkId=19222> and [The RPC Model](http://go.microsoft.com/fwlink/?LinkId=18762) at <http://go.microsoft.com/fwlink/?LinkId=18762>.

### IPSec Hardware Offload Adapters and Hardware Compatibility

- For information about the benefits of using IPSec hardware offload adapters, see [Intel PRO/100S Network Adapter, IPSec Offload Performance and Comparison](http://go.microsoft.com/fwlink/?LinkId=16469), at <http://go.microsoft.com/fwlink/?LinkId=16469>.

- 
- For information about hardware that is certified for Windows 2000, see [Search for Compatible Hardware Devices](http://go.microsoft.com/fwlink/?LinkId=3787), at <http://go.microsoft.com/fwlink/?LinkId=3787>.
  - For information about network adapters that are compatible with Windows XP, see [Browse Hardware, Networking and Modems, and LAN Cards](http://go.microsoft.com/fwlink/?LinkId=19934), at <http://go.microsoft.com/fwlink/?LinkId=19934>.
  - For information about network adapters that are compatible with Windows Server 2003, see [Networking and Modems, LAN Cards](http://go.microsoft.com/fwlink/?LinkId=19935), at <http://go.microsoft.com/fwlink/?LinkId=19935>.
  - For information about TCP/IP Task Offload, see [Task Offload](http://go.microsoft.com/fwlink/?LinkId=18763) at <http://go.microsoft.com/fwlink/?LinkId=18763>.
  - For information about the 10/100 MB Ethernet hardware offload network adapters that are available for Windows 2000 IPsec, see the following:
    - The [Intel Web site](http://go.microsoft.com/fwlink/?LinkId=16474), at <http://go.microsoft.com/fwlink/?LinkId=16474>.
    - The [3Com Web site](http://go.microsoft.com/fwlink/?LinkId=16475), at <http://go.microsoft.com/fwlink/?LinkId=16475>.