# Extending Access Management to maintain audit logs in cloud computing

Ajay Prasad
University of Petroleum and Energy Studies
Dehradun, India

Prasun Chakrabarti
Sir PadampatSinghania University
Udaipur, India

***Abstract*—considering the most often talked about security risks in cloud computing, like, security and compliance, viability, lack of transparency, reliability and performance issues. Bringing strong auditability in cloud services can reduce these risks to a great extent. Also, auditing, both internally and externally is generally required and sometimes unavoidable looking into the present day competition in the business arena. Auditing in web based and cloud based usage environments focuses mainly on cost of a service which determines the overall expenditure of the user organization. However, the expenditure can be controlled by a collaborative approach between the provider company and the user organization by constantly monitoring the end user access and usage of subscribed cloud services. Though, many cloud providers will claim of having a robust auditable feature, the generic verifiability with sustainable long term recording of usage logs do not exist at all. Certain access management models can be perfectly extended to maintain audit logs for long terms. However, maintaining long term logs certainly has storage implications, especially with larger organizations. The storage implications need to be studied.**

***Keywords— Cloud Computing; Access Management; Audit logs***

## I. INTRODUCTION

In most of the business audits the primary focus is on the bills of several activities that the employees or management incurs while performing their tasks. For example, the employee CTC and the formulations of overtime duties are considered while auditing. It goes even further as part of auditors to ascertain whether the employees' CTC and other costs are at par with their roles and tasks done for the company. The auditors also ensure that the facilities, equipment etc. provided to the employees are properly utilized and are not used for vested purposes as well as in malpractices. In similar fashion, on incorporating cloud computing, major tasks of an organization will shift onto cloud computing. Thus, the usage pattern of the cloud services by the employees will affect the overall billing of the cloud services to a great extent. Not only that, Gartner [1] emphasizes the difficulty of auditing for security. The major difficulty would be of tedious procedures and arrangements that needs to be made by the service providers to address an auditing demand. Looking at this, the service providers would be quite reluctant (though not openly) to support the auditing demands in totality. Gartner [1] suggests to have an internal audits done by professional services or IT consulting firms for cloud usage. The issue is not that simple though. The parameters of audit of usage and security should not be through a one sided records keeping. Hence, "audit yourself" as suggested by Gartner [1] may be very less comprehensive and may not address all the aspects of usage of cloud. When the organizations will start relying more and more on cloud services, the aspects of cost will come into play more and more. Apart from all these, the fact of mal-practices and threats are also an issue which needs a constant monitoring of usage of services by the employees of the organizations.

### *A. Audit of usage*

Major parts of the audit of usage involves:

*a) Duration of usage Vs task assigned to the employees.*

*b) Services used Vs task assigned to the employees.*

*c) Uploads and downloads to and fro from cloud Vs Compliance.*

The services used by the employee is related to the roles assigned to him her. The roles are managed by the management which can be formulated through the policies of the management. The overall policies and roles are/can be managed by an almost automated access management coordinating with the cloud provider. Financial audits, operational audits and compliance audits as in [2] are also necessary and applicable on cloud usage. The [2] also describes the roles of internal and external auditors in general businesses. The [3] emphasizes the importance of auditing at scheduled intervals mutually agreed between providers and users based on mutual agreements. While putting on important points of risks in cloud computing [4] puts forth 2 points namely, investigative support and long term viability of usage in clouds. Both these points are more or less related to long term audit ability in cloud scenarios.

## II. AUDITABILITY AND ACCESS MANAGEMENT

Currently many methodologies of access controls are used namely Discretionary Access Control (DAC) [5] where the access is granted based on discretion of the user of the resource; the Mandatory Access Control (MAC) [6] is mostly designed to grant access only through the mandatory system enforcement policies and not that of users. However, MAC doesn't discriminate over the various types of resources available to the user once he complies the mandatory entry requirements. This can lead into authorization issues. The Identity based Access Control (IBAC) [7] grants access through access control lists (ACL) [8], which is a list based on the users' identity. The Role based Access Control (RBAC) [9] provides access to a resource based on the user's role in the organization or system. The Lattice Based Access Control (LBAC) [10] maintains an ordered set of security labels combined with a set of categories. Authors of [11] notes that

the effectiveness of all above access mechanisms are not sufficient for maintaining fine grained access control policies in today's collaborative environments including clouds. [11] goes further to propose Attribute Based Access Control model (ABAC). The model consists of two aspects i) a policy model and ii) an architectural model. The policy in policy model is applied to the web service access control through architectural model. The ABAC model can be used to have a structured log maintained for audit purposes and having fine grained policy frameworks. The aspect of maintaining long term logs can be realized without losing too much space.

Policies → access → policy based audit logs

A typical scenario of maintaining ABAC based audit logs is shown in figure 1.

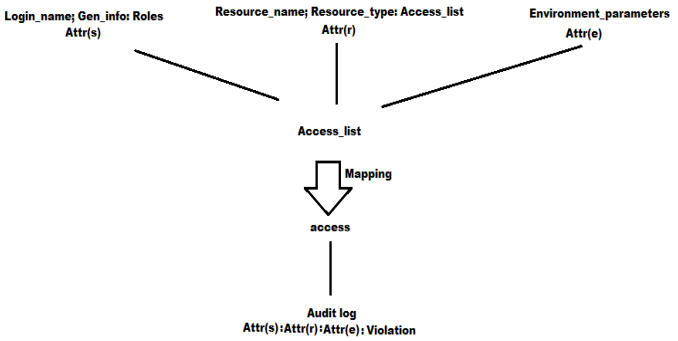


Fig. 1. Attribute Based Access Control

Thus, audit logs for each request can be maintained in a structured and precise fashion. Also, we can have automated analytics for the auditors using the ABAC audit logs. The audit log can be formalized based on the ABAC model as follows:

$$Attr(e); Attr(s); Attr(r); Access(s, r, e); Matched\_Violation$$

The time stamp is assumed to be integral part of access. Matched violation, is the factor or measure of violation (if any) that has occurred due the access. For example, in most of the cases the user tries to access the resource off time or during the time the resource is under secret editing. Although, the access may not be granted to the resource but, the move by an employee to access it is in fact a violation. The measurement of violation factor can be established by the one to one mapping of all possible violations with number sequences with attached weight age.

$$Violation\_Set = \{V_1, V_2, V_3, \ldots, V_n\}$$

$$Offence\_weightage = \{W_1, W_2, W_3, \ldots, W_n\}$$

Table I shows an example set of violation types and corresponding offence weight age based on the seriousness of the violation.

TABLE I. EXAMPLE RESOURCE VIOLATIONS

| Violation | Offence Weightage |
|---|---|
| Accessing Resource/service not ready (offline access confidential resource) | 0.3 |
| Login Failure on R | 0.01 |
| Resource/service not ready (non-confidential) | 0.1 |
| Accessing Resource/service not allowed | 0.2 |
| Editing Resource/service finalized (edit offline) | 0.8 |

We propose another access model i.e. Bitwise Attribute Based Access Control (BABAC), where, the methodology of maintaining fine grained policies and roles is in bitwise fashion. Also, the mapping of policies and roles onto access is bitwise, which is more compact form than the ABAC model. This can lead to save more space in log maintenance. The user roles are maintained as states and the mapping is just done by 'AND'ing the resulting policies and obtaining the access to a set of fine grained services/resources. The figure 2 depicts the access in BABAC model of mapping.

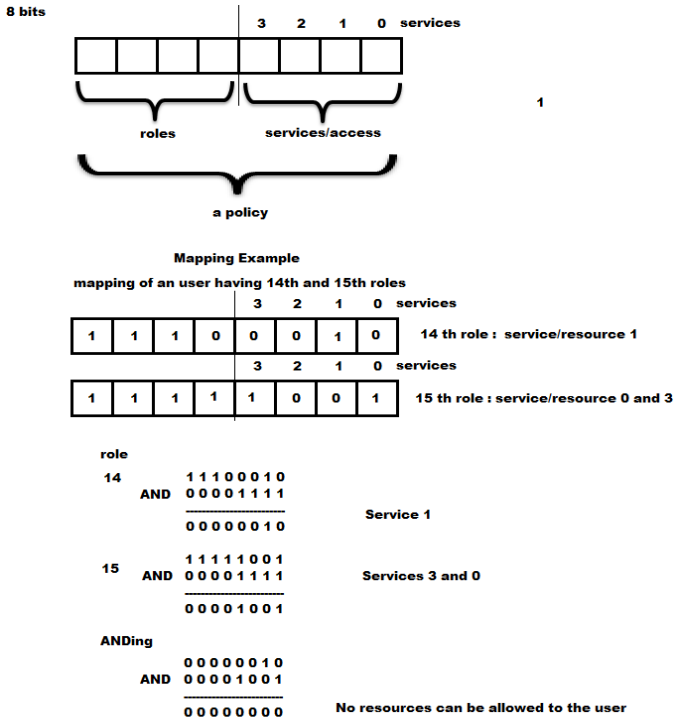


Fig. 2. Mapping in BABAC

The process of mapping is simple, however, maintaining these kind of policies can be little complex, however, one can make automated system consoles which can form these Bitwise policies with very less effort as in case of ABAC. In case of ABAC the Granularity and scaling can be done resource-wise whereas, in case of BABAC the scaling is restricted to an allowable bit size. In fact, methods can be devised to come out of these constraints as well. The audit log can be formalized based on the BABAC model as follows:

$$States\_word(s,r,e); Access\_word(s,r,e); Matched\_Violation$$

## III. Simulation and Results

A simulation in cloud environment was made to study the storage requirements of various methods discussed above. The simulation was carried out for 3 aspects of (long term) log maintenance while monitoring:

*a) Generic monitoring.*

*b) Generic monitoring with ABAC logs;*

*c) Generic monitoring with BABAC logs;*

Cloudsim tool [12] was used to simulate for the above factors. The simulation was made for 10 to 20 users with subjects and resources scaling from 10 each to 20 each. Although this might not be a real scenario but its more than enough to get good ideas about the storage implications of the long term log keeping for auditing purposes.

To understand the study of the logs we have to go along the three most important parts of the ABAC logs and BABAC logs, namely:

*a) Subject Pattern*

*b) Resource Pattern*

*c) Environment Pattern*

### A. In case of ABAC:

Typically, subject pattern involves login name, general information and state/role. The criteria for the state/role can be in a range (from no role to all roles). An organization can have as large as 50 roles on a particular application. Studies can't be made on all aspects of role. However, a typical 10-20 role subject pattern is studied in our simulation. That is, a user might not have any roles as well as at the same time a user can have 20 roles. The subject pattern is shown in figure 3a.

Resource pattern can depend upon number of atomically usable resources leased in cloud by the organization. It can be a database, it can be a file, it can be an application. It can be parts of application; it can be an OS or other platforms. Every resource can have access lists (typically ranging from 10-20) is classified by resource type, name and access list. Access list can be the set of roles that are allowed access on a particular resource. Thus, a resource pattern can be as shown in figure 3b.

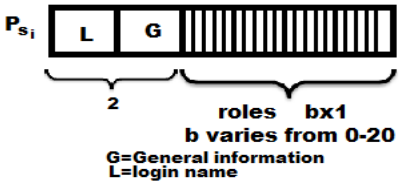


Fig. 3. a) Subject pattern (ABAC)

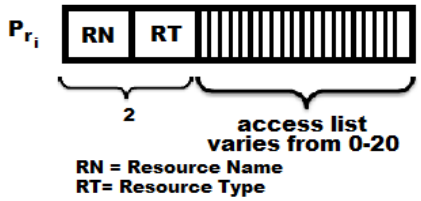


Fig. 3. b) Resource pattern (ABAC)

However, mostly the resource pattern will be owing to access list in the same manner as roles in subject pattern. The environment pattern can mostly contain time ranges. In some cases they can be holding specifics but, in our case we will be filling it as constant. Hence, the log space can be calculated as:

$$Log_i = P_{s_i} + P_{r_i} + P_e + S_v$$

Where, $P_{s_i}$= Size of subject pattern for ith log. $P_{r_i}$= Size of resource Pattern for ith log. $P_e$= Size of environment pattern (constant) and $S_v$= Size of violation information.

### B. In case of BABAC:

In BABAC model the roles are made in bitwise fashion. Hence, scaling factors may arise. However, for 0-20 roles 4 bit is required which can be scaled to 128 roles on 8 bit. However, the user attributes remains the same.

The 128 roles bit can contain environmental constraints also. As shown in figure 4a) the login name and general information parts are same as in case of ABAC. However, the roles are now bitwise represented and given a space of 2 bytes (a word).

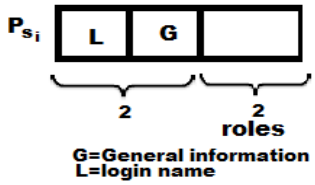


Fig. 4. a) State Pattern (BABAC)

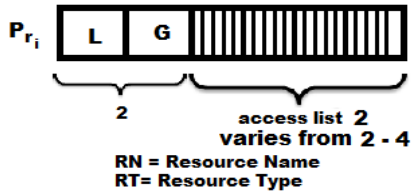


Fig. 4. b) Resource Pattern (BABAC)

The resource pattern will hold the resource name and resource type as in ABAC but the access list can be fixed to 16 bits (2 bytes) (see figure 4b).

However with 0-20 accesses will make us add 1 more word (after 16 accesses). In BABAC the environment pattern is part of the states in the subject pattern only. Thus, the overall log space can be calculated as:

$$Log_i = P_{s_i} + P_{r_i} + S_v$$

Where, $P_{s_i}$= Size of subject pattern. $P_{r_i}$= Size of resource Pattern for ith log. $S_v$= Size of violation information.

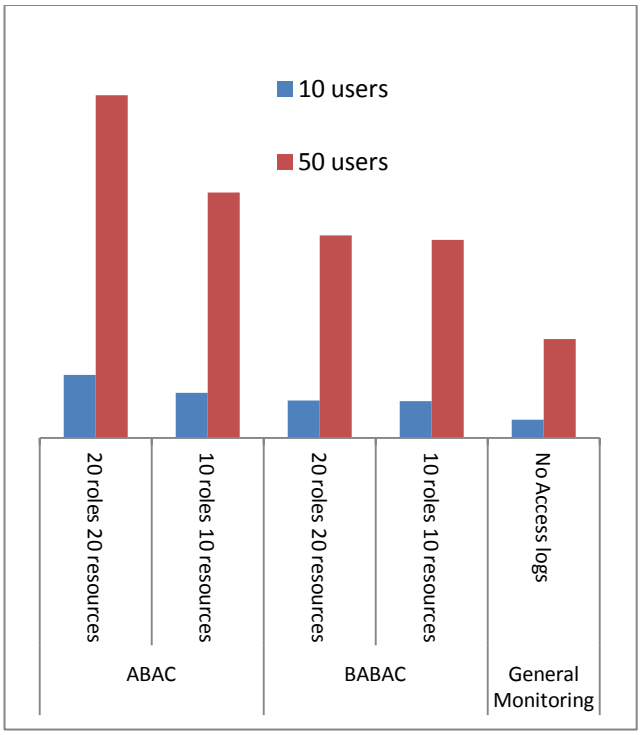


Fig. 5. Storage implications in various methodologies

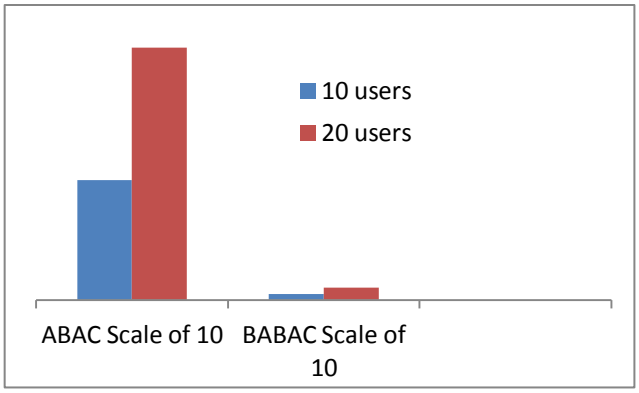


Fig. 6. Storage implications while scaling from 10-20 roles and resources

Various simulations were made to assess the storage implications, like, a) space required for 10-50 users in one year. b) change in space required if roles and access lists are scaled. The results were as expected. Considering the BABAC binary format its space requirements is much less than ABAC. However, ABAC is much simple to maintain in cases of huge number of roles and resources. As expected the storage requirements might be much more as revealing in figure 5, where the generic logs are requiring much lesser space than ABAC and CABAC logs. Figure 6 reveals the scaling factor, if roles and access lists (resources) are added, then logs sizes will get affected, as well as number of logs will also increase resulting in space requirements. Scaling BABAC looks much more efficient than ABAC. The change in case of BABAC is almost negligible. However, the overall results show that in any case ABAC and BABAC both are suitable and can be used with some costs pertaining storage.

## IV. Conclusion

The need for long term logs maintenance in cloud will facilitate both internal and external auditing. A more formal approach would be to extend the access management to provide long term log maintenance, say, for years. These logs can then be made readable through automated systems to the auditors for assessments. Among the various access management systems, the ABAC was found more suitable to the needs of long term log keeping. However, our proposed model BABAC (based on the principles of ABAC) can be more efficient in terms of space. Thus, overall long term monitoring can utilize ABAC or BABAC to facilitate the auditable cloud computing environments.

## References

[1] Daryl Plummer, "The Business Landscape of Cloud Computing," Gartner, at http://www.ft.com/cms/5e231aca-a42b-11e1-a701-00144feabdc0.pdf, retrieved Dec 20, 2013.

[2] Rick Hayes Roger Dassen Arnold Schilder Philip Wallage, "PRINCIPLES OF AUDITING- An Introduction to International Standards on Auditing "Prentice Hall FT, second edition, 2005

[3] Guiding Principles on Cloud Computing in Law Enforcement, IACP, Jan 31 2013.

[4] Cloud Computing in Law Enforcement: Survey Results and Guiding Principles, http://www.policechiefmagazine.org/magazine/index.cfm, retrieved 20 Jan 2014.

[5] National Computer Security Center (NCSC), "Glossary of Computer Security Terms" (NCSC-TG-004), October 21, 1988, http://csrc.nist.gov/secpubs/rainbow/tg004.txt

[6] D.E. Bell and L.J. LaPadula, "Secure Computer Systems: Mathematical Foundations and Model", Mitre Corp. Report No. M74-244, Bedford, MA, 1975

[7] Identity-based Access Control, Technical Brief, ProCurve Networking by HP, www.moonblinkwifi.com, retrieved 16 Feb 2014

[8] Internet Security Glossary, Version 2 RFC 4949 http://datatracker.ietf.org/doc/rfc4949/, retrieved Feb 2014.

[9] Ravi S. Sandhu et al., "Role-Based Access Control Models", IEEE Computer, February 1996, pp. 38-47

[10] Ravi S. Sandhu, "Lattice-Based Access Control Models", IEEE Computer, November 1993, pp. 9-19

[11] E Yuan, and J Tong. "Attribute Based Access Control (ABAC) for Web Services", In proceedings of the IEEE Conference on Web Services (ICWS'05), Orlando Florida, USA. July 2005

[12] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F. and Buyya, R., CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41: 23 – 50. doi: 10.1002/spe.995 2011.