Pub-Sub Profile for Authentication and Authorization for Constrained Environments (ACE)

draft-ietf-ace-pubsub-profile-02
Cigdem Sengul
cigdem.sengul@brunel.ac.uk

IETF 110
March 12, 2020
Updates to the document

• Restructured to describe CoAP and MQTT solutions

• Described MQTT case
  • Generally similar to CoAP client
  • Differences:
    • Publisher/Subscriber Clients are not separate
    • Subscriber Clients are also authorised

• Remaining
  • Incorporating changes in Scope parameter (AIF-MQTT etc.)
  • Resolving discussion points
Discussion: Architecture

Current Architecture

Pros
- AS2 can authorise and hand out keys in one request (token + keys) – however client may still need to contact AS before to learn algorithms etc.

Cons:
- AS1 and AS2 should have synchronised policies
- Subscriber authorisation can be set-up but not supported by default.

Proposed Change

Pros
- Can support single group policy
- AS can be flexible, two separate ASes or single AS (policy synchronisation is not a must, it’s a choice).
- Subscriber-authorisation supported by default
- May be simpler for nodes that are both pub and sub

Cons
- Need to get a token from AS to talk to KDC

Question: Single token for multiple use?
Discussion: Policy Synchronisation

Point

• Problem with AS1 and AS2 as being independent appliers of access control logic without any communication between them. AS1 needs the ability to give policy to AS2 on a topic after it has been created and before any subscribers get keys. In the case they are co-resident this is trivial; in other cases it may not be.

• If the publisher loses its membership in the group for any reason, what happens? When group membership changes, both should change/become invalid
  • Permissions towards broker
  • Permissions towards KDC

• Whose responsibility it is to revoke rights, AS1 or AS2?

Counter-point:

• AS1 and AS2 have clearly separated functions. There is some coordination involved (to gain knowledge of the policies), but this can be dealt as application specific.

• Revocation should be handled, but as a WG-level general solution.
Discussion: Group Join Request

- In groupcomm
  - Authorisation request scope may have multiple topics (groups)
  - Group join request is per group/topic

- Group join request to multiple topics (groups)?
  - mqtt using AIF = ["topic1", ["pub","sub"]], ["topic2/#","pub"], ["+/topic3","sub"]]
  - There needs to be a separate request for each topic filter.
  - In MQTT, topics are organized in topic trees. Depending on how topics are grouped, the KDC may have different sections of the tree keyed differently.
  - Subscription requests may include wildcards spanning several levels of the topic tree.
  - Two things may happen:
    - An MQTT node may be returned keys for a wider set of topics (groups) that their token permits them. However, since the Broker authorises all Clients (regardless of their role is only Publisher or Subscriber), the Clients cannot access any messages sent for a topic beyond their token's scope.
    - The Join request spans multiple groups? (need to fetch groupnames using topic filters as gids?)