International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-4, Issue-6)

Research Article

June 2015

Extended JV-RBAC Model with Secure API Access Control in Cloud

Jyoti Joshi
Department of Computer Science & Engineering
Seemant Institute of Technology,
Uttarakhand, India

Abstract–

As cloud is a growing paradigm of computing, it throws open various challenges and issues. The major issue hindering the growth of recognition of usage of cloud computing is Cloud security. Cloud security is security principles useful to protect data, applications and infrastructure connected within the cloud computing technology. There are many cloud security issues, of which this paper addresses the problem of insecure APIs. APIs act as the boundary between cloud provider and the customer and the security of cloud computing depends basically on the security of these APIs. Hence a strong API access control method is required. Access control, a method for constraining the interface between users and protected resources. Commonly, access control is concerned with controlling which users have access to which resources in computer systems. Role-Based Access control (RBAC) is one of the most powerful access control mechanisms with many possible applications. RBAC models have concerned significant research interest in past time due to their providing some flexibility to secure management and ability to model organizational structure and their capability to reduce organizational expenses. This paper proposes an access control mechanism implemented at the API level using the Joshi Vaisla-Role Based Access Control (JV-RBAC) model.

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

I. INTRODUCTION

Cloud computing (‘cloud’) is a growing term that describes the improvement of many open technologies and approaches to compute into something different. Cloud separates application and information resources from the basic infrastructure, and the mechanisms used to convey them. Cloud enhances association, agility, scaling, and availability, and provides the potential for cost reduction through optimized and efficient computing. More specially, cloud describes the use of a collection of services, applications, information, and infrastructure comprised of pools of compute, network, information, and storage resources. These components can be rapidly orchestrated, provisioned, implemented and decommissioned, and scaled up or down, providing for an on-demand utility-like model of share and utilization [1].

Cloud computing gets its name as a symbol for the Internet. In real meaning, cloud computing is a creation that allows you to access applications that actually reside at a location other than your computer or other Internet connected device. Cloud computing represents one of the most major shifts in information technology many of us are likely to see in our lifetimes. It is an rising edge 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 separate from themselves of infrastructure management and focus on core skills. Regardless of the attractive features of cloud computing, like on-demand provisioning of computing and the capability to align information technology with business strategies and needs more readily, there are concerns about the risks of cloud computing if not properly secured [2].

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. APIs exposes organization to a variety of security issues related to confidentiality, integrity, availability and accountability. Hence a strong access control mechanism is required to secure these APIs [3].

In this paper we propose an access control mechanism, using the JV- Role Based Access Control Model (JV-RBAC).

II. CLOUD COMPUTING

NIST (National Institute of Standards and Technology) defines cloud computing by describing five essential characteristics, three cloud service models, and four clouds deployment models. They are summarized in visual form in Fig. 1 and explained in detail below [1].

A. Essential characteristics of Cloud Computing

Cloud services show five essential characteristics that exhibit their relation to, and differences from, conventional computing approaches [1]:

1) *On-demand self-service:* A consumer can unilaterally provision computing capabilities such as server time and network storage as needed automatically, without requiring human interaction with a service provider.
2) *Broad network access***:** Capabilities are available over the network and accessed through standard mechanisms that promote use by mixed thin or thick client platforms (e.g., mobile phones, laptops, and PDAs) as well as other traditional or cloud-based software services [1].
3) *Resource pooling:* The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a degree of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources, but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines [1].
4) *Rapid elasticity***:** Capabilities can be rapidly and elastically provisioned- in some cases automatically- to quickly scale out, and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time [1].
5) *Measured service:* Cloud systems automatically control and optimize resource usage by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, or active user accounts). Resource usage can be monitored, controlled, and reported- providing transparency for both the provider and consumer of the service [1].

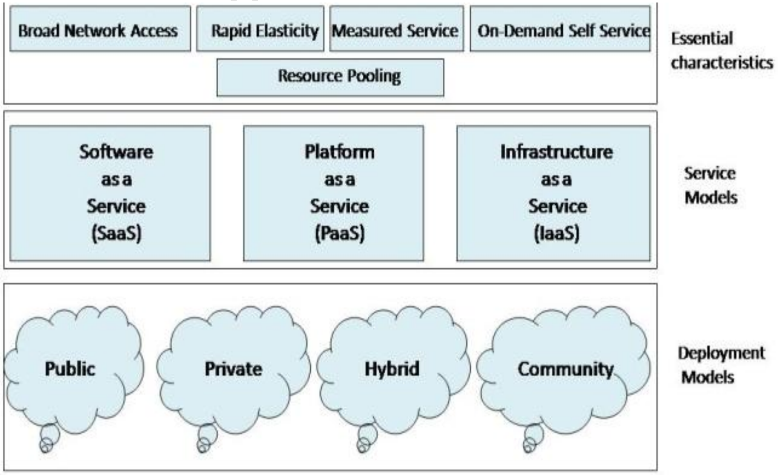

Fig . 1. NIST Visual Model of Cloud Computing Definition

B. Cloud Service Models

Cloud service is divided among three architectural models and various derivative combinations. The three fundamental classifications are often referred to as the "SPI Model", where 'SPI' refers to Software, Platform or Infrastructure (as a Service), respectively [1].

1) *Cloud Software as a Service (SaaS):* The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings [1].
2) *Cloud Platform as a Service (PaaS):* The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations [1].
3) *Cloud Infrastructure as a Service (IaaS):* The capability to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems; storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls) [1].

III. ACCESS CONTROL AND MODELS

Access control is a core concept in security. This can be done through authentication, authorization, and access control. These three mechanisms are distinctly different but usually generally it can effectively manage all requests for access systems and it can protect the authorized users to access information systems from unauthorized access. There are three most widely recognized access control models [8].

- 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. 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. 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 [8].

IV. ROLE BASED ACCESS CONTROL

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. Roles are created for the various job functions in an organization and users are assigned roles based on their responsibilities and qualifications. 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. The role's concept was introduced between the users and authority, and the roles and users will be linked, through the role authorization to control the access of system resources. There are three primary rules defined for RBAC [4]:

- Role Assignment
- Role Authorization
- Transaction Authorization

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

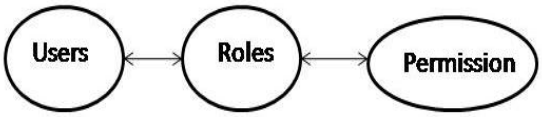

Fig 2. User, role and permission relationships in RBAC

V. JV-RBAC WITH API ACCESS CONTROL

The access control policy proposed here is implemented at the API level in JV-RBAC (Joshi-Vaisla Role Based Access Control) model. The basic representation of showing the interaction between the user (customer) and the cloud service through the cloud API is shown in the Fig. 3. 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.

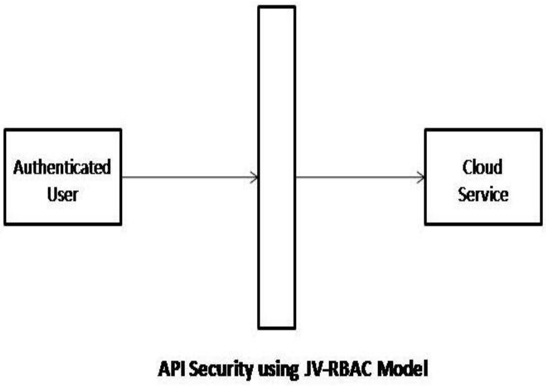

Fig 3. Representation of user-service interaction in cloud through cloud API using JV-RBAC model

As mentioned earlier, our access control policy is implemented at the API. The schematic of the three stage access control mechanism in JV-RBAC model is shown in the Fig. 4. When we refer to "user", we mean an "authenticated user".

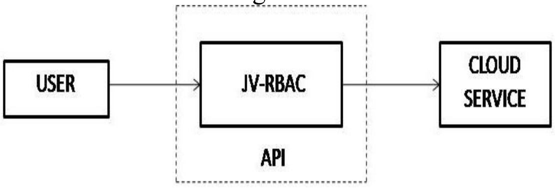

Fig .4 A broad view of the proposed Three-Stage Access Control Policy

When the user is authenticated, along with his identification, 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 a business 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. registers itself with the cloud service provider, both the organization and the cloud service provider agree upon a set of roles, the permissions related with the roles and the users attached to these roles. Any change in the organization's policy needs to be updated directly to avoid security breaches [2].

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 allowed to access the resource with a set of permissions mapped to his role.

Clearly the first stage acts as the first line of security 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 item (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 interface with the primary database and making policy decisions.

The detailed representation of API access control in clouds in JV-RBAC model can be seen in the Fig. 5. shown below. In this model, we presented such aspects as access rules, access control granularity, modules, time constraints, as well as auditing strategy at the API level.

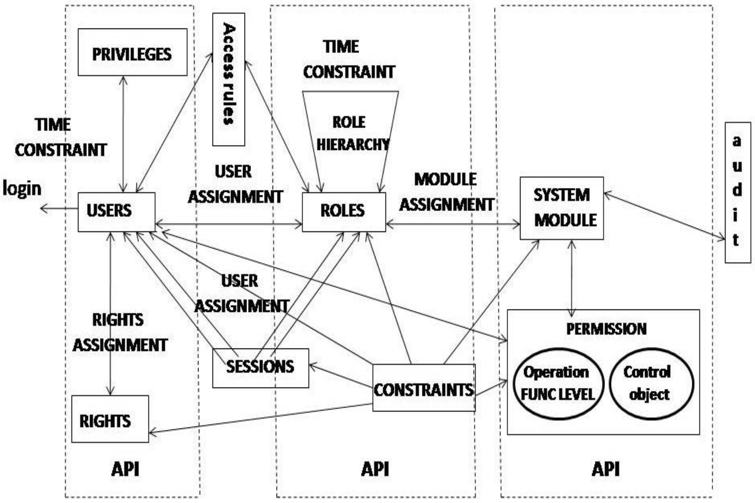

Fig .5 API access control in JV-RBAC model

The introduction of modules, the objects can be fine-grained, multi-dimensional (at the time, space, timing, etc.) access control. The modules and roles must combine to assign permission, which to combine with user authorization and roles authorization to make a more flexible authorization.

In unity with the special requirements use the related authorization. To refine the role in each of the modules and systems departments, that each role can only operate specific modules.

The introduction of the time constraint, users can lock their own account number while they aren't on line. Even if the illegal use of such accounts, lock-in period of time cannot be used, as to improve protection against unauthorized access.

The introduction of the audit not only can reproduce the original process and issues, but also to track and record operation of the administrators and prohibited users in the log. It is required to track responsibility and data recovery. The introduction of access rules, to make up for lack of the traditional RBAC static constraint and dynamic constraint. It provides more fine-grained access control for System, but also can reduce the workload of the traditional role of the RBAC model in the distribution of the authorization.

VI. CONCLUSION

This model ensures a three 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 and the output of second stage is used as an input for the third stage. JV-RBAC Model has been chosen because it is greatly suited for the business 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.

REFERENCES

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

[2] A. Sirisha, G .Geetha Kumari, "*API Access Control in Cloud Using the Role Based Access Control Model*", Trendz in Information Sciences & Computing (TISC),pp.135-137, 2010.

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

[4] David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli, “*Role Based Access Control*”, Artech House, Boston, London, 2003

[5] Dr. K. S. Vaisla, J.Joshi, "Modified and Improved Extended –Role-Based Access Control Model", Proc. of National Conference on Business Intelligence and Data Warehousing, by Narosa Publication pp.206-213, March 2012.

[6] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E.Youman, “*Role-based access control models*”, IEEE Computer, vol.29, pp.38-47, February 1996

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

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