Oracle 最佳安全配置

指导手册

■ 文档编号	RP-2016-01	■ 密级	完全公开
■ 版本编号	V1.0	■ 日期	2017.12.14

目录

一. Oracle 数据库

Oracle 数据库由于其高可靠性和易用性而成为世界上知名数据库之一。它被广泛使用在各个行业。像大多数产品一样，Oracle 安装极其简单。通常，在安装这种产品时安全性不列为主要的考虑因素。

本文旨在帮助用户在快速完成 Oracle 部署后，只需要很少的时间就可以完成 Oracle 的安全配置。这些安全配置可以帮助您抵御大多数常用形式的黑客攻击行为。我们选取现今最流行的 Oracle 12C 作为安全配置演示对象。所有安全配置项都会向大家展示检测方式和修补方法。以帮助您快速判断是否存在配置问题并进行修补。

二. 安全配置项

2.1 安装和升级的安全配置

2.1.1 确保安装并升级到最终版本

建议理由：

数据库升级补丁非常重要，不但可以修复 bug，更可以修复漏洞。防止黑客利用已知漏洞对数据库发动入侵。

检查手段：执行命令行

opatch lsinventory | grep -e "^.*<latest_patch_version_numer>\s*.*$"

和官网比对判断是否是已经安装最新版本补丁

修复建议：

如果不是最新版本，请安装最新版本补丁

2.1.2 确保所有默认密码都已修改

建议理由：

Oracle 有个视图叫做 DBA_USERS_WITH_DEFPWD。其中保留了所有默认账号的密码信息。并且大部分默认密码都是已知的，黑客经常利用已知密码发动攻击。

检查手段：执行如下 sql 语句

SELECT USERNAME FROM DBA_USERS_WITH_DEFPWD WHERE USERNAME NOT LIKE '%XS$NULL%';

如果有返回值说明还存在默认密码的用户，建议修改默认用户的密码。

修复建议：建议手动修改所有默认用户的密码

PASSWORD <username>

2.1.3 确保删除所有样例数据和用户

建议理由：

样例数据和用户被黑客熟知。容易被黑客利用入侵数据库，建议删除样例数据和用户。主要还是用户例如：SCOTT、HR、OE、IX、SH、PM、BI

检查手段：执行如下 sql 语句

SELECT USERNAME FROM ALL_USERS WHERE USERNAME IN ('BI','HR','IX','OE','PM','SCOTT','SH')';

如果有返回值说明还存在样例的用户，建议删除样例用户以及和他们相关的数据。

修复建议：建议手动修改所有默认用户的密码

DROP USER <username> CASCADE;

2.2 监听安全配置

2.2.1 确保监听接收网络类型固定

建议理由：

数据库监听中含有在数据库关键信息，黑客可能通过某些网络方式绕过身份验证对监听的信息进行嗅探。为了防止这种行为，建议限定网络类型。

检查手段：检查配置文件 listener.ora

打开网络配置文件$ORACLE_HOME/network/admin/listener.ora

检查是否存在 SECURE_CONTROL_LISTENER1=TCPS 配置项

修复建议：

建议在所有监听点添加 SECURE_CONTROL_<listener_name>配置防止黑客利用网络嗅探数据库信息。

2.2.2 确保监听文件(listener.ora)中没有 extproc

建议理由：

Extproc 允许数据库程序调用操作系统库，利用操作系统库可以在操作系统上运行任意系统命令。如果黑客入侵数据库成功，可能会对操作系统也同时造成入侵。建议移除 extproc

检查手段：检查配置文件 listener.ora

grep -i extproc $ORACLE_HOME/network/admin/listener.ora

如果存在返回行，建议修复

修复建议：

在监听（listener.ora）文件中删除 extproc 内容

2.2.3 确保监听文件(listener.ora)中 Admin_restrictons 设置成 ON

建议理由：

admin_restrictions_<listener_name>被设置成 ON，可以阻止一般用户对监听文件进行修改。有助于提高监听文件安全性。

检查手段：检查配置文件 listener.ora

grep -i admin_restrictions $ORACLE_HOME/network/admin/listener.ora

如果存在返回行，建议修复

修复建议：

在监听（listener.ora）文件中设置

admin_restrictions_<listener_name> = on

2.2.4 确保监听文件(listener.ora)中 SECURE_REGISTER 设置成 TCPS 或 IPC

建议理由：

SECURE_REGISTER_<listener_name>用于指定链接到 TNS 监听的协议。设定固定协议有助于防止黑客对监听进行嗅探。

检查手段：检查配置文件 listener.ora

grep -i SECURE_REGISTER $ORACLE_HOME/network/admin/listener.ora

如果返回的不是 TCPS 或 IPC，建议修复

修复建议：

在监听（listener.ora）文件中设置

SECURE_REGISTER_<listener_name>=TCPS or

SECURE_REGISTER_<listener_name>=IPC

2.3 数据库安全配置

2.3.1 确保 AUDIT_SYS_OPERATIONS 设置成 ture

建议理由：

AUDIT_SYS_OPERATIONS 如果设置成 false 只会审计 sysdba/sysoper 的启动关闭数据库和登录信息，其他都不会审计。不利于日后追查安全问题。

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME) =
'AUDIT_SYS_OPERATIONS';
```

如果返回的不是 TURE，建议修复

修复建议：通过以下语句修复

```
ALTER SYSTEM SET AUDIT_SYS_OPERATIONS = TRUE SCOPE=SPFILE;
```

2.3.2 确保 Audit_trail 设置成‘OS’,’DB,EXTENDED’或‘XML,EXTENDED’

建议理由：

audit_trail 必须启用，建议按照实际需求设置成如果设置成‘OS’,’DB,EXTENDED’或‘XML,EXTENDED’。开启后有利于日后追查安全问题。

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME) =
'AUDIT_TRAIL';
```

如果返回的不是 TURE，建议修复

修复建议：通过以下语句修复

```
ALTER SYSTEM SET AUDIT_TRAIL = DB, EXTENDED SCOPE = SPFILE;
ALTER SYSTEM SET AUDIT_TRAIL = OS SCOPE = SPFILE;
ALTER SYSTEM SET AUDIT_TRAIL = XML, EXTENDED SCOPE = SPFILE;
ALTER SYSTEM SET AUDIT_TRAIL = DB SCOPE = SPFILE;
ALTER SYSTEM SET AUDIT_TRAIL = XML SCOPE = SPFILE;
```

2.3.3 确保 global_names 设置成 TRUE’

建议理由：

global_names 开启后会要求进行 DATABASE LINK 的两个数据库实例 global_names 一致，否则链接失败。这样可以提高安全性防止未经授权的链接。

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME) =
'GLOBAL_NAMES';
```

如果返回的不是 TURE，建议修复

修复建议：通过以下语句修复

```
ALTER SYSTEM SET GLOBAL_NAMES = TRUE SCOPE = SPFILE;
```

2.3.4 确保 local_listener 设置恰当

建议理由：

local_listener 设置指定一个网络名称可以有效的防止 TNS 劫持的攻击。TNS 劫持攻击可以盗取客户端到数据库的所有数据流中的数据。

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME) =
'LOCAL_LISTENER';
```

如果返回的不是类似

(DESCRIPTION=(ADDRESS= (PROTOCOL=IPC)(KEY=REGISTER))).，建议修复

修复建议：通过以下语句修复

```
ALTER SYSTEM SET
LOCAL_LISTENER='(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=RE
GISTER)))' SCOPE=BOTH;
```

2.3.5 确保 O7_dictionary_accessibility 设置 FALSE

建议理由：

O7_dictionary_accessibility 一个数据库初始化参数。开启后允许 EXECUTE ANY PROCEDURE 和 SELECT ANY DICTIONARY 访问 SYS 模式中的对象；此功能是为了便于从 Oracle 7 数据库迁移到更高版本而创建的。但同时也为黑客打开了便利之门，提高了数据库的安全风险。建议关闭。

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE
UPPER(NAME)='O7_DICTIONARY_ACCESSIBILITY';
```

如果返值是 false 则安全，否则建议修复

修复建议：通过以下语句修复

```
ALTER SYSTEM SET O7_DICTIONARY_ACCESSIBILITY=FALSE SCOPE =
SPFILE;
```

2.3.6 确保 os_roles 设置 FALSE

建议理由：

os_roles 开启后允许将外部创建的组应用于数据库管理。通常此举会消弱数据库安全等级。建议关闭

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='
OS_ROLES';
```

如果返值是 false 则安全，否则建议修复

修复建议：通过以下语句修复

```
ALTER SYSTEM SET OS_ROLES = FALSE SCOPE = SPFILE;
```

2.3.7 确保 remote_listener 设置为空

建议理由：

remote_listener 允许监听链接到数据库实例可能存在潜在欺骗链接，会影响数据库的的安全性。建议该项设置为空建议

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='
REMOTE_LISTENER';
```

如果返值是 false 则安全，否则建议修复

修复建议：通过以下语句修复

```
ALTER SYSTEM SET REMOTE_LISTENER = '' SCOPE = SPFILE;
```

2.3.8 确保 remote_login_passwordfile 设置为 none

建议理由：

remote_login_passwordfile 是用来指定多少个 oracle 实例可以使用密码文件登录数据库。使用密码文件登录数据库，会把密码以某种形式记录在文件中，并不安全。建议该项设置为 none 注意 none 表示禁止使用密码文件。EXCLUSIVE 表示只有一个数据库实例可以使用此密码文件。SHARED 表示可以有多个数据库实例使用此密码文件。

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='
REMOTE_LOGIN_PASSWORDFILE';
```

如果返值是 none 则安全，否则建议修复

修复建议：通过以下语句修复

```
ALTER SYSTEM SET REMOTE_LOGIN_PASSWORDFILE = 'NONE' SCOPE =
SPFILE;
```

2.3.9 确保 remote_os_authent 设置为 false

建议理由：

remote_os_authent 设置成 ON，表示允许远程客户端使用操作系统用户来登录数据库。使用操作系统账号登录数据库会降低数据库安全。

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='
REMOTE_OS_AUTHENT';
```

如果返值是 false 安全，否则建议修复

修复建议：通过以下语句修复

```
ALTER SYSTEM SET REMOTE_OS_AUTHENT = FALSE SCOPE = SPFILE;
```

2.3.10 确保 remote_os_roles 设置为 false

建议理由：

remote_os_roles 设置成 ON，表示允许以操作系统用户来登录数据库。使用操作系统账号登录数据库会降低数据库安全。

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='
REMOTE_OS_ROLES';
```

如果返值是 false 安全，否则建议修复

修复建议：通过以下语句修复

```
ALTER SYSTEM SET REMOTE_OS_ROLES = FALSE SCOPE = SPFILE;
```

2.3.11 确保 utl_file_dir 设置为空

建议理由：

utl_file_dir 设置的路径的文件允许 utl_file 包对该文件进行读取/写入/修改/删除。

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='
UTL_FILE_DIR';
```

如果返值是 false 安全，否则建议修复

修复建议：通过以下语句修复

```
ALTER SYSTEM SET UTL_FILE_DIR = '' SCOPE = SPFILE;
```

2.3.12 确保 SEC_CASE_SENSITIVE_LOGON 设置为 TRUE

建议理由：

SEC_CASE_SENSITIVE_LOGON 信息决定在登录时密码是否需要区分大小写。密码开启大小写会有助于提高数据库密码安全性。尤其次设置可以减小 CVE-2012-3137 的威胁。

检查手段：通过 sql 检查

```
SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='
SEC_CASE_SENSITIVE_LOGON';
```

如果返值是 true 安全，否则建议修复

修复建议：通过以下语句修复

ALTER SYSTEM SET SEC_CASE_SENSITIVE_LOGON = TRUE SCOPE = SPFILE;

2.3.13 确保 SEC_MAX_FAILED_LOGIN_ATTEMPTS 设置为适当值（10）

建议理由：

SEC_MAX_FAILED_LOGIN_ATTEMPTS 决定失败登录次数，后关闭登录，锁定账号。具体数据建议按照安全程度来设置。建议使用默认 10 次，也可以使用认为合适的值。

检查手段：通过 sql 检查

SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='SEC_MAX_FAILED_LOGIN_ATTEMPTS';

如果返值是 10 或符合预设，否则建议修复

修复建议：通过以下语句修复

ALTER SYSTEM SET SEC_MAX_FAILED_LOGIN_ATTEMPTS = 10 SCOPE = SPFILE;

2.3.14 确保 SEC_PROTOCOL_ERROR_FURTHER_ACTION 设置为‘DELAY,3’或‘DROP,3’

建议理由：

SEC_PROTOCOL_ERROR_FURTHER_ACTION 决定对畸形包响应。响应分为三种 CONTINUE、DELAY 和 DROP。第一种继续可能会允许恶意语句进入，我们建议选择后两种。同时最好设定延迟时间参数，防止被恶意占据资源。

检查手段：通过 sql 检查

SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='SEC_PROTOCOL_ERROR_FURTHER_ACTION ';

如果返值是 DELAY,3’ 或‘DROP,3，否则建议修复

修复建议：通过以下语句修复（两选一）

ALTER SYSTEM SET SEC_PROTOCOL_ERROR_FURTHER_ACTION = 'DELAY,3' SCOPE = SPFILE;

ALTER SYSTEM SET SEC_PROTOCOL_ERROR_FURTHER_ACTION = 'DROP,3' SCOPE = SPFILE;

2.3.15 确保 SEC_PROTOCOL_ERROR_TRACE_ACTION 设置为‘log’

建议理由：

SEC_PROTOCOL_ERROR_TRACE_ACTION 设置有四种模式，是用来记录畸形包的。建议使用 log 模式。既保证性能又能对畸形包的情况进行必要信息的记录。对日后追查安全问题很有好处。

检查手段：通过 sql 检查

SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='SEC_PROTOCOL_ERROR_TRACE_ACTION ';

如果返值是 log 安全，否则建议修复

修复建议：通过以下语句修复

ALTER SYSTEM SET SEC_PROTOCOL_ERROR_TRACE_ACTION=LOG SCOPE = SPFILE;

2.3.16 确保 SEC_RETURN_SERVER_RELEASE_BANNER 设置为‘false’

建议理由：

SEC_RETURN_SERVER_RELEASE_BANNER 如果设置成 On ,则会允许数据库返回补丁版本信息。黑客可能借此知道数据库存在哪些安全漏洞，实施攻击。

检查手段：通过 sql 检查

SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='SEC_RETURN_SERVER_RELEASE_BANNER ';

如果返值是 false 安全,否则建议修复。

修复建议：通过以下语句修复

ALTER SYSTEM SET SEC_RETURN_SERVER_RELEASE_BANNER = FALSE SCOPE = SPFILE;

2.3.17 确保 SQL92_SECURITY 设置为’TRUE’

建议理由：

SQL92_SECURITY 开启会要求 where 或 set 字句中 update 或 delete 操作之间必须被授予 select 权限。如果没有这个防护，黑客可能会通过尝试猜测的方式，猜出表的数据结构甚至内容，导致数据泄露。

检查手段：通过 sql 检查

SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='SQL92_SECURITY ';

如果返值是 true 安全,否则建议修复。

修复建议：通过以下语句修复

ALTER SYSTEM SET SQL92_SECURITY = TRUE SCOPE = SPFILE;

2.3.18 确保_trace_files_public 设置为'FALSE'

建议理由：

_trace_files_public 开启，任意用户可以读取跟踪文件。跟踪文件中很可能记录了一些敏感信息。此举可能导致敏感信息外泄。建议不启用。

检查手段：通过 sql 检查

SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='_trace_files_public ';

如果返值是 false 安全,否则建议修复。

修复建议：通过以下语句修复

ALTER SYSTEM SET "_trace_files_public" = FALSE SCOPE = SPFILE;

2.3.19 确保 RESOURCE_LIMIT 设置为'true'

建议理由：

RESOURCE_LIMIT 不开启，不会执行数据库配置文件中的系统限制，这可能会在某些异常或恶意程序的影响下数据库过度使用系统资源，最后导致系统 CPU 用尽。建议开启可以避免此类数据库问题对系统的影响。

检查手段：通过 sql 检查

SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='RESOURCE_LIMIT ';

如果返值是 false 安全,否则建议修复。

修复建议：通过以下语句修复

ALTER SYSTEM SET RESOURCE_LIMIT = TRUE SCOPE = SPFILE;

2.4 链接和登陆安全配置

2.4.1 确保 failed_login_attempts 设置小于等于 5

建议理由：

failed_login_attempts 决定失败登录次数。建议设置小于等于 5 次后锁定账号。因为连续错 5 次已经可以认为是有人在对数据库进行暴力破解。

检查手段：通过 sql 检查

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='FAILED_LOGIN_ATTEMPTS'
AND
(
    LIMIT = 'DEFAULT'
    OR LIMIT = 'UNLIMITED'
    OR LIMIT > 5
);
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
ALTER PROFILE <profile_name> LIMIT FAILED_LOGIN_ATTEMPTS 5;
```

2.4.2 确保 PASSWORD_LOCK_TIM 设置大于等于 1

建议理由：

PASSWORD_LOCK_TIME 决定多次失败登录锁账号后过多少天才能解锁账号。建议设置大于等于 1 天

检查手段：通过 sql 检查

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='PASSWORD_LOCK_TIME'
AND
(
LIMIT = 'DEFAULT'
OR LIMIT = 'UNLIMITED'
OR LIMIT < 1
);
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
ALTER PROFILE <profile_name> LIMIT PASSWORD_LOCK_TIME 1;
```

2.4.3 确保 password_life_time 设置小于等于 90

建议理由：

password_life_time 决定密码过期时间。建议设置时间小于等于 90 天。90 天内数据库现有的密码强度是很难通过暴力破解算出密码的。

检查手段：通过 sql 检查

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
```

```
WHERE RESOURCE_NAME='PASSWORD_LIFE_TIME'
AND
(
    LIMIT = 'DEFAULT'
    OR LIMIT = 'UNLIMITED'
    OR LIMIT > 90
);
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
ALTER PROFILE <profile_name> LIMIT PASSWORD_LIFE_TIME 90;
```

2.4.4 确保 password_reuse_max 设置大于等于 20

建议理由：

password_reuse_max 决定重用密码之间的间隔。例如原来用过密码 A，如果要在用密码 A 就必须在密码 A 之后换过 N 个密码才能再用密码 A。从经验来看，建议设置间隔大于等于 20 个。

检查手段：通过 sql 检查

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='PASSWORD_REUSE_MAX'
AND
(
    LIMIT = 'DEFAULT'
    OR LIMIT = 'UNLIMITED'
    OR LIMIT < 20
);
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
ALTER PROFILE <profile_name> LIMIT PASSWORD_REUSE_MAX 20;
```

2.4.5 确保 password_reuse_time 设置大于等于 365

建议理由：

password_reuse_time 决定相同密码之间的时间间隔。例如原来用过密码 A，如果要再用密码 A 就必须间隔 N 天。从经验来看，建议设置间隔大于等于 1 年（365 天）。

检查手段：通过 sql 检查

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='PASSWORD_REUSE_TIME'
```

```
AND
(
    LIMIT = 'DEFAULT'
    OR LIMIT = 'UNLIMITED'
    OR LIMIT < 365
);
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
ALTER PROFILE <profile_name> LIMIT PASSWORD_REUSE_TIME 365;
```

2.4.6 确保 password_grace_time 设置小于等于 5

建议理由：

password_grace_time 决定用户登陆权限被锁之前与用户密码到期之后的时间间隔。建议不要长于 5 天

检查手段：通过 sql 检查

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='PASSWORD_GRACE_TIME'
AND
(
    LIMIT = 'DEFAULT'
    OR LIMIT = 'UNLIMITED'
    OR LIMIT > 5
);
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
ALTER PROFILE <profile_name> LIMIT PASSWORD_GRACE_TIME 5;
```

2.4.7 确保 password_verify_function 在策略中开启

建议理由：

password_verify_function 决定密码的复杂度要求。本身是一个脚本需要额外执行脚本才能开启。开启后数据库用户在 sql 端就无法设置简单密码了

检查手段：通过 sql 检查

```
SELECT PROFILE, RESOURCE_NAME
FROM DBA_PROFILES
WHERE RESOURCE_NAME='PASSWORD_VERIFY_FUNCTION'
AND (LIMIT = 'DEFAULT' OR LIMIT = 'NULL');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句开启复杂密码策略

$ORACLE_HOME/rdbms/admin/utlpwdmg.sql

2.4.8 确保 SESSIONS_PER_USER 设置小于等于 10

建议理由：

SESSIONS_PER_USER 决定允许同时打开的最大用户会话数量。限制这个数量有助于防止因为过量会话资源占满，导致的拒绝服务。

检查手段：通过 sql 检查

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='SESSIONS_PER_USER'
AND
(
    LIMIT = 'DEFAULT'
    OR LIMIT = 'UNLIMITED'
    OR LIMIT > 10
);
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
ALTER PROFILE <profile_name> LIMIT SESSIONS_PER_USER 10;
```

2.4.9 确保没有非系统用户使用默认配置

建议理由：

建议使用适当策略的配置文件创建用户。默认配置文件经常随数据库升级变化，且缺乏足够的限制。建议自定义用户不要使用默认配置。

检查手段：通过 sql 检查

```
SELECT USERNAME FROM DBA_USERS WHERE PROFILE='DEFAULT'
AND ACCOUNT_STATUS='OPEN' AND USERNAME NOT IN ('ANONYMOUS',
'CTXSYS', 'DBSNMP', 'EXFSYS', 'LBACSYS', 'MDSYS',
'MGMT_VIEW','OLAPSYS','OWBSYS', 'ORDPLUGINS','ORDSYS', 'OUTLN',
'SI_INFORMTN_SCHEMA','SYS','SYSMAN', 'SYSTEM', 'TSMSYS', 'WK_TEST',
'WKSYS','WKPROXY', 'WMSYS', 'XDB', 'CISSCAN');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
ALTER USER <username> PROFILE <appropriate_profile>
```

2.5 角色权限安全配置

Public 是 Oracle 数据库的一个重要角色。所有用户都具有这个角色的权限。所以这个角色的权限必须受到严格控制，否则很可能导致越权攻击 或透过数据库攻击操作系统的情况发生

2.5.1 确保 public 角色没有执行 DBMS_ADVISOR 的权限

建议理由：

DBMS_ADVISOR 可以允许未经授权的用户，访问本地操作系统文件。黑客可能利用该包，通过数据库对操作系统发动攻击。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME='DBMS_ADVISOR';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_ADVISOR FROM PUBLIC;
```

2.5.2 确保 public 角色没有执行 DBMS_CRYPTO 的权限

建议理由：

DBMS_ADVISOR 用于确定加密应用程序数据的加密算法的强度。过程中会有借用 SYS 权限的地方。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND TABLE_NAME='DBMS_CRYPTO';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_CRYPTO FROM PUBLIC;
```

2.5.3 确保 public 角色没有执行 DBMS_JAVA 的权限

建议理由：

DBMS_JAVA 包可能被用于通过数据库攻击操作系统。不建议 public 角色有该包的执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME='DBMS_JAVA';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_JAVA FROM PUBLIC;
```

2.5.4 确保 public 角色没有执行 DBMS_JAVA_TEST 的权限

建议理由：

DBMS_JAVA_TEST 包可能被用于通过数据库攻击操作系统。不建议 public 角色有该包的执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME='DBMS_JAVA_TEST';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_JAVA_TEST FROM PUBLIC;
```

2.5.5 确保 public 角色没有执行 DBMS_JOB 的权限

建议理由：

DBMS_JOB 包允许未经授权的用户禁用或超载作业队列。可能会引起一系列的安全问题。不建议 public 角色有该包的执行权限。并且该包已经被取代。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_JOB';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_JOB FROM PUBLIC;
```

2.5.6 确保 public 角色没有执行 DBMS_LDAP 的权限

建议理由：

DBMS_LDAP 包可以通过 DNS 将信息向外发送，引起数据泄露。所以不建议 public 角色有该包的执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_LDAP';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_LDAP FROM PUBLIC;
```

2.5.7 确保 public 角色没有执行 DBMS_LOB 的权限

建议理由：

DBMS_LOB 包允许操作实例上的 BLOB，CLOB，NCLOB，BFILE 和临时 LOB，可能会销毁数据。所以不建议 public 角色有该包的执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_LOB';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_LOB FROM PUBLIC;
```

2.5.8 确保 public 角色没有执行 DBMS_OBFUSCATION_TOOLKIT 的权限

建议理由：

DBMS_OBFUSCATION_TOOLKIT 用于确定加密应用程序数据的加密算法的强度。过程中会有借用 SYS 权限的地方。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME='
DBMS_OBFUSCATION_TOOLKIT';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_OBFUSCATION_TOOLKIT FROM PUBLIC;
```

2.5.9 确保 public 角色没有执行 DBMS_RANDOM 的权限

建议理由：

DBMS_RANDOM 用生成随机数，可能被不合理使用而导致安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_RANDOM';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_RANDOM FROM PUBLIC;
```

2.5.10 确保 public 角色没有执行 DBMS_SCHEDULER 的权限

建议理由：

DBMS_SCHEDULER 用于运行数据库操作任务。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_SCHEDULER';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_SCHEDULER FROM PUBLIC;
```

2.5.11 确保 public 角色没有执行 DBMS_SQL 的权限

建议理由：

DBMS_SQL 用于运行 sql 语句，会被黑客用来实施提权操作。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_SQL';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_SQL FROM PUBLIC;
```

2.5.12 确保 public 角色没有执行 DBMS_XMLGEN 的权限

建议理由：

DBMS_XMLGEN 可以被用于搜素数据库中的敏感信息。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_XMLGEN';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_XMLGEN FROM PUBLIC;
```

2.5.13 确保 public 角色没有执行 DBMS_XMLQUERY 的权限

建议理由：

DBMS_XMLQUERY 可以被用于搜素数据库中的敏感信息。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_XMLQUERY';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_XMLQUERY FROM PUBLIC;
```

2.5.14 确保 public 角色没有执行 UTL_FILE 的权限

建议理由：

UTL_FILE 可以读取数据库所在操作系统上的文件，从中盗取敏感信息。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' UTL_FILE';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON UTL_FILE FROM PUBLIC;
```

2.5.15 确保 public 角色没有执行 UTL_INADDR 的权限

建议理由：

UTL_INADDR 可以将获取的数据通过 DNS 向外传输。多被 sql 注入所利用。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' UTL_INADDR';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON UTL_INADDR FROM PUBLIC;
```

2.5.16 确保 public 角色没有执行 UTL_TCP 的权限

建议理由：

UTL_TCP 可以用来读写文件到网络层中。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
```

AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' UTL_TCP';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON UTL_TCP FROM PUBLIC;

2.5.17 确保 public 角色没有执行 UTL_MAIL 的权限

建议理由：

UTL_MAIL 可以用破坏 SMTP 服务甚至导致拒绝服务。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'

AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' UTL_MAIL';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON UTL_MAIL FROM PUBLIC;

2.5.18 确保 public 角色没有执行 UTL_SMTP 的权限

建议理由：

UTL_SMTP 可以用破坏 SMTP 服务甚至导致拒绝服务。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'

AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' UTL_SMTP';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON UTL_SMTP FROM PUBLIC;

2.5.19 确保 public 角色没有执行 UTL_DBWS 的权限

建议理由：

UTL_DBWS 可以用于将文件读写入基于 Web 的文件中或者破坏 HTTP 的传输。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'

AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' UTL_DBWS';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON UTL_DBWS FROM PUBLIC;

2.5.20 确保 public 角色没有执行 UTL_ORAMTS 的权限

建议理由：

UTL_ORAMTS 用于执行 http 请求。可以向外部发送明感信息造成安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' UTL_ORAMTS';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON UTL_ORAMTS FROM PUBLIC;
```

2.5.21 确保 public 角色没有执行 UTL_HTTP 的权限

建议理由：

UTL_HTTP 用于执行 http 请求。可以向外部发送明感信息造成安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' UTL_HTTP';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON UTL_HTTP FROM PUBLIC;
```

2.5.22 确保 public 角色没有执行 HTTPURITYPE 的权限

建议理由：

HTTPURITYPE 用于执行 http 请求。可以向外部发送明感信息造成安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' HTTPURITYPE';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON HTTPURITYPE FROM PUBLIC;
```

2.5.23 确保 public 角色没有执行 DBMS_SYS_SQL 的权限

建议理由：

DBMS_SYS_SQL 包可以执行任意 sql 语句，可能被低权限账号利用这点实施越权攻击。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_SYS_SQL';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_SYS_SQL FROM PUBLIC;
```

2.5.24 确保 public 角色没有执行 DBMS_BACKUP_RESTORE 的权限

建议理由：

DBMS_BACKUP_RESTORE 包可以执行任意 sql 语句，可能被低权限账号利用这点实施越权攻击。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME='
DBMS_BACKUP_RESTORE';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_BACKUP_RESTORE FROM PUBLIC;
```

2.5.25 确保 public 角色没有执行 DBMS_AQADM_SYSCALLS 的权限

建议理由：

DBMS_AQADM_SYSCALLS 包可以运行某些 sql 语句，可能被低权限账号利用这点实施越权攻击。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME='
DBMS_AQADM_SYSCALLS';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_AQADM_SYSCALLS FROM PUBLIC;
```

2.5.26 确保 public 角色没有执行 DBMS_REPCAT_SQL_UTL 的权限

建议理由：

DBMS_REPCAT_SQL_UTL 包可能允许低权限用户以 sys 用户身份执行 SQL 命令。容易出现安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME='
DBMS_REPCAT_SQL_UTL';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON DBMS_REPCAT_SQL_UTL FROM PUBLIC;
```

2.5.27 确保 public 角色没有执行 INITJVMAUX 的权限

建议理由：

INITJVMAUX 包可能允许低权限用户以 sys 用户身份执行 SQL 命令。容易出现安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' INITJVMAUX';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ON INITJVMAUX FROM PUBLIC;
```

2.5.28 确保 public 角色没有执行 DBMS_STREAMS_ADM_UTL 的权限

建议理由：

DBMS_STREAMS_ADM_UTL 包可能允许低权限用户以 sys 用户身份执行 SQL 命令。容易出现安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

```
SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME='
```

DBMS_STREAMS_ADM_UTL';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON DBMS_STREAMS_ADM_UTL FROM PUBLIC;

2.5.29 确保 public 角色没有执行 DBMS_AQADM_SYS 的权限

建议理由：

DBMS_AQADM_SYS 包可能允许低权限用户以 sys 用户身份执行 SQL 命令。容易出现安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'

AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_AQADM_SYS';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON DBMS_AQADM_SYS FROM PUBLIC;

2.5.30 确保 public 角色没有执行 DBMS_STREAMS_RPC 的权限

建议理由：

DBMS_STREAMS_RPC 包可能允许低权限用户以 sys 用户身份执行 SQL 命令。容易出现安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'

AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_STREAMS_RPC';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON DBMS_STREAMS_RPC FROM PUBLIC;

2.5.31 确保 public 角色没有执行 DBMS_PRVTAQIM 的权限

建议理由：

DBMS_PRVTAQIM 包可能允许低权限用户以 sys 用户身份执行 SQL 命令。容易出现安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_PRVTAQIM';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON DBMS_PRVTAQIM FROM PUBLIC;

2.5.32 确保 public 角色没有执行 LTADM 的权限

建议理由：

LTADM 包可能允许低权限用户以 sys 用户身份执行 SQL 命令。容易出现安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' LTADM';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON LTADM FROM PUBLIC;

2.5.33 确保 public 角色没有执行 WWV_DBMS_SQL 的权限

建议理由：

WWV_DBMS_SQL 包可能允许低权限用户以 Application Express（APEX）用户权限执行 SQL 命令。容易出现安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' WWV_DBMS_SQL';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON WWV_DBMS_SQL FROM PUBLIC;

2.5.34 确保 public 角色没有执行 WWV_EXECUTE_IMMEDIATE 的权限

建议理由：

WWV_EXECUTE_IMMEDIATE 包可能允许低权限用户以 Application Express（APEX）用户权限执行 SQL 命令。容易出现安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME='
WWV_EXECUTE_IMMEDIATE';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON WWV_EXECUTE_IMMEDIATE FROM PUBLIC;

2.5.35 确保 public 角色没有执行 DBMS_IJOB 的权限

建议理由：

DBMS_IJOB 包可能允许低权限用户以作业中的用户权限执行 sql 命令。容易出现安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME=' DBMS_IJOB';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON DBMS_IJOB FROM PUBLIC;

2.5.36 确保 public 角色没有执行 DBMS_FILE_TRANSFER 的权限

建议理由：

DBMS_FILE_TRANSFER 包允许从一个数据库传输文件到另一个数据库中。有可能会出现安全问题。不适合被 public 角色调用。建议删除执行权限。

检查手段：通过 sql 检查

SELECT PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE' AND TABLE_NAME='
DBMS_FILE_TRANSFER';

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE ON DBMS_FILE_TRANSFER FROM PUBLIC;

2.6 权限安全配置（4.3）

2.6.1 确保非特定系统用户或角色不被授予 SELECT ANY DICTIONARY 权限

建议理由：

SELECT ANY DICTIONARY 权限允许用户访问 sys 对象。可以借此获得数据库密码哈希值。可能会给暴力破解密码带来可能性。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='SELECT ANY DICTIONARY'
AND GRANTEE NOT IN ('DBA','DBSNMP','OEM_MONITOR',
'OLAPSYS','ORACLE_OCM','SYSMAN','WMSYS','SYSBACKUP','SYSDG
');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE SELECT_ANY_DICTIONARY FROM <grantee>;
```

2.6.2 确保非特定系统用户或角色不被授予 SELECT ANY TABLE 权限

建议理由：

SELECT ANY TABLE 权限允许用户访问除去 SYS 表外的任意表。可能会从中盗取敏感数据。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='SELECT ANY TABLE'
AND GRANTEE NOT IN ('DBA', 'MDSYS', 'SYS', 'IMP_FULL_DATABASE',
'EXP_FULL_DATABASE',
'DATAPUMP_IMP_FULL_DATABASE','WMSYS','SYSTEM','OLAP_DBA','
DV_REALM_OWNER');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE SELECT ANY TABLE FROM <grantee>;

2.6.3 确保非特定系统用户或角色不被授予 AUDIT SYSTEM 权限

建议理由：

AUDIT SYSTEM 权限允许用户访改变审计策略。导致某些恶意行为被有目的的掩盖。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='AUDIT SYSTEM'
AND GRANTEE NOT IN
('DBA','DATAPUMP_IMP_FULL_DATABASE','IMP_FULL_DATABASE',
'SYS','AUDIT_ADMIN');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE AUDIT SYSTEM FROM <grantee>;
```

2.6.4 确保非特定系统用户或角色不被授予 EXEMPT ACCESS POLICY 权限

建议理由：

EXEMPT ACCESS POLICY 权限允许用户访问机密数据甚至更改。可能造成数据泄露的问题。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='EXEMPT ACCESS POLICY';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXEMPT ACCESS POLICY FROM <grantee>;
```

2.6.5 确保非特定系统用户或角色不被授予 BECOME USER 权限

建议理由：

BECOME USER 权限允许用户访未经授权使用其他用户权限。可能造成安全问题。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='BECOME USER'
AND GRANTEE NOT IN ('DBA','SYS','IMP_FULL_DATABASE');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE BECOME USER FROM <grantee>;
```

2.6.6 确保非特定系统用户或角色不被授予 CREATE PROCEDURE 权限

建议理由：

CREATE PROCEDURE 权限允许用户以调用者权限创建存储过程，在一定条件下可能造成安全问题。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='CREATE PROCEDURE'
AND GRANTEE NOT IN
( 'DBA','DBSNMP','MDSYS','OLAPSYS','OWB$CLIENT',
'OWBSYS','RECOVERY_CATALOG_OWNER','SPATIAL_CSW_ADMIN_USR',
'SPATIAL_WFS_ADMIN_USR','SYS','APEX_030200','APEX_040000',
'APEX_040100','APEX_040200','DVF','RESOURCE','DV_REALM_RESOURCE',
'APEX_GRANTS_FOR_NEW_USERS_ROLE','APEX_050000','MGMT_VIEW',
'SYSMAN_MDS','SYSMAN_OPSS','SYSMAN_RO','SYSMAN_STB');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE CREATE_PROCEDURE FROM <grantee>;
```

2.6.7 确保非特定系统用户或角色不被授予 ALTER SYSTEM 权限

建议理由：

ALTER SYSTEM 权限允许用户改变数据库运行状态，在一定条件下可能造成安全问题。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='ALTER SYSTEM'
AND GRANTEE NOT IN
('SYS','SYSTEM','APEX_030200','APEX_040000',
'APEX_040100','APEX_040200','DBA','EM_EXPRESS_ALL','SYSBACKUP','GSMADMIN_ROLE',
'GSM_INTERNAL','SYSDG','GSMADMIN_INTERNAL');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE ALTER SYSTEM FROM <grantee>;
```

2.6.8 确保非特定系统用户或角色不被授予 CREATE ANY LIBRARY 权限

建议理由：

CREATE ANY LIBRARY 权限允许用户创建和共享库多个关联对象，在一定条件下可能造成安全问题。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='CREATE ANY LIBRARY'
AND GRANTEE NOT IN
('SYS','SYSTEM','DBA','IMP_FULL_DATABASE');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE CREATE ANY LIBRARY FROM <grantee>;
```

2.6.9 确保非特定系统用户或角色不被授予 CREATE LIBRARY 权限

建议理由：

CREATE LIBRARY 权限允许用户创建和共享库多个关联对象，在一定条件下可能造成安全问题。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='CREATE LIBRARY'
AND GRANTEE NOT IN
('SYS','SYSTEM','DBA','MDSYS','SPATIAL_WFS_ADMIN_USR',
'SPATIAL_CSW_ADMIN_USR','DVSYS','GSMADMIN_INTERNAL','XDB');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE CREATE LIBRARY FROM <grantee>;
```

2.6.10 确保非特定系统用户或角色不被授予 GRANT ANY OBJECT PRIVILEGE 权限

建议理由：

GRANT ANY OBJECT PRIVILEGE 权限允许用户有可能读取一些数据信息，在一定条件下可能造成安全问题。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='GRANT ANY OBJECT PRIVILEGE'
AND GRANTEE NOT IN
('DBA','SYS','IMP_FULL_DATABASE','DATAPUMP_IMP_FULL_DATABASE',
'EM_EXPRESS_ALL', 'DV_REALM_OWNER');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE GRANT ANY OBJECT PRIVILEGE FROM <grantee>;
```

2.6.11 确保非特定系统用户或角色不被授予 GRANT ANY ROLE 权限

建议理由：

GRANT ANY ROLE 权限允许用户有可能读取一些数据信息，在一定条件下可能造成安全问题。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='GRANT ANY ROLE'
AND GRANTEE NOT IN
('DBA','SYS','DATAPUMP_IMP_FULL_DATABASE','IMP_FULL_DATABASE',
'SPATIAL_WFS_ADMIN_USR','SPATIAL_CSW_ADMIN_USR',
'GSMADMIN_INTERNAL','DV_REALM_OWNER', 'EM_EXPRESS_ALL',
'DV_OWNER');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE GRANT ANY ROLE FROM <grantee>;
```

2.6.12 确保非特定系统用户或角色不被授予 GRANT ANY PRIVILEGE 权限

建议理由：

GRANT ANY PRIVILEGE 权限允许为用户赋予一些特殊权限。用户有可能读取一些数据信息，在一定条件下可能造成安全问题。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='GRANT ANY PRIVILEGE'
AND GRANTEE NOT IN
('DBA','SYS','IMP_FULL_DATABASE','DATAPUMP_IMP_FULL_DATABASE'
'DV_REALM_OWNER', 'EM_EXPRESS_ALL');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE GRANT ANY PRIVILEGE FROM <grantee>;
```

2.6.13 确保非特定系统用户或角色不被授予 DELETE_CATALOG_ROLE 权限

建议理由：

DELETE_CATALOG_ROLE 权限允许为用户消除某些在日志中的记录。可能是为了掩盖某些对安全进行攻击的行为。

检查手段：通过 sql 检查

```
SELECT GRANTEE, GRANTED_ROLE
```

FROM DBA_ROLE_PRIVS
WHERE granted_role='DELETE_CATALOG_ROLE'
AND GRANTEE NOT IN ('DBA','SYS');
如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE DELETE_CATALOG_ROLE FROM <grantee>;

2.6.14 确保非特定系统用户或角色不被授予 SELECT_CATALOG_ROLE 权限

建议理由：

SELECT_CATALOG_ROLE 权限允许为用户访问数据字典中的信息。可能会导敏感数据泄露。

检查手段：通过 sql 检查

SELECT GRANTEE, GRANTED_ROLE
FROM DBA_ROLE_PRIVS
WHERE granted_role='SELECT_CATALOG_ROLE'
AND grantee not in
('DBA','SYS','IMP_FULL_DATABASE','EXP_FULL_DATABASE',
'OEM_MONITOR', 'SYSBACKUP','EM_EXPRESS_BASIC','SYSMAN');
如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE SELECT_CATALOG_ROLE FROM <grantee>;

2.6.15 确保非特定系统用户或角色不被授予 EXECUTE_CATALOG_ROLE 权限

建议理由：

EXECUTE_CATALOG_ROLE 权限允许用户执行一些 sys 模式下的包。在一定情况下会引起一些安全风险。

检查手段：通过 sql 检查

SELECT GRANTEE, GRANTED_ROLE
FROM DBA_ROLE_PRIVS
WHERE granted_role='EXECUTE_CATALOG_ROLE'
AND grantee not in
('DBA','SYS','IMP_FULL_DATABASE','EXP_FULL_DATABASE');
如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

REVOKE EXECUTE_CATALOG_ROLE FROM <grantee>;

2.6.16 确保非特定系统用户或角色不被授予 DBA 角色

建议理由：

DBA 角色含有数据库大量的权限。如果不合适的账号用户该角色在一定情况下会引起一些安全风险。

检查手段：通过 sql 检查

```
SELECT GRANTEE, GRANTED_ROLE
FROM DBA_ROLE_PRIVS
WHERE GRANTED_ROLE='DBA'
AND GRANTEE NOT IN ('SYS','SYSTEM');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE DBA FROM <grantee>;
```

2.6.17 确保非特定系统用户或角色不被授予 SYS.AUD$的 all 权限

建议理由：

SYS.AUD $表包含数据库的所有审计记录。All 权限会让用户可以对该表任意操作。可能人为手动修改审计结果，破坏记录内容。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME='AUD$'
AND GRANTEE NOT IN ('DELETE_CATALOG_ROLE');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE ALL ON AUD$ FROM <grantee>;
```

2.6.18 确保非特定系统用户或角色不被授予 SYS.USER_HISTORY$的 all 权限

建议理由：

SYS.AUD $表包含用户修改密码的所有历史信息。历史信息可能会引导致敏感信息泄露。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME='USER_HISTORY$' AND OWNER = 'SYS';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE ALL ON USER_HISTORY$ FROM <grantee>;
```

2.6.19 确保非特定系统用户或角色不被授予 SYS.LINK$的 all 权限

建议理由：

SYS.LINK$表包含链接的所有用户密码和链接信息。被低权限用户读取可能会造成安全隐患。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME='LINK$'
AND GRANTEE NOT IN ('DV_SECANALYST')
AND OWNER='SYS';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE ALL ON LINK$ FROM <grantee>;
```

2.6.20 确保非特定系统用户或角色不被授予 SYS.USER$的 all 权限

建议理由：

SYS.USER$表包含所有用户密码信息。被低权限用户读取可能会造成安全隐患。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME='USER$' AND OWNER='SYS'
AND GRANTEE NOT IN ('CTXSYS','XDB','APEX_030200','SYSMAN',
'APEX_040000','APEX_040100','APEX_040200','DV_SECANALYST','DV
SYS','ORACLE_OCM');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE ALL ON SYS.USER$ FROM <username>;
```

2.6.21 确保非特定系统用户或角色不被授予 DBA_%的 all 权限

建议理由：

DBA_视图显示与管理相关的所有用户信息。低权限用户可以读取可能导致信息泄露。

检查手段：通过 sql 检查

```
SELECT grantee||'.'||table_name FROM DBA_TAB_PRIVS
WHERE TABLE_NAME LIKE 'DBA_%'
AND GRANTEE NOT IN
('DBA','AUDIT_ADMIN','AUDIT_VIEWER','CAPTURE_ADMIN','DVSYS',
'SYSDG','DV_SECANALYST','SYSKM','DV_MONITOR','ORACLE_OCM','
DV_ACCTMGR',
'GSMADMIN_INTERNAL','XDB','SYS','APPQOSSYS','AQ_ADMINISTRAT
OR_ROLE','CTXSYS',
'EXFSYS','MDSYS','OLAP_XS_ADMIN','OLAPSYS','ORDSYS','OWB$CLI
ENT','OWBSYS',
'SELECT_CATALOG_ROLE','WM_ADMIN_ROLE','WMSYS','XDBADMIN'
,
'LBACSYS', 'ADM_PARALLEL_EXECUTE_TASK','CISSCANROLE')
AND NOT REGEXP_LIKE(grantee,'^APEX_0[3-9][0-9][0-9][0-9][0-9]$');
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE ALL ON DBA_ FROM <Non-DBA/SYS grantee>;
```

2.6.22 确保非特定系统用户或角色不被授予 SCHEDULER$_CREDENTIAL 的 all 权限

建议理由：

SCHEDULER$_CREDENTIAL 表包含数据库调度相关的一些信息。被低权限用户读取可能会造成安全隐患。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME='SCHEDULER$_CREDENTIAL' AND
OWNER='SYS';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE ALL ON SYS.SCHEDULER$_CREDENTIAL FROM
<username>;
```

2.6.23 确保 sys.user$mig 已经被删除

建议理由：

sys.user$mig 中记录了数据迁移所使用的数据库账号密码信息。攻击者可能通 sys.user$mig 获得数据库密码散列。

检查手段：通过 sql 检查

```
SELECT OWNER, TABLE_NAME
FROM ALL_TABLES
WHERE OWNER='SYS'
AND TABLE_NAME='USER$MIG';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
DROP TABLE SYS.USER$MIG;
```

2.6.24 确保 OUTLN 不被授予 EXECUTE ANY PROCEDURE 权限

建议理由：

OUTLN 不应该被授予过多的权限，尤其是执行任意存储过程权限。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='EXECUTE ANY PROCEDURE'
AND GRANTEE='OUTLN';
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ANY PROCEDURE FROM OUTLN;
```

2.6.25 确保 DBSNMP 不被授予 EXECUTE ANY PROCEDURE 权限

建议理由：

DBSNMP 不应该被授予过多的权限，尤其是执行任意存储过程权限。

检查手段：通过 sql 检查

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='EXECUTE ANY PROCEDURE'
AND GRANTEE=DBSNMP;
```

如果没有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
REVOKE EXECUTE ANY PROCEDURE FROM DBSNMP;
```

2.7 审计安全配置

数据库 12C 开始审计分为两种，一种是向前兼容的传统审计，另一种就是新的统一审计。12C 默认同时开启。可以通过操作配置只使用其中一种或两种混用。具体配置和检查方法见附录 A。

无论哪种审计默认都会开启一些审计项，但根据常年的安全经验，哪些审计项是不足的所以本章主要提醒用户还需要额外开启哪些审计项，来帮助您应对可能存在的安全风险。

2.7.1 确保审计到关键操作和信息（传统审计）

建议理由：

审计到关键操作行为非常关键，这些信息为日后安全追责奠定了基础。关键权限包含：USER、ALTER USER、DROP USER、ROLE、SYSTEM GRANT、PROFILE、ALTER PROFILE、DROP PROFILE、DATABASE LINK、PUBLIC DATABASE LINK、PUBLIC SYNONYM、SYNONYM、GRANT DIRECTORY 、 SELECT ANY DICTIONARY 、 GRANT ANY OBJECT PRIVILEGE 、 GRANT ANY PRIVILEGE 、 DROP ANY PROCEDURE、PROCEDURE、ALTER SYSTEM、TRIGGER 和 CREATE SESSION

检查手段：通过 sql 检查（已 user 为例，其他只需要替换 where 后对象即可。）

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='USER'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

如果有返回行安全,否则建议修复。

修复建议：通过以下语句修复（已 user 为例，其他只需要替换 AUDIT 后参数即可。）

```
AUDIT USER;
```

2.7.2 确保 SYS.AUD$ 所有操作被审计（传统审计）

建议理由：

SYS.AUD $表包含用户修改密码的所有历史信息。建议把所有操作都审计下来。

检查手段：通过 sql 检查

```
SELECT *
FROM DBA_OBJ_AUDIT_OPTS
WHERE OBJECT_NAME='AUD$'
AND ALT='A/A'
AND AUD='A/A'
AND COM='A/A'
AND DEL='A/A'
AND GRA='A/A'
AND IND='A/A'
AND INS='A/A'
AND LOC='A/A'
AND REN='A/A'
AND SEL='A/A'
AND UPD='A/A'
AND FBK='A/A';
```

如果有返回行安全,否则建议修复。

修复建议：通过以下语句修复

```
AUDIT ALL ON SYS.AUD$ BY ACCESS;
```

2.7.3 确保审计到关键操作和信息（统一审计）

建议理由：

审计到关键操作行为非常关键，这些信息为日后安全追责奠定了基础。关键权限包含：CREATE USER、ALTER USER、DROP USER、CREATE ROLE、ALTER ROLE、DROP ROLE、GRANT、REVOKE、CREATE PROFILE、ALTER PROFILE、DROP PROFILE、CREATE DATABASE LINK、ALTER DATABASE LINK、DROP DATABASE LINK、CREATE SYNONYM、ALTER SYNONYM、DROP SYNONYM、SELECT ANY DICTIONARY、CREATE PROCEDURE、CREATE FUNCTION、CREATE PACKAGE、CREATE PACKAGE BODY、ALTER PROCEDURE、ALTER FUNCTION、ALTER PACKAGE、ALTER PACKAGE BODY、DROP PROCEDURE、DROP FUNCTION、DROP PACKAGE、DROP PACKAGE BODY、ALTER SYSTEM、CREATE TRIGGER、ALTER TRIGGER、DROP TRIGGER、LOGON和LOGOFF

检查手段：通过 sql 检查（已 CREATE USER 为例，其他只需要替换 where 条件中 AUD.AUDIT_OPTION 后对象即可。）

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION,
AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD,
AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'CREATE USER'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

如果有返回行安全,否则建议修复。

修复建议：通过以下语句修复（以 CREATE USER 为例，其他只需要替换 CREATE USER 即可。）

1. 如果没有策略，创建策略

```
create audit policy DBSEC_UNIFIED_AUDIT_POLICY actions ALTER
USER;
```

2. 生效策略组

```
audit policy DBSEC_UNIFIED_AUDIT_POLICY;
```

3. 添加策略

```
ALTER AUDIT POLICY DBSEC_UNIFIED_AUDIT_POLICY ADD
ACTIONS CREATE USER;
```

附录 A

检查采用哪种审计方式

```
SELECT PARAMETER,VALUE FROM V$OPTION WHERE PARAMETER = 'Unified
Auditing';
```

TRUE表示完全的统一审计有效。FALSE表示并非是完全的统一审计。

切换到只使用统一审计

```
sqlplus /as sysdba
shutdown immediate
lsnrctl stop listener_name
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk uniaud_on ioracle ORACLE_HOME=$ORACLE_HOME
lsnrctl start listener_name
sqlplus /as sysdba
startup
```

完全关闭统一审计

```
sqlplus /as sysdba
```

shutdown immediate
lsnrctl stop listener_name
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk uniaud_off ioracle lsnrctl start listener_name
sqlplus /as sysdba
startup