How to Install SSL Certificates on Microsoft Servers

Realtime publishers

Dan Sullivan

Copyright Statement

Chapter 4: Installing SSL Certificates in Exchange Server, SharePoint, and SQL Server

SSL certificates are often associated with Web servers such as Microsoft IIS, but they are actually used in a variety of Microsoft applications, including Microsoft Exchange email server, Microsoft SharePoint collaboration server, and the Microsoft SQL Server database. The process of installing an SSL certificate has both common and application-specific steps across these applications. This final chapter discusses how to install a SSL certificate in Microsoft Exchange Server, Microsoft SharePoint Server, and Microsoft SQL Server. We begin with a quick overview of the common parts of the installation process, then discuss each application in more detail.

Common Operations

As you may remember from earlier chapters, the Certificate Store is used to manage SSL certificates for applications, individuals, and trusted authorities.

Step 1: Prepare the Microsoft Management Console

The Microsoft Management Console (MMC) provides a snap-in (see Figure 4.1) for managing certificates on a server (see Figure 4.2).

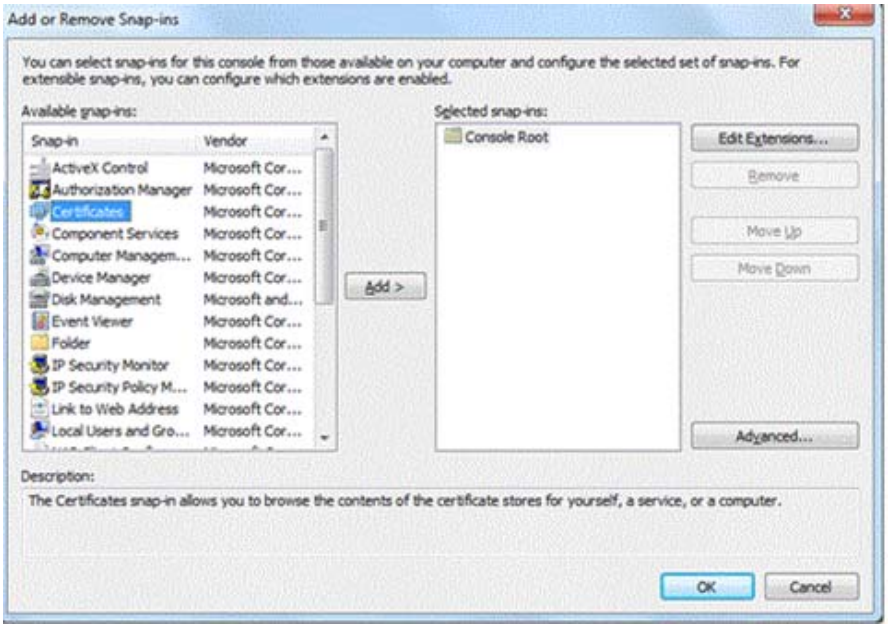

Figure 4.1: The Certificate Store is managed through the Microsoft Management Console.

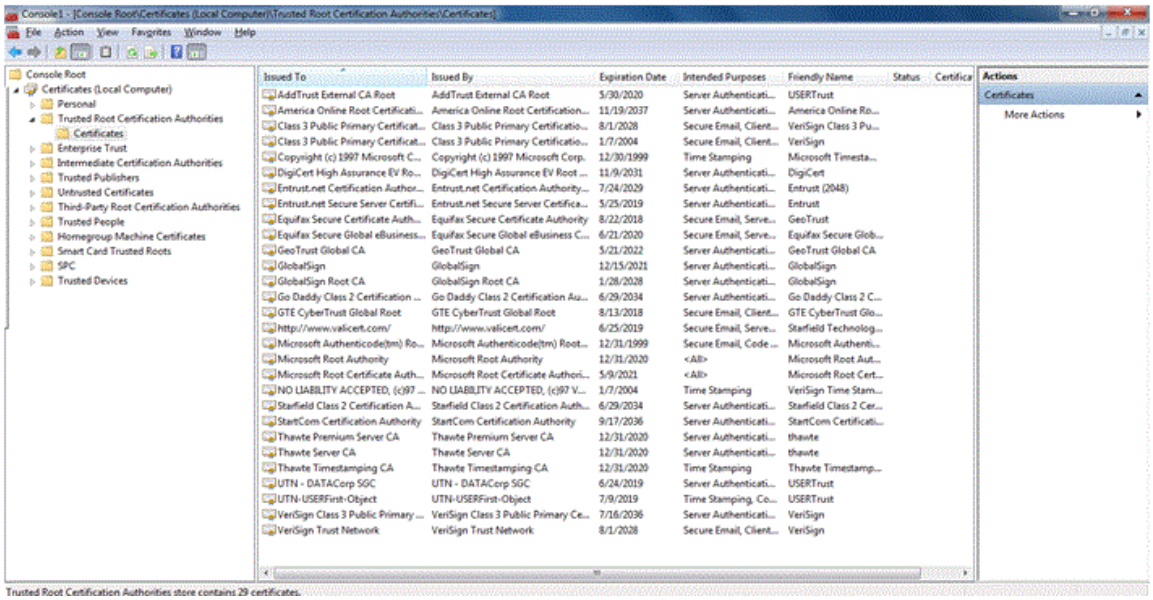

Figure 4.2: The Certificate snap-in to the MMC is the primary method for managing certificates on a server.

Step 2: Acquiring an SSL Certificate

Once the Certificate Store has been set up on a server (see Chapter 2 for details), we need to acquire an SSL certificate for the application we are working with. We can use a certificate from a trusted third-party provider or we can generate a self-signed certificate. As noted earlier, the former is appropriate for production servers that will be used by the public or business partners outside of the organization; the latter is appropriate for testing and development.

Certificates from Trusted Third-Party Providers

When using a third-party provider, generate a certificate signing request (CSR). This is an encrypted set of data about the server that will use the certificate. The CSR includes information such as:

- The fully qualified domain name (FQDN) of the server
- Organization
- Organization unit (OU)
- Location information, such as city, state, and country
- Email address for a contact at the company requesting the certificate
- A public key that will be included in the certificate

CSRs can be generated in a number of ways. Many applications that regularly use SSL certificates provide a way to generate certificates.

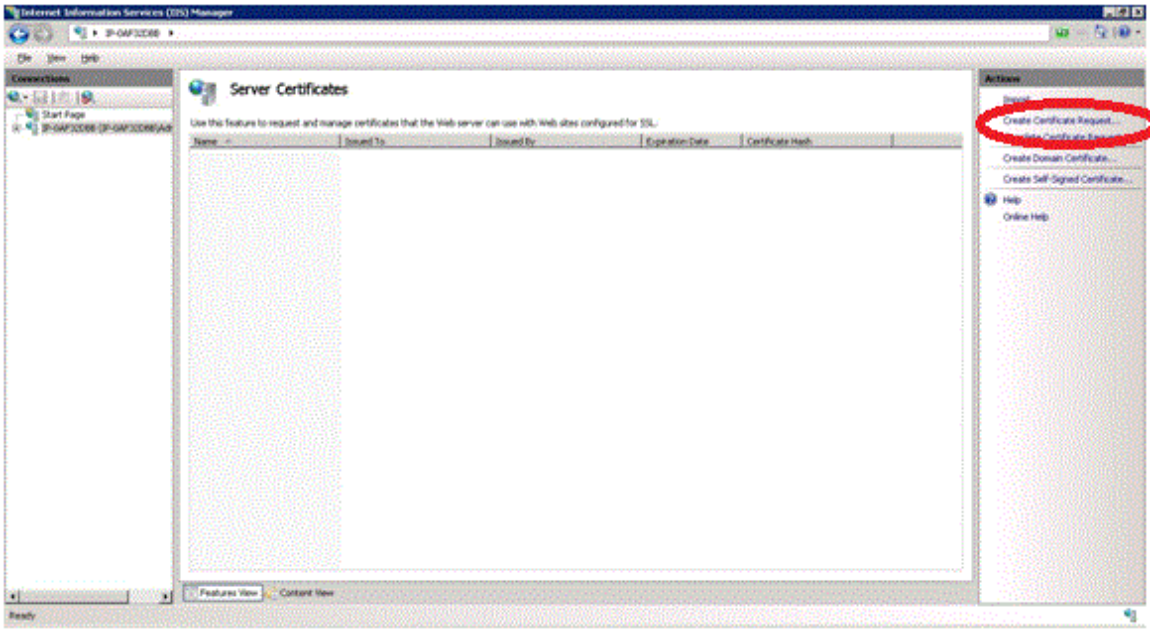

Figure 4.3: Enterprise applications that use SSL certificates provide features to generate CSRs.

Once you have created a CSR, you send it to the trusted third-party provider, which will generate and send to you an SSL certificate in the form of a file. You can import that file into the Certificate Store using the method described in Chapter 2.

Generating a Self-Signed Certificate

If you prefer to work with a self-signed certificate, there are a couple of ways to do so. You can use an application-provided feature to generate a self-signed certificate.

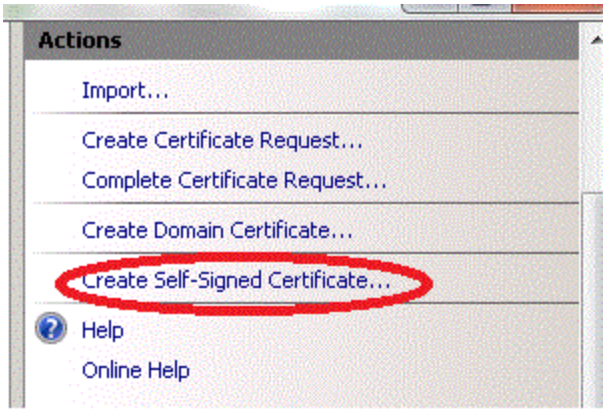

Figure 4.4: Enterprise applications sometimes provide a feature that generates a self-signed certificate, as in Microsoft IIS.

Another option for generating a self-signed certificate is to use a command-line program, such as MakeCert.exe from Microsoft. MakeCert.exe is a program included in the .Net Framework and the Windows SDK.

> **Resource**
> The Windows SDK can be downloaded from http://msdn.microsoft.com/en-us/windowsserver/bb980924.aspx.

Making a certificate is as easy as issuing a simple command:

```
makecert testCert.cer
```

Of course, command-line options are available to specify details of the certificate.

> **Resource**
> Information about options to the MakeCert.exe program can be found at http://msdn.microsoft.com/en-us/library/bfsktky3(v=vs.80).aspx.

Another option for generating self-signed certificates is the Open SSL package. This package may be especially useful in heterogeneous environments that run both Windows and Linux servers. The commands are similarly simple in OpenSSL. For example, to generate a CSR, use a command such as:

```
openssl req -new
   -key srv_private_key.pem
   -out srv_cert.csr
```

A self-signed certificate can then be generated with the command:

```
openssl req -new
   -x509
   -key srv_private_key
   -out srv_cert.pem
   -days 365
```

Resource

Information on OpenSSL and related downloads is available at http://www.openssl.org/.

Figure 4.5 provides a summary of common steps to installing an SSL certificate in an enterprise application.

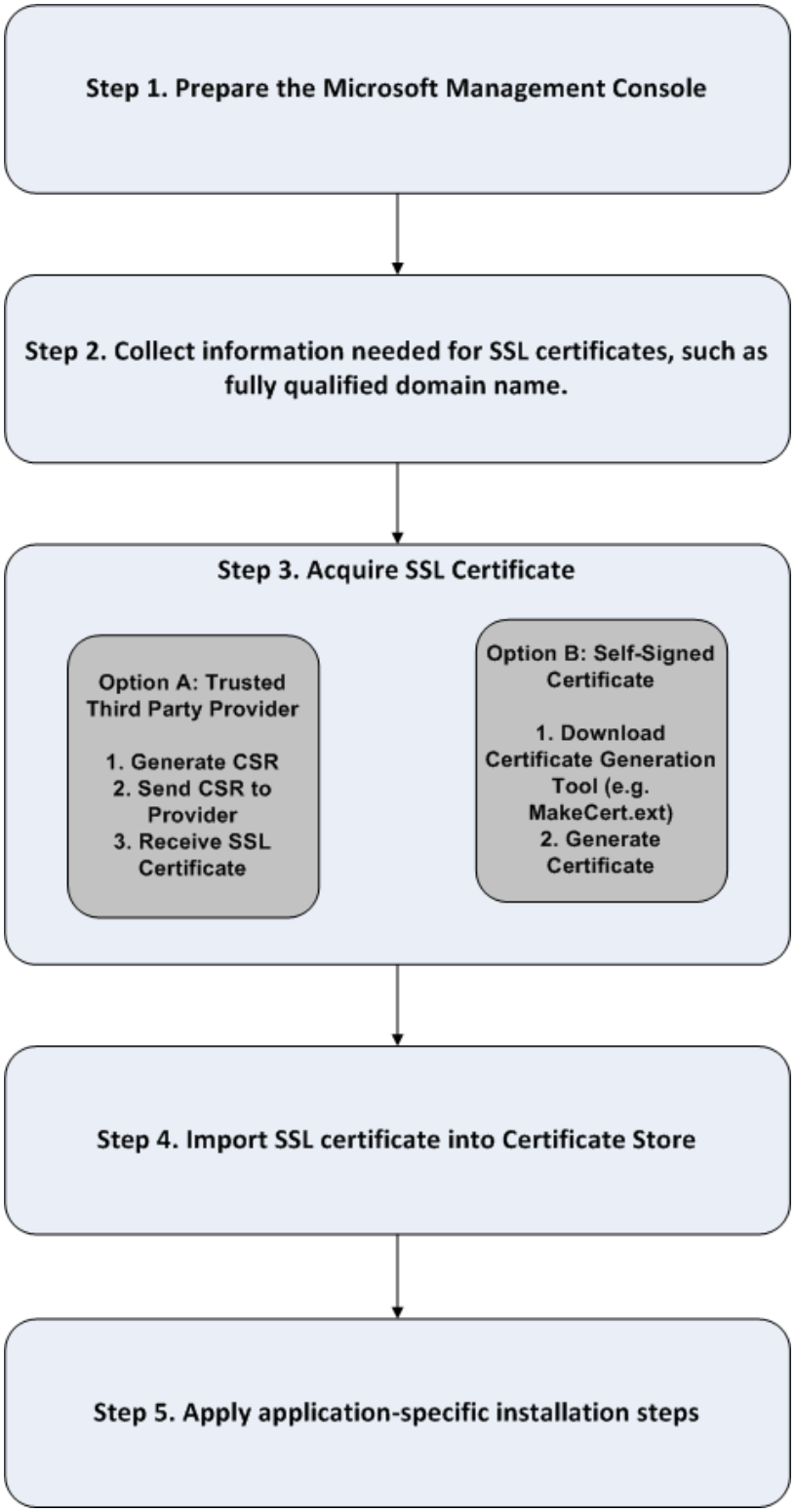

Figure 4.5: Steps to acquiring and installing SSL certificates for enterprise applications.

Once you have a certificate, it is time to configure your application to use it.

Installing SSL Certificates in Microsoft Exchange Server

Microsoft Exchange Server is a widely used email server and a logical candidate for supporting encrypted communications. In this section, we will discuss the need for SSL in Exchange Server, how to install an SSL certificate in Exchange Server, and how to verify an SSL certificate in Exchange Server. Let's start with the motivation for using SSL certificates in email servers.

The Need for SSL in Exchange Server

Email is commonly used to share private and confidential information. Without encrypting email traffic, there is a risk of outsiders capturing an email conversation and compromising confidentiality. From a technical perspective, encryption is needed for multiple services provided by email servers:

- Authentication—Passwords should be encrypted when passed over an unsecure network such as the Internet
- Message transmission—Email messages should be encrypted to prevent a breach of confidentiality and privacy
- Message storage—There may be a need to save message folders in encrypted form to prevent a privacy breach from someone with access to files storing messages; for example, a backup administrator should not have the ability to read private messages

From a strategic perspective, encryption is required so that email users have assurances that their communications will be kept private. Without this, there is a risk of undermining the use of email and forcing potential users to employ slower, most costly, and possibly less secure means of communication.

Is an Extended Validation SSL Certificate Right for You?

Now that we are discussing trust, it is a good time to consider the use of an extended validation (EV) SSL certificate. As with conventional SSL certificates, an EV certificate supports secure communication and can be used to authenticate the person or server in possession of the certificate. However, with an extended SSL certificate, a trusted third-party provider performs a greater degree of due diligence in verifying the identity of the business requesting a certificate.

> **Resource**
>
> For more information about EV SSL certificates, see *The Shortcut Guide to Extended Validation SSL Certificates*. This book provides background on the benefits provided by EV SSL certificates and the business drivers for using them.

Acquiring and Installing an SSL Certificate in Microsoft Exchange Server

As noted earlier, many enterprise applications support the ability to generate requests for certificates or self-signed certificates. In Microsoft Exchange 2010, you can follow these two sets of steps: acquiring a certificate and installing a certificate.

Acquiring an SSL Certificate in Microsoft Exchange 2010

Generating a CSR in Microsoft Exchange involves several more steps than does generating a CSR for Microsoft IIS; the reason is the number of components within Microsoft Exchange. The basic steps are:

1. From the Start menu on the server, go to the Microsoft Exchange 2010 group and select the Exchange Management Console program.
2. Click on Server configuration in the left panel.
3. Select New Exchange Certificate.
4. Enter basic certificate information, such as a friendly certificate name.
5. Secure client access server by specifying domain names for Outlook Web App and internal and Internet servers.
6. Enter the domain name for the ActiveSync domain.
7. Specify whether Exchange Web Services, Outlook Anywhere, and Autodiscover are enabled by selecting the appropriate check boxes.
8. To secure email communications, select the Use Mutual TLS to help secure Internet email option in the Hub Transport server dialog box.
9. Specify the name of the Hub Transport server.
10. Verify the names of services.
11. Provide organization and server information, such as name and location.
12. Click New in the final dialog box to generate the CSR.

Installing an SSL Certificate in Microsoft Exchange 2010

The first step to installing an SSL certificate in a Microsoft Exchange Server is to copy the certificate sent from the trust third-party provider or generated on site to the Microsoft Exchange Server. The following steps are required to install the certificate:

1. From the Start menu on the server, go to the Microsoft Exchange 2010 group and select the Exchange Management Console program.
2. Once the Exchange Management Console starts, select Manage Databases.
3. Select Server Configuration.
4. Click the certificate in the Exchange Certificate section.
5. Click the Actions menu, and select Complete Pending request.

6. From the file browser, navigate to the certificate file and select open.
7. Click Complete and Finish.
8. Click the Actions menu and select Assign Services to Certificate
9. Click the server name in the list.
10. Select services and assign them to the certificate.
11. Click on Finish

Microsoft provides the Test-OwaConnectivity cmdlet in Microsoft Exchange Management Shell, which can be used to verify the installation of Microsoft Exchange in general and secure communications in particular. The cmdlet is run from the command line. A basic test of https on the server exchange1.dspragtech.com is:

```
Test-OwaConnectivity –URL:https://exchange1.dspragtech.com
      -MailboxCredential:(get-credential dspragtech/john1)
```

This test would https: on the server exchang1.ldspragtech.com using the login credentials for the user john1.

There have been some problems in the past with unexpected error messages generated by Test-OwaConnectivity. For example, login attempts sometimes failed in Microsoft Exchange 2007 and required Update Rollup 7 for Microsoft Exchange 2007. There were even some cases in which the logins failed. Microsoft corrected that problem by requiring Update Rollup 9 for Microsoft Exchange. The moral of the story is, if you run into problems testing connectivity and you are sure the configuration is correct, the next step should be to verify you have the latest service packs and hotfixes installed.

Resources

Check out the Microsoft Exchange support page at http://support.microsoft.com/ph/13965 for more tips and tools. See especially the Tools and Security options on that page. For details, see the Microsoft Technet description of TestOwaConnectivity at http://technet.microsoft.com/en-us/library/aa997682.aspx.

For more information about past problems with Test-OwaConnectivity, see http://support.microsoft.com/kb/954213 and http://support.microsoft.com/kb/968224.

The following list provides a summary of the high-level steps for acquiring an SSL certificate for Microsoft Exchange:

- Start Microsoft Exchange management console.
- Select new exchange certificate, and enter basic certificate information
- Specify domain names for Outlook Web App.
- Set up services such as Exchange Web Services, Outlook Anywhere, and Autodiscover.
- Configure Hub Transport.
- Specify organization and server information.

The high-level steps to assign an SSL certificate are:

- Start Microsoft Exchange management console.
- Select server configuration.
- Select complete pending request.
- Assign services to the certificate.

Further Considerations for Using SSL Certificates with Microsoft Exchange Server

Enterprise Microsoft Exchange deployments can require a somewhat complex architecture. The reason is that Microsoft Exchange has been designed to maintain adequate performance levels while scaling to a large user base. The common way to deal with the need for scalability (as well as reliability) is to distribute the workload over multiple servers. In the case of Microsoft Exchange, distribution of workload has been organized around several roles that can be run on different servers:

- Mailbox role for managing mailboxes, folders, and calendars
- Client access role for supporting Outlook Web Access, Microsoft ActiveSync, Outlook Anywhere, and some email-related protocols
- Hub transport role supporting message transport, journaling, and some security services
- Edge server role for routing external traffic; supports some security services
- Unified messaging role for integrating email with voice and fax services

While supporting scalability, the option of running role services on multiple services can add to system management overhead. Of particular importance to this discussion are the implications for SSL-secured communications. Fortunately, a specialized type of SSL certificate, known as a Subject Alternative Name (SAN) SSL certificate can help reduce some of the management overhead.

A SAN SSL is designed to support multiple servers using a single certificate. The basic idea behind a SAN SSL certificate is that multiple servers can be listed in a single certificate. For example, if your Microsoft Exchange deployment requires several servers, say one for each of the five roles listed earlier, you could secure these with five separate SSL certificates or with a single SAN SSL certificate.

Most of the major browsers in use today, including Internet Explorer, Mozilla Firefox, Opera, and Apple Safari, support SAN SSL certificates. When the browser is working with an SSL-based connection, it can authenticate a server in a few ways:

- The host name of the server is the same as the common name in the SSL certificate
- A wildcard pattern, such as *.domainname.com, matches the common name in the SSL certificate
- The host name of the server matches one of the host names listed in the Subject Alternative Name field in the SSL certificate

SAN SSL certificates work well in the Microsoft Exchange environment and Microsoft recommends their use as a best practice. A potential problem with SAN SSLs is that you forget to include one or more of the server names in the CSR. Fortunately, the CSR wizard in Microsoft Exchange 2010 is designed to help avoid this problem by collecting information about which services you want to include in the certificate. It uses this information to make sure all the needed servers are included in the CSR.

Next, we will consider a similar process for installing SSL certificates in SharePoint servers.

Installing SSL Certificates in SharePoint Server

SharePoint server is a collaboration portal. Documents, calendars, images, and other potentially confidential and private documents are stored and exchanged in this application. Secure communications are important here just as they are in Microsoft Exchange.

Why Use SSL Certificates in Microsoft SharePoint?

Microsoft SharePoint, as the name implies, is a collaboration application. The name of the game is making content easy to find, access, and update. The problem is we do not want just anyone to view or update our content. We can mitigate the risks of someone tampering with our SharePoint site in a number of ways, the most important being access controls and encrypted communications. Access controls will keep unauthorized users from viewing or revising content when it is within the Microsoft SharePoint database but not when the information is transmitted. That is when SSL-secured communications are required.

We can easily fool ourselves into a false sense of security with assumptions like "With all the data on the Internet, what are the chances mine would be targeted?" or "My data does not require secure transmission because it's not that important." It only takes a single disgruntled ex-employee to target a business and cause substantial harm. (For proof of this, browse the database at http://www.justice.gov/criminal/cybercrime/ for employees that have targeted their employers.) As for cases where secure transmission may not be required, consider some of the regulations that may apply to your business:

- Sarbanes-Oxley, or SarBox, which requires publically traded companies to protect the integrity of their financial reporting. Companies that use Microsoft SharePoint to store and transmit spreadsheets or other documents used to compile financial reports should consider the risk of exposing information or having it tampered with during transmission.
- HIPAA, which governs the protection of private healthcare information. Included in the HIPAA directives are requirements for protecting private information. Transmitting unencrypted private health information may not meet the letter or spirit of the law.
- Various privacy regulations, from trans-national regulations such as the European Privacy directive to state-level regulations such as those in California and Massachusetts in the United States, which specify protections for personally identifiable information.

You may also be confident that none of the information stored in your Microsoft SharePoint repository is subject to regulations or would be of value to a former employee. That is good news for you, at least in the short term. We cannot always anticipate how applications and services will be used in the future. What starts today as a basic collaboration tool may store strategic information next year. Some security measures are cumbersome and businesses will avoid them because they interfere with business. That is understandable and is a matter of balancing costs and benefits. SSL-based security is one of the technologies that puts little burden on the user but provides more protection against the risk of tampering and disclosure than non-SSL based communications. Fortunately, the process of setting up SSL-secured communications is not unreasonable.

Setting Up SSL in Microsoft SharePoint

SharePoint Server2010 has a multi-step installation process supported by the installation wizard. Once SharePoint is installed, though, additional steps are required to install an SSL certificate.

Installing SSL certificates in SharePoint Server follows a similar pattern to other SSL certificate installations. In fact, the steps should be familiar to anyone who has worked with IIS:

1. Open the IIS Manger for the Web server for SharePoint.
2. From the Central Administration Web site, right-click and select Edit Bindings.
3. Click Add to display the Add Site Binding dialog box.
4. Select https from the drop-down box and the name of the certificate from the SSL certificate drop-down box. (This assumes the certificate has been installed in the Certificate Store as described in Chapter 2).
5. If an http binding is listed in the Site Bindings dialog box, remove it to use only https.
6. From the IIS Manger, select SharePoint Central Administration in the left panel, under the Sites folder.
7. Open the SSL Settings (double-click the icon in the central pane).
8. Select the Require SSL and Require 128-bit SSL options.
9. You can select "Ignore" under Client Certificates if you are not concerned with authenticating clients.
10. Click Apply to save the settings.

To verify the certificate is correctly installed, open a browser and navigate to the SharePoint site using https as the protocol in the URL. This should generate an error. Then browse to the page by specifying https:// in the URL; the SharePoint page should be displayed correctly.

Next, we will consider the steps required to install an SSL certificate in Microsoft SQL Server.

Installing SSL Certificates in Microsoft SQL Server

Databases are repositories for a wide variety of enterprise information. Much of that information is confidential and in some cases subject to regulations governing how securely it should be stored and transmitted. SQL Server is Microsoft's relational database, and like many other enterprise applications from that vendor, it supports the use of SSL certificates to improve the security of communications between the database server and client devices. Again, we assume you have purchased or generated an SSL certificate for use with your SQL Server database.

The installation process starts with the SQL Server Configuration Manager, which is available in the SQL Server 2008 group under the Start menu. After the SQL Server Configuration Manager starts, click the SQL Server Network Configuration option in the left panel.

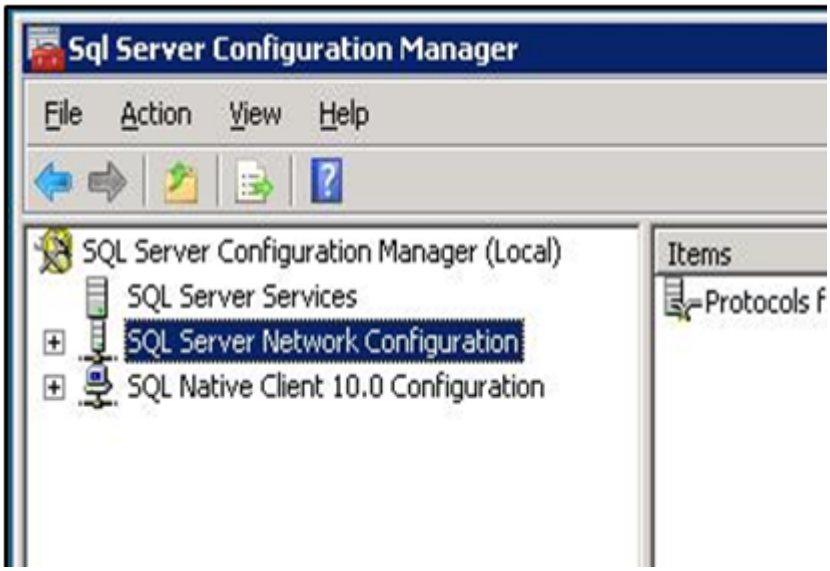

Figure 4.6: Select SQL Server Network Configuration from the SQL Server Configuration Manager.

Click on the Protocols for SQLExpress in the main panel. This will open a protocols dialog box (see Figure 4.7).

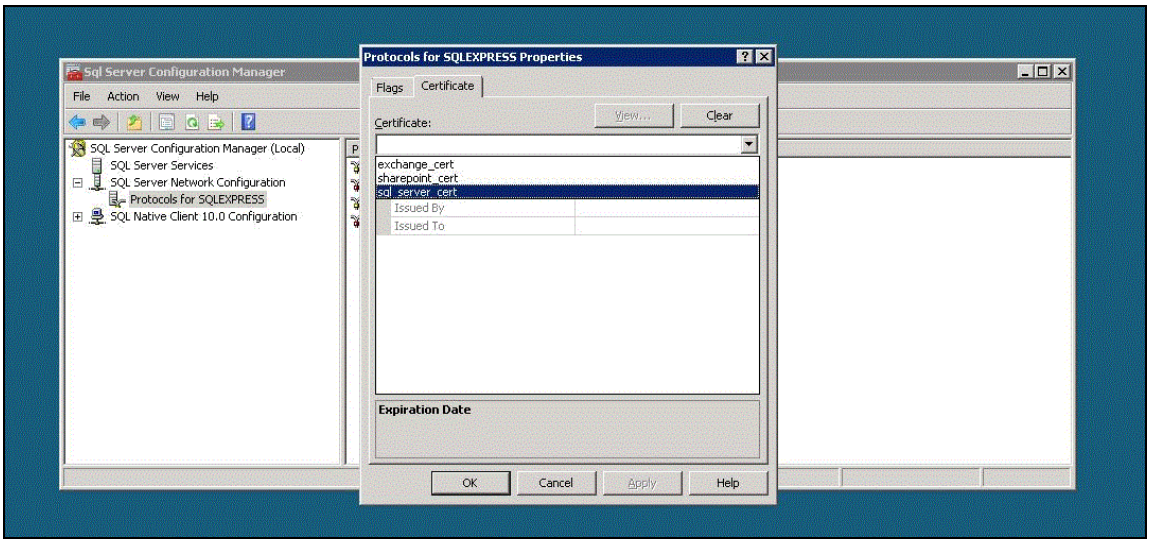

Figure 4.7: The Network Protocols dialog box allows you to select SSL certificates to use with this database.

In the Protocols dialog box, click the Certificates tab. Near the top is a drop-down box listing certificates stored in the Certificate Store. The list in Figure 4.7 corresponds to the certificates in the Certificate Store shown in Figure 4.8.

Figure 4.8: Certificates listed in the Certificate Store are available in the Protocols dialog box in SQL Server.

Select the certificate you want to install on this database server.

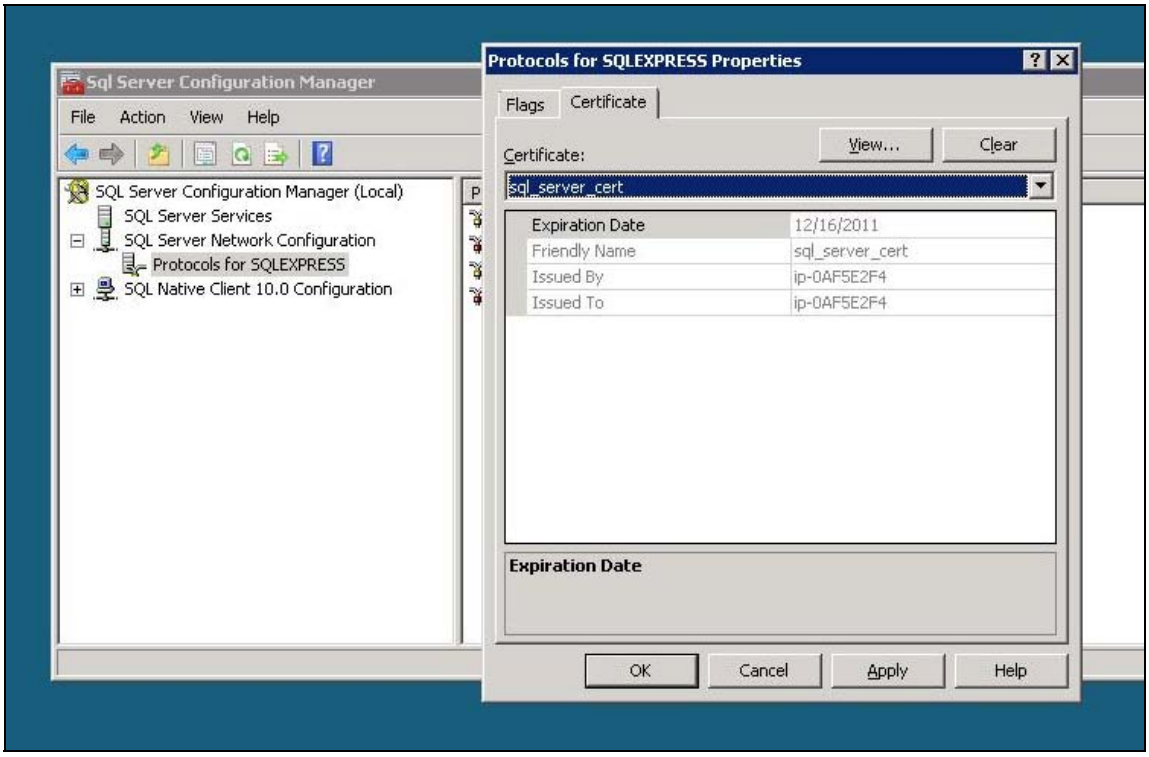

Figure 4.9: When a certificate is selected, the dialog box will display basic information such as issuer and expiration date.

After the certificate is selected, click the Flags tab to continue with the configuration. The Flags tab includes options for forcing encryption. Choose this option to ensure that communications between the database server and clients is encrypted. This is especially important if you will have clients accessing the server via the Internet. By setting the Force Encryption option, you will require an encrypted communications channel.

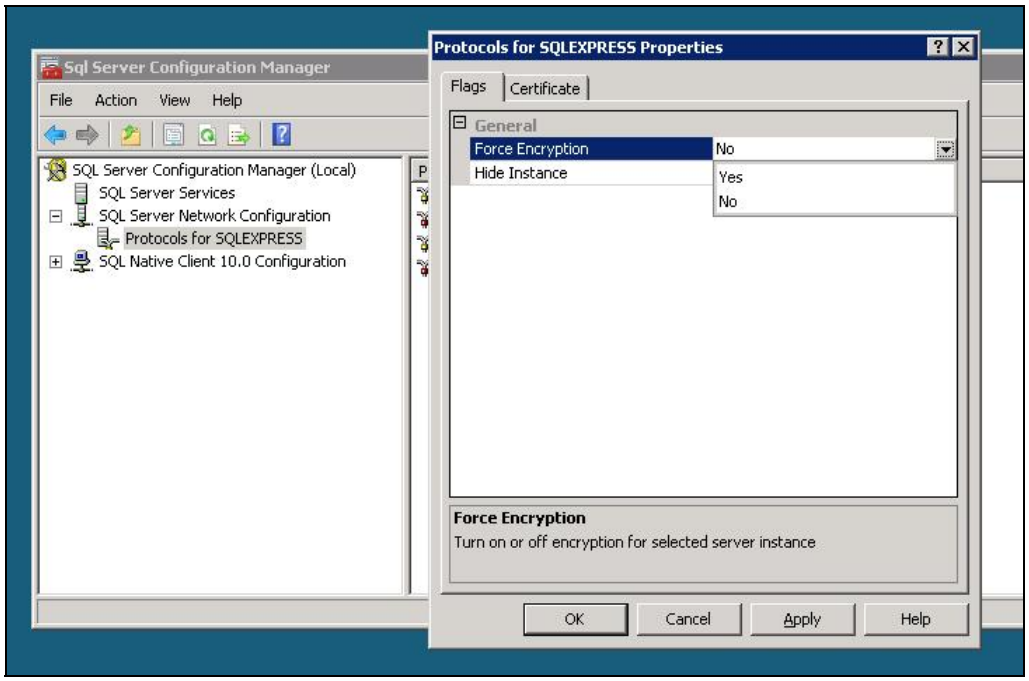

Figure 4.10: The Protocols dialog box allows you to specify that encryption is required for client/server communications.

After selecting the appropriate options on the Flags tab, click Apply. If you receive a message, click OK and restart the database service.

Summary

SSL certificates are useful for improving the security of many enterprise applications. Microsoft Exchange, Microsoft SharePoint Server, and Microsoft SQL Server can all take advantage of the benefits of SSL certificates. Fortunately for Windows systems administrators, there is a good amount of overlap between these applications when it comes to installing SSL certificates. Of course, each application has specific requirements as well. These tend to focus on options, such as whether to enforce encryption on all communications between clients and servers, or on configuring application-specific features, such as in the case of Microsoft Exchange.

Download Additional Books from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this book to be informative, we encourage you to download more of our industry-leading technology books and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.