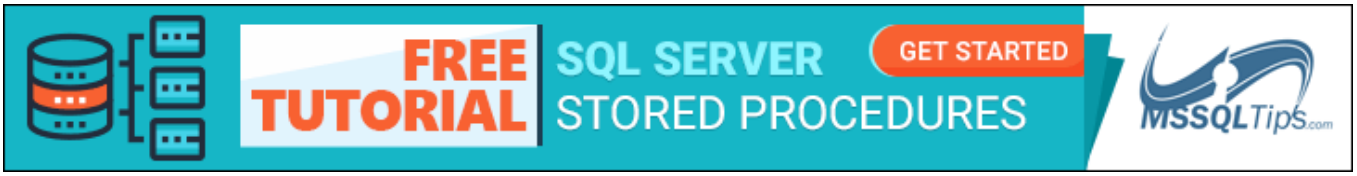
MSSQLTips Free SQL Server Conference Registration + Travel Expenses click to learn more

How to troubleshoot SSL encryption issues in SQL Server



Free Webcast - Simplify SQL Server Administration

Problem

I tried to configure SSL encryption for my Microsoft SQL Server, but it does not work properly. Could you please help me identify the root cause of the issue?

Solution

The Secure Sockets Layer (SSL) uses certificates to validate the server and the client should verify the certificate using the chain of trust where the trust anchor is the root certificate authority. This requires that the client computer should trust the root authority of the certificate used by your SQL Server. To establish an encrypted connection, the certificate should be valid and both the server and the client should be properly configured.

How are certificates validated during the SQL Server SSL handshake?

We can better understand the SSL certificate issues if we take a look at the certification validation process. The client application checks the following properties during the SSL handshake when they connect to your SQL Server using SSL encryption:

- The certificate was issued by a trusted certificate authority and none of the certificates in the chain have been revoked.
- The certificate must be enabled to be used for server authentication.
- The certificate is not expired: the Valid To property is greater than the system time.
- The Common Name (CN) in the Subject property or any of the DNS names in the Subject Alternate Name (SAN) property of the certificate matches the serverName value specified in the connection string or the optionally supplied hostNameInCertificate value.

If all of the above validation steps pass, then the authenticity of the server has been validated and the encrypted connection can be established.

SQL Server SSL Troubleshooting Steps

1. Check if the connection is encrypted

You can query the *sys.dm_exec_connections* dynamic management view (DMV) to see if the connections to your SQL Server is encrypted or not. If the value of *encrypt_option* is "TRUE" then your connection is encrypted.

```
SELECT session_id, encrypt_option
FROM sys.dm_exec_connections
```

2. Check if SQL Server service has been restarted after the configuration changed

You can query the *sys.dm_os_sys_info* DMV to see when the SQL Server service has been started, just look for the *sqlserver_start_time* value.

```
SELECT sqlserver_start_time
FROM sys.dm_os_sys_info
```

Alternatively you can check this tip to see other methods to find out the last restart time.

3. Try to connect using the Fully Qualified Domain Name

It can cause an issue if you use only the computer name in the connection string. It is better to use the Fully Qualified Domain Name (FQDN) e.g. YourSQLServer.YourCompany.int\YourSQLServerInstance

4. Check the connection string of your application

Pay attention to the following properties of the connection string:

- *encrypt*
- *trustServerCertificate*

The value of the *encrypt* property should be 'true' to enable SSL encryption. If *trustServerCertificate*=true then it is possible to connect to the SQL Server using a self-signed certificate, but this scenario is recommended only in test environments.

5. Check the certificate properties

The certificate should be valid (**Valid From** and **Valid To** properties), the **Common Name (CN)** in the **Subject** property of the certificate must be the same as the fully qualified domain name (FQDN) of the server, the **Enhanced Key Usage** property should include '*Server Authentication (1.3.6.1.5.5.7.3.1)*' and the certificate must be created by using the **KeySpec** option of '*AT_KEYEXCHANGE*'.

6. Check if the certificate is trusted by the client

The client machine should trust the certificate so there are two options:

- The SQL Server's certificate should be installed on the client machine to establish a direct trust.
- The certificate of the root certificate authority and the intermediate/chain certificates should all be trusted. This way you can take advantage of the chain of trust, the core principle of SSL certificate hierarchy.

Next Steps

- Follow the steps to find out the root cause of your SSL problem.
- You can find more articles in the Security category.
- Read more tips by the author here.

Last Update: 2014-11-25

You're invited
to our next free
SQL Server webcast!

click to learn more

Next Tip

Click Here

About the author

Tibor Nagy is a SQL Server professional in the financial industry with experience in SQL 2000-2012, DB2 and MySQL.

View all my tips

Related Resources

- More SQL Server DBA Tips...

Post a comment or let the author know this tip helped.

All comments are reviewed, so stay on subject or we may delete your comment. Note: your email address is not published. Required fields are marked with an asterisk (*).

*Name *Email Notify for updates

*** NOTE *** - If you want to include code from SQL Server Management Studio (SSMS) in your post, please copy the code from SSMS and paste the code into a text editor like NotePad before copying the code below to remove the SSMS formatting.

p

Send me SQL tips:

I'm not a robot

reCAPTCHA
Privacy - Terms

Save Comment

Monday, December 08, 2014 - 11:24:44 AM - Encryption cipher suite? Back To Top

I agree that SELECT session_id, encrypt_option FROM sys.dm_exec_connections is a good way of checking to see if SQL Server thinks a connection is NOT encrypted (FALSE). However, does anyone have a good way of finding out the precise cipher suite and protocol in use, so as to be able to provide evidence to HIPAA/HITECH or PCI-DSS or other regulatory based auditors that when you say "encrypted", it qualifies as "properly encrypted" (i.e. NOT using RC-4, NOT using MD5, YES using approved ciphers in approved modes with approved hashes)? For anyone who needs to comply with specific requirements, including FIPS 140-2 or NIST SP800-131A, this is very important.

Get Free SQL Tips

enter business email Join

SQL Server, Azure and AWS. Owned.
DOWNLOAD NOW

SQL Monitor
Get your free ROI calculation
Start now

Get Free SQL Server Tips

Learning

DBAs

Developers

BI Professionals

Careers

Q and A

Today's Tip

Resources

Tutorials

Webcasts

Whitepapers

Tools

Copyright (c) 2006-2018 Edgewood Solutions, LLC All rights reserved

Some names and products listed are the registered trademarks of their respective owners.