Connection Encryption when using custom FQDN

Last updated by | Vitor Tomaz | Jun 8, 2022 at 5:31 AM PDT

Contents

Issue

Explain how connection encryption works when connecting to Azure SQL DB. expecilly when using the connection string properties Encrypt and TrustServerCertificate

From Azure SQL DB we know that it enforce encryption for every connection. From Client we know that we can set true or false for the following settings [Encrypt] and [TrustServerCertificate]

Investigation/Analysis

When using custom FQDN (or IP address) to connect to my SQL DB and [encrypt] is set to true the connection fail unless [TrustServerCertificate] is set to true as well. This does make sense as the certificate could not be validated with this name (this is different than the name in the certificate) However, if Encrypt=false, we know that server enforce the encryption so why in this case the connection succeeded although [TrustServerCertificate]=false how could that be that the connection successfully encrypted (as it was enforced by server) while we are using different name when connecting?

Mitigation

When Encryption enforced by server, the client is not following the procedure of validating the certificate. So, we do have encrypted connection, however if the certificate is not validated by the client we are exposed to man in the middle attack. Therefore, customers should not use the IP address or custom DNS name with the Azure SQL DB to keep their connection safe from attacks.

Public Doc Reference (optional)

Source: https://docs.microsoft.com/en-us/azure/azure-sql/database/security-overview#transport-layer-security-encryption-in-transit “As a best practice, recommend that in the connection string used by the application, you specify an encrypted connection and not trust the server certificate. This forces your application to verify the server certificate and thus prevents your application from being vulnerable to man in the middle type attacks.”

Root Cause Classification

Cases resolved by this TSG should be coded to the following root cause:
Azure SQL v3\Connectivity\How-to/advisory

How good have you found this content?