نگاهی کوتاه به امنیت در اوراکل

اشاره :

اطلاعات صحیح در هر سازمانی بسیار مفید است و باعث موفقیت آن سازمان می‌گردد . اما اگر این اطلاعات دستکاری شود ، بدون شک تأثیرات بدی در سازمان خواهد داشت . بانک اطلاعاتی اوراکل با محدود کردن دسترسی به داده‌ها می‌تواند از اطلاعات شما محافظت کند . اوراکل با استفاده از Granting ، یا محدود کردن مجوزهای کاربران و فراهم کردن دستوراتی مانند Create User،Create Role و Grant برای مدیران پایگاه‌داده می‌تواند دسترسی به اطلاعات را مدیریت کند . هر کاربر اوراکل دارای نام و رمز ورودی است و صاحب چند جدول ، View یا اجزای دیگری است که به وجود می‌آورد . یک Oracle Role (یا نقش) در واقع شامل امتیازاتی است که کاربر می‌تواند برای دسترسی به اشیای بانک اطلاعاتی داشته باشد . می‌توانید امتیازی (Privilege) را به نقشی بدهید و سپس آن نقش را به یک کاربر اهدا کنید . در این مقاله قصد ندارم شما را با مباحث پیچیده امنیت در اوراکل آشنا کنم ، هدف ، آشنا کردن شما به صورت کاملاً عملی با پایه امنیت در اوراکل g10 است .

ایجاد کاربر

هر سیستم اوراکل به صورت پیش‌فرض دارای چندین کاربر مانند SYSTEM و SYS است . کاربر SYS صاحب جداول داخلی بانک اطلاعاتی است (که وظیفه مدیریت بانک اطلاعاتی را به عهده دارند) و کاربر SYSTEM صاحب جداول دیگر مدیریتی است و برای این‌که بتوانیم کاربری را در یک بانک اطلاعاتی اضافه کنیم یا تغییراتی را در مجوزهای آن بدهیم ، باید با این کاربر به سیستم وارد شویم . فرمت دستوری که می‌توانید با آن کاربر جدیدی را به سیستم اضافه کنید، به صورت زیر است :

در شکل 1 مراحل ساخت یک کاربر به نام Amin نشان داده شده است . در مرحله اول کاربر با استفاده از دستور Create User ساخته شده است . در مرحله دوم رمز کاربر تعویض شده است و در مرحله سوم برای این‌که به کاربر اجازه ورود بدهیم ، یک Session جدید به نام او ساخته‌ایم .

```
SQL> create user amin identified by password;          1
User created.
SQL> alter user amin identified by newpassword;        2
User altered.
SQL> grant create session to amin                      3
  2  ;
Grant succeeded.
SQL> |
```

شکل 1

مدیریت رمزها

کاربر بانک اطلاعاتی می‌تواند پسورد خود را با استفاده از دستور password عوض کند ، ولی در اوراکل می‌توانیم رمزهای کاربران را به صورت پیشرفته مدیریت نماییم . مثلاً می‌توانیم رمز ورودی را منقضی کنیم DBA و... می‌تواند پروفایل‌های خاصی را برای مدیریت امنیت تعریف نماید و وقتی کاربر را ایجاد می‌کند ، آن پروفایل را به آن اختصاص دهد .

شکل 2

Profile می‌تواند محدودیت‌هایی از قبیل طول عمر یک رمز ، مدت زمانی که کاربر باید رمز خود را عوض کند ، تعداد دفعات ورود به سیستم با رمز اشتباه برای قفل کردن سیستم ، تعداد روزهایی که حساب کاربر بسته باشد ، تعداد روزهایی که باید بگذرد تا کاربر دوباره از یک رمز استفاده کند ، طول رشته رمز و محدودیت‌های دیگر را در خود داشته باشد .

در شکل 2 طریقه ایجاد یک پروفایل را مشاهده می‌کنید که تعداد اشتباه در ورود روز را «2» تعیین کرده است . همان‌طور که در این کدها مشخص شده است ، این پروفایل به کاربر amin داده می‌شود و حساب او پس از دو اشتباه قفل می گردد . البته DBA می‌تواند با دستور زیر Account او را مجدداً باز کند .

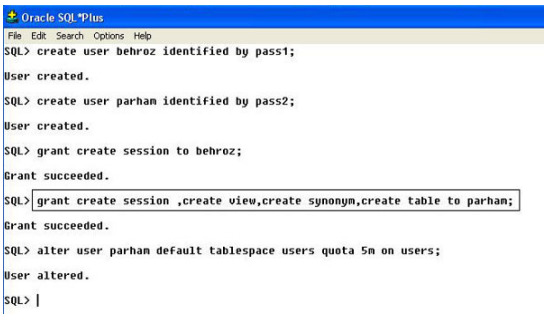
```
Alter user amin account unlock
```

ایجاد نقش برای کاربر

در مراحل قبل کاربر amin را ساختیم و رمز جدیدی به آن دادیم. حال او دارای حساب است ، ولی نمی‌تواند کار خاصی انجام دهد ؛ چراکه هیچ نقش و مجوزی جز Create Session به او Grant نشده است . مهم‌ترین نقش‌های اوراکل برای کاربران در جدول زیر مشاهده می‌شود :

CONNECT RESOURCE DBA	ارتباط و دسترسی به اشیاء و نقش مدیر سیستم
EXP_FULL_DATABASE IMP_FULL_DATABASE	برای Import و Export اطلاعات
DELETE_CATALOG_ROLE EXECUTE_CATALOG_ROLE SELECT_CATALOG_ROLE	برای دسترسی به Data Dictionaryها
AQ_USER_ROLE AQ_ADMINISTRATOR_ROLE	برای استفاده از Advanced Querying
SNMPAGENT	استفاده در Enterprise Manager Intelligent Agent
RECOVERY_CATALOG_OWNER	استفاده در ایجاد Recovery Catalog Schema
SCHDULER_ADMIN	برای زمانبندی Procedureها

حال فرض کنید که کاربر amin صاحب جدول emp و نقش مدیر سیستم را عهده دار است . حال مطابق شکل 3 دو کاربر به اسامی zbehro و Parham می‌سازیم و به هر دوی آن ها اجازه ورود به سیستم را می‌دهیم و به Parham اختیارات دیگری نیز می‌دهیم .



شکل 3

حال سؤال اینجاست که با وجود اختیاراتی که به Parham داده شده است ، آیا او می‌تواند به جداول amin دسترسی کامل داشته باشد یا خیر؟ برای دسترسی کاربری به جدوال خود از دستور grant به نوعی دیگر استفاده می‌کنیم . مثلاً می‌توانیم با دستور زیر بگوییم که کاربر Parham می‌تواند از جدول emp کاربر amin استفاده کند :

```
Grant select on amin.emp to Parham ;
```

اضافه بر نقش‌هایی که به صورت پیش‌فرض در اوراکل وجود دارد ، می‌توانید نقش‌های دیگری نیز در اوراکل درست کنید . مثلاً دو دستور زیر دو نقش جدید به سیستم اضافه می‌کند .

```
Create role modeirkol ;
Create role karmand;
```

حال همان‌طور که در کدهای زیر می‌بینید، می‌توانید امتیازها را به نقش بدهید :

```
Grant select on emp to modeirkol;
Grant create session to karmand;
Grant create session,create view to modeirkol;
```

بررسی عملکرد کاربران

در اوراکل می‌توانیم به راحتی تمامی اعمالی که اتفاق افتاده است را بررسی کنیم . تمامی این اعمال به صورت رکوردهایی در بانک اطلاعاتی ثبت می‌شود . در اوراکل امکان بررسی یا Audit سه چیز وجود دارد : برقراری ارتباط با سیستم ، دسترسی به اشیای بانک اطلاعاتی ، و اعمالی که روی بانک اطلاعاتی انجام می‌گیرد .

برای فعال‌سازی این گزینه ، باید مقدار AUDIT_TRAIL در فایل DB،Init.ora یا OS باشد . برای این‌که ارتباطات کاربران به بانک اطلاعاتی را کنترل کنیم ، می‌توانیم از دستورات زیر استفاده نماییم :

```
audit session;

audit session whenever successful;
audit session whenever not successful;
```

در این حالت سیستم از تمامی ارتباطات موفق و غیرموفق کاربران رکورد برداشت می‌نماید . برای مشاهده این اطلاعات می‌توانیم از جدول dba_aduit_session استفاده نماییم و با استفاده از فیلد returncode که در این جدول است ، خطای مربوطه که معمولاً ORA_1017 و ORA_1005 است را استخراج نمود .

ORA_1005 وقتی است که کاربر بدون کلمه عبور می‌خواهد وارد شود و خطای ORA_1017 زمانی است که کاربر رمز اشتباه را وارد می‌نماید . برای غیر فعال کردن بررسی ارتباطات کاربران ، می‌توانیم از دستور session NOAUDIT استفاده نماییم .

برای بررسی عمل کاربرها روی اشیایی مانند Database Link، Tablespace،User و Index که غالباً Drop، Alter و Create را اجرا می‌کنند ، باید دستور AUDIT ROLE را اجرا کنیم و با جست‌وجوی زیر عملکرد کاربران را مشاهده کرد :

```
Select name,action from audit_actions;
```

اضافه بر کنترل دسترسی بر اشیا ، می‌توانیم اعمال دستکاری داده ها بر اشیا را نیز بررسی نماییم . مثلاً اعمالی مانند Insert، Select و Delete. برای بررسی این قسمت مانند دستورات قبلی از دستور Audit استفاده می‌کنیم ، ولی عبارت اضافی By Session یا Access By نیز به این دستور اضافه می‌شود . این دستور به سیستم می‌گوید :

برای هر دسترسی یک رکورد جمع‌آوری کند . مثلاً برای بررسی اعمالی که روی جدول emp انجام می‌شود ، باید دستورات روبه‌رو را نوشت :

```
SQL> audit insert on scott.emp;
Audit
SQL> audit all on scott.bonus;
Audit succeeded.
SQL> audit delete on scott.dept by session;
Audit succeeded.
```

با این کار هر گونه دستکاری از قبیل اضافه کردن رکورد و حذف آن در جدولDBA_AUDIT_OBJECT ذخیره می‌شوند و شما می‌توانید به راحتی این اطلاعات را مشاهده کنید .