



RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE
RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88
<http://www.ifri-uac.net> Courriel : contact@ifri.uac.bj



MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Sécurité Informatique

Présenté par :

Jospy GOUDALO

Prototype d'un système de paiement électronique des factures d'électricité de la SBEE dans un environnement sécurisé.

Sous la supervision :

Professeur Eugène C. EZIN

Enseignant - chercheur en Informatique

Université d'Abomey-Calavi

Année Académique : 2017-2018

Table des matières

Sigles et Abréviations	iii
Dédicace	v
Remerciements	vi
Résumé/Abstract	vii
Introduction	1
1 Contexte et justification	1
2 Problématique	1
3 Objectifs	2
4 Environnement de stage	2
5 Organisation du mémoire	3
1 Revue de littérature	4
1.1 Gd'Or	4
1.1.1 Présentation	4
1.1.2 Quelques choix techniques concernant Gd'Or	5
1.1.3 Limites	6
1.1.4 Participation de Gd'Or dans notre travail	6
1.2 Sécurité des applications Web	7
1.2.1 Notion de hacker et de cracker	7
1.2.2 Mode opératoire d'une attaque informatique	7
1.2.3 Menaces et risques applicatifs	8
2 Conception et Matériels	15
2.1 Conception	15
2.1.1 Méthode de modélisation	15
2.1.2 Diagramme de cas d'utilisation	15
2.1.3 Diagramme de séquence	16
2.2 Matériels	17
2.2.1 Principe de fonctionnement de notre solution	17
2.2.2 Contrôle des accès : RBAC	18
2.2.3 Solutions de paiement	19
2.2.4 Langages de développement et outils	20
2.2.5 Environnement de Production	22
3 Résultats et Discussion	28

3.1	Présentation des résultats des tests	28
3.1.1	Interfaces Abonné	29
3.1.2	Interfaces Administrateurs	32
3.2	Discussion	33
	Conclusion et Perspectives	34
	Bibliographie	35
	Webographie	36

Table des figures

1.1	Représentation cyclique de la méthodologie ZEH.	8
2.1	Diagramme de cas d'utilisation	16
2.2	Diagramme de séquence	16
2.3	Illustration du principe de fonctionnement	18
2.4	Représentation Utilisateurs - Rôles - Permissions	19
2.5	Architecture MVT Django	21
2.6	Topologie du réseau	23
2.7	Pourcentages de sites Web utilisant différents serveurs Web [14]	24
2.8	Architecture de NGINX[11]	25
3.1	Page d'inscription de l'application	28
3.2	Page de connexion	29
3.3	Liste des factures de l'abonné	29
3.4	Vue d'une facture (Format d'impression)	30
3.5	Vue d'une facture (Liste d'informations)	30
3.6	Liste des factures choisies	31
3.7	Le compte d'un utilisateur	31
3.8	Utilisateurs de la plateforme	32
3.9	Fichiers importés	32

Sigles et Abréviations

ACID : Atomicité, Cohérence, Isolation et Durabilité: propriétés ACID [25](#)

API : Application Programming Interface [9](#)

CORS : Cross-origin resource sharing [11](#)

CSRF : Cross-Site Request Forgery [20](#)

DSI : Direction des Systèmes d'Information [2](#)

FTP : File Tranfert Protocol [25](#)

GDPR : General Data Protection Regulation/ Union Européenne [10](#)

HIDS : Host Intrusion Detection System [25](#)

HTML : HyperText Markup Language [12](#)

HTTP : Hypertext Transfer Protocol [11](#)

HTTPS : HyperText Transfer Protocol Secure [20](#)

IDS : Intrusion Detection System [25](#)

IPS : Intrusion Prevention System [25](#)

JSON : JavaScript Object Notation [11](#)

LTS : Long-Term Support [21](#)

MVC : Model View Controller [20](#)

MVT : Model View Template [20](#)

NAT : Network Address Translation [26](#)

NIDS : Network Intrusion Detection System [25](#)

NoSQL : Non Structured Query Language [9](#)

ORM :	Object Relational Mapping 9
OWASP :	Open Web Application Security Project 8
PCI-DSS :	Payment Card Industry Data Security Standard 33
SBEE :	Société Béninoise d’Energie Electrique 1
SFTP :	SSH File Transfer Protocol 16
SMS :	Short Message Service 2
SONEB :	Société Nationale des Eaux du Bénin 4
SQL :	Structured Query Language 9
SSL :	Secure Sockets Layer 20
URI :	Uniform Resource Identifier 10
URL :	Uniform Resource Locator 12
VPN :	Virtual Private Network 6
XML :	Extensible Markup Language 10
XSS :	Cross-Site Scripting 11

Dédicace

A

mon Père **Noll GOUDALO**

ma Mère **Nadine SODEDJI**

et mon Frère **Brice GOUDALO**

Remerciements

Nous exprimons notre vive gratitude aux personnes physiques et institutions qui ont contribué à rendre meilleur notre travail, notamment :

- M. Eugène C. EZIN, notre Maître de mémoire et Directeur de l’Institut de Formation et de Recherche en Informatique (IFRI) ;
- M. Armand ACCROMBESSI, Développeur et enseignant à l’IFRI ;
- Tous les enseignants de l’IFRI pour nous avoir donné les enseignements nécessaires tout au long de notre formation ;
- M. Camille METINHOUE, Directeur des Systèmes d’Information de la SBEE ;
- M. Idriss SANTANNA et M. Arron ALLADAYE, mes encadreurs de stage respectivement Chef Service Système, Sécurité et Maintenance des Infrastructures et Chef Service Réseaux Télécommunications et TIC à la Direction des Systèmes d’Information de la SBEE ;
- Tout ceux du personnel de la SBEE qui nous ont apportés leur soutien tout au long de notre stage ;

Que tous ceux qui nous ont aidé, de près ou de loin, trouvent ici l’expression de nos sentiments les meilleurs.

Résumé

L'accès à Internet devient de plus en plus croissant. Cela peut s'expliquer par les nombreux avantages qu'offre le digital et précisément Internet. Parmi ces nombreux avantages, il permet aux entreprises d'offrir des services aux clients depuis n'importe quel endroit. C'est grâce à cet avantage que nous avons voulu mettre à la disposition de la population béninoise, une application Web leur permettant de consulter et de payer leurs factures grâce à un navigateur web. Cela leur permettrait essentiellement d'éviter les pertes de temps dues aux longues queues aux guichets de la SBEE et d'avoir un accès plus facile à leurs factures.

Bien qu'Internet dispose de beaucoup d'avantages, il n'est pas pour autant sans danger. La cybercriminalité prend de l'ampleur et les tentatives d'intrusion dans les systèmes informatiques croissent de façon exponentielle. Au regard de ces faits, il faut que des mesures de sécurité soient prises afin d'éviter au maximum des intrusions au niveau de notre application et des différents serveurs. Nous proposons ainsi une architecture sécurisée pour la mise en production de notre application qui sera gérée en interne (déployée au sein de la SBEE).

Mots clés : Sécurité Informatique, Paiement en ligne, Factures, SBEE

Abstract

Internet access is becoming increasingly popular. This can be explained by all the advantages offered by digital and precisely Internet. Among these advantages, it allows companies to offer services to customers whatever they are. Based upon this advantage, we wanted to make available to the Beninese people, a web application allowing them to view and pay their bills from a web browser. This would essentially prevent the lost of time due to the long queues at the counter and to have easier access to their invoices.

Taking into account the security aspect, despite all benifits from Internet, some drawbacks are known like cybercriminality, intrusion attempts into computer systems. Cybercrime is on the rise and intrusion attempts into computer systems is growing exponentially. With regard to these, several measures must be taken in order to avoid as much as possible intrusions at the level of our application and the various servers. We therefore propose a secured architecture customized for the production of our application, which will be managed internally (i.e. deployed within SBEE).

Key words: IT Security, Online Payment, Invoices, SBEE

Introduction

L'informatique, science du traitement rationnel et automatisé de l'information, est devenue aujourd'hui un outil indispensable pour l'évolution de toute entreprise. Elle est passée d'un statut luxueux à un statut nécessaire. La preuve : toute structure de renom investit un budget non négligeable pour l'informatisation de ses opérations. Cependant, certaines entreprises ne sont toujours pas en phase avec l'évolution de la technologie en rapport avec les services qu'ils offrent et la sécurité de ces derniers. Dans le même temps, des individus profitent de ce phénomène pour piéger d'autres ignorant le fonctionnement de l'ordinateur dans le but de voler ou altérer des informations précieuses et confidentielles pour des fins diverses : c'est la cybercriminalité. Vu l'ampleur que prend ces différents crimes, il devient nécessaire de voir autrement l'importance de la sécurité dans tous les systèmes d'information.

1 Contexte et justification

Le Bénin s'inscrit depuis près de deux ans dans une politique de restructuration du secteur numérique pour y insuffler une nouvelle dynamique. Plusieurs projets et programmes sont donc nés de cette volonté et constituent une feuille de route pour les acteurs au cœur de cette restructuration.

Dans l'optique de contribuer au développement de ce secteur et surtout de faciliter la vie à la population béninoise, nous avons pensé à mettre en place une plateforme permettant le paiement en ligne des factures d'électricité de la SBEE. Cependant, il s'agit d'un système assez complexe lorsque nous considérons la quantité et la criticité du flux d'informations que nous aurons à traiter. La sécurité de l'information revêtant une importance capitale pour la survie d'une entreprise, il faudra aussi mettre en oeuvre un système de défense face aux menaces pour réduire l'impact des attaques et essayer, au maximum, d'éliminer les risques.

2 Problématique

Il n'est plus à démontrer que l'électricité est à la base de tout développement. De la disponibilité de l'énergie dépend la satisfaction de tous les besoins humains fondamentaux : l'eau, l'alimentation, la santé, l'éducation. L'option la plus accessible est de souscrire à un abonnement post-payé avec la SBEE. Néanmoins il n'est pas rare de constater qu'après de longues heures d'attente aux guichets de la SBEE, l'on vous dise : "Désolé monsieur nous avons un problème de connexion. Veuillez repasser demain.". Il faudra donc repasser plus tard pour solder la facture alors que nous n'avons pas forcément assez de temps pour cela.

Nous sommes aussi parfois confrontés à l'oubli des factures non payées. Cependant quand vous n'êtes pas à jour après un délai d'environ un mois, un agent peut passer à tout moment couper le courant et vous allez devoir payer des pénalités.

Notre travail consistera donc à mettre en place une application web permettant le paiement des factures depuis

un téléphone portable ou un ordinateur connecté à Internet. Cependant les données gérées par cette application sont sensibles et critiques. A titre d'exemple les informations de votre carte VISA¹ pourraient être volées. La carte sera donc utilisée à votre insu et votre argent ne vous appartiendra plus. Il sera donc nécessaire de prendre en compte toutes les menaces possibles et d'y apporter des solutions efficaces. Ainsi nous pourrions avoir une application sécurisée permettant le paiement des factures électriques depuis un appareil connecté à Internet.

3 Objectifs

Notre projet a pour objectif principal de faciliter le paiement des factures électriques de la SBEE et de mettre en place les garde-fous nécessaires pour réduire les attaques informatiques.

Plus précisément, il s'agira de :

- rappeler aux personnes utilisant la plateforme qu'ils ont des impayés (via des SMS et/ou des emails);
- assurer la disponibilité complète de la plateforme afin que les consultations et paiements puissent se faire à n'importe quel moment;
- garantir la sécurité de toutes les transactions financières effectuées via la plateforme;
- construire un historique afin d'avoir une trace des factures ainsi que des paiements.

4 Environnement de stage

Ce travail a été réalisé dans les locaux de la Direction des Systèmes d'Information (DSI) de la Société Béninoise d'Energie Electrique (SBEE) - Direction Générale. La SBEE a pour mission de produire, de transporter et de distribuer l'énergie électrique sur l'ensemble du territoire national. Mais elle produit aussi de l'énergie électrique pour combler le déficit énergétique en cas de besoin. Elle a l'obligation de satisfaire les exigences de sa clientèle qui sont :

- la bonne qualité de l'énergie distribuée ;
- la disponibilité de l'énergie ;
- l'acquisition de l'énergie à moindre coût.

Ces trois exigences constituent à tout moment des défis à relever pour la société. Pour améliorer la qualité de ses prestations, la SBEE nourrit un certain nombre d'ambitions :

- disposer d'un réseau stable ;
- renforcer sa capacité de production d'énergie électrique ;
- protéger et assurer efficacement la maintenance de ses installations.

Elle a son siège social à Cotonou - Ganhi et couvre le territoire national à travers huit (8) Directions Régionales et quarante-deux (42) agences géographiquement réparties dans tous les départements du pays.

¹Carte de paiement émise par un établissement bancaire

5 Organisation du mémoire

Le présent travail se présente en trois (03) chapitres. Le premier concerne la revue de littérature sur le système existant (actuel) permettant le paiement des factures et quelques notions par rapport à la sécurité des applications Web. Le deuxième chapitre aborde les choix organisationnels (procédures) et techniques opérés en vue de la conception et de la réalisation des solutions proposées. Le troisième chapitre, quant à lui, fait une analyse critique des résultats issus de nos tests après les avoir exposés.

Revue de littérature

Introduction

Pour bien réaliser un projet, un état des lieux permettant de faire un point sur le sujet du projet est requis. Dans un premier temps, nous présenterons Gd'Or le logiciel métier utilisé à la SBEE puis nous parlerons de la sécurité des applications Web dans un second temps.

1.1 Gd'Or

1.1.1 Présentation

Gd'Or est le logiciel métier utilisé par la SBEE. Il couvre la plupart des besoins des besoins de l'entreprise et intègre plusieurs modules dont :

- la Gestion Clientèle Electricité et/ou Eau ;
- la Comptabilité Générale, Analytique, Budgétaire ;
- la Paie et la Gestion des Ressources Humaines ;
- la Gestion des Stocks ;
- la Gestion des Approvisionnements ;
- la Monétique ;
- les Immobilisations ;

G d'Or a été conçu pour faciliter l'informatisation des sociétés de distribution d'Eau et d'Electricité. Il est utilisé par plusieurs entreprises des pays de l'Afrique de l'ouest dont le :

- **Tchad** : Société Nationale d'Electricité (SNE) et Société Tchadienne des Eaux (STE) ;
- **Burkina-Faso** : Office National de l'Eau et de l'Assainissement (ONEA) ;
- **Niger** : Société Nigérienne d'Electricité (NIGELEC) et Société d'Exploitation des Eaux du Niger (SEEN) ;
- **Togo** : Société Togolaise des Eaux (TDE) et Compagnie Energie Electrique du Togo (CEET) ;
- **Bénin** : Société Nationale des Eaux du Bénin ([SONEB](#)) et Société Béninoise d'Energie Electrique (SBEE) ;

1.1.2 Quelques choix techniques concernant Gd'Or

1.1.2.1 Langage de développement

Gd'Or est un progiciel de Gestion intégré qui tourne sur une plateforme propriétaire [IBM](#) (International Business Machines Corporation) appelée AS/400 (que l'on devrait aujourd'hui appeler system i). Les Power Systems de IBM sont des serveurs d'applications très robustes et supportant plusieurs OS (Windows, Linux et IBM i). Dans le cadre spécifique de Gd'Or l'OS est IBMi V7r2 et le langage de programmation utilisé est le [RPG](#) (Report Programming Generator).

RPG ¹ est un langage de programmation de haut niveau pour les applications métier qui a été créé en tant que programme de création de rapports utilisé dans les systèmes d'exploitation de DEC et d'IBM. Il a ensuite évolué en un langage de programmation entièrement procédural. Sa dernière version, VisualAge RPG, est prise en charge par le principal système de mini-ordinateurs d'IBM, AS / 400.

Historiquement, RPG a probablement été le deuxième langage de programmation le plus utilisé, après [COBOL](#), pour les applications commerciales sur les ordinateurs de milieu de gamme. Le RPG est semblable au COBOL, mais est plus concis et soi-disant plus facile à utiliser pour les non-programmeurs [\[7\]](#).

Le langage inclut la possibilité de contrôler le moment où les entrées et les sorties ont lieu, une plus grande flexibilité dans le contrôle du formatage des rapports², la capacité de traitement par matrice, la simplification du codage et de nombreuses autres fonctionnalités pour des techniques de traitement plus sophistiquées.

1.1.2.2 Base de données

Pour le stockage et l'organisation des données, Gd'Or utilise le Système de Gestion de Base de Données (SGBD) **DB2** de IBM. DB2 est un SGBD relationnel qui selon IBM, est leader en termes de part de marché et de performances de base de données. Bien que les produits DB2 soient proposés pour les systèmes UNIX et les systèmes d'exploitation pour ordinateurs personnels, DB2 se démarque des produits de base de données Oracle dans les systèmes UNIX et de Microsoft Access dans les systèmes Windows.

DB2 est conçu pour stocker, analyser et extraire efficacement les données. Le produit DB2 est étendu à la prise en charge des fonctions orientées objet et des structures non relationnelles avec XML, JSON (Java Script Object Notation) et bien d'autres. Utilisé par des entreprises de toutes tailles, DB2 fournit une plateforme de données pour les opérations transactionnelles et analytiques. En assurant la disponibilité continue des données, il utilise le langage SQL tout comme Oracle, PostgreSQL ou MySQL. Il est déployé sur les Mainframes, systèmes UNIX, Windows, Mac/OS et Linux.

Outre ses offres pour les systèmes d'exploitation mainframe OS / 390 et ses systèmes AS / 400 de milieu de gamme, IBM propose des produits DB2 pour un spectre multiplateforme comprenant Linux, HP-UX, Sun Solaris, UNIX, Windows 2000, les systèmes antérieurs de Microsoft et bien d'autres. Utilisé par des entreprises de toutes tailles, DB2 fournit une plateforme de données pour les opérations transactionnelles et analytiques. En assurant la disponibilité continue des données, il utilise le langage SQL tout comme Oracle, PostgreSQL ou MySQL. Il est déployé sur les Mainframes, systèmes UNIX, Windows, Mac/OS et Linux. Les bases de données DB2 sont accessibles depuis n'importe quel programme d'application à l'aide de l'interface ODBC (Open Database Connectivity) de Microsoft, de l'interface JDBC (Java Database Connectivity) ou d'un courtier d'interface CORBA.

¹Ne pas confondre avec Role-Playing Game

²Les factures électriques sont par exemple des rapports

1.1.2.3 Sécurité par rapport à Gd'Or

Gd'Or intègre une sécurité basée sur la gestion des types d'utilisateurs que nous appellerons profils. La notion de profil est un pilier dans le fonctionnement de Gd'Or. Ce sont les profils qui accréditent les utilisateurs de certains droits. Ils permettent aussi de sécuriser et d'isoler les données à certains utilisateurs. Un profil peut être associé à un utilisateur et/ou à une entité. Pour les cas où des utilisateurs ont besoin d'une ressource, certains profils ont la possibilité d'accorder l'accès à la ressource juste pour que l'action soit exécutée. Les profils sont bien sûr adaptés aux rôles et aux permissions de chaque utilisateur sur la plateforme. Ils peuvent tous être réunis en trois catégories dont : les administrateurs, les opérateurs, et les utilisateurs (simples). Cette mise en place permet d'assurer la confidentialité des informations et elle permet à chaque utilisateur du système de se concentrer sur son travail.

L'architecture utilisée actuellement est une architecture centralisée. Le serveur central est situé à la Direction Générale. Il héberge le logiciel métier Gd'Or. Les clients sont tout ceux qui utilisent Gd'Or. Chaque action représente une ou plusieurs requêtes envoyées vers le serveur central. Les agences sont connectées à la Direction Générale via des liaisons radios faites avec des équipements de marque Ubiquiti. Les informations transitent via des fréquences libres et sont chiffrées avec le chiffrement **Triple Data Encryption Algorithm (Triple DES ou 3DES)** par les routeurs. Il s'agit d'un chiffrement par bloc à clé symétrique, ce qui signifie que la même clé est utilisée pour chiffrer et déchiffrer des données dans des groupes de bits de longueur fixe, appelés blocs. Il s'appelle "Triple DES" car il applique le chiffrement DES trois fois lors du chiffrement des données. Cette architecture isole le réseau et limite une grande partie des attaques. Pour avoir accès aux données il faudrait soit avoir accès aux coordonnées [VPN](#), soit être directement dans le réseau.

1.1.3 Limites

Quoique robuste et assez performant, l'ergonomie de Gd'Or est assez basique du fait des choix faits par l'éditeur (Groupe Cauris Informatique) de rester en mode texte (écran vert/noir, menu interactifs, etc.). Il fonctionne dans une console et les entrées sont prises ligne par ligne. Gd'Or est en exploitation depuis plus de 15 ans et perd de son attractivité par rapport aux autres [ERP](#) disponibles sur le marché (Odoo ERP, SAP HANA, etc.). Bien qu'il soit toujours fonctionnel et en adéquation avec les besoins de l'entreprise, il serait quand même bien de penser à lui fournir une interface graphique plus complète afin de faciliter sa prise en main et son utilisation.

Il est bien vrai que l'architecture actuelle préserve la SBEE des attaques venant de l'extérieur. Cependant il existe beaucoup d'inconvénients et nous avons principalement les pertes de connexion. En réalité ces pertes de connexion dans les agences ne sont pas dues à une perte d'accès à Internet ou tout autre chose y ayant trait. Il s'agit plutôt de la surcharge de la fréquence sur laquelle émet l'agence vers Direction Générale (et vice-versa) via les PowerStation de Ubiquiti. Pour résoudre le problème, un agent situé au siège doit changer rapidement la fréquence en se plaçant sur une autre qui est plus libre. Mais cette est toujours temporaire et le processus doit être répété à chaque fois qu'il y a une perte de connexion.

1.1.4 Participation de Gd'Or dans notre travail

Gd'Or dispose du contenu. Il possède toutes les factures impayées comme payées. C'est Gd'Or qui fournit quotidiennement à notre application, les factures à payer. Il génère des fichiers *.csv* et les transmet à l'application. Pour éviter les redondances, il ne transmet que les nouvelles factures puisque les autres sont déjà présentes dans l'application. A chaque fin de journée, il reçoit un bilan des factures payées via l'application web. Ainsi il procède aux vérifications et à l'apurement des comptes.

1.2 Sécurité des applications Web

A l'ère de l'information, la cyber-sécurité est un défi omniprésent. Les attaques peuvent cibler n'importe quel site Internet pour un nombre croissant de raisons. Un manque de sécurité dans une application peut causer des dysfonctionnements, des arrêts complets de service ainsi que des pertes de données dommageables pour les utilisateurs et l'image de l'entreprise. Pour s'en prémunir, les entreprises doivent mettre en place des mesures de sécurité strictes dans le processus de développement de leur applications Web.

Cette section expose quelques failles de sécurité liées aux applications web et quelques bonnes pratiques pour y remédier.

1.2.1 Notion de hacker et de cracker

Un hacker est une personne qui, par jeu, goût du défi ou souci de notoriété, cherche à contourner les protections d'un logiciel, à s'introduire frauduleusement dans un système ou un réseau informatique [8]³.

Un cracker, quant à lui, s'introduit tout aussi frauduleusement dans un système informatique pour en entraver ou en fausser le fonctionnement. Son action est souvent plus dévastatrice.

1.2.2 Mode opératoire d'une attaque informatique

Une attaque informatique peut être décomposée en une suite d'étapes ou phases. Lorsqu'elles sont réunies, ces étapes forment une méthodologie complète pour mener à bien une attaque informatique. L'établissement d'une méthodologie permet de décomposer une procédure complexe en une suite de tâches gérables de taille plus réduite. Ainsi nous regroupons cette méthodologie en quatre étapes qui sont **la reconnaissance, les scans, l'exploitation, la post-exploitation et le maintien d'accès**⁴[2].

1.2.2.1 La reconnaissance

La reconnaissance, ou recueil d'informations, est probablement la plus importante des quatre phases. Plus le hacker passe du temps à collecter des informations sur sa cible, plus les phases suivantes auront une chance de réussir[2]. En effet la reconnaissance permet de connaître la cible dans les détails, de connaître les points forts et surtout les points faibles afin de notifier les prochaines possibilités d'attaque.

1.2.2.2 Les scans

Les scans sont des procédés ayant pour objectif d'identifier les systèmes actifs et les services qui existent sur les systèmes scannés. Dans ce cadre, le hacker prend le soin de vérifier l'activité d'un système, de trouver les portes ouvertes (les ports), de vérifier les processus tournant sur le système et d'aller à la recherche des vulnérabilités. Ce stade requiert une compréhension plus avancée des systèmes informatiques pour mieux comprendre les résultats recueillis⁵[2].

1.2.2.3 L'exploitation

En termes simples, l'exploitation consiste à obtenir un contrôle sur un système. Toutefois, il est à notifier que tout exploit ne conduit pas à la compromission intégrale d'un système. Un hacker peut se servir donc d'un exploit pour télécharger des contenus dont il ne détient pas la propriété pendant qu'un autre utilise un

³Recommandation officielle : fouineur.

⁴La post exploitation et le maintien d'accès forment une étape.

⁵Informations recueillies au cours du scan

exploit pour crypter les fichiers du système. L'utilisation de l'un des exploits ⁶ dépend donc de l'objectif visé par le hacker [4].

1.2.2.4 Post exploitation et maintien d'accès

Cette étape consiste à couvrir les traces de l'intrus agissant afin de ne pas se faire repérer [4]. Il permet aussi à ce dernier de faciliter ses prochains accès à la machine victime par l'installation de portes dérobées communément appelées "Backdoor". Ainsi, il n'aura plus besoin de reprendre toutes les étapes de son processus pour accéder à la machine dont il a eu à prendre le contrôle.

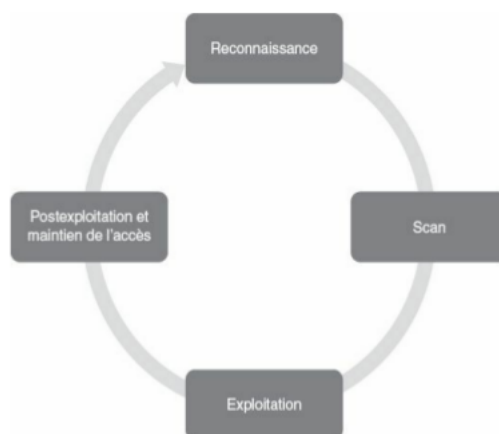


FIGURE 1.1 – Méthodologie ZEH, Patrick Engbreton, "Les bases du hacking", PEARSON 2013

La méthodologie d'attaque étant cernée, nous allons maintenant présenter les différentes failles auxquelles sont exposées les applications web.

1.2.3 Menaces et risques applicatifs

1.2.3.1 Failles de Sécurité

Le projet Open Web Application Security Project ([OWASP](#)) est une communauté ouverte destinée à permettre aux organisations de développer, d'acheter et de gérer des applications et des API fiables. Tous les outils, documents, vidéos, présentations et chapitres OWASP sont gratuits et ouverts à toute personne intéressée par l'amélioration de la sécurité des applications. L'approche de la sécurité des applications en tant que problème de personnes, de processus et de technologie est préconisée, car les approches les plus efficaces en matière de sécurité des applications nécessitent des améliorations dans ces domaines. OWASP produit aussi de nombreux types de matériaux de manière collaborative, transparente et ouverte. Le projet soutient la recherche innovante en matière de sécurité avec des subventions et des infrastructures.

L'OWASP Top 10 est un document de sensibilisation puissant pour la sécurité des applications Web. Il représente un large consensus sur les risques de sécurité les plus critiques pour les applications Web. Les membres du projet incluent une variété d'experts en sécurité du monde entier qui ont partagé leur expertise pour produire cette liste. Nous recommandons à toutes les entreprises d'adopter ce document de sensibilisation au sein de leur organisation et de commencer à faire en sorte que leurs applications Web minimisent ces risques. L'adoption du Top 10 de l'OWASP est peut-être la première étape la plus efficace pour changer la culture de développement de logiciels au sein de votre organisation en une culture qui produit du code sécurisé.

⁶Un exploit est le moyen par lequel un attaquant, ou un pentester en l'occurrence, profite d'un défaut dans un système, une application ou un service.

Cette mise à jour majeure (celle de 2017) ajoute plusieurs nouveaux problèmes, dont deux problèmes sélectionnés par la communauté - **A8 : 2017 - Désérialisation non sécurisée** et **A10 : 2017 - Insuffisance de Logs et de Surveillance**. Les deux principales différences par rapport aux versions précédentes du Top 10 d'OWASP sont les retours substantiels de la communauté et les données complètes rassemblées par des dizaines d'organisations, probablement la plus grande quantité de données jamais rassemblées dans la préparation d'une norme de sécurité applicative. Cela nous permet de croire que le nouveau Top 10 d'OWASP aborde les risques de sécurité applicative les plus importants auxquels sont confrontées les entreprises.

L'un des principaux objectifs du Top 10 d'OWASP est d'éduquer les développeurs, les concepteurs, les architectes, les gestionnaires et les organisations sur les conséquences des faiblesses les plus courantes et les plus importantes de la sécurité des applications Web. Le Top 10 fournit des techniques de base pour se protéger contre ces zones problématiques à haut risque et fournit des indications sur les endroits où aller.

Les attaquants peuvent potentiellement utiliser de nombreux chemins différents à travers votre application pour nuire à votre entreprise ou organisation. Chacun de ces chemins représente un risque qui peut ou non être suffisamment sérieux pour justifier une attention. Parfois, ces chemins sont triviaux à trouver et à exploiter, et parfois ils sont extrêmement difficiles. De même, le préjudice causé peut être sans conséquence, ou vous mettre à la faillite. Pour déterminer le risque pour votre organisation, vous pouvez évaluer la probabilité associée à chaque agent de menace, vecteur d'attaque et faiblesse de la sécurité et le combiner avec une estimation de l'impact technique et commercial sur votre organisation. Ensemble, ces facteurs déterminent votre risque global.

1.2.3.2 OWASP Top 10 - 2017

Le rapport « OWASP Top 10 » permet ainsi à l'équipe projet de se focaliser sur la protection de l'application Web face aux menaces les plus importantes, ce qui est moins coûteux et plus facilement réalisable que d'essayer de se protéger de tous les dangers. L'OWASP établit le classement 2017 ci-dessous, dont chacune des failles est développée dans les sous-sections suivantes :

i) Injection

Des erreurs d'injection, telles que l'injection [SQL](#), [NoSQL](#), se produisent lorsque des données non fiables sont envoyées à un interpréteur dans le cadre d'une commande ou d'une requête. Les données hostiles de l'attaquant peuvent amener l'interpréteur à exécuter des commandes inattendues ou à accéder aux données sans autorisation appropriée.

L'injection peut parfois conduire à une prise en charge complète de l'hôte. L'impact métier dépend des besoins de l'application et des données.

La prévention de l'injection nécessite de séparer les données des commandes et des requêtes. L'option préférée consiste à utiliser une [API](#) sécurisée, qui évite l'utilisation complète de l'interpréteur ou fournit une interface paramétrée, ou migre pour utiliser les outils [ORM](#) (Object Relational Mapping Tools)

ii) Authentification brisée

L'authentification brisée est généralement connue en anglais sous le nom de **Broken Authentication**. Les fonctions d'application liées à l'authentification et à la gestion de session sont souvent incorrectement implémentées, permettant aux pirates de compromettre les mots de passe, clés ou jetons de session ou d'exploiter d'autres failles d'implémentation pour prendre temporairement ou définitivement les

identités des autres utilisateurs. La prévalence de l'authentification brisée est généralisée en raison de la conception et de la mise en œuvre de la plupart des contrôles d'identité et d'accès. La gestion de session est le fondement des contrôles d'authentification et d'accès et est présente dans toutes les applications avec état.

Les attaquants doivent avoir accès à seulement quelques comptes ou à un seul compte administrateur pour compromettre le système. Selon le domaine de l'application, cela peut permettre le blanchiment d'argent, la fraude à la sécurité sociale et le vol d'identité, ou divulguer des informations hautement sensibles protégées par la loi.

Lorsque cela est possible, implémentez l'authentification multi-facteur pour éviter les attaques automatisées, le bourrage des informations d'identification, la force brute et la réutilisation des informations d'identification volées. Ne pas envoyer ou déployer avec des informations d'identification par défaut, en particulier pour les utilisateurs d'administration.

iii) Exposition de données sensibles

L'exposition de données sensibles est connue en anglais sous le nom **Sensitive Data Exposure**. De nombreuses applications Web et API ne protègent pas correctement les données sensibles, telles que les données financières, les soins de santé et les informations personnelles. Les attaquants peuvent voler ou modifier de telles données faiblement protégées pour effectuer une fraude par carte de crédit, un vol d'identité ou d'autres crimes. Les données sensibles peuvent être compromises sans protection supplémentaire, telles que le cryptage au repos ou en transit, et nécessitent des précautions spéciales lors d'un échange avec le navigateur.

Au cours des dernières années, cela a été l'attaque la plus courante. La faille la plus fréquente est simplement de ne pas chiffrer les données sensibles. Lorsque la cryptographie est utilisée, la génération et la gestion de clés faibles et la faible utilisation d'algorithmes, de protocoles et de chiffrements sont courants, en particulier pour les techniques de stockage de hachage avec mot de passe faible. Généralement, ces informations incluent des informations personnelles sensibles telles que des dossiers médicaux, des informations d'identification, des données personnelles et des cartes de crédit, qui nécessitent souvent une protection telle que définie par les lois ou réglementations telles que le [GDPR](#) ou les lois locales sur la confidentialité.

Classer les données traitées, stockées ou transmises par une application. Identifiez les données sensibles en fonction des lois sur la confidentialité, des exigences réglementaires ou des besoins de l'entreprise. Appliquer les contrôles selon la classification.

iv) Entités externes [XML](#)

Les entités externes XML sont connues en anglais sous le nom **XML External Entities**. 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.

Par défaut, de nombreux processeurs XML plus anciens permettent la spécification d'une entité externe, un [URI](#) qui est déréférencé et évalué pendant le traitement XML. Ces failles peuvent être utilisées pour extraire des données, exécuter une requête à distance à partir du serveur, analyser des systèmes internes, effectuer une attaque par déni de service et exécuter d'autres attaques.

Autant que possible, utilisez des formats de données moins complexes tels que [JSON](#) et évitez la sérialisation des données sensibles. Corrigez ou mettez à niveau tous les processeurs et bibliothèques XML utilisés par l'application ou sur le système d'exploitation sous-jacent.

v) **Contrôle d'accès brisé**

Le contrôle d'accès brisé est appelé en anglais **Broken Access Control**. Les restrictions sur ce que les utilisateurs authentifiés sont autorisés à faire ne sont souvent pas correctement appliquées. Les attaquants peuvent exploiter ces failles pour accéder à des fonctionnalités et / ou données non autorisées, telles que l'accès aux comptes d'autres utilisateurs, l'affichage de fichiers sensibles, la modification des données d'autres utilisateurs, la modification des droits d'accès, etc.

Les faiblesses du contrôle d'accès sont courantes en raison du manque de détection automatisée et du manque de tests fonctionnels efficaces par les développeurs d'applications. L'impact technique est celui des attaquants agissant en tant qu'utilisateurs ou administrateurs, ou des utilisateurs utilisant des fonctions privilégiées, ou créant, accédant, mettant à jour ou supprimant chaque enregistrement.

Le contrôle d'accès n'est efficace que s'il est appliqué dans un code côté serveur approuvé ou dans une API sans serveur, où l'attaquant ne peut pas modifier la vérification du contrôle d'accès ou les métadonnées. À l'exception des ressources publiques, refus par défaut. Implémentez les mécanismes de contrôle d'accès une fois et réutilisez-les dans toute l'application, y compris en minimisant l'utilisation de [CORS](#).

vi) **Mauvaise configuration de la sécurité**

La mauvaise configuration de la sécurité est appelée en anglais **Security Misconfiguration**. La mauvaise configuration de la sécurité est le problème le plus souvent rencontré. Ceci est généralement le résultat de configurations par défaut non sécurisées, de configurations incomplètes ou ad hoc, d'un stockage cloud ouvert, d'en-têtes [HTTP](#) mal configurés et de messages d'erreur détaillés contenant des informations sensibles. Non seulement tous les systèmes d'exploitation, cadres, bibliothèques et applications doivent être configurés de manière sécurisée, mais ils doivent également être corrigés / mis à niveau en temps opportun.

Une mauvaise configuration de sécurité peut se produire à n'importe quel niveau d'une pile d'application, notamment les services réseau, la plateforme, le serveur Web, le serveur d'applications, la base de données, les frameworks, le code personnalisé et les machines virtuelles pré-installées. De telles failles donnent souvent aux attaquants un accès non autorisé à certaines données ou fonctionnalités du système. Parfois, de tels défauts entraînent un compromis complet du système.

Des processus d'installation sécurisés devraient être mis en œuvre. Par exemple, un processus de renforcement répétable qui permet de déployer rapidement et facilement un autre environnement correctement verrouillé. Les environnements de développement, d'assurance qualité et de production doivent tous être configurés de manière identique, avec des informations d'identification différentes utilisées

dans chaque environnement. Ce processus devrait être automatisé pour minimiser les efforts requis afin d'installer un nouvel environnement sécurisé.

vii) Cross-Site Scripting (XSS)

Les failles XSS se produisent chaque fois qu'une application inclut des données non fiables dans une nouvelle page Web sans validation ou échappée, ou met à jour une page Web existante avec des données fournies par l'utilisateur en utilisant une API de navigateur pouvant créer du code [HTML](#) ou JavaScript. XSS permet aux attaquants d'exécuter des scripts dans le navigateur de la victime, ce qui peut détourner des sessions utilisateur, dégrader des sites Web ou rediriger l'utilisateur vers des sites malveillants. XSS est le deuxième problème le plus répandu dans le Top 10 d'OWASP, et se retrouve dans environ les deux tiers de toutes les applications.

L'impact de XSS est modéré pour XSS réfléchi et DOM XSS, et sévère pour XSS stocké, avec l'exécution de code à distance sur le navigateur de la victime, comme voler des informations d'identification, des sessions, ou livrer des logiciels malveillants à la victime.

La prévention de XSS nécessite la séparation des données non fiables du contenu du navigateur actif. Cela peut être réalisé en utilisant des frameworks qui échappent automatiquement à XSS par conception, comme le dernier Ruby on Rails, React JS. Apprenez les limites de la protection XSS de chaque framework et gérez correctement les cas d'utilisation qui ne sont pas couverts. L'échappement de données de requête HTTP non fiables en fonction du contexte de la sortie HTML (corps, attribut, JavaScript, CSS ou [URL](#)) résoudra les vulnérabilités XSS Reflected⁷ et Stored⁸

viii) Désérialisation non sécurisée

La désérialisation non sécurisée est appelée en anglais **Insecure Deserialization**. Cette menace 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.

Certains outils peuvent détecter des défauts de désérialisation, mais une assistance humaine est souvent nécessaire pour valider le problème. L'impact des défauts de désérialisation ne peut pas être sur-estimé. Ces failles peuvent mener à des attaques d'exécution de code à distance, l'une des attaques les plus graves possibles.

Le seul modèle architectural sûr est de ne pas accepter les objets sérialisés provenant de sources non fiables ou d'utiliser des supports de sérialisation qui autorisent uniquement les types de données primitifs. Si cela n'est pas possible, il faut faire une implémentation de contrôles d'intégrité tels que les signatures numériques sur tous les objets sérialisés pour empêcher la création d'objets hostiles ou la falsification de données.

⁷la requête envoyée vers le serveur contient le script malveillant qui est ensuite retourné et exécuté par le navigateur

⁸injection du script conservé en permanence par l'application cible

ix) Utilisation de composants avec des vulnérabilités connues

L'utilisation de composants avec des vulnérabilités connues est appelée en anglais **Using Components with Known Vulnerabilities**. Les composants, tels que les bibliothèques, les frameworks et autres modules logiciels, fonctionnent avec les mêmes privilèges que l'application. Si un composant vulnérable est exploité, une telle attaque peut faciliter la perte de données sérieuse ou la prise de contrôle du serveur. Les applications et les API utilisant des composants présentant des vulnérabilités connues peuvent compromettre les défenses de l'application et permettre diverses attaques et impacts.

La prévalence de ce problème est très répandue. Les modèles de développement à forte composante peuvent amener les équipes de développement à ne plus comprendre quels composants ils utilisent dans leur application ou leur API, et encore moins les tenir à jour. Bien que certaines vulnérabilités connues n'entraînent que des impacts mineurs, certaines des violations les plus importantes à ce jour reposent sur l'exploitation de vulnérabilités connues dans les composants. Selon les actifs que vous protégez, ce risque devrait peut-être figurer en tête de liste.

Un processus de gestion des correctifs devrait être mis en place pour supprimer les dépendances non utilisées, les fonctions inutiles, les composants, les fichiers et la documentation. Abonnez-vous à des alertes par e-mail pour connaître les failles de sécurité liées aux composants que vous utilisez. N'obtenez que des composants de sources officielles sur des liens sécurisés. Préférer les packages signés pour réduire les risques d'inclusion d'un composant malveillant modifié.

x) Insuffisance de Logs et de Surveillance

L'insuffisance de Logs et de surveillance est appelée en anglais **Insufficient Logging & Monitoring**. 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 dépasse 200 jours, généralement détectés par des parties externes plutôt que par des processus internes ou de surveillance.

Une stratégie pour déterminer si vous avez une surveillance suffisante est d'examiner les journaux après les tests de pénétration. Les actions des testeurs doivent être enregistrées suffisamment pour comprendre les dommages qu'ils ont pu infliger. La plupart des attaques réussies commencent par un sondage de vulnérabilité. Permettre de telles sondes de continuer peut augmenter la probabilité d'exploitation réussie à près de 100%.

En fonction du risque de stockage ou de traitement des données par l'application : Assurez-vous que tous les échecs de connexion, de contrôle d'accès et de validation des entrées côté serveur peuvent être consignés avec un contexte utilisateur suffisant pour identifier les comptes suspects ou malveillants, et conservés suffisamment longtemps pour permettre une analyse légale retardée. Assurez-vous que les journaux sont générés dans un format pouvant être facilement utilisé par des solutions de gestion de journaux centralisées.

Conclusion

Les attaques informatiques sont nombreuses et de différentes formes. Dans ce chapitre, nous avons exposé le fonctionnement de quelques-unes d'entre elles après avoir présenté l'ERP utilisé par la SBEE pour la gestion des factures et bien d'autres entités. Dans le chapitre suivant, nous aborderons notre solution à travers sa conception et les outils utilisés pour sa réalisation.

Pour conclure, la sécurité des applications web est un point qu'il ne faut pas négliger. C'est un travail qui peut paraître fastidieux, coûteux, pas forcément très utile pour des petites structures mais quoi de mieux pour la sérénité et la satisfaction client que de savoir que son application est robuste et ne flanchera pas sous les attaques du premier pirate venu ?

Conception et Matériels

Introduction

La programmation d'une application requiert d'abord l'organisation et la documentation de ses idées. La définition des modules induit les différentes étapes de sa réalisation. C'est cette démarche antérieure à l'écriture que l'on appelle modélisation. Ainsi, notre modélisation est axée autour de deux diagrammes : le diagramme des cas d'utilisation recense les différentes fonctionnalités de notre système ; le diagramme de séquence illustre le fonctionnement interne du système dans le temps.

2.1 Conception

2.1.1 Méthode de modélisation

Afin de modéliser les fonctionnalités de notre solution, nous avons choisi le langage UML *Unified Modeling Language* [6]. Issu d'un large consensus, le langage UML garantit la stabilité et la performance d'un projet grâce à son caractère formel et industrialisé. Aussi facilite-t-il la compréhension du système par l'usage de représentations graphiques appelées diagrammes. Ces derniers nous ont permis de modéliser notre solution en utilisant les diagrammes de cas d'utilisation et de séquence.

2.1.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.

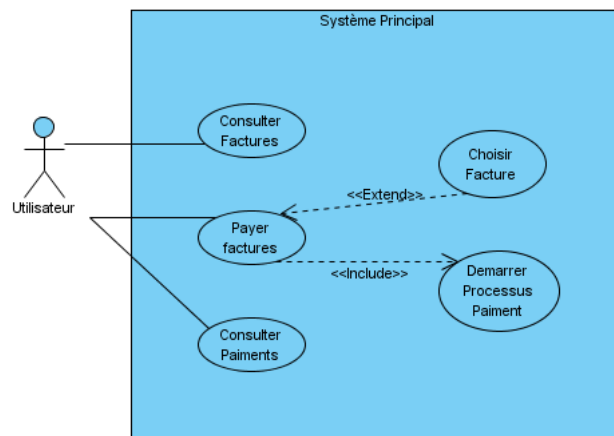


FIGURE 2.1 – Diagramme de cas d'utilisation

Pour une première utilisation, il est nécessaire de créer un compte. Les informations à fournir sont : un nom, un email, un numéro de téléphone, la référence abonnée et le mot de passe pour la protection du compte. Ces informations sont obligatoires. Un mail d'activation est envoyé et l'utilisateur après activation de son compte peut se connecter à l'application. Une authentification est nécessaire afin d'avoir accès aux cas d'utilisations de notre diagramme (ou aux fonctionnalités de l'application). Sur notre diagramme de cas d'utilisation, nous pouvons identifier un client et un administrateur. Le client peut consulter et payer ses factures, et il peut aussi voir ses paiements. L'administrateur quant à lui gère les comptes abonnés et à accès à certaines informations comme le nombre de factures déjà vendues au cours de la journée et le montant total.

2.1.3 Diagramme de séquence

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur. Ce mode de représentation effectue la description du fonctionnement dynamique du système. En d'autres termes, il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre.

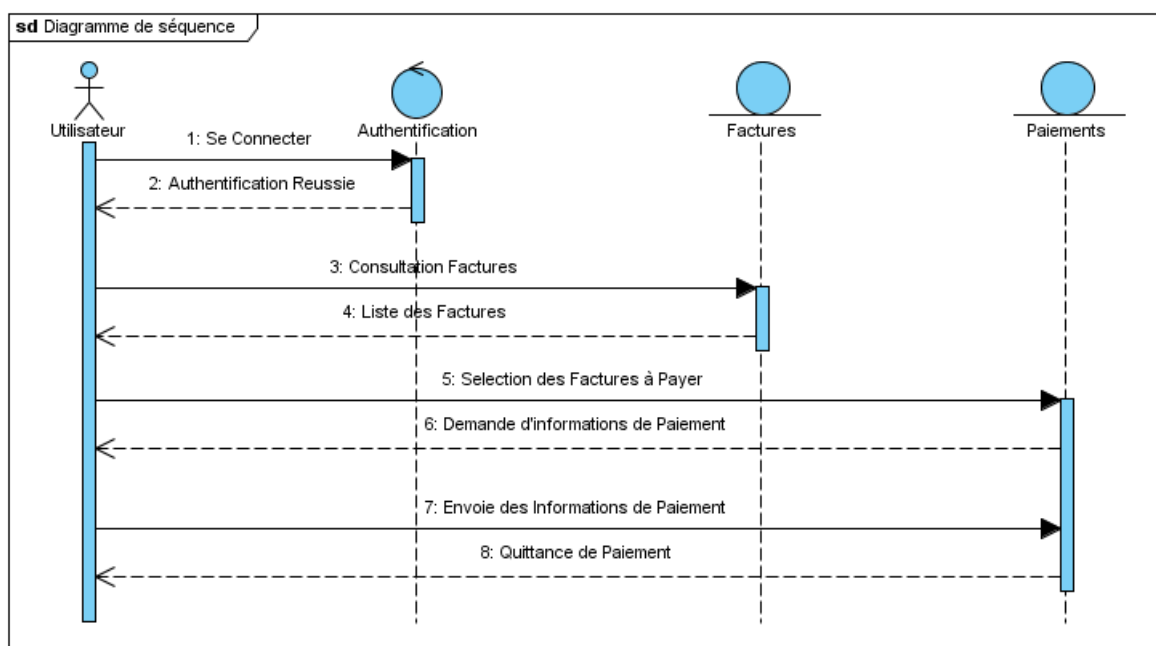


FIGURE 2.2 – Diagramme de séquence

Après que l'utilisateur ait entré ses informations, identifiants pour se connecter, il a accès aux factures. Là il peut ajouter les factures impayées à la liste des factures à payer (même concept que le panier d'un site d'e-commerce). Il pourra donc procéder au paiement de ses factures via le mode de paiement qui lui convient. Une quittance lui est générée et est ajoutée à son historique de paiements.

2.2 Matériels

2.2.1 Principe de fonctionnement de notre solution

Les factures d'un utilisateur sont identifiées grâce à la référence abonnée fournie à l'inscription. Une fois connecté à l'application, le client dispose de la liste de ses factures. Il peut donc sélectionner la ou les factures qu'il désire solder et procéder au paiement. Le progiciel Gd'Or génère tous les jours à une heure précise 00H un fichier contenant les factures impayées de tous les compteurs conventionnels de la SBEE. Ce fichier est reçu par le serveur d'application et est ensuite chargé dans la base de donnée par une tâche automatique à 00h 30min. Ainsi nous disposons des impayées au niveau des compteurs. Cependant une tâche doit être effectuée au préalable. A chaque fin de journée, à 00H 5min, notre application génère un fichier contenant toutes les factures qui ont été payées dans la journée précédente c'est à dire avant 00H. Nous considérons ce fichier comme un bilan fait par notre application à Gd'Or. Ce fichier permet à Gd'Or de valider les paiements et d'apurer les comptes en fonction des informations fournies par les prestataires de solution de paiement avant la génération des impayés du jour suivant.

En effet, les prestataires des différentes solutions de paiements sont aussi tenus de rendre des comptes à Gd'Or. Etant donné qu'il s'agit de finances, rien ne doit être pris à la légère. Gd'Or évalue minutieusement les différents fichiers reçus et génère des fichiers retours en cas d'erreurs. Les erreurs notifiées sont immédiatement signalées aux entités concernées et qui doivent y apporter des solutions le plus tôt possible. Aussi un agent vérifiera tous les matins si tout s'est bien passé la nuit précédente. Il s'agit juste d'un contrôle de routine. L'automatisation de ces tâches, nous permet d'éviter au maximum des erreurs ou des retards dus à l'interaction homme-machine. Tous les échanges de fichiers se feront via un canal sécurisé [SFTP](#) (SSH File Transfer Protocol) ou VPN (Virtual Private Network).

Les règles appliquées aux différents pare-feu et serveurs nous permettent de limiter les échanges entre les serveurs et les autres entités. Le fichier facture contenant des impayées ne peut donc pas être reçu depuis n'importe quelle destination. Gd'Or est la seule entité capable d'effectuer cette action. Après utilisation, les fichiers sont archivés et compressés de façon automatique en format .xz afin de garder une trace des échanges, des flux de données et de sauver considérablement de l'espace disque.

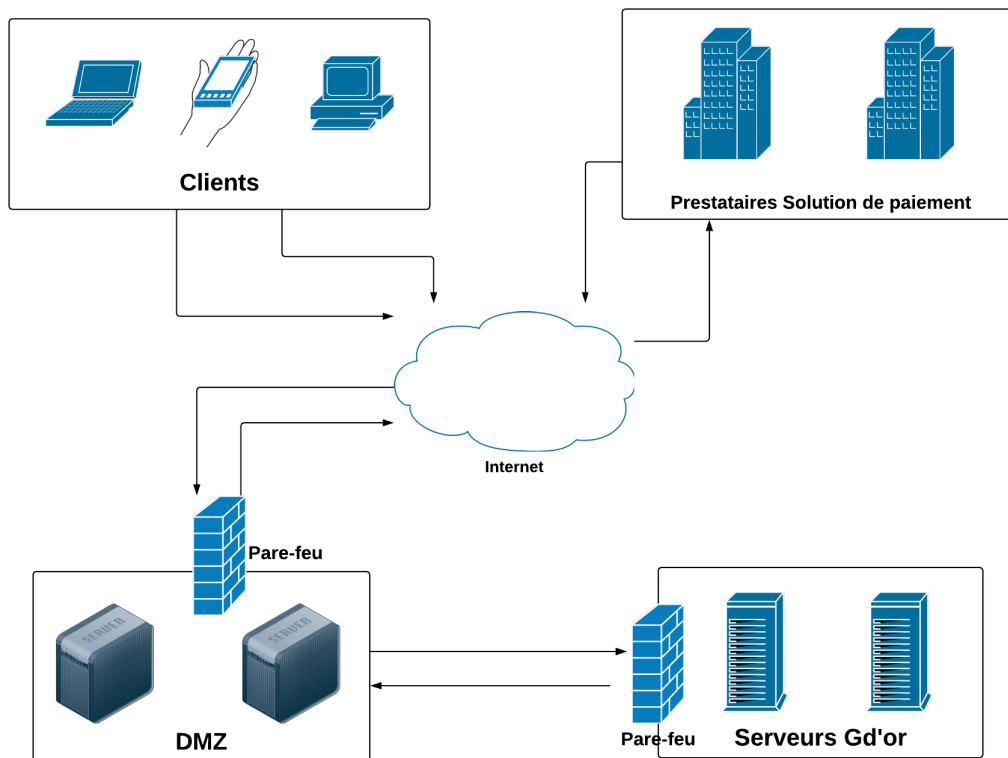


FIGURE 2.3 – Illustration du principe de fonctionnement

2.2.2 Contrôle des accès : RBAC

Il existe plusieurs façons de gérer les autorisations des utilisateurs dans un projet. Dans notre cas nous avons pensé à ce mécanisme appelé en anglais **Role Based Access Control (RBAC)**. On pourrait juste pour faire simple considérer un rôle comme un type d'utilisateur. Ce type utilisateur n'ayant pas droit à toutes les fonctionnalités de la plateforme, est limité par des permissions ou autorisations. Conceptuellement les rôles représentent une collection nommée d'autorisations. Par exemple supposons que nous ajoutons une nouvelle fonctionnalité qui permet à un utilisateur de modifier certains paramètres importants. Cette fonctionnalité doit être disponible uniquement pour les administrateurs. Dans ce cas on associe au rôle **Administrateur** cette autorisation : **Modifier tels paramètres**. Ainsi seuls les administrateurs ont la possibilité d'effectuer cette action. L'implémentation du contrôle d'accès basé sur les rôles permet de protéger les données contre les fuites, de limiter les actions, de réduire le travail de support administratif et informatique et de répondre plus facilement aux exigences d'audit.

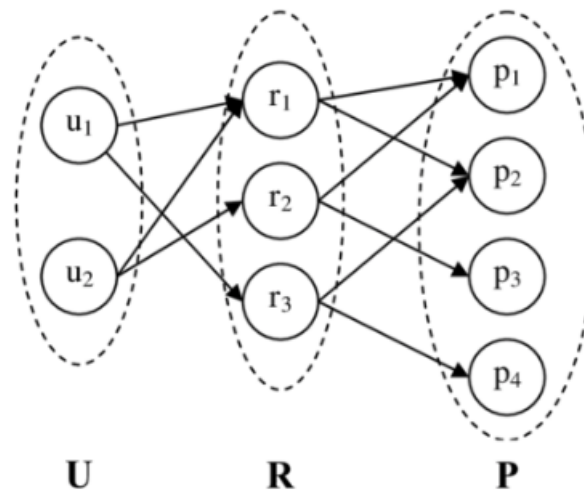


FIGURE 2.4 – Représentation Utilisateurs - Rôles - Permissions

Dans notre cas de figure nous disposons de trois types d'utilisateurs :

- **Abonné**

Les abonnés, ou clients, ou utilisateurs simples sont ceux qui possèdent un compte. Tout compte étant associé à une référence abonné, ce type d'utilisateur dispose de plusieurs permissions. Ils sont habilités à voir leurs propres factures, les payer, voir et modifier les informations de leur compte.

- **Vendeur**

Les vendeurs ont été pensés afin de permettre aux personnes incapables d'utiliser le numérique, de payer leurs factures via le même canal. Un particulier grâce à son stand pourrait procéder au paiement des factures. Il aurait la possibilité de voir les factures d'autres personnes. Cependant une discussion par rapport à la confidentialité d'une facture et à la gestion des informations contenues sur cette dernière nous permet par exemple d'aller en profondeur par rapport à ce type d'utilisateur. Pour nos livrables ce sera donc uniquement les abonnés et les administrateurs responsables du bon fonctionnement de la plateforme.

- **Administrateur**

Les administrateurs sont responsables du bon fonctionnement de la plateforme. Ils interviennent comme un support pour les abonnés et ils vérifient si les différentes tâches du serveur s'exécutent normalement et avec succès. Par défaut, ils sont aussi des abonnés. Ils possèdent alors toutes les permissions de ces derniers en plus des leurs. Leur interface leur permet de procéder à une gestion efficace de la plateforme et des utilisateurs. Par exemple les administrateurs sont les seuls à pouvoir désactiver un compte.

2.2.3 Solutions de paiement

- **Mobile Money**

MTN Mobile Money est un moyen abordable et sécurisé d'effectuer des transactions financières mobiles sans compte bancaire. Son API permet à aux utilisateurs de MTN d'acheter des biens et des services en ligne en bénéficiant de la confiance et la fiabilité associées à MTN Mobile Money. Il est actuellement disponible pour les marchands ou entreprises, qui peuvent l'intégrer sur leur site Web et application grâce à quelques lignes de code. L'interface de paiement MTN Mobile Money est rapide, facile et, surtout, complètement sécurisé. [9]

- **PayPal**

PayPal est un service qui vous permet de payer en ligne, d'envoyer et de recevoir de l'argent sans partager vos informations bancaires. Le compte PayPal vous permet de regrouper tous vos modes de paiement en ligne dans un seul portemonnaie numérique. Avec Paypal vous payer avec juste avec votre

adresse email et votre mot de passe. C'est nettement plus facile, mais aussi plus rapide et plus sûr. PayPal est disponible dans 202 pays donc le Bénin. [10]

2.2.4 Langages de développement et outils

Notre solution est une application web. Elle est donc accessible depuis tous les systèmes d'exploitations du moment qu'un navigateur et une connexion internet sont disponibles. Elle a été réalisée grâce au framework Django.

2.2.4.1 Utilisation d'un framework

Lorsque l'on réalise des sites Internet, certaines tâches sont toujours effectuées :

- réalisation et codage du design ;
- réalisation des modules :
 - réalisation du modèle de données concernant le module,
 - réalisation des formulaires d'ajout, modification et suppression des données :
- réalisation des pages d'affichage du contenu du site ;
- réalisation d'une page d'administration pour gérer les modules ;
- réalisation d'un espace utilisateur avec des droits sur l'accès aux données ;
- mise en place d'un plan du site ;

Tout cela est relativement répétitif, et si, la première fois, ça peut paraître très amusant, on en arrive rapidement à faire des copier/coller, assez mauvaise méthode car source de nombreuses erreurs. Finalement on regroupe des morceaux de code en fonctions réutilisables. À ce moment, on se rapproche de plus en plus de la notion de framework. L'avantage d'utiliser un framework existant et surtout Open Source tel que Django, c'est que nous ne sommes pas les seuls à l'utiliser, que les bugs sont donc corrigés plus rapidement, et les améliorations sont exécutées par plusieurs personnes et de manière bien mieux réfléchi. C'est d'ailleurs tout l'intérêt d'utiliser un framework. En faire moins, pour en faire plus dans le même temps.

2.2.4.2 Présentation de Django

Il existe de nombreux framework web, dans différents langages de programmation. Pourquoi utiliser spécifiquement Django et pas un autre ? Nombreuses sont les raisons qui motivent ce choix :

- la simplicité d'apprentissage ;
- la qualité des applications réalisées ;
- la rapidité de développement ;
- la sécurité de l'application ;
- la facilité de maintenance des applications sur la durée ;
- la performance et l'optimisation de l'application.

Outres ces avantages, on bénéficie de la clarté de Python, qui permet à plusieurs développeurs de travailler sur le même projet. Le style est imposé, donc tout le monde suit les mêmes règles, ce qui facilite les travaux en équipe et la clarté du code.

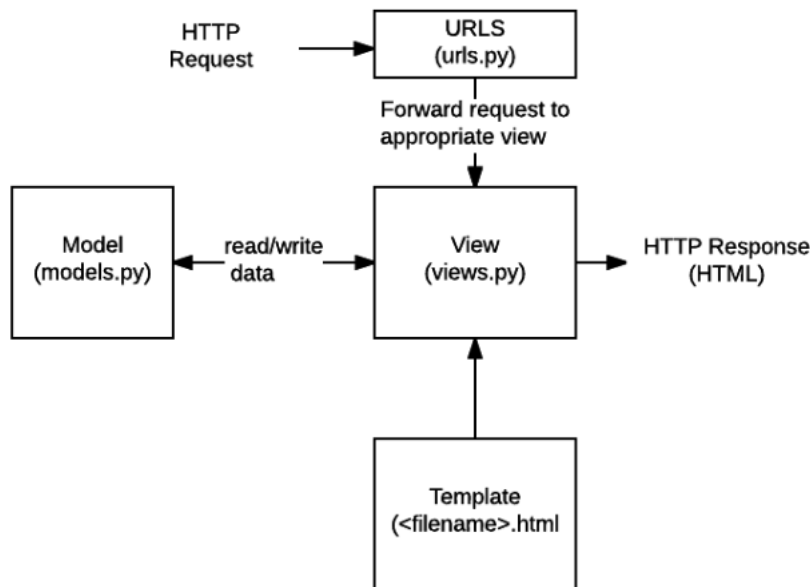


FIGURE 2.5 – Architecture MVT Django

Django se réfère à cette organisation en tant qu'architecture "Model View Template ([MVT](#))". Il présente de nombreuses similitudes avec l'architecture plus familière de Model View Controller ([MVC](#)).

2.2.4.3 Sécurité avec Django

La sécurité est un sujet d'importance capitale dans le développement d'applications Web et Django offre plusieurs outils et mécanismes de protection.

- Protection contre le « Cross site scripting » (XSS)
- Protection contre le « Cross site request forgery » ([CSRF](#))
- Protection contre l'injection SQL
- Protection contre le détournement de clic (« clickjacking »)
- Configurations [HTTPS/SSL](#)
- Validation de l'en-tête Host

Même si Django offre nativement de bonnes protections de sécurité, il est toujours important de déployer proprement les applications et de profiter des protections de sécurité du serveur Web, du système d'exploitation et d'autres composants. Le framework offre dans sa documentation plusieurs astuces et conseils pour améliorer la sécurité de ses applications.

2.2.4.4 Compression et Format XZ

XZ est un format de fichier destiné à accueillir des données compressées. C'est une spécification ouverte. Le but de ce format est d'éviter la multiplication des formats à chaque nouvel algorithme de compression. Ce format permet de choisir entre plusieurs algorithmes de compression mais aussi entre plusieurs algorithmes de vérification d'intégrité. La méthode de compression par défaut du format XZ est l'algorithme de compression LZMA2, qui est une nouvelle version de l'algorithme LZMA. Cependant XZ ne permet actuellement de choisir qu'entre ces deux algorithmes. Les algorithmes disponibles pour la vérification de l'intégrité des données compressées sont CRC-32, CRC-64 et SHA-256. Par défaut, il utilise CRC-64, qui a semblé être aux concepteurs un bon compromis entre la vitesse et la garantie d'intégrité.

2.2.4.5 PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD. En substance, cette licence dit : « Nous mettons ce logiciel à votre disposition en l'état. Faites en ce que vous voulez. Vous pouvez le modifier ou le vendre si vous le souhaitez. Nous vous demandons juste de rappeler que nous en sommes les créateurs ». Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MariaDB et Firebird), ou propriétaires (comme Oracle, MySQL, DB2 et Microsoft SQL Server). Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises.

PostgreSQL est l'une des nombreuses bases de données populaires gratuites et est fréquemment utilisée pour les bases de données Web. C'était l'un des premiers systèmes de gestion de base de données à être développé, et il permet aux utilisateurs de gérer des données structurées et non structurées. Il peut également être utilisé sur la plupart des grandes plates-formes, y compris celles basées sur Linux, et il est assez simple d'importer des informations à partir d'autres types de bases de données.

PostgreSQL se concentre traditionnellement sur la robustesse et la fiabilité, l'intégrité des données et les fonctionnalités destinées aux développeurs d'applications. PostgreSQL dispose d'un planificateur de requêtes sophistiqué, capable de joindre efficacement un assez grand nombre de tables. Il dispose de plusieurs avantages et son efficacité n'est plus à démontrer.

- Ce moteur de gestion de base de données est évolutif et peut gérer des téraoctets de données.
- Il dispose d'une implémentation solide des spécifications SQL standard
- Il prend en charge les fonctionnalités SQL avancées telles que les expressions de table communes et les fonctions Windows
- PL / pgSQL qui est son langage de programmation procédural interagit très bien avec SQL
- Si vous êtes habitué aux mécanismes d'optimisation des performances Oracle ou MS SQL Server, PostgreSQL est votre choix
- PostgreSQL a un avantage en matière de conservation et de maintien de l'intégrité des données

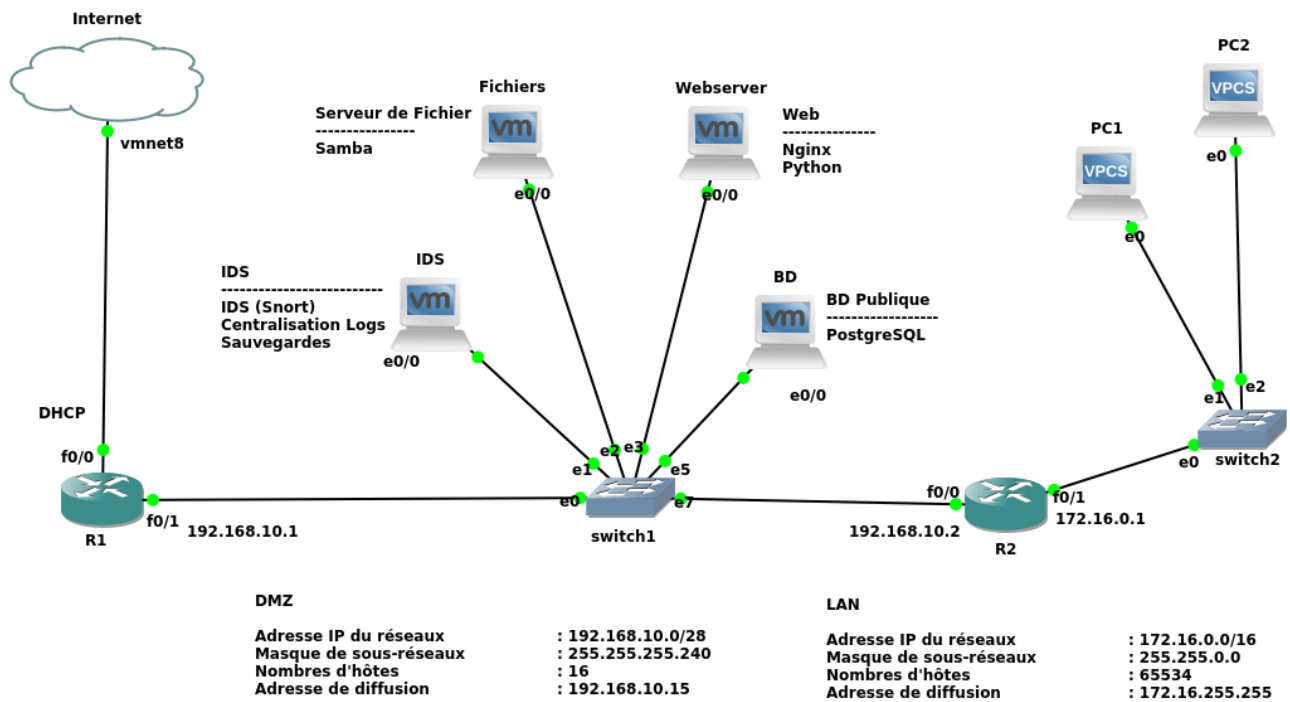
2.2.4.6 Ubuntu 16.04 LTS

Ubuntu est un système d'exploitation GNU/Linux basé sur la distribution Linux Debian. Ubuntu 16.04 [LTS](#) est la 6ème version LTS¹ de ce système d'exploitation. Elle apporte plusieurs nouvelles fonctionnalités et améliorations. Compte tenu de leur cycle de support, les versions LTS conviennent mieux aux entreprises et aux utilisateurs finaux qui n'aiment pas mettre à jour leur système d'exploitation de temps en temps. Il est simple et est constitué de milliers de paquets. Les paquets sont des composants logiciels précompilés conçus pour s'installer facilement sur la machine hôte. Son aspect libre permet à plusieurs programmeurs de pouvoir identifier les failles de sécurité ou de créer plusieurs modules pour faire des tâches qui s'avèrent indispensables. Tous nos travaux ont été réalisés sous ce système d'exploitation.

2.2.5 Environnement de Production

L'architecture est la façon dont les composants d'une chose s'organisent. S'agissant d'un système d'information en réseau, l'objectif est d'organiser et d'exploiter ce système de manière à pouvoir le contrôler les entités et à détecter des activités inattendues, indésirables et malveillantes. Actuellement, la SBEE ne dispose pas d'une DMZ au niveau de son architecture réseau. Sur la figure suivante, nous proposons la mise en place d'une DMZ afin de mieux assurer la bon fonctionnement et la sécurité de tout le système après la mise en production de la solution, qui sera géré à l'interne.

¹Long Term Support

FIGURE 2.6 – Topologie du réseau²

2.2.5.1 Serveur Web

Le terme de serveur Web peut en général se référer à deux choses différentes : soit au **logiciel d'un serveur Web**, soit à la **machine** sur laquelle s'exécute le programme. Lorsqu'il s'agit de la seconde définition, on parle généralement d'hébergeur ou d'hôte (un tel hébergeur peut abriter plusieurs programmes de serveur Web). Dans la suite de ce guide, nous parlerons de **logiciels de serveurs Web** (ou programmes) ou d'**hébergeurs** (hôtes) pour distinguer ces deux définitions.

1. Nginx

Chaque site Web avec un trafic croissant ou des pics de trafic importants est vulnérable aux problèmes de performances et aux temps d'arrêt, qui surviennent souvent au pire des moments, c'est-à-dire aux heures les plus chargées. En outre, presque tous les sites Web souffrent de problèmes de performances et de temps d'arrêt au fur et à mesure que le volume de trafic augmente régulièrement ou qu'ils subissent des pics d'utilisation importants. Nginx a été initialement développé pour résoudre le problème C10K³, c'est-à-dire pour prendre en charge facilement 10 000 connexions simultanées ou plus. L'utilisation de Nginx comme serveur Web pour notre application Python nous offre de meilleures performances.

²Réalisé par moi même avec GNS3

³Problème qu'avait les autres serveurs à gérer 10 000 connexions simultanées

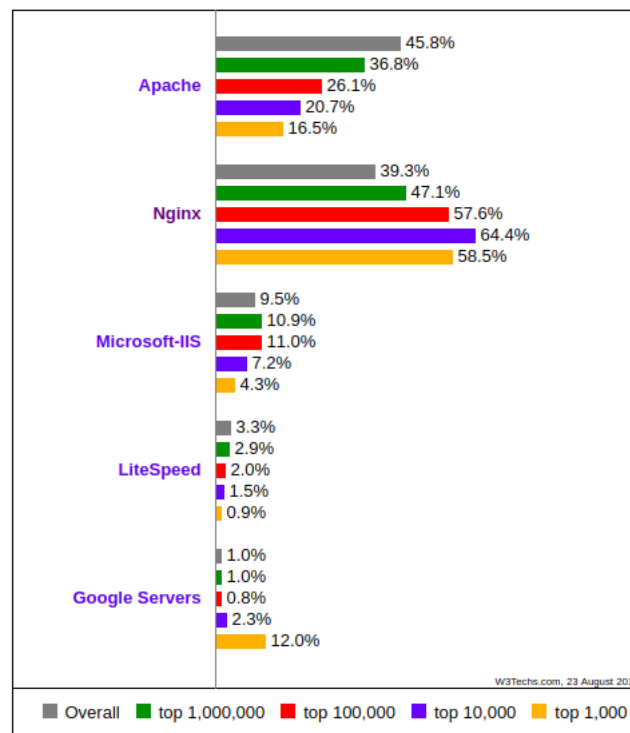


FIGURE 2.7 – Pourcentages de sites Web utilisant différents serveurs Web [14]

NGNIX améliore les performances d'un site Web de trois manières différentes :

- En tant que serveur Web
- En tant que serveur proxy inverse
- En tant qu'équilibreur de charge pour plusieurs serveurs d'applications

Plusieurs sites à succès utilisent nginx : Disqus, Instagram, Twitch, Wordpress, Pinterest, etc.

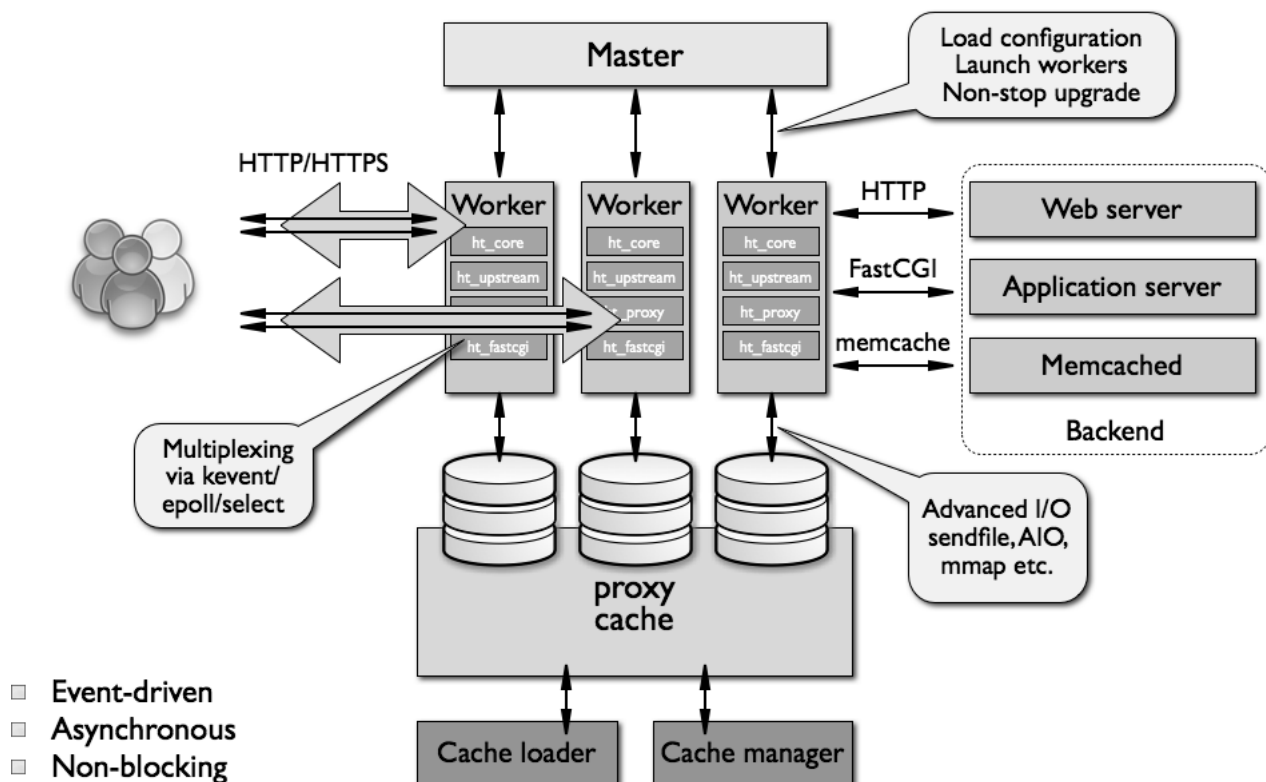


FIGURE 2.8 – Architecture de NGINX[11]

Dans le schéma, un serveur d'applications Python s'inscrit dans le bloc du serveur d'applications du serveur principal, et FastCGI l'a montré. NGINX ne sait pas comment exécuter Python, il a donc besoin d'une passerelle vers un environnement qui le fait. FastCGI est une interface largement utilisée pour Python, PHP et d'autres langages.

2. Python

Python est un langage de programmation populaire. Il a été créé en 1991 par Guido Van Rossum. Il est utilisé pour :

- le développement web (côté serveur),
- le développement de logiciels,
- les mathématiques et bien d'autres choses.

Django est un framework Web Python qui encourage un développement rapide et une conception propre et pragmatique. Conçu par des développeurs expérimentés, il prend en charge une grande partie des problèmes de développement Web, vous pouvez donc vous concentrer sur l'écriture de votre application sans avoir à réinventer la roue. Il est gratuit et open source. Django prend la sécurité très au sérieux et aide les développeurs à éviter de nombreuses erreurs de sécurité courantes. Pour notre application, nous avons utilisé la version 1.11 qui est une version LTS et la version 3.6 de Python.

2.2.5.2 Serveur de base de données

PostgreSQL est un système de gestion de base de données à usage général et relationnel. PostgreSQL est un logiciel gratuit et open source. Il nécessite des efforts minimums de maintenance grâce à sa stabilité. Par conséquent, pour des applications basées sur PostgreSQL, le coût total relatif à la maintenance est faible par rapport aux autres systèmes de gestion de base de données. Nous utiliserons la version 10 de PostgreSQL pour la mise en place de notre base de données.

2.2.5.3 Serveur de Fichiers

Un serveur de fichiers permet de partager des données à travers un réseau. Il possède généralement une grande quantité d'espace disque où sont déposés des fichiers. Les utilisateurs peuvent ensuite les récupérer au moyen d'un protocole de partage de fichier. Nous utiliserons Samba pour la mise en place de notre serveur de fichiers. Samba est un outil qui permet de partager des fichiers ou des dossiers entre différentes machines (ordinateur, tablettes, smartphones) qui peuvent tourner sous différents systèmes d'exploitation (Windows, OSX, Linux, Android)[13]. Cette compatibilité est son principal avantage mais il y en a bien d'autres :

- Il possède la fiabilité et la stabilité de Linux
- Il est très facile à mettre en œuvre
- Il est sûr
- Il est gratuit

2.2.5.4 Intrusion Detection System (IDS)

Disposer d'un IDS ou d'un IPS est essentiel dans une architecture de passerelle sécurisée. Généralement, l'IDS/IPS s'appuie sur une base de données de signatures pour détecter les intrusions potentielles ou les violations de la politique de sécurité, comme l'utilisation de protocoles non autorisés. La base de données de signatures dans un IDS est comparable à celle utilisée dans un système de détection de virus, notamment en cela qu'il ne produira aucune alerte pour une signature d'intrusion absente de sa base de données. Celle-ci doit donc être mise à jour régulièrement, tout comme avec un système de détection de logiciels malveillants.

- **Snort**

Snort est capable d'effectuer en temps réel des analyses de trafic et de logger les paquets sur un réseau IP. Pour effectuer ces analyses, Snort se fonde sur des règles [3]. Il est fourni avec certaines règles de base mais cependant, comme tout logiciel, Snort n'est pas infaillible et demande donc une mise à jour régulière. Snort peut également être utilisé avec d'autres projets open sources tels que SnortSnarf, ACID, sguil et BASE afin de fournir une représentation visuelle des données concernant les éventuelles intrusions. Il est l'un des plus actifs NIDS Open Source et possède une communauté importante contribuant à son succès. Snort sera installé et configuré sur une machine à l'entrée de la DMZ avant le switch. Nous pourrions détecter les attaques qui n'ont pas été filtrées par le pare-feu et qui relèvent d'un certain niveau de compétence. Aussi les logs seront ici plus clairs à consulter puisque les attaques bénignes⁴ ne seront pas recensées.

- **Fail2ban**

Fail2ban est un HIDS qui bloque les adresses IP appartenant à des hôtes qui tentent de casser la sécurité du système. Il lit les logs de divers services (SSH, Apache, FTP) à la recherche d'erreurs d'authentification répétées et ajoute une règle iptables pour bannir l'adresse IP de la source [5]. Il est capable de réduire le taux de tentatives d'authentification incorrectes, mais il ne peut pas éliminer le risque que présente une authentification faible. La grande force de Fail2Ban est sa grande modularité que cela soit au niveau des mécanismes de détections basées sur les expressions régulières ou sur les actions à mener qui peuvent aller de l'expédition d'un mail à la mise en place de règles de Firewall. Fail2ban sera installé et configuré sur tous les serveurs de notre architecture.

2.2.5.5 Centralisation des journaux d'événements (logs)

Les logs sont des éléments indispensables à toutes les applications pour comprendre le fonctionnement, analyser, diagnostiquer et intervenir en conséquence. Le dossier "/var/log" stocke l'ensemble des logs systèmes et applicatifs sous Linux via rSysLog. La centralisation des journaux d'événements permet d'avoir une vue d'ensemble d'éléments cruciaux à la bonne gestion d'un SI pour y mener des traitements. En cas de crash ou

⁴Trop faciles

de suppression des logs sur une entité, elle permet de diagnostiquer un crash et de garantir la survie des logs à une suppression. La centralisation des logs, lorsqu'elle est couplée à des outils d'analyse, de traitements, d'indexation et encore mieux, de graphage, permet d'avoir de toutes ces lignes d'information un ensemble cohérent de données qu'il est possible de corrélérer. Notre choix s'est donc porté sur Elastic Stack (ELK). Il est composé de trois outils dont Elasticsearch (indexation et recherche de données), LogStash (récolte et traitement des logs) et Kibana (visualisation de donnée sous formes de graphiques).

2.2.5.6 Les règles Iptables

Un pare-feu (appelé aussi coupe-feu, garde-barrière ou firewall en anglais), est un système permettant de protéger un ordinateur ou un réseau d'ordinateurs des intrusions provenant d'un réseau tiers en filtrant les flux de données. Lorsqu'un paquet IP rencontre une chaîne de règles dans son cheminement à travers le kernel, celui-ci vérifie si certaines règles sont respectées :

- Dès qu'une règle s'applique à un paquet, l'action prévue dans la règle est effectuée : transmettre le paquet, le supprimer ou le renvoyer au destinataire.
- Lorsqu'aucune des règles ne peut s'appliquer pour le paquet, c'est la politique par défaut qui entre en vigueur. Là encore, on peut se retrouver avec les trois cas de figure : transmettre, supprimer, rejeter.

La configuration d'un pare-feu consiste donc à définir la politique par défaut ainsi qu'une série de règles pour chacune des chaînes de filtres essentielles[12]. Iptables est une solution complète et fiable de pare-feu et il dispose de très nombreuses options qui permettent de faire du filtrage très fin. Précisons dès maintenant que le module qui fournit au noyau Linux les fonctions de pare-feu, de partage de connexions internet (NAT) et d'historisation du trafic réseau s'appelle Netfilter. iptables est en fait juste l'outil qui permet à un administrateur de configurer Netfilter en mode utilisateur. Pour notre architecture, nous bloquons d'abord tout le trafic entrant par défaut. Ensuite nous autorisons au cas par cas : le trafic appartenant ou lié à des connexions déjà établies et le trafic à destination des serveurs (web, ssh, etc.) que nous mettons à disposition.

2.2.5.7 Ubuntu Server 18.04 LTS

Ubuntu 18.04 LTS est la dernière version LTS mise en ligne le 26 Avril 2018. Elle apporte plusieurs nouvelles fonctionnalités et améliorations dont l'installateur du système d'exploitation (Subiquity Installer). Ubuntu serveur ne dispose pas par défaut d'une interface graphique. Tous les serveurs entrant en jeu dans notre travail tournent sous ce système d'exploitation.

Conclusion

Dans ce chapitre, nous avons présenté les différents choix opérés ainsi que notre solution à travers sa modélisation, son principe de fonctionnement et les outils utilisés. Le solution proposée a été réalisée grâce au framework Django, très robuste et performant. Ensuite pour l'organisation et la structuration des différentes données, nous utilisons PostgreSQL, qui est un système de gestion de base de données relationnelle complet, stable, performant, riche de nombreuses années de développement, en évolution constante et soutenu par une communauté active. Enfin vu que l'architecture existante à la SBEE ne contient pas de DMZ nous en avons proposée une autre qui comble ce manque et qui nous donne un aperçu de l'environnement de production puisque le tout sera géré à l'interne. Le chapitre suivant exposera les différents résultats et quelques critiques.

Résultats et Discussion

Introduction

Dans ce chapitre, nous présentons les résultats des simulations faites pour tester le fonctionnement de notre solution. Dans un premier temps, nous allons présenter l'application, une attaque pour tester la sécurité de notre réseau et nous aborderons ensuite une discussion.

3.1 Présentation des résultats des tests

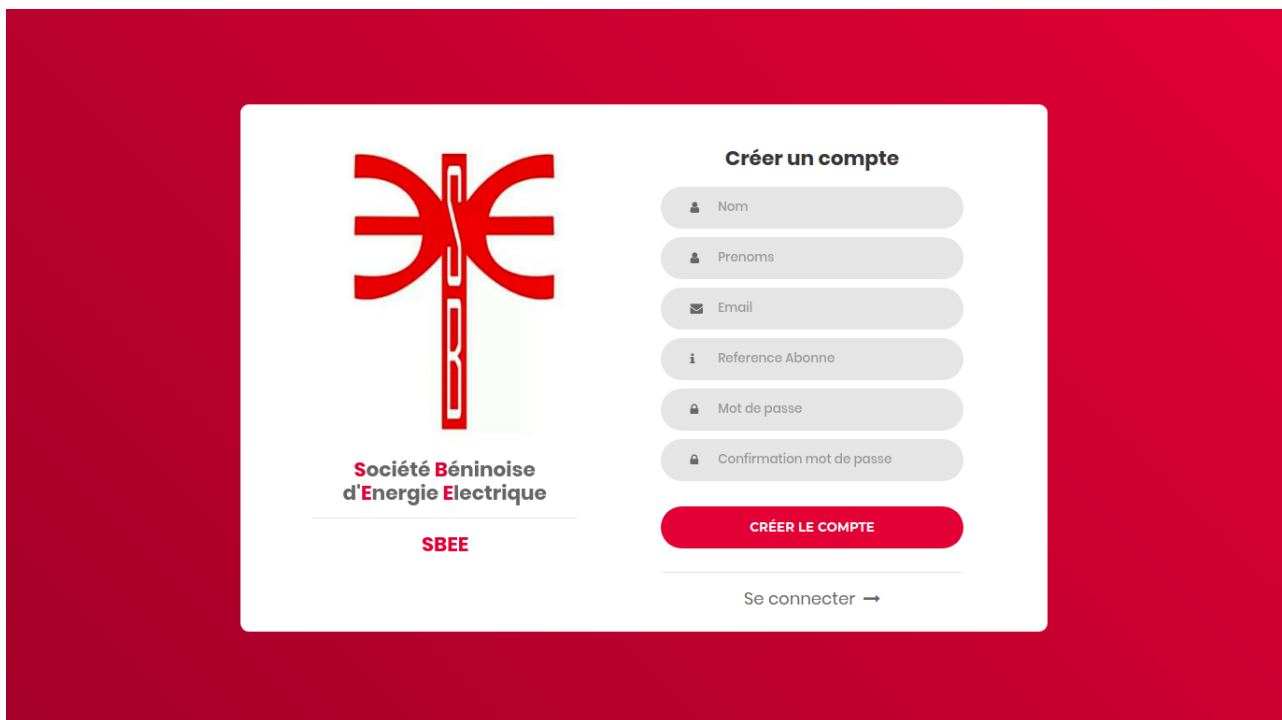


FIGURE 3.1 – Page d'inscription de l'application

La figure 3.1 présente la page d'ouverture de compte de notre application. L'utilisateur entre son nom, prénom(s), adresse email, la référence abonnée et le mot de passe. Un mail d'activation lui est envoyé et il est

redirigé vers la page de connexion.

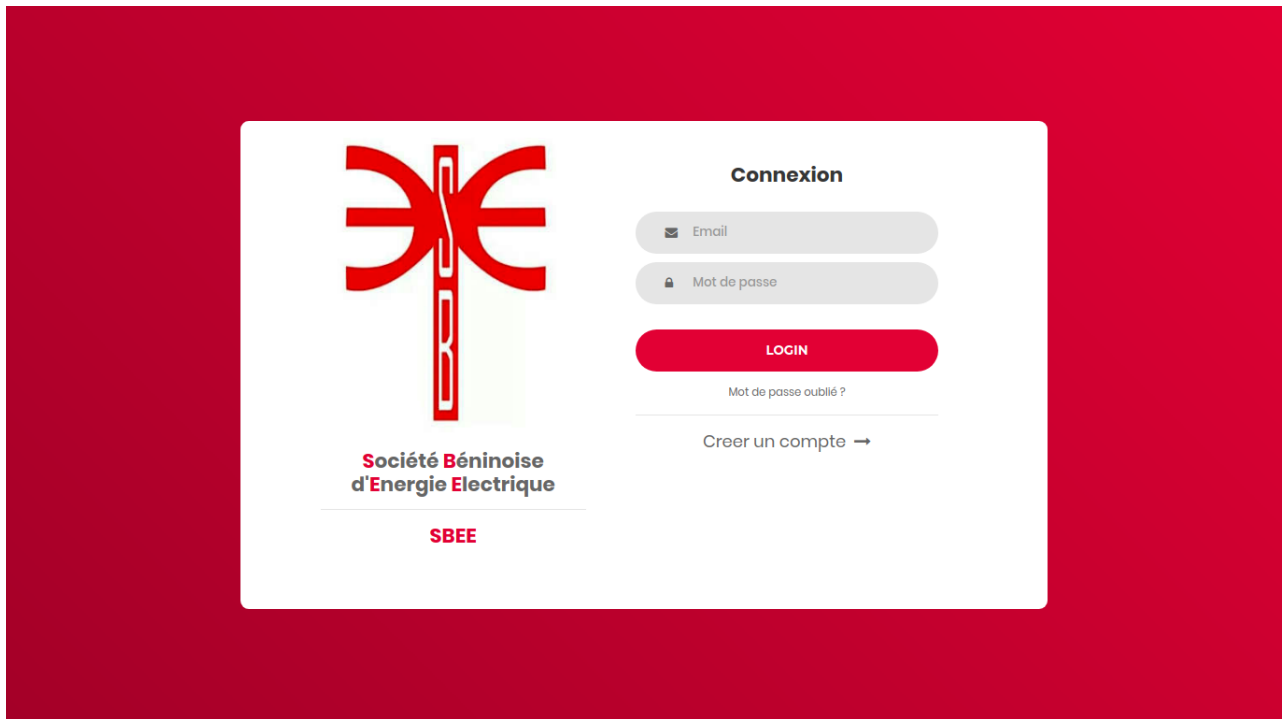


FIGURE 3.2 – Page de connexion

L'utilisateur entre son email et son mot de passe pour se connecter à l'application.

3.1.1 Interfaces Abonné

Ce tableau contient la liste de vos factures

Show 10 entries Search:

Référence	Période	Date Limite	Abonné	Montant	Options
110204410561NA	May 2018	29 Jun 2018	COCOU VINCENT	1586 FCFA	Voir

Showing 1 to 1 of 1 entries Previous 1 Next

Copyright © 2018 SBEE. Version 1.0

FIGURE 3.3 – Liste des factures de l'abonné

Tous les utilisateurs ont accès à la liste de leurs factures. Ils peuvent cliquer sur le bouton **Voir** pour voir la facture sous le format habituel qu'ils connaissent (le format d'impression et de distribution), ou juste afficher les informations sous forme de liste.

Facture du 5 / 2018

[Ajouter la facture](#)

[Vue Graphique](#) [Details](#)

Votre facture

SOCIETE BENINOISE D'ENERGIE ELECTRIQUE
ETABLISSEMENT PUBLIC NATIONAL
01 B.P. 123 COTONOU C.C.P.N° 71-58 - ECOBANK : 050110039 - N° IFU : 3200800563113

Service réclamation: 31-22-47 / 31-24-10

Merçi de payer votre facture à bonne date

FACTURE DE CONSOMMATION D'ELECTRICITE
Relevé pour la période du date ici au date la

NOM ET ADRESSE DE L'ABONNE
COCOUC VINCENT
C/1481

ZONE: NUMERO ABONNE: 110204410561NA REGT: 000000

FACTURATION DE COCOUC VINCENT
ZONE: NUMERO ABONNE: 110204410561NA REGT: 000000

NUMERO DE POLICE	PUISSANCE	TARIF	INDEX COMPTEUR		COEFF	CONSO	CONSOMMATION ELECTRIQUE (en kwh)				MONTANT CONSO	LOCATION ENTRETIEN COMPTEUR	TAXE SUR ELECTRICITE	MENSUALITE BRANCHEMENT	AUTRES LOCATIONS	MONTANT A FACTURER	VIDE A COMBLER	NUMERO DE POLICE	MONTANT A FACTURER
			ANCIEN	NOUVEAU			TRANCHE 1	TRANCHE 2	TRANCHE 3	TRANCHE 4									
VK024442	COCOUC VINCENT	BT1	22333	22345	COCOUC VINCENT	12	12				936	500	90.0			1496		VK024442	1496

CONTRIBUT* ELEC/RURALE: 36

DATE DE PRESENTATION
La présente facture ne peut tenir lieu de quittance.
Le règlement du montant total qui y est porté doit être effectué dans les 15 jours qui suivent la date de présentation.
Passé ce délai, la fourniture d'électricité sera coupée sans préavis pour non règlement de factures.
Pour le règlement se présenter au guichet de : date ici

FACTURE A PAYER AU PLUS TARD LE 29 JUN 2018

TIMBRE: ok
TVA: ok
NET A PAYER: 1586

FIGURE 3.4 – Vue d'une facture (Format d'impression)

Il s'agit d'une représentation de la facture physique que nous connaissons tous.

MENU

- Mes Factures
- Paiements
- Mon compte
- Deconnexion

[Ajouter la facture](#)

[Vue Graphique](#) [Details](#)

AGENCE PAYEUR : 12
CENTRE PAYEUR : 1
CODE PAYEUR : 110204410561NA
PANNEAU : VK024442
ANNEE DE CONSOMMATION : 2018
PERIODE DE CONSOMMATION : May
ANNEE DE FACTURATION : 2018
PERIODE DE FACTURATION : May
MONTANT CONSOMMATION : 936 FCFA
MONTANT HORS TAXE : 1496 FCFA
digits(nufmmr) : 581021201863
MONTANT SURTAXE : 24 FCFA
CONTRIBUT* ELEC/RURALE : 36
MONTANT TTC : 1586 FCFA
NET A PAYER : 1586
TYPE PIECE CONSOM. : FE
TAUX TVA : 18
MONTANT TVA : 0 FCFA
Conso facturée : 12 FCFA
NBRE JOURS CONSO : 28
ANCIEN INDEX : 22333
NOUVEL INDEX : 22345
DATE RELEVÉ : 20180522
TARIF : BT1
DATE LIMITE PAIEMENT : 20180629
NOM DU PAYEUR : COCOUC VINCENT
ADRESSE : C/1481
MONTANT LOCATION HT : 500 FCFA
MONTANT AUTRES LOCATIONS : 0 FCFA

FIGURE 3.5 – Vue d'une facture (Liste d'informations)

Ce format permet d'avoir les informations d'une façon plus lisible et plus claire.

The screenshot shows the SBEE application interface. The top navigation bar is red with the SBEE logo, a menu icon, and user information for 'COCOU Vincent'. The left sidebar contains a 'MENU' with options: 'Mes Factures', 'Paiements', 'Mon compte', and 'Deconnexion'. The main content area is titled 'Ce tableau contient la liste des factures de votre panier' and shows a 'Total: 1586.00 FCFA'. Below this is a table with columns: 'Référence', 'Période', 'Date Limite', 'Montant', and 'Options'. The table contains one entry with the reference '110204410561NA', period 'May 2018', due date '29 Jun 2018', and amount '1586 FCFA'. The 'Options' column has two buttons: 'Voir' (green) and 'Retirer la facture' (orange). Below the table is a 'Checkout' button (green). The page footer shows 'Copyright © 2018 SBEE.' and 'Version 1.0'.

Référence	Période	Date Limite	Montant	Options
110204410561NA	May 2018	29 Jun 2018	1586 FCFA	Voir Retirer la facture

FIGURE 3.6 – Liste des factures choisies

Afin de permettre de solder plusieurs factures en une fois, l'utilisateur choisit toutes les factures qu'il désire régler et ensuite il procède au checkout¹.

The screenshot shows the 'Details du compte' page in the SBEE application. The top navigation bar and left sidebar are the same as in Figure 3.6. The main content area is titled 'Details du compte' and displays user information: 'Référence Abonne : 110204410561NA', 'Nom : COCOU', 'Prenoms : Vincent', 'Email : vincent@gmail.com', and 'Téléphone : +22967929211'. Below this is a 'Modifier' button (blue). Below the user information is a section titled 'Factures impayées' (Unpaid Invoices) which contains a table with columns: 'Référence', 'Période', 'Date Limite', 'Abonné', 'Montant', and 'Options'. The table contains one entry with the reference '110204410561NA', period 'May 2018', due date '29 Jun 2018', subscriber 'COCOU VINCENT', and amount '1586 FCFA'. The 'Options' column has a 'Voir' button (green). Below the table is a 'Showing 1 to 1 of 1 entries' message and pagination controls (Previous, 1, Next).

Référence	Période	Date Limite	Abonné	Montant	Options
110204410561NA	May 2018	29 Jun 2018	COCOU VINCENT	1586 FCFA	Voir

FIGURE 3.7 – Le compte d'un utilisateur

L'utilisateur a accès aux informations de son compte. Et il peut actuellement modifier son nom, prénom et numéro de téléphone.

¹Expression anglaise : régler la note, procéder au paiement

3.1.2 Interfaces Administrateurs

Les Administrateurs ont accès à bien plus de fonctionnalités que les abonnés simples :

- La liste des utilisateurs de la plateforme (abonnés simples et administrateurs)

The screenshot shows the SBEE administrator interface. The top navigation bar is red with the SBEE logo, a hamburger menu, and user information for GOUDALO Jospy. The left sidebar contains a menu with options: Mes Factures, Paiements, Mon compte, Utilisateurs, Fichiers, and Deconnexion. The main content area is divided into two sections: 'Administrateurs' and 'Abonnés'.

Administrateurs

Show 10 entries Search:

Référence	Nom	Prénoms	Email	Activé	Options
110204410495CA	GOUDALO	Jospy	jospy@gmail.com	True	Voir
Référence	Nom	Prénoms	Email	Activé	Options

Showing 1 to 1 of 1 entries Previous 1 Next

Abonnés

Référence	Nom	Prénoms	Email	Activé	Options
110204410561NA	COCOU	Vincent	vincent@gmail.com	True	Voir
Référence	Nom	Prénoms	Email	Activé	Options

FIGURE 3.8 – Utilisateurs de la plateforme

- Les fichiers : la liste de tous les fichiers factures chargés dans la base de données ainsi que les règlements,

The screenshot shows the SBEE administrator interface with the 'Fichiers Archivés' section selected. The top navigation bar and left sidebar are the same as in Figure 3.8. The main content area displays a table of archived files.

Fichiers Archivés

Show 10 entries Search:

Nom du fichier	Montant	Lignes	Chargé le
facture_13-09-18.csv	1928042	99	Sept. 13, 2018, 10:03 p.m.
Nom du fichier	Montant	Lignes	Chargé le

Showing 1 to 1 of 1 entries Previous 1 Next

Copyright © 2018 SBEE. Version 1.0

FIGURE 3.9 – Fichiers importés

3.2 Discussion

Notre projet est né de plusieurs constats dont entre autre les pertes de temps dans les guichets de la SBEE, et les multiplications croissantes des attaques et intrusions informatiques. Des travaux n'ont pas encore été faits à l'interne dans les locaux de la SBEE pour palier à ces problèmes observés. Ceci représente donc un premier pas fait vers le paiement des factures numériques.

Ainsi, d'une part, notre solution est simple à utiliser et est utilisable depuis un navigateur web. Elle est accessible via internet et ne requiert aucun équipement ou installation spécifique chez l'utilisateur. Elle permet aux utilisateurs de consulter leurs factures et ainsi donc de suivre leur consommation plus facilement. Cependant la société choisira elle-même via un appel d'offre ses prestataires pour les différentes solutions de paiement. Nous avons eu à faire quelques recommandations dont : **Mobile Money de MTN** ainsi que le service mondialement reconnu **PayPal**.

Aussi existe-t-il des risques de double paiements lors des factures. Etant donné que Gd'Or fonctionne en mode non connecté et que les factures ne sont validées qu'après 00h, un utilisateur qui va en agence pourrait payer une facture qui a déjà été payée en ligne. Cependant ces cas seront remboursés par les acteurs des différentes solutions de paiement car Gd'Or détecte automatiquement les doublures lors de l'apurement des comptes et renvoie des fichiers contenant les erreurs aux entités correspondantes. Celle dernière procédera à un remboursement des fonds aux utilisateurs concernés.

D'autre part, l'architecture et les outils proposés nous permettent d'assurer un minimum de sécurité pour notre application et dans notre réseau. Notre stratégie se base essentiellement sur la maîtrise du flux de données et de la connaissance effective de chaque entité du réseau. Et ceci est un prérequis pour assurer la sécurité : **On ne protège bien que ce que l'on connaît**. Cependant cette architecture ne nous met pas à l'abri de toutes les attaques, mais nous protège quand même contre beaucoup d'attaques courantes, ce qui n'est pas négligeable.

Conclusion

Dans ce chapitre, nous avons exposé les résultats des tests de notre application et réalisé quelques critiques concernant ses performances et ses insuffisances. Ces insuffisances peuvent être perçues comme des perspectives afin d'améliorer le travail fait pour une utilisation plus efficiente.

Conclusion et Perspectives

Une entreprise doit apprendre à fidéliser ses clients. On dit chez nous que le client est Roi. Pour une entreprise qui fournit des services à l'échelle nationale, savoir que chaque client qui vient en agence solder sa facture perd du temps, ce n'est pas une information agréable. Ces pertes de temps peuvent être expliquées par les longues queues, les problèmes de connexion.

L'objet de ce mémoire a donc été de proposer un prototype d'un système de paiement électronique des factures d'électricité de la SBEE dans un environnement sécurisé. Il s'agit d'une part de l'application web réalisée et d'autre part de la proposition de l'environnement de la mise en production de cette application. La solution est accessible via internet et ne requiert aucun équipement ou installation spécifique chez l'utilisateur et plusieurs mesures ont été prises pour assurer son fonctionnement en toute quiétude.

Dans ce mémoire, nous avons tout d'abord fait la revue de littérature autour du progiciel permettant le paiement des factures et quelques notions par rapport à la sécurité des applications Web. Nous avons ensuite réalisé la conception de la solution et proposé une architecture pour sa mise en production. Enfin nous avons exposé les résultats et critiques de l'application réalisée.

Bien que notre solution réponde au besoin énoncé, il faut noter certaines insuffisances telles que les doublures lors des paiements des factures. La disponibilité limitée de l'application pour réduire les cas de doublures. D'une part, un autre axe de recherche en ce qui concerne les travaux futurs sera de permettre l'achat de crédit prépayé depuis la même plateforme.

Notre travail s'est déroulé dans les locaux de la SBEE. Cependant nous tenons à préciser que la Société Nationale des Eaux du Bénin (SONEB) et la Société Béninoise d'Energie Electrique ont exactement la même organisation et donc le même principe de fonctionnement. En réalité Gd'Or est aussi le logiciel métier utilisé par la SONEB. Notre travail est donc tout aussi valide pour les factures d'eau que les factures d'électricité. Une entente entre ces deux sociétés permettrait d'avoir depuis la même plateforme les deux types de factures. Ce travail devrait donc être présenté à la SONEB afin qu'ils puissent aussi en bénéficier.

Nous souhaiterions que les différentes règles par rapport aux vendeurs soient définies afin que nous puissions les intégrer à la plateforme. Cela nous permettrait de créer des emplois et d'aider les personnes non habilitées à bénéficier des services de la plateforme elles aussi.

Les cartes bancaires pouvant être choisies comme l'une des options pour les solutions de paiement, Une évaluation devrait être faite afin de permettre à l'entreprise de s'adapter et se conformer à la norme [PCI-DSS](#) relative aux paiements électroniques via les cartes de crédit.

Bibliographie

- [1] Le petit Larousse Illustré 2010
- [2] Patrick Engebretson, "Les bases du hacking", Pearson 2013
- [3] Kerry Cox, Christopher Gerg, "Sécurité réseau avec Snort et les IDS" O'Reilly 2004
- [4] David Kennedy, Jim O'Gorman, Devan Kearns et Mati Aharoni "Hacking, Sécurité et tests d'intrusion avec Métasploit", Pearson 2013
- [5] Jesse Russell and Ronald Cohn , "Fail2ban", Book on Demand 2012
- [6] Alexandre Gachet, "Introduction à UML", O'Reilly 2005

Webographie

- [7] RPG, <https://search400.techtarget.com/definition/Report-Program-Generator>, consulté le 29 Juillet 2017
- [8] Larousse, <http://www.larousse.fr/dictionnaires/francais/hacker/38812>, consulté le 28 Mai 2017
- [9] MoMo API - MTN Bénin, <https://www.mtn.bj/particuliers/mobile-money/momo-api/>, consulté le 08 Août 2018
- [10] Paypal, <https://www.paypal.com>, consulté le 08 Août 2018
- [11] Nginx, <http://www.aosabook.org/en/nginx.html>, consulté le 22 Août 2018
- [12] Iptables, <https://blog.microlinux.fr/iptables/>, consulté le 20 Août 2018
- [13] Samba, <https://openclassrooms.com/fr/courses/2929586-mettre-en-place-un-serveur-samba>, consulté le 02 Septembre 2018
- [14] W3techs, https://w3techs.com/technologies/cross/web_server/ranking, consulté le 02 Septembre 2018
