Wrocław University of Science and Technology

W04N Cryptography and Computer Security

Z21-22 Cryptography and Security K06-93a

Grzegrzółka Mikołaj

ABAC

-homework

1. Task

Write an access control policy according to XACML standard. The situation is that there is a set of articles to be reviewed and a set of reviewers. Some of the reviewers are simultaneously the authors of the papers.

Create a policy that captures the conflicts of interest: the decision should be to accept or deny an access to write a review on a given paper.

Capturing conflicts of interest in this case is uneasy task and you might be creative.

Definitely:

- nobody should be given the right to review own paper,
- if A has a paper co-authored by B, then A must not review any paper of B,
- nobody should be given the right to review the paper of the own boss.

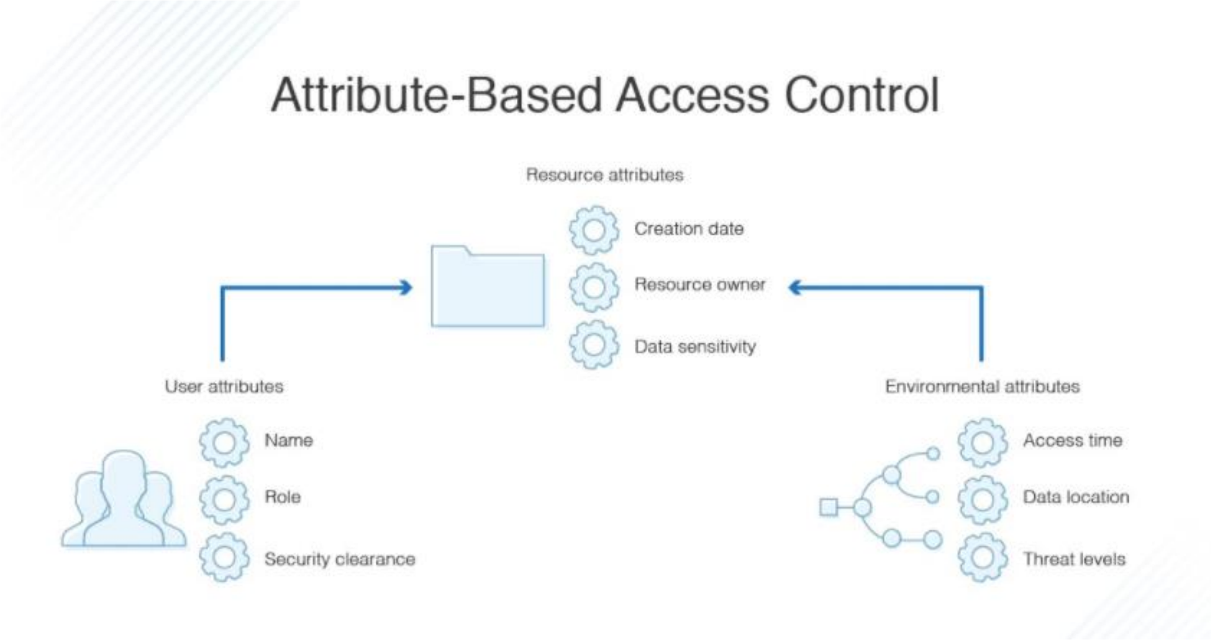
It is recommended to use one of free policy editors available in the internet.

In this case please specify what you have used and look through the code generated.

2. Introduction to XAMCL and ACAB explanation

Attribute-Based Access Control (ABAC) is an access control paradigm whereby access rights are granted to users through the use of policies that combine attributes together. The policies can use any type of attributes (user attributes, resource attributes, object, environment attributes, etc.). This model supports Boolean logic, in which rules contain "IF, THEN" statements about who is making the request, the resource, and the action.

For example: IF the requestor is a manager, THEN allow read/write access to sensitive data. It can be considered completly different approach compared to **Role-Based Access Control (RBAC)**, in which case access to authorized users is based on their role.

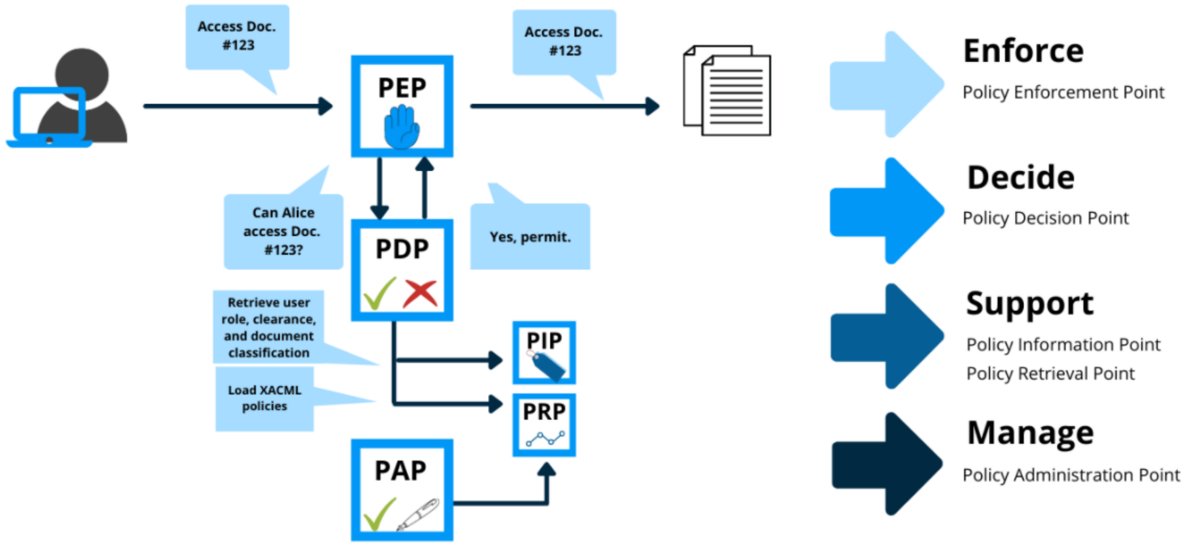


Attribute-based access control draws on a set of characteristics called "attributes." This includes user attributes, environmental attributes, and resource attributes, in which we differ:

- **User attributes** include things like the user's name, role, organization, ID, and security clearance.
- **Environmental attributes** include the time of access, location of the data, and current organizational threat levels.
- **Resource attributes** include things like creation date, resource owner, file name, and data sensitivity.

ABAC is implemented to reduce risks due to unauthorized access, as it can control security and access on a more fine-grained basis. For example, instead of people in the HR role always being able to access employee and payroll information, ABAC can place further limits on their access, such as only allowing it during certain times or for certain branch offices relevant to the employee in question. This can reduce security issues and can also help with auditing processes later.

XACML (eXtensible Access Control Markup Language) is an XML-based language for access control that has been standardized by the Technical Committee of the OASIS consortium. XACML is popular as a fine grain authorization method among the community. XACML describes both an access control policy language, request/response language and reference architecture. The policy language is used to express access control policies (who can do what when). The request/response language expresses queries about whether a particular access should be allowed (requests) and describes answers to those queries(responses). The reference architecture proposes a standard for deployment of necessary software modules within an infrastructure to allow efficient enforcement of policies.



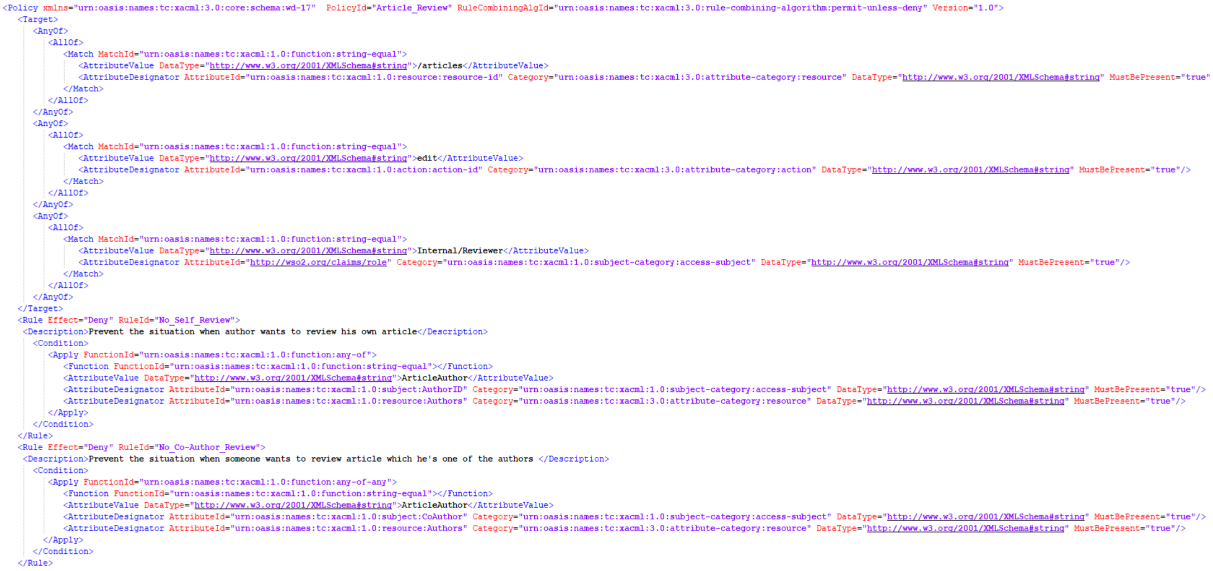
1. A user sends a request which is intercepted by the Policy Enforcement Point (PEP)
2. The **PEP** converts the request into a XACML authorization request

3. The PEP forwards the authorization request to the Policy Decision Point (PDP)
4. The **PDP** evaluates the authorization request against the policies it is configured with. The policies are acquired via the Policy Retrieval Point **(PRP**) and managed by the Policy Administration Point (PAP). If needed it also retrieves attribute values from underlying Policy Information Points **(PIP)**.
5. The PDP reaches a decision (Permit / Deny / NotApplicable / Indeterminate) and returns it to the PEP

3. Practical XACML policy

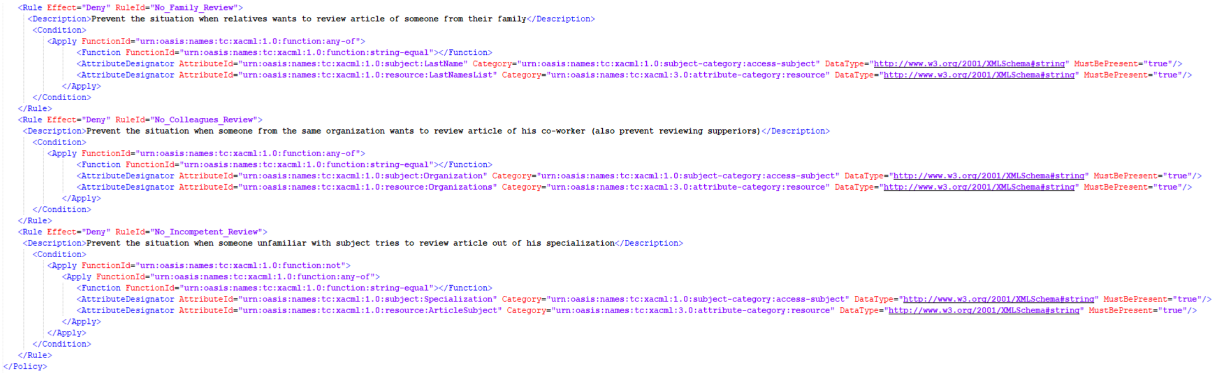
The policy editor I used was WSO2 Identity Server. In the specific scope of the algorithm used, I assumed that we should use something like Permit if there is no Deny.

Rather, the construction is basic, because taking into account each of the assumptions I have defined, the whole policy could be created over the next week 😄. Thus only theorycrafting …



```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"  PolicyId="Article_Review" RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:permit-unless-deny" Version="1.0">
   <Target>
      <AnyOf>
         <AllOf>
            <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">/articles</AttributeValue>
               <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"
            </Match>
         </AllOf>
      </AnyOf>
      <AnyOf>
         <AllOf>
            <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">edit</AttributeValue>
               <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </Match>
         </AllOf>
      </AnyOf>
      <AnyOf>
         <AllOf>
            <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Internal/Reviewer</AttributeValue>
               <AttributeDesignator AttributeId="http://wso2.org/claims/role" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </Match>
         </AllOf>
      </AnyOf>
   </Target>
   <Rule Effect="Deny" RuleId="No_Self_Review">
    <Description>Prevent the situation when author wants to review his own article</Description>
      <Condition>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"></Function>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">ArticleAuthor</AttributeValue>
            <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:AuthorID" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Authors" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
         </Apply>
      </Condition>
   </Rule>
   <Rule Effect="Deny" RuleId="No_Co-Author_Review">
    <Description>Prevent the situation when someone wants to review article which he's one of the authors </Description>
      <Condition>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"></Function>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">ArticleAuthor</AttributeValue>
            <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:CoAuthor" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Authors" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
        </Apply>
      </Condition>
   </Rule>
```

We assume that the author should not, for obvious reasons, be able to evaluate his own work. Additionally, if the Reviewer is a co-author of a given work, we also assume that its evaluation may be biased and should not be able to evaluate. We compare any-of-any authors database with assigned co-authors.



```
  <Rule Effect="Deny" RuleId="No_Family_Review">
   <Description>Prevent the situation when relatives wants to review article of someone from their family</Description>
    <Condition>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
             <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"></Function>
             <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:LastName" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
             <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:LastNamesList" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
          </Apply>
    </Condition>
  </Rule>
  <Rule Effect="Deny" RuleId="No_Colleagues_Review">
   <Description>Prevent the situation when someone from the same organization wants to review article of his co-worker (also prevent reviewing supperiors)</Description>
    <Condition>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
             <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"></Function>
             <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Organization" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
             <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Organizations" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
          </Apply>
    </Condition>
  </Rule>
  <Rule Effect="Deny" RuleId="No_Incompetent_Review">
   <Description>Prevent the situation when someone unfamiliar with subject tries to review article out of his specialization</Description>
    <Condition>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
             <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"></Function>
             <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Specialization" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
             <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:ArticleSubject" Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
          </Apply>
       </Apply>
    </Condition>
  </Rule>
</Policy>
```

We also assume that relatives should not be able to rate their articles for biased reviews. Of course, in this case, I assumed that people with the same surname will be relatives in a given organization, which often happens in the case of universities, but of course there may be a situation where people with the same surname will not be relatives. Place of birth could also be used as an additional checking factor, but the basic policy sheet did not allow me to add additional variables for the classification. Nevertheless, it would still not be an appropriate solution.

We also take into account that colleagues from the same university / organization will also leave more flattering feedback to their colleagues. Therefore, we prefer not being able to evaluate the work of their colleagues, it can be considered an exaggeration, but it is a more sensible solution than specifying the entire chain of command for the organization. Because if we assume that someone is our boss in the organization of a teaching / research institution, he will also be in the same organization as us, the result - we do not evaluate the work of our supervisor.

The last situation that seems obvious in the realities of scientific conferences is the situation of people with mixed specializations. Therefore, it seems obvious that the specialization of the person reviewing a given article should be the same as the subject of a given article.