iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** X **Month of publication:** October 2018

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue X, Oct 2018- Available at www.ijraset.com

An Enhanced Cloud Computing Security Model by using RBAC Access Controls

Miss Nidhi Saxena
Department Of Computer Science and Information Technology, Gautam Buddha University,Greater Noida ,Uttar Pradesh

Abstract: Cloud computing is a fast growing field which is arguably a new computing paradigm. In cloud computing, computing resources are provided as services over the Internet and users can access resources on based on their payments. This paper discusses cloud computing and its related security risks, with a focus on access control. As a traditional access control mechanism, role-based access control (RBAC) model can be used to implement several important security principles such as least privilege, separation of duties, and data abstraction. This paper shows an on-going effort by refining entities in RBAC used for cloud computing, and further discusses their security implications. We argue that RBAC is well suited to many situations in cloud computing where users or applications can be clearly separated according to their job functions.
Keywords: Cloud computing, cloud security, Role-Based Access Control (RBAC)

I. INTRODUCTION

Cloud computing is a fast growing field which is arguably a new computing paradigm. In cloud computing, computing resources are rendered as services over the Internet. Various definitions on cloud computing can be seen from the academia, the government, and the industry

According to the US National Institute of Science and Technology (NIST), "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [2] In a more precise definition, cloud computing is defined as "both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services." [16] As a low-cost solution, cloud computing provides computation, software, data access, and storage services that are transparent to end users. Its basic characteristics can be briefly summarized as follows [2].

1) On-demand self-service. A consumer can unilaterally provision computing capabilities as needed with little or no human intervention.
2) Broad network access. Resources can be accessed over the network using existing mechanisms (e.g., current Internet protocols).
3) Resource pooling. Resources are pooled to provide services to multiple consumers. Physical and virtual resources can be dynamically assigned and reassigned according to consumers' demand.
4) Rapid elasticity. Resources can be rapidly and elastically provisioned to meet customers' needs.
5) Measured Service. Resources are automatically controlled and optimized depending on the type of service (e.g., storage, processing, bandwidth, and active user accounts). Their usage can be monitored, controlled, and reported, and the process is transparent to both the provider and the consumer [2].

In cloud computing, resources could include storage, processing, memory, network bandwidth, and virtual machines. The service models of cloud computing generally follow the Software-Platform-Infrastructure (SPI) model. It represents three major services provided through the cloud [1][2][3]: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

Depending on the nature of customers, a cloud can be deployed as a private cloud, community cloud, public cloud, and hybrid cloud. Cloud computing is essentially a centralized (from the users' perspective) computing facility built on a large-scale service model. It has been argued, especially by the academia, that cloud computing is nothing new than its predecessors such as autonomic computing, client-server model, grid/cluster computing, mainframe computers, utility computing, service-oriented computing, Web 2.0, platform virtualization, Service Oriented Architecture (SOA), and peer-to-peer networks, although the resources can be provided on a much larger scale compared to previous applications [3].

Cloud computing has been quickly promoted by the industry during the past five years. Aside from the huge marketing efforts, cloud security has been criticized for its unknown privacy and security protection. There could be benefits from a security perspective since most customers utilizing cloud may not have the expertise to safeguarding their information assets using

traditional IT approaches, and using cloud services could mitigate this problem. On the other side, companies hosting the cloud services have in general full control over the services they provide.
They could control and monitor data essentially at will.
There could also be other security issues such as access control, data protection, and management of cloud resources. It has been noted by the research community that confidentiality and auditability are one of the top 10 obstacles to the growth of cloud computing [16].
This paper intends to focus on the access control aspect of cloud security, and provide a protection mechanism based on the Role Based Access Control (RBAC) model. The use of RBAC in cloud computing and database systems is not new; however in this paper we intend to use security patterns to show the mechanism. We also briefly show our preliminary identification of several important entities in cloud computing so that the RBAC could be instantiated
[17]. The research being discussed here shows an on-going research effort on applying RBAC in cloud computing.
The paper is organized as follows. Section 2 provides a general discussion on cloud security from the providers' perspective. Section 3 shows a formal definition on RBAC. Section 4 discusses in detail how RBAC could be employed in cloud security. Section 5 demonstrates implications of using RBAC and some areas in which RBAC may not fit well. Section 6 gives conclusion and directions for future research.

II. SECURITY RISKS FOR CLOUD PROVIDERS

Security risks in cloud computing environments involve traditional paradigms in information security such as confidentiality, integrity, and availability (sometimes referred to as the CIA triad). However they have contextual characteristics in cloud computing. For example, for most service models, the security is largely the responsibility of the cloud providers. It is then essential to identify risk issues faced by the virtualized systems. These issues include the following [3]

A. Complexity of configuration. Due to more complex usage of networks and systems, the possibility of improper configuration may increase. Such information may not be aware to consumers until some security incidents happen.
B. Privilege escalation. An attacker may take advantage of different levels of access controls of Virtual Machines (VM) and escalate its access privileges through the use of hypervisor – a virtual machine monitor/controller that facilitates hardware virtualization and mediates all hardware access [3].
C. Inactive virtual machines. Data stored in inactive virtual machines may contain sensitive information and has the potential to be accessed by unauthorized users.
D. Segregation of duties. Since a VM provides access to different components using different mechanisms, properly identify access roles and segregate their duties could be difficult.
E. Poor access controls. A hypervisor is basically a single point of access. It has the risk of exposing trusted network resources through poorly defined access control systems.

In addition to these issues, there are also risks related to data encryption, the use of traditional security network protocols (for example, XML security, transport layer security or TLS), browser security, middleware security, denial of service attacks, among others [4]. It appears that policy and management issues are more evident and play a bigger role in cloud security. These issues include disaster recovery and business continuity, regulatory compliance, secure design and test process, among others. So far no comprehensive study on cloud security can be observed in the literature, partly due to the immaturity of cloud computing. As many of these security issues are highly specific to cloud providers, general solutions to these issues could be difficult to obtain.
Like the definition of term "cloud computing" itself, there were some arguments in the community that very few security issues in cloud computing are fundamentally new or intractable compared to those of in-house IT environments [16][18]. It has been argued that two relatively "new" and fundamentally important security problems are: complexities of multi-party trust considerations (when a long trust chain is formed due to the use of multiple service providers) and the ensuing need for mutual auditability (as both the cloud provider and the cloud user could be the source or the target of an attack.) [18].
Access control is the process of limiting access to system resources for only authorized people, programs, processes, or other system components.
Access control is one fundamental aspect of information security, and directly ties to the three aspects of the information security triad. From the perspective of access control, cloud computing providers should provide the following basic functionalities [5].

1) Control access to the cloud service's features based on policies specified by the customer, and the level of service purchased by the customer.
2) Control access to one consumer's data from other customers in multi-tenant environments. Also control access to both regular user functions and privileged administrative functions.
3) Keep user profile information and access control policy accurate.
4) Provide optional notification of account creation and removal.
5) Provide adequate liability/audit logs on consumer and service provider activities. This seems important in the context of the previous discussion on the mutual auditability problem.

These functionalities can also be observed in access control mechanisms for traditional IT projects. To enforce access control, it has been mentioned that all the following traditional models can be used in cloud computing: mandatory access control (MAC), discretionary access control (DAC) (for example, access control lists or ACLs), and nondiscretionary access control (for example, RBAC or task-based access control) [3][5]. The use of a model is highly specific to cloud providers. It should be noted that there are special security implications due to the use of single sign-on (SSO) in cloud computing. SSO is a mechanism that one user provides an ID/password pair (or other identity information such as biometric features or digital certificates) per work session, and is then automatically granted access to all the required applications. Although this seems convenient to consumers, it creates a single point of failure that may be highly susceptible to external attacks.

We argue that RBAC is a traditional access control model and may be well-adapted to some situations in cloud computing, especially in situations where users or applications are clearly separable according to their job functions. We will continue the discussion with a formal definition and a UML representation on RBAC, followed by its potential usage in cloud computing.

Despite the arguments in the community, this paper intends to focus on the access control aspect of cloud security, and provide a protection mechanism based on the Role Based Access Control (RBAC) model. The use of RBAC in cloud computing and database systems is not new; however in this paper we intend to use security patterns to show the refined mechanism. We also identify several important entities in cloud computing at a refined level so that the RBAC could be instantiated.

III. ROLE-BASED ACCESS CONTROL (RBAC)

The RBAC model is shown using the following formal notations. Suppose *U* is a set of users, *R* is a set of roles, *OBS* is a set of protected objects, *OPS* is a set of operations, and *S* is a set of sessions. We then have the following definitions [6][7]:

: A many-to-many user-to-role assignment relation;

: A many-to-many object-to-role assignment relation;

: A set of permissions in relation to the operations and protected objects;

: A many-to-many mapping permission-to-role assignment relationship;

:Permission-to- operation mapping that gives the set of operations associated with permission p;

: Permission-to-object mapping that gives the set of objects associated with permission p;

! " : A mapping of each user *u* onto a set of sessions;

"#$% : A mapping of session s_i to a single user. For example, $user(s_i) = u_j$ is valid during the session's lifetime;

% &$# ' : A mapping of each session s_i to a set of roles;

In the definition it should be noted that a session belongs to a single user, and a user may assume many roles. The RBAC model is also used to define Role Hierarchies (RH). A partial ordering is defined as: if ( ) * , then role *x* inherits the permissions and users of role *y*. We have

If % &$# #+ %, -%) % . "#$% #+ / % 0 1 , then session s_i has access to protected objects $2_{303\ 456\ 67}$, -% 8 % . / % 0 1 , which are all protected objects that can be accessed by roles at lower orders [10].

We can also define

9 *:* A partially ordered role hierarchy relation [6][7].

To show the dynamic nature of the RBAC model, we can use the UML class diagram shown in Figure 1 [11]. The UML representation of security models is an extension on design patterns (referred to as security patterns). A design pattern is s a formal way to document a solution to a design problem in a particular field of expertise. In the area of software engineering, a design pattern is a reusable solution to a repeatable problem within a given context in software design. It is commonly adapted to the object-oriented paradigm [11]. Security patterns are extensions of design patterns in the security field. In this figure, *User* refers to registered users (persons or software agents). *Role* refers to predefined roles. Roles are assigned according to the security policy. Users are assigned as members of roles. *ProtectedObject* defines objects that should be protected according to the security policy.

Right defines the access type that is granted to a role. Sometimes a right is also called *permission*. *Group* defines a group of users that can be assigned the same role. A *Composite Role* defines roles that have permissions from several roles. A *Simple Role* defines a role that is not composed of others. A composite refers to an association that models a whole-part relationship. Through composite roles, more complex roles can be structured and defined in an object-oriented context. *Session* defines the way to use a role. Each session belongs to only one user. During runtime, each user can establish a session and activate several roles where they are members. The permission of a user is the union of all permission belonging to the activated roles. *AdminRole* defines an administrator's role. *AdminRight* is the permissions owned specifically by an administrator [11][12].

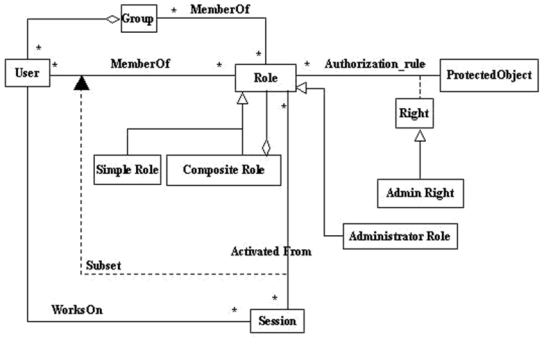


Figure1. UML representation of the RBAC model [4]

RBAC models can be used to implement three important security principles: least privilege, separation of duties, and data abstraction [8]. The principle of least privilege means that roles only have the minimum access rights required in a time period. The second principle, separation of duty, means access permissions are separated among roles to facilitate the management of different security levels. Data abstraction means abstract data access permissions instead of a simple categorization (for example, read, write, and execute), can be established in this model [8].

RBAC is different from Discretionary Access Control (DAC). Access control in the RBAC model is based on the job functions of users. Users have no right to pass their permissions to other users at their discretion, as that in the DAC model. RBAC can be viewed as a form of Mandatory Access Control (MAC) without multilevel security requirements [6]. The RBAC model has been applied in a variety of real-world management systems. Other than its traditional use in database systems, Bacon et al. at the University of Cambridge applied an RBAC system called OASIS to achieve interoperation in an open and distributed environment [9]. Ferraiolo et al. at the National Institute of Standards and Technology (NIST) applied an enhanced RBAC model to network Web servers [6]. There are also other systems developed for the banking community, network applications, and high-performance cluster computing environments [12]. It has been identified in the literature that database operations such as transaction processing services, may be best served by RBAC models, possibly complemented by data-centric policy implemented in underlying databases [5]. In this paper, we mainly follow the mechanism described in [12]. The RBAC model mechanism has proven itself to be a useful access control method. Because cloud computing has close connections with other fields such as cluster/high performance computing and network applications, it is natural to extend the model in cloud computing.

IV. USING RBAC IN THE CLOUD

Each access control model should be tailored to its application environment based on the context and scope of protection required by the environment. It is straightforward to extend RBAC from traditional fields (for example, database systems and network applications) to cloud computing environments. To successfully employ the RBAC model, the first task is to identify corresponding entities defined in Figure 1. Due to the difference in nature and scope, it is important to separate the identification among each of the three services (SaaS, PaaS, and IaaS) in the SPI model.

A. Users/agents

The RBAC model provides an authorization service for users. In this context, "users" refers to human beings or software/hardware/network components inside the cloud. Under the coverage of security policies, users of the secure cloud-computing environment can be classified according to their job functions. In Software as a Service (SaaS), users could include individual consumer, corporate consumer, and web services that request resources. Platform as a Service (PaaS) is a newer service and haven't been fully deployed yet, and its security policies are yet to discover. Users in PaaS could include all users identified in

SaaS. A PaaS customer would need administrative access to specify policy for their application running in the cloud, but they should not affect other policy domains. As such, Users in PaaS could be defined in a finer granularity. For example, consumers have the privilege to define their own users in a policy domain. Infrastructure as a Service (IaaS) is the newest service provided by the cloud. In IaaS, users may include those defined in PaaS, although web services are less likely. In most cases, an IaaS customer will receive access to a virtual machine and be responsible for configuring all aspects of the system. As such consumers have the most control over the definition of users/agents when it is compared to other two services, and the users could be defined at a finer level.

B. Roles

Roles are categorized according to their job functions. Examples include database operator and system administrator. The definition of roles is central to the RBAC model and is highly related to application environments. In cloud computing, roles can be defined at a high level. They may include consumer, tenant, and service providers. Using a finer granularity, these roles can be further split in several ways. The first is according to the specific accesses such as program access, data access, Internet access, and server access. For example, a role may be defined as “consumer for Internet access”. The second way is by following the three service models – SaaS, PaaS, and IaaS. For example, a role may be defined as “tenant for corporate data access.” The third way to split is by following cloud architectures defined by vendors [13]. Along this line the roles can be further split at finer levels and the entire structure of roles finally form a hierarchy, in which roles could inherit permissions and functions defined in a parent role. Inheritance is one distinct characteristic in RBAC and seems to fit well in this situation. For example, at a coarse granularity level, in Amazon’s Elastic Compute Cloud (EC2), roles might be defined as EC2 instance 1 administrator, EC2 instance 1 operator, etc. In IBM’s Blue Cloud, roles could be defined as DB2 administrator, DB2 database operator, Tivoli provisioning manager, Tivoli monitoring agent, virtual machine instance 1 user, etc.

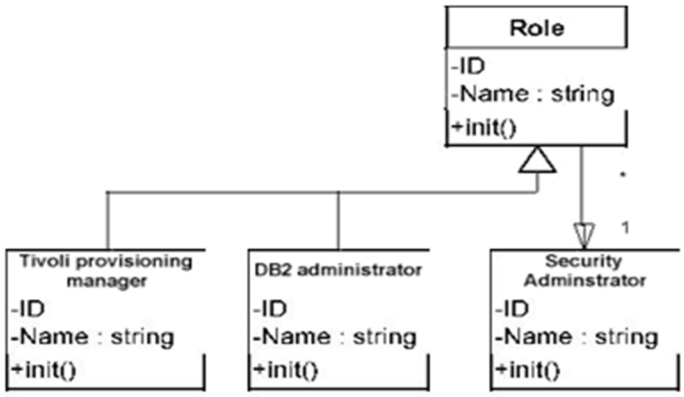


Figure 2. Example Admin roles in IBM Blue Cloud

Figure 2 shows an example how the administrator roles could be organized in IBM Blue Cloud. Similar break-downs can be performed for daily operator roles. The security administrator is a single role that oversees all security functionalities in the system. It is also a composite role that inherits privileges from “Tivoli provisioning manager” and “DB2 administrator”. When more roles are designed, they can be organized into hierarchies to facilitate their management. How the roles are designed and managed greatly affects the scalability of RBAC in the cloud.

It can be seen that, in these given examples, the roles in the cloud platforms are relatively easy to identify because their job functions can be clearly separated, and users can be assigned accordingly. These roles are highly specific not only because of their service providers, but also be tailored to their business requirements. These are the situations in which RBAC is the one of the best fit access control model for cloud computing.

In the IaaS model or a private cloud, since consumers have full control over the virtualized environment, the roles can be defined according to their specific needs. For example, for a banking system, traditional roles such as “cashier” or “accountant” could be defined. In the system there could also be roles adapted to the cloud environment, for example, a role can be named as “cloud security administrator.” It is quite possible that a role with the same name may incorporate completely different access privileges in different environments.

C. Permissions

Permissions are defined according to job functions of roles. In a secure cloud computing environment they can generally be classified into the following groups: data access permissions, program access permissions, and service access permissions. Access permissions should be configured according to the environmental requirements. For example, program permissions include read, write, execute, create, and delete. Service permission may include bandwidth utilization, computational power utilization, etc. Similar to the discussion presented in the previous section, the permissions can be further split in finer granularities and in hierarchies.

D. Protected Objects

In the object-oriented domain, protected objects represent resources within the clouds. There are in general three groups of protected objects, data, program, and service. These objects correspond to permissions mentioned earlier. If permissions are defined at a finer granularity, the project objects should be identified accordingly. For example, in Amazon's EC2, objects can be defined as EC2 instance 1, EC2 instance 2, etc. There are three EC2 instances (protected domains) in this example.

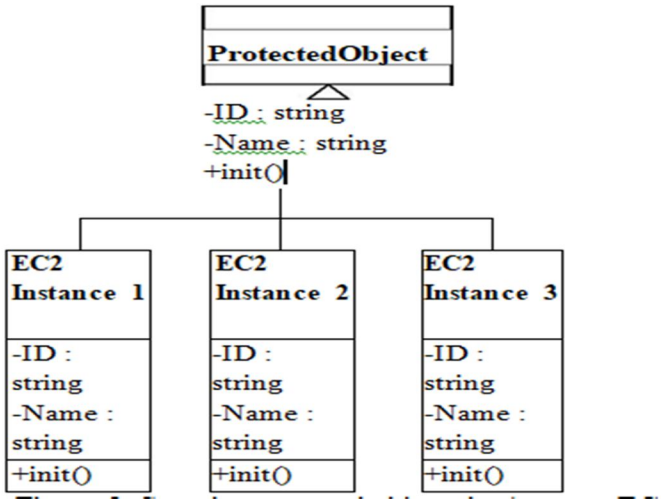


Figure 3. Sample protected objects in Amazon EC2

E. Sessions

Sessions are used in RBAC models to define ways to use multiple roles. The use of sessions in cloud computing is almost the same as that in traditional application environments. The process can be briefly shown in Figure 4. In the figure, attribute *ActiveRoleList* provides links to current active roles for a user. Attribute *RoleList* provides links to all roles where the current user is a member. *ActiveRoleList* is a subset of *RoleList*. We can check whether a user is a member of a role by invoking *checkUserRole*(). To activate or deactivate roles where a user is a member, we can invoke *roleActivation*() and *roleDeactivation*() routines. This mechanism can be used to enforce role exclusion or inclusion at execution time. A session can be established using *SessionEstablishment*(). The session can be disabled using *SessionRevocation*(). By using sessions, one can manage dynamic role inclusion/exclusion owned by each user. Permissions can be disabled and enabled through the use of sessions. It should be noted that according to the definition by Ferraiolo et al. [14], "each session is a mapping between a user and an activated subset of roles that are assigned to the user". A session is not necessarily limited to one user or one role. In cluster-computing environments, parallel assignments of multiple sessions to one role can be easily achieved. Parallel implementations can also be done for user-to-role, object-to-role, permission-to-role, permission-to-operation and permission-to-object assignments.

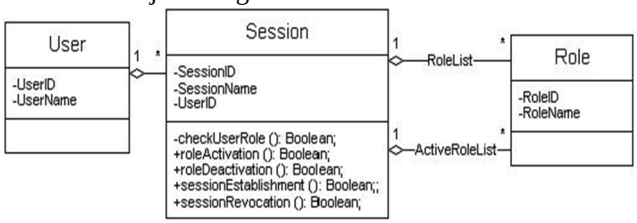


Figure 4. Relationship among User, Role and Session [12] In addition to the identification of different entities in

RBAC, there are also some management issues should be considered. One issue is who should manage the assignment of the roles. In cloud computing, there needs collaboration between cloud vendors and consumers so that a fine-grained definition on roles can be achieved. This is especially true for the SaaS and PaaS model. Another issue is how to handle dynamic role management. In RBAC, an end user may change its role from to another. The revocation of access to data and services should be carefully evaluated so that the change does not grant unauthorized access in the cloud. The change may have cascading effect when a multi-party trust chain is formed – this may require coordination of multiple cloud providers. In addition the use of RBAC should be closely administered and monitored so that policy and compliance (for example, Health Insurance Portability and Accountability Act or HIPPA – a US law that address the security and privacy of health data) could be enforced.

V. DISCUSSION

It is obvious that RBAC has great potentials to be used in cloud computing in various settings. However due to the lack of standards and commonly-agreed best practices in the field, the utilization of RBAC is vendor specific. It appears that the SaaS is by far the most mature service models in cloud computing. The use of RBAC can be greatly enhanced by the promotion of the cloud vendors, and could glean experiences when the model is expanded to other service models. Large- corporate consumers will need to invest in designing a role model that maps user roles to their internal business functions in order to effectively manage the RBAC model. Some existing experiences with RBAC could be applied during the process.

Consumers of RBAC should note that even with an industry standard, the vendor and consumer still need to agree on the names and semantics used within the cloud service. For example, a “security administrator” could mean different roles in the corporate setting and in a cloud setting. An approach is need so that corporate roles can be separated from roles defined by cloud providers [5].

It should also be noted that RBAC may not fit into all security domains in cloud computing. Its strength appears to lie in areas in which job functions of different roles can be clearly defined and separated, and roles have an object-oriented nature and could form a hierarchy. In many situations other access control model may be employed [5]. For example, the MAC model is necessary to make access control decisions based upon the classification of assets or information. Web service access to resources in the cloud is generally best supported by a DAC (i.e., Access Control List or ACL) model. In some situations, cloud environments may impose quota-based or task-based access control, depending on the service agreement between consumers and cloud providers. There are also industry standard employed in cloud computing. One example is the eXtensible Access Control Markup Language (XACML), which aims “to define a core schema and corresponding namespace for the expression of authorization policies in XML against objects that are themselves identified in XML.” [15]

VI. CONCLUSIONS AND FUTURE RESEARCH

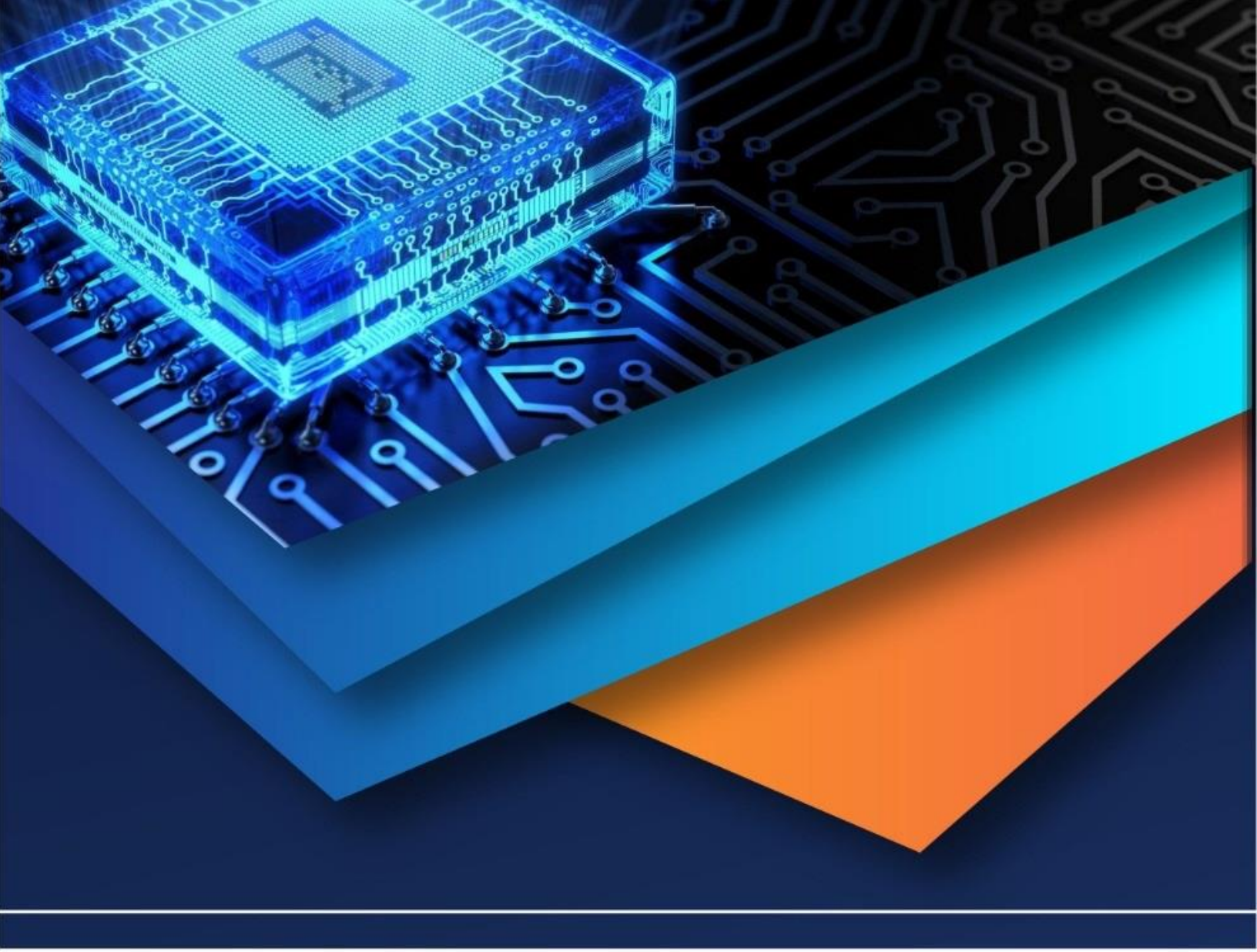
This paper presents a brief discussion on cloud computing and its related security risks, with a focus on access control and RBAC. RBAC models can be used to implement three important security principles: least privilege, separation of duties, and data abstraction. Through the discussion it can be seen that RBAC has great potentials to be employed in cloud computing. However due to the lack of standard and best practices in the field, and also due to the immaturity of the cloud computing itself, huge efforts are still necessary to promote RBAC and make access control effective.

Future research efforts should include a more formal identification process for different entities in RBAC in the context of cloud computing, industrial standards and best practices of using RBAC in their clouds, and large scale experiments to show the promise of RBAC. It appears that the success or RBAC will be in pace with the maturity of the cloud computing platform.

REFERENCES

[1] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner. A break in the clouds: towards a cloud definition. ACM SIGCOMM
[2] Computer Communication Review, Vol. 39 Issue 1, January 2009, pp. 50-55.
[3] National Institute of Science and Technology. The NIST Definition of Cloud Computing (Draft). Retrieved 24 July 2011 from http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf.
[4] R. L. Krutz, and R. D. Vines. Cloud Security: A Comprehensive Guide to Secure Cloud Computing. Wiley Publishing, Inc, 2010.
[5] M. Jensen, J. Schwenk, N. Gruschka, and L. L. Iacono, On Technical Security Issues in Cloud Computing, 2009 IEEE International Conference on Cloud Computing, pp. 109-116.
[6] Cloud Security Alliance. Domain 12: Guidance for Identity & Access Management V2.1, Retrieved 28 July 2011 from https://cloudsecurityalliance.org/wp-content/uploads/2011/07/csaguide-dom12-v2.10.pdf.
[7] D. Ferraiolo and D.R. Kuhn, “Role Based Access Control,” Proceedings of the 15th Natl. Computer Security Conference, 1992, Baltimore, MD, October 1992, pp. 554-563.

[8] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn and R. Chadramouli, "Proposed NIST Standard for Role-Based Access Control," ACM Transactions Information and System Security, Vol. 4, No. 3, August 2001, pp. 224-274.
[9] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-Based Access Control Models," IEEE Computer, Vol. 29, No. 2, February 1996, pp. 38-47.
[10] J. Bacon, K. Moody and W. Yao, "Access Control and Trust in the Use of Widely Distributed Services," Proceedings: Middleware 2001, Heidelberg, Germany, November 2001, IEEE, pp. 300-315.
[11] M. E. Shin and G. Ahn, "UML-Based Representation of Role-Based Access Control," Proceedings: IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'00), Gaithersburg, Maryland, 2000, IEEE Computer Society, pp. 195-200.
[12] E. B. Fernandez and R. Pan, "A Pattern Language for Security Models," Conference on Pattern Languages of Programs (PLoP) 2001, Retrieved 24 July 2011 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.5898&r ep=rep1&type=pdf.
[13] W. Li, E. Allen. An Access Control Model for Secure Cluster-Computing Environments. Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS'05). Big Island, HI, January 3-6, 2005, p. 309.
[14] G. Boss, P. Malladi, D. Quan, L. Legregni, and H. Hall. Cloud computing. IBM White Paper, 2007, Retrieved 24 July 2011 from http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/ Cloud_computing_wp_final_8Oct.pdf.
[15] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chadramouli, "Proposed NIST Standard for Role-Based Access Control," ACM Transactions Information and System Security, Vol. 4, No. 3, August 2001, pp. 224-274.
[16] Extensible Access Control Markup Language (XACML), OASIS, Retrieved 24 July 2011 from http://xml.coverpages.org/xacml.html.
[17] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", Tech. Report UCB/EECS-2009-28, EECS Dept., University of Califonia, Berkeley, 2009.
[18] W. Li, S. Li, and H. Wan, "An Access Control Mechanism in Cloud Computing Environments," Proceedings: 2011 National Annual High Performance Computing of China (HPC China 2011). October 2011, Jinan, China, China Computer Federation.
[19] Y.Chen, V.Paxson, and R.H.Katz, What s new about Cloud Computing security , Tech. Report UCB/EECS-2010-5, EECS Dept., University of Califonia, Berkeley, 2010



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY