International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 1 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT2062125

An ABAC Based Policy Definement for Enriching Access Control in Cloud

Yagnik A. Rathod[1], Dr. Chetan B. Kotwal[1], Dr. Sohil D. Pandya[3]
[1]Research Scholar, Computer/IT Engineering, Gujarat Technological University, Ahmedabad, Gujarat, India
[2]Professor & Head, Electrical Engineering Department, SVIT, Vasad, Gujarat, India
[3]Assistant Professor and Head, MCA Department, SVIT, Vasad, Gujarat, India

ABSTRACT

Cloud Computing becomes most preferable solution for satisfying the various requirements of organizations and institutions. Different types of clouds like IaaS, PaaS, SaaS makes cloud capable to fulfills the client's different kind of needs like computer processing power, storage spaces, databases, software, application, web based solutions. Cloud computing can also be useful and worthy in providing certain customized solutions to enhance the capability of legacy systems in terms of effectiveness, reliability and optimization by replication of environment up to satisfactory extent. To provide adequate security solutions for cloud is still a challenging task and access control mechanism is one of the domain which demands significant attention on the mission towards securing clouds. In this paper, our work primarily focus on defining ABAC components, mapping functions and access control policies composed by access rules. Amazon Web Services is one of the most prominent cloud providers. Identity and Access Management (IAM) and Amazon S3 are access management and storage facilities of AWS respectively. ABAC based access policies are attached with the user and storage components for authorization.

Keywords : Cloud Computing, ABAC, IAM

I. INTRODUCTION

Nowadays cloud computing provides storage resources as an effective, scalable and easy to use by sharing it among different participants for optimal use. But no one can afford to ignore the risk of breach in security to compromise the crucial data in such a distributed environment. With this modern era, handling of personal or professional data is very important and primary condition for making a call whether to participate in distributed environment or not. Management of data is a challenging task specifically for distributed systems where data is in different formats and have to strictly follow the norms as per geographical location or rules of the participating institutes or organization. Therefore methodology used to obtain data objects should be capable of protecting it with appropriate standard security provisions. Any access to data object or resources should not be approved without concern of it and this should be assured by appropriate policies for regulating access request. One of the most preferred approaches that provide assurance to protect data and resources from malicious activities is access control mechanism. Defining access control policies, analysis of policies for validation, verification and approach for enforcing the methodologies are few among the all challenges. In today's multi cloud solutions emerged by integration of solutions from all over the world where data

resides on different geographical locations with various format and security constraints for access policies to control the access to data. Well known traditional access control models are like Mandatory Access Control, Discretionary Access Control, Role Based Access Control and Attribute Based Access Control. Mandatory Access Control model gives access based on a decision of centralize component and facing challenge like separation of duties with no flexibility towards hierarchical structure of access rights and make it inappropriate for cloud like structure[1]. Discretionary Access Control works on the principle of ownership in which access rights to particular object is controlled by owner of that object. It does not suited to environment like cloud as appropriate mechanism of access rights is missing[1]. Role Based Access Control as shown in figure 1[1] works on identification and assignment of roles and membership to user or group of user but for cloud it faces challenges like role engineering and role explosion. Attribute Based Access Control as shown in figure 2[2] uses attributes of subject, object and operations to conclude the access decisions and these attributes are important for defining access policies. Identifying appropriate and applicable attributes in cloud is challenging task and need to be deal with for its adoption in cloud environment. For multi-cloud infrastructures the challenges access control faces are like Continuous access, Different regulations, Dynamic access policies, User-friendly management, Continuous control, Scalability, Resource sharing and interoperability, Validation and verification of access control policies, On line tracing of access control polices execution and Testing of access control systems[3]. Attribute-Based Access Control (ABAC) provides greater flexibility to express fine-grained access control policies in a simple and more powerful way based on attributes of users, subjects, and objects[4][5][6]. Verification of policies for getting desired outcomes as per the definition and validation of policies to determine that is it the correct definition as per requirements of the systems and thus proper verification and validation are very crucial aspects for improving desired level of security for cloud like infrastructure. Currently there are various methods of policy specification and analyses are available with different policy languages which can be categorized as rule based, logic based or ontology based. It is still demand of time to provide language to specify access policies which is easy to use, stretchy and expressive. This language should be bundled with characteristic like simple yet powerful syntax with careful semantics and openness for verification and validation. Policies are distributed to various places among the different stakeholders and that's why it should be enriched by policy specification and management facilities which can be easily used by non IT participants. Traditional access control models are mostly static in nature and coarsely grained. These models are not tailored for critical infrastructure where it requires controlling their assets on a fine level of granularity.

There are many cloud service provider in all over the world but we opted to utilize the functionalities of Amazon Web Services (AWS) as it suits well to our purpose. AWS is a flexible, cost-effective, easy-to-use, reliable, scalable, having High Performance and secure cloud computing platform and provides a mix of IaaS, PaaS, and SaaS. AWS is capable to make available the computer processing capabilities, disk space, networking, facilities of database and similar kind of infrastructure on readily available mode as and when required with prices-as-use method. Amazon Simple Storage Service referred as S3 is developed to offer affordable virtual disk space which is long-lasting, scalable, highly responsive and online object base storage. AWS provides facility to store the registered client's data as an object of S3 into S3 bucket to keep it in proper structured way. Identity and Access Management component provided by AWS with the objective of enriching its clients by facilitating IT admin to decide what kind of action user can perform on the resources for which it is authorized for access. The purpose is to help

administrators for management of AWS user identities and their varying levels of access to AWS Resources and provides secure and efficient way for connecting AWS users to AWS resources for their specific requirements.

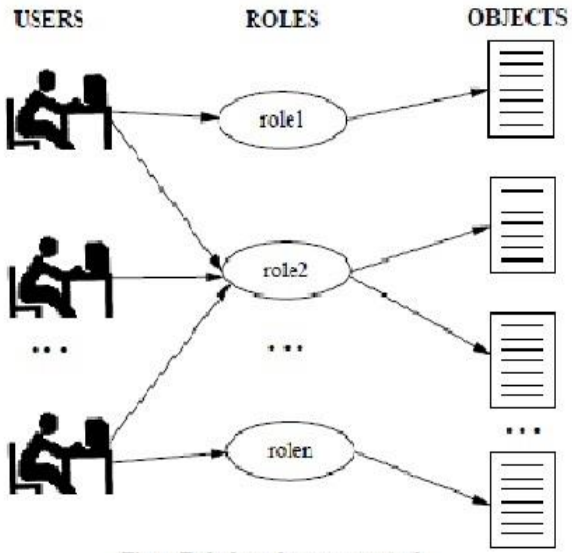

Fig 1: Role-based access control

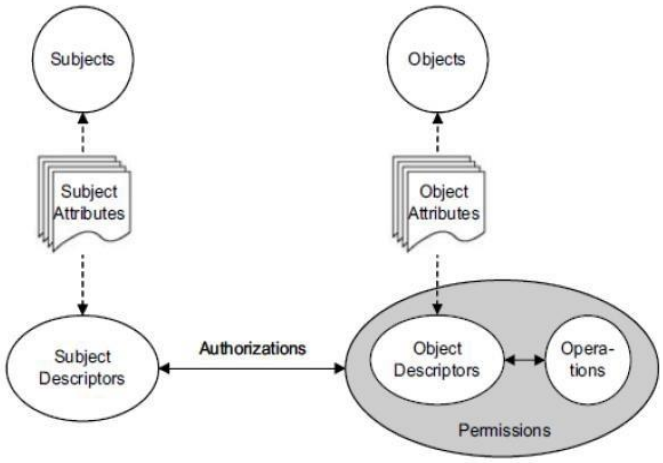

Fig 2: Overview of the ABAC model

II. RELATED WORK

Cloud computing platform is wonderful for coordination among the users for effective usage of shared resources. Resource sharing is one of the most popular models of current era. These models can only be accepted if its proven worthy by appropriate security model which are capable to differentiate authorized and unauthorized users and well aware about cyber attacks most likely happen.

A New Attribute Based Access Control Model (AR-ABAC)[7] is proposed to enhance cloud security by integrating attribute-rules. This model is base on relationship between subject and object with consideration of its level of sensitivity. For experimental purpose authors used 24 attributes with keystone and noticed on 25% increase in token generation on average compared to no attribute and it's considered to be in acceptable range.

Cloud computing can adopt methodology of self healing to counter the attacks done intentionally or by mistake by legitimate users[8]. Cloud computing platform like OpenStack has a component known as Keystone for identity management service for managing constraint access to its resources. To identify and managing good numbers of scenarios which replicate the insider attack is still a challenge but very crucial and important for understanding behaviors and base on that generating the patterns. Auditing is one of area of concern for ABAC and integration of SQL database with log services of OpenStack can be used to register noticeable improvement compared to other approaches[9]. Flexibility of access control can be improved by the ABAC Extension which includes user attributes additional to OSAC model without modifying RBAC architecture of OpenStack to save efforts of altering existing access control framework of it[10]. There are many interesting capabilities of PM that can be explored as extensions to the proposed model such as applying combination of different access control policies defined in PM, or incorporating deny relations and constraints in the policies[11].

Dynamic access control model is based on the analysis of different access request context and it selects most appropriate context with reference to

analysis. To dynamically manage access request, base on the outcome of analysis appropriate applicable context-aware security policies has been identified from repository. It is believed that the proposed framework will help in working out essential access control challenges during set up process of the smart grid in the upcoming years[12].

Attribute Based Access Control (ABAC) is the most appropriate and promising for cloud computing which is highly dynamic in nature[13]. Element for Privacy awareness can be added to ABAC model for building trust while sharing the attributes for access decisions.

Realization through implementation and assessment of authentication and authorization framework of generic ABAC solution for Future Internet test beds federation and it is envisioned for a generic access interface which make it possible to do the communication between both parties[14]. For specific authorization access control policies crafted and written using XACML for smart grid for SealedGRID project[15]. Attribute-Based Access Control as a Service of cloud is proposed to be integrated by any institution within their environmental configuration and for supporting all cloud platform and that service is recognized as ABACaaS[16]. The concept of firewall that do realization of Attribute-based Access Control drafted using XACML for securing systems residing at far end from unlawful access which consist of the definition and realization of attribute-based access control policies[17].

III. PROPOSED WORK

In this paper our main focus is on defining ABAC components and policies which are the key component for decision to be made for authorization under access control model.

Basic components of ABAC are defined as below.

- Subject(S): Collection of Authorized users. S_i is member of S, where $1 \le i \le |S|$.
- Resource (R): Collection of Resources whose accesses need to be constrained. R_i is a member of R, where $1 \le i \le |R|$.
- Environment (E): Collection of operational Conditions needs to be satisfied. E_i is a member of E, Where $1 \le i \le |E|$.
- Operations ( ): is a collection of possible operations.
- $_i$ is a member of , Where $1 \le i \le |$ $|$.

Policy (P): P is described as collection of authorization policies. P_i is a member of P where $1 \le i \le |P|$.

Attributes of these components are defined as below.
Subject Attributes (S_{attr}): A collection of attributes identified for participating subjects.
S_{attri} is a participant of S_{attr}, where $1 \le i \le |S_{attr}|$.
Resource attributes (S_{attr}): A collection of attributes identified for participating resource.
R_{attri} is a participant of R_{attr}, where $1 \le i \le |R_{attr}|$.
Environmental attributes (E_{attr}): A collection of constraint specific attributes.
E_{attri} is a participant of E_{attr}, where $1 \le i \le |E_{attr}|$.
Function for mapping attribute values to its attributes of participating entities is as below.
F_s: $S \times S_{attr} \rightarrow$ {A | A is a value of an attribute for subject from applicable domain of attribute values} where function Fs are a Subject-Subject attribute assignment to its instance.
F_R: $R \times R_{attr} \rightarrow$ {A | A is a value of an attribute for resource from applicable domain of attribute values} where function F_R are Resource - Resource attribute assignment to its instance.

F_E: $E \times E_{attr} \rightarrow$ {A | A is a value of an attribute for environment from applicable domain of attribute

values} where function F_E is environment-environment attribute assignment to its instance.

For any authorization, required instance of ABAC is defined as below.

ABAC Instance (Ø): Ø can be describes using is a 4-tuple ‹ F_S, F_R, F_E, P > where F_S, F_R, F_E and are described above.

For example rules in Policy P can be described as below.

P 1: Manager can add technician to his project only.

<{ROLE=Manager},{Type='project', object.project_assigned = resource.project_assigned}, {Shift = Day Shift}, add >

P 2: Technician can create faults to the project he is allocated.

< {ROLE=Tester}, {Type = 'Fault', Subject.project_assigned =resource.project_assigned&&subject.name=resource. assignd to}, {Shift = Day Shift}, Create >

Policies and rules of policy crafted using components of ABAC are stored in policy repository. Policy specification are created using format of JSON. These policies are used in cloud platform to support and enhance the authorization. For our experiment we have used Amazon Web Services (AWS) with the services provided by this cloud. We have used AWS's S3 service for object storage and Identity and access management (IAM) for effectively manage users or group of users and roles. Amazon Simple Storage Service known as Amazon S3 provides facilities to store and manage data objects. We created storage object known as S3 bucket to kept data. For our project, each data resources of needs are kept in different S3 buckets we have created using Amazon S3. Within every bucket we can create data objects like folders in windows platform and kept data with provisions of access rights as per configuration made for S3 bucket as whole or smallest data object at lower level. Access rights for each bucket are specified using existing bucket policies or our customized access policies can be added. AWS IAM is used for creating users and controlling its access rights to various data resources. We have created different users using IAM and attach our ABAC policy to respected user which gives access rights to the data objects of the projects to whom the user is authorized. These resource policies and policies attached to IAM user grant the permission to specific user to perform specific actions on that resource and define under what conditions these policies applies.

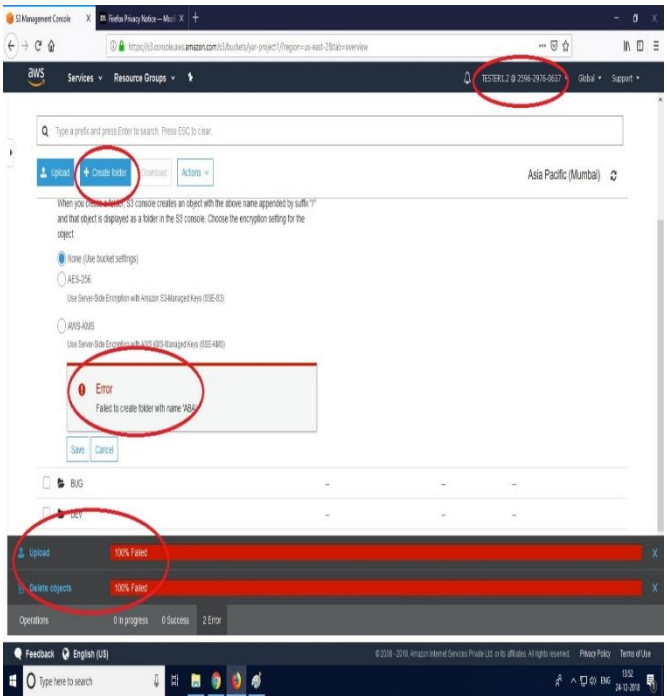

Fig 3. Denial based on policy evaluation

When the user tries to perform action on the bucket's object, the request is intercepted by the bucket and then the permission attached to the bucket is verified to see whether the given user is allowed to perform certain action or not. If the user is allowed to perform such action the operation is successful else it fails. Figure 3 is a one of the screenshot of experiment for denial of creating a data object in bucket to which user is not authorized.

IV. CONCLUSION

We have specified components of ABAC based access control model and defined access rules which are responsible for approving or denying access request of a user for any resources. Access rules are combined to form access policies and defined using JSON. These access policies are attached to user, resources or both using AWS framework for authorization purpose. Extending our work of defining ABAC components and access policies for critical infrastructure like smart grid with suitable framework and tool for policy generation and management is our future work. We acknowledge AWS for providing support for registration and providing trial accounts for many days.

V. REFERENCES

[1]. B. Jayant.D, U. A, A. S, and M. G, "Analysis of DAC MAC RBAC Access Control based Models for Security," Int. J. Comput. Appl., vol. 104, pp. 6–13, 2014, doi: 10.5120/18196-9115.

[2]. T. Priebe, D. Wolfgang, S. Christian, and K. Nora, "Supporting Attribute-based Access Control in Authorization and Authentication Infrastructures with Ontologies," J. Softw., vol. 2, 2007, doi: 10.4304/jsw.2.1.27-38.

[3]. F. Lonetti and E. Marchetti, "Issues and Challenges of Access Control in the Cloud," in WEBIST, 2018.

[4]. V. Hu, D. Kuhn, and D. Ferraiolo, "Attribute-Based Access Control," Computer (Long. Beach. Calif)., vol. 48, pp. 85–88, 2015, doi: 10.1109/MC.2015.33.

[5]. V. Hu et al., "Guide to attribute based access control (ABAC) definition and considerations," Natl. Inst. Stand. Technol. Spec. Publ., pp. 162–800, 2014.

[6]. X. Jin, R. Krishnan, and R. Sandhu, "A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC BT - Lecture Notes in Computer Science," Lect. Notes Comput. Sci., vol. 7371, no. Chapter 4, pp. 41–55, 2012, Online]. Available: http://dx.doi.org/10.1007/978-3-642-31540-4_4%5Cnpapers2://publication/doi/10.1007/978-3-642-31540-4_4.

[7]. K. Riad, H. Hu, Z. Yan, H. Hu, and G. Ahn, "AR-ABAC: A New Attribute Based Access Control Model Supporting Attribute-Rules for Cloud Computing," in 2015 IEEE Conference on Collaboration and Internet Computing (CIC), Oct. 2015, no. December 2016, pp. 28–35, doi: 10.1109/CIC.2015.38.

[8]. C. E. Da Silva et al., "Self-adaptive authorisation in OpenStack cloud platform," J. Internet Serv. Appl., vol. 9, no. 1, p. 19, 2018, doi: 10.1186/s13174-018-0090-7.

[9]. S. Patel and Y. Rathod, "An Auditable Attribute Based Access Control Mechanism in Openstack Cloud Environment," Int. J. Innov. Res. Comput. Commun. Eng., vol. Vol. 4, no. Issue 5, pp. 10241–10246, 2016, doi: 10.15680/IJIRCCE.2016.

[10]. B. Tang and R. Sandhu, "Extending OpenStack Access Control with Domain Trust," in Network and System Security, 2014, pp. 54–69.

[11]. S. Bhatt et al., "An Attribute-Based Access Control Extension for OpenStack and Its Enforcement Utilizing the Policy Machine," in 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC), Nov. 2016, pp. 37–45, doi: 10.1109/CIC.2016.019.

[12]. S.-S. Yeo, S.-J. Kim, and D.-E. Cho, "Dynamic Access Control Model for Security Client Services in Smart Grid," Int. J. Distrib. Sens. Networks, vol. 2014, pp. 1–7, 2014, doi: 10.1155/2014/181760.

[13]. M. Ed-Daibouni, A. Lebbat, S. Tallal, and H. Medromi, "Toward a New Extension of the Access Control Model ABAC for Cloud

Computing," in Advances in Ubiquitous Networking, 2016, pp. 79–89.

[14]. E. F. Silva and C. M. Saade, "ACROSS-FI : Attribute-Based Access Control with Distributed Policies for Future Internet Testbeds," in ICN 2015 : The Fourteenth International Conference on Networks ACROSS-FI:, 2015, no. c, pp. 198–204.

[15]. G. Suciu, C. Istrate, A. Vulpe, M.-A. Sachian, and M. Vochin, "Attribute-based Access Control for Secure and Resilient Smart Grids," 2019, doi: 10.14236/ewic/icscsr19.9.

[16]. A. Meshram, S. Das, S. Sural, J. Vaidya, and V. Atluri, "ABACaaS: Attribute-Based Access Control as a Service," 2019, pp. 153–155, doi: 10.1145/3292006.3302381.

[17]. C. Ruland and J. Sassmannshausen, "Firewall for Attribute-Based Access Control in Smart Grids," in 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), Aug. 2018, pp. 336–341, doi: 10.1109/SEGE.2018.8499306.

Cite this Article :

Yagnik A. Rathod, Dr. Chetan B. Kotwal, Dr. Sohil D. Pandya, "An ABAC Based Policy Definement for Enriching Access Control in Cloud", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 1, pp. 586-592, January-February 2019. Available at doi : https://doi.org/10.32628/CSEIT2062125
Journal URL : http://ijsrcseit.com/CSEIT2062125