IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering
(IJERCSE)
Vol 3, Issue 12, December 2016

Survey on Access Control Policies in Mining

[1] Ajinkya Kalaskar [2] Vaishali Barkade
[1][2] Dept. of Computer Engineering
Rajashri Shahu College of Engineering,
Savitribai Phule Pune University, Pune, India.

***Abstract :--* Access control is the process of maintaining every user request and allow to access the resources and data manage by a system. It also decide whether to grant or denied the requester. The decision of access control is depend on the rules and regulation applied by security policies. Different access control policies corresponds to the different criteria such as what is allowed and what is not allowed with defining security means. The two main challenging problems of access control policies are: inconsistencies and incompleteness. To detect and resolve such problems, various access control policy validation techniques are proposed by the researchers. In this survey we identify the basic concepts of access control policies and different security requirements. We studied and compare various policies and models based on their advantages and limitations.**

***Key Words:* -- Access control; Usability; Security; Metrics; Formal logic, data mining.**

I. INTRODUCTION

The efficient administration of employees’ access to sensitive applications and data is one of the biggest security challenges for today’s organizations. Typically, large organizations manage millions of user access privileges across thousands of IT resources. Due to ineffective and application-specific user management, employees accumulate excessive access rights over time. As a consequence, most users are overprivileged, meaning they are assigned more permissions than necessary to perform their work. At the same time, organizational guidelines and policies can hardly be enforced in a decentralized environment. As a result, organizations implement a company-wide identity and access management (IAM) system for the centralized management of digital identities. This enables organizations to implement standardized user lifecycle processes, reduce security vulnerabilities and comply with existing national and international regulations..

In general, typical IAM systems are built on three pillars: processes, technologies and policies. Core identity lifecycle processes like user provisioning or access privilege management are implemented using available automation technologies. Existing products offer a variety of functionalities like identity directories for data storage, provisioning engines for user management or workflow capabilities. Both processes and technologies are controlled by a set of company-specific policies. These policies control technological aspects like data synchronization or data storage. At the same time, they are responsible for process-related aspects like access privilege management, provisioning processes, and security management within the IAM.

While property verification is useful to detect faults, in practice, most policies are not equipped with properties. In addition, manually writing properties is not a trivial task for two reasons. First, the policy authors must have sufficient domain knowledge of a given policy to identify properties for the policy. Second, as the size of a policy increases and the structure of a policy becomes complex, identifying properties is more challenging.

While available systems offer a variety of technologies and functionalities for implementing user management processes, policies have received little attention among researchers and practitioners so far. Policy management commonly still needs to be carried out manually by IT administrators with hardly any means for structured policy definition or ongoing policy management being available. Moreover, only static data is employed (e.g. department of an employee), letting valuable data lie fallow. As a result, only a small number of basic policies are defined and implemented in practice. These policies are commonly extracted from partly documented internal regulations and requirements and remain unchanged during system operation. This results in a situation where policies outdate over time, leading to security vulnerabilities, essentially reducing the advantages of a centralized user management. Consecutively, it is mandatory that policies evolve over time in order to reflect organizational and technological changes within a company.

II. LITERATURE REVIEW

This paper [1] presents an ABAC approach mining algorithm. To the best of their insight, it is the primary ABAC policy mining algorithm. Their algorithm iterates over tuples in the given user-permission relation, utilizes chose tuples as seeds for developing candidate rules, and attempts to generalize every candidate rule to cover extra tuples in the user-permission relation by replacing conjuncts in attribute expressions with requirements.

In this paper [2], we formally define the difficulties that clients face while generating usable access control rule sets and give formal tools to handle them all the more effectively. We began our research with a pilot study in which specialists were interviewed. The goal was to list ease of use difficulties in regards to the administration of access control rule sets and check how those difficulties were handled by specialists.

In this article [3], authors characterize the thought of weighted structural complexity measure and propose a role mining algorithm that mines RBAC frameworks with low structural complexity. Another key issue that has not been sufficiently addressed by existing role mining methodologies is the way to find roles with semantic meanings. In this article, they concentrate on the issue in two essential settings with various data accessibility. At the point when the main data is user-permission relation, they propose to find roles whose semantic meaning depends on formal concept lattices.

The paper [4] examines role mining with noisy input data and recommends separating the issue into two stages: noise removal also, candidate role generation. We introduce a methodology with use (non-binary) rank decreased matrix factorization to recognize noise and experimentally demonstrate that it is effective at recognizing noise in access control information. Client and permission attributes can further be utilized to enhance accuracy.

In this paper [5], authors present a comprehensive structure for assessing role mining algorithms. Authors classify role mining algorithms into two classes based on their outputs; Class 1 algorithms output a sequence of prioritized roles while Class 2 algorithms output complete RBAC states.

This paper [6] proposes new algorithms for role mining. The algorithms can easily be used to optimize a variety of policy quality metrics, including metrics based on policy size, metrics based on interpretability of the roles with respect to client attribute data, and compound metrics that consider size and interpretability. The algorithms all start with a stage that builds a set of applicant roles.

This paper [7] characterizes a parameterized RBAC (PRBAC) structure in which clients and permissions have attributes that are implicit parameters of roles and can be utilized in role definitions. Parameterization essentially upgrades the scalability of RBAC, by permitting considerably more concise policies. This paper presents algorithms for mining such policies and reports the results of assessing the algorithms on contextual analyses.

In this paper [8], authors give statistical measures to investigate the relevance of various types of business data for characterizing roles. Authors then present a methodology that incorporates relevant business data into a probabilistic model with an associated algorithm for hybrid role mining.

Table 1. Survey Table

Sr. No.	Title	Paper Details	Method Used	Advantages	Disadvantages
1.	Mining Attribute-based Access Control Policies	iterates over tuples in the given user-permission relation	ABAC approach mining algorithm	reduce the cost of migration to ABAC	Does not suppor additional ABAC policy language features
2.	Formal definitions for usable access control rule sets—From goals to metrics	introduced security and usability metrics	Deny precedence etc.	can be used as optimization criteria to generate usable access control rule sets and to improve their manage-ability	No tool present that can be integrated in the daily working environment
3.	Mining roles with multiple objectives	weighted structural complexity measure	role mining algorithm	balance the semantic guarantee of roles with system complexity	---
4.	Mining roles with noisy data	role mining with noisy input data	noise removal also, candidate role generation	method performs better than the approach of mining noisy data directly	Can be improved for predicting missing values in any access control matrix
5.	Evaluating role mining algorithms	comprehensive structure for assessing role mining algorithms	role mining algorithms	Can evaluate role mining algorithms	---
6.	Algorithms for mining meaningful roles	role mining algorithm, the elimination algorithm	policy quality metrics.	easily be used to optimize a variety of policy quality metrics	Scalability can be improved
7.	Mining parameterized role-based policies	parameterized RBAC (PRBAC)	RBAC	can significantly reduce the cost of migration to RBAC	---
8.	A probabilistic approach to hybrid role mining	divided the hybrid role mining problem into two parts and provided solutions for them:	hybrid role mining algorithm.	increased security	Can extend given approach for analyzing and merging two given RBAC systems

III. PROPOSE SYSTEM

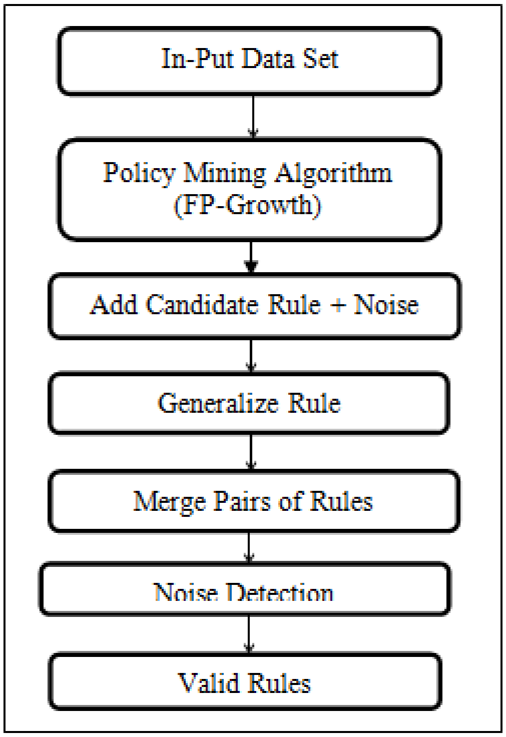

Fig 1. Propose System

This system used the synthetic attributes dataset as a input. The synthetic attributes dataset contain the number of departments for the university and project management sample policies, and the number of wards for the health care sample policy. Next module is policy mining algorithm where FP-Growth algorithm is used to genrate the rules. This rules are unique and also reduce the memory consumption and increase the performance of the system. It iterates over tuples in the given userpermission relation, uses selected tuples as seeds for constructing candidate rules, and attempts to generalize each candidate rule to cover additional tuples in the user-permission relation by replacing conjuncts in attribute expressions with constraints. After genrating the rules system added the noise on that candidate rule. After constructing candidate rules that together cover the entire userpermission relation, it attempts to improve the policy by merging and simplifying candidate rules. After that system detect the noise and remove the redudant rules and selects the highest-quality candidate rules for inclusion in the generated policy.

IV. CONCLUSION

From this survey, we conclude that the access control policies are mainly grouped into three types such as Discretionary (DAC) policies control access based on the identity of the requestor, Mandatory (MAC) policies control access based on mandated regulations determined by a central authority and Role-based (RBAC) policies control access depending on the roles that users have within the system. On the basis of these catagories various new access control policies has been developed by researchers. Such techniques are comparatively studied in this survey.

REFERENCES

[1] Zhongyuan Xu and Scott D. Stoller, “Mining Attribute-based Access Control Policies”, IEEE Transactions on Dependable and Secure Computing.

[2] M. Beckerle and L. A. Martucci, “Formal definitions for usable access control rule sets—From goals to metrics,” in Proceedings of the Ninth Symposium on Usable Privacy and Security (SOUPS). ACM, 2013, pp. 2:1–2:11.

[3] I. Molloy, H. Chen, T. Li, Q. Wang, N. Li, E. Bertino, S. B. Calo, and J. Lobo, “Mining roles with multiple objectives,” ACM Trans. Inf. Syst. Secur., vol. 13, no. 4, 2010.

[4] I. Molloy, N. Li, Y. A. Qi, J. Lobo, and L. Dickens, “Mining roles with noisy data,” in Proc. 15th ACM Symposium on Access Control Models and Technologies (SACMAT). ACM, 2010, pp. 45–54.

[5] I. Molloy, N. Li, T. Li, Z. Mao, Q. Wang, and J. Lobo, “Evaluating role mining algorithms,” in Proc. 14th ACM Symposium on Access Control Models and Technologies (SACMAT). ACM, 2009, pp. 95–104.

[6] Z. Xu and S. D. Stoller, “Algorithms for mining meaningful roles,” in Proc. 17th ACM Symposium on Access Control Models and Technologies (SACMAT). ACM, 2012, pp. 57–66.

[7] Zhongyuan Xu, Scott D. Stoller, "Mining parameterized role-based policies," in Proc. Third ACM Conference on Data and Application Security and Privacy (CODASPY). ACM, 2013.

[8] M. Frank, A. P. Streich, D. A. Basin, and J. M. Buhmann, "A probabilistic approach to hybrid role mining," in ACM Conference on Computer and Communications Security (CCS). ACM, 2009, pp. 101–111.