Contributions in role based access control for e-Commerce applications

PhD Candidate: Eng. Mihaela Georgetta Ordean

PhD Coordinator: Prof.dr.eng. Dorian Gorgan

- Jully 2008-
1 Introduction

Access control has as objective protection of resources against unauthorized access. An access control system implies a policy which specifies the users who have the rights to access a resource and the manner s/he can do it. The policies are generally expressed by the current state of the system and the deriving states. If the control access system is seen as a transitional system, it consists from the set of states, transition rules and a set of properties or interrogations.

Nowadays, there are an increasing number of Web applications that require authorization decisions. These applications include (but not limited to), e-Commerce applications, management and sharing of distributed resources, execution of downloaded code, etc. Authorizing these kinds of applications is significantly different of that of centralized systems and even of that of relatively small distributed systems.

A system for access control is in the same time an instance of a scheme and specifies the rules for the transitions between states [TL04]. There are several models for access control like MAC (Mandatory Access Control), DAC (Discretionary Access Control) and more recent: RBAC (Role Based Access Control).

2 PhD thesis objectives

PhD thesis refers to a particular domain of authorization and proposes the SCAR-ACE model for role based access control in e-Commerce applications. e-Commerce applications become increasingly more complex, requiring access to heterogeneous resources of users in different roles. Access control in e-Commerce applications is an important subject of nowadays scientific research.

(1) Research purpose

In [B05] are defined the research directions for models, architectures and technologies in access control. The PhD thesis subscribes to these directions namely the realization of a model that supports access control in a large distributed system. SCAR-ACE model extends the approach described in [B05] by using state machines.

Another objective is the flow control modeling in distributed systems.

(2) Problem to be solved

One of the obstacles in accessing the resources over Internet is the inability to manage the authorization in an efficient way without increased costs regarding time and money. In [PSA01] are presented two implementations for access control over Internet. Both solutions are based on cookies, implying a security leak.

(3) Proposed solution

The current work proposes a safe model for role based access control without using cookies. The proposed model allows the access to system resources only for authorized users.

In order to determine the flow and to control the access to the resources in a distributed application, is introduced the notion of role as an intermediary between a user and its permissions. Each role has attached a set of permissions (or privileges) to access the resources and operations. Figure 1 shows the user access to operations and resources by the mean of roles and state machines according to SCAR-ACE model.
3 The components of the SCAR-ACE model

The components of the RBAC model are [FSG01]: RBAC Kernel, Roles Hierarchies, Static Relations, Dynamic Relations and Extensions. SCAR-Ace model is conforming to the RBAC model.

The SCAR-ACE Kernel includes six base elements and relations between these elements: users, roles, sessions, state machine, permissions, operations and objects.

Each role has attached a state machine SCAR-ACE specific. The user access to a set of permissions takes place by the mean of a state machine and this is the major difference in comparison with the existing models.

A role hierarchy is mathematically speaking a partial or total order that defines the relations between roles, where senior roles gain the permissions of junior roles [FSG01]. There are many types of role hierarchies [SFK00]:
- general hierarchies – include multiple inheritance of permissions;
- limited hierarchies – impose restrictions inside the roles hierarchy.

SCAR-ACE hierarchy is a limited hierarchy and it is a tree hierarchy. The roles in the hierarchy are restricted to a single descendant and a single ascendant.

RBAC relations are relations for separation of duties and impose constraints over the model. Static separation of duties relations (SSD) are used to avoid conflicts of interests inside a company (prevent a user to exceed the authorization level corresponding to his/her position).

SCAR-ACE model contains SSD and these are implemented by the use of state machines.

4 Conceptual models

In order to analyze different versions of RBAC, there was defined a family of four conceptual models [SCH96]. RBAC$_0$ is the base model and is the minimum condition of a role based access system. The advanced models RBAC$_1$ and RBAC$_2$ include RBAC$_0$ model. RBAC$_1$ introduces the roles hierarchy while RBAC$_2$ adds constraints. RBAC$_3$ includes RBAC$_1$ and RBAC$_2$ and, by transition, RBAC$_0$.

SCAR-ACE model refers to a RBAC$_3$ and for SSD implementation imposes constraints. By parallelism with RBAC, SCAR-ACE model is composed from SCAR-ACE$_0$, SCAR-ACE$_1$, SCAR-ACE$_2$ and SCAR-ACE$_3$. 
5 Formal definition of SCAR-ACE

In order to formalize SCAR-ACE model I have defined a language to express the authorization constraints [OG’07]. This language specifies a set of constants, variables and predicates.

There are defined the following sets of predicates:
- specification predicates for application definition;
- planning predicates for application constraints;
- execution predicates for application execution.

6 Phases of model specification

In this section are presented the necessary steps for SCAR-ACE creation, each step being realized by a certain component of the authorization system.

First step, named static analysis determine the static part of the model constraints and verifies their consistency. If this phase is successful, then we move to the next step for verification/updating by which are realized the assignments of roles to operations, up to the static analysis results. The planning phase has as input the previously defined constraints and as output the states machine for each role, by associating the states, the transitions between states and the events. If this phase is successful, is associated to the state machine the active part of graphical user interface (are established the menus and the buttons of each page).

The phases are shown in figure 2.
7 Experimental results

There were performed functional and data security tests. The functional tests determine the performance level of the distributed application and the efficiency of the scalability. The tested application implements the SCAR-ACE model and, the using of state machines, might involve execution delays.

The security tests were tests for simulated attack and tests for conformity with the requirements of SCAR-ACE model.
7.1 Functional tests

Performance tests

*The hypothesis to be verified:* state machines introduce delay times at execution.

The system configuration was: server – HP computer, Pentium(R) 4 CPU 2.53GHz, 1GB RAM and two clients with the same characteristics.

The tests aim was to verify the response time of the distributed application which implements the SCAR-ACE model when are running 300 parallel tasks. The tests verify the response times for login and logoff operations. The result is illustrated in figure 3.

![User session initialization](image)

**Figure 3. Response time at authentication**

By analyzing the figure 3 we notice that the average time is about two times less with respect to the login time and about 50% bigger than total execution time. In the same chart, the time for the 20th user is very high. This is because this is the moment when the state machine is loaded into the memory. This action happens once, when the first task needs a state machine, and after this, the state machine persist in memory until is discarded. After the 20th user, there are still time oscillations but not so important. The total time is relatively constant.

The response time for logoff is presented in the thesis and is similar with login time.

*Conclusion:* state machines introduce increased times at load and discard. The total execution time is not strongly influenced by the state machines.

Scalability tests

*The hypothesis to be verified:* the scaled SCAR-ACE application has lower execution times than un-scaled application that implements the order forms model.

The system configuration was: 2 servers and 2 clients with the following characteristics: HP computers, Pentium(R) 4 CPU 2.53GHz, 1GB RAM

There were performed tests for two use cases; one of them is illustrated in figure 4 and the other in the extended thesis.
There were considered 100 execution threads and the response times are illustrated in figure 5. We can notice better times of scaled SCAR-ACE application.

![Figure 4. Use case for scalability test](image)

**Conclusion:** Scaled SCAR-ACE model has lower execution times than un-scaled order form model.

### 7.2 Security tests

**Tests for a simulated attack**

*The hypothesis to be verified:* the credential can’t be obtained from any state of the application.

The attack consists in attempts to penetrate the system by fraud in order to obtain confidential information. There were analyzed two possibilities for obtaining a credential: by direct access or by reconstruction. Both tests proved that the application can’t be penetrated and the credential is safe. On the other side, if an attacker might obtain (by absurd) a credential, this is not longer valid because the life time of any credential is very short.
Conclusion: a credential can’t be obtained from any state of the application.

Tests for conformity with the SCAR-ACE model requirements

There were tested the following requirements: separation of duties and roles inheritance.
There were imagined significant test to prove the separation of duties by attempts to escalate the responsibilities of a visitor. The tests demonstrated a correct separation of duties. For role inheritance verification were created use cases for access from a senior state machine to a junior state machine.

The conclusion is that the distributed application is conform to the SCAR-ACE model requirements.

8 Own contribution

In this PhD thesis is described a model (SCAR-ACE) that supply a policy for role based access control in open and distributed systems, model that is the main contribution. In detail the contribution consists of:
- an analysis of existing RBAC models, of the current state of the art and of research teams;
- an original approach of RBAC components in a manner that provides a solution for role based access control in open and distributed systems, more precisely in e-commerce applications. In this respect, was extended the standard RBAC with state machines by whom the permissions are granted at the operation level. The state machines were included as model components and were expressed the relationships between elements. The SCAR-ACE model includes a tree role hierarchy and allows roles inheritance;
- a model for constraints relationships and for static separation of duties by using a states machine for each role. A user whom was assigned a role has attached one and only one states machine;
- SCAR-Ace modeling based on levels: SCAR-ACE\textsubscript{0}, SCAR-ACE\textsubscript{1}, SCAR-ACE\textsubscript{2} and SCAR-ACE\textsubscript{3};
- the definition of the formal model by analyzing the constraints realization and permissions assignment in SCAR-ACE. There were determined and described the levels for permissions granting by the use of UML;
- the identification of rights tuning in SCAR-ACE by the mean of credentials and the description the credential components;
- the definition of permission assignment policy and its algebra;
- the modeling of the access by using state machines and the control gaining in SCAR-ACE;
- formal definition of the SCAR-ACE model and exemplifying for a specific case;
- identification of constraints phases;
- implementation and validation of SCAR-ACE;
- analyzing of e-Commerce applications, of roles and existing architectures;
- implementation of case studies and analyzing the experimental results.

9 Future work

The PhD thesis might be considered the initial step in the research of access control policies in open and distributed systems. There are many possible future works detailed in next paragraphs.

The approached policy extends the base RBAC in a way to be included components for access control by using on state machines. These components might be examined with the purpose to determine is they could be used in the transition from one RBAC system to another one.
The SCAR-ACE model could become complex enough if we consider different types of hierarchies, the current model being specific for tree-like role hierarchies. A possible extension could consider other types of hierarchies like reverse tree or lattice.

Present approach organizes and structures the access control policies in e-Commerce applications but the model could also be investigated for other types of applications. The state machines which constitute the RBAC extension are, generally speaking, graphs and their implementation requires advance knowledge. A new research direction might implement a graphical user interface for the design of SCAR-ACE based applications, an interface whose inputs are: the states, the events and transitions between states, and as output the state machines.

The e-Commerce application offers the support in the implementation of SCAR-ACE model for role-based access control. This application includes both: business components and administration components. Some components require prerequisite conditions. A research direction consists in the extension of the business part in a manner to allow the management of different verticals and different facilities (e.g. books acquisition with user profile filters included).

On the other side, the component policy might be marked with context elements, allowing in this way, to the application administrator, to group different components based on certain conditions. This would allow views over components policies, possible to be exploited as visualization tools. Up to the viewing angle might be determined, for instance, buyers who satisfy certain criteria either from security point of view or from acquired goods.

Another possible model extension implies roles association for business processes. In this case would be allowed the use of the model for a workflow where the states machine follows the flow of the business part. In the actual work, the model follows the business flow, and the required modifications would be minimal and would consist in adding process labels.

SCAR-ACE considers high level restriction while the primitive elements are roles, permissions and state machines. In order to smooth the differences between different RBAC models would be of interest a research where the primitives are the tri-tuples (object, access, action) and the extension by creating equivalent policies. This research would consist mainly in the determination of constraints.

Another possible extension might be the encapsulation of SCAR-ACE components into an interface between different existing RBAC models. Each model would be a state in the states machine and could realize the cooperation between different RBAC models. In these environments, the interface policies could be used in differences mediation between the different domains of access control. In the case of cooperation, this might depend on administration policies. In certain cases, the transitions between models could be acquired based on trust, without the necessity of authentication. This way appears the possibility of extension in conformity policies for decisions of inter-domain collaboration, decisions based on trust.

The policy for rights management provides a way for access control at a very fine granularity. The rights are granted in SCAR-ACE by the mean of state machines. These rights might be organized into a hierarchy which would allow a more concise policy for their granting. A possible rights extension would be their association with a risk factor. This way, the permitted actions are not equal from the point of view of the damage they might cause to the system in case of a mall usage. By rights association with a risk factor, we might issue some policies to specify supplementary supervising in access granting for the right with high risk factor. This way, a risk would be a value which, up to one ore many thresholds, would or wouldn’t require supplementary supervision.

Between the main motivations of the current research is the developing of a access policy in open and distributed systems, for access control based on user role. An extension would be the use of the model in domains like: health care, telecommunications, human resource management, or any other Web services which require roles and privileges.