

第23章 安全管理

本章要点：

用户验证

数据库权限管理

Oracle企业安全管理器

监控数据库资产

保护数据完整性

Oracle 8i因特网安全特性

防火墙支持

好的粒状存取控制

数据库资源管理器

硬件安全

恢复丢失的数据

Oracle 8i有一套给人留下深刻印象的全新的安全特性。安全的宗旨是机密性、完整性和可用性。Oracle已经增强了所有这些领域中的安全性。作为 Oracle安全模型基础的安全原理是基于最小特权原则。此原则认为用户应该仅有完成他的任务所必需的特权。这为一个开发安全策略的粒状方法做准备。这些特性可以被分成两大类：数据库安全与用于因特网处理的数据库安全。一个用于因特网的 Oracle数据库包括所有典型的数据库安全外加支持因特网的标准安全选项。因特网数据库需要特别的安全保障。因特网数据库与标准数据库的安全问题将在本章中讨论并详细阐述。

23.1 用户验证

用户在访问你的数据库以前必需被识别与验证。用户将被 Oracle识别，但是他们可以用3种不同的方法得到验证：数据库验证、外部验证或企业验证。另外，数据库和 Web服务器也可以被验证。这些内容将在稍后的 23.6节中讨论。

23.1.1 数据库验证

当创建用户和指定口令时使用数据库验证。这对于一个没有可以利用的辅助安全产品的小型用户团体来说是件好事情。其他类型的验证需要使用 external保留字来代替用户口令。

当创建用户时，必需为该用户选择口令。在用户口令上运用规则被叫做口令管理，一个公司应该有口令准则。强大的口令是不会轻易地被猜到的口令，它应超过 5个字节并不是一个在字典中可以找到的单词。如果口令是字典中的一个单词，那么一个计算机破坏者或许能够通过使用“强力攻击”的方法猜到该口令（强力攻击是一种方法：计算机破坏者使用一个用户标识并编写一个程序用来试验由字典产生的不同的口令）。口令在一定时间后也会过期且不能被同一个用户重复使用。

当Oracle使用数据库验证时便具有了提供口令管理的能力。这可以通过在一个环境资源文件中设置参数并把那个环境资源文件分配给一个用户的方式来实现。环境资源文件（profile）是一个规定了资源限度的数据库实体。当一个环境资源文件被分配给一个用户时，它在该用户上实施这些限制。可以使用企业管理器或 SQL*Plus 创建环境资源文件。数据库一定有资源限制，环境资源文件资源限制能够受此影响。要做到这一点，在 init.ora 文件中把 RESOURCE_LIMIT 参数设置为 TRUE。一个环境资源文件可以限制会话的数量、每个会话使用的 CPU 时间、CPU 调用的次数、逻辑读的次数、每次调用的逻辑读的次数、空闲时间以及连接时间。环境资源文件可以防止计算机破坏者利用所有的资源以拒绝服务的攻击方式破坏计算机。

环境资源文件可以实施口令管理规则，你可以把它们作为选项选择使用：

用户帐户的锁定——当一个用户多次注册失败后，该帐户能够被锁定一段指定的时间。

口令使用期与到期——一个给定的口令在使用一段时间后过期，此时必须修改该口令。在口令过期后，用户会得到一个宽限期；假如用户不修改口令的话，那么该帐户将被锁定。数据库/安全管理员也可以把口令设置为过期状态。

口令历史——口令历史选项检查每一个新近设定的口令以确保口令在指定长度的时间内或指定次数的口令修改中未重复使用。数据库管理员可以用 CREATE PROFILE 语句配置口令重复使用的规则。

口令复杂性验证——复杂性验证检查口令的长度以使它更难被计算机破坏者所破解。

缺省的口令复杂性验证程序要求每个口令符合以下原则：

- 长度上至少为4个字符。
- 不要与用户标识相同。
- 至少包含一个字母字符、一个数字字符和一个标点符号。
- 不要与简单单词的一个内部列表上的任何单词匹配，例如欢迎、帐户、数据库、用户等等。
- 与以前的口令至少有3个字符不同。

数据库管理员验证——由于数据库管理员任务（例如关闭或启动一个数据库）的特权性，所以他们需要一个更安全的验证模式。可以使用操作系统或口令文件实现附加验证。

操作系统——假如操作系统提供一种将用户分割到诸如 UNIX 或 NT 组中的方法，那么 Oracle 将推荐把数据库管理员放置在一个特殊的组中。这样就使 Oracle 可以通过组 ID 的辅助验证知道某个用户是数据库管理员。

使用一个口令文件验证数据库管理员——用于数据库管理员的口令文件是可选的，可以使用 ORAPWD 口令工具来设置。口令文件将把管理特权仅限于知道口令且已经被授予了一个特殊角色的用户。这些角色是 SYSOPER 和 SYSDBA：

- SYSOPER 使你能够执行 STARTUP、SHUTDOWN、ALTER DATABASE OPEN/MOUNT、ALTER DATABASE BACKUP、ARCHIVE LOG 和 RECOVER 并且包含了 RESTRICTED SESSION 特权。
- SYSDBA 包含了具有 ADMIN OPTION 的所有系统特权以及 SYSOPER 系统特权；它使你能够执行 CREATE DATABASE 和基于时间的恢复。

1. 使用ORAPWD

ORAPWD工具在操作系统的命令提示符下执行。使用如下步骤：

1) 使用ORAPWD工具创建口令文件：

```
ORAPWD FILE=filename PASSWORD=password ENTRIES=max_users
```

filename是口令文件的实际文件名，password是以数据库管理员身份注册到数据库的口令，max_users是具有数据库管理员特权的用户的最大值。

2) 把初始化参数REMOTE_LOGIN_PASSWORDFILE设置为一个有效值。该参数有3个有效值：NONE、SHARED、EXCLUSIVE。假如口令文件不存在的话，NONE值使Oracle开始工作；这是缺省值。EXCLUSIVE值完成如下工作：

- 限制一个数据库使用这个口令文件。
- 允许被授予SYSDBA和SYSOPER角色的用户。

SHARED值使一个口令文件能够被多个数据库使用。然而，一个SHARED口令文件能够识别的用户只是SYS和INTERNAL，你不能把用户添加到一个SHARED口令文件中。

3) 只要口令文件处于EXCLUSIVE模式下，通过使用GRANT命令将数据库管理特权分配给合适的用户，用户就能被添加到口令文件中。下面便是一个例子：

```
GRANT SYSDBA TO jefferson
```

```
GRANT SYSOPER TO smith
```

注意 使用REVOKE命令从口令文件中删除用户。

2. 使用SYSDBA

特权SYSDBA使用户能够执行与OSDBA相同的操作。同样，特权SYSOPER使用户能够执行与OSOPER相同的操作。

特权用户可以通过使用如下的命令连接到数据库：

```
CONNECT jefferson/password@prddb.hq.com AS SYSDBA
```

假如被操作系统验证的用户满足了操作系统验证准则，那么使用口令文件不能阻止他们进行连接。

想要列举口令文件的成员，V\$PWFILERS视图显示了所有已被授予数据库的SYSDBA和SYSOPER系统特权的用户。

23.1.2 外部验证

外部验证依赖于操作系统或网络验证服务。尽管Oracle仍然对用户进行识别，但是外部验证等于在Oracle外放置了用于口令管理和用户验证的控制。此类登录不需要数据库口令。要使用该选项，在数据库init.ora文件中设置OS_AUTHENT_PREFIX参数。这告诉Oracle任何同此值有相同前缀的用户将要在外部得到验证。例如，假如该值被设置为ops\$并且你有两个分别名为ops\$jones与smith的用户，那么Oracle不需要来自ops\$jones的口令，但是它需要来自smith的口令。此参数能够被设置为你所希望的任何前缀，甚至能够通过指定一组空的双引号被设置为一个空串。init.ora参数REMOTE_OS_AUTHENT必须被设置为TRUE（缺省值是FALSE）以使Oracle能够使用来自一个非安全连接的用户名。这样可以避免一个潜在的计算机破坏者伪装成一个合法的用户。

网络验证用 Oracle 高级安全 (OAS) 选项来实现并且能够用如下技术验证用户：

网络验证服务 (例如 Kerberos 和 SESAME) —— 启用一个中央资源用于口令管理并且能够使用第三方的软件实施单独注册。用户在他将要使用的每个数据库中创建并且数据库特权被分配给该用户，但是口令是保留字 `external`。这告知 Oracle 仅识别用户并启用外部资源对口令进行验证。OAS 使用一个具有用户名、口令和主机名的验证服务器来验证口令。假如口令得到验证，那么用户便能够对 Oracle 数据库进行存取。

令牌设备——令牌是用户建立一个到数据库的连接时必须的物理设备。令牌可以是在设备中产生的一次性数字口令，其容量是一个厚信用卡的容量。此数字口令必须用来同一个短的数字型个人识别号码 (PIN) 相连接。Oracle 服务器有一个添加到配置中的辅助安全服务用来跟踪令牌的口令。另一种方法是询问/应答。给用户发送一个数字 (询问)，用户在一台设备上键入该数字，该设备给出作为口令使用的另一个数字 (应答)。

生物统计学设备——使用一个对于个人来说是唯一的物理特征，当前能够用于 OAS 的是指纹扫描设备。用户指纹必须首先记录在系统中，然后该用户指定 Oracle 服务并把他的手指放置在指纹读取器上。在读取器上得到的指纹将与数据库中的指纹进行比较。

23.1.3 企业验证

企业验证启用一个中央资源用于口令管理并能够使用 Oracle 安全服务 (OSS) 实施单独注册。该用户被称为全局用户 (global user) 并且必须在每个使用全局口令的数据库中创建。这告诉 Oracle 仅识别这个用户并且启用 Oracle 安全服务验证该口令以及传送用户企业验证。假如口令得到验证，用户便能够访问 Oracle 数据库。Oracle 安全服务与 Oracle 企业管理器的接口集中了安全角色管理和企业验证。这使得当前正在被管理的用户具有全局同一性。

1. 企业角色

企业角色是一个或多个全局角色的容器。企业角色现在存储在一个因特网目录中，它们也可以存储于一个允许使用轻量目录访问协议 (LDAP) 的目录服务器。

全局角色不同于数据库角色，因为它们使你能够把验证信息跨多个数据库分配给 (全局) 用户。当一个全局用户登录到一个数据库时，全局角色便被动态地分配给该用户。全局角色必须首先被分配给 Oracle 安全服务器中的一个全局用户，然后全局角色具有与数据库服务器中的每个全局角色相关的特权。与全局角色相关的特权在不同的数据库中是不同的。

2. 表空间分配与使用

当创建用户时，你必须告诉 Oracle 你想把用户在数据库中创建的对象存储在哪里；如果一个存储子句没有指定放置对象的位置，就放置在用户的缺省表空间 (default tablespace)。应当指定缺省表空间以防止诸如表或索引的数据库对象在系统表空间中创建；假如用户没有在数据库中创建对象的特权，你可以把缺省值设置为系统表空间。如果用户在系统表空间中创建对象，会形成表空间碎片或空间溢出。临时表空间实际上是一个排序工作区并且由 SQL 语句使用 (例如 ORDER BY 与 GROUP BY)。当用户建立一个索引时也使用临时表空间。由于与数据字典的争用不断增加，所以你应当指定一个临时表空间，而不是系统表空间。

表空间定额限制了用户在表空间中创建的数据库对象的大小。缺省情况下没有定额容量限制，但是假如用户已经在一个表空间中分配了数据库对象并且想要限制该表空间的使用，那么就把那个用户的表空间的定额设置为 0。有了此限制，在那个表空间中的当前对象不能被

分配任何更多的空间，但是它们仍然在表空间中。

23.2 数据库权限管理

特权可以是对象特权，也可以是系统特权。有超过 60种不同的系统特权用于用户在数据库中执行管理活动（参见表 23-1）。

表23-1 系统特权

特 权	所能实现的操作
分析	
ANALYZE ANY	分析数据库中的任何表、簇或索引
审计	
AUDIT ANY	审计数据库中的任何模式对象
AUDIT SYSTEM	启用与停用语句和特权的审计选项
簇	
CREATE CLUSTER	在自有的模式中创建一个簇
CREATE ANY CLUSTER	在任何一个模式中创建一个簇；操作类似于 CREATE ANY TABLE
ALTER ANY CLUSTER	改变数据库中的任何一个簇
DROP ANY CLUSTER	删除数据库中的任何一个簇
数据库	
ALTER DATABASE	改变数据库；不管操作系统的特权，经由 Oracle把文件添加到操作系统中
数据库链接	
CREATE DATABASE LINK	在自有模式中创建专用数据库链接
索引	
CREATE ANY INDEX	在任何表的任何模式中创建一条索引
ALTER ANY INDEX	改变数据库中的任何索引
DROP ANY INDEX	删除数据库中的任何索引
库	
CREATE LIBRARY	在自有模式中创建调出库
CREATE ANY LIBRARY	在任何模式中创建调出库
DROP LIBRARY	删除自有模式中的调出库
DROP ANY LIBRARY	删除任何模式中的调出库
特权	
GRANT ANY PRIVILEGE	授予任何系统特权（不包括对象特权）
过程	
CREATE PROCEDURE	在自有模式中创建存储的过程、函数和包
CREATE ANY PROCEDURE	在任何模式中创建存储的过程、函数和包（这要求用户还要有ALTER ANY TABLE、BACKUP ANY TABLE、DROP ANY TABLE、SELECT ANY TABLE、INSERT ANY TABLE、UPDATE ANY TABLE、DELETE ANY TABLE或GRANT ANY TABLE特权
ALTER ANY PROCEDURE	编译任何模式中的任何存储的过程、函数或包
DROP ANY PROCEDURE	删除任何模式中的任何存储的过程、函数或包
EXECUTE ANY PROCEDURE	执行任何过程或函数（独立的或成组的），或在任何模式中引用任何公共包变量
环境资源文件	
CREATE PROFILE	创建环境资源文件
ALTER PROFILE	改变数据库中的任何环境资源文件
DROP PROFILE	删除数据库中的任何环境资源文件
ALTER RESOURCE COST	设置所有的用户会话中使用的资源开销

(续)

特 权	所能实现的操作
公共数据库链接	
CREATE PUBLIC DATABASE LINK	创建公共数据库链接
DROP PUBLIC DATABASE LINK	删除公共数据库链接
公共同义词	
CREATE PUBLIC SYNONYM	创建公共同义词
DROP PUBLIC SYNONYM	删除公共同义词
角色	
CREATE ROLE	创建角色
ALTER ANY ROLE	改变数据库中的任何一个角色
DROP ANY ROLE	删除数据库中的任何一个角色
GRANT ANY ROLE	授权数据库中的任何一个角色
回滚段	
CREATE ROLLBACK SEGMENT	创建回滚段
ALTER ROLLBACK SEGMENT	改变回滚段
DROP ROLLBACK SEGMENT	删除回滚段
会话	
CREATE SESSION	连接到数据库
ALTER SESSION	发出ALTER SESSION语句
RESTRICTED SESSION	当数据库利用 STARTUP RESTRICT 启动时进行连接 (OSOPER 与 OSDBA 角色包含此特权)
序列	
CREATE SEQUENCE	在自有模式中创建序列
CREATE ANY SEQUENCE	在任何模式中创建任何序列
ALTER ANY SEQUENCE	在任何模式中改变任何序列
DROP ANY SEQUENCE	在任何模式中删除任何序列
SELECT ANY SEQUENCE	在任何模式中引用任何序列
快照	
CREATE SNAPSHOT	在自有模式中创建快照 (用户还必须具有 CREATE TABLE 特权)
CREATE ANY SNAPSHOT	在任何模式中创建快照 (用户还必须具有 CREATE ANY TABLE 特权)
ALTER SNAPSHOT	改变任何模式中的任何快照
DROP ANY SNAPSHOT	删除任何模式中的任何快照
同义词	
CREATE SYNONYM	在自有模式中创建同义词
CREATE ANY SYNONYM	在任何模式中创建任何同义词
DROP ANY SYNONYM	在任何模式中删除任何同义词
系统	
ALTER SYSTEM	发出ALTER SYSTEM语句
表	
CREATE TABLE	在自有模式中创建表。还使被授权者能在自有模式下的表中创建索引, 包括那些用于完整性约束的索引 (被授权者必须有表空间的定额或 UNLIMITED TABLESPACE 特权)
CREATE ANY TABLE	在任何模式中创建表 (假如被授权者有 CREATE ANY TABLE 特权并在另一个用户模式中创建了一张表, 那么拥有者必须在那个表空间上有空间定额。表的拥有者不必具有 CREATE [ANY] TABLE 特权)
ALTER ANY TABLE	改变任何模式中的任何表并编译任何模式中的任何视图
BACKUP ANY TABLE	在任何模式中使用表的导出工具执行一个增量导出操作
DROP ANY TABLE	删除或截断任何模式中的任何表
LOCK ANY TABLE	锁定任何模式中的任何表或视图

(续)

特 权	所能实现的操作
COMMENT ANY TABLE	对任何模式中的任何表、视图或列进行注释
SELECT ANY TABLE	对任何模式中的任何表、视图或快照进行查询
INSERT ANY TABLE	把行插入到任何模式中的任何表或视图中
UPDATE ANY TABLE	修改任何模式中的任何表或视图中的行
DELETE ANY TABLE	删除任何模式中的任何表或视图中的行
表空间	
CREATE TABLESPACE	创建表空间；不管用户有何操作系统特权，经由 Oracle把文件添加到操作系统中
ALTER TABLESPACE	改变表空间；不管用户有何操作系统特权，经由 Oracle把文件添加到操作系统中
MANAGE TABLESPACE	使任何表空间脱机，使任何表空间联机，开始和结束对任何表空间的备份
DROP TABLESPACE	删除表空间
UNLIMITED TABLESPACE	使用任何没有数量限制的表空间。此特权忽略了所分配的任何具体定额。假如被取消的话，被授权者的模式对象仍然保留，但是进一步的表空间分配被拒绝，除非这一分配是具体的表空间定额允许的。此系统特权仅可以授予用户，而不授予角色。一般而言，应分配具体的表空间定额，而不授予此系统特权
事务	
FORCE TRANSACTION	强迫提交或回滚本地数据库中悬而未决的自有的分布式事务
FORCE ANY TRANSACTION	强迫提交或回滚本地数据库中悬而未决的任何分布式事务
触发器	
CREATE TRIGGER	在自有模式中创建触发器
CREATE ANY TRIGGER	在任何模式中创建与任何模式的任何表相关的任何触发器
ALTER ANY TRIGGER	启用、停用或编译任何模式中的任何触发器
DROP ANY TRIGGER	删除任何模式中的任何触发器
用户	
CREATE ANY USER	创建用户；分配任意表空间上的定额，设置缺省和临时表空间，指定一个环境资源文件（在 CREATE USER 语句中）
BECOME ANY USER	成为另一个用户（这是任何一个执行完全数据库导入的用户所需要的）
ALTER USER	改变其他用户：修改任意用户的口令或验证方法，分配表空间定额，设置缺省或临时表空间，在 ALTER USER 语句中指定环境资源文件与缺省角色（不必改变自有口令）
DROP USER	删除另一个用户
视图	
CREATE VIEW	在自有模式中创建视图
CREATE ANY VIEW	在任意模式中创建视图。要在另一个用户模式中创建视图，你必须具有 CREATE ANY VIEW 特权，拥用者必须在该视图引用的对象上具有所需的特权
DROP ANY VIEW	删除任意模式中的任意视图

你应把这些特权仅授予管理数据库的用户。对象特权允许对数据库对象的存取与维护；此类特权用于终端用户。对象特权（参见表 23-2）能够直接分配给用户，或者该特权被授予一个角色，然后该角色被授予给用户。

表23-2 对象特权

对 象	所能使用的SQL语句
ALTER	ALTER对象（表或序列）
DELETE	DELETE FROM对象（表或视图）

(续)

特 权	所能实现的操作
EXECUTE	EXECUTE对象（过程或函数）；引用公共包变量
INDEX	CREATE INDEX ON对象（仅有表）
INSERT	INSERT INTO对象（表或视图）
REFERENCES	在对象（仅有表）上定义一个 FOREIGN KEY 完整性约束的 CREATE 或 ALTER TABLE 语句
SELECT	SELECT...FROM对象（表、视图或快照）；使用序列的 SQL 语句
UPDATE	UPDATE对象（表或视图）

23.2.1 理解安全角色

角色（role）是一个数据库实体，该实体是一个已命名的特权组。这样就在特权与角色之间创建了一个多对一关系，所以对于一个角色可有许多特权。角色是数据库中唯一的数据库实体并且不被用户所有。你可以使用 CREATE ROLE 语句创建一个角色，该角色不能与用户同名。假如用户有创建角色的特权的话，他们也可以创建角色。当你创建一个角色时，该角色成为你的缺省角色集的一部分。

1. 角色验证

角色的使用可以通过口令得到验证。这增强了角色的安全性并且使不知道口令的用户不能启用该角色。不幸的是，该角色的口令必须在应用程序的某处进行硬编码。当一个角色已经被授予用户但还不是该用户缺省角色集的一部分时，SET ROLE 命令能够动态地启用该角色。只要用户在 Oracle 中创建一个会话，缺省角色集便自动被启用。在 Oracle 以前的版本中，这是防止用户从除了应用外的其他任何地方存取数据的最好办法。例如，一个用户或许已经经由 payroll_role 角色存取 PAYROLL 表。假如该角色是用户缺省角色集的一部分，那么该用户就可以使用软件而不是工资单应用程序修改数据。即便使用 SET ROLE 命令，用户仍然可以使这个高级的安全特性失效。Oracle 现在已经增加了安全应用角色的功能。这些新类型的角色仅能通过一个信任包被启用。信任包证实用户不是直接连接到数据库，而是正在使用应用。创建一个安全应用角色的语法如下：

```
CREATE ROLE payroll_role IDENTIFIED USING payroll_pkg.admin;
```

该角色是 payroll_role，它正在使用信任包 payroll_pkg.admin。

2. 通过操作系统的角色验证

下面的语句创建一个名为 ACCTS_REC 的角色并需要操作系统验证它的使用：

```
CREATE ROLE role IDENTIFIED EXTERNALLY;
```

仅在操作系统可以动态地将操作系统特权（OSP）与应用链接时才能实现操作系统角色验证。当用户启动一个应用时，操作系统把一个 OSP 授予给该用户。被授予的 OSP 对应于与该应用有关的角色。此时，应用能够启用应用角色。

如果一个角色被操作系统授权，你必须在操作系统级上配置每个用户的信息。此项操作依赖于操作系统。

数据库有对于用户可以具有的作为缺省角色集一部分的角色数量的软限制。如果用户具有的角色数多于限制值，那么在登录时，他会收到一条错误信息。如果发生了此错误，在 init.ora 文件中检查 MAX_ENABLED_ROLES；或许需要增加该值。一个角色还可以被授予给

另一个角色，这样被授予的角色的所有特权可以得到继承。

23.2.2 理解管理

管理员仅有他们需要的用于数据库管理的权限。一个数据库环境中的主要管理任务可以被分为两个方面：安全管理与数据库管理。安全管理员（ security administrator ）有权创建、改变并删除用户以及维护安全角色和用户环境资源文件，但是他没有数据库管理员的特权，例如打开或关闭一个数据库。

当你把一个角色授予给一个管理角色特权的用户时，你必须用 ADMIN选项来完成这项操作。该选项使用户能够执行管理操作，例如改变或删除角色、使用 ADMIN选项把角色授予给其他用户。除了管理员以外，该选项不应给任何人。

23.3 Oracle企业安全管理器

Oracle企业安全管理器使管理员能够使用一个工具来管理遍及该企业的 Oracle数据库安全的所有方面。该工具允许你创建一个用户、为用户分配缺省表空间和临时表空间以及他们的口令。一旦创建了用户，那么随之为他们分配特权。同全局角色一样，也能够在这里创建和维护非企业角色。单一的控制台使你能够在因特网目录中创建用户、在多个 Oracle 8i数据库中创建用户、创建并管理企业角色。要想了解此产品的更多信息，参见第 27章。

23.4 监控数据库资产

Oracle能够审计并记录数据库中发生的活动。你可以使用 AUDIT SQL命令允许审计操作或使用NOAUDIT SQL命令禁止审计操作。有三种类型的审计操作：登录尝试、对象存取（具体对象上的具体语句）和数据库操作（具体的系统特权和语句，不考虑对象）。

任何一条命令，不管是成功的还是不成功的都可以在这些类别中进行审计。要创建审计系统视图，以用户SYS的身份运行CATAUDIT.SQL文件脚本并设置init.ora参数AUDIT_TRAIL。你可以把AUDIT_TRAIL设置为写入数据库或一个操作系统文件：AUDIT_TRAIL=DB写入数据库，AUDIT_TRAIL=OS写入一个操作系统文件。为了使新的 init.ora参数生效，你必须重新启动Oracle实例。假如你把AUDIT_TRAIL参数设置为DB，那么所有已审计的活动将被写入SYS.AUD\$表中。你应将SYS.AUD\$表的表空间存储参数从系统表空间修改为审计目的而创建的表空间。你能够从SQL*WORKSHEET或SQL*Plus中发出所有的审计命令。

23.4.1 审计登录

AUDIT ANY特权是发出审计命令所必须的。使用 AUDIT SESSION命令完成审计登录。此命令可以用来对所有成功和不成功的与数据库建立连接的尝试进行审计。使用 AUDIT SESSION WHENEVER NOT SUCCESSFUL命令仅审计不成功的尝试。使用AUDIT SESSION WHENEVER SUCCESSFUL命令仅审计成功的尝试。如果审计数据存储在 SYS.AUD\$表中的话，你可以生成审计报表。使用DBA_AUDIT_SESSION视图报告登录尝试：

```
SELECT
os_username, /* O/S user name */
username, /* Oracle user name */
to_char(timestamp,'DD-MON-YY HH24:MI'), /* Login time */
```

```
to_char(logoff_time, 'DD-MON-YY HH24:MI') /* Logoff time */
FROM dba_audit_session
ORDER BY os_username;
```

```
OUTPUT TO QUERY
```

```
OS_USERNAME    USERNAME    TIMESTAMP    LOGOFF
JOHNES         JOHNES      24-MAR-97 18:00    14-FEB-97 18:05
```

23.4.2 审计数据库操作

通过审计数据库操作，可以在语句和系统特权级上进行审计，而不必考虑具体的数据库对象。AUDIT语句的审计功能特性使你能够审计不止一条 SQL语句。例如，使用审计语句ROLE审计SQL语句CREATE ROLE、ALTER ROLE、SET ROLE和DROP ROLE。要审计系统特权，你必须指定该特权。系统特权 ALTER DATABASE不包含在语句审计选项中，但是你仍然可以审计它，因为它是一个系统特权。有些语句审计选项使用与系统特权相同的名称。你可以把AUDIT语句用于用户、会话或存取。你可以进一步地定义它以便仅对成功或不成功的语句进行审计。可以被审计的语句见表 23-3。

表23-3 语句审计选项

语 句	选 项
OPTION	SQL STATEMENT
ALTER SYSTEM	ALTER SYSTEM
CLUSTER	CREATE CLUSTER, ALTER CLUSTER, TRUNCATE CLUSTER, DROP
CLUSTER	
DATABASE LINK	CREATE DATABASE LINK, DROP DATABASE LINK
INDEX	CREATE INDEX, ALTER INDEX, DROP INDEX
NOT EXISTS	因为指定的结构或对象不存在而返回错误的所有 SQL语句
PROCEDURE	CREATE [OR REPLACE] FUNCTION, CREATE [OR REPLACE] PACKAGE, CREATE [OR REPLACE] PACKAGE BODY, CREATE [OR REPLACE] PROCEDURE, DROP PACKAGE, DROP PROCEDURE
PUBLIC DATABASE LINK	CREATE PUBLIC DATABASE LINK, DROP PUBLIC DATABASE LINK
PUBLIC SYNONYM	CREATE PUBLIC SYNONYM, DROP PUBLIC SYNONYM
ROLE	CREATE ROLE, ALTER ROLE, SET ROLE, DROP ROLE
ROLLBACK SEGMENT	CREATE ROLLBACK SEGMENT, ALTER DROPBACK SEGMENT, DROP ROLLBACK SEGMENT
SEQUENCE	CREATE SEQUENCE, DROP SEQUENCE
SESSION	连接和断开连接
SYNONYM	CREATE SYNONYM, DROP SYNONYM
SYSTEM AUDIT	AUDIT, NOAUDIT
SYSTEM GRANT	GRANT系统特权/角色TO用户/角色REVOKE系统特权/角色FROM用户/角色
TABLE	CREATE TABLE, ALTER TABLE, DROP TABLE
TABLESPACE	CREATE TABLESPACE, ALTER, TABLESPACE DROP TABLESPACE
TRIGGER	带有ENABLE、DISABLE和DROP子句的CREATE TRIGGER, ALTER TRIGGER, ENABLE或DISABLE, ALTER TABLE
USER	CREATE USER, ALTER USER, DROP USER
VIEW	CREATE[OR REPLACE] VIEW, DROP VIEW

23.4.3 审计数据库对象上的DML

你可以使用如下语法审计一个具体的模式对象：

```
AUDIT object_opt ON schema.object  
  BY SESSION/ACCESS WHENEVER NOT/SUCCESSFUL;
```

你可以指定一个对象选项（例如插入或更新）或者使用关键字ALL来指定所有的对象选项。

```
AUDIT insert,update  
ON scott.emp_table  
WHENEVER NOT SUCCESSFUL;  
  
AUDIT ALL  
  ON scott.emp_table  
  WHENEVER NOT SUCCESSFUL;
```

23.4.4 管理审计

如果你选择在数据库中存储审计记录，那么最好每天运行审计表上的报告，然后删除审计数据以节省空间。审计表只能进行有限地访问，对于 SYS.AUD\$表的所有活动应通过使用下面的语句被记录：

```
AUDIT INSERT, UPDATE, DELETE, SELECT  
ON sys.aud$  
BY ACCESS;
```

审计可能在数据库中造成过多的系统开销，因此它是可选的。审计语句中的 BY SESSION子句使Oracle为在同一个会话中使用的相同类型的所有SQL语句只写下一条单独的记录。在审计语句中的BY ACCESS子句使Oracle为每一条已审计的语句写下一条记录。假如你审计数据定义语言（DDL）语句，不管指定哪一条子句，Oracle都会自动地使用BY ACCESS。为了维护一个安全环境，你应当执行一个策略以决定所要审计的操作。

下面的AUDIT语句显示如何使用某些更具限制性的选项：

```
AUDIT statement_opt/system_privileges  
  BY user (optional)  
  BY session/access WHENEVER NOT/SUCCESSFUL;
```

下面的代码示例显示了如何审计与角色和会话有关的语句：

```
AUDIT role;  
AUDIT session whenever not successful;.
```

23.5 保护数据完整性

你可以通过使用安全用户连接与加密算法来保护数据完整性。当用户注册到一个 Oracle数据库时可以对口令进行加密。使用一个修改的 DES（数据加密标准）算法对口令进行加密。密码通过在客户机上设置一个环境变量以及在数据库服务器上设置 init.ora参数来打开。你必须把客户机的环境变量ORA_ENCRYPT_LOGIN设置为TRUE。

注意 此变量根据操作系统而变化。你还必须在数据库服务器上把 init.ora参数DBLINK_ENCRYPT_LOGIN设置为TRUE。

当一个数据库会话被初始化时，Oracle对用户标识对应的口令进行加密，但是它不对口令的修改进行加密。

计算机破坏者能够通过修改在网络上传播的数据或 DML损害数据的完整性，这会使数据库的逻辑遭到破坏。当数据库的完整性出现问题时，重新恢复的代价很大并且非常花费时间。

把你的所有数据加密对潜在的安全破坏来说是理想的解决方案，因为如果数据被动地受到计算机破坏者的监控，他们不能使用这些数据，可能会去寻找容易一些的破坏对象。Oracle的高级安全措施（OAS）提供了辅助的保密措施。Oracle的高级安全措施当前提供了两种加密算法：RSA和DES，它们具有不同的密钥长度。对于在美国和加拿大使用的产品，可以使用56位的RSA RC4、56位的DES和128位的RSA；对于美国和加拿大以外的出口产品，可以使用40位的RSA RC4和40位的DES40、3DES。数据完整性通过使用密码校验和得到保护，这个密码校验和使用MD5算法或SHA算法（安全哈希算法）。这将保证数据不会在它离开客户工作站与到达数据库服务器之间的这段时间里被篡改。

23.6 Oracle 8i因特网安全特性

Oracle 8i高级安全选项提供了一组给人深刻印象的安全特性，这些安全特性的设计考虑到了因特网标准。今天的体系结构是多层的并可能通过许多网络连接。这造成了一个令人担心的安全问题。存在连接到应用服务器的瘦客户以及连接到数据库服务器的应用服务器。数据库服务器不仅必须能确认用户就是他自己所说明的身份，而且应能确认应用服务器就是它自己所说明的身份。可以使用两种方法进行这些类型的验证：口令与数字证书。口令已经在前面的23.1节中讨论过了，所以这里要讨论数字证书。

23.6.1 使用数字证书

数字证书能够用来验证用户或机器。此类证书是 X.509版本3，是一个基于因特网标准的公用密钥基础结构（PKI）。证书包含如下内容：

- 可以被辨别的证书拥有者的名字。
- 可以被辨别的证书管理机构的名字。
- 证书拥有者的公用密钥。
- 证书有效的数据范围。
- 发行人的签名。
- 证书的序列号。

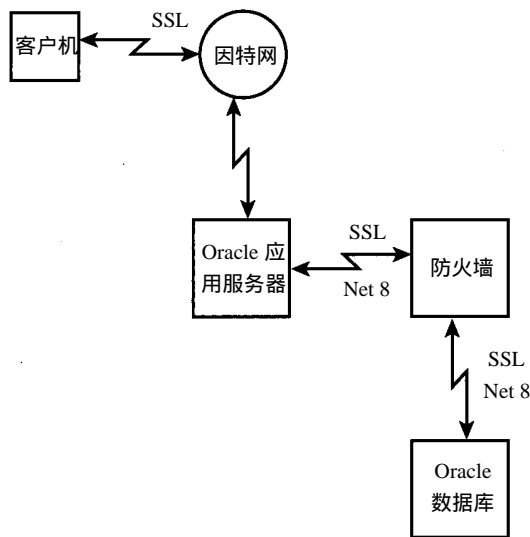


图23-1 使用数字证书进行通信

这些证书必须以网络安全套接字层协议（SSL）的方式使用。SSL也提供了跨越网络的数据加密和数据完整性检查。加密是保持数据私有的关键。当数据位于到数据库的途中时（参见图23-1），数据完整性检查使计算机破坏者不能修改该数据。Oracle证书管理机构，或另一个证书管理机构在Oracle目录服务器或另一个LDAP目录结构中发布“钱包”。Oracle使用一个“Oracle钱包”存储证书和私有密钥。Oracle钱包管理器是管理钱包内容的一个主要工具。

23.6.2 使用RADIUS协议的高级验证

既然每一台机器或用户可以有一个数字证书，那么Oracle可以使用一种比口令更强大的验

证方法。RADIUS（远程拨号用户服务器验证）协议已经变成了一种因特网标准，Oracle 8i高级安全选项同任何符合RADIUS的安全服务器一起工作。此协议支持口令、令牌、生物学统计以及智能卡。

23.7 防火墙支持

防火墙用于防止公共因特网通信量进入到一个公司的专用网络中。它还用于保护与连接到因特网的服务器进行通信的服务器（参见图 23-2）。主要的防火墙厂商现在提供代理来使 NET 8 和 SQL*NET通信量能够穿过防火墙，这样为数据库服务器提供了保护，使之免遭计算机破坏者的攻击。计算机破坏者或许试图在数据库服务器操作系统上实施攻击，但是如果防火墙仅允许 NET8 和 SQL*NET通信量通过的话，那么这个攻击是不可能实现的，所以数据库服务器是安全的。

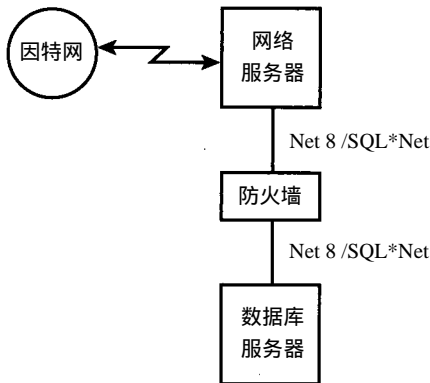


图23-2 防火墙支持

23.8 好的粒状存取控制

虚拟专用数据库保证用户仅能查看他有权使用的数据。这使得一个已存在的成品数据库能够在Oracle的担保下在因特网上被访问，Oracle保证终端用户仅能查看他自己的数据。此安全措施保存在数据库服务器上。这防止了它被诸如报告书写器等客户软件所忽视。一个虚拟专用数据库被相关的表或视图定义为一个安全策略。在此策略执行后，对表或视图的任何存取会引发数据服务器调用一个实施该策略的函数。在 SQL语句被执行以前，该函数返回被添加到WHERE子句中的代码。这个附加的代码包含了已经用于保证数据安全的存取规则。你可以为同一张表的不同 DML定义不同的策略（例如，雇员表的 INSERT可以有与雇员表的 DELETE不同的策略）。这些SQL语句被进行完整的语法分析与优化，然后存在于共享池中，在可用时由其他用户使用。

23.9 数据库资源管理器

数据库资源管理器管理多个 CPU以及多个实例能够从操作系统中获得的并行度。这是一个重要的安全与管理特性。从安全的角度出发，它能够防止对因特网上使用的生产环境进行的拒绝服务攻击。假如那个数据库同其他没有用于因特网的数据库共享系统资源的话，它是一个管理附加物。有四种机制用于管理数据库资源：

- 1) 在用户层分配资源消费者组，所有这些会话是该消费者组的一部分。
- 2) 资源计划包含资源消费者组或其他资源计划；它们还包含一个如何划分资源的规范。
- 3) 资源分配方法是某些资源采用的策略。资源分配方法由资源计划和资源消费者组来使用。
- 4) 资源计划指令让管理员为计划分配成员，然后管理那些成员使用的资源。

通过创建一个因特网资源组并限制该组资源数量，管理员能够更好地管理连接到因特网上的产品环境。他可以预防一个资源组首先执行并随之消耗掉操作系统中所有的计算机资源。

23.10 硬件安全

计算机硬件需要存放在一个限定的区域中，在这里，存取受到限制并且有合适的火灾控制机制。电流应当是清洁的，这意味着不会遭受电流起伏或停电的干扰。清洁的电流通常由不间断电源（UPS）提供。

23.11 恢复丢失的数据

在你的安全计划中，你必须指定如何恢复由于违反了安全性而丢失的数据。我认为：“你的系统实际上等于你最后的备份与你的恢复能力”。这句话概括了在数据库恢复方面所采用的基本原理。下面是一些涉及到备份的有益问题：

如何保证备份介质物理上的安全？

当备份完成时，备份介质是否要脱离站点以防止数据中心丢失的情况？

按照制造者的准则，备份介质被替换吗？

定期清理备份硬件吗？

备份是在恢复时是可读的、可用的？

存在一个显示备份介质在某日使用过以及介质已经备份了多少次的备份日志吗？

有两种方法备份 Oracle 数据库。第一种方法是通过使用一个叫做物理备份（physical backup）的操作系统备份；第二种方法是通过使用 Oracle 的导出工具对数据库进行备份，Oracle 导出工具创建的是逻辑备份（logical backup）。

23.11.1 操作系统备份

操作系统备份需要一个专用于操作系统的工具，它能够使数据库管理员通过使用一个操作系统恢复工具来恢复数据库。你可以在数据库关闭（冷备份）或打开（热备份）的情况下执行操作系统备份。热备份用在需要数据库可用性高的场合。热备份使表空间能够在联机时得到备份，或在脱机和不可用时用于事务处理。冷备份必须包括所有的 Oracle 数据文件、控制文件和联机重做日志文件（仅供参考，我把这些文件叫做操作系统备份文件集）。热备份被认为是部分备份，因为它仅备份所需的文件。

你可以在 ARCHIVELOG 模式和 NOARCHIVELOG 模式下运行数据库。ARCHIVELOG 模式指当联机重做日志被填充时，它们的内容被归档，这样便释放了当前的重做日志。归档重做日志用于恢复它们所记录的事务或在数据库恢复时重做它们。要执行一个热备份，数据库必须处于 ARCHIVELOG 模式。NOARCHIVELOG 模式意指联机重做日志在它们充满时并不归档，而是被最新的事务改写。因此，仅仅当前重做日志可用于恢复。假如不归档，你必须做一个操作系统文件集的备份，在数据库被恢复时，你必须恢复操作系统备份文件集中的所有文件。

冷备份的优点是恢复过程中的处理步骤最少。这通常意味着最小的出错几率，冷备份比热备份速度要快。

23.11.2 逻辑备份

使用 Oracle 导出工具的逻辑备份创建了一个文件，该文件含有用于重新创建数据结构和数

据库中所包含的数据的信息。该信息能够被存储在一个磁盘设备或磁带介质中。逻辑备份比物理备份花费的时间要长并且在准备恢复数据库时或许需要完成一些辅助的工作。有三种导出模式：

用户模式，备份用户所拥有的所有对象。

表模式，备份用户所拥有的指定的表。

完全数据库模式，备份数据库的所有对象。

你可以使用一个参数文件对数据库中的数据执行常规备份来控制导出工具。你可以在数据集合的三个层次上执行导出操作：增量、累积和完整。增量导出（incremental export）仅对那些自上次增量导出以来发生变化的数据进行备份。累积导出（cumulative export）仅从那些自上次累积导出以来发生变化的表中导出数据；此导出操作用于压缩增量导出。完全导出（complete export）导出数据库中包含的所有数据。由于所收集到的数据量较大，你只能有限地执行完全导出。

导出可以非常灵活，但是定期地执行导出是很重要的。一个好方案是在每月的第一天执行一次完全导出，每天执行一次增量导出，每周执行一次累积导出。