分布式环境下基于角色的访问控制①

Role Based Access Control in Distributed Environment

甘 泉 (上海中标软件有限公司 200233)
(中国科学院软件研究所信息安全工程研究中心 北京 100080)
韩乃平 (上海中标软件有限公司 200233)
贺也平 (中国科学院软件研究所信息安全工程研究中心 北京 100080)

摘要: 本文通过扩展权限的定义:给权限增加一个标示位来区别主体、访问的客体属于相同域和属于不同域时的权限,这样解决了采用对等角色时授予给非本域主体过大权限的问题。同时主体在访问非本域的客体时可以申请临时角色,这样避免了仅仅采用对等角色去访问非本域客体的简单化,更有利于最小权限的实现。通过这两点的改进,使基于角色的访问控制模型更加适合分布式访问控制的特点。

关键词: 访问控制 RBAC 对等角色 域

1 引言

基于角色的访问控制的基本思想是:用户被赋予角色,而权限不直接赋予用户而是赋予角色。用户通过担任某些角色而获得访问权限。这样就能极大地简化权限管理,减少管理访问控制策略的开销,并且易于描述和理解。

基于角色的访问控制1能够很好的应用在"紧耦合"型企业的访问控制中,并且人们已对此做了广泛的研究,在经典 RBAC96 模型的基础上做了很多改进和扩充,但是这些改进都是基于"紧耦合"型企业的假设,是一种集中式的访问控制。本文提出一种在"松耦合"型企业基于角色的访问控制模型,该模型解决在分布式企业中如何利用角色的概念实现访问控制,是对传统 RBAC 模型的一种改进。

2 已有的解决方法及其不足

正如前面所述那样,该类型企业的访问控制需要解决两方面的问题。对于第一个问题,由于基于角色的访问控制在主体和客体之间引入了角色的概念,所谓角色就是指一个或者一群用户在企业内可执行的操作的集合,系统通过角色来沟通主体和客体。【1】因而基于角色的访问控制与自主访问控制和强制访问控制相比更有利于模拟企业的组织结构,方便访问控制策略的制定和管理。我们可以为该类型企业选择基于角色的访问控制。对于第二个问题,由于经典的 RBAC96 模型是基于集中式访问控制假设的,它侧重于怎样控制属于同一个组织的主体如何去访问属于同一个组织的客体。它仅仅是把权限授予给角色,然后通过角色授予给主体来实现主体所具有的权限,而没有区别该角色是授予给与客体属于相同组织的主体还是授予给与客体属于不同组织的主体,也就是没有考虑主体客体的位置属性(即是否属于相同的组织)。虽然人们已对经典 RBAC96 模型做了很多扩充,例如引入了参数化角色、具有时间特性的角色访问控制、上下文相关的角色访问控制等等,但都是基于集中式访问控制假设的,而没有考虑分布式的特点,因而需要对 RBAC96 模型做进一步扩充,以便满足分布式访问控制的特点。在处理分布式访问控制的时候,人们提出了对等角色的概念,也就是属于组织 A 的主体 S1,他具有角色 R,如果 S1 对属于组织 B 的客体 O2 提出权限 P 的请求,则访问仲裁服务器认为 S1 在组织 B 中也具有对等角色(即角色 R),然后查看角色 R 是否具有对客体 O2 的

① 国家 863 项目资金资助:桌面操作系统及其配套环境,编号 2002AA 1Z2101,2004AA1Z2020

权限 P,以此做出接收或拒绝该请求的响应。虽然对等角色是一种解决分布式访问控制的方法,但是它过于简单,它只是简单的认为组织之间的同名角色应具有相同的权限,这将带来安全漏洞,不能很好的模拟现实世界。因此我们在本文中提出一种新的模型,强调主体客体的位置属性,区别角色授予给与客体属于相同组织的主体和授予给与客体属于不同组织的主体时的权限。同时对等角色没有给提出访问请求的主体以更多的角色选择。有的时候由于业务的需要,访问请求的主体只需要以比自己目前角色更低层次的角色去访问不属于自己组织的客体,如果这时仅仅采取对等角色的解决方案,那么主体只能以对等于自己目前的角色身份去获得权限,而不能选择低层次的角色身份,这样就会导致主体获得的权限大于实际需要的权限,不利于最小权限控制。

3 分布式环境下基于角色的访问控制模型

如上所述,基于角色的访问控制模型由于引入角色的概念,便于模拟企业的组织结构,简化了权限的管理,从而在企业的访问控制模型中得到了广泛的研究和应用。因而我们也选择基于角色的访问控制模型作为分布式环境下企业访问控制模型的基础,在经典 RBAC96 模型上做出一些修改,以适应分布式访问控制的特点。新模型主要做了两方面的修改:①引入域(domain)的概念,我们把域定义为在经济活动中具有相互合作关系的企业或组织。访问控制中的主体和客体具有域的属性,也就是它们一定隶属于某个域。把授予给角色的权限做出分类,一类是当该角色的被授予主体与访问请求的客体属于相同域时该角色具有的权限,另一类是当该角色的被授予主体与访问请求的客体属于不同域时该角色具有的权限。②在分布式访问请求时(即主体对属于与该主体不同域的客体提出访问请求时)允许主体申明自己以在对端域中的什么角色进行访问请求,而不是简单进行对等角色映射。

3.1 形式化定义

- user: =(username) 其中 username 表示该主体的名称
- U = {user1, user2, ⋯, usern} 表示所有用户的集合
- object: =(object_name) 其中 object_name 表示客体的名称
- O = {object1, object2, ⋯, objectn} 表示所有客体的集合
- role: =(role_name) 其中 role_name 表示角色的名称
- R = {role1, role2, ⋯, rolen} 表示所有角色的集合
- Ops = {operation1, operation2, ⋯, operationn} 表示所有操作的集合
- permission: =(operation, object_name, not_same_domain) 其中 not_same_domain 是该权限的属性,它是一个二值变量。not_same_domain 的语义就是用来标志当把具有该权限的角色授予给与该权限所描述的客体不属于同一个域的主体时,该主体是否具有该权限。not_same_domain 是一个二值变量,其取值范围是{0,1}当 not_same_domain =1 时表示当把具有该权限的角色授予给与该权限所描述的客体属于同一个域的主体时,该主体具有该权限;当把具有该权限的角色授予给与该权限所描述的客体不属于同一个域的主体时,该主体具有该权限。当 not_same_domain =0 时表示当把具有该权限的角色授予给与该权限所描述的客体属于同一个域的主体时,该主体具有该权限;当把具有该权限的角色授予给与该权限所描述的客体不属于同一个域的主体时,该主体不具有该权限。由此可以看出,当该权限所描述的客体与提出该权限请求的主体属于同一个域时,那么无论 not_same_domain 取何值,该主体都将具有该权限。而当该权限所描述的客体与提出该权限请求的主体不属于同一个域时,那么该主体是否具有该权限将依赖于该权限的 not_same_domain 属性值。
- P = {permission1, permission2, ⋯, permission} 表示所有权限的集合
- PA? PxR,从权限集合到角色集合的多对多映射,表示授予角色的权限
- UA? UxR,从用户集合到角色集合的多对多映射,表示授予用户的角色

3.2 主体和客体属于相同域的访问控制模型

这种情况下,新的访问控制模型与经典的 RBAC96 模型类似,管理员首先在数据库中创建一些与访问控制相关的表,如下:

表 1 主体－角色列表

User_name	Role_name	Issuer

该表的 issuer 列表示实施这次角色－用户授权的授权者。具体可参见 3.3 节。

表 2 角色－权限列表

Role_name	Permission_name

表 3 权限列表

permission	operation	object	Not_same_domain

其访问控制步骤如下：

(1) 主体向访问控制服务器提交访问请求。主体可通过访问请求函数向访问控制服务器提交访问请求。该函数定义如下：boolean access_request (string operation, string object, boolean same_domain)。该函数的第一个参数表示请求的操作。第二个参数表示请求的客体。第三个参数表示提出请求的主体与请求的客体是否属于同一个域，same_domain =1 表示提出请求的主体与请求的客体属于同一个域，same_domain =0 表示提出请求的主体与请求的客体不属于同一个域。因此在这种情况下 same_domain 的取值为 1。该函数的返回值是一个布尔值，1 表示请求成功，0 表示请求失败。

(2) 访问控制服务器查询数据库服务器。访问控制服务器判断传过来的请求的 same_domain 参数取值为 1，因而可以判定提出访问请求的主体与请求的客体属于相同域。查询表 1 得到授予给该主体的角色名，然后查询表 2 得到该角色所具有的权限名，最后查询表 3 判断该权限描述的 operation 和 object 是否与 access_request 函数的 operation、object 参数匹配，如果匹配则允许此次访问请求，否则拒绝此次访问请求。这儿由于主体和客体属于同一个域，因此在这儿不用考虑权限的 not_same_domain 属性。

(3) 数据库服务器把结果返回给访问控制服务器。

(4) 访问控制服务器把结果返回给提交请求的主体。

其模型示意图如图 1 所示：

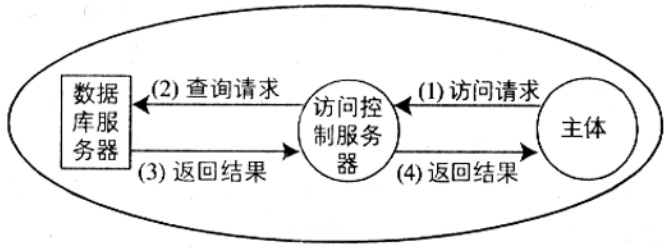

图 1 主体和客体属于相同域的访问控制模型

3.3 主体和客体属于不同域的访问控制模型

这种情况下，新的模型主要做两方面的改进。首先要改进对等角色仅仅以与访问主体对等的角色去访问不同域的客体的简单解决方法，其次是对与客体属于不同域和相同域的主体的权限应做出区分，具体访问控制步骤如下：

(1) 主体向要访问的客体所属域的域外角色申请管理服务器申请一个临时角色。主体要申请的该角色需要满足条件：要申请的临时角色在角色继承关系图中不能是该主体所属角色的父角色。该条件包含两方面含义：

① 要申请的临时角色在角色继承关系图中的层次不能高于目前该主体所具有的角色。该限制条件可以避免主体获取高于自己所应该获取的权限，避免安全漏洞。

② 要申请的临时角色不仅可以是该主体所属的角色，而且可以是该主体所属角色的子角色。因此要申请的临时角色可以根据需要选择较低层次的角色身份，便于最小权限的实现。因此该步骤可细分为：

· 主体向本域的域外角色申请管理服务器提出申请一个域外的临时角色。其请求语句为：SUBJECT request as ROLE in DOMAIN。例如：Alice request as PayrollSuper in Domain_a。

· 本域的域外角色申请管理服务器通过调用 is_child 函数查询本域数据库服务器，判断需要申请的临时角色与该主体所属角色在角色继承关系表中的层次关系，如果需要申请的临时角色是该主体所属角色或所属角色的子角色则返回 1，允许此次申请，否则返回 0，拒绝此次申请。Is_child 函数定义参见附录。

· 数据库服务器返回查询结果给本域的域外角

色申请管理服务器。

· 如果返回的结果是允许此次申请，则本域的域外角色申请管理服务器向对端域的域外角色申请管理服务器转发主体的申请临时角色请求。

· 对端域的域外角色申请管理服务器通过调用函数 GrantRoleToUser 向数据库中的主体 – 角色列表（该表定义参见表 1）插入一行，以分配一个临时角色给发起请求的主体。GrantRoleToUser 函数定义如下：void GrantRoleToUser (string user_name, string role_name, string issuer) 其第一个参数表示临时角色要授予的主体名，第二个参数表示授予的临时角色名，第三个参数表示这次角色授予的授权者。由于这种情况下该次角色授予的授权者是域外角色管理服务器，因此 issuer 取值为 RA，对应到表 1 的 Issuer 列，其取值也为 RA。当然如果角色授权的授权者是系统安全管理员，则对应到表 1 的 Issuer 列其取值应为 Administrator。所以表 1 的 Issuer 列如果取值为 RA，则表示授予给该用户的角色是一个临时角色；如果其取值为 Administrator 则表示授予给该用户的角色是一个永久角色。对于临时角色授权，系统在规定时间点上会激活一个函数来清理。

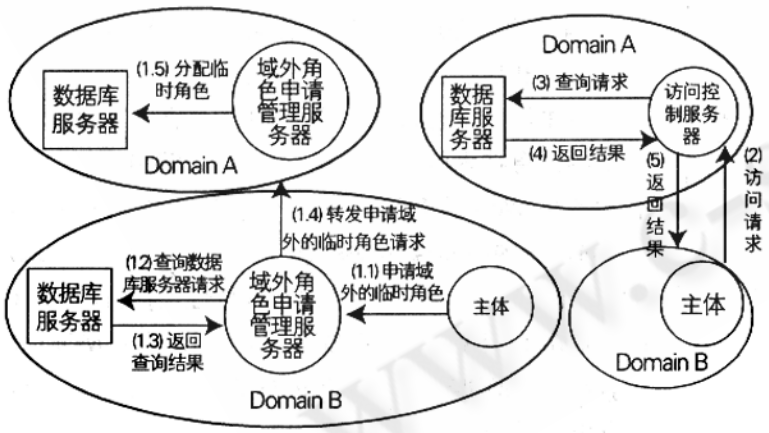

图 2　主体和客体属于不同域的访问控制模型

（2）主体向对端域的访问控制服务器通过 access_request 函数提交访问请求。由于这种情况下提出请求的主体与请求的客体不属于同一个域，因此 access_reauest 函数的 same_domain 参数取值为 0。

（3）访问控制服务器查询数据库服务器。访问控制服务器判断传过来的请求的 same_domain 参数取值为 0，接着查询表 1 得到授予给该主体的角色名，然后查询表 2 得到该角色所具有的权限名，最后查询表 3 判断该权限描述的 operation 和 object 是否与 access_request 函数的 operation、object 参数匹配，如果匹配并且表 3 的相应 Not_same_domain 列的值为 1 则允许此次访问请求，否则拒绝此次访问请求。

（4）数据库服务器把结果返回给访问控制服务器。

（5）访问控制服务器把结果返回给提交请求的主体。

其模型示意图如图 2 所示。

4　结论

该模型通过对 permission 定义的扩展，以此区分角色授予给与客体属于相同组织的主体和授予给与客体属于不同组织的主体时的权限，这样更好的模拟了现实世界，以及提出允许申请临时角色的概念对经典 RBAC96 模型进行了改进。这两方面的改进适应分布式访问控制的特点，使基于角色的访问控制应用范围从集中式访问控制领域扩展到分布式访问控制领域，更好的适应了目前企业规模不断扩大的需求。

参考文献

1　聂伯敏、熊桂喜，分布式环境下基于角色访问控制的实现，计算机工程，2002，28（8）：181 – 183。

2　吴和生、伍卫民、蔡圣闻，分布式环境下 RBAC 的高效实现，计算机工程，2003，29（6）：134 – 136。

3　吕宜洪，宋瀚涛，龚元明. 基于 RBAC 改进模型的角色权限及层次关系分析. 北京理工大学学报，2002，22（5）：611 – 614。

4　Sandhu R S, Coyne Ej, FeiMtein H L, Youman C E. Role – Based Access control Models. IEEE Computer, 1996, 29(2): 38—47.

5　Hristo Koshutanski, Fabio Massacci. An Access Control Framework for Business Processes for Web Services. ACM Workshop on XML Security, October 31, 2003: 15 – 24.