



realtimepublishers.com<sup>™</sup>

*Tips and Tricks Guide<sup>™</sup> To*

# Securing .NET Server



*Roberta Bragg*

**Note to Reader:** This book presents tips and tricks for eight Windows .NET Server security topics. For ease of use, the questions and their solutions are divided into chapters based on topic, and each question is numbered based on the chapter, including:

- Chapter 1: Understanding and Utilizing PKI in .NET
- Chapter 2: Securing Web Services and Web Servers—the Administrative Perspective
- Chapter 3: Understanding Active Directory Foundations
- Chapter 4: Fulfilling the Promises of Group Policy
- Chapter 5: Administrative Authority
- Chapter 6: Triple A’s—Authentication, Authorization, and Audit
- Chapter 7: Remote Access
- Chapter 8: Security Tools, Mechanisms, and Emerging Issues.

Chapter 1: Understanding and Utilizing PKI in .NET .....	1
Q 1.5: If a user’s private key is destroyed or becomes corrupted can his or her Encrypting File System files be recovered?.....	1
Prepare the CA for Key Recovery .....	1
Prepare EFS Custom Template for Key Archival.....	6
Using the New Certificates and Recovering Keys.....	8
Chapter 2: Securing Web Services and Web Servers—the Administrative Perspective.....	12
Q 2.5: We do not allow users to store data on their hard drives. They are provided a place on a file server. I can protect this area with discretionary access control lists, but how do I protect data during transport from client to file server? .....	12
Preparing and Securing Web Folders for WebDAV.....	12
Using SSL to Transfer Files to Web Folders.....	15
Chapter 3: Understanding Active Directory Foundations .....	19
Q 3.5: Is it possible to place the Active Directory database on a different drive from its logs? ...	19
Chapter 4: Fulfilling the Promises of Group Policy .....	21
Q 4.5: How can I prevent wireless access points from become an unguarded entry point into my network?.....	21
Securing Wireless Communications .....	22
Standard Security Options .....	23
Standard Security Options Plus Fire walling.....	24
Add 802.1x Technology.....	25
Windows .NET Server Wireless Network (IEEE 802.11) Policies .....	26
Chapter 5: Administrative Authority .....	31
Q. 5.5. What exactly can an Enterprise Administrator do? .....	31

Chapter 6: Triple A's—Authentication, Authorization, and Audit .....	35
Q 6.5: Which file activity should be audited and how do I do so? .....	35
Maintaining an Audit Trail .....	35
Monitoring a User's Activity .....	36
Compatibility Resolver .....	36
Log Changes in System Files.....	37
How to Set Up File Auditing .....	37
Set Audit Policy .....	37
Set SACLS on Files and Folders .....	38
SACL Inheritance .....	40
Chapter 7: Remote Access .....	41
Q 7.5: I set up a Windows 2000 virtual private network (VPN) for use by our salesmen to connect to the corporate LAN. It worked fine at first, but we had a security review, and the experts advised us to change the VPN protocol from Point-to-Point Tunneling Protocol (PPTP) to Layer 2 Tunneling Protocol (L2TP)/IPSec and change our authentication method to certificates. It seems to work in my test lab, but when I put it into production, I cannot get it to work. In addition, we must be accessible to Windows 98 clients. Will upgrading our Routing and Remote Access Service (RRAS) server to Windows .NET solve this problem? .....	41
VPN Design Issues for L2TP/IPSec .....	42
VPN Protocols Choices.....	44
PPTP .....	45
L2TP/IPSec .....	45
L2TP over IPSec and NAT—NAT-Traversal .....	46
Chapter 8: Security Tools, Mechanisms, and Emerging Issues.....	48
Q 8.5: I'd like to prevent clients on my network from using or receiving Telnet commands and from running a Web server. Can this be done?.....	48
IPSec Blocking Policy Basics.....	48
Creating and Testing the Policy .....	48

## Copyright Statement

© 2002 Realtimedpublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimedpublishers.com, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimedpublishers.com, Inc or its web site sponsors. In no event shall Realtimedpublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimedpublishers.com, please contact us via e-mail at [info@realtimedpublishers.com](mailto:info@realtimedpublishers.com).

## **Chapter 1: Understanding and Utilizing PKI in .NET**

### **Q 1.5: If a user's private key is destroyed or becomes corrupted can his or her Encrypting File System files be recovered?**

**A:** The mechanism for recovering files in Windows 2000 (Win2K) is to use a data recovery agent. Although Windows .NET still offers this option, it also lets you archive a user's Encrypting File System (EFS) private key and authorize a key recovery agent. To do either requires some planning.

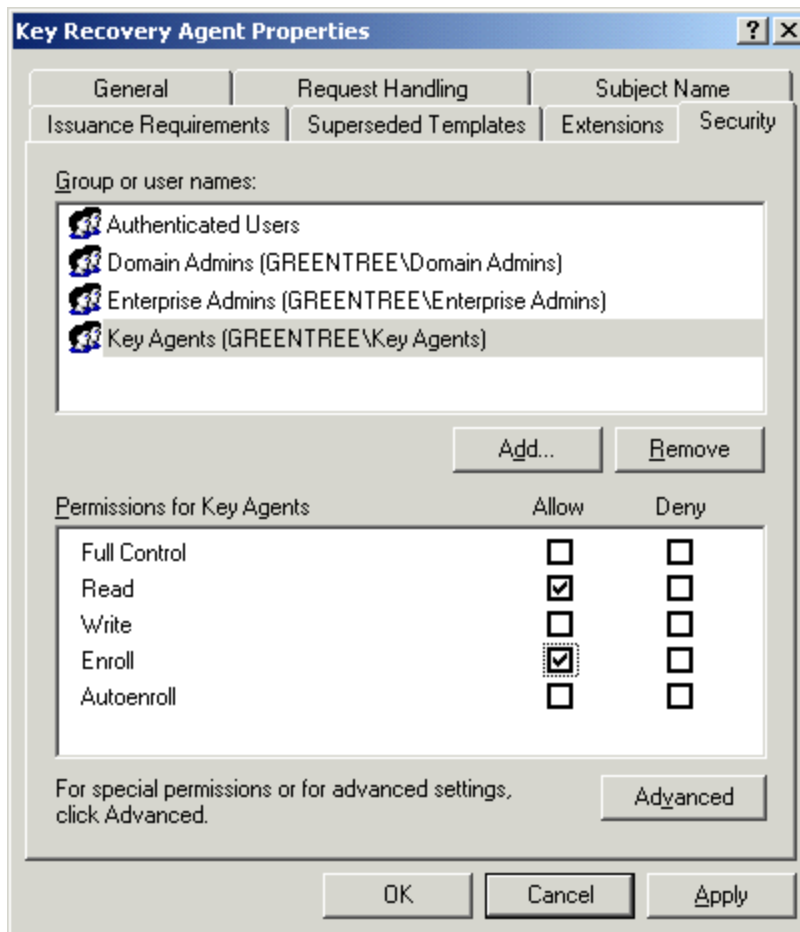
Key archival is the best way to go, but requires a 100 percent Windows .NET enterprise. Key archival is superior to file recovery because there is less risk of data exposure. In file recovery, the file might be exposed because it must be decrypted by the recovery agent, then later encrypted by the owner. Although the recovery agent does not have to open the file to decrypt it, the opportunity for data exposure exists. If a file recovery station is used, and it should be, the plain text file exists on the file recovery station and could be transported in clear text to the owner. With key recovery, the files remain decrypted and in their proper location. The original key of the user is recovered by the key recovery agent and made available to the original owner. No files are moved or decrypted in the process.

You must, however, configure your enterprise for key archival, or key recovery is not possible. Before users are allowed to encrypt files, establish the formal key recovery policy and configure EFS templates, assign key recovery agents, and configure your Windows .NET Certificate Authority (CA). The following conditions must be true: The enterprise must be 100 percent Windows .NET; no files have been encrypted yet, or users are ready to decrypt and re-encrypt with new EFS certificates; the Enterprise CA must be installed on a Windows .NET Enterprise Edition server.

#### ***Prepare the CA for Key Recovery***

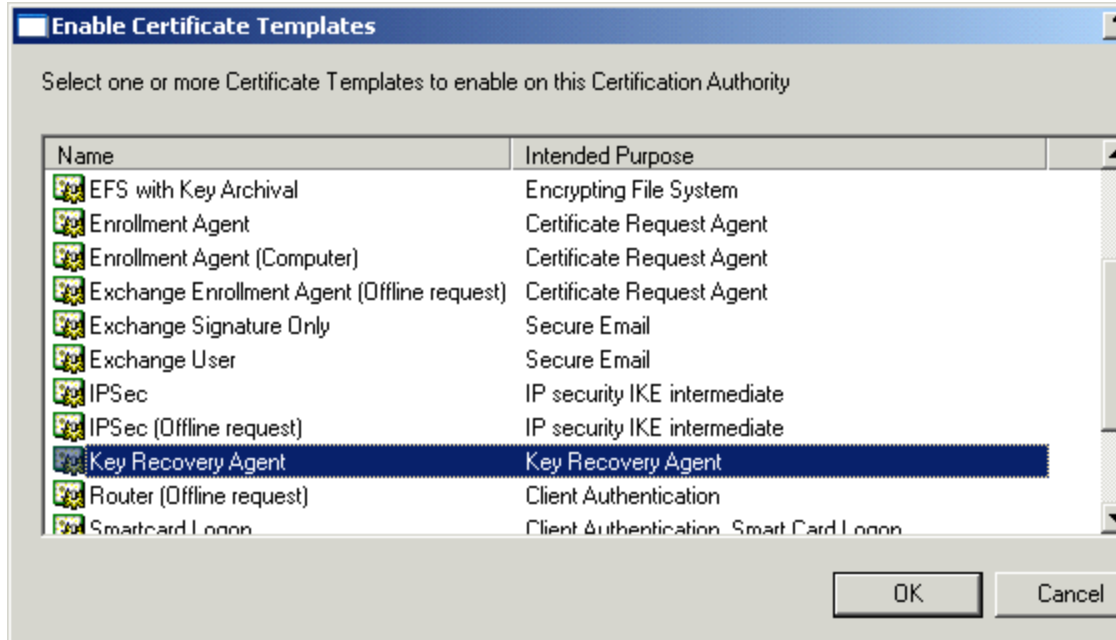
The CA is not configured for key recovery by default. To prepare it, you must perform several steps before you can issue certificates that will be able to archive their private key. You don't have to first decide who should be the key recovery agent; the account or accounts used for key archival should not be accounts used for ordinary user or administrator duties. These accounts should be created for this purpose. Although they should be assigned to specific individuals when key recovery is necessary, there is no reason to assign them before setting up key recovery.

The process of preparing a user account to be a key recovery agent is straightforward. First, a group is created whose members will be authorized to recover keys. This group must be given the right to enroll as a Key Recovery Agent and the Manage CA and Issue and Manage Certificate permission on the CA. To do so, open the Start menu, click Administrative Tools, and select the Certification Authority console. Permissions on certificates can be managed by right-clicking the Certification Authority\*name of CA* container and selecting Manage. In this window, right-click the Key Recovery Agent certificate, and select Properties. On the Security tab of the certificate (see Figure 1.20), add the group you have created (Key Agents created in this domain) and assign the Enroll permission. If your policy calls for it, you should also remove the Enroll permission from the Domain Admins and Enterprise Admins groups at this time. Create user accounts in Active Directory (AD) and add them to the key recovery agents group.



**Figure 1.20: Give the key recovery group permission to obtain the key recovery certificate.**

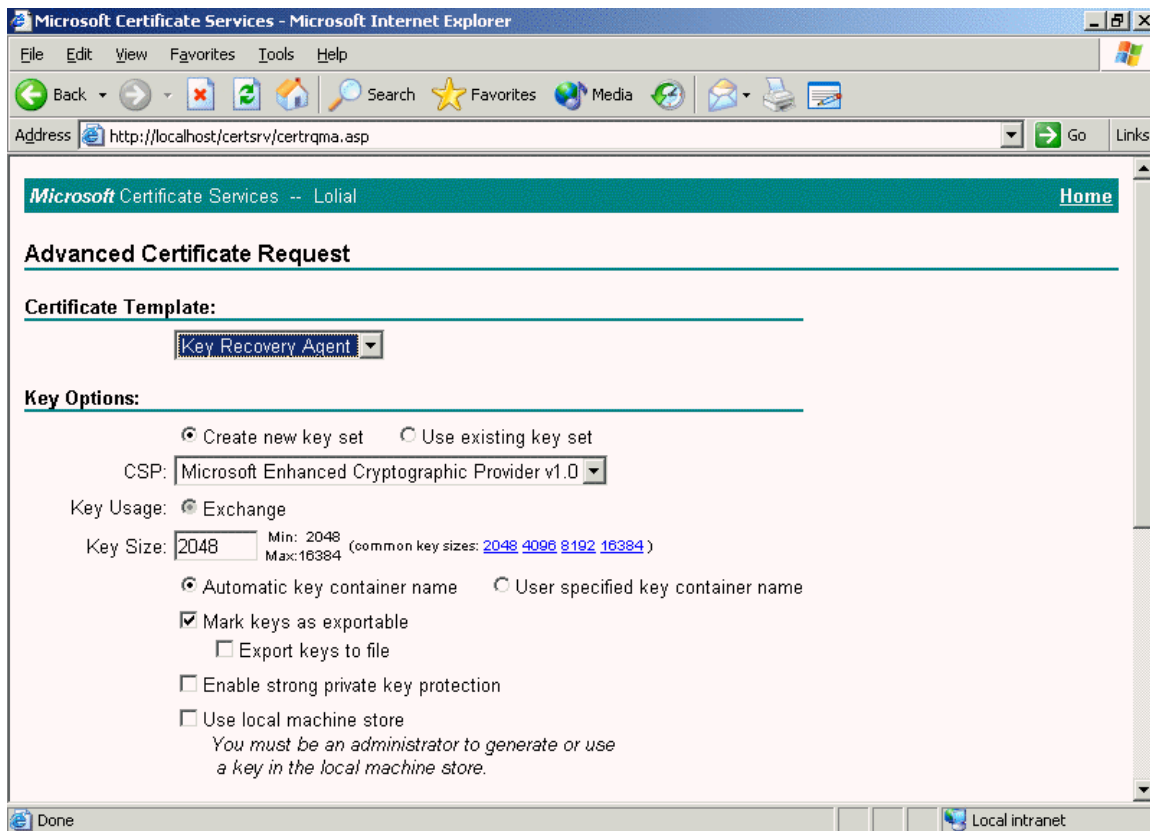
Next, the Key Recovery Agent certificate must be enabled for the CA. To do so, right-click Certificate Templates, select New, then Certificate Template to Issue. In the Enable Certificate Template dialog box, select the Key Recovery Agent (see Figure 1.21), and click OK.



**Figure 1.21: Enable the Key Recovery Agent certificate for the CA.**

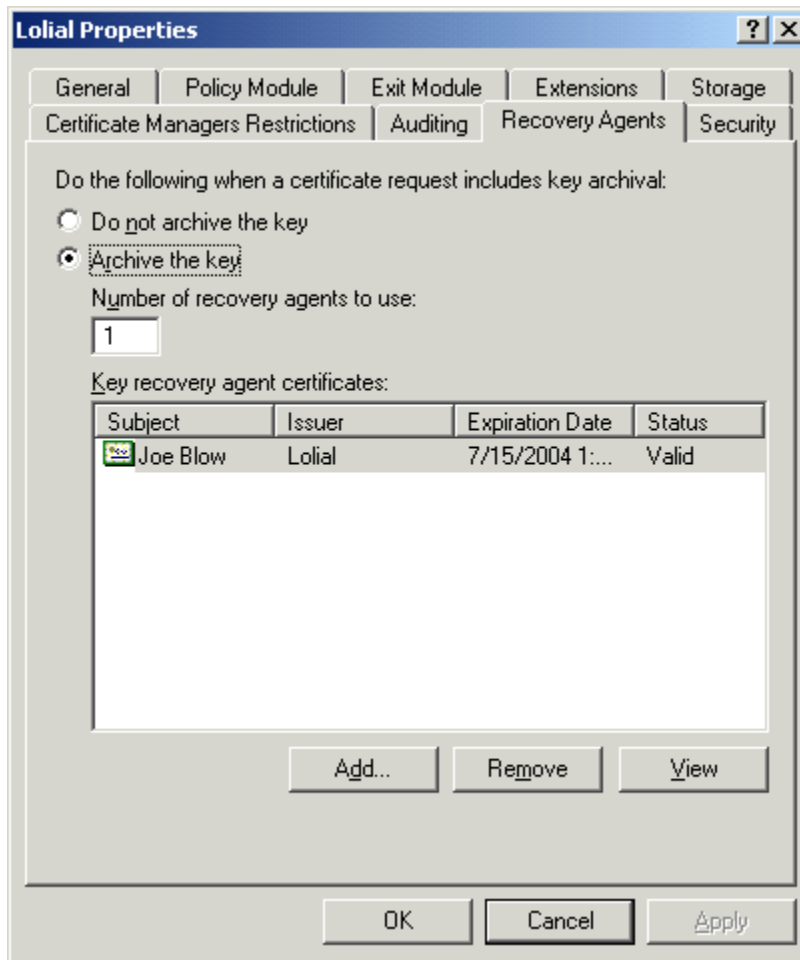
After the certificate is available, each user account, which will be used for key recovery, must request its key recovery agent certificate. If users have not been assigned, an administrator can log on as one of the approved key recovery agent accounts to obtain the certificate. An administrator should not retain control over this account. To obtain the certificate, log on using the selected Key Recovery Agent account. For the following example, we'll use the account Joe Blow.

Use Internet Explorer (IE) to browse to the CA Web location—<http://certserver/certsrv> (where certserver is the name of your CA). Click the Request a Certificate task, click the Advanced Certificate request link, click *Request and Submit a request to this CA*, and in the Certificate Template drop-down box, select Key Recovery Agent (see Figure 1.22). Next, click Submit. You will be informed that the request must be processed by an administrator and to return for your certificate later. Use Run as to open the Certification Authority console. Open the Pending Requests container. Locate the recently requested certificate, right-click it, and select Issue. Close the console, thus returning to your IE session as the agent, and return to the CA Web location (for this example, <http://certserver/certsrv>). Select *View the Status of a pending certificate request*, click on the certificate, click *Install this certificate*, and repeat this process for each Key Recovery Agent.



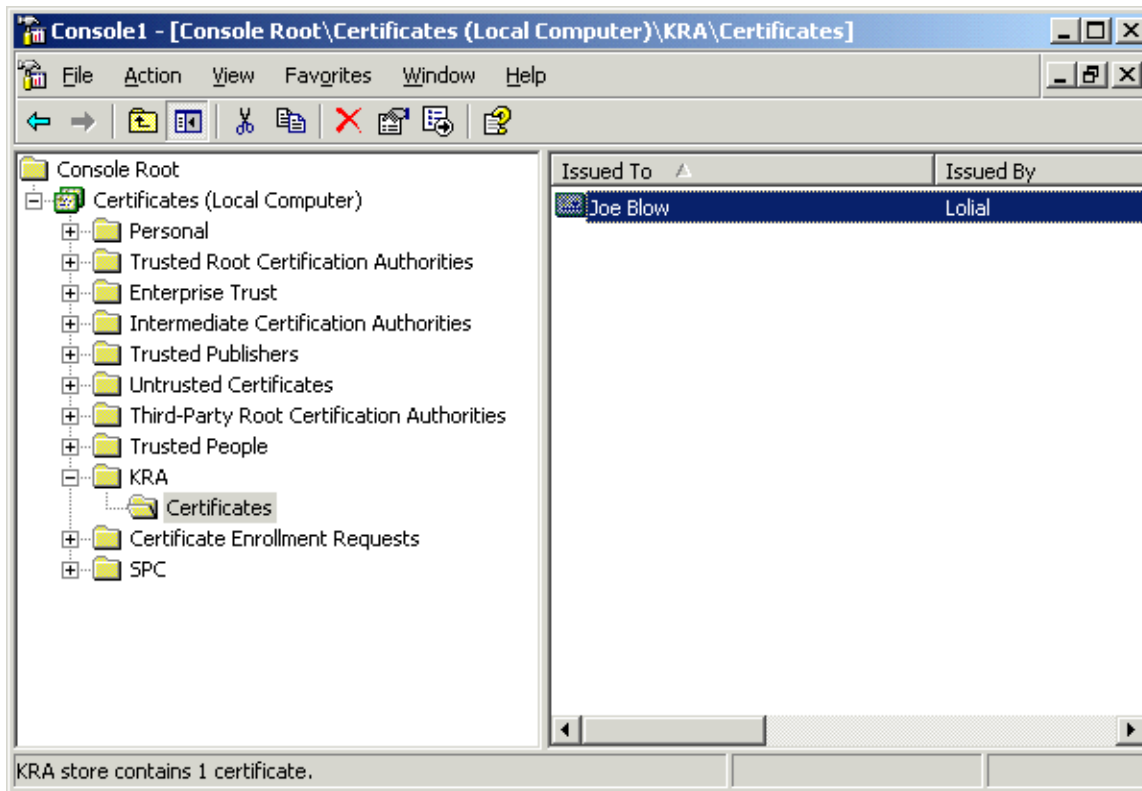
**Figure 1.22: Request a Key Recovery Agent Certificate.**

Before keys can be archived, the certificates issued for Key Recovery Agents must be assigned to the CA. To do so, right-click the name of the CA in the console, and select Properties. On the Recovery Agents tab, select *Archive the Key* under *Do the following when a certificate request includes key archival*. Enter the number of recovery agents to use in the text box provided. For each recovery agent, add its certificate by clicking Add and selecting it from the provided list, then clicking OK. A certificate icon with a red circle and a white cross over it will appear in the Certificates window. The status listed is Not Loaded. When you have added them all, click Apply, and click OK when prompted to restart the service. After the service has restarted (you might need to close and open the properties page), the red circle with the white cross should be gone and the status should be Valid (see Figure 1.23).



**Figure 1.23: Assigning the Key Recovery Agent Certificate to the CA.**

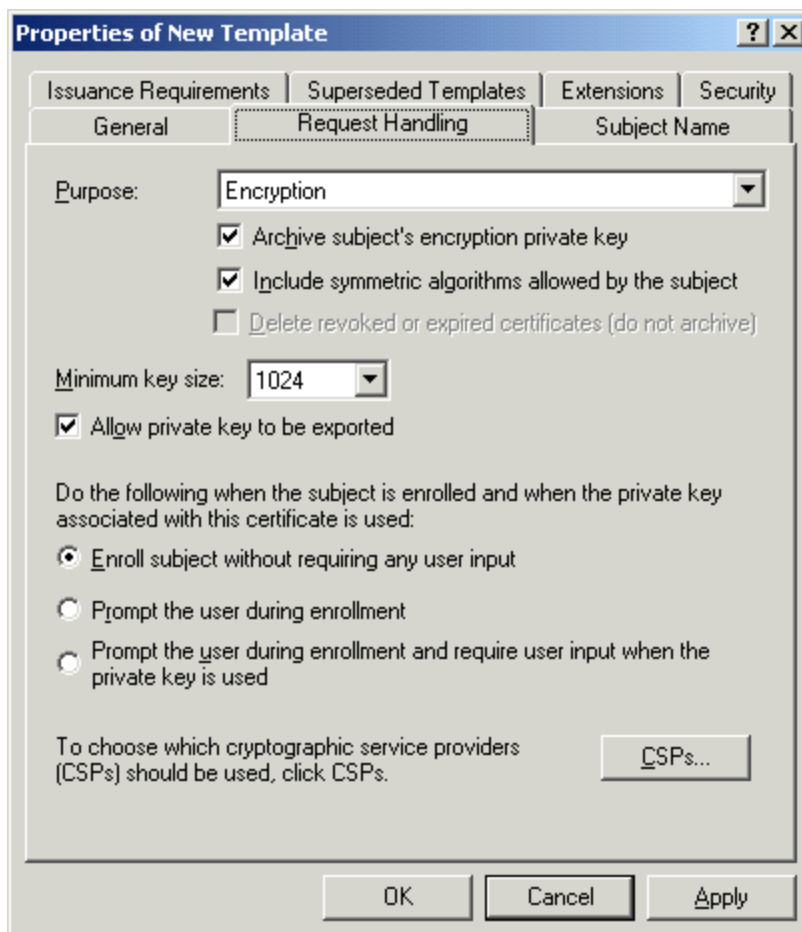
Another check of the status of Key Recovery Agent certificates can be made by opening the certificates console for the CA computer. The KRA certificate store, when inspected, should contain valid certificates for the Key Recovery Agents assigned (see Figure 1.24).



**Figure 1.24: Checking that Key Recovery Agent certificates have been assigned to the CA.**

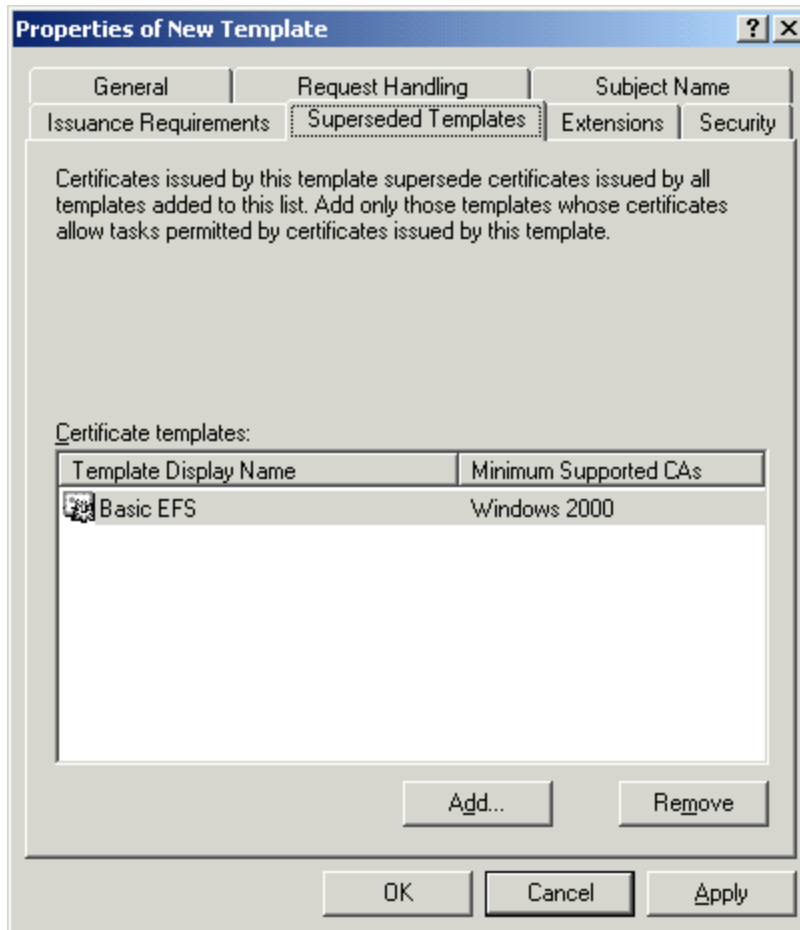
### **Prepare EFS Custom Template for Key Archival**

Now that the CA is prepared, it's OK to prepare the EFS certificate for key archival. To do so, you must create a custom template. This process is described in Question 1.1. In short, a custom template is created by duplicating an existing template, then modifying its properties. After duplication of the EFS Basic template, one change must be made for key archival, another change is optional. As Figure 1.25 shows, on the Request Handling properties page, select the *Archive subject's encryption private key* check box.



**Figure 1.25: Configuring EFS custom template for key archival.**

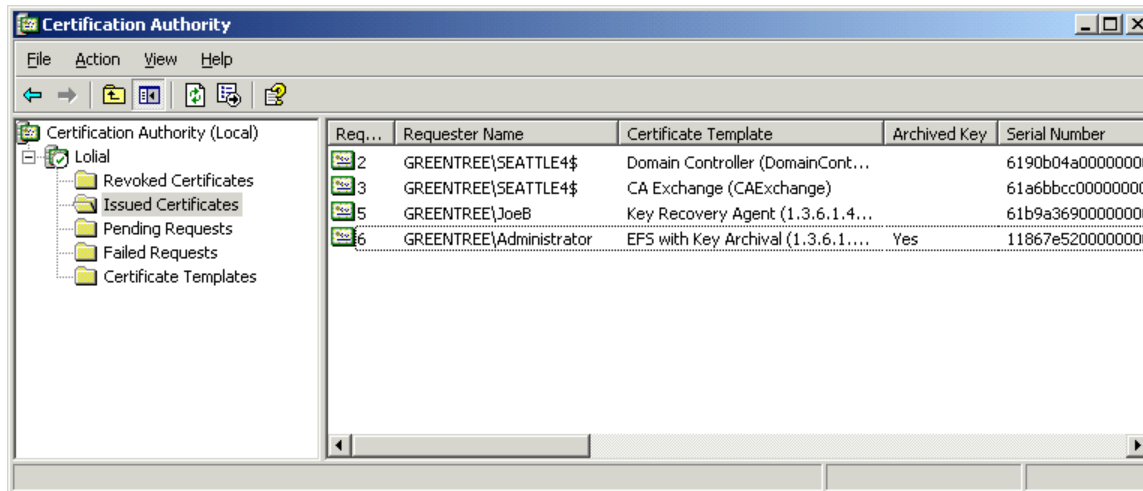
If EFS has been in use in your enterprise, it is a good idea to supersede the basic EFS template with the new one. Otherwise, until users request a new EFS certificate, their files will continue to use the basic template, which does not offer key archival. You can automatically require that the old template be superseded by entering its information on the Superseded Templates property page. Simply click Add, and select it from the list, then click OK. Figure 1.26 shows the page after this entry. When the new template is complete, it can be added to the CA in the manner previously described for the Key Recovery Agent certificate.



**Figure 1.26: Requiring that the new certificate supercede the old.**

### **Using the New Certificates and Recovering Keys**

Now that the certificates are ready and a Key Recovery Agent has been assigned to the CA, users must request the new certificates. If you configured the new EFS certificate to supercede an older EFS Basic certificate, the user will automatically receive a new certificate; however, if the user has been using a self-signed certificate, the user must replace it by requesting a recovery certificate. This action can be done from the Certificates console or from the certsrv Web page. To verify that the issued EFS certificates have archived their keys or to validate that an archived key exists before attempting recovery, you can view its status in the Certification Authority console. You might need to add the column Archived Key; you can do so from the View menu by selecting Add/Remove Columns while the Issued Certificates container is open. After the column has been added, it will display a Yes to tell you which certificates have valid archived private keys (see Figure 1.27).



**Figure 1.27: The Archived Key column.**

Users encrypt and decrypt files in the ordinary manner. However, if a user's private key is lost or becomes corrupt, you now have a way to retrieve the key. To do so, as Administrator, view the certificate of the user by double-clicking it in the Issued Certificates container of the CA. Select the Details page, and write down the certificate's serial number. (The serial number of the certificate is the serial number of the private key, and is a 20-digit hexadecimal number.) Close the Certification Authority console, and give the Key Recovery Agent the serial number. The Key Recovery Agent logs on. (From here, the Key Recovery Agent is doing the recovery. The Key Recovery Agent does not need to be a member of any administrative group.) At the C:\ command prompt, enter the following command to recover the key (see Figure 1.28):

```
certutil -getkey serialnumber outputblob
```

```

C:\>certutil -GetKey 11867e5200000000000006 outputblob
Querying seattle4.greentree.local\Lolial.....

"seattle4.greentree.local\Lolial"
Serial Number: 11867e5200000000000006
Subject: CN=Administrator, CN=Users, DC=greentree, DC=local
UPN:Administrator@greentree.local
NotBefore: 7/16/2002 1:40 PM
NotAfter: 7/16/2003 1:40 PM
Template: EFSwithKeyArchival, EFS with Key Archival
Version: 3
Cert Hash(sha1): 25 b0 20 8c de 37 ef b8 c3 de f8 f6 ae 38 b8 5c 13 f2 90 55

Recipient Info[0]:
Serial Number: 61b9a36900000000000005
Issuer: CN=Lolial, DC=greentree, DC=local
Subject: CN=Joe Blow, CN=Users, DC=greentree, DC=local
Cert Hash(sha1): 9c d8 04 66 5b 1f 02 e4 b2 c8 f9 e9 c4 1e 20 65 4c b0 3e 52
CertUtil: -GetKey command completed successfully.

C:\>whoami
greentree\joeb

C:\>_

```

**Figure 1.28: Key recovery.**

Enter

```
Dir outputblob
```

to determine whether the file was produced. If it was not, it might be because you have entered the serial number incorrectly. The outputblob file is placed on the hard drive. To retrieve the key and create a certificate file that the user can import to his or her certificate store, issue the following command (see Figure 1.29):

```
certutil -recoverkey outputblob username.pfx
```

When prompted, provide a password and confirm it. The pfx file and the password should be delivered to the user in a secure manner. The certificate and private key can then be imported into the user's personal certificate store.

```

C:\Documents and Settings\Administrator\Start Menu\Programs\Accessories\Command Prompt.lnk
C:\>certutil -recoverkey outputblob admin.pfx
----- CERT_CHAIN_CONTEXT -----
ChainContext.dwInfoStatus = CERT_TRUST_HAS_PREFERRED_ISSUER (0x100)
CertContext.dwRevocationFreshnessTime: 16 Hours, 9 Minutes, 42 Seconds

SimpleChain.dwInfoStatus = CERT_TRUST_HAS_PREFERRED_ISSUER (0x100)
SimpleChain.dwRevocationFreshnessTime: 16 Hours, 9 Minutes, 42 Seconds

CertContext[0][0]: dwInfoStatus=10c dwErrorStatus=0
Issuer: CN=Lolial, DC=greentree, DC=local
Subject: CN=Lolial, DC=greentree, DC=local
Serial: 2a2b85df55170fbb416b01124b8cc8ec
Template: CA
d8 e9 73 d4 50 10 27 a9 fb 53 c3 e0 19 0e 2e 0e 45 1f da d9
Element.dwInfoStatus = CERT_TRUST_HAS_NAME_MATCH_ISSUER (0x4)
Element.dwInfoStatus = CERT_TRUST_IS_SELF_SIGNED (0x8)
Element.dwInfoStatus = CERT_TRUST_HAS_PREFERRED_ISSUER (0x100)
CRL 1:
Issuer: CN=Lolial, DC=greentree, DC=local
18 02 69 7c 16 92 6c e4 2b 89 9f af 0a 5d c6 c0 37 6e 52 57
Delta CRL 1:
Issuer: CN=Lolial, DC=greentree, DC=local
f1 96 a2 07 49 81 21 f6 e5 17 54 de f3 3f b8 a0 bc 68 33 d9

Exclude leaf cert:
65 8e 69 31 58 7e e6 bd c9 7e 54 0f b8 4e 4e 4e ef 58 20 d7
Full chain:
3f 9b 11 31 a2 66 f0 c3 b0 c5 6f 16 b6 f6 22 c1 97 a9 fd 1f

Verified Issuance Policies: All
Verified Application Policies: All


Computed Hash: d5 9c b1 84 a0 3a e4 07 ee 6f d1 1c d1 42 00 9d f9 6e 9c 14
Decrypted PKCS7 Message Content

User Certificate:
Serial Number: 11867e5200000000000006
Issuer: CN=Lolial, DC=greentree, DC=local
Subject: CN=Administrator, CN=Users, DC=greentree, DC=local
Cert Hash(sha1): 25 b0 20 8c de 37 ef b8 c3 de f8 f6 ae 38 b8 5c 13 f2 90 55

Enter new password:
Confirm new password:
CertUtil: -RecoverKey command completed successfully.

```

Figure 1.29: Creating the certificate file for transfer to the user.

 The outputblob file is a PKCS#7 file. It contains the KRA certificate, the user certificate, and chain and Inner content—an encrypted PKCS#7 containing the user's private key. When the Key Recovery Agent uses the RecoverKey command, the user certificate and private key is extracted into a PKCS#7 file suitable for import into the user's certificate store. A password is placed on this file to prevent tampering.

---

## **Chapter 2: Securing Web Services and Web Servers—the Administrative Perspective**

**Q 2.5: We do not allow users to store data on their hard drives. They are provided a place on a file server. I can protect this area with discretionary access control lists, but how do I protect data during transport from client to file server?**

**A:** There are several ways to secure data in flight, including using virtual private networks (VPNs), IPSec, and the Secure Sockets Layer (SSL). VPNs are usually the methodology of choice when transferring data across the WAN, while transport-mode IPSec, explained in Question 8.5, is preferred for transferring files on the LAN. However, another methodology exists for protecting files in transport on the intranet, WebDAV over SSL.

WebDAV is the Microsoft implementation of the Distributed Authoring and Versioning extension to HTTP/1.1. You can read about DAV in Request for Comments (RFC) 1518. It was originally designed as an alternative to using FTP to publish files to a Web server, but can also be used as an alternative to SMB. If the Web client is installed, Internet Explorer (IE), Microsoft Office applications, and the Windows Desktop can be used to read and write files to a WebDAV-enabled folder. Office applications can also directly open files from and save files to the Web folder, much as they would use a regular local folder or shared folder on a file server. To use WebDAV securely requires securing the IIS Server, the Web folders and the Web site that hosts them. Our focus here is securing data in flight, but we'll start with a secure implementation of WebDAV.

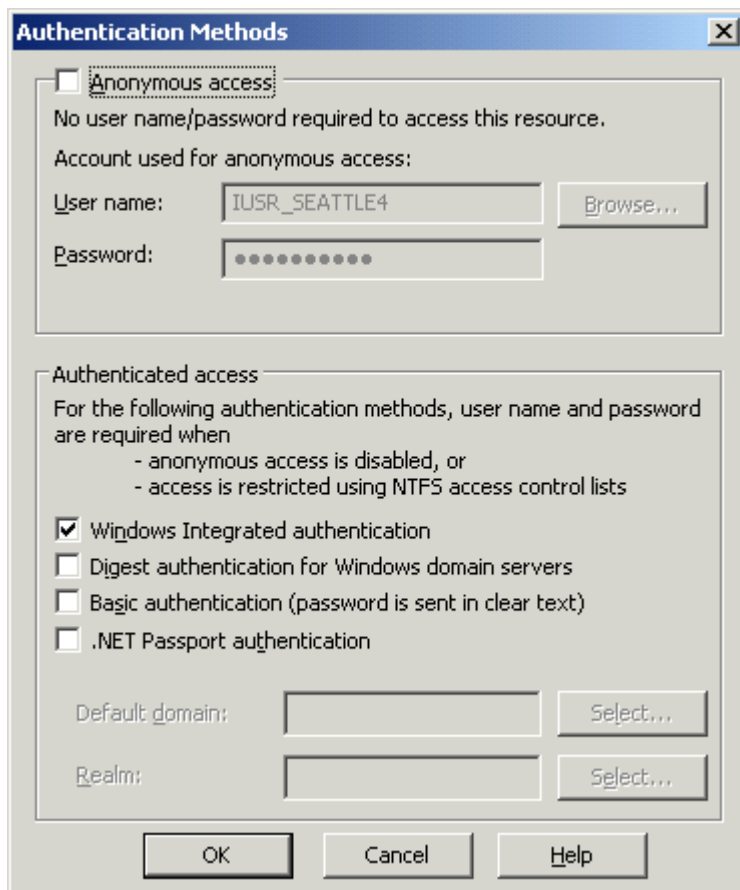
To use WebDAV in Windows .NET, you must WebDAV enable the IIS 6.0 Web server and create Web folders on it. (Web folders and WebDAV can also be used with IIS 5.0 and Windows 2000—Win2K.) Then, using the Web client, files can be transferred from the client computer to the Web folder using HTTP. No file share is necessary on the Web server. WebDAV itself does not provide any mechanism for protecting data in transport. However, you can protect data during transfer to the Web folders by establishing and using SSL—after authenticating the connection with the Web server, all data is encrypted during transport. Files saved in the WebDAV folders are not encrypted.

### ***Preparing and Securing Web Folders for WebDAV***

First, you must enable WebDAV, create and secure the Web folder. To do so, create a folder on the Web server, and apply NTFS file permissions to limit its access to the Windows groups that should use it. This first folder will be the location for the Web site. Next, open the Internet Information Services Console, right-click the Default Web Site (or Web site you have created), and select New, then Virtual Directory. Click Next on the New Virtual Directory Creation Wizard Welcome page. Give the virtual directory an alias to be used for accessing it (for this example, I'll name it Stuff), then click Next. Enter the path to the new folder or use the Browse button to browse to its location, then click Next. Set folder permissions.

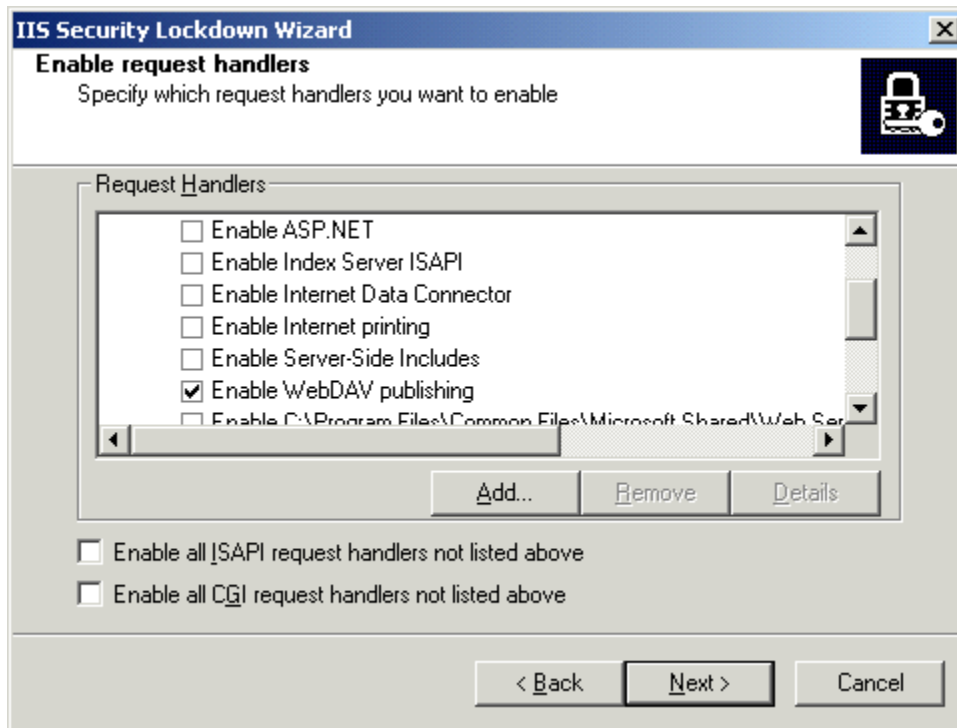
Because this folder will be the root for the WebDAV folders, you might want to set it with run scripts and read permission. I added write permission so that approved users can save files to the folder, and browse (Directory Browsing) so that users can see the files that are stored there. The appropriate permissions to set will depend on your implementation. You might not want, for example, users to see the files available or you might even want users to only be able to write files but not read them. Keep in mind these are virtual directory folder permissions. The underlying NTFS permissions further control who can do what with the files.

Click Next, then click Finish to complete the creation of the virtual directory. Set authentication. On a static Web server, anonymous connections are allowed; however, they are not a good idea when enabling folders for WebDAV. No one should be allowed to access a Web folder for reading or writing without proper identification. A good choice for Web-based authentication on the intranet is Windows integrated (see Figure 2.10). The dialog box that Figure 2.10 shows can be located on the property pages of the Web site. It is reached by clicking Edit at the top of the Directory Security page. Basic authentication means passwords are passed in the clear, which might be OK if you will also use SSL.




**Figure 2.10:** Set authentication methods.

Next, enable WebDAV by right-clicking in the IIS console on the local computer, and selecting Security. Doing so runs the IIS Security Lockdown Wizard. Click Next twice to advance to the Enable Request Handler's page. Scroll down and select the Enable WebDAV Publishing check box, as Figure 2.11 shows, click Next, then click Finish to complete the wizard.



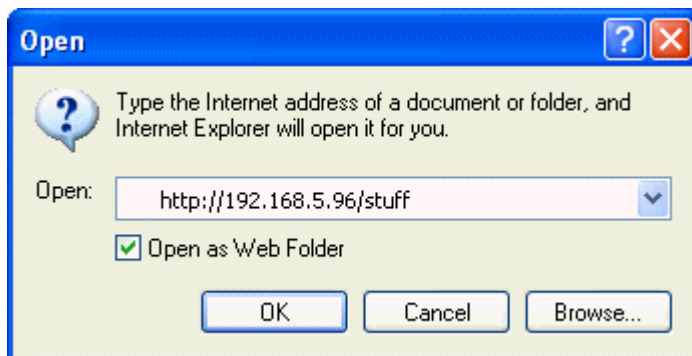
**Figure 2.11: WebDAV is disabled by default. To enable it, run the Security Lockdown Wizard.**

 The WebDAV ISAPI request handler is not enabled by default to prevent its malicious use. Before you enable WebDAV, you should thoroughly consider the additional risk it entails. Remember, WebDAV enables access to documents using Microsoft Office, many versions of IE, and other products that meet the HTTP/1.1 WebDAV specification. It does so over port 80. Therefore, unlike file sharing, which can be blocked at the firewall, WebDAV manipulation of this data can be accomplished across a port commonly open on the firewall. If your intention is to allow such access, you must ensure that other controls are in place. If you will merely be using WebDAV on your intranet, you must still take the appropriate action to block external access to port 80 of the WebDAV-enabled IIS server on your internal network. Security controls for WebDAV are summarized later.

Before setting up SSL, it's wise to test the Web folder for accessibility. To do so, make sure that your Windows XP client has the Web Client service enabled and started (you can check and also enable it if it is disabled by going to the Start menu, selecting Administrative Tools, then selecting Services). You will also need the appropriate URL. For this example, I used the IP address of the Web site, followed by the alias assigned to the virtual directory. Then, to test, try one of these options:

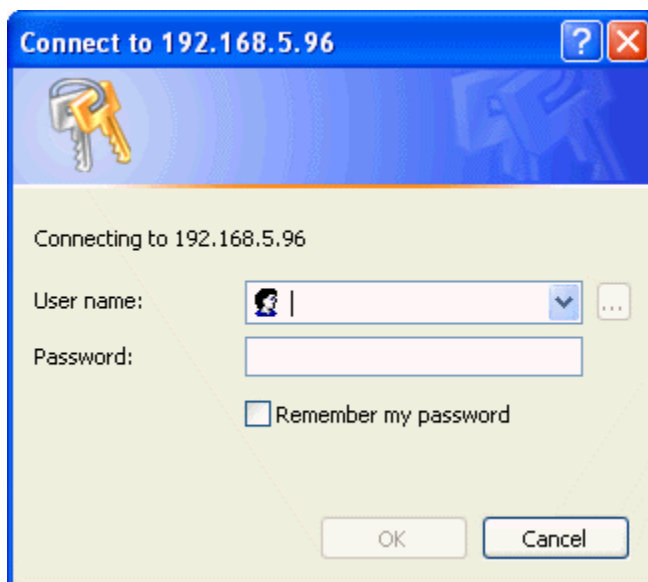
- Any Office product should be able to save or open files saved to the Web folder as if the folder existed on the local hard drive. Instead of using a local drive path or mapping a network drive, simply save to My Network Places, and select the correct Web folder or type the URL.
- Open My Computer, and select My Network Places, double-click *Add a new Network Place*, and enter the URL. A file can be dragged from the desktop or Explorer window to the Web folder.

- Open IE, from the File menu select open, input the URL, and select the *Open as a Web folder* check box. A file can be dragged from the desktop or Explorer window to the Web folder (see Figure 2.12).



**Figure 2.12:** Use the IE File menu's Open dialog box to open a Web folder.

If your logon does not match the NTFS permissions set on the underlying folders, you will be prompted for a user ID and password, as Figure 2.13 shows.



**Figure 2.13:** The site doesn't allow anonymous connection, so you might need to enter a user ID and password.

### **Using SSL to Transfer Files to Web Folders**

Setting up SSL for use with Web folders is simple and straightforward. The basic process is the same as for setting up SSL for other purposes. You must decide where the certificate should come from and where it should be installed? Because our example is an intranet Web site, the steps which follow obtain the certificate from an internal Windows .NET Certificate Authority (CA). If you choose, you might obtain a certificate from one of the commercial CAs.

IIS offers several locations where the certificate might be installed. It can be installed at the Web site level, and all connections to the Web folders can be required to use SSL. It can be installed on a Web folder. In this manner, connections elsewhere on the site do not have to use SSL. It can also be installed at a subfolder level. Our example consists of a Web site set up specifically for WebDAV, so the certificate will be installed at the Web site level.

To require SSL for Web folder access, you must request a certificate, install it, and require SSL. To request a certificate, in the Internet Information Services console, right-click the Web site, and select Properties, then select the Directory Security tab. Click Server Certificate in the Secure Communications area of the page, click Next on the wizard welcome page, select *Create a new Certificate*, and click Next. Select *Send the Request immediately to an on-line certification authority*, and click Next. (If you must obtain your certificate from a third-party CA, you will need to create a request for forwarding to the authority, then install the certificate when you receive it.)

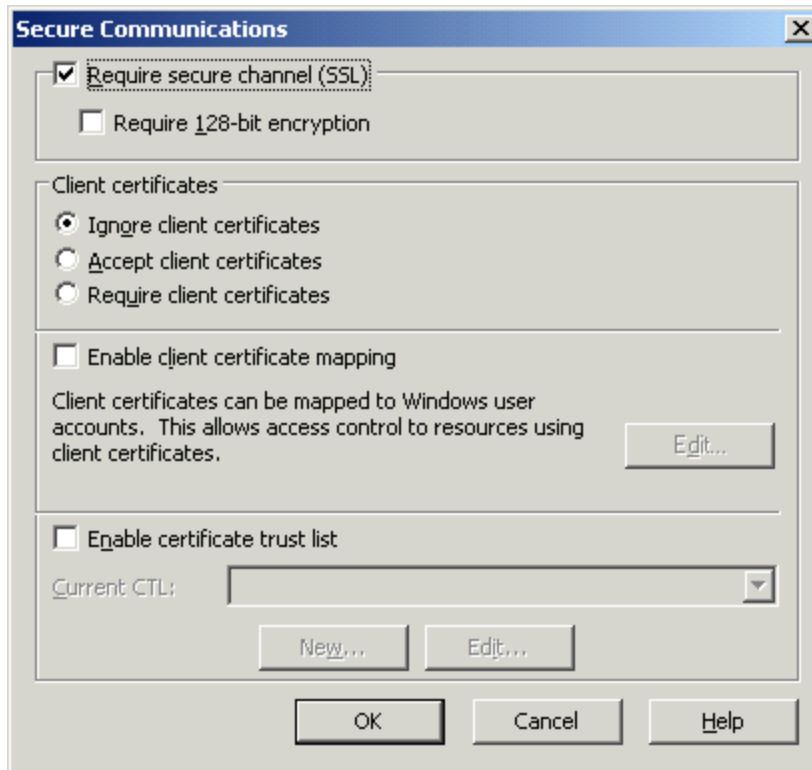
Next, you'll need to enter a name for the new certificate. Change the bit length if you desire stronger security. In general, the longer the bit length of the key, the better the security but the worse the performance. Click Next. Enter the legal name of the organization in the Organization text box, enter the departmental name in the organizational unit (OU) text box, and click Next. The NetBIOS name of the site (the server name) will appear in the Common Name text box. You might replace it with the fully qualified domain name (FQDN) or, because this is strictly planned for intranet access, leave the NetBIOS name. In this example, the FQDN was entered. Click Next.

In the next step, enter the city and state location of the Web server, and click Next. Select a CA, then click Next. All online CAs should be available in the drop-down box. Review the settings on the Certificate Request Submission, and click Next. Click Finish on the notice that the certificate has been installed, select the Web Site tab, enter the port number for SSL (the standard port number is 443), and click Apply.

Return to Directory Security page, and click Edit in the Anonymous Access and Authentication Control section. Change authentication method to Basic Authentication if desired. The SSL connection will occur before authentication and thus the entered user ID and password will be encrypted.

Click View Certificate to review the certificate details and verify that a Web Server certificate has been installed. The designation of the type of certificate is found by examining the Certificate Template Name on the Details page. The specified use of the certificate—Server Authentication—can be found on the General page. Click OK to close the view, then click OK to close the properties page.

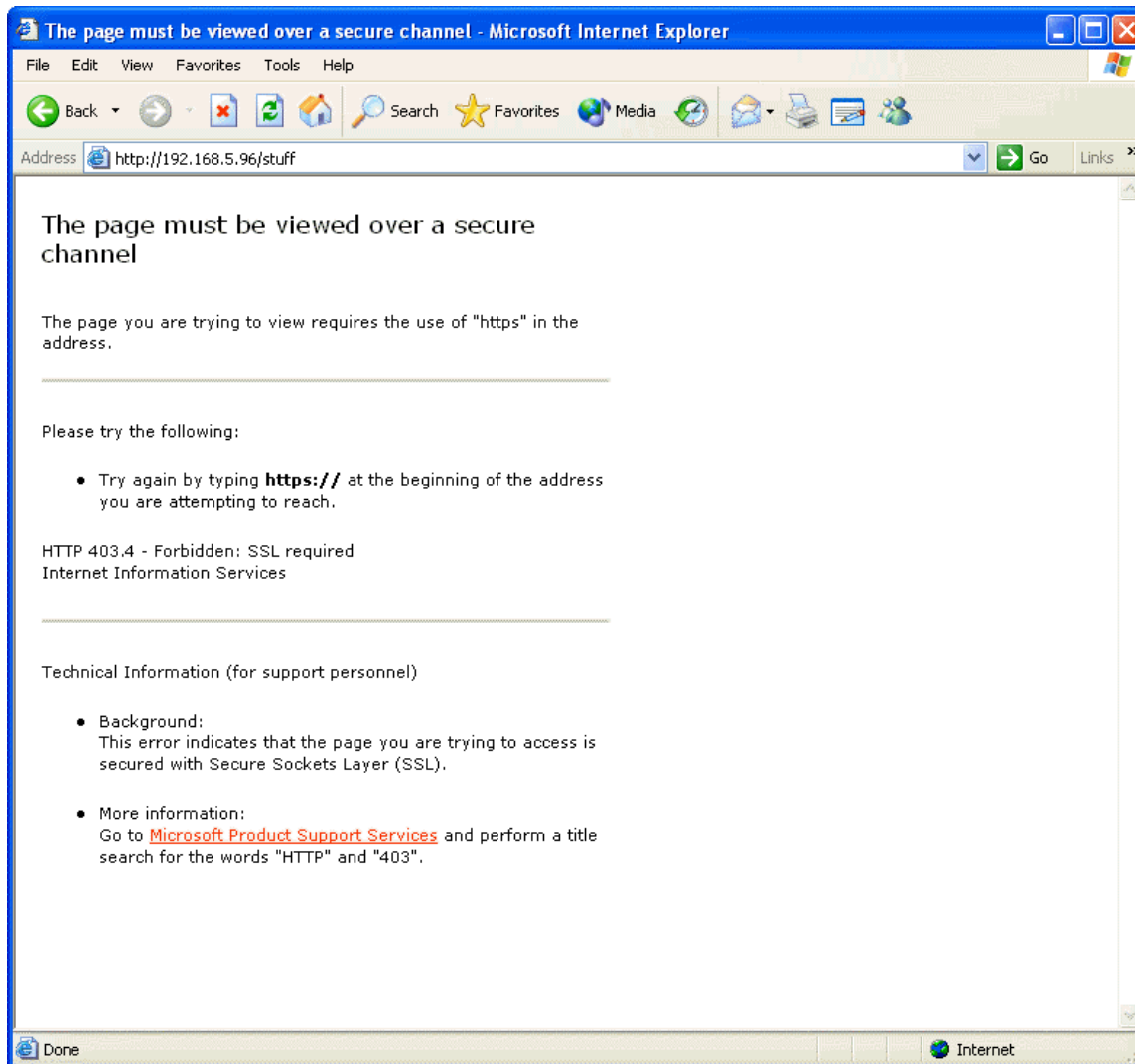
After the certificate is installed, HTTPS can be used to access the server and ensure encryption of the entire communication between the client and the server. However, ordinary HTTP can also be used. To ensure that all communication is encrypted, you must configure the Web server to only accept SSL. To do so, return to the Directory Security properties page for the Web site, click Edit on the Secure Communications section of the page, and on the Secure Communications dialog box, select Require Secure Channel. If all clients are capable of 128-bit encryption, select this option as well (see Figure 2.14). Click OK to close the window and apply the setting. Now all clients will be required to use HTTPS to connect.



**Figure 2.14: Require SSL, as clients will not remember to request it.**

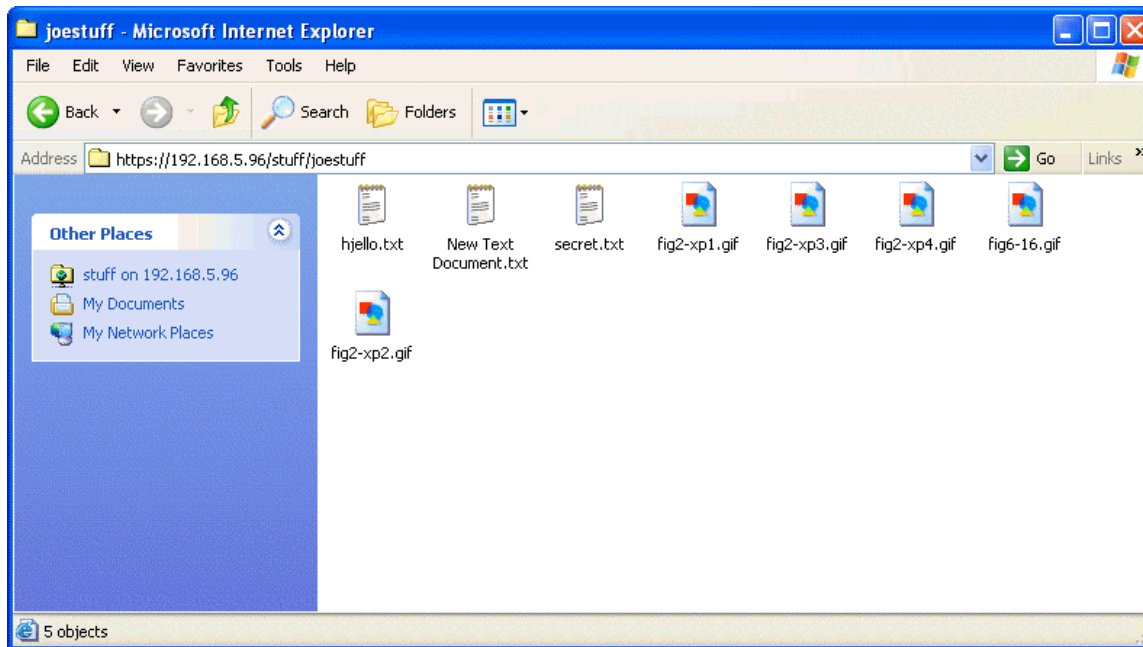
Click OK to close the Properties pages. To test, return to the client and enter the URL to access the Web folder. If you use HTTP, it will not work and provide an error message. Modify the URL to use HTTPS and access is allowed. Note that the failure came before the request to enter a user ID and password, thus showing that the SSL connection will occur before logon. Even use of the basic authentication clear-text password will actually be encrypted when using SSL.

Be sure to retest connections using HTTPS. If you forget, you will be warned, as Figure 2.15 shows.



**Figure 2.15: If properly configured, any attempt to access the Web folders without using HTTPS will fail.**

Fortunately, entering HTTPS and repeating the operation will get you to the Web folder, as Figure 2.16 shows.



**Figure 2.16:** After authentication, the Web folder is ready for use in reading and saving files.

✎ When securing your Web folders with SSL, it is wise to understand that a certificate used by this purpose must be from a trusted CA or you will receive warnings when first attempting access. On an intranet using an internal CA, it is likely that trust of the CA, and thus the certificate issued for the Web server, will already be established. However, if a Windows .NET standalone CA is used, or if clients not joined in the domain of an Enterprise CA are used, you might need to acquire a copy of the CA certificate and place it in the Certificate store of the client system. You can do so during the first access to the Web server, if users are properly instructed.

## Chapter 3: Understanding Active Directory Foundations

### **Q 3.5: Is it possible to place the Active Directory database on a different drive from its logs?**

**A:** Yes! It is possible, and there are very good reasons for doing so. First, having the database and log separated makes more room on the database drive. As the database expands, there will be more room on the drive because the log files are not taking up space (adequate room on the log file drive must also be provided). Furthermore, adequate space for log files can also be assured. Second, keeping log files on a separate physical disk from the database can assist in recovery. Third, should the hard drive on which the database is installed become uncomfortably full (Active Directory—AD—requires plenty of space and should not be pinched by lack of free space on its drive), moving the database to another, larger drive, is an alternative to rebuilding and restoring.

First, however, a little background is in order. Like many transaction-based databases, changes are not written directly to the AD database. Instead, changes are written to a transaction log and eventually the directory is then updated. This two-step process preserves the integrity of the database should processing be interrupted. Without the transaction log, an interruption might leave a transaction half done. Suppose, for example, that you instructed AD to move Joe Blow from the Accounting organizational unit (OU) to the Finance OU. Although you and I see the transaction as one smooth move, the database sees it as two. First, Joe is removed from Accounting, then he is added to Finance. If the power goes out after he is removed from Accounting but before he is placed in Finance, we'll have a problem. What's happened to Joe's account? When the server reboots, Joe's account might have disappeared from Active Directory Users and Computers. Worse still, some of Joe's data might have made the move, while other parts may be lost. However, the transaction log ensures that, even in the case of a power failure, transactions can be recovered.

During normal operation, each transaction log is *checkpointed*, that is, it is marked as to which transactions have completed. You can think of checkpointing like putting a check mark next to completed tasks in a task list. Changes to AD are written one after another to the log file. Meanwhile, previously written transactions are entered into the database. When a transaction is complete, the checkpoint is moved. Should the system go down, the checkpoint shows where completed processing was. When the computer reboots, the logs are checked, and if a transaction is incomplete, the data in the logs can enable its completion or at least the removal of partial changes so that the database is the same as it was before the transaction began. The ability to rollback or roll forward transactions cements the integrity of AD and is essential to its operation. In many cases, full recovery is possible after system failure.

If, however, the database should become corrupt or the problem is the hard drive itself, recovery from backup is facilitated if all logs as well as the database have been backed up. A database restore, along with the application of backed up logs can bring the directory very close to current.

If logs and database are on separate physical drives, then one of two scenarios is possible: If the database drive is lost, it may be possible to restore the database from backup, then apply the logs for the log disk (if the logs are online, this action is automatic). Everything can be returned to normal. If the log drive is lost and the current database is intact, the system will discard any partially completed transactions.

In many cases, of course, this choice might not be valid for recovery. If multiple domain controllers for the domain are present and online, the loss of one domain controller can be recovered by installing a new Windows .NET system and promoting it to domain controller. Normal replication with the other domain controllers will bring it up to snuff. However, lack of another domain controller or changes that had not replicated to other domain controllers might require the restoration of the failed domain controller's database. In any case, modifying the log location requires few steps. You will use the `ntdsutil.exe` command, a command that is used to manage AD. It can be used to maintain the database, move the logs and database, control single master operations, and remove metadata left behind by domain controllers that are not cleanly removed or that fail during install.

To do so, take the database offline. Moving the database or its logs cannot be accomplished while they are being used. To do so, shut down the domain controller. As it reboots, when the Starting Windows progress bar appears, press the F8 key to start the server in directory services restore mode. From the Advanced menu, select Directory Services Restore Mode. Use the `ntdsutil` utility to move the logs. This command moves the logs and changes the registry to reflect their location. To move the logs to drive D, type

```
ntdsutil
```

At the `ntdsutil` prompt type

```
files
```

Then type

```
move logs d:\ntds
```

A new directory, `ntds` is made on the drive and the files are moved. View the new location by typing

```
info
```

Verify the integrity of the database by typing

```
integrity
```

Back up the system state. (This will backup AD.) If you do not perform a backup, the location of the log files might not be retained. Finally, restart the computer.



If, instead of moving the log files, you want to move the directory database `ntds.dit`, use `ntdsutil` and the `files` command, then enter

```
move DB to D:\directory
```

where `directory` represents the name of the directory on the D drive that you would like the database to be.

## Chapter 4: Fulfilling the Promises of Group Policy

### **Q 4.5: How can I prevent wireless access points from become an unguarded entry point into my network?**

**A:** Wireless access points (WAPs) can become back doors into your network; but properly configured and managed, they can be safely used. Windows .NET Group Policy provides a tool that can assist in your security efforts.

Unrestricted proliferation of WAPs throughout your network can undermine many of your security defenses. It will not be easy to control them. Unlike many technological advances, WAPs have two things that make them immediate candidates for abuse. First, their price point is low. For a few hundred dollars, you can purchase an access point and a wireless network card for your PC or laptop. Many already have company-issued or came-with-the-laptop wireless connectivity installed. Second, the technology is engaging, convenient, and easy to implement. WAPs arrive preconfigured in some cases, and computer interfaces often automatically detect their presence. A technically unsophisticated, ordinary user can purchase the box, stick it under his or her desk, and cable it to the company LAN. With minimal to no help on the wireless LAN card enabled laptop, the user can now roam around his or her cubicle and to adjacent areas without the network cable tether.

It would certainly be nice if technology could step in and offer the ability to either prevent these access points from being able to exist on your network or immediately detect them for you. There is no way to do the former, nor easily the later. You currently cannot eliminate the problem, but you can take steps to manage it:

- Have a strong written policy that specifies appropriate use of wireless networking and the consequences of infractions.
- Have a strong education element. Teach users how to use the corporate wireless LAN and the problems rogue access points can cause.
- Use technology to control and secure authorized wireless communications.
- Adopt technology that seeks to manage and control wireless access.

You must implement the first two, and Wireless Access Policies in Windows .NET Server Group Policy can help with the last two.

### **Securing Wireless Communications**

Wireless communications have come under fire for their weak data-encryption implementation and lack of sound authentication mechanisms. Add to these shortcomings a cavalier implementation, and you have the recipe for disaster. Although techniques for spying on computer systems using specialized antennas, watching the blinking network access lights, telescopically viewing computer monitors, and other techniques have been used in the past, these techniques require some sophistication or clear line of sight to implement. Wireless networking as now used requires neither.

For most, the threat to data exposure has been perceived as limited to penetration of limited access points to the network. No access points, no threat. Limited access points with detection and firewalling lessen the threat. Difficult to entirely prevent exposure, but if properly designed and configured, an acceptable risk. However, wireless access to the corporate network exposes internal communications to external entities. The outsider doesn't have to physically connect to the internal LAN, penetrate the corporate firewall, nor discover unprotected dial-up access. He or she has only to sit within the range of the WAP (60 to 300 feet for most) and have his or her own wireless networking card. A single, improperly configured WAP serves up the network to any such passerby. Many properly configured WAPs are easily subject to penetration due to weak encryption implementations and the existence of tools that purport to decrypt communications.



Several wireless 802.xx standards exist. Many of the differences in technology are not important to security but can cause problems in implementation. Before adopting a standard, you will want to consider these differences, but first consider the security facilities offered by the particular solution. The information offered here is merely to help you differentiate between the currently available products:

802.11 1 or 2Mbps transmission, 2.4GHz band, using frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum encoding (DSSS) schemes

802.11a extends 802.11 to provide 54Mbps in the 5GHz band, uses orthogonal frequency division encoding scheme; typical access areas extend up to 60 feet

802.11b Wi-Fi extends 802.11 to provide 11Mbps in the 2.4GHz band and using only DSSS; typical access areas extend up to 300 feet

802.11g is an extension that provides 20+ Mbps in the 2.4GHz band

802.1x is currently a draft proposing authentication mechanisms for 802.11 wireless networks

Four security implementation opportunities exist for wireless networking: do nothing, use standard security options available for configuration on the access points, use standard security options plus firewall the access point, add 802.1x authentication and improved technology.

### Standard Security Options

Most commercially available WAPs include five purported security mechanisms. The following bullet points briefly discuss each mechanism.

- **Encryption**—The Wireless Encryption Protocol (WEP) can be switched on and will be used to encrypt data transmitted across the wireless LAN. Although the implementation has been demonstrated to be weak, it is nevertheless useful in thwarting casual eavesdropping and many intentional penetration attempts. The determined attacker will persevere. I recommend turning on encryption.
- **SSID**—The SSID is merely an identification mechanism and is not a security device. Many systems broadcast the SSID and many wireless connectivity applications automatically detect the SSID or all nearby wireless networks. For those that are not automatically broadcast, keeping such a secret is next to impossible. However, you can modify from the default. Most WAPs have a default SSID. These are not secret, and attackers will try these default SSIDs.
- **Turn off broadcast**—Broadcasting the availability and SSID of a WAP is helpful to network users. Unfortunately, it is also helpful to intruders. Preconfigure authorized systems to use authorized access points and turn off broadcasting. Yes, security through obscurity is not particularly good security, but it does limit exposure.
- **Use MAC addresses for authentication**—By limiting access to approved MAC addresses, you prevent most unauthorized computers from connecting. Take note, because the authentication is based on the network card address, possession of the computer or the card provides access. In addition, MAC addresses can be spoofed, so this plan isn't foolproof.

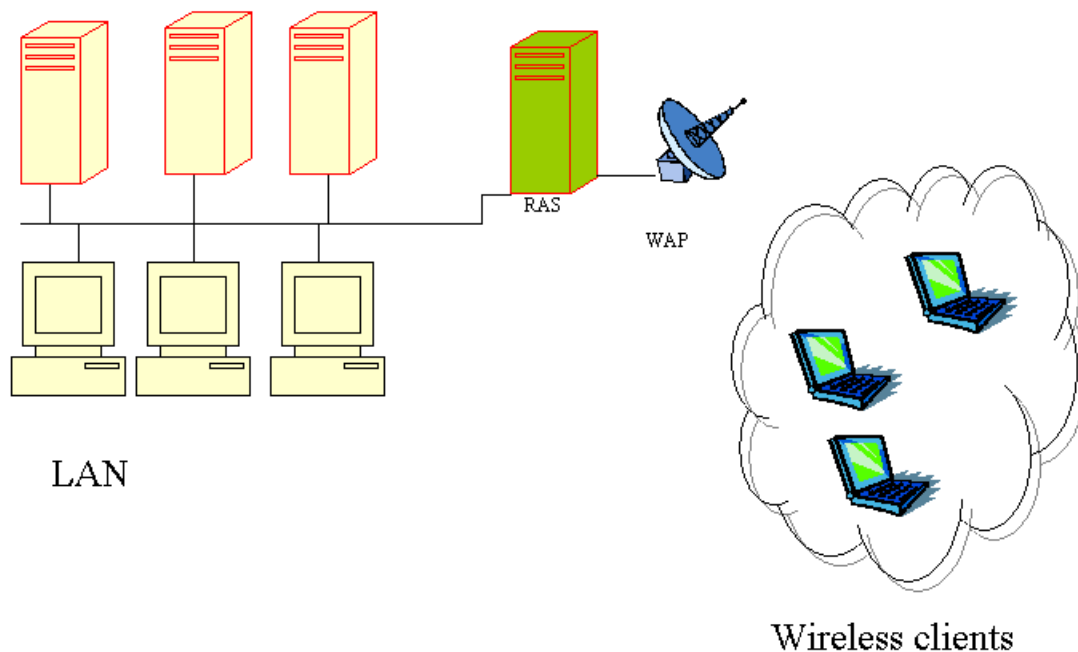


- Insist on direct connection for administration—Allowing wireless access for administration of the access point opens security configuration to anyone within range of the system. By configuring the system to require physical connection, you've limited this exposure. Some systems can require direct serial connection, while others require Ethernet cabling directly to the box.

Although these security mechanisms have their shortcomings, they at least provide some measure of security.

### Standard Security Options Plus Fire walling

Placing a WAP on the internal side of the corporate firewall is not firewalling the access point. Wireless connectivity does not go through the firewall. However, you can firewall the WAP by placing a firewall or remote access server between the WAP and the rest of your network. You can then require virtual private network (VPN) connectivity to the internal LAN, or at the very least, authentication to the remote access server. No internal network resources can be accessed until this authentication is accomplished. Figure 4.16 illustrates this technique. In the figure, the WAP is connected to the external card of a remote access server. Although it is possible to obtain connection to the WAP, connectivity to any internal resource can only be acquired if connection to the remote access server is successful. Additional restrictions such as port filtering or the addition of an actual firewall is also possible.



**Figure 4.16: Setting up a firewall for wireless clients.**

## Add 802.1x Technology

802.1x is currently a draft standard. In an 802.1x network, a network access server (NAS) requires authentication before allowing access to the network. The standard uses RADIUS as the mechanism for providing authentication. Current implementations include Windows .NET server and use certificates (either smart card based user/computer certificates or local certificates on the client computer). In essence, 802.1x places the responsibility for firewalling the access point on the access point itself.

The standard also describes approved authentication protocols and the implementation of approved data encryption processes that address the weaknesses of WEP. Two choices for authentication are Extensible Authentication Protocol (EAP) and Protected Extensible Authentication Protocol (PEAP). Windows .NET server provides support for both.

EAP is described in Request for Comments (RFC) 2284. It was designed to support multiple authentication methods including smart cards, Kerberos, public key, one-time passwords, and other methodologies not yet in existence. The goal of EAP was to offer the ability to provide different authentication methods without having to re-program the NAS. The NAS, which sits between the clients and a back-end authentication server, can simply act as a pass-through and does not need to understand multiple authentication methods or even specific implementations of them. The NAS simply needs to understand EAP, then let the client and the authentication server do the work. The original specification was developed for use with remote access servers such as RADIUS, with which clients typically connected to the network through dial-up or Internet connections.

EAPOL, or extensible authentication over LANs, is the adoption of this protocol to wireless network interfaces to traditional LANs. EAPOL is diagrammed in Figure 4.17.

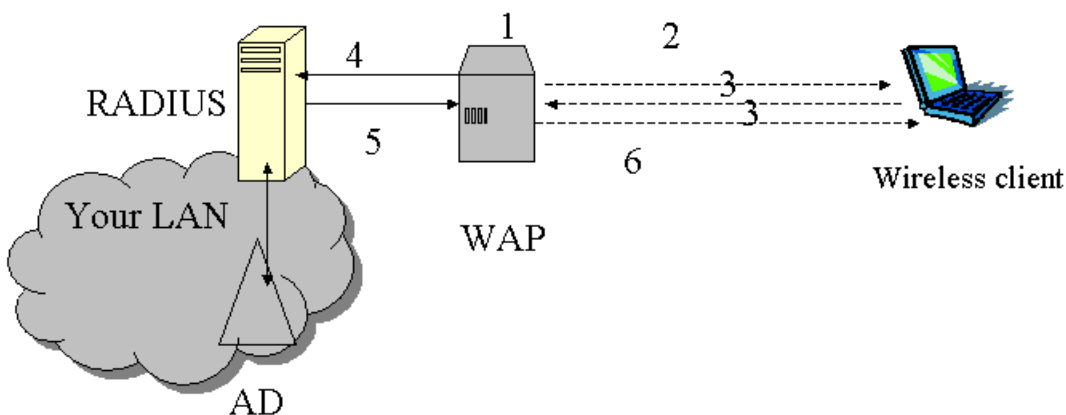


Figure 4.17: Steps in EAPOL.

As Figure 4.17 shows, the NAS, a switch (or, in our case, WAP) provides a point of entry for the user. The switch detects a client and sends the EAPOL Request-ID message to the client. The client responds with an EAPOL Response-ID that includes authentication information. The NAS encapsulates the Response-ID in a remote authentication dial-in user service request packet and forwards it to a RADIUS server. (The NAS acts as a relay of messages from the client using EAPOL and to the RADIUS server using RADIUS packets.) The RADIUS server responds with accept or deny packets that include encapsulated EAP success or failure packets. (RADIUS on a .NET network will use Active Directory—AD). The access server forwards to the client. If authentication is successful, the port on the access server is open and the user authenticated.

PEAP is essentially EAP wrapped in Transport Layer Security (TLS—a technology similar to SSL). Additional modifications also support improved security. In fact, it is being developed to address weaknesses in the EAP standard. Three improvements exist:

- Because EAP is wrapped in TLS, the EAP session between the back-end authentication server and the client is encrypted and the integrity is protected in a TLS channel. Mutual authentication between the back-end sever and the EAP client is required. EAP did not require mutual authentication and was felt to expose too much information about the process—information that would make it easier for an intruder to mount an attack. Now, no EAP conversation occurs until the TLS session is established and all EAP communication is encrypted.
- TLS provides built-in support session resumption and management of fragmentation and reassembly—two networking issues with EAP. Because EAP doesn't include these capabilities, each authentication method has to provide them, thus resulting in a duplication of effort as well as an additional exposure to denial of service or vulnerabilities due to poor code.
- TLS provides support for key exchange and the development of key hierarchy for the generation of authentication and encryption keys. To work with EAP, each authentication method had to do so. These techniques are complex and difficult to get correct. Requiring each implementer and authentication method provides too many opportunities for poor mechanisms and increased vulnerability.

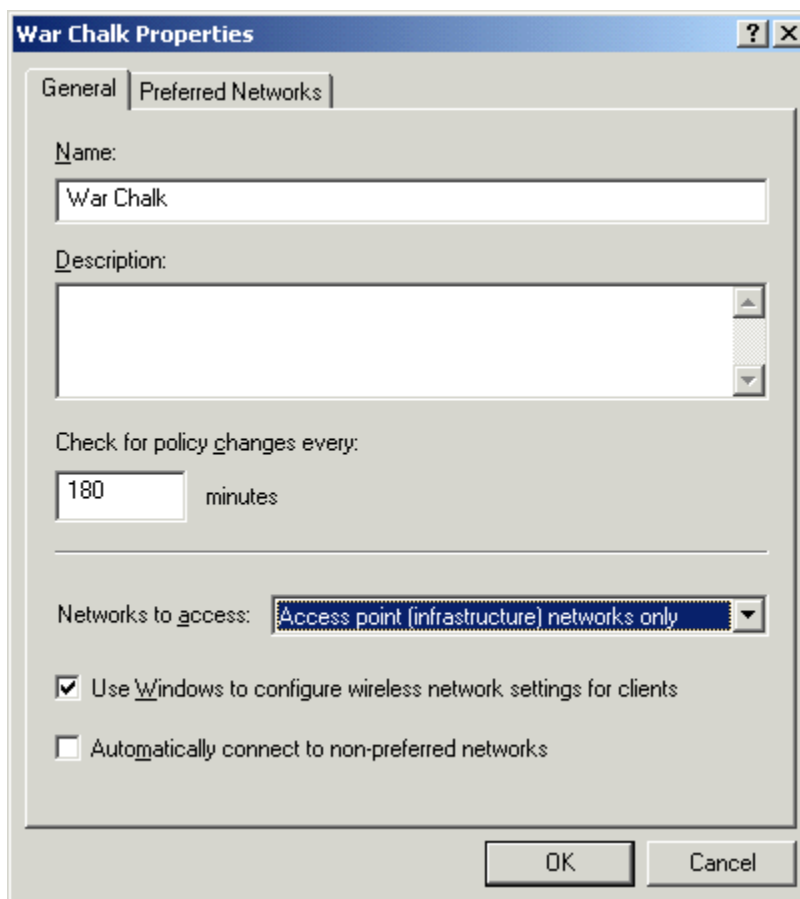
### **Windows .NET Server Wireless Network (IEEE 802.11) Policies**

Windows .NET Group Policy provides an opportunity to centrally control configuration of authorized wireless networks. Although it does not and cannot prevent the existence of unauthorized WAPs, use of a policy provides more than and administrative convenience. First, the policy provides a mechanism to present a list of preferred WAPs. The client will seek to connect to these access points first, in the order listed. A user would have to specifically configure his or her system to attempt connection to other systems. For most people, transparent connectivity is desired. The ordinary user just wants to be able to do his or her job, not figure out how to connect to the network. If there is no benefit to independently configuring their systems (users have wireless connectivity that they don't have to mess with), they won't. The determined individual will always find a way around policy. Second, although the attacker will be perfectly capable of configuring his or her computer, the policies that control authorized access points add technology to reduce that threat. By using Group Policy to set them, you can avoid the possibility that misconfiguration will produce a vulnerability.

Wireless policy can be set in the Computer Settings, Windows Settings, Security settings, Wireless Network Access Policy. First, create the wireless policy. In the GPO, right-click Wireless Network (IEEE 80.11) Policies, and select *Create wireless network policy*. Click Next on the welcome page, then enter a name and description for the policy, and click Next. Click Finish to complete the policy wizard and proceed to editing the policy settings. On the general properties page, the default is to *Use windows to configure wireless settings for clients* (see Figure 4.18), though you can select the *Automatically connect to non-preferred networks*. Important to consider are the choices that require client connections to either:

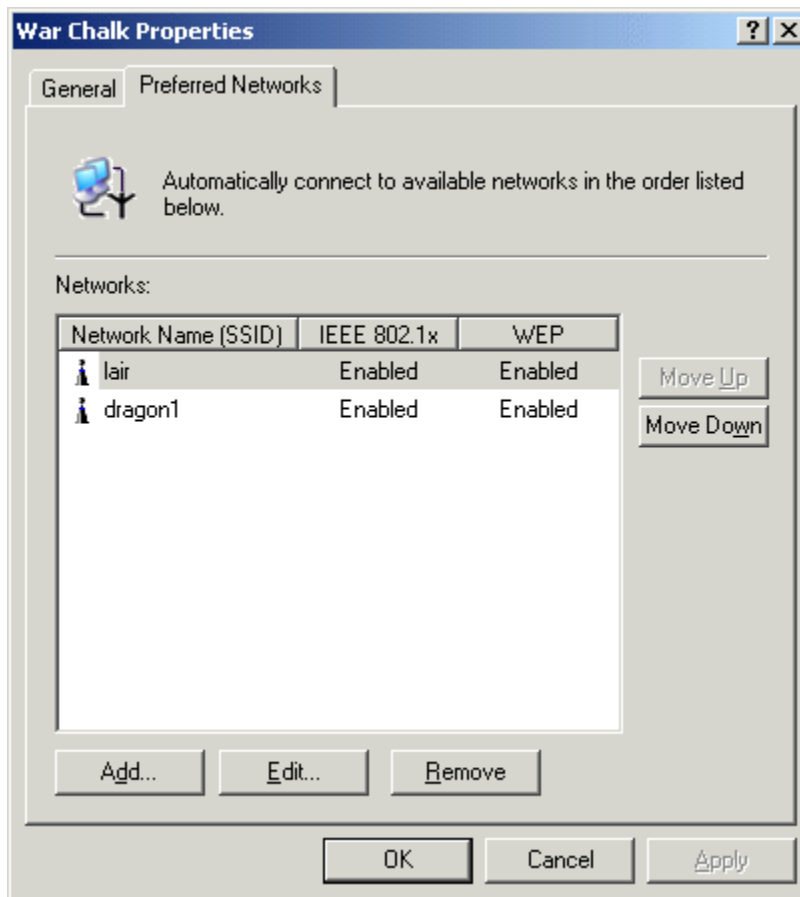
- Any available network (access point preferred)
- Access point (infrastructure networks) only
- Computer-to-computer (ad-hoc networks only)

You allow ad-hoc as well as infrastructure networks with the first choice; otherwise, set restrictions. You might need different policies if you allow a select group of users access to more than one kind of wireless network. Remember that you can create different policies and implement them at either the domain or OU level. What you cannot do is create multiple policies in a single Group Policy Object (GPO) and assign them to specific users or computers.



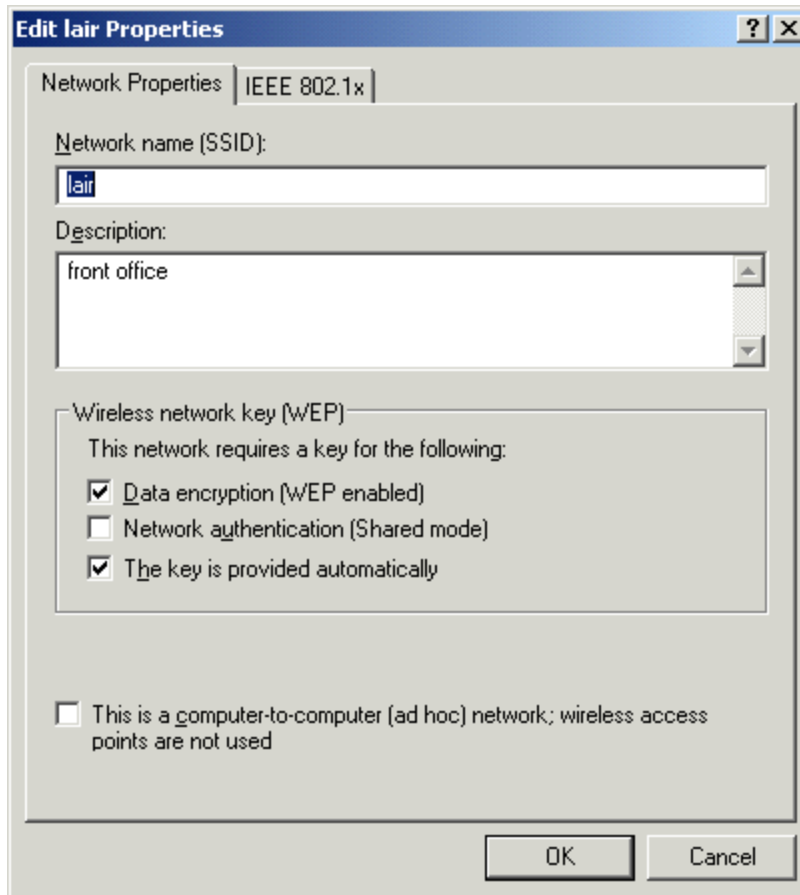
**Figure 4.18:** Identify the policy and the type of wireless network authorized.

The Preferred Networks tab, which Figure 4.19 shows, is used to identify configuration for each authorized access point. For each, click Add, then add the network information.



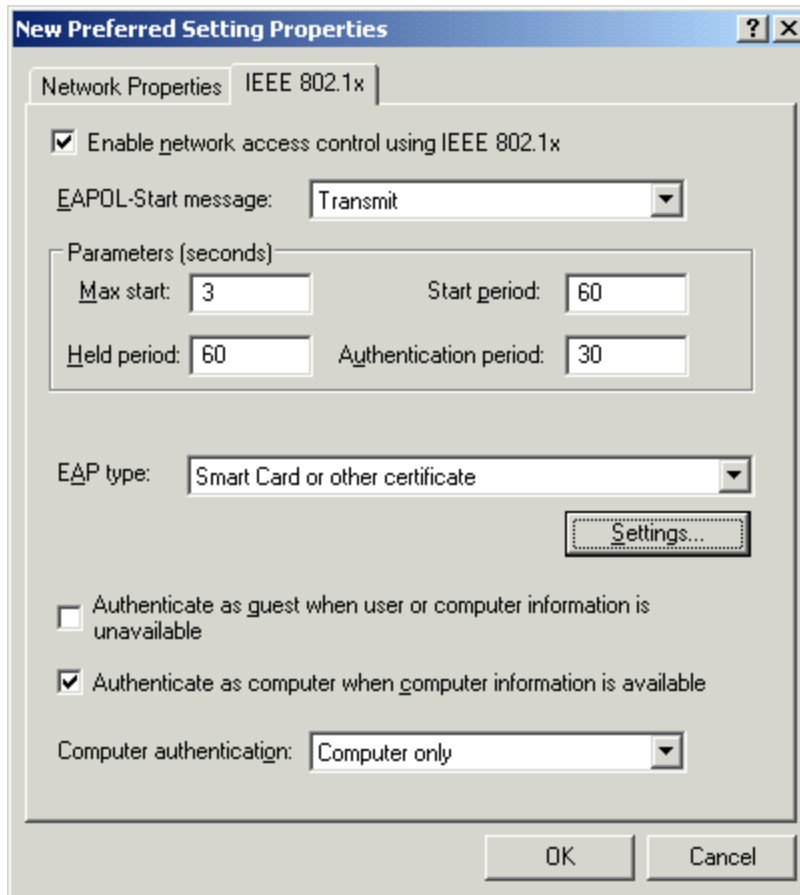
**Figure 4.19: Identify and add network information.**

On the Network Properties tab (see Figure 4.20), enter the network name, or SSID, and a description.



**Figure 4.20: Configure WEP.**

Choose the WEP key and authentication modes, or ad-hoc designation if appropriate. If IEEE 802.1x is supported, use the IEEE 802.1x tab (see Figure 4.21). Your first choice is to select between the use of EAP or PEAP.



**Figure 4.21: Configure 802.1x.**

Use the Settings button to configure authentication behavior. You can select whether smart card or computer resident certificates are required, and identify which CA(s) are to be trusted. If a Microsoft Enterprise CA is online, it will appear as an option (see Figure 4.22).

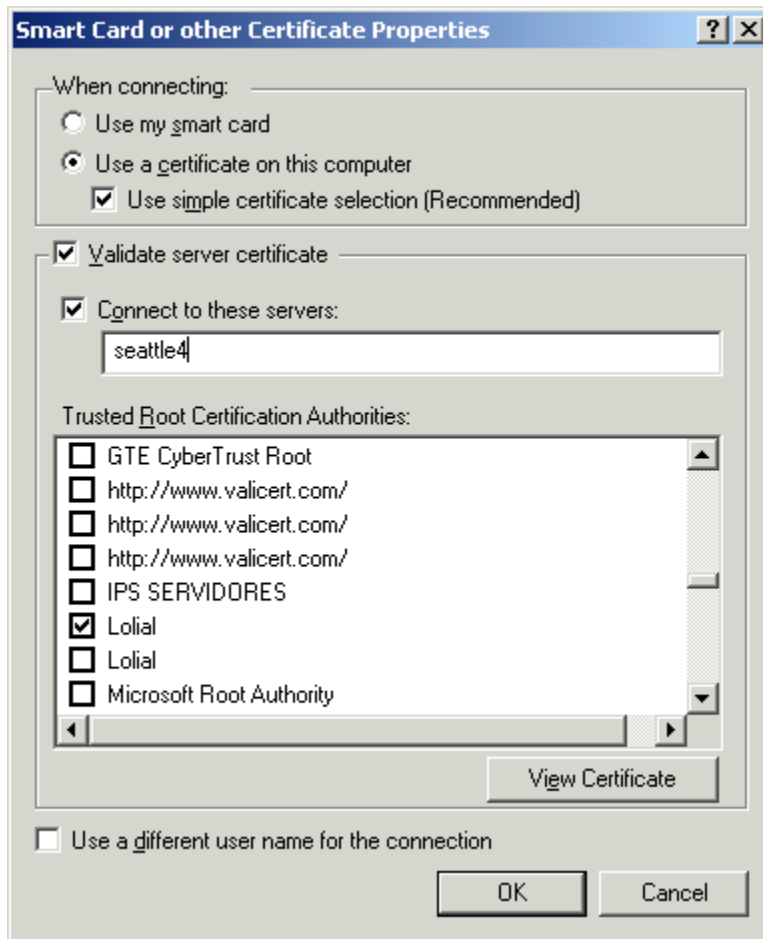


Figure 4.22: Select smart card or computer certificate and identify Trusted Root Certification Authorities.

## Chapter 5: Administrative Authority

### Q. 5.5. What exactly can an Enterprise Administrator do?


**A:** The Enterprise Administrator can do whatever the Enterprise Administrator wants to do. The Enterprise Administrator role was established in Windows 2000 (Win2K). An Enterprise Administrator can administer the entire forest. To be such an uber administrator requires membership in the Enterprise Admin group, which only exists in the root domain in the forest. By default, its only member is the root domain local Administrator account. Much of the authority for this group exists as a result of its membership in the Administrators group in each domain, but it also has specific rights of its own. These rights typically provide management of forest-wide resources. Examples of such rights are the right to install an Enterprise Certification Authority (CA) and the right to run Resultant Set of Policy (RSOP) across multiple domains in a forest.

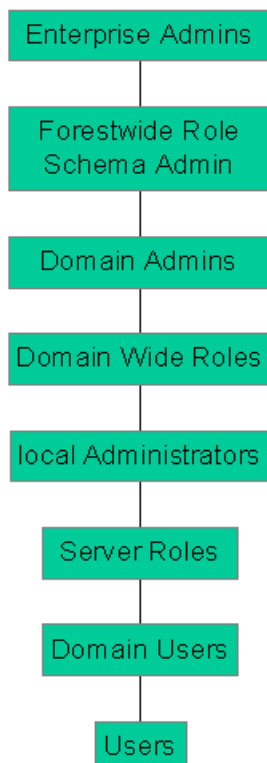
OK, so it's easy to figure out what an Enterprise Admin can do—just think anything administrative anywhere, but especially those things that require authority in multiple domains. To illustrate the difference between a Domain Administrator and an Enterprise Administrator, let's say, for example, that a Domain Administrator has been charged with providing email services or certificate services for his or her domain. The obvious choices are to install Exchange for email and to install Windows .NET certificate services. The first Exchange installation in a forest, however, must be initialized by an Enterprise Administrator, and subsequent installations must also have more rights than a Domain Administrator has. Installing enterprise certificate services is also not possible without Enterprise Admin membership. Other activities may also require rights beyond those of the Domain Administrator. Table 5.4 lists many of the activities that require rights beyond those of the Win2K and Windows .NET Domain Administrator, and indicates some rights that do not require Enterprise Admin membership.

In order to	You must be
Install enterprise root CA	Enterprise Admin, Forest Root Domain Admin
Install enterprise subordinate CA	Enterprise Admin, Forest Root Domain Admin
CA administrator standalone CA, no domain	Local Admin
CA administrator standalone CA in a domain	Local Admin, Domain Admin
Enforce role separation on CA	Local Admin
Authorize Dynamic Host Configuration Protocol (DHCP) in Active Directory (AD)	Domain Admin
Administer all DHCP servers in domain	Domain Admin
Administer the local DHCP server on the local computer	DHCP Admin
Administer every domain in the forest	Enterprise Admin
Run dcpolfix to restore default Group Policy Objects (GPOs)	Enterprise Admin
Create AD application partition	Enterprise Admin
Delegate RSOP control at site and domain level	Enterprise Admin
Enlist a Domain Name System (DNS) server in an AD application partition	Enterprise Admin
Run RSOP in domain	Domain Admin
Runs RSOP across domain boundaries	Enterprise Admin
Install first instance of Exchange (forest prep has not been run prior)	Enterprise Admin, Schema Admin, Domain Admin
Install additional instances of Exchange (or initial if forest prep has been run)	Either Enterprise Admin and local Administrators, or Domain Admin, local Administrators, and Full Exchange Administrator


**Table 5.4: Which rights and activities administrators can perform.**

Although this table doesn't provide a comprehensive list of every right that is considered expressly the domain of the Enterprise Administrator, one approach that might help you understand the role of Enterprise Administrator is to place that role in the context of others. To help you do so, I have put together a picture of role-based administration as represented by the hierarchical structure of groups in Windows .NET. (This representation is my own interpretation—I can find no official reference to hierarchical administration.) At the top of the hierarchy, of course is the Enterprise Admin, followed by the forest role of Schema Admin (a forest-wide role but not one that is all powerful in it). Next are domain roles, starting with Domain Admin, then by specific server or network administration roles. Next, are the Users, followed by specific accounts that have limited rights and permissions. It is not possible to place in this hierarchy those users who have been delegated rights. These are roles that you create by giving users or groups permissions on AD objects. They might also be roles created by assigning specific user rights to custom user groups. There is no one place where they fit, as the delegated or assigned rights might make them uber admins of organizational units (OUs) or minor players with little more than user-level rights on the system(s) they use. Figure 5.8 illustrates the hierarchy, while Table 5.5 places the default user and computer groups at each level. Outside of the base hierarchy, but of great importance, are the roles designated by specific services that might be added. Exchange Administrator is one of these, as are the certificate services roles Issue and Manage Certificates and Certificate Manager.

 This hierarchical viewpoint is meant merely as a way to consider administrative authority in a Windows .NET forest. It does not begin to address the concept of scope, which details where members can come from and where the groups can be given authority.



**Figure 5.8: Role-based administration in Windows .NET.**

 An example of an unusual delegated role is that created by delegating management of DHCP authorization on the network, which is done through the Sites and Services Console. The Ntsservices container is selected, and the delegation of control wizard used to give the Full control permission for this folder, existing objects, and creation of new objects, to a user group that you have created for this purpose.

<b>A Hierarchical View of Windows .NET Administration</b>		
Forest Role	Enterprise Admin	Uber admin
	Schema Admin	Modify Schema
Domain Role	Domain Admin	Administer domains
	Cert Publishers	Computers that can publish certificates to AD
	Group Policy Creator Owners	Members can create and edit group policy objects for the domain
	Domain Computers	Can be managed, domain administration
	Domain Controllers	Serve as domain controllers in domain
Server/Computer Role	Backup Operators	Backup and Restore
	Server Operators	Logon interactively, create shares, shut down computer, format drive, stop start services, backup and restore
	Account Operators	Create and manage groups, users, and computers. Cannot manage administrators, domain controllers.
	Print Operators	Manage printers and document queues
	Network Configuration Operator	Client side of network configuration; cannot install/remove drivers or services
	DHCP Administrators	Can administer local DHCP server service
	DnsAdmins	Can administer DNS Server service
	DnsUpdateProxy	DNS clients that are permitted to perform dynamic updates on behalf of other clients (common membership is a DHCP server)
	HelpServicesGroup	Rights common to all support applications; by default, group membership is the account associated with Microsoft support applications, such as Remote Assistance; users should not be added to this group; it is managed by the Help and Support service
	IIS_WPG	IIS 6.0 worker process group; an account is assigned here by the system to manage a namespace (Microsoft.com would be a namespace, as would peachweaver.com); do not add users to this group
	RAS and IAS Servers	Can access remote access properties of users
User Role	Domain Users	Logon to domain; run applications; access files; use printers; add workstation to domain; bypass traverse checking;

		shutdown workstation
	Domain Guests	Logon on to domain (account disabled by default.); bypass traverse checking; shutdown workstation
	Users	Logon locally; bypass traverse checking; shutdown workstation; run applications; access files, and use printers
	Guests	Logon on (account disabled by default.); bypass traverse checking; shutdown workstation
	Pre-Win2K Compatible Access	Adding the group Everyone here and rebooting domain controllers provides security level compatible with many pre-Win2K applications
	Remote Desktop Users	Log on to a computer from a remote location
	DHCP Users	View-only access to the DHCP Server service
	WINS Users (installed with WINS Server service)	View-only access to the WINS server service

*Table 5.5 Role Hierarchies*

## **Chapter 6: Triple A's—Authentication, Authorization, and Audit**

### **Q 6.5: Which file activity should be audited and how do I do so?**

**A:** You might want to audit file activity for several reasons. File auditing is used to maintain an audit trail of activity against a file or files that are critical, sensitive, or otherwise important, to monitor the activity of a particular user, to determine permissions required for an application to run, and to log changes to system files. Before you set up file auditing, you should determine which of these reasons applies, whether the auditing should be temporary or permanent, understand the proper steps that need to be taken to do so, and have a commitment to review the information collected in your audit. In addition, you need to take the time to understand exactly what information this type of audit can reveal. Auditing file activity does not stop someone from improperly accessing a file—auditing simply records file activity. If that information is to be useful, time has to be spent reviewing the audit logs and associating the records with behavior or results. You might be able to determine who modified the file, if you can determine what time the file was altered and who was accessing it at that time.

### ***Maintaining an Audit Trail***

An audit trail details the activity that occurs with respect to some physical item or process. It's possible that you will want a round-the-clock record of certain file access in conjunction with the processing of some activity or that certain files are so critical, that you will always want to know who touched them and what they did. There are three critical pieces of information to gather.

You must identify the files. This can be determined by combining the data owners' knowledge of the files and processes involved, and by understanding how the processes work on the systems involved. For example, if you need to record access to files involved in accounts receivable, the data owners, the Accounts Receivable department, might not even know the exact names of the files involved. However, the data might be in a SQL Server database, in which case the ability to audit data access would need to be approached from a database design and audit capability. Merely auditing access to the database file would be insufficient, as it would not reveal the activity within the database. You can gather from the data owners what information access needs to be monitored, but you may need to review the system operation to discover where the data is actually kept and the details of how it can be monitored. Other cases might prove more straightforward. For example, the system might include text or other files that can be audited at the file-system level. Alternatively, you might simply need to record access to sensitive documents stored directly in the file system. Maintaining an audit trail can be an ongoing activity or might be required for projects that have a limited timeframe.

### ***Monitoring a User's Activity***

There is no built-in magical button that will allow you to spy on an individual user's activity on your Windows .NET network. However, by establishing file auditing, user rights auditing, and logon auditing, and reviewing the audit logs, you can create a detailed picture. Remember, though, full accountability for every thing a user has done is not always what is required. (And might not be practical as it will produce large log files and long analysis times.) It might only be necessary to note their access to specific files over some timeframe. Such is often the case when an individual has come under suspicion, or doing so might be part of standard operating procedure when someone with access to sensitive files is leaving the company or at critical times such as during quarterly reporting. Although other auditing activities cannot be configured in a granular manner, file auditing can be set to specifically record the activity of a single user.

### ***Compatibility Resolver***

One of the most frequent abuses of administrative authority is the use of membership in the local Administrators group as the key to application compatibility. Windows NT, Windows 2000 (Win2K), and Windows .NET seek to restrict access to key system files and registry keys. In many areas, the Administrators group may have Full Control as its assigned privilege, while ordinary users are only allowed Read. Applications written to publish Microsoft standards do not require more than the configured access rights. Unfortunately, many applications are not written to these standards. They might have been written to require Full control over files and registry keys even when it is not necessary for their function or when it would have been easy to store information in separate containers where Full control could have been granted without violating the standard.

Sadly, in most environments, the answer is to simply put the users who need to run the program in the local Administrators group. (I've even seen some cases in which the user is placed in the Domain Admins group—eek!) Problem solved. Well, the problem of being able to run the application is solved. Unfortunately, the result of this solution is that too many people have administrative privileges. Not only does this run the risk of serious compromise of systems and allow more frequent successful virus infection and Trojan implantation, but it also results in many hundreds of hours wasted in reconfiguring users systems so that they work after the exuberant user has dinked with the settings.

There is an acceptable compromise here. However, it will take some work. The answer is to determine the files and registry keys that the application needs access to, and assign privileges on those files and registry keys. Simply create a group, assign the group the privileges, and add people who must run the application to the group. They now have more access than best practice recommends for some registry keys and files, however, they do not have that access everywhere, and they do not have administrative privileges. You can use auditing to determine to which files and registry keys access is required for an application.

### **Log Changes in System Files**

Ordinary maintenance of systems, including the addition of new services, service packs, and hot fixes, makes innumerable changes to system files. Although you should keep paper records of these activities, you cannot always know the exact changes that result from maintenance and you can't determine any unauthorized changing of files. Although you might not want to keep detailed records of every change that affects any file, you can selectively monitor those files or folders that you consider highly sensitive. (Good candidates are files that the Security Operations Guide templates `baseline.inf` and `baselineDC.inf` set additional security restrictions on. You can download these templates from <http://www.microsoft.com/technet/treeview/default.asp?URL=/technet/security/prodtech/windows/windows2000/staysecure/>.) If you set some auditing on systems files, especially on systems such as domain controllers and on those exposed to the Internet, such as IIS Server, you can match recorded changes against paper logs and determine whether intrusion attempts are occurring.

### **How to Set Up File Auditing**

File Auditing is configured in two steps. First, Object Access auditing must be configured in Group Policy in Computer Configuration, Windows Settings, Security Settings, Local Policy, Audit Policy. Next, each file or folder for which you want to audit access must be configured for auditing. Two procedural items must be considered regardless of the actually step-by-step process. First, the placement of the Audit policy in the GPO hierarchy must be considered; second, the affect of inheritance must be considered.

### **Set Audit Policy**

Audit Policy can be set at any level in the GPO hierarchy. Remember that Group Policy settings are cumulative except where there is a conflict. A specific example that you will need to attend to is that Audit Policy in the Default Domain Controllers policy is set to No Auditing. Setting the Audit Policy in the Default Domain Policy will not change that. You must specifically set (or change to Not Defined) Audit Policy in the Default Domain Controller default policy to audit activity on domain controllers.



Often, when reviewing the security policy of a company, I find that Object Access auditing has not been turned on. The reason? They erroneously believe that magically all object access is then turned on and their drives will fill up with audit records. Nothing could be further from the truth. Auditing must be specifically set on the objects themselves before audit records will be produced.

After you have determined which files you want to audit, to audit access to files and folders, you must first set the Object Access condition in the Audit Policy of the GPO (see Figure 6.14). Set the policy to record Success and Failure. The true measure of what actually is recorded is determined when the settings are made at the file and folder level.

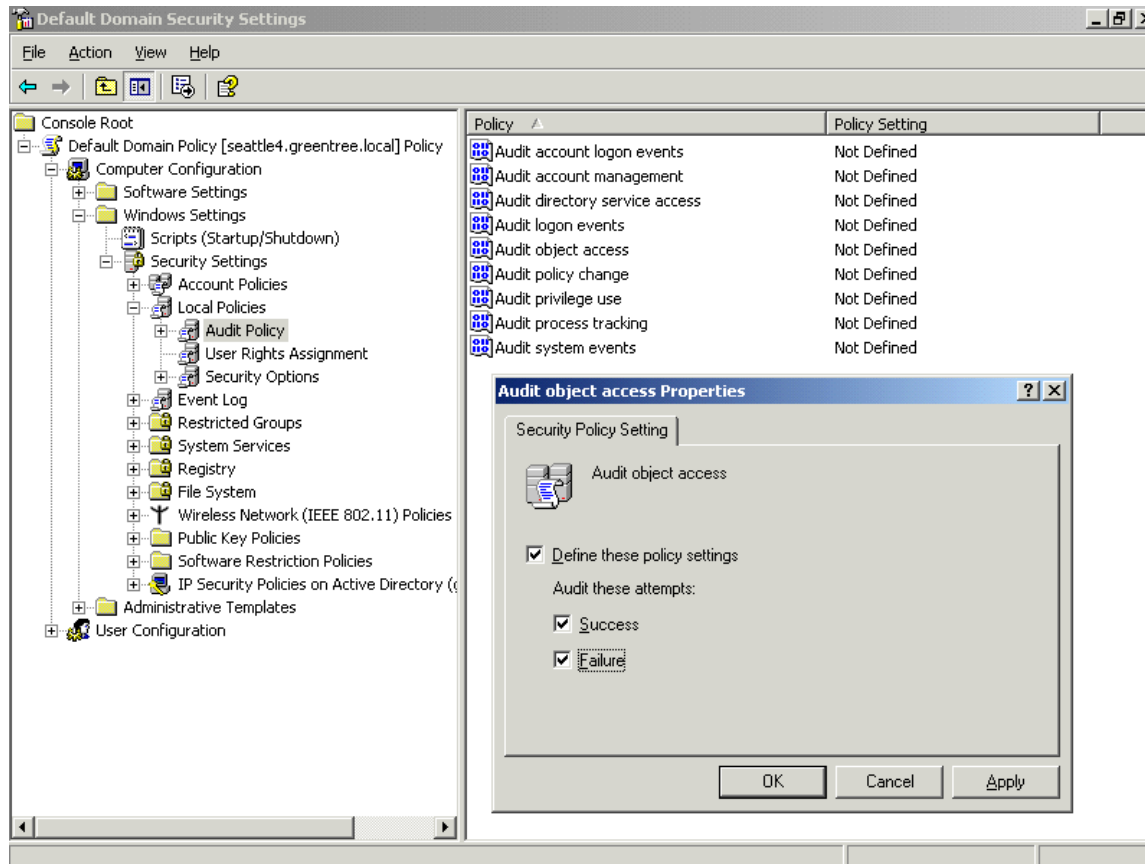
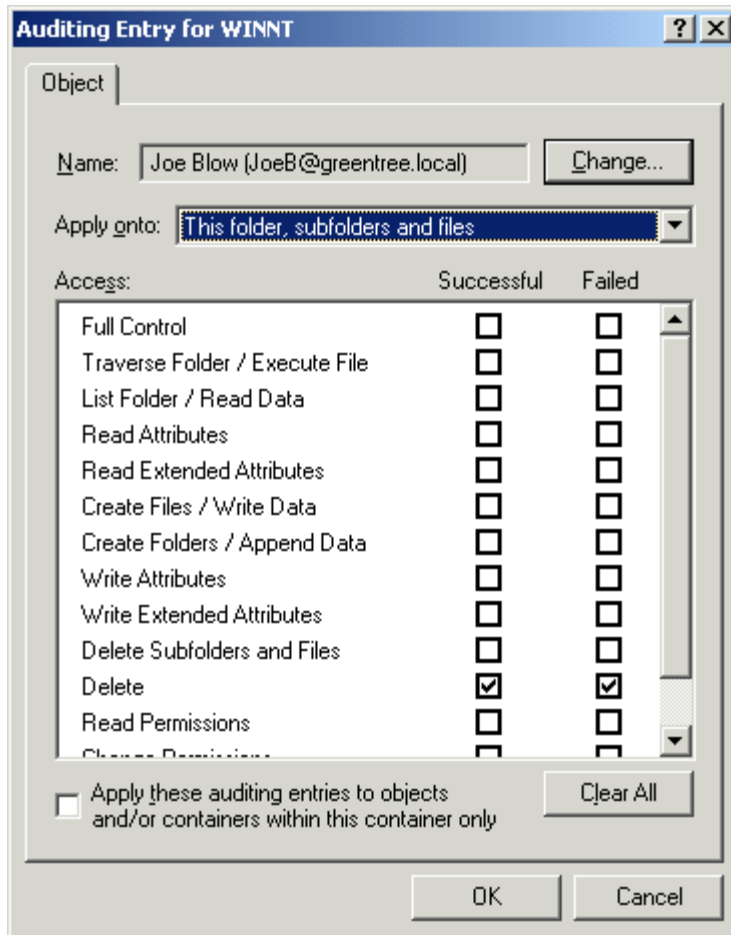


Figure 6.14: Turning on auditing for object access.

## Set SACLs on Files and Folders

SACLs are the instructions configured for auditing object access. After the audit policy has been correctly set, you must set auditing for each folder or file you want to audit. SACL settings on a folder can be inherited by the files within that folder, depending on its configuration. To set SACLs, navigate to the folder or file in Windows Explorer, right-click the object, and select Properties. Select the Security tab, click Advanced, click Auditing, and click Add to add a SACL. Enter the name of a user or group to audit, and click OK. In the Apply Onto text box, select appropriately depending on inheritance requirements. In the *Auditing Entry for* dialog box, select the access level to audit for, and select Success and/or Failure, then click OK (see Figure 6.15). Click OK again.



**Figure 6.15: Set access to audit for.**

Determine whether you want to clear the *Allow all inheritable auditing entries to propagate to this object and all child objects*. Include these with entries explicitly defined here check box, then click OK (see Figure 6.16). And click OK again to close the Properties page.

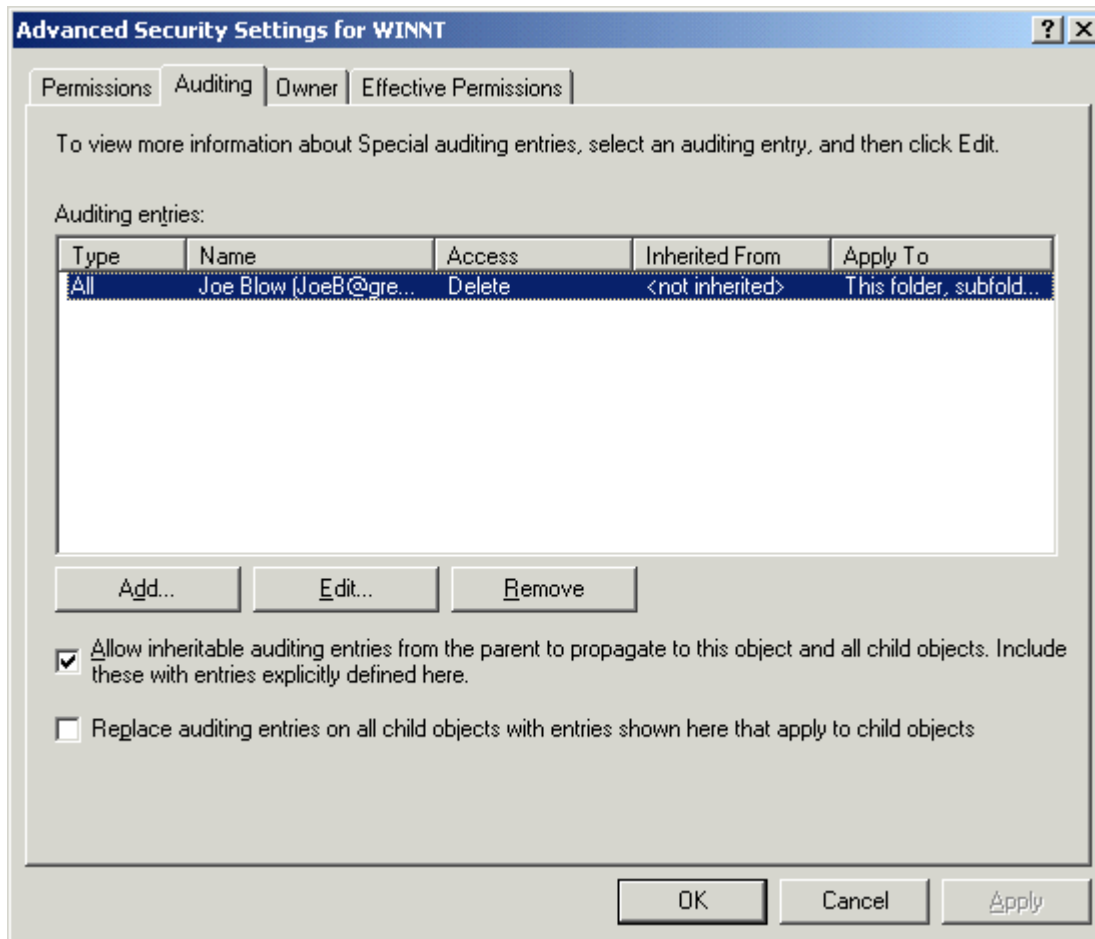


Figure 6.16: Determine changes to inheritance from above.


## SACL Inheritance

Like DACLs, SACLs can be inherited and can be blocked from subfolders. Three methods of affecting inheritance exist. First, when setting the SACL, you can choose to limit or expand its affect. When a SACL is applied to a folder, the Apply Onto drop-down box in the auditing entry page for each SACL lets you select one of the following:

- This folder, subfolders and files
- This folder only
- This folder and subfolders
- This folder and files
- Subfolders and files only
- Subfolders only
- Files only

The default setting is *This folder, subfolders and files*. Thus, all files and folders lower in the hierarchy than this folder will have the same SACL applied. If such is not your intent, you must choose one of the other possibilities.

Second, when configuring the SACL, you have the opportunity to clear the *Allow all inheritable auditing entries to propagate to this object and all child objects. Include these with entries explicitly defined here* check box. If you clear this check box, changes made to SACLs above this folder's hierarchy will not affect it. This setting is also available when setting a SACL specifically on a single file. In this way, you have granular control over which settings are applied where.

 The *Include these with entries explicitly defined here* check box for Windows .NET is different then in Win2K. In Windows .NET, new audit settings created above the file in the hierarch might either propagate to the folder and add to the settings already there, or can be prevented from having any affect. Changes made above the file or folder in the hierarchy will NOT overwrite settings made here.

## Chapter 7: Remote Access

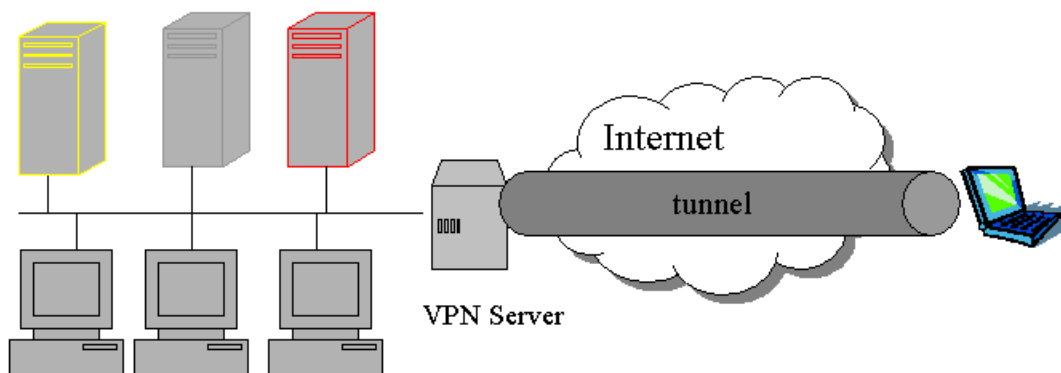
**Q 7.5: I set up a Windows 2000 virtual private network (VPN) for use by our salesmen to connect to the corporate LAN. It worked fine at first, but we had a security review, and the experts advised us to change the VPN protocol from Point-to-Point Tunneling Protocol (PPTP) to Layer 2 Tunneling Protocol (L2TP)/IPSec and change our authentication method to certificates. It seems to work in my test lab, but when I put it into production, I cannot get it to work. In addition, we must be accessible to Windows 98 clients. Will upgrading our Routing and Remote Access Service (RRAS) server to Windows .NET solve this problem?**

**A:** From what you're saying, I suspect that your environment uses Network Address Translation (NAT). As you know, NAT modifies the IP source address of all packets. Although this behavior does not cause a problem for Point-to-Point Tunneling Protocol (PPTP), your original VPN protocol, it does cause a problem for Layer 2 Tunneling Protocol (L2TP)/IPSec. In essence, IPSec sees the packet manipulation performed by NAT as tampering, and drops the packet. This behavior is not a design flaw in the Windows 2000 (Win2K) implementation of L2TP/IPSec, but rather a lack of NAT-related direction on the part of the standard, and the Win2K implementation is written to the standard. The short answer to your question about upgrading to Windows .NET is maybe. There is an emerging standard for NAT-Traversal that Microsoft has indicated will be supported by Windows .NET. However, we are talking about an emerging standard, and an operating system (OS) that, as I write this, has not yet shipped.

You should spend some time investigating this issue on three fronts. First, some non-NAT related issues of virtual private network (VPN) design have an impact on L2TP/IPSec implementations. Second, understanding the L2TP/IPSec implementation as it stands now and the problems that NAT can cause is important. If this is your problem, you will want to be able to document it. There is no sense getting in an argument over the security evaluation results. It is not always possible to implement the preferred solution, but you'll want to have valid reasons why you can't. Finally, you should understand the emerging standard for NAT-Traversal, as it might be an option you want to pursue.

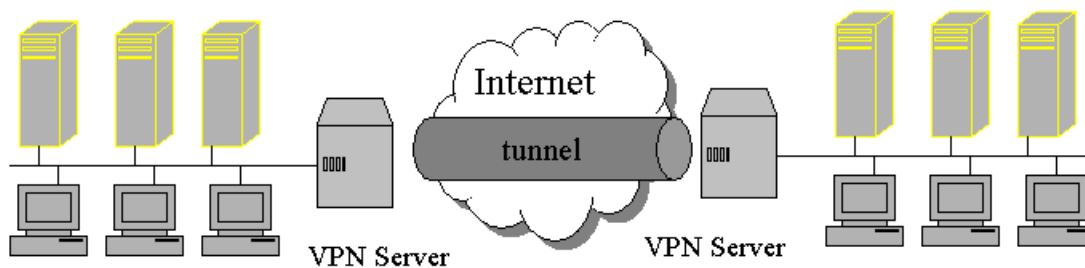
## VPN Design Issues for L2TP/IPSec

VPNs are good choices for secure communications because data is tunneled from one network to another across one or more other networks. Logically, it's as if a single tunnel connected the client directly to the server with no other devices between. Physically, such is not true, although all packets are encapsulated within the tunneling protocol, which affords some protection. Security, however, is ensured because of the data encryption and other characteristics of the tunneling protocols. Computer authentication, data integrity, and protection from replay attacks are common benefits. In addition, user authentication and access control can be enforced. Microsoft Windows .NET VPNs can exist as client-to-server VPNs and as endpoint-to-endpoint (server to server) demand-dial VPN connections. Figure 7.22 illustrates the client-to-server VPN that you have indicated is your design. Communications are tunneled and encrypted between the client and the server. Subsequent access by the client to resources on the network, although it is tunneled and encrypted between the client and the VPN server, is not tunneled or encrypted between the VPN server and the internal resource.



**Figure 7.22: Client-to-server VPN.**

Figure 7.23 shows the endpoint-to-endpoint configuration. All communications that originate in one network with a destination of another network are tunneled and encrypted between the Routing and Remote Access Service (RRAS) servers on the network boundaries. Communications between the client and the VPN server on one network—or between the VPN server on the other network and other resources—is not tunneled or encrypted.



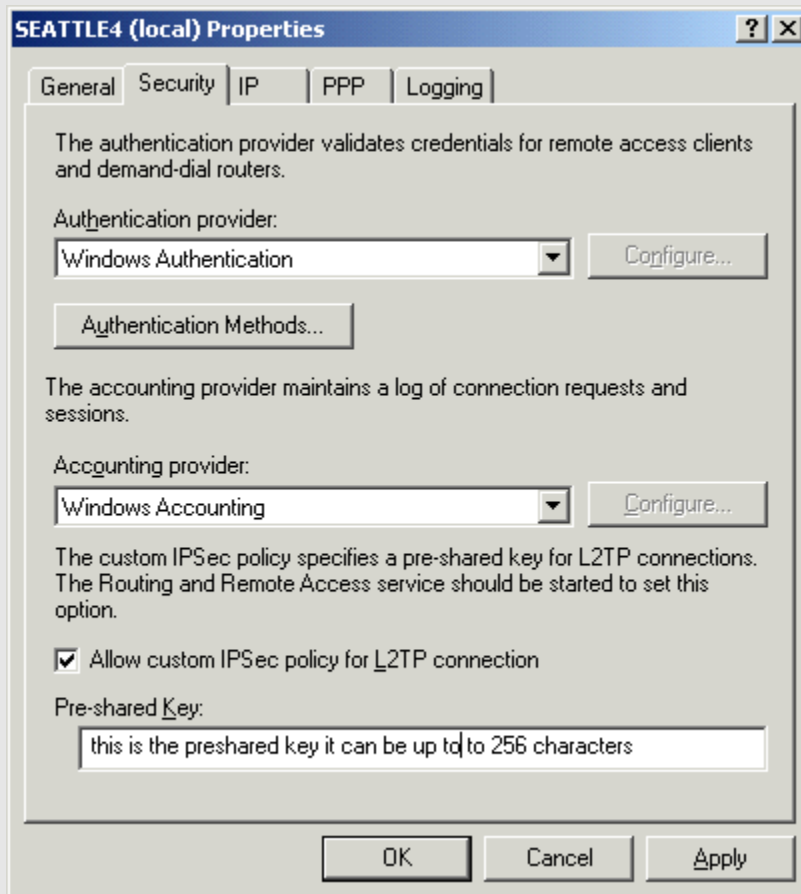
**Figure 7.23: Endpoint-to-endpoint VPN.**

Either configuration can work with L2TP/IPSec, but the use of the client-to-server model is more likely to produce unexpected results as the user might move and his or her connection might be unhampered at some locations and be blocked at others. In the endpoint-to-endpoint configuration, it should be easier to determine whether NAT is the problem.

A standalone Windows .NET RRAS server can be used as a VPN tunnel endpoint, as can a Windows .NET server that has been joined in a domain. Question 7.4 described a standalone RRAS server and available authentication methods. The advantage of using a domain member is that domain user accounts can be used for authentication and the user database does not lie on the RRAS server. One advantage to this setup is that, should the server be compromised, no useful account database information can be gained. In addition, as company needs grow, more servers can be added and user accounts do not have to change. Client computer domain membership and user accounts can also be used to push security settings to users and client machines, something unattainable with a standalone server.

You don't say whether your situation involves domain membership for the VPN server, and this information might have a bearing on your problem. Using certificates for authentication is also easier in a domain environment. Because an Enterprise Certification Authority (CA) can be used, certificates can be published to Active Directory (AD) and are then automatically available for user authentication. Computer certificates, which are required for the default implementation of L2TP/IPSec, can also be automatically published, and thus are readily available. However, there are numerous issues here. The CA must be correctly configured and protected. If it is compromised, all the certificates it has issued are compromised, and thus the identity of any client or server using these certificates is in doubt.

Although a Windows .NET L2TP/IPSec VPN can be configured to use shared keys instead of certificates, this setup is not recommended. The shared key, of which only one can be created, is visible in the configuration of both client and server and will not remain secret very long, as Figure 7.24 shows.




**Figure 7.24: The shared key is revealed.**

L2TP/IPSec can be configured for shared key instead of certificates. Such is also possible in Win2K but requires more than a GUI interface change. Although it is not recommended as an authentication measure, you should try this configuration in your situation to make sure that your problems are not certificate related. If you change nothing other than shared key configuration (don't forget to enter the same key on the client) and the clients are able to connect, your problem might be certificates, not NAT.

### VPN Protocols Choices

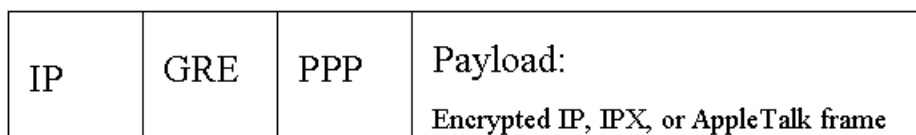
Because you are moving from PPTP to L2TP/IPSec, it's important to consider the differences between the two protocols. Both are choices for Windows VPNs. Of the two, L2TP/IPSec is considered the more secure but both have advantages and disadvantages.

 It is possible to have a non-encrypted PPTP or IPsec tunnel. By definition, VPN just means the connection of two networks across a third network. This type of a tunnel, one that does not use encryption, is not recommended. Tunneling alone affords little protection. The information about protocols provided here is very brief and introductory. One source for learning more about the protocols is through each protocol's Request for Comments (RFC), which can be found at <http://www.rfc-editor.org/rfcsearch.html>. L2TP is RFC 2661 and IPsec is RFC 2401.

## PPTP

PPTP is described is a standard that has primarily been implemented by Microsoft and has been available since Windows 98 and Windows NT 4.0. The first implementation came under public scrutiny and was strongly criticized for weaknesses in keying, authentication, and encryption algorithms. Microsoft subsequently revised the protocol, correcting these flaws. The improvements were acknowledged by the original critics, but PPTP remains flawed in the eyes of many simply because of the early criticism.

When a PPTP session is established, an IP, AppleTalk, or IPX frame is encapsulated with a GRE header and an IP header, the IP header contains the IP address of the VPN client and server. Figure 7.25 illustrates this design.

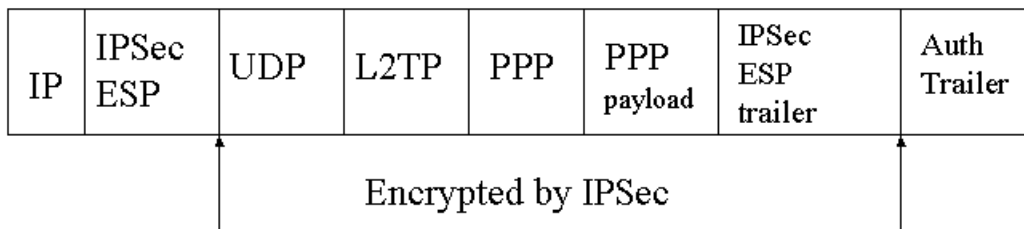


**Figure 7.25: PPTP encapsulation and encryption.**

The PPP frame is encrypted using keys generated by the MS-CHAP, MS-CHAP v2, or EAP-TLS authentication protocols. Only these authentication protocols can be used to provide an encrypted PPTP VPN solution. Microsoft Point-to-Point Encryption (MPPE) is the encryption algorithm used.

## L2TP/IPSec

If this combination is chosen for the VPN, L2TP uses IPsec for data encryption. (L2TP/IPsec is usually pronounced as L2TP over IPsec.) The L2TP encapsulation, like PPTP, works with a PPP frame but provides two layers of encapsulation. First, the PPP frame is wrapped with an L2TP header and a UDP header. Next, this message is wrapped with an IPsec header and trailer, an IPsec Authentication trailer (for message integrity and authentication), and finally, an IP header. Figure 7.26 illustrates this design. The IP header includes the source and destination address of the client and server.




**Figure 7.26: L2TP/IPSec encapsulation and encryption.**

As you can see, the entire message, exclusive of the IPSec header and trailer and the final IP header is encrypted. DES or 3DES is the encryption algorithm used.

### **L2TP over IPSec and NAT—NAT-Traversal**

One of the issues with IPSec and hence VPNs using L2TP over IPSec is the inability to use them in natted environments. In a typical scenario, a VPN tunnel is used to provide access from outside the firewall to inside by opening the ports on the firewall used by the VPN. Both PPTP and L2TP over IPSec VPNs can be configured this way—unless the firewall, router, or other remote access device, which sits between the VPN client and the VPN server, uses NAT. The current IPSec standard does not address this issue, in fact, an implementation—such as Win2K—written to the standard, sees the NAT manipulation of the addressing as tampering and drops the packets.

The problem with NAT comes about because the NAT device must translate the source address, and might assign a new source port to maintain a table to be used in routing replies back to the originating host. Here's what's happening: The NAT device modifies an outgoing packet by changing the real source address, the address of the sending client, to that of the Internet routable address provided to the NAT device. When packets from the Internet return to the NAT device, it is able to modify the destination address (which arrives using the Internet routable address assigned as the source address of the outgoing packet). How does it know the new source address to use? It knows because it keeps a table of sources addresses and ports mapped to the assigned source address and ports it replaced in outgoing packets. It is able to match the incoming packets and modify the destination address and port. However, because of the built-in security mechanisms of IPSec such tampering with the address is not allowed, hence the packets are dropped. This is why a Win2K to Win2K VPN that must pass through a NAT device can only use PPTP.

 Legacy clients, Windows 95, Windows 98, and Windows NT do not have native L2TP/IPSec ability to participate as VPN clients. However, Microsoft has recently released an L2TP/IPsec client for Windows 98, Windows ME, and Windows NT 4.0 Workstation that can be downloaded from <http://www.microsoft.com/windows2000/techinfo/howitworks/communications/remotearrress/l2tpclient/relnotes.asp>. The client is just that, a client. It will not enable any of these clients to become a server and will not install on Window NT Server. It also will not install on Win2K or Windows .NET. The new client also supports NAT-traversal.

Before any NAT-traversal can occur, the client must be capable of recognizing the use of NAT, and the server must be NAT-Traversal enabled. To fully understand the process, you should know something of how IPSec works. The following text explains, in simplified form, how

NAT-Traversal works. The client detects the NAT-Traversal capability of the server by an exchange of strings (in the draft an MD5 hash of the draft title) during the first messages of IPSec, Phase I negotiation. (Phase I negotiation of IPSec includes establishment of IKE communications and generation of master key that is used in Phase II to generate encryption keys.) If the server does not return a message that includes the hash, it is not considered to be NAT-Traversal enabled.

Next, the presence and location of a NAT device is detected by using the NAT-D payload message. To discover if NAT is being used, each side looks to see whether the IP address of the message has been modified. This is done by including a hash of the original source address and port. When received, a new hash of the existing source address and port is made. If the new hash matches the original (included in the payload), there is no NAT device in-between and processing continues per the IPSec standard. If the hash does not match, NAT is being used.

Although port 500 is the standard port for IKE traffic, IPSec-aware NAT devices may respond in a different way than standard NAT when they detect IKE traffic. Because NAT-Traversal does not include the ability to determine exactly what an IPSec-aware NAT device is doing, moving, or floating the IKE traffic to port 4500 avoids the problems that the non-standard IPSec-aware NAT device may pose. Additional modifications such as the inclusion of a non-ESP marker may also be necessary. After the initial NAT-Traversal communication is established, subsequent negotiations must start on port 4500. An illustration of the IKE floating can be found in Figure 7.27.

Source port	Destination port	length	checksum	Non-ESP marker	IKE header
-------------	------------------	--------	----------	----------------	------------

**Figure 7.27: IKE floating.**

Finally, negotiation of NAT-Traversal encapsulation occurs. Normal IPSec negotiations include the use of either Tunnel or Transport modes. NAT-Traversal requires the use of either of two new modes: UDP-Encapsulated-Tunnel and UDP-Encapsulated-Transport. Either of the modes allows NAT manipulation of the source IP and port information as this information is now available in a header designed for this purpose. Figure 7.28 is an example rendering of the UDP-Encapsulated-Transport mode.

Source port	Destination port	length	checksum	ESP Header
-------------	------------------	--------	----------	------------

**Figure 7.28: UDP-Encapsulated-Transport mode.**

---

## **Chapter 8: Security Tools, Mechanisms, and Emerging Issues**

**Q 8.5: I'd like to prevent clients on my network from using or receiving Telnet commands and from running a Web server. Can this be done?**

**A:** There are two ways to lock down these services. First, the services can be disabled on the workstation and in Group Policy. A reapplication of Group Policy will reset the services to disabled, should they have been improperly enabled. A second and more flexible way is to use and IPSec blocking policy.

### ***IPSec Blocking Policy Basics***

In many discussions of IPSec, very detailed information is given about creating negotiation policies to control and keep private communications between two computers. However, IPSec can also be used to create other types of policies. These policies do not engage in any negotiation, nor do they encrypt anything. These policies are written to either accept or block certain types of packets. They can be used to prevent communication from being passed up the stack for further processing, or to prevent communication from leaving the computer. They can also be used to allow communication, and often allow and blocking policies are used together for special effect. For example, you might deny all Telnet communication with a computer, but allow Telnet access from a particular computer.

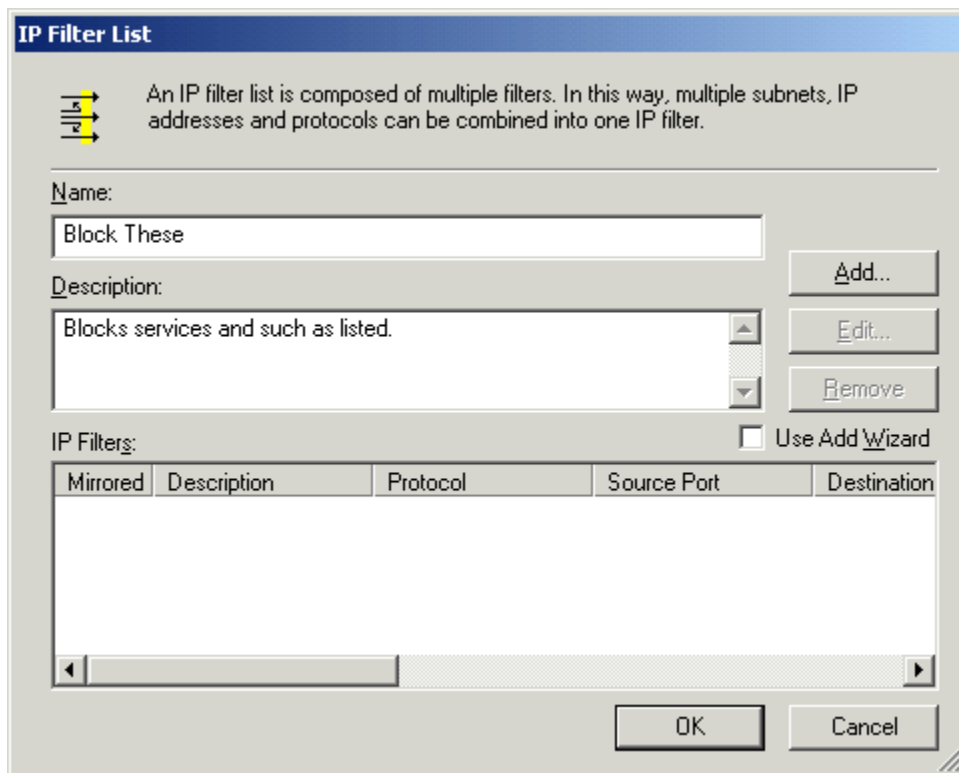
Blocking and allowing policies can be written by writing filters for specific protocols and ports or to block or allow traffic to or from an IP address, a range of addresses, DNS, WINS, or DHCP servers. To learn how to write these types of policies you should first start with a simple one. Your request is perfect for this introduction.

### ***Creating and Testing the Policy***

To test your understanding of this option, create the following policy on a single computer in your test lab that has IIS installed. You will use the Local Security Policy of the computer to create the policy. To test it, make sure that the Telnet service on the test server computer is not disabled. Use a test client computer to Telnet to the test computer, then end the session. Use this client to browse to the Web site on the test computer, assign the IPSec blocking policy as instructed later, and use the client to attempt to Telnet to the test computer. You should be blocked. Use the client to browse to the Web site, you should be blocked. Unassign the policy, and retest Telnet and the Web site. You should be successful.

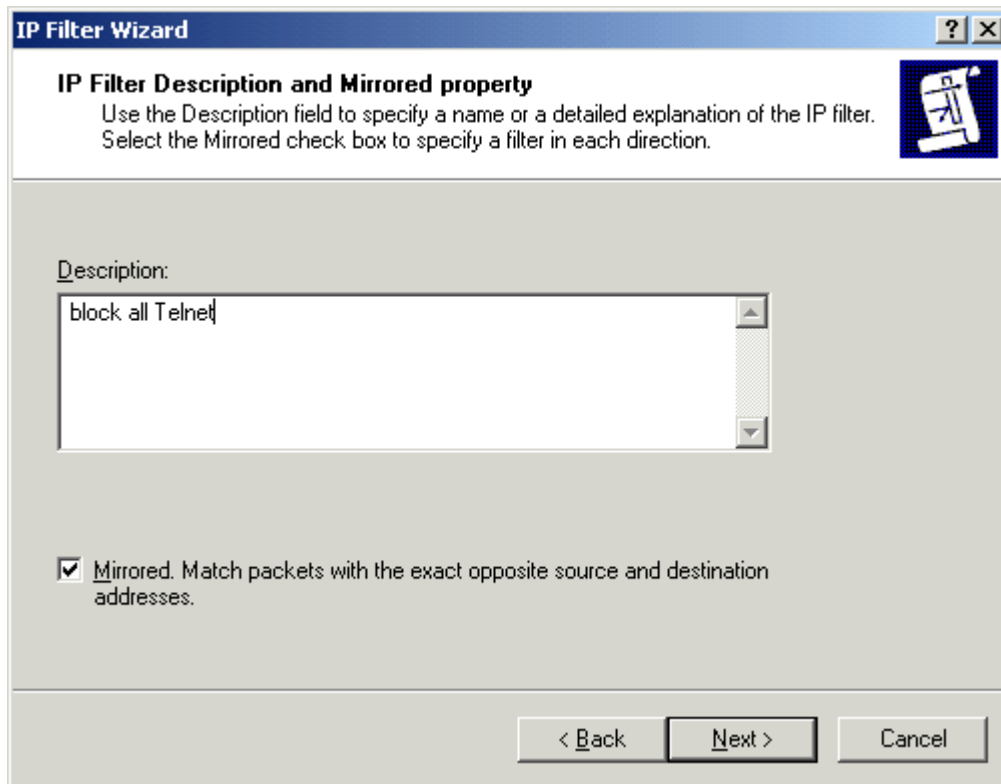
When you have completed a successful test, you can export the policy (right-click the IPsec Policies on local computer, and select Export policies to store the policies in a file). You can then import the policy to a Group Policy Object (GPO) you have created on the appropriate organizational unit (OU) in your domain. This OU should only have computer accounts for the computers that that you want to block these protocols for. To create the policy, create a console and add the Group Policy Snap in. Navigate to the Local Computer Policy, Computer Configurations, Windows Settings, Security Settings, IP Security Policy on Local Computer container, and select Create IP Security Policy. Click Next on the Welcome screen, then enter a name for the policy and a description, and click Next. Clear the *Activate the default response rule* check box. The default rule responds to IPsec requests from other computers. Our blocking rule has no need to do that. Click Next, then click Finish to close the wizard.

On the Rules page, clear the Add Rule Wizard check box. The wizard steps through configuration of the tunnel, authentication, and other properties that are not used in a blocking policy. Click Add to add a rule, and on the IP Filter List page of the New Rule properties, click Add to add a filter. Enter a name and description for the filter list, as Figure 8.13 shows, then click Add to add a filter.



**Figure 8.13: Naming the Filter List.**

Enter a filter description. If this filter is not to be mirrored, clear the Mirrored check box (see Figure 8.14). Click Next. A mirrored filter also matches packets with the exact opposite source and destination address. In this first example, we are writing a filter to block Telnet traffic initiated in either direction, so mirroring is appropriate.

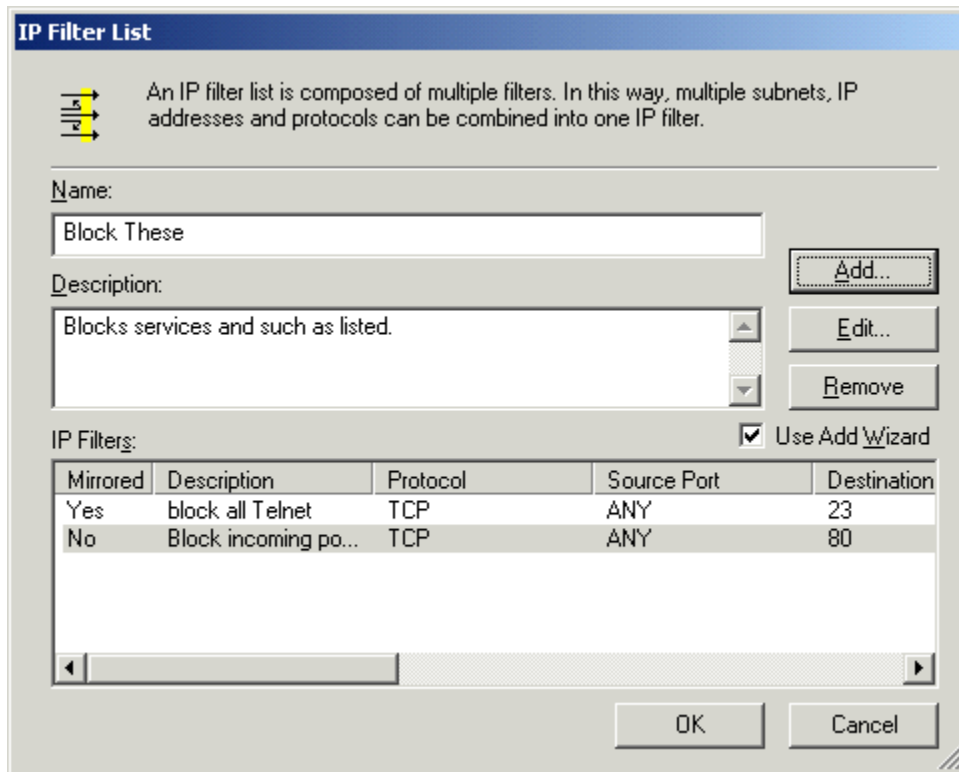


**Figure 8.14: Determine whether a filter should be mirrored.**

The source address is indicated as *My IP address*; change it to *Any address*, then click Next. The destination address is indicated as *Any IP address*, change it to *My IP Address*, then click Next. Use the drop-down box to select the TCP protocol, then click Next. Leave the *From any protocol* check box selected, and select the *To this port* check box, then enter 23 in the text box below, and click Next. Select the Edit Filter check box, and click Finish to end the wizard.

Examine the filter. You should have a filter that identifies any packets from any address going to the host computer on port 23. Because the filter is mirrored, it will also identify any packets from the host computer going to port 23 on any other computer. Click OK to close the filter properties pages.

Repeat this process, except this time you will create a non-mirrored filter to identify packets coming from any IP address and any port to this host on port 80. This filter will not be tripped by any packets leaving this computer to port 80 on another computer. This computer can be used to access Web sites. At this point, you should have two filters in the filter list (see Figure 8.15). Click OK to close the filter list.



**Figure 8.15: The filter list.**

On the IP Filter list property page, select the filter list you have created, select the Filter Action page, and click Add to add a filter action. Click Next on the welcome page, then enter a name for the action and a description, and click Next. Select block, click Next, select your filter action, and select the Connection page (notice that *All network connections* is selected by default). You could, instead, decide to only block these protocols if they were on the local LAN or from a remote source. Click Close to return to the Rules page, and click OK to close the policy. To activate the filter, right-click it in the details pane, and select Assign.