Study of Access Control Models

Mohammed ENNAHBAOUI, Said ELHAJJI

***Abstract*—The core of a company is its information system, and the least influential problem leads to major damages, requiring the implementation of a security policy. A logical security policy or, more precisely, the organization of rights is termed Access Control. This latter ensures two fundamental security properties namely: confidentiality and integrity. besides,to model a better policy, we must pass through an implementation of organizational security policy.**

A security model is an access control model. In this paper, we propose a basic study of acces control models by giving a deep description of models, and discussing their reinforcements and weaknesses. This will help us to find the major security problems that exist in information systems, such as the problem of covert channels.

***Index Terms*—security, access control, policy, model, the information system.**

I. Introduction

TODAY all companies and organizations, regardless of their diameters, have their own information system that plays a vital role in all their activities. Thus, any anomaly reaching this system could lead the company to serious damage, in the short term such as a fall in its turnover, or in the long term that influences its commercial image. There are several types of anomalies everywhere, but the one we are interested into in our study are security flaws that are increasing in an exponential way, day after day.

According to the statistics of the Kaspersky company about the third quarter of 2012, more than 30 million vulnerable programs and files were detected, with an average of eight vulnerabilities per user. It is obvious that the information systems security has become a major concern for most companies that has as objective to counter this type of problems and faults found in the information systems. In this sense, to ensure a system security, we propose mechanisms for authentication and authorization, whose the use becomes unavoidable. Authentication concerns identity proof, while authorization is equal to an access control that defines and imposes what is permitted and forbidden to do.

The relation between a security policy and an access control policy is that the first one is the set of rules and practices that ensure the way how sensitive information and other resources are managed, protected and distributed within a specific system [1]. By cons, the second one is a mechanism due to which a system allows or prohibits the actions requested by subjects (active entities) on objects (passive entities) [2]. Then, logical security policy and more specifically the organization of rights is what we call Access Control. With access control, we are interested to ensure two fundamental security properties that are confidentiality and integrity, and to guarantee a better policy, we must pass through modeling or implementing of organizational security policy. To conclude, a security model is an access control model.

In this paper we will try to make a study between the different types of access control models that exist to choose at the end the most suitable model and justify this choice. In the first section we will talk about historical models and their disadvantages. Then, in the second section we will justify the choice of a model with a detailed description. Our purpose is to understand the conception and the approach used in the access control models and try to solve the problem discussed earlier and find the access control model that meets the security needs of all levels (network, system, application) of an information system.

II. Historical models

A. Introduction

The access control policies are defined as high level directives (rules) [3] that specify who (subject) has permission to practice what (action) on which (object) data.

- **Subject**: active entity (user, application, IP address,...).
- **Object**: passive entity that represents the data to be protected (file, relational table, class,...).
- **Action**: represents the action that subject performs on the object (read, write, execute,...).

During this section, we will describe the three most famous models of access control: DAC, MAC and RBAC. Then, a comparison between these models is essential to get hold of the weak points of each.

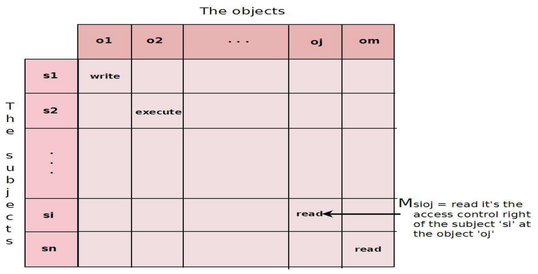
B. The DAC model (Discretionary Access Control)

The Discretionary Access Control model (DAC) [4] allows a subject to assign permissions to other subjects. This access control is flexible, but it can cause errors. The agreement or revocation of privileges is regulated by an administrative policy. The management access to the files of the operating system UNIX is a classic example of access control mechanism based on a discretionary policy. We will present in the following the two well known discretionary models, that are the Lampson model and HRU model (Harrison Ruzzo Ullman model).

1) The Lampson model: The concept of access rights specified by a matrix of access control was introduced in 1971 by Lampson [2]. This model is represented by a triple (S, O, M), where S denotes the subjects, objects O and M = (M_{so}) the access control matrix that associates to each couple (subject s, object o) a set of access rights that are usually: read, write, run.

Manuscript received March 18, 2013; revised March 18, 2013. This work was supported by the laboratory of Mathematics, Computing and Applications (LabMIA), part of the Faculty of sciences in the University of Mohammed V-Agdal of Rabat-Morocco. Study of Access Control Models.

Mohammed ENNAHBAOUI and Said EL HAJJI are members in Laboratory of Mathematics, Computing and Applications (LabMIA), part of the Faculty of sciences in the University of Mohammed V-Agdal of Rabat-Morocco. e-mail: (ennahbaoui.mohamed@gmail.com and elhajji@fsr.ac.ma)



The objects

The subjects	o1	o2	. . .	oj	om
s1	write				
s2		execute			
. . .					
si				read	
sn					read

Msioj = read it's the access control right of the subject 'si' at the object 'oj'

Fig. 1. Matrix of access control

The matrix shown in Figure 1 shows that the right of access is associated with the subject s_i and the object o_j. While the matrix is not fixed yet, it can be updated by the creation of new objects or subjects, by the destruction of the latter as well as the addition or removal of access rights.

There is an ambiguity in the adjective ”discretionary”, which may be understood :

- Either as the fact that rights are organized in a matrix of access control, but without specifying whether it is an authority that defines the rights, or if it is the users who can do.
- Either as is the fact that users themselves can define access rights on the resources they own (HRU model).

2) The HRU model: The Harrison Ruzzo Ullman model (HRU), formalized in 1976 [5] represents an improvement of Lampson model. This model uses a classical access matrix like the Lampson model. The difference lies in that HRU specifies the commands (a set of primitive operations) to assign access rights (read, write, own, etc.), as well as create and delete subjects and objects.In this model, if the right **”own”** is associated with a pair **(s, o)**, the subject s will be considered as the owner of the object o and it may assign its rights of access on the object o to other subjects. In other words, this action allows the subject to define the permissions on the entire column.

The possible primitive operations are: **Enter**: for adding rights, **Delete**: Delete rights, **Create subject**: to create new subjects, **Create object**: to create new objects, **Destroy subject**: the destruction of subjects and **Destroy object**: the destruction of objects.

The commands in the HRU mdel are built from primitive operations above and take as argument subjects and objects. We can add a right r in an access matrix M_{so} if there is a command C that adds the right r in a cell of M_{so}:

c : $M_{so} \longrightarrow M'_{so}$ i.e. $\exists s, o : r \notin M_{so} \wedge r \in M'_{so}$

The HRU command has an optional conditional part as well a body, it has the following format:

```
command c($x_1, ..., x_k$)
        if $a_1$ in $M_{s_1,o_1}$
          ...
          $a_n$ in $M_{s_m,o_m}$
        then $op_1$
            ...
            $op_n$
end
```

With n>0, m≥0, $a_1, ..., a_n$ are authorizations, $op_1, ..., op_n$ are primitive operations. HRU command may not have condition (m=0). We note that despite the fact that we trust users so they follow the policies of the organization, we can not trust the processes running on their behalf, hence the need to distinguish between users and processes that are running for their accounts (subjects).

In the next section, we will show how the MAC models (specifically multilevel model) distinguish between subjects and objects to solve Trojan horses and the information leakage they cause.

C. The MAC model (Mandatory Access Control)

The Mandatory Access Control model (MAC) [6] allows to create obligatory security policies that set essential rules to force compliance of access control requirements. Thus, unlike the DAC model, users can not define the rights of access control, because all objects are the exclusive property of the organization, which implies that in this model the access control policy is managed in a centralized manner.

The Mandatory Access Control [7] is based on the concept of security levels associated with each subject and object, from which are derived the permissions and actions.

A mandatory policy of security, is only a multi-level policy [8] , this latter has the notion of access class. A partial order relation is defined on the set of access classes, it is the dominance relation symbolized by $\geq$.

Each access class has two component:

- **Security Level**: is an element of a totally ordered set, e.g. top secret (TS), secret (S), confidential (C) and unclassified (N) where TS $\geq$ S $\geq$ C $\geq$ N. For objects, security level is called the classification level and for subjects it is called clearance level.
- **A set of categories**: describes the various fields of system in study. For example, in military systems, the categories are nuclear and army, in commercial systems the categories are rather financial, administrative...

Let L be the set of security levels, equipped with the partial order relation $\geq$, and C is the set of categories, equipped with the partial order $\supseteq$. Let l_1, l_2, two levels and c_1, c_2 two categories such as: $l_1 \in L$, $l_2 \in L$, $c_1 \in C$, $c_2 \in C$. Given two access classes ac_1 and ac_2, the dominance relation $\geq$ is defined as: $\forall ac_1 = (l_1, c_1), ac_2 = (l_2, c_2) : ac_1 \geq ac_2 \Longleftrightarrow l_1 \geq l_2 \wedge c_1 \supseteq c_2$

The structure of all access classes forms a trellis that is why multi-level policies are also called by LBAC (Lattice based access control).

To summarize, the multilevel model with all its variations is based on the trellis concept, it also uses an access matrix identical to the HRU model in order to present authorizations on which are added security levels.

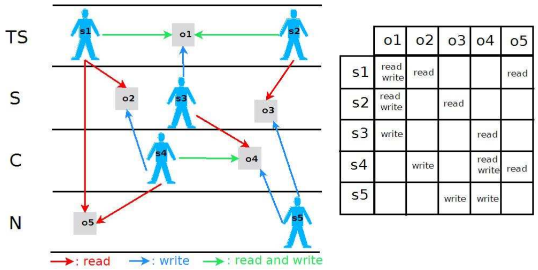
We will then speak of the two most famous models of Mandatory Access Control, that are the Bell-LaPadula model (BLP) which has the purpose to ensure confidentiality, and the Biba model which is interested to integrity.

1) The Bell-LaPadula model (BLP): The Bell-LaPadula model (BLP), developed in 1975 [4], [9] seeks to preserve the confidentiality of the data, that is to say that these latter are only accessible by authorized users. In this model, access rights depend classifications assigned to objects and authorizations granted to subjects, basing on two laws:

1) **Simple property (no read up)**: simply do not read up. In effect, this law prohibits a subject to have a read access to an object that has a higher classification

than the habilitation of the same subject:
$read \in M_{so} \Longrightarrow f(s) \geq f(o)$. $(f : S \cup O \longrightarrow L)$

2) **Star property (no write down)**: simply do not write down (write is used to mean the only writing or addition). In effect, this law prohibits a subject to have a write access to an object that has a classification lower than the habilitation of the same subject:
$write \in M_{so} \Longrightarrow f(o) \geq f(s)$.

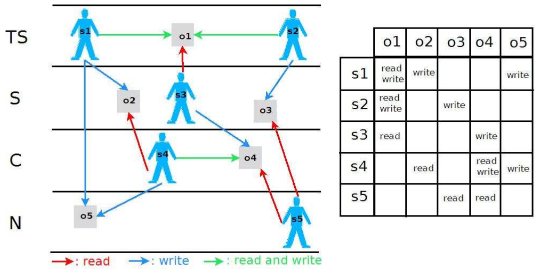


	o1	o2	o3	o4	o5
s1	read write	read			read
s2	read write		read		
s3	write			read	
s4		write		read write	read
s5			write	write	

Fig. 2. Two laws of BLP model

2) The Biba model: The Biba model developed in 1977 [4], [10] focuses on data integrity, what is missing in the BLP mode. Ensuring data integrity means that they can only be changed by authorized users. As in BLP, the Biba model is based on two laws:

1) **Simple property (no read down)**: simply do not read down. In effect, this law prohibits a subject to have a read access to an object that has a classification lower than the habilitation of the same subject:
$read \in M_{so} \Longrightarrow f(o) \geq f(s)$
2) **Star property (no write up)**: simply do not write up (write is used to mean the only writing or addition). In effect, this law prohibits a subject to have a write access to an object that has a higher classification than the habilitation of the same subject:
$write \in M_{so} \Longrightarrow f(s) \geq f(o)$



	o1	o2	o3	o4	o5
s1	read write	write			write
s2	read write		write		
s3	read			write	
s4		read		read write	write
s5			read	read	

Fig. 3. Two laws of Biba model

D. The RBAC model (Role-Based Access Control)

The access control model based on RBAC role is considered as an alternative approach to mandatory access control (MAC) and discretionary access control (DAC). RBAC was proposed for the first time in 1992 by David Ferrailo and Richard Kuhn [11], it aims to facilitate the administration of the access control policy. The core of a RBAC model is the **role**, that represents in an abstract way a function or a trade within an organization that combines the authority and responsibility assigned to a person who plays this role (eg, Professor, Director, Engineer, Technician ...). Each role is associated with permissions (or privileges) that constitute a set of rights corresponding to the tasks that can be performed by each role. Finally and contrary to the models that preceded RBAC, permissions are associated in direct way to the subjects, but through roles. The two relations of the figure below ”Detain(Role, Permission)” and ”Play(Subject, Role)” define precisely the permissions granted to each subject.
A role can have many permissions, and permission may be associated with multiple roles. Similarly, a subject may be a member of multiple roles and vice versa, a role can be performed by several subjects.
Thus, if Dr. Dupont is both surgeon and hospital director, as a surgeon ,he has the access right to medical records, while as a director, he will have access to administrative information.

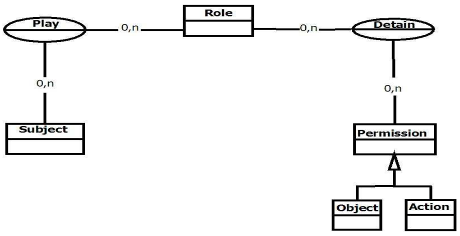


Fig. 4. The two relations of RBAC

As several roles models have been proposed, we will deal with a RBAC family of four models [12]:

1) The $RBAC_0$ model (Core RBAC): It covers the basic criteria included in all RBAC systems. It recognizes seven administrative elements:

TABLE I
ELEMENTS OF $RBAC_0$

Element	Definition
User	refers to the person who interferes with the computer system.
Subject	refers to the process playing on behalf of a user.
Session	it is the entity that represents an active user in the system.
Role	the functions or responsibilities of employees in an organization.
Operation	it is an active process invoked by the subject.
Object	any resource available in a computer system.
Permission	it is an authorization to perform an operation on an object.

This model has two laws to follow:

- **Authorization of a role**: a subject can never have an active role not allowed to him.
- **Authorization to access an object**: a subject s can perform an operation op on an object o only if there is:
  1) a role r belonging to the set of active roles on subject s.
  2) a permission granted to role r that allows it to perform the operation op on the object o.

2) The $RBAC_1$ model (the Hierarchy Role): The motivation to introduce this aspect in RBAC is that within an organization many roles may have several common permissions. As examples to general permissions there are an internal Web site access, the ability to upload documents, etc.These permissions can be granted to all employees or most of them. The inheritance relation of a role creates an authorization form. If the role A inherits from role B, this means that all permissions of role B are allowed by the role A. In other words, the permissions of B constitute a subset of the set of permissions of A, and all users playing the role A can also play the role B. Two types of hierarchy roles have been defined General and Limited. General allows multiple inheritance of permissions. This means that a role can simultaneously have one or more parents (inherits permissions from multiple sources). Limited is defined as the general hierarchy but it does not allow multiple inheritance.

3) The $RBAC_2$ model (the Constraints): The concept of separation of duties (SOD or Separation of Duty) ensures that no person has the ability to control all the steps involved in a high-risk operation, and no user has enough rights to abuse the system alone. Two SOD categories have been described by [13]:

1) **Static Separation of Duties**: no user can have two roles designated as mutually exclusive (conflicting).
2) **Dynamic Separation of Duties**: no user can have during a same session, two roles called mutually exclusive.

We can add to the constraint of separating tasks, another type called the Temporal Constraints [14].

4) The $RBAC_3$ model: The $RBAC_3$ model assembles $RBAC_1$ and $RBAC_2$ models respecting the properties and criteria of each one in order to have a stable security policy, that is complete and easy to administer. The figure below includes the principle operating of $RBAC_3$:

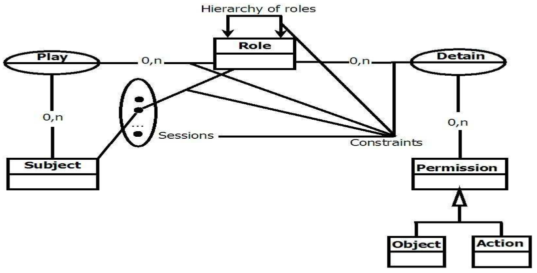


Fig. 5. The $RBAC_3$ model

E. Conclusion

To conclude, we have classified the historical access control models in a comparative table to give an overview of the conception that will be used to create a new generation of access control models.

TABLE II
THE COMPARATIVE TABLE OF THE HISTORICAL ACCESS CONTROL MODELS

Advantages	Disadvantages
DAC	
• Flexible. • Use in environments where the sharing of information is more important than protection.	• Updating the security policy is costly. • Vulnerable to Trojans [3] and to covert channels [15]. • Does not distinguish between users of the subjects. • Does not permit to express prohibitions, recommendations or obligations.
MAC	
• Rigid. • Distinguishes between users and subjects. • Conceived in the environments where the hierarchy of users is more important than sharing information.	• Updating the security policy is costly. • Vulnerable to covert channels. • Does not allow the flow of information between the different levels (problem related to the rigidity). • Does not permit to express prohibitions, recommendations or obligations.
RBAC	
• Updating the security policy is simple which explains the ease of administration policies based on this model. • Encompasses the advantages of the traditional models DAC and MAC. • Applied in complex and distributed areas the fact that this model is based on the concepts of constraints and inheritance.	• Impossible to express the rules depend on the context. • The distinction between the role concept and the group concept is vague. • The absence of generic structure of permissions. • Vulnerable to covert channels. • Does not permit to express prohibitions, recommendations or obligations.

III. THE ORBAC MODEL (ORGANIZATION BASED ACCESS CONTROL)

A. Introduction

The access control model based on organizations OrBAC (Organization Based Access Control) [15] was presented for the first time in 2003. This model aims to solve some problems met in the historical access control models, and to establish an access control policy more abstract. It is interested not only in the permissions, but also prohibitions, obligations and recommendations.

In addition to the role concept for structuring subjects, OrBAC introduces concepts for structuring objects and actions. As its name suggests, the main entity of the model is the **organization**.

An organization may be a structured group of subjects playing certain roles, or entities such as hospital, clinic, emergency department, IT department, ... The fact of introducing this organization concept as a basic element in the access control model solves the problems of RBAC that defines a set of binary relations between the user and role. This means that the user that plays several roles, can activate all roles or a subset of these roles. In practice, even if a user has multiple roles, he does not necessarily have the right to play them. In the OrBAC model, in addition to the notion of organization, seven entities were defined in two different levels to have a correspondence between the elements of each level: the abstract level or organizational (Role, Activity, View) and the concrete level (Subject, Action, Object). The seventh entity **Context** is between the two levels. In what follows we will try to define the entities of OrBAC model and describe the relations between the entities of the two levels. Then we will define the security policy of this model.

B. The entities

1) The subjects and roles: The definition of an entity of subject differs from one model to another. In OrBAC, a subject means either an active entity, i.e a user (John, Pascal, ...), or an organization (accounting department of the company, the university administration, ...). The entity **”role”** is the link between subjects and organizations. In a company developing information systems and software, the roles of developer, tester and project manager are played by users. By cons, the roles of security department and software marketing unit are played by organization. So as subjects play roles in the organization, there is a relation between these three entities, this relation is called ”Allows”: If Org is an organization, s is a subject and r is a role, then Allows(Org, s, r) means that Org allows subject s to play the role r.

Unlike the RBAC model, which considers only the binary relation between subjects and roles, OrBAC model defines a ternary relation between organizations, subjects and roles.

2) The objects and views: In OrBAC, an object is only a passive entity such as files or emails ... . In a company developing information systems and software, the objects can be projects specifications, client folders, personal folders ...

As the roles allow us to structure the subjects and facilitate the updating of the security policy when a new user is added. OrBAC offers the entity **”View”** that structures objects to facilitate the updating of these latter. A view is corresponding to a set of objects that satisfy a common property.

For example, in an administrative file system, the view of ”administrative file” refers to the set of all administrative folders of clients, while the view of ”project file” corresponds to folders of specifications of client projects. A view characterizes the manner how objects are used in the organization. Thus, there is a relation called ”Uses” that binds these three entities: If o is an object, org is an organization, and v is a view, then Uses(org, o, v) means that Org uses the object o in view v. The same view can be defined differently depending on the organization in question.

3) The actions and activities: In OrBAC model, an action includes computer actions such as read, write, send, ...

In order to provide an abstraction of the action entity as it is the case for roles and views, OrBAC model defines an entity that has the name of Activity whose aim is to bring together actions that have a common goal. The activities can be read, modify, transmit, ...

Different organizations may consider that the same action is used to carry out different activities, the relation ”Considered” is used to link the three entities Organization, action and activity as follows: If org is an organization, α is an action and a is an activity, then Considered(org, α, a) means that org considers the action α as part of the activity a.

4) Security Policy: The OrBAC model defines a security policy through permissions, prohibitions, obligations and recommendations that are applied to different entities of this model. As we said at the beginning, there are two levels abstract and concrete and in each one there are permissions, prohibitions, obligations and recommendations. In this paper, we will only define the prohibitions, considering that the same reasoning applies to prohibitions, obligations and recommendations.

5) The permissions and contexts: A permission in OrBAC model is an association that binds the entities: organizations, roles, views and activities as follows: If org is an organization, r is a role, a is an activity and v is a view, then Permission(org, r, a, v) means that the organization org grants to the role r the permission to perform the activity a on the view v. Then, there exists a relation called ”Defines” which combines between contexts, subjects, objects, actions, and organizations such as: If org is an organization, s is a subject, α is an action, o is an object and c is a context, then Defines(org, s, α, o, c) means that within the organization org, the context c is true between the subject s, the object o and the action α.

6) The abstract policy: After defining the context, we return to the Permission relation that will be reformulated in order to add to entities of this relation (roles, views, activities, organizations) the entity context. Let’s assume that if org is an organization, r is a role, a is an activity, v is a view and c is a context, then Permission(org, r, a, v, c) means that the organization org grants to the role r the permission to perform the action a on the view v in the context c. Similarly we can define the relations: Recommendation, Obligation and Prohibition.

7) The concrete policy: Permission is a relation that allows an organization to specify the permissions granted in a given context. As we have seen before, this relation takes place between roles, views and activities. So it is a relation of the abstract level, which is equivalent in concrete level to a relation between the subject, objects, and actions. This relation of low level is called Estpermis. Let’s suppose that s is a subject, α is an action and o is an object, then Ispermitted(s, α, o) means that the subject s has the permission to perform the action on the object o.

In the OrBAC model, the triplets of the relation ”Estpermis” are derived logically from permissions granted to roles, views and activities by the relation ”Permission”. Let’s suppose that org is an organization, r is a role, a is an activity, v is a view, c is a context, s is a subject, α is an action and o is an object, then the OrBAC model has the following axiom:

Permission(org, r, a, v, c) $\wedge$ Allows(org, s, r) $\wedge$ Uses(org, o, v) $\wedge$ Considered(org, α, a) $\wedge$ Defines(org, s, α, o, c) $\Longrightarrow$ Ispermitted(s, α, o).

This means that if the organization org, within the context c, grants the role r the permission to perform the activity a on the view v, if org allows the subject s in the role r, if org uses the object o in the view v, if org considers the action α as part of the activity a, and if within the org the context c is true between s, α and o, then the subject s has the permission to perform the action α on the object o. The same approach used to define the relation Ispermitted is applied to other relations of low level: Isprohibited, Isobligatory and Isrecommended. So it remains to mention the three axioms of these relations:

1) Obligation(org, r, a, v, c) $\Longrightarrow$ Recommendation(org, r, a, v, c): All obligations are also recommendations.
2) Recommendation(org, r, a, v, c) $\Longrightarrow$ Permission(org, r, a, v, c): All recommendations are also permissions.
3) Permission(org, r, a, v, c) $\Longrightarrow$ $\neg$ Prohibition(org, r, a, v, c): A permission implies a non prohibition.

The four axioms that we had just define below represents the security policy of an OrBAC model(at abstract and concrete levels). The diagram below summarizes the OrBAC model. It contains eight entities (Organization, Subject, Role, Object, View, Action, Activity and Context) and twelve relations (Enables, Uses, Considered, Permission, Prohibition, Obligation, Recommendation, Ispermitted, Isprohibited, Isobligatory, Isrecommended and Defines).

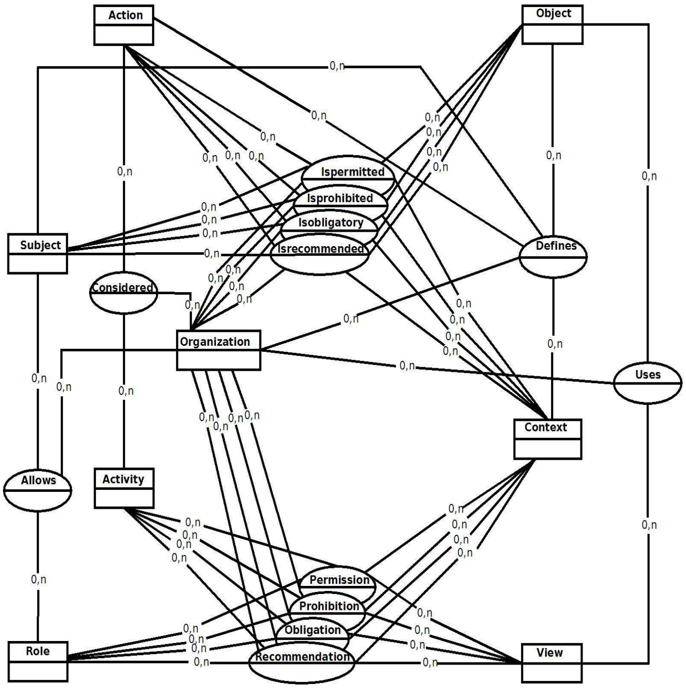


Fig. 6. The OrBAC model and these different components

The OrBAC model introduces also the concepts of hierarchy, constrains and conflicts [15].

C. Conclusion

In this section we have tried to explain the principle of operating of the most advanced access control model of our days.

This model with its recursive definition of the concept of organization, facilitates the representation of a hierarchy of organizations that can cooperate with each other. If we add to the notion of organization, the concept of levels: the organizational level (or abstract) (roles, views and activities) and the concrete level (subjects, objects and actions).

We can conclude that OrBAC is a model that is simple and scalable because it ensures a great ease either in the implementation or in the updating of security policy.

As we have seen, this model has the exclusivity to manage conflicts that arise between the security rules. Since it is the only one that defines the permissions, prohibitions, obligations and recommendations.

OrBAC is a model that tries to assemble the principles of operating of all models that came before it (DAC, MAC, RBAC, ...) in order to have a complete model, and it satisfies the requirements for general security, in particular those of access control, which justifies the fact that this model has all the advantages that are in the comparative table of this paper.

The only drawback remaining this model is the problem of covert channels which resides in all models and mechanisms for access control.

IV. Conclusion

Access control is a very important area in the security of information systems because it ensures the confidentiality and integrity of data. This is what motivated us to do a study about the most famous access control models.

As we have seen in this paper, we have tried to talk about two types of access control models, the Discretionary and Mandatory models, and then we have detailed the RBAC model that assembles the advantages of the two models that came before it, but it still has a major problem of the contexts definition.

To complete this first study on these three models, we have created a summary table containing an abstract of each one. In order to give solutions to problems met in RBAC model, we have described the OrBAC model, which still to this day, the most innovative and complete model that guarantees the satisfaction of any security policy.

In addition to solving the problems of other models, OrBAC eliminates conflicts between security rules, since it defines a security policy using the permissions, prohibitions, obligations and recommendations.

In general, the OrBAC model is the most reliable model so far. But this does not prevent that it is similar to other models in its vulnerability to covert channels. This vulnerability that we discovered during this study should allow us to detail covert channels and make a deep study trying to solve or find a way to deal with this type of vulnerability in information systems.

References

[1] K. Rihaczek, "The harmonized itsec evaluation criteria," *Comput. Secur.*, vol. 10, no. 2, pp. 101–110, Mar. 1991. [Online]. Available: http://dx.doi.org/10.1016/0167-4048(91)90003-V

[2] B. W. Lampson, "Protection," *SIGOPS Oper. Syst. Rev.*, vol. 8, no. 1, pp. 18–24, Jan. 1974. [Online]. Available: http://doi.acm.org/10.1145/775265.775268

[3] P. Samarati and S. D. C. di Vimercati, "Access control: Policies, models, and mechanisms," in *FOSAD*, 2000, pp. 137–196.

[4] C. E. Landwehr, "Formal models for computer security," *ACM Comput. Surv.*, vol. 13, no. 3, pp. 247–278, Sep. 1981. [Online]. Available: http://doi.acm.org/10.1145/356850.356852

[5] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman, "Protection in operating systems," *Commun. ACM*, vol. 19, no. 8, pp. 461–471, Aug. 1976. [Online]. Available: http://doi.acm.org/10.1145/360303.360333

[6] E. D. Bell and J. L. La Padula, "Secure computer system: Unified exposition and multics interpretation," Bedford, MA, 1976. [Online]. Available: http://csrc.nist.gov/publications/history/bell76.pdf

[7] R. S. Sandhu, "Lattice-based access control models," *Computer*, vol. 26, no. 11, pp. 9–19, Nov. 1993. [Online]. Available: http://dx.doi.org/10.1109/2.241422

[8] D. E. Denning, "A lattice model of secure information flow," *Commun. ACM*, vol. 19, no. 5, pp. 236–243, May 1976. [Online]. Available: http://doi.acm.org/10.1145/360051.360056

[9] D. E. Bell and L. J. LaPadula, "Secure computer systems: Mathematical foundations," pp. 74–244, 1973.

[10] K. Biba, "Integrity considerations for secure computer systems," Mitre corp Rep., Tech. Rep.

[11] D. F. Ferraiolo and D.R.Kuhn, "Role-Based Access Control," in *Proc. of the 15th National Computer Security Conference*, 1992, pp. 554–563.

[12] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, Feb. 1996. [Online]. Available: http://dx.doi.org/10.1109/2.485845

[13] R. T. Simon and M. E. Zurko, "Separation of duty in role-based environments," 1997, pp. 183–194.

[14] G.-J. Ahn and R. S. Sandhu, "Role-based authorization constraints specification," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 207–226, 2000.

[15] A. A. E. KALAM, "Modèles et politiques de sécurite pour les domaines de la santé et des affaires sociales," Ph.D. dissertation, Institut National Polytechnique de Toulouse, France, 2003.