République Algérienne Démocratique et Populaire

Ministère de l'enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahleb de Blida

Faculté de Math et Informatique

Département Informatique

Mémoire de fin d'étude en vue de l'obtention

Du diplôme de Master2 en Informatique

Spécialité : SIR

Système de contrôle d'accès basé sur le modèle RBAC

Présenté par : M^me^ Kebaili Fella.

M^elle^ Tazi Leila.

Devant le jury :

-Mme BOUSTIA Narimene Universite Saad Dahleb Blida	**Présidente**
-Mme MEZZI Melyara Universite Saad Dahleb Blida	**Examinatrice**
-Mr MENACER Djamel Eddine Ecole Supérieure d'Informatique-Oued.Semmar, Alger	**Promoteur**
Mr DAHAK Fouad Ecole Supérieure d'Informatique-Oued.Semmar, Alger	**CO-Promoteur**
-Mme ARKAM Meriem Universite Saad Dahleb Blida	**Encadreur**

Année Universitaire 2019-2020.

DEDICACES

Je dédie ce modeste travail à :

Ma Mère, Mon Mari et mes trois petits garçons.

Mes Frères et Sœurs.

A toute ma famille.

Tous mes amis.

A ma binôme Leila.

A tous ceux que m'en soutenus et encouragé.

F. KEBAILI

DEDICACES

Je dédie ce modeste travail à :

Mes très chers parents pour leurs affections et soutient durant toutes ces années.

A mes chères sœurs.

A Mes adorables nièces et neveux

A toute ma famille.

A Tous mes amis.

A ma binôme FELLA pour sa contribution pour moi.

A tous ceux que j'aime beaucoup.

A tous ceux que m'en soutenus et encouragé.

A Moi même

L. TAZI

Remerciements

Au terme de ce travail, nous tenons à remercier Dieu le tout puissant de nous avoir donné le courage, la volonté et la patience pour achever ce travail.

Nous avons l'honneur et le plaisir de présenter notre profonde gratitude et nos sincères remerciements à notre promoteur Mr D.E MENACER, Mr F. DAHAK Mme M.ARKAM, pour leurs précieuses aides, leurs orientations et le temps qu'il nous ont accordé tout au long de cette période.

Nous remercions profondément tous les enseignants qui nous ont encouragé et soutenus tout au long de cette année universitaire.

Nos remerciements les plus sincères et les plus profonds sont adressés au personnel de Département Informatique.

Nous remercions également tous ceux qui ont contribué de prêt ou de loin à l'achèvement de notre Travail.

Résumés

ملخص:

أصبح أمن الكمبيوتر قضية مهمة في تحضر نظم المعلومات. إنه يؤثر على طبقات نظام المعلومات (IS) والتطبيق والبنية التحتية. في جميع الحالات، يجب تأمين الخدمات والبيانات التي يجب أن تصل إليها هذه الخدمات.

في مشروعنا، اخترنا إعداد وحدة تحكم في الوصول بناءً على نموذج RBAC وتلك من أجل تأمين الوصول إلى قاعدة البيانات. تتكون سياستنا الأمنية من ثلاث وحدات أساسية: مصادقة المستخدم، والتحقق من التفويض، وإدارة سياسة RBAC.

تم تنفيذ النموذج خصيصًا لقواعد بيانات SQL Server ويهدف بشكل خاص إلى التحكم في الوصول إلى جداول قاعدة البيانات المحددة. أخيرًا ننتهي بعرض تقديمي واختبار للنظام.

الكلمات الرئيسية: الأمن، نظام الكمبيوتر، قاعدة البيانات، التحكم في الوصول، السياسة، خادم SQL،RBAC

RESUME :

La sécurité informatique devient un enjeu important dans l'urbanisation des systèmes d'information. Elle touche les couches SI (système d'information), Application et Infrastructure. Dans tous les cas, il faut sécuriser les services et les données auxquelles ces services accèdent.

Dans notre projet, nous avons choisi de mettre en place un contrôleur d'accès basé sur le modèle RBAC et ceux afin de sécuriser l'accès à la base de données. Notre politique de sécurité est constituée de trois modules essentiels : l'authentification des utilisateurs, vérification des autorisations et administration de la politique RBAC**.** La réalisation du modèle a été faite spécialement pour les bases de données SQL Serveur et vise spécialement à contrôler l'accès aux tables de la base de données sélectionnée. Enfin nous finissons par une présentation et un test du système.

Mots clés : Sécurité, Système informatique, Base de données, Contrôle d'accès, Politique, SQL Serveur, RBAC.

ABSTRACT:

Computer security is becoming an important issue in the urbanization of information systems. It affects the IS (information system), Application and Infrastructure layers. In all cases, the services and data to which these services access must be secured.

In our project, we chose to set up an access controller based on the RBAC model and those in order to secure access to the database. Our security policy is made up of three essential modules, user authentication, authorization verification, and RBAC policy administration. The realization of the model was made especially for SQL Server databases and specifically aims to control access to the tables of the selected database. Finally we end with a presentation and a test of the system.

Keywords : Security, Computer system, Database, Access control, Policy, SQL Server, RBAC.

La liste des figures

La liste des Tableaux

La liste des acronymes

Acronymes	Description
ABAC	**A**ttribute **B**ased **A**ccess **C**ontrol
ACL	**A**ccess **C**ontrol **L**ist
AJAX	**Asynchronous JavaScript And XML**
API	**A**pplication **P**rogramming **I**nterface
ASP	**A**ctive **S**erver **P**ages
BDS	**B**orland **D**eveloper **S**tudio
BLP	**B**ell-**L**a Padula **P**rivacy Model
CBAC	**C**ontext **B**ased **A**ccess **C**ontrol
CRBAC	**C**ontext **R**ole Based **A**ccess **C**ontrol
CSP	**C**ontent **S**ecurity **P**olicy
CSS	*Cascading Style Sheets*
DAC	**D**iscretionary **A**ccess **C**ontrol
DAST	**D**YNAMIC **A**PPLICATION **S**ECURITY **T**ESTING
DBX4	**DB**E**x**press 4
DDOS	**D**istributed **D**enial of **S**ervice
DOCX	**O**ffice **O**pen **X**ML **D**ocument
DOS	**D**enial **O**f **S**ervice
DTE	**D**omain and **T**ype **E**nforcement
ERP	**E**nterprise **R**esource **P**lanning
EXIF	**E**xchangeable **I**mage **F**ile
FTP	**F**ILE **T**RANSFER **P**ROTOCOL

GPS	**G**lobal **P**ositioning **S**ystem
HTML	**H**yper **T**ext **M**arkup **L**anguage
HTTP	**H**YPER **T**EXT **T**RANSFER **P**ROTOCOL
IBAC	**I**dentity **B**ased **A**ccess **C**ontrol
ID	**I**DENTIFIANT
IIS	**I**nternet **I**nformation **S**ervices
IOUG	***I**ndependent **O**racle **U**sers **G**roup*
IP	**I**nternet **P**rotocol
IPS	**I**ntrusion **P**revention **S**ystem
ITSEC	***I**nformation **T**echnology **S**ecurity **E**valuation **C**riteria*
JSON	**J**ava **S**cript **O**bject **N**otation
LDAP	**L**ightweight **D**irectory **A**ccess **P**rotocol
LRBAC	**L**ocation **A**ware **R**ole **B**ased **A**ccess **C**ontrol
MAC	**M**andatory **A**ccess **C**ontrol
OLE DB	Object Linking and Embedding, Database
OR-BAC	**O**rganization **R**ole **B**ased **A**ccess **C**ontrol
OSI	**O**pen **S**ystems **I**nterconnection
OWASP	**Open Web Application Security**
PHP	Hypertext Preprocessor
RBAC	**R**ole **B**ased **A**ccess **C**ontrol
RDBMS	**R**ELATIONAL **D**ATABASE **M**ANAGEMENT **S**YSTEM
REST	**R**epresentational **S**tate **T**ransfer
RGPD	**G**eneral **R**egulation **on D**ata **P**rotection

RPC	**R**emote **P**rocedure **C**all
SAP	**S**ystems, **A**pplications and **P**roducts for data processing
SAST	**S**tatic **A**pplication **S**ecurity **T**esting
SGBD	**S**ystème de **G**estion de **B**ase de **D**onnées
SGBDR	**S**ystème **de G**estion **de B**ase **de D**onnées **R**elationnel
SOAP	**S**imple **O**bject **A**ccess **P**rotocol
SOX	**S**ARBANES-**O**XLEY
SQL	**S**TRUCTURED **Q**UERY **L**ANGUAGE
SSRF	**S**ERVER **SID**E **R**EQUEST **F**ORGERY
SVG	**S**CALABLE **V**ECTOR **G**RAPHICS
TBAC	**T**ask **B**ased **A**ccess **C**ontrol
TMAC	**T**ea **M**based **A**ccess **C**ontrol
TR-BAC	**T**ask and **R**ôle **B**ased **A**ccess **C**ontrol
URL	**U**niform **R**esource **L**ocator
VCL	**V**ISUAL **C**OMPONENT **L**IBRARY
VPN	**V**irtual **P**rivate **N**etwork
WRBAC	**W**orflow **R**ole **B**ased **A**ccess **C**ontrôle
XML	**E**xtensible **M**arkup **L**anguage
DOM	**D**ocument **O**bject **M**odel
XSS	**C**ross **S**ite **S**cripting
XXE	**X**ML **E**XTERNAL **E**NTITY

Sommaire

TABLE DES MATIERES

Introduction générale

INTRODUCTION GENERALE :

Force est de constater que la sécurité des Systèmes d'informatique (SSI) est un sujet très débattu dans le monde informatique. Aujourd'hui, Ce concept et un enjeu pour les entreprises ainsi que pour l'ensemble des acteurs qui l'entourent. Il n'est plus confiné uniquement au rôle de l'informaticien. Sa finalité sur le long terme est de maintenir la confiance des utilisateurs. La finalité sur le moyen terme est la cohérence de l'ensemble du système d'informatique. Sur le court terme, l'objectif est que chacun ait accès aux informations dont il a besoin, le but est que chacun ait accès aux données dont il a besoin. Les normes actuelles insistent sur la disponibilité, l'intégrité, la confidentialité, la traçabilité, l'authentification et l'imputation. Une fois les objectifs atteints, les risques pesant sur chacun des éléments du système peuvent être estimés en fonction des menaces. Le niveau global de sécurité des systèmes d'information est défini par le niveau de sécurité du maillon le plus faible. Les précautions et contre-mesures doivent être envisagées en fonction des vulnérabilités propres au contexte auquel le système d'information est censé apporter service et appui. Il faut pour cela estimer, d'une part, la gravité des conséquences au cas où les risques se réaliseraient et d'autre part la vraisemblance des risques (ou leur potentialité, ou encore leur probabilité d'occurrence).

La sécurité informatique devient un enjeu important dans l'urbanisation des systèmes d'information. Elle touche les couches SI (système d'information), Application et Infrastructure. Dans tous les cas, il faut sécuriser les **services** et les **données** auxquelles ces services accèdent.

Dans le cas d'un serveur de bases de données, il faut sécuriser plusieurs niveaux :

- Les données stockées sur le serveur de bases de données (fichiers).
- Le service SGBD (par ex. PostegresSQL ou SQL Server).
- La communication réseau entre un client (qui accède aux données selon le rôle défini : lecture de données, modification...) et un serveur de base de données (canal de communication).

Dans notre projet intitulé sécurité d'un serveur de base de données, Pour réaliser une force de contremesures efficace, nous proposons d'intégrer les solutions suivantes :

1-Protection de la base de données avec une mise en place d'un contrôleur d'accès basé sur les rôles (RBAC), et cela afin d'assurer que les utilisateurs de la base de données n'accèdent qu'aux données par lesquels ils sont concernés.

2-Sécurisation des communications avec un canal TLS/PKI (au lieu d'un mur de pare-feu traditionnel)

3-Fiabilité et sécurité du service SGBD avec une solution cluster de Fail-Over derrière un mur de pare-feu NG (New Generation, filtrage multi-niveaux et IDS/IPS).

Notre mémoire est structuré une introduction générale, cinq chapitres organisés comme suit :

- Dans le premier chapitre, nous abordons la sécurité des systèmes informatique de façon générale à savoir les objectifs de la sécurité, les vulnérabilités, les menaces, les attaques et les contre-mesures.
- Dans le deuxième chapitre, nous abordons la sécurité des applications Web à savoir les architectures existantes, les attaques les plus critique selon OWASP, leurs impacts et les contre-mesures de chaque attaque.
- Le troisième chapitre aborde la sécurité des bases données, dans ce chapitre nous allons voir les principales menaces de bases de données, les attaques auxquelles elles sont exposées et comment y faire face.
- Le quatrième présente une analyse des différentes politiques de contrôle d'accès largement utilisées dans le monde de sécurité informatique, nous expliquons les différents modèles de contrôle d'accès largement utilisés dans le monde industriel (les modèles classiques et les modèles à base de rôle) en présentant ces principes de fonctionnement et ces points faibles.
- Le cinquième et dernier chapitre, lui explique l'environnement de travail et les outils de développement, les différentes étapes de la mise en place de notre application RBAC, et quelques captures d'écrans des principales fonctionnalités.
- Et en fin, une conclusion qui comporte les apports de notre travail ainsi que les perspectives envisagées

Chapitre 01 : Sécurité des systèmes Informatiques

1. INTRODUCTION :

L'information se présente sous trois formes les données, les connaissances et les messages. On désigne par système d'informatique l'ensemble des moyens informatiques et de télécommunication ayant pour finalité d'élaborer, traiter, stocker, acheminer, présenter ou détruire des données.

Le concept de sécurité des systèmes informatiques recouvre un ensemble de méthodes, techniques et d'outils chargés de protéger les ressources d'un système d'information afin d'assurer la disponibilité des services, la confidentialité des informations et l'intégrité des systèmes. Cependant, avec l'ouverture des entreprises et des personnes à Internet, l'assurance de la sécurité des systèmes devient très difficile, du fait que, les attaques et les intrusions augmentent de plus en plus et deviennent de plus en plus complexes et difficiles à éviter.

Ce chapitre présente les notions de base de la sécurité des systèmes informatiques à savoir, les domaines d'application de la sécurité, les différents types attaques, les types d'attaquants ainsi que les mécanismes et outils pouvant être mis en place pour assurer la sécurité.

2. DEFINITION LA SECURITE INFORMATIQUE :

La sécurité informatique est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaire mis en œuvre pour réduire la vulnérabilité d'un système informatique contre les menaces accidentelles ou intentionnelles. L'objectif de la sécurité informatique est d'assurer que les ressources matérielles et/ou logicielles d'un parc informatique soient uniquement utilisées dans le cadre prévu et par des utilisateurs autorisés **[1].** Il convient d'identifier les exigences fondamentales en sécurité informatique, qui caractérisent ce à quoi s'attendent les utilisateurs de systèmes informatiques au regard de la sécurité à savoir :

2.1. L'AUTHENTIFICATION :

L'authentification doit permettre de vérifier l'identité d'une entité afin de s'assurer entre autres, de l'authenticité de celle-ci. Pour cela, l'entité devra prouver son identité, le plus souvent en donnant une information spécifique qu'elle est censée être seule à détenir telle que, par exemple, un mot de passe ou une empreinte biométrique **[7],** Ensuite on passe au contrôle d'accès comme illustré dans la figure ci-dessous.

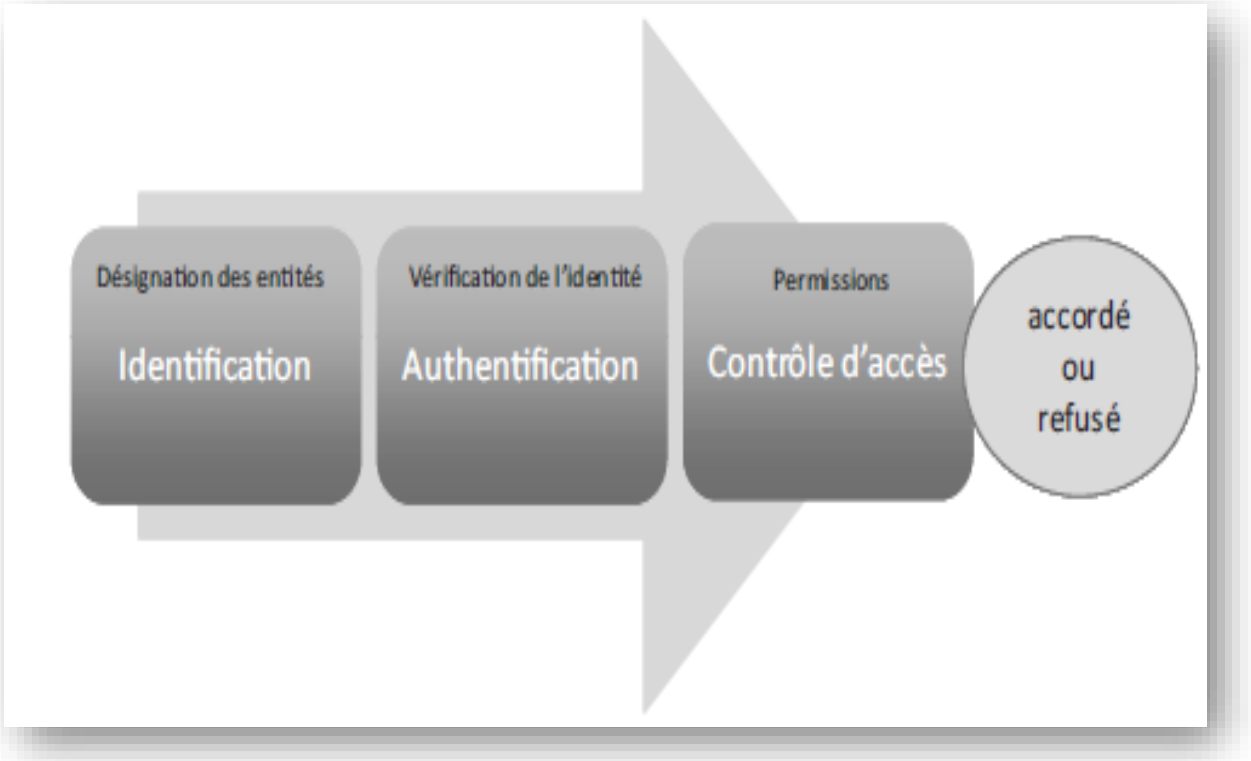

Figure 1.1. *Identification et authentification* **[7].**

2.2. LA CONFIDENTIALITE :

C'est le maintien du secret des informations peut être vue comme la « protection des données contre une divulgation non autorisée ». Il existe deux types d'actions complémentaires permettant d'assurer la confidentialité des données **[7]** :

- Limiter et contrôler leurs accès afin que seules les personnes habilitées à les lire ou à les modifier puissent le faire.
- Les rendre inintelligibles en les chiffrant de telle sorte que les personnes qui ne sont pas autorisées à les déchiffrer ne puissent les utiliser.

2.3. L'INTEGRITE :

Le critère d'**intégrité** des ressources physiques et logiques (équipements, données, traitements, transactions, services) est relatif au fait qu'elles n'ont pas été détruites (altération totale) ou modifiées (altération partielle) à l'insu de leurs propriétaires tant de manière intentionnelle qu'accidentelle. Une fonction de sécurité appliquée à une ressource pour contribuer à préserver son intégrité, permettra de la protéger plus ou moins efficacement contre une menace de corruption ou de destruction **[7].**

2.4. LA NON-REPUDIATION :

C'est le fait de ne pouvoir nier ou rejeter qu'un événement (action, transaction) a eu lieu. À ce critère de sécurité peuvent être associées les notions d'imputabilité, de *traçabilité* ou encore parfois *d'auditabilité* **[7]**.

- L'**imputabilité** se définit par l'attribution d'une action (un événement) à une entité déterminée (ressource, personne).
- La **traçabilité** permet de suivre la trace numérique laissée par la réalisation d'un événement (message électronique, transaction commerciale, transfert de données…).
- L'**auditabilité** se définit par la capacité d'un système à garantir la présence d'informations nécessaires à une analyse ultérieure d'un événement (courant ou exceptionnel) effectuée dans le cadre de procédures de contrôle spécifiques et d'audit.

2.5. LA DISPONIBILITE :

La **disponibilité** d'une ressource est relative à la période de temps pendant laquelle le service offert est opérationnel. Elle se mesure par un pourcentage qui se calcule en divisant la durée durant laquelle cet équipement est opérationnel par la durée durant laquelle il aurait dû être opérationnel. Il ne suffit pas qu'une ressource soit disponible, elle doit pouvoir être utilisable avec des temps de réponse acceptables. Sa disponibilité est indissociable de sa capacité à être accessible par l'ensemble des ayants droit **[7]**.

En conclusion, on mesure la sécurité d'un système entier à la sécurité du maillon le plus faible. Ainsi, si tout un système est sécurisé techniquement mais que le facteur humain, souvent mis en cause, est défaillant, c'est toute la sécurité du système qui est remise en cause. On peut schématiser les critères de sécurité par la figure suivante :

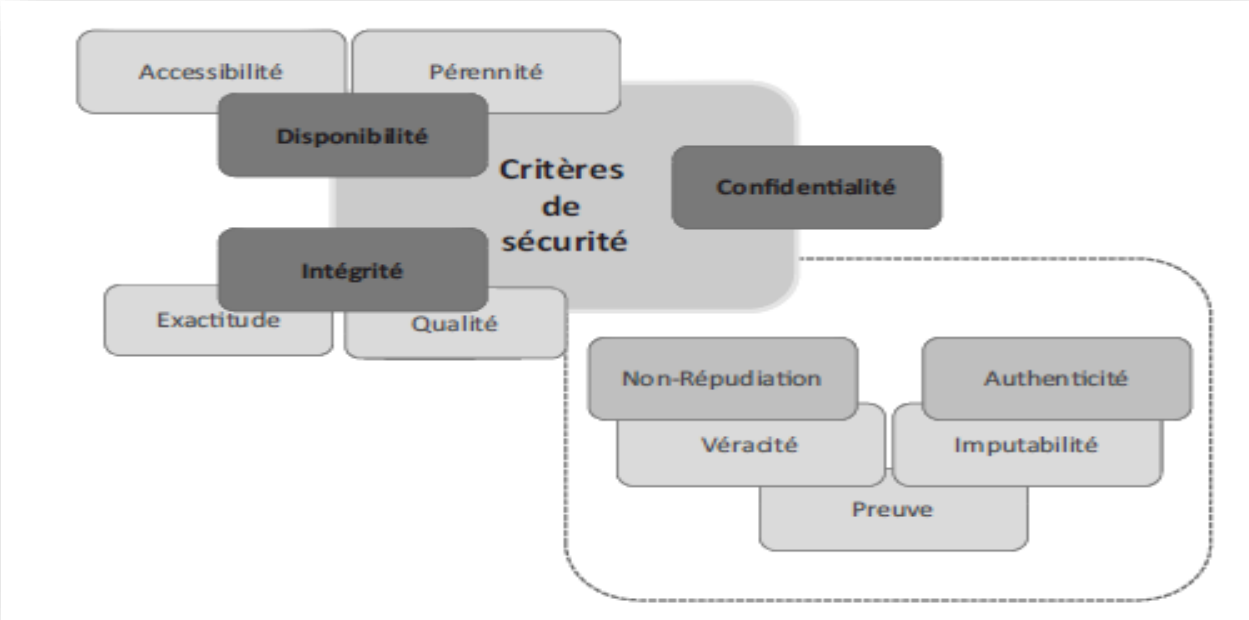

Figure 1.2. *Critères de sécurité* **[7].**

3. DOMAINES D'APPLICATION DE LA SECURITE INFORMATIQUE :

Pour une entreprise, toutes les sphères d'activité de l'informatique et des réseaux de communication sont concernées par la sécurité. En fonction de son domaine d'application, la sécurité informatique se décline en **[2] :**

- **Sécurité physique et environnementale**

La sécurité physique et environnementale concerne tous les aspects liés à la maîtrise des systèmes et de l'environnement dans lesquels ils se situent **[2].**

- **Sécurité de l'exploitation**

La sécurité de l'exploitation doit permettre un bon fonctionnement opérationnel des systèmes informatiques. Cela comprend la mise en place d'outils et de procédures relatifs aux méthodologies d'exploitation, de maintenance, de test, de diagnostic, de gestion des performances, de gestion des changements et des mises à jour **[2]**.

- **Sécurité applicative, sécurité logique et sécurité de l'information**

La sécurité applicative comprend le développement pertinent de solutions logicielles (ingénierie du logiciel, qualité du logiciel) ainsi que leur intégration et exécution harmonieuses dans des environnements opérationnels **[2]**.

La sécurité logique fait référence à la réalisation de mécanismes de sécurité par logiciel contribuant au bon fonctionnement des programmes, des services offerts et à la protection des données **[2]**.

La sécurité de l'information, c'est avant tout comprendre le rôle de l'information elle-même, son importance stratégique et l'impact des décisions qui la concernent. C'est

également assurer son exactitude et sa pérennité pour le temps nécessaire à son exploitation et à son archivage **[2]**.

- **Sécurité des infrastructures informatiques et de communication**

Un environnement informatique et de télécommunication sécurisé implique la sécurisation de tous les éléments qui le compose. La sécurité est toujours celle du maillon le plus faible. Implanter des mécanismes de chiffrement pour rendre les données transférées confidentielles est de peu d'utilité si d'aucun peut y accéder lorsqu'elles sont manipulées par des plates-formes matérielles et logicielles non correctement sécurisées. L'implantation de mesures de sécurité doit répondre à des besoins de sécurité clairement identifiés à la suite d'une analyse des risques auxquels peut être exposée une organisation **[2].**

4.TERMINOLOGIE DE LA SECURITE INFORMATIQUE :

4.1. VULNERABILITE :

C'est une faille ou une faiblesse dans la conception, l'implémentation, le fonctionnement ou la gestion d'un système qui pourrait être exploitée pour compromettre les objectifs de sécurité du système **[8]**.

4.2. MENACE :

Une menace peut être un attaquant externe malveillant, un utilisateur interne, une instabilité du système, etc. enfin tout ce qui peut nuire aux actifs appartenant à une application (ressources de valeur, telles que les données dans une base de données ou dans le fichier système) en exploitant une vulnérabilité **[8]**.

4.3. RISQUE :

C'est la probabilité qu'une menace exploite une vulnérabilité. Autrement dit, c'est une possibilité qu'un fait dommageable se produise **[8]**.

4.4. ATTAQUE :

C'est l'exploitation d'une vulnérabilité à des fins non connues de l'exploitant du système et généralement pour nuire **[8]**.

4.5. CONTRE-MESURE D'UNE ATTAQUE :

C'est l'ensemble des actions mises en œuvres afin de réduire le risque dans une organisation **[8]**.

Les points justement cités sont la base pour une bonne définition de la gestion du risque, nous pouvons voir les flux de gestion des risques dans la figure suivante :

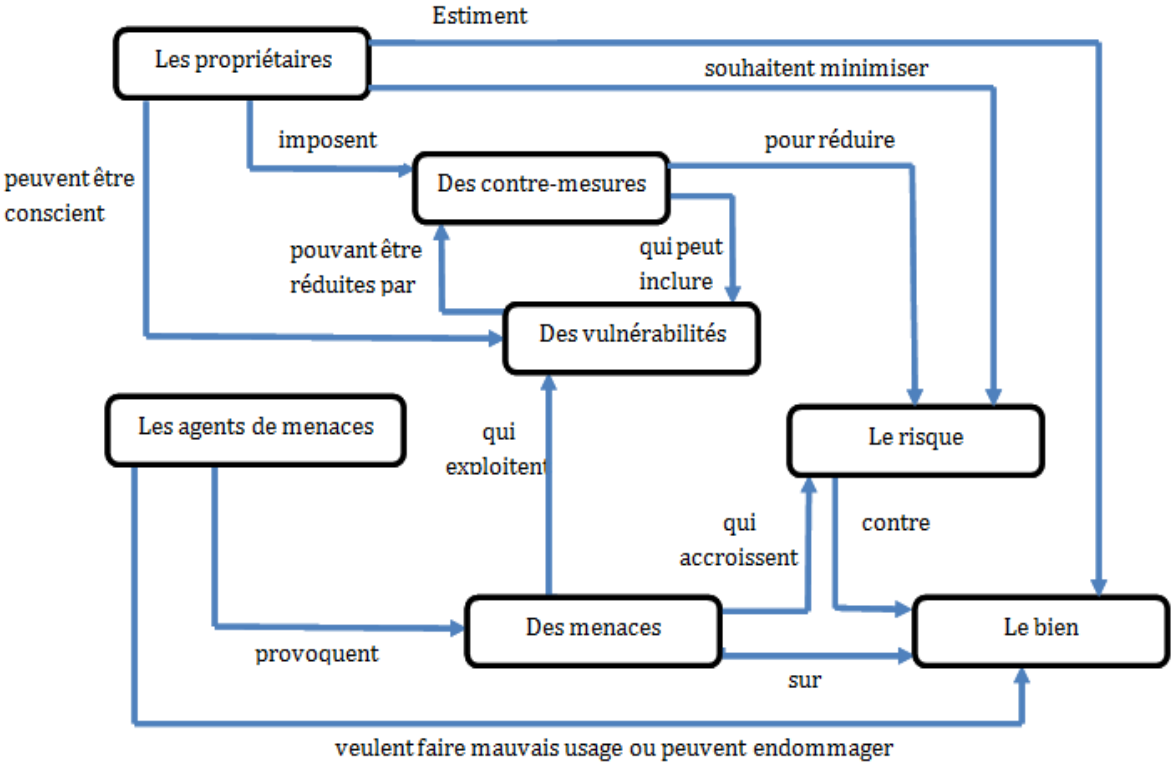

Figure1.3. *Flux de gestion de risque* ***[8]***

5.LES ATTAQUES :

5.1. TYPES D'ATTAQUE :

Les attaques peuvent à première vue être classées en deux grandes catégories **[3]** :

5.1.1. Les attaques passives :

Consistent à écouter sans modifier les données ou le fonctionnement du réseau. Elles sont généralement indétectables mais une prévention est possible.

5.1.2. Les attaques actives :

Consistent à modifier des données ou des messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau. Noter qu'une attaque active peut être exécutée sans la capacité d'écoute. De plus, il n'y a généralement pas de prévention possible pour ces attaques, bien qu'elles soient détectables (permettant ainsi une réponse adéquate).

5.2. ETAPES D'UNE ATTAQUE :

Le piratage est généralement divisé en 5 phases principales, comme illustré dans la figure ci-dessous, Dans ce qui suit nous allons présenter les étapes d'une attaque :

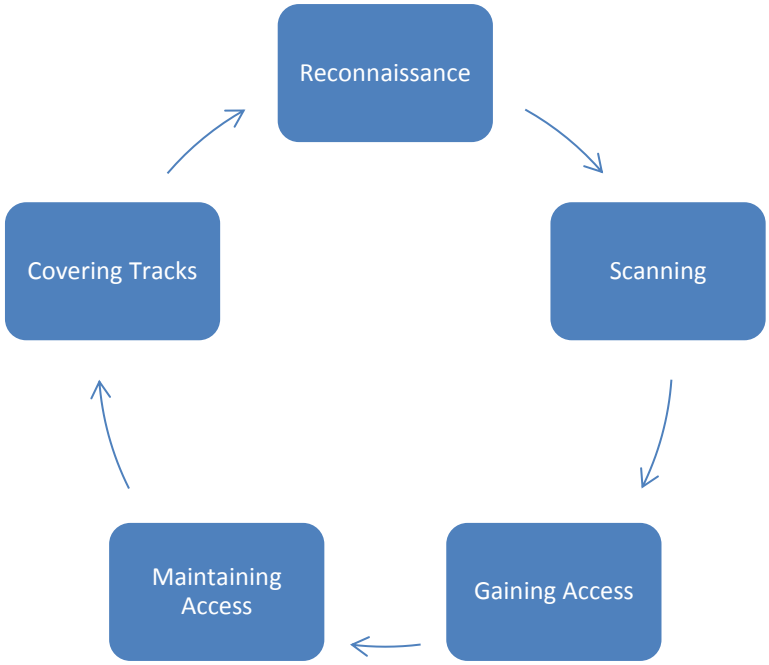

Figure1.4. *Etapes d'une attaque* ***[3]***

5.2.1. Reconnaissance :

Il s'agit de la première phase du piratage. Cette phase est également appelée étape d'enquête. Dans cette phase, le pirate essaie de collecter les informations de la cible illustrée à la figure 1.4 Cette phase peut inclure l'identification de la cible, la découverte de la plage d'adresses IP, du réseau ou du système de noms de domaine de la cible.

5.2.2. Balayage (Scanning) :

La collecte de plus d'informations à l'aide de techniques de reconnaissance complexes et agressives est appelée analyse. L'analyse est un ensemble d'étapes et de méthodes permettant d'identifier les hôtes actifs, les ports, les services et de découvrir les systèmes d'exploitation et l'architecture du système cible. Identifier les vulnérabilités, les menaces dans le réseau en scannant qui est utilisé pour créer un profil de l'organisation cible.

5.2.3. Gain d'accès (Gainnig access)

Dans cette phase, le pirate conçoit ou développe le plan ou l'architecture du réseau de la cible à l'aide de la phase 1 et de la phase 2. Le pirate a terminé la liste et l'analyse du réseau et, maintenant, le pirate a quelques options pour accéder au système ciblé.

5.2.4. Maintenir l'accès (Maintaining access)

C'est la phase la plus avancée du piratage. Car, une fois qu'un pirate a obtenu l'accès au système ciblé, le plus important est de conserver cet accès pour de futures attaques malveillantes. Lorsque le pirate a obtenu l'accès au système, il peut également utiliser ce dernier comme base pour d'autres attaques malveillantes.

5.2.5. Cacher les traces (Hiding Traces) :

La dernière phase du piratage consiste à cacher les pistes ou à effacer les traces. Effacer les pistes signifie qu'on peut les atteindre. L'attaquant doit changer son adresse IP et exécuter sa machine d'attaque ou son outil d'attaque via VPN pour couvrir ou cacher son identité. Cette phase est essentielle car l'attaque directe est bruyante et clairement identifiée par la cible.

5.3. TYPES D'ATTAQUANTS

Un pirate informatique est une personne qui s'intéresse intensément au fonctionnement de tout système d'exploitation informatique. Les pirates sont le plus souvent des programmeurs. Ils acquièrent des connaissances avancées des systèmes d'exploitation et des langages de programmation et découvrent les failles au sein des systèmes et les raisons de ces lacunes. Les pirates peuvent être largement classés en fonction de la raison pour laquelle ils piratent le système. Il existe trois types des pirates sur cette base comme le montre la figure suivante **[4]:**

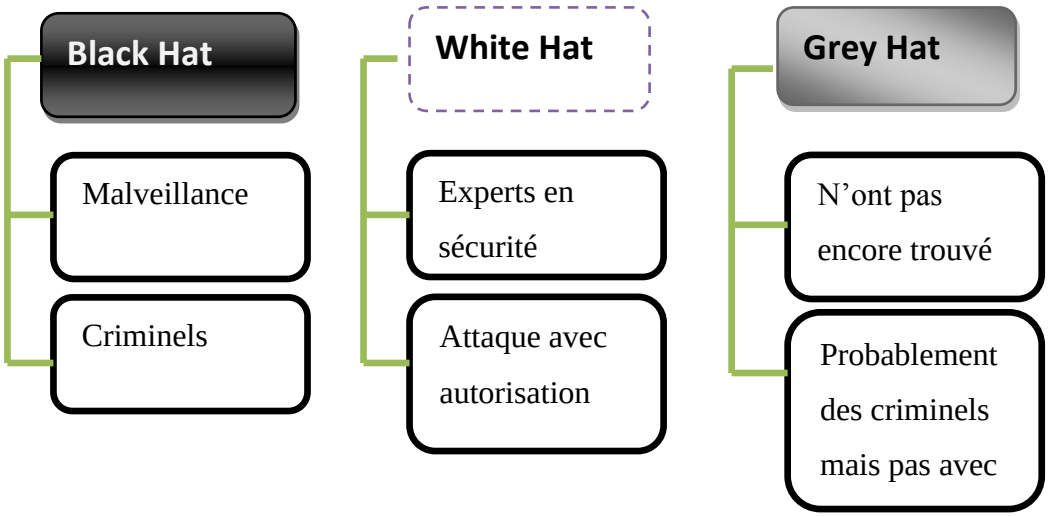

Figure1.5. *Types d'attaquants* ***[8]***

5.3.1. White-Hat :

Les **White-Hat** sont ceux qui sont autorisés ou les hackers certifiés qui travaillent pour le gouvernement et les organisations en effectuant des tests de pénétration et en identifiant les failles dans leur sécurité. Ils assurent également la protection contre les cyber crimes. Ils travaillent selon les règles et réglementations fournies par le gouvernement, c'est pourquoi ils sont appelés des pirates éthiques ou des experts en cyber sécurité.

5.3.2. Black-Hat:

Ils sont souvent appelés Crackers. Les hackers **Black-Hat** peuvent obtenir l'accès non autorisé à un système et détruire les données vitales. La méthode d'attaque qu'ils utilisent utilise des pratiques de piratage courantes qu'ils ont apprises plus tôt. Ils sont considérés comme des criminels et peuvent être facilement identifiés en raison de leurs actions malveillantes.

5.3.3. Grey-Hat :

Les **Grey-Hat** tombent quelque part dans la catégorie entre les **White-Hat** et les **Black-Hat**. Ce ne sont pas des hackers légalement autorisés. Ils travaillent avec de bonnes et de mauvaises intentions ; ils peuvent utiliser leurs compétences à des fins personnelles. Tout dépend du hacker. Si un Grey-Hat utilise ses compétences pour ses gains personnels, il est considéré comme un Black-Hat.

6. LES CYBERATTAQUES

Une **cyberattaque** est tout type d'action offensive qui vise des systèmes, des infrastructures ou des réseaux informatiques, ou encore des ordinateurs personnels, en s'appuyant sur diverses méthodes pour voler, modifier ou détruire des données ou des systèmes informatiques. Aujourd'hui, on décrit les types de cyber attaques les plus courantes [**5**]:

- **Attaques par déni de service (DoS) et par déni de service distribué (DDoS) :**

Une attaque par déni de service submerge les ressources d'un système afin que ce dernier ne puisse pas répondre aux demandes de service. Une attaque DDoS vise elle aussi les ressources d'un système, mais elle est lancée à partir d'un grand nombre d'autres machines hôtes infectées par un logiciel malveillant contrôlé par l'attaquant. Il existe différents types d'attaques DoS et DDoS, les plus courantes sont les **attaques *SYN flood*, les attaques *teardrop*, les attaques par rebond, le pingof death**et **les *botnets***.

- **Attaque de l'homme au milieu (MitM) :**

Une attaque de l'homme du milieu est un pirate qui s'insère dans les communications entre un client et un serveur. Il existe différents types d'attaques de l'homme du milieu, les plus courantes sont : ***Détournement de session, Usurpation d'IP***et***Relecture.***

- **Attaques phishing et spearphishing :**

L'hameçonnage consiste à envoyer des e-mails qui semblent provenir de sources fiables dans le but d'obtenir des informations personnelles ou d'inciter les utilisateurs à faire quelque chose. Cette technique combine ingénierie sociale et stratagème technique. Elle peut impliquer une pièce jointe à un e-mail, qui charge un logiciel malveillant sur un ordinateur. Elle peut également utiliser un lien pointant vers un site Web illégitime qui incite à télécharger des logiciels malveillants ou à transmettre des renseignements personnels.

- **Attaque par Drive by Download :**

Les attaques par téléchargement furtif sont une méthode courante de propagation des logiciels malveillants. Les pirates recherchent des sites Web non sécurisés et insèrent un script malveillant dans le code HTTP ou PHP de l'une des pages. Ce script peut installer des logiciels malveillants directement sur l'ordinateur d'un visiteur du site, ou rediriger celui-ci vers un site contrôlé par les pirates. Des téléchargements furtifs peuvent survenir lors de la visite d'un site Web ou de l'affichage d'un e-mail ou d'une fenêtre pop-up.

- **Attaque par mot de passe :**

Les mots de passe étant le mécanisme le plus couramment utilisé pour authentifier les utilisateurs d'un système informatique, l'obtention de mots de passe est une approche d'attaque courante et efficace. Le mot de passe d'une personne peut être obtenu en fouillant le bureau physique de la personne, en surveillant la connexion au réseau pour acquérir des mots de passe non chiffrés, en ayant recours à l'ingénierie sociale, en accédant à une base de données de mots de passe ou simplement en devinant.

- **Attaque par injection SQL :**

L'injection SQL est devenue un problème courant qui affecte les sites Web exploitant des bases de données. Elle se produit lorsqu'un malfaiteur exécute une requête SQL sur la base de données via les données entrantes du client au serveur. Des commandes SQL sont insérées dans la saisie du plan de données (par exemple, à la place du nom d'utilisateur ou du mot de passe) afin d'exécuter des commandes SQL prédéfinies. Un exploit d'injection SQL réussi peut lire les données sensibles de la base de données, modifier (insérer, mettre à jour ou supprimer) les données de la base de données, exécuter des opérations d'administration de la base de données (par exemple la fermer), récupérer le contenu d'un fichier spécifique, et, dans certains cas, envoyer des commandes au système d'exploitation.

- **Attaque XSS (Cross-site scripting) :**

Les attaques XSS utilisent des ressources Web tierces pour exécuter des scripts dans le navigateur Web de la victime ou dans une application pouvant être scriptée. Plus précisément, l'attaquant injecte un JavaScript malveillant dans la base de données d'un site Web. Lorsque la victime demande une page du site Web, le site Web transmet la page à son navigateur avec le script malveillant intégré au corps HTML. Le navigateur de la victime exécute ce script, qui envoie par exemple le cookie de la victime au serveur de l'attaquant, qui l'extrait et l'utilise pour détourner la session. Les conséquences les plus graves se produisent lorsque XSS sert à exploiter des vulnérabilités supplémentaires. Ces vulnérabilités peuvent non seulement permettre à un attaquant de voler des cookies, mais aussi d'enregistrer les frappes de touches et des captures d'écran, de découvrir et de collecter des informations réseau et d'accéder et de contrôler à distance l'ordinateur de la victime.

- **Attaque par écoute illicite**

Les écoutes clandestines sont le résultat d'une interception du trafic réseau. Elles permettent à un attaquant d'obtenir des mots de passe, des numéros de carte bancaire et d'autres informations confidentielles qu'un utilisateur envoie sur le réseau. Elles peuvent être passives ou actives :

- **Écoute clandestine passive** – Un pirate détecte des informations en écoutant la transmission de messages sur le réseau.
- **Écoute clandestine active** – Un pirate s'empare activement d'informations en se faisant passer pour une unité amie et en envoyant des requêtes aux transmetteurs. On appelle cela sonder, scanner ou saboter.

- **Attaque d'anniversaire**

Les attaques des anniversaires sont lancées contre les algorithmes de hachage qui vérifient l'intégrité d'un message, d'un logiciel ou d'une signature numérique. Un message traité par une fonction de hachage produit une synthèse du message de longueur fixe, indépendante de la longueur du message entrant ; cette synthèse caractérise de façon unique le message. L'attaque des anniversaires fait référence à la probabilité de trouver deux messages aléatoires qui génèrent la même synthèse lorsqu'ils sont traités par une fonction de hachage. Si un attaquant calcule la même synthèse pour son message que l'utilisateur, il peut tout à fait remplacer le message de l'utilisateur par le sien, et le destinataire ne sera pas en mesure de détecter le remplacement, même s'il compare les synthèses.

- **Attaque par des logiciels malveillants**

Un logiciel malveillant peut être décrit comme un logiciel indésirable installé dans un système sans aucun consentement. Il peut s'attacher à un code légitime et se propager, se cacher dans des applications utiles ou se reproduire sur Internet. Voici quelques-uns des types de logiciels malveillants les plus courants :

- **Macro-virus** : Ils infectent des applications comme Microsoft Word ou Excel.
- **Infecteurs de fichiers** : Ils s'attachent généralement à des codes exécutables, comme les fichiers .exe.
- **Infecteurs de système ou de secteur d'amorçage** : Ils s'attachent au secteur d'amorçage principal des disques durs.
- **Virus polymorphes** : Ils se cachent dans divers cycles de chiffrement et de déchiffrement.
- **Virus furtifs** : Ils prennent le contrôle de certaines fonctions du système pour se dissimuler. Pour ce faire, ils compromettent les logiciels de détection des logiciels malveillants de telle sorte que ceux-ci signalent qu'une zone infectée n'est pas infectée.
- **Chevaux de Troie** : Ce sont des programmes qui se cachent dans un programme utile et qui ont généralement une fonction malveillante. Une différence majeure entre les virus et les chevaux de Troie est que ces derniers ne se répliquent pas d'eux-mêmes.
- **Bombe logique** : Il s'agit d'un type de logiciel malveillant ajouté à une application et qui est déclenché par un événement spécifique, comme une condition logique ou une date et une heure spécifiques.
- **Vers** : Ils diffèrent des virus en ce qu'ils ne s'attachent pas à un fichier hôte, ce sont des programmes autonomes qui se propagent sur les réseaux et les ordinateurs. Les vers se propagent généralement via les pièces jointes aux e-mails.
- **Injecteurs** : Ce sont des programmes utilisés pour installer des virus sur les ordinateurs. Dans de nombreux cas, l'injecteur n'est pas infecté par un code malveillant et peut donc ne pas être détecté par un logiciel antivirus. Un injecteur peut également se connecter à Internet et télécharger des mises à jour de logiciel antivirus se trouvant sur un système compromis.
- **Rançongiciels (ransomware)** : Il s'agit d'un type de logiciel malveillant qui bloque l'accès aux données de la victime et menace de les publier ou de les supprimer à moins qu'une rançon ne soit versée.

- **Logiciels publicitaires (adware)** – Ce sont des applications logicielles utilisées par les entreprises à des fins de marketing ; des bannières publicitaires sont affichées pendant l'exécution d'un programme. Les logiciels publicitaires peuvent être téléchargés automatiquement sur un système lorsque on navigue sur un site Web quelconque et peuvent s'afficher dans des fenêtres contextuelles ou dans une barre qui apparaît automatiquement sur l'écran de l'ordinateur.
- **Logiciels espions (spyware)** – Ce sont des programmes installés pour recueillir des informations sur les utilisateurs, leurs ordinateurs ou leurs habitudes de navigation. Ils surveillent tout ce que on fait et envoient les données à un utilisateur distant. Ils peuvent également télécharger et installer d'autres programmes malveillants depuis Internet. Les logiciels espions fonctionnent comme les logiciels publicitaires, mais il s'agit généralement d'un programme distinct qui s'installe lorsque on installe une application gratuite.

7. MECANISMES DE SECURITE :

Un mécanisme est conçu pour détecter, prévenir et lutter contre une attaque de sécurité en d'autre terme il fournit et supporte les services de sécurité, on en distingue deux classes : mécanismes spécifiques à certains services et mécanismes génériques (voir ***Figure1.6***) **[9]**.

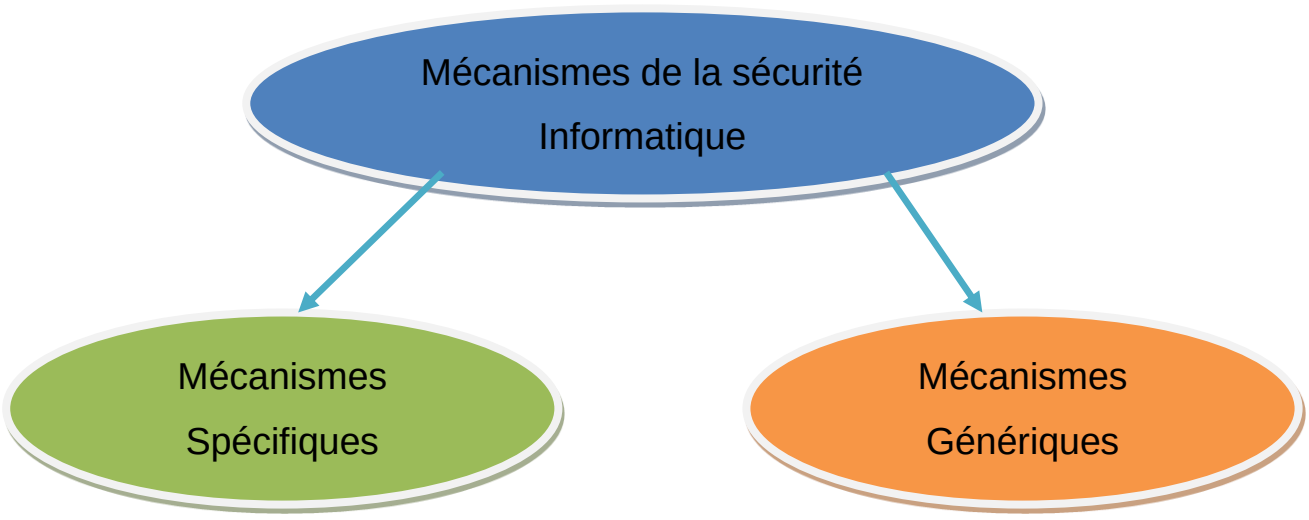

Figure1.6. *Types mécanismes de sécurité informatique*

- **Mécanismes spécifiques**

Les mécanismes suivants peuvent être incorporés dans la couche (N) appropriée pour fournir les services de sécurité cités ci-dessus. On en distingue huit types(***Figure1.7***) **[9]** :

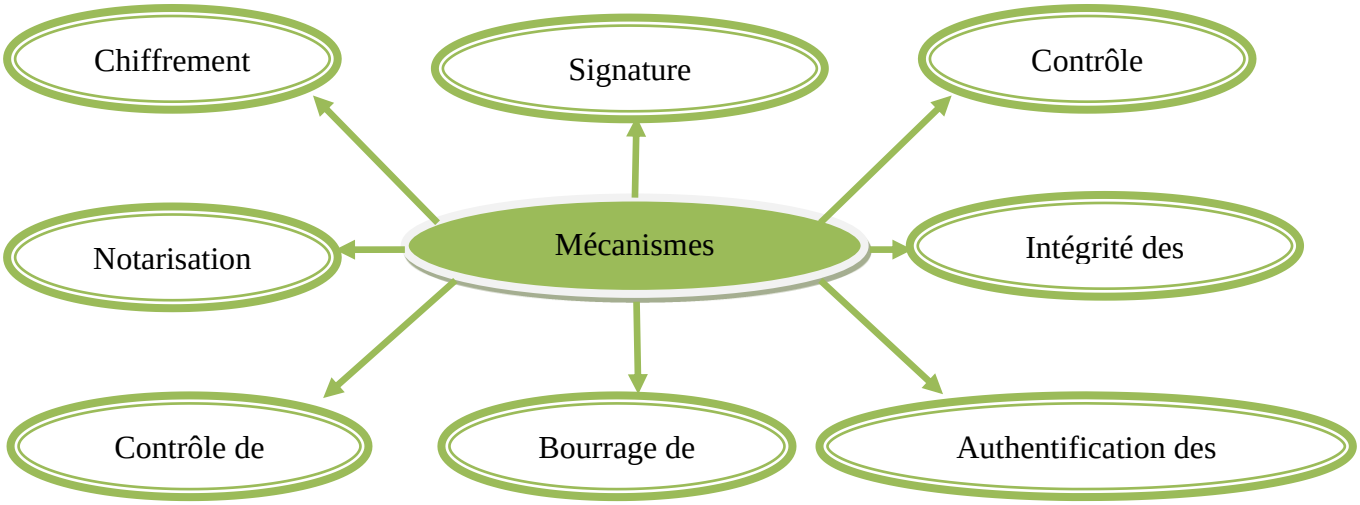

Figure1.7. *Mécanismes spécifiques*

- **Chiffrement :**

C'est l'utilisation d'algorithmes mathématiques pour transformer les données en une forme qui n'est pas facilement intelligible. La transformation et la récupération ultérieure des données dépendent d'un algorithme et de zéro ou plusieurs clés de chiffrement :

-Fournit la confidentialité des données ou bien flux de données.

-Peut jouer un rôle dans un certain nombre d'autres mécanismes de sécurité ou les compléter.

- **Signature numérique :**

Ce mécanisme définit deux procédures :

- Signature d'une unité de données.
- Vérification d'une unité de données signées.

La caractéristique essentielle du mécanisme de signature est que la signature ne peut être produite qu'en utilisant l'information privée du signataire. Par conséquent, lorsqu'on vérifie la signature, on peut prouver, par la suite, à une tierce partie (par exemple, un juge ou un arbitre), à tout moment, que seul le détenteur unique de l'information privée peut avoir produit la signature.

- **Contrôle d 'accès :**

Dans un système informatique, l'autorisation a pour but de permettre que les actions légitimes, c'est à dire empêcher qu'un utilisateur puisse exécuter des opérations qui ne lui sont pas permises. Pour définir quelles sont les opérations autorisées et celles qui sont interdites, il

faut établir une politique de **contrôle d'accès**. Le standard européen **ITSEC** définit une politique de sécurité comme étant "l'ensemble des lois, règles et pratiques qui régissent la façon dont l'information sensible et les ressources sont gérés, protégés et distribués à l'intérieur d'un système spécifique" (**Figure1.8**).

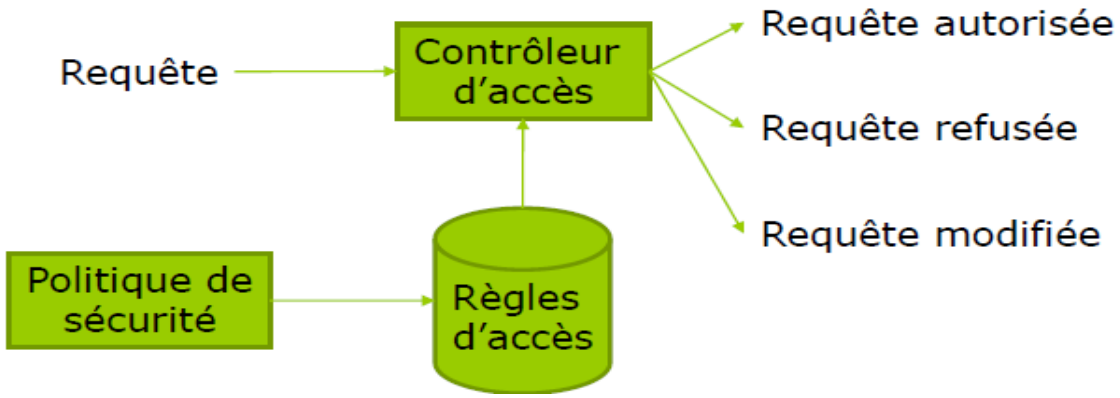

***Figure1.8**. P*rincipe du contrôle d'accès **[10].**

Plusieurs types ont été proposés pour répondre aux besoins de contrôle d'accès des applications, ces types sont généralement classés en trois types :

- Le **DAC** (Discretionary Access Control).
- Le **MAC** (Mandatory Access Control).
- Le **RBAC** (Role Based Access Control).

Nous détaillerons plus cette section dans le chapitre 4, car le mécanisme de contrôle d'accès basé sur les rôles (**RBAC**), fait l'objet de la partie implémentation de notre travail.

- **Intégrité des données**

La détermination de l'intégrité d'une unité de données unique implique deux processus, l'un au niveau de l'entité émettrice et l'autre au niveau de l'entité réceptrice. L'entité émettrice ajoute à une unité de données une grandeur qui est une fonction de la donnée. Cette grandeur peut être une information supplémentaire, tel qu'un code de contrôle par bloc ou une valeur de contrôle cryptographique, et peut elle-même être chiffrée. L'entité réceptrice génère une quantité correspondante et la compare à la grandeur reçue pour déterminer si les données ont été modifiées pendant le transit. Ce mécanisme seul ne protégera par contre pas, contre la répétition d'une seule unité de données. Dans les couches appropriées de l'architecture, la détection d'une manipulation peut conduire à une action de reprise (par exemple, via une retransmission ou une correction d'erreur) au niveau de cette couche ou au niveau d'une couche supérieure :

- Pour le transfert de données en mode connexion, la protection de l'intégrité d'une séquence d'unités de données nécessite en outre une certaine forme de séquencement explicite telle que la numérotation de séquence, l'horodatage ou le chaînage cryptographique.

- Pour la transmission de données en mode sans connexion, l'horodatage peut être utilisé pour assurer une forme de protection limitée contre le fait de répéter des unités de données individuelles.

- **Authentification des échanges**

Un mécanisme destiné à garantir l'identité d'une entité au moyen d'un échange d'informations. Certaines des techniques qui peuvent être appliquées aux échanges d'authentification sont les suivantes :

- Utilisation d'information d'authentification, telle que mots de passe – fournis par une entité émettrice et contrôlés par l'entité réceptrice.

- Techniques cryptographiques.

- Utilisation de caractéristiques et/ou de ce qui est propre à l'entité.

Le choix des techniques d'échange d'authentification dépendra des circonstances dans lesquelles elles seront utilisées. Très souvent, il sera nécessaire de les utiliser avec :

- Horodatage et horloges synchronisées.

-Deux et trois échanges (respectivement pour l'authentification unilatérale et mutuelle).

- Des services de non-répudiation réalisés par signature numérique et/ou mécanismes de notarisation.

- **Bourrage de trafic**

Les mécanismes de bourrage peuvent être utilisés pour assurer différents niveaux de protection contre l'analyse du trafic. Ce mécanisme ne peut être efficace que si le bourrage est protégé par un service de confidentialité.

- **Contrôle du routage**

Les routes peuvent être choisies soit de façon dynamique, soit par arrangement préalable de façon à n'utiliser que des sous-réseaux, relais ou liaisons physiquement sûrs. Les systèmes d'extrémité peuvent, lors de la détection d'attaques persistantes par manipulation, souhaiter demander au fournisseur du service de réseau d'établir une connexion via une route différente. La politique de sécurité peut interdire le passage de données portant certaines étiquettes de sécurité à travers certains sous-réseaux, relais ou liaisons. L'initiateur d'une connexion (ou l'expéditeur d'une unité de données en mode sans connexion) peut aussi spécifier des interdictions de routage prescrivant d'éviter des sous-réseaux, liaisons ou relais spécifiques.

➢ **Notarisation**

Des propriétés relatives à des données communiquées entre deux entités ou plus, telles que leur intégrité, leur origine, leur date et leur destination, peuvent être garanties par la fourniture d'un mécanisme de notarisation. La garantie est fournie par un notaire (tierce partie) en qui les entités communicantes ont confiance et qui détient les informations nécessaires pour fournir la garantie requise de manière vérifiable. Chaque instance de communication peut utiliser la signature numérique, le chiffrement et les mécanismes d'intégrité, de façon appropriée, pour le service que doit fournir le notaire. Lorsqu'on fait appel à ce mécanisme de notarisation, les données sont communiquées entre les entités communicantes via les instances de communication protégées et le notaire.

- **Mécanismes génériques**

Le présent paragraphe décrit un certain nombre de mécanismes qui ne sont pas spécifiques à un service particulier. Ainsi ils ne sont pas décrits explicitement comme faisant partie d'une couche(N). Certains de ces mécanismes de sécurité communs peuvent être considérés comme des aspects de gestion de sécurité.

L'importance de ces mécanismes est, en général, directement liée au niveau de sécurité requis. On en distingue cinq types(***Figure1.9***) **[9]**:

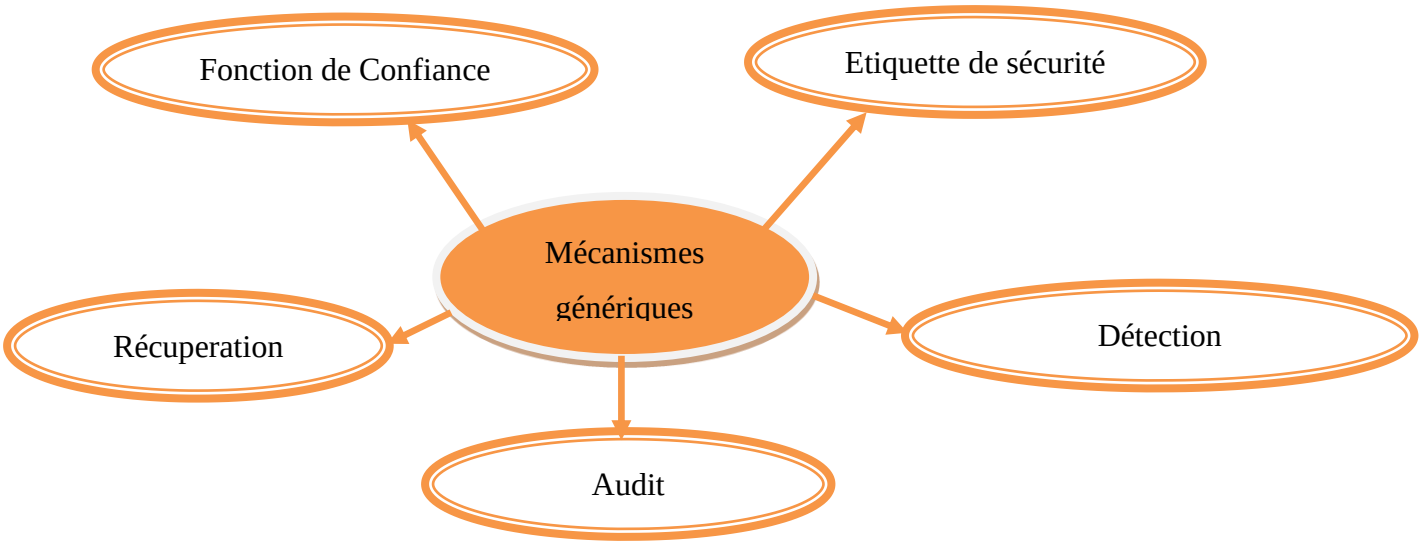

Figure1.9. *Mécanismes génériques*

➢ **Fonction de confiance**

Des fonctionnalités de confiance doivent être utilisées pour étendre le domaine d'application ou pour établir l'efficacité d'autres mécanismes de sécurité. Toute fonctionnalité qui fournit directement des mécanismes de sécurité, ou qui permet l'accès à ces mécanismes, devrait être digne de confiance. Les procédures utilisées pour assurer que l'on peut faire

confiance à un matériel et un logiciel n'entrent pas dans le cadre de la présente Recommandation et, en tout cas, varient selon le niveau de menace perçu et la valeur des informations à protéger.

Ces procédures sont en général coûteuses et difficiles à mettre en œuvre. On peut réduire au minimum les problèmes en choisissant une architecture qui permette la mise en œuvre de fonctions de sécurité en modules qui peuvent être séparés des fonctions non liées à la sécurité ou fournis par elles. Toute protection d'associations au-dessus de la couche sur laquelle porte la protection doit être fournie par d'autres moyens, par exemple, par une fonctionnalité de confiance appropriée.

- **Etiquette de sécurité (security labels)**

Les ressources comprenant des éléments de données peuvent avoir des étiquettes de sécurité qui leur sont associées, par exemple, pour indiquer un niveau de sensibilité. Il est souvent nécessaire d'acheminer l'étiquette de sécurité appropriée avec des données en transit. Une étiquette de sécurité peut être une donnée supplémentaire associée aux données transférées ou peut être implicite ; elle peut, par exemple, être la conséquence de l'utilisation d'une clé spécifique pour chiffrer les données ou résulter du contexte des données tel que la source ou la route. Les étiquettes de sécurité explicites doivent être clairement identifiables afin de pouvoir être vérifiées de façon appropriée. En outre, elles doivent être liées d'une manière sûre aux données auxquelles elles sont associées

- **Détection d ’événement (Event detection)**

La détection d'événements liés à la sécurité comprend la détection de violations apparentes de la sécurité et peut également inclure la détection d'événements « normaux », tels que l'accès réussi (ou demande de connexion). Dans l'OSI, les événements liés à la sécurité peuvent être détectés par des entités comprenant des mécanismes de sécurité. La spécification de ce qui constitue un événement est mise à jour par la gestion du traitement d'événements.

La détection des divers événements liés à la sécurité peut, par exemple, provoquer une ou plusieurs des actions qui peuvent donner lieu aux événements suivants par exemple :

- Une violation spécifique de la sécurité.
- Un événement spécifique choisi.
- Un dépassement du comptage d'un certain nombre d'occurrences.

La normalisation dans ce domaine tiendra compte de la transmission des informations pertinentes pour la notification et l'enregistrement d'événements, et de la définition syntaxique

et sémantique à utiliser pour la transmission de notifications et d'enregistrements d'événements.

- **Audit**

Les journaux d'audit de sécurité fournissent un mécanisme de sécurité appréciable étant donné qu'ils permettent potentiellement de détecter et d'enquêter sur les violations de sécurité en permettant un audit de sécurité ultérieur. Un audit de sécurité est une étude indépendante et un examen des enregistrements et des activités de système permettant de tester l'adéquation des contrôles, de s'assurer de la cohérence avec la politique établie et avec les procédures opérationnelles, d'aider à évaluer les dommages et de recommander des modifications dans les contrôles de la politique et les procédures. Un audit de sécurité nécessite l'enregistrement des informations relatives à la sécurité dans un journal d'audit de sécurité, ainsi que l'analyse et la production de rapports à partir des informations provenant d'un journal d'audit de sécurité. L'enregistrement est considéré comme un mécanisme de sécurité.

La collecte d'informations pour le journal d'audit de sécurité peut être adaptée à divers besoins en spécifiant le(s) type(s) d'événements relatifs à la sécurité à enregistrer. L'existence connue d'un journal d'audit de sécurité peut servir d'élément dissuasif pour certaines sources potentielles d'attaques de sécurité.

Les considérations liées à un journal d'audit de sécurité OSI tiendront compte du type d'information qui pourra, en option, être enregistrée, des conditions sous lesquelles cette information devra être enregistrée et de la définition syntaxique et sémantique à utiliser pour échanger des informations de journal d'audit de sécurité.

- **Récupération**

La reprise de sécurité traite des demandes provenant de mécanismes tels que les fonctions de traitement et de gestion des événements et entreprend des actions de reprise comme résultat de l'application d'un ensemble de règles. Ces actions de reprise peuvent être de trois types, Par exemple :

- Des actions immédiates peuvent créer une coupure immédiate des opérations, comme une déconnexion.

- Des actions temporaires peuvent produire l'invalidation temporaire d'une entité.

- Des actions à long terme peuvent être l'introduction d'une entité sur une « liste noire » ou le changement d'une clé.

Les sujets qui se prêtent à la normalisation comprennent des protocoles pour les actions de reprise et pour la gestion de reprise de sécurité.

8. CONCLUSION :

Bien que le domaine de la sécurité informatique soit très vaste, et qu'il est difficile de le cerner par une définition, la sécurité peut être considérée par le niveau de confiance donné à la confidentialité, l'intégrité et la disponibilité de l'information. La préservation de ces propriétés nécessite la mise en place des services de sécurité qui seront implémentés par des mécanismes de sécurité.

Ces services peuvent être la confidentialité (des données ou du flux de données), l'authentification (d'une entité ou de l'origine des données), le contrôle d'accès, l'intégrité ou encore la non répudiation (avec preuve de l'origine ou preuve de la remise). Les mécanismes peuvent être le chiffrement, l'authentification, l'intégrité, la signature numérique...etc.

Il faut noter que la sécurité ne peut être assurée à cent pour cent, et que les outils ne sont pas parfaits. Ils possèdent toujours des failles. Cependant, avec une bonne politique de sécurité, et un bon déploiement des outils, la sécurité peut être très proche des niveaux acceptés.

La sécurité devant être assurée du front-end au back-end, nous avons jugé utile d'aborder la sécurité de la partie application car c'est le niveau intermédiaire entre les deux parties justement citées, ça fera l'objet du prochain chapitre.

Chapitre 02 :
Sécurité des Applications Web

1. INTRODUCTION :

De nos jours les applications Web sont devenues omniprésentes et nous les utilisons au quotidien. Qui n'utilise pas une application en ligne de type Facebook, Webmail ou encore de partage de photos ou de documents ?

Toute entreprise ou administration se doit maintenant d'avoir une présence sur le Web que ce soit via un blog, un site de vente en ligne, intranet / extranet ou encore une application riche (réseau social, démarches administratives en ligne, …).

Le revers de la médaille est que ces applications étant disponibles, elles sont particulièrement exposées aux attaques par des personnes malveillantes. Ces personnes décident de tenter de contourner les mesures de sécurité mises en place par défis technique et intellectuel, revendication ou tout simplement pour en tirer un profit financier, Personne n'est à l'abri, pas même les plus grandes multinationales comme Sony Playstation ou les institutions nationales.

La sécurité des applications Web est donc devenue un enjeu stratégique aussi important que les fonctionnalités ou l'ergonomie, la peur de se faire voler des données sensibles (numéro de CB, documents, …), Il est donc primordial de nos jours de développer son application Web dans une logique de sécurité, dès le départ et tout au long du processus. **[10].**

Dans ce chapitre nous allons définir d'abord l'application web et son fonctionnement ensuite les différents architectures client-serveur et enfin Les dix attaques de sécurité des applications Web les plus critiques selon l'OWASP (**L'Open Web Application Security**), leurs impacts et leurs contres- mesures.

2. L'APPLICATION WEB

Une application web est une application manipulable directement en ligne grâce à un navigateur web et qui ne nécessite donc pas d'installation sur les machines clientes, contrairement aux applications mobiles. De la même manière que les sites web, une application web est généralement installée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, réseau local, etc.). Exemples :

- Des messageries web, les systèmes de gestion de contenu, les wikis et les blogs sont des applications web.
- Les moteurs de recherches, les logiciels de commerce électronique, les jeux en ligne, les logiciels de forum, les agrégateurs peuvent être sous forme d'application web.
- Des appareils réseau tels que les routeurs sont parfois équipés d'une application web dans leur micro logiciel.

Les applications web font partie de l'évolution des usages et de la technologie du Web appelée Web 2.0 **[11].**

2.1. ARCHITECTURE :

Le web est un ensemble de machines en réseau communiquant à l'aide d'un langage commun. Le web fonctionne en mode client/serveur **(**voir ***Figure 2.1***) c'est-à-dire qu'il y a des machines dites serveurs qui proposent des ressources et des machines appelées clients qui utilisent ces ressources. Les ressources sont par exemple des pages HTML, des images, des fichiers XML (*Extensible Markup Language*) ou encore des programmes (PHP, Java, ASP.NET, Python, Perl, ...) chargés de les générer à la demande. Le client accède aux ressources à l'aide du protocole de communication http.

Au niveau des serveurs, en plus du serveur web, nous pouvons avoir un serveur de données qui va héberger le Système de Gestion de Base de Données (SGBD). Et pour y accéder, on utilise le langage universel d'interrogation des bases de données : SQL. Ci-dessous l'architecture d'une application web **[12]**.

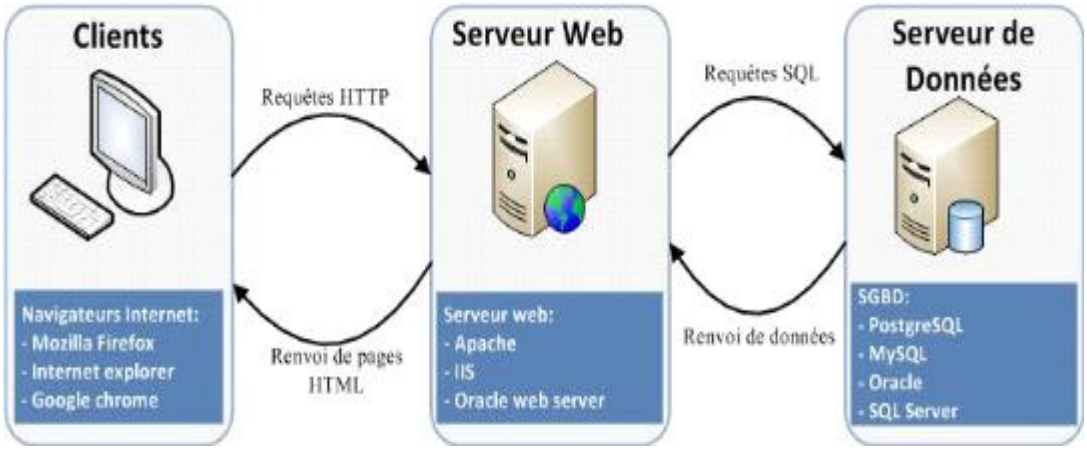

Figure 2.1. *Architecture d'une application web* **[12].**

2.2. PROTOCOLE HTTP :

Le protocole HTTP (HyperText Transfer Protocol) est un protocole de couche d'application, il permet la communication entre le navigateur Web de l'internaute et un serveur dans un format spécifique orienté requête/réponse **[13]**.

2.3. SERVEUR WEB :

Un serveur Web est un logiciel informatique qui permet d'héberger un ou plusieurs sites Internet. Il assure donc la communication avec le navigateur Internet utilisé par un internaute (grâce au protocole réseau HTTP). Un serveur Web est généralement capable de gérer à la fois du contenu statique (un logo, une page HTML simple) ou dynamique (contenu extrait de base de données...). Les serveurs Web les plus connus sont Apache, IIS, Lighthttp.

2.4. SERVEUR D'APPLICATION :

Un serveur d'application est un environnement informatique qui fournit les briques nécessaires l'exécution d'applications transactionnelles sur le web. Il doit répondre à cinq critères techniques :

- S'interfacer avec un serveur http (HTML, XML).
- Fournir un moteur d'exécution des traitements (ex : Java Virtual Machine).
-S'ouvrir sur le système d'information de l'entreprise (XML, web services, connecteurs SGBDR, ERP...)
- Permettre l'ajout de briques techniques et métiers.
- Répondre aux contraintes induites par les architectures centralisées :
  - ✓ Gestion de contextes (différenciation des clients/temps de session par le biais de cookies, d'URL long ou encore de variable cachée).
  - ✓ La répartition de charges (exécution de plusieurs instances réparties sur différentes machines) et le pooling de connexions (évitant de création de goulet d'étranglement).
  - ✓ Les reprises sur incident (l'application est répliquée sur plusieurs serveurs physiques. En cas de "plantage" au niveau applicatif ou serveur, la requête utilisateur est redirigée versun serveur disponible de manière transparente) **[13].**

3. ARCHITECTURE CLIENT/SERVEUR

Ce sont des machines clientes qui contactent un serveur pour leurs fournir des services (des programmes fournissant des données). Les services sont exploités par des programmes clients, s'exécutant sur les machines clientes : client FTP, client de messagerie, …etc [14].

3.1. AVANTAGES DE L'ARCHITECTURE CLIENT/SERVEUR :

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

- **Des ressources centralisées**

  Étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction.

- **Une meilleure sécurité**

  Car le nombre de points d'entrée permettant l'accès aux données est moins important.

- **Une administration au niveau serveur**

  Les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.

- **Un réseau évolutif**

  Grâce à cette architecture on peut supprimer où rajouter des clients sans perturber le fonctionnement du réseau et sans modifications majeures.

3.2. FONCTIONNEMENT D'UN SYSTEME CLIENT/SERVEUR :

Un système client/serveur fonctionne selon le schéma ci-après **(**voir ***Figure 2.2***) comme suit :

- Le client émet une requête vers le serveur grâce à son **adresse IP** et à son **port**, cette requête désigne un service particulier du serveur.
- Le serveur reçoit la demande et répond à l'aide de l'adresse IP de la machine cliente et de son port **[6].**

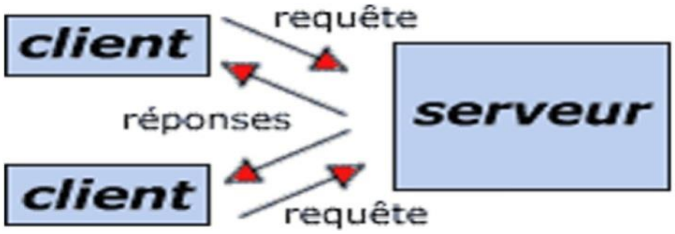

Figure 2.2. *Fonctionnement d'un système client/serveur* **[15].**

3.3. TYPES D'ARCHITECTURE CLIENT-SERVEUR :

3.3.1. Architecture à 2 niveaux :

L'architecture à deux niveaux (aussi appelée ***Architecture 2-tiers***) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service **(**voir ***Figure 2.3***) **[15].**

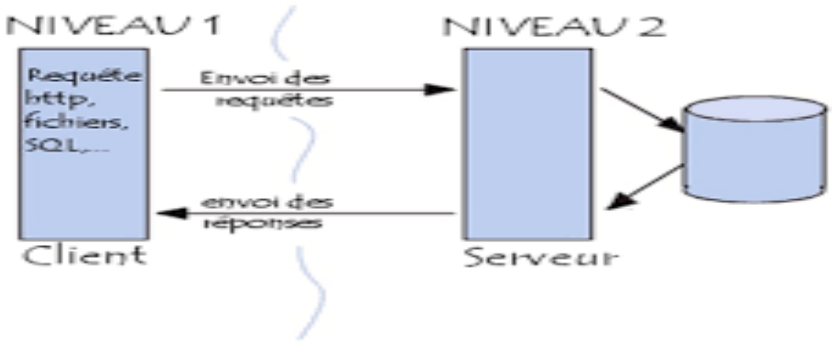

***Figure 2.3. A**rchitecture à 2 niveaux* **[15].**

3.3.2. Architecture à 3 niveaux :

Dans l'architecture à 3 niveaux, appelée aussi *architecture 3-tiers* **(voir *Figure 2.4***), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

1. Un client, c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation ;
2. Le serveur d'application (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur
3. Le serveur de données, fournissant au serveur d'application les données dont il a besoin.

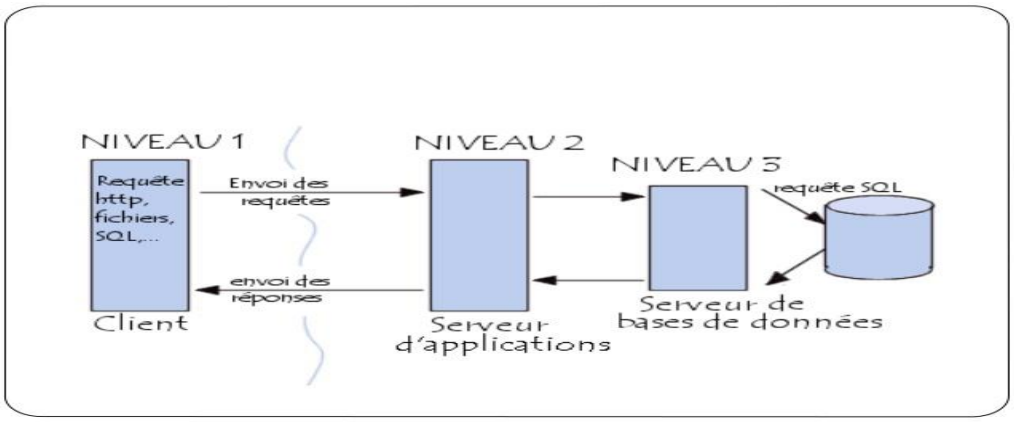

***Figure 2.4.** Architecture à 3 niveaux* **[15].**

Etant donné l'emploi massif du terme d'architecture à 3 niveaux, celui-ci peut parfois désigner aussi les architectures suivantes :

- Partage d'application entre client, serveur intermédiaire, et serveur d'entreprise ;

- Partage d'application entre client, serveur d'application, et serveur de base de données d'entreprise **[15]**.

3.3.3. Architecture N niveaux :

Dans l'architecture à 3 niveaux, chaque serveur (niveaux 2 et 3) effectue une tâche (un service) spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à N niveaux **(voir *Figure 2.5*) [15]**.

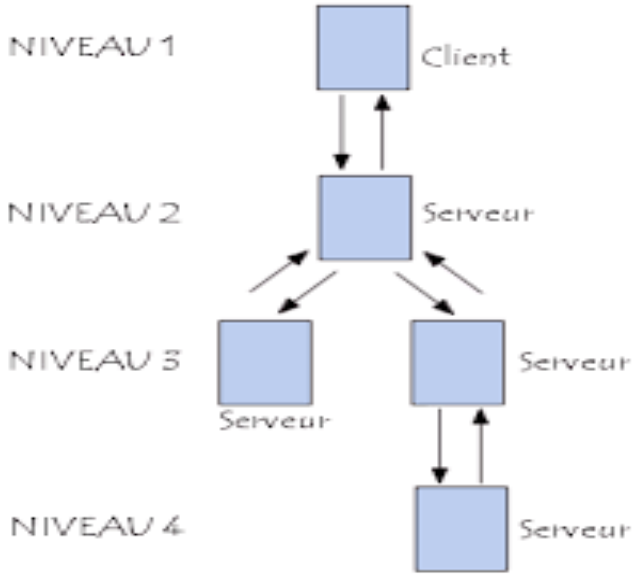

Figure 2.5. *Architecture à N niveaux* **[15].**

4. LE CHOIX DE L'ARCHITECTURE :

- Comparaison des deux types D'ARCHITECTURE :

L'architecture à deux niveaux est donc une architecture client/serveur dans laquelle le serveur est polyvalent, c'est-à-dire qu'il est capable de fournir directement l'ensemble des ressources demandées par le client. Dans l'architecture à trois niveaux par contre, les applications au niveau serveur sont délocalisées, c'est-à-dire que chaque serveur est spécialisé dans une tâche (serveur web/serveur de base de données par exemple). L'architecture à trois niveaux permet :

- Une plus grande flexibilité/souplesse.
- Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau.

- De meilleures performances, étant donné le partage des tâches entre les différents serveurs **[15].**

Vu les avantages de l'architecture de 3-tiers surtout cote sécurité, nous l'avons choisi pour implémenter notre cas d'utilisation.

5. SECURITE DES APPLICATIONS WEB

L'objectif de ce chapitre est de parler des attaques applicatives les plus potentielles qui peuvent compromettre les projets Web d'une part, et de comment s'en prémunir en adoptant la pratique du codage sécurisé (secure coding), et en modifiant certains paramètres des fichiers de configuration des serveurs Web et du moteur PHP d'une autre part. Il convient de mentionner qu'il existe une référence internationale des attaques des application Web connue sous le nom OWASP (**L'Open Web Application Security**) [16].

5.1. QU'EST-CE QUE L'OWASP ?

OWASP acronyme d'Open Web Application Security Project, est une fondation publique à but non-lucratif qui offre à des organismes la possibilité de développer, acheter et maintenir des applications sûres. L'OWASP laisse en accès libre et gratuit une panoplie de services dont les plus importants :

- Des normes et des outils de sécurité des applications.
- Des livres complets sur les tests de sécurité des applications, le développement de code sécurisé et l'audit de code.
- Des normes de contrôles de sécurité et des librairies.
- Des Chapitres locaux dans le monde entier.
- De la recherche de pointe.
- Des conférences à travers le monde.
- Des listes de diffusion.

La communauté en question dédie le lien www.OWASP.org/ aux organisations, administrateurs de systèmes d'information et aux étudiants pour profiter des services qu'elle propose, et le site www.owasp.org/index.php/Top_10 pour connaitre le Top dix OWASP. Il s'agit d'un document qui présente le top 10 des vulnérabilités de sécurité applicative les plus importantes. Le rapport OWASP (publié chaque 3ans) fournit pour chaque faille une description, des exemples, des recommandations pour s'en prémunir et des références pour s'enrichir. Il indique également quelles attaques sont possibles à partir de ses vulnérabilités. Le classement a été réalisé à partir du CVE (Common Weekness Enumeration) du Mitre **[13].**

5.2. TOP DIX DES VULNERABILITES SELON OWASP 2017 :

Les vulnérabilités jugées les plus critiques selon l'OWASP sont **[17]** :

1- Injection.

2 -Authentification brisée.

3-Exposition de données sensibles.

4- Entité externe XML (XXE).

5- Contrôle d'accès brisé.

6-Mauvaise configuration de la sécurité.

7- Cross-Site Scripting (XSS).

8-Désérialisation incertaine.

9- Utilisation de composants avec des vulnérabilités connues.

10- Insuffisance de l'enregistrement et de la surveillance.

5.2.1. Injection

Les failles d'injection, telles que l'injection SQL ou LDAP, se produisent lorsqu'un attaquant envoie des données non fiables à un interpréteur qui est exécuté comme une commande ou une requête sans autorisation appropriée (voir ***Figure 2.6)***. Elle peut entraîner la perte de données, la corruption, la divulgation à des parties non autorisées et même une prise de contrôle totale **[18]**.

Les défauts d'injection résultent d'une défaillance classique du filtrage des entrées non fiables. Cela peut arriver lorsque on transmet des données non filtrées au serveur SQL (injection SQL), au navigateur (XSS) – nous en reparlerons au serveur LDAP (injection LDAP) ou ailleurs. Le problème ici est que l'attaquant peut injecter des commandes à ces entités, ce qui entraîne une perte de données et le piratage des navigateurs des clients **[20]**.

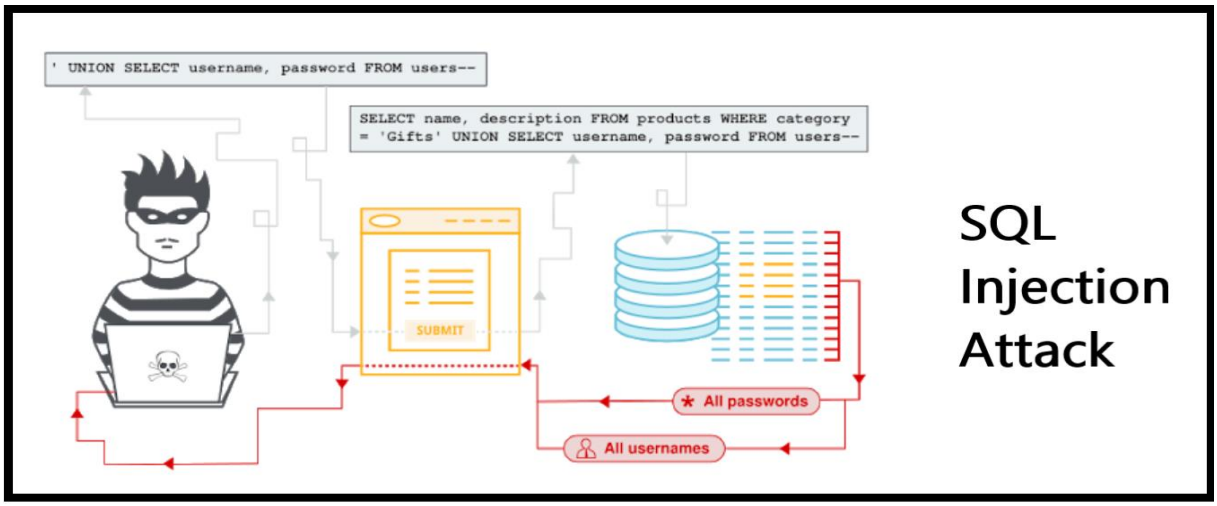

Figure 2.6. *Attaque injection SQL* **[20].**

➢ **Impacts :**

L'injection peut entraîner une perte de données, corruption ou divulgation aux parties non autorisées, perte de responsabilité ou refus d'accès. L'injection peut parfois conduire à une prise de contrôle complète de l'hôte. L'impact commercial dépend des besoins de l'application et les données [**17]**.

➢ **Exemple** :

Une application utilise des données non fiables dans la construction, en profitant de cela, l'attaquant modifie la valeur du paramètre dans le navigateur à envoyer. Cela modifie la signification des deux requêtes pour renvoyer tous les enregistrements de la table de comptes **[19]**.

➢ **Contre-mesures :**

- Garder les données séparées des commandes et des requêtes **[18]**.
- Il est préférable d'utiliser une API plus sûre qui évite totalement l'utilisation de l'interpréteur.
- Utilisation d'une validation d'entrée positive côté serveur.
- Utilisation de la syntaxe d'échappement spécifique pour l'interpréteur.
- Utiliser LIMIT et d'autres contrôles SQL dans les requêtes pour empêcher la divulgation massive d'enregistrements en cas d'injection SQL **[19]**.
- Effectuer des tests de sécurité des applications **[18]**

5.2.1. Authentification et gestion de session brisées

Une authentification des utilisateurs et la gestion des sessions mal configurée pourrait permettre aux attaquants de compromettre les mots de passe, les clés ou les jetons de session, ou de prendre le contrôle des comptes des utilisateurs afin d'assumer leur identité de manière temporaire ou permanente. Cela peut entraîner l'usurpation d'identité, le blanchiment d'argent et la divulgation d'informations hautement sensibles **[18]**.

OWASP définit Authentification et gestion de session brisées en tant que :

Les fonctions d'application liées à l'authentification et à la gestion de session ne sont souvent pas implémentées correctement, ce qui permet aux attaquants de compromettre les mots de passe, les clés ou les jetons de session, ou d'exploiter d'autres vulnérabilités d'implémentation pour prendre pour acquis l'identité d'autres utilisateurs (voir ***Figure 2.7)***.

En d'autres termes, un attaquant peut obtenir un accès non autorisé aux données d'un utilisateur en raison de failles dans la mise en œuvre. Avant d'exploiter cette vulnérabilité, on doit connaître quelques concepts :

- Pourquoi avons-nous besoin d'une session et qu'est-ce qu'une session ?

- Qu'est-ce qu'un cookie ?
- Qu'est-ce que l'authentification **? [20]**.

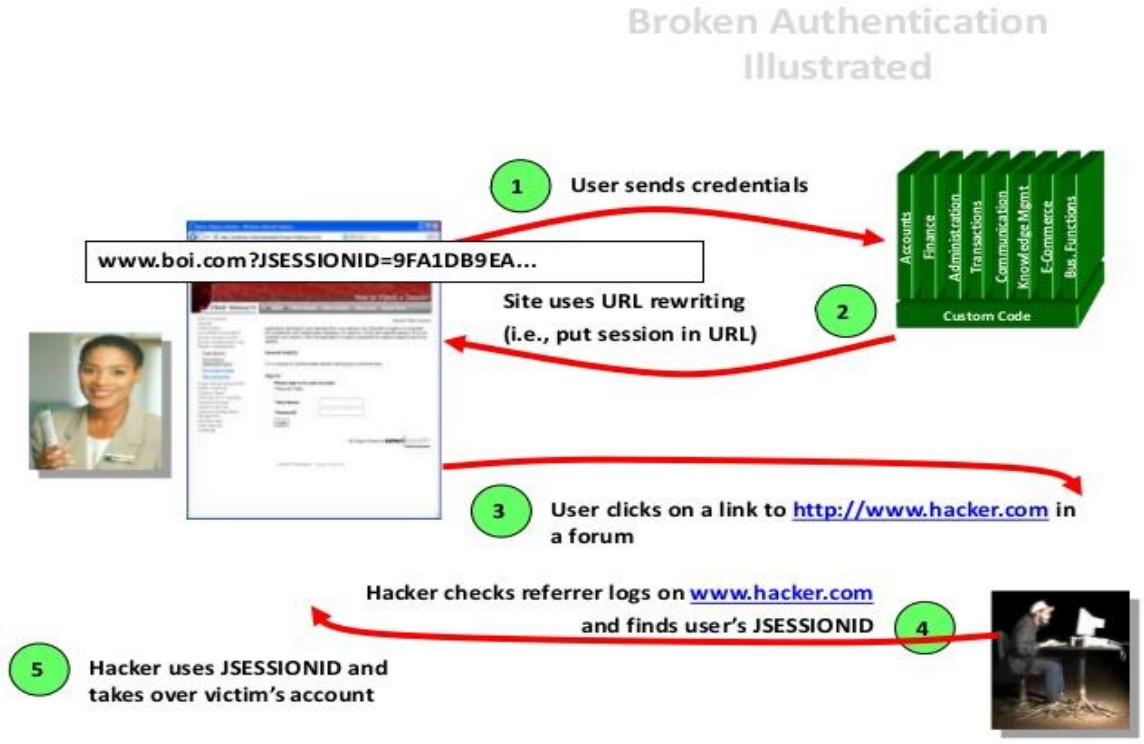

Figure 2.7. *Authentification* et gestion de session brisées **[20].**

➢ **Impacts :**

Les attaquants doivent accéder uniquement à quelques comptes ou un seul administrateur compte pour compromettre le système. En fonction du domaine de la demande, cela peut permettre de blanchiment de l'argent, fraude à la sécurité sociale et vol d'identité, ou divulguer légalement la protection des informations hautement sensibles **[17]**.

➢ **Exemple :**

Si l'utilisateur utilise un ordinateur public pour accéder à un site et qu'il oublie de se déconnecter, il le ferme directement et s'éloigne. Ensuite, l'attaquant utilise le même navigateur et le navigateur est toujours authentifié **[19]**.

➢ **Contre-mesures :**

- Les informations d'identification de l'utilisateur doivent être protégées.
- Ne pas exposer pas l'ID de session dans l'URL.
- Il devrait y avoir un délai d'expiration dans l'ID de session.
- Les mots de passe ne doivent pas être envoyés via des connexions non chiffrées **[19]**.
- Mettre en place une authentification à plusieurs facteurs si possibles.

- Mettre en place des contrôles de mots de passe forts.
- Ne pas expédier ou déployer avec des références par défaut **[18]**.

5.2.3. Exposition aux données sensibles

De nombreuses applications et API ne protègent pas correctement les données sensibles, telles que les données financières, les dossiers médicaux ou d'autres informations personnelles. Cela pourrait permettre aux attaquants d'accéder à ces informations pour commettre des fraudes ou voler des identités **[18]**.

Cette vulnérabilité permet à un attaquant d'accéder à des données sensibles telles que des cartes de crédit, des identifiants fiscaux, des identifiants d'authentification, etc. pour commettre une fraude sur une carte de crédit, un vol d'identité ou tout autre crime (voir ***Figure 2.8)*.** La perte de telles données peut avoir un impact important sur l'entreprise et nuire à sa réputation. Les données sensibles méritent une protection supplémentaire, telles que le cryptage à l'arrêt ou en transit, ainsi que des précautions spéciales en cas d'échange avec le navigateur **[20]**.

➢ **Impacts :**

L'échec compromet souvent tout données qui auraient dû être protégées. En règle générale, ces informations comprennent informations personnelles sensibles des données telles que les dossiers de santé, identifiants, données personnelles et cartes de crédit, qui nécessitent souvent une protection définis par des lois ou règlements tels que le RGPD de l'UE ou les lois locaux sur la confidentialité **[17]**.

➢ **Exemple :**

Une application crypte les numéros de carte dans une base de données par cryptage automatique de la base de données et nous décrypterons ces données automatiquement une fois récupérées. Permettre à une faille d'injection SQL de récupérer des données **[19]**.

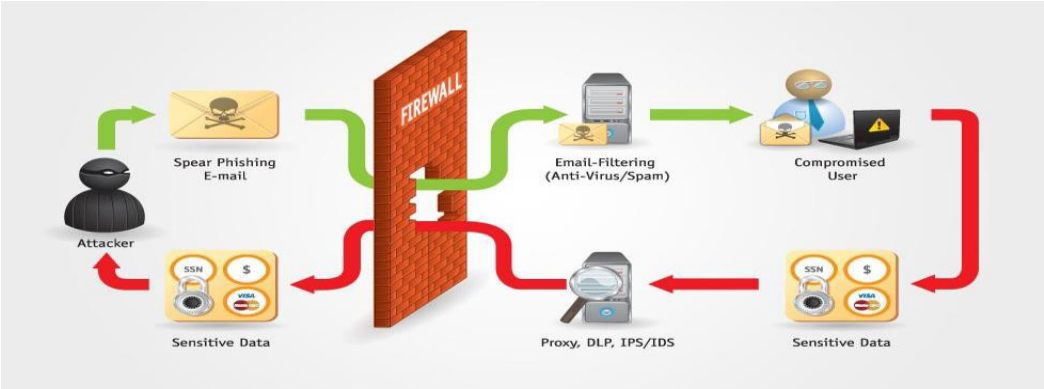

Figure 2.8. *Exposition aux données sensibles* ***[20]***

- **Contre-mesures**

- Classifier les données en fonction des réglementations légales.
- Ne pas stocker de données sensibles inutilement.
- Veiller à crypter toutes les données au repos et en transit **[18]**.
- Utiliser des mots de passe uniques et forts.
- Garder une trace des comptes bancaires et des transactions par carte.
- Utiliser des URL sécurisées
- Agir dès que possible si un suspect est détecté **[19]**.

5.2.4. Entité externe XML (XXE)

Les processeurs XML anciens ou mal configurés évaluent les références d'entités externes dans les documents XML. Ils peuvent être utilisés pour des attaques, y compris l'exécution de code à distance, pour scanner le système interne et pour divulguer des fichiers internes **[11]**.Entité externe XML (XXE) fait référence à un type spécifique de Contrefaçon de demande côté serveur (SSRF) attaque, par laquelle un attaquant peut provoquer un déni de service et accéder à des fichiers et services locaux ou distants en abusant d'une fonctionnalité largement disponible et rarement utilisée dans les analyseurs syntaxiques XML (voir ***Figure 2.9)***.XML est un format de données très utilisé que l'on trouve dans tout, des services Web (XML-RPC, SOAP, REST, etc.) aux documents (XML, HTML, DOCX) et aux fichiers d'image (SVG, données EXIF, etc.) qui utilisent XML. Naturellement, là où il y a XML, il y a un analyseur XML **[20]**.

- **Impacts**

Ces failles peuvent être utilisées pour extraire data, exécuter une requête à distance depuis le serveur, scanner les systèmes internes, effectuer une attaque par déni de service, comme ainsi d'exécuter d'autres attaques. L'impact commercial dépend des besoins de la protection de toutes les applications et données affectés **[17]**.

- **Exemple :**

L'attaquant tente d'extraire des données du serveur ou tente une attaque DoS en incluant des fichiers potentiellement infinis **[19]**.

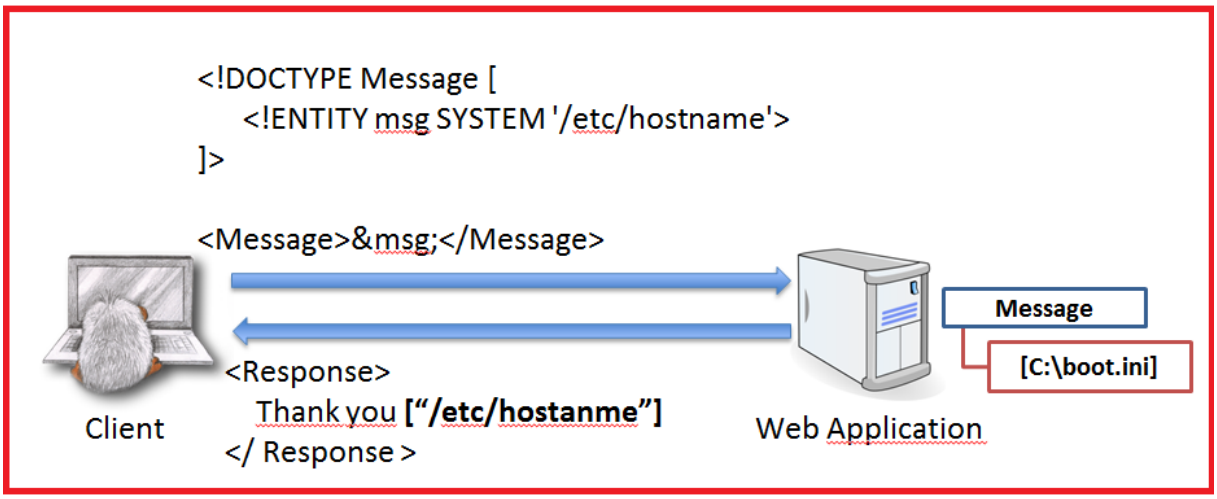

Figure 2.9. *Entité externe XML (XXE)* **[20]**

➢ **Contre-mesures**

- Patch ou mise à jour de tous les processeurs et bibliothèques XML.
- Utiliser les outils de test de sécurité des applications statiques (SAST) pour inspecter les dépendances **[18]**. Les outils SAST aident à détecter XXE dans le code source **[19]**.
- Former les développeurs à identifier et à atténuer le XXE **[18]**.
- Essayer d'utiliser des formats moins complexes tels que JSON.
- Mettre à niveau tous les processeurs et bibliothèques XML **[19]**.

5.2.5. Contrôle d'accès frauduleux

Des restrictions mal configurées ou manquantes sur les utilisateurs authentifiés leur permettent d'accéder à des fonctionnalités ou à des données non autorisées. Les attaquants peuvent accéder aux comptes d'autres utilisateurs, consulter des documents sensibles et modifier les données et les droits d'accès **[18]**.

Un contrôle d'accès interrompu est un échec du contrôle d'accès de tout utilisateur. Le contrôle d'accès est la politique selon laquelle les utilisateurs ne peuvent pas utiliser un accès autre que celui pour lequel ils ont reçu l'autorisation. Le non-respect de cette procédure peut entraîner la perte d'informations (voir Figure ***2.10)***.

Cela est dû aux faiblesses fournies par les développeurs d'applications. Par des tests manuels, nous pouvons détecter un contrôle d'accès inefficace **[19]**.

➢ **Impacts :**

L'impact technique est celui des attaquants agissant en tant qu'utilisateurs ou administrateurs, ou utilisateurs utilisant des fonctions privilégiées, ou créer, accéder, mettre à jour ou

supprimer chaque enregistrement. L'impact commercial dépend des besoins de protection de l'application et les données **[17]**.

➢ **Exemple** :

Des droits d'administrateur sont requis pour accéder à la page d'administration. Si un utilisateur non autorisé peut accéder à cette page, c'est une faille **[12]**.

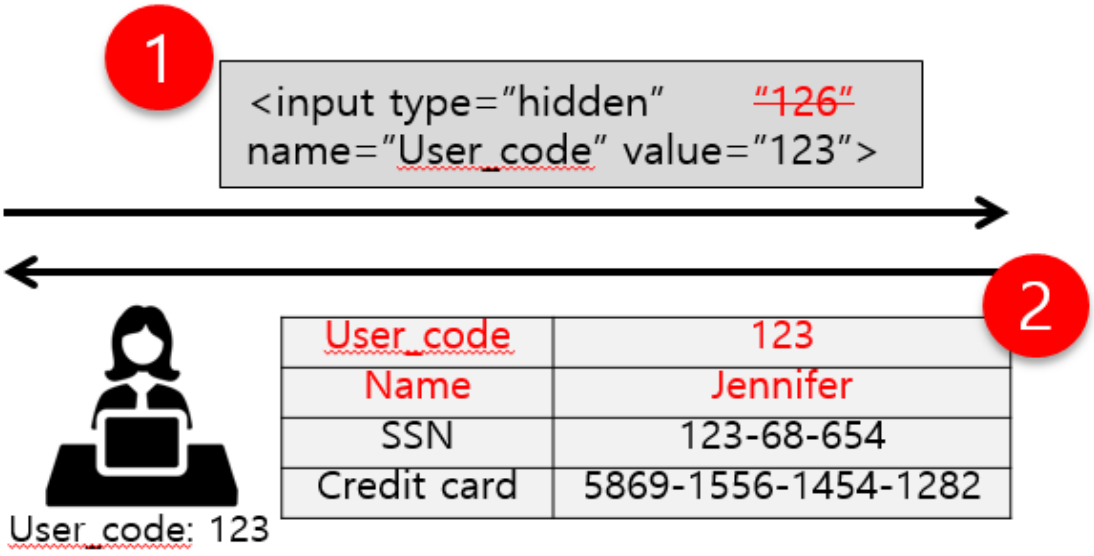

Figure 2.10. *Contrôle* d'accès frauduleux **[20].**

➢ **Contre-mesures :**

- Mettre en place des mécanismes de contrôle d'accès.
- Effectuer des tests de pénétration pour détecter les contrôles d'accès qui ne fonctionnent pas correctement **[18]**.
- Refuser en utilisant la valeur par défaut.
- Appliquer les droits de propriété au contrôle d'accès.
- Journaliser les échecs de contrôle d'accès **[19]**.

5.2.6. Mauvaise configuration de la sécurité

C'est le résultat courant de configurations par défaut peu sûres, de configurations incomplètes ou ad hoc, d'un stockage en nuage ouvert, d'en-têtes de sécurité mal configurés et de messages d'erreur verbeux contenant des informations sensibles **[18]**.

Les pirates informatiques gardent toujours une trace de la configuration de la sécurité. Ainsi, ils essaient toujours de nouvelles façons d'accéder aux sites Web. Une mauvaise configuration dans le système peut conduire à un moyen facile d'accéder à leurs sites Web. Puisque les

développeurs travaillent sur la fonctionnalité des sites Web, pas sur la sécurité. Les configurations sont effectuées sur le serveur d'applications, le serveur proxy de base de données et d'autres périphériques qui doivent être en ligne pour répondre aux exigences de sécurité **[19]**.

Une mauvaise configuration de sécurité se produit lorsque les paramètres de sécurité sont définis, implémentés et conservés par défaut (voir *Figure 2.11)*. Une bonne sécurité nécessite une configuration sécurisée définie et déployée pour l'application, le serveur Web, le serveur de base de données et la plate-forme. Il est également important que le logiciel soit à jour. Les architectures de sécurité des applications actuelles ne suivent pas la sécurité par défaut. Au contraire, les programmeurs doivent appliquer des mesures de sécurité pour éviter l'accès à des ressources privées ou confidentielles **[20].**

➢ **Impacts**

Comme les failles donnent souvent aux attaquants accès non autorisé aux certaines données de systèmes ou fonctionnalités. Parfois, ces défauts entraînent un compromis du système. L'impact commercial dépend des besoins de protection de l'application et les données **[17]**.

La mauvaise configuration du serveur ou de l'application Web entraînant diverses failles :

- Débogage activé.
- Autorisations de dossier incorrectes.
- Utilisation de comptes ou de mots de passe par défaut.
- Pages de configuration / configuration activées.

Toutes les données pourraient être volées ou modifiées lentement au fil du temps **[20]**.

➢ **Exemple :**

Nous utilisons la plupart du temps des exemples de produits qui présentent la plupart des failles de sécurité et les attaques peuvent également les utiliser pour pénétrer dans le serveur **[19]**.

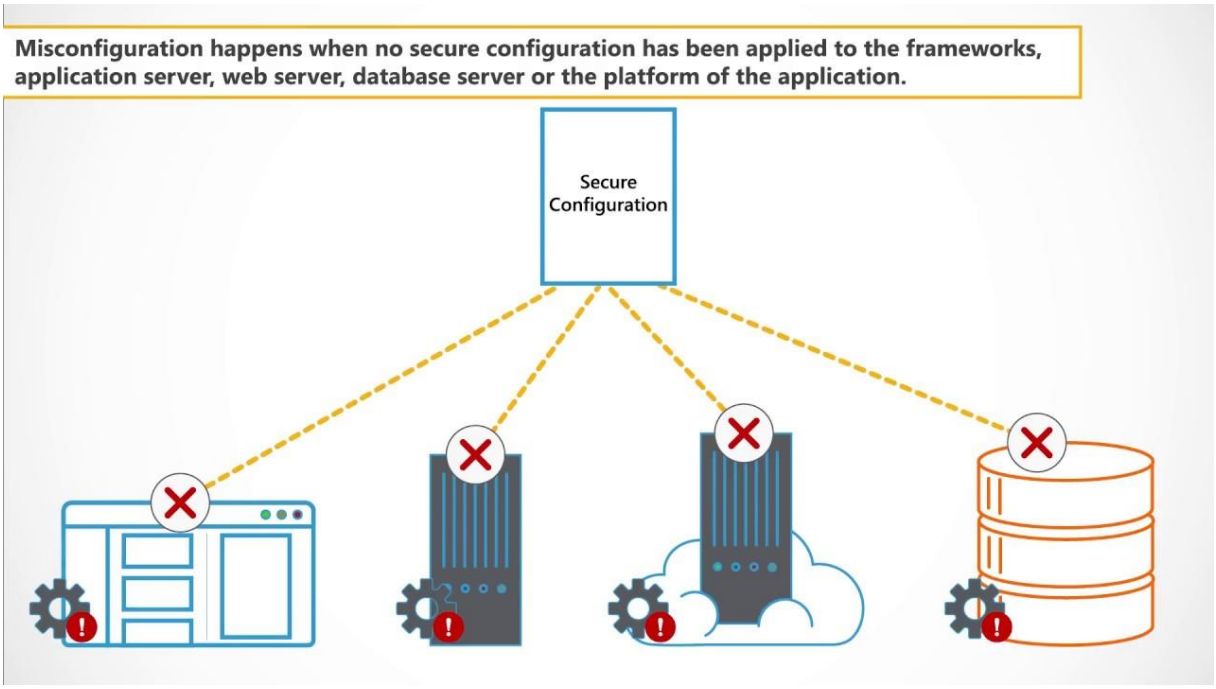

Figure 2.11. *Mauvaise configuration de la sécurité* ***[20].***

- **Contre-mesures**

- Corriger et mettre à jour régulièrement les systèmes, les cadres et les composants.
- Effectuer des tests dynamiques de sécurité des applications (**DAST**) **[18]**.
- Prévention des erreurs de configuration de sécurité.
- Désactiver l'utilisation des mots de passe par défaut.
- Configurer le serveur soi-même.
- Désactiver les interfaces d'administration.
- Désactiver le débogage **[19]**.

5.2.7. Cross Site Scipting (XSS)

Des failles dans les attaques XSS se produisent chaque fois qu'une application inclut des données non fiables dans une nouvelle page web sans validation ou échappatoire appropriée. XSS permet aux attaquants d'exécuter des scripts dans le navigateur de la victime, qui peuvent détourner des sessions d'utilisateurs, rediriger les utilisateurs vers des sites web malveillants ou dégrader des sites web **[18]**.

Le script intersites (XSS) est un code côté client (attaque par injection) . Un attaquant vise à exécuter des scripts malveillants dans un navigateur Web de la victime en incluant un code malveillant dans une page Web ou une application Web légitimes. (Voir ***Figure 2.12*****).** L'attaque réelle se produit lorsque la victime visite la page Web ou l'application Web

qui exécute le code malveillant. La page Web ou l'application Web devient un moyen de transmettre le script malveillant au navigateur de l'utilisateur. Les véhicules vulnérables couramment utilisés pour les attaques de script intersites sont les forums, les forums de discussion et les pages Web autorisant les commentaires.

Une page Web ou une application Web est vulnérable au XSS si elle utilise une entrée utilisateur non authentifiée dans la sortie générée. Cette entrée utilisateur doit ensuite être analysée par le navigateur de la victime. Les attaques XSS sont possibles dans VBScript, ActiveX, Flash et même CSS. Cependant, ils sont plus courants en JavaScript, principalement parce que JavaScript est fondamental pour la plupart des expériences de navigation **[20]**.

➢ **Impacts :**

L'impact d'XSS est modéré pour XSS reflétée et XSS DOM, et sévère pour XSS stockée, avec code à distance exécution sur le navigateur de la victime, comme le vol d'informations d'identification, sessions, ou la livraison de logiciels malveillants au victime **[17].**

Remarque : XSS reflétée, XSS DOM et XSS stockée sont des types d'attaque XSS que nous allons aborder en détails dans le chapitre 4.

➢ **Exemple :**

L'attaquant injecte une charge utile dans le site Web en soumettant un formulaire vulnérable avec un contenu JavaScript malveillant **[19]**.

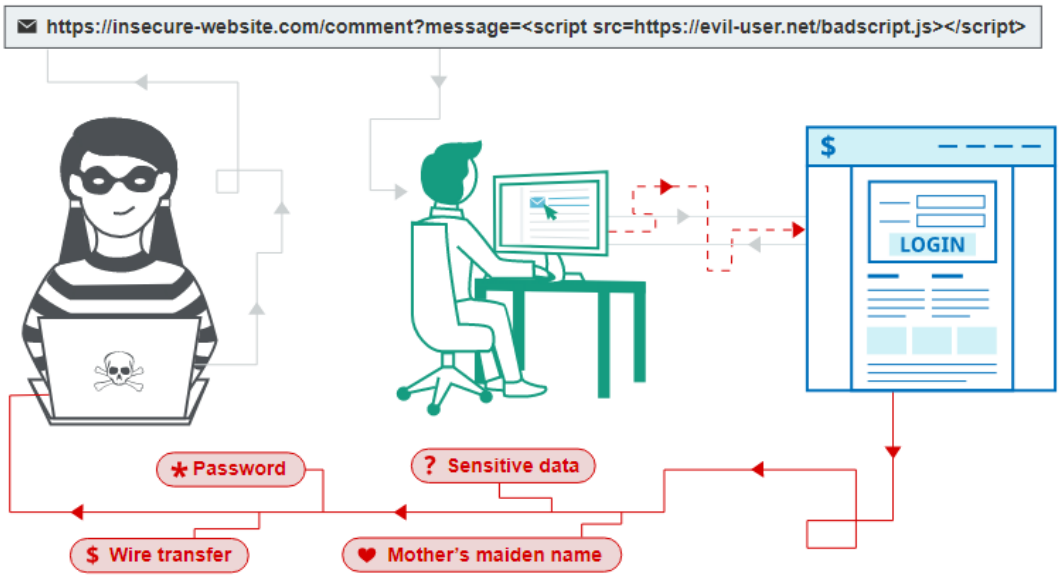

Figure 2.12. *Scripting de site croisé (XSS)* ***[20].***

- **Contre-mesures :**

- Séparer les données non fiables du contenu actif du navigateur.
- Utiliser des Framework échappant automatiquement à XSS par conception.
- Appliquer les meilleures pratiques de codage, comme la validation des entrées et le codage des données **[18], On** doit nettoyer l'entrée et encoder la sortie, Ne faire confiance à aucune entrée utilisateur. **[19]**.
- Utiliser des en-têtes de réponse appropriés.
- Utiliser la politique de sécurité du contenu **(CSP) [19]**.

5.2.8. Dé sérialisation incertaine

Une dé sérialisation non sécurisée peut conduire à l'exécution de codes à distance. Il peut également être utilisé pour effectuer des attaques par injection, par rediffusion et par escalade de privilèges **[18]**.

La plupart des risques critiques pour la sécurité des applications Web sont **Dé sérialisation incertaine (non sécurisée)**. Cette vulnérabilité se produit lorsque des données non fiables sont utilisées pour abuser de la logique d'une application ou d'une interface de programme d'application (API).

Par exemple, un attaquant peut rechercher un objet ou une structure de données dans l'intention de le manipuler à des fins malveillantes. OWASP a répertorié les types d'attaques principaux en tant qu'attaques par déni de service (DoS), contournements de l'authentification et attaques d'exécution de code / commande à distance, dans le cadre desquels les attaquants manipulent du code arbitraire lors de sa dé sérialisation **[20]**.

- **Impacts**

Il s'agit donc d'un grave problème de sécurité des applications qui affecte la plupart des systèmes modernes. L'impact des failles de dé sérialisation ne peut pas être sous-estimé **[19]**. Ces défauts peuvent conduire à l'exécution de code à distance attaques, l'une des plus graves attaques possibles. L'impact commercial dépend des besoins de protection de l'application et les données **[10]**.

- **Exemple :**

Un forum PHP utilise la sérialisation d'objets PHP pour enregistrer un cookie contenant l'ID utilisateur, le hachage du mot de passe **[19]**.

- **Contre-mesures**

- N’accepter jamais d’objets sérialisés provenant de sources non fiables, et n’utiliser pas de supports de sérialisation qui n’autorisent que des types de données primitives.
- Effectuer des tests de pénétration **[18]**.
- Mettre en œuvre des contrôles d'intégrité tels que les signatures numériques.
- Appliquer des contraintes strictes.
- Restreindre et surveiller la connectivité réseau entrante et sortante.
- Isoler et exécuter du code qui dé sérialise dans les environnements à faibles privilèges **[19]**.

5.2.9. Utilisation de composants présentant des vulnérabilités connues

Pour de nombreuses entreprises, ce risque devrait peut-être figurer en tête de liste. Il est fréquent que les développeurs ne sachent pas quels sont les composants (tels que les bibliothèques, les cadres, les modules) de leurs applications. Travailler avec des applications et des API qui utilisent des composants présentant des vulnérabilités connues peut conduire à l’exploitation de composants non sécurisés et à la prise de contrôle de l’infrastructure ou au vol de données sensibles **[18]**.

Nous pouvons tous convenir que le fait de ne pas mettre à jour tous les logiciels du backend et du front-end d’un site Web entraînera sans aucun doute de lourdes menaces pour la sécurité, le plus tôt possible.

Cela peut paraître un peu trop dramatique, mais chaque fois qu'on ignore un avertissement de mise à jour, on permet peut-être à une vulnérabilité désormais connue de survivre dans le système. Les cybercriminels sont prompts à enquêter sur les logiciels et à mettre à jour les changelogs **(**voir **Figure 2.13)**. Quelle que soit la raison pour laquelle on exécute un logiciel obsolète sur l’application Web, on ne peut pas le laisser sans protection. Les deux Sucuri et OWASP recommandent les patches virtuels dans les cas où l’application de correctifs n’est pas possible.

Les correctifs virtuels permettent de protéger les sites Web obsolètes (ou ceux qui présentent des vulnérabilités connues) des attaques en empêchant l’exploitation à la volée de ces vulnérabilités. Cela se fait généralement par un pare-feu et un système de détection d’intrusion **[20]**.

- **Impacts :**

Certaines vulnérabilités connues n'entraînent que des impacts mineurs, certaines les autres sont les plus grandes violations à ce jour ont s'appuyée sur l'exploitation connue des

vulnérabilités des composants. Selon les actifs qu'on a à protéger, peut-être que ce risque devrait être en haut de la liste **[17]**.

➢ **Exemple :**

Les composants s'exécutent généralement avec les mêmes privilèges que l'application elle-même, de sorte que les failles de tout composant peuvent avoir un impact sérieux. Ces défauts peuvent être accidentels ou intentionnels **[19]**.

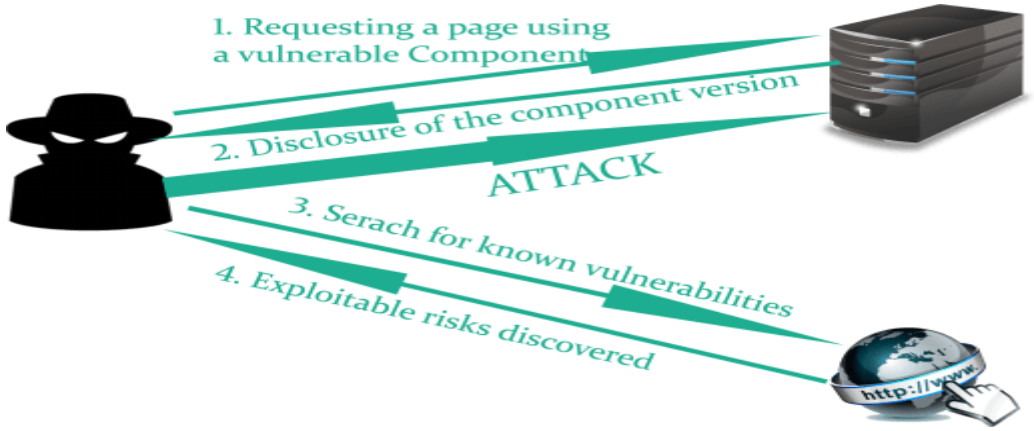

Figure 2.13. *Utilisation de composants présentant des vulnérabilités connues* ***[20]***

➢ **Contre-mesures**

- N'obtenir que des ressources de sources officielles via des liens sécurisés.
- Surveiller les bibliothèques et les composants non entretenus. Analyser les composants du logiciel **[18]**.
- Obtenir des composants uniquement auprès de sources officielles.
- Supprimer les dépendances et la documentation inutilisées **[19]**.

5.2.10. Insuffisance de l'enregistrement et de la surveillance

Une journalisation et une surveillance insuffisantes ainsi qu'une intégration inefficace avec les systèmes de réponse de sécurité permet aux attaquants d'attaquer davantage les systèmes, de maintenir la persistance, d'extraire ou de détruire des données et de pivoter vers d'autres systèmes **[18]**.

La plupart du temps, on a remarqué qu'on est redirigé vers des sites Web différents et malveillants sans aucune validation appropriée. Ils vérifient les échecs de l'utilisateur et vérifient le pare-feu sur toutes les tentatives de connexion. C'est donc le fondement de chaque incident majeur. Les pirates informatiques comptent sur le manque de surveillance et de réponse rapide pour atteindre leurs objectifs sans être détectés **[19]**.

➢ **Impacts :**

Les attaques les plus réussies commencent par détection de vulnérabilité. Permettre une telle détection à continuer peut augmenter la probabilité d'un exploit réussi pour près de 100%. En 2016, l'identification d'une violation a nécessité une moyenne de 191 jours, c'est beaucoup de temps pour les dommages à infliger.

➢ **Exemple :**

Un attaquant utilise des scans pour les utilisateurs utilisant un mot de passe commun. Ainsi, ils peuvent prendre en charge tous les comptes en utilisant ce mot de passe. Pour tous les autres utilisateurs, cette analyse ne laisse qu'une seule fausse connexion. Après quelques jours, cela peut être répété avec un mot de passe différent **[19]**.

➢ **Contre-mesures**

- Veiller à ce que toutes les défaillances (connexion, contrôle d'accès, validation des entrées côté serveur) qui puissent être enregistrées pour identifier les comptes suspects et les contrôler régulièrement.
- Effectuer des tests de pénétration **[18]**.
- S'assurer que les journaux sont générés dans un format facile à comprendre.
- Établir un plan de réponse aux incidents et de récupération.
- Mettre en place une surveillance et une alerte efficaces **[19]**.

5. CONCLUSION :

La dangerosité des attaques applications web se présente au niveau des données et services puisque nous avons remarqué que la plupart des attaquants ciblent les bases de données directement ou indirectement en utilisant le web comme un moyen d'accès, c'est pour cela que nous avons jugé qu'il est nécessaire d'aborder les types d'attaques les plus courants qui ciblent les bases de données qui fera l'objet du prochain chapitre et l'objet d'étude de notre travail qui consiste à sécuriser du serveur de base de donnée.

Chapitre 03 :
Sécurité des bases de données

1. INTRODUCTION :

Les données sont l'actif le plus précieux dans le monde d'aujourd'hui, car elles sont utilisées quotidiennement et par tous .Elles jouent un rôle crucial dans le succès ou l'échec d'une organisation car la plupart d'entre elles utilisent la base de données pour le stockage de leurs données majeures ou importantes, les données stockées ne sont pas que des détails sur l'utilisateur, mais elles peuvent contenir également des informations d'identification ou les informations sensibles d'une organisation.

Le principal objectif de la sécurité des bases de données est de protéger et maintenir la sécurité des informations et des données. Les concepts les plus élémentaires de la sécurité des bases de données sont authentification, confidentialité et intégrité. Ces techniques sont utilisées pour protéger la base de données.

Dans ce chapitre nous allons voir les principales menaces de bases de données, les attaques auxquelles elles sont exposées et comment y faire face.

2. DEFINITION :

Une base de données est un ensemble d'informations ou de données organisé de manière à être facilement accessible, géré ou actualisé. Dans le contexte de l'informatique, les bases de données sont parfois classées selon leur approche organisationnelle et l'une des approches les plus courantes est la base de données relationnelle qui est une représentation tabulaire des données **[21].** Les données sont définies de manière à pouvoir être réorganisées et accessible de différentes manières. En base de données distribuée, la donnée peut être diffusée ou répliquée entre différents points dans un réseau. Les bases de données permettent à tout utilisateur autorisé d'accéder, d'entrer ou analyser les données rapidement et facilement. C'est une collection de requêtes, vues et tables.

Le système de gestion de base de données (SGBD) est un programme logiciel informatique conçu comme moyen de gérer toutes les bases de données actuellement installées surtout disque dur système ou réseau **[22]**. La base de données contient les informations vitales du système. La sécurité de la base de données ne peut donc pas être ignorée. Protéger les données confidentielles et sensibles qui sont stockées dans une base de données est ce que nous appelons une base de données sécurisée **[23]**.

Fondamentalement, il existe cinq niveaux de sécurité : administrateur de base de données, administrateur système, responsable de la sécurité, développeur et employé. Ainsi, la

sécurité peut être affectée à n'importe quel niveau par un attaquant. Dans la sécurité des bases de données, les attaquants sont divisés en trois segments qui sont (voir **Figure 3.1**) **[24]** :

- **Pirate Administrateur**

Un administrateur est une personne autorisée qui a la permission de contrôler le système mais qui abuse de ses privilèges contre les politiques de sécurité pour obtenir les informations importantes.

- **Pirate Utilisateur**

Un employé est également membre d'un comité de confiance dans une organisation mais qui abuse de son autorité et souhaite obtenir des informations sensibles ou d'autres informations importantes.

- **Pirate externe**

Un intrus ne fait pas partie d'une organisation. En fait, il s'agit de personnes non autorisées qui accèdent aux données personnelles d'une organisation et souhaitent obtenir les informations sensibles.

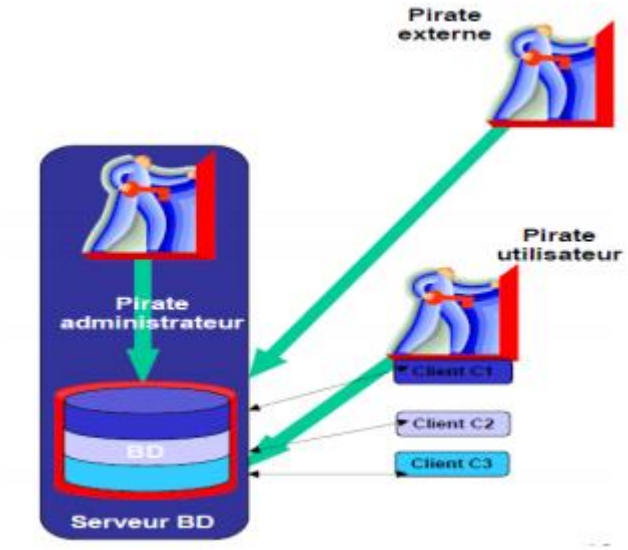

Figure 3.1. *Différents attaquants* ***[40]***

3. MENACES DE SECURITE DE BASE DE DONNEES :

En général, le risque majeur associé à toute attaque dépend de trois facteurs : menaces, vulnérabilités et impacts. Cette partie représente les dix principales menaces de sécurité dans les bases de données et comment elles fonctionnent **[26]** :

1. Injection SQL.
2. Abus excessif du privilège.

3. Abus du privilège légitime.

4. Elévation de privilèges.

5. Exploitation des vulnérabilités dans les zones vulnérables ouBases de données mal configurées.

6. Faiblesse de l'audit natif.

7. Déni de service.

8. Vulnérabilités des protocoles de communication des bases de données.

9. Copie non autorisée de données sensibles.

10. Exposition des données de sauvegarde.

L'infrastructure de la base de données d'une entreprise est soumise à un grand nombre de menaces, parmi les plus critiques Les menaces classées numéro 1 dans le monde numérique sont **SQL injection [27].** En luttant contre cette menace, les organisations répondront aux exigences de conformité mondiales telles qu'OWASP [28] et les meilleures pratiques de l'industrie pour la protection des données et l'atténuation des risques.

3.1. INJECTION SQL :

Dans une attaque par injection SQL, l'auteur insère généralement (ou "injecte") des informations de base de données non autorisées dans une chaîne de données SQL vulnérable. En règle générale, les données concernées incluent les chaînes, les procédures stockées et les paramètres d'entrée pour des applications Web. Cette information injectée est envoyée à la base de données où elle est exécutée. En utilisant l'injection SQL, les attaquants peuvent obtenir un accès illimité à une base de données **[38].**

3.2. ABUS EXCESSIF DES PRIVILEGES :

Lorsque les utilisateurs (ou applications) ont des privilèges d'accès à une base de données qui dépasse les exigences de leur fonction professionnelle, ils peuvent abuser de ces privilèges à des fins malveillantes.

Par exemple : *un doyen d'université/ dont la fonction principale ne requiert que la capacité pour modifier les coordonnées des étudiants peuvent bénéficier de privilèges excessifs de mise à jour de la base de données pour modifier les notes*.

Les utilisateurs de base de données peuvent se retrouver avec des privilèges excessifs pour la simple raison que la plupart du temps, les administrateurs de base de données n'ont pas le temps de définir ou de mettre à jour les mécanismes de contrôle / accès pour un utilisateur individuel. Au final tous les utilisateurs ou la majorité d'entre eux ont par défaut

des privilèges d'accès génériques qui dépassent de loin les exigences de leur fonction spécifique **[29]**.

3.3. ABUS DU PRIVILEGE LEGITIME :

Les utilisateurs peuvent également abuser des privilèges d'accès légitimes à une base de données à des fins non autorisées. Imaginons un potentiel d'agent de santé malveillant avec des privilèges de consultation des dossiers médicaux des patients via une application Web personnalisée, la structure de l'application Web limite normalement les privilèges des utilisateurs à la visualisation des dossiers médicaux d'un seul patient. Les consultations simultanées des dossiers et les copies électroniques ne sont pas permises. Cependant, l'attaquant peut contourner ces limitations en se connectant à la base de données en utilisant d'autres moyens tels que MS-Excel. Avec l'utilisation de MS-Excel et ses identifiants de connexion légitimes, l'employé peut récupérer et mettre à jour le dossier médical du patient. Il est peu probable que ces copies personnelles des dossiers patients soient conformes aux règles de protection des données des patients telle que définie par l'institutions des médicale, Il y a deux risques à considérer **[30]** :

- Le premier est le fonctionnaire malveillant qui tente de revendre les dossiers médicaux des patients.
- Le second (et peut-être le plus commun) est l'employé imprudent qui récupère et sauvegarde une grande quantité de données sur son ordinateur client à des fins commerciales légitimes. Une fois ces données sauvegardées sur un autre ordinateur, il devient vulnérable aux chevaux de Troie, au vol d'ordinateur portable, etc.

3.4. AUGMENTATION DES PRIVILEGES :

Les attaquants peuvent profiter de la vulnérabilité de la plate-forme logicielle de base de données pour transformer les privilèges d'accès d'un utilisateur ordinaire à ceux d'un administrateur. Toutes les vulnérabilités peuvent être trouvées dans la plupart des cas dans les Procédures, les fonctions intégrées, les protocoles implémentations, ou même dans les données SQL. Par exemple, un développeur de logiciels travaillant dans une institution financière peut profiter d'une fonction vulnérable pour revendiquer un privilège d'administrateur pour l'accès à la base de données. Avec ce privilège d'administrateur, le développeur malveillant peut désactiver les mécanismes d'audit, créer des comptes fantômes, transférer des fonds, etc **[31].**

3.5. EXPLOITATION DES VULNERABILITES DANS UNE BASE DE DONNEES VULNERABLE OU MAL CONFIGUREE :

Les bases de données sont souvent vulnérables, incorrectes, ou bien ont des configurations/comptes toujours définis par défaut. Alors que les fournisseurs développent des patchs pour corriger un système pour une vulnérabilité spécifique, les bases de données d'entreprises restent librement exploitables. Lorsqu'un correctif est publié, il n'est pas disponible immédiatement. Il y a plusieurs aspects à prendre en compte lors de l'application d'un correctif à une base de données. Tout d'abord, l'organisation doit d'abord évaluer la procédure de correction du système avec le correctif en question, en essayant de comprendre comment le correctif affecterait le système. Parfois une solution peut entrer en conflit avec le code existant ou impliquer d'autres opérations. Ensuite, le système subit un temps d'arrêt lorsque le serveur de base de données ne parvient pas à fournir aux utilisateurs un service pour le réparer. Enfin, les grandes entreprises avec des dizaines ou même des centaines de bases de données doivent fournir un plan de correction, en priorisant les bases de données, qui doivent être corrigé en premier. Par conséquent, il n'est pas surprenant de voir que pour de nombreuses entreprises, le processus de correction dure plusieurs mois, généralement entre 6 et 9 mois (durée établie sur la base des recherches menées par le groupe indépendant d'utilisateurs Oracle ou **IOUG**) **[32]**.

3.6. FAIBLESSE DE L'AUDIT NATIF :

Enregistrement automatique de tous les faits sensibles et / ou inhabituels les transactions de base de données devraient être la base sous-jacente de tout déploiement de base de données. Un faible audit de base de données représente de sérieux risques organisationnel pour plusieurs niveaux **[33] :**

- ***Risque réglementaire***

L'Organisations utilisant de faibles (ou parfois inexistants) mécanismes d'audit de base de données se rendront de plus en plus compte qu'ils ne respectent pas la réglementation fondamentale.

La loi réglementaire Sarbanes-Oxley (**SOX**) dans le domaine des services financiers et la loi réglementaire HIPAA (Healthcare Information Portability and Accountability Act) dans le domaine de la santé ne sont que deux exemples de réglementations gouvernementales avec des exigences claires en matière d'audit des bases de données.

- ***Dissuasion***

À l'instar des caméras vidéo qui enregistrent les visages des personnes entrant dans une banque, les mécanismes d'audit de base de données servent à dissuader les attaquants qui

savent que la surveillance des audits de base de données fournit aux enquêteurs des informations médico-légales sur les auteurs d'un crime.

- ***Détection et récupération***

L'audit est toujours la dernière étape pour défendre les bases de données. Si l'attaquant parvient à contourner les autres systèmes de défense, les résultats des audits peuvent identifier l'existence d'une violation après attaque. Les résultats de l'audit peuvent ensuite être utilisés pour lier une violation à un utilisateur particulier et / ou réparer le système. Les plates-formes logicielles de base de données intègrent généralement des fonctionnalités d'audit de base mais présentent de multiples faiblesses qui limitent ou empêchent leur déploiement.

- ***Manque de responsabilité utilisateur***

Lorsque les utilisateurs accèdent à une base de données via une d'application Web telle qu'**Oracle**, **SAP** ou **PeopleSoft**, généralement un mécanisme d'audit natif, ne connaît pas l'identité spécifique d'un utilisateur. Dans ce cas, toutes les activités de l'utilisateur sont associées au compte du nom de l'application Web. Par conséquent, lorsque les résultats des audits natifs révèlent l'existence de transactions frauduleuses dans la base de données, aucun lien ne peut être établi avec l'utilisateur responsable.

- ***Dégradation des performances***

Les mécanismes d'audit de base de données natifs sont connus pour consommer des ressources de processeur et de disque dur. La dégradation des performances observée lorsque les fonctionnalités d'audit sont activées oblige de nombreuses organisations à réduire le nombre d'audits ou simplement à les supprimer.

- ***Séparation des fonctionnalités***

Les utilisateurs disposant de droits d'accès administratifs (obtenus légitimement ou de manière malveillante) sur le serveur de base de données peuvent facilement désactiver la fonction d'audit pour masquer les activités frauduleuses. Idéalement, les fonctions d'audit devraient être séparées entre celles des administrateurs de base de données et de celles de la plate-forme du serveur de base de données.

- ***Granularité limitée***

La plupart des mécanismes d'audit natifs n'enregistrent pas les informations nécessaires pour prendre en charge la détection d'une attaque, même l'analyse médico-légale et la récupération. *Par exemple : l'application client de base de données, les adresses IP source, les éléments de réponse aux requêtes et les requêtes ayant échoué (un indicateur de reconnaissance d'attaque important) sont non enregistré par de nombreux mécanismes natifs.*

- ***Propriétaire***

Les mécanismes d'audit sont spécifiques à la plate-forme du serveur de base de données. Les résultats **Oracle** sont différents des résultats **MS-SQL**, les résultats **MS-SQL** sont à leur tour différents des résultats **Sybase**, et ainsi de suite. Pour les organisations qui combinent des environnements de base de données, cela élimine littéralement la mise en œuvre de procédures d'audit uniformes et évolutives dans l'entreprise.

3.7. DENI DE SERVICE :

Le déni de service (DOS) est une catégorie d'attaque générale qui refuse l'accès aux applications réseau à certains utilisateurs. Les conditions de déni de service peuvent être créées par de nombreuses techniques, dont beaucoup sont liées aux vulnérabilités susmentionnées. *Par exemple, un déni de service peut être obtenu en profitant de la vulnérabilité d'une plate-forme de base de données pour supprimer un serveur*.

La surcharge des ressources est une technique très courante dans les environnements de base de données. Les motivations des attaques par déni de service sont également diverses. Les attaques par déni de service sont principalement liées aux tentatives d'extorsion par lesquelles un pirate informatique installe à distance des serveurs jusqu'à ce que la victime place ses fonds sur un compte bancaire international. Le déni de service peut également être lié à une infection par un ver informatique. Quelle qu'en soit la source, le déni de service est une menace sérieuse pour de nombreuses organisations **[34].**

3.8. FAIBLESSE DES PROTOCOLES DE COMMUNICATION DE LA BASE DE DONNEES :

Un nombre croissant de vulnérabilités de sécurité est identifié dans les protocoles de communication de base de données conçus par tous les fournisseurs de bases de données. Les activités frauduleuses ciblant ces vulnérabilités peuvent aller de l'accès non autorisé aux données au déni de service. Le ver informatique **SQL slammer2**, par exemple, a profité d'une faille du protocole Microsoft SQL Serveur pour forcer un déni de service. Afin de compliquer la situation, il n'y a pas d'enregistrement de ces vecteurs de fraude dans le journalisme d'audit natif, car toutes les opérations de protocole ne sont pas couvertes par la majorité des mécanismes d'audit de base de données **[35].**

3.9. COPIE NON AUTORISEE DE DONNEES SENSIBLES :

De nombreuses entreprises s'efforcent de localiser et de maintenir correctement un inventaire de toutes leurs bases de données. De nouvelles bases de données peuvent être créées sans que l'équipe de sécurité en soit consciente et les données sensibles copiées dans

ces bases de données peuvent être exposées si les contrôles nécessaires ne sont pas appliqués. Ces bases de données « cachées » peuvent contenir des données potentiellement sensibles telles que les détails des transactions, ainsi que les informations de contact des clients et des employés. Cependant, si les personnes chargées de la sécurité des données ne connaissent pas le contenu de ces bases de données, il est difficile de s'assurer que les contrôles nécessaires ont été appliqués. Que ce soit intentionnellement ou non, les employés ou les pirates peuvent alors accéder illégalement à des données sensibles. Les anciennes bases de données qui ont été oubliées et laissées hors de portée en sont un exemple. Si personne ne gère ces bases de données, les données sont laissées sans surveillance en vue des regards indiscrets qui ne devraient pas accéder à ces données **[36].**

3.10. EXPOSITION DES DONNEES DE SAUVEGARDE :

Les périphériques de sauvegarde de base de données ne sont généralement pas protégés contre d'éventuelles attaques. Par conséquent, plusieurs failles de sécurité majeures sont apparues, notamment le vol de disques durs et de bandes de sauvegarde de bases de données **[37].**

4. TYPES D'ATTAQUES DE BASES DE DONNEES :

Les attaques effectuées sur la base de données sont essentiellement classées en deux segments **[24]** :

- **Attaques directes :**

Frapper directement les données cibles est appelé attaque directe. Ces attaques ne sont accessibles et réussies que si la base de données ne contient aucun système de protection. Si cette attaque échoue, l'attaquant passe à la suivante.

- **Attaques indirectes :**

Comme leur nom l'indique, les attaques indirectes ne sont pas exécutées directement sur la cible, mais les données provenant ou concernant la cible peuvent être collectées via d'autres objets de transition. Dans le but de tromper le système de sécurité, certaines des combinaisons de différentes requêtes sont utilisées. Ces types d'attaques sont difficiles à suivre.

Il existe une autre classification plus poussée des attaques contre la base de données comme suit :

- **Attaque passive (Interception, écoute, Analyse) :**

En cela, l'attaquant inspecte uniquement les données présentes dans la base de données et n'effectue aucune modification (voir **Figure 3.2**). L'attaque passive peut être effectuée des manières suivantes :

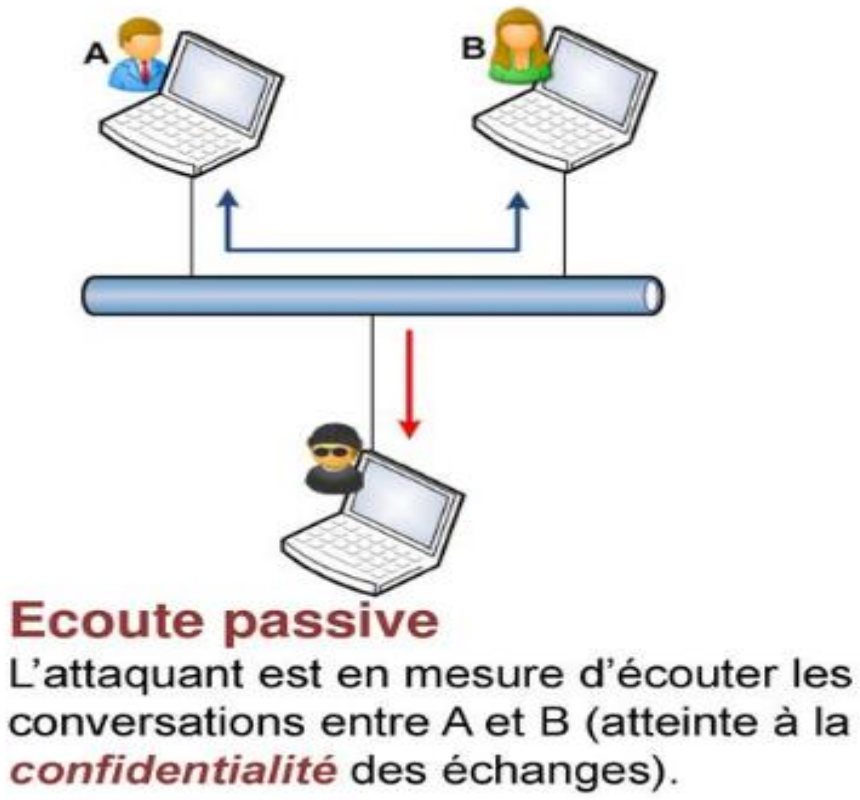

Figure 3.2. *L'attaque passive* ***[39]***

- **Fuite statique**

Dans ce type d'attaque, l'instantané de la base de données est observé dans le sens d'obtenir les valeurs de texte brut à un moment donné. La fuite statique ne traite que de l'observation des données dans la base de données uniquement à une période de temps spécifiée, mais après un certain temps, l'attaquant arrête l'observation sur les données Fondamentalement, ce n'est pas très dangereux car les données restent les mêmes et les données appropriées sont reçues par la bonne personne mais l'attaque se produit parce que l'attaquant observe simplement les données de la base de données. Elle est appelée fuite statique car elle n'est effectuée que pendant une période de temps spécifiée **[24]**.

- **Fuite de liaison**

Dans ce type d'attaque, la liaison entre la valeur de la base de données et la position de cette valeur spécifiée dans l'index est établie pour obtenir la valeur en texte brut. En cas de fuite de liaison, certaines étapes sont prises pour effectuer réellement l'attaque de liaison :

-La première étape de la fuite de liaison consiste à vérifier l'index de la base de données et à rechercher les données particulières sur lesquelles l'attaque doit être effectuée.
-Dans la deuxième étape, lorsque la valeur de données requise est trouvée dans un index de la base de données, les données sont liées à la valeur de la base de données. La fuite de liaison crée des problèmes mais elle n'est pas aussi dangereuse que d'autres attaques **[24].**

- **Fuite dynamique**

Dans ce type d'attaque, la valeur de texte brut peut être générée en observant les changements continus effectués dans la base de données pendant un moment donné. Ensuite, après avoir observé les changements, les données sont analysées, ce qui aide l'attaquant à obtenir les données associées sur la valeur de texte brut. Les principales étapes de la fuite dynamique sont **[24].**
- Dans la première étape, l'attaquant observe les données transmises entre les utilisateurs pendant un certain temps.
-Dans la deuxième étape, les données observées sont analysées, ce qui aboutit aux informations associées de la valeur en texte brut.

- **Attaques actives (modification, destruction/ interception, perturbation) :**

Lors d'une attaque active, les valeurs réelles de la base de données sont modifiées. Celles-ci sont plus problématiques que les attaques passives car elles peuvent induire un utilisateur en erreur (voir **Figure 3.3**). Par exemple, un utilisateur capturant des informations erronées à la suite d'une requête **[27]**. Il existe différentes manières pour effectuer ce type d'attaque qui sont mentionnées ci-dessous :

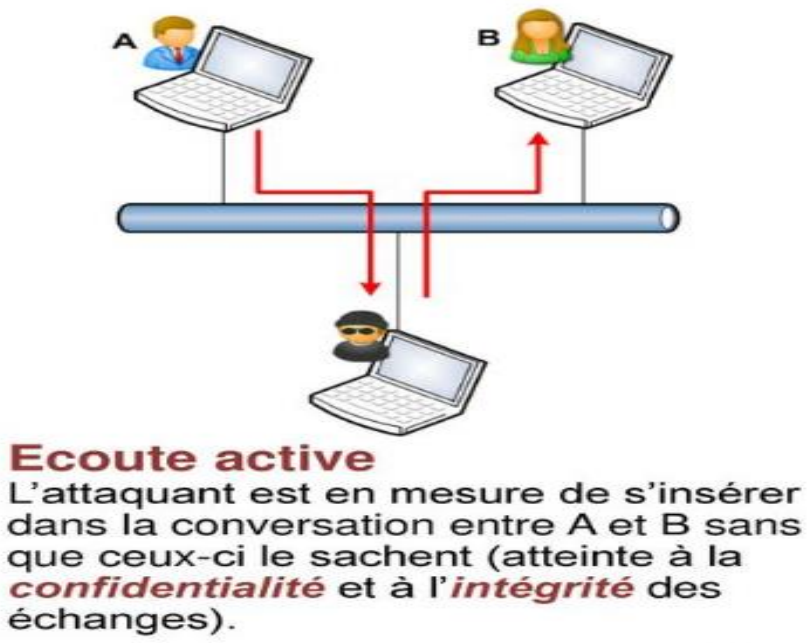

Figure 3.3**. L'attaque active **[39]

- **Usurpation d'identité (Spoofing)** :

Dans une attaque d'usurpation d'identité, l'attaquant remplace simplement un bloc de mémoire par des données malveillantes. En fonction de la structure du système, ainsi que du code opérant sur celui-ci, les données placées par l'attaquant peuvent créer des problèmes d'exécution du programme. Bien que le but de l'usurpation de mémoire ne soit pas d'obtenir des données sensibles, l'attaque peut permettre à l'attaquant d'accéder à des fonctionnalités non autorisées du système. Les attaques par usurpation d'identité sont également utilisées pour contourner les contrôles de licence dans les systèmes (voir **Figure 3.4**) **[25]**.

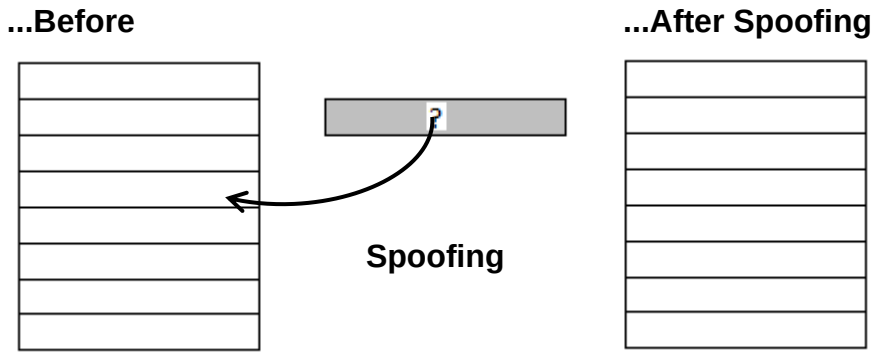

Figure3.4.** L'attaque spoofing **[25].

- **Epissage (Splicing)** ***:***

Dans une attaque d'épissage, l'attaquant intercepte une lecture depuis la mémoire, fournissant un bloc de données valide à partir d'une adresse différente de celle demandée. En faisant cela, le processeur fonctionnera désormais sur les données incorrectes. Une attaque

d'épissage est considérée comme une permutation spatiale du bloc mémoire (voir **Figure 3.5**) **[25].**

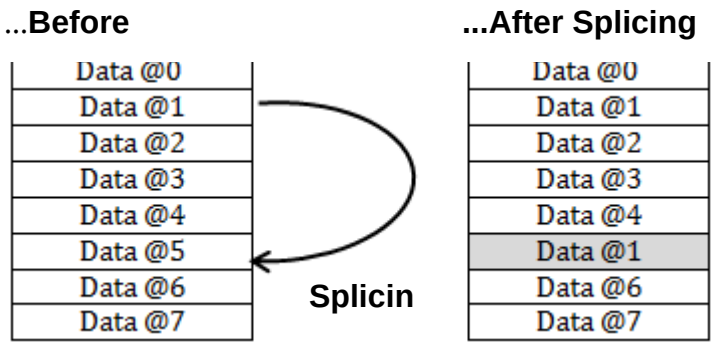

Figure 3.5. *L'attaque splicing* ***[25].***

- **Relecture (Replay)** :

Dans une attaque de relecture, l'attaquant remplace une valeur actuelle en mémoire par une valeur qui était précédemment stockée à l'adresse à un moment antérieur. En faisant cela, l'attaquant peut fournir des données pour accéder aux différentes fonctionnalités du système. Une utilisation courante d'une attaque de relecture se trouve dans le réseautage. Si un attaquant écoutant un canal est capable de capturer les données d'authentification d'un utilisateur lorsqu'elles sont transmises au serveur, il peut alors soumettre à nouveau ces données et accéder au système. Ce type d'attaque était l'une des invites pour un remplissage unique et une authentification unique dans ces systèmes. Dans le matériel, les attaques de relecture vont vers un objectif similaire (voir **Figure 3.6**) [**25**].

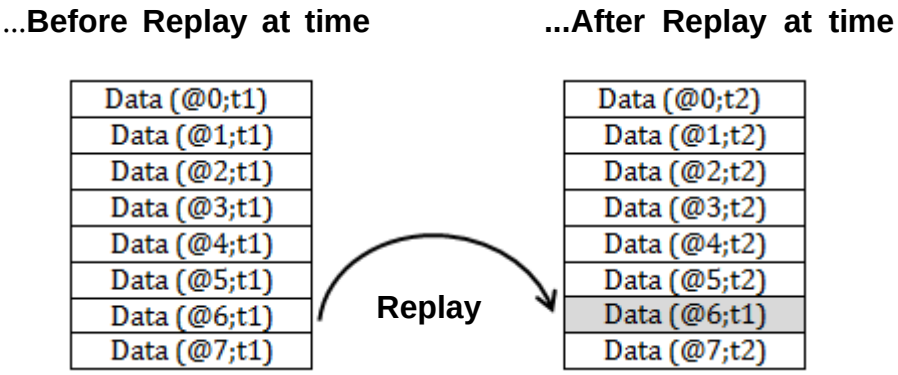

Figure 3.6. *L'attaque replay* ***[25].***

5. LES CONTRES MESURES :

Principalement la protection d'une base de donnes consiste à (voir **Figure**3.**7**) **[9]** :

-Sécuriser les applications qui y accèdent.

- Contrôler l'accès des utilisateurs.

- Crypter les communications.

- Crypter les données.

- Avoir un Audit.

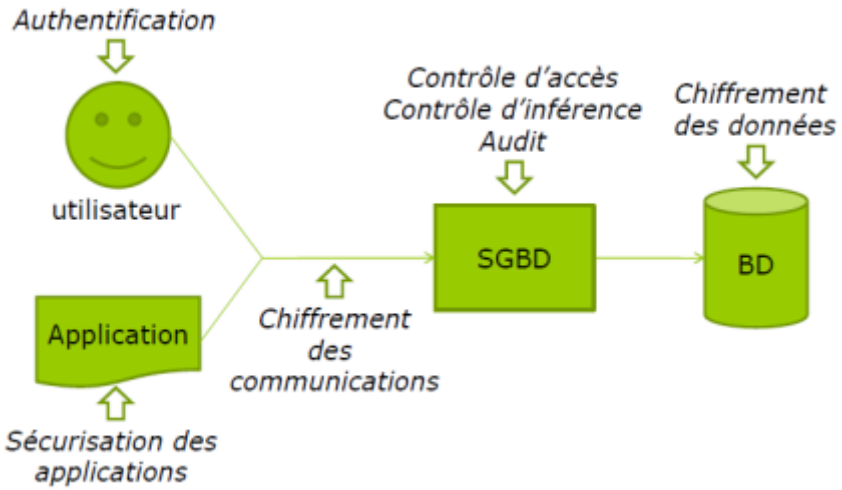

Figure3.7. *Sécurité d'une base de données* ***[32].***

Les mécanismes de sécurité des bases de données sont généralement classés en deux catégories **[9]** :

5.1. LES MESURES DE SECURITE STRATEGIQUES :

- ***Séparation des privilèges :***

- renseigner les rôles dans l'application (Admin, Mise à jour, Lecture ...)
- appliquer ces rôles dans les privilèges attribués.

- ***Audit d'application :***

- Parler de sécurité avec les développeurs (SQL Injection, débordement de tampon, validation des entrées ...).
- Recherche de points critiques, de flux d'utilisateurs ...
- Audit des sources Java, ASP, PHP, Perl, C, etc.

5.2. MESURES DE SECURITE TECHNIQUES :

Cette partie représente différentes techniques des mesures prises en compte pour assurer la sécurité de la base de données, en voici quelques-unes **[9]**:

- **Prévention de l'injection SQL :**

Trois techniques peuvent être combinées pour lutter efficacement contre l'injection SQL : la technologie de prévention des intrusions (IPS), le contrôle d'accès aux requêtes (voir Remplacement excessif des privilèges) et la corrélation d'événements. Le mécanisme IPS peut identifier les procédures stockées ou les chaînes d'injection SQL les plus vulnérables. Généralement, IPS seul n'est pas fiable car l'injection SQL donne beaucoup de faux positifs.

Les responsables de la sécurité qui se fient uniquement à la technologie IPS seraient bombardés d'alertes sur les « possibles » injections SQL. En pratique, en construisant une signature d'injection SQL avec un autre type de violation, comme une violation de contrôle d'accès aux requêtes, une réelle attaque peut être identifiée très précisément. Il est peu probable qu'une signature d'injection SQL et un autre type de violation apparaissent dans la même requête au cours d'une opération commerciale classique.

- **Prévention de l'abus excessif de privilèges :**

Une solution à l'abus excessif des privilèges consiste à éliminer les droits excessifs, ce qui nécessite la capacité d'identifier les droits excessifs, c'est-à-dire les droits qui ne sont pas nécessaires à l'utilisateur pour exercer sa fonction. Cela se fait en extrayant les droits des bases de données, en corrélant les droits avec l'utilisateur métier et enfin en analysant ces droits. Il s'agit d'une procédure décourageante qui, si elle est effectuée manuellement, nécessite à la fois du temps et des ressources. Une solution automatisée peut réduire considérablement le temps et les ressources nécessaires et raccourcir le processus d'analyse.

Afin de mieux appliquer les droits d'accès, des contrôles d'accès pour les requêtes granulaires sont également nécessaires. Le contrôle d'accès aux requêtes fait référence à un mécanisme qui limite les privilèges d'accès aux bases de données à un minimum d'opérations SQL (SELECT, UPDATE, etc.) et aux données. La granularité du contrôle d'accès aux données doit être étendue de la table simple aux lignes et colonnes spécifiques de la même table. Un mécanisme de contrôle d'accès aux requêtes suffisamment granulaire permettrait au principal universitaire malveillant décrit précédemment de mettre à jour les informations de contact de l'étudiant, mais déclencherait une alerte si l'étudiant tentait de modifier les notes. Le contrôle d'accès aux requêtes est utile non seulement pour détecter les abus de privilèges excessifs par des employés malveillants, mais également pour prévenir la plupart des 10 principales menaces décrites dans ce document.

- **Prévention de l'abus du privilège légitime :**

La solution d'abus de privilèges légitime est le contrôle d'accès à la base de données, qui s'applique non seulement aux demandes d'accès spécifiques décrites ci-dessus, mais également au contexte d'accès à la base de données. En appliquant une règle de contrôle pour les applications clientes, l'heure et le lieu de la demande d'accès, etc., il est possible d'identifier les utilisateurs qui utilisent des privilèges d'accès légitimes à la base de données de manière suspecte.

- **Prévention de l'élévation des privilèges :**

Les abus d'élévation de privilèges peuvent être évités en combinant un système de prévention des intrusions (IPS) traditionnel et un contrôle d'accès aux requêtes (voir la section Abus excessif de privilèges précédemment décrite). IPS inspecte le trafic de la base de données pour identifier les modèles qui correspondent aux vulnérabilités existantes. Par exemple, si une fonction spécifique est connue pour être vulnérable, une technologie IPS peut soit bloquer tout accès à la procédure vulnérable, soit (si possible) bloquer uniquement les procédures avec des attaques intégrées. Malheureusement, cibler uniquement les demandes d'accès à la base de données avec des attaques précisément intégrées peut être difficile en utilisant uniquement IPS. De nombreuses fonctions de base de données vulnérables sont utilisées à des fins légitimes. Il n'est pas recommandé de bloquer toutes les occurrences de ces fonctions. La technologie IPS sépare précisément les fonctions légitimes des fonctions qui incluent des attaques. Dans de nombreux cas, les variations infinies d'attaques rendent cette distinction impossible. Dans ces conditions, les systèmes IPS peuvent être utilisés en mode alerte uniquement (et non en mode blocage) car il y a des chances d'obtenir des faux positifs. Pour améliorer la précision, la technologie IPS peut être combinée avec d'autres indicateurs d'attaque tels que le contrôle d'accès aux requêtes. IPS peut être utilisé pour vérifier si la demande d'accès à la base de données utilise ou non une fonction vulnérable, tandis que le contrôle d'accès aux requêtes contrôle si la requête correspond ou non à un profil d'utilisateur typique. Si une seule requête indique l'accès à une fonction vulnérable ou à un profil utilisateur inhabituel, une tentative d'attaque est certainement en cours.

- **Prévention de l'exploitation des vulnérabilités**

Afin de limiter le risque de menace des bases de données non corrigées et vulnérables, il faut d'abord évaluer l'état de sécurité des bases de données et corriger les vulnérabilités et lacunes de sécurité identifiées. Les entreprises doivent périodiquement analyser les bases de données pour détecter les vulnérabilités et les correctifs manquants. Les évaluations de la configuration doivent fournir une vue d'ensemble claire de l'état actuel de la configuration des systèmes de données. Ces évaluations doivent également identifier les bases de données qui ne sont pas conformes aux règles de configuration définies. Tous les correctifs de sécurité manquants doivent être déployés dès que possible. Si une vulnérabilité est découverte alors que le correctif n'est pas encore disponible, soit parce qu'il n'a pas encore été lancé par le fournisseur, soit parce qu'il n'a pas encore été déployé, un correctif virtuel doit être défini. Une telle solution bloque les tentatives d'exploiter ces vulnérabilités. La réduction de la

fenêtre d'exposition obtenue en appliquant un correctif virtuel aidera à protéger la base de données des tentatives d'exploitation jusqu'à ce qu'un correctif soit déployé.

- **Prévention de la faiblesse de l'audit natif :**

Les systèmes d'audit de la qualité du réseau corrigent la plupart des faiblesses associées aux outils d'audit natifs.

Haute performance - les périphériques basés sur la qualité du réseau peuvent appliquer la vitesse de ligne sans aucun impact sur les performances de la base de données. En fait, en transférant la responsabilité des procédures d'audit aux applications réseau, les organisations peuvent espérer améliorer les performances des bases de données.

Les dispositifs d'audit en réseau peuvent fonctionner indépendamment des administrateurs de base de données, ce qui permet une séparation appropriée des fonctions d'audit des fonctions administratives. De plus, comme les périphériques réseau sont indépendants du réseau lui-même, ils sont également invulnérables aux attaques par élévation de privilèges par des utilisateurs non administrateurs.

Les périphériques d'audit basés sur le réseau prennent généralement en charge les principales plates-formes de bases de données qui peuvent appliquer des critères uniformes et des procédures d'audit centralisées dans des environnements de base de données volumineux et hétérogènes. Combinées, ces fonctionnalités réduisent les coûts d'exploitation du serveur de base de données, les exigences d'équilibrage de charge et les coûts administratifs. Ils offrent également une meilleure sécurité.

La surveillance régulière des journaux permet d'identifier les risques et les menaces susceptibles d'endommager les bases de données. Si, par exemple, un utilisateur malveillant (intrus) est capable de surpasser d'autres systèmes de défense, les audits peuvent identifier les violations après une attaque, et les journaux et audits peuvent également être utilisés pour réparer le système (avec des mises à jour du système) et revenir à l'identité de l'auteur de l'attaque.

- **Prévention du déni de service :**

La prévention du déni de service nécessite des protections à plusieurs niveaux. Ce chapitre traite des protections spécifiques à la base de données sans oublier les protections réseau, application et base de données nécessaires. Dans ce point précis, le déploiement de la connexion de flux de contrôle, de la technologie IPS, des applications de contrôle d'accès et du temps de réponse du contrôle sont recommandé.

En supprimant les fonctionnalités indésirables et en configurant uniquement ce qui est nécessaire pour une base de données, le déni de service (DoS) peut être évité dans une

certaine mesure. Les limites de ressources sont une autre mesure préventive qui peut rendre difficile les attaquer contre le système. Des correctifs de sécurité doivent être appliqués régulièrement et les administrateurs doivent exécuter un rapport de sécurité pour vérifier en permanence les vulnérabilités de sécurité pour empêcher DoS.

- **Prévention des vulnérabilités des protocoles de communication de la base de données** :

La technologie de validation de protocole peut être utile pour traiter les vulnérabilités du protocole de communication de base de données. Dans cette technologie, le trafic de la base de données est analysé et comparé à ce qui est vraiment attendu. Les chercheurs travaillent à créer un mécanisme qui peut fournir une validation proactive des messages de protocole lorsqu'ils circulent du client vers les serveurs. Tout message suspect non conforme au modèle attendu est signalé et ignoré. Ce mécanisme aidera grandement à détecter les bogues et les vers et limitera les vulnérabilités connues et inconnues.

- **Prévention de la copie non autorisée de données sensibles :**

Afin de maintenir un inventaire précis des bases de données et un emplacement précis des données sensibles, les organisations doivent identifier toutes les bases de données du réseau contenant des données sensibles. La deuxième étape consiste à déterminer quels types de données sensibles ou classifiées sont contenus dans les objets des bases de données. La classification des données représente deux difficultés majeures, la première étant de localiser les données sensibles parmi le grand nombre et les grandes tailles des tableaux. La deuxième difficulté est de trouver des combinaisons de données qui en elles-mêmes sont considérées comme inoffensives, mais qui, lorsqu'elles sont combinées avec d'autres données, forment une combinaison de données considérées comme sensibles. Afin de protéger adéquatement les données sensibles, les contrôles nécessaires doivent être définis conformément aux politiques d'accès aux données de l'organisation, une fois qu'un inventaire précis des bases de données et l'emplacement des données sensibles sont disponibles

- **Prévention de l'exposition des données de sauvegarde :**

Toutes les sauvegardes de base de données doivent être chiffrées. En fait, certains fournisseurs ont suggéré que les futurs systèmes de gestion de bases de données ne devraient pas prendre en charge la création de sauvegardes non chiffrées. Le cryptage des informations à partir de bases de données de production en ligne est souvent suggéré, mais les performances et les inconvénients de la gestion des clés cryptographiques rendent souvent cette solution peu pratique et est généralement reconnue comme un modeste substitut aux contrôles des droits d'auteur. Accès granulaire décrit ci-dessus.

6. CONCLUSION :

La protection des données cruciales est toujours une tâche difficile pour une organisation à tout moment. Les bases de données sont les cibles les plus préférées des attaquants en raison des informations qu'elles contiennent et de leurs volumes.

Dans ce chapitre nous avons vu différentes attaques qui peuvent détériorer la sécurité des bases de données. Nous avons aussi abordé les méthodes de prévention des menaces de sécurité de base de données.

Le prochain chapitre lui sera consacré à l'étude de différents types de contrôle d'accès.

Chapitre 04 :
Le contrôle d'accès

1. INTRODUCTION :

Généralement, dans un système informatique on considère souvent la confidentialité, l'intégrité et la disponibilité des données comme les trois propriétés fondamentales à respecter pour assurer la sécurité (Figure4.1). De nombreuses pratiques et dispositifs participent à garantir ces propriétés de sécurité : la protection des réseaux, le cryptage de l'information, la sauvegarde des données, les architectures d'authentification. Parmi les moyens mis en œuvre pour renforcer la sécurité, nous allons nous intéresser aux politiques de contrôle d'accès.

Ce chapitre commence par définir la politique de contrôle d'accès et ces types (statique et dynamique) et aussi par montrer son importance dans la sécurité des systèmes d'informatiques. Ensuite, nous montrons les étapes du mécanisme de moniteur de référence mis en œuvre pour réaliser le contrôle d'accès. Puis nous expliquons les différents modèles de contrôle d'accès largement utilisés dans le monde industriel (les modèles classiques et les modèles à base de rôle) en présentant ces principes de fonctionnement et ces points faibles.

2. LE CONTROLE D'ACCES :

Le contrôle d'accès, dans un système informatique, est l'ensemble des mesures mises en place pour restreindre l'accès aux ressources du système suivant des contraintes préétablies **[72]**. Le contrôle d'accès est aussi un mécanisme grâce auquel un système autorise ou interdit les actions demandées par des sujets (entités actives) sur des objets (entités passives). Il renforce particulièrement la confidentialité et l'intégrité de l'information, a fortiori sa disponibilité **[71]**.

Il existe de nombreux modèles de contrôle d'accès. Une instance d'un tel modèle représente **une politique de contrôle d'accès**. Cette dernière définit donc les accès aux ressources d'un système **[72].**

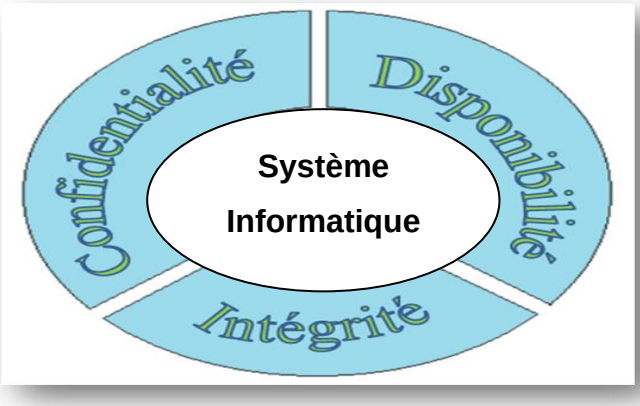

Figure4.1. Les trois propriétés fondamentales de la sécurité **[74]**

Un modèle de contrôle d'accès comprend **[75]** :

-Une politique de contrôle d'accès qui spécifie les accès autorisés aux données.

-Une politique d'administration qui indique comment la politique de contrôle d'accès peut être mise à jour.

2.1. POLITIQUES DE CONTROLE D'ACCES :

Les contraintes qui régissent les accès aux ressources d'un système peuvent être de nature statique ou dynamique. On distingue ainsi deux types de politiques **[72]** :

- Les politiques de contrôle d'accès statique.
- Les politiques de contrôle d'accès dynamique.

2.1.1. Politique de contrôle d'accès statique :

Pour un système informatique donné, une politique de contrôle d'accès statique est caractérisée par le fait que son état ne change pas par rapport à l'évolution dynamique du système, car elle comporte seulement des contraintes statiques. Celle-ci peut être mise à jour pour refléter divers changements dans l'organisation (par exemple, un changement d'affectation dans une organisation qui entraîne une augmentation des privilèges d'un utilisateur). L'initiation d'une action par un utilisateur déclenche l'évaluation de l'état de la politique. En fonction des autorisations accordées par la politique dans son état courant, l'exécution de l'action est permise ou pas. Dans la plupart des implémentations, une politique de contrôle d'accès statique associe les utilisateurs du système à leurs privilèges **[72].**

2.1.2. Politique de contrôle d'accès dynamique :

Pour un système informatique donné, une politique de contrôle d'accès dynamique possède plusieurs états, car elle est associée à l'évolution du système. L'autorisation de l'exécution d'une action est basée sur l'évaluation de l'état courant du système et sur la définition même de la politique. L'état courant est mis à jour lors de chaque exécution d'une action contrôlée. La version élémentaire de ce type de politiques de contrôle d'accès utilise un historique des actions exécutées par le système d'informatique qui est mise à jour par le gestionnaire de mise en œuvre de la politique. Les langages formels qui supportent les traces d'événements, comme les langages basés sur une algèbre de processus, se prêtent bien à l'expression de contraintes dynamiques **[72]**.

2.2. MONITEUR DE REFERENCE :

L'application du contrôle d'accès également appelée enforcement est assurée par un module appelé moniteur de référence **(**un moniteur, intermédiaire entre les utilisateurs et les

ressources auxquelles ces derniers essaient d'accéder) qui intercepte tous les accès aux données et détermine s'ils sont légitimes ou non. L'ensemble des droits et conditions appliqués par le moniteur de référence constitue la politique de contrôle d'accès qui peut être représentée et exprimée de multiples façons **[76].**

La définition d'un tel moniteur est délicat, car ce dernier doit être *incontournable, inviolable* et *vérifié*. Lorsqu'un utilisateur demande un accès, le moniteur va décider si cet accès est autorisé ou non d'après la politique de contrôle d'accès : une instance spécialisée de la politique de sécurité logique, qui s'attache à définir les droits des utilisateurs des systèmes **[71].** Le principe de fonctionnement d'un moniteur est décomposable en 6 étapes, comme l'illustre (voir **la figure 4.2) [71]** :

1. Envoi de la requête de l'utilisateur au moniteur.
2. Interrogation de la politique de contrôle d'accès.
3. Réponse de la politique.
4. Le moteur accède à la ressource si l'accès est autorisé pour exécuter la requête.
5. Retour de l'exécution de la requête.
6. Retour de la requête ou exception en cas d'accès non autorisé.

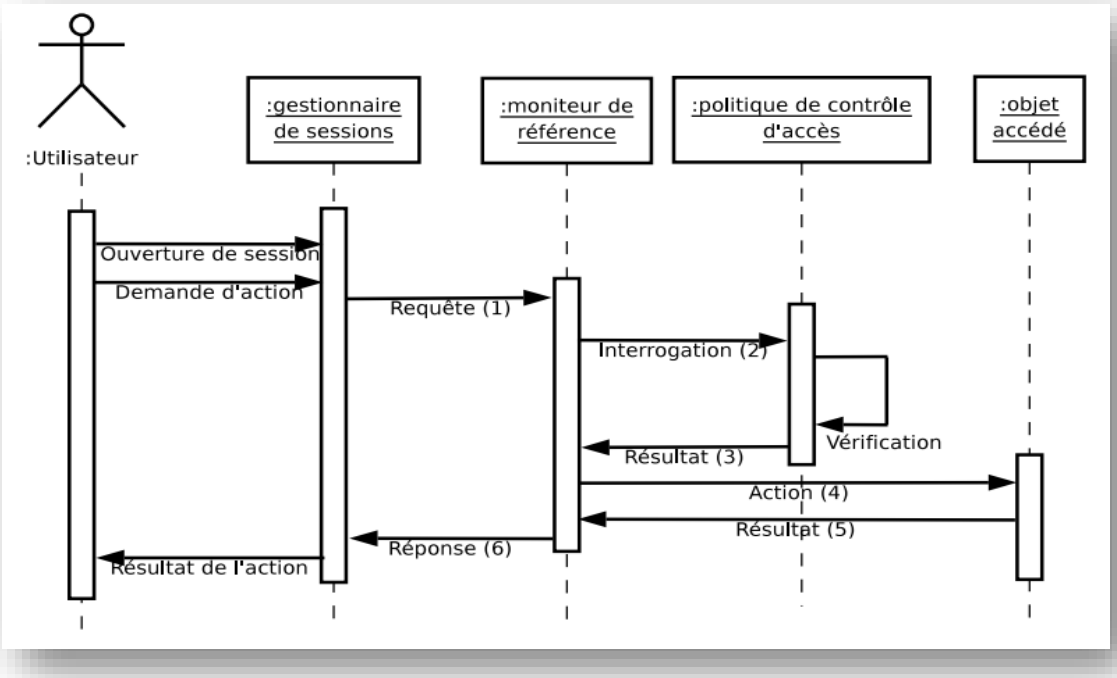

Figure 4.2. *Mécanisme de moniteur mis en œuvre pour réaliser le contrôle d'accès* **[71].**

2.3. FORMALISATION DU CONTROLE D'ACCES :

Le contrôle d'accès dans un système informatique permet de contraindre l'accès à ses ressources. Une façon d'interpréter une politique est de considérer un système composé d'un ensemble d'états *Q* et d'un ensemble de transitions entre ces états, où chaque transition représente un accès à ses ressources. Ainsi une politique de contrôle d'accès partitionne l'ensemble des états *Q* en un ensemble d'états autorisés $Qaut \subseteq Q$ et un ensemble d'états interdits. La mise en œuvre des politiques de contrôle d'accès est réalisé par un *mécanisme de sécurité* qui empêche le système de se retrouver dans un état interdit. Désignons *Qacc*l'ensemble des états accessibles par le mécanisme de sécurité. Le mécanisme est dit *sûr* lorsque $Qacc \subseteq Qaut$. Dans ce cas, les états accessibles par le mécanisme de sécurité sont tous des états autorisés par la politique de contrôle d'accès. Lorsque $Qacc = Qaut$, le mécanisme de sécurité est dit *précis* et tous les états autorisés par la politique sont accessibles par le mécanisme de sécurité et vice-versa. Dans le cas général où $Qacc \cap Qaut \neq \emptyset$, le mécanisme de sécurité est dit *large*, car le système peut se retrouver dans un état interdit. Les modèles présentés dans la suite sont des moyens qui précisent comment les di!érents accès au système sont réalisés de telle sorte que son état courant soit toujours dans l'ensemble *Qaut* **[72]**.

3. LES MODELES DE CONTROLE D'ACCES :

Il existe différents modèles de contrôle d'accès, chacun étant est adapté à des besoins différents. Nous détaillons ici quelques modèles de contrôle d'accès largement étudiés dans la sécurité informatique et beaucoup utilisés dans le monde industriel. Aussi nous expliquant les différents modèles de contrôle d'accès existants en indiquant leur domaine d'application. Ces modèles sont répartis dans différentes catégories **[75]** :

1. Les modèles de contrôle d'accès classiques :
   - Le modèle discrétionnaire (**DAC :** Discretionary Access Control).
   - Le modèle obligatoire (**MAC :** Mandatory Access Control).
2. Les modèles de contrôle à base de taches (**TBAC :** TaskBased Access Control**).**
3. Les modèles de contrôle à base de rôles (**RBAC :** RoleBased Access Control**)**.
4. Les modèles de contrôle à base d'organisation (**OR-BAC :** OganizationRoleBased Access Control).
5. Les modèles de contrôle d'accès contextuels (**CBAC :** Contexte Based Access Control**)**.

Pour qu'on se mette d'accord on considère les notions suivantes :

- **Un sujet (S)** : Un sujet peut être un processus, un utilisateur ou une application.
- **Un objet (O)** : Un objet est un conteneur d'informations sur lequel un sujet peut effectuer des actions (exemples : fichiers, sockets de communication, périphériques matériels. etc.).
- **Une action (A)** : représente l'action à traiter par le sujet sur l'objet. (Exemples : lecture, écriture, exécution d'un fichier, envoi de signaux ou de messages interprocessus, …).

3.1. CONTROLE D'ACCES DISCRETIONNAIRE DAC :

Désigné par l'acronyme **DAC**, la politique de contrôle d'accès discrétionnaire est représentée sous la forme d'une série de triplets (sujet, action, objet). Chaque triplet (**S,A, O**) signifie « le sujet **S**a la permission d'effectuer l'action **A** sur l'objet **O**», ce modèle permet d'associer l'identité d'un sujet à un ensemble d'autorisations sur des objets. Lorsqu'un sujet fait une requête sur un objet, l'action est accordée si et seulement si une autorisation le permet **[75]**.

3.1.1 Principe de la politique DAC :

La spécification de la politique DAC se concrétise par l'aspect discrétionnaire qui s'interprète par l'attribution des droits d'accès de la façon suivante :

- Un utilisateur du système peut lui-même attribuer des droits à d'autres sujets sur les objets dont il possède des autorisations. En revanche, il ne peut pas attribuer de droits qu'il ne possède pas (***Figure 4.3***) **[76]**.

- Le sujet propriétaire d'un objet a toute latitude pour décider quels autres sujets peuvent exercer des actions sur l'objet (lecture, écriture, exécution) **[69]**.

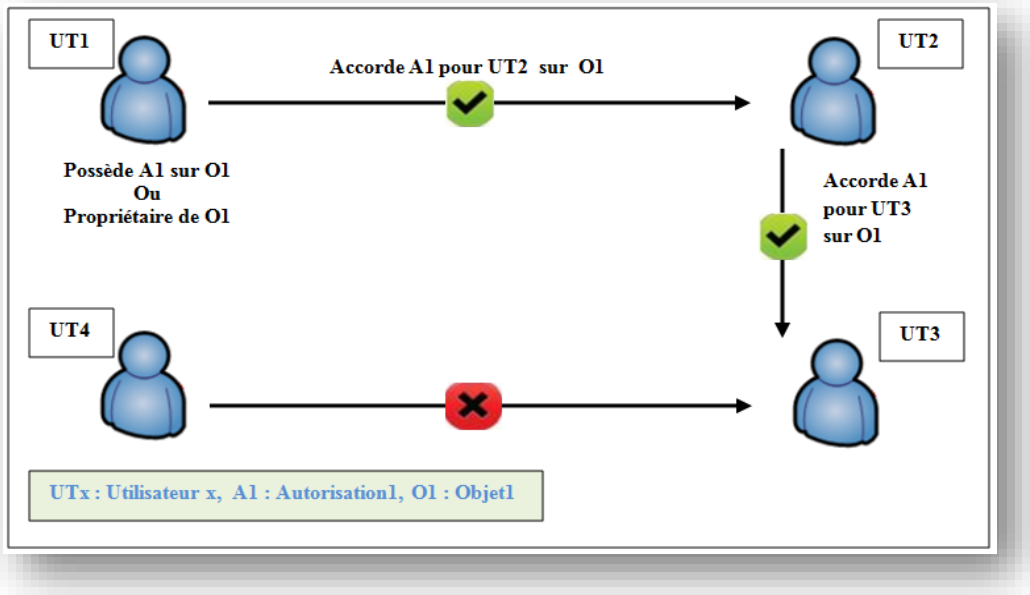

Figure 4.3. *Un exemple de modèle DAC* **[68].**

L'implantation de ce modèle a donné lieu à la constitution de matrices d'accès, l'état du système est défini par un triplé (*S*, *O*, *M*) où *S* représente l'ensemble des sujets (e.g. utilisateur, processus etc.) pouvant exercer un ensemble d'actions. *O* représente l'ensemble des objets (e.g. fichier, table, classe, programme etc.). Enfin, *M* représente la matrice d'accès, où les lignes correspondent aux sujets et les colonnes correspondent aux objets (***Tableau 4.1***) **[69]**.

Objets / Sujets	Fichier	Tables
Ahmed	Lire Ecrire	Ecrire Lire
Sara	Lire Ecrire Exécuter	Lire

Tableau 4.1. *Exemple d'une matrice d'accès* **[69].**

Les droits correspondent généralement à des actions élémentaires comme lire, écrire, exécuter ou posséder (mais ne sont pas limités à ces derniers). En effet, si de nouveaux objets, de nouveaux sujets ou de nouvelles actions sont ajoutés dans le système, il devient nécessaire d'enregistrer toutes les permissions accordées pour ces nouvelles entités [69].

Il existe en pratique deux approches pour implémenter la matrice d'accès **[69]** :

- Par une liste de contrôle d'accès (ou **ACL** pour **Access Control List**) : la matrice est stockée par colonne. A chaque objet est associée une liste de règles indiquant pour chaque utilisateur les actions pouvant être exercées par ce dernier sur cet objet.
- Par une liste de capacité (ou capabilité) : la matrice est stockée par ligne. A chaque utilisateur correspond une liste, appelée liste de capacité, indiquant pour chaque objet les actions que l'utilisateur est en droit d'effectuer sur cet objet.

Le contrôle d'accès **DAC** a été principalement implanté au sein des systèmes d'exploitation (Microsoft Windows, Solaris, Linux, FreeBSD). Les utilisateurs peuvent ainsi transférer leurs droits à d'autres utilisateurs sur les données ou services qu'ils contrôlent sans avoir besoin d'une autorité centrale qui pilote le tout. Cet aspect discrétionnaire le rend flexible et adapté à de nombreux systèmes multi-utilisateurs **[76]**.

3.1.2. Points faibles de la politique DAC :

Diverses études ont montré la faiblesse des modèles **DAC**. En effet, le contrôle d'accès discrétionnaire repose sur la capacité des utilisateurs à définir correctement les permissions sur les fichiers dont ils sont propriétaires. Toute erreur peut mener à une défaillance de sécurité, Cependant une fois qu'un propriétaire accorde un accès pour un utilisateur sur un objet, il n'a plus aucun contrôle sur les futurs accès qui pourraient être créés. Les attaques possibles contre les systèmes d'exploitation visent à obtenir un accès de niveau super-utilisateur (par exemple, root). Lorsqu'une telle attaque est réussie, elle obtient des pouvoirs qui outrepassent le **DAC** et donnent un accès complet à l'ensemble des ressources du système informatique. De fait, la faiblesse de ce contrôle est que la politique de sécurité peut être à tout moment modifiée par le super-utilisateur du système d'exploitation **[77]**. C'est exactement ce que les modèles de contrôle d'accès obligatoire **MAC** cherchent à éviter.

3.2. MODELES DE CONTROLE D'ACCES OBLIGATOIRES MAC :

Les modèles obligatoires de contrôle **MAC** ont été mise en œuvre afin d'apporter des solutions aux problèmes de sécurité informatique et pour rependre à la faiblesse des modèles **DAC.** La politique **MAC** repose sur la délégation du contrôle d'accès à une entité indépendante. L'existence de cette entité indépendante garantit que la politique de sécurité ne soit pas modifiable directement par les utilisateurs du système informatique **[77].**

On distingue deux orientations dans les modèles de type **MAC** :

- Le premier modèle, appelé modèle de **BelletLa Pudula**, a été développé pour le département de la défense américain et vise, plus particulièrement, à assurer la confidentialité des données dans le contexte de l'utilisation partagée de mainframes **[18]**. Ces modèles répondent à des problématiques très précises, par exemple la nécessité de disposer d'une habilitation adéquate pour la lecture de documents classifiés dans **BLP (**Modèle de confidentialité de Bell et La Padula**)**. Cependant, l'usage courant des systèmes d'exploitation moderne ne peut souvent pas être décrit par une seule de ces problématiques, mais relèvent plutôt de problématiques plus complexes comme le principe de moindre privilège, la confidentialité des données entre utilisateurs, la nécessité de prendre en compte les activités des utilisateurs sur le système **[77].**
- Le deuxième modèle, appelé modèle de **Biba** qui prend en compte les exigences commerciales s'intéresse plutôt à l'intégrité des informations **[78].** Ce modèle est appelé aussi le modèle **DTE** (Domain and Type Enforcement : Modèle de protection associant

des domaines aux sujets et des types aux objets). Plutôt que de considérer une problématique particulière, celui-ci fournit un mécanisme générique, et autorise la spécification de politiques adaptées à tout environnement **[77]**.

Les modèles MAC ont ensuite été implantés dans des systèmes variés dont voici une liste non exhaustive **[75]** :

- Dans la carte à puce Java multi-applications, les contrôles d'accès obligatoires sont utilisés pour réguler les flux d'informations entre les différents applets Java.
- Le module noyau Security **Enhanced Linux** permet d'activer les contrôles d'accès obligatoires dans les systèmes d'exploitation Linux et Android (depuis la version 4.3).
- Windows Vista et Windows 7 implantent des contrôles d'accès basés sur le modèle de **Biba** afin, en particulier, d'éviter qu'un processus utilisateur ne corrompe un objet système.
- Il existe des versions multi-niveaux de certains SGBD (par exemple Oracle Label Security).

3.2.1. Principe de la politique MAC :

Dans les modèles **MAC**, un niveau de sécurité est affecté à chaque sujet et à chaque objet. Le niveau de sécurité associé à un objet s'appelle le niveau de classification alors que le niveau de sécurité associé à un sujet s'appelle le niveau d'habilitation. La politique de sécurité est obligatoire c'est-à-dire qu'elle s'impose à tous les utilisateurs et ne peut être modifiée. Si l'objectif est de garantir la confidentialité des données alors la politique de sécurité obligatoire (que l'on appelle aussi la politique de sécurité **multi-niveaux**) est la suivante : « les utilisateurs ont l'interdiction de prendre connaissance des données ayant un niveau de classification supérieur à leur niveau d'habilitation mais ont la permission de prendre connaissance des données classifiées à un niveau égal ou inférieur à leur niveau d'habilitation » (***Figure 4.4***) **[75]**.

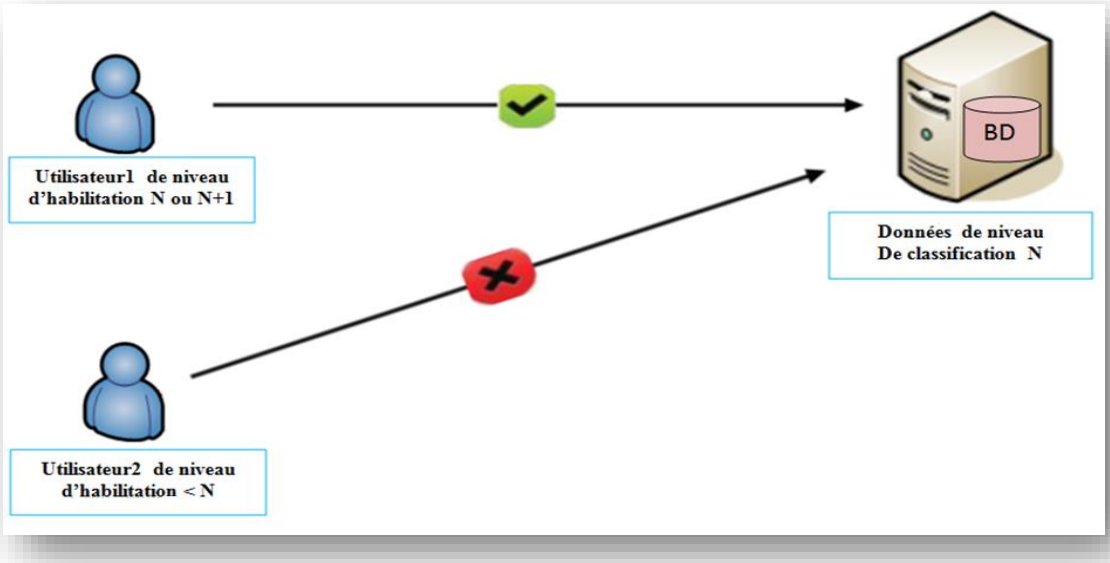

Figure 4.4. *Un exemple de modèle MAC* **[68].**

Bell&LaPadula ont montré qu'il était nécessaire d'appliquer les deux propriétés de contrôle d'accès suivantes pour garantir la politique de sécurité multi-niveaux **[75]** :

- ***No read up*** : cette propriété stipule qu'un sujet habilité à un certain niveau de confidentialité ne peut pas **lire** un objet classifié à un niveau supérieur.

- ***No write down*** : cette propriété stipule qu'un sujet habilité à un certain niveau de confidentialité ne peut pas **écrire** dans un objet classifié à un niveau inférieur.

Ces deux propriétés de contrôle d'accès ne sont toutefois pas suffisantes pour garantir la politique de sécurité multi-niveaux. Les modèles **MAC** s'inscrivent en fait dans la catégorie des modèles de contrôle de flux puisque le seul moyen de garantir totalement la sécurité multi-niveaux est de contrôler tous les flux d'informations possibles. L'information peut en effet transiter illégitimement par des canaux différents des simples opérations de lecture/écriture **[75]**.

3.2.2. Points faibles de MAC :

En revanche, l'utilisation des modèles **MAC** est complexe. Soit ils fournissent une politique trop restreinte, trop peu générale (BLP, BIBA, Clark-Wilson), et sont alors difficile à déployer en pratique. Soit ils fournissent des mécanismes génériques **DTE**, mais alors le travail à fournir pour définir la politique de sécurité est bien plus exigeant, et n'inclut pas de garantie contre les erreurs de l'administrateur qui la définit **[80]**. La politique **MAC** est quelques fois difficile à administrer, trop rigide, ce que limiter la diffusion de l'information (à l'inverse de **DAC**). Elle est très appropriée pour des systèmes de haute sécurité **[78]**.

3.3. CONTROLE D'ACCES A BASE DE ROLES RBAC :

Le coût de maintenance du contrôle d'accès peut rapidement devenir prohibitif dans les modèles précédemment cités. En particulier dans les systèmes industriels où l'accès aux objets peut se faire par des centaines ou milliers de sujets. La gestion de ces autorisations entraîne une explosion combinatoire qui ralentit l'évaluation du contrôle d'accès et rend complexe l'attribution ou la révocation de droits. De plus, ni la rigidité des modèles **MAC**, ni le manque de contrôle des modèles **DAC** ne sont réellement satisfaisants pour des systèmes industriels où les autorisations doivent pouvoir être déléguées, mais aussi contrôlées. C'est en réponse à ces problématiques qu'est apparu le contrôle d'accès à base de rôles, appelé **Role-Based Access Control (RBAC) [76]**.

3.3.1. Principe de la politique RBAC :

Le modèle de contrôle d'accès à base de rôle **RBAC** a été proposé pour présenter une nouvelle organisation des droits centrée sur le concept de rôle. Un rôle représente une fonction dans le cadre d'une organisation. Utiliser le rôle comme intermédiaire entre les sujets et les permissions facilite et simplifie les tâches d'administration en diminuant le nombre d'affectations à manipuler **[69]**.

La Figure 4.5 illustre la mécanique **RBAC** : les sujets obtiennent des autorisations sur des ressources grâce à des rôles qui leur sont attribués, eux-mêmes associés à un ensemble de permissions. Comme il y a normalement nettement moins de rôles que d'utilisateurs et de ressources, cela simplifie grandement la gestion des autorisations **[76]**.

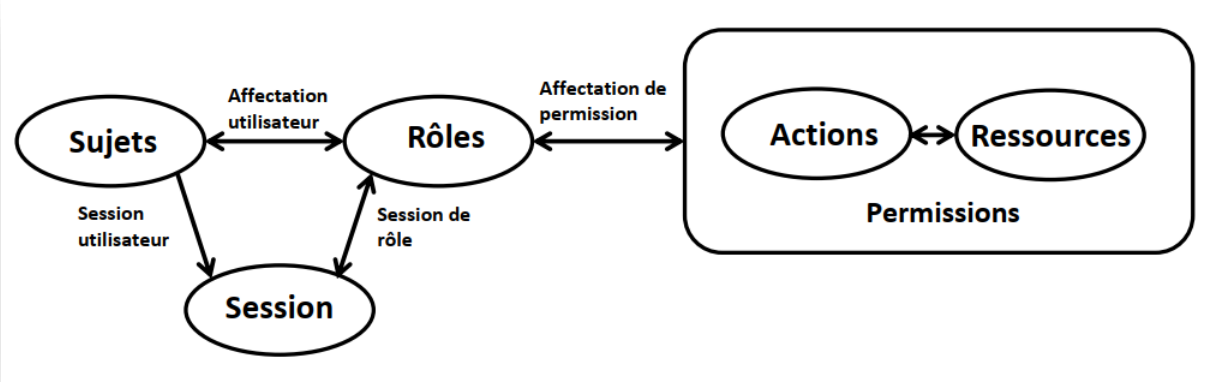

Figure 4.5. *Modèle RBAC* **[76].**

Les principes de base du modèle **RBAC** sont les suivants **[75]** :

- Alors que dans les modèles **DAC**, les permissions ont trait à des opérations de bas niveau telles que les opérations de lecture/écriture, dans les modèles **RBAC** elles

concernent des tâches de nature organisationnelle telles que « transférer de l'argent », « acheter un billet d'avion » etc.

- Dans les modèles **RBAC**, le concept de rôle correspond à une fonction professionnelle. Les permissions sont accordées à des rôles et non pas à des utilisateurs. Les rôles sont ensuite distribués aux utilisateurs en fonction de leurs responsabilités au sein de l'organisation. Une même permission peut être affectée à différents rôles et différents rôles peuvent être attribués à un même utilisateur.
- Les modèles **RBAC** offrent une solution pour implanter des mesures de type **séparation des tâches**. Le principe de la séparation des tâches prévoit qu'un même utilisateur ne peut effectuer des tâches qui pourraient être orchestrées pour mettre œuvre des opérations frauduleuses, comme par exemple « autoriser un paiement » et « effectuer un paiement ». Ce principe peut aisément être garanti avec les modèles **RBAC** dans la mesure où deux rôles peuvent être déclarés comme étant mutuellement exclusifs. Deux rôles mutuellement exclusifs ne peuvent alors être affectés à un même utilisateur.

Le **RBAC** est largement adopté par les entreprises et les industriels et a été appliqué dans de grandes structures. Les logiciels commerciaux Trusted Solaris, Windows Authorization Manager, Oracle 9, Sybase, Adaptive Server Microsoft Active Directory, la plupart des SGBD commerciaux, FreeBSD et Wikipedia ont mis en œuvre tout ou partie des principes des modèles à base de rôle **[71]**.

3.3.2. Les sous-modèles (famille) de RBAC :

La spécification de modèle RBAC comprend les sous-modèles suivants (***Figure 4.6***) **[69]** :

- Le modèle **RBAC0** ou « the flat model », qui présente les concepts et relations de base c.à.d. le noyau du modèle.
- Le modèle **RBAC1** ou « the hierarchical model », qui reprend le modèle RBAC0 et introduit la notion de hiérarchie entre rôles.
- Le modèle **RBAC2** ou « the constrained model », qui reprend le modèle RBAC0 et introduit la notion de contrainte.
- Le modèle **RBAC3** ou « the symmetric model », qui reprend les modèles RBAC1 et RBAC2 et prend en compte les interactions entre contraintes et hiérarchie.

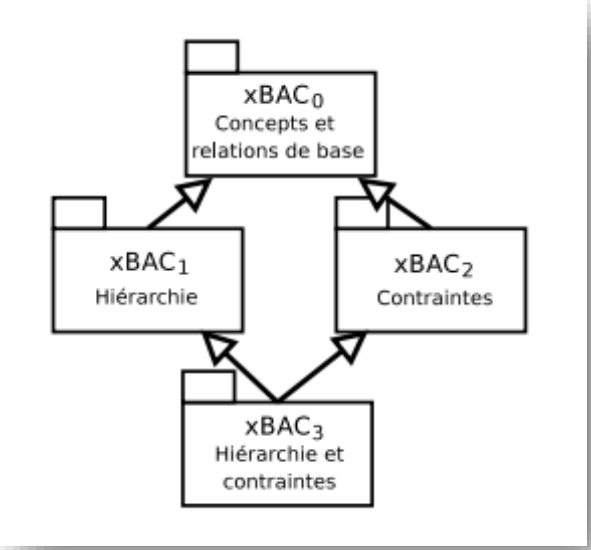

Figure 4.6. *Famille x-BAC (UML)* **[71].**

Ces raffinements successifs illustrent une orientation générale des recherches sur les modèles de contrôle d'accès : à partir d'un noyau, introduisant les concepts et relations principales du modèle, des enrichissements supplémentaires sont proposés. Cette structuration de la famille des modèles **RBAC** a été reprise par exemple dans les modèles **TBAC** (Task Based Access Control) et **GEO-RBAC (**Geospatial aware role based access control), c'est la raison pour laquelle le terme ***x*-BAC** est utilisé dans la figure précédente **[71]**.

3.3.3. Modèles de contrôle d'accès dérivés de RBAC :

Le modèle **RBAC** a été largement adopté par l'industrie et par la communauté de recherche. Il a déclenché un renouveau des modèles de contrôle d'accès et plusieurs propositions ont été faites pour ajouter de nouveaux concepts ou notions au modèle **RBAC** de base : par exemple le temps, la localisation, le contexte spatial, la position géographique de l'utilisateur etc. Nous classons dans cette section les modèles qui couvrent la plupart de ces nouvelles notions ou concepts dérivés de **RBAC [69]** :

- **La notion d'équipes introduite par le modèle TMAC :**

  La notion d'équipe a été proposée dans le modèle **TMAC** (TeaMbased Access Control). Les permissions sont associées aux rôles ainsi qu'aux équipes. La notion d'équipe a été introduite pour représenter des aspects transversaux des rôles qui ne sont pas directement exprimables dans les modèles **RBAC**. Dans **TMAC**, l'objectif est d'accorder à chaque utilisateur membre d'une équipe des permissions accordées aux autres membres de l'équipe qui sont actifs.

- La notion de localisation et d'information spatiale dans des applications mobiles introduites par le modèle **LRBAC (**Location-aware role-based access control) **:**

  LRBAC étend le modèle **RBAC** pour que le contrôle d'accès puisse être établi en prenant en compte les informations de localisation. Un tel modèle a été proposé pour autoriser ou interdire l'accès lorsque les systèmes sont dans ou hors d'une zone d'opération définie. Ce modèle, proposé dans le cadre de la prolifération d'équipements mobiles, utilise la localisation logique des utilisateurs et/ou des systèmes comme paramètre contextuel.
- La notion d'information spatiale et plus particulièrement la position physique de l'utilisateur dans des dispositifs comme **GPS** introduit par le modèle **Geo-RBAC (**Geospatial aware role based access control) **:**

  Le modèle **Geo-RBAC**, étend le modèle **RBAC** en définissant de nouveaux concepts spatiaux pour représenter la position des sujets et celles des objets. Ces nouveaux concepts sont utilisés pour limiter géographiquement l'utilisation des rôles. Le principe proposé dans **Geo-RBAC** est de comparer une position physique, supposée obtenue de façon fiable (par exemple la localisation GPS), à des positions logiques (exemples : route, ville, région) auxquelles sont associées des rôles géographiques
- La notion de temps et des contraintes de temps dans les systèmes de **« Workflow »** introduite par le modèle **GTRBAC (**Generalized temporal role based access control) **:**

  GTRBAC étend le modèle **RBAC** afin d'exprimer un large éventail de contraintes temporelles. En particulier, le modèle permet d'exprimer le temps et des contraintes temporelles sur les rôles, l'affectation des rôles aux utilisateurs, et l'affectation des permissions aux rôles. Ce modèle répond aux besoins précis d'applications avec une contrainte temporelle forte, comme les systèmes intégrant des workflows où la notion de temps est importante. Ces systèmes sont utilisés par des organisations désirant spécifier des règles d'autorisation qui permettent ou interdisent l'accès à des ressources pendant un intervalle de temps donné.

3.4. MODELES DE CONTROLE D'ACCES A BASE DES TACHES :

En parallèle des travaux originaux sur **RBAC**, le modèle **TBAC** (TaskBased Access Control) a été conçu afin d'activer une permission par rapport aux tâches effectuées par l'utilisateur. L'idée essentielle de ce modèle consiste à ajouter la notion de tâche dans des règles d'autorisation. Cela permet de définir les permissions qu'un sujet peut activer selon la tâche qui est en cours. Chaque étape d'autorisation correspond à certaines activités ou tâches dans le contexte plus large d'un workflow de l'organisation. Le modèle **TBAC** fut le premier modèle à introduire le concept de tâche. **TBAC** va au-delà des modèles **IBAC** (Identity Based Access Control) où les actions correspondent généralement à des commandes élémentaires (comme la lecture du contenu d'un objet ou l'écriture dans un objet) pour structurer et

contrôler la réalisation d'actions composites, appelées tâches ou activités. Il ajoute une étape d'autorisation (authorization step) qui permet de définir les permissions (enabled permissions) qu'un sujet (executor-trustee) peut activer selon la tâche qui est en cours. Ainsi, **TBAC** offre une approche pour différencier l'affectation et l'activation des permissions par rapport à des tâches données aux utilisateurs au sein de l'organisation. La notion de tâche permet de contrôler les activités exercées par les utilisateurs d'un système au sein de l'organisation **[69]**.

La figure suivante montre le cycle de vie d'une autorisation **TBAC** :

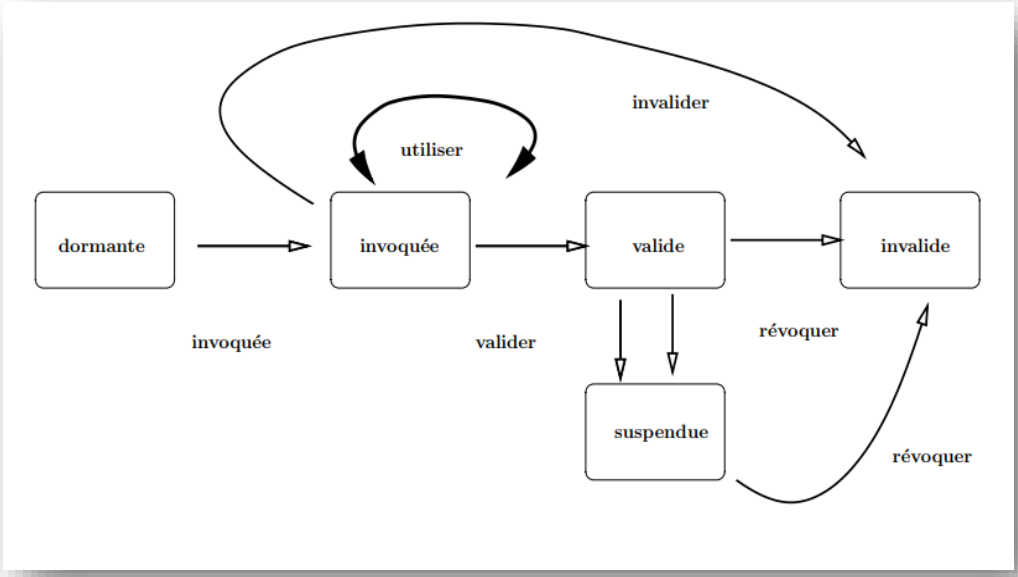

***Figure 4.7**. Cycle de vie d'une autorisation dans le modèle TBAC* **[79].**

TBAC peut parfaitement être adapté et intégrer la notion de rôle. C'est dans cet esprit que le modèle **TR-BAC** (Task and Rôle Based Access Control) a été défini. Dans ce cas, les droits sont activés en fonction d'un rôle et portent sur la réalisation des tâches. **TBAC** ou **TR-BAC** présente l'inconvénient de ne pas prendre en compte des contraintes sur les horaires ou périodes d'accès pendant lesquels les utilisateurs sont en charge de la réalisation de leurs activités. Le manque constaté est couvert par d'autres modèles formels de contrôle d'accès **[69]**.

3.5. MODELE DE CONTROLE D'ACCES A BASE D'ORGANISATION OR-BAC :

Le modèle de contrôle d'accès **Or-BAC** (Organization Based Access Control) vise à résoudre certains problèmes rencontrés par les premiers modèles de contrôle d'accès des années 90 et à établir une politique de contrôle d'accès plus abstraite. Il s'intéresse, non seulement aux permissions, mais aussi aux interdictions, obligations et recommandations dans une politique de sécurité. **Or-BAC** prend le concept de rôle dans **RBAC**. En plus de ce

concept, il ajoute des nouveaux concepts pour structurer les sujets, les objets et les actions **[69]**.

3.5.1. L'organisation :

Le concept central de ce modèle est la notion d'organisation comme son nom l'indique. Une organisation peut être un groupe structuré de sujets jouant des rôles déterminés. Ce peut être un hôpital, une clinique médicale, un service d'urgence…etc. L'organisation représente l'ensemble des rôles, des activités, et des vues qui représentent les abstractions respectives des utilisateurs, des opérations et des objets par rapport à une organisation donnée. Par exemple, un utilisateur est lié à un ensemble de rôles pour une organisation donnée. Il peut être affecté à d'autres rôles pour une autre organisation.

Le fait d'introduire ce concept « organisation » comme un élément de base dans le modèle de contrôle d'accès permet de structurer les droits en rassemblant plusieurs notions comme le rôle de **RBAC** et l'équipe de **TMAC**. Ces derniers définissent des relations binaires entre l'utilisateur et le rôle dans **RBAC**, ou entre l'utilisateur et l'équipe dans **TMAC**. **Or-BAC** définit des relations ternaires entre les organisations, les sujets et les rôles **[69]**.

3.5.2. Les sujets et les rôles :

L'entité **Sujet** est utilisée différemment selon les modèles de sécurité. Dans le modèle **OR-BAC**, un sujet peut être soit une entité active, c'est-à-dire un utilisateur, soit une organisation. Un sujet joue un rôle dans une organisation. Ce qui veut dire que l'utilisateur ayant plusieurs rôles peut activer soit tous les rôles soit un sous-ensemble de ses rôles, dans n'importe quelle équipe à laquelle il participe. Dans la pratique, même si un utilisateur possède plusieurs rôles, il n'a pas forcément le droit de les jouer dans toutes les équipes auxquelles il appartient **[70]**.

3.5.3. Les objets et les vues :

Dans notre modèle, l'entité **Objet** représente principalement les entités non actives comme les fichiers, les courriers électroniques, les formulaires imprimés, etc. Nous l'appelons : entité **Vue**. De manière intuitive, une vue correspond, comme dans les bases de données relationnelles, à un ensemble d'objets qui satisfait une propriété commune. Par exemple dans un système de fichier administratif, la vue « dossiers administratifs » correspond à l'ensemble des dossiers administratifs des patients, alors que la vue « dossiers médicaux » correspond aux dossiers médicaux des patients **[70]**.

3.5.4. Les actions et les activités :

Les politiques de sécurité spécifient les accès autorisés aux entités passives par des entités actives et régulent les actions opérées sur le système. Dans notre modèle, l'entité *Action* englobe principalement les actions informatiques comme lire, écrire, envoyer, etc. Le schéma dans la **Figure 4.8** fait apparaitre les deux niveaux de politique **Or-BAC** (abstrait et concret) ainsi que les différentes relations existant entre les entités de ces deux niveaux[70].

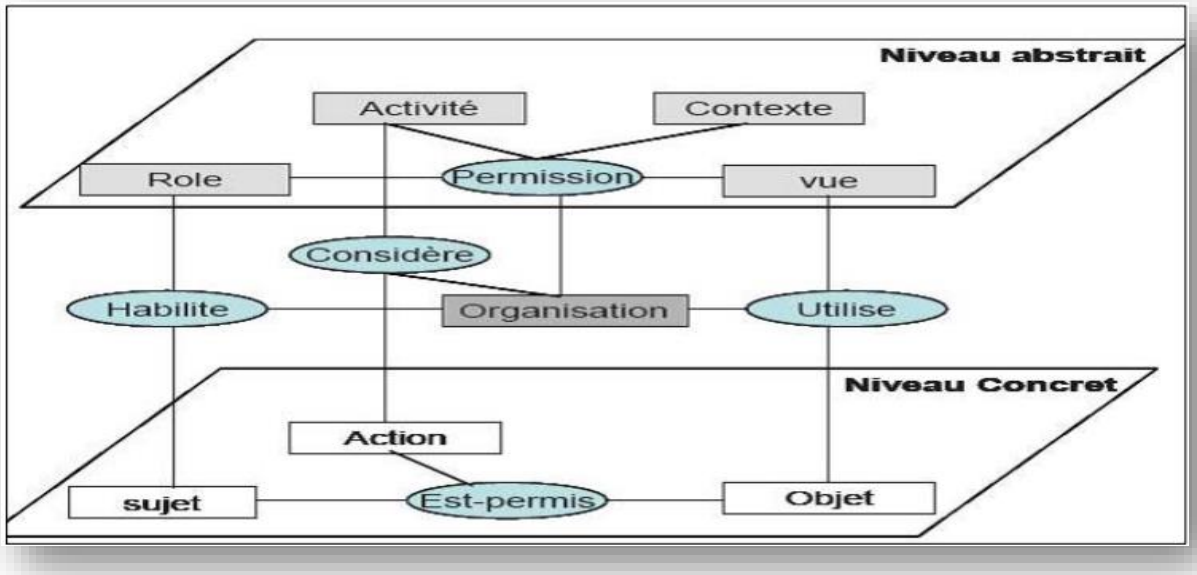

Figure 4.8. *Le modèle Or-BAC* **[69].**

3.6. MODELES DE CONTROLE D'ACCES A BASE DE CONTEXTE (CBAC) :

Dans un nombre d'applications de plus en plus grand, la politique de sécurité ne peut plus être définie au moyen de règles d'autorisation statiques. Dans de telles applications, les privilèges accordés aux utilisateurs dépendent de conditions contextuelles. Les modèles de contrôle d'accès qui permettent l'expression de règles dynamiques où la distribution des autorisations dépend de conditions contextuelles appartiennent à la famille des modèles **CBAC (**Context Based Access Control**) [79]**.

La définition du contexte en informatique est délicate, une définition communément admise est proposée par Dey: « *le contexte est l'ensemble de toutes les informations qui peuvent être utilisées pour caractériser la situation d'une entité. Une entité pouvant être un acteur, un lieu, ou un objet de l'environnement considéré comme utile à l'interaction entre un utilisateur et une application, y compris l'utilisateur et l'application eux-mêmes* » **[73]**.

Les modèles suivants peuvent être considérés comme étant des modèles **CBAC [73]** :

- Dans le modèle **ABAC** (Attribute Based Access Control), les autorisations dépendent de conditions booléennes s'appliquant aux attributs du sujet, de l'objet et de l'environnement.
- Certaines propositions étendent le modèle **RBAC** pour prendre en compte des conditions contextuelles telles que la position de l'utilisateur ou le temps. Dans la plupart de ces approches les rôles peuvent être activés en fonction de conditions spatiales ou temporelles.
- Le modèle formel de contrôle d'accès **CRBAC (**Context role based access control) introduit en plus du « rôle » la notion de « contexte ». Les rôles sont composés : rôles de sujets comme dans **RBAC**, et rôles contextuels pour capturer des informations de sécurité liées au contexte **[81]**.
- Le modèle **OrBAC**, définit une taxonomie complète de contextes (spatial, temporel, provisionnel etc.) et fournit un cadre formel fondé sur la logique du premier ordre pour exprimer des règles contextuelles (***Figure 4.9***). Il intègre le modèle d'administration **AdOrBAC (**Administration model for Or-BAC).

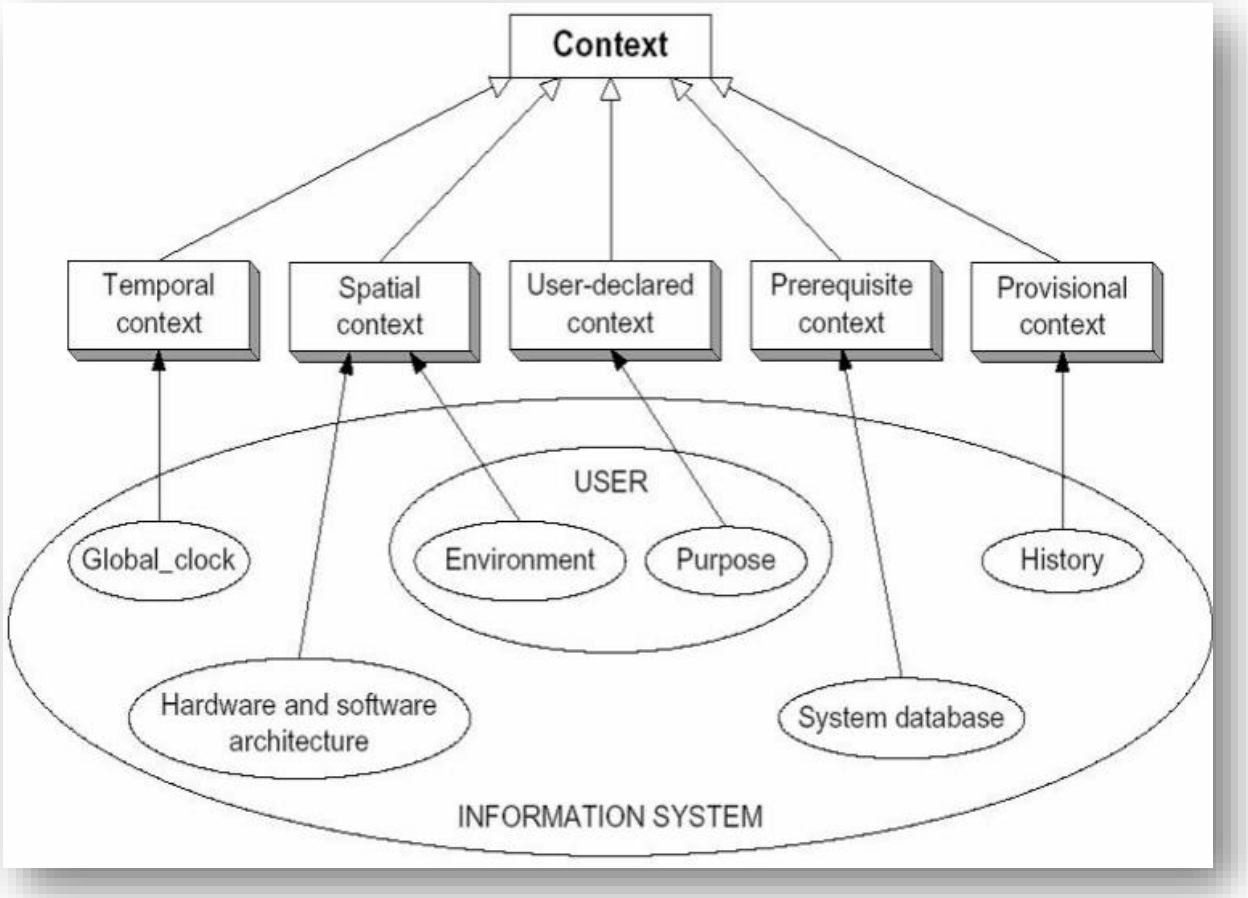

Figure 4.9. *Contexte dans OR-BAC* **[69].**

Une caractéristique importante des modèle **CBAC** réside dans le fait qu'ils permettent d'exprimer des règles d'autorisation qui ne requièrent pas d'authentifier les utilisateurs. Un utilisateur peut en effet obtenir un accès à une information simplement parce que certaines conditions contextuelles sont remplies. Cette capacité à accorder ou refuser un accès sans avoir à authentifier l'utilisateur est très utile dans le cadre d'applications Web interconnectées **[75]**.

4. CONCLUSION :

Dans ce chapitre nous avons présenté plusieurs modèles de contrôle d'accès existants qui ont été conçus pour assurer la sécurité des données dans des contraintes différentes. En commençant par les modèles de contrôle d'accès classiques (**DAC**, **MAC**) qui ont été proposés pour répondre aux exigences des systèmes d'exploitation. D'autres modèles sont bien adaptés aux structures très organisées, où la confidentialité est une priorité (les modèles à niveau dits aussi « orienté flux » comme **TBAC** et **WRBAC** : Worflow Role Based Access Controle). D'un autre côté, certaines structures ont des organisations beaucoup plus souples et complexes. Elles souhaitent alors définir leurs propres modèles de contrôle d'accès adaptés à leurs besoins. Ils réutilisent les concepts présentés précédemment (rôle, organisation, hiérarchies) pour créer leurs propres politiques de contrôle d'accès. D'autres modèles ont préféré compléter le concept de rôle dans des modèles **RBAC** par une ou plusieurs notions pour s'adapter à d'autres situations, comme le contexte dans **Or-BAC** ou **CRBAC.**

Après l'étude que nous avons faite sur les différents politiques de control d'accès existantes nous avant remarqué que :

- Les propositions basées sur les rôles forment la plus grande famille des modèles de contrôle d'accès et sont les plus étudiées et utilisés dans le monde de sécurité informatique.
- Il est difficile de considérer qu'un modèle de contrôle d'accès est meilleur qu'un autre : cela dépend essentiellement du domaine d'application et du type de l'organisation qui le met en œuvre

Chapitre 05 : Réalisation

1. INTRODUCTION :

Dans cette partie, nous allons présenter notre application qui consiste en la réalisation d'un contrôleur d'accès basé sur les rôles pour SQL SERVER en commençant par l'environnement de travail, les outils utilisés et enfin les différentes fonctionnalités de l'application.

2. ENVIRONNEMENT DE TRAVAIL :

- **SYSTEME D'EXPLOITATION UTILISE :**

On a utilisé une machine (i5/10GO/1TO) avec un système d'exploitation Windows 8 professionnel.

- **LE SERVEUR SGBD UTILISE :**

Notre application est conçue pour gérer les droits d'accès aux bases de données sous SQL SERVEUR, par conséquent le Serveur SGBD utilisé c'est SQL Server 2017.

3. L'OUTIL DE DEVELOPPEMENT UTILISE :

Pour le développement de notre application, on a choisi d'utiliser ***Delphi 2007 for Microsoft Windows- Entreprise avec Transact-SQL***

3.1. POURQUOI DELPHI 2007 FOR MICROSOFT WINDOWS- ENTREPRISE [82] :

CodeGear, un leader dans les outils de développement, donne aux développeurs la possibilité de créer rapidement et facilement des applications clientes riches qui vont pouvoir profiter des fonctionnalités des interfaces utilisateur Microsoft Windows et Vista Aero et de créer des applications web dynamiques utilisant AJAX. Delphi for Win32 inclut également une architecture de base de données de DBX4 qui supporte les versions des RDBMSs les plus populaires aujourd'hui comme Microsoft SQL Server, InterBase de CodeGear, MySQL, et Oracle, élargissant ainsi cette plateforme de développement déjà très populaire.

« Fait pour les ISVs, intégrateurs système, VARs et des petites à moyennes entreprises, Delphi for Win32 permet de développer des applications natives Microsoft Windows très performantes sur Microsoft Windows ou Vista qui non seulement supportent les deux plateformes, mais prennent en compte toutes les fonctionnalités de Vista, » (Michael Swindell).

« Le support de Microsoft et Vista aide les développeurs à se lancer sur les nouveau système

d'exploitation de Microsoft tout en continuant à proposer un support à leurs utilisateurs utilisant Windows. La VCL pour le Web permet aux développeurs de créer rapidement et visuellement des applications Web dynamiques avec des interfaces utilisant AJAX » (Swindell).

Voici quelques-unes des fonctionnalités de cette version :

- Support de Microsoft Windows Vista et AJAX.
- Développement sur toutes les plates forme Microsoft Windows et Vista.
- Support de Microsoft MSBuild.
- Nouvelle architecture Database DBX 4.
- Support des dernières versions d'InterBase, Microsoft SQL Server, MySQL, Oracle Et autres SGBD.
- Support des thèmes pour les applications.
- Support des effets Microsoft Windows Vista Aero glassing, composants spécifiques pour les boites de dialogue Vista.
- VCL pour le Web avec support d'AJAX.
- Compatibilité descendante pour les composants développés avec BDS 2006.

« Delphi for Win32 contient des centaines d'améliorations de qualité qui rendent Delphi et les applications créées avec Delphi plus fiables et plus robustes que jamais. Il inclut également les dernières versions de produits tiers populaires tels que TeeChart, Indy, et Rave Reports, » (Nick Hodges).

La figure suivante représente l'IDE Delphi 2007 :

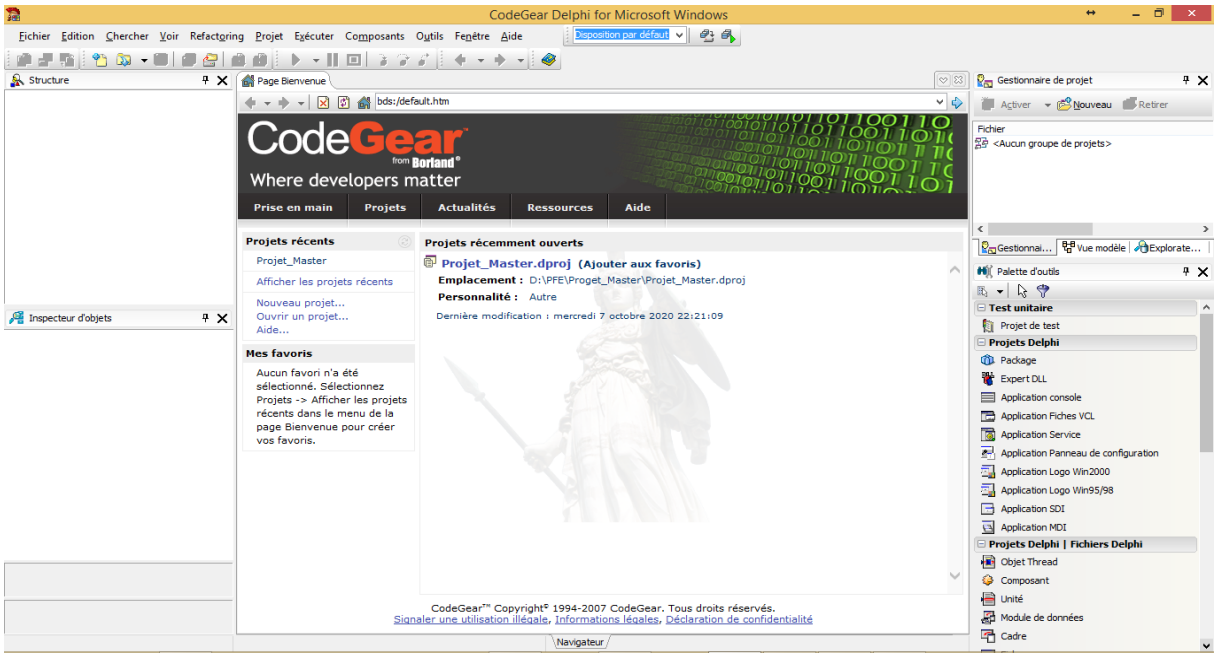

Figure 5.1. *L'environnement IDE de Delphi 2007.*

L'éditeur dans son intégralité se présente en un seul bloc, divisé en sous-blocs contenant des informations.

Au premier démarrage de Delphi 2007, la page de bienvenue est affichée sous forme de page Web locale, affichant plusieurs informations utiles telles qu'une aide pour la prise en main, ou encore les actualités courantes concernant Delphi.

- **Transact -SQL** :

Microsoft Transact SQL ou T-SQL est un langage de requêtes amélioré par rapport au SQL dont il reprend les bases. Le SQL (Structured Query Language) est le langage standard, créé par IBM dans les années 70, pour la gestion des SGBDR (Systèmes de Gestion de Bases de Données Relationnelles).

De plus, le Transact SQL prend en compte des fonctionnalités procédurales telles que la gestion des variables, les structures de contrôle de flux, les curseurs, et les lots d'instructions. C'est donc un langage complet qui comporte des instructions, qui manipule des objets SQL, qui admet la programmabilité et qui utilise des expressions.

3.2. DEMARCHE DE DEVELOPPEMENT DE NOTRE APPLICATION :

Etape 01 : Création d'un nouveau projet Delphi

- Lancement Delphi 2007.
  Afin de visualiser l'environnement, j'ai créé un projet nommé « Projet Master»:
  « Application Fiches VCL »

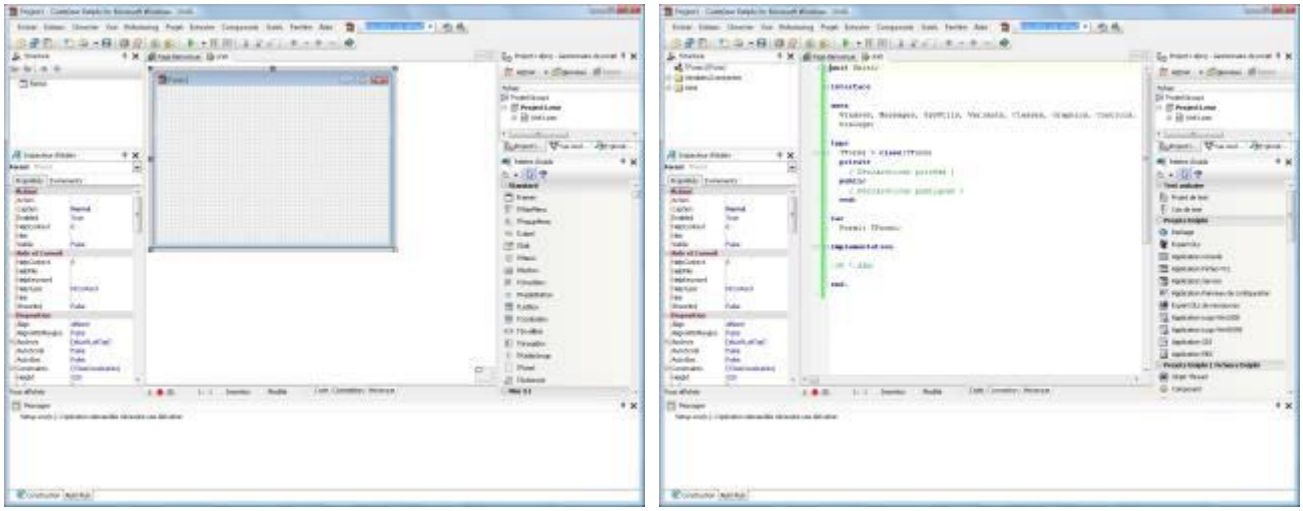

Figure 5.2. *Création de projet et choix de type d'application.*

Etape 02 : Connexion à SQL SERVEUR

Après la création de projet, on doit le connecter à SQL SERVEUR en respectant les étapes suivantes :

- Dans la fenêtre principale ajouter une TADOConnexion 

- Double cliquer dessus et choisir "Utiliser la chaine e connexion " puis cliquer sur construire comme ce ceci :

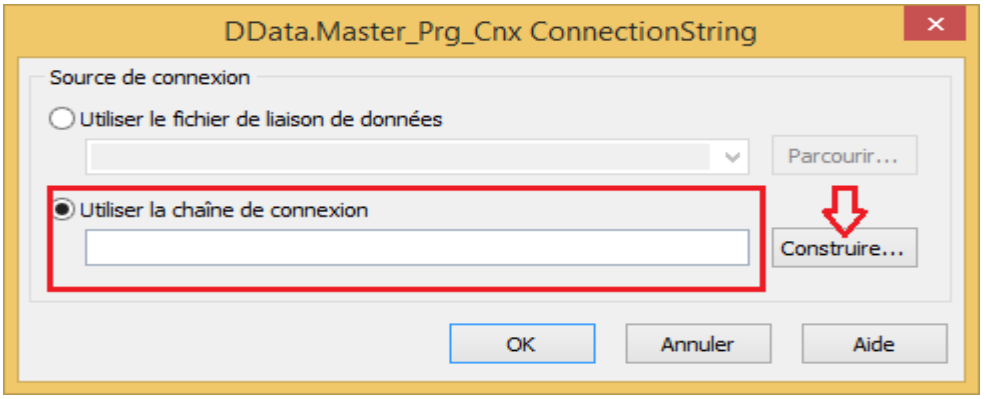

Figure 5.3. *Définir la chaine de connexion au Serveur.*

- La fenêtre suivante s'ouvre pour choisir le type de base de données avec lequel on va travailler, dans notre cas choisir Microsoft OLE DB Provider for SQL SERVER puis cliquer sur suivant comme ceci :

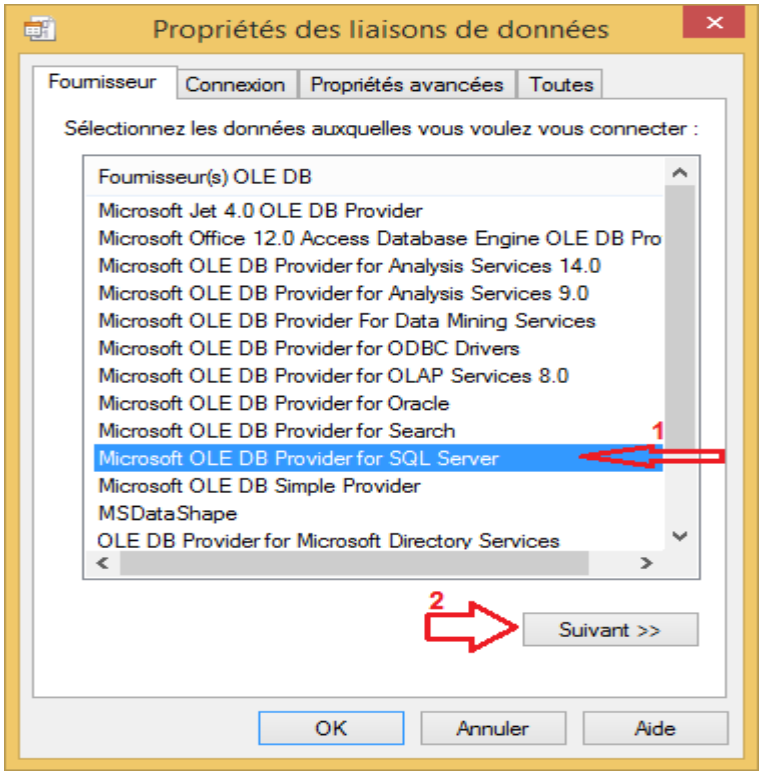

Figure 5.4. *Choisir le type de base de données.*

- La fenêtre suivante apparait pour renseigner les champs indiqués avec les valeurs adéquate, dans notre cas c'est :
  - ✓ Nom du serveur.

    HP-FELLA\SQLEXPRESS
  - ✓ Nom d'utilisateur et mot de passe (session administrateur de SQL Serveur Management Studio).

    Nom d'utilisateur : SA, mot de passe : Dylog2019
  - ✓ Nom de la base de données par défaut, peut été laissé à blanc

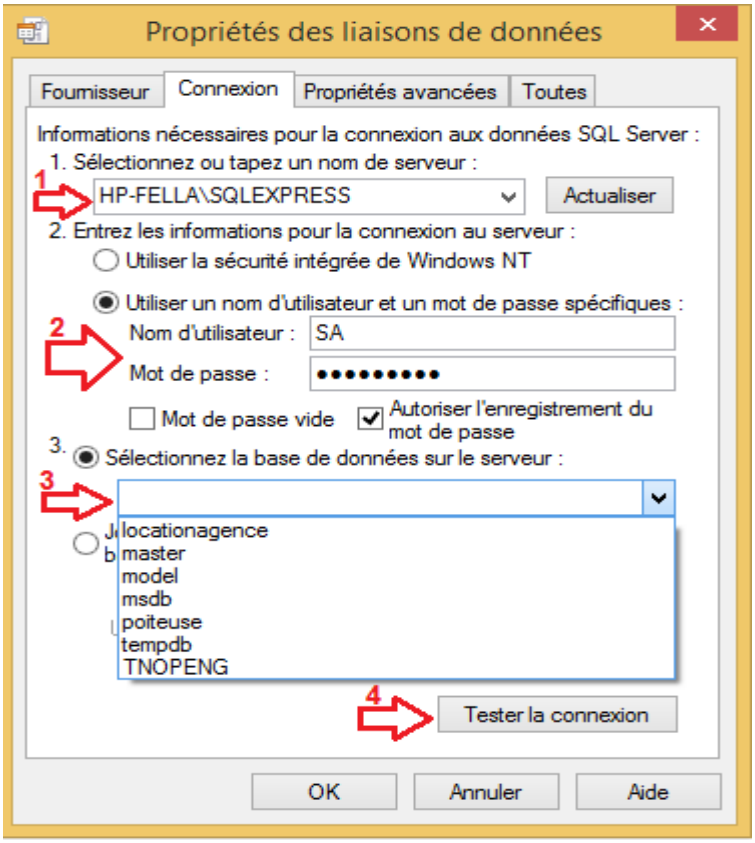

Figure 5.5. *Propriétés de liaison de données.*

Et enfin tester que la connexion a bien été établie.

- Tester que la connexion est activée comme ceci :

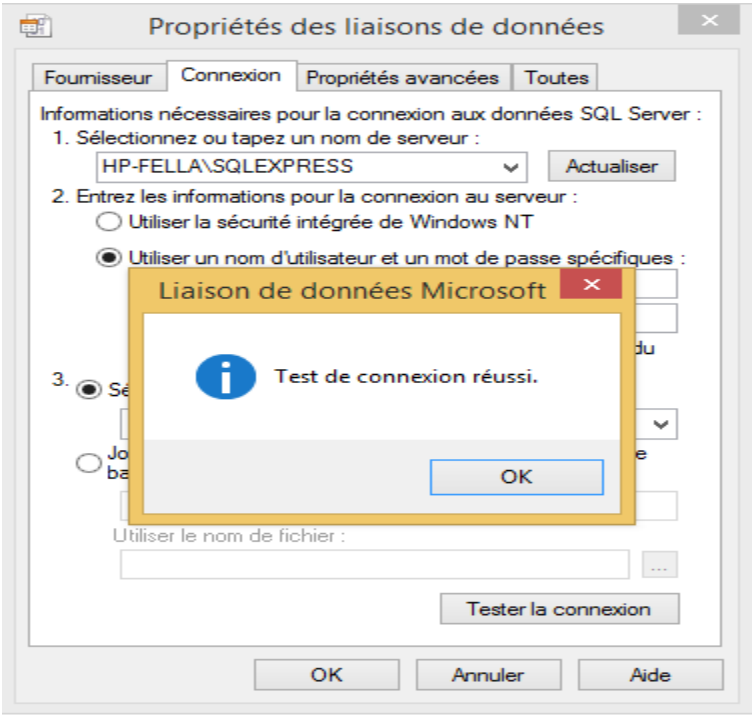

Figure 5.6. *Test de connexion de liaison.*

4. INTERFACE GRAPHIQUE ET LE TEST DU FONCTIONNEMENT DE L'APPLICATION RBAC SQL SERVEUR :

Dans cette partie, nous expliquerons les principales fonctionnalités de notre application, pour se faire, nous avons pris comme exemple une base de données "agence de location immobilière".

La première étape consiste en l'authentification de l'administrateur de base de données pour avoir accès à l'interface de configuration :

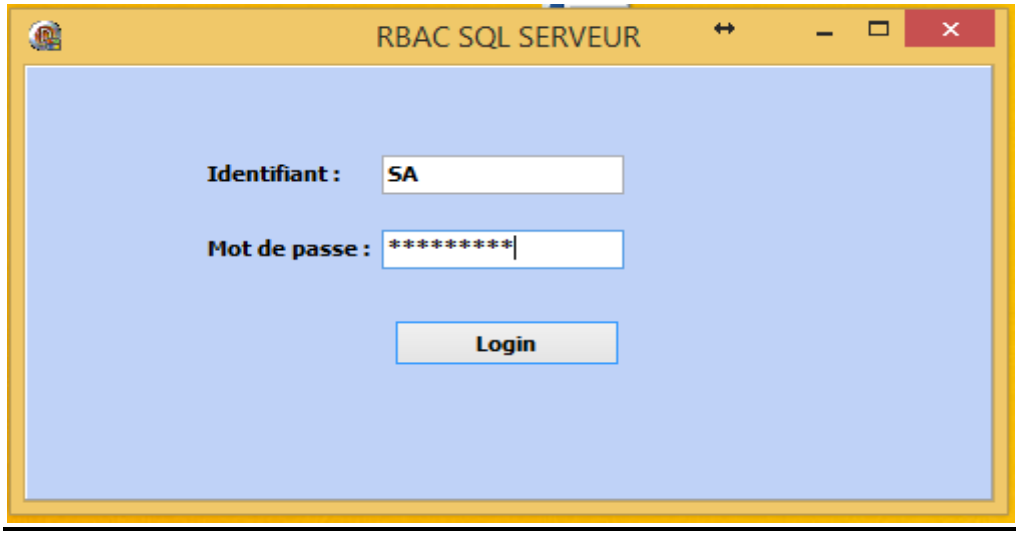

Figure 5.7. *Page d'accueil.*

Bien sûr, si l'identifiant ou bien mot de passe est incorrect, la fenêtre suivante s'affiche :

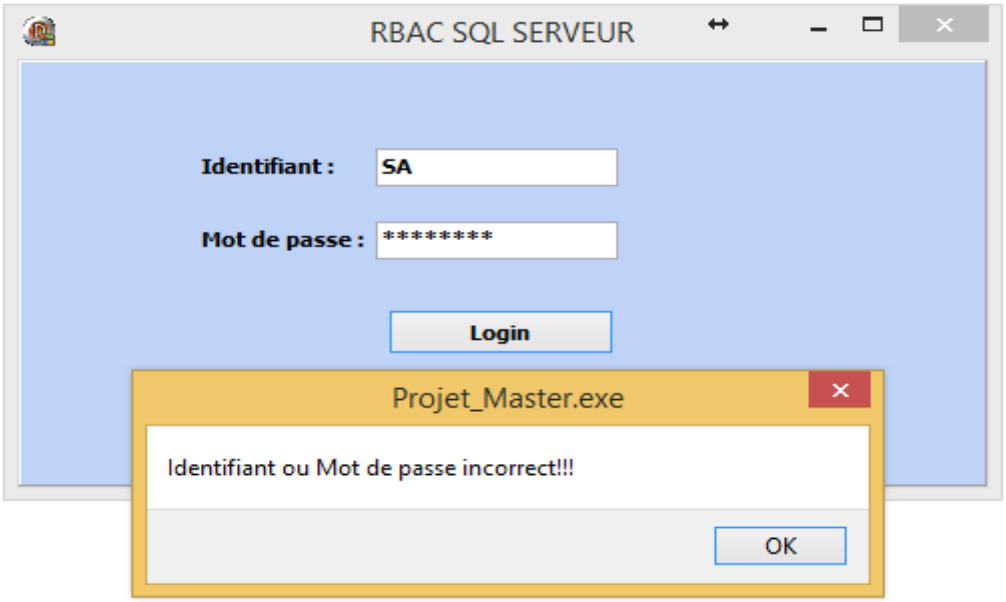

Figure 5.8. *Identifiant ou mot de passe erroné.*

Une fois authentifié, la fenetre suivante s'affiche:

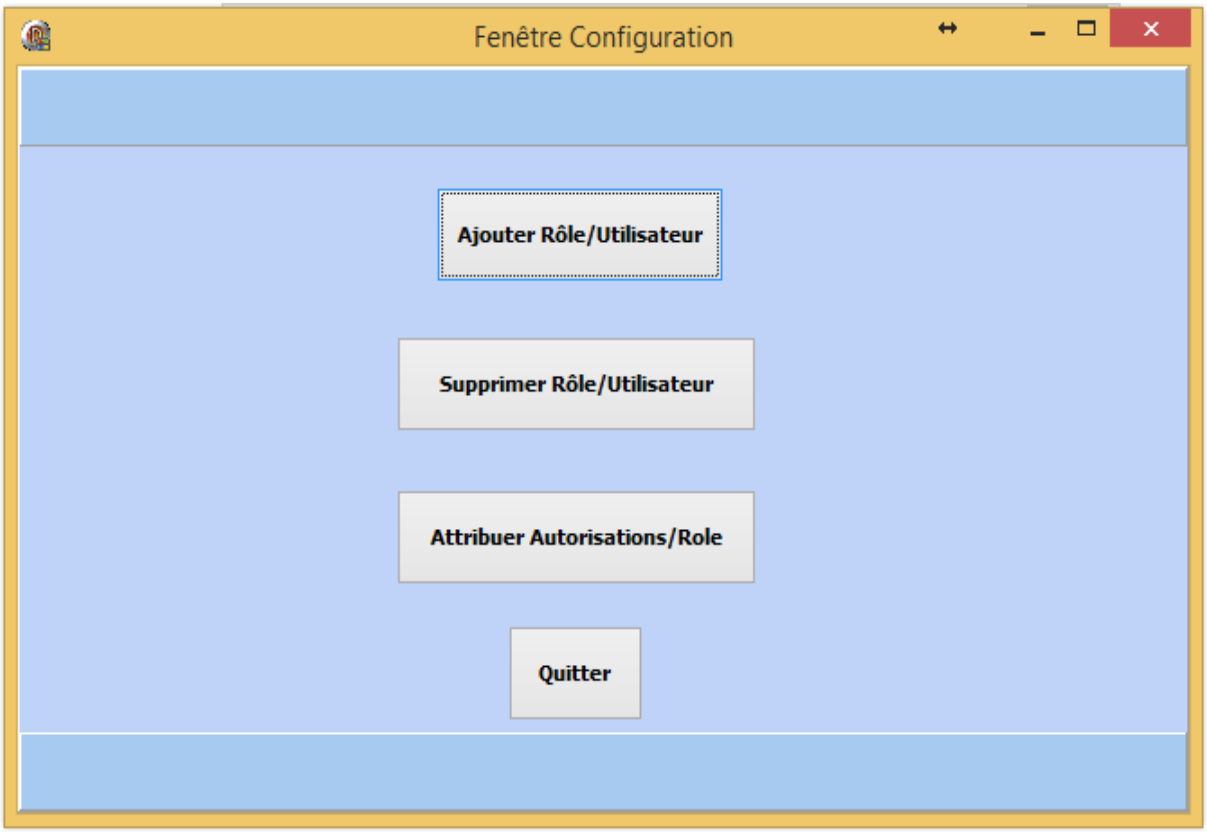

Figure 5.9. *Fenêtre de configuration.*

Ici nous avons les principales fonctionnalités de notre application, à savoir :

-Ajouter un nouveau rôle ou un nouvel utilisateur.

- Supprimer un rôle ou un utilisateur.

-Attribuer les privilèges au rôles.

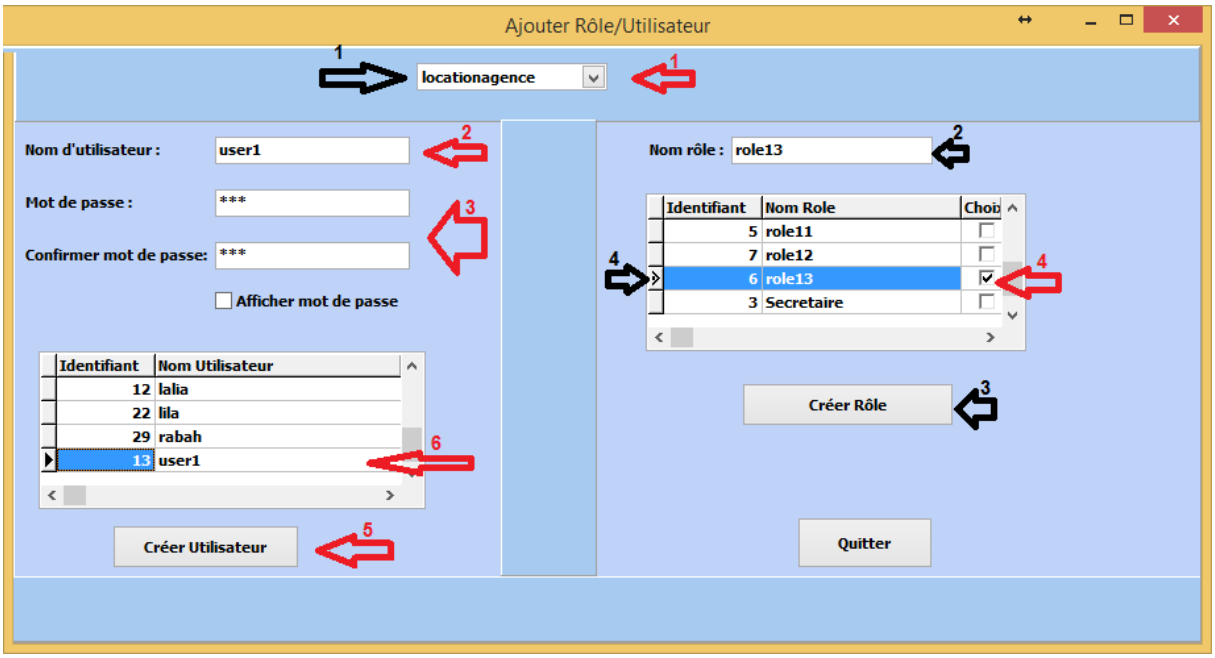

Figure 5.10. *Ajout Rôle/Utilisateur.*

Dans cette fenêtre :

✓ Nous pouvons créer un utilisateur avec mot de passe et un rôle par défaut, tout en sélectionnant la base de données en 1er lieu (indicateurs : flèches rouges).

1- Choisir le nom de la base de données.

2- Entrer le nom de l'utilisateur.

3- Entrer le mot de passe et le confirmer.

4- Sélectionner un ou plusieurs rôles.

5- Cliquer sur le bouton de création.

6- Le nouvel utilisateur figure dans la liste des utilisateurs de base de données.

✓ Nous pouvons créer un rôle, tout en sélectionnant la base de données en 1er lieu (indicateurs : flèches noirs).

1- Choisir le nom de la base de données.

2- Entrer le nom du rôle.

3- Cliquer sur le bouton de création.

4- Le nouvel utilisateur figure dans la liste des utilisateurs de base de données.

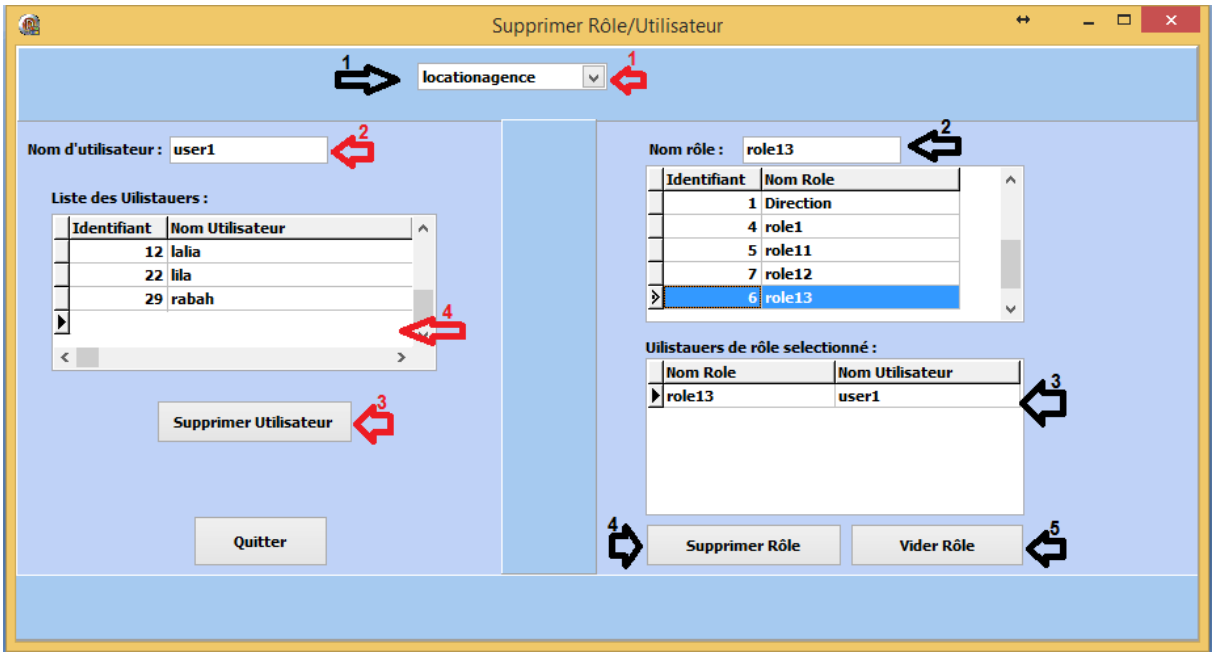

Figure 5.11. *Supprimer Rôle/Utilisateur.*

Dans cette fenêtre :

✓ Nous pouvons supprimer un utilisateur tout en sélectionnant la base de données en 1er lieu (indicateurs : flèches rouges).

1- Choisir le nom de la base de données.

2- Sélectionner l'utilisateur à supprimer.

3- Cliquer sur le bouton de Supprimer Utilisateur.

4- L'utilisateur a bien été supprimer.

✓ Nous pouvons supprimer un rôle, tout en sélectionnant la base de données en 1er lieu (indicateurs : flèches noirs).

1- Choisir le nom de la base de données.

2- Sélectionner le rôle à supprimer.

3- Les utilisateurs liés à ce rôle sont affichés.

4-Le rôle ne peut être supprimé tant qu'il a des utilisateurs qui lui sont attribués.

5- Il faut d'abord le vider pour pouvoir le supprimer.

Dans l'attribution rôle/permissions nous avons trois volets:

✓ Dans le volet Utilisateur, on peut voir les rôles auxquels il est affecté, comme le montre la figure suivante:

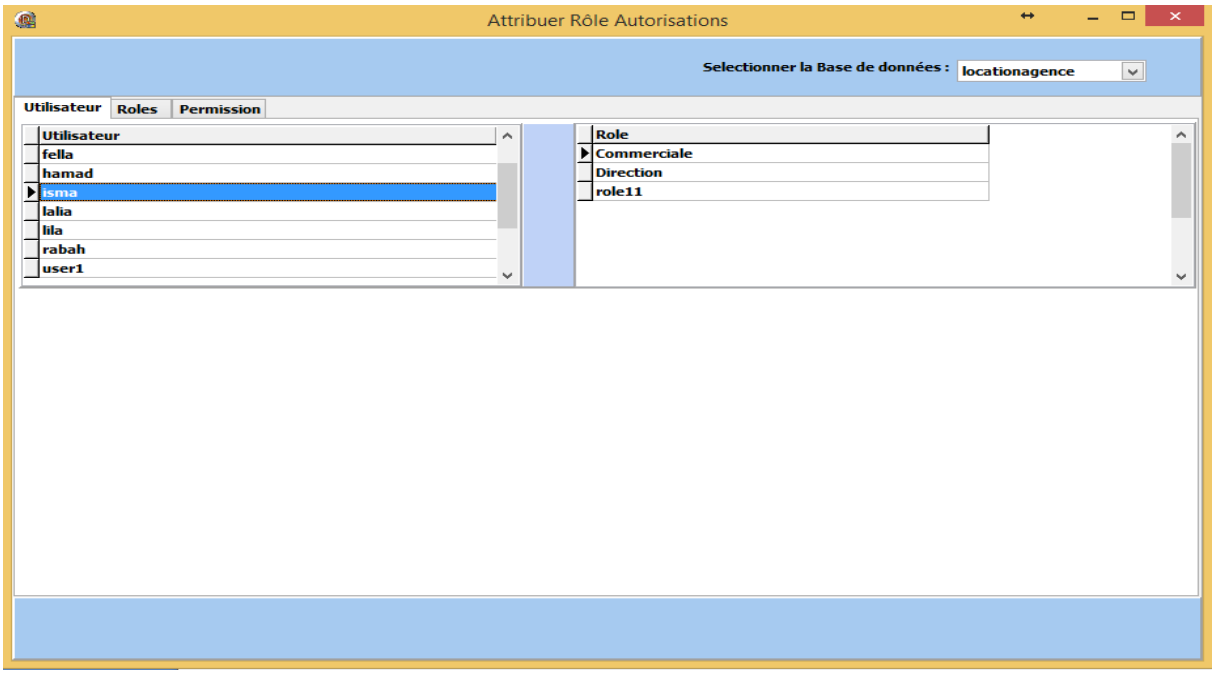

Figure 5.12. *Afficher les rôles de l'utilisateur.*

✓ Dans le volet Rôle, on peut voir les permissions qui lui sont affecté, comme le montre la figure suivante:

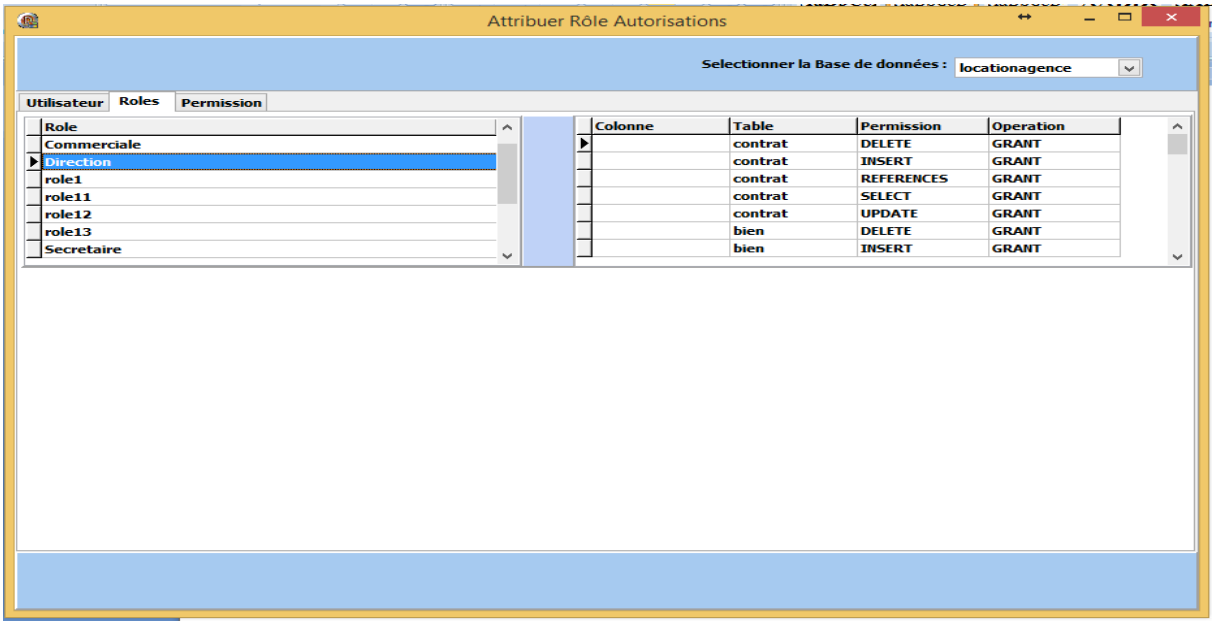

Figure 5.13. *Afficher les permissions des rôles.*

✓ Dans le volet Permission, on attribut les permissions au rôle selectionné:

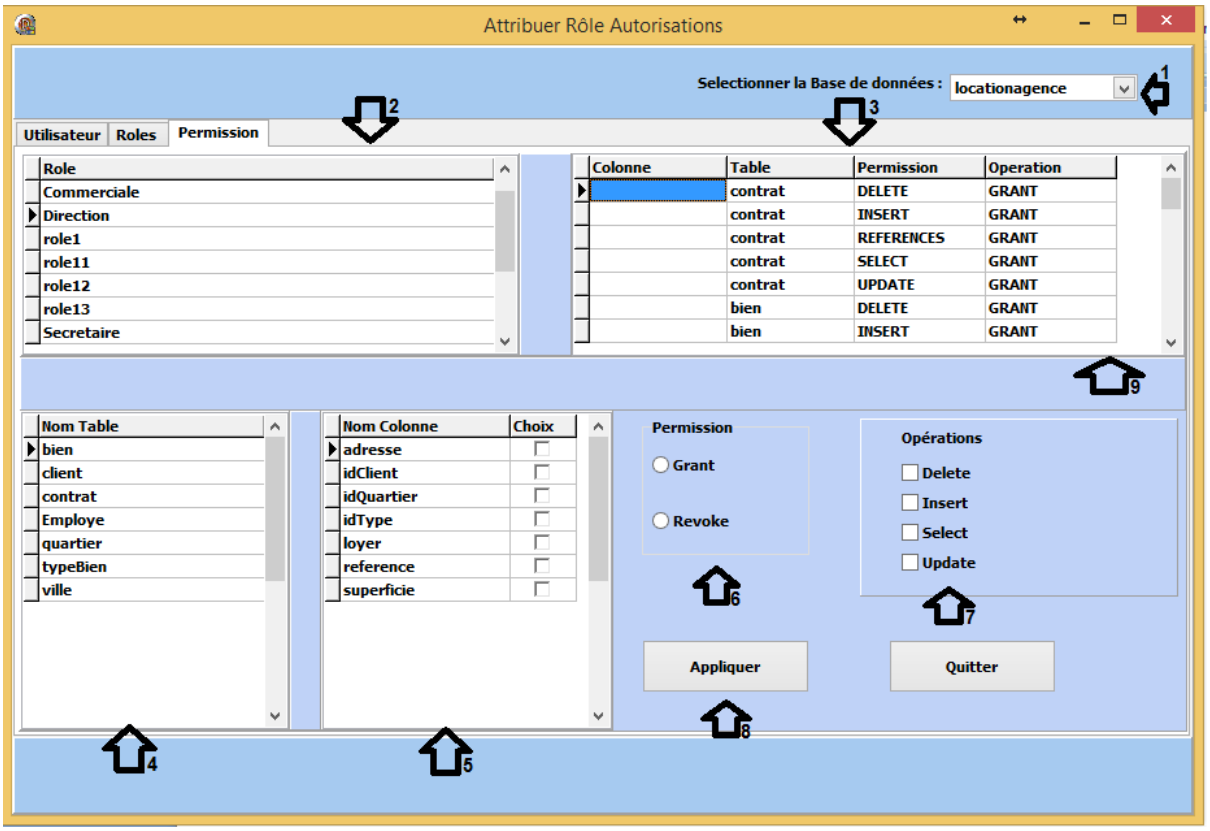

Figure 5.14. *Attribuer les permissions aux rôles.*

1- Choisir le nom de la base de données.

2- Affichage des roles de la base de donnees.

3- Affichage des permissions du rôle selectionné sur les tables de la base de données.

4- Affichage des tables de la base de données.

5- Affichage des colonnes de la table selectionnée.

6- Selectionner le privilège (Grant:autoriser, Revoke:refusé).

7- Selectionner l'operation sur le privilège.

8- Cliquer sur le bouton appliquer pour attribuer les privilèges.

9- Voir que les regles definies ont bien ete appliquees.

7. CONCLUSION :

Ce chapitre fait l'objet d'une démarche structurée pour aboutir à la mise en place d'un RBAC pour SQL SERVEUR, car à notre connaissance ce dernier, ne possède pas une application pour configurer les rôles et les permissions contrairement à Oracle.

En premier lieu, nous avons définit l'environnement de travail et les outils de développement, ensuite les étapes de connexion au serveur de base de données pour procéder au déploiement d'un contrôleur d'accès basée sur la méthode RBAC. Nous terminons le chapitre par la présentation des principales fonctionnalités de notre application RBAC SQL SERVEUR.

Conclusion générale

Dans ce mémoire, nous avons posé la problématique de la sécurité multiniveau d'un serveur de base de données. En particulier, nous nous sommes focalisés sur les politiques de contrôle d'accès pour assurer la sécurité des données. Afin d'atteindre cet objectif, nous avons au cours de l'étude bibliographique montré les notions théoriques principales entourant le sujet en commençant par des notions de base sur la sécurité informatique, passant par la sécurité des applications Web et celle des bases de données. Ensuite nous avons entamé l'étude des différents modèles de contrôle d'accès existants et plus spécifiquement le modèle RBAC (Role Based Access Control) basés sur les rôles. Ce dernier largement adopté par les entreprises et les industriels a été appliqué dans de grandes structures.

En s'appuyant sur le modèle de contrôle d'accès RBAC, le système proposé est doté d'un module d'authentification pour gérer les utilisateurs en leurs attribuant des rôles qui a leurs tours ont des permissions/restrictions bien définis.

La conception de notre système de contrôle d'accès a été faite selon le raisonnement de fonctionnement d'un RBAC. Espérant que notre contribution va ajouter un plus dans le domaine de la sécurité des bases de données, surtout qu'à notre connaissance SQL SERVEUR ne possède pas une application pour gérer les utilisateurs par rôle.

Tout travail de développement et de recherche n'est en réalité qu'une ouverture sur de futurs travaux susceptibles à être améliorés et enrichis, afin de réaliser un système plus performant. Vu les conditions exceptionnelles de cette année (**COVID-19**), il nous a été impossible d'achever les trois objectifs posés au départ c'est pour cela que nous terminons le mémoire par l'illustration des perspectives suivantes, espérant qu'elles enrichissaient notre système :

- Sécurisation des communications avec un canal TLS/PKI (au lieu d'un mur de pare-feu traditionnel)
- Fiabilité et sécurité du service SGBD avec une solution cluster de Fail-Over derrière un mur de pare-feu NG (New Generation, filtrage multi-niveaux et IDS/IPS).

Références Bibliographiques

[1] *Raphael Yende. «* ***Support de cours de sécurité informatique et crypto*** *», Master.Congo-Kinshasa. 2018.cel-01965300.*

Source *: https://hal.archives-ouvertes.fr/cel-01965300/document*

[2] *https://netafrica-sarl.com/fr/securite-des-système -d'informations.*

[3] « THE WORLD OF HACKING: A SURVEY »,

UNIVERSITY OF SINDH, JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY, VOL 4 NO 1 (2020).

[4] *https://www.geeksforgeeks.org/types-of-hackers/*

[5] *https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/*

[6] *https://www.expert-line.com/pssi-politique-de-securite-du-systeme-d-information/*

[7] *GHERNAOUTI Solange, «* ***Sécurité informatique et réseaux*** *», Dunod 4éme édition, Livre, Année 2013.*

[8] «Introduction to Ethical Hacker»,

Source*: Mile2.com: "Information Systems Security", accredited by the NICCS, 2020.*

[9] « Architecture de sécurité pour l'interconnexion en systèmes ouverts d'applications du CCITT »,

Source: *https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.800-199103- I!!PDF-F&type=items*

[10] *https://www.synbioz.com/blog/tech/les-enjeux-de-la-securite-des- applications-web.*

[11] *https://fr.wikipedia.org/wiki/Application_web.*

[12] *https://www.memoireonline.com/05/13/7195/m_Mise-en-place-dune-application-webmapping-de-geolocalisation-des-points-dintert-de-la-vill10.html.*

[13] *BELABSSIRSoukaina Et M. KEITABoubacar****, « Sécurisation des Applications Web Avec ModSecurity »****, Université Mohammed V-Agdal Faculté des Sciences de Rabat-Maroc, Mémoire de Master Spécialisé en Codes, Cryptographie et Sécurité de l'Information, Année 2014.*

[14] « Notions d'architecture client-serveur »,

Source:https://stph.scenari-Community.org/bdd/lap2prs/co/webUC002archi.html?mode=html.

[15] *«* ***Architecture Client/Serveur*** *»,*

Source *: https://web.maths.unsw.edu.au/~lafaye/CCM/cs/csintro.htm*

[16] « Sécurité des applications web- menaces et contre-mesures »,

***Source** : https://www.chiny.me/introduction-a-la-securite-applicative-14-1.php*

[17] «OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks»,

***Source** : https://owasp.org/www-pdf-archive/OWASP_Top_10- 2017_%28en%29.pdf.pdf.*

***[18]** https://lab.dev-code.fr/le-top-10-de-lowasp-2017-les-risques-les-plus-critiques-pour-la-securite-des-applications-web/.*

[19] « OWASP Top 10 Vulnerabilities and Preventions 2020 »,

***Source**: https://cybercrip.com/owasp-top-10/.*

***[20]** https://avengering.com/examiner-les-10-principaux-risques-de-securite-lies-a-owasp/*

***[21]** «Database Security: Attacks, Threats and Control Methods »,*

***Source:http://www.databasecompare.com**/what-is-data-database-(db)-dbms-and-dbs.html*

***[22]** Mubina Malik, Trisha Patel, « **Database security- attacks and control methods** » International Journal of Scientific & Engineering Research, Volume 7, Issue 12, December-2016 313 ISSN 2229-5518.*

***[23]** SweetyR.Lodha, S.Dhande, **« Web database security technique »,** International Journal of Advance Research in Computer Science and Management Studies.*

***[24]** Monika **«Attacks on database and database security»,** International Journal of Academic Research and Development, Volume 1, Issue 5, May 2016, Page No. 48-52 ISSN: 2455-4197*

***[25]** https://www.cs.rit.edu/~gxw9834/cryptotermpaper.docx*

***[26]** Simanta Shekhar Sarmah, **«Database Security –Threats & Prevention»,** International Journal of Computer Trends and Technology (IJCTT) – Volume 67 Issue 5- May 2019*

***[27]** Bertino, Elisa, and Ravi Sandhu. **«Database security-concepts, approaches, and challenges»** IEEE Transactions on Dependable and secure computing 1 (2005): 2-19*

***[28]** Shah, Arun, Robert F. Novy, and Robert A. Ertl**. «Database security** »U.S. Patent No. 7,167,859. 23 Jan. 2007.*

***[29]** Bertino, Elisa, Sushil Jajodia, and PierangelaSamarati. **«Database security: research and practice** »Information systems 20.7 (1995): 537-556.*

***[30]** Pernul, Günther, **«Database security»** Advances in Computers.Vol. 38. Elsevier, 1994.1-72.*

***[31]** Lunt, Teresa F., and Eduardo B. Fernandez. **«Database security»** IEEE Data Eng. Bull. 13.4 (1990): 43-50.*

***[32]** Garvey, Thomas D., and Teresa F. Lunt. **«Cover Stories for Database Security»**DBSec. 1991.*

[33] *Davida, George I., et al.* ***«Database security»*** *IEEE Transactions on Software Engineering 6 (1978): 531-533.*

[34] *Murray, Meg C.* ***«Database security: What students need to know»*** *Journal of information technology education: Innovations in practice 9 (2010): IIP-61.*

[35] *Burtescu,Emil.* ***«Database security-attacks and control methods»*** *journal of applied quantitative methods 4.4 (2009): 449-454.*

[36] *Shulman, Amichai, and C. T. O. Co-founder.* ***"Top ten database security threats."*** *How to Mitigate the Most Significant Database Vulnerabilities (2006).*

[37] *Basta, Alfred, and Melissa Zgola,* ***« Database security »****, Cengage Learning, 2 juil. 2011 - 480 pages.*

[38] *Monali Sachin Kawalkar, Dr. P. K. Butey* ***«An Approach for Detecting and Preventing SQL Injection and Cross Site Scripting Attacks using Query sanitization with regular expression»****. International Journal of Computer Trends and Technology (IJCTT) V49(4):237-245, July 2017.*

[39] *Heloïse Gauvin «M33-2.Cyber-attaques», cours de sécurité –Grenoble –cb- chapitre M33-2.Cyber-attaques ,2017-2018*

Source *: https://slideplayer.fr/slide/12355345/*

[40] *Jacques Le Maitre, « Sécurité des bases de données », Université du Sud Toulon-Var.*

[68] *Yasmina, GHEBGHOUB. «* ***La modélisation des aspects de sécurité dans le Cloud*** *». : Faculté des Sciences Département d'Informatique. Informatique décisionnel, Universite Saad Dahleb – Blida 1, Thèse de doctorat 2017.*

[69] *Marwan, CHEAITO.* ***« Un cadre de spécification et de déploiement de politiques d'autorisation »****, Toulouse III : Ecole Doctorale EDMITT : Mathématiques Informatique Télécommunications Toulouse, Thèse de doctorat : Année 2012.*

[70] *Abou El Kalam et al. «* ***Organization Based Access Control*** *», IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Come, Italy, 2003.*

[71] *R. Thion, «* ***Structuration Relationnelle des politiques de contrôle d'accès Représentation, Raisonnement et Vérification****, ». Thèse de doctorat. Institut National des Sciences Appliquées de Lyon, Année 2008.*

[72] *Michel, EmbeJiague. «* ***Mise en œuvre de politiques de contrôle d'accès formelles pour des applications basées sur une architecture orientée services*** *». Département*

d'informatique. Faculté des sciences, université de sherbrooke. these de doctorat, année 2012.

[73] *A. K. Dey, «* ***Understanding and Using Context. Personal Ubiquitous Computing »,*** *vol. 5, no. 1, pages 4–7, Springer-Verlag, London, 2001.41.*

[74] *J. McCumber,* ***«Information Systems Security:*** *A Comprehensive Model» in: Proceeding of the 14th National Computer Security Conference, NIST, Baltimore, MD, 1991.P 330.*

[75] *Alban Gabillon. «* ***Contrôler les accès aux données numériques****. » La Revue de l'Electricité et de l'Electronique, Société de l'Électricité, de l'Électronique et des Technologies de l'Information et de la Communication, 2013, 12 p. ffhal-02108021ff.*

[76] *Paul Tran Van. «* ***Partage de documents sécurisés dans le Cloud Personnel****. » Réseaux et télécommunications [cs.NI]. Université Paris-Saclay, 2018. Français. ffNNT : 2018SACLV015ff. fftel-01779315ff.*

[77] *Mathieu Blanc. «* ***Sécurité des systèmes d'exploitation répartis : architecture décentralisée de métapolitique pour l'administration du contrôle d'accès obligatoire*** *». Réseaux et télécommunications [cs.NI]. Université d'Orléans, 2006. Français. fftel-00460610.*

[78] *Cédric, Brancourt, « Le contrôle de droit d'accès et la sécurité de vos systèmes », 2015. Source : https://www.synbioz.com/blog/tech/autorisation-et-droits-d-acces.*

[79] *Odile, PAPINI. «* ***Contrôle d'accès : Cours 04*** *». ESIL. Université de la méditerranée.* ***Source :*** *http://odile.papini.perso.esil.univmed.fr/ sources/ SSI.htm*

[80] « Les menaces INFORMATIQUES », Source : https://www.cours-gratuit.com/cours-informatique/ cours-sur-les-menaces-informatiques

[81] *https://openclassrooms.com/ fr/courses/2356306-prenez-en-main- Windows-server/ 5835091-prenez-en-main-les-roles-et-fonctionnalites.*

[82] « Présentation du Delphi *»,*

Source:*https://delphi.developpez.com/codegear/2007/delphi-for-win32/annonce/*