AAD Auth Troubleshooting

Last updated by | Vitor Tomaz | Oct 18, 2022 at 3:52 AM PDT

Contents

Terminologies

Terminology	Description
Security Token Service (STS)	A service responsible for issuing security tokens for claim-based systems. For example, EvoSTS (a component of AAD) is an STS
Identity Provider STS (IdP-STS)	An STS that also acts as an Identity Provider, i.e. it manages/owns user identities. For example, AAD acts as an IdP-STS for managed accounts, i.e. accounts that “live” in AAD and password and metadata (e.g. group membership information) is stored in AAD.
Relying Party STS (RP-STS)	An STS that does not authenticate users (it is not an IDP), but instead it has trust relationships with one or more other IdPs and delegates authentication (redirects clients) to those IdPs. For example, EvoSTS (in AAD) acts as an RP-STS for federated accounts (for which, an IdP is an on premise Active Directory) or for Microsoft accounts (for which, and IdP is Microsoft Accounts).
AAD managed account	An account, for which AAD is an IdP-STS – account attributes are stored in an AAD directory
AAD federated account	
An account that is managed by an on premise Active Directory (account data/attributes are stored on premise), which has been federated with an AAD directory via Active Directory Federation Services (ADFS).	
Active Directory Authentication Library (ADAL)	A library that enables client application developers to easily authenticate users to cloud or on-premises Active Directory (AD), and then obtain access tokens and use them to connect services/call APIs. ADAL is an abstraction layer that makes differences in using on premise AD vs. AAD and managed vs. federated accounts transparent to the application.
Microsoft Accounts or MSA (previously Microsoft Wallet, Microsoft Passport, .NET Passport, Microsoft Passport	An Microsoft IdP service that allows users to log into cloud services, including Azure portals. AAD makes it possible to import MS accounts into AAD directories.

Terminology	Description
Network, Windows Live ID, and most recently Microsoft account)	
Multi-factor authentication (MFA)	An approach to authentication which requires the presentation of two or more of the three independent authentication factors: a knowledge factor ("something only the user knows"), a possession factor ("something only the user has"), and an inherence factor ("something only the user is"). A common example of MFA is smart card authentication – a user provides a smart card (a possession factor) and a smart card PIN (knowledge factor).
FedAuthinfo Token	The Client embeds an indication that it wants to use ADAL authentication in its LOGIN7 structure during TDS request. Azure SQL DB sends back a new TDS Token type called as FedAuthInfoToken once it understands that the client wants to use ADAL. In the FedAuthInfoToken Azure SQL DB sends back the SPN Name , TokenEndpoint similar to http://login.windows.net/TenentID (TenantID is a GUID which represents the customers Identifier in Azure AD)
FedAuthSecurity Token or FedAuth Token	The client utilizes the information provided in the FedAuthInfo Token and gets authenticated with Azure AD and gets the JSON WEB TOKEN , This JWT is presented back to Azure SQL DB and this Token in Azure SQL DB is called as FedAuthSecurity Token. This Token is a new type of TDS Token which is internally called as FedAuth Token as well.
JWT	JSON Web Token (JWT) is a compact token format intended for space constrained environments such as HTTP Authorization headers and URI query parameters. JWTs encode claims to be transmitted as a JavaScript Object Notation (JSON) object.
SAML	Security Assertions Markup Language (SAML) tokens are XML representations of claims.

Metadata Mapping

sys.user_token

sys.database_principals

select name, principal_id, type,type_desc, sid , CAST(sid as uniqueidentifier) as AzureADObjectID from sys.database_principals where name='user@microsoft.com'

```
name                    principal_id  type                 type_desc                              sid
user@microsoft.com  8             X      EXTERNAL_GROUP       0xEA3ACF302A8F06989898989089990        999999
```

PowerShell Windows Powershell

PS C:\windows\system32> Add-AzureAccount

PS C:\windows\system32> connect-msolservice -credential $msolcred

PS C:\windows\system32> Get-MsolUser -UserPrincipalName user@microsoft.com | select ObjectID

```
ObjectID
--------
99999999-8F2A-4906-A53F-00000000000
```

Troubleshooting

Troubleshooting approach can be divided into the below areas. Please make sure understand what is the exact activity that customer has performed as it will provide you indications on how to approach.

- Client Side
- Using Backend Telemetry

Client Side

- Home Grown Tools
- BID Trace
- Fiddler Trace
- Network Monitor
- HttpMessages Tracing

Backend Telemetry

Kusto /MDS

- MonLogin
- MonFedAuthTicketService
- MonAzureActivDirService
- MonFedAuthPrincipals
- MonLoginUserDDL
- AlrLoginLatency

- AlrLogin

MonLogin table has these interesting columns which can be utilized to validate if some customer questions about latency during login or ADAL Logins taking more time for authentication.

- Fedauth_library_type
- Fedauth_adal_workflow
- Fedauth_fetch_signingkey_refresh_time_ms
- Fedauth_jwt_token_parsing_time_ms
- Fedauth_signature_validation_timems
- Fedauth_context_build_time_ms
- Fedauth_group_expansion_time_ms
- Fedauth_token_wait_time_ms
- Fedauth_token_process_time_ms
- total_time_ms

Key Column values and details.

Fedauth_library_Type	Fedauth_adal_workflow	Authentication_type
0	0	SQL server authentication
2	0	Token based Auth
3	1	Azure AD Password Auth
3	2	Azure AD Integrated Auth
4		Windows Auth

Explanation of important columns in MonAzureActiveDirService

- sql_connection_id : Represent the connection id in the server side. Can be used to co-relate MonLogin, MonFedAuthTicketService & MonAzureActiveDirectory Tables
- error_state : Represents the state in which a particular error occurred (details in table below).
- error_code : Represents the system error(in most cases) which happened in that state
- operation_type : Represents the Operation happening (refer to the Operation Types table below)
- error_message : Contains error messages mostly around interaction with Graph Server failed with particular http status codes which are standard.

Please use LogicalServerName in case you want to join on logical server name from tables like MonLogin & MonFedAuthTicketService.

Explanation of important columns in MonFedAuthTicketService

- sql_connection_id : Represent the connection id in the server side. Can be used to co-relate MonLogin, MonFedAuthTicketService & MonAzureActiveDirectory Tables
- error_state : Represents the state in which a particular error occurred (details in table below).
- error_code : Represents the system error(in most cases) which happened in that state

Please use LogicalServerName and database_name in case you want to join on logical server name and logical database name from tables like MonLogin.

Extra Fields in Kusto (In MDS each of the name value pairs will have their own column types) (explained below)

TIp

Whenever in doubt about networking layer or connectivity layer to Azure AD utilize the below

- Validate same login is able to connect to the Azure Portal and query data.
- Use Basic Azure PowerShell commands to connect to Azure
- Use ExtractAndPrintClaims.zip to eliminate the issue outside SQL Azure DB

CMS / HttpQuerytool

- sql_instances
- logical_servers
- logical_databases
- fabric_properties

XTS

- Database Availability.xts
- FedAuthDebugProvisioning.xts
- Fedauth events.xts

Basics

- It's very important to understand that some part of Authentication happens between (Client & AAD) and some between (Azure SQL DB & AAD) , Based on at which layer the error got raised your troubleshooting approach will change.
- If you remember the architecture implementation SQLADAL.DLL is loaded by ADO.NET inside the client application address space, ADAL.DLL is loaded into Azure SQL DB address space.
- Interaction from Client to AAD happens using WinHTTP messages
- Interaction from Client to Azure SQL DB happens using TDS Packets
- Azure SQL DB uses and understands JWT (Jason Web Token)
- Azure Active Directory uses and understands JWT , SAML , OAuth-A , OAuth-T & WS-Fed
- The JWT Token is only valid for 60min in Azure SQL DB Server, If some operation tries to use the same token after expiration timeframe it will encounter an error.

- Azure SQL DB does not have an automatic Token Renewal mechanism.
- As there are multiple components involved always try to validate if we can find a way to eliminate the issue outside Azure SQL DB (Reproducing the issue without using Azure SQL DB), This will save lot of time.
- In the Backend Telemetry tables , For some tables data is only logged when there is an error (We dont log any data for successful scenarios)
- Connection Timeout should be extended to 30seconds (In SSMS the default is 15 seconds).
- While connecting please make sure to specify the exact context of the DB to which the login has access (Remember apart from AAD Admin login all other AAD Logins added are contained database users).
- The JWT Token is only valid for Azure SQL DB

Scenarios

Client Side Validation Error

- Client pushed a login request to Azure SQL DB using AAD Login and utilized (Authentication=ActiveDirectoryPassword).
- By mistake the PWD for the login was provided incorrectly or the login name was typed incorrectly. (This is not an uppercase/lowercase scenario for username)
- You will not see 18456 error message, As 18456 is defined inside SQL Engine. In this scenario login failure will happen when authentication is happening between Client & AAD
- Azure SQL DB will not even know that there was a login failure that occurred (Client just said to SQL it wants to use ADAL and SQL has just provided him back SPN,URL of STS) and waiting.

In the Application side you will notice error message similar to below

```
===================================

Cannot connect to pradmeasterling.database.windows.net.
===================================
Failed to authenticate the user pradm@microsoft.com in Active Directory (Authentication=ActiveDirectoryPasswor
Error code 0xCAA20003; state 10
ID3242: The security token could not be authenticated or authorized. (.Net SqlClient Data Provider)
------------------------------
For help, click: http://go.microsoft.com/fwlink?ProdName=Microsoft%20SQL%20Server&EvtSrc=MSSQLServer&EvtID=0&L
------------------------------
Server Name: pradmeasterling.database.windows.net
Error Number: 0
Severity: 11
State: 0
Procedure: ADALGetAccessToken
------------------------------

**And the stack would be as below**

at System.Data.SqlClient.SqlInternalConnectionTds.GetFedAuthToken(SqlFedAuthInfo fedAuthInfo)

   at System.Data.SqlClient.SqlInternalConnectionTds.OnFedAuthInfo(SqlFedAuthInfo fedAuthInfo)

   at System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dat

   at System.Data.SqlClient.TdsParser.Run(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataSt

   at System.Data.SqlClient.SqlInternalConnectionTds.CompleteLogin(Boolean enlistOK)

   at System.Data.SqlClient.SqlInternalConnectionTds.AttemptOneLogin(ServerInfo serverInfo, String newPassword

   at System.Data.SqlClient.SqlInternalConnectionTds.LoginNoFailover(ServerInfo serverInfo, String newPassword

   at System.Data.SqlClient.SqlInternalConnectionTds.OpenLoginEnlist(TimeoutTimer timeout, SqlConnectionString

   at System.Data.SqlClient.SqlInternalConnectionTds..ctor(DbConnectionPoolIdentity identity, SqlConnectionStr

   at System.Data.SqlClient.SqlConnectionFactory.CreateConnection(DbConnectionOptions options, DbConnectionPoo

   at System.Data.ProviderBase.DbConnectionFactory.CreateNonPooledConnection(DbConnection owningConnection, Db

   at System.Data.ProviderBase.DbConnectionFactory.TryGetConnection(DbConnection owningConnection, TaskComplet

   at System.Data.ProviderBase.DbConnectionInternal.TryOpenConnectionInternal(DbConnection outerConnection, Db

   at System.Data.ProviderBase.DbConnectionClosed.TryOpenConnection(DbConnection outerConnection, DbConnection

   at System.Data.SqlClient.SqlConnection.TryOpenInner(TaskCompletionSource`1 retry)

   at System.Data.SqlClient.SqlConnection.TryOpen(TaskCompletionSource`1 retry)

   at System.Data.SqlClient.SqlConnection.Open()

   at Microsoft.SqlServer.Management.SqlStudio.Explorer.ObjectExplorerService.ValidateConnection(UIConnectionI

   at Microsoft.SqlServer.Management.UI.ConnectionDlg.Connector.ConnectionThreadUser()

Notice the api name ADALGetAccesToken, This is the request that failed when the client application posted a To

// MessageId: ERROR_ADAL_SERVER_ERROR_INVALID_GRANT

// MessageText:

// Authorization grant failed for this assertion.

define ERROR_ADAL_SERVER_ERROR_INVALID_GRANT ((DWORD)0xCAA20003L)

If you try to look at the Telemetry data , All you will notice is that there was a connection request that was
```

```
Logon       Error: 33155, Severity: 20, State: 1.

Logon       A disconnect event was raised when server is waiting for Federated Authentication token. This coul
```

Mitigation

- You can request customer to connect to the Azure Portal using the same login name and password to validate if it encounters the same issue.
- You can use AzureAuthenticationExamples program to validate if we encounter the same issue.
- Ideally both scenarios mentioned above will fail with the same error, In case if it succeeds then we might have to take an IDNA of the client application and then open an ICM with Azure SQL Security Team.

Client side WinHttp Tracing

Below are the instructions on how to trace WinHttp Messages from the client machine

```
C:\windows\system32>netsh trace start scenario=InternetClient
Trace configuration:
-------------------------------------------------------------------
Status:             Running
Trace File:         C:\Users\pradm\AppData\Local\Temp\NetTraces\NetTrace.etl
Append:             Off
Circular:           On
Max Size:           250 MB
Report:             Off

Simulate the Scenario
C:\windows\system32>netsh trace stop
Correlating traces ... done
Merging traces ... done
Generating data collection ... done
The trace file and additional troubleshooting information have been compiled as
"C:\Users\pradm\AppData\Local\Temp\NetTraces\NetTrace.cab".
File location = C:\Users\pradm\AppData\Local\Temp\NetTraces\NetTrace.etl
Tracing session was successfully stopped.
Process the Trace
@tracerpt -y NetTrace.etl -of csv -o temp.txt
```

The output will look similar as below , You can engage WinInet team if we need detailed analysis of this log file

```
30861064999445422,       915,       2475, 0x119F3998, 0xFE00000230000006, 0xA1AF3F0, 0xFD00000220000008, "POS
Microsoft-Windows-WebIO,       Set ,           23,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,       Set ,           15,          0,          16,          4,          14,         201,
Microsoft-Windows-WebIO,     Start ,          130,          0,          16,          4,           1,         413,
```

Scenario 3 - Token Expired Issue

Token Expired Issue

A client connects to Azure SQL DB using the JWT Token that it has retrieved from AAD, It has been more than 60min since the token was issued. Since in Azure SQL DB the expiration timeframe is

set to 60min, Azure SQL DB will raise Token Expired error and user will encounter the below error message

Login failed for user 'NT AUTHORITY\ANONYMOUS LOGON

In the Azure SQL DB Errorlog You will notice an error as below

```
2015-09-14 14:15:12.44 spid91      Federated Authentication JWT ReadAndValidateToken failed with state: 9, JSO

2015-09-14 14:15:12.44 spid91      Federated Authentication Failed for LibraryType: 1, ErrorState: 132, HRESUL
2015-09-14 14:15:12.44 Logon       Error: 18456, Severity: 14, State: 132.
2015-09-14 14:15:12.44 Logon       Login failed for user 'NT AUTHORITY\ANONYMOUS LOGON'.Reason: There is a use
```

State 132 of 18456 error gets translated to

```
enum ELoginFailedState

{
                x_elfFedAuthAADLoginJWTUserError 132
}
```

x_elfFedAuthAADLoginJWTUserError gets mapped as

```
const LOGINFAILEDSTATETOREASONPHRASEMAP x_rgLoginFailedToReasonPhraseMaps[] =
{
                x_elfFedAuthAADLoginJWTUserError,                                                PH_FEDAUTH_AAD
}
```

PH_FEDAUTH_AAD_LOGINJWTUSERERROR is the error

```
// ErrorNumber: 47032
// ErrorSeverity: EX_INFO
// ErrorOwner: sipen
// ErrorFormat: Reason: There is a user error in FedAuth token parsing. There should be a seperate XEvent call
// ErrorCause: Usually an indication that users pass an incorrect formate of FedAuth Token

const int PH_FEDAUTH_AAD_LOGINJWTUSERERROR = 47032;
```

This error is getting raised in the below function

```
HRESULT JSONWebTokenService::ReadAndValidateToken
(               __in IMemObj* pmo,
                __in_bcount(ulcbToken) const WCHAR* const pwszToken,
                __in ULONG const& ulcbToken,
                __out IFedAuthTicket** ppFedAuthTicket,
                __out bool& fSDSServerAdmin,
                __out ELoginFailedState& eELoginFailedState,
                __out LONGLONG& llTokenValidity
)
{

                Exit:
                                if (hr != ERROR_SUCCESS)
                                {
                                                WCHAR wszDetailMessage[128] = {0};
                                                MapToELoginFailedState(eJWTErrorState,eELoginFailedState);
                                                TraceFailureToXEvent(eJWTErrorState);
                                                StringCchPrintfW(
                                                                STRING_AND_CCH (wszDetailMessage),
                                                                L"Federated Authentication JWT ReadAndValidate
                                                                (int)eJWTErrorState,
                                                                ulJsonResult,
                                                                dwError);

                                                SOSLogToErrorLog(wszDetailMessage);
                                  }
                                return hr;
```

Federated Authentication JWT ReadAndValidateToken failed with state: 9 would get translated as below

```
enum JSONWebTokenErrorState

{
                esDefault = 0,
                esTokenHeaderInvalidTokenType,
                esTokenHeaderInvalidSignatureAlgorithm,
                esEvoSTSUrlLookupInvalid,
                esOidToGUIDConversionError,
                esTokenAudienceInvalid,
                esTokenDecodeBase64HeaderFailed,
                esTokenDecodeBase64PayLoadFailed,
                esTokenDecodeBase64SignatureFailed,
                esTokenExpired,
                esTokenFailedConvertToBinaryRetrieveLength, //10
```

How good have you found this content?