限制用户的访问权限

限制用户的访问权限

在上一个技术专题中我们提到了数据库安全，本篇继续介绍数据库安全的内容：如何限制用户的访问权限，并详细阐述了有关如何限制单一用户的访问权限、限制多个用户的访问权限、确保不同用户的安全以及如何限制用户使用其他方式访问数据库这四个方面的内容。

限制单一用户访问权限

TNS 能限制单一用户吗？如：sqlplus scott/tiger@ONLYSCOTTALLOWED，其他用户都不会使用 ONLYSCOTTALLOWED db 命令。对于访问唯一授权的 Oracle 用户最好访问办法就是通过授权 DB 对象保证你有权使用这些对象。这样，通过 Web 应用程序就可以访问。若要为我们运行 Oracle 的用户开发一个安全系统而又要保证这些数据不被任何其他支持 Oracle 的用户看见，最佳方法就是创建一个帐户。V$SESSION 中的 program 栏通过应用程序将信息传送给 Oracle。但 Oracle 并不总能读取应用程序。想要阻止某些狡猾的用户未经确认就访问数据库，你可以输入一个简单的 AFTER LOGON 触发器。

- TNS 能限制单一用户吗
- 怎样访问唯一被授权的用户
- 如何保证用户安全登录到寄宿的数据库
- 如何通过触发器限制用户访问数据库
- 授予用户仅能访问视图不能访问表的权限

限制多个用户访问权限

有没有办法能让 Oracle8.0，8i 以及 9i 阻止某些应用程序如 Microsoft Access 访问数据库？一种方法就是使用安全应用程序的 Role。我们公司里的开发人员想要访问数据库，然而我不想让他们这样做。相反，我用 SQL*Plus HTML 的生成功能为开发者生成 Web 页面，并且调出了他们需要的信息，我认为 这种方法很好。

赋予 PUBLIC 权限更加简单，但也更加危险。对于数据库安全，有一个惯例就是只将权限赋予给需要它们的用户。此外，你只能让用户执行最小的命令。在今天的 IT 安全意识环境中，授予更多人更多的权限是打开你的数据库大的安全漏洞的途径之一，也值得你在它上面花心思。

- 如何阻止用户访问数据库
- SQL HTML
- 用 rolegrant 语句和 public synonym

确保不同用户的安全

如果在不同的环境中有不一样的 DBA 组群，那么你应该为他们每个人创建各自的 ORACLE_HOME。Oracle 二进制的 link time 能识别 OSDBA 和 OSOPER 组群身份。另外，在数据库安全方面，封闭的安全政策比开放的要好。如果想要限制拥有相同 role 的用户访问数据库的某些对象，最好为每个公司创建不同的 Role 文档。

- 在 Unix 上针对不同的用户群安全指南
- 封闭、开放的安全政策和 RBAC role 层级里的 permission 语句
- 怎样阻止越权存储到格式程序
- 限制用户从其他分公司上访问某些对象

限制用户用 SQL Plus 脚本或其他 IP 地址访问数据库

许多 Oracle 产品都使用 PRODUCT_USER_PROFILE，SYSTEM 帐户里的表，提供产品层安全来补充 SQL GRANT 产品层安全，并且撤销用户 role。这个表能用于限制用户从 SQL*Plus 访问 Oraclel 对象。我们提到的这个技巧适用于 Oracle 版本 8，8i 和 9i。这一技巧包括如何创建表，表结构、如何对它进行管理以及禁用 SQL*Plus、SQL 以及 PL/SQL 命令。此外，本节还详细解决了如何限制用户通过 SQL Plus 语句访问数据库或数据库里的对象等。

- 如何限制用户在 SQLPlus 脚本里执行具体命令？
- 限制用户通过 SQLPlus 访问数据库
- 限制用户通过 SQLPlus 语句访问 Oracle 对象
- 限制 Oracle 数据库里的特殊 IP 地址登录

TNS 能限制单一用户吗？

问：TNS 能限制单一用户吗？如：sqlplus scott/tiger@ONLYSCOTTALLOWED。其他用户都不会使用 ONLYSCOTTALLOWED db 连接。如果可以的话，请您教我一种解决方法。我正在 PHP 上开发一种 Web 应用程序，我想用这个程序来访问唯一授权的 Oracle 用户（这个单一的用户在其他用户表里仅有 select 权限）。

答：有一些 Oracle 用户能使用这一连接，所以通过 grant 或者 Role 语句还不能解决这个问题。

在 TNS 层上还没有建立允许结构。此外，我们不能将安全和不安全混淆。通过隐藏 TNS 连接说明符，你可以使用户更难连接数据库，但是你不能阻止他们访问数据库。

你该在数据附近每层建立安全。在 Oracle 环境中你应该从数据库安全开始着手。如果单一的用户需要进行特殊访问，他们就应该被授予数据库帐号和特殊访问权限。如果你的客户系统有静态 IP 地址，你可能希望用 sqlnet.ora 参数 TCP.VALIDNODE_CHECKING, TCP.INVITED_NODES 和 TCP.EXCLUDED_NODES 来控制让哪些客户连接你的用户。

唯一的解决方法：用合适的数据库 grant 或者 Role 语句来保护你的文件安全，任何其他的语句都无法保护你的数据。一般而言，在多个用户中共享数据库帐户是最不可取的做法。你购买的产品应该包括很多安全措施，但你必须用他们来保护自己！

(作者：Dan Norris　译者：April 来源：TT 中国)

怎样访问唯一被授权的用户

问：单一的 Oracle 用户可以用 TNS 服务限制吗？如：sqlplus scott/tiger@ONLYSCOTTALLOWED...

...没有人会用 ONLYSCOTTALLOWED db 连接。如果可能的话，请您告诉我关于这个问题的一种解决方法。

我正在开发 Web 应用程序，想利用这一程序访问唯一的被授权的 Oracle 用户。

答：最好的方法就是通过授权 DB 对象保证你有权使用这些对象，这样通过 Web 应用程序也可以访问它们。你可以拥有 Role 在这些对象上进行必要的选择、删除、更新和执行等等，并将该 Role 分配给 Web 用户。

请注意 SYS 和 SYSTEM 都创建了一些默认用户，在很多情况下这些用户的默认密码并没有改变（如 CTXSYS）。请你确保 DBA 用户的密码不是默认密码，另外你也许能改变默认用户的密码（如 CTXSYS）。如果你坚持单一的 schema 用户（如果你的 Web 应用程序通过 DB Properties 文件连接到数据库），那就需撤销你不需要的其他用户的 session 权限。然而，你需要保证 SYS 和 SYSTEM 的 session 权限没有被撤销。我不知道你是不是可以做到这一点。

(作者：Azmin Famin　译者：April 来源：TT 中国)

如何保证用户安全登录到寄宿的数据库

问：我们需要为我们的一个用户运行 Oracle 开发一个安全系统。然而，要求之一就是我们必须保证这些数据不被我们任何支持用户看见。由于可以密码重新找到，所以对数据加密并不是一种方法。请告诉我是不是 Oracle 对此有其他更好的解决办法。谢谢。

答：如果你只是打算拥有数据库而不是管理它，那么他们能通过拒绝给你帐户不让你访问数据库。如果你必须访问，你可以通过有限的权限创建一个帐户，但是你没有访问他们的数据库表的权限。

但是如果你必须拥有一个 DBA 层权限的帐户，那么你就无法保证他们的数据安全（就我所知）。DBA 权限的帐户基本上可以使用数据，所以他可以访问任何数据。

或许有第三方工具可以提供这一配置，但是很不幸的是，我目前还不知道。

(作者： Karen Morten 译者：April 来源：TT 中国)

如何通过触发器限制用户访问数据库

问：您好！我想在登录数据库之后在通过触发器限制用户数访问据库。你知道，当我们用 SQL*Plus 或 PL/SQL Developer 访问数据库时，程序名称显示在视图 v$session “program”栏中。然而，我有一个终端用户很狡猾，他访问时，在视图里没有显示程序名称，所以他不用通过确认就能正常登录。我不知道这是不是工具版本的原因还是因为机制的设置问题。您能不能给我一些建议，我怎样才能限制这他访问数据库？谢谢您。

答：V$SESSION 中的 program 栏通过应用程序将信息传送给 Oracle。Oracle 并不总是能读应用程序，所以 column 值是 0。要限制你这名用户访问数据库，你可以输入一个简单的 AFTER LOGON 触发器。这一密码的主体部分用于检验 PROGRAM 值是否有效。如果无效，就列举类似下面的例子：

```
SELECT program INTO v_program FROM v$session

WHERE audsid=SYS_CONTEXT('USERENV','SESSIONID');

IF (v_program IS NULL) THEN

   RAISE_APPLICATION_ERROR(-20001,'Not a valid program');

END IF;
```

(作者：Brain Peasland 译者：April 来源：TT 中国)

授予用户仅能访问视图不能访问表的权限

问：我正在寻找解决数据仓储的办法，我只想让用户访问视图。可问题是当他们访问下面的表时，他们通过 Analyser 仍然能够直接访问到视图。我能用什么办法阻止他们访问表呢？

答：授予用户只能访问视图不能访问表的权限，这样做就很安全了，我相信你也想得到这样的结果。即使是不很安全的 ADBMS 也应该支持这一操作。

（作者：Jason Law　译者：April 来源：TT 中国）

如何阻止用户访问数据库

问：有没有一种办法能让 Oracle8.0，8i 以及 9i 阻止某些应用程序如 Microsoft Access 访问数据库？

答：解决这一问题的方法之一就是使用安全应用程序的 Role。创建 Role 的过程如下：

```
CREATE ROLE app_role IDENTIFIED USING app_owner.some_package;
```

你在这一语句中写入 some_package 包并命名登录触发器中的一个程序。这一程序能够执行验证过程，如果它们通过验证，它会在 DBMS_SESSION.SET_ROLE 中设置 app_role。如果没有通过验证,那就表示还没有设置安全应用程序 Role。由于定义了 Role 的范围，使之生效的唯一方法就是运行 some_package 包中的一个程序。如果你所有的权限都在安全应用程序 Role 范围之内（或范围相似的几个 Role 里），那你就不必再阻止用户登录并且还要创建 Session。然而，一旦用其中某个工具登录成功，用户就没有任何应用程序权限，并且只能访问 PUBLIC 授予权限的对象。

(作者：Dan Norris 译者：April 来源：TT 中国)

SQL HTML

我们公司里的开发者要求访问数据库，但是我不想让他们访问。相反，我用 SQL*Plus HTML 的生成功能为开发人员生成 Web 页面，并且调出了他们所需要的信息。我认为这种方法很好。

一般来说，在开始生成脚本时用 Set Markup HTML ON，结束时用 Set Markup HTML OFF，介于这两者之间的脚本就是标准的 SQL 脚本，包括具体的栏的宽度和含有 COLUMN 语句的标题、COMPUTE SUM 总计的数量等等。set markup html on 仅提供 HTML 表，并将 SQL 语句的输出文本置入表格格式中。虽然输出 output 功能看不见，你还是能用 update 功能。当然，你需要避免用到 FEEDBACK, VERIFY 和其他一些类似的语句，因为这些语句会扰乱 HTML output 功能。

我曾经给出过一个例子，摘选的这个例子包括了大量有用的功能。虽然有这个目的，我觉得它和带有 Oracle 8.1.7.4 的 Unix 环境没有什么不同。该应用程序是用来显示数据库在某一特定日期的检查结果，显示出来的一系列的数据通过 CRON 在数据库服务器上聚集。我将下列三个文件也算在内： HTML 文件在 Apache 里脱离 Shell 脚本，Shell 脚本反过来又运行 SQL 脚本。许多 SQL 脚本都暴露在外，似乎是刚从各种动态视图中挑选出来的。通过在 Shell 脚本里生成 SQL 脚本， SQL 脚本就变成了动态的。这些都是你无法想象到的。例如生成一个包括所有数据库的列表，所以用户能够选择他/她想检查的数据库！一个非常简单的例子：

HTML 文件: healthcheck.html

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">

<HTML> <HEAD>

<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
```

```
<META http-equiv="expires" content="Mon, 01 Jan 1970 00:00:00 GMT">

<META NAME="Author" CONTENT="Vasan">

<TITLE>Online Database Statistics</TITLE>

</HEAD>

<BODY BGCOLOR="#ffffff"></body>

<H2> Online Database Statistics </H2>

<TABLE BORDER=10>

<TR>

<TD>

<form method=GET action="http://host/cgi-bin/hcheck.cgi" TARGET="_blank">

<input type=submit value="Database Statistics">

<TD>

<selectname="DBNAME" ALIGN=LEFT>

<option selected value=dbstring0" > Database zero

<option value="dbstring1" > Database one

<option value="dbstring2" > Database two

</select>
```

```
</TD>

</form>

</TD>

</TR>

</TABLE>

</HTML>
```

Shell 脚本：hcheck.cgi

```
#!/bin/ksh

#

#ident "@(#)$Source$ $Revision$"

# The line below is a sample URL that will be generated by the HTML file

# http://<server-name>/cgi-bin/healthcheck.cgi?DBNAME=dbstring1

# The next four lines are actually in a central file which I include in each
shell script

ORAENV_ASK=NO;export ORAENV_ASK

ORACLE_HOME=/opt/9ias/infrastructure;export ORACLE_HOME

PATH=PATH:/usr/local/bin:${ORACLE_HOME}/bin;export PATH

LD_LIBRARY_PATH=$ORACLE_HOME/lib; export LD_LIBRARY_PATH
```

```
TWO_TASK=$(echo ${QUERY_STRING} | sed "s/&DAY.*$//" | sed "s/DBNAME=//")

export TWO_TASK

if [ "${TWO_TASK}" = "dbstring0" -o "${TWO_TASK}" = "dbstring1"  ]

then

   uid="/"

else

   exit 1

fi   # Do not allow the script to proceed if this is dbstring2

echo Content-type: text/plain

echo

TMP_FILE="/tmp/$$.out"

rm -f ${TMP_FILE}

sqlplus ${uid} @healthcheck.sql ${TMP_FILE} > /dev/null

if [ -f ${TMP_FILE} ]

then

   cat ${TMP_FILE}

   rm -f ${TMP_FILE}
```

```
else
   echo "No Records Selected"
fi
rm -f ${TMP_FILE}
exit 0
```

SQL script: healthcheck.sql

```
set markup html on
set feedback off

spool &1
set linesize 200
set pagesize 10000
set heading on
ttitle "Statistics on Buffer Hit Ratio"
select trunc((1-(sum(decode(name,'physical reads',value,0))/
             (sum(decode(name,'db block gets', value,0))+
    (sum(decode(name,'consistent gets',value,0)))))
```

```
              )* 100) "Buffer Hit Ratio"
 from v$sysstat;
column "logical_reads" format 99,999,999,999
column "phys_reads" format 999,999,99999999
column "phy_writes" format 999,999,999
select a.value + b.value "logical_reads",
                c.value  "phys_reads",
                d.value  "phy_writes",

               round(100 * ((a.value+b.value)-c.value) / (a.value+b.value))
from v$sysstat a, v$sysstat b, v$sysstat c, v$sysstat d
where a.name = 'db block gets'
and   b.name = 'consistent gets'
and   c.name = 'physical reads'
and   d.name = 'physical writes';
ttitle "Statistics on Data Dictionary Hit Ratio"
column "Data Dict. Gets"   format 999,999,999
```

```
column "Data Dict. cache misses" format 999,999,999
select sum(gets) "Data Dict. Gets",
       sum(getmisses) "Data Dict. cache misses",
       trunc((1-(sum(getmisses)/sum(gets)))*100)
       "DATA DICT CACHE HIT RATIO"
from v$rowcache;
ttitle "Statistics on Library Cache Miss Ratio"
column "LIBRARY CACHE MISS RATIO" format 99.9999


column "executions" format 999,999,999
column "Cache misses while executing" format 999,999,999
select sum(pins) "executions", sum(reloads) "Cache misses while executing",
                (((sum(reloads)/sum(pins)))) "LIBRARY CACHE MISS RATIO"
from v$librarycache;
ttitle "Statistics on Library Cache"
column "reloads" format 999,999,999
select namespace, trunc(gethitratio * 100) "Hit ratio",
```

```
trunc(pinhitratio * 100) "pin hit ratio", reloads "reloads"

from v$librarycache;

spool off

set markup html off

exit
```

读者反馈：

Mohammed A: Vasan 先生，你这样做似乎是官僚作风。我相信你没有任何理由不让开发人员使用数据库。如果你关心安全和性能，那你就想别的办法吧。1、你的这种想法将扰乱开发人员的工作。放手让他们干吧！2、你的这种政策也只会让他们知道密码而不是他们告诉你。IT 管理的最好的办法之一就是——让专业人员来管理。

Raj P.：这一做法很矛盾：“我们的开发人员需要登录数据库，然而，我不喜欢让他们这样做。”这就像是鱼自由自在地在水里游，开发人员要登录数据库不受任何约束。

Dennis D.：我感觉我必须要防范 Vasan。不懂数据库登录原则的开发人员能将你的数据库弄得一团糟。开发人员需要登录，我们应该给他们登录的帐号、登录密码和重要帐号而不是给 DBA。你不能限制这些帐号，他们能创建临时表并且在短时间内即使不需要也会保留这个表。于是几年之后每个人都害怕弄丢这些彪，因为一些应用程序也许能在某些地方用到。开发人员能够创建反规范化表、没有 primary key 约束表（业务规则暗示要用 primary keys 约束）。他们能够创建带有存储参数的表。书写错误代码一件事，密码能够修改。创建有害的 schema 用户以及存储大量的行更难修改。

(作者：Vasan Srinivasan 译者：April 来源：TT 中国)

用 role/grant 语句和 public synonym

问：我作了如下的查询。我创建了两个用户，U1 和 U2，然后又创建了 role R。我在用户 U1 里创建表并且授予 role R 选择、插入、更新和删除的权限。于是我又将 role R 分配给 U2。现在如果我通过用户 U2 登录，我能够看见用户 U1 的所有对象，还能够通过在 U1 里创建 public synonym 进行简单的程序操作（如我在 U1 里创建了 public synonym 看见了 U2 的所有对象）。

这两种方法有什么不同呢？在功效方面有什么问题吗？如果有，那么这两种方法哪一种更好？

答：赋予 PUBLIC 权限更加简单，但也更加危险。对于数据库安全，有一个惯例就是只将权限赋予给需要它们的用户。此外，你只能让用户执行最小的命令。在今天的 IT 安全意识环境中，授予更多人更多的权限是打开你的数据库大的安全漏洞的途径之一，也值得你在它上面花心思。

用 role 管理安全。这也是他们原先设计的目标。我从来没有授予 PUBLIC 任何权限，也没有理由这么做。

(作者：Brain Peasland 译者：April 来源：TT 中国)

Unix 上的针对不同用户群安全指南

问：我目前正在研究如何最大限度地确保数据库环境安全。我们在 HP-UX.上安装了 9iR2 EE。

在数据库服务器上各种各样的数据库，每个数据库都有它的 DBA 和开发人员。服务器为 Raid 1+0。Oracle 安装在一个文件系统上面，我们想给每个数据库创建一个它们各自的文件系统。我知道它是 Oracle's OFA 转移过来的，但是在这种情况下 Unix 管理和安全更易于使每个数据库包含一个安装点。这样做合理吗，会不会引出一些问题？

对于 Unix 用户、群和安全你有些什么建议？此刻 OS 用户 Oracle 拥有的是软件，他还是 dba 群的用户。那么，其他用户、DBA 和开发人员的安全设置又如何呢？尤其我提到的是特殊的数据库 payroll。我不想其他 Unix 用户也能访问 Oracle 相关文件，SQL、报告等等。一些应用程序也已经有 OS 能鉴别的用户。我们的用户是一群 DBA 用户，但是这个群里 DBA 的数量将会减少。那么在这一设置中 init、log、trace、和.ora 文件会怎么样呢？Oracle 会有所有这些文件吗？

答：如果在不同的环境中有不一样的 DBA 组群，那么你应该为他们每个人创建各自的 ORACLE_HOME。Oracle 二进制的 link time 能识别 OSDBA 和 OSOPER 组群身份。所以如果你是在 ORACLE_HOME 里运行的一个数据库 DBA，那你就是在它里面运行的所有数据库的 DBA。在当前 Oracle 安装方法辨别权限的唯一方法就是将这些 ORACLE_HOME 分开，使它们都有各自的 DBA 组群（进行安装，要求他们填写在 OSDBA 组群的名字，选择一些叙述语，如用 dbapayroll 代替 payroll、用 HR 代替 dbahr 等）。Oracle 帐号也不相同，所以你也将拥有各种不同的软件（如 dbapay、dbahr、dbaerp 等）。有这种一对一映射，你能安全地维护同一硬件和 OS 上环境，并且不会危及到任何应用程序的安全。成本就是为 Oracle 二进制文件按预留较大的空间，但是磁盘很便宜，而安全不便宜。

（作者：Dan Norris 译者：April 来源：TT 中国）

封闭、开放的安全政策和 RBAC role 层级里的 permission 语句

问：你能跟我解释一下为什么在数据库安全方面封闭端的安全政策比开放的安全政策能更好地保护数据呢？你能对 RBAC role 层级里不受限的继承 permissions 这样的假设做出评价吗？

答：要说明为什么封闭的安全政策比开放的要好，首先我先举个例子。假设我的公司已经决定了 Oracle 用户使用 1599 端口（一个非默认端口），并且所有的密码都由 8 个字符和 2 个数字组成。如果在网上发布了这条信息，我现在让任一黑客详细知道那个端口连接到哪个点以及我的数据库密码是怎样组成的。你愿意将这些信息告诉黑客吗？最好还是在你的公司好好保存这些信息。

我并不是 Oracle 领域之外的 Role-Based Access Control 方面的专家。所以我只能告诉你我知道的信息。我不知道用 unconstrainted upward inheritence 是不是一件坏事。例如，可能我拥有 APPL_USER 的 Role，它被分配给了我所有的应用程序用户。但是这个 Role 只能允许用户执行只读 SELECT 语句或者特殊表上的操作任务，所以我还有一个 APPL_ADMIN role 作为应用程序管理员。该 role 通过改变表，它还继承了 APPL_USER role 的功能，所以它还可以读表。这是一种将一个 Role 的所有的权限转到另一 Role 上的简单方法。遗憾的是，APPL_ADMIN Role（被赋予了 APPL_USER Role 的权限）还继承了 APPL_USER role 的所有权限。你不能将 APPL_ADMIN 的某些权限赋予 APPL_USER。这对你的系统来说太强大了。更好的方法是将 APPL_ADMIN 和 APPL_USER 的某些权限加密。这样做就能区分这两种功能。

（作者：Brain Peasland 译者：April 来源：TT 中国）

怎样阻止越权存储到格式程序？

问：我担心我的数据库存在 Oracle Developer 应用程序安全问题。我们有许多系统开发商正在使用这种工具。问题是我们如何才能够阻止越权存储到格式程序？真正的策略就是通过 database roles 选项卡在菜单栏里定义安全范围。利用 Oracle Developer 2000 我们从哪里可以了解到更多的有关安全问题的信息？

答：最佳方法就是在数据库里正确设置安全码。这一方法无论我们用什么应用程序访问数据库，数据总是安全的。我们不该依靠应用程序维护数据安全。如果用户使用不同的应用程序，那么数据库就会丧失安全机制。Oracle在一些文件里有一些很好的建议。以下是关于Oracle安全的一个非常不错的网站：http://www.petefinnigan.com/。

(作者：Brain Peasland 译者：April 来源：TT 中国)

限制用户从其他分公司上访问某些对象

问：我们是一个金融机构，在全国各地都有分公司。我们有一个基于客户服务器的应用程序，该服务器最初用于安装在分公司。后来随由于业务需要，我们决定建立一个集中式的数据中心，所有分公司都通过.net 连接到同一个地址上，并且应用程序和数据库连接到了集中式的数据中心。

为了让所有的分公司都能用到该应用程序，我们在 Citrix 服务器上安装了该程序。在这一程序里，我们有特殊的 Role 文件，这一文件的权限需要授予该应用程序用户。随着授权 Role 文件，在该程序里自动弹出了一些按钮。当分公司使用该应用程序时，它的权限就变小了——只有一个用户有 Role，并且它还不能建立任何安全保证，但是现在不同公司的不同用户连接了同一个应用程序并拥有相同的 Role 是很危险的，因为一个分公司的用户都有权限访问另一个公司的数据。

我们有没有一种能够限制有同一 role 的用户访问某些对象呢？

答：最好的方法就是为每个公司创建不同的 Role 文件，然后合理分配。你还可以通过合并触发器和视图限制用户访问。但是这样做很麻烦。最后，你只需要模拟 Role 的职责。所以为什么不把这些时间花在重新为他们创建 Role 文件呢？

(作者：Brain Peasland 译者：April 来源：TT 中国)

如何限制用户在 SQL*Plus 脚本里执行具体命令？

问：在 Oracle 里有没有一种方法能限制用户在应用层执行一般操作，如 insert、delete、drop 等（也就是用户一查询，就能 SQL*Plus 里限制他，不让他访问数据库）？

答：如果你想阻止用户在 SQL*Plus 里执行具体命令，你可以使用 PRODUCT_USER_PROFILE 表建立你所提到的安全机制。详细资料请参考 SQL*Plus 用户手册和指南附录 E。

（作者：Karen Morton 译者：April 来源：TT 中国）

如何限制用户从 SQL*Plus 访问数据库

问：我有一个 8i 数据库，任何人都能从我的应用程序登录，但是当他们从 SQL*Plus 开始登录时就出现了问题。我用 PUP（product_user_profile）表限制他们用 SQL*Plus 命令登录，这一操作的效果很好，但是现在我发现他们从 ODBC 登录系统！我应该怎么做？

答：如果他们登录的是我的应用程序，那我首先就会使用 Oracle 提供的数据库安全功能（也就是将用户账号分开，有必要的话就使用数据库 roles and grants 和 VPD 功能）。在这种情况下，你不必管他们是如何访问数据库的，因为你的应用程序的所有安全功能都在数据库里面

。

如果这样做还能保证数据库安全（例如，用 COTS 应用程序），你可以使用 login 触发器检查用户使用什么其他程序连接数据库并且使 session 断开连接。

我熟悉的另外一种方法就是创建一个能用密码、拥有所有程序特权的 role 并且它仅使用于你的应用程序。在这种情形下，如果用户以应用程序户身份访问数据库，除非他们激活 role（需要用仅适合应用程序的密码），否则他们就没有权限登录。

(作者：Dan Norris 译者：April 来源：TT 中国)

限制用户从 SQL*Plus 访问 Oracle 对象

许多 Oracle 产品都使用 PRODUCT_USER_PROFILE，SYSTEM 帐户里的表，提供产品层安全来补充 SQL GRANT 产品层安全，并且撤销用户 role。这个表能用于限制用户从 SQL*Plus 访问 Oraclel 对象。我们提到的这个技巧适用于 Oracle 版本 8，8i 和 9i。

一般观点：

在 SQL*Plus 环境里数据库管理员能够用 PRODUCT_USER_PROFILE 使 SQL 和 SQL*Plus 命令无效。SQL*Plus 而不是 Oracle 加强了数据库的安全性能。这样数据库管理员还能够限制用户访问 GRANT, REVOKE 和 SET ROLE 命令、控制用户改变访问数据库权限的能力。

SQL*Plus 从 PRODUCT_USER_PROFILE 语句读取限制范围。用户下次登录到 SQL*Plus 时，PRODUCT_USER_PROFILE 的操作就会生效。

创建表

你可以通过运行带有 SQL 扩展名的 PUPBLD 命令文件，如 SYSTEM 创建 PRODUCT_USER_PROFILE 语句。准确地文件扩展名格式以及文件位置由 SYSTEM 决定。注意 Oracle 设置和用户指南为你或你的 DBA 提供了更多的有关操作系统方面的信息。

表结构

PRODUCT_USER_PROFILE 表由下面这些 column 组成：

```
PRODUCT  NOT NULL CHAR (30)
USERID  CHAR(30)
ATTRIBUTE  CHAR(240)
SCOPE  CHAR(240)
NUMERIC_VALUE  NUMBER(15,2)
CHAR_VALUE  CHAR(240)
DATE_VALUE  DATE
LONG_VALUE  LONG
```

对 Column 的说明和用途：

下面的列表示对 PRODUCT_USER_PROFILE 表里 Column 的说明以及对每个 column 的用法。

Product：必须包括产品名称（在 SQL *Plus 文件）。你不能在这一 column 里输入通配符或 NULL，还必须注意产品名称 SQL*Plus 必须在列入混合文件中，以便经过验证。

Userid：必须包括你想禁用的命令的用户名（在 uppercase 里）。为了使一个以上的用户不执行这个命令，使用 SQL 通配符(%)或输入多条语句。因此，下面的语句无效：

SCOTT
CLASS1
CLASS% （ 所有名字以 CLASS 开头的用户）
% (所有用户)

Attribute：必须包括 SQL、SQL*Plus 或你想禁用的 PL/SQL 命令（如 GET）。如果你禁用了一个 role，那么它必须包括字符串“ROLES”，但是你不能输入一个通配符。参考下面在 SQL 和能禁用的 SQL*Plus 列表中的“Administration”。

Scope：SQL*Plus 不用包括这个列，你只需要输入 NULL。其他产品可以在它里面存储一些数值。

Char_Value：必须包括字符串 DISABLED 用来禁用一个 SQL、SQL*Plus 或者 PL/SQL 命令。如果你正禁用 role，那么它一定要包含你想禁用的 role 的名称。你不能用通配符。

Date_Value：SQL*Plus 不用包含列。你可以在这一列里输入 NULL。其他的产品可在 column 里存储 DATE 值。

Long_Value：SQL*Plus 不包含这一列。你可以在它里面输入 NULL。其他的产品可以在这一列中储存 LONG 数值。

管理

DBA 用户 SYSTEM 拥有 PRODUCT_USER_PROFILE.上的所有权限（当用户 SYSTEM 登录时，SQL*Plus 不执行读 PRODUCT_USER_PROFILE.的任务。因此，对于用户 SYSTEM.没有权县范围的限制）。其他 Oracle 用户名只能通过 SELECT 访问这个表，这样他们就能够了解表对用户名和对 PUBLIC 的限制范围。当 PUPBLD 命令文件运行时，它就允许 SELECT 语句访问 PRODUCT_USER_PROFILE to PUBLIC。

禁用 SQL*Plus、SQL 以及 PL/SQL 命令

如果想让用户禁用 SQL*Plus、SQL 以及 PL/SQL 命令，就只需要在 Userid 栏里插入一行用户名、在 Attribute 列里插入一行命令名、在 Char_Value 栏里插入 DISABLED。Scope、Numeric_Value 以及 Date_Value 栏都应该包含 NULL。如：

PRODUCT USERID ATTRIBUTE CHAR_VALUE

```
--------  ------ ---------- ----------
SQL*Plus  SCOTT  HOST       DISABLED
SQL*Plus  %      INSERT     DISABLED
SQL*Plus  %      UPDATE     DISABLED
SQL*Plus  %      DELETE     DISABLED
```

再运行这些命令时，需要删除含限制范围的行。

你能禁用如下 SQL*Plus 命令：COPY、 EDIT、EXECUTE、EXIT、GET、HOST（或者是 HOST 操作系统的别名，如 VMS 上的$以及 UNIX 上的!）、QUIT、 PASSWORD、RUN、SAVE、SET（查看下面的注释）、 SPOOL 和 START。

注：禁用 SQL*Plus SET 指令的同时也会禁用 SQL SET ROLE 和 SET TRANSACTION 命令。禁用 SQL*Plus START 的同时也会禁用 SQL*Plus @和@@命令。

你同样也可以禁用下面的 SQL 命令：ALTER、ANALYZE、AUDIT、CONNECT、CREATE、DELETE、DROP、GRANT、INSERT、 LOCK、NOAUDIT、RENAME、REVOKE、SELECT、SET ROLE、SET TRANSACTION、TRUNCATE 和 UPDATE。

你还可以禁用 PL/SQL 命令：BEGIN,DECLARE。

注：禁用 BEGIN 和 DECLARE 并不妨碍使用 SQL*Plus EXECUTE 命令。SQL*Plus EXECUTE 必须分开禁用。

(作者：Murali Krishna 译者：April 来源：TT 中国)

如何限制 Oracle 数据库里的特殊 IP 地址登录

问：你有没有想过要限制特殊或一般的 IP 地址能访问你的 Oracle 数据库呢？方法如下：

答：首先，你必须创建一个文件，文件名为 protocol.ora，如下：

```
tcp.invited_nodes=(XXX.XY1.XY2.Y, XXX.XY2.XY3.Z)
tcp.validnode_checking=yes
```

比方说你要允许用户只用 192.168.11.20 or 192.168.10.12 这个 IP 地址登录数据库，那么你仅能用下面的方法来设置 protocol.ora 这个文件。

```
tcp.invited_nodes=(192.168.11.20, 192.168.10.12)
tcp.validnode_checking=yes
```

目标文件如 sqlnet.ora 和 tnsnames.ora.的设置方法也是一样。你需要阻止以及让用户重新设置。

同样，要阻止用另一些 IP 地址登录，必须在 protocol.ora 文件里按照上述方法设置 tcp.excluded_nodes 参量。

注：由于这种方法有些障碍，所以你的程序运行的平台、版本一定要与 Oracle 支持或者 Metalink 相符合。例如，在 Windows NT 和 Oracle 版本 8.1.x 上，必须创建 net8/admin 目录，然后将 protocol.ora 文件置入该目录中，而不是置入 network/admin 目录中。在 Unix 和 Oracle 版本 8.1.x 上，你就必须在 network/admin 目录里重新命名 protocol.ora 文件。

(作者：sameer wadhwa 译者：April 来源：TT 中国)