Security Assurance for Dynamic Role Mapping in a Multi-Domain Environment

Cuihua Zuo
College of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan 430074, P.R.China
zuocuihua@163.com

Ruixuan Li
College of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan 430074, P.R.China
rxli@hust.edu.cn

Hongmu Han
Computer Department, Wuhan Institute of Technology
Wuhan 430070, P.R.China

Zhengding Lu
Huazhong University of Science and Technology
Wuhan 430074, P.R.China

Abstract

Multi-domain application environments where distributed domains interoperate with each other are becoming a reality in Internet-based enterprise applications. The secure interoperation in a multi-domain environment is a challenging problem. Role-based access control (RBAC) is used for specifying the security requirements of multi-domain applications in this paper. Then, role mapping relationship between domains is described by XML documents. Furthermore, the situations where dynamic role mapping violates separation of duties (SoD) which is one of the three basic security principles for the RBAC model are analyzed in detail, and relevant algorithms to detect the above security problem are designed in this paper.

1. Introduction

The rapid proliferation of Internet and related technologies has created tremendous possibilities for the interoperability between domains in distributed environments. Interoperability provides a means for domains to share resources and services, which enhances performance and resource utilization. However, the interoperability does not come easy as it opens the way for several security and privacy breaches. The security problem can get magnified in collaborative environments where distributed, heterogeneous, and autonomous organizations interoperate with each other [1, 2], hence, security is hard to achieve in a multi-domain environment. Collaboration in a multi-domain environment requires integration of all local policies to compose a global access control policy for controlling information and resource sharing across multiple domains. In this paper, role-based access control will be used as a global access control policy.

The ultimate goal of our research is to solve the security problem of violating statically mutually exclusive role constraint due to dynamic role mapping between domains. Toward the end, XML documents will be introduced to store role mapping relationships which is benefit to saving space and modifying expediently. Then, analysis and classification will be given to the situations of violation of SoD, which is a dynamic constraint and is required in most commercial applications, including digital government, e-commerce and so on. Finally, algorithms are designed to detect the security problem of dynamic role mapping.

The rest of the paper is organized as follows. Section 2 describes the related work about secure interoperation between domains. Section 3 presents storage policy for role mapping relationships between domains and analyzes and classifies the situations of violation of SoD. Section 4 proposes the corresponding protection mechanism which solves the security

This work is supported by National Natural Science Foundation of China under Grant 60403027, Natural Science Foundation of Hubei Province under Grant 2005ABA258, Open Foundation of State Key Laboratory of Software Engineering under Grant SKLSE05-07.

0-7695-3072-9/07 $25.00 © 2007 IEEE
DOI 10.1109/CIS.2007.134

problem described in section 3. Section 5 concludes the research.

2. Related Work

Secure interoperation between domains is a crucial technique of resource sharing and security in distributed environment. The first and foremost challenge in establishing secure interoperation is the composition of a consistent and conflict-free interoperation policy that governs the information and resource exchange of all the domains. Several research efforts have been devoted to the topic of policy composition in the multi-domain environment [3]. IRBAC 2000 [4] presented by Kapadia et al is a model of secure interoperability using dynamic role translation base on RBAC [5]. However, this model did not consider the problem of the violation of SoD which is induced by role mapping between domains.

SoD prevents two or more subjects from accessing an object that lies within their conflict of interests or disallows a subject from accessing conflicting objects or permissions, for example, the same managers cannot authorize payments or sign the payment checks simultaneously [6]. Violations of SoD constraint may occur in an interoperation policy because of the interplay of various policy constraint across domains. The resolution of interoperation inconsistencies related to SoD constraint has not been adequately investigated and the existing approaches rely on manual intervention of policy administrators to resolve SoD conflicts [7].

SoD is the most important basic principle of access control, so distributed access control in multi-Domain should be able to support this constraint perfectly. [8, 9] explain the relationship between SoD and statically mutually exclusive role constraint. Recently, the research on SoD mainly concentrates on the description of constraint [10], the analysis of supporting SoD in various access control models [11, 12], and the expansion of SoD constraint [13, 14]. Therefore, in multi-domain environment, research on the satisfaction of constraints, namely security, is the basis of access control.

3. Role mapping between domains

3.1. Storage policy

To model the interoperability between different domains, this paper assumes that all the domains adopt a role-based access control (RBAC) model. However, if a domain that does not use RBAC as its access control model wishes to join the interoperability session then it can easily provide an export RBAC policy. In RBAC, permissions are associated with roles, and users are granted membership in appropriate roles, thereby acquiring the permissions. The access control policy for domain i is modeled as a directed graph Gi = (Ri, Hi) where the set Ri represents roles and the set Hi represents the hierarchy relationship between roles. For example, r1 ∈ Ri, r2 ∈ Ri, if (r1, r2) ∈ Hi, thus a user acquiring role r1 can acquire permissions assigned to role r2 by using the RBAC permission inheritance properties. Given n domains, the interoperation between these domains is achieved by introducing role mapping between n domains. Meanwhile, such mapping relates roles in different domains. The cross domain mapping is selected by the administrators of the domains according to the interoperability requirements of each system, and is described using a XML document which is stored in security agents of each domain. Each security agent could be not only a server node but also an ordinary node in the network.

Supporting security communication in multi-domain environment is the basis of large-scale distributed applications, and role mapping in multi-domain is a kernel problem. Figure 1 describes role mapping relationships of three domains, and the corresponding XML document is shown in Figure 2.

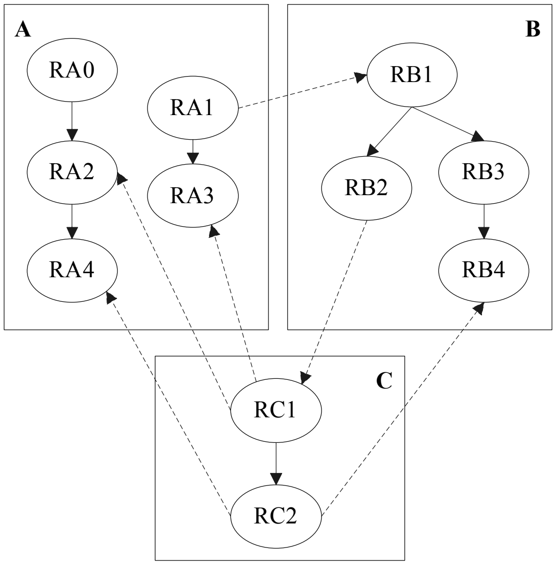

Figure 1. Role mapping relationships between domains

```
<MultiDomainMapping>
<Mapping DomainName=”A” DomainIndex=”1”>
<Role name=”RA1”>
<Domain DomainName=”B” DomainIndex=”2”>
<EntryRole>RB1</EntryRole>
</Domain>
</Role>
</Mapping>
<Mapping DomainName=”B” DomainIndex=”2”>
<Role name=”RB2”>
<Domain DomainName=”C” DomainIndex=”3”>
<EntryRole>RC1</EntryRole>
</Domain>
</Role>
</Mapping>
<Mapping DomainName=”C” DomainIndex=”3”>
<Role name=”RC1”>
<Domain DomainName=”A” DomainIndex=”1”>
<EntryRole>RA2</EntryRole>
<EntryRole>RA3</EntryRole>
</Domain>
</Role>
<Role name=”RC2”>
<Domain DomainName=”A” DomainIndex=”1”>
<EntryRole>RA4</EntryRole>
</Domain>
<Domain DomainName=”B” DomainIndex=”2”>
<EntryRole>RB4</EntryRole>
</Domain>
</Role>
</Mapping>
</MultiDomainMapping>
```

Figure 2. RoleMapping.xml

3.2. Separation of duties

The access between domains becomes easier by the analysis of the multi-domain role mapping relationships in the above section. But new problems occur as well, for instance, in Figure 1, if role RA2 represents “cashier” and role RA3 means “accountant”, it’s quite obvious to figure out that Role RA2 and RA3 belong to the statically mutually exclusive roles, and it’s easy to avoid one user to own the two roles at the same time by defining mutually exclusive roles set in one domain A. However, the user who possesses role RC1 in domain C can own the two mutually exclusive roles RA2 and RA3 during the process of role mapping between domains, therefore, it seriously violates SoD. Furthermore, after role mapping between domains, violation of SoD occurs not only in the above situation where one role of a foreign domain transfers directly to the two mutually exclusive roles. In this section, great details will be given to the situations of violation of SoD due to role mapping in multiple domains.

Assume that role RA4 and RA5 in domain A belong to the statically mutually exclusive ones. In Figure 3, role RB2 in Domain B can transfer to role RA2 and role RA3 through role mapping between domains. Meanwhile, role RA2 can inherit the permissions of role RA4 and role RA3 that of role RA5, therefore, role RB2 in domain B possesses the permissions of the two mutually exclusive roles RA4 and RA5 in domain A indirectly and the statically mutually exclusive role constraint is thus violated.

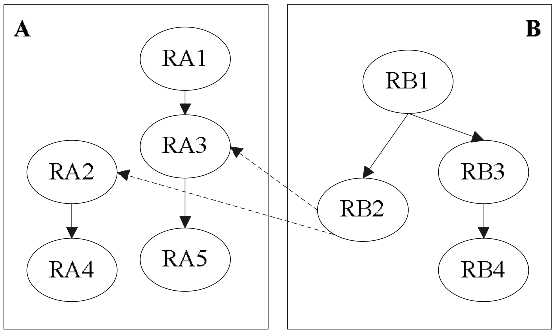

Figure 3. Role mapping between domains (1)

In Figure 4, since the ancestor roles own the permissions of the descendant ones, role RB1 can possess the permissions of role RB2 and RB3. Besides, role RB2 in Domain B can transfer to role RA4 and RB3 that of RA5. Consequently, role RB1 in Domain B acquires the permissions of the two mutually exclusive roles RA4 and RA5 in Domain A indirectly and violates the statically mutually exclusive role constraint.

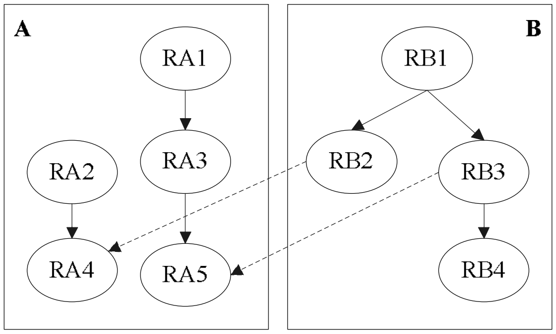

Figure 4. Role mapping between domains (2)

In Figure 5, role RB3 in domain B transfers directly to RA4 in domain A and role RB4 directly to RA5. Moreover, role RB3 is the ancestor of role RB4, role RB3 can therefore own the permissions of role RB4. Hence, the permissions of the two mutually exclusive roles RA4 and RA5 are acquired indirectly by role RB3 in domain B and the statically mutually exclusive role constraint is accordingly violated.

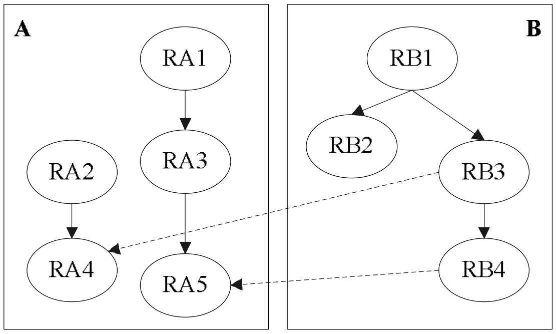

Figure 5. Role mapping between domains (3)

In Figure 6, role RB3 in domain B transfers directly to RA3 in domain A and role RB4 directly to RA2. Since role RB3 is the ancestor of role RB4, it can possess the permissions of role RB4. Furthermore, role RB3 owns the permissions of the two mutually exclusive roles RA4 and RA5, for role RA2 is the ancestor of RA4 and role RA3 is the ancestor of RA5. Hence, the violation of the statically mutually exclusive roles occurs.

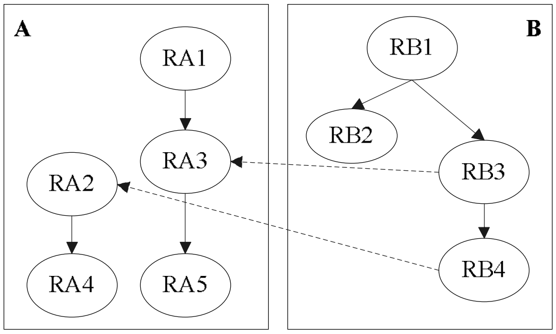

Figure 6. Role mapping between domains (4)

4. Security detection mechanism

According to the analysis in the above section, it is known that if the transference of the role mapping between domains violates the statically mutually exclusive role constraint, there must exist one of the following two possibilities: first, one of the roles in the foreign domain transfers to some roles in the native domain simultaneously, whereas the mutually exclusive roles or their ancestors exist in such role set; secondly, in some domain, two or more roles with the same ancestor or having their own hierarchy relationship transfer to the corresponding roles in the native domain, which form a role set where the mutually exclusive roles or their ancestors exist.

For the sake of description, set(RB1) represents the role set of domain A obtained by the dynamic role transference of role RB1 in domain B. RAi $\geq$ RAj means hierarchy relationship, meanwhile, role RAi and RAj belong to domain A. The statically mutually exclusive role constraint set is represented by SmrA = {(RAm, RAn), …}, in which (RAm, RAn) is a couple of the mutually exclusive roles.

Theorem 1: RA and RB are the role set of domain A and domain B respectively, (RAi, RAj) $\in$ SmrA. If $\exists$RBk ($\exists$RAp $\exists$RAq ((RAp $\in$ Set(RBk)) $\cap$ (RAq $\in$ Set(RBk)) $\cap$ (RAp $\geq$ RAi) $\cap$ (RAq $\geq$ RAj))) is true, the dynamic role mapping violates the statically mutually exclusive role constraint SmrA.

Proof: According to RAp $\in$ Set(RBk) and RAq $\in$ Set(RBk), both RAp and RAq are the mapping roles of RBk in domain A, in the other words, RBk possesses the permissions of RAp and RAq according to the transference relationships between roles of different domains. Besides, due to RAp $\geq$ RAi and RAq $\geq$ RAj, RBk owns both the permissions of RAi and those of RAj and violates the statically mutually exclusive role constraint SmrA.

Theorem 2: RA and RB are the role set of domain A and domain B respectively, (RAi, RAj) $\in$ SmrA. If $\exists$ RBt ($\exists$ RAp $\exists$ RAq $\exists$ RBk $\exists$ RBl ((RAp $\in$ Set(RBk)) $\cap$ (RAq $\in$ Set(RBl)) $\cap$ (RAp $\geq$ RAi) $\cap$ (RAq $\geq$ RAj) $\cap$ (RBt $\geq$ RBk) $\cap$ (RBt $\geq$ RBl))) is true, the dynamic role mapping violates the statically mutually exclusive role constraint SmrA.

Proof: According to RAp $\in$ Set(RBk), RAp is the mapping role of RBk in domain A, RBk thus acquires the permissions of RAp. Meanwhile, because of RAq $\in$ Set(RBl), RAq is the mapping role of RBl in domain A, thus RBl owns the permissions of RAq consequently. Moreover, due to RBt $\geq$ RBk and RBt $\geq$ RBl, RBt possesses the permissions of RAp and RAq. Because of RAp $\geq$ RAi and RAq $\geq$ RAj, RBt owns the permissions of RAi and RAj simultaneously and violates the statically mutually exclusive role constraint SmrA.

Theorem 1 detects that one of the roles in a foreign domain transfers directly to the two or more statically mutually exclusive roles in the native domain (as in Fig. 1), and one of the roles in a foreign domain transfers indirectly to the two or more statically mutually exclusive roles in the native domain (as in Fig. 3). In Theorem 2, detection is given to the violation of the statically mutually exclusive role constraint caused by the phenomenon that two or more roles in a foreign domain (more roles can be classified to pairs) transfer to the statically mutually exclusive roles in the native domain directly or indirectly (as in Fig. 4, 5, 6).

According to the above two theorems, the detecting algorithms concerning whether the role mapping between domains violates the statically mutually exclusive role constraint can be gained and will be demonstrated as follows:

```
Check1 (RBk, SmrA)
{
Flag = False;
// "False" represents satisfying constraint
If (Set(RBk) = $\Phi$ or Num(Set(RBk)) = 1)
Then return Flag;
For each (RAi, RAj) $\in$ SmrA {
For each {RAp, RAq} $\subseteq$ Set(RBk)
If (RAp $\geq$ RAi and RAq $\geq$ RAj)
Then { Flag = True; Return Flag; }
Return Flag;
}
}
```

```
Check2 (RBk, RBl, SmrA)
{
Flag = Flase;
If (Set(RBk) = $\Phi$ or Set(RBl) = $\Phi$)
Then return Flag;
For each (RAi, RAj) $\in$ SmrA {
For each RAp $\in$ Set(RBk)
For each RAq $\in$ Set(RBl)
If (RAp $\geq$ RAi and RAq $\geq$ RAj and
            RBt $\geq$ RBk and RBt $\geq$ RBl)
Then { Flag = True; Return Flag;}
Return Flag;
}
```

```
Check-Constraint ()
{
For each RBk $\in$ RB
Check1 (RBk, SmrA);
For each {RBk, RBl} $\subseteq$ RB
Check2 (RBk,RBl, SmrA);
}
```

In the above algorithms, function Check1 is designed according to Theorem 1, and function Check2 that of Theorem 2. The execution of the function Check-Constraint can restrain the situations of violation of SoD effectively.

5. Conclusion

In this paper, we analyze the importance of security of access control in multi-domain. We apply XML documents to store the role mapping relationships between domains, and propose the security problem introduced by role mapping. Furthermore, detecting algorithms are designed through analyzing and classifying the situations of violation of SoD.

References

[1] L. Gong and X. Qian, "Computational Issues in Secure Interoperation", IEEE Transaction on Software and Engineering, Vol. 22, No. 1, January 1996.

[2] J.B.D. Joshi, A. Ghafoor, W. Aref, and E.H. Spafford, "Digital Government Security Infrastructure Design Challenges", Computer, Vol. 34, No. 2, Feb. 2001, pp. 66-72.

[3] S. Dawson, S. Qian, and P. Samarati, "Providing Security and Interoperation of Heterogeneous Systems", Distributed and Parallel Databases, Vol. 8, Aug. 2000, pp. 119-145.

[4] Apu Kapadia, Jalal Al-Muhtadi, R. Campbell, et al., "IRBAC2000 : Secure interoperability using dynamic role translation", University of Illinois ,Urbana, IL, U.S.A. , 2000.

[5] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein , et al., "Role-based access control models", IEEE Computer, Vol. 29, No. 2, 1996, pp. 38-47.

[6] J.B.D. Joshi, E. Bertino, and A. Ghafoor, "Temporal Hierarchies and Inheritance Semantics for GTRBAC", Proceedings of the Seventh ACM Symp. Access Control Models and Technologies, June 2002, pp. 74-83.

[7] E. Lupu and M. Sloman, "Conflicts in Policy-Based Distributed Systems Management," IEEE Trans. Software Eng., Vol. 25, No. 6, Nov. 1999, pp. 852-869.

[8] Li, N., Z. Bizri, and M.V. Tripunitara, "On Mutually Exclusive Roles and Separation of Duty", Proceedings of the 11th ACM conference on Computer and communications security, 2004.

[9] Chen, H. and N. Li, "Constraint Generation for Separation of Duty", Proceedings of the 11th ACM symposium on Access control models and technologies, 2006.

[10] Ahn, G.-J. and R. Sandhu, "Role-based Authorization Constraints Specification", ACM Transactions on Information and System Security, Vol. 3, No. 4, 2000, pp. 207-226.

[11] Bertino, E., P.A. Bonatti, and E. Ferrari, "TRBAC: A Temporal Role-Based Access Control Model", ACM Transactions on Information and System Security, Vol. 4, No. 3, 2000, pp. 191-233.

[12] Joshi, J.B.D., et al., "A generalized temporal role-based access control model", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 1, 2005, pp. 4-23.

[13] C.Moon, et al., "Symmetric RBAC model that takes the separation of duty and role hierarchies into consideration", Computers & Security, Vol. 23, No. 2, 2004, pp. 126-136.

[14] Strembeck, M. and G. Neumann, "An Integrated Approach to Engineer and Enforce Context Constraints in RBAC Environments", ACM Transactions on Information and System Security, Vol. 7, No. 3, 2004, pp. 392–427.