Advanced Hybrid RBAC

Introduction

midPoint is using a (very) extended version of Role-Based Access Control (RBAC) mechanism. RBAC is originally defined as mostly static structure of users and roles. The original RBAC defines that user assigned to the role gets all the rights implied by the role. If two users have the same role, they have the same rights. However, this leads to the problem of Role Explosion. We hope to solve that problem by enhancing RBAC model with logic. We add ability to specify expressions in role definitions that determine how and when the role is used. Therefore the role can adapt to the attributes of user that has the role or even the role assignment itself can be parametrized. This allows to construct RBAC structures that are using fewer roles and that are much more manageable.

Roles

Roles define similar set of access rights. Roles are assigned to users; a user having a role gets the rights defined by the role. We do not place any constraints on the number of roles assigned to the user or the number of access rights (accounts) defined by the role. All the associations can be thought of as many-to-many. Basic role structure is illustrated in a following diagram.

Figure: User-Role-Resource

Traditional RBAC Is Not Enough

The usual use of roles is to imply accounts on the resources and therefore ease the user and account administration by employing RBAC approach. However, this is usually not enough to implement efficient and maintainable identity management solution. The simple approach leads to role explosion. The roles needs to contain parts of logic to make them efficient.

Implied Account Attributes

MidPoint role can also specific attributes for the account, e.g. a specific text in the account description field. Attribute values implied by the roles may be fixed (static), but that is usually not sufficient to avoid a role explosion problem. More frequently the implied attributes are derived from other values, e.g. fields of the User object. Mappings and expressions are used to define dynamic implied account attributes. The expression model is extensible, there are several supported scripting languages and even more expression languages may be added in the future.

Implied account attributes usually do not need to define the entire set of account attributes. There may be other roles that may assign different attributes to the same account, more values to the same attributes of the account and even conflicting values. The account may also have existing attributes that are managed by "native" tools (outside IDM) or there may be exceptions from the RBAC policy specified for that account using attribute specification in assignments.

Implied Account Entitlements

> **Work in progress**
> Entitlements are still work in progress. This section describe design of a feature that will be implemented in later midPoint releases.

But perhaps the most useful feature of roles is that a role can imply entitlements of account on the resource. E.g. the role can imply that the account of a user having such role will be entitled for (assigned to) the group managers on a specific LDAP server. This feature is quite seamlessly merging the concept of roles in an RBAC sense with a concept of resource entitlements such as groups and privileges. This part of

identity management configuration was quite a difficult task in the past. MidPoint aims at substantial improvement in this field.

Role Hierarchy

Roles can naturally contain other roles therefore creating a role hierarchy. Role hierarchies can be quite complex both in their structure and embedded logic. As midPoint uses the relative change model it is quite easy to merge values from many roles and therefore it allows creation of very complex RBAC structures. Important parts of the hierarchy are exposed to the expressions in individual roles therefore the role hierarchies can be combined with parametric roles (see below) to support very complex and flexible RBAC-like models.

Figure: Role Hierarchy

Parametric Roles

Roles in traditional identity management systems can only be simply assigned to a user or unassigned from a user. And that's all the flexibility. However this is not enough to efficiently model complex real-world scenarios. For example the role of `Assistant` can have some generic parts that are common to each assistant but there may be few parts that are specific for each sub-group of users or even for each individual user. For example identification of a building or department for which the assistant works, date of role activation and deactivation, the financial limit that an assistant is authorized to handle, etc. In traditional systems this leads to a necessity of creating roles such as `AssistantNewYork`, `Assistant London` and `AssistantBratislava`. This alone is quite difficult to manage because there is also need to `ClerkNewYork`, `ClerkLondon` and `ClerkBratislava` and the same for office manager, purchasing manager, ... And when it comes to roles such as `PurchasingManagerAssistant2013NewYork5000` it is quite sure that the solution got a severe role explosion problem.

Parametric roles provide a solution for some of the role explosion situations. Parametric roles allow to specify certain role parameters at *assignment time*. That means that one `Assistant` role is usually enough. The identification of branch, building, department, activation and deactivation dates and other role parameters is specified when the role is assigned to the user. The parameters can vary for each user. The expressions in the `Assistant` role can use such parameters and determine the specific account attributes and entitlements dynamically. The assignment parameters correspond to the concept of *contract* or *affiliation* that are frequently used in business modelling.

This approach dramatically reduces the number of roles needed for the IDM solution and makes the entire RBAC deployment considerably more manageable.

> **Partially supported**
> Parametric roles are fully supported by midPoint core (the "engine" or "IDM Model"). But user interface support for parametric roles is still missing. Parametric roles are inherently flexible and customizable thing. Therefore the user interface cannot be hardcoded to support them. User interface needs to adapt to parameters, that can be different for each and every role. Support for this method is feasible, but it is just not implemented yet because nobody had funded such development. In case that you are interested in funding user interface support for parametric roles please consider purchasing a subscription.

Meta-Roles

> This feature is available in midPoint version 3.0 and later.

Things may get really complicated when IDM solution is meant to synchronize much more than just users and accounts. And midPoint is designed to be very generic about what it synchronizes. E.g. an IDM solution may want to create groups on resource as an representation of midPoint roles. But how does midPoint know on which resources the groups should be created? And how they should look like? This is both easy and complex but there is a very elegant solution. MidPoint already has a mechanism for this: RBAC. And by following midPoint approach we try to apply existing mechanisms as much as possible and practical. Therefore we have applied the mechanism or roles to the roles themselves. Thus creating a concept of meta-roles (and meta-meta-roles and meta-meta-meta-roles, ...) This may sound crazy but it in fact a very elegant and powerful mechanism. See Roles and Policies Configuration and Generic Synchronization for more details.

See Also

- Roles and Policies Configuration section in Documentation
- Assignment
- Mappings and Expressions

External links

- What is midPoint Open Source Identity & Access Management
- Evolveum - Team of IAM professionals who developed midPoint