ISSN 2395-1621

Separation of Duty Constraints in Attribute-Based Access Control

[#1]Pranali Prabhakar Bhusare, [#2]Prof.Rakesh Shirsath

[1]bpranali05@gmail.com
[2]rakesh.shirsath@gmail.com

[#1]PG Student Computer Engineering Department, SPPU
SITRC, Nahik-422213, India
[#2]Assistant Professor, SPPU Computer Engineering Department
SITRC, Nahik-422213, India

ABSTRACT

Recently, attribute based access control (ABAC)has received considerable attention from the security community for its policy flexibility and dynamic decision making capabilities. The two main primary parameter related to a constraint are its specification and enforcement. From the different types of constraints ,enforcement of the Separation of Duty (SoD) constraint is considered to be the most important in commercial applications. We introduce the problem of SoD specification, verification, and enforcement in attribute-based access control(ABAC) systems. We then demonstrate the effect of modifications in the different components of ABAC on enforcement. As an contribution we propose an feasible fuzzy-extended ABAC (FBAC) technique to improve the fexibility in urgent exceptional authorizations and thereby improving the resource usability and business timeliness. We use the fuzzy assessment mechanism to evaluate the policy-matching degrees of the requests that do not comply with policies, so that the system can make special approval decisions accordingly to achieve unattended exceptional authorizations.

Keywords: Attribute based access control, separation of duty, mutually exclusive rules, enforcement, verification, Hierarchical attributes, FBAC

ARTICLE INFO

Article History

Received: 16th June 2019

Received in revised form :

16th June 2019

Accepted: 18th June 2019

Published online :

21st June 2019

I. INTRODUCTION

Over the last decades, attribute based access control (ABAC) has been developed as a flexible form of access control due to its policy-neutral nature (that is, an ability to express different kinds of access control policies including DAC, MAC and RBAC) and dynamic decision making capabilities. Generally ABAC regulates permissions of users or subjects to access system resources dynamically based on associated authorization rules with a particular permission. A user is able to exercise permission on an object if the attributes of the user's subject and the object have a configuration satisfying the authorization rule specified for that permission. Hence, proper attribute assignment to these entities is crucially important in an ABAC system for preventing unintended accesses. Introduce the problem of SoD verification in ABAC systems. The ABAC model suggested by NIST in [4]is used as the reference model. We study different scenarios under which SoD verification might become necessary and categorize the identified scenarios into different classes of problems. While the problem of SoD verification has been examined in the context of RBAC [7], it has not been extensively studied for ABAC except for [8], which introduces the problem. As discussed in detail later, the methods proposed for RBAC cannot be easily extended for efficiently solving SoD verification in ABAC. To summarize, our contributions in this paper are as follows:

- We give a precise definition of SoD and SoD verification problems in ABAC.
- We show that although directly verifying SoD is intractable, verification through mutually exclusive authorization rules can be done efficiently.
- We identify four different classes of SoD enforcement problems and provide approaches for solving them.

II. RELATEDWORK

A. Attribute Based Access Control

There is a sizable literature on ABAC in general. Damiani et al [8] described a framework for ABAC in open environments. Wang et al [9] proposed a framework that models an ABAC system using logic programming with set constraints of a computable set theory. The Flexible access control system [10] can specify various ABAC policies and provide a language that permits the specification of general constraints on authorizations.

B. Constraints:

Several authors have focused on issues in constraints specification in access control systems primarily in RBAC. Constraints in RBAC are often characterized as static separation of duty(SSOD) and dynamic separation of duty (DSOD). These two issues date back to the late 1980's.A number of attempts initiated afterwards to identify numerous forms of SSOD and DSOD policies and to specify them formally in RBAC systems.

C. Separation of Duty

Separation of Duty (SoD) is a security principle having the primary objective of ensuring that no single user is capable of executing all the steps involved in a critical task. The underlying concept behind this principle is that, the likelihood of a single person perpetrating a fraud is higher than that of a group of people colluding to do so. To achieve separation of duty with respect to a task, the set of permissions associated with the task is partitioned among multiple users. A k-n SoD (k-out of-n Separation of Duty) policy, which is a generalization of the above statement, states that, not less than k users together should get all the n permissions required to perform a task. This definition of SoD is valid irrespective of the underlying access control model. Two different approaches could be used to enforce separation of duty: Static SoD (SSoD) and Dynamic SoD (DSoD)

III. LITERATURE SURVEY

In the past few years, several attribute based access control models have been proposed. Jin et al. [5] present a generalized and formalized ABAC model that can be used to configure access control policies in DAC, MAC and RBAC. It establishes a formal connection between these three classical models and ABAC. Riad et al. [16] discuss an ABAC model that supports attribute-rules for access control in cloud environment. An attribute-rule specifies an agreement that determines what kind of attributes should be used and the number of attributes considered for making access decisions. Qi et al. [17] present a framework that combines role and attribute to design a distributed access control architecture. Jin et al. [18] introduce a role-centric attribute based access control model named RABAC that constraints the available set of permissions based on the user and object attributes. Servos and Osborn [19] propose a formal hierarchical model of ABAC named as HGABAC. It includes attribute inheritance through user and object groups as well as environment, connection and administrative attributes. Chatterjee et al. [20] present an attribute based access control scheme in which access control structures are defined using access tree made up of logic expressions over the attributes. Reference [4] provide a comprehensive definition for the ABAC model. Very recently, Biswas et al. presented an ABAC model named LaBAC(Label-Based Access Control) in [21]. The discussed model isa policy enumeration model, which uses one user attribute and one object attribute for authorization policy enumeration. Hsuand Ray [22] present a location aware ABAC model that canbe used to detect security violations in online social networks. In recent years, a few approaches for mining attribute based policies have also been proposed. Xu and Stoller [23], [24]present methodologies for mining ABAC policies from RBAC policies. Medvet et al. [25]. propose a multi-objective evolutionary approach for policy mining. It aims at learning a policy consistent with the input requests and does not use those attributes which uniquely represent user and resource identities and hence, exploits the true potential of the ABAC paradigm. Benkaouz and Freisleben [26] present a KNN based approach for classification and clustering of policies. The discussed approach aims to reduce the dimensionality of ABAC policies for large applications.

IV. PROBLEM DEFINITION AND MOTIVATION

To comply with organizational business requirements, authorization rules often need to be constrained. These constraints are specified in terms of policies. The access control model chosen to be deployed in the organization essentially forms the basis for putting these policies in place. The methods proposed for RBAC cannot be easily extended for efficiently solving SoD verification in ABAC. To summarize,

- Give a precise definition of SoD and SoD verification problems in ABAC.
- Show that although directly verifying SoD is intractable, verification through mutually exclusive authorization rules can be done efficiently.
- To identify four different classes of SoD enforcement problems and provide approaches for solving them

V. PROPOSED SYSTEM

Following are the details of the proposed work. Initially basic modules of the system are mentioned and later their detail working is explained.

- We will provide a precise definition of SoD and SoD verification problems in ABAC.
- We show that although directly verifying SoD is intractable, verification through mutually exclusive authorization rules can be done efficiently.
- We identify four different classes of SoD enforcement problems and provide approaches for solving them.
- FBAC Model. Te FBAC model wraps the standard ABAC as a preliminary screening module and integrates additional decision support components for improving the resource usability, thereby gaining better business timeliness.

A. Modules:

- User and Resources
- Check Candidate Constraint
- SOD Verification
- MER

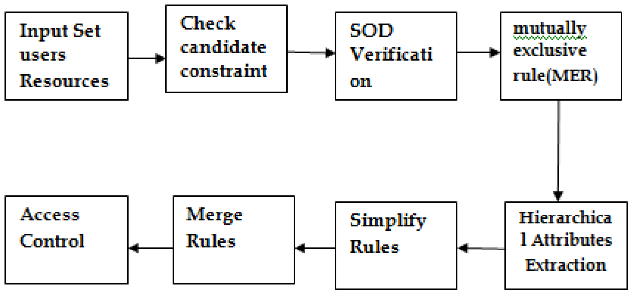
Input Set users Resources
Check candidate constraint
SOD Verificati on
mutually exclusive rule(MER)
Hierarchica l Attributes Extraction
Simplify Rules
Merge Rules
Access Control

Fig.1. System Architecture

A. User and Resources:

Users (U) and Objects (O): Represent a set of authorized users and a set of objects, respectively. Members of these sets are denoted as xi.

B. Check Candidate Constraint:

ABAC based on logic programming where policies are specified as stratified constraint flounder-free logic programs that admit primitive recursion". While their framework introduces hierarchical attributes something lacking from other models), it is largely focused on the representation, consistency and performance of attribute-based policies and their evaluation over providing a workable model of ABAC.

C. SOD Verification:

Separation of Duty (SoD) is a security principle having the primary objective of ensuring that no single user is capable of executing all the steps involved in a critical task. The underlying concept behind this principle is that, the likelihood of a single person perpetrating a fraud is higher than that of a group of people colluding to do so. To achieve separation of duty with respect to a task, the set of permissions associated with the task is partitioned among multiple users.

D. MER:

Several sets of MERs can be generated. Different sets of MER constraints put different levels of restrictiveness on the valid sets of users. A method for generating a t-m MER constraint from a k-n SoR. It can be observed that, while an ABAC-VF-SoD instance takes input in the form of SoD tuples, an ABAC-VF-MER instance takes input in the form of authorization rules. Therefore, one needs to reduce, a step that may or may not be required to be performed on-line, an instance of ABAC-VF-SoD to an instance of ABAC-VF-MER (otherwise, ABAC-VF-SoD would not be in coNP-complete, as already proved). To do so, an instance of ABAC-VF-SoD is initially transformed into an intermediate form, which is based on authorization rules and then from that intermediate form, it is transformed into an instance of ABAC-VF-MER

E. FBAC Model

The FBAC model wraps the standard ABAC as a preliminary screening module and integrates additional decision support components for improving the resource usability, thereby gaining better business timeliness.

B. Algorithm : FBAC Decision-Making Procedure

Input: qi, Cx

Output: Decision {granted, denied}

```
I.    If match any policy then
II.   Return granted
III.  End if
IV.   $\mu(qi) \leftarrow$ max i=1n]Vi(qi))
V.    Cost(qi) $\leftarrow 1 - \mu(qi)$
VI.   If $\mu(qi) < H$ or Cx <Cost(qi) then
VII.  Return denied
VIII. End if
IX.   Cx $\leftarrow$ (Cx $-$ Cost(qi))
X.    Return granted
```

Notations	Definitions
qi	The ith request.
Cx	To credit value of the subject x(a rational number in (0, cmax))
μ(qi)	To fuzzy membership function for calculating the membership degree of the qi to the policies
H	To rejection threshold

Table 1: Te major notations and definitions

VI.EXPERIMENTALRESULTS

For results we show only the effect of variation in the number of users and the number of rules on the execution time required for solving each of the above three categories of problems. This is because, number of authorization rules has the most significant impact on the solving time of an instance of ABAC-VF-MER as well as GenSoR. we present experiments aimed at evaluating the performance of our ABCL enforcement algorithm during user attribute assignment (discussed in section IV- 4). The experiments were conducted on a machine having the following configuration: 2.40GHz with 2GB RAM running a Windows 7 enterprise OS and .Net Framework 4.0

Dataset Used:

We can use companies database or dataset for this system. employees information or database will be used to maintain security for access control.

The effect of increase in the number of users on the time required to solve the third category of SoD enforcement problems is shown in Figure 2. below.

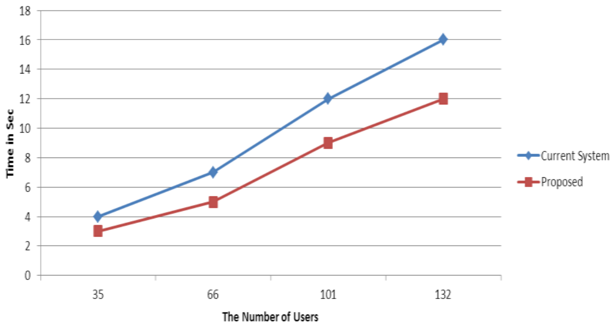

Fig. 2 Resulted Graph

No	Current System	Proposed
35	4	3
66	7	5
101	12	9
132	16	12

Table 2: Result table

VII. CONCLUSION

In this paper, we proposed first Separation of Duty constraints in terms of the components of the Attribute-based Access Control model. Next, the SoD verification problem has been introduced in the context of ABAC. It has been shown that, while SoD verification is intractable, a part of it can be efficiently solved using the principle of mutual exclusion. The remaining part can be processed through offline processing.

ACKNOWLEDGMENT

First and foremost, I would like to thanks my Head of department, Prof. Amol Potgantwar for his guidance and support. I will forever remain grateful for the constant support and guidance extended by guide Prof. Rakesh Shirsath to guide me for making proposed system, that is going to be implemented during my post-graduation ,I wish to express my sincere thanks to PG Coordinator and the departmental staff members for their support. Lastly I would like to thank Department of Computer College name for making all the requirements possible and simply available whenever required.

REFRENCES

[1] Sadhana Jha, Shamik Sural, Vijayalakshmi Atluri, and Jaideep Vaidya , “Specification and Verification of Separation of Duty Constraints in Attribute-Based Access Control” IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 13, NO. 4, APRIL 2018.

[2] Y. Benkaouz, M. Erradi, and B. Freisleben, “Work in progress: K-nearest neighbors techniques for ABAC policies clustering,” in Proc. ACM Int Workshop Attribute Based Access Control, 2016, pp. 72–75.

[3] Datasets for Policy Mining. Accessed: Sep. 1, 2016.[Online].Availablehttp://www.cs.stonybrook.edu/~stoller/software/ABAC-Mining.zip

[4] P. Biswas, R. Sandhu, and R. Krishnan, “Label-based access control:An ABAC model with enumerated authorization policy,” in Proc. ACM Workshop Attribute Based Access Control, 2016, pp. 1–12

[5] C. Hsu and I. Ray, “Specification and enforcement of location-awareattribute-based access control for online social networks,” in Proc. ACM Int. Workshop Attribute Based Access Control, 2016, pp. 25–34.

[6] K. Riad, Z. Yan, H. Hu, and G.-J.Ahn, “AR-ABAC: A new attribute based access control model supporting attribute-rules for cloud computing,” in Proc. IEEE Conf. Collaboration Internet Comput., Oct. 2015,pp. 28–35.

[7] H. Qi, H. Ma, J. Li, and X. Di, “Access control model based on role and attribute and its applications on space-ground integration networks,” in Proc. 4th Int. Conf. Comput. Sci. Netw. Technol., Dec. 2015,pp. 1118–1122.

[8] Z. Xu and S. D. Stoller, “Mining attribute-based access control policies,”IEEE Trans. Depend. Sec. Comput., vol. 12, no. 5, pp. 533–545, Sep./Oct. 2015.

[9] Medvet, A. Bartoli, B. Carminati, and E. Ferrari, “Evolutionary inference of attribute-based access control policies,” in Evolutionary Multi-Criterion Optimization. Cham, Switzerland: Springer, 2015,pp. 351–365. [Online]. Available: http://www.springer.com/us/book/9783319159331

[10] V. C. Hu et al., “Guide to attribute based access control (ABAC)definition and considerations,” Dept. Comput. Secur. Division, Nat. Inst.Standards Technol. (NIST), Gaithersburg, MD, USA, Tech. Rep., 2014.

[11] [Online].Available:https://csrc.nist.gov/publications/detail/sp/800-162/final

[12] Servos and S. L. Osborn, “HGABAC: Towards a formal model of hierarchical attribute-based access control,” in Proc. Int. Symp. Found.Pract.Secur., 2014, pp. 187–204.

[13] S. Chatterjee, A. K. Gupta, V. K. Mahor, and T. Sarmah, “An efficient fine grained access control scheme based on attributes for enterprise class applications,” in Proc. Int. Conf. Signal Propag. Comput.Technol.,Jul. 2014, pp. 273–278.

[14] Z. Xu and S. D. Stoller, “Mining attribute-based access control policies from RBAC policies,” in Proc. 10th Int. Conf. Expo Emerg. Technol. Smarter World, Oct. 2013, pp. 1–6.

[15] Jin, R. Krishnan, and R. R. Sandhu, “A unified attribute-based accesscontrol model covering DAC, MAC and RBAC,” in Proc. 26th Annu.IFIP WG Conf. Data Appl. Secur. Privacy XXVI, 2012, pp. 41–55.

[16] N. Li, M. V. Tripunitara, and Z. Bizri, “On mutually exclusive roles andseparation-of-duty,” ACM Trans. Inf. Syst. Secur., vol. 10, no. 2, 2007,Art. no. 5.

[17] P. Gomes, H. Kautz, A. Sabharwal, and B. Selman, Satisfiability Solvers. Amsterdam, The Netherlands: Elsevier, 2008, pp. 89–134

[18] Damiani, S. D. C. Di Vimercati, and P. Samarati. New paradigms for access control in openenvironments. In Proc.ofthe ISSPIT,2005

[19] L. Wang, D. Wijesekera, and S. Jajodia. A logicbasedramework for attribute based access control. In Proc. of the ACM FMSE, 2004.

[20] S. Jajodia et al. Flexible support for multipleaccess control policies. ACM TODS, 26(2):214–260, 2001.

[21] M. A. Harrison, R. L. Walter, and J. D. Ullman, “Protection in operating systems,” Commun. ACM, vol. 19, no. 8, pp. 461–471, 1976.

[22] S. Osborn, “Mandatory access control and role-based access controlrevisited,” in Proc. 2nd ACM Workshop Role-Based Access Control,1997, pp. 31–40.

[23] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman,“Role-based access control models,” IEEE Computer, vol. 29, no. 2,pp. 38–47, Feb. 1999.

[24] N. Creignou and M. Hermann, “Complexity of generalized satisfiability counting problems,” Inf. Comput., vol. 125, no. 1, pp. 1–12, 1996.

[25] R. J. Bayardo, Jr., and R. Schrag, “Using CSP look-back techniques to solve real-world SAT instances,” in Proc. Conf. Innov. Appl. Artif.Intell., 1997, pp. 203–208

[26] D. D. Clark and D. R. Wilson, “A comparison of commercial andmilitary computer security policies,” in Proc. IEEE Symp. Secur.Privacy (TSA), Apr. 1987, pp. 184–194.

[27] D. J. A. Welsh, Knots, Colourings and Countings. Cambridge, U.K.:Cambridge Univ. Press, 1993.