一种数据级安全访问控制方案①

唐 建, 徐 罡, 许舒人

(中国科学院软件研究所 软件工程技术中心, 北京 100190)

摘 要: 为了更好地保护 Web 应用系统中敏感数据不被非法访问. 在传统的基于角色的访问控制模型基础上提出了由用户集合和数据访问权限构成的数据访问策略, 并将数据访问策略关联到功能, 通过对原有业务 SQL 解析, 使用行级访问权限对数据记录进行行级过滤, 再根据列级访问权限对数据记录相应属性进行屏蔽处理来进行数据安全访问控制, 并设计了数据安全访问控制的框架. 最后将该方案应用到新发地农产品供应链管理平台中, 验证了该方案的可行性和有效性.

关键词: 数据访问策略; 行级权限; 列级权限; SQL 解析

A Solution of Data-Level Security Access Control

TANG Jian, XU Gang, XU Shu-Ren

(Technology Center of Software Engineering, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: To protect sensitive data in Web applications from unauthorized access, a data access strategy consisting of user set and data access authority is proposed, which is based on traditional role based access control model. The data access strategy is related to function. After parsing the original business SQL, row-level-rules are applied to filter the data records in row level, and column-level-rules are applied to mask the corresponding attributes of the data records. A data security access control framework is designed. Finally, this strategy is implemented in the Agricultural Products Supply Chain Management System of Xinfadi, and the validity and effectiveness of the presented strategy is demonstrated.

Key words: data access strategy; row-level-rules; column-level-rules; SQL analysis

访问控制是通过某种途径准许或限制用户对信息资源进行访问的一种安全技术, 它能保障有效授权用户访问系统资源, 而力图避免非法用户对有效资源的访问[1], 在 Web 应用系统中扮演着至关重要的作用. 一般地, 访问控制可分为功能级访问控制和数据级访问控制. 功能级访问控制很好的解决了系统中不同用户具有不同功能权限的问题, 已经有比较成熟的基于角色的访问控制模型(Role Based Access Control, RBAC)[2]. RBAC的基本思想是在用户和访问权限之间引入角色的概念, 将用户和角色联系起来, 通过对角色的授权来控制用户对系统资源的访问[3].

数据级访问控制主要解决的问题是: 具有同一功能的不同用户对数据资源的访问权限不同. 例如, 业务员和经理都有查询订单的功能; 业务员只能查询自己的订单信息, 经理可以查询所有业务员的订单信息. 对于数据级访问控制, RBAC 模型没有明确定义它的实现策略, 业界虽然有一些相关研究, 像访问控制列表[4]、基于属性规则的数据权限[5]、分级的行列级权限[6]等, 但没有一个通用的数据访问控制模型和解决方案. 而像新发地农产品供应链管理平台这样的Web应用系统有大量的敏感数据, 每个用户对系统中数据的访问有一定的限制, 如果恶意用户对数据进行越权或非法访问, 有可能会给其他用户带来经济损失.

1 相关工作

关于数据级访问控制, 国内外学者也做了一些相

① 基金项目:国家重点基础研究发展计划(973)(2009CB320704);国家高技术研究发展计划(863)(2012AA011204);国家科技支撑计划(2012BAH05F02)
收稿时间:2013-03-20;收到修改稿时间:2013-04-10

关工作. 主要有 Spring Security 安全框架中的访问控制列表(Access Control List, ACL)[4]. ACL 的思想是当用户向数据库新增一条记录时, 需要对这条记录设置访问控制属性(哪些用户对这条记录可以做哪些操作), 这样当用户对数据进行操作的时候, 根据访问控制属性判断该用户是否有权进行操作, 在删除记录的同时也要删除该记录的访问控制属性.

基于属性规则的数据权限模型[5], 通过定义实体对象的属性规则对数据访问进行约束, 并把对象属性规则约束关联到 RBAC 模型中的角色. 这样当拥有该角色的用户进行数据操作时, 会根据实体对象属性规则的定义把数据结果集中不满足的数据过滤掉, 形成新的数据结果集返回给用户.

分级的行列级权限[6], 通过在 RBAC 模型的权限和角色之间增加一层用户组进行分级授权, 通过对特定的 SQL 预定义行列级规则, 再把这些规则关联到用户组或者角色或者用户. 当用户进行数据访问时, 把原有 SQL 当成子查询, 把列级权限当成新的选择条件, 把行级权限当成新的查询条件, 形成新的 SQL 再进行数据访问.

以上模型在某些情况下不能很好的满足数据级安全访问的需求. 例如, ACL 当数据量比较大的时候, 由于新增或者删除记录都需要操作该记录的访问控制属性, 访问控制属性的开销会比较大, 性能也会有影响; 另外, 数据级访问控制不仅要考虑行级记录的访问控制, 还需要考虑列级属性的访问控制(例如, 经理无权查看业务员订单的客户信息), ACL 暂不支持列级的数据访问控制.

基于属性规则的数据权限模型, 如果结果集记录不包含属性规则定义的属性, 这种情况下将无法过滤无关数据. 例如, 订单实体对象有(订单号、订单金额、客户、订单录入人)等属性, 现在经理要查看订单金额大于 5000 的订单号与订单录入人之间的对应关系, 即结果集中每个订单实体只包含(订单号、订单录入人)这两个属性, 属性规则为订单金额大于 5000, 这时将无法根据属性规则对结果集进行过滤. 另外, 属性规则是关联到角色, 假设业务员角色同时有修改订单和查询订单这两个功能, 查询可以是任意金额的订单, 修改只能是金额小于 5000 的订单, 这种情况下如果给业务员角色定义属性规则为订单金额小于 5000, 会影响查询订单功能的数据访问.

分级的行列级权限, 行列级规则都是针对特定的 SQL, 并且预先定义好, 不方便规则的扩展. 另外, 由于这种方法会改变原 SQL 的选择属性, 导致不同用户结果集属性的不一致, 不便于结果集的统一处理. 例如部分用户可以查询的列级权限有(属性 1、属性 2、属性 3), 部分用户的列级权限只有(属性 1、属性 2), 这两类用户得到的结果集中属性是不一致的, 这将会影响到后续对结果集的处理.

2 数据安全访问策略模型

2.1 模型的建立

一般来说, 数据权限依赖于功能权限, 是对功能权限的进一步描述, 用来说明在指定的功能点上对具体某一个数据的权限控制[7]. 数据权限又与业务需求比较紧密, 随着业务发展、时间推移, 业务需求会不断发生变化, 数据权限也要随之变化(不断新增、修改或者删除数据权限), 所以数据级访问控制要具有明确性(要明确是哪个功能需要对数据访问进行控制)和灵活性(适应数据权限的不断变化).

在这样的背景下, 本文在传统的 RBAC 模型基础上提出了数据安全访问策略模型, 如图 1 所示.

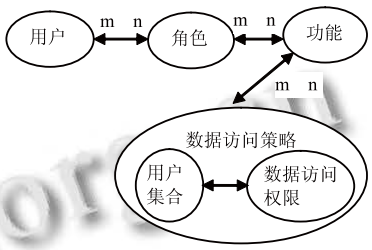

图 1 数据安全访问策略模型

对模型中的相关元素说明如下:

数据访问权限: 用户所拥有的数据权限, 分为行级权限和列级权限两类. 行级权限用来定义所能访问的业务数据的范围, 是一个 SQL 表达式. 例如, 只能修改金额小于 5000 元的订单对应的行级权限为: order.money < 5000. 列级权限用来定义数据记录的哪些属性不可以被访问, 用(表名.属性)的形式表示. 例如, 没有查看订单的客户信息的列级权限为: order.client.

用户集合: 具有相同数据访问权限用户的集合, 用 SQL 语句来描述. 例如具有经理角色的用户集合, 如图 2 所示.

```
select userId from userRole t1,role t2
where t1.roleId = t2.id
and t2.name = '经理'
```

图 2 描述用户集合的 SQL 语句

数据访问策略: 由用户集合和数据访问权限组成, 描述属于该用户集合的用户对数据记录具有怎样的访问权限. 例如, 数据访问策略(业务员只能查询自己的订单信息), 由用户集合(业务员)和访问权限(查询自己的订单信息)组成.

功能: 系统中的功能, 每个功能都对应一个 URL 路径. 比如查询订单功能的 URL(http://www.xxx.com:8080/xx.do?id=), xx.do 就是该 URL 的路径. 每个功能可以有任意条数据访问策略.

角色: 系统中定义的角色, 比如经理这个角色. 每个角色可以对应任意个功能.

用户: 系统中具体的用户, 比如系统管理员 admin. 每个用户可以拥有任意个角色.

例如, 对于查询订单功能的两种不同的数据安全访问策略: 业务员只能查询自己的订单信息, 经理可以查询所有业务员的订单信息, 但无权查询订单的客户信息. 包含两种用户集合: 业务员和经理; 三种数据访问权限: 查询自己的订单信息、查询所有的订单和不能查询订单的客户信息, 前两个数据访问权限属于行级权限, 最后一个属于列级权限.

2.2 模型的特点

数据安全访问策略模型通过把数据访问策略直接关联到传统 RBAC 模型的功能, 而不是把访问策略关联到角色或者用户, 因为每个用户或者角色都可能对应多个功能, 每个功能的数据访问权限不一定一样(上文提到的查询订单和修改订单的数据权限), 明确了数据访问策略对应的功能. 并将数据访问策略分成用户集合和数据访问权限两个部分, 可以同时为一个用户集设置数据访问权限, 这样不用为每个用户都设置安全访问权限, 提高了效率; 同时更加适应数据访问策略的灵活变化, 例如当用户集合没有发生变化, 而数据访问权限发生了变化, 只需修改数据访问权限部分即可; 另外用户集合的定义可以更灵活, 不局限在用户或者角色. 并将数据访问权限分成行级数据权限和列级数据权限, 来满足不同维度的访问控制需求, 例如有些需求是对行级记录进行访问控制, 有些需求则需要对记录的某些属性进行访问控制.

2.3 模型框架的设计

数据安全访问策略模型框架主要分为两部分: 设置安全访问策略和数据访问权限过滤. 设置安全访问策略, 如图 3 所示.

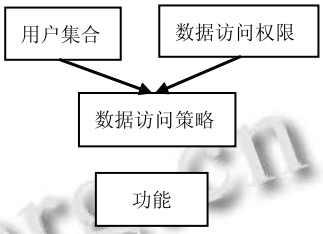

图 3 设置安全访问策略

对功能权限设置安全策略首先需要分别设置用户集合、数据访问权限(行级权限或者列级权限), 并将用户集合和数据访问权限组成数据安全访问策略, 然后将数据安全访问策略关联到某一具体的功能. 例如对于查询订单功能的两种不同的数据安全访问策略(业务员只能查询自己的订单信息, 经理可以查询所有业务员的订单信息), 需要设置两种用户集合(业务员和经理)和两种数据访问权限(查询自己的订单和查询所有的订单), 并将(业务员, 查询自己的订单)和(经理, 查询所有的订单)组成数据访问策略, 再将这两条数据访问策略分别关联到查询订单功能.

数据访问权限过滤主要有以下几个模块, 模块之间的调用关系用箭头指示, 如图 4 所示.

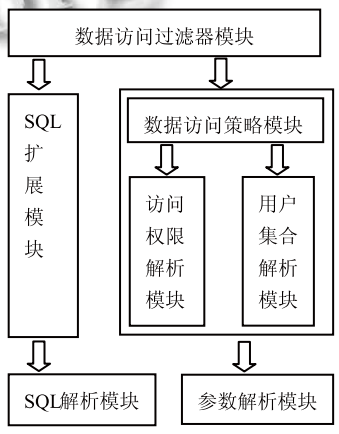

图 4 数据安全访问策略模型框架

2.3.1 数据访问过滤器模块

该模块是进行数据访问控制的入口, 主要功能是通过 Servlet 过滤器在用户进行数据访问前从

HttpRequest 中获取该请求的功能 URL，然后通过数据访问策略模块分别获取该功能 URL 路径对应的行级权限 SQL 表达式和列级权限列表，提供给 SQL 扩展模块使用.

2.3.2 数据访问策略模块

该模块的主要功能是根据功能 URL 路径和用户信息，获得该功能 URL 路径该用户所对应的行级权限 SQL 表达式和列级权限列表. 具体步骤如下:

① 根据 URL 路径获取行级的数据访问策略列表;

② 遍历行级数据访问策略列表，对每一条策略，先根据该条策略的用户集合和用户信息到用户集合解析模块判断该用户是否属于该用户集合. 若符合则根据该条策略的数据访问权限和用户信息到访问权限解析模块解析出具体的行级访问权限;

③ 拼接步骤②中所有符合的行级权限 SQL 表达式形成总的行级权限 SQL 表达式;

④ 根据 URL 路径获取列级的数据访问策略列表;

⑤ 遍历列级数据访问策略列表，对每一条策略，先根据该条策略的用户集合和用户信息到用户类型解析模块判断用户是否属于该用户集合; 若符合则根据该条策略的访问权限和用户信息到访问权限解析模块解析出具体的列级访问权限; 返回所有符合的列级权限列表.

2.3.3 用户集合解析模块

该模块的主要功能是根据用户信息和数据访问策略的用户集合判断该用户是否属于该用户集合. 首先查询描述用户集合的 SQL 语句，然后把用户集合 SQL 语句当作子查询，最后根据 SQL 查询结果判断用户是否属于该用户集合. 例如，判断用户是否属于经理角色，如图 5 所示.

```
select userId from user
where userId in (
    select userId from userRole t1,role t2
    where t1.roleId = t2.id
    and t2.name = '经理'
)
```

图 5　判断用户是否属于经理角色

2.3.4 访问权限解析模块

该模块的主要功能是根据数据访问策略的访问权限和用户信息解析出行级权限 SQL 表达式或者列级权限列表. 如果解析的是行级权限，首先查询描述行级访问权限的 SQL 表达式，然后通过参数解析模块把 SQL 表达式中的用户参数值串替换成相应的用户属性(在需要的情况下)，形成新的 SQL 表达式. 如果解析的列级权限，首先查询描述列级访问权限的串，然后解析成列表返回.

2.3.5 参数解析模块

该模块的主要功能是解析访问权限中和当前用户相关的值串，并把它们替换成相应的用户属性. 例如，只能查询自己的订单，如图 6 所示.

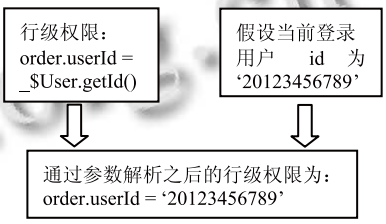

图 6　SQL 参数解析示例图

2.3.6 SQL 扩展模块

该模块的主要功能是根据数据访问过滤器模块传递过来的行级权限 SQL 表达式以及列级权限列表，并借助 SQL 解析模块对原有 SQL 解析后，进行行级与列级的扩展，然后再进行数据访问. 例如，只能查询金额小于 5000 的订单并且无权访问订单的客户信息，如图 7 所示.

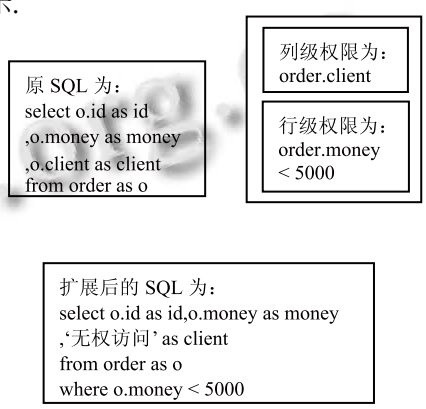

图 7　访问权限 SQL 扩展示例图

在行级扩展前，先判断原 SQL 的数据表是否包含行级权限中涉及的数据表，如不包含，则不能进行行级扩展; 如果包含，判断涉及的数据表是否有别名，如果有，把行级权限中的表名替换成对应的别名后，和原 SQL 进行拼接，拼接时需要根据原 SQL where 语句、group by 语句、order by 语句情况确定拼接位置.

在列级扩展前，先判断原 SQL 是查询(select)操作

还是更新(update)或者删除(delete)操作, 如果是更新或者删除则不支持列级扩展; 如果是查询操作, 先判断原 SQL 的选择列是否包含列级权限(表.属性)或者(表别名.属性), 如果包含则把列级权限替换成“无权访问”字样. 如果原 SQL 是(select * from)的形式, 并且涉及的数据表包含列级权限中的表, 则先把 SQL 选择语句重构成(表.属性)的形式, 再根据列级权限进行扩展.

2.3.7 SQL 解析模块

该模块的主要功能是借助 SQL 解析器[8]对原有 SQL 在扩展之前进行分析, 解析出 SQL 中所涉及的表、表和别名之间的映射关系以及 SQL 语句选择的属性列等信息.

整个数据访问控制流程: 当用户进行功能访问时, 在数据访问前需要通过数据访问控制过滤器, 该过滤器根据功能 URL 路径和当前用户, 并借助数据访问策略管理器分别解析出行级权限和列级权限. 若没有相应的数据访问权限, 则直接进行数据访问, 并返回结果. 反之, 在进行数据访问前, 对原有的 SQL 进行行级权限拼接后形成新的 SQL, 然后再根据列级权限列表对 SQL 选择语句中对应的属性进行屏蔽后, 执行 SQL 后返回结果.

3 实验

该数据安全访问策略模型已被应用到新发地农产品供应链管理平台中, 新发地农产品供应链管理平台是基于 Struts+Spring+Hibernate 架构的 Web 应用系统, 旨在为供应链会员营造一个农产品信息发布、交流、共享、协同作业的环境, 解决农产品日益严重的质量安全及流通问题.

实验采用的是测试数据和测试账号, 模拟的是经理可以查询所有业务员的订单信息和业务员只能查询自己的订单信息.

① 未使用数据访问策略前, 经理和业务员查询到的订单信息, 如图 8 所示.

订单号	金额（元）	客户	订单录入人
O20120921000001	5000	沃尔玛超市	ywy1
O20120930000002	3000	物美超市	ywy1
O20121030000003	6000	沃尔玛超市	ywy1
O20121115000003	4000	京客隆超市	ywy2
O20121116000004	7000	蔬菜批发市场	ywy2
O20121130000006	5500	北京饭店	ywy2
O20121220000012	8000	京客隆超市	ywy2

共 7 条记录，当前第 1 条到第 7 条。

图 8 未使用访问权限的订单信息

② 使用了行级数据访问策略(业务员只可以查询自己的订单信息)后, 业务员 ywy2 查询到的订单信息已不包括业务员 ywy1 的订单, 如图 9 所示.

订单号	金额（元）	客户	订单录入人
O20121115000003	4000	京客隆超市	ywy2
O20121116000004	7000	蔬菜批发市场	ywy2
O20121130000006	5500	北京饭店	ywy2
O20121220000012	8000	京客隆超市	ywy2

共 4 条记录，当前第 1 条到第 4 条。

图 9 使用行级权限后的订单信息

③ 使用了列级数据访问策略(经理无权查看订单客户信息)后, 经理查询到的订单信息的客户属性都显示“无权访问”, 如图 10 所示.

订单号	金额（元）	客户	订单录入人
O20120921000001	5000	无权访问	ywy1
O20120930000002	3000	无权访问	ywy1
O20121030000003	6000	无权访问	ywy1
O20121115000003	4000	无权访问	ywy2
O20121116000004	7000	无权访问	ywy2
O20121130000006	5500	无权访问	ywy2
O20121220000012	8000	无权访问	ywy2

共 7 条记录，当前第 1 条到第 7 条。

图 10 使用列级权限后的订单信息

4 结语

本文通过对数据级访问控制的分析, 在传统的基于角色访问控制的基础上, 设计了数据安全访问策略模型, 将用户集合和数据访问权限分开, 便于适应业务需求的变化. 并提出通过对原有 SQL 的分析, 对行级权限 SQL 表达式和列级权限列表进行替换后再进行 SQL 拼接, 使得访问权限更具有扩展性, 同时满足了数据访问不同维度的需求. 设计了模型的框架, 把模型应用到新发地农产品供应链管理平台中, 给平台的数据级访问控制带来了灵活性和更好的安全性. 目前还没有友好的界面支持数据安全访问策略的设置以及对该模型的性能进行分析, 这些将是后续工作的重点.

参考文献

1 李晶,李晓林,朱思斯.基于 RBAC 模型的多级权限访问控制设计.软件导刊,2009,8(4):140–142.

2 马林,黄文培,聂捷楠,汪凌峰.RBAC 的权限扩展和其在 Acegi 下的实现.微计算机信息,2008,24(2-3):34–36.

3 朱养鹏,张璟.SaaS 平台访问控制研究.计算机工程与应用,

(下转第 74 页)

该虚拟实践教学平台不但对酒店管理专业的学生在实践客房服务环节起了很好的帮助作用，而且大大方便了教师的教学管理工作. 同时为旅游学院教学提供了新的教学手段，既降低成本又增加趣味性，为后续酒店管理甚至旅游学院各个专业的虚拟实践教学打下了基础. 下一步的工作就是在本系统的基础上实现酒店管理的其他实践环节，如前台服务和餐饮服务等.

参考文献

1 宫勇,蒲小琼,张翔.虚拟场景漫游技术及其系统实现.计算机工程与应用,2007,43(15):89–91.

2 卞锋,江漫清,桑永英.虚拟现实及其应用进展.计算机仿真,2007,24(6):1–4.

3 曲丽荣,凌秀泽.远程虚拟实验室的研究与开发.计算机测量与技术,2011,19(11):2752–2754.

4 何增颖,陈建锐.基于虚拟技术的计算机实验教学.实验技术与管理,2012,29(1):79–82.

5 杨贵,李仁旺,刘海霞,张鹏举.基于 Web 的虚拟物流实验室设计与实现.计算机应用与软件,2009,26(3):21–23.

6 卜朱镇,韩秀玲.在线三维计算机组网虚拟实验室的研究与实现.计算机工程与设计,2012,33(7):2754–2759.

7 傅招国,王天威,倪小鹏,林砺宗.基于 Virtools 的虚拟现实技术及在特种设备教学中的应用.计算机工程与科学,2012,34(6):97–100.

8 刘志广,李艳芳,张永策,吕保和,罗丽萍.基于Virtools的三维交互虚拟啤酒灌装线的构建.计算机工程与设计,2009,30(23):5527–5530.

9 金勇进,吴产乐,叶刚.基于 Java3D 和 3DMAX 的虚拟实验元件建模与可视化研究.计算机应用研究,2010,27(7):2575–2578.

10 王海丰.基于 VRML 的虚拟酒店漫游系统建设研究.琼州学院学报,2011,18(2):26–31.

(上接第 85 页)

2011,47(24):12–16.

4 Domain Object Security (ACLs).http://static.springsource.org/spring-security/site/docs/3.0.x/reference/springsecurity.html.

5 欧阳荣彬,王倩宜,李丽,刘云峰.基于属性规则的数据权限模型研究与实现.大连海事大学学报,2010,36(2):81–83.

6 冯志亮,谭景信.分级的行列级权限系统的设计和实现.计算机工程与设计,2011,32(10):3274–3277.

7 林伟炬,刘列根,张宇.一个通用的权限管理模型的设计方案.微计算机信息,2009,22(15):1–3.

8 Antlr. http://antlr.org/.

9 蔡昭权.基于业务无关的权限管理的设计与实现.计算机工程,2008,34(9):183–185.

10 成富.使用 Spring Security 保护 Web 应用的安全.http://www.ibm.com/developerworks/cn/java/j-lo-springsecurity/.

11 杨柳,危韧,陈传波.一种扩展型基于角色权限管理模型(E-RBAC)的研究.计算机工程与科学,2006,28(9):126–128.

12 吴翰清.白帽子讲 Web 安全.北京:电子工业出版社,2012.205–219.