Example Dynamic role PEP

Dynamic role Enforcement Point

Use case example

We want to define a policy to restrict access to the Soffid console user's page (MainMenu > Administration > Resources > Users).

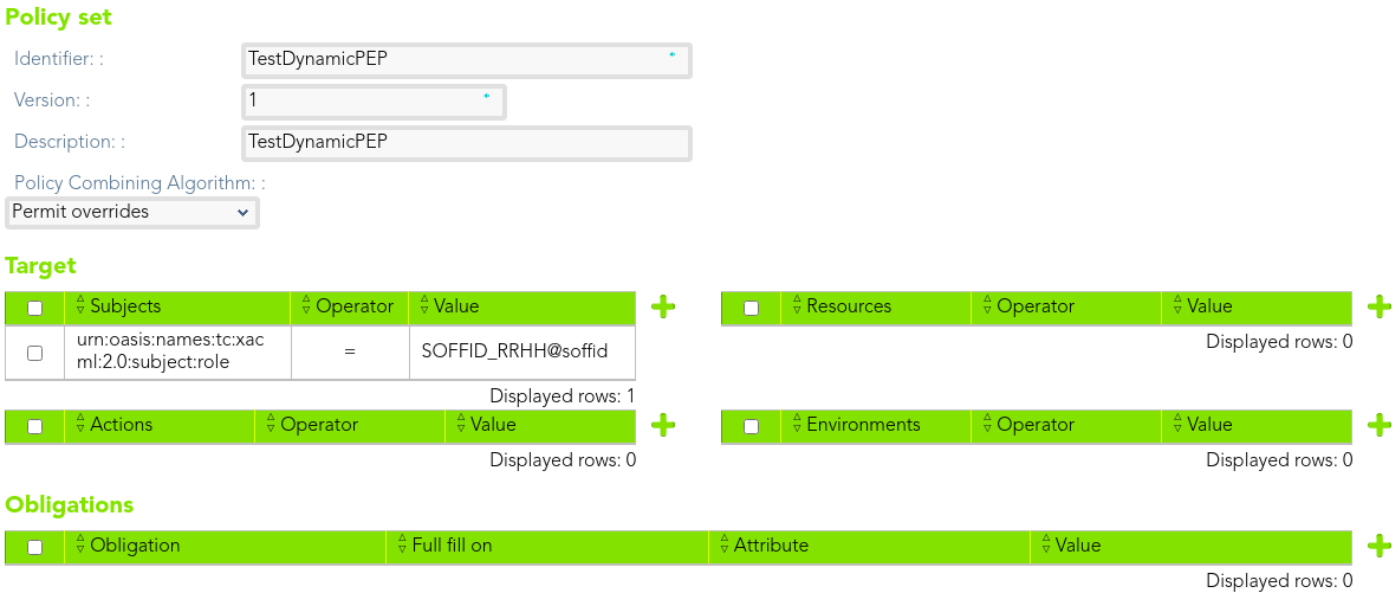
The users who are assigned to the SOFFID_RRHH role (from this point forward: *end-users*) will have limitations to perform some actions on the Soffid console users page:

1. The *end-users* only be able to query the information about the users who belong to the same primary group that them.
2. The *end-users* only be able to update the users with internal user type.
3. The *end-users* could not delete any user record.

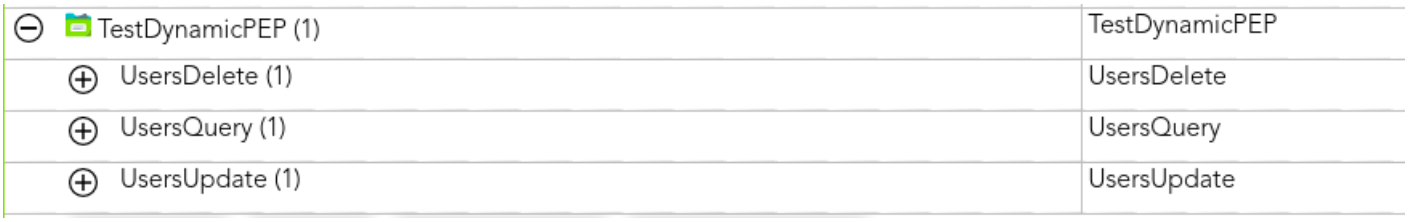
XACML Editor

Policy set

First of all, we define a policy set. We need to define the subject, in that case users with SOFFID_RRHH role assigned.



Then, we can create the policies, in that case, we can create three policies, one for each operation that we want to manage.



Policies

We can define a policy for each operation, to permit or deny access.

Also, we can define a variable that contains the *end-user* primary group in order to use it when we define the conditions.

Policy1

> The *end-users* only be able to query the information about the users who belong to the same primary group that them.

We need to define two rules, one to permit and other to deny access.

Policy

Identifier: : UsersQuery

Version: : 1

Description: : UsersQuery

Rule Combining Algorithm: : Permit overrides

Target

Subjects	Operator	Value

Displayed rows: 0

Resources	Operator	Value

Displayed rows: 0

Actions	Operator	Value

Displayed rows: 0

Environments	Operator	Value

Displayed rows: 0

Variables

Variable	Expression
group	primary group

Displayed rows: 1

Rules

Rule	Description	Effect
PermitQuery	PermitQuery	Permit
DenyQuery	DenyQuery	Deny

Displayed rows: 2

Obligations

Obligation	Full fill on	Attribute	Value

Displayed rows: 0

Rules

We define the rule that allow to the *end-user* to query users information who belong to the same primary group that the end-user.

Rule

Rule : PermitQuery

Description : PermitQuery

Effect : Permit

Target

Subjects	Operator	Value

Displayed rows: 0

Resources	Operator	Value
com:soffid:iam:xacml:1.0:resource:soffid-object	=	user

Displayed rows: 1

Actions	Operator	Value
urn:com:soffid:xacml:action:method	=	query

Displayed rows: 1

Environments	Operator	Value

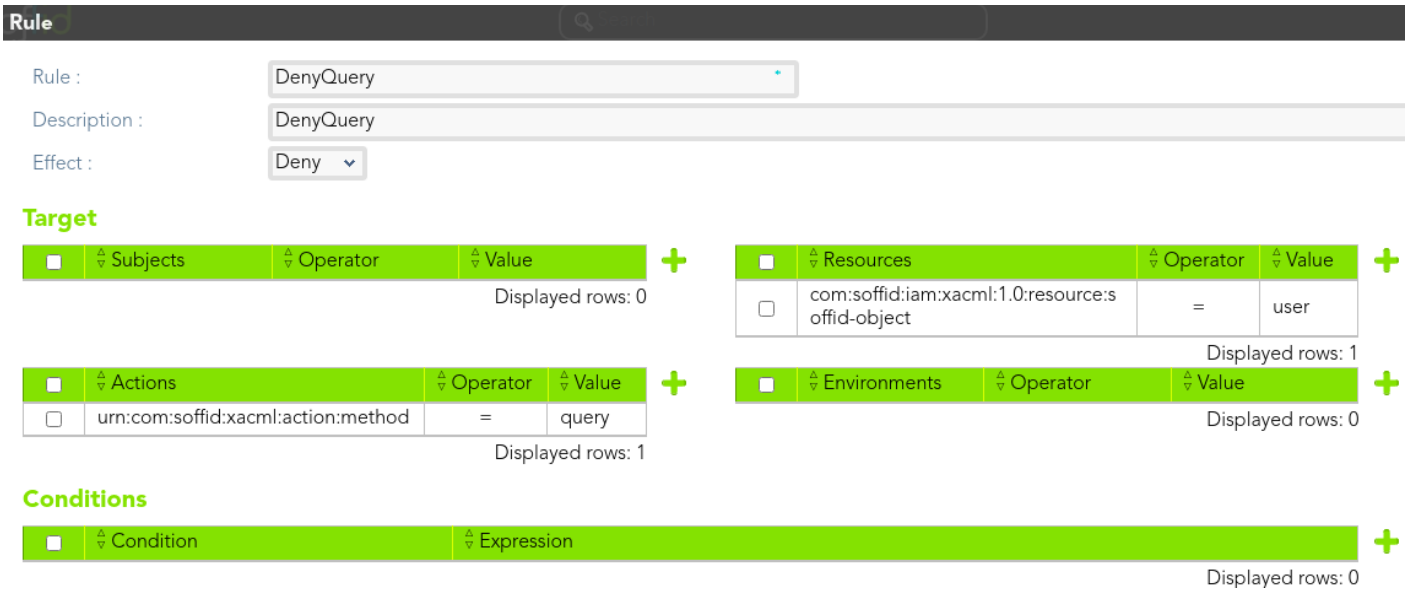
Displayed rows: 0

Conditions

Condition	Expression
Condition	(One and only(group) == One and only(null /primaryGroup/name))

Displayed rows: 1

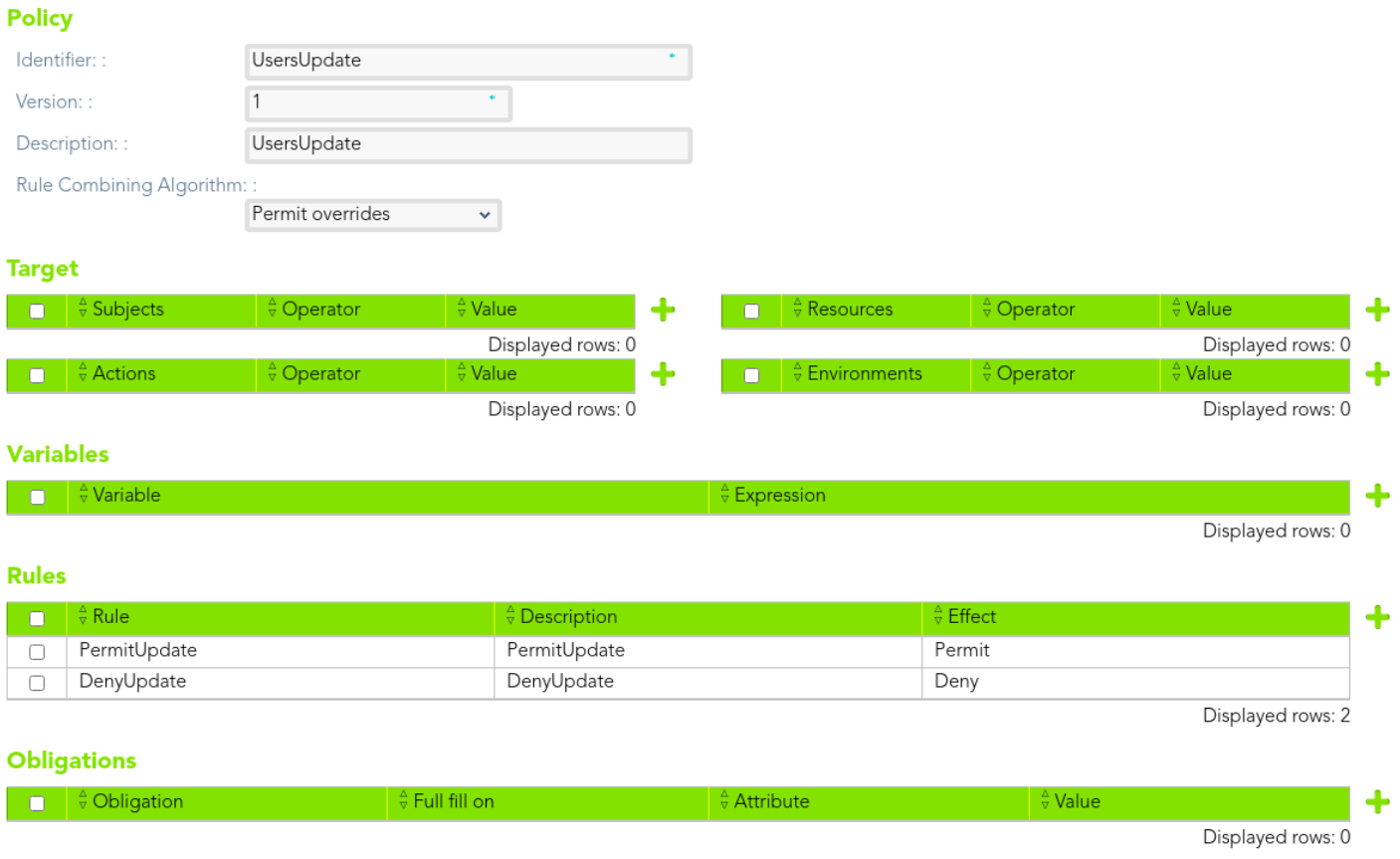
Then, we define the rule that denies access to *end-users* to query users information.



Policy 2

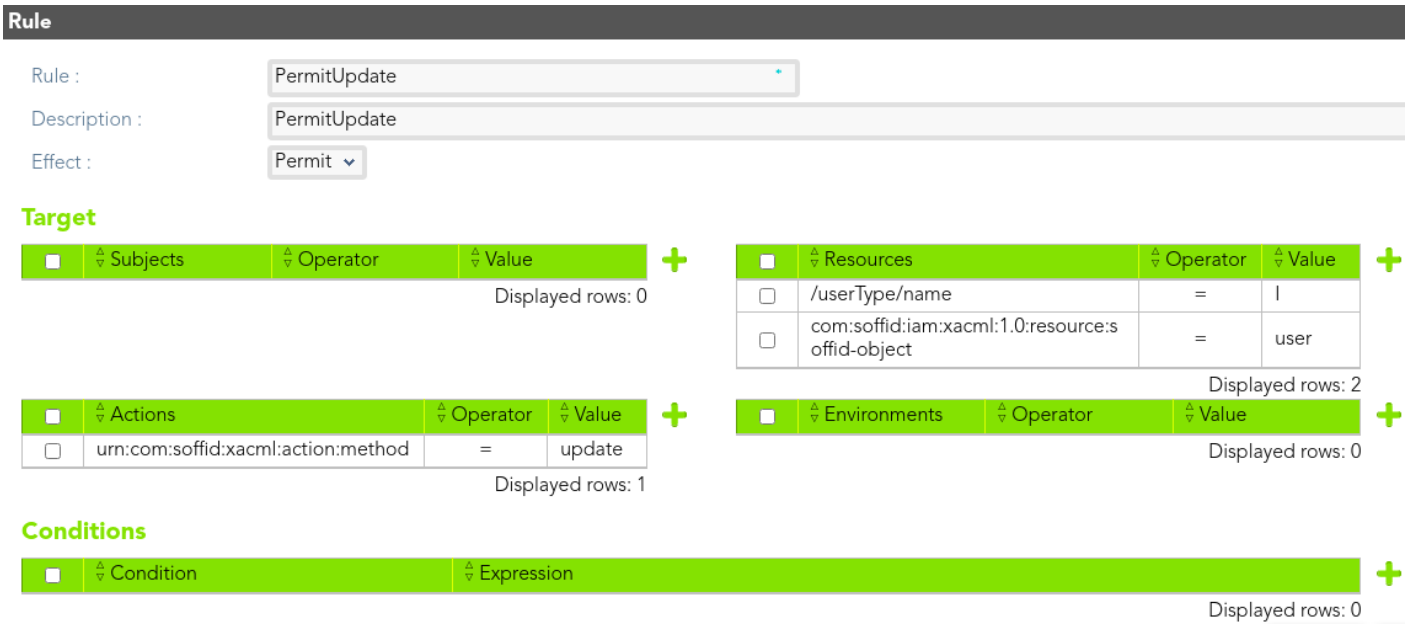
> *“* The *end-users* only be able to update the users with internal user type.

We need to define two rules, one to permit and other to deny access.

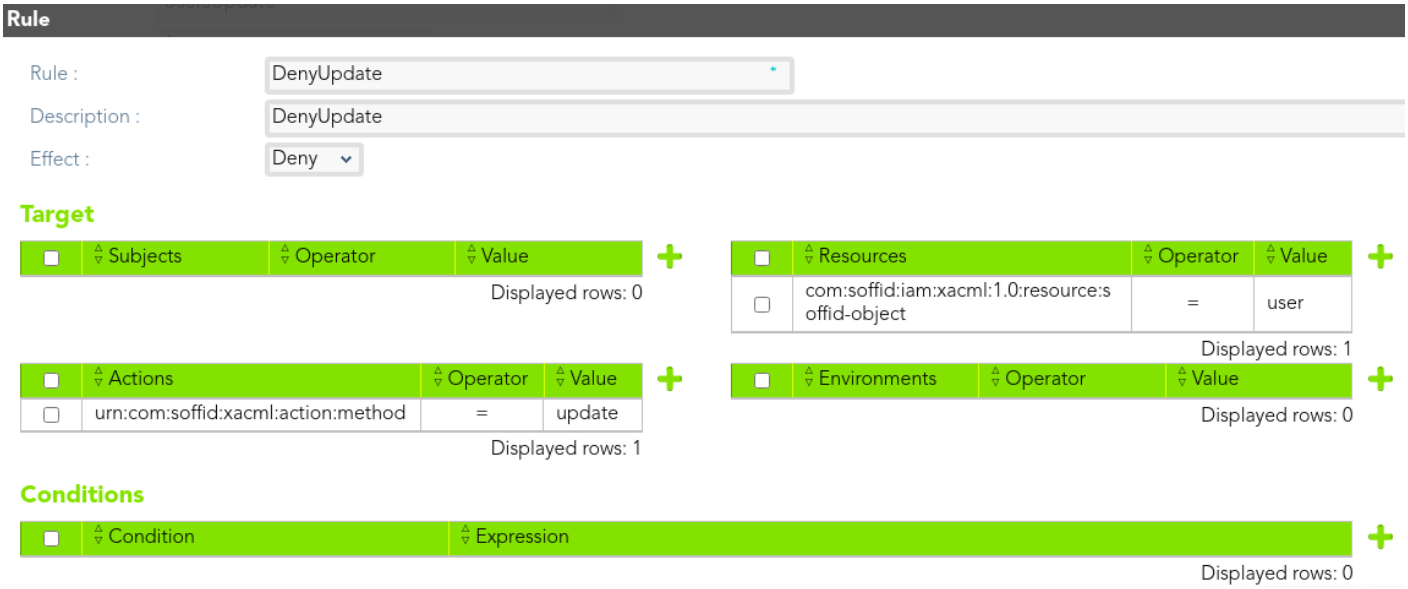


Rules

We define the rule that allow to the *end-users to* update users information who are internal users.



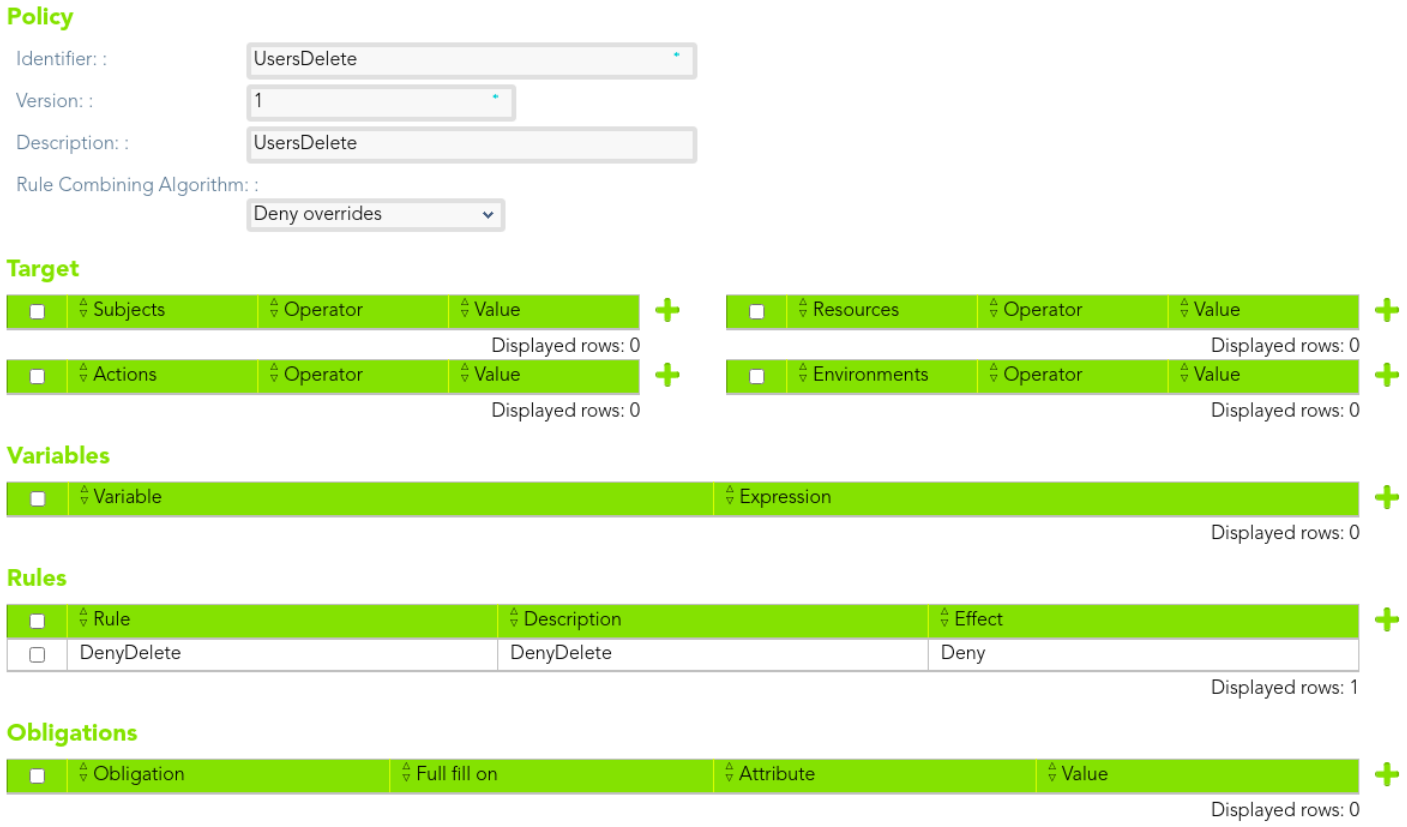
Then, we define the rule that denies access to *end-users* to update users information.



Policy 3

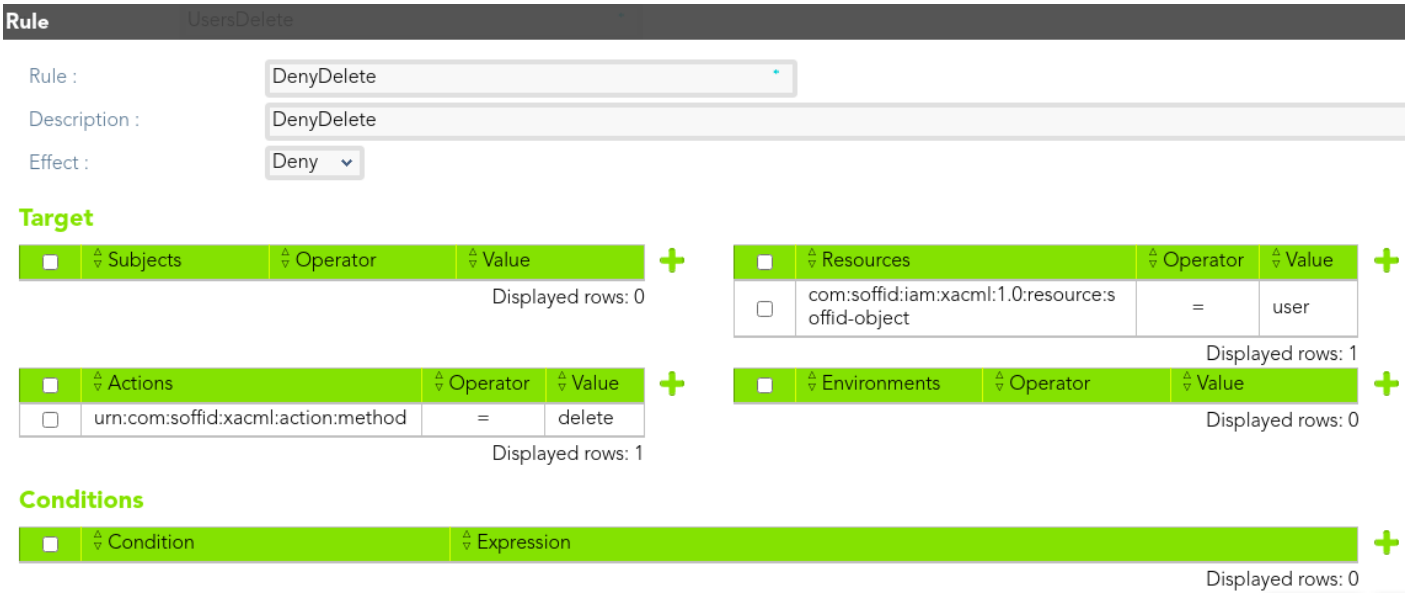
> The *end-users* could not delete any user record.

We need to define only one rule to deny access.



Rules

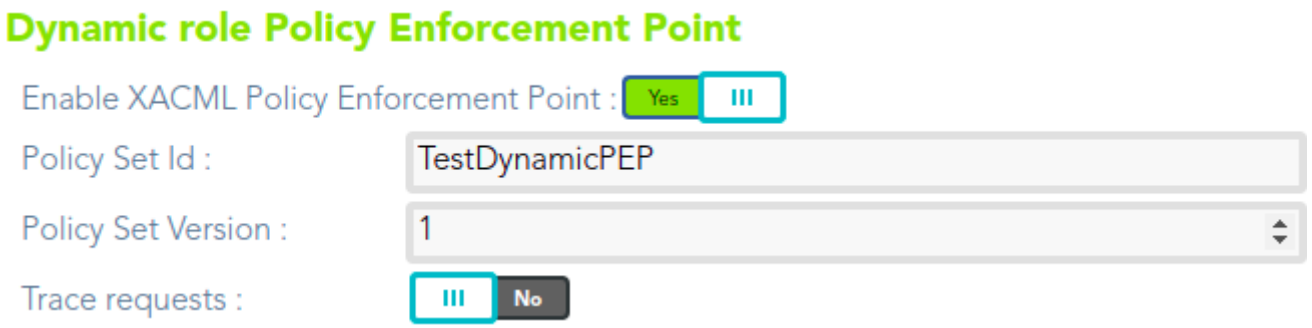
We define the rule that deny to the *end-user* delete any user.



Download XML

> You can download a XML file with the example: policy-TestDynamicPEP.xml

Configure PEP



Revision #20
Created 4 August 2021 13:48:52
Updated 6 August 2021 08:42:09