Error 47072 Login failed with invalid TLS version

Last updated by | Vitor Tomaz | Aug 5, 2020 at 12:39 PM PDT

Contents

Issue

The Minimal Transport Layer Security (TLS) Version setting allows customers to control the version of TLS used by their Azure SQL Database.

At present we support TLS 1.0, 1.1 and 1.2. Setting a Minimal TLS Version ensures that subsequent, newer TLS versions are supported. For example, e.g., choosing a TLS version greater than 1.1. means only connections with TLS 1.1 and 1.2 are accepted and TLS 1.0 is rejected.

For customers with applications that rely on older versions of TLS, we recommend setting the Minimal TLS Version per the requirements of your applications.

For customers that rely on applications to connect using an unencrypted connection, we recommend not setting any Minimal TLS Version.

After setting the Minimal TLS Version, login attempts from clients that are using a TLS version lower than the Minimal TLS Version of the server will fail with following error:
Error 47072
Login failed with invalid TLS version

Investigation/Analysis

Use the following query to find more about these attempts:

```
let server = "{ServerName}";
let database = '{DatabaseName}';
MonLogin
| where logical_server_name =~ server or LogicalServerName =~ server
| where database_name =~ database
| where error == 47072
| take 1000
| project TIMESTAMP, error, ['state'], ssl_protocol, ssl_cipher, tds_flags, tds_version, code_package_version,
```

The error can have 2 main causes:

Connection is using a TLS version lower than the minimum TLS version configured for the server

Check the **ssl_protocol** in the results, encrypted connections would have one of the following options:

- SNI_TLS1_1_SERVER
- SNI_TLS1_2_SERVER
- SNI_TLS1_SERVER

As an example, if customer configured TLS 1.2 as minimal, any attempt to connect using SNI_TLS1_SERVER or SNI_TLS1_1_SERVER will fail.

Unencrypted connection attempts while a minimum TLS version configured

Check the **ssl_protocol** in the results, unencrypted connections will have ssl_protocol as **SNI_UNDEFINED_PROTOCOL** Any unencrypted connections attempt while a minimum TLS version is configured will fail.

Mitigation

For customers with applications that rely on older versions of TLS, we recommend setting the Minimal TLS Version per the requirements of your applications.
For customers that rely on applications to connect using an unencrypted connection, we recommend not setting any Minimal TLS Version.

For SQL DB, at the moment (2020-06-22), disabling minimum TLS version can only be done via PG (raise an incident).

For SQL MI, customers can use the Portal, PowerShell, REST API, etc.:

Public Doc Reference

https://docs.microsoft.com/en-us/azure/azure-sql/database/connectivity-settings#minimal-tls-version

Root Cause Classification

Cases resolved by this TSG should be coded to the following root cause:
Azure SQL DB v2\Connectivity\Login Errors\Other

How good have you found this content?