SNI-Readtimeout

Last updated by | Vitor Tomaz | Dec 15, 2021 at 2:18 AM PST

Contents

SNI-Readtimeout

Issue

Logins attempts to {DatabaseName} on {ServerName} was *potentially* failing between {StartTime} and {EndTime} for {FailureCount} times, because the clients login packets taking were too slow (>=5 seconds) to respond Or send TLS handshake messages.

Analysis

This is typically a client side error involving high application CPU usage resulting in client taking > 5s to respond to a pre-login ack sent by server. After 5s, the connection is forcibly killed by the service to prevent trickle attacks.

The XDBGateway in control ring, and XDBHost in tenant ring will forcibly terminate these slow connections as part of security measure and a SNI Read Timeout trace will be emitted during this time. Since we forcibly terminate the login during the pre-login phase, we will not have data that arrives in the login TDS packets, particularly Database and Server Names.

This complicates the troubleshooting and the option left is to correlate by masked Ips and customer might be attempting to connect to database from different IPs.

- Gets all the client Ips reaching the customer's logical server.
- Join that IP set with the SNI read timeout errors
- Checks whether the resulting Ips are being used by other server names and prints out the count (is somebody else sharing the same masked IP?)

Client will see a TCP connection forcibly closed each time this happens

Sample client side error msg:

- System.Data.SqlClient.SqlException (0x80131904): A connection was successfully established with the server, but then an error occurred during the login process. (provider: SSL Provider, error: 0 - An existing connection was forcibly closed by the remote host.) ---> System.ComponentModel.Win32Exception (0x80004005): An existing connection was forcibly closed by the remote host

- Prelogin timeouts

Mitigation

If the issue persists after customer resource reduction, start network troubleshooting (Azure SQL Connectivity Checker, involve CloudNet, etc...) as packet loss might be happening.

To do for Customer

Request customer to check the following -

Client machine resource contention - This may cause clients to send packets much slower than normal. Note that single-core CPU contention could easily cause this, observing max (core) is recommended

Network Slowness - Intermittent network latency may also cause this slowness, however in this case the impact will not be limited to one resource/customer.

Customer Canned RCA

As a protection, Azure SQL Database endpoints forcibly terminate connections when the client login packets take several seconds to arrive to our service. The most likely cause of this problem is high resource contention on client VMs (CPU, threads, memory) causing clients to send packets much slower than normal. In rare instances, this issue is caused by intermittent network latency.

From our telemetry, we have found that most of the cases where this error condition is hit happens while the client is processing the TLS handshake. Typically, this pattern indicates high single-core CPU pressure on the client VM.

Classification

Root Cause: Azure SQL DB v2\Connectivity\Network Reliability\Client Network

How good have you found this content?