Deploying Access Control using Extended XACML in Open Web Service Environment

Thirumaran.M
Department of Computer Science and Engineering
Pondicherry Engg College
India

Dhavachelvan.P
Department of Computer Science and Engineering
Pondicherry University
India

Divya.A
Department of Computer Science and Engineering
Pondicherry Engg College
India

ABSTRACT

Now a days web services have a greater role in development of software systems. Generally, web server wants to determine which access can be granted and communicate with requesterfor open web service environment. These systems are designed to be open and web servers does not have determined conditions for communicate to the access requester. Such consideration makes traditional access control not applicable, which depends upon requester identification and authentication. Typically, XACML language is proposed for define the access control policy. It does not support novel features and not suitable in practice. In this paper, we extend the current XACML by providing novel features for controlling access in interoperable and flexible way, and then illustrate how such concepts can be deployed. Also, extend the standard XACML architecture to incorporate the new features for open web service systems

General Terms

Architecture, Policy

Keywords

Access Control, Web Service, Policy Evaluation Engine, XACML, Policy Decision Point, ABAC

1. INTRODUCTION

Web service systems are designed to be open, means where servers do not generally have any prior knowledge of requester, call for a different solution from a traditional access control based on preliminary identification and authentication of requesters. Traditional access control becomes crucial by means of such circumstance. Effectively tackling these issues introduces eXtensible Access Control Markup Language (XACML). XACML is a general-purpose access control policy language.This means that it provides syntax (defined in XML) for specifying and exchanging access control policy over the web. XACML have standard extension points is to define new functions, data types, and policy combination methods, thus exploiting the language's flexibility to adapt it to several different needs. But XACML still suffers from some limitations when it comes to its capabilities in supporting the requirements of open Web-based systems. XACML does not include features. In this paper, we introduce a novel concept that should be supported for practical access control. We have to introduce novel concepts to support features for providing expressive solution. To extend the XACML language by using XACML extension point. We illustrate how such novel concepts can be deployed in XACML. Specific changes needed in the architecture to incorporate the implementation of extension in XACML. In section 3 illustrate how such novel concepts can be deployed, particularly how XACML can be adapted and extended to support these new features and introducing the delegation to distribute the access control with another resource without exposing the whole policy database to them.

2. RELATED WORKS

Traditional XACML does not support the RBAC profile. Rodolfo Ferrini (1) introduced XACML with OWL techniques by extending the XACML language and architecture. It integrates with OWL ontology and XACML policy for supporting the RBAC profile (2). The author [4] proposed service-oriented role-based access control (SRBAC) model. It can be employed a SOAP Proxy to enforce access control for Web Services and security mechanisms are separated from the business logic.Sitaraman[5] established ws-security standards in cloud or distributed web services architecture to address the following security issues 1) exchanging various security tokens , 2) XML encryption 3) XML signature. NI Jun [7] Introduced WS-HPBAC based on SAML, XACML, OWL and policy protocol layer for providing the overall access control mechanism for WS，makes perfect WS authority management ability，and increases WS access control application security. XACML access control policy can be written to detect unauthorized access that permits unauthorized access when the requester takes multiple duties. This technique called as Multiple Duty Related Security Leakage introduced by JeeHyun Hwang [13]. KyuIlproposed a collaborative model based on XACML in pervasive Environments that defines a core schema and corresponding namespace for the expression of authorization policies in XML against objects that are themselves identified in XML [3]. Han Tao and JeeHyun described XACML based access control. Both author [6] proposed a framework which encourages the separation of authorization decisions from enforcement mechanisms.RafaeBhattiproposes an XML-based access control specification language to address this security challenge [8]. On Eric Yuan describes the ABAC which is based on its authorization architecture and policy formulation, and makes a detailed comparison between ABAC with traditional role-based models then show how this new model can be applied to securing web services invocations [10].The EnterprisePrivacy Authorization Language (EPAL) is presented by P. Ashley for writing enterprise privacy policies to govern data handling practices in IT systems [9].R. Bhatti introducedan XML-based RBAC policy specification framework forenforcing access control in dynamic XML based Web

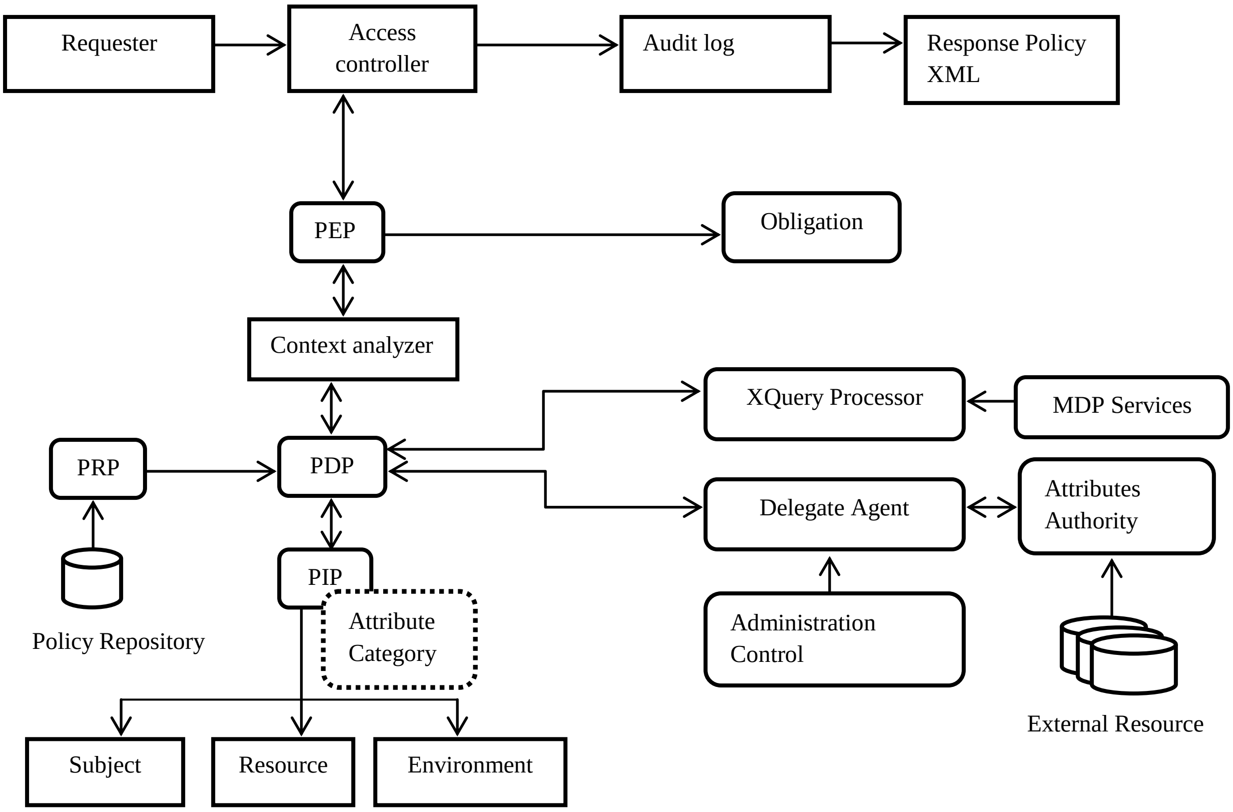

Fig 1: Extended XACML Architecture

services. But the X-RBAC framework only focuses on the XML-based document security, and cannot completely realize access control on service [11].Dong Seong Kim presented the combination of SAML and XACML for access control and authorization to ensure the securityin RFID multi domain environment. And show the detailed of their approach with case studies [14]. The author [15] proposed role based access control model with XML and XACML that uses the requester attributes and roles as a part of access control policy.

3. DEPLOYMENT IN XACML

Current XACML is not suitable for open web based scenarios because it has some limitations for ensuring access control policies. We identify the limitations in current XACML and introduce the novel concepts that the access control in open web services should be support. First existing XACML allows single XACML request with single decision referring to the different components of a policy. In the case of multiple decisions in single request, current XACML does not support these features. Second, XACML does not support abstraction. Third, traditional XACML does not accept the delegation that the global administrator does not delegate rights to local administrator without exposing the whole database to them. In this section we discuss about such limitations and how it can be counteract by adding novel features to XACML for open web services.

3.1 EXACML Architecture

Existing XACML requires some extension in standard XACML architecture to support the above new concepts and functionalities. The standard architecture has different functional components interacting to take an access control decision which are shown in figure 1. With the control of XACML, the requester sends access request to access controller which cans controls to access a certain resource. The access requester sends the request via access controller to PEP (Policy Enforcement Point), which takes control to take decision whether the request should be permit or denied. For policy evaluation the access requester has send all relevant information. Policy can be evaluated in policy evaluation engine. The PEP component sends request to context analyzer that performs format exchange among XACML components called XACML request context and sends it to PDP (Policy Decision Point), which makes decision by checking whether the attributesfulfill the policies associated with the invoked web services by means of PRP (Policy Retrieval Point) module. The PRP retrieves the required policy from policy repository for policy evaluation process. Policy repository stores the policy. The PDP may also get information from PIP (Policy Information Point) acting as source attributes values. During evaluation process, PIP to collects attributes related to the subject, the resource or the environment from the system information storage (e.g., DBMS). The policy can be evaluated by PDP to the context analyzer that translates the context into native format to PEP. The final decision is returned to PEP. If the request is fulfill the obligation request is permitted, otherwise access is denied. Finally audit log will store all information and obligation service is executed. It canbe used to review which access can be granted, and check any attributes are missing during evaluation process.

```
<Request>
<Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
<AttributeValue>
</Attribute>
<Attribute AttributeId="emoney" DataType="http://www.w3.org/2001/XMLSchema#emoney"
Issuer="admin@users.example.com">
<AttributeValue>book1|book2|book3</AttributeValue>
</Attribute>
</Subject>
<Resource>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
<AttributeValue>books</AttributeValue>
</Attribute>
</Resource>
<Action>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<AttributeValue>multipleresource</AttributeValue>
</Attribute>
</Action>
</Request>
```

Fig 2: An example XACML Policy for Multiple Decision Profile

```
<Request>
<Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
<Attribute AttributeId="emoney"
DataType="http://www.w3.org/2001/XMLSchema#emoney"Issuer="admin@users.example.com">
<AttributeValue>debit1</AttributeValue>
</Attribute>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
<AttributeValue>user1@users.example.com</AttributeValue
</Subject>
<Resource>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
<AttributeValue>http://server.example.com/</AttributeValue>
</Attribute>
</Resource>
<Action>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<AttributeValue>abstract</AttributeValue>
</Attribute>
</Action>
</Request>
```

Fig 3: An Example XACML Policy for Abstraction

```
<Request>
<Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
<AttributeValue>user1@users.example.com</AttributeValue>
</Attribute>
<Attribute AttributeId="emoney" DataType="http://www.w3.org/2001/XMLSchema#emoney"
Issuer="admin@users.example.com">
<AttributeValue>patient1:doctor3</AttributeValue>
</Attribute>
</Subject>
<Resource>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
<AttributeValue>http://server.example.com/</AttributeValue>
</Attribute>
</Resource>
<Action>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<AttributeValue>delegate</AttributeValue>
</Attribute>
</Action>
</Request>
```

Fig 4: An Example XACML Policy for Delegation

3.2 Extending XACML Architecture to Support Features

In standard XACML we added components to incorporate the extension. We need to extend PDP component by adding XQuery processor and delegate agent that should support the novel features during policy evaluation

3.2.1 Multiple Decision Profile

Multiple Decision profile is the important feature is to be able to define profile for specific uses. (e.g., in export control) Access requester can send the XACML request for multiple accesses. It means single XACML request represent multiple access control request. In Existing XACML does not support these features. By improving the XQuery processor with XPath function to manage such conditions. For this purpose, we use <MultipleRequest> element of XACML. It allows more fine grained selection of different combinations of multiple requests. For instance a requester wants to read many books which mean single XACML request contains multiple access control request for reading multiple books. Otherwise requester can send the access request each time for every single book. Also, we use <Attribute designator> element to avoid the reconstruct XML individual requests in multiple decision process. Attribute category also introduced in extended XACML for define the requester attributes in groups using <Attribute Category> instead of define the subject, resource and environment separately. It can be embedded in XACML language. Figure 2 shows an example XACML authorization. A requester wants to send a XACML request to read a multiple books (book1, book2 and book3) for multiple access it means single XACML request represent multiple access control request. It returns the single XACML response which contains multiple access response to permit to read the multiple books.

3.2.2 Abstraction

XQuery Processor also supports the abstraction. Traditional XACML does not support these feature. XQuery functions can defined the abstractions based on abstract concepts. It can be used to define the policy in easy and compact way. For instance id_document is abstraction form that is input and it represents a set of components {identity_card, driver_license,passport} as output. An example XACML request for abstraction depicts in figure 3. Abstraction head id_document can be defined as an abstraction for any element in the set of credential {debit card, credit card, passport} in abstraction tail. Here id_document can be defined in debit as output. While the PDP needs to conclude on abstractions, extract the abstraction form stored in XACML policies or profiles and evaluate it, while the PDP needs to conclude on abstractions.

3.2.3 Delegation

Delegation plays an important role in XACML. In this mechanism can be used to support the decentralized administration of access policies. It enables global administrator to delegate rights to local administrator. The current XACML does not support any form of delegation. During policy evaluation, delegate agent only controls such process for provide the controlled piece of policy administration to local administration. The Attribute Authority is the component which is trusted by the data subjects to manage their personal data as they specified. It provides the minimum information to make policy decisions. When an authorization decision comes to attribute authority from PDP, attributes authority extract the personal data concerned by response and encrypt it. Then, encrypted data forward to PEP, the PEP send to the access requester. Attribute authority takes attributes from an external resource (e.g., LDAP).Policy administration controlled administration of XACML policy.

Administration policy and access control work together to support delegation.Global administrator delegates its rights to local administrator. Let's us consider a situation doctor 3 wants to view the doctor 2 patients record incase doctor 2 may not present. So doctor 2 delegates his rights to doctor 3 for access the patient1 record. Doctor 2 provides the controlled piece of policy to doctor 3 to view the patient1 records which shownin figure 4.

Table 1.Shows the features of XACML with its elements

Features	Elements
Multiple Decision Profile	<Multiple Request>
Abstraction	XQuery Function
Delegation	XQuery Function

Table 1 summarizes the features of XACML and its elements. By improve the XACML language with these element to incorporate our extension for provide access control.

4. CONCLUSION

Traditional XACML does not suitable in web service. Because it does not have some features to provide expressive access control. So Extended XACML has been introduced. We provide the novel features and functionalities to support the multiple decision profile, abstractions, delegation. We have embedded the XQuery function in XACML language to incorporate the multiple decision and abstraction. Then, delegate agent can manage the delegation without exposing whole database by using attributes authority and administration control. Extended XACML architecture can incorporate the new features for fully support the open web-based scenarios.

5. REFERENCES

[1] C.A. Ardagna, S. De Capitani di Vimercati, S. Paraboschi, E. Pedrini, P. Samarati and M. Verdicchio, "Expressive and Deployable Access Control in Open Web Service Applications", IEEE Transactions On Services Computing, 2010.

[2] Rodolfo Ferrini and Elisa Bertino, "Supporting RBAC with XACML+OWL", ACM,June, pp.145-154, 2009.

[3] Kyu Il Kim, Hyuk Jin Ko, Won Gil Choi, EunJu Lee, and Ung Mo Kim, "A Collaborative Access Control based on XACML in Pervasive Environments" , in proceedings of the International Conference on Convergence and Hybrid Information Technology, IEEE Computer Society, 2008.

[4] XuFeng, Lin Guoyuan, Huang Hao, and Xie Li, "Role-based Access Control System for Web Services", in Proceedings of the Fourth International Conference on Computer and Information Technology, IEEE, 2004.

[5] SitaramanLakshminarayanan, "Interoperable security standards for web services", IEEE Computer Society, October, 2010.

[6] Han Tao, "A XACML-based Access Control Model for Web Service", IEEE, 2005

[7] NI Jun, CHEN Xiao-su，WU Jin-hu and LIU Hui-yu," Research on Hierarchical Policy-based Access Control Model for Web Services", IEEE, 2009

[8] RafaeBhatti Elisa BertinoArifGhafoo, and James B.D.Joshi, "XML-Based Specification for Web Services Document Security", IEEE Computer Society, April, 2004.

[9] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, Enterprise Privacy Authorization Language (EPAL), Research Report RZ 3485, IBM Research, March,2003.

[10] Eric Yuan and Jin Tong, "Attributed Based Access Control (ABAC) for Web Services", in Proceedings of the IEEE International Conference on Web Services, IEEE, 2005

[11] R. Bhatti, J. B. D. Joshi, E. Bertino, A. Ghafoor, "Access Control in Dynamic XML-based Web-Services with XRBAC", In proceedings of The First International Conference on Web Services, Las Vegas, June 23-26, 2003

[12] Shih-Chien Chou and Chun-Hao Huang, "An extended XACML model to ensure secure information access for web services" , The Journal of system software, pp.77-84, 2010

[13] JeeHyun Hwang, Tao Xie and Vincent C. Hu, "Detection of Multiple-Duty-Related Security Leakage in Access Control Policies", In proceeding of the IEEE conference on Secure Software Integration and Reliability Improvement, 2009

[14] Dong Seong Kim, Taek-Hyun Shin and Jong Sou Park, "Access Control and Authorization for Security of RFID Multi-Domain Using SAML and XACML",IEEE, 2006

[15] Yun-qing Fu and Chun-xiao Ye, "Using XACML to define access control policy in information system" , in proceeding of the IEEE international conference on wireless mobile and sensor network, 2007.