CHAPITRE II:

Sécurité dans les bases de données

1. Introduction

La préservation de la confidentialité est devenue une priorité pour les citoyens ainsi que pour les administrations. Le besoin d'accumuler, de partager et d'analyser des données personnelles est multiple : pour l'amélioration de la qualité des soins grâce au dossier médical électronique pour rendre plus simples et efficaces les procédures administratives, pour personnaliser les services rendus par une grande quantité d'objets éle²ctroniques dans un environnement d'intelligence ambiante ou même pour la lutte contre le terrorisme (croisement de bases de données commerciales et gouvernementales pour la recherche de suspects). Bien que le traitement de données personnelles ait généralement un but louable, il constitue une menace sans précédent aux droits élémentaires à la protection de la vie privée.

Partout dans le monde, les gouvernements adoptent des lois spécifiques pour cadrer l'utilisation de données personnelles comme le 'Federal Privacy Act' aux USA ou la directive pour la protection des données en Europe. Il est cependant difficile de traduire ces lois en moyens technologiques convaincants garantissant leur application.

Comme l'atteste le rapport 'Computer Crime and Security Survey' établi par le Computer Security Institute et le FBI, le nombre d'attaques de serveurs de bases de données est croissant malgré la mise en place de politiques de sécurité de plus en plus drastiques. Pire encore, presque la moitié de ces attaques sont conduites par des employés ayant légalement accès à tout ou partie des données. Ceci montre la vulnérabilité des techniques traditionnelles de sécurisation des serveurs bases de données.

2. Sécurité

Bien que la sécurité soit l'une des raisons de l'architecture trois couches, plusieurs challenges praticables sont enlevés lors de la construction du système tel que l'authentification des utilisateurs, le contrôle des accès et Audit les actions des utilisateurs, la protection des données entre les couches, la limitation des privilèges de l'intermédiaire, et la construction des systèmes extensibles.

2.1. Propriétés principales

La sécurité des bases de données inclus trois principales propriétés : **la confidentialité**, **l'intégrité** et **la disponibilité**.

- **Confidentialité :** L'information protégée ne doit pas être accessible aux utilisateurs ou un programme non autorisés. C'est crucial:
  - ❖ Dans des environnements critiques ou stratégiques: militaires ou commerciaux, par exemple.
  - ❖ Pour respecter le droit des individus à décider comment et dans quel but les informations les concernant peuvent être extraites, mémorisées ou transmises à d'autres individus.

- **Intégrité :** Les données ne peuvent être modifiées que par les utilisateurs habilités à le faire qu'elles soient dues à :
  - ❖ Des pannes de système,
  - ❖ Des manipulations erronées,
  - ❖ Des sabotages.

- **Disponibilité :** Il s'agit de détecter ou d'empêcher des **dénis de service**.

Il y a déni de service lorsqu'un utilisateur ne parvient pas à accéder dans un **délai raisonnable**, à une information ou à une ressource pour laquelle il a une autorisation d'accès. **[13]**

2.2. Processeur de sécurité

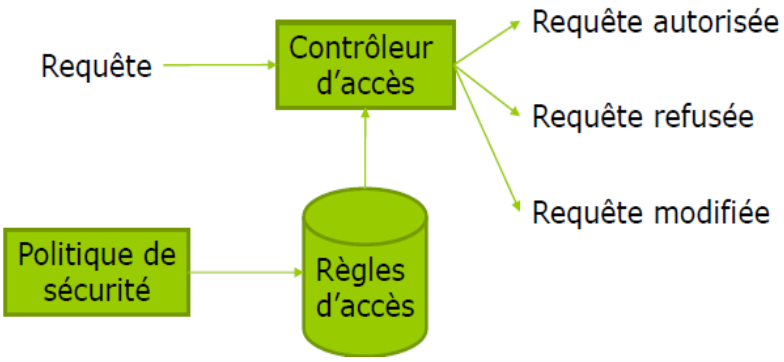

Figure II.1: Processeur de sécurité. [13]

Autorisation, interdiction et obligation :

- Les règlements les plus simples ne contiennent que des **autorisations**: ce qui n'est pas autorisé est interdit.
- Certains règlements incluent des **interdictions** à fin de spécifier des exceptions à des permissions générales. Exemple: les patients ont droit de consulter leur dossier médical sauf Jean Dupont.
- D'autres enfin, plus sophistiqués, incluent des **obligations**: difficiles à implanter dans les systèmes informatiques. **[13]**

2.3. Attaque

- Les violations de la sécurité d'une BD consistent en des lectures ou des mises à jour illicites.
- Les événements qui portent ces violations sont appelés des **attaques**.

Les attaques à une BD peuvent exploiter les failles des applications opérant sur cette BD:

- ❖ Stockage des mots de passe dans les fichiers de configuration de l'application,
- ❖ Scripts de connexion à la BD accessibles dans le code source de l'application,
- ❖ Attaques par injection SQL,
- ❖ Attaques exploitant les débordements de tampons. **[13]**

2.3.1. Types d'attaques

On distingue:

- Les attaques **non frauduleuses**:
  - ❖ Catastrophes naturelles,
  - ❖ Pannes de logiciel ou de matériel,
  - ❖ Erreurs humaines…
- Les attaques **frauduleuses**:
  - ❖ Utilisation abusive de leurs droits par les utilisateurs,
  - ❖ Agents hostiles exécutant des actions de destruction du logiciel ou du matériel, ou lisant ou mettant à jour des données protégées,
  - ❖ Ces agents peuvent être cachés dans des actions légales : **chevaux de Troie**. **[13]**

2.3.2. Qui attaque?

Dans cette partie on présente les différents attaquants :

- **Pirate externe :** il est capable de s'infiltrer sur le serveur BD et de lire ses fichiers, il peut aussi casser une clé de chiffrement avec un texte connu.
- **Pirate utilisateur :** ce type de pirate est reconnu par le SGBD et à accès à une partie des données suivant le mode de chiffrement, il a accès à certaines clés.
- **Pirate administrateur (DBA) :** employé peu scrupuleux ou pirate s'étant octroyé ces droits ; A accès à des données inaccessibles aux autres pirates (journal) et aussi peut espionner le SGBD pendant l'exécution. **[18]**

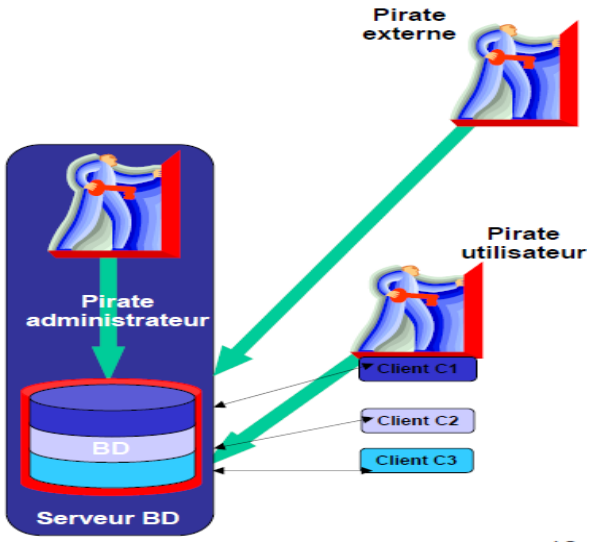

Figure II.2: Différents attaquants. [18]

2.3.3. Les risques encourus

Les risques propres à une source de données sont les suivants :

Le **vol de données** induit la perte de **confidentialité** des données stockées. La divulgation de données financières hautement confidentielles peut avoir un impact néfaste sur l'activité d'une entreprise : risque juridique, atteinte à l'image de marque, perte de confiance des partenaires industriels...

L'**altération de données** induit une perte d'**intégrité**, c'est-à-dire que les données ne sont plus dignes de confiance. En fonction de la rapidité de détection et de la qualité des sauvegardes, les conséquences peuvent en être réduites. Mais une application fonctionnant sur des données falsifiées peut voir son comportement fortement influencé : par exemple, un site de commerce électronique pourrait débiter le compte d'un autre client que celui réalisant la commande !

La **destruction de données** remet sérieusement en cause la **continuité de l'activité** de l'entreprise concernée. Privée de ses données clients, sans sauvegarde, c'est le dépôt de bilan garanti !

L'**augmentation du niveau de privilèges** d'un utilisateur d'une application est plus insidieuse que les risques précédents, car comme pour l'altération de données, il n'est remarqué qu'après un certain laps de temps durant lequel le pirate peut réaliser un grand nombre d'actions malveillantes. Il peut ainsi s'attribuer le droit d'accès à des informations confidentielles, le droit d'accès à des opérations sensibles, voire même prendre le contrôle d'une application.

Selon le SGBD utilisé, des **ressources systèmes** peuvent être attribuées à chaque utilisateur (nombre de requêtes par unité de temps...). Ces ressources peuvent être limitées par l'administrateur système afin d'éviter l'écroulement des capacités de traitement du serveur (**déni de service**) par un utilisateur malveillant. De plus, ceci permet de limiter la portée d'une attaque par altération ou vol de données en limitant le nombre d'opérations réalisables en un temps donné. La conséquence d'un tel risque peut être la paralysie du serveur (perte de **disponibilité**).

On le voit, les risques sont variés et leurs conséquences potentiellement dramatiques. Ainsi, il est nécessaire d'attribuer les droits d'accès avec parcimonie. **[21]**

2.4. Les types d'utilisateurs

Il faut identifier les utilisateurs ayant besoin d'un accès à la base de données, ils peuvent être de différents types :

L'**administrateur** est une personne physique ayant tous les droits sur le SGBD, mais pas forcément sur le contenu des bases de données : il peut réaliser des opérations de gestion des droits d'accès et des ressources systèmes mais on pourra choisir d'exclure ou non les droits d'accès en lecture et/ou écriture au contenu des bases de données. Bien que parfaitement logique d'un point de vu métier, pour la protection de données sensibles par exemple, retirer à un administrateur les droits de lecture et d'écriture sur le contenu d'une base de données n'a pas de sens d'un point de vu technique puisqu'il possède les capacités techniques de s'octroyer ses droits là. De plus, les opérations de

sauvegarde, de restauration et de maintenance après incident peuvent l'amener à devoir accéder au contenu d'une base de données.
Bref, normalement, c'est l'utilisateur qui a tous les droits sur le SGBD et les bases de données hébergées. C'est normalement une personne de confiance, compétente et prudente.

Une **application** peut être une application web, un outil de synchronisation entre sources d'informations ou tout programme accédant pour lui-même à la base de données. Ce type d'utilisateur logique n'a rien à voir avec l'utilisateur réel dénotant une personne physique ayant des besoins particuliers. Même si une application est utilisée par des personnes physiques, on pourra choisir de déléguer à l'application la gestion des droits d'accès à l'information en fonction des habilitations qu'elle décide de lui attribuer. Ainsi, une application peut être vue comme un utilisateur de base de données auquel on attribue des droits qu'elle pourra restreindre de façon transparente pour l'utilisateur final de l'application ainsi que pour le SGBD.

L'**utilisateur** est une personne physique se connectant directement à la base de données (commande *mysql* sous Linux) ou via une interface graphique (script *phpMyAdmin* sur un Intranet) ou utilisant une application qui va se connecter à la base de données sous l'identité de l'utilisateur (client lourd *MySQL Query Browser*). **[21]**

2.5. Politique de sécurité

Les ITSEC (**Information Technology Security Evaluation Criteria)** définissent la politique de sécurité comme l'ensemble des lois règles ou pratiques qui régissent la façon dont l'information sensible et les autres ressources sont gérées, protégées et distribuées à l'intérieur d'un système d'information. **[13]**

3. Les moyens de sécurité

- **Protections contre les attaques**

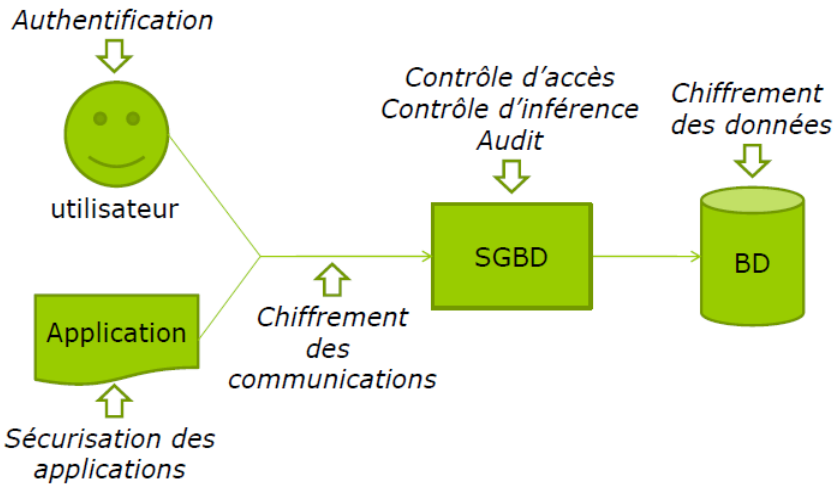

Figure II.3: Protections contre les attaques. [13]

Les SGBD fournissent différents moyens pour garantir la sécurité :

3.1 Vues

La base de données du fournisseur SQL Server comprend des vues prédéfinies qui permettent d'accéder aux données d'une fonctionnalité particulière sans accéder directement aux tables de base de données. L'accès aux vues fournies est en lecture seule. Vous ne devez pas essayer de mettre à jour les données de la base de données à partir des vues.

Importance des vues : elles permettent de définir de façon précise les portions d'une BD sur lesquelles des privilèges sont accordés. **[22]**

3.2 L'authentification des utilisateurs

- L'**authentification** a pour objectif d'assurer que l'utilisateur qui se connecte à la BD est:
  - ❖ Autorisé à se connecter,
  - ❖ Bien celui qui s'annonce.

- L'authentification repose sur :
  - ❖ La sécurité des mots de passe,
  - ❖ Des techniques d'identification biométriques. **[13]**

Dans les architectures à deux niveaux où les utilisateurs se connectent directement au serveur, la base de données fait authentifiée les utilisateurs lors de la connexion et leur associée le nécessaire pour travailler. Typiquement, dans les architectures trois tiers, l'intermédiaire est le responsable de l'authentification des utilisateurs, et de plus il traite les données envoyées par les utilisateurs avant de les transmis vers le serveur de base données. Un seul intermédiaire est commun entre les utilisateurs et la base de données (un seul point d'entrée), pour cette raison la base de donnée délègue l'authentification des utilisateurs à l'intermédiaire, dans ce niveau la réalisation d'une authentification est plus compliquée que l'architecture deux tiers. **[14]**

3.3 Le contrôle d'accès des utilisateurs

Afin de permettre l'implantation de politiques de confidentialité et d'intégrité en leur sein, les systèmes d'exploitation disposent de mécanismes de contrôle d'accès. Typiquement, ceux-ci fonctionnent sur le modèle suivant :

- Un ***sujet*** est une entité active inclut souvent les utilisateurs et les processus travaillant pour le compte des utilisateurs (qui peuvent être classés par groupes).
- Un ***objet*** est une entité passive, un conteneur d'information à protéger, sur lequel un sujet peut effectuer une action (les fichiers, Données, programmes, périphériques matériels) ;
- Une ***permission*** est une certaine action permettent aux sujets de manipuler les objets. Une permission peut être accordée ou refusée (par exemple, lecture, écriture, exécution).
- Un ***règlement de sécurité*** constitué d'un ensemble de **règles d'accès** traduisant la **politique de sécurité** du système d'information.
- Un ***processeur de sécurité*** qui vérifie que les requêtes adressées au système ne violent pas les règles d'accès et selon le cas autorise, modifie ou interdit la requête. **[13,20]**

Le contrôle d'accès est configuré par un ensemble de règles spécifiant un sujet, un objet et des droits d'accès. Un cas particulier de règles est celui spécifiant

une action d'un sujet vers un autre sujet (envoi de signal ou de message inter-processus).

Une fois que l'utilisateur est authentifié par l'intermédiaire, le système doit contrôler quelles données, applications et ressources l'utilisateur peut accéder dans le système. Les données ne doivent pas être protégées seulement contre les intrusions mais aussi les accès des utilisateurs ayant des limites qui doivent être respecté. Pour contrôler l'accès il faut d'abord renforcer la manière dont les utilisateurs font accès.

La sécurité typique dans les systèmes trois tiers, exige que chaque utilisateur soit limité par l'exécution des applications spécifiques sur l'intermédiaire, dépendant sur l'identité de l'utilisateur et le rôle lui accorder dans l'organisation. Un utilisateur qui accède à partir d'un intermédiaire ne doit pas avoir la permission d'accéder directement à la base de données**. [14]**

3.3.1 Modèles de contrôle d'accès

Définition : une politique de contrôle d'accès est un ensemble de règles.

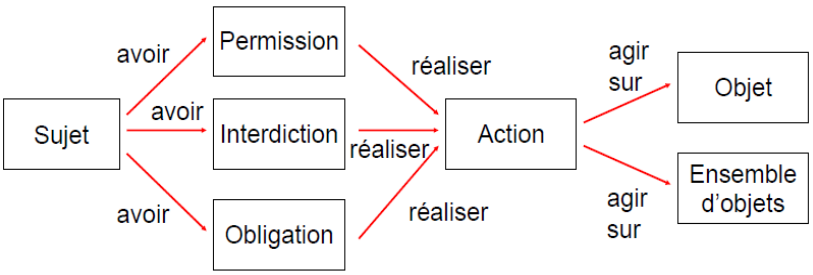

Figure II.4 : Format des règles. [18]

Les modèles de contrôle d'accès permettent de définir le cadre d'expression d'une politique de sécurité.

Cette partie on présente l'état de l'art des principaux modèles de contrôle d'accès statiques : DAC, MAC et RBAC.

Les parties suivantes présentent les différentes familles de modèles théoriques. On distingue principalement trois grandes catégories de modèles de contrôle d'accès :

❖ Modèles discrétionnaires

Contrôle d'accès discrétionnaire; L'administration des droits dans le modèle DAC repose sur la notion de propriétaire : chaque objet a un propriétaire qui décide quels sont les autres sujets qui ont accès à cet objet, étant donné que l'attribution des droits est faite par les utilisateurs et non pas par les administrateurs.

Voici les principes d'accès qui s'appliquent au modèle "discrétionnaire" :

- Un sujet dispose du plein contrôle des objets dont il est propriétaire.
- Le propriétaire est souvent le créateur des objets.
- Le propriétaire détermine les permissions et droits d'accès aux ressources sous son contrôle.

Le plus célèbre modèle discrétionnaire est le modèle **HRU** qui a été défini en 1976 par Harrison, Ruzzo et Ullman. Il s'agit d'un modèle matriciel défini à partir d'un ensemble de sujets, d'un ensemble d'objets et d'un ensemble fini de règles. Ces règles décrivent dans quelles conditions un sujet peut modifier le contenu de la matrice d'accès.
Typiquement, les droits d'accès sur un fichier sont positionnés par l'utilisateur déclaré comme propriétaire de ce fichier. **[19]**

Avantages et inconvénients

- Ils sont simples à mettre en œuvre.
- Ce sont les plus implantés: UNIX, SQL…
- Le contrôle de la propagation des droits et celui de la révocation des droits propagés posent des problèmes difficiles.
- Ils sont vulnérables aux **chevaux de Troie**.

❖ Modèles obligatoires

Contrôle d'accès obligatoire Contrairement au DAC, le MAC ou *Mandatory Access Control* délègue l'attribution des permissions à une entité tierce, typiquement un administrateur externe de la politique de sécurité. Ainsi, les utilisateurs du système ne peuvent pas intervenir dans l'attribution des

permissions d'accès, même s'ils disposent de droits d'administration dans le système d'exploitation. **[20]**

Ce module représente les mécanismes de contrôle d'accès implémentés par l'équipement auprès de l'agent de contrôle d'accès. Ce module reçoit les règles de contrôle d'accès de la part du module de configuration du contrôle d'accès (MCCA). Ces règles sont traduites par le MCA dans les syntaxes réelles des commandes nécessaires à la configuration des mécanismes implémentés. En retour de ces commandes le MCA reçoit des résultats correspondant à l'application des commandes et transmet ces résultats au module de gestion centralisée du contrôle d'accès (MGCCA). Ils sont des modèles **multi-niveaux**.

- Il s'agit de protéger le secret et l'intégrité.
- Les sujets (ou utilisateurs) et objets sont classés par niveaux (ex: Top secret, Secret, Confidentiel, Public).

1 . Modèle de Bell et LaPadula (BLP)

Le modèle de Bell et LaPadula a pour objectif la protection du secret des informations. Il est basé sur la classification des **objets** et des **sujets** par **niveaux de secret**.

L'ensemble des niveaux de secret est muni d'un ordre partiel (>).

Par exemple : Top secret (TS) > Secret (S) > Confidentiel (C) > Non classifié (NC).

Politique de sécurité : La politique de sécurité du modèle de Bell et LaPadula se résume dans les deux principes :

- **No Read Up Secrecy** : préserve de la lecture d'une information dans un objet par un sujet de niveau de secret inférieur.
- **No Write Down Secrecy** : préserve d'un transfert d'information d'un objet vers un autre objet de niveau de secret inférieur, par un utilisateur qui n'est pas de confiance. **[13]**

2. Modèle de Biba :

Le modèle de Biba (K. J. Biba, 1977) a pour objectif la protection de l'intégrité des informations. Il applique à la protection de l'intégrité une stratégie similaire à celle de la protection du secret par le modèle BLP : les sujets et les objets sont classés par **niveaux d'intégrité**.

L'ensemble des niveaux d'intégrité est muni d'un ordre partiel (>).
Par exemple : Crucial (TS) > Très important (TI) > Important (I) > Non classifié (NC)

Politique de sécurité : La politique de sécurité du modèle de Biba se résume dans les deux principes :

- **No Write Up Integrity** : préserve de l'écriture d'une information dans un objet par un sujet de niveau d'intégrité inférieur.
- **No Read Down Integrity :** préserve d'un transfert d'information d'un objet vers un autre objet de niveau d'intégrité supérieure par un utilisateur qui n'est pas de confiance.

Avantages et inconvénients

- Ils sont bien adaptés aux applications où la protection du secret et de l'intégrité est primordiale.
- Mais ils sont trop rigides et centralisés.
- La confidentialité est assurée au détriment de la disponibilité. **[13]**

❖ Modèles à base de rôles

Le modèle RBAC pour *Role-Based Access Control* a pour but de simplifier l'administration des droits d'accès des utilisateurs individuels en fournissant un niveau d'indirection supplémentaire. Plutôt que de donner directement des permissions aux utilisateurs, on définira différents rôles possibles pour l'utilisation du système d'exploitation, avec des droits d'accès associés. Ensuite chaque utilisateur a accès à une liste de rôle, suivant son activité. Un seul rôle peut être actif à un moment donné. **[20]**
On peut améliorer les modèles discrétionnaires en créant des rôles qui sont des ensembles d'autorisation. Les autorisations sont octroyées aux rôles et les rôles aux utilisateurs.

Pour pouvoir réaliser une action sur un objet un utilisateur doit ouvrir une session et activer celui de ses rôles qui contient l'autorisation de réaliser cette action.

Un rôle peut hériter des privilèges d'un autre rôle ; Sont bien adaptés aux applications de gestion. Permettent une administration souple des privilèges. **[13]**

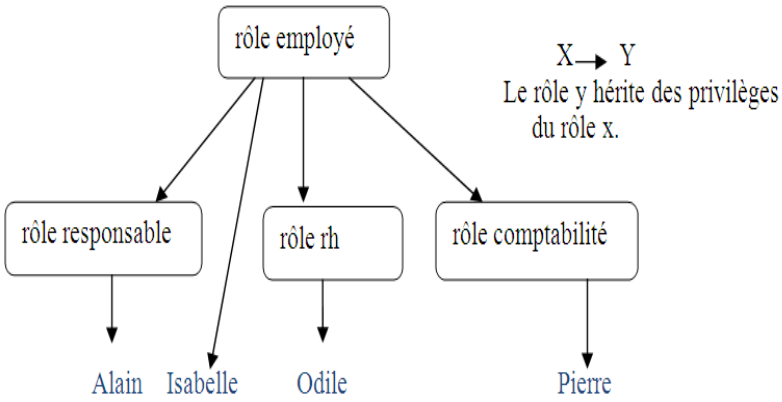

Figure II.5: Exemple d'héritage de rôle.

Les modèles de contrôle d'accès comme DAC, MAC ou RBAC ne permettent de modéliser que des politiques de sécurité qui se restreignent à des permissions statiques. Ils n'offrent pas la possibilité d'exprimer des règles contextuelles relatives aux permissions, aux interdictions, aux obligations et aux recommandations. Ce type de règle est particulièrement utile pour exprimer des politiques de sécurité dans le domaine médical. Dans l'article **[17]**, les auteurs proposent un nouveau modèle qui permet de spécifier de telles politiques de sécurité contextuelles. Ce modèle appelé Organisation Based Access Control (ORBAC) s'appuie sur un langage formel basé sur la logique du premier ordre L'approche ORBAC pour sécuriser l'interopérabilité repose sur la création de politiques de sécurité d'interopérabilité.

3.4 Protection des données de l'utilisateur

Le changement des données entre les trois tiers doits être protégé contre les révélations et les modifications non attendus, le cryptage est le mécanisme standard pour ce but. Le SSL (Secure Sockets Layer) est le protocole qui

assurer le cryptage et la communication confidentielle dans le réseau entre le client et l'intermédiaire aussi qu'entre l'intermédiaire et la base de données. **[14]**

La sécurisation des données permet de crypter ou de limiter la vision des données pour un utilisateur au niveau d'une base, d'une table, d'une colonne. La norme SQL ANSI permet déjà à travers la gestion des droits (GRANT, REVOKE) d'autoriser un utilisateur à voir ou modifier des informations sur une table, ou sur certaines colonnes. Par contre les données sont stockées en clair, non cryptées.

Dans le cas de données sensibles, il est important de proposer des solutions de cryptage des données. Pour répondre à ces problématiques, les éditeurs de SGBD proposent essentiellement deux solutions :

• La première consiste à utiliser des fonctions de cryptage directement dans le code SQL. L'emploi des fonctions de cryptage oblige à une modification du code applicatif. Les principaux algorithmes disponibles suivant les SGBD sont DES, 3DES, AES, MD5, MD4, SHA et SHA-1.

• La deuxième consiste à mettre en place un système de cryptage connu sous le nom générique de TDE (Transparent Data Encryption). Cette solution de cryptage est transparente pour les applications (Il n'y a pas besoin de modifier le code applicatif). En général, cette solution permet de protéger les fichiers de la base ainsi que les sauvegardes.

En ce qui concerne TDE (Transparent Data Encryptions), actuellement seul Oracle, SQL Server et Sybase proposent ce mode de cryptage. Sous Oracle la fonctionnalité **TDE** qui fait partie de l'option **Oracle Advanced Security,** permet d'effectuer un cryptage au niveau de la colonne ou du *tablespace*.

Pour cela, Oracle utilise une clé externe (master key) qui peut être stockée dans un Oracle wallet (protégé par mot de passe). Si un utilisateur accède à la base, les données seront décryptées automatiquement. **[16]**

3.5 Audit des accès de l'utilisateur

L'audit des accès est nécessaire pour déterminer l'utilisateur responsable de telle action dans la base de données, et comme les utilisateurs accèdent à partir d'un intermédiaire, il est difficile à un système audit de garder la trace et de corréler les activités qui peuvent être sensitives à la sécurité. **[14]**

Un outil indispensable pour assurer la sécurité d'une BD est l'**audit** basé sur un journal des différents types d'accès à la BD :

- Audit des entrées dans la base.
- Audit des utilisations de la BD en dehors des heures ouvrables.
- Audit de la manipulation du schéma.
- Audit des erreurs.
- Audit des modifications des sources des procédures stockées et des triggers.
- Audit des modifications des attributs de sécurité (login, privilèges...). **[13]**

3.6 Limitation du privilège de l'intermédiaire **[21]**

3.6.1 Principe du moindre privilège

Ce Principe stipule qu'un sujet ne doit disposer que des droits d'accès minimum pour assurer l'exécution des tâches qui lui sont assignées, pas un de plus.

Ex : ne pas donner les droits de l'administrateur à tout utilisateur d'un système (système d'exploitation, SGBD).

Puisque le mécanisme d'authentification de l'intermédiaire est moins fort que celui de la base de données et que l'intermédiaire situé en d'hors de la zone protégée par le firewall en face des intrusions interviennes de l'Internet, la base de données doit limiter les privilèges d'un intermédiaire, et de le permis d'accéder au nom des utilisateurs spécifiques. **[14]**

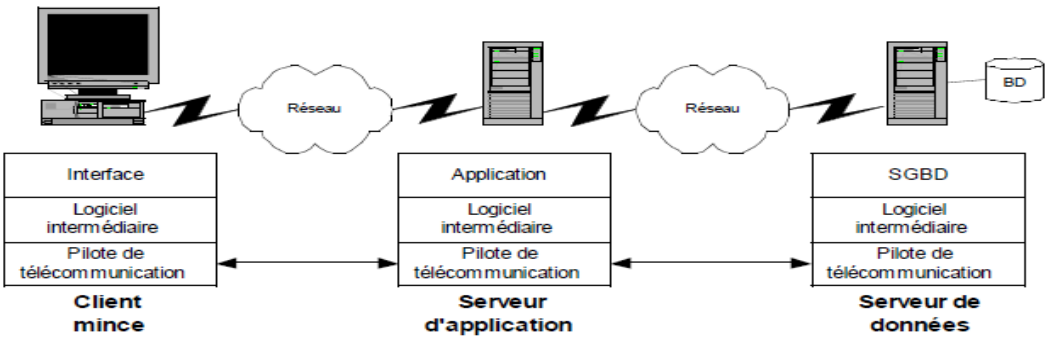

Figure II.6: Architecture à trois tiers. [14]

3.6.2 Politique de gestion des privilèges

A. Les privilèges

Il convient pour chaque compte d'accès d'identifier les privilèges minima à accorder ainsi que le niveau de granularité adéquat.
Ici ; le terme *utilisateur* désigne une application aussi bien qu'une personne physique.

A.1. Classes d'objets et granularité

Les SGBD permettent généralement de spécifier assez finement les privilèges d'un utilisateur en fonction des objets manipulé :

- Base de données.
- Table (relation).
- Colonne (attribut).

Ainsi, un utilisateur peut se voir attribuer un privilège pour toute une base de données, ou seulement pour quelques tables, ou encore sur uniquement quelques colonnes de certaines tables.

A.2. Classes de privilèges

Les privilèges s'organisent autour de plusieurs classes :

- Accès au contenu de l'information.
- Gestion du schéma de la base de données.
- Gestion des privilèges utilisateurs.
- Gestion des paramètres systèmes.

B. Règles d'attribution des privilèges

Règle fondamentale n°1 : attribution du moindre privilège.
Les utilisateurs ne doivent avoir que le minimum de droits, ceux strictement nécessaires à l'accomplissement de leurs tâches.

Règle n°2 : contrôle de la population.

Les privilèges doivent êtres synchrones avec la réalité de la population : il faut supprimer les comptes des utilisateurs quittant l'entreprise et de ceux n'étant plus affectés à telle ou telle tâche.

Règle n°3 : supervision de la délégation des tâches d'administration.

Un administrateur peut être amené à déléguer auprès d'une autre personne les tâches d'attribution des privilèges de tout ou partie de la population des utilisateurs. Un contrôle *a posteriori* doit être réalisé afin de vérifier que le résultat de cette délégation est conforme à la politique adoptée.

Règle n°4 : contrôle physique des connexions.

Il est nécessaire de restreindre les connexions à des hôtes spécifiques connus.

Règle n°5 : limitation des ressources utilisées.

Le SGBD offre souvent la possibilité de restreindre les ressources de calcul disponibles pour un utilisateur. Il est recommandé de configurer ces limitations de ressources en fonction de la charge maximale attendue pour un utilisateur.

Règle n°6 : journaliser les comportements suspects.

Certains SGBD permettent de conserver dans des **journaux de log** les requêtes non conformes aux privilèges accordés à un utilisateur. Il peut être intéressant de les surveiller afin de détecter toute anomalie dénotant des tentatives de piratage.

Règle n°7 : restrictions sur une application en fonction du public.

Une même application web peut avoir plusieurs interfaces différentes selon le contexte d'utilisation : internet / intranet.

C. Contrôle des privilèges

La principale question qui se pose lors du développement d'une application, c'est quelle stratégie adopter vis à vis des utilisateurs : contrôle de leurs droits d'accès par l'application ou par le SGBD ?

Par le SGBD. Dans le cas où toute l'information métier repose sur une base de données comportant également toutes les procédures stockées de contrôle de l'intégrité, de la logique métier et des actions utilisateurs, il est logique de déléguer au SGBD le contrôle d'accès et les habilitations. Ceci suppose que l'administrateur de bases de données réalise les opérations d'attribution des

privilèges et de synchronisation avec l'annuaire des utilisateurs du système d'information de l'entreprise. L'application ne devient alors qu'une interface graphique ergonomique d'interrogation de la base de données métier.

Par l'application. Dans le cas où l'application gère elle-même le niveau d'accréditation des utilisateurs, elle va se connecter sous sa propre identité logique à la base de données et décider des informations et des opérations que l'utilisateur peut voir, modifier et réaliser. C'est la stratégie employée par les applications dont la logique métier n'est pas intégrée directement dans la base de données et qui gèrent plusieurs sources de données.

D. Contrôle d'inférence

L'objectif du **contrôle d'inférence** est protéger une BD des attaques consistant à déduire des données non autorisées à partir de données autorisées.

Ex : Interdire l'accès à des données individuelles dans une BD statistiques à partir de requêtes agrégatives (comptage, somme, moyenne).

3.7 Base privée virtuelle (Virtual Private Database ou VPD)

A l'incrémentation des besoins des organisations et des entreprises, la base privée

virtuelle proposée par oracle pourvoi une grande sécurité d'accès contrôle. Et ceci par la création d'une politique de sécurité -au niveau de la base de données- commun entre toutes les applications, ce qui permet à chaque utilisateur d'accéder seulement à ces données, et quelque soit la manière de la quelle l'utilisateur accède à la base il ne peut pas dépasser cette politique de sécurité. **[14]**

Principe

- Associer une règle de sécurité à une table ou une vue.
- Si la VPD est activée, toute requête qui accède à cette relation ou cette vue est automatiquement modifiée en incluant une clause WHERE.

4. La protection d’une base de données

Selon la référence **[15]** la protection d’une base de données suit les étapes suivantes :

4.1 Connaître son besoin

La sécurité de la base de données commence par une réflexion sur les usages et la population d'utilisateurs accédant à celle-ci, ainsi que sur la manière dont la connexion s'effectue. Est-ce directement par les utilisateurs ou par le biais d'un applicatif (interface Web, progiciel, etc.). Il est indispensable de connaitre la méthode et la nature des accès afin de définir une politique de sécurité adaptée.

"La connexion d'un SGBD avec un progiciel, qui nécessite une méthode d'interconnexion spécifique, peut avoir pour effet d'abaisser le niveau de sécurité. Les équipes sécurité et intégration, dont les missions ne sont pas forcément en accord, doivent souvent trouver un accord".

La sécurité ce n'est cependant pas uniquement la protection contre les attaques. D'autres questions se posent sur la garantie de la traçabilité, l'intégrité, l'audibilité, la confidentialité et la sauvegarde des données. La politique va par conséquent dépendre de ce que l'on souhaite garantir en fonction des besoins identifiés. Il sera ainsi incontournable de redonder les équipements d'interconnexion si la disponibilité est critique.

4.2. Une sécurité en amont

Le déploiement d'une base de données est souvent la brique d'un projet plus global. La sécurité doit donc être pensée pour l'ensemble des éléments, surtout dans le cas d'un applicatif accédant à la base. Celle-ci peut être protégée mais si l'outil utilisé pour s'y connecter est vulnérable, il ouvrira des portes. Un SGBD ne pourra pas faire la différence entre une connexion légitime et une attaque par le biais d'un frontal Web.

4.3. Supervision

Un suivi des indicateurs de la base de données doit être assuré afin de détecter les anomalies, prévenir les interruptions de service et intervenir dans les meilleurs délais. La majorité des SGBD du marché embarquent désormais des systèmes de supervision. Charge ensuite à l'administrateur de base de données (DBA) de concevoir des filtres appropriés pour diagnostiquer toute évolution du mode de fonctionnement de la base.

4.4. Sensibiliser les DBA

L'administrateur doit être sensibilisé aux problématiques de sécurité, aux risques, à la criticité des contenus dont il a la charge et pas seulement à la performance. Un DBA peut avoir à superviser une dizaine de bases sans bénéficier de visibilité sur les données qu'elles hébergent et risquer par conséquent de ne pas avoir les bons réflexes.

4.5. Durcir le socle système

Une base de données repose sur une couche système. Cette dernière ne doit donc pas être négligée et faire l'objet d'un durcissement fort. Une base de données ne sera pas en mesure de se défendre contre une personne détenant des droits administrateur sur l'OS. Ce durcissement comprend l'application d'une politique de gestion des correctifs et du moindre privilège, la limitation des services (réseau et système) et applicatifs, la segmentation des droits ou encore une authentification via des mots de passe fort. Attention au paramétrage des SGBD lors de migration de versions.

4.6. Renforcer la couche BD

Tout comme le système, les correctifs de sécurité doivent être appliqués à la base de données. Pour des exigences de disponibilité, le patch management est cependant complexifié. Il faut veiller en outre au durcissement de l'installation par défaut.

4.7. Gestion des comptes

Les comptes par défaut doivent être verrouillés et les mots de passe remplacés pour respecter les normes de sécurité. La notion de politique du moindre privilège s'applique. C'est-à-dire qu'un utilisateur n'ayant par exemple besoin que de consulter les données ne doit en aucune façon disposer de droits en écriture. De même, la base doit être correctement segmentée pour qu'une habilitation ne concerne qu'un périmètre défini des données.

4.8. Méthodes d'accès

L'entrée sur la base de données doit être autorisée selon des méthodes précises. C'est à ce niveau que le filtrage sera défini. Si la connexion se fait depuis une application Web, alors seule celle-ci et le DBA seront autorisés à accéder. Ce filtrage est toutefois complexifié lors de l'intégration avec un PGI ou de connexion depuis une application en client lourd installée sur de nombreux postes.

Interviennent alors des aspects de gestions des profils et des utilisateurs, d'évolution des droits. Une cartographie des données et des habilitations doit être dressée pour définir les types de populations accédant à la base et les parties de celle-ci qu'ils sont autorisés à consulter.

4.9. Chiffrer les flux de données

Les informations envoyées en réponse à une requête ne doivent pas circuler en clair sur le réseau. Nul besoin de durcir l'accès et l'OS, s'il suffit d'écouter le trafic réseau. Les flux seront par conséquent chiffrés entre la base et les différents composants.

5. Conclusion

La sécurité des accès à une base de données est une préoccupation de tous les instants. Les privilèges doivent être restreints à l'indispensable et être actualisés régulièrement. Quant aux applications web, vulnérables par essence, elles doivent être développées de manière à réduire le risque d'accès frauduleux aux données. Une collaboration étroite entre administrateur de base de données et développeur

web permet de réduire considérablement les risques liés à la sécurité des bases de données. Les éléments de sécurité existant ne sont pas suffisants.