VOL. 7, NO. 5, MAY 2012 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

ACCESS CONTROL IN CLOUD COMPUTING ENVIRONMENT

Abdul Raouf Khan
Department of Computer Sciences, King Faisal University, Saudi Arabia
E-Mail: raouf_ark@yahoo.com

ABSTRACT

Cloud computing is one of the emerging technologies. The cloud environment is a large open distributed system. It is important to preserve the data, as well as, privacy of users. Access Control methods ensure that authorized user's access the data and the system. This paper discusses various features of attribute based access control mechanism, suitable for cloud computing environment. It leads to the design of attribute based access control mechanism for cloud computing.

Keywords: cloud computing, access control, role based access control, RBAC

1. INTRODUCTION

Access control is generally a policy or procedure that allows, denies or restricts access to a system. It may, as well, monitor and record all attempts made to access a system. Access Control may also identify users attempting to access a system unauthorized. It is a mechanism which is very much important for protection in computer security. Various access control models are in use, including the most common Mandatory Access Control (MAC), Discretionary Access Control (DAC) and Role Based Access Control (RBAC). All these models are known as identity based access control models. In all these access control models, user (subjects) and resources (objects) are identified by unique names. Identification may be done directly or through roles assigned to the subjects. These access control methods are effective in unchangeable distributed system, where there are only a set of Users with a known set of services.

Nowadays, very large distributed open systems are developing very rapidly. These include Grid Computing and Cloud Computing. These systems are like virtual organizations with various autonomous domains. The relationship between users and resources is dynamic and more ad-hoc in cloud and inter cloud systems. In these systems, users and resource providers are not in the same security domain. Users are normally identified by their attributes or characteristics and not by predefined identities. In such cases, the traditional identity based access control models are not very much effective and therefore, access to the system must be done on decisions based on certain attributes.

In addition, in the cloud system, autonomous domains have a separate set of security policies. Hence, the access control Mechanism must be flexible to support various kinds of domains and policies. With the development of large distributed systems attribute based access control (ABAC) has become increasingly important.

2. ACCESS CONTROL METHODS

The first way a system provides security to its resources and data, is by controlling access to the resources and the system itself. However, access control is more than just controlling which users (subjects) can access which computing and network resources. In addition, access control manages users, files and other resources. It controls user's privileges to files or resources (objects). In access control systems various steps like, identification, authentication, authorization and accountability are taken before actually accessing the resources or the object in general.

In early stages of computing and information technology, researchers and technologists realized the importance of preventing users from interfering each other on shared systems. Various access control models were developed. User's identity was the main index to allow users to use the system or its resources. This approach was called Identification Based Access Control (IBAC). However, with the growth of the networks and the number of users, IBAC was found to be weak to defend such a large growth. Advanced concepts in access control were introduced which included owner/ group/ public. IBAC proved to be problematic for distributed systems as well. Managing access to the system and resources became hard and vulnerable to errors. A new method known as Role Based Access Control (RBAC) was introduced [1]. Role based Access Control (RBAC) determines user's access to the system based on the Job role. The role a user is assigned to be basically based on the least privilege concept [2]. The role is defined with the least amount of permissions or functionalities that is necessary for the job to be done. Permissions can be added or deleted if the privileges for a role change. However, problems became apparent when RBAC was extended across administrative domains. And it proved difficult to reach an agreement on what privileges to associate with a role. Accordingly, a policy based access control known as Attribute Based Access Control (ABAC) came into existence [3, 4]. In ABAC, access is granted on attributes that the user could prove to have such as date of birth or national number. However, reaching to an agreement on a set of attributes is very hard, especially across multiple agencies or domains and organizations. All access control methods rely on authentication of the user at the site, as well as, at the time of request. Sometimes they are labeled as authentication based access control. In all these methods, tight coupling among domains are required. This is done to merge identities or define the meaning of attributes or roles. Furthermore, all these approaches make it difficult to assign subsets of privileges of an administrator. This result

in that common use patterns, can be implemented by cutting functionalities or violating the principle of least privilege.

In [5], an attempt was made to provide a uniform frame work for ABAC specification and enforcement. A uniform framework to formulate and reason both service access and disclosure constraints based on related entity attributes was presented in [6]. A framework proposed in [7] models an ABAC system using logic programming, with set constraints of a computable set theory. Recently, the attributes based access control mode in term of its authorization architecture and policy formulation was proposed in [8]. In grid computing, ABAC systems are very actively researched and systems like PERMIS and Shibboleth have been proposed [9, 10]. Another system using ABAC is known as VOMS [11] proposed by European Data grid. It manages authorization about its own members and accordingly supplies information as attribute certificate. PERMS and Shibboleth support their own policies and cannot support other multiple policies. Hence, a more scalable and flexible solution is needed to achieve high level access control effectiveness for the heterogeneous grid environment. In addition, a more reasonable policy model is required that acts as a basic theory and uses an open architecture.

Attribute based access control extends role based access control, in general, with the following features:

i) Delegation of attribute authority
ii) Decentralization of attributes and
iii) Interference of attributes

ABAC provides policies for sensitivity of credentials. It allows organization to maintain their autonomy while collaborating efficiently. In addition, it provides an automated trust negotiation, which is auditable as and when that capability is required.

Microsoft Windows Azure platform [12] is a cloud platform used to built host and scale web applications through Microsoft data centers. It is classified as a service and forms part of Microsoft's cloud computing strategy, along with Microsoft online services. The windows Azure AppFabric's access control service is a simplified access control for web service providers. It has reduced cost and complexity for integrating various customer identity technologies. Web services can integrate with AppFabric's access control easily, instead of having to address different client identity technologies. Web services can also put together all identity models and technologies that AppFabric's access control supports through a provision process and REST (Representational state Transfer) based management API (Application Programming Interface). Web services can also allow this access control to serve as a point of integration for service customers.

3. ATTRIBUTE BASED ACCESS CONTROL

The most important security mechanism in cloud service is Access Control and traditional access control models cannot be applied to cloud services due to its characteristics. Large amount of resources, lots of dynamic users, dynamic and flexible constructions are some important features of Cloud services. In addition, each autonomous domain in cloud system has its own security policy. This may include ACL (Access Control List), SAML authorization decision assertion, and XACML policy statement. So it is important to have the access control to be flexible to have these multiple policies in multiple domains. In attribute based access control, access control decisions are taken on the basis of the attributes of the requestor, the service, the resources and the environment. Attribute Based Access Control is composed four entities.

- **A Requestor (Req):** sends requests to the cloud and invokes actions on the service.
- **A Service (Serv):** software and hardware with a network based interface and pre-defined operations.
- **A Resource (Res):** that is acted upon by one or more cloud services, with a specific set of state data in XML document.
- **An environment (Env):** contains information that might be useful in taking the access decision, such as date and time. It may not be related with any entity.

Each entity has attributes defining the identity and characteristics of its corresponding entity. The attributes of identities is defined in [13] as follows:

- $Attr(\mathrm{Re}\, q) = \{\mathrm{Re}\, qAttr_i \,|\, i \in [1, I]\}$
- $Attr(Serv) = \{ServAttr_j \,|\, j \in [1, J]\}$
- $Attr(\mathrm{Re}\, s) = \{\mathrm{Re}\, sAttr_k \,|\, k \in [1, K]\}$
- $Attr(Env) = \{EnvAttr_l \,|\, l \in [1, L]\}$

Where I, J, K and L are integers and represent the maximum number of attributes for each entity.

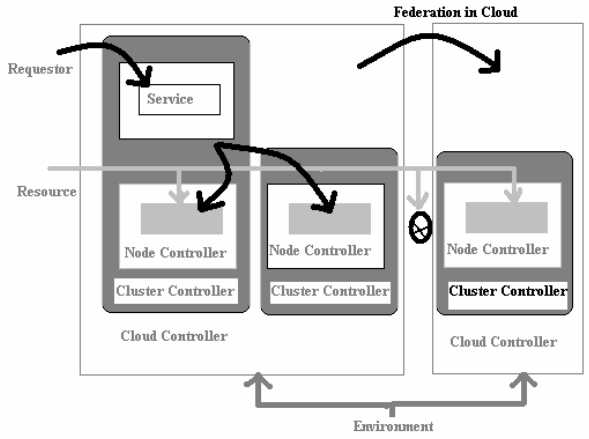

Figure-1. Cloud Environment Entities.

Security policies are supported by the authorization system of the cloud. Each system may have its own policy description method. To ensure integration of various policies and to make access control more

scalable, each policy is encapsulated as an independent unit. The policy that ABAC supports as a superset of these policies is defined as:

- Policy = $\{P_m \in [I, M], Pm$ is a policy$\}$

Access decision is taken on policy evaluation and the decision is made by the decision function df ( ). $P_{n-}df(\ )$ is the evaluation function of policy p_n and is defined as:

- $P_n\_df(Attr(req), Attr(Serv), Attr(\mathrm{Re}s), Attr(Env)) =$ permit or deny

The attributes of the entities and their attributes in ABAC cloud system is defined as follows:

- ReqAttr1 = Attribute (GID= "admin" = "#######")
- ServAttr1 = Attribute (Special Type = "PaaS", Service Name = "plateform Creation")
- ResAttr1 = Attribute (Computing = "Node1 and Node2" , networking ="switch1")
- EnvAttr1 = Attribute (Service Time = "current Time", domain = "Cloud1.Cluster1 and Cloud2. Cluster1")

And policy is evaluated by passing the attributes of the entities to the decision function df ( ).

- Decision_ABAC = df(Requestor, Service, Resource, Environment) = $P_1\_df(\mathrm{Re}quest) \& P_2\_df(Service) \& P_3\_df(\mathrm{Re}source) \& P_4\_df(Environment)$

ABAC Characteristics

- Hierarchical policy structure based on the concept of abstraction and encapsulation
- Policy set is composed of various policies that need to be supported
- Policies have their own decision and decision making algorithms
- Does not use a unified method to describe each policy
- Effective supports of multiple policies
- Model is more flexible and scalable

CONCLUSIONS

Access control decisions are very important for any shared system. However, for a large distributed system like a cloud system, access decision needs to be more flexible and scalable. This paper presents various access control methods used in cloud computing and highlights features of attribute based access control features, which are important for designing an attribute based access control.

REFERENCES

[1] Ferraiolo DF and Kuhun DR. 1992. Role Based Access Control. Proceeding of 15th National Computer Security Conference, Baltimore MD. pp. 554-563.

[2] R. Lehtinen, D. Russell and G. Gangemi Sr. 2006. Computer Security Basics. O Reilly publications, 2nd edition.

[3] M. Blaze and J Feigenbaum *et al*. The Keynote trust management system. Version 2, IETF RFC 270.

[4] A. Pimlott and O. Kiselyov. 2006. SOUTEI, A Logic Based Trust Management System. Proceeding of 8th international symposium on Functional and Logic Programming, Springer, Japan. pp. 130-144.

[5] E. Damiani *et al*. 2005. New Paradigm for Access Control in Open Environment. Proceeding of 5th IEEE International Symposium on Signal Processing and Information.

[6] P. Bonatti and P. Samarati. 2002. A unified framework for regulating access and information release on the web. Journal of computer Security. 10(3): 241-272.

[7] L. Wang, D. Wijesekera and S. Jajodia. 2004. A logic based framework for attribute based access control. Proceeding of ACM workshop on formal methods in Security Engineering. pp. 45-55, ACM press.

[8] E. Yuan and J. Tong. 2005. Attribute Based Access Control (ABAS) for web Services. Proceeding of IEEE Conference on Web Service.

[9] V. Welch *et al*. 2005. Attributes, Anonymity and Access: Shibboleth and Globus Integration to Facilaitate Grid collaboration. Proceeding of 4th annual PKI (R and D) workshop.

[10] T. Barton *et al*. 2006. Identity Federation and Attribute Based Authorization through the Globus Toolkit, Shibboleth, Gridshib and My Proxy. Proceeding of 5th Annual PKI (R and D) workshop.

[11] R. Alfteri *et al*. 2003. An Authorization System for Virtual Organizations. Proceeding of 1st European across Grids conference.

[12] http://www.microsoft.com/windowsazure.

[13] B. cha, J Seo and J. Kim. 2011. Design of Attribute Based Access Control in cloud computing. Proceeding of International conference on IT convergence and Security, Springer. pp. 41-50.