I. Généralités sur la Sécurité des bases de données

La sécurité des Bds désigne un ensemble de moyens, de contrôles et de mesures conçus pour protéger les bases de données contre les menaces accidentelles et intentionnelles afin de préserver la confidentialité, l'intégrité et la disponibilité d'une bd. Elle doit pouvoir gérer et protéger les éléments suivants :

- Les données de la bd
- Le SGBD
- Toutes les applications associées
- Le serveur de base données qu'il soit physique ou virtuel

Les bases de données sont des niches qui regorgent d'informations précieuses car aujourd'hui celui qui détient l'information détient le pouvoir, ils constituent donc une cible de choix pour les pirates. Une entreprise dont les données ne sont pas sécurisées ou dont les données ont été volées peut avoir des conséquences sur les points suivants :

- Une compromission de la propriété intellectuelle : en cas de vol ou de divulgation de la propriété intellectuelle, il peut etre difficile de maintenir ou de récupérer votre compétitivité.
- Continuité des opérations : certaines entreprises ne peuvent pas continuer à fonctionner tant que la violation n'est pas résolue.
- Couts de la remédiation des violations, perte de chiffre d'affaires : frais d'enquête, investigation numérique, frais juridiques, assistance aux victimes restauration des systèmes affectés etc
- Atteinte à la réputation de la marque, perte de clients : les clients hésitent à faire confiance à des entreprises qui ne protègent pas les données personnelles ou qui sont sujettes aux attaques.

La sécurité des bds trouve donc tout son sens car elle permet de protéger toutes les données contre tout accès non autorisé, toute corruption ou tout vol de données depuis

la sécurité physique du matériel jusqu'aux contrôles des administrateurs et des accès en passant par la sécurité logicielle des applications.

1. Les 3 principaux piliers de la sécurité des bds

La sécurité des bds inclus trois principales propriétés : La confidentialité, l'intégrité et la disponibilité.

- L'intégrité : Consiste à s'assurer que les données ne soient ni modifiées, ni falsifiées par des parties non autorisées. Toutefois, l'intégrité d'une base de données consiste à empêcher les parties de faire ce qu'ils veulent tant que les conditions ne sont pas respectées, ceci grâce aux contraintes d'intégrité..

  Menaces : introduction de données valides mais inexactes : les données respectent les contraintes d'intégrité mais ne sont pas exactes. Ex : enregistrement d'une valeur du montant d'un salaire qui ne correspond pas à un salarié,
  Introduction de données invalides : qui ne respectent pas les contraintes d'intégrité :
  Introduction d'une quantité négative, fausse manœuvre de l'utilisateur : exécution d'une procédure inadéquate
  Abus de privilèges légitimes : donner un droit de modification de note à une secrétaire

- La disponibilité : Consiste à s'assurer de l'accessibilité des données 24h/24 – 7J/7.

  Menaces : incident détruisant le support : crash du disque, déconnexion du serveur, incident matériel affectant le serveur,
  Incident logicielle affectant le serveur
  Attaques Deni de service

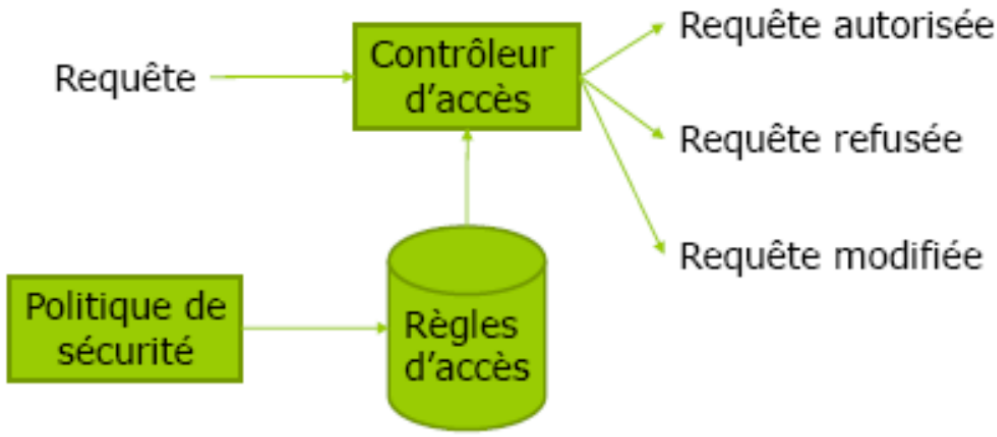
- La confidentialité des données : Consiste à s'assurer que seules les personnes ou programmes autorisés aient accès aux données.

Menaces : un utilisateur accède à des données qui ne lui sont pas autorisées

Abus de privilèges légitimes :

Copies non autorisées des données sensibles

2. Processeur de sécurité dans une base de données



Il s'agit de contrôler l'accès d'un programme ou d'un utilisateur aux données ou aux objets de la base de données en vérifiant que les requêtes adressées au système ne violent pas les règles d'accès et selon le cas autorise, modifie ou refuse la requête en fonction de la politique de sécurité du système d'information qui est traduite sous la forme de règles d'accès (Confère la partie sur les modes d'accès).

3. Les attaques

Une attaque est une action destinée à porter atteinte à la sécurité d'un système d'information, elle représente la concrétisation d'une menace informatique. (? action ou évènement qui peut exploiter une vulnérabilité et susceptible de nuire. ? vulnérabilité : faiblesse observée au sein d'un système). Toute attaque contre les données met en danger le SI et par ricochet l'entreprise.

3.1 Types d'attaques

On distingue deux types d'attaques : les attaques non frauduleuses et les attaques frauduleuses.

- Les attaques non frauduleuses : catastrophes naturelles, pannes logicielles ou de matérielles, erreurs humaines : (mots de passe faibles, partage de mots de passe... , coupure de courant…
- Les attaques frauduleuses : Les attaques sur une bd peuvent exploiter les menaces suivantes qui peuvent etre à la fois interne et externe :
    1. Abus de privilège excessif : Lorsque des utilisateurs ou programmes ont des privilèges d'accès à une base de données excédant les exigences de leur fonction professionnelle, ils peuvent abuser de ces privilèges à des fins malveillantes. (Pourquoi un utilisateur peut se trouver avec des privilèges excessifs : l'admin n'a pas le temps de définir, ni le temps de mettre à jour les mécanismes de contrôle d'accès, certains utilisateurs sont créés avec des privilèges par défaut qui ne cadrent pas avec leur fonction professionnelle, abus de confiance.
    2. Abus de privilège légitime : Les utilisateurs peuvent également abuser de leurs privilèges légitimes afin d'accéder à la base de données à des fins non autorisés.
    3. Élévation de privilèges : Technique qui consiste à un utilisateur disposant d'un accès restreint à des SI d'élargir le périmètre et l'étendue de ses autorisations voir à celui de l'administrateur. Pour cela, les hackers tirent profit la plupart du temps des failles de

conception ou vulnérabilités des applications web.On distingue deux catégories d'élévation de privilèges :

- L'élévation horizontale : qui consiste à obtenir un accès privilégié à un compte utilisateur standard doté de privilèges de niveau inférieur et une fois dans le SI, l'attaquant étend son accès privilégié.
- L'élévation verticale : qui consiste à obtenir un accès à des comptes dotés de privilèges et d'autorisations de niveau supérieur.

4. Injection SQL ou SQLI : Elle consiste à contrefaire une instruction SQL de manière à la détourner de son objectif initial, permettant ainsi à l'attaquant d'altérer, voler ou de détruire des données et dans le pire des cas d'accéder de manière totale à la bd.

Principe : Une injection SQL se produit lorsqu'un utilisateur malveillant communique une entrée qui modifie la requête SQL envoyée par l'application web à la bd lui permettant alors d'exécuter d'autres requêtes SQL non souhaitées directement sur la bd. Pour ce faire, l'attaquant doit injecter du code en dehors des limites de l'entrée utilisateur attendue afin qu'il ne soit pas exécuté comme une entrée standard.

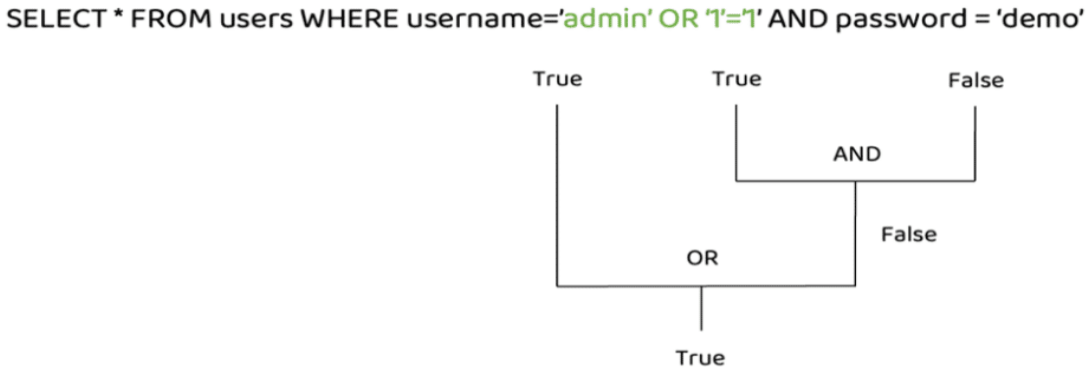
Cas pratique : contournement d'une authentification

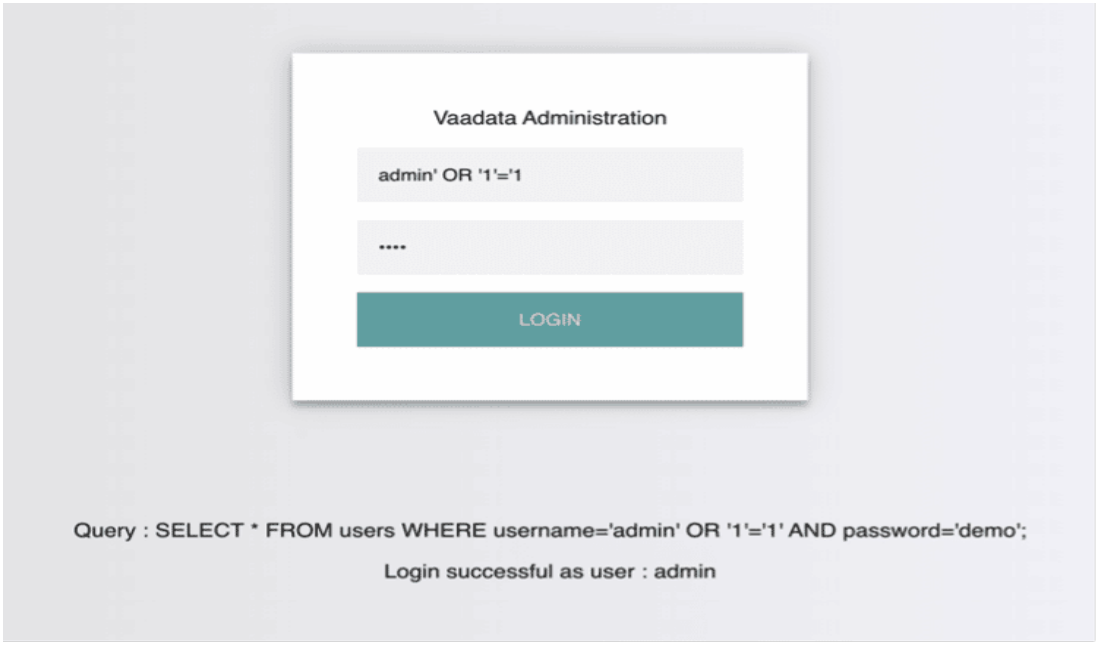
Payload pouvant etre utilisés : ', <<, #, ;,)

- Utilisation de l'opérateur OR : pour contourner l'authentification, il faudrait que la requête renvoie true, quels que soient le nom d'utilisateur et

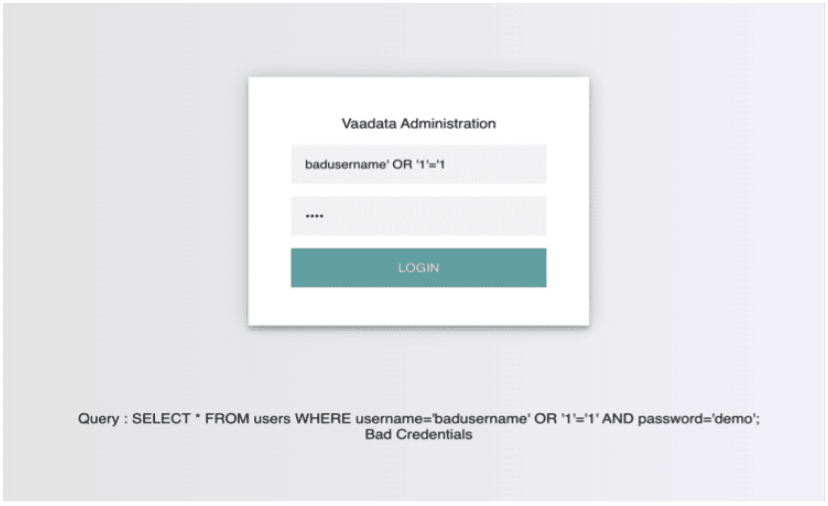
le mot de passe saisis. En MYSQL, l'opérateur AND est évalué avant l'opérateur OR

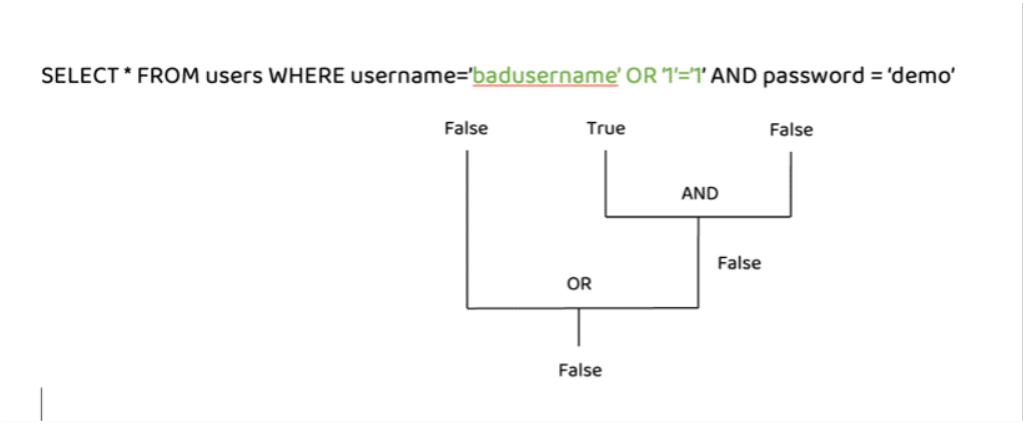
```
select * from login where username = 'estelle' or '1'='1' and
password= 'estelle'; //le mot de passe étant faux
```



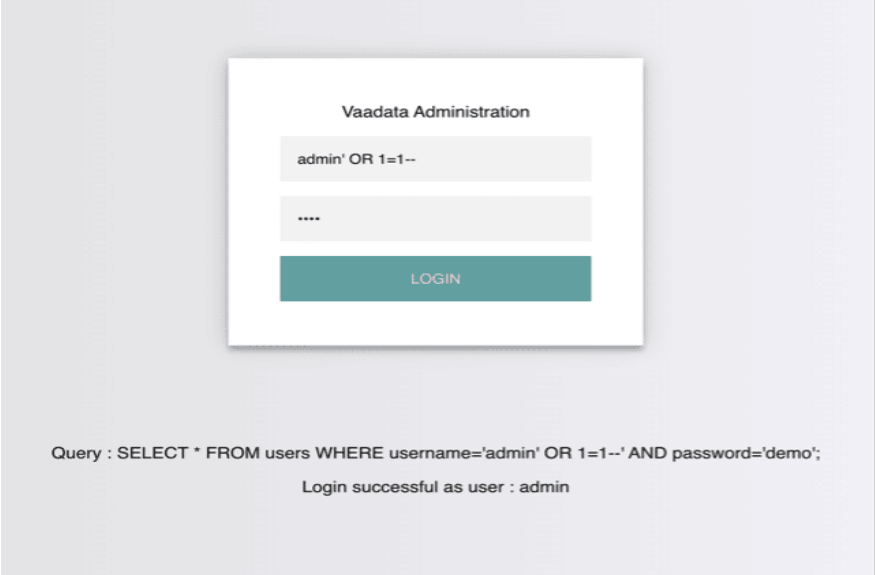


Si le nom d'utilisateur n'est pas valide, la connexion va échouer parce qu'il n'existe pas dans la table et a donné lieu à une fausse requête globale.





- En utilisant les commentaires :select * from login where username = 'estelle' or 1=1;# and password = 'grayce';



N.B :L'injection SQL fait référence aux attaques contre les bases de données relationnelles telles que MySQL, Oracle Database ou Microsoft SQL Server. En revanche, les injections contre les bases de données non relationnelles, telles que MongoDB ou CouchDB, sont des injections NoSQL.

5. Attaque par déni de service : Dans une attaque par déni de service distribué (DOS), le pirate inonde le serveur de base de données d'autant de demandes de manière à ce que celui-ci ne puisse plus répondre aux demandes légitimes des utilisateurs réels poussant ainsi le serveur à etre instable ou bloquer.

3.2 Les types de pirate

Sur une bd, on peut avoir différents types d'attaquants :

- Pirate externe : personne externe au système capable de s'infiltrer sur le serveur de base de données dans le but d'altérer, voler ou supprimer des données.
- Pirate utilisateur : Personne reconnu par le SGBD et voulant accéder à des données dont il n'a pas les autorisations nécessaires.
- Pirate administrateur : personne qui se sert de ses droits administrateurs ou qui s'est octroyé ces droits pour mieux espionner le SGBD à des fins malveillantes.

3.3 Les risques encourus

- Le vol de données conduit à la perte de la 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, une application fonctionnant sur des données falsifiées peut voir son comportement fortement influencé et corrompu: 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 !

3.4 Les types d'utilisateur

On distingue 3 principaux types d'utilisateurs d'une bd :

Utilisateurs finaux	Interagissent avec la base de données principalement pour l'**interroger** et **imprimer des rapports**. Tout dépendant leur niveau de connaissance du langage d'interrogation, ils peuvent soit utiliser des requêtes prédéfinies ou en formuler eux-mêmes.
Administrateurs de bases de données	Assurent la **gestion technique** nécessaire pour **implémenter les SGBD** : définition de la structure conceptuelle et physique, définitions des règles de sécurité, interaction avec les utilisateurs finaux, supervision des performances, etc.
Programmeurs d'application	Programment des **applications** pour **interagir** avec la base de données (par exemple des pages ASP ou PHP pour mettre une base de données en ligne).

Les **professionnels de l'information** interviennent principalement comme **utilisateurs finaux** ou **administrateurs de bases de données**. Il leur est aussi possible d'agir comme *programmeurs d'application* pour un système à petite échelle

4.Les mesures de protection

Afin de garantir la sécurité d'un SGBD plusieurs moyens peuvent etre utilisés :

4.1 Les vues

Une vue est une table virtuelle qui contient le résultat d'une requête, elle ne stocke pas les données mais conserve juste la requête permettant de les créer. Elle est utilisée pour faciliter certaines requêtes ou les exprimer plus simplement mais également pour protéger les données donnant ainsi la possibilité de cacher certains champs aux utilisateurs et de personnaliser l'affichage des informations suivant le type d'utilisateur.

Cas pratique : création de vues en fonction du type d'utilisateur

Création de deux tables : étudiant(matricule, nom, prenom, quartier, date_naissance) et note(avec clé étrangère)

Création de 3 utilisateurs : daac, daac adjoint, secdaac : create user 'daac'@'localhost' identified by 'daac' ;

- Creation de vue pour la daac :

  create view v_daac as select * from etudiant inner join note on etudiant.matricule= note.matricule ;

  grant all privileges on v_daac to 'daac'@'localhost'

- Creation de vue pour le daac adjoint :

```
create view v_daacadjoint as select
etudiant.nom_etudiant,etudiant.prenom,etudiant.sexe,
note.code, note.note from etudiant inner join note on
etudiant.matricule = note.matricule ;


grant select, update on v_daacadjoint to
'daacadjoint'@'localhost';
```

- Création de vue pour la sécretaire du daac

```
create view v_secdaac as select etudiant.sexe, note.code,
note.note from etudiant inner join note on etudiant.matricule
= note.matricule ;

grant select on v_secdaac to 'secdaac'@'localhost' ;
```

4.2 L'authentification

L'authentification de la base de données est le processus ou l'acte de confirmation qu'un utilisateur qui tente de se connecter à une base de données est bien celui qu'il prétend etre . Avec une stratégie d'authentification et d'autorisation bien réfléchie, les entreprises peuvent vérifier de manière efficace l'identité de chaque utilisateur et ce à quoi ils ont accès ce qui est un facteur de renforcement de la sécurité de ces entreprises favorisant une meilleure productivité.

4.3 Le contrôle d'accès

C'est un élément essentiel de la stratégie de la sécurité et qui consiste à s'assurer qui est autorisé ou pas à accéder à certaines ressources de la bd.

Un système de contrôle d'accès comprend :

- Des sujets ou entité active: entité qui initie la demande d'accès (utilisateurs, processus qui s'exécute pour le compte d'un utilisateur …)
- Des objets ou entité passive : entité à laquelle le sujet souhaite accéder et qui contient des informations. (fichiers, programmes…) (un objet peut également etre un sujet : expliquer)
- Des opérations : lecture, ajout, modification, suppression etc déclenchées par des sujets sur des objets
- d'un ensemble de règles d'accès ou permissions traduisant la politique de sécurité de l'entreprise. En fonction de ces règles, le système va autoriser, modifier ou interdire les requêtes venant des sujets.

Modèles de Contrôle d'accès

Les modèles de contrôle d'accès constituent la base de toute politique en matière de sécurité. Il en existe plusieurs :

- **Modèles discrétionnaires** : Encore appelés DAC ou Discretionary Access Control

Principe :

- Le créateur d'un objet est le propriétaire de cet objet
- Le propriétaire peut transmettre ou retirer à sa discrétion des autorisations sur ses objets à d'autres utilisateurs
- Le propriétaire peut transmettre à d'autres utilisateurs le droit de transmettre ces autorisations
- Ce qui n'est pas autorisé est interdit

Exemples de modèle discrétionnaire : Modèles basés sur les matrices de contrôle d'accès - Modèle de Lampson et le modèle HRU

La notion de matrice de contrôle d'accès est dédiée à la représentation des droits d'accès sous forme de matrice et a été introduite par Lampson des 1971. La structure de ce modèle est représentée sous forme d'un triplet (S,O,M) avec S : ensemble de sujets, O : ensemble d'objets et M matrice de contrôle d'accès. Chaque cellule M(s,o) contient les droits d'accès que le sujet s possède sur l'objet o. Les objets représentent les colonnes et les sujets les lignes.Ce modèle a été amélioré pour donner naissance au modèle HRU (harrison,ruzzo et ullman) en 1976 qui est défini par un quadruplet (S,O,R,M) où :

S : ensemble de sujets, O : ensemble d'objets, R :ensemble des modes d'accès et M la matrice d'accès

R comprend les accès suivants : read, write, append : ajout de contenu à un objet sans voir son contenu, execute, own : possession. La différence réside dans le fait que HRU spécifie les commandes à attribuer des droits d'accès ainsi que de créer et supprimer des sujets et des objets. On peut ajouter un droit a dans une matrice d'accès s'il existe une commande C qui ajoute le droit a dans une cellule M (s,o).

Syntaxe d'une commande :

command nom_command(X1 , X2 , …, Xk)

if r1 in (Xs1, Xo1) and … and rm in (Xsm, Xom)

then op1 … opn

end

Il existe 6 opérations primitives :

- Enter a
- Delete a
- Create subject s
- Delete subject s
- Create object o
- Delete object o

exemple : command GRANT_read(x1,x2,y) if `own' in [x1,y] then enter `read' into [x2,y] end

Exemple de matrice d'accès

OBJETS / sujets	fichier F_1	fichier F_2
utilisateur U_1	own U_1 a tous les droits sur F_1 dont celui de transmettre tout ou partie de ces droits à un autre utilisateur	x* U_1 a le droit d'exécuter F_2 et de transmettre ce droit à un autre utilisateur
utilisateur U_2	_	own
programme P	r, w P a le droit de lire ou de modifier F_1	_

N.B : dans le modèle HRU, les sujets sont représentés en lignes et les sujets et les objets en colonne.

	Objects					
Subjects	S_1	S_2	S_3	O_1	O_2	O_3
S_1	control			owner	read write	
S_2		control	read*			execute
S_3			control		owner	

Exercice d'application :

1. Le sujet s1 a un droit de lecture et d'exécution sur le fichier toto.exe. Le sujet s2 peut exécuter toto.exe, lire et ecrire le fichier tata.txt et enfin s1 a un droit de lecture sur le fichier tata.txt. Modéliser la matrice de contrôle d'accès qui représente ces autorisations.

   Sujets : s1, s2

   Objets : toto.exe :f1, tata.txt : f2

   M(s1,f1) : read, execute

   M(s2,f1) :execute

   M(s2,f2) :read, write

   M(s1,f2) : read

2. Soient trois utilisateurs : s1,s2,s3 et deux objets : une imprimante imp et un fichier toto.txt. Soient les autorisations suivantes :
   - S1 est le propriétaire de toto.txt
   - Imp est d'accès libre pour tout le monde
   - Imp ne peut imprimer un fichier que si la requête d'impression provient d'un sujet qui a le droit d'imprimer ce fichier
   - S2 peut lire toto.txt
   - S3 peut imprimer tous les fichiers sur lesquels ils disposent d'un droit de lecture

- S3 n'a aucun droit sur toto.txt

Modéliser la matrice de contrôle d'accès du modèle Lampson et HRU

Avantages

- Facile à implémenter ;
- Offre une grande flexibilité ;
- Intégré à la plupart des systèmes unix, sgbd

Inconvénients

- Le propriétaire de l'objet définit les privilèges d'accès aux objets plutôt que par le biais d'une stratégie système reflétant les exigences de sécurité de l'organisation ;
- Modèle inadapté à un système comportant un nombre important d'utilisateurs (scalabilité) ;
- Risque d'explosion des ACL ;
- Ne reflète pas le flux réel de l'information dans un système, les informations autorisées pouvant être copiées d'un objet à un autre ;
- Aucune restriction ne s'applique à l'utilisation des informations lorsque l'utilisateur les a reçues ;
- Sujet à de nombreuses erreurs lors de l'attribution des autorisations par le propriétaire de l'objet.
- Ils sont vulnérables aux chevaux de troie

- **Modèles obligatoires** : Encore appelés MAC : Mandatory Access Control

Ils ont été développés pour des systèmes d'information pour lesquels la préservation du secret est primordial : (armée, gouvernement par exemple). L'accès aux objets est

restreint en fonction de la sensibilité des informations contenues dans les objets et du niveau de l'autorisation de l'utilisateur de disposer d'une telle sensibilité.

Principe :

Les objets se voient attribuer une classification (top secret, secret, confidentiel, public etc) tandis que les sujets possèdent une habilitation (niveau d'autorisation). Les règles qui régissent les autorisations d'accès sont basées sur une comparaison de l'habilitation de l'utilisateur et de la classification de l'objet.

Le contrôle d'accès obligatoire est utilisé lorsque la politique de sécurité des systèmes d'information impose que les décisions de protection ne doivent pas etre prises par le propriétaire des objets concernés. Généralement, la personne chargée d'administrer les droits d'accès est quelqu'un qui connait bien la politique de sécurité de l'entreprise (généralement l'administrateur système)

Exemples de modèles obligatoires

1. Modèle de BELL et LAPADULA (BLP)

Modèle développé par David Bell et Leonard Lapadula en 1973 pour formaliser la politique de sécurité multi-niveau du département de la défense des Etats-unis. Ce modèle met l'accent sur la confidentialité des données :

- Interdire toute fuite d'information d'un objet avec un certain niveau de classification vers un objet de niveau de classification inférieur
- Interdire à tout sujet d'une certaine habilitation d'obtenir des informations d'un objet de classification supérieur à cette habilitation

L'ensemble des niveaux de secret est muni d'un ordre partiel > =: Top secret >=Secret >=Confidentiel >= Non classifié

Propriétés de sécurité :

Un état du système est sécuritaire s'il vérifie les propriétés suivantes :

- La simple propriété de sécurité (SS-propriété) ou propriété no read-up : Un sujet ne peut accéder à un objet que si son niveau de sécurité est supérieur ou égal à celui de l'objet.

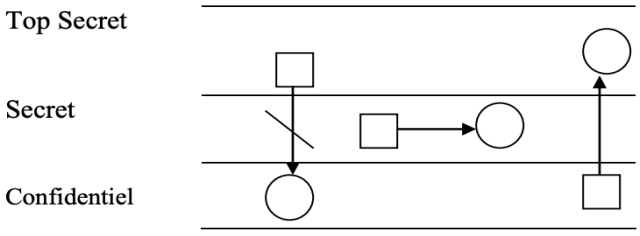


Figure 3 : Ne pas lire en haut

Rond :sujet, carré :objet

Explication :Un sujet qui a l'habilitation confidentiel n'a pas le droit d'accéder en lecture à un objet qui a la classification Top secret parce que le niveau de sécurité confidentiel est inférieur à celui de Top secret, par contre un sujet qui a l'habilitation top secret a le droit d'accéder en lecture à un objet qui a la classification confidentiel parce que le niveau de sécurité confidentiel est inférieur au niveau de sécurité top secret. (quand est-il de secret et secret, est-ce possible ? oui). La satisfaction de cette propriété assure qu'un sujet n'accédera pas à une information classée à un niveau plus haut que lui.

- La propriété étoile ou no write-down : Un sujet ne accéder en écriture à un objet si et seulement si le niveau de sécurité du sujet est inférieur ou égal au niveau de classification de l'objet.

Explication :Un sujet qui a l'habilitation confidentiel a le droit d'accéder en écriture à un objet qui a la classification top secret parce que son niveau de sécurité confidentiel est inférieur au niveau de sécurité top secret par contre un sujet qui a l'habilitation top secret n'a pas le droit d'accéder en écriture à un objet qui a la classification confidentiel parce que le niveau de sécurité confidentiel est inférieur au niveau de sécurité top secret.

Exemple d'application :

Soit les éléments suivants :

S = {Bob, Sonia}

O = {fichiers personnels, fichiers du courriel, fichiers du log, fichiers des coordonnées}

L = {Top Secret, Secret, Confidentiel, Non classé} avec Top Secret représente le haut du système et Non classé représente le bas du système.

Niveau de sécurité	Sujet	Objet
Top Secret	Bob	fichiers personnels
Secret		fichiers du courriel
Confidentiel		fichiers du log
Non classé	Sonia	fichiers des coordonnées

Écrivez les règles qui découlent de ce tableau en utilisant le modèle de Bell Lapadula, puis représentez la matrice de contrôle d'accès suivant le modèle de Lampson

- Bob a le droit d'accéder en lecture à tous les fichiers parce qu'il a la classification Top Secret qui est supérieure aux classifications de tous les fichiers.
- Bob a le droit d'accéder en écriture aux fichiers personnels parce que son habilitation Top Secret est égale à la classification des fichiers personnels, mais il n'a pas le droit d'écrire dans les autres fichiers parce que leurs classifications sont inférieures à son habilitation.

• Sonia a le droit d'accéder en écriture à tous les fichiers parce qu'elle a la classification Non classé qui est inférieure aux classifications de tous les fichiers.

• Sonia a le droit d'accéder en lecture aux fichiers des coordonnées et n'a pas le droit d'accéder en lecture aux autres fichiers parce que son habilitation Non classé est égale à la classification des fichiers des coordonnées, mais elle n'a pas le droit d'accéder en lecture aux autres fichiers parce que leurs classifications sont supérieures à son habilitation.

2. Modèle de Biba

C'est un modèle développé par Kenneth J.Biba en 1977 qui formalise, la sécurité multi-niveaux et met l'accent sur l'intégrité des données c'est-à-dire interdire toute propagation d'information d'un objet situé à un certain niveau d'intégrité vers un objet de niveau d'intégrité inférieur, interdire à tout sujet situé à un certain niveau d'intégrité de modifier un objet possédant niveau d'intégrité supérieur. Tout comme le modèle Bell LaPadula, le modèle de Biba est défini par deux propriétés : la propriété simple et la propriété étoile.

- Propriété simple ou no read down :cette propriété interdit à un sujet d'accéder en lecture à un objet qui a une classification moins élevée que l'habilitation du même sujet.

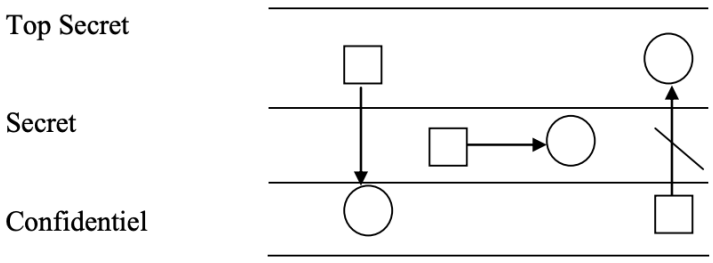


Figure 6 : Ne pas lire en bas

Rond :sujet, carre :objet

Explication :un sujet qui a l'habilitation Confidentiel a le droit d'accéder en lecture à un objet qui a la classification Top Secret parce que le niveau de sécurité Confidentiel est inférieur au niveau de sécurité Top Secret. Et un sujet qui a l'habilitation Top Secret n'a

pas le droit d'accéder en lecture à un objet qui a la classification Confidentiel parce que le niveau de sécurité Confidentiel est inférieur au niveau de sécurité Top Secret.

- La propriété étoile ou no write up :cette propriété interdit à un sujet d'accéder en écriture à un objet qui a une classification plus élevée que son habilitation.

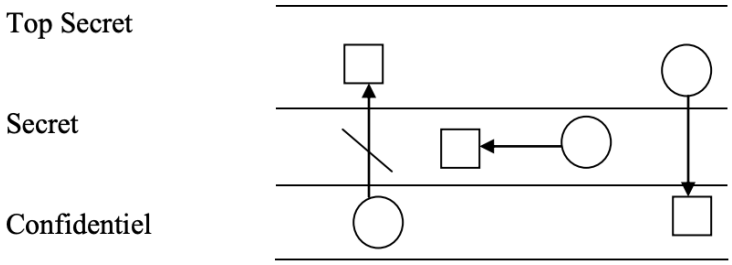


Figure 7 : Ne pas écrire en haut

Explication :un sujet qui a l'habilitation Confidentiel n'a pas le droit d'accéder en écriture à un objet qui a la classification Top Secret parce que le niveau de sécurité Confidentiel est inférieur au niveau de sécurité Top Secret par contre un sujet qui a l'habilitation Top Secret a le droit d'accéder en écriture à un objet qui a la classification Confidentiel parce que le niveau de sécurité Confidentiel est inférieur au niveau de sécurité Top Secret.

Exemple d'application : Reprendre l'exemple précédent

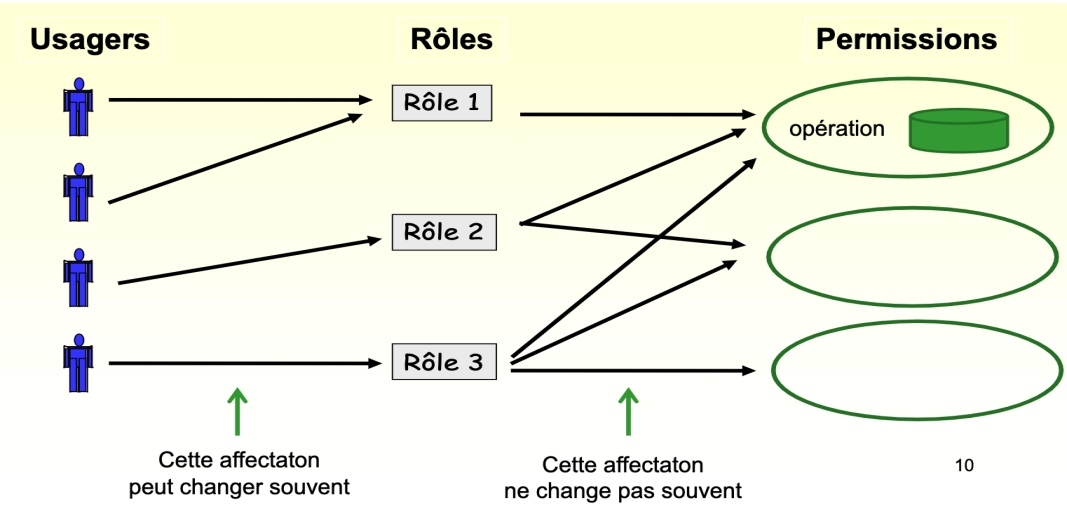
- Bob a le droit d'accéder en écriture à tous les fichiers parce qu'il a la classification Top Secret qui est supérieure aux classifications de tous les fichiers.
- Bob a le droit d'accéder en lecture aux fichiers personnels parce que son habilitation Top Secret est égale à la classification des fichiers personnels, mais il n'a pas le droit d'accéder en lecture aux autres fichiers parce que leurs classifications sont inférieures à son habilitation.

- Sonia a le droit d'accéder en lecture à tous les fichiers parce qu'elle a la classification Non classé qui est inférieure aux classifications de tous les fichiers.
- Sonia a le droit d'accéder en écriture aux fichiers des coordonnées parce que son habilitation Non classé est égale à la classification de ces fichiers, mais elle n'a pas le droit d'accéder en écriture aux autres fichiers parce que leurs classifications sont supérieures à son habilitation.

  Devoir : avantages et inconvénients du MAC.

- **Modèles à base de rôles : e**ncore appelés RBAC pour Role Based Access Control peut être considéré comme une approche alternative au contrôle d'accès obligatoire (MAC) et le contrôle d'accès discrétionnaire (DAC). Dans ce modèle, les permissions sont affectées à des rôles spécifiques au lieu d'être affectés directement à des sujets.

La motivation principale autour de la proposition d'un modèle de contrôle d'accès à base de rôle (RBAC) était de faciliter l'administration des privilèges d'accès pour un grand nombre d'utilisateurs accédant à des ressources distribuées. Le but est de de regrouper les utilisateurs dans des rôles reflétant la structure organisationnelle de l'entreprise et puis, de distribuer les permissions à ces rôles au lieu de le répéter par individu.



Explication : Plusieurs sujets peuvent être attribués à plusieurs rôles comme ils peuvent n'avoir aucun rôle, plusieurs rôles peuvent détenir plusieurs permissions comme ils peuvent ne détenir aucune permission, un sujet a une permission p si et seulement si ce sujet est attribué à un rôle qui détient cette permission.

RBAC est considéré comme un système « idéal » pour les entreprises dont la fréquence de changement du personnel est élevée

Create role nom_role ;
Grant nom_privileges to role ;
Grant role to nom_user ;
Set role nom_role ; activation du role
Grant nom-role1 to nom-role2

TP : création de la bd gestetudiant, avec 3 tables : etudiants, notes, matieres
A l'iai Cameroun, la direction des affaires académiques est chargée de la gestion des notes des étudiants et a à sa tête une directrice qui possède les pleins droits sur les différentes entités, les responsables de filière sont chargés de consulter, d'insérer, modifier les notes étudiants et ont également une vue totale sur les informations des étudiants ainsi que des matières dispensées, le directeur adjoint quant-à lui est chargée de la gestion complète des étudiants et a une vue globale sur les notes et matieres dispensées. Les secrétaires quant-à elles ne peuvent que insérer,consulter et modifier les informations personnelles des étudiants.
Soient les utilisateurs suivant Anga directrice des affaires académiques, Salabessies directeur adjoint, Agbor responsable de filière software, Belinga responsable de filière, Angounda secrétaire de la directrice et Tsama secrétaire des responsables de filière.
Représentez le modèle RBAC de cette direction.

4.4 La protection des données utilisateurs

Elle peut se faire à travers le cryptage des données, la sauvegarde et la sécurité physique.

4.5 Audit des accès

La maintenance d'une base de données est indispensable. Dans ce cas-là, l'audit est une opération de routine, destinée à maintenir la qualité de la base de données. Il consiste à analyser l'existant d'une base de données (ou d'un ensemble cohérent de bases de données) plus ou moins afin d'en diagnostiquer l'état et de préconiser des améliorations, essentiellement sur le plan de la conformité et des performances. On distingue les audits de structure : il s'agit de montrer si la bd est en adéquation avec les exigences fonctionnelles (respect des formes normales, contraintes de domaine….), les audits de qualité : il s'agit de vérifier que la bd n'est pas polluée par de nombreuses données inutiles ou erronées (on vérifiera l'existence de contraintes : clé primaire, clé étrangère, unicité, validation etc.), les audits de configuration et de performances : il s'agit de vérifier que la configuration du sgbd et du serveur physique est conforme aux exigences du service : RAM, Processeurs etc…, l'audit des requêtes (procédures, transactions, analyse, traçage de l'activité du moteur, revue de code etc, l'audit d'infrastructure réseau : il s'agit de vérifier ce qui se passe entre le SGBD et les clients : mesurer les temps de réponse. Un admin de bd doit donc pouvoir faire des audits sur par exemple sur l'utilisation de la bd en dehors des heures ouvrables, l'audit sur la manipulation du schéma de la bd, des erreurs, modification des fonctions et procédures stockées etc…

4.6 Limitation de privilèges

Un privilège est un droit ou autorisation accordé à un processus ou un utilisateur pour réaliser une opération. . On distingue les privilèges objets (qui concernent les opérations précises sur les tables, vues etc..) et les privilèges systèmes (qui concernent des opérations sur la structure des objets et sur une catégorie d'objet : create any table). Cependant la mauvaise gestion de ces privilèges peut etre une ouverture dans le système.

4.6.1 Principe du moindre privilèges

Le principe du moindre privilège fait référence à un concept de sécurité dans lequel on accorde à un utilisateur le niveau d'accès (ou les permissions) minimum requis pour accomplir son travail. Pour implémenter le principe du moindre privilège, vous pouvez procéder de la manière suivante :

- **Auditer** la totalité de l'environnement afin d'identifier les comptes à privilèges (mots de passe, clés SSH, codes de hachage de mots de passe ou clés d'accès) sur site, dans le cloud, dans les environnements DevOps et sur les terminaux.

- Éliminer les **privilèges d'administrateur local inutiles** et s'assurer que tous les utilisateurs humains et non humains disposent uniquement des privilèges nécessaires pour accomplir leur travail.
- Provisionner les identifiants de comptes à privilèges dans un **coffre-fort numérique** pour commencer à sécuriser et à gérer ces comptes.
- Renouveler immédiatement tous les mots de passe administrateur après chaque utilisation pour invalider tous les identifiants qui auraient pu être enregistrés par un keylogger et pour diminuer le risque d'attaque **Pass-the-Hash**.
- Superviser en continu toutes les activités liées aux comptes administrateur afin de faciliter la détection et la création d'alertes sur les activités anormales susceptibles d'indiquer qu'une attaque est en cours.

4.6.2 Politique de gestion 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. Les privilèges peuvent évoluer au cours du temps, car les besoins et les tâches affectées ne sont pas immuables, mais à un moment donné, seuls les droits indispensables doivent être fournis à un utilisateur.

Règle n° 2 : contrôle de la population.
Le personnel d'une entreprise bouge, il y a des départs, des arrivées, des promotions… Les privilèges doivent être 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.
La connexion d'un utilisateur à une base de données peut être réalisée depuis n'importe où dans le monde grâce à Internet. Il est nécessaire de restreindre les connexions à des hôtes spécifiques connus. Par exemple, le compte d'accès d'une application hébergée sur un serveur devrait voir ses privilèges restreints à l'hôte (ou son domaine) sur lequel elle est hébergée.

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. Une personne physique n'a pas besoin de réaliser 100 requêtes à la seconde, mais au contraire, une application gérant elle-même les habilitations peut avoir de gros besoins qui peuvent être cependant limités raisonnablement afin de ne pas compromettre les accès directs en ligne de commande au serveur de base de données…

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.

II. Les 8 étapes de la sécurité des bases de données

1. La découverte
2. Évaluation des vulnérabilités et de configuration
3. Renforcement
4. Audit des modifications
5. Surveillance de l'activité
6. Audit
7. Authentification, contrôle d'accès et gestion des droits
8. Chiffrement

THEMES :

1. Attaques de type pass the hash
2. Attaques par force brute
3. Attaque de type Password spraying
4. Password Auditor : outils de sécurité des mots de passe
5. Invicti : Analyseur de vulnérabilité
6. SCuba database Vulnerability scanner
7. DBdefense : outil de sécurié pour les bds sql server
8. OScanner : outil d'analyse et d'évaluation des bds oracle
9. dbForge Security Manager : outil de gestion de la sécurité pour Mysql

https://www.crowdstrike.fr/cybersecurity-101/privilege-escalation/

https://geekflare.com/fr/database-threats-and-prevention-tools/