Short Paper: Towards a Location-Aware Role-Based Access Control Model*

Indrakshi Ray Lijun Yu
Department of Computer Science
Colorado State University
Fort Collins, CO 80523-1873
Email: {iray,lijun}@cs.colostate.edu

Abstract

With the growing use of wireless networks and mobile devices, we are moving towards an era where location information will be necessary for access control. The use of location information can be used for enhancing the security of an application, and it can also be exploited to launch attacks. For critical applications, a formal model for location-based access control is needed that increases the security of the application and ensures that the location information cannot be exploited to cause harm. In this paper, we show how the Role-Based Access Control (RBAC) model can be extended to incorporate the notion of location. We show how the different components in the RBAC model are related with location and how this location information can be used to determine whether a subject has access to a given object. This model is suitable for applications consisting of static and dynamic objects, where location of the subject and object must be considered before granting access.

1. Introduction

With the increase in the growth of wireless networks and sensor and mobile devices, we are moving towards an age of ubiquitous computing where location information will be an important component of access control. The traditional access control models, such as Discretionary Access Control (DAC) or Role-Based Access Control (RBAC), cannot provide such location-based access control. These traditional models need to be augmented so that they can provide location-based access.

Denning and MacDoran [2] and other researchers have advocated that location information can be used to provide additional security. For instance, a user should be able to control or fire a missile from specific high security locations only. Moreover, the missile can be fired only when it is in a certain location. For such critical applications, we can include additional checks, such as verification of the location of the user and the location of the missile, that must be satisfied before the user is granted access. With the reduction in cost of Global Positioning Systems (GPS) and infra-red sensors, this indeed is a viable option.

Using location information for providing security has its own drawbacks as well. For example, information about the location of a user can compromise his privacy. Alternately, malicious users can observe the presence of a person in a certain location and infer the activities being performed by the person. The use of location information must be carefully controlled to prevent malicious users from launching attacks. Such attacks may have disastrous consequences for critical applications such as the military. In short, a formal model is needed for performing location-based access control.

In this paper we show how RBAC can be extended to incorporate the concept of location. We illustrate how the different components in RBAC are related with location and how location impacts these different components. Finally, we show how this location information can be used to determine whether a user has access to a given object.

The rest of the paper is organized as follows. Section 2 shows how the different components of RBAC are related with location and the constraints that location-based access control imposes on these components. Section 3 mentions some work related to this area. Section 4 concludes the paper with pointers to future directions.

2 Extending RBAC to Incorporate Location-Based Access Control

First, we present our definition of location and the operations that can be performed on location information.

*This work was partially supported by AFOSR under Award No. FA 9550-04-1-0102

Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'0
0-7695-2369-2/05 $20.00 © 2005 IEEE

Definition 1 [Location] *A location Loc_i is a non-empty set of points $\{p_i, p_j, \ldots, p_n\}$ where a point p_k is represented by three co-ordinates.*

Definition 2 [Contained in Relation] *Location Loc_j is said to be contained in Loc_k, denoted as, $Loc_j \subset Loc_k$, if the following condition holds: $\forall p_i \in Loc_j, p_i \in Loc_k$. The location Loc_j is called the contained location and Loc_k is referred to as the containing or the enclosing location.*

Definition 3 [Equality Relation] *Two locations Loc_i and Loc_j are equal, denoted as $Loc_i = Loc_j$ if $Loc_i \subset Loc_j$ and $Loc_j \subset Loc_i$.*

With this background, we are ready to discuss how the RBAC components are related with location. This is shown pictorially in Figure 1. The multiplicity of these relationships are indicated by presence or absence of an arrowhead. The absence of an arrowhead indicates a multiplicity of "one" and the presence of arrowhead indicates a multiplicity of "many".

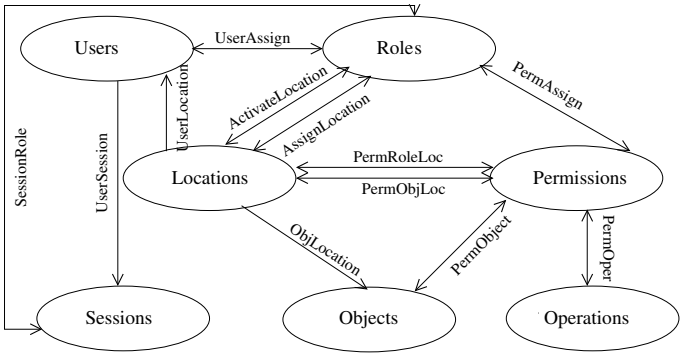

Figure 1. Relationship of RBAC entities with Location

2.1 Users

We assume that each user carries a locating device which is able to track his location. The association between user and location is indicated by the edge labeled *UserLocation* in Figure 1. Each user is associated with one location at any given instant of time. However, a single location may be associated with multiple users. We define a function $UserLocation$ that gives the location associated with a user. That is, $UserLocation(u)$ returns the location of user u.

2.2 Roles

In our model, roles are associated with locations. In fact, there are two kinds of associations roles can have with locations. These associations are indicated by the labeled edge *AssignLocation* and *ActivateLocation* in Figure 1.

Often times, the assignment of user to roles is location dependent. For instance, a person can be assigned the role of U.S. citizen only in certain designated locations. To get the role of conference attendee, a person must register at the conference location. Thus, for a user to be assigned a role, he must be in certain designated locations. In our model, each role is associated with a set of locations where the role can be assigned. We define a function $AssignLocation$ that takes as input a role and returns the set of locations in which that role can be assigned. A user u can be assigned the role r only if $UserLocation(u) \subseteq AssignLocation(r)$.

Some roles can be activated only if the user is in some specific locations. For instance, the role of audience of a theater can be activated only if the user is in the theater. The role of conference attendee can be activated only if the user is in the conference site. Each role is also associated with a set of locations from where this role can be activated. The function $ActivateLocation$ takes as input a role r and returns the set of locations where the role r can be activated. A user u can activate role r only if $UserLocation(u) \subseteq ActivateLocation(r)$.

2.3 Sessions

To create a session, a user submits a set of roles that must be activated in that session. If the location of the user satisfies the location constraints for all the roles he is trying to activate, then the session will be created. Otherwise he is notified and the session will not be created. The location associated with the session is the location associated with the user when he created that session. The function $CreateSession$ in the Core RBAC needs to be changed because the session creation depends on successful role activation, which, in turn, depends on the location of the user.

2.4 Objects

Objects can be physical or logical. For the sake of simplicity, we assume that each object is associated with one location only. Each location can be associated with many objects. This is shown by the association *ObjLocation* in Figure 1. The function $ObjLocation$ takes as input an object and gives the locations associated with the object. Formally, $ObjLocation(o)$ returns the location of object o.

2.5 Permissions

Location-based access control models provide more security than their traditional counterparts. This happens be-

cause the location of a user and that of an object are taken into account before making the access decisions. Location-based access control also allows us to model real-world requirements where access is contingent upon the location of the user and object. For example, a vendor may provide some free services only to the local customers. Our model should be capable of expressing such requirements.

In the Core RBAC model, permissions are associated with roles, objects and operations. These associations are labeled as $PermAssign$, $PermObject$ and $PermOper$ respectively in Figure 1. $PermAssign$ is a function that takes as input a permission and returns the set of roles assigned to that permission. $PermOper$ is a function that takes as input a permission and returns the set of operations associated with this permission. $PermObj$ is a function that takes as input a permission and returns the set of objects associated with the permission. A role can perform an operation on an object, if there is a permission that authorizes the role to perform the operation on the object. To incorporate location-aware access, we associate a permission with two locations: allowable role location and allowable object location. These associations are indicated by the edges labeled *PermRoleLoc* and *PermObjLoc* in Figure 1. The function *PermRoleLoc* takes as input a permission p and returns the set of allowable locations for the role associated with this permission. The function $PermObjLoc$ takes as input a permission p and returns the set of allowable locations for the object.

In location based access control, a user u is allowed to perform an operation op on an object o in a session s, if there is a permission p such that $PermAssign(p)$ includes an activated role r, $PermOper(p)$ includes op, $PermObject(p)$ includes o, the $UserLocation(u)$ is contained in $PermRoleLoc(p)$ and the $ObjLocation(o)$ is contained in $PermObjLoc(p)$.

3 Related Work

Denning and MacDoran [2] propose many motivating examples as to why location-based security is important for applications. Their argument is that use of location information can enhance the security of applications. The authors discuss how location-based authentication can be achieved using GPS and a tool called Cyberlocator.

Leonhardt and Magee [3] discuss how location-based access can be provided over existing matrix-based access control models and mandatory access control models. The authors do not discuss how the different components of an access control model are impacted by location and what constraints are necessary on the location-based model. In our work we try to address these issues and complement the above mentioned work.

Sampemane et al. [4] present a new access control model for active spaces. Active space denotes the computing environment integrating physical spaces and embedded computing software and hardware entities. Environmental aspects are adopted into the access control model for active spaces, and the space roles are introduced into the implementation of the access control model based on RBAC.

Covington et al. [1] introduce environment roles in a generalized RBAC model (GRBAC) to help control access control to private information and resources in ubiquitous computing applications. Each element of permission assignment is associated with a set of environment roles, and environment roles are activated according to the changing conditions specified in environmental conditions, thus environmental properties like time and location are introduced to the access control framework.

4 Conclusion

In this paper, we have proposed a location-based access control model that is based on RBAC. This model will be useful for pervasive computing applications in which the role of the user as well as his location will be used to determine if the user has access to some resource. We have shown how the different components of the core RBAC model are related with location, and what constraints are needed to perform location-based access. In future, we would like to extend this model by taking into account the effects of the constraints imposed by role hierarchy, static separation of duty, and dynamic separation of duty. We also plan to propose mechanisms by which such a model can be implemented. Typically location of a user or an object changes with time. Our future plans include proposing a model that takes into account these kinds of temporal constraints.

References

[1] M. J. Covington, W. Long, S. Srinivasan, A. Dey, M. Ahamad, and G. Abowd. Securing Context-Aware Applications Using Environment Roles. In *Proceedings of the 6th ACM Symposium on Access Control Models and Technologies*, pages 10–20, Chantilly, VA, USA, May 2001.

[2] D. E. Denning and P. F. MacDoran. Location-Based Authentication:Grounding Cyberspace for Better Security. In *Computer Fraud and Security,Elsevier Science Ltd*, February 1996.

[3] U. Leonhardt and J. Magee. Security Consideration for a Distributed Location Service. *Imperial College of Science, Technology and Medicine, London, UK*, 1997.

[4] G. Sampemane, P. Naldurg, and R. H. Campbell. Access Control for Active Spaces. In *Proceedings of the Annual Computer Security Applications Conference* , pages 343–352, Las Vegas, NV, USA, December 2002.