Error 33155 State 1

Last updated by | Vitor Tomaz | Dec 15, 2021 at 2:34 AM PST

Handling Azure AD Error - 33155

Contents

Issue

Error 33155 typical occurs when Client stopped talking to AAD server, failing Active Directory auth to SQL DB because the client is not being able to grab an authentication token from Active Directory (AD) or Azure Active Directory (AAD).

Analysis/Triage

From the server side, login will fail with error code 33155 (visible in MonLogin).

From the client side, error will fail with error code 0xCAA82EFD

// MessageId: ERROR_ADAL_INTERNET_CANNOT_CONNECT

// MessageText:

// The attempt to connect to the server failed.

// #define ERROR_ADAL_INTERNET_CANNOT_CONNECT ((DWORD)0xCAA82EFDL)_

This is a clear indication that one or more endpoints for authentication are blocked, see "Troubleshooting" and "Required Endpoints" sections below.

Troubleshooting

When SQL Client uses AD Authentication, there are some extra calls the client needs to do to Azure Active Directory and to the customers' Active Directory in order to generate a token to be sent as part of the LOGIN packet. This diagram explains: <link>

If the calls client->AAD or client->(on premises AD) fail, client will either timeout or generate an invalid token which SQL DB will reject. At this point should be clear that the problem is within customer configuration and not with SQL DB, however more often than not customer needs hints on where to look next. Most common case is client communication to AD/AAD is being blocked by a firewall or NSG. To confirm

STEP 1 Ask the customer to take a packet capture of the repro of the problem. (Use Azure SQL Connectivity Checker)

STEP 2 Open the trace (netmon, wireshark should work) Identify the SQL login traffic, look for the last login: in case of Proxy usage, the last login will go to only Gateway thus only using port 1433. If traffic is redirected, it will use the port of the DB node. If redirected, use Kusto MonLogin to find out port of the DB note (SQL Gateway/Connectivity expert knows how to identify this).

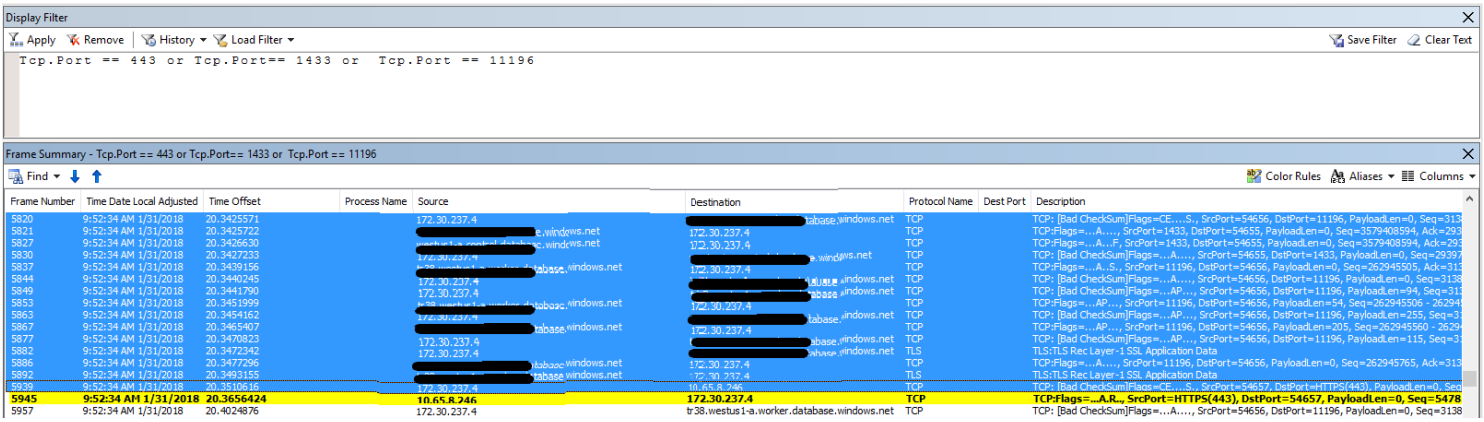
STEP 3 - Once backend port is identified, filter the trace. In this example, customer logins were getting redirected to DB Node 196.

STEP 4 Find the first TCP message of the *last* SQL connection (the redirected one in case of redirection). The first TCP message is always a SYN packet (Flags=CE.....S.), first one on screenshot above Note: Network captures is always noisy, it really helps if the machine were the trace is taken is not doing anything else 😃

STEP 5 - You are seeing a TDS handshake, and SQL Client will try to go to AAD on the client-side Login phase, right after the TLS handshake. Note the "SSL Application Data" on Frames 5886, 5892, 5939 - This is the TLS handshake happening and the next expected message is a Login packet from client. *However in AAD case, a Client will initiate calls on port 443 to get an AuthN token before sending the Login packet*. You should see those next. In this case, client (172.30.237.4) started an AAD HTTPS (port 443) connection to 10.65.8.246 on frame 5945 but that IP rejected it (TCP RST, frame 5945 in yellow). This typically means that a firewall on 10.65.8.246 explicitly rejected the connection. Next step is to let customer know that the client VM needs to talk to their Active Directory sitting on 10.65.8.246 and they need to debug further to find out where the firewall is.

Pro tip: Network Monitor has Color Filters (CTRL+D). This case is so typical that I just have a rule to color all TCP RSTs in yellow:

Pro tip: Network Monitor has Color Filters (CTRL+D). This case is so typical that I just have a rule to color all TCP RSTs in yellow:



Screen clipping taken: 1/31/2018 2:35 PM

Required Endpoints

Using PowerShell, customer can check connectivity from application to the following endpoints: AAD endpoint for password and integrated auth: tnc login.windows.net -port 443 AAD endpoint for Universal with MFA auth: tnc login.microsoftonline.com -port 443 [OPTIONAL, check only if domain is Federated, see below how to figure out endpoint] Endpoint to on-premises Active Directory Federation Server (ADFS), this is required for all auth modes when domain is Federated: tnc <customer specific="" adfs="" endpoint=""> -port 443

This is the minimum connectivity requirement, it's possible that additional endpoints are required, which depends on customer's AAD and on-premises AD setup. We have seen in the past issues due to blocked endpoints for certificate revocation list (CRL). Capturing and debugging network traces is what usually helps in those situations.

How to figure out customer's ADFS endpoint?

This may be determined from "Azure AD Explorer" in ASC report for your CRI: Open Azure AD Explorer tab -> select Domains tab on the left -> select Federation tab on top -> Look at Federation Brand Name column (in this case it's sts.firstam.com) Then, test connectivity using: tnc sts.firstam.com -port 443

Public Doc Reference

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-aad-authentication

Internal Reference

TSG shared above was from Product Team and available for TSG with limitations. If identified the TSG contains [Internal Only] information, please open a repair item to fix it.

Classification

Root cause Tree - Security/Active Directory Authentication

How good have you found this content?