Pratique

Introduction à la sécurité sous Oracle

Mikoláš Panský 

Degré de difficulté

Cet article porte essentiellement sur le niveau de sécurité des serveurs de base de données Oracle. Dans cet article nous parlerons de l'histoire d'Oracle, des produits autour des bases de données et de leur architecture, aux techniques de base de Hacking sur Oracle et aux techniques permettant de se protéger de ce genre d'attaques.

L'histoire d'Oracle Corporation commence en 1977, date à laquelle son activité est fondée sur les laboratoires de développement logiciel. En 1979 : SDL est renommé Relation Software, Inc (RSI). Cette même année, l'entreprise lance Oracle v2 en tant qu'un des premiers systèmes commerciaux de bases de données relationnelles.

Cette version implémentait des fonctions basiques de SQL : requêtes et jointures. Oracle Corporation a ce nom depuis 1983, date à laquelle a été lancée la version 3 écrite en C et supportant les transactions. En 1984, paraît la version 4, 1985 version 5 (modèle client-serveur), 1989 Oracle Corp. sur le marché des applications avec Oracle Financial et implémentation de PL/SQL. En 1992, sort la version 7h – entrepôt de données (*DataWarehouse*) avec le support de l'intégrité référentielle, les procédures stockées et les triggers. En 1997, la version 8 supporte l'approche orientée objet et les applications multimédia, en 1999 la version 8i supporte Internet et la Machine Virtuel Java (mieux connue sous le nom : JVM). L'année 2001 voit la sortie de Oracle 9i avec la possibilité de lire les documents XML, et également les RAC (*Real Application Clusters*). Aujourd'hui, la version actuelle 10g Release 2 supporte le Grid.

Oracle est disponible dans différentes versions. Chacune a des applications spécifiques. Nous verrons ici les jeux de données d'Oracle. La base de données d'Oracle a plusieurs éditions : l'édition standard (utilisation de 4 CPU maximum, sans limite de mémoire et est utilisable en *Cluster*), l'édition d'entreprise (EE) inclut quelques fonctions avancées de sécurité. Il est possible d'ajouter le *Database Vault*, qui permet la protection des données contre les administrateurs de bases de données (DBA), la sécurité avancée permet la communication réseau cryptée :

Cet article explique...

- Des informations générales sur Oracle.
- Les hacks de base sur Oracle.
- Techniques basiques pour se défendre.

Ce qu'il faut savoir...

- Élements de base sur le serveur de base de données Oracle.

cryptage des données dans une base de données, authentification plus forte et finalement un Label de Sécurité qui permet de définir les privilèges de sécurité et le label des utilisateurs – la sécurité de base. Il y a également l'édition standard Ed : *Standard Edition One* avec support de 2 CPU maximum, l'édition personnelle : *Personal Edition* sans la RAC ciblée pour les développeurs et *l'Edition Express* avec 1 CPU, 1Go de RAM et 4 Go limite.

Le système de base de données d'Oracle, d'un point de vue physique, se compose de processus, qui s'exécutent sur des systèmes d'exploitation hôtes, la structure de la mémoire logique (Instance) et la structure physique des fichiers - base de données. Les processus sont divisés en processus utilisateur et en processus serveurs. Quand l'utilisateur exécute l'application, le processus utilisateur se connecte à l'instance. Si la communication est établie, la session démarre.

Pour chaque utilisateur, le serveur alloue un PGA (*Program Global Area*) dans lequel il stocke les variables de session. Une instance d'Oracle est faite par la mémoire principale : SGA (*System Global Area*) et les processus en tâches de fond. Les processus les plus importants sont le *System Monitor* – SMON (prend soin d'une restauration en cas de problème, compacte l'espace libre dans une Base de Données), *Process Monitor* – PMON (surveillance des processus actifs et assure leur support), DBW – *Database Writer* et *Log Writer* – LGWR (écrit les enregistrements permettant un retour en arrière). Oracle est composé de fichiers de contrôle (les contrôles de fichiers incluent le nom de la base, l'emplacement des fichiers de données and refont des Logs), de fichiers de données et les Redo Logs (enregistrant l'ensemble des changements dans une *Bdd*). L'information sur les processus en cours est placée dans les tableaux *V$PROCESS* et *V$SESSION*. La communication avec le monde extérieur est gérée par le Listener d'Oracle. Sa configuration se situe dans le fichier :.ora. SID (Oracle System Identifier, qui resout les instances de bdd et identifient les bdd), le protocole et le port sont stockés dans : listener.ora. Le Listener écoute pour des requêtes sur la base de données. Après avoir perçu n'importe quelle connexion, il envoie le numéro de port au client. Le client se connecte ensuite au port et s'authentifie. Le Listener pourrait aussi être utilisé par le package PL/SQL ou par des procédures externes.

La structure logique de la base est composée d'utilisateurs, schemas (objects appartenant à l'utilisateur), droits, rôles, profils et objets. Les utilisateurs dans la base ont des identités uniques, qui ont accès à des objets de la base. Les utilisateurs sont le plus fréquemment identifiés par des mots de passe. Chaque utilisateur a un Schema, lui appartenant et où ses objets sont stockés. Les privilèges sont un ensemble d'opérations qu'un utilisateur pourrait utiliser. Les profils sont un ensemble d'options qui limite l'utilisation de la base de données. Ils pourraient définir des tentatives maximum d'essais de mot de passe avant que le compte ne se ferme etc. Les *Tables* ont des lignes et colonnes. L'accès aux tables peut être défini et limité sur la base des lignes avec le *Virtual Private Database* (Base de donnée virtuel privé). Les triggers (déclencheurs) sont des instructions stockées, qui s'exécutent sur des événements comme : une insertion dans une table ou l'arrêt de la base de données. Les procédures stockées sont des programmes écrits sous PL/SQL (langage de programmation SQL). Toutes les informations sur la base de données sont stockées en dictionnaire de données.

Hacking Oracle

Avant de commencer, il doit y avoir une phase d'analyse du réseau. Cette phase nécessite la recherche d'informations essentielles, qui pourraient être récupérées par une base de données Whois, des moteurs de recherche, Serveurs DNS ou par du *Social Engineering*. Le moteur de recherche pourrait également être utilisé pour trouver un système précis suivant les termes de la recherche, qui est un identifiant unique pour la bonne page. Ce terme pourrait par exemple rechercher isqlplus (interface web pour tapper des requêtes sur des bdd Oracle), des fichiers de configuration ou les Editions Express. Ces termes pourraient ressembler à : *intitle:icql intitle:release inurl:isqlplus, listener filetype:ora či inurl:apex intitle: Application Express Login*.

La prochaine étape est un scan plus approfondi de l'OS, à l'aide d'outils comme (`nmap`, `amap`, `tsnping`) ou passifs (`scanrad`). La première étape consiste à voir les ports ouverts. Oracle, dans sa configuration standard, écoute sur les ports basiques qui pourraient être identifiés. Pour trouver des Listeners en exécution, utilisez des outils comme : TSNPING. Après que le serveur de bdd ait été trouvé, essayons d'obtenir sa version, la Plate-forme, le SID et la configuration, ceci grâce à TSNLSNR IP Client, qui permet les commandes pings, et qui obtient la version, les services et l'état en cours du serveur. L'information demandée n'est obtenue que lorsque l'Administrateur n'a pas mis de mot de passe au Listener d'Oracle. Si un mot de passe est attribué au Listener, il n'est plus possible d'obtenir des informations. Il y a d'autres outils disponibles pour l'exploration des Listeners : TNSCmd et OScanner. NGSSQuirrel est quant à lui, un produit commercial. Ce programme est complexe mais dispose de nombreuses fonctions. Certaines sont seulement disponibles avec un compte Oracle, cependant il permet également des attaques de type : Brute Force ou avec des Dictionnaires sur un compte utilisateur. Si un Listener est non-sécurisé, plusieurs angles d'attaques sont envisageables. Dans le passé, il y avait beaucoup d'alertes de sécurité. Certaines sont des attaques : *NERP* DoS, requêtes de versions illégales, transfert de données trop faible, attaques de fragmentations ou des attaques *SERVICE_NAME* DoS. De plus, il est possible de changer le mot de passe du Listener qui conduit au *HiJacking*,

à l'arrêt du Listener ou de l'ensemble des paramètres avec la commande SET. Quand le SID et la version sont connus, différents noms d'utilisateurs et mots de passe peuvent être testés : d'abord, les mêmes noms d'utilisateurs et mots de passe. Ensuite, testez les noms d'utilisateurs et mots de passe usuels. La prochaine étape dans l'ordre serait une attaque par dictionnaire et finalement une brute force. Hydra permet de vérifier les noms d'utilisateurs et mots de passe.

L'autre possibilité, pour obtenir l'accès au serveur de bdd Oracle est de sniffer la connexion. Si la communication entre l'utilisteur et le client n'est pas sécurisée, elle peut être sniffée par n'importe quel sniffer sur le réseau. En premier lieu, l'utilisateur envoie le nom d'utilisateur à la base de données. Si le nom d'utilisateur existe alors le serveur vérifie le cryptage *hash* du mot de passe de l'utilisateur. Il utilise un numéro secret basé sur l'heure du système.

Après avoir obtenu l'accès à la base, il est nécessaire de vérifier s'il est possible d'avoir accès en chaîne aux droits sur le système. Les méthodes les plus connues sont : les Injections SQL, *Buffer Overflow* et le *Cross – Site Scripting*. La logique de base des injections PL/SQL est d'attaquer les programmes, autorisant les entrées utilisateur.

Cette entrée permet à un hacker d'écrire son propre code. Cette méthode est utilisée par exemple pour outrepasser le *DBMS_ASSERT* (Oracle 10g R2) – ceci est utilisé pour vérifier les données saisies. Il y a aussi une autre méthode appelée le Dangling Cursos Snarfing. Le principe réside sur le fait qu'Oracle ne ferme pas tous les curseurs après son utilisation. Si un utilisateur avec des privilèges crée un curseur, ce dernier peut être utilisé par utilisateurs ayant moins de privilèges pour parcourir en cascade les droits sur les utilisateurs ayant plus de privilèges. Pour contrer cette méthode, les curseurs ouverts devraient être fermés juste après leur utilisation. Après avoir parcouru les privilèges, plusieurs choses sont possibles. Créer un Rootkit pour installer une porte dérobée (backdoor) ou mettre en place d'autres choses plus malicieuses sans être découvert.

Une autre technique pour parcourir en cascade les privilèges est de décrypter les mots de passe d'autres utilisateurs depuis la table *SYS.USER$*. Oracle utilise des algorithmes de hashing basés eux-mêmes sur des algorithmes DES. Le principe de cet algorithme de cryptage le salt (pass aléatoire) d'un mot de passe. Dans Oracle, cependant, il y a une vulnérabilité à choisir salt, caractères non sensibles à la casse et des algos de hash vulnérables. L'accès aux tables *SYS.USER$* est lié au droit d'accès `SELECT ANY DICTIONARY`.

Les vecteurs d'attaque consistent à sniffer les communications sur le réseau, à procéder à des injections SQL ou accéder à la table *SYSTEM* (*system.dbf*) depuis l'OS hôte. Le langage PL/SQL est basé sur le langage de programmation ADA. PL/SQL permet de compiler le code en M-CODE, ensuite passé à la Machine Virtuel. Dans la version 9i il y avait la possibilité de deviner le but d'un code grâce au reverse engineering. Dans ce code, les éléments de base étaient retrouvés (structure de données, pointant sur la variable, fonction d'un type de données dans le code source). Dans la version 10g, la Table Symbol n'est plus visible. Oracle 10g R2 a de nouvelles fonctionnalités pour empaqueter par *DBMS_DLL* (fonction `CREATE _ WRAPPED`).

Même pour le système de base de données, il pouvait exister un ver. Il y a déjà un *Proof of Concept* (Preuve par concept) nommé Oracle Voyager Worm. Ce ver essaye d'effectuer certaines actions : grant DBA to PUBLIC, supprimer le trigger et créer un trigger, exécuté après login entré et accès à Google, il essaye également d'envoyer des mails avec le *Oracle password Hashes*. Par la suite, il essaye de scanner l'existence d'une autre base et tente de s'y connecter et établir un lien.

Listing 1. *Création d'un nouveau profil*

```
CREATE PROFILE paranoid LIMIT
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_LOCK_TIME 30
PASSWORD_LIFE_TIME 90
PASSWORD_GRACE_TIME 3
PASSWORD_VERIFY_FUNCTION check_the_password;
```

Listing 2. *Exemple de fonction qui vérifie le mot de passe*

```
CREATE OR REPLACE FUNCTION check_the_password
   (i_am_user_id VARCHAR2, new_magic_word VARCHAR2, old_magic_word VARCHAR2)
   RETURN BOOLEAN IS
   BEGIN
      IF length(new_magic_word) < 5 THEN
         raise_application_error(-20001, 'Your Magic Word Is Too Short!');
      END IF;
      IF NLS_LOWER(new_magic_word) IN ('password', 'drowssap') THEN
         raise_application_error(-20002, 'I will Not Accept Your Magic
               Word');
      END IF;
      RETURN TRUE;
   END;
```

Listing 3. *Fonction qui retourne la chaîne qui sera ajoutée à la requête*

```
CREATE OR REPLACE FUNCTION deny_table_rows (
   usr_schema VARCHAR2,
   usr_object VARCHAR2) RETURN VARCHAR2 AS
BEGIN
   RETURN 'user != SYS';
END;
```

Se défendre sous une base de données Oracle

La première chose à faire pour sécuriser la bdd est une restriction physique à celle-ci. Il est important d'effectuer cette étape afin de se prémunir des redémarrages ou d'un arrêt. La tendance pour l'authentification sont les dispositifs biométriques. Ceux-ci incluent les empreintes digitales, l'identification au niveau de l'iris ou une identification du visage.

La prochaine étape au niveau sécurité est de protéger les systèmes d'exploitation hôtes. Cette partie inclut la désinstallation de l'ensemble des services non nécessaires (*ftp*, *telnet* etc.), permettre l'activation du pare-feu et la mise en place d'une politique de sécurité. Avant de raccorder Oracle au réseau, il est important de contrôler les droits d'accès pour chaque fichier et répertoire. Supprimer les comptes utilisateurs qui ne sont pas indispensables, supprimer les logiciels non indispensables et les Systèmes de Detections d'Intrusions (IDS).

Désinstaller également les bannières pour empêcher la détection du système d'exploitation, avoir un anti-virus, des points de contrôles réguliers, la surveillance au niveau des logs et restreindre le nombre de super-utilisateurs. Hormis la sécurité relative aux systèmes d'exploitations, il est également important de sécuriser les stations de travail. Celles-ci doivent être sécurisées en différents niveaux selon leur type d'utilisation (Administration des Bdd, Developpement, les applications en cours). Certains vecteurs d'attaque utilisent les fonctionnalités des clients SQL comme TOAD ou SQL*Plus.

L'attaque pourrait cibler les fichiers ou les enregistrements du registre qui permettent l'exécution de code après authentification. Beaucoup de clients stockent également des mots de passe. Même si le mot de passe stocké est crypté, il est indiqué. Dans le domaine de la sécurité réseau, il est indispensable d'implémenter des restrictions physiques au réseau (Cf. Limiter l'obtention des adresses IP avec le DHCP seulement pour les adresses MAC connues). Il est important aussi de placer le serveur de Bdd derrière un pare-feu lequel doit être placé en dehors du réseau protégé et il est nécessaire d'ouvrir les ports et protocoles sécurisés. Enfin, il est recommandé d'utiliser *l'Oracle Connection Manager.*

OCM permet de sécuriser de façon significative l'accès au serveur de base de données via le réseau. Il est important également de sécuriser le Listener d'Oracle, changer les ports par défaut, utiliser le Node Filtering, filtrant les clients selon leurs adresses IP. Une des tâches courantes doit être la vérification des Logs du Listener d'Oracle. Il y a une option dans l'authentification utilisateur : l'Identification par Système d'Exploitation.

Cette option n'est plus sûre. Il n'est pas recommandé de l'utiliser, tout simplement parce qu'elle est vulnérable. Dans le processus d'authentification, il est bon de définir des droits, rôles, profils et restreindre les ressources disponibles. Les droits courants du système sont obtenus à partir de la vue *USER_SYS_PRIVS*. Les droits d'accès aux tables sont stockés dans *USER_TAB_PRIVS*.

La colonne *ADMIN_OPTION* montre, s'il est possible d'accorder des droits à un autre utilisateur. En raison du besoin de grouper les droits, groupons-les au sein du rôle. Il y a des rôles prédéfinis : *CONNECT*, *RESOURCE* et *DBA*. Il est essentiel d'en prendre soin, car le rôle *CONNECT* ne sert pas seulement à connecter l'utilisateur à la base, mais il permet aussi la création de tables, synonymes ou vues. Pour récupérer le rôle d'un utilisateur, utilisez la vue *USER_ROLE_PRIVS*. Pour protéger les ressources de la base, utilisez les profils.. *DBA_PROFILES* liste les enregistrements de la base à propos

Listing 4. *Police, qui ajoute la fonction deny_table_rows à la table sec_table*

```
BEGIN DBMS_RLS.add_policy (
      object_schema => 'sec_user',
      object_name => 'sec_table',
      policy_name => 'sec_table_policy',
      policy_function => 'deny_table_rows');
END;
```

Listing 5. *Bloque PL/SQL anonyme qui crypte la chaîne en AES 256-bit*

```
/* CRYPT IT ROUTINE IN AES 256-bit */
DECLARE
   k4y RAW (32);
   t0p_s3cr3t_3nc RAW (2000);
   t0p_s3cr3t_d3c RAW (2000);
BEGIN
   /* 256 bit key - 32 byte */
   k4y := DBMS_CRYPTO.RANDOMBYTES(256/8);
   t0p_s3cr3t_3nc := DBMS_CRYPTO.ENCRYPT (
      src => UTL_I18N.STRING_TO_RAW ('h4x0rIzN0tD34d', 'AL32UTF8'),
      typ => 4360,
      /* encryption type - DBMS.CRYPTO.ENCRYPT_AES256 + DBMS_CRYPTO.CHAIN_CBC
       + DMBS_CRYPTO.PAD_PKCS5 */
      key => k4y
   );
   t0p_s3cr3t_d3c := DBMS_CRYPTO.DECRYPT (
      src => t0p_s3cr3t_3nc,
      typ => 4360,
      key => k4y
   );
   DBMS_OUTPUT.PUT_LINE (UTL_I18N.RAW_TO_CHAR (t0p_s3cr3t_d3c, 'AL32UTF8'));
END;
```

des profils. L'administrateur crée son propre profil (Cf. Listing 1).

En outre, dans le profil, peut être déterminé le nombre d'essais que possède un utilisateur pour entrer un mot de passe avant que le compte ne soit bloqué. PASSWORD_LOCK_TIME définit le temps de blocage du compte après le nombre maximum d'essais pour entrer le mot de passe. *PASSWORD_LIFE_TIME* définit la durée de vie du mot de passe en jours. *PASSWORD_GRACE_TIME* définit le nombre de jours avant l'expiration du mot de passe lorsque Oracle affiche l'avertissement sur l'expiration du mot de passe. Il y a une possibilité intéressante : créer sa propre fonction (Cf. Listing 2) qui vérifiera le mot de passe avant son changement.

La fonction de vérification permet de contrôler la bonne longueur du mot de passe ou s'il ne s'agit pas d'un mot de passe issu d'un dictionnaire. Le profil est accordé à l'utilisateur lors de sa création ou avec la commande :

```
ALTER USER n1c3_us3r PROFILE paranoid
```

Une autre fonctionnalité au niveau de la sécurité est la restriction de l'espace dans la table. Voici un exemple d'une commande à utiliser :

```
ALTER USER n1c3_us3r 100M ON USERS;
```

D'autres étapes apportent la preuve du hacking de la base de données Oracle. L'une de ces étapes est l'installation des composants juste nécessaires. Il est recommandé d'utiliser le principe de la configuration la plus minime possible. Les options installées sont récupérées depuis la vue `V$OPTION`.

Selon les vecteurs d'attaque, il est nécessaire de se défendre contre l'intrus, vérifiant le couple usernames/passwords. Il est bon de vérouiller ces comptes par une requête `ALTER USER` hr `ACCOUNT LOCK` et/ou changer le mot de passe : `ALTER USER hr IDENTIFIED BY n1c3n3wp4ss;` Attention à ne pas donner des privilèges : `ANY`. Si ce privilège est accordé, il devient alors possible de travailler avec le Dictionnaire des Données, ce qui est à éviter. Ajouter le paramètre d'initialisation `07_DICTIONARY_ACCESSIBILITY = FALSE` étend la protection du dictionnaire des données. Ce paramètre restreind le privilège `DELETE ANY`. Il est également intéressant de ne donner à l'utilisateur que les privilèges nécessaires, rien de plus, tout comme il convient de restreindre le rôle par défaut `PUBLIC`.

Le rôle `PUBLIC` est attribué par défaut à chaque nouvel utilisateur Oracle. Dans la configuration de base, ce rôle permet de travailler avec des paquetages qui pourraient être compromis. Ceux-ci incluent : `UTL_SMTP` (pour l'envoi d'e-mails), `UTL_TCP` (pour l'utilisation de TCP/IP), `UTL_HTTP` (pour l'accès au Web), `UTL_FILE` (pour l'accès au système de fichier) et un paquet de cryptage *DBMS_CRYPTO*. Un contrôle effectif est atteint en utilisant un paramètre d'initialisation `REMOTE_OS_AUTH = FALSE`.

Pour les tâches courantes d'administration (démarrage, fermeture, sauvegarde, restauration et archivage) utilisez le rôle : `SYSOPER` (au lieu de `SYSDBA`). La base de données Oracle offre plusieurs niveaux de sécurité. Ce type de sécurité fait

À propos de l'auteur

Mikoláš Panský est employé chez Czech computer company Cleverlance Enterprise Solutions en tant que développeur de bases de données. Il est également professeur à la Charles University Faculty of Education, où il est allé après avoir passé son master d'informatique.
Contact avec l'auteur : *mikolas.pansky@gmail.com*

Sur Internet

- *http://en.wikipedia.org/wiki/Oracle_Database*,
- *http://www.oracle.com/database*,
- *http://www.red-database-security.com/whitepaper/oracle_default_ports.html*,
- *http://www.dokfleed.net/duh/modules.php?name=News&file=article&sid=35*,
- *http://www.jammed.com/~jwa/hacks/security/tnscmd/tnscmd*,
- *http://www.ngssoftware.com/squirrelora.htm*,
- *http://xforce.iss.net/xforce/alerts/id/advise82*,
- *http://www.appsecinc.com/resources/alerts/oracle/02-0013.shtm*,
- *http://www.thc.org/thc-hydra/*,
- *http://www.cqure.net/wp/?page_id=3*,
- *http://www.petefinnigan.com/orasec.htm*,
- *http://www.dba-oracle.com/t_oracle_biometrics_security.htm*,
- *http://www.databasejournal.com/features/oracle/article.php/3644956*.

Référence

- Alexander Kornbrust, 2006. *Oracle rootkits*, Hakin9 1/2006,
- Joshua Wright, Carlos Sid, 2005. *An Assesment of the Oracle Password Hashing Algorhytm*,
- Alexander Kornbrust, 2005. *Hardening Oracle Administration– and Developer Workstations*,
- William Heney, Marlene Theriault, 1998. O'Reilly – *Oracle Security*,
- David Know, 2004. *Effective Oracle Database 10g Security*,
- Integrity, 2004. *Oracle Database Listener Security Guide*,
- Pete Finningan, 2006. *How to unwrap PL/SQL*,
- Marlene Theriault, Aaron Newman, 2001. *Oracle Security Handbook*.

partie du Virtual Private Database (VPD).

Le VPD assure les briques de base de la sécurité. Ces derniers définissent des fontions PL/SQL, retournant des chaînes. Cette fonction est par la suite ajoutée à l'objet sélectionné (table, vue ou synonyme), que nous souhaiterions protéger avec le paquetage PL/SQL DBMP_RLS.

Si une requête SQL est produite, Oracle ajoute en fin de requête la chaîne résultante de la fonction définie. Cette fonction peut ensuite être une restriction, supprimant des lignes, et qui contient dans la colonne utilisateur la valeur : SYS (Cf. Listing 3).

La règle, qui assure, que la réponse de la requête `SELECT` ne contient pas certaines lignes, est également définie par le paquetage : `DBMS _ RLS` (Cf. Listing 4). Pour de plus amples informations, Cf. l'article sur le VPD à : *www.databasejournal.com*.

Il y a plusieurs raisons au cryptage des données d'une base : par exemple, se protéger et cacher les informations contre les administrateurs : *DBA* ; ou atteindre une sécurité standard. Pour crypter les données, utilisez *DMBS_CRYPTO* (il devrait remplacer : *DBMS_OBFUSCATION_TOOLKIT*). *DBMS_CRYPTO* est orienté sur le travail avec des types de données RAW.

Cela n'empêche pas la possibilité de convertir `VARCHAR2` vers du `RAW` et vice-versa avec le pack *UTL_RAW*. Ce paquetage offre DES (qui n'est par ailleurs plus recommandé aujourd'hui), triple-DES avec 3 clés, AES avec plusieurs clés de tailles différentes et des algorithmes RC4. Le Listing 5 montre un exemple d'un cryptage AES de 256-bit avec *Cipher-Block-Chaining* selon le standard PKCS#5 (Cf. RFC 2898).

Conclusion

Je voulais vous présenter au travers de cet article des concepts de base en sécurité des Bases de Données Oracle de 2 points de vues différents: l'attaque et la défense. ●