MIT Subject Management API - System rules for Security

Subject Management API - System rules for Security

Please note that this page and related pages are being developed as part of the CIM Courses Project and are subject to change.

The Subject Management API - System rules for Security was built as part of the CIM Courses Project. This article describes the security system rules implemented in Subject Management API.

Table of Contents:

- Subject Management API - System rules for Security
  - Brief Overview
  - Security Measures
    - Client ID Enforcement
    - Roles Policy (v2.0.0)
    - MIT Roles Service
  - Authorizations
    - Authorization rules
    - Authorization rules to limit applications' access:
      - Backfill_CIS Role
    - Authorization Process Technical Overview
  - Developer's notes
  - Subject Management Documentation Index

Brief Overview

Subject Management API contains multiple endpoints to view and update the subject data. The API restricts access to these endpoints, limits updating only certain the subject data component(s) via some endpoints, and also allows viewing and browsing all or a subset of subject data by an individual user. For example, the term plan app is used to view and update only the term plan and display control data of a subject. Among all users that can access the term plan app, some users can only view and update subject data of department(s) that they are authorized to access and some users can view and update subject data of all departments.

Security Measures

The Security for Subject Management API is accomplished via implementing client id enforcement and roles policy in API Manager under Mule Anypoint Platform. The client id enforcement allows only authorized applications to access the endpoints available under the Subject Management API. The roles policy implementation allows only individual user who is granted specific user role(s) in MIT Roles Services to access specific endpoints available under Subject Management API. Besides roles for a user, a different type of role (client role) can be configured with each authorized client id via the roles policy. In addition, security in Subject Management API are configured at different tiers. Depending on our needs, we can have different security measures enforced in each tier.

Both Client ID Enforcement and Roles Policy are managed (enable, disable or remove) via the Policies menu under the API Manager in Mule Anypoint Platform.

Client ID Enforcement

Each application or API that wants to utilize endpoints available under Subject Management API has to request access via the Mule Anypoint Platform Exchange. Before requesting access, the application or API must configure as an application under Mule Platform with its own client id and client secret. After the request of access is approved, the applications' client id(s) can be found under Contracts in Mule API Administration.

Roles Policy (v2.0.0)

Roles Policy, a customized policy available in API Administration for MIT mule app, is used to configure the authorization of a client id and the category of roles service authorizations the Roles policy to gather for a requesting user. Both the client id role(s) and the user role(s) from the roles service in specified categories are then made available to the API via the a flow variable. In this way, APIs can use authorizations to make access decisions without making calls to the Roles API. Currently, the authorization configured for all client id(s) is NONE:TRUSTED_CLIENT and the categories of role database authorizations to retrieve for each user are REG and SIS in test and production tiers. Also, several client id are assigned client role: BACKFILL_CIS via the roles policy.
MIT Roles Service

The MIT Roles Service is used to grant roles for an individual user based on their kerberos id. Currently, users who can access the Subject Management API endpoints will have one of the following roles in specific category (category shown in parenthesis):
- SCHED_REGISTRAR(REG), SCHED_DEPT_ADMIN(REG), SDM_REGISTRAR_UPDATER(SIS), SUBJ_DEPT_ADMIN(REG), SUBJ_DEPT_ADMIN_FSEM(REG) and SUBJ_REGISTRAR(REG). These roles can be classify into 2 main categories:
  - roles that have access to subject data of all departments
    - include SCHED_REGISTRAR (REG), SDM_REGISTRAR_UPDATER(SIS), and SUBJ_REGISTRAR(REG)
    - qualifier_code is not required for these roles.
  - roles that can access only subject data of specific department(s)
    - include SCHED_DEPT_ADMIN(REG), SUBJ_DEPT_ADMIN_FSEM (REG) and SUBJ_DEPT_ADMIN(REG)
    - qualifier code which defines access to specific department(s) is required to be configured for these roles
    - access to multiple departments are configured as a comma-separated string

For details on roles that are used and their respective accesses to applications that are supported by the Subject Management API, please refer to the Subject Management Authorizations document.

Authorizations

Authorization rules

The Subject Management API has implemented the following authorization rules:

1. Subject Data Update Authorization
   This authorization rule used to determine if a user or a client can update all components of subject data. A request user or a client id (which represents a client application) which has either the SDM_REGISTRAR_UPDATER(SIS) or the SUBJ_REGISTRAR (REG) role is authorized to update all components related to a subject.

2. Subject Term Plan and Display Control data Update Authorization
   This authorization rule used to determine if a request user or a client id can update the term plan and display control data of a subject. A request user or a client id which has the SCHED_REGISTRAR(REG) role can update term plan and display control data of any department. A request user or a client id which has the SCHED_DEPT_ADMIN(REG) role can only update subjects of specific department(s) defined in the qualifiers associated with the role. The process will check if the client id or user has the all department role first before proceeding to check if the user has the department specific role.

3. Backfill CIS Authorization
   This authorization rule used to determine whether the subject data update initiated by a client id be backfilled to the CIS legacy tables. The client id that is eligible for CIS backfill must have the role: BACKFILL_CIS(REG).

4. Subject Data View Authorization
   This authorization rule includes processes to determine if a request user accessed the Subject Management API via a client id can view data of a subject.
   a. A request user can view subject data of any department if the user in the header
      i. has the SCHED_REGISTRAR(REG) role and accessed the API via the term plan application ui client id
      ii. has the SUBJ_REGISTRAR(REG) or SDM_REGISTRAR_UPDATER(SIS) role and accessed the API via the subject management ui client id.
   b. A request user can view subject data of a specific department if the user in the header
      i. has the SCHED_DEPT_ADMIN (REG) role and the department of the requested subject is equal to one of the departments specified in the qualifier for the role
      ii. has the SCHED_DEPT_ADMIN (REG) role and the department of any cross list subject (if there is any) of the requested subject is equal to one of the departments specified in the qualifier for the role.

Authorizations rules to limit applications’ access:

The Subject Management API enforces authorizations for all post endpoints and one get endpoint. The following are authorization rules implemented to limit application’s access to various endpoints:

1. Subject Data Update Authorization has been enforced in the following endpoints:
   a. Create Subject
   b. Edit Subject
   c. Subject Deactivation
   d. Subject Reactivation
   e. Requisite update
   f. Backdate subject effective term (SCASUBJB)

2. Subject Term Plan and Display Control Data Update Authorization has been enforced in the following endpoints:
   a. Subject Term plan data update
   b. Subject Display control data update

3. Subject Data View Authorization has been enforced for one get endpoint for the term plan app:
   a. Search by subject number and term
Backfill_CIS Role

1. This is a role that is granted to client id(s) whose post request data is backfilling to CIS tables. The following client id(s) are assigned this role via the Roles Policy:
   a. Term Plan UI
   b. Courseleaf’ CIM Client

Authorization Process Technical Overview

1. All the processing for authorization can be found in mule config file: subjects-api-security.xml. The following flows contain the implementations of the various authorization rules in the Subject Management API:
   a. Subject Data Update Authorization : SUBJ_SECURITY-check-if-authorized-to-create-and-update
   c. Subject Data View Authorization: check-if-authorized-to-search-subject-by-component

2. Mule Roles Policy created a flow variable: flowVars.authorization which contains roles of the requested user and the client id. The flow variable is accessible by the Subject Management API for processing the authorization logic.

3. In the secure-<tier>.properties file, configure the following property that specify the role the API use to determine if a client id is authorized for CIS backfill:
   a. backfill.process.role.authorized.for.cis.backfill=REG:BACKFILL_CIS

4. In the secure-<tier>.properties file, configure the following properties that specify roles with different access level - i.e. admin user roles with access to all departments and departmental user roles with access to only particular department:
   a. roles.authorized.termplan.all=REG:SCHED_REGISTRAR
   b. roles.authorized.termplan.qualifier=REG:SCHED_DEPT_ADMIN
   c. roles.authorized.search.all=REG:SCHED_REGISTRAR,REG:SUBJ_REGISTRAR,SIS:SDM_REGISTRAR_UPDATER
   d. roles.authorized.search.limit.by.qualifier=REG:SCHED_DEPT_ADMIN,REG:SUBJ_DEPT_ADMIN,REG:SUBJ_DEPT_ADMIN_FS
   e. roles.authorized.search.all.termplan=REG:SCHED_REGISTRAR
   f. roles.authorized.search.limit.by.qualifier.termplan=REG:SCHED_DEPT_ADMIN
   g. roles.authorized.search.all.sms=REG:SUBJ_REGISTRAR,SIS:SDM_REGISTRAR_UPDATER
   h. roles.authorized.search.limit.by.qualifier.sms=REG:SUBJ_DEPT_ADMIN,REG:SUBJ_DEPT_ADMIN_FSEM

5. If a user or a client has multiple relevant roles, the admin roles will supersede roles that authorized only for specific departmental access. For example, a user or client id has SCHED_REGISTRAR role is authorized to update the term plan and display control data of a subject of any department even though the same user or client id is also assigned the SCHED_DEPT_ADMIN role with access only for department 8 subjects. For user or client id that only has departmental access roles, one of qualifier(s) specified for role must match the subject department of the edited subject

Developer's notes

1. Since the roles policy is not available to developer's local environment, the API is implemented to create a mock authorization object for endpoints that authorization are implemented when it detects the API is running in a developer local environment. The mock authorization object contains the relevant admin role to allow full access to respective endpoint. The following are the flows that create the mock authorizations for various authorization rule:
   a. create-mock-up-authorization-for-testing - create mock authorization for endpoints that will enforce the Subject Data Update Authorization and Subject Data View Authorization
   b. create-mock-up-authorization-for-termplan-app-testing - create mock authorization for endpoints that will enforce the Subject Term Plan and Display Control Data Update Authorization and Subject Data View Authorization

   The mock authorizations can also be modified to test and verify the authorization implementation itself.

1. API manager configuration - for more information on how to configure the Mulesoft API roles policy in Mulesoft API Roles Policy, please refer to document: Mulesoft API Roles Policy v2. Please note, only authorized user can access the document

Subject Management Documentation Index

The Subject Management Documentation Index is the central listing for documentation pertaining to Subject Management.