API Access Control in Cloud Using the Role Based Access Control Model

Avvari Sirisha
Computer Science
BITS Pilani Hyderabad Campus
Hyderabad, India
sirisha.avvari91@gmail.com

G. Geetha Kumari
Department of Computer Science and Information Systems
BITS Pilani Hyderabad Campus
Hyderabad, India
geethamaruvada@gmail.com

Abstract— **As cloud is an emerging paradigm of computing, it throws open various challenges and issues. The major issue hindering the growth of popularity of usage of cloud computing is Cloud security. There are numerous cloud security issues, of which this paper addresses the problem of insecure APIs. APIs act as the interface between cloud provider and the customer and the security of cloud computing depends largely on the security of these APIs. Hence a strong API access control mechanism is required. This paper proposes a two stage access control mechanism implemented at the API level using the Role Based Access Control Model (RBAC).**

Keywords-Cloud Computing; Cloud security; API Access Control; RBAC

I. INTRODUCTION

Cloud computing gets its name as a metaphor for the Internet. In essence, cloud computing is a construct that allows you to access applications that actually reside at a location other than your computer or other Internet connected device. [1]. It is an emerging frontier in Computer Science. It is service oriented and provides Infrastructure as a service (IaAS), Platform as a service (PaAS) and Software as a service (SaAS). It holds great promise for its users who take it as an opportunity to divest themselves of infrastructure management and focus on core competencies. Despite the attractive features of cloud computing, like on-demand provisioning of computing and the ability to align information technology with business strategies and needs more readily, there are concerns about the risks of cloud computing if not properly secured.[3]

This paper talks about the “Insecure APIs”, one of the major security concerns in the cloud. Cloud computing providers expose a set of software interfaces or APIs that customers use to manage and interact with cloud services. Provisioning, management, orchestration, and monitoring are all performed using these interfaces. From authentication and access control to encryption and activity monitoring, these interfaces must be designed to protect against both accidental and malicious attempts to circumvent policy. Furthermore, organizations and third parties often build upon these interfaces to offer value-added services to their customers. This introduces complexity of the new layered API; it also increases risk, as organizations may be required to relinquish their credentials to third parties in order to enable their agency [3]. Hence a strong access control mechanism is required to secure these APIs.

In this paper we propose a two stage API access control mechanism, using the Role Based Access Control Model (RBAC)

II. ACCESS CONTROL AND ACCESS CONTROL MODELS

Access control is a mechanism by which user access to certain areas and resources is controlled by an authority. There are three most widely recognized access control models. [5]

- Discretionary Access Control (DAC)
- Mandatory Access control (MAC) and
- Role Based Access Control (RBAC)

DAC is an access policy determined by the owner of an object. The owner decides who is allowed to access the object and what privileges they have. [5] MAC is an access policy determined by the system and not the owner. It is used in multilevel systems that process highly sensitive data, such as classified government and military information. [5]

RBAC is an access policy which revolves around the central concept of “role”, where role is a semantic construct around which access policy is formulated. RBAC is well suited for enterprises and commercial applications, and that is the reason why it has grown popular over the years.

III. ROLE BASED ACCESS CONTROL MODEL

The central notion of RBAC is that permissions are associated with roles and the users are assigned to appropriate roles. Thus roles serve as a layer of abstraction between the users and permissions. This greatly simplifies the management of permissions. Users can be easily reassigned from one role to another. Roles can be granted new permissions as new applications and systems are incorporated, and permissions can be revoked from roles as needed [2]. There are three primary rules [6] defined for RBAC

- Role Assignment
- Role Authorization
- Transaction Authorization

978-1-4244-9008-0/10/$26.00 ©2010 IEEE

Core RBAC recognizes five administrative elements (1) Users (2) Roles and (3) Permissions, where permissions are composed of (4) Operations applied to (5) Objects [2]. The user, role and permission relationships are depicted in the "Fig. 1".

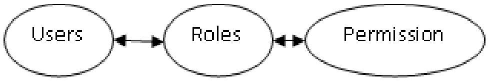

Figure 1. User, role and permission relationships in RBAC

This arrangement provides great flexibility and granularity of assignment of permissions to roles and users to roles. Any increase in flexibility in controlling access to resources also strengthens the application of the principle of least privilege [2].

IV. The Two Stage Api Access Control Policy

The access control policy proposed here is implemented at the API level. The basic representation of the showing the interaction between the user (customer) and the cloud service through the cloud API is shown in the "Fig. 2". Before accessing any resource, the user must be authenticated. As authentication is not in the purview of this paper, we assume that the user has already been authenticated by some mechanism.

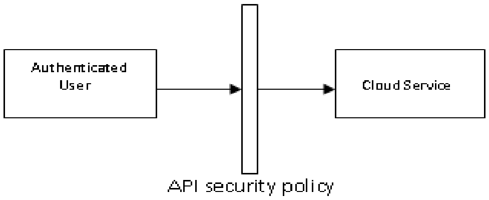

Figure 2. The basic representation of user-service interaction in cloud through the cloud API

As mentioned earlier, our access control policy is implemented at the API. The schematic of the two stage access control mechanism is shown in the "Fig. 3". When we refer to "user", we mean an "authenticated user".

When the user is authenticated, along with his credentials, his attributes are also taken. These attributes may be the IP address of the machine he is using to access the cloud service or the domain name. Suppose that the user belongs to an organization which has registered itself with the cloud service provider. Then these attributes help in identifying the organization to which the user belongs to. This is done by checking the attributes against the database which maintains the list of registered users with their domain names or any other identifier.

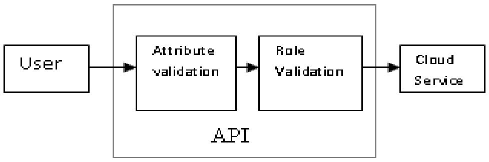

Figure 3. A broad view of the proposed Two- Stage Access Control Policy

registers itself with the cloud service provider, both the organization and the cloud service provider agree upon a set of roles, the permissions associated with the roles and the users attached to these roles. Any change in the organization's policy needs to be updated immediately to avoid security breaches.

So once the user logs in, and his organization is identified (by the domain), the user role is determined with the help of the database maintained for the same. Then the user is permitted to access the resource with a set of permissions mapped to his role.

Clearly the first stage acts as the first line of defense and also provides the information required for the second stage of access control. Once the user crosses the first stage, his/her actual permissions to the object (here, the cloud service) are determined by his/her role and hence the Role Based Access Control takes over from here.

Each of the stages mentioned above is a mechanism in itself, involving interaction with the underlying database and making policy decisions.

The detailed representation of these two stages can be seen in the "Fig. 4" and "Fig. 5" respectively.

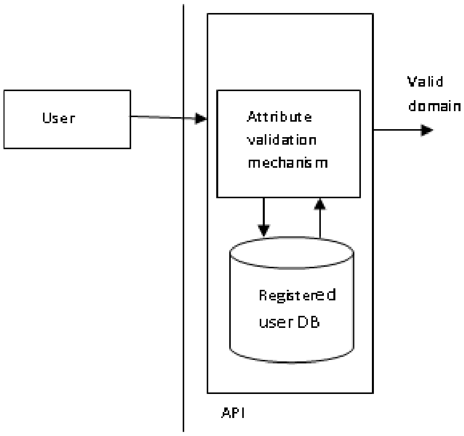

Figure 4. The First stage of Access Control

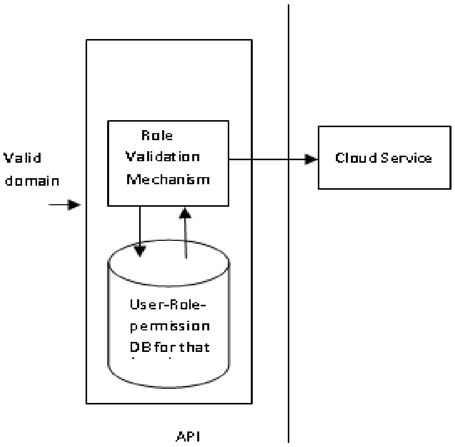

Figure 5. The Second stage of Access Control

IV. CONCLUSION

This model ensures a two stage security at the API level. The first stage to ensure that only registered users from white listed domains can access the cloud service and at the same time extracts the required input for the second stage. Role Based Access Control Model has been chosen because it is greatly suited for the commercial and enterprise needs. It also becomes easy for the organization to map a user's organization role (local role) onto the role with respect to the service to be accessed (global role). The implementation of this model is in progress.

ACKNOWLEDGMENT

I'd like to thank my guide and the co-author of this paper, Dr. G. Geetha Kumari for her constant support and guidance throughout this project.

REFERENCES

[1] Anthony. T. Velte, Toby. J. Velte, Robert Elsenpeter, "Cloud Computing: A Practical Approach", The McGraw-Hill Companies, 2010.

[2] David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli, "Role Based Access Control", Artech House, Boston, London, 2003

[3] Cloud Security Alliance, "Top threats to Cloud Computing V1.0", Cloud Security Alliance, March 2010

[4] Cloud Security Alliance, "Security guidance for Critical Areas of focus in Cloud Computing V2.1", Cloud Security Alliance, December 2009

[5] Wikipedia – Access control http://en.wikipedia.org/wiki/Access_control

[6] Wikipedia – Role Based Access Control http://en.wikipedia.org/wiki/Role-based_access_control