Policy2001: Workshop on Policies for Distributed Systems & Networks, Bristol, UK.
(Extended Abstract)

The Incorporation of Control Principles into Access Control Policies

Andreas Schaad[1] & Jonathan D. Moffett
Department of Computer Science, University of York
Heslington, York YO10 5DD, UK
{andreas, jdm}@cs.york.ac.uk

1 Introduction

Access control policies should be based upon the goals of an organisation, as expressed in its control principles, but the principles are not normally visible in the access control system (ACS). It would be desirable to represent them explicitly in the ACS so that they can be used in access control policies and rules.

In this paper we discuss common control principles and how they could be represented within an ACS. We have started with the control principle of Separation of Duties, and produced a prototype simulation tool which shows the effect of administrators' actions on the separation of duties constraints of a RBAC (Role Based Access Control) system.

2 Organisational Control Principles

2.1 Control Principles

In order to achieve and maintain control, organisations set out control principles which are used to guide its decisions, but they have not become explicitly represented in ACSs. This leads to the problem that proposed actions which would breach the principles are not recognised by the ACS, and may therefore be wrongly permitted.

2.2 Common Control Principles

Each organisation uses a different set of control principles as the individual control requirements are very diverse. Some common control principles are described below.

Separation of Duties: By partitioning critical transactions and assigning sub-tasks to different entities we prevent any one person from performing the whole transaction, thus reducing the risk of any error or fraud.

Delegation: Delegation is an important part of any working organisation, since the main task of management is to get work done through the efforts of other people. Delegation of authority can be seen as a specialisation of tasks and responsibilities, through which a superior delegates or transmits pieces of authority downward in the organisational chain along with the obligation to perform specific duties.

Supervision, Review and Audit: Supervision and review control whether delegated tasks are carried out as required. Supervision is a general activity carried out by a person in a superior position. Reviewing is task-specific and does not necessarily need to be performed by a superior position. Auditing in general serves as an activity of checking that a system performs its required function.

3 Security Policies and Control Principles

As shown in figure 1, the ADF makes its decision based on individual access rules, on information about system users, on the system state (e.g. time) and on fixed security policies. Both fixed security policies and mutable access rules are incorporated into the reference monitor. On the other hand control principles are used by human beings outside the access control system to determine fixed policies and access rules. This makes the enforcement of control principles difficult to achieve reliably, because it is carried out on an *ad hoc* basis by human beings who are liable to error. It would be desirable to incorporate them into the reference monitor, so that is becomes possible to detect, within the system, if they are being violated.

[1] Sponsored by the EPSRC under award no 99311141

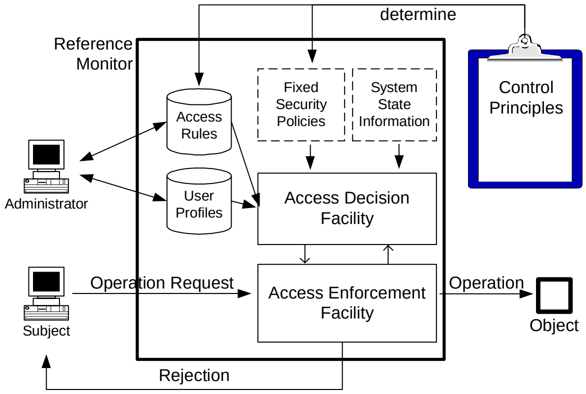


Figure 1: The Reference Monitor in MAC/DAC systems

4 Separation of Duties in Role-Based Environments

Role based access control (RBAC) systems, e.g. Sandhu's RBAC96 model [1], are a development of traditional MAC or DAC based systems, providing a more abstract approach to access control than their predecessors.

RBAC provides the mechanisms that are needed for the integration of Separation of Duties into an access control system, by introducing a set of pairs of mutually exclusive roles (conflict set).

4.1 Separation of Duties - Related Work

The two initial papers on issues of separation of duties are the Clark-Wilson [2] and Nash-Poland [3] papers, emphasising its importance, while not attempting to integrate it into a formal model.

Kuhn addresses the mutual exclusion of roles to implement separation of duty in a role-based access control system [4]. Simon et al. [5], show different variations of the separation of duty in role-based environments. The two categories of separation of duties that they identify are strong (Static) and weak (Dynamic) exclusion. Gligor et al. [6] use the observations made in [5] for a more formal description of separation of duties characteristics.

Nyanchama et al. [7] introduce a taxonomy of types of conflict of interest in their role graph model. It puts emphasis on the different types of conflict of interest in the three planes of users, roles and permissions and the relations between and among them.

4.2 Role Hierarchies and their Impact on Separation of Duties

Role hierarchies are partial orders, and are therefore transitive. Thus, if a user is a member of a pair of roles which is not in the conflict set, there may still be a violation of a separation of duties policy as expressed by the conflict set. The possible consequences of role hierarchies and their interaction with control principles is described in [8].

5 Animating Separation of Duties in a Role-Based Environment.

One of the aims of our research is to prove properties of experimental configurations of access control systems. Ideally, this would be done by formal proof but, unfortunately, currently available proof support is not able to deal with systems which are at all complex. We are therefore using simulation to examine the results of our experiments. Although it is not capable of providing positive proof of correctness, it can show, in many situation, that our design is wrong, or has unintended consequences. Indeed, it has already done so!

We wish to validate the state of an access control system with respect to separation of duties. We use Prolog and Visual Basic as the underlying technologies for simulation. The result is the SoDA (Separation of Duties Animator) tool that can be used to analyse role-based access control models for static separation of duties conflicts.

5.1 Using Prolog for the Simulation of Separation of Duties Properties.

We are using Prolog for modelling Separation of Duties properties because it handles recursive queries naturally.

We have used a Prolog database of facts for our database. Upon these facts we build some rules. The model that we chose was Sandhu's RBAC96 ($RBAC_1$) model as it easy to implement, sufficiently formalised and provides us with the concept of role hierarchies.

Using a Prolog query interface we can ask our system about facts such as existing roles, users or permissions ①, all mutually exclusive roles ②, a certain pair of exclusive roles or all the roles a user is directly assigned to ③. We can then use these basic queries and combine them in rules such as: asking for all roles that a user has also inherited as a result of being assigned to a role; or for a direct violation when a user is assigned to a pair of mutually exclusive roles ⑤. A combination of rules ④ and ⑤ enables us to find violations due to inheritance.

① *role*(R), *user*(U), *permission*(P).
② *exclusive*(Role1,Role2).
③ *ur_assignment*(User,Role).
④ *inherits_from(*Super_Role,Sub_Role*):-*
  *is_a(*Super_Role,Sub_Role*).*
  *inherits_from(*Super_Role,Sub_Role*):-*
  *is_a(*Super_Role,Sub_Sub_Role*),*
  *inherits_from(*Sub_Sub_Role,Sub_Role*).*
⑤ *show_direct_violation(*User,Role1,Role2*) :-*
  *user(*User*), role(*Role1*), role(*Role2*),*
  *ur_assignment(*User,Role1*),*
  *ur_assignment(*User,Role2*),*
  *exclusive(*Role1,Role2*).*

5.2 An Example System

Our example system is that of a software development company. Within that company their exist a variety of roles that company members can take.

Several people will be assigned to the role of a programmer whilst it is imaginable that the same person works as a requirements engineer or on the design of the graphical user interface. Also people work on different projects at the same time.

Certain roles are required to be exclusive, either directly, or by inheritance through the role hierarchy.

The mutually exclusive roles are represented in figure 2. A user must not be assigned to two roles which are directly connected.

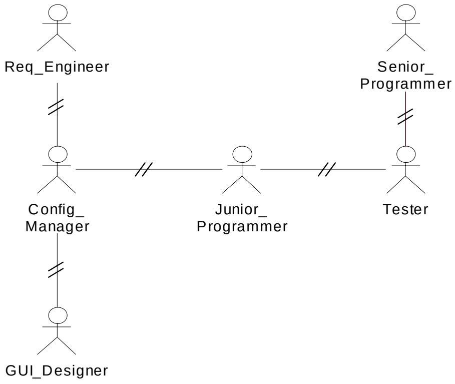


Figure 2: Mutually Exclusive roles

The role hierarchy is graphically represented in figure 3.

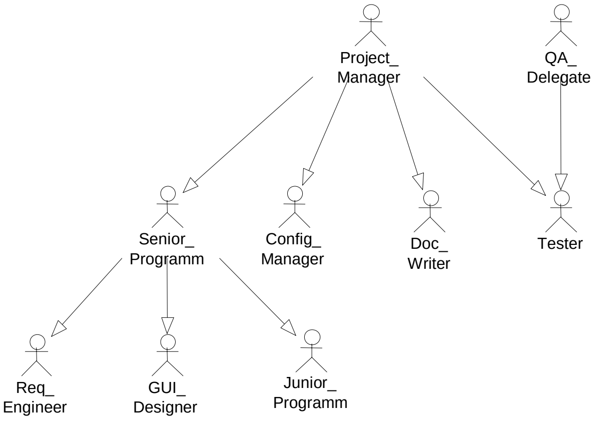
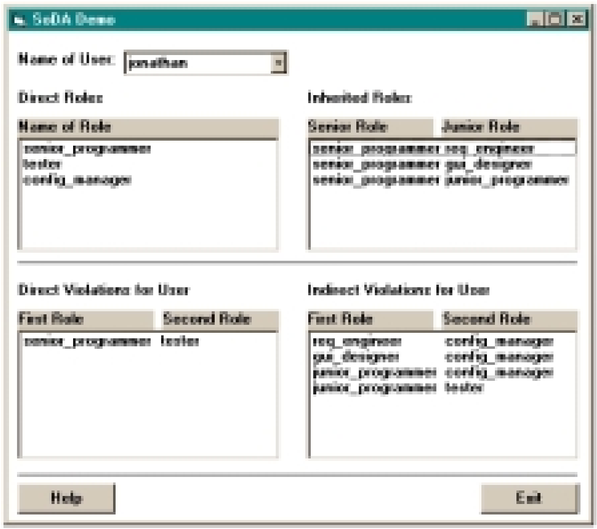


Figure 3: Roles and Role Hierarchy in the Company

5.3 The SoDA (Separation of Duties Animator) tool

The SoDA GUI is an extension to the Prolog query interface.



Name of User: jonathan

Direct Roles

Name of Role
senior_programmer
tester
config_manager

Inherited Roles

Senior Role	Junior Role
senior_programmer	req_engineer
senior_programmer	gui_designer
senior_programmer	junior_programmer

Direct Violations for User

First Role	Second Role
senior_programmer	tester

Indirect Violations for User

First Role	Second Role
req_engineer	config_manager
gui_designer	config_manager
junior_programmer	config_manager
junior_programmer	tester

Figure 2: The SoDA user interface

Looking at figure 5, we can see that the tool has found direct and an indirect (by inheritance) violations of our mutual exclusion constraints for the user jonathan. As we deliberately assigned our user jonathan with the two exclusive roles of senior_programmer and tester the direct violation is easy to explain.

*ur_assignment(*jonathan,senior_programmer*).*
*ur_assignment(*jonathan,tester*).*

Of more interest is the fact that we also have an indirect violation for the user jonathan. He is directly assigned to the roles of the senior_programmer, config_manager and tester. We know which roles the role of the senior_programmer inherits (figure 3) and we can see that all of these are mutually exclusive to the role of the config_manager, and one of them to the tester role as well (Figure 4). This explains the indirect violations as indicated in the lower right box.

6 Conclusion

Technology

We are developing a second prototype with a facility for integrating any ODBC supporting database in order to allow the basic facts to be held in a relational database. This would allow for the direct run-time manipulation of the system and a stronger separation of program logic from the facts.

Separation of Duties

For the future we plan on extending the tool to handle dynamic separation of duty constraints as they provide a more flexible approach than the static separation of duties. Also we are considering studying roles and their activation in different projects using the Chinese Wall approach [9].

Other Control Principles

The techniques that we have used on separation of duties appear to be possible to extend to the control principle of delegation by using delegate roles. It is perhaps more important, from a practical point of view, to provide some means of integrating the requirements of supervision, review and audit into a system. This complex task requires further work.

References

[1] Sandhu R., E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models." *IEEE Computer*, vol. 29, pp. 38-47, 1996.

[2] Clark D. and D. Wilson, "A Comparison of Commercial and Military Security Policies." presented at IEEE Symposium on Security and Privacy, Oakland, California, 1987.

[3] Nash M. and K. Poland, "Some Conundrums Concerning Separation of Duty." presented at IEEE Symposium on Security and Privacy, Oakland, CA, 1990.

[4] Kuhn R., "Mutual exclusion of roles as a means of implementing separation of duty in role-based access control systems." presented at Proceedings of the second ACM workshop on Role-based access control, 1997.

[5] Simon R. and M. Zurko, "Separation of Duty in Role-Based Environments." presented at Computer Security Foundations Workshop X, Rockport, Massachusetts, 1997.

[6] Gligor V., S. Gavrila, and D. Ferraiolo, "On the Formal Definition of Separation-of-Duty Policies and their Composition." presented at IEEE Symposium on Security and Privacy, Oakland, CA, 1998.

[7] Nyanchama M. and S. Osborn, "The role graph model and conflict of interest." *Transactions on Information Systems Security*, vol. 2, pp. Pages 3 - 33, 1999.

[8] Moffett J., "Control Principles and Role Hierarchies." presented at 3rd ACM Workshop on Role Based Access Control (RBAC), George Mason University, Fairfax, VA, 1998.

[9] Brewer D. and M. Nash, "The Chinese Wall Security Policy." presented at IEEE Symposium on Security and Privacy, Oakland, CA, 1989.