Security in complex APIs
Considerations and Pitfalls

Hylke van der Schaaf, Fraunhofer IOSB
2023.06.07
Level 0
All or Nothing

- A user can either access the service (and do anything) or not at all
  - Simple to solve with a reverse proxy
Level 1
Action Based

- Per action policies: C, R, U, D
  - Solvable with a reverse proxy: rights on
    - GET (read)
    - POST (create)
    - PUT/PATCH (update)
    - DELETE (delete)
  - BEWARE: Some API features break this:
    - Batch-processing allows read, update & delete with a POST
Level 2
Entity-Type Based

- Per action policies: C, R, U, D
  On a per-Type basis
  - Solvable with a reverse proxy: rights on
    - GET / POST / PUT / PATCH / DELETE
  - Based on the last section of the Path
    - /Observations
      - /Datastreams(x)/Observations
      - /FeaturesOfInterest(x)/Observations
  - BEWARE: Some API features break this:
    - Batch-processing allows read, update & delete with a POST to v1.1/$batch
Level 3
Entity Based

- CRUD-access based on individual entities or related entities
  - User S can create and read Observations for Datastreams of Thing-1, but nothing else
  - Can not be solved through a proxy, especially read restrictions due to indirect data leaking
Level 3: Read
The Problem

- ProxyFail Example 1: expand

1) User A can read:
   - Thing-1,
   - Datastreams of Thing-1 (DS-1),
   - ObservedProperties of these Datastreams (OP-1)

2) User A executes query:
   ObservedProperties?$expand=Datastreams($select=name)

3) Proxy will have a very hard time determining which of the embedded Datastreams should not be there
Level 3: Read
The Problem

- ProxyFail Example 2: filter

1) User A can read:
   - Thing-1,
   - Datastreams of Thing-1 (DS-1),
   - ObservedProperties of these Datastreams (OP-1)

2) User A executes query
   ObservedProperties?$filter=Datastreams/id gt 1

3) Result is ObservedProperty 1, which User A is allowed to read, so proxy allows the result

4) User A now knows there is at least 1 Datastream with an id greater than 1
Level 3: Read

Solutions

- Must be solved at a low (database) level
  - Per user group database views
  - Row-Level-Access in PostgreSQL
  - Query-modifications in the server
- This is not specific to STA
  - Relevant for any system that can filter across relations
Level 3: CUD
Many Questions

- Can a User
  - Create Entities of EntityType-X (new Observations)
  - Link new Entity-X to Entity-Y (new Observation in DS-1)
  - Update properties of Entity-X (Patch/Put Observation-1)
  - Change a link of Entity-X from Entity-Y1 to Entity-Y2 (Move Observation-1 from DS-1 to DS-2)
  - Delete Entity-X
FROST-Server Implementation
A prototype

Extended data model
FROST-Server Implementation
A prototype

- Extended data model
- Two user-rights levels
  - Global
    - Users have 0..* (global)Roles
  - Project related
    - Things assigned to Projects
    - Users have 0..* (project)Roles
      in 0..* Projects
FROST-Server Implementation
A prototype

- Admin-Only EntityTypes
  - visible for global-admin users
  - do not exist for others
- Global Roles apply everywhere
- Global Read can read all
- Global Create can create all
FROST-Server Implementation
A prototype

- Project Roles for details

- Users can read Observations in Datastreams of Things linked to a Project they have any role in.

- User with Role obsCreate on ProjectX can create Observations in Datastreams of Things linked to ProjectX
FROST-Server Implementation
A prototype

- Read restrictions not yet applied to MQTT
- Data model and security rules configurable using JSON files
Questions?

Dr. Hylke van der Schaaf
Information Management and Production Control
hylke.vanderschaaf@iosb.fraunhofer.de

Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB
Fraunhoferstraße 1
76131 Karlsruhe, GERMANY
www.iosb.fraunhofer.de