Enable Encrypted Connections to the Database Engine

12/21/2017 • 4 minutes to read • Contributors

In this article

To provision (install) a certificate on the server
To export the server certificate
To configure the server to force encrypted connections
To configure the client to request encrypted connections
To encrypt a connection from SQL Server Management Studio
See Also

THIS TOPIC APPLIES TO: SQL Server Azure SQL Database Azure SQL Data Warehouse Parallel Data Warehouse

This topic describes how to enable encrypted connections for an instance of the SQL Server Database Engine by specifying a certificate for the Database Engine using SQL Server Configuration Manager. The server computer must have a certificate provisioned, and the client machine must be set up to trust the certificate's root authority. Provisioning is the process of installing a certificate by importing it into Windows.

The certificate must be issued for **Server Authentication**. The name of the certificate must be the fully qualified domain name (FQDN) of the computer.

Certificates are stored locally for the users on the computer. To install a certificate for use by SQL Server, you must be running SQL Server Configuration Manager with an account that has local administrator privileges.

The client must be able to verify the ownership of the certificate used by the server. If the client has the public key certificate of the certification authority that signed the server certificate, no further configuration is necessary. Microsoft Windows includes the public key certificates of many certification authorities. If the server certificate was signed by a public or private certification authority for whi

FEEDBACK

client does not have the public key certificate, you must install the public key certificate of the certification authority that signed the server certificate.

> ⓘ **Note**
>
> To use encryption with a failover cluster, you must install the server certificate with the fully qualified DNS name of the virtual server on all nodes in the failover cluster. For example, if you have a two-node cluster, with nodes named test1.*<your company>*.com and test2.*<your company>*.com, and you have a virtual server named virtsql, you need to install a certificate for virtsql.*<your company>*.com on both nodes. You can set the value of the **ForceEncryption**option to **Yes**.

> ⓘ **Note**
>
> When creating encrypted connections for an Azure Search indexer to SQL Server on an Azure VM, see [Configure a connection from an Azure Search indexer to SQL Server on an Azure VM](#).

To provision (install) a certificate on the server

1. On the **Start** menu, click **Run**, and in the **Open** box, type **MMC** and click **OK**.
2. In the MMC console, on the **File** menu, click **Add/Remove Snap-in**.
3. In the **Add/Remove Snap-in** dialog box, click **Add**.
4. In the **Add Standalone Snap-in** dialog box, click **Certificates**, click **Add**.
5. In the **Certificates snap-in** dialog box, click **Computer account**, and then click **Finish**.
6. In the **Add Standalone Snap-in** dialog box, click **Close.**
7. In the **Add/Remove Snap-in** dialog box, click **OK**.
8. In the **Certificates** snap-in, expand **Certificates**, expand **Personal**, and then right-click **Certificates**, point to **All Tasks**, and then click **Import**.
9. Right-click the imported certificate, point to **All Tasks**, and then click **Manage Private Keys**. In the **Security** dialog box, add read permission for the user account used by the SQL Server service account.
10. Complete the **Certificate Import Wizard**, to add a certificate to the computer, and close the MMC console. For more information about adding a certificate to a computer, see your Windows documentation.

To export the server certificate

1. From the **Certificates** snap-in, locate the certificate in the **Certificates** / **Personal** folder, right-click the **Certificate**, point to **All Tasks**, and then click **Export**.

2. Complete the **Certificate Export Wizard**, storing the certificate file in a convenient location.

To configure the server to force encrypted connections

1. In **SQL Server Configuration Manager**, expand **SQL Server Network Configuration**, right-click **Protocols for** *<server instance>*, and then select**Properties**.

2. In the **Protocols for** *<instance name>* **Properties** dialog box, on the **Certificate** tab, select the desired certificate from the drop-down for the **Certificate** box, and then click **OK**.

3. On the **Flags** tab, in the **ForceEncryption** box, select **Yes**, and then click **OK** to close the dialog box.

4. Restart the SQL Server service.

> **Note**
>
> To ensure secure connectivity between client and server, configure the client to request encrypted connections. More details are explained [later in this article](#).

Wildcard Certificates

Beginning with SQL Server 2008, SQL Server and the SQL Server Native Client support wildcard certificates. Other clients might not support wildcard certificates. For more information, see the client documentation. Wildcard certificate cannot be selected by using the SQL Server Configuration Manager. To use a wildcard certificate, you must edit the

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQLServer\SuperSocketNetLib
```

registry key, and enter the thumbprint of the certificate, without spaces, to the **Certificate** value.

> **Warning**
>
> Incorrectly editing the registry can severely damage your system. Before making changes to the registry, we recommend that you back up any valued data on the computer.

To configure the client to request encrypted connections

1. Copy either the original certificate or the exported certificate file to the client computer.
2. On the client computer, use the **Certificates** snap-in to install either the root certificate or the exported certificate file.
3. In the console pane, right-click **SQL Server Native Client Configuration**, and then click **Properties**.
4. On the **Flags** page, in the **Force protocol encryption** box, click **Yes**.

To encrypt a connection from SQL Server Management Studio

1. On the Object Explorer toolbar, click **Connect**, and then click **Database Engine**.
2. In the **Connect to Server** dialog box, complete the connection information, and then click **Options**.
3. On the **Connection Properties** tab, click **Encrypt connection**.

See Also

TLS 1.2 support for Microsoft SQL Server

> **Note**
>
> The feedback system for this content will be changing soon. Old comments will not be carried over. If content within a comment thread is important to you, please save a copy. For more information on the upcoming change, we invite you to read our blog post.

2 Comments

Sign in

1 person listening

+ Follow

Post comment as...

Gobeman Mar 8, 2018

Yes you can if you install dbatools (Get-DbaForceNetworkEncryption -SqlInstance <ComputerName>).ForceEncryption

or query reg key

"HKLM:\Software\Microsoft\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQLServer\SuperSocketNetLib";

Like Reply

SQLHands Oct 25, 2017

Are there any tools to monitor the Force Encryption flag to make sure it is set to On on all SQL Servers?

Like Reply

English (United States)

Previous Version Docs • Blog • How to contribute • Privacy & Cookies • Terms of Use • Feedback • Trademarks