维普资讯 http://www.cqvip.com

基于Oracle 10g的数据安全性技术研讨*)

徐 玲[1] 文俊浩[1] 李立新[1,2]

(重庆大学软件学院 重庆400030)[1] (信息工程大学电子技术学院 郑州450004)[2]

摘 要 Oracle数据库系统是目前应用最广泛的大型关系数据库系统之一。本文从数据库安全角度出发,从不同侧面和角度分析和讨论了Oracle数据库10g中数据加密算法的实现,基于角色的权限管理,细粒度的审计检测(FGA)以及强大的RMAN备份和恢复工具等方面分析了Oracle数据库系统的安全保护机制。

关键词 Oracle 10g,数据库安全,安全机制

Research in Database Security Based on Oracle10g

XU Ling[1] WEN Jun-Hao[1] LI Li-Xin[1,2]

(College of Software Engineering , Chongqing University,Chongqing 400030)[1]

(Electronic College of Information Engineering University,Zhengzhou,450004)[2]

Abstract Oracle Database is one of the most widely applied large-scaLe RDBMS. The paper, based on Oracle database10g security, discusses the security protected Oracle mechanism from different point of view. The techniques of data encrypition, Oracle authority limits management based roles, FGA auditing, RMAN backup and resume are presentd.

Keywords Oracle 10g, Database security, Security mechanism

1 引言

在当今开放式的互联网时代,许多关键的业务系统运行在数据库平台上,数据库系统中的数据为众多用户所共享,如果数据库安全无法保证,其上的应用系统也会被非法访问或破坏,所以数据库系统的安全保护措施是否有效已成为现代数据库系统的重要性能指标之一。

Oracle作为大型的数据库系统,以其优异的性能在各个领域得到广泛应用。但由于计算机软、硬件故障、口令泄密、黑客攻击等等因素,都可导致数据库系统不能正常运转,造成大量数据信息丢失,数据被恶意篡改,甚至使数据库系统崩溃。忽视Oracle数据库的安全问题,必将给用户带来巨大的损失。

本文基于Oracle 10g,对数据系统的安全技术进行探讨。

2 数据库系统的安全需求分析

信息在网络(如Internet)上传输,非法用户就可通过网络监听等非法手段获得信息;信息在系统内部的网络中传输时恶意用户通过网络监听等非法手段同样可能获取非授权的数据。针对数据库应用系统所面临的安全风险,数据库应用系统的安全需求主要包括:

(1)*严格的身份认证* 在一个数据库应用系统中,每一个合法用户的身份标示是不同的,其操作权限也不同。非法用户为获取信息,常常伪装成合法用户,数据库系统需要对用户进行身份认证。

(2)*数据在网络上以加密形式传输* 用户向数据库服务器发送的查询请求和数据库服务器返回的查询结果都应该以高强度的加密算法加密以防被窃取。

(3)*数据能以密文形式存放在数据库中* 对一个数据库系统来讲,只要数据以可读的形式存储在数据库中,就可能存在潜在的危险,为此必须采用数据库加密的方法。加密后的数据库称为密文数据库。

(4)*用户授权机制* 数据库应用系统应该实现授权用户的授权管理,实现授权用户对授权对象的访问权限的分配、回收、定义和控制。

(5)*数据的备份和恢复* 在数据库系统中,意外情况总是不可避免的,为了保证故障发生后,数据库中的数据能从错误状态恢复到某种逻辑一致的状态,数据库管理系统中就必须有备份/恢复子系统。

以上简要分析了数据库应用系统的安全需求,在数据库应用系统的实际开发中,须根据实际需求选择相应的技术加以满足。

3 Oracle数据库安全保护机制

安全性宗旨是保证数据的机密性、完整性和可用性。Oracle数据库系统有着十分严格而灵活的数据库安全保护机制,可以防止未受权的数据库访问,防止对具体对象的未授权访问、控制磁盘及系统资源(如cpu时间)的分配和使用、稽核用户行为等,能够从多方面避免数据的泄漏。

3.1 基于角色的权限管理

管理用户对资源存取的传统方法通常是直接为每个用户赋予一组权限,这里用户、角色与权限之间是一种多对多的关

*)基金项目:"十五"国家科技计划项目(2002BA107B);重庆市自然科学基金支持项目。徐 玲 硕士研究生,主要研究方向:数据挖掘、数据库安全、网络数据库;文俊浩 副教授,博士,主要研究方向为数据挖掘、软件工程、数据库安全;李立新 副教授,博士,主要研究方向:数据库安全,数据挖掘。

维普资讯 http://www.cqvip.com

系。这种机制的特点是权限管理任务繁重、且不灵活，安全管理层次低、易错。基于角色的访问控制是 Oracle 等大型数据库提供的一种新型的数据库权限管理机制。它的基本思想就是根据组织视图中不同的职能岗位划分角色，资源访问权限被封装在角色中，用户被指派到角色，通过角色间接访问资源。在基于角色的权限管理中把角色当作主体，图1展示了基于角色访问控制的一个基本模型。类 User 和 Role 分别表示已注册的用户和预定义的角色，用户被赋予角色，按照角色的职能，赋予用户特定权限。关联类 Right 定义一个角色用户被授权访问保护对象的权限类型。实际上，综合 Role、ProtectionObject 和 Right，就是一个授权模式的实例。同样，谓词表示依内容而定的限定，用来选择特定对象，copy_flag 属性表示一个布尔值，true 或 false。

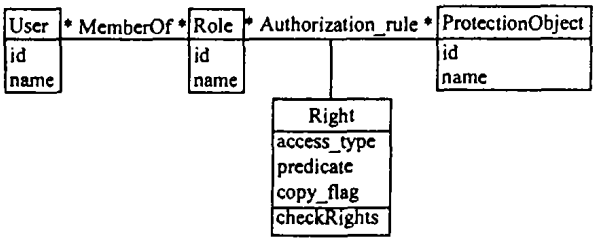

图1　基于角色的权限管理模式

这种将若干特定的用户集合和某种授权连接在一起的授权管理与个体授权相比较，具有强大的可操作性和可管理性，因为角色的变动远远少于个体的变动。基于角色的安全机制为管理大量的访问权限提供了一种灵活的、动态的方法，可以对角色灵活地配置和再配置。同时，它作为一种存取控制策略，实现了最基本的安全原则：最小特权原则和职责分离原则，这为开发安全策略的粒状方法做出了充分准备。Oracle 数据库中对预定义角色提供内镶权限，如 CONNECT、RESOURCE、DBA 权限等。

Oracle 的角色控制具有很强的实时性，授予角色的权限立即被分配给角色的用户使用；从一个角色取消某项权限，那么就会立即阻止与该角色有关联的所有用户使用该权限；角色还可根据需要处于激活或停顿状态，以便控制用户所授予的角色的活动。另外角色本身可以得到口令保护。基于角色的权限管理极大地加强了系统的安全控制的灵活性，进一步增强了系统的安全性能。

3.2　数据加密

数据加密就是将明文数据经过一定的交换（一般为变序和代替）变成密文数据。数据脱密是加密的逆过程，即将密文数据转变成可见的明文数据。

一个密码系统包含明文集合、密文集合、密钥集合和算法，其中密钥和算法构成了密码系统的基本单元。算法是一些公式、法则或程序，规定明文与密文之间的变换方法，密钥可以看作算法中的参数。数据库密码系统要求将明文数据加密成密文数据，数据库中存储密文数据，查询时将密文数据取出脱密得到明文信息。

较之传统的数据加密技术，数据库密码系统有其自身的要求和特点。传统的加密以报文为单位，加密脱密都是从头至尾顺序进行。而数据库中的数据必须以字段为单位进行加密，否则该数据库将无法被操作。同时由于数据库中的数据是共享的，有权限的用户随时需要知道密钥来查询，修改，删除和插入数据，这样就要随时对数据库中数据进行加解密处理。因此，数据库加密脱密密钥应该是相同的，采用对称密码体制是比较适宜的。

Oracle 支持对诸如信用卡号码的敏感数据和一些高度敏感的业务数据进行加密，例如，可以使用 TRANSLATE 函数将包含在字符串中的正文进行转换，返回一个由另一个字符串中相关字符替换的字符串，它可以在 PL/SQL 中使用以加密和解密数据。加密时，可通过从系统时间的秒数产生一个半随机数作为偏移量，这样每次运行该函数时都产生一个不同的加密值，这样使得解码十分困难，可以有效防止别人获取加密数据信息。

DES（数据加密标准）是国际上承认的安全数据加密标准，三重 DES（tripe DES）是对最初的 EDS 算法进行的最新精练。DBMS_OBFUSCATION_TOOLKIT 包含了执行 DES 加密/解密的 PL/SQL 过程和函数，以及用于三重 DES 加密/解密的过程和函数。

下面举例来构建加密函数 MY_ENCRYPT，其中 INPUT STRING 用来进行加密或者解密的输入字符串。输入的字符串是8个字节的倍数。KEY STRING 用来加密期间作为密匙的字符串。在执行 DES3ENCRYPT 函数之前，自动地将 INPUT STRING 值和 KEY STRING 值填充到所需的长度。如下所示：

```
SQL)create or replace function my_encrypt (
2        p_input_string in varchar2,
3        p_key_string in varchar2)
4 return varchar2
5 as
6   l_encrypted_value varchar2 (4000);
7   l_input_string varchar2 (4000);
8   l_key_string varchar2 (4000);
9 begin
10 --
11 --Pad the input string to multiple of 8 bytes(required)
12 --
13   l_input_string:= rpad(p_input_sting,
14     (trunc (length (p_input_string)/8)+1) * 8,chr(0));
15 --
16 --Pad the key string to 16 bytes (required)
17 --
18   l_key_string:= rpad(p_key_string,16,chr(0));
19   l_encrypted_value:=
     dbm_obfuscation_toolkit.des3encrypt (
20     input_string =>l_input_string,
21   key_string =>l_key_string);
22   return l_encrpted_value;
23   end;
24   /
```

通过采用相同的方式，我们能够建立函数 MY_DECRYPT 来执行解密，将输入字符串和密匙值填充到合适的长度。

通过使用这两个函数，我们现在就能够很容易地（而且安全地）加密和解密任意的数据值，下面是一个使用 My_ENCRYPT 加密函数的示例：

```
SQL > Select rawtohex ( my_encrypt ('MyData',
'thekey'))from dual;
```

尽管在数据库中直接执行 DES3加密很方便，但由于这个算法中有个相关联的密钥，从理论上讲，任何能够访问密匙的用户都有能力解密数据库。所以，通过限制能够访问用来加密用户数据的密匙的方法和用户。

考虑到 Oracle 加密的局限性和安全性，可以考虑在现有的 Oracle 数据库系统做一个外层 DBMS 工具。采用这种加密方式时，加脱密运算都可以放在客户端/进行，这样做不会加重数据库服务器的负载并可实现网上传输加密。图2展示了数

维普资讯 http://www.cqvip.com

据库加密系统的体系结构。

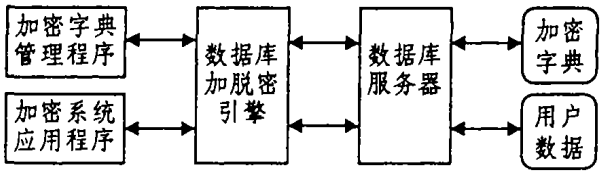

图2 数据库加密系统体系结构

在这种体系结构中，数据库加密系统将用户对数据库信息具体的加密要求记载在加密字典中，加密字典是数据库加密系统的基础信息。加密字典管理程序，是管理加密字典的实用程序，是数据库管理员变更加密要求的工具。加密字典管理程序通过数据库加脱密引擎实现对数据库表的加密、脱密及数据转换等功能，此时，它作为一个特殊客户来使用数据库加/脱密引擎。数据库加/脱密引擎是数据库加密系统的核心部件，负责在后台完成数据库信息的加/脱密处理，对应用开发人员/和操作人员是透明的。

3.3 细粒度审计监控

传统的 Oracle 数据库审计选件允许在宏观级别上跟踪用户在对象上所执行的操作。例如，如果您审计对某个表的 SELECT 语句，则可以跟踪是谁从表中选择了数据。但是，不知道他们选择了什么。利用数据操纵语句、触发器或使用 Oracle LogMiner 实用程序来分析归档日志，从而捕获任何的更改。因为简单的 SELECT 语句是不操纵数据的，它们既不启动触发器，也不记入到那些以后可以进行挖掘的归档日志中，所以这两种技术在涉及到 SELECT 语句的地方无法满足要求。

细粒度审计(FGA)，可以用于监视基于内容的数据访问，它能够记录 SCN 号和行级的更改以重建旧的数据，该特性允许您将单个的 SELECT 语句联同用户提交的确切语句一起进行审计。除了简单地跟踪语句之外，FGA 还通过在每次用户选择特定的数据集时执行一段代码，提供了一种方法来模拟用于 SELECT 语句的触发器。

Oracle 10g 的版本增强了 FGA 的功能，除了支持 SELECT 语句外，还可支持插入、更新和删除操作，这些更改都记录在同一个位置，即表 FGA_LOG $ 中，并通过 DBA_FGA_AUDIT_TRAIL 视图显示出来。如果采用触发器的方式，最大的问题是触发器是对每行触发的，而不是每条语句一次。触发器捕获旧的值和新的值，并填充 AUDIT_LOG 表。如果需要，还可以将它变为自动事务。最大的问题是触发器是对每行触发的，而不是每条语句一次。例如，以下语句：

Update accounts set balance = 1200 where balance >= 3000;

对全部 10,000 条记录触发，在审计表中插入 10,000 行。这种方法可能严重地损害 update 语句的性能，甚至可能因审计线索中的空间问题而导致失败。使用语句触发器也无济于事，因为它不能捕获个别记录的任何新的或旧的值。比较而言，在 FGA 方法中，只创建一条记录，并且插入只在每条语句上执行一次，而不是每行一次，即使有的话，对性能的影响也很小。在 FGA 中，无论何时被(无论它们被修改与否)，审计就开始。这个特性使得 FGA 比触发器具有更多的功能。

另一个优点是 FGA 工具的适用性。有时，在一个视图上定义的 INSTEAD OF 触发器在基表上更新视图；另一个 INSTEAD OF 触发器不能捕获由其它的触发器所作的修改，因此这些修改不能被记录。然而，FGA 是建立在视图或表的基础上的，它能够捕获变化，而不论变化来自哪里——用户语句或触发器。

FGA 能以一个非常详细的级别来捕获用户的行为。它还结合了标准审计和 FGA 的跟踪，这使其更易于跟踪数据库访问，而不用考虑它是如何生成的。Oracle 10 g 把这些跟踪合并到一个称为 DBA_COMMON_AUDIT_TRAIL 的通用跟踪中，它是 DBA_AUDIT_TRAIL 视图和 DBA_FGA_AUDIT_TRAIL 视图的一个 UNION ALL 视图。

3.4 RMAN 的备份和恢复

Oracle 目前的备份与恢复基础架构是业界最强大、最可靠的。Oracle 恢复管理器 (RMAN)是一个强大的工具，能够简化、自动化及改善备份与恢复操作。常见的备份方式有物理备份和逻辑备份，这两种备份方式，一般比较熟悉，这里不再详述，主要讨论一下 Oracle 提供的 RMAN 的增量备份方式。

Recovery Manager(RMAN)是一种用于备份(backup)、还原(restore)和恢复(recover)数据库的 Oracle 工具。它能够备份整个数据库或数据库部件，如表空间、数据文件、控制文件、归档文件以及 Spfile 参数文件。

RMAN 提供的最重要的新特性是能够执行数据文件的增量物理备份，增量 RMAN 备份是时间和空间有效的，因为它们只备份自上次备份以来有变化的那些数据块。在一个全(叫做0级)数据文件备份中，数据文件中用过的所有数据块都要进行备份，在一个累积(叫做1级)数据文件备份中，上次数据文件全备份后用过的所有的数据块都要进行备份，而在增量(2级)数据文件备份中，仅备份那些在上一次累积备份或全备份后发生变化的数据块。图3显示了差异增量备份(默认的增量备份方式)和累积增量备份的示意图。

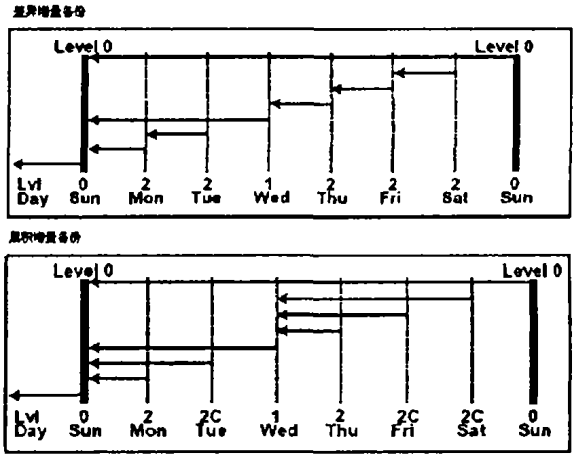

图3 差异增量备份与累积增量备份

可以看到，差异增量是备份上级或同级备份以来变化的块，累计增量是备份上级备份以来变化的块。累计增量增加了备份的时间，但是因为恢复的时候，需要从更少的备份集中恢复数据，所以，为了减少恢复的时候，累计增量备份将比差异增量备份更有效。

Oracle 版本10在增量备份上做了很大的改进，可以使增量备份变成真正意义的增量，因为通过特有的增量日志，使得 RMAN 没有必要去比较数据库的每一个数据块，当然，代价就是日志的 I/O 与磁盘空间付出，完全还是不适合 OLTP 系统。另外，版本10 通过备份的合并，使增量备份的结果可以合并在一起，而完全地减少了恢复时间。

可以看出，RMAN 具有如下优于传统的基于操作系统的物理备份方法的优点：

(1)热备份不会导致与传统的 Begin/End 备份方法有关的重做日志产生率；

维普资讯 http://www.cqvip.com

(2)作为备份过程的一个集成部分,检查数据库块是否残缺不全,这可以减少对整个数据库频繁使用 Analyze Table Validate Structure 语句的要求;

(3)支持增量物理备份,这个与导出/导入的增量截然不同;

(4)支持多线程备份;

(5)提供一个集成的分类系统,该分类系统在鉴别所需的备份磁带时能够把混乱减到最小。

结束语 Oracle 本身提供了强大的安全机制,但同时也存在着一些漏洞,普遍存在于 Oracle 数据库中的用户问题:

(1)权限过大:对 Oracle 数据库编程和浏览的一般用户常常具有 DBA (数据库管理员权限),能对数据库系统做任何修改或删除。

(2)安全性差:很多 Oracle 用户缺省存储位置都在系统表空间,这样不仅影响系统的正常工作,而且不同用户的数据信息互相影响、透明,保密性差。随着数据的不断加入,有可能使整个数据库系统崩溃。

(3)密码有规律:在 Oracle 调试初期形成的用户名和密码一致的不良习惯保留到现在;系统用户 SYS 和 SYSTEM 的密码也众所皆知。

因此真正保障数据库应用系统的安全性,仅仅保证 DBMS 的数据库安全保障体系还不能彻底解决系统的安全问题,保障数据库应用系统的安全性还需要从安全策略上加以保障,需要从体系结构上保障,需要增强物理安全、网络安全以及系统安全性,DBA 需要经常检查安全策略与措施,堵塞漏洞,防范于未然。

参 考 文 献

1 Oracle9i Security Overview [M]. Oracle Corporation, 2002. 60～90
2 Oracle9i 基础与提高[M]. 飞思科技产品研发中心编著. 北京:电子工业出版社,2003. 370～450
3 Dillon S. Oracle 编程入门经典[M]. 清华大学出版社,2002. 458～498
4 Introduction to Oracle Identity Management [M]. An Oracle White Paper, Dec. 2003. 4～6
5 Gorman T. Unraveling the Sweater Oracle Database Security [M]. 2002. 4～8
6 蒋继洪. 计算机系统、数据库系统和通信网络的安全与保密[M]. 成都:电子科技大学出版社,1995
7 Nanda A. 现实问题的细粒度审计. http://www.oracle.com/technology/global/cn/pub/articles/nanda_fga_pt2.html 2004-1-9
8 陈吉平. Oracle 9i RMAN 参考使用手册[J]. 2002. 10～12
9 朱鲁华,陈荣良. 数据库加密系统的设计与实现. 计算机工程,2002(8)

(上接第71页)

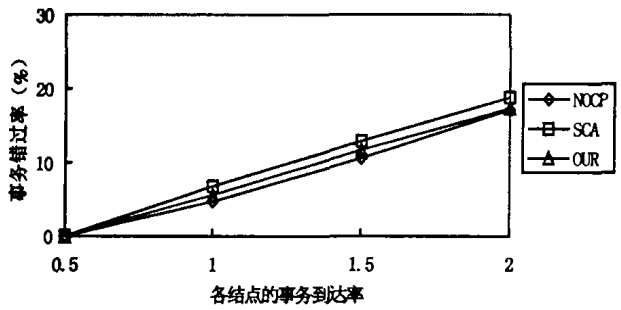

图2 轻载荷下事务错过率比较

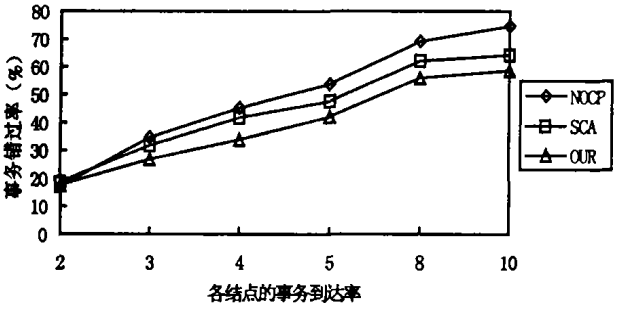

图3 重载荷下事务错过率比较

轻、重载荷下不同事务到达率时的 MR 如图2、3所示,从图中可以看出在系统处于轻载荷时,采用检验点技术的事务错过率比未采用时略高,并且采用 SCA 比采用本文方法的事务错过率要高。这是由于轻载荷下系统中事务数较少,发生数据竞争的可能性比较小,事务故障是导致事务重启的主要原因,这使得总事务重启率相对较小。采用检验点技术多花费了记检验点的时间,因此事务错过率相对未采用时略高。SCA 由于检验点延迟比本文方法法要大,因此有更高的事务错过率。注意在轻载荷下采用本文方法对系统性能并没有太大的影响。

当系统处于重载荷时,采用检验点技术的事务错过率比未采用时要低很多,检验点技术的优势逐渐显示出来。这是因为在重载荷下系统中事务数量不断增多,数据竞争的概率增加造成总事务重启率变大(注意合作事务集中任一事务重启都会导致事务集中其它相关事务的重启),采用检验点技术则不用完全重启事务而减少了计算量的损失,因此有利于事务定时约束的满足,提高了系统性能。本文方法的时间代价要优于 SCA,因此有更低的事务重启率。

结论 随着实时数据库应用的深入和计算机网络应用的普及,分布式实时数据库成为新的研究热点。在一类分布式合作实时数据库应用中,不同结点的事务通过消息通信结成合作关系。为了减少因事务故障和数据冲突造成的事务重启而损失的工作量,可以采用检验点来定期记录事务执行的快照,本文的主要工作有包括提出了一种新的分布式合作实时事务经历模型;给出了事务间由于消息通信而形成的检验点依赖关系的形式化定义;提出并实现了一种高效的并行最小协同检验点方法,该方法能降低一个全局检验点的时间延迟,从而更有利于实时事务定时限制的满足;设计并实现了一个实验系统,对文中方法进行了性能评价。

参 考 文 献

1 Grey J, Reuter A. Transaction Processing: Concepts and Techniques. San Mateo, Calif. Morgan Kaufmann, 1993
2 Prakash R, Singhal M. Low-cost checkpointing and failure recovery in mobile computing systems. IEEE Tran. Parallel and Distributed Systems, Oct. 1996. 1035～1048
3 Koo R, Toueg S. Checkpointing and roll-back recovery for distributed systems. IEEE Tran. on Software Engineering, Jan. 1987. 23～31
4 Cao Guohong, Singhal M. On coordinated checkpointing in distributed systems. IEEE Tran. on Parallel and Distributed Systems, Dec. 1998. 1213～1225
5 Randell B. System structure for software tolerance. IEEE Transactions on Software Engineering, 1975, SE-1(2): 220～232
6 Deng Y, Park E K. Checkpointing and Rollback-Recovery Algorithms in Distributed Systems. Journal of Systems and Software, Apr. 1994. 59～71
7 汪东升,邵明珑. 具有 O(n)消息复杂度的协调检查点设置算法. 软件学报,2003,14(1):43～48
8 刘云生著. 现代数据库技术. 北京:国防工业出版社,2001