计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(7):215−220 [doi: 10.15888/j.cnki.csa.005927]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于时域反射数据库的访问控制①

龙晓泉

(中移互联网有限公司, 广州 510640)

摘　要: 数据库访问控制策略对用户定义了不同的权限. 为确保数据库系统的安全性, 需实施数据库安全策略以保护对数据库中数据的合法访问, 如同确保数据的完整性、一致性. 为了实现数据库访问控制策略的一致性, 在本文中, 我们提出了一个新的访问控制策略, 即时域反射数据库访问控制(TRDBAC), 旨在解决RDBAC对时间约束的局限性. 我们以一个学生信息系统为例, 展示了TRDBAC的研究结果. 最后, 我们分析了TRDBAC新模型的应用效果.

关键词: 访问控制策略; 访问控制; RDBAC; TRDBAC

引用格式: 龙晓泉.基于时域反射数据库的访问控制.计算机系统应用,2017,26(7):215–220. http://www.c-s-a.org.cn/1003-3254/5927.html

Access Control Based on Temporal Reflective Database

LONG Xiao-Quan

(China Mobile Internet Company Limited, Guangzhou 510640, China)

Abstract: The access control policy defines the rights and privileges for different users on database objects. In order to maintain the security of these database systems, the database security should be controlled to protect the contents of the access database, as well as to preserve the integrity and consistency of the data. In order to achieve consistent database access control strategy, in this paper, we propose a temporal reflection database access control (TRDBAC) , a new access control strategy, aiming to eliminate the limitations of RDBAC. To express the time limit our new strategy has demonstrated the results of a case study on a student information system, where strategies are written in the reflective database access control (RDBAC) for extended time based. Finally, we analyze the behavior of the new model.

Key words: access control policies; controlled access; RDBAC; TRDBAC

1　引言

数据库系统是任何系统中的核心部件之一. 伴随着数据的迅猛增长, 访问控制机制应具有足够的灵活性, 实现多种不同类型的访问控制策略.

数据库系统面临系统内外部的多重威胁. 这些威胁导致不同的数据安全漏洞, 例如未授权的数据访问, 不正确的数据修改等[1]. 理想的数据库系统应该是足够安全的, 以便它可以保护系统的重要数据不受任何类型的威胁.

在数据库系统中, 访问控制机制的主要目标是确保数据的机密性. 到目前为止, 学术界已经描述了许多类型的关系数据库访问控制系统[2]. 例如:

(1) 自主访问控制(DAC)

(2) 强制访问控制(MAC)

(3) 基于角色的访问控制(RBAC)

(4) 基于时间约束的角色访问控制(TRBAC)

(5) 反射数据库访问控制(RDBAC)

在本文中, 我们访问控制策略的实施基于RDBAC的学生成绩信息系统. 调取相关的数据记录(TD)写入访问控制策略中, 这些策略将通过SQL语句实现.

在该系统应用场景下, 我们需要考虑到时间的约束性. 由于RDBAC没有明确考虑时间约束. 因此, 我们扩

① 收稿时间: 2016-10-12; 收到修改稿时间: 2017-01-09

展了RDBAC, 并提出时域反射数据库访问控制(TRDBAC)的想法. 该模型将基于实时数据库的访问控制策略的设计与实现.

2 访问控制模型

2.1 自主访问控制(DAC)

在DAC中访问数据库系统的权限限制是基于访问或受控对象的身份, 他们属于授权规则[3]. DAC在某种意义上说, 它可以让管理员授予数据库访问控制等权限. 由于授予权限的灵活性, 自主访问控制模型是通过DBMS实现的, 即Oracle10g、MySQL等数据库管理系统, 并且由管理员设定相关的权限. DAC具有授权访问的灵活性, 由于访问控制的不当传递, 存在非法访问机密信息的缺点等.

2.2 强制访问控制(MAC)

MAC用于将系统中的信息分密级和类进行管理, 以保证每个用户只能访问到那些被标明可以由他访问的信息的一种访问约束机制[4]. MAC在RDBMS中的实现已经完全集中在多级安全(MLS). 多级安全(MLS)是一种功能, 允许有不同的分类信息, 以便在信息系统中允许具有不同的安全许可和授权的用户使用, 同时防止用户访问他们所没有授权的信息. 由于这种受限制的MAC已被用在大多数的商业系统应用中, 在一些商业系统中RDBMS提供了基于MAC地址的访问控制如Oracle、Informix在线/安全和Sybase SQL服务器的安全等. 除了限制MAC的一个重要优势, 这种安全产品的重要优点是它们可以用于定义标签访问策略和为用户分配访问标签[5].

2.3 基于角色的访问控制(RBAC)

在RBAC中, 策略是在用户方面, 对主题, 角色, 角色等级, 操作, 关系和约束的描述. 用户根据自己的能力和责任将成员授予角色. 用户必须激活一个角色并由安全管理员操作授权. RBAC的主题主要是它授权予管理员可以限制角色授权、角色激活, 执行和操作的能力. RBAC的一个因素是访问控制, 简化了访问策略的理解和管理的非任意性. 此外, 它有利于管理员在一个抽象的层次上控制访问[6].

2.4 基于时间约束的角色访问控制(TRBAC)

TRBAC是RBAC的扩展, 支持对角色的启用/禁用时间约束. 这种约束表示通过角色触发器(当指定的操作发生时自动执行的规则)也可以被用来约束一个特定的用户在一个给定的时间内激活角色的一组角色器. 触发器的触发可能会导致一个角色立即在一个明确的指定的时间被启用/禁用[7]. 基于TRBAC的权限特性, 可以在任何特定时间了解用户所属角色的状态, 例如某些触发器, 可以使用户属于多个角色, 而不是属于某一个角色.

2.5 反射数据库访问控制(RDBAC)

RDBAC是新兴的访问控制模型. RDBAC试图克服实施复杂的访问控制模型的困难. 如前所述当数据库使用概念简单的访问控制模型, 例如访问控制矩阵和角色的定义等, 并在此基础上允许用户执行读取、插入、更新或执行表, 视图等数据库资源. 但这些技术面临的弱点是, 在更细粒度的水平上, 数据库表中某些部分的访问控制, 需要被授予创建一个单独的视图, 其中包含所需的数据, 然后用户就可以访问该视图. 在某些情况下, 这些模型具有一定的灵活性, 用户可以与用户相关表的访问权限, 而不需要超级用户权限[7]. 然而, 这些模型仅限于表达策略的程度, 如“Alice可以查看Alice的数据”, “Bob可以查看Bob的数据”等, 而非策略的意图, 如“每个员工都可以在表中查看自己的数据, 每个用户都需要一个单独的表/视图, 以及一个单独的角色来访问视图”. 但是, 策略管理变得繁琐, 其中数据会变化很快, 修改数据库架构和新的用户可能会需要更快的速度[8].

RDBAC是一个模型, 其中一个数据库的权限表示一个数据库查询本身, 而不是一个包含在访问控制矩阵的静态的特权. 在这个模型中的访问控制策略的决定可以依赖于数据库中的其他部分, 如用户的属性, 被查询的数据的属性, 或用户之间的关系和数据. 在这个模型中做了优化以消除上述限制, 并允许策略来引用数据库的任何部分. RDBAC的主要优点是, 它有助于提高访问控制策略的表现. RDBAC的好处是“谁是管理者就可以查看他们所管理的员工的数据”, 假设我们授予员工管理者的角色来访问其他所有员工的数据. 当一个管理者查询该表中, 该策略将首先检查该用户是否确实是一个管理者, 然后检索管理者的部门, 并最终返回该部门. 这种方法有两个好处. 首先, 该策略利用已存储在数据库中的数据; 其次, 策略描述了其意图, 而不是它的程度. 因此, 当数据库更新时(例如, 当一名员工提升到管理者)时, 系统会自动更新权限, 以防止在不一致的状态下数据库更新异常[9,10].

RDBAC的缺点, 没有任何有效的方法为实际数据库系统访问控制策略的形成相应的数学模型. 为此RDBAC策略提供了一个强大的语法和语义, 但是它缺乏对某些常用操作中的SQL语法和语义进行大规模的执行和测试过.

一些产品如Oracle的虚拟专用数据库(VPD)是一个基于该策略的功能可以对查询结果进行过滤改写的访问控制模型. 该策略功能可能包含其他的查询, 这些查询可能基于其他策略功能被改写过. 另外一个例子是Sybase的Adaptive Server Enterprise数据库同样采用基于查询重写的逻辑条件, 包括用户定义的任意Java函数的逻辑重写. 但这些仍缺乏正式的数学模型.

2.6 时域反射数据库访问控制(TRDBAC)

TRDBAC是RDBAC的扩展, 时域数据无处不在, 而几乎任意数据库都可以存储时域数据. 但是不同的数据能支持的查询类型并不相同. TRDBAC是用于管理时域数据的专业化数据库. 区别于传统的关系型数据库, 时域数据库针对时间数据的存储、查询和展现进行了专门的优化, 从而获得极高的数据压缩能力、极优的查询性能, 特别适用于物联网应用场景(物联网应用往往需要处理海量的时域数据).

TRDBAC的主要功能特点如下:

① 不要限制数据模型, 支持多个维度, 支持多个值, 维度要可以支持中文, 允许一个周期内存多个值;

② 能够按时间范围快速读取原数据;

③ 对于选择性高, 或者常用的维度, 希望能够彼此隔离, 也就是指定了维度去查的时候可以不用扫描所有的数据;

④ 服务器端高效地完成维度聚合;

⑤ 尽可能的利用时间维度和其他维度的重复性减少存储空间, 存储自身是压缩的, 占用越小越好.

根据TRDBC的特点, 我们以一个学生信息系统为例对TRDBAC进行研究.

3 案例分析: 实现TRDBAC的学生成绩信息系统

在这个案例中, 我们对学生成绩信息系统的访问控制策略进行实施. 该系统便于教师上传成绩, 考试协调员管理成绩, 并允许学生查看自己的成绩. 管理员必须监控和管理整个系统的访问. 让我们以学生成绩系统的详细概述为例.

实施访问控制策略的基本目的是执行成绩信息系统中的保密策略. 以下要点是要讨论的, 以便设计出访问控制策略, 防止未经授权的访问.

数据被存储在结果表中, 并且该系统的用户如教师, 学生等可根据用户类型授予的权限访问数据. 教师只能上传所授的课程的学生的成绩. 一旦教师将成绩上传, 成绩将无法更改. 如果需要任何类型的变更, 都需要将变更函发送至考试协调员.

最后的成绩显示策略也允许学生的家长/监护人查看最终成绩和评分. 实施这一策略的机制将确保家长/监护人能看到自己孩子的成绩.

考试协调员具有查看/更改所有学生最终成绩的能力. 当更改学生的成绩时, 需要记录更改的原因. 这些动作将被记录并审核, 防止未经授权的数据改变, 以保持数据的完整性.

管理员是唯一能够授权改变其他用户包括学生, 教师, 考试协调员等状态(有效/无效)的人员, 管理员还可以查看系统日志. 以下策略定义鉴于上述规则.

3.1 策略

3.1.1 一般情况

① 用户只能查看自己的个人信息;

② 用户只能更新自己的个人信息和联系方式;

③ 数据库中任何缺失的记录都不会被真正删除, 但会被标记为无效.

3.1.2 教师策略

① 教师只能上传他/她所录取的学生的成绩信息;

② 教师只能查看所授课程的学生成绩和联系方式;

③ 教师只能更新, 删除由教师标记为完成之前的数据;

④ 教师一旦将学生的成绩确定, 便不能再更改;

⑤ 教师还可以更改学生成绩信息的现状, 像“公布中”到“公布”, 作为学生参加课程的唯一途径;

⑥ 只有教师和管理员可以查看考试协调员的联系成绩.

3.1.3 学生策略

① 学生只能查看他/她自己的课程的成绩;

② 学生可以查看他/她就读课程的教师的联系信息;

③ 学生可以更新他/她自己的联系信息;

④ 学生只能联系相对应科目的教师查看成绩.

3.1.4 考试协调员策略

① 考试协调员只能查看当前所有被录取的学生的成绩;

② 考试协调员可以查看与课程有关的任何教师的联系信息;

③ 考试协调员可以在系统有任何问题的情况下联系管理员.

3.1.5 家长/监护人策略

① 家长/监护人只能在网站上公布查看到的自己孩子的成绩;

② 家长/监护人也可以查看他们的孩子就读课程的教师的联系信息.

3.1.6 管理员策略

① 管理员可以查看任何表中的所有数据;

② 管理员和教师只能查看考试协调员的联系信息;

③ 只有管理员可以查看在系统中执行各种操作所产生的更改日志;

④ 只有管理员才能在系统中添加新用户.

3.2 策略实施中RDBAC

3.2.1 基本状态策略

我们首先定义该系统的相关表, 并设置用户是有效, 能够访问这些信息.

① 讲师表

Teachers (Id, Name, Contact, .., .., .., IsActive)

Values (1, ABC, 234, .., .., .., 1)

② 学生表

Student (Id, FName, LName, .., .., IsActive)

Values (3, A, B, .., .., 1)

③ 监考员

Exam (Id, Name, .., .., IsActive)

Values (2, TH, ..,.., 1)

3.2.2 基本访问策略

具有该系统权限的用户可以访问该系统资源下的访问策略.

导师希望看到学生的信息:

hasAccess(IID, SID):

Teachers(IID, Name, , , 1) Student(SID, FName, LName, , , 1) Courses(CID, Code, Title, , , IID, 1) StudentCourses(SID, CID, , ,)

考试协调员希望看到学生信息:

hasAccess(ECID, SID):

Exam(ECID, Name, , , 1)

Student(SID, FName, LName, , , 1)

3.2.3 查看结果策略

学生只能查看他/她自己的结果; 同时教师可以查看所有已登记学生的成绩. 考试协调员可以查看所有学生参加任何学科的成绩. 家长/监护人还可以查看自己孩子的成绩. 所以, 下面是实现上述过程的策略:

① 学生查看结果

View.viewResults(FName, LName, CourseName, Marks, Grade, CGPA):

Student(SID, FName, LName, 1)

Courses(CID, Title, 1)

StudentCourses(SID, CID, 1)

Results(SID, CID, Marks, Grade)

② 教师查看结果

View.viewResults(FName, LName, CourseName, Marks, Grade, CGPA):

Teachers(IID, 1)

Student(SID, FName, LName, 1) Courses(CID, Title, IID, 1)

StudentCourses(SID, CID, 1) Results(SID, CID, Marks, Grade)

③ 考试协调员查看结果

View.viewResults(FName, LName, CourseName, Marks, Grade, CGPA): -

Exam(ECID, 1)

Student(SID, FName, LName, 1) Courses(CID, Title, 1) StudentCourses(SID, CID, 1) Results(SID, CID, Marks, Grade)

④ 家长/监护人查看结果

View.viewResults(FName, LName, CourseName, Marks, Grade, CGPA): -

Exam(ECID, 1)

Student(SID, FName, LName, PID, 1) Courses(CID, Title, 1) StudentCourses(SID, CID, 1) Parent(PID, 1)

Results(SID, CID, Marks, Grade)

⑤ 管理员查看结果

View.viewResults(FName, LName, CourseName, Marks, Grade, CGPA): -

Admin(AID, 1)

Student(SID, FName, LName, 1)

Courses(CID, Title, 1) StudentCourses(SID, CID, 1) Results(SID, CID, Marks, Grade)

3.2.4 插入结果策略

只有课程的教师被允许插入就读与他/她的学生的

结果. 但是, 一旦教师提交结果, 教师就不能插入或更新. 在这种情况下, 教师会要求到考试协调员添加/更新任何学生的结果. 以下是插入结果策略的执行:

通过导师插入结果

View.Ins.sResults(CID, SID, Marks, Grade, 1)

Results.IsInActive(SID, CID) || Result.NotExist (SID, CID) Teachers(IID, 1)

Student(SID, 1)

Courses(CID, IID, 1) StudentCourses(SID, CID, 1) Ins.sResults(SID, CID, Marks, Grade, 1)

3.2.5 更新结果策略

只有课程的教师被允许更新就读与他/她的课程, 另外考试协调员还可以更新结果提交后由于导师的要求需要更新的任何学生的结果.

通过导师查看更新结果

View.Upd.sResults(CID, SID, RID, Marks, Grade, 1)

Results.IsActive(SID, CID, RID, 1) || Result.Exists (SID, CID, RID) Teachers(IID, 1)

Student(SID, 1) Courses(CID, IID, 1) StudentCourses (SID, CID, 1) Results(RID, CID, SID, 1)

Upd.sResults(SID, CID, RID, Marks, Grade, 1)

3.2.6 删除结果策略

学生的课程成绩只有导师允许删除. 如果成绩已提交, 则只有考试协调员可以根据导师的要求删除学生成绩.

通过导师删除结果

View.Del.sResults(CID, SID, RID, Marks, Grade, 1)

Results.IsActive(SID, CID, RID, 1) || Result.Exists (SID, CID, RID, 1)

Teachers(IID, 1) Student(SID, 1)

Courses(CID, IID, 1)

StudentCourses(SID, CID, 1) Results(RID, CID, SID, 1) Del.sResults(SID, CID, RID, 1)

3.3 实现TRDBAC策略

下面的策略需要使用时间限制来实现:

如果由于某种原因必要要修改考试的结果, 则我们可以允许教师在一定的时间段更新结果. 例如, 教师被允许从2016年6月28日上午9时至2016年6月30日下午5点更新结果.

以下是这一策略的交易数据记录(TD)的说明 View.Upd_Temporal.sResults(CID, SID, RID, Marks, Grade, 1)

(Results.IsActive(SID, CID, RID, 1) || Results.Exists (SID, CID, RID, 1)) && !DateExpired

(IID, Current_DateTime)

Teachers(IID, Additional_Role, ‘6/28/2010 9:00:00 AM’, ‘6/30/2016 5:00:00 PM’, 1)

Student(SID, 1) Courses(CID, IID, 1) StudentCourses (SID, CID, 1)

Results(RID, CID, SID, 1)

Upd.sResults(SID, CID, RID, Marks, Grade, 1) SQL Implementatio

Create View Upd_Temporal_Results AS (UPDATE RESULTS r, Teachers i SET r.sTitle=’Title’, r.Marks= ’Marks’, r.Grade=’Grade’ WHERE CURRENT_ DATETIME

BETWEEN i.FromDateAND i.ToDate AND i.Additional_Role=1 AND i.IID = CURRENT_USER)

更新视图结果如表1所示.

表1 允许教师更新成绩视图表

教师ID	教师姓名	课程	年级	开始时间	结束时间
90003601	周意竹	高数	大一	2016-06-28 09:00:00	2016-06-30 17:00:00
90003602	陈俊	计算机	大一	2016-06-28 09:00:00	2016-06-30 17:00:00
90003603	吴音巧	英语	大一	2016-06-28 09:00:00	2016-06-30 17:00:00
90003604	陈国柏	体育	大一	2016-06-28 09:00:00	2016-06-30 17:00:00
90003605	王成文	数据库	大一	2016-06-28 09:00:00	2016-06-30 17:00:00
……	……	……	……	……	……

另一种情况, 我们需要实现TRDBAC让学生观察一些特定的时间段的结果. 比如, 学生被允许查看从2016年7月1日12点至2016年7月5日下午5点的成绩.

然后我们有以下策略描述:

View.view_Temporal.sResults(SID, RID, Marks, Grade, 1)

(Results.IsActive(SID, CID, RID, 1) || Results.Exists (SID, CID, RID, 1)) && !DateExpired (SID, CID, Current_DateTime)

Student(SID, Additional_Role, '7/1/2016 00:00:00 AM', '7/1/2016 5:00:00 PM')

Courses(CID, 1) StudentCourses(SID, CID, 1) Results(RID, CID, SID, 1)

View.sResults(SID, CID, RID, Marks, Grade, 1)

SQL Implementation

Create View View_Temporal_Results AS (SELECT r.sTitle, r.Marks, r.Grade FROM Results r INNER JOIN Courses c ON c.CID = r.CID INNER JOIN

StudentCoursessc ON sc.CID = c.SID INNER JOIN Students s ON s.SID = sc.SID WHERE CURRENT_DATETIME BETWEEN s.FromDate

AND s.ToDate AND s.Additional_Role=1 AND s.SID = CURRENT_USER)

在上述情况下, 需要TRDBAC对SQL的实现进行明确说明. 更新视图结果如表2所示.

表2 允许学生查看成绩视图表

学生ID	学生姓名	课程	年级	分数	开始时间	结束时间
1471087	贺易	高数	大一	86	2016-07-01 12:00:00	2016-07-05 17:00:00
1471088	张之娴	计算机	大一	100	2016-07-01 12:00:00	2016-07-05 17:00:00
1471089	刘侨枚	英语	大一	98	2016-07-01 12:00:00	2016-07-05 17:00:00
1471090	梁真	体育	大一	60	2016-07-01 12:00:00	2016-07-05 17:00:00
1471091	梁乐巧	数据库	大一	92	2016-07-01 12:00:00	2016-07-05 17:00:00
……	……	……	……	……	……	……

4 结论

RDBAC能够执行访问控制策略, 而不是实施应用级的数据库访问. 在某些情况下, 我们不仅在实施数据库级别的策略上感兴趣, 也对在一定时间限制下的策略感兴趣. 为了这个目的, 我们引入了TRDBAC, 这是在RDBAC的基础上进行了延伸, 实践证明这是一个很好的时间模式, 让我们的策略有了一定的时间限制.

参考文献

1 Tsai WT, Shao QH. Role-based access-control using reference ontology in clouds. Proc. 10th International Symposium on Autonomous Decentralized Systems. Tokyo, Hiroshima, Japan. 2011. 121–128.

2 王广宇. 云计算环境中的访问控制策略合成研究[硕士学位论文]. 西安: 西安电子科技大学, 2014.

3 贺正求, 张叶琳, 许俊奎, 等. Web服务访问控制策略研究. 计算机应用, 2015, 35(8): 2184–2188. [doi: 10.11772/j.issn.1001-9081.2015.08.2184]

4 Lazouski A, Martinelli F, Mori P. Usage control in computer security: A survey. Computer Science Review, 2010, 4(2): 81–99. [doi: 10.1016/j.cosrev.2010.02.002]

5 马学彬. 工作流中基于D-TRBAC的转授权问题的研究[硕士学位论文]. 大连: 大连理工大学, 2007.

6 沈晴霓, 杨雅辉, 禹熹, 等. 一种面向多租户云存储平台的访问控制策略. 小型微型计算机系统, 2011, 32(11): 2223–2229.

7 Bertino E, Bonatti PA, Ferrari E. TRBAC: A temporal role-based access control model. ACM Trans. Information and System Security, 2011, 4(3): 191–233.

8 陈颖, 杨寿保, 郭磊涛, 等. 网格环境下的一种动态跨域访问控制策略. 计算机研究与发展, 2006, 43(11): 1863–1869.

9 王旺. 基于Spring MVC框架和TRBAC访问控制模型的工作流系统的设计[硕士学位论文]. 合肥: 合肥工业大学, 2014.

10 Raje S, Davuluri C, Freitas M, *et al*. Using ontology-based methods for implementing role-based access control in cooperative systems. Proc. 27th Annual ACM Symposium on Applied Computing. Trento, Italy. 2012. 763–764.