Policy-based Framework for Access Control in Cloud Computing

Subhash Chandra Patel*  
Department of Computer Engineering  
Indian Institute of Technology, Banaras Hindu University  
Varanasi, India  
scpatel.rs.cse@itbhu.ac.in

Lokendra Singh Umrao  
Department of Computer Engineering  
Indian Institute of Technology, Banaras Hindu University  
Varanasi, India  
Lokendra.manit@gmail.com

Dr. Ravi Shankar Singh  
Department of Computer Engineering  
Indian Institute of Technology, Banaras Hindu University  
Varanasi, India  
ravi.cse@itbhu.ac.in

Abstract— One of the most challenging issue in cloud computing is access control and data security because User of the cloud outsourced their sensitive data and information to cloud provider’s servers which is not within the same trusted domain as data owner. Cloud Computing is a large-scale distributed computing paradigm which is based on Web Services over the Internet. Internet has many inherent security defects because of its openness. It has many other attacks and threats. In this paper we propose a Framework and the components of policy base access control mechanism that is suitable for access control.

Keywords- cloud computing; internet; risks and security

I. INTRODUCTION

Cloud Computing is a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet[1]. Cloud computing is only a different way to deliver computing resources, rather than a new technology, it has a revolution in the way organizations provide information and service. changes the way we think about technology[2]. Cloud is a computing model providing web-based software, middleware and computing resources on demand, in which services, resources, and applications are provided on metered basis over the Internet. It is an Internet-based computing solution where shared resources/services are provided like electricity distributed on the electrical grid. The cloud computing provides basically three type of services. Software as a service(SaaS) in which the cloud service provider provides applications and software over a network. Google Docs[3], Facebook[4], Gmail[5], Yahoo[5] are the example of SaaS. Platform as a Service(PaaS) provides application or development platform in which user can create their own application that will run on the cloud, example of PaaS are Microsoft’s Azure[6], Google’s App Engine(App Engine)[7], Yahoo Pig[8] and third type of cloud service is Infrastructure as a service (IaaS), the whole cloud infrastructure, including servers, routers, hardware based load-balancing, firewalls, storage and other network equipment is provided by the IaaS provider i.e. Amazon S3[9], Amazon EC2[10]. The cloud computing can be deployed as public cloud ,private cloud, hybrid cloud and community cloud. Public clouds are publicly available and can serve multiple tenants, examples of public Cloud: Google App Engine, Microsoft Windows Azure[11], IBM Smart Cloud[12] and Amazon EC2 while private cloud is typically a tailored environment with dedicated virtualized resources for particular organization examples of private clouds are Eucalyptus[13], Ubuntu Enterprise Cloud – UEC[14], Amazon VPC (Virtual Private Cloud)[15], VMware Cloud Infrastructure Suite[16], Microsoft ECI data center[17]. Similarly , community cloud is tailored for a particular group of customers Google Apps for Government, Microsoft Government Community Cloud are the example of community cloud. Hybrid cloud is composed of multiple clouds like Windows Azure (capable of Hybrid Cloud), VMware vCloud (Hybrid Cloud Services).

II. LITERATURE REVIEW

To overcome from security during access the services or applications many researchers has presented effective mechanisms and methods for access control i.e. Vibhaj et al [18] proposed an innovative mechanism to achieve flexible access control during co-ordinate resource sharing in enterprise grid environment. The mechanism proposes a new model for users, resources and their relationships which define co-ordination and access control. The solution targets to satisfy the demands usually found in enterprise environments with respect to controlled sharing of resources.

In addition Ravi Shankar et al [19] presented a Key-Chain-Web access control mechanism which expands the usability of the mechanism in different scenarios. The Key-Chain-Web mechanism is based on three entities Users (keys), Resources (chains) and Relationships (webs). In addition, chain members representing relationship among users and...
resources, particularly in case of a collaborative environment where sharing may be achieved through members. [19].

Sangroya et al [20] describe the security mechanisms that are used by major cloud service providers and propose a risk analysis framework that can be used to analyze the data security risks before putting confidential data into a cloud computing environment. The research illustrates the need for new authorization services and models suitable for the cloud computing to provide access control in a distributed, multi-tenant, federated environment.

Eucalyptus [21] is an open-source cloud computing system. This system provides an authorization system to control the execution of the virtual machines (VMs) that compose the cloud infrastructure. This authorization system protects running VMs to ensure that only administrators and the owner of these VMs are able to access them.

Berger et al [22] promote an authorization model based on both role-based access control and security labels. The security labels represent colors and these colors are used to control the access to data sharing, to virtual machines, network resources, etc. Zhang et al [23] promote an authorization scheme for the cloud computing based on session keys shared between the cloud services and the users in order to be able to access to the protected resources.

Danwei et al [24] propose an authorization architecture based on UCON suitable for the cloud computing. The main contribution of this proposal is the inclusion of a negotiation model in the authorization architecture to enhance flexibility of the access control on the cloud services. When the access requested mismatches with access rules, it allows users to get a second access choice through negotiation in certain circumstances, instead of refusing access directly.

Hu et al [26] presented a new semantic access control policy language (SACPL) to describe access control policies in cloud computing environments. To tackle the interoperability issue of distributed access control policies, they propose an ontology-based access control. Interoperability between different schemes and models are focused in his research. Yu et al [28] and Di Capitani et al [27] presented a novel solutions for the enforcement of access control and the management of its evolution in cloud environments. The former applies selective encryption as a method to enforce authorizations whereas the latter enforces access policies based on data attributes. These researches can be used to complement the authorization model proposed by combining the enforcement of the access control information on the untrusted cloud servers.

Authorization deals with the verification of an action that an entity can perform after authentication is performed successfully. In a grid, resource owners will require the ability to grant or deny access based on identity, membership of groups or virtual organizations, and other dynamic considerations. Thus policies must be established that determine the capabilities of allowed actions. Thus the issue of “who has got the access” is authorization and “under what condition” is access control Chakrabarti et al [29]. Thus term authorization refers to a statement or combination of statements that prove a subject’s right to access a resource, or to describe the act of validating that a subject has such a right.

III. PROBLEM STATEMENT

Cloud service is based on Web Services [30], and Web Services are based on Internet. Internet has many inherent security flaws because of its openness, and it also has many other attacks and threats. Therefore, cloud services will face a wide range of security issues. The most challenging ongoing research in cloud computing are access control and data security, because of users outsourcing their sensitive data to cloud providers. Existing solutions that use pure cryptographic techniques to mitigate these security and access control problems suffer from heavy computational overhead on the data owner as well as the cloud service provider for key distribution and management. To overcome from the above problem we proposed a policy based framework for access control in cloud computing environment.

IV. ACCESS CONTROL COMPONENTS FOR CLOUD COMPUTING

A system that controls access to services/resources made by cloud users based on authentication, authorization attributes of subjects, attributes of objects/resource as well as system attributes which conforms to policies. Each entity i.e. subject and object/resource is identified by its attribute. Subject’s attribute are divided in two categories mutable and immutable[31]. Mutable attributes are those which may change during access operation e.g. location, usage status etc. Immutable attributes are those which are independent of access operation e.g. identity of user. The novel feature of the framework is that each domain has an attribute authority which acts a filter which filters the rights of the requesting user for the target site. Since there is no central authority which provides attributes to user and also there is no need to have each user to have account on the target resource.

In the framework following elements are used.

- **Policy enforcement point (PEP)**: The system entity that performs access control, by making decision requests and enforcing authorization decisions.
- **Policy-combining algorithm**: The procedure for combining the decision and obligations from multiple policies.
- **Policy decision point (PDP)**: The system entity that evaluates applicable policy and renders an authorization decision.
- **Policy information point (PIP)**: The system entity that acts as a source of attribute values.
- **Policy administration point (PAP)**: The system entity that creates a policy or policy set.
• **Subject**: Subject is an entity that wants to access services/resources. It can be a user, service or any other entity on behalf of user/service.

• **Service/Resource**: Service/resource is the object which is accessed by user e.g. CPU, Storage device, data, instruments etc.

• **Service Policy**: It is a set of rules associated with service/resource. Subject must conform to service policy in order to access that service.

• **Policy database**: It is policy store.

• **Filter/Privilege Authority**: This component filters rights of the subject for the domain to which it is making request. During this process every domain informs other domain about rights that this domain gives to other domain. So every domain maintains a database about rights which other domains have given to it which acts as filter.

• **Mutable Attribute Repository**: This is central repository which contains subject’s mutable attribute e.g. usage quota (computation time, storage space) etc and system attributes e.g. time, location etc.

• **Authorization Assertion**: Authorization Assertion which state whether user has access to the target domain.

• **Rule-combining algorithm**: The procedure for combining decisions from multiple rules.

V. **METHOD FOR ACCESS CONTROL OF SERVICES AND APPLICATIONS**

There are mainly two steps for accessing resources i.e. obtaining credentials and Authorization decision as follow

1. **Getting credentials**: Requesting subject first finds out authentication requirement of resource/service (X.509, Kerberos ticket etc.) [31]. Each domain maintain database of user’s credentials.

   There are several steps to obtaining the credential (user name / password) as bellow

   a) Access the required credential from the database.

   b) The root authority has right to issuing the certificate to individual domain. (There is one root authority on which all the participating domains agree)

   c) The domain issue certificate to users(subject)

   d) After obtaining authentication credentials/identity certificate the subject contacts filter/privilege authority in its home domain for obtaining authorization assertion.

2. **Reaching Authorization Decision**: The authorization decision are made on the basis of attribute values [31], there are several steps to authorization decision as bellow

   a) Subject (request for service/resource) sends request to the target site with authentication certificate (which authenticate subjects identity). The request also include persistent attributes (i.e. role membership).

   b) The request is intercepted by PEP (Policy Enforcement Point) [32] at the target site, PEP forwards request to PDP (Policy Decision Point) [32] for making access decision.

   c) PDP retrieves corresponding policy from policy store, which matches with subject’s resources and action values. If policy requires some mutable attributes of user then it fetches from mutable repository and object’s attributes from Resource monitor.

   d) PDP evaluates policy and decision upon request and responds to PEP which enforce decision.

The working of PEP can be written in XACML [33]. The standard defines a declarative access control policy language implemented in XML [34] and a processing model describing how to evaluate authorization requests according to the rules defined in policies. that is used to determine the Access control procedure (request/response) i.e. deny or/ grant access as bellow

Procedure PEP [31]

```
{ Call ValidateCertificate(Path)
//checks the certificate validity by accessing certificate
If (certificate==valid)
Call PDP (attributes of subject, resource, action)
}
```

**Combining algorithms**: XACML [33] defines a number of combining algorithms that can be identified by a RuleCombiningAlgId or PolicyCombiningAlgId attribute of the <Policy> or <PolicySet> elements, respectively. The rule-combining algorithm defines a procedure for arriving at an authorization decision given the individual results of evaluation of a set of rules. Similarly, the policy-combining algorithm defines a procedure for arriving at an authorization decision given the individual results of evaluation of a set of policies. Standard combining algorithms are:

- Deny-overrides (Ordered and Unordered), Permit-overrides (Ordered and Unordered),
- First-applicable and
- Only-one-applicable.

In the case of the Deny-overrides algorithm, if a single <Rule> or <Policy> element is encountered that evaluates to "Deny", then, regardless of the evaluation result of the other <Rule> or <Policy> elements in the applicable policy, the combined result is "Deny" [33].

Here in this paper Deny-overrides (Ordered and Unordered) policy combining algorithm are used to making decision authorization for accessing application/ services i.e

Procedure combinePolicy(r[])

```
{ for (i=0; i<lengthof(r); i++)
  If (r[i]=deny)
  return deny;
```
else continue; 
}

**Policy Decision Point (PDP):** After getting request from PEP, PDP retrieves corresponding policy from the policy store which matches with the subject’s, resource’s and action’s values.

```
Procedure PDP
{
    Call getpolicies (“path”) // find all policies applicable to the request based on attributes of subject, resource and store result in array r[].
    Call combinepolicy (r[].) // combine result of all policies.
    return result;
}
```

**Working of PDP [31]**

Procedure for sending request to PDP  
Procedure validating user and sending request  
{  
    Call IsValidUser ()  
    If o.IsValidUser(Login1.UserName, Login1.Password) Then Response.Redirect (“~/ToEnterDiscoveryServices.aspx”)  
    Else MsgBox (“Incorrect UserName/Password”)  
    Response.Redirect (“~/Login.aspx”)  
    End If  
    // if (user==valid) returns true; //if user is registered member of domain.  
    IsValidCertificateCredentials() // get identity certificate from domain authority.  
    Call SearchTheServices() // redirects request to discovery service in coregrid which returns list of services in domains. User selects the service and make access request  
    Call SendRequestToOtherDomain (attributes of subject, resource, action, authorization assertion)  
    //returns authorization credential from filter authority to enter into domain and calls PEP of other domain[31].

VI. ACCESS CONTROL FRAMEWORK WITH PEP AND PDP COMPONENTS

In the working of framework, End user requests access to an application/services. Request is routed through a PEP. PEP transfers request details to a PDP for evaluation and authorization decision. The PDP refer to a policy repository(store) and possibly a policy information point. Policy is administered through a ‘central’ PAP. PEP enforce the decision of the PDP either it is deny or grant access by end user.

VI. CONCLUSIONS

One of the biggest challenges in Cloud security is Access Control. It determines how do you control who has access to what systems and technology within your enterprise? Operating systems and applications all have different ways of managing this. As a result, the more applications you use, the more challenging it is to safely and securely manage your users. This problem becomes even more difficult in the public cloud. The ability to control the technology is limited and it’s difficult to leverage tools such as single sign on/federation products. The framework is about controlling access to resources based on policies;

**ACKNOWLEDGMENT**

The guidance and support given by Shri Ravi Shankar Singh is gratefully acknowledged.

**REFERENCES**

1. Ian Foster, Yong Zhao, Ioan Raicu, Shiyong LuDepartment of Computer Science, University of Chicago, Chicago, IL, USA, 2Computation Institute, University of Chicago, Chicago, IL, USA, 3Math & Computer Science Division, Argonne National
I, USA, An Access Control Framework for...

googledocs/
make-cloud-computing-better
5. http://gigaom.com/cleantech/how-cloud-based-gmail-is-energy-
efficient/
7. www.research.yahoo.com/node90
8. en.wikipedia.org/wiki/Amazon_S3
9. en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud
10. en.wikipedia.org/wiki/Windows_Azure
11. en.wikipedia.org/wiki/IBM_cloud_computing
12. en.wikipedia.org/wiki/Eucalyptus_(computing)
14. www.scribd.com/doc/36472413/51/Tutorial-5-Amazon-
Virtual-Private-Cloud-Amazon-VPC
15. www.cloud.crls.com/Cloud-Services
16. www.microsoft.com/Private Cloud whitepaper
17. V. Rajan, R.S. Singh, “A Mechanism for Flexible Access Control
During Co-Ordinated Resource Sharing in Enterprise Grids” 1st
Int’l Conf. on Recent Advances in Information Technology | 
RAIT-2012 |
mechanism based on Key-Chain-Web model using authorization
contexts”, International Journal of Computer Applications (0975 –
8887), Volume 50 - Number 13
19. J. D. Amit Sangroya, Saurabh Kumar and V. Varma, “Towards
analyzing data security risks in cloud computing environments,” in 
255256.
20. D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Somani,L.
Youseff, and D. Zagorodnov, “The eucalyptus open-source cloud-
computing system,” in Proceedings at 9th IEEE/ACM Inter-
Rao, E. Rom, R. Sailer, W. Schildhauer, D. Srinivasan, S. Tal, and 
E. V. and, “Security for the cloud infrastructure: Trusted virtual
data center implementation,” IBM Journal, Resources & Developments, vol. 53, no. 4, p. 12, 2009
22. Z. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S.
Jenq, “Securing elastic applications on mobile devices for cloud
computing”, in Proceeding at ACM Cloud Computing Security
Workshop (CCSW), 2009.
23. C. Danwei, H. Xiuli, and R. Xunyi, “Access control of cloud
24. S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable,
and fine-grained data access control in cloud computing,” in 29th
Grid Environment” Indian Journal of Computer Science and Engineering (IJCSE)
28. www. stevenimmons.org/2012/02/policy-enforcement-point-
pattern/
29. www.docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-
en.pdf//PDP, Damian, S. De Capitani di Vimercati, S.
Paraboschi, and P. Samarati. Design and implementation of an
access control processor for XML documents. Computer Networks,