



UNIVERSITÉ D'ANTANANARIVO
SCIENCES ET TECHNOLOGIES
PHYSIQUES ET APPLICATIONS



MÉMOIRE

Pour l'obtention du diplôme de

Master d'Ingénierie en Systèmes Électroniques et Informatiques

Sur :

MISE EN PLACE D'UN SERVEUR WEB SECURISÉ

Présenté par :

RAZANANIRINA Angelica Etienne

Devant la commission d'examen composée de:

Président:

M. RANAIVO-NOMENJANAHARY Flavien Noël

Professeur Titulaire

Examineurs :

Mme. RANDRIAMANANTANY Zely Arivelo

Professeur Titulaire

Mme. RAZANAMANAMPISOA Harimalala

Maître de Conférences

Rapporteur :

M. RAKOTOARIMANANA Liva Graffin

Maître de Conférences

Avril 2018



12 Avril 2018



UNIVERSITÉ D'ANTANANARIVO
SCIENCES ET TECHNOLOGIES
PHYSIQUES ET APPLICATIONS



MÉMOIRE

Pour l'obtention du diplôme de

Master d'Ingénierie en Systèmes Électroniques et Informatiques

Sur:

MISE EN PLACE D'UN SERVEUR WEB SECURISÉ

Présenté par :



RAZANANIRINA Angelica Etienne

Devant la commission d'examen composée de :

Président:

M. RANAIVO-NOMENJANAHARY Flavien Noël

Professeur Titulaire

Examineurs :

Mme. RANDRIAMANANTANY Zely Arivelo

Professeur Titulaire

Mme. RAZANAMANAMPISOA Harimalala

Maître de Conférences

Rapporteur :

M. RAKOTOARIMANANA Liva Graffin

Maître de Conférences

Avril 2018

REMERCIEMENTS

Ce mémoire est l'étape finale, le fruit, la concrétisation de nos cinq années d'études théoriques et pratiques au sein du domaine Sciences et Technologies. Ce travail a été réalisé aux aides bienveillants, aux directives efficaces et à la collaboration étroite des personnes auxquelles je ne peux pas les nommer toutes. Toutefois, je suis profondément reconnaissante, sans pour autant omettre tous les services qui ont témoigné leur soutien, combien inestimable, pour la réalisation de ce travail.

En premier lieu, je tiens à adresser des remerciements à l'endroit des personnes suivantes :

- Monsieur RAHERIMANDIMBY Marson, Professeur Titulaire, Responsable du Domaine des Sciences et de Technologies à l'Université d'Antananarivo, de m'avoir accepté notre inscription dans ladite du Domaine ;
- Monsieur RAKOTONDRAMIANANA Hery Tiana, Maître de Conférences, Responsable de la Mention Physique et Applications, pour sa volonté de nous avoir accordés cinq années d'études au sein de ladite Mention ;
- Madame RAZANAMANAMPISOA Harimalala, Maître de Conférences, Responsable du Parcours « Master d'Ingénieur en Systèmes Electronique et Informatique » non seulement pour nous avoir reçue et admise au sein de ce parcours, mais aussi pour ses précieux conseils et son dévouement.

Je souhaite également adresser mes vifs remerciements aux membres du jury notamment :

- Monsieur RANAIVO-NOMENJANAHARY Flavien Noël, Professeur Titulaire, pour sa volonté à bien vouloir à présider la séance ;
- Madame RANDRIAMANANTANY Zely Arivelo, Professeur Titulaire et Madame RAZANAMANAMPISOA Harimalala, Maître de Conférences, examinateurs de ce mémoire pour l'apport de ses jugements en vue d'enrichir ce travail.
- Monsieur RAKOTOARIMANANA Liva Graffin, Maître de Conférences, mon encadreur pédagogique, qui m'a aidé à réaliser ce mémoire.

Mes remerciements s'adressent également :

- A tous les enseignants formateurs de MISEI qui ont partagé leur savoir-faire tout au long de mes études.
- A tous les étudiants de la promotion, qui m'ont encouragé tout au long de mes études.

Enfin, j'adresse mes vifs remerciements à ma famille, à mes amis, et surtout à mes parents, qui m'ont toujours soutenue moralement et financièrement.

TABLE DES MATIÈRES

REMERCIEMENTS	i
LISTE DES ABRÉVIATIONS	iv
LISTE DES FIGURES	vi
LISTE DES ANNEXES	vii
INTRODUCTION.....	1
CHAPITRE I : GÉNÉRALITÉ SUR LA SÉCURISATION DE SERVICE INTERNET	3
I. Généralités sur la sécurisation de service Internet	4
I.1 Serveur.....	4
I.1.1 Serveur classique	4
I.1.2 Serveur autonome	4
I.2 Eléments du serveur web	5
I.2.1 Apache	5
I.2.2 PHP (Préprocesseur Hypertexte)	5
I.2.3 MySQL	6
I.3 Vulnérabilité du serveur web.....	6
I.3.1 Attaque de serveur web.....	6
I.3.2 Attaque de base de données	7
CHAPITRE II : METHODE DE SECURISATION DE SERVEUR WEB	8
II. Méthode de sécurisation de serveur web	9
II.1 Sécurisation au niveau réseau	9
II.1.1 ACL.....	9
II.1.2 Firewall.....	10
II.2 Sécurisation du système.....	11
II.2.1 Sécurisation par cryptage	11
II.2.2. Sécurisation par authentification	12
II.2.3. Sécurisation par filtrage	13

CHAPITRE III : MISE EN PLACE DE LA SÉCURISATION DU SERVEUR WEB DYNAMIQUE	15
III. Mise en place de la sécurisation du serveur web dynamique.....	16
III.1 Installation de serveur web	16
III.1.1 Installation Apache.....	16
III.1.2 Installation de PHP.....	18
III.1.3 Installation de MySQL	19
III.2 Sécurisation de serveur Apache.....	19
III.2.1 Activation de SSL	19
III.2.2 Authentification <i>.htaccess</i>	20
III.2.2.1 Création du fichier <i>.htaccess</i>	21
III.2.2.2 Création du fichier <i>.htpasswd</i>	21
III.2.3 iptables pare-feu	22
III.2.3.1 Filtrage par IP	22
III.3 Sécurisation de base de données.....	25
III.4 Résultats et discussions	26
III.5 Statistiques de piratage de site web	31
III.6 Analyse pour le cryptage	33
CONCLUSION	39
RÉFÉRENCES.....	41

LISTE DES ABRÉVIATIONS

ACL: Access Control List

AIX: Advanced Interactive eXecutive

BSD: Berkley System Development

DoS: Deny of Service

HTML: Hypertext Mark-Up Language

HTTP: Hypertext Transfer Protocol (protocole de transfert hypertexte)

HTTPS: Hypertext Transfer Protocol Secure

HIDS : Système de détection d'intrusion machine

HP-UX: Hewlett Packard UNIX

IP: Internet Protocol

MISEI : Master d'Ingénierie en Systèmes Electronique et Informatique

OSSEC: Open Source HIDS SECurity

PHP : Préprocesseur Hypertexte

PC: Personal Computer

SGBD : Système de Gestion de Bases de Données

SQL: Structured Query Language

SSL: Secure Sockets Layer

ToS: Type of service

TCP: Transmission Control Protocol

TLS: Transport Layer Security

URL: Uniform Resource Locator

WWW: World Wide Web

SSH: Secure Shell

FTP: File Transfer Protocol

LISTE DES FIGURES

Figure 1: Schéma synoptique de sécurisation par filtrage.....	9
Figure 2: Affichage du bon fonctionnement Apache	16
Figure 3 : Affichage d'erreur du serveur Apache	17
Figure 4 : Configuration de l'adresse IP	17
Figure 5 : Affichage du bon fonctionnement de PHP	18
Figure 6 : Schéma d'illustration de filtrage IP 1	23
Figure 7 : Schéma d'illustration de filtrage IP 2	24
Figure 8 : Schéma du système client/serveur	26
Figure 9 : Résultat du test ping entre client/serveur.....	27
Figure 10 : Résultat du serveur authentifié	27
Figure 11 : Résultat du serveur dans le page web	28
Figure 12 : Résultat du serveur lorsqu'il ne répond pas.....	29
Figure 13 : Vérification d'access.log	30
Figure 14 : Vérification d'error.log.....	31
Figure 15 : Statistique des piratages d'un site web depuis 6 ans	32
Figure 16 : Illustration sur le principe de confidentialité d'un message crypté et décrypté	35
Figure 17 : Illustration sur le principe d'authentification d'un message crypté et décrypté	36
Figure 18 : Illustration de http over SSL.....	37
Figure 19 : Code de César	38

LISTE DES ANNEXES

ANNEXE 1: Installation Apache2	43
ANNEXE 2: Installation de PHP	44
ANNEXE 3: Installation de MySQL.....	45
ANNEXE 4: Configuration de PHP.....	46

INTRODUCTION

La sécurité informatique n'évoque pas simplement le fait de se protéger contre un éventuel piratage, elle évoque également le fait de protéger ses données contre des pertes ou altérations. Durant mon stage, j'ai été formée dans la société AINA Consulting pour apprendre la base de la commande Linux et de connaître comment faire installer le système d'exploitation Linux.

Les mécanismes de sécurité mis en place peuvent néanmoins provoquer une gêne au niveau des utilisateurs, les consignes et règles deviennent de plus en plus compliqués au fur et à mesure que le réseau s'étend. Ainsi, la sécurité informatique doit être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils puissent utiliser le système d'information en toute confiance.

La sécurité des réseaux informatiques est un sujet essentiel pour favoriser le développement des échanges dans tous les domaines et à ne pas mettre de côté pour tout administrateur système qui se respecte.

Le problème se pose, pourquoi doit-on sécuriser le serveur web ? Du point de vue technique, la sécurité recouvre à la fois l'accès aux informations sur les postes de travail, sur les serveurs ainsi que le réseau de transport des données ou des fichiers. Sécuriser les données, c'est garantir la confidentialité pour éviter que les données soient lues par des systèmes ou des personnes non autorisées. Par le souci de sécurité de données sur l'étoile de web, notre travail consiste à traiter « La mise en place d'un serveur web sécurisé ».

Nombreux sont les articles sur internet expliquant comment sécuriser les applications web, comment sécuriser le serveur Web. Ce mémoire a pour but de rassembler et surtout compléter l'existant afin de faire un tutorial des plus complets. La sécurisation de serveur web est une étape cruciale pour le webmaster ou un administrateur systèmes. Ceci doit être mis en place, et en même temps, c'est une procédure continue qui doit être lancée régulièrement.

Le principal objectif est de mettre en place un serveur web et assurer sa sécurisation qui est basée sur le cryptage en utilisant SSL, l'authentification avec *.htaccess* et le filtrage d'adresse IP par *iptables*. Toutes les données à sécuriser se trouvent dans le serveur Apache et le SGBD (Système de Gestion de Bases de Données). La sécurisation d'un site est une course de fond perpétuelle, rythmée par la cadence imposée par les pirates. Compte tenu de sa

robustesse et son fort potentiel en matière de sécurité, notre serveur fonctionne sur le système d'exploitation Linux dont la distribution *Ubuntu*.

Afin de mener à bien cette étude et dans le but de mieux expliciter le comment et le pourquoi de ce qui a été fait ; en premier lieu, nous allons voir les généralités sur la sécurisation de service internet. Le deuxième chapitre sera dédié à la méthode de sécurisation de serveur web. Enfin, nous allons voir la mise en place de sécurisation de serveur web dynamique.

CHAPITRE I : GÉNÉRALITÉ SUR LA SÉCURISATION DE SERVICE INTERNET

I. Généralités sur la sécurisation de service Internet

I.1 Serveur

Par définition, un serveur est une machine ou un programme qui offre un service à un client et tout simplement un ordinateur qui contient des données (ou des fichiers) qui compose le site Internet. Ce serveur est accessible grâce à sa connexion permanente au réseau Internet par les lignes téléphoniques. Le serveur généralement situé dans une salle d'hébergement va donc permettre d'envoyer les données aux ordinateurs qui en font la demande.

Il existe deux types de serveur dans le domaine de service internet, d'abord le serveur classique et ensuite le serveur autonome.

I.1.1 Serveur classique

Le serveur classique (ou traditionnel) est un dispositif informatique (matériel ou logiciel) offrant des services à des multiples clients. La majorité d'entre eux appartiennent à des sociétés commerciales (comme le Google) ou à des institutions (comme les universités ou les écoles). Il permet un accès rapide, direct pour les entrées et sorties vers les fichiers de la base de données pour les connexions locales sous Linux. Sous Windows, on doit faire les connexions locales avec la notation réseau, en se connectant à *localhost*. Il est très stable sous Linux ; encore pour certains points expérimentaux sous Windows.

I.1.2 Serveur autonome

Le serveur autonome (ou serveur *standalone*) est une sorte de serveur sur un réseau qui n'est pas dépendant des autres, mais est indépendante en soi. Le serveur autonome fournit le même type de service (messagerie mail et discussion instantanée, hébergement, blogs, etc.), exception faite que leur fonctionnement ne dépend pas d'une société ou d'une institution (qu'elle soit privée ou publique). Le fonctionnement du serveur autonome recherche à diffuser des informations et des documents en dehors des circuits commerciaux traditionnels. Ils représentent une initiative visant à démocratiser l'accès à l'information et à la production de contenus en ligne. La fonction principale d'un serveur autonome est de séparer certaines tâches beaucoup mieux s'il n'est pas connecté au contrôleur de domaine.

I.2 Eléments du serveur web

En informatique, le mot serveur web désigne à la fois une machine physique et un logiciel. Dans le premier cas, il s'agit d'un ordinateur relié à Internet et hébergeant des ressources. Ces ressources peuvent être des fichiers, des programmes ou des bases de données. Dans ce serveur web, on peut travailler sur un serveur Apache et une base de données (MySQL). Apache c'est la grande source de cette sécurisation c'est-à-dire que toutes les sécurisations se placent dans cet Apache.

Le serveur généralement situé dans une salle d'hébergement va donc permettre d'envoyer les données aux ordinateurs qui en font la demande. Le serveur web est assuré par trois logiciels comme :

- Apache
- PHP
- MySQL

I.2.1 Apache

Apache est un serveur Web gratuit fonctionnant sous Linux; et largement utilisé sur internet. Il est un serveur HTTP créé et maintenu au sein de la fondation Apache et un logiciel libre aussi. C'est le serveur HTTP le plus populaire du World Wide Web (www). La disponibilité d'Apache fonctionne principalement sur les systèmes d'exploitation Linux et Windows. Apache est le serveur le plus répandu sur Internet. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows.

I.2.2 PHP (Préprocesseur Hypertexte)

Le serveur d'application PHP est un langage de scripts libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande. Ce langage est principalement utilisé pour produire un site web dynamique. Il est courant que ce langage soit associé à une base de données, tel que

le MySQL. Lorsqu'une page PHP est exécutée par le serveur, alors celui-ci renvoie généralement au client (aux visiteurs du site) une page web qui peut contenir du HTML.

I.2.3 MySQL

MySQL est le serveur de bases de données le plus répandu pour les serveurs web. Il n'utilise pas les comptes Unix mais gère ses propres utilisateurs et mots de passe. La société MySQL fournit de nombreuses solutions pour les bases de données. Ce MySQL permet de modifier le mot de passe d'un utilisateur existant. Grâce à sa performance, sa fiabilité et sa facilité d'utilisation, MySQL est devenu le principal choix de la base de données pour les applications web, utilisées par des propriétés web de haut niveau.

I.3 Vulnérabilité du serveur web

Les vulnérabilités des serveurs en particulier le serveur web sont de plus en plus rares car au fur et à mesure des années les principaux développeurs de serveurs web ont renforcé leur sécurisation. De logiciel comme OSSEC (Open Source HIDS SECurity), un système de détections d'intrusion de type HIDS (Système de détection d'intrusion machine) s'installe sur les serveurs de type Linux, Solaris, AIX, HP-UX, BSD, Windows, Mac and VMware ESX et offre les fonctionnalités suivantes :

- Vérification de l'intégrité des fichiers systèmes.
- Supervision, analyse et corrélation des logs du système.
- Prévention active en cas de détection d'attaques.
- Détection des outils de dissimulation d'activité appelés souvent *rootkit*.

I.3.1 Attaque de serveur web

Le protocole HTTP (ou HTTPS) est le standard permettant de véhiculer les pages web par un mécanisme de requêtes et de réponses. Il est utilisé essentiellement pour transporter des pages web informationnelles (pages web statiques), le web est rapidement devenu un support interactif permettant de fournir des services en ligne. Les premières attaques réseau exploitaient des vulnérabilités liées à l'implémentation des protocoles de la suite TCP/IP. Dans la mesure où les serveurs web sont de plus en plus sécurisés, les attaques se sont progressivement décalées vers l'exploitation des failles des applications web.

Les vulnérabilités des applications web peuvent être catégorisées de la manière suivante :

- La manipulation des URL consiste à modifier manuellement les paramètres des URL afin de modifier le comportement attendu du serveur web ;
- L'exploitation des faiblesses des identifiants de session et des mécanismes d'authentification ;
- L'injection de code HTML et Cross-Site Scripting (attaquer les utilisateurs de l'application plutôt que l'application elle-même) ;
- L'injection de commandes SQL

Tous les services réseau peuvent faire l'objet d'attaques de type "Déni de service" ou *Deny of Service* (DoS) qui tentent de les empêcher de répondre aux clients en saturant leurs ressources. Une attaque DoS consiste à tenter de bloquer l'accès à un serveur en monopolisant toutes ses ressources. Il est impossible de se prémunir totalement contre ce type d'attaques, mais certaines actions peuvent s'avérer utiles afin de minimiser les problèmes qu'elles créent. Malgré le risque, certaines actions préventives permettent de limiter la casse.

I.3.2 Attaque de base de données

Une attaque est une action qui exploite une vulnérabilité ou exécute une menace.

Par exemple, envoyer des données d'entrée malveillants à une application ou saturer un réseau en vue d'entraîner un refus de service.

On a plusieurs risques sur le système d'information, qui sont :

- Les vols,
- La destruction de données ou de matériels,
- Les captations d'informations, indisponibilité du système,
- Une origine qui peut être externe mais souvent interne (malveillance ou négligence).

Il y a aussi des risques pour quelques mauvaises configurations comme :

- Modes d'authentification dégradés
- Changement de mot de passe par défaut
- Risque des fichiers de la base de données non sécurisés
- Risque des pertes de données
- Incohérences et indisponibilité des données

CHAPITRE II : METHODE DE SECURISATION DE SERVEUR WEB

II. Méthode de sécurisation de serveur web

II.1 Sécurisation au niveau réseau

Le réseau c'est un ensemble d'équipements interconnectés pouvant communiquer, et il a pour but de transmettre des informations d'un ordinateur à un autre. Et concernant la sécurité, elle est un ensemble de stratégies, conçues et mises en place pour détecter, prévenir et lutter contre une attaque. La sécurisation au niveau réseau a besoin d'ACL et du firewall parce que ces deux sécurisations de réseaux ont des règles très efficaces pour filtrer les adresses IP.

Voici quelque schéma synoptique de la démarche de cette sécurisation du serveur web :

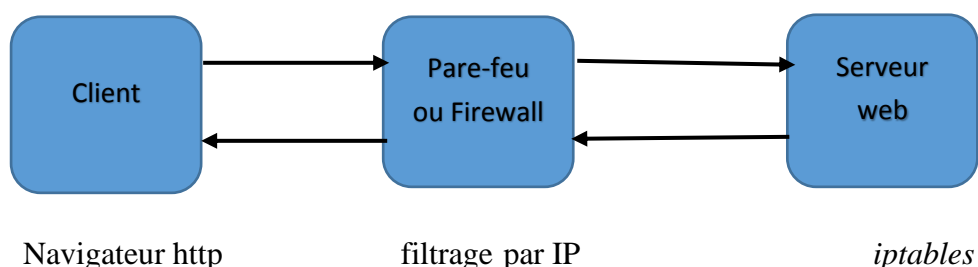


Figure 1: Schéma synoptique de sécurisation par filtrage

Par définition :

- Le client est un système (programme ou ordinateur) accédant à des ressources éloignées, en se branchant via un réseau informatique sur un serveur ;
- Le serveur est un ordinateur détenant des ressources particulières et qu'il met à la disposition d'autres ordinateurs par l'intermédiaire d'un réseau.

II.1.1 ACL

Une ACL est une liste de règles permettant de filtrer ou d'autoriser du trafic sur un réseau en fonction de certains critères (IP source, IP destination, port source, protocole, port destination, etc.). Il permet de soit autoriser du trafic (*permit*) ou de le bloquer (*deny*) par le dispositif de filtrage. Les ACL peuvent être utilisés comme une extension du concept traditionnel d'autorisation d'accès aux fichiers.

Les ACL sont divisés en trois grandes catégories : l'ACL standard, l'ACL étendue et la nommée-étendue :

- L'ACL standard : ne peut contrôler que deux ensembles, l'adresse IP source et une partie de l'adresse IP de destination, au moyen de masque générique.
- L'ACL étendu : peut contrôler l'adresse IP de destination, la partie de l'adresse de destination, le type de protocole (TCP), le port source et de destination, les flux TCP, IP TOS (Type of service) ainsi que les priorités IP.
- L'ACL nommée-étendu : est une ACL étendue à laquelle on a affecté un nom.

II.1.2 Firewall

Un firewall est un outil informatique (matériel et/ou logiciel) conçu pour protéger les données d'un réseau (protection d'un ordinateur personnel relié à Internet par exemple, ou protection d'un réseau d'entreprise) et un élément indispensable pour sécuriser un serveur. Il permet d'assurer la sécurité des informations d'un réseau en filtrant les entrées et en contrôlant les sorties selon des règles définies par son administrateur.

Le firewall va réaliser les tâches suivantes :

- Blocage d'accès à des services non autorisés
- Interdiction d'accès à des systèmes
- Protection contre les attaques de type DoS (Deni de Service)

Les firewalls peuvent intégrer des techniques de détection d'intrusions et aussi peuvent envoyer des alertes afin de prévenir les équipes de surveillance technique. Ces équipements prennent en compte un ensemble de règles qui doivent être définies en fonction des besoins d'une entreprise ou d'une administration. Ils vont en effet filtrer tout le trafic en n'autorisant que les échanges permis par l'administrateur. Sans firewall correctement réglé, tous les trafics sont plus ou moins permis (c'est-à-dire qu'un attaquant puisse faire ce qu'il veut chez nous) et ce genre de faille est détectable par un simple scan de ports. Or, le noyau Linux offre déjà un pare-feu à l'utilisateur, qu'il est possible de configurer via le logiciel *iptables*.

Les bases d'*iptables* doivent correspondre aux règles suivantes :

- *ACCEPT* : Cela signifie que le paquet sera autorisé à passer.

- *DROP* : Cela signifie que le paquet ne sera pas autorisé à passer.
- *RETURN* : Cela signifie ignorer la chaîne actuelle et revenir à la règle suivante de la chaîne dans laquelle elle a été appelée.

Pour les besoins de ce didacticiel *iptables*, nous allons travailler avec l'un des tableaux par défaut appelés filtre. Le tableau des filtres comporte trois chaînes (ensembles de règles).

- *INPUT* : Cette chaîne est utilisée pour contrôler les paquets entrants sur le serveur.
- *FORWARD* : Cette chaîne est utilisée pour filtrer les paquets qui entrent sur le serveur mais doivent être transférés ailleurs.
- *OUTPUT* : Cette chaîne est utilisée pour filtrer les paquets qui sortent sur le serveur

II.2 Sécurisation du système

La sécurisation du système est une étape pour sécuriser un serveur, dont on a besoin de la :

II.2.1 Sécurisation par cryptage

Le cryptage ou le chiffrement des données, c'est aussi un mécanisme de sécurité, qui consiste à traduire un message clair, dit originel en un message incompréhensible, inintelligible. Le résultat du processus de cryptage est appelé « texte chiffré ou message codé ». Le processus de cryptage repose à la fois sur des algorithmes puissants et sur les paramètres appelés clés. Le cryptage peut faire des transferts de données.

Le protocole sécurisé SSL lui fonctionne comme une sous-couche inférieure du même modèle TCP/IP, mais il crypte le message HTTP avant sa transmission et la déchiffre à l'arrivée. On peut donc dire que HTTPS n'est pas un protocole séparé, mais se réfère à l'utilisation du HTTP ordinaire via une connexion SSL cryptée. En utilisant le HTTPS, les données échangées sont cryptées. Les données cryptées apportent une protection contre le piratage des informations échangées. Le HTTPS permet de rendre un site plus sûr pour les internautes qui le visitent.

En termes de sécurité, le protocole HTTPS assure :

- La confidentialité, contre le piratage dans l'espionnage des données personnelles
- L'intégrité du HTTPS qui protège l'altération des données échangées

- L'Authenticité permet de garantir l'identité du programme, de la personne ou de l'entreprise qui édite le site

La différence entre HTTP & HTTPS :

Il y a de nombreux critères qui différencient HTTP de HTTPS, voici les 3 principaux :

1. Pour le schéma d'URL :

- Les URLs HTTPS commencent par "https : //" et utilise le port 443 par défaut (l'URL est déjà cryptée)
- Les URLs HTTP commencent par "http : //" et utilisent le port 80.

2. Pour la sécurité : HTTP n'est pas sécurisé et est soumis à de nombreuses attaques, qui peuvent laisser des attaquants avoir accès à des informations sensibles, en revanche un site Web tout en HTTPS est conçu pour résister et protéger contre de telles attaques.
3. Pour les couches réseau : HTTP fonctionne sur la couche la plus élevée du modèle TCP/IP qui est la couche application.

II.2.2. Sécurisation par authentification

L'authentification consiste à demander à un utilisateur de prouver son identité. Il est un mécanisme de sécurité qui consiste à assurer l'identité d'un utilisateur, ou d'une machine voulant accéder au système ; ainsi on vérifie que la station ou la personne, est bien celle qu'elle prétend être. En effet dans la plupart de temps, l'authentification s'agit du couple « nom d'utilisateur et mot de passe », c'est un mécanisme qui constitue une sécurité relativement fiable lorsqu'il est bien mis en œuvre. Grâce au *.htaccess* et *.htpasswd* qu'on peut authentifier l'accès au serveur web.

Les possibilités de configuration d'Apache sont une fonctionnalité *phare*. Le principe repose sur une hiérarchie de fichiers de configuration, qui peuvent être gérés indépendamment. Cette caractéristique est notamment utile aux hébergeurs qui peuvent ainsi servir les sites de plusieurs clients à l'aide d'un seul serveur HTTP. Pour les clients, cette fonctionnalité est rendue visible par le fichier *.htaccess*.

II.2.3. Sécurisation par filtrage

Le filtrage est un outil de sécuriser un donné ou un dossier dans le serveur c'est-à-dire le serveur est filtré par un adresse IP. C'est aussi une méthode pour bloquer l'adresse IP d'un client pour la sécurisation. Le filtrage permet de laisser passer certaines informations, mais seulement après l'analyses de ces dernières. Ce sont ces analyses qui permettront de déterminer quelles informations sont autorisées à arriver jusqu'à leurs destinataires. Le filtrage permet de contenir la trame de paquet internet TCP/IP et les résultats de Netstat comme IP source, port destination, etc.

Donc, une adresse IP est un identifiant similaire à un numéro de téléphone. Elle permet d'identifier un composant tel qu'un ordinateur, un routeur sur un réseau supportant le protocole IP (Internet Protocol). L'adresse IP (version IPv4) est un nombre de 32 bit. Pour la simplicité, lorsqu'on écrit une adresse IP, on note séparément les 4 valeurs correspondantes aux octets, en les séparant par des points.

Exemple d'adresse IP : 192.168.1.1

Les adresses IP contenues dans les paquets permettent d'identifier la machine émettrice et la machine cible, tandis que le type de paquet et le numéro de port donnent une indication sur le type de service utilisé. Le fonctionnement d'un système pare-feu contient un ensemble de règles prédéfinies permettant :

- D'autoriser la connexion (*allow ou accept*)
- De bloquer la connexion (*deny ou drop*)

Voici quelque commande de filtrage :

Pour le filtrage par adresse IP (bloqué ou filtré l'adresse IP du client), on a plusieurs règles qu'on doit suivre successivement :

Méthode d'installation de filtrage par IP pour la sécurisation :

- L'installation d'*iptables* sous Linux lorsqu'on autorise le paquet d'entrée de l'adresse IP du client entre dans le serveur :

```
$ sudo iptables -A INPUT -s 192.168.123.10 -p tcp -dport 80 -j ACCEPT
```

- L'installation d'*iptables* sous Linux lorsqu'on ne n'autorise pas le paquet de l'adresse IP du client :

```
$ sudo iptables -A INPUT -s 192.168.123.10 -p tcp -dport 80 -j DROP
```

Par raison de précaution, lorsqu'on passe à la commande `iptables -L`, on efface d'abord toutes les règles dans le filtrage, c'est-à-dire on passe à la commande `iptables -F` pour supprimer les commandes dans le filtrage et après, on fait un retour par la commande `./firewall.sh`, et enfin la commande `iptables -L`.

CHAPITRE III : MISE EN PLACE DE LA SÉCURISATION DU SERVEUR WEB DYNAMIQUE

III. Mise en place de la sécurisation du serveur web dynamique

III.1 Installation de serveur web

III.1.1 Installation Apache

Apache doit être installé dans une ligne de commande en tant que *root*, qui est :

```
# apt-get install apache2
```

```
# nano /etc/apache2/apache2.conf
```

Redémarrage d'Apache pour que les modules soient pris en compte :

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

Ou

```
# Service Apache2 restart
```

Pour tester le bon fonctionnement d'Apache, il suffit de taper l'adresse `http://localhost/` du serveur dans le navigateur (si le serveur est à distance `http://192.168.123.2/`) :

Et si le message « **It Works !** » apparaît, le serveur Apache fonctionne.

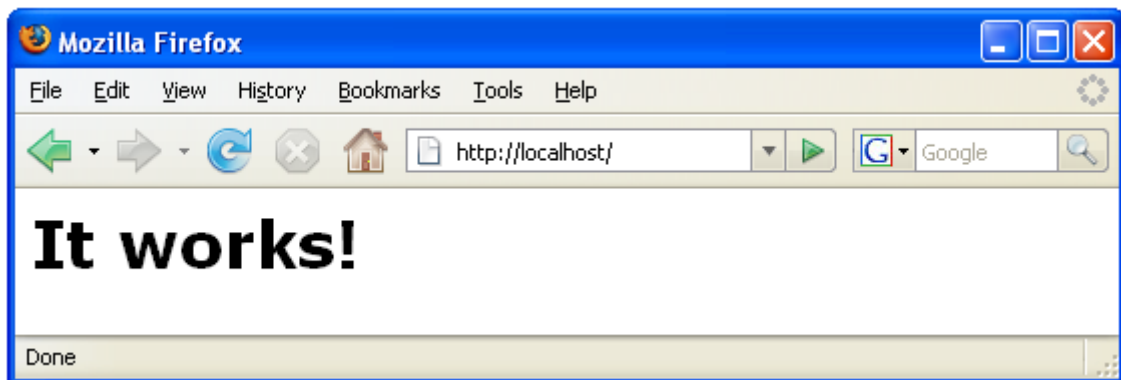


Figure 2: Affichage du bon fonctionnement Apache

Comme par défaut, Apache n'affiche pas certaines informations sur les pages (c'est-à-dire comme la page d'erreur par exemple).



Figure 3 : Affichage d'erreur du serveur Apache

La configuration de l'adresse IP se fait par le mode commande `ifconfig`. Le but est de fixer l'adresse du serveur parmi toutes les machines dans le réseau. Puisqu'on tape la commande `ifconfig`, on a les résultats des différentes adresses réseau comme :

- `eth0` : l'interface du serveur
- `lo` : *loopback* ou boucle locale

```
user@serveur-web: ~  
To access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
Last login: Wed Feb 14 16:13:26 2018  
user@serveur-web:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 00:1c:25:9f:e1:d5  
          inet addr:192.168.123.2  Bcast:192.168.123.255  Mask:255.255.255.0  
          inet6 addr: fe80::21c:25ff:fe9f:e1d5/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:880 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:46 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:100  
          RX bytes:66181 (66.1 KB)  TX bytes:6619 (6.6 KB)  
          Memory:fc000000-fc020000  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:100 (100.0 B)  TX bytes:100 (100.0 B)  
  
user@serveur-web:~$
```

Figure 4 : Configuration de l'adresse IP

III.1.2 Installation de PHP

L'installation du PHP sous Linux se fait de la manière suivante :

```
$ sudo apt-get update && apt-get install php5 && apt-get install libapache2-mod-php5
```

Afin de tester le module php installé, on crée un code dans un fichier test.php dont le contenu est :

```
< ?php  
print("PHP has been installed successfully ! \n<br>");  
?>
```

Ensuite, dans le navigateur, on entre l'adresse URL <http://localhost/test.php>

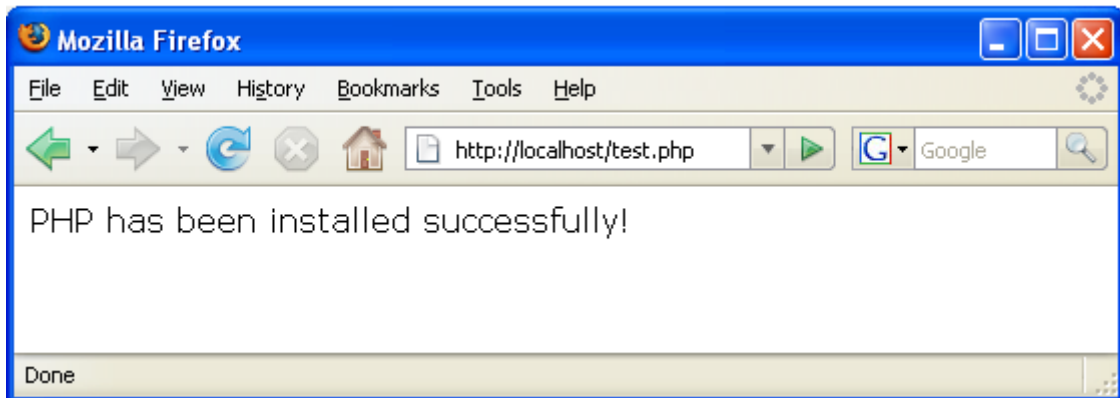


Figure 5 : Affichage du bon fonctionnement de PHP

Le plus simple dans le cas de l'utilisation de PHP dans un site web, c'est de faire appel à la fonction `phpinfo()`. Exemple, le serveur du PHP peut configurer dans le fichier **/etc/php5/apache2/php.ini**. Le fichier `php.ini` permet de configurer toute une plaquette de paramètres du langage PHP. Voici le code du PHP permettant de connaître la configuration ainsi que les variables prédéfinies du serveur :

```
< ?php  
phpinfo();  
?>
```

Le résultat de ce script se trouve en annexe 4.

III.1.3 Installation de MySQL

Un serveur de base de données sert à stocker, à extraire et à gérer les données dans une base de données. Il permet également de gérer la mise à jour des données. Il assure la sécurité et l'intégrité des données. Les bases de données servent à plusieurs fins comme :

- La gestion de documents
- La gestion de dossiers
- L'indexation pour moteur de recherche
- L'utilisation de serveurs de courriel
- La dynamisation de contenu de page Web

Le MySQL doit être installé dans un terminal tant que *root*, sur la commande :

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

Pour redémarrer MySQL, on exécute la commande :

```
# /etc/init.d/mysql restart
```

Le fichier de configuration de MySQL se trouve dans le fichier */etc/mysql/my.cnf*

III.2 Sécurisation de serveur Apache

Pour la sécurisation du serveur Apache, on a besoin de :

- L'activation de SSL
- Authentification *.htaccess*
- *Iptables* pare-feu

III.2.1 Activation de SSL

Le SSL (Secure Sockets Layer) est donc un protocole réseau qui gère l'authentification des serveurs et des clients, ainsi que la communication chiffrée entre eux. Cette sécurisation est efficace pour rentrer sur le site car elle correspond à une authentification de la demande du client par le serveur via un mécanisme de sécurité (certificat). Et il permet d'améliorer la communication cryptée et authentifiée entre le client et le serveur basée sur la cryptographie à

clé publique. Le protocole SSL a toujours été utilisé pour crypter et sécuriser les données transmises.

L'activation du protocole HTTPS pour le site implique l'obtention d'un certificat de la sécurité. Une fois le serveur authentifié, le client et le serveur établissent des paramètres de chiffrement et une clé partagée servant à chiffrer les informations qu'ils vont échanger au cours de la session. Cela permet de garantir la confidentialité et l'intégrité des données. Tout ce processus reste invisible dans l'utilisateur. Par exemple, si une page Web exige une connexion SSL, l'URL changera de HTTP en HTTPS, et une icône de cadenas apparaîtra dans le navigateur une fois le serveur authentifié. Pour activer le protocole HTTPS il suffit d'installer le paquet SSL dans le serveur web par la commande :

```
$ sudo apt-get install openssl
```

III.2.2 Authentification *.htaccess*

La première étape pour sécuriser un dossier c'est de créer un fichier *.htaccess*. Ce fichier est un moyen rapide et efficace pour restreindre, et de sécuriser un dossier et permet d'indiquer au serveur où se situent le pseudo et le mot de passe. Lorsqu'on travaille avec un serveur Apache, la présence d'un fichier particulier nommé *.htaccess* indique au serveur que les fichiers de ce dossier et tous les sous-dossiers ne doivent pas être diffusés tout à fait librement.

Pour activer *.htaccess*, on doit changer la ligne mentionnant '*AllowOverride None*' en '*AllowOverride All*' dans le fichier de la configuration apache.

```
# nano /etc/apache2/sites-available/default
```

```
< Directory "/var/www/" >
```

```
Options Indexes FollowSymLinks MultiViews
```

```
AllowOverride All
```

```
Order allow, deny
```

```
Allow from all
```

```
</Directory>
```

Une fois terminé, on redémarre le service apache avec la commande :

```
$ sudo service apache2 restart
```

III.2.2.1 Création du fichier *.htaccess*

Le code de *.htaccess* doit être placé dans le fichier */var/www/.htaccess* avec le contenu

```
AuthUserFile /etc/apache2/.htpasswd  
AuthGroupFile /dev/null  
AuthName "Page SECURISE!"  
AuthType Basic  
Require valid-user
```

Pour valider ces modes de sécurisation, on doit enregistrer le fichier texte et après on peut redémarrer le serveur Apache2 pour valider les modifications.

III.2.2.2 Création du fichier *.htpasswd*

Ce fichier *.htpasswd* contient le nom des utilisateurs autorisés à accéder à la zone protégée, ainsi que leur mot de passe encryptés. On tape la commande *htpasswd -c .htpasswd* dans l'utilisateur pour créer le fichier et ajouter un premier utilisateur. Le programme est de demandé de saisir un mot de passe, puis le vérifier en le demandant à nouveau. Par sécurité, toutes erreurs sur un fichier *.htaccess* bloquera l'accès à tous les utilisateurs voulant accéder à la partie du site sur lequel ce fichier est positionné.

Le fichier *.htpasswd* peut être édité manuellement ou par la commande. Exemple pour l'utilisateur jsmith avec le mot de passe 'awesome', dans le fichier *.htpasswd*, ce paire d'authentification ressemble à :

```
jsmith: VtweQU73iyETM
```

Pour faire l'encryptage de mot de passe, on peut utiliser le programme du PHP suivant :

```
<? php  
  
$password = 'awesome';  
$hash = crypt ($password);  
  
?>
```

Pour créer cet utilisateur jsmith dans le fichier *.htpasswd* il suffit de taper la commande :

```
# htpasswd -c .htpasswd jsmith  
  
Adding password for user jsmith  
  
New password: awesome  
  
Re-type new password: awesome
```

III.2.3 iptables pare-feu

iptables est une application en ligne de commande et il est utilisé pour surveiller le trafic entrant et sortant vers un serveur et le filtrer en fonction des règles définies par l'utilisateur, afin d'empêcher toute personne d'accéder au système. En utilisant *iptables*, on définit des règles qui n'autoriseront que le trafic sélectionné sur le serveur. Toutes les données sont envoyées sous forme de paquets sur Internet. Le noyau Linux fournit une interface pour filtrer les paquets de trafic entrants et sortants à l'aide de tableaux de filtres de paquets.

III.2.3.1 Filtrage par IP

1^{er} cas : Considérons le cas où l'on autorise uniquement une adresse IP du client 1 dans le serveur web. Dans la Figure 2, l'un de ces deux clients a une autorisation d'entrée dans le serveur, on suppose le client 1.

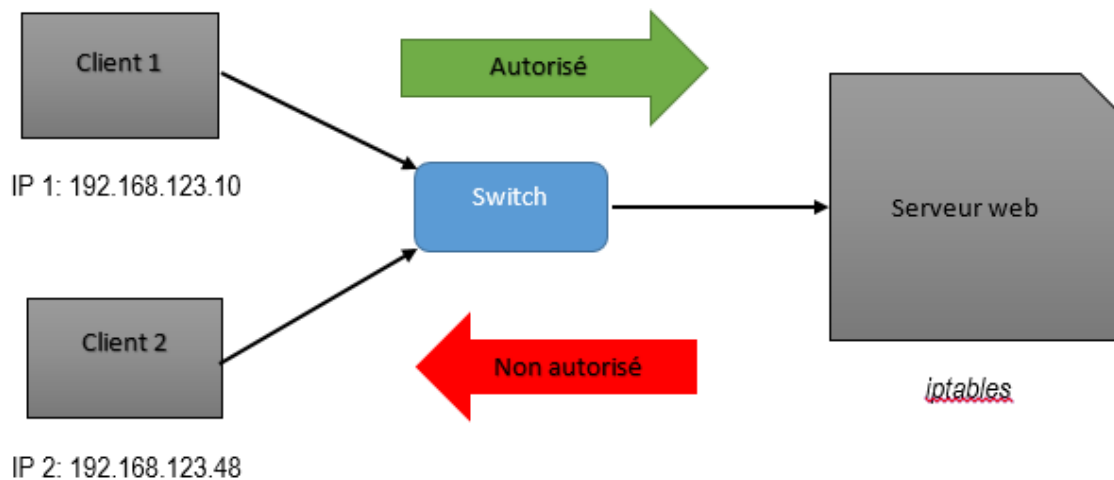


Figure 6 : Schéma d'illustration de filtrage IP 1

Soit on doit éditer d'abord un fichier avec la commande `nano firewall.sh` en y tapant les commandes pour le filtrage d'adresse IP suivantes :

```

iptables -F

iptables -t filter -A OUTPUT -p tcp -dport 80 -j ACCEPT

iptables -t filter -A OUTPUT -p tcp --dport 443 -j ACCEPT

iptables -A INPUT -s 192.168.123.2 -p tcp --dport 80 -j ACCEPT

iptables -A INPUT -s 192.168.123.48 -p tcp --dport 80 -j DROP
  
```

Puis on modifie le fichier `firewall.sh` en mode exécutable avec la commande :

```
# chmod a+x firewall.sh
```

Enfin, on exécute le fichier par :

```
# ./firewall.sh
```

Soit on tape directement sur la ligne de commande le filtrage d'adresse IP :

```

# sudo iptables -A INPUT -s 192.168.123.10 -p tcp -dport 80 -j ACCEPT

# sudo iptables -A INPUT -s 192.168.123.48 -p tcp -dport 80 -j DROP

# iptables -L
  
```

2^{ème} cas : Considérons le cas où les deux clients 1 et 2 ne sont pas autorisés à entrer dans le serveur web,

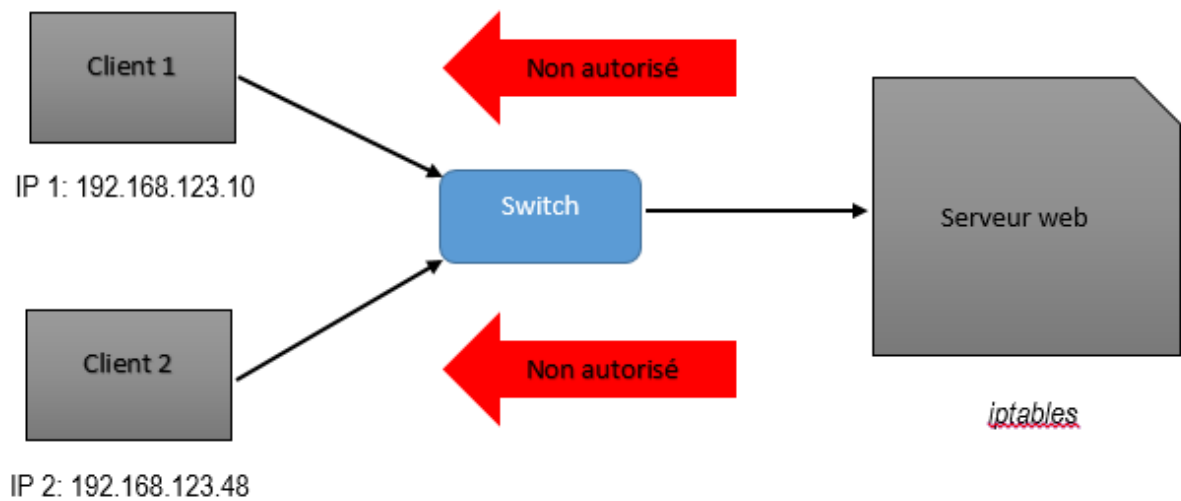


Figure 7 : Schéma d'illustration de filtrage IP 2

Dans ce cas, le script de *firewall.sh* ressemble à ceci :

```
iptables -F  
  
iptables -t filter -A OUTPUT -p tcp -dport 80 -j ACCEPT  
iptables -t filter -A OUTPUT -p tcp --dport 443 -j ACCEPT  
iptables -A INPUT -s 192.168 .123.2 -p tcp --dport 80 -j DROP  
iptables -A INPUT -s 192.168 .123.48 -p tcp --dport 80 -j DROP
```

Ou sous forme de commande le filtrage d'adresse IP se fait par :

```
# sudo iptables -A INPUT -s 192.168.123.10 -p tcp -dport 80 -j DROP  
# sudo iptables -A INPUT -s 192.168.123.48 -p tcp -dport 80 -j DROP  
# iptables -L
```

III.3 Sécurisation de base de données

Après avoir installé MySQL, le mot de passe de l'utilisateur *root* est vide. Pour entrer dans le MySQL sur Linux, on tape la commande :

```
# mysql -u login root -p
```

Dans la ligne de commande MySQL, pour ajouter ou modifier le mot de passe d'utilisateur *root* par exemple, on doit taper la commande MySQL :

```
mysql> SET PASSWORD FOR 'root'@'localhost'=PASSWORD ("*****");
```

○ Création d'utilisateur :

Pour créer le nom d'utilisateur avec le mot de passe, on doit taper sur la ligne de commande MySQL :

```
mysql> CREATE USER 'Rakoto'@'localhost' IDENTIFIED by 'password';
```

○ Attribution de privilèges :

Pour l'attribution de privilèges, on doit taper sur la ligne de commande MySQL :

```
mysql> GRANT SELECT insert on dbtot. E1 to 'Rakoto'@'localhost';
```

```
mysql> GRANT ALL privileges on dbtoto. * to 'toto'@'localhost' with grant option;
```

La plupart des sites Internet utilisent une base de données pour stocker toutes sortes d'informations comme les articles ou les comptes utilisateurs par exemple. Il va donc falloir installer MySQL qui est un Système de Gestion de Bases de Données (SGBD). Le SGBD organise les données et donne aux utilisateurs des moyens pour extraire de l'information. Or, cette information est basée sur des données comme des fonctions statiques. Les données restent privées et elles ne peuvent pas être vues par des utilisateurs non autorisés.

III.4 Résultats et discussions

La conception de serveur web sécurisé rassemble plusieurs compétences en matière de choix de paquet à installer, l'installation de paquet ainsi que la configuration nécessaire pour assurer la bonne marche de chaque module. Mise à part la conception du serveur web, notre tâche consiste à sécuriser ce serveur. Au niveau théorique et pratique, il y a beaucoup de dépendance au niveau de la démarche à faire. Après installation de serveur, rien n'est encore filtré. La Figure 8 présente un système client/serveur qui fonctionne sans filtre ou sécurisation.

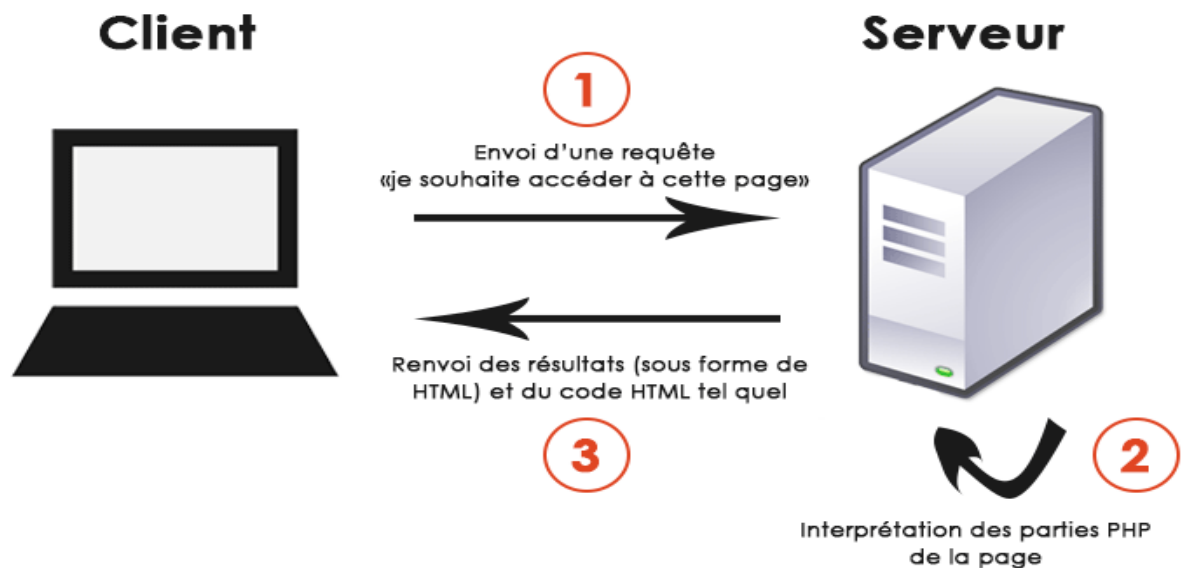
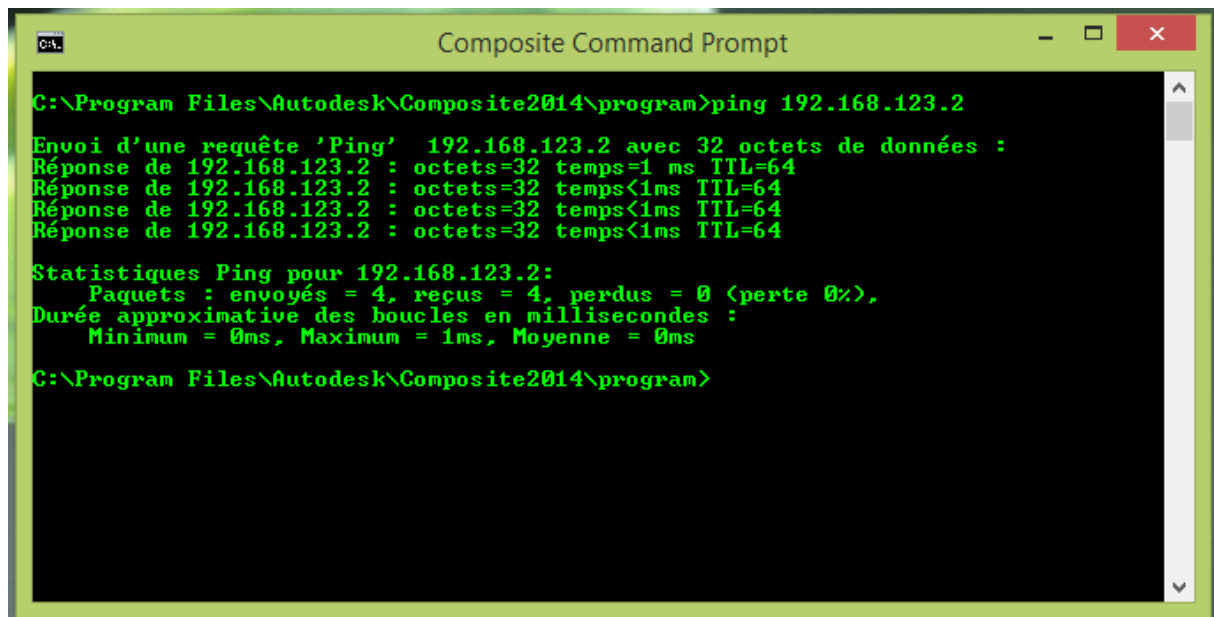


Figure 8 : Schéma du système client/serveur

Dans notre installation, deux ordinateurs ont été utilisés : le serveur et le client. Et un système d'exploitation linux *Ubuntu* a été installé pour le serveur.

Avant toute teste, il faut assurer la connexion entre la machine cliente et le serveur. Le résultat de la Figure 9 montre que la liaison entre client/serveur marche bien.



```
C:\Program Files\Autodesk\Composite2014\program>ping 192.168.123.2

Envoi d'une requête 'Ping' 192.168.123.2 avec 32 octets de données :
Réponse de 192.168.123.2 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.123.2 : octets=32 temps<1ms TTL=64
Réponse de 192.168.123.2 : octets=32 temps<1ms TTL=64
Réponse de 192.168.123.2 : octets=32 temps<1ms TTL=64

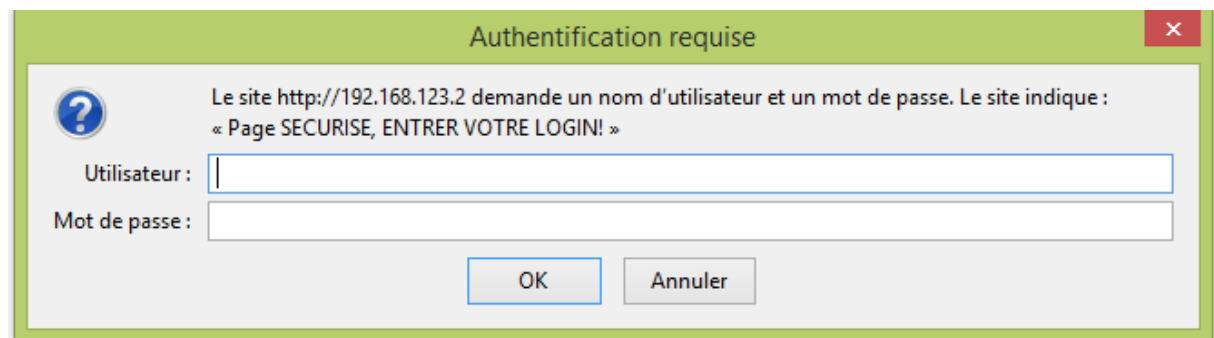
Statistiques Ping pour 192.168.123.2:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms

C:\Program Files\Autodesk\Composite2014\program>
```

Figure 9 : Résultat du test ping entre client/serveur

Pour le teste de serveur web, on ouvre le navigateur web et on tape l'adresse URL <http://192.168..123.2> et <https://192.168..123.2>

Voici donc le résultat lorsque le serveur est bien sécurisé avec le mot de passe :



Authentication requise

Le site <http://192.168.123.2> demande un nom d'utilisateur et un mot de passe. Le site indique :
« Page SECURISE, ENTRER VOTRE LOGIN! »

Utilisateur :

Mot de passe :

OK Annuler

Figure 10 : Résultat du serveur authentifié

Et lorsque tout s'est bien sécurisé ou le serveur accepte le client, on doit taper le login et le mot de passe et on est dans le serveur.

Voici donc le résultat lorsque le client est bienvenu dans le serveur :

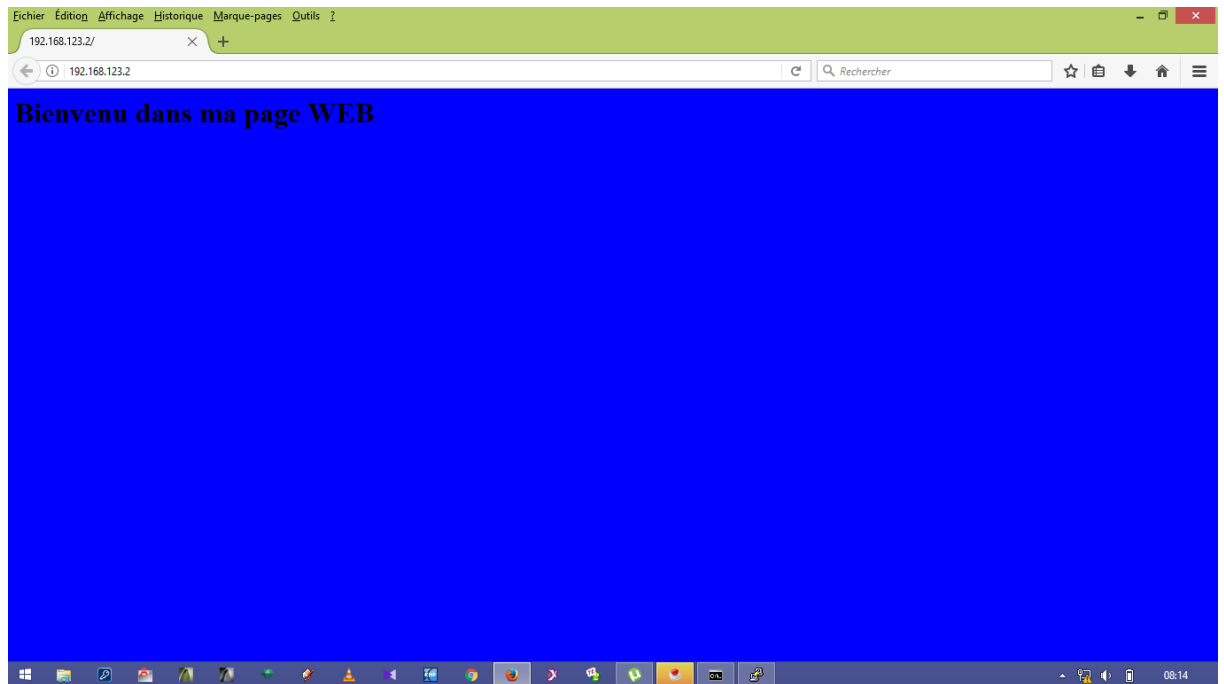


Figure 11 : Résultat du serveur dans le page web

Ce page web est programmé à partir du page html c'est-à-dire il est programmé sur la commande en ligne dans le répertoire /var/www qui est dans le fichier index.html.

Concernant le filtrage par IP, la politique de filtrage est consultable avec la commande #iptables -L

Chain INPUT (policy ACCEPT ou DROP)

Target	prot	opt	source	destination
ACCEPT	tcp	--	192.168.123.10 anywhere	tcp dpt: www
DROP	tcp	--	192.168.123.48 anywhere	tcp dpt: www

Chain OUTPUT (policy ACCEPT)

ACCEPT	tcp	--	anywhere	anywhere	tcp dpt: www
ACCEPT	tcp	--	anywhere	anywhere	tcp dpt: https

Lorsque le serveur accepte uniquement le client ayant l'adresse IP 192.168.123.10. Puis le filtrage est relayés par l'authentification de *.htaccess*. Contrairement, le client ayant l'adresse IP 192.168.123.48 est directement bloqué par filtrage d'*iptables*. Lorsque le serveur ne répond pas, le client est déjà bloqué grâce au filtrage d'adresse IP (Figure 12).

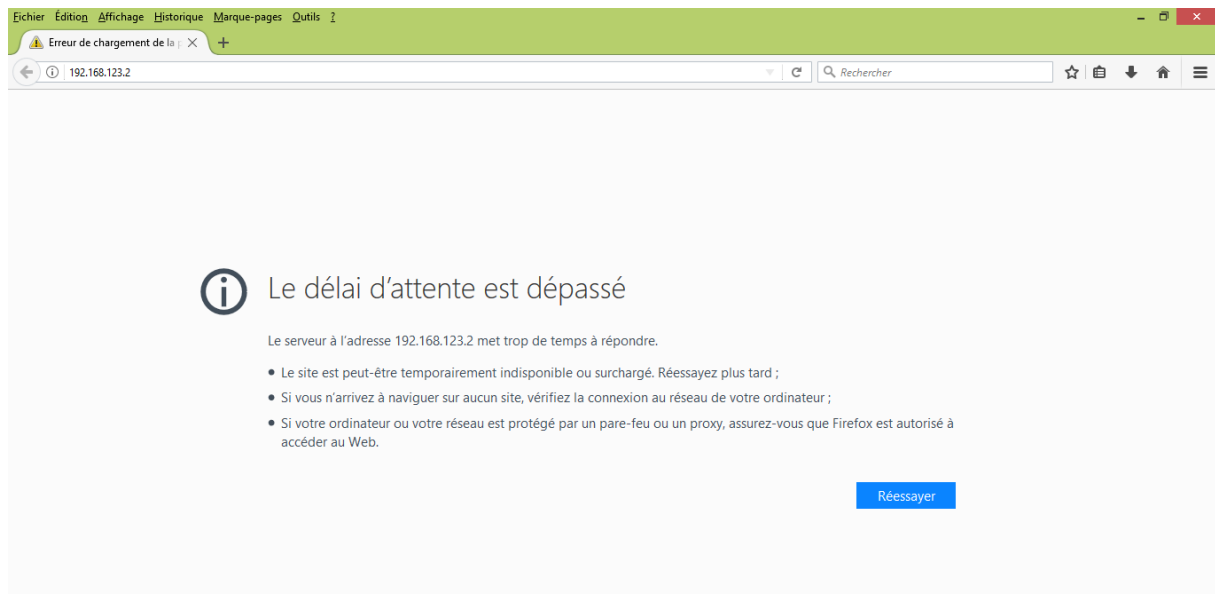


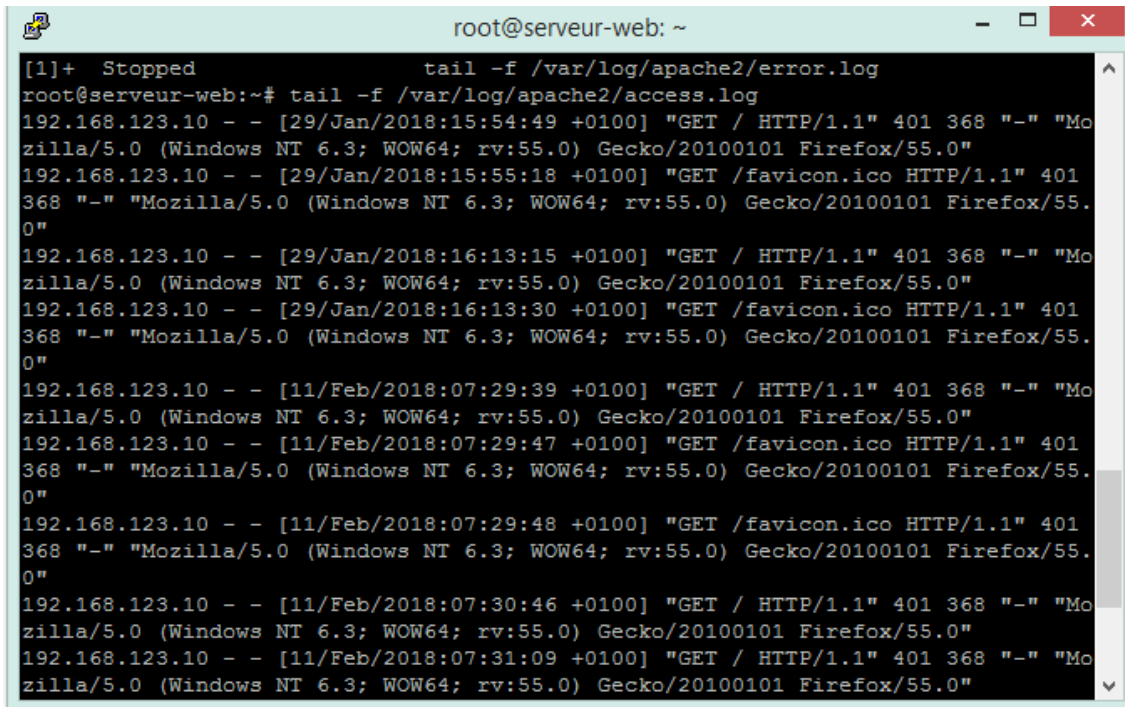
Figure 12 : Résultat du serveur lorsqu'il ne répond pas

La sécurisation du serveur web par le filtrage d'adresse IP et la sécurisation par login avec mot de passe sont l'une des plus fiables et simple à mettre en place dans le domaine de sécurisation d'un site web. Seul ceux qui sont inscrits dans le programme peuvent d'y accéder. Ceci dit beaucoup de type de sécurisation sont mise en place dans diverse société, comme la demande de login pour une ouverture d'un site web, comme le filtrage par pare-feu sous d'autre système d'exploitation, et tant d'autre selon les capacités des chaque administrateur pour la gestion de leur parc informatique.

L'entrée et/ou nonaccès au serveur peut être surveillé où justifier lorsque le client tente d'entrer dans le serveur. L'évènement log affiche les erreurs de l'envoi ou l'accès autorisé. Les fichiers *error.log* et *access.log* commande peuvent montrer la raison où lorsqu'il y a une erreur ou pas.

La Figure 13 montre bien que le client web qui a fait la requête http est autorisé.

```
# tail -f /var/log/apache2/access.log
```



```
root@serveur-web: ~
[1]+  Stopped                  tail -f /var/log/apache2/error.log
root@serveur-web:~# tail -f /var/log/apache2/access.log
192.168.123.10 - - [29/Jan/2018:15:54:49 +0100] "GET / HTTP/1.1" 401 368 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0"
192.168.123.10 - - [29/Jan/2018:15:55:18 +0100] "GET /favicon.ico HTTP/1.1" 401 368 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0"
192.168.123.10 - - [29/Jan/2018:16:13:15 +0100] "GET / HTTP/1.1" 401 368 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0"
192.168.123.10 - - [29/Jan/2018:16:13:30 +0100] "GET /favicon.ico HTTP/1.1" 401 368 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0"
192.168.123.10 - - [11/Feb/2018:07:29:39 +0100] "GET / HTTP/1.1" 401 368 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0"
192.168.123.10 - - [11/Feb/2018:07:29:47 +0100] "GET /favicon.ico HTTP/1.1" 401 368 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0"
192.168.123.10 - - [11/Feb/2018:07:29:48 +0100] "GET /favicon.ico HTTP/1.1" 401 368 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0"
192.168.123.10 - - [11/Feb/2018:07:30:46 +0100] "GET / HTTP/1.1" 401 368 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0"
192.168.123.10 - - [11/Feb/2018:07:31:09 +0100] "GET / HTTP/1.1" 401 368 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0"
```

Figure 13 : Vérification d'access.log

Quand la requête http est refusée par le serveur, les erreurs s'inscrivent dans le fichier error.log (Figure 14).

```
# tail -f /var/log/apache2/error.log
```



```
root@serveur-web: ~  
RX packets:2 errors:0 dropped:0 overruns:0 frame:0  
TX packets:2 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:0  
RX bytes:100 (100.0 B) TX bytes:100 (100.0 B)  
  
root@serveur-web:~# tail -f /var/log/apache2/error.log  
[Mon Jan 29 16:15:26 2018] [notice] caught SIGTERM, shutting down  
[Tue Jan 30 10:39:28 2018] [notice] Apache/2.2.11 (Ubuntu) PHP/5.2.6-3ubuntu4 wi  
th Suhosin-Patch configured -- resuming normal operations  
[Tue Jan 30 10:48:08 2018] [notice] caught SIGTERM, shutting down  
[Tue Jan 30 20:33:05 2018] [notice] Apache/2.2.11 (Ubuntu) PHP/5.2.6-3ubuntu4 wi  
th Suhosin-Patch configured -- resuming normal operations  
[Tue Jan 30 20:42:19 2018] [notice] caught SIGTERM, shutting down  
[Thu Feb 08 17:12:59 2018] [notice] Apache/2.2.11 (Ubuntu) PHP/5.2.6-3ubuntu4 wi  
th Suhosin-Patch configured -- resuming normal operations  
[Thu Feb 08 17:19:22 2018] [notice] caught SIGTERM, shutting down  
[Thu Feb 08 20:16:26 2018] [notice] Apache/2.2.11 (Ubuntu) PHP/5.2.6-3ubuntu4 wi  
th Suhosin-Patch configured -- resuming normal operations  
[Thu Feb 08 20:19:54 2018] [notice] caught SIGTERM, shutting down  
[Sun Feb 11 07:26:00 2018] [notice] Apache/2.2.11 (Ubuntu) PHP/5.2.6-3ubuntu4 wi  
th Suhosin-Patch configured -- resuming normal operations
```

Figure 14 : Vérification d'error.log

III.5 Statistiques de piratage de site web

Un site web est un vecteur de communication. La qualité de cette communication doit pouvoir être évaluée à tout moment. Un fonds documentaire comporte un volume d'informations qui peut vite devenir très important : mettre en avant les bonnes informations est primordial pour générer du trafic sur son site.

La statistique du nombre de visiteurs par jour peut nous donner une bonne quantité d'information sur l'efficacité de notre promotion sur Internet et de notre présence web. Il peut être très intéressant d'observer et d'analyser la fluctuation et la variation de ce nombre. La majorité des pirates exploitent les failles de sécurité dues aux absences de mise à jour des logiciels. Il existe de nombreuses manières d'hacker un site web WordPress, les pirates les connaissent mieux que tout le monde. 80 % des entreprises françaises ont constaté au moins une cyberattaque dans l'année et un cas de piratage sur un site web de Statistique Canada mène à la fermeture temporairement l'accès à plusieurs de ses services en ligne.

Voici quelques chiffres provenant d'un client depuis 2010 :

- 60% des personnes qui se font pirater un site n'ont aucune idée de comment cela est arrivé,
- 25% des pirates ont utilisé une vulnérabilité dans un plugin ou un thème,
- 6,5% sont entrés en trouvant le mot de passe par brute-force,
- 3% ont utilisé une faille venant du cœur de WordPress non à jour,
- 1,5% se sont fait piraté à cause de leur hébergeur,
- 0,6% des sites qui avaient encore d'anciens fichiers de WordPress dans leur installation,
- 0,5% à cause des mauvais droits sur les fichiers (chmod),
- 0,5 à cause du vol du mot de passe (sans brute-force)
- 0,4% se partagent d'autres raisons comme l'ordinateur sans antivirus, le serveur non à jour, le logiciel FTP désuet, etc.

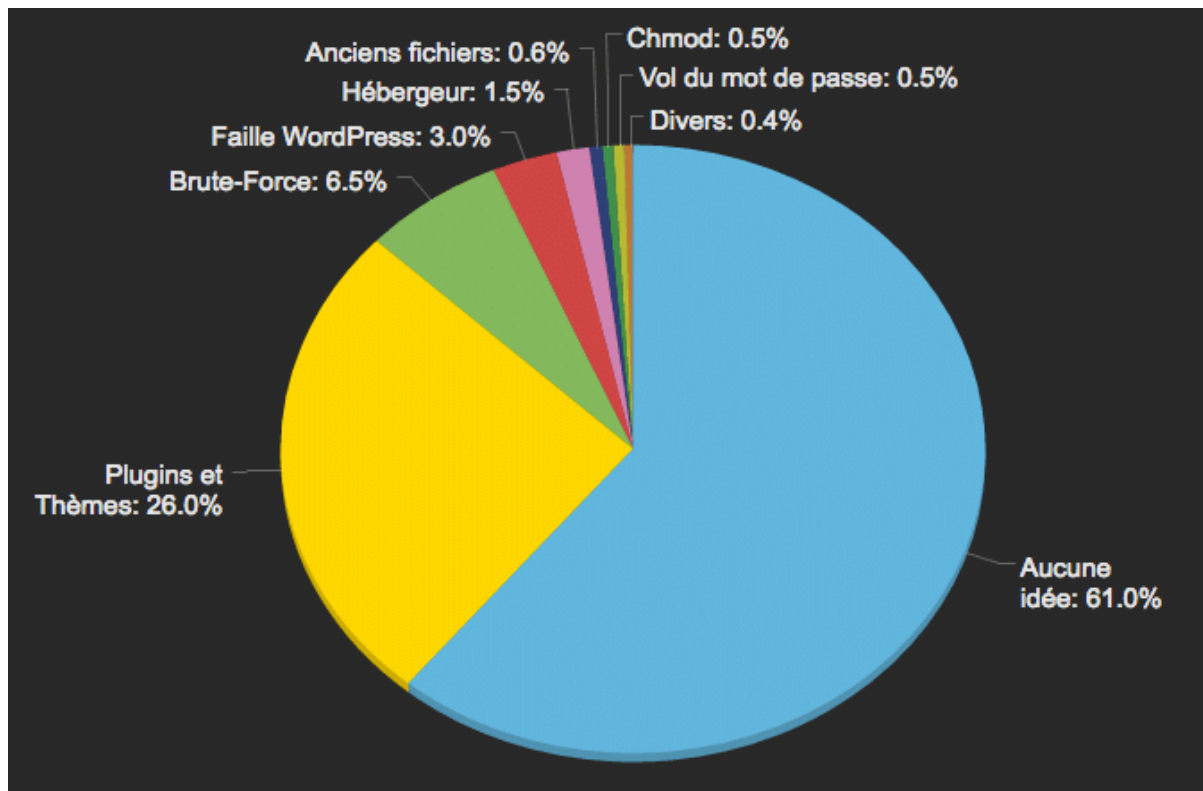


Figure 15 : Statistique des piratages d'un site web depuis 6 ans

Source : selon Tim Berners-Lee (inventeur du Web), Septembre 2014

L'origine du trafic est une statistique de site web incontournable. 5 entreprises sur 6 employant plus de 2 500 personnes ont été la cible de cyberattaques en 2014. Entre 2013 et

2014, le nombre des cyberattaques a augmenté de 120% dans le monde et le coût estimé de la cybercriminalité pour les entreprises s'élève en moyenne à 7,6 millions de dollars par an, soit une augmentation de 10%.

Selon l'enquête d'Imperva, les sites de commerce en ligne sont attaqués deux fois plus souvent que des sites plus classiques. Les attaques durent aussi plus longtemps : près de deux fois plus longtemps qu'en 2013. 40 % des attaques par injection SQL et 64 % des campagnes de trafic http malveillant concernent les sites de commerce en ligne. Après avoir analysé les données de plus de 100.000 incidents de sécurité sur 10 ans, Verizon a indiqué que 92 % des attaques peuvent être réparties en 9 types de menaces :

- Les attaques de malwares,
- La perte ou le vol d'appareils,
- Les attaques de type DoS,
- Les arnaques à la carte bancaire,
- Les attaques d'applications web,
- Le cyber-espionnage,
- Les intrusions,
- Le vol interne
- Les erreurs humaines

Ce qui signifie que les entreprises font toujours face aux mêmes risques et aux mêmes attaques, depuis tout ce temps, et à plusieurs reprises. On a donc 400 Millions de personnes sont concernées par des cyberattaques chaque année. Il est possible de prévenir les failles de sécurité avant que cela ne cause de terribles dégâts. Il vaut mieux déceler cette faille avant qu'un cybercriminel ne le fasse. Un contrôle de sécurité d'un site Web est pour cette raison la première chose à faire pour assurer la sécurité de l'activité en ligne et les données.

III.6 Analyse pour le cryptage

Le cryptage (ou le chiffrement) des connexions sur le Web se fait via le protocole HTTPS. Ce dernier n'est que l'adjonction à son aîné HTTP d'une couche de chiffrement SSL/TLS. HTTPS utilise les algorithmes de chiffrement à clefs publiques ; la gestion des clefs passe par des certificats. Les rudiments sont expliqués sur la page de notions élémentaires et sur le chiffrement (SSH, SSL, TLS, ...). Ces certificats sont signés par un tiers de confiance et l'autorité de certification (une organisation qui délivre et gère des certificats de sécurité et des

clés publiques pour le cryptage de messages). Le principe de fonctionnement des certificats SSL est basé essentiellement sur le chiffrement et l'authentification.

Pour cela, il existe deux méthodes de chiffrement ou cryptographie: cryptographie symétrique et cryptographie asymétrique.

- La cryptographie asymétrique : est une méthode cryptographique utilisant une paire de clés publique et privée combinée, pour crypter et décrypter des messages. Pour envoyer un message crypté, un utilisateur crypte un message avec la clé publique du récipient. Après la réception, le message est décrypté avec la clé privée du récipient.
- La cryptographie symétrique : est une méthode de cryptage où la même clé est utilisée pour le cryptage et le décryptage. Cette méthode est handicapée par les risques de sécurité impliqués par la distribution sûre de la clé étant donné qu'elle doit être communiquée à la fois au récepteur et à l'émetteur sans qu'elle soit divulguée à des tiers.

Le cryptage SSL (remplacé depuis par TLS) est basé sur l'utilisation du cryptage asymétrique. La cryptographie à clé publique (ou cryptographie asymétrique), est une méthode de chiffrement qui utilise deux clés qui se ressemblent mathématiquement mais qui ne sont pas identiques : une clé publique et une clé privée. A l'inverse des algorithmes de cryptographie symétrique qui dépendent d'une seule clé pour le chiffrement et le déchiffrement, les clés de la cryptographie asymétrique ont chacune une fonction bien spécifique : la clé publique sert à chiffrer et la clé privée sert à déchiffrer. Or, les clés privées restent secrètes, ce qui garantit que seul leur propriétaire peut déchiffrer du contenu et créer des signatures numériques.

L'utilisation de clés différentes pour réaliser les fonctions de cryptage et de décryptage est connue comme étant une fonction unidirectionnelle de trappe, c'est-à-dire que la clé publique est utilisée pour crypter le message, mais ne peut pas être utilisée pour décrypter le même message. Mais sans connaître la clé privée, il est pratiquement impossible d'inverser cette fonction quand un cryptage puissant et moderne est utilisé.

Par définition :

- La clé privée : est un code numérique utilisé pour décrypter des messages cryptés avec une unique clé publique correspondante. L'intégrité du cryptage dépend de ce que la clé privée reste secrète.
- La clé publique : est un code numérique qui permet un cryptage de messages transmis au propriétaire de l'unique clé privée correspondante. La clé publique peut circuler librement sans pour autant compromettre le cryptage tout en augmentant l'efficacité et **la commodité de permettre une communication cryptée.**

Le principe repose sur des fonctions mathématiques difficilement réversibles : on chiffre un message avec la clef et on déchiffre à l'aide de la même clef. On utilise une fonction et sa réciproque mais une seule et même clef. Lorsqu'un utilisateur désire envoyer un message à un autre utilisateur, il lui suffit de chiffrer le message à envoyer au moyen de la clé publique du destinataire (qu'il trouvera par exemple dans un serveur de clés). Ce dernier sera en mesure de déchiffrer le message à l'aide de sa clé privée (qu'il est seul à connaître).

Pour envoyer un message confidentiel à mon correspondant, je l'encrypte d'abord avec sa clef publique et il le décrypte à l'aide de sa clef privée. Puisqu'il est le seul à posséder cette clef, il est le seul à pouvoir lire mon message, nous avons réalisé une communication confidentielle sans échanger auparavant aucun secret. Le message ne doit être pouvoir lu que par le destinataire.

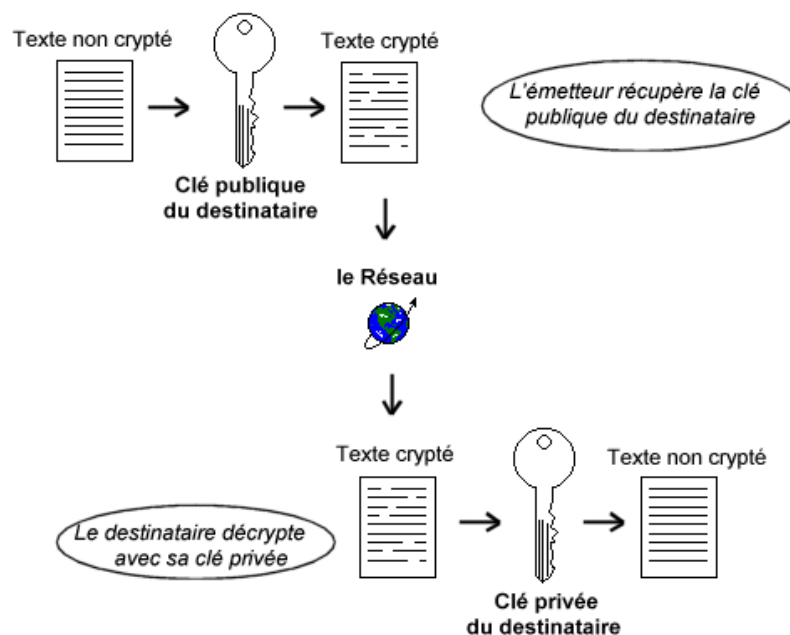


Figure 16 : Illustration sur le principe de confidentialité d'un message crypté et décrypté

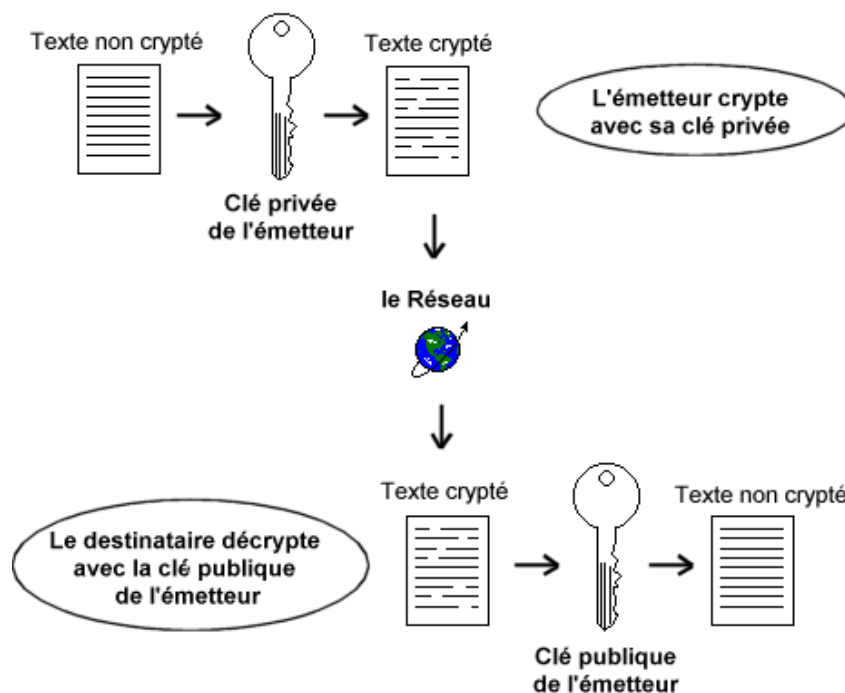


Figure 17 : Illustration sur le principe d'authentification d'un message crypté et décrypté

L'émetteur va encrypter le message avec sa clé privée. Le destinataire pourra alors vérifier l'identité de l'émetteur en décryptant le message avec la clé publique de l'émetteur. Comme seul le détenteur de la clé privée est capable de crypter un message déchiffrable par la clé publique, on est sûr de l'identité de l'émetteur.

En ce qui concerne le HTTP, il a été nécessaire de définir une nouvelle méthode d'accès dans les URL baptisée sur HTTPS pour se connecter au port d'un serveur utilisant le SSL qui porte par défaut le numéro 443.

Le mécanisme est illustré par la figure suivante :

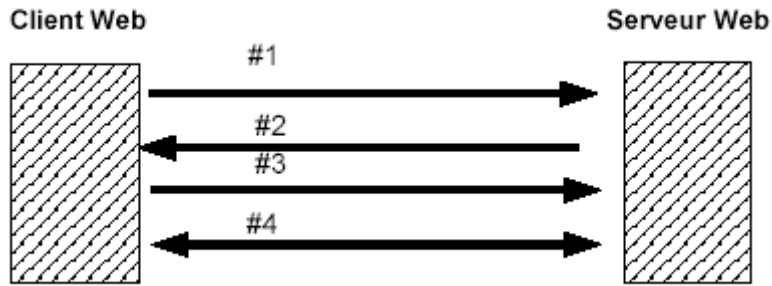


Figure 18 : Illustration de http over SSL

1 - Le navigateur fait une demande de transaction sécurisée au serveur en envoyant sa requête HTTPS://, il demande donc le certificat garantissant la clé publique du serveur.

2 - Le serveur lui envoie son certificat d'authentification délivré par une autorité de certification Ce certificat comporte une clé publique.

3 - Le navigateur s'assure tout d'abord que le certificat délivré est validé, puis il envoie au serveur une clé secrète codée issue de la clé publique. Seul le serveur sera donc capable de décoder cette clé secrète car il détient la clé privée. Cette clé secrète ainsi créée sera utilisée pour encoder les messages (cryptographie symétrique). L'algorithme à clé secrète utilisé est négocié entre le serveur et le client.

4 - Le serveur et le client possède maintenant une clé secrète partagée (la clé de session) et les échanges sont faits par l'intermédiaire de cette clé. Pour assurer l'intégrité des données, on utilise un algorithme de hash. S'il y a déconnexion, une nouvelle clé de session sera négociée.

En cryptographie, le chiffrement par décalage, aussi connu comme le chiffre de César ou le code de César est une méthode de chiffrement très simple utilisée par Jules César dans ses correspondances secrètes. Il consiste à remplacer une lettre par celle 3 rangs plus loin (A est remplacé par D, B est remplacé par E, C'est remplacé par F, etc...).

Ainsi SECRET se code VHFUHW.

Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
en clair	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Alphabet	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	0	1	2
codé	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Figure 19 : Code de César

Le chiffrement peut être représenté par la superposition de deux alphabets, l'alphabet clair présenté dans l'ordre normal et l'alphabet chiffré décalé, à gauche ou à droite, du nombre de lettres voulu. Nous avons ci-dessous l'exemple d'un encodage de 3 lettres vers la droite. Le paramètre de décalage (ici 3) est la clé de chiffrement :

- Message clair : ABCDEFGHIJKLMNOPQRSTUVWXYZ
- Message chiffré : DEFGHIJKLMNOPQRSTUVWXYZABC

Pour encoder un message, il suffit de regarder chaque lettre du message clair, et d'écrire la lettre encodée correspondante. Et pour déchiffrer, on fait tout simplement l'inverse.

- Message original : WIKIPEDIA L'ENCYCLOPEDIE LIBRE
- Message encodé : ZLNLSHGLD O'HQFBFORSHGLH OLEUH

CONCLUSION

Pour conclure, la sécurisation est une étape primordiale pour chaque entreprise, les risques sont nombreux, notamment dans un monde informatisé. La sécurité du Web est une des grandes problématiques actuelles. La difficulté est d'avoir la capacité de protéger automatiquement une application Web, c'est-à-dire être capable de filtrer les données entrantes (en ne laissant que les caractères attendus) tout en garantissant l'intégrité des données envoyées par l'internaute. En effet, par défaut il ne faut faire aucunement confiance aux données reçues et ne déléguer aucun traitement critique au niveau du client.

Un serveur web permet à des clients d'accéder à des pages web, c'est-à-dire des fichiers au format HTML à partir d'un navigateur installé sur leur ordinateur distant. La sécurisation d'un serveur web permet de protéger le système pour que les clients malfaiteurs n'aperçoivent pas ou ne peuvent y accéder.

Donc, le but de notre projet est de mettre en place un serveur web et assurer sa sécurisation qui est basée sur le cryptage en utilisant SSL, l'authentification avec *.htaccess* et le filtrage d'adresse IP par *iptables*. Ce filtrage est un outil pour élucider l'adresse IP d'un client afin que n'importe quels clients n'entrent pas dans le serveur pour voler des dossiers ou des données. Grâce à l'authentification *.htaccess* et le filtrage d'adresse IP par *iptables* notre serveur web est bien sécurisé. Ils sont très efficaces pour la sécurisation d'un serveur car ce dernier ne s'ouvre pas ni brutalement ni par la force. Vient après, les données MySQL prises comme la base qui peut sécuriser le nom de l'utilisateur avec le mot de passe en tant que *root*. Par ailleurs, l'utilisateur du serveur présente des privilèges.

Concernant la sécurisation, notre serveur a besoin d'être bien sécurisé avec le mot de passe. Elle est très efficace. Mais dans le web, il y a toujours des clients qui font des piratages de données ou de dossiers dans le serveur des autres. C'est pour cela qu'on doit sécuriser le serveur dans le but d'éviter les autres arnaqueurs d'entrer au niveau du serveur web. Posséder son propre serveur et le gérer soi-même donnent une plus grande flexibilité à son propriétaire que les serveurs dits « clé en main » ou autres serveurs mutualisés.

Ce mémoire ne traite que certains aspects de la sécurité d'un serveur web et il faudra creuser encore plus profond pour obtenir un système pouvant être à l'épreuve des attaques. Les attaques sont dues non pas à une faille sur le serveur mais à une négligence dans le développement d'un site web. Grâce à ces perspectives d'améliorations, le site web sécurisé

sera en mesure d'être utilisé sans aucun doute dans la société avec l'ajout de quelques fonctionnalités supplémentaires qui tendent encore vers un surplus de perfectionnement. Le protocole SSL est actuellement le seul protocole de sécurisation déployé et utilisé à grande échelle, son grand avantage étant sa transparence par rapport au protocole TCP. Il garantit l'authentification, la confidentialité et l'intégrité des données.

RÉFÉRENCES

BIBLIOGRAPHIQUES

- [1] Sébastien SAUVAGE (alias MWPC2), les lois françaises et internationales, protégées jusqu'aux 70 ans, 1984
- [2] Mathieu Nebra, Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms
- [3] Dew (Alsacrérations, Strasbourg), créé le 16 Mars 2008, mis à jour le 22 Septembre 2013
- [4] Sébastien Decamme, Développeur de formation
- [5] Brandon, de Prestataire informatique, le 28 Mars 2013
- [6] GeekPress, Sécurité WordPress, le 29 Mars 2013
- [7] Wikipédia Fondation, Inc, sous licence Créative Commons attribution, partage dans la même condition, le 27 octobre 2017
- [8] Thierry Lévy-Abénoli, « IPv6 : opérateurs et hébergeurs en plein chantier » sur le site zdnet.fr, le 15 avril 2011
- [9] Simon Fesnien, Mise en place d'un serveur web sécurisé (<https>), Avril 2014
- [10] Emmanuel Dreux, La sécurité sous Windows Vista, Editions ENI, 2009, 323 p. (ISBN 274604708X et 9782746047082, lire en ligne [archive]), p. 170
- [11] « Signature Numérique - La clé privée » [archive], sur le site de l'ANSSI, date inconnue (consulté le 13 septembre 2013)
- [12] Tim Berners-Lee (inventeur du Web), Septembre 2014

WEBOGRAPHIQUES

- [13] <http://sebsauvage.net>, janvier 2018
- [14] <http://www.lea-linux.org>, avril 2017
- [15] <http://glossaire.infowebmaster.fr>, novembre 2017
- [16] <http://www.php.net>, décembre 2017
- [17] <http://www.apache.org>, septembre 2017
- [18] <http://www.mysql.org>, octobre 2017
- [19] <https://www.Rapidenet.ca>, novembre 2017
- [20] https://fr.wikiversity.org/w/index.php?title=Filtrage_des_informations/Techniques_employées&oldid=679865, octobre 2017
- [21] <http://pro.clubic.com/it-business/serveur-informatique/actualite-575108-ovh-datacenter-serveur-dedie-kimsufi-3.html>, septembre 2017
- [22] <http://www.keyserver.net/>, avril 2017
- [23] <httpS://www.eila.univ-paris-diderot.fr/sysadmin/securite/ca/chiffrement>, décembre 2017
- [24] <http://blog.neocamino.com/statistique-de-site-web/#ixzz59WGgtUfc>, janvier 2018

ANNEXE 1: Installation Apache2

```
root@SRV-TEST:~# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  php5-common
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  apache2-mpm-worker apache2.2-bin apache2.2-common
Suggested packages:
  apache2-suexec apache2-suexec-custom ufw
The following packages will be REMOVED:
  apache2-mpm-prefork libapache2-mod-php5 php5
The following NEW packages will be installed:
  apache2-mpm-worker
The following packages will be upgraded:
  apache2 apache2.2-bin apache2.2-common
3 upgraded, 1 newly installed, 3 to remove and 37 not upgraded.
Need to get 3052kB of archives.
After this operation, 8499kB disk space will be freed.
Do you want to continue [Y/n]? █
```

ANNEXE 2: Installation de PHP

```
root@SRV-TEST:~# apt-get install php5
Reading package lists... Done
Building dependency tree... Done
The following extra packages will be installed:
  libapache2-mod-php5 php5-common
Suggested packages:
  php-pear php5-suhosin
The following NEW packages will be installed:
  libapache2-mod-php5 php5 php5-common
0 upgraded, 3 newly installed, 0 to remove and 41 not upgraded.
Need to get 3547kB of archives.
After this operation, 9519kB of additional disk space will be used.
Do you want to continue [Y/n]? █
```

ANNEXE 3: Installation de MySQL

```
root@SRV-TEST:/etc/apache2/sites-available# cd /
root@SRV-TEST:/# apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  php5-common
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient16 libnet-daemon-perl libplrpc-perl
  mysql-server-5.1 mysql-server-core-5.1
Suggested packages:
  dbshell libipc-sharedcache-perl tinyca
The following NEW packages will be installed:
  libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient16 libnet-daemon-perl libplrpc-perl
  mysql-server mysql-server-5.1 mysql-server-core-5.1
0 upgraded, 12 newly installed, 0 to remove and 37 not upgraded.
Need to get 24.3MB of archives.
After this operation, 61.1MB of additional disk space will be used.
Do you want to continue [Y/n]? █
```

ANNEXE 4: Configuration de PHP

PHP Version 5.3.3-7+squeeze17



System	Linux woinux.fr 3.2.13-grsec-xxxx-grs-ipv6-32 #1 SMP Thu Mar 29 09:43:21 UTC 2012 i686
Build Date	Aug 26 2013 07:26:17
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
Additional .ini files parsed	/etc/php5/cgi/conf.d/lcePHP.ini, /etc/php5/cgi/conf.d/MurmurPHP.ini, /etc/php5/cgi/conf.d/curl.ini, /etc/php5/cgi/conf.d/gd.ini, /etc/php5/cgi/conf.d/idn.ini, /etc/php5/cgi/conf.d/imagick.ini, /etc/php5/cgi/conf.d/imap.ini, /etc/php5/cgi/conf.d/mcrypt.ini, /etc/php5/cgi/conf.d/memcache.ini, /etc/php5/cgi/conf.d/ming.ini, /etc/php5/cgi/conf.d/mysql.ini, /etc/php5/cgi/conf.d/mysqli.ini, /etc/php5/cgi/conf.d/pdo.ini, /etc/php5/cgi/conf.d/pdo_mysql.ini, /etc/php5

TITRE : MISE EN PLACE D'UN SERVEUR WEB SECURISÉ

RÉSUMÉ

La sécurité informatique n'évoque pas simplement le fait de se protéger contre un éventuel piratage, elle évoque également le fait de protéger ses données contre des pertes ou altérations.

Un serveur web est un ordinateur connecté à Internet qui héberge des données et des fichiers et composé de page html. Nous avons installé un serveur web sécurisé sous Linux basé sur le cryptage utilisant le SSL, l'authentification avec *.htaccess* et le filtrage d'adresse IP par *iptables*. La sécurisation du SGBD consiste à renforcer le mot de passe de super-utilisateur root et à attribuer de privilèges aux utilisateurs de la base de données MySQL. Ce serveur sécurisé est surtout utile aux entreprises où les échanges des données sont ultra-sécurisés.

Mots clés : Serveur web, authentification, filtrage, cryptage, MySQL

ABSTRACT

Computer security is not just about protection against possible hacking. It also refers to protecting data against loss or corruption. A web server is a computer connected to the Internet that hosts data and files are consisted of html page. We set up a Linux secured web server using SSL encryption, authentication with *.htaccess* and IP address filtering by *iptables*. For database security, we reinforced the root passcode's and attribute privileges to users dealing with MySQL database. This type of server is mostly used to the company where data are drastically secured.

Keywords: Web server, authentication, filtering, encryption, MySQL

Encadreur :

M. RAKOTOARIMANANA Liva Graffin

Tel : 033 11 590 74

Maître de Conférences

E-mail : graffinliva@gmail.com

Impétrante :

RAZANANIRINA Angelica Etienne

Tel : 032 51 492 45

E-mail : angelicaetienner@gmail.com

Adresse : Bloc 63 porte 5 Ankatso II