Gestion des droits d'accès

- Menaces
- Modèles de contrôle d'accès
- Contrôle d'accès dans SQL
- Chiffrement de bases de données

Où/par qui nos données sont-elles stockées ?

- Des données disséminées partout, parfois silencieusement
  - SI nationaux (Impôts, fichiers de police, BaseEcole, DMP …)
    - Les plus encadrés car focalisent toute l'attention
  - SI d'entreprise (employeurs, assurances, EDF …)
  - BD personnelles (factures, fichiers perso …)
  - BD "ambiantes" (Pass Navigo, Telco, télé-surveillance, log de requêtes Web …)
- Des données bien organisées
  - Données centralisées, structurées, cohérentes, à jour
  - Facilement accessibles, exploitables, croisables
  - donc intéressantes !
  - Et ce qui est intéressant pour le gestionnaire l'est pour l'attaquant

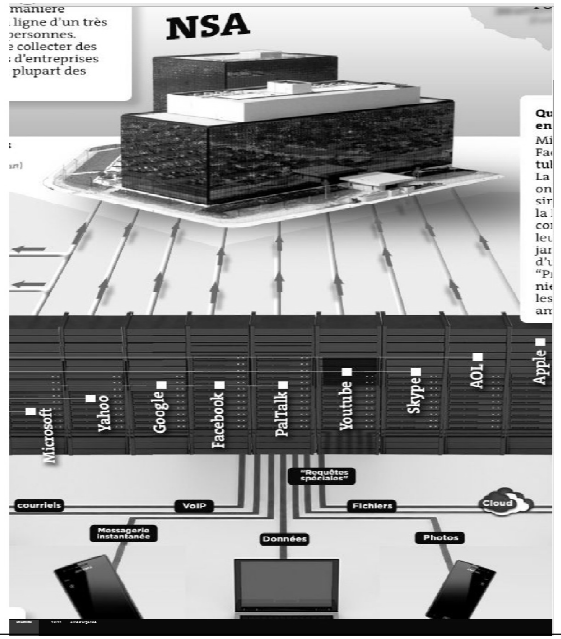
Sont-elles bien protégées ?

- La négligence à l'origine de nombreux trous de sécurité
  - 2 sites de référence répertoriant les plus grandes fuites d'information
    - DataLossDB (mondial)
    - ZATAZ (francophone, environ 50 nouvelles entrées/mois)
  - Quelques exemples emblématiques
    - Données personnelles de 25 Millions de contribuables Anglais égarées (BBC, nov 07)
    - Données de 70 Millions de vétérans US égarées (DataLossDB, oct 09)
    - Achats d'or des clients du CIC (Canard enchaîné, déc 2011)
    - …
- Même les sites les plus sûrs sont piratés
  - Rien qu'en 2014 : Amazon, Sony, Apple …
  - FBI, NASA, Pentagone n'échappent pas à la règle
  - Rapport annuel CSI/FBI
    - Les SGBD constituent la 1ère cible des attaques
    - 45% des attaques sont internes

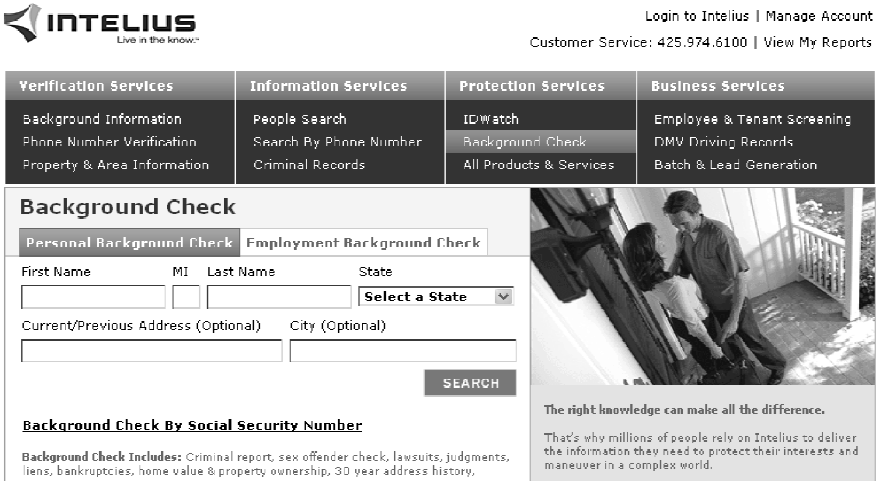
Existe-t-il une parade technique fiable ?

- Oracle's 'Unbreakable' Boast Attracts Hackers
  - Hack attempts on the company's website have increased to 30,000 per week.
- Some days after ….
  - 'When they say their software is unbreakable, they're lying.'
    -- Bruce Schneier
  - U.K. security researcher David Litchfield revealed that a common programming error -- a buffer overflow -- was present in Oracle's application server
  - http://www.theregister.co.uk/2002/02/07/how_to_hack_unbreakable_oracle/

Et le problème est-il uniquement technique ?

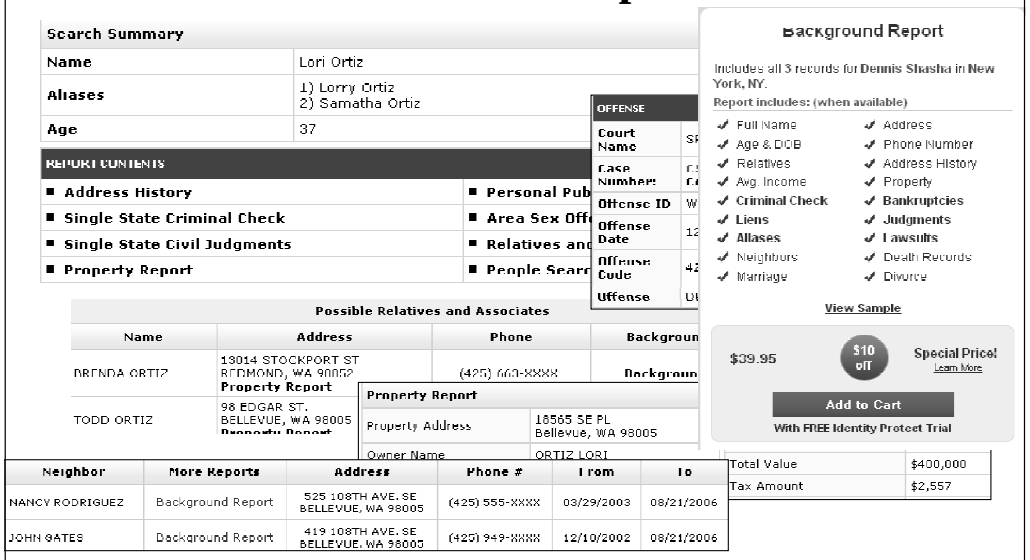


Mais suis-je réellement concerné ?



Que voulez-vous savoir sur vos amis, voisins, nourrice, employés, assurés …?

Le marché des données personnelles



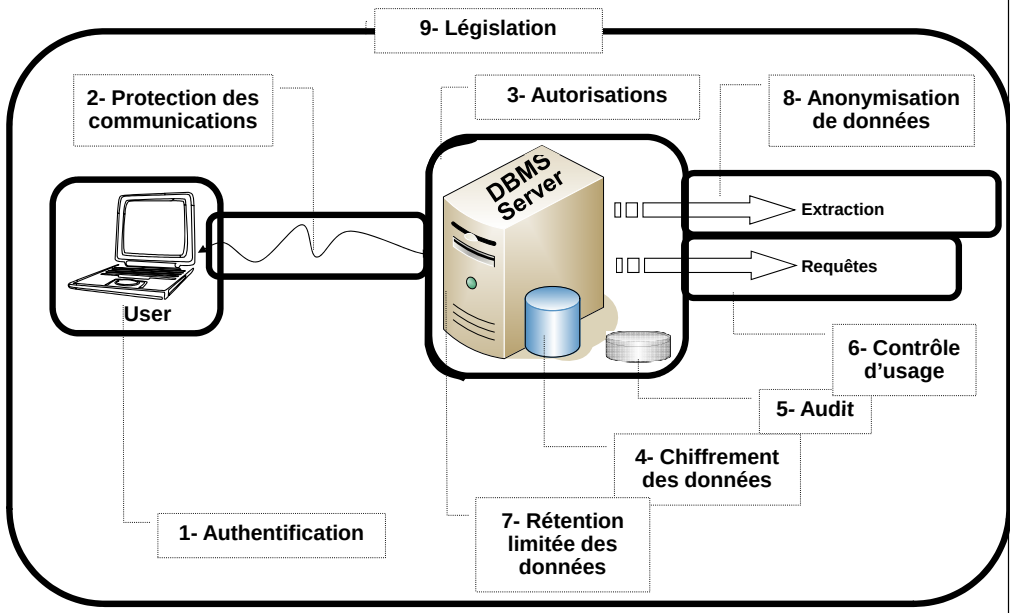
« The data within the report is compiled from thousands of different sources that include government, property, and other public record repositories. »

Et la parade (lucrative) s'organise => ReputationDefender.com

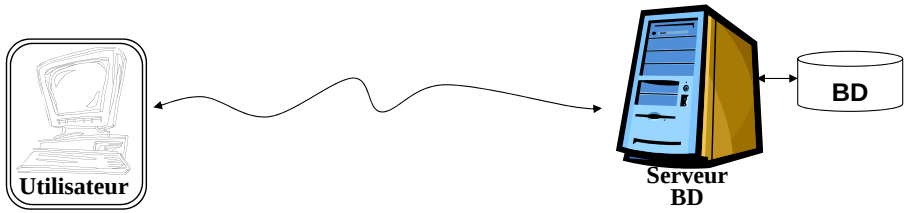
Sécurité des Systèmes d'Information : définition

- Confidentialité
  - Seules les personnes autorisées ont accès aux ressources du SI
  - *Ressources BD: données stockées dans la base ainsi que traitements activables sur ces données*
- Intégrité
  - Les ressources du SI ne sont pas corrompues
  - *BD: toute modification illicite (destruction, altération, substitution, rejeu) des données stockées et échangées doivent pouvoir être détectée*
- Disponibilité
  - L'accès aux ressources du SI est garanti de façon permanente
  - *BD: idem*

Une combinaison d'outils sécuritaires

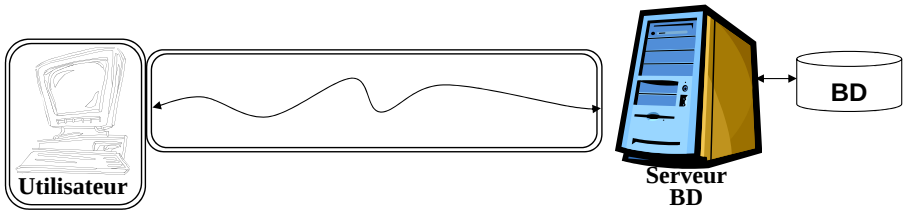


T1 : Identification/authentification



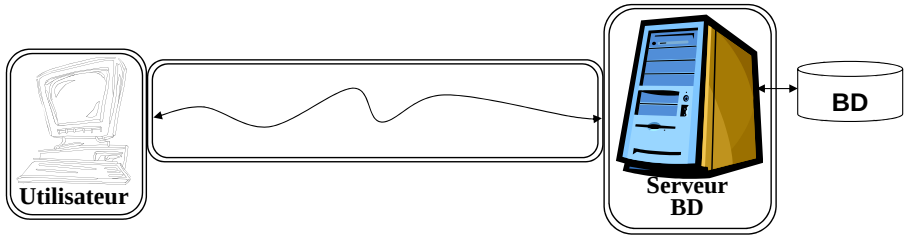
- Au minimum : login (identification) + password (authentification)
- Assuré par le SGBD et/ou l'OS et/ou l'application
- Authentification forte :
  - Forte = 2 éléments d'authentification distincts parmi :
    - Ce que l'entité connaît : password, pin code, etc
    - Ce que l'entité détient : carte à puce, token, badge RFID, etc
    - Ce que l'entité est : empreinte biométrique

T2 : Chiffrement des communications



- Technologie éprouvée (ex: SSL)
- Assure la confidentialité des messages
- Techniques cryptographiques complémentaires
  - Hachage : intégrité des messages
  - Signature : authentification et non répudiation du message

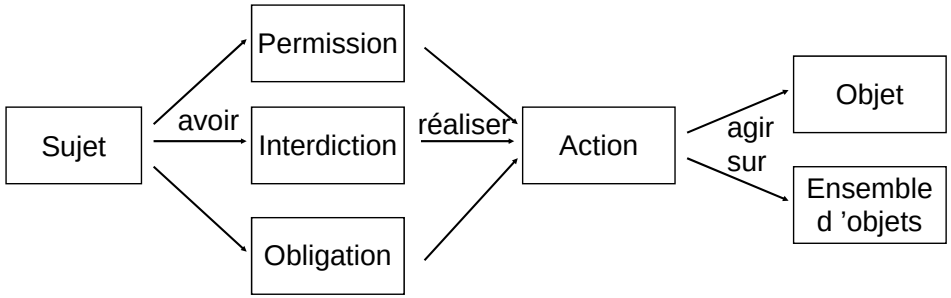
T3 : Contrôle d'accès



- Contrôle d'accès sophistiqué dans les SGBD
  - Autorisations affectées à des utilisateurs, groupes ou rôles
  - Peut porter sur des objets d'une granularité variée : tables, vues, procédures stockées …

Politique de contrôle d'accès = ensemble de règles

- Précise qui est autorisé à faire quoi sur quelles données et sous quelles conditions
- Format des règles :



Modèle discrétionnaire (DAC)

- DAC = Discretionary Access Control
  - Contrôle d'accès discrétionnaire
- Principes de DAC
  - Le créateur d'un objet fixe la politique de contrôle d'accès sur cet objet
  - Les sujets reçoivent des permissions pour réaliser des actions sur des objets
  - Les sujets ont l'autorisation de transférer certaines permissions à d 'autres sujets
    - Droits discrétionnaires de donner des permissions à d'autres sujets

Commandes SQL Grant

- GRANT <liste privileges>
  ON <table ou vue ou procedure stockée >
  TO <liste utilisateurs>
  [WITH GRANT OPTION] ;
- WITH GRANT OPTION
  - est optionnel
  - signifie que l 'utilisateur qui obtient le privilège peut ensuite accorder ce privilège à un autre utilisateur

- Ex: GRANT All
  ON Prescriptions
  TO Dupont // Dupont est médecin
  WITH GRANT OPTION

Privilèges SQL

- Principaux privilèges (permissions) possibles
  - SELECT : permet la consultation de la table
  - INSERT : permet l 'insertion de nouvelles données dans la table
  - UPDATE : permet la mise à jour de n 'importe quelle colonne de la table
  - UPDATE(nom_colonne) : permet la mise à jour d 'une colonne spécifique de la table
  - DELETE : permet de supprimer n'importe quelle donnée de la table
  - ALTER : Modifier la définition d'un objet
  - EXECUTE : Compiler et exécuter une procédure utilisée dans un programme
  - REFERENCE : référencer une table dans une contrainte
  - INDEX : Créer un index sur une table
  - Ainsi que les fonctions d'administration: CREATE/ALTER/DROP TABLE et CREATE/DROP USER

Commande SQL Revoke

REVOKE [GRANT OPTION FOR] <liste privileges>
ON <table ou vue ou procédure stockée >
FROM <liste utilisateurs>
[option_propagation] ;

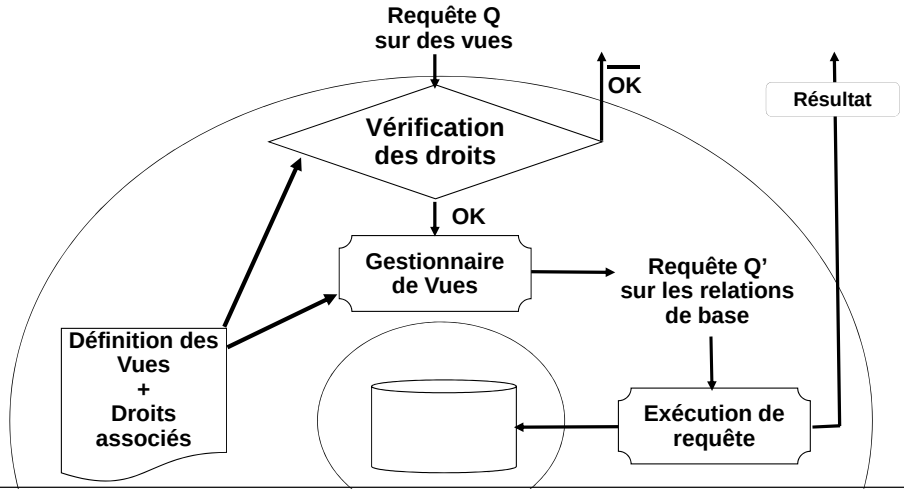
– [GRANT OPTION FOR]
  • signifie que seul le droit de transfert est révoqué
– [option_propagation] = RESTRICT ou CASCADE
  • Supposons que A accorde le privilège p à B et B accorde ensuite p à C
  • CASCADE : si A révoque p à B alors C perd aussi le privilège
  • RESTRICT : si A révoque p à B alors la révocation échoue

• Ex: REVOKE Delete, Update ON Prescriptions
FROM Dupont
CASCADE

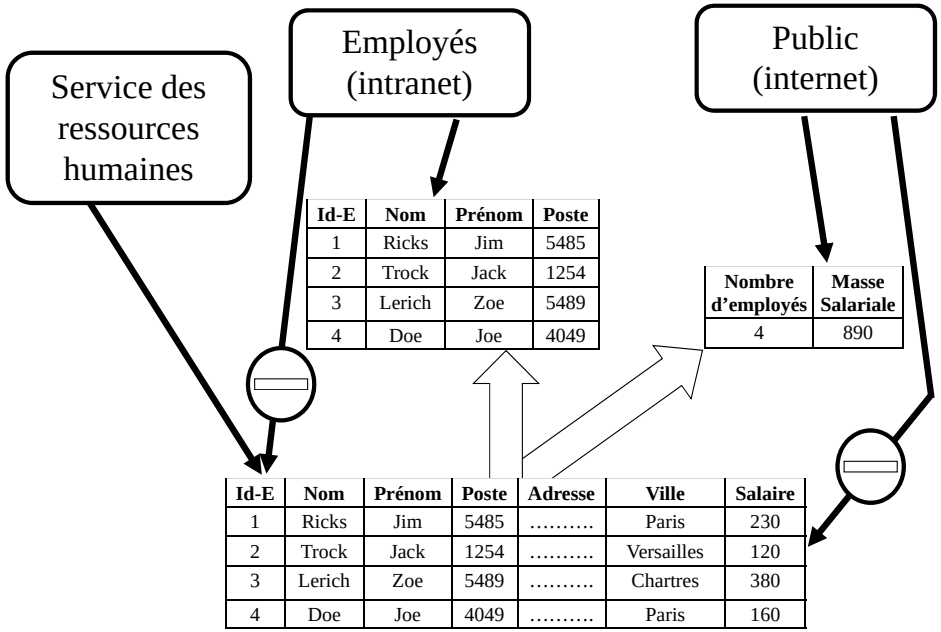
Et si un utilisateur U a reçu le privilège p de A et de B (sans relation entre A et B) ?

Confidentialité via les vues

Principe : Restreindre l'accès à la BD en distribuant les droits via des vues :



Droits d'accès sur des vues



Id-E	Nom	Prénom	Poste
1	Ricks	Jim	5485
2	Trock	Jack	1254
3	Lerich	Zoe	5489
4	Doe	Joe	4049

Nombre d'employés	Masse Salariale
4	890

Id-E	Nom	Prénom	Poste	Adresse	Ville	Salaire
1	Ricks	Jim	5485	…….	Paris	230
2	Trock	Jack	1254	…….	Versailles	120
3	Lerich	Zoe	5489	…….	Chartres	380
4	Doe	Joe	4049	…….	Paris	160

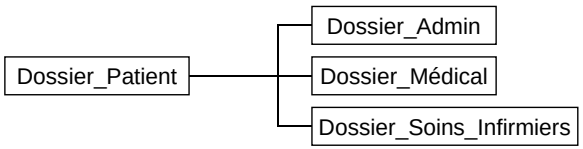
Ex. Système d'information médical

• Sujets = Personnels du groupe médical

Médecin
Jean Jeanne

Secrétaire médicale
Nadine

• Objets = Dossiers des patients décomposé en 3 tables



Expression des règles (exemples)

- R1 : La secrétaire médicale a la permission de gérer le « Dossier_Admin » des patients du groupe médical

```
GRANT ALL PRIVILEGES
    ON dossier_admin
    TO Nadine ;
```

- R2 : Le médecin a la permission de consulter l'intégralité du dossier de ses propres patients

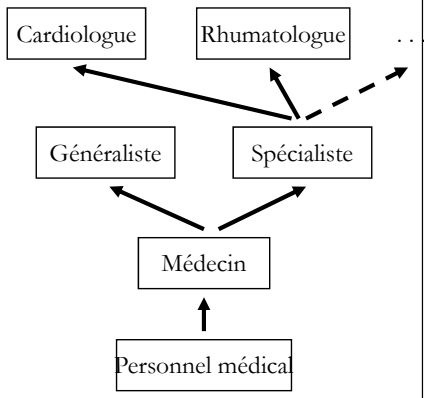
```
CREATE VIEW dossier_patient_du_medecin AS
        SELECT *
        FROM dossier_admin DA, dossier_medical DM, dossier_soins DS
        WHERE DA.medecin_traitant = CURRENT_USER
                and DA.IdPatient = DM.IdPatient and DA.IdPatient = DS.IdPatient;
```

(CURRENT_USER : opérateur prédéfini SQL)

```
GRANT SELECT
    ON dossier_patient_du_medecin
    TO Jean, Jeanne ;
```


RBAC : Role-Based Access Control

- Rôle = ensemble de privilèges associés à une fonction
- Les utilisateurs sont habilités à jouer certains rôles
- Les rôles peuvent être organisés en hiérarchie
- Factorise la gestion des privilèges



- R1 rôle senior de R2

  si chaque fois qu'un utilisateur joue le rôle R1, cet utilisateur joue aussi le rôle R2

- Permet d'exprimer
  - Spécialisation/Généralisation
  - Relation hiérarchique entre employés

RBAC : Gestion des rôles dans SQL

- Création des rôles
  - CREATE ROLE <nom_role> ;
    - Création d'un nouveau rôle nom_role
  - DROP ROLE <nom_role> ;
    - Suppression du rôle nom_role
  - SET ROLE <liste_roles> ;
    - Permet à un utilisateur d 'activer un ensemble de rôles pendant la durée d'une session SQL
- Affectation des privilèges aux rôles

```
GRANT <liste privileges>
ON <table ou vue ou procédure>
TO <liste roles>
[ WITH GRANT OPTION ] ;
```

- Affectation des rôles aux utilisateurs

```
GRANT <liste roles>
TO <liste utilisateurs>
```

- Rôle junior et rôle senior

```
GRANT <role1> TO <role2>
```

Le rôle role2 reçoit tous les privilèges du rôle role1

Exemple d'utilisation

Tous les médecins ont accès aux dossiers de leurs patients

- Définition d'une vue

```
CREATE VIEW dossier_patient_du_medecin AS
        SELECT *
        FROM dossier_admin DA, dossier_medical DM, dossier_soins DS
        WHERE DA.medecin_traitant = CURRENT_USER
                and DA.IdPatient = DM.IdPatient and DA.IdPatient = DS.IdPatient;
```

- Création du rôle médecin

  Create role medecin

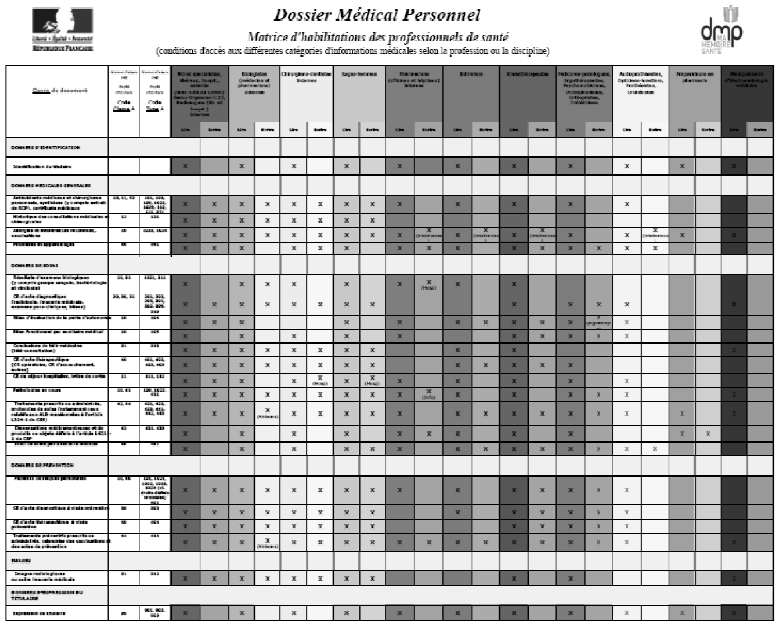
- Affectation des droits au rôle médecin

  Grant all **on** dossier_patient_du_medecin **to** medecin

- Affectation du rôle aux médecins

  Grant medecin **to** jean, jeanne

Exemple de politique RBAC



Dossier Médical Personnel

Matrice d'habilitations des professionnels de santé

(conditions d'accès aux différentes catégories d'informations médicales selon la profession ou la discipline)

MAC : Mandatory Access Control

- Objectif : lutter contre les chevaux de Troie
  - transmission illégale d'informations via un programme malveillant
- Politique de sécurité multi-niveaux (simplifiée)
  - Niveaux de sécurité hiérarchiques
    - Unclassified < Confidentiel < Secret < Très Secret …
  - Le niveau de sécurité d'un utilisateur = niveau d'accréditation
  - Le niveau de sécurité d'un objet = niveau de classification
- Principe du contrôle
  - No read up : read permis si accréditation ≥ classification
  - No write down : write permis si accréditation ≤ classification
- Ex : Oracle Label Security, Label-Based Access Control DB2, Label Security SQL-Server

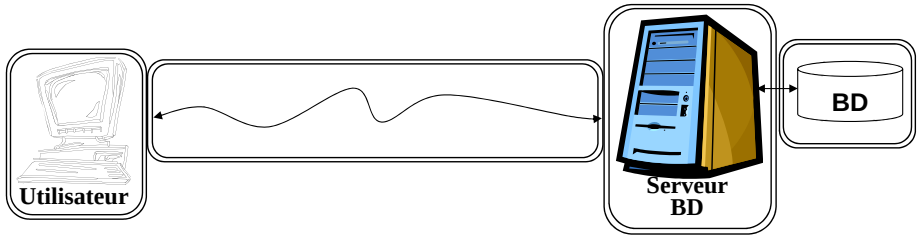
Synthèse sur les modèles de contrôle d'accès

- Principe fondateur

- DAC
  - Permet de structurer les Objets
- RBAC
  - Permet de structurer les Sujets
- MAC
  - Lutte contre les programmes malveillants
  - Mais offre peu de souplesse dans la définition des politiques

➔ **Mais tout cela suppose que l'utilisateur passe "par la porte d'entrée" !!**

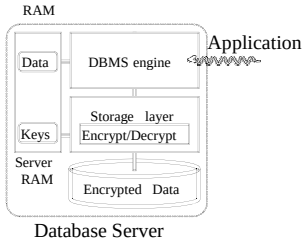
T4 : Chiffrement de la BD



- Principe : chiffrer l'empreinte disque de la BD pour résister aux attaques sur les fichiers
- Exemples :
  - ➔ Oracle Transparent Data Encryption, DB2 Native Encryption …
  - ➔ Protegrity Secure.Data
    - Oracle, Informix, Sybase, SQL Server, IBM DB2

Chiffrer à quel niveau ? (1)

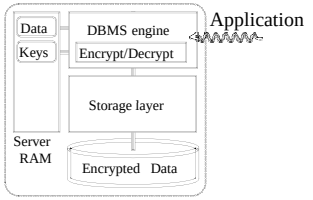
- Chiffrement niveau OS ou Tablespace (chiffrement fichiers/stockage)



- ✚
- ▭ Transparent pour le SGBD et l'application
- – … mais chiffrement non sélectif ➔
  - Granularité = fichier
- ▭ • Chiffrement partiel proscrit (=> performances ?)
- – … et ne résiste pas aux attaques du DBA

Chiffrer à quel niveau ? (2)

- Chiffrement niveau moteur SGBD
  - ✚ Chiffrement sélectif (spécifique)
    - Chiffrer selon les privilèges utilisateur
    - Chiffrer les données les plus sensibles
      - – au niveau table, ligne, colonne…
      - – … de façon conditionnelle (salaire >10K)
  - ✚ Transparence pour l'application
  - ▭ … mais mécanismes internes SGBD à revisiter
    - Evaluation de requête + indexation sur des données chiffrées impossible
      - – sauf chiffrement à propriétés particulières (préservant égalité ou l'ordre => dangereux pour la sécurité)
    - Surtout dans un contexte où le serveur n'est pas de confiance (approche client)
  - ▭ … et problème de performance en perspective
  - ▭ … et ne résiste toujours pas aux attaques du DBA



Application du chiffrement au contexte BD

- Si je chiffre une BD avec un algorithme sûr, le résultat est-il sûr ?
  - – Non !
  - – La robustesse aux attaques des (meilleurs) algorithmes de chiffrement dépend de leur mise en œuvre (mode opératoire/protocole)
    - Qui détient les clés de chiffrement ?
    - Comment sont-elles protégées ?
    - La distribution des données chiffrées ne révèle-elle pas les données en clair ?
- Le contexte BD a des spécificités difficiles à prendre en compte
  - – Gros volume de données, performance des requêtes
  - – Motifs répétés, distribution qui peuvent être connues
  - – Données modifiables
  - – Durée de stockage quasi-illimitée

Un vœu pour le futur : Les Dix commandements des BD Hippocratiques

- Spécification des objectifs:
  - – Description des objectifs d'une collecte d'information et stockage de ces objectifs avec les données
- Consentement:
  - – Doter l'utilisateur du droit de consentir ou pas à divulguer ses données par rapport à un objectif donné.
- Limitation de collecte:
  - – Ne collecter que les données strictement nécessaires à la réalisation des objectifs spécifiés
- Limitation d'usage:
  - – Spécification des couples (objectifs / destinataires) conformes aux préférences et aux directives de privacité.
- Limitation de divulgation:
  - – Ne divulguer les données personnelles qu'aux organismes autorisés avec le consentement du client.

Un vœu pour le futur :
Les Dix commandements des BD Hippocratiques (2)

- Limitation de conservation:
  - Suppression des données dépassant le délai de conservation
- Exactitude:
  - Vérification des données fournies par l'utilisateur
- Sûreté:
  - Protection des données contre les attaques
- Franchise:
  - Doter l'utilisateur du droit d'accès et de modification de ses données personnelles.
- Conformité:
  - Doter l'utilisateur du droit d'auditer le système et de juger de sa conformité