Optimized Multi-Domain Secure Interoperation using Soft Constraints

Petros Belsis, Stefanos Gritzalis, Sokratis K. Katsikas
Laboratory of Information and Communication Systems Security
Department of Information and Communication Systems Engineering
University of the Aegean, Karlovasi, Samos, Greece
{pbelsis, sgritz, ska}@aegean.gr

Abstract. Building coalitions between autonomous domains and managing the negotiation process between multiple security policies in a multi-domain environment is a challenging task. The negotiation process requires efficient modeling methods for the determination of secure access states and demands support from automated tools aiming to support administrators and to minimize human intervention; thus making the whole process more efficient and less error-prone. In this paper we define a framework that enables the representation of policy merging between autonomous domains, as a constraint satisfaction problem, while remaining neutral in regard to the policy language. Role and permission hierarchies are modeled using the constraint programming formalism. Policy mappings are utilized in order to enable cross-organizational role assignment. Further optimization on policy mappings is achieved by casting the problem to a partially ordered multi-criteria shortest path problem.

1. Introduction

With the proliferation of Internet based technologies and the advances in networked systems we have witnessed a raising necessity for flexible access control schemes over distributed environments. Many approaches attempt to provide support for authorization decisions within a single domain framework. Powerful languages have also emerged [7][6], able to express different policies; still their applicability has been enforced on a single domain basis. In many collaborating environments coalitions between autonomous domains are formed to enable mutual sharing of resources and applications, in order to achieve a common goal. Security considerations can rise in magnitude in collaborative environments where different information systems form coalitions, sharing resources and applications. The nature of the coalitions can be dynamic, meaning that domains may join or leave at any moment, or that role and permission determination policy updates reflect in

Please use the following format when citing this chapter:
Belsis, Petros, Gritzalis, Stefanos, Katsikas, Sokratis, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 78–85

necessary updates in the global policy. Member domains of the coalition perform common operations over shared resources. Access to shared resources must be consistent with the individual policies of coalition members. Secure interoperation should retain two basic principles [1]:

- Autonomy principle: if access is permitted within an individual system it should also be permitted under secure interoperation.
- Security principle: if access is not permitted within an individual system, it must not be permitted under secure interoperation.

We are investigating the problem of enabling coalition formation between autonomous domains. We propose a flexible way to enable cooperation between separate Role Based Access Control (RBAC) oriented policies and through the policy mappings we enable assignment of roles to users belonging to different domains [2]. In order to facilitate the coalition management and to make it less error-prone human intervention has to be reduced by the use of automated tools [3]. We also utilize a powerful mathematical framework based on constraint satisfaction, to which the formalization of the problem can be cast. Under this framework and through the concept of policy mappings, we transform the aforementioned problem to a partially ordered multi-criteria shortest path problem, which can be guided using soft constraints. Among the contributions of this paper are the following:

- Secure interoperation is enabled through the concept of policy mappings, while by modeling the different policies using soft constraints we allow for the determination of additional role mappings, leading thus to the creation of optimal solutions.
- We allow for the execution of actions over the shared resources for roles that have not been explicitly mapped to other roles by the administrator; therefore, we introduce a way to determine automated mappings, avoiding at the same time violations of role hierarchy constraints.

The rest of the paper is organized as follows: after a brief introduction in section 1, section 2 presents the formalism principles and their applicability to security models, section 3 presents related work and a brief comparison with our approach, while section 4 concludes the paper and provides the directions of our future work.

2. Problem Formulation

2.1 The RBAC model

The basic notions behind the RBAC [4] models are users, roles, and permissions. A user represents a human entity or an autonomous agent. A role is associated with a post in an organization assigned to the execution of a specific task, while a collection of permissions are assigned to each role, enabling the fulfillment of the obligations associated with such a task. To extend the support for the least privilege principle (that allows to a user the minimum privileges necessary to fulfill a task), sessions are introduced.

A complete RBAC model includes the following variables and functions:

- The sets U (users), R (roles), P (permissions) and S (sessions)
- User to role assignment $UA \subseteq U \times R : U \rightarrow 2^R$

- Permission to role assignment $PA \subseteq P \times R$: $R \rightarrow 2^P$
- A mapping of sessions to a single user assignment US: $S \rightarrow U$
- A mapping from sessions to the set of roles associated with each session $S \rightarrow 2^R$
- A partial ordering $RH \subseteq R \times R$, represented by the symbol: $\geq$, which defines role hierarchy. $R_1 \geq R_2$ implies that R_1 inherits permissions from R_2.

RBAC is a dominant security model due to its flexibility and due to the fact that it reflects organizational hierarchy; moreover, its parameters can be easily codified. For this purpose, several RBAC security policy representation languages have emerged, ranging from formal, graphically annotated to expressive full-scale policy management systems with software tools support.

We do not intend to create a new policy representation language. Our work focuses on enabling the coalition of autonomous systems, where each one retains its own security policy. In fact, there is no restriction that all the domains should follow the same policy language; the only requirement being adherence to the RBAC principles.

Given the fact that permissions are a set of Boolean constraints associated with a given role, we can consider policy representation as a set of Boolean constraints. Multi-domain policy merging can then be cast to a condition of joint satisfaction of a constraint-programming problem. In our approach, the administrators of each domain codify the policies. We do not also consider the case where domains for any reason would attempt to conceal policy related information, as in the case where policies contain sensitive information. For example in the case where ministries cooperate there is no danger that policy disclosure would result in potential danger, since all the parties are cooperating on the basis of a common target. Our approach intends to reduce the administrator's involvement overhead by proposing access states that satisfy the pre-specified preferences of each domain.

2.2 Soft constraint satisfaction

Constraint programming is an emerging technology in the area of artificial intelligence [10]. A constraint satisfaction problem (CSP) includes a set of problem variables, a domain of possible values and a set of constraints defined over these variables. Semiring based CSPs or SCSPs [10] are an extension of CSPs where the constraints are defined over an appropriate semiring. We will mainly adopt the notation introduced in [10], [11]. A semiring is a tuple $<A,+,*,\mathbf{0},\mathbf{1}>$ where

- A is a set with $\mathbf{0},\mathbf{1} \in A$
- + the additive operation is closed, commutative and associative over A with **0** as the absorbing element
- *, the multiplicative operation is closed and associative over A with **1** as its identity element and **0** as its absorbing element
- * distributes over +

A constraint semiring (c-semiring) is a tuple $<A,+,*,\mathbf{0},\mathbf{1}>$ where the idempotency of the additive operation defines a partial ordering such as $a \leq_s b$ iff $a+b=b$. Additionally * is intensive, that is, $\forall a,b \in A \Rightarrow a * b \leq_s a$.

A semiring-based constraint system is a tuple <S,D,V> where S is a semiring, D is a finite set and V is an ordered set of variables. A constraint over such a system is a tuple <def,con> where con⊆V is the type of constraint and def contains the value of the constraint. Thus def assigns a value from the semiring to each combination of values of the variables in con. This value can be a probability, a cot, a preference etc. A SCSP then is a tuple <C,v> where v⊆V and C is a set of constraints.

Given two constraints $<def_1,con_1>$ and $<def_2,con_2>$ over the above constraint system, their combination is defined as $<def,con>=<def_1,con_1> \otimes <def_2,con_2>$ where $Con=con_1 \cup con_2$, where $\cup$ is the union operation over sets, $def= def_1(t \downarrow_{con_1}^{con_2}) * def_2(t \downarrow_{con_1}^{con_2})$ where $t \downarrow_{con_1}^{con_2}$ denotes the part of the tuple t corresponding to variables in con_1. The $\otimes$ operation is commutative and associative, since the * operation is. Moreover, since * is monotone over ≤s, adding constraints will not increase the value associated with any tuple t.

For a given constraint system CS=<S,D,V> where c= <def,con> a constraint over CS, and a set I of variables with I ⊆V, the projection of c over I, $c\Downarrow_I$ is the constraint <def',con'> over CS with con'=I∩con, where ∩ is the intersection operation over sets and $def'(t')= \sum_{\{t | t \downarrow_{I \cap con}^{con} = t'\}} def(t)$. The solution sol(P) of a constraint problem P=<C,con> over a constraint system CS is defined as sol(P)= $(\otimes C)\Downarrow_{con}$. The optimum level of consistency oLevel(P) is obtained if we first obtain the solution and then projects it over the empty set of variables. Typically the oLevel(P) yields an estimation of how much the solution satisfies the constraints of the problem.

2.3 Modeling RBAC policies using soft-constraints

2.3.1 RBAC hierarchies' representation using soft constraints

We can consider two partial orders in an RBAC system [5]: the hierarchy of roles and the hierarchy of permissions. An example of a role hierarchy in a medical domain is given in Figure 1a. Figure 1b shows an example of permissions hierarchy, adjusted to the UNIX permissions representation. Privileges are hierarchically assigned, so that ancestor roles are assigned additional privileges than their descendant roles. A suitable choice of semirings for a multi-domain policy representation can be as follows:

The role hierarchy can be represented by the role semiring: $< R,+_R, *_R, R_0, R_\infty>$, where

- R is the set of roles in the system
- The $+_R$ operation is defined as: $(R_1 +_R R_2)$ is the highest common descendant of roles R_1 and R_2 in role hierarchy
- The $*_R$ operation is defined as the common ancestor of roles R_1 and R_2 in role hierarchy
- R_∞, R_0 are the roles with maximum and minimum privileges. For example in the hierarchy of Fig 1a for the roles in a hospital the Ward Managers have fewer privileges than Hospital Manager, while the least privileges are assigned to nurses.

Next we consider the permission hierarchy and we define the appropriate semiring $<P,+_P,*_P,P_\infty,P_0>$, where

- P is the set of permissions in the system
- The +P operation is defined as: (P1+P P2) is the highest permission between P1 and P2
- The *P is defined as the lowest permission
- P∞, P0 are highest and lowest permission in the hierarchy respectively.

One solution to the problem, adopted in [5] could be to use the SCSP induced by the domain's assignments of permissions to local roles to find the permissions associated with this particular assignment P_1 of local roles to global roles. Then the SCSP P_2 describing the access rights over the shared workspace is being built. If P_1 dominates P_2 then solution is achieved. Among the limitations of this method we can recognize the fact that no roles are considered as critical and that a possible assignment of permissions to local roles could violate several restrictions defined by the local policies, resulting in a security violation as described in Section 1 for the resulting global policy.

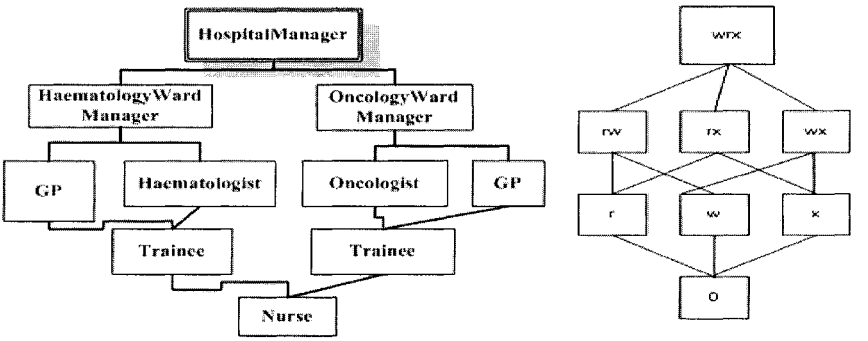

Fig. 1.a (left): An example of a role hierarchy for a medical domain. 1b (right): Example of a permissions hierarchy (adopted from [5])

The above method can be utilized as a recommendation in order to facilitate the administrator's overhead when attempting to merge the local policies. We will expand the applicability of this framework to support the correspondence of roles from one domain to the other, when these roles are not explicitly mapped. At the same time we avoid hierarchy violations during this policy merging process.

2.3.2 Formulating role mappings as a soft constraints multi-criteria shortest path problem

Consider the case where we have two different role hierarchies (Fig 1a). We can represent the roles in this hierarchy by considering a graph G=(N,E) where the roles are represented as nodes in the graph and we assign a weight to each arc e ∈E from node p to node q (p,q ∈N). This weight can be a pair of values, associated with the level of each role in the hierarchy (a parameter that defines how important is a role in the organizational hierarchy) and the criticality associated with each role. Now this

example may be modeled by two semirings. For the first parameter, we can define a semiring <N, +, min*, 0, +∞> where min* defines the minimum difference considering the result is positive, and + with the classical meaning. For the first parameter of the label, related with the criticality, we define a semiring <N, +, min, 0, +∞>, where min and + are defined with the classical meaning. Consider now the following scenario: According to the technique mentioned in the previous paragraph some of the roles are merged and a number of mappings are established. Now in the case where a role from one domain needs to be assigned the permissions for a role in another domain, we can formalize the problem so as when there is not a direct established mapping from one role in the domain to the other, the system will find (if there exists) the optimal path without additional action to be taken by the domain's administrators. The system simply queries for the target role's permissions. Then we just have to find the shortest path from role p to role v. The cost is measured always counting the parameters assigned to each role. The sole check that needs to be performed, is that there is no hierarchy violation, since the difference from the source to the intermediate roles is not negative at any stage of the path resolution procedure. Of importance is also the discovery of paths where the right-hand (second) terms have minimal differences, implying a similarity in their criticality.

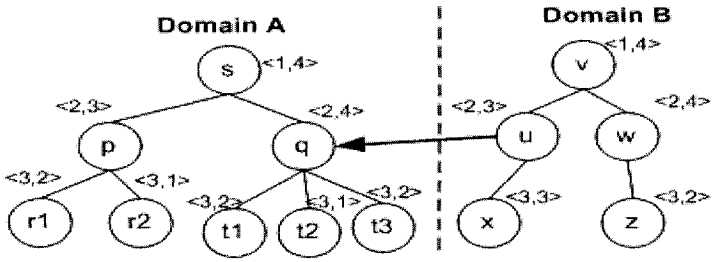

Fig. 2. Example of a role mapping and role hierarchy representation with costs

The problem can be formulated into a Soft Constraint Logic Programming (SCLP) [12] [10] program, which works over an appropriate semiring. In order to find a path that does not violate hierarchy constraints, we calculate the differences between the first values in the pair assigned to each node. We only allow positive differences, meaning that the target role has to be lower in the hierarchy (we consider the different hierarchies and the positions at the same depth as equivalent, independent of the domain to which they belong). Additionally we want to calculate minimal differences based on the second value, so that the criticality of the assigned path is minimal. In the example of Fig. 2, let us consider that a user assigned to role v wants to access some shared resources, which originally demand access rights of role q on the Domain A. There is a direct mapping from role u to role q. The SCLP program will work as follows: v:- c_{vu}, u c_{vu}:-<1,7>, The first term of c_{vu} is calculated by subtracting the hierarchy differences (considering they are positive) $\sum_{i}^{i,j:neighbours} \sum_{j,i \le j} (x_i - x_j)$, while the second term $\{\min[\sum_{i}^{i,j:neighbours} \sum_{j} (y_i + y_j)]\}$ is based on the sum of the criticalities which can be set arbitrarily, to hinder administrators

from activating these intermediate roles unnecessarily. Accordingly, for transition from u to q we have, u:- c_{uq} ,q c_{uq} :-<0,7>. Additionally, we can pose different restrictions, or there is the case that the role assignment is not allowed since there will be some violation of the hierarchy, or that there does not exist mapping. In this case there are two options: either the request is denied either the demand is resolved based on the administrator's intervention, who should create a new appropriate mapping. By modeling the network as described, we enable policy merging to a high extent, retaining hierarchy related restrictions and thus enabling a secure and scalable solution for the problem of secure interoperation.

3. Related work and Discussion

The problem of enabling the establishment of a multi-domain coalition is a challenging one and attracts lately considerable research focus due to the impact and benefits related with its realization.

In [3] a negotiation language is introduced, based on the RCL2000 [6] RBAC policy language. All the language statements have an equivalent in Restricted First Order Predicate Logic (RFOPL) statements. This framework is flexible, though the number of coalition parameters as well as the presence of the coalition access matrix makes it hard to scale for large number of domains and large number of resources.

In [8] interface policies are introduced. Interface policies enable the determination of role mappings; still the proposed framework does not allow optimization and poses the burden of coalition establishment on the administrator, making it less flexible. Additionally there is no specific formalism and support from tools to facilitate the formation of the coalition from the beginning.

Joshi et al.[9], define a multi-domain policy language based on their X-RBAC model. Under this framework, role codification parameters are stored in XML (eXtensible Markup Language) files, for interoperability reasons. Role mappings are manually specified in separate files, demanding a lot of human effort in order to set up the coalition. There is no support yet from automated tools while updates to local policies are difficult to reflect in the global policy.

In [5] a negotiation scheme, which utilizes soft constraints, is being introduced. We extend this model by incorporating the notion of policy mappings that enable cross-organizational role assignment and by retaining at the same time the basic principles of security and autonomy under secure interoperation. Additionally, in our approach, by modeling role hierarchies using the graph approach and assigning weights to roles we enable the determination of optimal paths and additional policy mappings not explicitly stated by domain's administrators, without violating role hierarchy restrictions and by activating the minimum number of critical roles. Under this prism, our framework proves to be more flexible by incorporating more parameters in the role determination process and allowing optimization by codifying the domain's preferences as soft constraints.

4. Conclusions

We have expressed the negotiation problem between autonomous domains as a constraint satisfaction problem. Interoperation is achieved through role mappings, which can be established as a solution to the constraint satisfaction problem. In order to enable role additional assignments not explicitly stated by the administrators, without violation of security constraints, we cast the problem of role assignment to a multi-criteria shortest path problem. Our solution is scalable and can be used as a support tool for the coalition responsible administrators.
Future work can address issues like negotiating policies when there is no established mutual trust between the domains and thus policy exposure would result to some domains attempting to gain advantage over others.

References

1. Gong L. and Qian X. "The complexity and composability of secure interoperation". In *Proceedings of the Symposium on Security and Privacy*, pages 190–200, Oakland, CA. IEEE Press, 1994.
2. Belsis P., Gritzalis S., Katsikas S., "A scalable Security Architecture enabling coalition formation between autonomous domains". To appear In "*Proceedings of the IEEE ISSPIT International Conference on Signal Processing and Information Technology*", December 2005 Athens, Greece.
3. Khurana H., Gligor V. D. and Linn J., "Reasoning about Joint Administration of Coalition Resources", In *Proc. of IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp.429-439, Vienna, Austria, July 2002, IEEE press..
4. Sandhu R., Ferraiolo D., and Kuhn R. "The NIST model for role-based access control: towards a unified standard". In *Proceedings of the Fifth ACM Workshop on Role-Based Access Control (RBAC'00)*, pages 47–63, 2000, ACM press.
5. Bharadwaj V. and Baras J. "Towards automated negotiation of access control policies", In *Proceedings of the 4th IEEE International workshop on Policies for distributed Systems and Networks (POLICY 03), pp. 77-86, IEEE press*
6. Ahn G-J. and Sandhu R., "Role-based Authorization Constraints Specification", *ACM Trans. on Inf. System Security*, pages 207-226, Vol. 3, No. 4, Nov. 2000.
7. Organization for the Advancement of Structured Information Standards (OASIS), XACML Extensible access control markup language specification 2.0, OASIS Standard, (available at http://www.oasis-open.org) (Accessed May 2005).
8. Belokolsztolszki A., Eyers D., Moody K., "Policy Contexts: Controlling Information Flow in Parameterised RBAC", In *Proc. of the 4th International Workshop on Policies for Distributed Systems and Networks (POLICY'03)*, IEEE Press, pp. 99-110.
9. Joshi J.B.D., Bhatti R., Bertino E., Ghafoor A., "Access Control Language for Multi-Domain Environments", *IEEE Internet Computing*, Nov. 2004, pp. 40-50, IEEE press.
10. Bistarelli S., "Semirings for Soft Constraint Solving and Programming", Springer Lecture Notes in Computer Science, Vol. 2962, 2004.
11. Bistarelli S., Montanari U., Rossi F. "Semiring-Based Constraint Logic Programming: Syntax and Semantics, , in *ACM Transactions of Programming. Languages and Systems (TOPLAS)*, ACM Press, Pages: 1 - 29 Vol. 23, issue 1, 2001
12. Bistarelli S., Montanari U. and Rossi F. "Semiring-based Constraint Solving and Optimization", in *Journal of the ACM*, vol.44, n.2, pp. 201-236, March 1997.