Access Control in Multi Tenant and Diverse Cloud Computing Environment

Anjali Chhabra*
Vanita Kareer**
D. Santhadevi***

Abstract

Cloud computing is a buzzword in field of information technology today. Despite a long list of advantages, the organizations are not switching to cloud at the pace expected due to a single major drawback, which is security. Researchers are putting their head and heart to analyze and solve security challenges faced by cloud computing environment. Access control to sensitive data is one such major challenge before cloud computing proponents.

Wikipedia defines access control as selective restriction of access to a place or other resource.The paper describes uniquefeatures ofcloud computing which make access control environment of cloud computing different from standard enterprise environment.Author tries to compare various access control techniques available and their potential applicability to cloud computing environment.

Keywords: Cloud Computing, Security, Discretionary Access Control, Mandatory Access Control, Role Based Access Control, Attribute based access control.

Introduction

Cloud Computing represents one of the most significant shifts in information technology many of us arelikely to see in our lifetimes. Reaching the point where computing functions as a utility has great potential, promising innovations we cannot yet imagine.

Customers are both excited and nervous at the prospects of Cloud Computing. They are excited by theopportunities to reduce capital costs. They are excited for a chance to divest them of infrastructure management, and focus on core competencies. Most of all, they are excited by the agility offered by theon-demand provisioning of computing and the ability to align information technology with businessstrategies and needs more readily. However, customers are also very concerned about the risks of CloudComputing if not properly secured, and the loss of direct control over systems for which they areNone the less accountable.

Access control i.e. giving the right things to right person is one of the major challenges faced by cloud computing proponents. Access control techniques currently available are effective for enterprise-based solutions where we have a known set of users and known set of services. These techniques are not very effective in large open distributed systems like cloud computing and grid computing where we have dynamic relationship between users and resources. Section 2 throws some light on related concepts. Section 3 discusses the features of cloud computing which make traditional access control methods ineffective for cloud. Section 4 gives a brief description of access control methods currently available. Section 5 discusses Attribute based Access Control (ABAC) used in cloud computing in detail.

Related Concepts

The section discusses the application field and variety of techniques under the scope of this field.

Anjali Chhabra*
IINTM, GGSIPU

Vanita Kareer**
IINTM, GGSIPU

D. Santhadevi***
IINTM, GGSIPU

Cloud Computing

Cloud computing is used to describe a variety of computing concepts that involve a large number of computers connected through a real-time communication network such as the Internet. In science, cloud computing is a synonym for distributed computing over a network, and means the ability to run a program or application on many connected computers at the same time.

Multi Tenancy

Multitenancy refers to a principle in software architecture where a single instance of the software runs on a server, serving multiple client-organizations (tenants). Multitenancy contrasts with multi-instance architectures where separate software instances (or hardware systems) operate on behalf of different client organizations. With a multitenant architecture, a software application is designed to virtually partition its data and configuration, and each client organization works with a customized virtual application.

Access Control Challenges in Cloud Computing

In Cloud Computing, cloud acts as a third party which manages customers data and resources as compare to traditional computing environment.In this section we review the specific features of cloud computing which make its access control environment different from traditional enterprise environment:

Multi-tenancy

Multi-tenancy introduces new requirements to access control as intra-cloud communication (*i.e.,* provider-tenant and tenant-tenant) is becoming more popular. For example, Amazon provides their tenants with services such as SimpleDB and Simple Queue Service (SQS); there are also tenants that provide services to other tenants, *e.g.,* mapreduce++ and desktop services. The intra-cloud communication is likely to require new types of access control policies such as fairsharing between tenants, and rate-limiting tenants.

Network-diversity

Network-diversityis another new challenge for access control. The architecture of data centers has evolved significantly from that of the traditional enterprises and is currently in flux, with much new architecture. These new architectures typically employ multiple paths and require specific routing algorithms and address assignments.Therefore, they severely limit the applicability of current mechanisms such as VLANs and firewalls. Furthermore, today's clouds house tens of thousands of physical machines, and even more virtual machines that are constantly added and removed. Current access control mechanisms were not designed to handle such scale and churn. For example, firewalls have problems scaling to large numbers of entries and coordinating access control across multiple firewalls is complex, while VLANs do not support dynamic configuration, are limited in scalability and complex to setup and configure. More generally, our observation is that as clouds scale to large numbers of users, they will face many of the problems traditionally associated with the public Internet, including DoS attacks between cloud tenants. Such attacks are known to be very difficult to tackle but are not typically the concern of (internal) enterprise access control mechanisms, diversity in cloud network architectures the large scale and the high dynamism of cloud infrastructure.

Based on these features, we realize that Access Control solution in cloud computing should have following characteristics:

- *Flexibility* in providing support for policies in multi-tenant environments such as tenant isolation, fair-sharing, and rate limitingpolicies
- *Network-independence* in decoupling access control from the network topology, routing and addressing.
- *Scalability* in handling hundreds of thousands of machines and users.

Access Control Techniques

Traditional access control techniques is broadly classified in to three types:

- Discretionary Access Control (DAC)
  A system that uses discretionary access control allows the owner of the resource to specify which subjects can access which resources.Access control is at the discretion of the owner.
- Mandatory Access Control (MAC)
  Access control is based on a security labeling system. Users have security clearances and resources have security labels that contain data classifications.This model is used in environments where information classification and confidentiality is very important (e.g., the military).
- Role Based Access Control (RBAC)
  Role Based Access Control (RBAC) uses a centrally administered set of controls to determine how subjects and objects interact. Is the best system for an organization that has high turnover?

All these models areknown as identity based access control models. In all these, we assume that users and resources can be uniquely identified by their names or by their roles. These techniques fit well in a static distributed system with a known set of users and services.

In cloud computing, the relationship between users and resources is dynamic. Here the users and resources may belong to different domains So identifying users by name or job role is not feasible. It gave birth to Attribute Based Access Control (ABAC) techniques.

Attribute-Based Access Control

Attribute-based access control (ABAC) defines a new access control paradigm whereby access rights are granted to users based on policies/rules which combine attributes together. Rules use attributes as building blocks. Attributes are sets of labels or properties that can be used to describe all the entities that must be considered for authorization purposes. Attributes can be compared to static values or to one another e.g. “Role=Manager”, thus enabling relation-based access control.

It helps achieve efficient regulatory compliance, effective cloud services, reduced time-to-market for new applications, and a top-down approach to governance through transparency in policy enforcement.

There can be different types of attributes

- Subject Attributes
  - The subject is “who is demanding access to an information asset?”. It can be a user, application or a process.
  - Subject attributes are attributes of subject that defines its identity and characteristics.
  - E.g. name, job title,identifier
- Resource Attributes
  - Resource identifies the information asset or object impacted by the action.
  - Resource attributes are characteristics associated with a resource (web service, system function, or data)
  - E.g. Dublin Core metadata elements
- Environment Attributes
  - Describes the operational, technical, or situational environment or context in which the information access occurs
  - E.g. current date time, current threat level, network security classification

ABAC Policy Formulation

ABAC can combine any number and type of attributes to define policies. The figure on the next page (figure 1) summarizes the ABAC policy formulation.

Mathematical Model

1. *Let Sub*, *Res*, and *Env* are subjects, resources, and environments respectively;

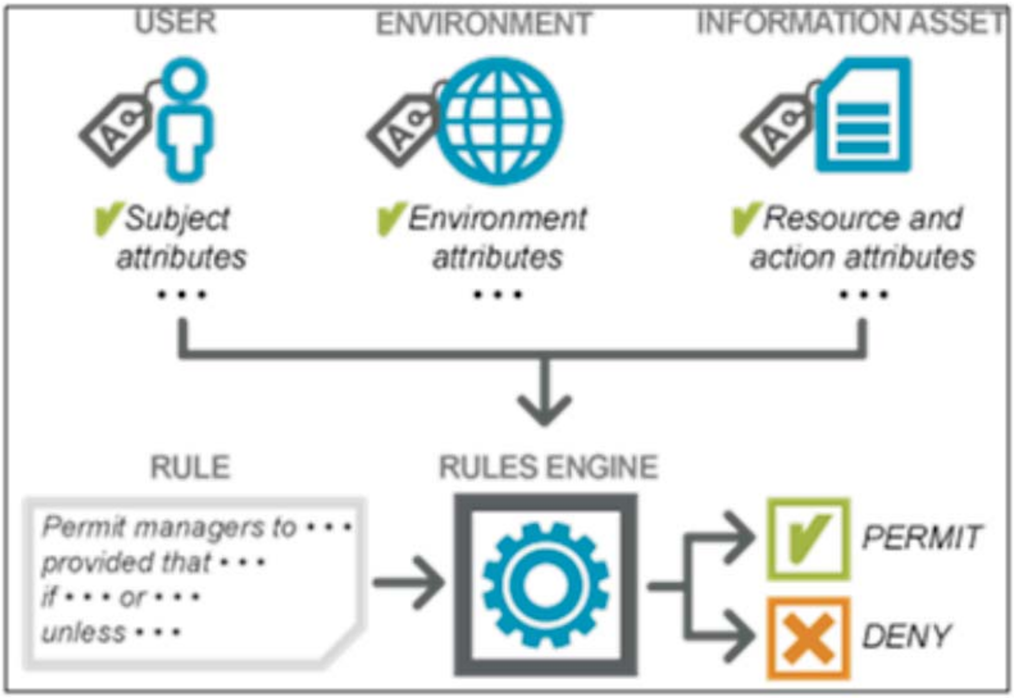

Fig. 1: ABAC Policy Formulation

2. $SubA_k$ ($1 \leq k \leq K$), $ResA_m$ ($1 \leq m \leq M$), and $EnvA_n$ ($1 \leq n \leq N$) are the pre-defined attributes for subjects, resources, and environments, respectively;
3. *ATTR(sub)*, *ATTR(res)*, and *ATTR(env)* are attribute assignment relations for subject *s*, resource *r*, and environment *e*, respectively:

   $ATTR(sub) \subseteq SubA1 \ X \ SubA2 \ X \ SubA3$

   $ATTR(res) \subseteq ResA1 \ X \ ResA2 \ X \ ResA3$

   $ATTR(env) \subseteq EnvA1 \ X \ EnvA2 \ X \ EnvA3$
4. We also use the function notation for the value assignment of individual attributes. For example:

   Role (sub) = “Service Consumer”

   ServiceOwner(res) = “XYZ,Inc”

   Current Date (env) =“21-02-2014"
5. In the most general form, a *Policy Rule* that decides on whether a subject *sub* can access a resource *res* in a particular environment *env*, is a Boolean function of *sub*, *res*, and *env*’s attributes:

   Rule Y: can_access (sub, res, env) ← f (ATTR(Sub), ATTR(res), ATTR(env))

The access control decision process in essence amounts to the evaluation of applicable policy rules in the policy store.

Comparison with RBAC Models

- Inherent limitation in RBAC is the single dimension of roles
  – Finer-grained access control policies often involve multiple subject and object attributes
  – As more attributes are involved, number of roles and permissions needed to encode these attributes will grow exponentially, thereby making User Assignments and Permission Assignments difficult to manage
- Various research work has tried to extend the basic RBAC model, however most are constrained by the inherent limitations of RBAC

  - Rule-based RBAC [8],
  - Inclusion of subject-resource relationship [4]
  - Use-condition policies specified by stakeholders [3]
- RBAC usually needs centralized management of user-to-role and permission-to-role assignments
  - Not well suited for a highly distributed environment
  - Even more difficult when subject and resource belong to different security domains.
- RBAC doesn't consider environment attributes explicitly
  - E.g., continuing with the previous example, suppose an additional requirement states "Regular customers in general may not watch new releases, but may be allowed in promotional periods"
  - In ABAC, a new rule can be easily added, involving an environment attribute *CurrentDate(env)*
- RBAC doesn't handle MAC
  - In ABAC, security labels can be treated naturally as attributes

Futuristic Research

The Current ABAC policy and architecture models, though very powerful, only focus on authorization of requests from information consumers to providers. The end-to-end security architecture requires more than just the access control model. To utilize ABAC to its full potential, we need a systematic methodology around how attributes are managed throughout their life cycle:

- Attribute definition and "provisioning";
- Cryptographic mechanisms to "bind" attributes to subjects and objects they describe;
- Discovery mechanisms for attribute definitions and attribute assignments;
- A "feedback loop" through which attribute usage can be monitored and audited

Future research should focus on utilizing ABAC model in all stages of lifecycle of attribute.

Conclusion

Access control is one of important security issue for any shared system like cloud. The paper analyses various access control techniques and their suitability to cloud computing environment.It discussed the features, advantages and characteristics of Attribute based Access Control (ABAC) technique in detail. It also discusses future area of research in strengthening ABAC.

References

1. Al-Kahtani, M., and Sandhu, R. A model for attribute-based user-role assignment. Proceedings of the18th Annual Computer Security Applications Conference (Las Vegas NV, December 2002).
2. E. Yuan and J. Tong. 2005. Attribute Based AccessControl (ABAS) for web Services. Proceeding ofIEEE Conference on Web Service.
3. J. Barkley, K. Beznosov, and J. Uppal, "Supporting Relationships In Access Control Using Role Based Access Control,*" In proceedings of the fourth ACM workshop on role-based access control on Role-based access control*, Fairfax, VA USA, October 28 - 29, 1999, pp. 55-65.
4. L. Wang, D. Wijesekera and S. Jajodia. 2004. A logicbased framework for attribute based access control.Proceeding of ACM workshop on formal methods inSecurity Engineering. pp. 45-55, ACM press.
5. R. Alfteri *et al.* 2003. An Authorization System forVirtual Organizations. Proceeding of 1st Europeanacross Grids conference.
6. Thompson, M., Johnston, W., Mudumbai, S., Hoo, G., Jackson, K., and Essiari, A. 1999. Certiûcatebased access control for widely distributed resources. In Proceedings of the 8th Usenix SecuritySymposium, Aug. 1999.
7. T. Barton *et al.* 2006. Identity Federation andAttribute Based Authorization through the GlobusToolkit, Shibboleth, Gridshib and My Proxy.Proceeding of 5th Annual PKI (R and D) workshop.
8. V. Welch *et al.* 2005. Attributes, Anonymity andAccess: Shibboleth and Globus Integration toFacilaitate Grid collaboration. Proceeding of 4thannual PKI (R and D) workshop.