Texto Introductorio GBAC

GBAC (group-based access control)

Como vimos en clases anteriores, **Keycloak** nos permite administrar grupos (**>groups**) dentro de nuestro **reino** que los usuarios pueden formar parte de un **group** ya sea para representar su función dentro de una unidad de negocio de la organización o para agruparlos de acuerdo a los roles que cumplen dentro del contexto de la aplicación —como podría ser, por ejemplo, un grupo de usuarios administradores—.

Cuando utilizamos este tipo de políticas de control de acceso entra en juego GBAC (Group-Based Access Control), asignando roles a los diferentes groupos. De esta manera, **Keycloak** hace mucho más fácil la tarea de administrar roles comunes para múltiples usuarios sin obligarnos a otorgar o revocar roles a cada usuario de forma individual dentro del reino.

Los **grupos** dentro de Keycloak son jerárquicos y, cuando se emiten los **tokens**, se puede atravesar esta jerarquía observando el path del grupo.

Supongamos un ejemplo: tenemos un grupo de recursos humanos llamado **“human resource”**. Como hijo de este grupo, tenemos un grupo de administradores llamado **“manager”**.

Cuando Keycloak almacena información sobre grupos dentro de los tokens, la información debería llegar en el siguiente formato: **/human resource/manager**. Esta información estará disponible para cada token emitido por el servidor, donde el sujeto (usuario) sea miembro del group.

A diferencia de los roles, la información sobre el grupo no se incluye de forma automática dentro de los tokens, sino que debemos asociar un protocolo de mapeo al cliente (o un client scope con el mismo mapeo) como lo hicimos en clases anteriores. **¿No te acordás cómo hacerlo?** Te recordamos que podés encontrar los videos paso a paso y los tutoriales en PDF en clases anteriores.

Como este tema ya lo vimos en detalle anteriormente, en esta sección solo recordamos un poco su uso y la fundamentación teórica que posee.