Policy Based Role Centric Attribute Based Access Control Model

Policy RC-ABAC

.Vijayaraghavan Varadharajan[1]
Infosys Ltd.
Bangalore, India
Vijayaraghavan_v01@infosys.com

Alon Amid
Intern, Infosys Ltd.
Bangalore, India
Alon_520298@infosys.com

Sudhanshu Rai
Infosys Ltd.
Bangalore, India
Sudhanshu_Rai@edgeverve.com

***Abstract*— As network speed and storage capacities are rising faster and higher, the remote storage and 'cloud' concept is gaining momentum and is becoming increasingly popular with personal users as well as business users. At the same time, privacy awareness and business confidentially are important factors in every cloud and data sharing decision. For these and other reasons, the concept of access control models was developed in order to present complete solutions for privacy and confidentiality control in modern systems.**

Various access controls methodologies have been suggested in the literature, each simplifying and providing flexibility to previous methods. In this paper we will focus on two main methods which are Role Based Access Control (RBAC) and Attribute Based Access Control (ABAC). We will discuss the benefits and drawbacks of each of them, and therefore the need for a novel approach that combines the benefits of these two techniques. We are proposing a different hybrid model and justify how our suggested model and present why we think our suggested model (Policy RC-ABAC) will be more beneficiary in relation to a specific set of needs focusing on flexibility and auditability.

Keywords—access control; authorization; ABAC; RBAC

I. INTRODUCTION

Access control is defined as restriction of access to a resource or place to a selective group. This restriction is achieved by creating a set of rules to access the resource. Controlling access of a resource becomes very difficult when the volume of users is very high. In this paper we are referring access control as providing access to a resource over internet.

According to latest news of IEEE there will be 50 billion devices connected with internet by 2020 [14]. This enormously increasing number of users imposes some new challenges to access control systems. There are many different requirements for an access control system including scalability, flexibility, auditability and efficiency. Access control systems must be able to adapt to rapid changes, and yet be efficient both in the access authorization process and the auditing review process. Access control methods are many times differentiated as "coarse grain" and "fine grain" to describe the scope of the method. Considering the dynamic nature, usually more "fine grained" methods are preferred.

Many access control models have been researched and suggested in the literature, the two main ones are Role Based Access Control (RBAC) and Attribute Based Access Control (ABAC).

A. Role Based Access Control (RBAC)

In Role Based Access Control (RBAC), a set of permissions is associated with a role, and each role has a different set of permissions. Roles are assigned to users which in turn associate users with sets of permissions. RBAC has increased flexibility compared to previous identity based access control in terms of providing access in general, not for each person separately. We can easily audit all access permissions associated with a user by checking the permissions associated with the relevant role. This also makes it easy to determine risk exposure associated with a user. On the other hand, this easiness comes at the cost of research on determining roles and associated permissions, called as Role Engineering. In addition, increasing numbers of users and various dimensions associated with rules such as time of the day and location makes RBAC difficult to implement. If we examine the example of "an employee can book a meeting room" then we can associate 'book meeting room' permission with 'Employee' role. But if we have to add context like 'time of the day' or 'employee location' to the rule, then we must add additional roles to match this rules. When attempting to create a fine grained access control system this leads to the creation of many roles almost to the extent of a role for each user and in some systems more roles than users. This phenomenon is generally described as "Role Explosion". In addition, RBAC has few drawbacks which include the amount of effort involved in role engineering and the fact that it is not dynamic i.e. it requires advance knowledge of the subject and resources to determine access (doesn't perform runtime decisions).

[1]THIS WORK IS SUPPORTED BY COMMIT/ (A PUBLIC-PRIVATE RESEARCH COMMUNITY) PROGRAM.

B. *Attribute Based Access Control (ABAC)*

In Attribute Based Access Control (ABAC) users are assigned a set of attributes. In order for a user to be authorized access privilege to a resource, the user attributes, context attributes and resource attributes must be verified with the object's access policy. In this model the decision process of granting/denying access to a resource is performed during runtime, therefore it is considered as a dynamic model. This model allows fine grained access control without the need of investing major role engineering efforts. Any rule can be set by adding and removing attributes.

ABAC also has drawbacks, the main one being a very inefficient auditing process. For a system having n attributes, ABAC requires to check 2^n combinations to perform a full system audit. Auditing and determining risk exposure for a particular user becomes almost an impossible task for a high value of *n*.

In section 2 of the paper we present previous hybrid RBAC and ABAC models. In section 3 we describe our suggested improved model. We present and analyze the metrics which proves that our model is an improved model in section 4 and section 5 concludes and presents future works.

II. Related Works: Hybrid RBAC and ABAC Models

As seen in the literature, both RBAC and ABAC have their benefits and drawbacks. So if we need efficient auditing as provided by the RBAC model, yet the flexibility that is provided by the ABAC model, then the two models must be combined to a hybrid solution. A hybrid solution will not have the full power of each of the models' benefits, but it will improve performance and various problems of popular scenarios.

There have been many hybrid access control models suggested in the literature. From the various approaches in which RBAC-ABAC hybrid models have been discussed, three popular approaches are Role Centric, Attribute Centric and Dynamic Roles.

In the Role Centric-Attribute Based Access Control model, the user is assigned a static role. The role is assigned with a set of permissions. This is the maximum permission which can be associated with a role. Additional attributes restrict the permissions associated with role to create the final access policy. This model describes the permission filtering process which reduces the number of combinations required to check actual permissions associated with a user in a session using the Permission Filtering Policy (PFP) component. This model allows for an efficient audit as in the RBAC model with the addition of few calculations of the restricting attributes. However, the main drawback of this model is that it is only able to restrict permissions according to attributes, but it does not allow to add permissions, since role permissions are static.

In the Attribute Centric model, roles are assigned as just another attribute in an ABAC model. In this model a role does not have a set of permissions associated with it. As this model is again Attribute Based, without adding additional information to the 'role' attribute, it will have the flexibility of the attribute based models, but again we will have the same auditing inefficiency problem.

In the Dynamic Roles models, a user is assigned a temporary dynamic role based on the user's attributes and on context attributes. This model has few variations based on different sessions and different dynamic roles, but the common concept is that roles are assigned dynamically according to a set of attributes that is used in different access policies in the system. When using this model, one must evaluate carefully the amount of dynamic roles allowed in the system (in order the prevent role explosion), and on the other hand not to limit the number of roles as to the extent that the model will be once again similar to the standard ABAC. In any variation on this model, its implementation is not simple since an intelligent dynamic role mechanism must be implemented in order to create and maintain an efficient database of dynamic roles.

In addition, some double level hybrid models have been suggested such as the Attribute Based Policies RBAC model (ABP-RBAC) and the Bi-Layer Access Control Model (BLAC). These double level models try to optimize the flaws of the main hybrid models by splitting the access procedure into two parts, and therefore using the advantages of each access control model in a separate part. We will address these models in the metrics section.

III. Proposed Policy RC-ABAC Hybrid Model

The proposed model is mainly based on the Role-Centric and Attribute-Centric models. In the proposed model, the Role Centric and Attribute Centric Access Control methods have been combined and modified to allow the full flexibility of ABAC, and yet improve the efficiency of auditing. This modification is done using the logical gates AND/OR, and the definition of a mandatory 'ROLE' attribute.

A role should be defined in a non-specific (broad) way so it may apply to many people (good examples: Project Manager, Intern, and Software Engineer. Bad examples: Project Manager A, Project Manager B, Project Manager C, College Intern, High School Intern, Engineering Intern, Business Intern, Java Software Engineer, C# Software Engineer etc.). Privileges are granted based on a combination of role and attributes.

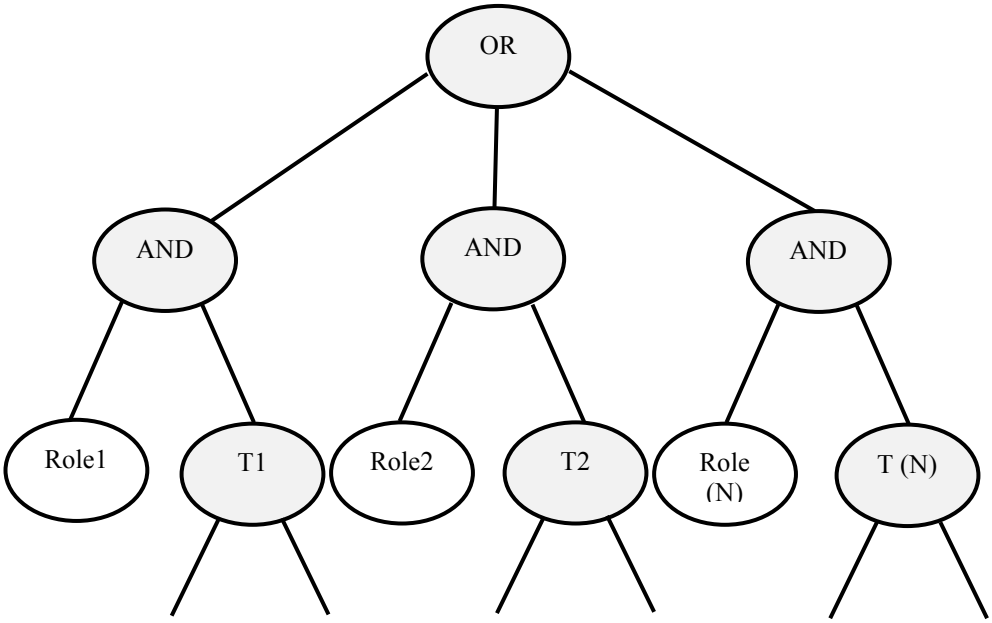

Fig. 1. Access Tree (AT)

When we define the "role" attribute as a mandatory attribute in each access policy boolean expression (of course each expression also has a combination of attributes), we get the same properties of the Role Centric hybrid model, but with the ability to add access instead of just limiting access.

Let *AT* be the access policy tree for an Object *O*. This is a standard attribute based access tree (a boolean access policy expression represented using logical gates and attribute leaves in a tree), as described in [12].

Figure 1 shows the Access Tree. In this Access Tree (*AT*), we define four requirements for the model:

- The tree root most be an OR gate.
- Each child of the tree root must be an AND gate. This must be a binary gate.
- For each sub tree of the level 1 AND gates, the right_child must be an attribute access policy tree (T (K)). This may also be an empty tree. This tree is named the "attribute subtree associated with Role K".
- For each sub tree of the level 1 AND gates, the left_child must be a Role attribute (Role (K)). This is a leaf. It may be an empty tree if and only if the right_child of the level 1 AND gate is an empty tree as well.

Once *AT* is defined, the rest of the model is Attribute Based Access Control [13]. This model has two main advantages that suit our requirements for a hybrid model which are not optimal in the Attribute Centric and the Role Centric Hybrid Models. They are:

- Auditing efficiency
- Policy design flexibility

The Role Centric Hybrid Model does not have the full flexibility of ABAC since it only restricts access after defining role privileges, and it cannot add privileges to new roles after defining role privileges (since role privileges are static).

The Attribute Centric Hybrid Model treats 'role' as a standard attribute (it is not required to be mandatory). Therefore, there might be access policies that don't include a role attribute, and the auditing efficiency advantage of the hybrid model is lost.

IV. Discussion: Metrics Comparsion

In order to understand why the Policy RC-ABAC is an improved model, we chose four main metric that relate to our subject problem: Auditing, Policy design, Implementation and Maintenance. In this section, we present the four metrics and how each of the existing models and our suggested model correspond with the metric.

A. Auditing

As explained, from the ABAC and RBAC models, ABAC auditing efficiency is low since for user auditing we must check if the user attributes will match the access policy of all the objects in the system and for object auditing we must check for all the users in the system if they match the access policy of the particular object. In Contrast, RBAC auditing efficiency is high since a user is associated with a role and a role is associated with the static set of privileges. Since the set of privileges is static, we can save it and access it with a complexity of $O(1)$.

The auditing efficiency of the Attribute Centric model remain the same as the ABAC auditing efficiency, since we are not making any structural changes in the access policy. We are just adding another attribute.

In the Dynamic Role and the Role Centric models, we have a basic static role structure associated with privileges (as in the RBAC model), with the addition of restricting attributes. Therefore, for user auditing we find the privileges associated with the role (O(1) or O(n) depending on the design of the structure), and then filter them according to the restricting attributes in O(m), where m is the average number of restricting attributes in the PFP. For object auditing, we check the users associated with the roles, they have privileges to the object according to the roles structure (O(1) or O(n) depending on the design of the structure), and then go through the list of users associated with these roles and check if each of them match the restricting attributes in the object's access policy in the PFP (O(n)).

Since the Policy RC-ABAC model is based on the ABAC access tree structure, we must make use of the extra information given to use in the mandatory first levels of the tree to increase auditing efficiency. For user auditing, we find all the objects that are associated with the user's Role (O(1) or O(n) depending on the extra Roles structure implementation). For each object we found, we check if the user matched the role's associated attribute subtree (instead of checking the entire access tree) (O(m)). Therefore the complexity of this operation is relative to the number or attributes in the role's subtree, and not to the entire attribute list such as in the Attribute Centric model. For object auditing, for each role in level 2 of the object's access control tree, we find the associated users to this roles (O(1) or O(n) depending on an extra Roles structure). For each user found, we check if the user matched his role's associated attribute subtree in the object (O(m)).

We can see that in the auditing metric, there is a strong disadvantage in the auditing efficiency of the Attribute Centric Hybrid model compared to the other three models since it does not use the "extra" information given to it by the 'role' attribute.

B. Policy Design: Defining the policy

Attribute Centric, Dynamic Role and Policy RC-ABAC models have similar policy design methods since they are completely based on a policy tree, and therefore every policy modification (addition and revocation of privileges) can be represented in the access tree.

The Role Centric model is based on the static Role privileges structure with the addition of the PFP component. Since the Role structure is static, we have the full flexibility of revoking and restricting privileges, but we do not have the option of adding privileges to a user that does not have original role privileges to the object. Therefore, in order to design a flexible access policy we must allow access to all the roles in the system, and restrict many of them using attributes in the PFP.

We can see that in the Policy Design metric, there is a strong disadvantage in the Design Flexibility of the Role Centric Hybrid model compared to the other three models.

C. Implementation

All of the models described (Attribute Centric, Role Centric, Dynamic Role and the Policy RC-ABAC) involve an attribute tree structure which is relatively easy to implement.

However, each model except for the Attribute Centric model has an additional component which must be addressed.

The Role Centric model's main component is the Role list and its associated privileges. As we are familiar with this structure from the standard RBAC model, it is relatively simple to implement. In addition, it has the Permission Filtering Policy (PFP) component, which is the attribute filtering component of the model that may be implemented using the above mentioned tree.

The Policy RC-ABAC is based as well on the above mentioned tree, with the addition of the required/mandatory root and first 2 levels of the tree (the 'OR' root, the AND gates 1st level, and the Role and attribute subtree 2nd level). These limitations are relatively not complicated to implement therefore.

The Dynamic Role model has the most complicated additional component which is the dynamic role procedure. This procedure most carefully create a structure of attribute based dynamic roles and their associated privileges according to relevant policies, common use and TTL (Time To Live) properties.

We can see that in the implementation metric, there is a strong disadvantage in the implementation of the Dynamic Role model compared to the other three models.

D. Maintenance

Attribute Centric and Policy RC-ABAC are both mainly based on attributes, and therefore are easy to maintain since attributes do not have a rigid association structure, and many attributes can describe one object (a non-injective function).

Maintenance of the Role Centric model requires more effort, since the addition of privileges and addition of users require the addition of Roles. In order to prevent "role explosion" and "database garbage" we must constantly maintain our database with the most recent and updated role structure.

In the Dynamic Role model, we must maintain the relevant "TTL" fields of the different dynamic role categories for the same reason mentioned above for prevention of role explosion. If the TTL fields have a high value, we have an accumulation on dynamic roles which decrease efficiency, on the other hand if the TTL fields have a value to low, we have a high update rate which load the system.

We can see that in the maintenance metric, there is a disadvantage in the maintenance time cost of the Dynamic Role and the Role Centric models compared to the other 2 models.

E. Other Comparisons

Some of the Hybrid Models suggested in the literature are double level based such as the Attribute Based Policies RBAC

TABLE I. COMPARISON OF DIFFERENT ACCESS MODELS

CHARACTERISTIC	RBAC	ABAC	ABP-RBAC	RABAC	BLAC	POLICY RC-ABAC
FLEXIBILITY	Low	High	Low	Low	High	High
GRANULARITY	Low	High	Low	High	High	High
AUTHORIZATION COMPLEXITY	Static	Dynamic	Static	Dynamic	Dynamic	Dynamic
PRIVILEGE MODIFIABILITY	Simple	Complex	Simple	Simple	Complex	Simple
PERMISSION MODIFIABILITY	Simple	Complex	Simple	Simple	Complex	Simple
REVOCABILITY	Simple	Complex	Simple	Simple	Simple	Simple
PERMISSION REVIEWABILITY	Simple	Complex	Simple	Simple	Simple	Simple
SETUP COMPLEXITY	Complex	Simple	Simple	Complex	Simple	Complex

model (ABP-RBAC) and the BiLayer Access Control Model (BLAC). These double level models are able to optimize the flaws of the main hybrid models by splitting the access process to different parts. In the BLAC model [8], the authors compared different access control models according to different metrics than we have suggested. In regards to that comparison shown in Table I, the Policy RC-ABAC model still performs better than the rest of the models in most metrics.

In a general analogy, we can include Privilege Modifiability, Permission Modifiability and Revocability characteristics in our "Policy Design" metric. The Flexibility and Granularity characteristics are included both in our Flexibility metric. Permission Reviewability is equivalent to our auditability metric and setup complexity is partially included in our implementation metric. Part of the Setup complexity characteristic includes the initial policy design and the role engineering. This is a procedure that must be addressed in any model, and has a variety of optimized solutions that are not addressed in Access Control models [10]. Therefore, the actual complexity of the role centric models can be drastically reduced, and even referred to as simple by using the correct role engineering model. We have not address the Authorization Complexity characteristic, but since our model is attribute based, then obviously the authorization is dynamic.

V. IMPLEMENTATION USING ATTRIBUTE BASED ENCRYPTION

This access control model can also be implemented using Ciphertext-Policy Attribute-Based Encryption (CP-ABE) as presented in [12]. Since the model is based on an access tree as suggested in the CP-ABE scheme, all that is needed to complete the model is to implement the mandatory tree structure and modify the decryption algorithm to perform a "post-order tree-walk" so the performance improvements is applied.

```
If Tree_root is not OR then
    Throw exception
For each k=child of Tree_root
    If k is not AND then
        Throw exception
    Else if k.left_child is not of type ROLE then
        Throw exception
```

Assuming we receive the tree in the mandatory structure we required, to decrypt the tree we use the next code segment.

```
DecryptTree(CT, SK, Tree_root)
    For each k=child of Tree_root
        If k.left_child matches the user role than
            Return Polynomial interpolation of
            k.left_child with DecryptNode(CT,
            SK, k.right_child)
    Return NIL
```

DecryptNode(CT, SK, Node) as defined in the [12].

VI. CONCLUSIONS

We have presented the Policy RC-ABAC model as an improved solution to the flexibility and auditability combination problem of access control hybrid models. This model does not provide a complete solution to the auditability efficiency problem of ABAC, but it does present an efficiency improvement that allows us to maintain the fine grain flexibility need for access control of internet based resources. In future works, we will try to optimize the policy design.

References

[1] S. G. Weber, Designing a Hybrid Attribute-Based Encryption Scheme Supporting Dynamic Attributes

[2] J. Huang, D. M. Nicol, R. Bobba and J. H. Huh, *A Framework Integrating Attribute-based Policies into RBAC,* Urbana-Champaigen, IL, USA, 2012

[3] J. R. Muhlbacher, C. Praher, *"DS RBAC – Dynamic Sessions in Role Based Access Control"* in Journal of Universal Computer Science, vol. 15, no. 3, 2009

[4] D. R. Kuhn, E. J. Coyne and T.R. Weil, "Adding Attributes to Role-Based Access Control" in *IEEE Computer*, vol. 43, no. 6 (June, 2010) , pp. 79-81

[5] X. Jin, R, Sandhu and R. Krishnan, *"RABAC: Role-Centric Attribute-Based Access Control",* Computer Network Security, LNCS, Volume 7531, 2012, pp 84-96.

[6] X. Jin, R, Sandhu and R. Krishnan, *"A Role-Based Administration Model for Attributes",*SRAS '12 Proceedings of the First International Workshop on Secure and Resilient Architectures and Systems, PP 7-12, 2012, USA.

[7] S. Verma, S. Kumar, M. Singh, "Hybrid Access Control Model in Semantic Web" in *International Journal of Information Technology (IJIT), Volume – 1, Issue – 1,* August 2012.

[8] S. Alshehri and R.K. Raj, "Secure Access Control for Health Information Sharing Systems", Rochester, NY, May 2013.

[9] R. Sandhu, D. Ferraiolo and R. Kuhn. "The NIST Model for Role-Based Access Control: Towards A Unified Standard".

[10] A. Colantonio, R.D. Pietro, L. Chierchia "Role Mining Techniques To Improve RBAC Administration" Rome, 2011.

[11] empowerID, "Best Practices in Enterprise Authorization: The RBAC/ABAC Hybrid Approach".

[12] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption", Proceedings of the 2007 IEEE Symposium on Security and Privacy, PP 321-334, IEEE Computer Society, Washington DC, USA, 2007.

[13] J. Zhu and W. Smari, "Attribute Based Access Control and Security for Collaboration Environments", IEEE National Aerospace and Electronics Conference, 2008. NAECON 2008, PP 31-35.

[14] D.A. Reed, D.B. Gannon, and J.R. Larus, "Imagining the Future: Thoughts on Computing," Computer, vol. 45, no. 1, pp. 25-30, jan. 2012.