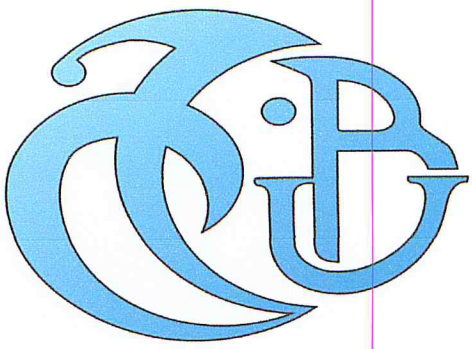
République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université De Saad Dahleb Blida-1

Faculté Des Sciences

Département d'informatique



Smart Solutions Hosting

Projet de fin d'études

Pour l'obtention du diplôme Master en Informatique

Spécialité : **Sécurité des Systèmes d'Informations**

Thème : Réalisation d'un système d'audit de sécurité pour les applications web

Organisme d'accueil : Smart Solutions Hosting

Structure d'accueil : SSH Sec

Promotrice : BOUSTIA.N

Encadrant au sein de la structure d'accueil : GUIA Brahim Fouad

Réalisé par:

DEMIAI Ahmed

DJILLALI Houssem

Président du jury : CHERIF-ZAHAR.A

Examinateur: BENYAHIA

Promotion: 2017/2018

Je dédie ce modeste travail à :

Mon modèle, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que dieu te garde dans son vaste paradis, à toi **mon père.**

A la lumière de ma vie, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur, **ma mère.**

Aux personnes dont j'ai bien aimé la présence dans ce jour, à **mes sœurs**, je dédie ce travail dont le grand plaisir leurs revient en premier lieu pour leurs conseils, aides, et encouragements.

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures, mes aimables amis, collègues d'étude.

Mes professeurs de l'USDB qui doivent voir dans ce travail la fierté d'un savoir bien acquis.

Enfin, recevez mes salutations les plus sincères.

DEMIAI AHMED.

Je dédie ce modeste travail à :

Ma mère, qui a tant œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, recevez à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude que de simples mots ne peut exprimer.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte ses fruit ; Merci pour toutes les valeurs nobles, l'éducation et le soutient permanent venu de vous.

Mes frères et sœurs qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

Mes professeurs de l'USDB qui doivent voir dans ce travail la fierté d'un savoir bien acquis.

Mme Rezoug qui nous a tant appris, et qui, sans elle on ne serait pas là où nous en sommes.

Enfin, recevez les salutations d'un fils et d'un frère qui vous a toujours aimés et que vous avez toujours aimées.

DJILLALI HOUSSEM.

On remercie d'abord Dieu le miséricordieux et le tout puissant de nous avoir permis et donner la force d'arriver là où nous en sommes.

On voudrait tout d'abord remercier notre promotrice Ph.D Boustia Narhiman pour sa patience, sa disponibilité et surtout ses bons conseils qui ont amélioré notre façon de travailler.

On remercie notre encadreur et directeur de stage à SSH Mr. Brahim Guïa Fouad qui nous a beaucoup aidé et dirigé dans nos recherches et notre travail, mais aussi pour toute l'expérience qu'il nous a apporté, sans laquelle nous n'aurions pas pu atteindre notre but.

Sans oublier la gratitude qu'on doit à Mr. DERRAR qui nous a honorés en présidant ce jury ainsi qu'à Mr. BENYAHIA pour nous gratifier de sa présence en tant qu'examinateur.

Aussi on désire remercier tous nos professeurs qui ont contribué à notre formation et qui ont enrichie notre ignorance de leur savoir.

On remercie aussi tous ceux qui nous ont soutenus et encouragé au cours de notre parcours en tant qu'étudiants, ou bien qui nous ont aidé dans le cadre du travail accomplis.

Résumé

Ces dernières années, l'évolution d'internet fut exponentielle, les services en ligne sont importants, partant de la simple messagerie instantané jusqu'au paiement en ligne et transaction monétaire. Toutes ces nouvelles fonctionnalités ont vu le jour grâce aux nouveaux langages dynamiques, d'où l'obtention de sites plus interactifs.

De même Le Web, de plus en plus de sites et d'applications Web apparaissent chaque jour. De ce fait, Le Web est aujourd'hui le vecteur d'attaque le plus prisé des cybercriminels. La raison est très simple. Avec des millions de sites Internet à leur disposition, la toile est un terrain d'actions illimitées. C'est pourquoi il est indispensable de considérer une sécurité appropriée avant de mettre en ligne de telles applications. Il est donc nécessaire d'auditer régulièrement les applications Web pour vérifier la présence De vulnérabilités exploitables et ceci, peut être réalisé notamment par des « SCANNER » de Vulnérabilités Web.

On envisage la réalisation d'une solution logicielle pour l'audit et la détection de vulnérabilité des applications web Avant la mise en ligne et afin d'anticiper tout blocage ou dysfonctionnement lors de la Mise en service d'une Solution Web.

Cette solution sera un dispositif WEB qui va permettre à SSH (Smart Solutions Hosting) d'auditer les sites de leurs clients et de consulter les corrections pour chaque vulnérabilité détectée, et prendre ainsi les mesures correctives indiquées avant le lancement.

Mots clés : Sécurité des applications web, Scanner web, Audit de vulnérabilités.

Liste des figures et tableaux

Figures

Tableaux

Introduction Générale

"As with any new class of technology, web applications have brought with them a new range of security vulnerabilities." - Dafydd Stuttard & Marcus Pinto [26]

Internet ne cesse d'évoluer, les services en ligne sont de plus en plus nombreux, et de plus en plus varié allant jusqu'aux paiements et transactions monétaires en ligne. Toutes ces nouvelles fonctionnalités sont devenus accessibles grâce aux nouvelles technologies qui permettent l'obtention de sites plus interactifs notamment les nouveaux langages dynamiques.

Du fait que de plus en plus de sites et d'applications Web apparaissent chaque jour en utilisant ces nouvelles technologies.

Comme (Dafydd Stuttard & Marcus Pinto [26]) l'ont affirmé dans leur livre, avec des millions de sites Internet utilisant les nouvelles technologies, le web est aujourd'hui le vecteur d'attaque le plus prisé des cybercriminels, et la toile est devenu un terrain d'actions illimitées. C'est pourquoi il est indispensable de considérer une sécurité appropriée avant de mettre en ligne de telles applications.

Il est donc primordial d'auditer les applications Web afin d'identifier les vulnérabilités exploitables, pour ceci il existe des outils notamment les « SCANNER » de vulnérabilités Web.

Smart Solutions Hosting étant une entreprise de développement et d'hébergement qui garantit une SLA (*Service Level Agreement*) qui s'agit d'une clause contractuelle qui définit les objectifs précis et le niveau de service qu'est en droit d'attendre un client de la part du prestataire signataire, par conséquent SSH s'engage à assurer ses services sans défaut et surtout en matière de sécurité du stockage et la gestion des données personnelles du client.

Compte tenu de ce qui précède, Smart Solutions Hosting envisage la conception et la mise en œuvre d'une solution logicielle pour l'audit et la détection de vulnérabilité des applications web avant la mise en ligne de leurs solutions web et afin d'anticiper tout blocage ou dysfonctionnement lors de leur mise en service.

Notre solution sera un dispositif WEB qui va permettre d'auditer les applications web notamment leurs sites afin d'identifier les vulnérabilités et y pallier.

Afin de bien mener notre étude, nous avons organisé le projet de fin d'étude en cinq chapitres :

Le premier chapitre est composé de deux parties, sur la première nous présentons l'organisme d'accueil (Smart Solutions Hosting), quant à la deuxième partie nous parlerons des différents moyens employé par SSH pour assurer la sécurité de ces plateformes.

Dans le deuxième chapitre nous parlerons sur l'univers d'internet, des applications web ainsi que l'évolution et le fonctionnement de ces dernières.

Dans le troisième chapitre nous allons aborder les généralités de l'aspect sécuritaire des systèmes d'information, nous présenterons aussi les différentes failles les plus répandues des applications web, la façon dont elles sont exploitées a été évoquées afin de mieux comprendre leurs fonctionnement.

Apres avoir étudié ces failles, vient la phase de conception ou, nous décrirons le fonctionnement général de notre scanner, avec sa décomposition en quatre modules distincts et l'explication du fonctionnement de chacun-deux.

Enfin le chapitre cinq se focalise sur la réalisation du scanner à savoir les outils de développement ainsi que certaines de ces fonctionnalités offertes et bien sûr nous ferons des tests de comparaison avec un des meilleurs scanners web disponibles sur le marché.

Chapitre I

Etude de l'existant

I. Etude de l'existant

A. L'organisme d'accueil (Smart Solutions Hosting) :

1. Présentation :

Smart Solutions Hosting (SSH) est une société Algérienne qu'on pourrait définir comme un *provider* (fournisseur) de solutions cloud et digitales pour entreprises comme pour particuliers, leurs services s'étendent de l'hébergement CLOUD aux solutions de gestion pour entreprises, en passant par le développement web et le E-marketing.

L'idée de cette entreprise a vu le jour en réponse à toutes les avancées technologiques, et principalement à la demande croissante en exponentiel des services de création de sites Web, et d'hébergement en général, que ce soit pour les particuliers ou les entreprises, ce qui signifiait un marché florissant et potentiellement stable.

Du fait de l'intérêt croissant que portait et continue de porter SSH au marché Algérien de l'hébergement et des solutions Web en générale, l'entreprise a été créée en 2015, par un groupe d'experts en sécurité des systèmes d'informations et de développeurs toutes plateformes, son activité a commencé le 20 Décembre 2015 avec l'ouverture de son agence principale à Staouali.

La société use des technologies *Cloud* pour ses services et donc ils ont une fiabilité maximale en termes de sécurité et en disponibilités des services alloués.

2. Les activités et les Services de SSH :

Vision : La vision de SSH est d'améliorer et inculquer la culture des services Web à la communauté Algérienne ce qui implique une contribution au développement technologique du pays.

Mission : Assurer aux entreprises et même au particuliers un large éventail de services qui vont de l'hébergement web jusqu'aux solutions de E-marketing, en passant par le développement web et mobile, tout ceci dans un environnement fiable.

Objectifs :

- Mettre au service de la clientèle Algérienne, les technologies du *Cloud*.

- Offrir des services d'hébergement personnalisés de qualité pour la clientèle appartenant aussi bien au secteur public qu'au secteur privé.
- Promouvoir et mettre en place de nouveaux services High-Tech.
- Répondre aux besoins de la clientèle dans tous les domaines.
- Conseiller et assister la clientèle dans la réalisation de leurs systèmes d'informations.
- Œuvrer en général à faire de SSH une institution performante en tous domaines et devenir numéro 1 en Algérie.
- Avec une stratégie bien définie, une mobilisation de ses différents organes et une volonté ferme de ses employés, SSH est bien positionné pour saisir les opportunités et les défis qui se profilent à l'horizon.

Produits & Services : SSH fournit ses services sous différentes branches : Le Développement, L'Infrastructure, La Sécurité, Le Marketing

- Le Développement :

  SSH offre des services de développement multi plates-formes :

  - Mobile IOS :

    C'est le développement d'applications pour les appareils fonctionnant sous system d'exploitation IOS d'*Apple*.

  - Mobile Android :

    C'est le développement d'applications pour les appareils fonctionnant sous system d'exploitation *Android*.

  - Mobile Windows :

    C'est le développement d'applications pour les appareils fonctionnant sous system d'exploitation *Windows Phone* de *Microsoft*.

  - Sites Web et Applications Web [Intranet/SaaS] :

    SSH offre aussi comme service le développement et la création de sites Web et Applications Web interactives en Intranet ou en SaaS (*Sofware as a Service*).

- L'Infrastructure:

  SSH propose aussi les infrastructures suivantes :

- Hebergement Mutualisée :

  Un site web partage les ressources d'un serveur ou plus généralement d'un groupe de serveurs avec d'autres sites web. C'est l'hébergement de base d'un site web qui suffit souvent dans la plupart des cas.

  Pour cette infrastructure Toutes les interventions techniques sont à la charge de l'hébergeur.[1]

- Hebergement Dédiée :

  Un hébergement sur serveur dédié vous est entièrement réservé. Vous avez l'entière responsabilité de la machine et des programmes, logiciels et sites que vous installez.

  Cette solution est réservée aux sites ayant une audience importante et un fort contenu dynamique. Un serveur dédié demande également du temps et des connaissances pour son administration.[2]

- Hebergement VPS (*Virtual Private Server*):

  Un hébergement sur serveur virtuel est une solution intermédiaire entre l'hébergement mutualisé et dédié : Le serveur virtuel se comporte (théoriquement) comme un serveur dédié, mais sur une infrastructure mutualisée spéciale. Cette infrastructure se charge de toujours fournir les ressources CPU et mémoire selon les configurations des serveurs virtuels.

  Cette solution est intéressante car elle élimine une grande part des interventions techniques de mise à jour des composants qui est à la charge de l'hébergeur.[3]

- Infogerance :

  SSH propose un service où un client (notamment une société) délègue la gestion, l'exploitation, l'optimisation et la sécurisation de son système d'information.

- La Sécurité :

  SSH assure aussi la sécurité pour ses clients :

- *Pentesting* (Test d'intrusion) :

  Le test d'intrusion est une partie de l'audit de sécurité et consiste à se mettre dans la peau d'un attaquant souhaitant s'introduire dans le système d'information pour y effectuer des méfaits (espionnage, sabotage, etc.). Comme son nom l'indique, le

[1] http://www.binghost.com/def-hebergement-mutualise.php (date 2018)
[2] http://www.binghost.com/def-hebergement-dedie.php (date 2018)
[3] http://www.binghost.com/def-hebergement-virtuel.php (date 2018)

test d'intrusion vise à s'introduire sur le réseau ou dans une partie spécifique du réseau.[4]

- *Forensic* (Investigation):

  C'est l'application des techniques d'investigation et d'analyse pour rassembler des évidences (preuves/traces) à partir d'un appareil informatique, généralement de manière à ce que ces évidences soient recevables dans une cour martiale. Le but de cette « investigation numérique » est d'effectuer une enquête structuré par des protocoles qui décrivent les étapes à faire, tout en conservant des preuves documentés, afin de qu'est-il arrivé à l'appareil informatique et qui en est responsable.

- *CodeReview* (Audit du code):

  L'audit du code est une pratique qui consiste à parcourir un code source à la recherche de vulnérabilités ou un non-respect des règles de bonne pratique. Cette approche permet de déceler un grand nombre de vulnérabilités à la source, et est plus rapide et complète qu'un test d'intrusion.

  Configuration Audit (Audit de configuration) :

  Un audit de configuration a pour objectif de vérifier le paramétrage d'un équipement ou d'une solution technique par rapport aux risques de sécurité. Le résultat de l'audit permet de déterminer si l'implémentation technique auditée est conforme aux bonnes pratiques de sécurité et ne présente pas de risque pour le reste du système d'information.[5]

- Le Marketing :

  SSH offre des services de développement multi plates-formes :

- SMSing (SMS marketING):

  Le SMS marketing regroupe toutes les formes d'usage du SMS dans le cadre d'objectifs commerciaux ou marketing.

- *Emailling* :

  L'emailing désigne généralement les campagnes de marketing direct effectuées par email.

- *FaceBook-Ads* :

[4] https://www.information-security.fr/test-dintrusion-pentest-presentation-methodologies/ (date 2018)
[5] https://sysdream.com/audits/audit-de-configuration/ (date 2018)

Facebook Ads est une fonctionnalité offerte par Facebook pour promouvoir ou annoncer une page de fans qui a déjà été faite par des utilisateurs de Facebook avec une portée différente et peut être définie par l'annonceur.Dans la publicité sur Facebook , l'annonceur doit avoir une page avant une page de fans qui permet aux autres utilisateurs de Facebook de donner le LIKE ou devenir des fans sur la page Fan et Fan Page peuvent être une entreprise, services produits, particuliers, marques, etc.

- *Google-Ads* :

Google-Ads ou Google adwords es la régie publicitaire de google qui permet aux annonceurs d'attirer plus de visiteurs à leurs sitewebs pour accroitre leurs business. Les annonceurs paient lorsque l'internaute clique sur la publicité selon un système d'enchère et de qualité.

- *Community management* :

L'activité de *community management* désigne l'activité de gestion de la présence d'une marque ou organisation sur les réseaux sociaux et autres espaces communautaires.

3. Organigramme de SSH :

SSH nous a fourni l'organigramme suivant :

- Le conseil d'administration : est composé de 3 membres qui sont les fondateurs.
- Chaque service dispose d'éléments actifs supervisés par un responsable de service.
- Le service Marketing a des sous-services qui sont des divisions affiliées.

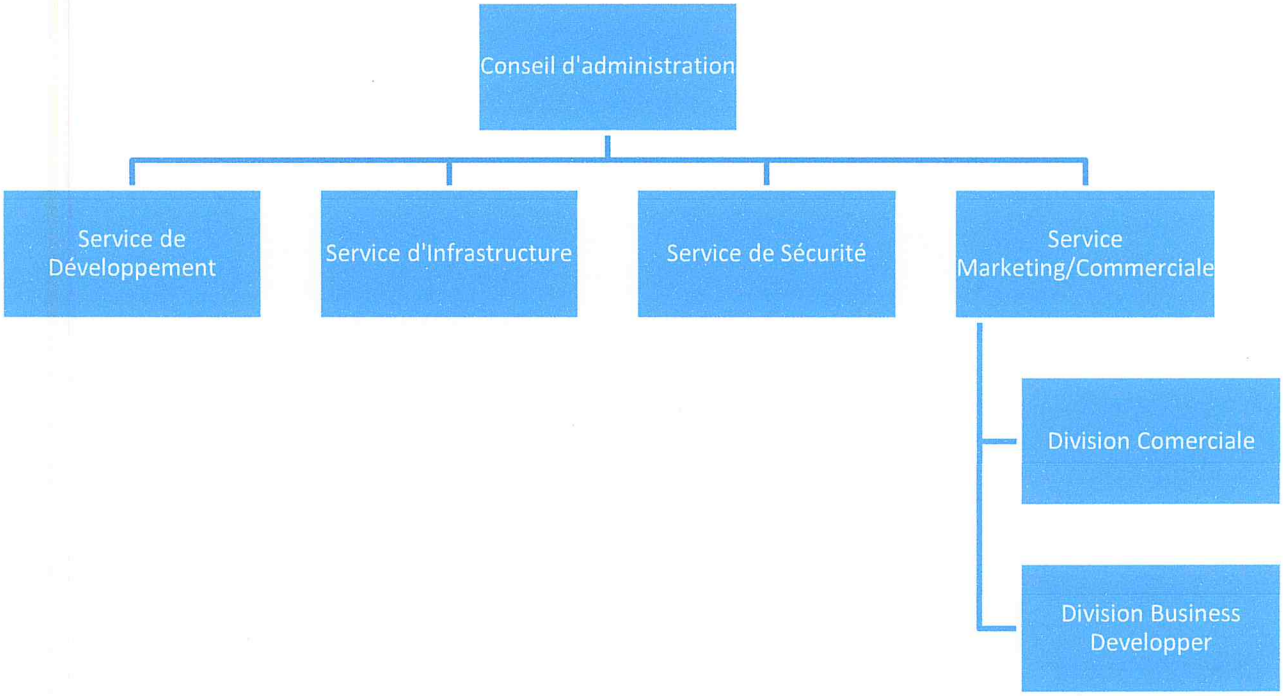


Figure I-1 Organigramme de SSH

B. Présentation de la structure d'accueil (SSH Sec) :

En raison de la nature de notre sujet on a été affectée au niveau du service de sécurité de SSH (SSH Sec) où on a eu le droit d'accès à toutes les informations auprès des personnes concernée de près ou de loin par le projet.

Organigramme de SSH Sec :

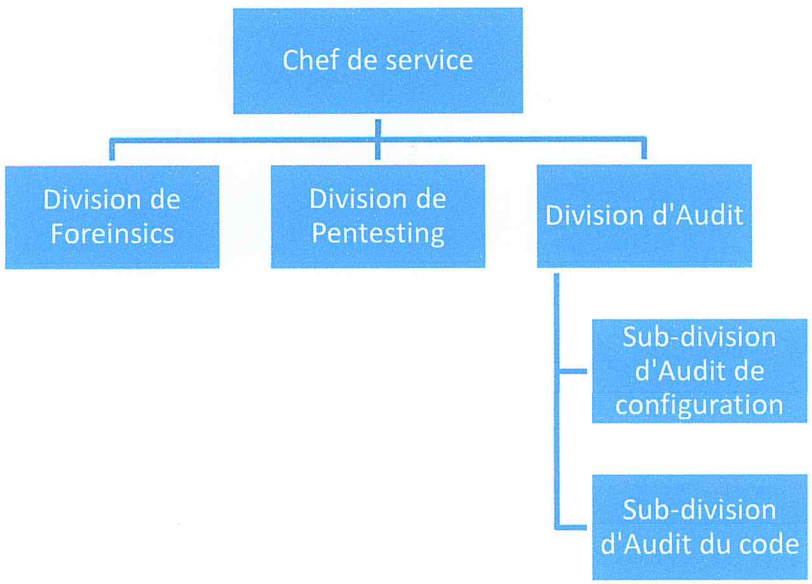


Figure I-2 Organigramme de SSH Sec

Les Interlocuteurs :

La réalisation du présent projet, a impliqué des réunions préliminaires dont l'objectif est de recueillir auprès des responsables concernés un ensemble d'informations nécessaires à sa parfaite mise en œuvre.

Ces séances de travail ont été faites avec Mr. Guia Fouad afin de définir :

- Les besoins de chaque responsable pour avoir différents points de vue et connaitre les détails technique de la plateforme et du réseau existant.
- L'organisation de l'entreprise.
- Les ressources consacrées au projet.
- Afin de mieux connaitre les difficultés rencontrées actuellement et les besoins et améliorations souhaités.

C. Etude de l'existant :

La sécurité des plateformes est assurée par deux niveaux de firewalls hardware en redondance à chaque niveau, supportant un nombre de connexions simultanées supérieur à 2 millions, puissants, imprégnant les leaders mondiaux dans les solutions de sécurité des réseaux à savoir Juniper et Cisco ;

En plus des firewalls réseau, deux Firewalls applicatifs sont intégrés dans la plateforme de services permettant le partage de charges entre les serveurs, offrant le même service, et protégeant ces derniers des attaques au niveau applicatif destinées aux applications Web.

Aussi, Les plateformes disposent de deux sondes de détection et de prévention des intrusions, des solutions antivirales, des certificats SSL, et d'une solution de cache.

1. Les niveaux de firewall réseau :

Les deux niveaux de sécurité permettent de sécuriser le trafic d'une manière à ce que même si une faille est survenue sur le premier niveau, elle ne puisse pas affecter le deuxième niveau et vis-versa.

- Premier niveau de firewall (Juniper) :

Ce niveau est composé de deux firewall Juniper en redondance qui fonctionnent en mode actif/actif, permettant d'exploiter toutes les ressources des deux boitiers, ce sont des firewalls de type *statefull inspection*, puissant et performant traitant un trafic pouvant aller jusqu'à 6 Gbps afin de garantir la sécurité entre le trafic du réseau interne et public.

- Deuxième niveau de firewall (CISCO) :

  Ce niveau est composé de deux firewall Cisco en redondance, fonctionnant en mode actif/actif, les firewalls Cisco combinent les meilleurs services de sécurité et de VPN pour constituer une solution de sécurité spécifique, ils gèrent un trafic pouvant atteindre 2Gbps et traitant le trafic entre le premier niveau (firewalls Juniper) et les serveurs.

2. Les firewalls applicatifs :

Les plateformes de services disposent aussi de deux boîtiers F5-BIG-IP en redondance Actif / Standby, assurant les fonctionnalités suivantes :

- Cache proxy ou reverse proxy
- Equilibrage de charge
- Accélération SSL

3. Sonde de prévention d'intrusion :

Les sondes de prévention d'intrusion déployées sont les IPS de CISCO en redondance en mode actif/actif, qui identifient avec précision et stoppent l'activité malicieuse, comme les vers, les attaques ciblées, le déni de service (DDOS) et les violations de protocoles, ils surveillent et analysent chaque paquet de données entrant et sortant vu qu'ils sont déployés en mode in-ligne (coupure) pour agir directement sur le trafic.

4. Solutions antivirale :

Afin de se protéger contre les attaques virales à savoir les virus, les vers, les chevaux de trois, et pour ne pas s'infecter ni infecter les internautes qui accèdent aux services, une solution antivirus est acquise, reconnus mondialement (Kaspersky).

Cette solution est capable de protéger l'ensemble des serveurs.

5. Solutions anti-spam :

En plus de la solution d'antivirus déployée au niveau des serveurs et particulièrement sur la messagerie, cette dernière dispose de nombreux moyen de lutte contre le spam à travers une solution anti spam et qui consiste principalement à contrôler les expéditeurs en fonction de leur IP, en consultant des systèmes de *blacklist* (RBL).

6. Certificat SSL :

SSH peut offrir ces services via son site sécurisés en HTTPS à travers l'acquisition de certificats SSL, permettant le cryptage des données transitant sur le réseau évitant les attaques « man in the middle » et pour que l'internaute soit sur d'être connecté au site de confiance demandé évitant les attaques et le vol d'identité,

Plusieurs types de certificats SSL sont installés actuellement sur les serveurs sensibles des plateformes (messagerie, serveurs web d'authentification…) issue des autorités de certification reconnues mondialement à savoir verisign, comodo et geotrust

7. Solution de cache :

Les plateformes sont équipées de deux boitiers Bluecoat en redondance (Actif/Standby) déployés pour assurer la mise en cache des différentes mises à jour des systèmes et des applicatifs des plateformes, cette mise en cache augmente le niveau de sécurité des serveurs en leur permettant un téléchargement indirect (sans accéder à internet) en plus un gain en bande passante.

8. Stockage :

Les données des plateformes sont stockés dans des baie de stockage installé au niveau de la plateforme offrant un espace de stockage important qui dépasse les 140 TB en brut avec un haut niveau de sécurité et un ensemble de dispositifs redondants échangeables à chaud :

- Les contrôleurs sont doublés, avec des caches en miroirs, chaque contrôleur dispose d'alimentation doublée, de batteries et de ventilations sécurisées.
- Les contrôleurs et leurs caches sont synchronisés entre les contrôleurs HSV au travers de deux liens Fibre- Channel full-duplex, dédiés et redondants. En cas de défaut sur ces liens, les liens disques sont employés pour garantir la continuité d'exploitation.
- Les disques utilisés disposent de double port FC. Chaque contrôleur accède aux 2 ports FC des disques pour une redondance et des performances complètes.
- Les étagères de disques disposent de ventilateurs et d'alimentations redondantes.

9. Sauvegarde :

Les données de la plateforme d'hébergement et messagerie dispose d'un outil très puissant pour la sauvegarde et restauration spécialement conçue pour les environnements intra entreprise et les environnements partagés.

Il offre les avantages suivants:

- Une protection fiable des données.
- Une grande facilité d'accès aux données des applications.
- Une nouvelle approche de la sauvegarde par l'utilisation de fonctionnalités avancées des baies de disques de nouvelle génération. Cette fonctionnalité permet une restauration en quelques minutes au lieu de quelques heures initialement.
- Une solution de sauvegarde sur disque
- Une solution de réorganisation des bandes en vue d'une restauration rapide
- Nouvelle méthode de sauvegarde en « incrémental » continue grâce à la nouvelle fonction « *synthetic full* ».
- Une architecture évolutive
- Une administration facile et centralisée
- Une installation facile pour les environnements mixtes
- La grande disponibilité des données
- Une procédure de restauration facile
- Opération automatisée ou sans surveillance
- Surveillance, rapports et notifications
- L'intégration avec les applications de base de données en ligne

D. Conclusion :

SSH fournit des services pour les administrations, entreprises, institutions publiques et privées, etc.., Sur des serveurs Cloud de renommée mondiale, et l'un d'eux est l'hébergement de sites et applications Web.

Pour assurer la sécurité et l'intégrité de ces site et applications Web, SSH a opté pour les meilleures solutions du marché et les efforts de ses experts en sécurité, mais leur

infrastructure numérique devenant de plus en plus complexe et interconnectée augmente de façon exponentielle la difficulté à parvenir à une sécurité des applications et sites Web.

Compte tenu des risques, SSH Sec envisage la conception et la mise en œuvre d'une solution logicielle pour l'audit et la détection de vulnérabilité des sites et applications web de leurs clients avant leur déploiement sur les serveurs.

Objectifs du Projet :

Afin de résoudre les problèmes cités auparavant et pour être en mesure de répondre aux différents besoins du responsable, l'entreprise s'est engagée dans une démarche de développement d'un système d'audit de vulnérabilités automatisé.

Notre travail est donc de réaliser ce système, dont le but principal est de simplifier l'audit de sécurité des différentes applications avant leur déploiement, et de permettre automatiquement un *reporting* complet des failles.

Les objectifs de notre travail visent 5 points :

- L'intégrité, c'est-à-dire garantir que les données sont bien celles que l'on croit être ;
- La confidentialité, consistant à assurer que seules les personnes autorisées aient accès aux ressources échangées ;
- La disponibilité, permettant de maintenir le bon fonctionnement du système d'information ;
- La non répudiation, permettant de garantir qu'une transaction ne peut être niée ;

L'authentification, consistant à assurer que seules les personnes autorisées aient accès aux ressources.

Chapitre II

Evolution Du Web

II. L'évolution du web

A. Introduction :

Internet, au tout début un petit réseau de communication, a connue des avancées fulgurantes durant seulement quelques années principalement due au potentiel qu'il montrait en matière de traitement d'information. L'internet a connu une révolution en s'emparant de notre vie au quotidien, engendrant un changement fondamentale de nos habitudes en apportant de nouvelles façons de penser et de traiter l'information.

B. Chronologie du Web :

Depuis le début d'Internet, le Web a évolué par paliers et plusieurs phases se sont succédé avant d'obtenir les applications en ligne que l'on utilise aujourd'hui.

Au début du Web, les pages HTML se limitaient à l'affichage de simples textes et à quelques illustrations (dont l'affichage était d'ailleurs souvent bloqué pour améliorer la fluidité de la navigation sur les réseaux à faible débit de l'époque). Au fil des années, avec l'avènement d'Internet tel qu'on le connaît, les exigences des utilisateurs ont évolué. En effet, la seule interactivité possible sur les pages HTML était l'affichage d'une nouvelle page lors d'un clic sur un lien hypertexte. Il était donc impossible d'obtenir le moindre effet visuel sans avoir recours à une technologie complémentaire. De plus, l'envoi d'une requête au serveur Web, suite à un clic sur un lien hypertexte par exemple, engendrait un cycle de traitement long et fastidieux qui freinait considérablement la réactivité des applications sur des réseaux et des serveurs souvent sous-dimensionnés pour le trafic sans cesse croissant de l'époque.

1. Les sites statiques :

Les sites statiques sont constitués d'un ensemble de pages HTML reliées entre elles par des liens hypertextes qui permettent de naviguer de l'une à l'autre. Le protocole utilisé pour transférer des informations Web sur Internet s'appelle **HTTP**. Une requête HTTP (http://www.mon-site.com/page.htm par exemple) est envoyée vers le serveur afin d'accéder à la page désirée et de la visualiser dans le navigateur du poste client.

Lorsque le serveur Web reçoit cette requête, il recherche la page demandée parmi toutes les pages HTML présentes sur le site concerné et la renvoie ensuite au client.

Le code HTML reçu par le poste client est alors interprété et affiché par le navigateur.

C'est ce qu'on appelle l'architecture client-serveur (je demande, on me sert) : le client est le navigateur Internet (Internet Explorer, Firefox...) et le serveur est le serveur Web sur lequel sont stockées les informations du site Internet. **[4]** (Figure II-1)

Ce type de site est très simple à réaliser et la plupart des premiers sites ont été conçus sur ce modèle.

Cependant, ce concept est limité car il manque d'interactivité.

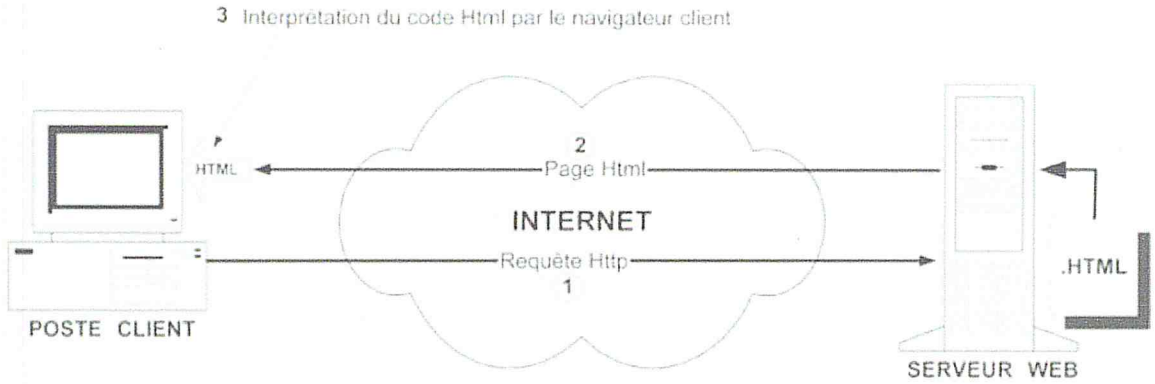


Figure II-1 L'architecture client-serveur d'un site statique

HTTP est donc un protocole assez simple par lui-même. Ce qui complique la compréhension de l'ensemble des processus mis en œuvre, c'est toute " l'intelligence " qui est ajoutée, tant du côté serveur que du côté client.

Au départ, un client envoie une requête à un serveur HTTP et celui-ci y répond. Toute la difficulté vient de deux aspects qui sont indépendants du protocole HTTP lui-même :

- Le traitement de l'information pratiqué par le serveur avant d'envoyer le résultat de la requête.
- Le traitement de l'information pratiqué par le client (navigateur) avant d'afficher le résultat de la requête.

Le protocole HTTP englobe un contenu HTML (HyperText Markup Language) qui est le langage de description d'une page Web. Ce langage s'appuie sur un ensemble de balises standards interprétées par le navigateur afin de définir le contenu et la mise en forme de la page.

Le XHTML (eXtensible HyperText Markup Language) quant à lui, est une évolution du précédent langage conforme aux contraintes du XML et impose le respect de certaines règles pour qu'une page soit bien formée (noms des balises en minuscule, attributs des balises obligatoirement encadrés par des guillemets, fermeture obligatoire de toutes les balises…).

2. Le Web orienté client :

Pour remédier au manque d'interactivité et aux problèmes d'engorgement des réseaux et de saturation des serveurs Web, les développeurs ont commencé à mettre en œuvre diverses Technologies côté client afin de délester le serveur (réduisant ainsi le trafic sur le réseau) de tâches pouvant être traitées directement par le navigateur. Ainsi chaque éditeur de navigateur Web a rapidement commencé à implémenter dans son logiciel des interpréteurs pour son propre langage. Aussi, Netscape avec JavaScript et Microsoft avec JScript permirent de pouvoir enfin exécuter des scripts côté client. (Figure II-2)

Ces nouvelles technologies client ont soulevé aussi un autre problème : celui de la sécurité des utilisateurs. Ainsi les éditeurs des navigateurs durent rapidement ajouter dans les options de leurs logiciels la possibilité de désactiver l'exécution des différentes technologies client pour répondre à la crainte des utilisateurs. Le fait même que certains navigateurs ne puissent plus exécuter les scripts client a constitué un frein important à leur usage car les développeurs devaient alors prévoir des alternatives en mode dégradé pour permettre à tous les utilisateurs d'utiliser leur application.

Par la suite, d'autres sociétés ont développé des programmes propriétaires (applets Java, ActiveX, Flash…) pouvant être intégrés dans une page Web et exécutés dans le navigateur grâce à un plug-in (extension du navigateur). Le Web disposait alors d'une pléthore de technologies client mais le manque de standardisation et l'hétérogénéité des navigateurs en rendaient leur usage très difficile.

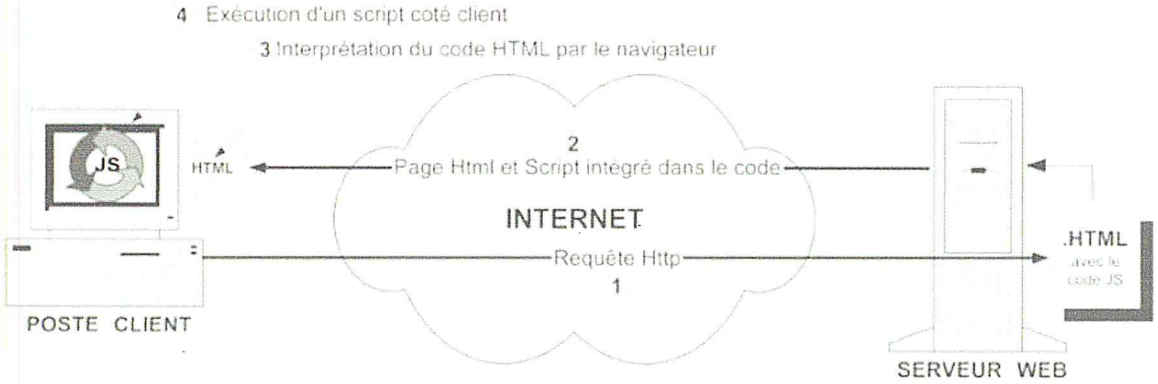


Figure II-2 Utilisation d'un script côté client avec JavaScript

En revanche, les programmes JavaScript souffrent de problèmes de compatibilité avec la configuration du client sur lequel ils s'exécutent et peuvent se comporter différemment selon le type d'ordinateur et la version du navigateur.

D'autres problèmes liés à la sécurité des données constituent aussi un frein à l'usage des technologies client. En effet, le code source des programmes étant intégré dans la page renvoyée par le serveur au client, il devient facile pour un développeur mal intentionné d'altérer le fonctionnement des scripts en consultant simplement le code source de la page HTML.

3. Le Web orienté serveur :

À partir des années 2000, l'évolution croissante des complications rencontrées avec les technologies client a entraîné une migration progressive des applications côté serveur.
Motivés par les problèmes de compatibilité et de sécurité liés aux applications côté client, bon nombre de développeurs ont adapté et installé leur programme côté serveur pour mieux satisfaire les internautes (ce qui explique en partie l'extraordinaire développement de langages serveurs comme le PHP).

En quelques années la majorité des sites ont subi des refontes structurelles pour s'adapter à une infrastructure Web exploitant principalement des applications côté serveur et cela malgré une organisation du serveur Web plus complexe liée à l'usage de ces technologies serveur.

Cependant, l'utilisation intensive des technologies serveur n'est pas non plus sans inconvénient. En effet, un usage exclusif d'applications serveur (alors que certaines gagneraient à être exécutées côté client) entraîne l'échange, entre le client et le serveur, d'un

grand nombre de requêtes qui ont vite fait d'engorger le réseau et de ralentir fortement la réactivité de l'application.

De même, à chaque requête, le serveur envoie la page HTML complète avec tout son lot d'informations redondantes, ce qui ralentit fortement l'échange d'informations entraînant des temps d'attente importants pour l'utilisateur.

Lorsque l'interactivité est placée côté serveur, le serveur Web doit disposer d'un préprocesseur PHP afin de traiter les scripts PHP intégrés dans la page avant d'envoyer celle-ci au poste client qui en a fait la demande.

Si on le compare avec un script côté client, la réaction d'un script côté serveur à un événement est beaucoup plus lente car elle nécessite l'envoi d'une requête au serveur, son exécution sur le serveur, le retour de la réponse par le réseau Internet et le chargement d'une page HTML complète dans le navigateur.

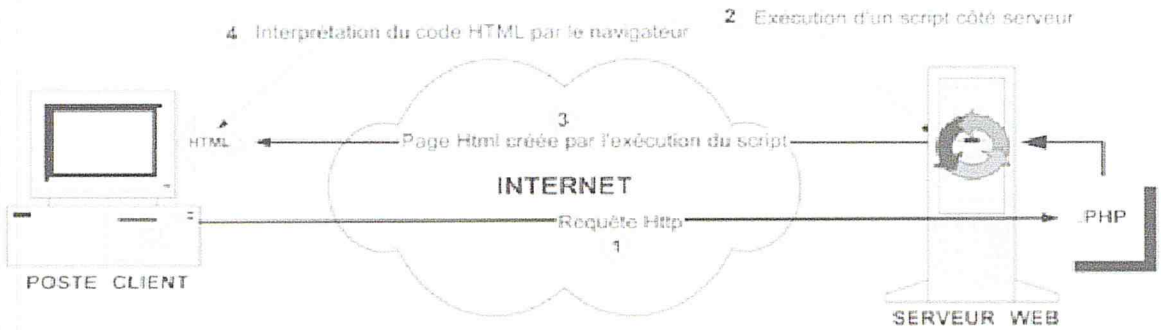


Figure II-3 L'utilisation d'un script côté serveur

En revanche, les langages côté serveur sont indépendants de la plate-forme du client ou de la version de son navigateur. En effet, l'interprétation du script est réalisée côté serveur et le code envoyé vers l'ordinateur du client est compatible avec le standard HTML et donc interprété de la même manière par tous.

Parmi les inconvénients des scripts côté serveur, il faut signaler que leur utilisation nécessite la disponibilité d'un serveur adapté. Même si les offres des hébergeurs qui proposent des serveurs intégrant des scripts dynamiques sont désormais très accessibles.

4. Les sites dynamiques :

L'exécution du script côté serveur permet de créer une page « à la volée » lors de son exécution par le préprocesseur PHP intégré au serveur. La page ainsi crée contient les mêmes informations qu'une simple page HTML. Elle peut donc être interprétée sans problème par le navigateur côté client.

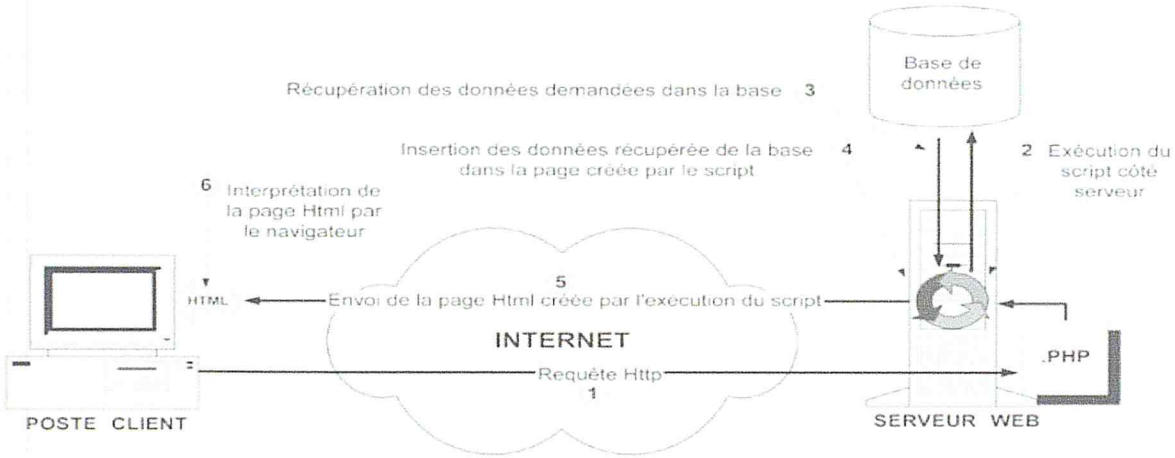


Figure II-4 Utilisation d'un script côté serveur avec accès à une base de données

Lors de la création de cette page, les scripts (PHP par exemple) intégrés au fichier dynamique sont exécutés et, si nécessaire, établissent une connexion à un serveur de données. Avec ce processus, la page dynamique devient un modèle de présentation des informations. Ce modèle pouvant être personnalisé par des contenus différents selon la requête du client.

Il n'est donc plus nécessaire, par exemple, de créer une page spécifique pour présenter chaque produit d'un catalogue : une seule page dynamique peut être utilisée.

Il suffit de lui indiquer l'identifiant du produit demandé grâce à une variable qui lui est transmise en même temps que son appel ; la page renvoyée au client contient alors toutes les informations et photos relatives au produit concerné.

L'arborescence du site est simplifiée puisque cette page dynamique remplace les nombreuses pages statiques correspondant à chaque produit.

C. Le compromis client-serveur tant attendu :

Heureusement, les navigateurs les plus courants se sont améliorés en attachant progressivement plus d'importance aux standards (même s'il reste encore des divergences entre certains d'entre eux…), diminuant ainsi les problèmes de compatibilité liés à l'usage de

technologies côté client. De même, la valeur ajoutée résultant des applications client sans cesse plus puissantes a compensé rapidement les craintes des utilisateurs à leur égard.

Le fait que bien des sites populaires exploitent désormais le JavaScript a entraîné progressivement une disparition des utilisateurs qui désactivaient les technologies client dans leur navigateur.

Ces évolutions ont eu une incidence bénéfique sur les ventilations des applications et ont permis un retour à un juste équilibre des tâches entre le client et le serveur.

Maintenant, les applications peuvent être équitablement réparties entre le client et le serveur favorisant ainsi une meilleure réactivité des systèmes même si certaines tâches, comme la conservation des données (la liaison avec les bases de données est toujours réalisée côté serveur) ou la gestion de l'authentification restent encore le privilège des technologies serveur.

D. Web 2.0 :

Le Web 2.0 n'est pas une mise à jour technique mais un changement de comportement des internautes. Comme évoqué précédemment, le Web avait pour but initial de mettre à disposition des informations. L'utilisateur était passif face aux sites Web. Puis le Web est devenu collaboratif, l'utilisateur est alors devenu créateur de contenu sans avoir à connaître les protocoles techniques sous-jacents. L'internaute ne consulte plus l'information, il publie du contenu quel que soit le média (texte, vidéo, musique). **[5]**

Les services offerts par le web 2.0 sont nombreux on trouve principalement :

- Wiki : site web a participation collective, qui permet aux visiteurs d'ajouter, de modifier et d'enregistrer les contenus, par exemple le site Wikipedia
- Réseaux sociaux : comme Facebook, LinkedIn, Instagram …
- Bureautique en ligne : comme Google docs

E. Le web 2.0 et les CMS :

La rupture technologique qui indique une évolution du web 2.0 s'est produite lors de la création des premiers CMS (Systèmes de gestion de contenu).

Un Gestionnaire de Contenu (CMS) est un logiciel qui permet de concevoir et de gérer un site Internet sans qu'il soit nécessaire de connaitre un langage informatique. Les CMS sont,

dans la plupart des cas, des logiciels libres de droit. Par définition, ils sont en libre distribution avec un code source accessible par tout le monde. Si le code est connu par tous, les failles le sont aussi. **[6]**

Exemples de CMS :

- **WordPress :** Le leader du marché. WordPress est un système de gestion de contenu (CMS) qui permet de créer et gérer facilement l'ensemble d'un site web ou simplement un blog. Gratuit et libre, WordPress est personnalisable grâce à de nombreux thèmes et plugins.
- **Joomla :** Une solution séduisante et très répandue, mais qui se révèle vite limitée à l'usage par rapport aux besoins des entreprises, la sécurité des données étant son point faible majeur.

F. Conclusion :

Dans ce chapitre nous avons vu les différentes phases d'évolution du web avec ces applications, qui peuvent présenter des failles de sécurité. Cela est d'autant plus grave que ces dernières manipulent parfois des données confidentielles (mots de passe, numéros de cartes bancaires) et qu'elles sont généralement déployées sur Internet et donc exposées au public, ce dernier inclus des personnes mal intentionnées qui on fait du web leur terrain de jeu.

Il existe une grande variété de vulnérabilités visant les applications Web. Toute fois certaines sont plus connues et plus dangereuses que d'autres, nous tâcherons à exposer les plus importantes dans le prochain chapitre.

Chapitre III

Introduction à la sécurité informatique

III. Introduction à la sécurité informatique

A. Introduction:

L'informatique, grâce au avantages qu'elle confère est devenu omniprésente dans tous les secteurs et domaines tel que la défense et l'économie, pour cela internet est un outil incontournable pour le stockage et le transfert des données et informations sensibles, ce qui fait de ces informations une proie plus abondante et plus accessible pour les hackers.

C'est là qu'entre en jeu la sécurité informatique qui a pour finalité de protéger ces informations et en même temps ses utilisateurs.

B. Généralité sur la sécurité informatique :

1. Définition de la sécurité informatique :

C'est l'ensemble des moyens mis en œuvre pour minimiser la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles, elle consiste à assurer que les ressources matérielles ou logicielles sont utilisés dans un cadre bien défini et que seules les personnes autorisées peuvent y accéder grâce à des authentifications et des contrôles. [7]

L'aspect humain est primordial dans la sécurité. De par les maladresses d'une personne comme partir et laisser sa session ouverte par exemple ou le manque de maintenance tel le fait de ne pas installer les différentes mises à jour restent une des sources à problèmes de la sécurité. [8]

2. Objectifs fondamentaux de la sécurité informatiques :

La sécurité informatique vise généralement cinq principaux objectifs :

- **La confidentialité :**

Assure que seulement les personnes autorisées aient accès aux ressources, l'effort appliqué à la protection de la confidentialité dépend de la sensibilité de l'information et de la probabilité d'être observé ou intercepté.

- **L'intégrité :**

C'est la certitude que les données n'ont pas été altérées durant le traitement ou la communication de manière fortuite ou intentionnelle.

- **La disponibilité :**

L'objectif de la disponibilité est de garantir l'accès à un service ou à des ressources à tout moment et le maintien du bon fonctionnement du système d'information.

- **La non-répudiation :**

La non-répudiation permet garantir qu'aucun des correspondants ne pourra nier son implication dans l'usage des données.

- **L'authentification :**

L'authentification consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté ou une empreinte digitale) l'accès à des ressources uniquement aux personnes autorisées.

3. Les causes de l'insécurité :

A l'heure où l'Internet devient un vecteur de l'économie mondiale (développement du commerce électronique), toute capacité de nuisance devient une arme dans la guerre économique. De nouvelles formes de délinquance se développent et parmi les causes de cette insécurité : **[9]**

- Un niveau de protection insuffisant :

  Les entreprises ne se protègent pas suffisamment pour diverses raisons :

  - manque de budget (50%) ;
  - manque de personnel (40%) ;
  - manque de temps (20%) ;
  - réticence du management (15%) ;
  - méconnaissance des enjeux (15%)

- La concentration des informations sur des serveurs connectés à l'Internet.
- La technologie et l'information technique accessibles à tous.
- La nature humaine (faiblesses, vénalité).
- Le manque de formation, une culture de la sécurité insuffisante.
- La conception d'Internet (réseau ouvert, protocoles non sécurisés).
- Les maladresses des utilisateurs.

4. Les menaces contre la sécurité informatique :

Avec l'accroissement des domaines spécifiques d'utilisation des systèmes d'information, les menaces qui pèsent sur la sécurité informatique semblent se multiplier de jour en jour. Cela ne signifie pourtant pas une fragilité grandissante des systèmes informatiques, vu que des nouveaux moyens de protection font aussi leur apparition, mais la prise de conscience de ces menaces s'avère importante.

Les spécialistes de la sécurité numérique ont mis en évidence trois types de risques, portant sur les systèmes informatiques, selon leur origine respective : menace d'origine opérationnelle, menace d'origine physique et menace d'origine humaine : **[10]**

a) Menaces d'origine opérationnelle :

Ces menaces sont liées à un état du système d'information à un moment donné, provenant, soit d'un bogue logiciel (*Buffer Overflows, format string ...etc.*), soit d'une erreur de paramétrage, filtrage des entrées (typiquement les *XSS* et *SQL injection*) ou de fonctionnement d'un logiciel.

b) Menaces d'origine physique :

Ces menaces peuvent être d'origine accidentelle, naturelle ou criminelle. Elles portent atteintes aux systèmes informatiques en les détériorant (ex : les catastrophes naturelles) ou en empêchant leurs bon fonctionnement (ex : les coupures d'électricité et les pannes de matériel).

c) Menaces d'origine humaine :

Elles sont liées aux actions directes des concepteurs et utilisateurs du système : programmation, paramétrage, configuration, etc. Elles peuvent être aussi bien intentionnelles qu'involontaires.

C. Cybercriminalité :

L'expansion du cyberespace (internet) et toutes les possibilités illégales qu'il offre comme le piratage des serveurs, le vol des cartes bancaires et sans oublier le détournement des sites web a permis la création d'une sphère de criminalité (Cybercriminalité) en relation directe avec cette technologie.

1. Définition de la cybercriminalité :

La cybercriminalité est le terme employé pour désigner l'ensemble des actes malveillants et les infractions pénales via les réseaux informatiques, notamment, sur le réseau Internet,

Ce terme désigne à la fois: **[11]**

- **Les atteintes aux biens**: fraude des cartes bancaires sur Internet sans le consentement de son titulaire; vente par petites annonces ou aux enchères d'objets volés ou contrefaits; encaissement d'un paiement sans livraison de la marchandise ou autres escroqueries en tout genre.
- **Les atteintes aux personnes**: diffusion d'images pédophiles, de méthodes pour se suicider, de recettes d'explosifs ou d'injures à caractère racial, atteinte à la vie privée.

La cybercriminalité constitue une nouvelle ère de la criminalité classique car un délit habituel comme le vol d'informations ou de numéro de carte bancaire se fait actuellement via l'outil informatique, ces actes sont commis par des cybercriminel (pirates).

2. Classifications des pirates (hackers):

En général, les hackers sont des individus qui possèdent un niveau très avancé en informatique, et ce sont également des personnes avides de connaissances, qui désirent tout comprendre sur le mécanisme de fonctionnement d'un système informatique, afin d'en localiser les failles de sécurité et les exploiter à son avantage. On peut distinguer plusieurs catégories de hackers. **[12]**

Les différents types de hackers :

Les « White Hat hackers » : «chapeaux blanc» Ce premier groupe est constitué de hackers de bonne volonté, qui ne ménagent pas leurs efforts pour chercher des innovations dans le cadre de la sécurisation d'un système informatique. Les chapeaux blancs contribuent habituellement à l'identification et à la réparation de failles de sécurité à l'intérieur d'un système, et ne peuvent être assimilés à des cybercriminels. **[12]**

Les « Black Hat hackers » : «chapeaux noir» ces hackers ne respectent pas la loi, ils pénètrent par effraction dans les systèmes dans un intérêt qui n'est pas celui des propriétaires du réseau. L'intérêt y est personnel, généralement financier, en tout cas le but est nuisible à la personne (physique ou morale) visée. **[13]**

Les « Grey Hat hackers » : Le hacker au chapeau gris est un peu un hybride du chapeau blanc et du chapeau noir. Il s'agit d'un hacker compétent, qui agit parfois avec l'esprit d'un *white hat*, parfois avec celui d'un *black hat*. Son intention n'est pas forcément mauvaise mais il commet cependant occasionnellement un délit. **[13]**

Les « *Lamers* » et « *Script Kiddies* » : Ce sont des utilisateurs peu expérimentés mais susceptibles d'altérer et de détruire tout un système par simple maladresse. Ils utilisent généralement des logiciels et des codes malveillants faits par d'autres hackeurs.

Les *Hacktivistes* : Les *hacktivistes* sont des hackers dont la seule motivation repose sur des idéologies sociales, politiques, religieuses ou autres. **[14]**

Les *Crackers* : Les crackers sont plutôt doués pour cracker des programmes, leur principal objectif consiste à violer les systèmes de protection contre la copie des logiciels payants.

Les *Phreakers* : Ce sont également des pirates, mais spécialisés en « *phreaking* » ou piratage de lignes téléphoniques.

Les *Carders* : Les *Carders* sont des hackers spécialisés en matière de « piratage de cartes à puce ».

3. Motivations et objectifs des cybercriminels :

Les cybercriminels on divers motivations et objectifs dont [15] :

Motivations :

- Cupidité, vengeance...
- Curiosité.

- Argent (espionnage industriel ou contrat avec une compagnie concurrente, détournement d'argent…).
- Recherche d'admiration (Désir de reconnaissance).
- Défis intellectuels (challenges).
- Idéologique (spécialement Pour les *hacktivistes*)

Objectifs :

- Vol et modification d'informations.
- Prise de contrôle de la machine pour utiliser ses ressources ex : installer un site Warez.
- Utiliser la machine comme rebond.
- Désinformation ex : mettre hors service les sites web.
- Bloquer l'accès à une ressource.

D. Aspect juridique :

Selon les Nations Unies, un cyber crime est «toute infraction susceptible d'être commise à l'aide d'un système ou d'un réseau informatique, dans un système ou un réseau informatique ou contre un système ou un réseau informatique. Il englobe, en principe toute infraction susceptible d'être commise dans un environnement électronique » **[16]**

Les délits de la cybercriminalité :

Les délits de la cybercriminalité se divisent en deux catégories d'infractions: **[16]**

- Les infractions où l'informatique est l'objet du délit: les atteintes à la sécurité des réseaux informatiques; à la confidentialité, l'intégrité, à l'authenticité et à l'intégrité des systèmes et données informatiques.
- Les infractions où l'informatique est le moyen du délit: la pornographie, les atteintes à la vie privée, les atteintes à la propriété intellectuelle et les infractions racistes

E. Les attaques informatiques :

Tout ordinateur connecté à un réseau informatique est potentiellement vulnérable à une attaque.

1. Définition d'une attaque :

Une « attaque » est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant des systèmes et généralement préjudiciables.

Sur internet des attaques ont lieu en permanence, à raison de plusieurs attaques par minute sur chaque machine connectée. Ces attaques sont pour la plupart lancées automatiquement à partir de machines infectées (par des virus, chevaux de Troie, vers, etc.), à l'insu de leur propriétaire. Plus rarement il s'agit de l'action de pirates informatiques.

Afin de contrer ces attaques il est indispensable de connaître les principaux types d'attaques afin de mettre en œuvre des dispositions préventives. **[17]**

2. Définition d'une faille :

Une vulnérabilité ou faille est une faiblesse dans un système informatique, permettant à un attaquant de porter atteinte à l'intégrité de ce système, c'est-à-dire à son fonctionnement normal, à la confidentialité et l'intégrité des données qu'il contient.

Ces vulnérabilités sont la conséquence de faiblesses dans la conception, la mise en œuvre ou l'utilisation d'un composant matériel ou logiciel du système, mais il s'agit généralement de l'exploitation de bugs logiciels. Ces dysfonctionnements logiciels sont en général corrigés à mesure de leurs découvertes, c'est pourquoi il est important de maintenir les logiciels à jour avec les correctifs fournis par les éditeurs de logiciels.

Il arrive que la procédure d'exploitation d'une faille d'un logiciel soit publiquement documentée et utilisable sous la forme d'un petit logiciel appelé exploit. **[18]**

3. Classes d'attaques informatiques :

Les systèmes informatiques mettent en œuvre différentes composantes, allant de l'électricité pour alimenter les machines au logiciel exécuté via le système d'exploitation et utilisant le réseau.

Les attaques peuvent intervenir à chaque maillon de cette chaîne, pour peu qu'il existe une vulnérabilité exploitable.

Il peut être utile de distinguer deux catégories d'attaques : les attaques passives et les attaques actives.

- **Attaques passives :** elles ne modifient pas le comportement du système, et peuvent ainsi passer inaperçues. Objectifs :
    - Obtention d'informations sur un système, sur un utilisateur ou un projet.
- **Attaques actives :** elles modifient le contenu des informations du système ou le comportement du système. Elles sont en général plus critique que les passives.

Objectifs :

- modification ou destruction de données ou de configurations.
- utilisation des ressources de façon clandestine sur un système.
- perturbation d'un échange par le réseau, d'un service ou d'un accès à un service.

Dans ce qui suit nous allons présenter quelques types d'attaque en bref, et nous tacherons à bien détaillé les attaques web :

a) Les attaques système :

Les virus: Un virus est un programme informatique qui, à l'insu de l'utilisateur, exerce une action nuisible à son environnement: la principale étant la modification ou la destruction des données.

Le virus est un ensemble d'instruction parasites qui s'introduisent et se cachent à l'intérieur d'autres programmes.

Un virus se développe généralement en trois phases:

- le virus s'implante dans un programme sain
- le virus se propage de façon transparente dans les autres programmes sains
- le virus déclenche son action

b) Les attaques réseaux :

Le déni de service :

Les attaques par déni de service ont pour seul but d'empêcher le bon fonctionnement d'un système et non de récupérer des informations. Elles utilisent une faiblesse de l'architecture d'un réseau.

Une technique de déni de service : **le *smurf***

c) Les attaques web :

Les vulnérabilités au sein des applications Web sont désormais le vecteur le plus important des attaques dirigées contre la sécurité des entreprises. En 2008, près de 55% des vulnérabilités dévoilées concernaient les applications Web **[19].**

Par conséquent, plusieurs communautés ont vu le jour, dans le but d'améliorer la sécurisation des applications web tel qu'OWASP.

L'Open Web Application Security Project (OWASP) est une communauté en ligne qui travaille sur la sécurité des applications Web. Sa philosophie est d'être à la fois libre et ouverte à tous d'où son nom « Open … ». Elle a pour vocation de publier des recommandations de sécurisation Web et de proposer aux internautes, administrateurs et entreprises des méthodes et outils de référence permettant de contrôler le niveau de sécurisation de ses applications Web.

Tous les trois ans environs, elle publie le classement des dix failles de sécurité les plus dangereuses dans le document « **OWASP Top 10** ». Dans sa dernière version de 2017, la liste a été réévaluée afin de prendre en compte les risques et non plus le danger représenté par ces vulnérabilités. En effet, les failles sont maintenant évaluées en fonction de la facilité à trouver la faille, son occurrence, l'attaquer les applications Web par ce biais et du préjudice que ces attaques peuvent causer.

4. Les vulnérabilités des applications web :

L'objectif principal du Top 10 de l'OWASP est d'informer les développeurs, concepteurs, architectes, managers, et les entreprises au sujet des conséquences des faiblesses les plus

Importantes inhérentes à la sécurité des applications Web.

a. Injection :

Une faille d'injection, telle l'injection SQL, OS et LDAP, se produit quand une donnée non fiable est envoyée à un interpréteur en tant qu'élément d'une commande ou d'une requête. Les données hostiles de l'attaquant peuvent duper l'interpréteur afin de l'amener à exécuter des commandes fortuites ou accéder à des données non autorisées. **[22]**

Les failles d'Injection se détectent facilement via le code, difficilement via le test.

Exemple :

L'attaque par injection SQL consiste à injecter du code SQL qui sera interprété par le moteur de base de données. Le code malicieux le plus répandu est d'ajouter une instruction pour faire en sorte que la requête sous-jacente soit toujours positive. Cela permet par exemple d'usurper une identité pour se connecter à une application Web, de rendre l'application inutilisable ou de supprimer toutes les données de la table visée, voire de la base de données complète.

La (Figure III-1) décrit le scénario d'une attaque par injection SQL :

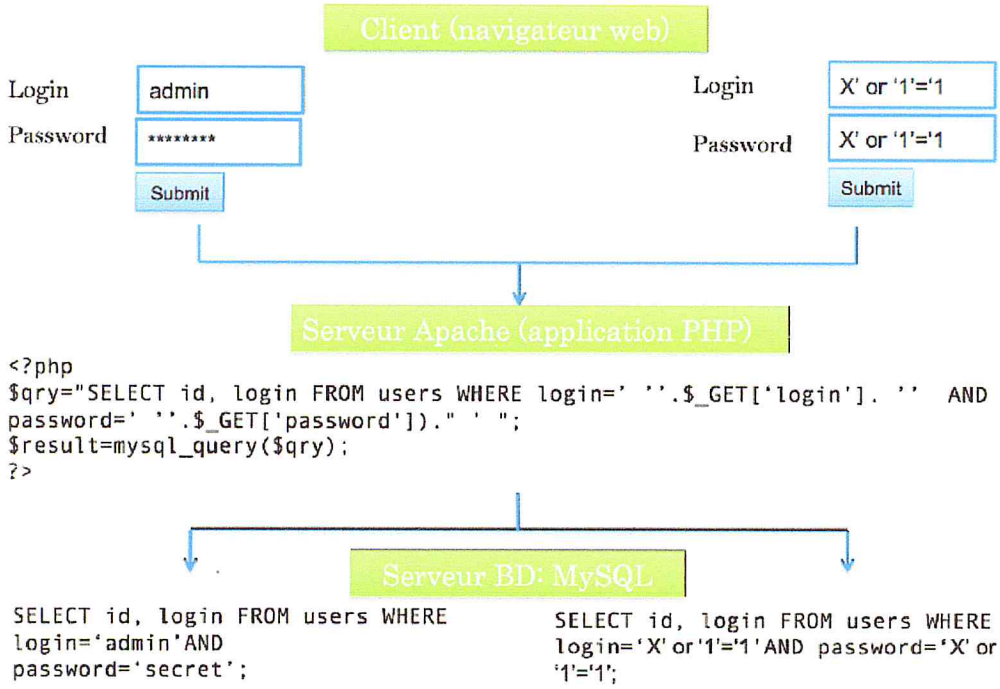


Figure III-1 Attaque par injection SQL

b. **Violation de Gestion d'Authentification et de Session :**

Les fonctions applicatives relatives à l'authentification et la gestion de session ne sont souvent pas mises en œuvre correctement, permettant aux attaquants de compromettre les mots de passe, clés, jetons de session, ou d'exploiter d'autres failles d'implémentation pour s'approprier les identités d'autres utilisateurs.

L'attaquant exploite des fuites/failles dans les fonctions de gestion de sessions et d'authentification (e.g. comptes, mots de passe, IDs de session) pour usurper l'identité des utilisateurs. **[22]**

Développer correctement un système d'authentification ou de gestion de sessions est difficile. En conséquence, ces schémas personnalisés ont souvent des failles dans des domaines tels la déconnexion, la gestion de mots de passe, l'expiration de session, la fonction "se souvenir de moi", la question secrète, la mise à jour de compte, etc...

De telles failles permettraient la compromission d'une partie voir de tous les comptes. Une fois effectuée, l'attaquant peut faire tout ce que la victime peut. Les comptes à privilèges sont souvent ciblés.

Exemple :

Le système de réservation d'une compagnie aérienne réécrit les URLs en y plaçant le jeton de session :

http://example.com/sale/saleitems;jsessionid=2P0OC2JDPXM0OQSNDLPSKHCJUN2JV

?dest=Hawaii

Un utilisateur authentifié souhaite recommander une offre à ses amis. Il leur envoie le lien par e-mail, sans savoir qu'il y inclut l'ID de session. Quand les amis cliquent sur le lien, ils récupèrent sa session, ainsi que ses données de paiement.

c. ***Cross-Site Scripting* (XSS) : [22]**

Les failles XSS se produisent chaque fois qu'une application accepte des données non fiables et les envoie à un browser web sans validation appropriée. XSS permet à des attaquants d'exécuter du script dans le navigateur de la victime afin de détourner des sessions utilisateur, défigurer des sites web, ou rediriger l'utilisateur vers des sites malveillants.
L'attaquant envoie des scripts qui exploitent l'interpréteur dans le navigateur. Toute source de donnée peut être un vecteur d'attaque y compris des sources internes telles que les données d'une base interne.

XSS est la faille la plus répandue dans les applications web. Les failles XSS ont lieu lorsqu'une application inclut des données fournies par l'utilisateur dans une page envoyée au navigateur, sans validation ou échappement correct de ce contenu.

Il en existe trois types connus:

- Stockée : Il s'agit ici d'exploiter une vulnérabilité d'un site Web de façon à y stocker de façon permanente du code exécutable malveillant (par l'intermédiaire de l'écriture de messages dans un forum par exemple). Ce code sera par la suite exécuté par tous les utilisateurs qui visiteront ensuite la partie forum du site Web.
- Réfléchie : Le principe de l'attaque reste le même que dans le cas persistant, à la différence que le code malveillant n'est pas stocké de façon permanente sur le serveur vulnérable. Il peut, par exemple, être inclus dans un paramètre de requête que l'on soumet au site vulnérable. L'attaquant, dans ce cas, doit trouver un moyen de forcer sa victime à invoquer cette URL avec ce paramètre particulier (par exemple, en lui proposant de cliquer sur un lien dans un email).
- Basée sur DOM (*Document Object Model*): C'est une attaque XSS où le code malveillant est exécuté comme résultat de la modification de l « environnement » DOM dans le navigateur de la victime via le script original interprété par son navigateur. Les attaques XSS basées sur DOM ne peuvent être stoppées par les filtres coté-serveur car c'est une modification « locale » qui ne passe pas par les serveurs ce qui fait sa dangerosité.

La détection de la plupart des failles XSS est assez simple par test ou analyse de code.

L'attaquant peut exécuter des scripts dans le navigateur de la victime pour détourner des sessions, défigurer des sites, insérer du contenu hostile, rediriger l'utilisateur vers un site malveillant, etc.

Exemple :

Les applications Web sont responsables de l'affichage des courriers électroniques : les Webmail. Pour consulter son courrier, l'utilisateur va préalablement s'authentifier et ses informations d'identification seront stockées dans des cookies. Un courrier malveillant peut intégrer du code JavaScript qui sera interprété par le navigateur. Ce code sera capable de récupérer les cookies et envoyer les informations à l'attaquant.

d. **Références directes non sécurisées à un objet :**

Une référence directe à un objet se produit quand un développeur expose une référence à un objet d'exécution interne, tel un fichier, un dossier, un enregistrement de base de données ou

une clé de base de données, sans un contrôle d'accès ou autre protection, les attaquants peuvent manipuler ces références pour accéder à des données non autorisées. [22]

Les applications incluent souvent les identifiants techniques des objets au sein des pages générées (nom, clé, etc.). La vérification des autorisations de l'utilisateur avant accès aux objets n'est pas systématique. On parle dans ce cas de références directes non sécurisées. Il est facile de détecter cette vulnérabilité en modifiant la valeur des paramètres lors de tests.

Toutes les données référencées par le paramètre vulnérable sont concernées.

Exemple :

Une application qui utilise un paramètre non validé pour construire une requête SQL d'accès aux informations d'un compte et affiche son résultat :

*String query = "SELECT * FROM accts WHERE account =?";*

PreparedStatement pstmt = connection.prepareStatement(query , ...);

pstmt.setString(1, request.getParameter("acct"));

ResultSet results = pstmt.executeQuery( );

Si l'attaquant remplace simplement la valeur du paramètre acct dans son navigateur par une autre valeur, l'application lui retournera les détails d'un compte potentiellement non autorisé :

http://example.com/app/accountInfo?acct=notmyacct

e. **Mauvaise configuration Sécurité:**

Cette faille de sécurité regroupe toutes les vulnérabilités laissées ouvertes aux différents niveaux de l'architecture de l'application Web. Pour chacun des serveurs impliqués dans l'activité de l'application, le problème concerne le système d'exploitation ainsi que les outils installés pour servir l'application.

Pour chacun de ces composants, des failles sont connues du domaine public, ce qui facilite les attaques. S'ils ne sont pas mis à jour, l'attaquant peut exploiter les failles non corrigées.

Pour de nombreux outils, des options sont installées par défaut alors qu'elles ne sont pas nécessaires au bon fonctionnement de l'application. Cette situation offre plus d'opportunités pour un attaquant.

De même de nombreuses applications sont installées avec des comptes créés avec des mots de passe par défaut. Ces comptes et mots de passe sont les cibles privilégiées des usurpations d'identité.

Exemple :

L'affichage du contenu des répertoires d'un serveur Web est possible, suite à une mauvaise configuration du serveur. L'attaquant le découvre et télécharge les classes java compilées, qu'il décompile pour obtenir le code source qu'il peut ensuite analyser pour y chercher des vulnérabilités.

f. **Exposition de données sensibles :**

Beaucoup d'applications web ne protègent pas correctement les données sensibles telles que les cartes de crédit, identifiants d'impôt et informations d'authentification. Les pirates peuvent voler ou modifier ces données faiblement protégées pour effectuer un vol d'identité, de la fraude à la carte de crédit ou autres crimes. Les données sensibles méritent une protection supplémentaire tel un chiffrement statique ou en transit, ainsi que des précautions particulières lors de l'échange avec le navigateur. **[22]**

La principale erreur est de ne pas chiffrer les données sensibles. Les autres erreurs fréquentes sont: génération de clés faibles, choix et configuration incorrects des algorithmes et protection insuffisante des mots de passe. Les faiblesses dans le navigateur sont répandues et simples à détecter mais difficiles à exploiter. En général, les faiblesses cryptographiques sont plus difficiles à identifier et exploiter de l'extérieur, en raison d'un accès limité.

Exemple :

Une application chiffre des cartes de crédit dans une base de données. La BD est configurée pour automatiquement déchiffrer les requêtes sur la colonne des cartes, permettant une faille d'injection SQL afin de récupérer tous les numéros en clair. Le système devrait avoir été configuré afin de ne permettre qu'aux applications internes de les déchiffrer, et non l'application Web publique.

g. **Manque de contrôle d'accès au niveau fonctionnel:**

Pratiquement toutes les applications web vérifient les droits d'accès au niveau fonctionnel avant de rendre cette fonctionnalité visible dans l'interface utilisateur. Cependant, les applications doivent effectuer les mêmes vérifications de contrôle d'accès sur le serveur lors de l'accès à chaque fonction. Si les demandes ne sont pas vérifiées, les attaquants seront en mesure de forger des demandes afin d'accéder à une fonctionnalité non autorisée.

Les applications ne protègent pas toujours certaines fonctionnalités. Parfois, les protections de niveau fonctionnel sont gérées par configuration et le system est mal configuré. Parfois les développeurs oublient d'intégrer les vérifications logicielles adéquates.

L'attaquant modifie simplement l'URL ou les paramètres d'une fonction privilégié. Si l'accès est autorisé cela signifie que les utilisateurs peuvent accéder à des fonctionnalités privées non protégées. [22]

Exemple:

L'attaquant force simplement la navigation d'URLs cibles. Considérons les URLs suivantes censées toutes deux exiger une authentification. Des droits Administrateur sont également requis pour accéder à la page 'admin_getappInfo'.

http://example.com/app/getappInfo

http://example.com/app/admin_getappInfo

Si l'attaquant n'est pas authentifié et que l'accès à l'une des pages est accordé, alors un accès non autorisé est permis. Si un utilisateur non administrateur authentifié est autorisé à accéder à la page 'admin_getappInfo', il existe une faille pouvant conduire l'attaquant à accéder à d'autres pages non protégées réservées aux administrateurs. De telles failles sont fréquemment introduites lorsque des liens et des boutons sont simplement masqués aux utilisateurs non autorisés et que l'application ne protège pas les pages ciblées.

h. **Falsification de requête intersites (CSRF) :**

(Cross-Site Request Forgery ou session riding ou CSRF ou XSRF) a un fonctionnement assez proche d'une attaque XSS. La principale différence est que l'utilisateur au travers de son navigateur ne sera pas la victime mais sera celui qui va effectuer une action malveillante sur l'application cible. Une attaque CSRF va exécuter du code malveillant dans une application Web au travers de la session d'un utilisateur connecté. **[24]**

Une attaque CSRF force le navigateur d'une victime authentifiée à envoyer à une application Web vulnérable une requête HTTP forgée, comprenant le cookie de session de la victime ainsi que toute autre information automatiquement inclue. En effet, cette attaque tire avantage des applications Web dont les structures des requêtes sont prédictibles. Parce que l'envoi des données de sessions telles que des cookies dans ces applications peut se faire automatiquement, les attaquants peuvent insérer des pages Web malveillantes invisibles qui génèrent des requêtes forgées qui ne sont pas distinguables des légitimes. L'attaquant forge une requête HTTP et amène une victime à la soumettre via une balise d'image (<IMG>), ou de nombreuses autres techniques. Si l'utilisateur est authentifié lors de l'exécution de cette requête, elle va s'exécuter avec succès, permettant ainsi à l'attaquant de réaliser son opération malveillante. Les attaquants peuvent par exemple faire modifier à sa victime une donnée dont elle est propriétaire, ou exécuter une action sous son identité. **[23]**

Il est à préciser que la présence d'une vulnérabilité XSS rend généralement inopérantes les défenses contre les attaques CSRF.

Exemple :

L'application permet à un utilisateur de soumettre un changement d'état qui ne contient aucun secret.

http://example.com/app/transferFunds?amount=1500&destinationAccount=4673243243

L'attaquant construit une requête qui transfèrera un montant d'argent de la victime vers son propre compte. Il imbriquera ensuite cette attaque sous une balise d'image ou un IFRAME, pour finalement les placer dans différents sites sous son contrôle.

cliquer sur le lien inclus dans l'email, puisqu'il pointe vers un site valide. De telles redirections peuvent permettre par exemple d'installer des logiciels malveillants, de capturer des informations confidentielles de l'utilisateur ou de contourner les contrôles d'accès.[22]

Exemple :

L'application Web sur le site www.example.com possède une page nommée « redirect.jsp » prenant en compte un seul paramètre nommé "url". L'attaquant fabrique une URL malveillante redirigeant les utilisateurs vers un site réalisant de l'hameçonnage (phishing).

http://www.example.com/redirect.jsp?url=evil.com

k. **Entités eXternes de l'XML (XML eXternal Entities-XXE):**

De nombreux processeurs XML anciens ou mal configurés évaluent les références d'entités externes dans les documents XML. Les entités externes peuvent être utilisées pour divulguer des fichiers internes à l'aide du gestionnaire d'URI de fichier, des partages de fichiers internes, de l'analyse de port interne, de l'exécution de code à distance et des attaques par déni de service.

l. **Désérialisation non sécurisée:**

La désérialisation non sécurisée conduit souvent à l'exécution de code à distance. Même si les failles de désérialisation n'aboutissent pas à l'exécution de code à distance, elles peuvent être utilisées pour effectuer des attaques, y compris des attaques de relecture, d'injection et d'escalade de privilèges.

m. **Journalisation et surveillance insuffisantes :**

Une journalisation et une surveillance insuffisantes, couplées à une intégration manquante ou inefficace avec la réponse aux incidents, permettent aux attaquants d'attaquer davantage les systèmes, de maintenir la persistance, de pivoter vers plus de systèmes et d'altérer, extraire ou détruire des données. La plupart des études de violation montrent que le temps de détection d'une violation varie aux alentours 200 jours, généralement détectés par des parties externes plutôt que par des processus internes ou de surveillance.

n. **Remote Command Execution RCE :**

est une attaque dans laquelle l'objectif est l'exécution de commandes arbitraires sur le système d'exploitation hôte via une application vulnérable. Les attaques par injection de commandes sont possibles lorsqu'une application transmet des données fournies par l'utilisateur dangereuses (formulaires, cookies, en-têtes HTTP, etc) à un Shell système. Dans cette attaque, les commandes du système d'exploitation fourni par le pirate sont habituellement exécutées avec les privilèges de l'application vulnérable. Les attaques par injection de commandes sont possibles en grande partie grâce à la validation d'entrée insuffisante

o. **Local File Inclusion LFI** (Inclusion de fichier local):

Elle permet à un utilisateur d'inclure des fichiers locaux (appartenant donc au serveur externe) à partir d'une URL.

Ces fichiers peuvent très bien être en dehors du répertoire racine du site web. Des fichiers sensibles comme ceux contenant des données personnelles et notamment des mots de passe peuvent donc être inclus et récupérés.

p. **Remote File Inclusion RFI :**

S'apparente à la faille LFI. Elle permet également d'inclure des fichiers appartenant à un serveur externe à partir d'une URL. Mais elle permet surtout d'inclure n'importe quel fichier sur le serveur distant.

Soyez rassurés, cette faille se fait de plus en plus rare suite aux mises à jour des serveurs web et des systèmes.

Cette faille permet souvent de placer un Shell sur le serveur afin de l'administrer à distance.

Des commandes peuvent donc être exécutées et d'une manière générale n'importe qui peut contrôler votre site web faillible via celles-ci.

Il est bien sûr possible d'inclure n'importe quel autre fichier sur un tel serveur. Et il est donc possible d'effectuer beaucoup d'actions diverses et variées en exploitant cette faille.

q. **Full Path Disclosure FPD :**

Consiste en la révélation du chemin d'exploitation complet d'un script vulnérable. Le bug de FPD est exécuté par injection caractères inattendus dans certains paramètres d'une page web.

Le script ne prévoit pas le caractère injecté et renvoie un message d'erreur qui inclut l'information de cette erreur, ainsi que la course d'actionnement du scénario cible.

Vulnérabilités FPD sont généralement observées comme des menaces à faible risque, trop souvent négligés par les webmasters en rien à craindre, ou caractéristiques du langage de script. Alors que ce dernier est vrai, le webmaster ne-devrait jamais voir la sortie des messages d'erreur.

r. **Local File Disclosure LFD :**

c'est une vulnérabilité qui donne la possibilité de lire la source des fichiers sur le serveur et ainsi lire les code source par exemple des scripts du site web ou trouver des informations importantes comme ftp, mysql logins, etc …

s. **Arbitrary File Deletion AFD :**

C'est une vulnérabilité qui donne la possibilité de suppression de fichiers arbitraires sur le serveur. Elle permet à un attaquant d'influencer l'appel aux fonctions de suppressions. En raison d'un manque de validation d'entrée, un attaquant peut fournir des séquences de traversée de répertoire suivi d'un nom de fichier arbitraire pour supprimer les fichiers spécifiques.

F. Les Scanners Web :

Tenant compte de l'augmentation des menaces ciblant les applications Web et les vulnérabilités qui peuvent porter atteinte à des propriétés essentielles telles que la confidentialité, l'intégrité ou la disponibilité des systèmes d'information. Il est nécessaire de mettre en œuvre des contre-mesures de sécurité efficaces et performantes pour faire face à ces malveillances, en développant des mécanismes de protection et de test (pare-feu, système de détection d'intrusion, scanner Web, etc.) qui soient efficaces. Dans notre cas on va se concentrer sur les scanners web pour atteindre nos objectifs.

1. Définition d'un scanner web :

Un scanner de sécurité des applications Web est un outil automatisé qui test les applications Web en communiquant avec, afin d'identifier les failles de sécurité potentielles dans l'application Web et les faiblesses architecturales les plus courantes.

Les scanners Web sont généralement de 3 types selon leurs approches:

1. Scanner Black Box
2. Scanner Grey Box
3. Scanner White Box

2. Type de scanners web:

a) *Scanner Black Box:*

- Le scanner de l'application teste l'état de la sécurité d'un point de vue extérieur
- Explorer les perspectives de vulnérabilité d'un point de vue extérieur
- Déduit que certaines vulnérabilités existent en envoyant les entrées à l'application et en faisant l'analyse des résultats
- Pas d'accès au code source de l'application

Des exemples de scanner Black Box :

- HP (SPI Dynamics) WebInspect & DevInspect
- IBM Rational (Watchfire) AppScan
- Cenzic Hailstorm
- NT Objectives NTO Spider
- Acunetix Web Vulnerability Scanner

b) *Scanner Grey Box:*

- C'est une combinaison entre White Box et Black Box.
- La structure interne est partiellement connue (tel que les algorithmes).
- Les tests sont faits de la même manière que la méthode Black Box (niveau utilisateur sans privilèges).

c) *Scanner White Box:*

- Audit du code source de l'application
- En règle générale complétée par un examen de la conception architecturale pour l'identification des problèmes non-codes
- Accès au code source de l'application et aux documents de l'architecture.

Des exemples de scanner White Box :

- Fortify Source Code Analyzer

- Ounce Labs
- Coverity Prevent SQS

3. Utilisation des scanners:

Le scanner de vulnérabilités est utilisé par des spécialistes de sécurité au cours d'un audit, pour identifier les vulnérabilités les plus fréquentes et ainsi faciliter le test de pénétration manuelle qui vient après.

Il se peut qu'il soit utilisé par des pirates pour des fins totalement malveillantes. Mais quel que soit le domaine d'utilisation du scanner de vulnérabilité il reste un outil de sécurité non pas un outil de piratage.

4. Fonctionnement d'un scanner : [23]

Les attaques les plus courantes concernant les serveurs et applications Web sont les attaques d'injection, qui proviennent de l'absence de test de conformité des paramètres d'URL ou des données fournies dans les champs des formulaires.

Pour vérifier si ces attaques d'injection de code sont possibles, les outils de détection de vulnérabilités envoient des requêtes particulières et analysent les réponses retournées par le serveur. Un serveur peut répondre avec une page de rejet ou une page d'exécution.

La page de rejet correspond à la détection par le serveur de valeurs d'entrée malformées ou invalides. Une page d'exécution est renvoyée par le serveur suite à l'activation réussie de la requête.

Elle peut correspondre soit au scénario "normal", dans le cas d'une utilisation légitime du site, soit à un détournement de son exécution via l'exploitation réussie d'une injection de code (via des entrées non conformes).

Pour identifier les vulnérabilités d'un site Web, les scanners soumettent au site des requêtes contenant des données non conformes correspondant à des attaques potentielles.

Les réponses sont alors analysées afin d'identifier les pages d'exécution. Si une page d'exécution est identifiée, la page correspondante est considérée vulnérable.

C'est ainsi que les outils détectent l'absence de test de conformité des paramètres.

On peut distinguer deux principales classes d'approches adoptées par les scanners de vulnérabilités :

- par reconnaissance de message d'erreurs dans les requêtes renvoyées par le serveur
- par l'étude de similarité des pages renvoyées.

a) Approche par reconnaissance de messages d'erreurs :

Pour identifier les injections possibles, cette approche consiste à envoyer des requêtes d'un format particulier et chercher des motifs spécifiques dans les réponses tels que les messages d'erreurs.

L'idée fondamentale est que la présence d'un message d'erreur dans une page HTML de réponse signifie que la requête correspondante n'a pas été vérifiée par l'application Web avant d'être transmise aux fonctions.

Par conséquent, le fait que cette requête a été envoyée inchangée révèle la présence d'une vulnérabilité.

Les scanners tels que W3af (moduleSQLI), Wapiti et Secubat adoptent une telle approche.

b) Approche par étude de similarité des réponses :

Le principe de cette approche consiste à envoyer différentes requêtes spécifiques aux types de vulnérabilités recherchées et à étudier la similitude des réponses renvoyées par l'application en utilisant une distance textuelle. En fonction des résultats obtenus et de critères bien définis, on conclut sur l'existence ou non d'une vulnérabilité.

Le scanner développé par Google, Skipfish adopte cette approche.

Parmi les inconvénients de cette approche :

- La distance considérée pour l'étude de similarité considère la fréquence des mots sans tenir compte de l'ordre des mots dans un texte.
  Ignorer l'ordre des mots peut amener à ignorer la sémantique d'une page et à nouveau peut amener à mal juger si deux pages sont identiques ou non. Par exemple, les pages suivantes partagent le même vocabulaire, mais elles correspondent à une authentification réussie et échouée respectivement :
  - *Your are authenticated, you have not entered a wrong login.*

- Your are not authenticated, you have entered a wrong login.

5. Etude comparative entre les différents scanners : [25]

Dans cette partie du chapitre nous allons élaborer une étude comparative entre plusieurs scanners de vulnérabilités web.

Le (Tableau III-1) représente une comparaison entre les scanners selon le nombre des types de vulnérabilités traités

Le (Tableau III-2) représente une comparaison entre les scanners selon leurs échelle d'utilisation, couvertures de paramètres d'entrées, méthodes d'initialisation du scan et leurs sorties.

Vulnérabilités / ScannerID	Count	SQL	BSQL	BSQL	RXSS	PXSS	DXSS	Redirect	Bck	Auth	CRLF	LDAP	Xpath	MX	Session Test	SSI	SSI	RFI/LFI	Cmd	Buffer	CSRF	ADoS
IBM Rational	25	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui		Oui		Oui	Oui	Oui	Oui	
WebInspect	21	Oui		Oui	Oui	Oui		Oui	Oui	Oui	Oui	Oui	Oui				Oui	Oui	Oui	Oui	Oui	Oui
W3AF	21	Oui		Oui	Oui	Oui	Oui	Oui	Oui		Oui	Oui	Oui	Oui	Oui		Oui	Oui	Oui	Oui	Oui	Oui
Cenzic Hailstorm Professional	20	Oui		Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui			Oui		Oui	Oui	Oui		Oui	
unetix WVS (Commercial Editi	18	Oui		Oui	Oui	Oui	Oui	Oui	Oui		Oui	Oui	Oui	Oui				Oui	Oui		Oui	
Nessus	18	Oui		Oui	Oui	Oui		Oui	Oui	Oui	Oui				Oui		Oui	Oui	Oui	Oui	Oui	Oui
NTOSpider	14	Oui		Oui	Oui	Oui	Oui			Oui	Oui				Oui			Oui	Oui		Oui	
SandcatCS	14	Oui			Oui				Oui		Oui	Oui	Oui	Oui	Oui			Oui	Oui	Oui		
ShipFish	14	Oui		Oui	Oui	Oui		Oui	Oui				Oui					Oui	Oui		Oui	
Sandcat Pro	14	Oui		Oui	Oui				Oui		Oui	Oui	Oui	Oui	Oui			Oui	Oui	Oui		
Sandcat Free Edition	13	Oui		Oui	Oui				Oui		Oui	Oui	Oui	Oui	Oui			Oui	Oui	Oui		
arachni	13	Oui		Oui	Oui		Oui	Oui	Oui		Oui	Oui	Oui					Oui	Oui		Oui	
Wapiti	12	Oui		Oui	Oui	Oui			Oui		Oui	Oui	Oui					Oui	Oui			Oui
JSkiOui (Commercial Edition)	12	Oui		Oui	Oui			Oui	Oui		Oui	Oui	Oui					Oui	Oui			
Iverspanner (commercial	11	Oui		Oui	Oui	Oui		Oui	Oui		Oui							Oui	Oui			
Burp Suite Professional	11	Oui		Oui	Oui	Oui		Oui			Oui	Oui			Oui			Oui	Oui			
Vega	11	Oui		Oui	Oui			Oui			Oui							Oui	Oui			
Grendel Scan	8	Oui			Oui				Oui	Oui	Oui				Oui			Oui			Oui	
ZAP	7	Oui		Oui	Oui				Oui		Oui						Oui					
PowerFuzzer	7	Oui			Oui	Oui					Oui	Oui	Oui						Oui			
ParosPro	7	Oui		Oui	Oui			Oui	Oui		Oui						Oui					
Andiparos	7	Oui		Oui	Oui			Oui	Oui		Oui						Oui					
Uber Web SecuriOui Scanner	6	Oui			Oui							Oui						Oui				

Tableau III-1 Tableau représentatif du nombre de vulnérabilités traitées par les scanners les plus connu

Scanner	Échelle d'utilisation					couverture de paramètre d'entrée				Méthodes d'imitation de scan			Sortie	
	GUI	Config	Usage	Stabilité	Performance	GET	Post	Cookie	http Header	Site map	Crawl manuel	L'analyse de fichier	Raport	Journalisation
Acunetix WVS (Edition commerciale)	Oui	Très Simple	Très Simple	Très Stable	Rapide	Oui	Oui	Oui	Oui	Oui	Oui		Oui	Oui
Acunetix WVS Free Edition	Oui	Très Simple	Très Simple	Stable	Rapide	Oui	Oui	Oui	Oui	Oui	Oui		Oui	Oui
aidSQL		Simple	Complex	Unstable	Rapide	Oui				Oui				Oui
Andiparos	Oui	Très Simple	Très Simple	Stable	Rapide	Oui	Oui			Oui	Oui		Oui	Oui
arachni	Oui	Simple	Complex	Stable	Rapide	Oui	Oui	Oui	Oui	Oui	Oui		Oui	
Burp SuiteProfessional	Oui	Très Simple	Simple	Très Stable	Rapide	Oui	Oui	Oui	Oui	Oui	Oui		Oui	Oui
Cenzic Hailstorm Professional	Oui	Très Simple	Très Simple	Stable	Rapide	Oui	Oui	Oui	Oui	Oui	Oui		Oui	
crawlfish	Oui	Simple	Simple	Unstable	Rapide	Oui				Oui				
Davon Small SQLi Scanner (DSSS)		Très Simple	Très Simple	Stable	Très Rapide	Oui				Oui				Oui
Ganja		Très Simple	Simple	Très Stable	Rapide	Oui	Oui			Oui			Oui	
Grabber		Complex	Complex	Stable	Lent	Oui	Oui			Oui		Oui		
Grendel Scan	Oui	Simple	Simple	Stable	Lent	Oui	Oui			Oui	Oui		Oui	Oui
IBM RationalAppScan	Oui	Simple	Simple	Stable	Rapide	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
iScan	Oui	Très Simple	Simple	Unstable	Rapide	Oui				Oui			Oui	
JSky (Edition commerciale)	Oui	Très Simple	Simple	Stable	Rapide	Oui	Oui			Oui			Oui	
JSKY FreeEdition	Oui	Simple	Simple	Stable	Rapide	Oui				Oui				
LoverBoy	Oui	Très Simple	Complex	Fragile	Lent	Oui				Oui				Oui
Mini MySqlatOr	Oui	Simple	Simple	Très Stable	Très Rapide	Oui	Oui			Oui				
Nessus	Oui	Complex	Complex	Stable	Lent	Oui	Oui	Oui	Oui	Oui			Oui	
Netsparker (Edition commerciale)	Oui	Très Simple	Très Simple	Stable	Rapide	Oui	Oui			Oui	Oui		Oui	Oui

Tableau III-2 Tableau représentatif de quelques caractéristiques et de l'ergonomie des scanners d'applications Web

G. Conclusion :

Les applications web sont la cible privilégiée d'attaques par Internet, il est nécessaire de mettre en œuvre des contre-mesures de sécurité efficaces et performantes.

Nous avons présenté dans ce chapitre les différentes classes de vulnérabilités pouvant être exploitées par les attaquants de différentes manières. Aujourd'hui nul ne peut affirmer qu'il existe des mécanismes de sécurité informatique infaillibles, d'une part, parce que les vulnérabilités ne sont pas toutes connues, d'autre part, parce-que les systèmes et technologies évoluent rapidement avec à chaque fois un lot de nouvelles vulnérabilités.

Des précautions existent pour se prémunir d'un ensemble d'attaque. Il s'agit par exemple, pour les webmasters, des outils développés spécialement qui ont pour objectif l'audit de sécurité des applications web tel que les scanners web précédemment vus, qui sont des outils favorables pour l'automatisation de l'audit des applications web.

Dans le prochain chapitre on aura l'occasion de voir l'étude conceptuelle de notre scanner, ainsi que l'approche adopté.

Chapitre IV

Etude Conceptuelle

IV. Etude Conceptuelle

A. Description de la solution à concevoir :

Notre but est de concevoir un scanner qui traitera les vulnérabilités les plus répondues.

Les vulnérabilités qu'on va incorporer dans notre solution sont :

- Injection SQL (SQLI)
- Cross site Scripting (XSS)
- Remote Command Execution (REC)
- Remote File Inclusion (RFI)
- Local File Inclusion (LFI)
- Full Path Disclosure (FPD)
- Ip_Falsification
- File_Deletion
- File_Disclosure

B. Conception de la solution :

1. Vue globale :

Dans cette partie du chapitre, nous allons détailler le fonctionnement de notre scanner, l'architecture des modules et leurs interactions. (Figure IV-1)

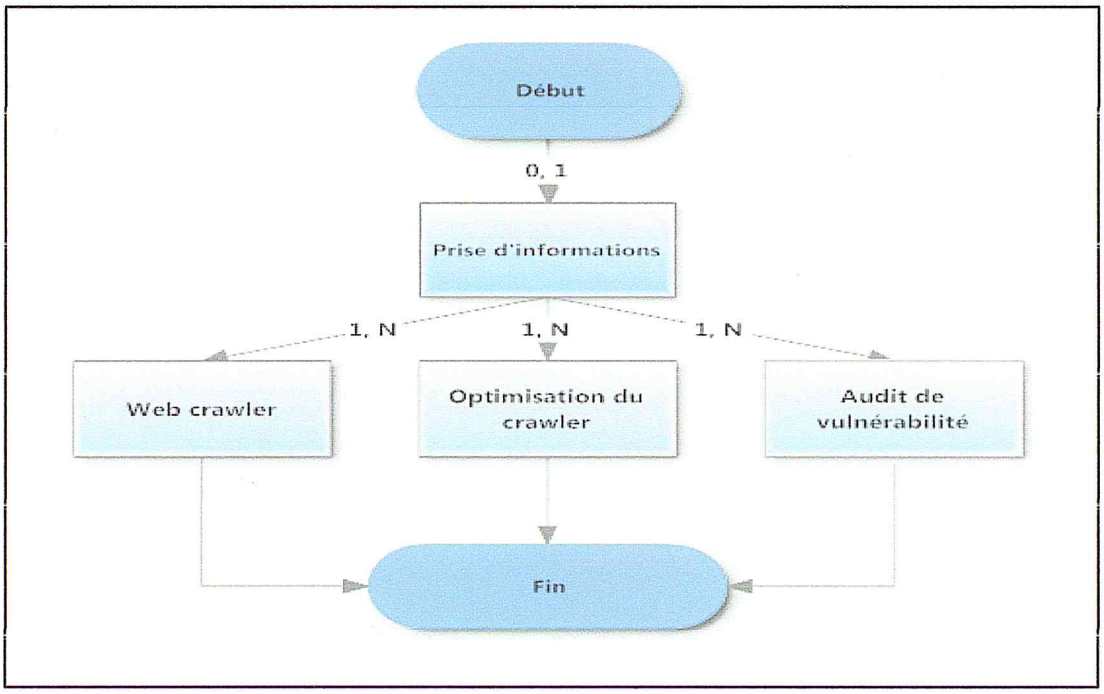


Figure IV-1 Vue globale du système

Le scan débute à l'insertion de l'url, dès lors le module qui s'occupe de la prise d'informations analyse le *host* et collecte les données nécessaires pour orienter le scan du site lors des prochaines étapes.

Suite à la collecte d'informations une multitude d'instances des modules suivants peuvent être lancé, le module « web crawler » a pour but de parcourir le site et le cartographier, sachant que ce module collabore étroitement avec le module « Optimisation du crawler » qui lui a pour but de trouver une logique entre les liens du site pour ainsi éviter de parcourir tout le site inutilement.

En parallèle le module « Audit de vulnérabilité » donne tout son sens à notre application en essayant les différentes vulnérabilités à travers divers déclencheurs d'erreurs, le « *Multi Processing* » et la synchronisation des traitements se fait grâce à une base de données relationnelle.

Avant le développement des différents modules, nous avons commencé par créer trois classes nommés « http », « Mysql » et « Tools » ces classes seront utilisées dans la classe de chaque module cité précédemment.

Chaque module se compose de :

1. Une classe regroupant l'ensemble des fonctions nécessaires qui lui sont propres.
2. Une partie procédurale qui utilisera les fonctions et méthodes de la classe qui l'accompagne pour accomplir ses tâches.

Ce qui nous donne une architecture logicielle qui nous permet une meilleure flexibilité d'exécution.

2. La base de données :

Une base de données est un ensemble structuré de données et d'informations non redondantes régit par un modèle de données.

Nous avons fait l'usage de cette base de données afin de stocker les informations et synchroniser les quatre modules pour une exécution en parallèle.

Notre base de données est composée de 9 tables, le schéma complet de notre base de données se présente comme suit :

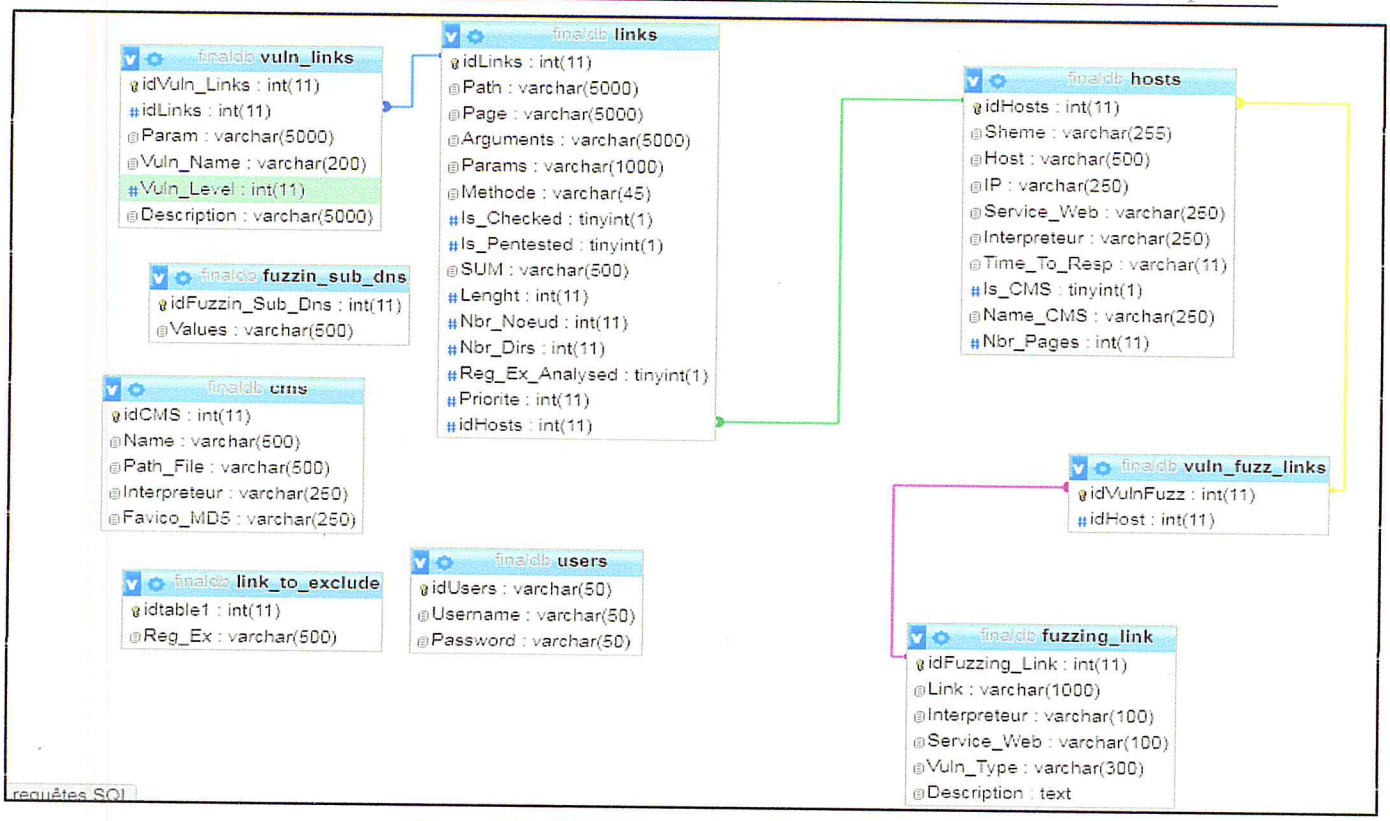


Figure IV-2 Schéma relationnel de la BDD

En premier temps on a fait un schéma relationnel normalisé qui comptait 33 tables, ce qui nous coutait très chère en ressources, car pour accéder à une donnée on devait faire des jointures entre plusieurs tables à chaque appel.

Pour y remédier on a réduit le nombre de tables nécessaires à 9 au détriment de la normalisation, ce qui nous permis de gagner énormément en terme en temps et en espace sans perte d'informations et donc gagné en efficacité.

Le schéma de la BDD normalisé se trouve dans la partie **Annexe 1.**

Explication du schéma relationnel de la BDD :

- Ce schéma nous permet de garder une traçabilité de tous les liens valides rencontrés durant l'exécution qu'ils soient vulnérables (« vuln_links ») ou non (« hosts », « links »), la table « fuzzing_link » contient les liens entrés nécessaires pour faire l'audit enfin la table « vuln_fuzz_links » nous permet de garder trace des entrés qui ont portés leurs fruits.
- La table « Users » contient tous les noms d'utilisateurs et le hash de leurs mots de passe.

- Dans la table « fuzzin_sub_dns » sont stockés les sub-dns les plus courants, pour les utiliser au cours de l'extraction des sous-domaines.
- La table « CMS » contient les données spécifiques des CMS qui nous permettent leurs identification si utilisés.
- La table « link_to_exclude » contient les expressions régulières des liens à exclure.

Après avoir donné une vue globale sur notre conception nous allons détailler le principe de fonctionnement de chacun modules cités.

3. Le module « Prise d'informations » :

Toute attaque nécessite une phase de préparation correspondante à la collecte des informations. Cette phase, aussi appelée prise d'empreinte (*Finger-printing* en anglais), rassemble l'ensemble des techniques permettant à l'attaquant de collecter le maximum d'informations sur sa cible, de la connaître, afin de mener l'attaque de façon efficace, et d'attaquer les points sensibles. Pour cela, certaines sources d'informations sont très faciles d'accès pour tout un chacun. Cela dit d'autres le sont moins.

Ce module se base sur une multitude de techniques qui serait bon de définir :

- **Google est notre ami :**

  -Le premier outil indispensable à toute collecte d'informations, est bien sûr le moteur de recherche Google.

  -Google est en possession d'une base de données immense contenant des informations sur tous les sujets, pratiquement toutes les personnes.

  -Une simple recherche peut mener très loin.

  -Dans notre cas nous avons employé le « *google dork* » pour avoir une estimation du nombre de pages que compte le site.

- **Interroger les serveurs DNS:**

  -Côté technique, de nombreux outils peuvent nous renseigner sur l'architecture d'un réseau cible.

-DIG par exemple est un programme informatique de débogage de serveurs DNS. Il signific *Domain Information Groper*, littéralement Chercheur d'Information sur les Domaines.

-Utilisable en ligne de commande, il permet aussi d'interroger le serveur DNS de son choix.

-La commande DIG peut aussi être utilisée dans le cadre d'un transfert de zone DNS

- **Transfert de zone DNS :**

-C'est un type de transaction DNS.

-C'est l'un des nombreux mécanismes disponibles pour répliquer les bases de données distribuées contenant les données DNS au travers d'un ensemble de serveurs DNS.

- **Le Fuzzing** (Ou test à données aléatoires):

-C'est une technique pour tester des logiciels.

-L'idée est d'injecter des données aléatoires dans les entrées d'un programme.

-Si le programme échoue (par exemple en plantant ou en générant une erreur), alors il y a des défauts à corriger.

-Dans le premier module, nous avons utilisé le principe de cette attaque pour injecter des « SUB-DNS » selon un dictionnaire.

a) Fonctionnement du module « prise d'informations »

a. **Traduire l'HOTE en adresse IP :**

Ceci représente la première tâche du module « prise d'informations » qui est de récupéré l'adresse IP du *host*, mais aussi s'assurer que le *host* existe bien. Le test de traduction de l'adresse du *host* en une adresse IP se fait trois fois pour pallier aux éventuels problèmes de connections.

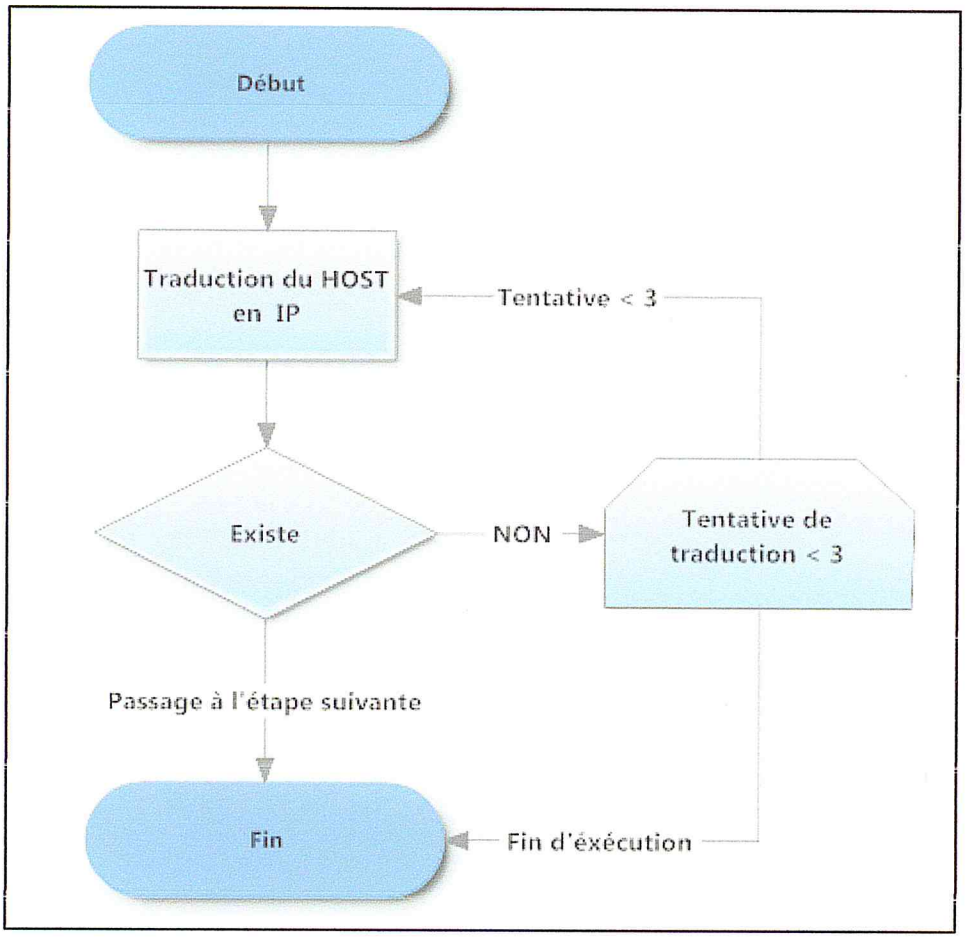


Figure IV-3 Organigramme de résolution d'adresse IP

b. **Déterminer le type de l'adresse IP (Publique/Privée) :**

Pour déterminer le type de l'adresse IP on se sert d'une expression régulière a la recherche des classes IP A/B/C/D/E. Récolter cette information nous permets d'écarter certaines vérifications ou procédures à venir.

c. **Estimer le nombre de pages :**

L'estimation de page ce fait via un (GHDB) « google dork », cette information nous aidera à estimer le temps moyen pour traiter toute l'application. Naturellement ce n'est qu'une estimation qui peux être faussé par plusieurs facteurs comme des troubles de la

connectivité ou encore une saturation de la bande passante Client ou Server, un ban depuis les serveurs de Google ou une multitude d'instances pour les modules.

L'estimation de page ne peut se faire que dans le cas où l'IP du *host* est une adresse IP publique.

La formule qui nous permet de calculer le temps de traitement de l'application est la suivante :

(Temps de téléchargement d'une page * Nombre de pages).

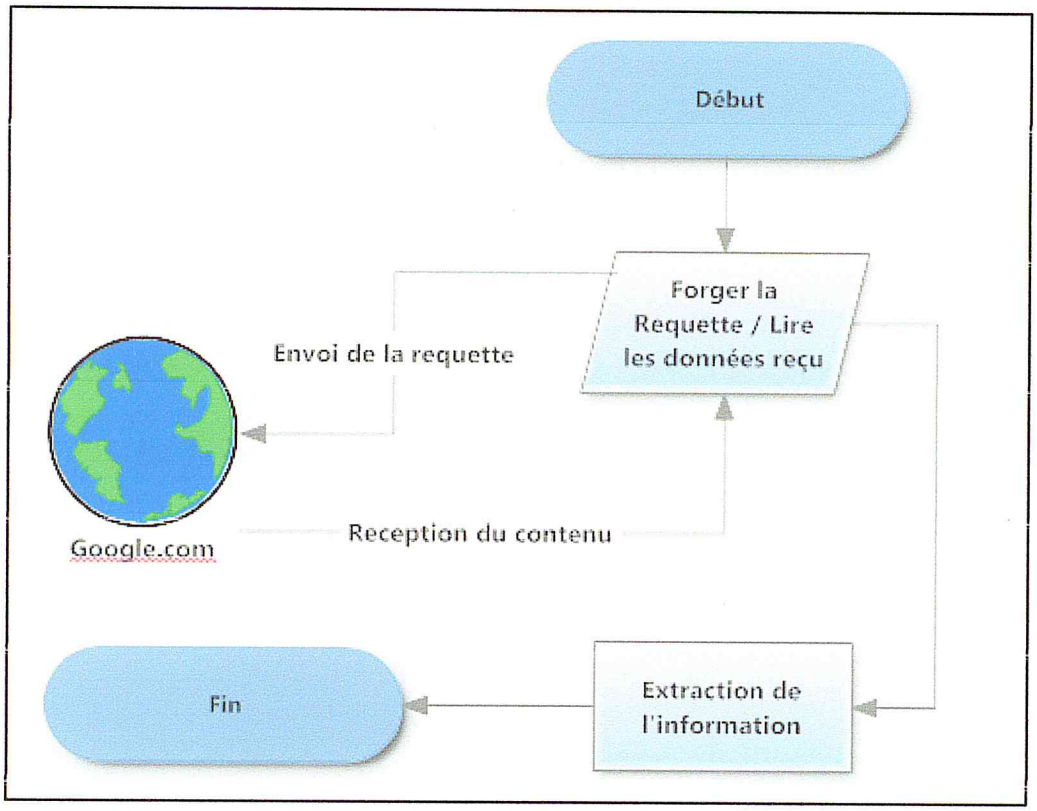


Figure IV-4 Organigramme d'estimation de nombre de page

d. **Déterminer si l'application est un CMS :**

Détecter si notre cible est bien un CMS est l' une des informations qui orientera notre scan de vulnérabilité, les CMS nativement ont tendance à être non sensible au scan de vulnérabilité en conséquence de la communauté très active qui en est responsable, toute fois les vulnérabilités découvertes sur les modules et composants des CMS suite au audit en white box ont tendance à porter leurs fruits, dans ce cas-là on aura à orienter notre scan

de vulnérabilité vers un *fuzzing* des failles connues ou encore vers les disfonctionnements dans la configuration du CMS.

La détection du CMS ce fait grâce à plusieurs méthodes (Figure IV-5) :

- La lecture du contenu de la balise *meta-generator*
- Vérification de la signature du *favicon*
- *Finger-print* par un dossier ou un fichier

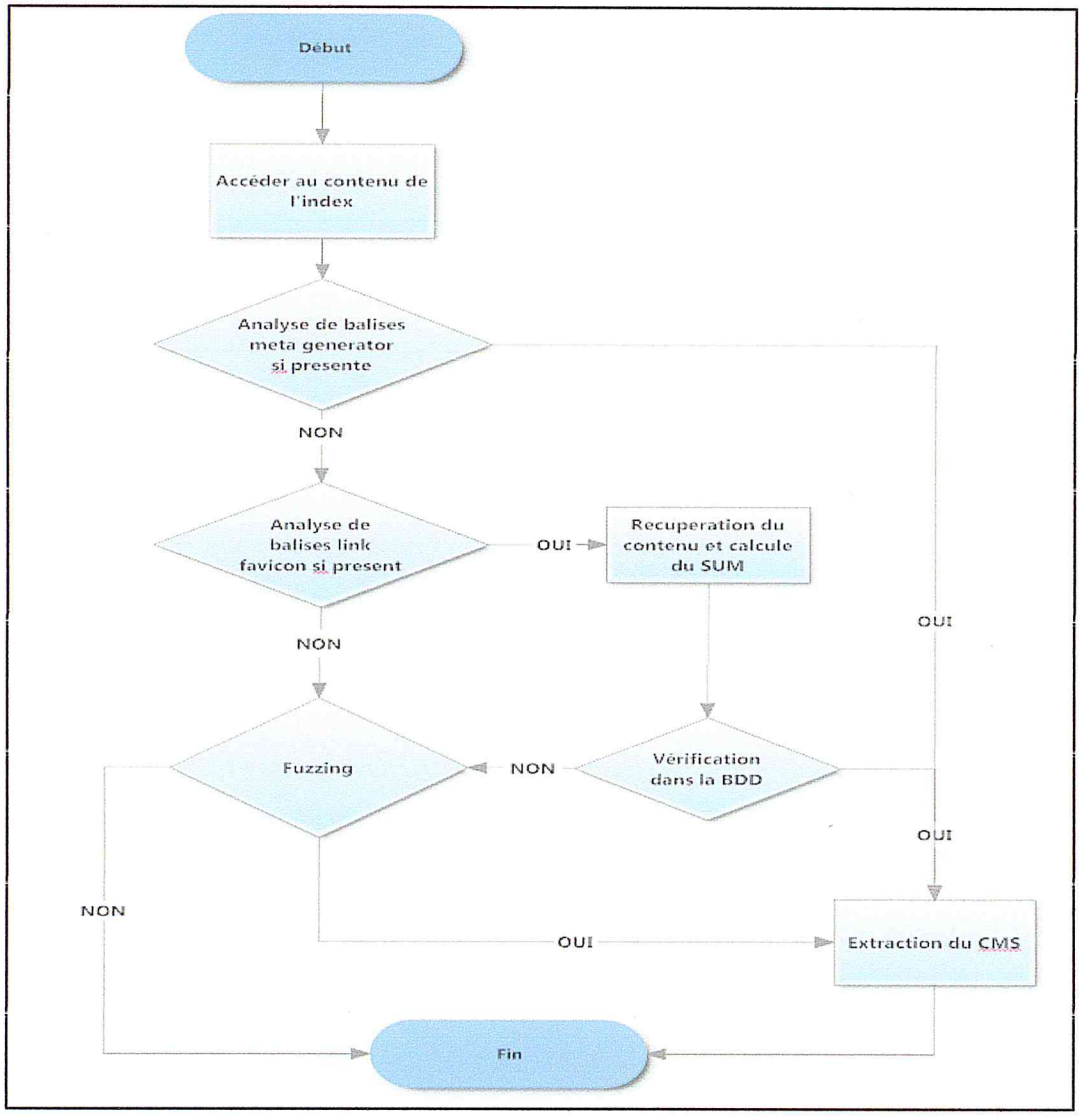


Figure IV-5 Organigramme d'identification du CMS

e. **Analyse de la configuration du serveur web :**

Une analyse de la configuration du serveur web qui a pour but d'identifier le service web qui s'occupe de traiter les requêtes reçus tel que (apache, nginx etc …) et l'interpréteur de code tel que (PHP, Python, JAVA …), ces deux informations seront des informations

complémentaires importantes dans l'analyse de vulnérabilité effectué sur l'application Web.

Ces informations sont présentes dans l'entête HTTP des réponses retournés par les serveurs web. (Figure IV-6).

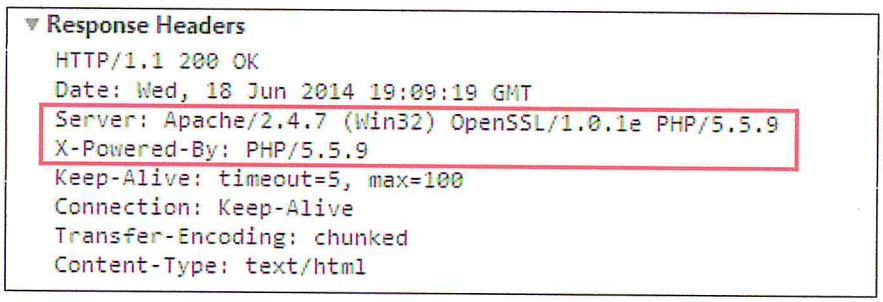


Figure IV-6 Header de Réponse HTTP

Pour notre cas, après avoir envoyé une requête nous avons extrait la header de la réponse puis nous avons effectué une analyse qui nous a permis de récupérer ces informations (Figure IV-7)

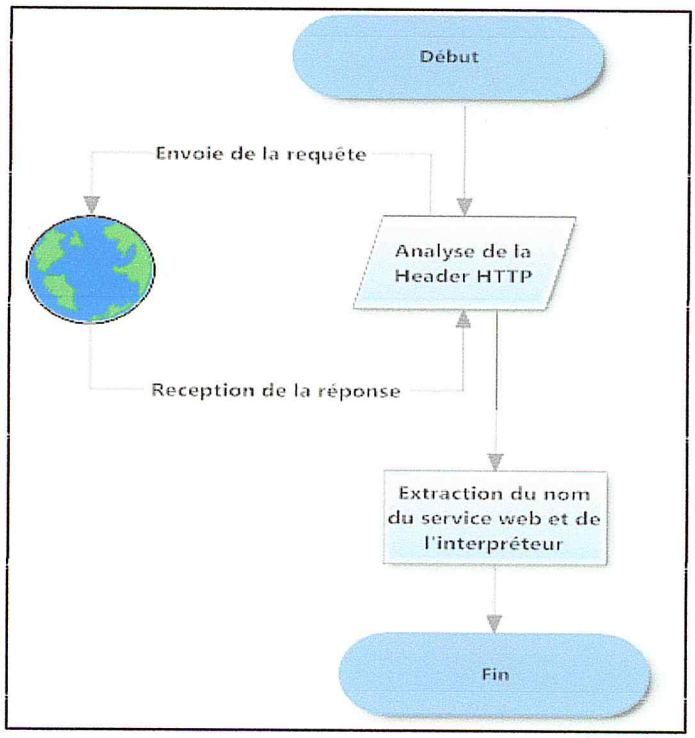


Figure IV-7 Organigramme d'identification de service web

f. Récupération des sous domaine (*sub-domain*):

Les applications web peuvent être de différentes tailles ce qui les amènes à utiliser ou à subdiviser leur applications sur différents sous domaines ou par simple aspect organisationnelle tel que : « admin.host.com ou encore forum.host.com etc… ». De ce fait, il serait intéressant pour un audit complet de l'application ou du site web de ne pas se limiter au host principale www.host.com mais de découvrir les différentes branches hébergées sur la même machine.

On a eu recours à deux méthodes pour faire la découverte des sous domaines :

- Le transfert de zone DNS en faisant appel à la commande système DIG
- Le fuzzing basé sur un dictionnaire déjà stocké dans la base de donnée, ce dernier regroupe les noms de sous domaine les plus répondus.

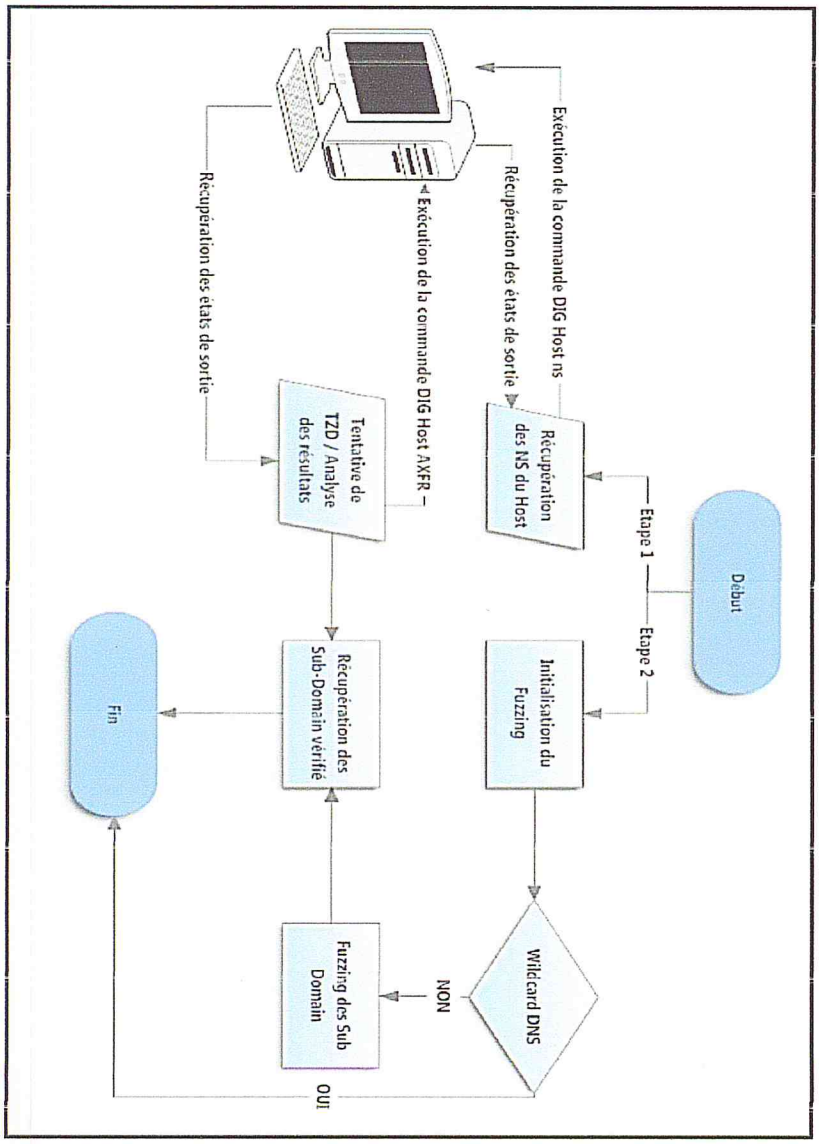


Figure IV-8 Organigramme de récupération des sous domaine

La Figure IV-9 résume le fonctionnement global du module « prise d'informations » :

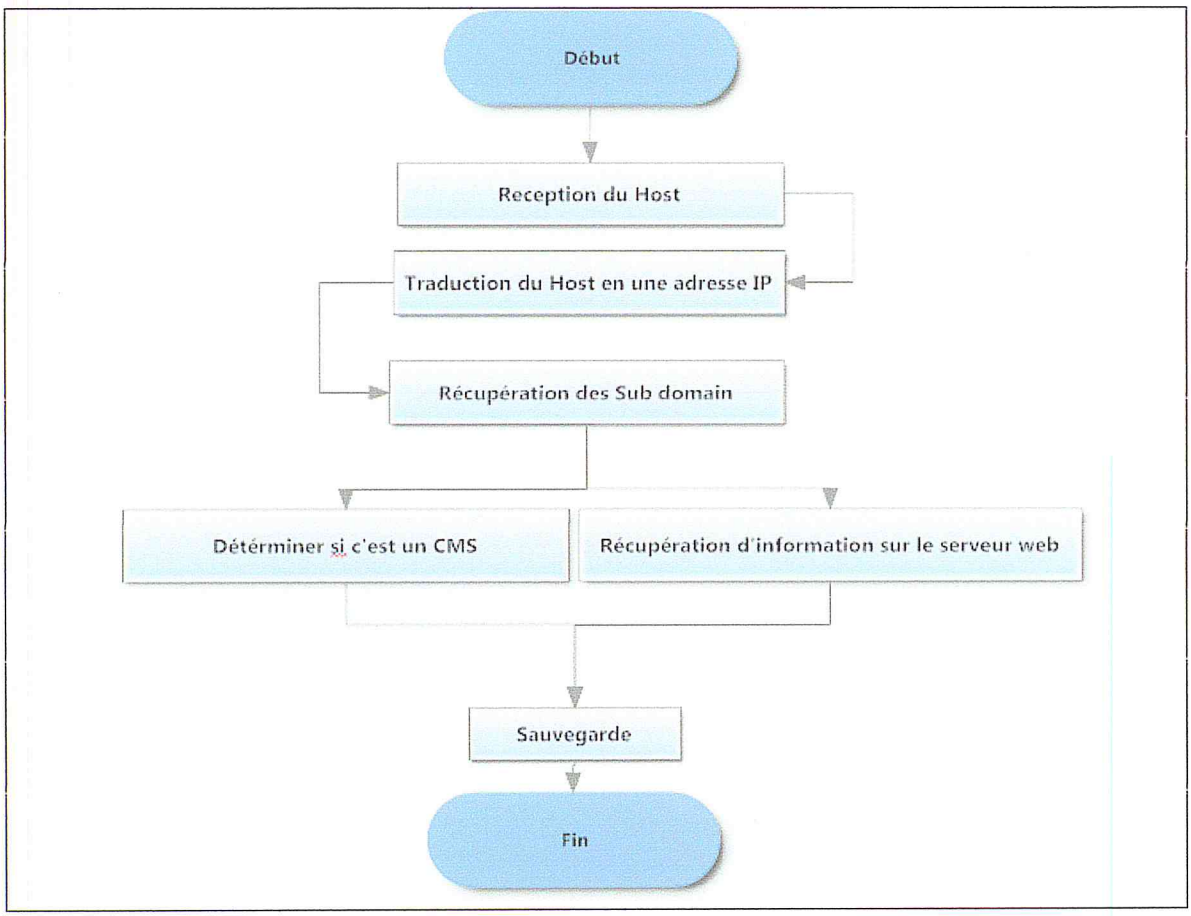


Figure IV-9 Organigramme de fonctionnement global du module 1(prise d'informations)

Après la récupération des sous domaine ainsi que les informations concernant le CMS (si toute fois l'application est un CMS) et la récupération des informations sur le serveur web (service, interpréteur). Ce module se concrétise par la sauvegarde sur la base de données dans la table « HOSTS »

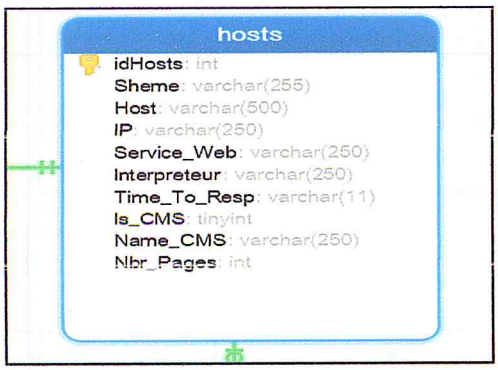


Figure IV-10 Table HOSTS

NB : un TRIGGER s'occupe de pré-remplir la table « links » avec les liens racines qui correspondent à chaque « host ».

4. Le module « Crawler » :

Une application web est un ensemble de pages liées de différentes manières, le crawler a pour but de parcourir chaque page en indexant et en collectant les navigations possibles.

Notre module est composé de plusieurs fonctions interagissant avec la base de données. (Figure IV-11) :

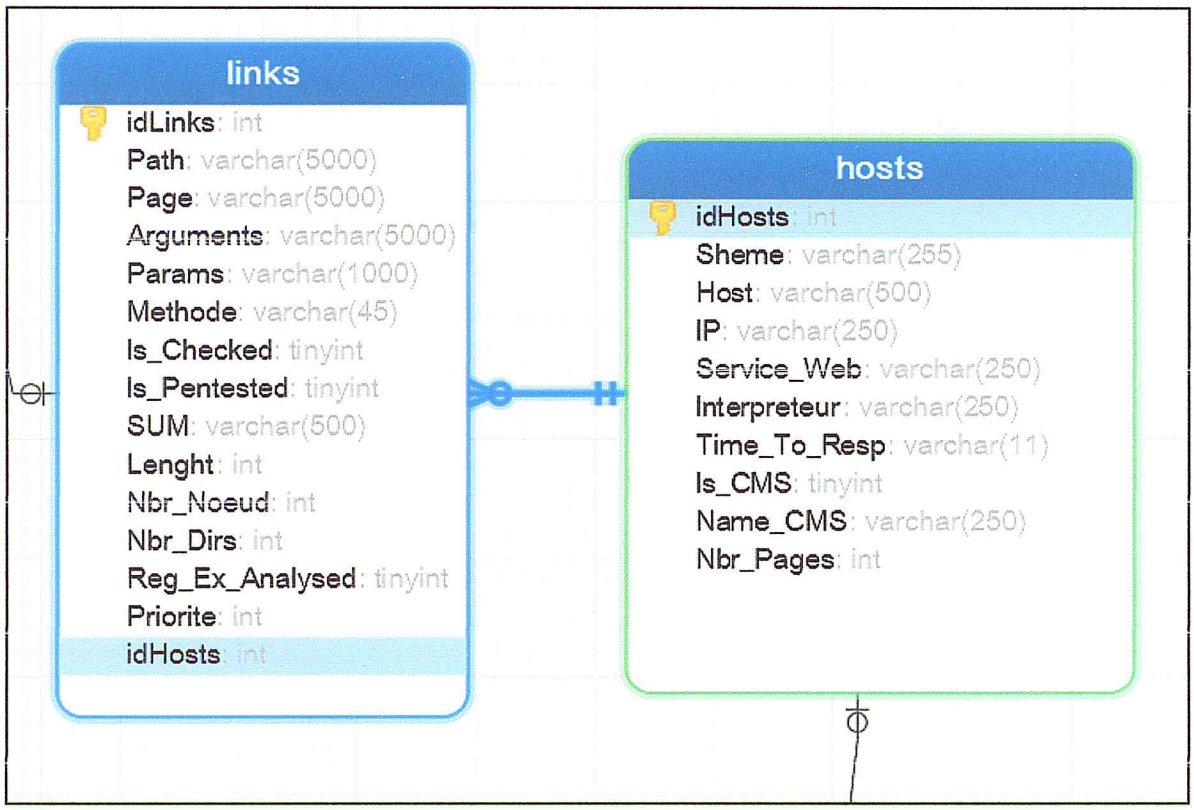


Figure IV-11 Table hosts et links

a) Principe de fonctionnement :

Le principe de fonctionnement de notre crawler est présenté dans la Figure IV-12

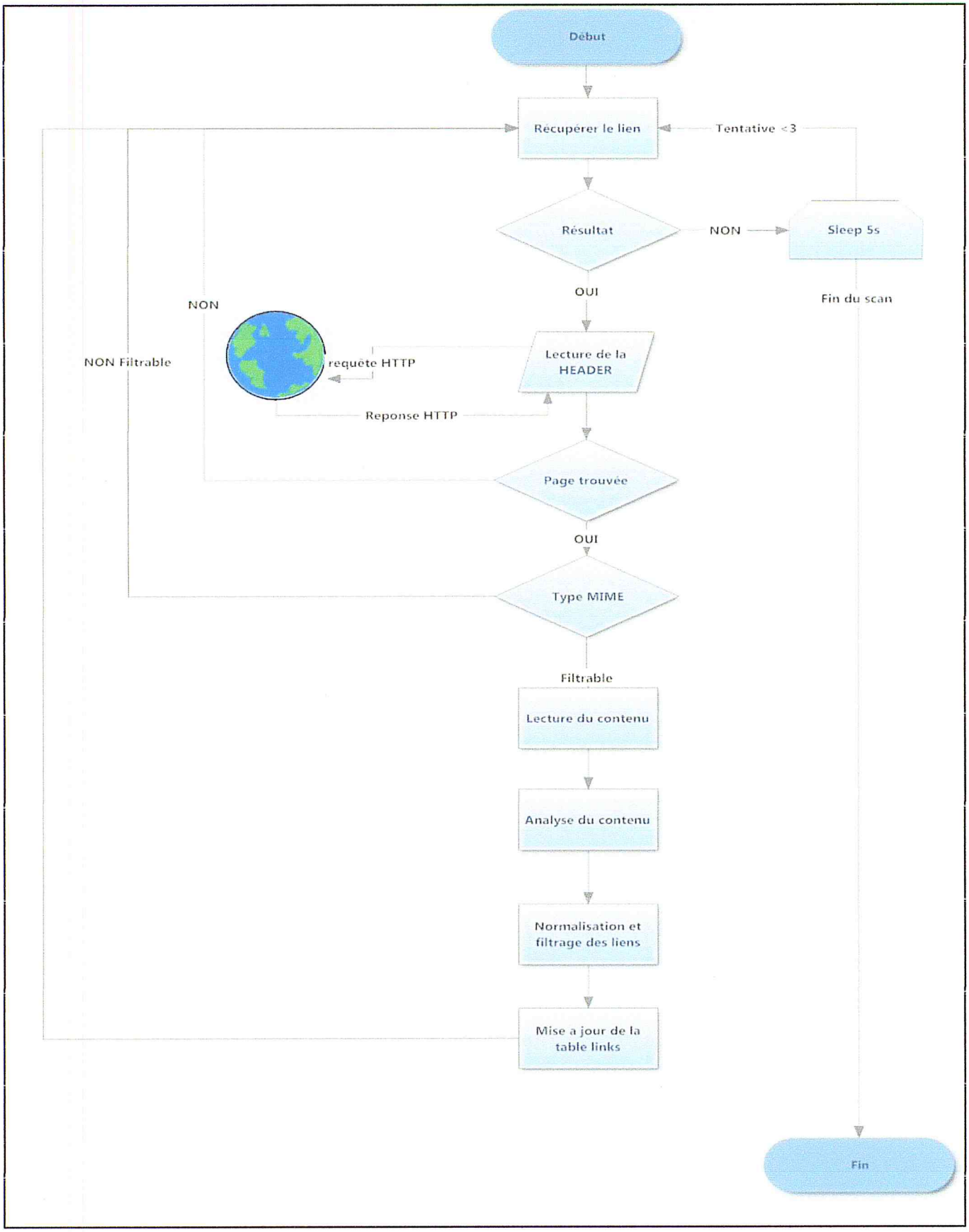


Figure IV-12 Organigramme de fonctionnement du crawler

b) *Explication des fonctions du crawler :*

- **Récupération du lien :** La récupération du lien se fait depuis la base de données (Table « links »), la sélection du lien met à jour le champ « Is_Checked » pour indiquer qu'il est en cours de traitement et éviter la cohésion.
- **Lecture de la header :** La lecture de la header suite à la demande de la page indique si cette page est valide.
- **Type mime :** Selon (https://tools.ietf.org/html/rfc2045) mime veut dire «Multipurpose Internet Mail Extensions/ Extensions de messagerie Internet multifonction» et il indique le type et le sous-type du contenu, il sert à établir une vérification sur le type mime nous permet d'éviter la lecture de contenu non filtrable (image, son, etc …)
- **Lecture et analyse du contenu :** Le contenu une fois récupéré subit une analyse, et une récolte d'informations (calcule du SUM, Nombre de nœuds etc. …) sur la page courante qui met à jour la table « links », puis cible les balises contenant des liens menant vers d'autres pages.
- **Normalisation des liens** : Les liens récupérés peuvent être de différents types et formes ce qui nous a amené à les normaliser selon un standard commun (voir Tab 4.3)
- **Filtrage des liens :** Récupération des liens qui appartiennent au même domaine et ignorer les liens aux quelles les expressions régulières générés par le module « optimisation du crawler » font référence.
- **Mise à jour de la table « links » :** Sauvegarde des liens récupérés, seulement les nouveaux liens non répertoriés seront insérés dans la table « links ».

Hôte	hyperliens	Lien normalisé
http://www.monsite.com/Rép1/page.php	?id=34	http://www.monsite.com/page.php?id=34
http://www.monsite.com/Rép1/page.php	/Rép2/Rép3/page2.php	http://www.monsite.com/Rép2/Rép3/page2.php
http://www.monsite.com/Rép1/page.php	Rép2/Rép3/page2.php	http://www.monsite.com/Rép1/Rép3/page2.php
http://www.monsite.com/Rép1/page.php	www.monsite.com	http://www.monsite.com
http://www.monsite.com/Rép1/page.php	monsite.com	http://www.monsite.com
http://www.monsite.com/Rép1/page.php	//www.monsite.com/Rép2/page.php	http://www.monsite.com/Rép2/page2.php
http://www.monsite.com/Rép1/page.php	/../../Rép2/../	http://www.monsite.com
http://www.monsite.com/Rép1/page.php	/Rép1/page.php#MessageTxt	http://www.monsite.com/Rép1/page.php

Tableau IV-1 Normalisation des liens

5. Le module « Optimisation du crawler » :

Les sites web ayant un contenu d'informations riche et utilisant le mod_rewrite pour la réécriture des Urls sont amenés a générer un nombre important de liens qui dans une approche classique se verrons tous cartographier, ce qui nous a poussé à réaliser une optimisation du crawling en se basant sur la distance de levenshtein et le nombre de nœuds, c'est-à-dire que ce module compare une multitude de liens ayant le même nombre de nœud (nombre de balises structurante de la page), puis grâce à la fonction de Levenshtein, analyser la distance entre les Urls. Si le taux de correspondance et supérieur ou égale à 80%, la partie variable est déterminée et nous permet de générer une expression régulière qui correspond aux liens sélectionnés, cette dernière est stockée dans la base de donnée puis utilisée par le module précèdent (Crawler).

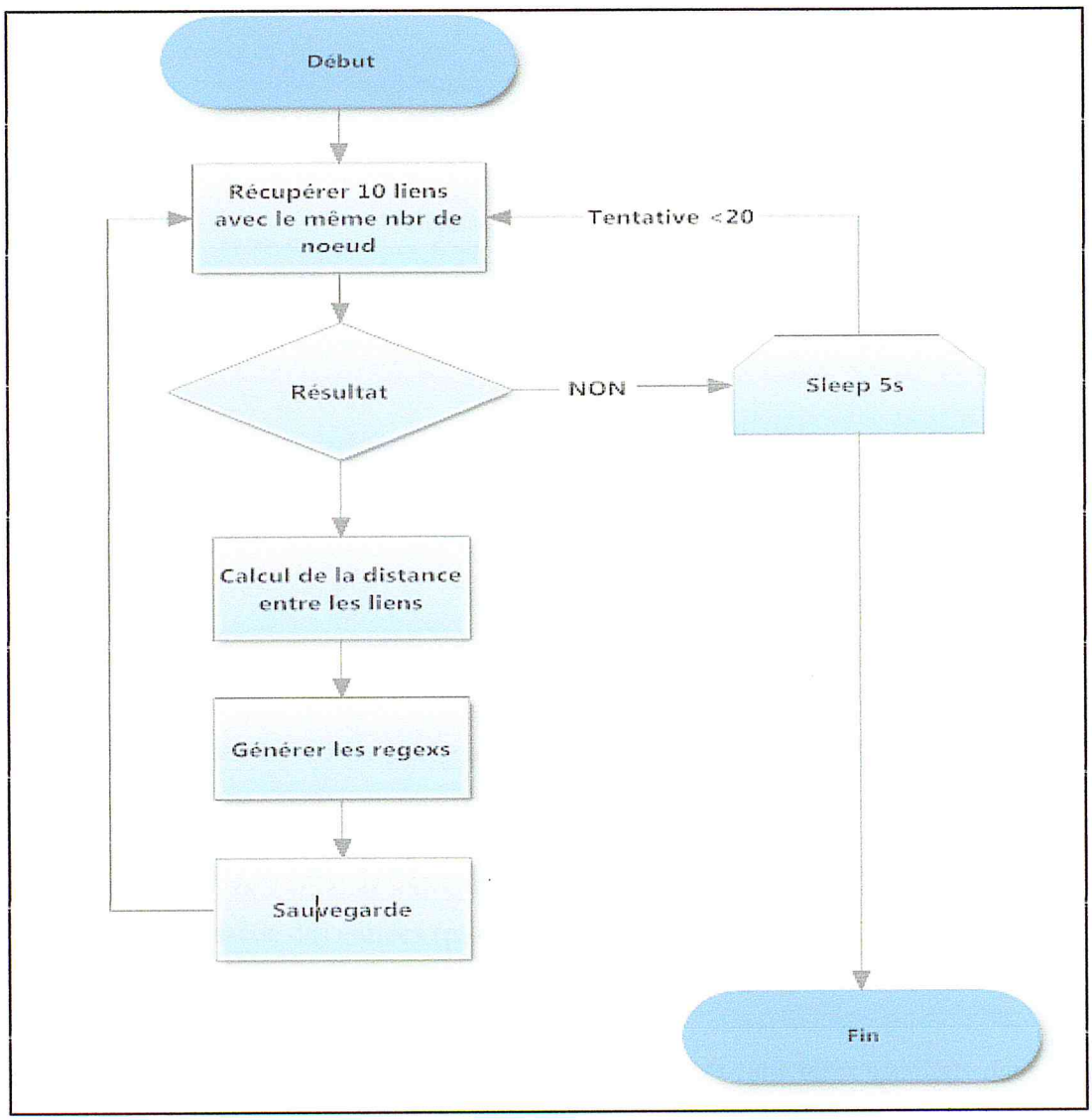


Figure IV-13 Organigramme de fonctionnement du module 3(optimisation du crawler)

nous tenterons de changer le type de chaque variable ce qui aura pour but de trouver les fonctions liées.

Une fois l'injection faite, nous analysons le contenu de la page de réponse. Les vulnérabilités seront identifiées grâce à des expressions régulières qui balayeront les messages d'erreurs présents sur la page.

Chaque injection qui se solde par un succès est stockée dans la base de données (Figure IV-15):

Figure IV-15 Table Vuln_links

En finalisant l'injection dans tous les liens, le module s'attaque à une analyse par fuzzing qui a pour but de trouver sur le host des fichiers compromettant la sécurité de l'application. Ces résultats sont stockés dans la base de données (Figure IV-16) :

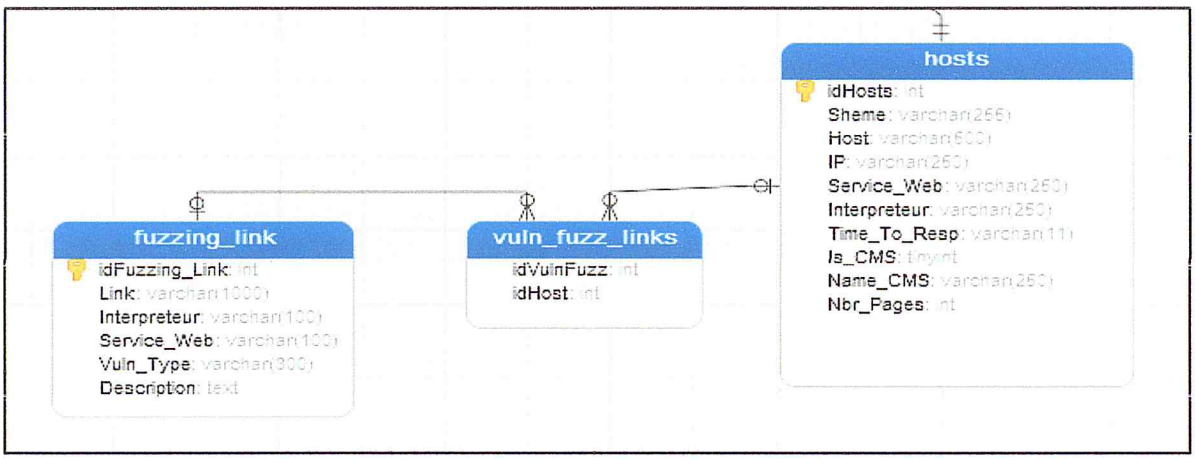


Figure IV-16 Tables des fichiers compromettants

C. Conclusion :

Dans ce chapitre, nous avons abordé en détail chaque module de notre scanner à travers des organigrammes explicites. La conception nous a permis de mettre en œuvre la faisabilité de notre application à travers la détermination des principaux composants et des scenarios pour comprendre le fonctionnement de notre system.

Dans le chapitre suivant, nous allons nous intéresser à l'implémentation de notre application en se basant sur la conception qu'on a présentée dans ce chapitre, ainsi qu'aux tests de notre system d'audit.

Chapitre V

Réalisation

V. Réalisation

A. Introduction

La réalisation de notre projet consiste à appliquer au mieux ce qui a été décrit au niveau de la conception.

Dans ce chapitre nous présentons l'environnement et les outils de développement utilisés ainsi que les fonctionnalités générales du scanner illustrées par des captures d'écran commentées et des tests de validation que nous avons effectué pour nous assurer de son bon fonctionnement.

B. Environnement de développement :

1. Système d'exploitation :

Nous avons développé notre application sous Ubuntu 16.04.3 LTS.

2. Environnement logiciel :

Pour l'implémentation de l'application, nous avons utilisé le langage open source python 2.7

Python est un langage portable, dynamique, extensible, gratuit qui permet une approche modulaire et orienté objet de la programmation.

Il est développé depuis 1989 par Guido Van Rossum et de nombreux contributeurs bénévoles.

- Python est portable, non seulement sur les différentes variantes « d'UNIX », mais aussi sur les OS propriétaires : « Mac Os » et les différentes variantes de Windows
- La syntaxe de « Python » est très simple et combinée à des types de données évolués (listes, dictionnaires,..), et conduit à des programmes à la fois très compacts et très lisibles
- Il intègre, un système d'exceptions, qui permettent de simplifier considérablement la gestion des erreurs
- Il est orienté-objet
- Python est un langage de programmation haut niveau, interprétable et complet grâce aux nombreuses bibliothèques spécialisées.
- Il est adapté pour un usage professionnel, en sécurité informatique, une multitude d'applications et d'exploits sont écrits en Python. [25]

Et pour l'affichage des résultats nous avons opté pour une interface web en utilisant tout de « HTML », « CSS », « PHP », et « jQuery »

Notre application interagie avec une base de données à différents niveaux de l'exécution comme expliqué lors de la conception.

Nous avons utilisé le SGBD MySql Version 5.5

MySQL est à la fois une des bases de données très utilisées au monde essentiellement grâce à ses performances sur le Web parmi ces avantages :

1. Il est open source, ce qui signifie qu'il est gratuit et que tout le monde peut l'utiliser et le modifier
2. Il est largement déployé : « MySQL » peut être installé sur de multiples et différents plateformes, et il est habituellement un standard sur la plupart des configurations d'hébergement Web
3. Il est facile à utiliser : Mettre en place et travailler avec des bases de données « MySQL » est relativement simple
4. Connexion et sécurité : « MySQL » dispose d'un système de sécurité permettant de gérer les personnes et les machines pouvant accéder aux différentes bases.
5. Il fonctionne bien avec « PHP »

Pour modéliser notre base de donnés nous avons utilisé « MySQL Workbench »

« MySQL Workbench » est un logiciel de schématisation de tables, de gestion et d'administration de bases de données « MySQL ».

Via une interface graphique intuitive, il permet, entre autres, de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données. Pour ce faire, il doit être connecté à un serveur « MySQL ».

5.2.4 Modules et outils utilisée « Python » :

Durant notre développement des différents module, nous avons utilisé quelques module propres a Python qui contienne une multitude de méthodes qui nous ont facilité la tâche dans ce qui suit nous citerons les modules utilisé au cours de notre développement :

- Urllib : Ce module fournit une interface de haut niveau pour récupérer des données sur le Web. En particulier, la fonction urlopen() est similaire à la fonction intégrée open(), mais accepte les URLs au lieu des noms de fichiers, Certaines restrictions s'appliquent, on ne peut ouvrir des URL qu'en lecture, et on ne peut pas effectuer d'opération de recherche
- Urlib2 : Le module urllib2, version plus avancée qu'urllib, utilise le module httplib, pour proposer des fonctionnalités d'accès à des URL (Universal Ressource Locator). Urllib2 gère tous les aspects du protocole HTTP, comme l'authentification, les cookies, les redirections, ou encore le flux sécurisés.
- Urlparse : comme son nom l'indique, ce module sert à décomposer une URL en ses différents éléments par exemple :

  Url=http://www.exemple.com/Rep1/Rep2/Rep3/page.php ?Arg1=123&Arg2=hell

  Quand nous appliquons urlparse sur notre url (Url), nous aurons le résultat suivant :

  - Sheme='http'
  - netloc='www.exemple.com'
  - path= '/Rep1/Rep2/Rep3/page.php '
  - query =' Arg1=123&Arg2=hello'
- Time : Ce module nous fournit des fonctions de manipulations de temps
- Os : Le module Os regroupe quelques 200 fonctions ou objets qui sont dans certains cas des alias vers des éléments d'autres modules. On peut regrouper ces éléments en quatre sous-ensembles :
  - opérations sur les descripteurs de fichiers ;
  - manipulation des fichiers et répertoires ;
  - manipulation des processus ;
  - informations sur le système.
- Re (Regular Expression) : Les expressions régulières sont un puissant moyen de rechercher et d'isoler des expressions d'une chaîne de caractères. Le module « re » nous a permis de manipuler et de réaliser très rapidement et facilement des recherches sur des chaines de caractères.
- Requests : Request est une bibliothèque Apache2 licence HTTP, écrite en Python, pour les êtres humains elle permet d'envoyer des demandes HTTP/1.1. Nous pouvons

ajouter des en-têtes, les données de formulaire, et les paramètres avec un simple dictionnaire de Python, et accéder à des données de réponse de la même manière.

- BeautifulSoup : Ce module est très fiable et couramment utilisé en tant que parseur. Il permet le parcours, la lecture et la modification d'arborescences HTML et XML, le point fort de BeautifulSoup est qu'il est capable de comprendre un document HTML mal formé en utilisant des heuristiques semblables à celle utilisées par les navigateurs dans le même but.

C. Architecture du système :

L'architecture d'un système définit sa structure en termes de composants fonctionnels ou physiques réalisant des exigences qui interagissent entre eux et l'environnement. Dans les applications web, et quel que soit leurs complexités, il est toujours conseillé d'analyser leurs architectures selon deux niveaux logique :

- La logique présentative : présente des résultats à l'utilisateur, gère l'affichage, gère l'affichage, et prend en charge les traitements locaux.
- La logique applicative : gère les données de l'application et exécute les traitements globaux.
- Données : c'est l'ensemble des moyens et mécanismes qui permettent de gérer les données que l'application utilise.

D. Présentation du scanner :

Comme cité précédemment, nous avons opté pour une interface web qui est facile d'utilisation et offre une interaction minime avec l'utilisateur, puisque ce dernier n'aura pas à interagir avec l'application à plusieurs reprises. Après l'insertion d'adresse du lien (*host*) à scanner, nos modules (scripts) s'exécutent en arrière-plan et l'exécution sera transparente vis-à-vis de l'utilisateur puisque nous avons réservé un champ qui afficheras en temps réel le contenue de certaines tables de notre base de données.

Notre application est hébergé sur un serveur externe fournis par SSH, qui a l'adresse « 94.177.197.20 » et qui est géré par un software de gestion des fonctionnalités Web (tel que la gestion de la base de données et interactions) nommé « BlueBox » sous Linux Ubuntu 16.04.3 LTS et qui utilise les services « *apache/2.4.18* » et « *MySQL* ».

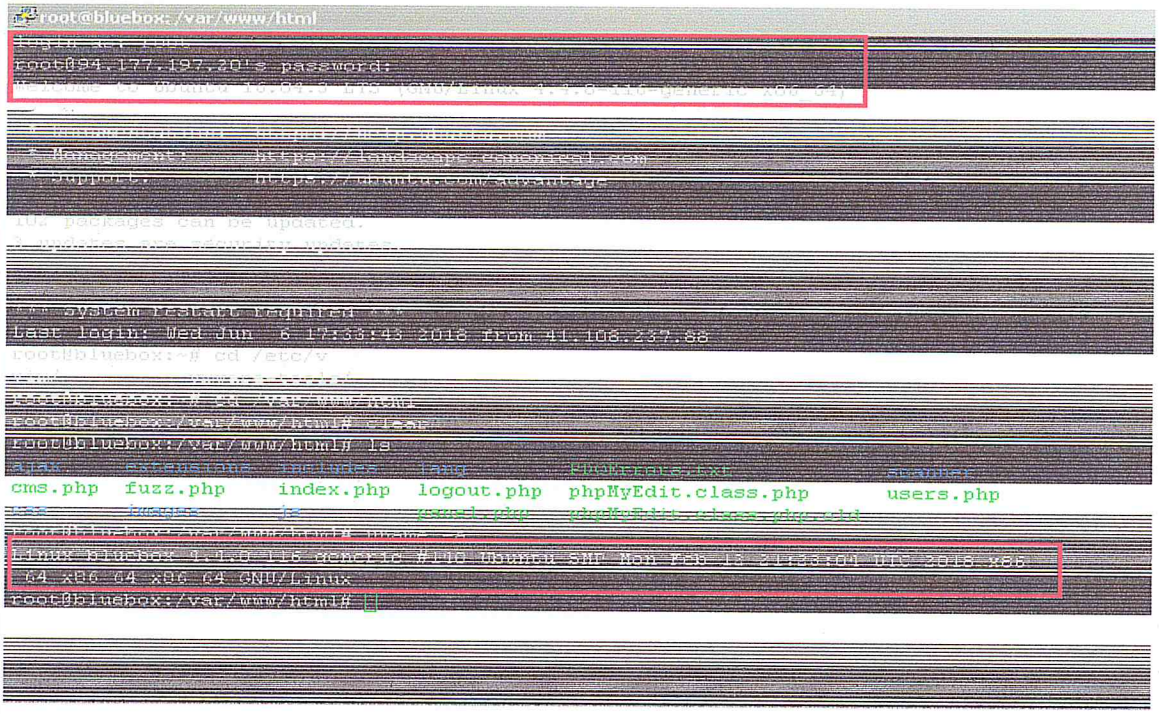


Figure V-1 Les informations du serveur externe

1. Page d'accueil « index.php » :

Pour avoir accès à l'application web nous allons ouvrir un navigateur et y introduire l'adresse de notre application « 94.177.197.20 »

Figure V-2 Demande d'authentification

Une authentification est nécessaire pour continuer vers la page d'accueil et l'interface principale de l'application.

2. Interface principale :

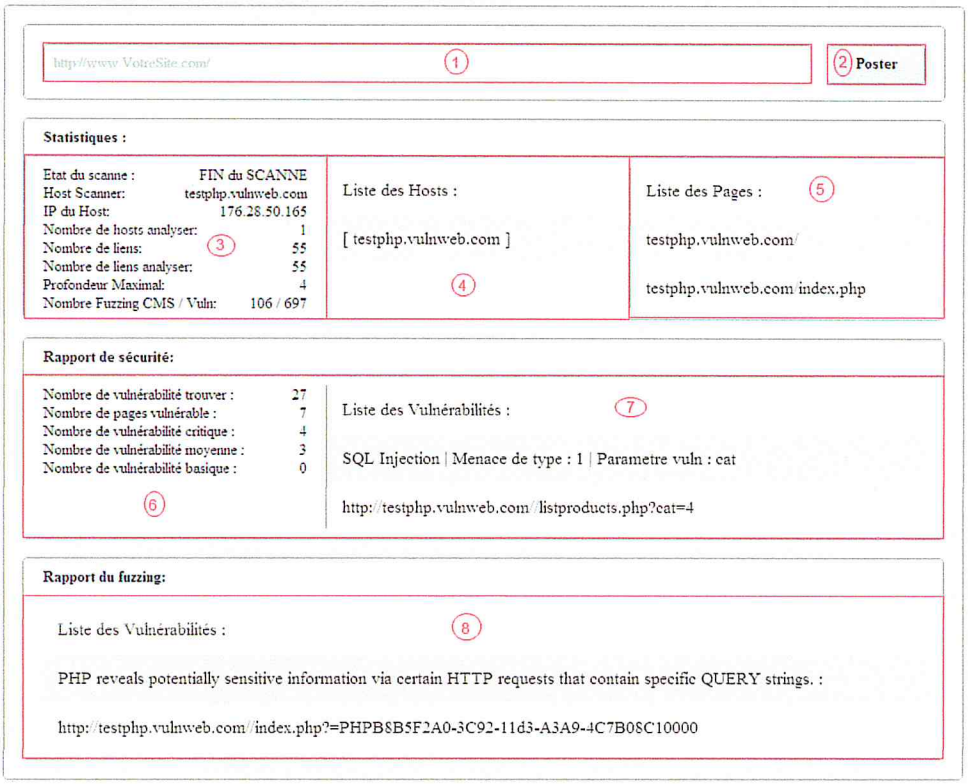


Figure V-3 Interface principale

1- Insertion d'adresse valide du *host* à scanner
2- Le bouton « Poster » permet de lancer le scan
3- La récolte d'informations sur le *host* inséré commence, nous trouvons par exemple :
   - IP de l'Hôte : la traduction en IP de l'adresse du *host* inséré.
   - Etat du scanne : permet de savoir si le scan est toujours en cours ou non.
   - Nombre de liens : le nombre de liens « crawlé » ou trouvé.
   - Nombre de lien analysé : c'est le nombre de lien qui ont subis une injection.
4- Liste des *hosts* à scanner : la racine et les sous domaines en cas où ces derniers existent.
5- Liste des pages : regroupe les pages trouvés par le crawler (module deux de notre conception)
6- Rapport de sécurité : il affiche un genre de rapport ou statistique qui regroupe :

- Nombre de vulnérabilités trouvés : la somme de toutes les vulnérabilités trouvé au cours de son scan.
- Nombre de pages vulnérables.
- Nombre de vulnérabilité critique : le nombre de vulnérabilité avec une criticité élevé (chaque vulnérabilité a une criticité).

7- Listes des vulnérabilités : ce champs regroupe et affiche les liens vulnérables en incluant le *PATH*, le paramètre ou l'argument vulnérable ainsi que le type de vulnérabilité.

8- Listes des vulnérabilités : Ce champs regroupe et affiche les liens vulnérables en incluant le *PATH* ainsi que le type de la vulnérabilité est cela grâce à la technique du *fuzzing*.

La figure suivante nous montre d'autres fonctionnalités offertes par notre application

Figure V-4 Autres fonctionnalités de l'application

1. Scanners de vulnérabilité : permet de faire un retour vers le scanner (page principale).
2. Gestion des CMS : ce lien nous permet de manipuler la base de donnée plus précisément la table CMS, qui nous permet de faire un *fuzzing* sur les noms des CMS, donc ce lien assure l'ajout et la modification de ces derniers (Figure V-5)
3. Gestion des utilisateurs : comme son nom l'indique, ce lien nous permet de gérer les utilisateurs qui ont accès à cette interface (Figure V-6)
4. Gestion du *Fuzzing* : ce lien nous dirige vers l'insertion ou la modification de nouveau lien d'attaque par *fuzzing*
5. Déconnexion : ce lien nous permet de se déconnecter de la session en cours

3. Gestions des CMS :

v	Name	Path File	Interpreteur	Favico MD5
X	Tri: IdCMS croissant			
•	Apache on Redhat		Unknown	71e30c507ca3fa005e2d1322a5aa8fb2
○	iPlanet Web Server Enterprise Edition 6.0		Unknown	b25dbe60830705d98ba3aaf0568c456a
○	Netscape 4.1		Unknown	226ffc5e483b85ec261654fe255e60be
○	Netscape 6.0		Unknown	41e2c893098b3ed9fc14b821a2e14e73
○	NetScreen WebUI or 3Com Router		Unknown	f1876a80546b3986dbb79bad727b0374
○	JBoss Server		java	799f70b71314a7508326d1d2f68f7519
○	Horde Groupware Webmail 1.0.1 (Mnemo Theme)		Unknown	f5f2df7eec0d1c3c10b58960f3f8fb26
○	Apache Tomcat		Unknown	4644f2d45601037b8423d45e13194c93
○	Apache 2.2.4		Unknown	31aa07fe236ee504c890a61d1f7f0a97
○	Serena Collage 4.6		Unknown	bd0f7466d35e8ba6cedd9c27110c5c41
○	Horde IMP 3.1.4 or Horde Groupware Webmail 1.0.1		php	7cc1a052c86cc3d487957f7092a6d8c3
○	Horde IMP 4.1.4 or Horde Groupware Webmail 1.0.1		php	f567fd4927f9693a7a2d6cacf21b51b6
○	SunOne 6.1		java	a28ebcac852795fe30d8e99a23d377c1
○	Horde Groupware Webmail 1.0.1 (Nag Theme)		php	81df3601d6dc13cbc6bd8212ef50dd29
○	Horde Groupware Webmail 1.0.1 (Ingo Theme)		php	919e132a62ea07fce13881470ba70293
○	Horde Groupware Webmail 1.0.1 (Turba Theme)		php	ff260e80f5f9ca4b779fbd34087f13cf
○	Google Web Server		Unknown	4987120f4fb1dc454f889e8c92f6dabe
○	Horde Groupware Webmail 1.0.1 (Kronolith Theme)		Unknown	a5b126cdeaa3081f77a22b3e43730942
○	Aruba Networks device		Unknown	dc0816f371699823e1e03e0078622d75
○	Apache HTTP Server on Apple Mac OS X Server		Unknown	d41d8cd98f00b204e9800998ecf8427e

<< < Ajouter Afficher Modifier Copier Supprime > >> Aller a 1 ▾ Page: 1 / 6 Enregistrements: 106

Figure V-5 Gestion des CMS

Le lien de Gestion des CMS nous amène vers cette page qui nous donne le choix d'ajouter ou de modifier ou bien même de supprimer un des CMS déjà présents sur la table

Par exemple si nous voulons ajouter une ligne dans cette table, on clique sur le bouton ajouté qui nous dirige vers une nouvelle page (Figure V-6)

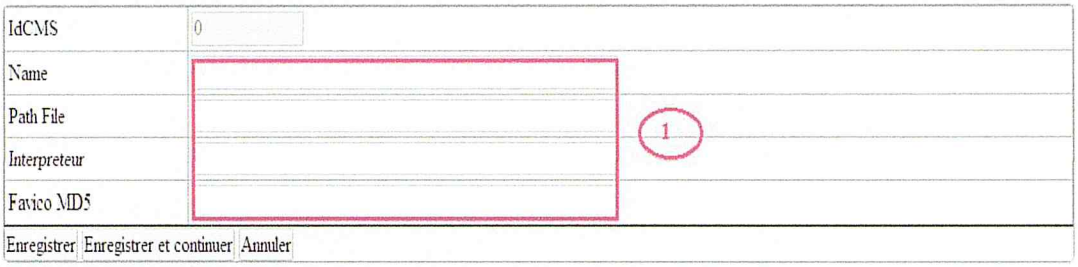


Figure V-6 Ajout d'un CMS

Cette figure nous montre l'opération d'ajout d'un CMS.

4. Gestions des utilisateurs :

Le lien de Gestion des utilisateurs nous amène vers cette page qui nous donne le choix d'ajouter ou de modifier les utilisateurs

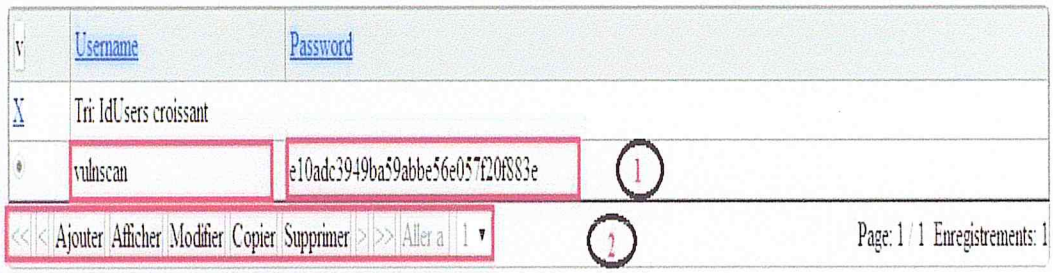


Figure V-7 Gestions des utilisateurs

1- Nom et mot de passe hashé d'un utilisateur
2- Cette barre nous permet de manipuler les utilisateurs

5. Gestion du *Fuzzing* :

Pour la gestion du *fuzzing* on a au préalable importé (depuis la base de connaissance de Nikto) et fabriqué des *templates* ou des liens d'attaque par *fuzzing*. Aussi on peut gérer ces liens-là.

v	Link	Interpreteur	Description	Find	CMS
X	Tri IdFuzzing Link croissant				
	vgn/record previewer		Vignette CMS admin maintenance script available.	200	Vignette
	vgn/stylepreviewer		Vignette CMS admin maintenance script available.	200	Vignette
	vgn/vr/Deleting		Vignette CMS admin maintenance script available.	200	Vignette
	vgn/vr/Editing		Vignette CMS admin maintenance script available.	200	Vignette
	vgn/vr/Saving		Vignette CMS admin maintenance script available.	200	Vignette
	vgn/vr/Select		Vignette CMS admin maintenance script available.	200	Vignette
	blah_badfile.shtml	JSP	Allaire ColdFusion allows JSP source viewed through a vulnerable SSI call.	200	
	blah-whatever-badfile.jsp	JSP	The web server is configured to respond with the web server path when requesting a non-existent .jsp file.	Script	
	vgn/style		Vignette server may reveal system information through this file.	200	Vignette
	SiteServer/Admin/commerce/foundation/domain.asp	ASP	Displays known domains of which that server is involved.	200	
	SiteServer/Admin/commerce/foundation/driver.asp	ASP	Displays a list of installed ODBC drivers.	200	
	SiteServer/Admin/commerce/foundation/DSN.asp	ASP	Displays all DSNs configured for selected ODBC drivers.	200	
	SiteServer/admin/findvserver.asp	ASP	Gives a list of installed Site Server components.	200	
	SiteServer/Admin/knowledge/dsmgr/default.asp	ASP	Used to view current search catalog configurations	200	
	basilix/mbox-list.php3	PHP	BasiliX webmail application prior to 1.1.1 contains a XSS issue in 'message list' function page	200	

<< < Ajouter Afficher Modifier Copier Supprimer > >> Aller a 2 ▾ Page: 2 / 47 Enregistrements: 697
23.002 miliseconds

Figure V-8 Gestion du Fuzzying

6. Insertion du host :

Au moment de l'insertion du lien a scanné notre application vérifie la validité du lien, il faut que le lien soit conforme et normalisé et que le champ ne soit pas vide avant son insertion, dans le cas contraire notre scanner affiche une alerte (le champ d'insertion en rouge Figure V-7), si le lien est conforme et valide après une vérification faite par l'application le champ devient vert signe de validité du lien inséré (Figure V-8)

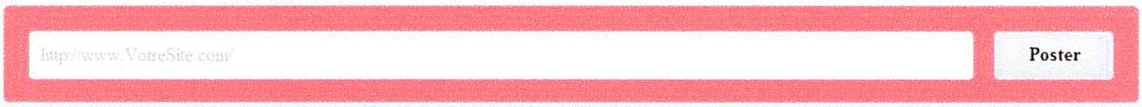


Figure V-9 Cas d'un lien non valide (lien vide)

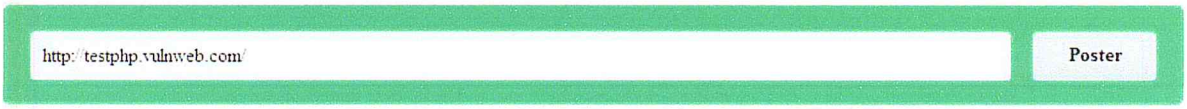


Figure V-10 Cas d'un lien valide (lien de test)

E. Test d'évaluation :

Nos tests d'évaluation vont être sur un site vulnérable programmé exprès avec des vulnérabilités pour des fins de tests d'intrusions. Ce site a été développé par l'équipe d'« Acunetix » http://testphp.vulnweb.com. Notre scanner en a détecté la plupart de ces dernières, au cours de notre scan on a recensé plus de vingt vulnérabilités sur différentes pages la figure suivante :

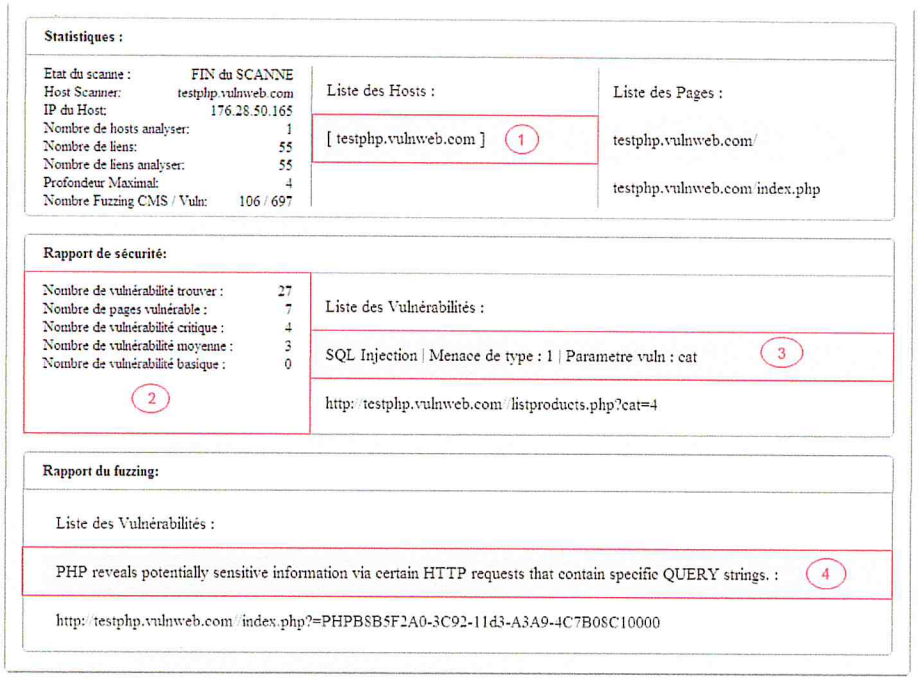


Figure V-11 Résultat d'un Scan

La (Figure V-10) nous résume ces failles dans un rapport de sécurité généré par notre application où :

1. Le host scanné qui bel est bien http://testphp.vulnweb.com
2. Le rapport de sécurité
3. Liste de vulnérabilités : il affiche quelques détails sur les vulnérabilités
   - Nom de la vulnérabilité : SQL Injection
   - Le Type de la menace : 1 (critique)
   - Paramètre Vulnérable : dans notre teste c'est le paramètre « test »
4. Liste de vulnérabilités : il affiche quelques détails sur les vulnérabilités
   - Description de la vulnérabilité : dévolution d'information sensible http
   - Le lien de la vulnérabilité

Maintenant qu'on sait que notre Scanner est fonctionnel on peut passer aux tests et comparaisons avec les autres scanners.

Expérimentations et résultats comparatifs :

On fait nos tests sur un site créé spécialement par l'équipe d'Acunetix pour faire des tests de vulnérabilités l'adresse de ce site est : « http://testphp.vulnweb.com/ ».

On a tenu à faire des tests comparatifs avec Acunetix(Consultant Edition) car il est considéré comme l'un des meilleurs scanners web.

Statistiques :

Etat du scanne :	FIN du SCANNE	Liste des Hosts :	Liste des Pages :
Host :	testphp.vulnweb.com		
IP du Host:	176.28.50.165	[testphp.vulnweb.com]	testphp.vulnweb.com/
Nombre de hosts analyser:	1		
Nombre de liens:	55		testphp.vulnweb.com/index.php
Nombre de liens analyser:	55		
Profondeur Maximal:	4		
Fuzzing CMS / Vuln:	106 / 6845		

Rapport de sécurité:

Nbr de vulnérabilité trouver :	27	Liste des Vulnérabilités :
Nbr de pages vulnérable :	7	
Nbr de vulnérabilité critique :	4	SQL Injection | Menace de type : 1 | Parametre vuln : test
Nbr de vulnérabilité moyenne :	0	
Nbr de vulnérabilité basique :	3	http://testphp.vulnweb.com//search.php?test=query

Figure V-12 Une vulnérabilité SQL Injection détecté

Statistiques :

Etat du scanne :	FIN du SCANNE	Liste des Hosts :	Liste des Pages :
Host :	testphp.vulnweb.com		
IP du Host:	176.28.50.165	[testphp.vulnweb.com]	testphp.vulnweb.com/
Nombre de hosts analyser:	1		
Nombre de liens:	55		testphp.vulnweb.com/index.php
Nombre de liens analyser:	55		
Profondeur Maximal:	4		
Fuzzing CMS / Vuln:	106 / 6845		

Rapport de sécurité:

Nbr de vulnérabilité trouver :	27	XSS Injection | Menace de type : 3 | Parametre vuln : artist
Nbr de pages vulnérable :	7	
Nbr de vulnérabilité critique :	4	http://testphp.vulnweb.com/listproducts.php?artist=2
Nbr de vulnérabilité moyenne :	0	
Nbr de vulnérabilité basique :	3	XSS Injection | Menace de type : 3 | Parametre vuln : pp

Figure V-13 Une vulnérabilité XSS détecté

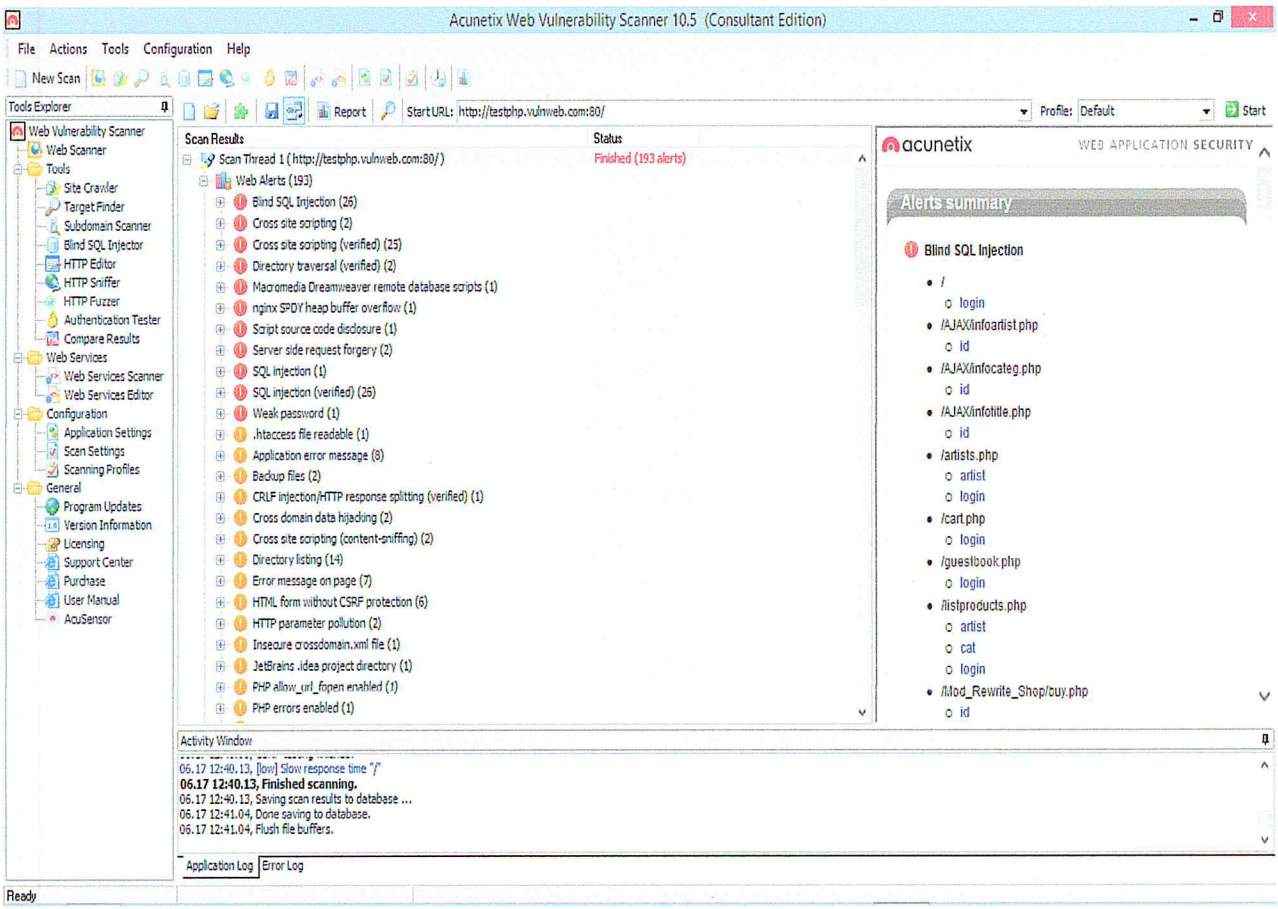


Figure V-14 Résultats d'un scan avec Acunetix(Consultant Edition)

Notre scanner a réussis à trouver : 29 failles notamment SQL Injection et XSS (des types qu'on a définis dans les objectifs de conception) sur différentes pages et causés par diffèrent attributs.

Quant au scanner Acunetix, il en a trouvé 193 dont 54 qui font partie des vulnérabilités traités par notre scanner.

Nos résultats de scan représentent 53,7% d'efficacité en comparaison avec ceux d'Acunetix ce qui est due à leurs bases de connaissances qui sont plus grandes mais aussi aux grammaires plus riches et plus exigeantes qu'ils ont amélioré au cours des versions.

Nos résultats comparés à un scanner surpuissants sont très satisfaisants compte tenu à toute la connaissance, l'expérience accumulé par l'éditeur des scanners mais surtout que ce scanner Acunetix (Consultant Edition) coute énormément chère (plus de 6000$[6]).

F. Conclusion :

Dans ce chapitre, nous avons décrit l'implémentation de notre scanner de vulnérabilités web et les tests qui ont été menés pour voir l'efficacité de notre scanner ainsi que sa validation.

Dans la partie implémentation nous avons détaillé les outils et les langages de programmation utilisés pour mettre en œuvre notre solution.

Nous avons achevé ce chapitre par un ensemble de tests qui nous ont permis de voir réellement comment fonctionne le scanner et voir l'efficacité des fonctionnalités de ce dernier en comparaison avec un scanner qui a déjà fait ses preuves.

La comparaison a indiqué que les tests ont donné de bons résultats.

[6] https://www.acunetix.com/ordering/

Conclusion

Toutes ces nouvelles technologies web ont créé un vaste champ d'attaque pour les cybercriminels et plus précisément, les applications web qui sont les plus touchés au dépit de toute catégorie d'utilisateurs légitimes, des propriétaires de sites web et des fournisseurs de solutions web.

Pour y remédier il existe des outils qui aident à garder un certain contrôle sur la sécurité des applications, notamment, les scanners web qui ont pour fonctionnalité d'identifier les vulnérabilités et les brèches afin de les colmater.

Notre travail a consisté à réaliser un system d'audit des applications web (scanner web) en boite noir en utilisant l'approche par reconnaissance de messages d'erreurs, ce qui a été accomplis par étapes en passant par l'étude des vulnérabilités et des scanners web pour pouvoir faire une étude conceptuelle efficace qui comportait quatre modules essentiels et complémentaires qui interagissaient avec notre base de donnée selon un schéma spécifique pour une traçabilité et une efficacité accrue, l'implémentation de notre scanner web a été faite sur un system Linux qui nous permit une utilisation plus aisé des outils complémentaires à notre scanner. La mise en circuit de notre scanner a été faite sur un serveur distant pour un accès non contraignant.

On a pu voir les fonctionnalités fournis par notre scanner, ainsi que les tests qui ont été faits sur un site dédié aux tests de scanners de vulnérabilité web créé par Acunetix.

On a fait une étude comparative entre notre scanner et le scanner d'Acunetix (Consultant Edition) qui est considéré comme l'un des meilleurs, l'étude a été faite sur leur site où on a pu voir l'efficacité de notre scanner par comparaison de celui d'Acunetix en fonction des vulnérabilités trouvés, le taux du nombre des failles trouvé en comparaison au scanner d'Acunetix était de 53,7% ce qui est assez satisfaisant. Et donc nos objectifs atteints.

Pour les prochaines versions et améliorations il est à prévoir d'augmenter le nombre de vulnérabilités traités par notre scanner, mais aussi mettre un system de prévention des faux positifs, aussi on envisage de mettre en place un système de détection des vulnérabilités coté serveur, et ce qui ne serais pas impossible c'est de faire en sorte que notre scanner proposes des corrections pour les failles et vulnérabilités rencontrés.

Bilan du stage

Bilan du stage

A l'issue de ce stage, les principaux objectifs définis lors de la présentation du projet sont atteints. En effet les premiers modules de l'application sont actuellement en cours de test chez l'organisme d'accueil (Smart Solutions Hosting).

La mise en place des modules développés pour le « moyen terme » et constituant la nouvelle version 2.0 du logiciel sera proposé à Smart Solutions Hosting dans les mois à venir.

Des évolutions du logiciel restent néanmoins possibles, afin d'améliorer les possibilités offertes par les nouveaux modules ou pour corriger d'éventuels bugs ou problèmes d'utilisation pouvant apparaître au cours de son exploitation.

L'ajout de nouveaux modules à d'ores et déjà été prévu pour permettre dans une nouvelle version d'avoir accès à d'autre types de test et représentations statistiques.

Une fois le projet terminé et en attendant d'éventuels retours de la part de SSH, On a pris part au développement d'une autre application de monitoring des points d'accès, des serveurs etc …

Ce stage au sein de l'entreprise Smart Solutions Hosting, sous la responsabilité de Monsieur Guia Brahim Fouad, nous a beaucoup apporté aussi bien d'un point de vue professionnel que personnel.

Tout d'abord, on a pu développer et parfaire nos connaissances dans le domaine de la sécurité informatique.

Tout comme sa nous a permis de nous rendre compte des difficultés et points forts de python pour le développement d'applications pour l'audit.

Durant ce projet on a aussi pu apprendre à gérer un projet de l'analyse à la livraison du programme en suivant les exigences du client.

Le stage a été une bonne opportunité d'utiliser nos connaissances pour l'organisation du projet, au niveau de la rétro-conception, de l'analyse et du développement. On a ainsi pu mettre en pratique nos connaissances en programmation PHP et SQL, Python etc.

Parfaire l'utilisation de logiciels comme Acunetix et découvrir de nouveaux langages et outils.

D'un point de vue personnel, nous sommes satisfait d'avoir fait des recherches au point de vue technique et mis en place de nouveaux modules et techniques pour le scanner de vulnérabilité.

On a en plus acquis une certaine autonomie de travail ainsi qu'une plus grande assurance dans les choix techniques.

En ce qui concerne la vie en entreprise, il nous a été facile de nous intégrer dans une entreprise composée d'une équipe accueillante, toujours prête à répondre à nos questions.

Bibliographie

Bibliographie et Webographie

[1] K. Hafner et M. Lyon. Les sorciers du Net. Calmann-Lévy, 346p, 1999

[2] L. Shklar ET R. Rosen. *Web Application Architecture: Principles, Protocols and Practices*. John Wiley & Sons Ltd, 372p, 2003

[3] D. Cederholm. *Bonnes pratiques des standards du web*. Pearson Education, 300p, 2010

[4] Premières applications Web 2.0 avec Ajax et PHP, *Eyrolles, 445p, 2008*

[5] J. Governor, D. Hinchcliffe ET D. Nickull. ***Web 2.0 Architectures***. O'Reilly Media, 248p, 2009

[6] Dictionnaire du Web (en ligne ; 2018)
https://www.1min30.com/dictionnaire-du-web/gestionnaire-de-contenu-cms

[7] Laurent Bloch & Christophe Wfhugel, "Sécurité informatique : Principes et méthode" édition 2, P : 4- 32, 2009.

[8] Gnu/Linux Fedora - Sécurité du Système, des Données, Pare-Feu, Chiffrement, Authentification ... Franck huet, christian verhille

[9] Introduction à l'insécurité informatique Dr. Pierre BARTHELEMY

[10] Les menaces contre la sécurité informatique agence Anti-Cybercriminalité (ACC) (en ligne ; 2018)
http://www.anti-cybercriminalite.fr/article/les-menaces-contre-la-securite-informatique

[11]Conseils pratiques (en ligne ; 2018)
http://www.interieur.gouv.fr/A-votre-service/Ma-securite/Conseils-pratiques/Sur-internet/

[12] Les différentes catégories de hackersagence Anti-Cybercriminalité (ACC) (en ligne ; 2018)
http://www.anti-cybercriminalite.fr/article/les-diff%C3%A9rentes-formes-de-cybercrimes-existantes

[13] Sécurité informatique - Ethical Hacking

ACISSI - Marion AGÉ - Sébastien BAUDRU - Nicolas CROCFER - Robert CROCFER - Franck EBEL - Jérôme HENNECART - Sébastien LASSON - David PUCHE - Raphaël RAULT

[14] Linux : Sécuriser un réseau Éditions
Eyrolles Bernard Boutherin, Benoît Delaunay.

[15] Sécurité Informatique
Patrick Ducrot

[16] Les aspects juridiques et judiciaires liés à la Cybercriminalité en Algérie Abdelkrim Djadi. *http://slideplayer.fr/slide/1136438/*

[17] Introduction aux attaques (en ligne ; 2018)
https://www.commentcamarche.com/contents/47-piratage-et-attaques-informatiques#introduction-aux-attaques

[18] Definition of a Security Vulnerability (en ligne ; 2018)
https://technet.microsoft.com/fr-fr/library/cc751383%28en-us%29.aspx

[19] Rapport IBM ISS X-Force sur les tendances et risques pour 2008 (en ligne ; 2018)

https://vdocuments.mx/xforce-2008-annual-report.html

[20] Failles de sécurité des applications Web Principe, parades et bonnes pratiques de développement Guillaume HARRY

[21] W.G.J.Halfond, J.Viegas, A.Orso, "A Classification of SQL Injection Attacks and Countermeasures", Proc. of the International Symposium on Secure Software Engineering, 2006.

[22] OWASP Top 10 2013 Les Dix Risques de Sécurité Applicatifs Web les Plus Critiques

[23] Analyse de vulnérabilités et évaluation de systèmes de détection d'intrusions pour les applications Web. Rim AKROUT université de Toulouse.

[24] D. Gollmann. Securing Web applications. Dans Information Security Technical Report, chapitre 1-9, Elsevier, 2008.

[25] **Guia Brahim Fouad.** Conception et Réalisation d'un système d'audit de sécurité des applications web. « Web Application Security Audit ». 3IL école d'ingénieurs. 2014.

[26] Dafydd Stuttard & Marcus Pinto. The Web Application Hackers Handbook 2nd edition.

Glossaire

Glossaire

-A-

ARPA

Advanced Research Project Agency

-C-

CERN

Centre Européen de Recherche Nucléaire

CMS

Content Management System, cela désigne une famille d'applications qui ont pour but de créer et mettre à jours facilement un site web dynamique. Le nom de ces logiciels se traduit en français par "*Système de Gestion de Contenu*"

CSS

Cascading Style Sheets ou "feuilles de style en cascade" en français. Le CSS est un langage informatique utilisé pour mettre en forme les fichiers HTML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML

-D-

DARPA

Defence Advanced Research Project Agency

DDoS

Distributed Denial of Service ou déni de service distribué, Une attaque DDoS vise à rendre un serveur, un service ou une infrastructure indisponibles en surchargeant la bande passante du serveur, ou en accaparant ses ressources jusqu'à épuisement.

Lors d'une attaque DDoS, une multitude de requêtes sont envoyées simultanément, depuis de multiples points du Net. L'intensité de ce "tir croisé" rend le service instable, ou pire, indisponible.

DIG

Domain Information Groper, est un programme informatique de débogage de serveurs DNS. Il signifie, littéralement *Chercheur d'Information sur les Domaines*.

DNS

Domain Name System" (système de noms de domaine) ou alors "*Domain Name Server*". D'une manière générale les DNS permettent de faire la correspondance entre un nom de domaine et une adresse IP

-E-

En tête HTTP

Les entêtes HTTP sont un ensemble de lignes envoyés par un serveur au navigateur web lors d'une requête HTTP. Ces lignes donnent des informations telles que la version du protocole utilisé, l'interpréteur, le service qui tourne au niveau du serveur web, et des informations complémentaires.

-F-

Faux Positif

C'est quand il y a un message d'erreur/vulnérabilité mais l'erreur/vulnérabilité indiqué par ce message est fausse (fausse alerte).

-G-

GHDB

Google Hack DataBase est une base de données avec près de 8.000 entrées. Elle permet aux administrateurs de vérifier leur site pour les vulnérabilités basées sur des données

stockées dans Google. Avec cet outil, vous pouvez savoir si votre site est indexé dans les vulnérabilités sur Google.

-H-

HTTP

HyperText Transfer Protocol, est le protocole client/serveur utilisé pour accéder aux informations situées sur le Web au moyen d'un navigateur, tel qu'Internet Explorer, C'est le protocole de transfert sur internet le plus courant.

-J-

Javascript

JavaScript est un langage informatique utilisé sur les pages web. Ce langage à la particularité de s'activer sur le poste client, en d'autres mots c'est votre ordinateur qui va recevoir le code et qui devra l'exécuter. C'est en opposition à d'autres langages qui sont activé côté serveur. L'exécution du code est effectuée par votre navigateur internet tel que Firefox ou Internet Explorer.

jQuery

jQuery est une bibliothèque JavaScript libre et multi-plateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web.

-L-

Ubuntu x.x **LTS**

Long Term Suport qui veut dire que pour cet OS le support et maintenance est assuré pour une période de temps plus longue que la normale.

-M-

Mime

Multipurpose Internet Mail Extensions, est un standard internet qui étend le format de données des courriels pour supporter des textes en différents codage de caractères autre que l'ASCII, des contenus non textuels, des contenus multiples, et des informations d'en tête en d'autres codage que l'ASCII

-N-

POO

Programmation Orienté Objet

-S-

Script

Liste de commandes écrites dans un certains langage destinée à être interprétée pour effectuer une certaine tache.

-T-

TCP /IP

Transmission Control Protocol/ Internet Protocol, La suite TCP/IP est l'ensemble des protocoles utilisés pour le transfert des données sur Internet

Trigger

Déclencheur, son déclenchement est en relation avec un évènement.

-U-

URL

Uniform Resource Locator est couramment appelé adresse web. Cette adresse sert à désigner une ressource présente sur le web par une suite de caractère ASCII. Les ressources peuvent être variées (page web, vidéo, son, image, animation, adresse email ...). De manière concrète, l'URL de cette page est indiqué dans le navigateur web.

User Agent

User Agent est une chaîne de caractères envoyé par le navigateur au serveur lorsqu'un internaute visite une page web. Il est aussi envoyé lorsqu'un robot ou un logiciel quelconque visite une page web, Ce petit bout de texte est envoyé dans le header HTTP dans l'entête "User-Agent"

-W-

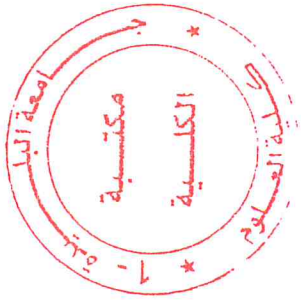
W3C

World Wide Web Consortium, organisme de standardisation à but non-lucratif, il n'émet pas de norme au sens européen, il est chargé de promouvoir la compatibilité des technologies du WWW, telles que HTML, XHTML, XML ..

-X-

Xhtml

Extensible HyperText Markup Language, est un langage de balisage servant à écrire des pages pour le World Wide Web. Conçu à l'origine comme le successeur de HTML

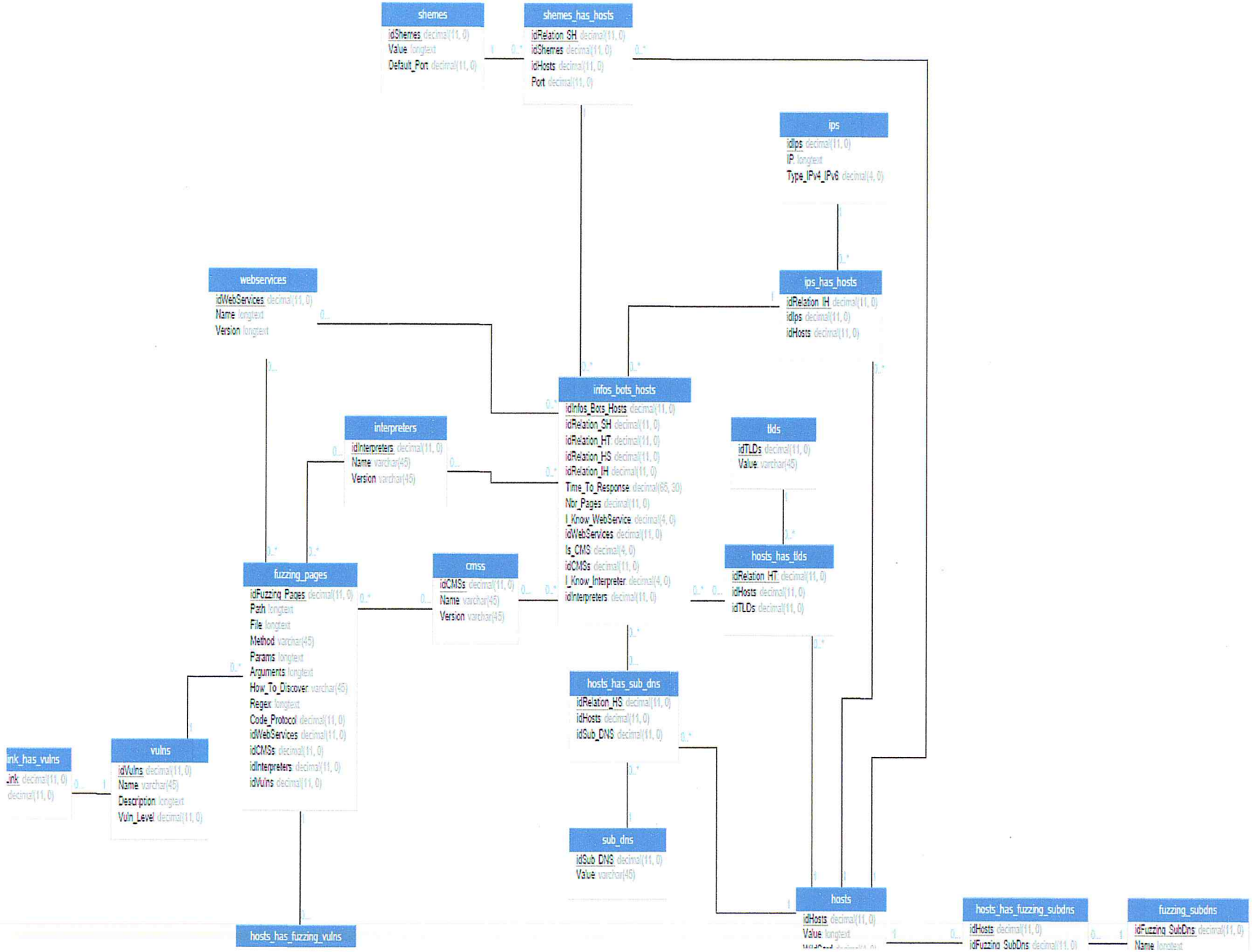


Annexe

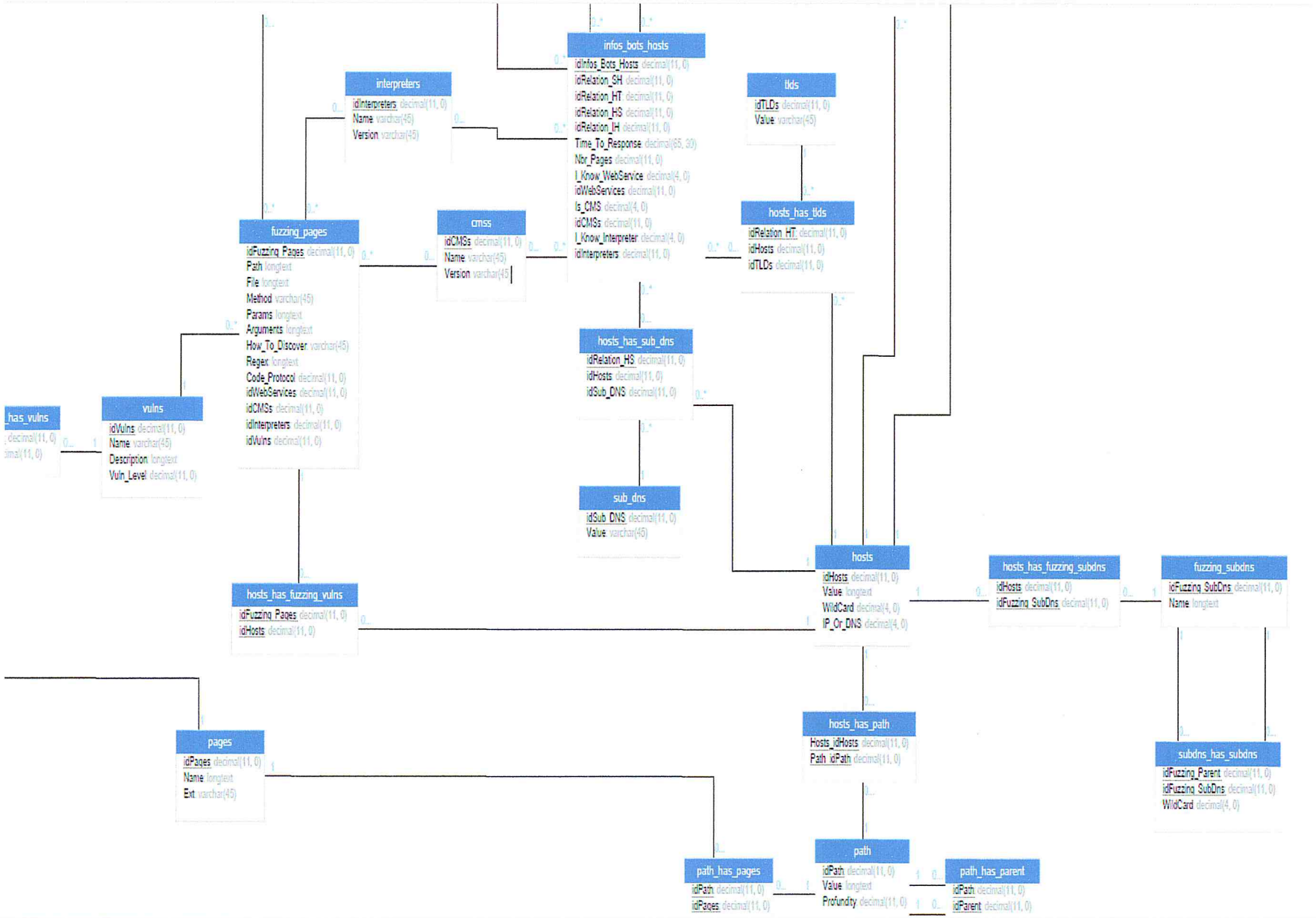
Annexes

Annexe 1 : Le schéma relationnel de la base de données normalisé, le schéma est trop volumineux et donc découpé en 3 parties.

Partie (gauche) du schéma



Partie (droite-haut) du schéma



Partie (droite-bas) du schéma

Annexe 2 : Project Management, dans cette annexe on va exposer comment on a géré notre projet en utilisant le diagramme de Gantt.

Liste des tâches (nom, durée, date début, date fin, prédécesseur) :

DIAGRAMME DE GANTT

	Mode Tâche	Task Name	Durée	Début	Fin	Prédécesseurs
1		**Developpement de l'application**	**104 jours**	**Mer 17/01/18**	**Lun 11/06/18**	
2		**Analyse préliminaire**	**7 jours**	**Mer 17/01/18**	**Jeu 25/01/18**	
3		Determiner le perimetre du stage	6 jours	Mer 17/01/18	Mer 24/01/18	
4		Connaissance de l'existant	1 jour	Jeu 25/01/18	Jeu 25/01/18	3
5		**Analyse des besoins**	**20 jours**	**Ven 26/01/18**	**Jeu 22/02/18**	**2**
6		Analyser les besoins	1 jour	Ven 26/01/18	Ven 26/01/18	2
7		Avant-projet et spécifications du logiciel	1 jour	Lun 29/01/18	Lun 29/01/18	6
8		Spécifications du logiciel et examen budgétaire avec l'équipe	2 jours	Mar 30/01/18	Mer 31/01/18	7
9		Incorporer les commentaires sur les spécifications de l'application	6 jours	Jeu 01/02/18	Jeu 08/02/18	8
10		Elaboré le calandrier de livraison	1 jour	Ven 09/02/18	Ven 09/02/18	9
11		Obtenir l'approbation de procéder (concept, calendrier, budget)	2 jours	Lun 12/02/18	Mar 13/02/18	10
12		Obtenir les ressources nécessaires	6 jours	Mer 14/02/18	Mer 21/02/18	11
13		Cloturé l'analyse	1 jour	Jeu 22/02/18	Jeu 22/02/18	12
14		**Conception**	**28 jours**	**Ven 23/02/18**	**Mar 03/04/18**	**13**
15		Élaborer des spécifications fonctionnelles	7 jours	Ven 23/02/18	Lun 05/03/18	13
16		Conception des organigrammes	7 jours	Mar 06/03/18	Mer 14/03/18	15

DIAGRAMME DE GANTT

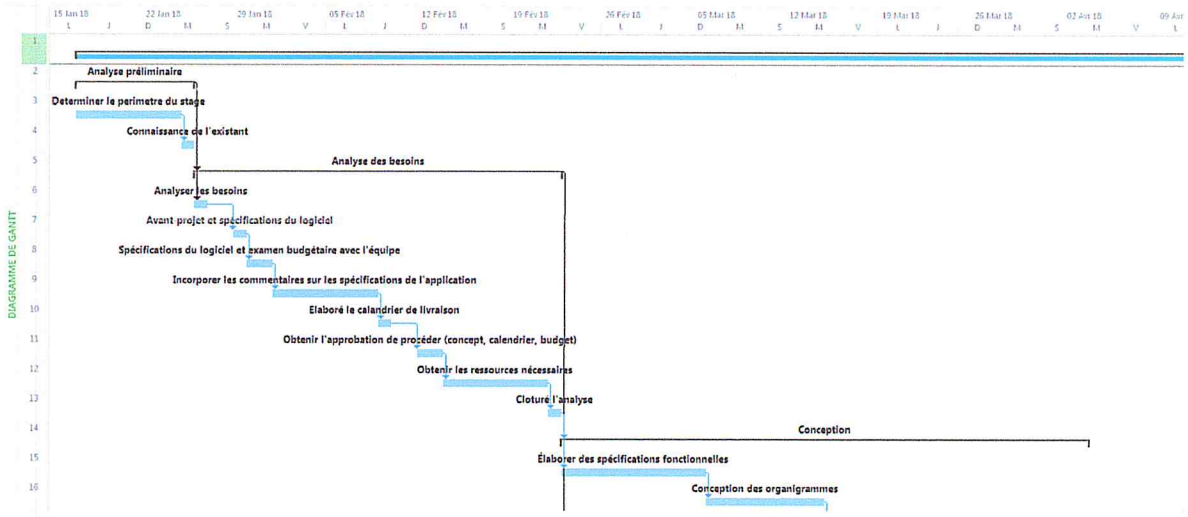
	Mode Tâche	Task Name	Durée	Début	Fin	Prédécesseurs
16		Conception des organigrammes	7 jours	Mar 06/03/18	Mer 14/03/18	15
17		Conception des diagrammes UML	5 jours	Jeu 15/03/18	Mer 21/03/18	16
18		Concevoir un prototype basé sur les spécifications fonctionnelles	2 jours	Jeu 22/03/18	Ven 23/03/18	17
19		Revoir les spécifications fonctionnelles	1 jour	Lun 26/03/18	Lun 26/03/18	18
20		Incorporer les commentaires et spécifications fonctionnelles	4 jours	Mar 27/03/18	Ven 30/03/18	19
21		Obtenir l'approbation de procéder	1 jour	Lun 02/04/18	Lun 02/04/18	20
22		Conception complète	1 jour	Mar 03/04/18	Mar 03/04/18	21
23		**Développement**	**20 jours**	**Mer 04/04/18**	**Mar 01/05/18**	**22**
24		Revoir les spécifications fonctionnelles	2 jours	Mer 04/04/18	Jeu 05/04/18	22
25		Identifier les paramètres de conception modulaire à plusieurs niveaux	2 jours	Ven 06/04/18	Lun 09/04/18	24
26		Identifier les paramètres de conception anti-pattern et multi-processing	2 jours	Mar 10/04/18	Mer 11/04/18	25
27		Refonte de la BDD	1 jour	Jeu 12/04/18	Jeu 12/04/18	26
28		Realisation et coding du Robot 1	3 jours	Ven 13/04/18	Mar 17/04/18	27
29		Realisation et coding du Robot 2	3 jours	Mer 18/04/18	Ven 20/04/18	28
30		Realisation et coding du Robot 4	3 jours	Lun 23/04/18	Mer 25/04/18	29
31		Realisation et coding du Robot 3	3 jours	Jeu 26/04/18	Lun 30/04/18	30

DIAGRAMME DE GANTT

	Mode Tâche	Task Name	Durée	Début	Fin	Prédécesseurs
31		Realisation et coding du Robot 3	3 jours	Jeu 26/04/18	Lun 30/04/18	30
32		Realisation et coding de l'interface web	1 jour	Mar 01/05/18	Mar 01/05/18	31
33		**Teste et debugs**	**8 jours**	**Mer 02/05/18**	**Ven 11/05/18**	**32;28;29;31;30**
34		testes	2 jours	Mer 02/05/18	**Jeu 03/05/18**	**32;28;29;31;30**
35		debugs	2 jours	Ven 04/05/18	Lun 07/05/18	34
36		modifier le code	2 jours	Mar 08/05/18	Mer 09/05/18	35
37		re-tester les modules	1 jour	Jeu 10/05/18	Jeu 10/05/18	36
38		validation	1 jour	Ven 11/05/18	Ven 11/05/18	37
39		**Déploiement**	**4 jours**	**Lun 14/05/18**	**Jeu 17/05/18**	**33**
40		Déterminer la stratégie de déploiement final	1 jour	Lun 14/05/18	Lun 14/05/18	33
41		Élaborer une méthodologie de déploiement	1 jour	Mar 15/05/18	Mar 15/05/18	40
42		Complète de déploiement	2 jours	Mer 16/05/18	Jeu 17/05/18	41
43		Documentation & Redaction rapport	70 jours	Ven 23/02/18	Jeu 31/05/18	5

Maintenant on va exposer le diagramme de Gantt :

1er partie :



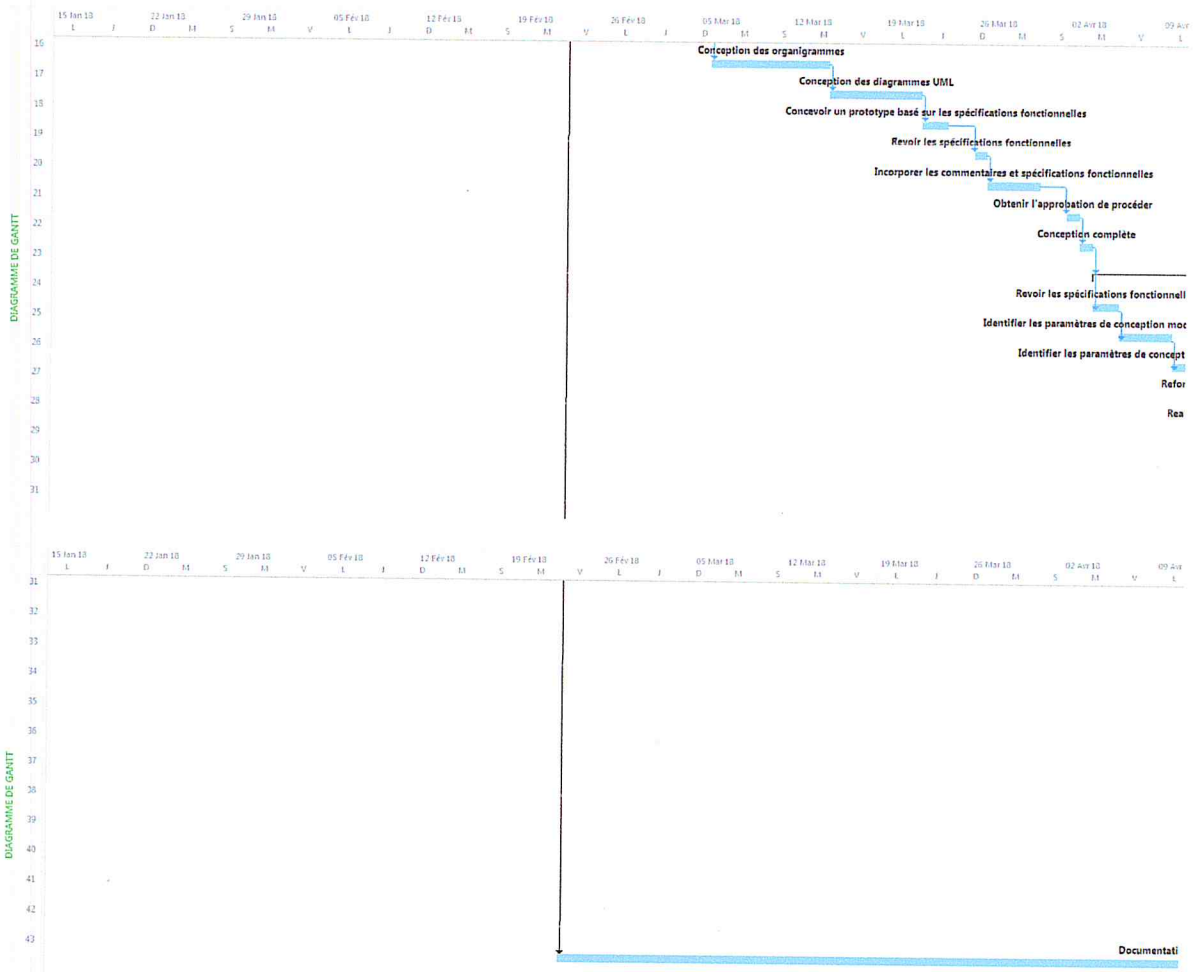


DIAGRAMME DE GANTT
Conception des organigrammes
Conception des diagrammes UML
Concevoir un prototype basé sur les spécifications fonctionnelles
Revoir les spécifications fonctionnelles
Incorporer les commentaires et spécifications fonctionnelles
Obtenir l'approbation de procéder
Conception complète
Revoir les spécifications fonctionnell
Identifier les paramètres de conception moc
Identifier les paramètres de concept
Refor
Rea
DIAGRAMME DE GANTT
Documentati

2ème partie :


DIAGRAMME DE GANTT