

Sécurisez votre instance Oracle



par fadace ([Fabien Celaia](#))

Date de publication : 12.05.2007

Dernière mise à jour : 26.05.2007

...ou comment compliquer la vie des pirates... sans se laisser envahir par le côté obscur de la force.

I - Introduction.....	3
II - Sources.....	3
III - Types de dangers.....	4
IV - Dangers et mesures.....	4
IV-A - Schémas / Logins.....	4
IV-B - Mots de passe.....	4
IV-B-1 - Mots de passe simplistes.....	4
IV-B-2 - Mots de passe par défaut.....	5
IV-B-3 - Mots de passe codé en dur.....	5
IV-B-4 - Spoliation temporaire de mot de passe.....	6
IV-C - Intrusion via système d'exploitation.....	7
IV-D - Intrusion par application frontale.....	7
IV-C-1 - Informations de connexion stockées en local.....	8
IV-C-2 - Clients non patchés / non sécurisés.....	8
IV-C-3 - Fichiers autoexécutables.....	8
IV-E - Intrusion par sniffage du processus d'écoute.....	8
IV-F - Scannage de port.....	9
IV-G - Intrusion par binaires non sécurisés.....	9
IV-H - Injection SQL.....	10
IV-I - Détection par mauvais paramétrage.....	11
IV-J - Politique du droit juste et nécessaire.....	11
IV-K - Sauvegardes / exports / imports.....	13
IV-L - Documentation libre, sur Internet, externe.....	14
V - Conclusion.....	14

I - Introduction

Que ce soit dans les anciennes ou les nouvelles implémentations, le besoin de revoir les aspects sécuritaires de ces environnements s'avère nécessaire, tant du point de vue de la confidentialité d'accès aux données que de l'intégrité de celles-ci.

Les lois européennes sur la protection des données sont claires sur le fait qu'il incombe de veiller à la protection des données personnelles que l'on stocke.

D'autres informations sont jugées sensibles au sein des entreprises, comme par exemple la liste des clients, le prix des produits ou les salaires des collaborateurs de l'entreprise.

La menace interne est généralement sous-estimée et donc souvent peu prise en compte, alors que la majorité des analyses disponibles sur le sujet indiquent que la sécurisation périmétrique ne suffit plus, une grande partie des menaces provenant désormais de l'interne.

Le but de cet article est la revue des menaces impliquant les informations hébergées au sein des instances Oracle et de proposer un catalogue de mesures permettant d'y remédier ou tout le moins d'en minimiser les risques.

"Mieux vaut connaître son ennemi pour le combattre !" Conscient que donner ce type d'information sur des brèches sécuritaires peut pousser certaines personnes mal intentionnées à s'y engouffrer, je ne peux que vous conseiller de ne pas "plonger duc ôté obscur de la force"... et d'envisager ce type d'information de manière la plus professionnelle qu'il soit.



II - Sources

Le but principal est la diminution sensible des risques d'accès ou de modification inappropriée des données. Il se base principalement sur les sources suivantes:

- les compétences et expériences des DBA
- les différents scénari d'intrusions disponibles sur Internet
- les outils **BackTrack 2**
- les informations reçues lors du cours  **Oracle anti-hacking** donné par M. Alexander Kornbrust, un des spécialistes de la sécurité sur Oracle
- le  **site de Pete Finnigan**
- la documentation officielle d'Oracle :  **Oracle Database Security Guide**
- les diverses autres documentations délivrées par Oracle sur le sujet (via  **Metalink**,  **Grid control**, etc.)
- la [FAQ](#) **Faq Oracle Sécurité de developpez.com**

III - Types de dangers

Les dangers peuvent être catégorisés de la manière suivante :

No	Type de danger	Explication
1	Déni de service	Plantage volontaire du SGBDR rendant l'accès aux données impossible
2	Intrusion	Accès malintentionné afin d'obtenir plus d'informations que nécessaire
3	Injection SQL	Utilisation de failles du SGBDR afin de s'octroyer des droits excessifs
4	Corruption de données	Pourrissement des données métier par mauvaise configuration, envie de nuire, ...

IV - Dangers et mesures

Ici sont exposés les dangers latents et les mesures à apporter afin de les éviter.

IV-A - Schémas / Logins

Oracle mixte la notion de schéma et de login. Un Schéma est en fait lié à un utilisateur qui est propriétaire d'au moins un objet. Un schéma, dans son rôle de conteneur d'objets devrait en fait être un login verrouillé dont le mot de passe ne devrait même pas être connu.

Pour les schémas métier créés par vos équipes de développement, c'est généralement le cas. Personne ne se connecte directement sur ces logins, mais un utilisateur spécifique et applicatif peut y accéder.

Dans les fait malheureusement, la plupart des applicatifs externes n'utilisent pas le prédicat du schéma pour accéder aux tables. Il leur faut donc soit se connecter directement sous le nom du schéma, soit utiliser la notion de synonyme.

Couplé à un trigger sur connexion, la commande `ALTER SESSION SET CURRENT_SCHEMA` offre cependant une alternative élégante et permet la dissociation du schéma et du login utilisé.

Pour couronner le tout, certains logins/schémas système Oracle sont installés par défaut dans toute nouvelle instance et ont des droits excessifs ou/et des failles sécuritaires. Il est donc primordial de ne laisser sur une instance Oracle que les schémas nécessaires. Au besoin, il est toujours possible de recréer ces schémas via les scripts Oracle. Le surplus doit être désinstallé ou, au minimum, verrouillé.

La liste des schémas Oracle, une explication sommaire et les scripts de création et de désinstallations sont décrits dans l'article [Les schémas Oracle](#).

IV-B - Mots de passe

IV-B-1 - Mots de passe simplistes

Type	2 (intrusion)
Contrôle	Vérification régulière à l'aide d'outils Backtrack (<code>checkpwd + francais.txt</code>), basé sur la comparaison de chaînes hachées. Vérification périodiques en brute force
Mesure	Demander aux utilisateurs de modifier les mots de passe faibles; forcer la complexification des mots de passe via les profils Oracle

```

CREATE PROFILE Compexite_Pwd_profile
LIMIT PASSWORD_LIFE_TIME 30
LIMIT PASSWORD_REUSE_TIME 180
LIMIT PASSWORD_REUSE_MAX 3 ;

ALTER USER scott PROFILE my_profile;
    
```

```

C:\oracle_checkpwd>checkpwd122.exe system/monpwd@hostdvp:1521/ORA10G.DVP.COM default_passwords.txt
Checkpwd 1.22 - (c) 2007 by Red-Database-Security GmbH
Oracle Security Consulting, Security Audits & Security Trainings
http://www.red-database-security.com
    
```


```

initializing Oracle client library
connecting to the database
retrieving users and password hash values
disconnecting from the database
opening weak password list file
reading weak passwords list
checking passwords
Starting 1 threads
JRULES OK [OPEN]
SNPM OK [OPEN]
SNPW OK [OPEN]
MIDOFF OK [OPEN]
WAREHOUSE has weak password BUSINESS [OPEN]
DISTRIBUTE has weak password HOUSE [OPEN]
MDSTATUS has weak password MDSTATUS [OPEN]
QC OK [OPEN]
DASHBOARD OK [OPEN]
SYS_USER OK [OPEN]
DEPLOYER OK [OPEN]
TSMSYS OK [LOCKED]
DBSNMP OK [OPEN]
EXFSYS OK [LOCKED]
XDB OK [LOCKED]
SYSMAN OK [LOCKED]
MGMT_VIEW OK [OPEN]
SYS OK [OPEN]
SYSTEM OK [OPEN]
OUTLN OK [LOCKED]
    
```

```

Done. Summary:
  Passwords checked      : 43241
  Weak passwords found  : 3
  Elapsed time (min:sec) : 0:00
  Passwords / second    : 43241
    
```

IV-B-2 - Mots de passe par défaut

Type	2 (intrusion)
Contrôle	Validation régulière à l'aide d'outils Backtrack ( checkpwd + default_passwords.txt), basé sur la comparaison de chaînes hachées. Contrôle périodiques en brut force
Mesure	Modification des mots de passe par la DBA. Eviter de rejouer catproc.sql (mise à défaut de mors de passe)

IV-B-3 - Mots de passe codé en dur

Type	2 (intrusion)
Contrôle	Via trigger sur connexion (cf. ci-dessous), en s'assurant que l'applicatif appelant est le bon.
Mesure	Externalisation du mot de passe crypté dans fichiers de paramètres

Documentation pour chacune des data sources
 Modification chronique de ces mots de passe

Voici un exemple de déclencheur sur connexion qui permet de n'autoriser l'utilisation de certains applicatifs qu'à certains comptes.

```

create or replace TRIGGER "SYS"."BLOCK_USER_ACCESS"
AFTER LOGON ON DATABASE
DECLARE
    v_prog sys.v_$session.program%TYPE;
    v_dbuser sys.v_$session.username%TYPE;
    v_osuser sys.v_$session.osuser%TYPE;
    v_db sys.v_$database.name%TYPE;

/*
 * Auteur : Fabien Celaia
 * Date : 7 juillet 2006
 * Desc. : Empeche l'utilisation de certains logins avec certains applicatifs
 */

BEGIN
    /* Récupération des informations utiles dans v$session */

    SELECT upper(program), upper(username), upper(osuser)
    INTO v_prog, v_dbuser, v_osuser
    FROM sys.v_$session
    WHERE auid = USERENV('SESSIONID')
    AND auid != 0 -- N'impacte pas la connexion SYS
    AND rownum = 1; -- Gestion du parallélisme (memes AUIDSID)

    SELECT upper(name)
    INTO v_db
    from v$database ;

    /* Utilisation à proscrire */
    IF v_dbuser NOT IN ('SYSTEM', 'SYS', 'STREAM', 'DBSNMP')
    AND coalesce(v_osuser, 'SYSTEM') not in ('SYSTEM')
    AND (v_prog = 'TOAD.EXE'
    OR v_prog LIKE '%ACCESS.EXE'
    OR v_prog LIKE '%SQUIREL%'
    OR v_prog LIKE '%SQLPLUS%'
    OR v_prog LIKE '%SQL DEV%'
    OR v_prog LIKE '%MSQ%.EXE')
    THEN
        RAISE_APPLICATION_ERROR(-20000, v_osuser || ' n'est pas autorisé à utiliser ' ||
        v_prog || ' sur l''environnement ' || v_db) ;
    END IF;

    EXCEPTION
        WHEN NO_DATA_FOUND THEN NULL;
END;
    
```

Il s'agit là d'une sécurité toute relative compte tenu du fait qu'il est aisé de modifier le nom du programme appelant via PL-SQL, que ce trigger ne se déclenche pas avec sysdba... et pour de nombreuses autres raisons exposées dans la [FAQ Faq Oracle](#).

IV-B-4 - Spoliation temporaire de mot de passe

Avec des droits DBA, il est tout à fait aisé de s'approprier l'identité d'un utilisateur spécifique de manière temporaire:

Le mot de passe hashé est un md5 de la concaténation du login, du mot de passe et d'un bout de chaîne. Il est donc invariable d'une instance à une autre pour un login/mot de passe donné. De plus, le mot de passé hashé de scott/tigger sera le même que celui de sco/ttigger.

Dès la version 10g, le mot de passe devient sensitif, ce qui n'était pas le cas au préalable. Mixez donc majuscules et minuscules afin de complexifier la tâche aux outils de brute force.

Dans un premier temps, on relève son mot de passe crypté, connecté en tant que dba.

```
SQL> connect system/motdepasse
Connecté

SQL> select password from dba_users where username='MONCOCO'

PASSWORD
-----
F894765434402B67
```

Il est ensuite aisé de changer son mot de passe, puis de se connecter sous son profile

```
SQL> alter user MONCOCO identified by MonPwd

User altered

SQL> connect MONCOCO/MonPwd
Connecté
```

Notre méfait accompli, il ne nous reste plus qu'à remettre l'ancien mot de passe

```
SQL> connect system/motdepasse
Connecté

SQL> alter user MONCOCO identified by values 'F894765434402B67'
```

Il est possible de limiter ce genre d'intrusion, en empêchant par exemple la modification d'un mot de passe via trigger sur ALTER, mais cela ne limitera pas un compte de type sysdba.

IV-C - Intrusion via système d'exploitation

Une des principale faille de la sécurité Oracle est qu'il n'est quasi pas possible d'empêcher à un administrateur du système d'exploitation de se connecter en tant qu'administrateur de la base de données.

Sous Windows, un administrateur est tout à fait en mesure de se mettre dans le rôle système oradba. Sous Unix, idem pour le rôle, et de manière encore plus aisée, il sera en mesure d'exécuter un sudo - oracle.

Cette capacité à se connecter sur une instance Oracle sans mot de passe, directement en sqlplus / as sysdba permet à quiconque qui aurait des droits d'administration sur la machine de se connecter en tant que superuser sur Oracle.

Cela peut aussi être considéré comme un avantage lors du départ innopiné ou du décès du DBA !

Type	2 (intrusion)
Contrôle	Eviter, pour les DBA aussi, de se connecter en tant que sys
Mesure	Comptes DBA spécifiques et nommés, audit fait (par exemple, c'est le cas avec Oracle Grid Control)


IV-D - Intrusion par application frontale

Que ce soit via SQLDeveloper ou via Toad, les configurations de connexion sont stockées sur le PC client du développeur/DBA. Bien que ces 2 applicatifs stockent les mots de passe en cryptage EAS, il est possible, au travers d'un disque partagé par exemple, d'aller " vampiriser " le fichier de configuration afin de s'approprier les accès excessifs, sans pour cela connaître les mots de passe utilisés.

IV-C-1 - Informations de connexion stockées en local

Les fichiers sensibles sont


- pour SQL Developer : ...\\sqldeveloper\jdev\system\oracle.onlinedb.11.0.0.37.42\IDE*.*
- pour Toad : ...\\Quest Software\Toad for Oracle\User Files*.*
- pour sqlplus, sqlplusw : toute sorte de raccourci dans environnement Windows

Type	2 (intrusion)
Contrôle	Validation régulière à l'aide d'outils Backtrack ( checkpwd + default_passwords.txt), basé sur la comparaison de chaînes hachées.
Mesure	Suppression ou limitation du partage C\$ sur les postes. Interdiction faite aux utilisateurs d'utiliser toute forme de sauvegarde de mot de passe dans les connexions.

IV-C-2 - Clients non patchés / non sécurisés

Il est nécessaire de maîtriser l'installation des clients Oracle, ceux-ci étant aussi sensibles aux failles de sécurité. Bien souvent, l'installation des bases et de leurs binaires sont dévolues au DBA, mais les installations clientes Oracle sont souvent faites "à la sauvage", via des packages ou un département "bureautique" peu sensible aux soucis sécuritaires qui nous occupent.

Type	3 (injection SQL)
Contrôle	Eviter toute installation cliente sauvage
Mesure	Catalogage de toute installation cliente Patch sécurité à installer côté client aussi

Type	4 (corruption)
Contrôle	Installation des clients correcte, variables d'environnement adéquates (NLS) Pas de caractères étranges saisis
Mesure	Reprise en main des installations clientes Configuration adéquate des clients Oracle, et plus particulièrement des  variables NLS

IV-C-3 - Fichiers autoexécutables

Du code SQL néfaste peut être insidieusement placé sur le disque d'un super-utilisateur et autoexécuté au démarrage de sqlplus. Ceci se fait dans les fameux fichiers glogin.sql et login.sql, situés soit dans Oracle_home\bin, soit dans Oracle_home\sqlplus\admin.

Type	3 (injection SQL)
Contrôle	S'assurer que les fichiers login.sql et glogin.sql ne se trouvent pas dans le répertoire \${ORACLE_HOME}\bin des clients
Mesure	Démarrer sqlplus à l'aide d'un fichier script/batch nettoyant au préalable tout glogin.sql ou login.sql du répertoire incriminé.

IV-E - Intrusion par sniffage du processus d'écoute



Le listener d'Oracle peut être détourné à des fins néfastes. Il permet de sniffer le protocole de transfert Oracle non crypté. Les requêtes, les données passent en clair au travers de la trame. Plusieurs solutions peuvent être mise sur pied afin de pallier à ce problème. Il convient aussi de spécifier au listener de quelles bases il est censé gérer la communication.

Type	1 (dénier de service)
Contrôle	Check du listener.ora afin de s'assurer de sa bonne configuration
Mesure	- en version pré-10, mise en place d'un mot de passe pour le listener - dès la version 10, pas de mot de passe : la mise en place de ce dernier permettrait un accès distant beaucoup plus dangereux

Type	2 (intrusion)
Contrôle	Check du listener.ora, de \$TNS_ADMIN afin qu'ils ne comportent pas de redirection
Mesure	- en version pré-10, mise en place d'un mot de passe pour le listener - dès la version 10, pas de mot de passe : la mise en place de ce dernier permettrait un accès distant beaucoup plus dangereux, et il n'est pas supporté dans un environnement RAC tel que le nôtre. - acquisition d'Oracle Advance Security Option (~ 10'000 USD/CPU) incluant Network Security - Cryptage des données sensibles sur la base et décryptage sur le client

Type	3 (injection SQL)
Contrôle	Check du listener.ora, du tnsnames.ora, de \$TNS_ADMIN afin qu'ils ne comportent pas de configuration extproc
Mesure	Suppression de la possibilité d'utiliser des procédures externes

IV-F - Scannage de port

Pour les applications n'étant pas orientées extranet, le risque de subir ce genre d'attaque de l'extérieur est plus limité. Cependant, il existe en interne, mais est aisément détectable, les outils de scannage ( nmap,  amap, etc.) étant perçus comme des attaquants par les outils réseau. Le risque existe néanmoins puisque ce type d'attaque est en mesure de faire tomber certains modules Oracle (Notification Delivery Service).

Type	1 (dénier de service)
Contrôle	Par outils réseau interne (IDS) détectant ce type de scannage.
Mesure	Audit réseau

IV-G - Intrusion par binaires non sécurisés

Les failles de sécurité n'étant fixées que dans les dernières versions, les anciens binaires Oracle sont plus sensibles. Il convient donc de supprimer toute ancienne version de binaire.

Type	3 (intrusion)
Contrôle	Suppression de dumpsga (permettant de flusher la mémoire sur disque). Cryptage des datafiles
Mesure	<pre>rm \$ORACLE_HOME/bin/dumpsga</pre> Utilisation de Transparent Data Encryption d'Oracle

Type	4 (injection SQL)
Contrôle	Suppression des binaires non patchés
Mesure	<pre>rm \$ORACLE_HOME/bin/*.bak
rm \$ORACLE_HOME/bin/*.0</pre>

IV-H - Injection SQL

Certains modules Oracle offrent des possibilités d'intrusion SQL. Il est donc nécessaire de réduire au maximum les droits (par défaut ALL to PUBLIC) sur ces modules. Ceux-ci ne peuvent pas être purement et simplement supprimés, car appartiennent au noyau Oracle.

```

revoke execute on sys.dbms_obfuscation_toolkit from public ;
grant execute on sys.dbms_obfuscation_toolkit to sysman ;
grant execute on sys.dbms_obfuscation_toolkit to dbsnmp ;
grant execute on sys.dbms_obfuscation_toolkit to wksys ;
Grant execute on sys.dbms_obfuscation_toolkit to flows_030000 ;
revoke execute on sys.utl_tcp from public ;
grant execute on sys.utl_tcp to sysman ;
revoke execute on sys.utl_http from public ;
grant execute on sys. utl_http to ordplugins ;
revoke execute on sys.utl_smtp from public ;
grant execute on sys.utl_smtp to sysman ;
revoke execute on sys.utl_file from public ;
grant execute on sys. utl_file to ordplugins ;
Grant execute on sys.utl_file to mitg_rml ;
Grant execute on sys.utl_file to mitg ;
grant execute on sys. utl_file to xdb ;
grant execute on sys. utl_file to sysman ;
grant execute on sys. utl_file to dmsys ;
revoke execute on sys.dbms_lob from public ;
grant execute on sys.dbms_lob to xdb ;
Grant execute on sys.dbms_LOB to ctxsys ;
Grant execute on sys.dbms_lob to flows_030000 ;
Grant execute on sys.dbms_lob to mitg_rml ;
Grant execute on sys.dbms_lob to mitg ;
grant execute on sys.dbms_lob to dmsys ;
grant execute on sys.dbms_lob to exfsys ;
grant execute on sys.dbms_lob to dba ;
grant execute on sys.dbms_lob to ctxsys ;
grant execute on sys.dbms_lob to mdsys ;
grant execute on sys.dbms_lob to ordsys ;
grant execute on sys.dbms_lob to wksys ;
grant execute on sys.dbms_lob to olapsys ;
grant execute on sys.dbms_lob to ordplugins ;
revoke execute on sys.dbms_job from public ;
grant execute on sys.dbms_job to dbsnmp ;
grant execute on sys.dbms_job to wksys ;
grant execute on sys.dbms_job to exfsys ;
Grant execute on sys.dbms_job to flows_030000 ;
revoke execute on sys.utl_inaddr from public ;
revoke execute on sys.dbms_export_extension from public ;
grant execute on sys.dbms_export_extension to dba ;
grant execute on sys.dbms_export_extension to system ;
revoke execute on sys.dbms_backup_restore from public ;
revoke execute on sys.dbms_sql from public ;
grant execute on sys.dbms_sql to sysman ;
grant execute on sys.dbms_sql to system ;
grant execute on sys.dbms_sql to xdb ;
grant execute on sys.dbms_sql to exfsys ;
Grant execute on sys.dbms_sql to flows_030000 ;
grant execute on sys.dbms_sql to dmsys ;
Grant execute on sys.dbms_sql to mitg_rml ;
Grant execute on sys.dbms_sql to mitg ;
grant execute on sys.dbms_sql to mdsys ;
grant execute on sys.dbms_sql to olapsys ;
grant execute on sys.dbms_sql to ctxsys ;
grant execute on sys.dbms_sql to oracle_ocr ;
revoke execute on sys.dbms_ldap from public ;
revoke execute on sys.dbms_advisor from public ;
Grant execute on sys.dbms_ldap to flows_030000 ;
GRANT EXECUTE ON SYS.DBMS_LDAP TO sysman ;

```


Sans oublier pour chaque schéma pouvant être importé:

```
grant execute on sys.dbms_export_extension to MonSchema ;
```

Pour diverses raisons (installation de modules Oracle, tel que la dbconsole par exemple, il est souhaitable d'avoir les droits initiaux sur ces modules. Voici donc les requêtes SML de rétro conversion :

```
grant all on sys.dbms_obfuscation_toolkit TO PUBLIC ;
grant all on sys.utl_tcp TO PUBLIC ;
grant all on sys.utl_http TO PUBLIC ;
grant all on sys.utl_smtp TO PUBLIC ;
grant all on sys.utl_file TO PUBLIC ;
grant all on sys.dbms_lob TO PUBLIC ;
grant all on sys.dbms_job TO PUBLIC ;
grant all on sys.utl_inaddr TO PUBLIC ;
grant all on sys.dbms_export_extension TO PUBLIC ;
grant all on sys.dbms_backup_restore TO PUBLIC ;
grant all on sys.dbms_sql TO PUBLIC ;
grant all on sys.dbms_ldap TO PUBLIC ;
grant all on sys.dbms_advisor TO PUBLIC ;
```

IV-I - Détection par mauvais paramétrage

En spécifiant une SORT_AREA_SIZE = 0, les tris sont redirigés sur le disque (tablespace temp) plutôt que la mémoire, et donc éditables via des outils tels que  BBED.

Type	2 (intrusion)
Contrôle	S'assurer que la sort_area_size ne soit pas égale à 0
Mesure	Evite la gestion des tris sur disque plutôt que mémoire

Type	2 (intrusion)
Contrôle	Crypter les tablespaces
Mesure	Utiliser le module Oracle Transparent Data Encryption

Ceci étant dit, cette attaque est purement théorique puisque

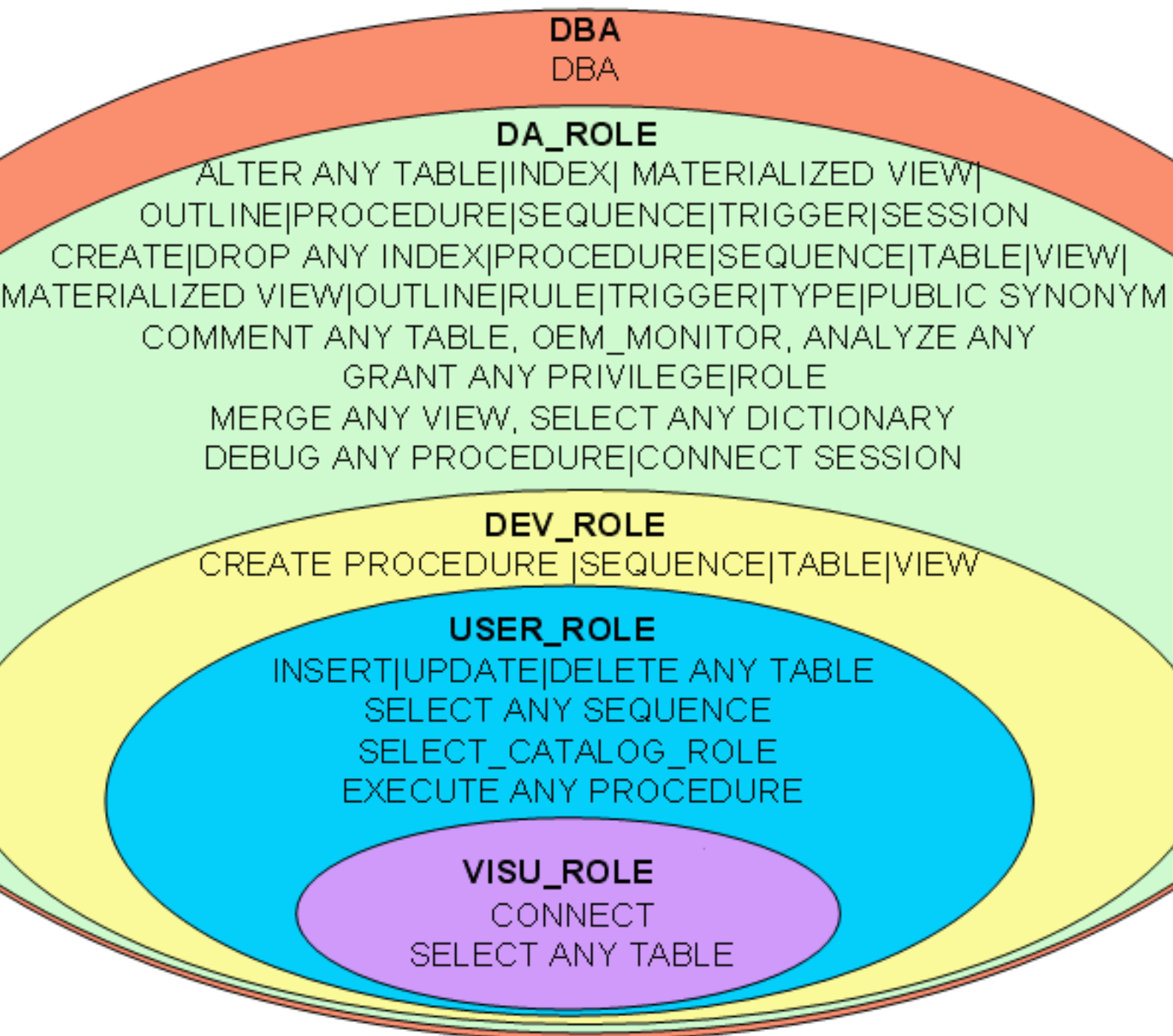
- BBED nécessite un accès local sur le serveur
- BBED peut aussi attaquer directement les tablespaces de données

Elle reste cependant une alternative de contournement pour les données protégées par Virtual Private DB, Label Security, et autre Database Vault.

IV-J - Politique du droit juste et nécessaire

Souvent malheureusement, la priorité est donnée au développement, le côté sécuritaire étant temporairement mis de côté afin d'éviter de ralentir l'avancée des projets. Les droits sont souvent attribués de manière lâche afin de permettre des installations peu problématiques, les spécificités logicielles en la matière étant souvent minimums et lacunaires. Il convient de reprendre la main sur cet état de fait afin de n'attribuer à un compte que les droits justes et nécessaires.

De manière basique, certaines structures s'accomode d'une hiérarchie de rôles minimum, dont voici un exemple



Hiérarchie de rôles minimum

Une gestion de la sécurité plus sérieuse ne pourra se faire sans une analyse longue et rébarbative, et sans les couacs inhérents à une restriction de droits. Il va sans dire que plus cette tâche sera retardée et plus elle sera pénible à mettre en place.

Il convient donc de reprendre cette stratégie de rôles de manière plus sectorielle. Une première mesure serait de créer des droits en baissant la granularité actuelle qu'est la base de données à une granularité de schéma.

Type	2 (intrusion)
Contrôle	S'assurer qu'un utilisateur ayant un droit sur un domaine ne puisse traiter un autre domaine
Mesure	Pour chaque schéma spécifique, créer un droit en lecture seule et un droit supérieur en lecture/écriture Dissocier droits nécessaires à la mise en place du modèle et droits nécessaires à sa gestion.

Pour reprendre l'exemple le plus flagrant, en production, seul le déployeur devrait être en mesure de créer/modifier/supprimer des objets/d'utiliser le DML.

D'autres mesures, plus conceptuelles celles-ci, devraient être prise afin d'assurer un minimum de cohérence de données, et ce même dans un environnement 3-tiers.

Type	2 (intrusion)
Contrôle	Crypter les données, autoriser les accès à certains utilisateurs uniquement
Mesure	Installation de Oracle Database Vault Utilisation de fonctions de cryptage/hachage "maison" Application des contextes Oracle (protection au niveau tuple), utilisations de plusieurs pools de connexion avec logins spécifiques, attribution de permissions au niveau colonnes (protection au niveau colonne)

Oracle offre des possibilités énormes de gérer des hiérarchies de rôles, des profiles... malheureusement, certaines lacunes péjorantes existent et entravent la mise en place rigoureuse des autorisations.

- Certains droits doivent être donnés en direct sur les logins afin que les comportements soient corrects (surtout au niveau de la programmation de packages PL-SQL)
- Les quotas ne peuvent pas être attribués à des rôles
- etc.

IV-K - Sauvegardes / exports / imports

A partir d'un jeu de sauvegarde ou d'un fichier d'export, il est possible de remonter toute l'information sensible sur une instance étrangère/non sécurisée.

Il est dès lors nécessaire de gérer ces médias avec toute la prudence qui s'impose.

Type	2 (intrusion)
Contrôle	2 (intrusion)
Mesure	- S'assurer que les bandes sont confinées dans un lieu sûr. - Si possible techniquement, crypter les fichiers avant de les archiver. - Utiliser le module Secure Backup d'Oracle

Par défaut, les fichiers d'export ont des droits trop étendus car ils utilisent le masque (umask) par défaut. Il convient donc d'être prudent lorsqu'on réimporte un fichier ayant été stocké préalablement sur le disque ou venant de l'externe. Il est aisé d'inclure dans le code SQL du fichier d'export des portes dérobées.

Type	2 (intrusion)
Contrôle	Export dans environnement sécurisé
Mesure	- Ne pas laisser traîner des fichiers d'export sur les disques - Modifier leurs autorisations au niveau de l'OS

Type	3 (injection SQL)
Contrôle	Ne pas importer un fichier vérolé
Mesure	<ul style="list-style-type: none"> - Ne pas laisser traîner des fichiers d'export sur les disques - Modifier leurs autorisations au niveau de l'OS - Auditer tout fichier à importer venant de l'externe


IV-L - Documentation libre, sur Internet, externe

Il est primordial qu'aucune information interne concernant vos instances ne se retrouve en DMZ ou sur Internet.

Type	2 (intrusion)
Contrôle	Les informations se trouvant dans les SR ouvertes chez Oracle, et plus particulièrement dans les fichiers attachés à la SR doivent avoir une durée de vie limitée à la résolution du problème.
Mesure	<ul style="list-style-type: none"> - Confirmation d'Oracle que les fichiers attachés ne survivent pas à la clôture d'une SR. - Caviardage du nom des machines et des adresses IP lors du chargement de fichiers liés. - Aucune information de sécurité (mot de passe, login, adresse IP) ne devrait figurer en clair dans les SR de Metalink.

V - Conclusion

Ce document n'a pas pour ambition d'empêcher toute violation. Une personne motivée sera toujours à même de violer les systèmes informatiques, quels qu'ils soient. Prendre conscience de ces danger et y apporter un certain nombre de mesures complexifiera la stratégie d'attaque d'un expert, mais surtout freinera les ardeurs de la majeure partie des personnes peu formées qui pourraient être amenées à commettre des actes frauduleux sur vos systèmes.

Je vous conseille de plus d'utiliser  **Oracle Grid Control** pour gérer votre parc : la plupart des brèches sécuritaires y sont détectées.