Multi Tenancy Access Control Using Cloud Service in MVC

Sonia gupta

Abstract—Cloud Computing is the next generation Internet service and data center, and it is also used for public utilities and on-demand computing. Cloud computing is not a totally new technology, but rather a derived concept of application and service innovation in which, multi-tenancy is one of the important issues among the core technologies of cloud computing applications. Many tenants can access the different applications and computing resources in the same cloud server, whereas concurrent use by many users on a database or application will lead to large data volume, time consuming and security issues. Under these circumstances, it is particularly important to separate application and data for conflicts avoidance to enhance the system and data security. This paper emphasizes the cloud service model under a Multi-Tenant Architecture (MTA), using identity management and Role Based Access Control, to propose a Design Security Multi-Tenancy Access Control (DS-MTAC).The DS-MTAC applies identity management to determine the user's identity and applicable roles, since different users possess different functional roles with respective privileges for processing. Such role-based assignments can easily and efficiently manage a user's access rights to achieve application independence and data isolation for improving the processing performance of cloud multi-tenant services and hardening the security and privacy of cloud applications

Index Terms—Azure Development,MVC Cloud,Security,DLL

1 INTRODUCTION

People used to draw a cloud to represent the Internet, and that is the cloud we know today. There are an increasing number of people setting up their data or websites in the cloud, even using software based on cloud services (i.e. Salesforce.com) to help with corporate operations. There are also many organizations and corporations that have started to develop cloud computing in Taiwan. The Multi-Tenancy Control is a cloud platform technology, which covers SaaS, PaaS and IaaS. The multi-tenancy access control to the cloud platform is considerably important. When a massive number of tenants share the same hardware/software resources, each tenant undergoes customized configuration according to the resources needed without affecting the usage of other users. Research shows that the top concerns among corporations with regards to SaaS service are in security and privacy issues and service quality with 80% and 70% of the respondents respectively. Such data clearly indicate that users are concerned about the cloud security and privacy, therefore application independence and data isolation becomes an issue that can not be neglected under the MultiTenant Architecture (MTA) today. Also,, the multi-tenancy system is considerably more important in the SaaS cloud service under MTA.This paper emphasizes on the use of identity management and Role-Based Access Control (RBAC) under the multi-tenant cloud environment, so that each user is assigned one to many roles while each role is assigned to different privileges. Such privilege assignment can easily and effectively manage the access rights of users, thereby maintaining application independence and data isolation or protection while improving process performance and cloud security.

This paper takes into consideration the different perspectives of roles, since different users can be assigned to different role sets, which corresponds to different privileges. Moreover, the different privileges allow access to different Web sites and different data in order to improve security and efficiency. Using the characteristics of different user authority could produce environmental isolation and data isolation. As distinguished from the Discretionary Access Control and Mandatory Access Control, the RBAC offers more flexibility in the granting of privileges, making it easier to manage. Users are assigned to a role and granted privileges that correspond to that role directly. Using the role-based access control simplifies the management of privileges, since the addition, deletion, query, and modification of privileges are applied to the "role" instead of the individual user. Based on the above reasons, this paper proposes RoleBased Multi-Tenancy Access Control called RB-MTAC that integrates a set of identity management and RBAC with consideration of multi-tenant and multi- user cloud environment. The RB-MTAC method can easily assign the functions or resource with access privilege to users, in order to enhance processing performance, quality of service (QoS), and security as well as privacy on the cloud. Hence, the purposes of RB-MTAC scheme in cloud computing are follows.

(1) RB-MTAC combines the identity management and role-based access control of a multi-tenancy environment in cloud computing, which is effective and simple to manage privileges that protect the security of application systems and data privacy.

(2) The method provides good identity management and access control to privileges to block a non-tenant user accessing through identity management, while access control prevent tenant users without specific privilege from viewing and accessing specific applications and database.

(3) The method also offers a mechanism that manages user privileges using role-based viewpoints, so administrators only need to modify role privileges to easily change user privileges, so that would reduce potential errors from constant modifications.

(4) Based on the RB-MTAC method, building a prototype system and simulation experiment, which supports this is better than user-based system.

(5) The application independence and data isolation of different tenants. Under the cloud environment, the systems and data of different tenants could be stored in the same place and that can prevent other tenants using the data and system of that tenant either intentionally or unintentionally.

LITERATURE REVIEW

For both the grid computing and cloud computing paradigms, there is a common need to be able to define the methods through which consumers discover, request, and use resources provided by third-party central facilities, and also implement highly parallel and distributed computations that execute on these resources. Grids came into existence in the mid 90s to address execution of large scale computation problems on a network of resource-sharing commodity machines that would deliver the same computation power affordable only with expensive supercomputers and large dedicated clusters at that time. A grid could typically comprise of compute, storage and network resources from multiple geographically distributed organizations, and these resources are normally considered to be heterogeneous with dynamic availability and capacity. The two primary concerns for grid were interoperability and security, as resources come from different administrative domains with varying global and local resource usage policies, as well as different hardware and software configurations and platforms. Most grids employ a batch-scheduled compute model with suitable policies in place to enforce the identification of proper user credentials under which the batch jobs will be run for accounting (e.g., the number of processors needed, duration of allocation, etc) and security purposes. Condor [1] is a centralized workload management system suited for computation-intensive jobs executed in local closed Grid environments. Its resource management mechanism is similar to that of UNIX (discretionary access control), with some additional modes of access besides the traditional read and write permissions. Legio uses an object-oriented approach wherein all files, services and devices are considered as objects, and are accessed through functions of these objects. Each object can define its own access control policy, typically done using access control list and authentication mechanisms, in a default *MayI* function that is invoked before any other functions of the object may be called. The Globus Grid Toolkit (GT) proposes mechanisms to translate users' grid identities into local identities (which can in turn be verified by the resource providers using appropriate local access control policies) and also allow users' certificates be delegated across many different sites. With the single sign-on mechanism (e.g., Open Grid Service Infrastructure, OGSI [4]), users can login only once and have access to multiple grid sites, as well as programs can be authorized to access resources on a user's behalf and can further delegate them to other programs. The OGSI operates in conjunction with resource usage brokers (e.g. Gruber) that act as distributed policy enforcement points to enforce both local usage policies and global service level agreements (SLAs) and allow resources at individual sites to be efficiently shared across multiple sites. In the authors propose a flexible attribute-based multi-policy access control (ABMAC) model for grid computing systems in which each autonomous domain may have its own security policy. ABMAC is based on the idea of integrating the individual authorization decisions arrived at for user requests to access resources/services (all of which are identified with their characteristics or attributes) according to the security policy of each domain and arriving at a final decision using a combination algorithm that can be adapted to suit to the resource/operating constraints. The ABMAC approach is more scalable compared to developing a superset of individual domain policies and evaluating user request for resource access according to this superset.

Task Role-based access control model (TRBAC) has been considered a viable model for cloud computing environment wherein the traditional static access control models such as discretionary, mandatory or simple role-based models cannot be employed. TRBAC can dynamically validate access permissions for users based on the assigned roles and the task the user has to perform with the assigned role. Tasks could be classified as workflow tasks (those that need to be completed

in a particular order) that require active access control and non-workflow tasks (those that can be completed in any order) that require passive access control. Workflow tasks driven active role-based access control is time sensitive and the access permissions assigned for users performing these tasks change dynamically with time, depending on the order in which the tasks are to be executed.

PROBLEM FORMULATION

To be scalable, access control policies need to be defined for groups of VMs that comprise a tenant. Due to the characteristic of sharing of physical resources among tenants whose trustworthiness cannot be easily captured, there is an increased risk of side-channel attacks based on information obtained from physical implementation (e.g., time- or bandwidth-monitoring attacks). Also, interference of computation from multiple tenants (mainly due to the possibility of existence of covert channels with flawed access control policies) can result in unauthorized information flow on the physical host. A centralized mechanism to globally manage access control can involve a significantly larger number of authorization rules that grows substantially with an increase in the granularity of resources, as well as with the number of users and services supported by the cloud. Today's cloud computing environments demand a varying degree of granularity in the access control mechanisms due to the heterogeneity of services provided.

In a multi-tenant control or system, tenants are included in all system data that could be identified for assigned users, such as the account number and statistical data; users can build the various data in the system, and the customized applications of users all belong to the scope of the tenant. The tenants use the application system or computing resources developed or built by the cloud suppliers, where the application system is designed with the capacity of multiple tenants under the same environment. To provide multiple tenants with the ability to use the same application on the same cloud computing environment, the application and computing environments should be designed carefully. In addition to allowing multiple identical applications to concurrently execute on the cloud platform, the protection of the tenant's data security and privacy is also one of the key multi-tenancy technologies. Basically, the key technology of a multi-tenancy system implementation lies on Application Context Independence and Data Isolation of different tenants in order to maintain the different applications between tenants without mutual interference, while maintaining adequate data confidentiality.

1) Applications: The procedures or carriers environment for supporting the concurrent execution of multiple applications or using the way of threads in the same server program are used for program isolation.

2) Data: The different mechanisms are used to isolate the different tenant data. For example, the Force.com adopts metadata technique for data isolation and Microsoft NSDN isolates the technical files via using the structured definitions.

In this paper the following problems has been addressed:-

-No reliability over application programming, interface due to login in main dedicated server of AWS/AZURE for assigning role and policy.
-In order to make a profile system or application machine id, we are not able to configure database over the server due to license or technology.

The major objectives of this paper are:

-Develop a membership based system on an intranet portal or a dedicated server IP configuration.
-Introduce verification steps for user.
-The following points should be taken care of:-
- Development on client interface in AZURE
- Manually add my own membership based database structure
-Provide an open VPN/SSH certification to every role register with intranet application
- Update every single access password role on a repository
-API/SDK implementation to use azure

METHODOLOGY

This paper proposes the Design Security –MTAC (DS-MTAC) mainly to integrate the identity management and role-based access control under the multi-tenant cloud environment. In DS-MTAC, cloud tenants may achieve identity management and conform to the privilege attributed to the roles in order to determine whether or not the various application functions and system data can be displayed or modified. Basically, the preliminary plan for the operating procedures of DS-MTAC is described below.

(1) When logging in, the users must be authenticated as a user of the tenancy.

(2) If the user is a valid tenant member and both the account number and password are correct, the user role set will be captured.

(3) An account lockout setting of three attempts is proposed to prevent unauthorized personnel from gaining access to the system.

(4) Generating ACL through the users' role set.

(5) After entering the system's application home page,the user's ACL will be applied to determine whether the application's function and content should be displayed or hidden, and then present the results to the application.

The major steps are given below:

Every tenant or user need to login into the system using an account number and password

Through identity management, it authenticates the user or tenant account.

If the users are valid, the role assignment will capture the role corresponding to the user from the RB-MTAC database and assign the role and access rights, which belong to the user.

The user can carry out activities, access functions and computing resources in the system through RB-MTAC.

RB-MTAC combines the identity management and role-based ace ess control of a multi tenancy environment. It manages the privileges from viewing and accessing specific applications and database.

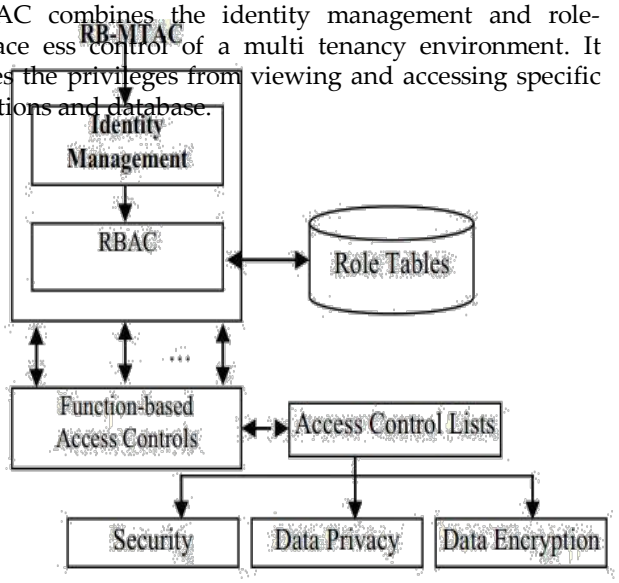

Fig 1: DS-MTAC Architecture

CONCLUSION

A Design Security Multi-tenancy based system has been designed and implemented and the results have been shown and obtained as desired. The scheme proposed in this paper can separate the online tenants from cloud servers while capturing a user's roles and privileges immediately after logging into the system. The users' privileges are clear in the cloud application, which also helps the platform administrator to manage the user's privileges. The main contribution of this paper is to provide a set of privileges and the identity management scheme for corporations in cloud computing environment. Such a scheme can be used to easily change employee privileges in the event of personnel changes and modify the role privileges directly when adding new functions to the system without the need to modify all employee privileges one by one.

REFERENCES

[1] L. Popa, M. Yu, S. Y. Ko, S. Ratnasamy and I. Stoica, "CloudPolice: Taking Access Control out of the Network," Proceedings of the 9th ACM Workshop on Hot Topics in Networks, October 2010.

[2] S. Oh and S. Park, "Task-role-based Access Control Model," Information Systems, vol. 28, no. 6, pp. 533-562, September 2003.

[3] H. A. J. Narayanan and M. H. Gunes, "Ensuring Access Control in Cloud Provisioned Health Care Systems," Proceedings of the IEEE Consumer Communications and Networking Conference, 2011.

[4] S. Sanka, C. Hota and M. Rajarajan, "Secure Data Access in Cloud Computing," Proceedings of the 4th IEEE International Conference on Internet Multimedia Services, December 2010.

[5] S. Yu, C. Wang, K. Ren and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud

Computing," Proceedings of the 29th IEEE International Conference on Information Communication, pp. 534-542, 2010.

[6] E. E. Mon and T. T. Naing, "The Privacy-aware Access Control System using Attributed-and Rolebased Access Control in Private Cloud," Proceedings of the 4th IEEE International Conference on Broadband Network and Multimedia Technology, pp. 447-451, October 2011.

[7] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based Encryption for Fine-Grained Access Control of Encrypted Data," Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89-98, 2006.

[8] J. Bethencourt, A. Sahai and B. Waters, "Cipher text-Policy Attribute-Based Encryption," Proceedings of the IEEE Symposium on Security and Privacy, pp. 321-334, 2007.

[9] K. Yang and X. Jia, "Attribute-based Access Control for Multi-Authority Systems in Cloud Storage," Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems, pp. 536- 545, 2012

[10] T. Ristenpart, E. Tromer, H. Shacham and S. Savage, "Hey, You, Get off my Cloud: Exploring Information Leakage in Third-Party Compute Clouds," Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 199-212, 2009.