基于 RBAC 模型多级角色的 SQLite3 安全访问控制①

白晋国, 胡泽明, 孙红胜

(信息工程大学 信息系统工程学院, 郑州 450001)

摘 要: 针对嵌入式数据库 SQLite3 没有完整的安全访问控制的问题, 本文基于 RBAC 模型, 结合 SQLite3 数据库内核源码, 设计了多级角色, 分析了其访问控制的实现方法与步骤, 详细设计并实现了 SQLite3 的安全访问控制. 实验测试结果表明该方法在继承 SQLite3 原有优点的同时, 提高了 SQLite3 的安全访问控制能力.

关键词: 嵌入式数据库;SQLite3;多级角色;RBAC;访问控制

Secure Access Control for SQLite3 Based on RBAC and Multi-Level Roles

BAI Jin-Guo, HU Ze-Ming, SUN Hong-Sheng

(School of Information Systems Engineering, PLA Information Engineering University, Zhengzhou 450001, China)

Abstract: SQLite3 is an embedded database but lack of complete secure access control mechanism. To solve this problem, this paper first designs a multi-level role function, then makes an analysis of the specific methods and steps of the access control mechanism for SQLite3 based on the RBAC model and combined with the source code of the SQLite3 database. Finally, it makes a detailed design and implementation of this access control mechanism for SQLite3. It is indicated by the experimental tests that the new secure access control mechanism not only succeeds the original advantages of SQLite3, but also improves the security of SQLite3.

Key words: embedded database; SQLite3; multi-level roles; RBAC; access control

随着社会信息化、网络化和智能化的发展, 计算机体积越来越小, 功能越来越强大, 单个芯片的处理能力越来越强. 为了满足社会应用的需求, 嵌入式系统(Embedded System)与嵌入式数据库[1](Embedded Database)应运而生, 并且在计算机领域中拥有广阔的发展空间.

SQLite[2]是D.Richard Hipp用C语言开发实现的一款开源的、嵌入式关系型数据库, 目前, 已经发展到了版本 3. 它具有零配置、开源、便携、易用、紧凑和可靠等特点, 支持绝大部分 SQL92 标准;因此, 在嵌入式数据库领域得到了广泛的应用, 可以编译运行在 Windows、Linux、BSD、Mac OS X 以及商用的 Unix 系统如 Solaris、HPUX 和 AIX, 还可以应用于很多嵌入式平台如 QNX、VxWorks、Symbian、Palm OS、Windows CE、Android、iOS 等.

然而, 开源的嵌入式数据库 SQLite3 并没有提供完整可靠的安全存储和访问控制功能, 任何拥有 SQLite3 数据库副本的用户都能通过 SQLite3 API 接口函数访问甚至修改 SQLite3 数据库记录, 这对有安全需求的用户和系统而言, 是一个致命的缺点.

本文结合 SQLite3 数据库内核源码, 在 RBAC 策略中引入多级角色的概念, 设计并实现了 SQLite3 的访问控制功能, 提高了嵌入式数据库 SQLite3 的安全访问控制能力.

1 多级角色的RBAC模型

访问控制[3](Access Control)是数据库安全最基本、最核心的技术, 是指通过某种途径显式地准许或限制能力及其范围, 以防止非法用户的侵入或合法用户的不慎操作所造成的破坏. 传统的访问控制类型主要有

① 收稿时间:2014-09-13;收到修改稿时间:2014-10-28

2 类:自主访问控制(Discretionary Access Control, DAC)和强制访问控制(Mandatory Access Control, MAC). 20 世纪 90 年代以来, 随着对在线的多用户、多系统的研究不断深入, 角色的概念逐渐形成, 并逐步产生了基于角色的访问控制(Role-Based Access Control, RBAC)模型[4].

在 RBAC 模型中, 角色是实现访问控制策略的基本语义体, 不仅仅是用户的集合, 也是一系列权限的集合. RBAC 模型的核心思想是将权限同角色关联起来, 而用户的授权则通过赋予相应的角色来完成, 用户所能访问的权限由该用户所拥有的所有角色的权限集合的并集决定. 当用户权限变更时, 可以很灵活地将该用户从一个角色转移到另一个角色来实现权限的转换, 降低了管理的复杂度. RBAC 模型的理论基础是 Sandhu 提出的核心模型[5,6], 包含 5 个基本集合:用户集(Users)、角色集(Roles)、对象集(Objects)、操作集(Operators)和会话集(Sessions).

本文结合 SQLite3 数据库内核源码, 重新分析了角色的权限类型和权限之间的相互关系, 将 RBAC 核心模型中的角色进行了进一步细致地划分, 引入了一级角色、二级角色、三级角色的概念, 建立了基于多级角色的 RBAC 模型, 如图 1 所示.

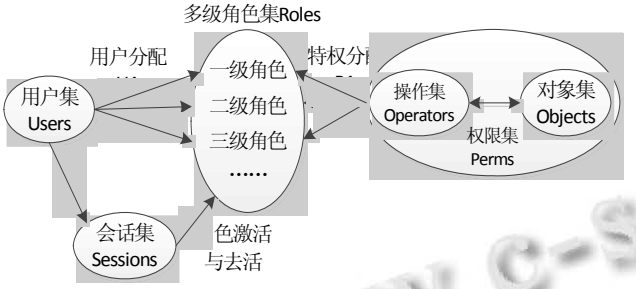

图 1　基于多级角色的 RBAC 模型

基于多级角色的 RBAC 访问控制策略的形式化描述如下:

①<Users,Roles,Perms,Sessions,Operators,Objects>其中, Users 是数据库中用户主体的集合;Roles 是数据库中多级角色的集合(包括一级角色、二级角色、三级角色等);Perms 是数据库中操作权限的集合, 每个操作权限都可以表示为一个二元组<Operators,Objects>, 其中, Operators 是具体的操作行为集合, Objects 是数据库中的访问对象集合;Sessions 是用户在数据库中的会话, 负责激活用户在当前会话活动中拥有的角色及其具有的权限.

②用户分配 $UA \subseteq Users \times Roles$:用户集到角色集的多对多映射, 表示用户被赋予角色的关系.

权限分配 $PA \subseteq Perms \times Roles$:权限集到角色集的多对多映射, 表示角色被赋予权限的关系.

③ $assigned\_users(r:Roles) = \{u \in Users \mid (u,r) \in UA\} \rightarrow 2^{Users}$:表示被直接指配到某一个具体角色的用户集合.

$assigned\_perms(r:Roles) = \{p \in Perms \mid (p,r) \in PA\} \rightarrow 2^{Perms}$:表示被直接指配到某一个具体角色的权限集合.

④ $allowed\_access(u,op,ob) \Rightarrow \exists u \in Users, \exists r \in Roles, \exists op \in Operators, \exists ob \in Objects \exists p \in Perms, (u,r) \in UA, <op,ob> \subseteq p, (p,r) \in PA$:表示当一个用户 u 所属的角色 r 具有权限 p(其中, 权限 p 为对客体对象 ob 的访问操作 op)时, 允许执行用户主体 u 请求的对客体对象 ob 的访问操作 op.

2　基于RBAC模型多级角色的SQLite3安全访问控制

基于多级角色 RBAC 的访问控制是以身份认证为前提的, 只有具有相应权限的合法用户才能对指定数据库中的数据记录进行相应的授权操作. 用户请求对数据库记录对象访问时, 系统根据用户的安全级别、角色访问权限以及存取访问的检查规则, 决定是否允许主体对客体请求的存取方式(插入、更新、删除、查询等)的访问.

SQLite3 的安全访问控制主要由两大模块组成:身份认证模块和访问控制模块;其中, 身份认证模块是访问控制模块的基础, 只有身份合法的用户才能进行相应的授权访问. 如图 2 所示, 当用户访问具有安全访问控制功能的 SQLite3 数据库时, 首先, 通过身份认证模块验证用户身份的合法性, 对于身份验证失败的用户, 系统返回 SQLITE_USER_INFO_ERROR 的错误提示信息并结束此次访问;其次, 通过访问控制模块对合法用户的具体数据库操作执行权限验证, 权限验证成功时执行相应的数据库操作, 权限验证失败时返回 SQLITE_DENY 的拒绝访问信息并结束此次访问.

2.1 身份认证模块

身份认证选用传统的用户名/口令的身份认证, 提供的是不依赖于硬件的跨平台的功能. 用户名/口令的

身份认证技术适合于嵌入在数据库系统内核以及相应的嵌入式设备中, 从而提高嵌入式数据库的安全性.

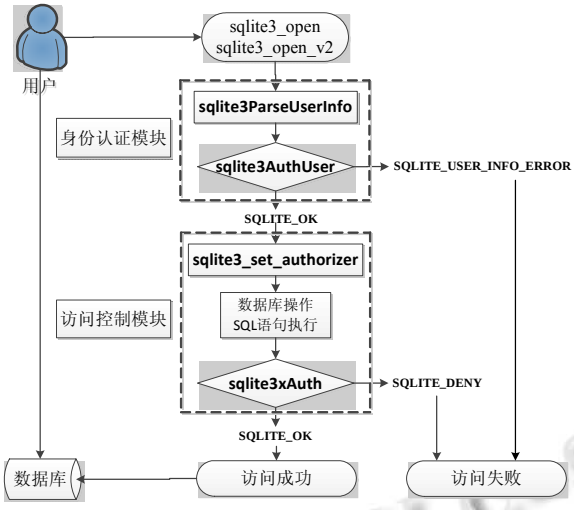

图 2 SQLite3 的访问控制设计图

SQLite3 身份认证模块主要包含两个阶段:解析参数 sqlite3ParseUserInfo 和身份验证 sqlite3AuthUser, 当身份验证成功即为授权用户时, 身份认证模块返回 SQLITE_OK 正确结果码;当身份验证失败即为未授权用户时, 身份认证模块返回 SQLITE_USER_INFO_ERROR 错误结果码. 因此, 可以由返回的结果码来判断当前用户是否为授权用户. 表 1 列出了身份认证模块中的核心 API 接口函数的相关信息及设计说明.

表 1 身份认证相关的 API 接口设计说明

API 接口	sqlite3ParseUserInfo	sqlite3AuthUser
功能	从用户提供的信息参数解析出用户名、口令以及数据库名	认证用户是否为合法用户
输入	*zUserInfo* 用户信息参数	*user_info* 用户信息结构体
输出	*UserName* 用户名; *UserPwd* 用户口令; *DbName* 数据库名	SQLITE_OK 或 SQLITE_USER_INFO_ERROR 身份认证结果码

2.2 访问控制模块

基于多级角色的 SQLite3 访问控制的基本思想是:分配给每个 SQLite3 用户一个合适的角色, 每一个角色都具有其对应的权限. 角色是安全控制策略的核心, 用户与角色关联, 角色与权限关联;用户拥有某个权限, 当且仅当这个用户所拥有的某个角色同该权限相关联. 用户、角色、权限之间的关联可以灵活转换, 一个用户可以拥有多个角色, 同样, 一个角色也可拥有多个权限;角色是用户与权限之间的纽带, 通过角色, 可以实现用户与权限的合理分配.

2.2.1 SQLite3 多级角色

在访问控制模块的策略中, 结合 SQLite3 数据库内核源码, 重新分析了角色的权限类型和权限之间的相互关系, 将 RBAC 核心模型中的角色进行了进一步细致地划分, 引入了一级角色、二级角色、三级角色的概念. SQLite3 数据库内核源码中定义了 32 种授权事件, 不同的数据库 SQL 操作需要不同的授权事件. 因此, 不同的角色具有不同的授权事件, 从而实现多级角色的权限划分.

一级角色具有数据库操作的所有权限, 可以执行所有的数据库操作, 是权限级别最高的角色, 包含所有二级角色和三级角色的权限, 如表 2 所示.

表 2 一级角色说明

一级角色	一级角色码	一级角色说明	包含的子角色
DbAdmin	100	数据库管员, 可以执行所有的数据库操作	所有的二级角色和三级角色

二级角色是数据库表、视图、触发器、索引等大子集的操作角色, 通常由三级角色组成, 如表 3 所示(表 3 中未列出视图、触发器、索引等不常用的二级角色说明).

表 3 常用的二级角色说明

二级角色	二级角色码	二级角色说明	包含的三级角色
TbOperater	101	表操作者	TbCreator、TbDroper
DtOperater	105	数据操作者	DtWriter、DtDeleter、DtUpdater、DtReader、
AllCreator	106	创建表、视图、触发器、索引	TbCreator、VwCreator、TgCreator、IxCreator
AllDroper	107	删除表、视图、触发器、索引	TbDroper、VwDroper、TgDroper、IxDroper

三级角色是数据库表、视图、触发器、索引、数据记录等的具体子操作(如创建、删除、查询等)角色, 如表 4 所示(表 4 中未列出视图、触发器、索引等不常用的三级角色说明).

表 4 常用的三级角色说明

三级角色	三级角色码	需要的授权事件	权限码	权限说明
TbCreater	1001	SQLITE_CREATE_TABLE	2	创建

		SQLITE_INSERT	18	表
		SQLITE_UPDATE	23	
		SQLITE_READ	20	
TbDroper	1002	SQLITE_DROP_TABLE	11	删除表
		SQLITE_DELETE	9	
		SQLITE_UPDATE	23	
		SQLITE_READ	20	
DtWriter	1009	SQLITE_INSERT	18	插入数据
DtDeleter	1010	SQLITE_DELETE	9	删除数据
		SQLITE_READ	20	
DtUpdater	1011	SQLITE_UPDATE	23	更新数据
		SQLITE_READ	20	
DtReader	1012	SQLITE_SELECT	21	查询数据
		SQLITE_READ	20	

三级角色之间具有相互包含关系, 例如, 拥有数据库表删除权限的角色同时拥有数据记录删除、更新的权限, 表 5 所示的就是三级角色之间的这种相互包含关系.

表 5 三级角色的相互包含关系

三级角色	三级角色码	包含的三级角色	包含的三级角色码
TbCreater	1001	DtWriter、DtUpdater	1009、1011
TbDroper	1002	DtDeleter、DtUpdater	1010、1011

2.2.2 SQLite3 权限验证

基于多级角色的 SQLite3 访问控制通过授权钩子函数 sqlite3_set_authorizer 安装授权回调函数 sqlite3xAuth 来实现. SQLite3 在编译 SQL 语句时, 为各种数据库事件调用授权回调函数. 授权回调函数 sqlite3xAuth 调用权限验证函数 sqlite3OpAuth, 目的就是使应用程序可以安全地执行用户提供的 SQL 语句, 根据用户的权限允许或拒绝某种 SQL 操作以及对数据库中特定表或字段的访问.

表 6 列出了访问控制模块中的核心 API 接口函数的相关信息及设计说明.

表 6 访问控制相关的 API 接口设计说明

API 接口	功能	输入	输出
sqlite3xAuth	授权回调函数	用户信息结构体 *user_info*, SQL 事件码,	权限验证结果码 SQLITE_OK 或 SQLITE_DENY
sqlite3OpAuthCode	用户所属角色返回相对应的角色码	用户角色 *UserRole*	用户角色码 *roleId*
sqlite3OpAuth	验证用户所属角色是否具有操作权限	用户信息结构体 *user_info*, 操作权限码 *op_type*	权限验证结果码 SQLITE_OK 或 SQLITE_DENY
sqlite3OpAuth_1	一级角色权限验证	用户角色码 *roleId* 操作权限 *op_type*	权限验证结果码 SQLITE_OK 或 SQLITE_DENY
sqlite3OpAuth_2	二级角色权限验证	用户角色码 *roleId* 操作权限 *op_type*	权限验证结果码 SQLITE_OK 或 SQLITE_DENY
sqlite3OpAuth_3	三级角色权限验证	用户角色码 *roleId* 操作权限 *op_type*	权限验证结果码 SQLITE_OK 或 SQLITE_DENY

其中, 核心权限验证函数 sqlite3OpAuth 的实现步骤如下:

步骤 1: 检查用户信息结构体 *user_info* 的角色成员 *UserRole* 是否为空, 如果为空则终止, 否则由用户所属角色返回相对应的角色码:

roleId = *sqlite3OpAuthCode*(*user_info->UserRole*) ;

步骤 2: 检查用户角色是否属于一级角色, 如果属于一级角色, 验证此角色是否具有操作权限 *op_type*: sqlite3OpAuth_1(*roleId* , *op_type*) ;否则, 执行步骤 3;

步骤 3: 检查用户角色是否属于二级角色, 如果属于二级角色, 验证此角色是否具有操作权限 *op_type*: sqlite3OpAuth_2(*roleId* , *op_type*) ;否则, 执行步骤 4;

步骤4: 检查用户角色是否属于三级角色, 如果属于三级角色, 验证此角色是否具有操作权限 *op_type*: sqlite3OpAuth_3(*roleId* , *op_type*) ;否则, 执行步骤 5;

步骤5: 根据权限验证结果, 返回相应的权限验证结果码 SQLITE_OK(允许访问)或 SQLITE_DENY(拒绝访问), 权限验证结束.

2.3 基于 RBAC 模型多级角色的 SQLite3 访问过程

基于 RBAC 模型多级角色的 SQLite3 访问过程如下:

步骤 1: 调用 SQLite3 的 C API 接口 sqlite3_open 或 sqlite3_open_v2 打开或创建数据库 :sqlite3_open("*user_name&user_pwd@db_name*",&db);

步骤 2: 由用户提供的信息参数解析出用户名、口令以及数据库名 :sqlite3ParseUserInfo (*zUserInfo*,

UserName, *UserPwd*, *DbName*);

步骤 3: 认证用户是否为合法用户 sqlite3AuthUser (*user_info*), 若为非法用户, 则结束访问, 返回 SQLITE_USER_INFO_ERROR 错误标志码;

步骤 4: 执行编译数据库操作与相应的 SQL 语句;

步骤 5: 查看合法用户是否具有执行此数据库操作和 SQL 语句的权限:sqlite3xAuth(*user_info, op_type, tb_name*, *col_name*, *db_name, view_name*), 若此合法用户没有执行此数据库操作和 SQL 语句的权限, 则结束访问, 返回 SQLITE_DENY 拒绝访问标志码;

步骤 6: 调用 SQLite3 的 C API 接口 sqlite3_close (db)关闭数据库.

3 实验测试

基于多级角色的 SQLite3 安全访问控制实验分为两大功能测试:身份认证功能测试与权限控制功能测试. 实验测试选用版本号为 3.8.0.2 的 SQLite3 数据库. 实验测试前, 首先要初始化系统数据库 SystemDB 和存储用户信息的数据表 UserInfoTB. 存储用户信息的数据表 UserInfoTB 的结构为用户名 UserName, 类型为 TEXT, 用户密码 UserPwd, 类型为 TEXT, 用户角色 UserRole, 类型为 TEXT. 系统数据库只有数据库管理员可以访问, 其他任何数据库角色都不具有访问权限.

实验前, UserInfoTB 中存储一个测试用户的信息, 用户名 UserName 为“XiaoHui”, 用户密码 UserPwd 为“1234abcd”, 用户角色 UserRole 为“DtReader”.

(1)身份认证功能测试

实验分 4 种情况进行功能测试:

① 未提供用户名或用户口令;

② 提供未授权的用户名和正确的口令;

③ 提供授权的用户名和错误的口令 ;

④ 提供授权的用户名和正确的口令.

图 3 SQLite3 身份认证测试结果

实验测试结果如图 3 所示:由实验测试结果可知, 基于用户名/口令的身份认证模块能够有效地阻止未授权用户和错误用户的访问, 返回相应的“user name or password is error”错误信息提示.

(2)权限控制功能测试

实验测试基于身份认证模块中的身份认证, 只有身份合法的用户才能进行下一步的授权访问. 实验中授予合法用户“XiaoHui”查询数据库记录的角色“DtReader”. 实验测试结果如图 4 所示:由实验测试结果可知, 基于多级角色的 RBAC 策略可以有效地控制合法用户的授权访问, 防止非法用户的非法访问和合法用户的越权访问, 返回未被授权的“not authorized”提示信息, 保证了 SQLite3 数据库数据的安全性.

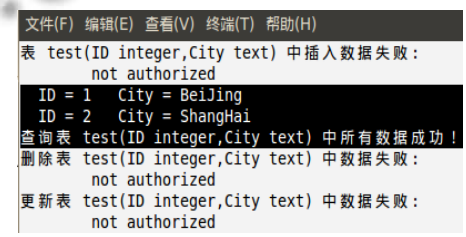

图 4 SQLite3 权限控制测试结果

SQLite3 作为一款嵌入式数据库, 运用于各种嵌入式设备和嵌入式环境中, 其资源的占用大小将直接影响嵌入式系统或设备的性能. 通过编译 SQLite3 源码, 可以发现:未添加访问控制功能的动态库“sqlite3.dll”大小为 519KB;添加访问控制功能后的动态库“sqlite3.dll”大小为 524KB, 资源消耗仅增加了 5KB. 由此可见, 本文的安全访问控制方法以增加较少的资源消耗为代价加强了 SQLite3 的安全访问控制功能, 为嵌入式数据库 SQLite3 的使用提供了安全保障.

4 结语

SQLite3 是一款开源的、嵌入式关系型数据库, 并没有提供完整的安全访问控制功能. 本文结合 SQLite3 数据库内核源码, 重新分析了角色的权限类型和权限之间的相互关系, 在 RBAC 策略中, 引入多级角色的概念, 建立了基于多级角色的 RBAC 模型, 详细设计并实现了 SQLite3 的安全访问控制功能. 最后通过实验测试验证了该方法的可行性和有效性, 测试结果表明基于 RBAC 模型多级角色的 SQLite3 访问控制在继承 SQLite3 原有优点的同时, 安全性也得到了提高.

参考文献

1 史恒亮,白光一.嵌入式数据库的现状和发展趋势.计算机系统应用,2010,19(2):205–208.

2 格兰特·艾伦,迈克·欧文斯.SQLite 权威指南.第 2 版.杨谦,刘义宣,谢志强译.北京:电子工业出版社,2012.

3 赵宝献,秦小麟.数据库访问控制研究综述.计算机科学,2005,32(1):88–91.

4 Sandhu S, Coynek EJ, Feinsteink HL, et al. Role based access control models. IEEE Computer, 1996, 29(2): 38–47.

5 Ravi S. Rationale for the RBAC96 family of access control models. Proc. of the 1st ACM Workshop on Role Based Access Control. ACM. 1997.

6 Moyer MJ, Abamad M. Generalized role-based access control. 21st International Conference on Distribued Computing. 16–19 April 2001. 391–398.