Attribute-Based Access Control Model for Web Services in Multi-Domain Environment

Lanjing Wang
Department of computer
North China Electric Power University
Baoding, Hebei, China
wanglanjing0218@gmail.com

Baoyi Wang
Department of computer
North China Electric Power University
Baoding, Hebei, China
wangbaoyi@126.com

Abstract—**As Internet enhancing its distributed computing characters, Web Services have gained great development and multi-domain environment application becomes more and more popular. But across different domains, the traditional RBAC model is facing some troublesome questions, such as user-role assignment and mapping difficulty. Web Services access control model based on attributes in multi-domain environment is presented. Attributes, extending from roles, can make up those shortcomings of RBAC, and also make access control more flexible, dynamic and fine-grained. Meta-attribute and meta-policy are presented to describe the attributes and policies in local domains.**

Keywords- Access Control; Role ; RBAC ; multi-domain ; Attributes

I. INTRODUCTION

With the development of the Internet enhancing its distributed computing characters, XML(eXtensive Markup Language) based Web Services(WS for short) have gained great development and extensive applications for its characters of loose coupling, language neutral, platform independent and openness. Distributed systems can be viewed composed of several administrative domains, while strengthens the information sharing and collaboration across domains, but how to insure Web Services security in each domain must be focus on, especially the access control issues in [1]. Most of the existing access control models are designed for single local domain and haven't taken into the considerations of communication with the users and applications in foreign domains, for the definition and design of policies and roles in different domains may be quite different from each other. And this situation has greatly hindered the collaborations between different domains, and becomes a critical issue of making access control to the authorized users (no matter in local or foreign domains).In this paper, an attribute based access control model for Web Services in multi-domain environment is designed, and meta-policy and meta-attribute are presented to describe the policy and attribute of local domain. The design can be adapt well to the distributed environment, satisfy the dynamic and heterogeneity characters of Web Services and make the system flexible and scalable for the use of XACML access control model.

Sponsored by the Youth Teacher's Research Foundation of North China Electric Power University (No. 200911018)

II. RELATED WORKS

The multi-domain environment, which is appearing with the development of distributed applications such as Web Services, is quite complex compared with the traditional single domain. For the application and security systems are both based on their own domain, not considering the information sharing issues, especially the access control for the users from the foreign domains. In such situation, how to make access control to such users becomes an important question.

Access control across multi-domain in essence is to authorize the users of foreign domains in local domain. In role based access control (RBAC) systems, it refers to mapping the roles of foreign domains to that of local domains in [2]. The role mappings between domains can be defined directly, or by the trusted third-party authorized by the delegation mechanism. Reference [3] discusses the dynamic cross-domain role mapping, and introduces the concept of secure virtual domain in multi-domain environment with its construction method and dynamic modifying rules, but it is not easy to quantitatively define and judge the criteria of the trust degree.

Considering the distributed and heterogeneous characters of Web Services environment, several requirements in [4] for the Web Services access control have been addressed, including assertion-based, mechanism-independent, policy management, access control integration and standard-based implementation. Additionally, PERMIS has drawn the conclusion that XML is the most suitable language to define the policies after the researches to the Ponder and Keynote and some other policy languages in [5]. In our model, we use XACML in [6] to define the access control policies, and its generic architecture to make access control decision making, which can be seamlessly integrated with the MD_ABAC model which we present next.

RBAC model makes user and permission logically separated from each other by "roles" in [7]. It doesn't need to directly authorize the so many users of the system, but just make management to the user-role and role-permission relationships. Roles own stronger stabilization and less quantity than users, so it can greatly reduce the complexity of the authorization and the administrator's workload. RBAC can also support role hierarchies and constrains models.

978-1-4244-5326-9/10/$26.00 ©2010 IEEE

But in RBAC models, the most roles are just referred to the positions in the organizations, which will bring two shortcomings: one is the definition of roles are so simple that cannot provide enough semantics information to satisfy the real applications. For in the real world, besides the roles, the users may own many other attributes, just using the role information has difficulty to define the fine-grained access control policies; the other problem is that it becomes an onerous task to do the user-role assignment for the complex roles (attribute based roles), especially in large systems whose number of users is quite large. In [8], a mechanism of dynamically assigning users to roles based on a set of rules pre-defined by the enterprise is presented. Although this may greatly reduce the administrators' workload, but it may also lower the efficiency and bring the interoperability issues across different domains, for the role mapping can be complex and make difficulty for the integration of and access to the Web Services.

Based on the above considerations and attribute based access control ABAC in [9], ABAC for multi-domain(MD_ABAC) is presented. ABAC in essence is an extension of RBAC, for role itself is one of attributes, but won't include the attribute-based roles. Mata-attribute and meta-policy are presented to respectively describe the basic information of the attributes and policies in local domain, and the latter can also be used to describe the attribute mapping of service-level.

III. MD_ABAC MODEL

The MD_ABAC model can be expressed as $MD\_ABAC=\{S,R,E,A,P,AA,PA,MA,MP\}$,and respectively referred to the subject, resource, environment, attribute, permission, attribute assignment, permission assignment, meta-attribute and meta-policy correspondingly. The model is defined as figure 1, and the detailed description for the model is as follows:

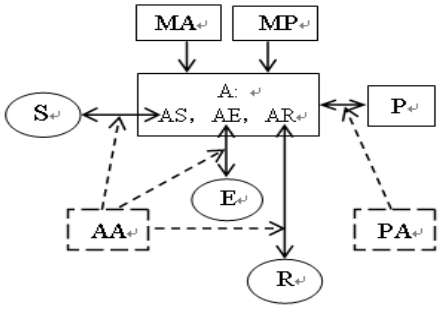

Figure 1. Attribute based access control model for multi-domain environment

(1) S is the entity($et \in S \cup R \cup E$) that makes access to the resource, and can be a user, an application or a process; R is the entity that will be accessed by the subject; E refers to the access context which will provide the access-related operational, technical or situational information .

(2)Attribute $A=\{AS, AR, AE\}$ is predefined by administrator of the application system and stored in the Attribute field of the attribute certificate (AC). Attribute is composed of attribute type AT and attribute value AV, and each attribute type $at_i \in AT$ will correspond to a set of values $AV_i \in AV$. AV_i can be expressed as $AV_i=\{av_{i1},...av_{ij}...,av_{in}\}$, in which n is the total value number corresponding to at_i and av_{ij} refers to the j_{th} attribute value. For some entity et, if $\exists at_i \in AT$ and $\exists av_{ij} \in AV_i$, then $at_i(et)=av_{ij}$, and this process is called attribute assignment AA.

AS (Attribute of Subject) : is used to describe the security-related characters of the subject which is referred to the entity (e.g., a user, an application, or a process) that requests for accessing. Role can also be viewed as a special AS. $AS(s) \subseteq ASV_1 \times ASV_2 \times \cdots \times ASV_k \times \cdots \times ASV_K$, where K is the predefined total number of AS.

AR (Attribute of Resource): used to describe the characters of the resource. $AR \subseteq ARV_1 \times ARV_2 \times \cdots \times ARV_m \times \cdots \times ARV_M$, where ARV_m is one attribute value set corresponding to art_m and M is the total number of AR.

AE (Attribute of Environment): It is used to provide some environment and context information while the access is occurring, for example the current date and time, the security level and other information which is not associated with the subject or the resource, but relevant in making access control decisions. The design of AE makes it easier to realize the dynamic and fine-grained access control. $AE \subseteq AEV_1 \times AEV_2 \times \cdots \times AEV_n \times \cdots \times AEV_N$, where AEV_n is an attribute value set and N is the predefined total number of AE.

(3) $AA \subseteq (S \bigcup R \bigcup E) \times AS$, a many-to-many S (subject) to AS (Attribute of Subject) assignment relation;

$PA \subseteq A \times P$, a many-to-many A (attribute) to P (permission) assignment relation;

(3) MA (Meta_Attribute) and MP (Meta_Policy) are using XML to respectively describe the attributes and policies. They are presented to support the openness, flexible and distributed characters of the Web Service. They will make all the information, including the attribute and policy, can be retrieved by the metadata, and this provides a highly autonomy environment for the new security container. The process for the domain administrators managing their resources and making access control to resources will both need meta-attribute and meta-policy to provide information for corresponding attribute and policy. When making access control decisions, context handler will query the attribute and policy information by MA and MP. The definition for MA and MP are shown in figure 2.

Meta-attribute are constructed by both the AA and administrator of the native domain, and used to describe the basic character of the attributes to provide the basic information for the attribute mapping between domains. There are different definitions of the attributes in different domains, so the attributes of the foreign domain users need to be mapped into the local attribute. Only in this way, the policy in local

domain can be used to make the decisions to judge whether the subjects own the corresponding permissions.

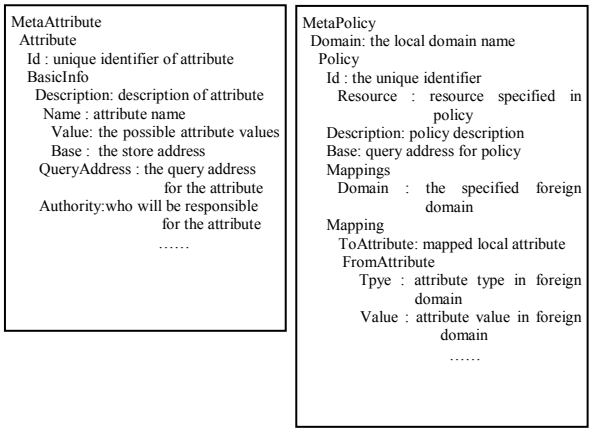

Figure 2. The element definitions for meta-attribute and meta-policy

Besides realizing the policy management in [10], meta-policy can also be used to specify where to query the needed policy (such as the address of LDAP server). This takes into considerations that the policies can be stored in different databases, and is quite suitable for the owners of WS to autonomic define the access policy. It also makes easier to integrate with the original system seamlessly. Meta-policy can also realize the attribute mapping of the service level. It is constructed by the constructer of the corresponding policy (Issuer in the policy AC), and <Mappings> are created by the system administrator after getting the access policy and <BasicInfo> of the meta-attribute in foreign domains.

IV. MD_ABAC BASED DECISION ALGORITH

The MD_ABAC based access control decision algorithm is described using pseudocode as follows:

Algorithm name: Make_Access_Control_Decisions

Input : the access information provided by the subject (subject $s \in S$,resource $r \in R$)

Output : $result \in \{permit, deny, indeterminate\}$

Method : {

step1. making ass , ars , aes , $fass$, $fass'$, as 和 a as Φ

Setp2. /*query the meta-policy MP according to the r */

$mp = MP\_Quering(MP, r)$;

Step3. /*get the policy */

$p = Policy\_Quering(mp)$;

Step4. /*get the needed foreign subject attribute set $fass'$ and mapped into the local attribute */

if (s is from foreign domain 'dom')

{ $fass = FA\_Quering(mp, dom)$;

for each $fas \in fass$

{ $fma = get\_FMA(fas, dom)$; $fasv = get\_FASV(fas, fma)$;

$fass' = fass' \cup \{(fas, fasv)\}$;

}

$ass = SA\_Mapping(fass', mp)$;

}

else $ass \leftarrow p.AS$;

step5. /*query the needed attribute and environment attributes */

if ($p.AE \neq \Phi$) $aes \leftarrow p.AE$;

if ($p.AR \neq \Phi$) $ars \leftarrow p.AR$;

step6. /* query the needed attribute values */

$ats = ass \cup aes \cup ars$;

for each $at \in ats$

{ $ma = get\_MA(at)$; $av = get\_AV(at, ma)$; $a = a \cup \{(at, av)\}$;}

Step7. /*making access decisions according p and a'*/

$result = Decision\_Making(p, a)$;

return $result$;

}

The main idea of MD_ABAC based access control decision algorithm is based on the distributed character of the resource of subject and the storage of attributes and policies. The Sequence map of MD_ABAC based access control is shown in figure 3.

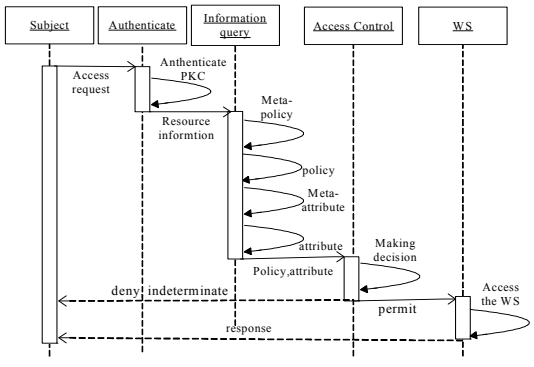

Figure 3. Sequence map of MD_ABAC based decision algorithm

The function of $MP\_Quering$ is used to query meta_policy of the needed resource r from the MA file in local domain; $Policy\_Quering$ and $FA\_Quering$ can respectively return the policy p and the foreign domain attribute set of fass according to the Base and Mapping element in mp; then the meta attribute ma and the foreign attribute set fasv can be got from the get_FMA and get_FASV, and $SA\_Mapping$ can finish the mapping from the foreign domain attribute to the local attributes. It is the same for AE and AR to realize the query using the address information of meta_attribute, that is to get meta_attribute ma using $get\_MA(at)$ and attribute value av using $get\_AV(at,ma)$. Then the decision result can be made by $Decision\_Making$ using the attribute and policy information p.

The advantages of meta-attribute and meta-policy introduced MD_ABAC model:

(1)ABAC model is based on RBAC but extending the "role", so it possesses the advantages of RBAC, such as making the user logically separate from the permission, lowering the complexity and workload of the authorization work. It can also make up several shortcomings of the attribute-based roles, such as user-role assignment and mappings in multi-domain. Assuming there are m attribute in one domain, and for each attribute there are ki value, so there are $\prod_{i=1}^{m} k_i$ roles in RBAC and $\sum_{i=1}^{m} k_i$ attributes in MD_ABAC model. Moreover, the introducing of attributes are consistent with the cognizing of the real world, intuitive to make analysis and modeling the access control policy, and more flexible and powerful to describe the fine-grained access control policy, which is especially suitable for the dynamic characters of Web Services.

(2) Meta-attribute and meta-policy are presented to describe the attributes and policies in local domain. The address information of the attribute and policy can give enough support for the autonomy management of Web Services; the service-level attribute mapping in meta-policy makes stable foundation for the multi-domain access control.

V. CONCLUSION

With the rapid development of Web Services, its security questions should be paid enough attentions, especially making access control to the WS requesters in a relative complex multi-domain environment. So an attribute-based access control model for Web Services in multi-domain environment (MD-ABAC) is presented. It possesses the advantages of RBAC but makes up some shortcomings of RBAC. The meta-attribute and meta-policy are presented to make attribute mapping in the distributed multi-domain environment. In conclusion, the MD-ABAC for WS model can adapt well to the distributed environment, satisfy the dynamic and heterogeneity environment of Web Services and make the system flexible and scalable for the use of XACML access control model. But there are still some potential research needs to be done, such as the perfect definition of the meta-attribute and meta-policy, and the overall end-to-end security architecture.

REFERENCES

[1] Yue Kun, Wang Xiao-Ling, Zhou Ao-Ying: Underlying Techniques for Web Services: A Survey. Journal of Software. vol.15, 2004, pp. 428-442

[2] Zhu Xian，Hong Fan，Duan Su-juan: The Trust Propagation Policy for Secure Interoperability in Multi—Domain Environments. Computer and Engineering & Science. Vol. 27, 2005, pp. 15-17, 37

[3] Hong Fan, Li Cheng-bing: Role-Based Access Control for Multi-Domain Coalition Environments. Computer and Engineering & Science. Vol.27，No.6,(2005)1-3，34

[4] M. Coetzee, J.H.P. Eloff.: Towards Web Service access control .Computers & Security, 2004, pp. 23, 559－570

[5] David W Chadwick, Alexander Otenko, Edward Ball. Implementing Role Based Access Controls Using X.509 Attribute Certificates – the PERMIS Privilege Management Infrastructure.IS Institute, University of Salford, M5 4WT, England, 2003

[6] OASIS.Extensible Access Control Markup Language (XACML) Version 1.0. OASIS Standard, http://www.oasis-open.org/xacml/,2002-02-18

[7] Ravi Sandhu ， Edward J.Coyne, Role-Based Access Control Models.IEEE,Computer, 1996, pp. 12,38-47

[8] Mohammad A. Al-Kahtani, Ravi Sandhu. A Model for Attribute-Based User-Role Assignment. 18th Annual Computer Security Applications Conference , 2002

[9] Eric Yuan，Jin Tong，Attribute Based Access Control (ABAC) for Web Services，Proceedings of the IEEE International Conference on Web Services, 2005

[10] Chadwick, D.W., Otenko, A. "RBAC Policies in XML for X.509 Based Privilege Management" , SEC 2002, Egypt, 2002