AN EFFICIENT ROLE-BASED ACCESS CONTROL APPROACH IN IRBAC 2000

XIAOPU MA1,∗, NING CHENG1, LI ZHAO1 AND RUIXUAN LI2

1School of Computer and Information Technology
Nanyang Normal University
No. 1638, Wolong Road, Wolong District, Nanyang 473061, P. R. China
∗Corresponding author: mapxiao@nymu.edu.cn

2School of Computer Science and Technology
Huazhong University of Science and Technology
No. 1037, Luoyu Road, Hongshan District, Wuhan 430074, P. R. China
rxli@mail.hust.edu.cn

Received May 2013; accepted August 2013

ABSTRACT. The Interoperable Role Based Access Control Model 2000 (IRBAC 2000) provides us with a model for the interoperability between Role-Based Access Control (RBAC) secure domains. It translates the roles from foreign domains to local domains so that the foreign users can get the local access authorization. On the basis of this model, we present an efficient RBAC approach which is based on artificial neural network in order to calculate the permissions for the foreign users instead of searching the role-permission assignments directly. This approach considers the hierarchical relationships among roles and trains the artificial neural network using selected samples of roles as input vectors, and the according permissions as output vectors which can improve the efficiency of access control for the foreign users. Experiments on performance study prove the superiority of the method.

Keywords: Role-based access control, IRBAC 2000, Artificial neural network

1. Introduction. Role-Based Access Control (RBAC) [1,2] is the most popular access control model, and widely deployed as an alternative to traditional discretionary access control (DAC) [3] and mandatory access control (MAC) [4] in enterprise security management and enterprise management products. Perhaps the most distinctive and important feature of the RBAC is the desire to specify and enforce the enterprise-specific security policies in a way that maps naturally to an organization’s structure. Its emphasis on controlling who has access to operations on what objects is fundamentally different from information flow security in multi-level secure systems [5]. In this security model, permissions are no longer assigned individually to users, but as a set of permissions through roles. This change in how permissions are administrated often reduces the complexity of access control because the number of roles in an organization is significantly smaller than that of users [6]. Furthermore, it can support three well-known security principles: least privilege [7,8], separation of duties [9] and data abstraction. As a result, RBAC has been implemented successfully in a variety of commercial systems, such as insurance company and bank, and has become the norm in many applications.

Now with the rapid development of network technology and distributed applications, information interaction and cooperation in multi-domains have become increasingly frequent. In order to accomplish the interoperation problem, Kapadia et al. [10] presented the Interoperable Role-Based Access Control Model 2000 (IRBAC 2000) which provides a secure interoperability using dynamic role translation based on RBAC. In this model, the foreign users need to directly search the role-permission assignments in order to get the local access authorization. However, these techniques are time consuming and costly.
when there are dozens of roles, permissions and tens of thousands of users between the multi-domains. Thus, we present an RBAC approach based on neural network in order to get the local access authorization efficiently.

The remainder of this paper is organized as follows. We discuss related work in Section 2. The limitation in existing applications for getting the local access authorization drives our motivation and Section 3 proposes our notation about how to train the artificial neural network using selected sample of roles as input vectors and the according permissions as output vectors. A summary of our experimental results on simulated data is discussed in Section 4. Finally, Section 5 provides some insight into our ongoing and future work.

2. Related Work. Recently, considerable attention has been paid to researching and addressing the security needs of commercial and civilian government organizations. In those types of access controls, DAC requirements have been perceived as being technically correct for commercial and civilian government security needs, as well as for single-level military systems. MAC is used for multi-level secure military systems, but its use in other applications is rare. However, each organization has unique security requirements, many of which are difficult to meet using traditional MAC and DAC controls. As a result, the RBAC model can meet with these demands. The concept of RBAC began with multi-user and multi-application on-line systems pioneered in the 1970s. The central notion of RBAC is that permissions are associated with roles, and users are assigned to appropriate roles [11]. The most distinctive and important feature of the RBAC is the desire to specify and enforce enterprise-specific security policies in a way that maps naturally to an organization’s structure [12].

However, with the rapid proliferation of Internet and related technologies has created tremendous possibilities for the interoperability between domains in the distributed environments [13]. Interoperability provides a means for domains to share resources and services, which enhances performance and resource utilization. Furthermore, the interoperability does not come easy as it opens the way for several security and privacy breaches. Secure interoperation between domains is a crucial technique of resource sharing and security in the distributed environment. With respect to establish interoperability between domains, Kapadia presented an IRBAC 2000 model of secure interoperability using dynamic role translation based on RBAC. However, the traditional approach is costly and time-consuming since the number of roles, users and permissions in the multi-domains may be very large.

To this aim, this research tries to use artificial neural network to provide the local access permissions for the foreign uses in order to improve the mapping efficiency in a feasible way. Our focus is on how to train the artificial neural network and how to generate the permissions based on the foreign users mapping roles. The experimental results are tested to show the effectiveness of our findings.

3. Searching Permissions Based on Artificial Neural Network.

3.1. Preliminaries. This section gives a precise description for the IRBAC 2000. It assumes that there are four countably infinite sets: $R$ (the set of all possible roles), $U$ (the set of all possible users), $P$ (the set of all possible permissions), and $D$ (the set of all possible domains).

Definition 3.1. (IRBAC state). An IRBAC state $\gamma$ is a 4-tuple $(UA, PA, RH, RP)$, in which the user assignment relation $UA \subseteq U \times R$ associates users with roles in the local domain, the permission assignment relation $PA \subseteq R \times P$ associates roles with permissions in the local domain, the role hierarchy relation $RH \subseteq R \times R$ specifies an acyclic relation among roles, and the role mapping relation between multi-domains $RP \subseteq RL \times RF$ reflects role mappings between local domain roles $RL$ and foreign domain roles $RF$. 
An IRBAC state $\gamma = \{UA, PA, RH, RP\}$ determines the set of roles of which each user is a member, and the set of permissions for which each user is authorized, and the set of role mappings for establishing a flexible dynamic role translation between different domains. Now we use an example to illustrate the concepts in this paper. This example is shown in Figure 1.

![Figure 1. Associations between multi-domains](image)

In Figure 1, there are two domains, one is the local domain, the other is the foreign domain. The interoperation between these domains is achieved by introducing role mapping between the local domain and the foreign domain. There are two types of role mapping. One is transitive associations, such as $Professor_{Foreign} \rightarrow Committeeman_{Local}$ which is labeled as 1 in Figure 1. In this situation, the role $Professor_{Foreign}$ from the foreign domain will be translated to the role of $Committeeman_{Local}$ in the local domain. This role mapping also implies that all the ancestors of role $Professor_{Foreign}$ from the foreign domain will map to the role $Committeeman_{Local}$ in the local domain. The other is non-transitive associations, such as $AssoProfessor_{Foreign} \rightarrow Secretary_{Local}$ which is labeled as 2NT in Figure 1. In this situation, the role $AssoProfessor_{Foreign}$ will be translated to $Secretary_{Local}$ and deny $Professor_{Foreign}$ and $Administrator_{Foreign}$ from inheriting this association[14].

Here we can use an $m \times n$ matrix $M$ to describe the relationships between roles and permissions where $m$ is the number of roles and $n$ is the number of permissions for the local domain. The element $M\{i, j\} = 1$ denotes that the $i$th role has the $j$th permission or the $j$th permission belongs to the $i$th role; otherwise, the element $M\{i, j\} = 0$ indicates that the $i$th role has not the $j$th permission. We use $fu_i$ to indicate that the $i$th foreign user and $UserRoles(fu_i)$ ($i = 1, \ldots, k$) to indicate the set of mapping roles in the local domain for the $i$th foreign user.

### 3.2. Algorithms

This section presents the details of the new approach using Artificial Neural Network to obtain the foreign users’ access permissions. We use Back Propagation Neural Network (BPNN) [15] for training the role-permission assignments. Figure 2 shows the structures of the BPNN. It consists of three layers: input layer is applied to the inputs of the information; and signals propagate through the hidden layers to the output layer. Each link between neurons has a unique weighting value. In this figure, $x_1, x_2, \ldots, x_n$ means the input information, $w_{i,j}$ means the weight between the $i$th neuron in input layer
and the $j$th neuron in hidden layer, $v_j$ means the weight between the $j$th neuron in hidden layer and the neuron in output layer, and $y_1, y_2, \ldots, y_n$ means the output information.

Based on the artificial neural network, the process of the efficient role-based access control approach in IRBAC 2000 can be decomposed into two sub-processes:

(1) Training Stage: In this stage, we analyze the relationships between roles and permissions in the local domain, especially the hierarchy among roles, then choose roles in $R$ as the input sample for the neural network, permissions $P$ as the outputs of the network correspondingly to train the BPNN;

(2) Operation Stage: In this stage, we achieve UserRoles of the foreign user, then receive the role encoding of the foreign user. Furthermore, we regard the role encoding as the input vector. Finally, we can get the output vector as the access permissions for the foreign user.

The following example demonstrates the idea of the proposed method. Assume a hypothetical local organization has 7 roles. Table 1 shows the encoding of roles and the relationships between roles and permissions. In the training stage, we use $\langle 1000000 \rangle$, $\langle 0100000 \rangle$, $\langle 0010000 \rangle$, $\langle 0001000 \rangle$, $\langle 0000100 \rangle$, $\langle 0000010 \rangle$ as the input vector, $\langle 11111111111 \rangle$, $\langle 11111011000 \rangle$, $\langle 00111101000 \rangle$, $\langle 0000000111 \rangle$, $\langle 0000101111 \rangle$ as the output vector respectively to train the BPNN. In the operation stage, the system will obtain the permissions of the foreign user when setting the foreign user’s mapping roles UserRoles as the input information for the artificial neural network. Hence, the foreign user can obtain the access authorization in the local domain through the dynamic role mapping efficiently. For example, the input information is $\langle 0110000 \rangle$ when the UserRoles($fu_i$) = $\{r_2, r_3\}$, the output information is $\langle 1111111000 \rangle$.

Table 1. Sample data for an example organization

<table>
<thead>
<tr>
<th>coding</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$p_3$</th>
<th>$p_4$</th>
<th>$p_5$</th>
<th>$p_6$</th>
<th>$p_7$</th>
<th>$p_8$</th>
<th>$p_9$</th>
<th>$p_{10}$</th>
<th>$p_{11}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$r_2$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_3$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_5$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_6$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$r_7$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
4. **Experimental Results.** In this section, we will implement the proposed method of obtaining the access authorization based on BPNN on an Intel(R) Core(TM) i5-2400 @CPU 3.1G machine with 4GB memory to evaluate how well our method performs using different metrics.

To study the performance of our method, we generate the synthetic test data as follows. First, we use for loop to create the relationships between roles and permissions. For each role, a random number of permissions are chosen. The value of each element in the matrix is randomly chosen as 0, indicating that the role has no such permission, or 1, indicating that the role has such permission. Finally, we randomly choose the role set as $UserRoles(fu_i)$ for the $i$th foreign user through the dynamic role mapping.

We present the evaluation of our method (BP method) with the directly searching role-permission assignments method (DS method). We are interested in two things: the accuracy of our method, and how quickly it finds it. Table 2 shows the test parameters.

<table>
<thead>
<tr>
<th>Name of data</th>
<th>Number of roles</th>
<th>Number of permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>data1</td>
<td>10000</td>
<td>100</td>
</tr>
<tr>
<td>data2</td>
<td>20000</td>
<td>100</td>
</tr>
<tr>
<td>data3</td>
<td>100000</td>
<td>1000</td>
</tr>
<tr>
<td>data4</td>
<td>200000</td>
<td>2000</td>
</tr>
</tbody>
</table>

**Figure 3.** Performance and accuracy comparison under the different data sets

Figure 3(a) shows the accuracy of the different algorithms under the different data sets. From the figure, we can see that both of the approaches have the same accuracy. Furthermore, we can also see that the accuracy of the two approaches is quite good, with the largest number of roles and permissions getting to 100%. Figure 3(b) shows the average search time under the different number of roles and permissions. We can see that our approach costs less time. If the number of permissions and roles in the role permission assignments is larger, the advantage of our approach will be more obvious. Hence, it can decrease the workload for the foreign user to obtain the access authorization in the local domain through the dynamic role mapping.

5. **Conclusions and Future Work.** RBAC has become the norm in many applications. Furthermore, the IRBAC 2000 provides us with a model for the interoperability between RBAC secure domains. Hence, how to get the local access authorization for the foreign users efficiently is a considerable task before the foreign users can access the objects in the local domain. Hence, in this paper, we present an efficient RBAC approach which is based on artificial neural network in order to calculate the permissions of the foreign users instead of searching the role-permission assignments directly. This approach considers the hierarchical relationships between roles and trains the artificial neural network using roles...
as input vectors, and the according permissions as output vectors which can improve the efficiency of access control. As a result, the proposed approach has superior performance to traditional methods in speed. For the future work, we will evaluate our technique with others especially the algorithms based on neural network.

**Acknowledgment.** This work is supported by National Natural Science Foundation of China under grants 61173170, 60873225 and 70771043, National High Technology Research and Development Program of China under grant 2007AA01Z403, Natural Science Foundation of Hubei Province under grant 2009CDB298, Innovation Fund of Huazhong University of Science and Technology under grants 2011TS135 and 2010MS068, CCF Opening Project of Chinese Information Processing, and Innovation Fund of Nanyang Normal University under grants ZX2013013. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

**REFERENCES**


