RESEARCH ARTICLE OPEN ACCESS

Securing the Cloud Resources Using Attribute Based Access Control

Pradnya Agale
Department of Computers Science
Mumbai University
Mumbai -India

ABSTRACT

Attribute based access control (ABAC) model can define permission based on just about any security relevant characteristics, known as attributes. Attribute based access control is an access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attribute of object, environment conditions and a set of policies that are specified in terms of those attributes and conditions .

Keywords:-Cloud Computing; attribute based access control(ABAC);access control .

I. INTRODUCTION

Attribute based access control defines an access control paradigm whereby access rights are granted to users through the use of policies which combine attributes together. the policies can use any type of attributes. (user attributes, resource attributes, environment attributes etc).attribute values may be set-valued or atomic valued. set valued attributes contain more than one atomic values. atomic valued attributes contain only one atomic value. ABAC is highly flexible method for providing access control based on the evaluation of attributes. ABAC avoids the complex structure of multi-attribute and solves the problem that relevant dynamic authorization and permission changes.

II. ATTRIBUTE BASED ACCESSS CONTROL

The Attribute Based Access Control (ABAC) model consists of two aspects: the policy model which defines the ABAC policies, and the architecture model which applies the policies to web services access control.[2]

The ABAC model can define permissions based on just about any security relevant characteristics, known as attributes. For access control purposes, there are three types of attributes.

Subject Attributes-A subject is an entity (e.g., a user, application, or process) that takes action on a resource. Each subject has associated attributes which define the identity and characteristics of the subject. Such attributes may include the

Subject’s identifier, name, organization, job title, and so on[2]

A subject’s role, naturally, can also be viewed as an attribute.

Resource Attributes-A resource is an entity (e.g., a web service, data structure, or system component) that is acted upon by a subject. As with subjects, resources have attributes that can be leveraged to make access control decisions. A Microsoft Word document, for example, may have attributes such as title, subject, date, and author. Resource attributes can often be extracted from the metadata of the resource. In particular, a variety of web service metadata attributes may be relevant for access control purposes, such as ownership, service taxonomy, or even Quality of Service (QoS) attributes.[2]

Environment Attributes-These attributes describe the operational, technical, and even situational environment or context in which the information access occurs. For example, attributes such as current date and time, the current virus or hacker activities, and the networks security level (e.g., Internet vs. Intranet), are not associated with a particular subject nor a resource, but may nonetheless be relevant in applying an access control policy.[2]

By treating role and identity as characteristics of a principal, ABAC fully encompasses the functionality of both IBAC and RBAC approaches. Therefore, we believe that the ABAC is the natural convergence of existing access control models and surpasses their functionality. Policy representation is semantically richer and more expressive, and can be more fine-grained within ABAC because it can be based on any combination of subject, resource, and environment attributes.[2]

SYSTEM ARCHITECTURE

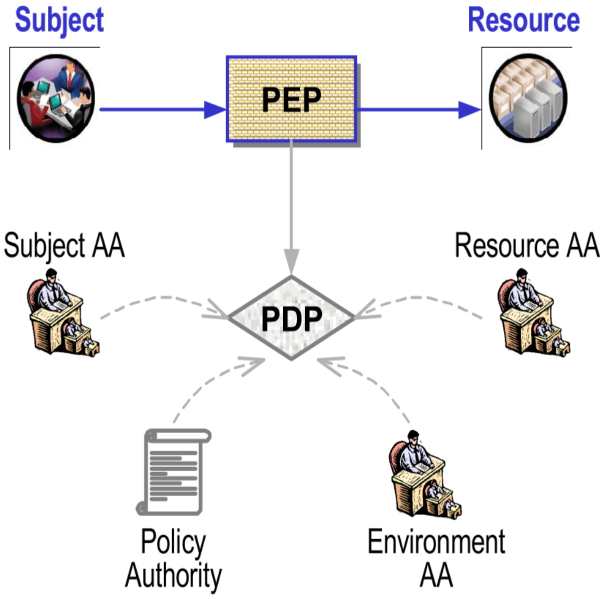

Figure 1: ABAC Authorization Architecture

III. FUNCTIONAL MODULES

1) The message processing module

This module have two parts: the client module and server-side message processing module, client module encrypts login information and decrypts the token, the server-side module decrypts login information and encrypts the token.

2) The authentication and token management module

The authentication analyzing login request, to obtain the user's identity information and authenticate the user based on the user information database. Token management module provides identity token to user, and verifies the user's identity token for authentication, it also extracts the relevant user information from user information database according to the token.[2]

3) Access control module

Attribute-based access control module is the core of the system which contains the Policy Enforcement Point, Policy Decision Point, Policy Authority and Attribute Authorities.[2]

- The *Attribute Authorities (AA)* is responsible for creating and managing the attributes for subjects, resources, and the environment, respectively. As a logical entity, an AA may or may not store the attributes by itself (e.g., a Subject AA may choose to retrieve attributes from the organization LDAP directory), but it is responsible for binding attributes to an entity of interest, and plays an important role in the provisioning and discovery of attributes.[2]

- The *Policy Enforcement Point (PEP)* is responsible for requesting authorization decisions and enforcing them. In essence, it is the point of presence for access control and must be able to intercept service requests between physically distributed throughout the network. The most important security engineering consideration for the implementation of PEP is that the system must be designed such that the PEP cannot be bypassed in order to invoke a protected resource[2].

- The *Policy Decision Point (PDP)* is responsible for evaluating the applicable policies and making the authorization decision (permits or deny). The PDP is in essence a policy execution engine. When a policy references a subject, resource, or an environment attribute that is not present in the request, it contacts the appropriate AA to retrieve the attribute value(s).

- The *Policy Authority (PA)* creates and manages access control policies. The policies may consist of decision rules, conditions, and other constraints for accessing the resources.[2]

4) The service management module

The module consists of two modules: the service registration and service queries. All services within the security architecture must be unified registration in order to find the service conveniently. Service information management module extracts resources services to respond to requests of users.[2]

ACCESS CONTROL PROCESS

Before access to the resource, you must first log in to the system, the system uses a single sign-on mechanism, through an identity authentication, and users can seamless access to the service from different security domain. When the users need to access another security domain services, it will occur a cross security domain service access[2].Typical cross-security domain services call process:

1) After log in, the user proposes an access request.

2) Authentication module authenticates the user, sent the user's information and access requests to the access control module.

3) According to the user information and the resource URL of requests access, respectively, the access control module obtained attributes which is available for decision-making from the attribute authority and resource library.

4) The Policy Decision Point permits or denies this access based on the algorithm and policy repository.

IV. CONCLUSION

The ABAC model brings out many advantages over traditional identity or role based models:

- It is intuitive to model and manage real-world access control policies[1]
- It is more flexible and more powerful to describe complex, fine-grained access control semantics, which is especially suitable for the dynamic, ad-hoc environments for Web services[1]
- Under ABAC, the management of security information is spread over a number of Attribute and Policy Authorities, which can be distributed over the network, even across organizational boundaries – also something that is suited for the service-oriented architectures of future enterprises.
- Similarly, this "divide and conquer" approach significantly reduces overall system complexity, allowing different system actors (User Directory,Service Registry, Policy Server, etc.) to focus on their respective administrative needs[1]

The ABAC model, though very powerful, only focuses on authorization of requests from information consumers to providers. An end-to-end security architecture,
however, is concerned not just with the access control model. Rather, it covers the whole set of components, interfaces, and
data that allows authorization decisions to be made and enforced. To utilize the ABAC model to its full potential, many other aspects of the entire attribute management "life
cycle" needs to be considered, such as attribute provisioning, cryptographic binding of attributes to entities, attribute discovery, and the feedback loop on attribute usage. These are all potential research and engineering topics to be explored[1]

REFERENCES

[1] Eric Yuan, Jin Tong Attribute Based Access Control(ABAC)for Web Services[C]//IEEE International Conference on Web Services(ICWS'05).2005

[2] Ni Dan, Shi Hua-ji Attribute Based Access Control(ABAC) -based cross-domain access control in service oriented architecture (SOA)[c]// IEEE International Conference on Web Services(ICWS'05).2012

[3] J. F. Barkley, K. Beznosov, and J. Uppal, "Supporting Relationships in Access Control Using Role Based Access Control", *ACM Workshop on Role-Based Access Control*, ACM Press, 1999, pp. 55-65

[4] W. Johnston, S. Mudumbai, and M. Thompson, "Authorization and Attribute Certificates for Widely Distributed Access Control", *Proceedings of the IEEE 7th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 1998, p. 340

[5] M. Thompson, W. Johnson, S. Mudumbai, G. Hoo, K. Jackson, and A. Essiar, "Certificate-based Access Control for Widely Distributed Resources", *Proceedings of 8th USENIX UNIX Security Symposium*, USENIX Association, Washington D.C., 1999, pp. 215-227

[6] R. Bhatti, E. Bertino, and A. Ghafoor, "A Trust-based Context-Aware Access Control Model for Web-Services", *IEEE International Conference on Web Services (ICWS'04) Proceedings*, March 2004

[7] Apache Axis Project Home Page, http://ws.apache.org/axis

[8] ByungRae Cha,Jongwon Kim "Security Tactics For Secured Cloud Computing Resources"IEEE 2013.