

Changes to Azure SQL DB TLS Root Certificate Authorities

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:29 AM PST

Contents

- [Introduction](#)
- [What is Changing?](#)
- [What's the Impact?](#)
- [What to do](#)
- [Installing Root CA Certificates](#)
- [SQL Server IaaS is not Affected](#)
- [Summary](#)

Introduction

Transport Layer Security (TLS) provides server authentication and channel defenses (encryption and integrity verification) for communication between two applications such as a web browser and a web server. Optionally, TLS can provide client authentication, too. Most TLS connections today use X.509 certificates, and core to certificates are root Certificate Authority (CA) certificates. For a client to successfully establish a secure connection to a server using TLS, the client system must trust the CA that issued the server's certificate. The word 'trust' in this scenario means the client has the CA's root certificate installed in the client system.

Starting Jan 2023, we will update the root CA certificates used by all Azure services, including the database services such as Azure SQL Database, Cosmos DB, Azure Database for PostgreSQL, and Azure Database for MySQL. And this change might have implications for Azure SQL DB customers.

What is Changing?

Azure is changing the set of root certificates used by Azure services. Right now, almost all Azure services use one root CA certificate for TLS:



Baltimore CyberTrust Root By the end of the calendar year 2022, Azure services will chain up to one of the following CAs:


- DigiCert Global Root G2
- DigiCert Global Root CA
- Baltimore CyberTrust Root
- D-TRUST Root Class 3 CA 2
- Microsoft RSA Root Certificate Authority 2017
- Microsoft ECC Root Certificate Authority 2017



What's the Impact?

Most Azure database users will see no impact because the new set of root CA certificates are commonly installed on systems including mobile devices. Their client code will continue to make secure connections to back-end databases on Azure.

The potential issue is if developers design their code in a way that restricts which root CA certs are valid and trusted. This is called Certificate Pinning. You could have, for example, a dozen roots on a device, but your application only trusts one specific root. So, in your code, you explicitly check for that certificate when making a TLS connection. This is usually performed by checking the thumbprint of the certificate in your code.

At the time of its invention, pinning seemed like a good idea, but it has since fallen out of favor as it leads to fragility. You can read one point of view from DigiCert at [Stop Certificate Pinning | DigiCert.com](#) . OWASP has an article on the topic of pinning too, with sample code [Certificate and Public Key Pinning | OWASP Foundation](#) .


A more concrete example is if your code pins the Baltimore CyberTrust Root CA certificate, but Azure SQL DB uses the DigiCert Global Root CA certificate, then the client will not connect to Azure SQL DB. If your code is C/C++ Windows code, another way to mimic certificate pinning is to use [Certificate Trust Lists \(CTL\)](#) .

You can learn more about the certificates we will use at [Azure TLS Certificate Changes](#)  | Microsoft Docs and the Cosmos DB specific post is at [Upcoming changes to Azure Cosmos DB TLS certificates - Azure Cosmos DB Blog](#) .

What to do

Review all your code that interacts with Azure services, including our PaaS database products and make sure TLS connections do not limit which root CA certificates are valid. For Windows C/C++ code, make sure there is no use of CTLs.

Installing Root CA Certificates

You probably won't need to add the new root CA certificates to your clients, but if you do, here is how to do it on Ubuntu Linux [Installing a root CA certificate in the trust store](#)  and how to use Group Policy on Windows to deploy certificates to your enterprise [Distribute Certificates to Client Computers by Using Group Policy | Microsoft Docs](#) or manually [How to install Windows 10/11 root certificates](#). Some applications might have their own root CA store and not rely on the operating system, however.

SQL Server IaaS is not Affected

Note that SQL Server running in a Windows or Linux Virtual Machine is not affected by this change because you can configure any certificate and root used by the database server within the operating system. You cannot do this when using PaaS Azure databases because certificates are handled by Azure.

Summary

Most people will see no issues at all with this update. Just perform a little due diligence on your client-side code to make sure it is not restricting CA certs.

How good have you found this content?



-