ACL Permission Denial Warning Analysis (from ECG)

1. Introduction

For now (SOA release 0.67.12), the SOA ACL is in permissive mode meaning that publishes and subscribes do not need to be whitelisted. Once the SOA team will switch to non-permissive mode early December, any topic not registered in a descriptor will be denied.

In order to prepare the switch to **non-permissive** mode, the teams will have to do a final check on their component to make sure ACL permissions are correct.

The recommended way to proceed relies on the fdp log with a specific filter to isolate the ACL violation for a particular component.

2. Prerequities

The ACL database is persistent. It means that it is created the first time the ECG boots up. If any change is made to the descriptors later, the database may not be updated as the descriptor will have the same version.

This applies for the ECG, TCU and SYNC descriptors.

So it is recommended to start with an newly flashed ECG with SOA release 0.67.12 at the minimum.

2.1. Clearing Service Manager Database

The ACL database is persistent so if you use the same version of the descriptor, Service Manager is not going to update it. The simplest method is to re-flash the ECG or bump the version number of the descriptor.

Alternatively, you can use the following command to restart Service Manager and clear the database:

```
# serviceManager -b / -u svcmgr --cleanSession &
```

You can confirm that Service Manager database was cleaned, by searching for the following log entry

```
<132>12018-01-01T00:35:18.247457Z - serviceManager 2601004 soa [fdp@30513tid="1"][meta sequenceId="62221"]
[serviceManager.1] 06714 - ServiceManagerDaemon: Service Manager started in clean session mode, clearing
databases
```

3. ACL Denied messages

In permissive mode, a warning is shown in the fdp log for a topic not whitelisted in a descriptor. This is an example of what can be seen:

<timestamp> - mosquitto 1359898 SoaAuthPlugin [fdp@30513 tid="1"][meta sequenceId="29170"] mosquitto_auth_acl_check(): perm denied, ACL data: { "mqtt_client_id": "ecg_200", "mqtt_user_name": "DefaultECGUser", "topic": "ecg_200/SERVICES/DATA/ECG/CM/SYNC", "access_type": "WRITE" }

There are three interesting parts. The first one (in green) highlights a way to identify the ACL violation. This will be used to setup the appropriate filter in ULT to get the ACL relevant information. Notice the **ecg_200** in this example. This is the identifier of the MQTT client that is connected to the ECG SOA broker. For clients on the ECG, it consists of the ecu name and user id of the process. UID's are in /etc/passwd or in this wiki for ECG uids: https://www.eesewiki.ford.com/pages/viewpage.action?pageId=24139093 . 200 is the uid "connmgr", so *ecg_200* is connectivityManager on the ECG.

SYNC, TCU and any other Ethernet connected ECU have only one MQTT client connected to the SOA broker, which is the SOA Gateway. It is named "sync_gateway" or "tcu_gateway" or ... <ecu>_gateway.

The 2nd part of the log highlighted in light blue is the topic. In that example, this topic is from the same client (connectivityManager). Topics in a publish are prepended with the ecuid_ud identifier of the publisher.

The 3rd part is READ for subscribe and WRITE for publish

The following table summarizes the information

Message Type	Severity	Times	Application	Tag	Process	Message
PUBLISH	Warning	N/A	mosquitto	SoaAuthPlugin	N/A	mosquitto_auth_acl_check(): perm denied, ACL data: { "mqtt_client_id": "ecg_220", "mqtt_user_name": "DefaultECGUser", "topic": "ecg_220/SERVICES/REQUEST/ECG/SPCM/CM/DID_READ", "access_type": "WRITE" }
SUBSCRIBE	Warning	N/A	mosquitto	SoaAuthPlugin	N/A	mosquitto_auth_acl_check(): perm denied, ACL data: { "mqtt_client_id": "ecg_220", "mqtt_user_name": "DefaultECGUser", "topic": "ecg_130/SERVICES/RESPONSE/ecg.fnv.net/fdpstore/1413166/100c7780/1", "access_type": "READ" }

4. Log Filters

4.1. Common Log Filter

Filters all ACL permission denied messages for any soa component

Application (optional)	mosquitto
Tag (optional)	SoaAuthPlugin
Message	mosquitto_auth_acl_check(): perm denied

Main_Filter

4.2. ECG Component Filter Template

Filters ACL permission denied messages for an ECG component

ECG_Component_Filter_Template

Application	mosquitto
Tag	SoaAuthPlugin
Message	mosquitto_auth_acl_check(): perm denied, ACL data: { "mqtt_client_id": "ecg_**UID**"

Where **UID** is a component uid. Must be the same as provided in the component descriptor file.

Example: Filter for SPCM Component

ECG_SPCM_Filter

Application	mosquitto
Tag	SoaAuthPlugin
Message	mosquitto_auth_acl_check(): perm denied, ACL data: { "mqtt_client_id": "ecg_130"

4.3. SYNC Main Filter

Filters out ACL permission denied messages for all SYNC SOA clients

SYNC_Main_Filter

Application	mosquitto
Tag	SoaAuthPlugin
Message	mosquitto_auth_acl_check(): perm denied, ACL data: { "mqtt_client_id": "sync_gateway"

4.4. SYNC Publish Errors Filter Template

Filters out ACL permission denied messages for a SYNC SOA client

SYNC_Publish_Filter_Template

Application	mosquitto
Tag	SoaAuthPlugin
Message	mosquitto_auth_acl_check(): perm denied, ACL data: { "mqtt_client_id": "sync_gateway", "mqtt_user_name": "DefaultECGUser", "topic": "sync_<**UID**>"

4.5. TCU Main Filter

Filters out ACL permission denied messages for all TCU components

TCU_Main_Filter

Application	mosquitto
Tag	SoaAuthPlugin
Message	mosquitto_auth_acl_check(): perm denied, ACL data: { "mqtt_client_id": "tcu_gateway"

TCU Publish Errors Filter Template

Filters out ACL publish permission denied messages for a TCU component

TCU_Publish_FIlter_Template

Application	mosquitto
Tag	SoaAuthPlugin
Message	mosquitto_auth_acl_check(): perm denied, ACL data: { "mqtt_client_id": "tcu_gateway", "mqtt_user_name": "DefaultECGUser", "topic": "tcu_<**UID**>"

Where **UID** is a component uid. Must be the same as provided in the component descriptor file.

5. How to analyze the log

Once you have your log, the following information may help you narrow down the issue. Remember that in any SOA communication there is a service provider and in the case of RPC transactions, there is a service consumer and provider. A SOA client may be a provider or a consumer, depending on the transaction in question to which the log pertains.

5.1.1. Not authorized to publish/WRITE

Log:

mosquitto_auth_acl_check(): perm denied, ACL data: { "mqtt_client_id": "ecg_200", "mqtt_user_name": "DefaultECGUser", "topic": "ecg_200/SERVICES/DATA/ECG/CM/SYNC", "access_type": "WRITE"

Cause:

Connectivity manager publishes on a topic to which it doesn't have write access. Three possibilities :

1. If connectvity mgr is the service provider and the topic is owned by *connectivity mgr,* then it is a data endpoint that has not been listed in connectivity mgr's SOA descriptor. This is likely the case in the example above since the topic is named to represent a data topic. Descriptor would be something like :

```
   <services>
    <shared-service id="FNV_SYNC_CM_CTRL">
      <data-endpoints>
        <data-endpoint>SERVICES/DATA/ECG/CM</data-endpoint>
      </data-endpoints>
    </shared-service>
  </services>
```

2. If connectvity mgr is the service provider and the topic is not owned by *connectivity mgr ,* then it is the service consumer's response topic on which it needs to publish a response to an RPC call. In this case, the omission is in the consumer's SOA descriptor in which it needs to request access to the appropriate connectivity mgr service.

3. If connectivity mgr is the service consumer, then the topic is not owned by *connectivity mgr.* It is the service provider's request topic on which connectivity mgr has to publish a request. In this case *connectivity mgr* has not requested permission to use the soa service in the soa-consumer-permissions section of the SOA descriptor. Something like :

```
<soa-consumer-permissions>

     <endpoint>SERVICES/RESPONSE/ECG/CM</endpoint>
         <soa-service-requests>
        <soa-service-request service-id="FNV_TCU_CM_CTRL" access="FULL"/>
                :
         </soa-service-requests>

</soa-consumer-permissions>
```

5.1.2. Not authorized subscribe/READ

Log:

mosquitto_auth_acl_check(): perm denied, ACL data: { "mqtt_client_id": "ecg_200", "mqtt_user_name": "DefaultECGUser", "topic": "tcu_2007/SERVICES/DATA/TCU/CM/ECG", "access_type": "READ" }

Cause: multiple cause are possible.

1. For service consumers, it could indicate a denial to read a service provider's data topic which is the case for the log above and would indicate a missing service request in the SOA descriptor as illustrated below.

```
  <soa-permissions>
    <soa-consumer-permissions>
      <endpoint>SERVICES/RESPONSE/SYNC/CM</endpoint>
      <soa-service-requests>
        <soa-service-request service-id="FNV_TCU_CM_CTRL" access="FULL"/>
      </soa-service-requests>
    </soa-consumer-permissions>
  </soa-permissions>
```

2. For service consumers, it could indicate a denial to read it's own response topic. This would indicate an error in specifying the response endpoint in the "soa-consumer-permissions" section of the descriptor.

```
<soa-permissions>
    <soa-consumer-permissions>
      <endpoint>SERVICES/RESPONSE/SYNC/CM</endpoint>
      <soa-service-requests>
        :
      </soa-service-requests>
    </soa-consumer-permissions>
  </soa-permissions>
```

3. For service providers it could indicate a denial to read it's own request topic. This would indicate an error in specifying the request-endpoint in the "service" section of the descriptor.

```
<service id="ECG_POWER_REBOOT_REQUEST" group-id="ECG_POWER_SERVICES">
    <request-endpoint>SERVICES/REQUEST/ECG/POWER/REBOOT</request-endpoint>
    <data-endpoints>
        <data-endpoint>SERVICES/DATA/ECG/POWER/STATE</data-endpoint>
    </data-endpoints>
</service>
```