N° d'ordre : 17 / L3 /TCO Année universitaire : 2009/2010

UNIVERSITE D'ANTANANARIVO

ECOLE SUPERIEURE POLYTECHNIQUE

DEPARTEMENT TELECOMMUNICATION

MEMOIRE DE FIN D'ETUDES

en vue de l'obtention

du DIPLOME de
LICENCE ES SCIENCES TECHNIQUES
en TELECOMMUNICATION

par : **RAZAFINDRAKOTO Arnaud Ludovic**

MISE EN PLACE D'UN SYSTEME D'INFORMATION SECURISE SOUS LINUX

Soutenu le 05 Mars 2011 à 8h devant la commission d'Examen composée de :

Président :

M. **RANDRIARIJAONA** Lucien Elino

Examinateurs :

M. **RAKOTOMALALA** Mamy Alain

Mme. **ANDRIANTSILAVO** Haja

M. **RANDRIAMITANTSOA** Andry Auguste

Directeur de mémoire :

M. **RAZAKARIVONY** Jules

REMERCIEMENTS

Ce travail de mémoire de fin d'étude n'aurait pu être réalisé sans la grâce de Dieu, le soutient, les encouragements, les apports et les contributions de nombreuse entités.

Ensuite, je tiens aussi à remercier les personnes suivantes sans qui je n'aurai accompli l'étude que j'ai suivie à l'E. S. P. A., parmi lesquelles :

Monsieur **ANDRIANARY Philippe**, Directeur de l'Ecole Supérieure Polytechnique d'Antananarivo qui m'a permis d'effectuer mes études au sein de l'établissement.

Monsieur **RAZAKARIVONY Jules**, Chef de département de la filière Télécommunications, directeur de ce mémoire et qui m'encadre malgré ses grandes responsabilités, si consacré pour bien ses encadrements et ses précieux conseils.

Monsieur **RANDRIARIJAONA Lucien Elino**, le président du jury qui m'a fait l'honneur de présider ma soutenance de mémoire ; et à Madame et Messieurs les membres de jury :

Monsieur **RAKOTOMALALA Mamy Alain ;**

Madame **ANDRIATSILAVO Haja ;**

Monsieur **RANDRIAMITANTSOA Andry Auguste ;**

Qui ont accepté de sacrifier leur temps pour assister à la présentation de ce mémoire.

Tous les enseignants et tout le personnel de l'Ecole Supérieure Polytechnique d'Antananarivo, en particulier ceux du département Télécommunication.

A toute ma grande famille, mes amis, mes collègues et tous ceux qui de près ou de loin ont contribué à la réalisation de ce travail.

Je vous remercie tous et que le ciel vous donnera tout le bonheur que vous souhaitez.

TABLE DES MATIERES

NOTATIONS ET ABREVIATIONS

AES	American Encryption Standard
ARPANET	Advanced Research Projects Agency Network
ASP	Active Server Page
ASCII	American Standard Code for Information Interchange
CA	Certification Authority
CETE	Centre d'Etude Technique de l'Equipement
CGI	Common Gateway Interface
CIT	Centre Technique d'informatique
DES	Data Encryption Standard
DNS	Domaine Name Server
DSS	Digital Signature Standard
EDI	Echange des Données Informatisées
FAI	Fournisseur d'accès Internet
FDDI	Fiber Distributed Data Interface
FI	Forms Interpreter
FTP	File Transfert Protocol
FTPS	File Transfert Protocol Secure
HTTP	HyperText Transfert Protocol
HTTPS	HyperText Transfert Protocol Secure
ICMP	Internet Control Message Protocol
IGMP	Internet Group Management Protocol
IP	Internet Protocol
ISO	International Standards Organization
ISP	Internet Service Provider
IUT-T	International Union of Telecommunication
LAMP	Linux Apache Mysql Php
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MCD	Modèle Conceptuelle de Données
MD5	Message Digest 5

MLD	Modèle Logique de données
MPD	Modèle Physique de données
NNTP	News Network Transfer Protocol
OSI	Open System Interconnection
PC	Personal Computer
PEM	Privacy Enchaced Message
PHP	Personal Home Page
RPM	RedHat Package Manager
RSA	Riverst Shamir Adleman
SGBD	Système de Gestion de Base de Données
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Langage
SSL	Secure Socket Layers
TCP	Transport Control Protocol
TLS	Transport Layer Security
UDP	User Datagramme Protocol
URL	Uniform Ressource Locator
WAMP	Windows Apache MySQL PHP
WWW	World Wide Web

INTRODUCTION

Actuellement, l'Internet a constitué une évolution majeure. Il est maintenant si naturel, qu'on croirait qu'il a toujours existé. Les facilités offertes pour les transferts de fichiers, le courrier électronique, les listes de diffusion, les web, les forums et surtout les commerces électroniques ou e-commerce.

L'ouverture d'Internet à des activités privées ou commerciales qui confirment la nécessité, pour les organismes qui veulent pouvoir librement communiquer, stocker et traiter les données, de protéger leurs systèmes d'informations.

Un système d'information, c'est l'ensemble organisé de ressources (personnel, données, procédures, matériel, logiciel), permettant d'acquérir, de stocker, de structurer et de communiquer des informations sous forme de textes, images, sons, ou de données codées dans des organisations.

Sur le but du développement de l'internet, la sécurité est une grande notion pour la mise en place d'un serveur web. Ce n'est pas le cas des virus mais pour les pirates informatiques. On peut minimiser les virus par l'utilisation du système UNIX comme le serveur. Mais pour les pirates informatiques, la meilleure solution est de chiffrer les échanges par des différents mécanismes des cryptographies. C'est le cas du protocole SSL qui est le moyen le plus utilisés pour crypter des données aux échanges entre client et serveur.

Ce mémoire de fin d'études est orienté dans cette direction intitulée « MISE EN PLACE D'UN SYSTEME D'INFORMATION SECURISE SOUS LINUX », il est organisé de la manière qui suit.

Le premier chapitre nous informe sur l'environnement web et système d'information. Le second nous explique ce que sont l'architecture du client-serveur et sécurité. Et pour le troisième chapitre qui termine cet ouvrage affirme la mise en œuvre de l'application sous Linux.

CHAPITRE 1
ENVIRONNEMENT WEB ET SYSTEME D'INFORMATION

1.1 Environnement web

1.1.1 Notion sur Internet et Intranet

1.1.1.1 Internet

a. Définition

L'internet est un réseau international (réseaux de réseaux ou inter-réseaux) d'ordinateurs communiquant entre eux grâce à un protocole d'échanges de données standardisé (TCP/IP ou Transport Control Protocol/Internet Protocol). Chaque ordinateur de réseau possède une adresse, appelé adresse IP ou adresse internet, qui est unique [1].

L'Internet a été créé surtout en vue de la transmission de données, cependant l'usage de la transmission de la voix et de la vidéotransmission croît sans cesse [2].

b. Historique

Voici l'évolution de l'internet [3] [4]:

En 1957 : Création de l'ARPA par le ministère américain de la défense

En 1966 : L'ARPA devient l'ARPANET.

En 1972 **:** Démonstration par ARPANET d'une communication avec 40 ordinateurs.

En 1973 **:** Développement du protocole TCP/IP.

Premières connexions internationales d'ARPANET avec la Norvège et la Grande- Bretagne.

En 1974 : Mise au point de la norme IP

En 1982 : 400 ordinateurs connectés à USENET (utilisateurs UNIX).

On commence à parler de « l'Internet ».

En 1983 : 1er janvier, bascule d'ARPANET sur le protocole TCP/IP.

Invention du DNS (Domain Name System).

En 1989: Naissance du World Wide Web.

En 1990 : Fin de l'ARPANET.

En 1991 : Le 6 août, le WWW est rendu public.

En 1996 : 10 millions de machines sont connectées.

Actuellement, le nombre d'utilisateurs de l'Internet dépassera celui des utilisateurs du téléphone.

c. Services Internet

Un certain nombre de services sont disponibles sur internet [5] :

- Echanger des courriers et des documents : il permet d'échanger des courriers avec toutes personnes possédant une adresse électronique ;
- Protocole utilisé : Simple Mail Transfer Protocol (SMTP).
- Transférer des fichiers d'une machine à une autre : il permet directement de transférer des fichiers d'une machine à une autre. Protocole utilisé : File Transfer Protocol (FTP) ;
- Participer à des groupes de discussions : il s'agit d'un immense ensemble de forum. Protocole utilisé : News Network Transfer Protocol (NNTP) ;
- Accéder à des pages hypertexte et hypermédia : World Wide Web, il permet d'accéder aux pages web qui écrite en langages HTML (Hyper Text Markup Language) et peut contenir du texte, des images (statiques ou animés), des séquences vidéos, des sons et des liens.

d. Dénomination de domaine à l'internet

Afin de rationnaliser les millions d'adresses IP que constituent l'internet, la technologie DNS a été choisie afin de faire correspondre les adresses IP en nom intelligibles. Par exemple,

www.microsoft.com = 207.46.197.113.

Un certain nombre de domaines principaux ont été définis pour normaliser les noms :

- **.com** correspond aux entreprises à vocation commerciales (désormais ce code de domaine ne rime plus à grand chose et est devenu international) ;
- **.edu** correspond aux organismes éducatifs ;
- **.gov** correspond aux organismes gouvernementaux ;

- **.mil** correspond aux organismes militaires ;
- **.net** correspond aux organismes ayant trait aux réseaux ;
- **.org** correspond aux entreprises à but non lucratif ;
- **.xx** correspondant aux codes pays (Par exemple : **.mg** [Madagascar]).

1.1.1.2 Intranet

a. Définition

L'intranet est un réseau interne d'une entreprise qui utilise les mêmes logiciels que sur Internet pour diffuser ses informations ou permettre la communication entre les utilisateurs.

b. Description et ses services

L'intranet standardise aussi le modèle du client/serveur comme à l'internet en utilisant le protocole TCP/IP.

L'architecture de l'intranet compose trois niveaux comme l'internet :

- Le client, navigateur internet ;
- Le serveur d'application : un serveur web permettant d'interpréter des scripts CGI, PHP, ASP ou autres, et les traduire en requêtes SQL afin d'interroger une base de données ;
- Le serveur de bases de données.

Les services d'intranet sont :

- Mise à disposition d'informations sur l'entreprise (panneau d'affichage) ;
- Mise à disposition de documents techniques ;
- Moteur de recherche de documentations ;
- Un système de gestion ;
- Un échange de données entre collaborateurs ;
- Annuaire du personnel ;
- Gestion de projets, aide à la décision, agenda, ingénierie assistée par ordinateur ;

- Messagerie électronique ;
- Forums de discussion, listes de diffusions, chat en direct ;
- Visio conference ;
- Portail vers internet.

1.1.2 Web

1.1.2.1 Quelques définitions

Voici quelques définitions qui entourent au mot web [2] [6] :

Le **Web** est l'interface des navigateurs du réseau Internet.

Un **serveur Web** est un hôte sur lequel fonctionne un **serveur HTTP** (ou **serveur Web**). Un serveur Web **héberge** les ressources qu'il dessert.

Un **navigateur Web** est un logiciel client HTTP conçu pour accéder aux ressources du Web. Sa fonction de base est de permettre la consultation des documents HTML disponibles sur les serveurs HTTP. Le support d'autres types de ressource et d'autres protocoles de communication dépend du navigateur considéré.

Une **page Web** (ou **page**) est un document destiné à être consulté avec un navigateur Web. Une page Web est toujours constituée d'une ressource centrale (généralement un document HTML) et d'éventuelles ressources liées automatiquement accédées (typiquement des images).

Un **éditeur HTML** (ou **éditeur Web**) est un logiciel conçu pour faciliter l'écriture de documents HTML et de pages Web en général.

Un **site Web** (ou **site**) est un ensemble de pages Web et d'éventuelles autres ressources, liées dans une structure cohérente, publiées par un propriétaire (une entreprise, une administration, une association, un particulier, etc.) et hébergées sur un ou plusieurs serveurs Web.

Visiter un site Web signifie « consulter ses pages ». Le terme **visite** vient du fait que l'on consulte généralement plusieurs pages d'un site, comme on visite les pièces d'un bâtiment. La visite est menée par un **utilisateur** (ou **visiteur** ou **internaute**). La mesure d'audience est obtenue en copiant le code en JavaScript d'un lien vers le site d'un prestataire spécialisé suivant la technique du marqueur à distance.

Une **adresse Web** est une URL de page Web, généralement écrite sous une forme simplifiée limitée à un nom d'hôte. Une adresse de site Web est en fait l'adresse d'une page du site prévue pour accueillir les visiteurs.

Un **hébergeur Web** est une entreprise de services informatiques hébergeant (mettant en ligne) sur ses serveurs Web les ressources constituant les sites Web de ses clients.

Une **agence Web** est une entreprise de services informatiques réalisant des sites Web pour ses clients.

L'expression **surfer sur le Web** signifie « consulter le Web ». Elle a été inventée pour mettre l'accent sur le fait que consulter le Web consiste à suivre de nombreux hyperliens de page en page. Elle est principalement utilisée par les médias ; elle n'appartient pas au vocabulaire technique.

Un **annuaire Web** est un site Web répertoriant des sites Web.

Un **portail Web** est un site Web tentant de regrouper la plus large palette d'informations et de services possibles dans un site Web. Certains portails sont thématiques.

Un **service Web** est une technologie client-serveur basée sur les protocoles du Web.

1.1.2.2 Principe du web [7]

Le web forme un vaste réseau d'ordinateurs reliés les uns aux autres. Sur le web, le protocole de transmission de données s'appelle HTTP ou HyperText Transfer Protocol. Grace è ce protocole, les réseaux se relient à l'aide d'un lien. Par ailleurs, les données sont organisées en « pages » d'informations. Ces pages répondent à la norme HTML ou HyperText Markup Langage, langage de balisage qui définit la mise en forme des pages d'un site web (texte, images, du son, etc.) à savoir la création de documents hypertextes affichables par un navigateur web. Une page au format HTML peut donc inclure du texte ainsi que des images fixes ou animées, du son, de la vidéo, des programmes interactifs.

Le web apparait donc comme une immense toile de pages d'informations qui sont reliées entre elles par des liens logiques (liens hypertextes). Ces liens permettent de naviguer facilement et de manière quasi transparente d'un site à l'autre, sur le réseau Internet constituant ainsi un vaste maillage à travers le monde.

L'internet ou Interconnected Network est l'interconnexion de plusieurs réseaux locaux prenant référence aux modèles OSI (Open System Interconnection) et géré par divers protocoles comme le TCP/IP.

L'une des notions les plus importantes à saisir lorque l'on travaile sur web concerne le format URL ou Uniform Ressource Locator : localisateur uniforme de ressource). En fait, l'URL est une adresse sur le web, de la forme « http://machine hote/chemin d'accès » ou « https://machine hote/chemin d'accès », qui identifie chaque page web et chaque document de manière unique.

1.1.3 Généralités sur le site web [2] [8]

Un site web est un principe pour naviguer à l'internet.

Les sites web sont les éléments constitutifs sur web. Techniquement, le site web est un ensemble cohérent de pages web liées par des liens hypertextes et articulées autour d'une page d'accueil commune.

L'ensemble des pages du site se trouve généralement sous un même nom de domaine. De façon pus précise, il s'agit d'un serveur d'information inscrit sur le web grâce à son adresse unique URL.

La consultation des pages d'un site s'appelle une visite, car les hyperliens entre les pages permettant de consulter toutes les pages du site sans le quitter (sans devoir consulter un web hors site). A part les nombreuses informations disponibles sur les sites web, ils peuvent aussi nous offrir un grand nombre de service. Selon leurs catégories, on distingue les moteurs de recherches comme le Google, Yahoo, les annuaires web, les commerces électroniques, les messageries électroniques sur Internet.

Pour consulter un site web, l'utilisateur doit posséder un logiciel que l'on appelle navigateur web ou browser en anglais. Ces logiciels qui sont capables de communiquer avec les serveurs web, d'interpréter les instructions contenues dans les pages HTML et les remettre en page pour l'utilisateur comme l'Internet Explorer, Firefox Mozilla, Netscape.

Il y a différents types de sites web: site vitrine pour présenter son association, site de contenus pour les grandes masses d'informations, structuration de cet information et à propos éditorial et site pour vendre en ligne comme l' e-commerce.

On a deux caractéristiques du site web :

- site web statique ;
- site web dynamique.

1.1.3.1 Site web statique

Les pages web statiques sont des simples fichiers textes codées HTML.

Ils sont très pratiques pour créer un site contenant quelques dizaines de pages mais possédant leurs limites, c'est-à dire les documents ne peuvent pas modifier par les visiteurs sauf que le créateur. Voici un schéma qui se passe quand l'utilisateur consulte une page web dite « statique » [18].

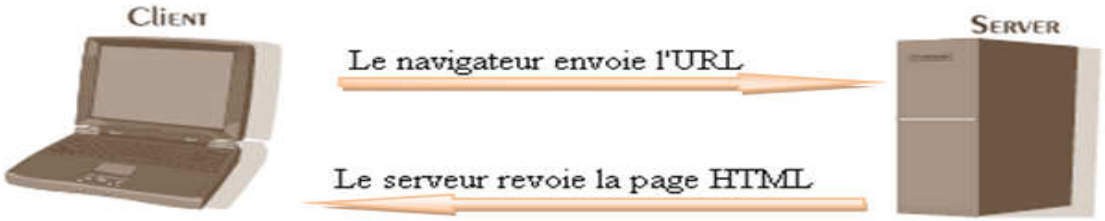

Figure 1.01: *Consultation d'un site statique*

1.1.3.2 Site web dynamique

Les pages web statiques sont souvent utilisées pour construire une page d'accueil d'un site web, tandis que les pages web dynamiques nous rendent familier à la création, à la manipulation puis à la gestion d'une base des données qui transforment la page dynamique.

Les pages web dynamiques sont de pages web avec des informations qui changent ou sont changées automatiquement en fonction d'une base de données ou d'élément provenant de l'utilisateur. La plupart des fois, les adresses URL de ces pages se terminent avec les extensions suivantes : **.asp, .cfm, .cgi, .shtml**. Mais, il est aussi possible d'avoir des pages avec un contenu dynamique et portant les extensions habituelles notamment **.html** ou **.htm** [18].

Les moteurs de recherche référent ces pages dynamiques de la même manière que les pages avec un contenu statique.

Voici le chemin des requêtes lorsqu'un visiteur visite uns site web dynamique :

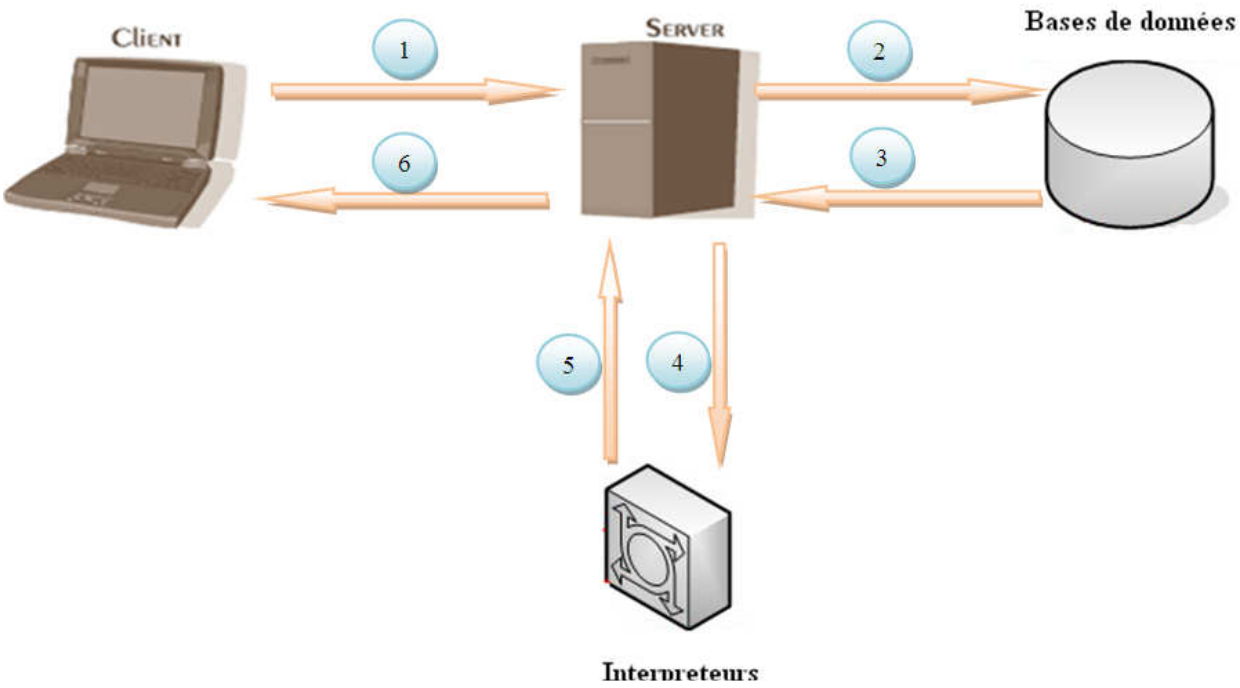

Figure1.02 : *Un site web dynamique*

1: Le visiteur envoie une requête

2: Le serveur consulte cette requête aux bases de données

3: Résultat via des bases de données : le fichier contient du html, du JavaScript, du PHP

4: Le PHP est extrait et interprété par un interpréteur : wamp, lamp ou xamp

5: L'interprétation du PHP génère du html, du JavaScript qui remplace le PHP

6: le tout (en html, JavaScript) est retourné au navigateur web

Le langage PHP est donc un langage de script côté serveur, ce qui veut dire que c'est le serveur (la machine qui héberge la page web en question) qui va interpréter le code PHP et générer du code (par exemple du HTML, du CSS, et parfois même du JavaScript) qui pourra être retourné et interprété par le navigateur web.

1.1.4 Les langages de développements sur web

1.1.4.1 Le langage HTML

a. Définition

Le langage HTML (HyperText Markup Language) est utilisé pour la rédaction de pages Web. Il se compose de balises, désignant des caractéristiques spéciales. Certaines modifient l'aspect d'une page Web, d'autres son fonctionnement. Des balises sont ainsi prévues pour :

- la création de liens vers d'autres pages Web ou vers un élément particulier d'une page donnée ;
- les éléments structurels (tableaux ou listes) ;
- les éléments graphiques, par exemple des illustrations, des mini-applications

(« Applets ») ou des séquences QuickTime.

b. La syntaxe HTML

Un fichier HTML comprend d'une part les informations que vous souhaitez publier et d'autre part les commandes HTML désignées sous le nom de balises.

Les Balises sont facilement identifiables dans un fichier, elles sont entourés des symboles< >.

De nombreuses commandes HTML se composent souvent d'une balise d'ouverture et d'une balise de fermeture. Par exemple : ouverture et fermeture.

Voici un exemple la balise qui permet de mettre en texte en gras, le texte entre ces deux balise sera en gras .

Il existe aussi des balises autonomes, qui n'ont pas de balise de fermeture, la zone d'application de la commande n'ayant pas de sens : par exemple la commande
 saut de ligne.

Il est possible de doter les balises de valeurs spécifiques, en règle générale ces valeurs sont placées dans la balise d'ouverture.

Par exemple <h1 align=center> ce titre est centré </h1>.

c. Quelques exemples des balises HTML

Voici quelques exemples des balises HTML :

Balises	Description
Fichier Html	
<HTML>...</HTML>	Début et fin de fichier HTML
<TITLE>...</TITLE>	Titre affiché par le navigateur (élément de HEAD)
<BODY>...</BODY>	Début et fin du corps du fichier HTML
<BODY bgcolor="#XXXXXX">	Couleur d'arrière-plan (en hexadécimal)
<BODY background="xyz.gif">	Image d'arrière-plan
Hyperliens	
<A href="http://...">...</A>	Lien vers une page Web
<A href="mailto:...">...</A>	Lien vers une adresse de courriel
Images	
<IMG src="xyz.gif"> <IMG src="xyz.pjg>	Insertion d'une image au format gif ou jpeg
Mise en forme des caractères	
<FONT color="#XXXXXX">...</FONT>	Texte en couleur où XXXXXX est une valeur hexadécimale
<U>...</U>	Texte souligné
<P>...</P>	Nouveau paragraphe

Tableau 1.01: *Exemple des balises HTML*

1.1.4.2 Le langage JavaScript [9]

a. Présentation

JavaScript est utilise dans des millions de pages web afin d'améliorer leur conception. Il s'agit d'une couche de programmation supplémentaire qui vient s'ajouter au langage HTML. Le code HTML est le langage de base que toute page Internet se doit d'utiliser : en plus de son rôle proche d'un traitement de texte, ce langage permet de surfer grâce aux liens hypertextes. Quant à JavaScript, il a été conçu pour donner plus d'interactivité aux pages HTML. Le mot script indique qu'il s'agit d'un langage de programmation simplifie qui s'exécute en local sur l'ordinateur qui est en train de lire la page web. Ce langage, comme l'HTML, ne nécessite l'achat d'aucune licence pour pouvoir l'utiliser.

Initialement, JavaScript a été développé par Netscape, mais maintenant la plupart des explorateurs permettant de naviguer sur Internet sont compatibles avec JavaScript.

b. JavaScript et Java sont deux langages différents

Il est important de bien préciser que même si les noms sont très proches, Java et JavaScript sont deux langages bien distincts, autant du point du vue de leurs concepts que de leur conception. Java est développée par Sun Microsystems et est un langage de programmation bien plus puissant et complexe que JavaScript. Java peut se comparer au langage C++.

c. Utilité de JavaScript

Les possibilités de JavaScript sont multiples :

- JavaScript livre aux concepteurs de pages web un outil de programmation avec une syntaxe élémentaire. Contrairement à un vrai langage, presque tout le monde peut insérer un petit bout de code JavaScript dans leur page web ;
- JavaScript permet l'utilisation de textes dynamiques dans une page web. Un texte dynamique est un texte qui change selon les conditions de son environnement (on peut par exemple afficher la date d'aujourd'hui) ;
- JavaScript peut réagir à un événement particulier. Par exemple quand une page a terminé son chargement ou quand un utilisateur clique sur un élément HTML ;
- JavaScript peut lire et écrire des éléments HTML.

1.1.4.3 Le langage PHP

a. Historique [10]

PHP a été crée en 1995 par Rasmus Lerdorf. Son premier nom était *Personal Home Page Tools* et il s'agissait à l'époque d'une bibliothèque de scripts Perl. C'est ensuite devenu PHP/FI (*Personal Home Page /Forums Interpreter*) car, chose rare à cette époque, PHP/FI gérait les formulaires. Puis PHP/FI 2 est sorti en 1997. C'est alors que PHP 3(pour *PHP Hypertext PreProcessor*) fut publié en 1998. C'est cette version 3 qui a fait mondialement connaitre PHP. En 200, c'est PHP 4 qui a vu le jour et plus de 300.000 sites sont déjà tourné. Cette version est aujourd'hui la plus répandue dans le monde.

b. Définition [11]

PHP est un langage dédié à la production de pages HTML générées dynamiquement. C'est un langage de script HTML qui fonctionne cote serveur (l'interpréteur de PHP est intégré au serveur web).

Ce qui distingue le PHP des langages de script comme le JavaScript est que le code est exécute sur le serveur. Si vous avez un script similaire sur votre serveur, le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat. Vous pouvez configurer votre serveur web afin qu'il analyse tous vos fichiers HTML comme des fichiers PHP. Ainsi, il n'y a aucun moyen de distinguer les pages qui sont produites dynamiquement des pages statiques

c. Interprétation du PHP par le serveur

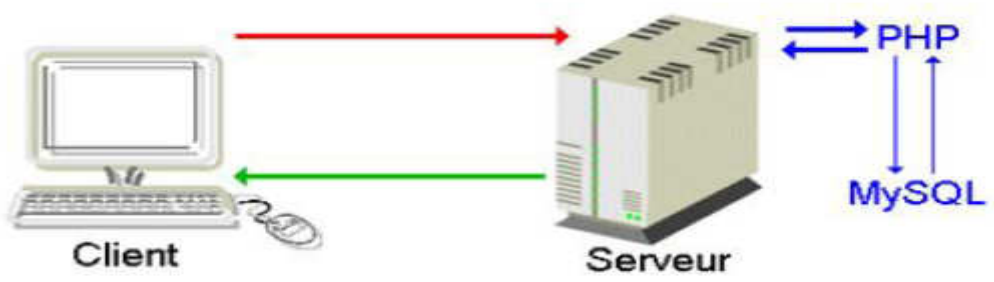

Figure 1.03: *Interprétation du PHP par le serveur*

Détaillons ce qu'il se passe lorsque vous consultez une page Html dite dynamique :

- Votre navigateur envoie l'adresse que vous avez encodée.
- Le serveur Web cherche dans son arborescence si le fichier existe et si celui-ci porte une extension reconnue comme une application PHP (.php, .php3). Si c'est le cas, le serveur Web transmet ce fichier à PHP.
- PHP interprète le fichier, c'est-à-dire qu'il va analyser et exécuter le code PHP. Si ce code contient des requêtes vers une base de données MySQL, PHP envoie la requête SQL. La base de données renvoie alors les informations voulues au script qui peut les exploiter (pour les afficher par exemple).
- PHP continue d'interpréter la page, puis retourne le fichier dépourvu du code PHP (puisqu'il est exécuté) au serveur Web.
- Le serveur Web renvoie finalement le fichier au navigateur de l'utilisateur. Cet fichier ne contient que du HTML.

Un script PHP est commencé toujours par le tag **<? php** et finit par **?>**.

1.2 Systeme d'information

1.2.1 Définition

Le système d'information est l'ensemble des moyens techniques et humains qui permet de stocker, de traiter ou de transmettre l'information. [19].

1.2.2 Objectifs

Un système d'information a pour but de [19]:

- Contribuer pour le pilotage de l'entreprise ou de ses activités en fournissant de l'information pour le management.
- Supporter la réalisation des activités de l'entreprise en traitant de la matière information.

1.2.3 Principe

En principe il exerce les fonctions suivantes [1]:

- Collecter ou acquérir les informations**:** C'est-à-dire saisie et consultation des bases de données de l'entreprise.

- Stocker ou mémoriser les informations: Enregistrement des informations sur des supports, en générale organisé en base de données.
- Traiter les informations: Transformation du contenu ou de la forme des informations par des programmes informatiques ou des interventions manuelles.
- Diffuser les informations: Transmission d'information entre différentes acteurs ou fonction.

1.2.4 Conception de la base de données

Notre système d'information est un site web d'hôtel nommé « Gas'Hotel ».

1.2.4.1 Introduction

Il est assez difficile de définir ce qu'est une base de données si ce n'est que de dire trivialement que tout système d'information peut être qualifie de base de données. Il semble plus facile de définir l'outil principal de gestion d'une base de données : le système de gestion de bases de données ou SGBD.

- C'est un outil permettant d'insérer, de modifier et de rechercher efficacement des données spécifiques dans une grande masse d'informations.
- C'est une interface entre les utilisateurs et la mémoire secondaire facilitant le travail des utilisateurs en leur donnant l'illusion que toute l'information est comme ils le souhaitent.

Chacun doit avoir l'impression qu'il est seul à utiliser l'information.

1.2.4.2 Définition

Une base de données est par définition une entité dans laquelle in est possible de stocker des données de façon structurées et avec le moins de redondance possible. Ces données sont conçues pour pouvoir être utilisées par des programmes et par des utilisateurs différents. Ainsi, la notion de base de données est généralement associés à celle de réseau, afin de pouvoir mettre en commun ces informations, d'où le nom de « base » [17].

On parle souvent de système d'information pour indiquer toute la structure rassemblant les méthodes mise en place pour partager les données.

Une base de données permet de mettre des données à la disposition d'utilisateurs pour une consultation, une saisie ou bien une mise à jour, tout en respectant les droits réservés à ces derniers.

Une base de données peut être locale, c'est-à dire accessible sur une machine par un utilisateur, ou bien repartie, cela veut dire que les informations sont stockées sur des machines distant et partagées en réseaux.

1.2.4.3 Traitement du système d'information avec Merise

Merise est une méthode française pour avoir un système d'information automatisée, efficace, flexible et adaptée à l'entreprise.

Elle a pour objectif d'améliorer le traitement de l'information dans une organisation c'est-à dire amélioration des traitements, collection, saisie, transmission et stockage de l'information.

Voici le processus de conception d'une base de données :

- Analyse et spécification des besoins ;
- Modèle conceptuel des données ;
- Modèle relationnel.

Voici un schéma de ce processus de conception d'une base de données :

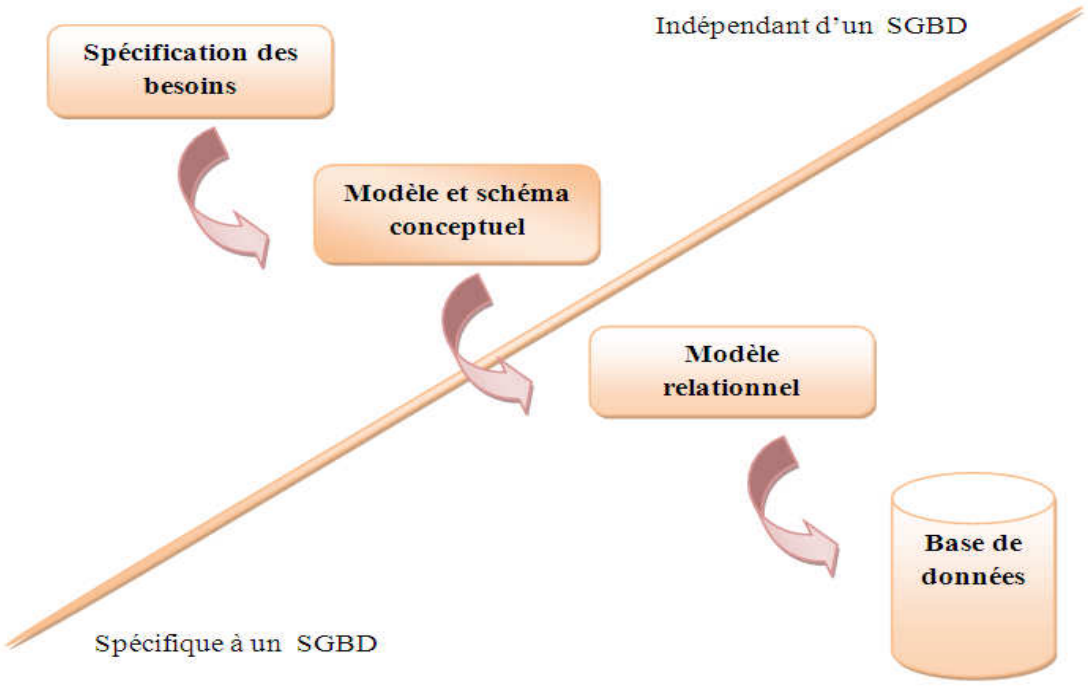

Figure 1.04: *Conception d'une base de données*

1.2.4.4 Spécifications des besoins

a. Descriptions des données dans l'environnement du site web

Notre étude consiste à réaliser une base de données d'un site web d'Hôtel. Cette base de données occupe tous les enregistrements des données ou des stockages des données sur ce site dynamique. Après une analyse approfondie, on a les informations suivantes liées principalement à cette base :

- Les clients : qui occupent les chambres, faire une réservation des chambres ;
- Les chambres : qui se font réservées par les clients ;
- Les réservations : qui sont effectuées par les clients.

b. Descriptions des logiciels utilisées

- Macromedia Dreamweaver

Macromedia Dreamweaver est un éditeur html professionnel concerné aux codages et au développement de sites, des pages et des applications web. Dans notre site web dynamique, il

permet de créer des applications web reposant sur des bases de données dynamiques au moyen de technologie des serveurs.

Figure 1.05 : *Macromedia Dreamweaver*

- Wamp serveur

Wamp serveur installe et configure automatiquement un environnement de travail pour un site web. En effet, il est donc un outil de développement et non pas de production. C'est en fait un pack regroupant :

- Le serveur web apache 2.2.11 ;
- Le moteur de script PHP 5.3.0 ;
- La base de données MYSQL 5.1.36 ;
- Le module d'administration de la base PHPMyAdmin 3.2.0.1.

Figure 1.06: *Wamp Server 2.0*

Lorsque vous le démarrez, vous pouvez juste voir une icône à droite de la barre des tâches (pas loin de l'horloge) :

Figure 1.07: *Icône Wamp*

- Win'Design

C'est un logiciel qui sert à élaborer les modélisations des données telque la modélisation conceptuelle des données et la modélisation relationnel des données.

Figure 1.08 : *Présentation du Win'Design*

1.2.4.5 Modèle conceptuel des données

Le Modèle Conceptuel de Données est la représentation de l'ensemble des données du domaine, sans tenir compte des aspects techniques et économiques de mémorisation et d'accès et sans se référer aux conditions d'utilisation par tel ou tel traitement.

Dans un système d'information en fonctionnement, données et traitements apparaissent intimement liés (surtout du point de vue de l'utilisateur). L'ensemble des informations utilisées et échangées constitue l'univers du discours du domaine. Dans cet univers du discours, on fait référence à des objets concrets ou abstraits (l'assuré, le contrat) et à des associations entre ces objets (le contrat comporte des garanties). L'objectif du modèle conceptuel de données consiste à identifier, décrire par des informations et modéliser ces objets et associations.

Le formalisme utilisé dans Merise est désigné par le terme « entité-association ». Il comporte quatre concepts types de base [17]:

- Deux concepts structuraux : l'entité et l'association ;

- Un concept descriptif : l'attribut ;
- Un concept qualificatif : la cardinalité, qui qualifie la liaison entre entité type et relation type.

a. L'entité

L'entité est une population d'individus homogènes. Il possède un identifiant et des propriétés.

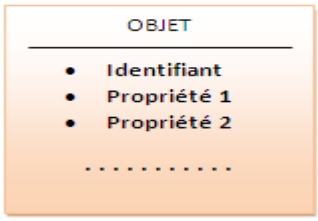

Figure 1.09 : *Entité*

b. L'association

L'association est une liaison qui a une signification précise entre plusieurs entités.

Figure 1.10: *Association*

c. L'attribut

L'attribut est une propriété d'entité ou d'une association. Elle est une information élémentaire prise sur une entité.

d. La cardinalité

La cardinalité d'un lien entre entité et association précise le minimum et le maximum de fois qu'un individu de l'entité peut être concerné par l'association.

La cardinalité d'une relation est composée d'un couple comportant une borne maximale et une borne minimale, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur :

- la borne minimale (généralement 0 ou 1) décrit le nombre minimum de fois qu'une entité peut participer à une relation ;
- la borne maximale (généralement 1 ou n) décrit le nombre maximum de fois qu'une entité peut participer à une relation ;
- Une cardinalité 1.N signifie que chaque entité appartenant à une classe d'entité participe au moins une fois à la relation ;
- Une cardinalité 0.N signifie que chaque entité appartenant à une classe d'entité ne participe pas forcément à la relation.

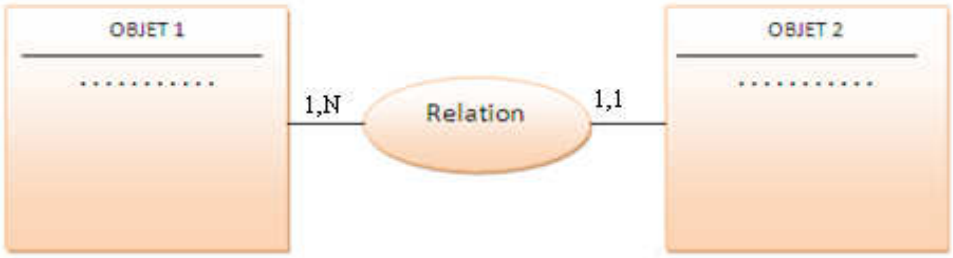

Figure 1.11 : *Cardinalité*

1.2.4.6 Modèle relationnel de données

Le modèle relationnel de données ou modèle logique de données est une description des données, issue de la modélisation conceptuelle et organisationnelle des données [17].

Elle est exprimée dans un formalisme général (SQL) compatible avec l'état de l'art technique des systèmes de gestion des bases de données.

La modélisation logique des données s'obtient aux opérations suivantes :

- Transformation du MCD/MOD, exprimé en formalisme entité-relation, en un MLD exprimé en formalisme logique
- Optimisation générale tenant compte des aspects coûts/performance, des compléments propres à la modélisation relationnelle et des contraintes techniques d'implémentation.

Le modèle logique sera ensuite transformé et adapté en fonction des spécificités du langage de définition de données spécifique à l'outil (par exemple, SGBD) retenu, pour devenir modèle physique de données.

Le modèle relationnel s'appuie sur les concepts de base suivants : la table, l'attribut, la clé primaire et la clé étrangère.

A partir de la modèle conceptuelle entité association, nous allons créer le modèle relationnel, qui nous permettra d'élaborer les tables de la base de données.

Modèle conceptuel (entité association)	Modèle relationnel
Association, entité	Table
Propriété	Attribut
Identifiant	Clé primaire

Tableau 1.02: *Passage du modèle conceptuel au modèle relationnel*

Le passage du modèle conceptuel au modèle logique au niveau des classes de relation se fait selon les cardinalités des classes d'entité participant à la relation, donc il est nécessaire de respecter les règles ci-dessous.

Règle 1 : Toute classe d'entité du diagramme entité/association est représentée par une relation dans le schéma relationnel équivalent. La clé se cette relation est l'identifiant de la classe d'entité correspondante.

Règle 2 : Toute classe d'association est transformée en relation. La clé de cette relation est composée de tous les identifiant des entités participantes.

Règle 3 : Toute classe d'association reliée à une classe d'entité avec une cardinalité de type 0,1 ou 1,1 peut être fusionnée avec la classe d'entités. Dans ce cas, on déplace les attributs de la classe d'associations vers ceux de la relation traduisant la classe d'entités.

Remarque :

- Une même table peut avoir plusieurs clés étrangère mais une seule clé primaire ;
- Une colonne clé étrangère peut aussi être primaire ;

- Une clé étrangère peut être composée ;
- Implicitement, chaque colonne qui compose une clé primaire ne peut pas recevoir la valeur vide ;
- Par contre, si une colonne clé étrangère ne doit pas recevoir la valeur vide, alors il faut le préciser dans la description de colonnes.

1.2.4.7 Modèle physique de données

Un modèle physique de données est l'implémentation particulière du modèle logique de données par un logiciel.

La traduction d'un MLD relationnel en un modèle physique est la création (par des requêtes SQL de type CREATE TABLE et ADD CONSTRAINT) d'une base de données hébergées par un SGBD relationnel particulier. Il peut s'agir d'une base Oracle, d'une base SQL Server, d'une base Access ou d'une base DB2, par exemple. Le fait que tous les SGBDR reposent sur le même modèle logique (le schéma relationnel) permet à la fois la communication entre des bases hétérogènes et la conversion d'une base de données d'une SGBDR à l'autre.

1.3 Conclusion

Dans ce chapitre, nous avons analysé tous les informations concernant à l'Internet. L'étude de la conception de bases de données utilisait la method de Merise.

CHAPITRE 2
LE SYSTEME DU CLIENT / SERVEUR ET SECURITE

2.1 Le système client /serveur

2.1.1 Définition

2.1.1.1 Client

Le client est un ordinateur qui veut des ressources aux autres ordinateurs.

Ces ressources peuvent être:

- Espace disque ;
- Informations ;
- Bases de données ;
- Accès à des périphériques (imprimantes, modems) ;
- Puissance de calcul ;
- Courrier électronique ;
- Sauvegarde centralisée ;
- Traitements automatisés.

2.1.1.2 Serveur

Un serveur est un ordinateur qui met ses ressources à la disposition d'autres ordinateurs sous la forme de services.

2.1.2 Fonctionnement

De nombreuses applications fonctionnent selon un environnement client/serveur, notamment le réseau local et surtout l'internet, dans lequel des machines clients contactent un serveur, une généralement très puissante en termes de capacité d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que des pages web, des fichiers, du courrier électronique, et bien d'autres.

Le système client/serveur fonctionne selon le schéma ci-dessous [2]:

- Le client émet une requête vers le serveur grâce à son adresse (éventuellement le port qui désigne un service particulière du serveur) ;
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine client et de son port.

Voici le schéma du fonctionnement du système client/serveur:

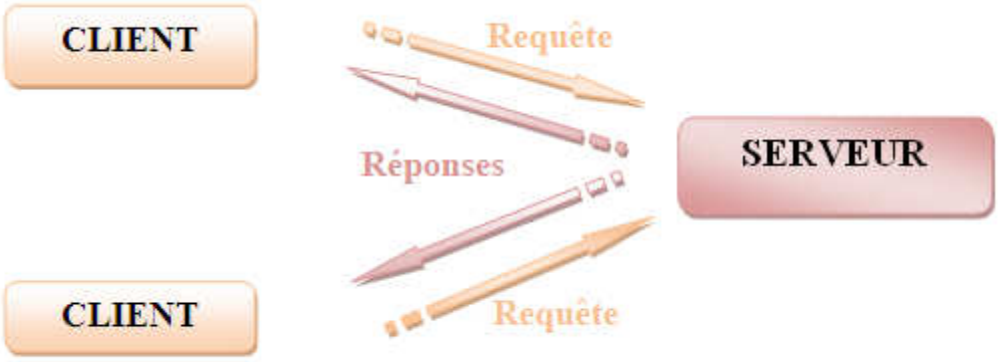

Figure 2.01: *Fonctionnement client/serveur*

2.1.3 Présentation de l'architecture client/serveur :

Le système client/serveur peut encore avoir différent architecture, telles que les architecture:

- 2-tiers ;
- 3-tiers ;
- Et parfois n-tiers.

2.1.3.1 L'architecture 2-tiers

L'architecture 2-tiers ou l'architecture à 2 niveaux signifie que les systèmes client/serveur dans laquelle le client envoie une requête au serveur et ce dernier répond directement sans appel à une autre application pour fournir le service [2].

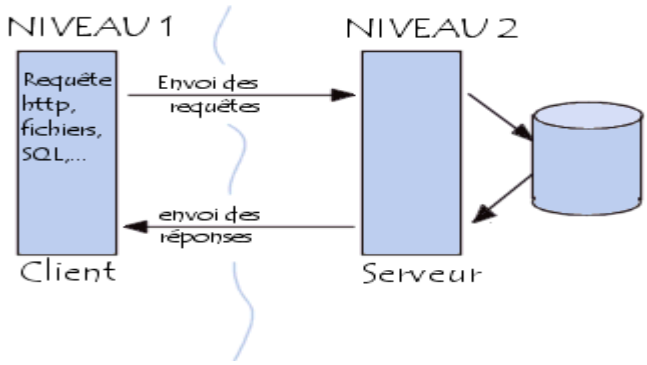

Figure 2.02: *Architecture à 2 niveaux*

2.1.3.2 L'architecture 3-tiers

Dans une architecture à 3 niveaux ou 2-tiers, on distingue:

- Le client : demandeur de ressources ;
- Le serveur d'application ou middleware : le serveur chargé de fournir mais faisant appel à un autre serveur ;
- Le serveur secondaire ou généralement un serveur de base de données fournissant un service au premier serveur.

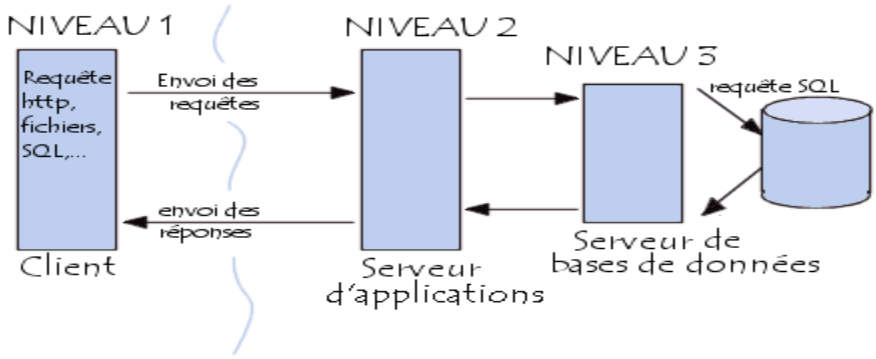

Figure 2.03 : *Architecture à 3 niveaux*

2.1.3.3 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.

2.1.3.4 L'architecture à n-tiers

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 [2].

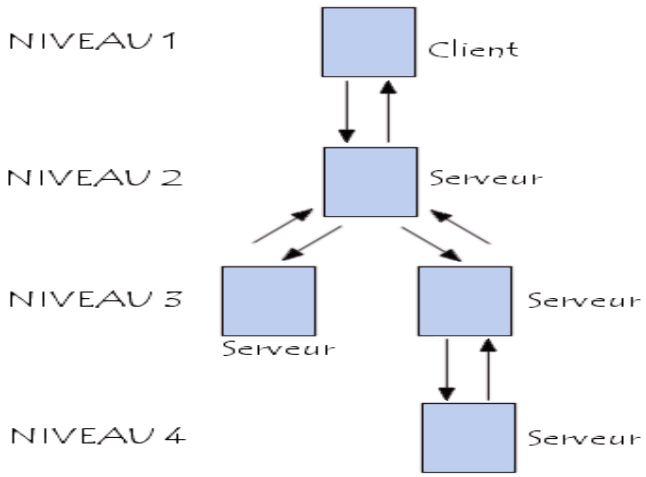

Figure 2.04: *Architecture à n-tiers*

2.1.3.5 Avantages et inconvénients du système client/serveur

a. Avantages

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 peu supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modifications majeures.

b. Inconvénient

- Un coût élevé dû à la technicité du serveur ;
- Sensible à l'attaque extérieure.

2.1.4 Notion sur HTTPS ou HTTP Secure

2.1.4.1 Définition

C'est un protocole issu de *Netscape* et lié à une connexion par socket sécurisée, autrement dit du HTTP avec une pincée de SSL (*Secure Socket Layer*). Sa principale utilisation est le commerce électronique.

Le protocole HTTPS utilise le port 443 par défaut et son rôle est de crypter la communication entre le client et le serveur. (Suis-je bien "connecté" au serveur auquel je pense ?). Pour cela, il est nécessaire d'établir une session SSL avant la connexion HTTP, l'utilisation du protocole HTTP est ensuite complètement identique.

2.1.4.2 Principe

Habituellement, les accès à des pages WEB se font à l'aide du protocole HTTP, en empruntant le réseau Internet. Aucune garantie de confidentialité n'est assurée lors des accès à des données soumises à authentification (échange de login - mot de passe) dans le cadre d'applications de commerce électronique, par exemple. Il faut savoir que, dans ce cas, il n'est pas très difficile à un pirate d'intercepter ces informations confidentielles, y compris le mot de passe du client, et ainsi

d'usurper son identité, voire récupérer son code de carte bleue. Afin de palier à ces inconvénients, le protocole HTTPS peut être mis en œuvre.

En outre, on n'a pas une certitude absolue d'être en cours de consultation du site auquel on croit être connecté.

D'une manière très schématique, HTTPS permet d'encapsuler et de crypter le trafic HTTP; ainsi, il sera quasiment impossible à un pirate qui intercepterait des accès à des pages chargées via le protocole HTTPS, de décrypter cet échange, et donc de récupérer des informations confidentielles. De surcroît, HTTPS permet de s'assurer que le serveur auquel on accède est bien celui que l'on croit.

HTTPS offre d'autres possibilités qui ne sont pas abordées ici (par exemple, authentifier la personne qui accède au serveur). Sans entrer dans du détail technique, les échanges HTTPS sont cryptés et décryptés à l'aide d'un couple de « clés informatiques » qui sont propres à un serveur :

- La clé privée, qui n'est connue que de ce serveur ;
- La clé publique qui est connue du monde entier.

Le navigateur qui accède à un serveur doit récupérer la clé publique de ce serveur; celle-ci lui est transmise depuis le serveur, encapsulée dans un certificat X509 (c'est un fichier informatique). Ce certificat contient donc la clé publique du serveur, validée ("signée") par un organisme reconnu, appelé *Certificate Authority* (CA).

2.2 Le protocole SSL

2.2.1 Présentation du protocole SSL

2.2.1.1 Définition

SSL ou Socket Secure Layer, est un procédé de sécurisation des transactions effectuées via Internet. Le standard SSL a été mis au point par Netscape, en collaboration avec MasterCard, Bank of America, MCI et Silicon Graphics. Il repose sur un procédé de cryptographie par clef publique afin de garantir la sécurité de la transmission de données sur internet. Son principe consiste à établir un canal de communication sécurisé (chiffré) entre deux machines (un client et un serveur) après une étape d'authentification [13].

2.2.1.2 Positionnement du protocole sur les couches d'OSI

SSL se situe au sommet de la couche TCP/IP, et au-dessous de la couche d'application.

Pour mettre en place une connexion SSL, il faut d'abord établir une connexion TCP/IP, car SSL utilise certaines "primitives" de TCP/IP.

Ainsi SSL peut être vu comme un canal sûr au sein de TCP/IP, où tout le trafic entre deux applications "peer to peer" est échangé de manière cryptée. Tous les appels de la couche d'application à la couche TCP, sont remplacés par des appels de l'application à SSL, et c'est SSL qui se charge des communications avec TCP [13].

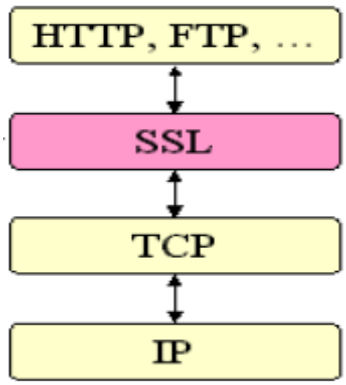

Figure 2.05: *SSL selon modèle OSI*

2.2.2 Service offert par SSL

L'objectif de SSL est de vérifier l'identité des parties impliquées dans une transaction sécurisée et de s'assurer que les données échangées sont protégées de toute interception ou modification.

Les principales fonctions assurées par SSL sont décrites ci-dessous [14].

- Authentification ;
- Confidentialité ;
- Intégrités.

2.2.2.1 Authentification

Dans SSL v3.0 l'authentification du serveur est obligatoire. Elle a lieu à l'ouverture de la session. Elle emploie pour cela des certificats conformes à la recommandation X.509 v3. Cela permet au

client de s'assurer de l'identité du serveur avant tout échange de données. Dans la version actuelle de SSL l'authentification du client reste facultative.

2.2.2.2 Confidentialité

Elle est assurée par des algorithmes de chiffrement symétriques. Bien que le même algorithme soit utilisé par les deux parties chacune possède sa propre clé secrète qu'elle partage avec l'autre. Les algorithmes utilisés sont: le DES, le 3DES, le RC2, le RC4.

2.2.2.3 Intégrités

Elle est assurée par l'application d'un algorithme de hachage aux données (SHA ou MD5) transmises. L'algorithme génère à partir des données et d'une clé secrète appelée code d'authentification de message, une suite de bits. Cette suite sert de signature pour les données. Ainsi tout changement appliqué aux données implique un changement de la suite de bits générée par l'algorithme de hachage et en conséquence provoquera la génération d'un message d'erreur côté récepteur du fait que les deux suites sont différentes.

2.2.3 Système de sécurisations utilisé par SSL

2.2.3.1 Système de chiffrement asymétrique

Le principe de chiffrement asymétrique ou chiffrement à clés publiques est apparu en 1976, avec la publication d'un ouvrage sur la cryptographie par Whitfield Diffie et Martin Hellman

Dans un cryptosystème asymétrique cryptosystème à clés publiques, les clés existent par paires [15]:

- Une clé publique pour le chiffrement ;
- Une clé secrète pour le déchiffrement.

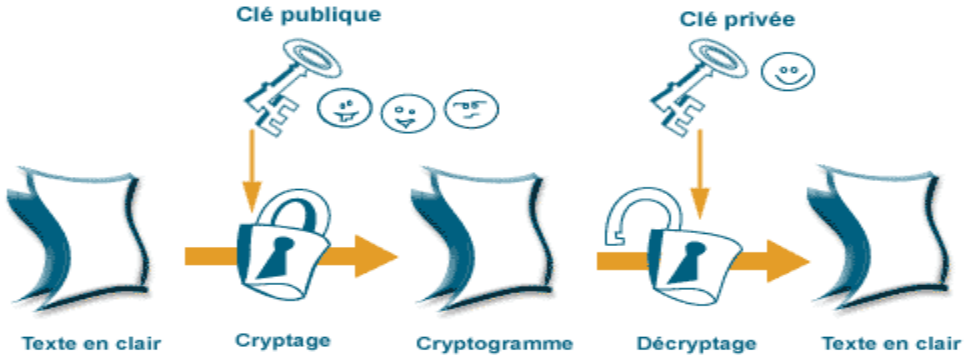

Figure 2.06 : *Cryptographie asymétrique*

Voici quelques exemples des algorithmes des clés publiques :

- RSA: c'est un système publié par R. Rivest, A. Shamir et M. Adleman en 1978. Ce système est élégant, raisonnablement efficace. C'est le plus célèbre et le plus répandu des systèmes de chiffrement à clé publique. Il est implémenté dans les navigateurs web comme Netscape Navigator et Microsoft Internet Explorer ou encore dans certaines cartes à puces comme les cartes VISA;
- Diffie-Hellman: Cet algorithme est le premier algorithme à clé publique inventé par Whitfield Diffie et Martin Hellman en 1976, et sa sécurité repose sur la difficulté de son algorithme mathématique (logaritme discret). Ce protocole sert à convenir d'une clé aléatoire commune sans la révéler en clair sur le canal de communication et sans connaissance préalable d'un quelconque secret commun;
- DSS: c'est le Digital Signature Standard conçu par la NSA (National Sécurity Agency). Il permet d'effectuer des signatures numériques. Il est plus lent que RSA mais a été conçu parce que RSA était protégé par un brevet. Il est aussi référencé sous le nom de DSA (Digital Signature Algorithm).

2.2.3.2 Système de chiffrement symétrique

Le chiffrement symétrique ou chiffrement à clé privée ou chiffrement à clé secrète consiste à utiliser la même clé pour le chiffrement et le déchiffrement.

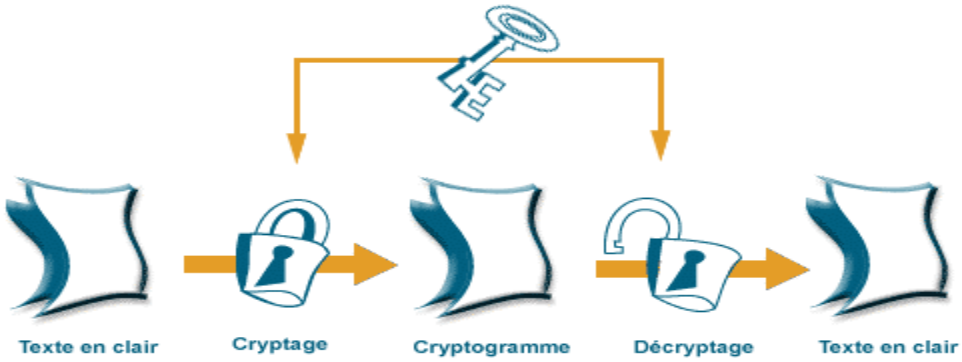

Figure 2.07: *Cryptographie symétrique*

Exemples des algorithmes symétriques:

- DES: Data Encryption Standard » conçu par IBM et la NSA en 1977 à partir du système Lucifer. Il utilise des clés de 56 bits et chiffre des messages de 64 bits;
- Triple DES: 3DES est une combinaison de trois DES, qui opère également sur des messages de 64 bits, mais qui fournit une sécurité supérieure (au prix d'un triplement du temps de chiffrement). La sécurité est équivalente à celle d'une clé de 112 bits (bien que la clé elle-même fasse 168 bits dans certaines variantes);
- IDEA: IDEA a été créé par X. Lai et J. L. Massey en 1992 ; il utilise des clés de 128 bits et chiffre des blocs de 64 bits. Il est breveté et nécessite une license pour une utilisation commerciale. IDEA est utilisé traditionnellement dans PGP (Pretty Good Privacy);
- RC5: c'est un système de chiffrement hautement paramétrable conçu par Ron Rivest. On peut régler la taille des blocs, la taille de la clé. Habituellement, les blocs font 64 bits, la clé aussi;
- AES: est le futur standard de chiffrement américain. Il a été conçu par Joan Daemen et Vincent Rijmen. L' « Advanced Encryption Standard » est actuellement en cours de mise au point qui devrait se terminer au début du troisième millénaire. Il s'agira d'un système de chiffrement de blocs de 128 bits avec des clés de 128 à 256 bits.

2.2.3.3 Système de signature cryptographique

La signature électronique ou numérique est un mécanisme permettant lors d'échanges sécurisés d'assurer l'authentification, l'intégrité des données transmis et la non-répudation de la transaction

ou l'émetteur d'un message ne puisse pas nier l'avoir envoyé et le récepteur l'avoir reçu. Les transactions commerciales ont absolument besoin de cette fonction.

Pour générer une signature électronique, il faut dans un premier temps utiliser une fonction de hachage. C'est une fonction mathématique qui à partir d'un texte de n'importe quelle longueur génère un nombre, suite de bits de taille fixe, bien inférieure à la taille du texte initial. Cette fonction est telle que si un bit du texte d'origine est modifié, le résultat de la fonction sera différent. Cette suite de bits est ainsi appelée condensé ou empreinte [15].

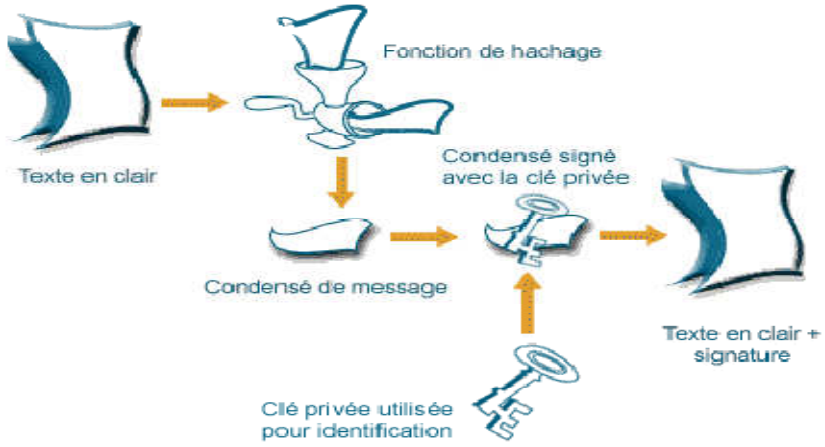

Figure 2.08: *Création d'une signature numérique*

Voici deux exemples de l'algorithme de hachage:

- MD5: MD5 (MD pour Message Digest) est une fonction de hachage très répandue développée par Ronald Rivest, elle crée une empreinte fixe de 128 bits;
- SHA: Secure Hash Algorithm crée des empreintes de 160bits.

2.2.3.4 Les certificats

a. Définition

Un certificat est un document électronique qui fait correspondre une clé publique avec une entité (entreprise ou organisation), il est validé par une autre autorité de certification. In contient un ensemble de données comme le nom du certificat, son utilisation, des informations identifiant le propriétaire, la clé publique, la date d'expiration et le nom de l'organisme certificateur.

Le passage par une autorité de certification générant le certificat est obligatoire, incontournable pour mise place du système sécurisé d'un commerce électronique.

b. Types de certificats

Il existe typiquement deux types de certificats utilisés avec SSL:

- Pour le client, sert à identifier un utilisateur et contiendra donc des informations sur cet utilisateur ;
- Pour le serveur, le certificat a pour but de l'authentifier et l'organisme qui l'exploite?

c. Type de certificat suivant la signature

Les signatures de certificat peuvent distinguer en deux :

- Le certificat signé par le fournisseur de certificat
- Le certificat autographe

d. Différence entre les deux signatures de certificat

Pour le certificat signé par un fournisseur :

- Reconnaitre automatiquement par les navigateurs ;
- Garantir l'identité de l'organisation qui fournit les pages web au navigateur.

Pour le certificat autographe, il n'est pas accepté automatiquement par les navigateurs web. C'est-à dire tous les cas contraire du celle signé par un fournisseur.

2.2.4 Fonctionnement d'un protocole sécurisé

SSL utilise les fonctionnalités offerte par TCP/IP pour permettre aux couches supérieures d'acceder à un mode [16].

2.2.4.1 Authentification du serveur

Cela permet à un utilisateur d'avoir une confirmation de l'identité du serveur. Par contre, un programme client SSL utilise des méthodes de chiffrement à clé publique pour vérifier si le certificat présent dans la liste de ceux connus par les navigateurs.

2.2.4.2 Authentification du client

La technique est ici extactement la même que pour d'authentification du serveur.

2.2.4.3 Chiffrement des données

Toutes les données issues de l'entité émettrice sont chiffrées et dechiffrées par l'entité réceptrice, ce qui permet également la garantir de confidentialité des données.

2.2.5 Les sous-protocoles SSL

SSL est composé de quatre sous-protocoles : Handshake, Record, ChangeCipherSpec et Alert. Ces sous protocoles assurent les fonctions décrites dans le paragraphe précédent.

- Handshake, assure l'authentification des parties et la négociation des algorithmes de chiffrement et de hachage ;
- Record, assure la protection des données des applications et des messages des autres sous protocoles, en mettant en œuvre les paramètres de sécurité négociés durant la phase de Handshake ;
- ChangeCipherSpec, a pour rôle de signaler à la couche Record toute modification de paramètres ;
- Alert, a pour fonction de signaler les erreurs survenant dans les messages.

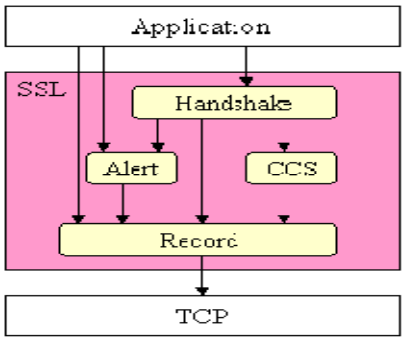

Figure 2.09: *Sous-protocole SSL*

2.2.6 Echange entre SSL et HTTPS

Généralement, lorsqu'un client avec son navigateur se connecte sur un serveur Web sécurisé, ce dernier lui envoie son certificat pour fournir sa clé publique. Si ce certificat a été délivré par une

autorité de certification reconnue par le navigateur du client, il est accepté de manière transparente pour l'utilisateur; sinon le navigateur va demander à l'utilisateur s'il accepte ce type de certificat.

Une clé de session est un algorithme de chiffrement hybride qui allie les performances de la cryptographie symétrique et la souplesse de gestion de clés de la cryptographie à clé publique en utilisant en même temps le chiffrement asymétrique et asymétrique au cours d'un échange.

- Le client envoie au serveur sa version du protocole SSL, ses paramètres de chiffrement, des données générées aléatoirement et d'autres informations dont le serveur a besoin ;
- Le serveur revoie sa version SSL ses paramètres de chiffrement, des données générées aléatoirement et d'autres informations dont le client a besoin. Le serveur envoie également son propre certificat, et si le client demande une information nécessitant un certificat, il demande le certificat client ;
- Le client utilise les informations envoyées par le serveur pour l'authentifier ? si le serveur ne peut pas être authentifié, la connexion n'a pas lieu ;
- Avec les données préalablement échangées, le client est en mesure d'envoyer au serveur un pré clé secrète, qu'il chiffre avec la clé publique du serveur. Si le serveur a requis une authentification du client, ce dernier renvoie également au serveur un bloc de données signé ainsi que son certificat.
- Si le serveur a requis une authentification, il authentifie le client. Le serveur utilise alors sa clé privé de façon à pouvoir déchiffrer le pré clé secrète. Le serveur effectue alors une suite d'actions pour obtenir une clé secrète à partir de pré clé secrète ;
- Le client et le serveur utilisent la clé secrète pour générer des clés de session qui seront les clés symétrique utilisées pour le chiffrement, le déchiffrement des données et l'intégrité ;
- Le client envoie alors un avertissement au serveur le prévenant que les prochains messages seront chiffrés avec la clé de session. Puis il envoie un message chiffré indiquant que la phrase de négociations est terminée.
- Le serveur envoie au client un avertissement que le prochain message est chiffrés avec la clé de session et indique la terminaison de la négociation.
- La phase de la négociation est alors terminée.

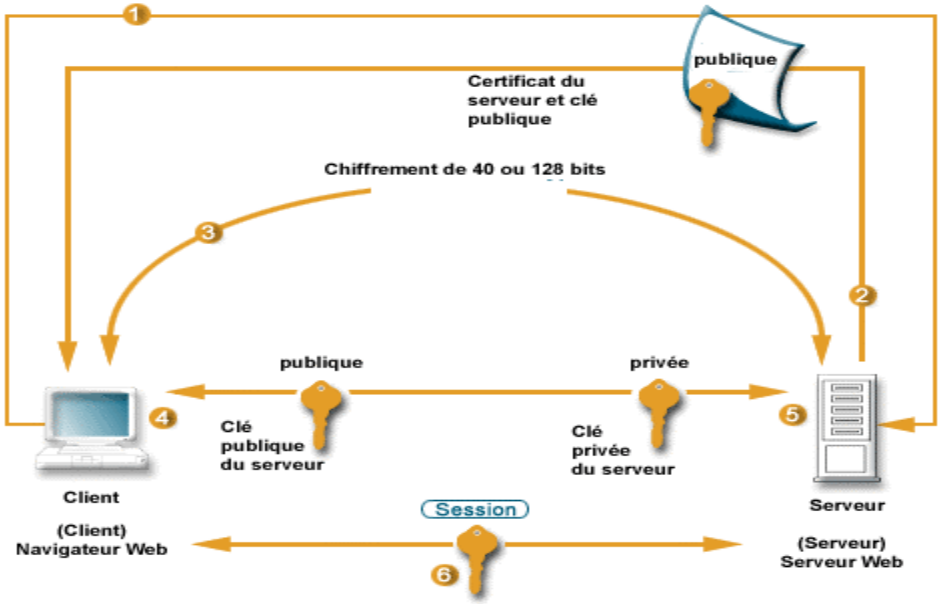

Figure 2.10: *Echange avec https et ssl*

2.3 Conclusion

Ce chapitre nous donne quelques architecture sur le systeme de client et serveur. Pendant les transferts de données entre client et serveur, on utilise le protocole SSL pour sécuriser les données transmis jusqu'au serveur web.

CHAPITRE 3
MISE EN ŒUVRE DE L'APPLICATION SOUS LINUX

3.1 Mise en place du système d'information

3.1.1 Introduction

Notre but est de développer une application fournissant la possibilité de réserver une chambre d'hôtel via la carte bancaire en toute sécurité. Afin de garantir la sécurité de l'application, il s'avère nécessaire de filtrer les données, et de plus l'authentification des clients avant toute transaction.

Voici quelques règles qu'on peut décrire à partir de la description du fonctionnement de ce système d'information :

Règle 1 : Tous les internautes peuvent réserver une chambre.

Règle 2 : Un client qui réserve une chambre doit être client d'une banque.

Règle 3 : Tout visiteur du site peut s'inscrire et devenir membre

Règle 4 : Les renseignements concernant le membre et sa banque doivent être égaux.

Règle 5 : Un membre authentifié obtient une session jusqu'à sa déconnexion.

Règle 6 : Un membre peut faire 0 ou plusieurs réservations.

Règle 7 : Une réservation peut être payée par carte bancaire d'un tarif une nuit.

Règle 8 : Le mode de paiement se fait par carte bancaire.

Règle 9 : Le paiement par carte bancaire comporte:

- Les 16 chiffres apparus sur la carte
- La date d'expiration de la carte

Règle 10 : Toute réservation et transaction doivent être enregistrées

Règle 11 : Le total des tarifs des chambres a réservé doit être inférieur ou égal au solde du compte bancaire.

Règle 12 : Tous messages des clients sont reçus par l'operateur du site.

3.1.2 Accès des visiteurs au site d'application

Tous les visiteurs peuvent être réservés une chambre.

Tous les visiteurs peuvent aussi s'inscrire aux négociations entre client et l'hôtel.

L'inscription est lieu en deux étapes :

1er étape : pour les renseignements des clients

2ème étape : pour les sécurités de votre compte

Toutes visiteurs connectés sur ce site sont automatiquement ajouter dans les listes des membres et ils peuvent directement de faire une réservation.

Après la connexion, les visiteurs peuvent :

- Envoyer un message ou un courrier aux opérateurs d'hôtel ;
- Voir leurs messages envoyer par l'hôtel ;
- Modifier leurs mots de passe de connexion sur ce site.

3.1.3 Accès aux opérateurs d'hôtel

Pour mettre à jour le site, l'opérateur du site hôtel peut aussi accéder au site.

L'opérateur peut voir:

- Toutes les listes d'informations de chambres libre, occupés ou non ;
- Les listes de réservations ;
- Les listes des membres ;
- Les messages envoyés par les membres.

Il peut aussi de :

- Modifier l'état des chambres après la réservation libre ou occupé;
- Modifier les accessoires des chambres ;
- Supprimer une chambre ;
- Répondre les messages des membres.

3.1.4 MCD

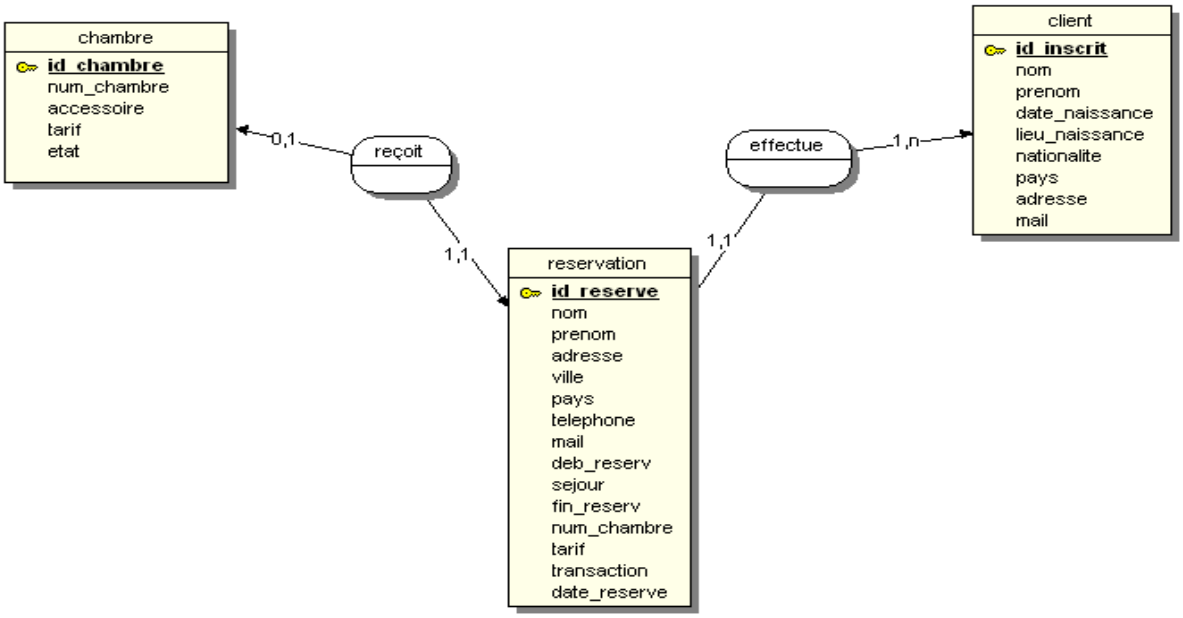

Figure 3.01 : *Modèle conceptuel des données*

Significations :

- Une chambre reçoit au moins 0 réservation et au plus 1 réservation (0,1) ;
- Une réservation est reçu par au moins une chambre et au plus une chambre (1,1) ;
- Une réservation est effectuée par au moins 1 client et au plus 1 client (1,1);
- Un client effectue au moins une réservation et au plus n réservations (1, N).

3.1.5 MLD

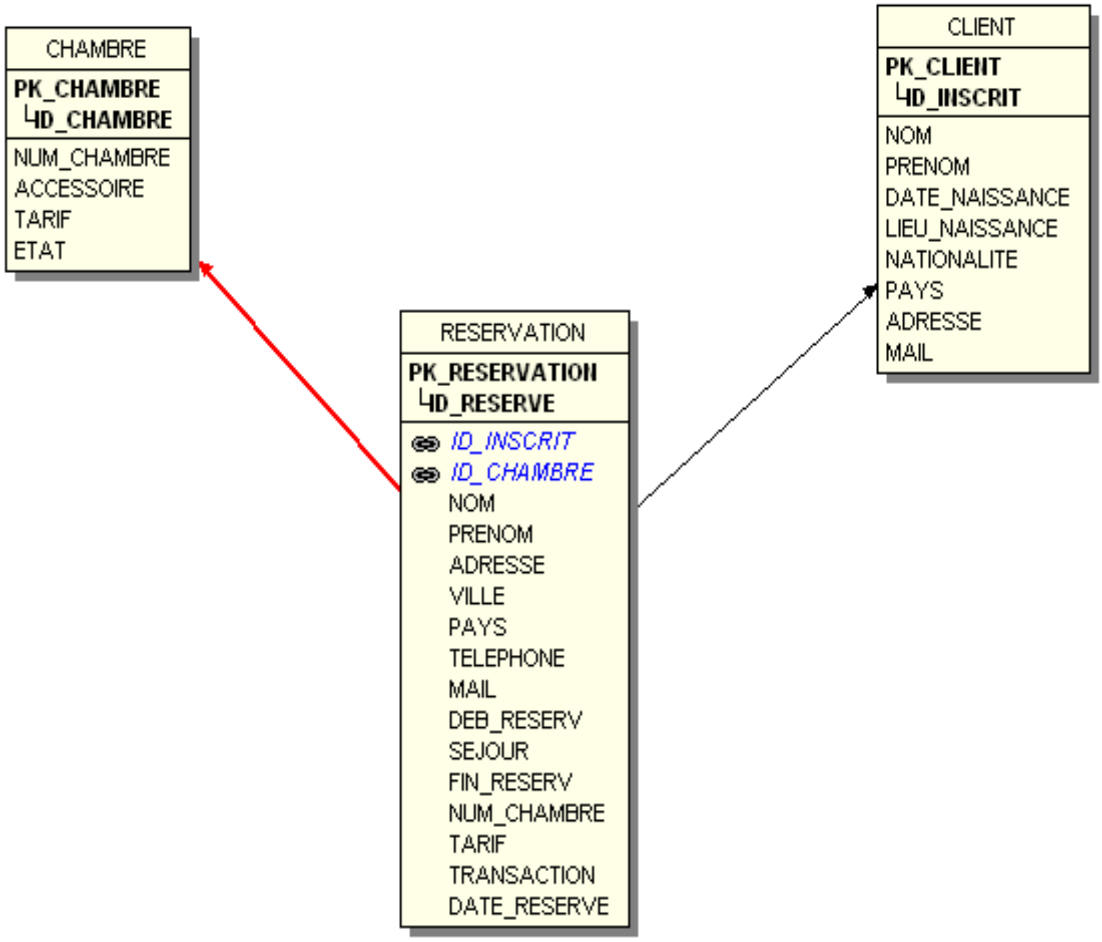

Figure 3.02: *schéma relationnel de données*

3.1.6 MPD

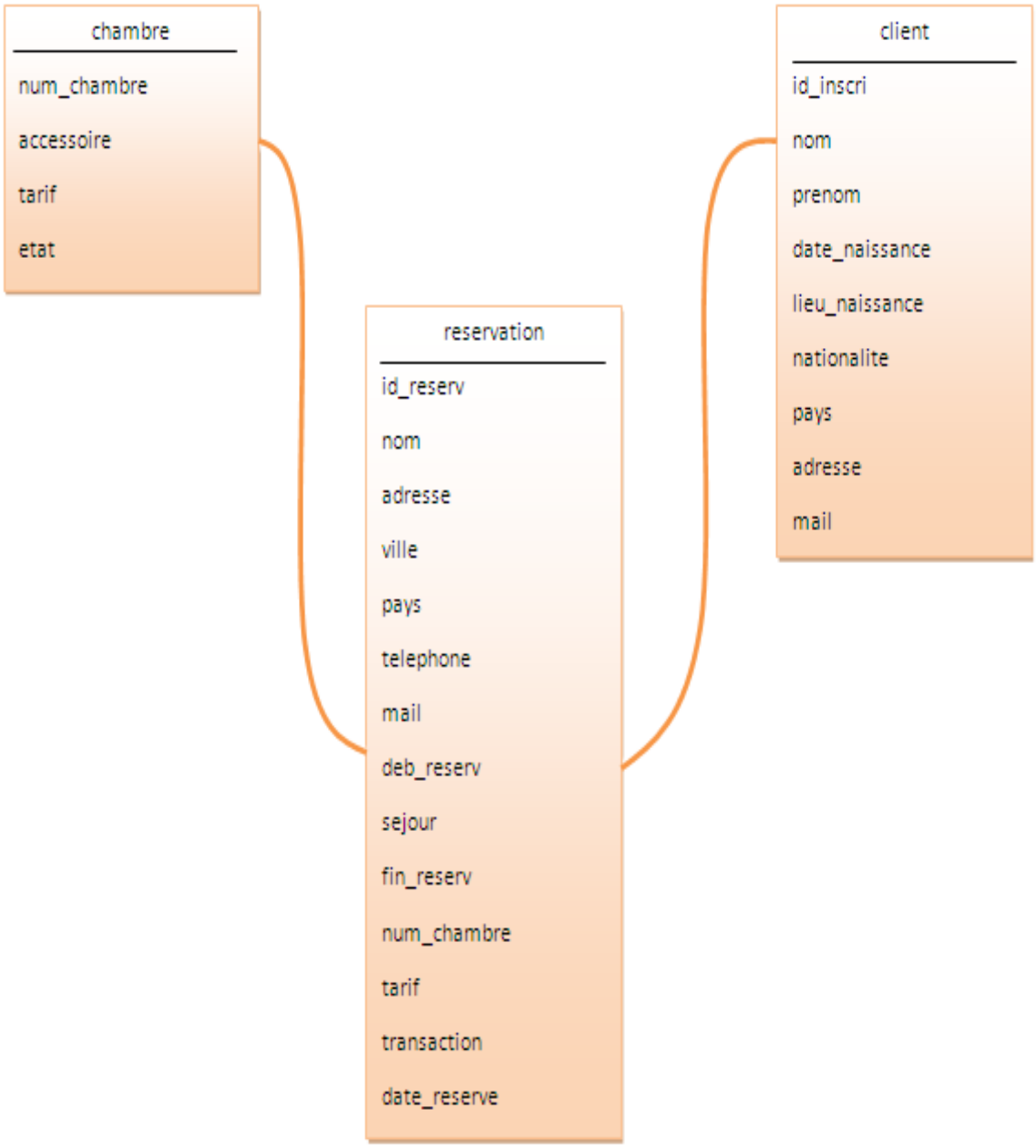

Figure 3.03: *Modèle physique de données*

3.2 Installation et configuration d'un serveur web apache non sécurisé

3.2.1 Système d'exploitation Linux

3.2.1.1 Historique

1969 : Ken Thompson et Dennis Ritchie écrivent une première version du noyau d'un système d'exploitation pour les laboratoires BELL, le nomme UNIX.

Le même Dennis Ritchie invente en 1973 un langage de programmation nommé le langage C. Peu de temps après, Thompson et Ritchie ré-écrivent le noyau d'UNIX en C "rompant avec une tradition qui voulait que le noyau d'un système d'exploitation soit écrit en langage assembleur.

3.2.1.2 Description

Voici les points communs des systèmes UNIX :

- systèmes d'exploitation Multitâche Multi-utilisateurs ;
- écrit en C ;
- materiel: n'importe quel 486 et 8 M de RAM.

3.2.1.3 Ses avantages et ses inconvénients

Ses avantages : toutes les sources sont fournies, il est libre, très stable, très réactif (la détection d'un bug est très rapidement corrigée), il est gratuit, compatible Posix, c'est un challenger et une alternative très sérieuse en ce qui concerne les serveurs, il est très lié au hardware (on peut recompiler les programmes pour les adapter parfaitement à sa machine), il est multi-plate-forme (Alpha, PowerPC, Macintosh, IBM BULL), et enfin il offre des performances très impressionnantes.

Ses inconvénients : Il est austère (il faut aimer la ligne de commande), touffu (il faut plusieurs années pour vraiment avoir une vue d'ensemble d'un système UNIX en général), pas ou peu supports techniques, il n'y a pas tous les drivers, et enfin qu'en est il de la pérennité du produit.

3.2.1.4 Les possibilités serveurs de Linux

Linux en tant que serveur Intranet / Internet peut devenir l'ensemble des solutions suivantes et il est bien entendu possible qu'un seul et même ordinateur gère toutes ces possibilités:

- un serveur WEB classique (HTTP) ;
- un serveur FTP ;
- un serveur de mail (SMTP, POP) ;
- un serveur Proxy ;
- un Firewall ;
- un serveur DNS ;
- un routeur

Linux peut gérer un réseau d'entreprise, comme :

- un serveur de fichiers ;
- un serveur d'impression ;
- un serveur de fax ;
- un serveur de connexion Dial−Up (permet de devenir fournisseur d'accès à Internet)
- un serveur de partage de connexion ;
- un serveur de sauvegarde

Pour transformer un serveur Linux en serveur de base de données, il suffit de coupler le logiciel de base de données (comme MySQL) avec le serveur Web Apache via un langage comme PHP. Un simple navigateur Web suffit alors pour accéder à l'application voulue, ce qui permet d'alimenter et de consulter très facilement des bases de données.

3.2.2 Linux et le serveur Apache

Le serveur Web Apache propose une qualité de service que peu d'offres commerciales peuvent concurrencer, preuve en est la formidable part de marché de cette solution. En Janvier 2002, Apache représente 62% des serveurs Web dans le monde contre 27% pour Microsoft Internet Information Server.

Apache tourne sur Unix, que ce soit Linux ou un UNIX BSD, ainsi que sur WindowsNT, W2K, et WXP. Plus d'informations sur le site d'Apache (http://httpd.apache.org). Une nette majorité des serveurs web tournent sous Unix (dont une bonne partie sous Linux), pour des raisons de performance et surtout de fiabilité.

Le serveur Web Apache peut être utilisé comme simple serveur web, ou bien comme serveur d'application et interface de base de données avec les logiciels PHP et MySQL.

De plus utiliser des logiciels libres, par opposition à des logiciels payants d'origine US, est d'une part nettement moins cher, et un moyen de préserver l'indépendance technologique des pays.

3.2.3 Installation d'Apache

Il existe trois méthodes principales d'installation d'Apache sous une machine Linux [22].

3.2.3.1 Pendant l'installation du système

Cette manière est la plus simple et la plus aboutie. Il s'agit d'installer Apache pendant l'installation même du système.

Elle présente plusieurs avantages comparativement aux deux autres. L'un de ces plus grands avantages est que le programme d'installation du système se charge de toute la configuration de base, en intégrant surtout les différents modules à prendre en charge, par exemple le module d'interprétation des scripts PHP et Perl. Pour installer Apache de cette manière, il suffit de cocher la case Serveurs Web, dans le choix des paquetages à installer, et vérifier dans les détails, que le paquetage httpd, au moins, (sinon httpd-devel est aussi requis) est bien coché. Et bien entendu, pour que les modules d'interprétations suscités soient pris en compte, il faut aussi cocher leurs paquetages d'installations.

Cependant, cette manière n'est valable que lorsque vous êtes aux premières mises en place de votre système. Il ne s'agit pas alors de réinstaller votre OS, auxquels cas vous perdrez vos configurations précédentes avec tout le reste.

Pour alors pallier ce cas, nous avons deux autres manières, qui, en fait, sont les plus standard.

3.2.3.2 Installation avec les paquetages

Il s'agit ici d'installer le paquet compilé d'Apache. Ce paquet est disponible sur les DVDs du système. Mais, il est aussi téléchargeable sur différents sites des distributions. Cependant, la source la plus conseillée est celle des CDs d'installation, car les paquetages se trouvant sur les DVDs sont directement liés aux contraintes de version et de compatibilité de leur distribution, ce qui n'est pas toujours le cas avec les installables téléchargés.

Sous Debian, les paquetages sont d'extension **.deb** . La méthodologie d'installation est analogue à celle sous Fedora.

Mais Le paquetage d'Apache y est désigné par **apache2.deb** (La version 2 d'Apache).

En mode graphique, un double-clique sur le paquet lance l'installation, ou encore se servir du gestionnaire de paquets synaptics.

En mode texte, exécuter la commande **`apt-get install apache2`**. L'exécution de cette commande va rechercher les paquets requis, construire les dépendances et lancer l'installation.

Pour un paquet construit manuellement (par téléchargement, par exemple,), il faut installer avec la commande **`dpkg -i /chemin/vers/apache2.deb.`**

3.2.3.3 Compiler les fichiers sources

Les sources sont compressées sous les extensions **.tar** et / ou **.gz**. C'est en fait, les différents codes sources des applications qui régissent le fonctionnement d'Apache. Ils sont téléchargeables sur Internet. Il va donc falloir les configurer, les compiler puis les installer.

Pour cela:

- savoir préalablement ce qu'on veut faire, car c'est délicat de procéder ainsi. Il faut disposer d'un compilateur C comme **gcc**, par exemple, pour y arriver.
- se positionner dans le répertoire **/etc/httpd/**: **cd /etc/httpd/**
- décompresser le paquetage dans ce répertoire: **tar zxvf nom_paquetage.tar.gz**
- entrer dans le nouveau répertoire, crée après décompression: **cd nouveau_rep** (nouveau_rep peut être **apacheversion** ou **httpd-version**)
- configurer l'ensemble: **`./configure --prefix=$DEST --enable-module=most --enable-mods-shared=all`** **`$DEST`** désigne la racine d'installation (les binaires seront copiés sur **`$DEST/bin`** lors du make **`install`**, les fichiers de configuration vers **`$DEST`*`/etc`***, etc.). ***`$DEST`*** peut être, par exemple, **/usr/local/**.
- compiler le tout: **`make`**
- faire l'installation de la compilation: **`make install`**

Pour tester l'installation d'Apache, on lance **http://127.0.0.1** ou **http://localhost** dans un navigateur après avoir démarré le service correspondant.

Voici la commande pour démarrer apache sous Debian :

```
/etc/init.d/apach2 reload
```

Ensuite, on verra un message d'accueil « it works ».

3.2.4 Configuration

3.2.4.1 Fichier de configuration

Le nom donné au fichier de configuration varie selon les distributions Linux. Il est généralement désigné par **httpd.conf** (sous Fedora, Redhat) ou par **apache2.conf** (sous Debian, Mandrake).

Il est situé dans le répertoire **`/etc/httpd2/conf`** ou **`/etc/apache2/conf`**, selon les distributions [20] [21].

Pour faire la configuration, il est conseillé de se "loguer" en tant qu'utilisateur et non en tant que root, afin de prévoir tous désagréments dus aux erreurs indésirées. Il faudra donc accorder les droits de lecture et d'écriture aux utilisateurs requis pour la configuration. On pourra le leur enlever, une fois la configuration terminée.

Pour accorder les droits: `chmod 777 /etc/httpd/conf/httpd.conf`

Pour enlever les droits: `chmod 644 /etc/httpd/conf/httpd.conf`: c'est suffisant pour assurer la sécurité de notre serveur, du moment où il faut permettre aux utilisateurs de le faire fonctionner.

Pour faire plus simple, il est possible d'utiliser la commande **sudo** avec la syntaxe suivante:

sudo nom-de-lacommande-à-éxécuter

password: taper-le-mot-de-passe-du-root

3.2.4.2 Directives de configuration les plus utiles

La configuration d'Apache prend en compte beaucoup d'éléments qui peuvent être regroupés en trois grandes sections. Mais, pour chaque section, présentons les éléments les plus essentiels. Par souci de cohérence, le paramétrage doit se faire dans l'ordre indiqué.

Il convient aussi de remarquer le fichier de configuration d'Apache comporte des commentaires, qui présentent et explicitent brièvement en anglais les différentes directives de configuration.

A l'intérieur du fichier de configuration **httpd.conf,** on trouvera les principales variables que sont :

ServerName www.gashotel.com: Nom du serveur. Il ne se s'agit pas du nom du serveur pour lequel Apache répond mais du nom avec lequel Apache envoie sa réponse.

BindAdress 192.168.0.1: Adresse IP du serveur

Port 80: Port à écouter.

Il est possible de remplacer les configurations **Port** et **BindAdress** par la directive:

Listen 192.168.0.1 : 80, l'avantage est qu'il est possible de spécifier plusieurs fois la directive **Listen**, ce qui n'est pas les cas pour **BindAdress.**

ServerType standalone : Apache est autonome. **ServerType inetd :** Apache n'est lancé que lorsque **inetd** reçoit une requête sur les ports pour lequel il est configuré.

Utilisateur et groupe : dans le but de rendre Apache moins vulnérable aux éventuelles attaques, il possible de le configurer pou qu'il s'exécute sous utilisateur et un groupe ayant des droits restreints.

UserBody

GroupBody

ServerAdmin web@gashotel.com : adresse électronique de l'administrateur

ServerRoot/etc/httpd: racine du serveur

ErrorLog logs/error_log: journal d'erreur par défaut

DocumentRoot /var/www/html : emplacement par défaut des pages html

DirectoryIndex index.htm index.html index.php index.xhtml: pages par défaut

3.2.4.3 Hôtes virtuels

Les hôtes virtuels sont des identifiants utilisés pour indiquer les différents sites hébergés sur un serveur. Cela permet l'hébergement de plusieurs sites Web sur un seul serveur.

Ce qu'il faut surtout comprendre, c'est que le virtualhost est en fait, un serveur virtuel qui est crée sur le serveur réel.

Le site web porte donc le nom de ce serveur virtuel. A cela, il faut aussi ajouter que le virtualhost peut aussi porter sur le port du serveur web.

NameVirtualHost *: 80>

<VirtualHost *:80/>

ServerAdmin root@localhost>

DocumentRoot /var/www/domaine/site1>

ServerName www.site1.domaine>

ErrorLog logs/www.site1.domaine_log

CustomLog /www.site1.domaine_log common

<VirtualHostr/>

3.2.4.4 Arrêt et demarrage d'apache

Systèmes	Démarrage du service	Arrêt	Etat du service
Redhat	Service httpd Start	Service httpd stop	Service httpd status
Mandrake	/etc/init.d/httpd stop	/etc/init.d/httpd stop	Etc/init.d/httpd status
Debian	/etc/init.d/httpd start	/etc/init.d/http stop	/etc/init.d/httpd status

Tableau 3.01 : *Arrêt et demarrage d'Apache*

3.3 Installation d'un serveur web sécurisé

Le Web est le principal moyen de partager des informations de nos jours. Nous allons mettre en place un serveur HTTP (apache) gérant le PHP5 puis nous allons le sécuriser avec OpenSSL. Enfin, nous allons installer une base de données MySQL et l'outil PhpMyAdmin pour la gérer.

3.3.1 Installation des différents paquets

3.3.1.1 Installation d'OpenSSL

OpenSSL est une boite à outils de chiffrement comportant deux bibliothèques (une de cryptographie générale et une implémentant le protocole SSL) [22].

On l'installe avec la commande suivante:

```
apt-get install openssl
```

3.3.1.2 Installation de Mysql

MySQL est un système de gestion de base de données (SGBD). Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle ou Microsoft SQL Server. On l'installe avec la commande suivante :

```
apt-get install mysql-server
```

3.3.1.3 Installation de PHP5

PHP, est un langage de scripts libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP. On l'installe avec la commande suivante (noyau et interface avec MySQL) [11]:

```
apt-get install php5 php5-mysql
```

Grâce aux dépendances, l'installation de ce paquet entraînera l'installation du serveur HTTP Apache 2 qui permettra la communication entre le client web et PHP.

3.3.1.4 Installation su serveur Apache2

Apache HTTP Server est serveur HTTP produit par l'Apache Software Foundation. C'est le serveur HTTP le plus populaire du Web. Si l'installation n'a pas été faite avec PHP5, on tape la commande suivante :

```
apt-get install apache2
```

3.3.1.5 Installation de PhpMyAdmin

On peut installer phpmyadmin pour gérer la base de données à partir d'une interface Web.

La commande suivante installe le logiciel :

```
apt-get install phpmyadmin
```

Lors de l'installation, on sélectionne apache2 comme serveur Web afin d'automatiser l'intégration puis on relance de dernier.

3.3.2 Configuration

3.3.2.1 Création des certificats OpenSSL

On crée les certificats nécessaires à la mise en place du serveur HTTPS avec la commande suivante:

```
openssl req -x509 -nodes -days 365 -newkey rsa:1024 –out
/var/key/server.crt -keyout /var/key/server.key
```

Détail de la commande :

- **x509 -nodes** donne le type de certificat voulu
- **days 365** indique la durée de validité (en jours) du certificat
- **newkey rsa**:1024 demande une clé RSA de 1024 bits
- **out /var/key/server.crt** est le chemin du certificat
- **keyout /var/key/server.key** est le chemin de la clé privée

3.3.2.2 Configuration d'Apache2

On commence par charger le module ssl:

```
a2enmod ssl
```

Par défaut, Apache2 est configuré pour écouter sur le port 80. Cependant, le protocole SSL a besoin d'un port spécifique pour pouvoir fonctionner, le port 443.

Nous allons donc rajouter une directive de configuration nommée Listen qui permettra d'indiquer à Apache2 qu'il doit aussi écouter sur le port 443.

Pour ce faire, on édite le fichier `/etc/apache2/ports.conf` et on rajoute les lignes suivantes, si elles ne sont pas présentes:

```
<IfModule mod_ssl.c>
Listen 443
</IfModule>
```

On crée ensuite le fichier `/etc/apache2/sites-available/site1` et on le remplit avec la configuration suivante:

```
NameVirtualHost 192.168.0.12:443
<VirtualHost *:80>
ServerName site1.com/
Redirect / https://192.168.0.12/
</VirtualHost>
<VirtualHost 192.168.0.12:443>
ServerName site1.com
DocumentRoot /var/www/
SSLEngine on
SSLCertificateFile /var/key/server.crt
SSLCertificateKeyFile /var/key/server.key
</VirtualHost>
```

Si l'utilisateur arrive sur le port 80, on le redirige sur le 443.

On relance ensuite apache.

3.3.3 Test

Lorsque le serveur sécurisé est installé, une clé aléatoire et un certificat générique sont installés à des fins de test. On peut se connecter au serveur sécurisé à l'aide de ce certificat.

Le démarrage d'un serveur web Apache sécurisé sous SSL est avec le démarrage du systeme.

Les pages d'un site protégé par un certificat comportment les caractéristiques suivante:

- L'URL des pages sécurisé commence par **https:**// et non par **http://;**
- Un petit cadenas s'affiche dans la barre d'état du navigateur.

Nous pouvons consulter le certificat et le type de niveaux de cryptage en cliquant sur ce petit cadenas.

3.4 Conclusion

Le système UNIX est le système d'exploitation le plus utilisé à l'Internet. Il est bien sécurisé avec Apache comme le serveur.

CONCLUSION GENERALE

En guise de conclusion, la mise en place d'un système d'information sécurisé sous Linux nous donne de nombreuses connaissances sur l'environnement web ou le développement web, la conception de bases de données et la configuration d'un serveur web non sécurisé ou sécurisé.

Pour créer un système d'information, on doit maitriser quelques langages de développement sur web. Par exemple, le langage HTML, le langage JavaScript, le langage PHP. Dans ce mémoire, nous avons fait la mise en place d'un site web d'hôtel pour que les clients rendent facilement leurs réservations des chambres en ligne.

Cet ouvrage nous rappelons aussi la mise en place d'un serveur web sécurisé sous SSL. SSL est un protocole de sécurisation sur les échanges électroniques. Il offrait ses services pour la confidentialité des données, l'intégrité de données et l'authentification des clients et de serveur grâce à l'utilisation des systèmes de chiffrement asymétrique ou symétrique, un système de signature cryptographique et les certificats. Les configurations du serveur demandent quelques connaissances sur les commandes de base au système d'exploitation Linux ou plus précisément sur Debian.

Votre site web sera bien sécurisé sous le protocole SSL. Vous pouvez faire tranquillement des échanges sur l'internet avec confiance.

ANNEXE 1: E-COMMERCE

A1.1: Definition

Le commerce électronique peut être défini comme l'ensemble des échanges électroniques liés aux activités commerciales. Il recouvre toute opération de vente de biens et de services via un canal électronique.

Internet n'est donc qu'un support parmi d'autres du e-commerce avec, entre autres, l'EDI (échanges des données informatisées), le Minitel (en France) voire même le téléphone (Audiotel) ou la télévision (Pay-Per-View).

Le site commercial, vitrine de l'entreprise sur Internet et lieu de vente, n'est donc qu'un élément du processus de vente en en ligne, au même titre que l'emploi de la messagerie électronique pour optimiser la relation client ou la création d'un extranet pour faciliter les échanges commerciaux avec les différents partenaires et les fournisseurs.

Le commerce électronique offre aux entreprises des perspectives de croissance considérables. Mais, il représente aussi un enjeu majeur, nécessitant des changements profonds en termes d'organisation interne, de fonctionnement et de stratégie.

A1.2: Objet et forme d'e-commerce

La mise en place d'une stratégie d'e-business au sein de l'entreprise offre des avantages dans différents domaines :

- En interne, une optimisation du processus de production (objectifs « juste à temps » et « zéro stock ») et un meilleur partage des connaissances (knowledge management),
- En externe, l'échange rapide d'informations avec les partenaires de l'entreprise (fournisseurs, clients) via des extranets spécialisés, la gestion de la relation client dans une optique one to one et la vente en ligne proprement dite, qui permet de proposer à un public mondial une offre plus importante, mieux présentée, avec des coûts de distribution moindre (notamment grâce au phénomène de désintermédiation).

Sur un plan macroéconomique, le phénomène e-business s'accompagne du développement de trois grands types d'entreprises dont les activités sont liées à Internet :

- Les générateurs de trafic (portails, fournisseurs d'accès), points d'accès généralistes à Internet, qui concentrent la plus grosse part de l'audience sur le Web,
- Les « faciliteurs » (logiciels, service, fonds de capital risque),
- Les sites de commerce électronique, qui se subdivisent en différents domaines en fonction des acteurs impliqués et des produits et services vendus ou échangés.

A1.3: Différentes forme d'e-commerce

B to C : Vente de produits physiques, biens immatériels (informations, vidéos, jeux, logiciels) et de services (réservations, services à domicile).

B to B stratégique : Achats répondant à un besoin professionnel individuel (livres, publications, informations, séminaires). Achats de fonctionnement : fournitures de bureau, consommables, équipement électrique, télécom et informatique.

B to B non-stratégique : Consommations intermédiaires : matière première, produits semi-finis servant à la réalisation d'un produit final. Produits finis achetés pour être distribués au client final.

A1.3: Types de paiement en e-commerce

Paiement en ligne via sa banque :

- la plupart des banques proposent aujourd'hui des terminaux en ligne ;
- paiement via la saisie d'un numéro de carte bancaire sur le site ;
- les transactions sont gérées par la banque comme tout TPE ou Terminal de Paiement Electronique.

Paypal [24]:

- les transactions ne passent plus par un organisme bancaire ;
- tout reste virtuel ;
- vous obtenez l'équivalent d'un compte bancaire chez Paypal :
  - vous pouvez recevoir des paiements
  - et en effectuer

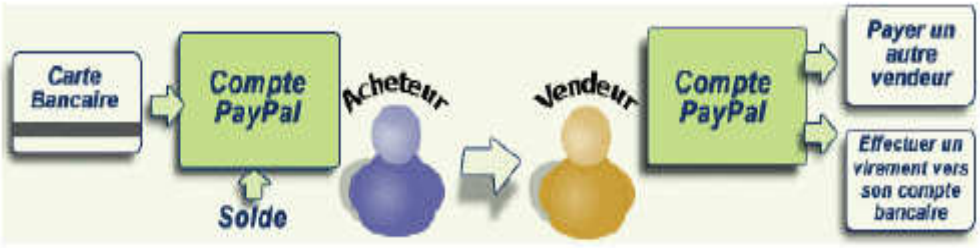

Figure A01 : *Principe de Paypal*

A1.4 : Avantages et inconvénients d'e-commerce

Avantage:

- Il ouvre un nouveau chemin de distribution pour certains produits et services de l'entreprise;
- Il favorise une relation personnelle avec le consommateur ou le client, facilitant la vente;
- Il permet d'envisager la fidélisation du client à travers une offre de services;
- Il facilite les transactions en évitant à l'acheteur de se déplacer;
- La recherche du meilleur prix;
- Un gain de temps;

Inconvenient:

- L'incertitude et le manque de confiance autour de la sécurisation des moyens de paiement;
- Le pistage informatique à partir des cookies, c'est-à-dire ces petits fichiers qui identifient l'ordinateur appelant de façon unique afin pouvoir retracer toutes les habitudes d'appel et de consommation.
- L'insécurité des paiements ;
- Le manque de relations humaines et le sentiment d'isolement devant sa machine.

ANNEXE 2 : RESEAU SOUS GNU/LINUX

En réseau, l'unité de transmission est le paquet [23].

A2.1 Modèle de couche

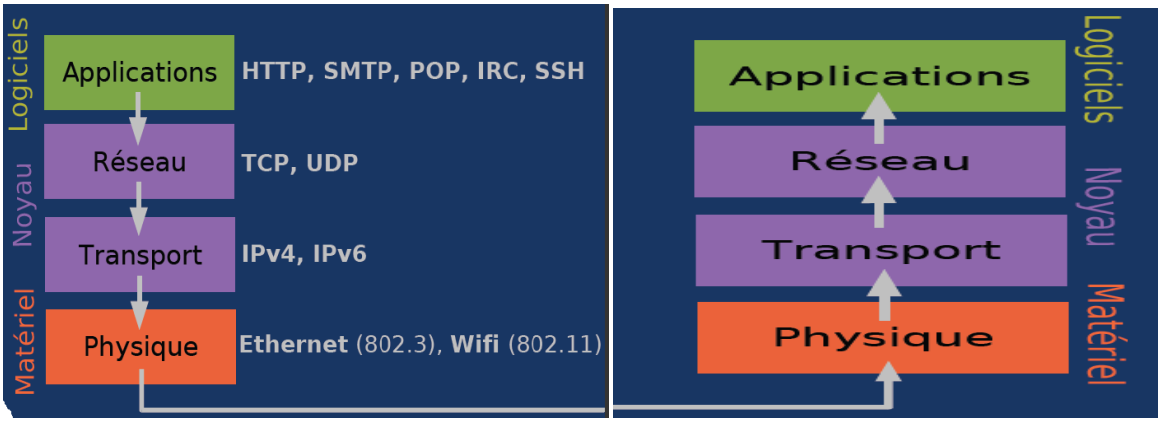

Figure A.02 : *Modèle de couche*

Sur la source, descente dans les couches : **encapsulation**

Sur la destination : remontée dans les couches : **désencapsulation**

A2.2 Adresse MAC :

Adresse **unique** pour chaque interface Ethernet (carte réseau)

- Fixée par le constructeur du matériel ;
- Difficilement modifiable ;
- Utilisée au niveau physique (Ethernet) ;
- 6 octets ;
- MAC = Medium Access Control.

Commande :

```
# ifconfig eth0
eth0 Lien encap:Ethernet HWaddr 00:E0:4C:EC:32:3B
```

A2.3 Adresse IP :

- Utilisées pour structurer un réseau de manière logique ;
- Une ou plusieurs par interface réseau ;

- Sur 4 octets ;
- Utilisées au niveau du protocole IP (couche «Transport») ;
- IP = Internet Protocol.

Commande :

```
# ifconfig eth0
eth0 Lien encap:Ethernet HWaddr 00:E0:4C:EC:32:3B
inet adr:192.168.1.2 Bcast:192.168.1.255 Masque:255.255.255.0
```

A2.4 Relation entre IP et MAC

Trouver la MAC à partir d'une IP : protocole **ARP** ou Address Resolution Protocol

A2.5 Masque de sous-réseau :

Associé à chaque IP, un masque de sous-réseau **:**

- forme : 255.255.255.0
- détermine la « portée » du réseau accessible directement par l'interface
- fonctionne par **ET** logique
- Ex: (192.168.1.2 & 255.255.255.0) = 192.168.1.0
- 255.255.255.0 : 24bits
- 255.255.0.0 : 16bits
- 255.0.0.0 : 8bits

A2.6 Adresse IP spécial :

- 127.0.0.1 : Adresse IP «locale» ;
- 192.168.1.255 : Adresse de diffusion ou «broadcast» ;
- 192.168.1.0 : Adresse «réseau».

Commande :

```
# ifconfig lo
lo Lien encap:Boucle locale
inet adr:127.0.0.1 Masque:255.0.0.0
UP LOOPBACK RUNNING
```

```
# ifconfig eth0
eth0 Lien encap:Ethernet HWaddr 00:E0:4C:EC:32:3B
inet adr:192.168.1.2 Bcast:192.168.1.255 Masque:255.255.255.0
```

A2.7 Adresse IP et internet

- Adresses IP sur 4 octets : 2^{32} adresses disponibles ;
- Adresses uniques sur tout Internet ;
- Sauf classes réservées pour réseaux locaux ;
  - 10.0.0.1 à 10.255.255.254
  - 172.16.0.1 à 172.31.255.254
  - 192.168.0.1 à 192.168.255.254
- Nombre limité d'adresses IPs : utilisation de passerelles.

A2.8 Configuration IP

- directement avec la commande ifconfig pour du statique
  - `ifconfig eth0 192.168.1.32`
- directement avec un client DHCP pour du DHCP
  - `dhclient eth0`
  - `pump -i eth0`
- Debian : dans le fichier **/etc/network/interfaces**

Commande :

```
# /etc/network/interfaces
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.1.1
netmask
255.255.255.0
auto eth1
iface eth1 inet dhcp
```

ANNEXE 3 : CODE SOURCE DE RESERVATION D'UNE CHAMBRE

RESERVE.PHP :

```
<?php include("includes/connexion.php");
      $conn=connexion();
?>
<?php
if(isset($_POST['envoye'])        &&        !empty($_POST['envoye'])        &&
$_POST['envoye']="envoyer")
{
      $nom=$_POST['nom'];
      $prenom=$_POST['prenom'];
      $adresse=$_POST['adresse'];
      $ville=$_POST['ville'];
      $pays=$_POST['pays'];
      $telephone=$_POST['telephone'];
      $mail=$_POST['mail'];
      $debut=$_POST['debut_reserv'];
      $nb_j=$_POST['sejour'];
      $fin=$_POST['fin_reserv'];
      $num_ch=$_POST['genre_id'];
      //$prix=$_POST['prix'];
      $hide=$_POST['hide'];


      $req="select from chambre where num_chambre='".$num_ch."'";
      $res=mysql_query($req);
            while($user=mysql_fetch_array($res))
            {
                $num[]=$user['num_chambre'];
                $acc[]=$user['accessoire'];
                     $tar[]=$user['tarif'];
                    $etat[]=$user['etat'];
                 }
            $v=0;
      for($i=0;$i<sizeof($num);$i=$i+1)
            {
              if($num[$i]==$num_ch && $etat[$i]=='libre') //&& $tar[$i]==$prix
            {
            $num=$num_ch;
```

```
switch ($num)
{
case 21:
{
$prix=42000;
break;
}
case 22:
{
$prix=21000;
break;
}
case 23:
{
$prix=26000;
break;
}
case 24:
{
$prix=31000;
break;
}
case 25:
{
$prix=26000;
break;
}
case 26:
{
$prix=26000;
break;
}
case 27:
{
$prix=31000;
break;
}
case 28:
```

```
{
$prix=42000;
break;
}
case 29:
{
$prix=15000;
break;
}
case 30:
{
$prix=26000;
break;
}
case 31:
{
$prix=21000;
break;
}
case 32:
{
$prix=21000;
break;
}
case 33:
{
$prix=21000;
break;
}
case 34:
{
$prix=26000;
break;
}
case 35:
{
$prix=42000;
break;
```

```
            }
            }
                 {
                 $v=1;
               }
            }
            if($v==1)
            {
      if($hide!=$nom && $hide!=$prenom && $hide!=$adresse && $hide!=$ville && $hide!=$pays && $hide!=$telephone && $hide!=$mail && $hide!=$debut && $hide!=$fin && $hide!=$prix)
      {
      $insert_reserv="insert into facturation(nom,prenom,adresse,ville,pays,telephone,mail,debut_reserv,nb_jour,fin_reserv,num_chambre,tarif) values ('".$nom."','".$prenom."','".$adresse."','".$ville."','".$pays."','".$telephone."','".$mail."','".$debut."','".$nb_j."','".$fin."','".$num_ch."','".$prix."')";
      $query=mysql_query($insert_reserv);
      if($query)
      {
      $sql="update chambre set etat='occupe' where num_chambre='".$num_ch."'";
      $query_ch=mysql_query($sql);
      }
      if($query_ch){
      echo"<script type=\"text/javascript\">
alert(\"Cliquer sur OK pour suivre le paiement de votre reservation!!\");
document.location.href=\"paiement.php\";
</script>";
            //header("Location:paiement.php");
}
            else {
      echo"<script type=\"text/javascript\">
alert(\"Erreur!!\");
document.location.href=\"reservation.php\";
</script>";
      }
      }
      else
      {
```

```
      echo"<script type=\"text/javascript\">
alert(\"Erreur sur les remplissages!!\");
document.location.href=\"reservation0.php\";
</script>";
      //header("Location:reservation0.php");
      }
      }
      else
      {
      echo"<script type=\"text/javascript\">
alert(\"Ce chambre n'est pas disponible pour cette reservation.\");
document.location.href=\"libre.php\";
</script>";
      }
      }
      }
?>
```

ANNEXE 4 : CODE SOURCE DE PAIEMENT

PAYE.PHP :

```
<?php
include("includes/connexion.php");
$conn=connexion();
$sql="select * from facturation";
$data=select($sql);
?>
<?php
for($i=0;$i<sizeof($data);$i++){
if($i%2==0)
{
 $coul="style=\"background:#ffffff;\"";
}
Else
{
$coul="style=\"background:#ffffff;\"";}
$nom=$data[$i]['nom'];
$prenom=$data[$i]['prenom'];
$adresse=$data[$i]['adresse'];
$ville=$data[$i]['ville'];
$pays=$data[$i]['pays'];
$telephone=$data[$i]['telephone'];
$mail=$data[$i]['mail'];
$debut=$data[$i]['debut_reserv'];
$nb_j=$data[$i]['nb_jour'];
$fin=$data[$i]['fin_reserv'];
$num_ch=$data[$i]['num_chambre'];
$prix=$data[$i]['tarif'];
}
// Affichage de la page
if(isset($_POST['envoyer']) && !empty($_POST['envoyer']))
{
      //variable paiement
 $nu_carte=$_POST['num_carte'];
$date_expire=$_POST['date_expire'];
//select banque du client
```

```
mysql_connect("localhost","root","ludovic");
$base=mysql_select_db("banque_client");
$req1="select * from compte_client";
$res1=mysql_query($req1);
while($ban_cli=mysql_fetch_array($res1))
{
 $id_cmpt_cli[]=$ban_cli['id_compte'];
$num_cart_cli[]=$ban_cli['num_cart_ban'];
$date_exp[]=$ban_cli['date_expire'];
$num_compte[]=$ban_cli['num_compte'];
$compte[]=$ban_cli['solde_client'];
$date=date("Y-m-d : h:s");
}
//comparaison des données
$v=0;
for($i=0;$i<sizeof($num_cart_cli);$i++)
{
if($num_cart_cli[$i]==$nu_carte  &&  $date_exp[$i]==$date_expire  &&  $date  < $date_expire )
      {
      $compte_cli=$compte[$i];
      $id_cp_cli=$id_cmpt_cli[$i];
      $v=1;
      }
}
if($v==1)
{
//si même=> test si compte<prix
if($compte_cli<$prix)
      {
      //si ok=> compte insufisant
      echo"<script type=\"text/javascript\">
      {
      alert(\"Vous  ne  pouvez  faire  de  transaction,  votre  compte  est insufisant!!!\");
      document.location=\"confirmation.php\";
     }</script>";
      }
      else
```

```
       {
            //si non=>restant=compte-prix
            $reste=$compte_cli-$prix;


            //update compte client =>restant
      $mod_cmpt_cli="update compte_client set solde_client='".$reste."' where
id_compte='".$id_cp_cli."'";
      $resu=mysql_query($mod_cmpt_cli);
      //si update ok=> update cmpte commercant=>compte=cmpteactuelle+prix
      if($resu)
      {
      //selection du compte commerçant
      mysql_connect("localhost","root","ludovic");
      $base=mysql_select_db("banque_hotel");
      $sel_com="select * from compte_hotel";
      $res_com=mysql_query($sel_com);
      while($ban_comm=mysql_fetch_array($res_com))
      {
      $id_cmpt_com[]=$ban_comm['id_compte'];
                              $num_compte_com[]=$ban_comm['num_compte'];
      $compte_com[]=$ban_comm['solde_hotel'];
      }
      for($n=0;$n<sizeof($num_compte_com);$n++)
      {
       $id_cp_com=$id_cmpt_com[$n];
       $nu_cp_com=$num_compte_com[$n];
       $cp_com=$compte_com[$n];
      }
      $nouvo_cmp=$cp_com+$prix;
      //modif
      $trans="update compte_hotel set solde_hotel='".$nouvo_cmp."' where
id_compte='".$id_cp_com."'";
      $res_trans=mysql_query($trans);
      if($res_trans)
      {     mysql_connect("localhost","root","ludovic");
            $base=mysql_select_db("hotel");
            $date_reserv=date("Y-m-d : h:s");
            $paye="paye";
```

```
            $reserv="insert                                                    into
reservation(nom,prenom,adresse,ville,pays,telephone,mail,deb_reserv,sejour,fin_
reserv,num_chambre,tarif,transaction,date_reserve) values
('".$nom."','".$prenom."','".$adresse."','".$ville."','".$pays."','".$telephone
."','".$mail."','".$debut."','".$nb_j."','".$fin."','".$num_ch."','".$prix."','
".$paye."','".$date_reserv."')";
$query=mysql_query($reserv);
if(   $query)
{     mysql_connect("localhost","root","ludovic");
      $base=mysql_select_db("hotel");
      $fact="truncate table facturation";
      $query2=mysql_query($fact);
      if($query2)
      {
            echo"<script type=\"text/javascript\">
     {
      alert(\"Merci pour votre confiance!!\");
      document.location=\"index1.php\";
     }
</script>";
        }
else{
      echo"<script type=\"text/javascript\">
     {
      alert(\"Erreur de connexion!!\");
      document.location=\"paiement.php\";
     }
</script>";
    }
      }
      }
      else{ echo"<script type=\"text/javascript\">
  {   alert(\"Votre compte n'existe pas, veuillez réessayer ?\");
      document.location=\"paiement.php\";
  }
</script>";
            }
}
?>
```

ANNEXE 5 : EXEMPLE DES PAGES CLIENTS

A5.1 Page d'accueil

Figure A04: *index.php*

A5.2 Chambre

Figure A05 : *chambre.php*

A5.3 Réservation

Figure A06: *reservation.php*

A5.4 Paiement

Figure A07: *paiement .php*

ANNEXE 6: EXEMPLE DES PAGES D'ADMIN

A6.01: Modification des chambres

Figure A08: *etat_admin.php*

A6.02: Liste des reservations

Figure A09: *reservation_admin.php*

A6.03: Message d'information

Figure A10: *message_admin_nouveau.php*

ANNEXE 7 : CONFIGURATION D'UN SERVEUR WEB APACHE AVEC SSL

A7.1 Fichier à télécharger :

- Openssl-0.9.8a.tar.gz sur www.openssl.org**;**
- Apache_2.0.54.tar.gz sur www.apache.org.

Nous avons copié les fichiers téléchargé dans le repertoire **/home.**

A7.2 Installation des parquets

- **Openssl-0.9.8a.tar.gz**

Voici la commande:

```
#cd/usr/src/
#gunzip< /home/openssl-0.9.8a.tar.gz|tar xv-
#cd /usr/src/ openssl-0.9.8a
#./config
#apt-get install
```

- **Apache2**

On a deux methods pour installer apache2 sur notre serveur:

- Apache est déjà installer avec l'installation du systeme

Commande:

```
#aptitude install apache2
++ insertion de DVD
```

- Apache télécharger: apache_2.0.5.4.tar.gz

Commande:

```
#cd/usr/src/
#gunzip< /home/apache_2.0.5.4.tar.gz|tar xv-
#cd /usr/src/ apache_2.0.5.4
#./configure
#apt-get install
```

A7.3 Configuration d'Apache avec SSL

- **Configuration d'un serveur web non sécurisé**

1- Configuration de apache2.conf

```
#vi /etc/apache2/apache2.conf
Verifier
      User www-data
      Group www-data
Ajouter
      DirectoryIndex index.html index.htm index.php index.xhtml
      UserDir public_html
A la fin du fichier
      Include /etc/apache2/sites-enabled/[^.#]*
Sauver et fermer
```

2- Repertoire public des utilisateurs

```
Créer lien symbolique
#ln     –s     /etc/apache2/modes-available/userdir.conf     /etc/apache2/modes-enabled.conf
#ln     –s     /etc/apache2/modes-available/userdir.load     /etc/apache2/modes-enabled.load
```

3- Redémarrer Apache2

```
#/etc/init.d/apache2 reload
```

4- Installation des sites virtuels

```
#mkdir /etc/skel/public_html
#mkdir /etc/skel/logs
#useradd –g www-data –m gashotel
```

5- Creation du fichier du virtualhost

```
#vi /etc/apache2/sites-available/gashotel.com
      NameVirtualHost 192.168.0.1
<VirtualHost 192.168.0.1>
ServerAdmin postmaster@gashotel.com
ServerName www.gashotel.com
ServerAlias gashotel.com *.gashotel.com
DocumentRoot /home/gashotel/public_html
<Directory /home/gashotel/public_html/>
      Options –Indexes FollowSymLinks MultiViews
      AllowOverride All
</Directory>
ErroLog /home/gashotel/logs/error.log
LogLevel warn
```

```
CustomLog /home/gashotel/logs/access.log combined
ServerSignature Off
</VirtualHost>
```

6- Lien symbolique

```
#ln -s /etc/apache2/sites-available/gashotel.com /etc/apache2/sites-enabled/gashotel.com
```

7- Redémarrer apache2

```
#/etc/init.d/apache2 reload
```

8- Editer /etc/hosts

Ajouter:192.168.0.1 www.gashotel.com

9- Copier votre site dans /home/gashotel/public_html

10- Configuration BIND et SERVEUR DNS

- **Installation et configuration de named.conf**

```
#apt-get install bind9
#vi /etc/bind/named.conf
Ajouter à la dernière zone
      Zone “gashotel.com”
      {type master;
      File “/etc/bind/db.gashotel.com”;
      };
```

- **Duplication db.local en db.gashotel.com et editer**

```
;
;BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA expert.gashotel.com root.expert.gashotel.com.(
                  1           ; Serial
                  604800      ; Refresh
                  86400 ; Retry
                  2419200     ; Expire
                  604800)     ; Negative Cache TTL
;
@     IN    NS    expert.gashotel.com.
@     IN    NS    www.gashotel.com.
@     IN    A     192.168.0.1
Expert      IN    A     192.168.0.1
```

```
www-gashotel.com  IN CNAME expert.gashotel.com
ftp  IN  CNAME    expert.gashotel.com.
```

11- Test du serveur DNS

```
#ping gashotel.com
```

- **Configuration d'Apache2 avec SSL**

1- Création d'une autorité de certificats avec SSL

Avant de créer l'autorité de certification, il faut preparer quelques repertoire

```
$su
#cd
#mkdir /var/key
#cd /var/key
#openssl req –new –out server.csr
```

Après la demande de signature, nous avons comme ci-dessous e tapant **ls**

```
Server.csr privkey.pem
```

2- Creation du certificat pour Apache

```
#openssl rsa –in privkey.pem –out server.key
read RSA key
Enter PEM pass phrase:
Writing RSA key
#ls
Server.csr server.key privkey.pem
```

3- Signature du certificat d'Apache par l'autorité de certification

```
#openssl x509 –in server.csr –out server.crt –req –signkey server.key –days 365
Signature OK
subject=/C=FR/ST=France/L=PARIS/O=Mon_organisation/OU=mon_domaine.tld/CN=mon_do
maine.tld/Email=mon@adresse.mail
Getting Private key
#ls
Server.csr server.crt server.key privkey.pem
```

4- Configuration d'Apache2

```
#a2enmod ssl
#vi /etc/apache2/port.conf
Ajouter
      <IfModule mod_ssl.c>
      Listen 443
```

```
      </IfModule>
Sauver et quitter
#vi /etc/apache2/sites-available/gashotel.com
```
Ajouter
```
NameVirtualHost 192.168.0.1:443
<VirtualHost *:80>
ServerName site1.com/
Redirect / https://192.168.0.1/
</VirtualHost>
<VirtualHost 192.168.0.1:443>
ServerName gashotel.com
DocumentRoot /home/gashotel/public_html/
ServerAlias gashotel.com *.gashotel.com
DocumentRoot /home/gashotel/public_html
<Directory /home/gashotel/public_html/>
      Options –Indexes FollowSymLinks MultiViews
      AllowOverride All
</Directory>
ErroLog /home/gashotel/logs/error.log
LogLevel warn
CustomLog /home/gashotel/logs/access.log combined
SSLEngine on
SSLCertificateFile /var/key/server.crt
SSLCertificateKeyFile /var/key/server.key
</VirtualHost>
```
Si l'utilisateur arrive sur le port 80, on le redirige sur le 443.

5- Redémarrer Apache2 et terminer la configuration

BIBLIOGRAPHIE

[1] *« Internet »,* http://fr.wikipedia.org/wiki/Internet

[2] http://www.commentcamarche.net/

[3] *« Internet »,* http://fr.wikipedia.org/wiki/Internet

[4] *« Internet »,* http://www.computerhistory.org/internet_history/

[5] Guy Vastersavendts, *Utiliser internet et ses services,* Octobre 1998

[6] « *Web »,* http://fr.wikipedia.org/wiki/Web

[7] « *Web »,* http://www.defidoc.com

[8] « Site *Web »,* http://www.ritimo.org

[9] « JavaScript », http://devedge.netscape.com/central/JavaScript/

[10] *« PHP »,* http:// developpez.com

[11] *« PHP »,* http://www.php.net

[12] http://www.neuronnexion.com/

[13] *« SSL »,* http://securiteinfo.com

[14] *« SSL »,* http://sebsauvage.net

[15] NGAKA Francis, « *Certification & Cryptage »,* DESS CCI FC 2002

[16] *« SSL »,* http://secubook.tuxfamily.org/secubook.pdf

[17] Ramafiarisona Malalatiana, « *Bases de données* », Cours L3, Dép. TCO.-ESPA, AU.: 2009-2010

[18] Bruno Razafindradina, « *web et XML* », Cours L3, Dép. TCO.-ESPA, AU. : 2008-2009

[19] Réiny C, « *Système d'information* »

[20] http://www.apache.org

[21] http://www.debian.org

[22] *« openssl »,* http:// www.openssl.org

[23] Thomas Petazzoni, « *Réseau sous GNU/Linux »*

[24] http://www.paypal.com

FICHE DE RENSEIGNEMENT

Nom : RAZAFINDRAKOTO

Prénoms : Arnaud Ludovic

Adresse de l'auteur : Lot 22B Mahazoarivo - FKT Angoja

Ambatosoratra - Ambatondrazaka 503

Téléphone : +261 33 07 756 92

E-mail : razafindrakoto.ludovic@gmail.com

Titre de mémoire : **MISE EN PLACE D'UN SYSTEME D'INFORMATION SECURISE SOUS LINUX**

Nombre de pages : 80

Nombre de tableaux : 3

Nombre de figures : 33

Mots clés

Application web – Merise - Système d'information –SSL – Cryptographie – Apache - HTTPS

Directeur de mémoire : Monsieur RAZAKARIVONY Jules

RESUME

Grace à l'évolution de l'Internet et les différentes technologies de la télécommunication, les Entreprises peuvent échangés des biens et de services entre eux ou entre leurs clients.

Ainsi, l'implantation de la sécurisation du réseau à l'aide des algorithmes cryptographiques nous garanti la confidentialité, l'integrité durant les échanges de données ou les traitements des informations sur web.

De plus, l'installation des câbles sous-marins internationaux à Madagascar donne de l'existence de la bande passante illimitée et l'augmentation des débits à l'Internet.

ABSTRACT

Thanks to the evolution of Internet and the different technology of the telecommunication, the companies want to practice their exchanged with their customer.

On the one hand, the implementation of security in the network by using specific algorithms cryptographic guaranteed us confidentiality, the integrity during the data exchange and the information treatments.

On the other hand, the installation of the optical fiber's international to Madagascar gets the existence of the unlimited width-band and feels good to be of the fasted Internet users on the planet.