About one extended role-based access control formal model

Levon Berberyan
Russian-Armenian(Slavonic) university
Yerevan, Armenia
Email: levon711@mail.ru

ABSTRACT

This paper proposes an extended role-based access-control formal model base with some specific advantages.

KEYWORDS

Information security, role-based access, access control models, NIST RBAC, information technologies

1. INTRODUCTION

The aim of this work is to develop and offer an access control model base with some specific advantages.

Three main access control models and their some modifications have been analyzed in order to show some disadvantages.

Some claims like ease of administration combined with the meeting of some requirements of hierarchy schemas were made to model.

The remainder of the article is organized as follows. Section 2 describes some disadvantages of various access control models. Section 3 formally presents the proposed model base. Section 4 checks whether the model, presented in this paper, meets the requirements of NIST RBAC. Section 5 reviews some advantages of the proposed model. Section 8 concludes the article and outlines future research directions.

2. PRELIMINARIES

Mandatory model [1] does not provide the information exchange tools between subjects of the same level. Second, among the subjects with different levels, there is only one-way communication. Third, there is also a potential problem in the administration of the access system, because it provides only a linear hierarchy providing lack of solutions to practical problems.

With a large number of subjects and objects computing for Discretionary model [2] may become bulky and it also may become more difficult to modify the rules in the key of administration. In order to simplify the administration of discretionary model method of grouping subjects and objects to be accessed can be used, but this approach is beyond the scope of the model itself.

Role based access model [3], unfortunately, also has some disadvantages. Roles in the model are global in nature, so that the rights of the subjects are rights across the system. Thus, having a set of rights, a subject can apply them to any objects in the system, no matter to which types they belong[4]. It turns out that the subject, which should not have to interact with some object, abuses his rights and his behavior can hurt other users and the system as a whole. Another problem is that there is no concept of copyrights. Also, in a role based access model set of operations is presented only for roles, but not for the objects[5].

A common problem for most access control models is the disregard of the time parameters. Very often in real systems it's necessary to be able to give particular individuals the right of access to particular objects for a certain period of time or according to a regular schedule.

3. MODEL OVERVIEW

First of all let's divide the set of all entities into the set of subjects and objects. This separation is useful, because in every type of such interactions these two main types of entities mostly exist.

Generally objects are information resources, which can be represented in various forms, for example, in the form of data files. At the same time subjects are users of information system, which can interact with resources through defined access model.

Subjects can be represented for example by user accounts. Resources can be grouped according to various factors and the same goes with users. Thus, the security of information system is achieved by controlling access of subjects to objects with security policies, which are sets of rules and restrictions.

It was decided to divide the set of objects into entities, named classes and deal with the set of classes **C**. Class in the simplest case is an abstract description of a resource r_i, **i=1,2,...,n**, where **n**-number of currently defined resources. Each class has identifier **Id**, which allows to determine uniquely the object among others. Several types of data (**CD)** can be stored in class' fields. One type is the name of the object, also called **Title**. There is also a type, which represents the time variable **T**, another type of data is the privileges set **P**. Time variable provides greater flexibility to access control model. **T** allows to solve urgent tasks related to role time restrictions.

At a higher level L_j class is an abstract descriptions collection of resources group. Classes can be linked by the relations of inheritance. In this context a class in a higher level contains all the data of the class on a lower level, but only on the same chain of parent-child relations.

$CD(C_x, L_j) \subseteq CD(C_y, L_k)$ where $j<k$ and C_x is in parent-child relations with C_y, according to which C_x is the child of any degree and C_y is the parent of any degree**.**

Classes can be also placed on the same level so as $CD(C_x, L_j) \neq CD(C_y, L_j)$. This disposition allows to reflect relations between users in real groups. Hierarchy of classes can be shown visually using a projection on a tree model [6].

To perform operations with resources the concept of operations buffer is introduced. Operations buffer OB_u **u=1,2,...,q**, where **q** - a number of currently defined buffers, is a console which receives as input actions that must be applied to resource. Actions are nothing more than elementary operations e_z **z=1,2,...,e**, where **e** - a number of currently defined elementary operations on resources with some options. Output of operations buffer shows its response to input. For example ifb jec resource is represented by data file operations buffer can allow such operations that reading with options like start and end point of reading. Instead of reading there can be usb operation of writing or another necessary one.

As resources are represented using classes it is necessary to associate classes with operations buffers. To organize this approach we connect every class at any necessary level of any chain of classes with parent-child relations to operations buffer, which allows limited set of actions **{a1, a2,...,af}**. Thus, class with connected operations buffer $C_i(OB_u)$ can perform limited set of actions on resources described by it and also classes on a higher level on the same chain of parent-child relations have access to that operations buffer with same permissions.

$C_i(OB_u)$ => $C_m(OB_u)$, if $C_j(L_k)< C_m(L_b)$ and C_j is in parent-child relations with C_m, according to which C_j is the child of any degree and C_m is the parent of any degree**.**

Classes can be connected to any number of operations buffers.

Elementary operations can be divided in order to implement separation of duties approach. For example, suppose we have the following set of elementary operations **{read, write}**. Each of these operations can be divided into two ones.

read = grant read access –> read
write = write –> grant write access

Of course, the decomposition can be set in different ways according to the situation and conditions.

The set of subjects is divided into entities, named roles and we deal with the set of roles **R**. There are also privileges related to roles **P**. Privileges define the set of elementary operations e_z, that users are allowed while having particular role with time restrictions. Roles can be with wider privileges and the others with poor rights set.

$$P(R_i) \subseteq P(R_j)$$

In fact, there is one main role which can be divided into simpler roles. In turn, more simple roles can also be divided into more simple ones. For example $R_1=<R_2,R_3>$; $R_2=<R_4, R_5>$; $R_5=<R_6>$. Thus, the set of roles also enters the relations of inheritance. In this case there are levels and roles on the higher level of chain have all privileges of the role in the lower level, but only for the roles on the same chain.

$P(R_x, L_j) \subseteq P(R_y, L_k)$ where $j<k$ and R_x is in parent-child relations with R_y, according to which R_x is the child of any degree and R_y is the parent of any degree

Every user can be assigned to multiple roles and, if it happens, then these roles unite and so privileges do. One role can be assigned to many users, that's why a concept of session with specific logging is introduced.

Sets of classes and roles are being constructed and viewed simultaneously. This happens, because these sets are closely linked. The level and its position among the classes correspond to the level and its position among roles. It means that if the role is on level **H**, has an index **S** among the ones on the same level and a set of privileges **P**, then the user with such a role can use its privileges on the class on the level **H** with the index **S** and privileges **P'**, if $\mathbf{P'} \subseteq \mathbf{P}$. This role according to parent-child relations has access to child classes.

The set of classes contains greater or equal number of elements than the set of roles $|\mathbf{C}| \geq |\mathbf{R}|$. It takes place, because the set of objects as a rule contains more elements, than the set of subjects acquiring an access to them.

Interactions with sets of classes and roles must be logged to specific journals.

4. NIST RBAC REQUIREMENTS OVERVIEW

The NIST RBAC model [7] is a standardized definition of a role-based access control.

Users are assigned to roles which are related to privileges, so being the member of the role user acquires privileges related to it. Within the defined access control model the same user can be assigned to many roles and the same role can be assigned to many users. Obviously, in this context the same rule works for permissions. Thus, NIST requirement, which says that user-role and permission role can be many-to-many, is satisfied.

The requirement for user-role review whereby the roles assigned to a specific user can be determined as well as users assigned to a specific role is complied, because interactions with classes and roles are logged.

Information system users with specified access control model can simultaneously exercise permissions of multiple roles according to the model construction.

This model is supporting the restricted roles hierarchy, which can be visually shown via trees.

According to a requirement of constrained RBAC the enforcing separation of duties takes place. For previously described example the privilege of granting read access can be related to one role and privilege of reading can be related to another role.

Symmetric RBAC requires the roles to which a particular permission is assigned can be determined as well as permissions assigned to a specific role and it is complied due to the presence of logging.

Thus, our model meets the requirements of NIST RBAC.

5. REVIEW OF MODEL'sSOME ADVANTAGES

First, the two main sets may be implemented independently of each other, depending on the particular system. This means the complete independence of each implementation that allows selection of one of the tools for the classes set and the other

for the roles set. In practice, the object set can be much more complex than the set of roles and according to structure of described model it is possible to select a tool, convenient for the construction of such structures and other tool for roles set to build more simple structures, ensuring rapid movement within it. In some cases, it can be necessary to check the data only about the role, whereas in this approach you don't have to consider information about the objects and computing speed for such task will greatly increase.

Second, if an intruder somehow gains access to one of the sets, it is not enough to control the system. As mentioned above, the design of these two sets can use different tools, so that unauthorized access to one of these structures is not enough to get access to the second one. This fact certainly complicates the attacker's actions.

Third, the more simple the structure is, the easier it is to modify and that is the advantage of the administration. In the case of merging two sets into one, we would have had to build two levels for each element: the level of the object and the level of the role. Each layer also may consist of multiple components. Thus, the separation of two sets simplified c eu mechanism for adding, deleting and other type of modifying parts and components.

Fourth, when properly implemented, advantages in computing speed can be achieved. Today, there are many useful tools for the implementation of parallel computing. In this scenario, it became more favorable in terms of speed to implement algorithms in a partially parallel manner, rather than completely consistent, what is expected within the proposed model.

6. MAIN RESULTS

Thus, the aim of the work was achieved: a model base, which is an extended variation of role-based with principles of mandatory and discretionary models was proposed. This model doesn't have some of their disadvantages. The model has an hierarchy, which is typical for the mandatory model, however, there is no principle of linear relations within the whole scheme.

Also it is possible to establish a two-way communication between the subjects with different or the same level in the hierarchy. Each role has its place in the hierarchy, so that it does not need to be global. The system is easy to administrate, because of easier structure.

REFERENCES

[1] Leonard J. LaPadula and Elliott D. Bell, "Secure Computer Systems: A Mathematical Model", MITRE Corporation Technical Report 2547, Volume II(31 May 1973).

[2] "Trusted Computer System Evaluation Criteria", United States Department of Defense(1985), DoD Standard 5200.28–STD.

[3] Ferraiolo D. F., Kuhn D. R., "Role Based Access Control", 15th National Computer Security Conference (October 1992), p. 554–563.

[4] Nyanchama M., Osborn S., "Access Rights Administration in Role-Based Security Systems", in Database Security VIII: Status and Prospects, J.Biskup et al., eds., Elsevier North-Holland, 1994, pp. 37-56.

[5] S.H. von Solms, I. van der Merwe, "The Management of Computer Security Profiles Using a Role Oriented Approach", Computers & Security, Vol. 13, No. 8, 1994, pp. 673-680.

[6] Берберян Л.С., "Разработка и реализация модели информационной системы с возможностью управления доступом", Вестник РАУ, 2012, стр. 42-48.

[7] Sandhu R., Ferraiolo D.F. and Kuhn D.R., "The NIST Model for Role Based Access Control: Toward a Unified Standard", *5th ACM Workshop Role-Based Access Control*: 47—63, July 2000.