**Incident Report**

**APIM Live – Account Manager API, Branch Locator API**

**November 20th 2020**

**Account Manager and Branch Locator API’s both located on 10.10.7.213:8090 were called successfully from Azure APIM TEST instance on 18th of November 2020. However, the Azure APIM LIVE instance of the same API’s could not be called and returned persistently a 401 – Unauthorized response.**

**API Repository:** [**https://dev.azure.com/TruLab/Account%20Manager%20API**](https://dev.azure.com/TruLab/Account%20Manager%20API)

**APIM TEST**

**=====================**

**Call “BranchLocationAPI” – GET on Azure APIM TEST**

[**http://10.10.7.213:8090/api/branch/location/search?location=lekki**](http://10.10.7.213:8090/api/branch/location/search?location=lekki)

**Response – 200**

**Response Status: Successful**

**Location List.**

**Database Server – 10.10.3.12**

**Database – AccountManagerDB**

**APIM LIVE**

**=====================**

**Call “BranchLocationAPI” – GET on Azure APIM TEST**

[**http://10.10.2.100:8080/api/branch/location/search?location=lekki**](http://10.10.2.100:8080/api/branch/location/search?location=lekki)

**Response – 401**

**Response Status: Unauthorized**

**Location List.**

**Database Server – 10.10.7.60**

**Database – AccountManagerDB**

**SNAPSHOT ON THE AUTHENTICATION/AUTHORIZATION OF THE API**

**The API based on findings uses a** [CustomAuthorize] **on each of the Controllers in the AccountManagerAPI. And this [CustomeAuthorize] is wired into an** CustomAuthorizeAttribute class.

For Successful authorization, 3 header Parameters are required

* **Client-id**
* **Client-key**
* **ClientIpAddress**

**The Client-Id and Client-Key are passed in the headers BUT the ClientIpAddress is automatically fetched by a Utility method – GetClientIp – that is inside the AuthenticateClient class.**

**Based on the header parameters passed (client-id, client-key and ipaddress calculated), the profile of the client (service calling the request) - ClientProfile - is validated using the passed-in header params AND the information saved in the ClientProfiles Table inside the AccountManagerDB DATABASE.**

***See*** BaseService.UtilityService.ClientAuthenticationServices in the AccountManagerAPI code.

It uses this validation to get the clientprofile from the database which includes a ‘permitted-ips’ among other ClientProfile vars to set the final value of “isAuthenticated” to either ‘true’ or ‘false’

So, for every request, it checks if client-id, client-key and the ip-address OF the calling service ARE all in the ClientProfiles Table on the AccountManagerDB. If they are, Authorization is passed and set as true else, the app will not Authorized and request will not hit the Controller.

**ACTIONS TO RESOLVE ISSUE**

**Issue: AccountManagerAPI on APIMTEST could be called without any authentication/authorization parameters and request is successful Whereas AccountManagerAPI on APIMLIVE could not be called successfully.**

Clone code from Azure git repo, branched out to debug locally.

We checked config (Web.config) files on the two deployments – test instance deployed onto 10.10.7.213 AND the live instance which is deployed on 10.10.2.100

We verified the values of Client-Id, Client-Key, ClientIpAddress from the the ClientProfiles table on the two different DATABASES (10.120.3.12.AccountManagerDB.dbo.ClientProfiles) for test AND (10.10.7.60.AccountManagerDB.dbo.ClientProfiles) for Live.

We also tried to verify if it’s the same deployment that’s hosted on both servers (test – 10.10.7.213:8090 and live – 10.10.2.100:8080). We backed up “AccountManagerAPI” deployment on TEST(10.10.7.213) and deployed what’s on production(10.10.2.100) on test expecting it to fail and expect auth headers now BUT surprisingly REQUEST was still successful.

So we deployed a debugged local version of the AccountManagerAPI onto test (10.10.7.213) to further explore the authorization. Now, logging many variables in the Application: See the log file snapshot attached below…

**2020-11-20 11:42:00.9065 INFO ClientId: APIMSandbox ClientKey: 1E0DB47D-77B3-402E-B3D3-EB52E6C989B5**

**ClientIpAddress: 172.33.10.196**

**2020-11-20 11:42:00.9495 INFO oClientProfile - BaseCore.Model.ClientProfile**

**2020-11-20 11:42:00.9525 INFO OClientProfile.ClientId: APIMSandbox**

**2020-11-20 11:42:00.9525 INFO OclientProfile.ClientDescription: API Manager Sandbox**

**2020-11-20 11:42:00.9525 INFO OclientProfile.ClientIpAddress: 172.33.10.196**

**2020-11-20 11:42:00.9525 INFO OclientProfile.ClientProfileId: 2006**

**2020-11-20 11:42:00.9525 INFO OclientProfile.ClientKey: 1E0DB47D-77B3-402E-B3D3-EB52E6C989B5**

**2020-11-20 11:42:00.9525 INFO OclientProfile.DateCreated: 9/24/2019 12:00:00 AM**

**2020-11-20 11:42:00.9525 INFO OclientProfile.DateDeleted:**

**2020-11-20 11:42:00.9525 INFO OclientProfile.Unrestricted: True**

**2020-11-20 11:42:00.9525 INFO OclientProfile.Status: A**

**2020-11-20 11:42:00.9525 INFO IsAuthenticated: True**

**2020-11-20 11:42:00.9525 DEBUG Authenticated: APIMSandbox, 1E0DB47D-77B3-402E-B3D3-EB52E6C989B5 , /api/branch/location/search , 172.33.10.196**

**2020-11-20 11:42:00.9525 INFO Request Made by172.33.10.196 to /api/branch/location/search**

**2020-11-20 11:42:01.0285 INFO location Request: lekki**

**2020-11-20 11:42:32.2451 INFO ClientId: ClientKey: ClientIpAddress: 172.33.10.100**

**2020-11-20 11:42:32.2451 INFO oClientProfile -**

**2020-11-20 11:42:32.2451 DEBUG System.NullReferenceException: Object reference not set to an instance of an object at BaseService.UtilityService. ClientAuthenticationServices.AuthenticateClient(HttpRequestMessage request) in C:\Users\GD0024\source\repos\AccountManagerAPI\BaseService\UtilityService\ClientAuthenticationServices.cs: line 51**

The logs above showed that the Public IPAddress of APIMTEST that is sending the request is 172.33.10.196 which is actually in the 10.120.3.12.AccountManagerDB.dbo.ClientProfiles table that AccountManagerAPI on TESTServer(10.10.7.213) speaks to… It also showed the ClientProfile fetch from the same table above. It also shows that isAuthenticated is ‘True” and the ClientId as “**APIMSandbox”** which is also in the database. The request was successful again.

But for AccountManagerAPI on APIMLIVE, the logs show that the Public IPAddress of APIMTEST that is sending the request is 172.33.10.196 which is actually in the 172.33.10.100 WHICH IS CONTRASTING TO information of Azure APIMLIVE stored in 10.10.7.60.AccountManagerDB.dbo.ClientProfiles. In this database, ClientIpAddress for APIMLIVE is 172.33.10.101 I THINK THE NETWORKING TEAM MIGHT BE ABLE TO EXPLAIN WHY IT IS SO.

**DISCOVERY**

**Why was AccountManagerAPI and BranchLocator on APIMTEST able to work perfectly without a Client-Id and Client-Key Header?**

**We found out in the Design Settings for AccountManagerAPI and BranchLocator API on Azure –** [**https://portal.azure.com/**](https://portal.azure.com/) **- Inner Response Headers are already set for Client-Id and Client-Key both obtained from APIMTEST info in the Database. That was the Eureka! Moment.**

**So we only have to PASS in a Client-Id and Client-Key headers for APIMLIVE also and updated the correct IpAddress(from log Info) of the APIMLIVE in the Database to get AccountManagerAPI and BranchLocatorAPI on APIMLIVE too to Work just perfectly.**

**FIX**

Revert all changes made on TEST Server.

Replaced the ClientIpAddress of the Azure APIMLIVE saved in 10.10.7.60.AccountManagerDB.dbo.ClientProfiles with the correct one (obtained from log information) – 172.33.10.100

**So we only have to PASS in a Client-Id and Client-Key headers for APIMLIVE also and updated the correct IpAddress(from log Info) of the APIMLIVE in the Database to get AccountManagerAPI and BranchLocatorAPI on APIMLIVE too to Work just perfectly.**

TEST the API again on Azure APIMTEST and APIMLIVE.

Both now work perfectly…

**N.B:**

**What exactly is the public IP Address of APIMLIVE on Azure? Is it** 172.33.10.100 (which works) Or 172.33.10.101 (which is on properties tab of APIMLIVE on Azure)?

**What exactly is the public IP Address of APIMTEST on Azure? Is it** 172.33.10.196 (which works) or 172.33.10.197 (which is on properties tab of APIMTEST on Azure)?