Component: ID and Access Management

Table of Contents

- Overview
  - Responsibilities
  - Component Interfaces
- Configuration
  - Keycloak Realm
    - References
  - Keycloak Clients
    - Figure 1: KeyCloak Clients
    - References
  - User Accounts
    - References
  - Logging and Auditing
  - Nonfunctional Requirements

Overview

This page provides a detailed design for the Identity and Access Management (IdAM) Component of the COPS Platform. It is based around the use of KeyCloak.

Responsibilities

- Authenticate users via an external IdAM system (such as a Directory Server) or with internal user/group configuration
- Retrieve Authorization information from an external IdAM system (such as a Directory Server)
- Respond to queries about accesses for specific users
- Handle login across multiple applications

Component Interfaces

Incoming connections on TCP Port 443 (from services requesting authentication)

Outgoing connections to TCP Port 514 (Logs sent to rsyslog)

Configuration

Keycloak is the central application for this component. It includes a single Realm with multiple clients.

The external IdAM connection will not be configured during the initial setup. That will wait until integration on the customer's network.

Keycloak Realm

All configuration should be done in a new Realm named "mlsapi" (not the 'master' Realm, unless the Keycloak documentation requires it).

References

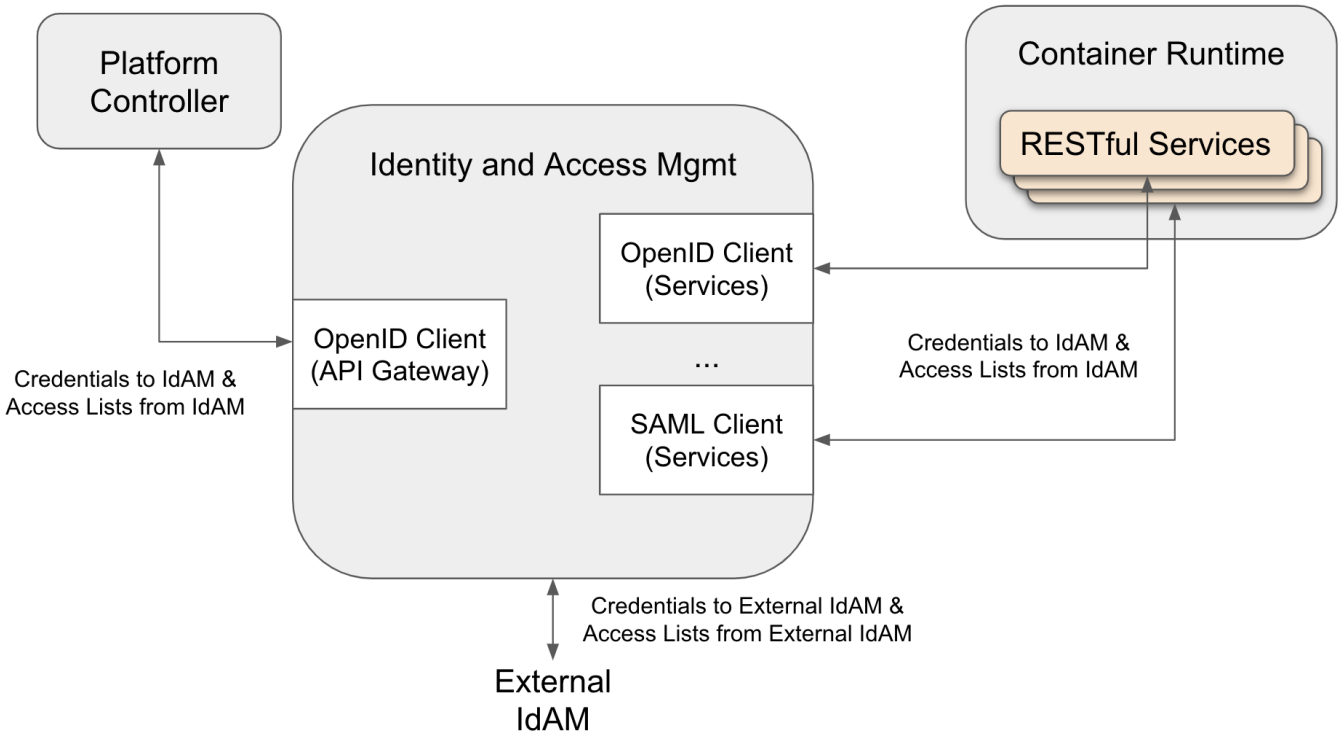
- Creating a Realm: https://www.keycloak.org/docs/latest/server_admin/#_create-realm

Keycloak Clients

Figure 1 shows the anticipated clients.

1. Configure an OpenID Client for use by the Platform Controller (according to documentation provided by the Platform Controller).
2. Configure at least one example OpenID Client and one SAML Client in KeyCloak for integration testing with simple RESTful Services in the Container Runtime. (Use sensible placeholder values for all required fields, they will be updated during integration testing)

Figure 1: KeyCloak Clients



References

- Keycloak Documentation: https://www.keycloak.org/documentation.html
- Client configuration: https://www.keycloak.org/docs/latest/server_admin/#_clients
- SSO Protocols: https://www.keycloak.org/docs/latest/server_admin/#sso-protocols
- External users (LDAP or Active Directory): https://www.keycloak.org/docs/latest/server_admin/#_user-storage-federation

User Accounts

In the customer's environment, these are expected to be sourced from an external IdAM provider.

1. For integration and testing purposes, create the following test users.
    a. Tester #1
        i. **Username**: tester
        ii. **First Name**: Tester
        iii. **Last Name**: One
        iv. **Attributes**:
            1. **sensitivity_range**: "s0-s4"
            2. **categories**: "c0,c12,c42"
    b. Tester #2
        i. **Username**: testertwo
        ii. **First Name**: Tester
        iii. **Last Name**: Two
        iv. **Attributes**:
            1. **sensitivity_range**: "s0-s6"
            2. **categories**: "c2.c15"

For integration and testing, these provide a way to validate that only Tester #2 can access data at sensitivity level s5, and only Tester #1 can access data with category c42 (as long as the sensitivity level is s4 or lower).

When accounts are imported from an external IdAM, an attribute mapper should be configured to use these same two attribute names. If the levels and categories are not in this format, then an extension or application will need to be developed to translate.

References

- Create a new user: https://www.keycloak.org/docs/latest/server_admin/#_create-new-user
- Add user attributes: https://www.keycloak.org/docs/latest/server_admin/#user-attributes
- MLS Statements (explaining sensitivity and category): https://selinuxproject.org/page/MLSStatements
- User Attribute Mappers: https://www.keycloak.org/docs/latest/server_admin/#sync-of-ldap-users-to-keycloak

Logging and Auditing

All logs must be sent to the external Log Aggregator (currently expected to be rsyslog).

Nonfunctional Requirements

- Configuration details from the initial implementation must be documented on Confluence. Eventually, the install/configuration will be automated.