Design and Implementation of Access Control and Delegation Model in Road Transport Management System

Xingang, Wang

Design and Implementation of Access Control and Delegation Model in Road Transport Management System

SHI Qianzhu [1,2]
1.School of Information Science and Engineering
Yanshan University
Qinhuangdao 066004, China
shi_qianzhu@126.com

WANG Xingang[2], LIU Zhipeng[1]
2.Faculty of Engineering and Computing
Coventry University
Coventry CV1 5FB, United Kingdom
Xingang.Wang@coventry.ac.uk, victor_lzp@126.com

Abstract—**Software security is known to be a primary bottleneck in web-based software system. This paper describes how the Role-Based Access Control (RBAC) model can be applied to a road transport management system and demonstrates the design and implementation of the RBAC model in detail. In addition, a time limit and trust degree decreasing delegate authorization model is also proposed in order to give support to task allocation in the domain of time and to enhance the efficiency of cooperation between different departments and extraction of data. This design enables the system to be secure, reliable and convenient in daily use and it has been running stably and effectively for some time. It shows that the above design ensure security of the system and flexibility of authority management and meet current security standards in open environment.**

Keywords- Road Transport Management; RBAC; Delegation

I. INTRODUCTION

Access control is an important method to ensure security of a software system. It indicates what operations one can do and what one can't do. The goal of access control is to prevent any unauthorized access to system resources (such as computing resources, communication resources or information resources). Among all the access models, Role Based Access Control (RBAC) is the most influential one, which has become a general method in this area[1,2]. Compared to any other access control models, RBAC model can not only simplify the strategy set, reduce the cost of authority management, but also can support the authorization constraints effectively.

The Road Transport Management System (RTMS) is based on the B/S(Brower/Server) structure which integrates services of freight transport, passenger transport and taxi service. The system includes three different kinds of tasks, every task needs collaborative work among different departments. The most challenging task is how higher level departments grant authority to lower level counterparts. Similar to most software systems, this system has potential security problems, especially in the authority management. Lack of consideration for system security will lead to dangerous unauthorized operation and make it hard to apply the system to open environment [3-5].

After intensive study of the Chinese Ministry of Transportation standard documents, we develop an integrated system including freight subsystem, passenger subsystem and taxi subsystem and we successfully apply RBAC model in it. Furthermore, we introduce a time limit and trust degree decreasing delegation authorization model in the domain of time in order to strengthen the safety control. The introduction of modern access control model and collaboration mechanism enables the system to become a successful case of applying RBAC model into an actual software application system.

II. SECURITY ANALYSIS OF RTMS

Road transport includes freight transport, passenger transport and taxi services. The end users of the system are scattered in three level departments: transport stations of all districts of the city, city service center and city transport bureau. Also includes some staff in taxi affiliated companies. Safety challenge of the system described as follows:

(1) Operation rights access control. A task can't be accomplished by a single department, but involves three different levels departments: management stations scatter in different district of the city, city service center and city transport bureau. So we have to control the operation rights authorization respectively.

(2) Sensitive data access control. Some tasks are quite special and we name them sensitive tasks because such tasks need to obtain permission from the relevant departments' leader. It is necessary to refine the authority control on these sensitive operations. For example, if the task is modifying the unique identification number of a vehicle or a vehicle owner's ID number, higher level authorization is needed because we have to record the responsible person who performs this operation.

(3) Business scope access control. Some staff's operation types are the same, but the processing of their operation are different. For instance, when business scope goes to issuing freight vehicle license for dangerous goods transport or create freight station, it need to be issued by the city transport bureau director---the highest authority. However, for other business scope, issuing tasks can be approved by lower level

This research was supported by State Scholarship Fund of China Scholarship Council (CSC)(201207815006) and The Science and Technology Research and Development Plan of Qinhuangdao City (201101A091).

departments. Therefore, for some special operation we need to control its access rights according to the business scope it belongs to.

Due to the total number of tasks in RTMS are quite large------nearly over one hundred items, authority maintenance workload is relatively tremendous. To solve the above problem, we need to use RBAC to reduce the intensive maintenance workload, and to produce delegate authorization mechanism to achieve grant rights, so as to ensure the security of the tasks and the data.

III. THE ANALYSIS AND DESIGN OF ACCESS CONTROL FRAMEWORK

A. CORE RBAC

We deploy core RBAC frame of ANSI RBAC to RTMS because it is more practical to apply it to applications. Core RBAC defines almost all the objects and their relations that basic RBAC should have, as shown in Figure 1.

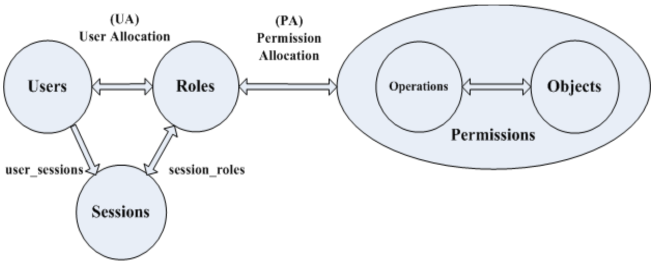

Figure 1. Core RBAC

Core RBAC defines three basic sets, including the set of roles, permissions and users. It is the role not the user who gets permissions. It creates a level of indirections between users and permissions through roles. Sessions are the platform for a user to get a role (roles), so sessions are the methods that a user be transferred to a role.

B. Design of RBAC Framework

In this paper, we design and improve the mapping and storage from the operating permissions to the roles. Permission itself is the same with the definition in core RBAC. The permission-role assignments have three types: the first one is Boolean, indicates whether or not a role has a specific permission; The second one is Numerical , shows to what degree a role owns a permission; And the third is Mixed, including multiple permissions.

There are four categories of objects in RTMS, including vehicles, owners, operators and business scope. The most important property of a vehicle is the district it belongs to(For example: District A or District B of a city) and its service type. For users we focus on the districts they belong to. For a operator we need to know the department he (she) works for. Business scope may need to focus for the rights to granting permissions to perform certain tasks.

For a role, operation permission is the start point of a task. Because a lot of tasks are quite similar, we design a uniform function for them. When we assign certain permission to a role we just modify parameters of the function. Therefore the process of creating a role is the instantiation of a specific task.

C. Analysis and design of a time and trust-degree-threshold based delegate authorization model

Delegation is an important security policy, its basic idea is that a user gives some or all of its permissions to other users, let the receiving authorized users instead of himself (herself) to perform certain tasks.

Roles once defined, with relative stability, like the enterprise or the company's management system once formed, will not randomly change relevant responsibilities. So when a role does not have the appropriate permissions but need to perform certain operations, delegation will be used[6]. Usually it happens in the following cases:

(1) In case of a user A goes on a business trip or gets ill, in order to continue carrying out his (her)responsibility of the work, delegation need to give his access rights to others. When user A returns, permissions should be returned to him(her).

(2) If user A could have some access rights in certain period of time or certain time point depends on if user B ‘s delegation is valid.

(3) In some cases, different groups of users need to work together and accomplish one task. Some members of a team have to be assigned certain access rights in order to work together smoothly.

As can be seen from (1), delegation is always temporary because the expiration date of delegation is limited, the number of use is limited, and it must happen on a certain point. From (2) we can infer that delegation is time sequence dependent. For example, a temporary access right, allowing operating only once, but authorization for the object is two users A and B. If user A has used the access right, then this access right is no longer valid and it can’t be use by user B. From (3) as you can see, the master role tends to expand the access permissions. If granting rights are given to user A , then A may authorize these access rights to other users. Thus from the perspective of the user A, with the increase of entrust depth, a object is more and more far away from authorized source and the credible degree is lower and lower. Therefore, we should control entrust depth.

The essence of delegation is that some users who own the grant privilege do approval operating on some specific tasks. Approval in the system includes designating agents, regulating valid time, specifying a service object and service type. An authorized user is entitled to execute certain operation on some specific objects after login during some specific period of time.

Some special tasks such as modifying vehicle information and road transport license need restrictions. A car, for example, in the final gets its license after going through all kinds of formalities from four departments. The process is the result of all the four departments’ jointly permission. So data is needed to be changed, it is necessary to seek approval from all the four departments, otherwise it is likely to cause confusion among vehicle data, there is no guarantee for the normal management order.

In normal task process, relevant formalities are needed to ensure the legitimacy of the process. To modify the data of a vehicle, an application should be submitted first, then modify data after getting permissions.

For vehicles which have got licenses, modifying their technical parameters and identification information means there are mistakes. It is an abnormal status no matter from which layer error come. In such situation, permissions are demanded if willing to continue the process, so as to ensure the process nationally and logically.

In addition, overlarge or overweight vehicles are all unable to get normal maintenance data like regular vehicles. So in such case delegate authorization is also needed. After all, maintenance data of such vehicle belong to the exception, so getting special permissions to carry on the process conforms to normal management process.

IV. IMPLEMENTATION OF ACCESS CONTROL AND DELEGATION AUTHORIZATION MODEL

A. Implementation of RBAC

The core of the RBAC design is the definition of the set of roles. In RTMS, the definition of the role is in accordance with its departments. Firstly, the level of department is considered. There are three levels of departments: Transport Stations scatted in all the districts of the city (TS), City Service Centre (CSC) and City Transport Bureau (CTB). Each TS is lower than CSC, CSC is below the CTB. From requirement analysis, we can put most access rights in CSC and CTS only manages some sensitive permissions. According to different districts and different service types we design the roles as follows: District A Freight Operator, District B Taxi Operator and so on. We can know their job responsibility even from the name of the role. So we design the role layer scheme as shown in Figure 2.

In Figure 2, vehicle service type, delegate authorization and districts property are affiliated permissions. There are unable to be allocated to roles without specific tasks in detail.

We create 7 roles, including District A Freight Operator, District B Passenger Operator, District C Taxi Operator, Passenger Director, Freight Director, Taxi Director and Service Agent.

B. Analysis and design of a time and trust degree threshold based delegation model

1) Modeling

RT0 is a model for describing delegation between entities, mainly in the distributed environment. But it has some limitations, which is unable to control depth and valid time length of the delegation. Subsequent versions of RT0 made a lot of great improvements but still has not solve the two main problems of it. Unfortunately the two main problems are exactly what we most frequently used in daily management work. Some articles propose role compatible models for delegation, but the delegation granularity is still too coarse which is still not suited for delegation with property assignment. This paper expands RT0 to a delegation authorization model with a trust degree threshold and time limitation.

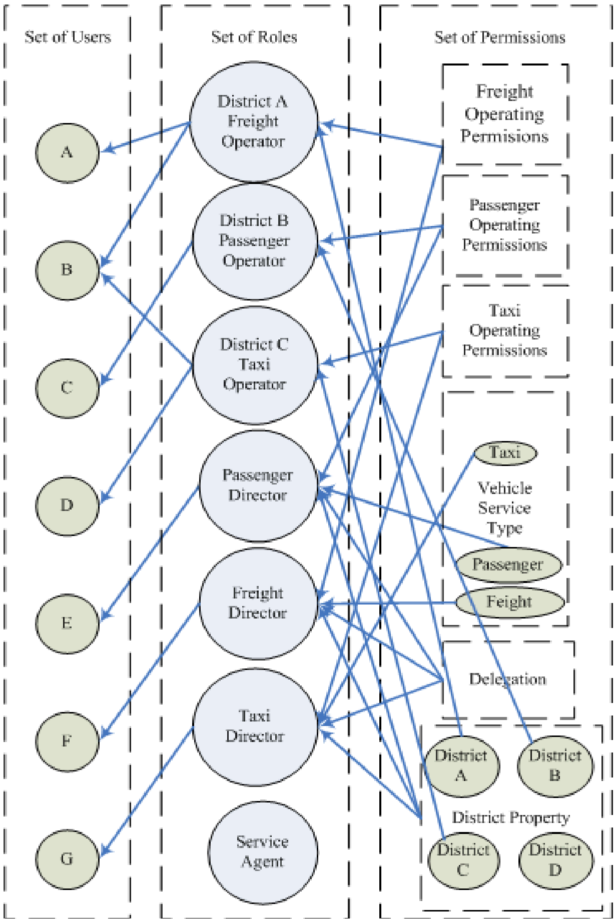

Figure 2. Role Layer Scheme

Definition 1 (Trust degree, Trust Degree Threshold, Trust Degree Threshold Attenuation Coefficient): TS for set of trust set, TMS for set of trust degree threshold, TLS = TTS = [0, 1]. Any t ∈ TS, if t = 1 indicates that the entity is fully credible; T = 0 means the entity is not to be trusted.TRS = [0, 1] is definition domain for trust degree attenuation coefficient.

Definition 2 (Time Limit): binary relation tl=(ts,te) is a time limit relation, ts is the starting time, te is the finish time , ts⩽te. te could be ∞, indicate there is no finish time. LS={tl1,tl2,…,tln} show set of time limit. For tl∈LS, tl is valid if and only if the current time tcur meet ts⩽tcur⩽te. valid(tl) is the function to test and verify if tl is valid, return 1 when tl is valid, otherwise return 0.

Definition 3 (Role/Permission assignment with trust degree threshold and time limit): Role/Permission assignment with trust degree threshold and time limit is PAT⊆ROLES×PRMS×TMS×LS. Set (r,p,t,tl)∈PAT, so an entity with trust degree t'(t'⩾t) may apply permission p when it get role r with a valid tl. Trust degree threshold value and time limit is constraints for the use of permissions.

Definition 3 describes how to assign permission to a role, especially when many entities own the same role. When delegate and give access right to a single user, we can create an agent role, then make the delegation to the agent, finally assign that role to the user.

Definition 4 (Trust degree threshold attenuation table on role layer): DCS⊆{DRH×TDCS} shows a relationship set made up of direct inheritance relationship and trust degree threshold attenuation coefficient relationship set. TDCS expresses the set of trust degree threshold attenuation coefficient. If role r assign permission p to r1, that is ((r1,r),dcs)∈DCS∧(r,p,t,tl)∈PAT. The trust degree to p of r1 is dcs*t*valid(tl).

Definition 5 (Trust degree threshold attenuation function in role layer): The attenuation coefficient function for direct inheritance relationship is fGetDcsValue(drh:DRH). The attenuation coefficient function for indirect inheritance relationship is fGetComDcsValue(l:RH)。

We can infer easily (drh,fGetDcsValue(drh)) ∈ DCS. Suppose r and r' are roles not having inheritance relationship, l=(r,r') ∈ RH, then the coefficient value from fGetComDcsValue(l) function is the minimum multiple multiply result of the path among them.

2) Implementation of delegate authorization

At the same time delegation authorization is finished, a trust degree threshold attenuation path is also formed. After activated a role is requesting for permission, the system will check the authorization certificate. Under the initial state, initial value is 1, accumulate attenuation coefficient in every certificate, so as to get a trust degree value, circulating until no delegation certificate left. Compare trust degree value with permissions confidence threshold value. If the trust degree value is greater than or equal to the corresponding threshold, the request permission license its performance; if less than the threshold, refused to authorize. Role authorization table is as shown in TABLE 1.

TABLE I. ROLE-PERMISSION ALLOCATION TABLE

Name of Role	Permissions	Threshold
District A freight operator	Freight operating permissions	0.8
District B Passenger operator	Passenger operating permissions	0.8
District C Taxi operator	Taxi operating permissions	0.8
Passenger director	Delegation permissions	0.9
Passenger director	District property	0.7
Freight director	Delegation permissions	0.9
Freight director	District property	0.7
Taxi director	Delegation permissions	0.9
Taxi director	District property	0.7

Figure 3 shows a simple direct inheritance relationship of some roles. Each arrow indicates directly inheritance relationship. The weights on the arrow show trust degree attenuation coefficient. For example, taxi director delegates rights to a District A freight operator, the coefficient is 0.8, other routes are same meaning. Setting principle for attenuation coefficient is: having direct inheritance relationship with the directors decays relatively small, the operator is relatively large. Because we believe that in a department senior managers compared with the ordinary staff has a higher degree of confidence.

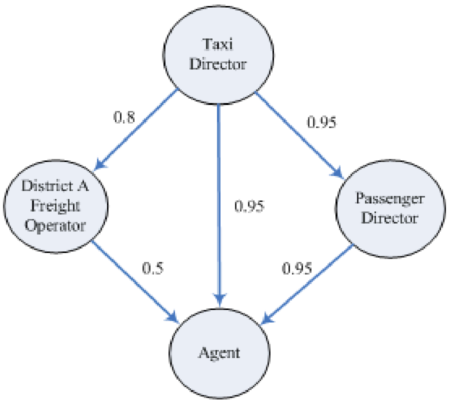

Figure 3. Trust-Degree Threshold Attenuation Relationship

According to the attenuation function in definition 5, if the agent role get operation permissions from District A freight operator, the trust degree value is minimum, which is 1* 0.8 *0.5 = 0.4, less than threshold 0.8, so the agent role owning authorization is not credible, thus realize the restrictions on terminal operator's ability to assign permissions outward. If the agent role gets operation permissions from taxi director, the trust degree value should be 1*0.95 = 0.95 which is greater than the threshold value of 0.8, so to be believed. Trust attenuation relations also need initialization and maintenance work; the system administrator can adjust it according to the actual situation.

V. CONCLUSIONS

In this paper, we discuss the security problems in RTMS in detail. In order to guarantee safe access control to the system, we propose an access control model, set up its framework and add a time and trust-degree threshold delegation authorization model. With these design, management for the access right is more flexible, safer. Furthermore, sensitive data is managed very well.

Practice show that the above design ensures the system running smoothly and effectively for a long time. RTMS is quite import for regular running of the city transport market and the management effective the stability of the urban road transportation market order, improve the efficiency of transportation management department management play a positive role. And for the widely application of RBAC model provides a valuable reference of reality.

REFERENCES

[1] Manisha Sharma, Shamik Sural, Jaideep Vaidya. AMTRAC: An administrative model for temporal role-based access control. Computers & Security, 39(2013),pp.201-218

[2] Bertino E., Bonatti P.A., Ferrari E. TRBAC: A temporal Role-Based access control model. ACM Transactions on Information and System Security, 4(3),pp.191-223, August 2001

[3] Wang Huiqin, Li Ming, Li Xiaoli. Security study in management information system. Chinese Journal on Computer Engineering and Application, 2001,(10),pp.91-93

[4] Wang Ming. Website security and solve of ASP.NET and SQL Server. Chinese Journal on Computer Security,2007,(5),pp.77-78

[5] M. Morrison, J. Morrison, A. Keys. Integrating Web Sites and Databases. Communications of the ACM, 2002,45(9),pp.81-86

[6] Zhang Hong, He Yeping, Shi Zhiguo. A Delegation Model for Periodicity Constraints-Based DAC. Chinese Journal Of Computers,2006,29(8),pp.1427-1437